QSG105

GETTING STARTED WITH SILICON LABS WIRELESS NETWORKING SOFTWARE

This quick start guide provides basic information on configuring, building, and installing applications using the Thread, EmberZNet RF4CE, and Connect protocols. If you are developing with EmberZNet PRO, including developing a dual-protocol network using EmberZNet PRO along with EmberZNet RF4CE, see QSG106, *Getting Started with Silicon Labs EmberZNet PRO*, for similar information.

New in This Revision

Initial release.

Contents

QSG105

1 Overview

This guide is designed for new users of one the Silicon Labs development kits, such as the EM35x Development kit, who are just getting started using one of the example applications provided with the software installer. This guide assumes that you have already:

- Opened and set up your development kit as described in the Quick Start Guide (QSG) included with the kit.
- Registered your kit and downloaded the software stack as described either in the QSG itself or by going to the URL in the QSG.
- Downloaded the remaining software components as described in the release notes included with the software stack.

Depending on your software, you may find additional 'Getting Started' software detail in other documents.

Connect

AN888: EZR32 Simple TRX Application Quick Start Guide

AN889: Silicon Labs Connect Quick Start Guide

RF4CE

The Quick Start Guide included in both EM34x development kit variants also has information on instructions for loading a prebuilt sample application.

1.1 Support

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

1.2 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. Documents are also available through the support portal as Adobe PDF files.

1.3 Software Components

1.3.1 Network Stack

The network stack is an advanced implementation of a wireless protocol stack. The network stack API is documented in online API reference as well as other documents installed with the stack installer, or available through the development environment. The network 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.

1.3.2 Development Environment

Along with the stack, Silicon Labs provides a development environment, Ember Desktop or Simplicity Studio. If the stack uses Ember Desktop, a separate compiler, IAR-EWARM, is also required. Simplicity Studio has an integrated compiler. Both Ember Desktop and Simplicity Studio contain two tools that you will use to get started: Application Builder and Network Analyzer.

Application Builder, sometimes referred to as AppBuilder, is an interactive GUI tool that allows you to configure a body of Silicon Labs-supplied code to implement devices, such as a ZigBee Smart Energy meter or a Connect temperature sensor, within an application profile.

Network Analyzer manages the Development Kit hardware and displays network and node activity in real time. It includes:

- Provides tiered views of network activity, letting you drill down from a high-level map of node interactions to the details of each packet.
- Customizable filters that let you specify exactly which network activities to display.
- Log files that save captured data, so you can step through transactions and events for detailed analysis.

For detailed information about these tools and the development environment itself, refer to its online help.

1.3.3 Application Frameworks

Application frameworks are, in essence, working software designs that you can modify to meet your needs using Application Builder. They provide 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 with each Silicon Labs stack release. A pre-built application, Node Test, is also provided for functional testing of RF modules. This guide covers getting started with application frameworks for Thread, single-protocol RF4CE networks, and Silicon Labs Connect. The AFV2 application framework, based around the ZCL (ZigBee Cluster Library), is for use when developing ZigBee PRO applications with the EmberZNet stack, including dual protocol networks using both EmberZNet PRO and EmberZNet RF4CE. AFV2 has a different development workflow that is documented in QSG106, *Getting Started with Silicon Labs EmberZNet PRO*.

1.3.4 Hardware Abstraction Layer

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.

2 Building a Sample Application

To build a sample application you must:

1. Configure Application Builder so it can find the stack you are using.
2. Create a new application based on a sample
3. Generate the application files
4. Build the application
5. Load the application onto your device.

You can then interact with the application on the device to form a network or send data.

For some steps, the procedure varies slightly depending on whether you are working with Ember Desktop or Simplicity Studio. For these steps, two sets of instructions are provided.

2.1 Configure Application Builder with Your Stack

1. Open the Preferences window.

   Ember Desktop: Go to **File > Preferences**.
   Simplicity Studio: Click the gear icon in the top right-hand corner of the window.

