

# **FIMRWARE UPDATES**

## **USER GUIDE**

Friday, 11 October 2019

Document Revision: 2.2



## **Copyright © Silicon Labs**

All rights reserved.

Silicon Labs assumes no liability or responsibility for any errors, mistakes or inaccuracies in content. Silicon Labs reserves the right to change products or specifications without notice, and does not make any commitment to update the information herein.

Silicon Labs' products are not authorized for use as critical components in life support devices or systems.

The *Bluetooth*® word mark and logos are registered trademarks owned by the *Bluetooth*® SIG, Inc. USA.

All other trademarks and trade names listed herein belong to their respective owners.

Information is subject to change without notice.

## VERSION HISTORY

Version	Comment
1.0	First version
1.1	Minor changes
1.2	Troubleshooting updated
1.3	DFU instructions updated
1.4	Minor updates
1.5	SerialDFU instructions added
1.6	Minor improvements
1.7	Small updates
1.8	iWRAP5 updates
1.9	SerialDFU PSR clarifications
2.0	iWRAP6 updates and some style and contact info updates
2.1	Template changed
2.2	Links updates

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Firmware update over SPI</b>	<b>6</b>
2.1	Prerequisites	6
2.2	iWRAP update client	6
2.3	Firmware update with iWRAP update client	7
2.4	Firmware update with BlueFlash	9
<b>3</b>	<b>Firmware update over UART</b>	<b>10</b>
3.1	Prerequisites	10
3.2	SerialDFU	10
3.3	Firmware update with SerialDFU	11
3.4	DFUWizard	14
3.5	DFU protocol	14
<b>4</b>	<b>Troubleshooting</b>	<b>15</b>
4.1	iWRAP update client is unable to identify or connect to the module	15
4.2	SerialDFU fails to start the firmware update	15

# 1 Introduction

This document describes the basic methods of updating the firmware of Bluegiga *Bluetooth* modules. The manual also contains step-by-step instructions to perform firmware update over SPI or UART interfaces.

## 2 Firmware update over SPI

Firmware update over SPI interface is the fastest and most reliable way of updating the firmware on your *Bluetooth* module. The downside however is that you need a physical access to the SPI pins of the module and since the SPI protocol is not generally available a Windows software for performing the update. However in many cases the SPI interface is the only way to update the firmware.

### 2.1 Prerequisites

1. You need to have a physical access to the SPI interface of your *Bluetooth* module.
2. You need to have iWRAP update client installed on your PC and your PC must have an available LPT port. **You may need to manually run the \CSR\BlueSuite 2.5\drivers\win32\InstParSpi.bat file and reboot after installing in order for the LPT driver to work properly.**
3. You need to have On-board Installation Kit, which is a LPT to SPI converter cable. The cable is delivered with the Bluetooth module development kits and the schematics are available upon request.

### 2.2 iWRAP update client

iWRAP update client is a simple Windows software, which enables you to update the firmware of your *Bluetooth* module. The following chapter walks you through the firmware update with iWRAP update client.

iWRAP update client package is available for example in <https://www.silabs.com/products/development-tools/software/bluegiga-iwrap-bluetooth-software-stack/> web page.

**Note:**

iWRAP update client always restores the factory settings of your module.

**Never install the .XPV and .XDV files delivered with iWRAP update client directly using BlueFlash software, since it will corrupt the unique settings of your module.**

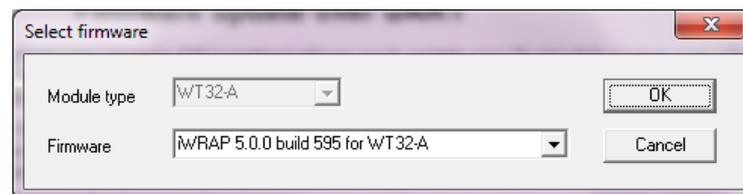
## 2.3 Firmware update with iWRAP update client

1. Connect the On-board Installation kit to the SPI interface of your *Bluetooth* module and power up the module.
2. Start the iWRAP update client.



