



A20 音频切换策略

V1.2

2014-06-16

Confidential

Revision History

Version	Date	Changes compared to previous issue
V1.0	2014-06-05	初建版本
V1.1	2014-06-14	增加待机唤醒和开关机策略，对 USB 音频策略作补充，增加音频策略定制方法。
V1.2	2014-06-16	补充透传模式下的策略

Confidential



目录

1. 前言-----	4
1.1. 编写目的-----	4
1.2. 适用范围-----	4
1.3. 相关人员-----	4
2. 音频输出切换策略-----	5
3. 音频输出策略定制-----	7
4. 音频输入策略-----	11
5. Declaration-----	12

Confidential

1. 前言

1.1. 编写目的

本文档目的是描述 A20 盒子音频切换策略。

1.2. 适用范围

本模块说明适用于 A20 平台

1.3. 相关人员

音频系统开发人员，普通用户。

Confidential

2. 音频输出切换策略

音频输出策略如表所述。

表 1 TV-BOX 音频输出切换策略

当前状态	音频输出	具体操作	操作后显示状态	音频输出
只插着 HDMI	HDMI 输出或用户手动更改	拔出 HDMI 线	输出 NONE	不做任何处理
		插入 CVBS 线	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	HDMI 输出	添加 HDMI 输出
只插着 CVBS	CODEC 输出或用户手动更改	拔出 CVBS 线	输出 NONE	不做任何处理
		插入 HDMI 线	切换到 HDMI	添加 HDMI 输出
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	CVBS 输出	添加 CODEC 输出
同时插着 HDMI 和 CVBS	默认跟随当前显示输出或用户手动更改	拔出当前显示类型的设备线	切换到另一显示类型	跟随显示添加输出
		拔出非当前显示类型的设备线	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		手动切换到设备支持的显示类型, 包括 setting 和遥控 (phoneWindowManage)	成功切换后, 保存该显示类型到数据库	跟随显示添加输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	HDMI 输出	添加 HDMI 输出
无插着任何设备	默认 CODEC 输出	插入 HDMI 线	HDMI 输出	添加 HDMI 输出
		插入 CVBS 线	CVBS 输出	添加 CODEC 输出
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	输出 NONE	不做任何处理
USB 音频输出	仅 USB 音频输出	插拔 HDMI, 插拔 CVBS, 待机, 唤醒, 关机, 开机	相应变化	不做任何处理
		拔出 USB 音频设备	不做任何处理	切换到当前显示对应音频输出
		手动选择其他音频	不做任何处理	切换到对应输出

开启透传模式	SPDIF 透传	插拔 HDMI, 插拔 CVBS, 待机, 唤醒, 关机, 开机	相应变化	不做任何处理
	HDMI 透传	插拔 CVBS, 待机, 唤醒, 关机, 开机	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		拔出 USB 音频输出设备	不做任何处理	切换到当前显示对应音频输出

说明:

1、关于 SPDIF 的输出问题:

SPDIF 仅支持手动设置输出。无法自动检测切换。**除了插入 USB 音频输出设备外, 其他情况的插拔不会影响 SPDIF 的输出状态。**

2、插入 USB 音频设备后不能复选多路。

3、系统第一次开机前插入 USB 音频设备, 系统只会切到对应的显示音频输出上。

4、显示自动切换到 CVBS 时, CODEC 会自动开启。但手动拔出 CVBS 后 CODEC 会保留。只有在插入 USB 音频或手动清除选项后才会不复选上 CODEC。

5、勾选了透传模式, 只能选择单路输出。

6、只有 HDMI 和 SPDIF 才能实现透传。透传模式下, 选择 CODEC 和 USB 音频输出, 不会有透传的效果, 同时也不会影响正常的音频输出。

7、透传模式下无法使用遥控器控制音量和静音。



3. 音频输出策略定制

1、相关文件

1) sugar/android/frameworks/base/services/java/com/android/server/SystemServer.java

文件作用：初始化音频输出通道

相关内容说明：

```
AudioManager audioManager = new AudioManager(context);
if(audioManager == null){
    Log.e(TAG, "audioManager is null");
}else {
    //音频输出初始化内容
    //读取数据库音频输出通道记录
    String audioOutputChannelName =
        Settings.System.getString(context.getContentResolver(),
                                   Settings.System.AUDIO_OUTPUT_CHANNEL);
    //处理 USB 音频输出
    if(audioOutputChannelName.contains("USB")){

    //配置非 USB 音频输出
    if(!usbAudioOutputUsedAndExist){
        if(audioOutputChannels.contains(AudioManager.AUDIO_NAME_SPDIF)){
            activatedChannels.add(AudioManager.AUDIO_NAME_SPDIF);
        }
        if(audioOutputChannels.contains(AudioManager.AUDIO_NAME_CODEC)){
            activatedChannels.add(AudioManager.AUDIO_NAME_CODEC);
        }
        if(finalFormat.mOutputType ==
            DisplayManagerAw.DISPLAY_OUTPUT_TYPE_HDMI){
            activatedChannels.add(AudioManager.AUDIO_NAME_HDMI);
        }else if(!audioOutputChannels.contains
            (AudioManager.AUDIO_NAME_CODEC))
        {
            activatedChannels.add(AudioManager.AUDIO_NAME_CODEC);
        }
    }
}
//记录输出通道状态
/* record the audio output channel */
String st = null;
for(int i = 0; i < activatedChannels.size(); i++){
```



```
if(st == null){
    st = activatedChannels.get(i);
} else {
    st = st + "," + activatedChannels.get(i);
}
}
Settings.System.putString(context.getContentResolver(),
    Settings.System.AUDIO_OUTPUT_CHANNEL,st);
Log.d(TAG,"audio channels change to {" + st + "}");
```

