Lab 2: BLE Scanning

Introduction

The purpose of today's lab is to enable you to scan and explore BLE communication. We'll take our first steps loading code onto the nRF52840DK and thinking about the Nordic software ecosystem as well.

Goals

- Set up nRF Connection on your system
- Enable BLE Scanning with the nRF52840DK and Wireshark
- Explore BLE advertisements in the nearby environment

Equipment

- Computer
- nRF52840DK + USB cable
- Smartphone (optional)

Partners

- This lab should be done individually

Submission

- Write your answers up and submit a PDF to Gradescope.

  Remember: I'm not looking for a formal lab report. Just your answers in any format that makes sense. The goal is to prove that you did the lab and spent some time thinking about it.

1. Install nRF Connect for Desktop

Nordic has a suite of *really* nice software tools that help support experimentation with their hardware platforms.The app will work on Windows, MacOS, and Linux. and it uses your nRF52840DK hardware to actually interact with devices.

Not everything in the nRF Connect panel is supported by the nrf52840DK (and some things that look like they wouldn't be supported, are; e.g. the "RSSI Viewer" works fine, despite saying it's for the nRF52832).

Download and install the nRF Connect for Desktop tools:
https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-desktop

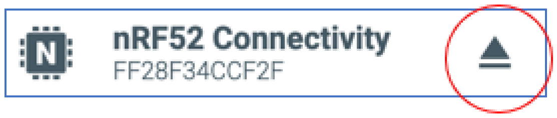
While that's installing, you can optionally install the nRF Connect app on your phone too (it's just called 'nrf Connect for Mobile' probably easier to search, but here are links nonetheless):

- https://apps.apple.com/us/app/nrf-connect-for-mobile/id1054362403
- https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp

By default, the app is just an empty shell that can install sub-apps. Go ahead and install the *Bluetooth Low Energy* app, the *Programmer*, and the *RSSI Viewer*.

The *Bluetooth Low Energy* app functions very similarly to the nRF Connect app on your phone. Both allow you to scan for nearby devices, connect to them, and investigate services they provide. Play around for a bit and see what's nearby. You might be surprised by what you find.

When you finish using an app, be sure to disconnect from it:



TASK: None. Continue to the next section.

2. Integrate BLE Scanning into Wireshark

Next, we're going to add an external capture source to Wireshark that allows it to sniff BLE communication by using the nRF52840DK. The full guide that we're following is here: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_sniffer_ble%2FUG%2Fsniffer_ble%2Finstalling_sniffer.html

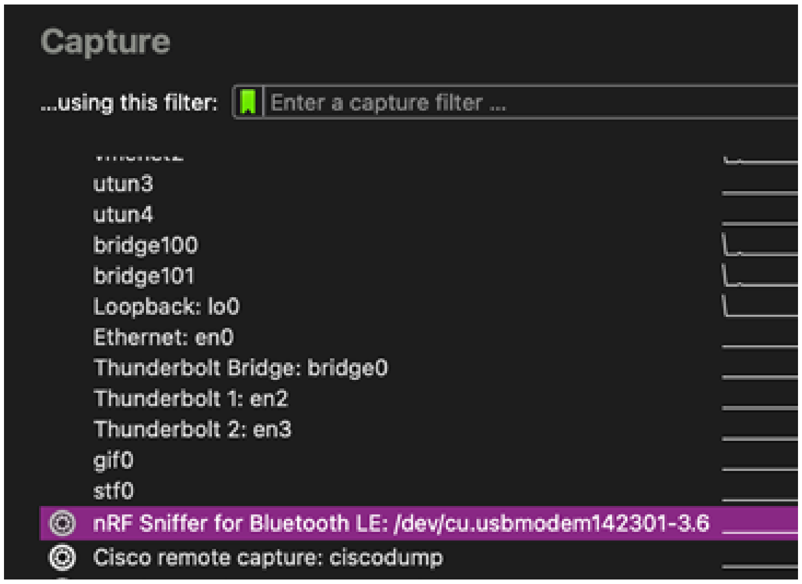
1. Get a copy for the sniffer ZIP: https://www.nordicsemi.com/Products/Development-tools/nrf-sniffer-for-bluetooth-le/download

2. Open the *Programmer* app, and drag the `/hex/sniffer_nrf52840dk_nrf52840_4.1.0.hex` precompiled firmware over for programming. Then write that firmware to your nRF52840DK.

3. The sniffer receiver is written in Python. You'll need Python3 and `pyserial >= 3.5`. If you don't have Python3, follow the python install guide. For pyserial, you can run `python3 -m pip install pyserial` once Python is installed. This *does* work on Windows with a little bit of effort.

4. We need to copy over the "extcap" stuff to the correct folder so Wireshark can find it. I can't write better instructions than Nordic already did: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_sniffer_ble%2FUG%2Fsniffer_ble%2Finstalling_sniffer.html (Note: we've already handled the python requirements from step 1 by installing `pyserial`)

5. Finally, make sure you have your nRF52840DK reprogrammed and connected over USB, then either restart Wireshark or go to "Capture Menu -> Refresh Interfaces". You should now see a new capture interface: "nRF Sniffer for Bluetooth LE".



utun3
utun4
bridge100
bridge101
Loopback: lo0
Ethernet: en0
Thunderbolt Bridge: bridge0
Thunderbolt 1: en2
Thunderbolt 2: en3
gif0
stf0
nRF Sniffer for Bluetooth LE: /dev/cu.usbmodem142301-3.6
Cisco remote capture: ciscodump

Double-click it to start capturing!

