# AN1346: Running the BLE Interoperability (IOP) Test Application Note

Interoperability (IOP) is one of the key value propositions of Bluetooth Low Energy and something that consumers have come to expect from Bluetooth enabled end products.

This document describes Silicon Labs IOP test framework comprising of hardware kits, embedded software, and mobile app. It also explains the requirements for building the IOP test setup, running the test, and collecting data for further analysis.

**Note:** This application note is valid for GSDK 3.2.x where interoperability is only available as a pre-compiled demo. For GSDK 4.0 and onward please refer to the readme file which is presented when opening the interoperability sample application.

**KEY FEATURES**

- Software and Hardware requirements to run the IOP test
- Bringing up the test environment and running the test
- Collecting test data from EFR32 and mobile phone

 **Copyright © 2022 by Silicon Laboratories** 

# Table of Contents

# 1. Introduction

IOP is a cornerstone of Bluetooth and one of the key reasons why this wireless technology has become ubiquitous. It enables end users to mix and match devices between different vendors without fearing connectivity issues. For example, whether a heart rate monitor from a company A will connect to a smart watch from a company B to retrieve and display the heart rate information.

It is therefore essential that when a customer is looking for a Bluetooth solution supplier for his design, the supplier can provide means to test IOP between his Bluetooth solution and 3rd party devices.

One of the most common use cases for Bluetooth enabled devices is interaction with smartphones where a mobile app is used for command and control of the Bluetooth device. This use case places IOP in the spotlight because of the large number of permutations between smartphone hardware (namely Bluetooth chipset), low level firmware (typically BLE link layer), mobile OS (typically BLE host stack), and mobile OS version.

Silicon Labs provides a framework to test IOP between the EFR32 family of SoCs and a large number of smartphones currently on the market. This framework is used to run IOP testing against a large list of devices periodically. AN1309 contains both IOP test results and the IOP test plan.

Subsequent sections list the requirements for the IOP test framework. Subsequent chapters describe how to bring up the test environment, run the IOP test, and collect data for further analysis.

## 1.1 Hardware Requirements

The IOP embedded software is available for the following radio boards:

- BRD4104A (based on the xG13 SoC)
- BRD4181A (based on xG21 SoC) – currently in obsolete state and may no longer be available for purchase. However, if you have already purchased this board, it can be used for IOP testing
- BRD4181B (based on xG21 SoC – a new version of BRD4181A with a corrected pinout, more details in the PCN.
- BRD4182A (based on the xG22 SoC)

You can purchase the following boards stand-alone or as a part of a starter kit.

- BRD4104A can be purchased stand-alone with OPN SLWRB4104A or as part of SLWSTK6020B starter kit
- BRD4181B can be purchased stand-alone with OPN SLWRB4181B or as part of SLWSTK6006A starter kit
- BRD4182A can be purchased stand-alone with OPN SLWRB4182A or as part of SLWSTK6021A starter kit

## 1.2 Software Requirements

The IOP embedded software is available in the Silicon Labs GSDK starting with version 3.2.0 and later. Users should install Simplicity Studio 5 and Bluetooth SDK 3.2.0 or newer, which is part of GSDK 3.2.0. For more information about installing Simplicity Studio 5, see QSG169: Bluetooth® SDK v3.x Quick-Start Guide.

## 1.3 Mobile App Requirements

To enable IOP testing framework on mobile, install EFR Connect mobile app, version 2.3 or newer. The app is available for both Android and iOS and the source is available on GitHub.

### 1.3.1 Minimum Mobile Operating System Versions

The minimum OS versions supported by EFR Connect mobile app are Android™ 9 and iOS®12.

## 2. Bringing up the Test Environment

The IOP test consists of a sequence of BLE operations executed between a mobile device and an EFR32 SoC running the interoperability test embedded software.

