



(12) 发明专利申请

(10) 申请公布号 CN 103235736 A

(43) 申请公布日 2013. 08. 07

(21) 申请号 201310101090. 9

(22) 申请日 2013. 03. 27

(71) 申请人 四川长虹电器股份有限公司

地址 621000 四川省绵阳市高新区绵兴东路
35 号

(72) 发明人 周建波

(74) 专利代理机构 成都虹桥专利事务所(普通
合伙) 51124

代理人 任虹 刘世平

(51) Int. Cl.

G06F 9/455(2006. 01)

G06F 3/16(2006. 01)

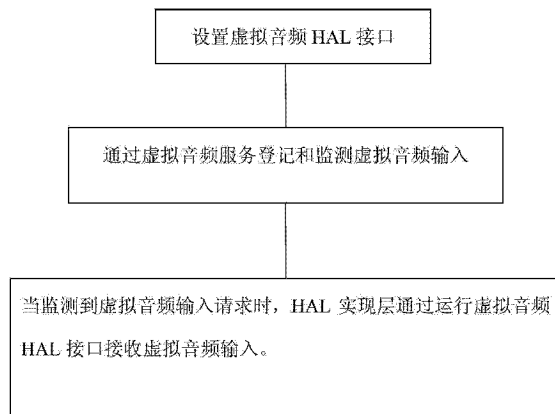
权利要求书1页 说明书4页 附图2页

(54) 发明名称

基于 Android 平台支持虚拟音频设备的方法
及系统

(57) 摘要

本发明公开了一种基于 Android 平台支持
虚拟音频设备的方法和系统,包括以下步骤:设
置虚拟音频 HAL 接口;通过虚拟音频服务登记和
监测虚拟音频输入;当监测到虚拟音频输入请
求时,HAL 实现层通过运行虚拟音频 HAL 接口接
收虚拟音频输入。本发明可用于基于 Android 平
台的智能消费电子产品,可以使用户能够便捷、
低成本地使用各种音频装置进行语音输入。



1. 基于 Android 平台支持虚拟音频设备的方法,其特征在于包括以下步骤:
设置虚拟音频 HAL 接口;
通过虚拟音频服务登记和监测虚拟音频输入;
当监测到虚拟音频输入请求时, HAL 实现层通过运行虚拟音频 HAL 接口接收虚拟音频输入。
2. 根据权利要求 1 所述的基于 Android 平台支持虚拟音频设备的方法,其特征在于所述的虚拟音频 HAL 接口包括打开音频流接口、读取音频数据接口、停止音频数据采集接口、关闭音频流接口。
3. 根据权利要求 1 所述的基于 Android 平台支持虚拟音频设备的方法,其特征在于还包括设置设备参数,该设备参数包括有两种状态,指示 HAL 实现层是接收原生的物理音频输入,还是接收虚拟音频输入。
4. 根据权利要求 3 所述的基于 Android 平台支持虚拟音频设备的方法,其特征在于还包括基于应用层面设立一种后台服务,该后台服务在满足虚拟音频设备启用的条件下对上述设备参数进行设置。
5. 基于 Android 平台支持虚拟音频设备的系统,其特征在于包括:
音频终端;
虚拟音频设备端,该设备端基于 Android 平台,包括有虚拟音频服务模块、虚拟音频 HAL 接口、HAL 实现层模块,虚拟音频服务模块对音频终端进行登记和监测,当监测到音频终端有虚拟音频输入请求时, HAL 实现层模块通过运行虚拟音频 HAL 接口对音频终端进行控制并接收音频输入。

基于 Android 平台支持虚拟音频设备的方法及系统

技术领域

[0001] 本发明属于数字音频处理技术领域,具体涉及一种基于 Android 平台支持虚拟音频设备的方法及系统。

背景技术

[0002] 目前消费电子产品已进入智能时代,例如在智能电视中,音频输入设备已经普遍在应用中使用,但使用电视终端时,用户是有一定观看距离的,导致对电视终端的音频输入设备要求很高,如采用带背景噪声消除的阵列麦克风、RF 音频或蓝牙音频设备输入。这种音频设备配置成本较高,一般不可能在所有智能电视终端配置,而没有语音输入设备的电视终端,其智能电视应用的丰富性、可使用性都会下降。而现阶段大部分智能电视采用 Android 系统,因此如何在基于 Android 平台的消费电子产品上使用户能够便捷、低成本地使用各种音频装置进行语音输入,是需要解决的一个技术问题。

