

## ► Table of Contents

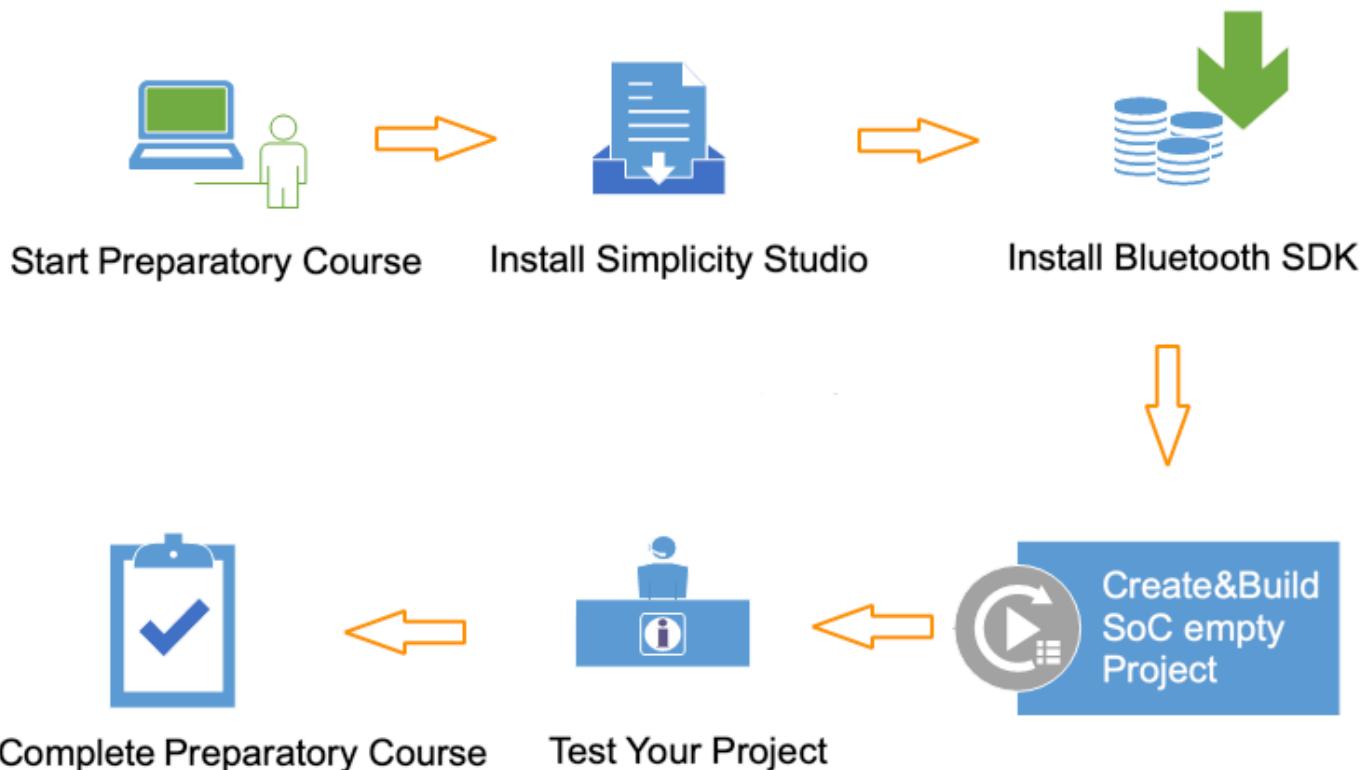
- [1. Introduction](#)
  - [2. Prerequisites](#)
    - [2.1. Hardware requirements](#)
      - [2.1.1 Explorer Kit](#)
      - [2.1.2 Working Computer](#)
    - [2.2. Software requirements](#)
  - [3. Getting started with BLE Application Development](#)
    - [3.1. Connect your Hardware](#)
    - [3.2. Using Gecko Bootloader](#)
    - [3.3. Create and Build the BLE project](#)
    - [3.4. Test your project](#)
  - [4. Fundamental Knowledge of BLE](#)
  - [5. Conclusion](#)
- 

## 1. Introduction

This is a preparatory course that you should take if you are new to the Silicon Labs development environment and BLE solution. The document describes how to get started with BLE development using the Bluetooth SDK and Simplicity Studio with a compatible Explorer kit(EK) board, and also the fundamental BLE knowledge you should understand before beginning to study the BG24 training courses.

And step-by-step instructions are provided to demonstrate how to create a basic BLE project for checking if the development environment is ready on your end.

The figure below illustrates the working flow for setting up the development environment.



## 2. Prerequisites

### 2.1. Hardware requirements

#### 2.1.1 Explorer Kit

Before following the procedures in this guide you should have purchased one of the Bluetooth development kits. You can use the [EFR32xG24 Dev Kit](#) or [xG24 Explorer Kit](#) in the training hands-on.

#### 2.1.2 Working Computer

Below is the system requirements of the Simplicity Studio v5, and it's strongly recommended to prepare a working computer has at least 8GB RAM for wireless project development.

Operating System	Version
Windows	Windows 10 (64-bit)
macOS	10.15 Catalina
Linux	Ubuntu 20.04 LTS

Hardware Component	Item
CPU	1 GHz or better
Memory	<b>8 GB for Wireless Protocol development</b>
Disk Space	7 GB for Wireless Dynamic Protocol support

## 2.2. Software requirements

To develop BLE applications, you will need to setting up your software development environment by installing Simplicity Studio, Bluetooth SDK and the compatible toolchains.

