Contents

[I. Bluetooth overview 1](#_Toc130836358)

[1. Bluetooh Device roles 2](#_Toc130836359)

[1.1. Peripheral 2](#_Toc130836360)

[1.2 Central 2](#_Toc130836361)

[II. Android Bluetooth Architecture 3](#_Toc130836362)

[1. Bluetooth stack 4](#_Toc130836363)

[III. HCI Layer 5](#_Toc130836364)

[1. Initialize 5](#_Toc130836365)

[2. Transmit data 6](#_Toc130836366)

[IV. HIDL Layer 6](#_Toc130836367)

[1. Initialize 6](#_Toc130836368)

[2. Send data 7](#_Toc130836369)

[3. Callback functions 8](#_Toc130836370)

[V. Vendor implementation 8](#_Toc130836371)

[1. Lib-bt 9](#_Toc130836372)

[1.1. Initialized 9](#_Toc130836373)

[1.2. Cleanup 9](#_Toc130836374)

[1.3. Create dynamic library entry point 10](#_Toc130836375)

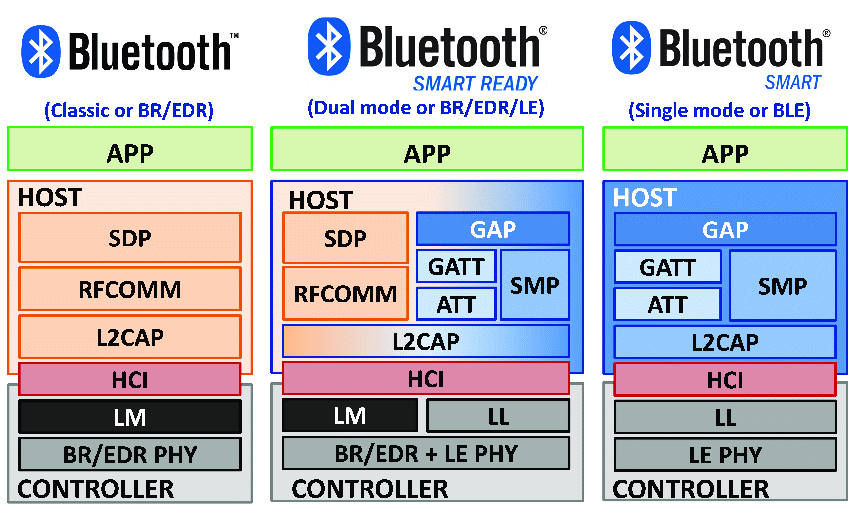
[1.4. Vendor specific operation 11](#_Toc130836376)

[2. Vendor\_interface.cc 12](#_Toc130836377)

[2. Send data functions 13](#_Toc130836378)

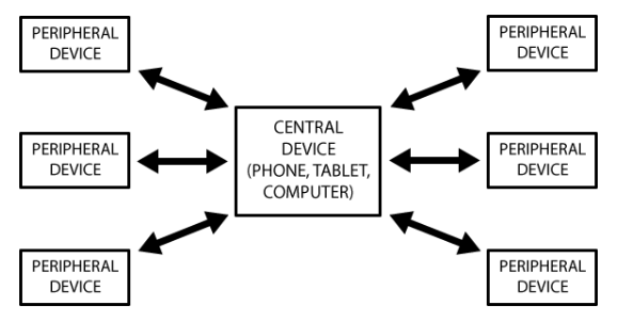
[VI. Appendx - Where is the bluetooth components directory 13](#_Toc130836379)

# I. Bluetooth overview



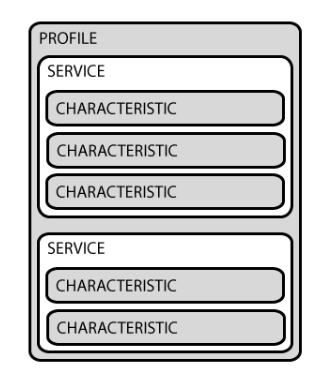
* GAP: defines procedures concerned with device discovery and establishing connections between 2 devices
* GATT: defines higher-level data types based on the attributes held in the attribute table
* ATT: contains a handle, a Universally Unique Identifier (UUID), a value and a set of permission.
* SMP: A protocol used during the execution of security procedures such as pairing
* L2CAP: responsible for protocol multiplexing, flow control and segmentation and reassembly of service data units (SDUs)
* HCI: defines a standardized interface via which a host can issue commands to the controller and a controller can communicate with the host

## 1. Bluetooh Device roles



### 1.1. Peripheral

Usually acts as a server



* Characteristic composed of
* A single value (int, float, string)
* Many discriptors
* Profile: Group of services
* Service: Group of related characteristics
* Identified using UUIDs

Ex: Smart Watch

* Profile: Heart rate
* Services: Heart rate
* Characteristics:

- Current heart rate (read/subcribe)

- Settings (read/write)

### 1.2 Central

Usually acts as a client

* Can connect to up to 7 BLE servers simultaneously
* Reads/Writes/Subscribes to characteristics

2. Bluetooth profiles

2.1 Advanced Audio Distribution Profile (A2DP)

This profile is used for streaming high-quality audio from one device to another, such as from a phone to a speaker or headphones

2.2. Hands-Free Profile (HFP)

This profile is used for making and receiving phone calls over Bluetooh. It is commonly used is car audio systems and bluetooth headset.

2.3. Human Interface Device Profile (HID)

This profile is used for connecting input devices such keyboards, mice and game controllers to other device over bluetooth.

2.4.Object Push Profile (OPP)

This profile is used for transferring files between devices over bluetooth. It is commonly used to share photos and other files between phones or between a phone and a computer.

