中'zhong﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽驱动会抛出唤醒信号和与

**思必驰**

**麦克风阵列模组SPI驱动移植介绍**

2016-06-04

# 1. 功能概述

思必驰麦克风阵列模组主要实现了多通道的语音数据采集和处理，同时具有回声消除、语音唤醒和声源定位的功能。

麦克风阵列模组由麦克风阵列采集板和上位机驱动组成。采集板支持7路模拟麦克风输入以及1路用于回声消除的参考音，通过SPI接口将多通道语音传输到上位机。

上位机驱动支持Android和Linux两种平台。

# 2. SPI通信

引脚定义表：

|  |  |  |
| --- | --- | --- |
| 引脚名称 | 功能描述 | 备注 |
| CS | SPI片选信号 | 低电平有效 |
| SCK | 时钟信号 | 10MHZ |
| MOSI | 主输出从输入 |  |
| MISO | 主输入从输出 |  |

SPI：cpol = 0， cpha=0

数据帧格式：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Magic Number (8) | | | | | | | | | | |
| Version (1) | Header Length (1) | Total Length (2) | | | | Sequence Number (4) | | | | |
| OP Code (1) | Overrun Flag (1) | Test Mode (1) | | Sample Size (1) | | | Padding (7) | | | |
| CHDB0 (1) | CHDB1 (1) | CHDB2 (1) | CHDB3 (1) | | CHDB4 (1) | | | CHDB5 (1) | CHDB6 (1) | CHDB7 (1) |
| PCM Data (0 - 7680) | | | | | | | | | | |
| Margin Padding (16) | | | | | | | | | | |

Magic Number：定义为“AISPEECH”，用来区分有效数据帧

Version：版本信息

Header Length：数据帧头长度，32字节

Total Length：数据帧有效长度，不包括Margin Padding长度

Sequence Number：主要用于检测数据是否有丢帧

OP Code：保留字段

Overrun：标识采集板是否已经出现丢帧异常

Test mode：标识采集板当前是否处于测试模式

Sample Size：指示当前音频采样精度, 24或者16比特

CHDB0 – 6：Codec音频增益[0,80]，计量单位1/2db

PCM Data：音频数据

Margin Padding：冗余字段，确保整个数据帧传输完整。

**注意**：对于部分MTK平台，每次传输的SPI数据帧长度需要是1024的整数倍，因此如果有如下类似错误提示，需要调整Margin Padding长度。

[116:spiaud\_transfer]mt-spi mt-spi.0: ERROR!!The lens must be a multiple of 1024, your len 5184

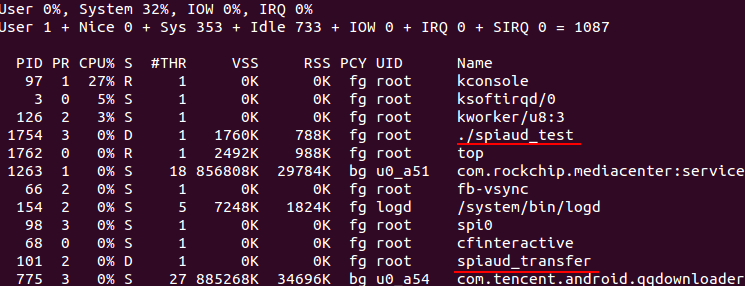
Margin Padding长度定义在源文件aispeech-spi-audio.c里

#define SPI\_AUDIO\_TAIL\_MARGIN\_LEN 16

# 3. SPI驱动示例

SPI传输方式的验证是基于瑞芯微开放平台。上位机可以正常跟下位机进行SPI通信，接收下位机传过来的音频采集数据。

经过测试程序验证，CPU占有率还是比较低的，如下图：



spiaud\_transfer是kernel里创建的线程，专门负责跟SPI slaver通信。

spiaud\_test是应用层的测试进程，通过kernel提供的接口（/dev/spiaud）读取SPI传过来的音频采集数据。

# 4. SPI驱动移植

用户系统需要做如下几点修改：

1. 定义出SPI的DTS（Device Tree Source）节点，类似如下：

kernel/arch/arm/boot/dts/firefly-rk3288.dts

&spi0 {

status = "okay";

max-freq = <24000000>;

spi\_audio@00 {

compatible = "**aispeech,spi\_audio**";

reg = <0>;

spi-max-frequency = <10000000>;

};

};

1. 修改aispeech-spi-audio.c里Device Tree match table

static const struct of\_device\_id spi\_audio\_dt\_match[] = {

{ .compatible = "**aispeech,spi\_audio**", },

{},

};

1. 把aispeech-spi-audio.c和aispeech-spi-audio.h放到kernel/drivers/spi目录下，修改kernel编译配置

kernel/drivers/spi/Kconfig

config SPI\_SPIAUD

tristate "AISpeech spi audio code"

default y

kernel/drivers/spi/Makefile

obj-$(CONFIG\_SPI\_SPIAUD) += aispeech-spi-audio.o

1. 修改system/core/rootdir/ueventd.rc，将/dev/spiaud的属性设置为普通用户可读写

# SPI audio interface

/dev/spiaud 0666 system system

# 5. 移植结果验证

根据sample目录下的Android.mk和测试源码可以直接编译出spiaud\_test测试程序

|  |
| --- |
| LOCAL\_PATH:= $(call my-dir) |
| include $(CLEAR\_VARS) |
|  |
| LOCAL\_SHARED\_LIBRARIES += \ |
| libcutils \ |
| libutils |
|  |
| LOCAL\_C\_INCLUDES:= |
|  |
| LOCAL\_SRC\_FILES := \ |
| spi\_audio.c \ |
| spiaudiotest.c |
|  |
| LOCAL\_MODULE := spiaud\_test |
| include $(BUILD\_EXECUTABLE) |

将编译出来的spiaud\_test程序用adb push到系统/data目录下，就可以验证SPI音频驱动的结果。

命令如下：

adb push spiaud\_test /data/spiaud\_test

adb shell chmod 777 /data/spiaud\_test

adb shell /data/spiaud\_test

录制的音频默认保存在/data/spi.pcm

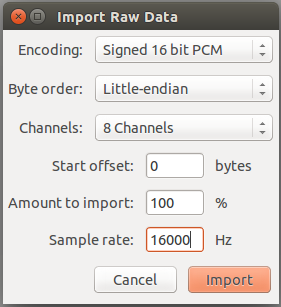
用命令adb pull可以将保存好的音频文件抓取出来

adb pull /data/spi.pcm

音频文件可以用audacity这个软件打开，打开方式：

File → Import → Raw data

参数选择如下：



然后就可以直接播放导入的音频。

如果能正常获取音频数据，kernel输出来的log如下：

<4>[ 53.239222] spiaud\_open() status=0, user=0  
<4>[ 53.245020] spiaud\_process\_recv\_data(), context len:5120, base sequence number:93963  
<4>[ 62.963750] spiaud\_release() user=1  
<4>[ 62.963771] SPI transfer, cur=1865 cost(us)[4734,4827,4717,5717,4783,4716,4731,5767,4824,4711]  
<4>[ 62.966956] spiaud\_loopTranferFunc wait

如果有异常（出现**warning**关键字），现有的log无法定位的话，需要开启更多log

命令：

adb shell 'echo F > /proc/spiaud/debug'

查看kernel log的命令：

adb shell cat /proc/kmsg