- 2) sugar/android/frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/policy/SoundController.java

文件作用：处理 USB 音频插拔切换

相关内容说明：

```
public void onReceive(Context context, Intent intent) {
    //音插拔事件接收器。

    if(name.equals(AudioManager.AUDIO_NAME_HDMI)
        || (name.equals(AudioManager.AUDIO_NAME_CODEC) && ((extra == null)
        || (extra != null) && !extra.equals(AudioDeviceManagerObserver.H2W_DEV))))
        || (name.equals(AudioManager.AUDIO_NAME_SPDIF))) {
        //handleInternalDevice(name, type, state, extra);
        //处理非音频设备事件，转到 DisplayController.java 处理。
    } else {
        handleExternalDevice(name, type, state, extra);
        //处理 USB 音频插拔事件。
    }
}
```

- 3) sugar/android/frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/policy/DisplayController.java

文件作用：处理显示热插拔事件，及跟随显示情况切换音频

相关内容说明：

```
public void onReceive(Context context, Intent intent) //插拔事件接收器
private void switchDisplayOutput(int type, int format) //真正执行切换的方法
```

- 4) sugar/android/device/softwinner/common/packages/TvdSettings/src/com/android/settings/AudioChannelsSelect.java

文件的作用：控制系统设置中音频输出模式的选择。

控制多选和单选：

```
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
    CheckBox check = (CheckBox)arg1.findViewById(R.id.check_audio_selected);
    if(check.isChecked()){
```




```
check.setChecked(false);
mChannels.remove(mTable.getItem(arg2));
}else{
    check.setChecked(true);
    for(String st:mChannels){
        if(st.contains("USB")){
            mChannels.clear();
        }
    }
    if (mEnablePassThrough || mTable.getItem(arg2).contains("USB"))
        mChannels.clear();
    //注销 mChannels.clear();的过滤条件即可实现单选。
    mChannels.add(mTable.getItem(arg2));
}
```

- 5) sugar/android/frameworks/base/media/java/android/media/AudioManager.java
文件的作用：实现 setAudioDeviceActive 接口，同时过滤透传和非透传模式。
相关代码：

```
public void setAudioDeviceActive(ArrayList<String> devices, String state){
    .....
    if (enablePassThrough && state.equals(AUDIO_OUTPUT_ACTIVE)) {
        boolean usbConnected = false;
        for(String dev_name:devices){
            if(dev_name.contains("USB")){
                usbConnected = true;
                audio = dev_name;
                break;
            }
        }
        if(usbConnected){
            MediaPlayer.setRawDataMode(MediaPlayer.AUDIO_DATA_MODE_PCM);
            //如果是 USB 设备，在透传模式勾选的情况下，输出非透传数据。
        }else{
            if (devices.contains(AUDIO_NAME_SPDIF)) {
                audio = AUDIO_NAME_SPDIF;
                MediaPlayer.setRawDataMode(MediaPlayer.AUDIO_DATA_MODE_SPDIF_RAW);
                //如果是 SPDIF 设备，在透传模式勾选的情况下，输出 SPDIF 透传数据。
            } else if (devices.contains(AUDIO_NAME_HDMI)){
                if(hdmi) {
                    audio = AUDIO_NAME_HDMI;
                    MediaPlayer.setRawDataMode(MediaPlayer.AUDIO_DATA_MODE_HDMI_RAW);
                    //如果是 HDMI 设备，在透传模式勾选的情况下，输出 HDMI 透传数据。
                }
            }
        }
    }
}
```



```
        } else {  
            audio = AUDIO_NAME_CODEC;  
            MediaPlayer.setRawDataMode(MediaPlayer.AUDIO_DATA_MODE_PCM);  
            //如果是 CODEC 设备，在透传模式勾选的情况下，输出非透传数据。  
        }  
    } else {  
        .....  
    }  
    if (!enablePassThrough) {  
        //如果是在透传模式未勾选的情况下，输出非透传数据。  
        MediaPlayer.setRawDataMode(MediaPlayer.AUDIO_DATA_MODE_PCM);  
    }  
    setParameter(state, audio); //真正设置参数。  
}  
}
```

2、单路与多路

Sugar SDK2.1 支持在系统设置中选择多路输出。SDK2.1 以前则只能单选。若有需要修改为单选，则把上述文件还原到 SDK2.0 即可。

4. 音频输入策略

音频输入策略如表 2 所述。

表 2 音频输入策略

当前设备	音频输入	具体操作	音频输入
CODEC	内置 MIC 或 LINE IN	插入 USB 音频输入设备	切换到 USB 音频输入
USB 音频设备	USB 音频输入	拔出 USB 音频设备	切换到 CODEC 音频输入

说明：

- 1、系统默认的音频输入设备是 CODEC。
- 2、内置 MIC 或者 LINE IN 使用的均为 CODEC 声卡。
- 3、USB 音频输入设备一般有：USB MIC、带 MIC 的 USB Camera、带 MIC 的 USB 空鼠。

5. Declaration

This is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

Confidential