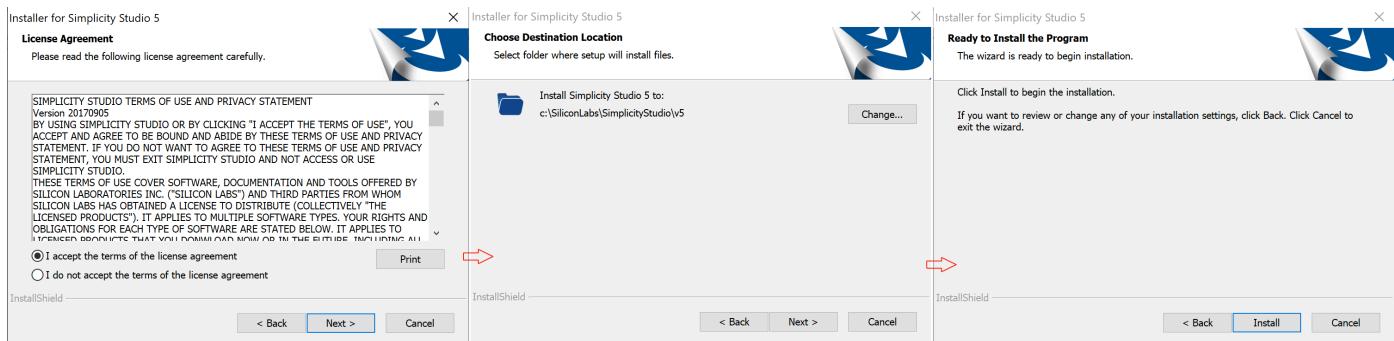
Simplicity Studio is the unified development environment for all Silicon Labs technologies, SoCs, and modules. It provides you with access to the target device-specific web and SDK resources, software and hardware configuration tools, and an integrated development environment (IDE) featuring industry-standard code editors, compilers, and debuggers. With Simplicity Studio, you get a complete set of advanced value-add tools for network analysis and code-correlated energy profiling. GSDK v4.1.2 or later will be require for the hands-on section.

If you do not have Simplicity Studio V5 installed, please connect to [Simplicity Studio 5](#) to download the installation package, and run the Simplicity Studio installation application.

**Note:** For better wireless protocol development experience, please check the items below one by one.

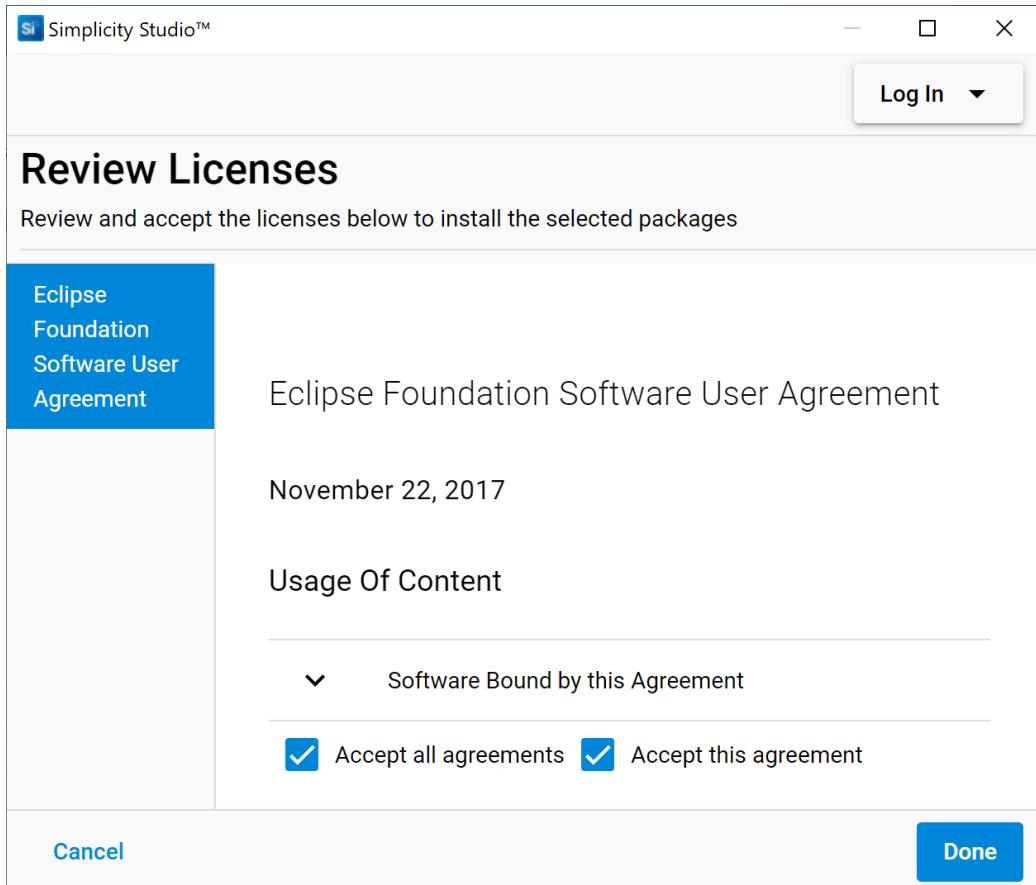
1. Make sure that your PC has 8 GB or more RAM memory and more than 7 GB disk space available.
2. If you are using the Windows (Windows 10 is recommended), it highly recommended to install the Simplicity Studio to the **C:** drive. Otherwise, you need to create another workspace locates at the same drive as your Simplicity Studio installed, and switch to that workspace.

When Simplicity Studio first launches, it presents a License Agreement dialog. Accept the terms of the agreement and click [Next].



Next step is SDK location selection, recommend to use default setting, click [Next] > and then click [Install].

License Agreement dialog again, all check on, click [Done].