To flash the embedded software into one of the supported kits, plug the kit into the PC and open Simplicity Studio 5. In the Simplicity Studio launcher, select the “Example Projects & Demos” tab then filter by “Bluetooth” technology type and disable the “Example Projects” toggle switch. This will narrow down the results to a few ready-to-flash demos, which include the IOP test, as shown below

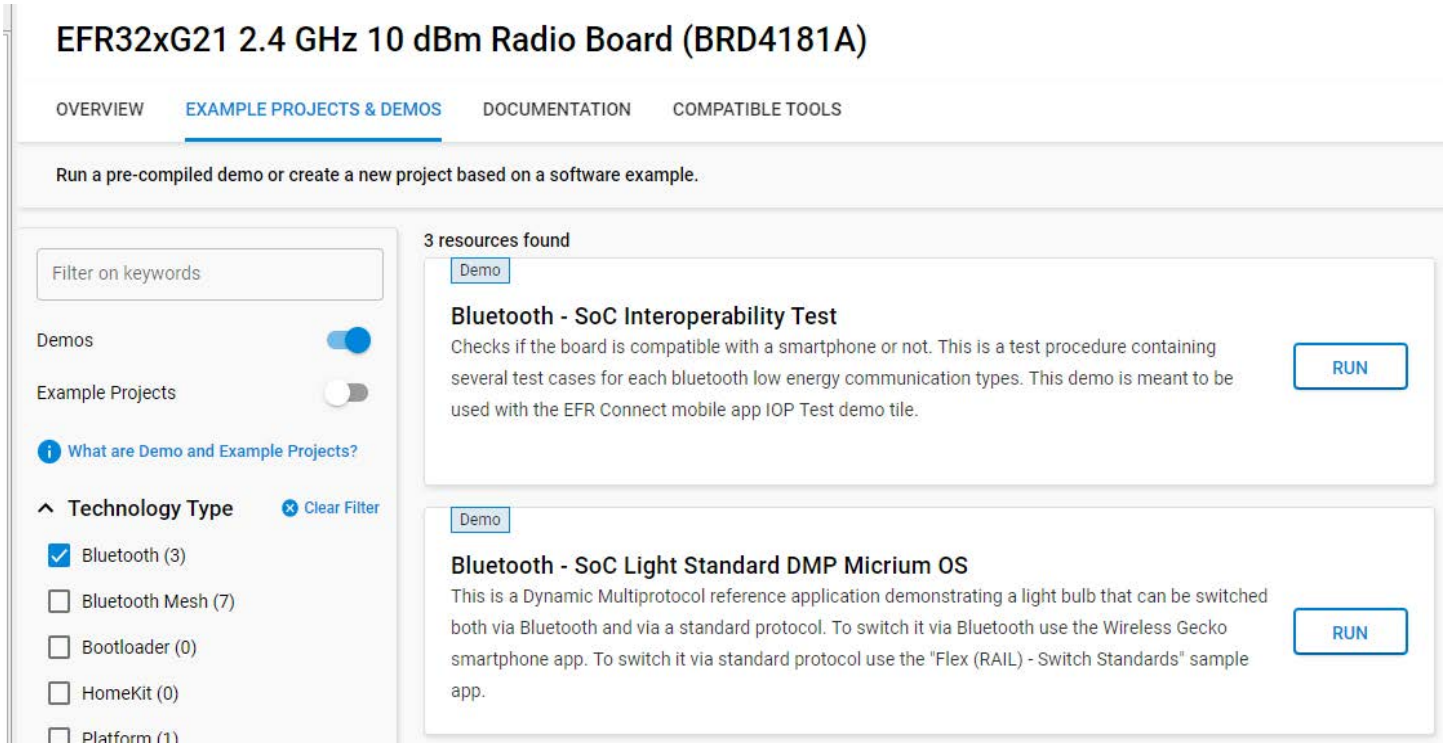


**Figure 2.1. IOP Test Demo in Simplicity Studio 5**

The embedded software can be flashed to the kit by clicking the “Run” button. After the flashing is complete, you should see the following information on the kit display, as shown below.