发明内容

[0003] 本发明的目的在于提出一种基于 Android 平台支持虚拟音频设备的方法及系统,可以使用户能够便捷、低成本地使用各种音频装置进行语音输入,解决传统技术中对用户语音输入设备要求高的技术问题。

[0004] 本发明是这样实现的,基于 Android 平台支持虚拟音频设备的方法,包括以下步骤:

[0005] 设置虚拟音频 HAL 接口;

[0006] 通过虚拟音频服务登记和监测虚拟音频输入;

[0007] 当监测到虚拟音频输入请求时,HAL 实现层通过运行虚拟音频 HAL 接口接收虚拟音频输入。

[0008] 本发明的另一目的在于提供一种基于 Android 平台支持虚拟音频设备的系统,该系统包括:

[0009] 音频终端;

[0010] 虚拟音频设备端,该设备端基于 Android 平台,包括有虚拟音频服务模块、虚拟音频 HAL 接口、HAL 实现模块,虚拟音频服务模块对音频终端进行登记和监测,当监测到音频终端有虚拟音频输入请求时,HAL 实现层模块通过运行虚拟音频 HAL 接口对音频终端进行控制并接收音频输入。

[0011] 本发明的有益效果是:由于本发明从 Android 平台的 HAL 层入手,使音频设备能够接收各种外部虚拟音频输入,也就是说,这种发明实际上是延伸了物理设备的概念,消费者就无需购买专门的音频输入设备,使用已有的、特别是便携语音终端如手机,就可进行语音输入,因此具有成本低、使用便捷的优点。

附图说明

[0012] 图 1 是本发明的流程图。

[0013] 图 2 是本发明的系统架构图。

[0014] 图 3 是本发明的打开虚拟音频流接口、读取虚拟音频数据接口流程图。

具体实施方式

[0015] 下面结合附图和具体实施例对本发明作进一步说明。

[0016] 在本发明实施例中,包括一种方法和一种系统,其中方法流程如图 1,系统架构如图 2,相关的虚拟音频流的打开和读取流程则如图 3 所示。

[0017] 如图 1,一种基于 Android 平台支持虚拟音频设备的方法,包括以下步骤:

[0018] 设置虚拟音频 HAL 接口;

[0019] 通过虚拟音频服务登记和监测虚拟音频输入;

[0020] 当监测到虚拟音频输入请求时,HAL 实现层通过运行虚拟音频 HAL 接口接收虚拟音频输入。

[0021] 进一步详细说明如下:

[0022] 构建包含音频输入的 HAL 实现层对应的打开音频流接口、读取音频数据接口、停止音频数据采集接口、关闭音频流接口的虚拟设备的 HAL 接口,以及一个设备描述字段,接口如下:

[0023]

```
typedef struct {  
    char devdesc[128]  
  
    void (openVirtual*)(uint32_t sample_rate, int format, int channel_count);  
    ssize_t (readVirtual*)(void *buffer, size_t bytes);  
    int (standbyVirtual*)(struct audio_stream* stream);  
    void (closeVirtual*)();  
}VirtualInputStreamer;
```

[0024] 其中的 open 对应打开音频流接口,read 对应读取音频数据接口,standby 对应停止音频数据采集接口,close 对应关闭音频流接口。

[0025] 实现上述虚拟设备的 HAL 接口,并编译成动态链接库,用于系统运行时动态加载;

[0026] 在 android 的音频服务中设置设备参数变量,该参数可通过 android 原生的 AudioSystem 接口设置,该设备参数包括有两种状态,指示 HAL 实现层是接收原生的物理音频输入,还是接收虚拟音频输入。在本实施例中,其值为空表示采用原生的物理音频输入设备,否则就表示要接收虚拟音频输入;

[0027] 基于应用层面设立一种后台服务,该后台服务在满足虚拟音频设备启用的条件下,对该设备参数进行设置,以告知 android 系统中的 audio 的 HAL 实现层将采用该虚拟音频输入设备;

[0028] 在 android 系统音频输入的 HAL 实现层对应的音频流打开接口中,依据实现设备参数的状态实现对本发明的虚拟设备的 HAL 接口动态链接库的加载;