2. In the Preferences window's left frame, click **Network Applications** to open the configuration dialog for Application Builder (Figure 1)

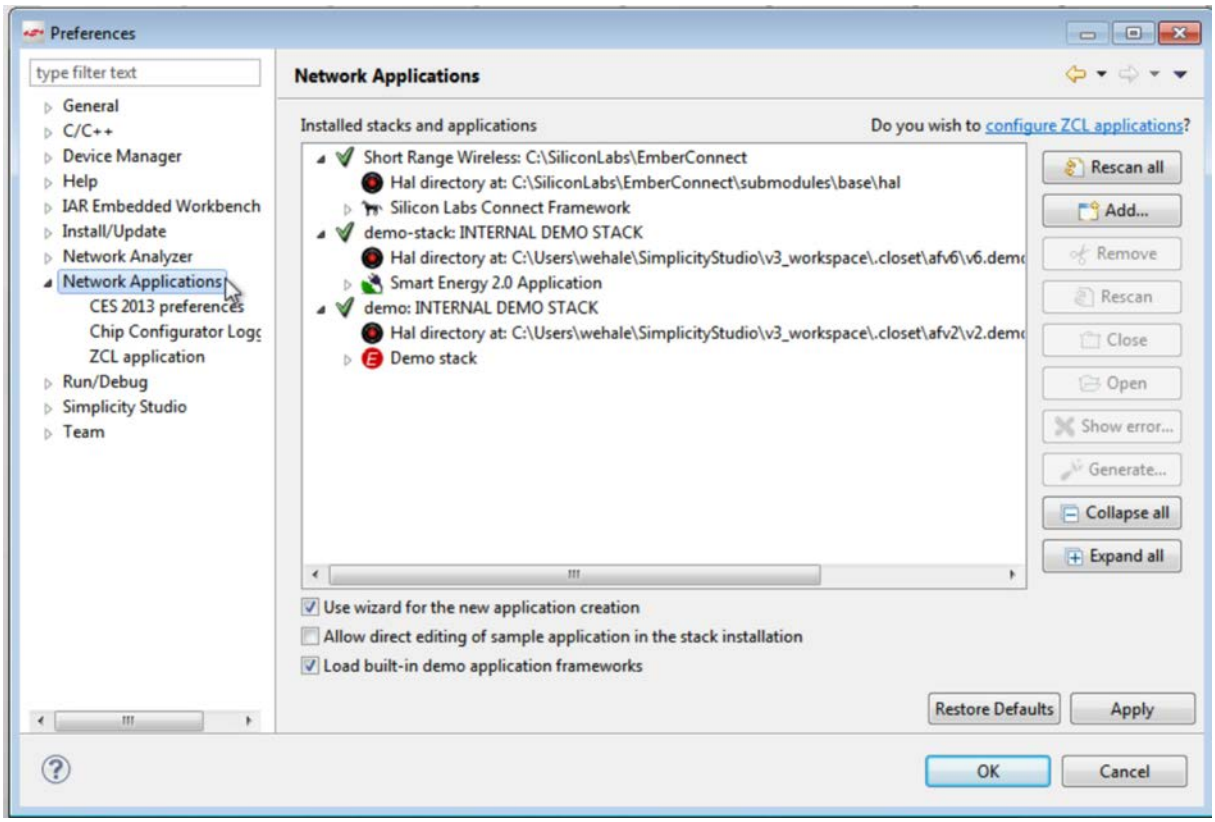

Figure 1. Application Builder Configuration

3. Click **Add**.
4. Browse to the location where you have installed the stack.
5. Click **Open**.
6. Verify the list of installed stacks now contains now contains your stack.
7. Click **OK**.

2.2 Create a New Application Based on a Sample

1. Select a sample application

In Simplicity Studio:

1. In Application Builder, go to **File > New > Project > Silicon Labs AppBuilder Project…**
2. In the Applications and Stacks window, select the framework you will be using.
3. Click **Next >**.
4. In the Select Sample Application dialog, uncheck `Start with blank application.`
5. Select a sample application.
6. Click **Next >.**

In Ember Desktop:

1. Go to **File > New > Application Framework Configuration...**
2. Select an Application Type.

3. Click **Next >**.
4. Select the stack you loaded in the first step.
5. Click **Next >**.
6. Click `Start from a selected sample application`.
7. Select one of the provided sample applications.
8. Click **Next >**.