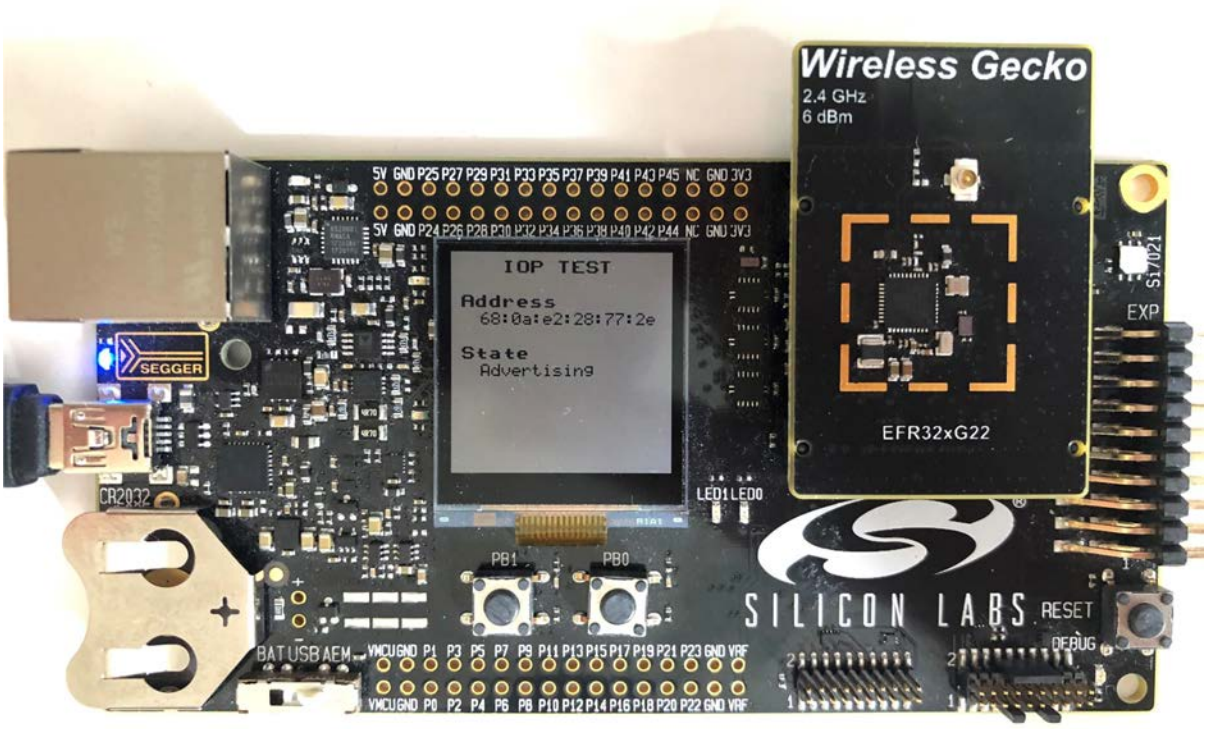


**Figure 2.2. BRD4182A Running the IOP Embedded Software**

For mobile, open EFR Connect mobile app which will automatically open in Develop view. Tap on the Interoperability Test Tile, which brings up the initial information pop-up with a summary of the feature and a link to this document. It is also pre-populated with the default device name. After tapping OK, it will switch to the IOP screen where the you can tap on “Run Tests” option to get started.

