

Bluetooth For Pythonistas

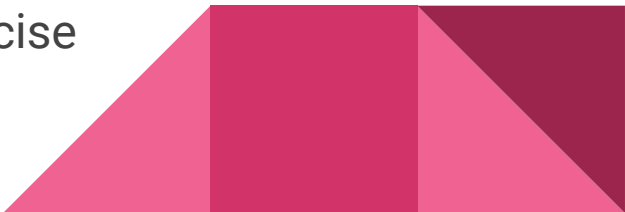
Barry Byford

@uk_baz

<https://ukbaz.github.io/>

Tonight's exercise...

- Came about when trying to solve a problem on my project
- Project was:
 - Accessing Bluetooth on Linux with Python
 - Bluetooth stack on Linux uses BlueZ
 - BlueZ has an API based on DBus
- You don't need to know any of that for tonight's exercise
 - However it does give context

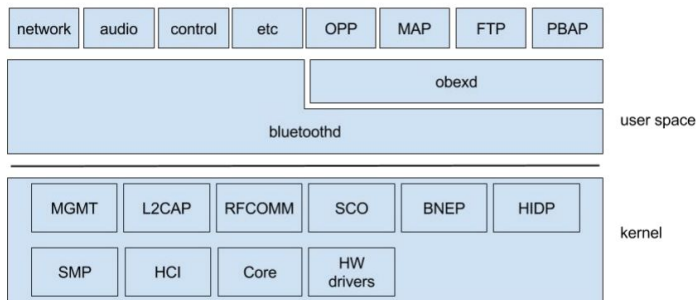


Why Bluetooth?

It's there on your phone

And on your maker boards

Complexity Challenge: Embedded vs Application



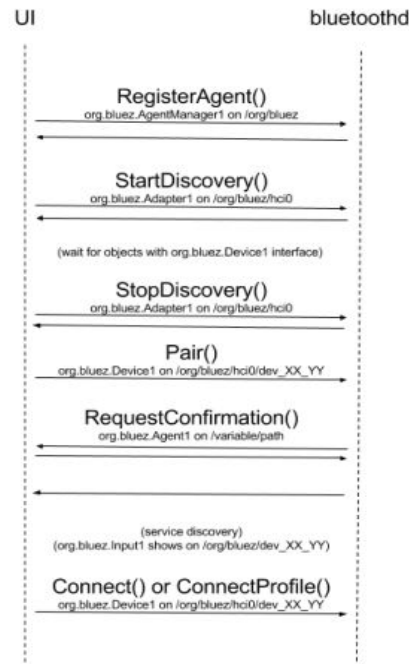
| | |
|------------------------|--------------------------------------|
| IO Pin Service | E95D127B-251D-470A-A062-FA1922DFA9A8 |
| IO Pin Characteristics | E95Dxxxx-251D-470A-A062-FA1922DFA9A8 |

| Characteristic Name | UUID | Write value |
|---------------------|--------|-------------|
| pin IO config | [B9FE] | 0x00000000 |
| pin AD Config | [5899] | 0x00000000 |
| pin data | [8D00] | 0x0000 |

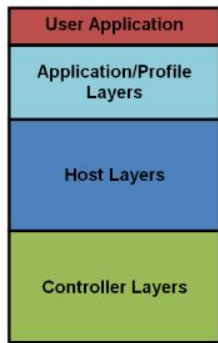
| Characteristic Name | UUID | Write value |
|---------------------|--------|-------------|
| pin IO config | [B9FE] | 0x00000000 |
| pin AD Config | [5899] | 0x00000000 |
| pin data | [8D00] | 0x0001 |

| Characteristic Name | UUID | Write value |
|---------------------|--------|------------------|
| pin PWM | [D822] | 0x000001E1080000 |

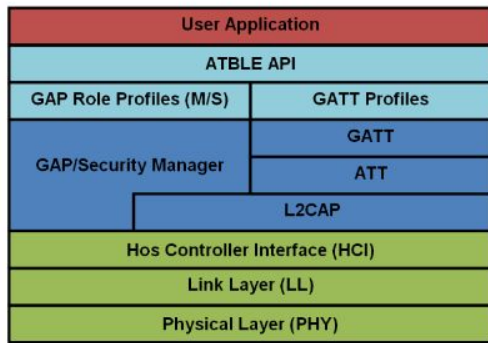
| Characteristic Name | UUID | Write value |
|---------------------|--------|------------------|
| pin PWM | [D822] | 0x000002E1080000 |



