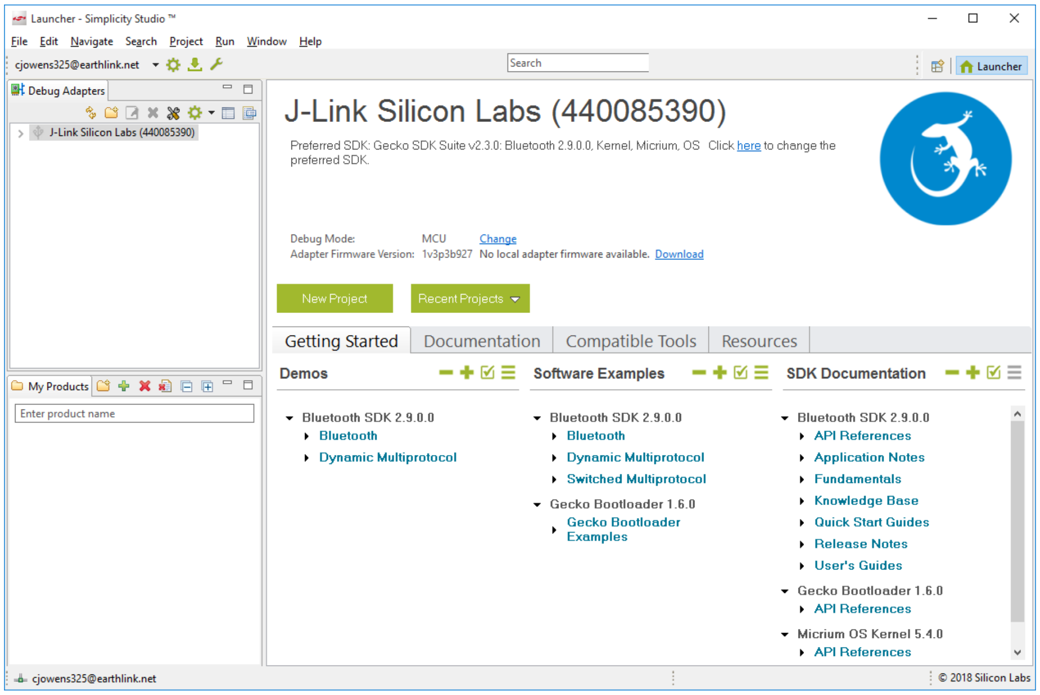
QSG139: *Bluetooth*® Development with Simplicity Studio

This document describes how to get started with Bluetooth development using Silicon Labs Simplicity Studio, and how to use the various applications included with it.

KEY POINTS

- Installing the Bluetooth SDK in Simplicity Studio.
- Introducing the key components of the SDK.
- Tutorial on how to start application development for Bluetooth devices with Simplicity Studio.



J-Link Silicon Labs (440085390)

Preferred SDK: Gecko SDK Suite v2.3.0: Bluetooth 2.9.0.0, Kernel, Micrium, OS Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 1v3p3b927 No local adapter firmware available. Download

Demos
- Bluetooth SDK 2.9.0.0
  - Bluetooth
  - Dynamic Multiprotocol

Software Examples
- Bluetooth SDK 2.9.0.0
  - Bluetooth
  - Dynamic Multiprotocol
  - Switched Multiprotocol
- Gecko Bootloader 1.6.0
  - Gecko Bootloader Examples

SDK Documentation
- Bluetooth SDK 2.9.0.0
  - API References
  - Application Notes
  - Fundamentals
  - Knowledge Base
  - Quick Start Guides
  - Release Notes
  - User's Guides
- Gecko Bootloader 1.6.0
  - API References
- Micrium OS Kernel 5.4.0
  - API References

1 Introduction

Simplicity Studio is a free Eclipse-based Integrated Development Environment (IDE), and a collection of value-add tools. Provided by Silicon Labs, developers can use Simplicity Studio to develop, debug and analyze their Bluetooth and other Silicon Labs SDK (Software Development Kit) applications. Its main goal is to reduce development time so that you can focus on your application instead of researching the Bluetooth specification and hardware reference manuals. See section 3 Functionality in the Launcher Perspective for a review of Simplicity Studio's features.

If you have already installed Simplicity Studio with a different protocol stack, see section 3.1 Downloading Updates or Installing Additional Components for instructions on updating the installation with the Bluetooth stack.

1.1 Prerequisites

As well as Simplicity Studio v4, you should have the following before beginning application development:

- A basic understanding of Bluetooth technology and terminology. *UG104.13: Application Development Fundamentals: Bluetooth Technology* provides a good starting point if you have not yet learned about Bluetooth.
- A compatible compiler:
  - Simplicity Studio comes with free GCC C-compiler.
  - IAR Embedded Workbench for ARM (IAR-EWARM) can also be used as the compiler for all Silicon Labs protocols. See the release notes for the Bluetooth SDK for the compatible IAR-EWARM version.
- A registered account at Silicon Labs is required in order to download the Silicon Labs Bluetooth SDK.
  - You can register at https://siliconlabs.force.com/apex/SL_CommunitiesSelfReg?form=short.

To get a 30-day evaluation license for IAR:

- Go to the Silicon Labs support portal at https://www.silabs.com/support.
- Scroll down to the bottom of the page, and click **Contact Support**
- If you are not already signed in, sign in.
- Click the Software Releases tab. In the View list select **Development Tools**. Click **Go**. In the results is a link to the IAR-EWARM version noted above.
- Download the IAR package (takes approximately 1 hour).
- Install IAR.
- In the IAR License Wizard, click **Register with IAR Systems to get an evaluation license**.
- Complete the registration and IAR will provide a 30-day evaluation license.
- Once IAR-EWARM is installed, the next time Simplicity Studio starts it will automatically detect and configure the IDE to use IAR-EWARM.

1.2 Support

You can access the Silicon Labs support portal at https://www.silabs.com/support through Simplicity Studio's Resources tab, as described in section 3.4 Accessing Documentation and Other Resources. Use the support portal to contact Customer Support for any questions you might have during the development process.

1.3 Documentation

Stack documentation is accessed through Simplicity Studio, as described in section 3.4 Accessing Documentation and Other Resources. Simplicity Studio also provides links to hardware documentation and other application notes.

2 Getting Started

2.1 Connect your Hardware

Connect your WSTK using a USB cable to the PC on which you will install Simplicity Studio. By having it connected when Simplicity Studio installs, Simplicity Studio will automatically obtain the relevant additional resources it needs.

Note: For best performance in Simplicity Studio, be sure that the power switch on your WSTK is in the Advanced Energy Monitoring or "AEM" position as shown in the following figure.

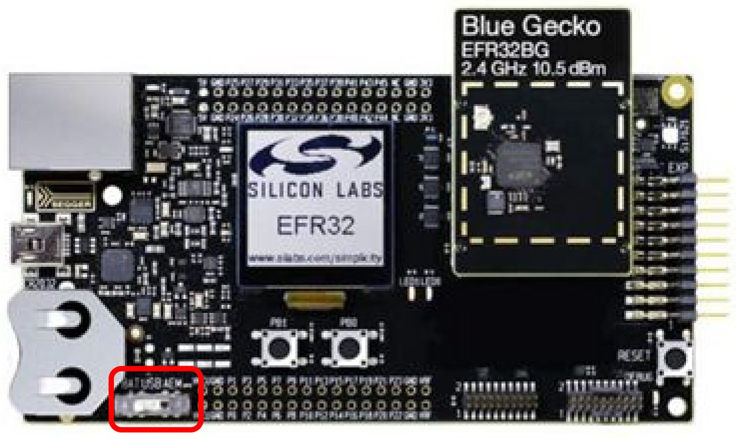
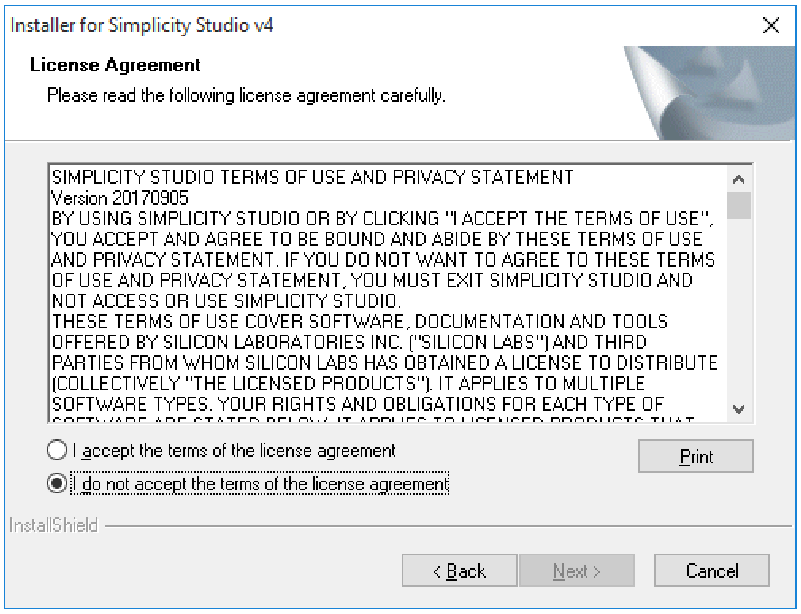


Figure 1. EFR32BG12 on a WSTK

2.2 Installing Simplicity Studio and the Gecko Suite with the Bluetooth Stack

1. Run the Simplicity Studio installation application.
2. When Simplicity Studio first launches, it presents a License Agreement dialog. Accept the terms of the agreement and click **Nex**t.



Installer for Simplicity Studio v4

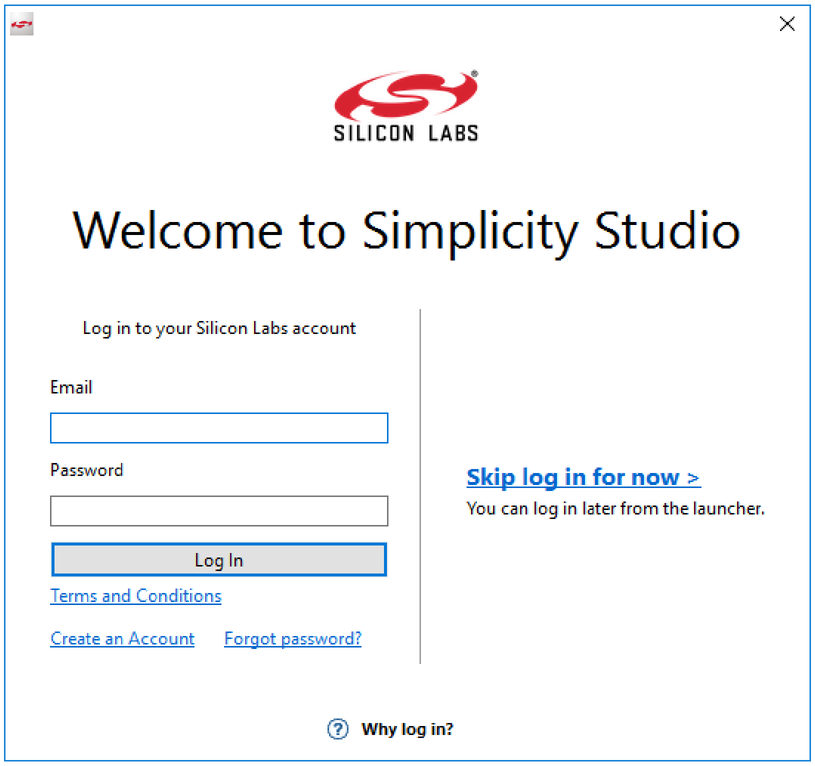
License Agreement

Please read the following license agreement carefully.

SIMPLICITY STUDIO TERMS OF USE AND PRIVACY STATEMENT
Version 20170905
BY USING SIMPLICITY STUDIO OR BY CLICKING "I ACCEPT THE TERMS OF USE", YOU ACCEPT AND AGREE TO BE BOUND AND ABIDE BY THESE TERMS OF USE AND PRIVACY STATEMENT. IF YOU DO NOT WANT TO AGREE TO THESE TERMS OF USE AND PRIVACY STATEMENT, YOU MUST EXIT SIMPLICITY STUDIO AND NOT ACCESS OR USE SIMPLICITY STUDIO.
THESE TERMS OF USE COVER SOFTWARE, DOCUMENTATION AND TOOLS OFFERED BY SILICON LABORATORIES INC. ("SILICON LABS") AND THIRD PARTIES FROM WHOM SILICON LABS HAS OBTAINED A LICENSE TO DISTRIBUTE (COLLECTIVELY "THE LICENSED PRODUCTS"). IT APPLIES TO MULTIPLE SOFTWARE TYPES. YOUR RIGHTS AND OBLIGATIONS FOR EACH TYPE OF

I accept the terms of the license agreement [✓ I do not accept the terms of the license agreement]

3. Next you are invited to log in. Log in using your support account username and password. Although you can skip log in here, you must be logged in to access protected content.

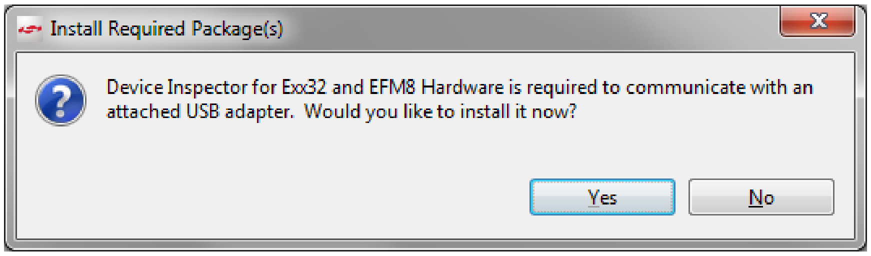


Welcome to Simplicity Studio

Log in to your Silicon Labs account

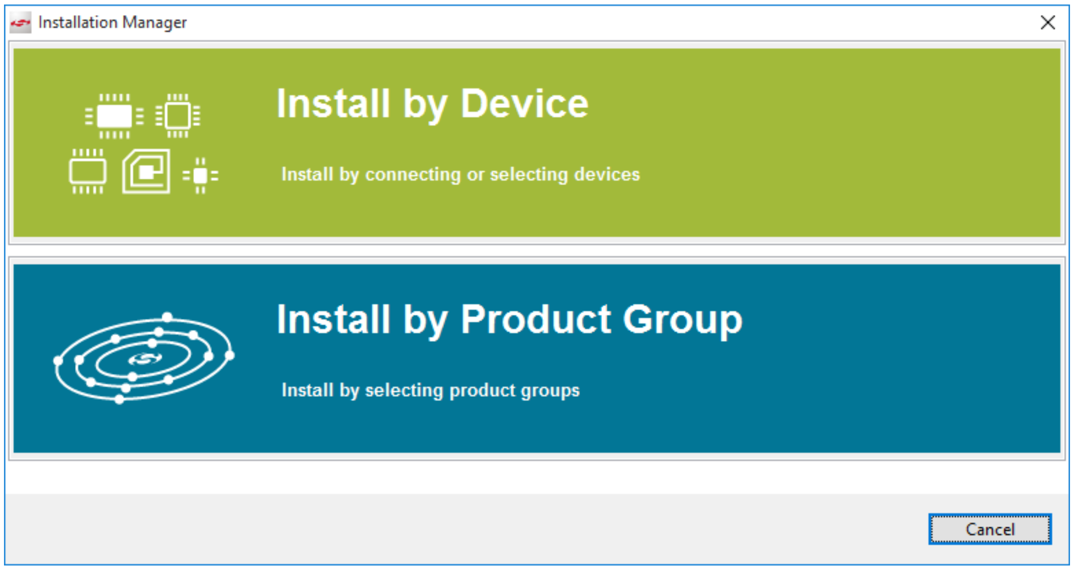
You can log in later from the launcher.

4. After login, Simplicity Studio adds software information. Once initial software installation is complete, Simplicity Studio checks for connected hardware. If you have the WSTK connected by USB cable, Simplicity Studio will detect the USB cable and prompt you to download a Device Inspector. Click **Yes**.



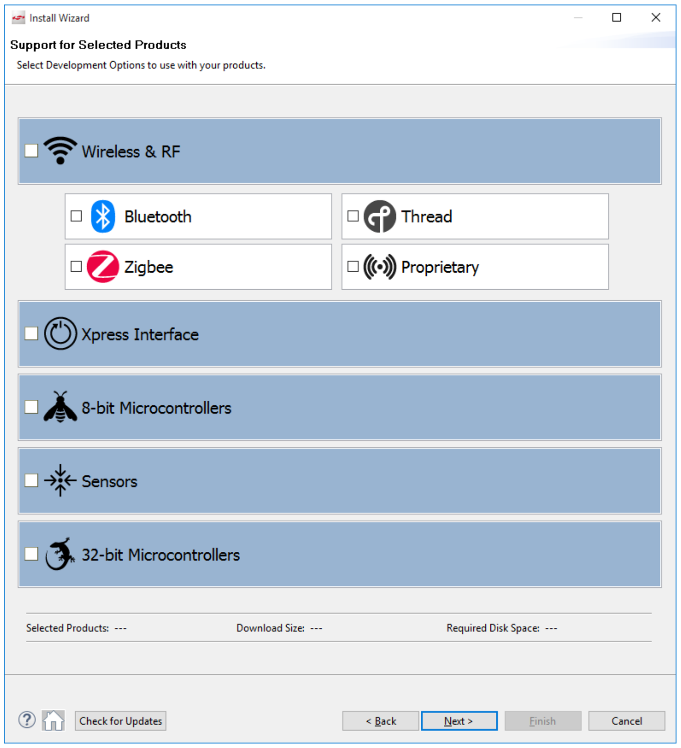
Device Inspector for Exx32 and EFM8 Hardware is required to communicate with an attached USB adapter. Would you like to install it now?

5. After some additional items are installed, you are offered the option of installing by device (steps 8 and 9) or installing by product group (step 7).