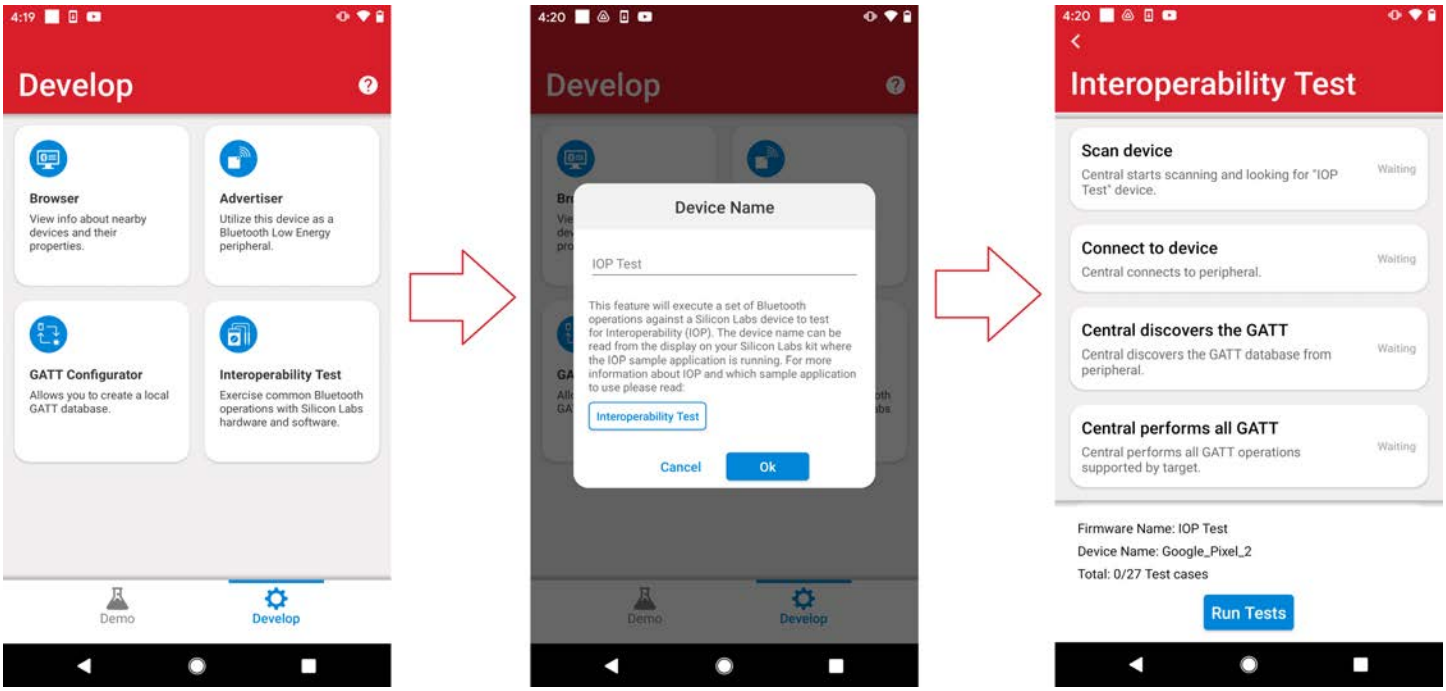


**Figure 2.3. Launching the IOP Test on EFR Connect Mobile App**

# 3. Running the IOP Test

After the IOP test sequence starts running, the mobile app will scroll through the test cases and indicate Pass/Fail when the test is completed, as shown below.

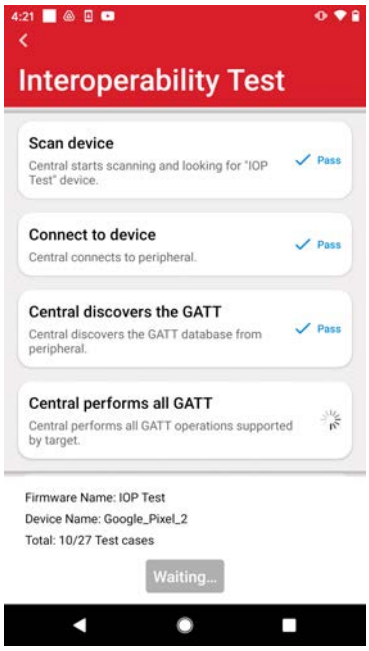


**Figure 3.1. Ongoing IOP Test**

Most tests do not require user intervention, the exception being the security tests. Both perform bonding where the user is prompted several times to bond with the device on the mobile app side. Some of those prompts require simple confirmation, such as Just Works pairing, while other prompts require inputting a pin code for authenticated pairing, which can be read from the kit LCD or from the UART logs.