Simplified Stack



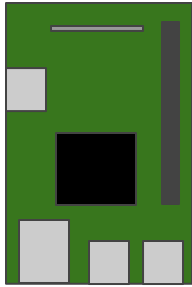
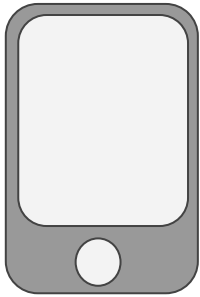
Detailed Stack



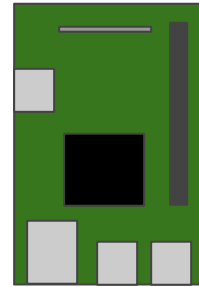
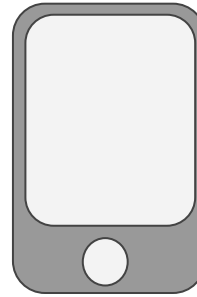
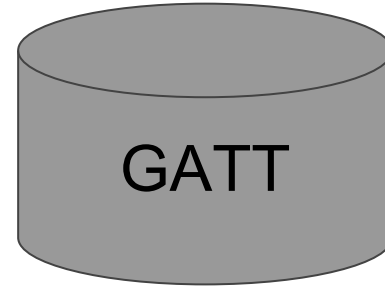
Source: blueradios.com

Two Types of Bluetooth

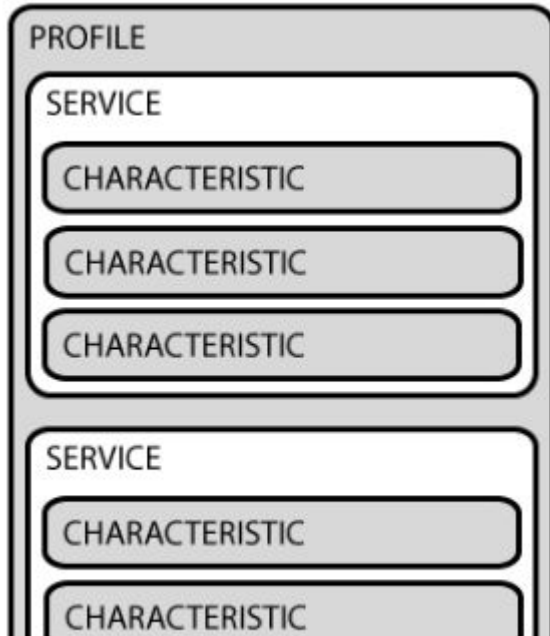
Classic (Serial Port Profile)



Bluetooth Low Energy (BLE)



The Generic ATtribute (GATT) And DBus



```
-> /org/bluez/hci0 {org.bluez.Adapter1: Address}
|
|-> dev_xx_xx_xx_xx_xx_xx org.bluez.Device1: Address}
| |
|   -> servicexxxx {org.bluez.GattService1: UUID}
|   |
|       -> charxxxx {org.bluez.GattCharacteristic1: UUID}
|       |
|           -> descxxxx {org.bluez.GattDescriptor1: UUID}
|       |
|       -> charxxxx {org.bluez.GattCharacteristic1: UUID}
|
|-> dev_xx_xx_xx_xx_xx_xx org.bluez.Device1: Address}

-> /org/bluez/hci1 {org.bluez.Adapter1: Address}
```

The Generic Attribute Profile (GATT) procedures define standard ways that services, characteristics and their descriptors can be discovered and then used to allow Bluetooth Low Energy devices to transfer data. The profile describes a use case, roles and general behaviors based on GATT functionality

Micro:bit BLE GATT Profile UUID's

| Description | Service | Characteristic |
|-----------------------------|---------|----------------|
| Button Service | 9882 | |
| Button A Value | | DA90 |
| Button B Value | | DA91 |
| Accelerometer Service | 0753 | |
| Accelerometer Data | | CA4B |
| Accelerometer Notify Period | | FB24 |