2.5. Personal Area Network Profile (PAN)

This profile is used for connecting devices to form a personal area network (PAN) over bluetooth. It’s commonly used for sharing a internet connection between devices or for connecting devices in a home network.

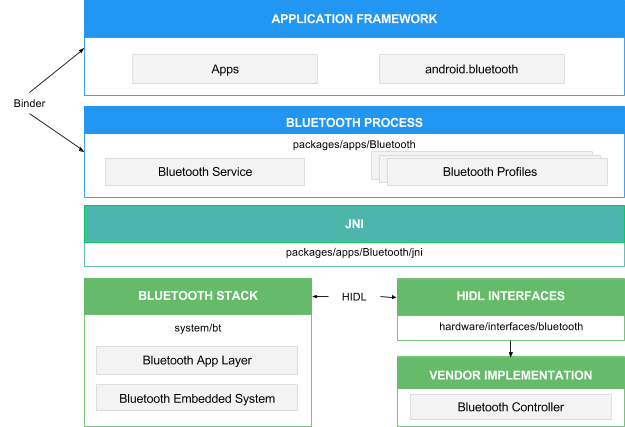
2.6. Serial Port Profile (SPP)

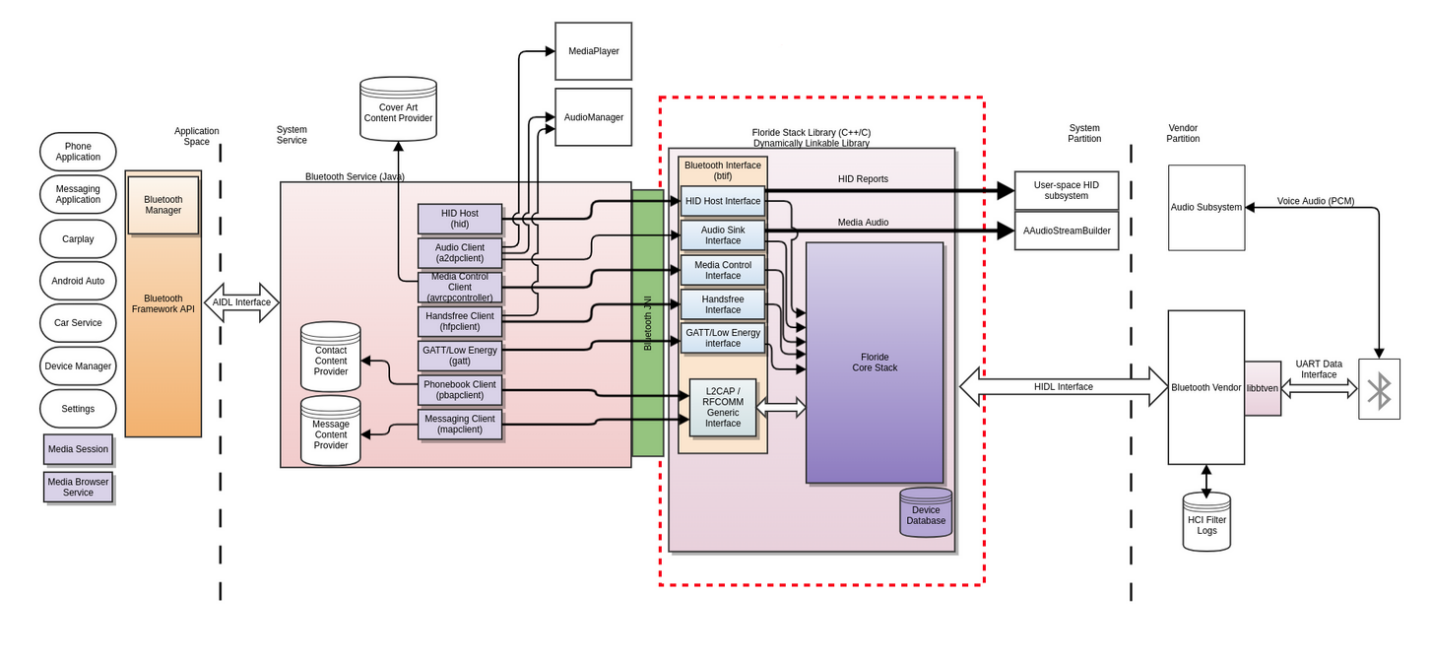
This profile is used for establishing a virtual serial port over bluetooth. Is’s commonly used in industrial and medical devices to communicate with a computer or other devices.

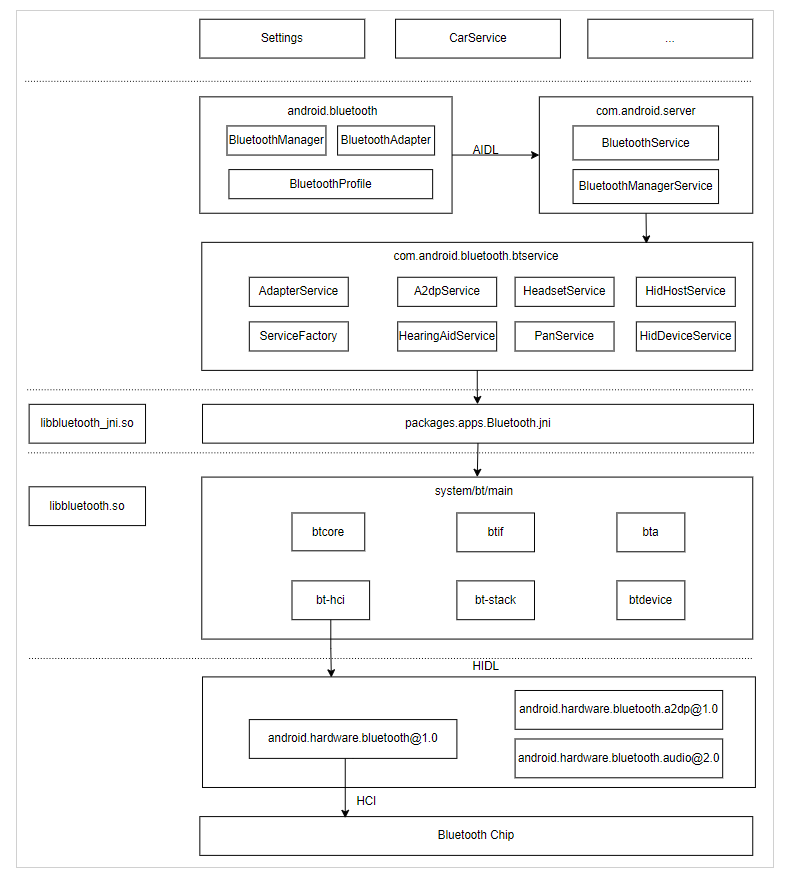
2.7. Generic Attribute Profile (GATT)