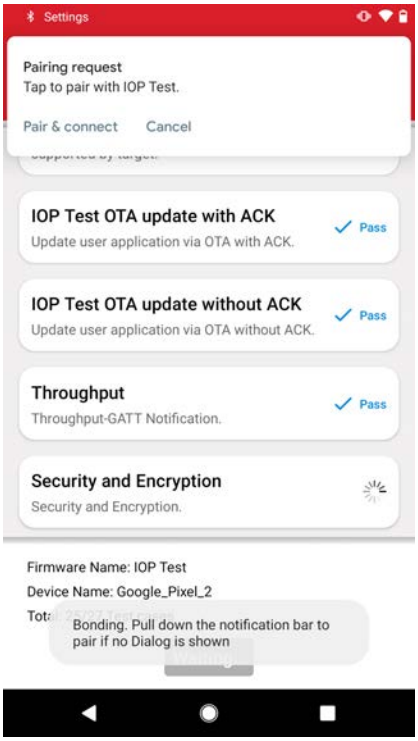


**Figure 3.2. Just Works Pairing**

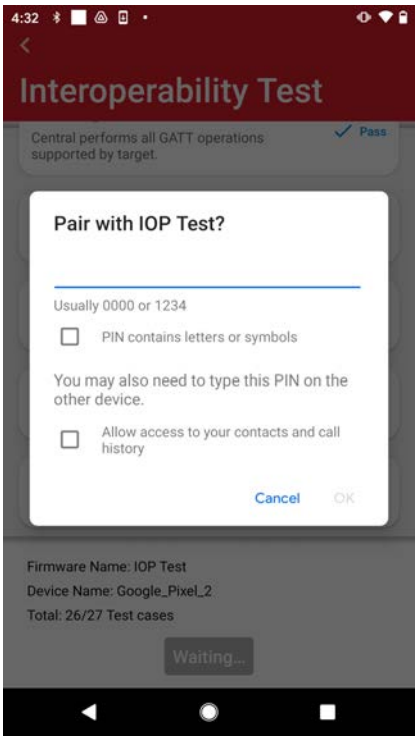


**Figure 3.3. Authenticated Pairing**

Ensure that there is no existing bond with the embedded device before initiating the IOP test sequence, which you can check from the phone Bluetooth settings. If the device is already bonded, the bond must be removed before proceeding with IOP testing.

# 4. Logging and Sharing Data

After the test is finalized on the mobile app, the user is given the option to rerun the test or share the results.

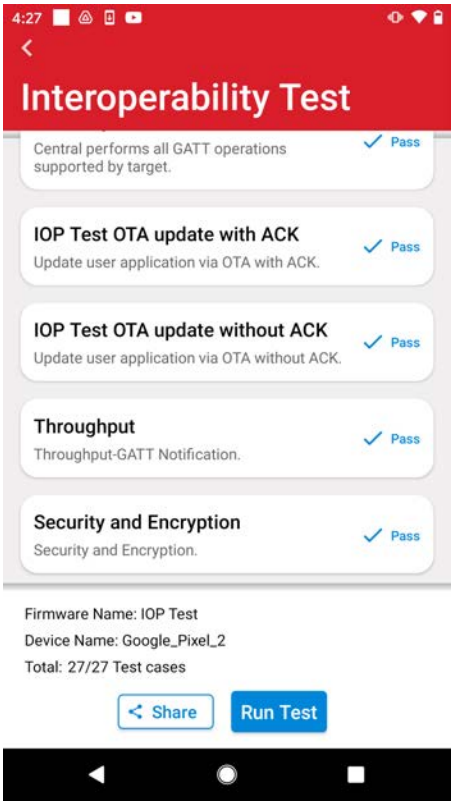


**Figure 4.1. Run and Share Test Options**

To rerun the tests, first reset the embedded device through the reset push button on the bottom right side of the kit mainboard. Additionally, remove the bond from the phone's Bluetooth settings.

