

Introducing the Adafruit Bluefruit LE Sniffer

Created by Kevin Townsend

Capturing from \\.\pipe\wireshark_nordic_ble [Wireshark 1.12.1 (v1.12.1	.1-0-g01b65bf from master-1.12)]
<u>File Edit View Go Capture Analyze Statistics Telephony Tools</u>	s Internals <u>H</u> elp
◎ ◎ ∡ ■ ₫ 🖿 🗎 🗙 🔁 역 수 수 🗛 👍 🚣	L 🗐 🖬 Q, Q, Q, 💟 🖉 🕺 🧏 🎉 🙀
Filter	Expression Clear Apply Save
No. Time Source Destination	Protocol Length Info
55593 2854.95085 slave Master	LE LL 26 Empty PDU
55594 2854.99892 Master Slave	LE LL 26 Empty PDU
55595 2855,00055 Slave Master	ATT 34 Rcvd Handle Value Notification, Handle: 0x001c
55596 2855.04974 Master Slave	LE LL 26 Empty PDU
55597 2855.05107 slave Master	LE LL 26 Empty PDU
55598 2855.09619 Master Slave	LE LL 26 Empty PDU
55599 2855.09776 Slave Master	LE LL 26 Empty PDU
55600 2855.19274 Master Slave	LE LL 26 Empty PDU
55601 2855.19395 Slave Master	LE LL 26 Empty PDU
55602 2855.29033 Master Slave	LE LL 26 Empty PDU
55603 2855.29210 slave Master	LE LL 26 Empty PDU
55604 2855.33914 Master Slave	LE LL 26 Empty PDU
55605 2855. 34070 slave Master	LE LL 26 Empty PDU
55606 2855. 38816 Master Slave	LE LL 26 Empty PDU
55607 2855.38971 slave Master	LE LL 26 Empty PDU
Frame 55595: 34 bytes on wire (272 bits), 34 byte	es captured (272 bits) on interface 0
∃ Nordic BLE sniffer meta	
Bluetooth Low Energy Link Layer	
Access Address: 0x84d81175 Data Header: 0x080a	
Data Header: 0x080a 000 = RFU: 0	
0 = More Data: False	
1 = Sequence Number: True	
0 = Next Expected Sequence Number: Fa	alse
	or a complete L2CAP message with no fragmentation (0x02)
000 = RFU: 0	or a comprete Ezew message with no tragmentation (0x02)
0 1000 = Length: 8	
Bluetooth L2CAP Protocol	
Length: 4	
CID: Attribute Protocol (0x0004)	
Bluetooth Attribute Protocol	
Opcode: Handle Value Notification (0x1b)	
Handle: 0x001c	
Value: 54	
0000 11 06 1b 01 6d 49 06 0a 01 0d 2b 2d 85 99 00	
0010 00 75 11 d8 84 0a 08 04 00 04 00 1b 1c 00 54	
0020 e9 73	.5
Bluetooth Attribute Protocol (btatt), 4 bytes Packets: 57843 · Di	Profile: Default

https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-sniffer

Last updated on 2021-12-16 12:50:28 AM EST

Table of Contents

Using with Sniffer V2 and Python35BLE Sniffer Hardware6Silicon Labs VCP Driver7Python 38Python Serial Support8Install Wireshark9Install BLE Sniffer Plugin9Final Check and Test Capture11Next Steps12Working with Wireshark13Capturing Exchanges Between Two Devices14Scan Response Packets15Connection Request16Wotly Event Date16Wotly Event Date18Closing Wireshark and nRF-Sniffer20Notify Event Date18Closing Wireshark and nRF-Sniffer20Notify Event Date21Notify Event Date22V2 Wireshark M Usage (old)21Nordic User Manual21Nordic User Manual22V2 Wireshark Vagge (old)23Install Wireshark23Install Wireshark23Install Wireshark23Install Wireshark Pugin23Install Wireshark Pugin23Install Wireshark Pugin23Install Wireshark Pugin23Install Wireshark Pugin23Install Wireshark29V2 Wirew Requirements (Bluck Boards)29V1 Sniffer V1 (old)28USB Driver Install29V1 Sniffer Software30V1 Sniffer Software30V1 Sniffer Sniffer31Nordicis nRF Sniffer Utility (Windows only)30V	Introduction	5
 Silicon Labs VCP Driver Python 3 Python 3 Install Wireshark Install BLE Sniffer Plugin Pinal Check and Test Capture Next Steps Working with Wireshark Control Request Connection Request Wire Request Consult Sniffer V2 (old) Varial Wireshark Sinstall Wireshark Sinstall Wireshark Sinstall Wireshark Scan Response Packets Connection Request Wire Request Colsing Wireshark an RF-Sniffer Colsing Wireshark Usage (old) Variation User Sniffer V2 (old) Install Wireshark Install Wireshark Sinstall Wireshark Sinstall Wireshark Sinstall Wireshark Sinstall Wireshark Sinstall Wireshark Sinstall Supplemental Information Variation Sniffer V1 (old) Sinstall Supplements (Black Boards) PitD Driver Requirements (Black Boards) PitD Driver Requirements (Black Boards) Python API (cross-Platform, no Registration) Son Constration Request Sniffer Son Son Son Son Son Son Son Son Son Son	Using with Sniffer V2 and Python3	5
 Silicon Labs VCP Driver Python 3 Python 3 Python 3 Install Wireshark Install BLE Sniffer Plugin Pinal Check and Test Capture Working with Wireshark Control Exchanges Between Two Devices Control Exchanges Between Two Devices Connection Request Concetton Request Working with Sniffer V2 (old) Nordic User Manual Install Wireshark Construction Request Nordic User Manual Nordic User Manual Install Wireshark Soland Response Packets Soland Response Packets Connection Request Wire Request Colsing Wireshark and nRF-Sniffer Colsing Wireshark and InRF-Sniffer Colsing Wireshark Construction Request Nordic User Manual Install Wireshark Install Wireshark Control User Manual Install Wireshark Control User Install Conduction With Sniffer V1 (old) Soland With Sniffer V1 (old) Soland With Sniffer V1 (old) VI Nordic nRF Sniffer Soland Wireshark Soland W	BLE Sniffer Hardware	6
• Python Serial Support8• Install Wireshark9• Install Oteck and Test Capture11• Next Steps12Working with Wireshark13• Working with Wireshark13• Capturing Exchanges Between Two Devices14• Scan Response Packets15• Connection Request16• Wirte Request16• Wireshark Notify Evchanges Between Two Devices17• Notify Evchanges Between Two Devices16• Wirte Request16• Wirte Request16• Notify Evch Data17• Consection Request16• Colsing Wireshark and nRF-Sniffer20• Notify Evch Data21• Using orth Sniffer V2 (old)21• Nordic User Manual21• Nordic User Manual22• V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Python 2 vs 323• Dealing Wirth Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• CP2104 Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30• V1 Nordic nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30• V1 Nordic nRF Sniffer Utility31• Getting Wireshark31	Silicon Labs VCP Driver	
• Python Serial Support8• Install Wireshark9• Install Oteck and Test Capture11• Next Steps12Working with Wireshark13• Working with Wireshark13• Capturing Exchanges Between Two Devices14• Scan Response Packets15• Connection Request16• Wirte Request16• Wireshark Notify Evchanges Between Two Devices17• Notify Evchanges Between Two Devices16• Wirte Request16• Wirte Request16• Notify Evch Data17• Consection Request16• Colsing Wireshark and nRF-Sniffer20• Notify Evch Data21• Using orth Sniffer V2 (old)21• Nordic User Manual21• Nordic User Manual22• V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Python 2 vs 323• Dealing Wirth Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• CP2104 Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30• V1 Nordic nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30• V1 Nordic nRF Sniffer Utility31• Getting Wireshark31		
 Install Wireshark Install BLE Sniffer Plugin Final Check and Test Capture Working with Wireshark Capturing Exchanges Between Two Devices Capturing Exchanges Between Two Devices Scan Response Packets Costing With Sniffer V2 (old) Nordic User Manual Install Wireshark Install Wireshark Install Wireshark Install Wireshark Install Wireshark Scan Response Packets Scan Response Packets Scan Response Packets Scan Response Packets With Request Regular Data Requests Closing Wireshark and nRF-Sniffer Obordic User Manual Install Wireshark Scale Response Packets Scale Response Scale Respons		
 Install BLE Sniffer Plugin Final Check and Test Capture Third Check and Test Capture Next Steps Working with Wireshark Capturing Exchanges Between Two Devices Capturing Exchanges Between Two Devices Connection Request Connection Request Connection Request Connection Request Notify Event Data Closing Wireshark and nRF-Sniffer Using with Sniffer V2 (old) Nordic User Manual Install Wireshark Plugin Install Wireshark NUsage (old) Vatireshark Plugin Install Wireshark Plugin Install Wireshark Plugin Closing With Sniffer V2 (old) Vatireshark Plugin Install Wireshark Bugin Install Wireshark Bugin Closing Vith Sniffer V2 (old) Vatireshark Plugin Scan Capture V2 Wireshark Usage (old) V3 Wireshark Plugin Dealing With Python 2 vs 3 Closing with Sniffer V1 (old) VB Driver Install CP210 Driver Requirements (Black Boards) V1 Sniffer Software V1 Nordic nRF Sniffer V1 Nordic nRF Sniffer Getting Wireshark Getting threshark Getting threshark Getting threshark Getting threshark Software Software Software Software Software Software Software Getting threshark Software Software<td></td><td></td>		
 Final Check and Test Capture Next Steps Working with Wireshark Sorting with Wireshark Contruing Exchanges Between Two Devices Connection Request Connection Request Wirte Request Regular Data Requests Closing Wireshark and nRF-Sniffer Colosing Wireshark Construction Structure Moving Forward Viris Sniffer V2 (old) Nordic User Manual Install Wireshark Install Wireshark Install Wireshark Install Wireshark Soft Provided Structure V2 Firmware Constructure V2 Sing with Python 2 vs 3 Install Wireshark Plugin Constructure V2 Sing with Sniffer V1 (old) V3 Sinffer Software V3 Sinffer V1 (old) V3 Sinffer Software V1 Sinffer Software V1 Nordic nRF Sniffer Utility (Windows only) Nordic's nRF Sniffer Utility (Windows only) Python API (cross-Platform, no Registration) V1 Nordic nRF Sniffer Utility Getting Wireshark Getting Wireshark Getting Wireshark Getting Wireshark Software Software Software Software Software Software Software Software Software Software		
Next Steps12Working with Wireshark13Working with Wireshark13Capturing Exchanges Between Two Devices14Scan Response Packets15Connection Request16Wite Request16Regular Data Requests17Notify Event Data18Closing Wireshark and nRF-Sniffer20Moving Forward21Using with Sniffer V2 (old)21Nordic User Manual21Nordic User Manual22V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark23Install Wireshark23Install Wireshark23Install Wireshark23V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark23Install Wireshark23Using with Sniffer V1 (old)28USB Driver Install29CP2104 Driver Requirements (Black Boards)29V1 Sniffer Software30V1 Nordic nRF Sniffer Utility (Windows only)30Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer Utility31Getting the Sniffer Utility31Getting threshark31	-	
Working with Wireshark13• Capturing Exchanges Between Two Devices14• Scan Response Packets15• Connection Request16• Write Request16• Write Request16• Regular Data Requests17• Notify Event Data20• Cosing Wireshark and nRF-Sniffer20• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• Nordic User Manual21• Nordic User Manual21• Nordic User Manual21• Nordic User Manual22• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Pugin23• Install Wireshark Pugin23• Dealing With Python 2 vs 325• Install Dependancies26• Va Sing with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Black Boards)29• VI Sniffer Software30• Using the Firmware VI Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Pieltorm, no Registration)30V1 Nordic nRF Sniffer31• Getting Wireshark31• Getting Wireshark31		
- Capturing Exchanges Between Two Devices14- Scan Response Packets15- Connection Request16- Write Request17Notify Event Data18- Closing Wireshark and nRF-Sniffer20- Moving Forward21Using with Sniffer V2 (old)21- Nordic User Manual21- Net Shiffer V2 Multi-Target Application22- V2 Firmware22V2 Wireshark Lusage (old)23- Install Wireshark Pugin23- Install Wireshark Pugin23- Dealing With Python 2 vs 325- Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29- CP2104 Driver Requirements (Black Boards)29- FTDI Driver Requirements (Black Boards)29- V1 Sniffer Software30- Using the Firmware V1 Sniffer30- Nordic's nRF Sniffer Utility (Windows only)30- V1 Nordic nRF Sniffer Utility (Windows only)30V1 Nordic nRF Sniffer Utility31- Getting Wireshark31- Getting Wireshark31	Working with Wireshark	13
- Capturing Exchanges Between Two Devices14- Scan Response Packets15- Connection Request16- Write Request16- Regular Data Requests17- Notify Event Data18- Closing Wireshark and nRF-Sniffer20- Moving Forward21Using with Sniffer V2 (old)21- Nordic User Manual21- Nordic User Manual21- Nordic User Manual21- Net Sniffer V2 Multi-Target Application22- V2 Firmware22V2 Wireshark Lusage (old)23- Install Wireshark Pugin23- Install Wireshark Pugin23- Dealing With Python 2 vs 325- Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29- CP2104 Driver Requirements (Black Boards)29- FTDI Driver Requirements (Black Boards)29- V1 Sniffer Software30- Using the Firmware V1 Sniffer30- Nordic's nRF Sniffer Utility (Windows only)30- V1 Nordic nRF Sniffer31- Getting the Sniffer Utility31- Getting Wireshark31	Working with Wireshark	13
Scan Response Packets15Connection Request16Write Request16Regular Data Requests17Notify Event Data18Closing Wireshark and nRF-Sniffer20Moving Forward21Using with Sniffer V2 (old)21Nordic User Manual21Nordic User Manual21Net Sniffer V2 Multi-Target Application22V2 Firmware22V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark Plugin23Install Wireshark Plugin23Install Wireshark Plugin23Using with Sniffer V1 (old)28USB Driver Install29CP2104 Driver Requirements (Black Boards)29FTDI Driver Requirements (Black Boards)29V1 Sniffer Software30Using the Firmware V1 Sniffer30V1 Nordic nRF Sniffer Utility (Windows only)30Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer Utility31Getting Wireshark31	-	
• Connection Request16Write Request16Regular Data Requests17Notify Event Data18• Closing Wireshark and nRF-Sniffer20• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• Nardic User Manual21• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Plugin23• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27USing with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• CP2104 Driver Requirements (Black Boards)29• CP2104 Driver Requirements (Black Boards)29• Using the Firmware V1 Sniffer30• Nordic's NRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting Wireshark31• Getting Wireshark31		
Write Request16Regular Data Requests17Notify Event Data18Closing Wireshark and nRF-Sniffer20Moving Forward21Using with Sniffer V2 (old)21Nordic User Manual21• NRF Sniffer V2 Multi-Target Application22V2 Firmware23• Install Wireshark23• Install Wireshark23• Install Wireshark23• Install Wireshark Plugin23• Install Opeendancies26• Test Capture27• Windows Install Supplemental Information27USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• TSD Triver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• V1 Nordic nRF Sniffer Utility30• Ordic nRF Sniffer Utility31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		
• Regular Data Requests17• Notify Event Data18• Closing Wireshark and nRF-Sniffer20• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• NRF Sniffer V2 Multi-Target Application22• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Pugin23• Install Wireshark Pugin23• Dealing With Python 2 vs 325• Install Supplemental Information27Vindows Install Supplemental Information27USINg with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31		
• Notify Event Data18• Closing Wireshark and nRF-Sniffer20• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• nRF Sniffer V2 Multi-Target Application22• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Plugin23• Using with Python 2 vs 325• Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31		
• Closing Wireshark and nRF-Sniffer20• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• nRF Sniffer V2 Multi-Target Application22• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark23• Install Wireshark Plugin23• Install Wireshark Plugin23• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27USING with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		
• Moving Forward21Using with Sniffer V2 (old)21• Nordic User Manual21• NRF Sniffer V2 Multi-Target Application22• V2 Firmware23• V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark23• Install Wireshark23• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's RRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		
Using with Sniffer V2 (old)21• Nordic User Manual21• nRF Sniffer V2 Multi-Target Application22• V2 Firmware22V2 Firmware23• Install Wireshark Usage (old)23• Install Wireshark Plugin23• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Install Suppendancies26• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting the Sniffer Utility31		
Nordic User Manual21nRF Sniffer V2 Multi-Target Application22V2 Firmware22V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark Plugin23Dealing With Python 2 vs 325Installing Dependancies26Test Capture27Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29 <cp2104 (black="" boards)<="" driver="" requirements="" td="">29<ptdi (black="" boards)<="" driver="" requirements="" td="">29V1 Sniffer Software30Using the Firmware V1 Sniffer30V1 Nordic's RF Sniffer Utility (Windows only)30Python API (cross-Platform, no Registration)31Getting the Sniffer Utility31Getting the Sniffer Utility31Getting Wireshark31</ptdi></cp2104>		
• nRF Sniffer V2 Multi-Target Application22• V2 Firmware22V2 Wireshark Usage (old)23• Install Wireshark23• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27USING with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's RF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		21
V2 Firmware22V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark Plugin23Dealing With Python 2 vs 325Installing Dependancies26Test Capture27Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		
V2 Wireshark Usage (old)23Install Wireshark23Install Wireshark Plugin23Dealing With Python 2 vs 325Installing Dependancies26Test Capture27Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Black Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31	nRF Sniffer V2 Multi-Target Application	22
Install Wireshark23Install Wireshark Plugin23Dealing With Python 2 vs 325Installing Dependancies26Test Capture27Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29· CP2104 Driver Requirements (Black Boards)29· FTDI Driver Requirements (Black Boards)29V1 Sniffer Software30· Using the Firmware V1 Sniffer30· Nordic's nRF Sniffer Utility (Windows only)30· Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31· Getting the Sniffer Utility31· Getting Wireshark31	• V2 Firmware	22
• Install Wireshark Plugin23• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29• V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31	V2 Wireshark Usage (old)	23
• Dealing With Python 2 vs 325• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31	Install Wireshark	23
• Installing Dependancies26• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31	Install Wireshark Plugin	23
• Test Capture27• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting Wireshark31	Dealing With Python 2 vs 3	25
• Windows Install Supplemental Information27Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)31• Getting the Sniffer Utility31• Getting Wireshark31	Installing Dependancies	26
Using with Sniffer V1 (old)28USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31	Test Capture	27
USB Driver Install29• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31	Windows Install Supplemental Information	27
• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31	Using with Sniffer V1 (old)	28
• CP2104 Driver Requirements (Black Boards)29• FTDI Driver Requirements (Blue Boards)29 V1 Sniffer Software 30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30 V1 Nordic nRF Sniffer 31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31	USB Driver Install	29
 FTDI Driver Requirements (Blue Boards) V1 Sniffer Software Using the Firmware V1 Sniffer Nordic's nRF Sniffer Utility (Windows only) Python API (Cross-Platform, no Registration) V1 Nordic nRF Sniffer Getting the Sniffer Utility Getting Wireshark 31 	CP2104 Driver Requirements (Black Boards)	
V1 Sniffer Software30• Using the Firmware V1 Sniffer30• Nordic's nRF Sniffer Utility (Windows only)30• Python API (Cross-Platform, no Registration)30V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting the Sniffer Utility31• Getting Wireshark31		
• Using the Firmware V1 Sniffer 30 • Nordic's nRF Sniffer Utility (Windows only) 30 • Python API (Cross-Platform, no Registration) 30 V1 Nordic nRF Sniffer 31 • Getting the Sniffer Utility 31 • Getting Wireshark 31		20
Nordic's nRF Sniffer Utility (Windows only) 30 Python API (Cross-Platform, no Registration) 30 V1 Nordic nRF Sniffer 31 Getting the Sniffer Utility 31 Getting Wireshark 31	V1 Sniffer Software	30
Python API (Cross-Platform, no Registration) 30 V1 Nordic nRF Sniffer 31 Getting the Sniffer Utility Getting Wireshark 31	-	30
V1 Nordic nRF Sniffer31• Getting the Sniffer Utility31• Getting Wireshark31		30
Getting the Sniffer Utility Getting Wireshark Site Stress S	Python API (Cross-Platform, no Registration)	30
Getting Wireshark 31	V1 Nordic nRF Sniffer	31
Getting Wireshark 31	Getting the Sniffer Utility	31