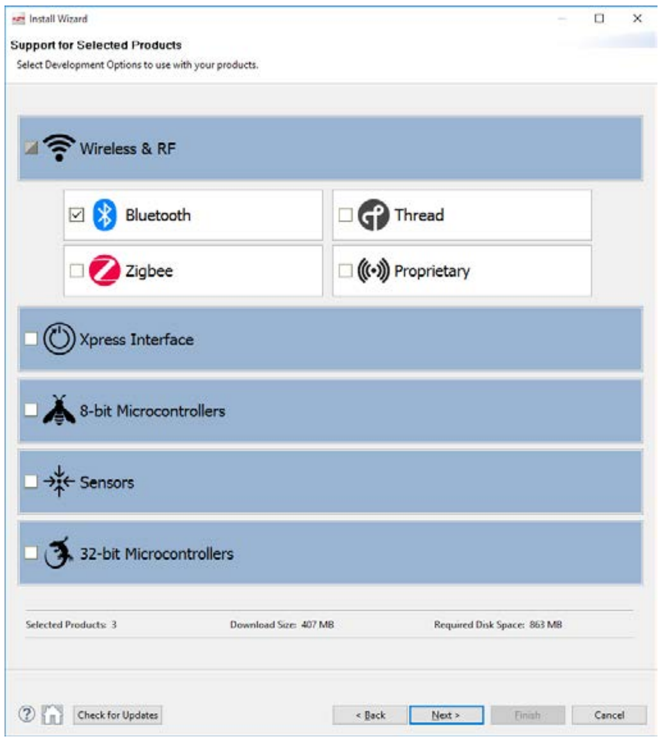

Installation Manager
Install by Device
Install by connecting or selecting devices
Install by Product Group
Install by selecting product groups
Cancel

Throughout these procedures at any time you can click Home (🏠) to return to this dialog.

6. If you click **Install by Product Group** you are offered a list of product groups. Click the SDKs you want to install, or click Wireless & RF to check all. Click **Next** and go to step 10. If you plan to work on the RAIL/Bluetooth Dynamic Multiprotocol examples, select both Bluetooth and Proprietary groups.

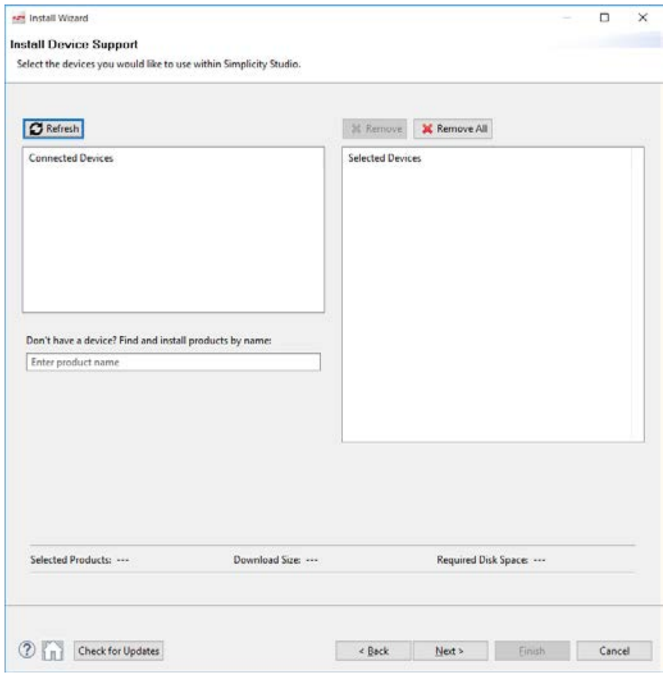


Selected Products: --- Download Size: --- Required Disk Space: ---

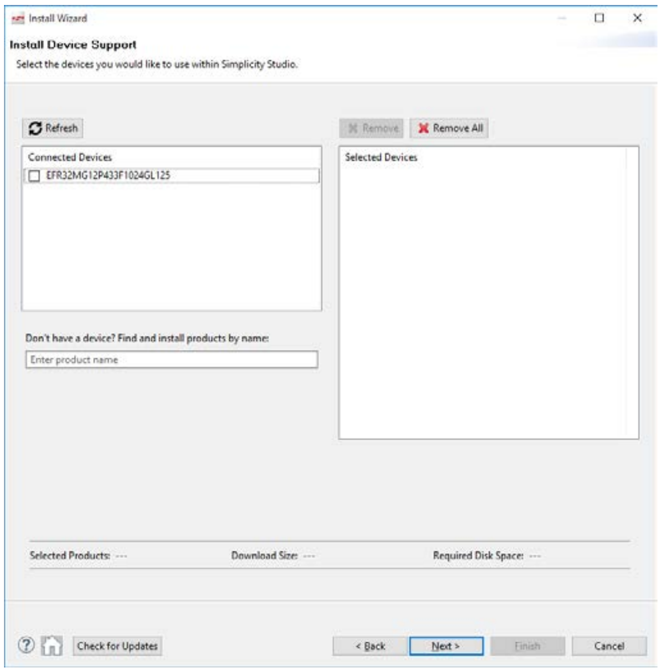


Selected Products: 3 Download Size: 407 MB Required Disk Space: 863 MB

7. If you click Install by Devices, an Install Device Support dialog appears. After a short delay, it shows your connected device. If the connected device does not show, click **Refresh**. The following figure shows the Install Device Support dialog before and after the connected device is displayed.



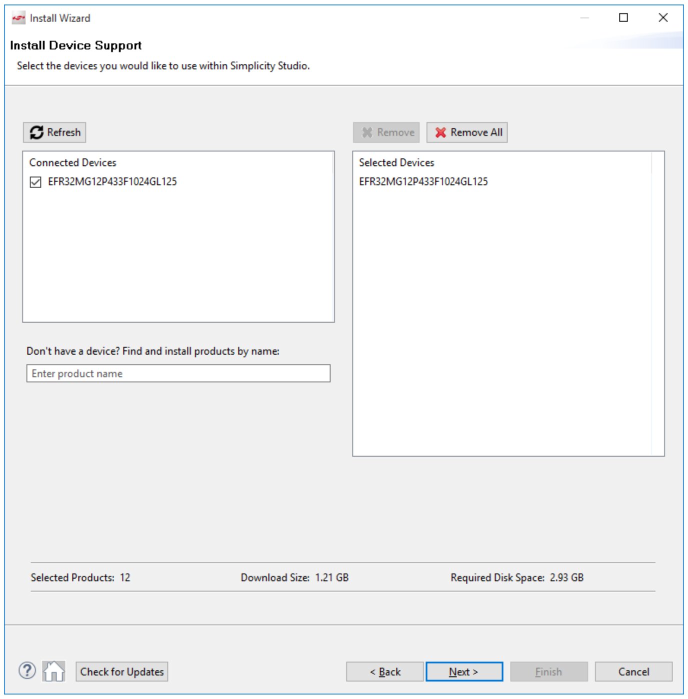
Selected Products: --- Download Size: --- Required Disk Space: ---



Connected Devices: EFR32MG12P433F1024GL125

Selected Products: --- Download Size: --- Required Disk Space: ---

8. Click the checkbox next to the device to select it. Selecting the device allows Simplicity Studio to present the relevant software packages for you to install. Click **Next**.



Install Wizard

Install Device Support

Select the devices you would like to use within Simplicity Studio.

Connected Devices

☑ EFR32MG12P433F1024GL125

Selected Devices

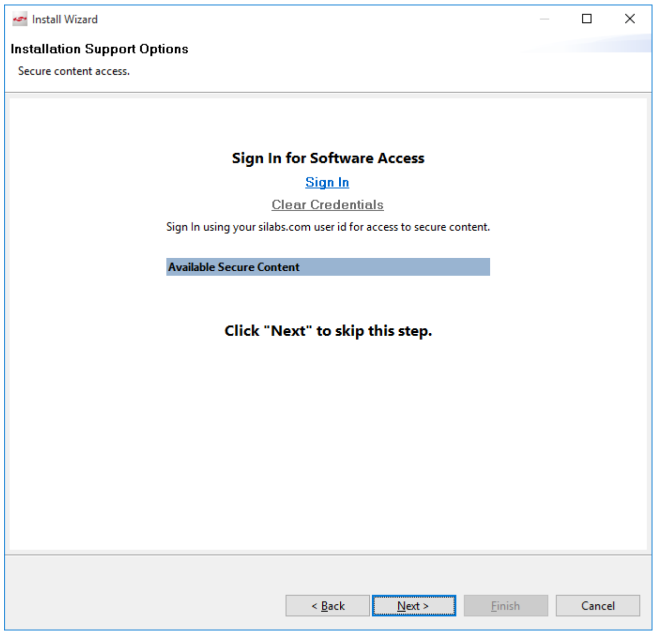
EFR32MG12P433F1024GL125

Don't have a device? Find and install products by name:

Enter product name

Selected Products: 12 Download Size: 1.21 GB Required Disk Space: 2.93 GB

9. The next dialog varies depending on whether or not you have signed in. If you have not signed in, you have no access to restricted content and must sign in first. Once you have signed in and see Bluetooth SDK on the list of available content, click **Next**.



Install Wizard

Installation Support Options

Secure content access.

Sign In for Software Access

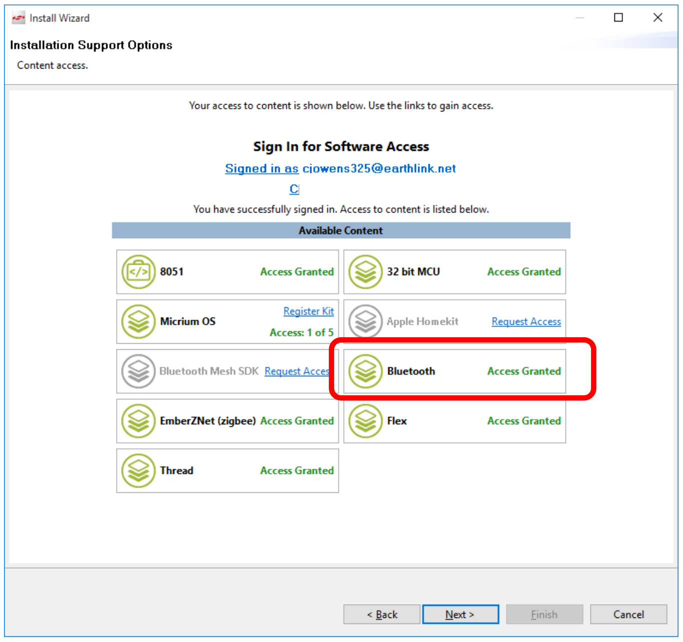
Sign In

Clear Credentials

Sign In using your silabs.com user id for access to secure content.

Available Secure Content

Click "Next" to skip this step.



Install Wizard

Installation Support Options

Content access.

Your access to content is shown below. Use the links to gain access.

Sign In for Software Access

Signed in as ciowens325@earthlink.net

C

You have successfully signed in. Access to content is listed below.

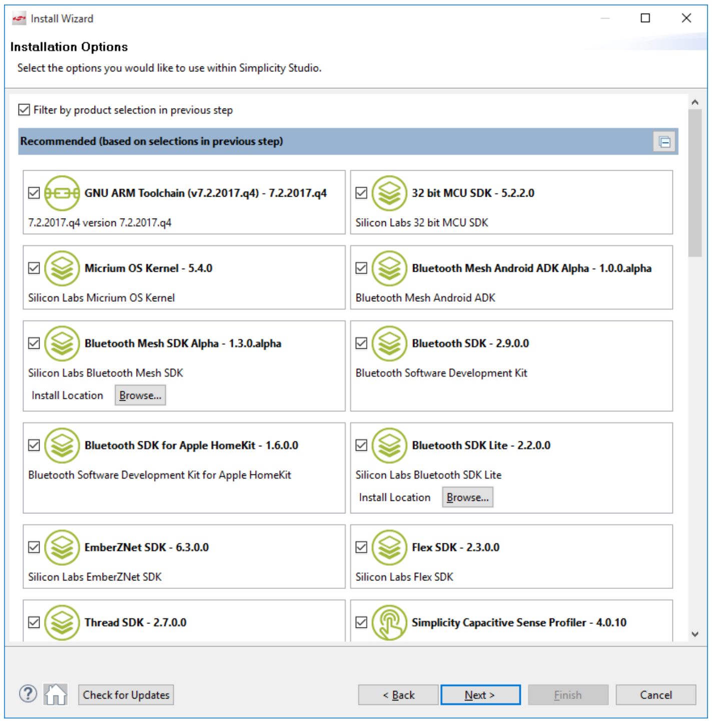
Available Content

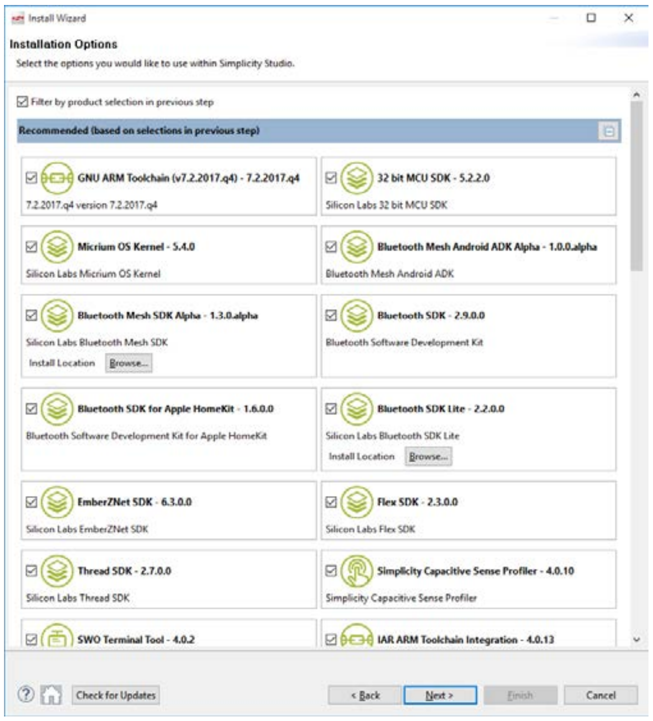
8051	Access Granted	32 bit MCU	Access Granted
Micrium OS	Register Kit Access: 1 of 5	Apple Homekit	Request Access
Bluetooth Mesh SDK	Request Access	Bluetooth	Access Granted
EmberZNet (zigbee)	Access Granted	Flex	Access Granted
Thread	Access Granted		

10. The **Installation Options** dialog shows the tools and software packages that can be installed (your versions may be different). The following shows Installation options after selecting the Proprietary product group, and after selecting an EFR32MG device. In both views you can uncheck anything you don't want to install. If you have installed by Product Group, the selection is filtered more specifically to your needs than if you have installed by device, and installing all checked options is recommended. If you have installed by device, and are unchecking items:

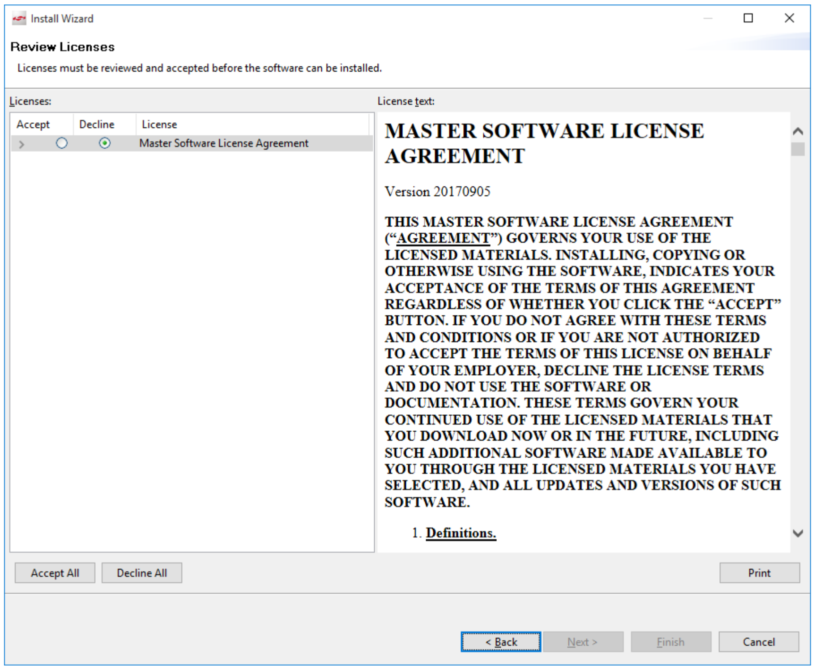
- If you plan to use GCC, leave GNU ARM Toolchain checked..
- If you plan to work on the RAIL/Bluetooth Dynamic Multiprotocol examples, leave Micrium OS Kernel and the Flex SDK checked.

Click **Next**.


