TWS无线耳机

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档名称 |  | 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 | 12 |

目录

[1 文档版本 3](#_Toc14091975)

[2 核心Task 3](#_Toc14091976)

[3 初始化 7](#_Toc14091977)

[4 耳机位置 9](#_Toc14091978)

[4.1 In case 10](#_Toc14091979)

[4.2 Out of case 10](#_Toc14091980)

[4.3 In Ear 10](#_Toc14091981)

[4.4 Out of Ear 10](#_Toc14091982)

[5 两只耳机的连接 11](#_Toc14091983)

[5.1 对等设备角色（Peer Device Roles） 11](#_Toc14091984)

[5.2 升级 12](#_Toc14091985)

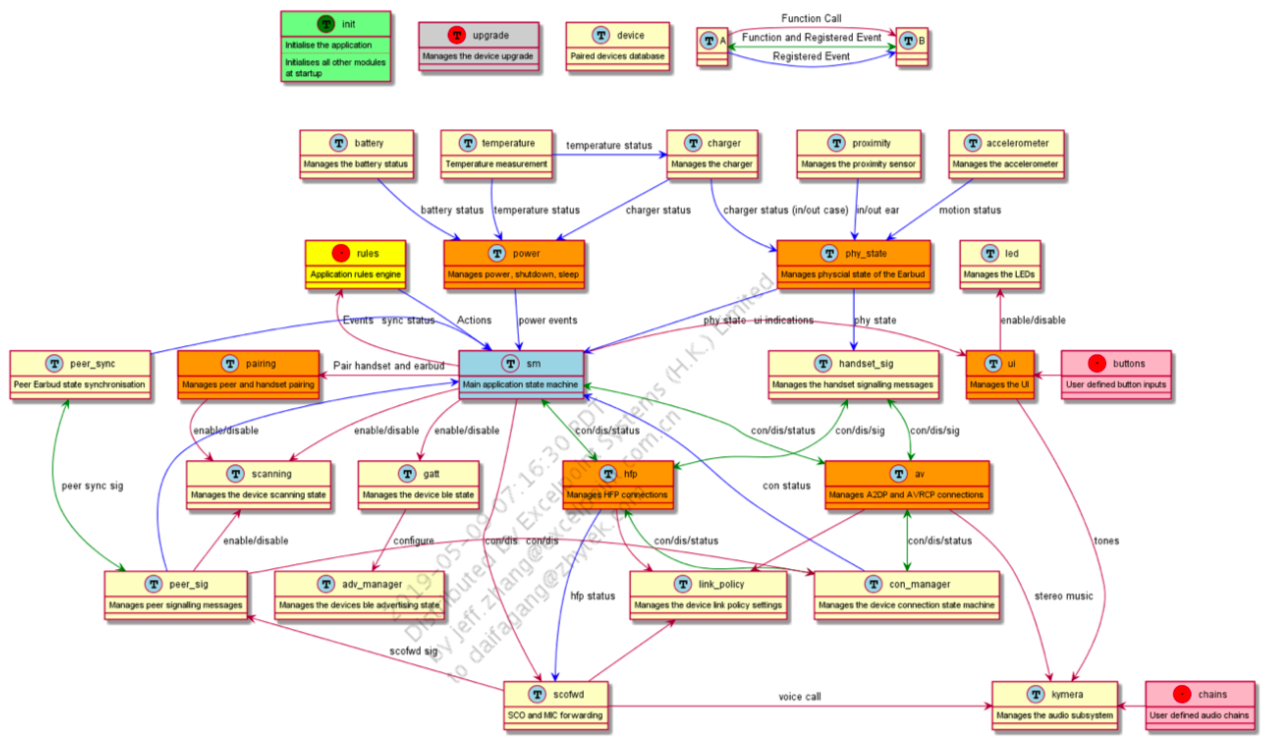
[6 参考文档 12](#_Toc14091986)

[7 目标 12](#_Toc14091987)

