

# AN1157: Developing Products Using *Bluetooth*<sup>®</sup> Xpress



Bluetooth Xpress streamlines every step of *Bluetooth*<sup>®</sup> Low Energy enabled system design from board to cloud.

Bluetooth Xpress provides a number of key advantages to speed application development, including the following:

- Pre-programmed serial cable replacement functionality
- Module certification
- Mobile app development framework
- Cloud connectivity for over-the-air updates

This document discusses each of these advantages in detail, as well as other considerations to be made when developing with the Bluetooth Xpress modules.

For a complete specification of commands and functionality discussed in this document, please see [docs.silabs.com/bgx](https://docs.silabs.com/bgx).

## KEY FEATURES

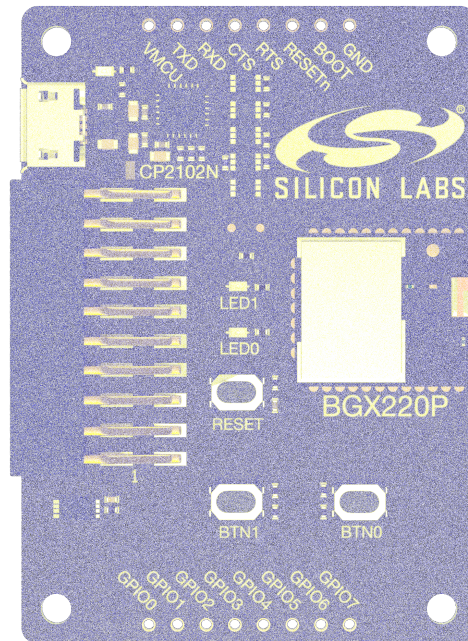
- Evaluate with Bluetooth Xpress expansion board
- Tweak settings with Xpress Configurator
- Develop apps with Bluetooth Xpress frame-work
- Easy certification with Bluetooth SIG

# Table of Contents

<b>1. Hardware and Software Setup</b>	<b>3</b>
<b>2. Configuring the Bluetooth Xpress Module</b>	<b>4</b>
2.1 Xpress Configurator	5
2.2 Terminal Interface	6
<b>3. Initial Evaluation with the BGX Expansion Board</b>	<b>7</b>
3.1 GPIO Pins on BGX Evaluation Board	7
3.2 Connecting BGX Expansion Board to Embedded Hosts	8
3.3 Expansion Header Pin Assignment	8
<b>4. Bluetooth Xpress Use Cases</b>	<b>9</b>
4.1 Security	10
4.2 Optimization for Low power/High Throughput.	11
<b>5. Building a Prototype</b>	<b>13</b>
5.1 Programming on Factory Floor vs. Ordering Custom OPNs through Xpress Configurator	13
<b>6. Bluetooth SIG Product Qualification.</b>	<b>14</b>
<b>7. Revision History</b>	<b>15</b>

## 1. Hardware and Software Setup

The Bluetooth Xpress expansion board is a great way to get started with Bluetooth Xpress development. The latest Bluetooth Xpress expansion board, which uses a Bluetooth Xpress BGX220P module, provides different options for connecting to other systems and is compatible with Simplicity Studio.



**Figure 1.1. BGX220P Expansion Board**

Simplicity Studio, which is available for download at <https://www.silabs.com/products/development-tools/software/simplicity-studio>, includes the Xpress Configurator utility. Xpress Configurator offers a number of features that make evaluating connection, configuration, and communication features of Bluetooth Xpress easy.

## 2. Configuring the Bluetooth Xpress Module

All configuration with the Bluetooth Xpress module occurs over the module's serial interface. The serial interface for the BGX operates in two modes: stream mode and command mode. Stream mode is used to transmit bytes when the BGX is connected to another BGX or to a mobile device. Xpress command API mode is mostly used when a BGX's BLE link is not active. When in Xpress command API mode, commands can be issued to the BGX to configure variables that control performance and operational features of the BGX.

The Bluetooth Xpress expansion board includes a UART to USB bridge device, which enables developers to easily connect to the BGX with a PC. This section discusses connecting to the serial link with either Simplicity Studio's Xpress Configurator or a simple terminal program.

## 2.1 Xpress Configurator

Simplicity Studio supports device configuration through **[Xpress Configurator]**. This tool configures and communicates with the BGX device using the USB connector. For BGX devices, users can configure:

- GPIOs
- UART baud rate & flow control
- Modes of operation
- System configurations
- Advertising & connection configurations

Please note that the use of Simplicity Studio's Xpress Configurator is optional, and is intended to offer developers a more context-rich user interface than is available through a terminal program.

Xpress Configurator automatically validates all configurable parameters to ensure no conflicts between settings. When settings have been successfully validated, the user can download the configuration to a device by clicking **[Program to Device]**. Custom devices can also be ordered through Xpress Configurator by clicking **[Order Parts]**.

Xpress Configurator receives any messages sent from the BGX Commander application over the serial connection when BGX13P is connected to Xpress configurator. These can be seen in the terminal of Xpress Configurator which can also be used to send commands or stream data.

