
Silicon Labs EFR32FG23 开发板 试用报告

2022 年 5 月

目 录

1 拆箱.....	1
2 开发环境搭建	2
2.1 下载并安装 Simplicity Studio 5 (SSv5)	2
2.2 使用 SSv5	5
3 SSv5 和 Flex SDK 使用	9
3.1 安装 SDK	9
3.2 创建工程	14
3.3 图形化工程配置器 (*.slcp)	16
3.4 工程评测	19
4 Connect 和 RAIL	24
4.1 Connect	24
4.2 RAIL	25
参考资料:	26

2	胶棒天线 2 根 (1 根 868 MHz、 1 根 915 MHz)	
3	Micro-USB 连接线 1 根	
4	说明卡纸 2 张	

2 开发环境搭建

2.1 下载并安装 Simplicity Studio 5 (SSv5)

SSv5 是由 Silicon Labs 提供的一个免费的基于 Eclipse 集成开发环境 (IDE) 和一组增值工具。开发人员可以使用 SSv5 开发、调试和分析他们的 Flex 和其他基于 Silicon Labs SDK

的应用程序。

从官网上下载 SSv5（下载地址：<https://cn.silabs.com/development-tools/wireless/proprietary/efr32fg23-868-915-mhz-14-dbm-dev-kit>），根据所用 PC 的操作系统（Windows、Mac、Linux），选择相应的版本下载并安装，如图 2 所示。

第一步：下载并安装 Simplicity Studio 版本 5

借助预先编译的演示、应用说明和示例快速构建和运行。使用包括能源分析功能在内的高级工具优化您的 MCU 应用。在下方选择您的软件封装。



图 2 SSv5 支持的操作系统类型

此处，以 Windows 下的安装程序为例说明。如果没有 Silicon Labs 账号，弹出需要创建一个 SILICON LABS 账号，如图 3 所示，点击 [Create an Account](#)。如果有账号，可以直接 [Log in](#)。

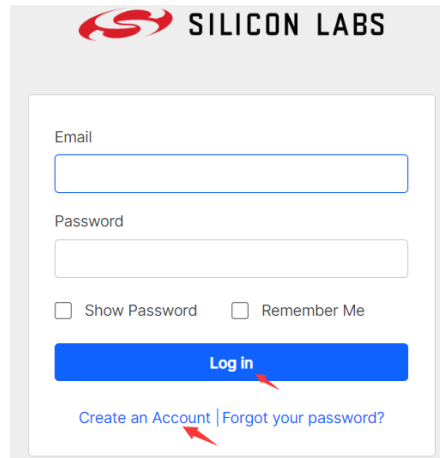


图 3 创建 SILICON LABS 账号

下载完成后，


SimplicityStudio-5	2022/5/9 23:29	光盘映像文件	534,912 KB
--------------------	----------------	--------	------------

，可以看到 SimplicityStudio-5.iso 是一个光盘映像文件，双击该文件可以自动导入虚拟光驱，如图 4 所示。

] > DVD 驱动器 (E:) SimplicityStudio-5

名称	修改日期	类型	大小
0x0409	2016/3/22 3:04	配置设置	22 KB
autorun	2022/4/21 3:47	安装信息	1 KB
data1	2022/4/21 3:46	Cab 文件	865 KB
data1.hdr	2022/4/21 3:46	HDR 文件	896 KB
data2	2022/4/21 3:47	Cab 文件	528,269 KB
ISSetup.dll	2018/9/20 19:15	应用程序扩展	1,588 KB
layout.bin	2022/4/21 3:47	BIN 文件	1 KB
setup	2022/4/21 3:44	BMP 文件	484 KB
Si setup	2022/4/21 3:47	应用程序	167 KB
setup	2022/4/21 3:46	配置设置	3 KB
setup.inx	2022/4/21 3:46	INX 文件	270 KB
z	2022/4/21 3:47	应用程序	1,200 KB

图 4 SimplicityStudio-5.iso 文件结构

点击 setup 应用程序  ，根据向导自动安装 SSv5，如图 5-图 6。

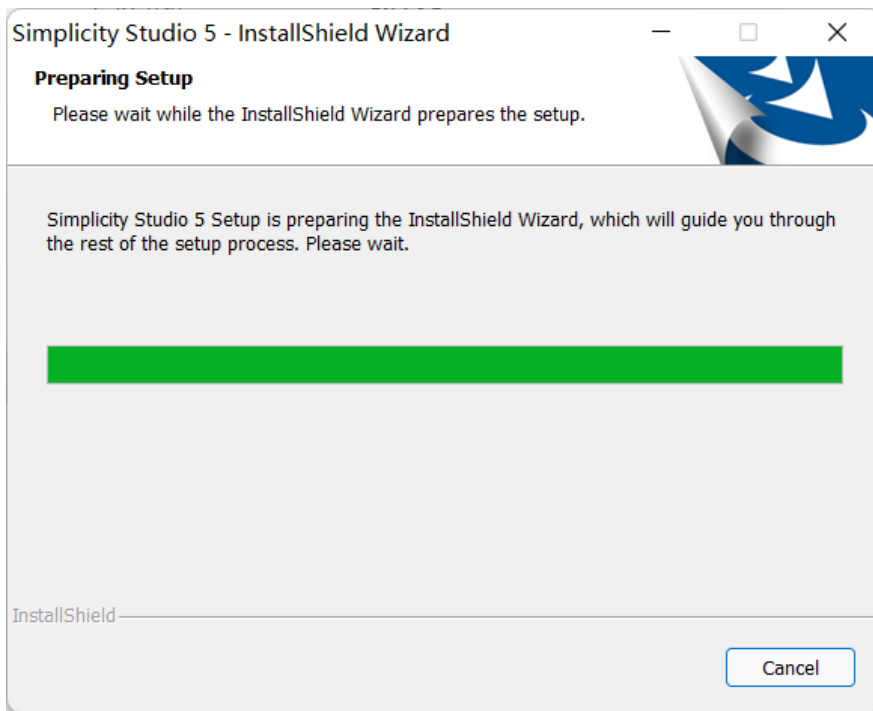


图 5 SSv5 自动安装 1

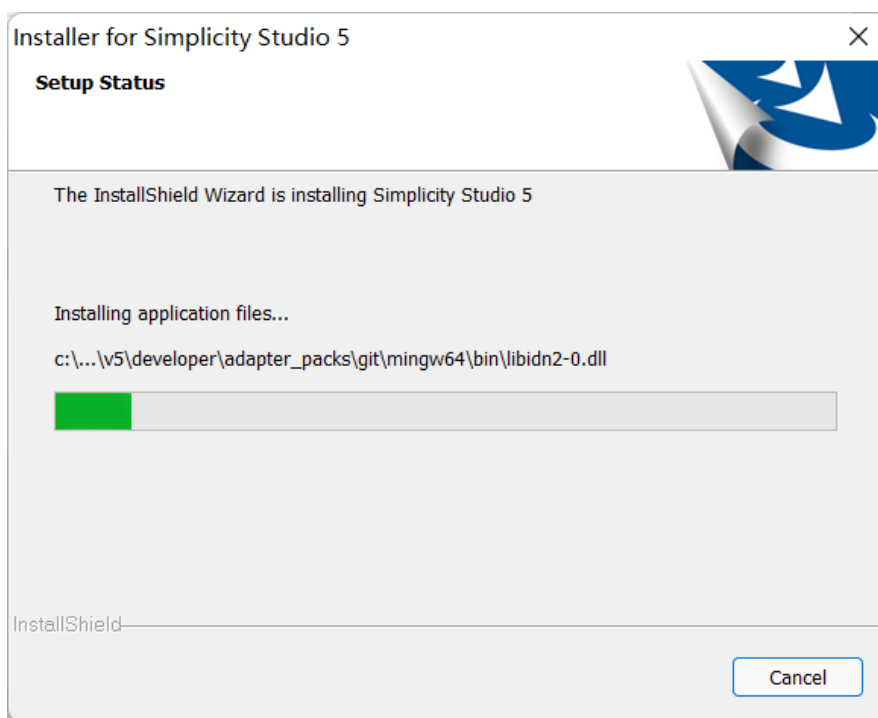


图 6 SSv5 自动安装 2

安装完成后，整个文件夹大约占用 3.5G 的存储空间（不包括后续需要升级、下载所需 SDK 等），如图 7 所示。

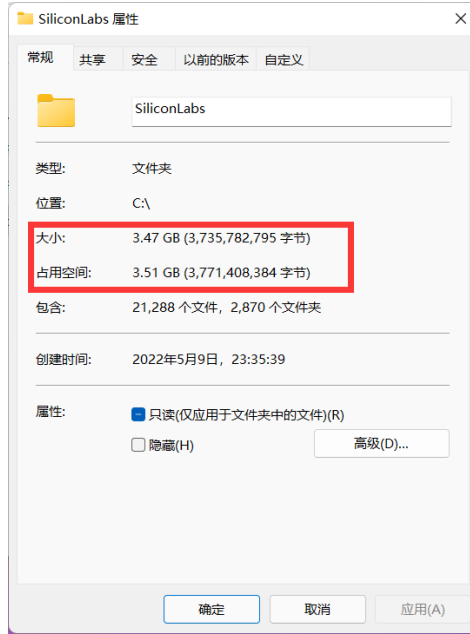


图 7 SSv5 占用存储空间

2.2 使用 SSv5

安装完成后，双击图标  打开 SSv5 。首次打开会检查 Licenses，如图 8 所示。

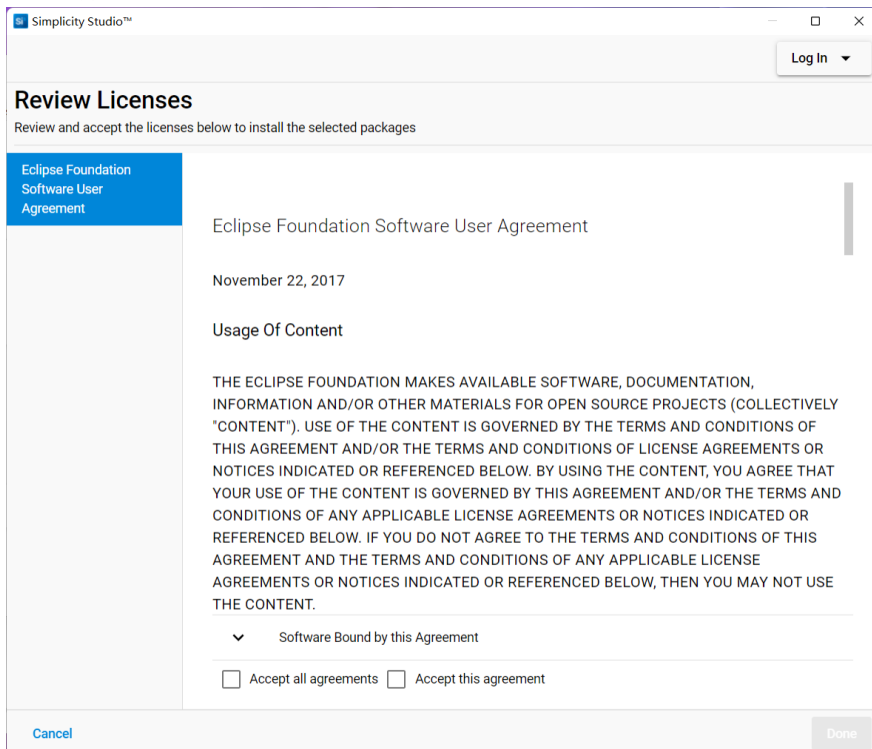
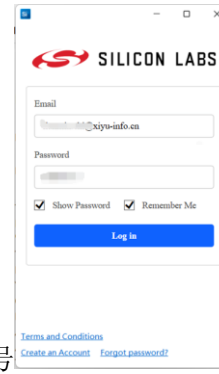
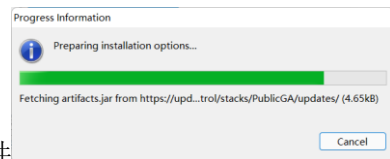


