MDM9x07&MDM9628 Audio API Guide Manual

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

This document applies to MDM9628 and MDM9X07 platforms.

History

Revision	Date	Author	Description
1.0	2017-11-15	Running	Initial

Contents

1. QuecOpen Audio API Introduce

QuecOpen provides a set of audio interfaces for playback and recording, all of which are PCM data streams.

1.1 Play API

（1）Call back function

```
// 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);
```

（2）open and close the device

```
/*****************************************************************
* 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);
```

（3） **playing**

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

（3） **control**

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

1.2 Record API

（1） **open and close device**

```
/****************************************************************
* 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);
```

（2） **recording**

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

（3） **control**

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

1.3 Tone API

```
/*******************************************
 * 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. Play

2.1 Program Steps

（1）Open the device. If you want to support remix, you need to open two devices.

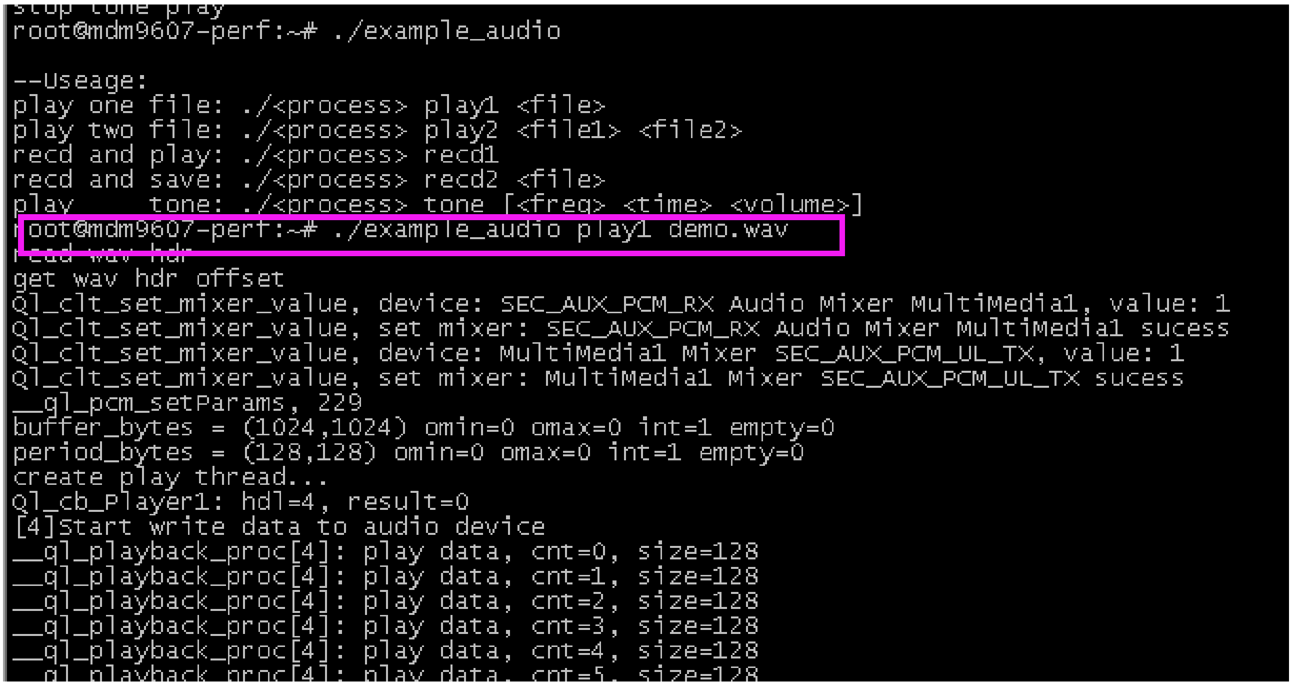
（2）If the PCM data is in the buff, call Ql_AudPlayer_Play() directly；

（3）If the PCM data is in the file, call Ql_AudPlayer_PlayFrmFile() directly

（4）close device

2.2 Demo

Please refer to ql-ol-sdk/ql-ol-extsdk/example/audio/ example_audio.c