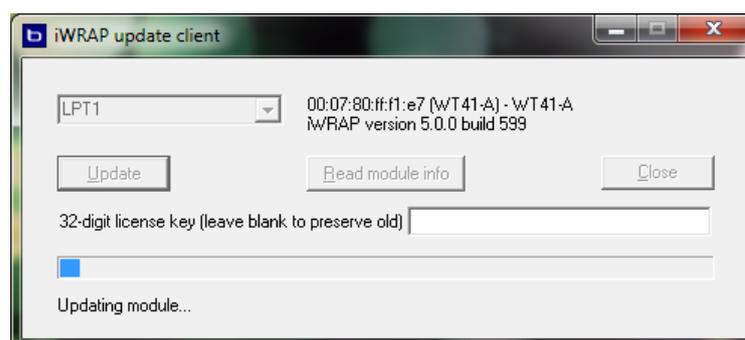
**Figure 1: iWRAP update client**

3. Press the **Update** button. iWRAP update client recognizes your module version and offers available firmware options.

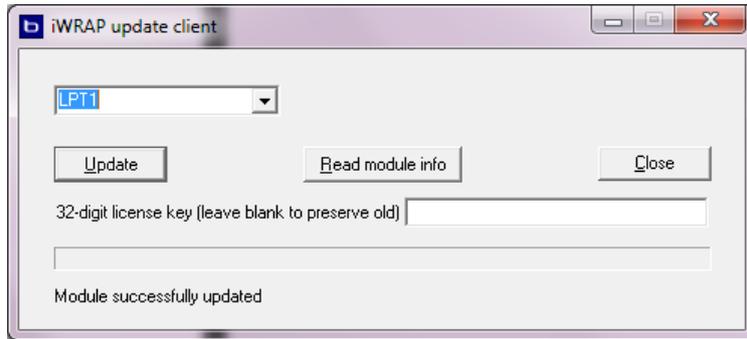


**Figure 2: Firmware selection**

4. Select one of the firmware options and press **OK**. iWRAP update client starts to update the firmware. This may take several minutes.



**Figure 3: Update in progress**



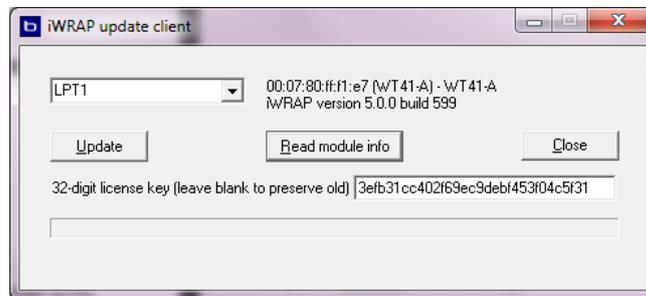
**Figure 4: Update completed**

5. Finally reset the module by powering it off.

### 2.3.1 iWRAP license key

iWRAP 5.0.0 and newer firmware version require a license key to operate. iWRAP update client software can be used to install a license key into the *Bluetooth* module. The license key needs to match the *Bluetooth* address of the device and license keys can be requested from [www.silabs.com/support](http://www.silabs.com/support).

The *Bluetooth* modules should have a license key pre-installed but if the flash of the module has been fully erased or the module does not have a proper license key, the license key can be entered to the license key box in the iWRAP update client software.



**Figure 5: Entering iWRAP license key**

## 2.4 Firmware update with BlueFlash

BlueFlash is a firmware update tool from Cambridge Silicon Radio (CSR). It can be used to update the firmware of your *Bluetooth* module in a similar fashion as with iWRAP update client.

However with BlueFlash separate firmware image files must be used. The firmware image files delivered with iWRAP update client must never be used with BlueFlash, since they will corrupt the unique settings of your module.

