QSG106: Getting Started with EmberZNet PRO

This quick start guide provides basic information on configuring, building, and installing applications for the EM35x and Mighty Gecko (EFR32MG) family of SoCs using the EmberZNet PRO stack v. 5.7 and higher and Simplicity Studio.

This guide is designed for developers who are new to EmberZNet PRO and the Silicon Labs development hardware. It provides instructions to get started using the example applications provided with the EmberZNet PRO stack.

KEY FEATURES

- Product overview
- Setting up your development environment
- Discovering your SDK
- Working with example applications
- Configuring the peripheral interface

1 Product Overview

Before following the procedures in this guide you must have

- Purchased your development hardware:
  - Mighty Gecko (EFR32MG) Mesh Networking Kit

    or

  - EM35x Development Kit
- Registered your kit on the Silicon Labs website. This gives you access to a portal from which you can download the EmberZNet PRO stack and other Silicon Labs software, and obtain support.
- 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.

1.1 Software Components

See the stack release notes for version restrictions and compatibility constraints for the stack and these component. To develop EmberZNet PRO applications, you will need the following:

- The Simplicity Studio development environment, which incorporates AppBuilder. 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.
  - Simplicity Studio operates on top of an application framework, a working software design that you can modify to meet your needs. It provides common basic services and, when combined with device-specific code, provide you with most of the over-the-air behavior for your device. Some common usage case examples based on the relevant application framework are provided along with a pre-built application, Node Test, for functional testing of RF modules.
  - The hardware abstraction layer (HAL) acts as a conduit between the network stack and the node processor and radio. Separating network stack functionality from the specific hardware implementation enables easy portability. HAL code is provided as a combination of pre-built libraries for complex, stack-critical functionality and C source code that you can alter in order to customize, extend, or reduce device functionality across various hardware platforms. The HAL API is documented in the online API reference.
- The EmberZNet PRO stack, an advanced implementation of a ZigBee stack. The stack API is documented in online API reference as well as in other documents installed with the stack installer, or available through the development environment. The stack is delivered as a collection of libraries that you can link to your applications. A description of each library is provided in the development environment. The release notes contain details on the folders installed along with their contents.
- IAR Embedded Workbench for ARM (IAR-EWARM) 7.30, used as a compiler in the Simplicity Studio development environment. Download the supported version 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 them for the tasks in this Getting Started guide, you may wish to become familiar with the manufacturing utilities available for your environment. The utilities are hardware-dependent.

- For both the EM35x Development Kit and the Mighty Gecko Mesh Networking Kit, the ISA3 utilities, downloaded from the support portal. The installer modifies your PATH environment variable so that the command line utilities can be easily executed from a Windows Command Prompt. See UG107, *ISA3 Utilities Guide*, for detailed information on the ISA3 utilities.
- For the Mighty Gecko Mesh Networking Kit, Simplicity Commander, installed along with Simplicity Studio in a subfolder of the install folder. See UG162, *Simplicity Commander Reference Guide*, for more information.

Finally, if you are working with an EM35x development kit and you want to use the USB interface of the breakout board for UART connectivity, download a driver for the FTDI USB <-> Serial converter from http://www.ftdichip.com/Drivers/VCP.htm

1.2 Support

Users can access the Silicon Labs support portal at https://www.silabs.com/support. Use the support portal to contact Customer Support for any questions you might have during the development process.

1.3 Documentation

The stack installer provides a documentation index (in documentation/index.htm and also linked from a Start Menu entry) that contains links to documentation locations and brief descriptions of each document’s purpose. Simplicity Studio provides links to hardware documentation and other application notes. 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 Install your Silicon Labs Stack or Software Development Kit

Install your Silicon Labs software (see section **Software Components** for more information).

2.3 Connect your Hardware

Connect your development hardware 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.

2.3.1 EFR32 Wireless Starter Kit (WSTK)

Connect your WSTK, with radio board mounted, to your PC using a USB cable.

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 (see figure below).

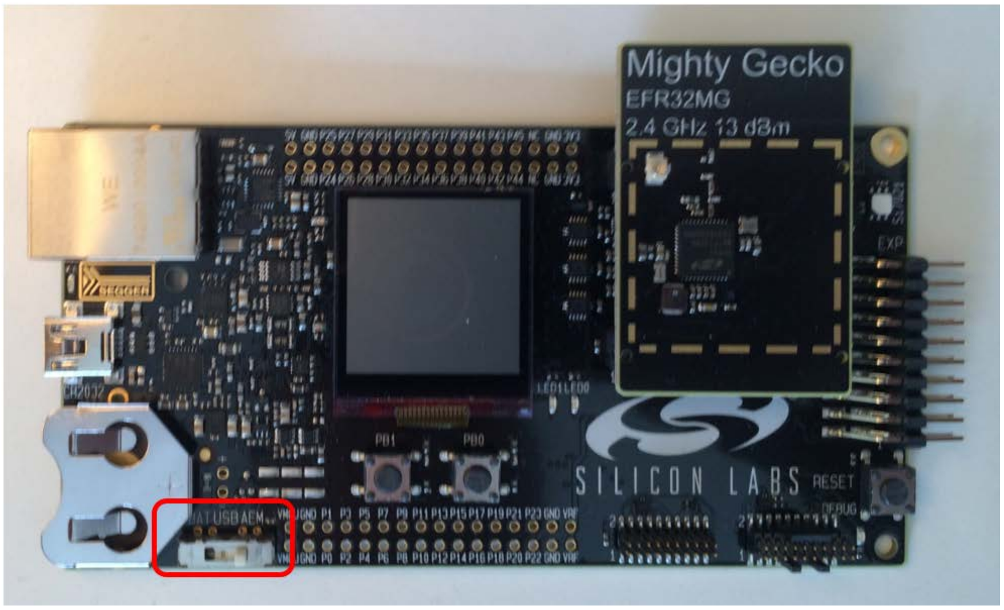


Figure 1. EFR32MG on a WSTK

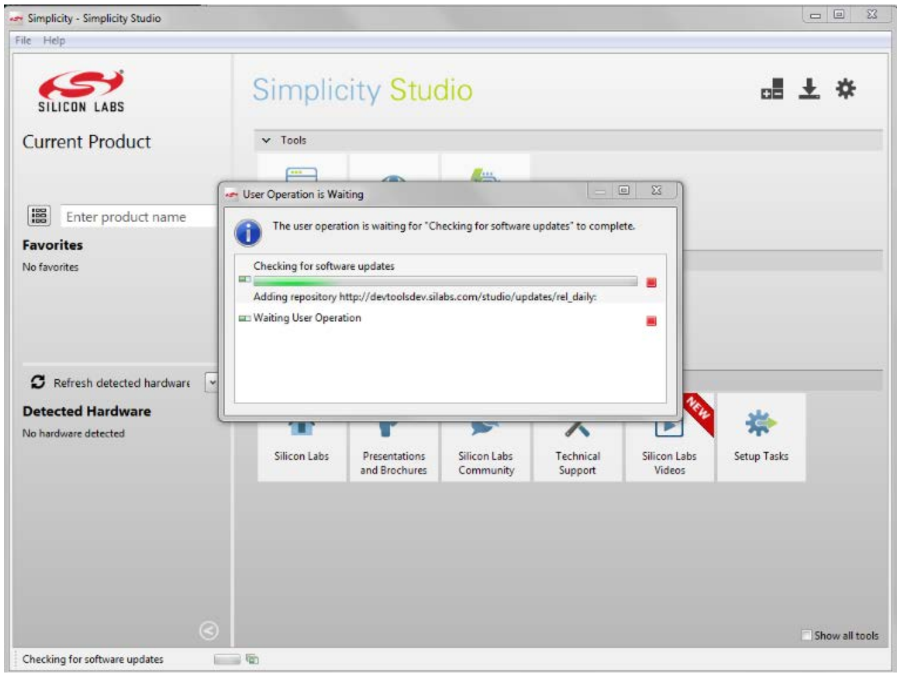
2.3.2 EM35x Development Kit

Follow the instructions on the Quick Start Guide included in the development kit to set up a development environment connected to your computer.

2.4 Install Simplicity Studio

During installation, Simplicity Studio obtains updates and additional packages specific to your connected hardware.

1. As soon as Simplicity Studio launches, it searches for updates. This operation can take several minutes.



User Operation is Waiting

The user operation is waiting for "Checking for software updates" to complete.

Checking for software updates

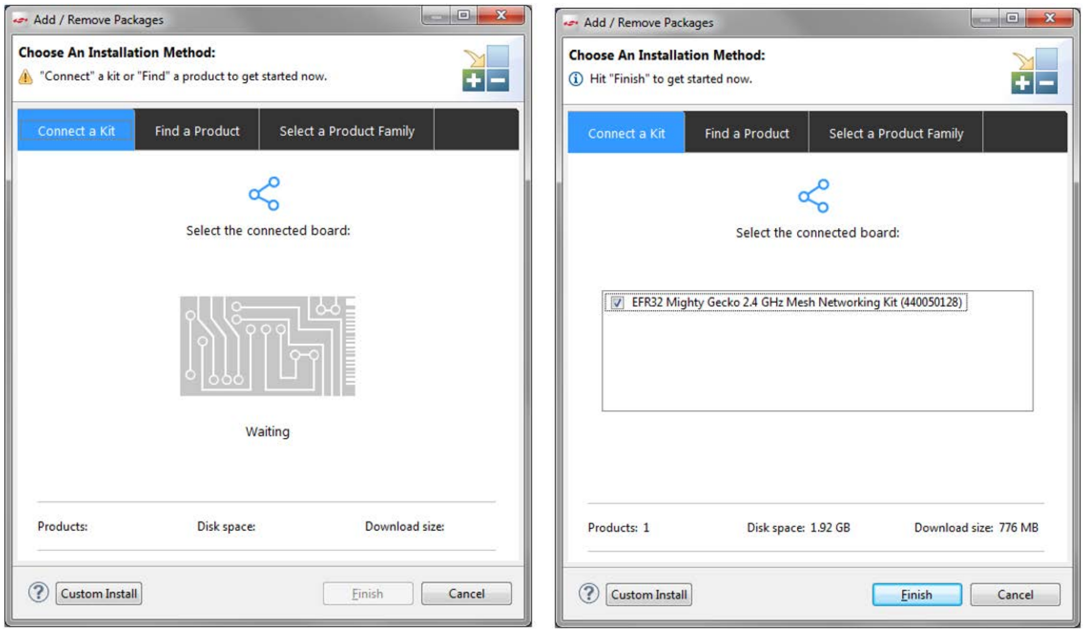
Adding repository http://devtoolsdev.silabs.com/studio/updates/rel_daily:

Waiting User Operation

Checking for software updates

Figure 2. Checking for Software Updates

2. Once software update is complete, Simplicity Studio checks for connected hardware. If you have not connected your WSTK, you are prompted to do so. Connect your hardware and, when the screen changes to show that the hardware has been found, click **[Finish]**.



"Connect" a kit or "Find" a product to get started now.

Select the connected board:

Waiting

Products: Disk space: Download size:

Hit "Finish" to get started now.

