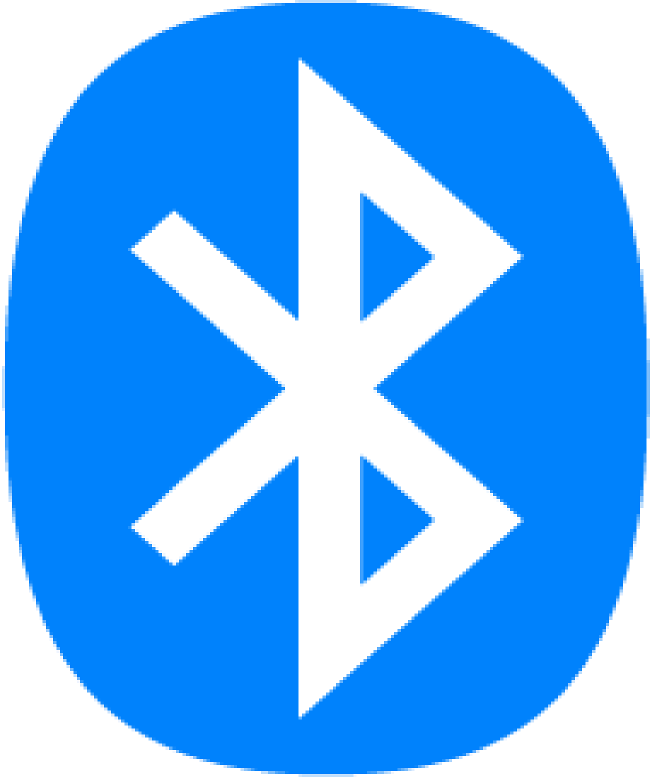
Aug 21-22 2020
Virtual Conference

BlueZ Cluez

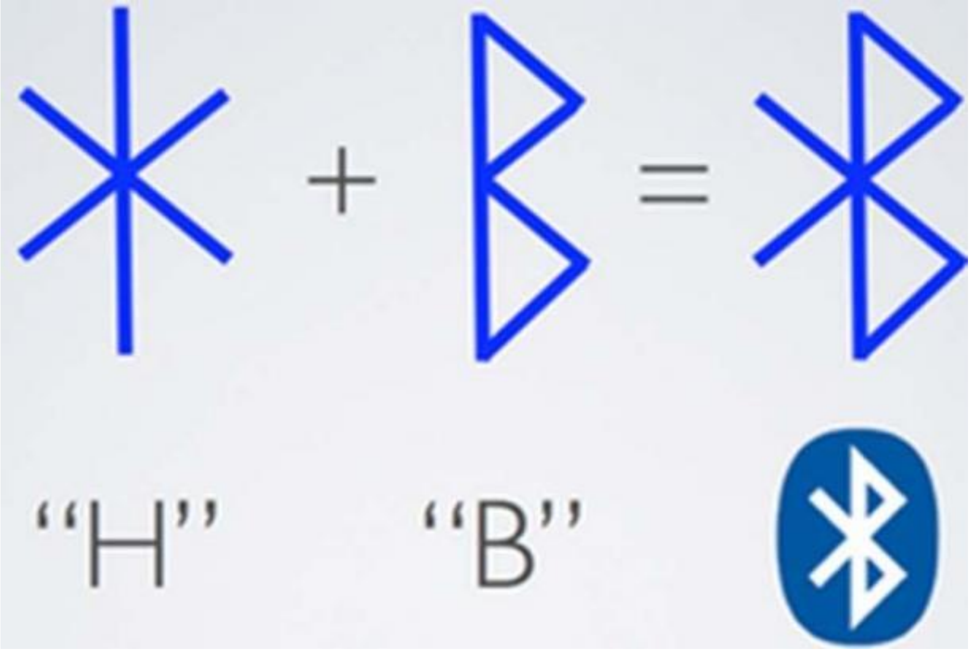
Ria Baldevia

The Historian Hat



The Historian Hat

Source: https://historiesoftheunexpected.com/magazine/harald-bluetooth-the-jelling-stones-and-the-fyrkat-ring-fortress/



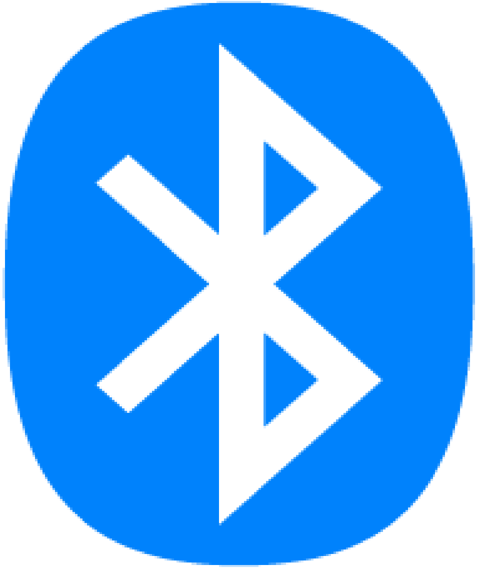
Source: https://www.ancient-origins.net/history-famous-people/bluetooth-why-modern-tech-named-after-powerful-king-denmark-and-norway-007398

The Historian Hat

- Named after a Viking King during the 10th Century, Harald “Blåtand” Gormsson.
- He united Scandinavia.
- Jim Kardach came up with the name in 1997. He was inspired by a book he was reading about the Vikings.
- The BT symbol is a combination of two runes from the runic alphabet that the Vikings used. The symbol is comprised of Harald “Bluetooth” Gormsson’s initials. The two initials merged is called a bindrune.
- Kardach was inspired by the book *Longships* by Frans G. Bengtsson. The book focused on Danish warriors who traveled the world looking for adventure.
- *The Vikings* by Gwyn Jones was the next book that exposed him to the runic stone.
- The premise of Bluetooth building networks was inspired by what King Gormsson had done in uniting Scandinavia.

Source: https://en.wikipedia.org/wiki/Bluetooth

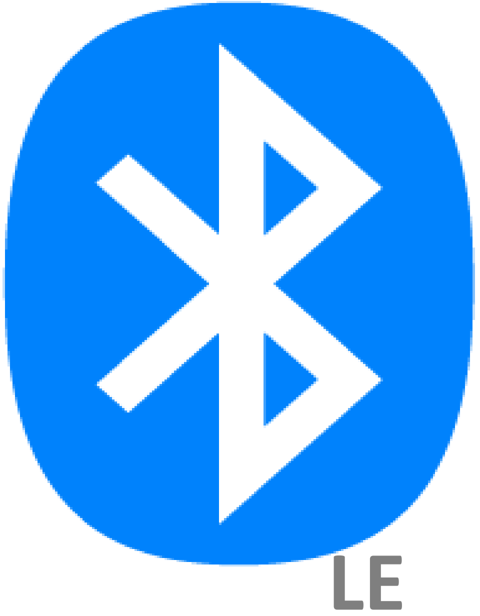
Bluetooth



Bluetooth
The Bluetooth wireless technology is a worldwide specification for a small-form factor, low-cost radio solution that provides links between mobile computers, mobile phones, other portable handheld devices, and connectivity to the Internet. The specification is developed, published and promoted by the Bluetooth Special Interest Group (SIG).

Source: https://developer.android.com/guide/topics/connectivity/bluetooth-le

Bluetooth LE



Bluetooth Low Energy (BTLE)
In contrast to Classic **Bluetooth**, **Bluetooth Low Energy** (**BLE**) is designed to provide significantly **lower** power **consumption**. This allows Android apps to communicate with **BLE** devices that have stricter power requirements, such as proximity sensors, heart rate monitors, and fitness devices.

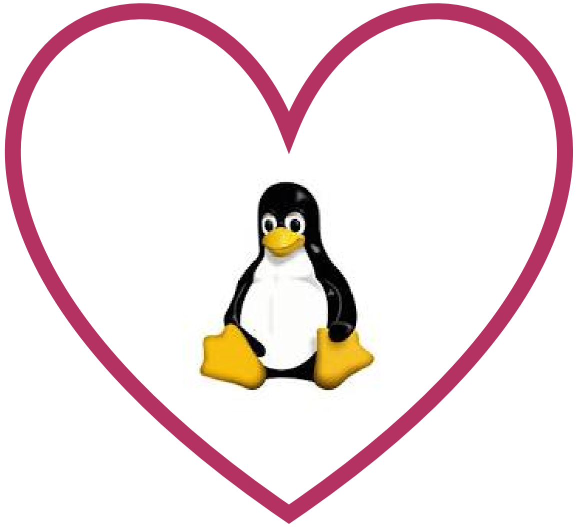
Source: https://developer.android.com/guide/topics/connectivity/bluetooth-le

BlueZ

What is BlueZ?

BlueZ is official Linux Bluetooth protocol stack. It is an Open Source project distributed under GNU General Public License (GPL).

- Developed by Max Krsnyansky and released it under GPL in 2001.
- In 2004, BlueZ was handed over to Marcel Holtmann.
- In 2005, maintenance responsibilities were transferred to Bluetooth SIG.



Source: http://www.bluez.org/about/history/

Features

BlueZ provides support for the core Bluetooth layers and protocols

- Complete modular implementation
- Symmetric multi-processing safe
- Multithreaded data processing
- Support for multiple Bluetooth devices
- Real hardware abstraction
- Standard socket interface to all layers
- Device and service level security support

Currently BlueZ consists of many separate modules:

- Bluetooth kernel subsystem core
- L2CAP and SCO audio kernel layers
- RFCOMM, BNEP, CMTP and HIDP kernel implementations
- HCI UART, USB, PCMCIA and virtual device drivers
- General Bluetooth and SDP libraries and daemons
- Configuration and testing utilities
- Protocol decoding and analysis tools

Source: BlueZ.org

Platforms & Distributions

Platforms

The BlueZ kernel modules, libraries and utilities are known to be working perfect on many architectures supported by Linux. This also includes single and multi processor platforms as well as hyper threading systems:

- Intel and AMD x86
- AMD64 and EM64T (x86-64)
- SUN SPARC 32/64bit
- PowerPC 32/64bit
- Intel StrongARM and Xscale
- Hitachi/Renesas SH processors
- Motorola DragonBall

Distributions

Support for BlueZ can be found in many Linux distributions and in general it is compatible with any Linux system on the market:

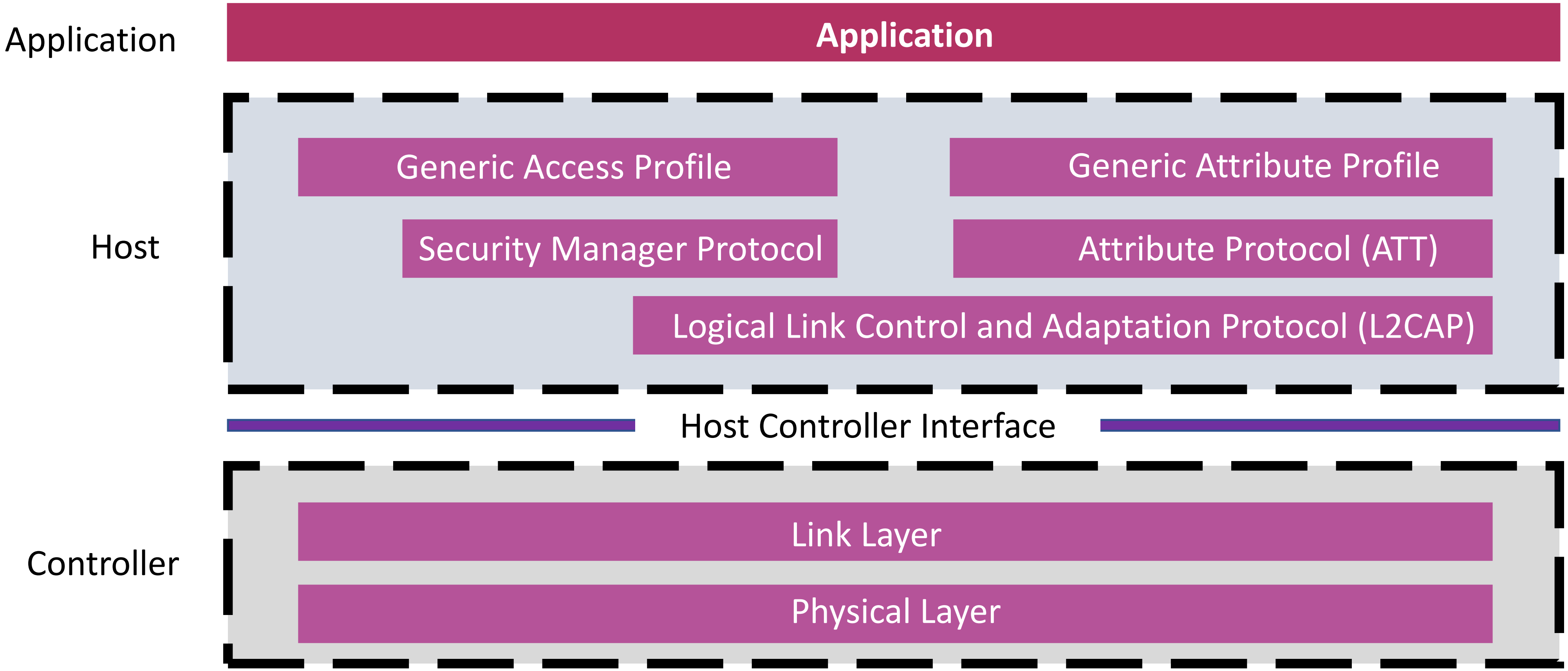
- Debian GNU/Linux
- Ubuntu Linux
- Fedora Core / Red Hat Linux
- OpenSuSE / SuSE Linux
- Mandrake Linux
- Gentoo Linux

Source: BlueZ.org

Commands

COMMAND	SHORT DESCRIPTION
bluez	The *bluetoothd* Bluetooth daemon
obex	The *obexd* OBEX daemon
bluetoothctl	A command-line interface to the BlueZ
obexctl	A command-line interface to the BlueZ for file transfers
hciconfig	HCI device configuration utility
hcidump	Reads raw HCI data and prints it on screen
hciattach	Attach a serial UART to the BT stack as a transport interface
hcitool	Tool used to configure Bluetooth connections
sdptool	A tool to perform SDP queries on Bluetooth devices
btattach	The successor to hciattach since bluez 5.37
btmgmt	Tool for management of the bluez daemon
btmon	Bluetooth event monitoring
meshctl	Used to provision mesh devices
Gatttool	Used to interact with BT devices/peripherals

