

# 团体标准

T/CAIACN 017—2025

## 平台与无线耳机的空间音频交互技术要求及评估方法

Technical Requirements and Test Methods of Spatial Audio  
Interactive Between Platform and Wireless Earphone

2025 - 11 - 05 发布

2026 - 01 - 05 实施

中国电子音响行业协会 发布



## 目 次

前言.....	III
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 空间音频交互概述.....	2
5 空间音频数据交互基本要求.....	2
6 空间音频播放流程.....	4
7 空间音频效果评估要求及测试方法.....	6
附录 A（资料性） 空间音频初始化流程接口参考示例.....	10
附录 B（资料性） 空间音频播放流程接口参考示例.....	13



## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

中国电子音响行业协会（China Audio Industry Association，简称CAIA）自1983年成立以来就以“服务企业，献策政府”为宗旨。是我国最早成立的跨地区、跨部门、跨系统，具有社团法人资格的全国性社会团体（国家一级行业协会）。

组织开展电子音响领域国际、国内标准化活动，制定中国电子音响行业协会团体标准（以下简称：中音协团标），满足行业需要，推动行业标准化工作，是中国电子音响行业协会的重要工作。协会的所有会员，均有权利提出制、修订中音协团标的建议并参与有关工作。

中电音协团标按《中国电子音响行业协会团体标准建设管理办法》进行制定和管理。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

在本标准实施过程中，如发现需要修改或补充之处，请将意见和有关资料报送中国电子音响行业协会，以便修订时参考。

本文件由腾讯音乐娱乐科技（深圳）有限公司、东莞市漫步者科技有限公司、深圳市三诺数字科技有限公司提出。

本文件由中国电子音响行业协会归口。

本文件起草单位：腾讯音乐娱乐科技（深圳）有限公司、东莞市漫步者科技有限公司、深圳市三诺数字科技有限公司、中国电子技术标准化研究院、深圳市云动创想科技有限公司、瀛通通讯股份有限公司、深圳市冠旭电子股份有限公司、万魔声学股份有限公司、广州酷狗计算机科技有限公司、深圳瑞利声学技术股份有限公司、广东阅真科技有限公司。

本文件主要起草人：周文江、曾泽雄、马晓明、卫文港、曾德钧、黄晖、彭久高、尹辉、杨再静、李德涛、蔡千寻、岳魏捷、罗彧成、曹硕、赵泽鹏、闫震海、刘馨璐、甘旭、杜承才、汪奕霖、李茹、宋哲超、乔峤、陈国平、邓森、曾义、王义多、贡维勇、漆原、邓伟、张洋，李胜波。

本文件为首次制定。



# 平台与无线耳机的空间音频交互技术要求及评估方法

## 1 范围

本文件规定了在音乐场景下平台与无线耳机的空间音频交互技术要求,包括空间音频交互逻辑架构、数据格式、播放流程以及空间音频的播放效果的技术要求。

本文件适用于空间音频交互流程的设计、开发以及评估。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 9002—2017 音频、视频和视听设备及系统词汇

T/CA 109—2019 蓝牙耳机技术要求

T/CADPA 26—2022 音乐平台术语

## 3 术语和定义

下列术语和定义适用于本文件。

### 3.1

**空间音频** spatial audio

在音频录制过程中,音频录制过程后,或音频重放时使用特定的技术和处理,以创造出具有立体、环绕或空间感效果的音频。

### 3.2

**音质** sound quality

经传输和处理后音频信号的保真度。

[来源: T/CADPA 26—2022]

### 3.3

**多声道** multi-channel

使用多个独立的音频通道来录制,传输和播放声音。多个独立的音频通道,可以确保声音信号在处理时互不干扰。

### 3.4

**跟踪运动** tracking motion

引起空间音频变化的头部运动,可以包含头部转动等运动。

### 3.5

#### 非跟踪运动 untracked motion

不会引起空间音频变化的头部运动。

## 4 空间音频交互概述

当前大多数移动智能终端系统已支持空间音频渲染功能，空间音频对于音乐和游戏来说，可以带来丰富的听觉效果，拓展了用户听觉的空间感体验。

由于无线耳机的品牌、硬件种类繁多，对于空间音频场景来说，要针对各硬件设备去开发适配对应的空间音频效果以及与硬件设备的通信，会造成极大的工作量。并且，由于平台（可以为APP）以及硬件的差异，很难实现统一的空间音频效果。

