高通系统学习记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档名称 | 高通系统学习记录 | 版本号 | 1.00.00 | | |
| 文档编号 |  | | |
| 文档类别 | 使用说明 | 文档阶段 | 初稿 | | |
| 项目名称 | TWS | 作者 | 何继胜 | | |
| 承担部门 | 软件研发部 | 批准 |  | | |
| 文档日期 | 2019/7/10 | 使用范围 | 公司内部 | 页数 | 18 |

目录

[1 文档版本 3](#_Toc15407116)

[2 Task系统 3](#_Toc15407117)

[2.1 SendMessage 3](#_Toc15407118)

[2.2 常用task 4](#_Toc15407119)

[3 神秘的RFC\_REGISTER\_(REQ/CFM) 4](#_Toc15407120)

[4 通道与UUID 4](#_Toc15407121)

[5 SPP 5](#_Toc15407122)

[5.1 Server 5](#_Toc15407123)

[5.2 Task模型 7](#_Toc15407124)

[5.3 问题 8](#_Toc15407125)

[5.4 最终解决方法 9](#_Toc15407126)

[6 GAIA 9](#_Toc15407127)

[6.1 命令格式 10](#_Toc15407128)

[6.1.1 命令格式 10](#_Toc15407129)

[6.1.2 帧格式 10](#_Toc15407130)

[6.1.3 参数含义 10](#_Toc15407131)

[6.1.4 ACK 10](#_Toc15407132)

[6.1.5 已经实现的命令 11](#_Toc15407133)

[6.2 初始化 11](#_Toc15407134)

[6.3 建立连接 13](#_Toc15407135)

[6.4 数据交互 13](#_Toc15407136)

[6.4.1 手机->TWS MESSAGE\_MORE\_DATA 13](#_Toc15407137)

[6.4.2 TWS->手机 MESSAGE\_MORE\_SPACE 14](#_Toc15407138)

[6.5 Other 14](#_Toc15407139)

[6.6 BLE/SPP/RFCOMM 15](#_Toc15407140)

[6.7 参考文档 15](#_Toc15407141)

[6.8 实现的例子 15](#_Toc15407142)

[7 GAIA与SPP 16](#_Toc15407143)

[8 小知识 17](#_Toc15407144)

[8.1 左右耳机区分 17](#_Toc15407145)

[8.2 Task 17](#_Toc15407146)

[8.3 耳机-盒子状态 17](#_Toc15407147)

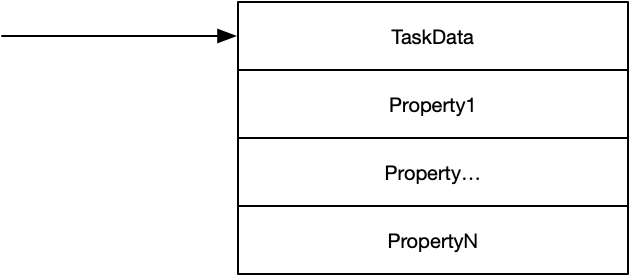
[9 TrueWireless Stereo 18](#_Toc15407148)

# 文档版本

|  |  |  |
| --- | --- | --- |
| 日期 | 版本号 | 简要描述 |
| 2019/7/4 | 1.00.00 | 初稿 |
| 2019/7/30 | 1.00.01 | 添加关于GAIA分析 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Task系统

TaskData与特定类型TaskData的关系。



一般情况下，将TaskData的数据传递与注册，等到TaskData的handler被调用的时候，一般会被强制转换为对应的真实类型。

## SendMessage

用途：立即向指定的任务发送一条消息。

定义：

/\*!

@brief Send a message to the corresponding task immediately.

The message will be passed to free after delivery.

@param task The task to deliver the message to.

@param id The message type identifier.

@param message The message data (if any).

\*/

void MessageSend(Task task, MessageId id, void \*message)

## 常用task

* CMTask
* SMTask
* SPPTask
* GaiaTask
* TransportTask
* UpgradeTask
* ……

# 神秘的RFC\_REGISTER\_(REQ/CFM)

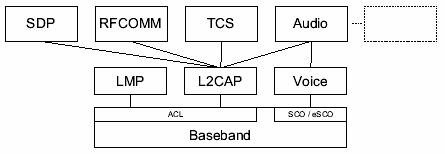
Create an entry in the connection map to enable incoming primitives on this server channel to be mapped to the correct task.

创建一个类似Server的task，用于管理连接，如果有新连接，再将数据丢给子task。

该请求实际请求L2CAP，带上channel，开启一个Server。（纯粹个人见解）

# 通道与UUID

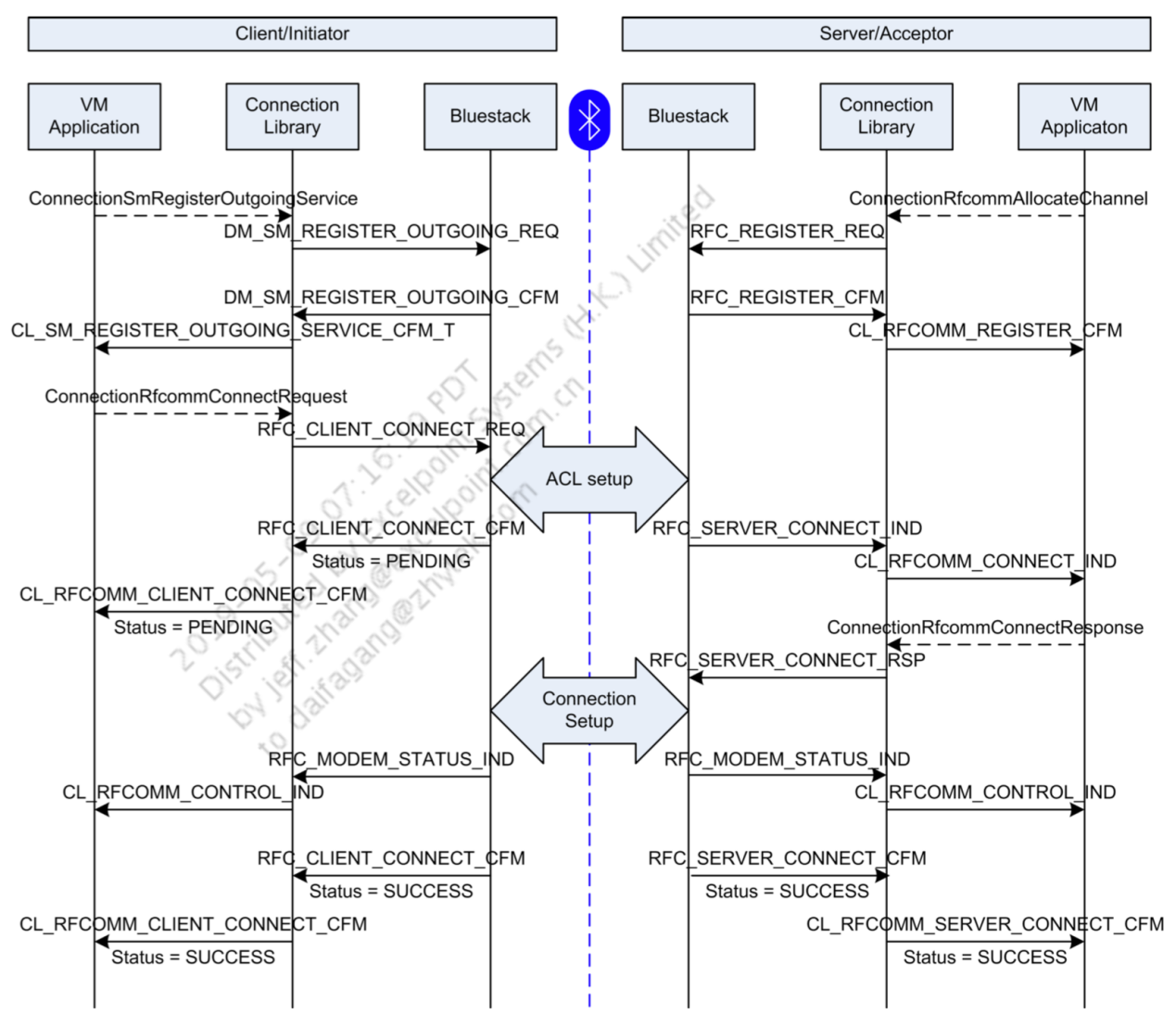
L2CAP：Logical Link Control and Adaptation Protocol，即**逻辑链路控制和适配协议。**



SPP、RFCOMM、TCS等服务，在SDP中注册一个特有的UUID。

注册的时候，需要告诉OS系统，使用的是那个Channel（L2CAP）。系统收到请求之后，给予确认消息（CFM）。

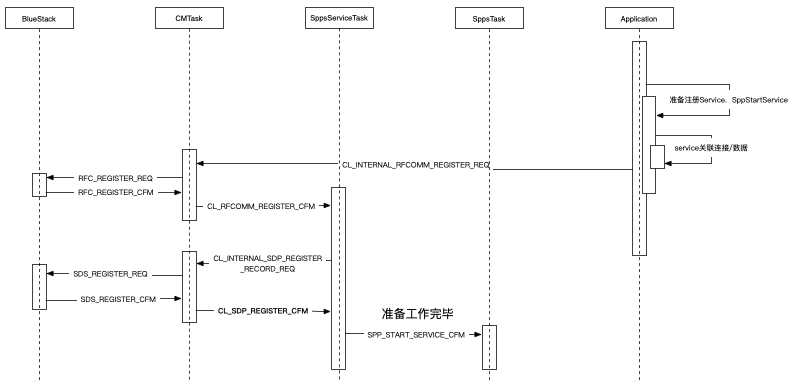
# SPP



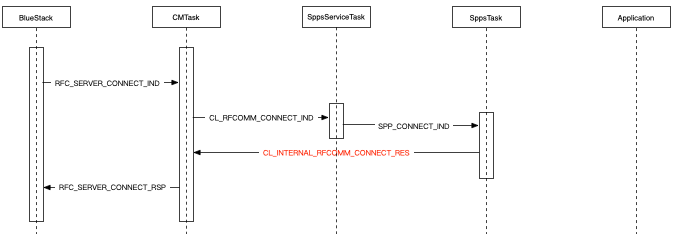
* Bluestack 系统层实现，无代码
* Connection Library 通讯库，厂商提供的库
* VM Application 公司需要使用的层

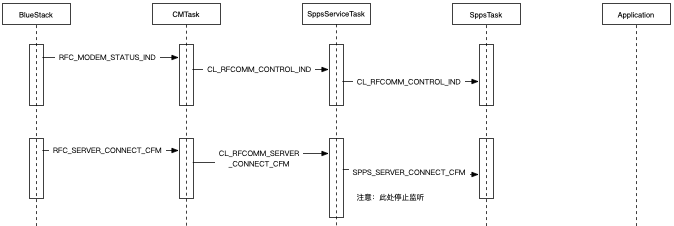
## Server

* 注册bt，通知bt-stack，提供spp功能，最终生成一个Listen的channel。

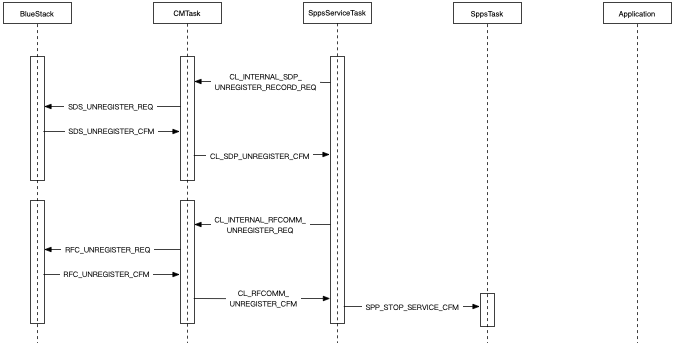


* 建立连接，建立connection。

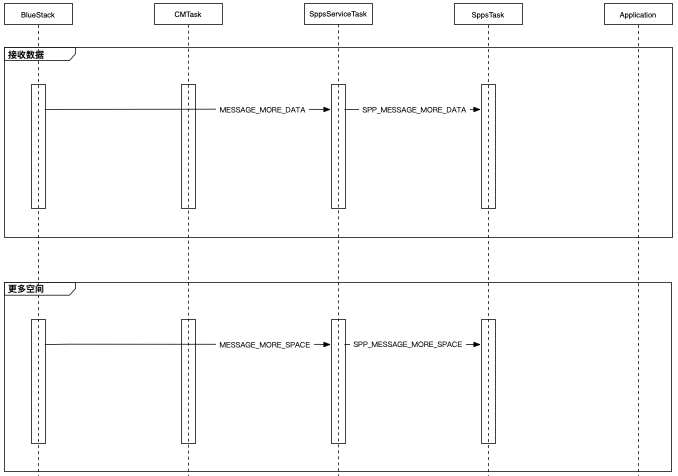




* 关闭监听，不再建立新链接。



* 收发数据。



Bluestack在查看是否有空间、有数据抵达的时候，会向所有的task发送通知，有数据来了、有空来了，快来处理。

## Task模型

每个模型，都有负责的工作，如果想去执行特定操作，需要发送消息到特定的task中。

CMTask：底层核心事件。

SppsServiceTask关注的事件：

* CL\_RFCOMM\_REGISTER\_CFM
* CL\_RFCOMM\_UNREGISTER\_CFM
* CL\_SDP\_REGISTER\_CFM
* CL\_SDP\_UNREGISTER\_CFM
* CL\_RFCOMM\_CONNECT\_IND 连接来的时候，来了此事件

SppsDataTask关注的事件：

## 问题

在一段时间之后，出现速度慢下去的情况。

/\*!

A Source has more data. The message content is a #MessageMoreData.

This message is sent to any task associated with the Source using MessageSinkTask().

