BG22 Lab 1: Out-of-the-box Thunderboard Example Project

These lab procedures walkthrough programming the EFR32BG22, creating a Bluetooth beacon out of the box and introduce the development tools we will be using throughout this workshop, including our new EFR Connect Bluetooth development app.

KEY POINTS

- Installing Simplicity Studio and SDKs
- GATT Configurator
- EFR Connect Bluetooth Development App

Getting Started

Review the following material before starting the Bluetooth 5 labs. Ensure that you have the correct hardware and software to successfully complete the labs.

1.1   Hardware Requirements

- 1 Thunderboard™ BG22 Kit
- 1 Micro USB cable
- iOS or Android Mobile device

1.2   Software Requirements

- Simplicity Studio v4 (Windows.exe, Mac.dmg, Linux .tar)
    - o   Bluetooth SDK 2.13.4
    - o   Gecko SDK Suite 2.7.4
- EFR Connect Mobile App

1.3   Simplicity Studio Installation

Option 1 New Installation – Offline Installer

1.   Install **Simplicity Studio v4 b**y launching **Offline Installer** (Windows.exe, Mac.dmg, Linux .tar)

2.   You'll need to create or sign in with your www.silabs.com account

Option 2 - Studio Previously Installed with Other SDKs

1.   Update existing Simplicity Studio Installation

2.   Update Protocol SDKs by clicking menu bar **Help** -> **Update Software**.
    a.   Click Package Manager

    b.   Click on tab for "SDKs" in Package Manager window
        i.   Select and Install Bluetooth SDK – 2.13.4.0
        ii.   Select and Install Gecko Platform – 2.7.4

## 2   GATT CONFIGURATOR TO CUSTOMIZE BLUETOOTH PROJECTS

Silicon Labs provides example projects that help evaluate different features of their Bluetooth solutions. This lab starts by loading a wireless starter kit with an example of a Thermometer device. Using the default image in a large group can cause issues when all of the devices broadcast the same device name.

The GATT Configurator in Simplicity Studio makes it easier to modify the characteristics within a Bluetooth device. Complete the following steps to help distinguish your wireless starter kit.

2.1   Create SoC Example Project

With Simplicity Studio open:
1.   Plug Wireless Starter Kit into USB Port. Click on the kit listed under Debug Adapters in upper left.
2.   Click on the **Getting Started** Tab from the Launcher dashboard in Simplicity Studio.
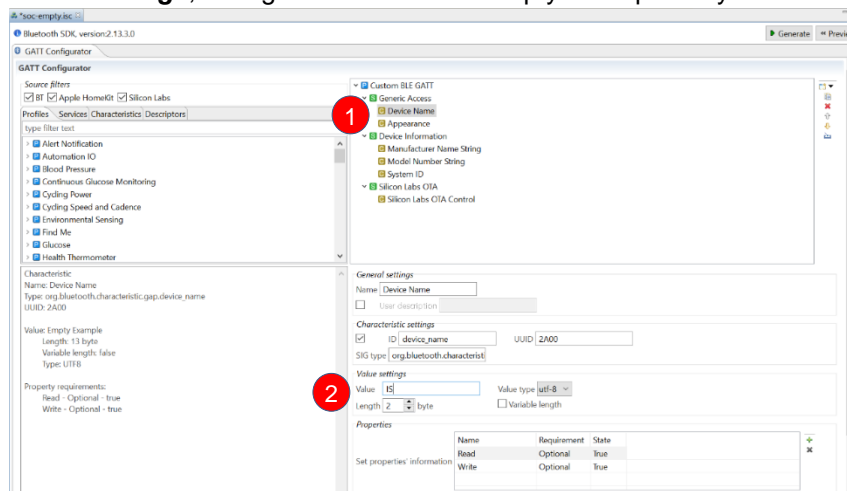3.   Click on the **SoC-Empty** project listed under Software Examples.
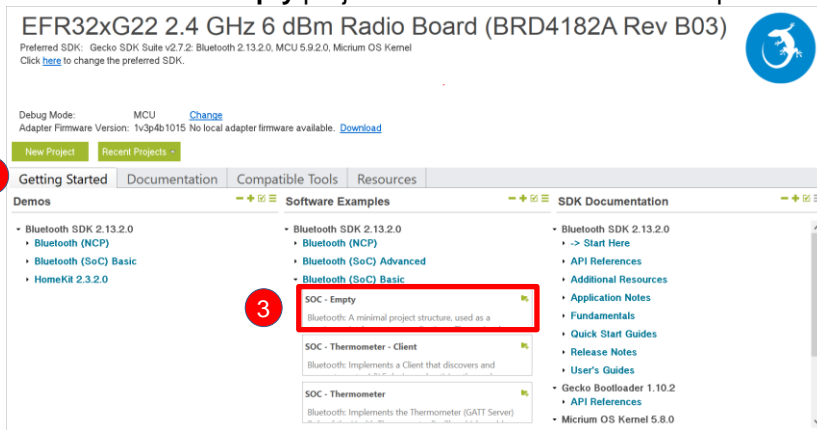
4.   Click **Yes** when asked to switch to Simplicity IDE perspective and create the project.
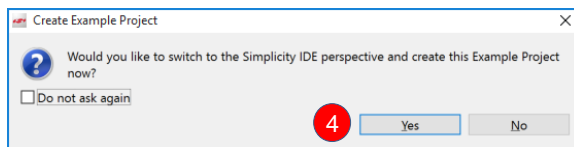
### 2.2   Modify GATT File for Empty Project

