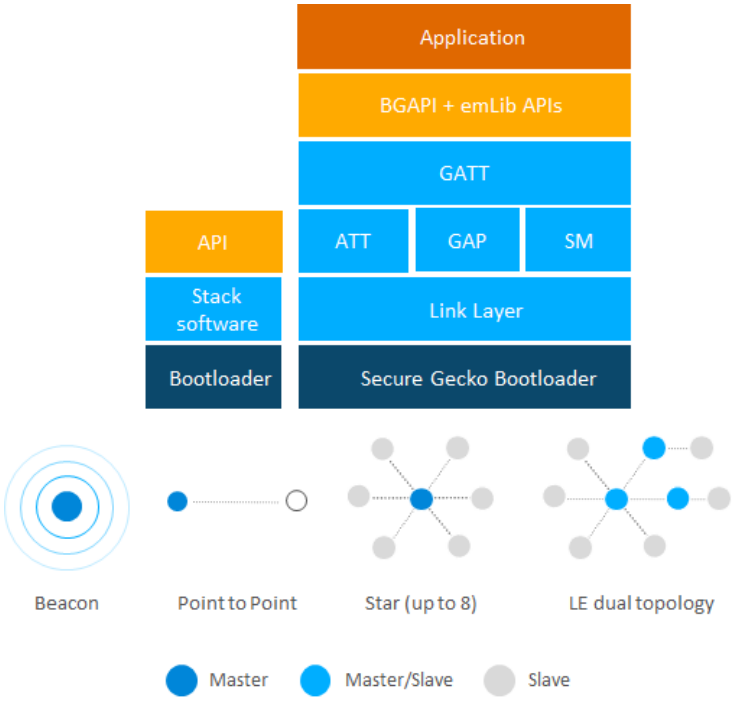
QSG108: Getting Started with Silicon Labs *Bluetooth*® Software

This document walks you through the Silicon Labs Bluetooth stack, SDK (Software Development Kit), and development tools for Wireless Geckos and also instructs you briefly how to get started with your own software development.

This document is written for the Silicon Labs Bluetooth SDK v.2.7.0 and higher.

KEY POINTS

- Introduction to the Silicon Labs Bluetooth Stack
- Description of the Bluetooth APIs
- Brief introduction to the Bluetooth SDK
- Description of the development tools
- Instructions for getting started with software development

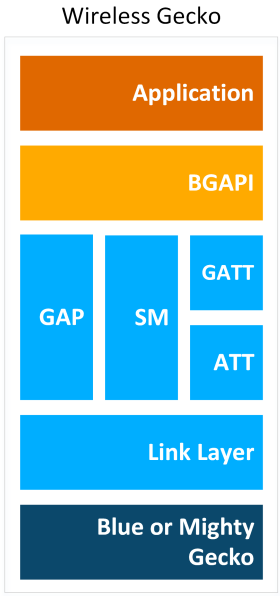


1. The Silicon Labs Bluetooth Stack

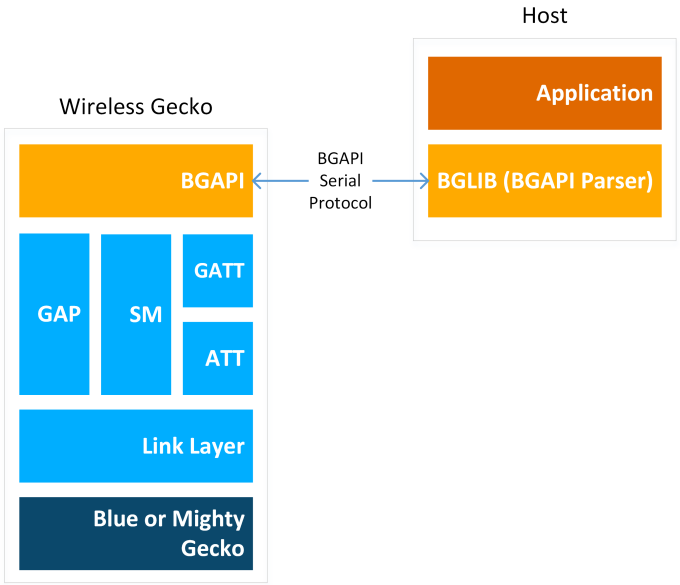
The Silicon Labs Bluetooth stack is an advanced Bluetooth 5-compliant protocol stack implementing the Bluetooth low energy standard. It supports multiple connections, concurrent central, peripheral, broadcaster, and observer roles. The Silicon Labs Bluetooth stack is meant for Silicon Labs Wireless Gecko SoCs and modules.

The Silicon Labs Bluetooth stack provides multiple APIs for the developer to access the Bluetooth functionality. Two modes are supported:

1. Standalone mode, where both the Bluetooth stack and the application run in a Wireless Geckos SoC or module. The application can be developed with C programming language.



2. Network Co-Processor (NCP) mode, where the Bluetooth stack runs in a Wireless Gecko and the application runs on a separate host MCU. For this use case, the Bluetooth stack can be configured into NCP mode where the API is exposed over a serial interface such as UART.



The features of the Silicon Labs Bluetooth stack are listed in the following table.

Table 1.1. Bluetooth Stack Features

Feature	Value and comments
Bluetooth version	Bluetooth 5
Bluetooth features	Bluetooth 5 2M PHY (EFR32xG12 and EFR32xG13) Bluetooth 5 LE Long Range (only EFR32xG13 products) Bluetooth 5 advertisement sets and scan event reporting Concurrent central, peripheral, broadcaster and observer modes LE secure connections LE Privacy 1.2 (peripheral) LE packet length extensions LE dual topology up to 8 master connections
Simulations connections	Up to 8 with simultaneous master and slave connections
Maximum throughput	1300 kbps over 2M PHY (Software 2.6.x and newer) 700 kbps over 1M PHY (Software 2.6.x and newer)
Encryption	AES-128
Pairing modes	Just works Man-in-the-Middle with numeric comparison and passkey
Number of simultaneous bondings	Up to 14
Link Layer packet size	Up to 128 B ((Software 2.3.x) Up to 160 B (Software 2.4.x) Up to 251 B (Software 2.6.x and newer)
ATT protocol packet size	Up to 58 B (Software 2.0.x) Up to 126 B (Software 2.1.x) Up to 250 B (Software 2.3.x and newer)
Supported Bluetooth profiles and services	All GATT based profiles and services are supported
Apple HomeKit	Apple HomeKit R9 compliant implementation Implements all Apple HomeKit profiles and services Available separately for Apple MFi licensees
Host (NCP) interfaces	4-wire UART with RTS/CTS control 2-wire UART GPIOs for sleep and wake-up management
Bootloaders	Secure Gecko Bootloader supporting authenticated and encrypted updates over OTA or UART and Secure Boot. The Gecko Bootloader also supports flash partitioning and both internal and external (SPI) flash.