2. In the Project Information window, specify a location for your application.

   Note: the location must be on the same Windows partition as the stack.

3. Specify a name for your application.
4. Click **Finish**.

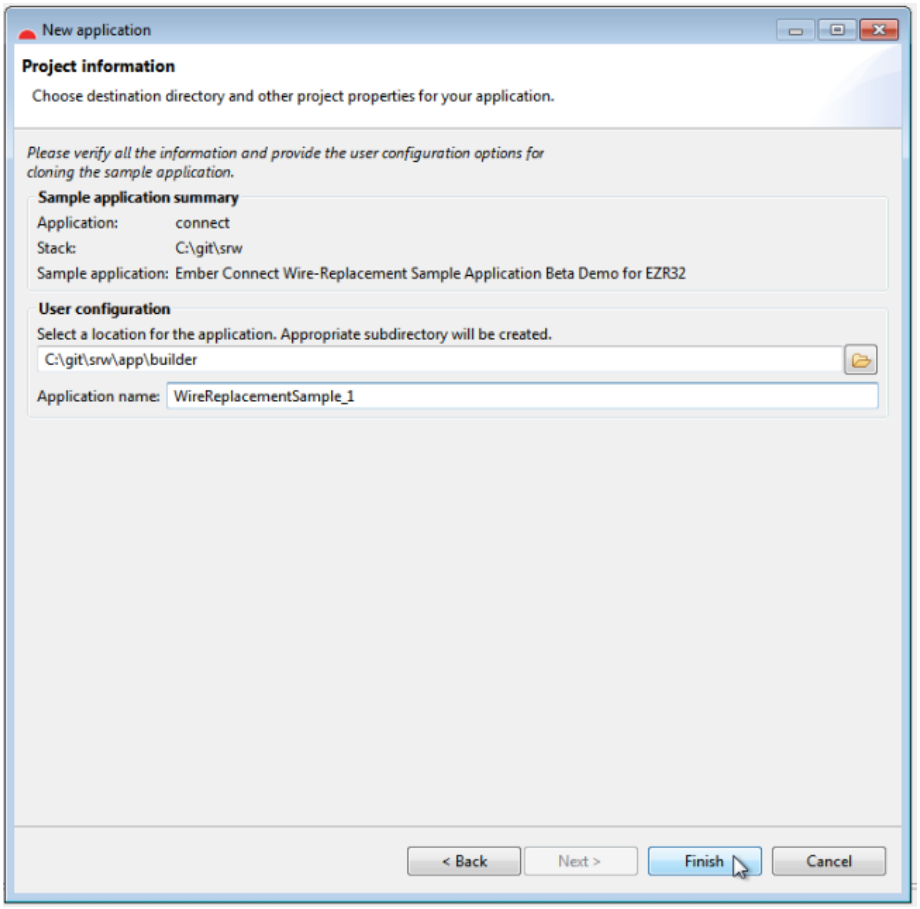

Figure 2. Project Information Window

Simplicity Studio

By default Simplicity Studio loads with the platform of your connected starter kit. To modify the platform for your application, click **Next >** Instead of **Finish** (Figure 3). This dialog also shows the various build configurations available.