Select the Sniffer Target	32
V1 OS X Support	34
V1 Python API	35
Requirements	36
Download the API	36
Using the sniffer.py Wrapper	37
• Linux	37
• OS X	38
• Windows	38
Scanning for Devices	38
Locating the Log File	39
Analyze Data in Wireshark	39
FAQs	40
Downloads	44
• Files	44
Schematic	44

Introduction

Using a special firmware image provided by Nordic Semiconductors and the open source network analysis tool Wireshark, the Bluefruit LE Sniffer (https://adafru.it/edE) c an be used as a low cost Bluetooth Low Energy sniffer.

NOTE: This product can only be used to sniff Bluetooth Low Energy devices. It will not work with classic Bluetooth devices or transactions.

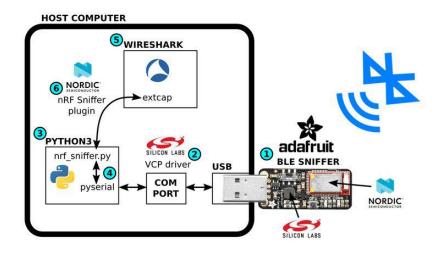


Since nRF-Sniffer is a passive solution that is simply scanning packets over the air, there is the possibility of missing packets using this tool (or any other passive sniffing solution). In order to capture as many packets as possible, be sure to run the sniffer on a USB bus that isn't busy and avoid running it in a virtual machine since this can introduce significant latency over USB.

Using with Sniffer V2 and Python3

Once things are all setup, usage is fairly easy. However, there are numerous separate items that need to be installed and configured. So the initial setup can be a bit cumbersome. We'll go through each step, but it can also help to have a general understanding of the overall setup.

Here's a simplified diagram of the setup:



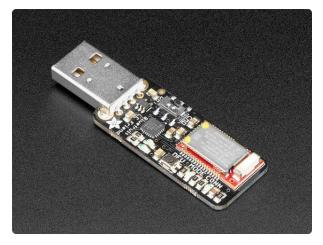
Here's a summary of all the parts needed:

- 1. The actual BLE sniffing hardware. This guide uses the Adafruit Bluefruit LE Sniffer with V2 firmware (https://adafru.it/edE).
- The BLE Sniffer uses a Silicon Labs CP2104 to provide USB to serial conversion. In order for this to show up as a COM port, the Silicon Labs <u>Virtual COM Port</u> driver (https://adafru.it/U3F) is needed.
- 3. The BLE sniffing plugin uses Python (https://adafru.it/deW).
- 4. To talk to the virtual com port from Python, the <u>pyserial module</u> (https://adafru.it/ S6E) needs to be installed.
- 5. Wireshark (https://adafru.it/eDB) is the main software front end used to facilitate BLE sniffing and decoding.
- 6. To talk to the BLE sniffer from Wireshark, the Nordic Semiconductor <u>nRF Sniffer</u> for BLE (https://adafru.it/VzF) plugin in is used.

These parts come from numerous different sources - at least 5 different vendors are shown in the diagram above. So this will be quite the journey. Here we go...

BLE Sniffer Hardware

You'll need one of these:



Bluefruit LE Sniffer - Bluetooth Low Energy (BLE 4.0) - nRF51822

Interested in learning how Bluetooth Low Energy works down to the packet level? Debugging your own BLE hardware, and trying to spot where something is going wrong? Or maybe you're...

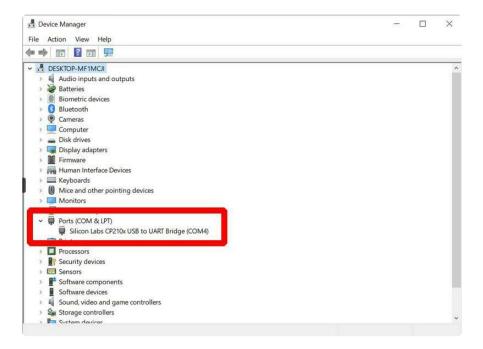
https://www.adafruit.com/product/2269

Silicon Labs VCP Driver

This driver allows the CP2104 chip on the Adafruit BLE Sniffer to show up as a COM port on your PC.



Once installed, a COM port should show up on your PC when the Adafruit BLE Sniffer is plugged into a USB port. It should have CP210x in the name.



This check does not require any of the other software components we install later. So if a COM port is not showing up at this point, do not proceed further until determining why.

Python 3

If Python 3 is not already installed on your system, go to the Python main page to learn how to download and install it for your specific system:



It should now be possible to launch Python and run some simple commands:



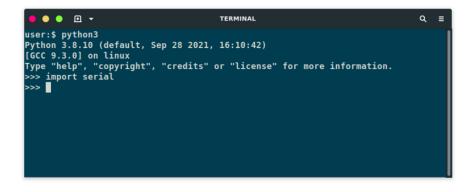
On Windows, try using py to launch Python.

Python Serial Support

To provide access to the COM port, install the pyserial package.



It should now be possible to launch Python and import the pyserial package:



NOTE: the import is actually **serial**, not pyserial.

Install Wireshark

Go to the Wireshark main page to learn how to download and install Wireshark for your specific system:



Once complete, it should be possible to run Wireshark and at least get the start screen:

••	•			The	Wire	shark	Netw	ork A	nalyzer			
<u>F</u> ile <u>E</u> dit	t <u>V</u> iew	<u>G</u> o	<u>C</u> apture	<u>A</u> nalyze	<u>S</u> tati	stics	Telep	hony	<u>W</u> ireless	5 <u>T</u> o	ols <u>H</u> elp	
	10			G Q	<	> 3		-1		÷	01	
Apply a	a display	filter	<ctrl-< td=""><th>/></th><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ctrl-<>	/>								
	Welco	me to	o Wiresh	nark								
	Captı	ure										
	using t	this filt	ter: 📕 E	nter a cap	ture fi	lter				•	All interfaces show	n -
	eno											
		pback:	: lo									
	any	tooth	monitor									
	nflo		monitor									
		ieue										
			ote captu	ure: ciscoo	lump							
	O Disp	layPo	rt AUX ch	nannel mo	nitor c	aptur	e: dpa	uxmor				
	Ran	dom p	acket ge	nerator: r	andpk	:						
				(port: sdjo								
				e: sshdun								
	UDP	^o Lister	ner remo	te capture	e: udpo	lump						
	Learr	ı										
	User's (Guide	Wiki	· Quest	ions a	and A	nswe	rs · I	4ailing I	Lists		
	You are	runnin	ig Wiresh	ark 3.2.3	(Git v3	.2.3 p	ackag	ed as	3.2.3-1).			
			-									
🖉 Rea	dy to loa	d or ca	apture						No Packe	ets	Profile	e: Default

Install BLE Sniffer Plugin

OK, finally, the thing we actually care about. The thing that will let us talk to the Adafruit BLE Sniffer and do some actual BLE sniffing. Let's download and install that BLE sniffing plugin!

Download Plugin from Nordic

Start by downloading the nRF Sniffer for BLE package from Nordic Semiconductor:

nRF Sniffer for Bluetooth LE

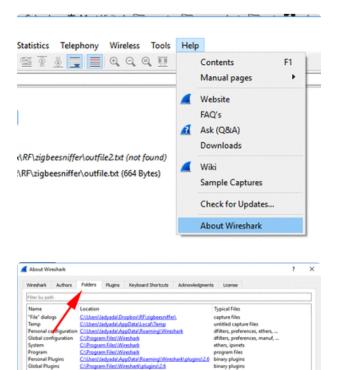
https://adafru.it/VzF

This will be a ZIP file. At the time of this guide, the version is 4.0.