Install Wizard
Installation Options
Select the options you would like to use within Simplicity Studio.
☑ Filter by product selection in previous step
Recommended (based on selections in previous step)
☑ GNU ARM Toolchain (v7.2.2017.q4) - 7.2.2017.q4 — 7.2.2017.q4 version 7.2.2017.q4
☑ 32 bit MCU SDK - 5.2.2.0 — Silicon Labs 32 bit MCU SDK
☑ Micrium OS Kernel - 5.4.0 — Silicon Labs Micrium OS Kernel
☑ Bluetooth Mesh Android ADK Alpha - 1.0.0.alpha — Bluetooth Mesh Android ADK
☑ Bluetooth Mesh SDK Alpha - 1.3.0.alpha — Silicon Labs Bluetooth Mesh SDK — Install Location Browse...
☑ Bluetooth SDK - 2.9.0.0 — Bluetooth Software Development Kit
☑ Bluetooth SDK for Apple HomeKit - 1.6.0.0 — Bluetooth Software Development Kit for Apple HomeKit
☑ Bluetooth SDK Lite - 2.2.0.0 — Silicon Labs Bluetooth SDK Lite — Install Location Browse...
☑ EmberZNet SDK - 6.3.0.0 — Silicon Labs EmberZNet SDK
☑ Flex SDK - 2.3.0.0 — Silicon Labs Flex SDK
☑ Thread SDK - 2.7.0.0
☑ Simplicity Capacitive Sense Profiler - 4.0.10
Check for Updates | < Back | Next > | Finish | Cancel


Install Wizard
Installation Options
Select the options you would like to use within Simplicity Studio.
☑ Filter by product selection in previous step
Recommended (based on selections in previous step)
☑ GNU ARM Toolchain (v7.2.2017.q4) - 7.2.2017.q4 — 7.2.2017.q4 version 7.2.2017.q4
☑ 32 bit MCU SDK - 5.2.2.0 — Silicon Labs 32 bit MCU SDK
☑ Micrium OS Kernel - 5.4.0 — Silicon Labs Micrium OS Kernel
☑ Bluetooth Mesh Android ADK Alpha - 1.0.0.alpha — Bluetooth Mesh Android ADK
☑ Bluetooth Mesh SDK Alpha - 1.3.0.alpha — Silicon Labs Bluetooth Mesh SDK — Install Location Browse...
☑ Bluetooth SDK - 2.9.0.0 — Bluetooth Software Development Kit
☑ Bluetooth SDK for Apple HomeKit - 1.6.0.0 — Bluetooth Software Development Kit for Apple HomeKit
☑ Bluetooth SDK Lite - 2.2.0.0 — Silicon Labs Bluetooth SDK Lite — Install Location Browse...
☑ EmberZNet SDK - 6.3.0.0 — Silicon Labs EmberZNet SDK
☑ Flex SDK - 2.3.0.0 — Silicon Labs Flex SDK
☑ Thread SDK - 2.7.0.0 — Silicon Labs Thread SDK
☑ Simplicity Capacitive Sense Profiler - 4.0.10 — Simplicity Capacitive Sense Profiler
☑ SWO Terminal Tool - 4.0.2
☑ IAR ARM Toolchain Integration - 4.0.13
Check for Updates | < Back | Next > | Finish | Cancel

11. Studio displays a Review Licenses dialog. Accept the licenses shown and click **Finish**. Note that this dialog will present again if in the future you install a component with a separate license.


Install Wizard
Review Licenses
Licenses must be reviewed and accepted before the software can be installed.

Licenses:

Accept	Decline	License
○	●	Master Software License Agreement

License text:

MASTER SOFTWARE LICENSE AGREEMENT

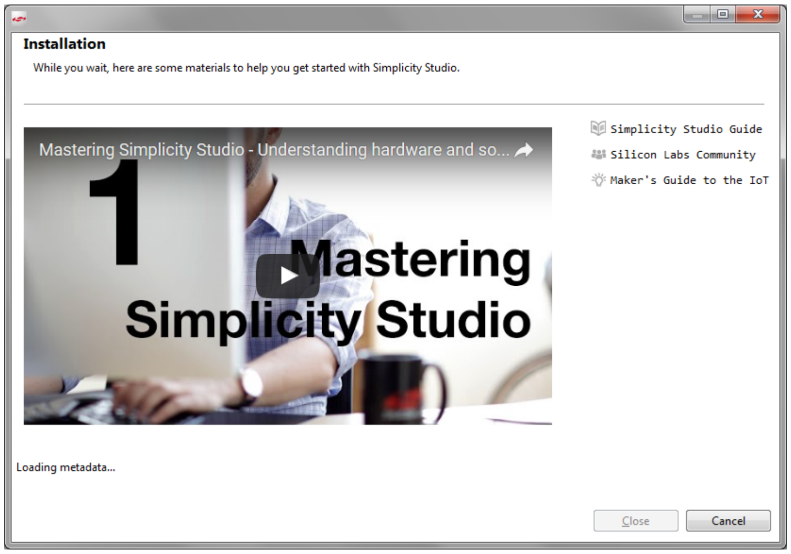
Version 20170905

THIS MASTER SOFTWARE LICENSE AGREEMENT ("AGREEMENT") GOVERNS YOUR USE OF THE LICENSED MATERIALS. INSTALLING, COPYING OR OTHERWISE USING THE SOFTWARE, INDICATES YOUR ACCEPTANCE OF THE TERMS OF THIS AGREEMENT REGARDLESS OF WHETHER YOU CLICK THE "ACCEPT" BUTTON. IF YOU DO NOT AGREE WITH THESE TERMS AND CONDITIONS OR IF YOU ARE NOT AUTHORIZED TO ACCEPT THE TERMS OF THIS LICENSE ON BEHALF OF YOUR EMPLOYER, DECLINE THE LICENSE TERMS AND DO NOT USE THE SOFTWARE OR DOCUMENTATION. THESE TERMS GOVERN YOUR CONTINUED USE OF THE LICENSED MATERIALS THAT YOU DOWNLOAD NOW OR IN THE FUTURE, INCLUDING SUCH ADDITIONAL SOFTWARE MADE AVAILABLE TO YOU THROUGH THE LICENSED MATERIALS YOU HAVE SELECTED, AND ALL UPDATES AND VERSIONS OF SUCH SOFTWARE.

1. **Definitions.**

Accept All | Decline All | Print
< Back | Next > | Finish | Cancel

Installation takes several minutes. During installation, Simplicity Studio offers you viewing and reading options to learn more about the environment.

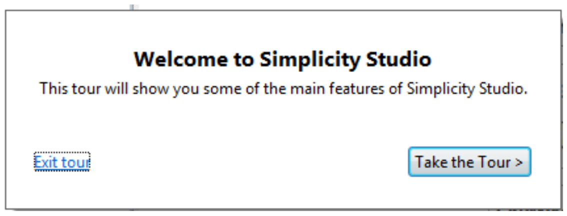

Installation

While you wait, here are some materials to help you get started with Simplicity Studio.

Mastering Simplicity Studio - Understanding hardware and so...
1 Mastering Simplicity Studio
Simplicity Studio Guide
Silicon Labs Community
Maker's Guide to the IoT

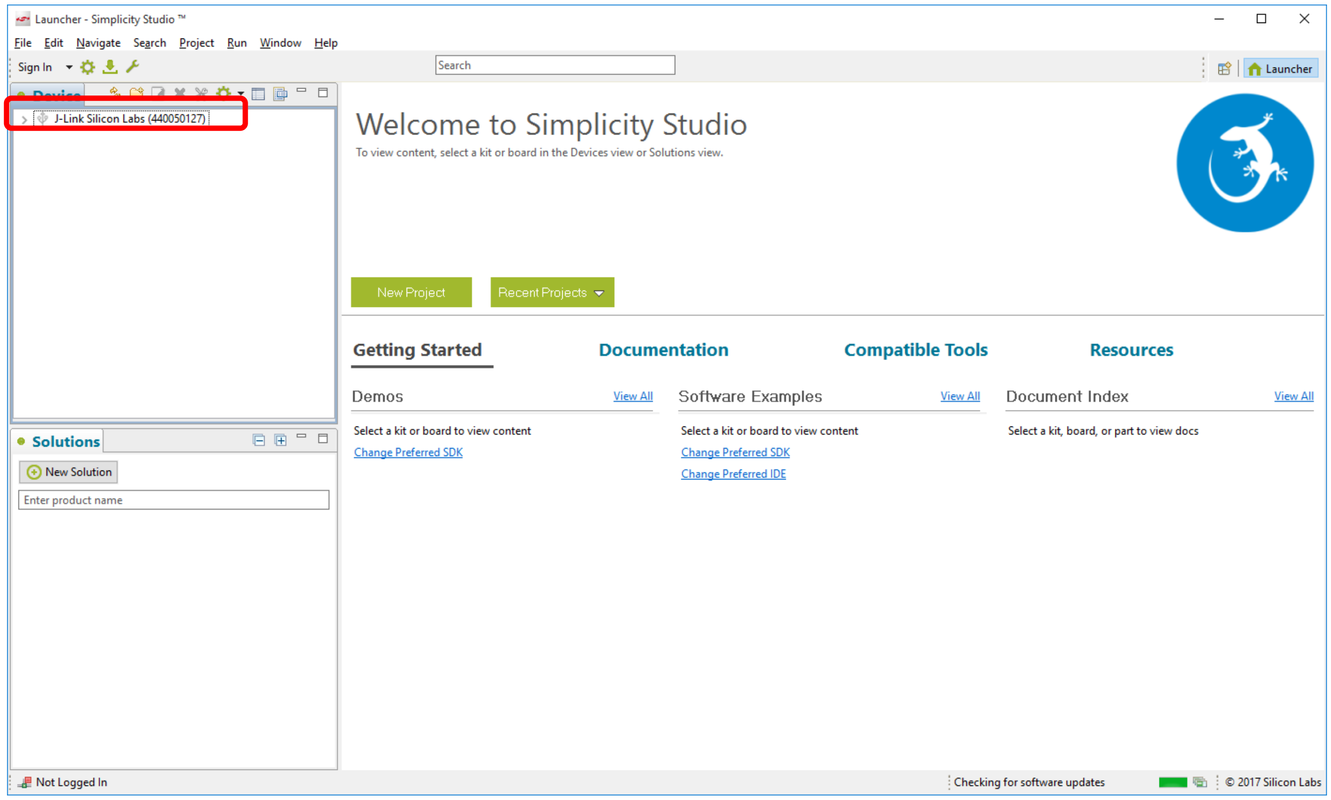
Loading metadata...

12. After installation is complete, restart Simplicity Studio.
13. When Simplicity Studio restarts, you are invited to take a tour. To clear this option now or at any time during or after the tour, click **Exit tour**.

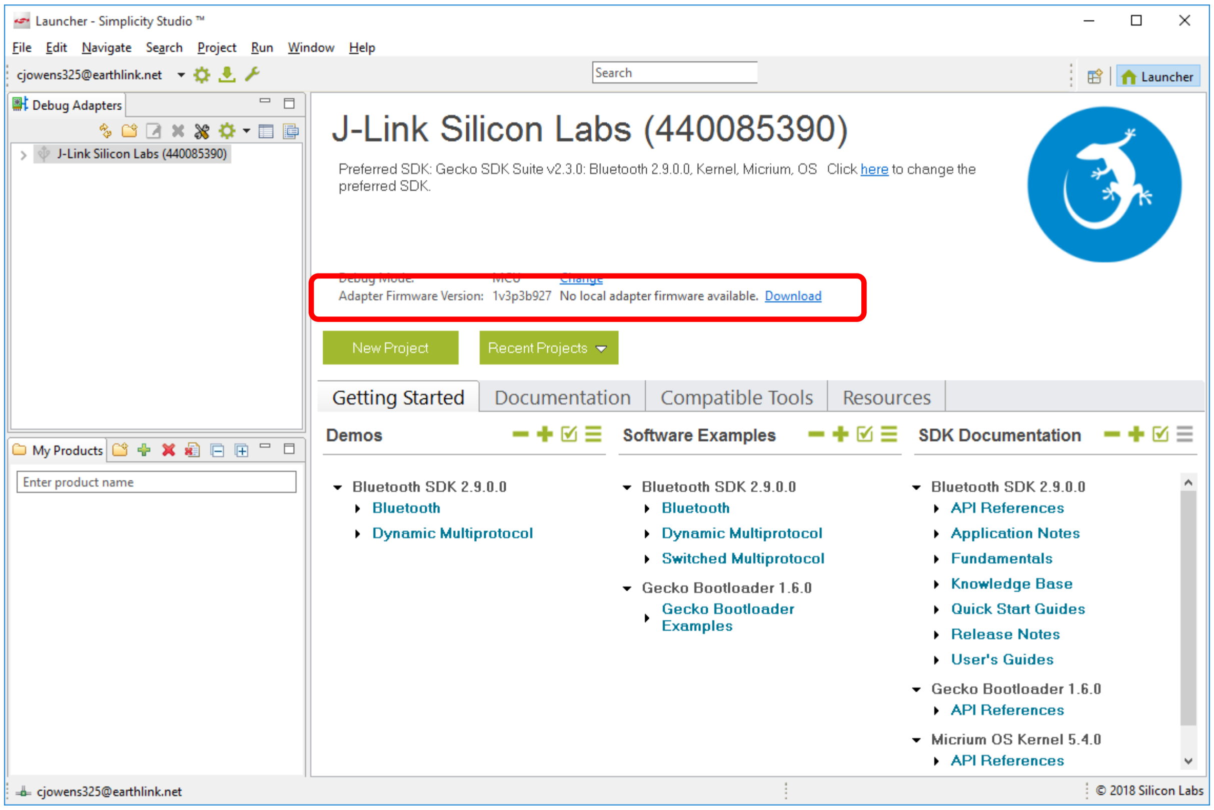

Welcome to Simplicity Studio

This tour will show you some of the main features of Simplicity Studio.

14. The Launcher perspective opens, but it is not yet fully populated. Click the J-link in the devices tab or part in a solution to populate it.


Launcher - Simplicity Studio ™
File Edit Navigate Search Project Run Window Help
Sign In
Search
Launcher
Devices
J-Link Silicon Labs (440050127)
Welcome to Simplicity Studio
To view content, select a kit or board in the Devices view or Solutions view.
New Project
Recent Projects
Getting Started
Documentation
Compatible Tools
Resources
Demos
View All
Software Examples
View All
Document Index
View All
Select a kit or board to view content
Change Preferred SDK
Select a kit or board to view content
Change Preferred SDK
Change Preferred IDE
Select a kit, board, or part to view docs
Solutions
New Solution
Enter product name
Not Logged In
Checking for software updates
© 2017 Silicon Labs

15. The Launcher perspective then is populated with the software components and functionality associated with your hardware and installed SDKs. Before proceeding, if you have a device connected update your device firmware as described in section 3.3 Updating Adapter Firmware.



3 Functionality in the Launcher Perspective

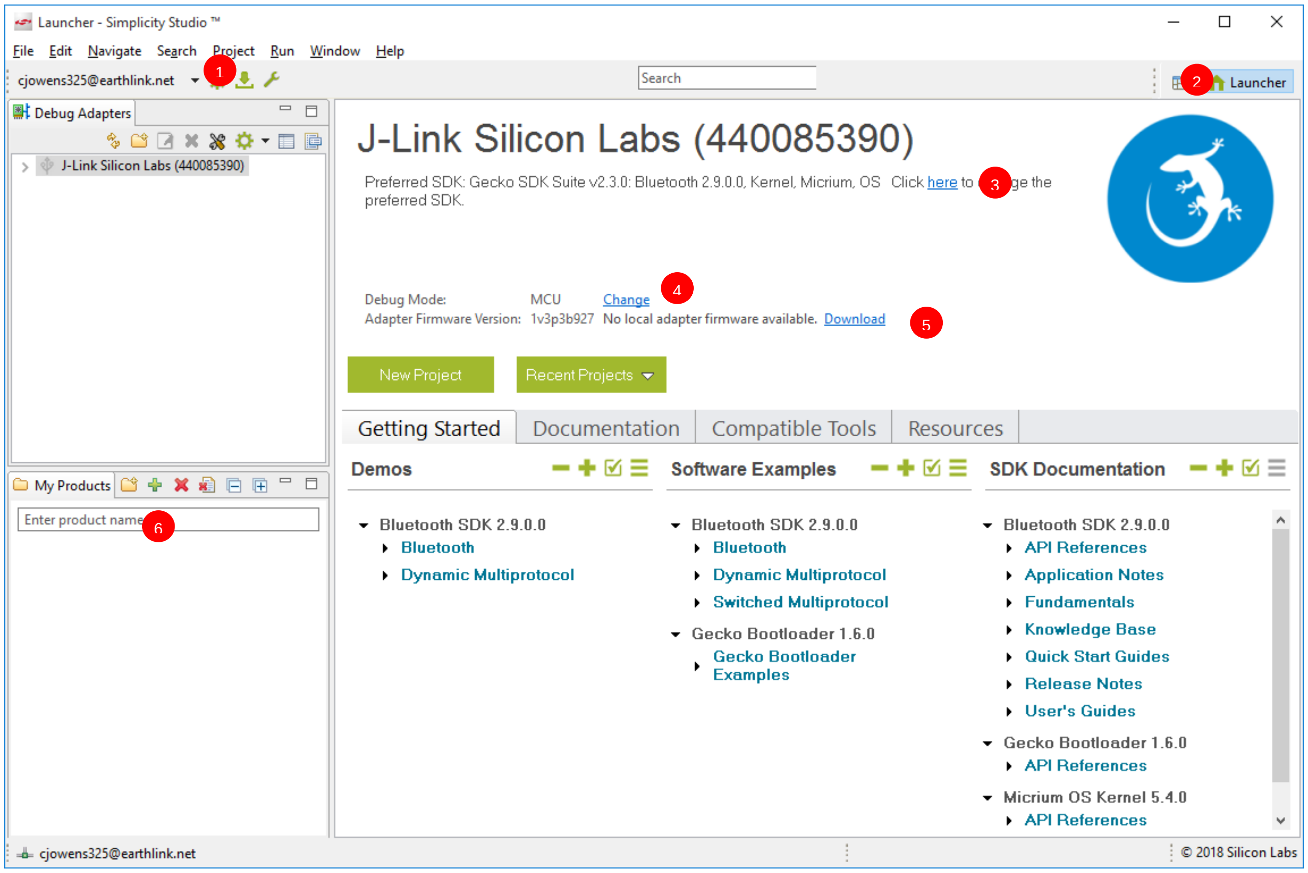
Perspectives are made up of a number of tiles or panes, called views, as well as the content in those views. You can perform a number of functions in the Launcher Perspective, as shown in the following figure. Additional information on some of these is provided later in the section. Note: Your installed version may be different than the version shown in the graphics in this section.