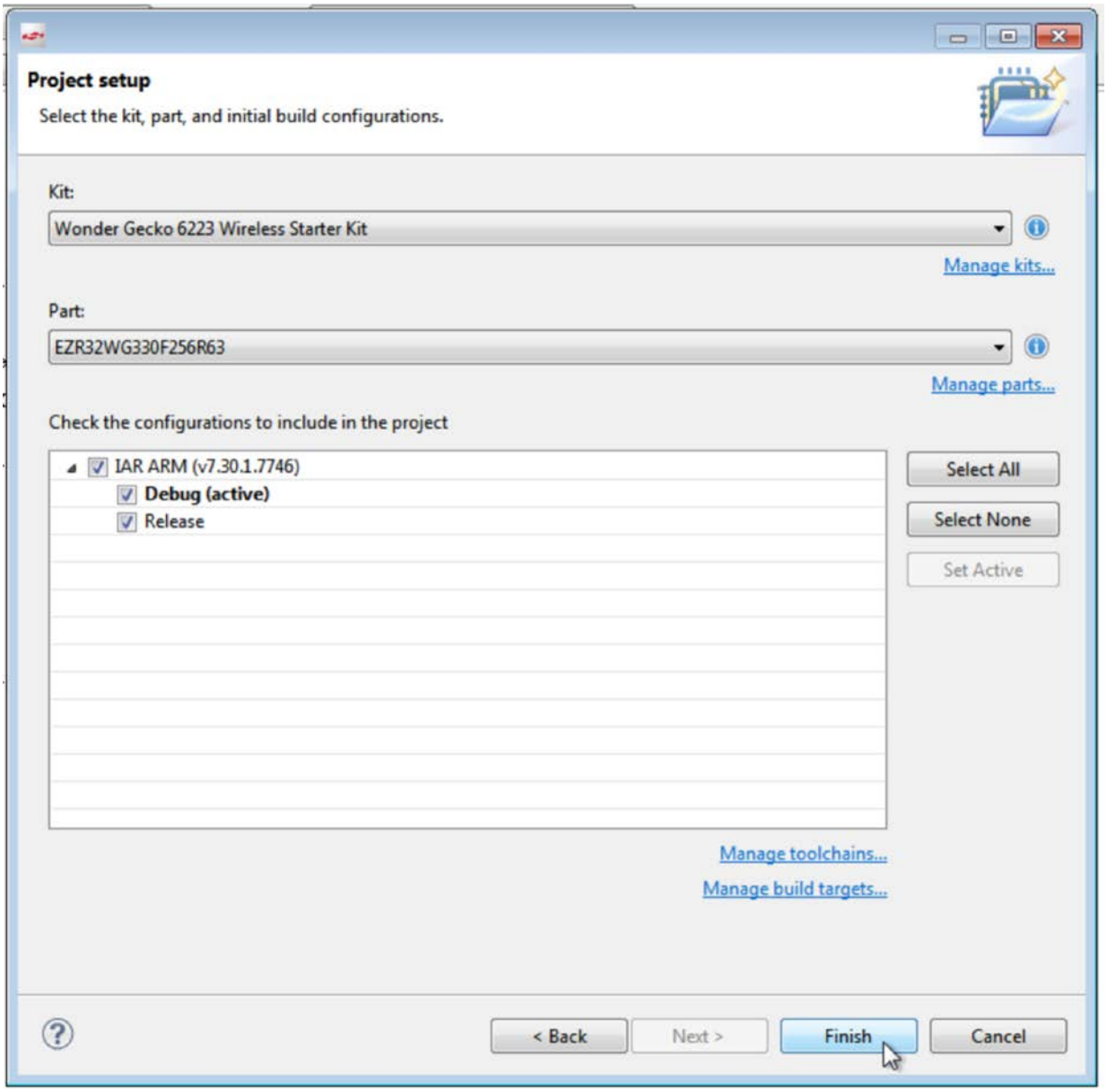

Figure 3. Platform Configuration

Note: You must have a Toolchain and Build target selected and configured for the **Finish** button to enable. If you do not see the **Finish** button enabled, check your Toolchains and Build targets by clicking on the links at the bottom of the dialog.

2.3 Generate the Application

When you finish creating your sample application, an Application Builder General tab opens (example shown in Figure 4).

