

nRF Sniffer for Bluetooth LE

v4.0.0

User Guide

Contents

Revision history	iii
1 Introduction	5
2 Installing the nRF Sniffer	6
2.1 Programming the nRF Sniffer firmware	6
2.2 Installing the nRF Sniffer capture tool	7
2.3 Adding a Wireshark profile for the nRF Sniffer	9
3 Running the nRF Sniffer	12
4 nRF Sniffer usage	15
4.1 Capturing from multiple hardware interfaces	17
4.2 Inspecting captured data	18
5 Common sniffing actions	22
5.1 Sniffing advertisements from all nearby devices	22
5.2 Sniffing advertisement packets involving a single Peripheral	22
5.3 Sniffing a connection involving a single Peripheral	22
5.4 Sniffing the pairing procedure of a connection	22
5.5 Sniffing a connection between bonded devices	23
6 Troubleshooting	24
Glossary	26
Acronyms and abbreviations	27
Legal notices	28

Revision history

Date	Description
2021-08-13	<ul style="list-style-type: none">• Updated to match nRF Sniffer for <i>Bluetooth</i>[®] LE v4.0.0• Added information about input keys and usage to nRF Sniffer usage on page 15 and Sniffing the pairing procedure of a connection on page 22• Added information about interface options to nRF Sniffer usage on page 15• Added Sniffing a connection between bonded devices on page 23• Updated screenshots• Added a glossary
November 2020	<ul style="list-style-type: none">• Updated Supported devices• Updated Programming the nRF Sniffer firmware on page 6• Editorial changes
January 2020	Corrected Python requirements
December 2019	<ul style="list-style-type: none">• Editorial changes to all sections• Updated to match nRF Sniffer for Bluetooth LE v3.0.0
September 2018	Updated content: <ul style="list-style-type: none">• Required software• Setting up the nRF Sniffer• Sniffer commands• Troubleshooting
January 2018	Updated content: <ul style="list-style-type: none">• Required software• Setting up the nRF Sniffer
November 2017	<ul style="list-style-type: none">• nRF Sniffer updated to work more closely with Wireshark• Updated software to support the nRF52 DK
April 2017	Updated content: <ul style="list-style-type: none">• Removed reference to nRF52 Series in Required hardware• Required software• Setting up the nRF Sniffer
March 2017	Updated content: <ul style="list-style-type: none">• Required hardware• Required software• Setting up the nRF Sniffer
July 2014	Updated content: <ul style="list-style-type: none">• Required hardware• Required software• Setting up the nRF Sniffer

Date	Description
	<ul style="list-style-type: none">• Running the Sniffer• Using the Sniffer• Using Wireshark• Wireshark Tips• Troubleshooting
April 2014	Updated firmware, now supports all versions of PCA10000 and PCA10001
December 2013	First release

Previous versions

PDF files for relevant previous versions are available here:

- [nRF Sniffer User Guide v3.2](#) (corresponds to nRF Sniffer v3.1.0)
- [nRF Sniffer User Guide v3.1](#) (corresponds to nRF Sniffer v3.0.0)
- [nRF Sniffer User Guide v2.2](#) (corresponds to nRF Sniffer v2.x)

1 Introduction

The nRF Sniffer for Bluetooth LE is a useful tool for learning about and debugging Bluetooth Low Energy (LE) applications. It provides a near real-time display of Bluetooth packets that are sent between a selected Bluetooth Low Energy device and the device it is communicating with, even when the connection is encrypted.

When developing a Bluetooth Low Energy product, knowing what happens over-the-air between devices can help you identify and fix issues quickly.

On startup, the nRF Sniffer lists all nearby Bluetooth Low Energy devices that are advertising, providing the Bluetooth address and address type, complete or shortened name, and *Received Signal Strength Indication (RSSI)*.

Supported development kits and dongles

- nRF52840 DK (PCA10056)
- nRF52840 Dongle (PCA10059)
- nRF52 DK (PCA10040)
- nRF51 DK (PCA10028)
- nRF51 Dongle (PCA10031)

Supported operating systems

- Windows 10
- 64-bit OS X/macOS 10.6 or later
- Linux (check the *Wireshark* prerequisites for version compatibility)

2 Installing the nRF Sniffer

The nRF Sniffer for Bluetooth LE software consists of firmware that is programmed onto a *Development Kit (DK)* or dongle and a capture plugin for [Wireshark](#) that records and analyzes the detected data.

Before you start setting up the nRF Sniffer, make sure that you have the following prerequisites installed on your computer:

- [Wireshark](#) v3.4.1 or later. *Wireshark* is a free software tool that captures wireless traffic and reproduces it in a readable format.
- [Python](#) v3.6 or later.

Download [nRF Sniffer for Bluetooth LE](#) v4.x or later and extract the archive into a folder of your choice. In the following sections, this folder is referred to as *Sniffer_Software*.

Then program the firmware to the *DK* or dongle, install the nRF Sniffer capture tool, and add a *Wireshark* profile for the nRF Sniffer as described in the following sections.

2.1 Programming the nRF Sniffer firmware

You must connect a *DK* or dongle running the nRF Sniffer firmware to your computer to be able to use the nRF Sniffer for Bluetooth LE.

See [Supported development kits and dongles](#) for a list of development kits and dongles that can run the nRF Sniffer firmware.

There are various ways to program the nRF Sniffer firmware. The following instructions use [nRF Connect Programmer](#), but you can also use the command-line tool `nrfjprog` (which is part of the [nRF Command Line Tools](#)).

To program your *DK* or dongle, complete the following steps:

1. Install nRF Connect Programmer.
See [Installing the Programmer](#) for instructions.
2. On macOS and Linux, install the SEGGER J-Link software.
It is available from [SEGGER J-Link Software](#).

Note: On Windows, the J-Link software is included in nRF Connect for Desktop, so you can skip this step.

3. Locate the firmware HEX file for your *DK* or dongle.

All firmware HEX files are located in *Sniffer_Software/hex/*. Use the suitable file for your *DK* or dongle:

Development kit/dongle	Firmware file name
nRF52840 DK (PCA10056)	sniffer_nrf52840dk_nrf52840_*.hex
nRF52840 Dongle (PCA10059)	sniffer_nrf52840dongle_nrf52840_*.hex
nRF52 DK (PCA10040)	sniffer_nrf52dk_nrf52832_*.hex
nRF51 DK (PCA10028)	sniffer_nrf51dk_nrf51422_*.hex
nRF51 DK (PCA10031)	sniffer_nrf51dongle_nrf51422_*.hex

Table 1: Firmware file names

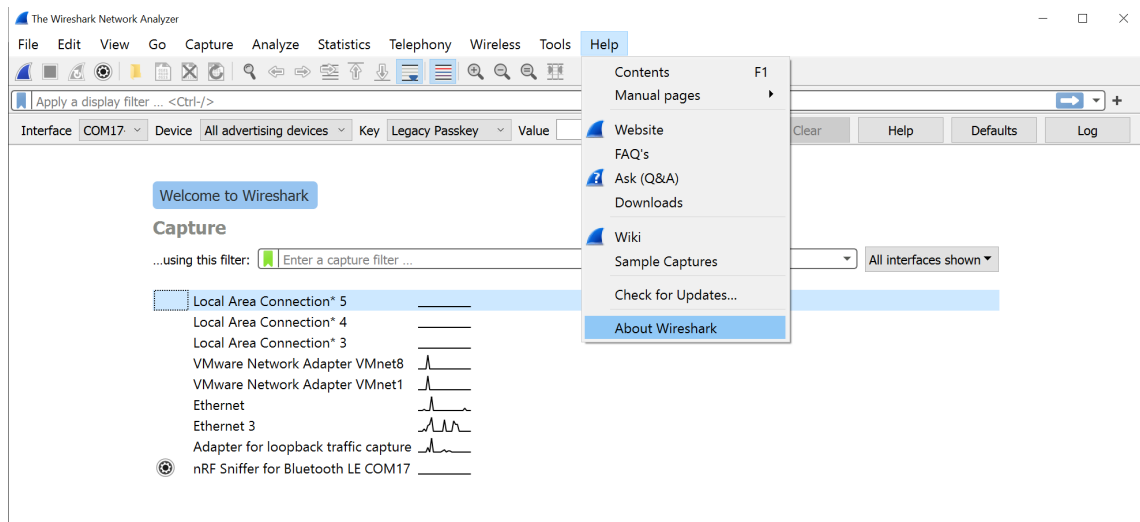
- Follow the instructions in [Programming a Development Kit or the nRF51 Dongle](#) or [Programming the nRF52840 Dongle](#) to program the HEX file.