# 文档版本

|  |  |  |
| --- | --- | --- |
| 日期 | 版本号 | 简要描述 |
| 2019/7/10 | 1.00.00 | 初稿 |
| 2019/7/15 | 1.00.01 | 添加Peer模式的分析 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 核心Task



|  |  |
| --- | --- |
| PhyState | 耳机状态 |
| This task manages the devices current physical state based on inputs from the charger, proximity sensor, and accelerometer. | |
| 硬件状态：耳机是否在盒子/耳朵的位置。  并且负责充电、接近传感器、加速度计。 | |
|  | |
| Accelerometer | 加速度计 |
| Manages the interface to a supported accelerometer to detect if the device is in motion or not. This task enables and configures the accelerometer as required. The application currently supports the adxl362 accelerometer, which is connected over an SPI using the bitserial hardware interface.  NOTE The CF376 development hardware does not have an accelerometer. | |
|  | |
| Proximity | 接近传感器 |
| This task manages the interface to a supported proximity sensor to detect if the device is in ear or not. It enables and configures the proximity sensor as required. The application currently supports the vncl3020 proximity sensor connected over an I2C using the bitserial hardware interface.  NOTE The CF376 development hardware does not have a proximity sensor. scan\_manager  This task manages enable page and inquiry scanning on the device including the specific page and inquiry scan parameters. | |
|  | |
| Charger | 充电 |
| This task manages the on-chip charger to report the current charging status to the phy\_state task and to the battery task.  NOTE The charging temperature limit configurations:  ■ appConfigBatteryChargingTemperatureMax  ■ appConfigBatteryChargingTemperatureMin  ■ appConfigBatteryDischargingTemperatureMax  ■ appConfigBatteryDischargingTemperatureMin  should be set according to the values defined in the associated battery datasheet. | |
|  | |
| SDP |  |
| This module defines and manages the TWS service record. It also includes helper functions to support TWS based service and attribute searches. | |
|  | |
| AV | A2dp/avrcp音频与控制 |
| Manages both the AVRCP and A2DP connections to remote devices. This task registers with the underlying A2DP and AVRCP libraries from which it receives indications based on the A2DP and AVRCP profiles.  A separate task is created to handle A2DP and AVRCP connections for each unique device. It supports up to two connected devices each with a single A2DP and AVRCP connection. It also interfaces the kymera task to manage the audio chains as required. | |
|  | |
| AdvManager | BLE |
| Manages the Bluetooth Low Energy advertising data including the advertising parameters such as the advertising type and interval. | |
|  | |
| Battery | 电池 |
| This task manages the reading and reporting of the current battery status. | |
|  | |
| ConManager | 连接管理 |
| This task manages the ACL level connections to the device and informs any registered tasks of any such connection changes. It also initiates and manages SDP search requests within the application after a successful ACL connection has been created. | |
|  | |
| Device | 设备当前状态 |
| This task manages the information regarding the current state of the device. This includes information regarding the device’s current paired/connected peer device and any currently connected handset. It is used by the AV and HFP application tasks to retrieve information on the current device state. | |
|  | |
| GATT | BLE |
| This task manages the Bluetooth Low Energy GATT, GATT Service, and GAP Server. It manages the overall Bluetooth Low Energy connections including the Bluetooth Low Energy GATT Battery Service. It uses the adv\_manger to enable and disable the required Bluetooth Low Energy advertising information. | |
|  | |
| Upgrade | 升级 |
| This task manages the over the air upgrade functionality. It uses the upgrade and GAIA libraries to perform the device upgrade over a BR/EDR. | |
|  | |
| UI |  |
| This task manages the user input from the xml defined button configuration to control the device. It also manages the led task and plays tones as required to indicate the current state of the device or specific events. | |
|  | |
| LED |  |
| This task manages the led status of the device as requested by the applications state machine. | |
|  | |
| HFP |  |
| This task manages the HFP profile connection to the handset. It registers with the underlying HFP library from which it receives indications based on the HFP profile. It also manages the current HFP call status and interfaces the kymera task to manage the audio chains as required. | |
|  | |
| LinkPolicyManager |  |
| This task manages the link policy settings of the device. It manages when the device enters sniff and active modes based on the defined power table settings for various device modes. It also attempts to manage the device role to try and avoid scatternets. | |
|  | |
| Kymera |  |
| This task manages the audio chains defined within the predefined xml chain descriptions. It is responsible for loading the DSP audio chains and configuring any operators as required. It is also responsible for configuring the volume, starting and stopping audio and playing any requested tones. | |
|  | |
| Temperature | 温度 |
| This task manages the interfaced to a supported temperature reading device to detect changes in temperate and notify tasks that are interested in device temperature. | |
|  | |
| PeerSignalling | 两个耳机之间的配对 |
| This task manages any signaling commands over vendor AVRCP command between the device and the connected peer device. This includes managing the connection and disconnection of the peer signaling channel along with any specific peer signaling commands. | |
|  | |
| Pairing |  |
| This task manages the peer device and handset pairing state. Additionally, it is responsible for setting any extended inquire response data used during the pairing discovery process. | |
|  | |
| PeerSync | 两个耳机之间的信息同步 |
|  | |
|  | |
| StateMachine |  |
| This task manages the overall device state machine based on event and indications from various other tasks. It interfaces with the rules engine that determines actions to take based on the current state of the device and new events received. | |
|  | |
| HandsetSignalling |  |
| This task manages any signaling over AVRCP or HFP between the device and the connected handset. This includes both profile-specific commands and Qualcomm TrueWirelessTM Stereo Plus (TWS+) specific commands between the device and the handset. | |
|  | |
| ScoFwd |  |
| This module manages the SCO and MIC forwarding state machine and control if enabled when connected to a Qualcomm TrueWirelessTM Stereo (TWS) Standard handset. | |
|  | |
| Init |  |
| This module manages the initialization of all the other modules and tasks within the application. | |
|  | |
| TaskList |  |
| This module manages a list of VM tasks. It is used by other tasks to manage their list of registered tasks. | |
|  | |
| Application |  |
|  | |
|  | |
| Sys |  |
|  | |
|  | |
| Codec |  |
|  | |
|  | |
| ScanManager |  |
|  | |
|  | |
| ConnectionRules |  |
|  | |
|  | |
| Gaia |  |
|  | |
|  | |
| Power | 电量 |
|  | |
|  | |
| AncTuning |  |
|  | |
|  | |