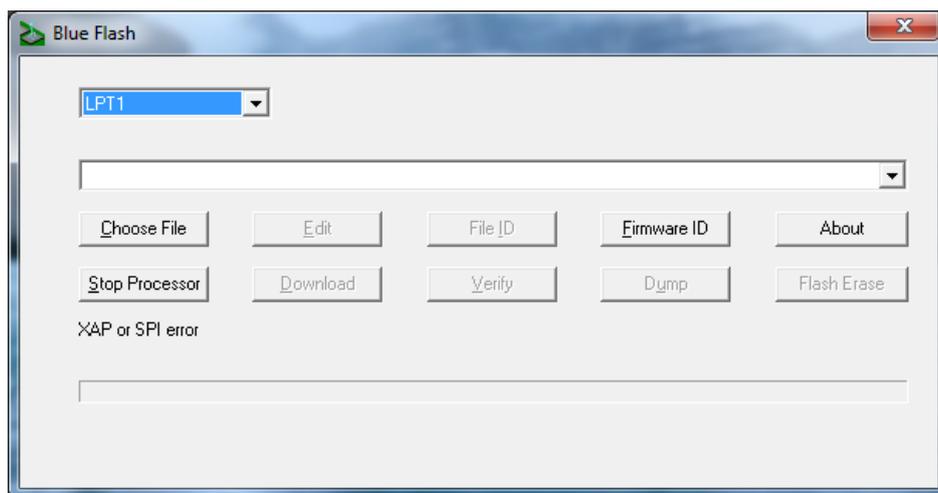


Figure 6: BlueFlash software

1. Download and install BlueSuite software package for example from Bluegiga iWRAP web page: <https://www.silabs.com/products/development-tools/software/bluegiga-iwrap-bluetooth-software-stack/>.
2. Connect your WT1x or WT3x module via SPI interface to a PC
3. Start BlueFlash application
4. Press **Stop processor**
5. Choose the .XPV file
6. Press **Download**
7. Once the download is complete, press **Start processor**
8. You are done

**Never install the .XPV and .XDV files delivered with iWRAP update client directly using BlueFlash software, since it will corrupt the unique settings of your module.**

## 3 Firmware update over UART

Device Firmware Upgrade (DFU) protocol allows you to update your *Bluetooth* module over either UART or USB interface. This may allow more flexible firmware updates than SPI, especially when the *Bluetooth* module is integrated onto an application board.

Unlike the SPI, the DFU is an open protocol and can therefore be implemented into microcontrollers and processors. This is might be useful in cases where a cabled connection from a PC to the *Bluetooth* module is not possible, but on the other hand the host processor could be used to perform the firmware update.

The following chapters describe how firmware update can be made over RS232 and using a Windows application called SerialDFU.

### 3.1 Prerequisites

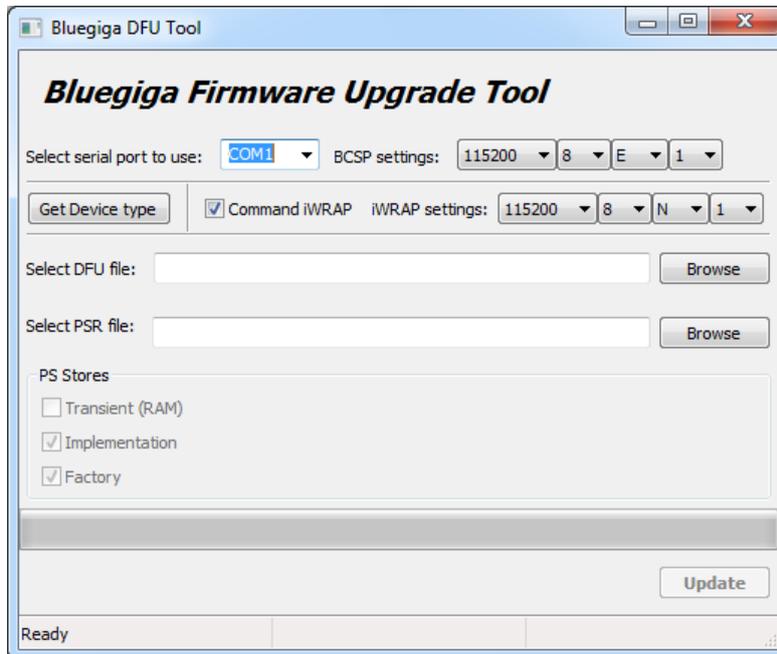
1. You need to have a physical access to the UART interface of your *Bluetooth* module.
2. You need to have a correct DFU file (.dfu) for your firmware and hardware version.
3. In case of performing firmware update over RS-232 interface and no hardware flow control (RTS and CTS) is used in the *Bluetooth* module, the flow control pins must be connected together in the RS-232 cable.