2.2 Installing the nRF Sniffer capture tool

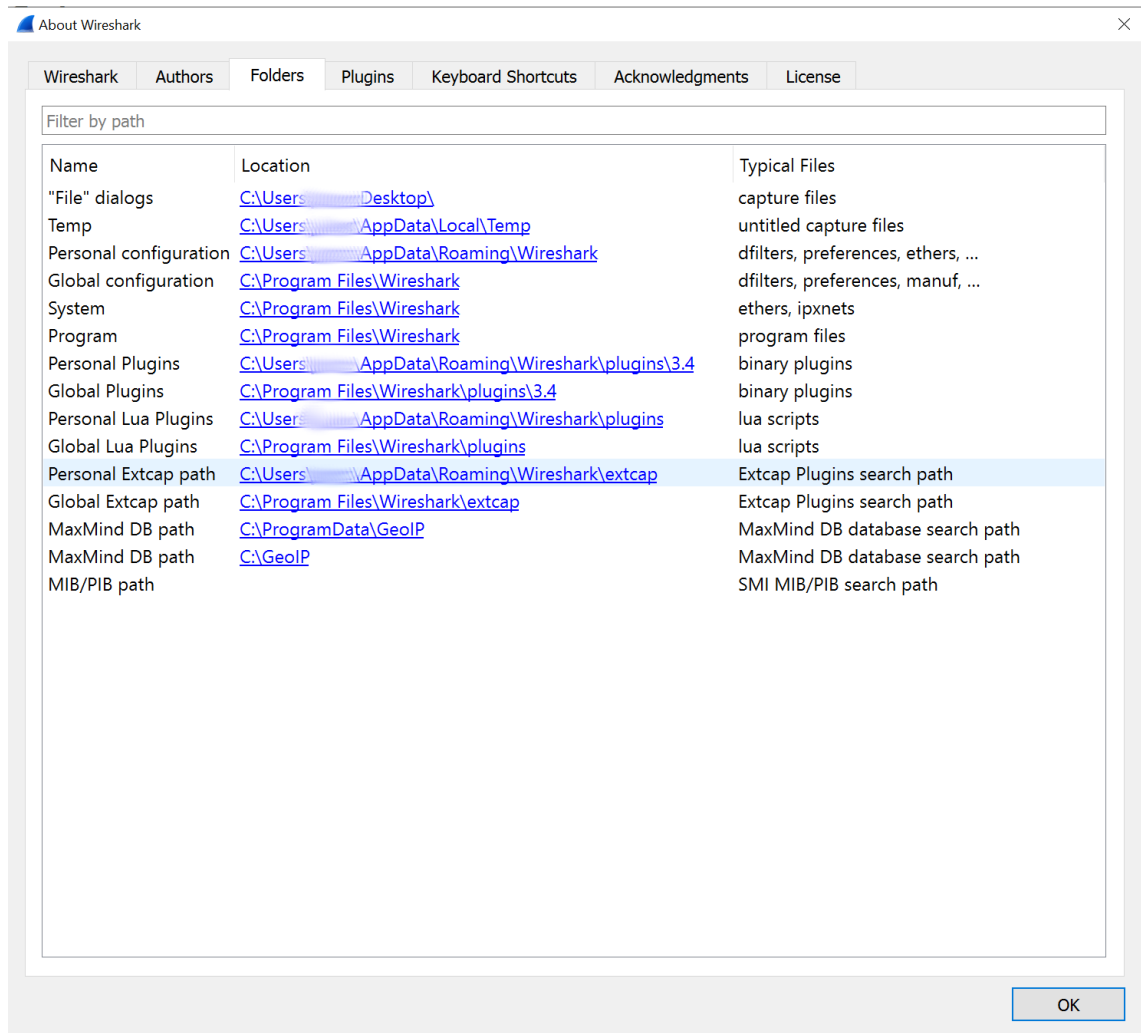
The nRF Sniffer for Bluetooth LE software is installed as an external capture plugin in *Wireshark*.

To install the nRF Sniffer capture tool, complete the following steps:

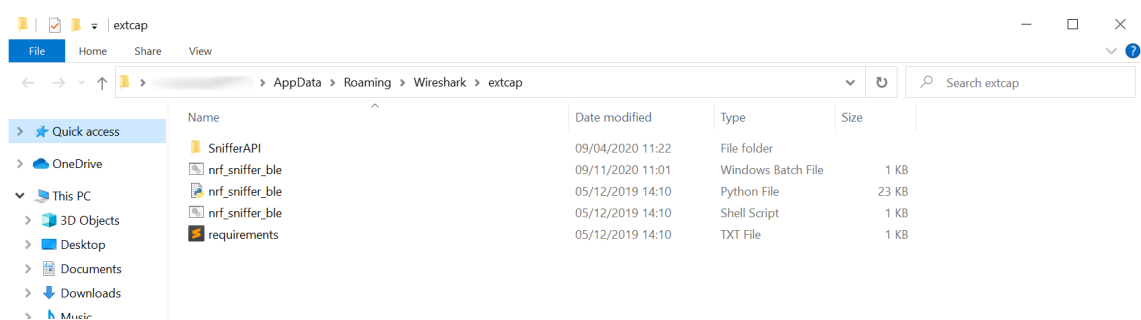
- Install the Python requirements:
 - Open a command window in the *Sniffer_Software/extcap/* folder.
 - Type `pip3 install -r requirements.txt` to install the requirements.
 - Close the command window.
- Copy the nRF Sniffer capture tool into *Wireshark's* folder for personal external capture plugins:
 - Open *Wireshark*.
 - Go to **Help > About Wireshark** (on Windows or Linux) or **Wireshark > About Wireshark** (on macOS).



- Select the **Folders** tab.
- Double-click the location for the **Personal Extcap path** to open this folder.



e) Copy the contents of the *Sniffer_Software/extcap/* folder into this folder.



3. Make sure that the nRF Sniffer files can be run correctly:

- Open a command window in *Wireshark's* folder for personal external capture plugins.
- Run the nRF Sniffer tool to list available interfaces.

On Windows, type `nrf_sniffer_ble.bat --extcap-interfaces`. On macOS or Linux, type `nrf_sniffer_ble.sh --extcap-interfaces`.

You should see a series of strings, similar to what is shown in the following screenshot.


```

C:\Users\... \AppData\Roaming\Wireshark\extcap>nrf_sniffer_ble.bat --extcap-interfaces
extcap {version=3.0.0-g96df3b6ea9e}{display=nRF Sniffer for Bluetooth LE}{help=https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF-Sniffer-for-Bluetooth-LE}
interface {value=COM17-None}{display=nRF Sniffer for Bluetooth LE COM17}
control {number=0}{type=selector}{display=Device}{tooltip=Device list}
control {number=1}{type=selector}{display=Key}{tooltip=}
control {number=2}{type=string}{display=Value}{tooltip=6 digit passkey or 16 or 32 bytes encryption key in hexadecimal starting with '0x', big endian format.If the entered key is shorter than 16 or 32 bytes, it will be zero-padded in front'}{validation="^([0-9]{6})|([0x0-9a-fa-f]{1,64})|([0-9a-fa-f]{2}[:-:]{5}){([0-9a-fa-f]{2}) (public|random)}$"}
control {number=3}{type=string}{display=Adv Hop}{default=37,38,39}{tooltip=Advertising channel hop sequence. Change the order in which the sniffer switches advertising channels. Valid channels are 37, 38 and 39 separated by comma.}{validation="^([0-9]{2},)*([0-9]{2})$"}{required=true}
control {number=7}{type=button}{display=Clear}{tooltip=Clear or remove device from Device list}
control {number=4}{type=button}{role=help}{display=Help}{tooltip=Access user guide (launches browser)}
control {number=5}{type=button}{role=restore}{display=Defaults}{tooltip=Resets the user interface and clears the log file}
control {number=6}{type=button}{role=logger}{display=Log}{tooltip=Log per interface}
value {control=0}{value=} {display=All advertising devices}{default=true}
value {control=1}{value=0}{display=Legacy Passkey}{default=yes}
value {control=1}{value=1}{display=Legacy OOB data}
value {control=1}{value=2}{display=Legacy LTK}
value {control=1}{value=3}{display=SC LTK}
value {control=1}{value=4}{display=SC Private Key}
value {control=1}{value=5}{display=LE address}
C:\Users\... \AppData\Roaming\Wireshark\extcap>