# 初始化

|  |
| --- |
| /\*! \brief Table of initialisation functions \*/  static const appInitTableEntry appInitTable[] = {  #ifdef INIT\_DEBUG  {appInitDebug, 0, NULL},  #endif  {appPioInit, 0, NULL},  {appUiInit, 0, NULL},  {appLicenseCheck, 0, NULL},  #ifdef INCLUDE\_TEMPERATURE  {appTemperatureInit, TEMPERATURE\_INIT\_CFM, NULL},  #endif  {appBatteryInit, MESSAGE\_BATTERY\_INIT\_CFM, NULL},  #ifdef INCLUDE\_CHARGER  {appChargerInit, 0, NULL},  #endif  {appLedInit, 0, NULL},  {appPowerInit, APP\_POWER\_INIT\_CFM, NULL},  {appPhyStateInit, PHY\_STATE\_INIT\_CFM, NULL},  {appConnectionInit, CL\_INIT\_CFM, appInitHandleClInitCfm},  #ifdef USE\_BDADDR\_FOR\_LEFT\_RIGHT  {appConfigInit, CL\_DM\_LOCAL\_BD\_ADDR\_CFM, appInitHandleClDmLocalBdAddrCfm},  #endif  {appLinkPolicyInit, 0, NULL},  {appConManagerInit, 0, NULL},  {appConnRulesInit, 0, NULL},  {appDeviceInit, 0, NULL},  {appScanManagerInit, 0, NULL},  {appAdvManagerInit, APP\_ADVMGR\_INIT\_CFM, NULL},  {appAvInit, AV\_INIT\_CFM, NULL},  {appPeerSigInit, 0, NULL},  {appPairingInit, PAIRING\_INIT\_CFM, NULL},  {appPeerSyncInit, 0, NULL},  {appHfpInit, APP\_HFP\_INIT\_CFM, NULL},  {appHandsetSigInit, 0, NULL},  {appKymeraInit, 0, NULL},  {appSmInit, 0, NULL},  {appScoFwdInit, SFWD\_INIT\_CFM, NULL},  {TransportMgrInit, 0, NULL},  #ifdef INCLUDE\_DFU  {appGaiaInit, APP\_GAIA\_INIT\_CFM, NULL}, // Gatt needs GAIA  #endif  #ifdef INCLUDE\_GATT  {appGattInit, APP\_GATT\_INIT\_CFM, appInitHandleGattInitCfm},  #endif  #ifdef INCLUDE\_DFU  {appUpgradeInit, UPGRADE\_INIT\_CFM, NULL}, // Upgrade wants to start a connection (can be gatt)  #endif  {NULL, 0, NULL}  }; |

