



QSG137: Getting Started with the Silicon Labs Flex Software Development Kit for the EZR32

This quick start guide provides basic information on configuring, building, and installing applications for the EZR32 using the Silicon Labs Flex SDK (Software Development Kit). The Flex SDK includes Silicon Labs Connect, which provides a fully-featured, easily-customizable wireless networking solution optimized for devices that require low power consumption and are used in a simple network topology.

This guide is designed for developers who are new to the Silicon Labs Flex SDK, the Simplicity Studio development environment, and Silicon Labs development hardware. It provides instructions to get started using the Connect examples provided with the Flex SDK on the EZR32.

If you are working with the EFR32, see *QSG138: Getting Started with the Silicon Labs Flex Software Development Kit for the Wireless Gecko (EFR32™) Portfolio*. To see a table showing the Connect features supported on the EZR32 and the EFR32, see **Appendix: EFR32/EZR32 Comparison**.

KEY FEATURES

- Product overview
- Setting up your development environment
- Discovering your SDK
- Working with example applications.
- Using the Hardware Configurator
- EFR32/EZR32 Flex Feature Comparison

1 Flex SDK Product Overview

The Silicon Labs Flex SDK supports developers who wish to take advantage of configurable protocol functionality provided in Silicon Labs Connect.

This section covers:

- Background information on Connect and the example applications included with the Flex SDK
- Prerequisites for application development using the Flex SDK
- Support
- Documentation

The Flex SDK supports application development using Silicon Labs Connect, which provides a fully-featured, easily-customizable wireless networking solution optimized for devices that require low power consumption and are used in a simple network topology. Connect example functionality is provided through easily-configurable plugins that can be turned on or off as desired.

The following provides additional detail on Connect, including brief descriptions of the example applications. When you create a project based on an example, the description on the Simplicity Studio IDE General tab provides additional detail about the example and interfacing with it. Note that the empty-flex example has no plugins selected and therefore will not compile without additional configuration.

1.1 Silicon Labs Connect

Silicon Labs Connect supports many combinations of radio modulation, frequency and data rates. The stack includes all MAC layer functions such as scanning and joining, setting up of a point-to-point or star network, device types such as sleepy end nodes, routers or coordinators, radio configuration, frequency hopping and LBT (Listen Before Talk) protocols required for regulatory compliance in each geographical region, and PHY configuration for each of these regions. With all this functionality already implemented in the stack, developers can focus on their application development and not worry about the lower level radio and network details.

The Flex SDK includes the following Connect example applications:

Silicon Labs Connect Sensor Sample Application and **Silicon Labs Connect Sink Sample Application:** The sensor and sink applications demonstrate how to set up a star network topology in which communication occurs in both directions between the sink and the sensor(s) nodes. These are not supported on the EZR32HG.

Silicon Labs Connect Wire-Replacement Sample Application: Demonstrates point-to-point bi-directional direct or indirect communication between two nodes.

Silicon Labs Connect Commissioned Direct Device Sample Application: Demonstrates direct communication between nodes in range, whereas network parameters are commissioned by the application.

1.2 Prerequisites

Before following the procedures in this guide you must have

- Purchased one of the EZR32 Wireless Starter Kits, including BRD4001A - Wireless Starter Kit Mainboard, with one set of the following
 - BRD4544A – EZR32HG; Si4463; 169 MHz +20 dBm
 - BRD4542B – EZR32HG; Si4455; 434 MHz +10 dBm
 - BRD4545A – EZR32HG; Si4463; 490 MHz +20 dBm
 - BRD4546A – EZR32HG; Si4468; 868 MHz +16 dBm
 - BRD4543B – EZR32HG; Si4468; 915 MHz +20 dBm
 - BRD4502A - EZR32LG 330 Direct Tie WSTK Radio Board; Si4460; 868 MHz +13 dBm
 - BRD4502B – EZR32LG 330 Switched WSTK Radio Board; Si4455; 434 MHz +20 dBm
 - BRD4502C - EZR32WG 330 Direct Tie WSTK Radio Board; Si4460; 868 MHz +13 dBm
 - BRD4502D – EZR32WG 330 Switched WSTK Radio Board; Si4455; 434 MHz +20 dBm
 - BRD4503A - EZR32LG 330 Switched WSTK Radio Board; Si4463; 915 MHz +20 dBm
 - BRD4503B - EZR32WG 330 Switched WSTK Radio Board; Si4463; 915 MHz +20 dBm
 - BRD4505B - EZR32WG 330 Switched WSTK Radio Board; Si4463; 490 MHz +20 dBm
- Downloaded the required software components. A card included in your development hardware kit contains a link to a Getting Started page, which will direct you to links for the Silicon Labs software products. See the Flex SDK release notes for version re-

strictions and compatibility constraints for Connect and RAIL and these components. To develop Silicon Labs Connect-based applications, you will need the following:

- The Simplicity Studio v4 development environment, which incorporates AppBuilder (see **Installing Simplicity Studio**). AppBuilder is an interactive GUI tool that allows you to configure a body of Silicon Labs-supplied code to implement applications. Online help for AppBuilder and other Simplicity Studio modules is provided.
- The Silicon Labs Flex SDK, installed through Simplicity Studio.
- IAR Embedded Workbench for ARM (IAR EWARM) 7.80.2, used as a compiler in the Simplicity Studio development environment. Download the supported version of IAR from the Silicon Labs support portal. Refer to the “QuickStart Installation Information” section of the IAR installer for additional information about the installation process and how to configure your license.

Although you will not need it for the tasks in this Getting Started guide, you may wish to become familiar with the functionality available through Simplicity Commander. Simplicity Commander is installed along with Simplicity Studio in a subfolder of the install folder. See *UG162: Simplicity Commander Reference Guide* for more information.

1.3 Support

Users can access the Silicon Labs support portal at <https://www.silabs.com/support> or through the **Resources** tab in the Simplicity Studio Launcher perspective. Use the support portal to contact Customer Support for any questions you might have during the development process.

1.4 Documentation

Documentation provided with the Flex SDK can be accessed through Simplicity Studio. Simplicity Studio also provides links to hardware documentation and other appnotes, as described in section **Accessing Other Resources**. See the release notes for details on other documentation available.

2 Setting Up Your Development Environment

2.1 Install Third-Party Tools

Install third-party tools, such as IAR Embedded Workbench for ARM (see section [Software Components](#) for the list).

2.2 Connect your Hardware

Connect your WSTK, with radio board mounted, 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.

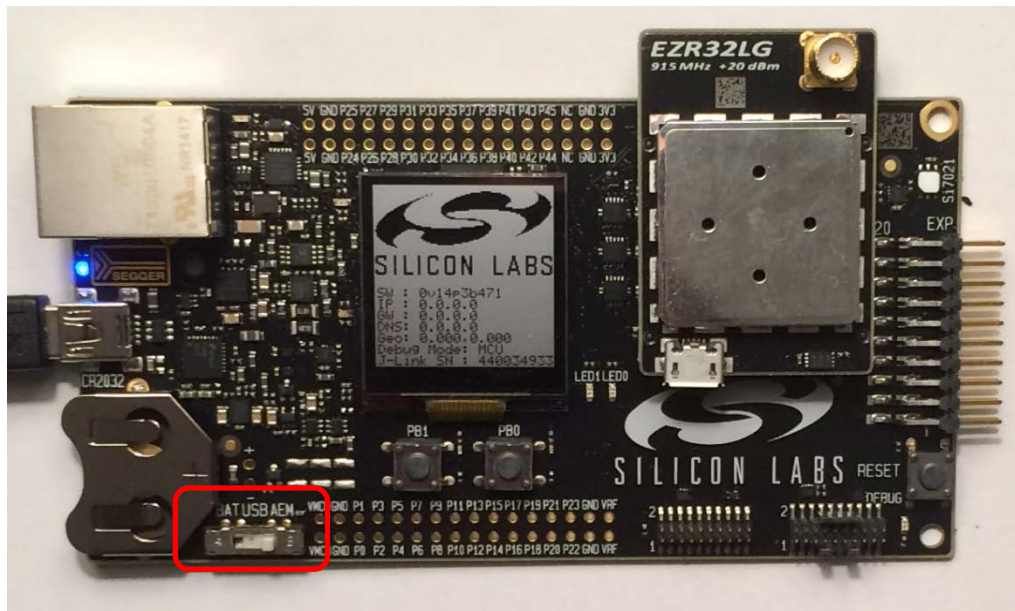
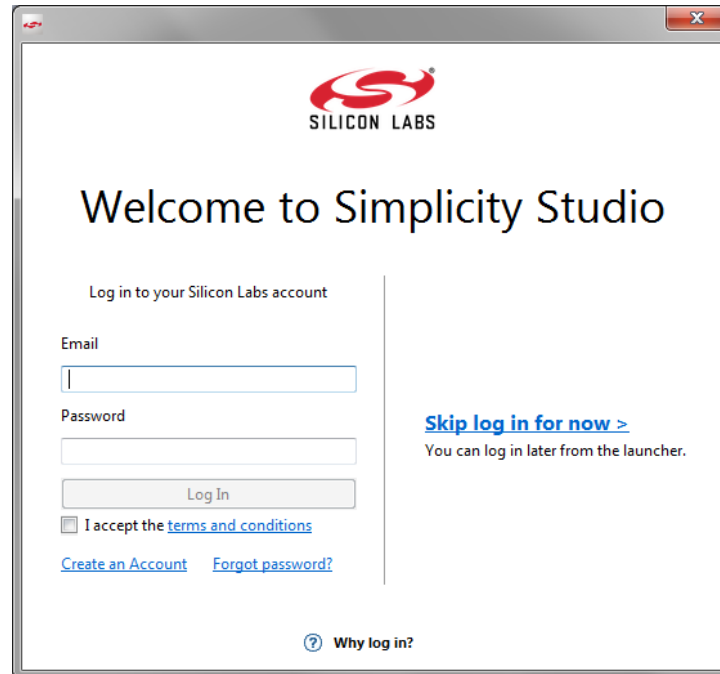


Figure 1. EZR32LG on a WSTK

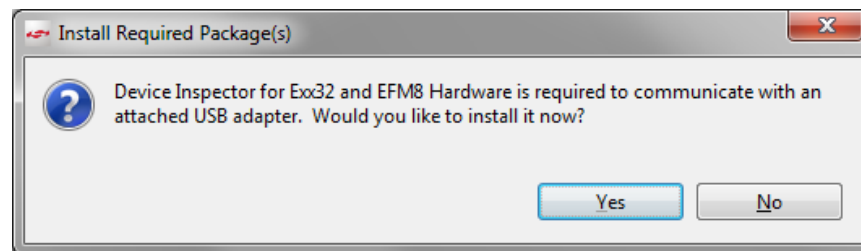
2.3 Install Simplicity Studio and the Flex SDK

1. Run the Simplicity Studio installation application.
2. When Simplicity Studio first launches, you are invited to log in. If you have already created a user account on the support portal, the login username and password is the same, or you can create an account here.