Lots of things can go wrong here! Be sure that you're following all the steps and didn't skip anything. Also check the troubleshooting steps here:
https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_sniffer_ble%2FUG%2Fsniffer_ble%2Ftroubleshooting.html&cp=10_5_6

If you're still having problems, definitely reach out and I'm happy to help!

TASK: None. Continue to the next section.

3. Investigating BLE Advertisements

Now that you've (hopefully) got the Wireshark external capture working, let's investigate some BLE packets! Run wireshark and collect packets for a few seconds. Then take a look at the packets you received and answer a few questions. Include screenshots as makes sense.

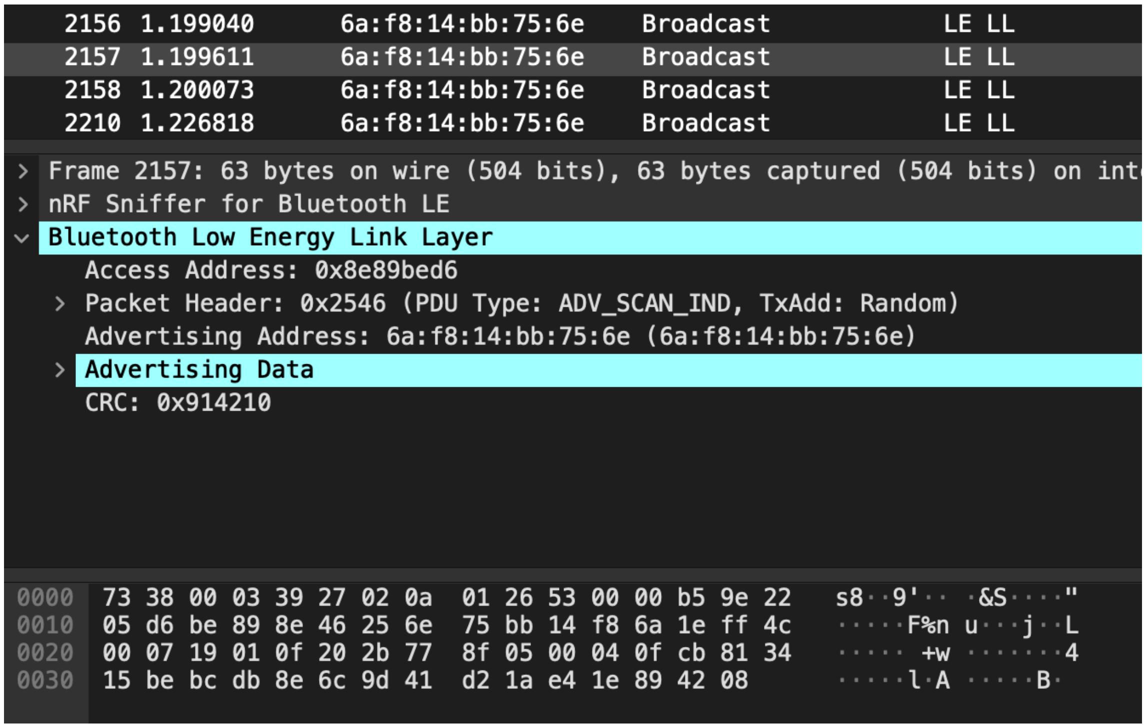
1. **TASK:** How many transmissions do you see in one second?

   Note: if you don't see many devices around first HOW?! and secondly, try again on campus. I was literally collecting *thousands* of packets from my office.

2. **TASK:** Pick a received packet and explain the meaning of all of the bytes of it.

   Note: you can ignore the bytes that are part of the "nRF Sniffer for Bluetooth LE". That appends extra bytes to the start with metadata.

   The real data should be 47 bytes or less and will be highlighted when you select "Bluetooth Low Energy Link Layer" in Wireshark. Clicking different parts within this will highlight the bytes that correspond to different fields.



```
  2156  1.199040      6a:f8:14:bb:75:6e    Broadcast          LE LL
  2157  1.199611      6a:f8:14:bb:75:6e    Broadcast          LE LL
  2158  1.200073      6a:f8:14:bb:75:6e    Broadcast          LE LL
  2210  1.226818      6a:f8:14:bb:75:6e    Broadcast          LE LL

> Frame 2157: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on int
> nRF Sniffer for Bluetooth LE
v Bluetooth Low Energy Link Layer
    Access Address: 0x8e89bed6
  > Packet Header: 0x2546 (PDU Type: ADV_SCAN_IND, TxAdd: Random)
    Advertising Address: 6a:f8:14:bb:75:6e (6a:f8:14:bb:75:6e)
  > Advertising Data
    CRC: 0x914210

0000  73 38 00 03 39 27 02 0a  01 26 53 00 00 b5 9e 22   s8··9'··  ·&S····"
0010  05 d6 be 89 8e 46 25 6e  75 bb 14 f8 6a 1e ff 4c   ·····F%n u···j··L
0020  00 07 19 01 0f 20 2b 77  8f 05 00 04 0f cb 81 34   ····· +w ·······4
0030  15 be bc db 8e 6c 9d 41  d2 1a e4 1e 89 42 08      ·····l·A ·····B·
```

3. **TASK:** Identify five different AD Types across various advertisements.

   Remember that the data within an advertisement is split into a series of “Length, Type, Value” fields concatenated together. The goal is to look at some packets and determine five different “Type”s that occur.

4. **TASK:** Identify fields in an advertisement and corresponding scan response for one device.

   It might take a couple of tries to find scan responses with payloads. If you’re having a hard time, doing a scan from your phone with the nRF Connect app should result in a lot of scan requests being sent in the environment.

   Find one device that sends both an advertisement and a scan response and figure out which fields it puts in each respectively.