The frequency of this message can be controlled with #VM\_SOURCE\_MESSAGES.

\*/

#define MESSAGE\_MORE\_DATA (SYSTEM\_MESSAGE\_BASE\_ + 33)

/\*!

A Sink has more space. The message content is a #MessageMoreSpace.

This message is sent to any task associated with the Sink using MessageSinkTask().

The frequency of this message can be controlled with #VM\_SINK\_MESSAGES.

\*/

#define MESSAGE\_MORE\_SPACE (SYSTEM\_MESSAGE\_BASE\_ + 34)

Spp在建立连接的时候（事件: CL\_INTERNAL\_RFCOMM\_CONNECT\_RES），

void connectionHandleRfcommConnectRes(const CL\_INTERNAL\_RFCOMM\_CONNECT\_RES\_T\* res)

{

MAKE\_PRIM\_T(RFC\_SERVER\_CONNECT\_RSP);

prim->conn\_id = PanicZero(SinkGetRfcommConnId(res->sink));

SourceConfigure( StreamSourceFromSink(res->sink), *VM\_SOURCE\_MESSAGES*, *VM\_MESSAGES\_NONE*);

MessageStreamTaskFromSink(res->sink, res->theAppTask);

if (res->response)

{

prim->response = *RFC\_ACCEPT\_SERVER\_CONNECTION*;

}

else

{

prim->response = *RFC\_DECLINE\_SERVER\_CONNECTION*;

}

prim->max\_payload\_size = res->config.max\_payload\_size;

/\* Modem Status parameters to be used during connection

\*/

prim->modem\_signal = res->config.modem\_signal & MODEM\_SIGNAL\_MASK;

prim->break\_signal = encode\_break\_signal(res->config.break\_signal);

prim->msc\_timeout = res->config.msc\_timeout;

/\* The following parameters are for AMP or have default values for

\* streams.

\*/

prim->flags = 0;

prim->priority = 0;

prim->total\_credits = 0;

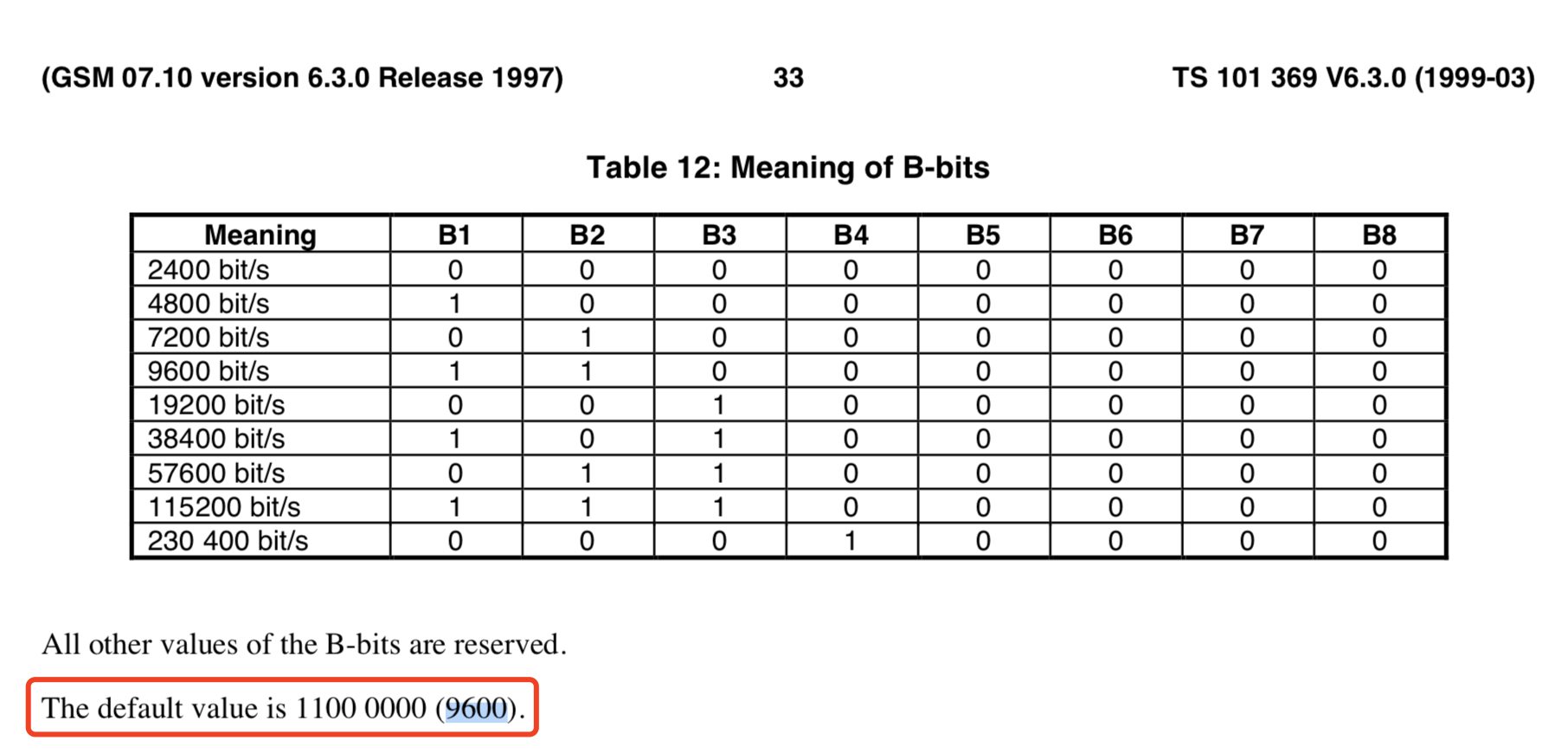
prim->remote\_l2cap\_control = 0;

prim->local\_l2cap\_control = 0;

VmSendRfcommPrim(prim);

}

是否是类似这个频率的设置，导致发送缓慢？



调试的时候，使用baud\_rate的值为3。

## 最终解决方法

在连接成功之后，调用void linkPolicyUseAvrcpSettings(Sink slcSink);

# GAIA

GAIA：Generic Application Interface Architecture（高通独有协议格式）。

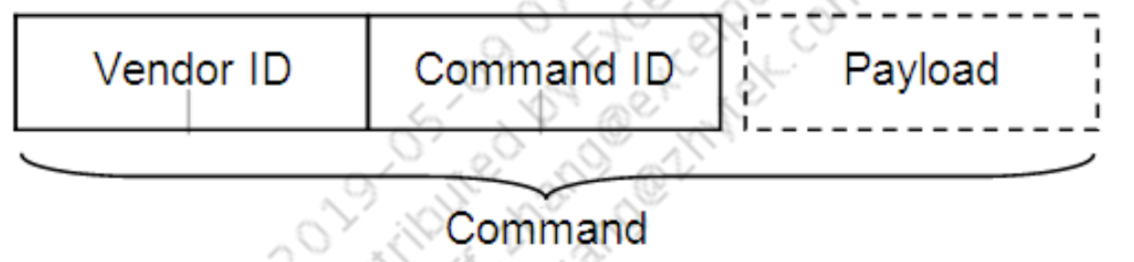
GAIA是基于SPP/RFCOMM/BLE的上层封装的实现。让用户不在关注到底是使用何种传输方式。

如果需要连接设备，使用标准的SPP的UUID或者GAIA特有的UUID。

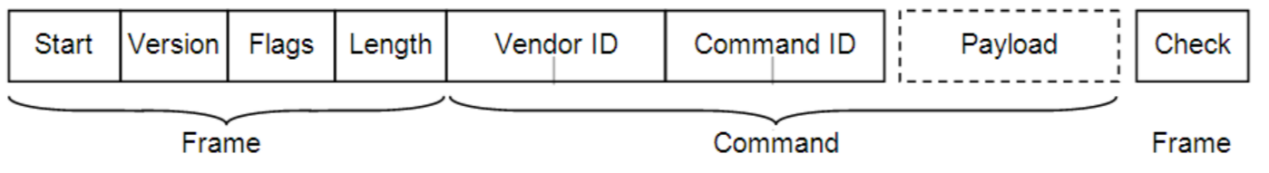
RWCP: Reliable Write Command Protocol (RWCP)。

## 命令格式

### 命令格式



### 帧格式



### 参数含义

* Start: One octet with the fixed value 0xff.
* Version: One octet. This field indicates the protocol version in use, currently 1.
* Flags: One octet. Bits within this field control protocol options.
  + Bit[0]: If set, a single octet check is present
  + Bit[1:7]: Reserved, must be 0
* Vendor ID: This 16-bit field qualifies the command ID. All commands in this document have the
* Vendor ID assigned to QTIL by the Bluetooth SIG, that is, 0x000a. Command ID: This 16-bit field identifies the individual command.
* Payload: The payload contains any information required to be passed by a specific command. It consists of zero or more octets depending on the command.
* Check: One octet. If present this field is determined by XORing together the other octets in the packet.

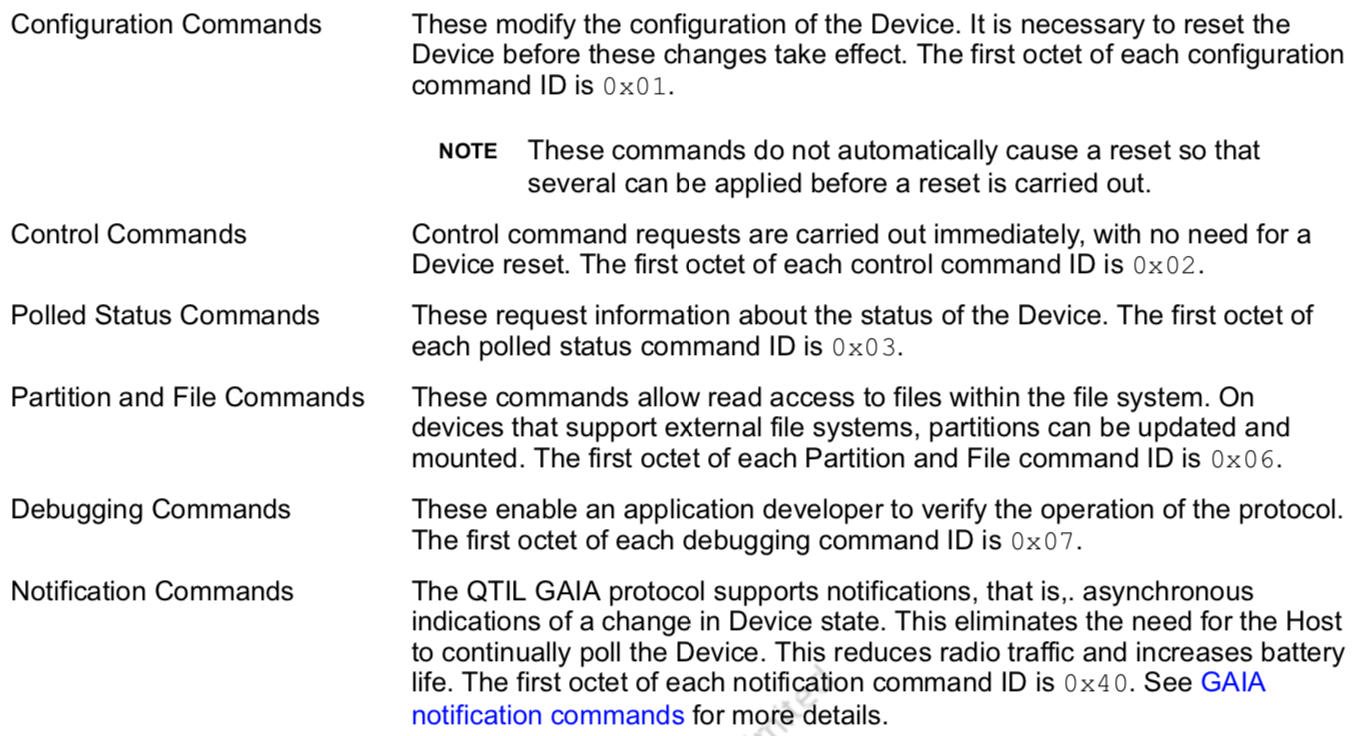
### ACK

每一个命令都需要ACK确认。

示例：发送Command=>0X0001的消息，对方需要回复Command=>0X8001的确认包，负载为状态码。

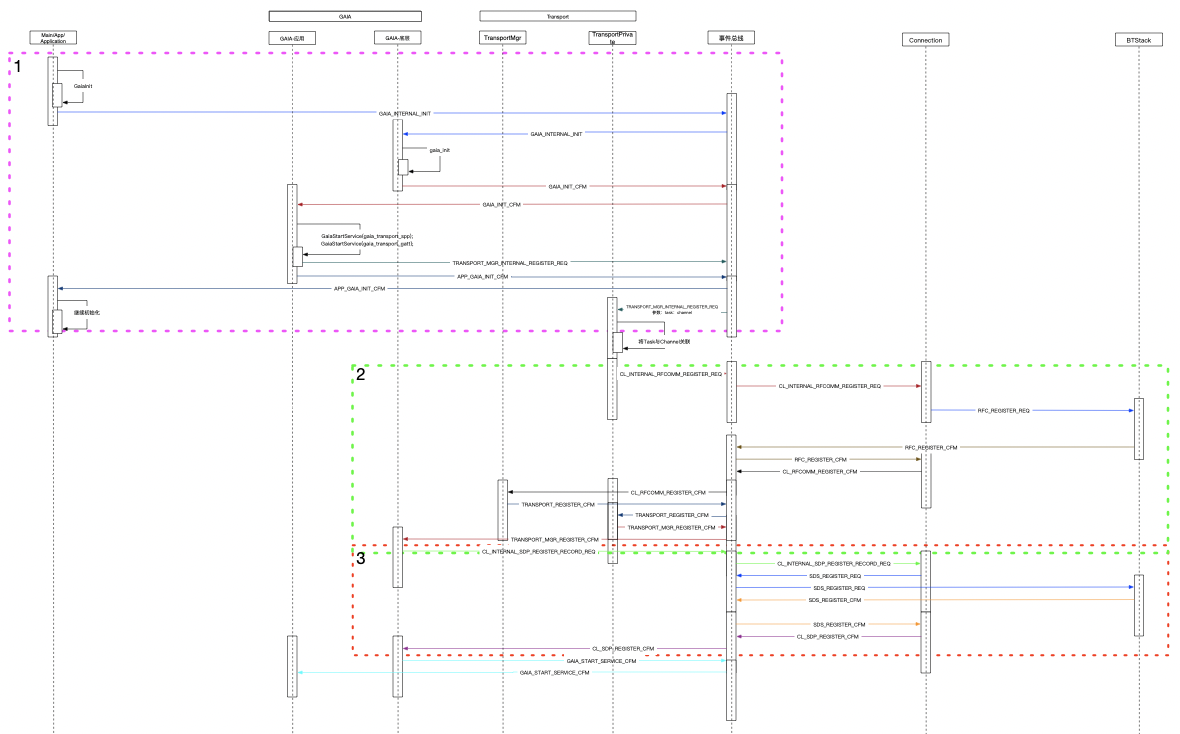
如果一次发送多个命令，不保证ACK回来的顺序和发送的顺序是一致的。“The QTIL GAIA protocol does not specify that acknowledgements are received in order.”