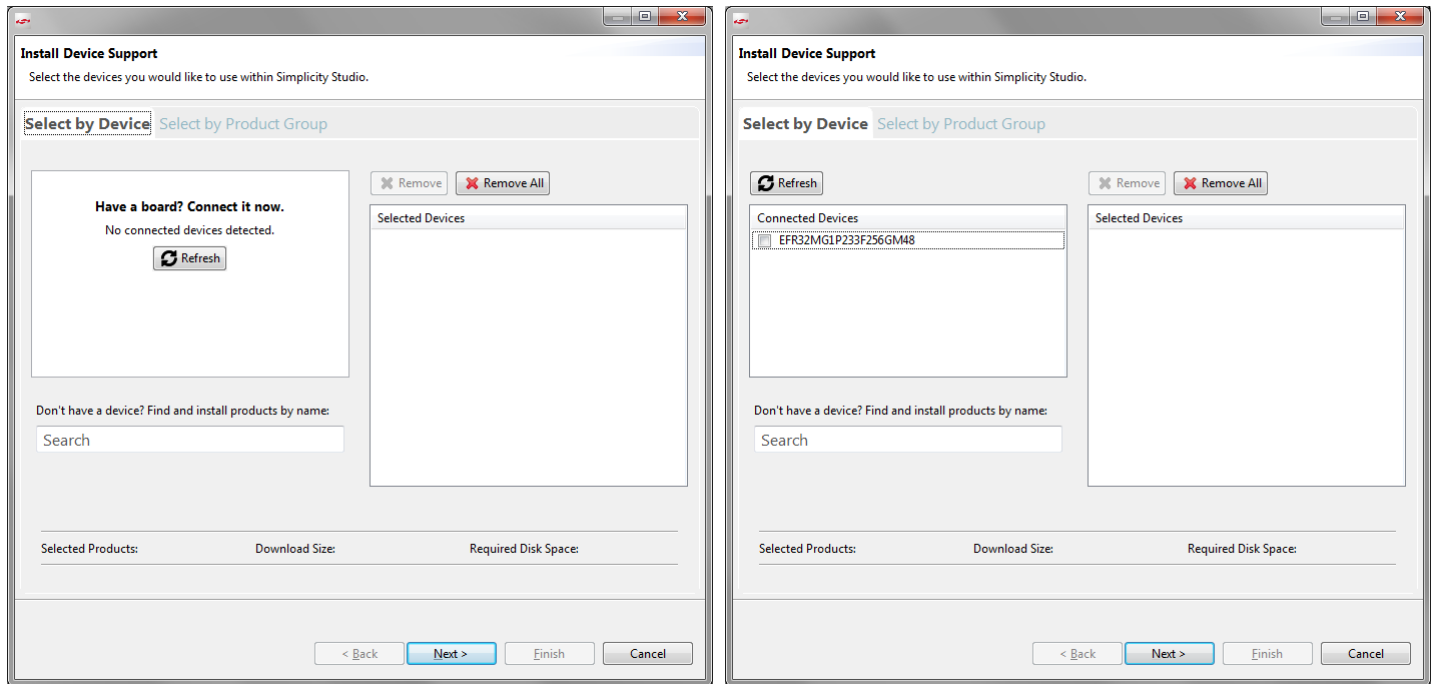
Note: Although some Silicon Labs stacks require that you have a valid Silicon Labs user account before they can be installed, the Flex SDK does not.



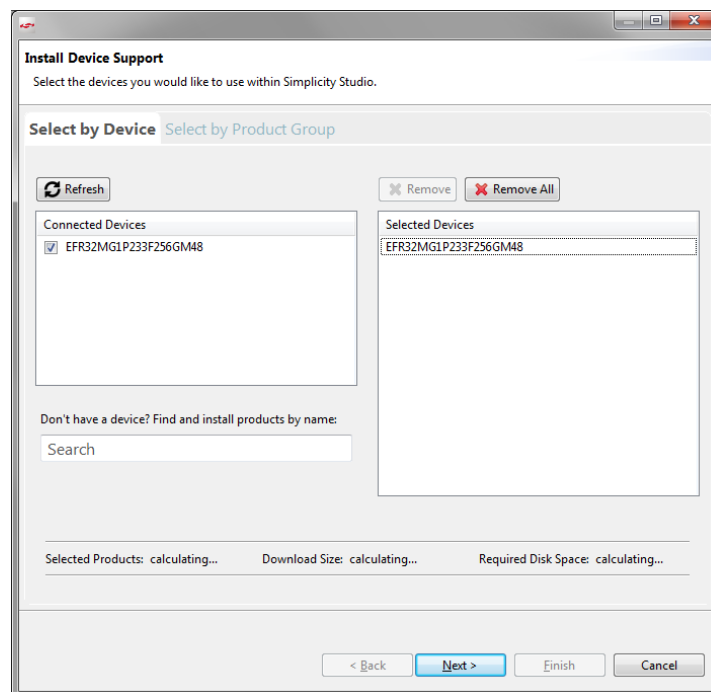
3. After login, Simplicity Studio adds software information. Once initial software installation is complete, Simplicity Studio checks for connected hardware. If you have the WSTK connected by USB cable, Simplicity Studio will detect the USB cable and prompt you to download a Device Inspector. Click **Yes**.



- 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 an example of the Install Device Support dialog before and after the connected device is displayed.

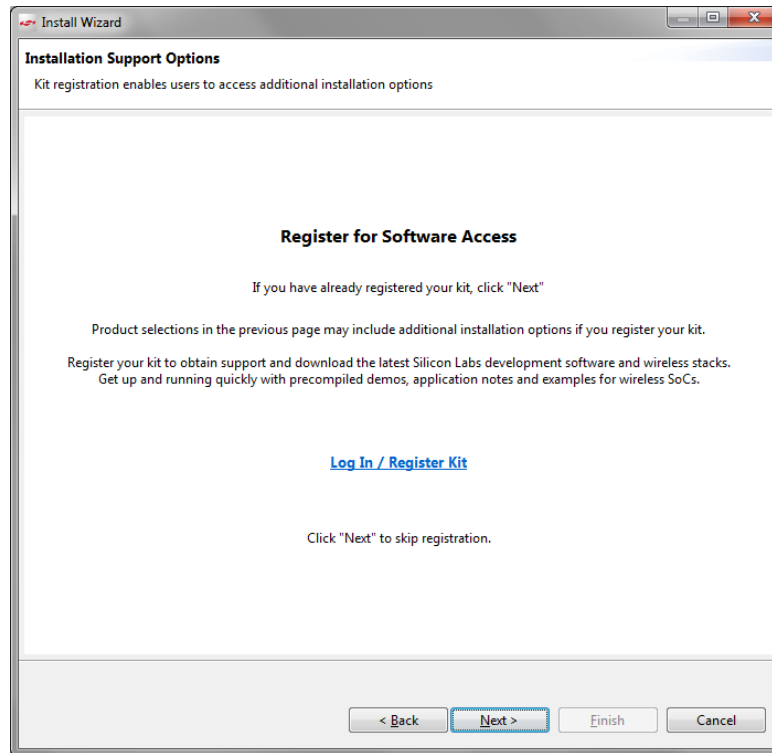


- Click the checkbox next to the device to select it. Selecting the device allows Simplicity Studio to present the relevant software packages for you to install. Click **Next**.

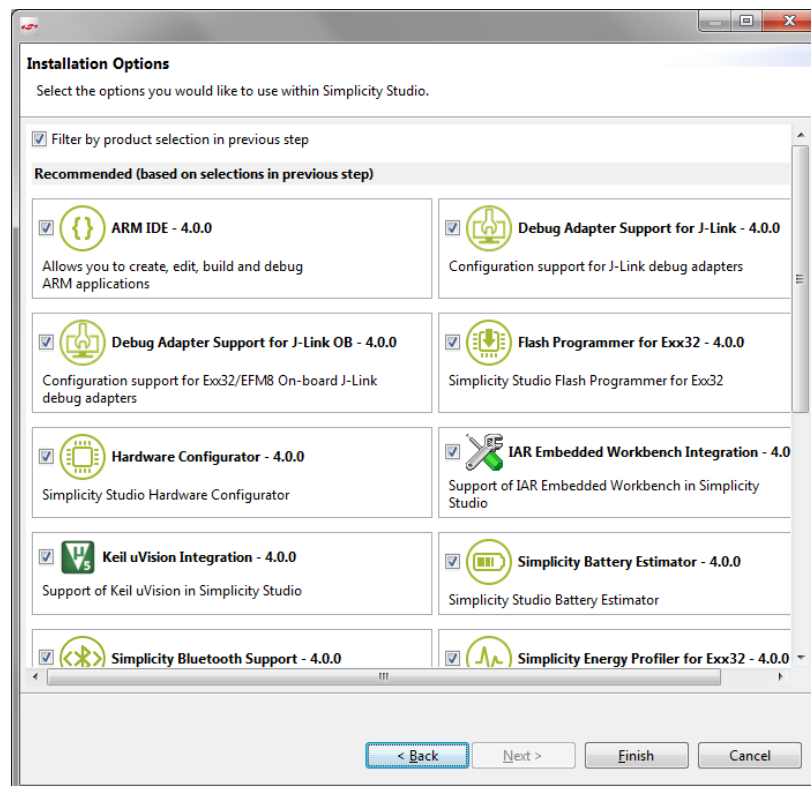


Note: You can also click the **Select by Product Group** tab to install device support for all devices in one or more product groups.