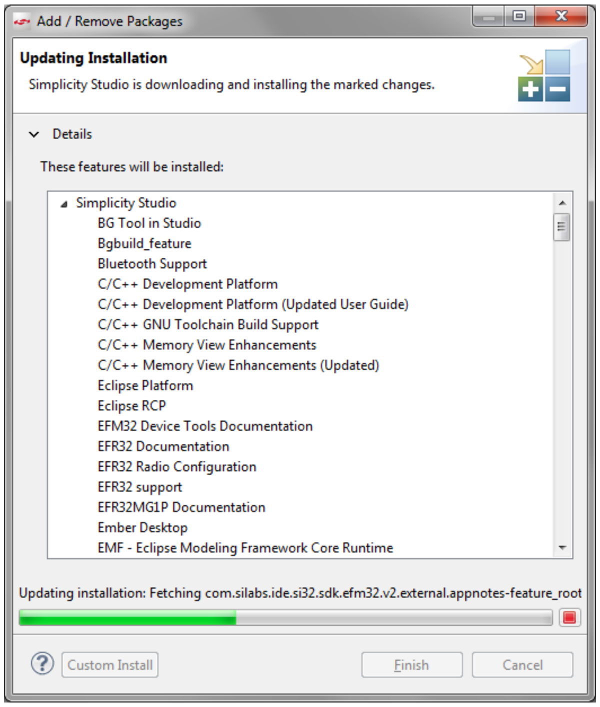
Select the connected board:

☑ EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit (440050128)

Products: 1 Disk space: 1.92 GB Download size: 776 MB

Figure 3. Connect a Kit Before and After Connection

3. Simplicity Studio then installs software packages related to your connected hardware, as shown in the figure below. This procedure can take some time, during which the green progress indicator may appear stationary. However, the update steps above the progress indicator are continuously refreshed.



Updating Installation

Simplicity Studio is downloading and installing the marked changes.

Details

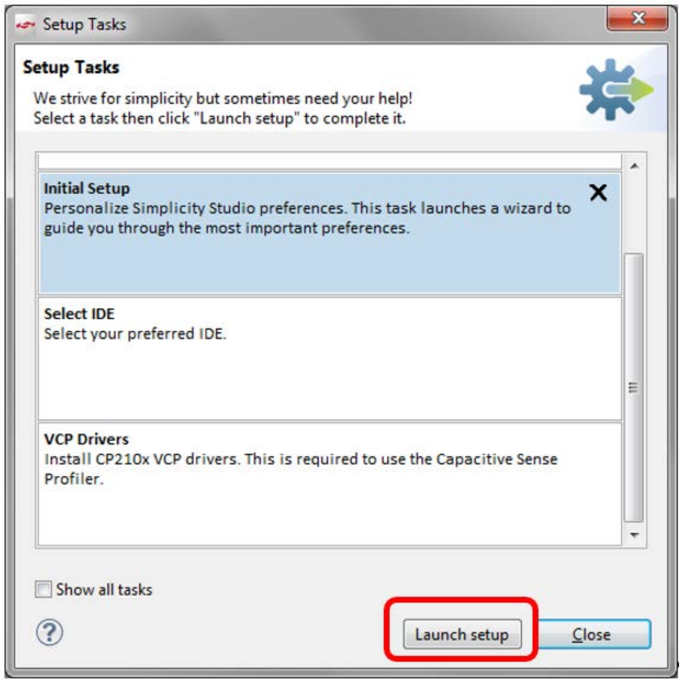
These features will be installed:

- Simplicity Studio
  - BG Tool in Studio
  - Bgbuild_feature
  - Bluetooth Support
  - C/C++ Development Platform
  - C/C++ Development Platform (Updated User Guide)
  - C/C++ GNU Toolchain Build Support
  - C/C++ Memory View Enhancements
  - C/C++ Memory View Enhancements (Updated)
  - Eclipse Platform
  - Eclipse RCP
  - EFM32 Device Tools Documentation
  - EFR32 Documentation
  - EFR32 Radio Configuration
  - EFR32 support
  - EFR32MG1P Documentation
  - Ember Desktop
  - EMF - Eclipse Modeling Framework Core Runtime

Updating installation: Fetching com.silabs.ide.si32.sdk.efm32.v2.external.appnotes-feature_root

Figure 4. Installation Update

4. After update is complete, restart Simplicity Studio.
5. Once restart is complete, a menu of setup tasks is displayed. Select **Initial Setup** and then click **[Launch setup]**.



Setup Tasks

We strive for simplicity but sometimes need your help!
Select a task then click "Launch setup" to complete it.

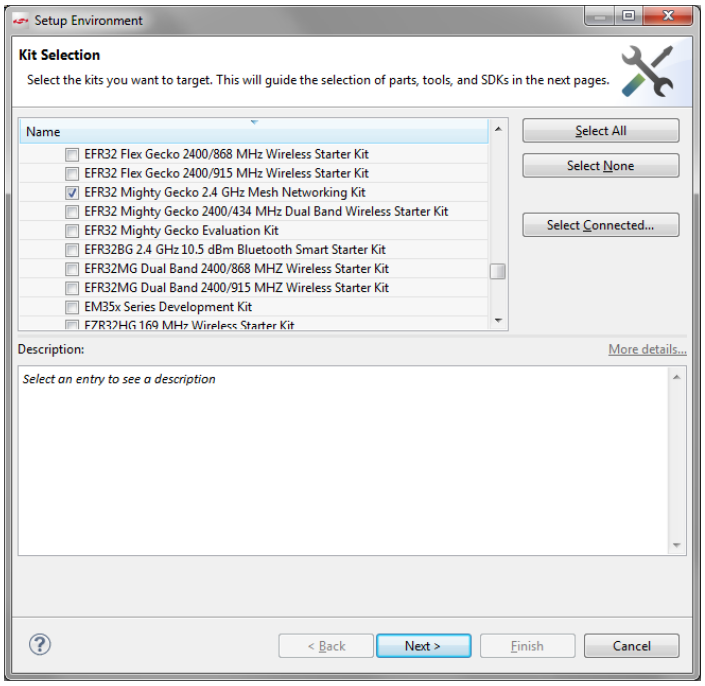
Initial Setup
Personalize Simplicity Studio preferences. This task launches a wizard to guide you through the most important preferences.

Select IDE
Select your preferred IDE.

VCP Drivers
Install CP210x VCP drivers. This is required to use the Capacitive Sense Profiler.

Figure 5. Setup Tasks

6. A kit selection dialog is displayed. Your connected kit should be selected. Click **[Next >]**.



Setup Environment

Kit Selection

Select the kits you want to target. This will guide the selection of parts, tools, and SDKs in the next pages.

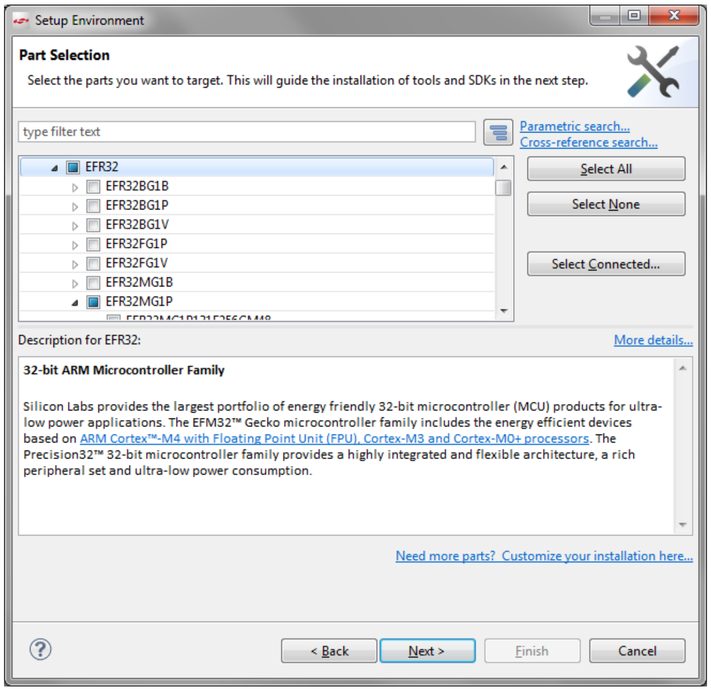
Name
☐ EFR32 Flex Gecko 2400/868 MHz Wireless Starter Kit
☐ EFR32 Flex Gecko 2400/915 MHz Wireless Starter Kit
☑ EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit
☐ EFR32 Mighty Gecko 2400/434 MHz Dual Band Wireless Starter Kit
☐ EFR32 Mighty Gecko Evaluation Kit
☐ EFR32BG 2.4 GHz 10.5 dBm Bluetooth Smart Starter Kit
☐ EFR32MG Dual Band 2400/868 MHZ Wireless Starter Kit
☐ EFR32MG Dual Band 2400/915 MHZ Wireless Starter Kit
☐ EM35x Series Development Kit
☐ EZR32HG 169 MHz Wireless Starter Kit

Description:

Select an entry to see a description

Figure 6. Kit Selection Dialog

7. A part selection dialog is displayed. Your connected part should be selected. Click **[Next >]**.



Setup Environment

Part Selection

Select the parts you want to target. This will guide the installation of tools and SDKs in the next step.

type filter text

- EFR32
  - EFR32BG1B
  - EFR32BG1P
  - EFR32BG1V
  - EFR32FG1P
  - EFR32FG1V
  - EFR32MG1B
  - EFR32MG1P

Description for EFR32:

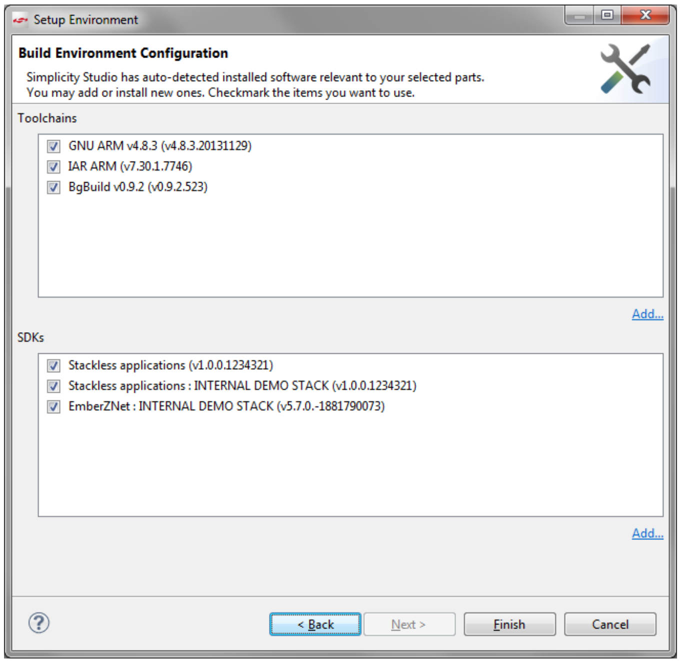
32-bit ARM Microcontroller Family

Silicon Labs provides the largest portfolio of energy friendly 32-bit microcontroller (MCU) products for ultra-low power applications. The EFM32™ Gecko microcontroller family includes the energy efficient devices based on ARM Cortex™-M4 with Floating Point Unit (FPU), Cortex-M3 and Cortex-M0+ processors. The Precision32™ 32-bit microcontroller family provides a highly integrated and flexible architecture, a rich peripheral set and ultra-low power consumption.

Figure 7. Part Selection Dialog

8. A Build Environment dialog (shown below) is displayed that shows detected items. If a Toolchain or SDK is not shown, you can click **Add…** to configure it now, or configure it later from the Settings control. Adding the EmberZNet PRO stack SDK is described in

section **Discovering the EmberZNet PRO Stack**. Click **[Finish]**. The Simplicity perspective, described in the next section, is displayed.


Setup Environment

Build Environment Configuration

Simplicity Studio has auto-detected installed software relevant to your selected parts.
You may add or install new ones. Checkmark the items you want to use.