### 已经实现的命令



可以使用已经实现的命令，测试开发板。

## 初始化

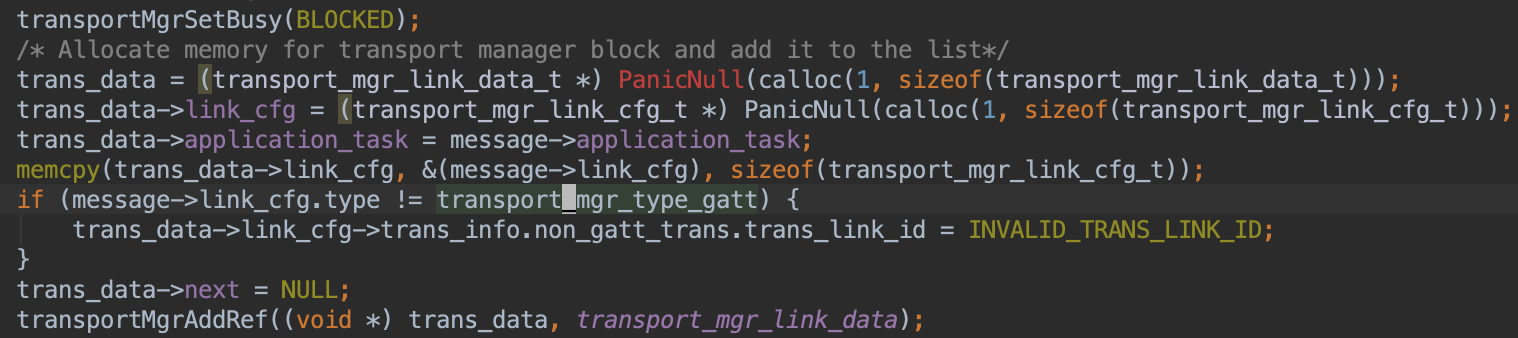


1. 初始化

主线程发起请求，初始化GAIA应用库和底层库；在底层库确认初始化完毕之后，发送GAIA\_INIT\_CFM消息给应用层，应用层请求传输层，请给我建立传输通道（请求消息内容【消息处理的Task：占用的消息通道】）。并通知Application/Main的Task，可以继续初始化其他任务了。

1. 建立Channel

传输层管理存储初始化信息到对应的数据中，如下图：



并设置当前传输层忙，不在接收新的连接。通知了Connection，让它转告BTStack，这个通道数据，转发到传输层，所有的数据，我来处理，因为我这里缓存了所有的连接请求数据，我知道该给谁。

在BTStack确认接受请求时，通知Connection，Connection在收到结果时，告知TransportMgr。TransportMgr最终在收到TRANSPORT\_REGISTER\_CFM时，解除传输层繁忙状态，并确定连接建立成功，告知GAIA底层，传输层建立连接完毕。

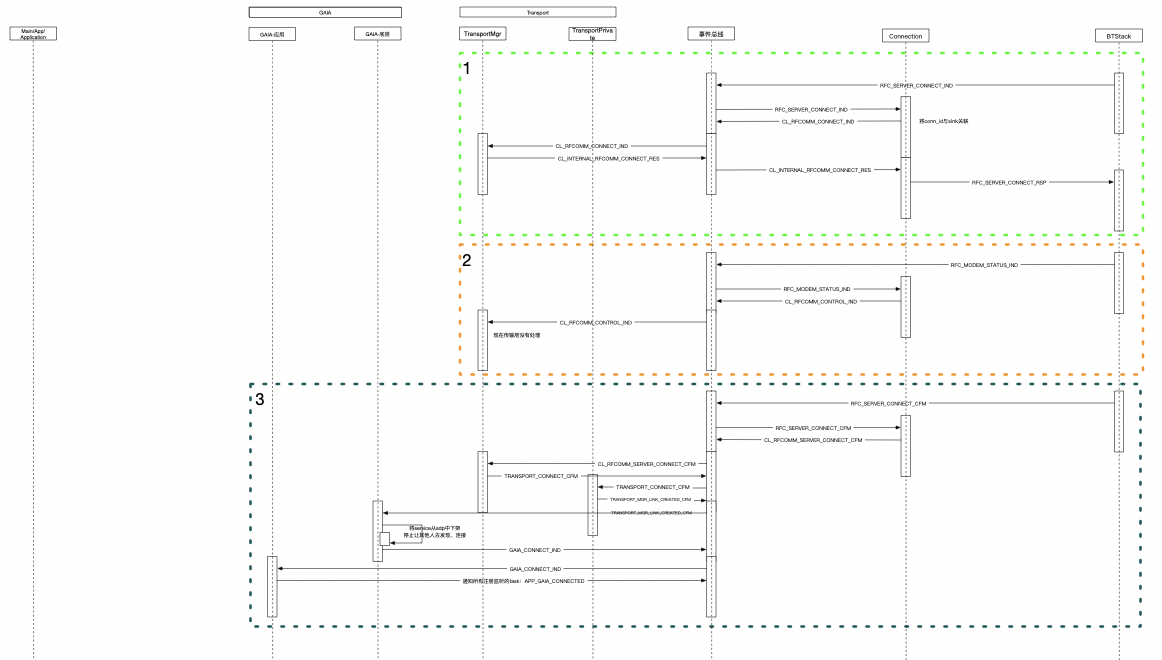
1. 建立Service

使用BTStack标准化的方式建立可以被发现的服务。最终GAIA底层在收到CL\_SDP\_REGISTER\_CFM消息时，通知GAIA应用层。

**各个命令的用途：**

* GAIA\_INTERNAL\_INIT：请求初始化GAIA。
* GAIA\_INIT\_CFM：GAIA底层初始化完毕
* APP\_GAIA\_INIT\_CFM：GAIA初始化请求处理完毕
* TRANSPORT\_MGR\_INTERNAL\_REGISTER\_REQ：请求初始化传输层
* CL\_INTERNAL\_RFCOMM\_REGISTER\_REQ：请求初始化RFCOMM传输
* RFC\_REGISTER\_REQ：请求BTStack分配传输端口
* RFC\_REGISTER\_CFM：确认BTStack分配传输端口
* CL\_RFCOMM\_REGISTER\_CFM：RFCOMM初始化完毕
* TRANSPORT\_REGISTER\_CFM：传输层分配完毕
* TRANSPORT\_MGR\_REGISTER\_CFM：通知GAIA，传输层准备完毕
* CL\_INTERNAL\_SDP\_REGISTER\_RECORD\_REQ：请求SDP对外发布服务
* SDS\_REGISTER\_REQ：请求BTStack对外开放服务
* SDS\_REGISTER\_CFM：确认BTStack对外开放服务
* CL\_SDP\_REGISTER\_CFM：SDP服务注册成功
* GAIA\_START\_SERVICE\_CFM：通知客户端，GAIA准备完毕

## 建立连接



1. 请求连接

协商连接参数，如：最大负载、连接超时等。

1. 模块状态

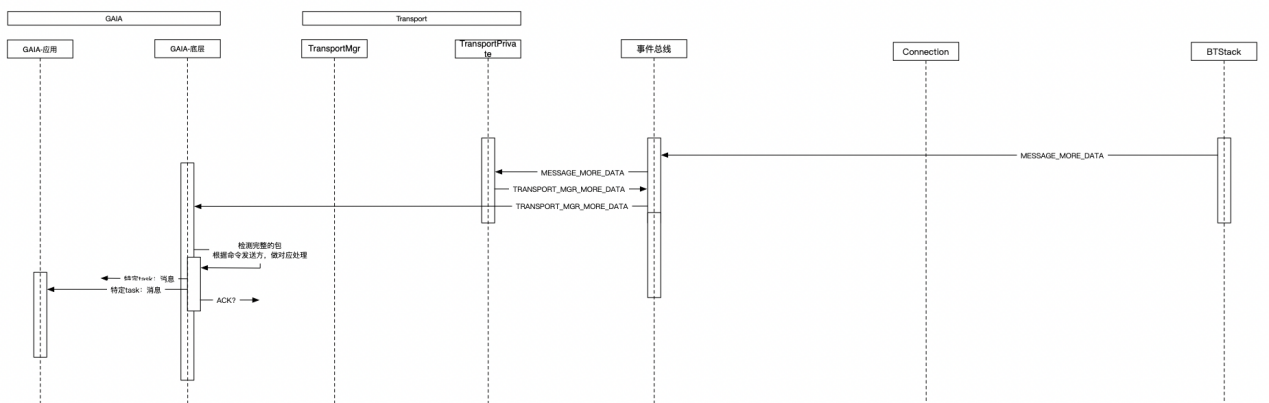
现在在应用层和库中，忽略该消息的处理。

1. 建立连接

最终建立连接，在GAIA应用层，会收到GAIG\_CONNECT\_IND消息。GAIA应用层，会通知所有注册的客户端，发送APP\_GAIA\_CONNECTED的消息通知。

## 数据交互

### 手机->TWS MESSAGE\_MORE\_DATA



GAIA命令中厂商字段，在发送新消息和ACK的时候，需要带上该字段。

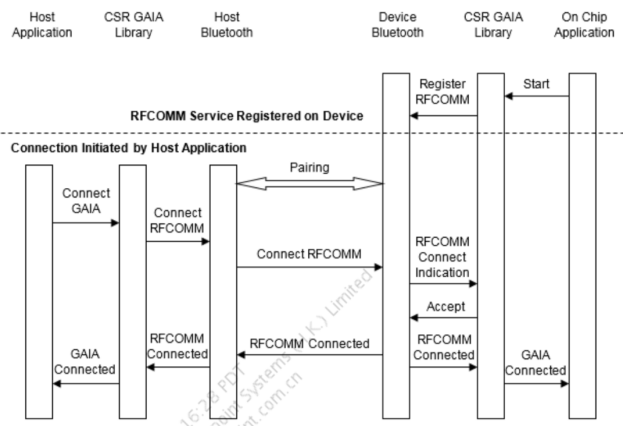
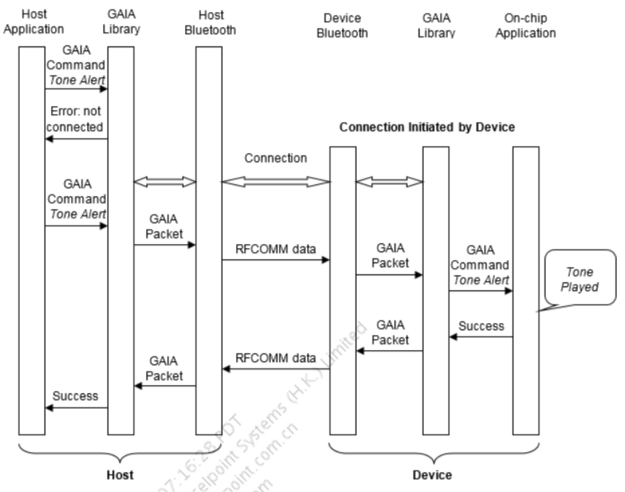
需要在GAIA底层中添加新的命令和消息体。使用新的message\_id，将消息转发给GAIA应用层。