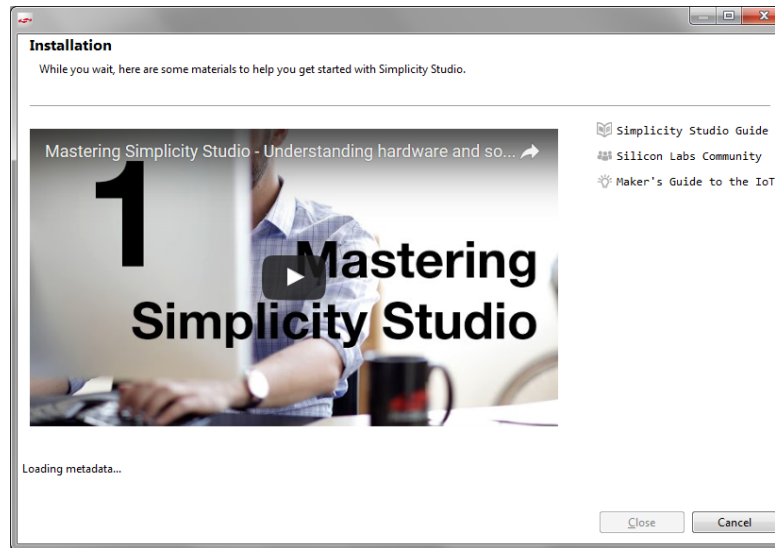
- A kit registration dialog is displayed. Unless you are also installing a stack, such as Silicon Labs, that requires registration, click **Next**.



- The **Installation Options** dialog shows the software packages that can be installed. By default the list is filtered by the product selection. To see all tools, uncheck the **Filter by product selection in previous step** box. In general, you will install all the selected tools. Also by default, the current versions of all Silicon Labs stacks that are compatible with this part are checked. Uncheck any you do not wish to install. Previous stack versions are shown under **Other Options**. Click **Finish**.



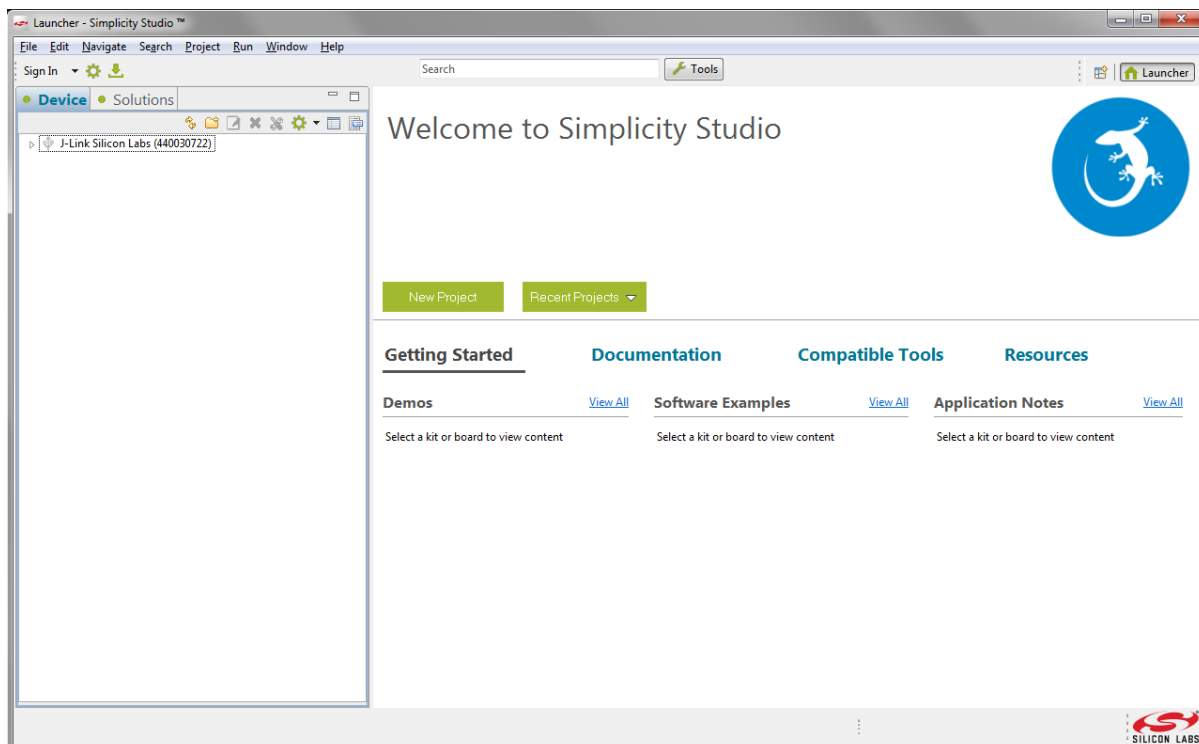
Installation will take several minutes. During installation, Simplicity Studio offers you viewing and reading options to learn more about the environment.



8. After installation is complete, restart Simplicity Studio.
9. 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**.



10. The Launcher perspective opens, but it is not yet fully populated. Expand the J-Link entry in the devices tab and click on the WSTK.



The Launcher perspective is then populated with the software components and functionality associated with your WSTK and stack, as described in the next section.

Before continuing with the example application, you may need to update your device firmware or change your preferred WSTK. These functions are described in sections [Updating Adapter Firmware](#) and [Changing the Preferred SDK](#), respectively.

Note: If you plan to work with NCP software communicating with a host MCU/PC, you should upgrade your adapter firmware to build 435 or later.

3 Functionality in the Launcher Perspective

In the Launcher perspective you can perform a number of functions. Additional information on some of these is provided later in the section.

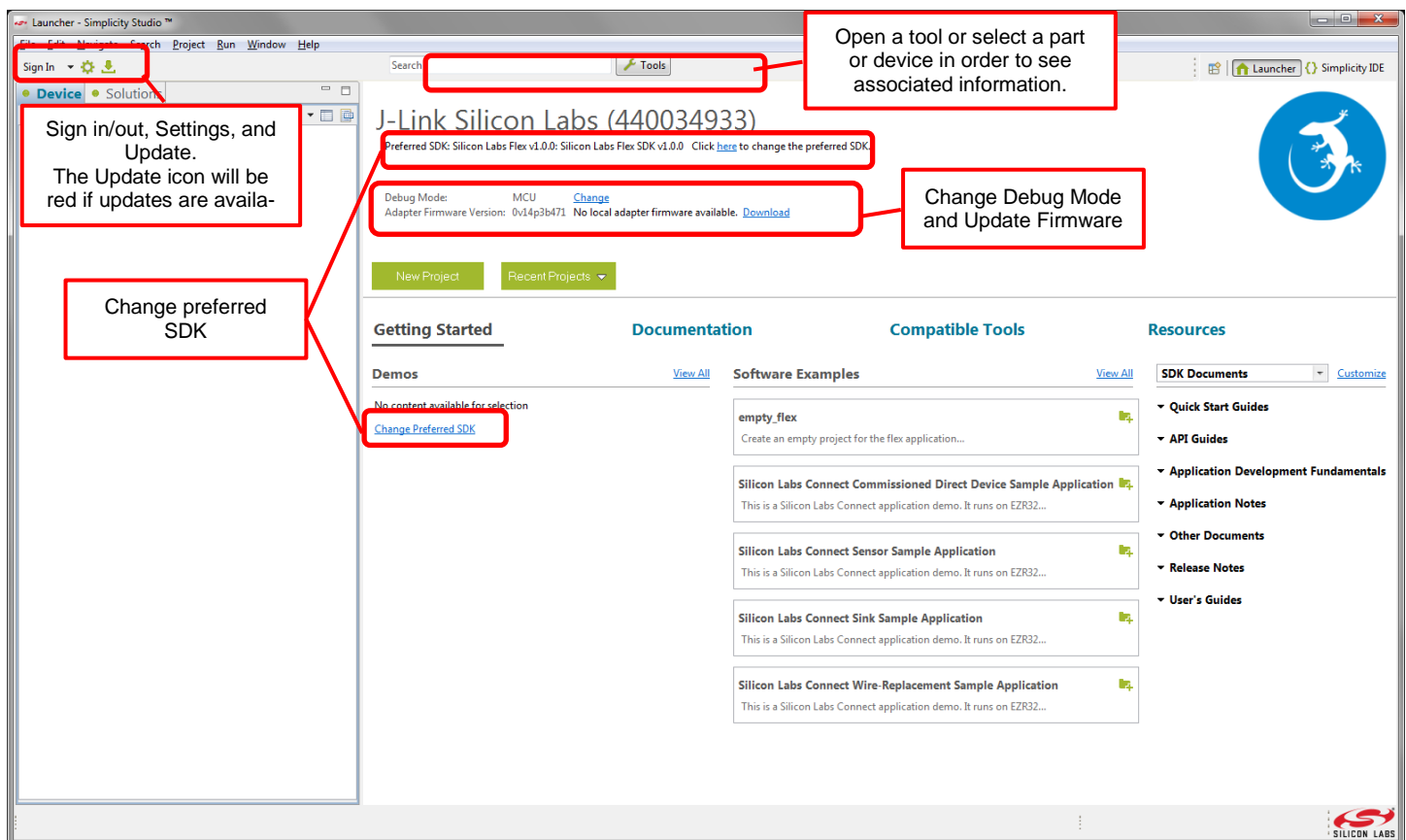
- Sign in (if you are not signed in already)
- Open application settings
- Update your software and firmware (see below for more information)
- Open tools or search

As you open tools or perspectives, buttons for them are displayed in the upper right. You can use those buttons to easily navigate back to the Launcher perspective or to other perspectives.

- Change your preferred SDK (see below for more information)
- Change debug mode
- Update adapter firmware (see below for more information)
- Access demos, examples, documentation, and other resources (see below for more information)

Note: Perspectives are made up of a number of tiles or panes, called views, as well as the content in those views. You can change the layouts of various perspectives by expanding or relocating views, or adding or removing views. If you want to return to the default layout, right-click the perspective button in the upper right and select **Default**.