Toolchains

- [✓] GNU ARM v4.8.3 (v4.8.3.20131129)
- [✓] IAR ARM (v7.30.1.7746)
- [✓] BgBuild v0.9.2 (v0.9.2.523)

SDKs

- [✓] Stackless applications (v1.0.0.1234321)
- [✓] Stackless applications : INTERNAL DEMO STACK (v1.0.0.1234321)
- [✓] EmberZNet : INTERNAL DEMO STACK (v5.7.0.-1881790073)

Figure 8. Build Environment Configuration

9. If you plan to work with NCP software communicating with a host MCU/PC, you should upgrade your WSTK board controller firmware to build 435 or later. Click the Kit Manager tile in the Simplicity perspective and click **[Yes]** in the confirmation dialog.

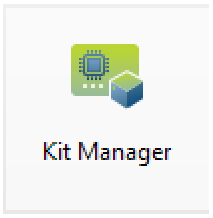
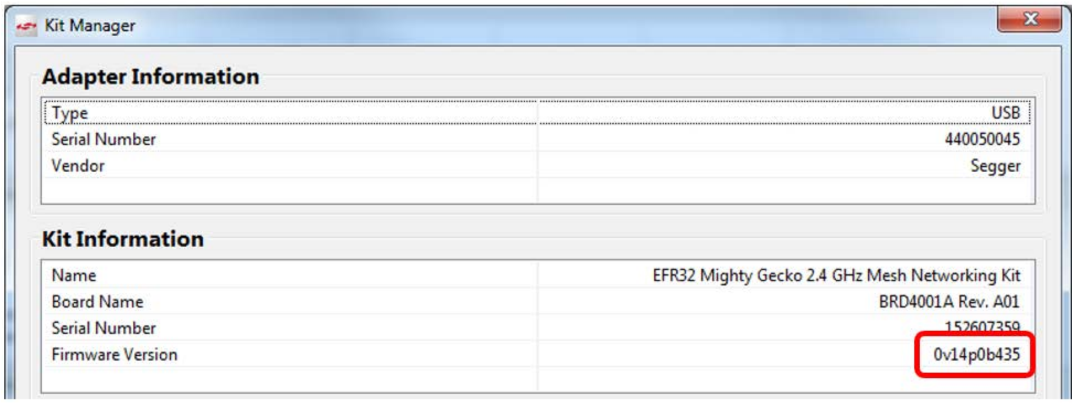


Figure 9. Kit Manager Tile

10. Your firmware is automatically updated to the latest version and displayed in the Kit Manager window once download is complete. Click **[Close]** to return to the Simplicity perspective.


Kit Manager

Adapter Information

Type	USB
Serial Number	440050045
Vendor	Segger

Kit Information

Name	EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit
Board Name	BRD4001A Rev. A01
Serial Number	152607359
Firmware Version	0v14p0b435

Figure 10. Kit Manager with Latest Board Controller Firmware

2.5 Navigation in Simplicity Studio

Simplicity Studio is built on the Eclipse platform. As such, it is broken up into different “perspectives,” each of which allows access to a specific set of functionality. Simplicity Studio starts up in the “Simplicity perspective,” sometimes referred to as the “Home Screen”.

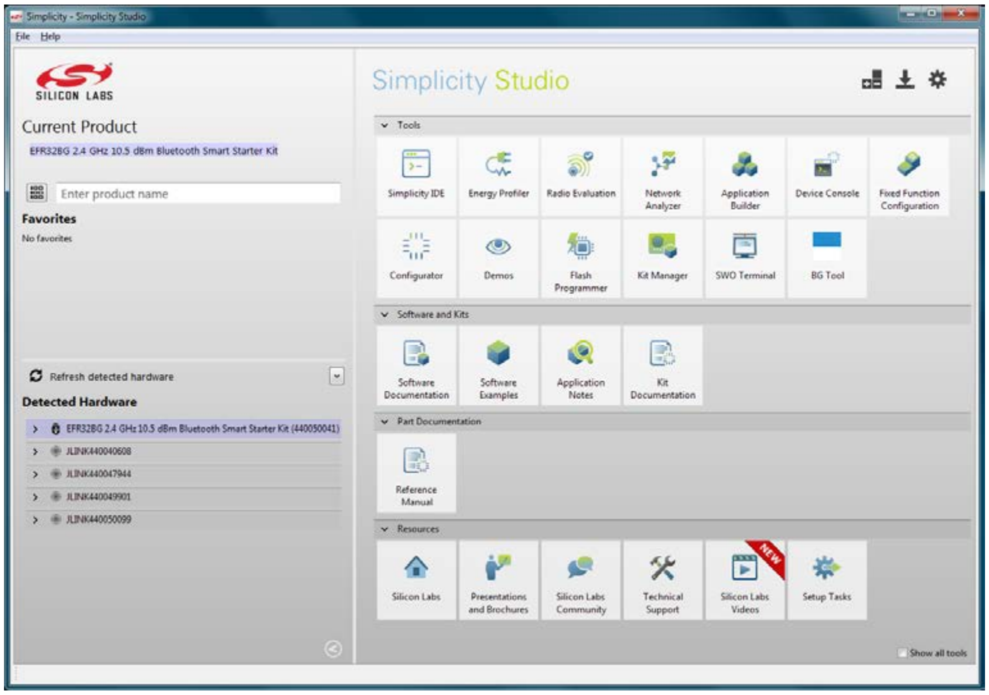


Figure 11. Simplicity Studio's Simplicity Perspective

From the Simplicity perspective, you can discover devices, configure Simplicity Studio, or navigate to another perspective for application development.

The Simplicity perspective shows large tile icons that represent the various sets of functionality within Simplicity Studio. Clicking a tile opens a different perspective. When you are in a different perspective, you can always return to the Simplicity perspective or any other perspective at any time by clicking on one of the tile icons in the top right-hand corner of your screen.

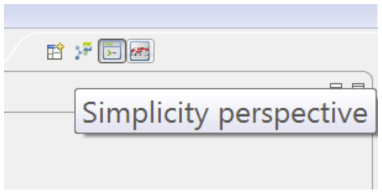


Figure 12. Navigation Tile Icons

Your detected hardware is shown in the **Detected Hardware** panel in the lower left of the Simplicity perspective. If you change connected hardware and it does not auto-detect, click **[Refresh Detected Hardware]**.

Three controls in the upper right allow you to maintain part-specific packages, install updates to the core Simplicity Studio software, and change configuration settings.

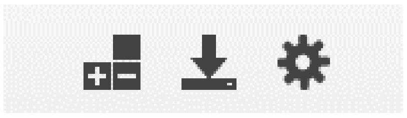


Figure 13. Maintenance Tools

3 Discovering the EmberZNet PRO Stack

If you are discovering from the initial setup process **Add …** step (Figure 8), go to step 4. If you are discovering the SDK after initial setup, begin with step 1.

1. Click the Settings icon on the top right-hand corner of the Simplicity perspective to open the Preferences window.

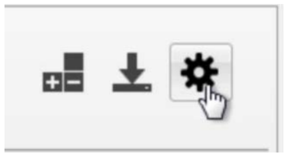
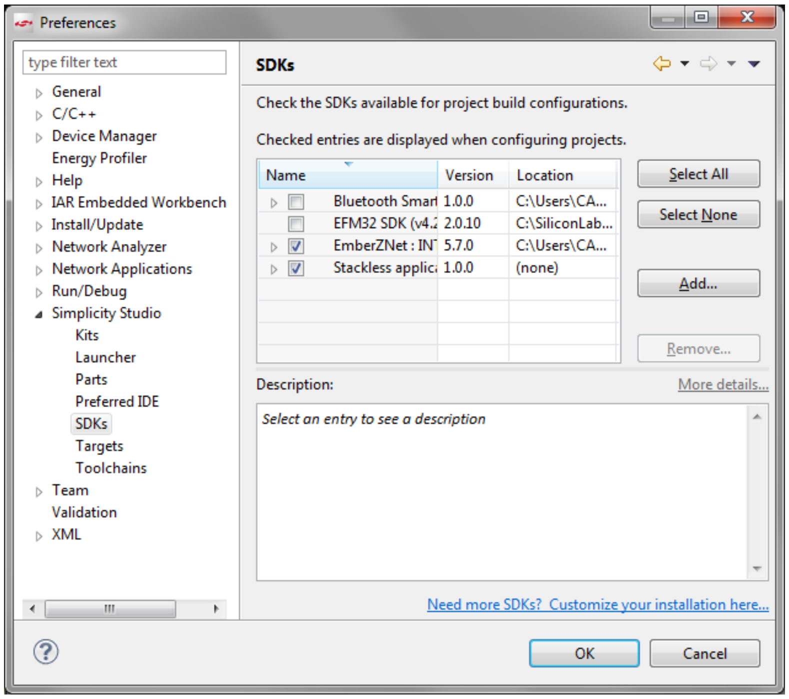


Figure 14. Settings Icon

2. In the Preferences window's left frame, click **Simplicity Studio > SDKs** to open the configuration dialog.



Check the SDKs available for project build configurations.

Checked entries are displayed when configuring projects.

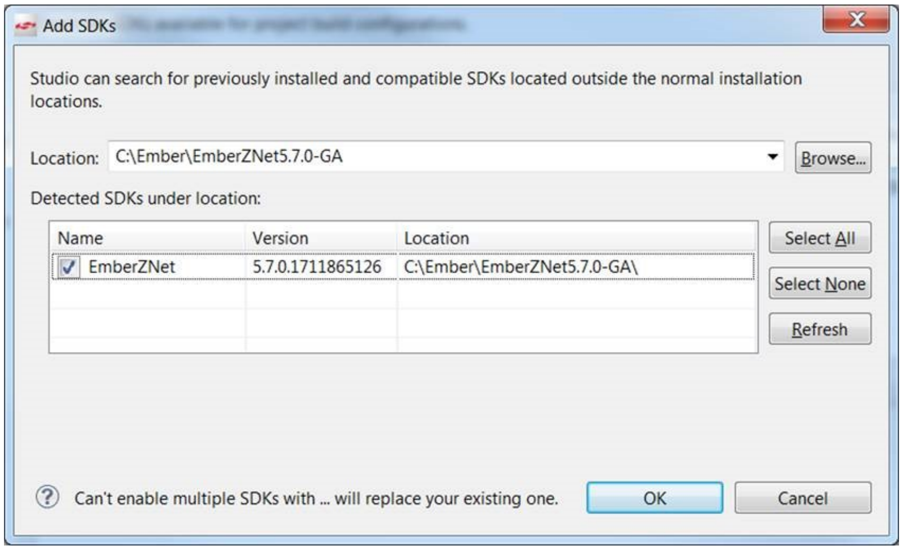
	Name	Version	Location
☐	Bluetooth Smart	1.0.0	C:\Users\CA...
☐	EFM32 SDK (v4.2	2.0.10	C:\SiliconLab...
✓	EmberZNet : INT	5.7.0	C:\Users\CA...
✓	Stackless applic	1.0.0	(none)

Select an entry to see a description

Figure 15. Silicon Labs SDK Configuration

3. Click **[Add]**.
4. Browse to the folder where you have installed the stack.
5. Click **[OK]**. The folder is scanned for required information.

6. Verify the software has detected the EmberZNet SDK and click **[OK]**.



Add SDKs

Studio can search for previously installed and compatible SDKs located outside the normal installation locations.