Then it require to log in. If you have no account, we will suggest click [Create an Account] for creating a new account and log in then. You can also skip the Log in process for now by clicking right side [Skip log in for now].



— □ ×



Email

Password

Show Password

Remember Me

**Log in**

**Skip log in for now >**

You can log in later from the launcher.

[Terms and Conditions](#)

[Create an Account](#)

[Forgot password?](#)

Click right side [Install by technology type].

Simplicity Studio™

— □ ×

## Installation Manager

**Log In** ▾



Install by connecting  
device(s)



Install by technology type  
(wireless, Xpress, MCU,  
sensors)

[Cancel](#)

Check on 32-bit and Wireless MCUS, this option will install Wireless SDK include Bluetooth SDK and relevant tooltaich.

Simplicity Studio™

## Installation Manager

Log In ▾

Select Technology Type    Select Development Packages    Review Licenses

### Select Technology Type

Select technology type to use with your products

Select All (5)

 32-bit and Wireless MCUs

Extensions:

Matter Enablement Package

Cancel Back Next

Package Installation Options, use default setting, click [Next].

Simplicity Studio™

## Installation Manager

Log In ▾

Select Technology Type    Select Development Packages    Review Licenses

### Package Installation Options

**Auto**  
Select this option to let Simplicity Studio install all the recommended development packages based on the previously-selected product.

**Advanced**  
Select this option to customize installed development packages based on your requirements.

Cancel Back Next

License Agreement dialog again, all check on, click [Next].

## Installation Manager

Log In ▾

 Select Technology Type Select Development Packages Review Licenses

## Review Licenses

Review and accept the licenses below to install the selected packages

## Master Software License Agreement

Silicon Labs VCP Driver End User License Agreement

## MASTER SOFTWARE LICENSE AGREEMENT

Version 20210909

▼ Software Bound by this Agreement

 Accept all agreements  Accept this agreement

Cancel

Back

Next

Installing, this will take time.

## Installation Manager

Log In ▾

Now installing...

Cloning SDK: Gecko SDK - 32-bit and Wireless MCUs Receiving objects: 76% (69492/91302), 113.82 MiB | 72.00 KiB/s