The BLE GATT Configurator window should be displayed in the Simplicity IDE once the project is created.
(Note: If not shown, Click **soc-empty.isc** in the Project Explorer pane on the left)
1.   Click on **Generic Access** and then **Device Name** under the Custom BLE GATT list in upper right.
2.   Under *Value settings*, change the value from "Empty Example" to your INITIALS and set length to 2 bytes.

3. Now add a new service by clicking on the Add button in the upper right corner and selecting New Service
4. Select the new Service
5. Add a new characteristic by clicking on the Add button in the upper right corner and selecting New Characteristic
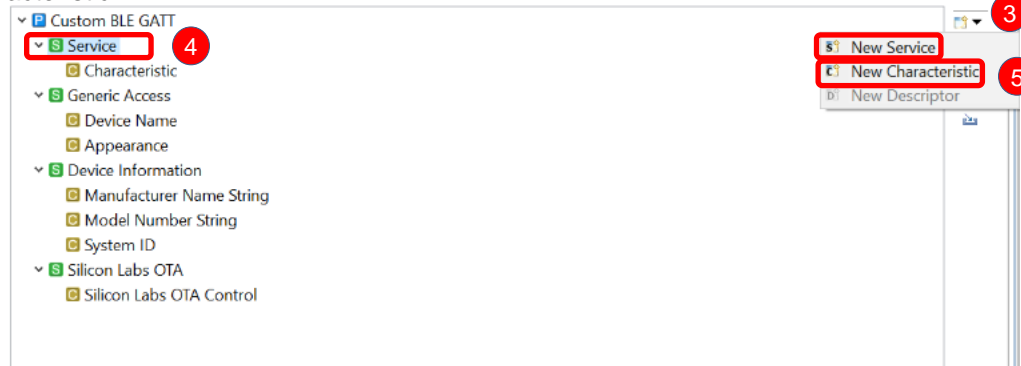
6. Now select the new characteristic
7. Change the name to **Read_Write**
8. Tick the checkbox next to the ID field in the Characteristic settings
9. Set the ID to "**Read_Write**"
10. Set the Value to "**R**"
11. Set the Value type to utf-8
12. Set the Length to 1
13. Add Read property by clicking on the Add button in the Properties tab and selecting Read
14. Add Write property by clicking on the Add button in the Properties tab and selecting Write

15. Click **Generate** to have the modifications updated in the project GATT file.
16. **Click OK** through the pop-up windows.