```
stop tone play
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
read wav hdr
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
__ql_playback_proc[4]: play data, cnt=5, size=128
```

3. Record

3.1 Program Steps

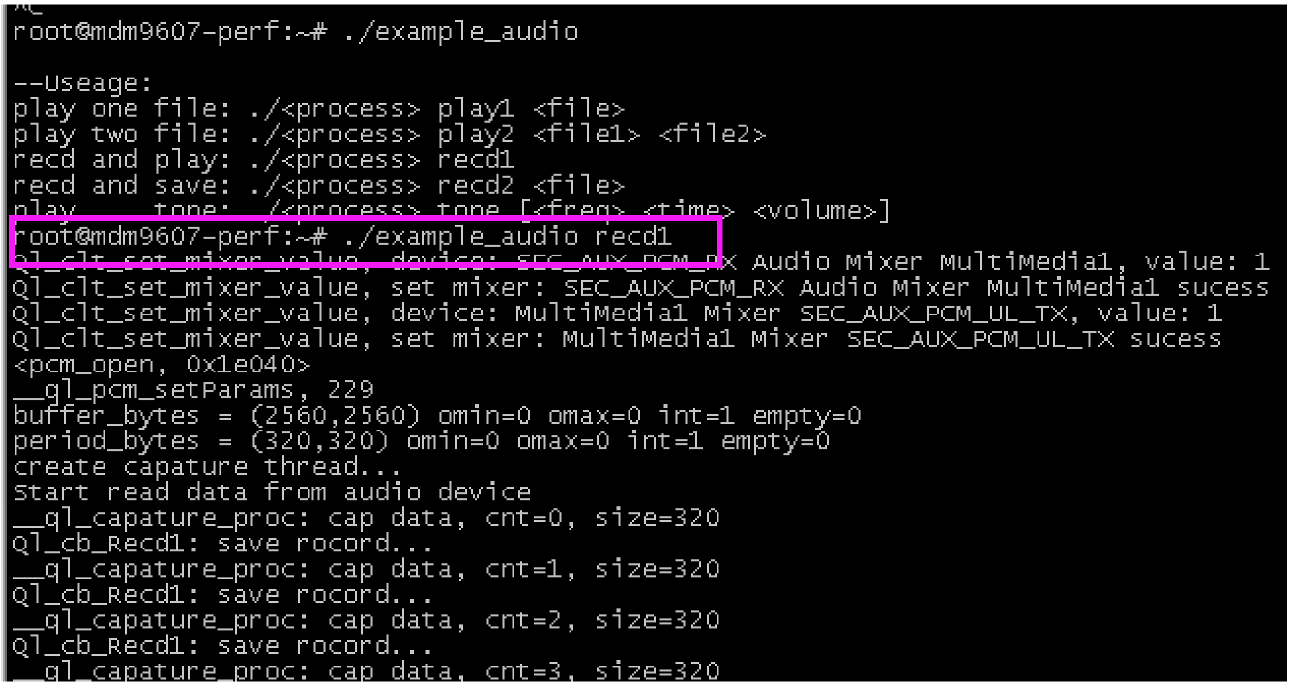
（1）open device, call Ql_AudRecorder_Open（）

（2）recording, call Ql_AudRecorder_StartRecord（）

（3）close device

3.2 demo

Please refer to ql-ol-sdk/ql-ol-extsdk/example/audio/ example_audio.c



```
^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

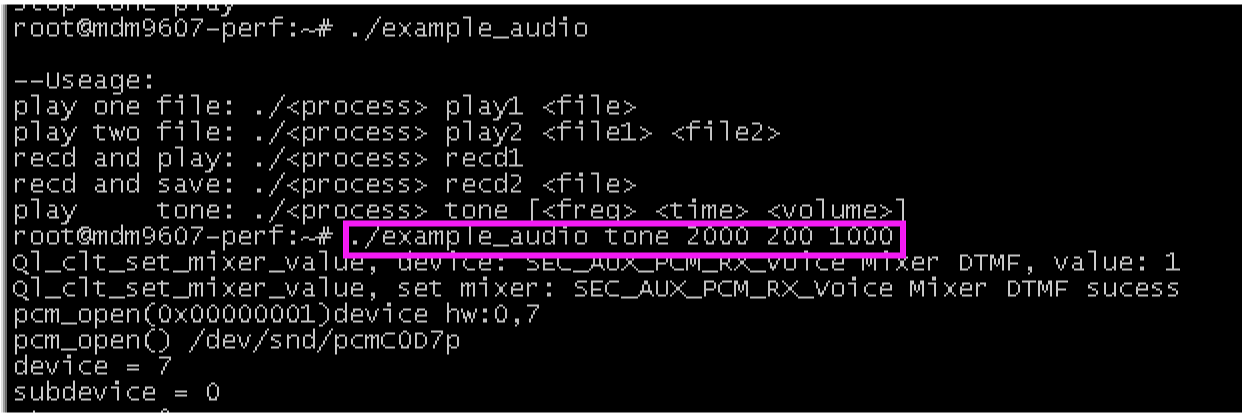
4.1 Program Steps

（1）open device

（2）play tone

（3）close device

4.2 demo



```
stop tone play
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 tone 2000 200 1000
Ql_clt_set_mixer_value, device: SEC_AUX_PCM_RX_Voice Mixer DTMF, value: 1
Ql_clt_set_mixer_value, set mixer: SEC_AUX_PCM_RX_Voice Mixer DTMF sucess
pcm_open(0x00000001)device hw:0,7
pcm_open() /dev/snd/pcmC0D7p
device = 7
subdevice = 0
```