EC2x&AG35-Quecopen

Audio API 使用指导

LTE Standard/Automotive Module Series

版本：EC2x&AG35-Quecopen_Audio_API_使用指导_V1.0

日期：2017-11-15

状态：临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司
上海市徐汇区虹梅路 1801 号宏业大厦 7 楼　　邮编：200233
电话：+86 21 51086236　　邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：
http://www.quectel.com/cn/support/sales.htm

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：
http://www.quectel.com/cn/support/technical.htm
或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2017-11-15	钱润生	初始版本

目录

1 QuecOpen Audio API 介绍

QuecOpen 提供了一组关于播放、录音的 audio 接口，所操作的数据流都是 pcm 数据流。

1.1. 播音接口

1.1.1. 播放回调函数

```
// Description:
//     This callback function handles the result of audio player.
//
// @param hdl:
//     Handle received from Ql_AudPlayer_Open().
// @param result:
//     the executing result for previous operation, such as Open, Play, Pause, Resume, Stop.
//     see the definition of Enum_AudPlayer_State for the specific meaning.
typedef int(*_cb_onPlayer)(int hdl, int result);
```

1.1.2. 播放设备打开与关闭

```
/*****************************************************************
* Function:        Ql_AudPlayer_Open
*
* Description:
*                  Open audio play device, and specify the callback function.
*                  This function can be called twice to play different audio sources.
*
* Parameters:
*                  device: a string that specifies the PCM device.
*                          NULL, means the audio will be played on the default PCM device.
*
*                          If you want to mixedly play audio sources, you can call this
*                          API twice with specifying different PCM device.
*                          The string devices available:
*                          "hw:0,0" (the default play device)
```

```
*                               hw:0,13" (this device can mix audio and TTS)
*                               "hw:0,14"
*
*                   cb_func: callback function for audio player.
*                               The results of all operations on audio player
*                               are informed in callback function.

* Return:
*                   pcm device handle
*                   NULL, fail
*****************************************************************/
int   Ql_AudPlayer_Open(char* device, _cb_onPlayer cb_func);
void Ql_AudPlayer_Close(int hdl);
```

1.1.3. 播放接口

```
int   Ql_AudPlayer_Play(int hdl, unsigned char* pData, unsigned int length);
int   Ql_AudPlayer_PlayFrmFile(int hdl, int fd, int offset);
```

1.1.4. 控制接口

```
int   Ql_AudPlayer_Pause(int hdl);
int   Ql_AudPlayer_Resume(int hdl);
void Ql_AudPlayer_Stop(int hdl);
```

1.2. 录音接口

1.2.1. 录音设备打开与关闭

```
/*****************************************************************
* Function:         Ql_AudRecorder_Open
*
* Description:
*                   Open audio record device, and specify the callback function.
*
* Parameters:
*                   device: not used. MUST be NULL.
*
*                   cb_func: callback function for audio player.
*                   The results of all operations on audio recorder
*                   are informed in callback function.
```

```
*
* Return:
*                   pcm device handle
*                   NULL, fail
*****************************************************************/
int   Ql_AudRecorder_Open(char* device, _cb_onRecorder cb_fun);
void Ql_AudRecorder_Close(void);
```

1.2.2. 录音接口

```
int   Ql_AudRecorder_StartRecord(void);
```

1.2.3. 控制接口

```
int   Ql_AudRecorder_Pause(void);
int   Ql_AudRecorder_Resume(void);
void Ql_AudRecorder_Stop(void);
```

1.3. Tone 音接口

```
/*******************************************
* Description:
*             open tone device
* Parameters:
*             device, must be NULL
*             cb, must be NULL
*Return:
*             if success, return 0;
*             if failed, return -1;
*******************************************/
int Ql_AudTone_Open(char* device, _cb_onPlayer cb);

struct Ql_TonePara {
    unsigned int lowFreq;     //100-4000HZ
    unsigned int highFreq;    //100-4000HZ
    unsigned int volume;      //0 -1000
    unsigned int duration;    // >0 ms
};
int Ql_AudTone_Start(int hdl, struct Ql_TonePara *para);
void Ql_AudTone_Stop(int hdl);
void Ql_AudTone_Close(int hdl);
```