基于以上问题，本文件定义了平台与无线耳机的空间音频交互要求。通过定义统一的空间音频交互接口，在空间音频播放时可根据无线耳机回传的头部运动数据在平台渲染器中进行实时渲染，渲染器包含头相关传递函数算法（HRTF），如图1所示。

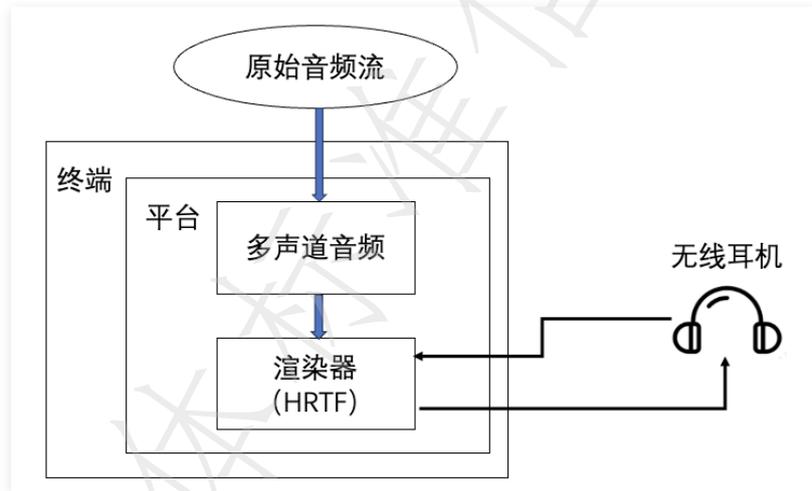


图 1 空间音频交互技术架构图

## 5 空间音频数据交互基本要求

### 5.1 设备基本要求

终端与无线耳机支持的多媒体格式宜满足以下要求：

- 支持多声道音频播放；
- 支持标准音视频编解码和传输；
- 支持蓝牙及wifi等标准传输协议。

在音乐播放模式下，无线耳机需保持与单个终端连接，确保音频输出独占一个无线耳机设备。

### 5.2 认证鉴权要求

事先分配鉴权信息，用于在通信过程中进行身份认证。其中鉴权信息包括：厂商ID，设备ID，及共享密钥等。

平台对无线耳机鉴权成功后，与无线耳机建立连接。

### 5.3 空间音频交互逻辑要求

**头部跟踪单元：**无线耳机包含6轴陀螺仪及加速度计等惯性传感器。这些惯性传感器跟踪无线耳机使用者的头部运动，记录头部运动数据。

**渲染器单元：**平台包含的HRTF计算模块及HRTF数据库，HRTF计算模块根据无线耳机反馈的头部运动数据，通过HRTF数据库并结合插值技术来生成合适的角度数据，再对多声道音频数据进行渲染生成空间音频数据。

空间音频交互逻辑如图2所示。

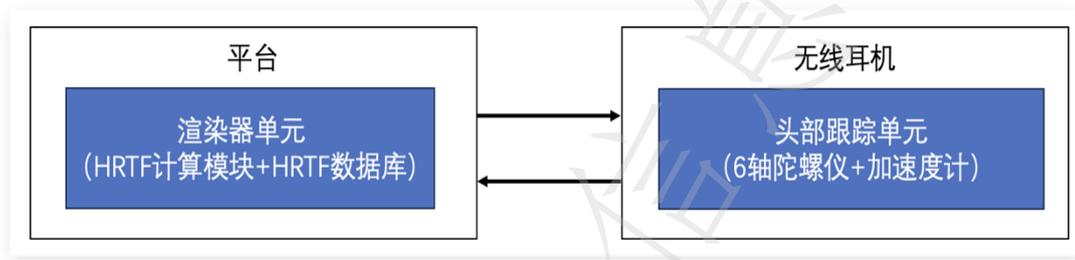


图 2 空间音频交互逻辑图

本文件中所涉及到使用的HRTF计算模块的计算方式如下：

$$H_L = H_L(r, \theta, \phi, f) = \frac{P_L(r, \theta, \phi, f)}{P_o(r, f)} \quad (1)$$