The Share option allows sharing the test log through OS-standard mediums, such as cloud storage (e.g., Dropbox, Google Drive, iCloud, and so on), email, or saving it locally. The log is in xml format and contains information about the phone model, OS version, Bluetooth connection parameters, and the result of each test. Below is an example of a test log from running IOP test on a Pixel 2 with Android 11.

```
<timestamp>2021-05-12 23:17:11</timestamp>
<phone_informations>
        <phone_name>Google_Pixel_2</phone_name>
        <phone_os_version>Android_SDK:_11_30</phone_os_version>
</phone_informations>
<firmware_informations>
        <firmware_version>2</firmware_version>
        <firmware_ic_name>xG22</firmware_ic_name>
        <firmware_name>IOP Test</firmware_name>
</firmware_informations>
<connection_parameters>
        <mtu_size>247</mtu_size>
        <pdu_size>251</pdu_size>
        <interval>15.0</interval>
        <latency>0</latency>
        <supervision_timeout>208</supervision_timeout>
</connection_parameters>
<test_results>
        Test case 1,Pass.
        Test case 2,Pass.
        Test case 3,Pass.
        Test case 4.1,Pass.
        Test case 4.2,Pass.
        Test case 4.3,Pass.
        Test case 4.4,Pass.
        Test case 4.5,Pass.
        Test case 4.6,Pass.
        Test case 4.7,Pass,(Testing time: 149ms;Acceptable time: 300ms).
        Test case 4.8,Pass,(Testing time: 207ms;Acceptable time: 300ms).
        Test case 4.9,Pass,(Testing time: 156ms;Acceptable time: 300ms).
        Test case 4.10,Pass,(Testing time: 178ms;Acceptable time: 300ms).
        Test case 5.1,Pass.
        Test case 5.2,Pass.
        Test case 5.3,Pass.
        Test case 5.4,Pass.
        Test case 5.5,Pass.
        Test case 5.6,Pass.
        Test case 5.7,Pass.
        Test case 5.8,Pass.
        Test case 6.1,Pass.
        Test case 6.2,Pass.
        Test case 7.1,N/A,(Throughput: 80975 Bytes/s;Acceptable Throughput: 42293 Bytes/s).
        Test case 7.2,Pass.
        Test case 7.3,Pass.
        Test case 7.4,Pass.
</test_results>
```

**Figure 4.2. IOP Test Log Example**

### 4.1 Collecting Additional Data from the Embedded Device

The IOP embedded software also sends logging data over UART, which can be captured by a terminal emulator on the PC. Furthermore, the Packet Trace Interface (PTI) is enabled which means that the radio traffic can be captured using the Network Analyzer.

For a more comprehensive data set around an individual IOP test sequence, capture both UART logs and radio traces using Simplicity Studio. Radio trace capture must be initiated before starting the IOP test.

To start the radio capture, right click on the debug adapter and select "Connect". Then, right click once again and select "Start Capture".

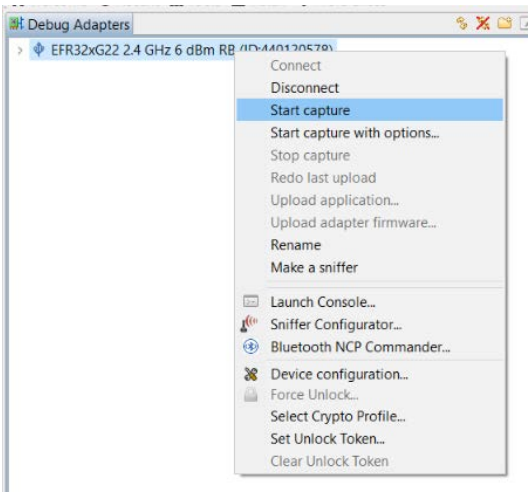


**Figure 4.3. Initiate Radio Traffic Capture Using Network Analyzer**

This will automatically bring up the Network Analyzer view where the traffic is logged and every packet can be decoded for further analysis if required. At the end of the IOP test, the trace can be saved through File -> Save as, as shown below.