```

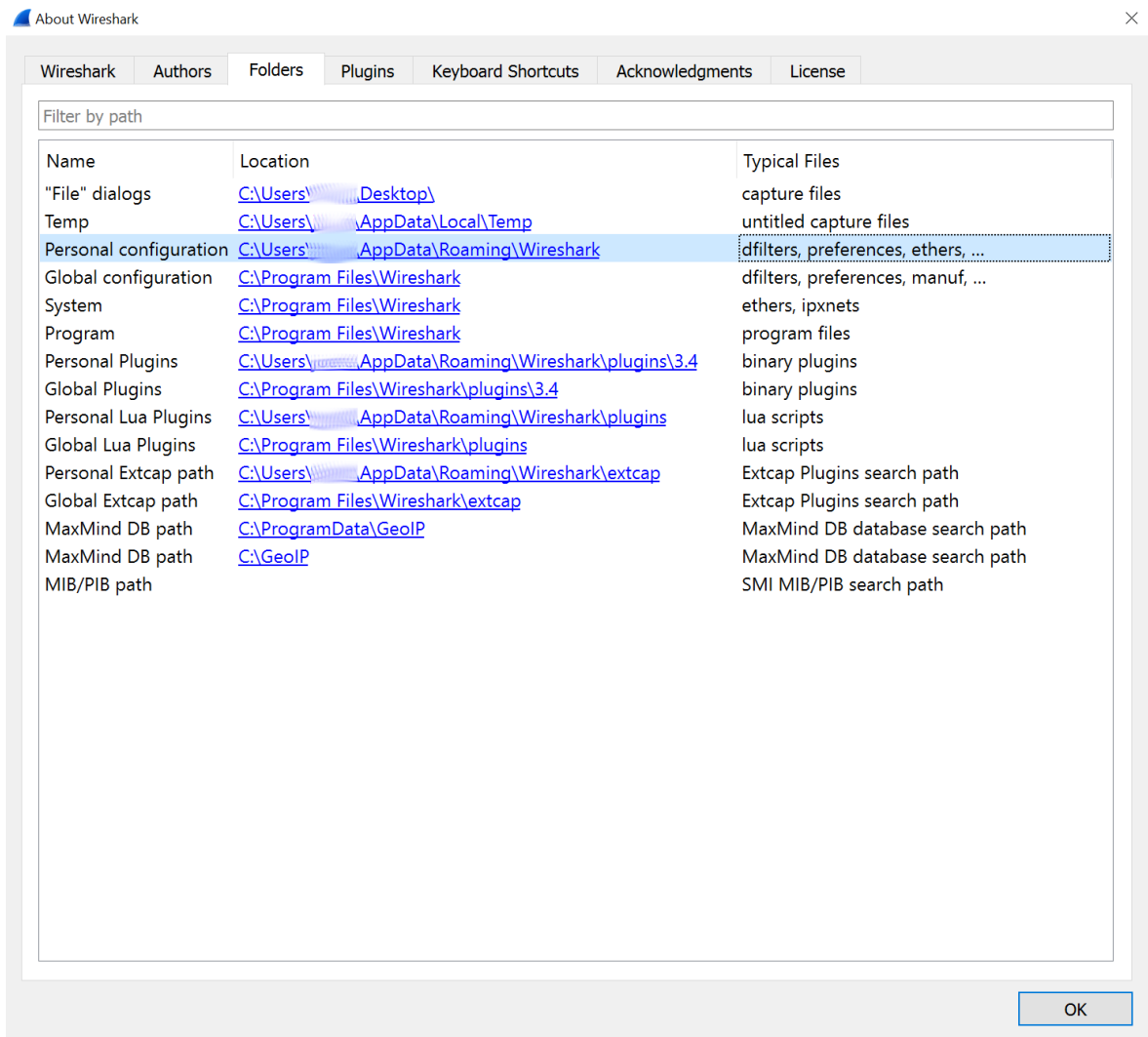
- c) If the previous step returned an error, verify that Python 3 is accessible.
On Windows, enter `python --version`. On macOS or Linux, enter `python3`. If the command cannot be found or the version is wrong, make sure that Python v3.6 or later is in your path and that it is the first Python version in the path.
 - d) For macOS or Linux: Verify that the `nrf_sniffer_ble.sh` file has the `x` permission.
If the `x` permission is missing, add it using `chmod +x nrf_sniffer_ble.sh`.
4. Enable the nRF Sniffer capture tool in *Wireshark*:
 - a) Refresh the interfaces in *Wireshark* by selecting **Capture > Refresh Interfaces** or pressing **F5**.
You should see that nRF Sniffer is displayed as one of the interfaces on the start page.
 - b) Select **View > Interface Toolbars > nRF Sniffer for Bluetooth LE** to enable the nRF Sniffer interface.

2.3 Adding a Wireshark profile for the nRF Sniffer

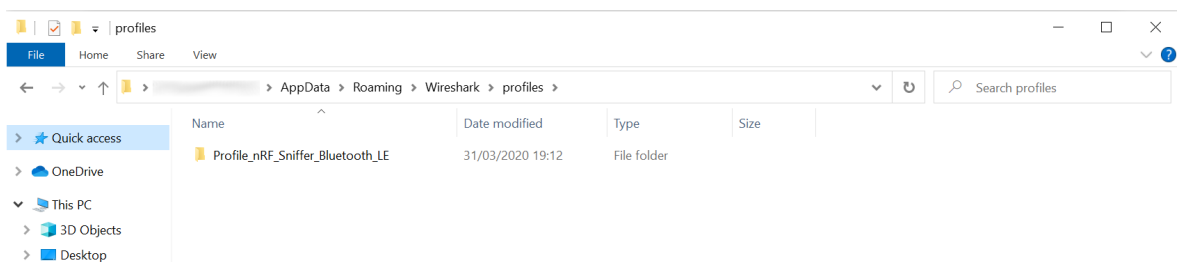
You can add a profile in *Wireshark* for displaying the data recorded by the nRF Sniffer for Bluetooth LE in a convenient way.

To add the nRF Sniffer profile in *Wireshark*, complete the following steps:

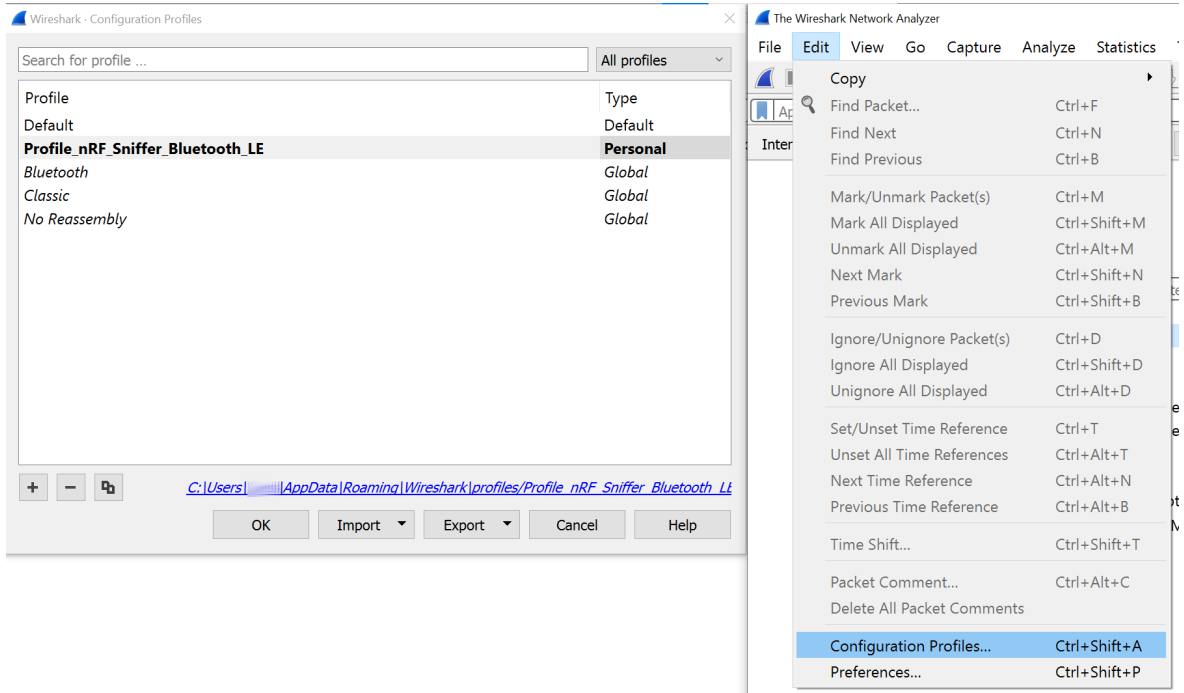
1. Go to **Help > About Wireshark** (on Windows or Linux) or **Wireshark > About Wireshark** (on macOS).
2. Select the **Folders** tab.
3. Double-click the location for the **Personal configuration** to open this folder.