Feature	Value and comments
Bluetooth QDIDs	Link layer QDID (Bluetooth 4.2): 81105 Host stack QDID (Bluetooth 4.2): 91422 Link layer QDID (Bluetooth 5): 99504 Host stack QDID (Bluetooth 5): 101550

1.1 The Bluetooth Stack APIs

This section briefly describes the different software APIs available for the developer.

1.1.1 The BGAPI Bluetooth API

The BGAPI is the Bluetooth API provided by the Silicon Labs Bluetooth stack. It provides access to all the Bluetooth functionality implemented by the Bluetooth stack, such as: the Generic Access Profile (GAP), connection manager, the security manager (SM), and GATT client and server.

In addition to the Bluetooth APIs, the BGAPI also provides access to a few other functions like the Direct Test Mode (DTM) API for RF testing purposes, the Persistent Store (PS) API for reading and writing keys to and from the devices flash memory, the DFU (Device Firmware Update) API for field firmware updates, and the System API for various system level functions.

1.1.2 CMSIS and emlib

The Cortex Microcontroller Software Interface Standard (CMSIS) is a common coding standard for all ARM Cortex devices. The CMSIS library provided by Silicon Labs contains header files, defines (for peripherals, registers and bitfields), and startup files for all devices. In addition, CMSIS includes functions that are common to all Cortex devices, like interrupt handling, intrinsic functions, etc. Although it is possible to write to registers using hard-coded address and data values, it is recommended to use the defines to ensure portability and readability of the code.

To simplify programming Wireless Geckos, Silicon Labs developed and maintains a complete C function library called **emlib** that provides efficient, clear, and robust access to and control of all peripherals and core functions in the device. This library resides within the em_xxx.c (e.g. em_dac.c) and em_xxx.h files in the SDK.

The emlib documentation is available on the Silicon Labs' website.

1.1.3 The BGAPI Serial Protocol and BGLIB Host API

When configured in NCP (network co-processor) mode, the Bluetooth stack also implements the BGAPI serial protocol. This allows the Bluetooth stack to be controlled over a serial interface such as UART from a separate host like an EFM32 microcontroller. The BGAPI serial protocol provides exactly the same Bluetooth APIs over UART as the BGAPI API when used in a standalone mode.

The BGAPI serial protocol is a lightweight, binary protocol that carries the BGAPI commands from the host to the Bluetooth stack and responses and events from the Bluetooth stack back to the host.

The Bluetooth SDK delivers a ready-made BGAPI serial protocol parser implementation, called BGLIB. It implements the serial protocol parser and C language function and events for all the APIs provided by the Bluetooth stack. The host code developed on top of BGLIB can be written to be identical to the code for the Wireless Gecko, which allows easy porting of the application code from the Wireless Gecko to a separate host or vice versa.

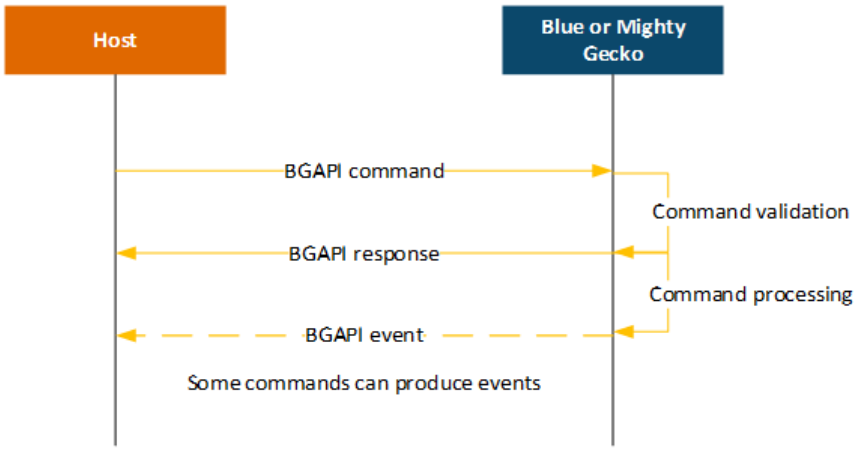


Figure 1.1. BGAPI Serial Protocol Message Exchange

The BGAPI serial protocol packet structure is described below.

Table 1.2. BGAPI Packet Structure

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4-255
Explanation	Message type	Minimum payload length	Message class	Message ID	Payload
Values	0x20: command 0x20: response 0xA0: event	0x00 - 0xFF	0x00 - 0xFF	0x00 - 0xFF	Specific to command, response, or event

1.1.4 The Bluetooth Profile Toolkit GATT Builder

The Bluetooth Profile Toolkit is a simple XML-based API and description language used to describe the GATT-based service and characteristic easily without the need to write them in code. The XML files can be easily written by hand or they can be auto-generated with the Bluetooth Developer Studio described later in this document.

The GATT database developed with the Profile Toolkit is converted to a .c file and a .h file and included in the application project as a pre-build step when the firmware is compiled. Then the GATT can be accessed with the Bluetooth stack GATT APIs or by a remote Bluetooth device.

```
<gatt>

    <service uuid="1800">

        <description>Generic Access Service</description>

        <characteristic uuid="2a00">
            <properties read="true" const="true" />
            <value>Blue Gecko BGM111</value>
        </characteristic>

        <characteristic uuid="2a01">
            <properties read="true" const="true" />
            <value type="hex">0768</value>
        </characteristic>

    </service>

</gatt>
```

Figure 1.2. Example of a Generic Access Service

2. The Bluetooth Software Development Kit (SDK)

The Bluetooth SDK is a full software development kit that enables you to develop applications on top of the Bluetooth stack using C programming language. The SDK also supports making standalone applications, where the Bluetooth stack and the application both run in the Wireless Gecko, or the network co-processor (NCP) architecture, where the application runs on an external host and the Bluetooth stack runs in the Wireless Gecko.

2.1 SDK Contents

This section describes the folder structure and contents of the SDK.

2.1.1 Libraries

The following libraries are delivered with the Bluetooth SDK and must be included in C application projects.