## Resources

Learn some tips and tricks to help you ramp up using Simplicity Studio

 Simplicity Studio User's Guide

The official Simplicity Studio 5 User's Guide

Open

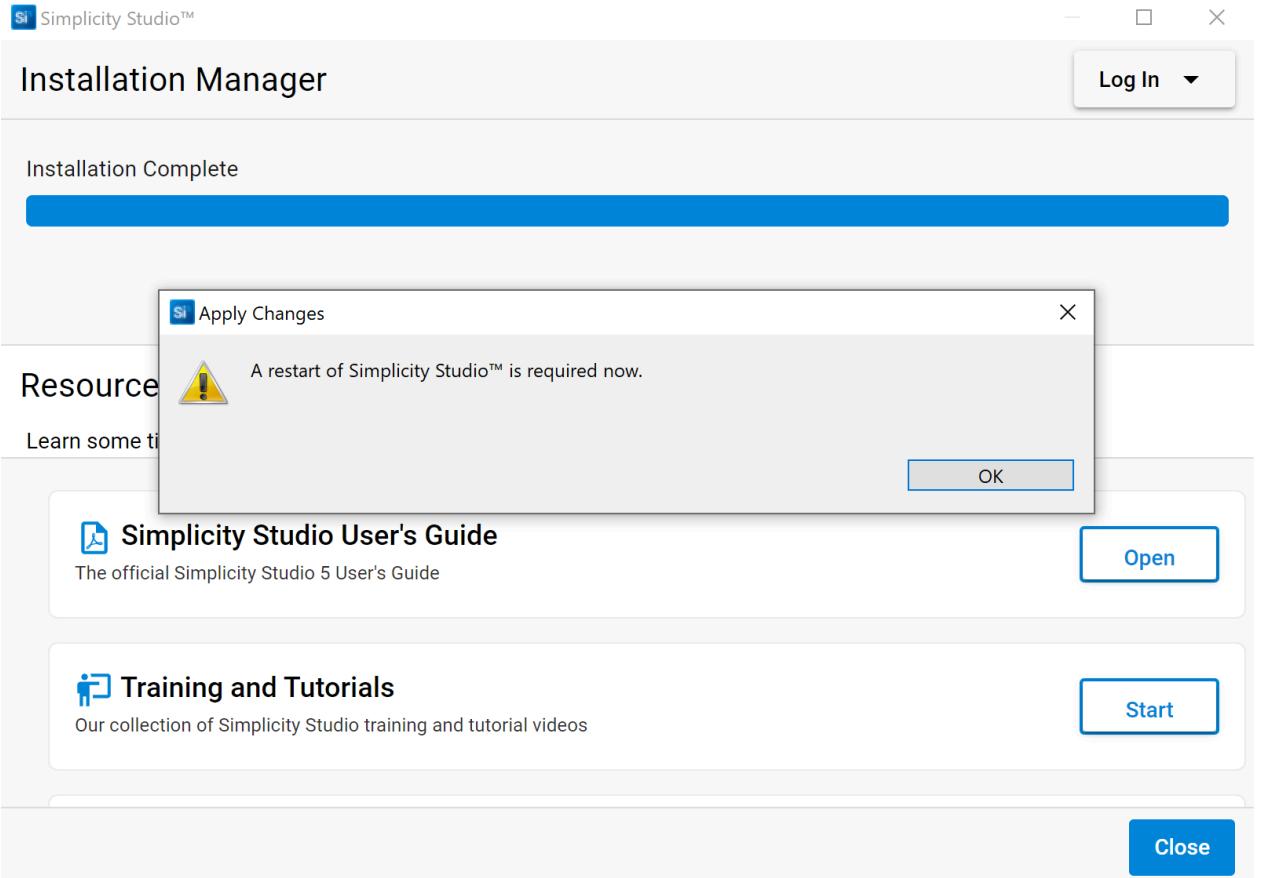
 Training and Tutorials

Our collection of Simplicity Studio training and tutorial videos

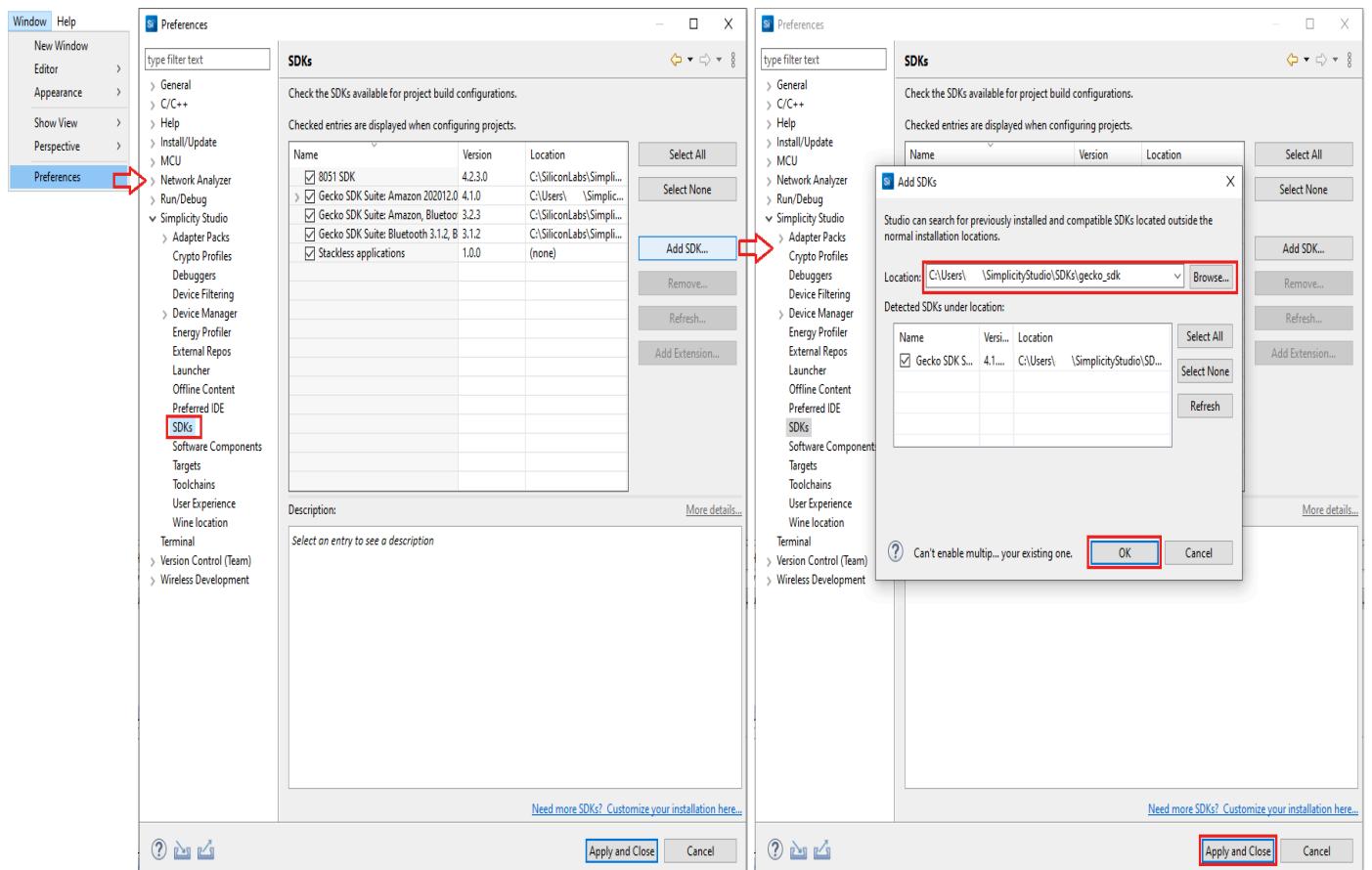
Start

Cancel

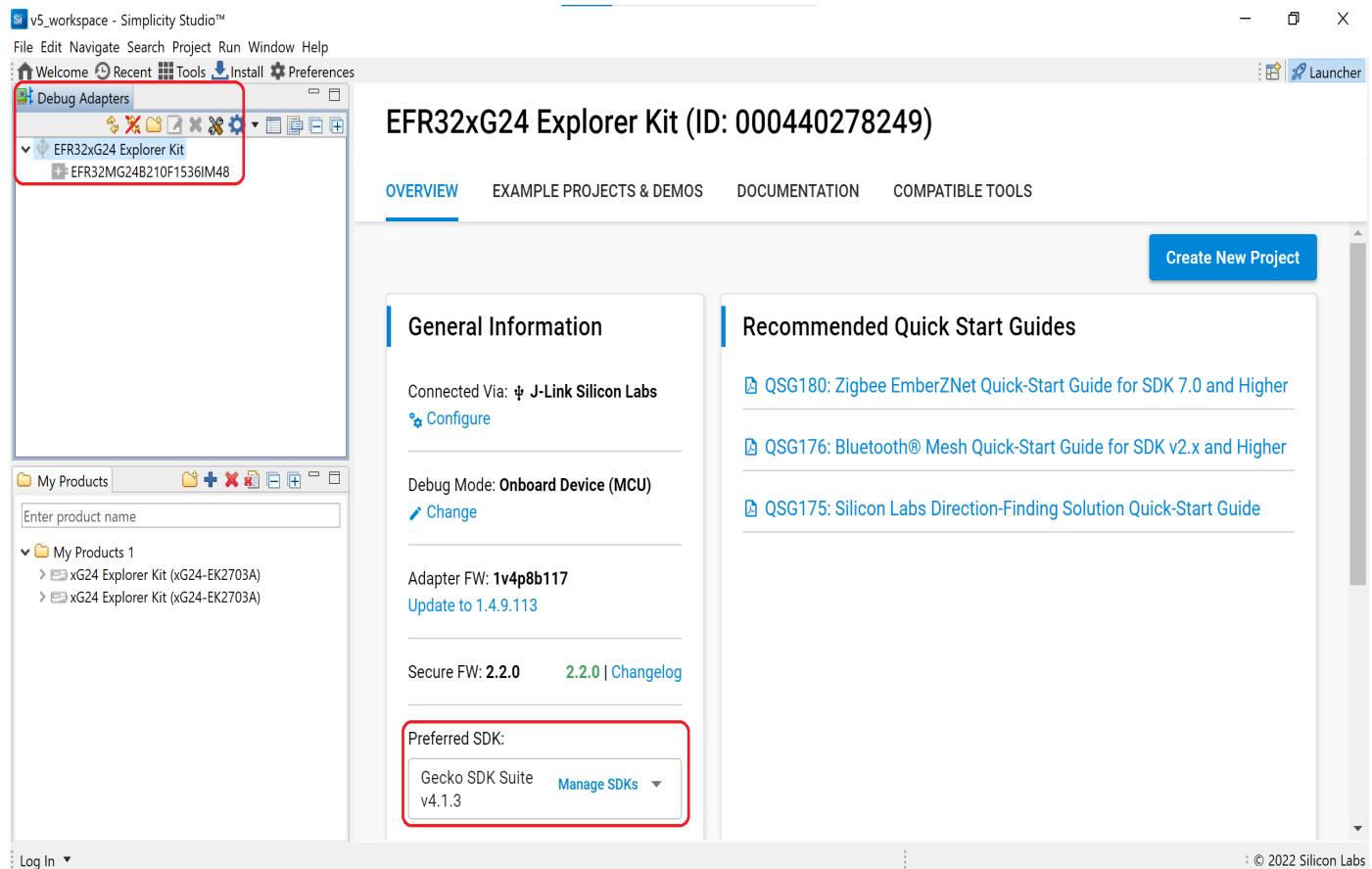
After GSDK installation complete, Bluetooth SDK have installed, the compiler toolchain GCC have installed as well, Simplicity Studio request a restart. If it show Installation Fail, then need to re-install.