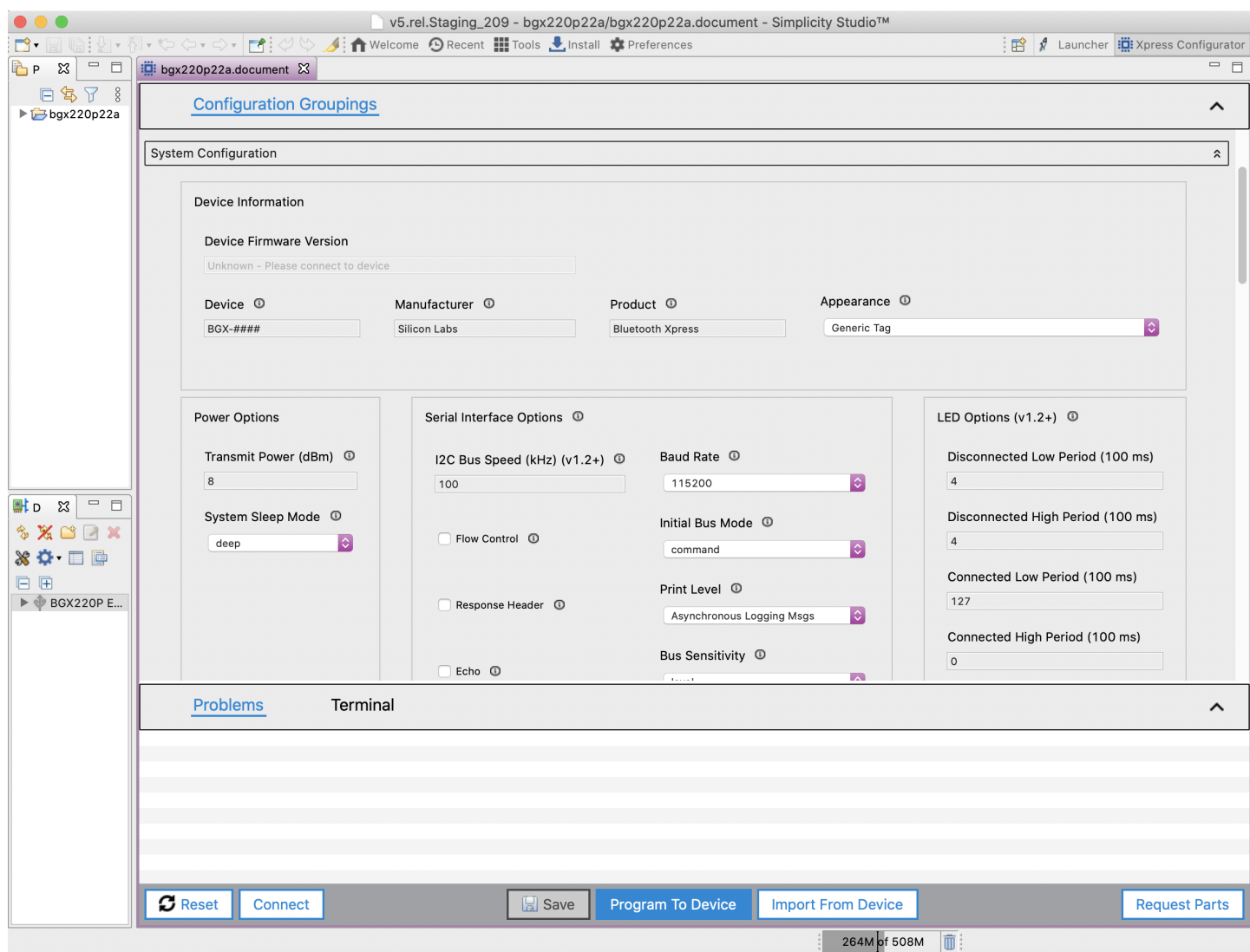


Figure 2.1. Xpress Configurator

The tool also includes datasheet-derived information about each configurable parameter. This information can be accessed by clicking on the small [i] buttons next to each parameter. The datasheet text will then appear in the documentation pane toward the bottom of the screen.