Library	Explanation	Mandatory
libbluetooth.a	Bluetooth stack library	Yes
librail_efr32xg1_gcc_release.a	RAIL library for GCC	Yes for GCC projects on EFR32xG1 platform
librail_efr32xg12_gcc_release.a	RAIL library for GCC	Yes for GCC projects on EFR32xG12 platform
librail_efr32xg13_gcc_release.a	RAIL library for GCC	Yes for GCC projects on EFR32xG13 platfor
librail_efr32xg1_iar_release.a	RAIL library for IAR	Yes for IAR projects on EFR32xG1 platform
librail_efr32xg12_iar_release.a	RAIL library for IAR	Yes for IAR projects on EFR32xG12 platform
librail_efr32xg13_iar_release.a	RAIL library for IAR	Yes for IAR projects on EFR32xG12 platform
libmbedtls.a	mbedtls library	Yes
libpsstore.a	PSStore library	Yes
binapploader.o	Apploader for OTA updates	No
libcoex.a	Wi-Fi and Bluetooth coexistence	No

2.1.2 Include Files

The following files are delivered with the Bluetooth SDK and must be included in C application projects.

Library	Explanation	When needed
native_gecko.h	Bluetooth stack API for standalone applications without Micrium RTOS	Must be included in standalone C applications where both Bluetooth stack and application run in a Wireless Gecko.
rtos_gecko.h	Bluetooth stack API for standalone applications with Micrium RTOS	Required when the Bluetooth stack is used together with Micrium RTOS.
ncp_gecko.h	Bluetooth stack API for NCP applications	Must be included in NCP applications where the host controls the device via BGAPI protocol over UART
gecko_configuration.h	Bluetooth stack configuration	Included automatically.
bg_errorcodes.h	Error codes produced by the Bluetooth stack	Included automatically.
bg_gattdb_def.h	Bluetooth GATT database structure definition	Included automatically.
bg_types.h	Simple data type definitions and structures	Included automatically.
gecko_bglib.h	An adaptation layer between host application and BGAPI serial protocol	Must be included in C applications developed for external hosts.
host_gecko.h	Bluetooth API for host (NCP) applications	Must be included in C applications developed for external hosts.

2.1.3 Platform Components

The following components are delivered with the Bluetooth SDK. The platform components are under the ***platform*** folder.

Folder	Explanation
bootloader	Gecko Bootloader source code and project files.
CMSIS	Silicon Laboratories CMSIS-CORE device headers. Documentation
Device	EFR32BG and EFR32MG device files. Documentation
emdrv	A set of function-specific high performance drivers for EFR32 on-chip peripherals. Drivers are typically DMA based and are using all available low-energy features. For most drivers, the API offers both synchronous and asynchronous functions. Documentation
emlib	A low-level peripheral support library that provides a unified API for all EFM32, EZR32 and EFR32 MCUs and SoCs from Silicon Laboratories. Documentation
middleware	Display driver for WSTK development kits.
radio	Silicon Labs RAIL (Radio Abstraction Interface Layer) library

2.1.4 Example and Demo Applications

Folder	Explanation
examples_btds	A placeholder folder for Bluetooth Developer Studio-generated C projects. When the projects are generated to this folder they compile without any modifications to the project files.
examples_ncp_host	Host code examples demonstrating NCP usage and other use cases. **Empty**: Minimalistic host code demonstrating advertisements and connections over NCP. **ota_dfu**: An application using WSTK in NCP mode to discover and perform an OTA update to a remote device. **uart_dfu**: An application demonstrating how to make firmware update over UART interface. **voice_over_bluetooth_low_energy_app:** GATT client side application for “Voice over Bluetooth Low Energy” example for Thunderboard Sense.

2.1.5 Bluetooth Developer Studio Plug-Ins

The Silicon Labs Bluetooth Developer Studio plug-ins are delivered in the SDKs btds_plugins folder. To use them, copy them to the Bluetooth Developer Studio’s installation’s subfolder named Plugins. (Typically C:\Program Files (x86)\Bluetooth SIG\Bluetooth Developer Studio\Plugins).

3. The Development Tools

3.1 Simplicity Studio

Simplicity Studio is a free-of-charge Silicon Labs Integrated Development Environment (IDE) and a collection of value-add tools that developers can use to develop, debug, and analyze their Bluetooth applications. For more information about using Simplicity Studio and the tools within it to develop Bluetooth applications, see *QSG139: Bluetooth® Development with Simplicity Studio*.

With Bluetooth SDK 2.3.0 and newer either the GCC or IAR C compilers can be used with Simplicity Studio.

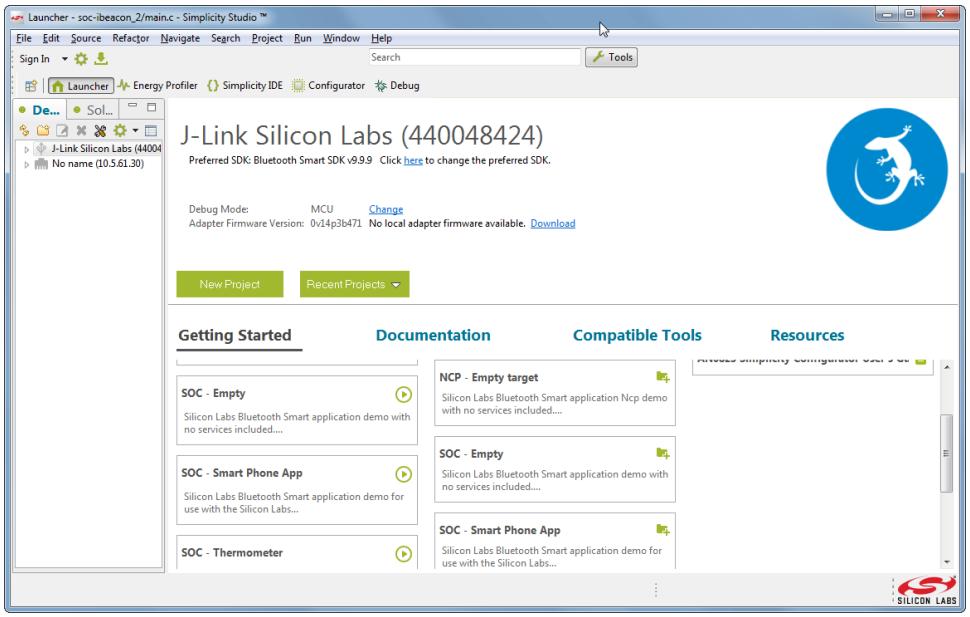


Figure 3.1. Simplicity Studio v4

Download and install Simplicity Studio from: http://www.silabs.com/simplicity-studio.

3.1.1 IDE and debugger

Simplicity Studio contains an Eclipse-based IDE, which can be used to develop and debug C applications for the Wireless Geckos.