On the toolbar (1) you can:

- Sign in or out
- Open application settings ( )
- Update your software and firmware (, see section 3.1 Downloading Updates or Installing Additional Components for more information)
- Open the Tools menu ( ) to access tools such as Simplicity Commander or Energy Profiler.
- Search for information including entries in the Community forums.
- Change perspectives (2). As you open the Simplicity IDE or other tools, buttons for their perspectives are displayed in the upper right. Use those buttons to easily navigate back to the Launcher perspective or to other perspectives. You can change the layouts of various perspectives by expanding or relocating views, or adding or removing views. To return to the default layout, right-click the perspective button in the upper right and select **Reset**.

In the main view you can:

- Change your preferred SDK (3, see section 3.2 Changing the Preferred SDK for more information).
- Change debug mode (4) (displayed only if a device is connected).
- Update adapter firmware (5, displayed only if a device is connected, see section 3.3 Updating Adapter Firmware for more information).
- Create solutions of multiple parts (6). If you developing for complex networks with a number of different parts involved, you can add them all to the solution and then select the one you are working on from the list. You do not need to have the hardware connected to your computer.
- Access demos, examples, documentation, and other resources from the Getting Started and other tabs (see section 3.4 Accessing Documentation and Other Resources for more information).



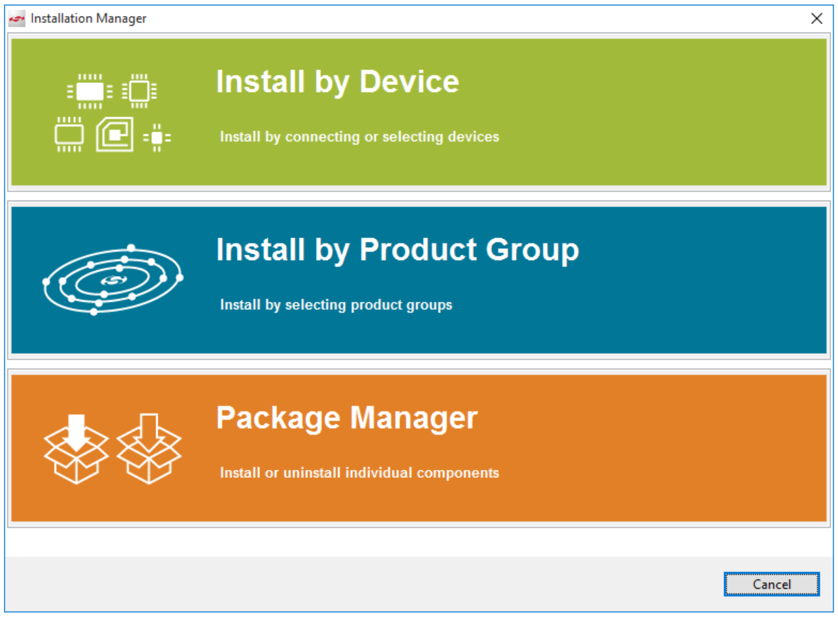
Preferred SDK: Gecko SDK Suite v2.3.0: Bluetooth 2.9.0.0, Kernel, Micrium, OS Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 1v3p3b927 No local adapter firmware available. Download

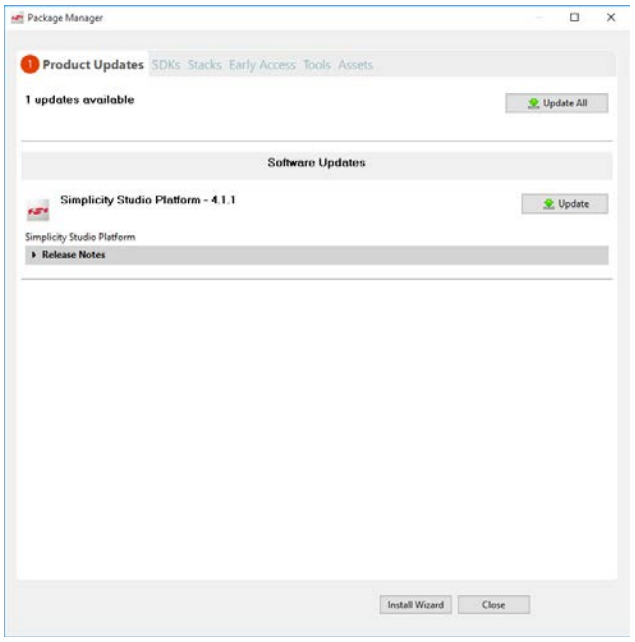
3.1 Downloading Updates or Installing Additional Components

The Download Update icon will be red if updates to installed components are available. If Simplicity Studio detects an available update, and you are in another perspective, you will be notified that an update is available.

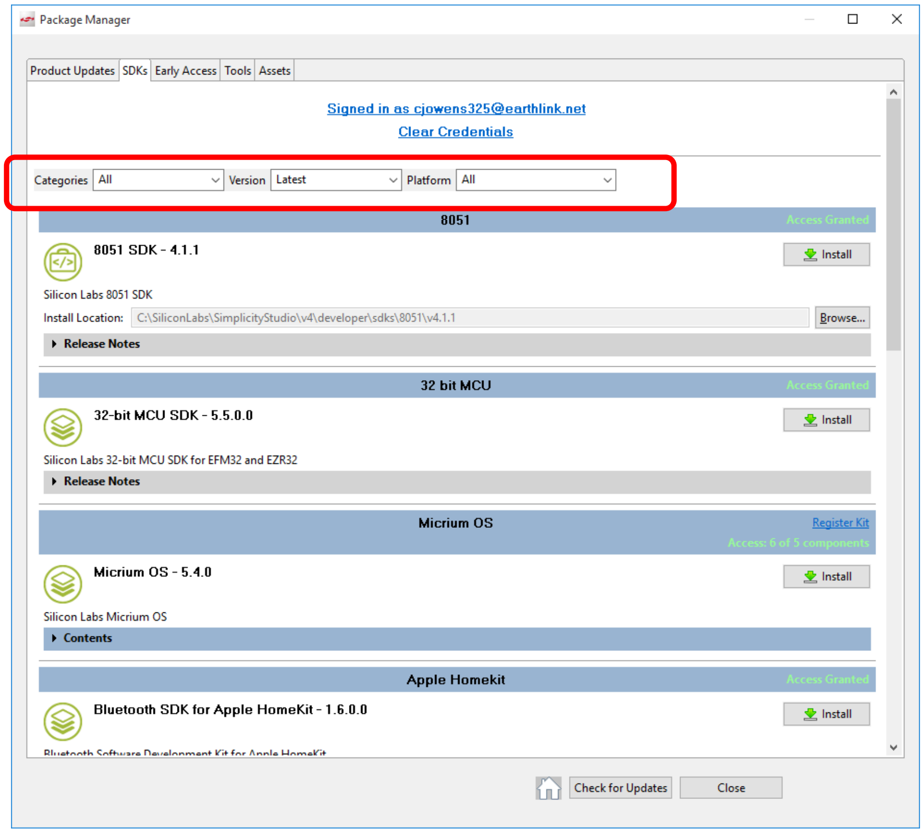
To download a new or updated component, click the Update Software icon. Click Package Manager. Note: If you are installing based on a new device, or want to install a new product group, you can do so through this dialog as well. In subsequent dialogs, click Home (🏠) to return to this dialog. Note that Studio does not show you options, such as the GNU ARM Toolchain, that have already been installed.


Installation Manager
Install by Device
Install by connecting or selecting devices
Install by Product Group
Install by selecting product groups
Package Manager
Install or uninstall individual components
Cancel

Simplicity Studio shows you available updates in the Package Manager dialog. You can update all or select individual updates for installation.



If you want to install a new component, the procedure is the same. Click the Update icon and wait until the Package Manager is displayed. Click the tabs in the Package Manager dialog to see other components available for installation. Use the filters to reduce long lists.



8051 — Access Granted

8051 SDK - 4.1.1

Silicon Labs 8051 SDK

Install Location: C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\8051\v4.1.1

32 bit MCU — Access Granted

32-bit MCU SDK - 5.5.0.0

Silicon Labs 32-bit MCU SDK for EFM32 and EZR32

Micrium OS — Access: 6 of 5 components

Micrium OS - 5.4.0

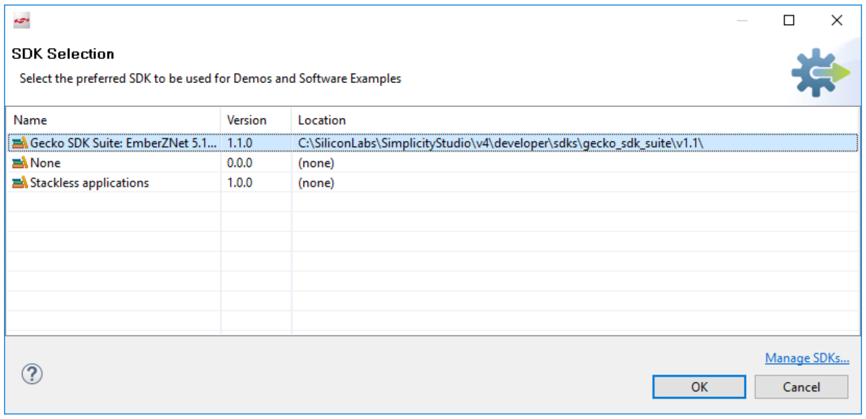
Silicon Labs Micrium OS

Apple Homekit — Access Granted

Bluetooth SDK for Apple HomeKit - 1.6.0.0

3.2 Changing the Preferred SDK

This is a legacy function. In general, most Silicon Labs protocol stack users will have one SDK available to them, the Gecko SDK Suite.



SDK Selection

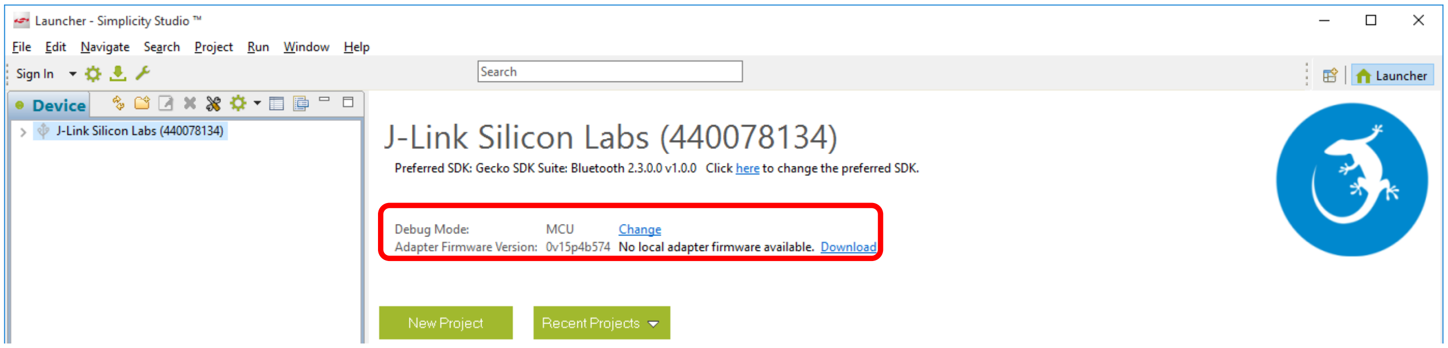
Select the preferred SDK to be used for Demos and Software Examples

Name	Version	Location
Gecko SDK Suite: EmberZNet 5.1...	1.1.0	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v1.1\
None	0.0.0	(none)
Stackless applications	1.0.0	(none)

Within that suite you can have multiple protocols installed. The protocol used in any given instance is controlled either by the example you select, or the stack you select if you go through the ‘New Project’ interface. In general, you should add or remove protocol stacks through the Simplicity Studio update manager. If you need to install a stack or the Gecko SDK Suite outside of the normal installation process, you will receive separate instructions.

3.3 Updating Adapter Firmware

Initially the Launcher perspective may display “No local adapter firmware available.” Click **Download** to download any updates.

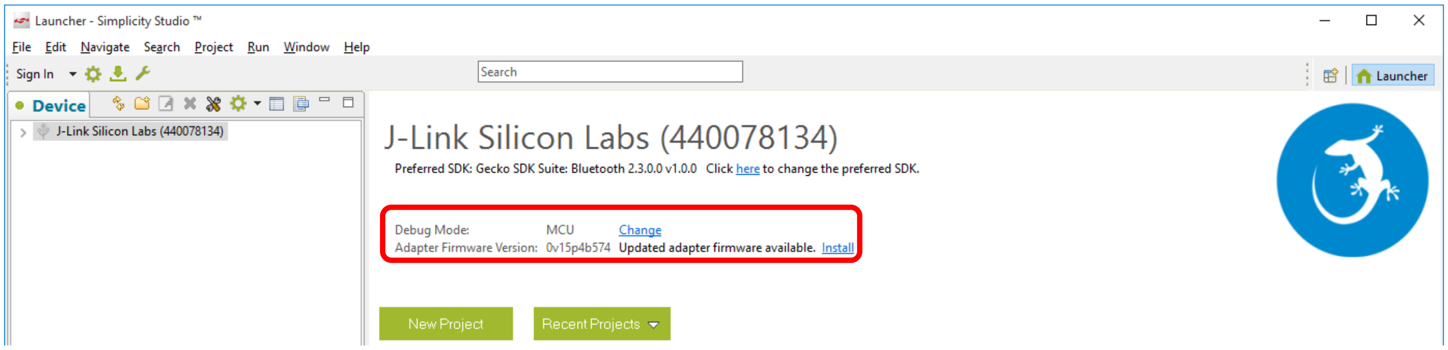

Launcher - Simplicity Studio ™
File Edit Navigate Search Project Run Window Help
Sign In
Search
Launcher
Device
J-Link Silicon Labs (440078134)
New Project
Recent Projects

J-Link Silicon Labs (440078134)

Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 No local adapter firmware available. Download

If an update is available, click **Install** to install the firmware.


Launcher - Simplicity Studio ™
File Edit Navigate Search Project Run Window Help
Sign In
Search
Launcher
Device
J-Link Silicon Labs (440078134)
New Project
Recent Projects

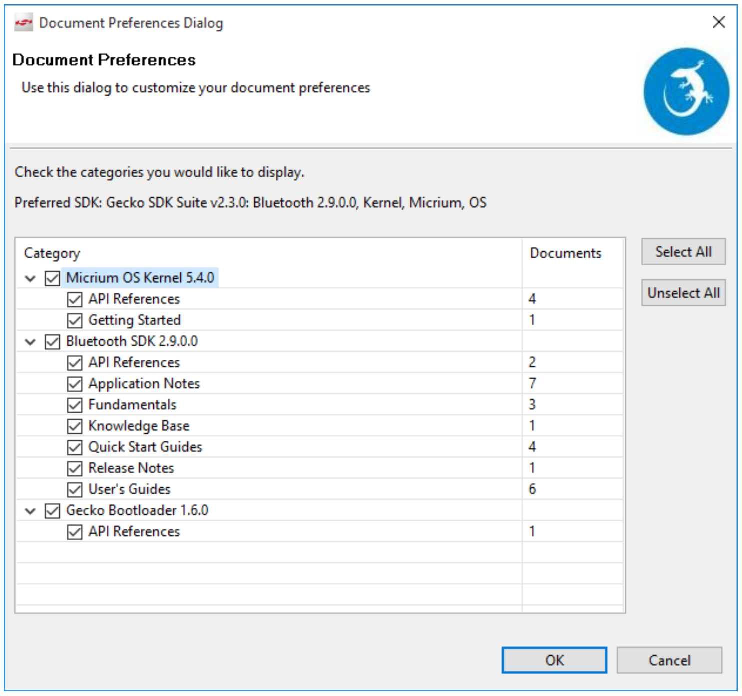
J-Link Silicon Labs (440078134)

Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 Updated adapter firmware available. Install

3.4 Accessing Documentation and Other Resources

The **Getting Started** tab provides access to demos, example applications, and stack related documentation. To show/hide specific categories, click ✓. Select or deselect categories, then click **OK**.



Document Preferences

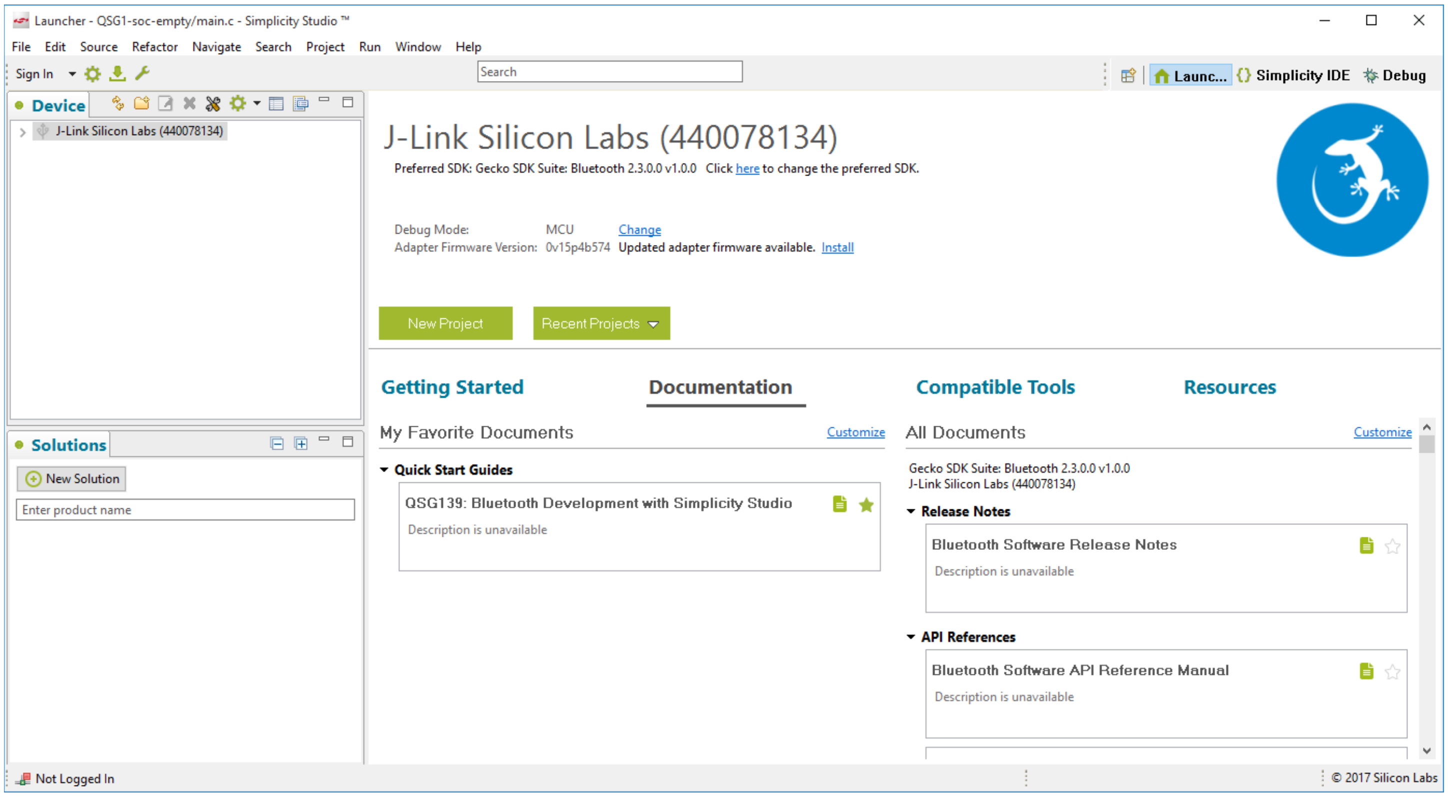
Use this dialog to customize your document preferences

Check the categories you would like to display.