BLE Protocol Stack



Application

Application

Host

Generic Access Profile

Generic Attribute Profile

Security Manager Protocol

Attribute Protocol (ATT)

Logical Link Control and Adaptation Protocol (L2CAP)

Host Controller Interface

Controller

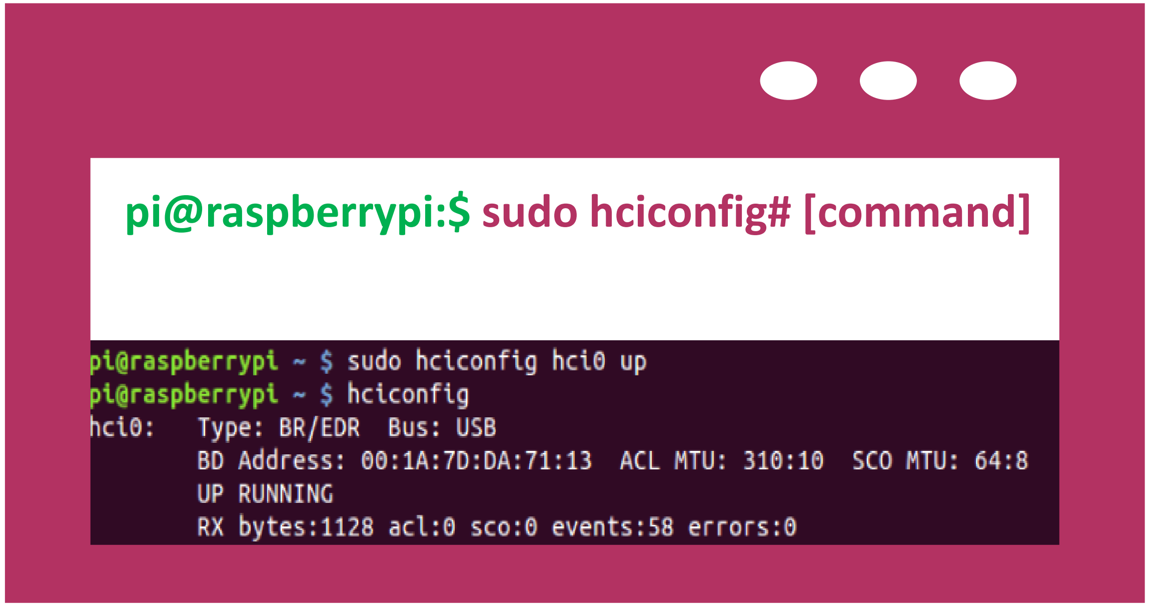
Link Layer

Physical Layer

Source: O'Reilly's Getting Started with Bluetooth Low Energy

HCI & GENERAL ATTRIBUTE

hciconfig is used to configure Bluetooth devices.



```
pi@raspberrypi:$ sudo hciconfig# [command]
```

```
pi@raspberrypi ~ $ sudo hciconfig hci0 up
pi@raspberrypi ~ $ hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING
        RX bytes:1128 acl:0 sco:0 events:58 errors:0
```

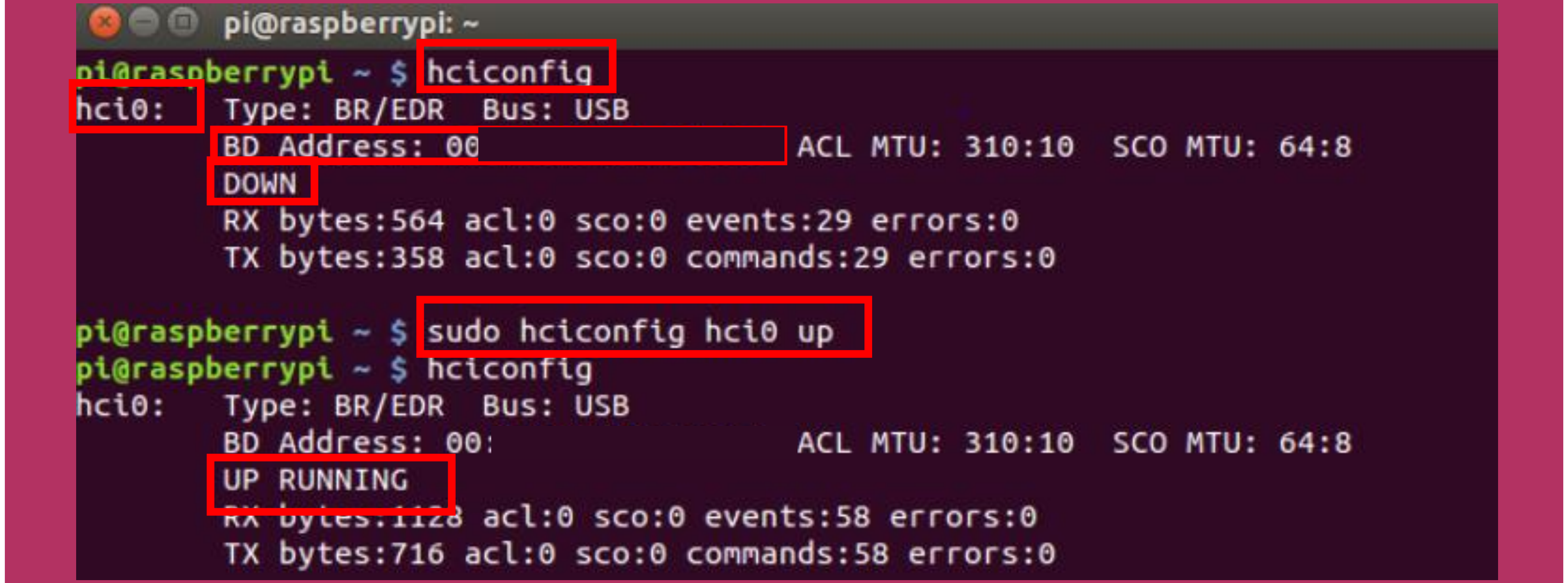
Source: https://linux.die.net/man/8/hciconfig

HCI

Command	Definition
up	Open and initialize HCI device
down	Close HCI device
reset	Reset HCI device
Rstat	Reset statistic encounters
Auth	Enable authentication
Noauth	Disable authentication
Encrypt	Enable encryption
Noencrypt	Disable encryption

Source: https://linux.die.net/man/8/hciconfig

HCICONFIG



```
pi@raspberrypi ~ $ hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00[redacted]  ACL MTU: 310:10  SCO MTU: 64:8
        DOWN
        RX bytes:564 acl:0 sco:0 events:29 errors:0
        TX bytes:358 acl:0 sco:0 commands:29 errors:0

pi@raspberrypi ~ $ sudo hciconfig hci0 up
pi@raspberrypi ~ $ hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:[redacted]  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING
        RX bytes:1128 acl:0 sco:0 events:58 errors:0
        TX bytes:716 acl:0 sco:0 commands:58 errors:0
```

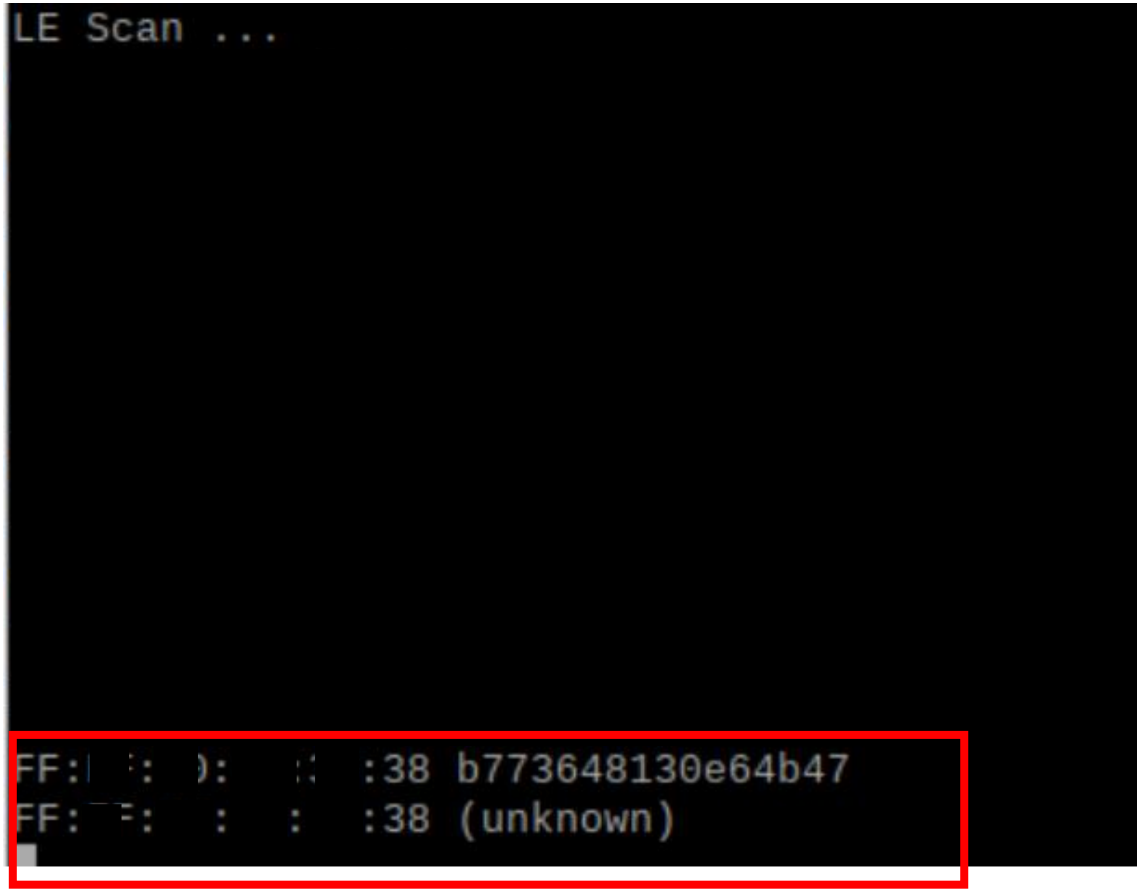
- Hciconfig
- Hci0
- BD Address
- DOWN
- Sudo hciconfig hcio up
- UP Running

HCITOOL

hcitool allows you to scan for BLE peripherals in range, connect to them, or optionally simulate a BLE device using any supported BLE 4.0 USB dongle. To scan for BLE devices in range you can issue the following command:

```
pi@raspberrypi:$ sudo hcitool –i hci0 lescan
```

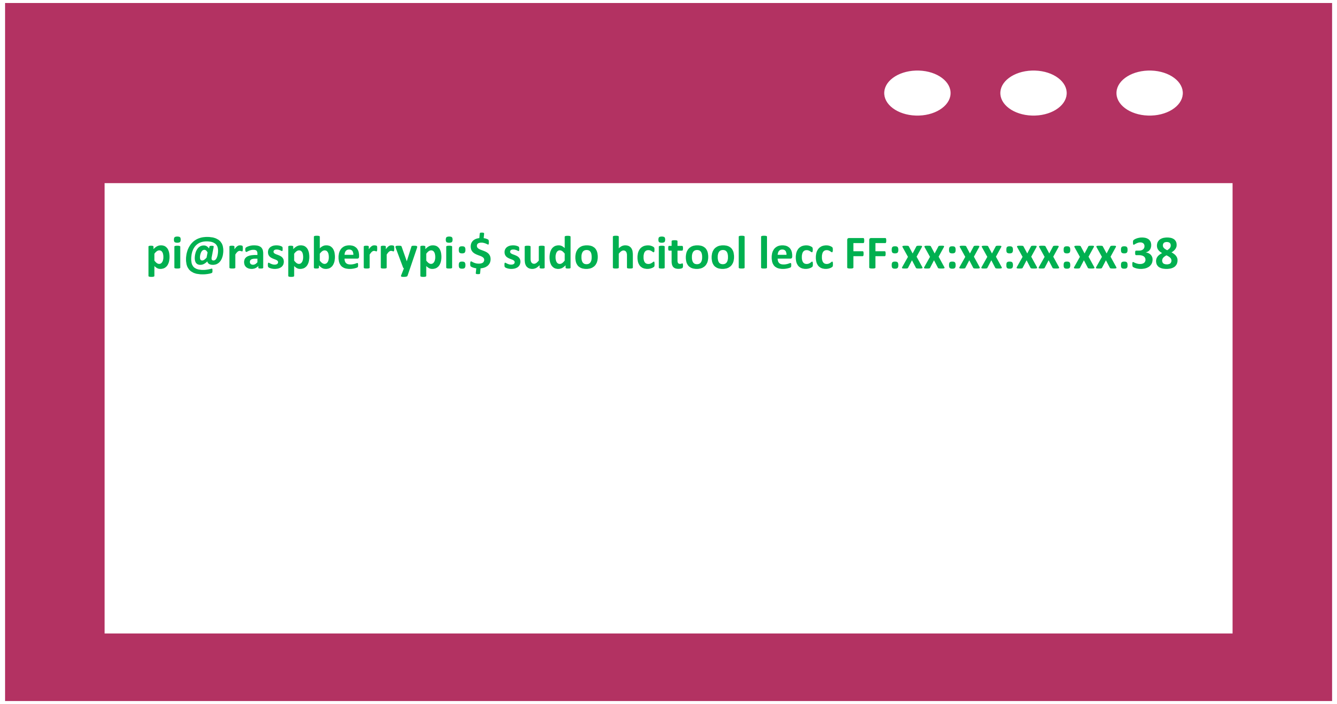
This output will showcase LE devices. You should be able to see your device.



```
LE Scan ...
FF:[redacted]:38 b773648130e64b47
FF:[redacted]:38 (unknown)
```