Base UUID = E95D**XXXX**-251D-470A-A062-FA1922DFA9A8

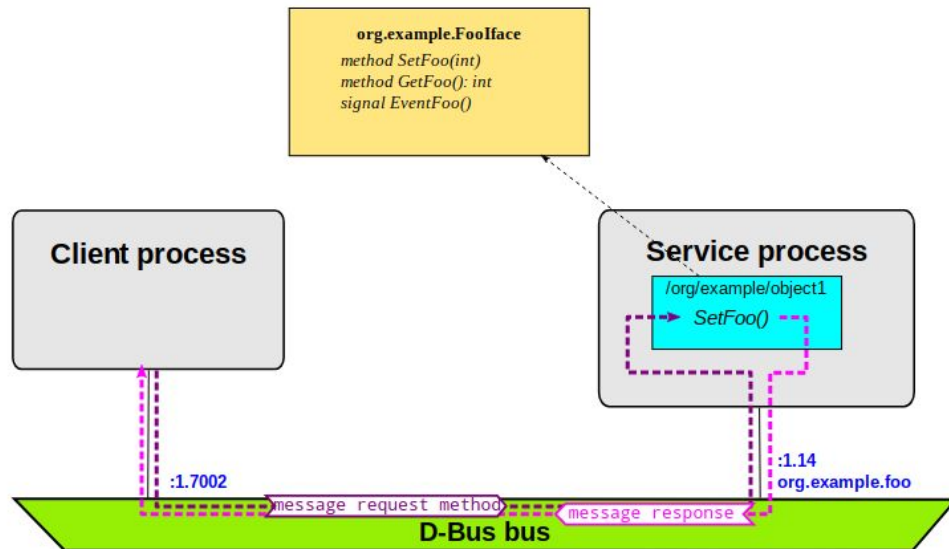
Using DBus Software Bus

- Objects

- /org/bluez/<adapter>/<device>/<service>/<char>/<desc>
- /org/bluez/hci0/dev_F7_17_E4_09_C0_C6/service000c/char000d/desc0003

- Interfaces

- org.bluez.Adapter1
- org.bluez.Device1
- org.bluez.GattService1
- org.bluez.GattCharacteristic1
- org.bluez.GattDescriptor1



Accessing the BlueZ DBus API

```
>>> import pydbus
>>> dbus = pydbus.SystemBus()
>>> ble = dbus.get('org.bluez', '/org/bluez/hci0')
>>> dir(ble)
['Address', 'Alias', 'Connected', 'Disconnect',
'Discoverable', 'DiscoverableTimeout', 'Discovering',
'Name', 'Pairable', 'PairableTimeout', 'Powered',
'PropertiesChanged', 'SetDiscoveryFilter', 'StartDiscovery',
'StopDiscovery', 'UUIDs', '__Introspect', '__class__',
'__delattr__', '__dict__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattr__', '__getitem__',
'__gt__', '__hash__', '__init__', '__le__', '__lt__', ...]
```

Using the Object

```
>>> ble.Name  
'Linaro-alip'
```

```
>>> ble.Powered  
True
```

```
>>> ble.Address  
'00:00:00:00:5A:AD'
```





Tonight's Exercise

Overview

- Write a function called `get_dbus_path()`
- Inputs are Bluetooth address and UUIDs for required path
- Returns DBus Object path
 - Gets DBus Object path from `GetManagedObjects()`



GetManagedObjects Structure

Template

```
{  
  
  <object>:  
  
    {  
  
      <interface>:  
  
        {  
  
          <key> : <value>  
  
        }  
  
      }  
  
    }  
  
}
```

Actual

```
{  
  '/org/bluez/hci0/dev_F7_17_E4_09_C0_C6/service000c/char000d':  
    {  
      'org.freedesktop.DBus.Properties': {},  
      'org.bluez.GattCharacteristic1': {  
        'Value': [],  
        'Service': '/org/bluez/hci0/dev_F7_17_E4_09_C0_C6/service000c',  
        'Flags': ['read'],  
        'UUID': '00002a24-0000-1000-8000-00805f9b34fb'},  
      'org.freedesktop.DBus.Introspectable': {}  
    }  
}
```