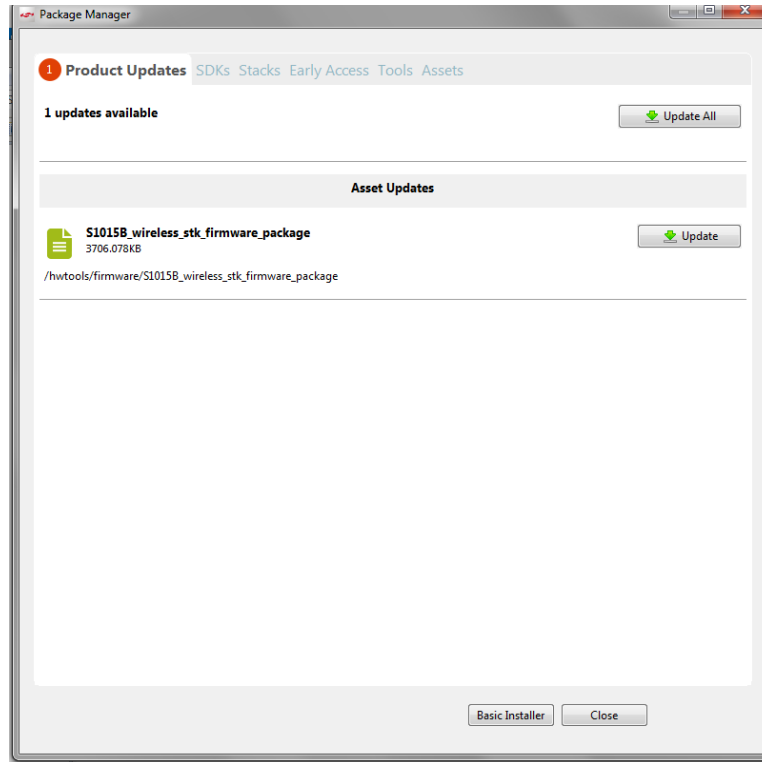
The Devices view on the left of the Launcher perspective also has a solutions tab. If you are developing for complex networks with a number of different parts involved, you can add them all to the solution and then select the one you are working on from the list.



3.1 Downloading Updates

The Download Update icon will be red if updates are available. If Simplicity Studio detects an available update, and you are in another perspective that does not display the update icon, you will be notified that an update is available.

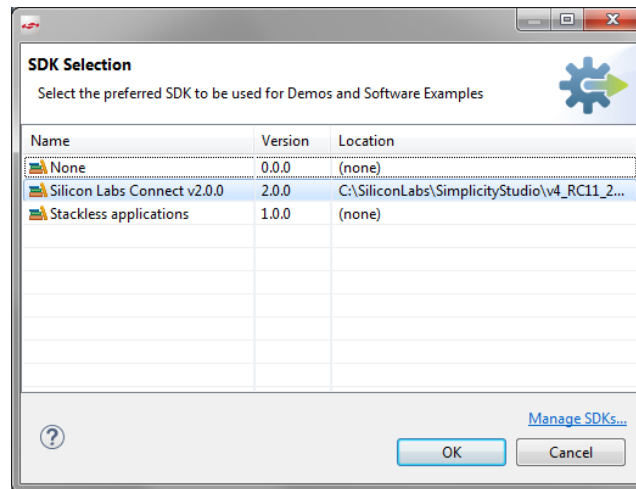
. When you click the Download Update icon or accept updates in the dialog, Simplicity Studio then shows you available updates in the Package Manager dialog. You can update all or select individual updates for installation. Click the tabs in the Package Manager dialog to see other components available for installation.



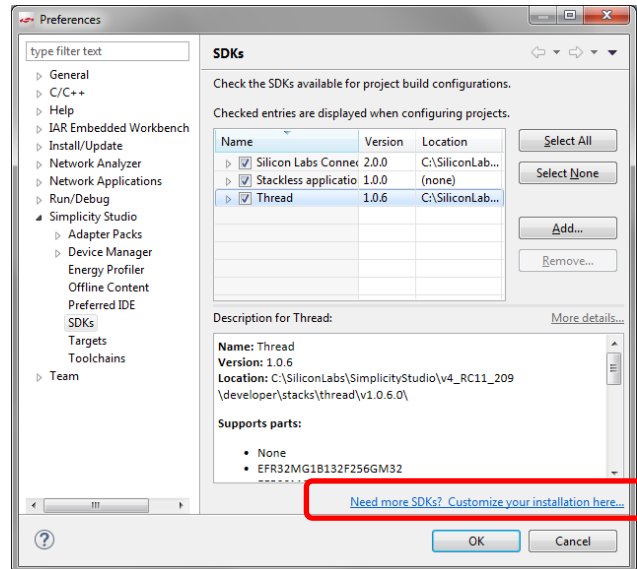
Note: If you are using a new device or product family, you can use the Basic Installer button to access the installation interface provided during initial installation. This makes installation of all components related to a selected device(s) or an entire family easier.

3.2 Changing the Preferred SDK

If you have more than one stack or SDK installed, you can change the preferred (or currently active) SDK by clicking one of the links. The SDK selection interface is displayed. Click an SDK, and then click **OK**.



If the SDK you want to use is not displayed, in the SDK Selection dialog click **Manage SDKs**. In the Preferences dialog, click **Need more SDKs ...**

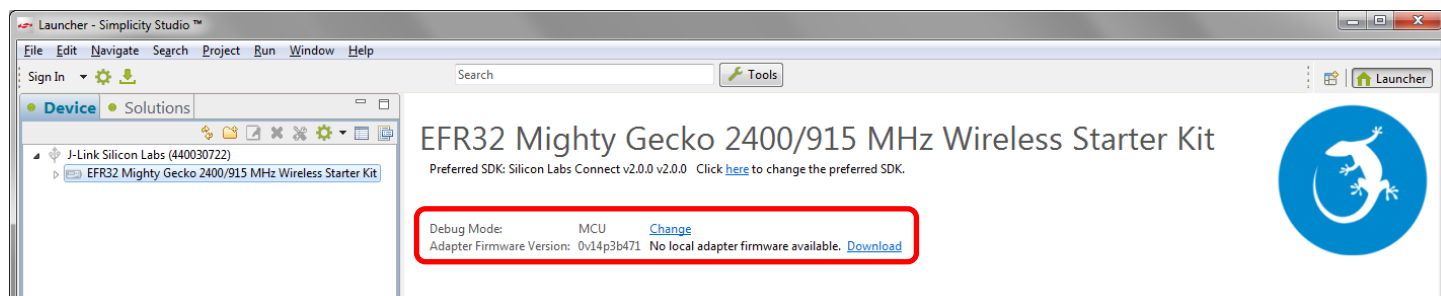


This updates your system and opens the Package Manager interface, the same as if you clicked the Download Updates icon. Click on either the SDK or the Stack tabs to see and install a new item. Then return to the SDK Selection dialog and select the recently installed SDK.

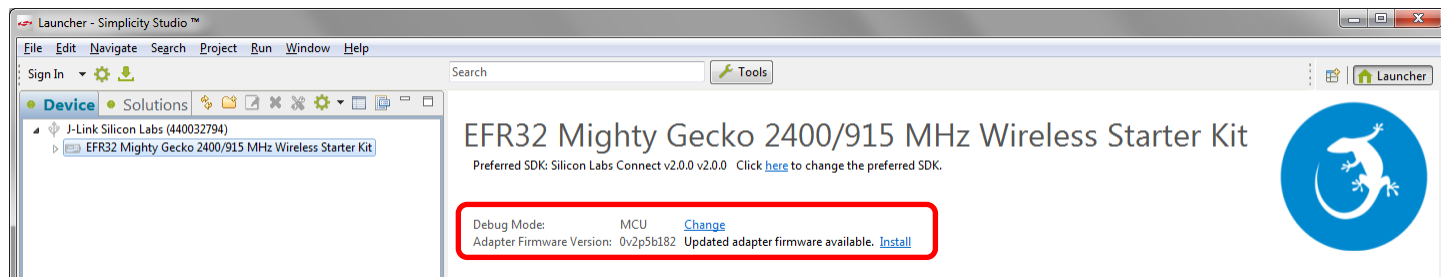
Note: If you are changing to an SDK you have installed outside of Simplicity Studio, first add it, which takes you back to the Launcher perspective. Then click one of the links to change the preferred SDK, select the SDK, and click **OK**.