HCI

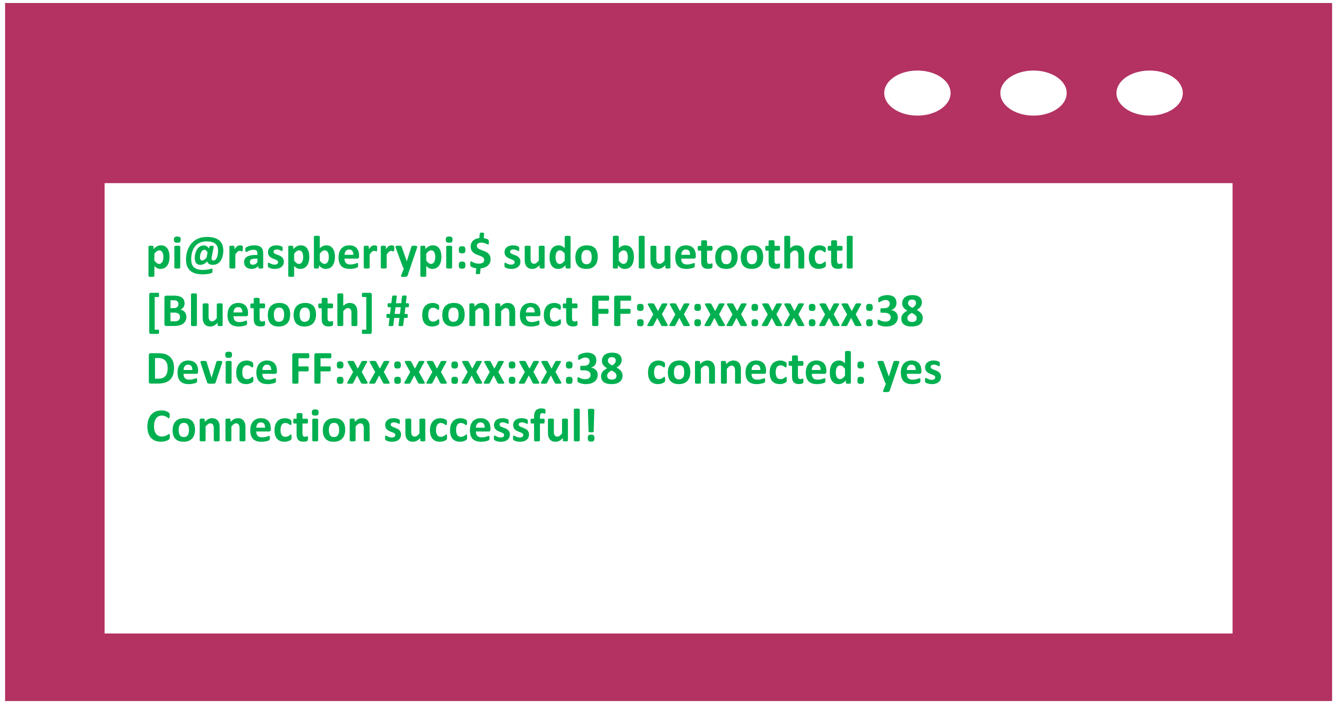
Once you have the device’s address, you can connect to the peripheral using the following command with its address:



```
pi@raspberrypi:$ sudo hcitool lecc FF:xx:xx:xx:xx:38
```

Bluetoothctl

Bluetoothctl: the command-line interface to BlueZ.



```
pi@raspberrypi:$ sudo bluetoothctl
[Bluetooth] # connect FF:xx:xx:xx:xx:38
Device FF:xx:xx:xx:xx:38  connected: yes
Connection successful!
```

Generic Attribute

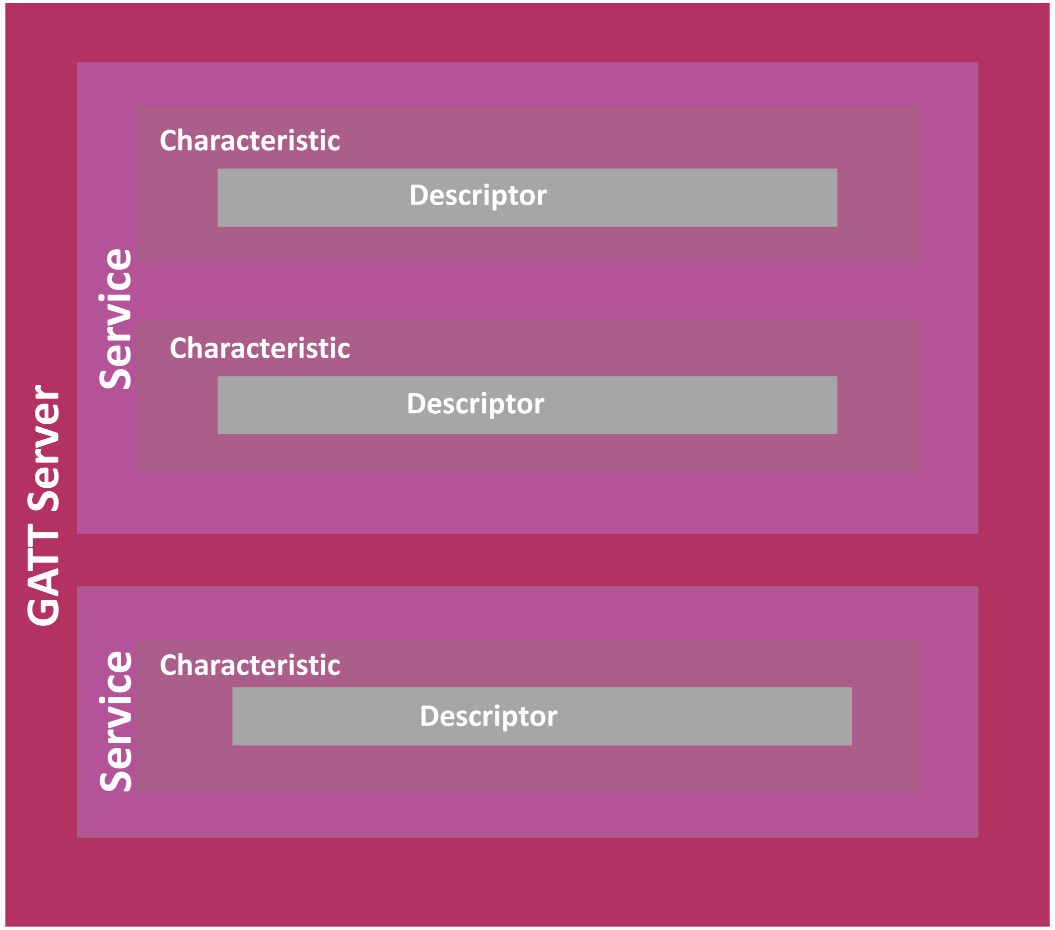
Client	Server
The GATT client corresponds to the ATT client. The GATT client does not know anything in advance about the server’s attributes, so it must first inquire about the presence and nature of those attributes by performing service discovery. After completing service discovery, it can then start reading and writing attributes found in the server, as well as receiving server-initiated updates.	The GATT server corresponds to the ATT server. It receives requests from a client and sends responses back. It also sends server-initiated updates when configured to do so, and it is the role responsible for storing and making the user data available to the client, organized in attributes. Every BLE device sold must include at least a basic GATT server that can respond to client requests, even if only to return an error response.

Source: O’Reilly’s Getting Started with Bluetooth Low Energy

Generic Attribute

- The Generic Attribute Profile (GATT) establishes in detail how to exchange all profile and user data over a BLE connection. GATT deals only with actual data transfer procedures and formats.

- GATT uses the Attribute Protocol (ATT) as its transport protocol to exchange data between devices. This data is organized hierarchically in sections called services, which group conceptually related pieces of user data called *characteristics*.

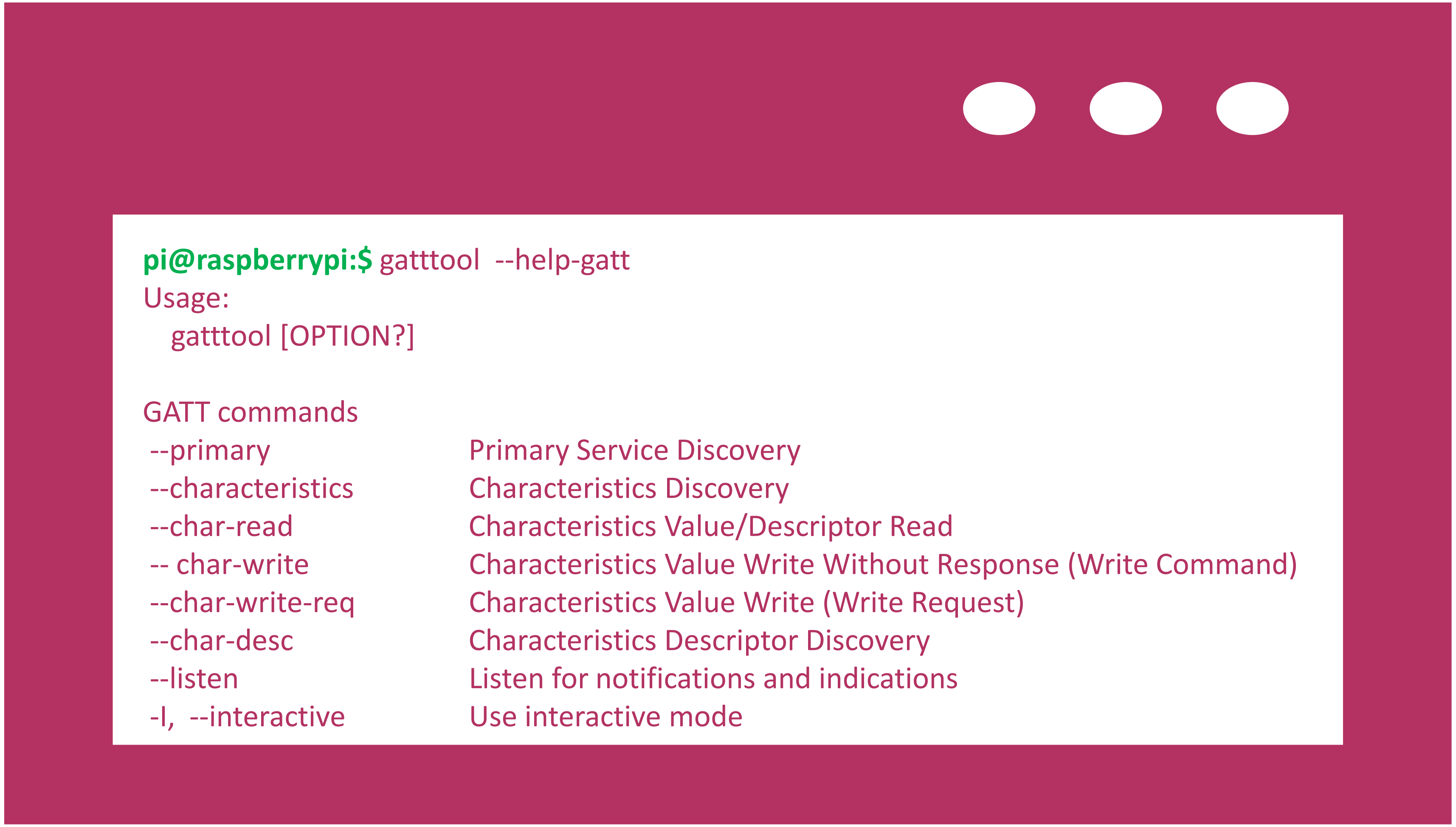


Source: O'Reilly's Getting Started with Bluetooth Low Energy

Gatttool

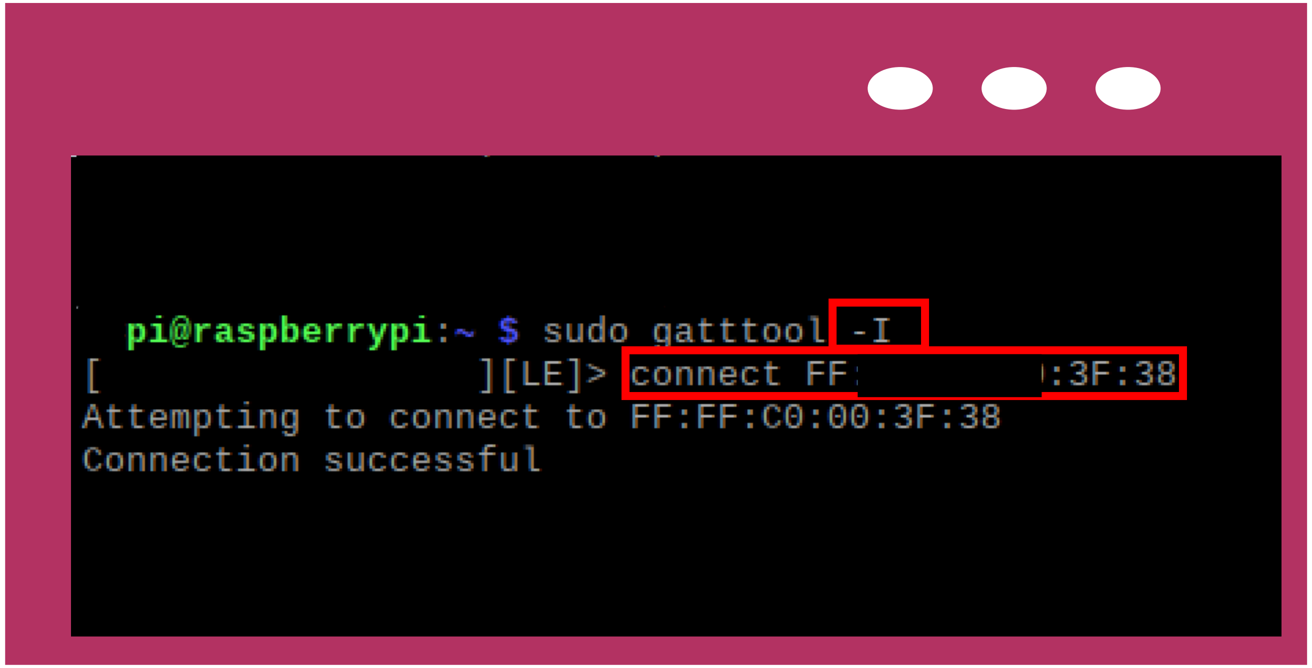
Gatttool

We can discover, read, and write characteristics with gatttool. It defines a data structure for organizing characteristics and attributes. Launch gatttool in interactive mode.



```
pi@raspberrypi:$ gatttool  --help-gatt
Usage:
   gatttool [OPTION?]

GATT commands
 --primary                    Primary Service Discovery
 --characteristics            Characteristics Discovery
 --char-read                  Characteristics Value/Descriptor Read
 -- char-write                Characteristics Value Write Without Response (Write Command)
 --char-write-req             Characteristics Value Write (Write Request)
 --char-desc                  Characteristics Descriptor Discovery
 --listen                     Listen for notifications and indications
 -I,  --interactive           Use interactive mode
```