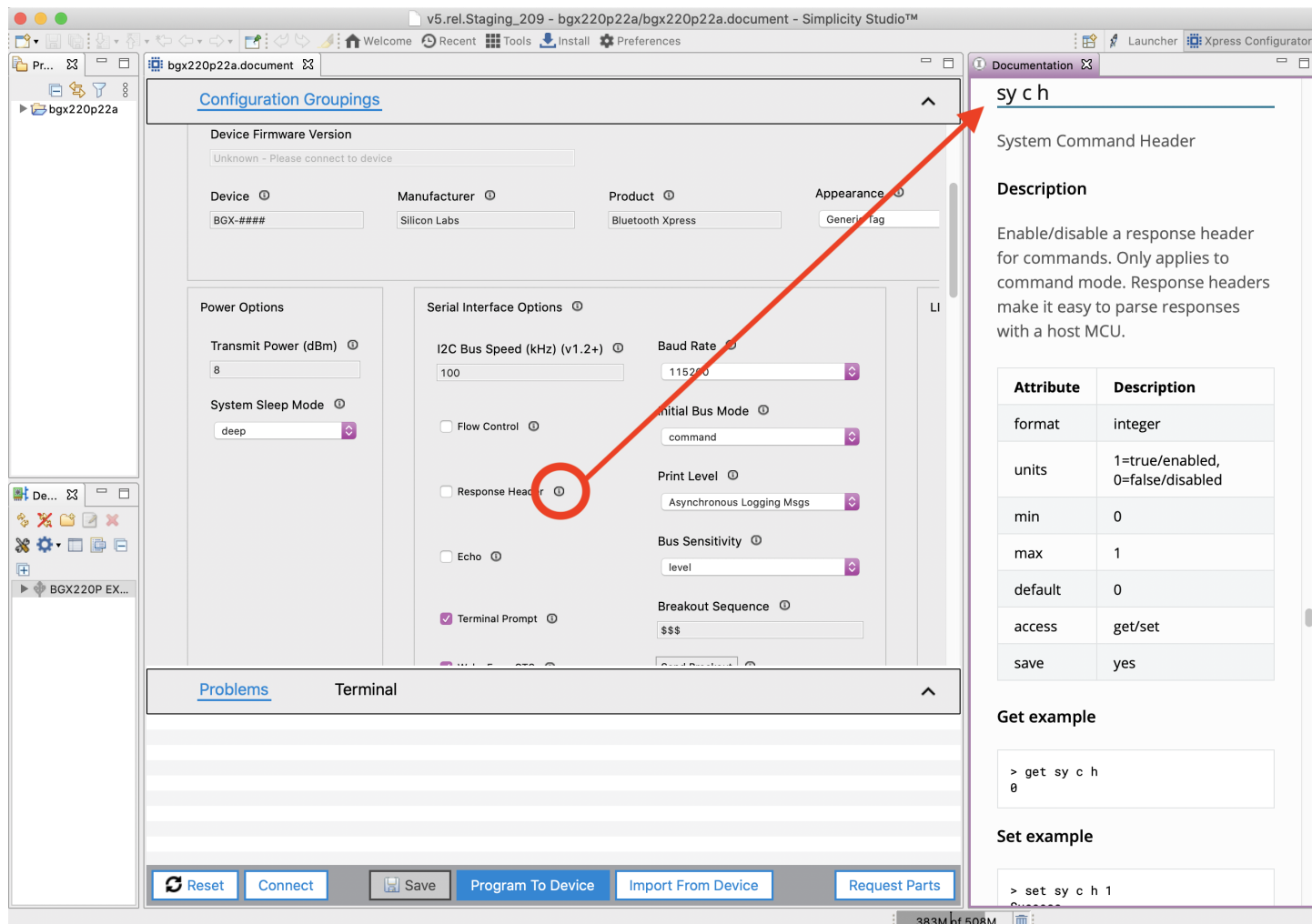


Figure 2.2. Xpress Configurator Parameter Documentation

## 2.2 Terminal Interface

While Xpress Configurator provides a full-featured, graphical interface for configuration and communication, users can also simply open the terminal program of their choice and manually test commands through the serial interface. A quick test can be done to check the terminal interface connection. Connect BGX to a PC through the USB connection to terminal application. BGX will appear as a COM port and can be opened in a terminal application which allows the user to communicate with the on-board Bluetooth Xpress module. For instance, issuing the `ver` command should give a version number as a response.

**Note:** By default, the UART baud rate of BGX is 115200 bps. The terminal application in use should be configured to use that baud rate.

### 3. Initial Evaluation with the BGX Expansion Board

The Bluetooth Xpress module offers 8 GPIOs, which can be used as a standard I/O and can be configured to special functions to be used within embedded systems. The section below discusses configuration and usage of the module's GPIOs.

#### 3.1 GPIO Pins on BGX Evaluation Board

The Bluetooth Xpress expansion board provides access to all 8 GPIOs. GPIOs are configurable for input and output which are usable as inputs, open-drain outputs, or push-pull outputs. The GPIO function and direction are set by using `gfu` and `gdi/gdis` commands. `gge/gges` and `gse/gses` commands are also used for configuring GPIOs.

The evaluation board routes two GPIOs to mechanical buttons and two GPIOs to LEDs to enable testing of different features. GPIO0/1 are connected to LED0/1 and GPIO 2/3 are connected to BTN0/1. These mechanical buttons and LEDs enable developers to quickly test out many I/O related features of the BGX, including the following:

- Controlling the operational state of the BGX by configuring the GPIO(BTN0/1) pin as input:
  - For GPIO 2's BTN0: `gfu 2 str_select`
  - For GPIO 3's BTN1: `gfu 3 str_select`
- Displaying BLE connection status by configuring the GPIO(LED0/1) pin as output:
  - For GPIO 0's LED0: `gfu 0 con_status_led`
  - For GPIO 1's LED1: `gfu 1 con_status_led`
- Indicating BLE Activity by configuring the GPIO(LED0/1) pin as output with:
  - For GPIO 0's LED0: `gfu 0 str_activity_led`
  - For GPIO 1's LED1: `gfu 1 str_activity_led`
- Standard IO control by a connected mobile app or second BGX as shown in the table below.

**Note:** These examples discuss the underlying commands used to configure pins to these states. Xpress Configurator provides a graphical interface for pin assignment, and selected settings in that interface generate Xpress command API instructions equivalent to the commands discussed here.

**Table 3.1. Example GPIO Pin Configurations**

To configure GPIO2(Button0) pin as input:	To configure GPIO0(LED0) pin as output
<pre>&gt; gfu 2 stdio Success &gt;gdi 2 in Success</pre> <p><b>Test:</b> Before pressing the button0(get status)</p> <pre>&gt; gges XX1XXXXX</pre> <p>While pressing the button0(get status)</p> <pre>&gt; gges XX0XXXXX</pre>	<pre>&gt;gfu 0 stdio Success &gt;gdi 0 ohi Success</pre> <p><b>Test:</b> To get the status of LED0</p> <pre>&gt; gges 1XXXXXX</pre> <p>To set the LED0</p> <pre>&gt; gges 0XXXXXXX</pre>

### 3.2 Connecting BGX Expansion Board to Embedded Hosts

The expansion board can be connected to Silicon Labs EFM8 and EFM32 starter kits through the expansion header. The board can also be connected to any host MCU using either the expansion header signals or the pin access provided by vias on the board.

An example project with an EFM8 MCU simulating an embedded host is available in the Simplicity Studio for EFM8SB1 and EFM8UB1 devices. This example demonstrates bi-directional communication between a mobile app and a Bluetooth Xpress module in Peripheral and Central mode. It can also be used to demonstrate BGX-to-BGX communication, if central is selected instead of peripheral in the startup screen. The source code for this example illustrates best practices for BGX communication and shows how easy the host-to-BGX link is to implement.

