Bluetooth Landscape Exploration & Enumeration Platform (B.L.E.E.P.)

Paul A. Wortman, PhD
Mauddib28

August 9, 2024

Table of contents

Whoami

- PhD
- Bluetooth Security Researcher
- Research Scientist

Whatisthis?

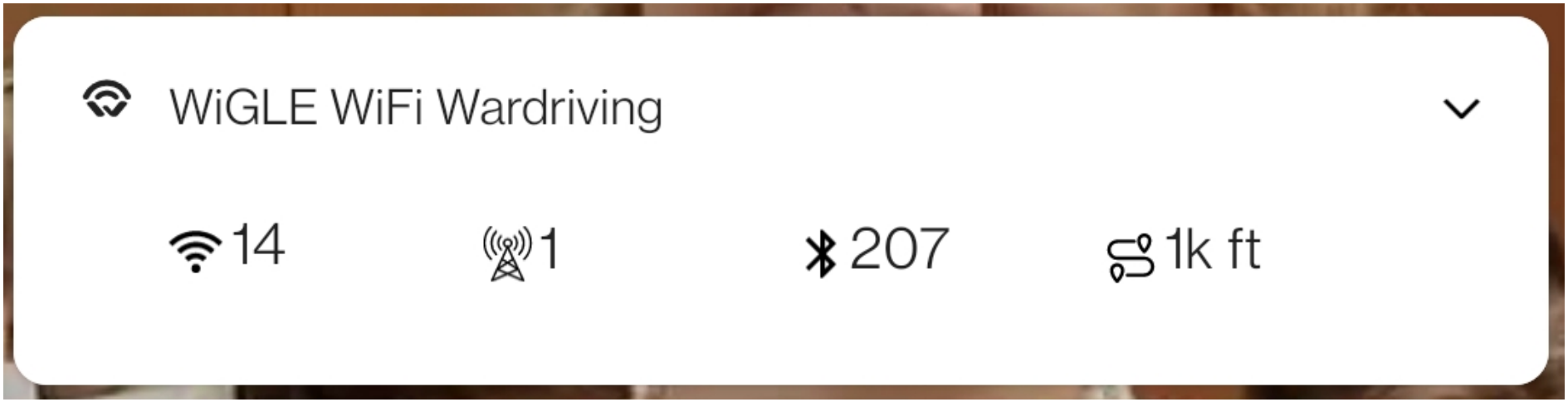


Figure 1: Wiggle War Drive - Landing Strip

“Lightning” Headphones That Require Bluetooth

Josh Whiton:

> A crazy experience — I lost my earbuds in a remote town in Chile, so tried buying a new pair at the airport before flying out. But the new wired, iPhone, lightning-cable headphones didn’t work. Strange.

What is this about?

- What are the underlying technologies BLEEP is built upon
- Technical details one will need to know when getting under the hood
- Advanced BLE interaction + structures
- Identifying Bluetooth landscape and wildlife (Demo baby demo!)

Knowledge Review

Bluetooth & D-Bus

BR/ED vs BLE

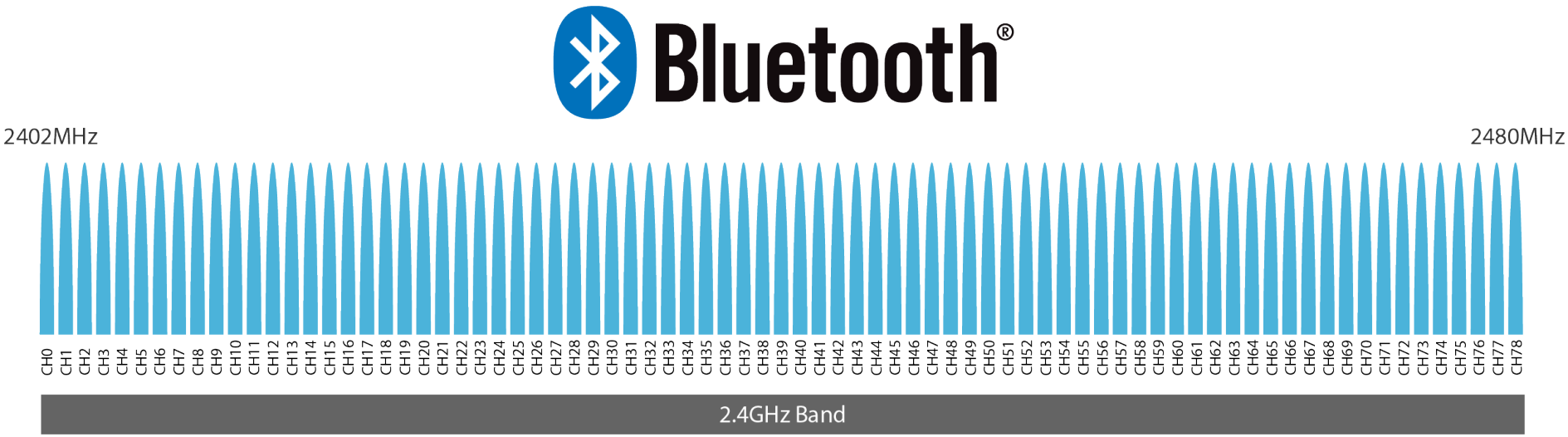


Figure 3: Bluetooth Classic Spectrum

- 79 number of (pairing) channels
- Freq. hoping to minimize interference
  - Defense in depth
- Larger power requirements minimize effectiveness for embedded systems

BR/ED vs BLE

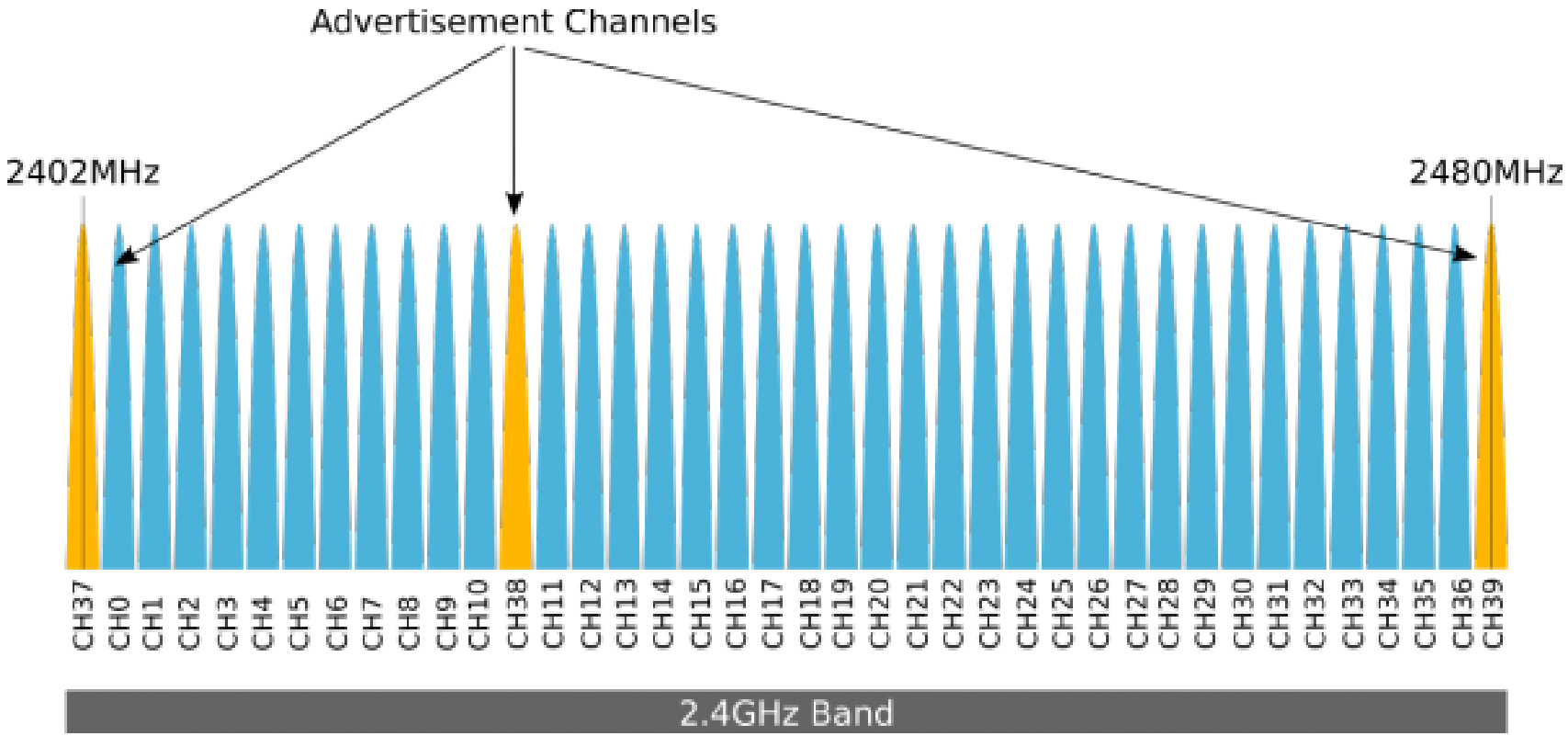


Figure 4: Bluetooth Low Energy Spectrum

- Three pairing channels; simplified sniffing of pairing
- Post connection return to classic frequency hopping
  - Capture of handshake provides easier eavesdropping
- Lower power requirements; ideal for embedded systems

