Developer Study Guide: An introduction to Bluetooth Mesh Networking

Orientation Guide

Release   :   2.0.0

Document Version :   2.0.0

Last updated   :   16th December 2019

Contents

Revision History

Version	Date	Author	Changes
1.0.0	15th June 2018	Martin Woolley Bluetooth SIG	Initial version
1.0.4	14th December 2018	Martin Woolley Bluetooth SIG	Minor errata: #11 - Exercising the generic level state would work visibly even if the onoff state was OFF. This was not correct. Imagine a dimmer control which when pressed acts as an on/off switch. Rotating the knob will have no effect if the lights are switched off. That's how the generic onoff server and generic level server should work when incorporated together in a device. The two states are now handled completely independently. Code and documentation adjusted accordingly. #12 - Light node should have subscribed to the group address once for each model. A bug in Zephyr 1.12 allowed a subscription to only one model to be sufficient for all models to have messages published to that address routed to them so there was no user discernible impact of this issue at Zephyr 1.12. Code has been adjusted.
2.0.0	16th December 2019	Martin Woolley Bluetooth SIG	The study guide no longer uses BBC micro:bits and instead is based upon boards which have more memory, allowing Bluetooth mesh features such as provisioning and configuration to be properly explored. Use cases the labs are built around are now on/off control and colour selection. There are now the following three coding labs; (1) generic on/off client, (2) generic on/off server, (3) light HSL model

Overview

Bluetooth Mesh

Bluetooth® Mesh is a Bluetooth networking technology whose specification was released in the summer of 2017. Mesh networks allow a many to many communication topology to be created between potentially very large numbers of devices. To communicate, devices send messages and the devices to which those messages are addressed do not have to be within direct radio range. Bluetooth mesh employs a strategy for propogating messages across the network which can involve messages being relayed from one device to another, in a series of *hops* until they reach their final destination, a technique known as *multi-hop delivery*. Furthermore, copies of messages get sent via different paths through the network (known as *multi-path delivery*), which makes a Bluetooth mesh network very reliable.

You'll learn more about the theory and practice of using certain aspects of Bluetooth mesh technology in this self-study resource.

About the Bluetooth Mesh Developer Study Guide

There are probably three types of Bluetooth mesh developer:

> **1. Mesh Stack Developer** - develops a full Bluetooth mesh stack, probably as part of an SDK.

> **2. Mesh Product Developer** - an embedded software engineer who develops the firmware for Bluetooth mesh products such as lights, switches and so on.

> **3. Mesh Application Developer** - develops applications for smartphones, tablets and desktop computers which allow the monitoring or control of mesh networks and their devices.

This resource is aimed at embedded software engineers who intend to become Mesh Product Developers (2). Mesh application developers (3) should download and work through the Mesh Proxy Kit from https://www.bluetooth.com/develop-with-bluetooth/build/developer-kits

Start by watching the overview video, "BMDSG overview.mp4" which is in the video/ folder.

Then, proceed by reading the basic theory document which will explain the key technical concepts of Bluetooth mesh that you'll need to know before proceeding to the hands-on coding labs.

In the hands-on coding labs, you'll implement a series of different Bluetooth mesh use cases and get a chance to both verify your understanding of the concepts and experience one way in which these concepts manifest themselves in a real SDK.

We provide full source code solutions for each of the coding labs so if you get stuck, you can refer to the solution for help.

Recommended Study Sequence

This study resource is modular in design and we recommend that you follow this sequence:

```
┌─────────────────────────────┐
│     1. Orientation Guide     │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│      2. Overview Video       │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│       3. Basic Theory        │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│           Coding             │
└─────────────────────────────┘
              │
              ▼
    ┌───────────────────────────────────┐
    │ 4. Coding Exercises Introduction   │
    └───────────────────────────────────┘
    ┌───────────────────────────────────┐
    │ 5. Coding – On/Off Client          │
    └───────────────────────────────────┘
    ┌───────────────────────────────────┐
    │ 6. Coding – On/Off Server          │
    └───────────────────────────────────┘
    ┌───────────────────────────────────┐
    │ 7. Coding – Light HSL Models       │
    └───────────────────────────────────┘
```
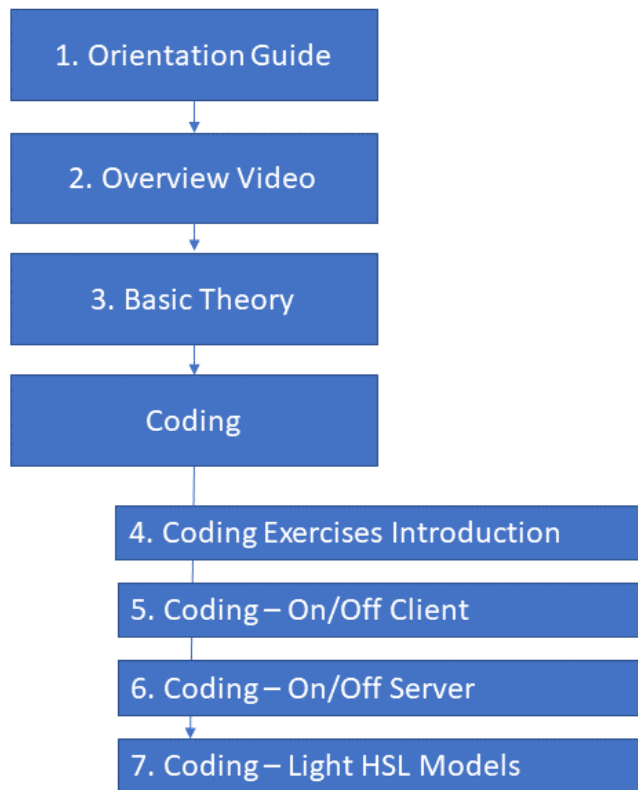
All documents including this one are in the docs folder of the kit. You'll find initial skeleton code and complete solutions in the code/ folder.

Good luck!