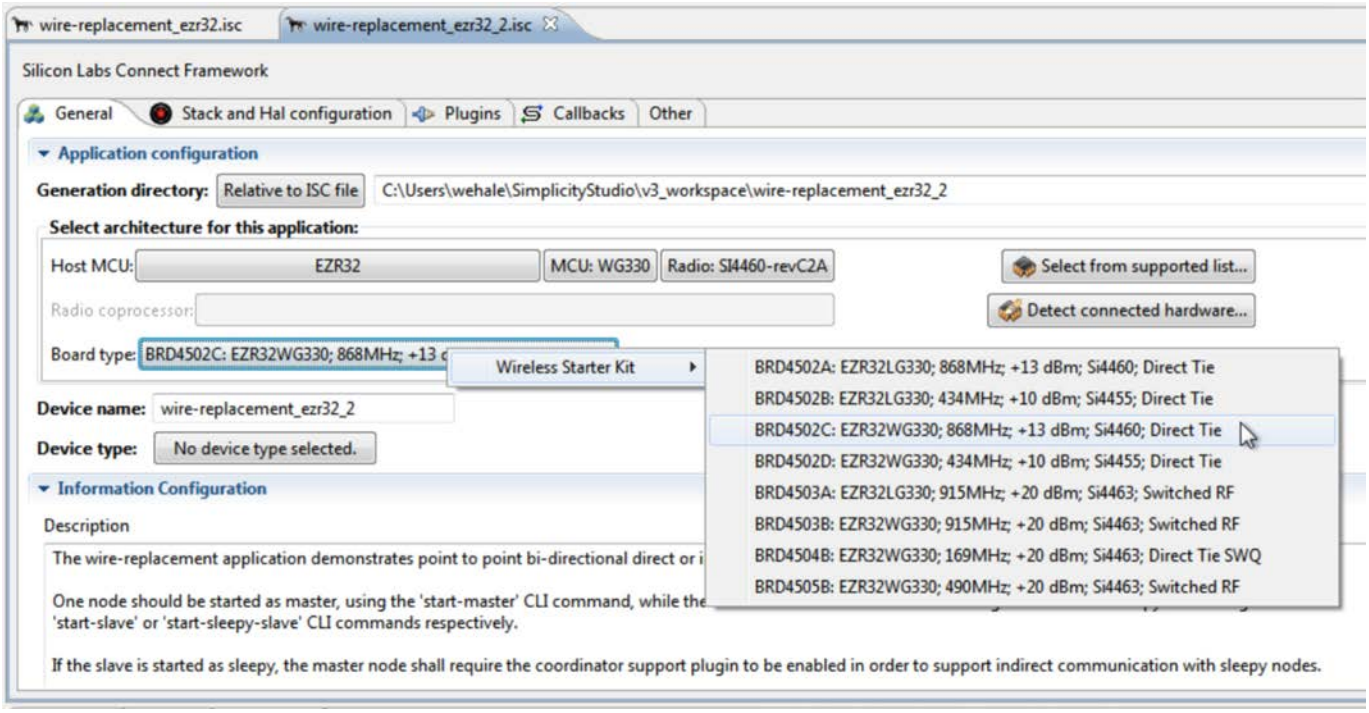

Figure 4. Application Builder General Tab

1. If the architecture parameters shown for MCU and Radio, and Board Type are not correct for your target device, change them by clicking on their respective button and selecting the correct value from the list.
2. Select your chosen module card from the Board Type List in the General Configuration Tab. This ensures that the generated PHY configuration is optimized for your radio board.
3. Click **Generate** in the upper right corner of the tab.
4. An overwrite dialog is presented. **CAUTION:** If you are starting from a sample application, uncheck the box for the callbacks file. The implementation of the application is contained in the callbacks file, and overwriting it will remove the sample functionality.
5. When generation is complete a dialog shows the generated files, one of which has the extension .eww. Click **OK.**

2.4 Build the Application

With Ember Desktop

1. Open IAR-EWARM.
2. Open the .eww file.
3. Build the application.

With Simplicity Studio

Click **Build** 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**.

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

2.5 Load the Binary onto your Device

Note: If you are using a wireless starter kit (WSTK), make sure that the power switch on your WSTK board is set in the “AEM” position when attempting to flash your device.