Overview Downloads		
v4.x.x	Selected version 4.0.0 nrf_sniffer_for_bluetooth_le_4.0.0.zip	
	Changelog:	
	④ 4.0.0	~

Determine Wireshark Plugin Folder Location (extcap)

We need to install items from the ZIP file downloaded from Nordic into a specific Wireshark folder location. This location is different on different systems. To determine it for your system, do this:



Open Wireshark, in the Help menu select About wireshark

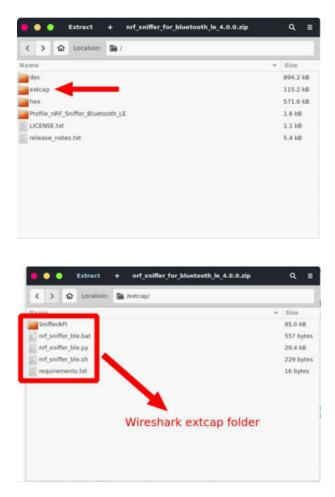
In the Folders tab, find the extcap path

We'll refer to this folder location as the Wireshark extcap folder.

Install BLE Sniffer Plugin into Wireshark

To install the plugin, simply copy the files shown below from the ZIP downloaded from Nordic into the Wireshark extcap folder location determined above.

Open the ZIP file downloaded from Nordic:



We only need the contents of the extcap folder from the ZIP file.

Extract and copy all of the contents of the extcap folder to the Wireshark extcap folder location.

Final Check and Test Capture

OK, now we can test things out with some real actual BLE sniffing! woot!

	apture Analyze Statisti	3-1111			
ily a display filter					
Welcome to	Wireshark				
Capture					
using this filte	r: 📕 Enter a capture filte	er		* All inte	rfaces shown -
enal					-
Loopback: I	0				
any bluetooth-m	onitor				
nflog		_			
nRF Sniffer	for Bluetooth LE: /dev/tty	US80-3.2			
DisplayPort	AUX channel monitor cap	pture: dpauxmo	n		
 Random par systemd jour 	cket generator: randpkt mal Export: sdjournal				
SSH remote	capture: sshdump				× .
Learn					
	Wiki · Questions an		-	ists	
You are running	Wireshark 3.2.3 (Git v3.2	2.3 packaged as	3.2.3-1).		
eady to load or cap	ture		No Packet		Profile: Default
,				-	
	Capture Analyze Statis				
📕 🛃 🛞 📄	n 🛛 🖉 🔍 🔇 🤇	tics Telephony		Tools Hel	1
📕 🛃 🛞 🚞	🖹 🕅 🔘 🔍 🄇 🕽) 3 I⊢ -II			
A display filter Time 1962 76.414719	No. 400 (17) Source 61:b1:f6:0e:63:36	Destination		Protocol Le	engthinfo 63 ADV NONCONN INI
Dely a display filter. Time 1062 76.434719 1063 76.518960 1064 76.625986	Ctrl/> Source 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36	Destination Broadcast Broadcast Broadcast		Protocol Le LE LL LE LL LE LL	engthinfo 63 ADV_NONCONN_INI 63 ADV_NONCONN_INI 63 ADV_NONCONN_INI
A Control Control Contro Contro Control Control Control Control	Ch1->> Source 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36	Destination Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol Le LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV_NONCONN_INI 63 ADV_NONCONN_INI 63 ADV_NONCONN_INI
ply a display filter.	<pre>Source Source 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36 61:b1:f6:0e:63:36</pre>	Destination Broadcast Broadcast Broadcast Broadcast		Protocol Le LE LL LE LL LE LL LE LL	engthinfo 63 ADV_NONCONN_INI 63 ADV_NONCONN_INI
20 20	CCD1-/> Source Source Sit bit 76:0e:63:36 61:bit 76:0e:63:36	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol La LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV. NONCONN IN 63 ADV. NONCONN IN
Display a display filter. Time 1062 76.414719 1063 76.318860 1064 76.625986 1065 76.625986 1065 76.6326746 1066 76.83267 1068 76.83267 ame 1: 63 bytes , rdia BLE Salffer	Controls	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol La LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV. NONCONN IN 63 ADV. NONCONN IN
Image: Constraint of the second sec	Controls	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol La LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV. NONCONN IN 63 ADV. NONCONN IN
Det a display filter. Time 1002 76.414719 1005 76.31850 1004 76.625086 1005 76.625086 1005 76.62746 1005 76.632507 1005 76.632507 ame 1: 63 bytes. 1016 876.831591 ame 1: 63 bytes.	Controls	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol La LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV. NONCONN IN 63 ADV. NONCONN IN
Det a display filter. Time 1002 76.414719 1005 76.31850 1004 76.625086 1005 76.625086 1005 76.62746 1005 76.632507 1005 76.632507 ame 1: 63 bytes. 1016 876.831591 ame 1: 63 bytes.	Controls	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast		Protocol La LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV. NONCONN IN 63 ADV. NONCONN IN
C C C	Control Control Contro Contro Control Control Control Control Co	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast	ed (504 b	Protocol Ld LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL	engthinfo 63 ADV, NORCONN, TW 63 ADV, NORCONN, TW 10 ADV, NORCONN, TW
Image: Control of the state of the	Control	Oestination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast 3 bytes captur 00 00 ce 03 0 b1 61 1e fr 0 95 5c cz ra	ed (504 b	M m M Protocol Le LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL BLS) on In BNS C a a	engthinfo 63 ADV, MORCONN, TW 63 ADV, MORCONN, TW 64 ADV, MORCONN, TW for Face / tsp/wiresh
Image: Control of the second	Control	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast	ed (504 b	M m M Protocol Le LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL BLS) on In BNS C a a	engthinfo 63 ADV, MORCONN, TW 63 ADV, MORCONN, TW 64 ADV, MORCONN, TW for Face / tsp/wiresh
Image: Control of the state of the	Control	Oestination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast 3 bytes captur 00 00 ce 03 0 b1 61 1e fr 0 95 5c cz ra	ed (504 b	M m M Protocol Le LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL BLS) on In BNS C a a	engthinfo 63 ADV, MORCONN, TW 63 ADV, MORCONN, TW 64 ADV, MORCONN, TW for Face / tsp/wiresh
Image: Control of the state of the	Control	Oestination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast 3 bytes captur 00 00 ce 03 0 b1 61 1e fr 0 95 5c cz ra	ed (504 b	M m M Protocol Le LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL BLS) on In BNS C a a	engthinfo 63 ADV, MORCONN, TW 63 ADV, MORCONN, TW 64 ADV, MORCONN, TW for Face / tsp/wiresh
0 0 0 0 Torse 1 0 0 0 062 74, 414750 0	Control	Destination Francist Francist	ed (504 b	Protocol Le LE LL LE LE LL LE	engthinfo 63 ADV, MORCONE, TW 63 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 65 ADV, MORCONE, TW
0 0 0 0 Torse 1 0 0 0 062 74, 414750 0	R C Q	Destination Francist Francist	ed (504 b	Protocol Le LE LL LE LE LL LE	engthinfo 63 ADV, MORCONE, TW 63 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 65 ADV, MORCONE, TW
0 0 0 0 Torse 1 0 0 0 062 74, 414750 0	R C Q	Destination Francist Francist	ed (504 b	Protocol Le LE LL LE LE LL LE	engthinfo 63 ADV, MORCONE, TW 63 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 64 ADV, MORCONE, TW 65 ADV, MORCONE, TW
C C C	Res Question	Destination Freedcast Breedcast Breedcast	ed (504 b 1 ···8 v 6 ···pKq. Displayed:	Protocol Le E LL E L E	engthinfo 63 ADV, MORCON, TW 63 ADV, MORCON, TW 64 ADV, MORCON, TW 65 ADV, MORCON,
Q Q Q ppy a display filter Tene 1962 76, 413/150 136969 1963 76, 313696 1966 76, 623746 1966 76, 623965 1966 76, 832297 ame 1: 63 bytes 1961 76, 832297 1969 06 38 61 77 1963 06 38 61 77 1960 06 38 61 77 1970 06 38 61 77 1960 06 38 61 77 1970 06 38 61 77 1960 06 38 61 77 1970 06 71 21 1960 77 8, 832297 1987 78, 832297 1960 06 38 61 77 1970 06 38 61 77 1970 06 38 61 77 1970 06 38 61 77 1980 78, 832297 1987 78, 832297 1980 79, 970 79, 71 21 1987 79, 970 71 21 1980 79, 970 40 71 21 1988 79, 810 40 71 21 1988 5, 50, 810 40 70 40 71 21 1988 6, 980 40 40 71 21	Image: Source Source Source Source Source Source Source	Destination Freedcast Breadcast Breadcast	ed (594 b 1	Pretocol Le E LL E L E	enghinfo 63 APV, HORCONE, TW 63 APV, HORCONE, TW 64 APV, HORCONE, TW 65 APV, HORCONE, TW
A Image Image opy a display filter Time 1062 76, 414759 1063 76, 514969 1063 76, 514969 1064 76, 425968 1066 76, 425968 1066 76, 425968 1067 76, 435951 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 1066 76, 432591 106 76, 432591 106 76, 432591 106 06 06 19 90 90 109 90 108 07 45 71 21 100 90 2 rAFF Smiller for I 108 0 109 00 108 0 109 00 109 00 109 71 21 2 rAFF Smiller for I 108 0 109 00 108 0 109 00 108 0 109 00 109 00 109 10	Image: Source Image: Source Image: Source Source Source Source Source	Destination Freedcast Breadcast Breadcast	ed (594 b 1	Protocol La Li	engthinfo 63 ADV, NORCONA, TM 63 ADV, NORCONA, TM 64 ADV, NORCONA, TM 64 ADV, NORCONA, TM 65 ADV, NORCONA, TM
C C	Image: Source Source Source Source Source Source Source	Destination Freedcast Breadcast Breadcast	ed (594 b 1	Pretocol Le E LL E L E	engthinfo 63 ADV, NORCONN, TW 63 ADV, NORCONN, TW 16 ADV, NORCONN, TW 16 ADV, NORCONN, TW 17 ADV, NORCONN, TW 17 ADV, NORCONN, TW 18 ADV, NORCONN, TW 19 ADV, NORCONN, TW 19 ADV, NORCONN, TW 10 ADV, NORCONN, TW
C C	Image: Source Image: Source Image: Source Source Source Source Source Source Source Source <td< td=""><td>Destination Freedcast Breadcast Breadcast</td><td>ed (594 b 1</td><td>Protocol La Li Li</td><td>the second second</td></td<>	Destination Freedcast Breadcast Breadcast	ed (594 b 1	Protocol La Li	the second
C C	Control	00 00 00 00 01 00 00 00 00 02 03 10 10 10 03 bytes captur 00 00 00 10 10 03 bytes captur 00 00 00 10 10 10 04 04 04 07 10	ed (594 b 1	Pretecci Li Full E Li	the second

- Plug in the Adafruit BLE Sniffer.
- Launch Wireshark.
- The sniffer should show up under the available capture devices.

- Double click on the sniffer capture device.
- This will open the device and start capturing.
- If there is BLE traffic, it will be seen right away.
- If there is no BLE traffic, it will look like this.
- Note the device has opened properly and is sniffing, there's just nothing to be seen.

Next Steps

nRF Sniffer for Bluetooth L... <live capture in progress> No Packets

Once everything is working as shown above, you are ready to move on to working with these BLE packets.

Profile: Default

Now go here to learn how to look at BLE packets with Wireshark

Working with Wireshark

This page will work with both V1 and V2 sniffer firmware, once you've got the software installed

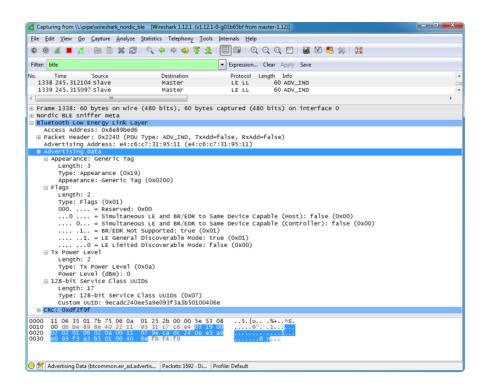
Working with Wireshark

Once Wireshark has loaded, you should see the advertising packets streaming out from the selected BLE device at a regular interval, as shown in the image below:

ile	Edit View	<u>Go</u> <u>C</u> apt	ure <u>A</u> na	lyze <u>S</u> t	atistics	Telepho	ny <u>I</u> od	ols Inter	nals <u>H</u> elp							
D) 🔏 🔳	đ 🗎 🖻	n x	2	0 4	🔿 🗳	· .				· • •	1	1 8 %	Ħ		
	btle							•	Expression	Class	Annha Ca					
												e				
0.	Time	Sour				estination	1			Length						
	10 38.042					aster			LE LL		ADV_IN					
	12 38.040					aster			LE LL		ADV_ING					
	13 38.596					aster aster			LE LL		ADV_INI ADV_INI					
	14 38.599					aster			LE LL		ADV_ING					
	15 38.601					aster			LE LL		ADV_IN					
	16 39.149					aster			LE LL		ADV_ING					
	17 39.152					aster			LE LL		ADV_IN					
	18 39.154					aster			LE LL		ADV_IN					
	19 39.697					aster			LE LL		ADV_IN					
	20 39.700					aster			LE LL		ADV_IN					
	21 39,702					aster			LE LL		ADV_IN					
	22 40.247					aster			LE LL		ADV_IN					
	23 40,250					aster			LE LL		ADV_IN					
	24 40.252					aster			LE LL		ADV_IN					
	25 40,800					aster			LE LL		ADV_IN					
	26 40,803					aster			LE LL		ADV_IN					
2	27 40,805	6880 sla	ive			aster			LE LL		ADV_IN					
2	28 41.352	2920 51a	ive		м	aster			LE LL	60	ADV_IN)				
2	29 41.355	4480 51a	ive		M	aster			LE LL	60	ADV_IN)				
2	30 41.357	2300 s1a	ive		м	aster			LE LL		ADV_IN					
	ame 1: 60	hater	an udna	(490	hite	60 k			d (400	dec) o	n dataa	Enco				
	rdic BLE :			. (480	Dits	, 60 6	ytes o	capture	20 (480	ones) o	n incer	race	,			
	uetooth L			. 1 31/01	r											
011	aecooch e	on energ	JY LIIK	Laye												
000	11 06 3	5 01 42	70 06	0a 01	1 25 2					%+.						
		e 89 8e			5 31 c					·1						
010		1 06 02 3 a3 b5			79ec efbf		4 0e e	25 a9		@ n	3					
010 020		asus	01 00	40 01	6 10 1					e 11						
010 020	60 95 1															
010	e0 95 1:															

One of the key benefits of WireShark as an analysis tool is that it understands the raw packet formats and provides human-readable displays of the raw packet data.