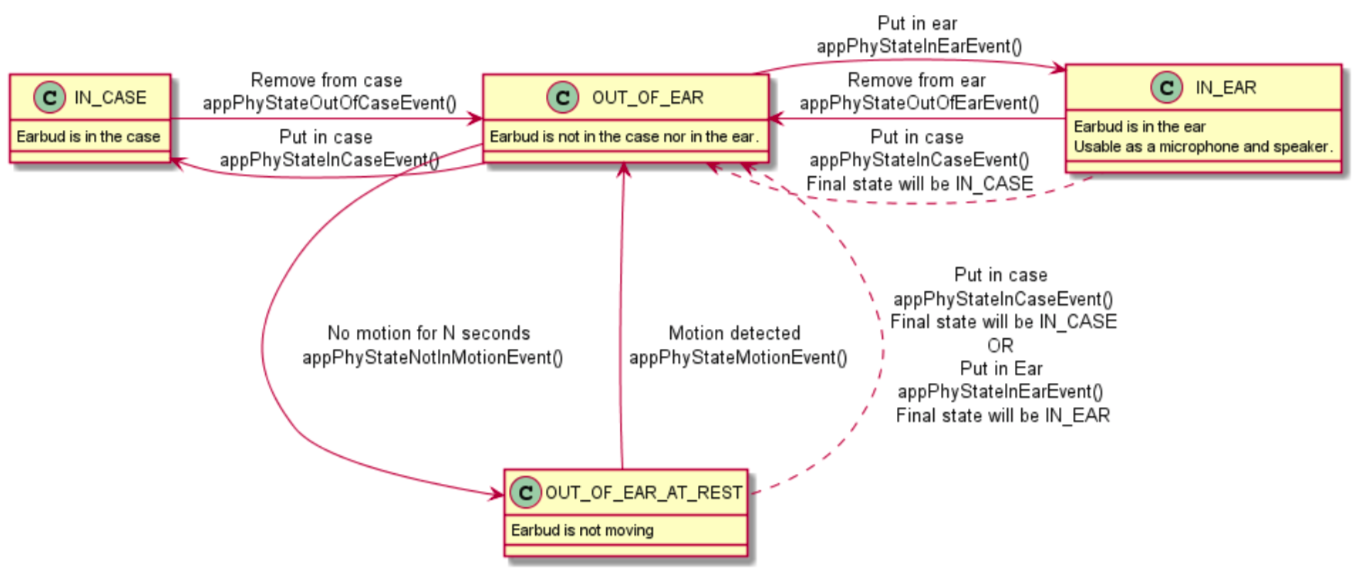
Main在初始化的时候，挨个调用初始化函数，如果需要等待指定的信号，需要等待收到了该信号之后，才能继续执行初始化。如果在等待的信号中还有handler，也需要调用。