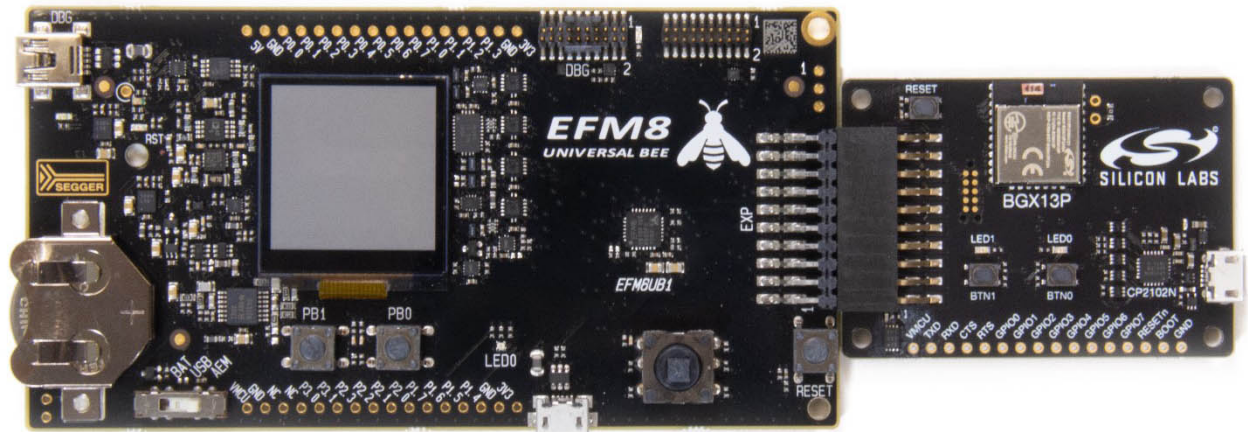


Figure 3.1. BGX Expansion Board Connected to EFM8

### 3.3 Expansion Header Pin Assignment

Bluetooth Xpress evaluation boards include a 20-pin expansion header that enables connection to Silicon Labs EFM8 and EFM32 starter kits. The pins in the header used by the BGX are listed in the table below.

Table 3.2. Pin Assignment of EXP Header

EXP header Pin number	Functionality
8	BTN0/GPIO2
10	BTN1/GPIO3
12	RXD
14	TXD
11	RTS
13	CTS



## 4. Bluetooth Xpress Use Cases

Bluetooth Xpress modules function in two use cases:

- *BGX to smart phone* – In this use case, the BGX acts as a peripheral, which is discoverable and connectable by a smart phone acting as the central.
- *BGX to BGX13* – In this use case, one BGX acts as a discoverable, connectable peripheral, and a second BGX acts as the central, which scans and connects.

When connected, a BGX can function in either a streaming mode, where data bytes are transmitted between the two points in a no overhead, cable-replacement style. A connected BGX can switch to Xpress command API mode with a configurable breakout sequence or through a GPIO pin configured as `str_select` input. When the embedded MCU mode-switches the BGX, commands can be executed. A connected BGX can also be placed into remote command mode, where the remote point in communication can execute commands on the local BGX.

In both defined use cases, flow control signals from the BGX indicate buffer status and can be used to indicate whether the embedded host can transmit data. Optionally, configurable outputs from the BGX, such as `con_active` and `str_active`, indicate whether a connection is active and whether streaming mode is currently active, respectively.