The main way to interact with BLE data packets is to select one of the packets in the main window, and then expand the Bluetooth Low Energy Link Layer treeview item in the middle of the UI, as shown below:



Clicking on the Advertising Data entry in the treeview will highlight the relevant section of the raw payload at the bottom of the screen, but also provides human readable information about the payload that can save you a lot of time trying to debug or reverse engineer a device.

We can see, for example, that the device is advertising itself as a Bluetooth Low Energy only device ('BR/EDR Not Supported'), with a TX Power Level of OdBm, and a single service is being advertised using a 128-bit UUID (the UART service in this case).

Capturing Exchanges Between Two Devices

If you wish to sniff data being exchanged between two BLE devices, you will need to establish a connection between the original device we selected above and a second BLE device (such as an iPhone or an Android tablet with BLE capabilities).

The nRF-Sniffer firmware is capable is listening the all of the exchanges that happen between these devices, but can not connect with a BLE peripheral or central device itself (it's a purely passive device).

Scan Response Packets

If you open up nRF UART on an Android or iOS device, and click the Connect button, the phone or tablet will start scanning for devices in range. One of the side effects of this scanning process is that you may spot a new packet in Wireshark on an irregular basis, the 'SCAN_REQ' and 'SCAN_RSP' packets:

Capturing	g from \\.\p	ipe\wiresha	rk_nordic_b	ole [W	ireshark	1.12.1 (v1.12.1-0	-g01b65	bf from r	naster-1	.12)]							- 0 x
Eile <u>E</u> dit	<u>V</u> iew <u>G</u> o	Capture	Analyze	Statisti	cs Tele	phony	Tools	Internal	: <u>H</u> elp									
D 0 /			X2	Q	\$	40 7	F 🕹			Q®		1	8	36	11			
								_										
ilter:								 Expr 	ession	Clear	Apply	Save						
	me	Source			Destina				tocol L									
	13.33929				Mast) ADV_ 3 SCAN							
	13.340/1				Mast				LL		SCAN							
	13.3439				Mast				LL		ADV_							
3894 7	13.88351	19 slave			Mast	er		LE	LL	60	ADV_	IND						
	13.88654				Mast	er			LL		ADV_							
	13.88828	35 slave			Mast	er		LE	LL	60) ADV_	IND						
		11																+
	3892: 38			(304	bits)	, 38	bytes o	captur	ed (30	4 bit:	s) on	inte	rface	0				
	BLE SN																	
	oth Low ss Addre			/er														
🗉 Scan	rtising Respons vertisir	e Data:	: e4:c6: 0509554			1 (e4	:c6:c7	:31:95	:11)									
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7:	:31:95	:11)									
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type:	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7:	:31:95	:11)					_				
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7:	:31:95	:11)	_	_	_		_	_	_	_	
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7	:31:95	:11)	_		_	_		_			
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7	:31:95	:11)			_			_	_	_	
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7:	:31:95	:11)									
⊟ Scan ⊟ Ad	Responsivertisin Device M Length Type: Device	e Data: ng Data Name: UA n: 5 Device Name:	0509554 RT Name (0)	15254		1 (e4	:c6:c7:	:31:95	:11)									
e Scan	Respons vertisin Device M Length Type: Device 0x68a41	e Data: 19 Data kame: UA 1: 5 Device Name: c	0509554 RT Name (0) JART	415254 (09)	1													
■ scan ■ Ad' ■ CRC:	Responsivertisin Device M Length Type: Device	e Data: 19 Data kame: UA 1: 5 Device Name: C 1 75 7ff 19 8e 44	0509554 RT Name (0) JART	(09)	25.0	0.00	97.00.0	00	:11)									
 ⇒ Scan ⇒ Ad' ⇒ ⇒ CRC: 	Responsive tisin Device N Length Type: Devicc 0x68a41	e Data: 19 Data kame: UA 1: 5 Device Name: C 1 75 7ff 19 8e 44	0509554 RT Name (0) JART	(09)	25.0	0.00	97.00.0	00	· · · · U. ·									

The Scan Response is an optional second advertising packet that some Bluetooth Low Energy periperhals use to provide additional information during the advertising phase. The normal mandatory advertising packet is limited to 31 bytes, so the Bluetooth SIG includes the possibility to request a second advertising payload via the Scan Request.

You can see both of these transactions in the image above, and the Device Name that is included in the Scan Response payload (since the 128-bit UART Service UUID takes up most of the free space in the main advertising packet).

For more information on Scan Responses and the advertising process in Bluetooth Low Energy see our Introduction to Bluetooth Low Energy Guide (https://adafru.it/iCo).

Connection Request

Once we click on the UART device in nRF UART, the two device will attempt to connect to each other by means of a Connection Request, which is initiated by the central device (the phone or tablet).

We can see this CONNECT_REQ in the timeline in the image below:

	dic_ble [Wireshark 1.12.1 (v1.1		((),(),(),(),(),(),(),(),(),(),(),(),(),	
<u>File Edit View Go Capture Anal</u>	ze Statistics Telephony To	ols Internals Help		
0 0 🗶 🔳 🔬 📄 🗙	2 🔍 🔶 🔶 🐼 :		우, 0, 🖂 🗃 🗹 🥵 % 📜	
Filter:		 Expression 	Clear Apply Save	
lo. Time Source	Destination	Protocol		
6498 1190, 19107 Slave	Master	LE LL	60 ADV_IND	
6499 1190, 73693 slave	Master	LE LL	60 ADV_IND	
6500 1190, 73983 slave	Master	LE LL	60 ADV IND	
6501 1190,74166 slave	Master	LE LL	60 ADV_IND	
6502 1191.29056 slave	Master	LE LL	60 ADV_IND	
6503 1191, 29414 slave	Master	LE LL	60 ADV_IND	
6504 1191.29600 slave	Master	LE LL	60 ADV_IND	
6505 1191.84026 slave	Master	LE LL	60 ADV_IND	
6506 1191, 84456 slave	Master	LE LL	60 ADV_IND	
6507 1191.84624 slave	Master	LE LL	60 ADV_IND	
6508 1192,40771 slave	Master	LE LL	60 ADV_IND	
6509 1192.41270 Slave	Master		60 CONNECT_REQ	
6510 1192.45658 Master	Slave	LE LL	26 Empty PDU	
6511 1192.45809 slave	Master	LE LL	26 Empty PDU	
6512 1192.50610 Master	slave	ATT		11.0
6513 1192. 50758 slave	Master		35 Rcvd Write Request, Handle: 0x00	JIE
6513 1192.50/58 STave	slave	LE LL	26 Empty PDU	
		LE LL	26 Empty PDU	
6515 1192.60471 slave	Master	LE LL	26 Empty PDU	
6516 1192.69980 Master	Slave	LE LL	26 Empty PDU	
6517 1192.70140 slave	Master	LE LL	26 Empty PDU	
6518 1192.74868 Master	Slave	LE LL	26 Empty PDU	
				,
E Frame 6509: 60 bytes on w	re (480 bits), 60 byt	es captured (48	30 bits) on interface 0	
Nordic BLE sniffer meta				
Bluetooth Low Energy Link	Laver			
57				
0000 11 06 35 01 ae 89 06 0	a 01 25 2c 00 00 98	00 00 E	۵/	
0010 00 d6 be 89 8e 85 22 0		35 31	%, k"1	
0020 c7 c6 e4 75 11 d8 84	f 2a c4 02 09 00 27	00 00u	*	
0030 00 bc 02 ff ff 03 e0 1	f b0 c9 cb 8f			

Write Request

Once the connection has been established, we can see that the nRF UART application tries to write data to the BLEFriend via a Write Request to handle '0x001E' (which is the location of an entry in the attribute table since everything in BLE is made up of attributes).

Capturing from \\.\pipe\wireshark_nordic_ble	[Wireshark 1.12.1 (v1.12.)	1-0-g01b65bf from m	naster-1.12)]	- 0 - X
<u>File Edit View Go Capture Analyze Stat</u>	stics Telephony <u>I</u> ools	Internals Help		
• • * = * b b x 2 •	(🗢 🔿 🖥 🕹		Q. Q. 🖸 🗃 🕅 🥵 % 🖼	
Filter:		 Expression 	Clear Apply Save	
No. Time Source	Destination	Protocol L	ength Info	
6508 1192.40771 slave	Master	LE LL	60 ADV_IND	
6509 1192.41270 slave	Master	LE LL	60 CONNECT_REQ	-
6510 1192.45658 Master 6511 1192.45809 Slave	slave	LE LL	26 Empty PDU	
6511 1192.45809 STave 6512 1192.50610 Master	Master Slave	LE LL ATT	26 Empty PDU 35 Rcvd Write Request, Handle: 0x001e	
6513 1192, 50758 Slave	Master	LE LL	26 Empty PDU	
6514 1192.60330 Master	slave	LE LL	26 Empty PDU	
6515 1192.60471 Slave	Master	LE LL	26 Empty PDU	
6516 1192.69980 Master	slave	LE LL	26 Empty PDU	
6517 1192.70140 slave	Master	LE LL	26 Empty PDU	•
< III				•
• Frame 6512: 35 bytes on wire (28	0 bits), 35 bytes	captured (280	0 bits) on interface 0	
Nordic BLE sniffer meta				
Bluetooth Low Energy Link Layer				
Access Address: 0x84d81175				
Data Header: 0x0902				
CRC: 0x32acff Bluetooth L2CAP Protocol				
Length: 5				
CID: Attribute Protocol (0x000	4)			
Bluetooth Attribute Protocol				
Opcode: Write Request (0x12)				
Handle: 0x001e				
value: 0100				
0000 11 06 1c 01 b2 89 06 0a 03	0b 2b 02 00 37 bd		. 7	
0010 00 75 11 d8 84 02 09 05 00			+7	
0020 4c 35 ff		L5.		
😑 💅 Bluetooth Attribute Protocol (btatt), 5 bytes	Packets: 31996	Profile: Default		

What this write request is trying to do is enable the 'notify' bit on the UART service's TX characteristic (https://adafru.it/k7b) (0x001E is the handle for the CCCD or 'Client Characteristic Configuration Descriptor (https://adafru.it/ecl)'). This bit enables an 'interrupt' of sorts to tell the BLEFriend that we want to be alerted every time there is new data available on the characteristic that transmits data from the BLEFriend to the phone or tablet.

Regular Data Requests

At this point you will start to see a lot of regular Empty PDU requests. This is part of the way that Bluetooth Low Energy works.

Similar to USB, all BLE transaction are initiated by the bus 'Main', which is the central device (the tablet or phone).

In order to receive data from the bus secondary (the peripheral device, or the BLEFriend in this particular case) the central device sends a 'ping' of sorts to the peripheral at a delay known as the 'connection interval' (not to be confused with the one-time connection highlighted earlier in this tutorial).

We can see pairs of transaction that happen at a reasonably consistent interval, but no data is exchanged since the BLEFriend (the peripheral) is saying 'sorry, I don't have any data for you':

Cile Cel	it View	Ge Canture	Analyze	Statistics Tele	heny Teel	Internals	Hele		_				
lie In	-			-	nony Tool	-							
DO	A .	🛃 🖿 🗋	XR	୍ 🔍 🗢 🔿	🤹 🐔 🛓		⊕ €] 🛛 🖉) 🕵 🎋	11		
ilter:						 Express 	ion C	lear App	lv Save				
0.	Time	Source		Destina	ion			igth Info	·				
		0771 Slave		Maste		LE L			V_IND				
		270 Slave		Maste		LEL			NNECT_REG	_			
		658 Master		Slave		LEL			pty PDU	4			
		5809 Slave		Maste					pty PDU				
)610 Master		Slave		ATT			vd Write	Pequest	Handle	• 0x001e	
		758 Slave		Maste		LEL			pty PDU	Request	, nanore	. 00016	
)330 Master		slave		LEL			pty PDU				
		471 Slave		Maste		LEL			pty PDU				
		980 Master		slave	-	LEL			pty PDU				
		0140 slave		Maste		LE L			DTV PDU				
		868 Master		Slave		LEL			pty PDU				
6519	1192.74	1986 Slave		Maste	r i	LEL			pty PDU				
		803 Master		slave		LE L	L		pty PDU				
6521	1192.79	943 slave		Maste	r	LE L	L		DTV PDU				
6522	2 1192.84	875 Master		slave		LE L	L	26 Em	pty PDU				
6523	3 1192.85	5002 slave		Maste	r	LE L	L		pty PDU				
6524	1192.94	425 Master		slave		LE L	L	26 Em	pty PDU				
6525	5 1192.94	553 Slave		Maste	r	LE L	L		PTY PDU				
6526	5 1193.04	087 Master		slave		LE L	L	26 Em	pty PDU				
6527	7 1193.04	208 slave		Maste	r	LE L	L	26 Em	pty PDU				
6528	8 1193.13	8933 Master		slave		LE L	L	26 Em	pty PDU				
6529	9 1193.14	1064 slave		Maste	r	LE L	L	26 Em	pty PDU				
	1193.23	8667 Master		slave		LE L	L	26 Em	DTV PDU				
				(208 bits)	26 byte	s captured	(208	bits) (on interf	ace O			
		niffer me											
		w Energy		/er									
		ress: 0x8	d81175										
		er: 0x0009											
CR	c: 0xb5e	ef 55											
000	11 06 13	01 b9 89	06 0a	09 11 2b 07	00 99 00	00		+					
010	00 75 11	d8 84 09	00 ad	f7 aa		.u.							
N 10	\nine\wirer	hark pordic b	et clive car	ture i Packets	33336	Profile: Defau	l+						

Notify Event Data

To see an actual data transaction, we simply need to enter some text in our terminal emulator SW which will cause the BLEFriend to send the data to nRF UART using the UART service.