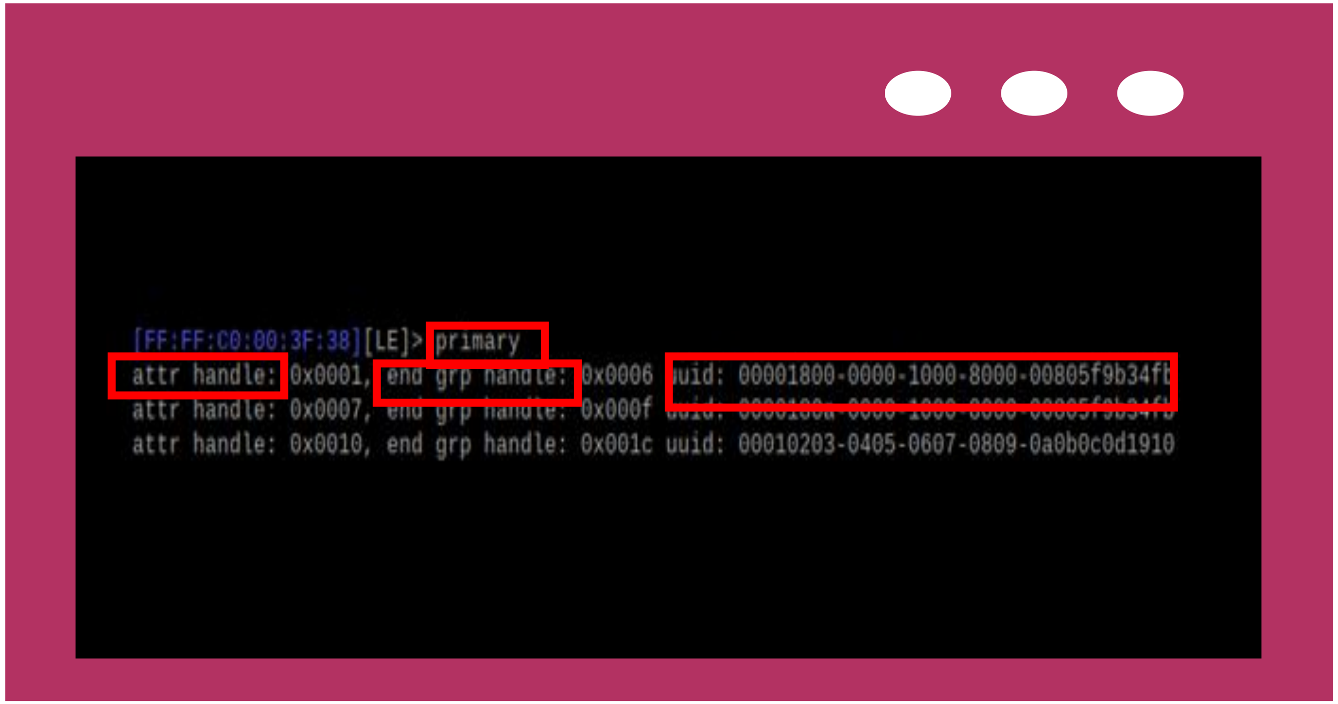
Gatttool



```
  pi@raspberrypi:~ $ sudo gatttool -I
[                 ][LE]> connect FF:[redacted]:3F:38
Attempting to connect to FF:FF:C0:00:3F:38
Connection successful
```

- ❖ -I
- ❖ Connect

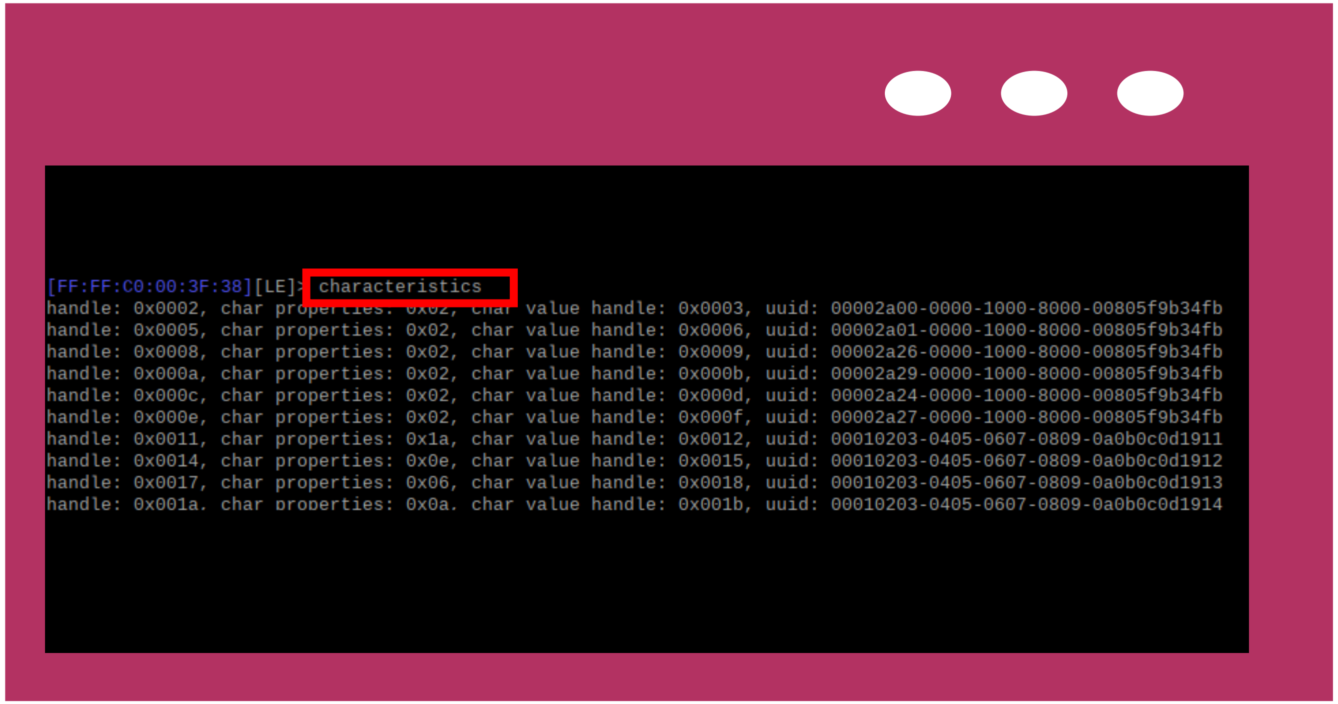
Gatttool



```
[FF:FF:C0:00:3F:38][LE]> primary
attr handle: 0x0001, end grp handle: 0x0006 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0007, end grp handle: 0x000f uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle: 0x0010, end grp handle: 0x001c uuid: 00010203-0405-0607-0809-0a0b0c0d1910
```

- Primary
- Attr handle
- UUID

Gatttool



```
[FF:FF:C0:00:3F:38][LE]> characteristics
handle: 0x0002, char properties: 0x02, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0005, char properties: 0x02, char value handle: 0x0006, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0008, char properties: 0x02, char value handle: 0x0009, uuid: 00002a26-0000-1000-8000-00805f9b34fb
handle: 0x000a, char properties: 0x02, char value handle: 0x000b, uuid: 00002a29-0000-1000-8000-00805f9b34fb
handle: 0x000c, char properties: 0x02, char value handle: 0x000d, uuid: 00002a24-0000-1000-8000-00805f9b34fb
handle: 0x000e, char properties: 0x02, char value handle: 0x000f, uuid: 00002a27-0000-1000-8000-00805f9b34fb
handle: 0x0011, char properties: 0x1a, char value handle: 0x0012, uuid: 00010203-0405-0607-0809-0a0b0c0d1911
handle: 0x0014, char properties: 0x0e, char value handle: 0x0015, uuid: 00010203-0405-0607-0809-0a0b0c0d1912
handle: 0x0017, char properties: 0x06, char value handle: 0x0018, uuid: 00010203-0405-0607-0809-0a0b0c0d1913
handle: 0x001a, char properties: 0x0a, char value handle: 0x001b, uuid: 00010203-0405-0607-0809-0a0b0c0d1914
```

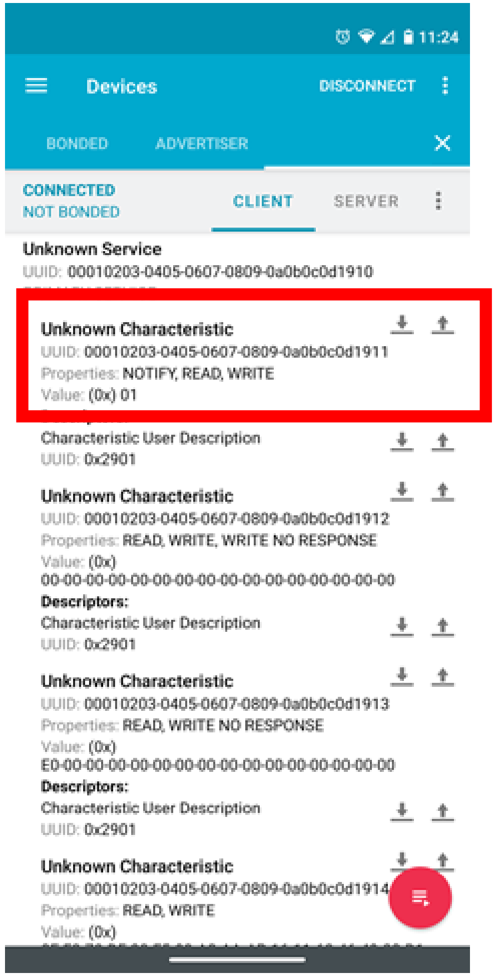
- ❖ Characteristics

Breaking Boundaries

The Diana Initiative

Byte by Byte

UUID & Handles



Unknown Service
UUID: 00010203-0405-0607-0809-0a0b0c0d1910

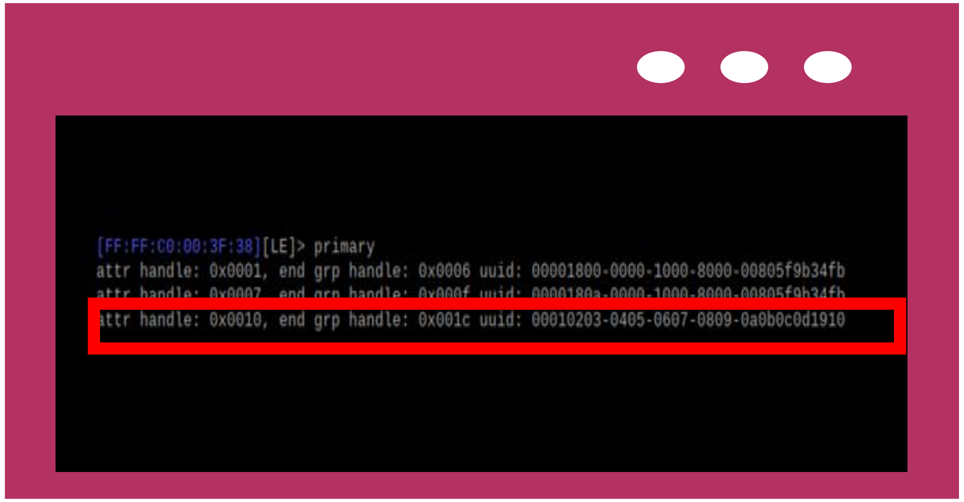
Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1911
Properties: NOTIFY, READ, WRITE
Value: (0x) 01

Characteristic User Description
UUID: 0x2901

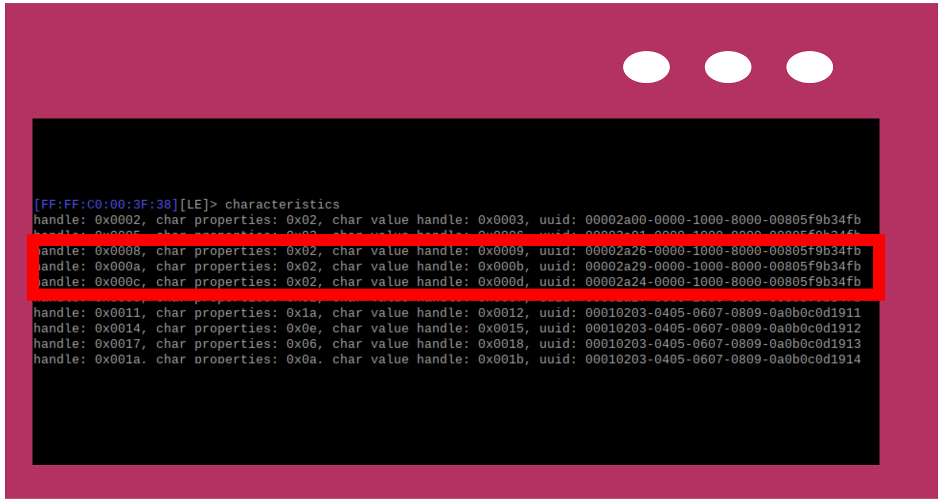
Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1912
Properties: READ, WRITE, WRITE NO RESPONSE
Value: (0x)
00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
Descriptors:
Characteristic User Description
UUID: 0x2901

Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1913
Properties: READ, WRITE NO RESPONSE
Value: (0x)
E0-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
Descriptors:
Characteristic User Description
UUID: 0x2901

Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1914
Properties: READ, WRITE
Value: (0x)



```
[FF:FF:C0:00:3F:38][LE]> primary
attr handle: 0x0001, end grp handle: 0x0006 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0007, end grp handle: 0x000f uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle: 0x0010, end grp handle: 0x001c uuid: 00010203-0405-0607-0809-0a0b0c0d1910
```



```
[FF:FF:C0:00:3F:38][LE]> characteristics
handle: 0x0002, char properties: 0x02, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0005, char properties: 0x02, char value handle: 0x0006, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0008, char properties: 0x02, char value handle: 0x0009, uuid: 00002a26-0000-1000-8000-00805f9b34fb
handle: 0x000a, char properties: 0x02, char value handle: 0x000b, uuid: 00002a29-0000-1000-8000-00805f9b34fb
handle: 0x000c, char properties: 0x02, char value handle: 0x000d, uuid: 00002a24-0000-1000-8000-00805f9b34fb
handle: 0x0011, char properties: 0x1a, char value handle: 0x0012, uuid: 00010203-0405-0607-0809-0a0b0c0d1911
handle: 0x0014, char properties: 0x0e, char value handle: 0x0015, uuid: 00010203-0405-0607-0809-0a0b0c0d1912
handle: 0x0017, char properties: 0x06, char value handle: 0x0018, uuid: 00010203-0405-0607-0809-0a0b0c0d1913
handle: 0x001a, char properties: 0x0a, char value handle: 0x001b, uuid: 00010203-0405-0607-0809-0a0b0c0d1914
```

UUID

UUID

A universally unique identifier (UUID) is a 128 bit number that should be unique. There are two formats:

- 16 bit
- 32 bit

- Shortened version is used for formats that are defined and listed as standard Bluetooth UUIDs which are approved by the Bluetooth SIG.