- Copy the profile folder *Sniffer_Software/Profile_nRF_Sniffer_Bluetooth_LE* into the *profiles* subfolder of this folder.



- In *Wireshark*, select **Edit > Configuration Profiles**.
- Select **Profile_nRF_Sniffer_Bluetooth_LE** and click **OK**.



3 Running the nRF Sniffer

To start sniffing, place the *DK* or dongle that runs the nRF Sniffer for Bluetooth LE firmware between the two devices that are communicating. Then open *Wireshark* and start recording packets.

Connect the *DK* or dongle to your computer and turn it on. Then place it between the Central and Peripheral device that you want to sniff.

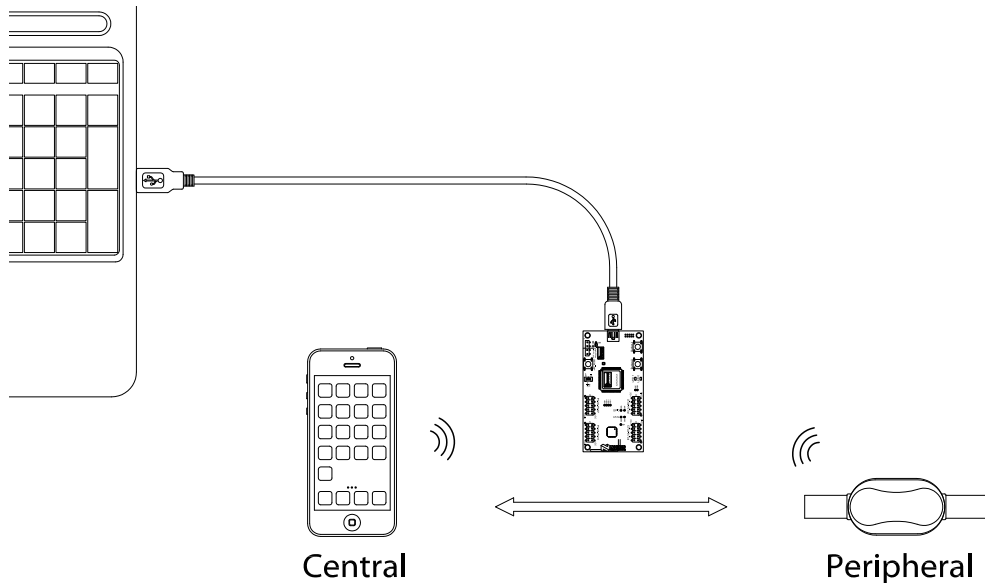


Figure 1: Hardware setup

When you open *Wireshark*, the *Wireshark* capture screen is displayed. It includes the *Wireshark* interface for managing packets that are captured, the nRF Sniffer toolbar, and the hardware interfaces connected to the nRF Sniffer.

Note: If the nRF Sniffer toolbar is not visible, select **View > Interface Toolbars > nRF Sniffer for Bluetooth LE**.

To start sniffing, double-click on the hardware interface (nRF Sniffer for Bluetooth LE COM17 in the following figure).

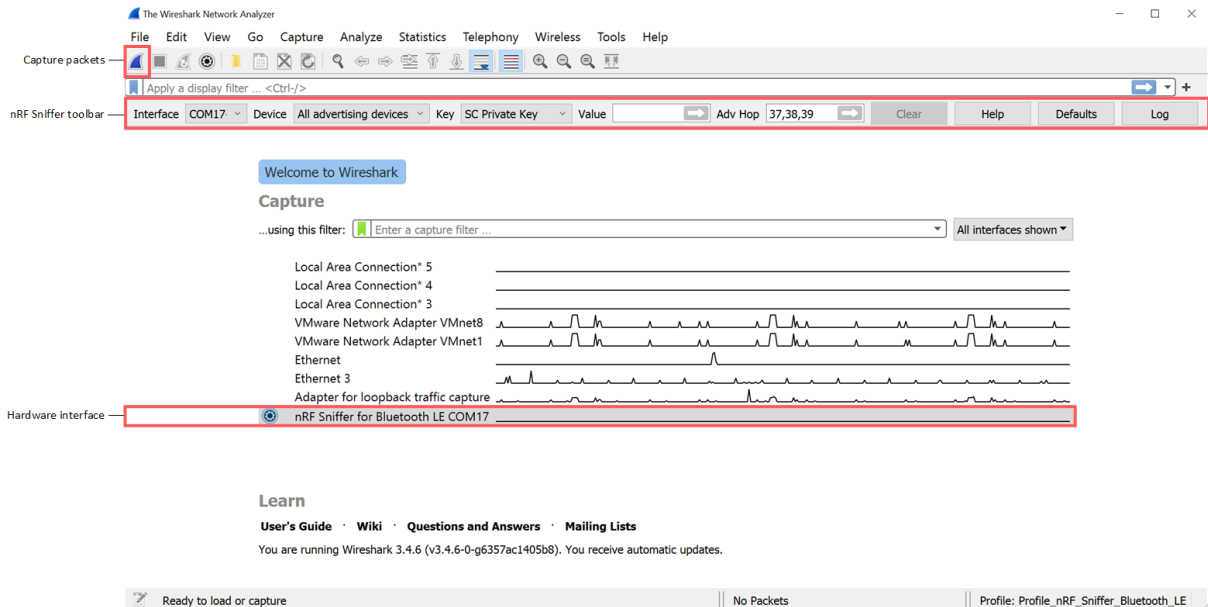


Figure 2: Wireshark capture screen

The following options are available from the capture screen in *Wireshark*:

RSSI filter

You can apply a *RSSI* filter on the packets that are being received. Only packets that match the filter are displayed.

You must set the capture filter in the capture screen by using the keyword `rssi`. For example, the filter `rssi >= -70` only captures packets that have an *RSSI* greater than or equal to -70 dBm.