Location: C:\Ember\EmberZNet5.7.0-GA

Detected SDKs under location:

Name	Version	Location
✓ EmberZNet	5.7.0.1711865126	C:\Ember\EmberZNet5.7.0-GA\

Can't enable multiple SDKs with ... will replace your existing one.

Figure 16. Add SDKs

7. Click **[OK]** to return to the starting configuration page.
8. Click **[Finish]** (from initial configuration) or **[OK]** (from the settings icon)**.**

4 Working with Example Applications

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

1. Select an example application. To demonstrate the connectivity features of a network, you may need to build two or more different example applications, such as a client and a server application or the Home Automation example scenario involving a gateway, a light, and a switch.
2. Generate application files.
3. Compile and flash the application to the radio board.
4. Interact with the application.

These steps are described in detail in the following sections.

4.1 Selecting an Example Application

1. In the Simplicity perspective, click the Software Examples tile.

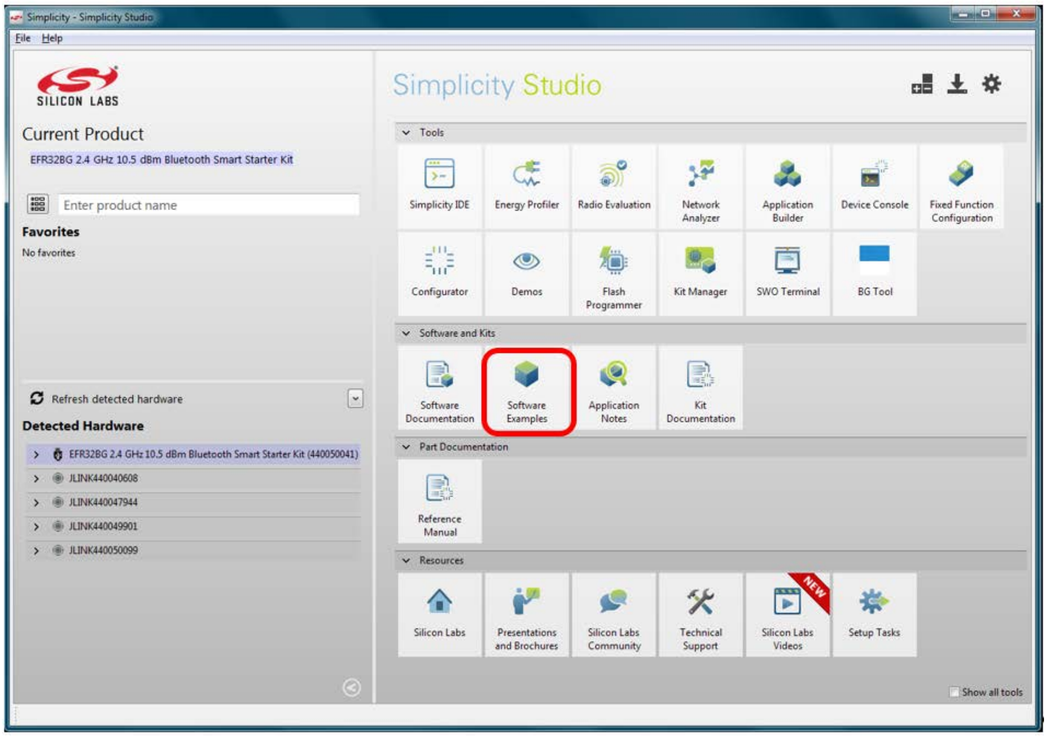
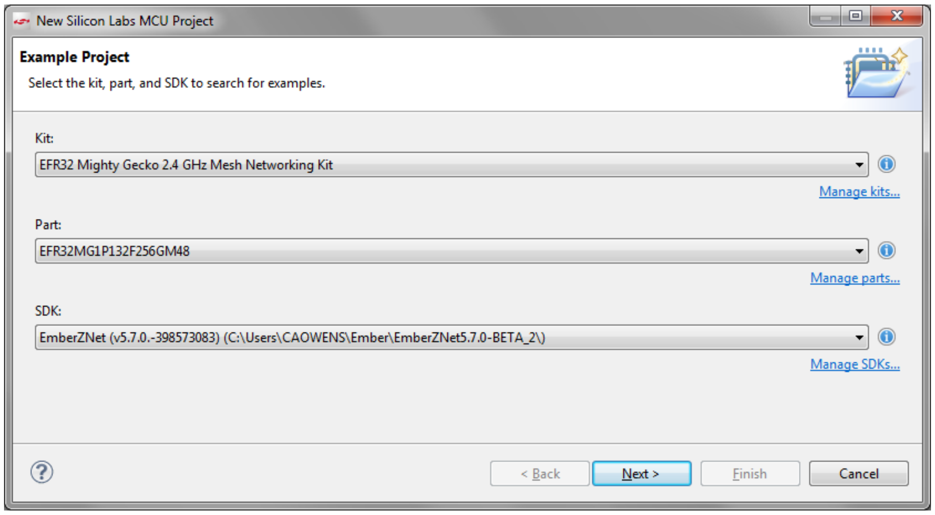


Figure 17. Software Examples Tile

2. In the New Example dialog shown below, enter the kit, part, and SDK. Kit and part will be filled by Simplicity Studio if the hardware was detected and displayed in the Simplicity perspective (Figure 11).


New Silicon Labs MCU Project

Example Project

Select the kit, part, and SDK to search for examples.

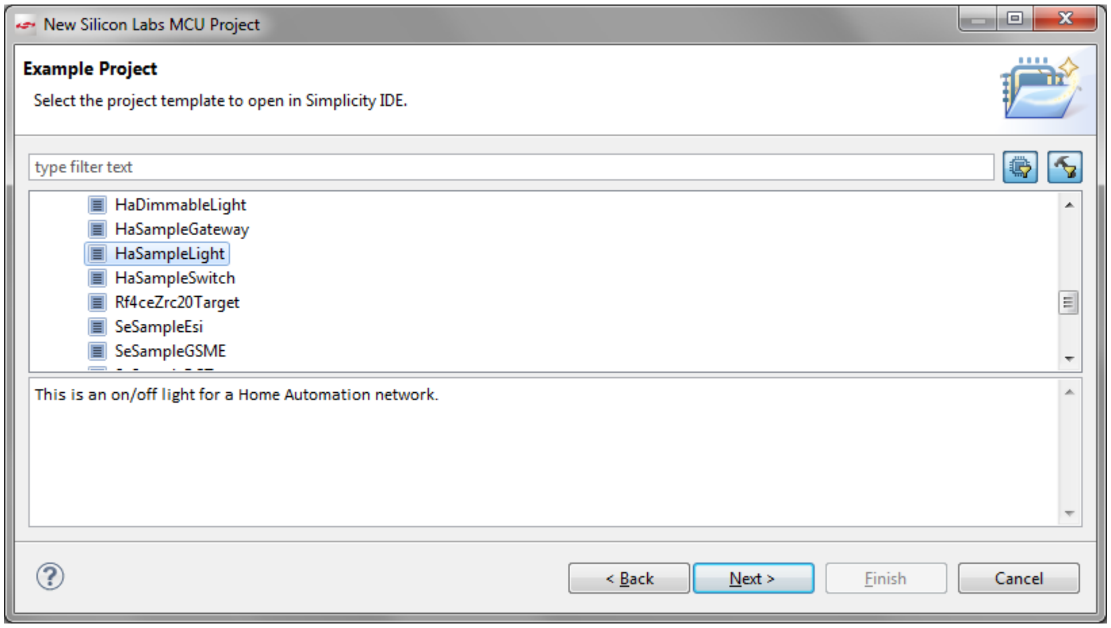
Kit: EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit

Part: EFR32MG1P132F256GM48

SDK: EmberZNet (v5.7.0.-398573083) (C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\)

Figure 18. New Example Project Dialog

3. Click **[Next]**.
4. Select an example from the list. For the purposes of this guide, select HASampleLight. Click **[Next >]**.


New Silicon Labs MCU Project

Example Project

Select the project template to open in Simplicity IDE.

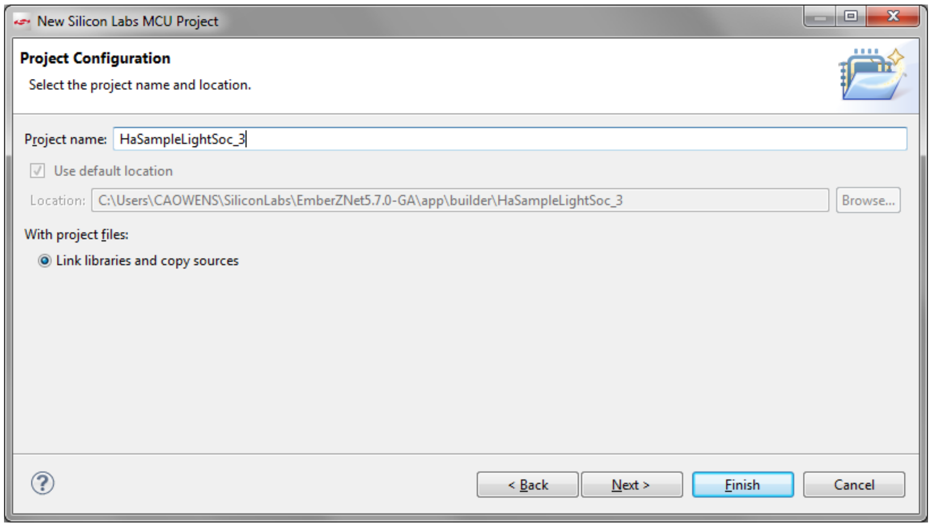
type filter text

HaDimmableLight
HaSampleGateway
HaSampleLight
HaSampleSwitch
Rf4ceZrc20Target
SeSampleEsi
SeSampleGSME

This is an on/off light for a Home Automation network.

Figure 19. Example Project List

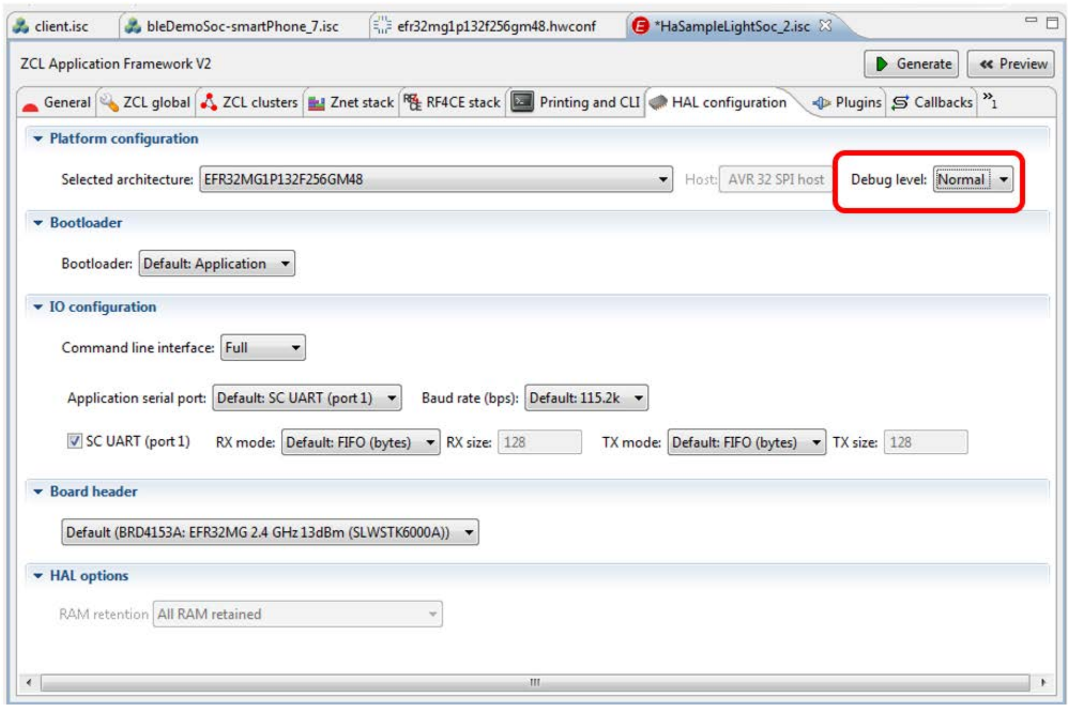
5. In the Project Configuration dialog, change the name of the project if you want, and click **[Finish]**. AppBuilder creates an example project and opens with the project in the AppBuilder General Tab (shown later in Figure 23).