If you have tried several times but not installed successfully, we recommend download GSDK from [github](#) then import it. Refer to the following figure.



After Simplicity Studio restart, then we can check the SDK version. The training hands-on request GSDK v4.1.2 or later.



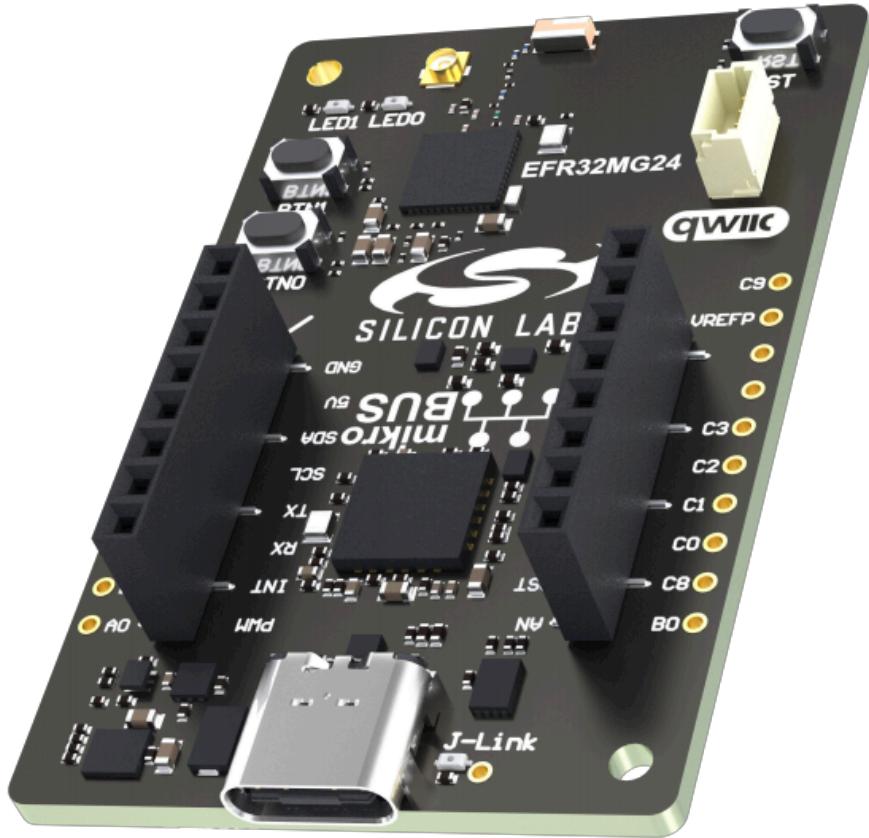
## 3. Getting started with BLE Application Development

As a preparatory course, the purpose of this section is providing a step-by-step instructions to demonstrate how to get start with the **Bluetooth - SoC Empty** example project to verify that the development environment is setting done on your side.