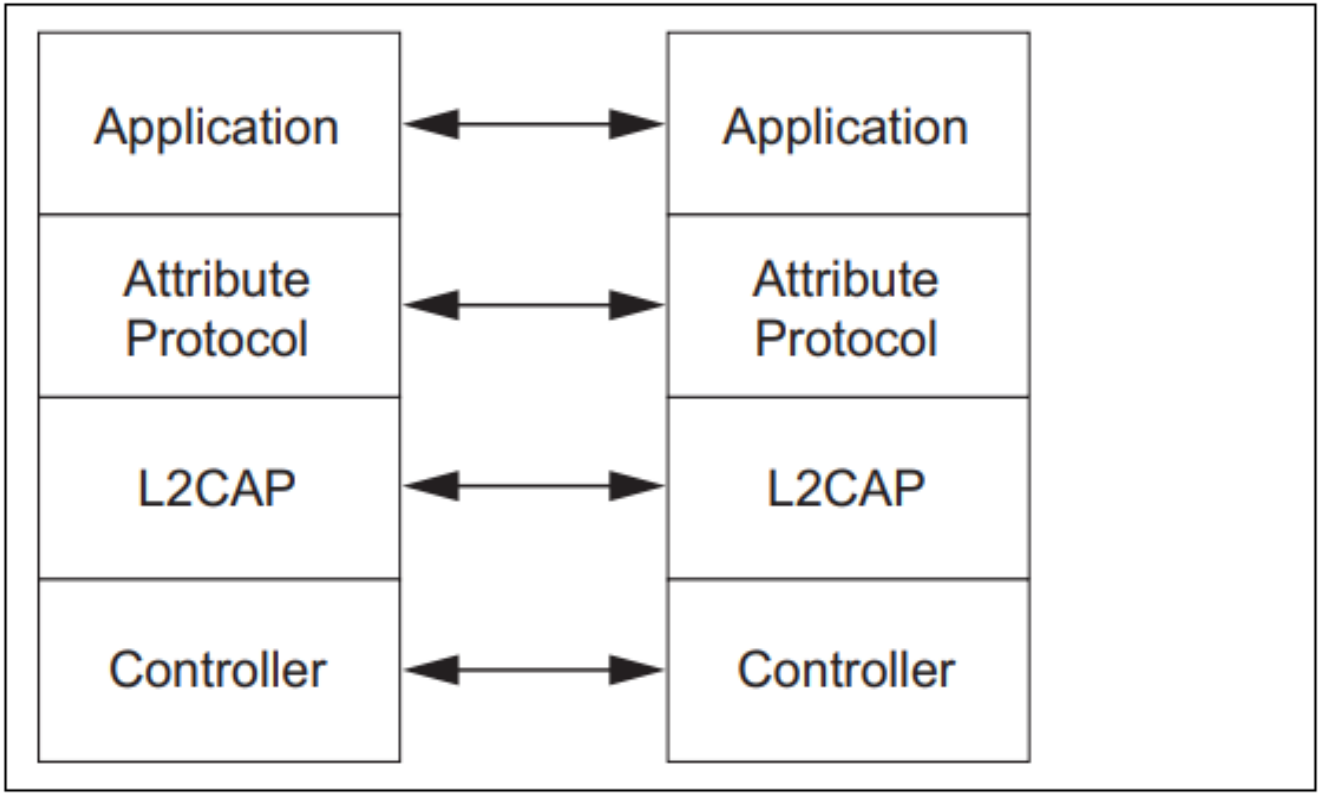


Figure 5: Protocol model [14]

- GATT Server
  - Inability to “change” a GATT server once initialized
  - Flavor differences; Arduino will not allow writing to Descriptors

BLE - Must Know Basics - High-level BLE Structure

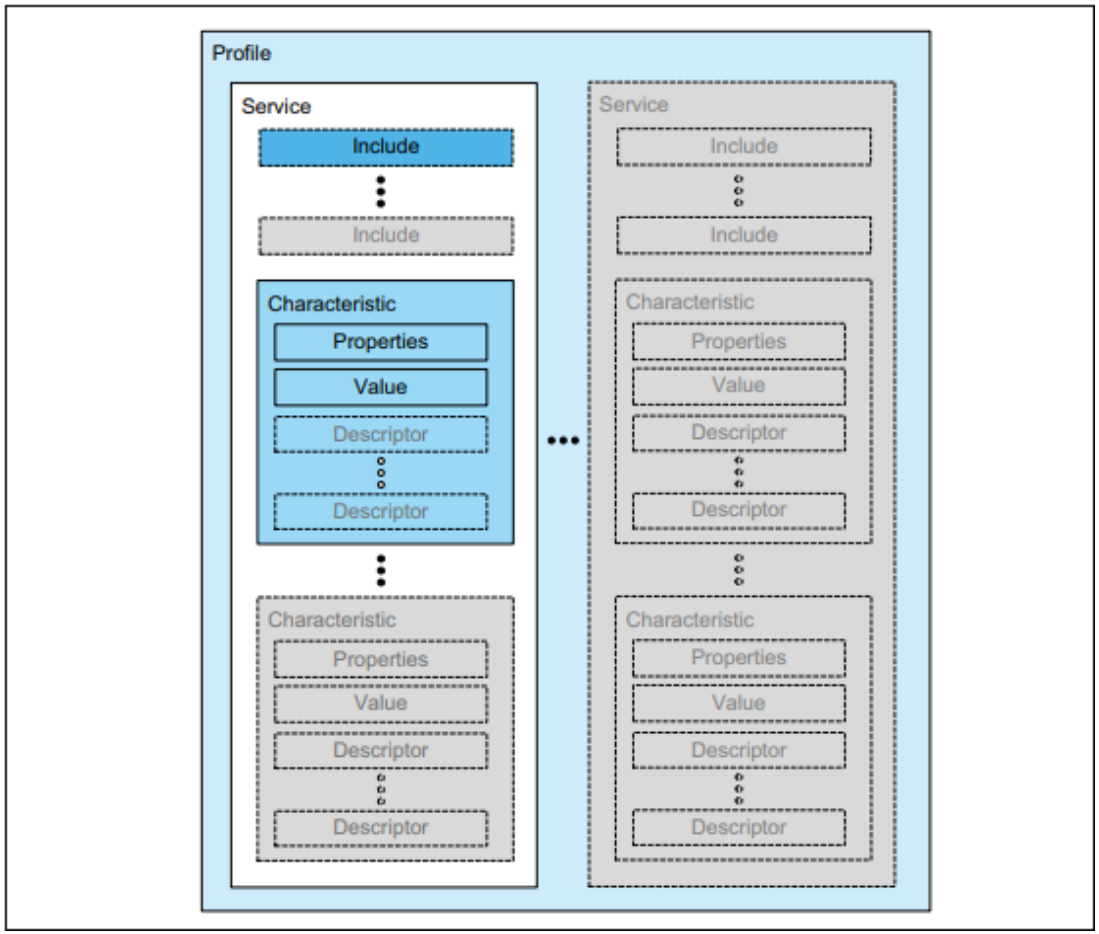


Figure 6: GATT-Based Profile hierarchy [14]

- Device, Services, Characteristics, Descriptors
- Characteristics have to be read once to populate data
  - Second read allows determining the data

Basics

- D-Bus developed and maintained by freedesktop.org
  - URL: www.freedesktop.org/wiki/Software/dbus
- D-Bus uses code agnostic low-level API reference implementation that works/portable to any Linux or UNIX flavor
  - Make use of Python API calls with tool
- D-Bus is a message bus system; providing a simple way for applications to talk to one another
  - Provides a system daemon and a per-user-login-session daemon [1]

B.L.E.E.P.

Methodology & Capabilities

What the BLEEP is this?

Bluetooth Landscape Exploration & Enumeration Platform
(B.L.E.E.P.)

Figure 7: The BLEEP-ing Logo

Methodology

- (Re-)Create basics using CLI tools
- Replicate basic structure in code
- Replicate CLI operations with code structures (classes)
- Sanity check “sight” from both CLI tools and BLEEP

Sanity Check via Tools

- Learning the basics with tools
  - dbus-send, dbus-monitor, gdbus, busctl
  - btmon, btmgmt, bluetoothctl

Sanity Check via Tools - bluetoothctl

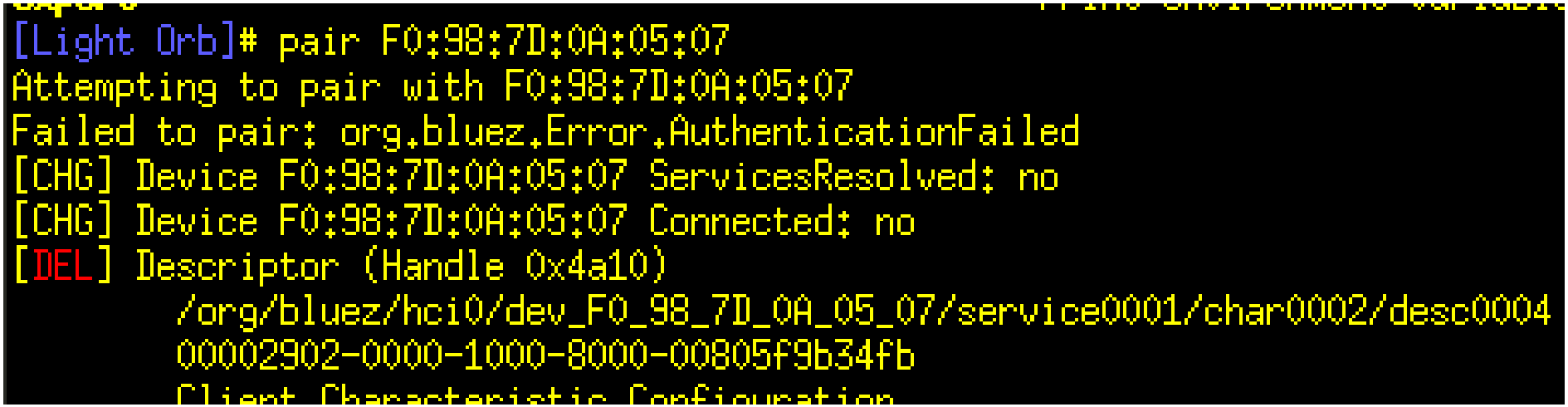


Figure 8: bluetoothctl - Light Orb Test

Scanning and Discovery

- Can I see THE ORB?!?!
- Can I track THE ORB?!?
- Can I connect to THE ORB?!
- Can I disconnect from THE ORB?

Read + Write

- Gotta ENUMERATE!
  - What structures are necessary?
  - What info MUST be tracked?
  - How can I find/recall this info?
- Maybe reads are easy?
- Writes are not easy.... unless you know how to read....
- Pretty print that BLE Device!

Time for D-Bus

Interfaces, D-Bus, and You

D-Bus Interfaces - Basics

- D-Bus interaction and enumeration is done via a series of interfaces
- Interfaces exist for each “layer” of interactivity with the GATT Server

D-Bus Interfaces - busctl

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl tree org.bluez
└─/org
  └─/org/bluez
    └─/org/bluez/hci0
      ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6
      │ ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001
      │ │ └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002
      │ │   └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002/desc0004
      │ └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0029
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002b
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002d
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002f
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0031
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0033
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0035
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0037
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0039
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003b
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003d
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003f
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0041
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0043
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0045
```

D-Bus Interfaces - Little Bit Deeper Now

- Several types of D-Bus interfaces; each with separate purposes
  - Methods = Functions
  - Signals =?? Periodic Communications ?? __(o.O)__/
  - Properties = Properties
- Examples of interfaces:
  - Device - org.bluez.Device1
  - GATT Service - org.bluez.GattService1
  - Properties - org.freedesktop.DBus.Properties
  - Introspection - org.freedesktop.DBus.Introspectable

D-Bus Interfaces - busctl - Introspection

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl introspect org.bluez /org/bluez/hci0/dev_CC_50_E3_B6_BC_A6
NAME                                TYPE      SIGNATURE RESULT/VALUE                                FLAGS
org.bluez.Device1                   interface -         -                                           -
.CancelPairing                      method    -         -                                           -
.Connect                            method    -         -                                           -
.ConnectProfile                     method    s         -                                           -
.Disconnect                         method    -         -                                           -
.DisconnectProfile                  method    s         -                                           -
.Pair                               method    -         -                                           -
.Adapter                            property  o         "/org/bluez/hci0"                           emits-change
.Address                            property  s         "CC:50:E3:B6:BC:A6"                         emits-change
.AddressType                        property  s         "public"                                    emits-change
.Alias                              property  s         "2b00042f7481c7b056c4b410d28f33cf"          emits-change writable
.Appearance                         property  q         -                                           emits-change
.Blocked                            property  b         false                                       emits-change writable
.Class                              property  u         -                                           emits-change
.Connected                          property  b         true                                        emits-change
.Icon                               property  s         -                                           emits-change
.LegacyPairing                      property  b         false                                       emits-change
.ManufacturerData                   property  a{qv}     -                                           emits-change
.Modalias                           property  s         -                                           emits-change
.Name                               property  s         "2b00042f7481c7b056c4b410d28f33cf"          emits-change
.Paired                             property  b         false                                       emits-change
.RSSI                               property  n         -                                           emits-change
.ServiceData                        property  a{sv}     -                                           emits-change
.ServicesResolved                   property  b         true                                        emits-change
.Trusted                            property  b         false                                       emits-change writable
.TxPower                            property  n         -                                           emits-change
.UUIDs                              property  as        3 "000000ff-0000-1000-8000-00805f9b34fb     emits-change
.WakeAllowed                        property  b         -                                           emits-change writable
org.freedesktop.DBus.Introspectable interface -         -                                           -
.Introspect                         method    -         s                                           -
org.freedesktop.DBus.Properties     interface -         -                                           -
.Get                                method    ss        v                                           -
.GetAll                             method    s         a{sv}                                       -
.Set                                method    ssv       -                                           -
```