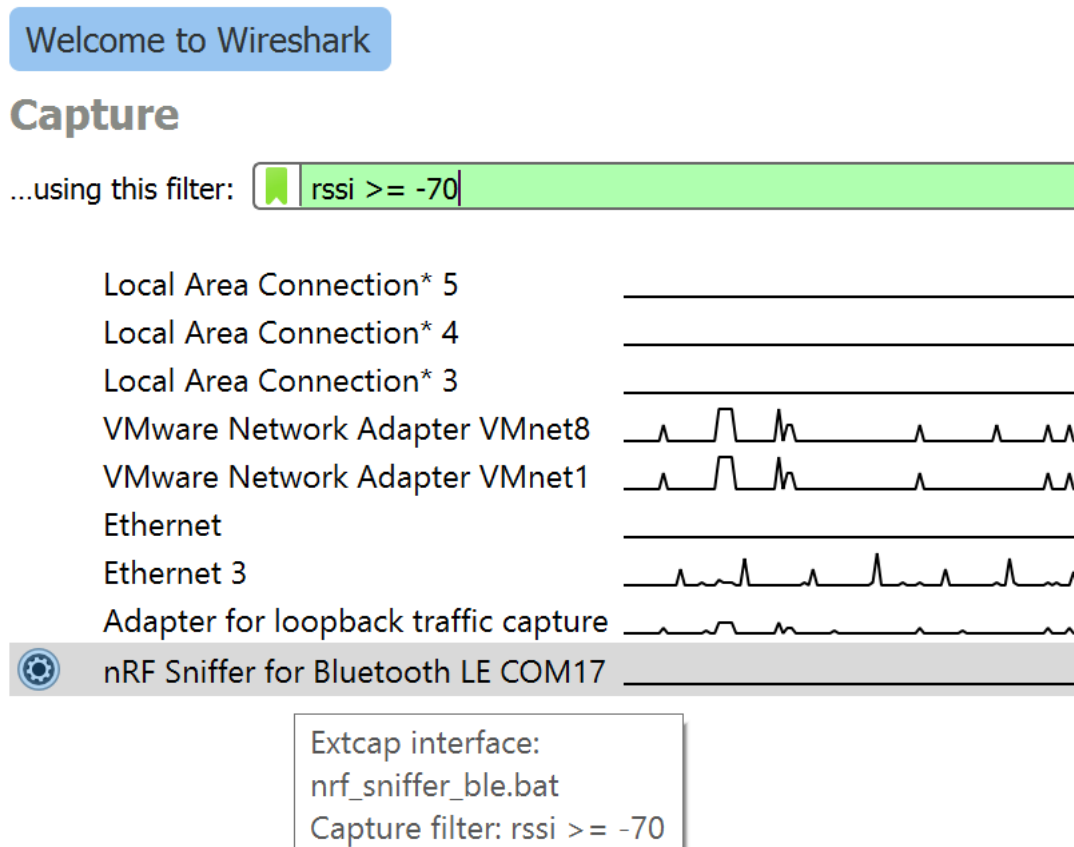


Figure 3: RSSI filter

Interface options

Click the gear icon next to the interface to configure additional options for the nRF Sniffer for Bluetooth LE.

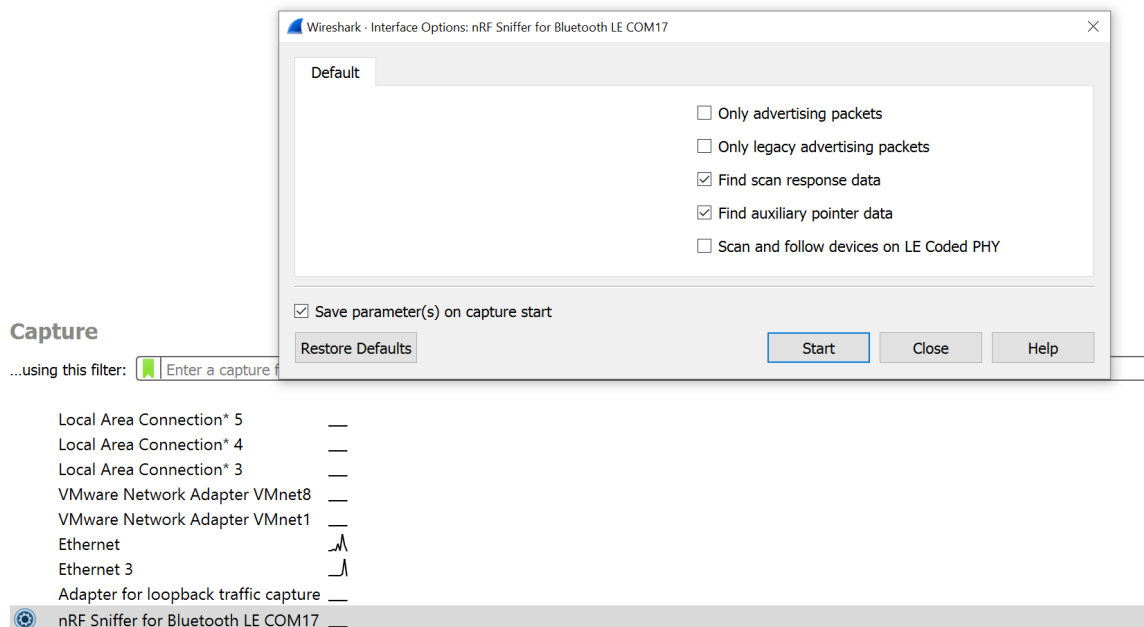


Figure 4: Interface options

The following options are available:

Only advertising packets

Sniff only the advertising packets of the given device. When a new connection is established, the nRF Sniffer ignores it.

Only legacy advertising packets

Sniff only the legacy advertising packets of the given device. The nRF Sniffer does not look for the advertiser's device address in extended advertising packets in the auxiliary advertising packets.

Find scan response data

Follow scan requests and scan responses when sniffing all advertising devices. This option is useful for finding the advertiser's name in the scan response data. You need an active scanner to generate the scan requests to follow.

Find auxiliary pointer data

Follow the auxiliary pointer for additional data when sniffing all advertising devices. This option is useful for finding the advertiser's address and name in the auxiliary advertising data.

Scan and follow devices on LE Coded PHY

Sniff on the LE Coded PHY when sniffing all advertising devices and a specific device. The nRF Sniffer follows the connection on any PHY it uses. To sniff on both LE 1M PHY and LE Coded PHY at the same time, use multiple sniffers.

4 nRF Sniffer usage

Once the nRF Sniffer for Bluetooth LE is running, it reports advertisements and lists nearby devices in the Device List. The software interface has several commands for controlling the operating mode of the nRF Sniffer.

Note: The nRF Sniffer might not pick up all connect requests and does not always pick up on a connection. In such cases, reconnect and try sniffing again. If you do not see any activity in your *Wireshark* console, see [Troubleshooting](#) on page 24.

The nRF Sniffer has two modes of operation:

1. Listen on all advertising channels to pick up as many packets as possible from as many devices as possible. This is the default mode.
2. Follow one particular device and try to catch all packets sent to or from this particular device. This mode catches all:
 - Advertisements and Scan Responses sent from the device
 - Scan Requests and Connect Requests sent to the device
 - Packets in the connection sent between the two devices in the connection

The software interface provides commands and options that control the nRF Sniffer operation.

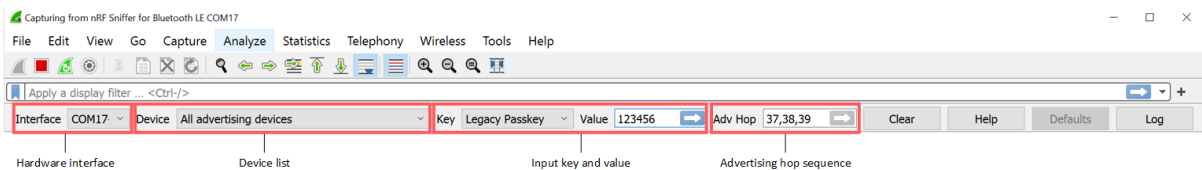


Figure 5: nRF Sniffer software interface

Hardware interface

This list shows the available hardware interfaces. If you have more than one *DK* or dongle with the nRF Sniffer firmware connected, you can choose which one to control with the toolbar. To use several hardware interfaces at the same time, see [Capturing from multiple hardware interfaces](#) on page 17.

Device list

This list shows nearby devices that are advertising. When you start sniffing, **All advertising devices** is selected. Choose a device from the list to sniff that specific device. When you select a different device while in a connection, the current connection is lost.

If the device that you want to sniff is not discovered, you can add it to the list manually. See [LE Address](#) on page 16.

Input key and value

Use this field to provide the nRF Sniffer with input information that cannot be captured from air-traffic alone. To do so, select the input key from the drop-down menu and enter the corresponding value in the input field.

The following input keys are available:

Legacy Passkey

If your device asks you to provide your passkey, type the 6-digit passkey in the passkey text field and press **Enter**. Then enter the passkey into the device.

Legacy OOB data

If your device uses a legacy pairing procedure with a 16-byte *Out of Band (OOB)* key, provide it in hexadecimal format (starting with 0x, big endian). You must do this before the device enters encryption. If the entered key is shorter than 16 bytes, it is padded with zeros in front.

Legacy LTK

If your device has an existing bond using a legacy *Long Term Key (LTK)*, provide it in hexadecimal format (starting with 0x, big endian). You must do this before the device enters encryption. If the entered key is shorter than 16 bytes, it is padded with zeros in front.

SC LTK

If your device has an existing bond using an LE Secure Connections *LTK*, provide it in hexadecimal format (starting with 0x, big endian). You must do this before the device enters encryption. If the entered key is shorter than 16 bytes, it is padded with zeros in front.