Preferred SDK: Gecko SDK Suite v2.3.0: Bluetooth 2.9.0.0, Kernel, Micrium, OS

Category	Documents
☑ Micrium OS Kernel 5.4.0	
☑ API References	4
☑ Getting Started	1
☑ Bluetooth SDK 2.9.0.0	
☑ API References	2
☑ Application Notes	7
☑ Fundamentals	3
☑ Knowledge Base	1
☑ Quick Start Guides	4
☑ Release Notes	1
☑ User's Guides	6
☑ Gecko Bootloader 1.6.0	
☑ API References	1

The **Documentation** tab lists documentation about the stack and about the hardware on the right, and documents you selected as favorites on the left. By default, only Application Notes are displayed. Click **Customize** to change the categories and the order in which they are displayed. Click the star icon on any document to show it in the My Favorite Documents list. You can customize this list's categories and category order as well.



J-Link Silicon Labs (440078134)

Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 Updated adapter firmware available. Install

My Favorite Documents

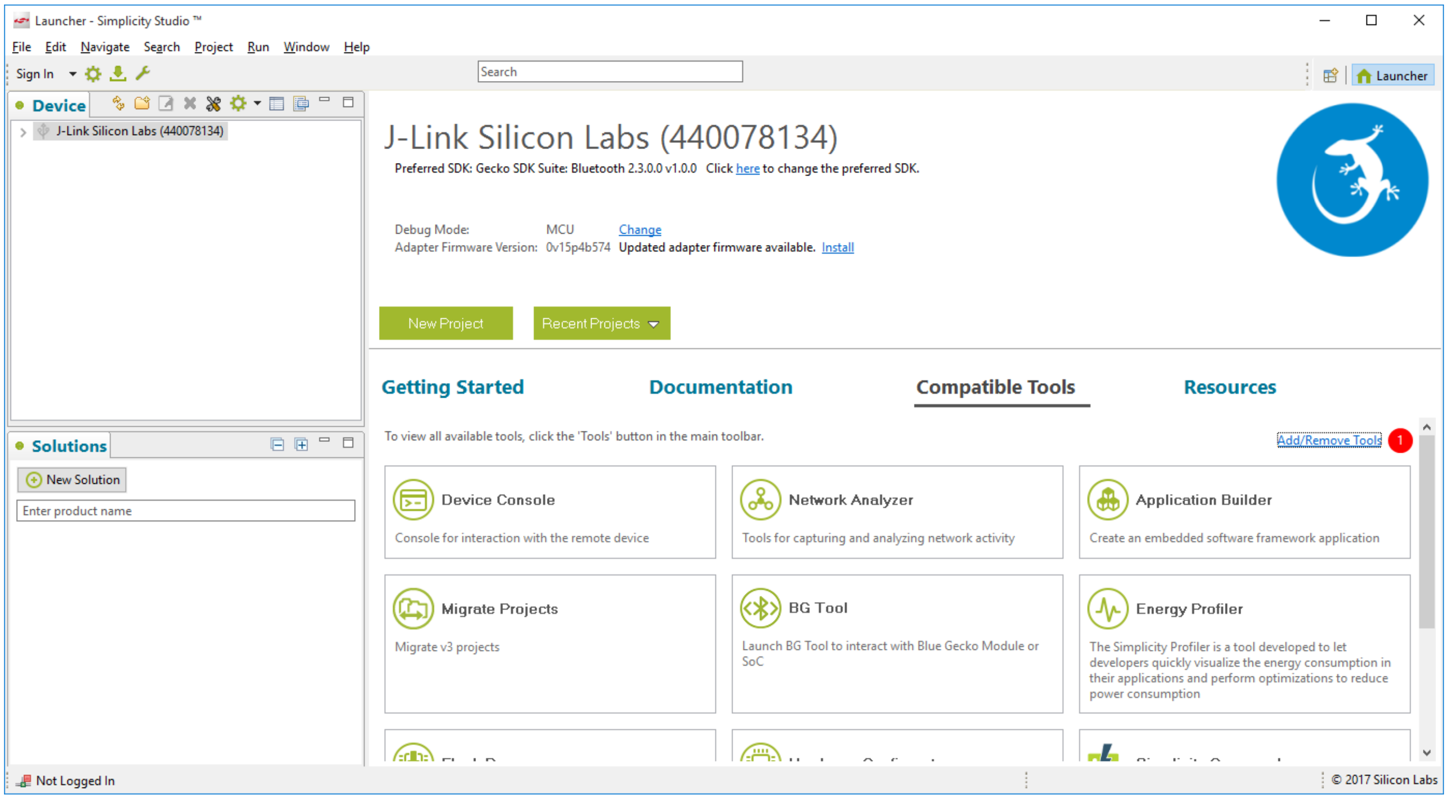
- Quick Start Guides
  - QSG139: Bluetooth Development with Simplicity Studio — Description is unavailable

All Documents

Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0
J-Link Silicon Labs (440078134)

- Release Notes
  - Bluetooth Software Release Notes — Description is unavailable
- API References
  - Bluetooth Software API Reference Manual — Description is unavailable

The **Compatible Tools** tab is an alternative way to access the tools available through the Tools dropdown.



J-Link Silicon Labs (440078134)

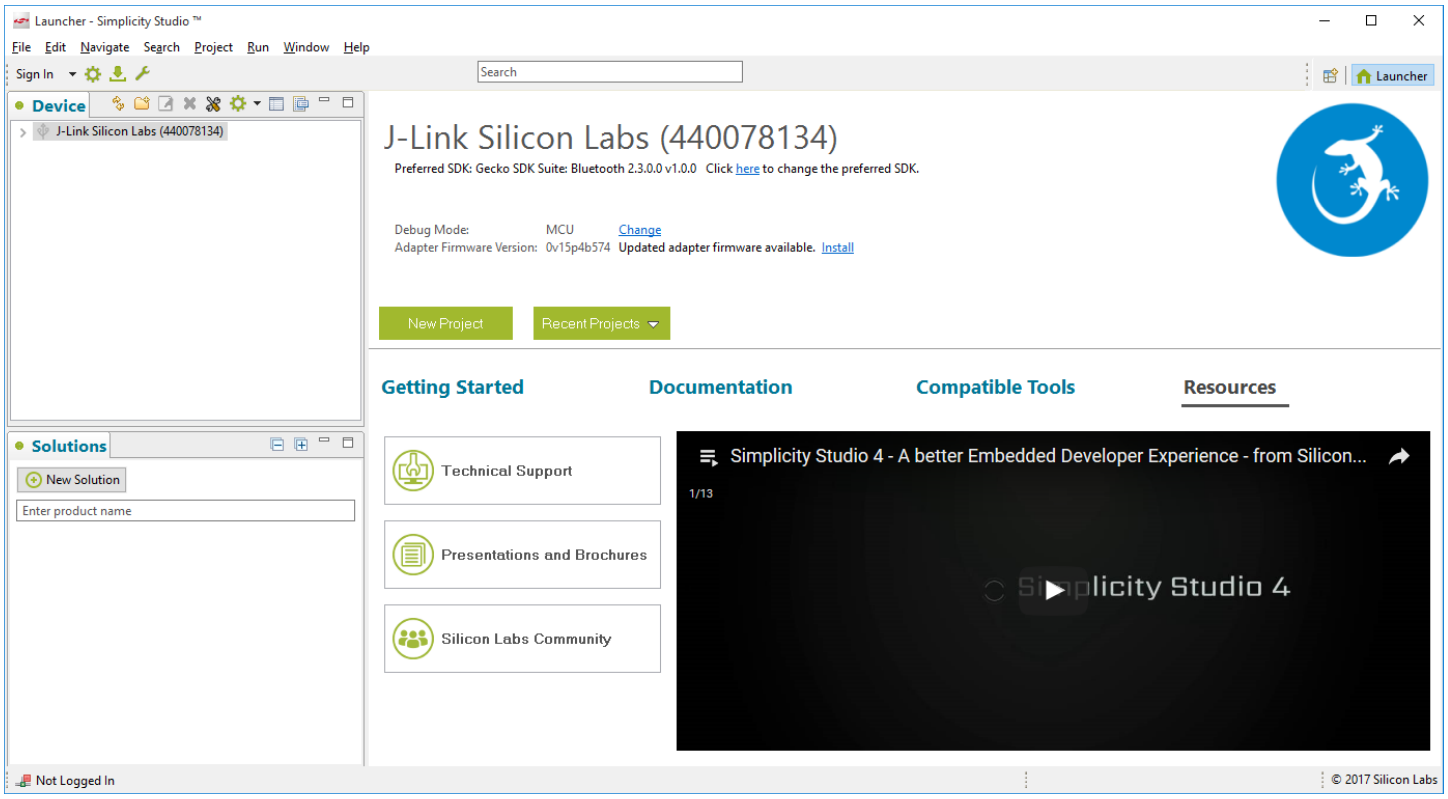
Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 Updated adapter firmware available. Install

To view all available tools, click the 'Tools' button in the main toolbar.

- Device Console — Console for interaction with the remote device
- Network Analyzer — Tools for capturing and analyzing network activity
- Application Builder — Create an embedded software framework application
- Migrate Projects — Migrate v3 projects
- BG Tool — Launch BG Tool to interact with Blue Gecko Module or SoC
- Energy Profiler — The Simplicity Profiler is a tool developed to let developers quickly visualize the energy consumption in their applications and perform optimizations to reduce power consumption

The **Resources** tab provides access to support, marketing collateral, and the Silicon Labs community.



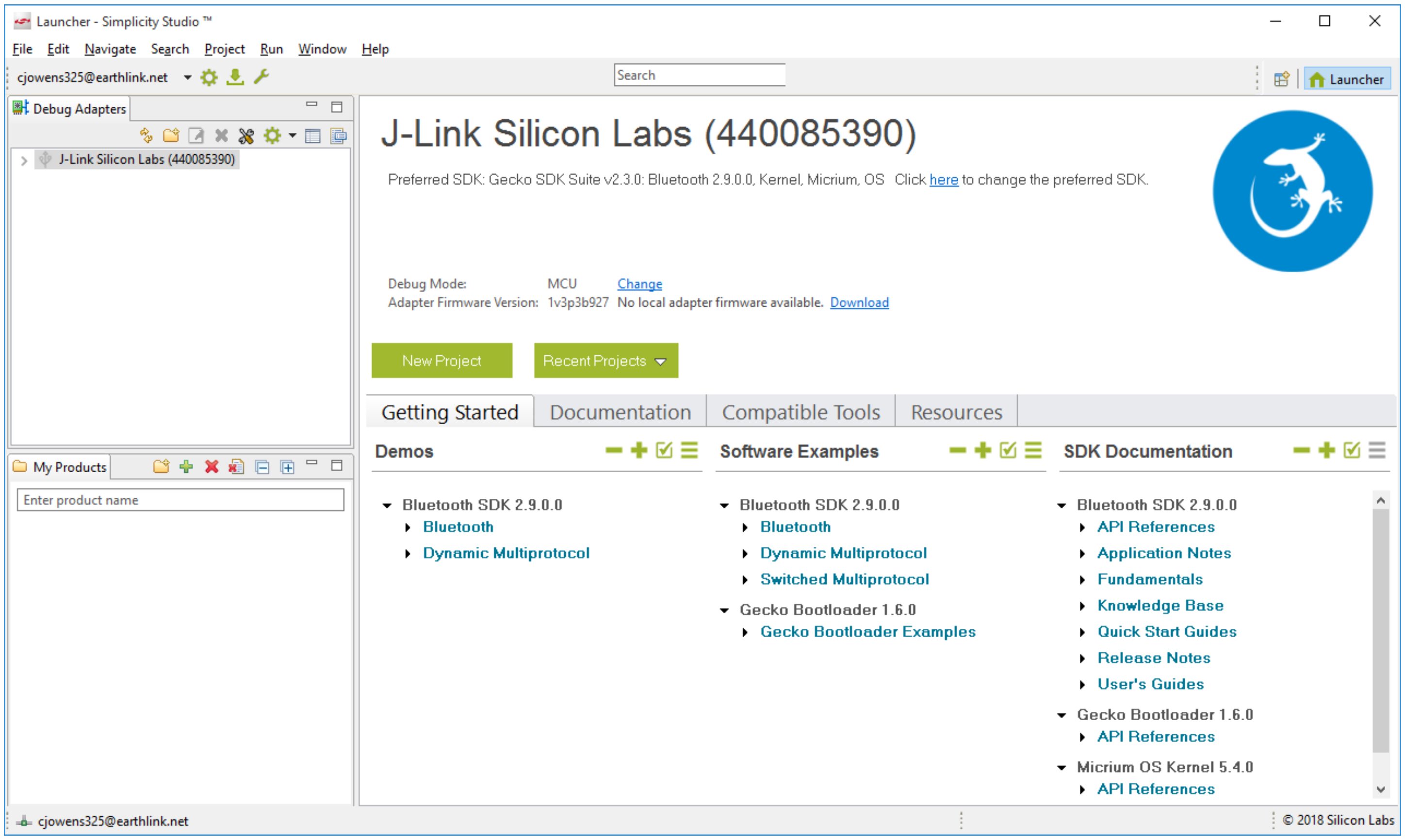
J-Link Silicon Labs (440078134)

Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 Updated adapter firmware available. Install

4 About Demos and Examples

Because starting application development from scratch is difficult, the Bluetooth SDK comes with a number of built-in demos and examples covering the most frequent use cases, as shown in the following figure. Demos are pre-built application images that you can run immediately. Software examples can be modified before building the application image. By starting with an example, you need only extend the code with your application code, and rewrite only what must be customized for your needs. The demos and examples you see are determined by the part selected. If you are using a custom solution with more than one part, be sure to click on the part you are working with to see only those items applicable to that part.



J-Link Silicon Labs (440085390)

Preferred SDK: Gecko SDK Suite v2.3.0: Bluetooth 2.9.0.0, Kernel, Micrium, OS Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 1v3p3b927 No local adapter firmware available. Download

Demos
- Bluetooth SDK 2.9.0.0
  - Bluetooth
  - Dynamic Multiprotocol

Software Examples
- Bluetooth SDK 2.9.0.0
  - Bluetooth
  - Dynamic Multiprotocol
  - Switched Multiprotocol
- Gecko Bootloader 1.6.0
  - Gecko Bootloader Examples

SDK Documentation
- Bluetooth SDK 2.9.0.0
  - API References
  - Application Notes
  - Fundamentals
  - Knowledge Base
  - Quick Start Guides
  - Release Notes
  - User's Guides
- Gecko Bootloader 1.6.0
  - API References
- Micrium OS Kernel 5.4.0
  - API References

4.1 Demos

Demos are prebuilt application examples that can be directly downloaded to your device. They are grouped by category. To see demos, you must select the board or JLink, not the chip.

Bluetooth

NCP target – empty: Use if you want your device to work in NCP (network co-processor) mode, so you can control it from another device or directly from your PC with BG Tool. NCP demo + BG Tool together provide an efficient tool to debug your application by issuing commands to the stack step-by-step.

SOC – Empty: Use to check connectivity with your phone. You can find the device among the discoverable devices and connect to it.

SOC – Smart Phone App: Use to see a variety of demo applications that you can control and verify with the Blue Gecko smartphone app. For more details visit: http://www.silabs.com/bluegeckoapp.

SOC - Thunderboard Sense, - Thunderboard Sense 2, and - Thunderboard React: Use to show the applicable Thunderboard features and an application compatible with the Thunderboard mobile app.

Dynamic Multiprotocol

- **SOC - Empty - RAIL - DMP:** A minimal project structure, used as a starting point for custom Dynamic Multiprotocol applications. The project has the basic functionality enabling peripheral connectivity, without GATT services. It runs on top of Micrium OS RTOS and multiprotocol RAIL.

- **SOC - Light - RAIL - DMP -** Use as part of the RAIL/Bluetooth Dynamic Multiprotocol demonstration, along with the Switch demo in the Flex SDK, and the Wireless Gecko Smartphone app. See *QSG155: Using the Silicon Labs Dynamic Multiprotocol Demonstrations* for information on demonstrating this functionality.

To download and run a demo on your device, click the demo. In the Mode drop-down in the next dialog, select Run. Click **Start**.

4.2 Software Examples

Software examples provide starting points for your application development. Silicon Labs recommends that you start with an example and modify it according to your needs instead of starting with a blank project. The examples provide default configurations needed by the stack and a basic application structure that you can build on. Note that the 'Empty' examples are not blank, but rather provide a minimal starting structure.