D-Bus Interfaces - Introspection

- Compare returned data against 'busctl' visibility
- Create rough structures
  - Enough to proceed with enumeration
- Understanding: Each interface is explored via an **introspection** interface
  - Using the introspection interface's introspect method **(-_-;)**

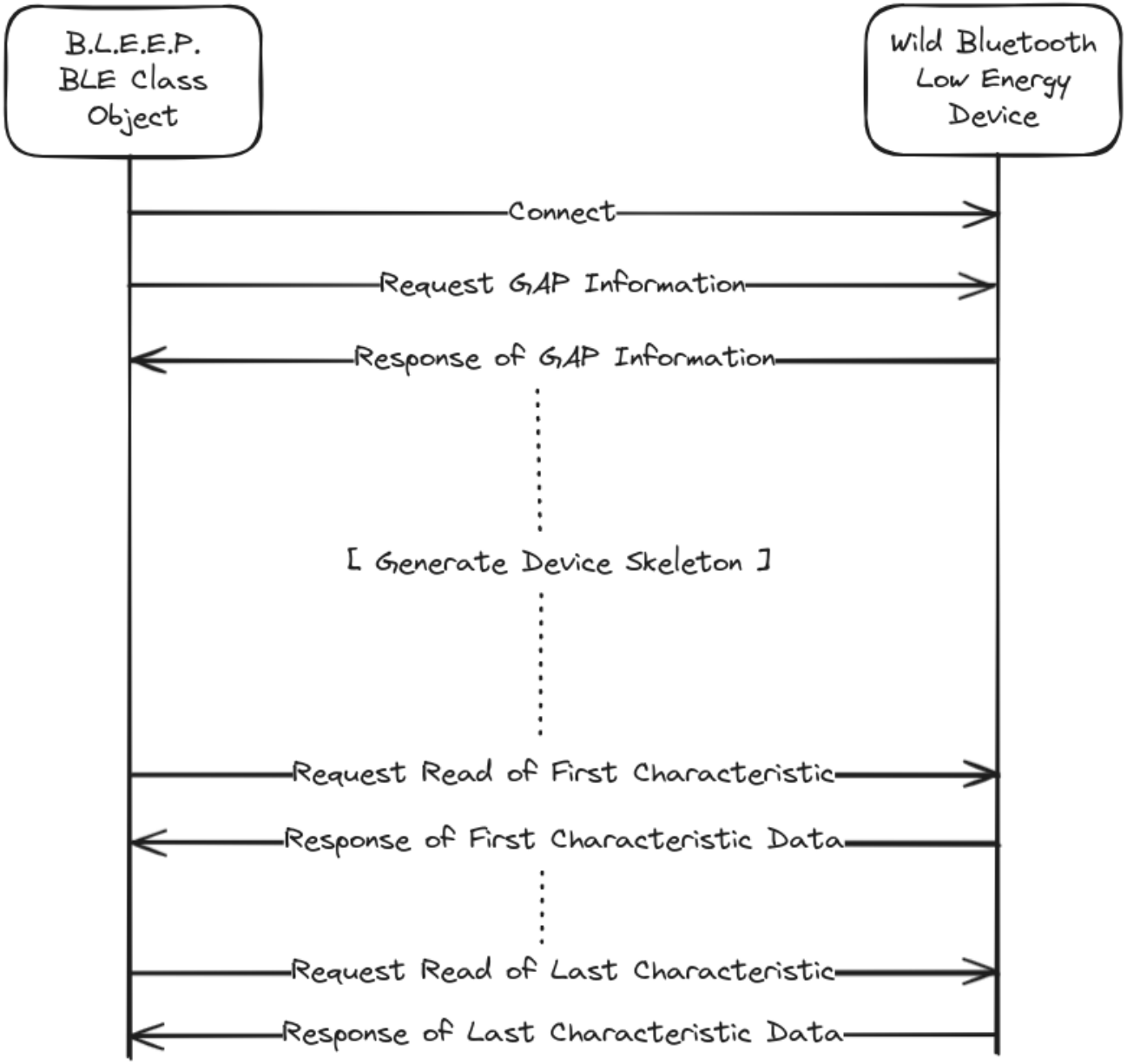
Abstracting Introspection

- Produces an XML map of the introspected interface's information
  - Contains (1) [Sub-]Interfaces and (2) [Sub-]Nodes associated to the introspected interface

```
[!] Introspection E-Tree Information:          <Element 'node' at 0x74b486f0c2c0>
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Introspectable'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.bluez.GattService1'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Properties'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char002f'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0031'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0034'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0037'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003a'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003d'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003f'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0042'} ]
[!] Introspection E-Tree Information:          <Element 'node' at 0x74b486f0cc70>
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Introspectable'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.bluez.GattCharacteristic1'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Properties'} ]
```