Entering the string 'This is a test' in the terminal emulator, we can see the first packet being sent below (only the 'T' character is transmitted because the packets are sent out faster than we enter the characters into the terminal emulator):

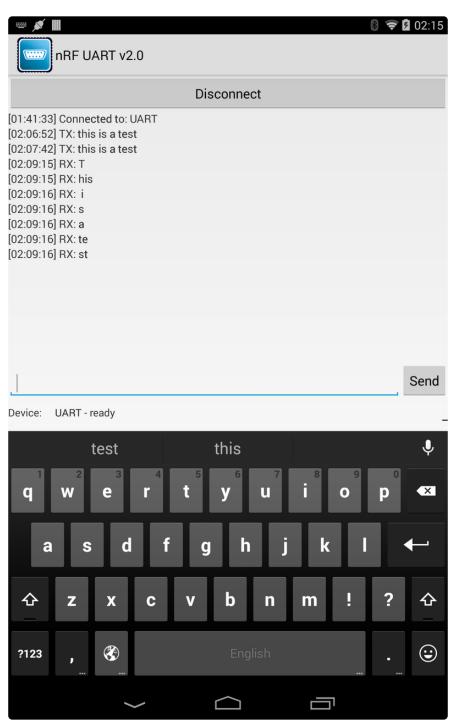
	lit <u>V</u> iew <u>G</u> o	<u>Capture</u> <u>Analyze</u>	Statistics Telephony Tools	Internals <u>H</u> elp		
DO	a 🔳 🙇		🔍 🗢 🔿 👍 🛂		0, 0, 🖸 🐺 🗹 🥵 % 💢	
ilter:				 Expression 	Clear Apply Save	
	Time	Source	Destination	Protocol	Length Info	
5559	3 2854.95085	slave	Master	LE LL	26 Empty PDU	
	1 2854.99892		Slave	LE LL	26 Empty PDU	
	5 2855.00055		Master	ATT	34 Rcvd Handle Value Notification, Handle: 0x001	C
	5 2855.04974		slave	LE LL	26 Empty PDU	
	7 2855.05107		Master	LE LL	26 Empty PDU	
	8 2855.09619		slave	LE LL	26 Empty PDU	
	9 2855.09776		Master	LE LL	26 Empty PDU	
	2855.19274 2855.19395		Slave Master	LE LL	26 Empty PDU	
	2855.29033		Slave	LE LL LE LL	26 Empty PDU 26 Empty PDU	
	2 2855.29033		Master	LE LL	26 Empty PDU 26 Empty PDU	
	4 2855.33914		Slave	LE LL	26 Empty PDU	
	5 2855, 34070		Master	LE LL	26 Empty PDU	
	5 2855. 38816		Slave	LE LL	26 Empty PDU	
	7 2855, 38971		Master	LELL	26 Empty PDU	
Ac	cess Address ta Header: (000 =	RFU: 0				
Ac Da	ccess Address ta Header: (000 = 0 = 0 = 10 = 000 = 0 1000 =	s: 0x84d81175 0x080a RFU: 0 More Data: F Sequence Num Next Expected LLID: Start o RFU: 0	alse ber: True d Sequence Number: Fa		L2CAP message with no fragmentation (0x02)	
	ccess Address ta Header: 0 000 = 0 = 1 = 10 = 000 = 0 1000 = c: 0x6e97ce	s: 0x84d81175 0x080a RFU: 0 More Data: F: Sequence Num Next Expecte LLID: Start (RFU: 0 Length: 8	alse ber: True d Sequence Number: Fa		L2CAP message with no fragmentation (0x02)	
AC Da • Da	ccess Address ta Header: (000 = 0 = 10. = 10 = 000 = 0 1000 = c: 0x6e97ce	s: 0x84d81175 0x080a RFU: 0 More Data: F: Sequence Num Next Expecte LLID: Start (RFU: 0 Length: 8	alse ber: True d Sequence Number: Fa		L2CAP message with no fragmentation (0x02)	
AC Da Da Da Da Da Da	ccess Address ta Header: (000 = 0 = 10. = 000 = 0 1000 = cc: 0x6e97ce tooth L2CAP	s: 0x84d81175 0x080a RFU: 0 More Data: F. Sequence Num Next Expecte LLID: Start RFU: 0 Length: 8 Protocol	alse ber: True d Sequence Number: Fa of an L2CAP message c		L2CAP message with no fragmentation (0x02)	
	ccess Address ta Header: (000 = 0 = 10. = 000 = 000 = 000 = c: 0x6e97ce tooth L2CAP ength: 4 D: Attribute	s: 0x84d81175 0x080a RFU: 0 More Data: F Sequence Num Next Expecter LLID: Start 0 LEngth: 8 Protocol e Protocol (0)	alse ber: True d Sequence Number: Fa of an L2CAP message c		L2CAP message with no fragmentation (0x02)	
AC Da Da Da Da Da Da Da Da Da Da	ccess Addres: ta Header: (000 = 1 = 10 = 000 = 10 = 000 = cc: 0x6e97ce ctooth L2CAP ingth: 4 D: Attribut.	<pre>s: Öx84d81175 Dx080a RFU: 0 More Data: F Sequence Num Next Expecte. LLID: Start G RFU: 0 Length: 8 Protocol e Protocol (0 bute Protocol) e Value Notif</pre>	alse ber: True d Sequence Number: Fa of an L2CAP message c		L2CAP message with no fragmentation (0x02)	
AC Da Da Da CF Blue CI Blue CI Blue CI CI CI CI CI CI CI CI CI CI CI CI CI	<pre>cess Addres: tta Header: 0 000 = 1 = 100 = 000 = 0000 = 0000 = 0000</pre>	s: 0x84d81175 0x080a RFU: 0 More Data: F. Sequence Num Next Expecte LLD: Start C RFU: 0 LLD: Start C RFU: 0 Length: 8 Protocol e Protocol (0 Dute Protocol c Sulte Notific c	alse ber: True d Sequence Number: Fa of an L2CAP message o x0004)	r a complete	L2CAP message with no fragmentation (0x02) + 	
AC Da Da Da CF Blue CI Blue CI Blue CI CI CI CI CI CI CI CI CI CI CI CI CI	<pre>ccess Address tta Header: (000 = 1 = 1.0 = 000 = 10 = 000 = 0 1000 = 0 10000 = 0 10000 = 0 10000</pre>	s: 0x84d81175 0x080a RFU: 0 More Data: F. Sequence Num Next Expecte LLD: Start C RFU: 0 LLD: Start C RFU: 0 Length: 8 Protocol e Protocol (0 Dute Protocol c Sulte Notific c	alse ber: True d Sequence Number: Fa of an L2CAP message o x0004) ication (0x1b) 01 0d 2b 2d 85 99 00	00mII 76 .u		

What this 4-byte 'Bluetooth Attribute Protocol' packet is actually saying is that attribute 0x001C (the location of the TX characteristic in the attribute table) has been updated, and the new value is '0x54', which corresponds to the letter 'T'.

Scrolling a bit further down we can see an example where more than one character was sent in a single transction ('te' in this case):

Image: Source Destination Source Destination Source Destination Source Destination Source Source Display /255.//961 bilave Master Source Slave Socrest Slave Socrest Socrest	Expression C Protocol Ler LE LL ATT LE LL LE LL	20 Empty PUU 26 Empty POU 35 Revd Handle Value Notification, Handle: 0x001c 26 Empty POU 26 Empty POU 26 Empty POU 35 Revd Handle Value Notification, Handle: 0x001c 26 Empty POU 26 Empty Andle Value Notification, Handle: 0x001c 27 Empty POU 26 Empty POU
Bitter Source Destination 3>b1_y 4253.//y81 JaVE MaStEr Slave 3>b21_y 4253.//y81 JaVE MaStEr Slave 35620 2855.827245 Master Slave Slave 55621 2855.827245 Master Slave Slave 55622 2855.87881 Master Slave Master 55622 2855.97485 Master Slave Slave 55622 2855.97459 Slave Master Slave 55622 2855.07489 Slave Master Slave 55622 2855.07305 Master Slave Slave 55622 2856.07305 Master Slave Slave 55622 2856.07305 Master Slave Slave 55622 2856.07305 Master Slave Slave 55632 2856.120200 Master Slave Master 55632 2856.21800 Master Slave Master 55632 2866.21800 Master Slave 55632 2866.21800 Master	Expression C Protocol Ler LE LL ATT LE LL LE LL	Clear Apply Save might Info 20 Empty PUU 26 Empty PDU 35 Rcvd Handle value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rcvd Handle value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Empty PDU 20 Empty P
Time Source Destination >>>b1 2535.7/96151aVE Master >>>b502 2655.8/226 Master Slave S5621 2655.8/226 Master Slave S5621 2655.8/226 Master Slave S5622 2655.8/264 Stave Master S5622 2655.8/264 Stave Master S5622 2655.9/245 Stave Master S5622 2655.9/245 Stave Master S5622 2656.0/2432 Slave S5623 2656.1/2020 Master Slave S5631 2656.1/2020 Master Slave S5632 2566.1/2020 Master	Protocol Ler LE LL ATT LE LL LE LL	ngth Info 26 Empty PDU 26 Empty PDU 35 Revd Handle value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Empty PDU 20 Empty P
>>bit >>bit Waster >>b502 2855.2728 Master Slave >55621 2855.82728 Master Slave >55621 2855.8788L Master Slave >55621 2855.8788L Master Slave >55622 2855.88012 Slave >55622 2855.88012 Slave >55624 2855.97489 Master >55625 2856.02266 Master Slave >55626 2856.02432 Slave >55627 2856.02432 Slave >55627 2856.02432 Slave >55627 2856.02432 Slave >55627 2856.02432 Slave >55629 2856.02432 Slave >55631 2856.12020 Master >55632 2856.12081 Master >55632	LE LL LE LL ATT LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL	20 Empty PUU 26 Empty POU 35 Revd Handle Value Notification, Handle: 0x001c 26 Empty POU 26 Empty POU 26 Empty POU 35 Revd Handle Value Notification, Handle: 0x001c 26 Empty POU 26 Empty Andle Value Notification, Handle: 0x001c 27 Empty POU 26 Empty POU
55621 2855. 82924 51ave Master 55622 2855. 87881 Master Slave 55622 2855. 87881 Master Slave 55624 2855. 97459 Slave Master 55624 2855. 97459 Slave Master 55625 2855. 97459 Slave Master 55626 2856. 02432 Slave Master 55626 2856. 02432 Slave Master 55626 2856. 02432 Slave Master 55627 2856. 02432 Slave Master 55628 2856. 07432 Slave Master 55639 2856. 07432 Slave Master 55631 2856. 12050 Master Slave 55631 2856. 12051 Master Slave 55632 2856. 12060 Master Slave 55632 2856. 12081 Slave Master Frame 55627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetoth Low Energy Link Layer Access Address: 0x4641175	LE LL ATT LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL ATT 	26 Empty PDU 35 Revd Handle value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Revd Handle value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Revd Handle value Notification, Handle: 0x001c
55622 22855.87881 Master 55623 2855.88012 Slave 55624 2855.97345 Master 55627 2856.02266 Master 55627 2856.02325 Slave 55627 2856.07305 Master 55627 2856.07305 Master 55627 2856.07305 Master 55637 2856.12020 Master 55632 2856.21800 Master 55632 2856.21800 <td< td=""><td>LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL ATT</td><td>26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rxvd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Fampty PDU 29 Fampty PDU 20 Fampty PD</td></td<>	LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rxvd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Fampty PDU 29 Fampty PDU 20 Fampty PD
Si622 2855.88012 Elave Master Si624 2855.9745 Master Slave Si625 2855.9745 9 slave Master Si626 2856.02266 Master Slave Si627 2856.02432 Elave Master Si628 2856.07305 Master Slave Si629 2856.0732 Elave Master Si629 2856.0732 Elave Master Si631 2856.12020 Master Slave Si631 2856.12155 Slave Master Si632 2856.12000 Master Slave Si631 2856.12020 Master Slave Si632 2856.21801 Master Slave Si632 2856.21801 Master Slave Si632 2856.21801 Master Slave Frame Si627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetoth Low Energy Link Layer Access Address: Ox46412175	LE LL LE LL LE LL ATT LE LL LE LL LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Empty PDU 20 Empt
55624 2855, 07345 Naster 55625 2855, 07365 Naster 55626 2856, 02266 Master 55627 2855, 07305 Naster 55627 2856, 02266 Master 55627 2856, 02325 Naster 55627 2856, 07305 Naster 55627 2856, 07305 Naster 55627 2856, 07305 Naster 55637 2856, 12020 Master 55637 2856, 12020 Master 55637 2856, 12020 Master 55637 2856, 12050 Slave 55637 2856, 12050 Slave 55637 2856, 12050 Slave 55637 2856, 12050 Slave 55637 2856, 12050 Nater 55637 5850, 13981 Nater 55637 5850, 13981 Nater 5637 5850, 13981 Nater 5637 5850, 13981 Nater 5637 5850, 13981	LE LL LE LL ATT LE LL LE LL LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 26 Empty PDU 37 Revd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 25 Revd Handle Value Notification, Handle: 0x001c
SS622 2855, 07/489 Slave Master SS622 2856, 02/266 Master Slave SS627 2856, 02/265 Slave Master SS628 2856, 07/325 Slave Master SS630 2856, 12/2020 Master Slave SS631 2856, 12/2020 Master Slave SS631 2856, 12/2020 Master Slave SS631 2856, 12/2020 Master Slave SS632 2856, 12/2020 Master Slave SS632 2856, 12/2020 Master Slave SS632 2856, 12/2021 Master Master Frame SS627: 35 bytes on wire (280 bits), 35 bytes Slove Slove Slove Slove Sordic RLE solffer meta Bluetoth Low Energy Link Layer Access Address: 0x46/1275 Slove	LE LL ATT LE LL LE LL LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 27 Empty PDU 28 Empty PDU 29 Empty PDU 20
55626 2836.02266 Master Slave 55627 2856.02325 Slave Master 55628 2856.07305 Master Slave 55629 2856.07305 Master Slave 55629 2856.07305 Master Slave 55629 2856.07325 Slave Master 55631 2856.12020 Master Slave 55632 2856.12020 Master Slave 55632 2856.12000 Master Slave 76740 Premes Slave 76740 2460 Master 76740 Premes Slave 810etcoth Low Energy Lifk Layer Access Address: Not4612175	LE LL ATT LE LL LE LL LE LL LE LL LE LL ATT	26 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 25 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c
S5627 Z856,02432 Slave Master S5628 Z856,07305 Master Slave S5628 Z856,07305 Master Slave S5630 Z856,12020 Master Slave S5631 Z856,12155 Slave Master S5632 Z856,12020 Master Slave S5631 Z856,12020 Master Slave S5632 Z856,13080 Master Slave Frame S10000 Master Slave Frame S5627 35 Bytes on wire (280 bits), 35 bytes Frame S1000000000000000000000000000000000000	ATT LE LL LE LL LE LL LE LL LE LL ATT	35 Rcvd Handle Value Notification, Handle: 0x001c 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c
55628 2856.07305 Master Slave 55629 2856.07325 Slave Master 55630 2856.12020 Master Slave 55631 2856.12155 Slave Master 55632 2856.12000 Master Slave 6100 Master Slave 6100 Master Slave 8100 Master Slave 8100 Master Slave 8100 Master Slave Access Address: Not4d8l175	LE LL LE LL LE LL LE LL ATT	26 Empty POU 26 Empty POU 26 Empty POU 26 Empty POU 26 Empty POU 35 Excvd Handle Value Notification, Handle: 0x001c
55629 2856.07432 Parter 55630 2856.12020 Master Slave 55631 2856.12020 Master Slave 55631 2856.12020 Master Slave 55632 2856.21800 Master Slave 55631 2856.2180 Master Master Frame 55637 355 Master Frame 55627 35 bytes on wire (280 bits), 35 bytes Frame 55627 135 bytes on wire (280 bits), 35 bytes Bluetooth Low Energy Link Layer Access Address: 0x8461175	LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Revd Handle Value Notification, Handle: 0x001c
55630 2856.12020 Master Slave 55631 2856.12155 Slave Master 55632 2856.21800 Master Slave Frame 55627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetoth Low Energy Link Layer Access Address: 0x84d81175	LE LL LE LL LE LL ATT	26 Empty PDU 26 Empty PDU 26 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c
5561 2856.12155 5lave Master 5563 2856.21800 Master Slave 5563 2856.21801 Slave Master 771 077 2170 100 Master 1 Frame 55627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetooth Low Energy Link Layer Access Address: 0x84681275	LE LL ATT	26 Empty PDU 26 Empty PDU 35 Rcvd Handle value Notification, Handle: 0x001c
55632 2856.21800 Master Slave 55633 2856.21981 Slave Master Frame 55627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetooth Low Energy Link Layer Access Address: 0x84d81175	LE LL ATT	26 Empty PDU 35 Rcvd Handle Value Notification, Handle: 0x001c
55633 2856, 21981 Slave Master Frame 55627: 35 bytes on wire (280 bits), 35 byte: Nordic BLE sniffer meta Bluetooth Low Energy Link Layer Access Address: 0X846B175	ATT	35 Rcvd Handle Value Notification, Handle: 0x001c
Frame 55627: 35 bytes on wire (280 bits), 35 bytes Nordic BLE sniffer meta Bluetooth Low Energy Link Layer Access Address: 0x84d81175		37 Participanti
Frame 55627: 35 bytes on wire (280 bits), 35 byte: Nordic BLE sniffer meta Bluetooth Low Energy Link Layer Access Address: 0x84d81175	s captured (280	D bits) on interface O
0 = More Data: False 0 = Sequence Number: False 1 = Next Expected Sequence Number: Tr 10 = LLID: Start of an L2CAP message or 000 = RFU: 0 0101 = Length: 9		2CAP message with no fragmentation (0x02)
Bluetooth L2CAP Protocol		
Length: 5		
CID: Attribute Protocol (0x0004)		
Bluetooth Attribute Protocol Opcode: Handle Value Notification (0x1b) Handle: 0x001c		
Value: 7465		
000 11 00 1- 01 0J 40 00 0- 01 10 2L 42 05 07 00		
000 11 06 1c 01 8d 49 06 0a 01 10 2b 42 85 97 00 010 00 75 11 d8 84 06 09 05 00 04 00 1b 1c 00 74 020 18 fc 6a		
Value (btatt.value), 2 bytes Packets: 64085 - Di	Profile: Default	