图 8 Review Licenses

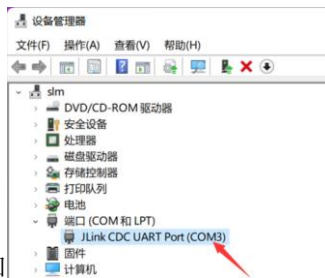
可以直接打钩 Accept all agreements，接受所有的 agreements。



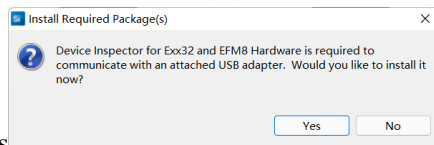
然后，建议选择右上角 ，登录  SILICON LABS 账号 ，如果没有账号，建议创建一个。



接受 agreements 后，会自动检查并下载一些更新文件。通过 USB Micro-B 线将 EFR32FG23 开发板连接到 PC 后，不需要安装驱动，可以在设



备管理器中看到 USB 虚拟串口 。同时，SSv5 界面自动弹出安装



所需的 Packages (例如：连接 Exx32 和 EFM8 硬件设备

需要 USB 适配器，这时将自动下载 USB 适配器驱动)。点击 Yes 按钮 自动安

装。然后，安装 Manager (packages)，安装方式有两种：(1) 通过连接的设备安装，本次试用以这种方式安装；(2) 通过技术类型 (无线、Xpress、MCU、传感器等) 安装，如图 9 所示。

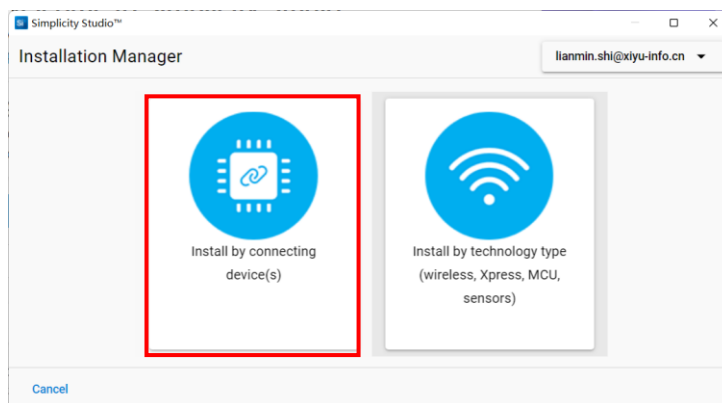


图 9 选择安装方式

点击通过连接的设备安装 ，SSv5 自动找到连接的设备，如图 10 所示。

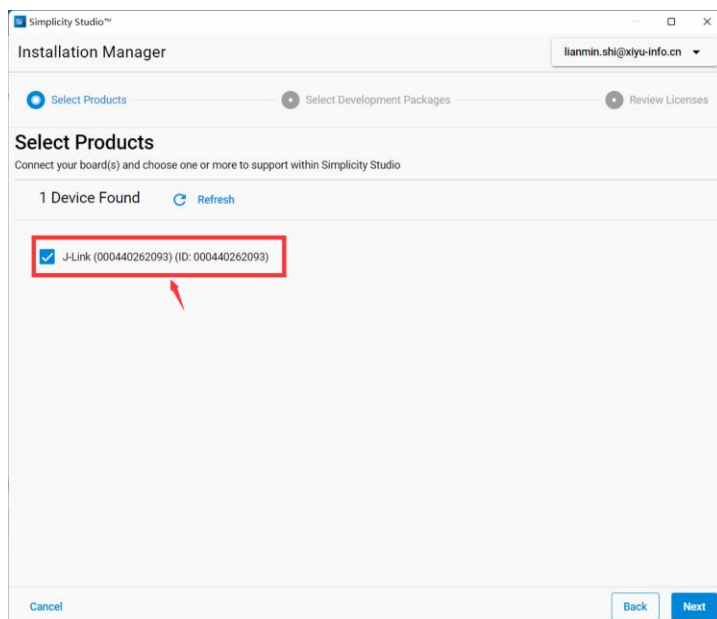



图 10 已连接的设备（J-Link 方式连接）

点击 Next  按钮，选择 Package 安装选项 **Package Installation Options**，两种选项：（1）**Auto**：自动方式（自动安装所有 SSv5 推荐的开发包，建议初次使用者选择该选项）；（2）**Advanced**：用户可以自定义选择，安装所需的开发包。如图 11 所示。

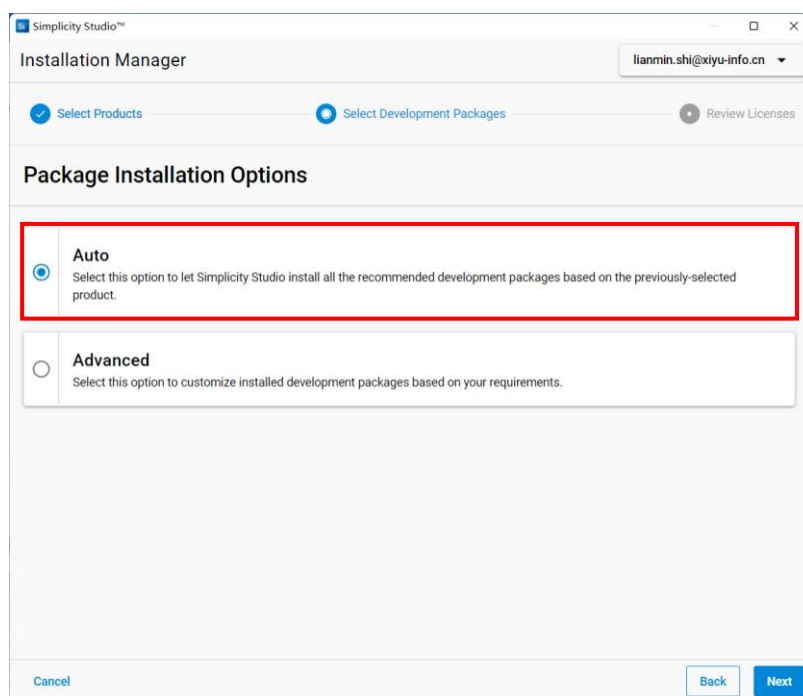


图 11 选择 Package 安装选项

点击 Next  按钮，自动下载安装，如图 12 所示。

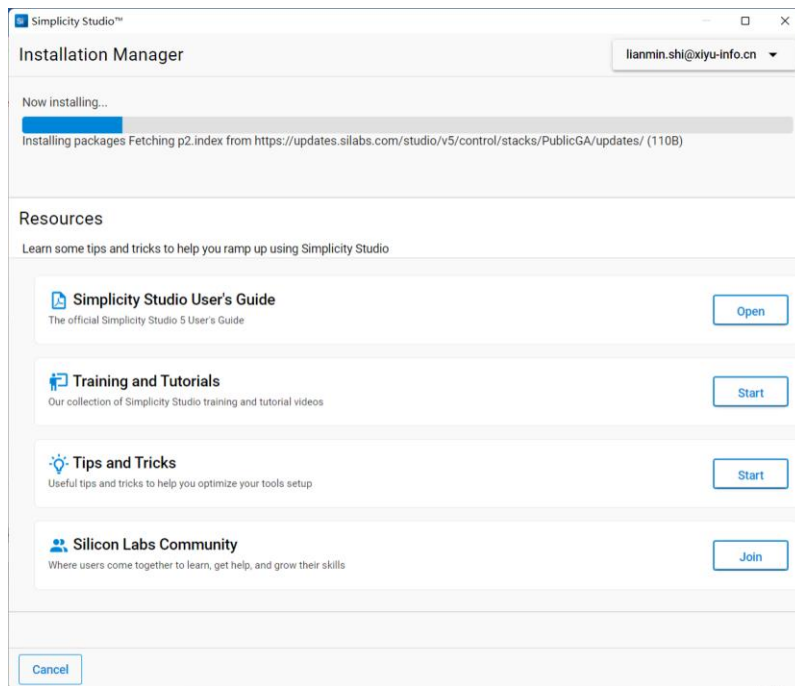
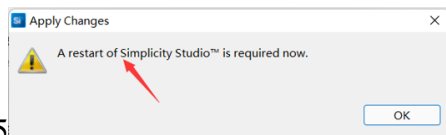


图 12 自动安装 Packages

Packages 安装完成后 ，需要重



启 SSv5，SSv5 重启后，初始页面如图 13 所示。

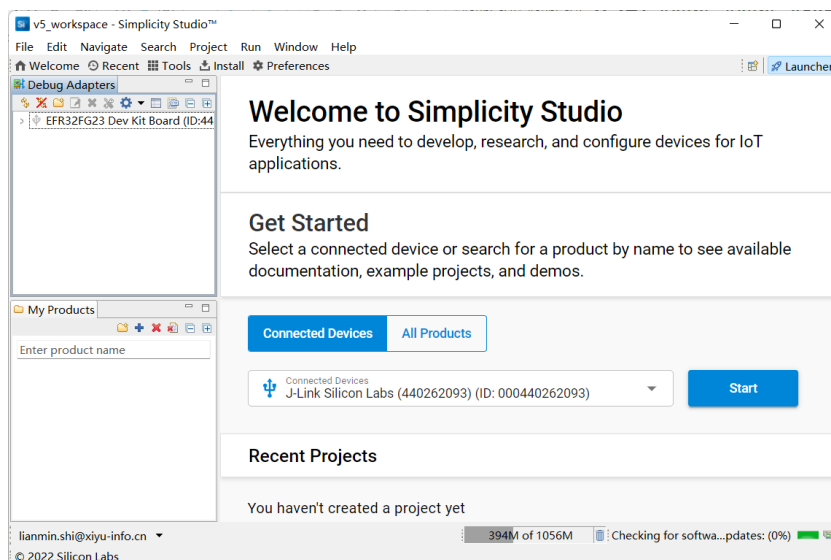


图 13 SSv5 启动后初始页面

SSv5 会自动检测已连接的设备

点击 **Start** 按钮，选择 **Connected Devices**（本次试用选择该选项）或者 **All Products**，

点击 **Start** 按钮，下载已连接的 EFR32G23 开发板的可用文档、例程、Demo 等。

下载完成后，在右下角的状态栏中显示文件大小 **616M of 732M**，双击 SSv5 左上角 **Debug Adapters** 下的 **EFR32FG23 Dev Kit Board (ID:440262093)**，展示 EFR32G23 开发板的相关例程、Demo、文档和兼容工具如图 14 所示。

