UG343: Multi-Node Energy Profiler User's Guide

Multi-Node Energy Profiler enables you to visualize the energy consumption of individual devices, multiple devices on one target system, or a network of interacting wireless devices in order to analyze and improve the power performance of these systems. Real-time information on current consumption is correlated with the program counter providing advanced energy software monitoring capabilities. It also provides a basic level of integration with the Network Analyzer network analysis tool.

KEY POINTS

- Analyze power consumption of entire systems
- Profile energy use in relation to code execution
- Investigate network communications in relation to power consumption

Table of Contents

1. Introduction . . . 3

1.1 Terms and Definitions . . . 3

2. Starting an Energy Analysis Session . . . 4

2.1 Starting an Energy Analysis Session with an SDK Example from the Simplicity IDE . . . 4

2.2 Starting an Energy Analysis Session from Energy Profiler . . . 6

2.3 Customer Hardware and Software Design Information . . . 7

2.3.1 Hardware Design . . . 7

2.3.2 Software Configuration . . . 7

3. Multi-Node Energy Profiler User Interface Overview . . . 11

3.1 Energy Analysis Views . . . 11

3.1.1 Single-Node View . . . 12

3.1.2 Multi-Node View . . . 13

3.1.3 Scope View . . . 15

3.2 Selecting Ranges . . . 19

4. Energy Statistics . . . 20

4.1 Energy Statistics Configuration . . . 21

4.2 Save Snapshot of Session Statistics . . . 22

4.3 Review Saved Session Statistics . . . 23

5. Play and Record Data Control . . . 24

5.1 Play Control . . . 25

5.2 Record Control . . . 25

6. Freeze and Record Triggers . . . 26

6.1 Trigger States . . . 27

6.2 Freeze Trigger . . . 28

6.3 Record Start/Stop Triggers . . . 29

7. Search Capability . . . 30

8. Profiling with Code Correlation . . . 32

8.1 Enabling Code Correlation Profiling . . . 32

8.1.1 Code Correlation Enabled from the Simplicity IDE . . . 32

8.1.2 Code Correlation Enabled from the Multi-Node Energy Profiler . . . 34

8.2 Code Correlation View . . . 35

8.3 Color Coding and Code Correlation . . . 36

9. Multi-Node Energy Profiler and Network Analyzer Integration . . . 37

9.1 Use Cases for Multi-Node Energy Profiler and Network Analyzer . . . 39

9.2 Post Analysis Using an ISD File . . . 40

1. Introduction

Energy saving and efficiency are at the top of developers' priorities for an ever-growing number of embedded systems applications. These constraints can be due to government regulations (for example, metering applications), a requirement to increase battery life, or simply a need to lower the electricity bill. In battery-operated systems, energy efficiency often plays the most important role. In cases where developers are satisfied with their system's battery life, increasing the energy efficiency means they can switch to a smaller and cheaper battery, which will lower the overall cost. There are also situations where the operating life must be extended to the absolute maximum, such as products where the battery cannot be replaced or replacement involves very high costs.

Having a low power MCU by itself does not mean you will have lower energy consumption. The trick is to optimize your software not just in terms of functionality, but also with respect to energy efficiency. Having full control of the hardware surrounding the MCU and optimizing the overall software and peripheral usage are crucial factors for reducing system energy consumption. Software is not usually seen as an energy drain, yet every clock cycle consumes energy. Minimizing the number of clock cycles becomes a key challenge in order to reduce overall system consumption.

Multi-Node Energy Profiler enables you to visualize the energy consumption of individual devices, multiple devices on one target system, or a network of interacting wireless devices in order to analyze and improve the power performance of these systems. Real-time information on current consumption is correlated with the program counter providing advanced energy software monitoring capabilities. It also provides a basic level of integration with the Network Analyzer network analysis tool.

1.1 Terms and Definitions

AEM: Advanced Energy Monitor

ISD: This is the file extension used on Multi-Node Energy Profiler and Network Analyzer files.

PTI: Packet Trace Information

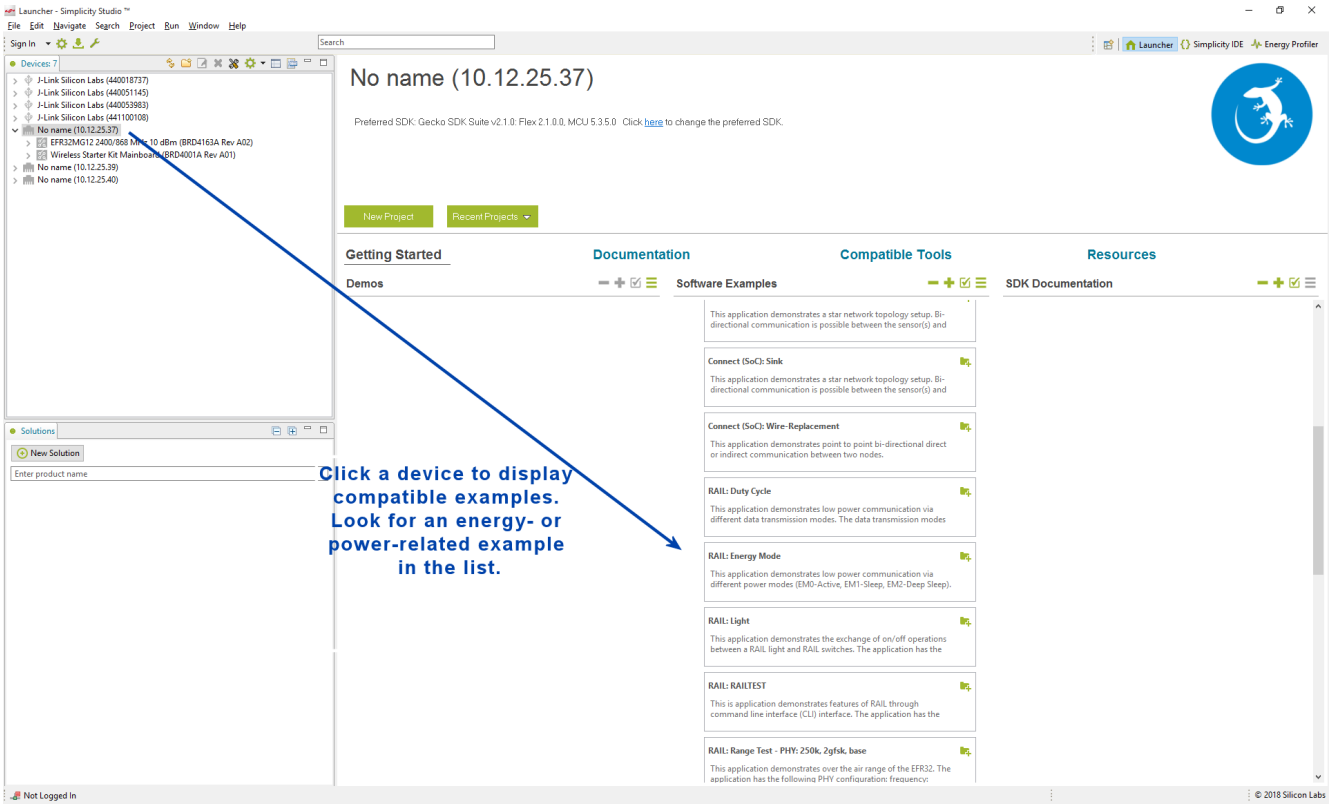
SDK: Software Development Kit

2. Starting an Energy Analysis Session

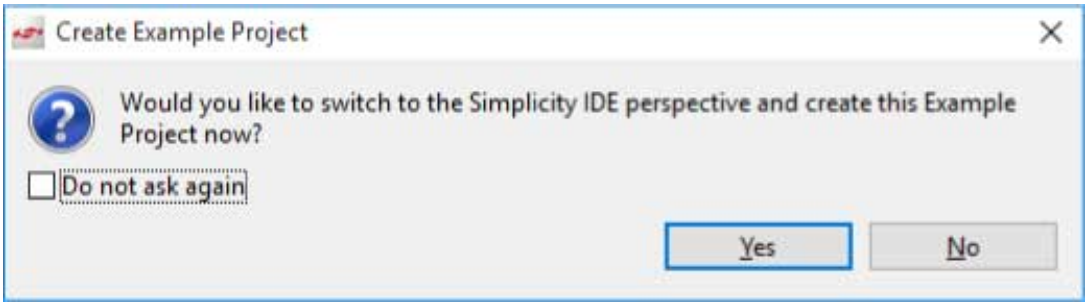
An energy analysis session can be started from either the Simplicity IDE or from within Multi-Node Energy Profiler.

2.1 Starting an Energy Analysis Session with an SDK Example from the Simplicity IDE

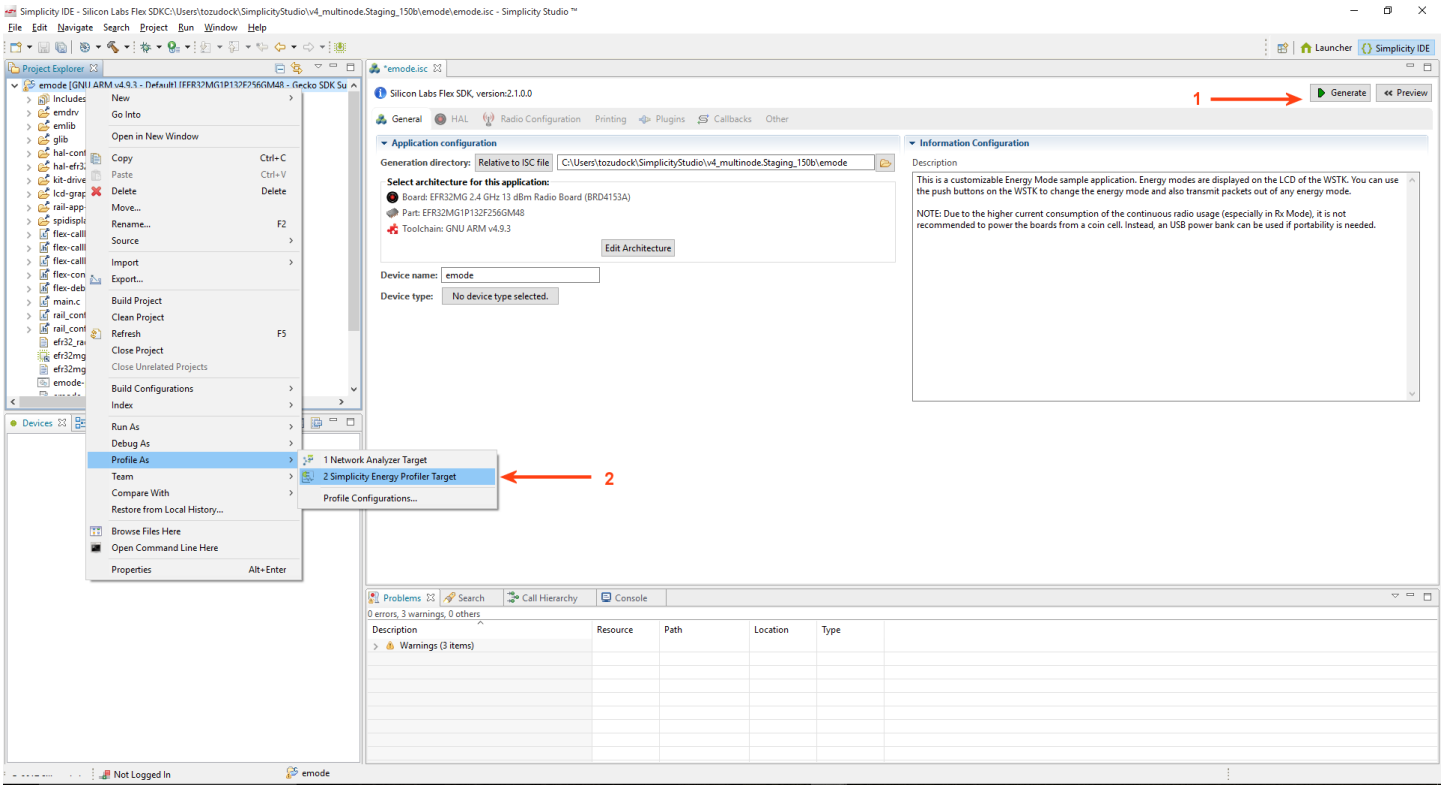
Many Silicon Labs SDKs contain examples that are pre-configured to deliver energy profiling data to Multi-Node Energy Profiler. In the Simplicity Studio Launcher perspective, you can find example titles such as "Powertest" or "Energy modes" or "Emode" for many development kits. In the figure below, the selected device is an EFR32 development kit. The Flex SDK has been installed and the Software Examples launcher column has been scrolled down to the "RAIL: Energy Mode" example. Click the example to create a new project based on that example.



Simplicity Studio prompts you to switch to the Simplicity IDE perspective, as shown in the following figure. Click [**OK**].



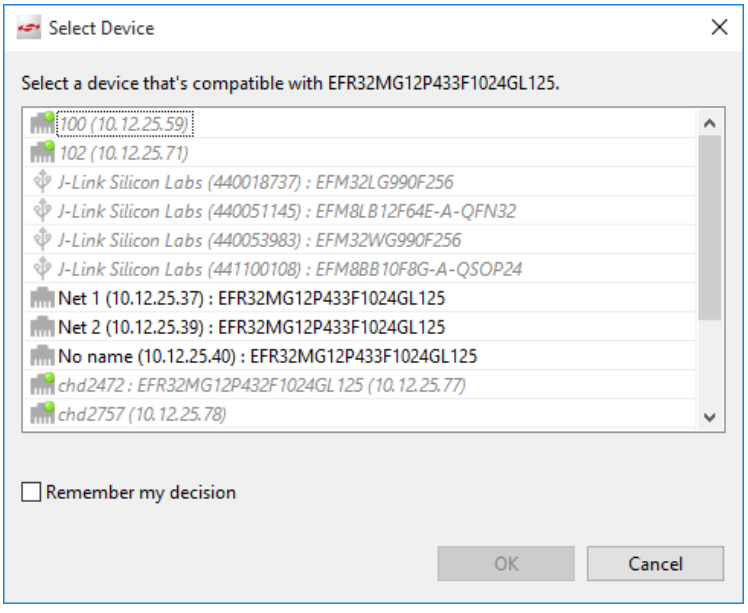
The example opens in the Simplicity IDE perspective. Because this is a wireless example, it opens in AppBuilder. First, click [**Generate**] to generate the project files. Next, open the project-level context menu and select "Profile as Simplicity Energy Profiler Target" as shown in the following figure.



Simplicity Studio then:

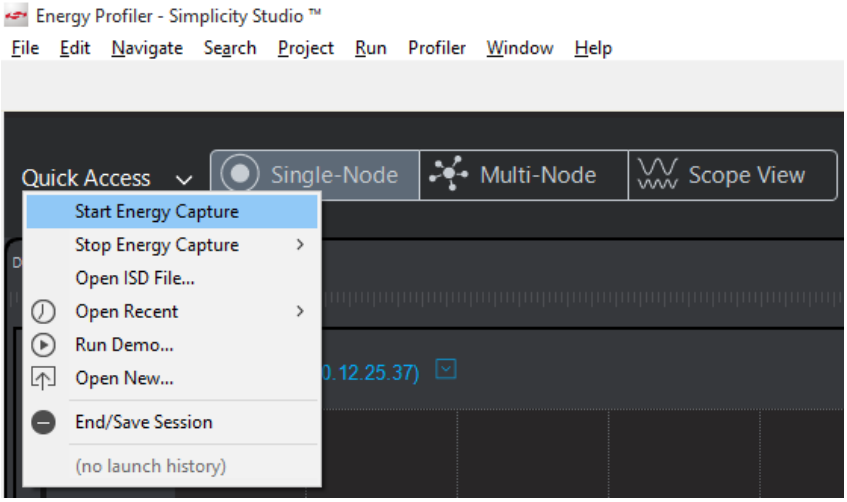
1. Rebuilds the entire project.
2. Flashes the application to compatible hardware.
3. Starts energy capture for that device.
4. Switches to the Multi-Node Energy Profiler perspective.
5. Displays the device data on Single-Node View, Multi-Node View, and Scope View.

In step 2 above, if more than one device is compatible with the application being profiled you will be presented with a list of connected devices. Select the compatible device of interest and click [**OK**].

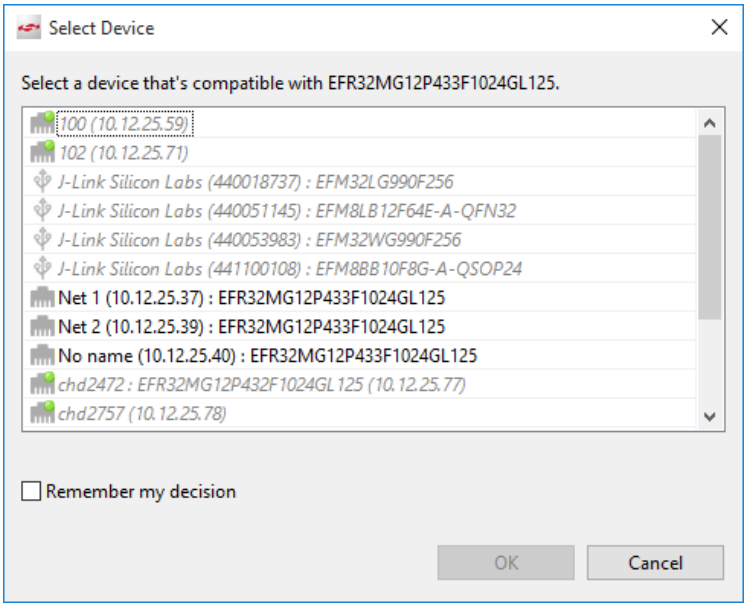


2.2 Starting an Energy Analysis Session from Energy Profiler

If an embedded application is configured to produce energy data and is already running on a device, it is possible to connect to it from the Multi-Node Energy Profiler tool. In the Multi-Node Energy Profiler perspective, click the "Quick Access->Start Energy Capture" menu as shown in the following figure.