This profile is used in Bluetooth Low Energy (BLE) devices to define how data is exchanged between devices. It’s commonly used in fitness tracker, smartwatchs and other wearables.

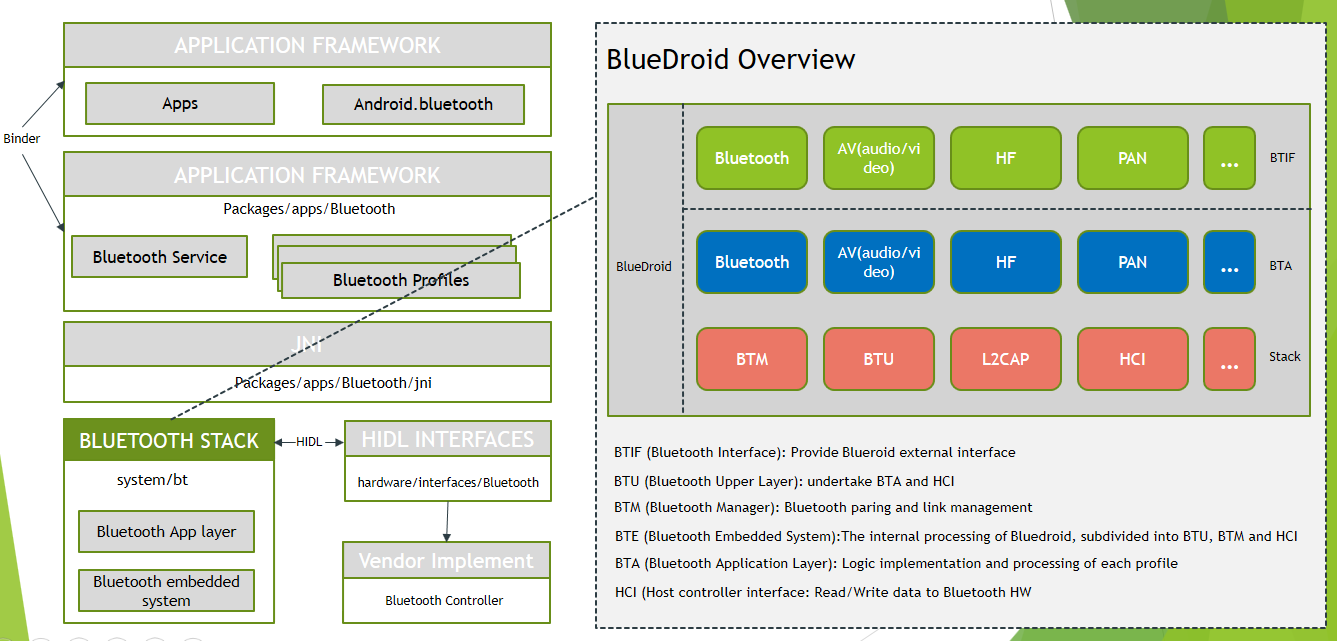