1. Go to the Adapters View in either the Application Builder Perspective or in the Network Analyzer Perspective.
2. Right-click on your device of choice, select **Connect** and then **Upload**. The Select Binary Image dialog is displayed (Figure 5).
3. Navigate to the .bin or .hex image you wish to upload.
4. Click **OK.**

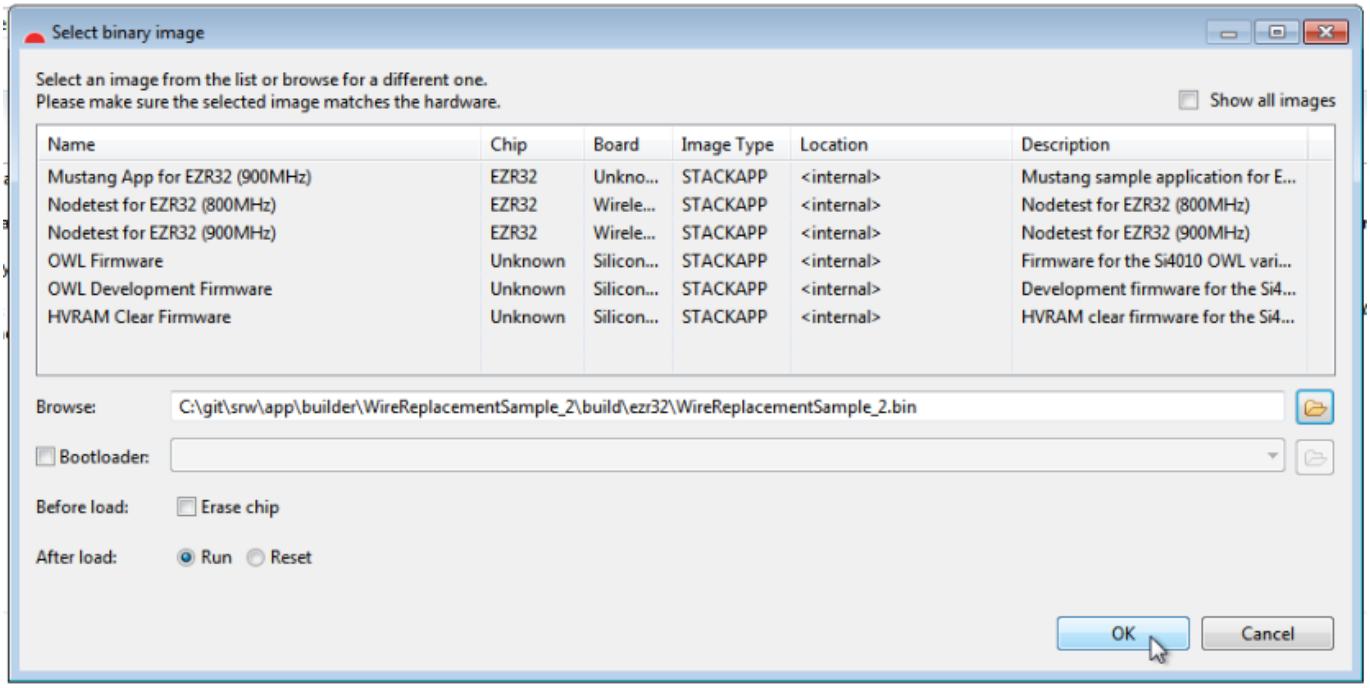

Figure 5. Uploading .bin or .hex Images

Simplicity Studio

If you have compiled your application in Simplicity Studio, you will notice some images listed under the Binaries menu item in the Project Explorer View. You can flash your image to a device by right-clicking on the image you wish to load and selecting **Flash to device**.

3 Interacting with your Sample Application

You can interact with your loaded sample application 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.

Right-click on your adapter in the Adapters View. Choose **Connect** and then **Launch Console**.