### 3.2 SerialDFU

SerialDFU is simple Windows based software that enables you to update the firmware of your *Bluetooth* modules over RS232. SerialDFU application is distributed together with iWRAP firmware releases.

### 3.3 Firmware update with SerialDFU

1. Connect your Bluetooth module via a RS232 to a PC.
2. Start the SerialDFU software.



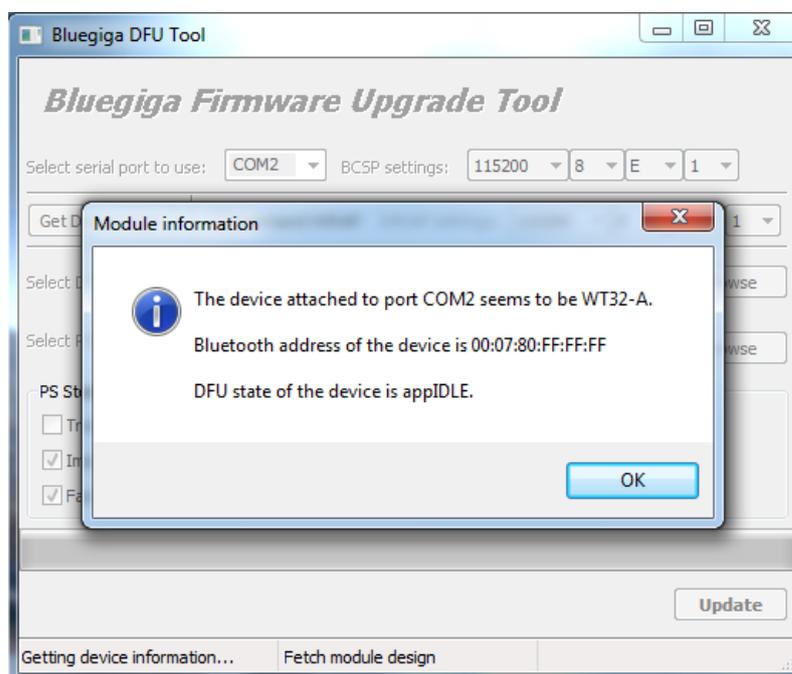
**Figure 7: SerialDFU software**

3. Select the correct COM port.
4. Select the correct baud rate for BCSP protocol used for the firmware update.

The default the BCSP baud rate is 115200 and if you have not changed it via PS keys, you should not touch this setting.

5. SerialDFU software can automatically put iWRAP firmware into firmware update mode. If you have not done this manually, please check **Command iWRAP** box.
6. Select the correct baud rate used with iWRAP firmware. Notice that this baud rate may be different from the BCSP baud rate.