# II. Android Bluetooth Architecture



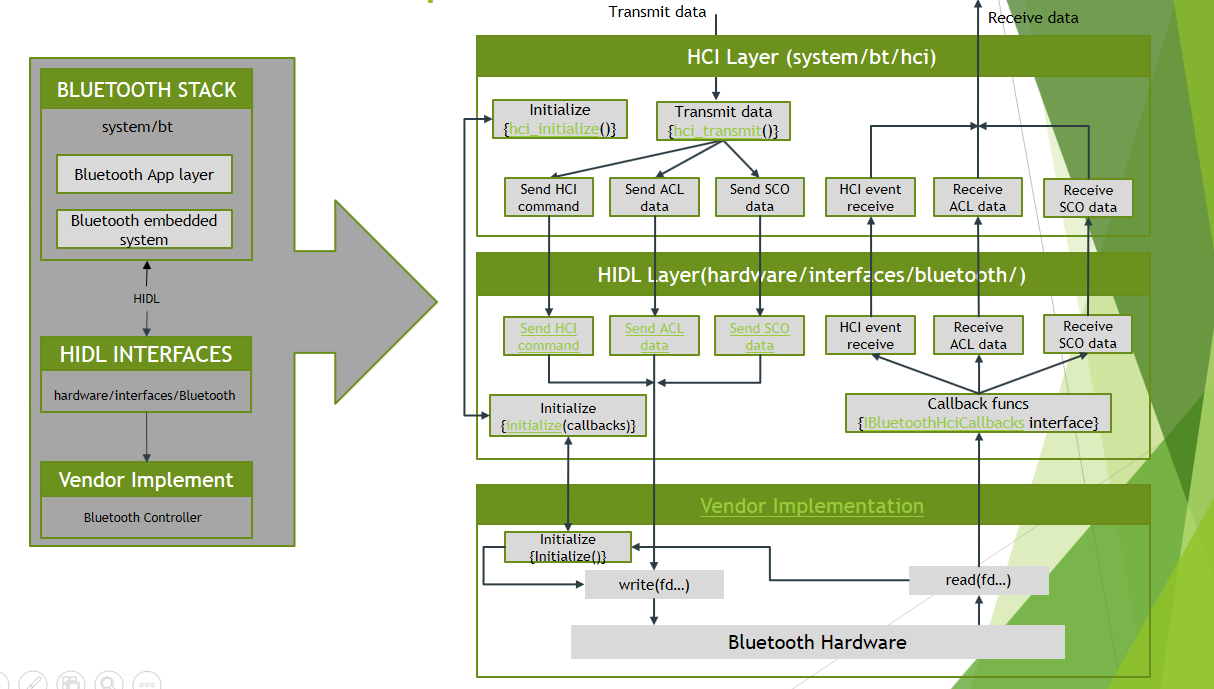




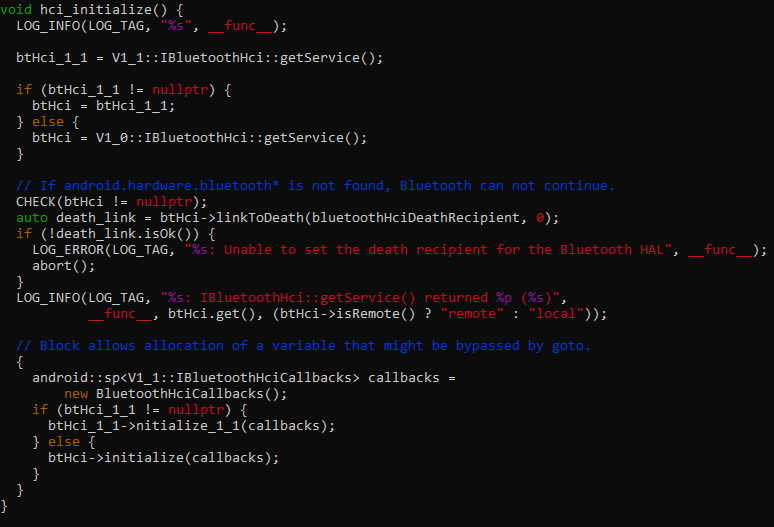
## 1. Bluetooth stack



# III. HCI Layer



## 1. Initialize



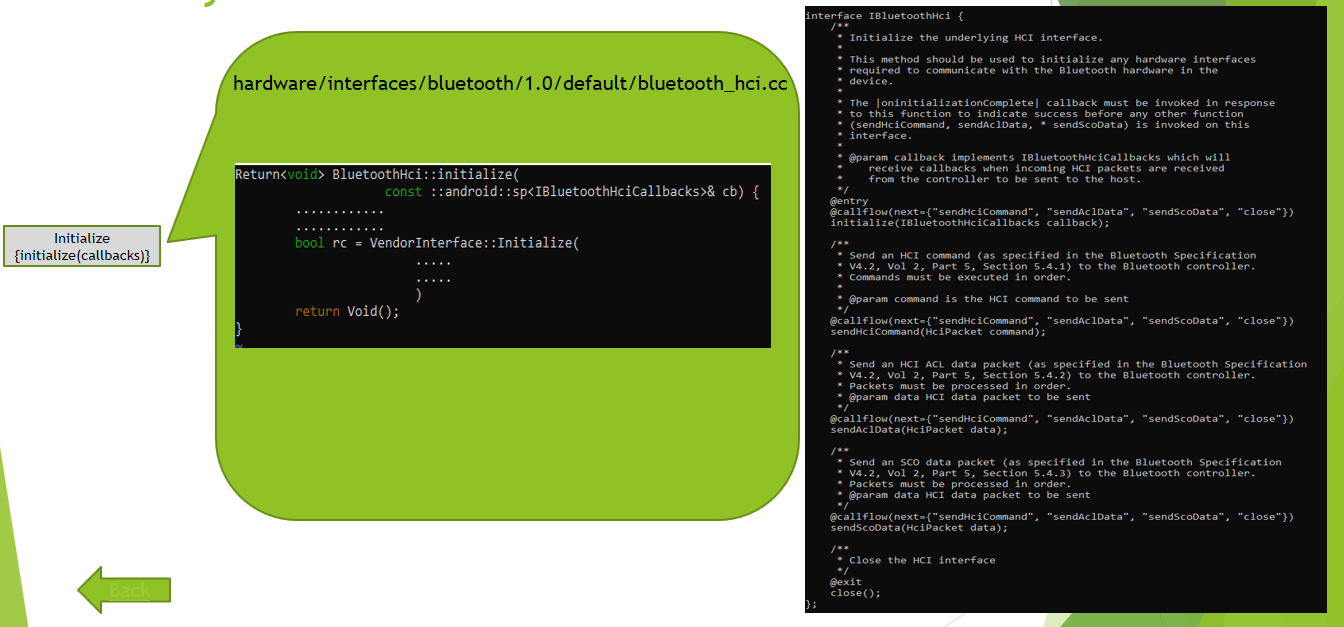
system/bt/hci/src/hci\_layer\_android.cc