SC Private Key

If your device uses LE Secure Connections pairing and neither of the devices is in debug mode (using the Debug private key), provide the 32-byte Diffie-Hellman private key of your device in hexadecimal format (starting with 0x, big endian). You must do this before the device starts the pairing procedure. If the entered key is shorter than 32 bytes, it is padded with zeros in front.

LE Address

If the device that you want to sniff is not currently advertising and therefore was not discovered, use this field to add its LE address to the device list. Input the full 6-byte LE address, separating each byte with a colon, and append the address type ("public" or "random"). For example:

```
57:25:b0:81:eb:e5 random
```

Note: If you add a device while capturing is stopped, the device does not show up in the device list until you start capturing.

See [Sniffing the pairing procedure of a connection](#) on page 22 for more information about providing the security credentials.

Advertising hop sequence

You can change the order in which the nRF Sniffer switches advertising channels when following a device. Define the order with comma-separated channel numbers, for example, 37, 38, 39. Press **Enter** when done.

With the default configuration, the nRF Sniffer waits for a packet on channel 37. After it receives a packet on channel 37, it transitions to sniffing on channel 38. When it receives a packet on channel 38, it transitions to sniffing on channel 39. When it receives a packet on channel 39, it starts sniffing on channel 37, and repeats the operation.

Clear button

Click this button to remove all entries in the device list and start scanning for new devices. This button is active only when capturing is ongoing.

Defaults button

Click this button to remove all entries in the device list and set all configuration options to their default values. This button is active only when no capturing is ongoing.

Help button

Click this button to open the documentation.

Log button

Click this button to display the debug log and information about the nRF Sniffer version. Check this log if you encounter any issues, and include the information when reporting issues.

4.1 Capturing from multiple hardware interfaces

You can capture packets from several hardware interfaces/devices simultaneously.

Note: On Windows, this feature is available in *Wireshark* v3.0.7 and v3.2.0 and later. If you are using an older version of *Wireshark*, you must run one instance of *Wireshark* for each nRF Sniffer hardware attached to the computer. Select only one hardware interface in each of the *Wireshark* instances.

To capture from multiple hardware interfaces simultaneously, select the hardware interfaces in the capture screen and click **Start Capturing packets**.

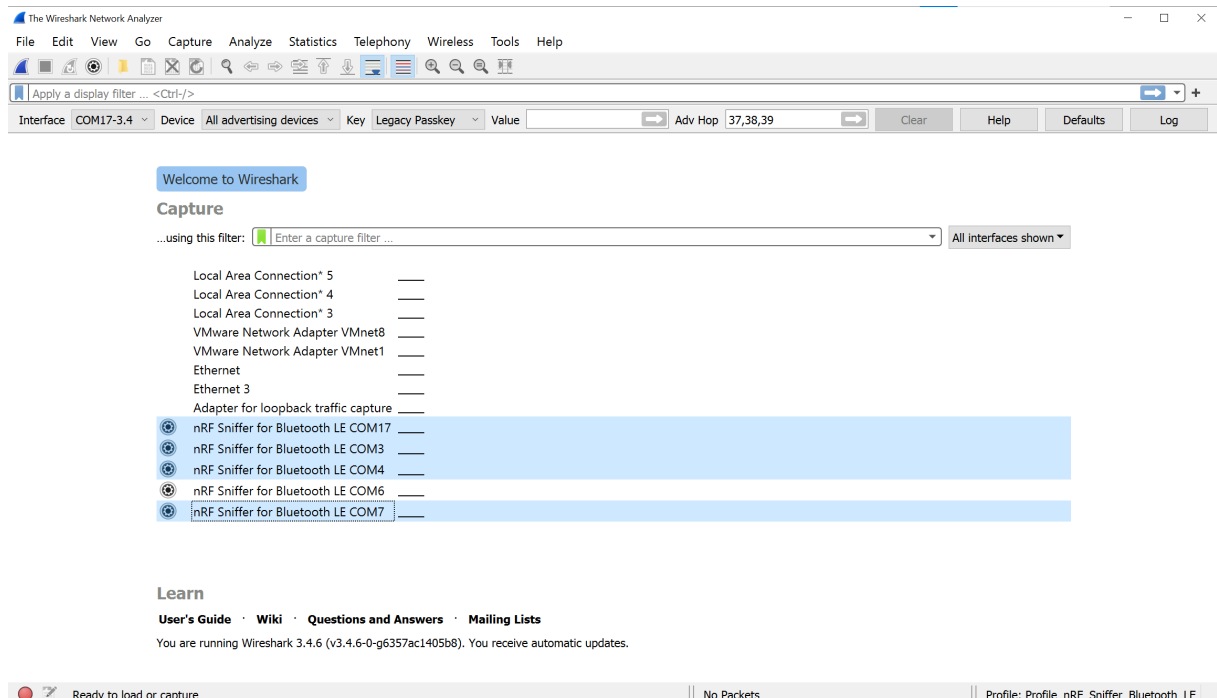


Figure 6: Select multiple hardware interfaces

The captured data contains the interface identifier used by *Wireshark* to identify the capture interface (`frame.interface_id`) and the hardware identifier for the *DK* or dongle running the nRF Sniffer firmware (`nordic_ble.board_id`).

Capturing from 5 interfaces

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Interface COM17-3.4 Device All advertising devices Key Legacy Passkey Value

No.	Source	Destination	Protocol	Length	More Data	Event counter	Board	Interface id	Info
22960	Apple_58:a0:e3	Broadcast	LE LL	30			0	7	4 ADV_IND
22961	Apple_58:a0:e3	Broadcast	LE LL	30			0	3	1 ADV_IND
22962	Apple_58:a0:e3	Broadcast	LE LL	30			0	7	4 ADV_IND
22963	3e:92:6e:68:00:03	Broadcast	LE LL	21			0	7	4 ADV_NONCONN_IND
22964	3e:82:6e:68:00:03	Broadcast	LE LL	21			0	7	4 ADV_NONCONN_IND
22965	3e:82:6e:68:00:03	Broadcast	LE LL	21			0	7	4 ADV_NONCONN_IND
22966	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	4	2 ADV_IND
22967	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	3	1 ADV_IND
22968	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	7	4 ADV_IND
22969	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	4	2 ADV_IND
22970	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	17	0 ADV_IND
22971	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	3	1 ADV_IND
22972	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	7	4 ADV_IND
22973	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	17	0 ADV_IND
22974	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	4	2 ADV_IND
22975	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	3	1 ADV_IND
22976	d9:c7:f9:71:46:c7	Broadcast	LE LL	37			0	7	4 ADV_IND
22977	SamsungE_35:9d:35	Broadcast	LE LL	34			0	4	2 ADV_NONCONN_IND
22978	SamsungE_35:9d:35	Broadcast	LE LL	34			0	17	0 ADV_NONCONN_IND
22979	SamsungE_35:9d:35	Broadcast	LE LL	34			0	3	1 ADV_NONCONN_IND
22980	SamsungE_35:9d:35	Broadcast	LE LL	34			0	7	4 ADV_NONCONN_IND

Figure 7: Data capture from multiple hardware interfaces

4.2 Inspecting captured data

All Bluetooth Low Energy packets detected by the Sniffer for Bluetooth LE are passed to *Wireshark*, where they are wrapped in a header containing useful meta-information not present in the Bluetooth Low Energy packet itself. *Wireshark* dissects the packets and separates the actual packet from the meta-information.

When you browse captured packets, select a packet in the **packet list** to show the breakdown of that packet in the **packet details pane**. The bytes of the packet are shown in the **packet bytes pane**. Click a value in the details to highlight it among the bytes, or click on the bytes to highlight it in the details.