$$H_R = H_R(r, \theta, \phi, f) = \frac{P_R(r, \theta, \phi, f)}{P_o(r, f)} \quad (2)$$

式中：

$P_L$ 、 $P_R$  —— 声源在左、右耳产生的频域复数声压；

$P_o$  —— 头部移开后声源在原头部中心位置处的频域复数声压；

$(r, \theta, \phi)$  —— 声源的空间位置， $r$  表示距离， $\theta$  和  $\phi$  分别表示水平和垂直的方位角度；

$f$  —— 声音频率。

### 5.4 头部运动数据结构要求

#### 5.4.1 原始数据结构要求

原始数据结构，如表1所示。

表 1 原始数据结构

	说明	必选/可选
Sensitive	int, 加速度传感器灵敏度 (单位: $rad/s^2$ )	必选
ZR	int, 陀螺仪零偏置值 (单位: LSB)	必选
Rx	int, X轴的转角加速度数据, 单位LSB	必选

表 1 原始数据结构（续）

	说明	必选/可选
Ry	int, Y轴的转角加速度数据（单位：LSB）	必选
Sensitive_LSB	int, 陀螺仪传感器灵敏度（单位：LSB/rad/s）	必选
interval	int, 数据更新的时间间隔（单位：ms）	必选
Rz	int, Z轴的转角加速度数据（单位：LSB）	可选
Mag[n]	int数组, 具有磁场辅助信息的n轴陀螺仪产生的n个维度的磁场辅助加速度值（单位： $^{\circ}/s^2$ ）	可选
threshold	int, 控制加速度（单位： $rad/s^2$ ）	可选

#### 5.4.2 欧拉角数据结构要求

空间音频欧拉角计算方式如下：

$$R = R_z(\psi) \times R_y(\theta) \times R_x(\varphi) \quad (3)$$

式中：

$R$  ——三维数组，由俯仰角、偏航角、滚转角三个数据构成，分别对应了绕 X、Y、Z 轴的旋转角度（单位： $^{\circ}$ ）；

$R_z(\psi)$ ——绕 Z 轴旋转 $\psi$ 角度的旋转矩阵；

$R_y(\theta)$ ——绕 Y 轴旋转 $\theta$ 角度的旋转矩阵；

$R_x(\varphi)$ ——绕 X 轴旋转 $\varphi$ 角度的旋转矩阵。

## 6 空间音频播放流程

### 6.1 设备初始化流程

设备初始化流程如图 3 所示。

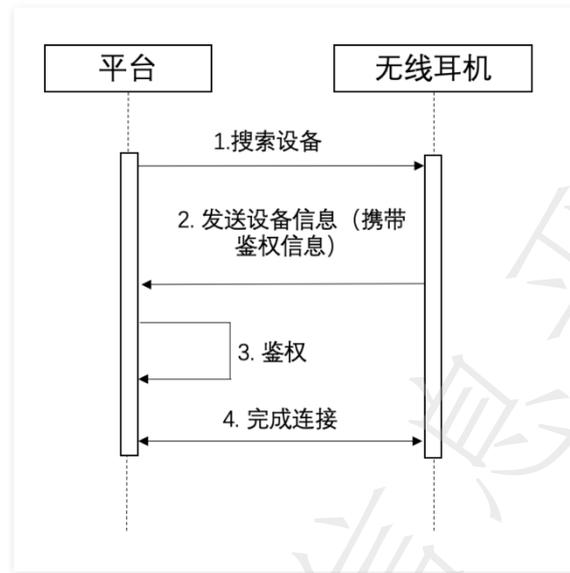


图3 设备初始化流程图

- a) 平台搜索无线耳机，进行初始化通信；
- b) 无线耳机发送设备信息，包括鉴权信息；
- c) 平台对无线耳机进行鉴权；
- d) 鉴权成功则平台与无线耳机建立连接。