Project Configuration
Select the project name and location.
Project name: HaSampleLightSoc_3
☑ Use default location
Location: C:\Users\CAOWENS\SiliconLabs\EmberZNet5.7.0-GA\app\builder\HaSampleLightSoc_3
With project files:
● Link libraries and copy sources

Figure 20. Project Configuration

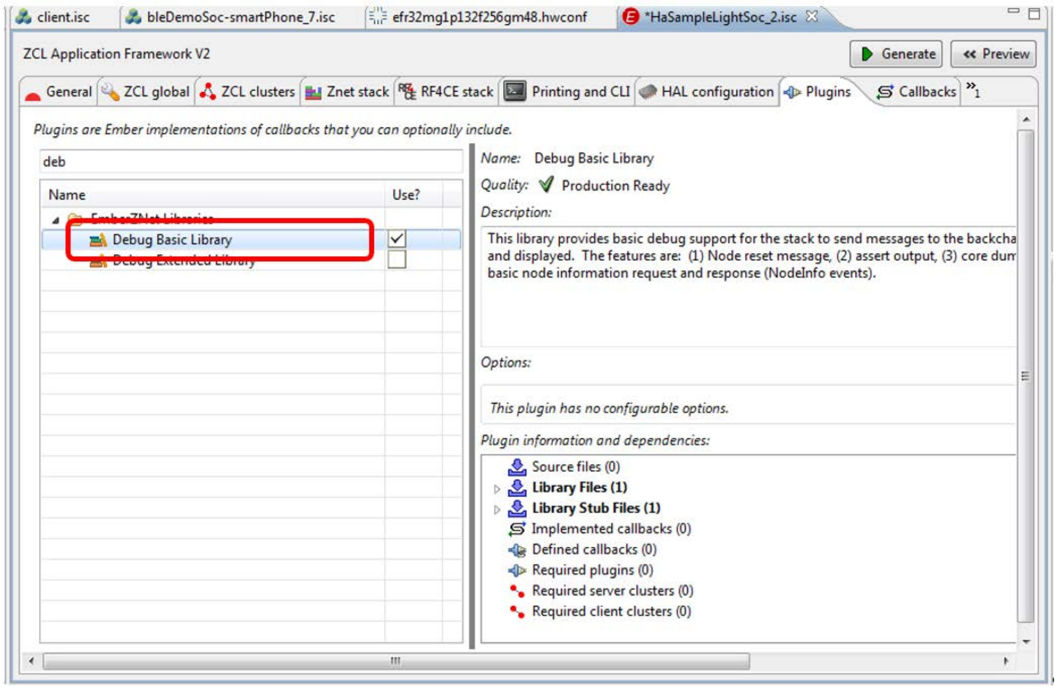
6. (Optional) If you would like to interact with your application through the serial console as instructed in section **Interacting with Your Example Application**, do the following two steps:
   1. On the HAL configuration tab, change **Debug Level** from Off to Normal.



Platform configuration
Selected architecture: EFR32MG1P132F256GM48
Host: AVR 32 SPI host
Debug level: Normal
Bootloader
Bootloader: Default: Application
IO configuration
Command line interface: Full
Application serial port: Default: SC UART (port 1)
Baud rate (bps): Default: 115.2k
☑ SC UART (port 1) RX mode: Default: FIFO (bytes) RX size: 128 TX mode: Default: FIFO (bytes) TX size: 128
Board header
Default (BRD4153A: EFR32MG 2.4 GHz 13dBm (SLWSTK6000A))
HAL options
RAM retention All RAM retained

Figure 21. HAL Configuration Tab

2. On the Plugins tab, check **Debug Basic Library**. Use the field above the plugins list to filter the list.



Plugins are Ember implementations of callbacks that you can optionally include.

deb

Name: Debug Basic Library

Quality: ✓ Production Ready

Description:

This library provides basic debug support for the stack to send messages to the backcha
and displayed. The features are: (1) Node reset message, (2) assert output, (3) core dum
basic node information request and response (NodeInfo events).

Options:

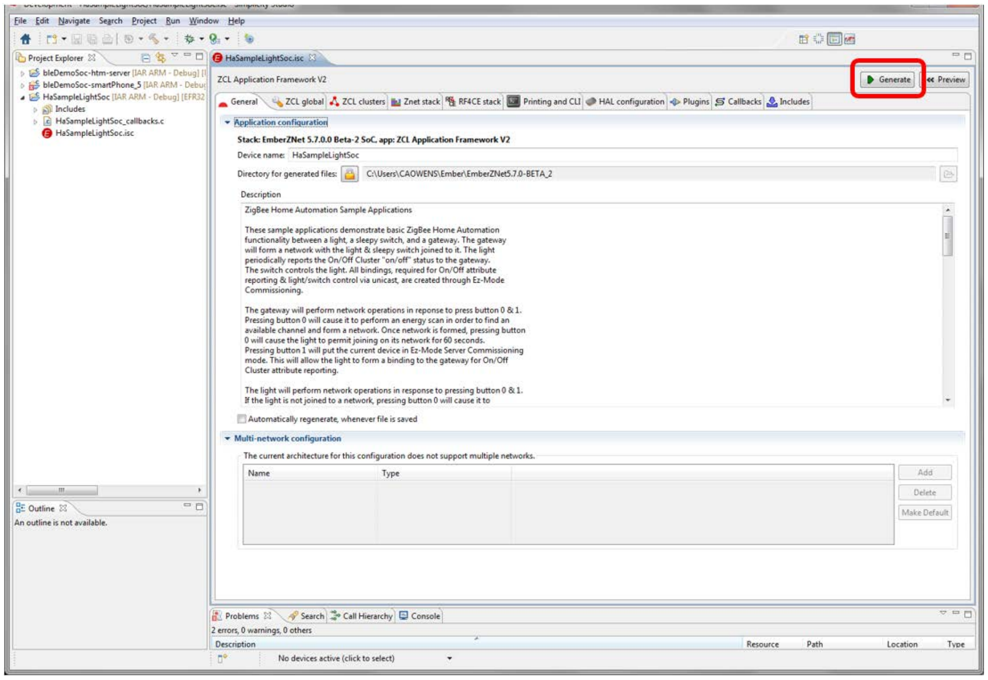
This plugin has no configurable options.

Plugin information and dependencies:

Figure 22. Plugins Tab

4.2 Generating the Application Source Files

1. When you finish creating your example project, an AppBuilder General tab opens. Click **[Generate].**



An outline is not available.

Stack: EmberZNet 5.7.0.0 Beta-2 SoC, app: ZCL Application Framework V2

Device name: HaSampleLightSoc

Directory for generated files: C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2

Description

ZigBee Home Automation Sample Applications

These sample applications demonstrate basic ZigBee Home Automation
functionality between a light, a sleepy switch, and a gateway. The gateway
will form a network with the light & sleepy switch joined to it. The light
periodically reports the On/Off Cluster "on/off" status to the gateway.
The switch controls the light. All bindings, required for On/Off attribute
reporting & light/switch control via unicast, are created through Ez-Mode
Commissioning.

The gateway will perform network operations in reponse to press button 0 & 1.
Pressing button 0 will cause it to perform an energy scan in order to find an
available channel and form a network. Once network is formed, pressing button
0 will cause the light to permit joining on its network for 60 seconds.
Pressing button 1 will put the current device in Ez-Mode Server Commissioning
mode. This will allow the light to form a binding to the gateway for On/Off
Cluster attribute reporting.

The light will perform network operations in response to pressing button 0 & 1.
If the light is not joined to a network, pressing button 0 will cause it to

Automatically regenerate, whenever file is saved

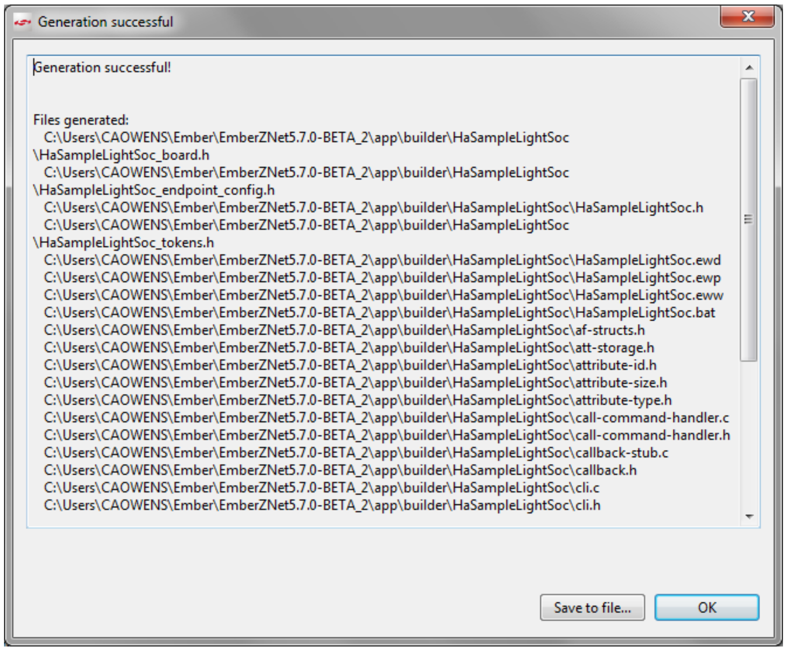
The current architecture for this configuration does not support multiple networks.

2 errors, 0 warnings, 0 others

No devices active (click to select)

Figure 23. AppBuilder General Tab

2. Once generation is complete, a dialog reporting results is displayed. Click **[OK]**.


Generation successful

Generation successful!