!!! If a company or developer creates a new requirement or capability and it does not fit with any of the standard UUIDs, then a custom UUID can be generated.

Standard	
Client	**Server**
Firmware Revision String UUID: 0x2A26	Link Loss UUID: 0x1803
Manufacturer Name String UUID: 0x2A29	Current Time Service UUID: 0x1805
Model Number String UUID: 0x2A24	Heart Rate UUID: 0x180D

Customized
Client
Unknown Service Unknown Characteristic UUID: 00010203-0405-0607-0809-0a0b0c0d1911

Source: O'Reilly's Getting Started with Bluetooth Low Energy

Attributes

Attributes	Description
Type	A UUID that determines the kind of data present in the value of the attribute.
Permissions	Metadata that indicate which ATT activities can be executed on each attribute: ❖ Access Permissions: None; Readable; Writable; and Readable and writable. ❖ Encryption: No encryption requires; Unauthenticated encryption required; and Authenticated encryption required. ❖ Authorization: No authorization required; and Authorization required.
Value	Actual data content of the attribute. A client can freely access to both read and write, based on permissions.
Handle	Unique identifier for each attribute on a GATT server. It makes the attribute addressable.

Source: O'Reilly's Getting Started with Bluetooth Low Energy

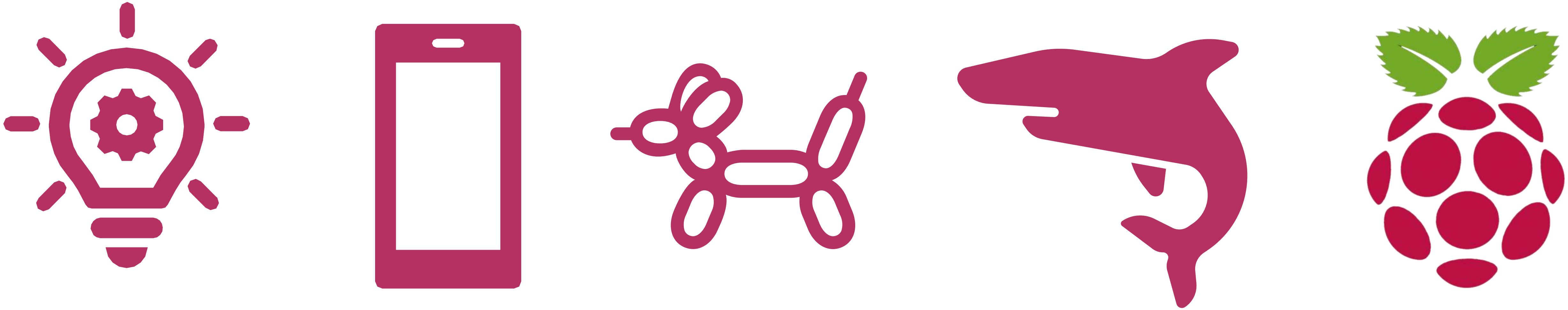
Attributes

Handle	Type	Permission	Value	Value Length
0x0201	UUID$_1$	Read only, no security	0x180A	2
0x0202	UUID$_2$	Read only, no security	0x2A29	2
0x0215	UUID$_3$	Read/write, authorization required	“a readable UTF-8 string”	23
0x030C	UUID$_4$	Write only, no security	{0xFF, 0xFF, 0x00, 0x00}	4
0x030D	UUID$_5$ (128-bit)	Read/write, authenticated encryption require	36.43	8
0x031A	UUID$_1$	Read only, no security	0x1801	2

Source: O’Reilly’s Getting Started with Bluetooth Low Energy

Objective

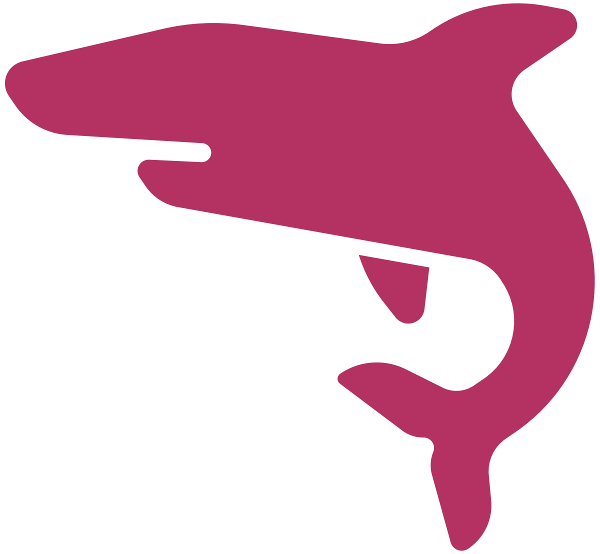
My Environment & Set-up



+ My Environment & Set-up

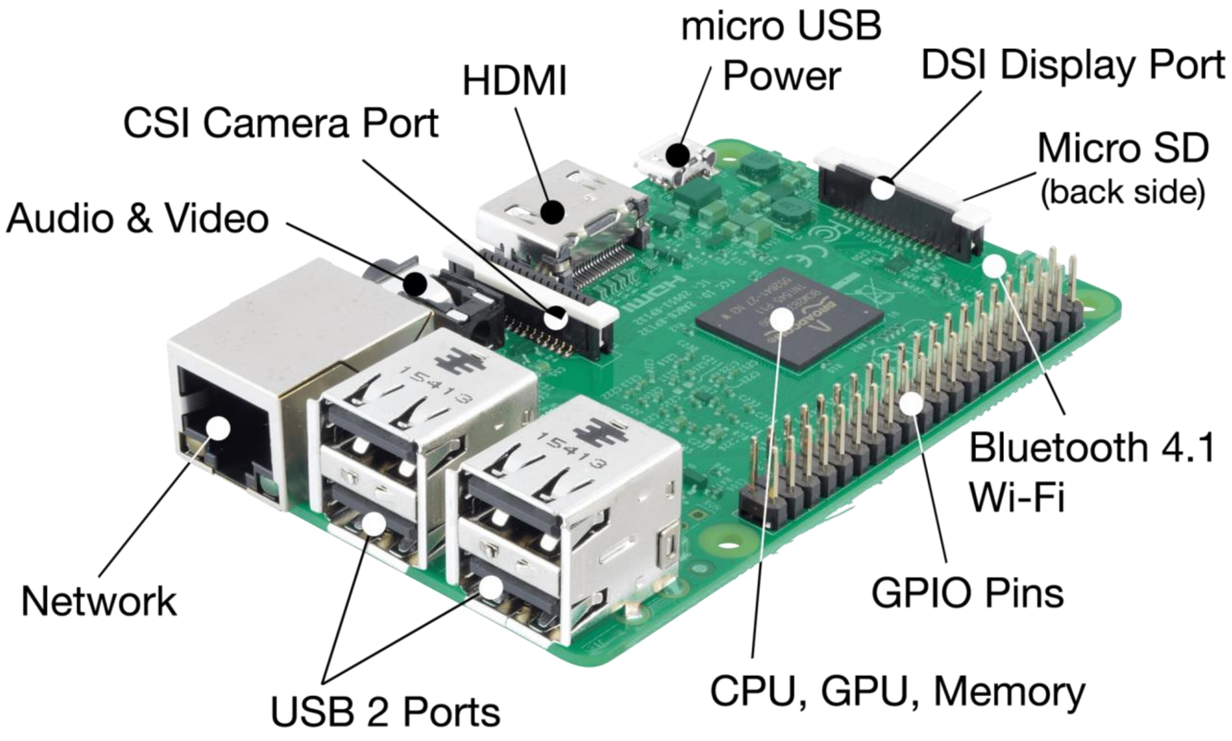
My Environment & Set-up

- Ubertooth One → ubertooth
- Micro:bit → btlejack “Swiss Army Knife for Bluetooth”
- Android Mobile → Developer Tool Option > Enable sniffer
- Adafruit Bluefruit LE Sniffer → V1



Packet analyzer: Wireshark

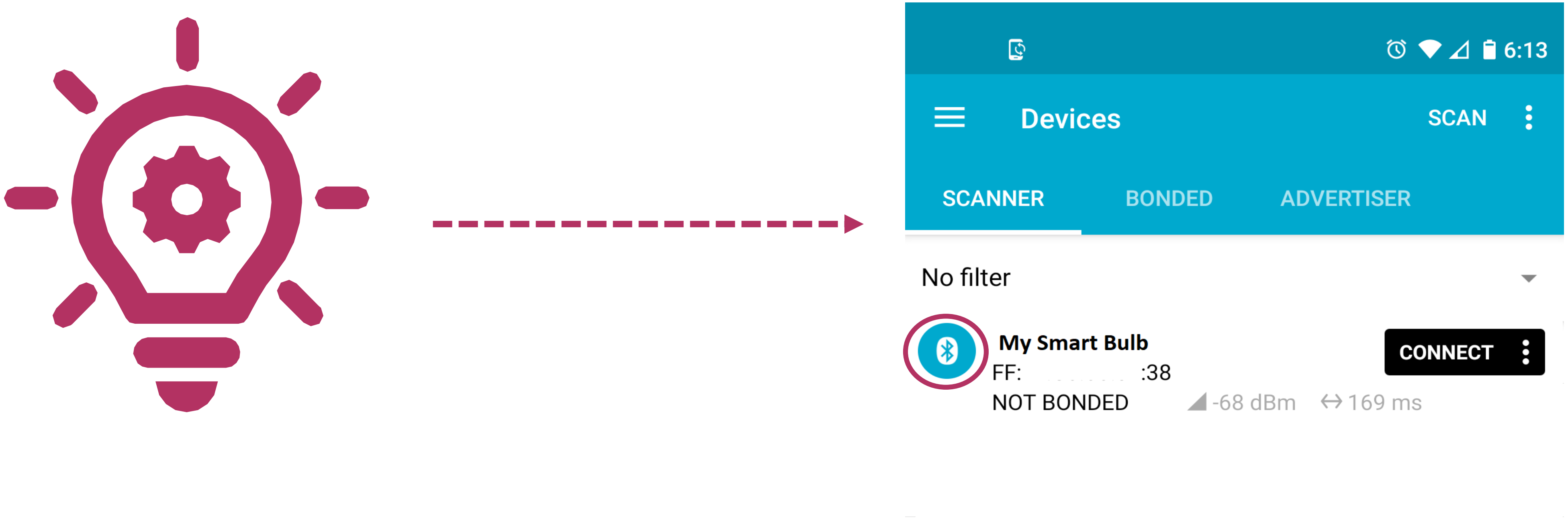
My Environment & Set-up



Source: https://medium.com/coinmonks/raspberry-pi-3-model-b-shell-scripting-door-monitor-b44944f82d87

- Raspbian, Debian-based operating system (OS)
- You can use Kali Linux for the Pi, too
- Raspberry Pi 3 Model B comes with Bluetooth, no need for a Bluetooth adapter
- Easy solution if you don’t have a dedicated Linux set-up at home

Scan



6:13

Devices

SCAN

SCANNER

BONDED

ADVERTISER

No filter

My Smart Bulb

FF: :38

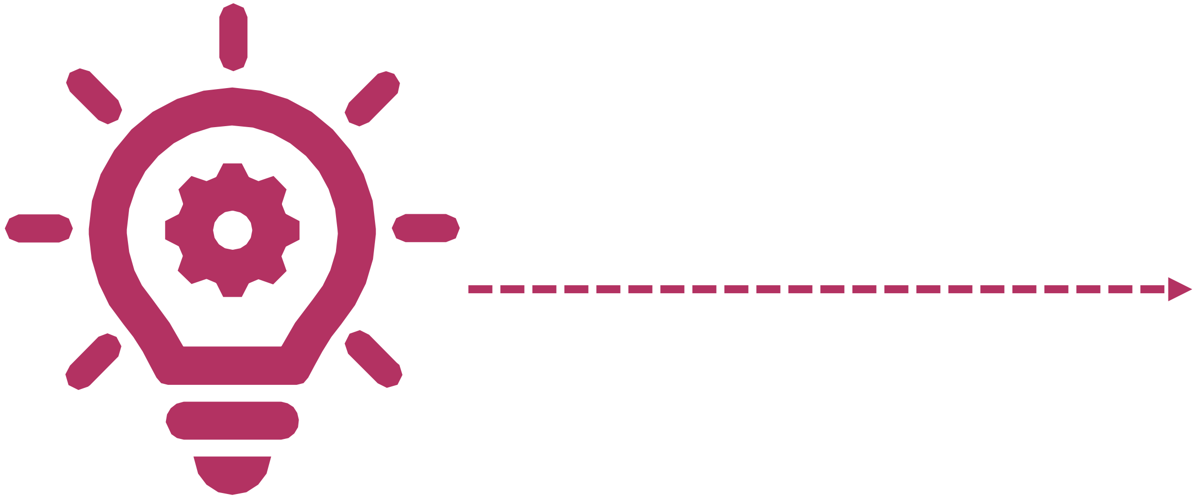
NOT BONDED

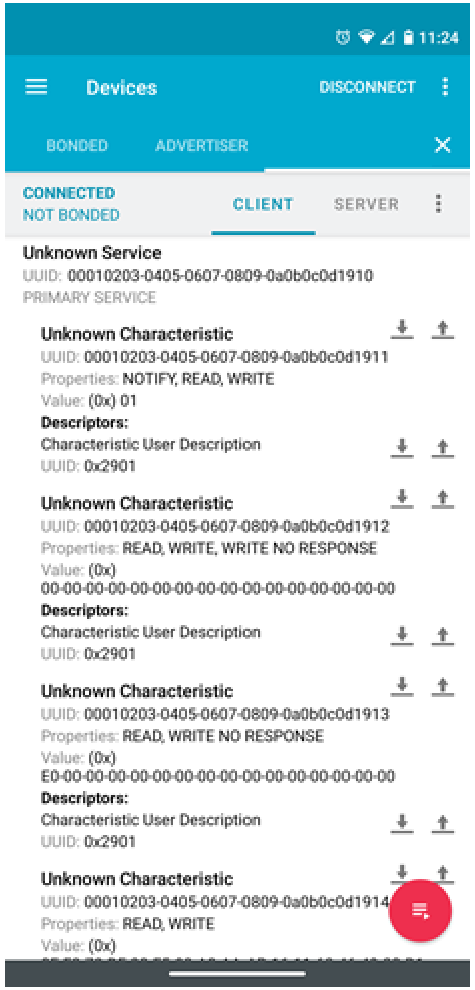
-68 dBm

↔ 169 ms

CONNECT

Scan





Unknown Service
UUID: 00010203-0405-0607-0809-0a0b0c0d1910
PRIMARY SERVICE

Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1911
Properties: NOTIFY, READ, WRITE
Value: (0x) 01
Descriptors:
Characteristic User Description
UUID: 0x2901

Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1912
Properties: READ, WRITE, WRITE NO RESPONSE
Value: (0x)
00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
Descriptors:
Characteristic User Description
UUID: 0x2901

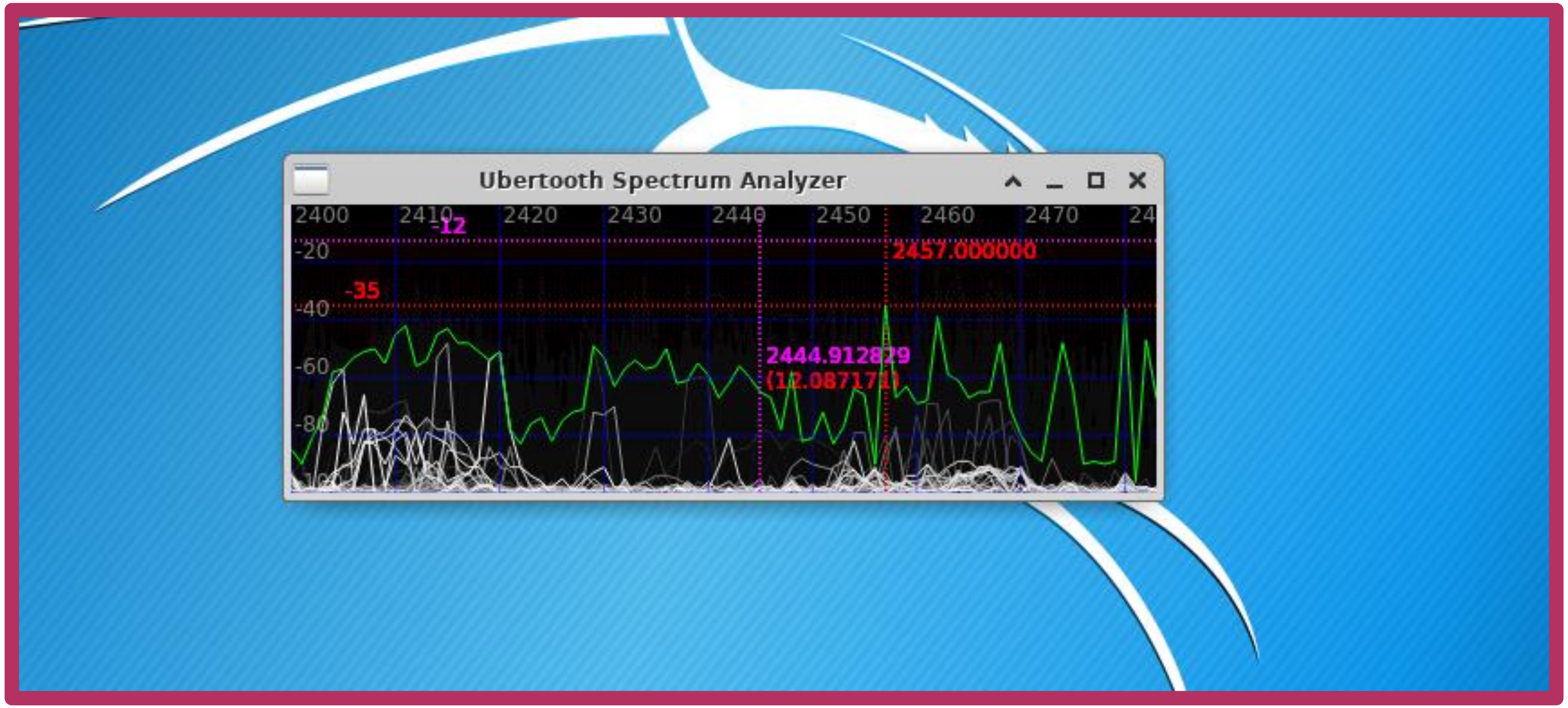
Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1913
Properties: READ, WRITE NO RESPONSE
Value: (0x)
E0-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
Descriptors:
Characteristic User Description
UUID: 0x2901

Unknown Characteristic
UUID: 00010203-0405-0607-0809-0a0b0c0d1914
Properties: READ, WRITE
Value: (0x)

Sniff

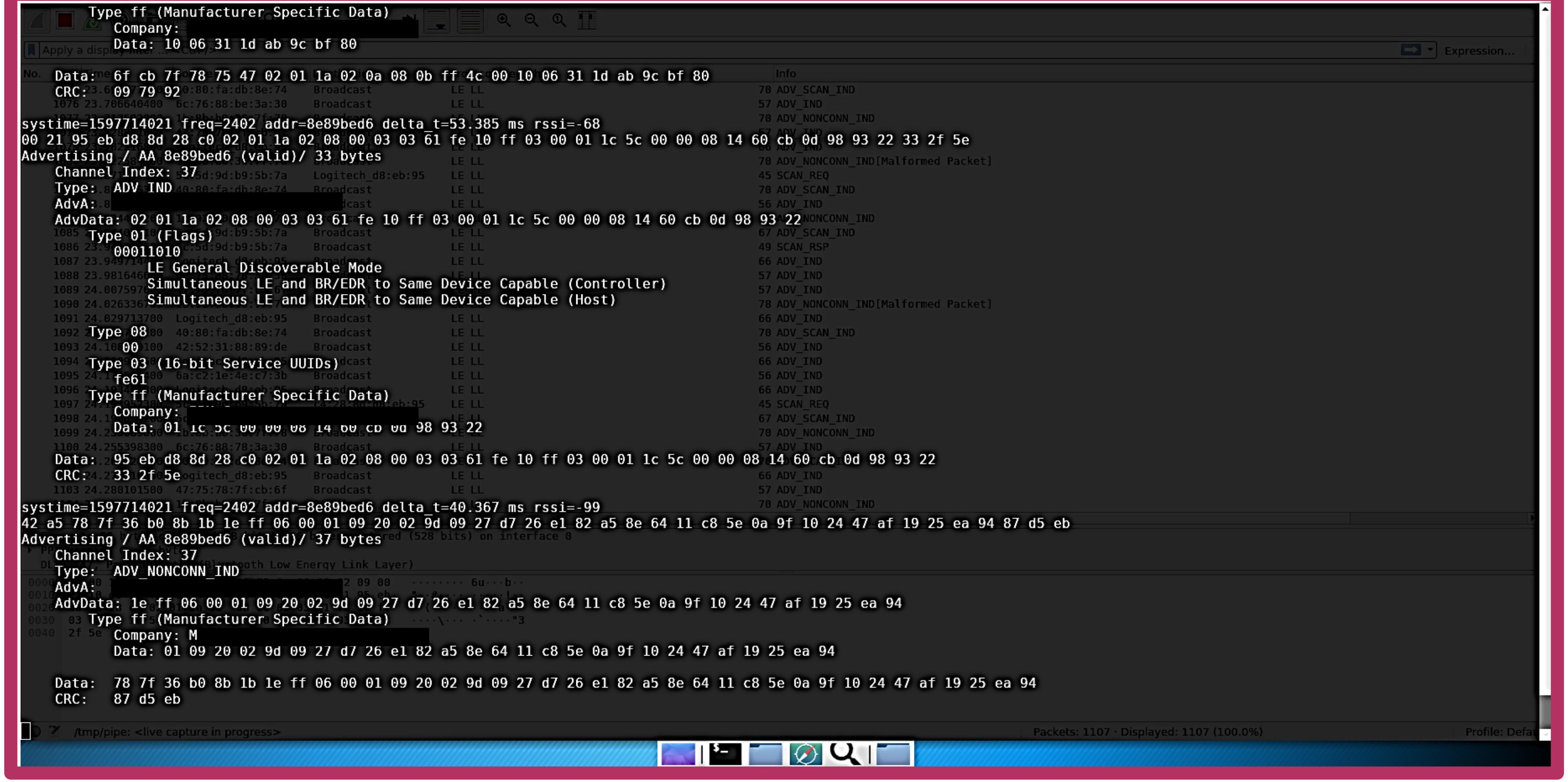
Sniff

It’s working!!!



Sniff

- Hard to analyze
- May miss packets
- I encountered a problem where I reached a limit so it stopped
- There is a way to get this fed into Wireshark



```
        Type ff (Manufacturer Specific Data)
            Company: [redacted]
            Data: 10 06 31 1d ab 9c bf 80

    Data:  6f cb 7f 78 75 47 02 01 1a 02 0a 08 0b ff 4c 00 10 06 31 1d ab 9c bf 80
    CRC:   09 79 92

systime=1597714021 freq=2402 addr=8e89bed6 delta_t=53.385 ms rssi=-68
00 21 95 eb d8 8d 28 c0 02 01 1a 02 08 00 03 03 61 fe 10 ff 03 00 01 1c 5c 00 00 08 14 60 cb 0d 98 93 22 33 2f 5e
Advertising / AA 8e89bed6 (valid)/ 33 bytes
    Channel Index: 37
    Type:  ADV_IND
    AdvA: [redacted]
    AdvData: 02 01 1a 02 08 00 03 03 61 fe 10 ff 03 00 01 1c 5c 00 00 08 14 60 cb 0d 98 93 22
        Type 01 (Flags)
            00011010
                LE General Discoverable Mode
                Simultaneous LE and BR/EDR to Same Device Capable (Controller)
                Simultaneous LE and BR/EDR to Same Device Capable (Host)

        Type 08
             00
        Type 03 (16-bit Service UUIDs)
            fe61
        Type ff (Manufacturer Specific Data)
            Company: [redacted]
            Data: 01 1c 5c 00 00 08 14 60 cb 0d 98 93 22

    Data:  95 eb d8 8d 28 c0 02 01 1a 02 08 00 03 03 61 fe 10 ff 03 00 01 1c 5c 00 00 08 14 60 cb 0d 98 93 22
    CRC:   33 2f 5e

systime=1597714021 freq=2402 addr=8e89bed6 delta_t=40.367 ms rssi=-99
42 a5 78 7f 36 b0 8b 1b 1e ff 06 00 01 09 20 02 9d 09 27 d7 26 e1 82 a5 8e 64 11 c8 5e 0a 9f 10 24 47 af 19 25 ea 94 87 d5 eb
Advertising / AA 8e89bed6 (valid)/ 37 bytes
    Channel Index: 37
    Type:  ADV_NONCONN_IND
    AdvA: [redacted]
    AdvData: 1e ff 06 00 01 09 20 02 9d 09 27 d7 26 e1 82 a5 8e 64 11 c8 5e 0a 9f 10 24 47 af 19 25 ea 94
        Type ff (Manufacturer Specific Data)
            Company: M [redacted]
            Data: 01 09 20 02 9d 09 27 d7 26 e1 82 a5 8e 64 11 c8 5e 0a 9f 10 24 47 af 19 25 ea 94

    Data:  78 7f 36 b0 8b 1b 1e ff 06 00 01 09 20 02 9d 09 27 d7 26 e1 82 a5 8e 64 11 c8 5e 0a 9f 10 24 47 af 19 25 ea 94
    CRC:   87 d5 eb
```

Apply a displ
Expression...
/tmp/pipe: <live capture in progress>
Packets: 1107 · Displayed: 1107 (100.0%)
Profile: Defa

Lesson Learned: Capture Set-up

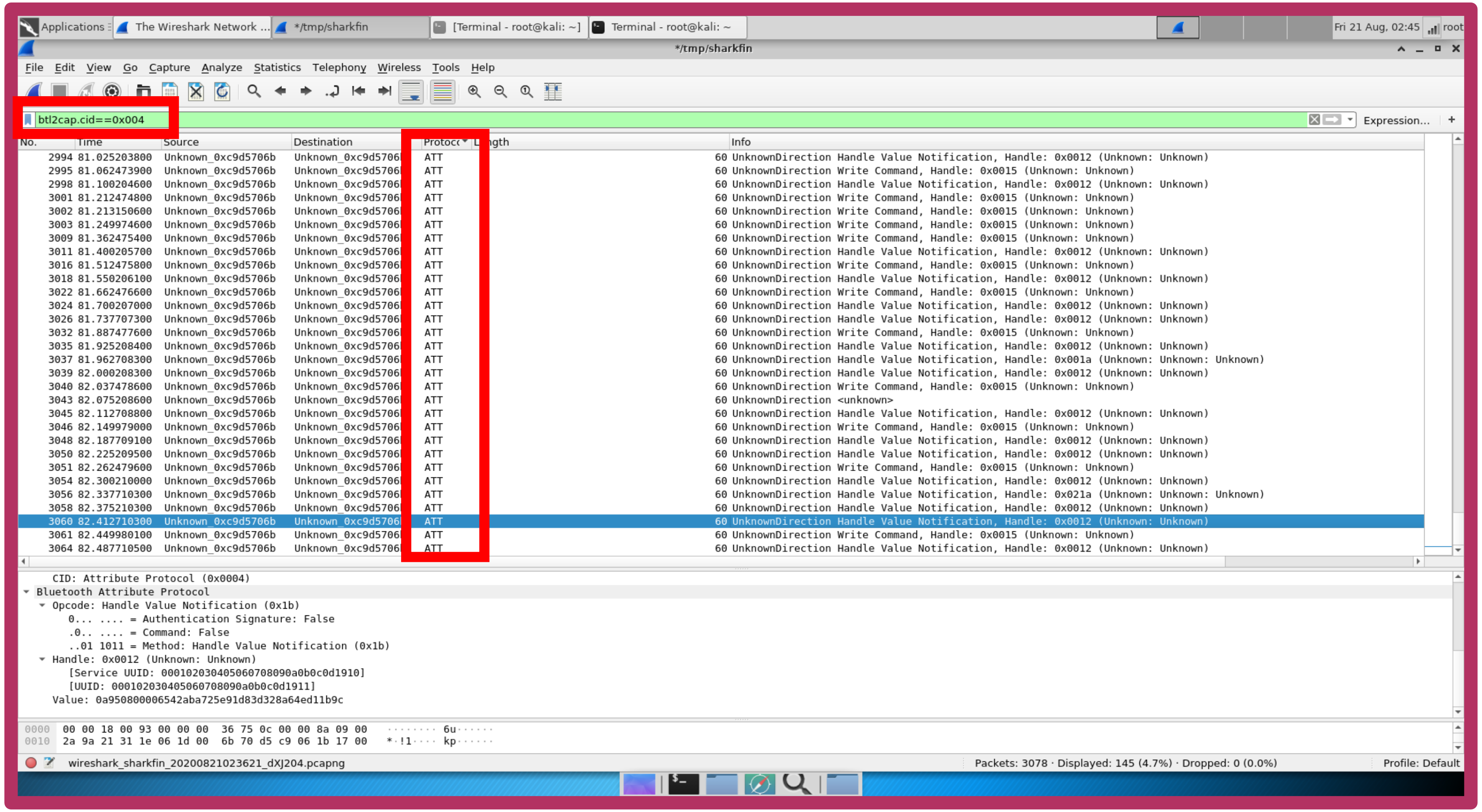
FIFO on Linux:
https://man7.org/linux/man-pages/man7/fifo.7.html#:~:text=A%20FIFO%20special%20file%20(a,writing%20it%20to%20the%20filesystem.