### 3.1. Connect your Hardware

**Note:** If you don't have Explorer kit yet, please just type the board name "EK2703A" in the "My Products" tab on the lower-left of the launcher perspective. And jump to the section [Create and Build the BLE project](#).

Connect your EK board using a type C USB cable to the PC on which has Simplicity Studio installed.



## 3.2. Using Gecko Bootloader

---

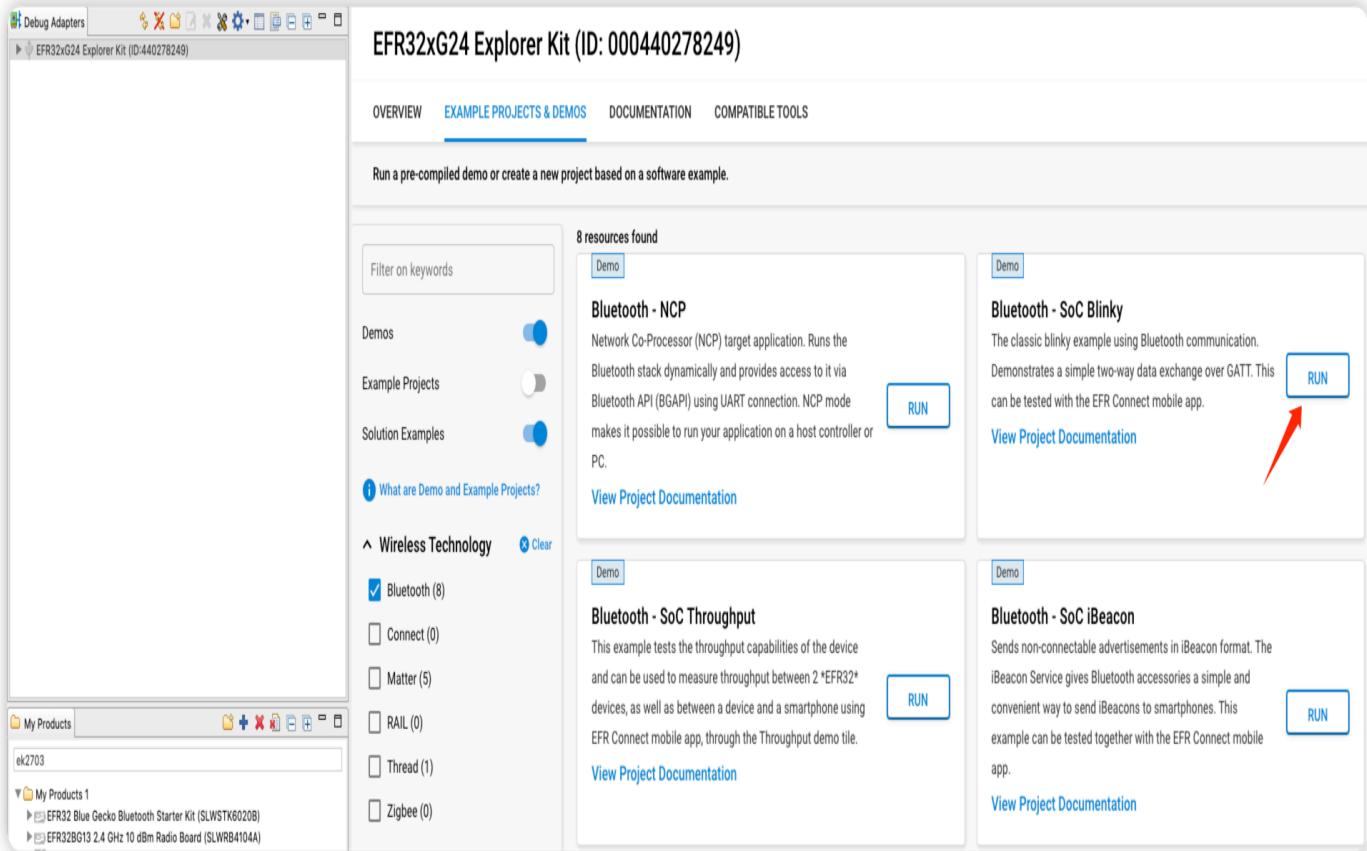
A bootloader is a program stored in reserved flash memory that can initialize a device, update firmware images, and possibly perform some integrity checks. If the application seems to do not running, always check the bootloader, because lack of it causes program crash.

There are two possible way to have a bootloader application.

- Use Silicon Labs pre-built image (not for all boards).
- Create your own bootloader project.

The 1st option is the easiest solution. Filter and left Demos list only, select **Bluetooth - SoC Blinky** and click RUN button.

Each Gecko SDK contain pre-built bootloader images for different boards.



The 2nd way is to create and build your own bootloader application. It is possible to customize and add new features to it, but the current lab doesn't detail these possibilities.

1. In Launcher perspective -> EXAMPLE PROJECTS & DEMOS, check Bootloader on, input "apploader" for filtering. Then we can see **Bootloader - SoC Bluetooth AppLoader OTA DFU** list on the top.