Files generated:
```
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc
\HaSampleLightSoc_board.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc
\HaSampleLightSoc_endpoint_config.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\HaSampleLightSoc.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc
\HaSampleLightSoc_tokens.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\HaSampleLightSoc.ewd
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\HaSampleLightSoc.ewp
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\HaSampleLightSoc.eww
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\HaSampleLightSoc.bat
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\af-structs.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\att-storage.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\attribute-id.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\attribute-size.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\attribute-type.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\call-command-handler.c
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\call-command-handler.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\callback-stub.c
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\callback.h
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\cli.c
  C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2\app\builder\HaSampleLightSoc\cli.h
```

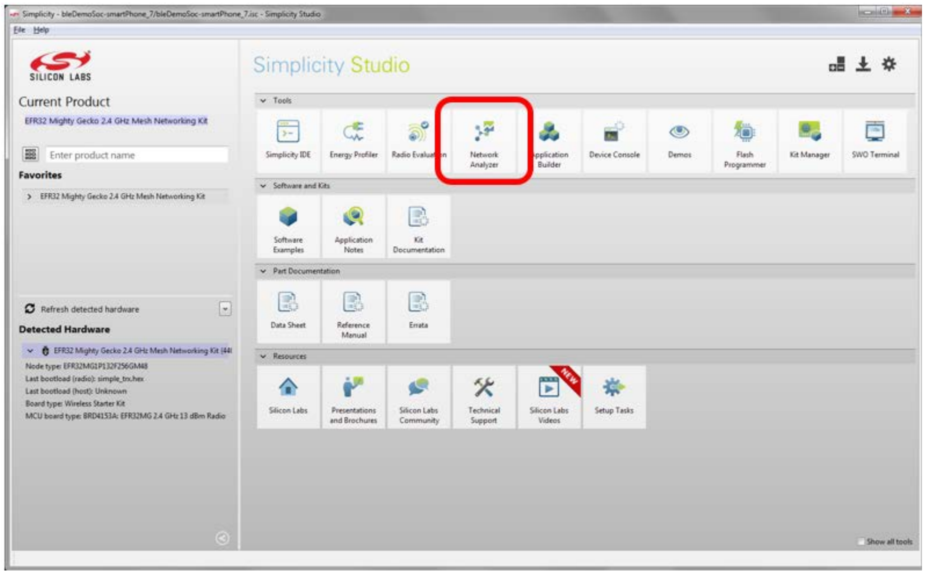
Figure 24 Generation Confirmation Dialog

4.3 Compiling and Flashing the Application

Because this sample application is built with a bootloader (see "Bootloader" option under HAL configuration tab in Figure 21), you will need to load a bootloader before you run the application for the first time. Once the bootloader is in place, unless you need to replace it with another bootloader type or version, you can update only the application. You can either compile and flash your application automatically, or manually compile it and then flash it.

4.3.1 Flashing a Bootloader

1. Connect your hardware and make sure it is detected in the Simplicity perspective.
2. Click the Network Analyzer tile.

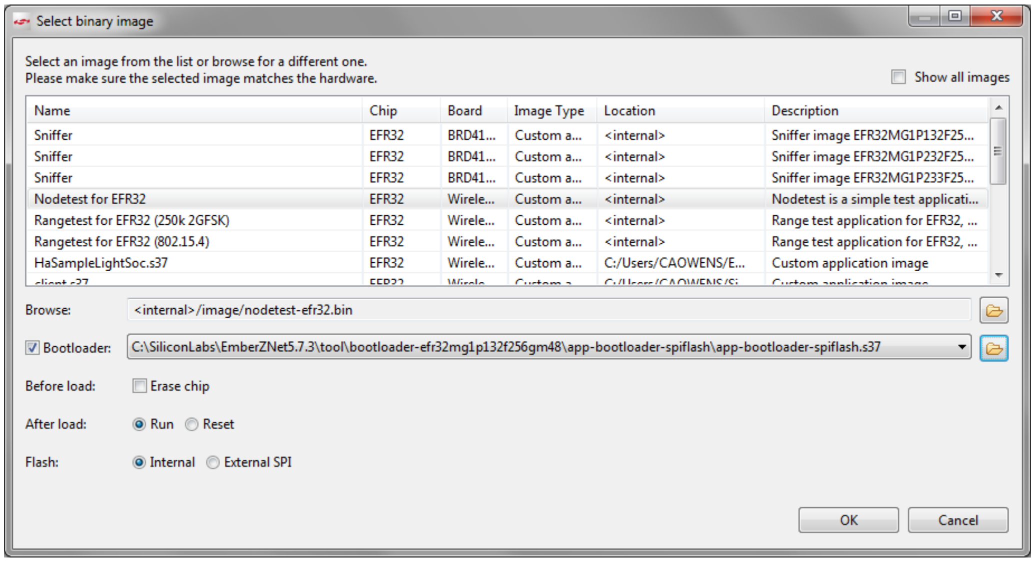

Simplicity - bleDemoSoc-smartPhone_7/bleDemoSoc-smartPhone_7.isc - Simplicity Studio

Current Product: EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit

Detected Hardware: EFR32 Mighty Gecko 2.4 GHz Mesh Networking Kit (44
Node type: EFR32MG1P132F256GM48
Last bootload (radio): simple_trx.hex
Last bootload (host): Unknown
Board type: Wireless Starter Kit
MCU board type: BRD4153A: EFR32MG 2.4 GHz 13 dBm Radio

Figure 25. Network Analyzer Tile

3. The Network Analyzer perspective opens. This perspective includes an Adapters View on the left-hand side of the screen. Right-click on your device of choice, and select Connect.
4. Right-click on the device in the Adapters view again and select **Flash / Upload**. The Select Binary Image dialog is displayed.



Select an image from the list or browse for a different one.
Please make sure the selected image matches the hardware.

Show all images

Name	Chip	Board	Image Type	Location	Description
Sniffer	EFR32	BRD41...	Custom a...	<internal>	Sniffer image EFR32MG1P132F25...
Sniffer	EFR32	BRD41...	Custom a...	<internal>	Sniffer image EFR32MG1P232F25...
Sniffer	EFR32	BRD41...	Custom a...	<internal>	Sniffer image EFR32MG1P233F25...
Nodetest for EFR32	EFR32	Wirele...	Custom a...	<internal>	Nodetest is a simple test applicati...
Rangetest for EFR32 (250k 2GFSK)	EFR32	Wirele...	Custom a...	<internal>	Range test application for EFR32, ...
Rangetest for EFR32 (802.15.4)	EFR32	Wirele...	Custom a...	<internal>	Range test application for EFR32, ...
HaSampleLightSoc.s37	EFR32	Wirele...	Custom a...	C:/Users/CAOWENS/E...	Custom application image

Browse: <internal>/image/nodetest-efr32.bin

[✓] Bootloader: C:\SiliconLabs\EmberZNet5.7.3\tool\bootloader-efr32mg1p132f256gm48\app-bootloader-spiflash\app-bootloader-spiflash.s37

Before load: [ ] Erase chip

After load: [✓ Run] Reset

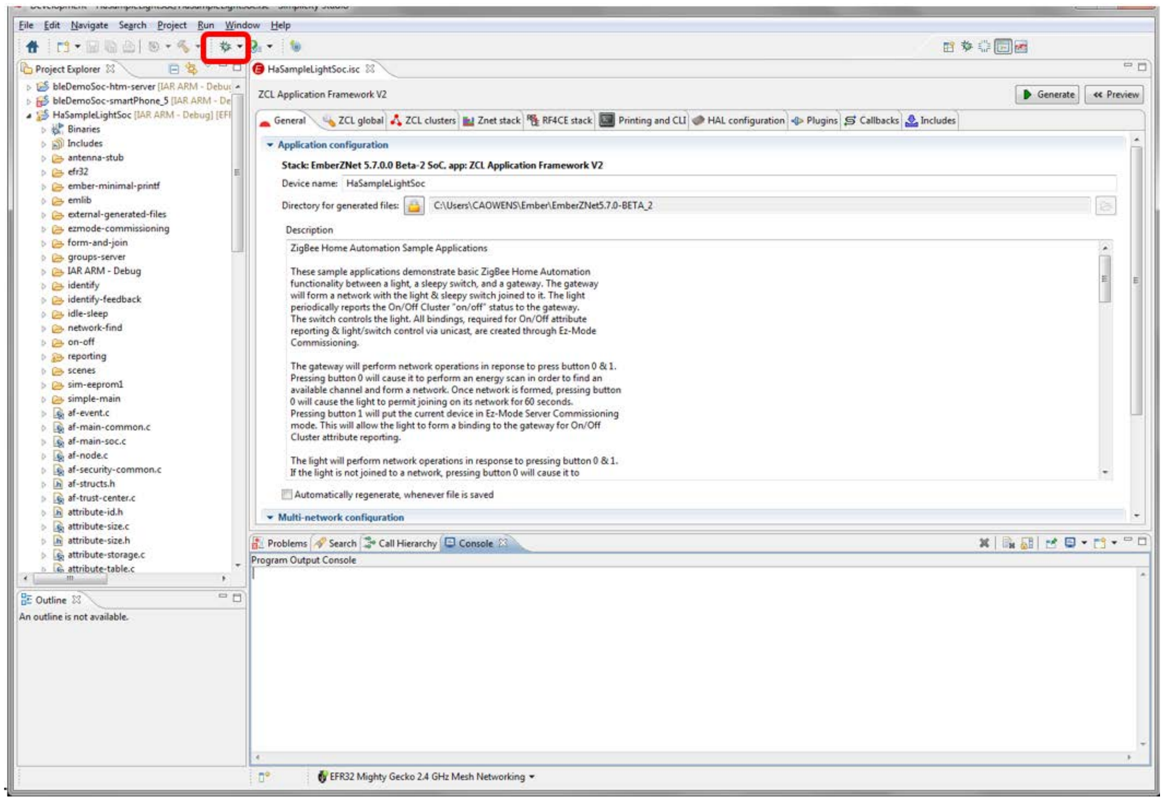
Flash: [✓ Internal] External SPI

Figure 26. Select Binary Image

5. Check **Bootloader**.
6. Navigate to the application bootloader image for your chip (in the tool folder of your installed stack).
7. Optionally, check **Erase Chip**, to make sure that the main flash block is erased before your new image is uploaded. New users will typically check this.
8. 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**.
9. Click **[OK].** You will be notified once the upload is complete.

4.3.2 Automatically Compile and Flash

1. AppBuilder can automatically compile and flash the application to your connected development hardware. After you click **[OK]** on the Generation Confirmation dialog, the AppBuilder General tab returns. Click the **Debug** control.



Stack: EmberZNet 5.7.0.0 Beta-2 SoC. app: ZCL Application Framework V2

Device name: HaSampleLightSoc

Directory for generated files: C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2

Description

ZigBee Home Automation Sample Applications

These sample applications demonstrate basic ZigBee Home Automation functionality between a light, a sleepy switch, and a gateway. The gateway will form a network with the light & sleepy switch joined to it. The light periodically reports the On/Off Cluster "on/off" status to the gateway. The switch controls the light. All bindings, required for On/Off attribute reporting & light/switch control via unicast, are created through Ez-Mode Commissioning.

The gateway will perform network operations in reponse to press button 0 & 1. Pressing button 0 will cause it to perform an energy scan in order to find an available channel and form a network. Once network is formed, pressing button 0 will cause the light to permit joining on its network for 60 seconds. Pressing button 1 will put the current device in Ez-Mode Server Commissioning mode. This will allow the light to form a binding to the gateway for On/Off Cluster attribute reporting.