3.3 Updating Adapter Firmware

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

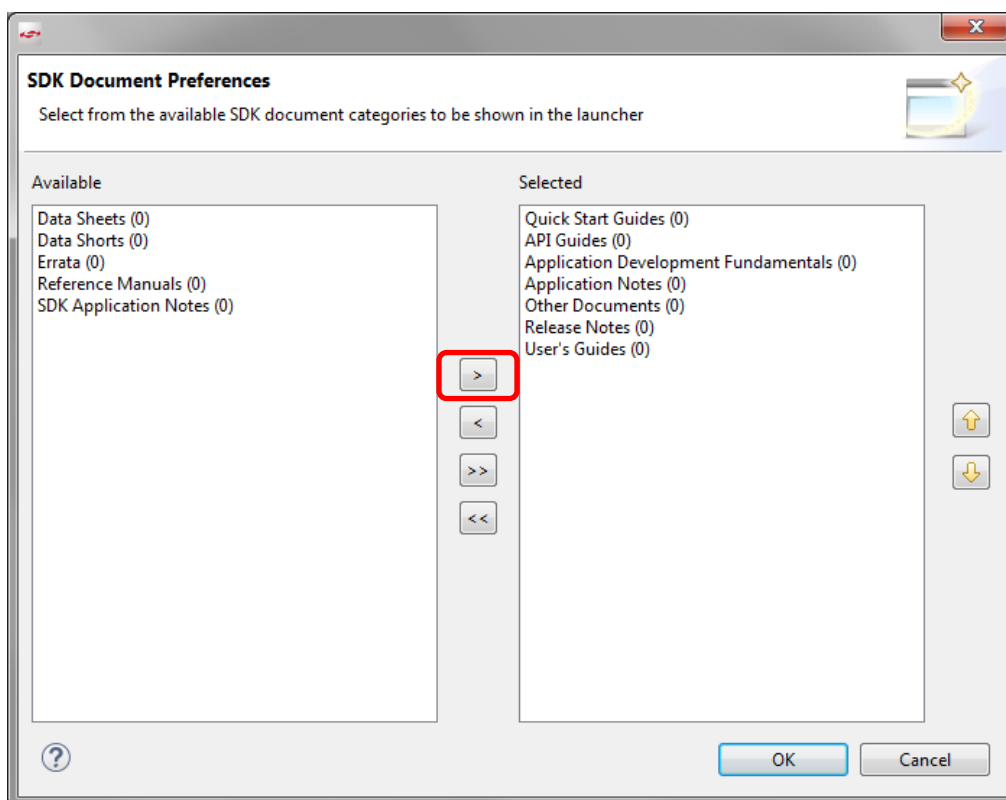


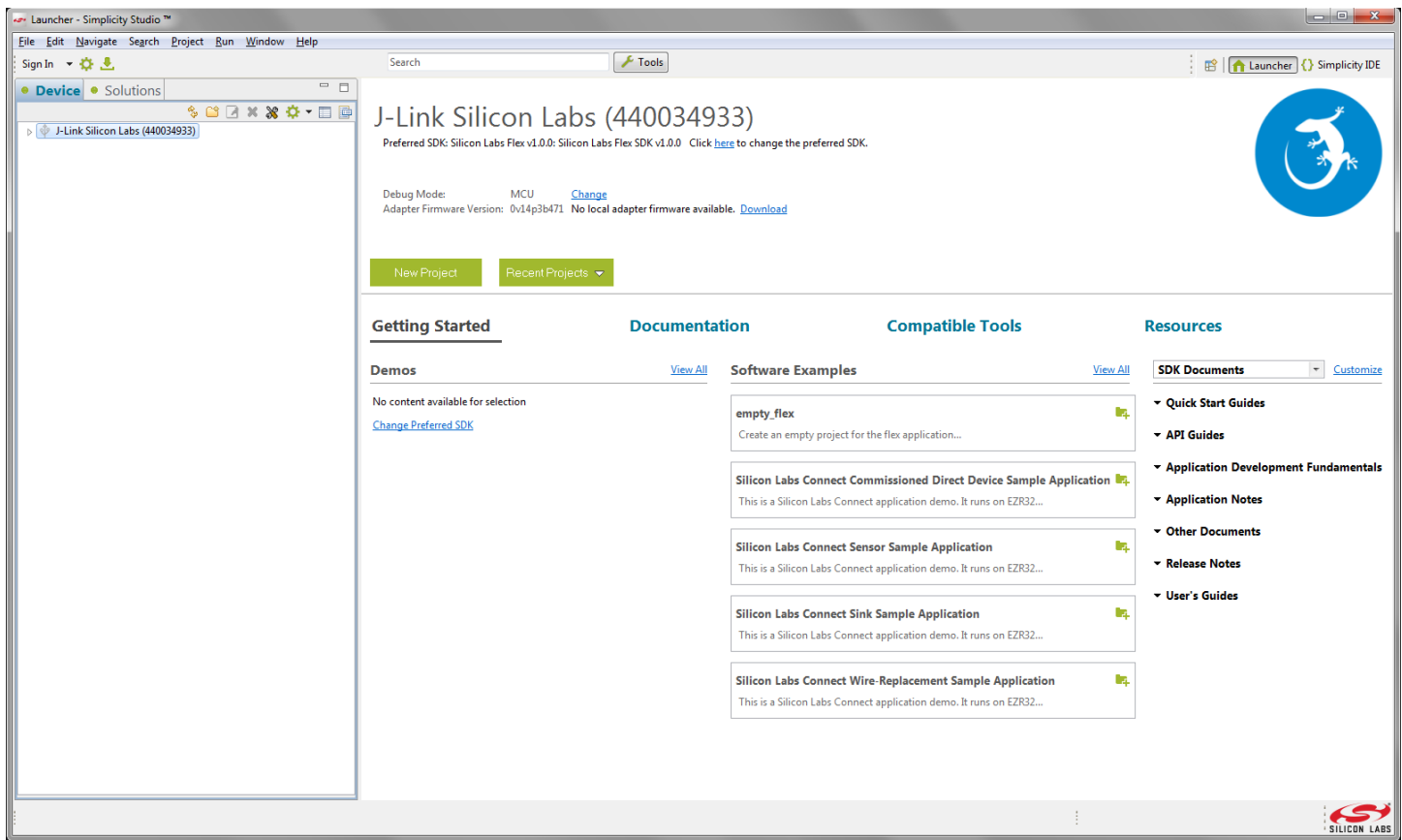
If an update is available, click **Install** to install the firmware.



3.4 Accessing Documentation and Other Resources

The **Getting Started** tab provides access to demos, example applications, and application notes and stack documentation. To see SDK documents, click **Customize**, select categories by moving them from the left to the right, and click **OK**.





The **Documentation** tab lists schematics, specifications, datasheets and other documentation about the kit hardware.

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

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

4 Working with Example Applications

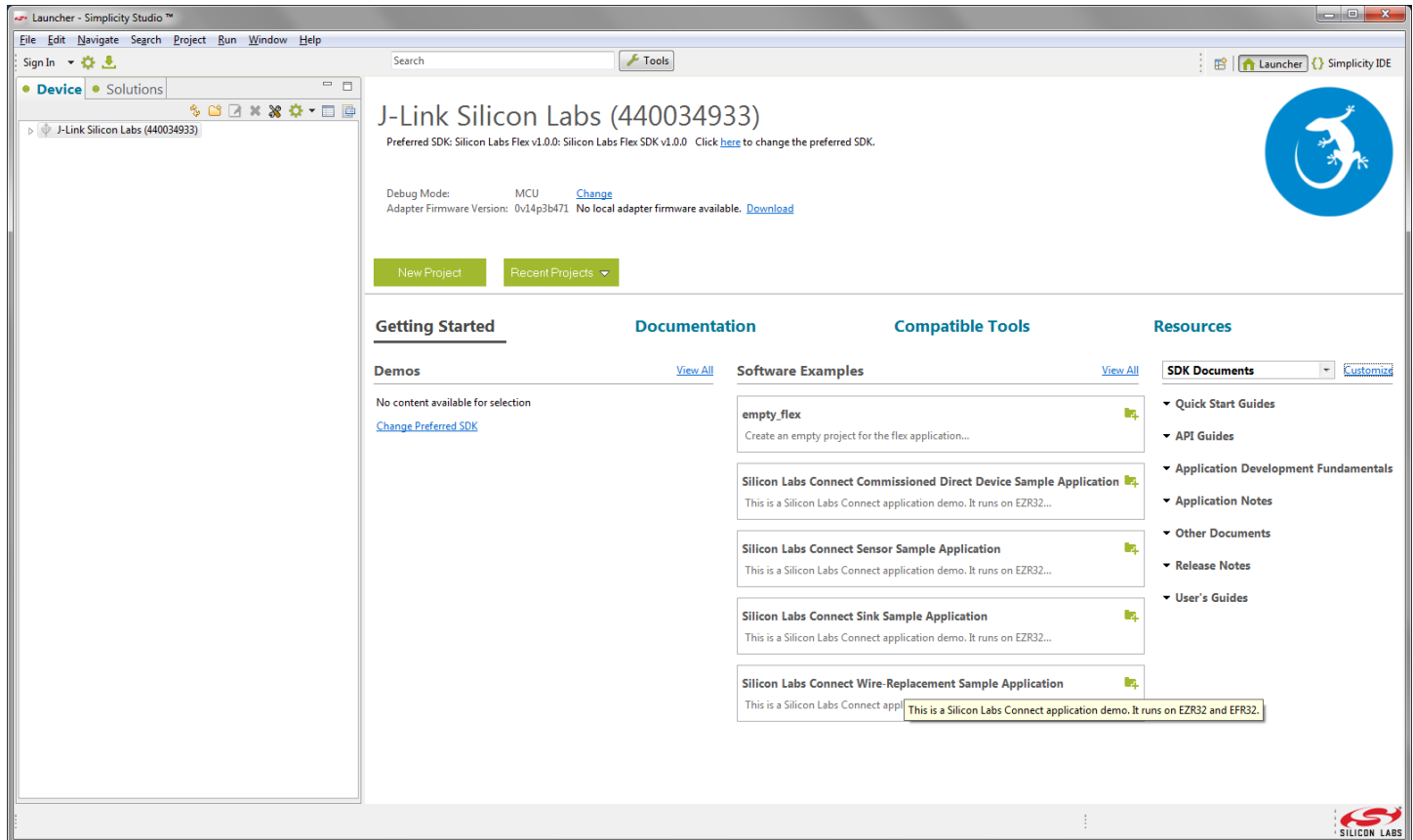
When working with example applications in Simplicity Studio, you will execute the following steps:

1. Select an example application.
2. Generate application files.
3. Compile and flash the application to the radio board.

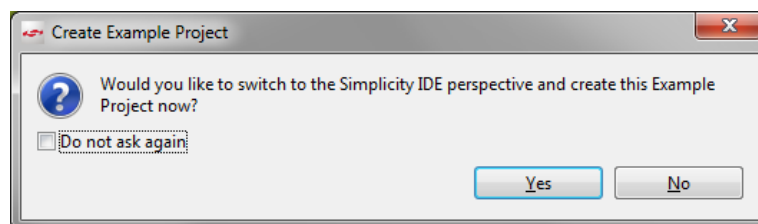
These steps are described in detail in the following sections.