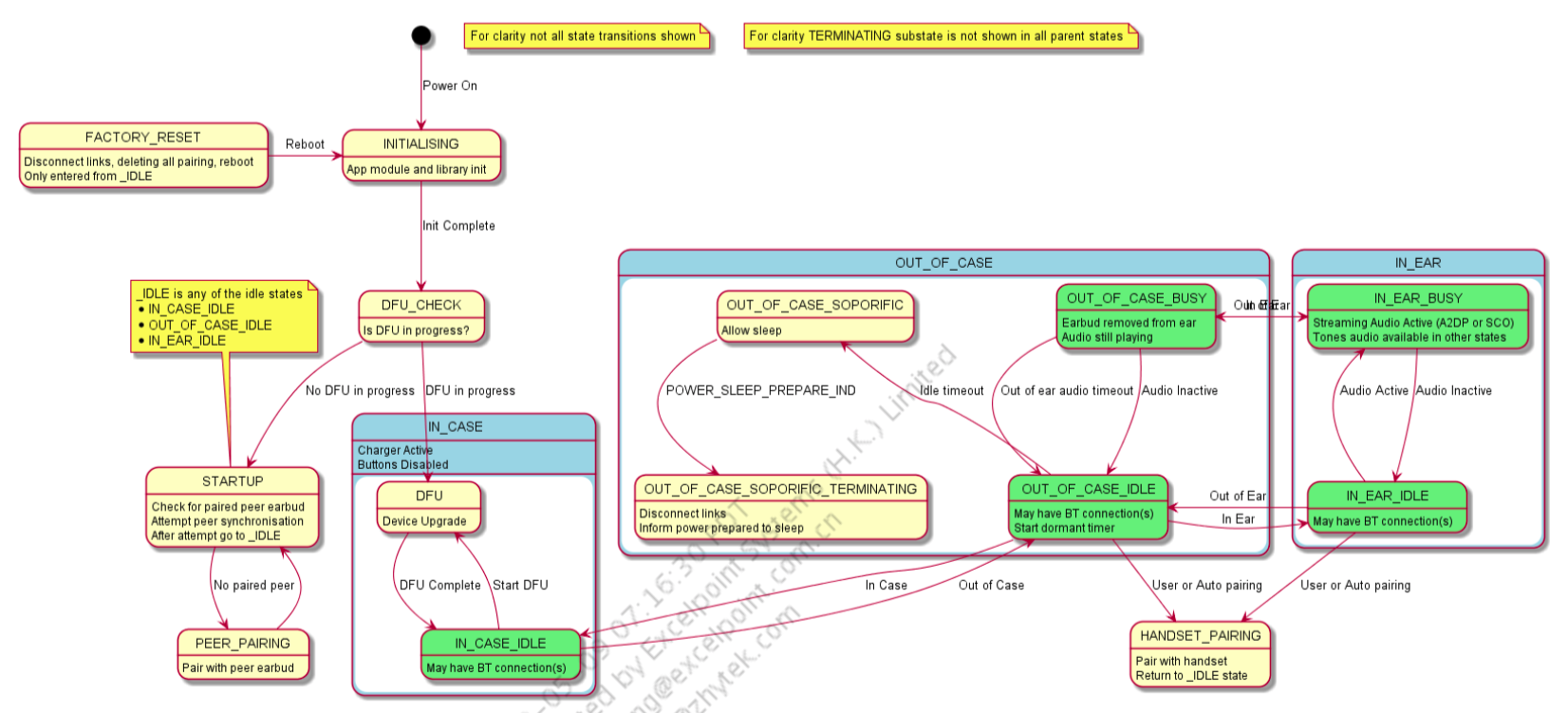
|  |
| --- |
| static void appHandleMessage(Task task, MessageId id, Message message) {  UNUSED(task);  if (id == appInitWaitingForMessageId()) {  appInitHandleMessage(id, message);  return;  }  …………………  } |

例如在初始化的时候，执行到了appTemperatureInit，之后，发现需要等待TEMPERATURE\_INIT\_CFM信号，停止初始化，等待该信号；此信号来了之后，继续执行初始化任务。

各任务定义：<ADK 6.3.X Earbuds Application User Guide.pdf> 4.4 Application module

# 耳机位置





## In case

When an earbud is in the case all button inputs are disabled. Additionally, when an earbud is put into the case it disconnects all links to a connected handset.

## Out of case

When an earbud is taken out of the case if it has not handset pairing it enters Handset Pairing mode automatically. If the earbud does have a paired handset, it attempts to reconnect back to the last connected handset.

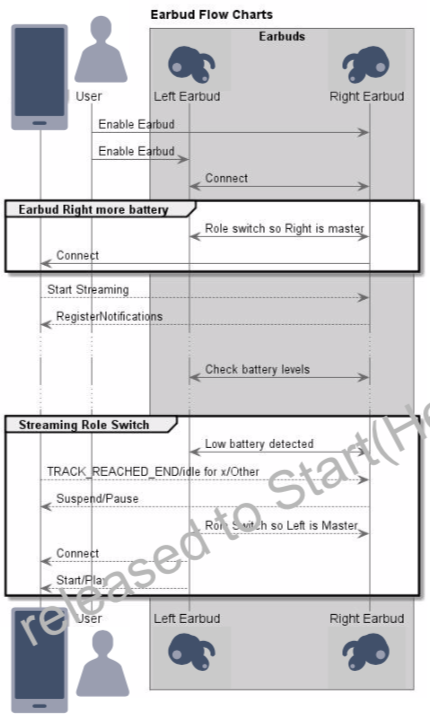
## In Ear

When an earbud is place in the ear, it disables the LEDs. If A2DP streaming is paused due to an earbud being taken out of the ear and the earbud is put back in the ear within 10 seconds it automatically attempts to resume the audio. If it is put back after 10 seconds it does not resume the audio and the user needs to press the play/pause button or restart the audio from the phone.