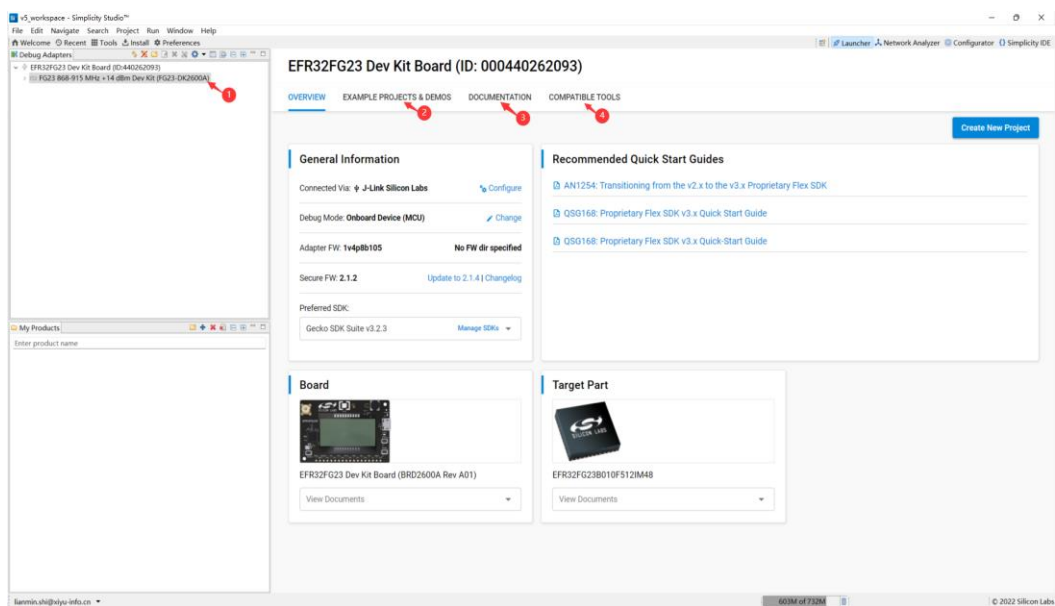


图 14 EFR32G23 开发板的相关例程、Demo、文档和兼容工具

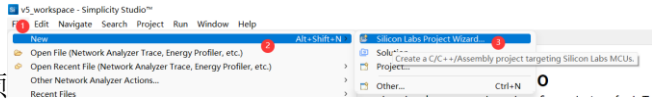
3 SSv5 和 Flex SDK 使用

SSv5 提供了多种使用示例应用程序开始工程项目的方法，在线 SSv5 用户指南 <https://docs.silabs.com/>和 SSv5 帮助菜单描述了这些方法。下面以使用 SSv5 中 File > New > Silicon Labs Project Wizard 方法新建项目。

说明：当用户在 SSv5 使用示例应用程序时，都需要执行以下步骤：

- 选择示例应用程序并创建项目；
- 修改并生成 Radio 配置；
- 必要时修改代码；
- 编译应用程序并将其下载到开发板。

3.1 安装 SDK

打开 Ssv5 后，依次点击菜单项 ，开始创建工程，在创建工程之前，先安装 SDK。选择目标板（Target Boards）、目标设备（Target Device）、SDK、工具链（Toolchain），如图 15 所示。

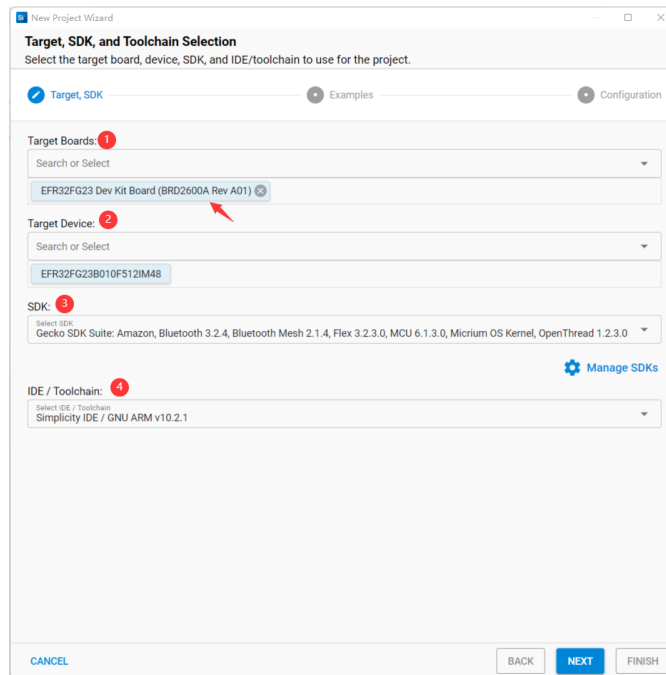



图 15 选择 Target Boards、Target Device、SDK、Toolchain

SSv5 会自动识别 Target Boards、Target Device、SDK、Toolchain，如果 SDK 没有正常安装，可以点击  **Manage SDKs**，检查可用于项目生成配置的 SDK，如图 16。

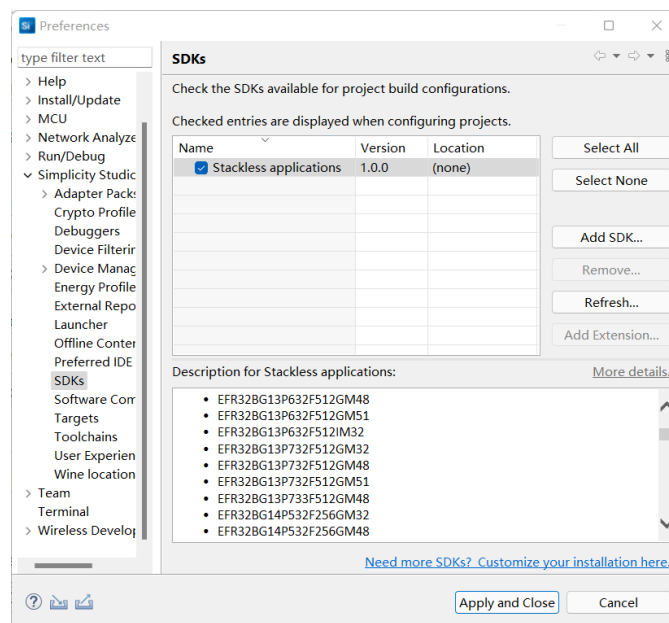


图 16 检查可用于项目生成配置的 SDK

点击 [Need more SDKs? Customize your installation here...](#)，安装其他项目所需的 SDKs。SSv5 提供了 3 种

方式，如图 17 所示。本次试用以  进行说明，其他方式的安装类似。

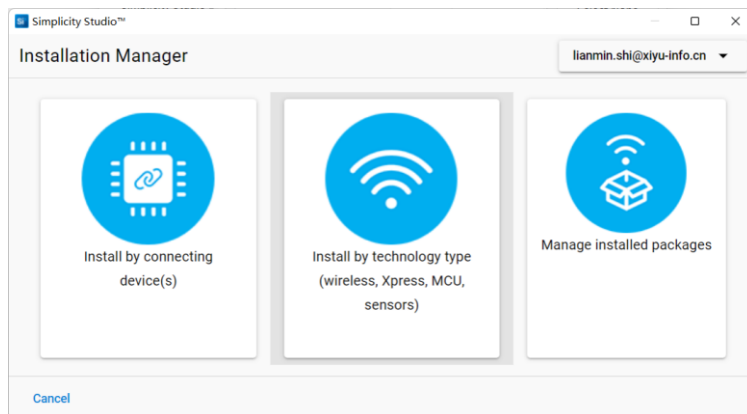
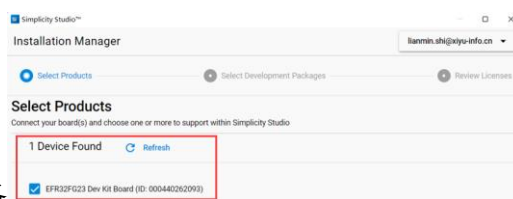




图 17 SDKs 安装方式选择

点击 ，SSv5 自动识别所连接的设备



点击  按钮，选择开发包 ，选择 Packages 安装选项（2 种选项：Auto 和 Advanced，此处选择 Advanced 方式，选择 Auto 加载较慢，可能跟网速有关），选择相应的开发包选项，如图 18 所示。

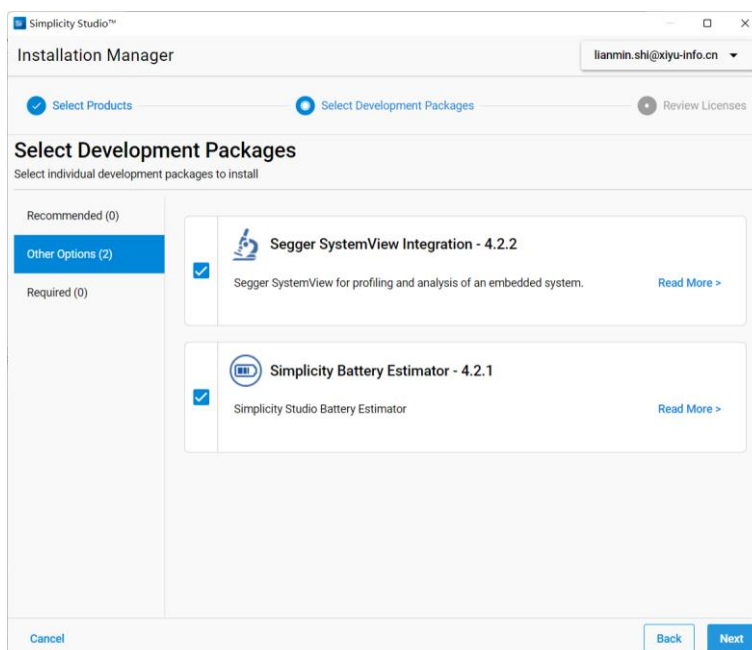


图 18 选择要安装的开发包

勾选可用的选项，点击 **Next**，进行安装，如图 19 所示。

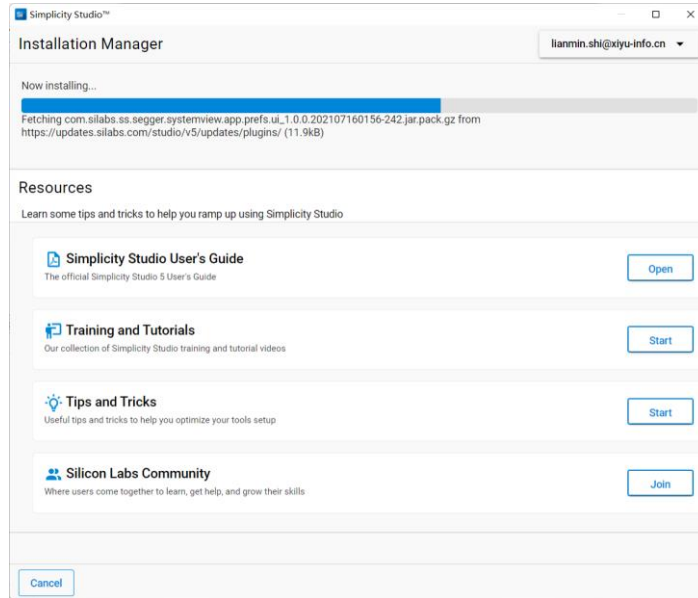


图 19 开发包安装过程

开发包安装完成后，需要重启 SSv5。

继续从图 15 开始，依次点击  **Manage SDKs**， **Need more SDKs? Customize your installation here...**，选择  **Manage installed packages**，
点击  **SDKs**，选择所需的 Gecko SDK 版本，勾选 **Agreements**，开始安装，如图 20 所示。