4.1 Selecting a Connect Example Application

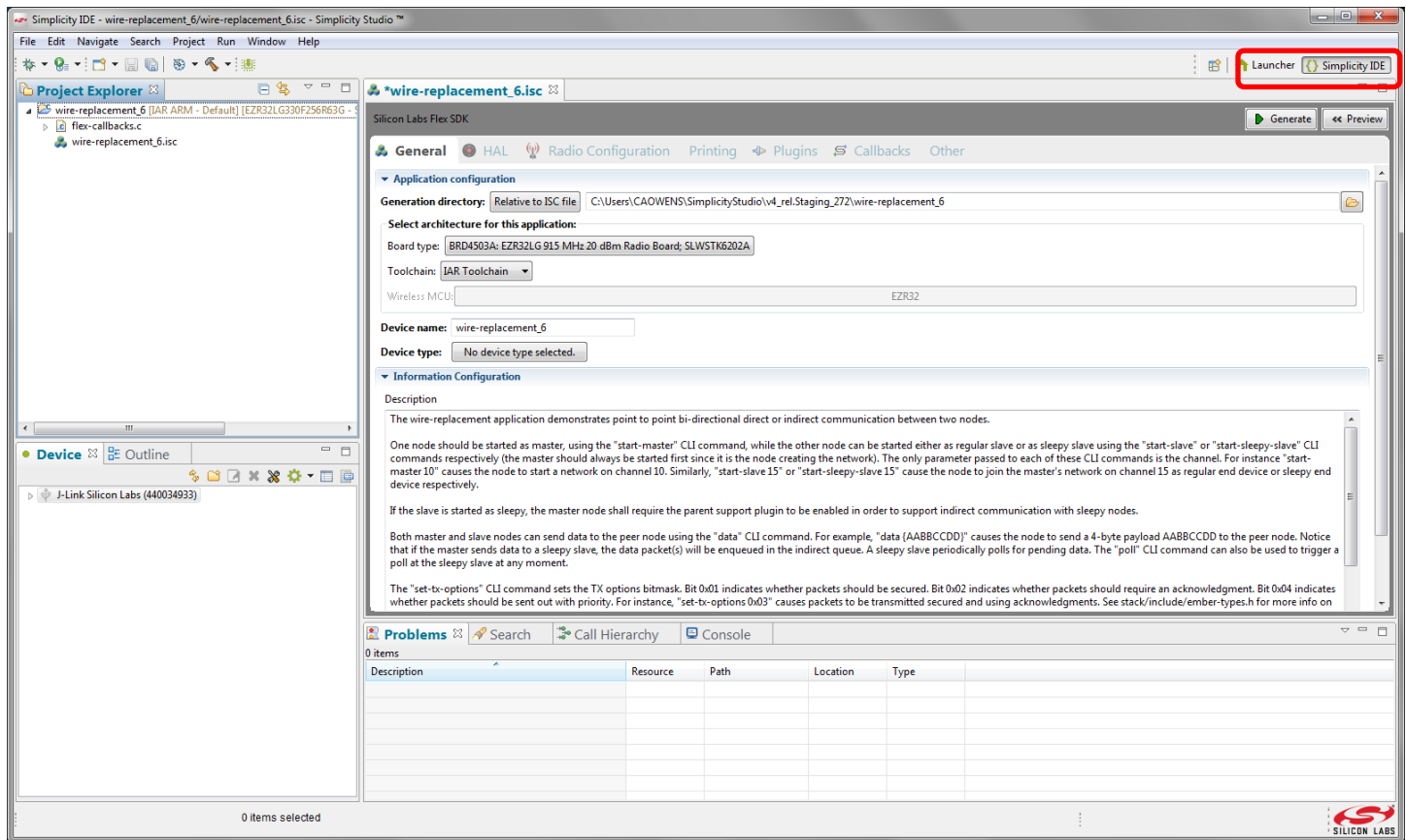
1. In the Launcher perspective, click on an example application, in this case the Wire Replacement Sample Application.



2. You are asked if you want to switch to the Simplicity IDE. Click **Yes**.

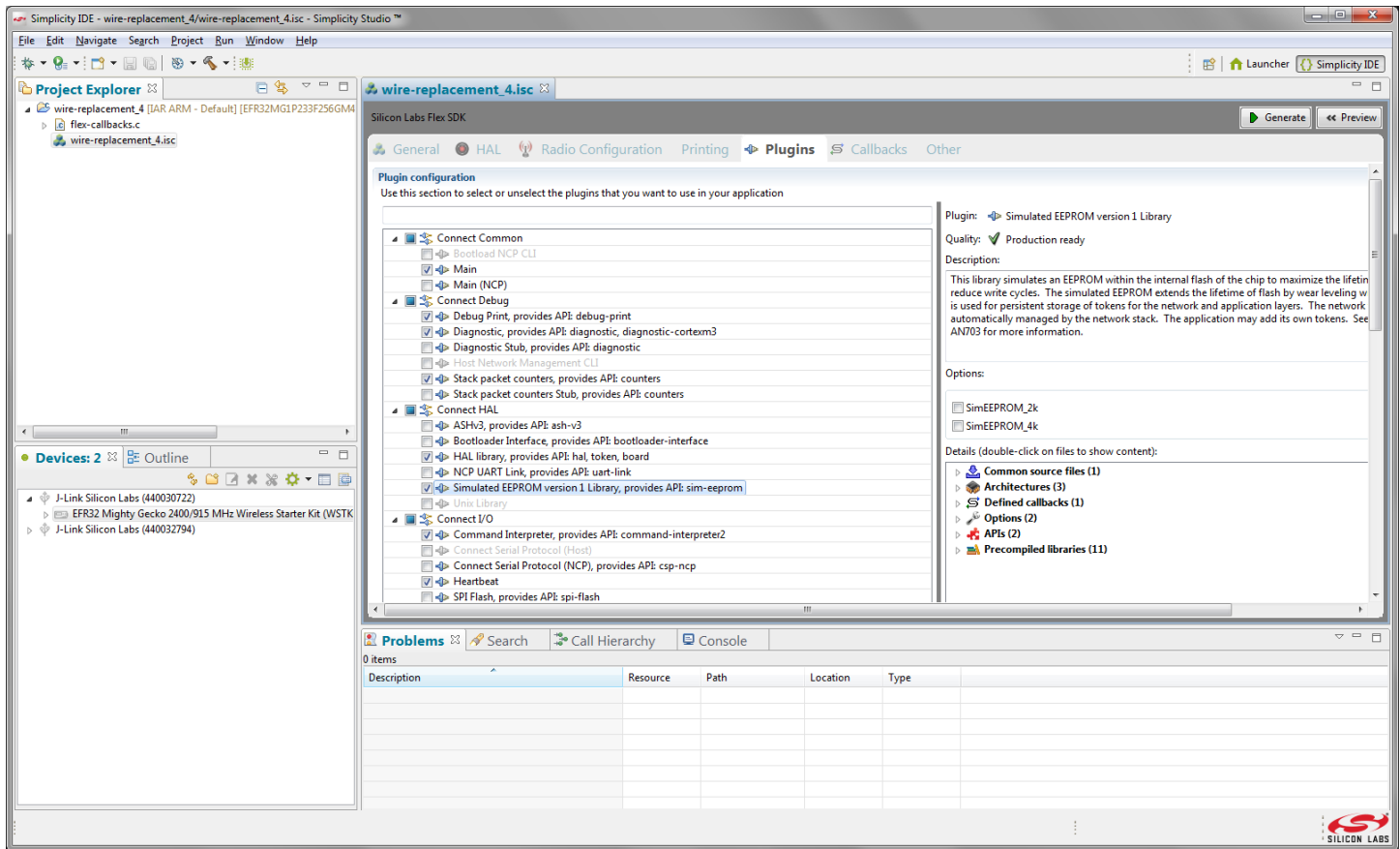


3. Simplicity IDE opens with the new project in AppBuilder view.



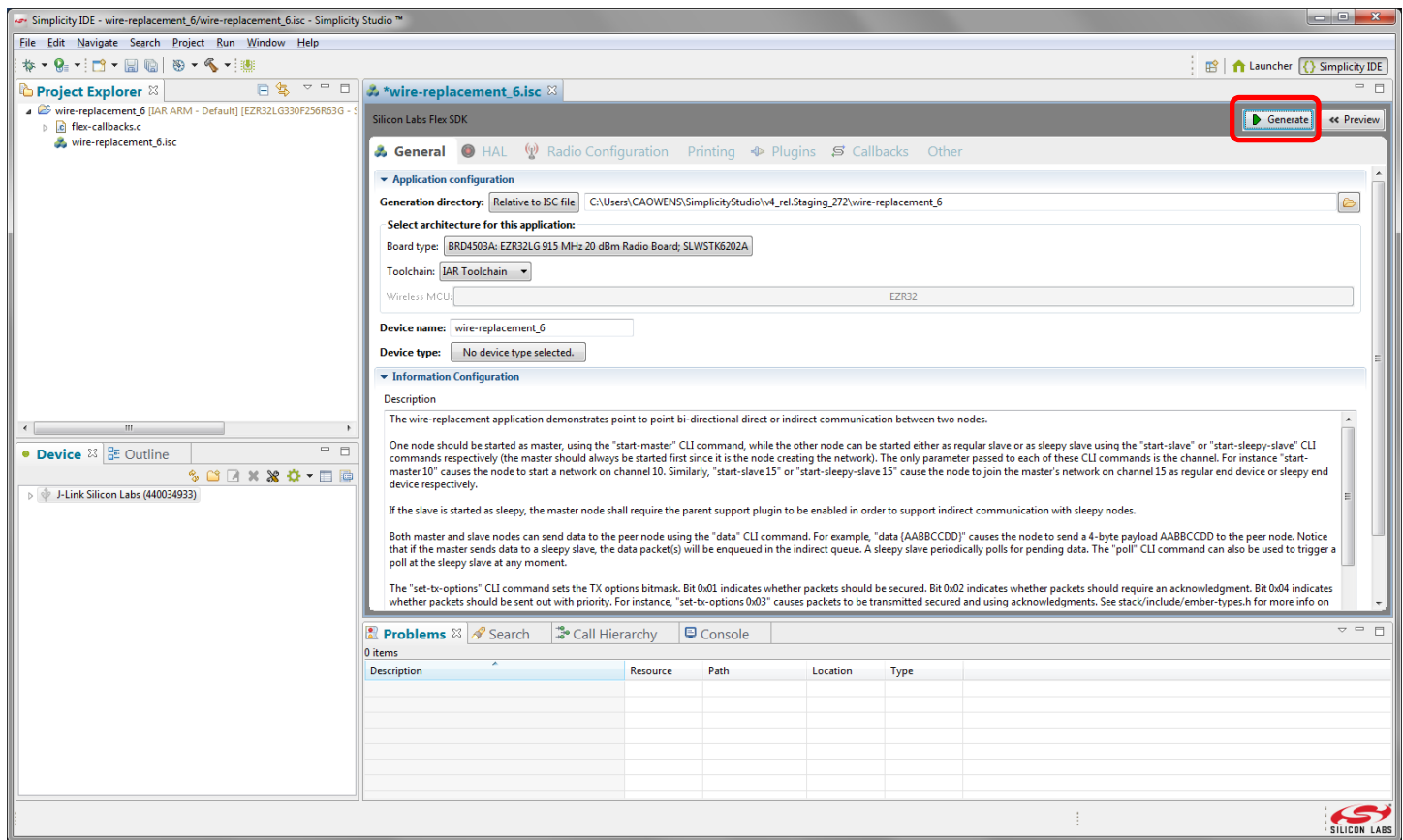
Note: You now have a Simplicity IDE button next to the Launcher button in the upper right.