The light will perform network operations in response to pressing button 0 & 1. If the light is not joined to a network, pressing button 0 will cause it to

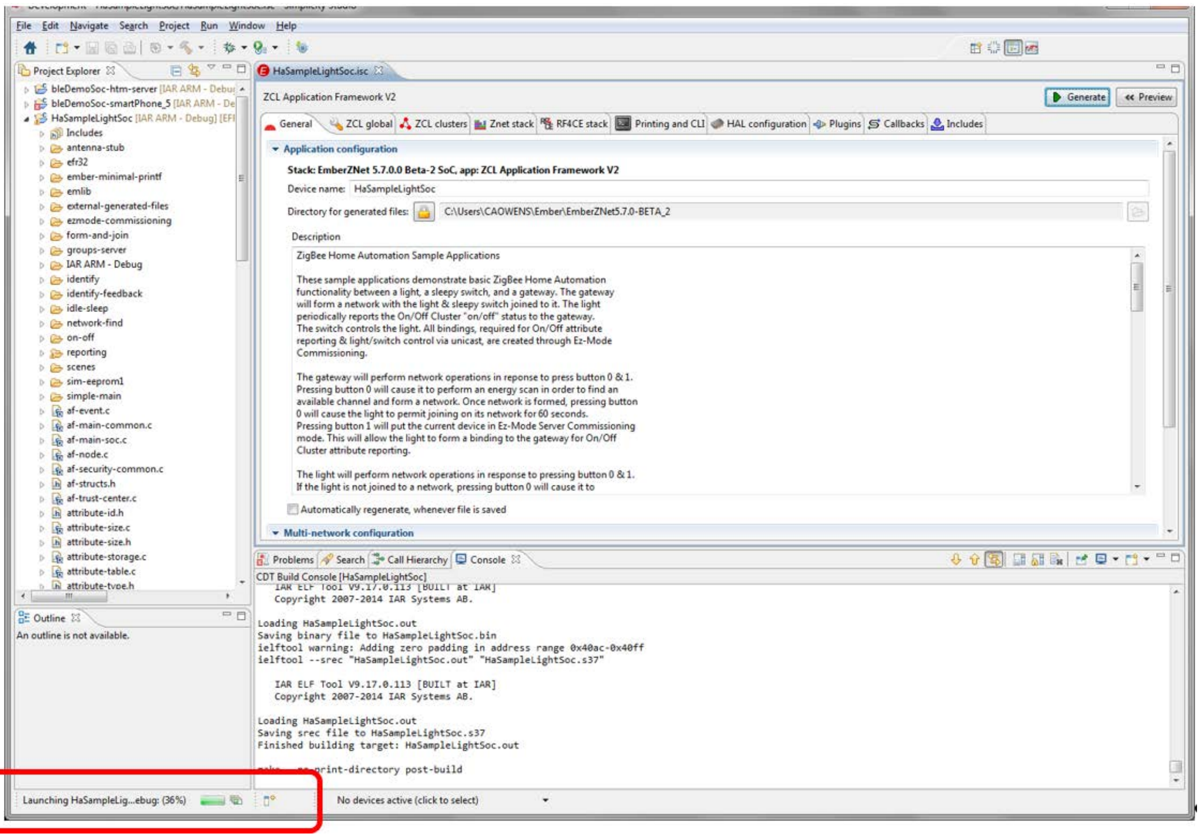
Automatically regenerate, whenever file is saved

Program Output Console

An outline is not available.

Figure 27. AppBuilder Debug Control

2. Progress is displayed in the lower left. Wait until flashing has completed and a debug window is displayed.



ZCL Application Framework V2

Application configuration

Stack: EmberZNet 5.7.0.0 Beta-2 SoC, app: ZCL Application Framework V2

Device name: HaSampleLightSoc

Directory for generated files: C:\Users\CAOWENS\Ember\EmberZNet5.7.0-BETA_2

Description

ZigBee Home Automation Sample Applications

These sample applications demonstrate basic ZigBee Home Automation functionality between a light, a sleepy switch, and a gateway. The gateway will form a network with the light & sleepy switch joined to it. The light periodically reports the On/Off Cluster "on/off" status to the gateway. The switch controls the light. All bindings, required for On/Off attribute reporting & light/switch control via unicast, are created through Ez-Mode Commissioning.

The gateway will perform network operations in reponse to press button 0 & 1. Pressing button 0 will cause it to perform an energy scan in order to find an available channel and form a network. Once network is formed, pressing button 0 will cause the light to permit joining on its network for 60 seconds. Pressing button 1 will put the current device in Ez-Mode Server Commissioning mode. This will allow the light to form a binding to the gateway for On/Off Cluster attribute reporting.

The light will perform network operations in response to pressing button 0 & 1. If the light is not joined to a network, pressing button 0 will cause it to

Automatically regenerate, whenever file is saved

Multi-network configuration

CDT Build Console [HaSampleLightSoc]

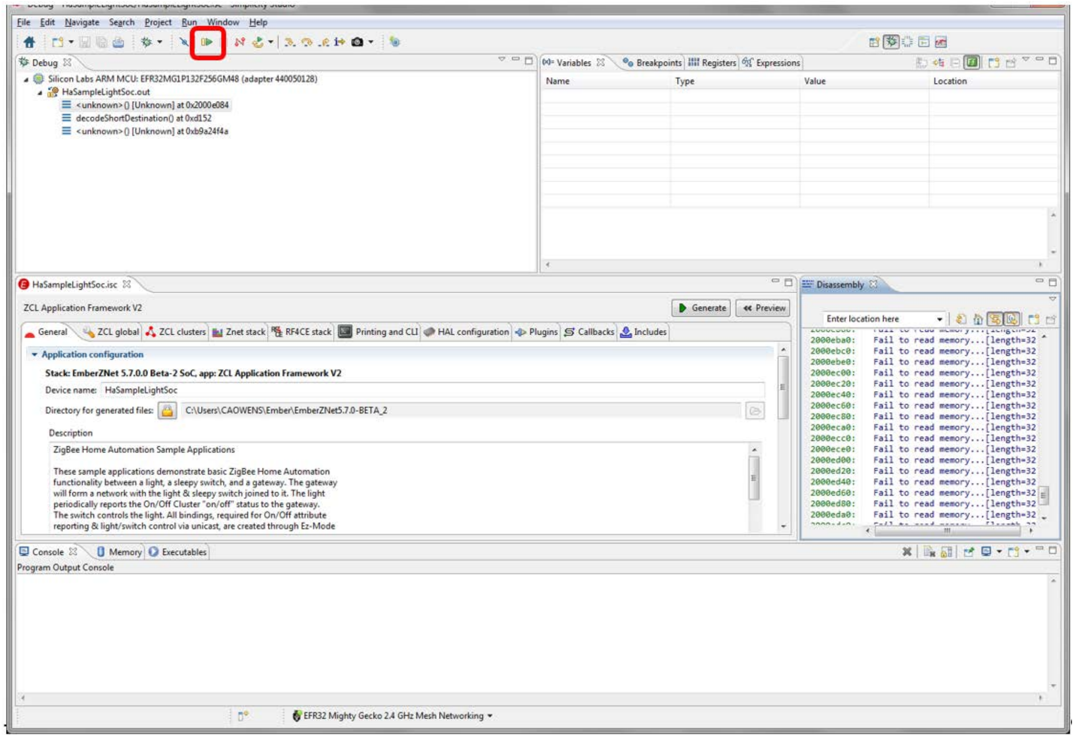
```
   Copyright 2007-2014 IAR Systems AB.

Loading HaSampleLightSoc.out
Saving binary file to HaSampleLightSoc.bin
ielftool warning: Adding zero padding in address range 0x40ac-0x40ff
ielftool --srec "HaSampleLightSoc.out" "HaSampleLightSoc.s37"

   IAR ELF Tool V9.17.0.113 [BUILT at IAR]
   Copyright 2007-2014 IAR Systems AB.

Loading HaSampleLightSoc.out
Saving srec file to HaSampleLightSoc.s37
Finished building target: HaSampleLightSoc.out

print-directory post-build
```

Launching HaSampleLig...ebug: (36%)

No devices active (click to select)

Figure 28. Compile Progress

3. In the Debug window, click the **Resume** control to start the application running on the WSTK.



Silicon Labs ARM MCU: EFR32MG1P132F256GM48 (adapter 440050128)
HaSampleLightSoc.out
<unknown>() [Unknown] at 0x2000e084
decodeShortDestination() at 0xd152
<unknown>() [Unknown] at 0xb9a24f4a

```
2000eba0:    Fail to read memory...[length=32
2000ebc0:    Fail to read memory...[length=32
2000ebe0:    Fail to read memory...[length=32
2000ec00:    Fail to read memory...[length=32
2000ec20:    Fail to read memory...[length=32
2000ec40:    Fail to read memory...[length=32
2000ec60:    Fail to read memory...[length=32
2000ec80:    Fail to read memory...[length=32
2000eca0:    Fail to read memory...[length=32
2000ecc0:    Fail to read memory...[length=32
2000ece0:    Fail to read memory...[length=32
2000ed00:    Fail to read memory...[length=32
2000ed20:    Fail to read memory...[length=32
2000ed40:    Fail to read memory...[length=32
2000ed60:    Fail to read memory...[length=32
2000ed80:    Fail to read memory...[length=32
2000eda0:    Fail to read memory...[length=32
```

EFR32 Mighty Gecko 2.4 GHz Mesh Networking

Figure 29. Debug Window - Resume Control

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

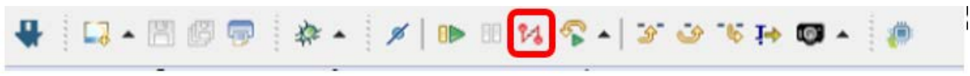


Figure 30. Disconnect Control

4.3.3 Manually Compile and Flash

After you generate your project files and are in AppBuilder, instead of clicking Debug, 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 in the Project Explorer View by right clicking on the project and going to **Build Configurations > Set Active**.

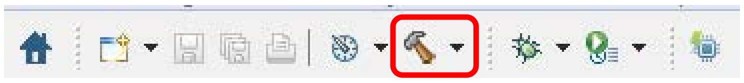
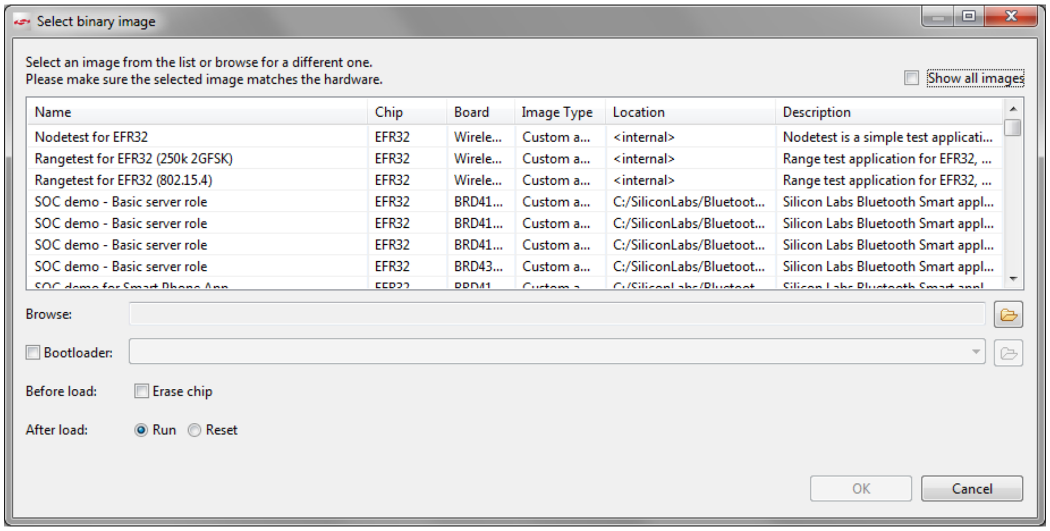


Figure 31. Build Control

The Network Analyzer perspective provides an alternate method of loading any binary image onto your device through the Adapters View.

1. Open the Network Analyzer perspective. In the Adapters View on the left-hand side of the screen, right-click on your device of choice, and select Connect.
2. Right-click on the device in the Adapters view again and select **Flash / Upload**. The Select Binary Image dialog is displayed.



Select binary image

Select an image from the list or browse for a different one.
Please make sure the selected image matches the hardware.