具体初始化流程接口可参见附录 A。

## 6.2 空间音频播放流程

空间音频播放流程如图4所示。

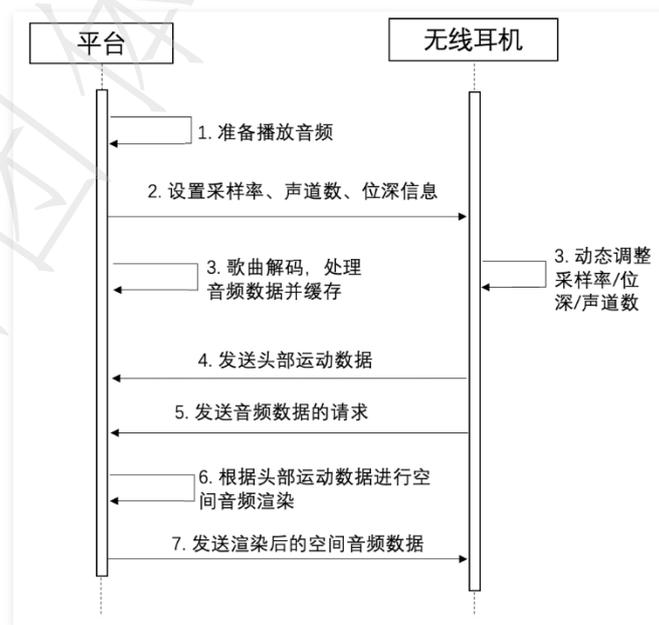


图4 空间音频播放流程图

- a) 平台准备播放音频；

- b) 在播放前，将音频的采样率、声道数和位深信息发送到无线耳机；
- c) 平台进入播放流程，解码音频数据并将音频数据在平台内进行缓存；同时无线耳机根据音频的采样率、位深、声道数进行动态调整；
- d) 无线耳机发送头部运动数据到平台；
- e) 无线耳机同时通过同一个通道，并发向平台发送 request 以请求音频数据；
- f) 平台收到无线耳机的头部运动数据和 request 后，从缓存中读取对应的音频数据，对其进行空间音频渲染处理，得到空间音频数据；
- g) 平台发送渲染后的空间音频数据到无线耳机；
- h) 当头部运动数据发生改变或需要新数据时，无线耳机重新发送头部运动数据和 request 到平台，平台根据最新的头部运动数据进行空间音频渲染处理，并返回渲染后的空间音频数据到无线耳机。

具体空间音频播放流程接口可参见附录 B。

## 7 空间音频效果评估要求及测试方法

### 7.1 空间音频全链路延时要求及测试方法

#### 7.1.1 空间音频全链路延时要求

空间音频全链路延时指平台与无线耳机播放空间音频全流程的延时，全流程包括三部分：无线耳机头部运动数据的回传；基于头部运动数据的音频渲染；渲染后的空间音频数据发送和播放。