Figure 11: Etree XML Example - Light Orb

Simple Method of Enumeration



Signature-ing a Time Sink - Characteristic

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl introspect org.bluez /org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002
NAME                                TYPE      SIGNATURE RESULT/VALUE                              FLAGS
org.bluez.GattCharacteristic1       interface -         -                                         -
.AcquireNotify                      method    a{sv}     hq                                        -
.AcquireWrite                       method    a{sv}     hq                                        -
.ReadValue                          method    a{sv}     ay                                        -
.StartNotify                        method    -         -                                         -
.StopNotify                         method    -         -                                         -
.WriteValue                         method    aya{sv}   -                                         -
.Flags                              property  as        1 "indicate"                              emits-change
.MTU                                property  q         500                                       emits-change
.NotifyAcquired                     property  b         -                                         emits-change
.Notifying                          property  b         false                                     emits-change
.Service                            property  o         "/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/   emits-change
.UUID                               property  s         "00002a05-0000-1000-8000-00805f9b34fb"    emits-change
.Value                              property  ay        0                                         emits-change
.WriteAcquired                      property  b         -                                         emits-change
org.freedesktop.DBus.Introspectable interface -         -                                         -
.Introspect                         method    -         s                                         -
org.freedesktop.DBus.Properties     interface -         -                                         -
.Get                                method    ss        v                                         -
.GetAll                             method    s         a{sv}                                     -
.Set                                method    ssv       -                                         -
.PropertiesChanged                  signal    sa{sv}as  -                                         -
```

Figure 13: busctl - Inspecting BlueZ Characteristic

Signature-ing a Time Sink

- Reading: straight forward and easy
  - Ex: `characteristic_interface.ReadValue({})`
  - Returns: `dbus.Array([dbus.Byte(0)], signature=dbus.signature('y'))`

Reads + Writes - Read Method I

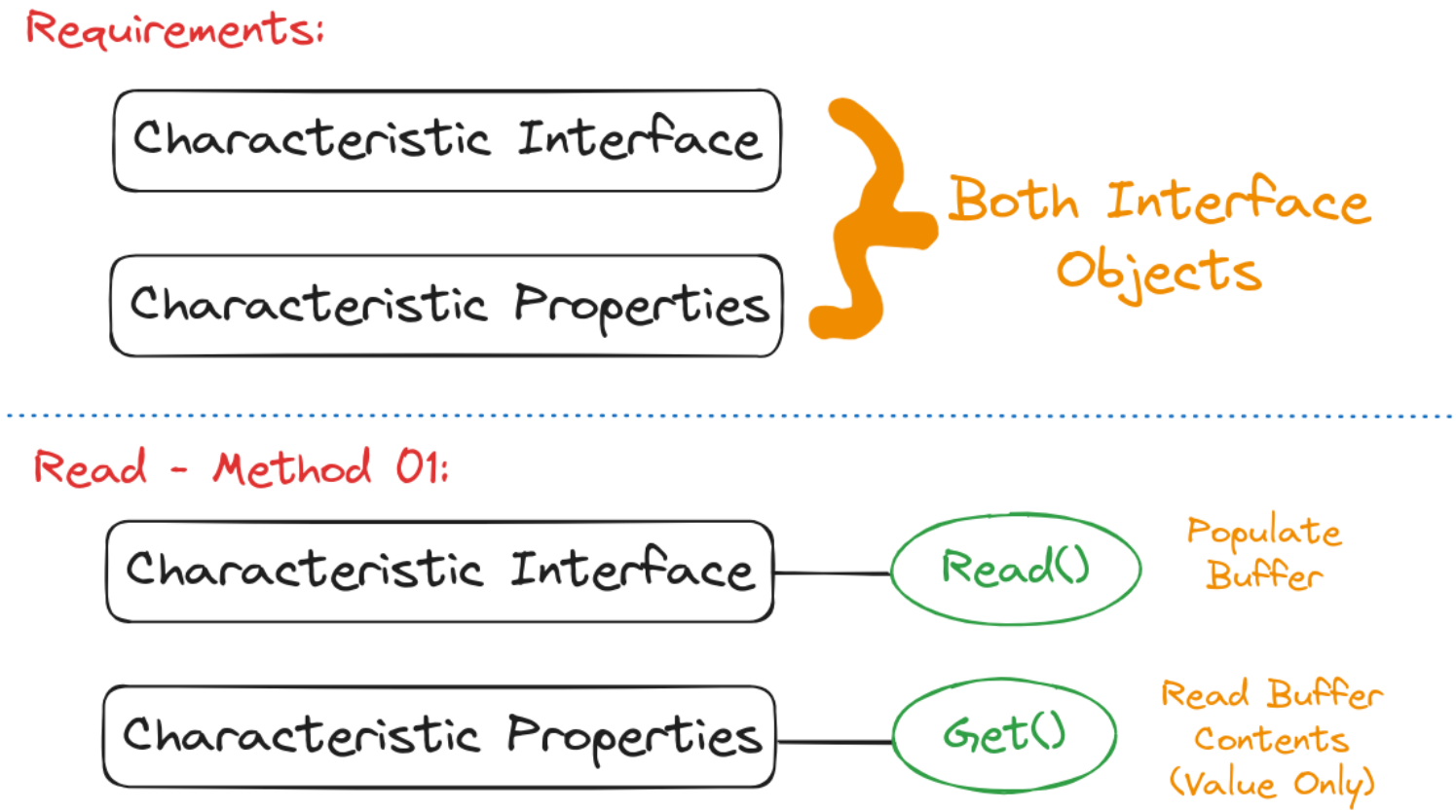


Figure 14: I/O - Read Method 01

Reads + Writes - Read Method II

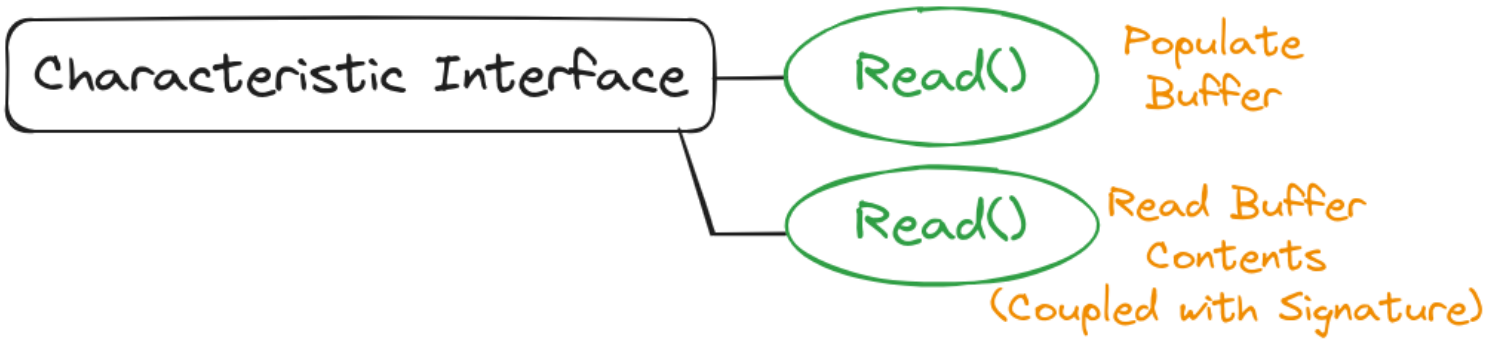


Figure 15: I/O - Read Method 02

Signature-ing a Time Sink

- Writing: more complex with lots of trail/error
  - Ex: characteristic_interface.WriteValue("1", {})
  - Returns: `ERROR: dbus.connection: Unable to set arguments ('1', {}) according to signature 'aya{sv}': < class 'TypeError' >: an integer is required (got type str)`
- Requirement was to pass the value as an array
  - Ex: `.WriteValue([1], {})`
  - *Note:* Information that can be decoded from the signature

Reads + Writes - Write Method

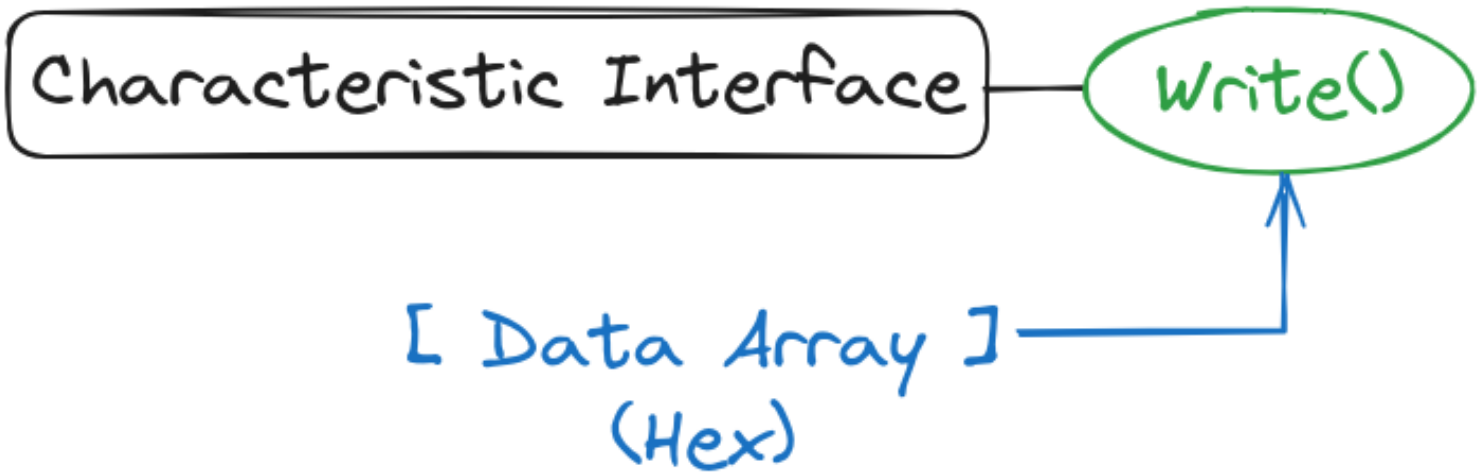


Figure 16: I/O - Write Method

Logging

- Global-variable log files
  - Set into /tmp
  - Differentiate important, general, debug, etc.
- Gotta reconstruct actions when separate threads are running (signals....)

Log Signaling D-Bus

Use those Signals!

What are signals

- Emissions that contain updates to information related to an interface
- Variety of information
  - Interface
  - Changed
  - Invalidated
- Attached to some Notify method of the interface

Automation + Logging

- Automated functionality for signal receiver, signal generation, and other necessary structures
- Implemented logging operation for post-operation analysis

Automation + Logging

```
[!] Received Signal! Debugging Signal   -       Catchall
Args:
        Interface:      org.bluez.GattCharacteristic1
        Changed:        dbus.Dictionary({dbus.String('Value'): dbus.Array([dbus.Byte(49), dbu
        Invalidated:    dbus.Array([], signature=dbus.Signature('s'))
Kwargs:
[+] Signal Debug Complete
```

Figure 17: Signal Testing - Debug Logging

What is required

- Intent.... Whatever that means....
- The name of the Signal
- Means of capturing the signal
- Callback function to process/act on the signal + associated data

Emissions - Signals + Notifications - Documentation Research

- Minimum steps for setting up notify listener:
  - Find a characteristic that has a Notify flag *(e.g. BLE CTF char003f; handle 0x0040)*
  - Create the associated characteristic interface
  - Toggle Notify using the characteristic interface
  - Create a signal receiver to capture the intended signal
  - Run the GLib.MainLoop()

What to expect when expecting

- Interface, Changed, Invalidated information
  - Had to generalize signal capture to figure this one out....
- D-Bus requires D-Engine!! (GLib.Mainloop!)
- Cannot listen to the radio station without the engine running/on

Cartography via D-Bus

Flippin' Around the Fruit

Safari Time - Apple iPhone Dive - Auth Error

```
< ACL Data TX: Handle 75 flags 0x00 dlen 7
      ATT: Read Request (0x0a) len 2
        Handle: 0x001b
> HCI Event: Number of Completed Packets (0x13) plen 5
        Num handles: 1
        Handle: 75
        Count: 1
> ACL Data RX: Handle 75 flags 0x02 dlen 9
      ATT: Error Response (0x01) len 4
        Read Request (0x0a)
        Handle: 0x001b
        Error: Insufficient Authentication (0x05)
< ACL Data TX: Handle 75 flags 0x00 dlen 11
      SMP: Pairing Request (0x01) len 6
        IO capability: NoInputNoOutput (0x03)
        OOB data: Authentication data not present (0x00)
        Authentication requirement: No bonding, No MITM, SC, No Keypresses, CT2 (0x28)
        Max encryption key size: 16
        Initiator key distribution: <none> (0x08)
        Responder key distribution: IdKey LinkKey (0x0a)
> HCI Event: Number of Completed Packets (0x13) plen 5
        Num handles: 1
        Handle: 75
        Count: 1
> ACL Data RX: Handle 75 flags 0x02 dlen 6
      SMP: Pairing Failed (0x05) len 1
        Reason: Unspecified reason (0x08)
@ MGMT Event: Authentication Failed (0x0011) plen 8
        LE Address: 7D:DC:43:0C:E9:C0 (Resolvable)
        Status: Authentication Failed (0x05)
```

Safari Time - Apple iPhone Dive - Remote Disconnect

```
> ACL Data RX: Handle 75 flags 0x02 dlen 6
      SMP: Pairing Failed (0x05) len 1
        Reason: Unspecified reason (0x08)
@ MGMT Event: Authentication Failed (0x0011) plen 8
        LE Address: 7D:DC:43:0C:E9:C0 (Resolvable)
        Status: Authentication Failed (0x05)
@ MGMT Event: Authentication Failed (0x0011) plen 8
        LE Address: 7D:DC:43:0C:E9:C0 (Resolvable)
        Status: Authentication Failed (0x05)
< HCI Command: Disconnect (0x01|0x0006) plen 3
        Handle: 75
        Reason: Authentication Failure (0x05)
> HCI Event: Command Status (0x0f) plen 4
      Disconnect (0x01|0x0006) ncmd 1
        Status: Success (0x00)
> HCI Event: Disconnect Complete (0x05) plen 4
        Status: Success (0x00)
        Handle: 75
        Reason: Connection Terminated By Local Host (0x16)
@ MGMT Event: Device Disconnected (0x000c) plen 8
        LE Address: 7D:DC:43:0C:E9:C0 (Resolvable)
        Reason: Connection terminated by local host (0x02)
@ MGMT Event: Device Disconnected (0x000c) plen 8
        LE Address: 7D:DC:43:0C:E9:C0 (Resolvable)
        Reason: Connection terminated by local host (0x02)
```

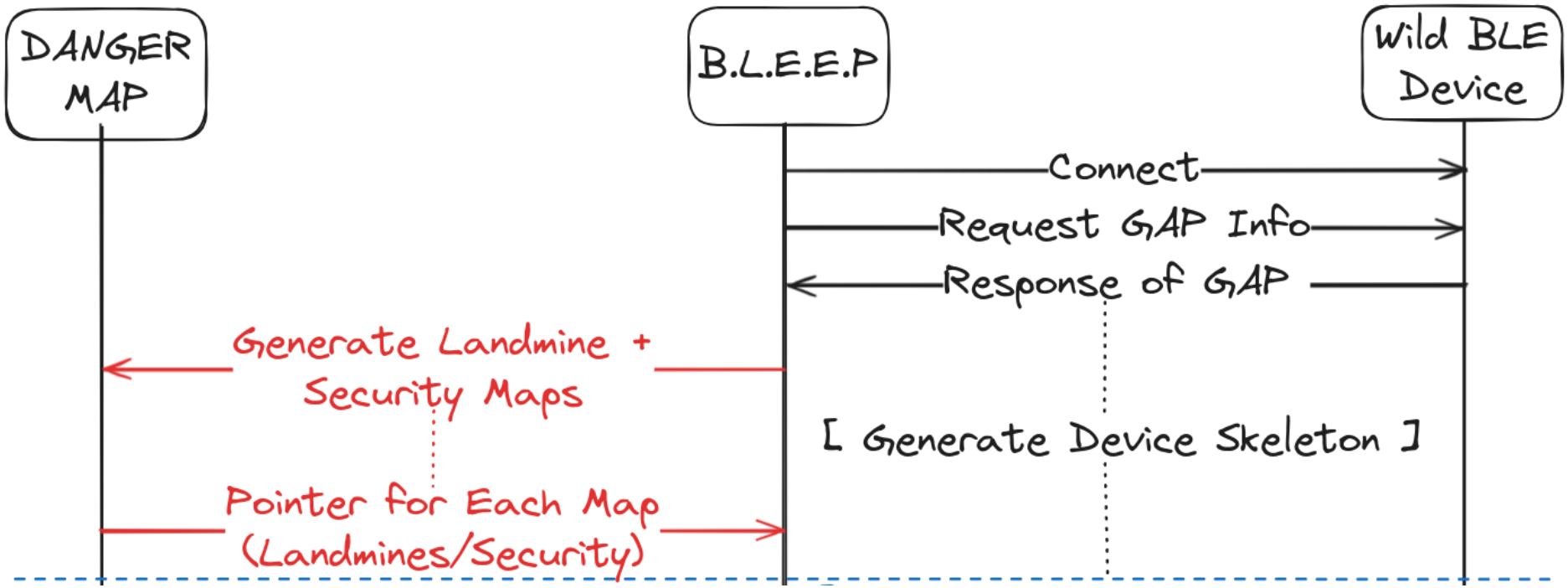
Safari Time - Apple iPhone Dive - Err Resp II