Example Call and Return

```
>>> import bluezero
```

```
>>> bluezero.get_dbus_path('00:00:00:00:5A:AD')
```

```
 '/org/bluez/hci0'
```

```
>>> bluezero.get_dbus_path('00:00:00:00:5A:AD',
```

```
... 'FD:6B:11:CD:4A:9B',
```

```
... 'e95d127b-251d-470a-a062-fa1922dfa9a8',
```

```
... 'e95d5899-251d-470a-a062-fa1922dfa9a8')
```

```
 '/org/bluez/hci0/dev_FD_6B_11_CD_4A_9B/service0020/char0021'
```



Running Tests

Single test:

```
$ python3 -m unittest test_get_dbus_path.GetPathTest.test_adapter
```

All tests:

```
$ python3 -m unittest test_get_dbus_path.GetPathTest
```

Or

```
$ python3 test_get_dbus_path.py
```





https://github.com/campug/bzero_kata.git

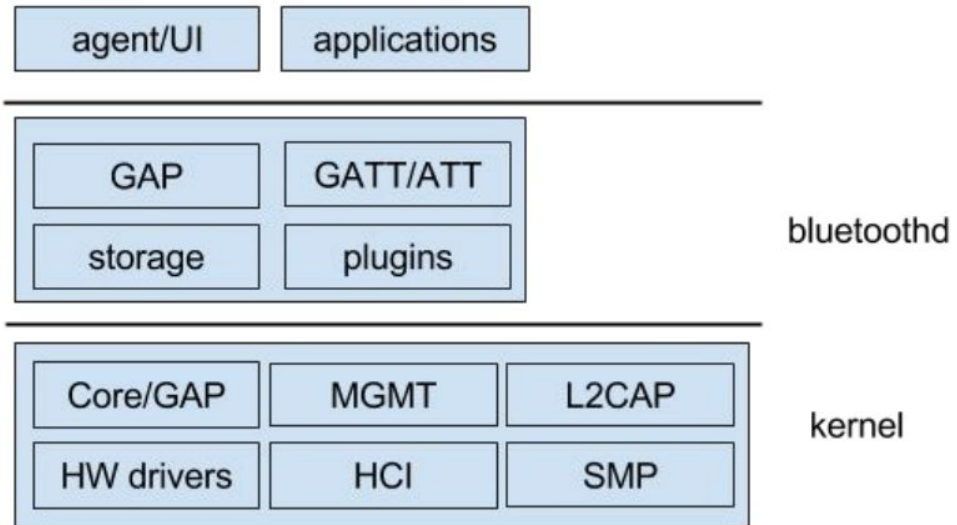
References

References:

| | |
|----------------|--|
| Bitty Software | <u>http://www.bittysoftware.com/</u> |
| App Inventor | <u>http://appinventor.mit.edu/explore/</u> |
| My blog | <u>https://ukbaz.github.io/</u> |
| PXT | <u>https://pxt.microbit.org/</u> |
| BlueZ API Docs | <u>https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/doc</u> |
| BlueZ Videos | <u>https://youtu.be/VMDyebKT5c4</u> <u>https://youtu.be/tclS9arLFzk</u> |



Bluetooth



GetManagedObjects Structure

Template

```
{  
  
  <object>:  
  
    {  
  
      <interface>:  
  
        {  
  
          <key> : <value>  
  
        }  
  
      }  
  
    }  
  
}
```

Actual

```
{  
  '/org/bluez/hci0/dev_F7_17_E4_09_C0_C6/service000c/char000d':  
    {  
      'org.freedesktop.DBus.Properties': {},  
      'org.bluez.GattCharacteristic1': {  
        'Value': [],  
        'Service': '/org/bluez/hci0/dev_F7_17_E4_09_C0_C6/service000c',  
        'Flags': ['read'],  
        'UUID': '00002a24-0000-1000-8000-00805f9b34fb'},  
      'org.freedesktop.DBus.Introspectable': {}}  
    }  
}
```