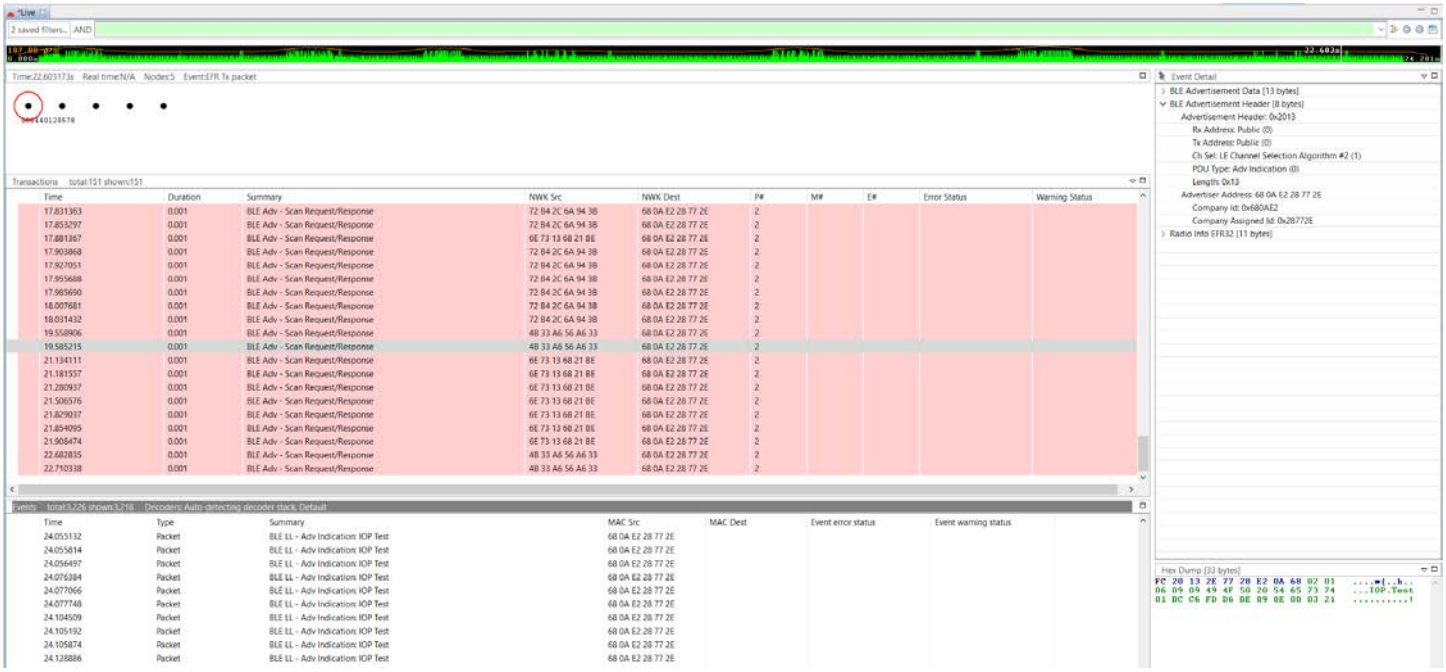

**Figure 4.4. Simplicity Studio Network Analyzer**

While UART logs have multiple COMPort emulators such as tera term, you can also capture within Simplicity Studio's own terminal. Similarly to the packet trace capture, right click on the debug adapter and select "Launch Console". Then, select "Serial 1" tab and enter any character into the prompt at the bottom of the screen. You should see the connection symbol changing, which indicates that the connection is successful and the logging should start, depending at which phase of the IOP test this is initialized. Otherwise, reset the board to ensure that the log is being received.

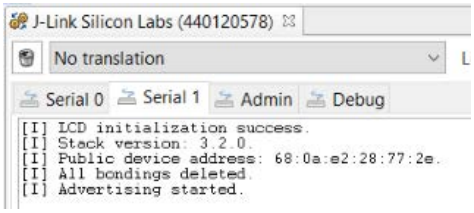


**Figure 4.5. Simplicity Studio Console**

# 5. Revision History

### Revision 0.2

January, 2022

- Added note to front page.

### Revision 0.1

May, 2021

- Initial release

# Smart. Connected. Energy-Friendly.


**IoT Portfolio**
www.silabs.com/products

**Quality**
www.silabs.com/quality

**Support & Community**
www.silabs.com/community

**Disclaimer**
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.
**Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project**

**Trademark Information**
Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals®, WiSeConnect , n-Link, ThreadArch®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, Telegesis, the Telegesis Logo®, USBXpress® , Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

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

**www.silabs.com**