The Plugins tab displays the various plugins that are preselected to enable the example's functionality. Click on a plugin to see its configurable options (if any), along with more information about the plugin.

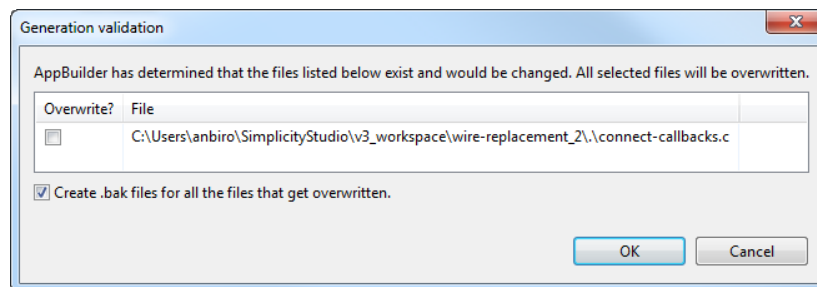


4.1.1 Generating Connect Application Files

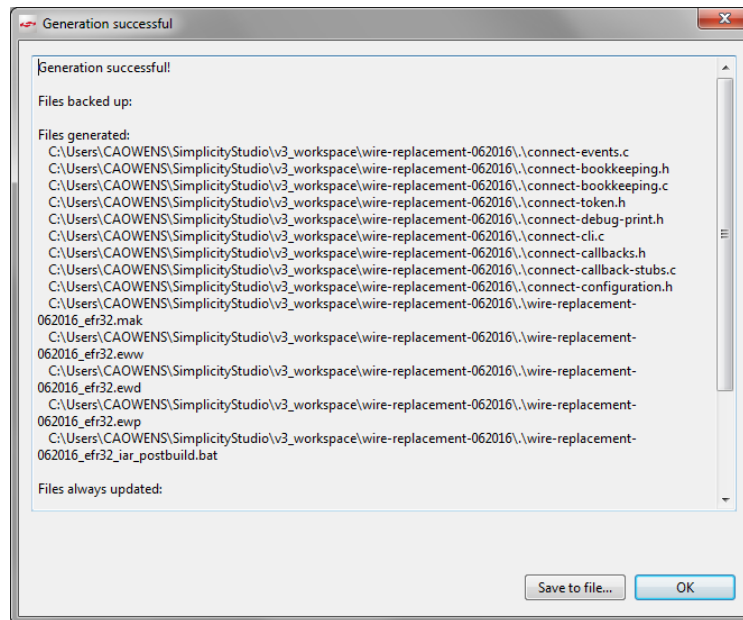
1. In the General tab, the Information Configuration block provides details about application setup and functionality (you may need to scroll down to see it). When you are ready to generate the sample project code, click **Generate**.



2. When asked about overwriting files, as long as you are working with examples do not check connect-callbacks.c. Only overwrite before starting your own coding, to create a file populated with stub callbacks for those you have indicated. Once you have made modifications to the callbacks in the file, be careful not to overwrite again.



3. Once generation is complete, a dialog reporting results is displayed. Click **OK**.

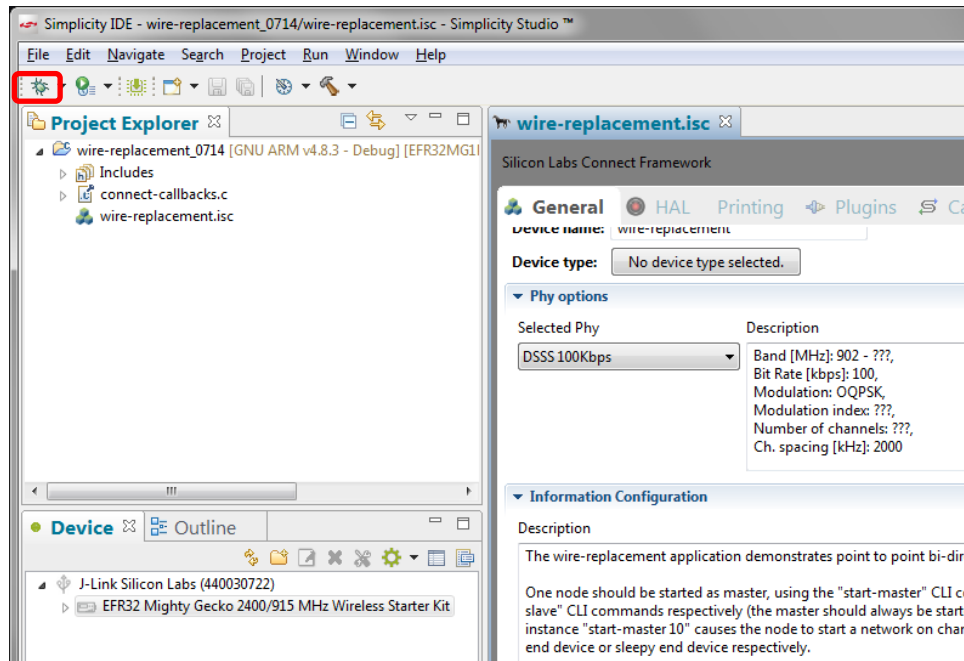


4.2 Compiling and Flashing the Connect Application

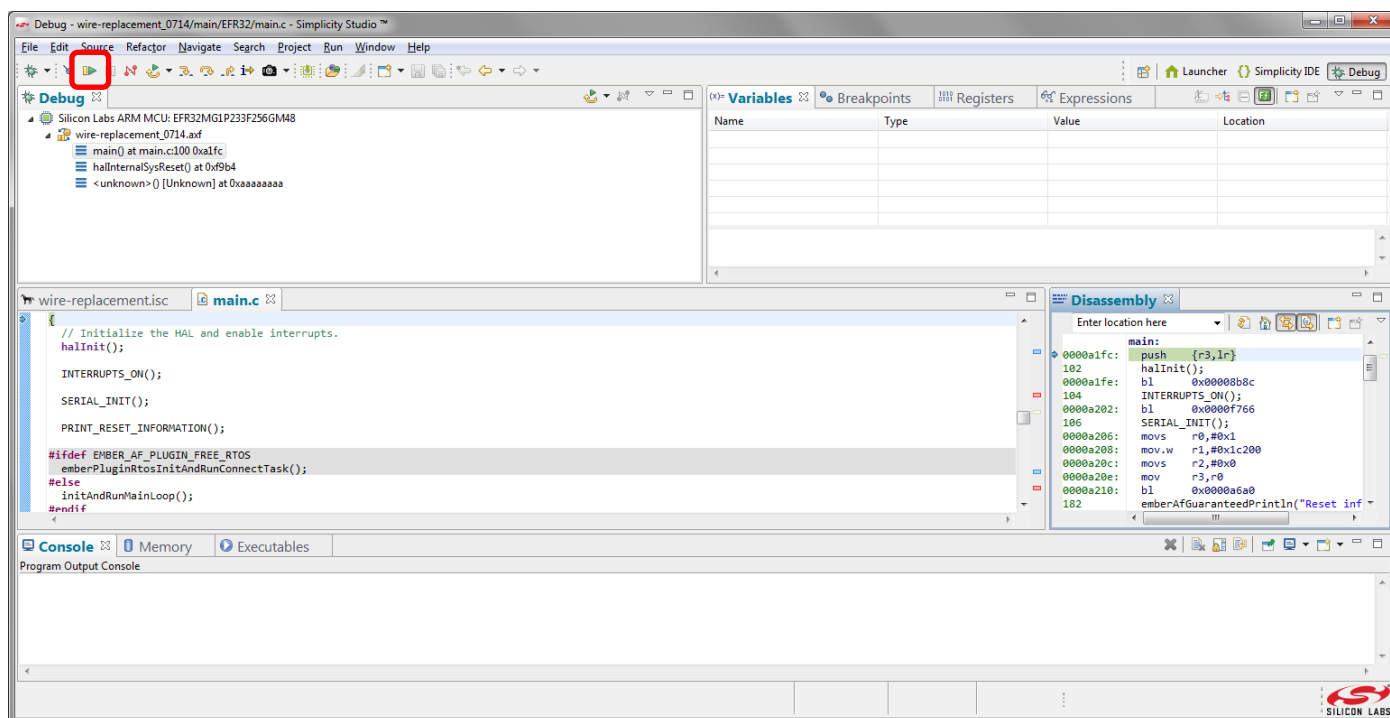
You can either compile and flash the application automatically, or manually compile it and then flash it.

Automatically Compile and Flash

1. You can automatically compile and flash the application to your connected development hardware in the Simplicity IDE. After you click **OK** on the Generation Confirmation dialog, the Simplicity IDE returns. Click the **Debug** control.



- Progress is displayed in the lower left. Wait until flashing has completed and a Debug perspective is displayed. Click the **Resume** control to start the application running on the WSTK.



Next to the Resume control are Suspend, Disconnect, Reconnect, and stepping controls. Click the **Disconnect** control when you are ready to exit Debug mode.



Manually Compile and Flash

After you generate your project files, instead of clicking Debug in the Simplicity IDE, click the **Build** control in the top tool bar. Your sample application will compile based on its build configuration. You can change the build configuration at any time by right clicking on the project and going to **Build Configurations > Set Active**.