### TWS->手机 MESSAGE\_MORE\_SPACE

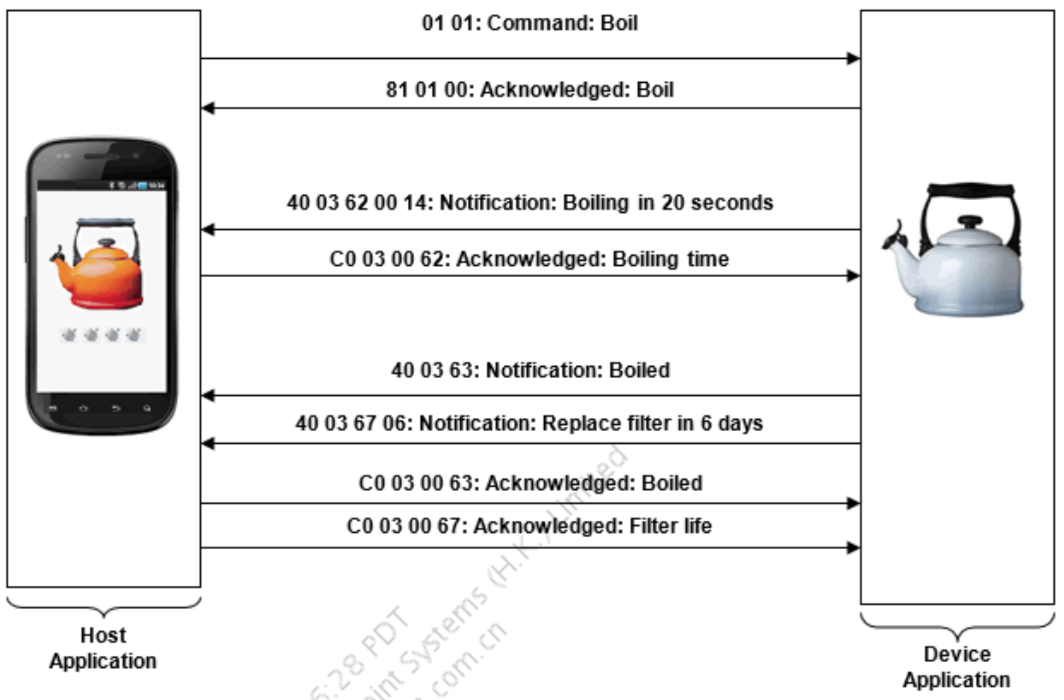


应用层在发送数据之后，这一块内存不能马上销毁，需要等到发送状态确认，才能决定是释放内存，还是继续发送。

## Other

初始化流程 发送命令流程



## BLE/SPP/RFCOMM

在gaia中，根据不同的初始化，使用不同的传输方式。

RFCOMM模式：00001107-D102-11E1-9B23-00025B00A5A5

SPP模式：00001101-0000-1000-8000-00805F9B34FB

BLE模式：

对于rfcomm和spp模式，使用的相同的蓝牙channel，但是对外uuid不同。

## 参考文档

《Qualcomm GAIA Command Reference.pdf》

《Qualcomm GAIA Ecosystem Reference.pdf》

《GAIA Control Demo Application for Android User Guide.pdf》

## 实现的例子

具备发送命令、定时发送音频数据。

在播放音乐的时候，速度会有一定的下降；在拨打电话的时候，直接下降的4k以下。需要查看hsp和a2dp协议的实现，需要确认是否降低了sink的powertable。

appLinkPolicyUpdatePowerTable该函数，在不同状态，设置不同的值。测试的时候发现设置为高功耗的时候，GAIA的速度，可以维持在60k以上。

# GAIA与SPP

SPP初始化的时候，设置RFComm

|  |
| --- |
| void SppStartService(Task theAppTask) {  /\* Is there already an SPP service initiated? \*/  if (sppsClientTask) {  sendSppStartServiceCfm(spp\_start\_already\_started);  } else {  sppsClientTask = theAppTask;  ConnectionRfcommAllocateChannel(  (Task) &sppsServiceTask,  SPP\_DEFAULT\_CHANNEL  );  }  } |

GAIA初始化的时候，设置RFComm

|  |
| --- |
| void gaiaTransportStartService(gaia\_transport\_type transport\_type)  {  #if defined GAIA\_TRANSPORT\_RFCOMM || defined GAIA\_TRANSPORT\_SPP  transport\_mgr\_link\_cfg\_t link\_cfg;  #endif  switch (transport\_type)  {  #if defined GAIA\_TRANSPORT\_RFCOMM || defined GAIA\_TRANSPORT\_SPP  case gaia\_transport\_rfcomm:  case gaia\_transport\_spp:  transport\_registered = transport\_type;  link\_cfg.type = transport\_mgr\_type\_rfcomm;  link\_cfg.trans\_info.non\_gatt\_trans.trans\_link\_id = SPP\_DEFAULT\_CHANNEL;  TransportMgrRegisterTransport(&gaia->task\_data, &link\_cfg);  break;  #endif  #ifdef GAIA\_TRANSPORT\_GATT  case gaia\_transport\_gatt:  break;  #endif  default:  GAIA\_TRANS\_DEBUG(("Unknown Gaia transport %d\n", transport\_type));  GAIA\_PANIC();  gaiaTransportCommonSendGaiaStartServiceCfm(transport\_type, NULL, FALSE);  break;  }  } |

GAIA与官方的SPPserver程序，初始化的时候，使用的channel，都是SPP\_DEFAULT\_CHANNEL，存在一定的冲突。建议修改SPP程序，使用其他的channel。

# Peer两只耳机

注意：以下的流程图中，有av和avrcp的task，可能有些错误，需要后期慢慢理解、理清。

两只耳机之间，使用a2dp和avrcp协议进行通讯。

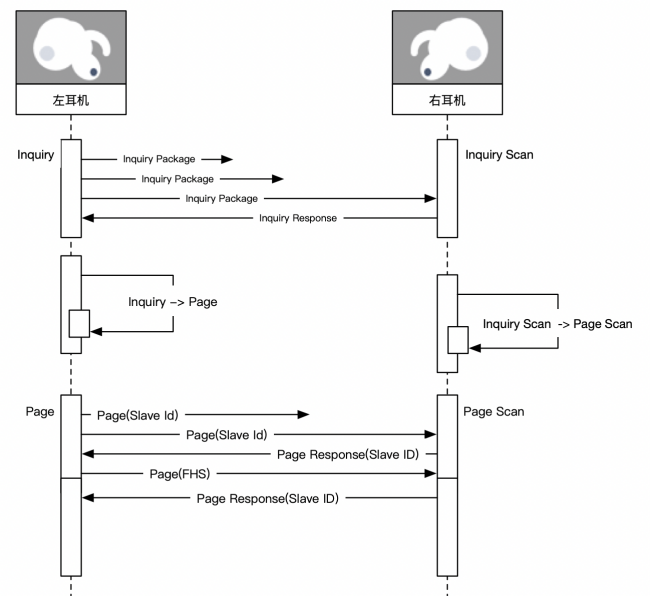
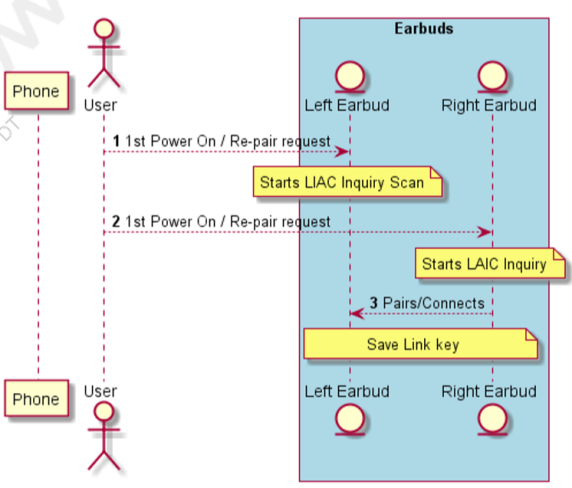
ACL：

## 状态切换



|  |
| --- |
| **PEER\_SIG\_STATE\_DISCONNECTED** |
| **Exit：**从任务中去掉当前的scan请求。如果还有其他类似请求，参数以其他请求为准；如果没有其它请求，停止scan。  **Enter：**通知Scan，扫描开始了，低速的，参数如下：SCAN\_MAN\_USER\_PEERSIG, SCAN\_MAN\_PARAMS\_TYPE\_SLOW。 |
| **PEER\_SIG\_STATE\_CONNECTING\_ACL** |
| 尝试建立ACL连接。ACL建立成功之后，触发PEER\_SIG\_INTERNAL\_STARTUP\_REQ请求。  MessageSendConditionally(&peer\_sig->task, PEER\_SIG\_INTERNAL\_STARTUP\_REQ, message, appConManagerCreateAcl(&req->peer\_addr)); 注意：完成ACL连接（不管失败或是成功）之后，都会发送PEER\_SIG\_INTERNAL\_STARTUP\_REQ事件。 |
| **PEER\_SIG\_STATE\_CONNECTING\_LOCAL** |
| 建立AvAvrcp连接(appAvAvrcpConnectRequest)。 |
| **PEER\_SIG\_STATE\_CONNECTING\_REMOTE** |
| 远程请求建立AvAvrcp连接。 |
| **PEER\_SIG\_STATE\_CONNECTED** |
| 连接建立成功。 |

## 配对



如何知道对方的？根据什么规则，允许对方接入的。

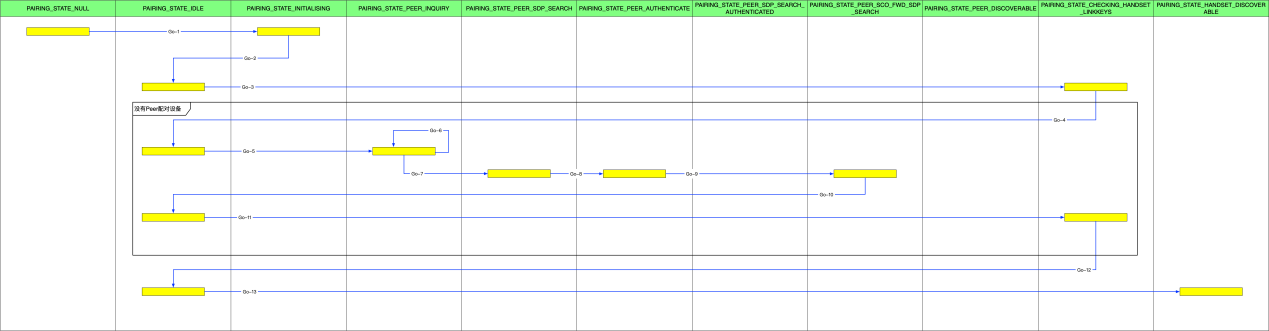
在配对的过程中，需要一定的时间。在指定的时间内，有两个RSSI信号比较，并设置阀值，如果小于该值，不进行配对。

记录的信号，应该是越来越强，如果变弱了，说明距离变远，不能使用。

appHanndleClDmInquireResult处理这些事情。

inquiry\_status\_result:收到多次设备的响应。在指定的时间到达时，状态变更为inquiry\_status\_ready。

在inquiry\_status\_ready状态下，如果有合适的peer地址，则进入PAIRING\_STATE\_PEER\_SDP\_SEARCH。



[Go-1]: 启动的时候初始化

[Go-2]: 初始化完毕

[Go-3]: 触发检测HandSetLinkKeys

[Go-4]: 检测HandSetLinkKeys完毕

[Go-5]: **规则**触发进行配对

[Go-6]: 定时检测配对的Peer地址

[Go-7]: 有合法Peer地址

[Go-8]: 发现到了设备

[Go-9]: 绑定Peer设备成功

[Go-10]: 查找该Peer设备是否有FWD-SCO的服务

[Go-11]: 校验Peer设备是否配对成功

[Go-12]: 有配对好的设备

[Go-13]: 可以让用户发现耳机了

|  |
| --- |
| PAIRING\_STATE\_INITIALISING  Enter：开启SDP服务  Exit：发送异步请求PAIRING\_INTERNAL\_CHECK\_HANDSET\_LINKKEYS |
| PAIRING\_STATE\_IDLE  Enter：停止Scan，释放Pair功能的锁  Exit：设置Scan为高频操作，并锁定Pair功能，不能让其他功能调用 |
| PAIRING\_INTERNAL\_CHECK\_HANDSET\_LINKKEYS  Enter：  Exit：None |
| PAIRING\_STATE\_PEER\_INQUIRY  Enter：让btstack去执行inquiry，btstack会发送搜索请求，设备收到了，会给出响应，事件为[DM\_HCI\_INQUIRY\_RESULT\_WITH\_RSSI\_IND]，应用层[inquirySendResult]会根据信号强度，进行过滤、排除。如果有合适的设备，则进入下一阶段[PAIRING\_STATE\_PEER\_SDP\_SEARCH]。如果长时间找不到合适的设备，会触发定时器，让设备重新inquiry。  Exit：删除定时器 |
| PAIRING\_STATE\_PEER\_SDP\_SEARCH  Enter：发送CL\_INTERNAL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_REQ，让btstack去根据指定的MAC地址，去查询属性信息。在有响应的时候，会收到CL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_CFM该响应。  Exit：None |
| PAIRING\_STATE\_PEER\_AUTHENTICATE  Enter：绑定之前查询到的设备的MAC。  Exit：None |
| PAIRING\_STATE\_PEER\_SCO\_FWD\_SDP\_SEARCH  Enter：发送CL\_INTERNAL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_REQ，让btstack去根据指定的MAC地址，去查询属性信息。在有响应的时候，会收到CL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_CFM该响应。  Exit：None |
| PAIRING\_STATE\_HANDSET\_DISCOVERABLE  Enter：让耳机可以被发现  Exit：停止可以被发现 |

Task流转图：



在找到合适的设备时，进行服务发现：



## 连接-同步

### 连接

绑定成功之后，进行设备连接：



### 同步

连接成功之后，进行设备交换数据：





交换完信息之后，触发事件[RULE\_EVENT\_PEER\_SYNC\_VALID]。

## 配对规则事件

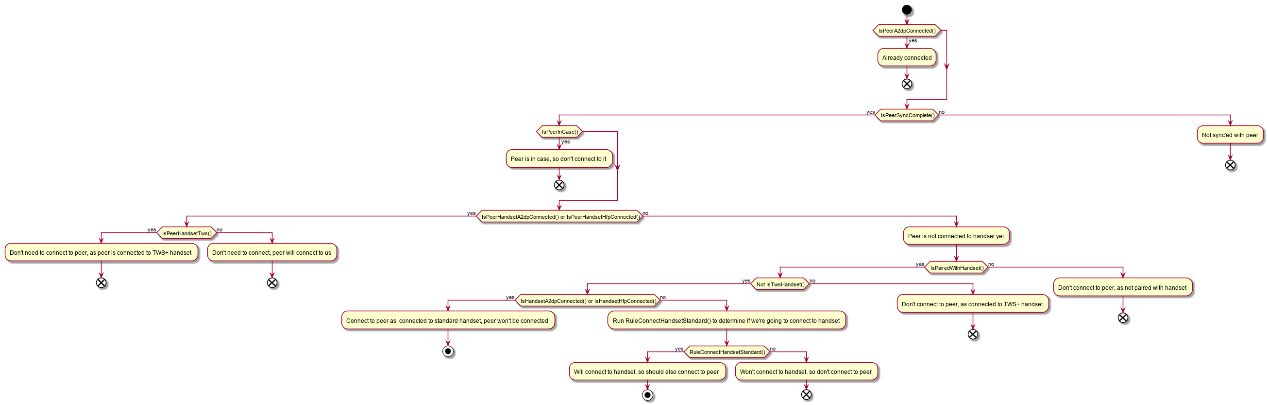
在同步完信息之后，触发事件【RULE\_EVENT\_PEER\_SYNC\_VALID】。校验如下条件。