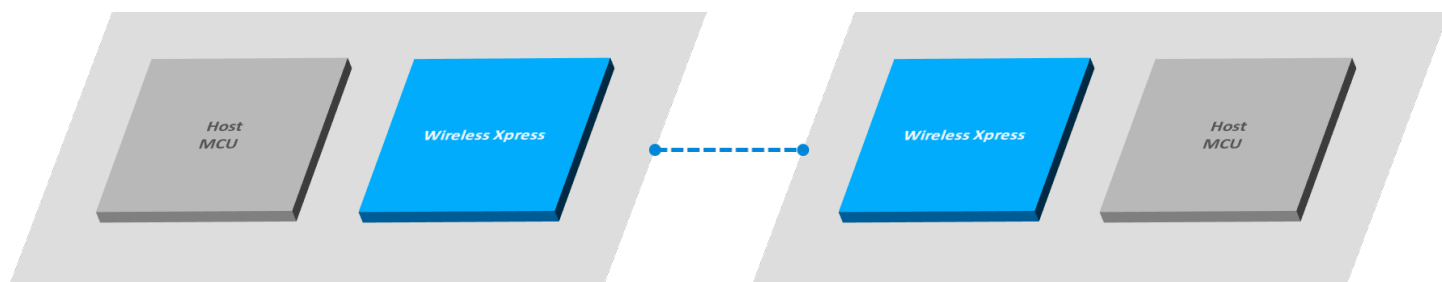


Figure 4.1. BGX to BGX Use Case

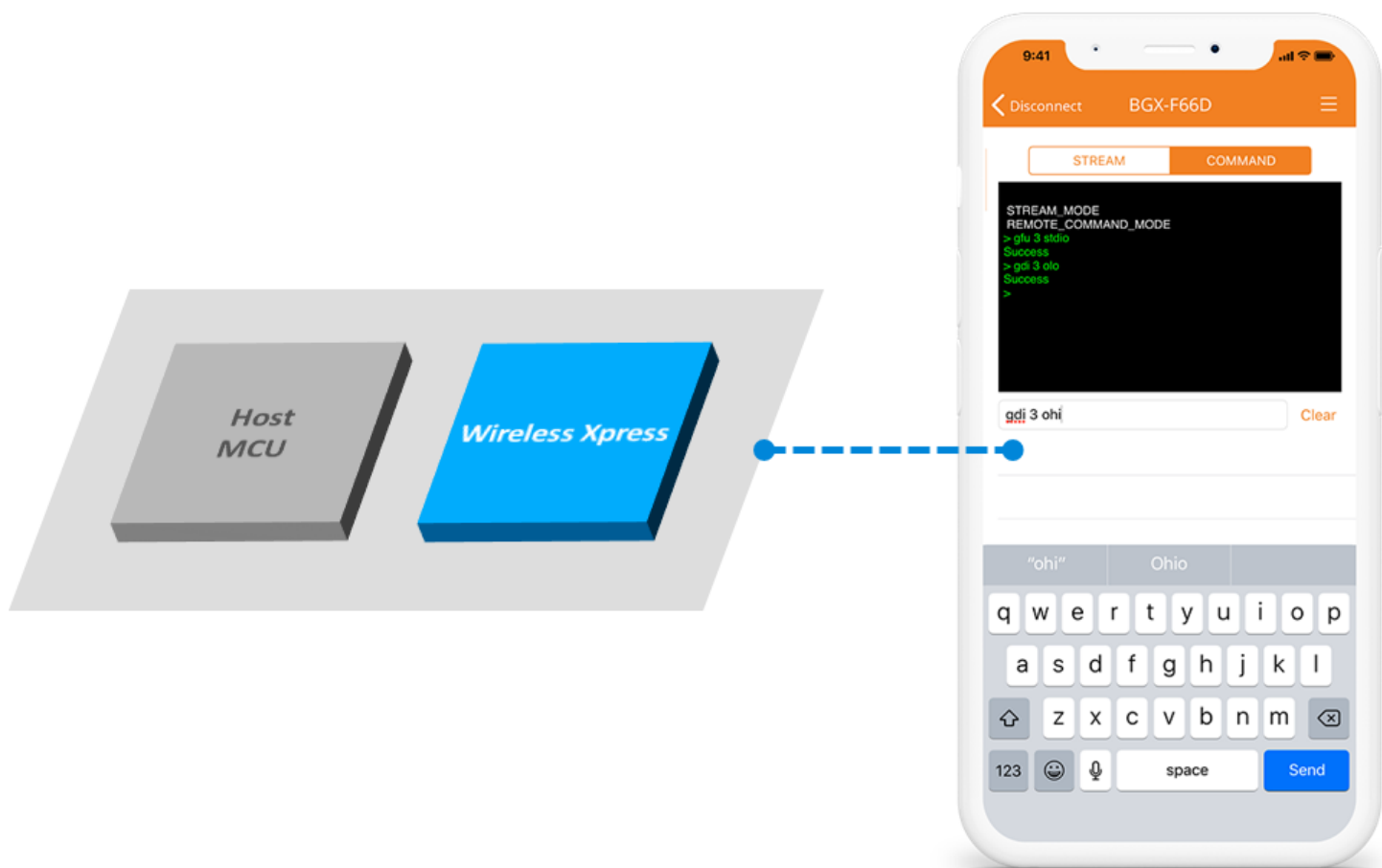


Figure 4.2. BGX to Phone Use Case

## 4.1 Security

BGX supports Security Mode 1 (encryption) with various levels (Levels 2 / 3/ 4). Encryption is provided by using one of the two possible key types, keyless, 6 digit pin code (passkey) which can be selected by setting the `blek` variable. BGX BLE encryption and pairing are managed with three encryption variables: `bleb` - Encryption bondable mode, `blek` - Encryption key, `blep` - Encryption pairing mode.

**Note:** Enabling bonding can result in a 'BOND FAILED' connection failure if either of the two points in the Bluetooth link clear bonding information. On the BGX, bonding information can be cleared using the `clrb` command.

By default, the BGX supports the ability to allow for remote command execution. This feature can be exercised by a mobile app by calling an API in the BG Xpress framework. Once in this mode, the mobile app has the ability to execute Xpress command API commands on the BGX device, including `get/set` commands that can alter configurable parameters in the BGX. Remote command mode access can be restricted using the `syre` command.

## 4.2 Optimization for Low power/High Throughput

BGX can be optimized for power-sensitive or high-throughput applications by customizing variables. Variables that effect the balance between throughput/responsiveness and low current draw are:

- Transmit power (*bl t c*) – higher transmit power means longer range but higher current draw in advertising and during connection
- Advertising intervals and durations (*bl v h i*, *bl v h d*, *bl v l i*, *bl v l d*) – the higher the interval and longer the duration, the higher the likelihood of being discovered by a scanning central, and also the higher the current draw
- Connection interval (*bl c i*) – a larger interval increases throughput but also increases current draw

The current draw is also determined by the UART baud rate. Figure 4.3, 4.4, and 4.5 are the Energy Profiler captures of the BGX which is acting as peripheral in active mode. Figures 4.3 and 4.4 show the current drawn for different configurations. First, BGX is configured not to advertise, later it is configured to advertise in low mode, and then it is connected to a BGX/mobile (stream mode).