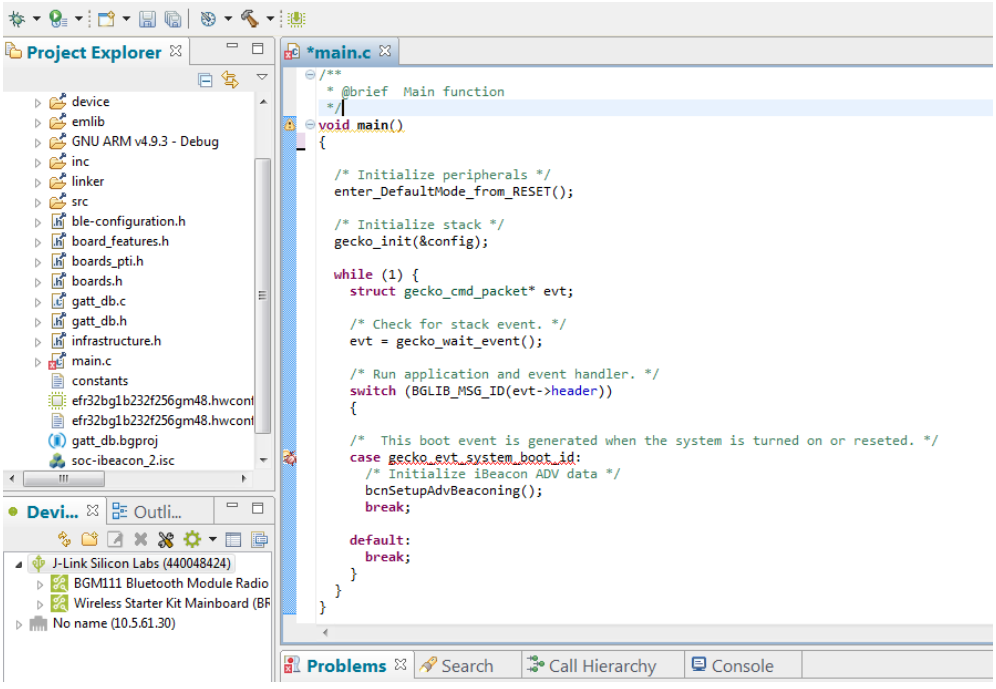


Figure 3.2. Simplicity Studio IDE and Debugger

3.1.2 Graphical GATT Editor

Beginning with the Bluetooth SDK 2.1.0 release, Simplicity Studio contains a graphical Bluetooth GATT database editor, which allows easy configuration of Bluetooth Profiles, Services, and Characteristics without directly editing the XML file used during the build process. The GATT editor also allows you to configure all the options for the services and characteristics, such as the UUID, security and Attribute Protocol properties, and other options.

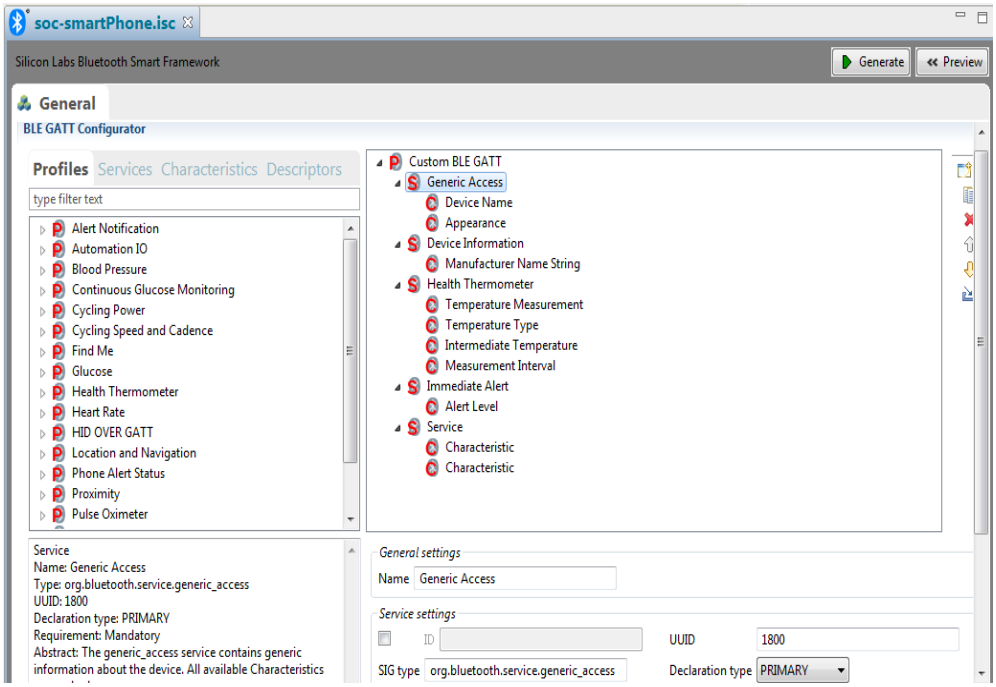


Figure 3.3. Simplicity Studio GATT Editor

3.1.3 Energy Profiler

The Energy Profiler is one of the value-add tool in Simplicity Studio. It enables developers to quickly and easily check the current and energy consumption of their applications.

The Energy Profiler features an enhanced energy graphing capability that has the familiar look and feel of an oscilloscope. The developer can now zoom in on the X (time) and Y (power) axes of the energy graph to analyze the details of energy consumption with greater precision. In addition, the Profiler provides a direct correlation between the energy graph, function analyzer, and application code. This three-way correlation capability enhances the developer's ability to optimize designs for ultra-low energy consumption.