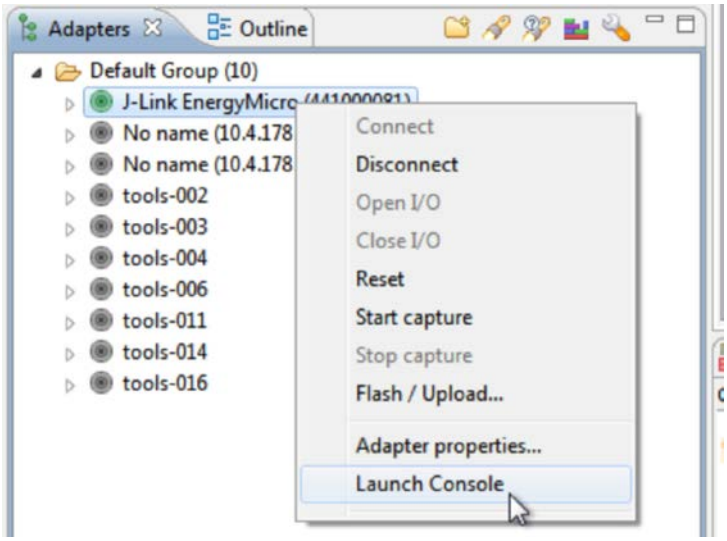

Figure 6. Launch Console

Each sample application includes information on the basic use of the Command Line Interface in the Sample Application Description displayed on the General Tab of your .isc file in the Application Builder perspective (Figure 7). Please refer to this documentation for more information on how to run the sample application you have chosen.

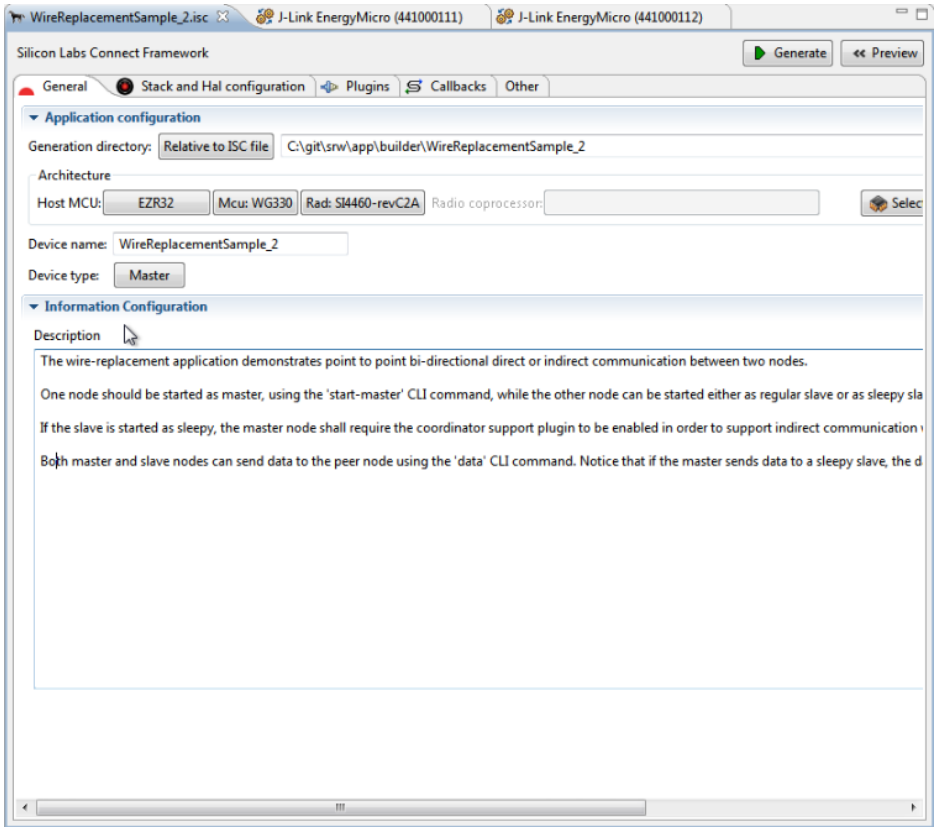

Figure 7, CLI Information

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

For additional information please visit the Silicon Labs Technical Support page:
http://www.silabs.com/support/Pages/default.aspx