The results of this transaction in the nRF UART application can be seen below:



Closing Wireshark and nRF-Sniffer

When you're done debugging, you can save the session to a file for later analysis, or just close Wireshark right away and then close the nRF-Sniffer console window to end the debug session.

Moving Forward

A sniffer is an incredibly powerful and valuable tool debugging your own hardware, reverse engineering existing BLE peripherals, or just to learn the ins and outs of how Bluetooth Low Energy actually works on the a packet by packet level.

You won't learn everything there is to know about BLE in a day, but a good book on BLE, a copy of the Bluetooth 4.1 Core Specification and a sniffer will go a long way to teaching you most of the important things there is to know about BLE in the real world.

Using with Sniffer V2 (old)

This page is deprecated. It is being left here for reference and for anyone requiring a Python 2 setup.

In mid 2018, Nordic release new Bluetooth LE sniffer firmware - this firmware works way better with Wireshark.

As of August 2018 we are only selling Sniffers pre-prorgrammed with Firmware version 2

If you have a firmware V1 (packaging doesn't say firmware V2, or you bought before August 2018) see the previous sections!

Nordic User Manual

You can grab the 'official' user manual from Nordic at https://www.nordicsemi.com/ eng/Products/Bluetooth-low-energy/nRF-Sniffer (https://adafru.it/C68) we include a mirror of the v2.1 instructions below

nRF_Sniffer_User_Guide_v2.1.pdf

https://adafru.it/C69

nRF Sniffer V2 Multi-Target Application

When downloading the desktop/Wireshark Sniffer tool, make sure to download BETA 1, which matches the firmware image below. You can select this version via the drop down selector shown below:

Changelog:	
• 2.0.0-beta-1 Multi-Platform	~
2.0.0-beta-2 Multi-Platform	~
2.0.0-beta-3 Multi-Platform	~

For convenience sake, the extcap folder contents for BETA 1 are also available for download here using the button below. See the user guide on how to install this in the correct location for Wireshark.

extcap.zip (BETA 1)

https://adafru.it/DvS

V2 Firmware

You need a J-Link or other SWD programming jig in order to install/change the sniffer firmware!

If by chance you have an nRF51822 board you want to load the firmware on, here's a hex that does not require the 32khz crystal (but does require the 16 mhz crystal)

This firmware is from the nrf_sniffer_2.0.0-beta-1_51296aa package



https://adafru.it/C6a

Again, we don't have a guide or tutorial on loading the firmware onto an nRF51. We don't have the original source code, that hex is from Nordic and they only release hex files

V2 Wireshark Usage (old)

This page is deprecated. It is being left here for reference and for anyone requiring a Python 2 setup.

For the V2 firmware, its recommended that you use Wireshark - the V1 had various methods such as a Python API but really they were all mediocre compared to the abilities of Wireshark



Install Wireshark

Start by installing Wireshark, a great cross-platform monitoring tool

Visit <u>https://www.wireshark.org/</u> (https://adafru.it/eDB) and download the latest version of Wireshark for your operating system

When installing on windows, check the box to also install WinPcap

Install Wireshark Plugin

Next up, you'll need the Nordic plugin software. We need to work with a specific release - 2.0.0 Beta 1. For convenience sake, the extcap folder contents for BETA 1 are available for download using the button below.



Download that zip file to your computer and see below for how to install the files into the necessary Wireshark folder.

For complete reference, the Nordic main page for the plugin software is here:

nRF Sniffer Software (reference only)

https://adafru.it/KLE

If you decide to go there, be sure to select the correct version for download.

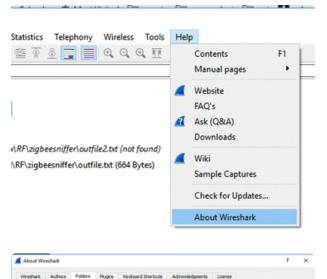
Changelog:	
2.0.0-beta-1 Multi-Platform	\sim
2.0.0-beta-2 Multi-Platform	~
2.0.0-beta-3 Multi-Platform	\sim

WARNING this file is huge - over 200MB! We suggest just using the extcap.zip file linked above.

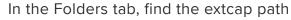
Now to find the Wireshark folder location to unzip these files into.

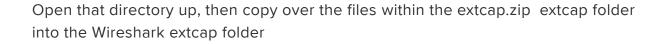
ical Dis

MaxMind DB database



Open Wireshark, in the Help menu select About wireshark





In the end, your Wireshark/extcap directory should contain nrf_sniffer.bat, nrf_sniffer. py and SnifferAPI folder.

File Home Share View		~ 0		
← → × ↑ 📙 « Local Disk (C:) → Progra	m Files → Wireshark → extcap v Č Search extcap	۹		
extcap ^ Name	Date modified Type	Size		
pCBs	This folder is empty.			
Dropbox (adafruit)	🔜 I 🕑 📴 🖛 I extcap		- 0)
1	File Home Share View			\sim
SF Dropbox (Personal	↓ · · · · · · · · · · · · · · · · · · ·	p ~ 0	Search extcap	۶
Computer	> Move to extcap ^ Name	Date modified	Туре	Siz
3D Objects	A Quick access			512
E Desktop	Desktop # SnifferAPI	8/11/2018 2:07 PM	File folder	
Documents	I three a state of the state of	8/11/2018 2:07 PM	Windows Batch File	
Downloads	Development x	8/11/2018 2:07 PM	PY File	
h Music	 Fritzing-Libra * 			
Fictures				
	🛃 Common 🖈			
Videos	🐉 Dropbox (ada 🖈			

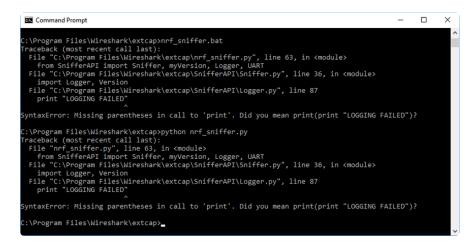
Now quit Wireshark so we can get things tested!

Dealing With Python 2 vs 3

Nordic's sniffer code is Python 2 only, so if you have Python 3 your default (which, by now, you probably do) you'll need to install Python 2.

The best way to test is to go to the extcap directory in your terminal software and try running nrf_sniffer.bat (Windows) or python nrf_sniffer.py (Mac/Linux)

If you get the error on the print "LOGGING FAILED" line



or the "No module named 'Logger'" error



Then you'll need to trick the sniffer software into using Python 2

For Windows, at least, I installed Python 2.7 into C:\Python27 (the default) and then edited the nrf_sniffer.bat file to say:

```
@echo off
C:\Python27\python "%~dp0nrf_sniffer.py" %*
```

note the explicit path!

Installing Dependancies

Once you get past that part, you can try rerunning the bat/py script and you may get other missing module errors like No module named serial



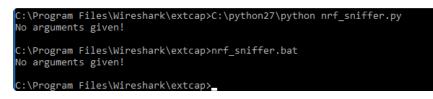
You'll need to install these with pip

Warning! Because you have to use Python2 here, make sure you're using pip2 or on windows, use the full path C:\python27\Scripts\pip2.exe

e.g. C:\python27\Scripts\pip2.exe install pyserial

C:\Program Files\Wireshark\extcap>C:\python27\Scripts\pip2.exe install pyserial
Collecting pyserial
Cache entry deserialization failed, entry ignored
Cache entry deserialization failed, entry ignored
Cache entry deserialization failed, entry ignored
Downloading https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3
915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193kB)
100% ###################################
Installing collected packages: pyserial
Successfully installed pyserial-3.4
Cache entry deserialization failed, entry ignored
You are using pip version 9.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip installupgrade pip' command.
C:\Program Files\Wireshark\extcap>

Eventually you'll get No arguments given! which means the script is, at least, fully running



Test Capture

OK finally once that works, start Wireshark again.

This time you'll see the nRF Sniffer capture device!

Welcome to Wireshark	
Open	
C:\Users\ladyada\Dropbox\RF\zigbeesniffer\outfile2.txt (not found)	
C:\Users\ladyada\Dropbox\RF\zigbeesniffer\outfile.txt (664 Bytes)	
Capture	
using this filter: 📙 Enter a capture filter	All interfaces shown 🔻
Local Area Connection 4 J.	
nRF Sniffer COM63	

Double Click on that line to start the Capture!

Now go here to learn how to look at BLE packets with Wireshark

https://adafru.it/C61

Windows Install Supplemental Information

Adafruit forums user @TomHildebrand put together a nice write up on their experience installing and setting up everything on Windows 10. It's generally the same

info as above, but may have some more explicit info that is useful. Checkout it out here:

Adafruit BLE Sniffer installation on Windows Step-by-Step

https://adafru.it/eca

Using with Sniffer V1 (old)

The original Bluetooth LE sniffer firmware from Nordic had some restrictions such as only being usable by Wireshark 1.

As of August 2018 we are only selling Sniffers pre-prorgrammed with Firmware version 2

However, we'll keep this documentation up in case its useful for people with old boards

You need a J-Link or other SWD programming jig in order to install/change the sniffer firmware!