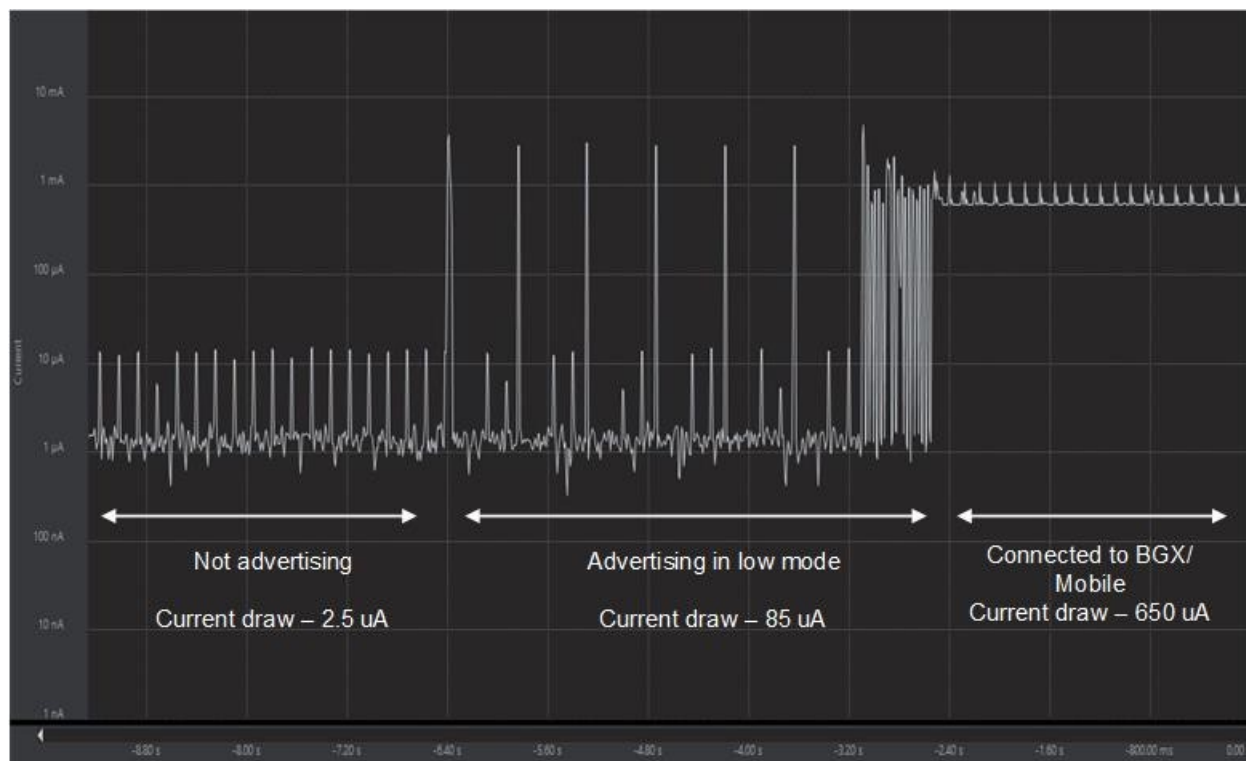
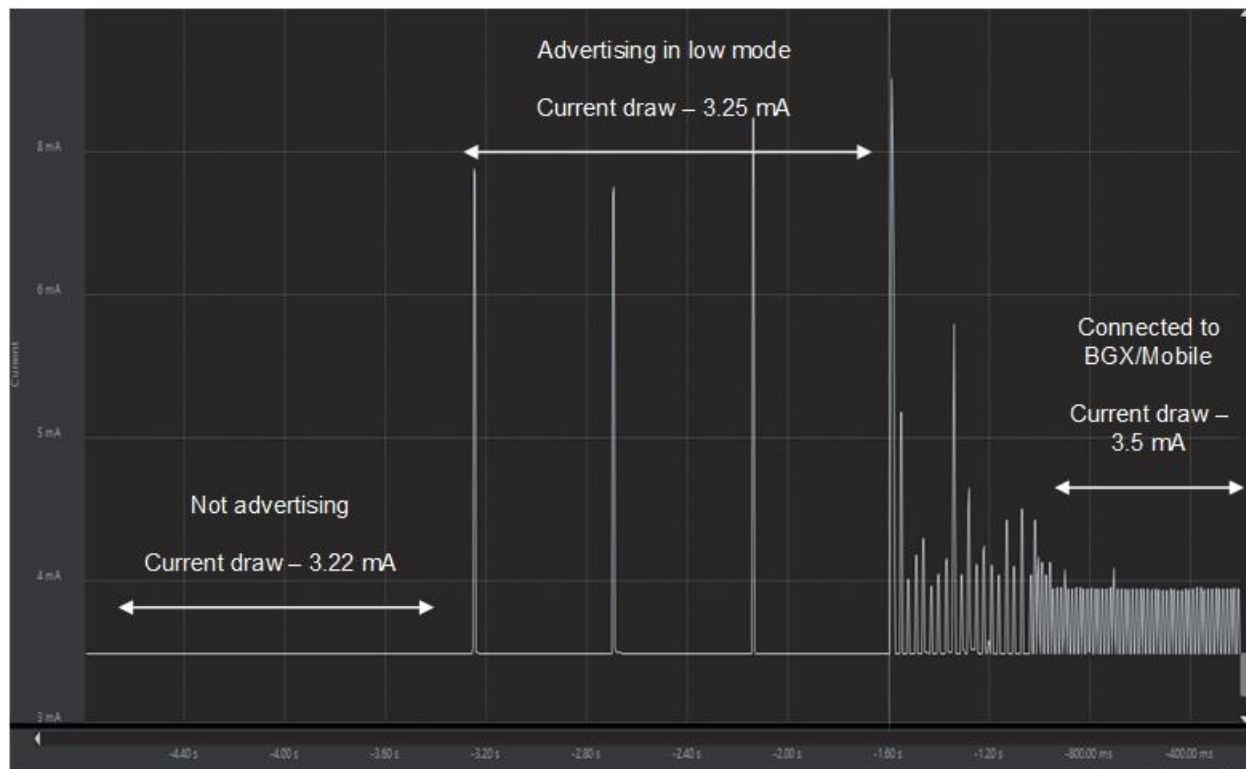
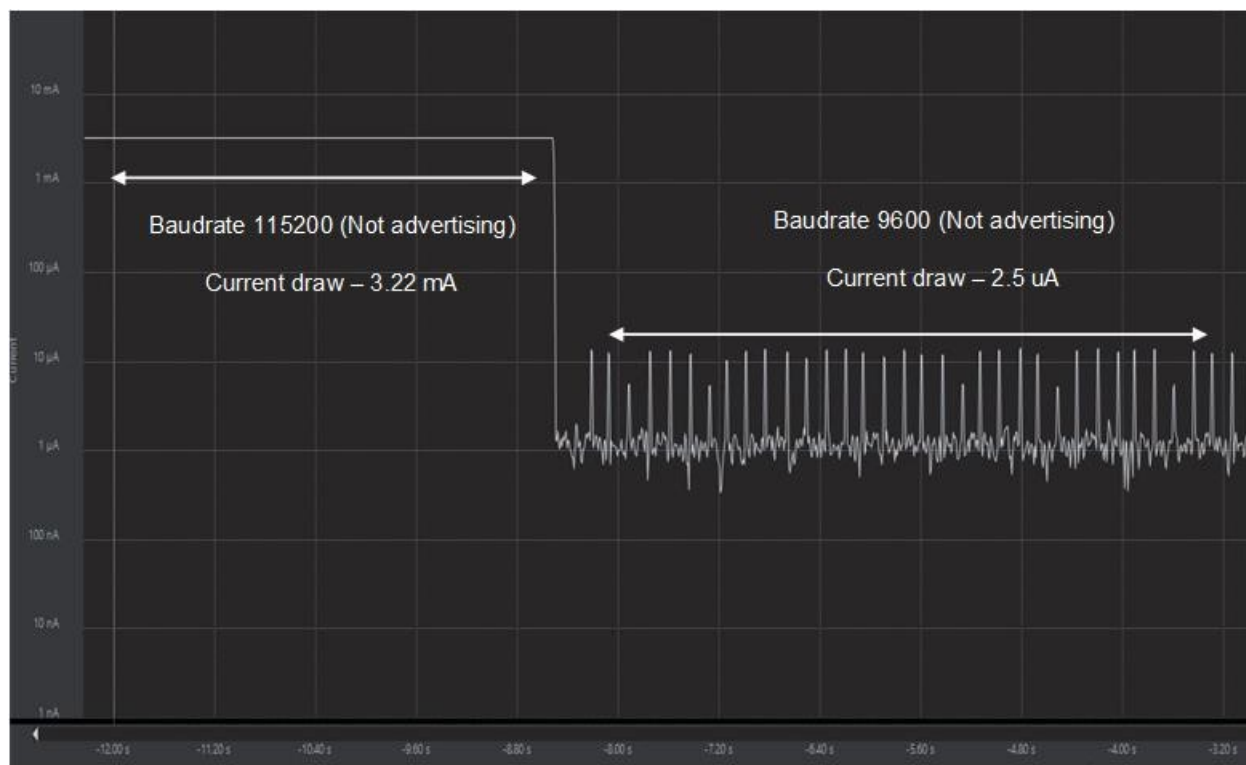