The screenshot shows the Wireshark interface with the following components:

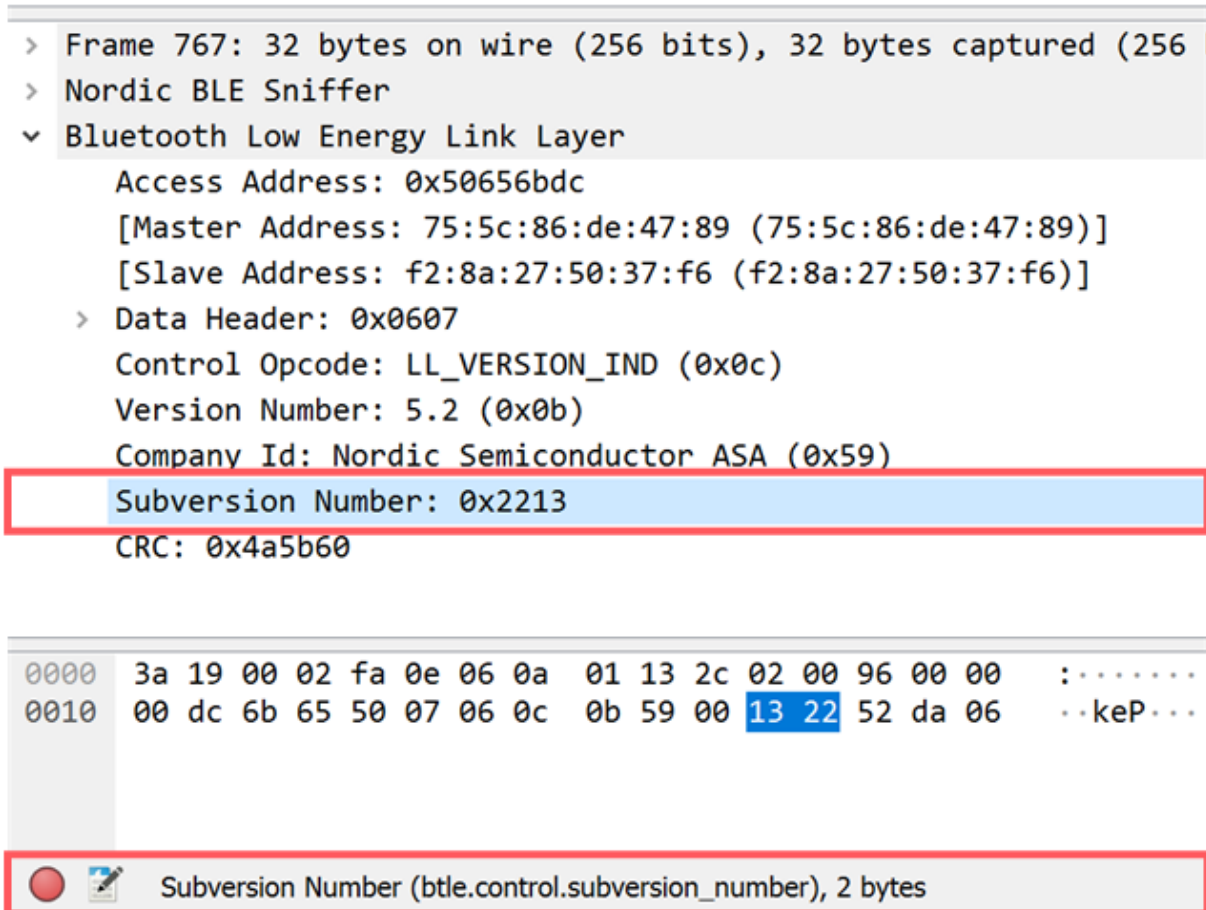
- Filtering:** `btle.length != 0`
- PACKET LIST:** A table of captured packets. Packet 249 is selected.

No.	Source	Destination	Protocol	Length	More Data	Event counter	Info
226	Master_0x9925cfbf	Slave_0x9925cfbf	L2CAP	10	False	2	Connection Para
227	Slave_0x9925cfbf	Master_0x9925cfbf	LE LL	9	True	2	Control Opcode:
229	Slave_0x9925cfbf	Master_0x9925cfbf	ATT	9	False	3	Rcvd Write Requ
230	Master_0x9925cfbf	Slave_0x9925cfbf	ATT	5	False	4	Sent Write Resp
233	Master_0x9925cfbf	Slave_0x9925cfbf	LE LL	12	False	6	Control Opcode:
247	Master_0x9925cfbf	Slave_0x9925cfbf	ATT	27	True	13	Sent Handle Val
249	Master_0x9925cfbf	Slave_0x9925cfbf	ATT	27	True	13	Sent Handle Val
- PACKET DETAILS:**
 - Extra packet information:** Board: 3, Header Version: 2, Packet counter: 51125, Length of packet: 10, Flags: 0x03, Channel: 14, RSSI: -34 dBm, Event counter: 13, Delta time (end to start): 150µs, [Delta time (start to start): 230µs], [Packet time (start to end): 296µs]
 - Bluetooth LE packet:**
 - Bluetooth Low Energy Link Layer:** Access Address: 0x9925cfbf, [Master Address: ff:ab:68:0c:34:fd (ff:ab:68:0c:34:fd)], [Slave Address: cd:27:b9:a3:61:f0 (cd:27:b9:a3:61:f0)], Data Header: 0x1b12, [L2CAP Index: 5], CRC: 0x872b9f
 - Bluetooth L2CAP Protocol:** Length: 23, CID: Attribute Protocol (0x0004)
 - Bluetooth Attribute Protocol:** Opcode: Handle Value Notification (0x1b), Handle: 0x0012 (Unknown), Value: 01010101010101010101010101010101
- PACKET BYTES:**

hexadecimal	ASCII
0000 03 2e 00 02 b5 c7 06 0a 03 0e 22 0d 00 96 00 00".....
0010 00 bf cf 25 99 12 10 17 00 04 00 1b 12 00 01 01	...%..
0020 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0030 01 01 e1 d4 f9

Figure 8: Wireshark interface

To view the display filter for any field, click a value in the packet details pane. The display filter is shown in the bottom left corner.



```

> Frame 767: 32 bytes on wire (256 bits), 32 bytes captured (256
> Nordic BLE Sniffer
  ▾ Bluetooth Low Energy Link Layer
    Access Address: 0x50656bdc
    [Master Address: 75:5c:86:de:47:89 (75:5c:86:de:47:89)]
    [Slave Address: f2:8a:27:50:37:f6 (f2:8a:27:50:37:f6)]
  > Data Header: 0x0607
    Control Opcode: LL_VERSION_IND (0x0c)
    Version Number: 5.2 (0x0b)
    Company Id: Nordic Semiconductor ASA (0x59)
    Subversion Number: 0x2213
    CRC: 0x4a5b60
  
```

```

0000  3a 19 00 02 fa 0e 06 0a 01 13 2c 02 00 96 00 00  :.....
0010  00 dc 6b 65 50 07 06 0c 0b 59 00 13 22 52 da 06  ..keP...
  
```

Subversion Number (btle.control.subversion_number), 2 bytes

Figure 9: Wireshark display filter

Use display filters to display a chosen packet subset. Most filters are based on the values of the packets, such as length or access address. The filter expressions use Boolean operators (&& || == != !). To construct a filter, click **Expression** in the filtering bar. See the following table for some examples.

Display filter	Description
btle.length != 0	Filter that displays only packets where the length field of the Bluetooth Low Energy packet is not zero, meaning it hides empty data packets.
btle.advertising_address	Filter that displays only packets that have an advertising address (advertising packets).
btle	Protocol filter that displays all Bluetooth Low Energy packets.
btatt, btcmp, btl2cap	Protocol filters for ATT, SMP, and L2CAP packets, respectively.
nordic_ble.channel < 37	Filter that displays only packets received on the data channels.

Table 2: Display filtering

The following tips can help when inspecting your data:

- Turn any field in the **packet details pane** into a column. To do so:
 - a) Right-click the value in the packet details.
 - b) Click **Apply as Column**.

```

  v Nordic BLE Sniffer
    Board: 3
    > Header Version: 2, Packet counter: 51125
    Length of packet: 10
    > Flags: 0x03
    Channel: 14
    RSSI: -34 dBm
    Event counter: 13
    Delta time (end to start): 150µs
    [Delta time (start to start): 230µs]
    [Packet time (start to end): 296µs]
  v Bluetooth Low Energy Link Layer
    Access Address: 0x9925cfbf
    [Master Address: ff:ab:68:0c:34:fd]
    [Slave Address: cd:27:b9:a3:61:f0]
    > Data Header: 0x1b12
    [L2CAP Index: 5]
    CRC: 0x872b9f
  
```

- Apply a value as a filter to, for example, see only operations affecting a particular handle. To filter packets that have a specific value for some field:
 - a) Right-click the value in the packet details.
 - b) Click **Apply as Filter**.
 - c) Click **Selected**.
- Save a set of captured packets to be able to look at them later. To do so:
 - a) Click the **Stop** button to stop capturing packets.
 - b) Click **File > Save As** to save all packets, or click **File > Export Specified Packets** to save a selection of packets.
- Clear the packet list and restart a capture by clicking the **Restart** button.