If by chance you have an nRF51822 board you want to load the firmware on, here's a hex that does not require the 32khz crystal (but does require the 16 mhz crystal)



https://adafru.it/C4k

Again, we don't have a guide or tutorial on loading the firmware onto an nRF51. We don't have the original source code, that hex is from Nordic and they only release hex files

USB Driver Install

CP2104 Driver Requirements (Black Boards)

The latest version of the sniffer uses the CP2104 USB to Serial bridge and drops the SWD connector, allowing us to sell the boards at a significant discount compared to version 1.0. To use these boards, though, you will need to install the <u>CP2104 VCP</u> driver from Silicon Labs (https://adafru.it/vrf):

S	简体中文 日本語	Log In Register
SILICON LABS	Parametric Search Cro	ss-Reference Search
About 🔻 Products 👻 Solutions 👻 Community & Support 👻	Search silabs.com	GO
Silicon Labs » Products » Development Tools » Software » USB to UART Bridge VCP Drivers		
CP210x USB to UART Bridge VCP Drivers		
The CP210x USB to UART Bridge Virtual COM Port (VCP) drivers are required for device operation as a Virtual COM Port t products. These devices can also interface to a host using the direct access driver. These drivers are static examples det Communications Guide for the CP210x, download an example below:		
AN197: The Serial Communications Guide for the CP210x		
Download Software		
The CP210x Manufacturing DLL and Runtime DLL have been updated and must be used with v6.0 and later of the CP211 Software downloads affected are AN144SW.zip, AN205SW.zip and AN223SW.zip. If you are using a 5.x driver and need su Application Note Software.		
Legacy OS software and driver package download links and support information >		

FTDI Driver Requirements (Blue Boards)

Before you can start talking to the sniffer, you'll need to install a standard FTDI driver for the FT231x located on the device.

Find the appropriate FTDI VCP installer on the FTDI Driver Download Page (https://adafru.it/aJv), install it on you system, and then insert the sniffer in any USB port on your system.

Currently Supported	VCP Drivers:								
			Processor Archit	ecture					
Operating System	Release Date	x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4	Comments
Windows*	2014-09-29	Available as se Contact support1@ftdichip.com driv	if looking to create cusomised	-	-	-		-	2.12.00 WHQL Certified Available as <u>setup executable</u> <u>Release Notes</u>
Linux	2009-05-14	1.5.0	1.5.0	+		-			All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0- 19 Refer to <u>TN-101</u> if you need a custom VCP VID/PID in Linux
Mac OS X	2012-08-10	2.2.18	2.2.18	2.2.18		1.1			Refer to TN-105 if you need a custom VCP VID/PID in MAC OS

V1 Sniffer Software

This page is for the V1 Sniffer firmware only! If you have V2, check the other page - the process has changed between versions.

Using the Firmware V1 Sniffer

There are currently two ways to use the sniffer:

Nordic's nRF Sniffer Utility (Windows only)

If you are on Windows, the best user experience will be had by using the official Nordic nRFSniffer application, available as a download from Nordic Semiconductors after creating a 'My Pages' account, and registering your device using the product ID located on the Bluefruit LE Sniffer packaging.

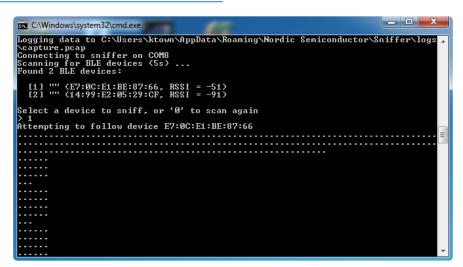
More information on using Nordic's nRF Sniffer application (https://adafru.it/k6F).

arrow keys [#] or ENTER e w v. c u b a p b a p o h s u	Start Wireshark, Exit Display filter: Display filter: Remove display f Passkey entry 00B key entry 00B key entry Define new adv h Get support Launch User Guid Re-program firmw	ice list. Use EN to sniff from li sniffer will onl the primary vie Nearest devices Nearest devices ilter. op sequence. e (pdf)	TER to select.	
# public	name	RSSI	device address	
[]0"" []1""			14:99:e2:05:29:cf 68:48:98:b8:e5:2b e4:c6:c7:31:95:11	public

Python API (Cross-Platform, no Registration)

If you are not using Windows, or don't wish to create a MyPages account, the alternative is to use a Python interface to communicate with the nRFSniffer firmware, which will log any traffic to a libpcap file that can be opened directly in Wireshark. This has been tested on OS X 10.10, Ubuntu 14.04 and Windows 7, but it currently doesn't support streaming data directly into Wireshark via named pipes (though this is possible with some platform-specific effort).

More information on using the Python API (https://adafru.it/k7a).



V1 Nordic nRF Sniffer

This page is for the V1 Sniffer firmware only! If you have V2, check the other page - the process has changed between versions.

The following guide will walk you through downloading, installing and using the official nRF Sniffer application for Nordic Semiconductors.

Getting the Sniffer Utility

The Bluefruit LE Sniffer comes pre-flashed with the special sniffer firmware image, but you'll need to go to Nordic's website and download the nRF-Sniffer package to capture the data on Windows and push it out into Wireshark for packet by packet analysis.

Go to the nRF Sniffer product page downloads tab (https://adafru.it/Dsm), then download the latest V1 version of the utility, and unzip it.

Inside this downloaded file you'll find the sniffer executable, which will open up the command-line tool when you click on it.

Getting Wireshark

In order to use the sniffer utility you'll also need to <u>download Wireshark</u> (https:// adafru.it/ecp), preferably verison 1.12.1 (the same one used in this tutorial). You may need to explore the download mirrors, such as https://l.na.dl.wireshark.org/ (https://adafru.it/C3w) to find the download link since they dont have a direct v1 link

Simply select the 32-bit or 64-bit Windows Installer and install it on your machine using the default settings:



Make sure that you install the libpcap library when installing Wireshark. Any log files captured by the python library are in libpcap format, and will require this library to work.

Running the Sniffer

Now that everything is installed, you can get started using the Bluefruit LE Sniffer and the sniffer bridge SW that pushes any sniffed data out into Wireshark ...

Select the Sniffer Target

The nRF-Sniffer can only sniff one device at a time, so the first step is getting the sniffer running and then selecting the device that you want to debug.

Start nRF-Sniffer by running the ble-sniffer_win executable (for example: ble-sniffer_win_1.0.1_1111_Sniffer.exe).

This will try to detect the device running the nRF-Sniffer firmware over a UART COM port.

If the board isn't detected right away type 'f' to erase any previous com port settings, or try removing and then re-inserting the sniffer while the console application is running.

Once the sniffer is found, you should see a list of all BLE devices that were detected in listening range:

If you see a warning in the application about your firmware being out of date and requesting to update it, IGNORE THE WARNING. The Adafruit boards run a slightly modified version of the sniffer firmware, which causes the tool to think it is out of date.

BLE Sniffer 1.0.1				_ D X
Commands: 1 arrow keys [#] or ENTER e w x/q c v b a p o h s u CTRL-R Available devi	Select a device Like ENTER, but Start Wireshark, Exit Display filter: Display filter: Display filter: Remove display f Passkey entry OOB key entry OOB key entry Define new adv h Get support Launch User Guid Re-program firmw	ice list. Use EN to sniff from li sniffer will onl the primary vie Nearest devices Nearest devices ilter. op sequence. e (pdf)	TER to select.	nts.
# public	c name	RSSI	device address	
[] 0 "" [] 1 "" [] 2 "" Scanning for de	vices.	-65 dBm	68:48:98:b8:e5:2b	public public random
Sent Key value	to sniffer			Ŧ

In this particular case, we'll select device number 2, which is a BLEFriend running the standard UART firmware.

Type the device number you want to sniffer (in this case '2'), and you should see the device highlighted in the list, similar to the image below:

💽 BLE Sniffer 1.0.1				- • ×
arrow keys	List the dev Navigate the Select a dev Like ENTER, Start Wiresh Exit Display filt Display filt Display filt Desplay filt Passkey entr OOB key entr Define new a Get support Launch User	ices available ff device list. Use ice to sniff fro but sniffer will ark, the primary er: Nearest devic er: Nearest devic ay filter. y dv hop sequence.	≥ ENTER to select. n list. only follow advertisem viewer for the sniffer es (RSSI > -50 dBm). es (RSSI > -70 dBm). es (RSSI > -90 dBm).	
Available devid # publid		RSSI	device address	
[] 0 "" [] 1 "" -> [X] 2 "" Sniffing device			14:99:e2:05:29:cf 68:48:98:b8:e5:2b	public

At this point you can type 'w', which will try to open wireshark and start pushing data out via a dedicate pipe created by the nRF-Sniffer utility.

Now go here to learn how to look at BLE packets with Wireshark

https://adafru.it/C61

V1 OS X Support

This page is for the V1 Sniffer firmware only! If you have V2, check the other page - the process has changed between versions.

If you are running OS X 10.9 or higher, you can also use the sniffer on OS X using the nrf-ble-sniffer-osx (https://adafru.it/ft6) package from Roland King. (Make sure you have the latest version, as of 20 June 2015, which is now compatible with the FTDI chip used on the Adafruit board.)

Setup instructions are available on the wiki page (https://adafru.it/ft7) for the project.

Be sure to download Wireshark version 2.0.x NOT the new 2.2.7 that was released June 2017

	usbserial-DN009WNO
Snit	ffer Information
►	Status : Sniffing <unknown></unknown>
	Packet Count: 15 358
	USB Device : FTDI - FT231X USB UART [s/n: DN009WNO]
Cur	rently Sniffing
	Device Name : <unknown></unknown>
	Address : dd:2b:48:7f:b5:8c (random)
	RSSI: -54
	Event Count: 0
Adv	ertisers
	ame: <unknown> iress: 46:8e:86:db:97:c1 (random)</unknown>
	ame : <unknown> iress : dd:2b:48:7f:b5:8c (random)</unknown>
Lis	t Devices Sniff Device Capture to Wireshark Capture To PCAP Enter Passkey

Please note that there can be a long delay (30-60 seconds) before Wireshark shows up using the tool, due to the X11 startup time, etc.

	X	Capturing from /var/folders/86/	b2vp14n5_5_yvdz_z8w9x_c000	0gn/T/BTLEyJUGqJ	88 [Wireshark 1.12.1 (v1.1	2.1-0-g01b65bf from master-1.12)]	
<u>Eile</u>	dit <u>V</u> iew <u>G</u> o <u>C</u>	apture <u>A</u> nalyze <u>S</u> tatist	ics Telephony <u>T</u> ools <u>I</u> r	nternals <u>H</u> elp			
0 6) 📶 🔳 🧕	🖻 🗎 🗙 😂 🔍	🗢 🌳 🤏 🛧 🎚	R. C. 🖭 📓 🕅	🎦 💥 🗮	
Filter:			- Expression	Clear Apply Sa	ave		
No.	Time	Source	Destination	Protocol L	ength Info		
	78 13.758362000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV_IND		
	79 14.302589000	dd:2b:48:7f:b5:8c	 broadcast>	LE LL	56 ADV_IND		
	80 14.303255000 81 14.303919000	dd: 2b: 48: 7f: b5: 8c dd: 2b: 48: 7f: b5: 8c	 dcast> dcast>	LE LL LE LL	56 ADV_IND		
	81 14.303919000	dd: 2b: 48: 7f: b5: 8c dd: 2b: 48: 7f: b5: 8c	<pre><broadcast> <broadcast></broadcast></broadcast></pre>	LE LL	56 ADV_IND 56 ADV_IND		
	83 14.850082000	dd: 2b: 48: 7f: b5: 8c	<pre><broadcast></broadcast></pre>	LE LL	56 ADV_IND		[
	84 14.850746000	dd: 2b: 48: 7f: b5:8c	<pre> droadcast></pre>	LE LL	56 ADV_IND		
	85 15.401586000	dd: 2b: 48: 7f: b5: 8c	<pre>shroadcast></pre>	LE LL	56 ADV_IND		
	86 15.402252000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV IND		
	87 15.402916000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV IND		
	88 15.948595000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV_IND		
	89 15.949262000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV_IND		
	90 15.949926000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV_IND		
-	91 16.499165000	dd:2b:48:7f:b5:8c	 dcast>	LE LL	56 ADV IND		
Nordi ⇒ Bluet Acc ▷ Pac Adb ⇒ Adb ▷ ↓	ic BLE Sniffer Me cooth Low Energy ress Address: OxE ket Header: OxIe vertising Address vertising Data Flags TX Power Level 128-bit Service (Length: 17 Type: 128-bit S	ta Link Layer e89bed6 40 (PDU Type: ADV_IND, T 40 (PDU Type: ADV_IND, T 1: dd:2b:48:7f:b5:8c (dd:	2b:48:7f:b5:8c)	interface 0			
P CR	: 0x47e7e5	ecauc240ee5a5e05515a5050,	004008				
0010 0 0020 0	2 06 31 01 67 41 0 d6 be 89 8e 40 2 0a 00 11 07 9e 5 01 00 40 6e e2	1e 8c b5 7f 48 2b dd 02 ca dc 24 0e e5 a9 e0 93	01 06@H+				
0 10	ustom UUID (btc	ommon.eir Packets: 1	89 · Displayed: 189 (100.0)%)		Profile: Defa	ult

If Wireshark doesn't show up and X11 has been installed correctly, try forcing X11 closed and trying a second time. The startup process can sometimes stall.

V1 Python API

This page is for the V1 Sniffer firmware only! If you have V2, check the other page - the process has changed between versions.

The Python interface requires a custom Wireshark library for Linux. We're currently working on adding support for this. Please use the Windows or OS X utility until the update is available.

Nordic provides a Python API for their sniffer firmware that makes it possible for us to use the sniffer on any platform, and we've put together a basic wrapper for this API to help you get started.