Figure 4.3. Energy Profiler Capture of BGX13 with 9600 bps UART Baud Rate



**Figure 4.4. Energy Profiler Capture of BGX13 with 115200 bps UART Baud Rate**

Figure 4.5 shows the current drop of BGX(idle) from 3.22 mA to 2.5 uA when the baud rate is changed from 115200 bps to 9600 bps.



**Figure 4.5. Energy Profiler Capture of BGX13 when Reducing Baud Rate from 115200 bps to 9600 bps**

## 5. Building a Prototype

BGX evaluation boards can be used to develop a first-stage prototype by flying wires from the expansion header or available vias to a breadboard or prototype system. This will enable the developer to establish a UART connection from the host processor to the BGX.

The host MCU can configure variables in the BGX through the serial interface when in command mode. One way to enable the host to write to these variables is to use the *bgx\_command\_list.txt* text file generated by Xpress Configurator, which lists all variable settings. This file is stored in Xpress Configurator's project explorer window, as shown in the figure below. This text file can be imported into a project as an array of strings to be written out sequentially to the BGX.

**Note:** Clicking on 'Program to device' in Xpress Configurator will perform this same task, setting each variable sequentially. This process is followed by an execution of the 'save' command, which stores variables in non-volatile memory as defaults in the BGX. Using this method of programming through Xpress Configurator, it is not necessary to import the variables text file into the host's project for configuration.