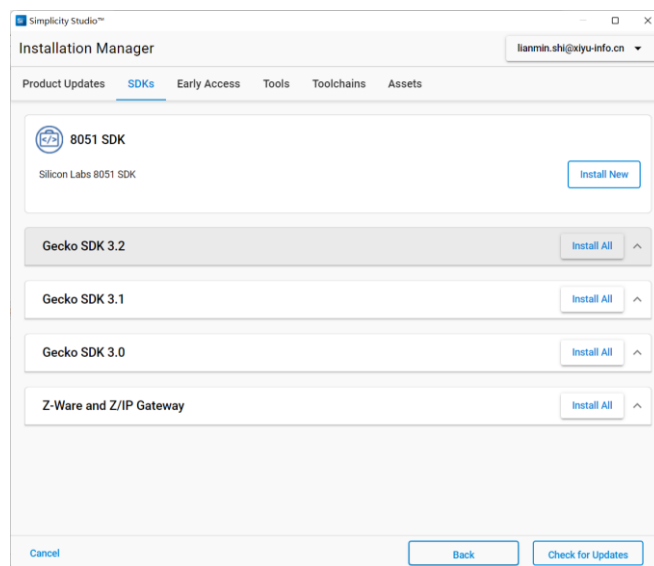


图 20 Gecko SDK 安装

Gecko SDK 安装过程如图 21 所示。

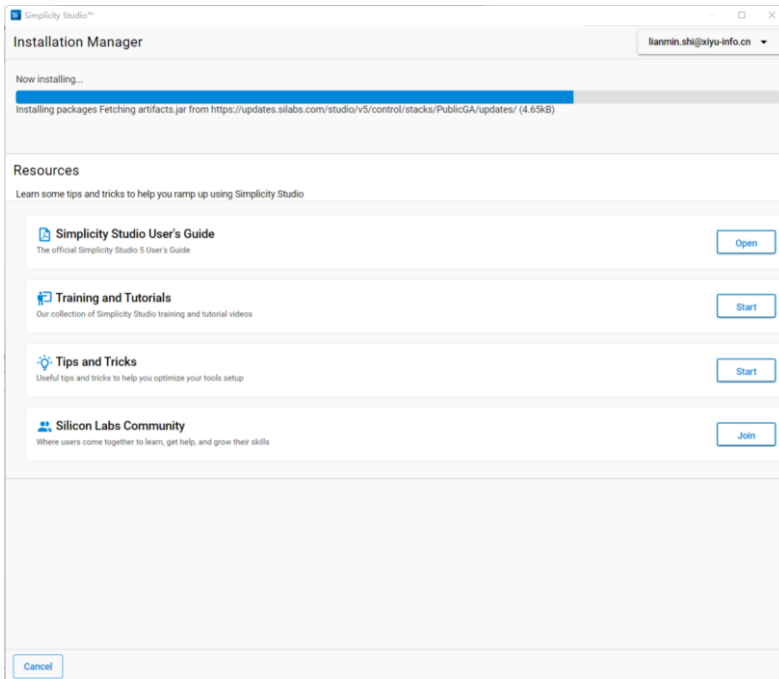


图 21 Gecko SDK 安装

Gecko SDK 安装完成后, 在 Target、SDK 和 Toolchain 选择框中就可以选择安装的 SDK, 如图 22 所示。

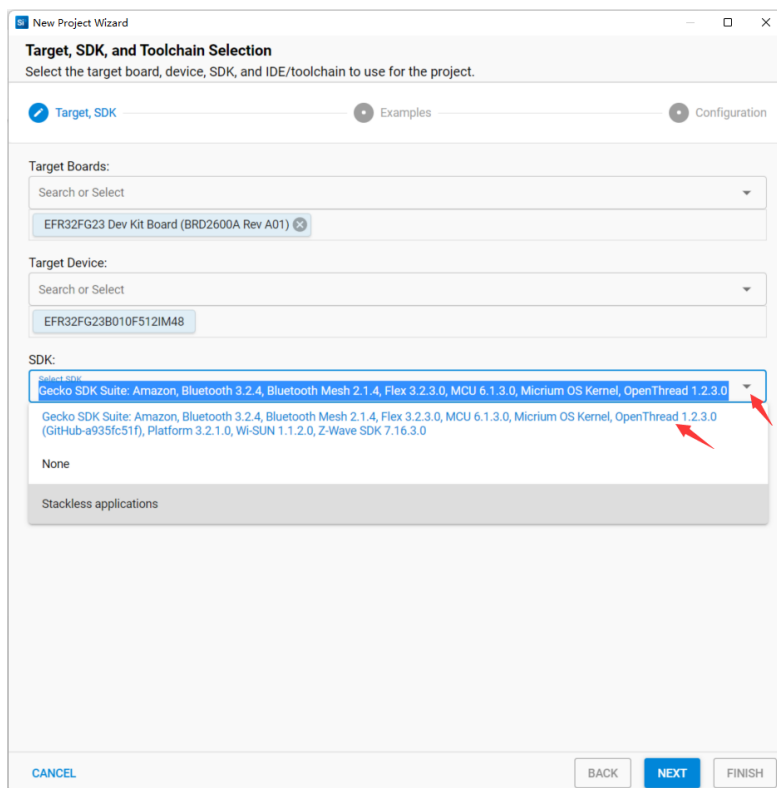


图 22 选择 Gecko SDK

除了通过 SSV5 直接安装 Gecko SDK, 还可以从 Silicon Labs 的 GitHub 上

(<https://github.com/SiliconLabs>) 下载安装，如图 23 所示。

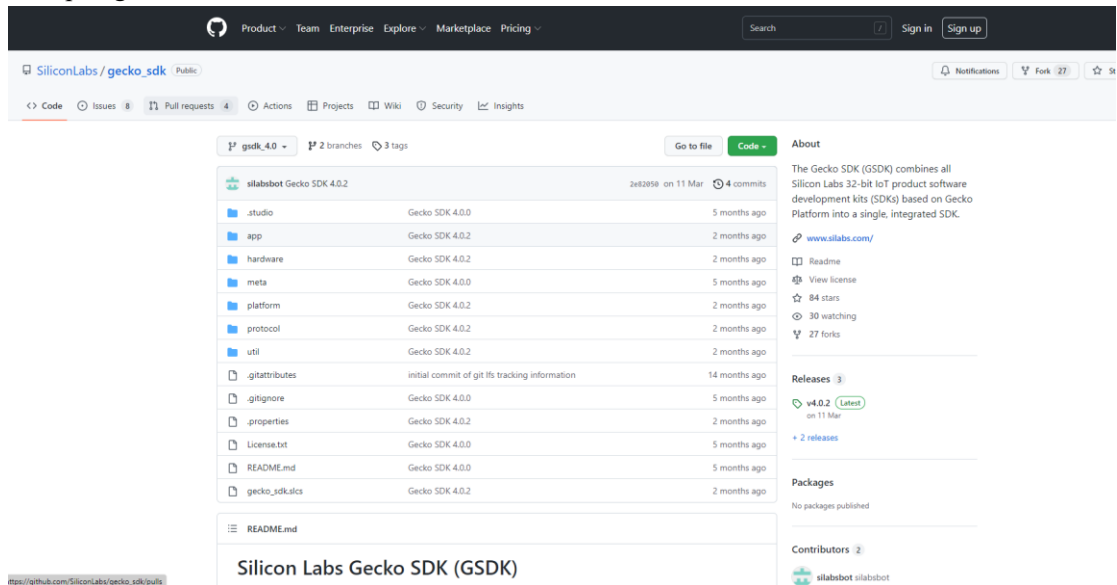
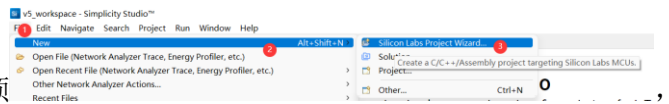


图 23 Gecko SDK (Silicon Labs GitHub)

3.2 创建工程

打开 SSV5 后，依次点击菜单项开始创建工程。



也可以直接在 EFR32FG23 Dev Kit Board 的 **OVERVIEW** 栏上点击 **Create New Project**，创建新的工程，如图 24 所示。

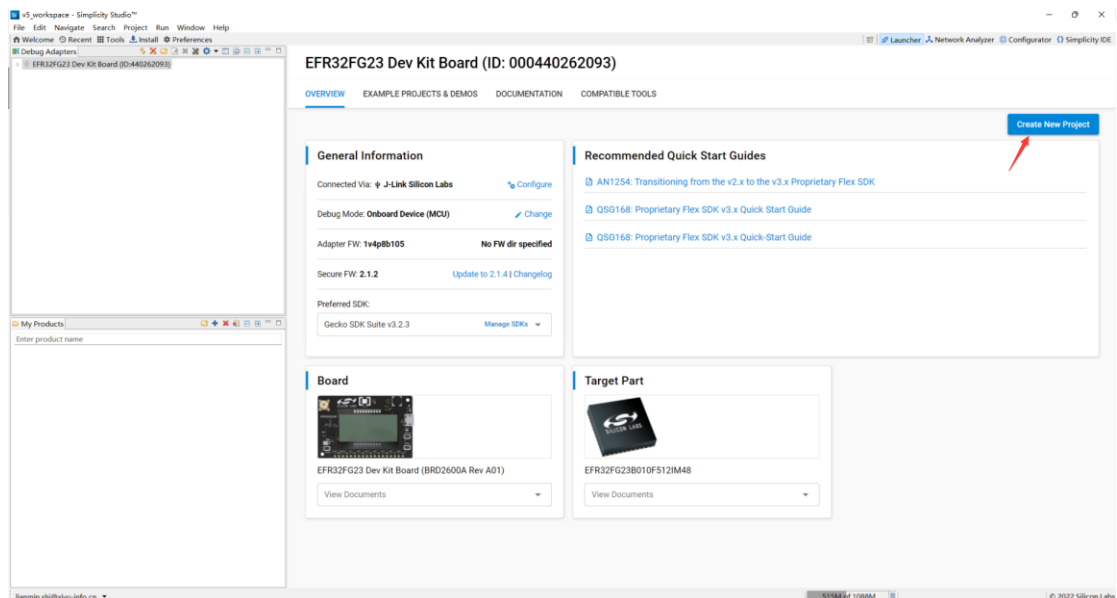


图 24 创建新的工程

选择相应的  Examples

作为模板，开始创建新的工程。在示例应用程序选择对话框

中，可以使用过滤条件快速定位应用工程模板。

本次试用选择 **Technology Type** -> Proprietary (9) -> 工程，如图 25 所示。

Flex (RAIL) - RAILtest
The RAILtest application (RAILtest) provides a simple tool for testing the radio and the functionality of the RAIL library via CLI. For more advanced...