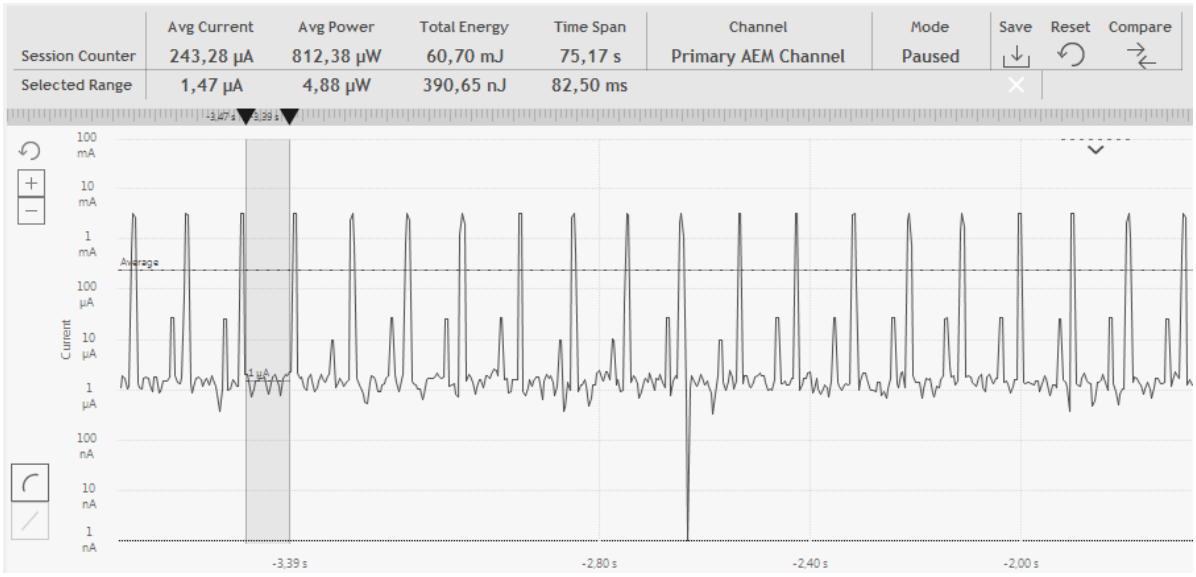


Figure 3.4. Energy Profiler

3.1.4 Packet Trace

Silicon Labs Network Analyzer is a free-of-charge packet capture and debugging tool that can be used to debug Bluetooth connectivity between Wireless Geckos and other Bluetooth devices. It significantly accelerates the network and application development process with graphical views of network traffic, activity, and duration.

The Packet Trace application captures the packets directly from the Packet Trace Interface (PTI) available on the Wireless Gecko SoCs and modules. It therefore provides a more accurate capture of the packets compared to air-based capture.

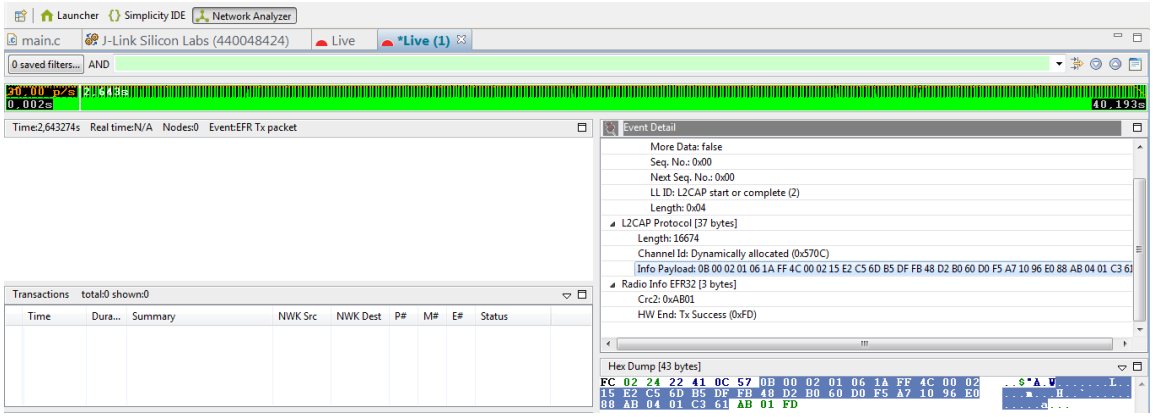


Figure 3.5. Bluetooth Traffic Capture with Packet Trace

3.1.5 Simplicity Commander

Simplicity Commander is a simple flashing tool, which can be used to flash firmware images, erase flash, lock and unlock debug access, and write-protect flash pages via the J-Link interface. Both GUI and CLI (Command Line Interface) are available. See *UG162: Simplicity Commander Refernce Guide* for more information.

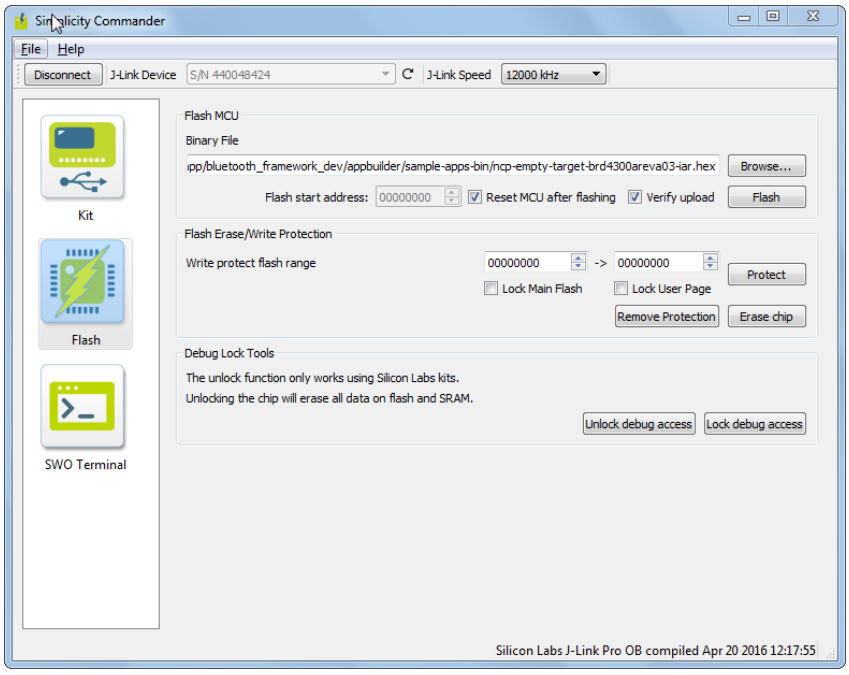


Figure 3.6. Simplicity Commander

3.1.6 BGTool Application

The BGTool application can be used to test and evaluate the Bluetooth SoCs and modules, and it can be used to control the Bluetooth hardware using the BGAPI Serial Protocol (NCP) over a Serial/UART interface.