注意：只有在盒子外面，才会通过规则校验。

### ruleSyncConnectPeer

根据当前状态，确定是否需要连接A2DP/AVRCP。

Condition:



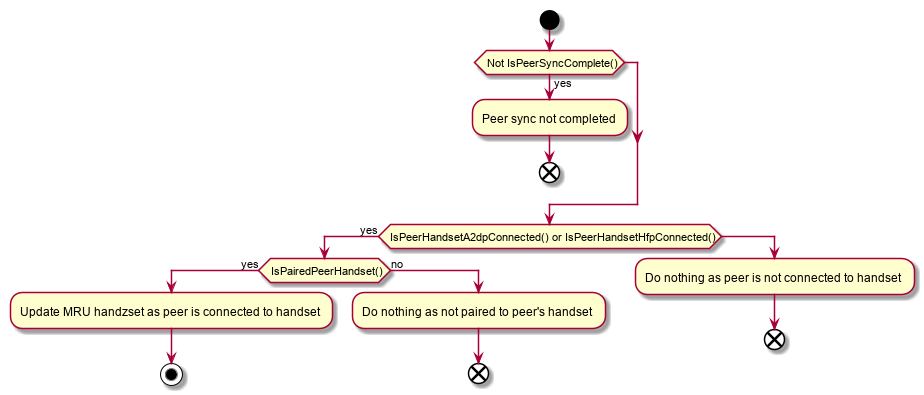
Action: CONN\_RULES\_CONNECT\_PEER

* appScoFwdConnectPeer();
* appAvConnectPeer();

### ruleUpdateMruHandset

MRU: most recently used。确定手机上的信息是否需要更新

Condition:



Action: CONN\_RULES\_UPDATE\_MRU\_PEER\_HANDSET

* appPeerSyncGetPeerHandsetAddr(&peer\_handset\_addr);
* appDeviceUpdateMruDevice(&peer\_handset\_addr);

### ruleSyncDisconnectPeer

Condition:Peer是否连接

Action: CONN\_RULES\_DISCONNECT\_PEER

* appScoFwdDisconnectPeer
* appAvDisconnectPeer

### ruleSyncDisconnectHandset

Condition:耳机是否与手机连接

Action:发送事件，断开连接（CONN\_RULES\_DISCONNECT\_HANDSET）

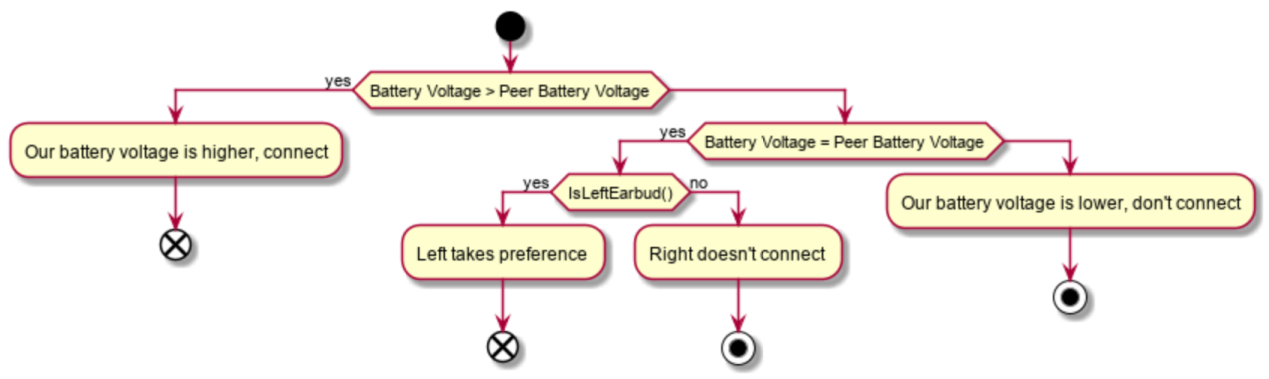
## 配对结束

## 耳机连接的动机

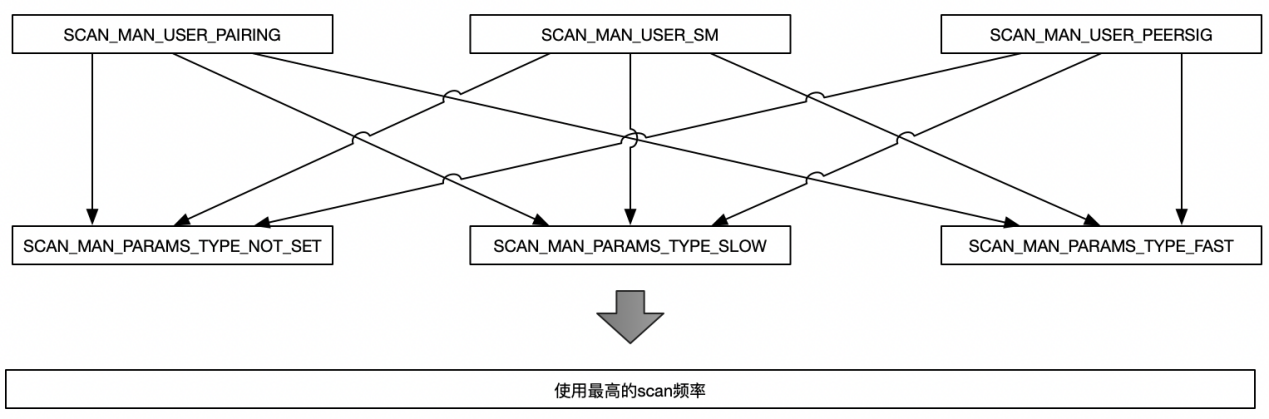
|  |  |
| --- | --- |
| 类型 | 说明 |
| RULE\_CONNECT\_USER |  |
| RULE\_CONNECT\_PAIRING |  |
| RULE\_CONNECT\_PEER\_SYNC |  |
| RULE\_CONNECT\_OUT\_OF\_CASE |  |
| RULE\_CONNECT\_LINK\_LOSS |  |
| RULE\_CONNECT\_PEER\_OUT\_OF\_CASE |  |

## 主从判断

### 电量判断



# Scan扫描



# 关注的事件

## 蓝牙事件ID分类

* MESSAGE\_BLUESTACK\_DM\_PRIM

DM：Device Manager

* MESSAGE\_BLUESTACK\_L2CAP\_PRIM

L2CAP：Logical Link Control and Adaptation Protocol

* MESSAGE\_BLUESTACK\_RFCOMM\_PRIM
* MESSAGE\_BLUESTACK\_SDP\_PRIM

CONN\_RULES\_SEND\_PEER\_SYNC

PEER\_SIG\_MSG\_CHANNEL\_RX\_IND

## Pairing

|  |
| --- |
| CL\_SDP\_REGISTER\_CFM  CL\_SDP\_UNREGISTER\_CFM  CL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_CFM  CL\_DM\_INQUIRE\_RESULT  CL\_DM\_LOCAL\_NAME\_COMPLETE  CL\_DM\_WRITE\_INQUIRY\_MODE\_CFM  CL\_DM\_WRITE\_INQUIRY\_ACCESS\_CODE\_CFM  CL\_SM\_GET\_AUTH\_DEVICE\_CFM  PAIRING\_INTERNAL\_PEER\_PAIR\_REQ  PAIRING\_INTERNAL\_HANDSET\_PAIR\_REQ  PAIRING\_INTERNAL\_CHECK\_HANDSET\_LINKKEYS  PAIRING\_INTERNAL\_TIMEOUT\_IND  PAIRING\_INTERNAL\_PEER\_PAIR\_CANCEL  PAIRING\_INTERNAL\_HANDSET\_PAIR\_CANCEL  PAIRING\_INTERNAL\_PEER\_SDP\_SEARCH  PAIRING\_INTERNAL\_HANDSET\_SDP\_SEARCH  PAIRING\_INTERNAL\_DISABLE\_SCAN  PEER\_SIG\_LINK\_KEY\_RX\_IND  PEER\_SIG\_LINK\_KEY\_TX\_CFM  PEER\_SIG\_PAIR\_HANDSET\_CFM  CL\_SM\_ADD\_AUTH\_DEVICE\_CFM |

## PeerSig

|  |
| --- |
| AV\_AVRCP |
| AV\_AVRCP\_CONNECT\_CFM  AV\_AVRCP\_CONNECT\_IND  AV\_AVRCP\_DISCONNECT\_IND  AV\_AVRCP\_VENDOR\_PASSTHROUGH\_IND  AV\_AVRCP\_VENDOR\_PASSTHROUGH\_CFM  AV\_AVRCP\_SET\_VOLUME\_IND  AV\_AVRCP\_VOLUME\_CHANGED\_IND  AV\_AVRCP\_PLAY\_STATUS\_CHANGED\_IND |
| PEER\_SIG\_INTERNAL |
| PEER\_SIG\_INTERNAL\_STARTUP\_REQ  PEER\_SIG\_INTERNAL\_INACTIVITY\_TIMER  PEER\_SIG\_INTERNAL\_LINK\_KEY\_REQ  PEER\_SIG\_INTERNAL\_PAIR\_HANDSET\_REQ  PEER\_SIG\_INTERNAL\_CONNECT\_HANDSET\_REQ  PEER\_SIG\_INTERNAL\_SHUTDOWN\_REQ  PEER\_SIG\_INTERNAL\_MSG\_CHANNEL\_TX\_REQ |

## PeerSync

|  |
| --- |
| PEER\_SIG\_MSG\_CHANNEL\_RX\_IND  PEER\_SIG\_MSG\_CHANNEL\_TX\_CFM  CONN\_RULES\_NOP |

## AV/AudioVideo

|  |
| --- |
| id >= AV\_INTERNAL\_AVRCP\_BASE && id < AV\_INTERNAL\_AVRCP\_TOP |
| AV\_INTERNAL\_AVRCP\_CONNECT\_IND  AV\_INTERNAL\_AVRCP\_CONNECT\_RES  AV\_INTERNAL\_AVRCP\_CONNECT\_REQ  AV\_INTERNAL\_AVRCP\_CONNECT\_LATER\_REQ  AV\_INTERNAL\_AVRCP\_DISCONNECT\_REQ  AV\_INTERNAL\_AVRCP\_DISCONNECT\_LATER\_REQ  AV\_INTERNAL\_AVRCP\_REMOTE\_REQ  AV\_INTERNAL\_AVRCP\_REMOTE\_REPEAT\_REQ  AV\_INTERNAL\_AVRCP\_VENDOR\_PASSTHROUGH\_RES  AV\_INTERNAL\_AVRCP\_VENDOR\_PASSTHROUGH\_REQ  AV\_INTERNAL\_AVRCP\_NOTIFICATION\_REGISTER\_REQ  AV\_INTERNAL\_AVRCP\_PLAY\_REQ  AV\_INTERNAL\_AVRCP\_PAUSE\_REQ  AV\_INTERNAL\_AVRCP\_PLAY\_TOGGLE\_REQ  AV\_INTERNAL\_AVRCP\_CLEAR\_PLAYBACK\_LOCK\_IND  AV\_INTERNAL\_AVRCP\_DESTROY\_REQ |
| id >= AVRCP\_MESSAGE\_BASE && id < AVRCP\_MESSAGE\_TOP |
| AVRCP\_CONNECT\_CFM  AVRCP\_CONNECT\_IND  AVRCP\_DISCONNECT\_IND  AVRCP\_PASSTHROUGH\_CFM  AVRCP\_PASSTHROUGH\_IND  AVRCP\_UNITINFO\_CFM  AVRCP\_UNITINFO\_IND  AVRCP\_SUBUNITINFO\_IND  AVRCP\_SUBUNITINFO\_CFM  AVRCP\_VENDORDEPENDENT\_CFM  AVRCP\_VENDORDEPENDENT\_IND  AVRCP\_SET\_ABSOLUTE\_VOLUME\_CFM  AVRCP\_SET\_ABSOLUTE\_VOLUME\_IND  AVRCP\_GET\_CAPS\_IND  AVRCP\_REGISTER\_NOTIFICATION\_IND  AVRCP\_EVENT\_PLAYBACK\_STATUS\_CHANGED\_IND  AVRCP\_EVENT\_VOLUME\_CHANGED\_IND  AVRCP\_REGISTER\_NOTIFICATION\_CFM |
| id >= AV\_INTERNAL\_A2DP\_BASE && id < AV\_INTERNAL\_A2DP\_TOP |
| AV\_INTERNAL\_A2DP\_SIGNALLING\_CONNECT\_IND  AV\_INTERNAL\_A2DP\_CONNECT\_REQ  AV\_INTERNAL\_A2DP\_CONNECT\_MEDIA\_REQ  AV\_INTERNAL\_A2DP\_DISCONNECT\_MEDIA\_REQ  AV\_INTERNAL\_A2DP\_DISCONNECT\_REQ  AV\_INTERNAL\_A2DP\_SUSPEND\_MEDIA\_REQ  AV\_INTERNAL\_A2DP\_RESUME\_MEDIA\_REQ  AV\_INTERNAL\_AVRCP\_UNLOCK\_IND  AV\_INTERNAL\_A2DP\_DESTROY\_REQ  AV\_INTERNAL\_A2DP\_INST\_SYNC\_IND  AV\_INTERNAL\_A2DP\_INST\_SYNC\_RES  AV\_INTERNAL\_A2DP\_CODEC\_RECONFIG\_IND |
| id >= A2DP\_MESSAGE\_BASE && id < A2DP\_MESSAGE\_TOP |
| A2DP\_SIGNALLING\_CONNECT\_IND  A2DP\_SIGNALLING\_CONNECT\_CFM  A2DP\_SIGNALLING\_DISCONNECT\_IND  A2DP\_MEDIA\_OPEN\_IND  A2DP\_MEDIA\_OPEN\_CFM  A2DP\_MEDIA\_START\_IND  A2DP\_MEDIA\_START\_CFM  A2DP\_MEDIA\_SUSPEND\_IND  A2DP\_MEDIA\_SUSPEND\_CFM  A2DP\_CODEC\_CONFIGURE\_IND  A2DP\_MEDIA\_CLOSE\_IND  A2DP\_ENCRYPTION\_CHANGE\_IND  A2DP\_MEDIA\_RECONFIGURE\_IND  A2DP\_MEDIA\_RECONFIGURE\_CFM  A2DP\_MEDIA\_AV\_SYNC\_DELAY\_IND  A2DP\_MEDIA\_AV\_SYNC\_DELAY\_CFM  A2DP\_MEDIA\_AV\_SYNC\_DELAY\_UPDATED\_IND |
| Other |
| AV\_AVRCP\_CONNECT\_CFM  AV\_AVRCP\_DISCONNECT\_IND  AV\_AVRCP\_SET\_VOLUME\_IND  AV\_AVRCP\_VOLUME\_CHANGED\_IND  AV\_AVRCP\_PLAY\_STATUS\_CHANGED\_IND |

