# Bluetooth 5 multi link hands on

This tutorial is a modification of

https://github.com/NordicPlayground/nrf52-ble-multi-link-multi-role

to make it suitable for the learning goal of the WSAN track.

### Overview

The scope of this hands on exercise is to get some experience with the multi link feature of the Nordic SoftDevices, as well as experiment with the different physical radio modes introduced by Bluetooth 5.

The example is based around an aggregator node, running on an nRF52DK. The aggregator node, so-called cluster head node, is set up to connect to any device in the area that supports either the Thingy:52 services. By using the 20 link feature of the S132 the aggregator can connect to up to 20 devices at the same time, where 19 of these devices will be Thingy devices, and 1 device can be a smart phone.

The aggregator will collect information on all the connected devices, and relay the information to the UART and to the connected smart phone over BLE. From the phone application the user can control the LED color and state of the connected devices, and get an overview over the following details from the connected devices:

- Advertising name

- Button state

- LED state

- RSSI

System Requirements:

To attend the tutorial as well as to do the practical assignments, each group will need at least a PC running either Windows, Linux, or Mac OS-X which have the following applications installed:

- Segger Embedded Studio (https://www.segger.com/products/development-tools/embedded-studio/ )

- The latest version of the [J-Link Software and Documentation pack] (https://www.segger.com/downloads/jlink# )

- nRF5\_SDK\_v15.2.0 (http://developer.nordicsemi.com/nRF5\_SDK/nRF5\_SDK\_v15.x.x/nRF5\_SDK\_15.2.0\_9412b96.zip)

- nRF Connect for desktop ( https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-desktop )

- A UART terminal such as TeraTerminal (Windows) or CoolTerm(MAC)

Each group will need at least an Android smartphone which have the following apps installed.

- nRF Connect for mobile

- Nordic Thingy

- the provided .apk

## Preparations

1) The \*ble\_app\_multi\_link\_multi\_role\* folder will have to be copied into your local SDK folder in order for the examples to compile.

Create a folder under \*\nRF5\_SDK\_15.2.0\examples\\\* called \*training\* and copy the ble\_app\_multi\_link\_multi\_role folder into it. When you are finished it should look like this:

2) One person in each group needs to set up the aggregator. Ask the Nordic attendants for help if you are unsure who should set up the aggregator.

\* Open the aggregator Segger project file: \*\nRF5\_SDK\_15.2.0\examples\training\ble\_app\_multi\_link\_multi\_role\ble\_aggregator\pca10056\s140\ses\ble\_aggregator\_pca10056\_s140.emProject\*

\* Build the project and download the code to an nRF52 PDK. In case it fails to download the compiled code with Segger, try nRF Connect – Programmer.

\* Connect to the aggregator board either over UART (baudrate 460800) or from the Android application

\* To use the Android application, install the Android apk located in \ble\_app\_multi\_link\_multi\_role\android\_apk

Once the code is running, open a serial terminal to verify that your board connects to the central device, and that you can see your name in the list.

```C

---- Device list overview (1 device connected) ----

ID Name Btn LED Phy RSSI

2 Thingy. 0 0 1Mbps -15

```

3) Press button 1 on your kit, and verify that the value is updated on the central device.

```C

---- Device list overview (1 device connected) ----

ID Name Btn LED Phy RSSI

2 Thingy 1 0 1Mbps -15

```

4) Press the button on a Thingy device, and verify that if the its state on the Android phone changes.

5) Manually control LED on Thingy devices via the apk app on the Android phone.

## Task 1 – Change advertising name of aggregator, defined in main.c, to something unique so that only Thingy devices of your group will be able to connect to your aggregator.

## Task 2 – Group Thingy devices of your group will be able to connect to your aggregator.

1) Change Thingy target name, defined in main.c, to something unique, so that only Thingy devices of your group will be able to connect to your aggregator.

2) Change Thingy devices’ name using Thingy app to something unique, so that only Thingy devices of your group will be able to connect to your aggregator.

3) Find and correct a bug so that the aggregator can filter out the Thingy devices which are not yours.

A hint to make your own unique name.

For example, if your group prefix is \*'GRP1:'\* and your name is \*'John'\* the advertising name of a Thingy should be \*'GRP1:John'\*.

WARNING: Don't make the name longer than 25 characters, or it won't fit in the advertise packet.

## Task 3 - Change the output power of your board and see how it affects the range

1) Inside \*gap\_params\_init()\* in main.c, change the TX output power of your aggregator.

2) Try different values and see how they affect the maximum range that you can get between the aggregator and the smartphone

How far can you move away from the aggregator device when the TX power is set to -40?

What about -20, 0 or 8?

Plot a graph: approximate max distance vs. TX power