7. If you do not know the hardware version of your *Bluetooth* module, use the **Get Device Type** button.

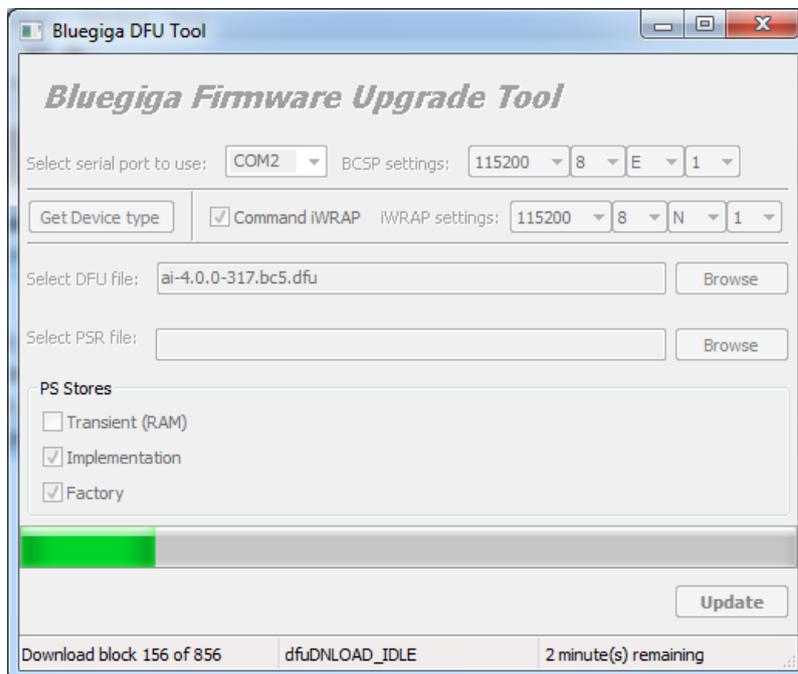


**Figure 8: Get Device Type**

8. Select the correct DFU file for your hardware.
9. SerialDFU is also able to write PS keys into the module (i.e. custom module/iWRAP configuration settings). You can optionally also select a PSR file to be installed into the module. Note that if you provide a PSR file, then the three checkboxes below will be enabled for you to check or uncheck. They concern the PS key settings supplied in the PSR file:
  - **Transient (RAM):** Write key values to RAM only. Settings will revert back to defaults on a cold reset.
  - **Implementation:** Write key values to flash in implementation (non-factory) slots. Settings will remain through warm or cold resets, but will revert back to factory defaults if you issue the “SET RESET” command in iWRAP.
  - **Factory:** Write key values to factory default slots. These are the settings that will be used if you use a “SET RESET”.

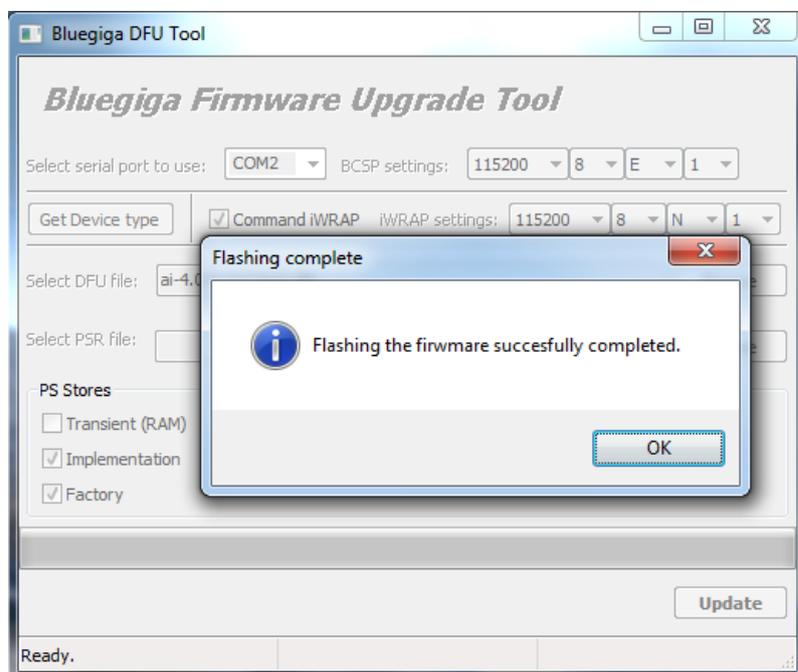
Note that iWRAP will attempt to load settings in a downward hierarchy upon booting. **Implementation** settings are loaded on boot where they exist. If a key does not have an implementation setting value, it will load them from **factory** settings. If it doesn't have a factory setting value, it will load the ROM default value (which cannot be modified).

10. Press **Update**.



**Figure 9: DFU update in progress**

11. Wait until the update finishes.



**Figure 10: Update complete**

12. Finally, reset the module.

### 3.4 DFUWizard

DFUWizard is a firmware update tool from CSR. It can be used to update the firmware of your *Bluetooth* module in a similar fashion as with SerialDFU:

The only difference is that DFUWizard also works over USB interface.