作为示例

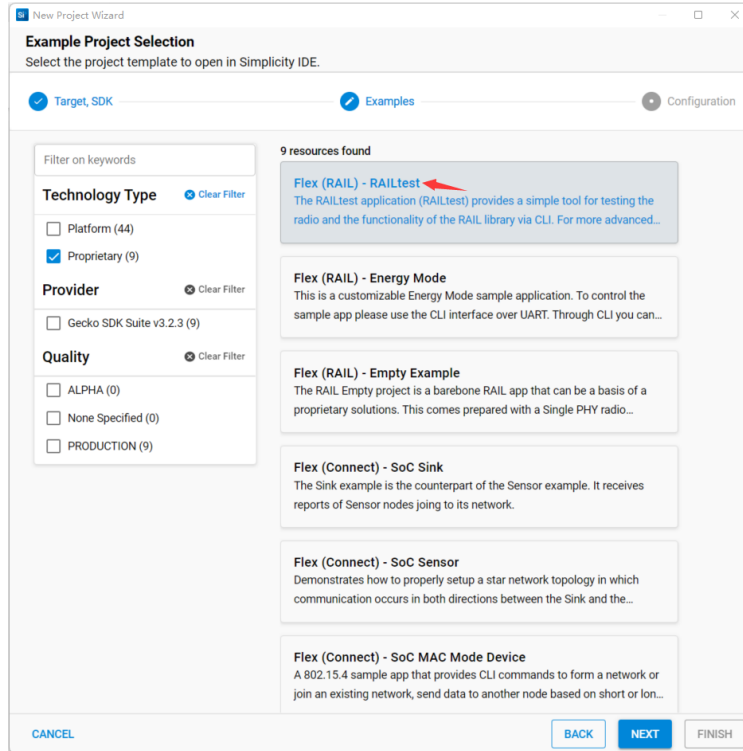


图 25 示例工程 (Examples) 选择

接着点击 **NEXT** 按钮，进行工程配置，包括工程名、工程文件位置等，如图 26 所示。

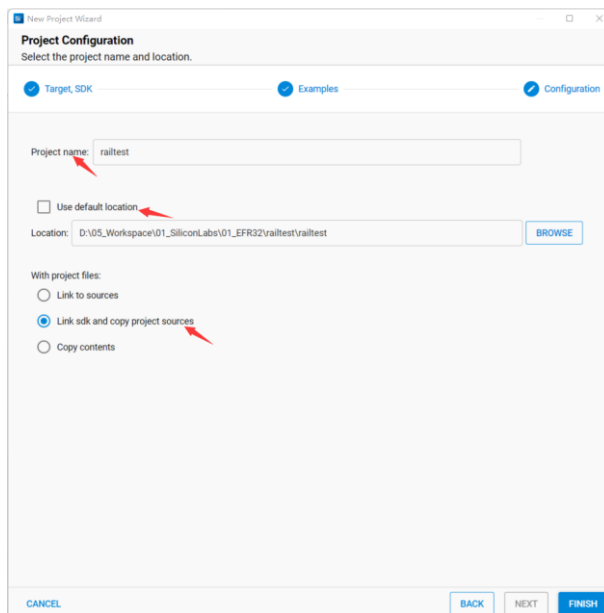
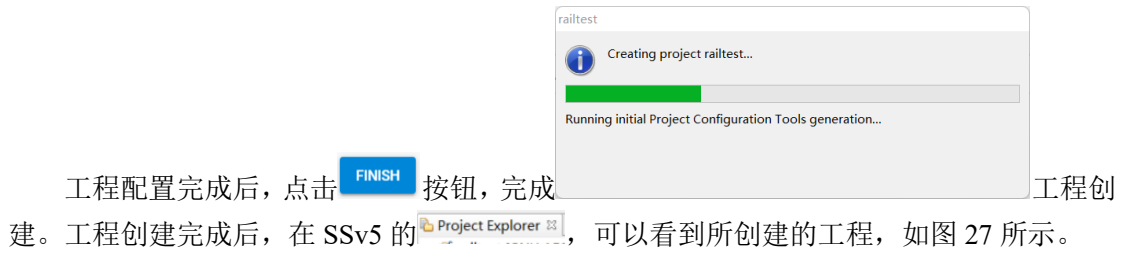


图 26 工程配置



工程配置完成后，点击 **FINISH** 按钮，完成工程创建。工程创建完成后，在 SSV5 的 Project Explorer 中，可以看到所创建的工程，如图 27 所示。

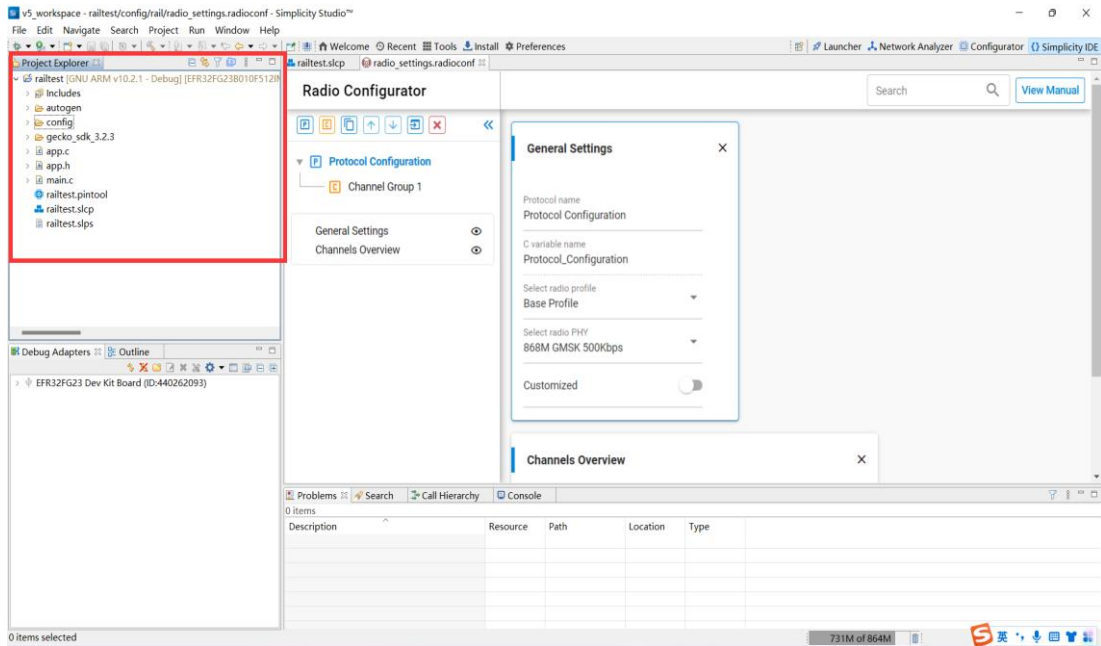


图 27 工程创建完成

工程结构说明如下：

- **autogen**：该文件夹包含生成的代码，它包括 PHY 配置（rail_config.c）、初始化代码、链接器脚本和组件使用的其他生成代码，如 CLI 接口的命令描述符。
- **config**：组件配置头文件（*.h）位于该文件夹中，可以使用 SSV5 IDE 组件编辑器或者直接编辑这些头文件。组件编辑器仅可用于可配置组件，它可以通过工程配置器的配置控件打开。
- **gecko_sdk_3.2.3**：包含由组件添加的源文件和二进制文件。

应用程序的特定文件通常放置于工程的根目录下，包括源文件（*.c）、工程配置器（*.slcp）、Pin 工具文件（*.pintool）。

所有的工程都包含主程序文件（main.c），官方建议不修改该文件。可以在 `app_init.c` 中增加初始化代码，在 `app_process.c` 中实现主循环。这样，系统组件可以初始化 components，并可以调用 components 的“进程(process)”功能。此外，启用实时操作系统 RTOS 时，将 app_process 的主循环转换为一个 RTOS 任务。

3.3 图形化工程配置器（*.slcp）

在 Project Explorer 视图下，双击工程配置器文件 railandtest.slcp，可以看到该文件的视图如图 28 所示。

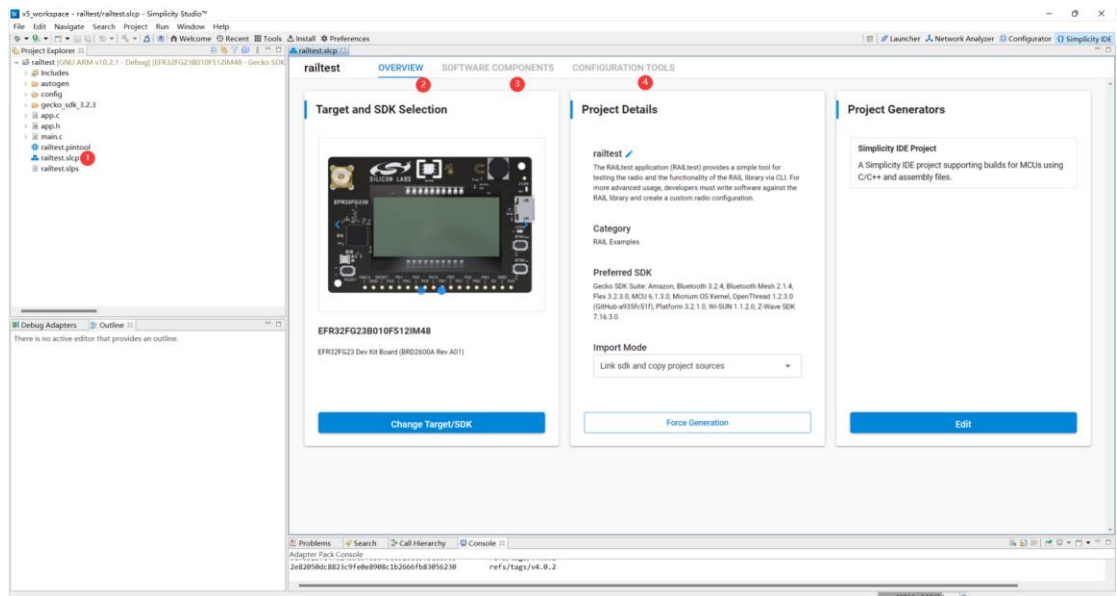


图 28 工程配置器文件 (*.slcp) 视图

该视图下有 3 个标签栏，分别为：[OVERVIEW](#)、[SOFTWARE COMPONENTS](#)、[CONFIGURATION TOOLS](#)。