```
< ACL Data TX: Handle 75 flags 0x00 dlen 7
      ATT: Read Request (0x0a) len 2
        Handle: 0x0011
> HCI Event: Number of Completed Packets (0x13) plen 5
        Num handles: 1
        Handle: 75
        Count: 1
> ACL Data RX: Handle 75 flags 0x02 dlen 9
      ATT: Error Response (0x01) len 4
        Read Request (0x0a)
        Handle: 0x0011
        Error: Insufficient Authentication (0x05)
< ACL Data TX: Handle 75 flags 0x00 dlen 11
      SMP: Pairing Request (0x01) len 6
        IO capability: NoInputNoOutput (0x03)
        OOB data: Authentication data not present (0x00)
        Authentication requirement: No bonding, No MITM, SC, No Keypresses, CT2 (0x28)
        Max encryption key size: 16
        Initiator key distribution: <none> (0x08)
        Responder key distribution: IdKey LinkKey (0x0a)
> HCI Event: Number of Completed Packets (0x13) plen 5
        Num handles: 1
        Handle: 75
        Count: 1
> ACL Data RX: Handle 75 flags 0x02 dlen 6
      SMP: Pairing Failed (0x05) len 1
        Reason: Unspecified reason (0x08)
@ MGMT Event: Authentication Failed (0x0011) plen 8
        LE Address: 7F:25:A6:ED:7F:16 (Resolvable)
        Status: Authentication Failed (0x05)
```

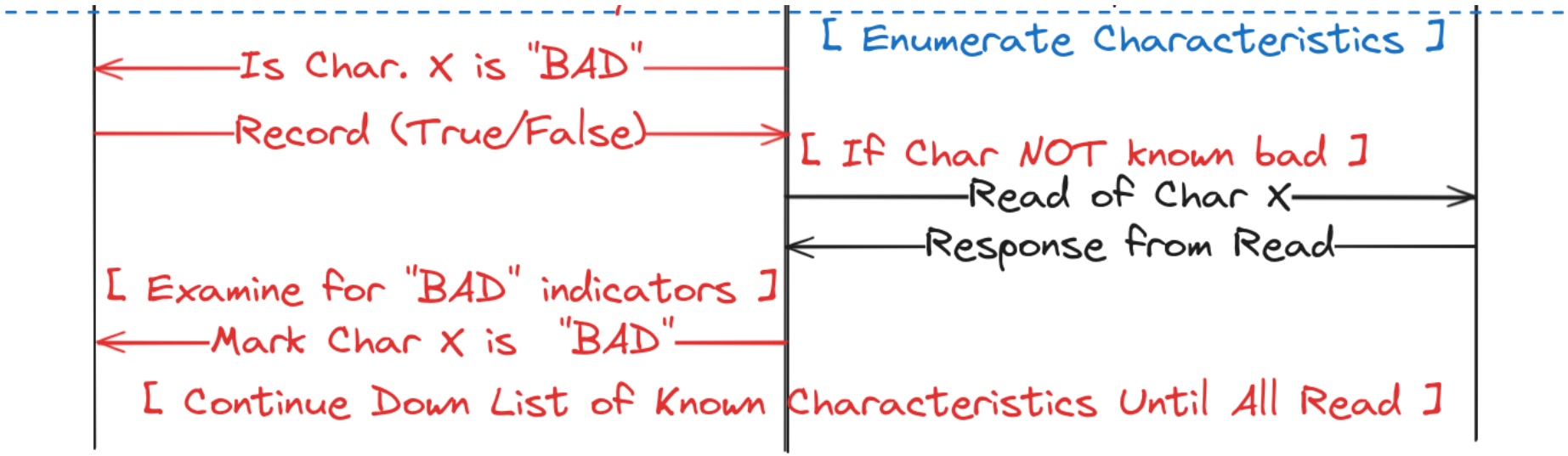
Safari Time - Decoding Scatt(ered data)

- Notice pattern of `Error Resp` to `Pairing Failure`
  - Enumeration of devices leads to disconnection occurring
  - Not initiated by **BLEEP** tool
- List of Observed Handles
  - `0x001b` - Laptop, Apple (iPhones, Watches, iPad), Unknowns
  - `0x0019` - Light Orb
  - `0x003b` - Unknowns
  - `0x002a` - Unknowns
- What do we notice about this data?
  - Attempt to read Characteristic data
  - Notice “tripping” of “panic reaction” for device disconnect

Safari Augmentation - New Enumeration



Safari Augmentation - New Enumeration

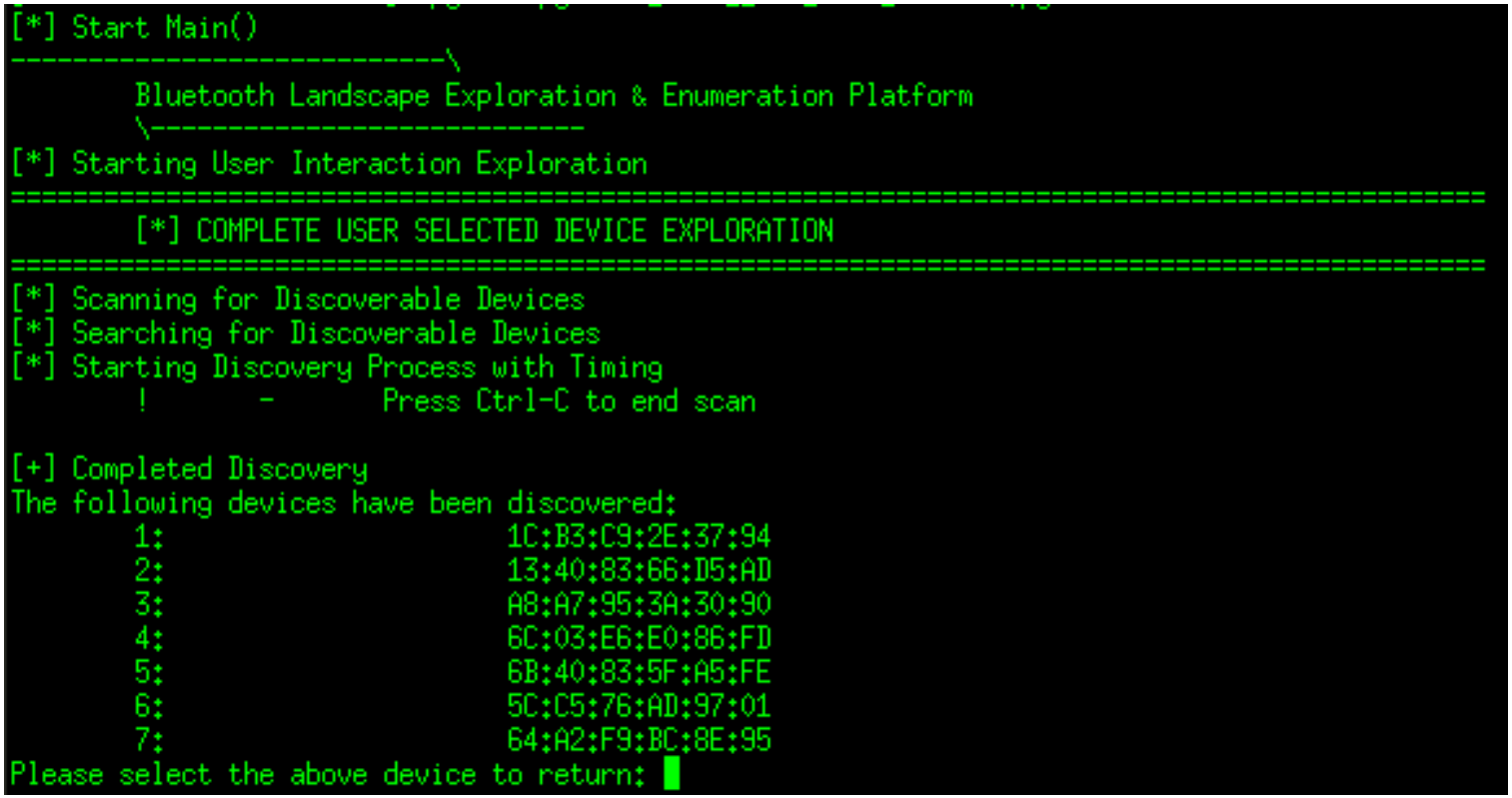


Questions

Questions?
(While Demoing Continues)

Research Details

- Git Repo: https://github.com/Mauddib28/bleep-tool



Appendix Slides

Specifics to Know

- Odd Behavior
  - Double read the GATT, Initialized GATT cannot be altered