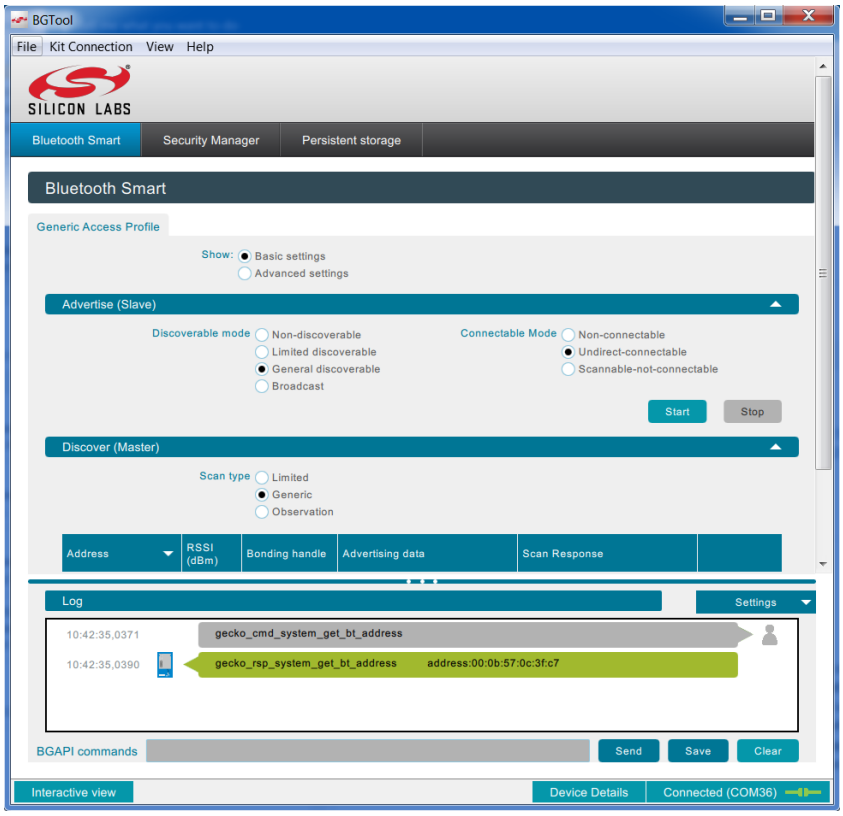


Figure 3.7. BGTool Application

3.2 IAR Embedded Workbench

IAR's Embedded Workbench can also be used as an IDE for developing and debugging Bluetooth applications. It is recommended to use 7.80 or newer version of IAR.

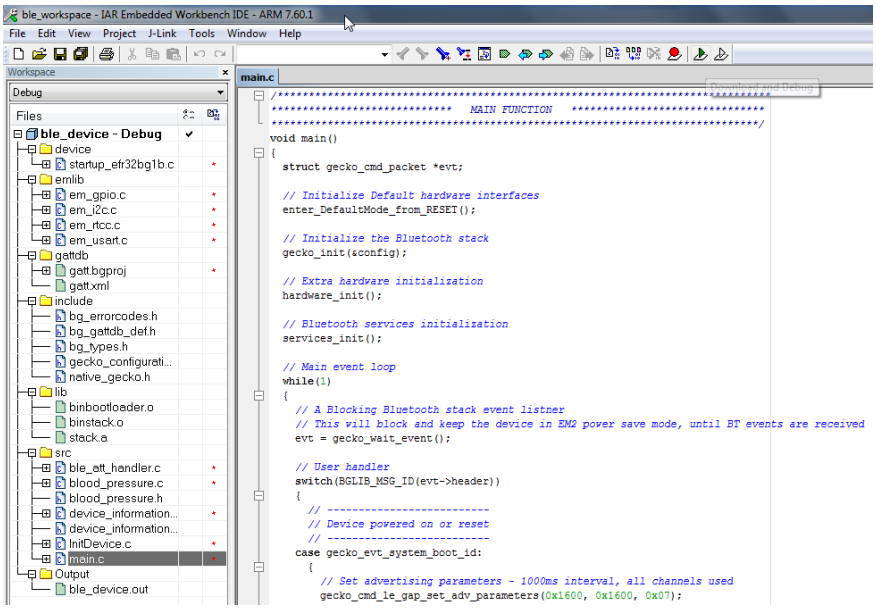


Figure 3.8. IAR Embedded Workbench

3.3 Bluetooth Developer Studio

Bluetooth Developer Studio is a powerful application development and debugging tool from Bluetooth SIG. It provides a simple interface for adding and modifying Bluetooth services and characteristics based on your application's needs.

Silicon Labs provides plug-ins for the Bluetooth Developer Studio, which auto-generate the GATT data base in the XML format used by the SDK as well as C application code implementing the service and characteristic handlers for the selected configuration.

Bluetooth Developer Studio greatly speeds up and simplifies Bluetooth application development.

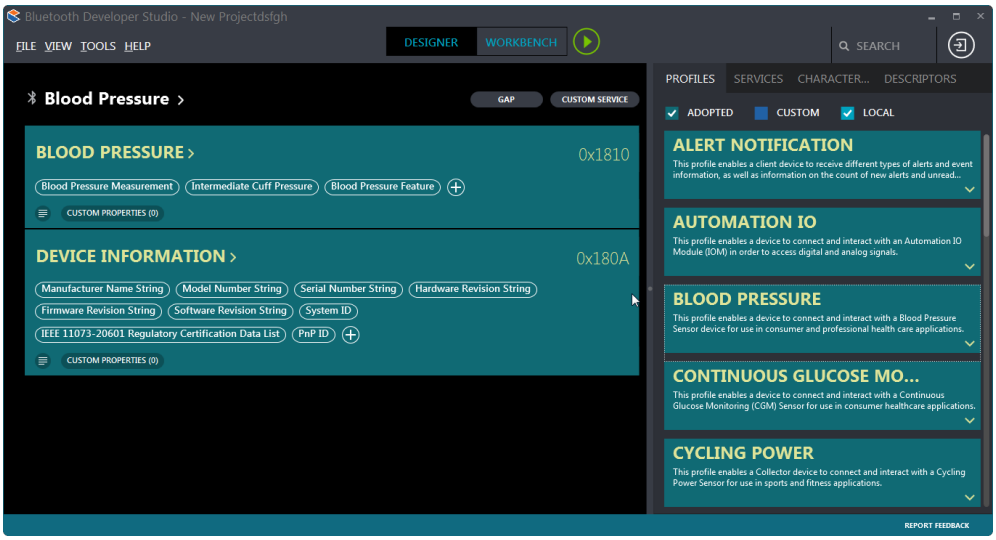


Figure 3.9. Bluetooth Developer Studio

Get started with the Bluetooth Developer Studio:

https://www.bluetooth.com/~/media/developer-studio/index

4. Getting Started with Bluetooth Development

4.1 Using Silicon Labs Simplicity Studio

1. Install Simplicity Studio v4 or newer from here.
2. Read document *QSG139: Bluetooth® Development with Simplicity Studio* for more detailed instructions.