**2.3   Program Thunderboard with Empty Project**

1. Click **Build icon** in the tool bar at the top of the Simplicity IDE.
   The project will compile based on its build configuration.

   **Note:** You may need to select the project in the Project explorer view on left

2. Press the **Debug icon** in the tool bar. This will flash the project onto the board if it was successfully built. This only updates the program memory.
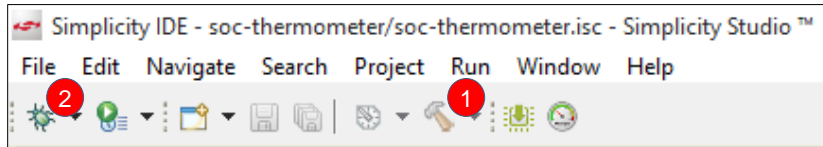
   Error: DP Write Failed - Press the Reset button on Thunderboard or unplug/replug then Flash again within 30 seconds.

   The Thunderboard demo app which ships on the boards goes into a low energy mode (EM2) after 30 seconds. When the device is in EM2, the debug interface is unavailable, and DP write fails. We can wake the device by resetting Thunderboard.

3. A device selection prompt will appear select your J-Link device
4. Next Click to Query the Lock Status
5. Click OK
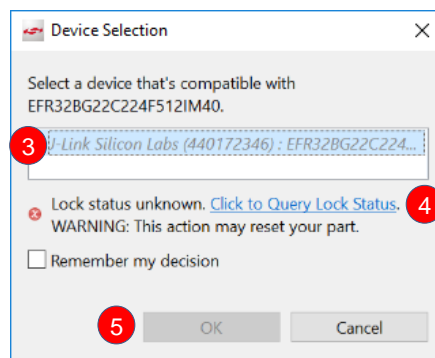
   Note: The prompt to select the device and the "Click to Query Lock Status" is expected for EFRBG22. Studio doesn't know if the device is secure debug locked or not. The only way to know is to send a DCI command which will reset the part. So, we don't allow any DCI commands to run in the background without user intervention. Studio caches the status, so you'll only see this happen the first time you use the device and when you re-connect the device.

6. Once the project is flashed, the Debug perspective will be shown. Click the **Resume icon** in the tool bar to have the application start.
7. Click the **Disconnect icon** in the tool bar to end the Debug session.
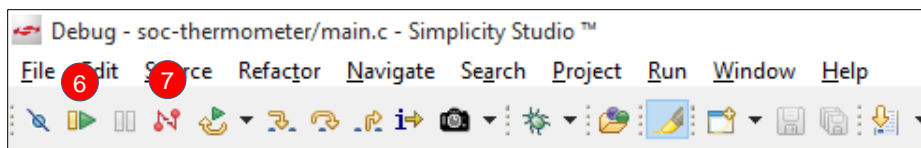
3   READ/WRITE GATT CHARACTERISTICS WITH EFR CONNECT MOBILE APP

With the launch of BG22 we also release our new Bluetooth Development app, EFR Conncect

New features offered by EFR Connect:
- Simultaneous connections with multiple Bluetooth Low Energy peripherals
- Bluetooth communication logging
- Advanced filtering for device discovery
- Custom naming for services and characteristics

Install/Open EFR Connect off your mobile app store to modify the name of your BG22 device.

1. Press **Develop** and select **Browser** to view nearby Bluetooth Devices.
2. Press the **Filter** tab to narrow your search and find your device.
3. **Input** your INITIALs and press search
4. Connect to your device to view the GATT Profile and Characteristics.
   > Note: Generic Services (UUID: 1800) shown below will not appear on iOS as Apple does not these to be visible to applications.
5. Tap **Unknown Service** and edit its name to Custom Service.
6. Tap **Unknown Characteristic** and edit its name to Read_Write

Note: there is no standard way of displaying the names of custom services/characteristics. When the app sees a 128-bit UUID it knows that it's custom, so the name is always "Unknown service", this is the same for most generic BLE apps out there.The workaround is to have your own app and then write the service name into a characteristic so that your app can read it from there. EFRConnect takes care of this to ease development.