  - D-Bus willingly forgets
  - W to Arduino Descriptors = panic crash
  - Default buffer character limit

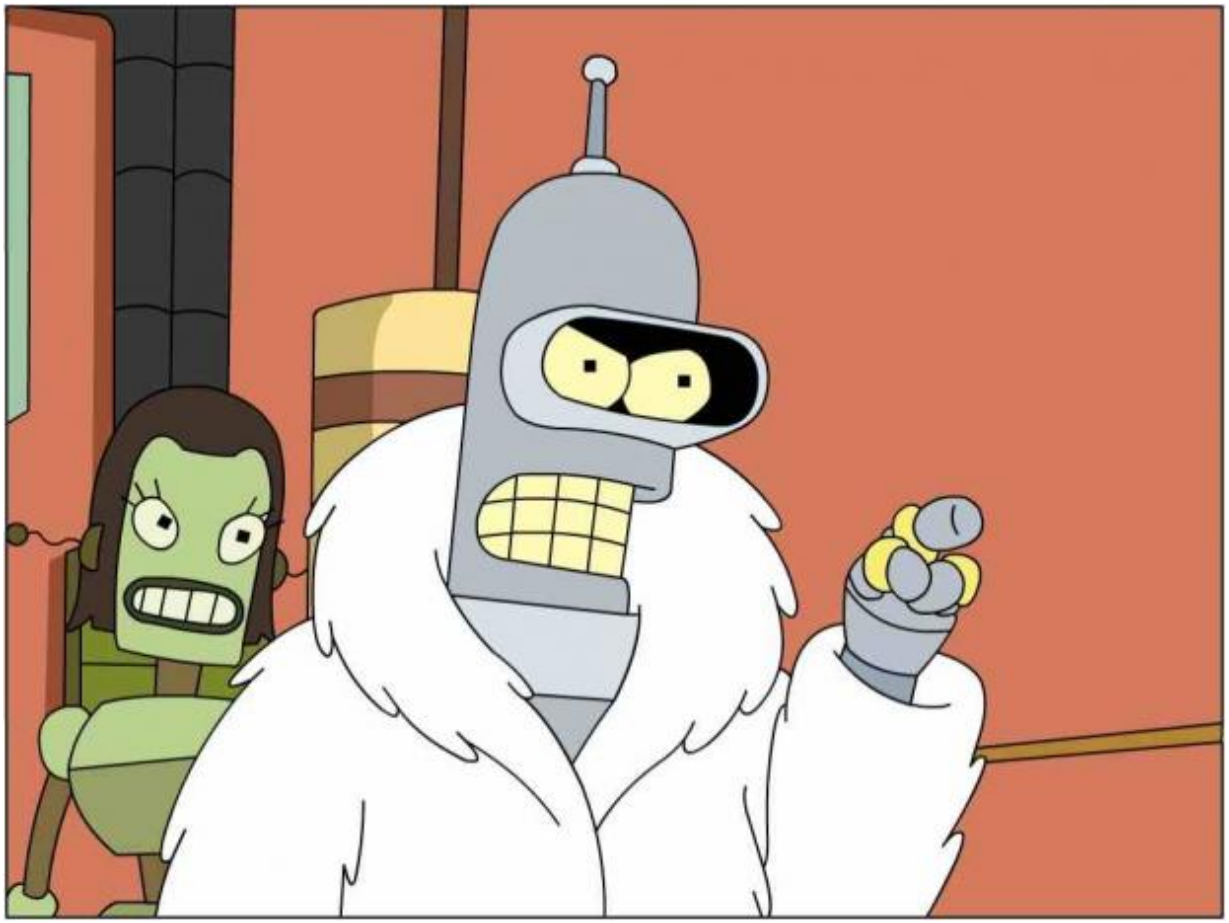
Proof-of-Concepts (PoC) Servers

Arduino

- Getting the Basics Down
- Simple structure with mild nesting
- Rudimentary R/W I/O
  - Descriptors... NO TOUCH!

BLE C[w]TF to Pico W - Signals Suck....

- Not seeing BLE CTF signals!?!?!!!
- Make a newer, better BLE server!



Pico W - It's Tricky!

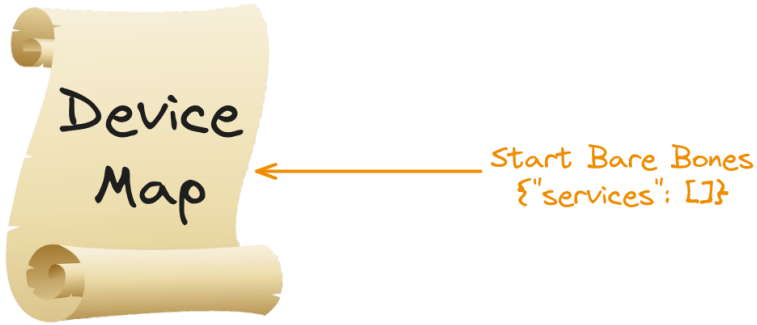
- To Read and Write, with tricks this time
- Can Gondor light the Signal beacons?
- What even is Order-of-Operations?!? State-machine?

Bluetooth State-Machine

- How does this sucker tick?
- Surprisingly basic; pass the butter...
  - Too many things = FIRE
- How much nesting is too much nesting?
- Trial and.... Tri.... Trial and Er.... Trail and Error! (Damn it...)

Basic Capabilities

Basic Capabilities - Device Mapping



1) Connect to the device
   - (i) Confirm ServicesResolved
   - (ii) Examine GAP information
   - (iii) Populate GAP info into the 'device_map'
     - Ex: Name, Addr, ManufacturerData, ServiceData Connected, Trusted, etc.

2) Recursively dive into each service to enumerate its details
   - Ex: Handle, UUID, Characteristics
   - (i) Recursively dive into each Characteristic associated w/ the current Service UUID
     - Ex: UUID, Handle, Flags, Value, Descriptors
     - (a) Dive into each Descriptor associated w/ the current Characteristic UUID
       - Ex: Flags, UUID, Value

[Continue recursive dive until all Service UUIDs enumerated]

3) During platform operation, BLEEP leverages the 'device_map' to track the Services/Characteristics/Descriptors & associated flags/values to each

Basic Capabilities - Device Mapping - Part I

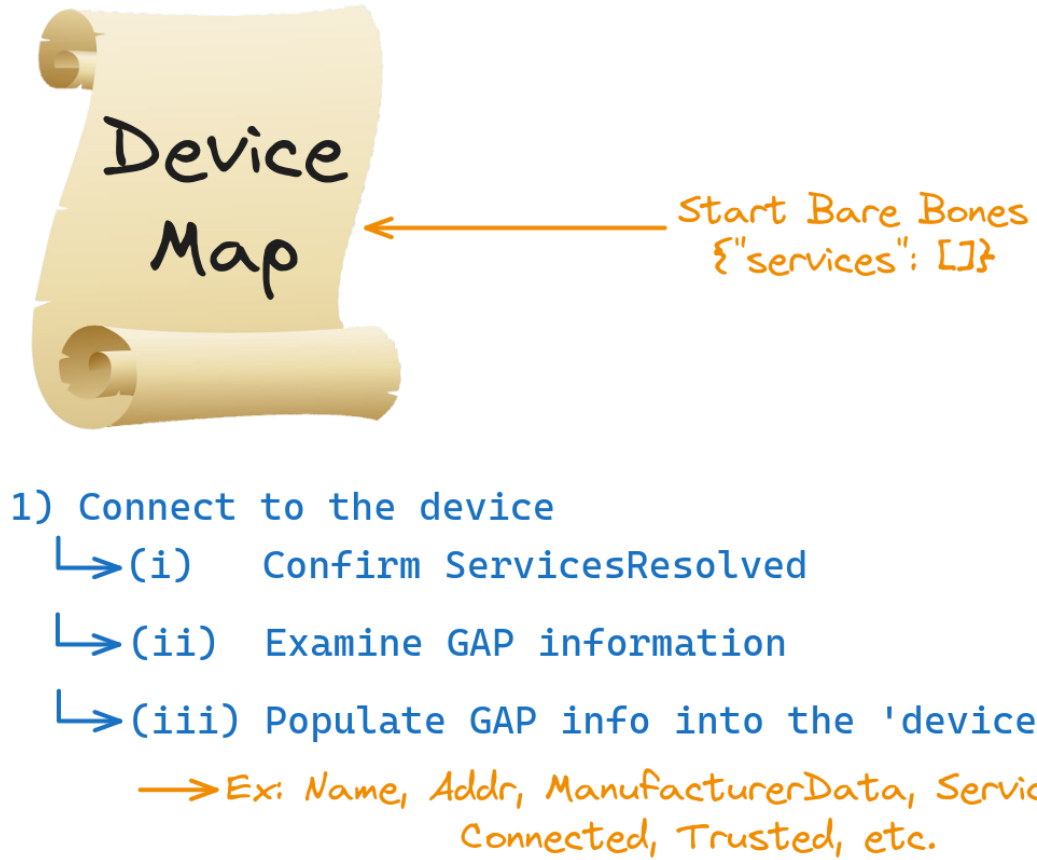


Figure 19: I/O - Device Mapping - Init + Step 1

Basic Capabilities - Device Mapping - Part II

2) Recursively dive into each service to enumerate its details
→ Ex: Handle, UUID, Characteristics
↳ (i) Recursively dive into each Characteristic associated w/ the current Service UUID
→ Ex: UUID, Handle, Flags, Value, Descriptors