## 2. Transmit data

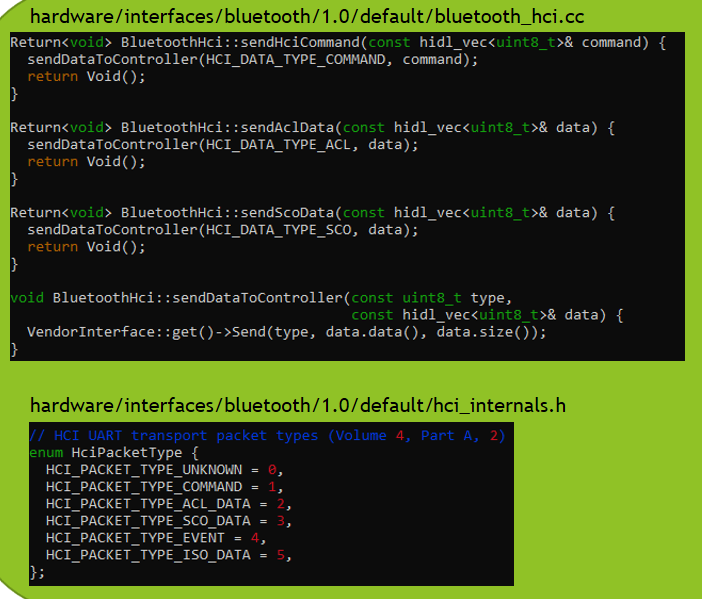


# IV. HIDL Layer

## 1. Initialize



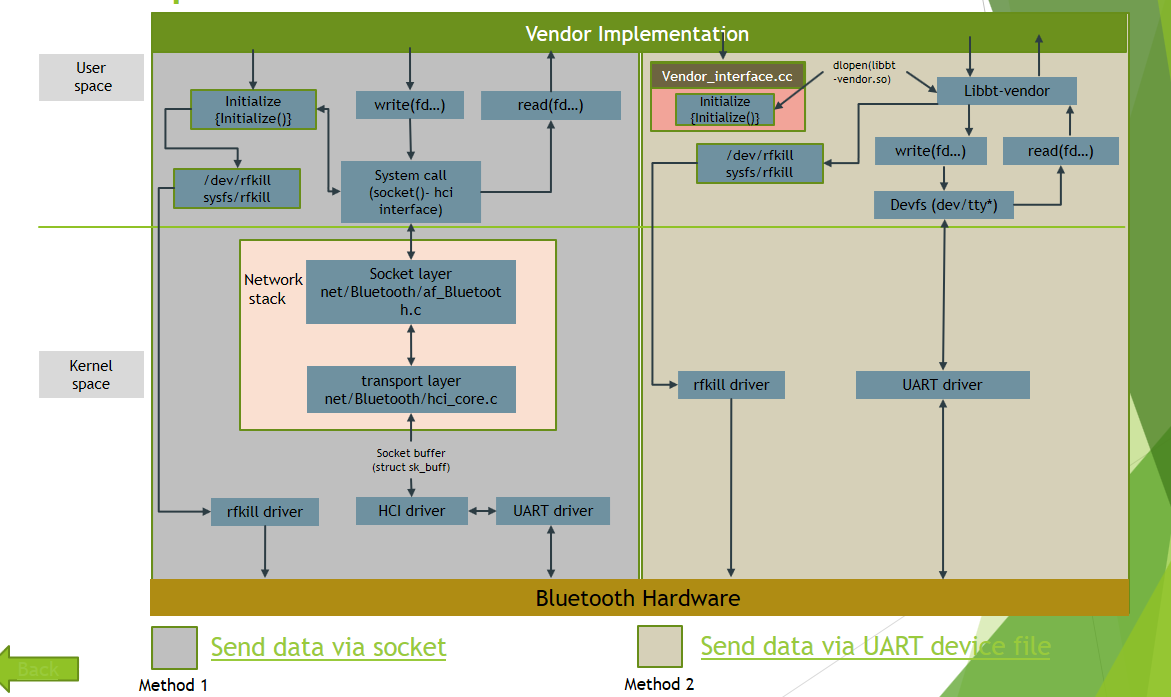
## 2. Send data



## 3. Callback functions



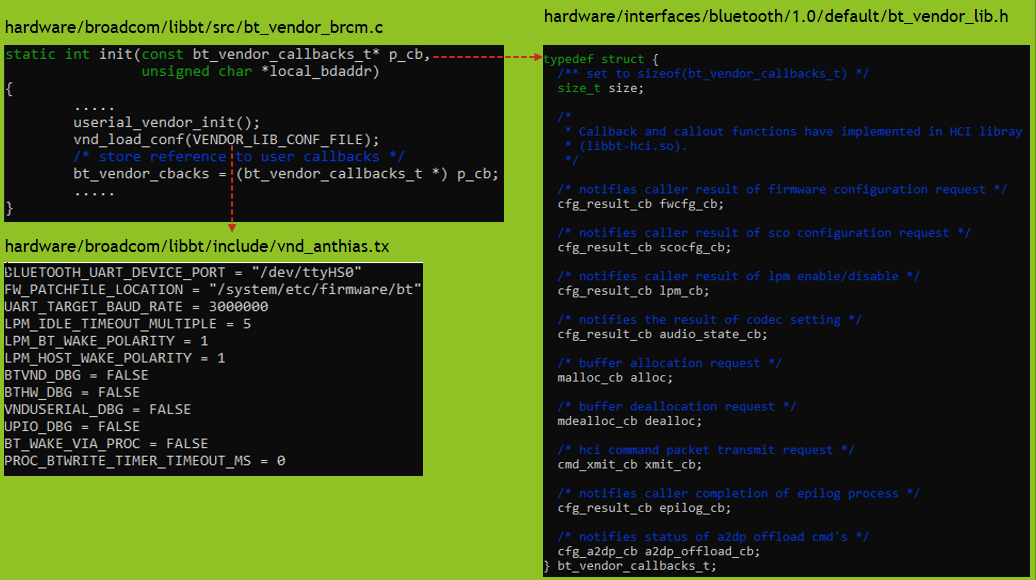
# V. Vendor implementation



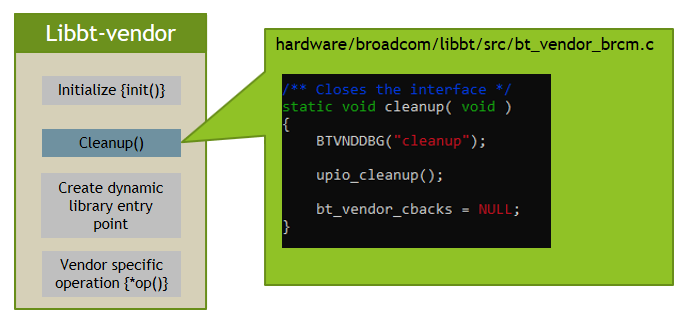
In this part, we discuss the ***lib-bt*** and ***Vendor\_interface.cc***. With the kernel space part, we have a UART document to refenrence.