If more than one device is compatible with the application being profiled you are presented with a list of connected devices. Select the compatible device of interest and click [**OK**].



The Multi-Node Energy Profiler tool:

- Starts energy capture for that device.
- Depending upon which view is currently active, displays the device's energy data on Single-Node view, Multi-Node view, or Scope view.

2.3 Customer Hardware and Software Design Information

2.3.1 Hardware Design

2.3.1.1 EFM8 Hardware Interface

To use the Multi-Node Energy Profiler functionality on a board design, the board needs to include a debug interface that can be connected to a Silicon Labs STK or WSTK. For the basic Multi-Node Energy Profiler current measurements the board must be powered from the VAEM supply of a Silicon Labs STK or WSTK. Both the current measurement and EFM8 C2 Debug interface can be obtained by using the Silicon Labs Mini Simplicity 10 pin connector. This connector is detailed in *AN958: Debugging and Programming Interfaces for Custom Designs* (an958-mcu-stk-wstk-guide.pdf). The EFM8 C2 interface signals are on SWDIO (Pin 7 C2D) and SWCLK (Pin 8 C2CK) pins shown in the following figure. Code correlation is not possible with the EFM8 parts as they do not include the SWO pin that is used to transmit the program counter information. As mentioned in AN958, using the Silicon Labs debug adapter board (BRD8010A) is the easiest way to get the Mini Simplicity pinout from a Silicon Labs STK or WSTK development kit.

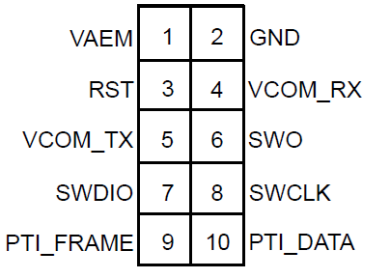


Figure 2.1. Mini Simplicity Connector Pinout

2.3.1.2 EFM32 and EFR32 Hardware Interface

To use the Multi-Node Energy Profiler P functionality on a board design, the board needs to include a debug interface that can be connected to a Silicon Labs STK or WSTK. For the basic Multi-Node Energy Profiler current measurements the board must be powered from the VAEM supply of a Silicon Labs STK or WSTK. To also include code correlation the debug interface must include the SWD interface. Both the current measurement and the code correlation (SWD) can be obtained by using the Silicon Labs Mini Simplicity 10 pin connector. This connector is detailed in *AN958: Debugging and Programming Interfaces for Custom Designs* (an958-mcu-stk-wstk-guide.pdf). The Mini Simplicity 10 pin connector can be used with all EFM32 and EFR32 parts. The pinout is shown in Figure 2.1 Mini Simplicity Connector Pinout on page 7. As mentioned in AN958, using the Silicon Labs debug adapter board (BRD8010A) is the easiest way to get the Mini Simplicity pinout from a Silicon Labs STK or WSTK development kit. The debug adapter board is not compatible with the older EFM32 Development Kits (DKs) and some of the older EFM32 starter kits (STK) that have a different debug connector on them (Gecko, Giant Gecko (EFM32GG-STK3700), Leopard Gecko, Tiny Gecko, Wonder Gecko, Zero Gecko).

2.3.2 Software Configuration

To use the current monitoring functionality provided by the AEM interface, no software changes or setup are required. To use the code correlation functionality the SWD interface must be configured to output periodic program counter information.

2.3.2.1 EFM8 Software Configuration

Currently code correlation is not possible with the EFM8 family, but the Multi-Node Energy Profiler can still be used to monitor the overall energy use of the parts over time / usage scenarios.

2.3.2.2 EFM32 Software Configuration

1. Add bsp_trace.c to the project. (from [SimplicityStudioSDKDirectory]/hardware/kit/common/bsp)
2. The macros BSP_TRACE_SWO_PIN, BSP_TRACE_SWO_PORT and BSP_TRACE_SWO_LOC must be defined.

   For EFM32 parts this is done by default for Silicon Labs development kits in hardware\kit\KIT_NAME\config\hal-config-board.h. For a custom board design the Silicon Labs hal-config-board.h can be used as an example for defining the values. The following is an example for the Pearl Gecko Starter Kit (SLSTK3401A):

```
// $[GPIO]
#define PORTIO_GPIO_SWV_PIN                       (2)
#define PORTIO_GPIO_SWV_PORT                      (gpioPortF)
#define PORTIO_GPIO_SWV_LOC                       (0)

#define BSP_TRACE_SWO_PIN                         (2)
#define BSP_TRACE_SWO_PORT                        (gpioPortF)
#define BSP_TRACE_SWO_LOC                         (0)
// [GPIO]$
```

3. The code must enable SWO output from the MCU. To enable this output, add '#include bsp_trace.h' to main.c and call `BSP_TraceSwoSetup()` during initialization after the `EMU_DCDCInit()` call in main().

```
#include <stdint.h>
#include "bsp_trace.h"
.
.
.
/* Initialize DCDC. Always start in low-noise mode. */
dcdcInit.dcdcMode = emuDcdcMode_LowNoise;
EMU_DCDCInit(&dcdcInit);

// Setup SWD for code correlation
BSP_TraceSwoSetup();
```

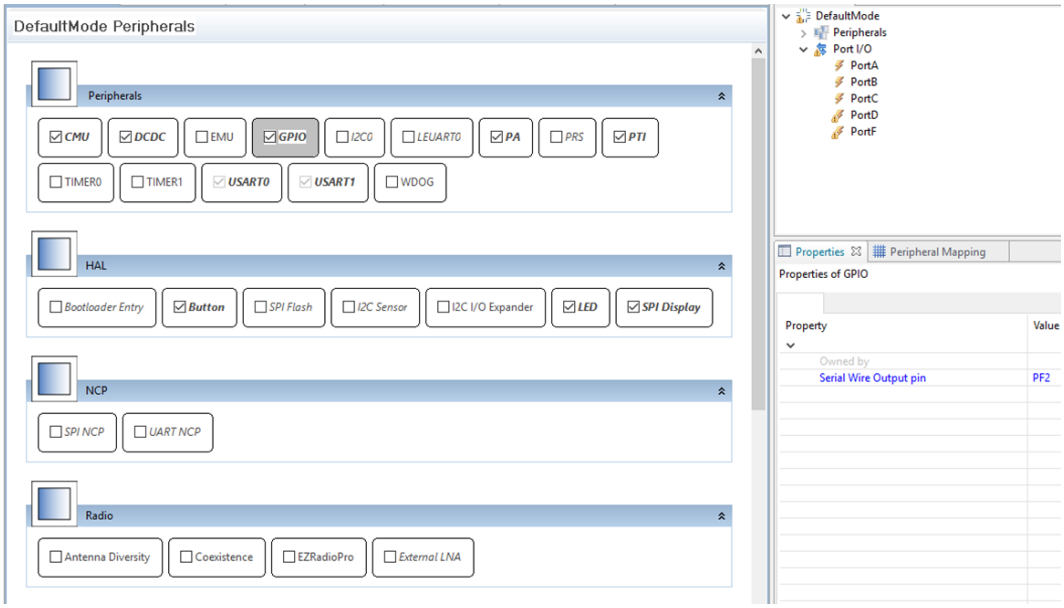
4. The program must be built with debug information enabled so that source code lookup is possible. If you create your project in Simplicity Studio, this is enabled by default. If you import a project into Studio, check compiler options in project context menu "Properties...-> C/C++ Build -> Settings". If you build your program outside of Simplicity Studio, check compiler options of the build tools.

2.3.2.3 EFR32 Software Configuration

1. Add bsp_trace.c to the project. (from [SimplicityStudioSDKDirectory]/hardware/kit/common/bsp)
2. The macros BSP_TRACE_SWO_PIN, BSP_TRACE_SWO_PORT and BSP_TRACE_SWO_LOC must be defined.
   a. With the BLE SDK this is done by default for Silicon Labs development kits in hardware\kit\KIT_NAME\config\hal-config-board.h. For a custom board design the Silicon Labs hal-config-board.h can be used as an example for defining the values. Below is an example of these macros for a SiLabs BGM121 radio board:

```
// $[GPIO]
#define PORTIO_GPIO_SWV_PIN                         (2)
#define PORTIO_GPIO_SWV_PORT                        (gpioPortF)
#define PORTIO_GPIO_SWV_LOC                         (0)

#define BSP_TRACE_SWO_PIN                           (2)
#define BSP_TRACE_SWO_PORT                          (gpioPortF)
#define BSP_TRACE_SWO_LOC                           (0)
// [GPIO]$
```

   b. For Flex and EmberZNet SDKs, in the Hardware Configurator enable the GPIO module. This adds the BSP_TRACE_SWO_* macros to the hal-config.h file.



3. The code must enable SWO output from the MCU. To enable this output, add '#include bsp_trace.h' to the appropriate module and call `BSP_TraceSwoSetup()` during initialization as indicated in the following sections, based on the product family and SDK being used:
   a. EFR32 - BLE SDK: Place `BSP_TraceSwoSetup()` in main.c after the `initApp()` call.

```
#include "bsp.h"
#include "bsp_trace.h"
.
.
.
// Initialize application
initApp();

// Setup SWD for code correlation
BSP_TraceSwoSetup();

// Initialize LEDs
BSP_LedsInit();
```

   b. EFR32 – Flex SDK: Place `BSP_TraceSwoSetup()` in main.c after the `BSP_Init()` call.

```
#include "hal_common.h"
#include "bsp_trace.h"
```

```
.
.
.
// Initialize the BSP
BSP_Init(BSP_INIT_BCC);

// Setup SWD for code correlation
BSP_TraceSwoSetup();
```

c. EFR32 – EmberZNet SDK: Place `BSP_TraceSwoSetup()` in af-main-soc.c after the `emberInit()` call.

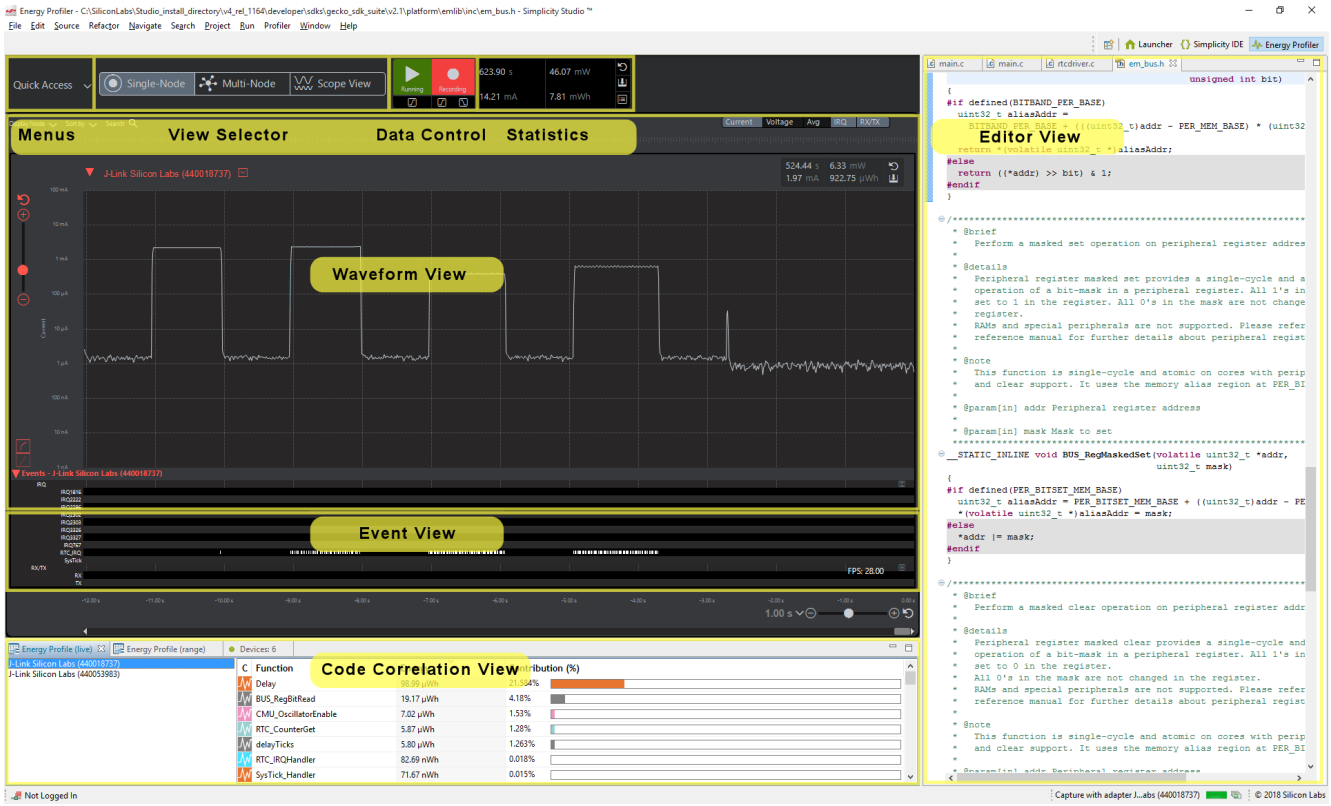
```
#include "afv2-bookkeeping.h"

#if defined(CORTEXM3_EFR32_MICRO) || defined(CORTEXM3_EMBER_MICRO)
  #define EXTENDED_RESET_INFO
  #include "hal/micro/cortexm3/diagnostic.h"
#endif
#include "bsp_trace.h"
.
.
.
int emberAfMain(MAIN_FUNCTION_PARAMETERS)
{
  EmberStatus status;

  {
    int returnCode;
    if (emberAfMainStartCallback(&returnCode, APP_FRAMEWORK_MAIN_ARGUMENTS)) {
      return returnCode;
    }
  }

  // Initialize the Ember Stack.
  status = emberInit();

  if (status != EMBER_SUCCESS) {
    emberAfCorePrintln("%pemberInit 0x%x", "ERROR: ", status);

    // The app can choose what to do here.  If the app is running
    // another device then it could stay running and report the
    // error visually for example. This app asserts.
    assert(false);
  } else {
    emberAfDebugPrintln("init pass");
  }
  // Setup SWD for code correlation
  BSP_TraceSwoSetup();
```

4. The program must be built with debug information enabled so that source code lookup is possible. If you create your project in Simplicity Studio, this is enabled by default. If you import a project into Studio, check compiler options in the project context menu "Properties...-> C/C++ Build -> Settings". If you build your program outside of Simplicity Studio, check compiler options of the build tools.

3. Multi-Node Energy Profiler User Interface Overview

The Multi-Node Energy Profiler user interface is divided into six primary sections. The following figure shows Single-Node view.

- Quick Access Menus – The most frequently used menus for Multi-Node Energy Profiler are available on the user interface itself and also from the Profiler menu at the top menu bar.
- View Selector – Three views for Multi-Node Energy Profiler can be selected by clicking the View Selector bar.
- Data Control – Controls in this area are used to enable and disable data flow to the user interface and to disk.
- Statistics – Energy data for the waveform is shown in the statistics area.
- Code Correlation – Function execution related to power consumption is shown in the Code Correlation view.
- Editor – Source code for the program is shown here and works in conjunction with the Code Correlation view.



3.1 Energy Analysis Views

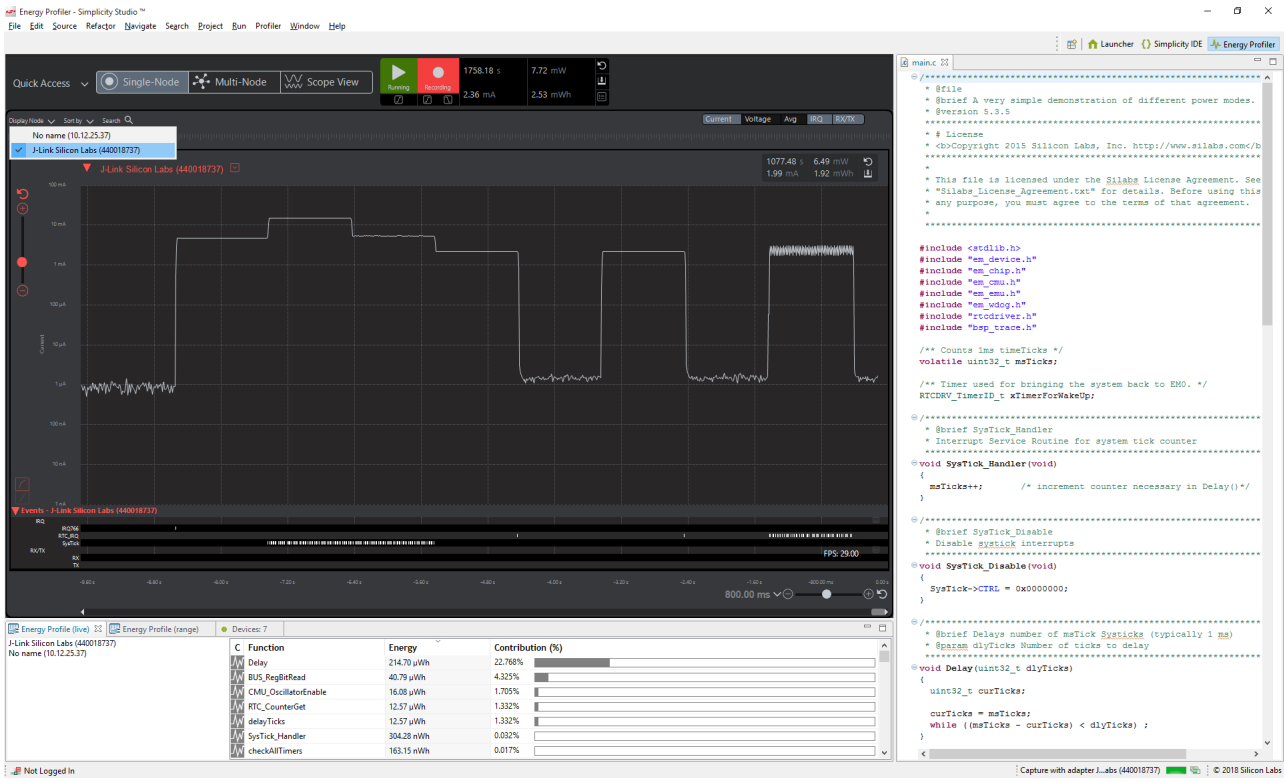
Three views are available for evaluating power performance:

- Single-Node view
- Multi-Node view
- Scope view

Select the desired view by clicking the view name in the View Selector bar of Multi-Node Energy Profiler. In the following figure, Single-Node view has been selected.