## SM

|  |
| --- |
|  |
| INIT\_CFM |
|  |
| PAIRING\_PEER\_PAIR\_CFM  PAIRING\_HANDSET\_PAIR\_CFM |
|  |
| CON\_MANAGER\_CONNECTION\_IND |
|  |
| AV\_A2DP\_CONNECTED\_IND  AV\_A2DP\_DISCONNECTED\_IND  AV\_AVRCP\_CONNECTED\_IND  AV\_AVRCP\_DISCONNECTED\_IND  AV\_STREAMING\_ACTIVE\_IND  AV\_STREAMING\_INACTIVE\_IND |
|  |
| APP\_HFP\_CONNECTED\_IND  APP\_HFP\_DISCONNECTED\_IND  APP\_HFP\_SCO\_CONNECTED\_IND  APP\_HFP\_SCO\_DISCONNECTED\_IND |
|  |
| PHY\_STATE\_CHANGED\_IND |
|  |
| APP\_POWER\_SLEEP\_PREPARE\_IND  APP\_POWER\_SLEEP\_CANCELLED\_IND  APP\_POWER\_SHUTDOWN\_PREPARE\_IND  APP\_POWER\_SHUTDOWN\_CANCELLED\_IND |
|  |
| CONN\_RULES\_PEER\_PAIR  CONN\_RULES\_SEND\_PEER\_SYNC  CONN\_RULES\_PEER\_SEND\_LINK\_KEYS  CONN\_RULES\_CONNECT\_HANDSET  CONN\_RULES\_CONNECT\_PEER  CONN\_RULES\_CONNECT\_PEER\_HANDSET  CONN\_RULES\_UPDATE\_MRU\_PEER\_HANDSET  CONN\_RULES\_SEND\_STATE\_TO\_HANDSET  CONN\_RULES\_A2DP\_TIMEOUT  CONN\_RULES\_SCO\_TIMEOUT  CONN\_RULES\_SCO\_TRANSFER\_TO\_EARBUD  CONN\_RULES\_SCO\_TRANSFER\_TO\_HANDSET  CONN\_RULES\_LED\_ENABLE  CONN\_RULES\_LED\_DISABLE  CONN\_RULES\_DISCONNECT\_HANDSET  CONN\_RULES\_DISCONNECT\_PEER  CONN\_RULES\_HANDSET\_PAIR  CONN\_RULES\_ENTER\_DFU  CONN\_RULES\_ALLOW\_HANDSET\_CONNECT  CONN\_RULES\_REJECT\_HANDSET\_CONNECT  CONN\_RULES\_PAGE\_SCAN\_UPDATE  CONN\_RULES\_SEND\_PEER\_SCOFWD\_CONNECT  CONN\_RULES\_SELECT\_MIC  CONN\_RULES\_SCO\_FORWARDING\_CONTROL  CONN\_RULES\_ANC\_ENABLE  CONN\_RULES\_ANC\_DISABLE  CONN\_RULES\_ANC\_TUNING\_START  CONN\_RULES\_ANC\_TUNING\_STOP  CONN\_RULES\_BLE\_CONNECTION\_UPDATE  CONN\_RULES\_DFU\_ALLOW |
|  |
| PEER\_SIG\_PAIR\_HANDSET\_IND  PEER\_SIG\_CONNECT\_HANDSET\_IND  PEER\_SIG\_CONNECT\_HANDSET\_CFM  PEER\_SYNC\_STATUS |
|  |
| APP\_GAIA\_UPGRADE\_ACTIVITY  APP\_GAIA\_DISCONNECTED  APP\_GAIA\_UPGRADE\_CONNECTED  APP\_GAIA\_UPGRADE\_DISCONNECTED |
|  |
| APP\_UPGRADE\_REQUESTED\_TO\_CONFIRM  APP\_UPGRADE\_REQUESTED\_IN\_PROGRESS  APP\_UPGRADE\_ACTIVITY  APP\_UPGRADE\_STARTED  APP\_UPGRADE\_COMPLETED |
|  |
| APP\_GATT\_CONNECTION\_MADE  APP\_GATT\_CONNECTION\_DROPPED  APP\_GATT\_CONNECTABLE |
|  |
| SM\_INTERNAL\_PAIR\_HANDSET  SM\_INTERNAL\_DELETE\_HANDSETS  SM\_INTERNAL\_FACTORY\_RESET  SM\_INTERNAL\_TIMEOUT\_LINK\_DISCONNECTION  SM\_INTERNAL\_ENTER\_DFU\_UI  SM\_INTERNAL\_ENTER\_DFU\_UPGRADED  SM\_INTERNAL\_ENTER\_DFU\_STARTUP  SM\_INTERNAL\_TIMEOUT\_DFU\_ENTRY  SM\_INTERNAL\_TIMEOUT\_DFU\_MODE\_START  SM\_INTERNAL\_TIMEOUT\_DFU\_AWAIT\_DISCONNECT  SM\_INTERNAL\_NO\_DFU  SM\_INTERNAL\_TIMEOUT\_OUT\_OF\_EAR\_A2DP  SM\_INTERNAL\_TIMEOUT\_IN\_EAR\_A2DP\_START  SM\_INTERNAL\_TIMEOUT\_OUT\_OF\_EAR\_SCO  SM\_INTERNAL\_TIMEOUT\_IDLE  SM\_INTERNAL\_SEND\_PEER\_SYNC  SM\_INTERNAL\_REBOOT  SM\_INTERNAL\_LINK\_DISCONNECTION\_COMPLETE  SM\_INTERNAL\_BREDR\_CONNECTED |
|  |

## PhyState

|  |
| --- |
| PHY\_STATE\_INTERNAL\_IN\_CASE\_EVENT  PHY\_STATE\_INTERNAL\_OUT\_OF\_CASE\_EVENT  PHY\_STATE\_INTERNAL\_IN\_EAR\_EVENT  PHY\_STATE\_INTERNAL\_OUT\_OF\_EAR\_EVENT  PHY\_STATE\_INTERNAL\_MOTION  PHY\_STATE\_INTERNAL\_NOT\_IN\_MOTION  CHARGER\_MESSAGE\_ATTACHED  CHARGER\_MESSAGE\_DETACHED  ACCELEROMETER\_MESSAGE\_IN\_MOTION  ACCELEROMETER\_MESSAGE\_NOT\_IN\_MOTION  PROXIMITY\_MESSAGE\_IN\_PROXIMITY  PROXIMITY\_MESSAGE\_NOT\_IN\_PROXIMITY |

## GAIA

|  |
| --- |
| GAIA\_INIT\_CFM  GAIA\_CONNECT\_IND  GAIA\_DISCONNECT\_IND  GAIA\_DISCONNECT\_CFM  GAIA\_START\_SERVICE\_CFM  GAIA\_DEBUG\_MESSAGE\_IND  GAIA\_UNHANDLED\_COMMAND\_IND  GAIA\_SEND\_PACKET\_CFM  GAIA\_DFU\_CFM  GAIA\_DFU\_IND  GAIA\_UPGRADE\_CONNECT\_IND  GAIA\_UPGRADE\_DISCONNECT\_IND  GAIA\_CONNECT\_CFM  GAIA\_VA\_START\_CFM  GAIA\_VA\_DATA\_REQUEST\_IND  GAIA\_VA\_VOICE\_END\_CFM  GAIA\_VA\_VOICE\_END\_IND  GAIA\_VA\_ANSWER\_START\_IND  GAIA\_VA\_ANSWER\_END\_IND  GAIA\_VA\_CANCEL\_CFM  GAIA\_VA\_CANCEL\_IND  GAIA\_STAROT\_FRIST\_COMMAND\_IND  GAIA\_STAROT\_START\_SEND\_AUDIO\_IND  GAIA\_STAROT\_STOP\_SEND\_AUDIO\_IND  GAIA\_STAROT\_AUDIO\_INTERVAL |

## Battery

|  |
| --- |
| MESSAGE\_ADC\_RESULT  MESSAGE\_BATTERY\_TEST\_PROCESS\_READING  MESSAGE\_BATTERY\_INTERNAL\_MEASUREMENT\_TRIGGER |

## GATT

|  |
| --- |
| GATT\_EXCHANGE\_MTU\_IND  GATT\_MANAGER\_REGISTER\_WITH\_GATT\_CFM  GATT\_MANAGER\_REMOTE\_CLIENT\_CONNECT\_CFM  GATT\_MANAGER\_REMOTE\_CLIENT\_CONNECT\_IND  GATT\_MANAGER\_CANCEL\_REMOTE\_CLIENT\_CONNECT\_CFM  GATT\_MANAGER\_DISCONNECT\_IND  GATT\_MANAGER\_SERVER\_ACCESS\_IND  GATT\_GAP\_SERVER\_READ\_DEVICE\_NAME\_IND  GATT\_SERVER\_READ\_CLIENT\_CONFIG\_IND  GATT\_SERVER\_WRITE\_CLIENT\_CONFIG\_IND  GATT\_SERVER\_SERVICE\_CHANGED\_INDICATION\_CFM  APP\_ADVMGR\_ADVERT\_SET\_DATA\_CFM |

## ScoFwd

|  |
| --- |
| PEER\_SIG\_MSG\_CHANNEL\_RX\_IND  PEER\_SIG\_MSG\_CHANNEL\_TX\_CFM  PEER\_SIG\_CONNECTION\_IND  CL\_L2CAP\_REGISTER\_CFM  CL\_SDP\_REGISTER\_CFM  CL\_SDP\_SERVICE\_SEARCH\_ATTRIBUTE\_CFM  CL\_L2CAP\_CONNECT\_IND  CL\_L2CAP\_CONNECT\_CFM  CL\_L2CAP\_DISCONNECT\_IND  CL\_L2CAP\_DISCONNECT\_CFM  APP\_HFP\_CONNECTED\_IND  APP\_HFP\_DISCONNECTED\_IND  APP\_HFP\_SCO\_INCOMING\_RING\_IND  APP\_HFP\_SCO\_INCOMING\_ENDED\_IND  APP\_HFP\_VOLUME\_IND  MESSAGE\_MORE\_DATA  MESSAGE\_MORE\_SPACE  MESSAGE\_SOURCE\_EMPTY  SFWD\_INTERNAL\_LINK\_CONNECT\_REQ  SFWD\_INTERNAL\_LINK\_DISCONNECT\_REQ  SFWD\_INTERNAL\_START\_RX\_CHAIN  SFWD\_INTERNAL\_STOP\_RX\_CHAIN  SFWD\_INTERNAL\_KICK\_PROCESSING  SFWD\_INTERNAL\_RX\_AUDIO\_MISSING  SFWD\_INTERNAL\_ENABLE\_FORWARDING  SFWD\_INTERNAL\_DISABLE\_FORWARDING  SFWD\_INTERNAL\_ROLE\_NOTIFY  SFWD\_INTERNAL\_PLAY\_RING  SFWD\_INTERNAL\_MIC\_CHAIN\_DETAILS  SFWD\_TIMER\_LATE\_PACKET  PHY\_STATE\_CHANGED\_IND |

# 小知识

## 左右耳机区分

左右耳机，依据蓝牙地址的最后一位来判断。

>ADK 6.3.X Earbuds Application User Guide.pdf 2.1

>Left/Right Earbud role is selected by the least significant bit of the Earbud's Bluetooth address.