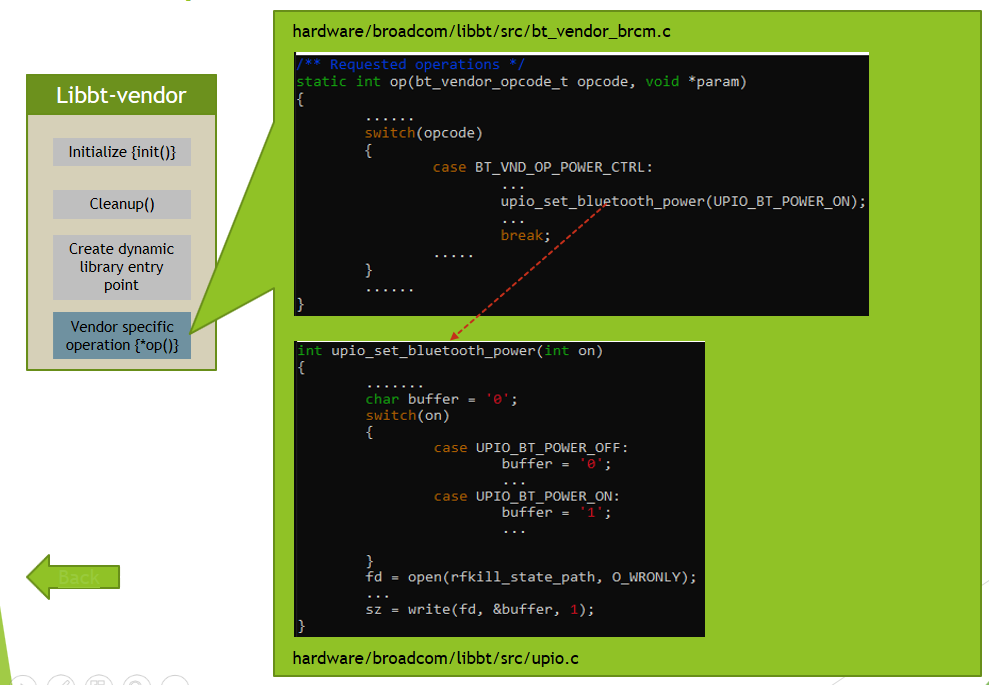
## 1. Lib-bt

### 1.1. Initialized

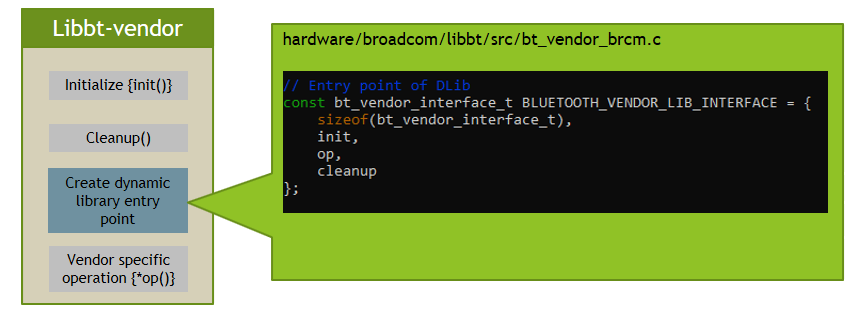


### 1.2. Cleanup

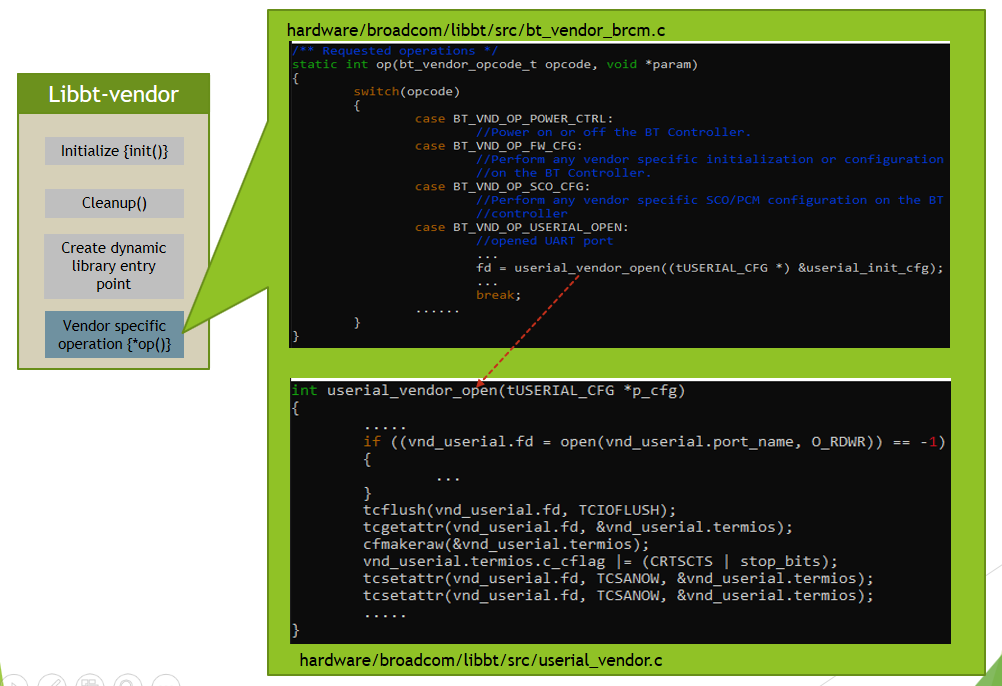




### 1.3. Create dynamic library entry point



### 1.4. Vendor specific operation



## 2. Vendor\_interface.cc

### 2.1 Initialize



### 2.2 Send data functions



# VI. Application

# VII. Appendx - Where is the bluetooth components directory

**A2SP software encoding**

**audio\_hearing\_aid\_hw**

**binder**

**device**

**conf**

**utils**

**bta**

**btcore**

**btif**

**common**

**device directory**

**embdrv**

**gd**

**The HCI**

**Include**

**internal\_include**

**Main**

**The OSI**

**Packet**

**Profile**

**Proto**

**The service**

**The stack**

1. a2dp

|  |  |
| --- | --- |
| File name | Function |
| a2dp\_acc.cc |  |
| a2dp\_aac\_decoder.cc |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Test (Bluetooth test function and script)**

**Types**

**udrv**

**utils: bt utils**

**vendor\_libs**

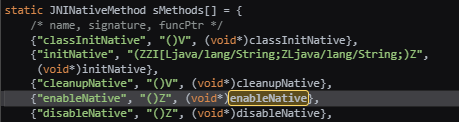