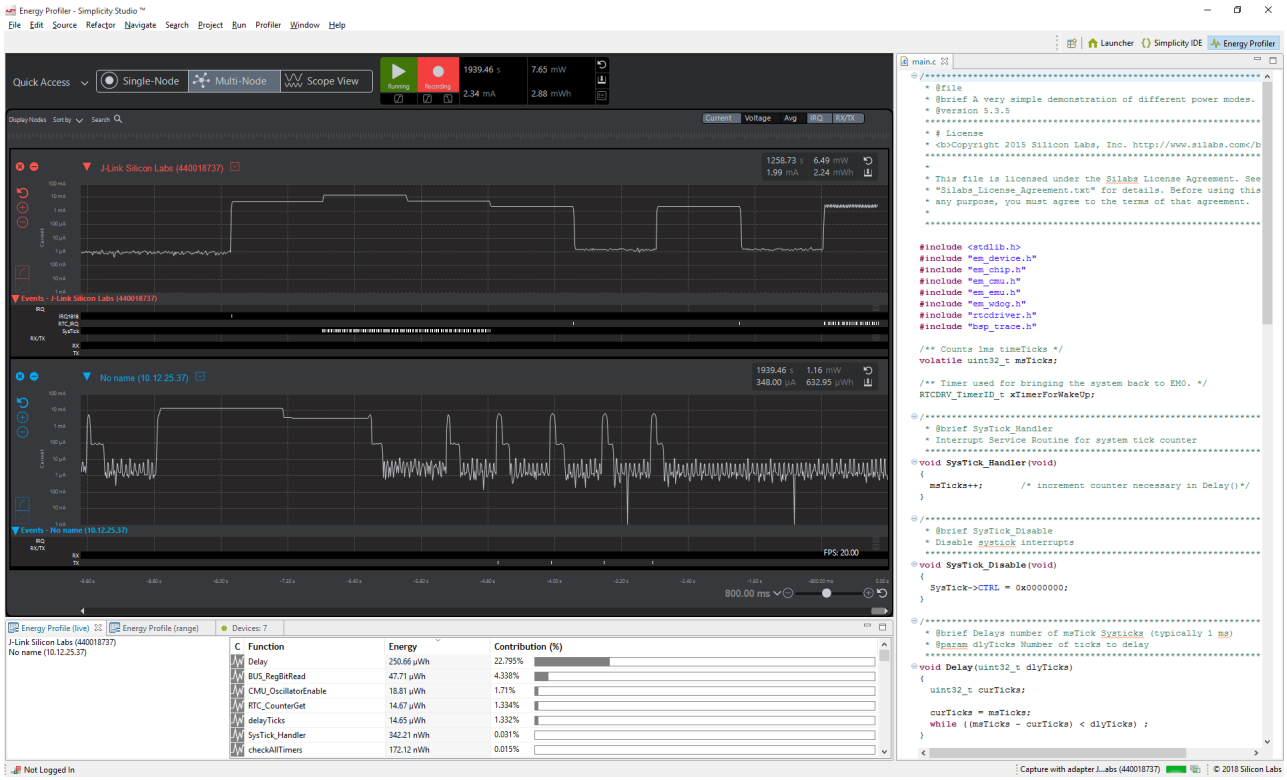
3.1.1 Single-Node View

Single-Node view is used to display a single device's waveform and events. It displays a high detail waveform and ensures that all events rows are visible, without the need to scroll the UI. The Energy Profiler perspective combines this Single-Node view, the Code Correlation view, and the source code Editor view, allowing you to perform in-depth analysis of a single device from many viewpoints. Any active device may be selected through the "Display Nodes" menu selector.



3.1.2 Multi-Node View

Multi-Node view is used to display multiple device waveform and events. It allows you to investigate system behavior and interactions between devices. Two devices are visible in the main portion of the UI, while others are viewed by scrolling. The Multi-Node view groups each individual device's waveform and events together. The Energy Profiler perspective combines this Multi-Node view, the Code Correlation view, and the source code Editor view, allowing you to perform in-depth analysis of multiple devices from many viewpoints.

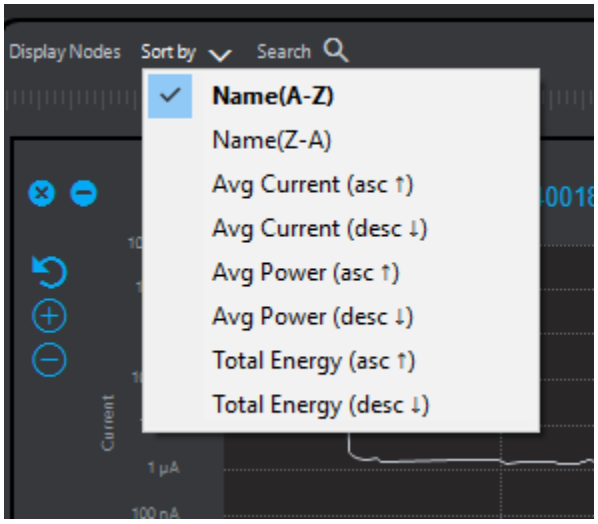


Multi-Node view offers the following capabilities:

- Sorting
- Node display control

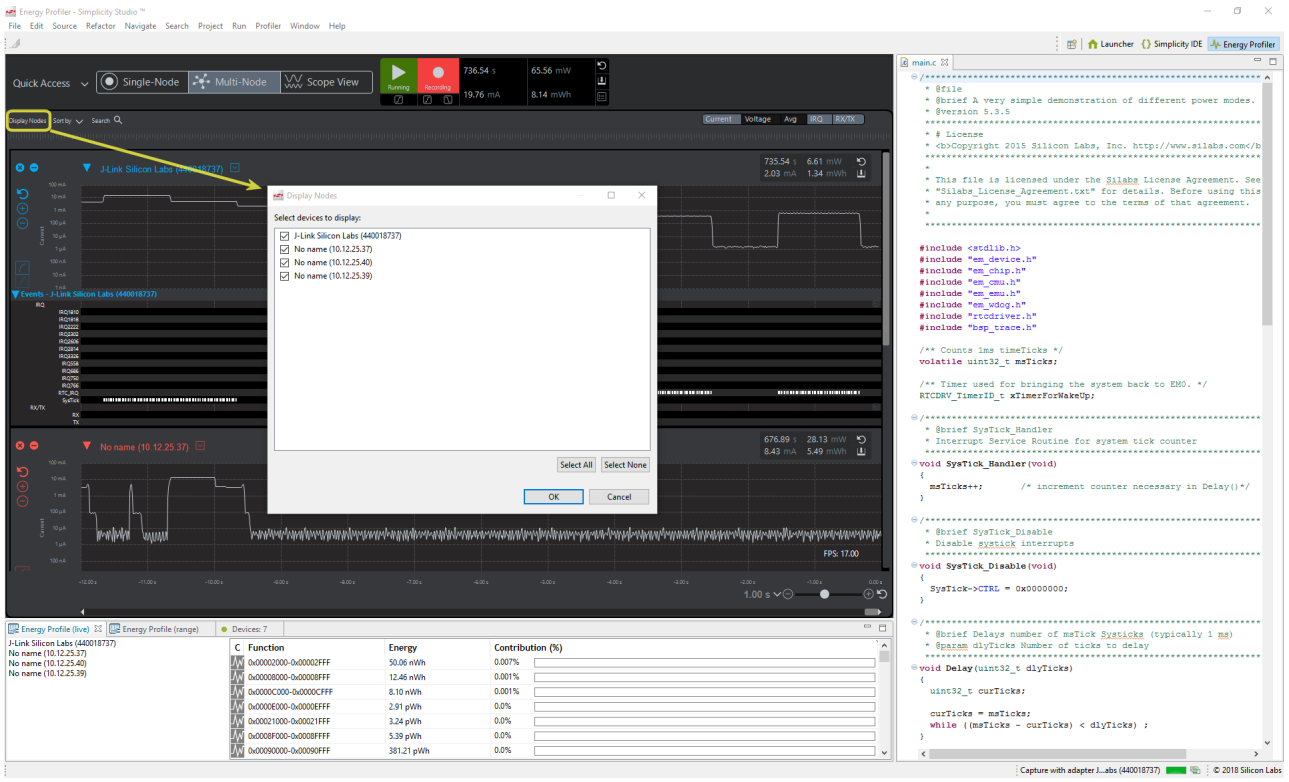
3.1.2.1 Sorting in Multi-Node View

Sorting in Multi-Node view allows you to order waveforms based upon several criteria. For example, it may be useful to sort descending by average power, such that the highest power consumers are easily identified at the top. As shown in the following figure, devices may be sorted ascending and descending by name, average current, average power, or total energy.



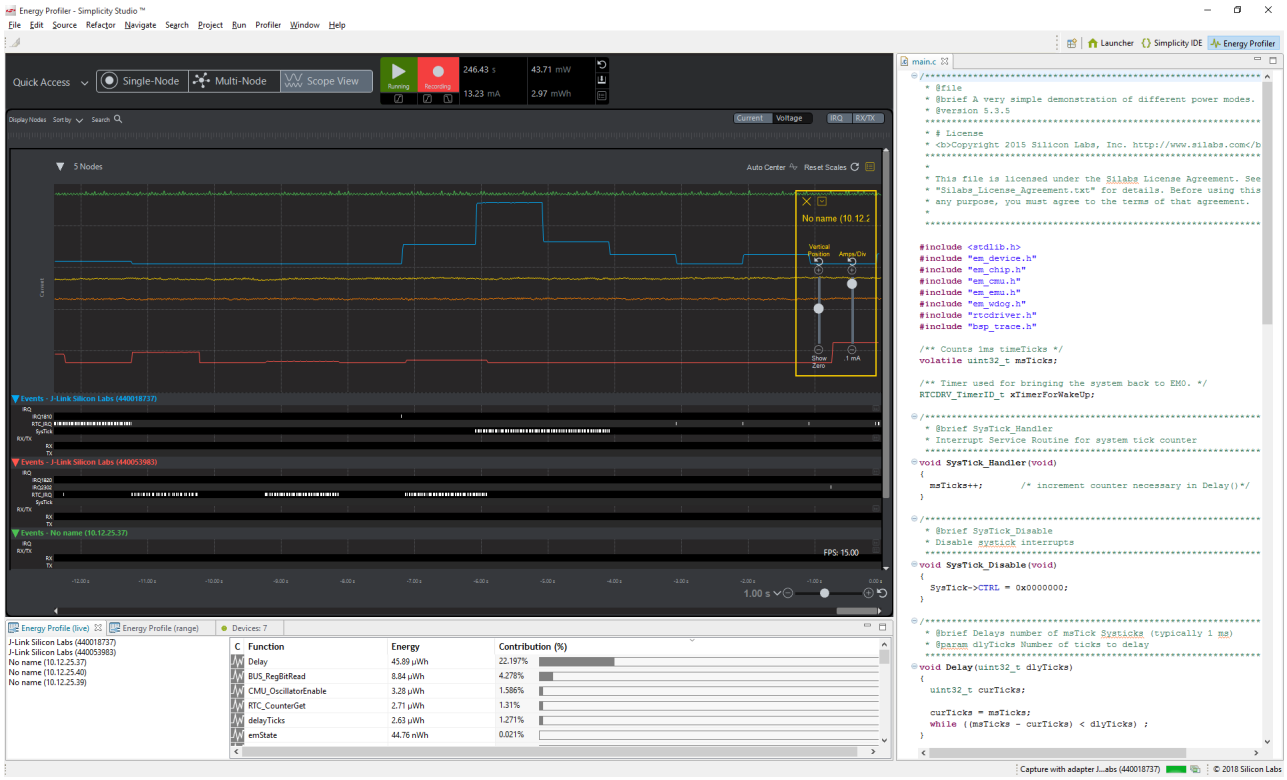
3.1.2.2 Displaying Nodes in Multi-Node View

Enabling capture on any device automatically adds it to the Multi-Node view. However, as the number of devices displayed increases, UI performance may be impacted. Click [**Display Nodes**] to select the devices that will be displayed. This selection only affects the waveform display. If recording is enabled, data for all actively capturing devices is streamed to disk regardless of whether or not they are displayed.



3.1.3 Scope View

Scope view is used to display multiple device waveforms and events, allowing you to investigate system behavior and interactions between devices. In Scope view, all waveforms are displayed in one waveform area, so that you can see all waveforms without needing to scroll the UI. It also enables the user overlay and align traces like an oscilloscope. All event traces are grouped below the waveform view. The Energy Profiler perspective combines this Scope view, the Code Correlation view, and the source code Editor view allowing, you to perform in-depth analysis of a single device from many viewpoints.

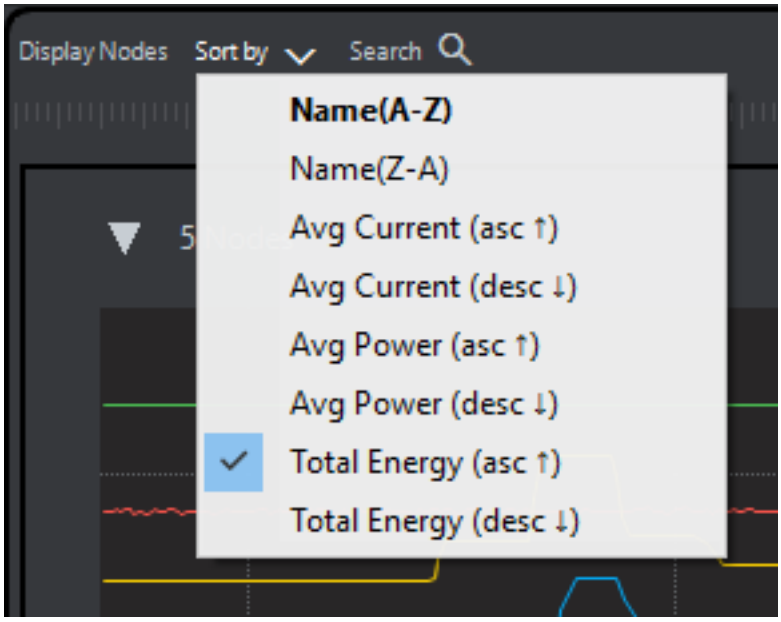


Scope view offers the following capabilities:

- Sorting
- Node display control
- Waveform display control

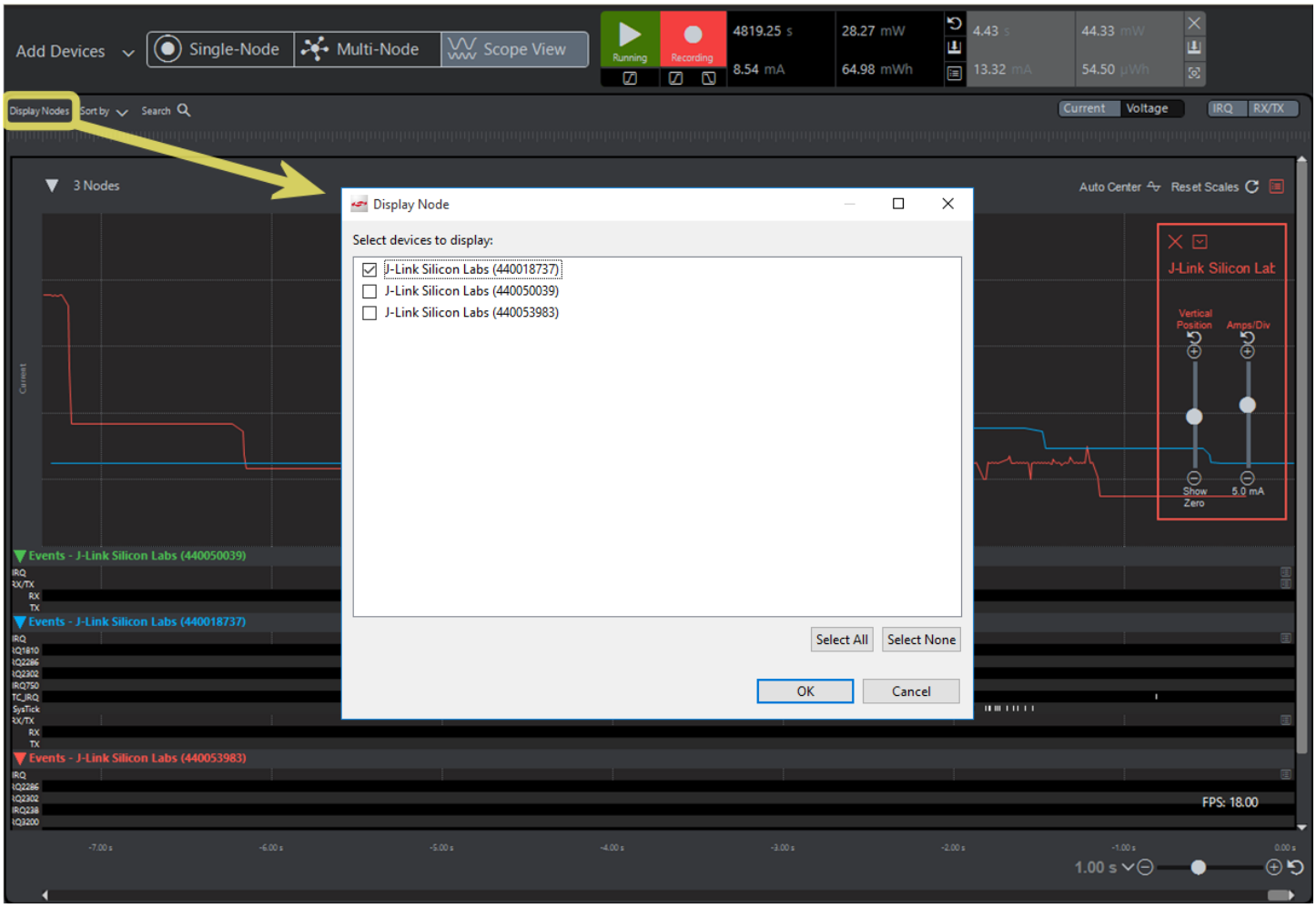
3.1.3.1 Sorting in Scope View

Sorting in Scope view allows you to order events based upon several criteria. For example, it may be useful to sort events descending by average power, such that the events for the highest power consumers are easily identified at the top.