[0029] 在 android 系统音频输入的 HAL 实现层对应的音频流读取接口中,调用动态链接库中的音频输入接口读取音频 PCM 数据;

[0030] 在 android 系统音频输入的 HAL 实现层对应的音频流停止接口中,调用动态链接库的 standby 接口;

[0031] 在 android 系统音频输入的 HAL 实现层所对应的音频流关闭接口中,调用动态链接库的关闭流接口,同时关闭对应的动态链接库对象;

[0032] HAL 实现层对接虚拟设备的音频流打开接口 openVirtual 及音频流读取接口 readVirtual 流程如图 3 所示,通过上述发明内容,应用调用标准的 Android 音频输入接口,通过 audio 系统的 HAL 层操作麦克风输入设备时,实际上已经操作到了虚拟的麦克风输入设备中;上述这些发明内容是一种通用的虚拟 HAL 层,只要按这种方式对接,就可以实现多种虚拟麦克风设备,而不局限于特定的一种虚拟麦克风接入方式。

[0033] 通过本发明虚拟设备接口的设置,可以有序地对虚拟设备进行操作,同时通过设备参数变量的设置,可以方便地兼容和切换原生的物理音频输入、虚拟音频输入两种状态。

[0034] 如图 2,一种基于 Android 平台支持虚拟音频设备的系统,该系统包括:

[0035] 音频终端;

[0036] 虚拟音频设备端,该设备端基于 Android 平台,包括有虚拟音频服务模块、虚拟音频 HAL 接口、HAL 实现层模块,虚拟音频服务模块对音频终端进行登记和监测,当监测到音频终端有虚拟音频输入请求时,HAL 实现层模块通过运行虚拟音频 HAL 接口对音频终端进行控制并接收音频输入。

[0037] 以实现一种通过网络从手机 /PAD 等移动终端获取 PCM 数据的虚拟音频设备为例,即将手机 /PAD 作为音频终端,具体说明如下:

[0038] 在虚拟音频设备端设立虚拟音频服务模块,以实现一个本地常驻服务,该服务模块监听移动终端的自举信息,将移动终端的 IP 地址记录下来,同时,如果接收到来自移动终端的启用麦克风的通讯协议,通过 AudioSystem.setParameters 函数,设置系统的设备参数 devinput 为“libvirtualmic.so”,将该值记录到系统属性 media.audio.devinput 中,供后续 HAL 实现层模块读取。

[0039] 虚拟音频设备端按如下方式实现发明内容中的虚拟音频 HAL 接口:

[0040] 打开音频流接口——openVirtual:按约定的网络协议将采样率、音频格式、通道数发送给移动终端,移动终端记录电视终端要求的采样率等信息,同时发送 ACK 应答;

[0041] 读取音频流接口——readVirtual:从约定好的端口按 UDP 协议读取指定字节数的数据,并设立超时机制;

[0042] 停止音频流接口——standbyVirtual:按约定好的网络协议发送信息给移动终端,告知移动终端停止麦克风数据的采集;

[0043] 关闭音频流接口——closeVirtual:本实施例不需要实现;

[0044] 上述接口方式实现完后,编译成动态链接库,名字与 devinput 参数一致,为 libvirtualmic.so,之后就可以加载运行了

[0045] android 系统按标准音频 HAL 接口打开音频输入流时,检测到系统属性 media.audio.devinput,如果不为空字符串 NULL,则加载对应的动态链接库,其对应对象记为 virtualInStreamer,转而调用 virtualInputStreamer 对象的 openVirtual 接口;

[0046] 如果 virtualInputStreamer 对象存在,则调用该对象的 readVirtual 接口读取音频数据;

[0047] 同样调用 virtualInputStreamer 对象的 standby 接口,以停止音频数据采集;

[0048] 如果 virtualInputStreamer 对象不为 NULL,则调用其 closeVirtual 接口,调用完后,将 virtualInputStreamer 对象置空并关闭其对应的动态链接库。

[0049] 这样虚拟音频设备端就能够从手机 /PAD 获取语音数据了,因此通过本发明,可以在智能电视机中设立该虚拟音频设备端,用户通过已有的手机 /PAD 对智能电视机进行各种语音操作,方便快捷,距离范围很宽,而无需购买带背景噪声消除的阵列麦克风等专用设备,从而节省了用户的使用成本。