Show all images

Name	Chip	Board	Image Type	Location	Description
Nodetest for EFR32	EFR32	Wirele...	Custom a...	<internal>	Nodetest is a simple test applicati...
Rangetest for EFR32 (250k 2GFSK)	EFR32	Wirele...	Custom a...	<internal>	Range test application for EFR32, ...
Rangetest for EFR32 (802.15.4)	EFR32	Wirele...	Custom a...	<internal>	Range test application for EFR32, ...
SOC demo - Basic server role	EFR32	BRD41...	Custom a...	C:/SiliconLabs/Bluetoot...	Silicon Labs Bluetooth Smart appl...
SOC demo - Basic server role	EFR32	BRD41...	Custom a...	C:/SiliconLabs/Bluetoot...	Silicon Labs Bluetooth Smart appl...
SOC demo - Basic server role	EFR32	BRD41...	Custom a...	C:/SiliconLabs/Bluetoot...	Silicon Labs Bluetooth Smart appl...
SOC demo - Basic server role	EFR32	BRD43...	Custom a...	C:/SiliconLabs/Bluetoot...	Silicon Labs Bluetooth Smart appl...
SOC demo for Smart Phone App	EFR32	BRD41	Custom a	C:/SiliconLabs/Bluetoot	Silicon Labs Bluetooth Smart appl

Browse:

Bootloader:

Before load: Erase chip

After load: [✓ Run] Reset

Figure 32. Select Binary Image

3. Navigate to the .s37, .bin or .hex image you wish to upload.
4. Optionally, check **Bootloader** and browse to a bootloader image, as described in section **Flashing a Bootloader**.
5. 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.
6. 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**.
7. Click [**OK].** You will be notified once the upload is complete.

5 Interacting with Your Example Application

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, click the Device Console tile in the Simplicity perspective. Alternatively, in the Network Analyzer perspective right-click on your device in the Adapters View. Choose **Connect** (if you are not already connected) and then **Launch Console**. To get a prompt on the Console Serial 1 tab, press Enter.

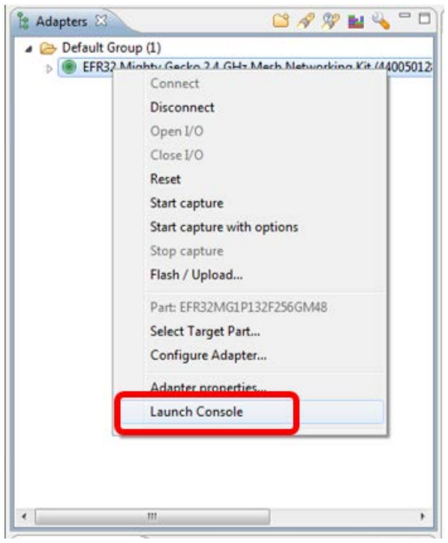


Figure 33. Launch Console

6 Add New Peripherals

In the future, if you want to use other peripherals of the EFR32 microcontroller, you can do so easily with Simplicity Studio's Hardware Configurator. When you generate the project from the template file with AppBuilder, a Hardware Configurator file is also generated and opened automatically (example interface shown in the following image). You can add your hardware changes to this model. The Hardware Configurator automatically generates initialization code for the selected hardware peripherals and port I/Os as soon as you click **[Save]**. The code is generated in InitDevice.c file, in the void `enter_DefaultMode_from_RESET(void)` function.

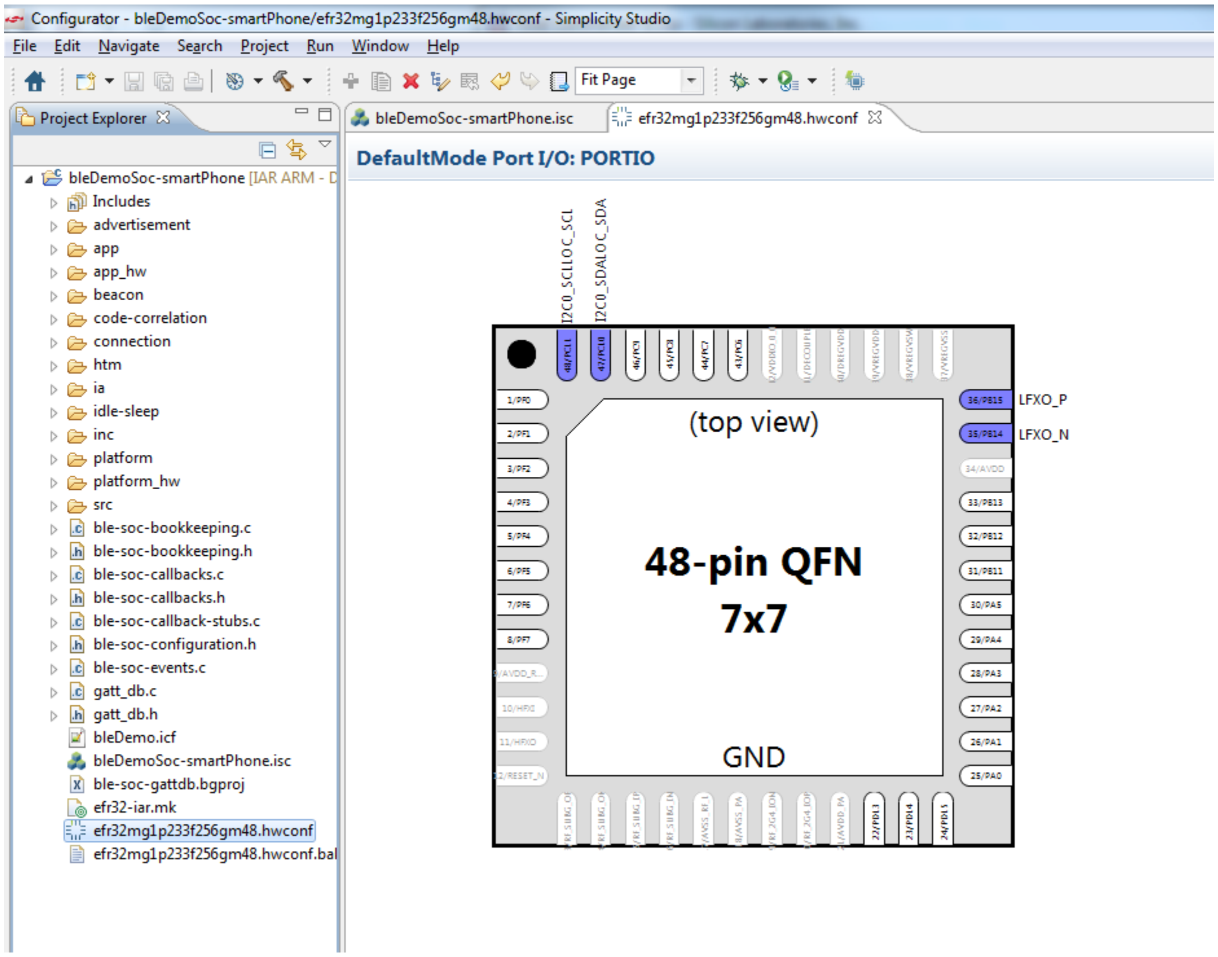
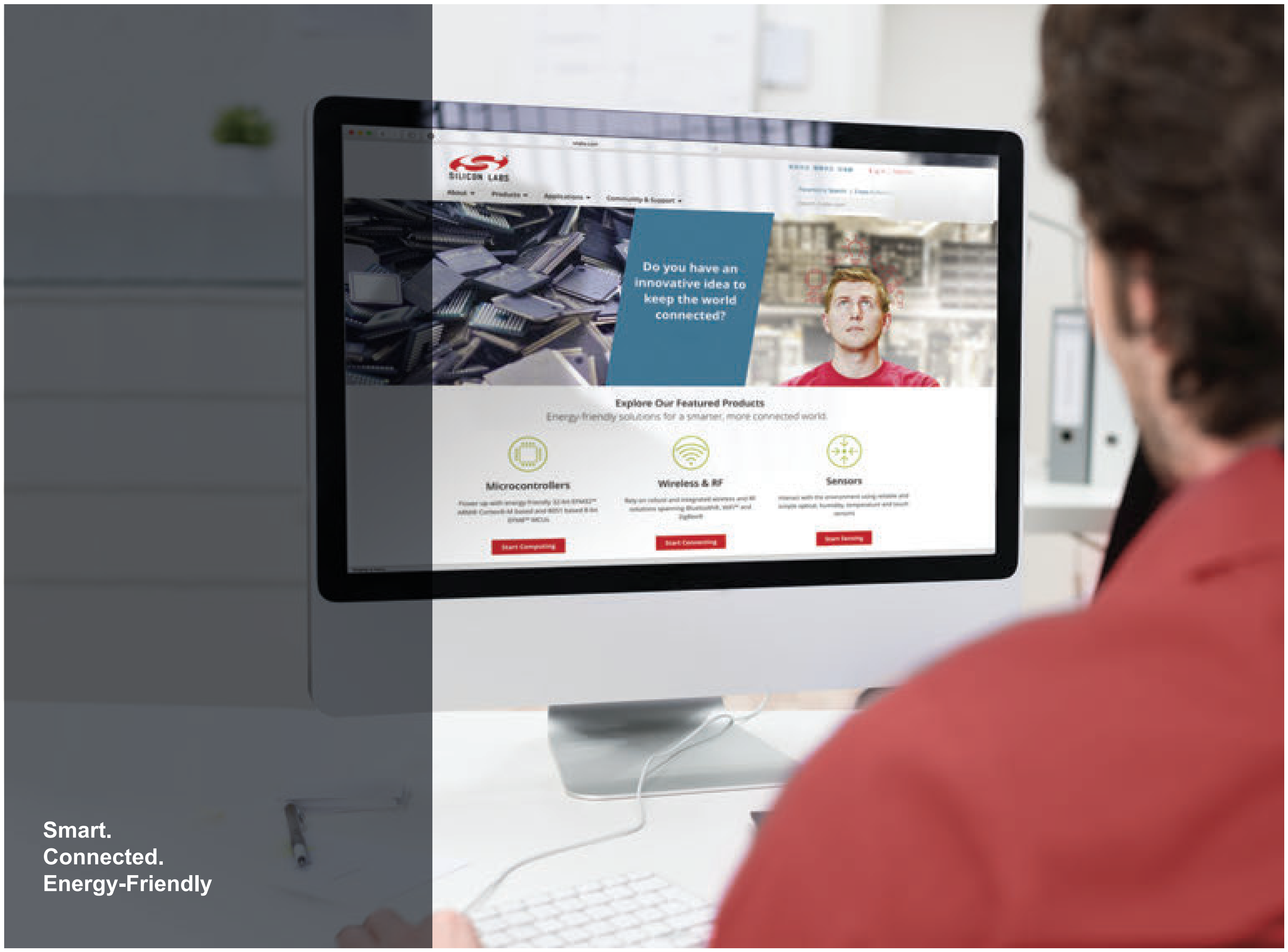


Figure 34. Hardware Configurator

Note: Some pins are greyed out and cannot be modified. These are used by the stack and the sample app's normal operation. If you would like to use any of these pins, please contact Technical Support for instructions on modifying the underlying configurator file.

For a more detailed description of the Hardware Configurator please refer to AN0823, *Simplicity Configurator User Guide*, available from the Application Notes tile in the Simplicity perspective.


Smart.
Connected.
Energy-Friendly
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world
Microcontrollers
Wireless & RF
Sensors

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

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

http://www.silabs.com