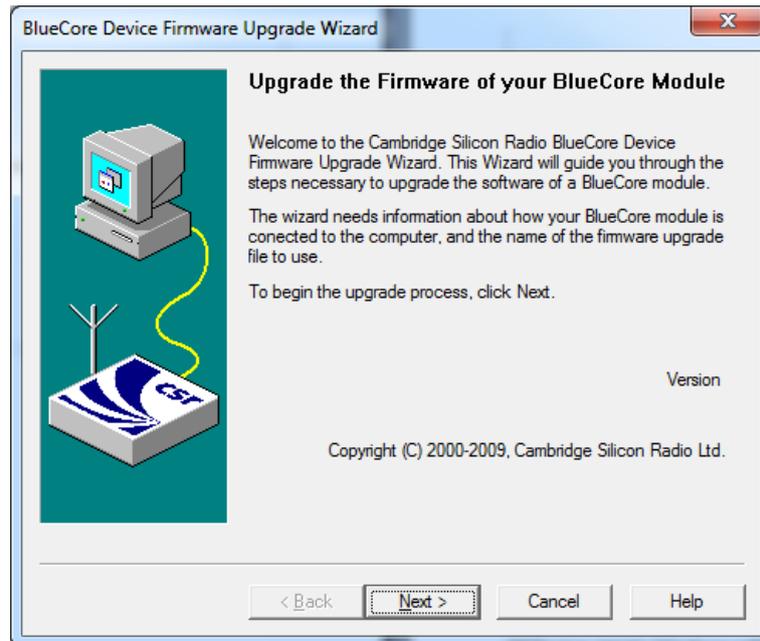


Figure 11: DFUWizard

### 3.5 DFU protocol

In case you want to implement firmware updates over DFU into your system, please contact [www.silabs.com/support](http://www.silabs.com/support) for the DFU protocol description.

The Python based source code of SerialDFU is also available upon request.

## 4 Troubleshooting

### 4.1 iWRAP update client is unable to identify or connect to the module

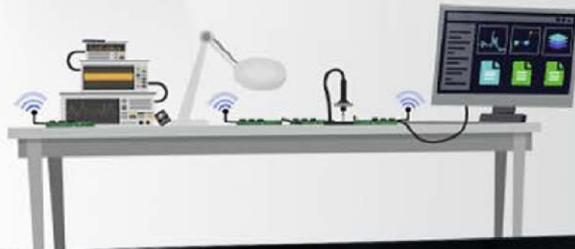
1. Verify your SPI connection and make sure the module is powered up.
2. From your PC's BIOS, please check that the LPT port is enabled and in ECP mode and try iWRAP update client again.
3. If that does not help, please reinstall the SPI device drivers by running **\\CSR\BlueSuite 2.5\drivers\win32\InstParSpi.bat** from the BlueSuite installation folder (typically C:\Program Files\CSR\BlueSuite 2.5). Restart the PC and try iWRAP update client again.
4. If you have tried all of the above, try to install iWRAP update client to a different PC. **Note that 64-bit operating systems do not work correctly with the parallel port programmer, regardless of which main version of Windows it is (Vista, 7, or 8).**
5. Contact [www.silabs.com/support](http://www.silabs.com/support).

### 4.2 SerialDFU fails to start the firmware update

1. Verify your UART connection and make sure the module is powered up.
2. Verify that the iWRAP baud rate is configured correctly.
3. Verify that the BCSP baud rate is configured correctly.
4. You can also manually put iWRAP into DFU update mode by issuing an iWRAP command "**BOOT 1**".

Silicon Labs

# Simplicity Studio™ 4



## Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio

[www.silabs.com/loT](http://www.silabs.com/loT)



SW/HW

[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



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 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 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®, Gecko OS, Gecko OS Studio, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, 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. Wi-Fi is a registered trademark of the Wi-Fi Alliance. 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>