[0050] 在本实施例中,语音终端可以有多个,即通过本发明,用户可以通过多个手机 /PAD 对智能电视机进行语音操作,使用上更加灵活方便。

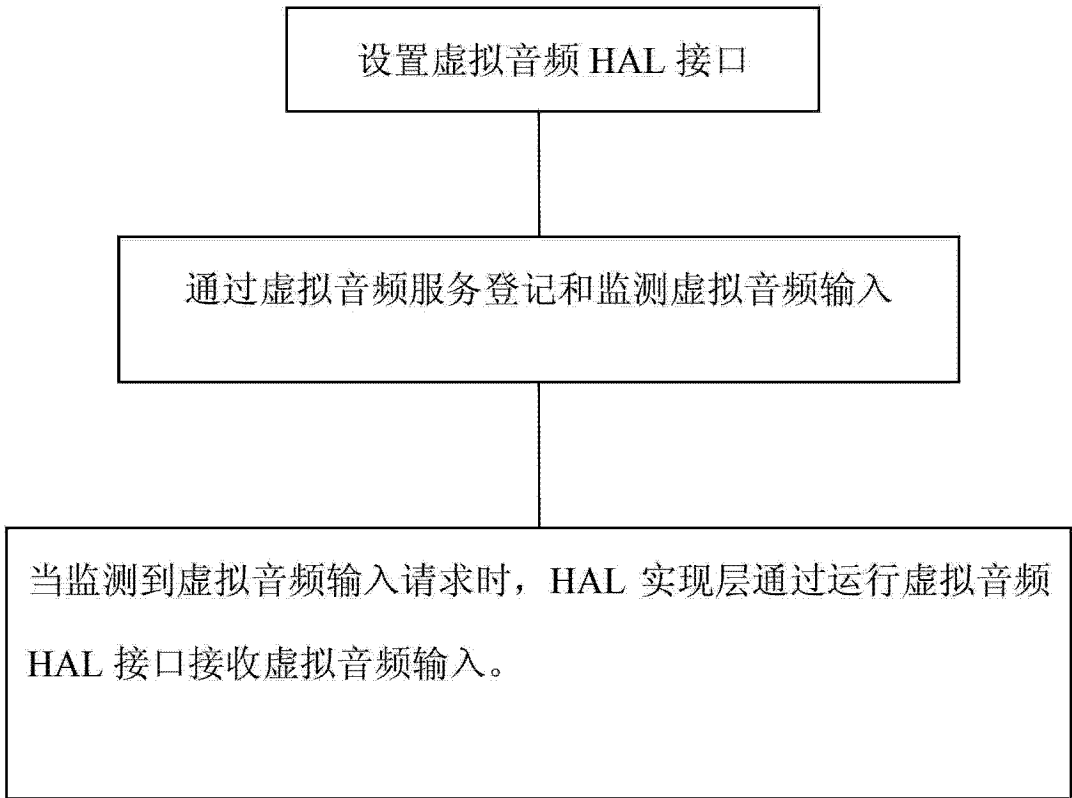