## Out of Ear

When an earbud is taken out of the ear during A2DP streaming it attempts to pause the music after 2 seconds. When an Earbud is taken out of the ear with an active SCO connection it attempts to transfer the call to the headset after 2 seconds.

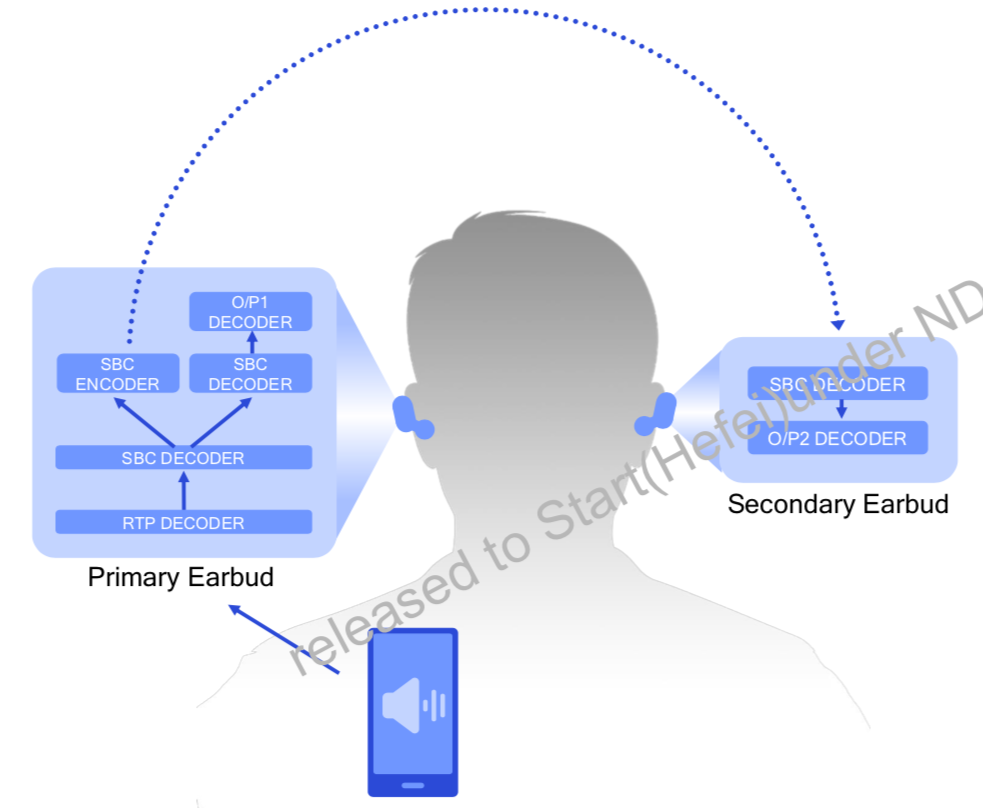
# 两只耳机的连接



## 对等设备角色（Peer Device Roles）

Each peer device has one of two roles:

* Peer Master: The Peer Master is the Peer Device whose audio source is relayed.
* Peer Slave: The Peer Slave is the Peer Device that receives relayed audio from a Peer Master.