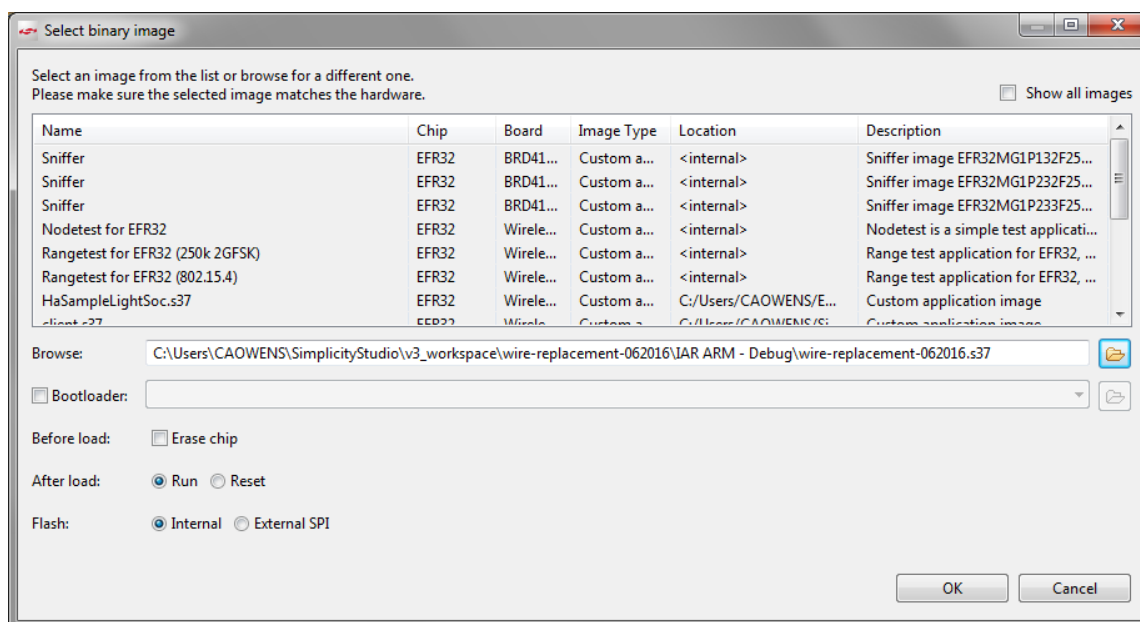


You can also build your application directly in IAR-EWARM by opening IAR-EWARM and opening the generated project file inside IAR.

- Open IAR-EWARM.
- Select File > Open > Workspace and navigate to the location you selected for your sample application.
- Select the application .eww file and click **[Open]**.
- Select Project > Make or press F7. If the application builds without errors, you are ready to install the image on a device.

You can load the binary image through the Devices view in the Simplicity IDE perspective.

1. In the Devices view, right-click on the J-Link device and select **Upload Application**. The Select Binary Image dialog is displayed.

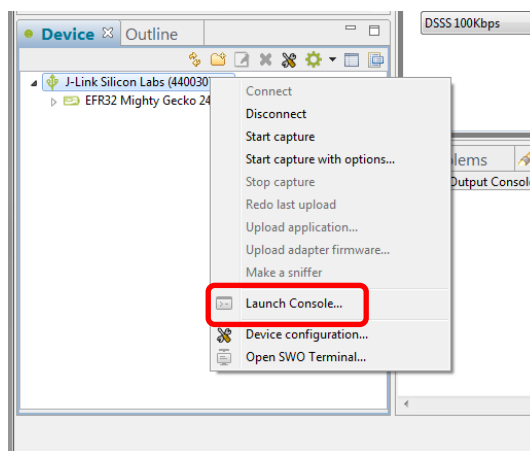


2. Navigate to the .s37, .bin or .hex image you wish to upload.
3. Optionally, check **Erase Chip**, to make sure that any previous bootloader or other non-volatile data is erased before your new image is uploaded. New users will typically check this.
4. The **After Load** options are **Run** (begin executing the code immediately) and **Reset** (wait for an event, such as a debugger to connect or manual initiation of a boot sequence). During initial development you will typically leave this set to **Run**.
5. Click **OK**.

4.2.1 Interacting with Connect Examples

Depending on the example application, you may be able to interact with it through your development environment's Console interface using a CLI (command line interpreter). The console interface allows you to form a network and send data.

To launch the Console interface, in the Simplicity IDE perspective right-click on your J-Link device in the Devices View. Choose **Connect** (if you are not already connected) and then **Launch Console**. Alternatively, from the Tools drop-down in the Launcher perspective select **Device Console**. To get a prompt on the Console Serial 1 tab, press Enter.



To create the network on the wire-replacement example (comments in parentheses), repeat the procedures in section **Selecting an Example Application** through **Compiling and Flashing the Application** to compile and load the Wire Replacement example on a second device.

Then use the following procedure.

1. On Device A, enter `start-master 0` (where 0 is the channel number. You should see `Network Up`).
2. On Device B, enter `start-slave 0` (where 0 is the channel number). You should see:

From Device B: `Network Up`

From Device A: `Slave 0x0001 joined as end device from device A`

3. On device B, enter `data {AA BB CC}`

From Device B: `TX: Data to 0x0000:{AA BB CC } : status=0x00`

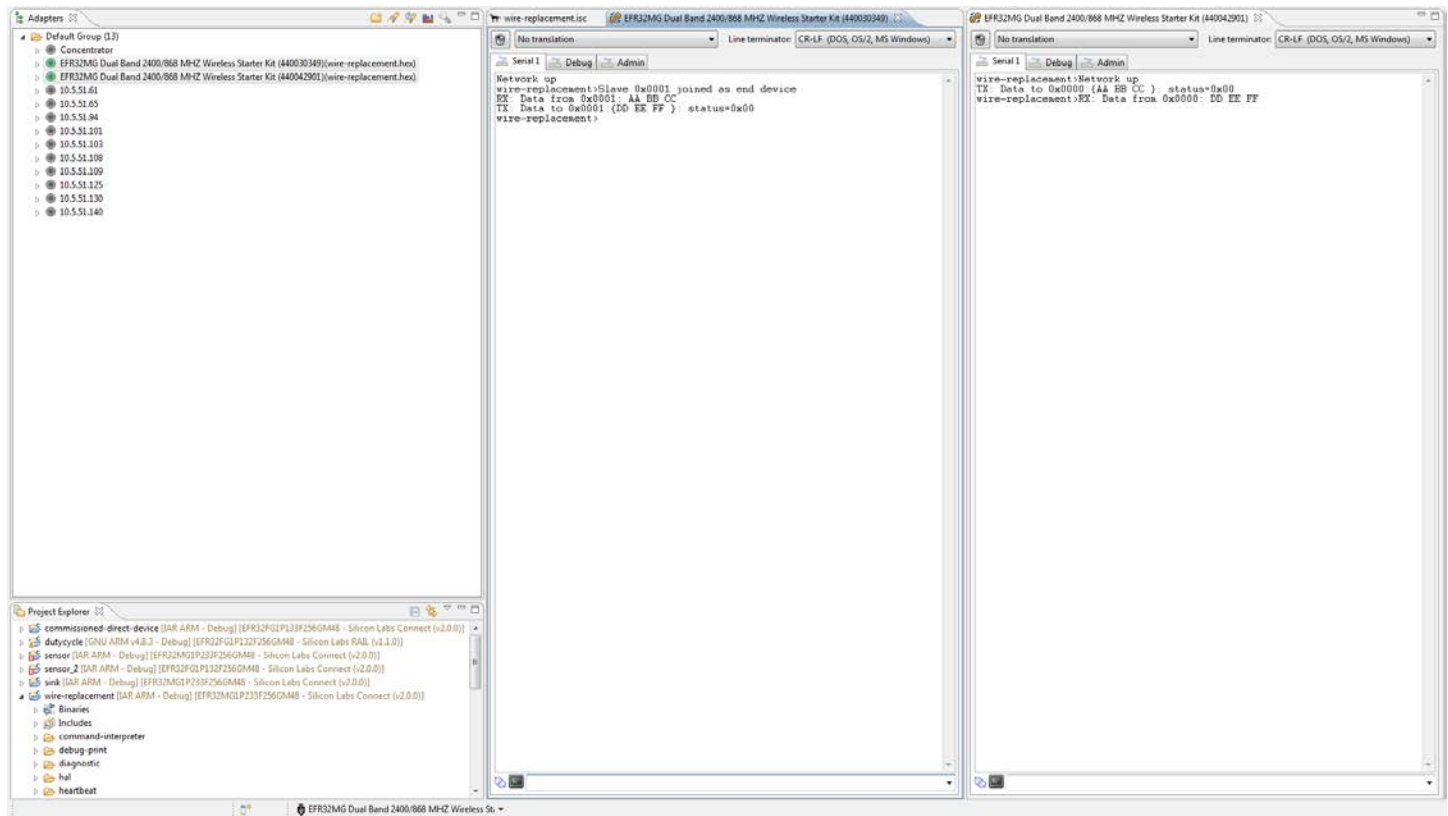
From Device A: `RX: Data from 0x0001: AA BB CC`

4. On Device A enter `data {DD EE FF}`

From Device A: `TX: Data to 0x0001:{DD EE FF } : status=0x00`

From Device B: `RX: Data from 0x0000: DD EE FF`

The following figure shows the console at the end of the procedure.



5 Appendix: EFR32/EZR32 Comparison

The following table compares the Connect features for the EZR32 with those for the EFR32.

Table 1. EZR32/EFR32 Connect Feature Comparison

Connect Feature	EFR32	EZR32
PHY Support (FCC, ETSI, Anatel, China, ARIB, KCC)	Yes	Yes
Networking Stack Functions (Network formation, Scanning, Association, CSMA-CA, Filtering, Retries, Auto-Acks)	Yes	Yes
Node Types (End Nodes, Sleepy End Nodes, Coordinator, Range Extender)	Yes	Yes
Network Topologies (Point-to-Point, Star, Sleepy Star, Extended Star)	Yes	Yes
DSSS	Yes	No
NCP Mode	Yes	No
Security (XTEA, AES)	Yes	Yes
AppBuilder Support in Simplicity Studio	Yes	Yes
Hardware Configuration in Simplicity Studio	No (Future Release)	No (Future Release)
Frequency Hopping	FCC - Yes (ETSI- Future Release)	No
Secure Bootloader	No (Future Release)	No
OTA Updates (Broadcast/Unicast)	Yes	No
FreeRTOS Support	No	No
Radio Configuration	Yes	No (Pre-Set Config.)
Compilers	IAR, GCC	IAR
Multi-Address Filtering	Yes	No
Antenna Diversity	No (future release)	No
Sample Applications	Point-to-Point Sensor/Sink	Point-to-Point Sensor/Sink