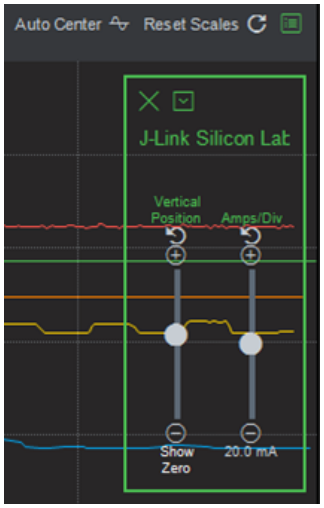
3.1.3.2 Displaying Nodes in Scope View

Enabling capture on any device automatically adds it to scope view. However, as the number of devices displayed increases, UI performance may be impacted. Click [**Display Nodes**] to select which devices will be displayed. This selection only enables or disables display of waveform and event data in the UI. If recording is enabled, data for all actively capturing devices is streamed to disk regardless of whether they are displayed.

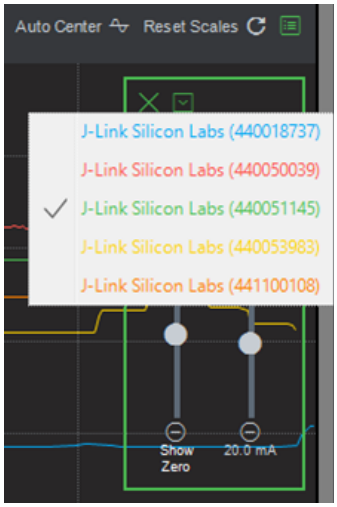


3.1.3.3 Adjust Waveform Size in Scope View

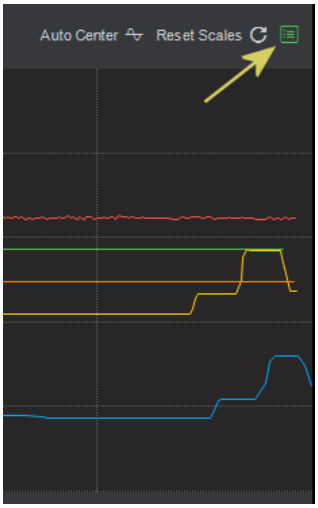
One of the Scope view's key benefits is the ability to view waveforms in a single view at difference scales, enabling you to compare waveform characteristics that would not be possible for two waveforms on the same scale. The waveform scale/offset controls are available by default at the upper right corner of the Scope waveform view. The color of the waveform scale/offset control indicates which waveform will be adjusted when the sliders are moved.



The drop-down menu at the top of the control allows you to select which device should be adjusted by the waveform scale and offset control. The debug adapter name colors match the colors in the waveform portion of Scope view.



Wave scale/offset control display may be toggled using the show/hide button above it.



3.2 Selecting Ranges

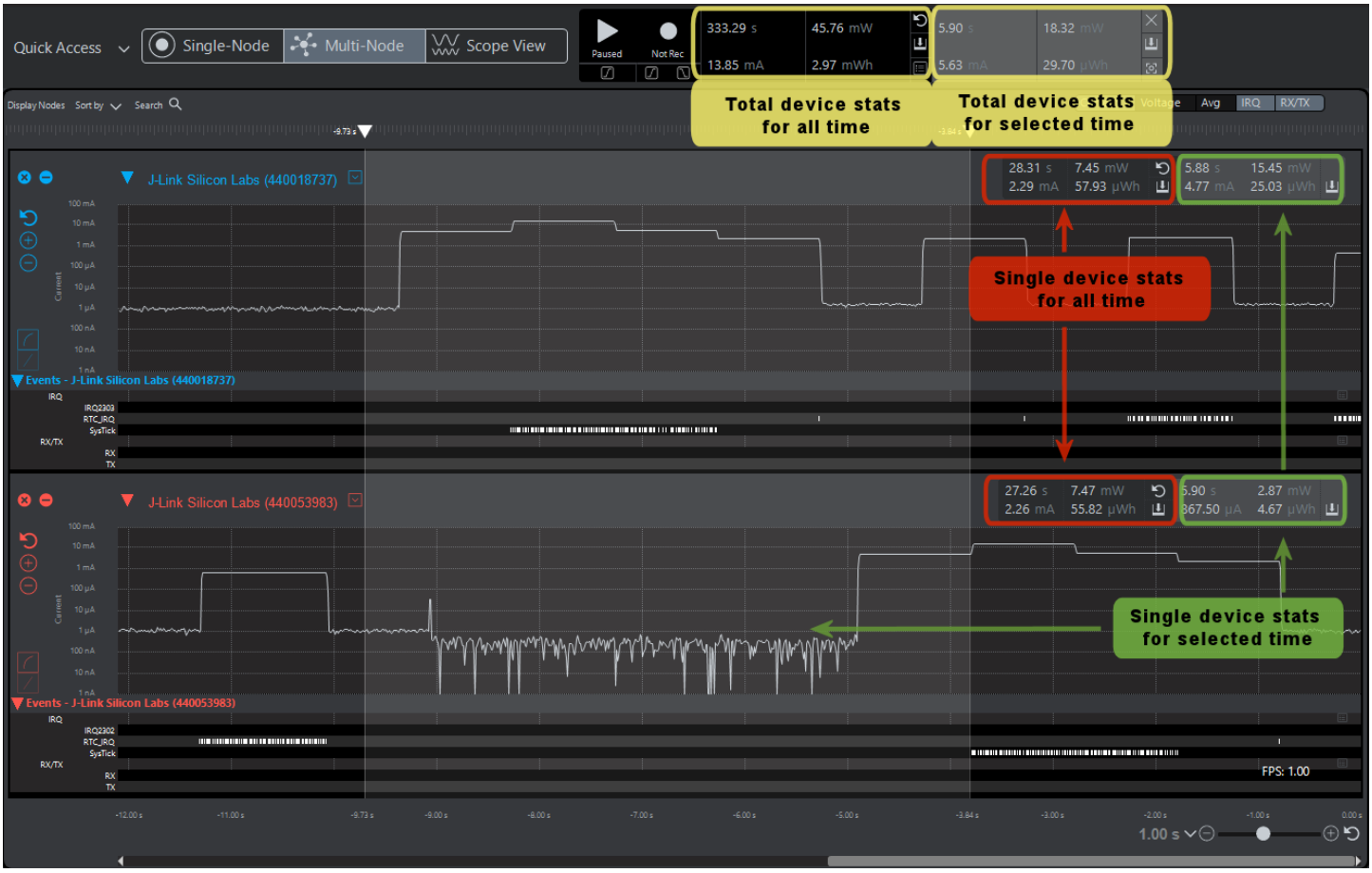
Multi-Node Energy Profiler provides the ability to select a range of data by clicking and dragging across the waveforms or events. This is useful for determining power statistics for regions of interest. Once a range is selected, a light gray section summarizing the energy statistics for the selected region is displayed. That section also allows you to re-center the selection in the waveform view if it has scrolled out of view and includes a button to close/deselect the region.

4. Energy Statistics

Multi-Node Energy Profiler calculates the following device and system statistics.

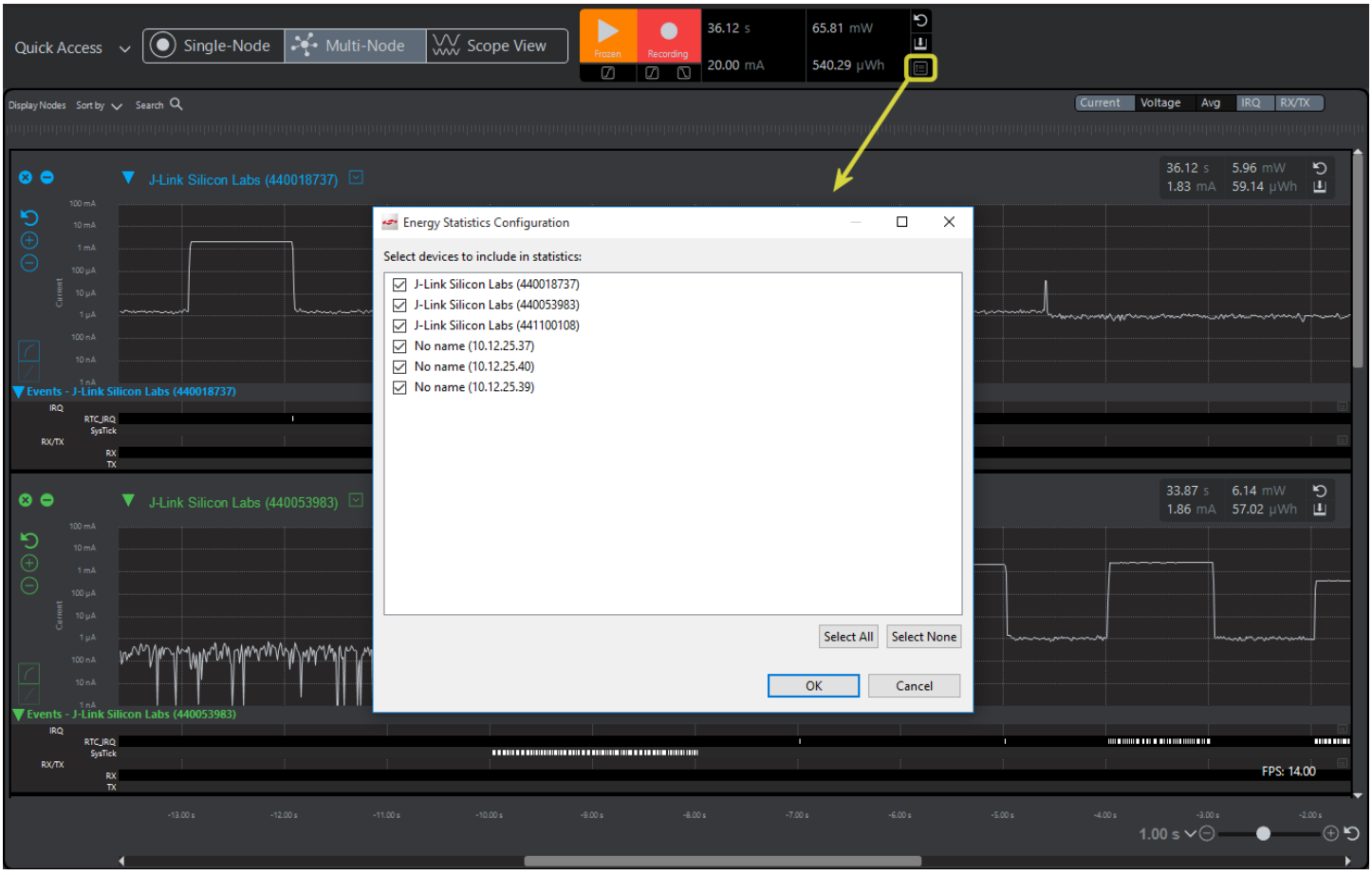
- Average current
- Average power
- Total energy

Statistics are provided for each individual device in the waveform's top bar on both Single-Node and Multi-Node views. The sum for all devices is provided in the top bar of Multi-Node Energy Profiler for all three views. Single-Node and Multi-Node views provide individual device statistics, while Scope view does not.



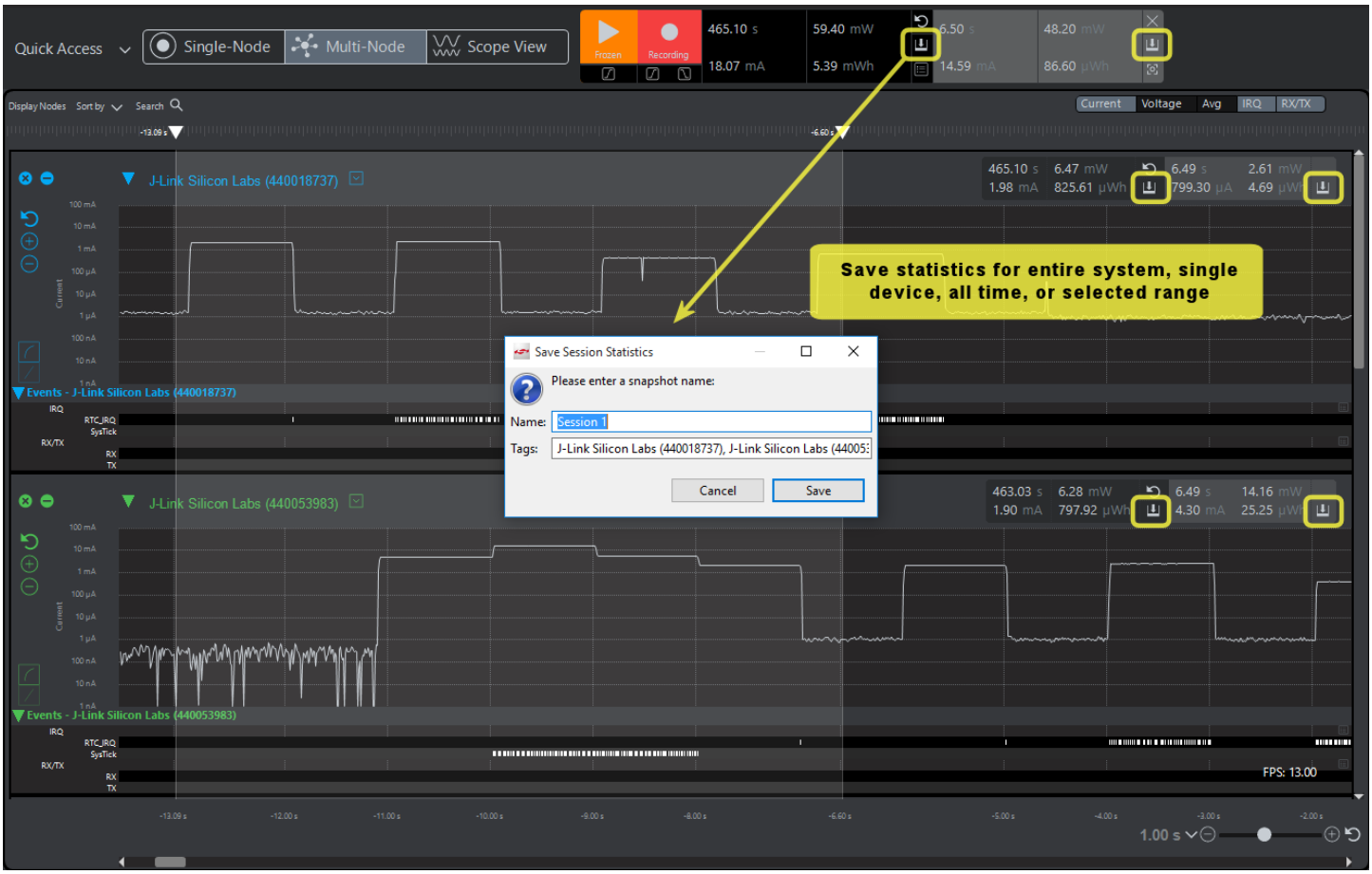
4.1 Energy Statistics Configuration

Multi-Node Energy Profiler allows you to select which of the actively capturing devices are included in the total system energy statistics calculation displayed in the top bar. Click the **Energy Statistics Configuration** icon and select the devices to be included in the calculation. It is important to note that the devices available for selection are those that are actively capturing data, which may be different from the devices that are selected for display in the UI.