2 播音

2.1. 编程步骤

（1）打开设备.如果要支持混音，需要打开两个设备。
（2）如果 PCM 数据放在 buff 里面，直接调用 QI_AudPlayer_Play()
（3）如果 PCM 数据放在文件里面，直接调用 QI_AudPlayer_PlayFrmFile()
（4）关闭设备

2.2. 例子使用

具体使用可以参照 ql-ol-sdk/ql-ol-extsdk/example/audio/ example_audio.c

```
root@mdm9607-perf:~# ./example_audio

--Useage:
play one file: ./<process> play1 <file>
play two file: ./<process> play2 <file1> <file2>
recd and play: ./<process> recd1
recd and save: ./<process> recd2 <file>
play     tone: ./<process> tone [<freq> <time> <volume>]
root@mdm9607-perf:~# ./example_audio play1 demo.wav
get wav hdr offset
Ql_clt_set_mixer_value, device: SEC_AUX_PCM_RX Audio Mixer MultiMedia1, value: 1
Ql_clt_set_mixer_value, set mixer: SEC_AUX_PCM_RX Audio Mixer MultiMedia1 sucess
Ql_clt_set_mixer_value, device: MultiMedia1 Mixer SEC_AUX_PCM_UL_TX, value: 1
Ql_clt_set_mixer_value, set mixer: MultiMedia1 Mixer SEC_AUX_PCM_UL_TX sucess
__ql_pcm_setParams, 229
buffer_bytes = (1024,1024) omin=0 omax=0 int=1 empty=0
period_bytes = (128,128) omin=0 omax=0 int=1 empty=0
create play thread...
Ql_cb_Player1: hdl=4, result=0
[4]Start write data to audio device
__ql_playback_proc[4]: play data, cnt=0, size=128
__ql_playback_proc[4]: play data, cnt=1, size=128
__ql_playback_proc[4]: play data, cnt=2, size=128
__ql_playback_proc[4]: play data, cnt=3, size=128
__ql_playback_proc[4]: play data, cnt=4, size=128
```

3 录音

3.1. 编程步骤

（1）打开设备。调用 Ql_AudRecorder_Open（）
（2）录音。调用 Ql_AudRecorder_StartRecord（）
（3）关闭设备

3.2. 例子使用

具体使用可以参照 ql-ol-sdk/ql-ol-extsdk/example/audio/ example_audio.c

```
root@mdm9607-perf:~# ./example_audio

--Useage:
play one file: ./<process> play1 <file>
play two file: ./<process> play2 <file1> <file2>
recd and play: ./<process> recd1
recd and save: ./<process> recd2 <file>
play    tone:  ./<process> tone [<freq> <time> <volume>]
root@mdm9607-perf:~# ./example_audio recd1
Ql_clt_set_mixer_value, device: SEC_AUX_PCM_RX Audio Mixer MultiMedia1, value: 1
Ql_clt_set_mixer_value, set mixer: SEC_AUX_PCM_RX Audio Mixer MultiMedia1 sucess
Ql_clt_set_mixer_value, device: MultiMedia1 Mixer SEC_AUX_PCM_UL_TX, value: 1
Ql_clt_set_mixer_value, set mixer: MultiMedia1 Mixer SEC_AUX_PCM_UL_TX sucess
<pcm_open, 0x1e040>
__ql_pcm_setParams, 229
buffer_bytes = (2560,2560) omin=0 omax=0 int=1 empty=0
period_bytes = (320,320) omin=0 omax=0 int=1 empty=0
create capature thread...
Start read data from audio device
__ql_capature_proc: cap data, cnt=0, size=320
Ql_cb_Recd1: save rocord...
__ql_capature_proc: cap data, cnt=1, size=320
Ql_cb_Recd1: save rocord...
__ql_capature_proc: cap data, cnt=2, size=320
Ql_cb_Recd1: save rocord...
__ql_capature_proc: cap data, cnt=3, size=320
```

4 Tone 音

4.1. 编程步骤

（1）打开设备
（2）播放 tone 音
（3）关闭设备

4.2. 例子使用