空间音频全链路时延应 $\leq 200\text{ms}$ ，宜 $\leq 100\text{ms}$ 。

空间音频全链路流程如图 5 所示。

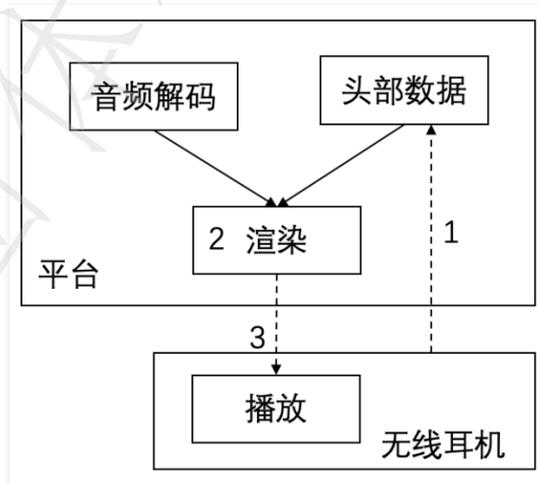


图 5 空间音频全链路图

#### 7.1.2 空间音频全链路延时测试方法

##### 1) 测试环境：

无传输信号干扰的空间环境，且环境噪声不超过 25dBA。

##### 2) 测试设备：

转动测试治具：可稳定佩戴无线耳机接收其发出的音频信号，并带动无线耳机同步转动，同时记录当前的转动角度、角速度、角加速度等数据；

无线耳机：可为头戴式、入耳式、半入耳式；

测试平台：可为具有播放能力的智能终端。

### 3) 测试要求：

将无线耳机正确的佩戴在转动测试治具上，并与测试平台建立无线连接（无线连接可以为蓝牙、wifi 或其他短距通信技术）。由测试平台、无线耳机、转动测试治具共同组成测试系统，如图 6 所示。

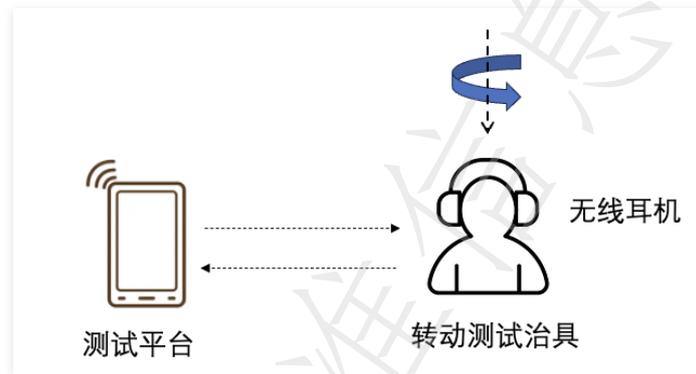


图 6 无线耳机、转动测试治具与测试平台的连接及转动

### 4) 测试步骤：

- a) 测试平台播放空间音频信号；
- b) 转动测试治具带动无线耳机从正方向  $0^\circ$  顺时针旋转至  $15^\circ$  后，测试平台记录每次转动测试治具开始转动的时间点  $T_0$ ；
- c) 测试平台收到当前角度渲染后的空间音频信号，记录其返回的时间点  $T_1$ ；
- d) 计算得到空间音频全链路延时  $T_d = T_1 - T_0 - T_t$ 。其中测试系统本身的固有延迟为  $T_t$ ，在测试前通过有线通路闭环测试得到；
- e) 转动测试治具回正后，再重复上述转动过程。旋转角度分别为  $30^\circ$ 、 $45^\circ$ 、 $60^\circ$  和  $90^\circ$ 。共计测试三组，每组包括 5 个角度的延时数据，取所有延时数据的平均值作为全链路延时。

## 7.2 空间音频的稳定性要求及测试方法

### 7.2.1 空间音频的稳定性要求

#### 7.2.1.1 跟踪运动的稳定性要求

无线耳机发生跟踪运动时，在进行固定的角度变化时，无线耳机的角度返回值与当前实际角度偏差超过  $\pm 5^\circ$  即为异常跳变。根据异常跳变计算空间音频跟踪运动的稳定性，其定义详见公式 4。

跟踪运动的稳定性要求应高于 95%。

$$\text{空间音频跟踪运动的稳定性} = 1 - \frac{\text{异常跳变次数}}{\text{总测试次数}} * 100\% \quad (4)$$

### 7.2.1.1 非跟踪运动的稳定性要求

无线耳机发生非跟踪运动时，在角度不变时，无线耳机的角度返回值与当前实际角度偏差超过 $\pm 5^\circ$ 即为异常跳变。根据异常跳变计算空间音频非跟踪运动的稳定性，其定义详见公式 5。