**vnd directory**

**VIII. Native layer**

The application part calls the enableNative function

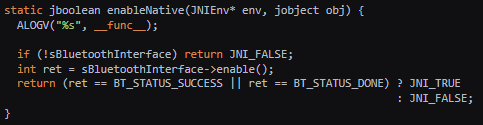
packages/modules/Bluetooth/android/app/src/com/android/bluetooth/btservice/AdapterState.java

Define enableNative method

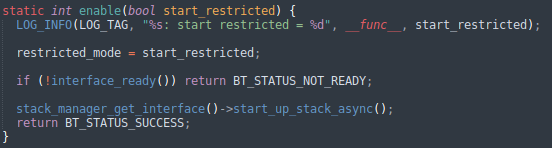


packages/modules/Bluetooth/android/app/jni/com\_android\_bluetooth\_btservice\_AdapterService.cpp

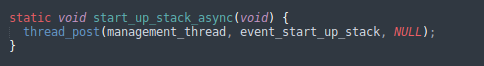
Implementation of JNI functions



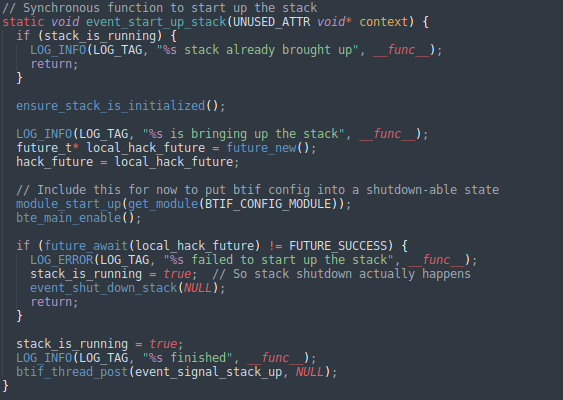
packages/modules/Bluetooth/android/app/jni/com\_android\_bluetooth\_btservice\_AdapterService.cpp



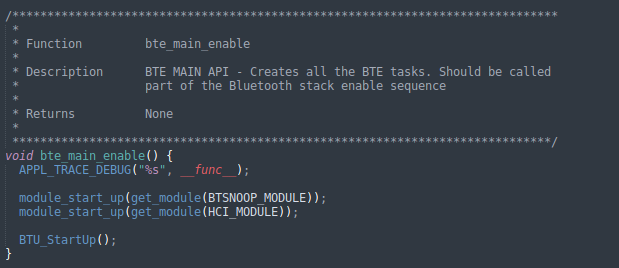
system/bt/btif/src/bluetooth.cc



system/bt/btif/src/stack\_manager.cc

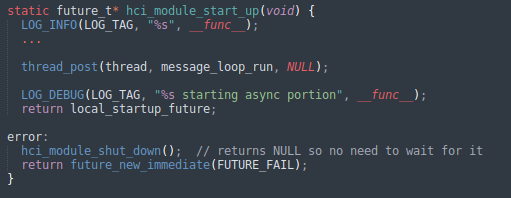


system/bt/btif/src/stack\_manager.cc



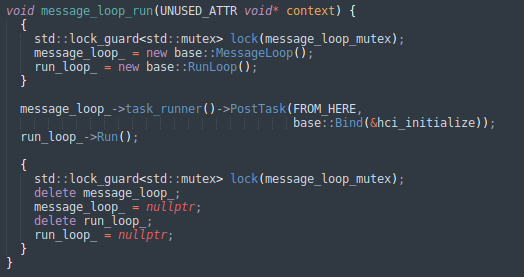
system/bt/main/bte\_main.cc

module\_start\_up(get\_module(HCI\_MODULE) will convert to call to hci\_layer.cc. In this file, hci\_module\_start\_up function will be called to init bluetooth hci.