See the documentation on the [Wireshark website](#) for more information.

5 Common sniffing actions

The nRF Sniffer for Bluetooth LE can help you explore and debug Bluetooth Low Energy communication in a number of typical scenarios.

5.1 Sniffing advertisements from all nearby devices

Use the nRF Sniffer for Bluetooth LE to see advertisements from all nearby devices.

1. Run the [nRF Sniffer](#) (if not already running).
2. Ensure that **All advertising devices** is selected in the device list.

5.2 Sniffing advertisement packets involving a single Peripheral

Use the nRF Sniffer for Bluetooth LE to see advertisement packets, scan requests, and scan responses to and from a single device.

1. Run the [nRF Sniffer](#) (if not already running).
2. Select your device from the device list.

5.3 Sniffing a connection involving a single Peripheral

Use the nRF Sniffer for Bluetooth LE to sniff a connection between a specific Peripheral and a Central.

1. Run the [nRF Sniffer](#) (if not already running).
2. Select your device from the device list.
3. Connect the Central to the Peripheral.

5.4 Sniffing the pairing procedure of a connection

Use the nRF Sniffer for Bluetooth to sniff an encrypted connection between paired devices by sniffing the pairing procedure.

Note: If the *DK* or dongle running the nRF Sniffer firmware is reset, stored bond information is lost.

1. Run the [nRF Sniffer](#) (if not already running).
2. Select your device from the device list.
3. Enter the credentials for pairing. The procedure depends on the type of encryption.
 - For connections that use legacy pairing with Just Works:
 - a. Initiate pairing between the devices if it does not happen automatically.No further action is required.
 - For connections that use legacy pairing with a passkey:
 - a. Initiate pairing between the devices if it does not happen automatically.

- b. Select **Legacy Passkey** as input key and type the 6-digit passkey that is displayed on either the Central or the Peripheral into the input field in *Wireshark*.
 - c. Press **Enter**.
 - d. Enter the passkey into the other device.
- For connections that use legacy pairing with *OOB*:
 - a. Before the devices initiate pairing, select **Legacy OOB data** as input key and type the *OOB* key in big-endian, hexadecimal format with a leading "0x" into the input field in *Wireshark*.
 - b. Press **Enter**.
 - c. Connect the Central to the Peripheral.
 - d. Initiate pairing between the devices if it does not happen automatically.
- For connections that use LE Secure Connections in debug mode:
 - a. Enable Secure Connections debug mode on one or both of the devices.
 - b. Initiate pairing between the devices if it does not happen automatically.

In debug mode, the connection uses the debug keys specified in the [Bluetooth Core Specification](#). The nRF Sniffer uses the same keys to decrypt the encrypted packets.
- For connections that use LE Secure Connections with a private key:
 - a. Before the devices initiate pairing, select **SC Private Key** as input key and type the 32-byte Diffie-Hellman private key of your device in big-endian, hexadecimal format with a leading "0x" into the input field in *Wireshark*.
 - b. Initiate pairing between the devices if it does not happen automatically.

5.5 Sniffing a connection between bonded devices

Use the nRF Sniffer for Bluetooth to sniff an encrypted connection between bonded devices. If the nRF Sniffer has previously successfully sniffed the pairing procedure, it remembers the *LTK* needed to decrypt the connection. Otherwise, you must provide the *LTK*.

1. [Run the nRF Sniffer](#) (if not already running).
2. Select your device from the device list.
3. Enter the *LTK* for the bond.
 - For connections that have an existing legacy bond, select **Legacy LTK** as input key and type the legacy *LTK* key in big-endian, hexadecimal format with a leading "0x" into the input field in *Wireshark*.
 - For connections that have an existing LE Secure Connections bond, select **SC LTK** as input key and type the LE Secure Connections *LTK* key in big-endian, hexadecimal format with a leading "0x" into the input field in *Wireshark*.
4. Initiate encryption between the devices (pairing is not performed when a bond exists).

6 Troubleshooting

If you have problems installing or using the nRF Sniffer for Bluetooth LE, see the following sections for troubleshooting information.

The nRF Sniffer for Bluetooth LE is not listed in the Wireshark interface

Check that the hardware is set up correctly:

1. Ensure that the *DK* or dongle has been enumerated on *Universal Serial Bus (USB)* and that the drivers are loaded.
2. Ensure that the firmware HEX file has been programmed.
3. Reset the hardware by unplugging the hardware, waiting 5 seconds, and plugging it back in.

If these steps do not help, verify that you have installed the nRF Sniffer capture tool correctly and that the Python script located in the extcap folder can be run, as described in [Installing the nRF Sniffer capture tool](#) on page 7.

The nRF Sniffer for Bluetooth LE does not show up as a toolbar

Make sure that you enabled the nRF Sniffer interface toolbar.

To do so, click **View > Interface Toolbars > nRF Sniffer for Bluetooth LE**.

The nRF Sniffer for Bluetooth LE does not receive packets

When [programming the nRF Sniffer firmware](#), make sure to use the latest SEGGER J-Link software. If you used an older version, update your J-Link and program the firmware again.

The nRF Sniffer for Bluetooth LE does not receive packets on Windows

On Windows, COM port numbers higher than 199 are not supported. If the COM port number is COM200 or higher, rename the COM port on Windows to a COM port number that is COM199 or lower. To do so, complete the following steps:

1. Open the Device Manager and click **Ports (COM & LPT)**.
2. Right-click on your COM port and click **Properties**.
3. In Properties, go to the **Port Settings** tab and click **Advanced**.
4. Change the COM port number by clicking the COM port number drop-down and selecting a COM port that is less than 200. Select a COM port number that is not in the list of devices currently attached to your computer. These are listed in the Device Manager under **Ports (COM & LPT)**.
5. Click **OK** and accept the changes when asked "Do you want to continue".

The nRF Sniffer for Bluetooth LE occasionally works but appears unstable

Make sure that you are using the correct software versions as specified in the [prerequisites](#) and that you have [installed the Python requirements](#).

When [programming the nRF Sniffer firmware](#), make sure to use the latest SEGGER J-Link software. If you used an older version, update your J-Link and program the firmware again.

If the problem persists, force J-Link to use flow control in the serial connection:

1. Open **JLink.exe** (Windows) or **JLink.exe** (macOS/Linux) in the installation folder of the required J-Link version.
2. Enter `sethwfc force`.
3. Exit the JLink software.

Packets are displayed incorrectly

Verify that the NORDIC_BLE protocol is enabled in *Wireshark*. To do so, click **Analyze > Enabled Protocols** and verify that the NORDIC_BLE protocol is selected.

Verify that a stable release of *Wireshark* is used. Development and user build versions are not supported. For example, v3.0.7 and v3.2.0 are stable versions of *Wireshark*, as indicated by the second number being an even number. Version 3.1.x is a development version of *Wireshark*, indicated by the second number being an odd number.

Glossary

Development Kit (DK)

A development platform used for application development.

Long Term Key (LTK)

A key that is stored by both devices after the Bluetooth Low Energy pairing procedure has establishing a bond. The Long Term Key is either distributed by the peripheral device (when using legacy pairing) or derived from a Diffie-Hellman exchange (when using LE Secure Connections). The key is stored in both devices and is used to encrypt connections between the two devices.

Out of Band (OOB)

A communication channel that is outside of the defined activity. For example, in Bluetooth Low Energy, Out of Band pairing can be used to share encryption keys or authentication data using a different communication channel (such as NFC).

Received Signal Strength Indication (RSSI)

An indication of the power of a received radio signal.

Universal Serial Bus (USB)

An industry standard that establishes specifications for cables and connectors and protocols for connection, communication, and power supply between computers, peripheral devices, and other computers.

Wireshark

A cross-platform network protocol analyzer that can be used to view, analyze, and troubleshoot packets sent over a data network.

Acronyms and abbreviations

These acronyms and abbreviations are used in this document.

DK

Development Kit

LTK

Long Term Key

OOB

Out of Band

RSSI

Received Signal Strength Indication

USB

Universal Serial Bus

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function, or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

© 2021 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