非跟踪运动的稳定性要求应高于 95%。

$$\text{空间音频非跟踪运动的稳定性} = 1 - \frac{\text{异常跳变次数}}{\text{总测试次数}} * 100\% \quad (5)$$

### 7.2.2 空间音频的稳定性测试方法

#### 7.2.2.1 跟踪运动的稳定性测试方法

##### 1) 测试环境

无传输信号干扰的空间环境，且环境噪声不超过 25dBA。

将无线耳机正确的佩戴在转动测试治具上，并与测试平台建立无线连接（无线连接可以为蓝牙、wifi 或其他短距通信技术），如图 7 所示。

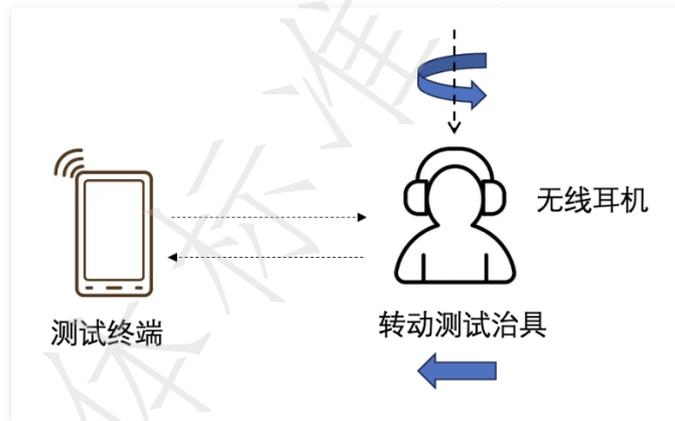


图 7 无线耳机、转动测试治具与测试平台的连接及平移/转动

##### 2) 测试设备，详见 7.1.2。

##### 3) 测试步骤：

- a) 测试平台播放空间音频信号；
- b) 对每个跟踪运动进行测试，转动测试治具带动无线耳机以旋转角速度  $10^\circ/\text{s}$  从正方向  $0^\circ$  顺时针旋转至  $15^\circ$  后停留 3 秒，再进行回正；
- c) 测试平台接收到转动测试治具的转动角度数据和无线耳机回传的转动角度数据，计算角度偏差。若角度偏差超过  $\pm 5^\circ$ ，则记为异常跳变；
- d) 再重复上述转动过程，转动角度分别为  $30^\circ$ 、 $45^\circ$ 、 $60^\circ$ 、 $90^\circ$ 、 $-15^\circ$ 、 $-30^\circ$ 、 $-45^\circ$ 、 $-60^\circ$  和  $-90^\circ$ 。旋转角速度每次增加  $10^\circ/\text{s}$ 。结合 10 个旋转角度和 10 个角速度，共计测试 100 组；
- e) 计算空间音频跟踪运动的稳定性。

#### 7.2.2.2 非跟踪运动的稳定性测试方法

测试步骤：

- a) 测试平台播放空间音频信号；
- b) 对每个非跟踪运动进行测试，转动治具带动无线耳机在角度不变的情况下以  $0.1m/s$  向前移动 30cm 后停留 3 秒，再进行归位；
- c) 测试平台接收到转动测试治具的角度数据和无线耳机回传的角度数据，计算角度偏差。若角度偏差超过  $\pm 5^\circ$ ，则记为异常跳变；
- d) 再重复上述移动过程，移动方向分别为向后、向左、向右。移动速度分别为  $1m/s$  和  $10m/s$ 。结合 4 个方向和 3 个角速度，重复 10 次，共计测试 120 组；
- e) 计算空间音频非跟踪运动的稳定性。

附录 A  
(资料性)  
空间音频初始化流程接口参考示例

以 Kotlin 语言开发实现为例，空间音频初始化流程接口示例代码如下：

### A.1 扫描设备

```
/**
 * 开始扫描设备
 * @param listener 监听回调
 */
fun startSearch(listener: SearchListener)

interface SearchListener {
    /**
     * 发现新的设备，有新的设备就回调该方法
     * @param deviceInfo 设备信息
     */
    fun onDeviceFound(deviceInfo: DeviceInfo)
}
```