- (1) [OVERVIEW](#)：选择目标板和 SDK、工程信息、工程生成器（Project Generators）。
- (2) [SOFTWARE COMPONENTS](#)：可以根据需要安装所需的软件组件，如图 29 所示。找到所需的组件后，点击 [Install](#) 按钮，自动安装。安装完成后，可以点击 [Configure](#)，通过图形化界面进行对安装的软件组件进行配置，也可以点击 [View Source](#)，编写源代码进行配置。

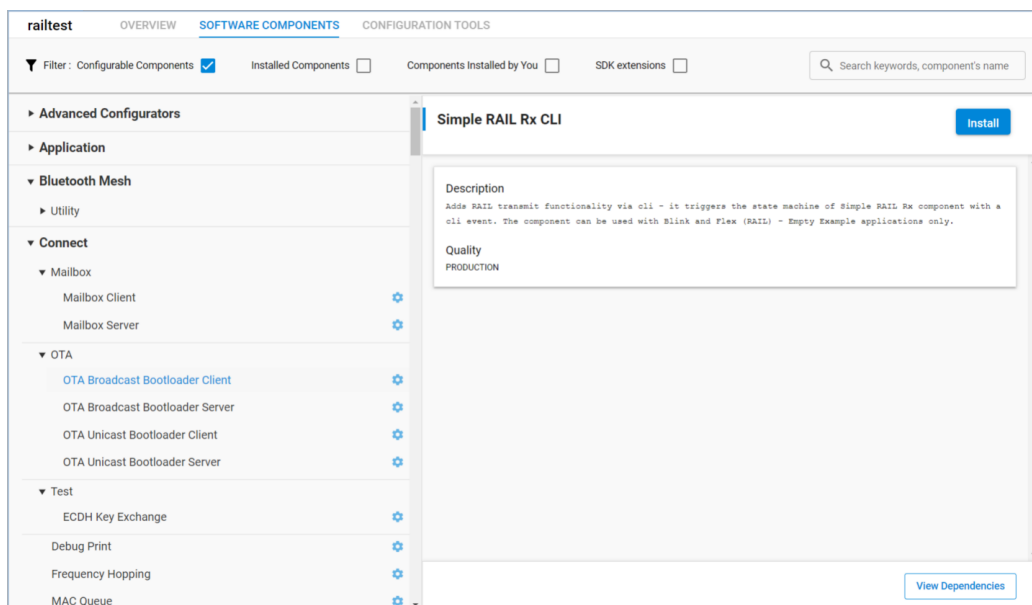


图 29 Software Components 安装

- (3) [CONFIGURATION TOOLS](#)：主要有 [Radio Configurator](#) 和 [Pin Tool](#)，如图 30 所示。

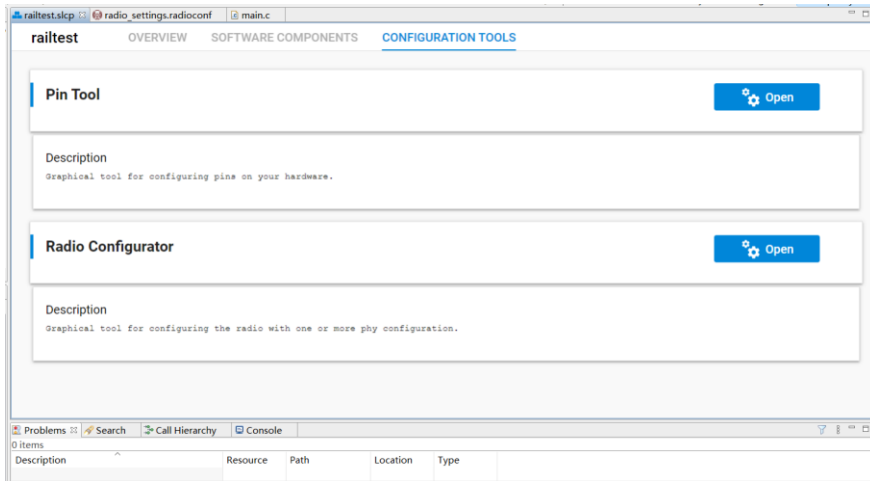


图 30 Configuration Tools

点击 **Radio Configurator** 下的 **Open** 按钮，进入图形化的 Radio Configurator 界面，用户可以直接配置 Radio 的各种参数，如图 31 所示。

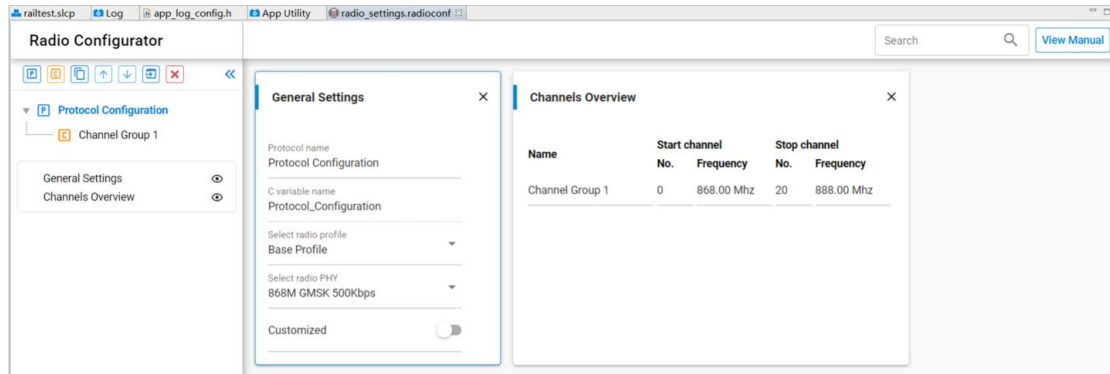


图 31 图形化 Radio Configurator 界面

如果对 Radio 的配置不清楚，可以点击图形化 Radio Configurator 界面右上角的 **View Manual** 。

Manual

AN1253: EFR32 Radio Configurator Guide

This document describes the Radio Configurator tool provided as part of Simplicity Studio® 5 (SSv5) for Proprietary applications. With the help of the Radio Configurator, users can create standard or custom radio configurations for their RAIL-based radio applications. This document explains the role of each item in the configuration.

If you are working with Proprietary SDK 2.7.n in Simplicity Studio 4, see AN971: EFR32 Radio Configurator Guide for RAIL in Simplicity Studio 4.

Proprietary is supported on all EFR32FG devices. For others, check the device's data sheet under Ordering Information > Protocol Stack to see if Proprietary is supported. In Proprietary SDK version 2.7.n, Connect is not supported on EFR32xG22.

查看相关说明

点击 **Pin Tool** 下的 **Open** 按钮，可以查看引脚信息，如图 32 所示。

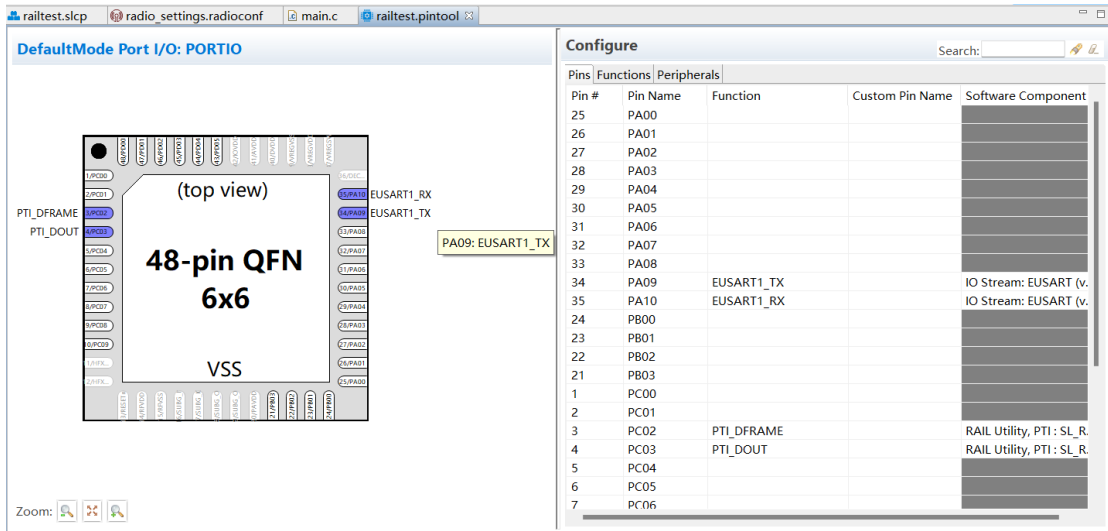


图 32 Pin Tool

用户可以在该图形化配置界面，将函数和对应的软件组件配置到相应的引脚上，例如：

引脚 PA09/10 的配置如下：

34	PA09	EUSART1_TX	IO Stream: EUSART (wcom): <input type="checkbox"/>
35	PA10	EUSART1_RX	IO Stream: EUSART (wcom): <input type="checkbox"/>

3.4 工程评测

1. main.c 文件

每个工程都有一个 main.c 文件，整个工程的程序入口：`int main(void)`。main.c 文件主要内容如图 33 所示。

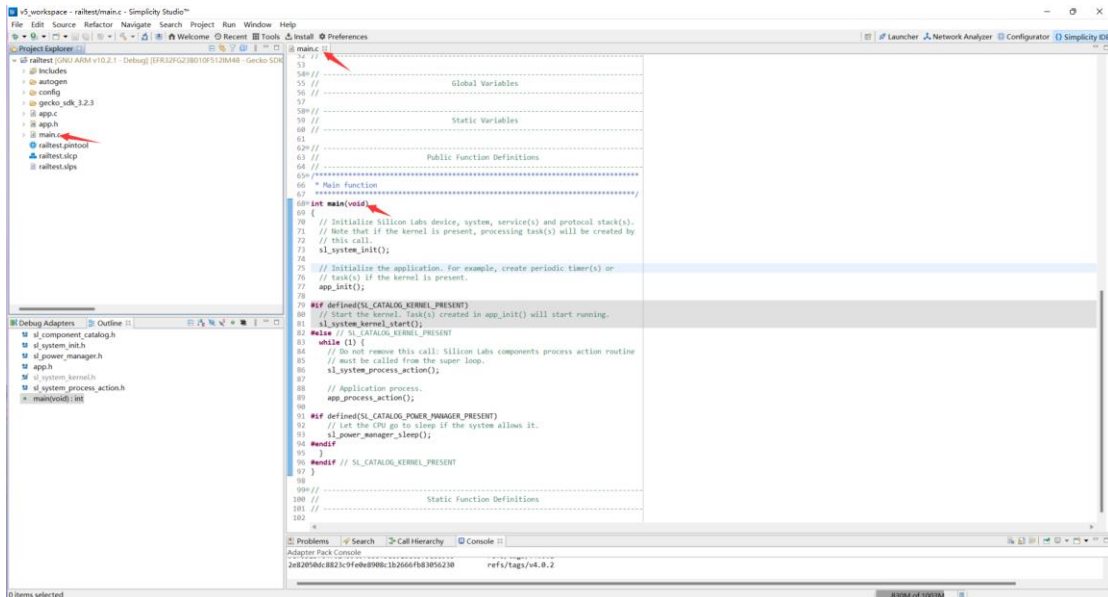


图 33 main.c 文件

说明：

Flex SDK v3.x 工程结合了 Gecko 平台基于组件的体系结构，可以通过组件库安装和配置软件特性和功能，安装过程包括：

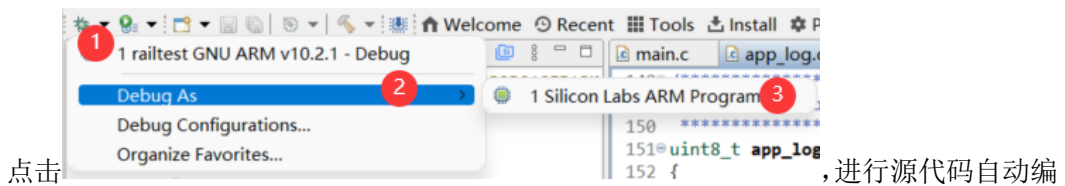
- (1) 将相应的 SDK 文件从 SDK 文件夹复制到工程文件夹中；

- (2) 将给定组件的所有依赖项复制到工程文件夹中；
- (3) 向工程设置中添加新的 include 目录；
- (4) 将配置文件复制到/config 文件夹中；
- (5) 修改相应的自动生成文件，将组件集成到应用程序中。