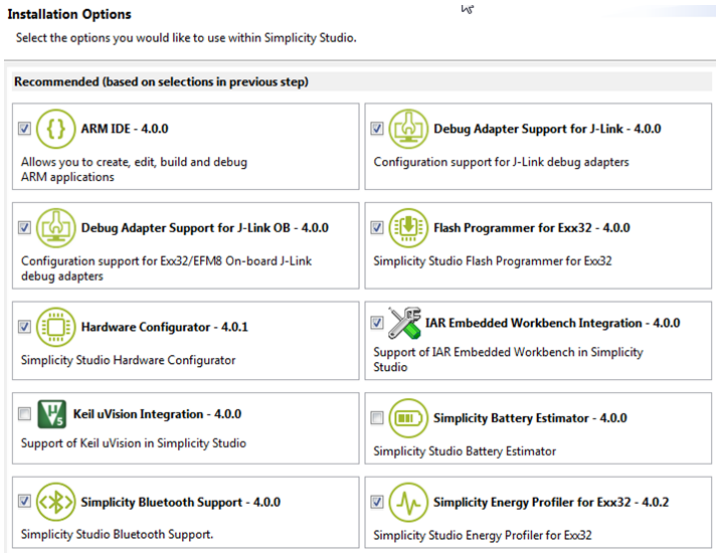


Figure 4.1. Installing Simplicity Studio

The Bluetooth SDK will typically be installed to the following folder on Windows:

C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suites

4.2 Using IAR Embedded Workbench

1. Follow the instructions in *QSG139: Bluetooth® Development with Simplicity Studio* to download and install the Bluetooth SDK.
2. Download and Install IAR Embedded Workbench (7.80 or newer is recommended).