The software examples with the same names as the demos provide the demo functionality, but they can be modified before downloading to the device.

To import software example code into your workspace as a new project using default project configurations, click the name of the example to open the Simplicity Studio IDE. To build the example project click Debug ( ) in the upper left corner of the Simplicity IDE perspective. The Debug perspective opens. Click Play ( ) to start running you project on the device. To create a project with different configurations, click **New Project** (above the Demos column in the Launcher perspective), and proceed as described in section 5 Getting Started with Application Development.

Examples provided for the EFR32xG12 and newer parts include the following:

- **Silicon Labs Gecko Bootloader examples** (see *UG266: Silicon Labs Gecko Bootloader User Guide* and *AN1086: Using the Gecko Bootloader with Silicon Labs Bluetooth Applications*
- **Bluetooth Examples**
  - **NCP target – empty**: Network co-processor target application with no GATT services included.
  - **SOC – DTM**: Use to run Bluetooth DTM (direct test mode) tests.
  - **SOC – Empty**: A minimal project structure, used as a starting point for custom applications. The project has the basic functionality enabling peripheral connectivity, without GATT service
  - **SOC – Smart Phone App**: For use with the Silicon Labs Smart Phone app (for more details visit: http://www.silabs.com/blue-geckoapp). Demonstrates the Health Thermometer Profile, Proximity Profile, and implements an iBeacon service that sends non-connectible advertisements in iBeacon format.
  - **SOC – Thermometer**: Implements the Thermometer (GATT Server) Role of the Health Thermometer Profile, which enables a Collector device to connect and interact with a Thermometer. The device acts as a connection peripheral.

    Note: Some radio boards will exhibit random pixels in in the display when this example is running because they have a shared pin for sensor and display enable signals.
  - **SOC – Thermometer - RTOS**: The SOC - Thermometer example based on Micrium RTOS.
  - **SOC – iBeacon**: Use to broadcast a beacon signal (in iBeacon format). Beacons are used to notify Bluetooth-enabled devices that they entered an area, or they can be used for distance measurement and positioning.
  - **SOC - Thunderboard React, - Thunderboard Sense, and - Thunderboard Sense 2**: Examples used to show the applicable Thunderboard features.
  - **SOC - Voice over Bluetooth Low Energy**
- **Dynamic Multiprotocol Examples** (see *AN1134: Dynamic Multiprotocol Development with Bluetooth and Proprietary Protocols on RAIL* for more information)
  - **SOC - Empty - RAIL - DMP:** A minimal project structure, used as a starting point for custom Dynamic Multiprotocol applications. The project has the basic functionality enabling peripheral connectivity, without GATT services. It runs on top of Micrium OS RTOS and multiprotocol RAIL.
  - **SOC - Light - RAIL - DMP -** Implements the Light (GATT Server) Role, which enables a Switch device to connect to and interact with it. The device acts as a connection Peripheral. This is a Dynamic Multiprotocol reference application, running on top of Micrium OS RTOS and multiprotocol RAIL.
  - **SOC -** Range Test - RAIL - DMP - RAIL: Range Test with Bluetooth connectivity. It runs on top of Micrium OS RTOS and multi-protocol RAIL.
- **Switched Multiprotocol Examples** (see *UG267: Switched Multiprotocol User Guide*)
  - **NCP Target - Switched Multiprotocol Trust Center**: Target application with Switched Multiprotocol GATT services included.
  - **SOC - Switched Multiprotocol Joining Device**: Switched multiprotocol joining device application.

5 Getting Started with Application Development

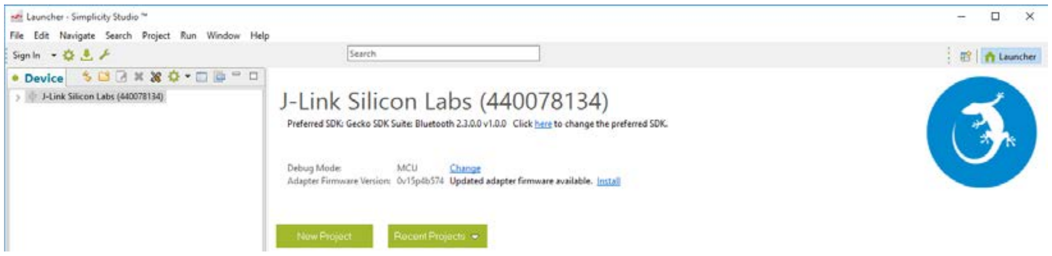
Developing a Bluetooth application consists of two main steps: defining the GATT database structure, and defining the event handlers for events such as `connection_opened`, `connection_closed`, and so on.

The most common starting point for application development is the **SOC – Empty** example. This project contains a simple GATT database (including the Generic Access service, Device Information service, and OTA service) and a while loop that handles some events raised by the stack. You can extend both the GATT database and the event handlers of this example according to your needs.

Note: Beginning with Bluetooth SDK version 2.7.0.0, all devices must be loaded with the Gecko Bootloader as well as the application. While you are getting started, the easiest way to do this is to load any of the precompiled demo images, which come with the bootloader configured as part of the image. When you flash your application it overwrites the demo application, but the bootloader remains. Subsequently you may wish to build your own bootloader, as described in *UG266: Silicon Labs Gecko Bootloader User Guide*. The first bootloader loaded on a clean device should always be the combined bootloader.

To start developing your application, follow these steps (illustrated in the following figure). **Note:** Your SDK version may be later than the version shown in the procedure illustrations. :

1. Click **New Project** in the Launcher perspective.
2. Select Bluetooth SDK and click **Next**.
3. Select **SoC – Empty** and click **Next**.
4. Name your project and click **Next**.
5. Verify that your selected part is shown, and select your preferred toolchain. Note: You should only select one toolchain. If you are using GCC, you must uncheck IAR. Otherwise the system will revert to IAR when you generate your files. Click **Finish**.

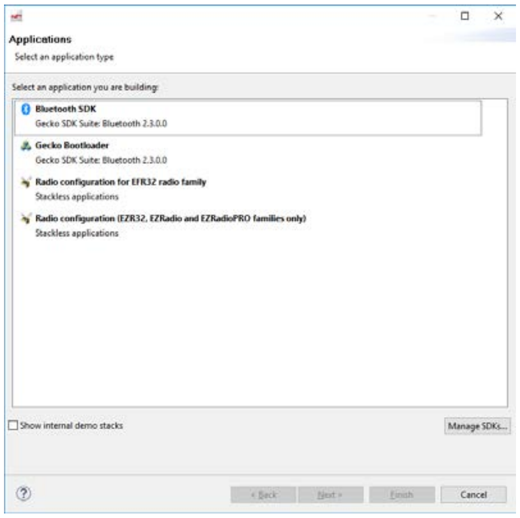

Launcher - Simplicity Studio ™
File Edit Navigate Search Project Run Window Help
Sign In
Search
Launcher
Device
New Project
Recent Projects

J-Link Silicon Labs (440078134)

J-Link Silicon Labs (440078134)

Preferred SDK: Gecko SDK Suite: Bluetooth 2.3.0.0 v1.0.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 0v15p4b574 Updated adapter firmware available. Install



Applications
Select an application type

Select an application you are building:

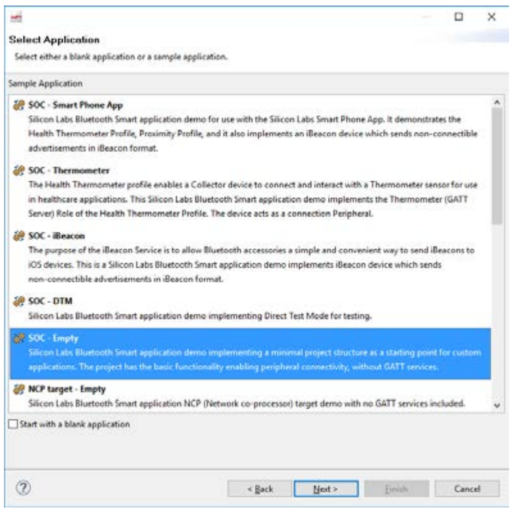
Bluetooth SDK
Gecko SDK Suite: Bluetooth 2.3.0.0

Gecko Bootloader
Gecko SDK Suite: Bluetooth 2.3.0.0

Radio configuration for EFR32 radio family
Stackless applications

Radio configuration (EZR32, EZRadio and EZRadioPRO families only)
Stackless applications

☐ Show internal demo stacks



Select Application
Select either a blank application or a sample application.

Sample Application

SOC - Smart Phone App
Silicon Labs Bluetooth Smart application demo for use with the Silicon Labs Smart Phone App. It demonstrates the Health Thermometer Profile, Proximity Profile, and it also implements an iBeacon device which sends non-connectible advertisements in iBeacon format.

SOC - Thermometer
The Health Thermometer profile enables a Collector device to connect and interact with a Thermometer sensor for use in healthcare applications. This Silicon Labs Bluetooth Smart application demo implements the Thermometer (GATT Server) Role of the Health Thermometer Profile. The device acts as a connection Peripheral.

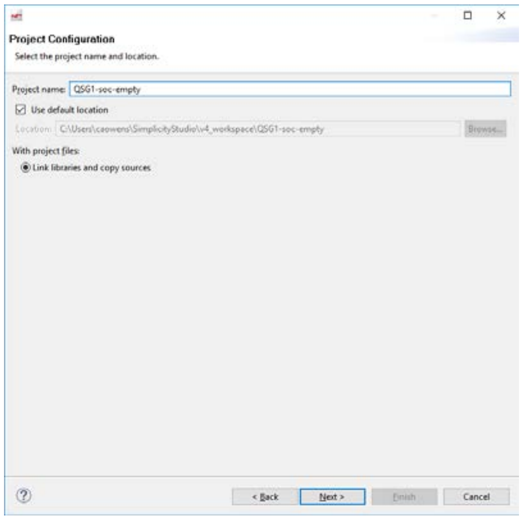
SOC - iBeacon
The purpose of the iBeacon Service is to allow Bluetooth accessories a simple and convenient way to send iBeacons to iOS devices. This is a Silicon Labs Bluetooth Smart application demo implements iBeacon device which sends non-connectible advertisements in iBeacon format.

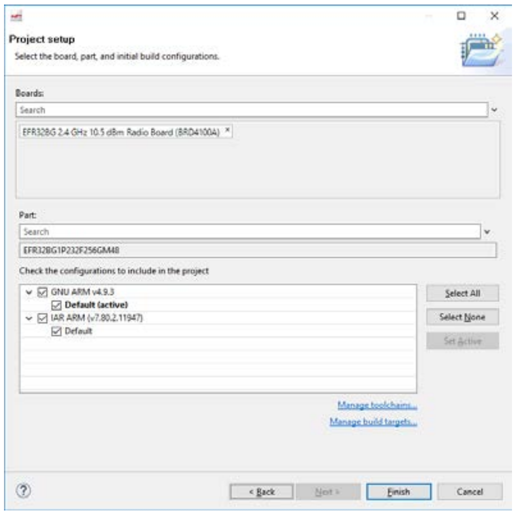
SOC - DTM
Silicon Labs Bluetooth Smart application demo implementing Direct Test Mode for testing.

SOC - Empty
Silicon Labs Bluetooth Smart application demo implementing a minimal project structure as a starting point for custom applications. The project has the basic functionality enabling peripheral connectivity, without GATT services.

NCP target - Empty
Silicon Labs Bluetooth Smart application NCP (Network co-processor) target demo with no GATT services included.

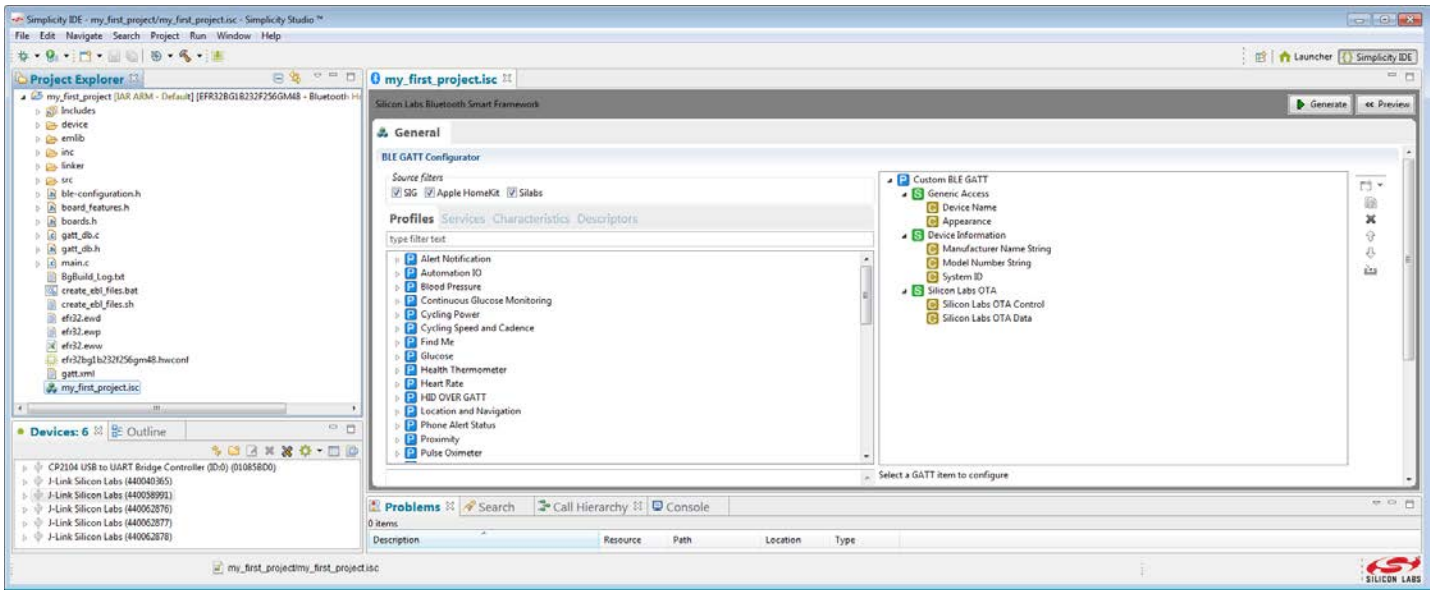
☐ Start with a blank application


Project Configuration
Select the project name and location.
Project name: QSG1-soc-empty
☑ Use default location
Location: C:\Users\caowens\SimplicityStudio\v4_workspace\QSG1-soc-empty
With project files:
● Link libraries and copy sources


Project setup
Select the board, part, and initial build configurations.
Boards: EFR32BG 2.4 GHz 10.5 dBm Radio Board (BRD4100A)
Part: EFR32BG1P232F256GM48
Check the configurations to include in the project
☑ GNU ARM v4.9.3
☑ Default (active)
☑ IAR ARM (v7.80.2.11947)
☑ Default

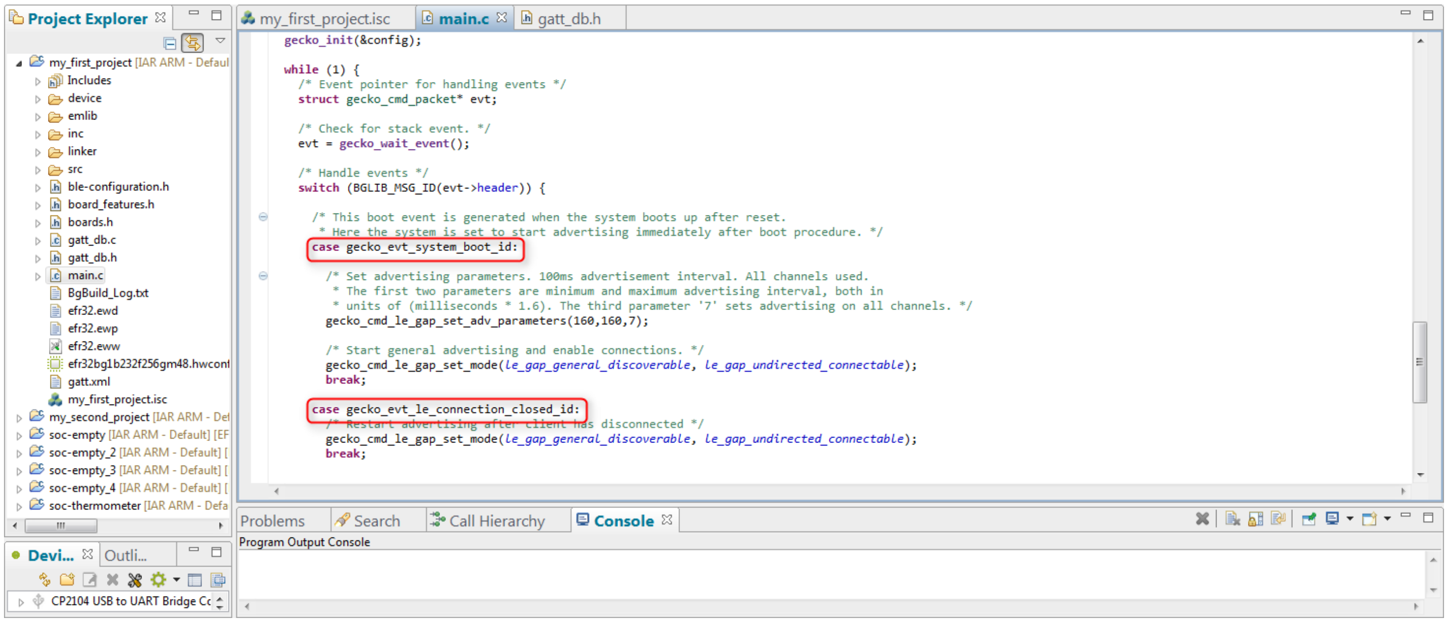
A visual GATT editor automatically appears after creating the project, to help you create your own GATT database with a few clicks. Note that a Simplicity IDE perspective button is now included in the upper right of the screen.