“init”类型的软件组件将实现给定组件的初始化代码，利用其相应的配置文件作为输入。一些软件组件（如：Parent support）将完全集成到应用程序中，以执行特定任务，而无需任何其他代码，而其他组件则提供应用程序中使用的 API。

通过安装和卸载组件以及配置已安装的组件来配置工程。

2. 编译、下载



下载，编译过程中会发现我们选择的 SDK 固件版本（SE firmware (2.1.4)）与目标板上的固件版本（SE firmware (2.1.2)）不匹配，如图 34 所示。

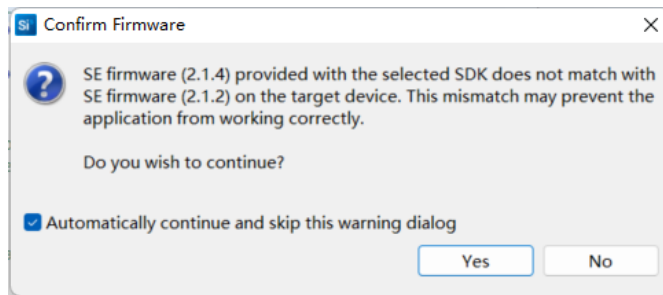


图 34 固件版本不匹配提示

点击 Yes 按钮后，固件将通过 J-Link 下载到目标板上，如图 35 所示。

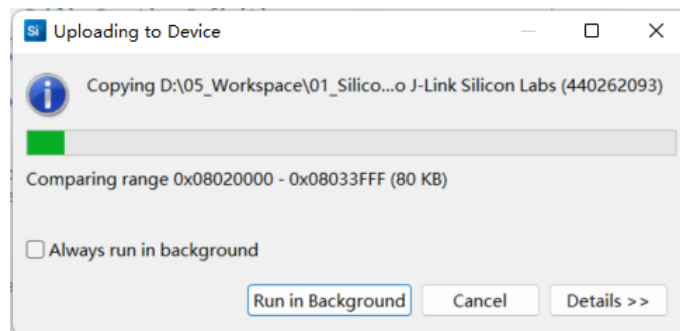
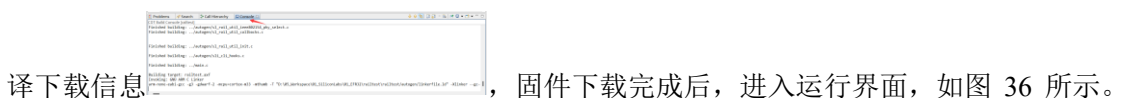


图 35 固件下载

固件下载进度现在 SSv5 的右下角 Launching railtest G...Debug: (4%)，同时，在 Console，显示相关编译



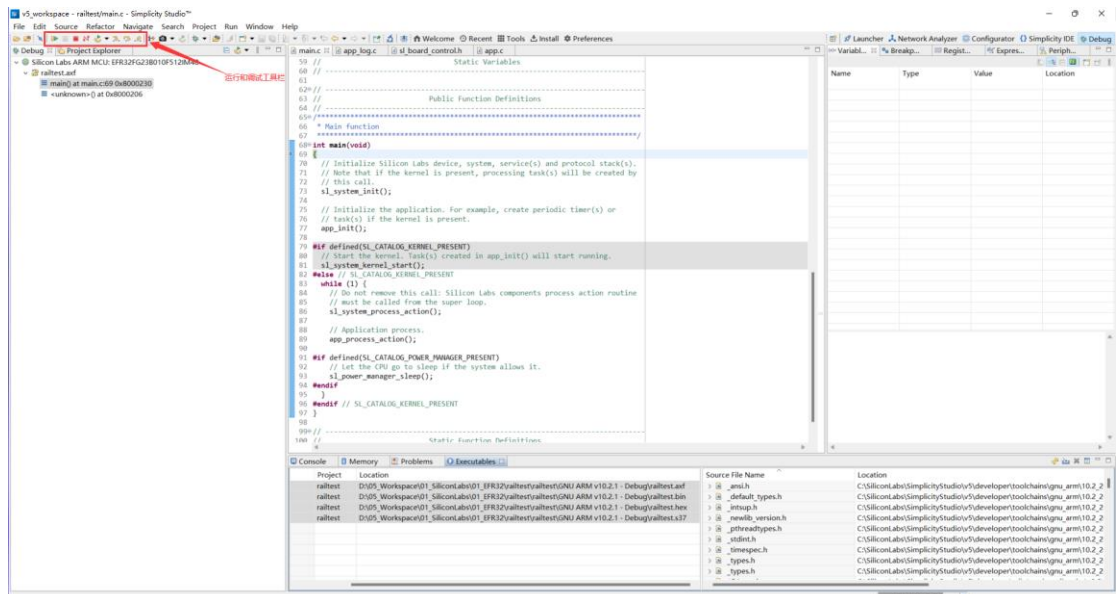


图 36 运行界面

固件编译完成后，在工程透视图中会生成编译文件夹及相关的机器码文件等（此处选择的 Toolchain 为 **Simplicity IDE / GNU ARM v7.2.1**），如图 37 所示。

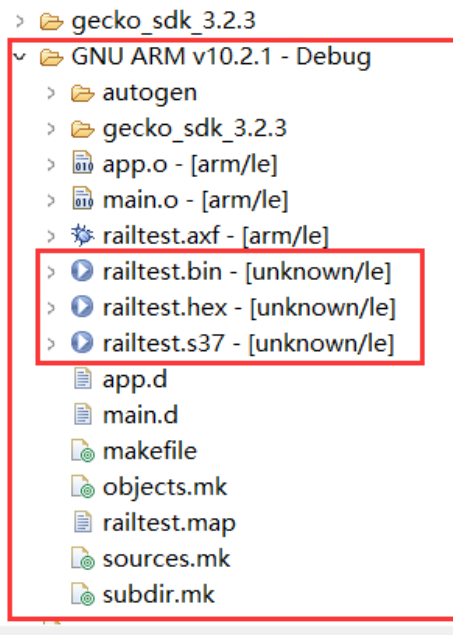


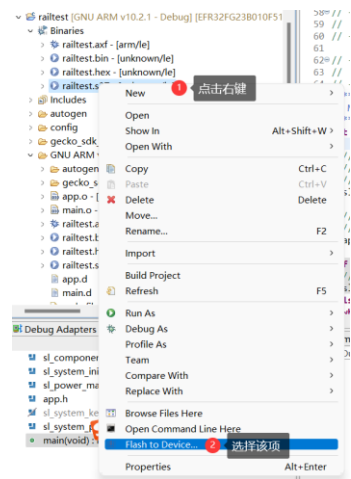
图 37 编译后生成的文件夹及相关的机器码文件等



固件下载到 EFR32FG23 开发板，点击 Resume 按钮运行，类似运行效果如图 38 所示，开发板通过 Micro-USB 线与 PC 电脑连接。



图 38 固件运行

如果要退出 Debug 模式，可以点击 Disconnect  按钮。



也可以手动下载固件，选择固件点击右键，选择  Flash to Device，
打开  Flash Programmer 工具，如图 39 所示。

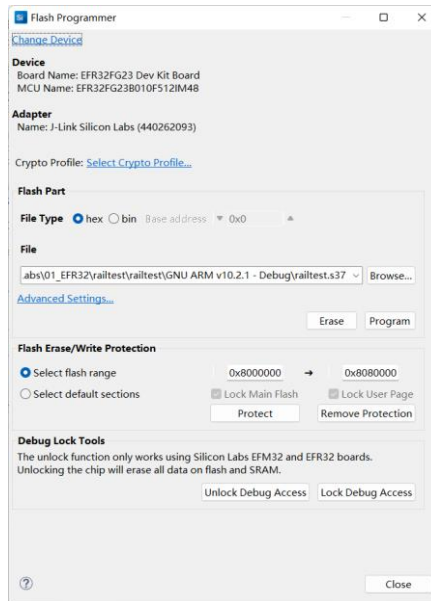


图 39 Flash Programmer

根据不同的示例应用程序，用户可以使用 CLI（Command Line Interpreter，命令行解释器）通过开发板上的按钮、LED 和 LCD 以及开发环境的控制台接口与之交互。

在 SSv5 IDE 透视图中，右键点击调试适配器视图中的调试适配器，启动 Console 接口



，如图 40 所示。

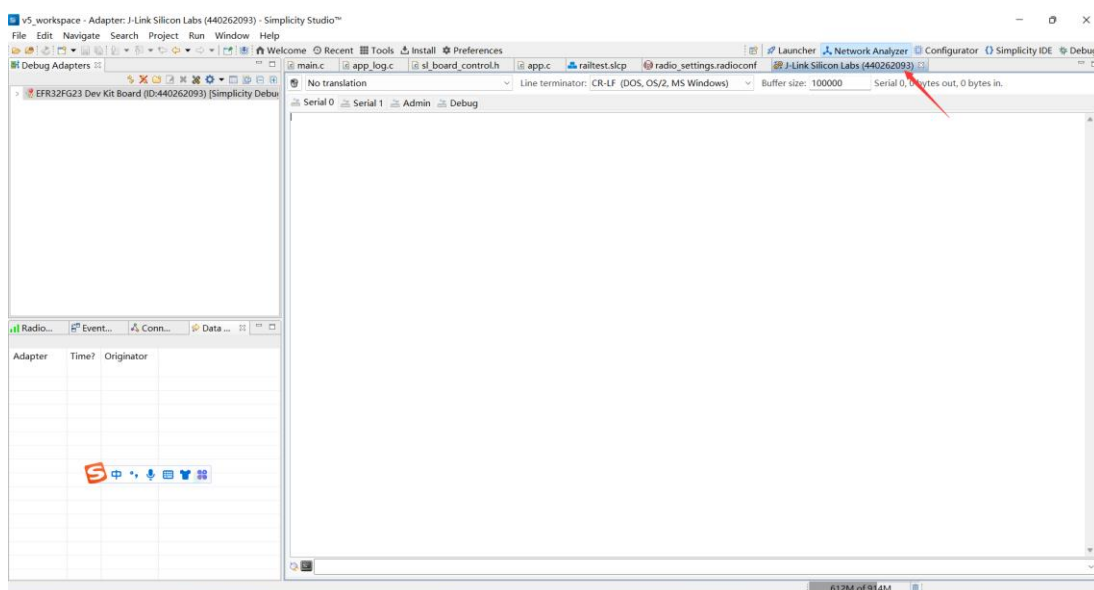


图 40 Console Interface

在 RailTest 示例程序中，Console Interface 允许测试 RAIL 的任何功能。输入“help”可以获取可用的 CLI 命令列表，如图 41 所示。