FIFO and pipes in Wireshark:
https://wiki.wireshark.org/CaptureSetup/Pipes

```
$ mkfifo /tmp/sharkfin
$ wireshark -k -i /tmp/sharkfin &
$ ubertooth-btle -f -c /tmp/sharkin &
```

Analyze

- **Filter**: btl2cap.cid == 0x004
- ATT Protocol


Applications
The Wireshark Network ...
*/tmp/sharkfin
[Terminal - root@kali: ~]
Terminal - root@kali: ~
Fri 21 Aug, 02:45
root
*/tmp/sharkfin
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
btl2cap.cid==0x004
```

```
No.   Time         Source              Destination        Protocol Length Info
 2994 81.025203800 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 2995 81.062473900 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 2998 81.100204600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3001 81.212474800 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3002 81.213150600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3003 81.249974600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3009 81.362475400 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3011 81.400205700 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3016 81.512475800 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3018 81.550206100 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3022 81.662476600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3024 81.700207000 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3026 81.737707300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3032 81.887477600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3035 81.925208400 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3037 81.962708300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x001a (Unknown: Unknown: Unknown)
 3039 82.000208300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3040 82.037478600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3043 82.075208600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection <unknown>
 3045 82.112708800 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3046 82.149979000 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3048 82.187709100 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3050 82.225209500 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3051 82.262479600 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3054 82.300210000 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3056 82.337710300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x021a (Unknown: Unknown: Unknown)
 3058 82.375210300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3060 82.412710300 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
 3061 82.449980100 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Write Command, Handle: 0x0015 (Unknown: Unknown)
 3064 82.487710500 Unknown_0xc9d5706b  Unknown_0xc9d5706  ATT      60 UnknownDirection Handle Value Notification, Handle: 0x0012 (Unknown: Unknown)
```

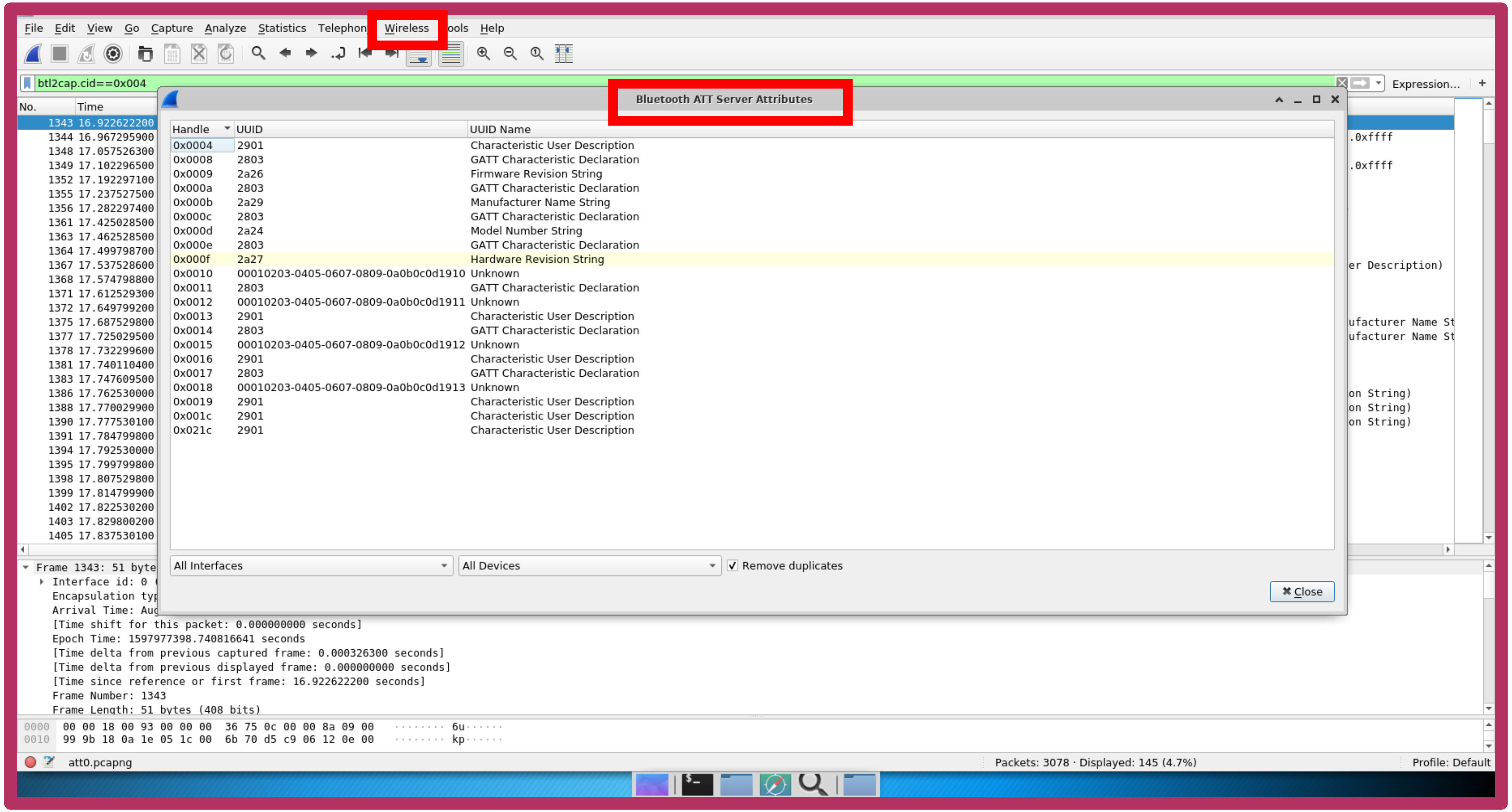
```
    CID: Attribute Protocol (0x0004)
▾ Bluetooth Attribute Protocol
  ▾ Opcode: Handle Value Notification (0x1b)
      0... .... = Authentication Signature: False
      .0.. .... = Command: False
      ..01 1011 = Method: Handle Value Notification (0x1b)
  ▾ Handle: 0x0012 (Unknown: Unknown)
      [Service UUID: 000102030405060708090a0b0c0d1910]
      [UUID: 000102030405060708090a0b0c0d1911]
    Value: 0a950800006542aba725e91d83d328a64ed11b9c
```

```
0000  00 00 18 00 93 00 00 00  36 75 0c 00 00 8a 09 00   ········ 6u······
0010  2a 9a 21 31 1e 06 1d 00  6b 70 d5 c9 06 1b 17 00   *·!1···· kp······
```

wireshark_sharkfin_20200821023621_dXJ204.pcapng   Packets: 3078 · Displayed: 145 (4.7%) · Dropped: 0 (0.0%)   Profile: Default

Analyze

- Handle
- UUID
- UUID Name



btl2cap.cid==0x004

```
No.   Time
1343 16.922622200
1344 16.967295900
1348 17.057526300
1349 17.102296500
1352 17.192297100
1355 17.237527500
1356 17.282297400
1361 17.425028500
1363 17.462528500
1364 17.499798700
1367 17.537528600
1368 17.574798800
1371 17.612529300
1372 17.649799200
1375 17.687529800
1377 17.725029500
1378 17.732299600
1381 17.740110400
1383 17.747609500
1386 17.762530000
1388 17.770029900
1390 17.777530100
1391 17.784799800
1394 17.792530000
1395 17.799799800
1398 17.807529800
1399 17.814799900
1402 17.822530200
1403 17.829800200
1405 17.837530100
```

```
.0xffff
.0xffff
er Description)
ufacturer Name St
ufacturer Name St
on String)
on String)
on String)
```

Handle	UUID	UUID Name
0x0004	2901	Characteristic User Description
0x0008	2803	GATT Characteristic Declaration
0x0009	2a26	Firmware Revision String
0x000a	2803	GATT Characteristic Declaration
0x000b	2a29	Manufacturer Name String
0x000c	2803	GATT Characteristic Declaration
0x000d	2a24	Model Number String
0x000e	2803	GATT Characteristic Declaration
0x000f	2a27	Hardware Revision String
0x0010	00010203-0405-0607-0809-0a0b0c0d1910	Unknown
0x0011	2803	GATT Characteristic Declaration
0x0012	00010203-0405-0607-0809-0a0b0c0d1911	Unknown
0x0013	2901	Characteristic User Description
0x0014	2803	GATT Characteristic Declaration
0x0015	00010203-0405-0607-0809-0a0b0c0d1912	Unknown
0x0016	2901	Characteristic User Description
0x0017	2803	GATT Characteristic Declaration
0x0018	00010203-0405-0607-0809-0a0b0c0d1913	Unknown
0x0019	2901	Characteristic User Description
0x001c	2901	Characteristic User Description
0x021c	2901	Characteristic User Description

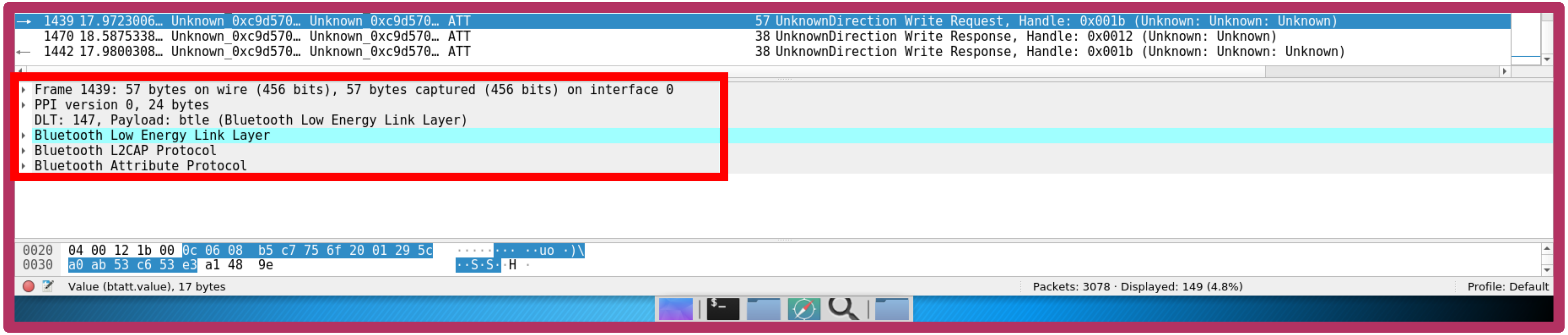
All Interfaces ▾ All Devices ▾ ☑ Remove duplicates

```
▾ Frame 1343: 51 byte
  ▸ Interface id: 0 (
    Encapsulation typ
    Arrival Time: Aug
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1597977398.740816641 seconds
    [Time delta from previous captured frame: 0.000326300 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 16.922622200 seconds]
    Frame Number: 1343
    Frame Length: 51 bytes (408 bits)
```

```
0000  00 00 18 00 93 00 00 00  36 75 0c 00 00 8a 09 00   ········ 6u······
0010  99 9b 18 0a 1e 05 1c 00  6b 70 d5 c9 06 12 0e 00   ········ kp······
```

att0.pcapng   Packets: 3078 · Displayed: 145 (4.7%)   Profile: Default

Analyze



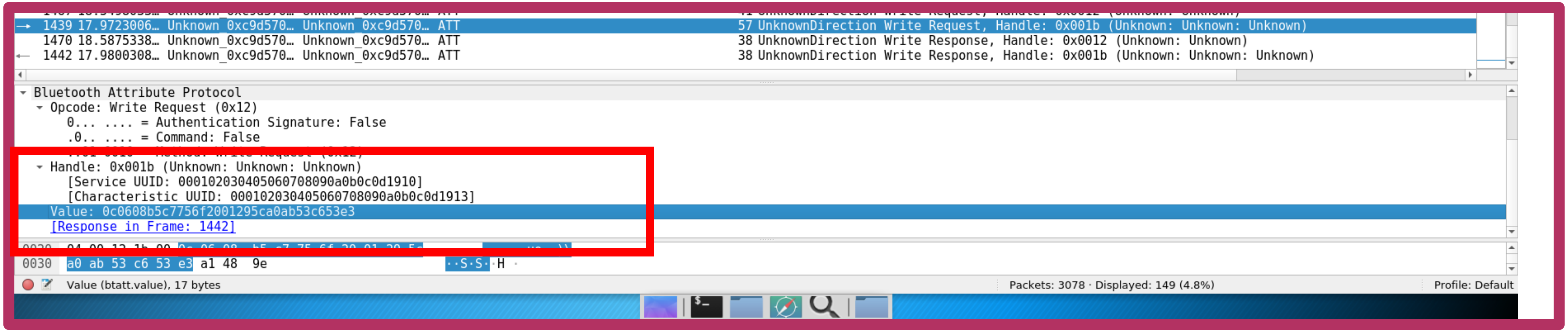
```
→ 1439 17.9723006… Unknown_0xc9d570… Unknown_0xc9d570… ATT        57 UnknownDirection Write Request, Handle: 0x001b (Unknown: Unknown: Unknown)
  1470 18.5875338… Unknown_0xc9d570… Unknown_0xc9d570… ATT        38 UnknownDirection Write Response, Handle: 0x0012 (Unknown: Unknown)
← 1442 17.9800308… Unknown_0xc9d570… Unknown_0xc9d570… ATT        38 UnknownDirection Write Response, Handle: 0x001b (Unknown: Unknown: Unknown)
```