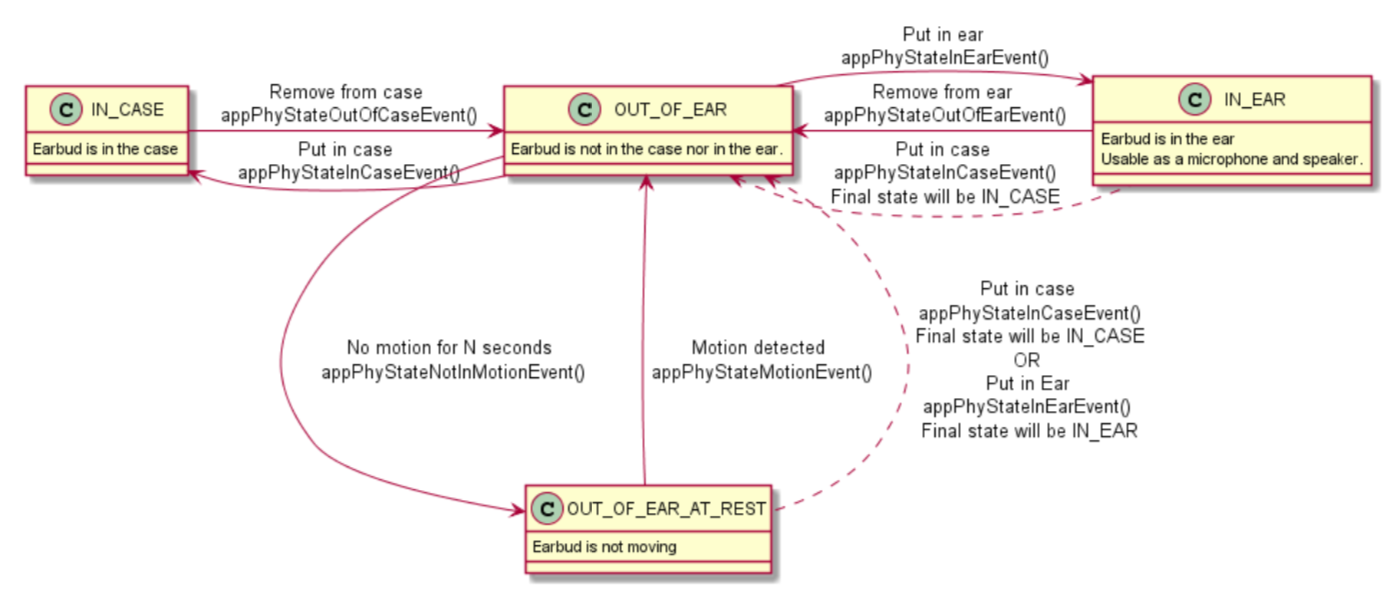
00025B00AF**BD**:1011 1101 左耳机

00025B00AF**BC**:1011 1100 右耳机

## Task

|  |  |
| --- | --- |
| Task名称 | 用途 |
| Sm | Main application state machine 状态机 |
| Ui | Manage the ui外围交互，处理按键、led灯等 |
| Kymera | Manages the audio chains管理音频 |

## 耳机-盒子状态



* IN\_CASE
* OUT\_OF\_CASE
* OUT\_OF\_EAR\_AT\_REST
* IN\_EAR

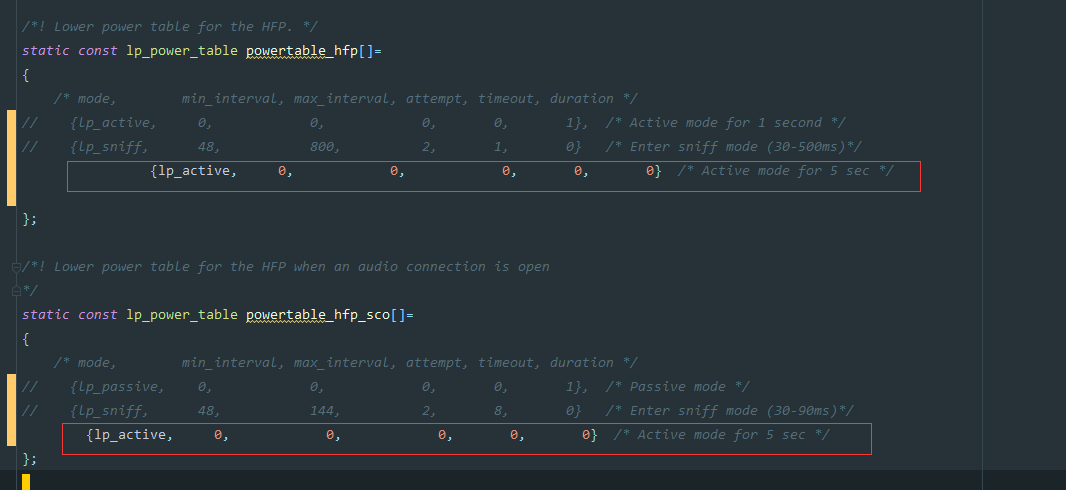
# Other

解决SPP慢的问题

CL\_RFCOMM\_SERVER\_CONNECT\_CFM

*if* (m->status == rfcomm\_connect\_success)  
{  
 msg->status = **SUCCESS**;  
 *static const* lp\_power\_table lp\_powertable\_data\_access[]={  
 */\* mode, min\_interval, max\_interval, attempt, timeout, duration \*/* {lp\_active, 0, 0, 0, 0, 0} */\* Go into active mode and stay there \*/* };  
 ConnectionSetLinkPolicy(m->sink, 1 ,lp\_powertable\_data\_access);  
}

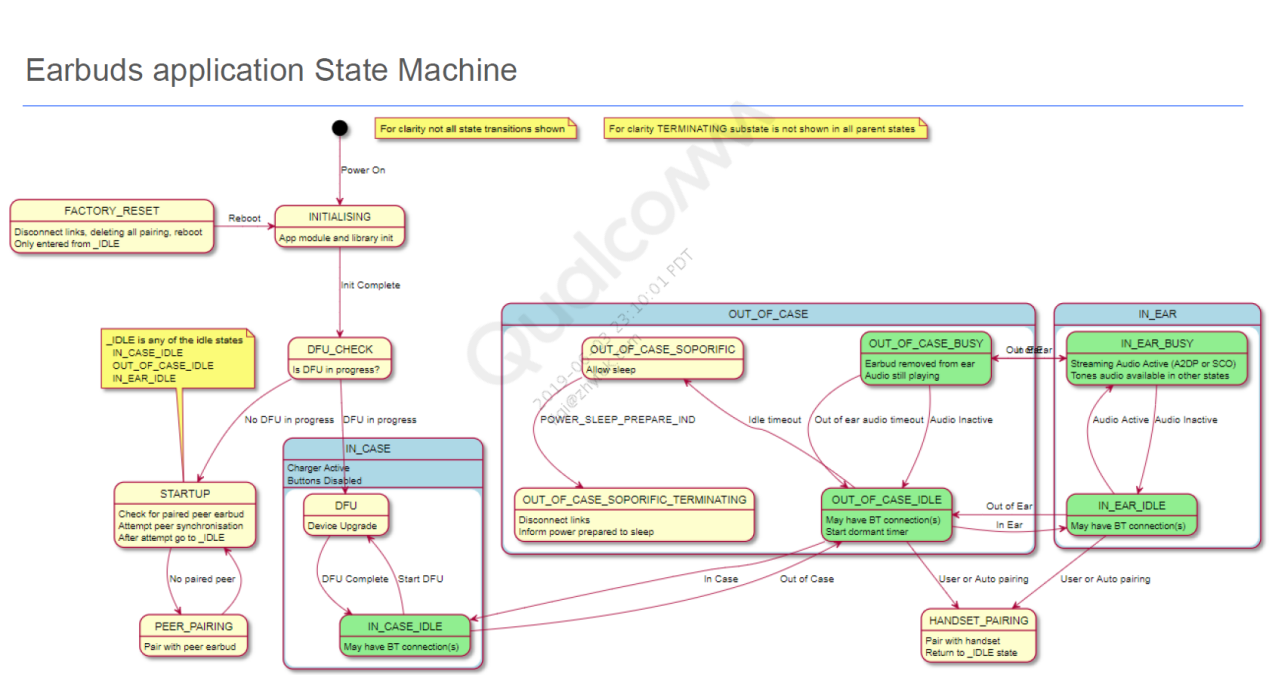
在通话时，gaia传输慢，修改功耗参数：



|  |
| --- |
| starot@raspberrypi:~ $ sdptool records 00:02:5B:00:FF:01  Service RecHandle: 0x10000  Service Class ID List:  "Audio Sink" (0x110b)  Protocol Descriptor List:  "L2CAP" (0x0100)  PSM: 25  "AVDTP" (0x0019)  uint16: 0x0103  Profile Descriptor List:  "Advanced Audio" (0x110d)  Version: 0x0103  Service RecHandle: 0x10001  Service Class ID List:  "Audio Source" (0x110a)  Protocol Descriptor List:  "L2CAP" (0x0100)  PSM: 25  "AVDTP" (0x0019)  uint16: 0x0103  Profile Descriptor List:  "Advanced Audio" (0x110d)  Version: 0x0103  Service RecHandle: 0x10002  Service Class ID List:  "AV Remote" (0x110e)  "AV Remote Controller" (0x110f)  Protocol Descriptor List:  "L2CAP" (0x0100)  PSM: 23  "AVCTP" (0x0017)  uint16: 0x0104  Profile Descriptor List:  "AV Remote" (0x110e)  Version: 0x0106  Service RecHandle: 0x10003  Service Class ID List:  "AV Remote Target" (0x110c)  Protocol Descriptor List:  "L2CAP" (0x0100)  PSM: 23  "AVCTP" (0x0017)  uint16: 0x0104  Profile Descriptor List:  "AV Remote" (0x110e)  Version: 0x0106  Service RecHandle: 0x10004  Service Class ID List:  UUID 128: 0000eb01-d102-11e1-9b23-00025b00a5a5  Profile Descriptor List:  "" (0x0000eb02-d102-11e1-9b23-00025b00a5a5)  Version: 0x0500  Service Name: Hands-Free unit  Service RecHandle: 0x10005  Service Class ID List:  "Handsfree" (0x111e)  "Generic Audio" (0x1203)  Protocol Descriptor List:  "L2CAP" (0x0100)  "RFCOMM" (0x0003)  Channel: 10  Language Base Attr List:  code\_ISO639: 0x656e  encoding: 0x6a  base\_offset: 0x100  Profile Descriptor List:  "Handsfree" (0x111e)  Version: 0x0107  Service RecHandle: 0x10006  Service Class ID List:  UUID 128: 0000eb03-d102-11e1-9b23-00025b00a5a5  Protocol Descriptor List:  "L2CAP" (0x0100)  PSM: 65279  Service Name: GAIA  Service RecHandle: 0x10007  Service Class ID List:  "Serial Port" (0x1101)  Protocol Descriptor List:  "L2CAP" (0x0100)  "RFCOMM" (0x0003)  Channel: 1  Language Base Attr List:  code\_ISO639: 0x656e  encoding: 0x6a  base\_offset: 0x100  Profile Descriptor List:  "Serial Port" (0x1101)  Version: 0x0102 |

# 规则引擎

通过规则(appConnRulesRunRules)，与SM配合。



## 状态机处理流程



## 规则状态



## 规则结果



## 耳机与盒子的规则



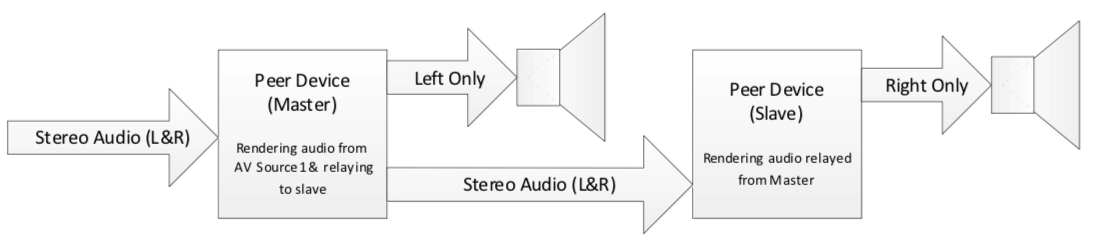
|  |  |
| --- | --- |
| IN-CASE   * 同步当前耳机到Peer上 * 与手机断开连接 * 与Peer断开连接 * 进入DFU模式 * 拒绝手机连接 * ANC处理 | OUT-CASE   * 同步当前耳机到Peer上 * 允许手机连接 * 处理与手机的配对信息 * 连接Peer * 连接手机 * ANC处理 |
| IN-EAR   * 同步当前耳机到Peer上 * LED禁用 * 开启SCO传输 * 选择MIC * ANC处理 | OUT-EAR   * 同步当前耳机到Peer上 * 定时断开A2DP * 定时断开SCO * 选择MIC * LED使能 * ANC处理 |