We've tested this wrapper with Python 2.7 on the following platforms:

- OS X 10.10
- Windows 7 x64
- Ubuntu 14.04

To stream live data into Wireshark the way the <u>official Windows app</u> (https://adafru.it/ k6F) from Nordic does you will need to compile a Wireshark utility that creates a name pipe that data gets pushed through.

To keep things simple, though, you can also just log sniffed traffic directly to a libpcap file, which can be opened directly in Wireshark when you are done, which is the easiest solution and what we'll be demonstrating here:

C:\Windows\system32\cmd.exe
Logging data to C:\Users\ktown\AppData\Roaming\Nordic Semiconductor\Sniffer\logs \capture.pcap Connecting to sniffer on COM8 Scanning for BLE devices (5s) Found 2 BLE devices:
[1] "" (E7:0C:E1:BE:87:66, RSSI = -51) [2] "" (14:99:E2:05:29:CF, RSSI = -91)
Select a device to sniff, or '0' to scan again > 1
Attempting to follow device E7:0C:E1:BE:87:66
······

Requirements

To use the example we provide for the Python API, you will require the following utilities:

- Python 2.7.x (https://adafru.it/edH) (we tested with 2.7.6)
- pySerial (https://adafru.it/cLU)

If you're new to Python and pySerial, have a look at our <u>Instaling Python and PySerial</u> (https://adafru.it/k7c) guide by Simon Monk.

Download the API

Once you have Python and pySerial installed on your system, you will need to download a copy of the Python API.

The latest version of the API is always <u>available on Github</u> (https://adafru.it/edJ), but you can also download a .zip file of the latest code directly using the button below:

Download the Python API from Github

https://adafru.it/edK

Unzipping the file should give you a file structure resembing the image below:

] SnifferAPI	28/11/2014 15:34	Dossier de fichiers	
Wireshark_dissector_source	27/11/2014 09:29	Dossier de fichiers	
.DS_Store	26/11/2014 22:05	Fichier DS_STORE	7 Ko
API Manifest.txt	27/11/2014 09:29	Document texte	1 Ko
ocumentation.html	27/11/2014 09:29	Chrome HTML Do	12 Ko
尾 example.py	26/11/2014 22:20	JetBrains PyCharm	2 Ko
LICENSE.txt	27/11/2014 09:29	Document texte	2 Ko
🔁 Nordic Semiconductor Sniffer API Guide	27/11/2014 09:29	Adobe Acrobat D	468 Ko
readme.md	27/11/2014 09:29	Fichier MD	2 Ko
尾 sniffer.py	28/11/2014 15:48	JetBrains PyCharm	7 Ko
sniffer_uart_protocol.xlsx	27/11/2014 09:29	Feuille de calcul	21 Ko

Using the sniffer.py Wrapper

To help you get started, we've made an easy to use wrapper called sniffer.py:

It takes a single argument, the COM port location, which will be something like 'COM15' on Windows, '/dev/ttyACM*' on Linux, or '/dev/tty.usbserial*' on OS X.

Linux

To run the sniffer wrapper on Linux, enter the following command (changing the serial port as necessary):

```
$ sudo python sniffer.py /dev/ttyACM0
```

OS X

To run the sniffer wrapper on OS X, enter the following command (changing the serial port as necessary):

```
$ python sniffer.py /dev/tty.usbserial-DN009MP6
```

Windows

To run the sniffer wrapper on Windows, enter the following command (changing the serial port as necessary):

```
You can find the serial port used by the Bluefruit LE Sniffer by opening the
Device Manager on your system and looking in the 'Ports' category:
```

```
python sniffer.py COM30
```

Scanning for Devices

If the wrapper was able to connect to the Bluefruit LE Sniffer, it will perform a 5 second scan for Bluetooth Low Energy devices in range, and ask you which device you want to listen to:

```
$ sudo python sniffer.py /dev/ttyACM0
[sudo] password for ktown:
Logging data to logs/capture.pcap
Connecting to sniffer on /dev/ttyACM0
Scanning for BLE devices (5s) ...
Found 2 BLE devices:
  [1] "" (E7:0C:E1:BE:87:66, RSSI = -52)
[2] "" (14:99:E2:05:29:CF, RSSI = -94)
Select a device to sniff, or '0' to scan again
>
```

Once you select a device, it will start scanning that specific device, and you will see an update every second of the number of packets 'sniffed' from the device (where each '.' represents a packet):

```
Select a device to sniff, or '0' to scan again
> 1
Attempting to follow device E7:0C:E1:BE:87:66
```

Locating the Log File

Once you've sniffed enough data, simply type CTRL+C to stop, and locate the libpcap log file at the path mentionned by the tool. This will normally be:

- Windows: 'C:\Users\ktown\AppData\Roaming\Nordic Semiconductor\Sniffer\logs \capture.pcap' (this will of course change based on your username)
- OS X/Linux: 'logs/capture.pcap' (relative to the location of the Python API)

Analyze Data in Wireshark

At this point, you simply need to open the capture.pcap file in Wireshark, and you can analyze the sniffed data!

The image below shows an advertising packet from a factory default Bluefruit LE Friend (https://adafru.it/edl) board:

-		SHOR THEFT (FAILER	-0-g01b65bf from master-1.12	2)]				
-	Edit View Go	<u>Capture</u> <u>Analyze</u>	Statistics Telephony Io	ols Internals <u>H</u> elp				
D	o ք 🔳 🙇	🖹 🗎 🗶 🔁	। 🔍 🗢 🔶 🐴 :		0, 0, 🖸 👹 🖻	🕵 % 😫		
ilter:				 Expression 	Clear Apply Save			
0.	Time	Source	Destination	Protocol L	ength Info			
	228 13.34712		Master	LE LL	60 ADV_IND			
	229 13.34844		Master	LE LL	60 ADV_IND			
	230 13.89326		Master	LE LL	60 ADV_IND			
	231 13.89627		Master	LE LL	60 ADV_IND			
	232 13.89870 233 14.44721		Master	LELL	60 ADV_IND			
	233 14.44721		Master	LE LL	60 ADV_IND 60 ADV_IND			0
	235 14.45134		Master	LE LL	60 ADV_IND			
	236 14.99882		Master	LE LL	60 ADV_IND			
		+ Stave	Hubeer		OU NOT_IND			
_	222. 60	horizon an ordere	(100 bins) (0 bins		hda - N			
	ame 232: 60 rdic BLE sn		(480 bits), 60 byte	s captured (480	Dits)			
		Energy Link La	wer					
		ess: 0x8e89bed						
			, J Type: ADV_IND, TXA	dd=false, RxAdd=	false)			
			:e1:be:87:66 (e7:0c					
	Advertising							
ļ	000 0 11 	= Simultaned = BR/EDR Not I. = LE General .0 = LE Limited Level 2 x Power Level (evel (dBm): 0 ervice Class U	Dus LE and BR/EDR to Dus LE and BR/EDR to Supported: true (0 Discoverable Mode: Discoverable Mode: (0x0a)	Same Device Cap x01) true (0x01))	
	Custom (28-bit Service UUID: 9ecadc240	Class UUIDs (0x07) Dee5a9e093f3a3b50100	406e				
	Type: 12 Custom C CRC: 0xf9229 E [Expert In	28-bit Service UUID: 9ecadc240 9f nfo (Chat/Proto		406e				
	Type: 1: Custom 0 CRC: 0xf922 E [Expert In [correct	28-bit Service UUID: 9ecadc240 9f nfo (Chat/Proto t]	0ee5a9e093f3a3b50100 0col): correct]	406e				
	Type: 1: Custom (CRC: 0xf9229 Expert In [correct [Severit	28-bit Service UUID: 9ecadc240 9f nfo (Chat/Proto t] ty level: Chat	0ee5a9e093f3a3b50100 0col): correct]	406e			_	
	Type: 1: Custom (CRC: 0xf9229 Expert In [correct [Severit	28-bit Service UUID: 9ecadc240 9f nfo (Chat/Proto t]	0ee5a9e093f3a3b50100 0col): correct]	406e				
	Type: 1: Custom (CRC: 0xf922) Expert I [Correct [Severin [Group: 08 06 35 (28-bit Service JUID: 9ecadc24(9f nfo (Chat/Proto 1] ty level: Chat Protocol] 11 dd Ob 06 0a	0ee5a9e093f3a3b50100 0col): correct]	02 00	- '4 			

Note that the utility will start sniffing data as soon as you connect to the Bluefruit LE Sniffer, so early packets in the log file might contain advertising packets from other devices in range. It will only start filtering packets once you select a specific device via the selection dialogue.

For information on how to use Wireshark, have a look at the notes on the official nRF Sniffer utility (https://adafru.it/k6F), which describes some of the packet types you might encounter working with Bluetooth Low Energy.

FAQs

I'm using the V2 (BETA 1) firmware, but can't seem to connect in Wireshark?

There are a number of possible issues here, mostly revolving around the fact that the system depends on a Python script piping data into Wireshark. You may find the following post from a user help to try to debug this, the cause being a potential conflict with multiple instances of Python on your system: https://forums.adafruit.com/viewtopic.php?f=53&t=146215#p726333 (https://adafru.it/eca)

When I connect to a Central device, I don't see any connection data, but when I disconnect I see the advertising packets again. How do I capture data with a connected peripheral?

This is a limitation of the sniffer firmware from Nordic. Advertising in Bluetooth Low Energy happens on three dedicated channels, each running at it's own frequency. For the sniffer to 'follow' the connection it needs to be looking at the right channel when the connection happens, and there is a 2/3 chance that it is looking at another channel at any given moment.

To capture the connection and see data exchanges post connection, you may need to connect several times until the channels are aligned between the sniffer and the BLE peripheral+central devices.

How do I convert between Sniffer and Bluefruit LE firmware using SWD?

Reflashing Bluefruit LE modules over SWD (ex. switching to the sniffer firmware and back) is at your own risk and can lead to a bricked device, and we can't offer any support for this operation! You're on your own here, and there are unfortunately 1,000,000 things that can go wrong, which is why we offer two separate Bluefruit LE Friend boards -- the sniffer and the normal Bluefruit LE Friend board with the non-sniffer firmware, which provides a bootloader with fail safe features that prevents you from ever bricking boards via OTA updates.

AdaLink (SWD/JTAG Debugger Wrapper)

Transitioning between the two board types (sniffer and Bluefruit LE module) is unfortunately not a risk-free operation, and requires external hardware, software and know-how to get right, which is why it isn't covered by our support team.

That said ... if you're determined to go down that lonely road, and you have a Segger J-Link (https://adafru.it/fYU) (which is what we use internally for production and development), or have already erased your Bluefruit LE device, you should have a look at AdaLink (https://adafru.it/fPq), which is the tool we use internally to flash the four files required to restore a Bluefruit LE module. (Note: recent version of AdaLink also support the cheaper STLink/V2 (http://adafru.it/ 2548), though the J-Link is generally more robust if you are going to purchase a debugger for long term use.)

To go from the sniffer to Bluefruit LE firmware the mandatory Intel Hex files are available in the Bluefruit LE Firmware repo (https://adafru.it/edX). You will need to flash:

- An appropriate bootloader image
- An appropriate SoftDevice image
- The Bluefruit LE firmware image
- The matching signature file containing a CRC check so that the bootloader accepts the firmware image above (located in the same folder as the firmware image)

The appropriate files are generally listed in the version control .xml file (https://adafru.it/fPr) in the firmware repository.

If you are trying to flash the sniffer firmware (at your own risk!), you only need to flash a single .hex file, which you can find here (https://adafru.it/fYV). The sniffer doesn't require a SoftDevice image, and doesn't use the fail-safe bootloader -- which is why changing is a one way and risky operation if you don't have a supported SWD debugger.

Adafruit_nF51822_Flasher

We also have an internal python tool available that sits one level higher than AdaLink (referenced above), and makes it easier to flash specific versions of the official firmware to a Bluefruit LE module. For details, see the Adafruit_nRF51822_Flasher (https://adafru.it/fVL) repo.

Why isn't the Firmware V1 plugin working in Wireshark?

The Sniffer Firmware V1 plugin was written for Wireshark 1.12.x and won't work with older versions of the tool or the new 2.x family. Be sure to download an appropriate version (for example 1.12.1, which is the version used in this guide).

Why am I being warned my Sniffer V1 firmware is out of data but updates fail?

The Adafruit board has a small difference compared to the original Nordic HW that Nordic wrote their sniffer firmware for. To keep the cost as low as possible, we don't populate the optional 32.768KHz RTC crystal on our boards, whereas it is present on the more expensive Nordic development kit. Because the startup code in the sniffer firmware from Nordic uses this crystal, we had to request a custom version from Nordic that uses the internal 16MHz RC oscillator instead. When providing us the custom firmware, they changed the version number slightly, which is the reason for the warning message.

You can safely ignore the firmware update warning and use the device as normal, and in fact updating to a firmware from Nordic won't work unless you also solder the optional 32.768KHz crystal on the bottom of your PCB as well.

How can I check that the sniffer is outputting data?

If you think there is a problem with your sniffer, you should look at the LED closest to the black SWD connector box at the end of the board. It should flash every time Bluetooth Low Energy activity is detected when the serial port is open.

You can also open a Terminal Emulator (Putty, RealTerm, etc.) with the following settings, and you should see data coming out almost as soon as you plug the sniffer in:

- Baud Rate: 460800
- HW Flow Control: RTS + CTS Enabled

What is the difference between blue boards and black board)?

The Black boards (hardware v3) uses the much cheaper CP2104 USB to Serial bridge, and drops the SWD connector which had to be manually placed during the manufacturing process. This allows us to offer the sniffer board at a significant discount compared to the original, without sacrificing functionality that 99% of customers required. (The SWD pins are still available as pads on the bottom of the PCB if you need them!).

If you have a Blue board - you definitely have hardware version 1 and Firmware version 1

If you have a Black board - you definitely have hardware version 3. You may have firmware version 1 or version 2 depending on when you purchased it. Check your order receipt to know!



Downloads

Files

- EagleCAD PCB files on GitHub (https://adafru.it/oYD)
- Bluetooth LE module Datasheet (https://adafru.it/oYE)

Schematic

Click to embiggen