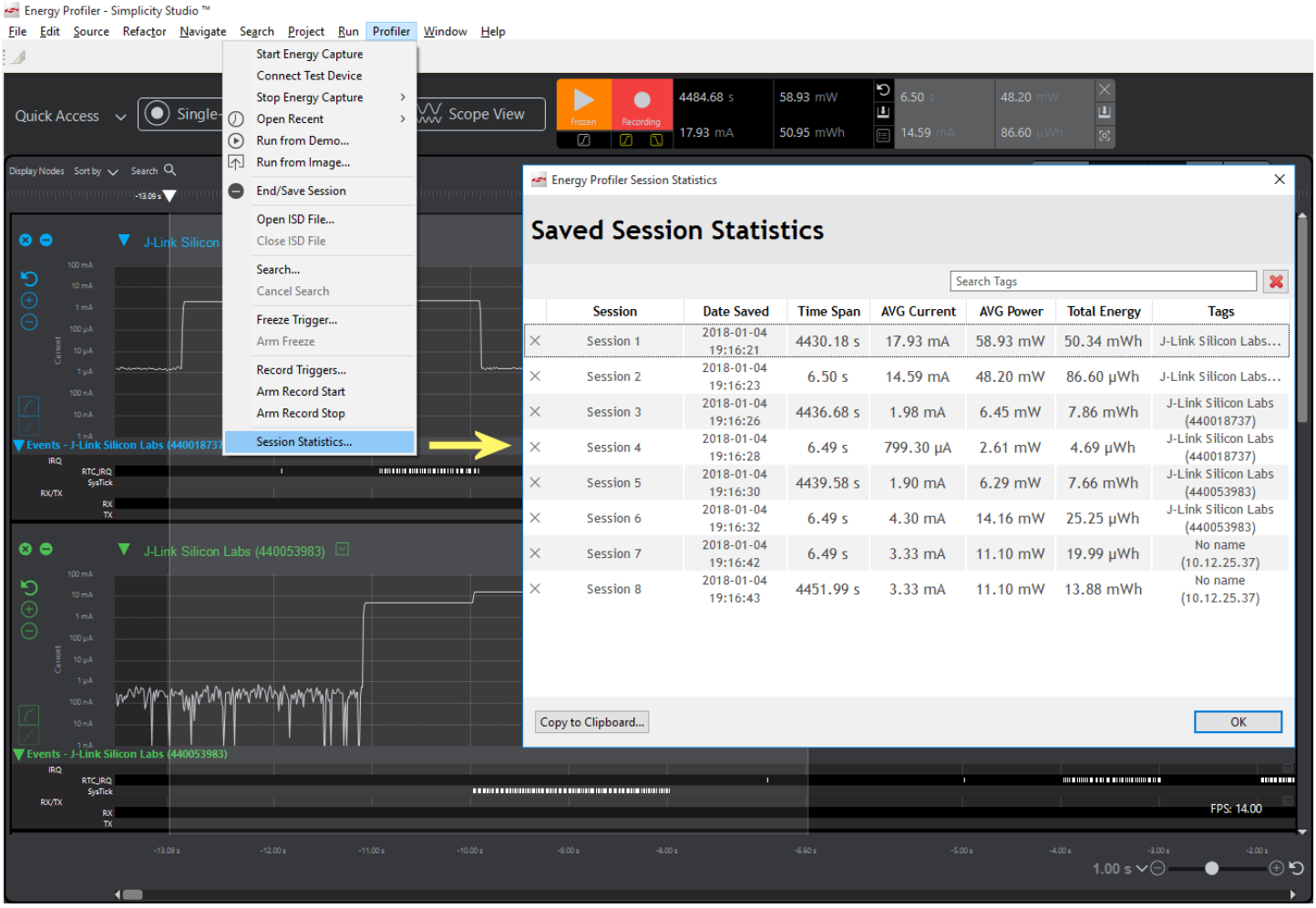
4.2 Save Snapshot of Session Statistics

Multi-Node Energy Profiler allows you to save statistics for future reference and comparison. Click the **Save Snapshot** icon on any of the energy statistics panels and be presented with a dialog to provide a name and tag for future reference.



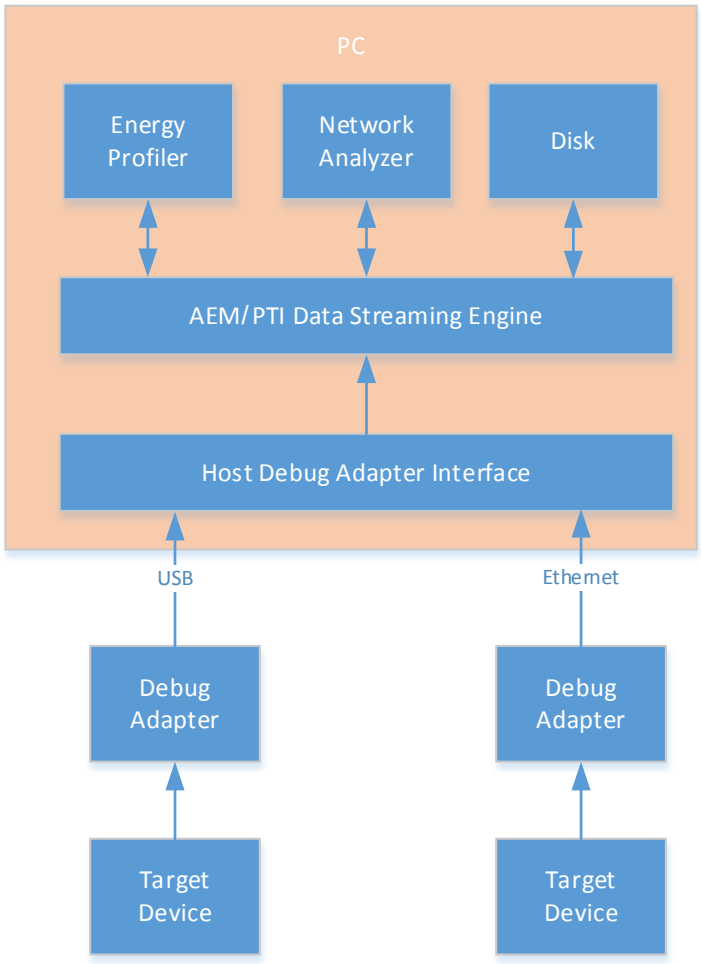
4.3 Review Saved Session Statistics

Once you have saved multiple snapshots, you may want to review them, or to copy them to the clipboard for use in other applications. This is possible through the "Profiler->Session Statistics…" menu.

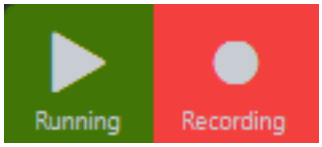


5. Play and Record Data Control

The diagram below provides a general description of data flow from target devices into Simplicity Studio profiling applications. Data is sourced from target devices from the debug adapter either through USB or Ethernet connectivity to the host PC. The host debug adapter software layer provides data to the AEM/PTI data streaming engine, which in turn provides data to analysis applications such as Multi-Node Energy Profiler or Network Analyzer. The AEM/PTI data streaming engine saves data to disk in temporary files during a capture session. When the user ends the session, Multi-Node Energy Profiler prompts the user with the option to save the data to file.



Play and record controls in the top bar of the Multi-Node Energy Profiler provide control over data being captured from connected devices.



5.1 Play Control

The Play control determines whether data being captured is displayed in the Multi-Node Energy Profiler UI. The Play control has three states:

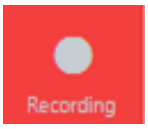
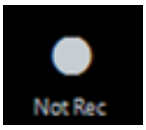
- Running (green)
- Paused (black)
- Frozen (orange)

	The "Running" state is entered when a capture is first started. When the Play control is green and displays "Running" it means energy capture data is actively arriving and being displayed on the UI. The visible portion of the waveform and events is the most recently captured data as it is arriving to Multi-Node Energy Profiler. The time position scroll bar is all the way to the right.
	The "Paused" state is entered by clicking the Play button while it is in the green "Running" state. The Play control turns black and displays "Paused" indicating energy capture data is actively arriving, but it is not being displayed on the UI. The visible portion of the waveform and events may be anywhere on the timeline of previously captured data. Pausing the display has no effect on whether data is saved to disk. Save to disk is determined by the state of the Record control.
	The "Frozen" state means energy capture data is actively arriving for display, but the visible portion of the waveform and events in the UI is previously captured data. The time position scroll bar may be anywhere on the timeline of previously captured data. The "Frozen" state is entered when the UI is in the play state and the time position scroll bar is scrolled or moved to the left. The "Frozen" state may also be entered as the result of a "Freeze trigger" condition. The "Frozen" Play control state has no effect on whether data is saved to disk. Save to disk is determined by the state of the Record control.

5.2 Record Control

The Record control manages data streaming to disk. There are two states to the Record control:

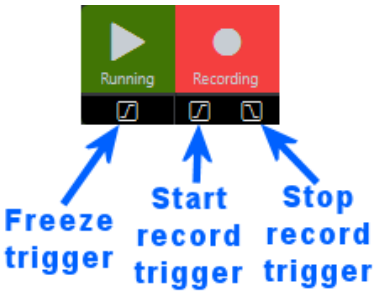
- Recording (red)
- Not recording (black)

	When the Record control is red and displays "Recording" it means energy capture data is actively arriving and is being saved to temporary files to disk. When an energy capture is started, the "Recording" state is entered automatically. The state of the Record control has no effect on the Play control or the data being displayed in the UI. Clicking the Record control while in the "Recording" state transitions to the "Not Rec" state (that is, not recording). The "Recording" state may also be entered by use of Record triggers.
	When the Record control is black and displays "Not Rec" (that is, Not Recording) it means energy capture data is actively arriving but it is not being saved to disk. The "Not Rec" state is entered by clicking the Record control when it is in the "Recording" state. The state of the Record control has no effect on the Play control or the data being displayed in the UI. Clicking the Record control while in the "Not Rec" state transitions to the "Recording" state. The "Not Rec" state may also be entered by use of Record triggers.

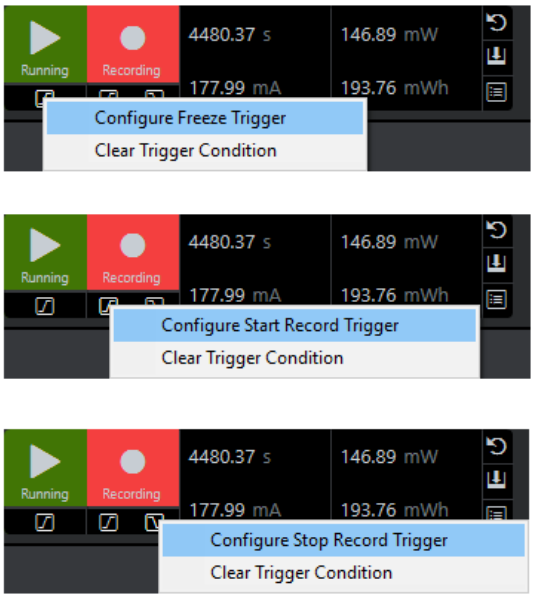
6. Freeze and Record Triggers

Triggers provide a method to automate freezing the display or recording to disk. Trigger icons are found below the Play and Record controls. Three triggers are available.

- Freeze trigger
- Record start trigger
- Record stop trigger



Triggers are driven by criteria set by the user. To set the criteria, use the context menu available from each trigger icon. The context menu also provides the ability to clear all conditions, which also disables the trigger.

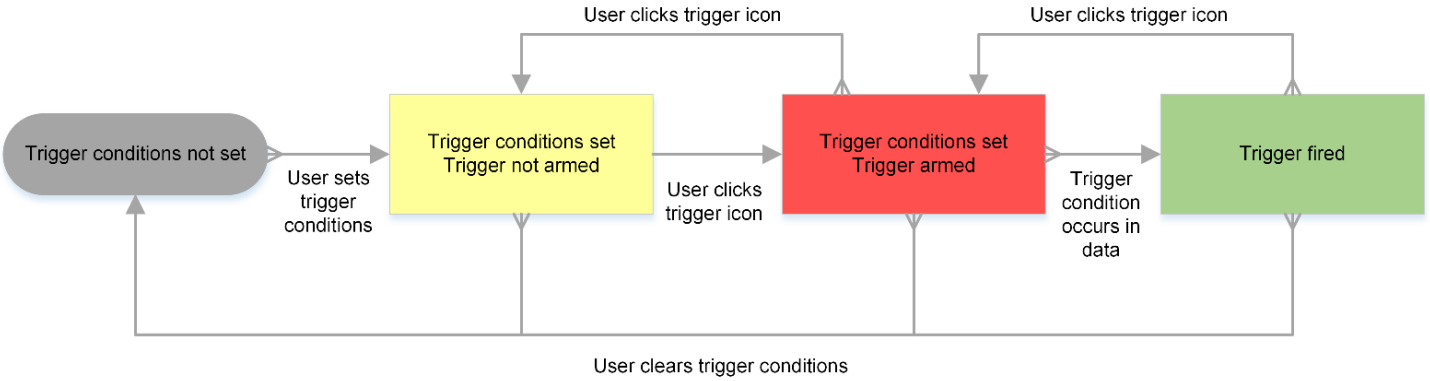


6.1 Trigger States

The four trigger states are each designated by a color:

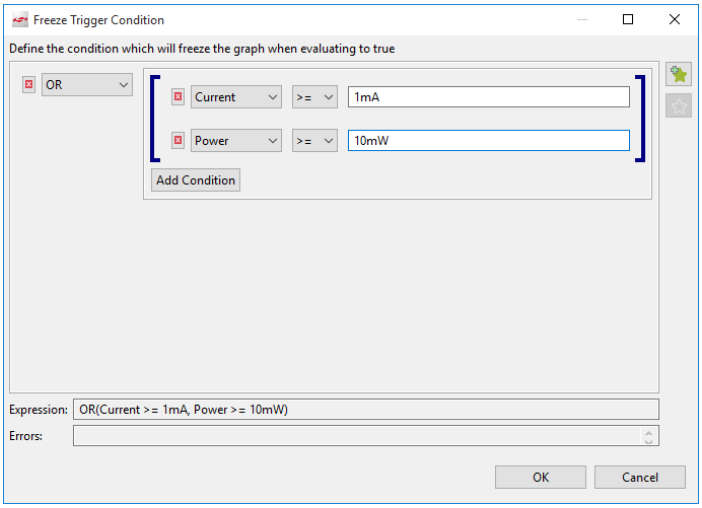
- Grey – Trigger conditions are not set
- Yellow – Trigger conditions are set, trigger is not armed
- Red – Trigger conditions are set, trigger is armed
- Green – Trigger has fired

When Multi-Node Energy Profiler first starts up in a new workspace, all trigger icons are grey indicating trigger conditions have not yet been set. Clicking a grey trigger icon opens the trigger configuration dialog box, allowing you to specify trigger conditions. Once trigger conditions have been set, the trigger icon turns yellow indicating the trigger conditions are set, but the trigger is not armed. Clicking a yellow trigger icon arms the trigger as indicated by the trigger icon turning red. Incoming data is now evaluated against the trigger condition. Once the trigger condition is satisfied, the trigger icon turns green indicating the armed trigger has fired. The relevant action (freeze, record start, or record stop), will have been executed. The following diagram shows the trigger state behavior.



6.2 Freeze Trigger

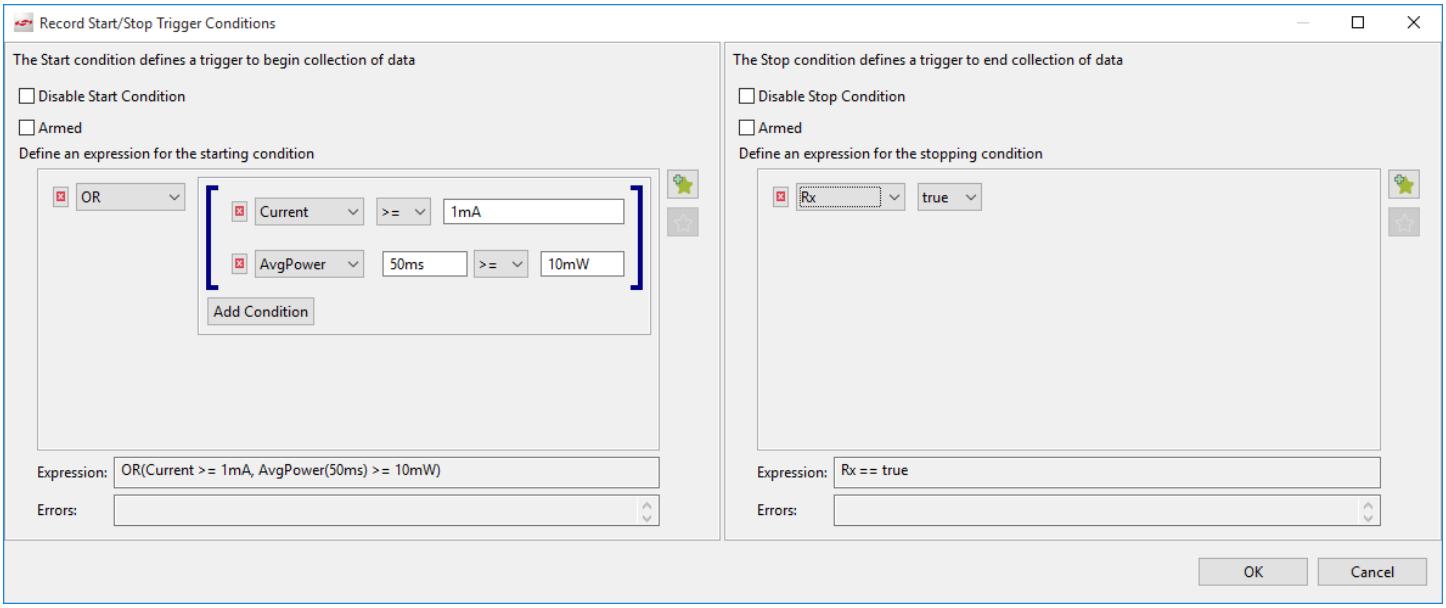
The freeze trigger transitions the user interface from play to freeze mode automatically based upon search criteria. The freeze criteria are set through the Freeze Trigger Conditions dialog, which is invoked through the "Profile->Freeze Trigger…" Simplicity Studio application menu or through the context menu "Configure Freeze Trigger" on the freeze trigger icon in Multi-Node Energy Profiler.