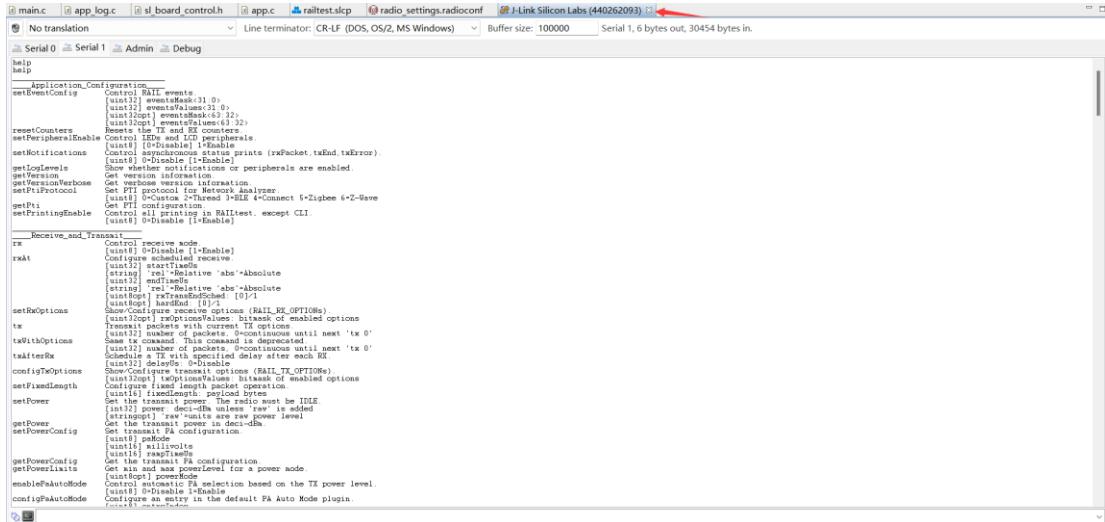


图 41 可用的 CLI 命令列表

在 Console 窗口输入 tx 1，该 Radio 节点将传输 1 包数据，然后进入 RX 状态，显示如下信息：

```
> tx 1
{{{(tx)}}{PacketTx: Enabled}{None: Disabled}{Time: 1070330200}}
{{{(tx)}}{PacketTx: Enabled}{PacketTx: Disabled}{Time: 1070334623}}
{{{(txEnd)}}{txStatus: Complete}{transmitted: 1}{lastTxTime: 1070334573}{timePos: 6}{lastTxStart: 1070334169}{ccaSuccess: 0}{failed: 0}{lastTxStatus: 0x00000000}{isAck: False}}.
```

目前，只有一个节点，无线数据传输的功能还无法实现。准备再购置一个节点，深度评测一下无线数据收发功能，并熟悉 RAIL 的所有功能。

4 Connect 和 RAIL

使用 Simplicity Studio 5 中的 Silicon Labs Flex SDK 可以为 EFR32FG23 开发板配置、构建和安装应用程序。Flex SDK 为应用程序的开发提供了 2 种途径：（1）使用 Silicon Labs RAIL (Radio Abstraction Interface Layer, Radio 抽象接口层)，RAIL 是一个直观、通用的 Radio 接口层，旨在实现任意的无线协议；（2）使用 Silicon Labs 连接协议栈 (Connect protocol stack)，该协议栈提供了一个功能齐全、易于定制的无线网络解决方案，并针对需要低功耗且用于简单网络拓扑的设备进行了优化。

无论用户是从 Connect 还是 RAIL 开始开发，都取决于用户在 Simplicity Studio 5 中选择作为起点的示例应用程序。Silicon Labs 提供的文档中建议，如果希望在不进一步开放的情况下包含以下功能，可以从 Connect 示例开始：

- （1）MAC 层功能，包含跳频和安全性；
- （2）网络生成，以及星型网络，路由支持；
- （3）应用级功能，诊断、I/O、邮箱、休眠终端设备管理；
- （4）Bootloading，包括串行和广播或单播 OTA。

4.1 Connect

Connect 示例功能通过易于配置的组件提供，这些组件可以按需打开或关闭。

EFR32 的 Connect 功能是在 RAIL 库上实现的，并支持多种 Radio 调制、频率和数据速率的组合。该协议栈包括所有 MAC 层功能，如扫描和连接、点到点或星型网络的设置、设备类型，如休眠端节点、路由器或协调器、Radio 配置、跳频和 LBT (Listen Before Talk) 协

议，这些协议是每个地理区域合规所需的，以及每个区域的 PHY 配置。所有这些功能都已经在堆栈中实现，开发人员可以专注于他们的应用程序开发，而不必担心较低级别的 Radio 和网络细节。

Flex SDK 包含很多 Connect 示例应用，例如：

- Flex (Connect) SoC-Empty: 最小的 Connect 工程结构；
- Flex (Connect) SoC-Empty Example DMP: 动态的多协议最小工程结构，可以用作运行 Connect 和 Bluetooth 协议的自定义应用程序的起点。
- Flex (Connect) SoC-Direct mode Device: 演示范围内节点之间的直接通信，网络参数由应用程序设置。
- Flex (Connect) SoC-MAC Mode Device: 演示范围内节点之间的直接 MAC 模式通信。
- Flex (Connect) SoC-Sensor 和 Flex (Connect) SoC-Sink: 演示了如何建立星形网络拓扑，在这种拓扑中，接收器和传感器节点之间会发生双向通信。

4.2 RAIL

RAIL 提供了一个直观、通用的 Radio 接口层，旨在支持专有的或标准的无线协议。RAIL 以库的方式提供，用户可以将其链接到自己的应用程序中。RAIL API 记录在一个在线 API reference 中，可通过 Simplicity Studio 获得，网址为 <https://docs.silabs.com/>。

RAIL 库支持的 APIs 包括

- 常用的 Radio 操作
- 通道定义和选择
- 输出功率配置
- 数据发送
- 发送前空闲信道评估
- 定时发送
- 能量检测
- 数据接收
- 数据包过滤
- 校准
- 载波 (CW: Carrier Wave) 发送
- 调制发送
- 作为唤醒源的 RFSense 配置

Flex SDK 包含 RAIL 应用程序示例源代码，用于演示设备和 RAIL 库的功能，具体示例源代码如下：

(1) Flex (RAIL) -RAILtest: RAIL 库的通用测试工具，在实现每个 RAIL 库功能时，会增加一条 RAILtest 串行命令，并允许脚本测试和 ad hoc 实验。用于实验室评估的 RAILtest 串行命令，包括：

发送和接收数据包；

- Schedule 在 RAIL 时基中的特定发送；
- 配置 RAIL 地址过滤去接收指定的数据包；
- 使能 CCA 机制 (CSMA/LBT) 以在数据发送前验证信道是否通畅；
- 在 RAIL 时基中设置定时器回调以查看 RAIL 定时器 API 工作机制；
- 改变当前配置频段内的发送信道；
- 改变发送功率等级；

- 在 2.4GHz 或 Sub-GHz 频段上启用指定持续时间的 RF 能量感应，并在该事件发生时唤醒；
- 为调试输出一个连续的未调制 tone；
- 为调试输出一个连续的调制 PN9 流；
- 当使用异步 GPIO 作为输入和输出发送和接收数据时，进入直接模式。

(2) Flex (RAIL) -Range Test: Range Test 示例支持用户定义参数的两个定制设备之间的空口测试，它被设计在 Silicon Labs 硬件上运行，而且不需要主机发出命令。

(3) Flex (RAIL) -Switch: 演示了 A light 和 A Switch 之间发送和接收操作的最简单交换，可与蓝牙/RAIL 多协议示例中的蓝牙 Light 应用一起使用，如 QSG155 所述：使用 Silicon Labs 动态多协议示例。

(4) Flex (RAIL) -Light: 演示了一个 RAIL Light 和一个 RAIL switch 之间最简单的发送和接收操作，Light 能定时并在状态变化时向 Switch 反馈其状态。这不是动态多协议 Light 示例应用程序。

(5) Flex (RAIL) -Simple TRX Multi-PHY¹: 演示了通过通道选择多个 PHYs 的使用。默认情况下，通道 0 被配置为 2.4GHz，250kbps，通道 1 被配置为 915MHz，500kbps，两个数据包都可通过使用正确预配置 PHY 的一个 PHY 应用程序接收。

(6) Flex (RAIL) -Simple TRX: 演示基于 RAIL 的最简单的发送和接收功能；

(7) Flex (RAIL) -Simple TRX with Auto-ACK: 演示了基于 RAIL 的两个节点之间最简单的交换传输和 ACK 操作。

(8) Flex (RAIL) -Long Preamble Duty Cycle: 演示了如何使用一个长前导实现 Rx 占空比循环；

(9) Flex (RAIL) -Burst Duty Cycle: 演示如何使用重复传输实现 Rx 占空比；

(10) Flex (RAIL) -Energy Mode: 演示 EFR32 低功耗模式 (EM0-Active, EM1-Sleep, EM2-Deep Sleep)；

(11) Flex (RAIL) -Empty Example: 最小的 RAIL 项目结构，用作用户应用程序的起点；

(12) Flex (RAIL) -WMBus Meter²: 实现无线 M-Bus 仪表应用程序；

(13) Flex (RAIL) -WMBus Collector: 实现无线 M-Bus Collector 应用程序。

参考资料：

[1] http://www.eccn.com/events/2022/silicon_chips/info.html

[2] [Silicon Labs Community \(silabs.com\)](https://siliconlabs.com/community)

[3] <https://github.com/SiliconLabs>

[4] <https://cn.silabs.com/developers/flex-sdk-connect-networking-stack>

[5] <https://docs.silabs.com/simplicity-studio-5-users-guide/latest>

¹ 详细信息参阅 AN971: EFR32 Radio 配置器指南。

² 详细信息参阅 AN1119: 将 RAIL 用于 EFR32 的无线 M-Bus 应用程序。AN1253: 用于 Simplicity Studio 5 的 EFR32 Radio 配置器指南。