↳ (a) Dive into each Descriptor associated w/ the current Characteristic UUID
→ Ex: Flags, UUID, Value

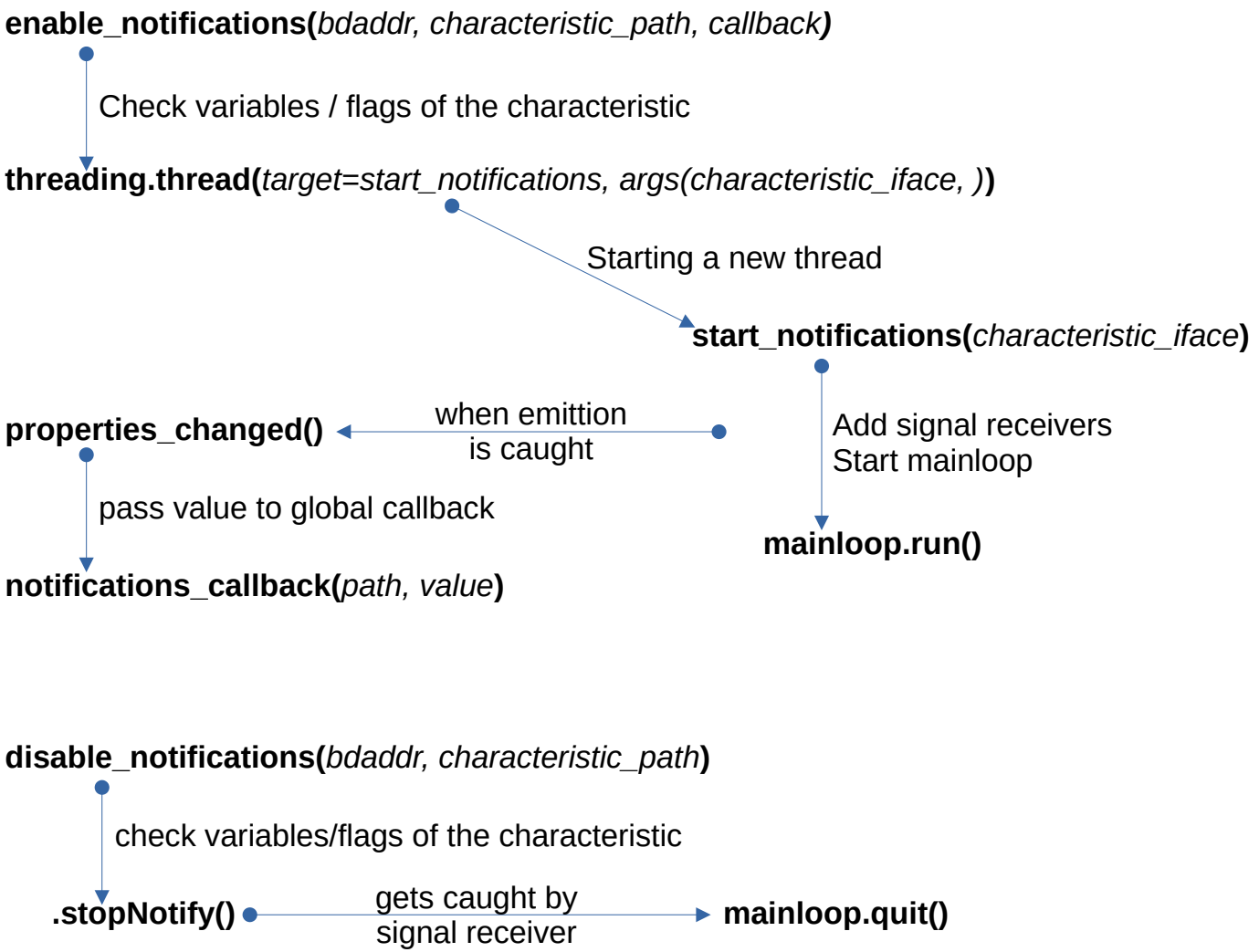
[Continue recursive dive until all Service UUIDs enumerated]

3) During platform operation, BLEEP leverages the 'device_map' to track the Services/Characteristics/Descriptors & associated flags/values to each

Figure 20: I/O - Device Mapping - Steps 2 + 3

Signals

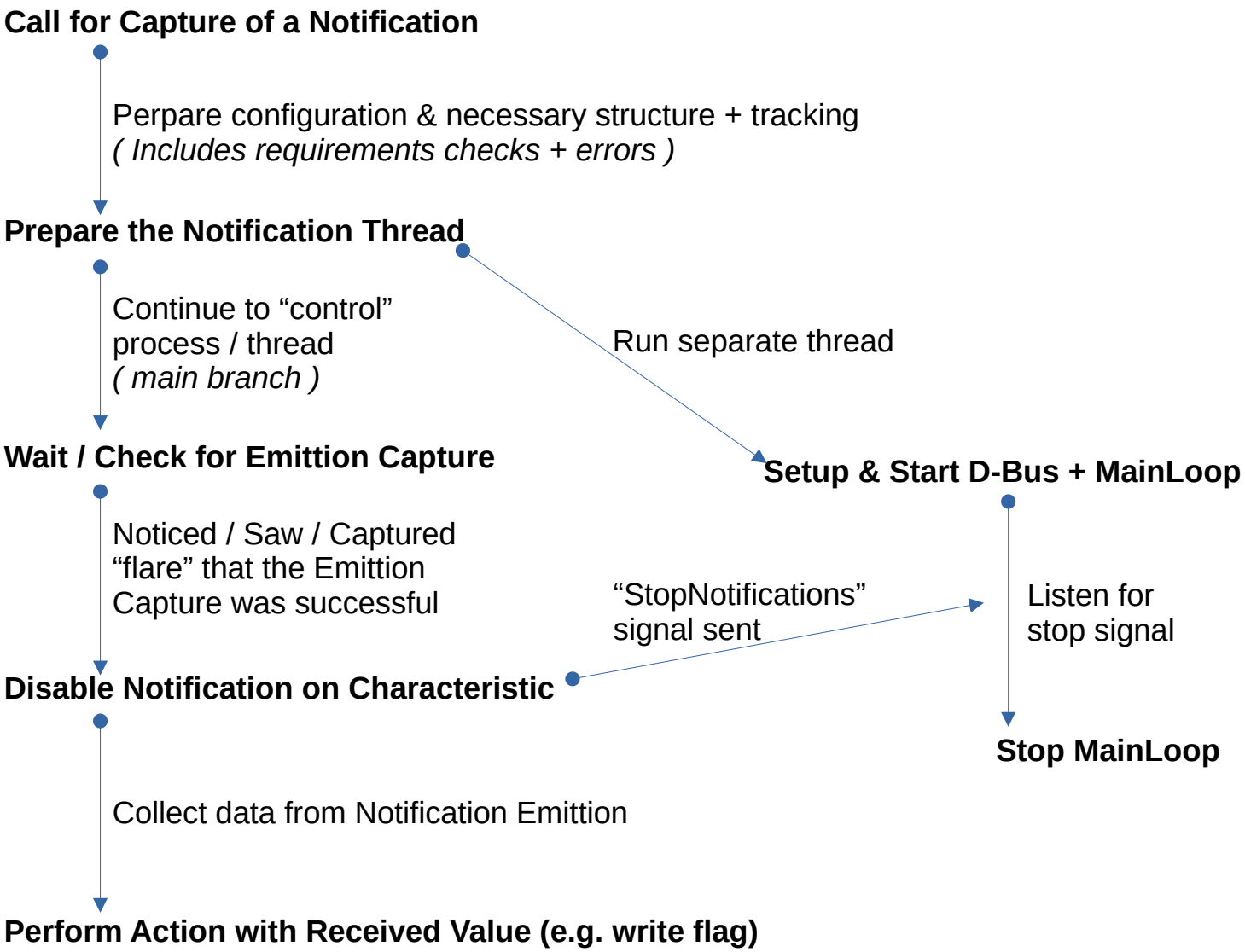
Emissions - Signals + Notifications - BT SIG Example Flow



Emissions - Signals + Notifications - Pseudo Code

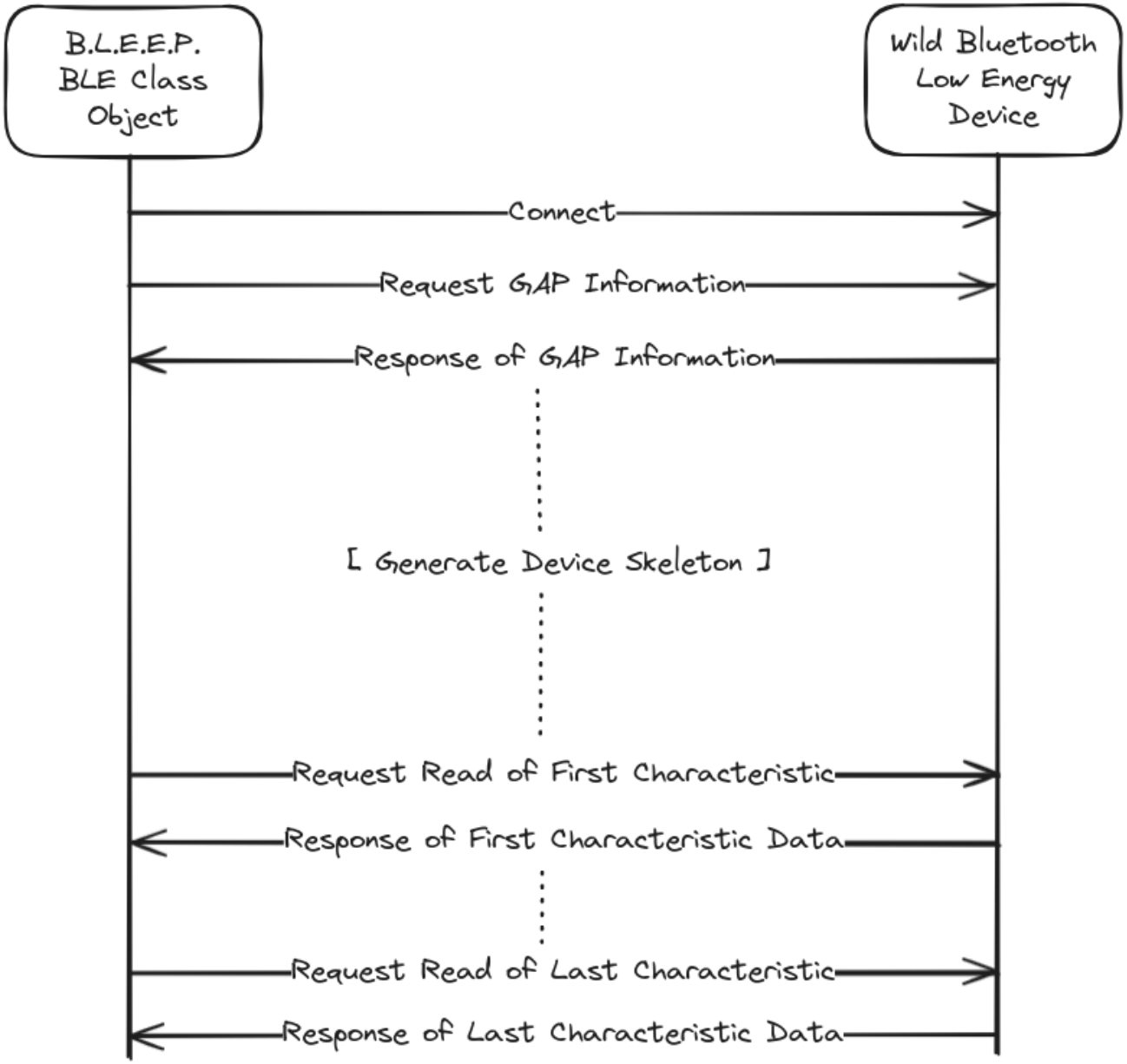
- Functionality Requirements:
  - Dissection of “PropertiesChanged” Emittion
  - Function to stop the GLib MainLoop
  - Function to configure D-Bus & GLib MainLoop
  - Function to validate the passed characteristic
  - Function to configure & run the thread
  - Function to act on Emission’s data

Emissions - Signals + Notifications - Developed Flow



Mapping Landmines/Canaries and Security

Old Method of Enumeration



Enumeration and Mapping

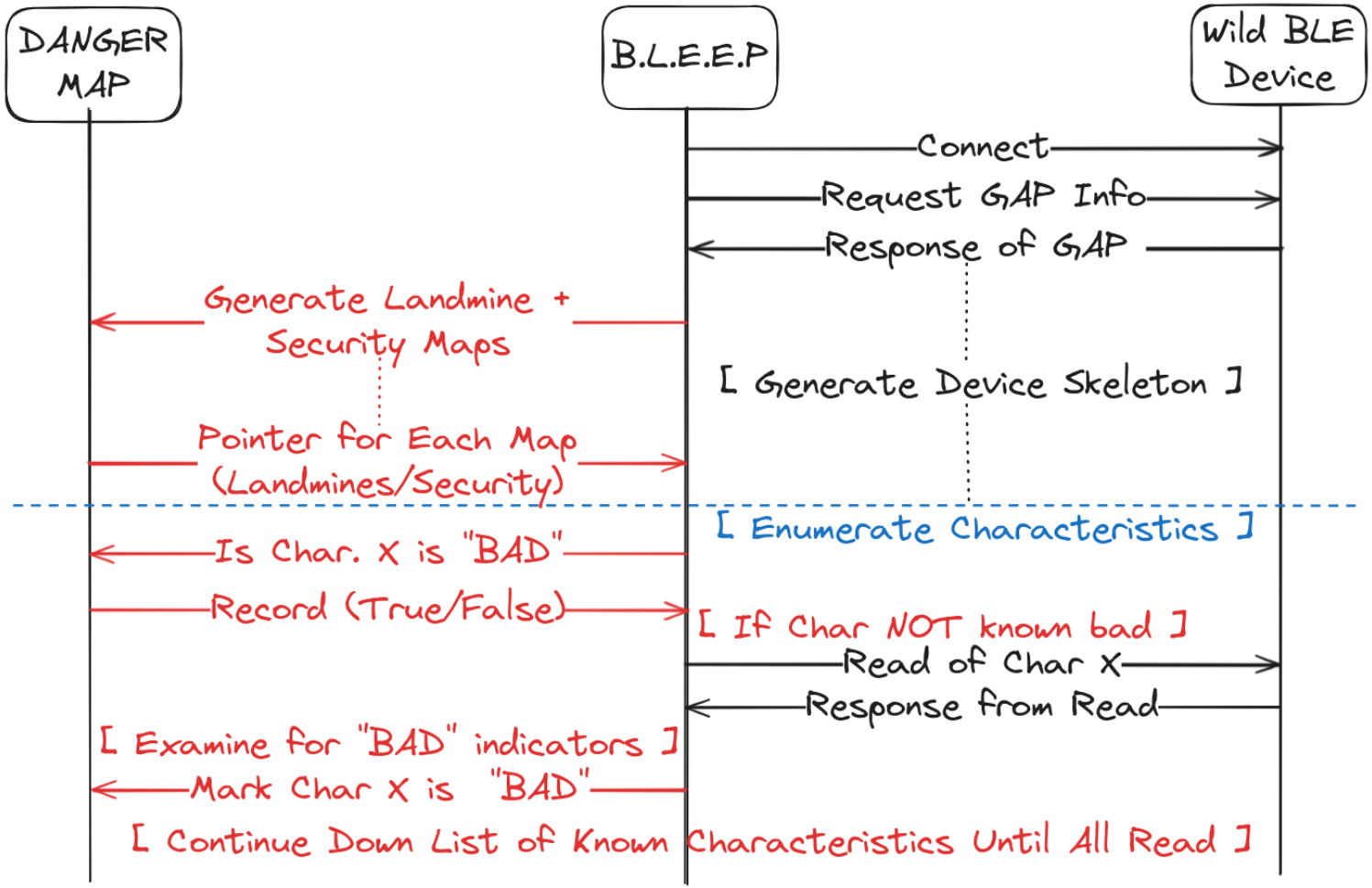


Figure 24: I/O - Mapped Enumeration

Enumeration and Mapping - Part I

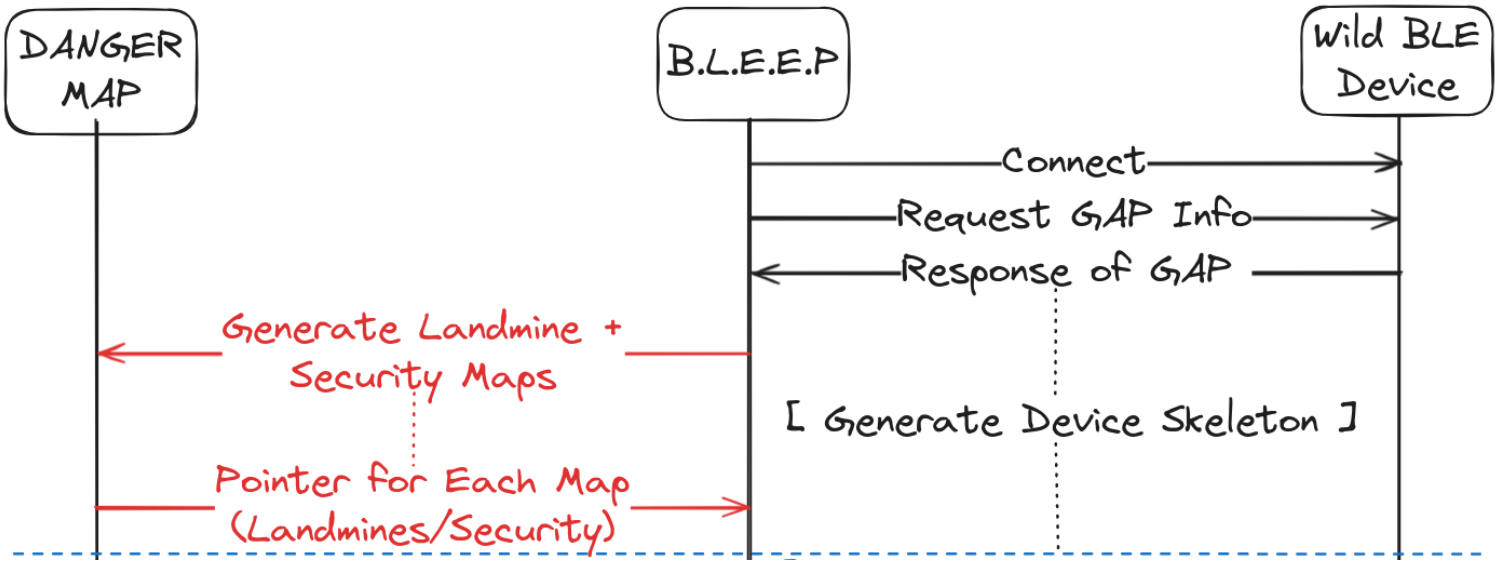


Figure 25: I/O - Mapped Enumeration - First Half

Enumeration and Mapping - Part II

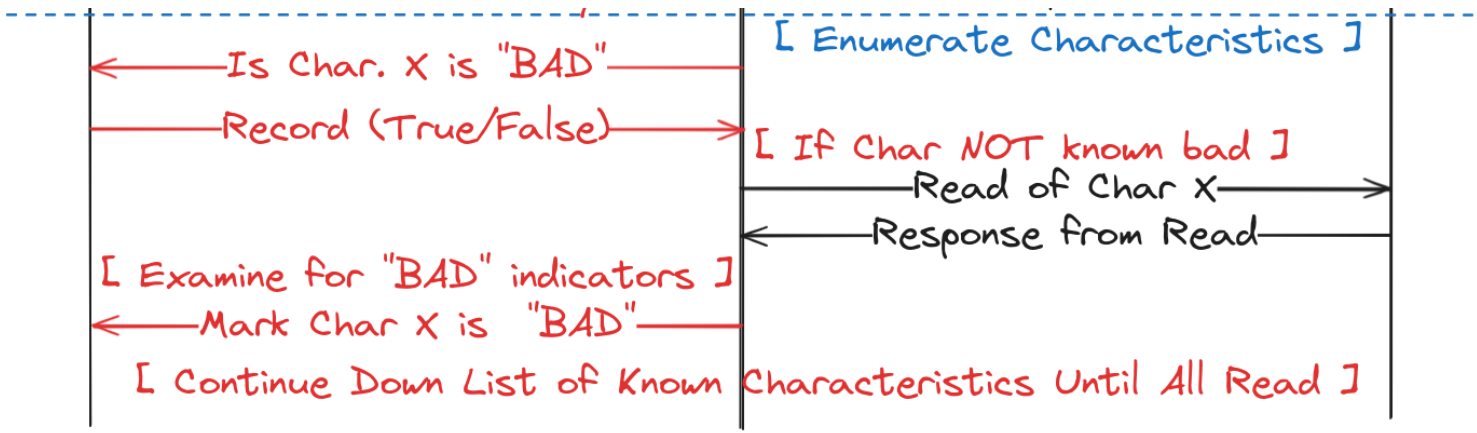


Figure 26: I/O - Mapped Enumeration - Second Half

Bibliography References I

Freedesktop

What is D-Bus, Published 2022

https://www.freedesktop.org/wiki/Software/dbus/

Last Accessed: 2024-03-28 22:43:19 EST

GNU

Knowing the Details of D-Bus Services

https://www.gnu.org/software/emacs/manual/html_node/dbus/Introspection.html

Last Accessed: 2024-04-01 19:48:34 EST

Programiz

Python Decorators

https://www.programiz.com/python-programming/decorator

Last Accessed: 2024-04-01 19:52:34 EST

Bibliography References II

hbldh

characteristic.py

https://github.com/hbldh/bleak/blob/63adefa24cb6ed11c8cf154fa41f51ecff1df98c/bleak/backends/characteristic.py

Last Accessed: 2024-04-01 19:56:34 EST

elsampsa

Bluetooth + DBus + gobject demo

https://github.com/elsampsa/btdemo