In the following diagram, the play freeze trigger condition has occurred in the incoming data. Freeze mode is automatically entered for the waveform display and data continues to arrive as indicated by the scroll bar automatically moving to the left as new data arrives. Notice that the freeze trigger icon is green, indicating the trigger condition has occurred in arriving data.

6.3 Record Start/Stop Triggers

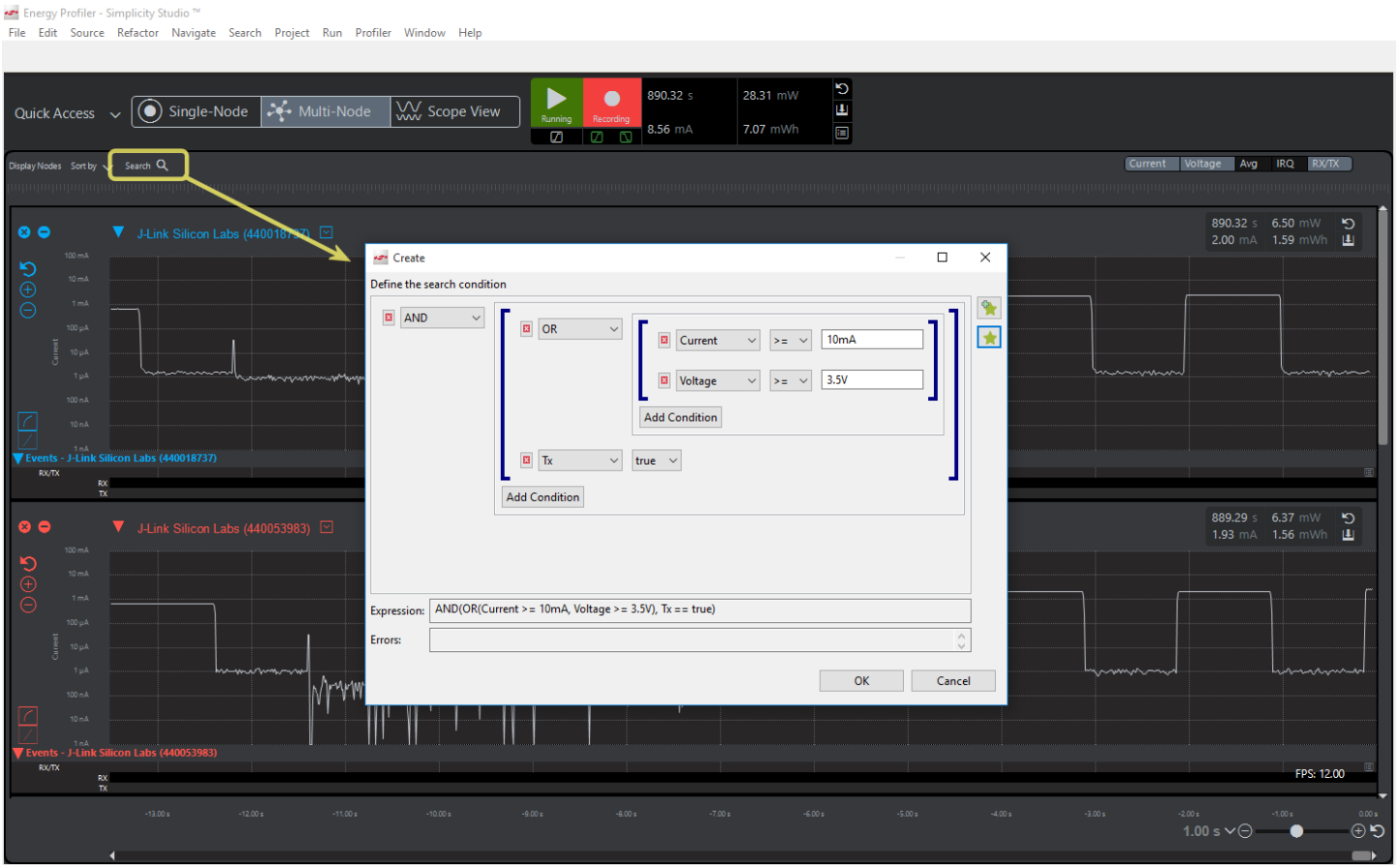
The record start/stop triggers automatically start and stop data record to disk based upon search criteria. This may be beneficial for capturing rare conditions rather than recording large amounts of data followed by search. The record start criteria are set through the Record Start/Stop Trigger Conditions dialog, which is invoked through the "Profile->Record Triggers…" menu or through the context menus "Configure Start Record Trigger" and "Configure Stop Record Trigger" on the trigger icons below the Record control in Multi-Node Energy Profiler. The record start/stop trigger conditions are set together in a single dialog as shown in the following figure.



In the following figure, the record start trigger condition has occurred in the incoming data as indicated by the green record start trigger icon. The record stop trigger condition has not yet occurred and data will be written to disk until it does.

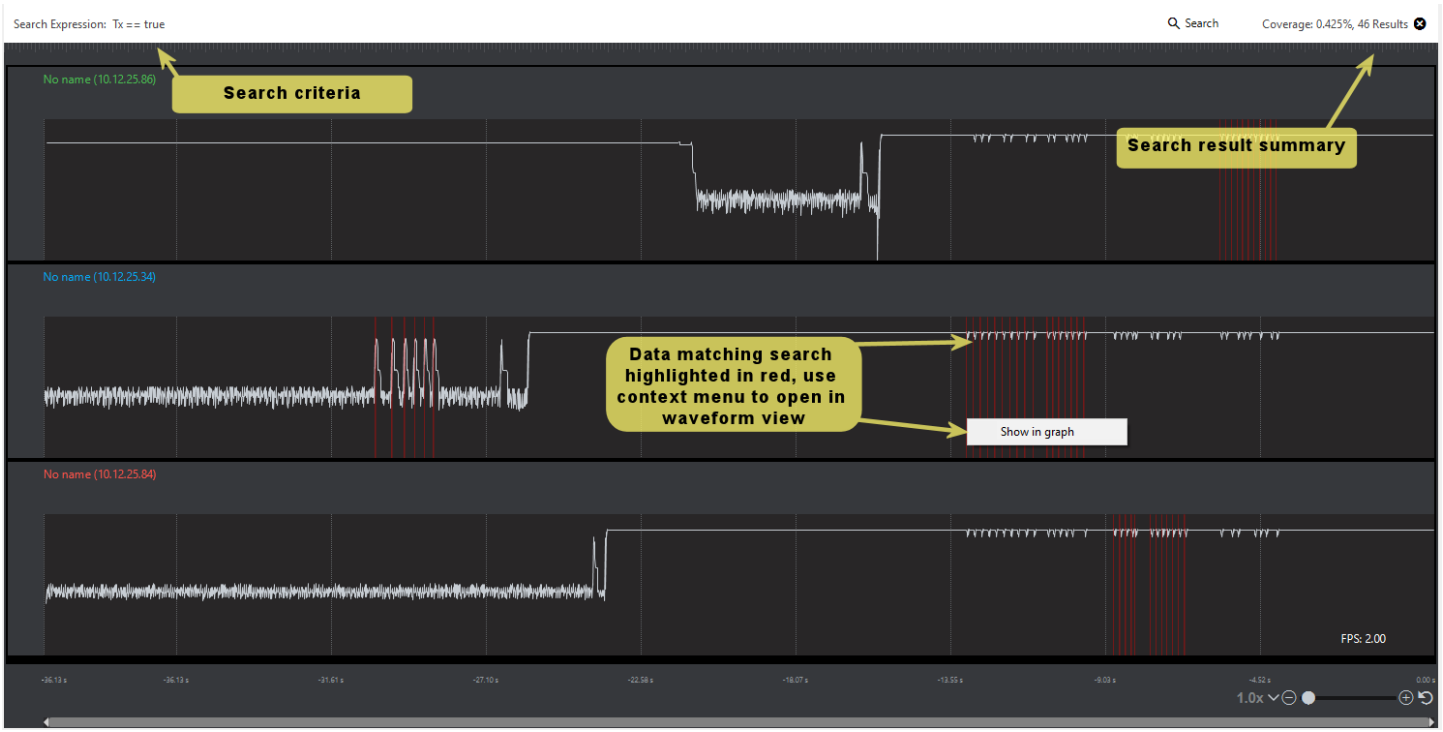
7. Search Capability

One of the most powerful aspects of Multi-Node Energy Profiler is the ability to search data. Search is available both during a live capture and with an offline ISD file. Click [**Search**] to open the Search Conditions dialog.



When you click [**OK**], a search is performed on all data currently available, whether from the currently active capture or in offline mode from an open file. The search results window is displayed upon completion.

Sections where the data matched the criteria are shown in red. A summary of the count and % coverage of the matching regions is shown at the upper right. Use the context menu to jump to the matching area in the waveform view from the search results window.



8. Profiling with Code Correlation

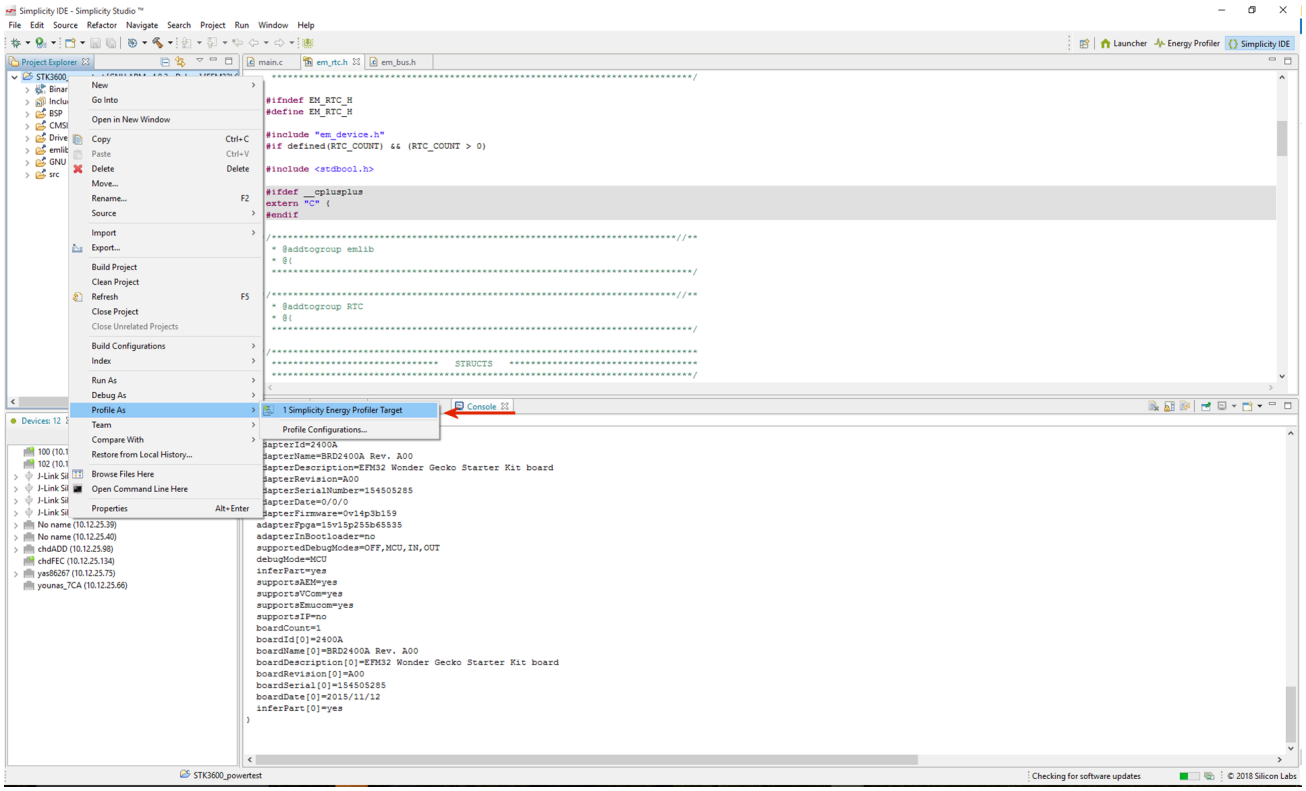
Code correlation is one of the most powerful features in Multi-Node Energy Profiler. Multi-Node Energy Profiler captures program execution in conjunction with the energy data. This allows Multi-Node Energy Profiler to calculate the power consumed by each function executed in the application. This data may then be sorted to highlight the portions of an application that consume the most power. This enables the application developer to know where they should concentrate their software development efforts to reduce power consumption. Additionally, Multi-Node Energy Profiler can color-code sections of the current waveform to represent which functions were executing when a given section of the current waveform occurred.

8.1 Enabling Code Correlation Profiling

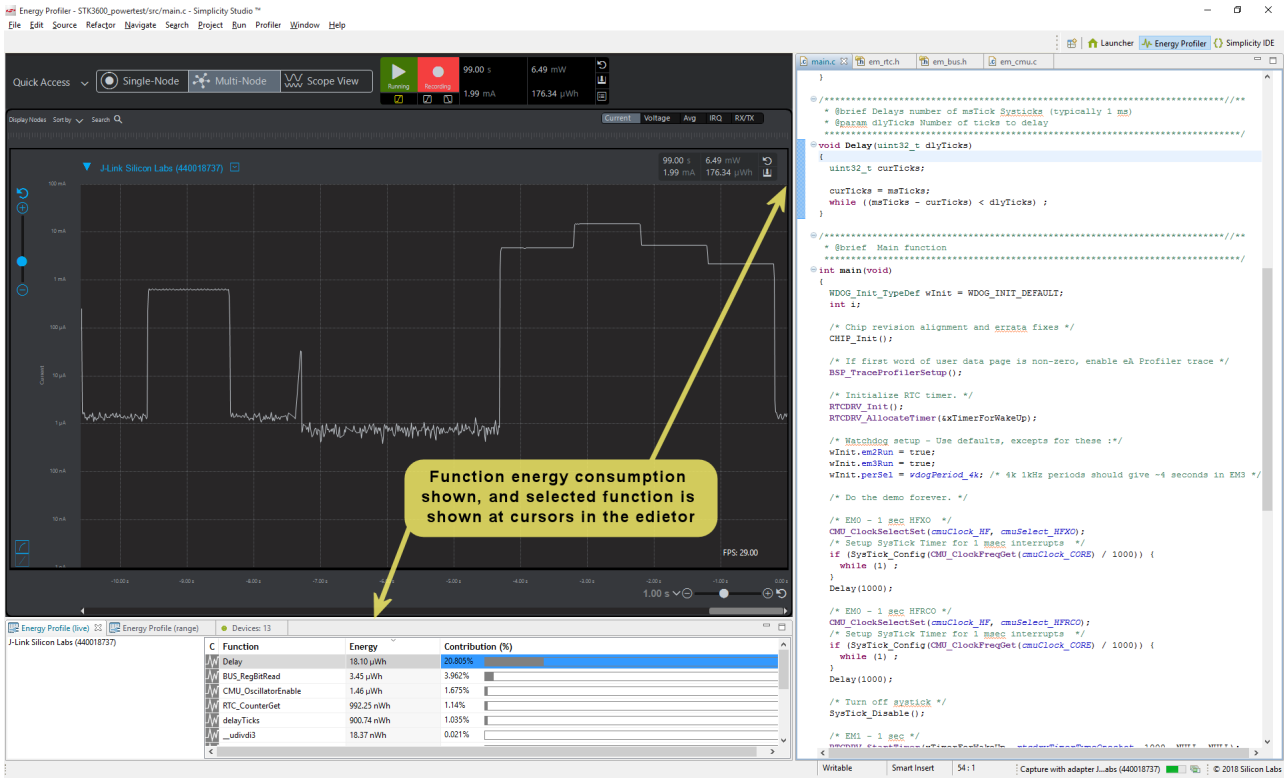
This section assumes an embedded application is already configured to produce AEM data through a debug adapter. For code correlation to operate, Multi-Node Energy Profiler needs access to debug information associated with the embedded application that is currently running. This information is available in the AXF file output from the build. Code correlation can be enabled when an application is started from the Simplicity IDE through context menu commands for a given project, or it can be associated with an application already running on an embedded device that is being profiled in Multi-Node Energy Profiler.

8.1.1 Code Correlation Enabled from the Simplicity IDE

One simple way to enable the code correlation view is to start the energy profiling session by using the context menu on the project from the Simplicity IDE as shown below. The associated debug file will be provided to Multi-Node Energy Profiler when the session is started.