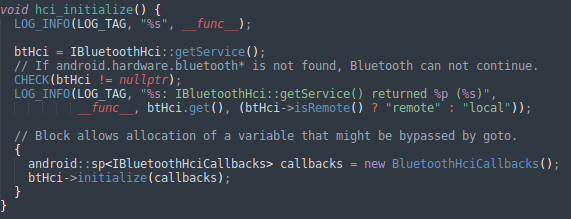


/system/bt/hci/src/hci\_layer.cc

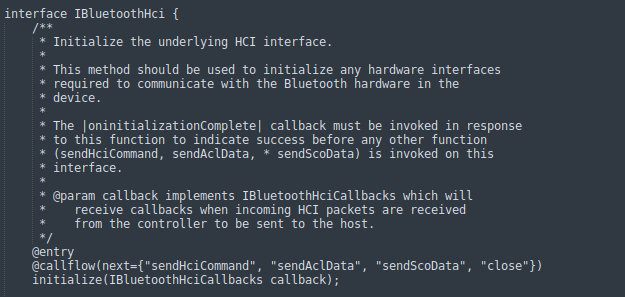
message\_loop\_run will call to hci\_initialize



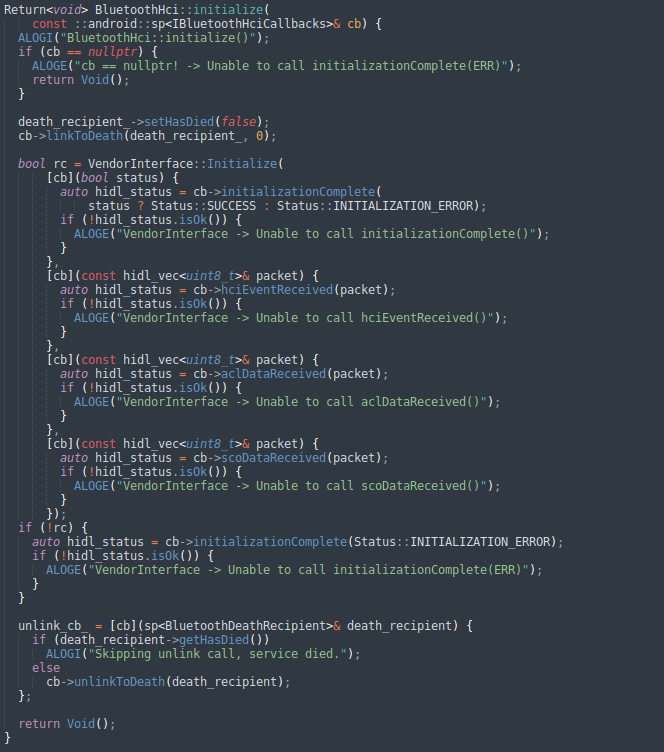
/system/bt/hci/src/hci\_layer.cc



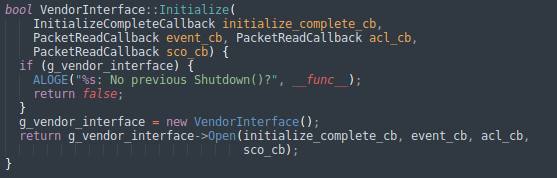
system/bt/hci/src/hci\_layer\_android.cc



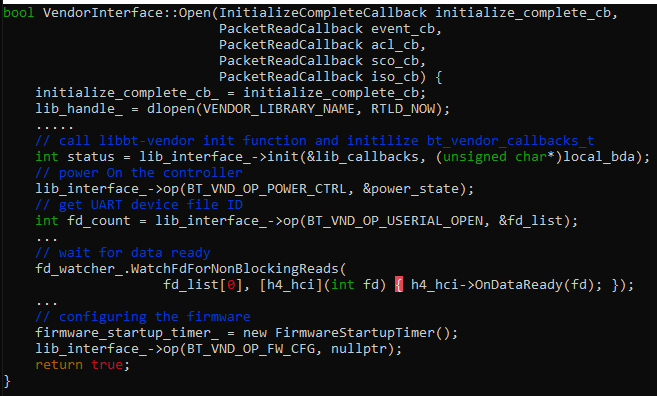
hardware/interfaces/bluetooth/1.0/IBluetoothHci.hal



hardware/interfaces/bluetooth/1.0/default/bluetooth\_hci.cc



hardware/interfaces/bluetooth/1.0/default/vendor\_interface.cc



hardware/interfaces/bluetooth/1.0/default/vendor\_interface.cc

**Refer**

https://blog.csdn.net/xiaojsj111/article/details/12647923?spm=1001.2101.3001.6650.11&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-11-12647923-blog-115728286.235%5Ev27%5Epc\_relevant\_3mothn\_strategy\_recovery&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-11-12647923-blog-115728286.235%5Ev27%5Epc\_relevant\_3mothn\_strategy\_recovery&utm\_relevant\_index=19

<https://blog.csdn.net/kv110/article/details/117457151?spm=1001.2101.3001.6650.14&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-117457151-blog-115728286.235%5Ev27%5Epc_relevant_3mothn_strategy_recovery&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-117457151-blog-115728286.235%5Ev27%5Epc_relevant_3mothn_strategy_recovery&utm_relevant_index=22>

<https://blog.csdn.net/edmond999/article/details/115728286>

https://blog.csdn.net/u011279649/article/details/119606531?spm=1001.2101.3001.6650.19&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-19-119606531-blog-115728286.235%5Ev27%5Epc\_relevant\_3mothn\_strategy\_recovery&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-19-119606531-blog-115728286.235%5Ev27%5Epc\_relevant\_3mothn\_strategy\_recovery&utm\_relevant\_index=27