### A.2 设备和鉴权信息

```
class DeviceInfo {
    val deviceName: String // 设备名称
    val deviceMacAddress: String // 设备 Mac 地址
    val authInfo: AuthInfo // 认证参数
}

class AuthInfo {
    val devcieId: String
    val deviceKey: String
    val productId: Int
}
```

### A.3 停止扫描

```
/**
 * 停止扫描设备
 */
fun stopSearch()
```

### A.4 连接设备

```
/**
 * 连接设备
 * @param deviceInfo 设备信息
 * @param listener 监听回调
 */
fun connectDevice(deviceInfo: DeviceInfo, listener:
DeviceInfoConnectListener)

interface DeviceConnectListener {
    /**
     * 设备连接成功
     * @param deviceInfo 设备信息
     */
    fun onDeviceConnected(deviceInfo: DeviceInfo): Boolean

    /**
     * 设备连接失败
     * @param deviceInfo 设备信息
     * @param errorCode 错误码（待定）
     */
    fun onDeviceConnectFailed(deviceInfo: DeviceInfo, errorCode:
Int): Boolean

    /**
     * 设备断连
     * @param deviceInfo 设备信息
     * @param reason 设备断连原因
     */
    fun onDeviceDisconnected(deviceInfo: DeviceInfo, reason:
Int): Boolean
}
```

#### A.5 断开设备

```
/**  
 * 断开设备连接  
 * @param deviceInfo 设备信息  
 */  
fun disconnectDevice(deviceInfo: DeviceInfo)
```

## 附录 B (资料性) 空间音频播放流程接口参考示例

以 Kotlin 语言开发实现为例，空间音频播放流程接口示例代码如下：

### B.1 准备播放

```
/**
 * APP 播放器就绪，准备播放，SDK 这时候可以清空 PCM 数据
 * @param MediaInfo 歌曲信息
 */
fun doOnPlayerReady(info: MediaInfo)

class MediaInfo {
    val sampleRateInHz: Int // 采样率
    val channelConfig: Int // 声道数
    val bitDepth: Int // 位深
}
```

### B.2 播放监听

```
interface PlayEventListener {
    /**
     * 陀螺仪信息更新
     * @param GyroscopeInfo 歌曲陀螺仪信息
     */
    fun onGyroscopeInfoChange(gyroscopeInfo: GyroscopeInfo)

    /**
     * 获取 PCM 数据
     * @param pcmSize 需要的 PCM 数据长度
     * @return pcm 数据
     */
    fun fetchPcmData(pcmSize: Int, pcmData: ByteArray)
}

class GyroscopeInfo {
    // TODO
}
```

### B.3 播放/暂停/停止/上一首/下一首

```
/**
 * 播放
 */
fun play()

/**
 * 暂停播放
 */
fun pause()

/**
 * 停止播放
 */
fun stop()

/**
 * 拖动进度条，清空 SDK 内 PCM 缓存
 */
fun seek()
```

### B.4 陀螺仪开启

```
/**
 * 是否开启陀螺仪
 * @param enable 是否开启陀螺仪: true -> 开启 false -> 关闭
 */
fun setGyroscopeEnabled(enable: Boolean)
```

### B.5 空间音频算法开启

```
/**
 * 是否开启空间音频算法
 * @param enable 是否开启空间音频算法: true -> 开启 false -> 关闭
 */
fun setSpatialAudioAlgorithmEnabled(enable: Boolean)
```

## B.6 传输模式设置

```
/**
 * 设置传输模式
 * @param transMode
 */
fun setSpatialAudioAlgorithmEnabled(transMode: TransMode)

enum class TransMode(val rawValue: Byte) {
    STABLIZE(0x00) // 稳定传输
    QUALITY(0x01) // 音质优先传输
}
```