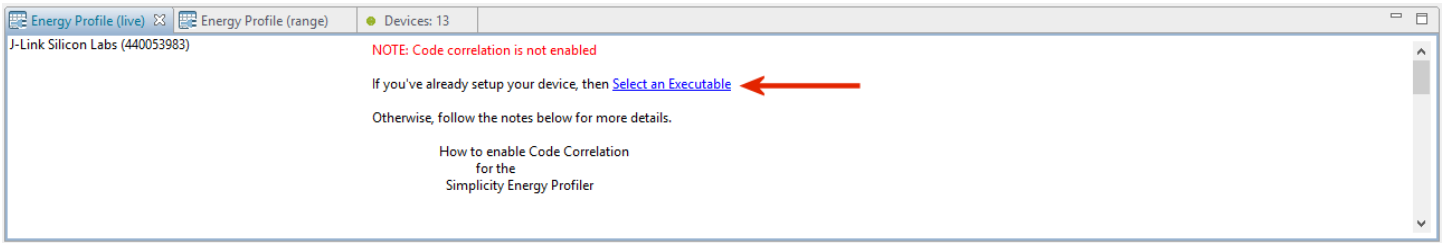


Once Multi-Node Energy Profiler starts, the following view is displayed. Notice in the Code Correlation view, energy consumed by each function is updated in real time. Clicking any of the functions will open the associated source code file and jump to that function in the file.

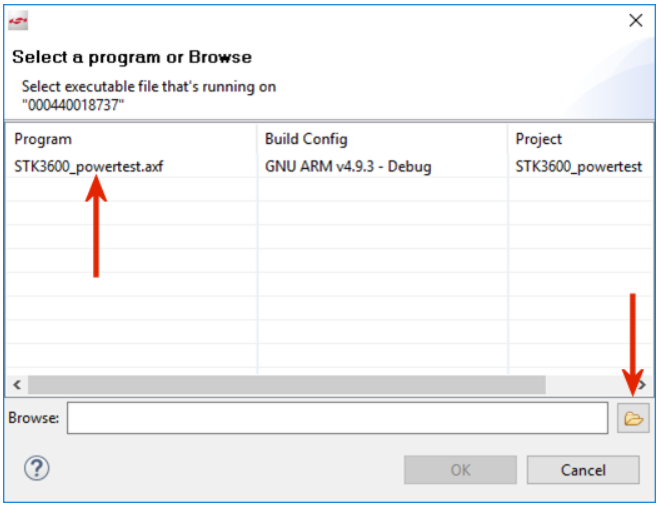


8.1.2 Code Correlation Enabled from the Multi-Node Energy Profiler

If you start energy capture with a running embedded application, Multi-Node Energy Profiler does not have the debug information it needs to associate program execution with source code. In these cases, the Code Correlation view is as shown in the following figure.

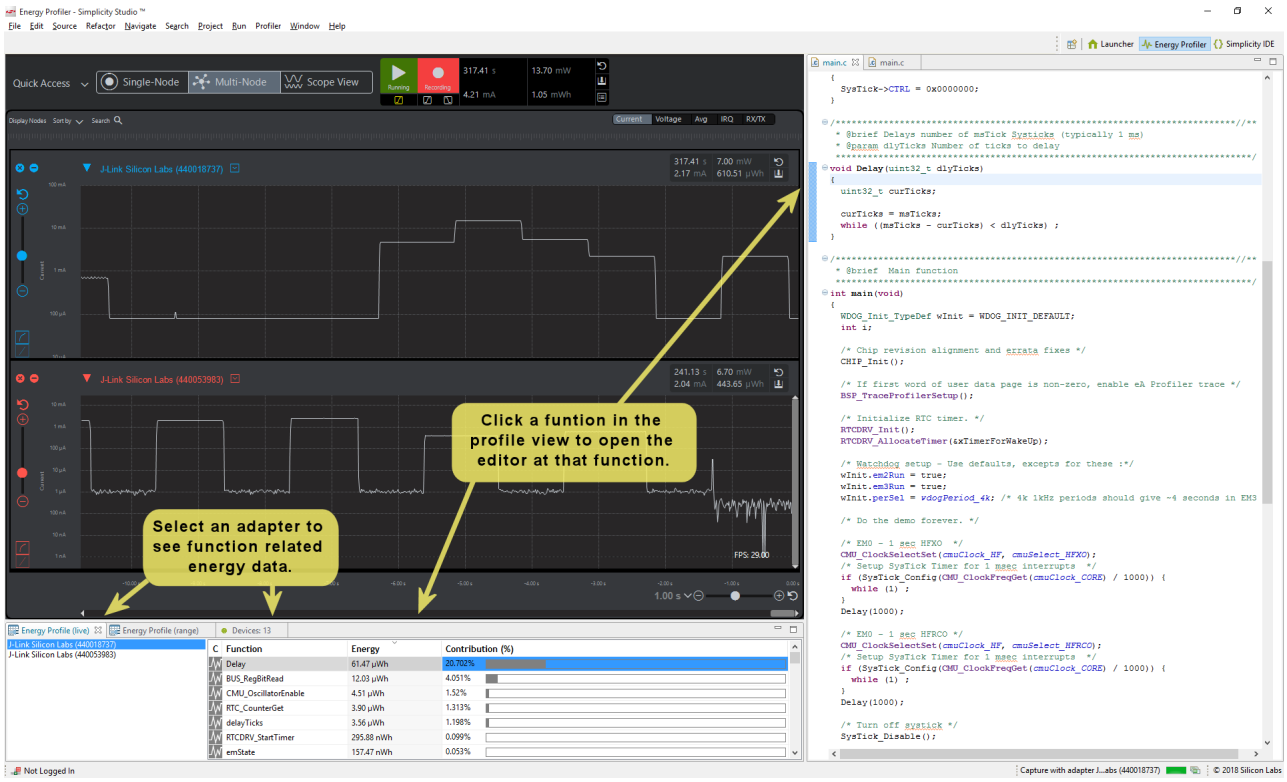


Click the "Select an Executable" link and the program selection dialog is displayed. Debug files from open projects are provided in the list. Alternatively, you may browse to select the debug file.



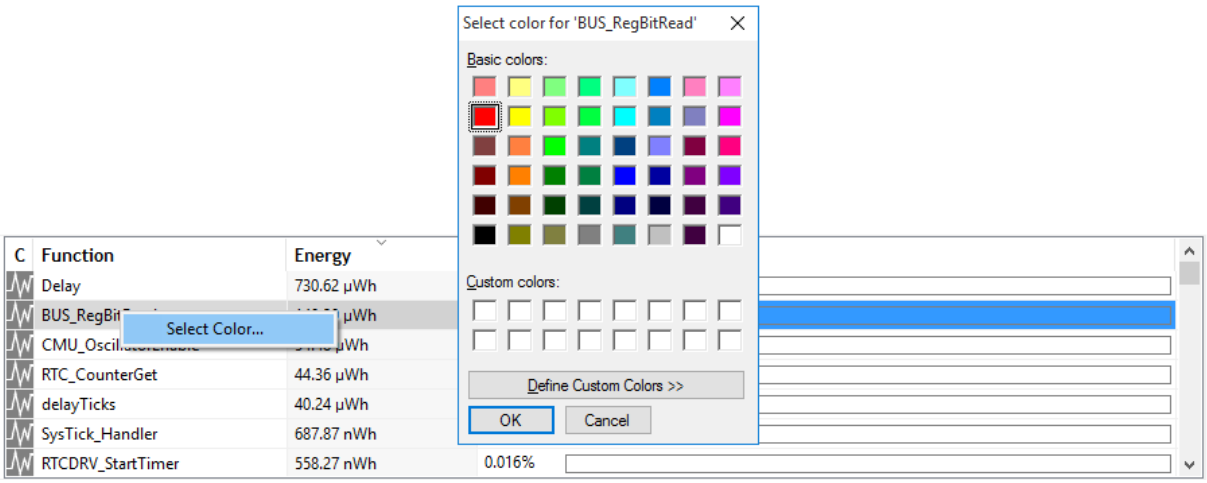
8.2 Code Correlation View

The Code Correlation view is shown below the waveform portion of the Multi-Node Energy Profiler Perspective, and works in conjunction with the Editor view. When multiple devices are displayed in the Multi-Node view, you can click on each adapter to view the function profile data associated with that device. The function data may be sorted as desired. Clicking on any function opens the associated source file in the Editor view with the start of scope for that function selected.

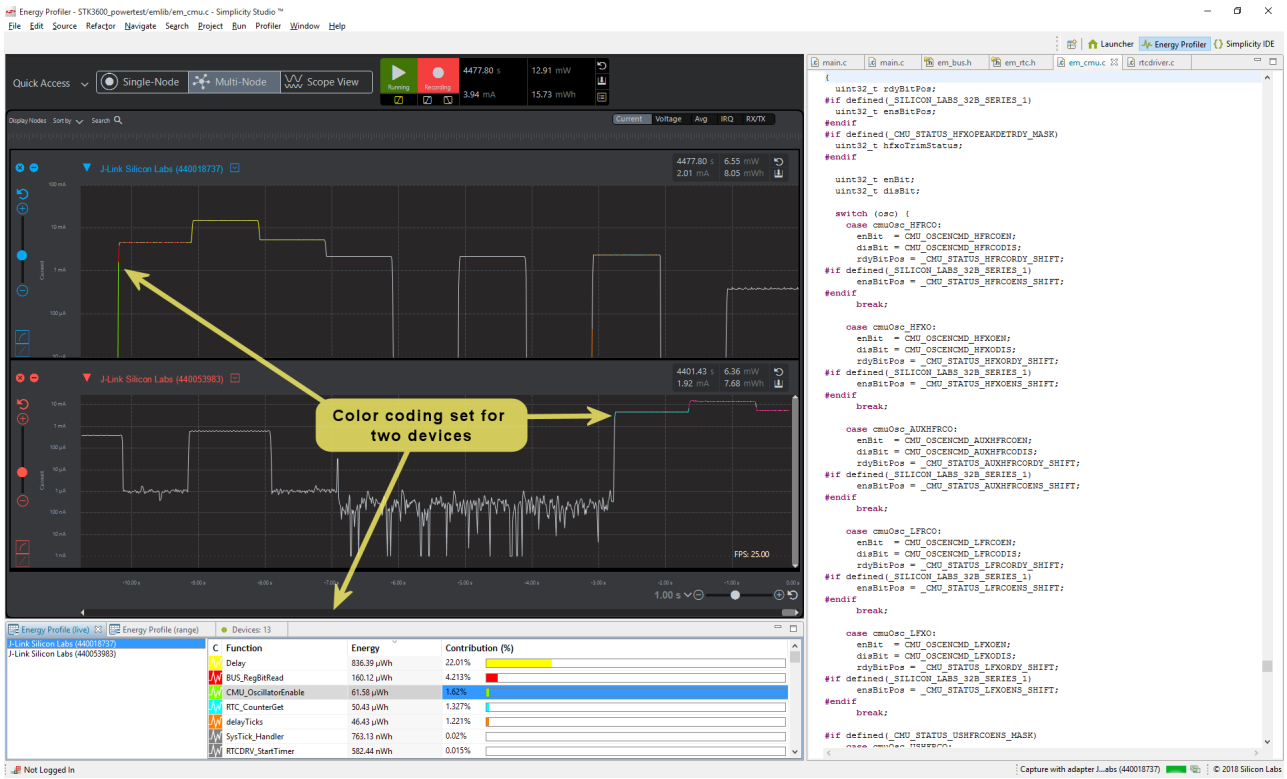


8.3 Color Coding and Code Correlation

Multi-Node energy profile provides the ability to color-code the current waveform based on program execution. You can set the color for each function using the context menus in the Code Correlation view.



The following figure shows the current waveform view with several functions set to different colors.

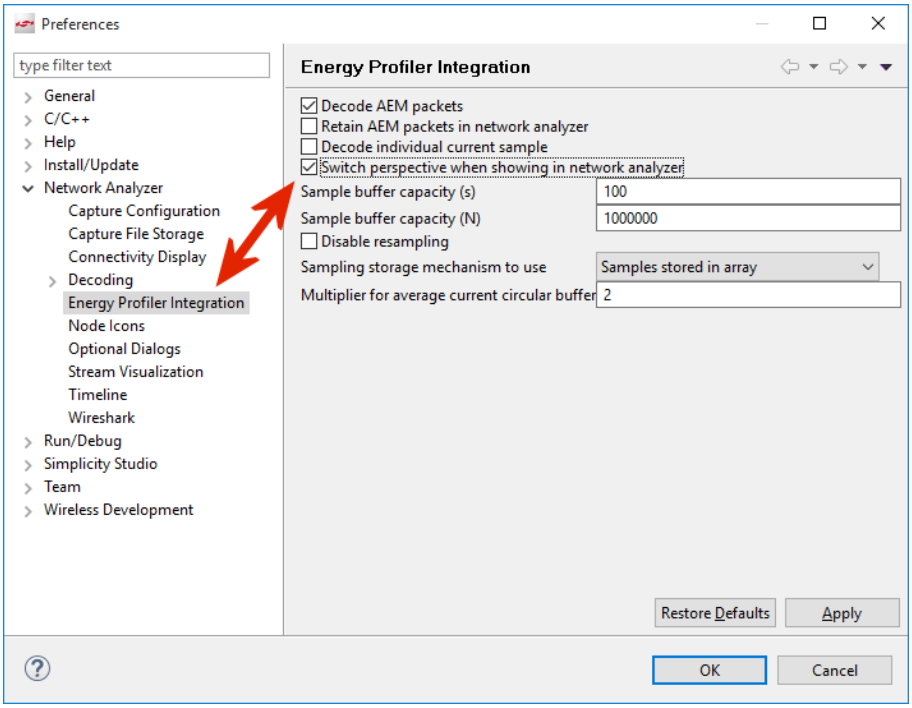


9. Multi-Node Energy Profiler and Network Analyzer Integration

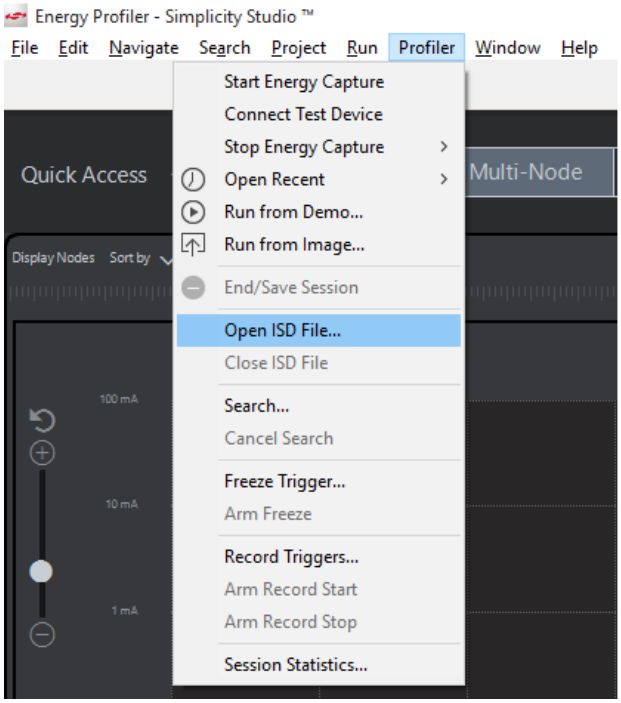
Note: This functionality is only available in offline mode with a previously saved ISD file.

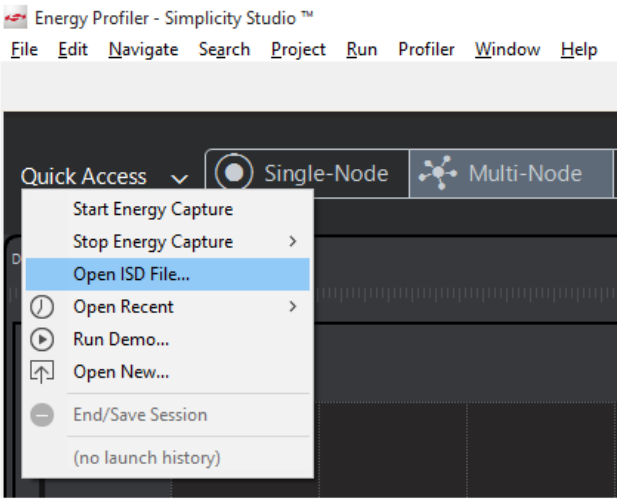
Multi-Node Energy Profiler and Network Analyzer now provide a basic level of integration when trace data from a debug adapter includes both AEM and PTI data. This integration allows you to easily move between the two applications by selecting an event in either one and selecting "Show in Network Analyzer" or "Show in Energy Profiler."

Simplicity Studio provides a preference that determines whether moving between applications leaves the Multi-Node Energy Profiler Perspective and opens the Network Analyzer perspective, or stays in the Multi-Node Energy Profiler perspective.