Last Accessed: 2024-04-01 19:54:52 EST

Freedesktop

DbusTools, Published May 07 2021

https://www.freedesktop.org/wiki/Software/DbusTools/

Last Accessed: 2024-03-28 22:57:29 EST

Bibliography References III


Bluetooth SIG
assigned_numbers
https://bitbucket.org/bluetooth-SIG/public/src/main/assigned_numbers/
Last Accessed: 2024-04-01 21:00:29 EST


Bluetooth SIG
public bitbucket
https://bitbucket.org/bluetooth-SIG/public/src/main/
Last Accessed: 2024-04-02 15:13:33 EST


Archlinux Forum
Activation via systemd failed for unit dbus-org.bluez.service
https://bbs.archlinux.org/viewtopic.php?id=155714
Last Accessed: 2024-04-02 15:09:42 EST

Bibliography References IV

oscaracena

gattlib.h

https://github.com/oscaracena/pygattlib/blob/7d08c0805313201b2ab12628e19544bb180218a8/src/gattlib.h

Last Accessed: 2024-04-02 15:09:42 EST

Bluetooth SIG

Bluetooth for Linux Developers Study Guide - Versions 1.0, 1.0.1

https://www.bluetooth.com/bluetooth-resources/bluetooth-for-linux/

Last Accessed: 2021-12-29 10:26:27 EST, 2022-10-18 18:54:02 EST

Bibliography References V

Bluetooth SIG

Developer Study Guide Bluetooth Internet Gateways - Version 2.0.0

https://www.bluetooth.com/blog/the-bluetooth-internet-gateway-study-guide/

Last Accessed: 2021-07-21 14:46:56 EST

Bluetooth SIG

Bluetooth LE Developer Study Guide - Version 5.2.0

https://www.bluetooth.com/bluetooth-resources/bluetooth-le-developer-starter-kit/

Last Accessed: 2023-02-16 11:36:57 EST

Bibliography References VI

Bluetooth SIG

Bluetooth Core Specification - Versions 5.3, 5.4

https://www.bluetooth.com/specifications/specs/core-specification-5-[3—4]/

Last Accessed: 2023-12-19 13:24:22 EST, 2023-12-16 11:33:37 EST

Bluetooth SIG

Generic Attribute Profile (GATT)

https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Core-54/out/en/host/generic-attribute-profile–gatt-.html

Last Accessed: 2023-12-16 11:22:03 EST

Marcel Holtmann, Maxim Krasnyansky, Qualcomm

BlueZ - Bluetooth protocol stack for Linux

https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/doc

Last Accessed: 2024-04-11 10:39:23 EST