耳机位置与触发的规则：

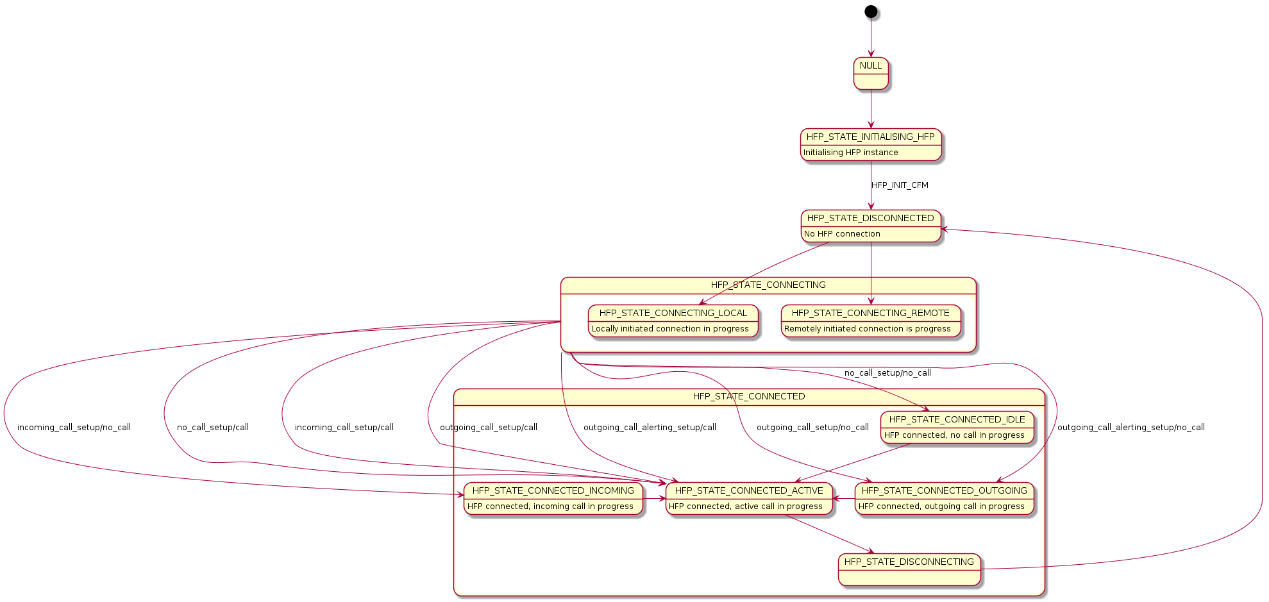
|  |  |
| --- | --- |
| **耳机位置** | **触发的规则** |
| **盒子中** | IN-CASE |
| **空中** | OUT-CASE  OUT-EAR |
| **耳朵中** | OUT-CASE  IN-EAR |

|  |  |
| --- | --- |
| **校验条件** | **RUN时发送的事件** |
| **rulePeerSync** | CONN\_RULES\_SEND\_PEER\_SYNC |
| **检测配对，如果符合条件，执行PeerSync操作，发送同步信息** | |
| **ruleInCaseDisconnectHandset** | CONN\_RULES\_DISCONNECT\_HANDSET |
| **与手机断开连接** | |
| **ruleInCaseDisconnectPeer** | CONN\_RULES\_DISCONNECT\_PEER |
| **与Peer耳机断开连接** | |
| **ruleInCaseEnterDfu** | CONN\_RULES\_ENTER\_DFU |
| **可以进入DFU模式** | |
| **ruleInCaseRejectHandsetConnect** | CONN\_RULES\_REJECT\_HANDSET\_CONNECT |
| **是否可以连接，** | |
| **ruleInCaseAncTuning** | CONN\_RULES\_ANC\_TUNING\_START |
| **调整ANC** | |
| **ruleOutOfCaseAllowHandsetConnect** | CONN\_RULES\_ALLOW\_HANDSET\_CONNECT |
| **允许耳机可以被连接** | |
| **ruleAutoHandsetPair** | CONN\_RULES\_HANDSET\_PAIR |
| **校验是否需要配对** | |
| **ruleOutOfCaseConnectPeer** | CONN\_RULES\_CONNECT\_PEER |
| **检测是否需要连接Peer** | |
| **ruleOutOfCaseConnectHandset** | CONN\_RULES\_CONNECT\_HANDSET |
| **是否可以连接手机** | |
| **ruleOutOfCaseAncTuning** | CONN\_RULES\_ANC\_TUNING\_STOP |
| **调整ANC** | |
| **ruleOutOfEarA2dpActive** | CONN\_RULES\_A2DP\_TIMEOUT |
| **是否在做A2DP，如果正在A2DP，定时停止** | |
| **ruleOutOfEarScoActive** | CONN\_RULES\_SCO\_TIMEOUT |
| **是否在做SCO，如果正在SCO，定时停止** | |
| **ruleSelectMicrophone** | CONN\_RULES\_SELECT\_MIC |
| **选择MIC** | |
| **ruleOutOfEarLedsEnable** | CONN\_RULES\_LED\_ENABLE |
| **调整LED灯** | |
| **ruleOutOfEarAncDisable** | CONN\_RULES\_ANC\_DISABLE |
| **调整ANC** | |
| **ruleInEarLedsDisable** | CONN\_RULES\_LED\_DISABLE |
| **关闭LED** | |
| **ruleInEarScoTransferToEarbud** | CONN\_RULES\_SCO\_TRANSFER\_TO\_EARBUD |
| **是否是Peer耳机，如果有，开启SCO** | |
| **ruleInEarAncEnable** | CONN\_RULES\_ANC\_ENABLE |
| **ANC使能** | |

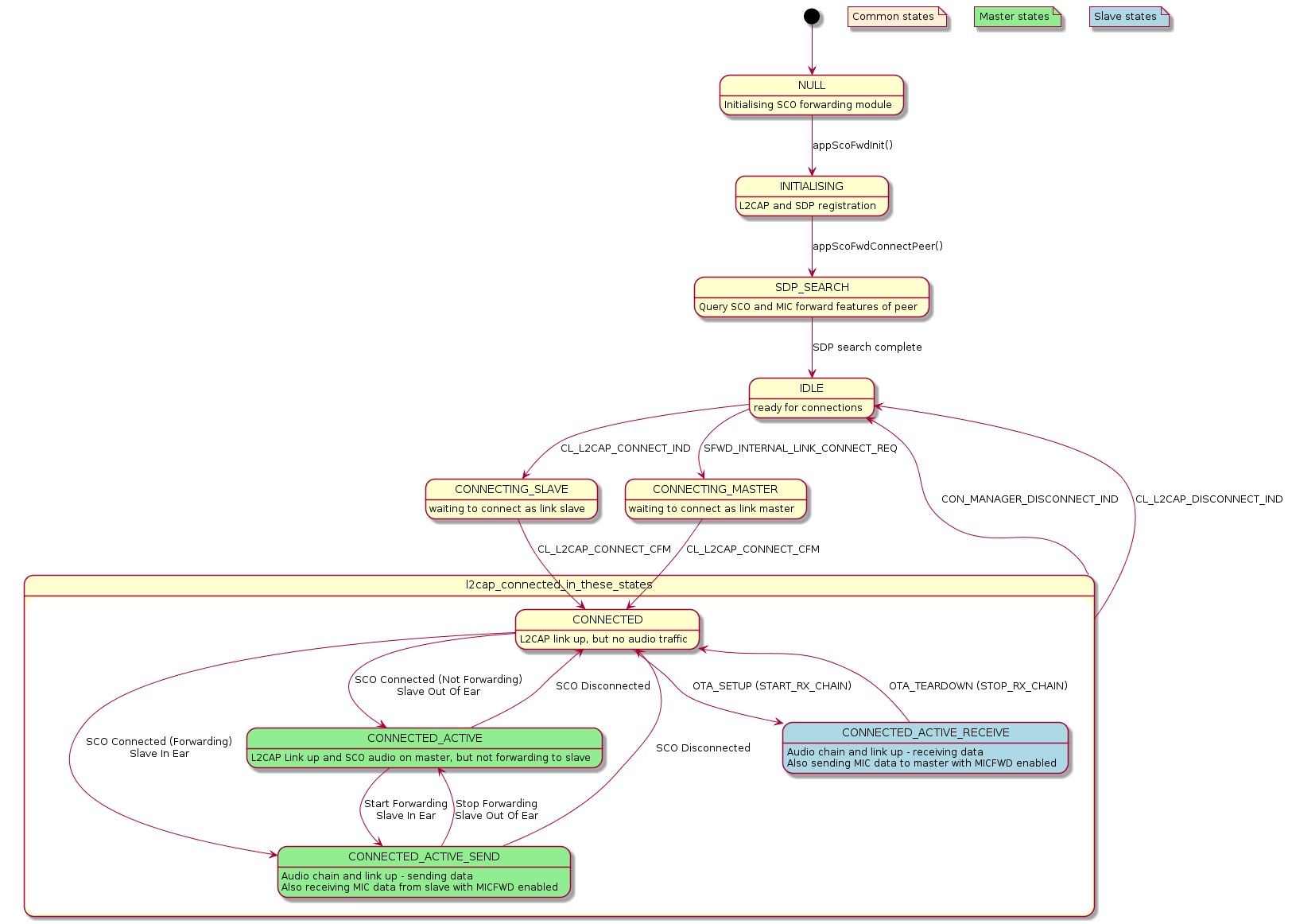
# [HFP-SCO/A2DP]-FWD

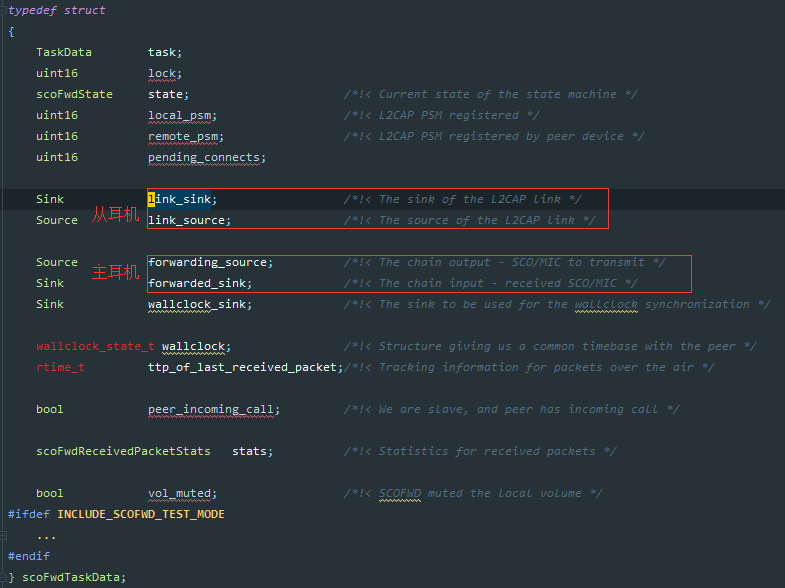


## HFP状态

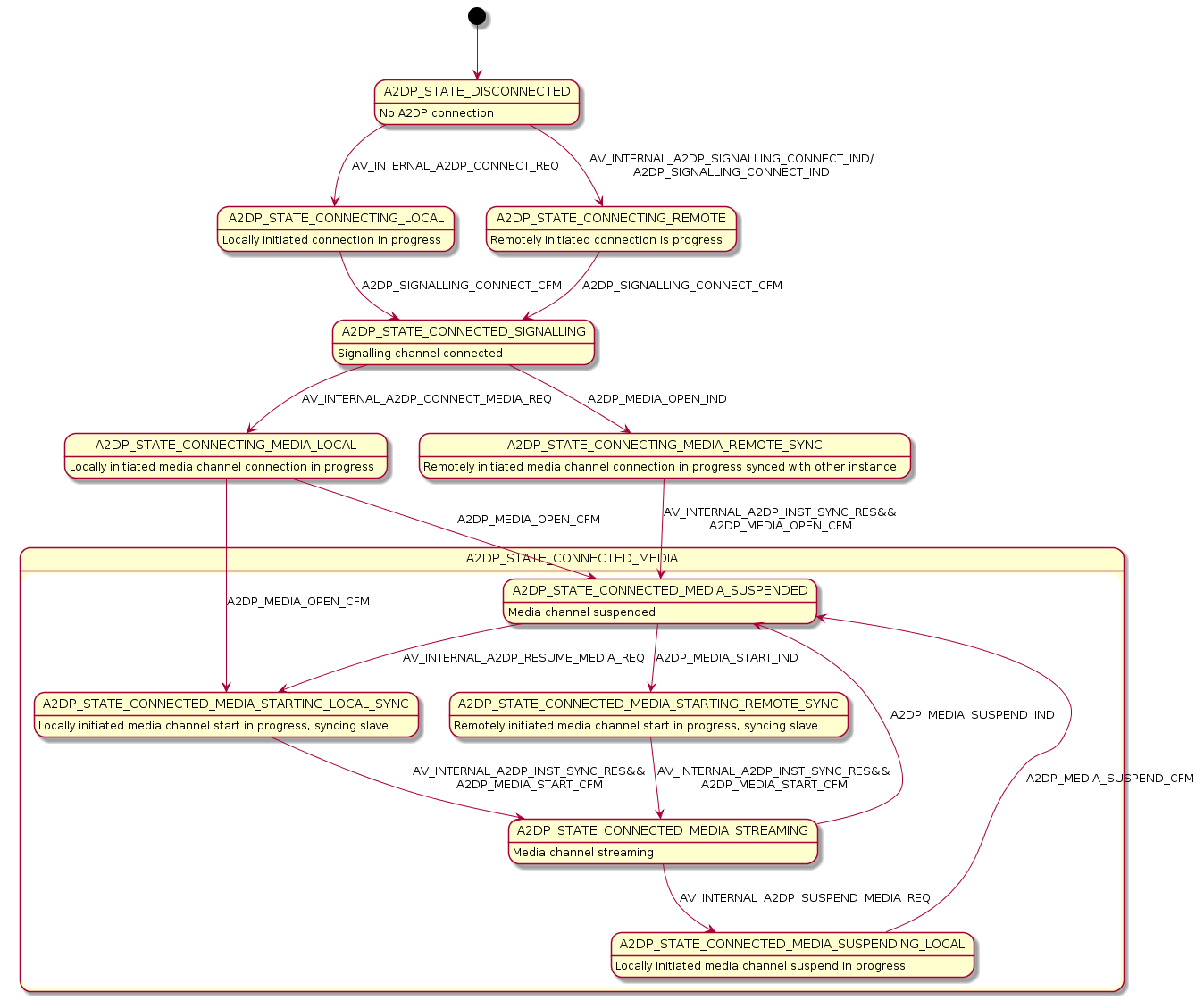


## ScoFwd状态





## A2DP状态



# 参考

* 《80\_CF848\_1\_AC\_ADK6\_Earbud\_Application.pdf》：介绍耳机的使用与开发。
* 《ADK 6.3.X Earbuds Application User Guide.pdf》
* 《ADK Audio Sing Application Peer Device User Guide.pdf》

# Test

ruleConnectHandsetStandard

ruleConnectBatteryVoltage