图 1

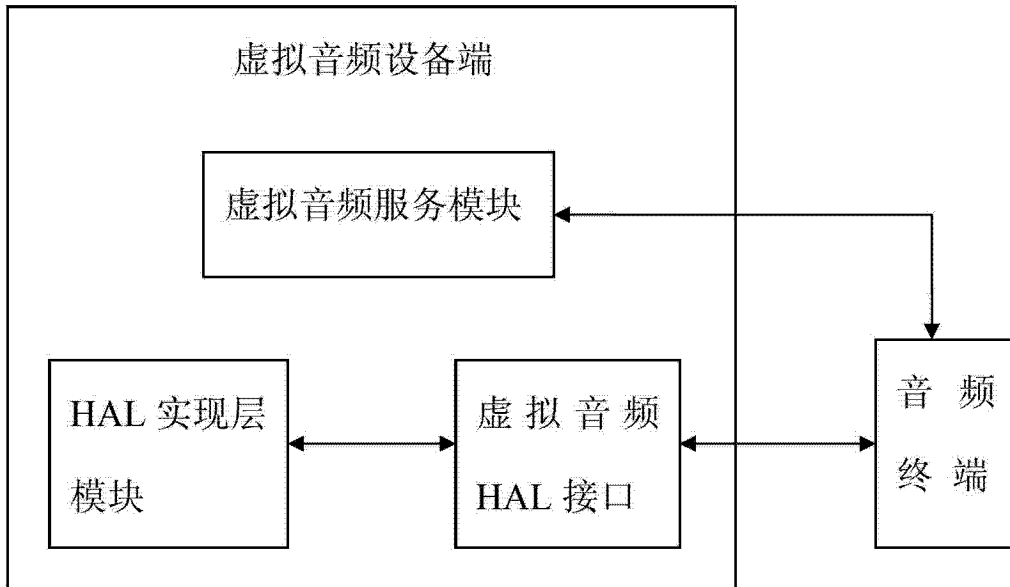


图 2

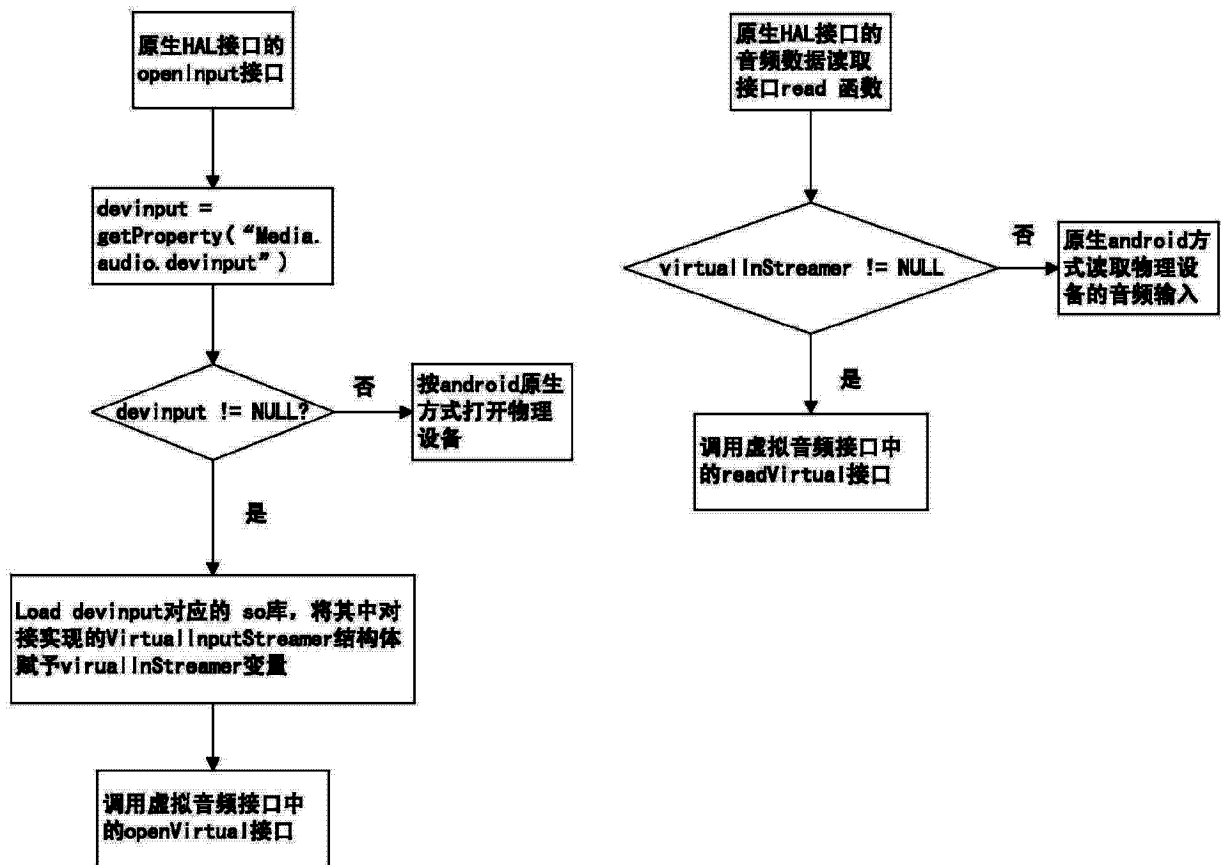


图 3