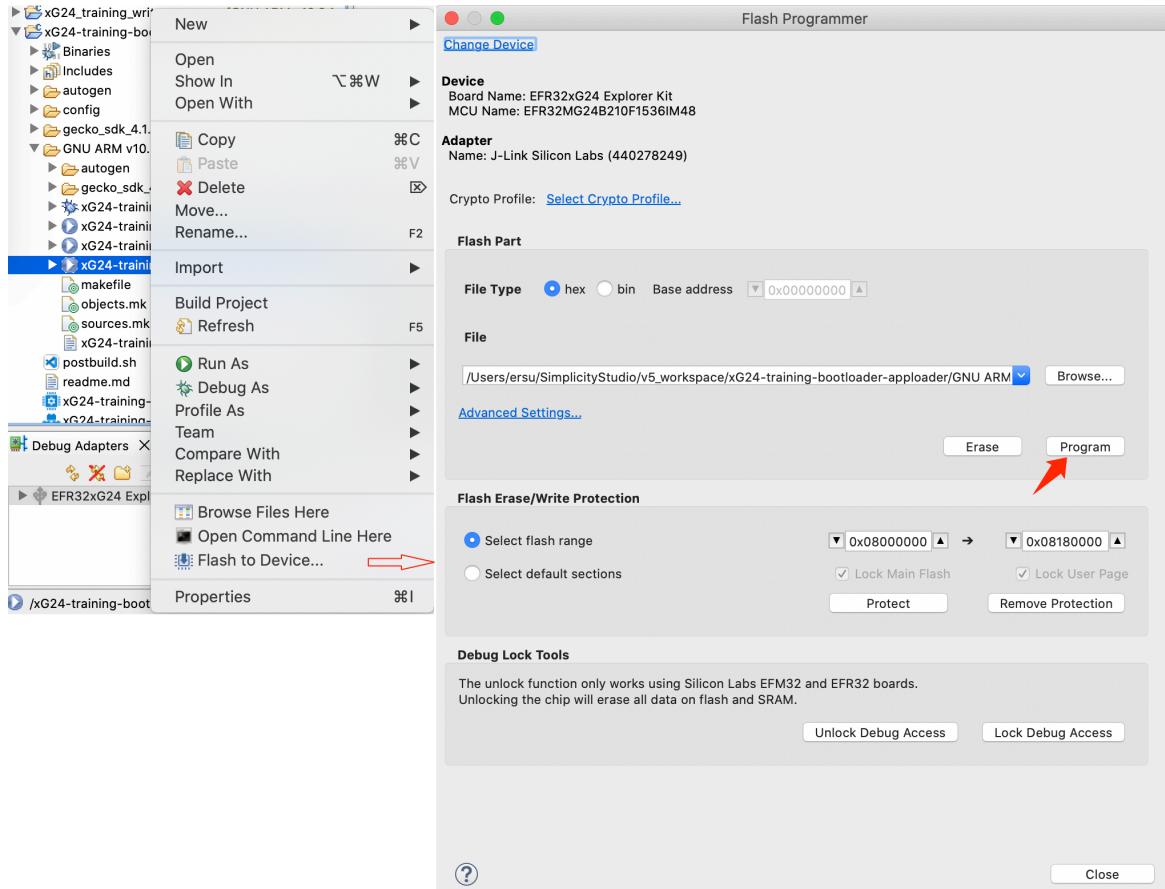
The screenshot shows the EFR32xG24 Explorer Kit software interface. At the top, there's a toolbar with various icons. Below it, the title bar says "EFR32xG24 Explorer Kit (ID: 000440278249)". Underneath the title, there are tabs: "OVERVIEW", "EXAMPLE PROJECTS & DEMOS" (which is selected), "DOCUMENTATION", and "COMPATIBLE TOOLS". A sub-header says "Run a pre-compiled demo or create a new project based on a software example." On the left, there's a sidebar with sections like "Demos", "Example Projects", "Solution Examples", and expandable categories for "Wireless Technology", "Device Type", "Ecosystem", "MCU", and "Capability". Under "MCU", "Bootloader (3)" is checked. A search bar at the top has "apploader" typed into it, with a red arrow pointing to it. To the right of the search bar, there's a "CREATE" button with a red arrow pointing to it. The main area shows search results: "Bootloader - SoC Bluetooth AppLoader OTA DFU" (with a detailed description and "View Project Documentation" link), "Bootloader - SoC Bluetooth AppLoader OTA DFU Non-Secure part of Bootloader using TrustZone" (with a "CREATE" button and "View Project Documentation" link), and "Bootloader - SoC Bluetooth AppLoader OTA DFU Secure part of Bootloader using TrustZone" (with a "CREATE" button and "View Project Documentation" link).

2. Select **Bootloader - SoC Bluetooth AppLoader OTA DFU**. Click [CREATE].

3. Rename the project, select “Copy contents”. Click [FINISH].

The screenshot shows the "New Project Wizard" - "Project Configuration" step. The title bar says "New Project Wizard". Below it, the section title is "Project Configuration" with the sub-instruction "Select the project name and location.". There are three tabs at the top: "Target, SDK" (selected, indicated by a blue checkmark), "Examples", and "Configuration". In the main area, there's a "Project name:" input field containing "xG24-training-bootloader-apploader", with a red arrow pointing to it. Below it is a "Use default location" checkbox with a checked blue box and a red arrow pointing to it. Under "Location:", there's a text input field with "/Users/ersu/SimplicityStudio/v5\_workspace/xG24-training-bootloader-apploader" and a "BROWSE" button. Further down, there's a section "With project files:" with three radio button options: "Link to sources", "Link sdk and copy project sources", and "Copy contents" (which is selected, indicated by a blue circle). At the bottom, there are "CANCEL", "BACK", "NEXT", and "FINISH" buttons.

4. Build by clicking on the  and select the s37 file, go to [Flash to Device...] and select the device for programming.



### 3.3. Create and Build the BLE project

By having the EK board connected, the Simplicity Studio will automatically list the available example application in the launcher perspective, click [CREATE] on example project, the Simplicity Studio will new a project based on the selected example.

**Note:** If you don't have EK board yet, you also can just type the board name "EK2703A" in the "My Products" tab on the lower-left of the launcher perspective. The Simplicity Studio will automatically list the available example application in the launcher perspective as well.