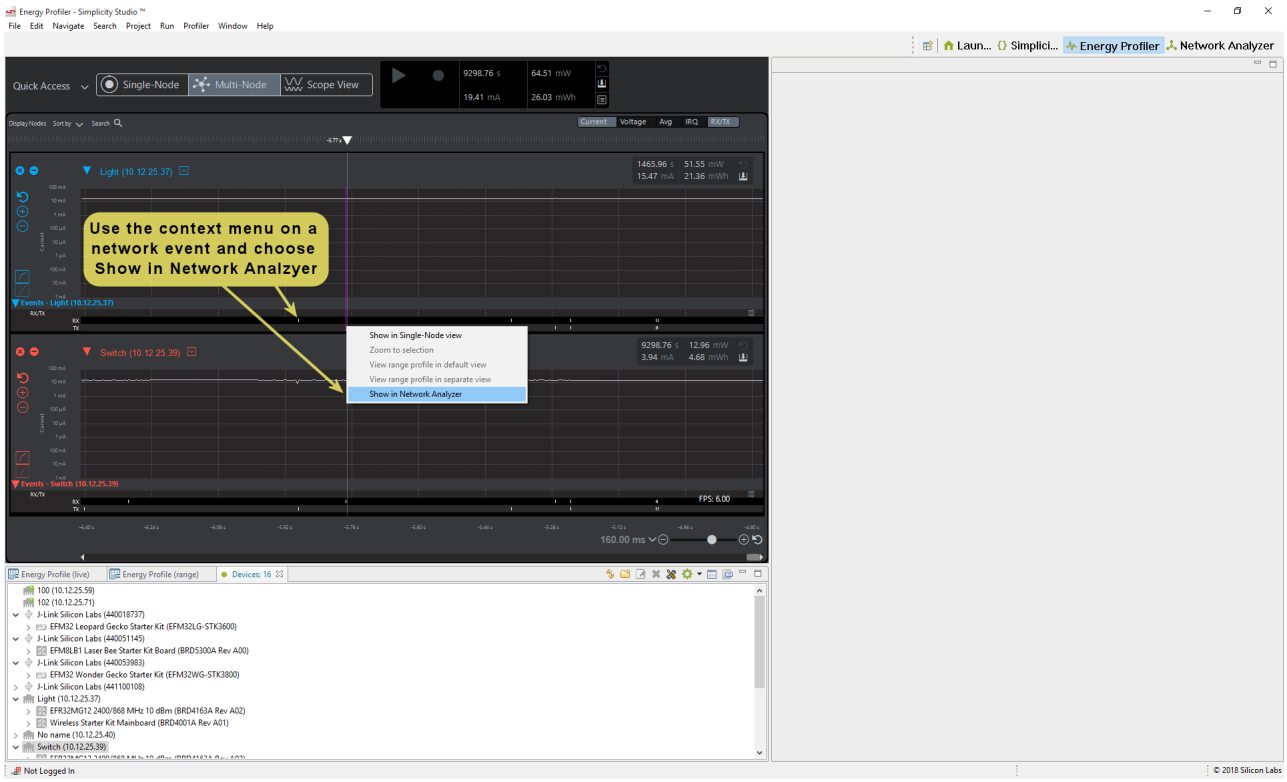


To open a previously saved ISD file, either use the top menu "Profiler->Open ISD File…" menu or the "Quick Access->Open ISD File…" menu, as shown in the following figure.

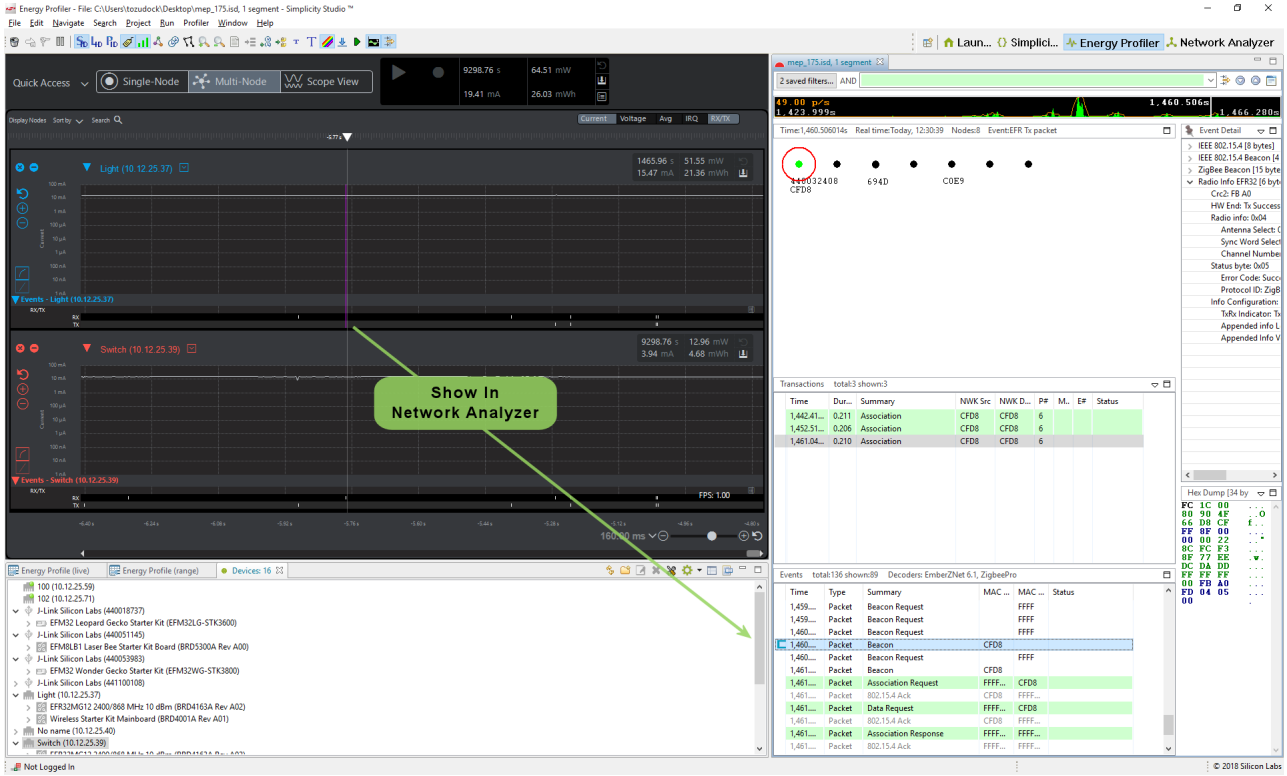




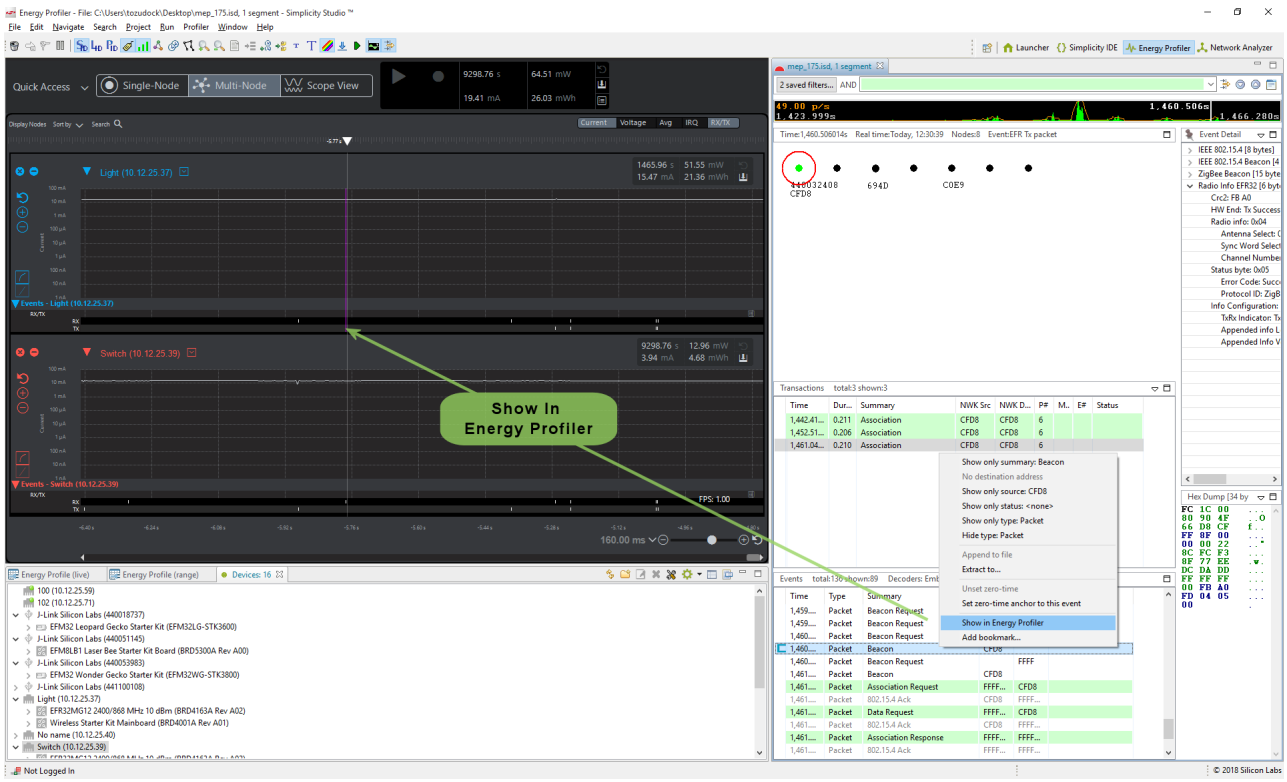
Once the file is open, you can examine previously recorded details. In the following figure, notice two devices transmitting and receiving packets. Select a TX or RX event and display the context menu. One of the options will be "Show in Network Analyzer."



Select that option and the Network Analyzer view opens either in the Multi-Node Energy Profiler perspective or the Network Analyzer perspective, based upon the preference setting.



Likewise, you can select an event in Network Analyzer and use the context menu to navigate back to Multi-Node Energy Profiler.



9.1 Use Cases for Multi-Node Energy Profiler and Network Analyzer

Depending upon your goals for an analysis session, there will be times when you will primarily use Network Analyzer, times when you will primarily use Multi-Node Energy Profiler, and times when you will find it advantageous to use them together.

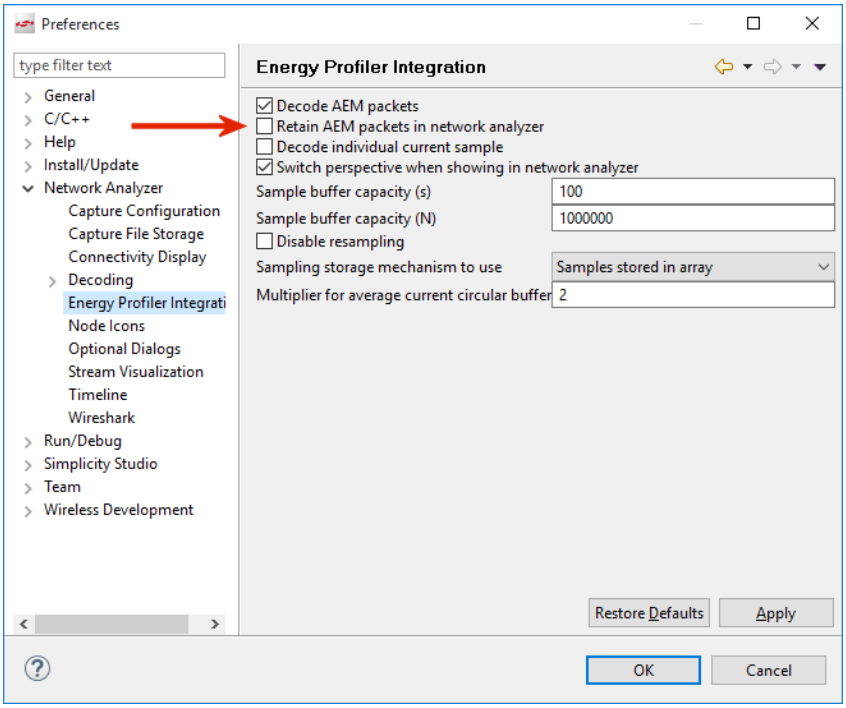
When you are interested in energy consumption of a single or multiple devices, Multi-Node Energy Profiler is where you will focus your time. From a network communication perspective, it provides only the start time of a TX or RX packet, and does not provide any decoding of the packets themselves. Still, this information can be revealing in your application's consumption of power relative to network activity. For example, by using the search tool, you can set search criteria that identify all points in the data where a packet was transmitted and the power exceeded a given value. If you have an expected power performance based upon your design, you can quickly identify when your system is operating outside its specification.

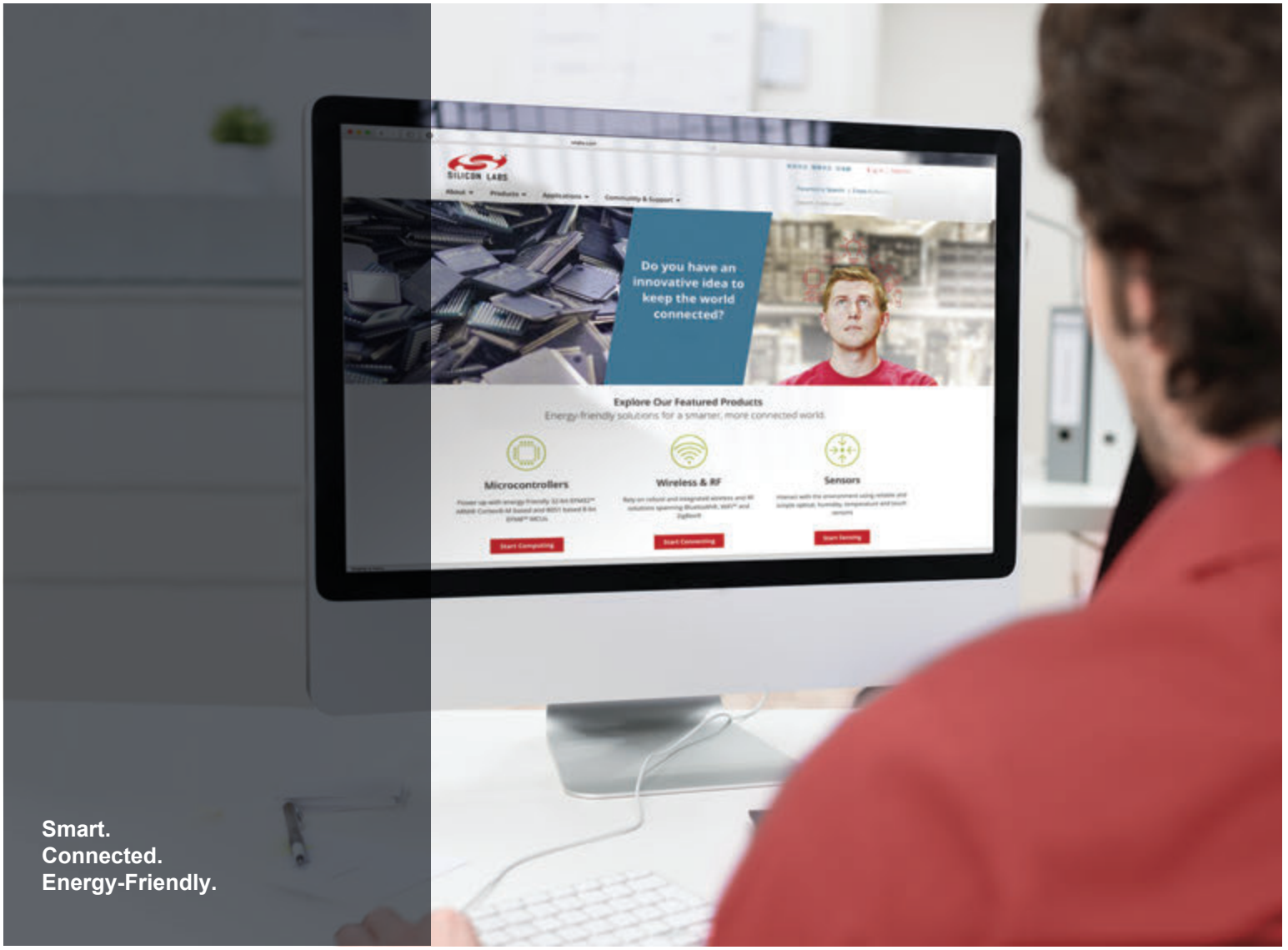
If you are interested in packet contents, node activity, and network interaction in a wireless network, Network Analyzer is where you will focus your time. Network Analyzer's multiple editor panes provide tiered displays of network activity, letting you drill down from a high-level map of node interactions to the details of each packet. Customizable filters enable you to specify exactly which network activities to display, allowing you to sift out information unrelated to a given task. These features allow you to determine when your network is behaving as expected, but are not available in Multi-Node Energy Profiler.

You will find yourself benefitting from the integration of these applications where these two information spaces intersect. For example, the Multi-Node Energy Profiler use case mentioned above ended at finding a transmit packet that exceeded expected power consumption. To investigate further, you could select "Show in Network Analyzer" to investigate that specific node's network activity and packet contents, in hopes of determining why the power was higher for that particular network transaction. In contrast, you may suspect that packets with specific content are the root cause of high power consumption. In Network Analyzer you could search for that data, and use "Show in Energy Profiler" to determine if indeed those packet contents are the root cause of high power consumption.

9.2 Post Analysis Using an ISD File

Both Network Analyzer and Multi-Node Energy Profiler capture data in the same format, which are saved in the ISD file format. Both applications can open an ISD file that has been save by either application. For example, if you have completed a Multi-Node Energy Profiler session and saved the ISD file, only to realize later that you are interested in the packet trace information, you would open that file from Network Analyzer to further investigate it. It is important to note, however, that captures started in Multi-Node Energy Profiler include packet trace data by default, but captures started in Network Analyzer only contain energy data if the preference to include it has been selected, as shown in the following figure.





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