You can create your own database at this point, or return to it later by clicking the .isc file in the Project Explorer pane on the left. For more information, see section 6 The GATT Editor.


Simplicity IDE - my_first_project/my_first_project.isc - Simplicity Studio ™

A reference for each characteristic is generated and defined in gatt_db.h. You can use this references in your code to read / write the values of the characteristics in the GATT database with `gecko_cmd_gatt_server_read_attribute_value()` / `gecko_cmd_gatt_server_write_attribute_value()` commands.

Open main.c by double-clicking it in Project Explorer. You will find the event handlers in the main loop. You can extend this list with further event handlers. The full list of events – and stack commands – can be found in the *Bluetooth Software API Reference Manual*.



```
  gecko_init(&config);

  while (1) {
    /* Event pointer for handling events */
    struct gecko_cmd_packet* evt;

    /* Check for stack event. */
    evt = gecko_wait_event();

    /* Handle events */
    switch (BGLIB_MSG_ID(evt->header)) {

      /* This boot event is generated when the system boots up after reset.
       * Here the system is set to start advertising immediately after boot procedure. */
      case gecko_evt_system_boot_id:

        /* Set advertising parameters. 100ms advertisement interval. All channels used.
         * The first two parameters are minimum and maximum advertising interval, both in
         * units of (milliseconds * 1.6). The third parameter '7' sets advertising on all channels. */
        gecko_cmd_le_gap_set_adv_parameters(160,160,7);

        /* Start general advertising and enable connections. */
        gecko_cmd_le_gap_set_mode(le_gap_general_discoverable, le_gap_undirected_connectable);
        break;

      case gecko_evt_le_connection_closed_id:
        /* Restart advertising after client has disconnected */
        gecko_cmd_le_gap_set_mode(le_gap_general_discoverable, le_gap_undirected_connectable);
        break;
```

To build and debug your project click Debug ( ) in the upper left corner of the Simplicity IDE perspective. It will build and download your project, and open up the Debug perspective. Click Play ( ) to start running you project on the device.

5.1 Enabling Field Updates

Deploying new firmware for devices in the field can be done by UART DFU (Device Firmware Update) or, for SoC applications, OTA DFU. For more information on each of these methods refer to *AN1086 Using the Gecko Bootloader with the Silicon Labs Bluetooth Applications*.

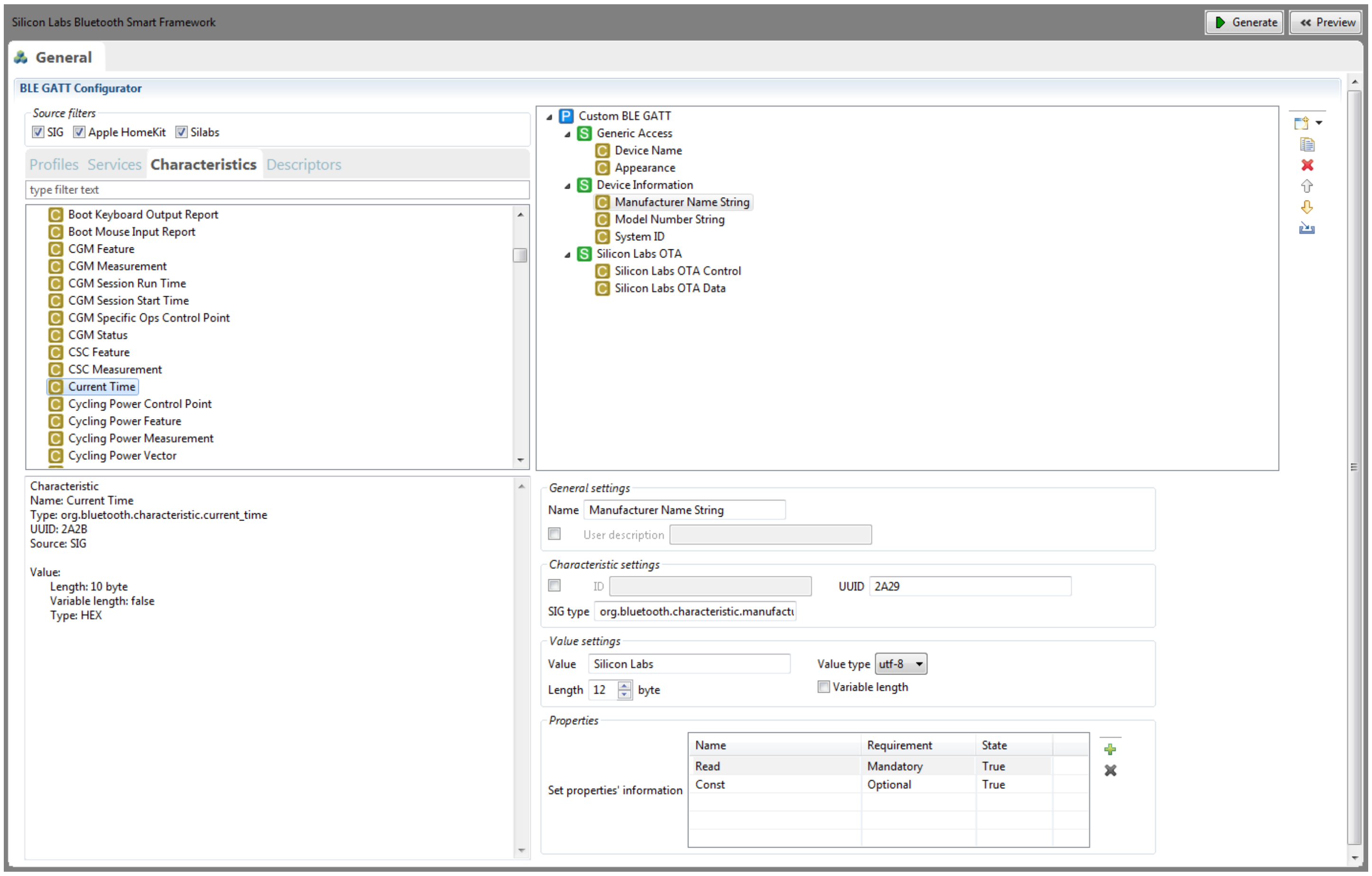
6 The GATT Editor

Every Bluetooth connection has a GATT client and a GATT server. The server holds a GATT database: a collection of *Characteristic*s that can be read and written by the client. The Characteristics are grouped into *Services*, and the group of Services determines a *Bluetooth Profile*.

If you are implementing a GATT server (typically on the peripheral device), you have to define a GATT database structure. This structure cannot be modified during runtime, so it has to be designed in advance. Clients (typically the central device) can also have a GATT database, even if no device will query it, so you can keep the default database structure in your code.

The GATT editor is a simple-to-use tool to help you build your own GATT database. A list of predefined Profiles/Services/Characteristics/Descriptors is shown in a pane in the upper left and your current GATT database structure is shown in a pane in the upper right. An options menu is provided to the right of the Database pane.

Click an item in the Database pane to see and modify its settings in a pane in the lower right. To add a Profile/Service/Characteristic/Descriptor to your database, simply drag and drop it from the list to your database.



Boot Keyboard Output Report
Boot Mouse Input Report
CGM Feature
CGM Measurement
CGM Session Run Time
CGM Session Start Time
CGM Specific Ops Control Point
CGM Status
CSC Feature
CSC Measurement
Current Time
Cycling Power Control Point
Cycling Power Feature
Cycling Power Measurement
Cycling Power Vector

Custom BLE GATT
- Generic Access
  - Device Name
  - Appearance
- Device Information
  - Manufacturer Name String
  - Model Number String
  - System ID
- Silicon Labs OTA
  - Silicon Labs OTA Control
  - Silicon Labs OTA Data

Characteristic
Name: Current Time
Type: org.bluetooth.characteristic.current_time
UUID: 2A2B
Source: SIG

Value:
Length: 10 byte
Variable length: false
Type: HEX

General settings
Name: Manufacturer Name String
User description

Characteristic settings
ID
UUID: 2A29
SIG type: org.bluetooth.characteristic.manufactu

Value settings
Value: Silicon Labs
Value type: utf-8
Length: 12 byte
Variable length

Properties
Set properties' information

Name	Requirement	State
Read	Mandatory	True
Const	Optional	True

To get more information about a Profile/Service/Characteristic/Descriptor, click it either in the list or in your database. The description is displayed in the lower-left pane. You can find a detailed description of any Profile/Service/Characteristic/Descriptor on https://www.bluetooth.com/specifications/gatt.

Characteristics are generally complex structures of fields. The GATT editor currently does not list the fields within a characteristic. If you want to know what fields a characteristic has, visit https://www.bluetooth.com/specifications/gatt/characteristics.

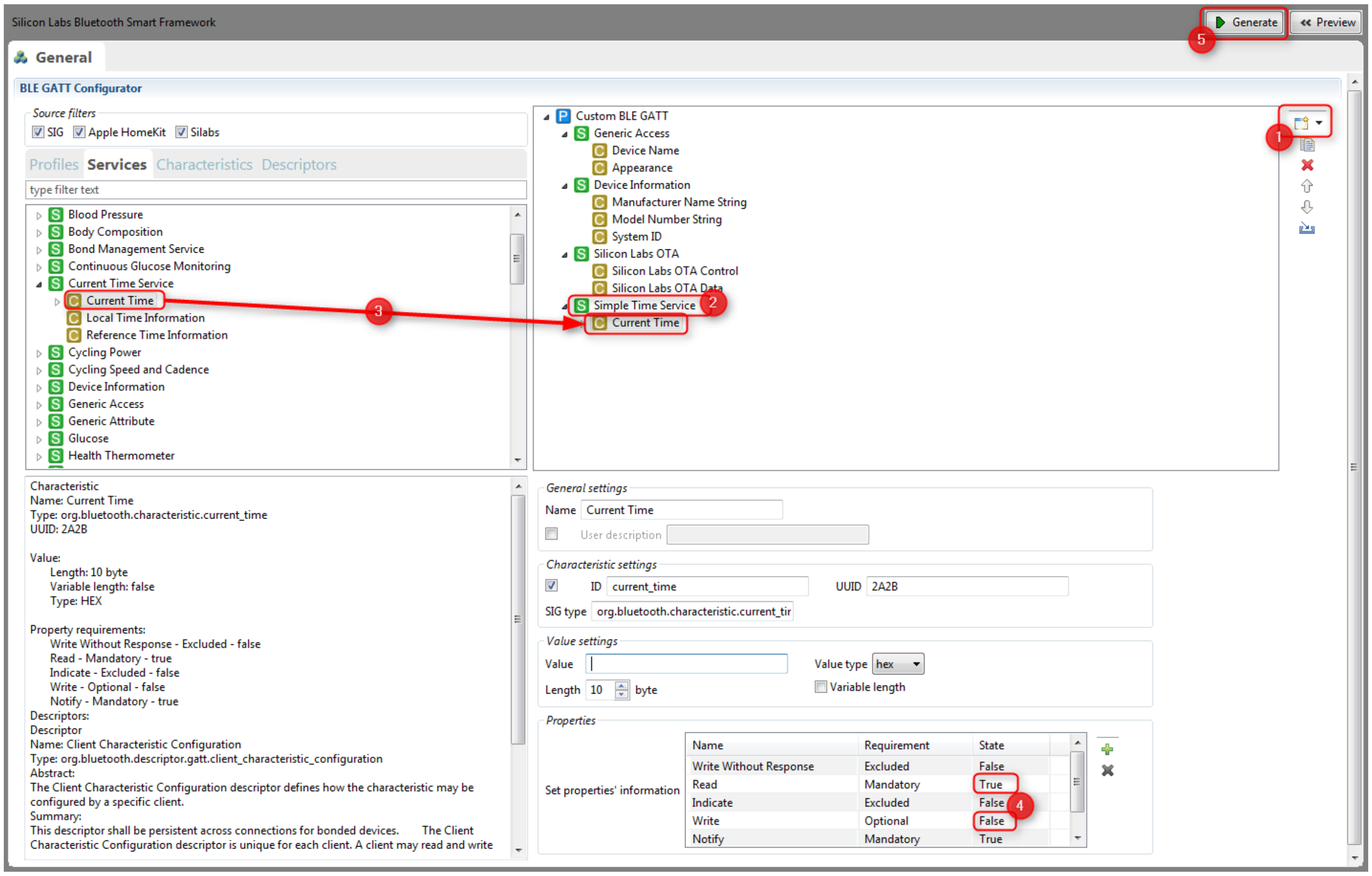
As an example of using the GATT editor, suppose you want to enable your application to query the current time on the device. A predefined *Time* Profile is provided for this purpose. This includes three services: Current Time Service, Next DST Change Service, and Reference Time Update Service. Assume the simplest case, where you are not interested in reference sources and DST changes but only want to know the current time. You can find the *Current Time* Service among the Services. It includes three Characteristics: Current Time Characteristic, Local Time Information Characteristic, and Reference Time information. You can simply drag and drop this Service to add it to your database.

However, suppose you do not even want to know about time zone, DST settings, and clock reference properties. You really only need the time. Since Characteristics can only be part of a Service, you have to define your own service, let us say Simple Time Service.

In the Options menu, click Create New Item (, 1 in the following figure) and add a new service. A Custom Service appears in the database. In the lower right pane, rename it to Simple Time Service (2). A 128-bit UUID is automatically generated for your custom service. Now you can drag and drop the Current Time Characteristic under your Simple Time Service (3).

Under Value Settings change the length from 2 to 10 bytes.

“Set properties information” is initially empty. Add the five properties shown (4). You can also set the write property to true, if you want to write the characteristic from a remote device.


Silicon Labs Bluetooth Smart Framework

General

BLE GATT Configurator

Source filters
☑ SIG ☑ Apple HomeKit ☑ Silabs

Profiles **Services** Characteristics Descriptors

type filter text

- Blood Pressure
- Body Composition
- Bond Management Service
- Continuous Glucose Monitoring
- Current Time Service
  - Current Time
  - Local Time Information
  - Reference Time Information
- Cycling Power
- Cycling Speed and Cadence
- Device Information
- Generic Access
- Generic Attribute
- Glucose
- Health Thermometer

Characteristic
Name: Current Time
Type: org.bluetooth.characteristic.current_time
UUID: 2A2B

Value:
Length: 10 byte
Variable length: false
Type: HEX

Property requirements:
Write Without Response - Excluded - false
Read - Mandatory - true
Indicate - Excluded - false
Write - Optional - false
Notify - Mandatory - true
Descriptors:
Descriptor
Name: Client Characteristic Configuration
Type: org.bluetooth.descriptor.gatt.client_characteristic_configuration
Abstract:
The Client Characteristic Configuration descriptor defines how the characteristic may be configured by a specific client.
Summary:
This descriptor shall be persistent across connections for bonded devices. The Client Characteristic Configuration descriptor is unique for each client. A client may read and write

- Custom BLE GATT
  - Generic Access
    - Device Name
    - Appearance
  - Device Information
    - Manufacturer Name String
    - Model Number String
    - System ID
  - Silicon Labs OTA
    - Silicon Labs OTA Control
    - Silicon Labs OTA Data
  - Simple Time Service
    - Current Time

General settings
Name: Current Time
☐ User description

Characteristic settings
☑ ID: current_time UUID: 2A2B
SIG type: org.bluetooth.characteristic.current_tir

Value settings
Value: Value type: hex
Length: 10 byte ☐ Variable length

Properties
Set properties' information

Name	Requirement	State
Write Without Response	Excluded	False
Read	Mandatory	True
Indicate	Excluded	False
Write	Optional	False
Notify	Mandatory	True

When you are finished editing the database, click **Generate** in the upper-right corner (5). This generates the following files:

gatt.xml – an xml format description of your database structure.

gatt_db.h – a header file that contains the definitions for your characteristic handles. You can read and write the values of your characteristics by referring to these definitions. The definition names are generated from the IDs given in the GATT editor.

gatt_db.c – a source file defining the default values of the characteristics.