7. Under you're Read_Write characterstic Tap Write.
8. Write **W** for your device to modify this characteristic and hit **Save.**
   EFRConnect can be used to devlop and test your GATT Services and Characteristics.
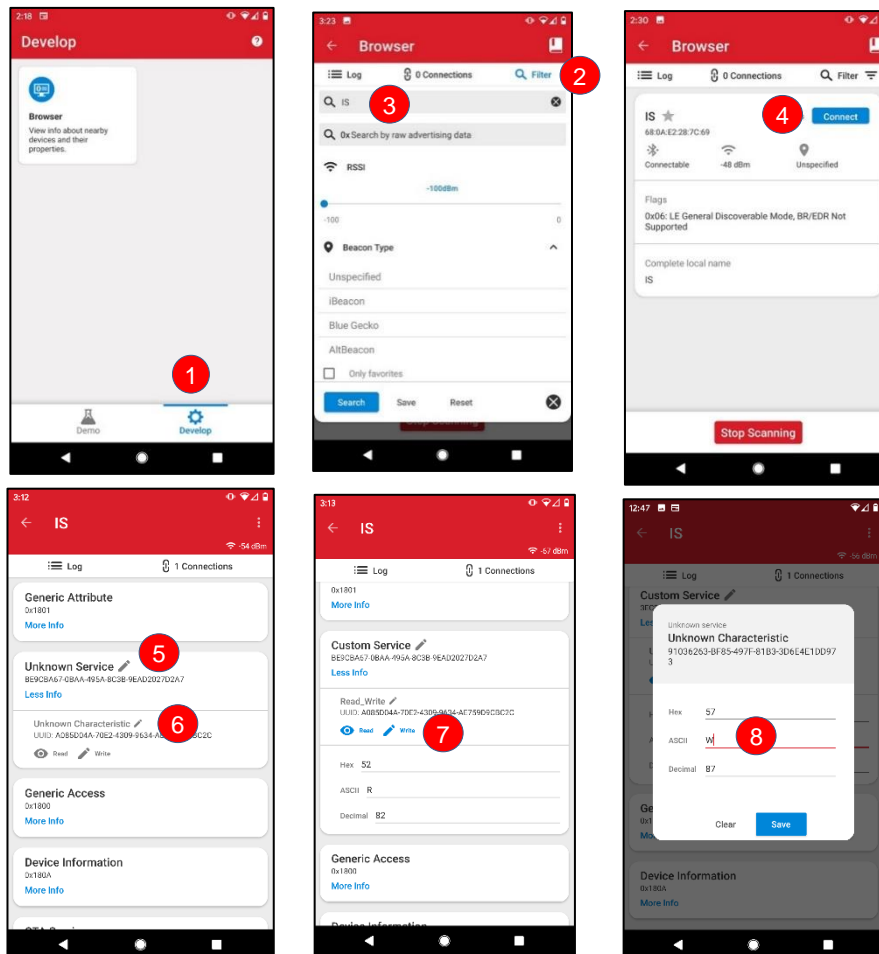
4   TROUBLESHOOTING

4.1   Problem Occurred: 'Programming Flash' has encountered a problem. DP write failed

The Thunderboard demo app which ships on the boards goes into a low energy mode (EM2) after 30 seconds. When the device is in EM2, the debug interface is unavailable, and DP write fails. We can wake the device by resetting Thunderboard.

**Press the Reset button on Thunderboard and Flash again within 30 seconds.**

If problem persists, we can force a 'Device info' DCI command to trigger a reset, which restores access.