```
› Frame 1439: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface 0
› PPI version 0, 24 bytes
  DLT: 147, Payload: btle (Bluetooth Low Energy Link Layer)
› Bluetooth Low Energy Link Layer
› Bluetooth L2CAP Protocol
› Bluetooth Attribute Protocol
```

```
0020  04 00 12 1b 00 0c 06 08  b5 c7 75 6f 20 01 29 5c   ········ ··uo ·)\
0030  a0 ab 53 c6 53 e3 a1 48  9e                        ··S·S··H ·
```

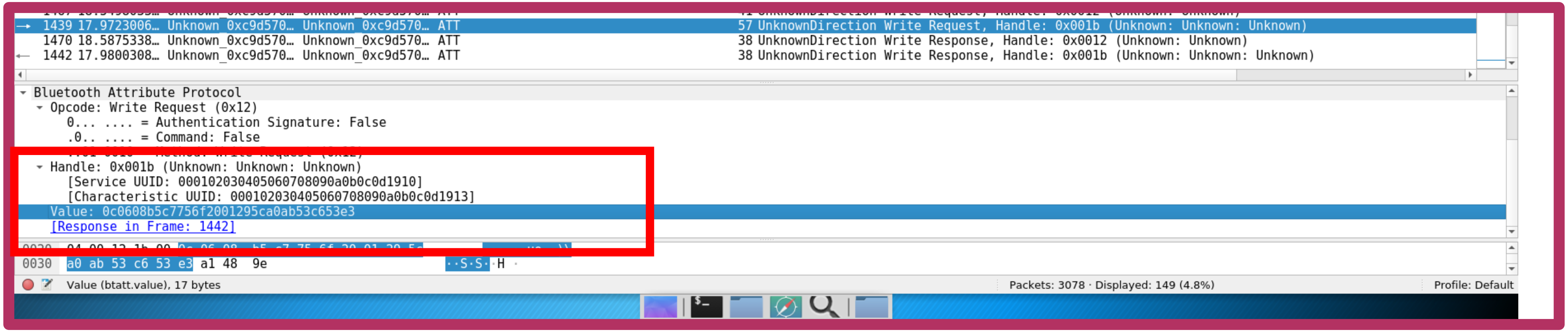
Value (btatt.value), 17 bytes   Packets: 3078 · Displayed: 149 (4.8%)   Profile: Default

Analyze



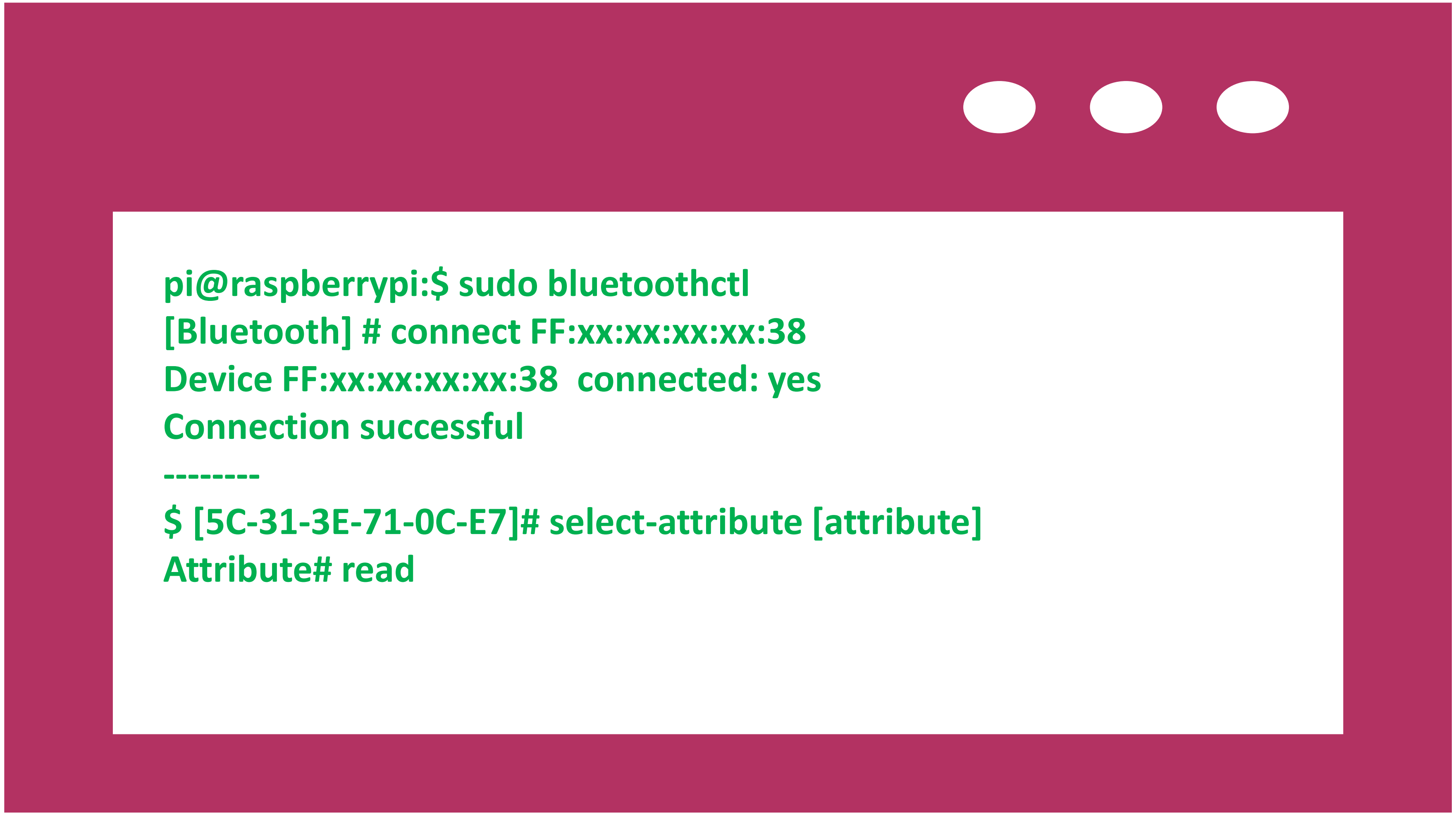
```
→ 1439 17.9723006… Unknown_0xc9d570… Unknown_0xc9d570… ATT    57 UnknownDirection Write Request, Handle: 0x001b (Unknown: Unknown: Unknown)
  1470 18.5875338… Unknown_0xc9d570… Unknown_0xc9d570… ATT    38 UnknownDirection Write Response, Handle: 0x0012 (Unknown: Unknown)
← 1442 17.9800308… Unknown_0xc9d570… Unknown_0xc9d570… ATT    38 UnknownDirection Write Response, Handle: 0x001b (Unknown: Unknown: Unknown)

▾ Bluetooth Attribute Protocol
  ▾ Opcode: Write Request (0x12)
      0... .... = Authentication Signature: False
      .0.. .... = Command: False
      ..01 0010 = Method: Write Request (0x12)
  ▾ Handle: 0x001b (Unknown: Unknown: Unknown)
      [Service UUID: 000102030405060708090a0b0c0d1910]
      [Characteristic UUID: 000102030405060708090a0b0c0d1913]
    Value: 0c0608b5c7756f2001295ca0ab53c653e3
    [Response in Frame: 1442]

0020  04 00 12 1b 00 0c 06 08 b5 c7 75 6f 20 01 29 5c
0030  a0 ab 53 c6 53 e3 a1 48 9e                         ··S·S··H ·

Value (btatt.value), 17 bytes                 Packets: 3078 · Displayed: 149 (4.8%)          Profile: Default
```

Analyze



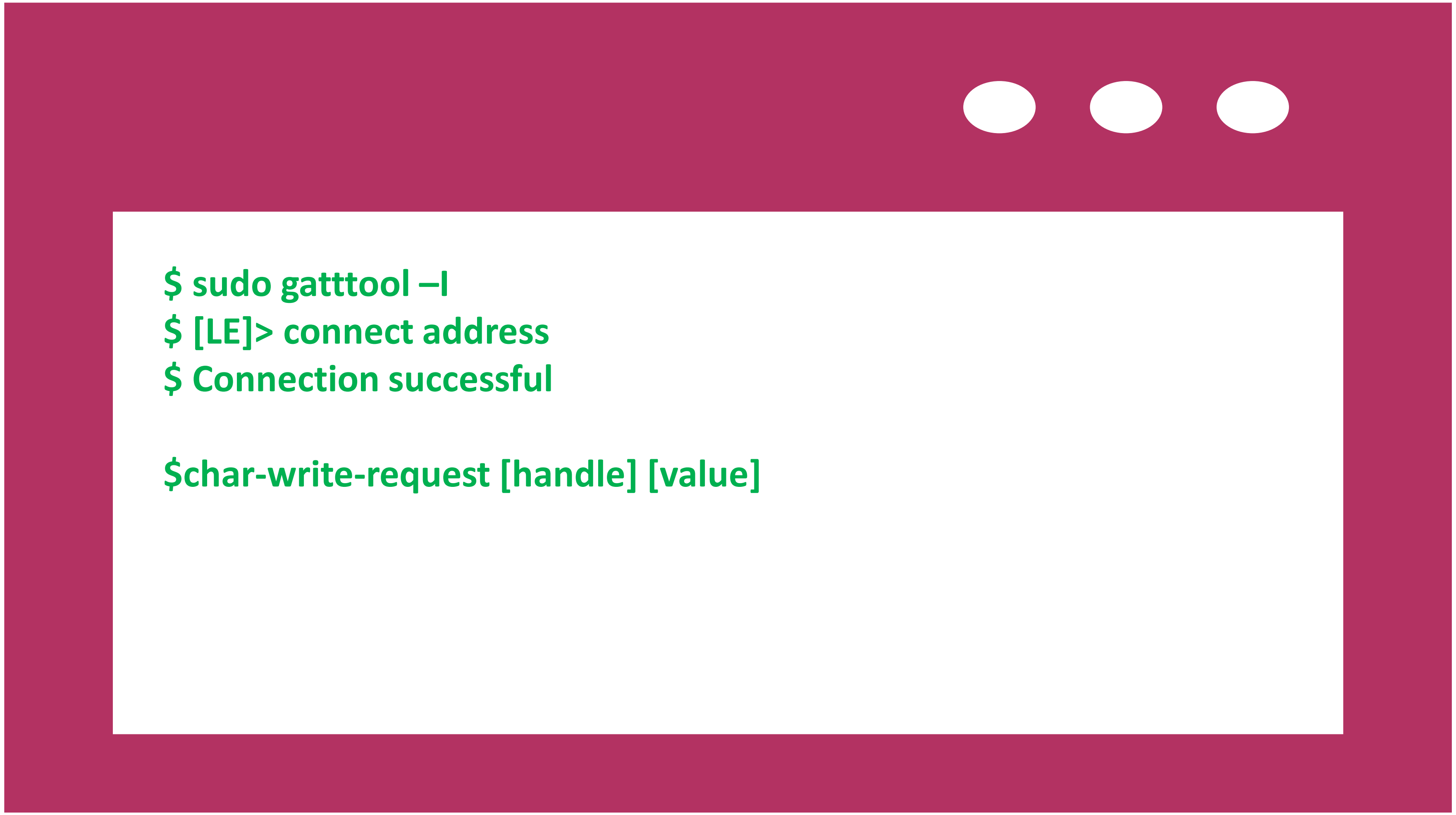
```
→ 1439 17.9723006… Unknown_0xc9d570… Unknown_0xc9d570… ATT    57 UnknownDirection Write Request, Handle: 0x001b (Unknown: Unknown: Unknown)
  1470 18.5875338… Unknown_0xc9d570… Unknown_0xc9d570… ATT    38 UnknownDirection Write Response, Handle: 0x0012 (Unknown: Unknown)
← 1442 17.9800308… Unknown_0xc9d570… Unknown_0xc9d570… ATT    38 UnknownDirection Write Response, Handle: 0x001b (Unknown: Unknown: Unknown)

▾ Bluetooth Attribute Protocol
  ▾ Opcode: Write Request (0x12)
      0... .... = Authentication Signature: False
      .0.. .... = Command: False
  ▾ Handle: 0x001b (Unknown: Unknown: Unknown)
      [Service UUID: 000102030405060708090a0b0c0d1910]
      [Characteristic UUID: 000102030405060708090a0b0c0d1913]
    Value: 0c0608b5c7756f2001295ca0ab53c653e3
    [Response in Frame: 1442]

0030  a0 ab 53 c6 53 e3 a1 48  9e                    ··S·S·H ·

Value (btatt.value), 17 bytes                    Packets: 3078 · Displayed: 149 (4.8%)          Profile: Default
```

Use Bluetoothctl



```
pi@raspberrypi:$ sudo bluetoothctl
[Bluetooth] # connect FF:xx:xx:xx:xx:38
Device FF:xx:xx:xx:xx:38  connected: yes
Connection successful
--------
$ [5C-31-3E-71-0C-E7]# select-attribute [attribute]
Attribute# read
```

Use Gatttool



```
$ sudo gatttool –I
$ [LE]> connect address
$ Connection successful

$char-write-request [handle] [value]
```

Sources to Get You Started

1. **Getting Started With Bluetooth Low Energy: Tools and Techniques for Low-Power Networking by Kevin Townsend, Carles Cufí, Akiba & Robert Davidson**
2. **The IoT Hacker’s Handbook: A Practical Guide to Hacking the Internet of Things by Aditya Gupta**
3. https://learn.adafruit.com/install-bluez-on-the-raspberry-pi/installation
4. http://www.bluez.org/

BREAKING BOUNDARIES
The Diana Initiative
BYTE by BYTE
Aug 21-22 2020
Virtual Conference

THANK YOU!