Since the field structure of the Characteristics is not defined, you have to add it manually based on https://www.bluetooth.com/specifications/gatt/characteristics. For example, the Current Time Characteristics has the following structure:

```
PACKSTRUCT(struct date_time_t{
       uint16     year;
       uint8      month;
       uint8      day;
       uint8      hours;
       uint8      minutes;
       uint8      seconds;
});


PACKSTRUCT(struct day_of_week_t {
       uint8                  day;
});


PACKSTRUCT(struct day_date_time_t {
       struct date_time_t      date_time;
       struct day_of_week_t    day_of_week;
});


PACKSTRUCT(struct exact_time_256_t {
       struct day_date_time_t  day_date_time;
       uint8                   fractions_256;
});

PACKSTRUCT(struct current_time_t {
       struct exact_time_256_t exact_time_256;
       uint8                   adjust_reason;
});
```

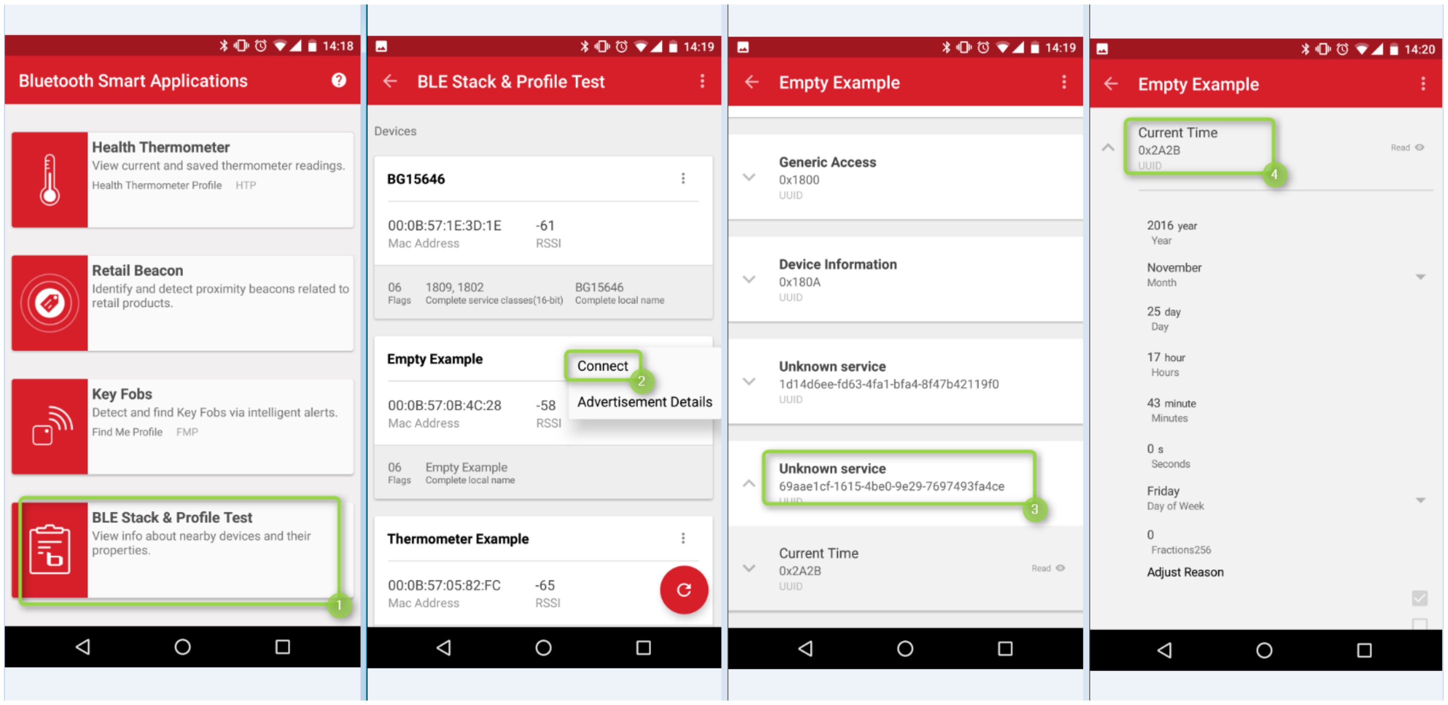
Most Characteristics have a much simpler structure than this, some of them simply a number or a string. If you calculate the size of this structure, you will get 10 bytes, which has to agree with the Length property in the Value settings of the Current Time Characteristic. A PACKSTRUCT directive is needed to avoid expansion of the structures to 32 bit.

Now you can modify the value of your characteristic in the database following the example below:

```
struct current_time_t current_time = {{{{2016,11,25,17,43,00},{5}},0},1};

gecko_cmd_gatt_server_write_attribute_value(gattdb_current_time,  0,  sizeof(struct  current_time_t),
(uint8*)&current_time);
```

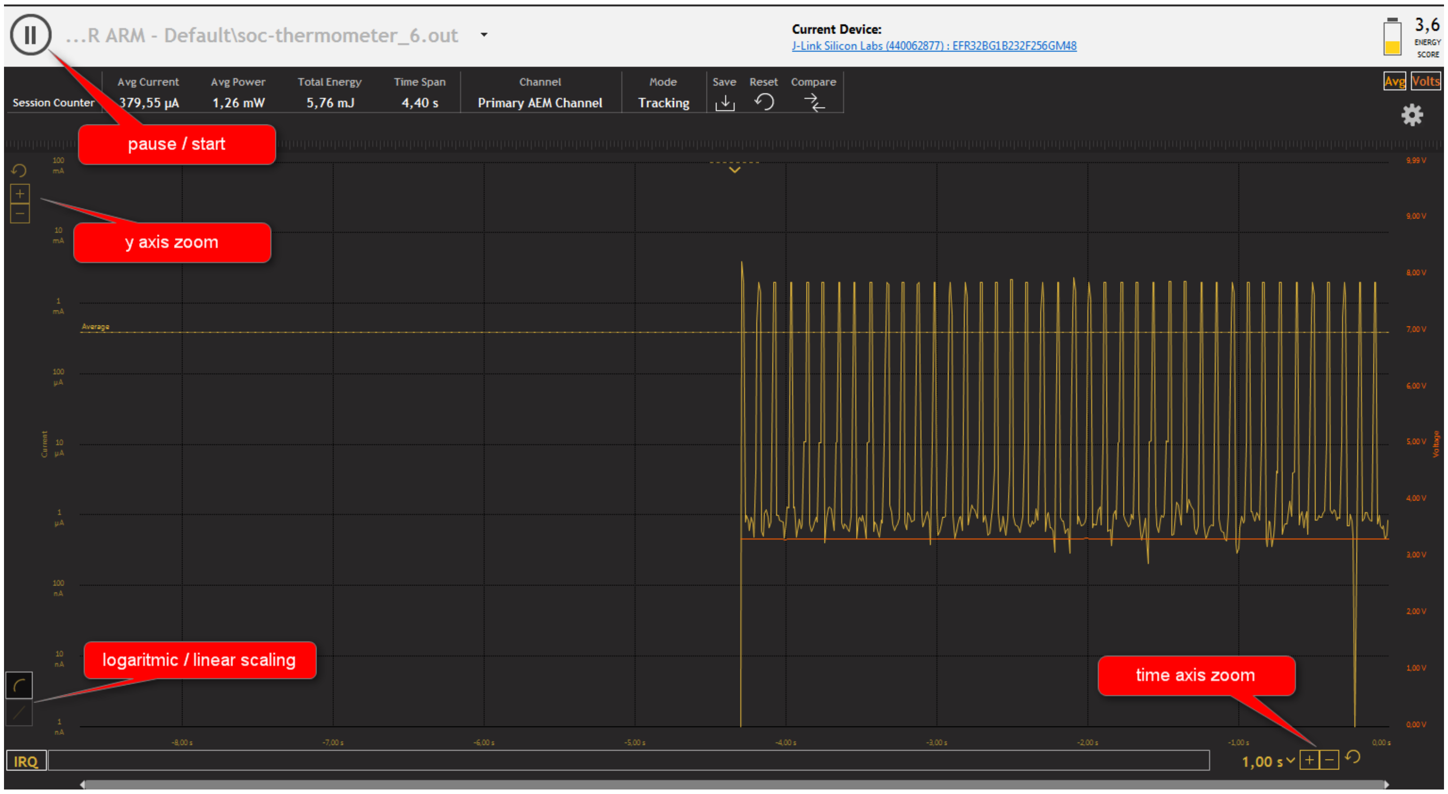
You can check the value by connecting to your device with a smartphone. Download the Blue Gecko app from the Android Market / App Store (http://www.silabs.com/bluegeckoapp), and select the BLE Stack & Profile Test menu (1). Here you can connect to your device (by default named as Empty Example) (2), and search for the Current Time characteristic. It will be found under an Unknown Service (3), since your custom Simple Time Service is not a standard service adopted by the Bluetooth SIG (Special Interest Group).


14:18
Bluetooth Smart Applications
Health Thermometer
View current and saved thermometer readings.
Health Thermometer Profile HTP
Retail Beacon
Identify and detect proximity beacons related to retail products.
Key Fobs
Detect and find Key Fobs via intelligent alerts.
Find Me Profile FMP
BLE Stack & Profile Test
View info about nearby devices and their properties.
1
14:19
BLE Stack & Profile Test
Devices
BG15646
00:0B:57:1E:3D:1E Mac Address
-61 RSSI
06 Flags
1809, 1802 Complete service classes(16-bit)
BG15646 Complete local name
Empty Example
Connect
2
Advertisement Details
00:0B:57:0B:4C:28 Mac Address
-58 RSSI
06 Flags
Empty Example Complete local name
Thermometer Example
00:0B:57:05:82:FC Mac Address
-65 RSSI
14:19
Empty Example
Generic Access 0x1800 UUID
Device Information 0x180A UUID
Unknown service 1d14d6ee-fd63-4fa1-bfa4-8f47b42119f0 UUID
Unknown service 69aae1cf-1615-4be0-9e29-7697493fa4ce UUID
3
Current Time 0x2A2B UUID Read
14:20
Empty Example
Current Time 0x2A2B UUID
4
Read
2016 year Year
November Month
25 day Day
17 hour Hours
43 minute Minutes
0 s Seconds
Friday Day of Week
0 Fractions256
Adjust Reason

7 Energy Profiler

Energy profiler is an add-on tool, with which you can easily measure the energy consumption of your device in runtime. You can easily find peak and average consumption, and check for sleep mode current.

To profile the current project, drop down the Profile as menu ( ) in the Simplicity IDE perspective and select **Profile as / Simplicity Energy Profiler target**. This automatically builds your project, uploads it to the device, and starts Energy Profiler. A new Energy Profiler perspective appears, shown in the following figure.



...R ARM - Default\soc-thermometer_6.out

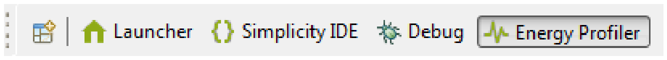
Current Device: J-Link Silicon Labs (440062877) : EFR32BG1B232F256GM48

3,6 ENERGY SCORE

Session Counter | Avg Current 379,55 µA | Avg Power 1,26 mW | Total Energy 5,76 mJ | Time Span 4,40 s | Channel Primary AEM Channel | Mode Tracking | Save | Reset | Compare

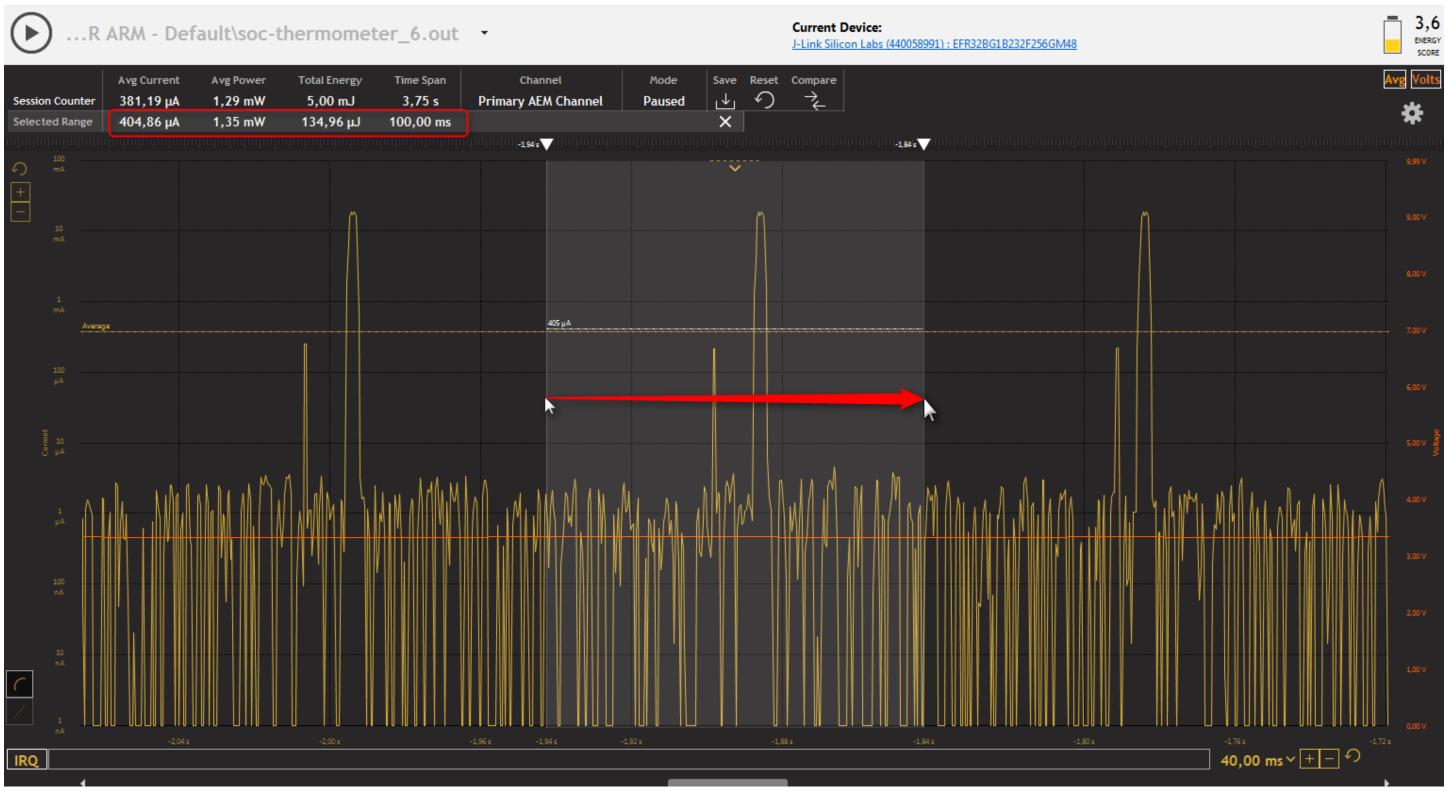
1,00 s

You can switch easily between Simplicity IDE and Energy Profiler perspectives using the Perspective buttons in the upper right corner of your current perspective.



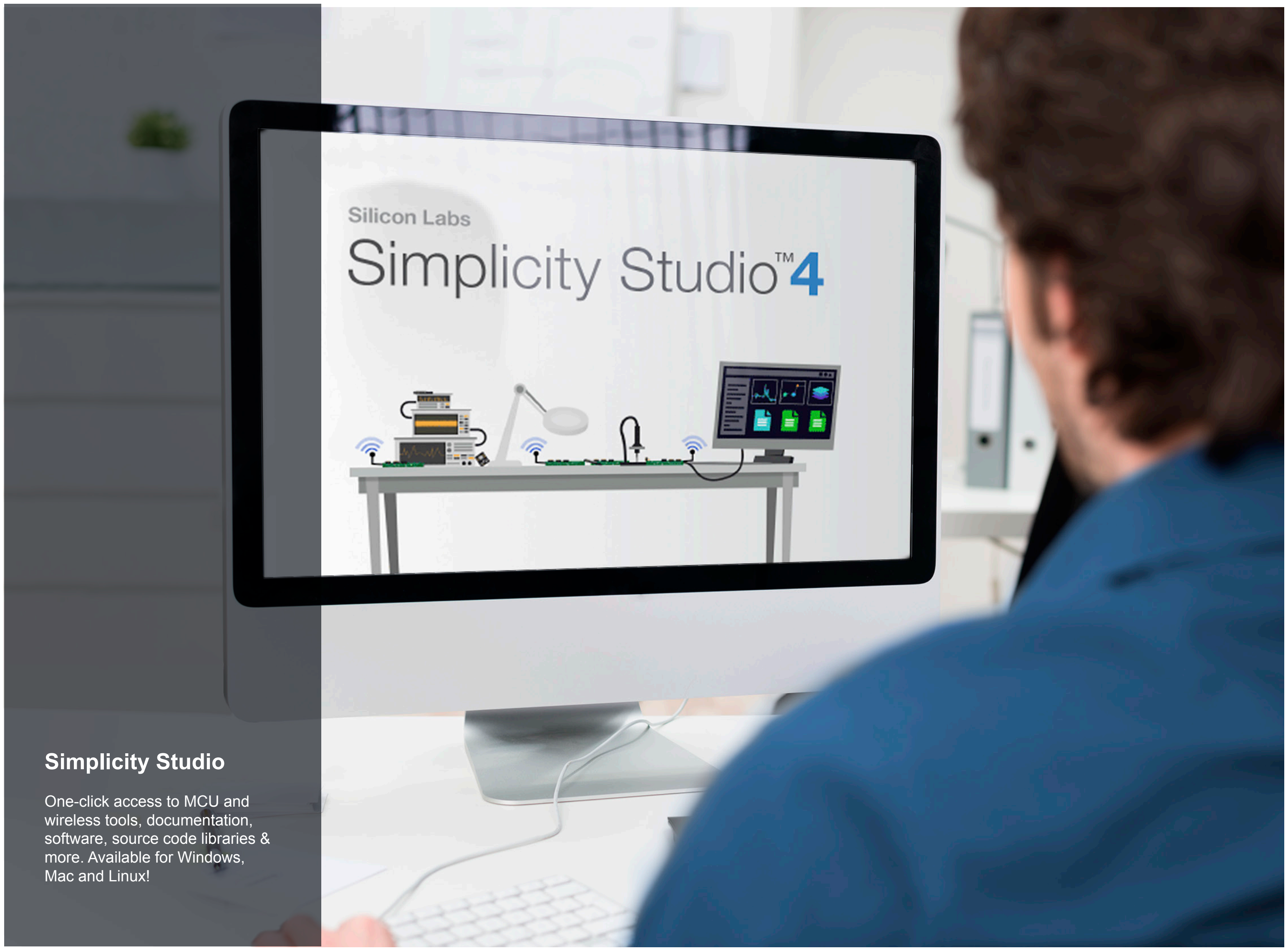
You can see peaks in the energy consumption diagram. Click one of them and zoom in with time axis zoom 1,00 s until you see small dots appearing on the diagram. These dots represent measurement points. Note, that the maximum consumption is much greater than it appeared on the diagram before you zoomed in. This is because in zoomed-out mode, the displayed values are averaged. If you need exact values, always zoom in until the dots appear.

To measure average consumption, simply click and drag your mouse over a time interval. A new bar appears showing consumption information for the given interval. Bluetooth communication typically has a periodicity: the advertisement or the connection interval. It is recommended to measure average over a connection interval to obtain a proper average consumption. Overall average is measured as well, but this is influenced by transient events.



Current Device:
J-Link Silicon Labs (440058991) : EFR32BG1B232F256GM48

	Avg Current	Avg Power	Total Energy	Time Span	Channel	Mode
Session Counter	381,19 µA	1,29 mW	5,00 mJ	3,75 s	Primary AEM Channel	Paused
Selected Range	404,86 µA	1,35 mW	134,96 µJ	100,00 ms		



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/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

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

http://www.silabs.com