The most direct way to do this is to **'Recover Bricked Device'** using Simplicity Commander Tool

1. Navigate to Launcher Window
2. Verify your device is selected
3. Click Compatible Tools Tab
4. Open Simplicity Commander
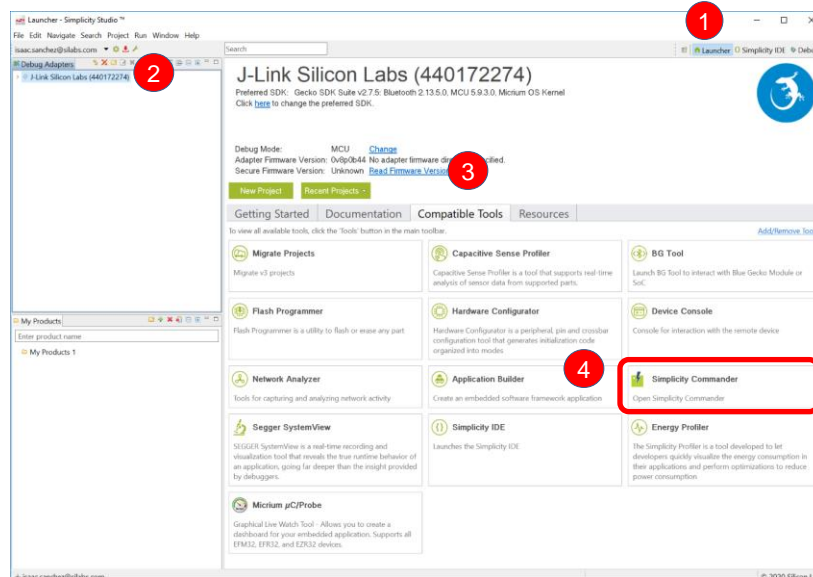
5. Connect to Adapter
6. Connect to Target
7. Select Flash
8. Click Recover bricked Device
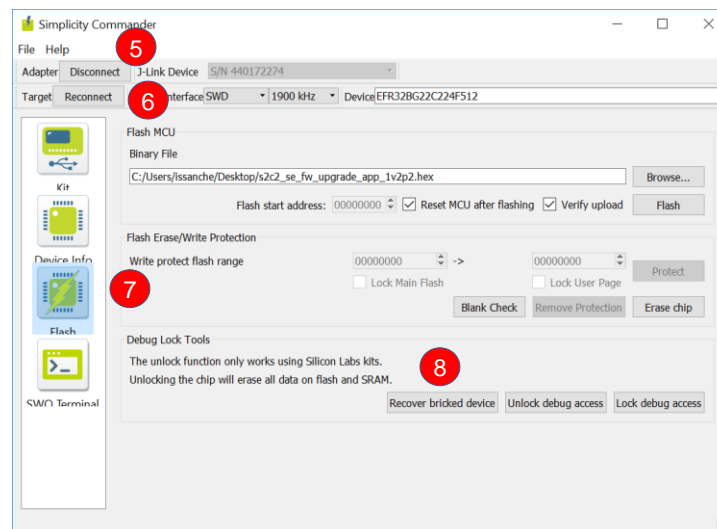
Note: This erases the image on the chip including the bootloader which is necessary for Bluetooth applications. Next section goes over a quick way to flash a bootloader to your device.

4.2 SoC-Empy Example Doesn't Appear in EFR Connect

After flashing the Thunderboard with the SoC-Empty image, you may not see it in the EFR Connect App if the device has no bootloader present. The bootloader is a program stored in resereved flash memory that can initialize the device update firmware imgages and possible perform some integrity checks. Out of the box, the Thunderboard Demo includes a bootloader image that will not be overwritten when a device is flashed with other examples unless otherwise dictacted by the user. To quickly replace a bootloader image, you can simply flash your device with one of the pre-built Demos in Simplicity Studio when you select your device from the debug adapters pane.

Flashing SoC-Empty Demo will run the empty beacon on Thunderboard with 'Empty Example' as the device name and no additional GATT service added in Lab 1.

1. Navigate to Launcher Window
2. Verify your device is selected
3. Click Getting Started tab
4. Expand Bluetooth (SoC) Basic under Bluetooth SDK Demos
5. Click 'SoC – Empty'
6. In the Pop-Up verify the mode is set to Run.
7. Click Start.