Energy Profiler and Network Analyzer Training Lab manual for Bluetooth SDK V2.0.0 or later

1 Requirements

To be able to use Multinode Energy Profiler you will need the followings:

- Simplicity Studio Platform 4.1.15 or later
- Simplicity Energy Profiler core 4.0.8 or later
- WSTK firmware v1.3.3 or later

2 Setting up Example Project

First let’s start with one of the SDK examples as a basis.

1. Make sure you have connected one of the Bluetooth SDK compatible boards to your computer, and the device is visible in Simplicity Studio v4. Click on the device to select it.

2. Check that the Preferred SDK is set to Bluetooth Smart SDK v2.0.1 or later.

3. Since Bluetooth SDK v2.7 the bootloader is not included in the example projects, so it has be flashed to the device first. The easiest way to do so is flashing a Demo to the device, that includes the bootloader

   a. click on SoC – Empty in the Demo column
   
   b. select Run mode
   
   c. click Start

4. Now create a new Thermometer project

   a. either click on SOC – Thermometer in the Software Examples column
   
   b. or click on ‘New Project’ and follow the steps
      (Bluetooth Smart framework → select latest SDK if prompted → SOC – Thermometer → name your project → select compiler)

5. The Simplicity IDE perspective is opened and your project is ready to be built.
3 Measuring Energy Consumption with Energy Profiler

Energy profiler can be used to monitor the energy consumption of a Blue Gecko Modules and SoCs online.

1. Click the “Profile as” button and select Simplicity Energy Profiler Target in the popup window

2. Click OK. This will automatically
   a. Build your project
   b. Upload it to the device
   c. Start Energy Profiler
3. A new Energy Profiler perspective appears. You can switch easily between Simplicity IDE and Energy Profiler perspectives in the perspective switcher of Simplicity Studio in the upper right corner of the main window.

4. Stop the recording of energy consumption by clicking on

5. You can see peaks in the energy consumption diagram. Click on one of them and zoom in with the time axis zoom
Note, that the maximum consumption is much greater than you could see it on the diagram before zoom! This is because in zoomed-out mode, the shown values are averaged. If you need exact values, always zoom in!

You can also see, that the peak consist of 3 distinguishable peaks. These peaks correspond to the 3 advertisement packets broadcasted on the 3 advertisement channel.

When the device is not advertising, it is put in deep sleep mode. Hence you can measure sleep current in other time instances.

6. To measure average consumption, simply click and drag your mouse over a time interval. A new box will appear showing consumption information for the given interval. Here we use 100ms advertisement interval, hence we chose the measured interval to be 100ms to obtain a proper average consumption. Overall average is measured as well, but this is influenced by transient events.
7. Click on the Quick Access button, and click on End Session to disconnect from the device.

8. Switch back to Simplicity IDE perspective

9. Open main.c and find the line
   
   ```c
   gecko_cmd_le_gap_set_adv_parameters(160,160,7);
   ```

   change it to
   
   ```c
   gecko_cmd_le_gap_set_adv_parameters(160,160,5);
   ```

   This will result in advertising only on 2 channels instead of 3 channels.
Also change

```
.sleeep.flags=SLEEP_FLAGS_DEEP_SLEEP_ENABLE,
```

to

```
.sleeep.flags=0,
```

This will disable sleep mode, i.e. the device will not be put into sleep mode if there is nothing to do.

10. Click on the Profile icon again to start profiling, pause measurement and zoom in to a peak. Now you can see, that there are only 2 peaks corresponding to the 2 advertisement packets, and higher consumption around them, since the device remains in EM0 mode, instead of being put into deep sleep.

11. If you have multiple devices connected to your PC, you can also add the other device to be profiled. It can be useful when debugging 2 devices communicating to each other.
   a. Select Multi-Node view at the top
   b. Click Quick Access
   c. Start Energy Capture
12. Now you can see the energy consumption of the two devices parallel. Each of them has own statistics.

13. To end your session and disconnect from your device(s) click on Quick Access button, and click on End Session

14. Switch back to Simplicity IDE perspective
4 Tracing packets with Network analyzer

EFR32BG devices have a so-called Packet Trace Interface (PTI) – a peripheral included in the MCU. With this interface all packets transmitted and received by the radio can be obtained from the device. PTI is enabled in all Software Examples by default, so you can readily use it, after flashing an example. PTI sends packet data to the WSTK which in turn sends it to Simplicity Studio in the correct format. You can use the Network Analyzer tool to display the packet data received from your device.

1. Click the arrow next to the “Profile as” button and select Profile As Network Analyzer Target

2. Select the device and click OK on the popup window. Uncheck “Launch console” and “start capture session” for now.

3. A new Network Analyzer perspective appears. You can switch easily between Simplicity IDE and Network Analyzer perspectives in the perspective switcher of Simplicity Studio in the upper right corner of the main window.

6. Unlike Energy Profiler, Network Analyzer does not automatically upload the newly built software, so first make sure, the software to be tested was uploaded to the device.
7. Select **File > Preferences**. Browse for **Network Analyzer > Decoding > Stack Versions**, and make sure Bluetooth Smart is your default stack profile. This ensures, that the packets are parsed as Bluetooth packets. Press OK.

8. Right click on your device and choose Start capture. This will open up a new window and start capturing your packets.
9. While capturing, open your Blue Gecko app on your smartphone, and connect to Thermometer Example.

Blue Gecko app - Connecting to thermometer demo
10. Stop capturing with right click on the device > Stop capture

11. Investigate the captured packets
You will find, that related packets like requests and responses make together a transaction. These transactions are listed separately in the Transactions window. To find the first packet of the transaction, simply click on the transaction. To see the details of the packet, simply click on the packet. You can see both the raw and the parsed format of the packet in the Hex Dump / Event Detail windows.

Now examine some packets:

a. All connection starts with advertising. One of the devices has to advertise itself, in order to make possible for others to connect to them. Find the ADV indication packet. Click on it, and find the name of the device in the Hex Dump and in the event details.

b. The connection is initiated with Connect Request packet. Find the Connect Request packet! Click on it, and in the Event Details (the parsed version of the packet) find the Connection Interval parameter. You can see the connection interval in ms. Compare this value with the transaction durations in the Transactions window. The transaction interval has to be slightly more than the connection interval (packet transmission time is added to the interval).

c. The thermometer example sends indications every second to report the temperature continuously. Find a Handle Value Indication packet. Find the Temperature Measurement Value, compare it to the measured temperature on your smart phone.

12. Filter the packets

Network Analyzer displays all packets by default. However, sometimes we are interested only in one type of the packets. To see through the chaos, filters can be applied. E.g. let’s filter for the packets that carries the temperature data

a. Find a Handle Value Indication packet that contains temperature data
b. In the Event detail window right click on the Method: Handle Value Indication line
c. Click Add to filter. This will add an expression to the filter conditions at the top of the window.

This expression means, that only Indication packets will be displayed.

![Filter expression example](image)

d. Click on the “show only events that match the expression” button to apply the filter.

e. To narrow down the packets to only those that contains temperature data find the **Attribute Handle** line in the Event Detail window. This shows the handle of the Temperature measurement characteristic.

f. Right click on the Attribute Handle

g. Click Add to filter, the Events view will be updated – although in our case all Indication transfer temperature data