Below is the step-by-step instructions.

1. Select your Device in the "Debug Adapters" or "My Products" tab on the left. And check the selected SDK version is the required one.
2. In EXAMPLE PROJECTS & DEMOS view, check Bluetooth on, input "empty" for filtering. Then we can see **Bluetooth - SoC Empty** list on the top.

EFR32xG24 Explorer Kit (ID: 000440278249)

OVERVIEW EXAMPLE PROJECTS & DEMOS DOCUMENTATION COMPATIBLE TOOLS

Run a pre-compiled demo or create a new project based on a software example.

Filter on keywords empty

Demos

Example Projects

Solution Examples

What are Demo and Example Projects?

Wireless Technology Clear

Bluetooth (5)

Connect (0)

Matter (0)

RAIL (2)

Thread (0)

Zigbee (0)

5 resources found

**Bluetooth - SoC Empty**

A minimal project structure, that serves as a starting point for custom Bluetooth applications. The application starts advertising after boot and restarts advertising after a connection is closed.

[View Project Documentation](#) **CREATE**

**Bluetooth RAIL DMP - SoC Empty FreeRTOS**

A minimal project structure, used as a starting point for custom Bluetooth + Proprietary DMP (Dynamic Multiprotocol) applications. It runs on top of FreeRTOS and multiprotocol RAIL.

[View Project Documentation](#) **CREATE**

**Bluetooth RAIL DMP - SoC Empty Micrium OS**

A minimal project structure, used as a starting point for custom Bluetooth + Proprietary DMP (Dynamic Multiprotocol) applications. It runs on top of Micrium OS and multiprotocol RAIL.

[View Project Documentation](#) **CREATE**

3. Select **Bluetooth - SoC Empty**, Click [CREATE] for create a new project, then click [FINISH] directly.
4. Build you project by clicking [Build] in the top tool bar.
5. The build should complete with no errors.
6. Right-click the generated hex file, go to [Flash to Device...] and select the device for programming.

**Note:** You should got similar as below at the end of the build log in the build console

```
CDT Build Console [bt_soc_empty]
building hex file: bt_soc_empty.hex
arm-none-eabi-objcopy -O ihex "bt_soc_empty.axf" "bt_soc_empty.hex"

Building bin file: bt_soc_empty.bin
arm-none-eabi-objcopy -O binary "bt_soc_empty.axf" "bt_soc_empty.bin"

Building s37 file: bt_soc_empty.s37
arm-none-eabi-objcopy -O srec "bt_soc_empty.axf" "bt_soc_empty.s37"

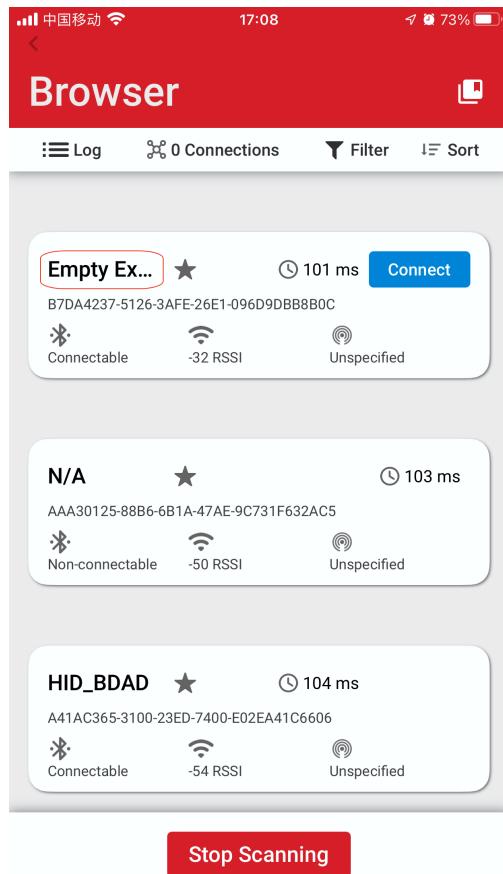
Running size tool
arm-none-eabi-size "bt_soc_empty.axf" -A
bt_soc_empty.axf :
section           size      addr
.text            180284  134291456
.ARM.exidx        8    134471740
.copy.table       12   134471748
.zero.table        0   134471760
.stack            2752   536870912
.data             952   536873664
.bss              6088   536874616
.heap             252352  536880704
.nvm              40960  134471760
.ARM.attributes     54      0
.comment          150      0
.debug_info        370102  0
.debug_abbrev      46066  0
.debug_loc         195958  0
.debug_aranges     7680      0
.debug_ranges      14312  0
.debug_macro       282332  0
.debug_line        356899  0
.debug_str         1366797  0
.debug_frame       23104      0
Total            3146862

16:33:50 Build Finished. 0 errors, 0 warnings. (took 10s.563ms)
```

## 3.4. Test your project

---

After the image has been downloaded to your development kit, it's possible to communicate with the device. The UART log is not enable by default, so we need to check it on smartphone. We need to install and open our Bluetooth app EFR Connect. In Brower view, we can see the "Empty Ex..." device, this is our target device.



## 4. Fundamental Knowledge of BLE

The BLE training will cover the topics of BLE Basic Knowledge, xG24 and SSv5 introduction, how to optimize for power consumption, how to implement OTA DFU , BGM/CGM applicaton, BLE smart Key, IOP and indoor position.

Although this series of training are designed for new to Silicon Labs BLE solution, it highly recommended to go though the documentations below before attending this series course.

[UG103.14: Bluetooth LE Fundamentals](#)

## 5. Conclusion

We hope that you have completed the preparatory course well, and set up the development environment as the preparation for the coming BLE training course.