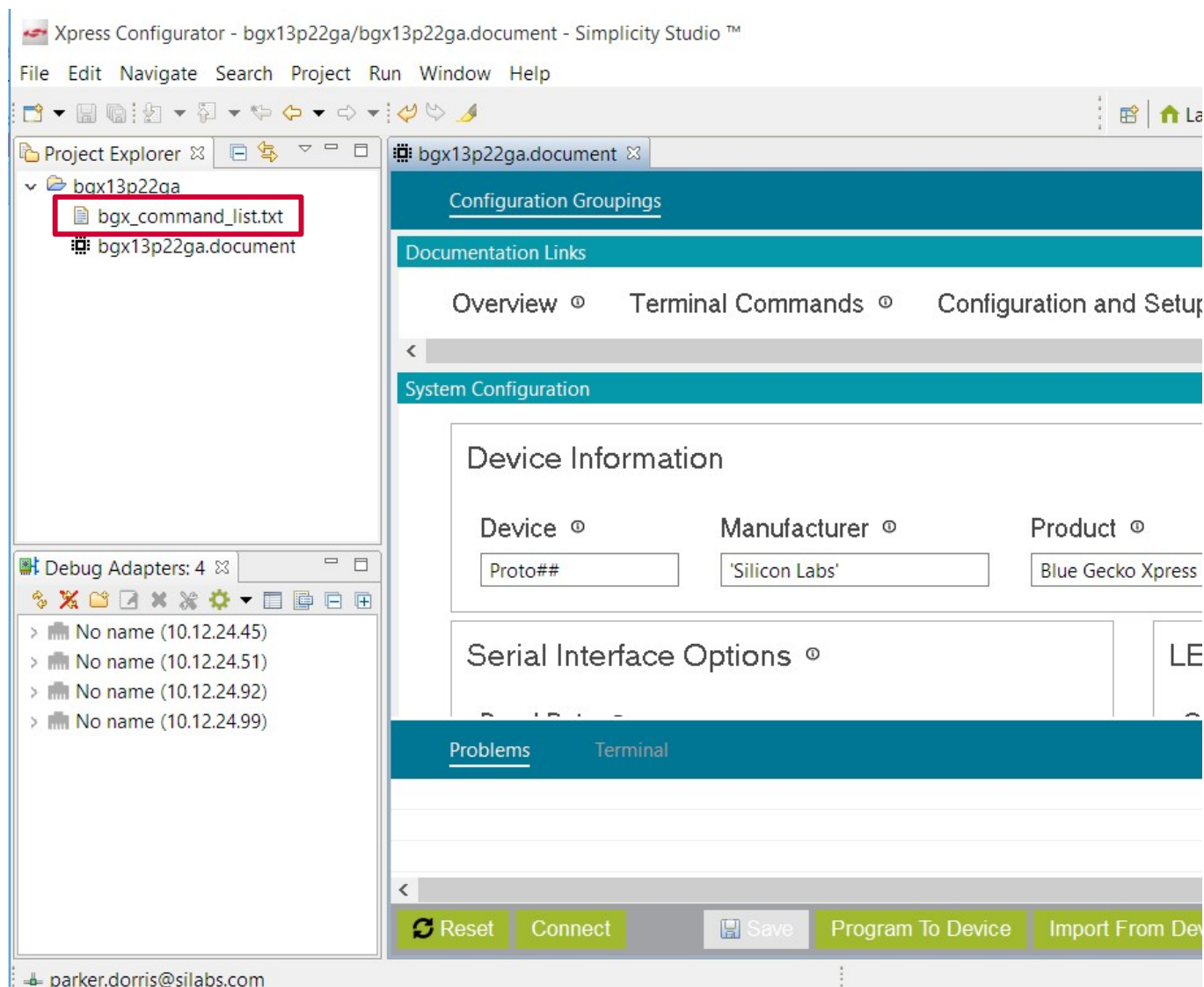


Figure 5.1. Xpress Configurator Generated *bgx\_command\_list.txt* File

### 5.1 Programming on Factory Floor vs. Ordering Custom OPNs through Xpress Configurator

A request to Silicon Labs for a custom OPN can be made through the Xpress Configurator tool by clicking on the 'Request Parts' button on the bottom right portion of the tool's perspective. This process will output a file that can be submitted directly to factory for first article generation and production orders if validated.

## 6. Bluetooth SIG Product Qualification

Before you can sell a Bluetooth-enabled product to the market, that product must be qualified as an end-product through Bluetooth SIG. The process is managed by the SIG's Launch Studio tool which is available at <https://launchstudio.bluetooth.com/>.

For customers using Bluetooth Xpress, the process is simplified because the underlying Bluetooth components have already been qualified. With these pre-qualified components, the starting point at the Launch Studio is the option *Start the Bluetooth Qualification Process with Required Testing*. Despite the name of the option, no testing is required since the pre-qualified BGX components have been already tested. Notice that one of the Launch Studio steps requests that the submitter use a Declaration ID in order to finalize the qualification process, and if the submitter does not have a Declaration ID, the tool will give the submitter the option to purchase one.

A submitter beginning the qualification process in Launch Studio will need to take the following into account:

- Components to import while qualifying the end-product, which can be determined using the table below
- Payment for purchase of a declaration ID for your product
- For official documentation on the qualification process, please refer to documentation on the Bluetooth SIG site at <https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/>

The qualification process in the Bluetooth SIG's LaunchStudio requires that the user know the Bluetooth Xpress product being qualified, and the Bluetooth Xpress firmware version running on that product. Bluetooth Xpress orderable part numbers are pre-programmed with the BGX firmware versions. Each BGX firmware version uses a corresponding version of the Silicon Labs BLE SDK.

Once the user has determined the BLE SDK version in use, please refer to [QSG139 \(Section 2.2\)](#) to determine the appropriate QDIDs to use during qualification.

BGX firmware version	Products supported	Pre-programmed option	BLE SDK version used
1.0.880.2	BGX13P/S	-	2.8.1
1.0.927.2	BGX13P/S	BGX13S22GA-V21 BGX13P22GA-V21	2.8.1
1.1.1524.0	BGX13P/S	-	2.10.1
1.2.2045.0	BGX13P/S	BGX13P22GA-V31 BGX13S22GA-V31	2.12.3
1.2.2376.0	BGX13P/S	-	2.12.3
1.3.2482.0	BGX220P/S	BGX220P22HNA21	2.13.6.0

## 7. Revision History

### Revision 0.2

September 2020

- Changed references to BGX13 to BGX or similar family terminology to cover both the BGX13 and BGX220 product families
- Updated Xpress Configurator images to show Simplicity Studio 5
- Added additional details to the Bluetooth SIG Product Qualification section

# 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 & Community**

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

## 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>