正常佩戴的时候，传输音频、音量等控制信息，基于L2CAP链路的封装。

* 谁先使能谁是主；
* 同时使能，谁电量多，谁是主。

av\_headset\_peer\_signalling.h 对等连接建立

av\_headset\_conn\_rules.h 角色选择

## PeerConnect

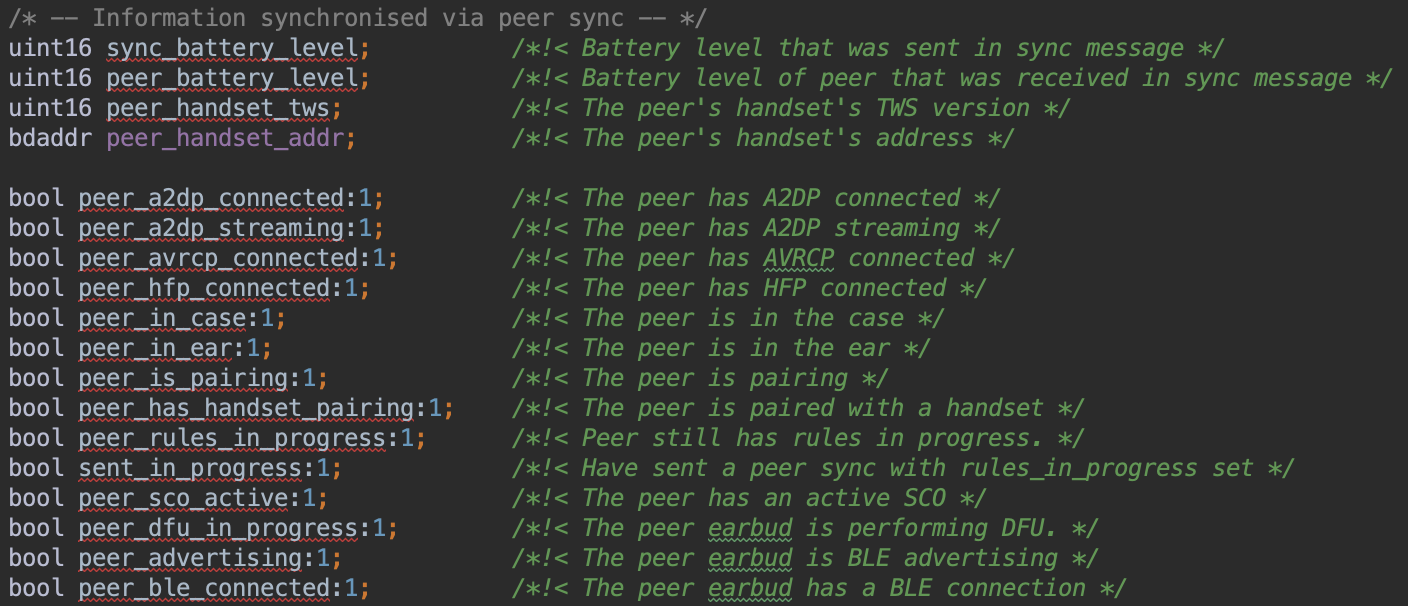
{appPeerSigInit, 0, NULL}

{appPairingInit, PAIRING\_INIT\_CFM, NULL}

{appPeerSyncInit, 0, NULL}

|  |
| --- |
| NULL -down-> INITIALISING : appPairingInit()  INITIALISING : Registering SDP record  INITIALISING -down-> IDLE : CL\_SDP\_REGISTER\_CFM  IDLE : Page and Inquiry scan disabled  IDLE -up-> CHECKING\_HANDSET\_LINKKEYS : CHECK\_HANDSET\_LINKKEYS  CHECKING\_HANDSET\_LINKKEYS : Derive handset link keys and send to peer.  CHECKING\_HANDSET\_LINKKEYS -up-> IDLE : Check Complete  state LeftEarbudPeerPairing {  LeftEarbudPeerPairing : Page scan enabled  state CancellableLeftPairing {  IDLE -left-> PEER\_INQUIRY : PEER\_PAIR\_REQ (LEFT)  PEER\_INQUIRY : Inquiry Enabled  PEER\_INQUIRY : Searching for right earbud  PEER\_INQUIRY -down-> PEER\_SDP\_SEARCH : Found peer perform SDP search  PEER\_INQUIRY -right-> IDLE : PEER\_PAIR\_CFM(timeout)  PEER\_SDP\_SEARCH : Find peer TWS+ version  PEER\_SDP\_SEARCH -right-> IDLE : PEER\_PAIR\_CFM(SDP fail)  }  CancellableLeftPairing -right-> IDLE : PEER\_PAIR\_CFM(cancelled)  PEER\_SDP\_SEARCH -down-> PEER\_AUTHENTICATE : SDP search ok start auth  PEER\_AUTHENTICATE -left-> IDLE : PEER\_PAIR\_CFM(success/fail)  PEER\_AUTHENTICATE : Pairing in progress  }  state RightEarbudPeerPairing {  RightEarbudPeerPairing : Page scan enabled  IDLE -right-> PEER\_DISCOVERABLE : PEER\_PAIR\_REQ (RIGHT)  PEER\_DISCOVERABLE : Inquiry scan enabled  PEER\_DISCOVERABLE : Awaiting left earbud connection  PEER\_DISCOVERABLE -left-> IDLE : PEER\_PAIR\_CFM(timeout/cancelled)  PEER\_DISCOVERABLE -down-> PEER\_PENDING\_AUTHENTICATION : Start auth  PEER\_PENDING\_AUTHENTICATION -down-> PEER\_SDP\_SEARCH\_AUTHENTICATED : Auth Success, SDP search  PEER\_PENDING\_AUTHENTICATION : Pairing in progress  PEER\_PENDING\_AUTHENTICATION --> IDLE : PEER\_PAIR\_CFM(failed)  PEER\_SDP\_SEARCH\_AUTHENTICATED --> IDLE : PEER\_PAIR\_CFM(success/failed)  PEER\_SDP\_SEARCH\_AUTHENTICATED : Find peer TWS+ version  }  state HandsetPairing {  HandsetPairing : Page scan enabled  IDLE -down-> HANDSET\_DISCOVERABLE : HANDSET\_PAIR\_REQ  IDLE -down-> HANDSET\_PENDING\_AUTHENTICATION : HANDSET\_PAIR\_REQ(known addr)  HANDSET\_DISCOVERABLE : Inquiry scan enabled  HANDSET\_DISCOVERABLE : Awaiting handset connection  HANDSET\_DISCOVERABLE -up-> IDLE : HANDSET\_PAIR\_CFM(timeout/cancelled)  HANDSET\_DISCOVERABLE -down-> HANDSET\_PENDING\_AUTHENTICATION : Start authentication  HANDSET\_PENDING\_AUTHENTICATION -down-> HANDSET\_SDP\_SEARCH\_AUTHENTICATED : Auth Success, SDP search  HANDSET\_PENDING\_AUTHENTICATION : Pairing in progress  HANDSET\_PENDING\_AUTHENTICATION --> IDLE : HANDSET\_PAIR\_CFM(failed)  HANDSET\_SDP\_SEARCH\_AUTHENTICATED --> IDLE : HANDSET\_PAIR\_CFM(success/failed)  HANDSET\_SDP\_SEARCH\_AUTHENTICATED : Find peer TWS+ version  } |

## PeerSync



## 升级

代理商回复：独立升级。

详见升级分析文档。

# 参考文档

* ADK 6.3.x Earbuds Application User Guide.pdf

# 目标

获取peer的蓝牙地址

主从判断

对外地址是否独立，耳机是否会发生蓝牙地址变更，类似mac地址交换。

# 术语

|  |  |
| --- | --- |
| ACK | Acknowledgement |
| ADK | Audio or Application Development Kit |
| API | Application Programming Interface |
| Bluetooth | Set of technologies providing audio and data transfer over short-range radio connections Bluetooth SIG. The Bluetooth Special Interest Group oversees the development of Bluetooth standards and the licensing of Bluetooth technologies and trademarks to manufacturers. |
| GAIA | Generic Application Interface Architecture |
| ID | Identifier |
| LED | Light-Emitting Diode |
| PEQ | Parametric Equalizer |
| PIO | Programmable Input/Output |
| PS | Persistent Store |
| QTIL | Qualcomm Technologies International, Ltd. |
| RSSI | Received Signal Strength Indication |
| RWCP | Reliable Write Command Protocol |
| SDP | Service Discovery Protocol |
| SIG | Special Interest Group |
| SPP | Serial Port Profile |
| SQIF | Serial Quad I/O Flash, a non-volatile memory technology |
| TWS | TrueWireless Stereo |
| UART | Universal Asynchronous Receiver Transmitter |
| USB | Universal Serial Bus |
| USB-IF | The USB Implementers' Forum is responsible for issuing USB vendor IDs to product manufacturers. |
|  |  |
|  |  |