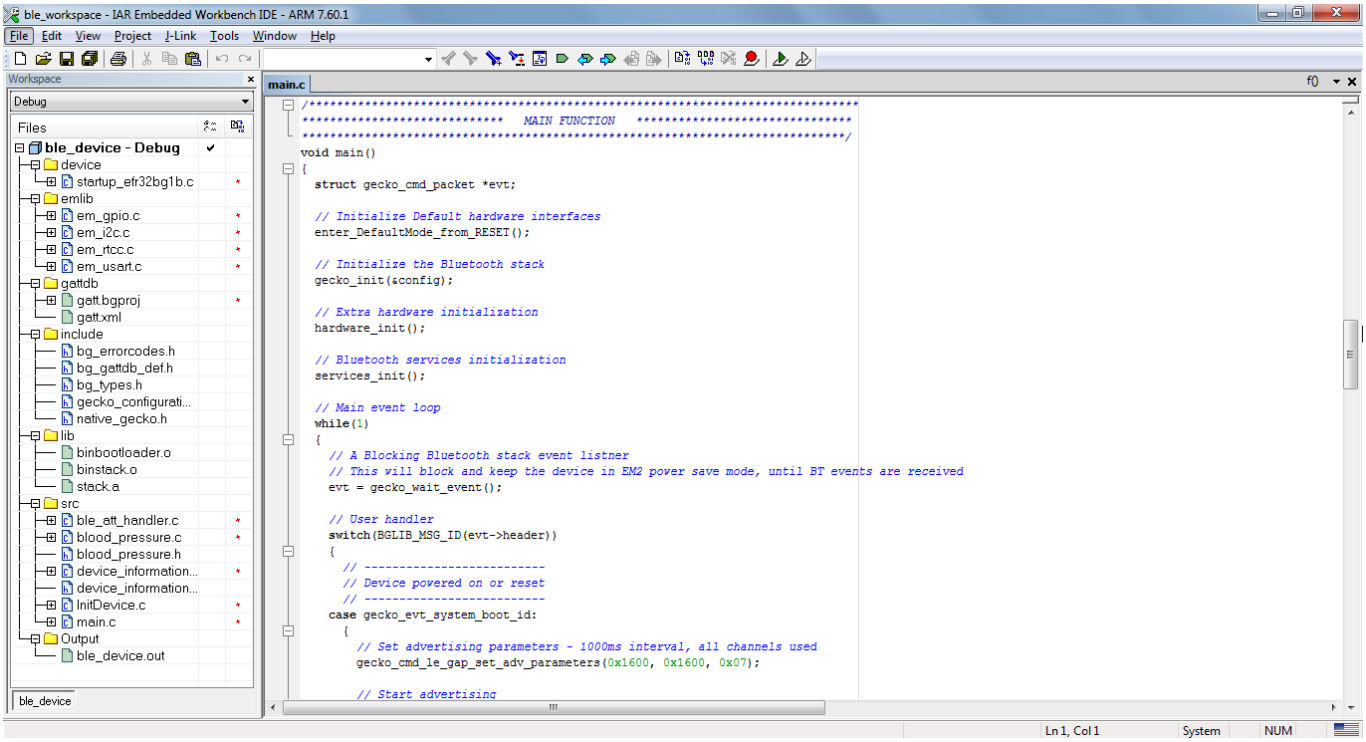


Figure 4.2. IAR Embedded Workbench

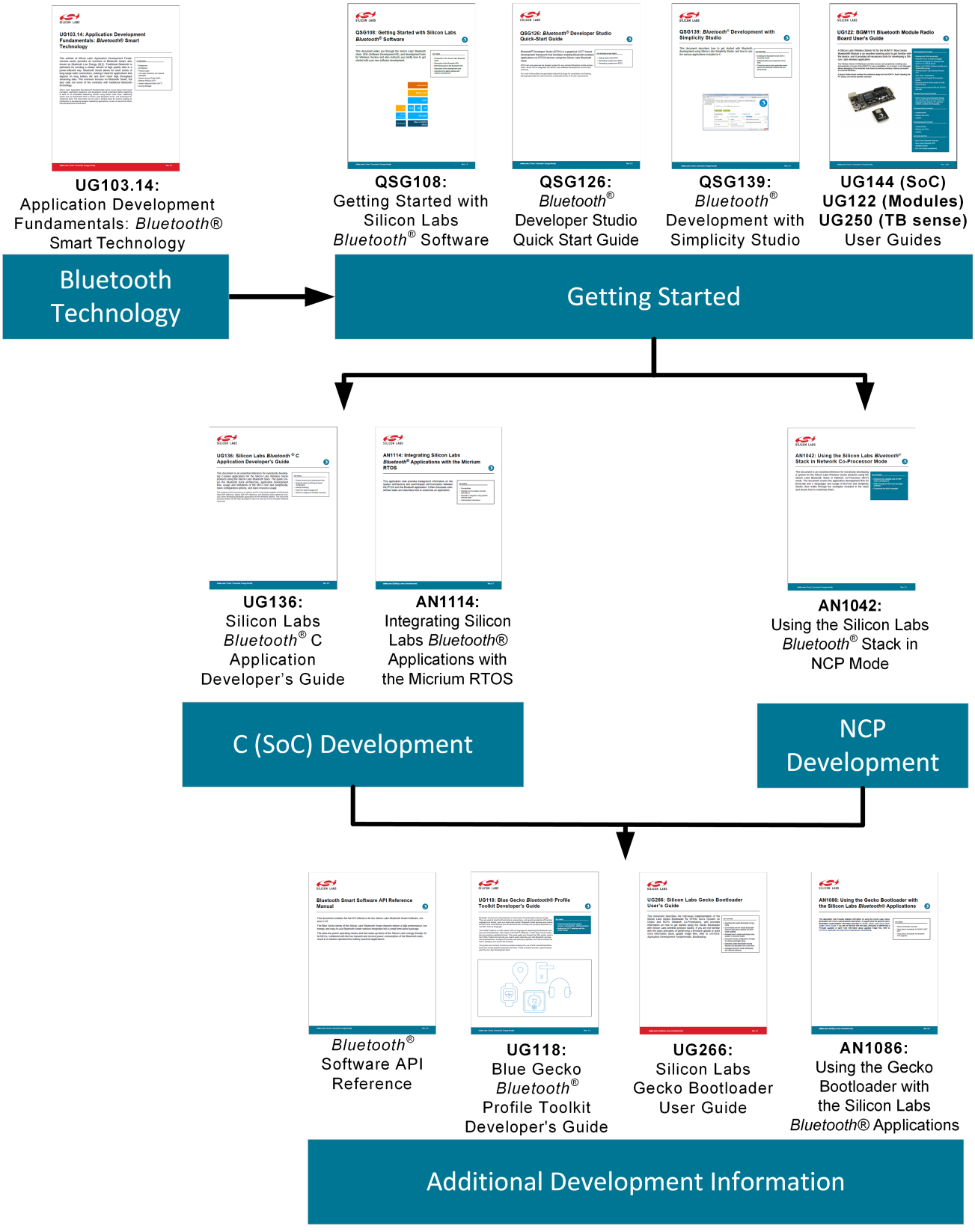
To learn more about generic Bluetooth C application development, see *UG136: Silicon Labs Bluetooth C Application Developer's Guide*.

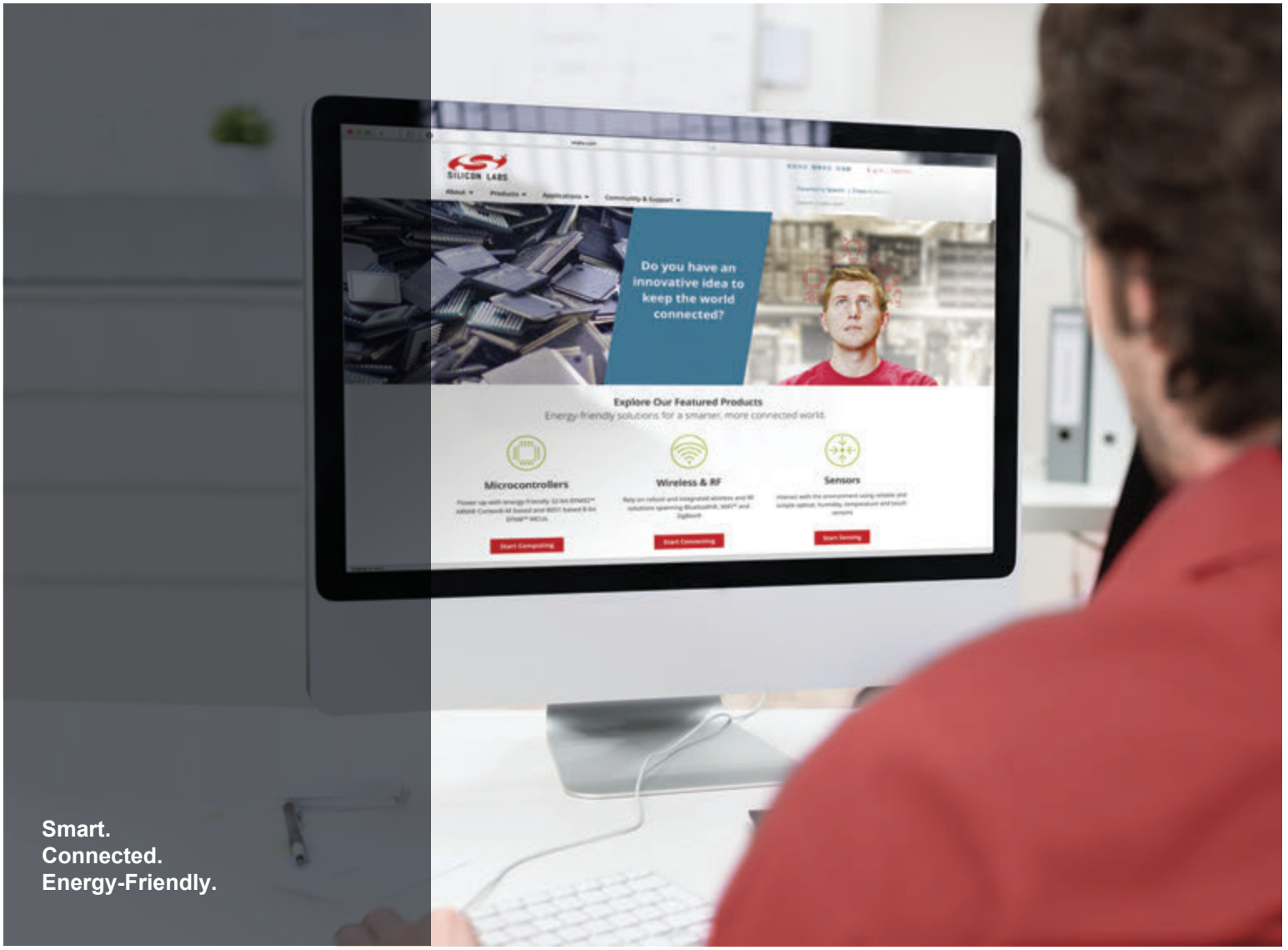
To learn how to auto-generate IAR projects, GATT database and source code with Bluetooth Developer Studio, see *QSG126: Bluetooth Developer Studio Quick Start Guide*.

4.3 Getting Started with NCP Development

1. Follow the instructions in *QSG139: Bluetooth® Development with Simplicity Studio* to download and install the Bluetooth SDK.
2. Please read: *AN1042: Using the Silicon Labs Bluetooth® Stack in Network Co-Processor Mode*.

5. Documentation





Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

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

http://www.silabs.com