TinaRecorder开发指南

1. TinaRecorder状态机



1. TinaRecorder结构图



3.TinaRecorder状态机简介

Init：初始状态，当使用 CreateRecorder 方法创建一个TinaRecorder handle并调用了TRreset()方法时，该TinaRecorder对象处于Init状态。在设定摄像头输入和麦克风输入之后将转换为Initialized状态。另外，在除Released状态外的其它状态通过调用reset()方法都可以使TinaRecorder进入该状态。

Initialized：已初始化状态，可以通过在Init状态调用TRsetCamera()或setAudioSrc()方法进入该状态。在这个状态可以通过TRsetPreview()和TRSetOutput()方法设置输出参数，此时TinaRecorder转换为DataSourceConfigured状态。另外，通过TRreset()方法进入Init状态。

DataSourceConfigured：数据源配置状态，这期间可以设定编码方式、输出文件、屏幕旋转、预览显示等等。可以在Initialized状态通过TRsetPreview()和TRSetOutput()方法进入该状态。另外，可以通过TRreset()方法回到Init状态，或者通过TRprepare()方法到达Prepared状态。

Prepared：就绪状态，在DataSourceConfigured状态通过TRprepare()方法进入该状态。在这个状态可以通过TRstart()进入录制或者预览状态。另外，可以通过TRreset()方法回到Initialized状态。

Recording/Previewing：录制状态，可以在Prepared状态通过调用TRstart()方法进入该状态。另外，它可以通过TRstop(hdl,STOP\_ALL)方法或reset()方法回到Init状态。如果调用TRstop(hdl,TR\_PREVIEW)或者TRstop(hdl,TR\_RECORD),可以单独关掉预览或者录制，单独关掉预览或录制时状态保持不变，当检测到预览和录制都被关掉时，会进入Init状态

Released：释放状态，可以通过在Initial状态调用TRrelease()方法来进入这个状态，这时将会释放所有TinaRecorder对象绑定的资源。

Error：错误状态，当错误发生的时候进入这个状态，它可以通过TRreset()方法进入Init状态。

提示：TinaRecorder录音录像时需要严格遵守状态图说明中的函数调用先后顺序，在不同的状态调用不同的函数，否则会出现异常。

4.TinaRecorder函数简介

（1）返回值：除特殊提示0为成功，-1为失败

（2）函数详细说明

MediaRecorderHandle \*CreateRecorder();

创建一个TinaRecorder,返回值为TinaRecorder的handle,错误时返回NULL

 int TRsetOutput(MediaRecorderHandle \*hdl,char \*url);

 设置存放文件的文件夹路径，录制时会自动根据文件夹位置存放文件。若设为NULL，则不开启录制功能。

 int TRsetCamera(MediaRecorderHandle \*hdl,int index);

 设置此路录制器数据源的camera的索引（前置还是后置），如果设置为TR\_CAMERA\_DISABLE，则屏蔽掉视频源数据，可以用来录制音频等操作。

 int TRsetAudioSrc(MediaRecorderHandle \*hdl,int index);

 设置此路录制器数据源的MIC索引（MIC0,MIC1)等，如不需要录制声音数据，可以设为TR\_AUDIO\_DISABLE

int TRsetPreview(MediaRecorderHandle \*hdl,int layer);

 设置preview的图层(disp layer0,disp layer1)。在camera设为TR\_CAMERA\_DISABLE时，对此函数的非DISABLE设置会返回失败

int TRstart(MediaRecorderHandle \*hdl);

 开始录制

int TRstop(MediaRecorderHandle \*hdl,int flags);

 停止录制或者预览，通过flags控制可以在同时录制和预览的情况下单独停掉录制或者预览，并保持状态机为Recording/Previewing。如果检测到录制和预览已经同时被关掉，状态机会进入Init状态

int TRrelease(MediaRecorderHandle \*hdl);

 在Init状态调用以释放资源

int TRprepare(MediaRecorderHandle \*hdl);

 准备TinaRecorder

int TRreset(MediaRecorderHandle \*hdl);

 重置TinaRecorder

int Trresume(MediaRecorderHandle \*hdl);

 必须在Recording/Previewing状态下调用，用于恢复被单独关掉的预览或者录制通路

int TRsetVideoEncodeSize(MediaRecorderHandle \*hdl,int width,int height);

 设置编码器的编码大小，如果此大小和摄像头输出不一致，会进行插值或者SCALE

int TRsetOutputFormat(MediaRecorderHandle \*hdl,int format);

 设置输出文件格式，此函数会判断AV source的配置情况返回成功或者失败（例如设为AAC格式，则Camera必须关掉而麦克风打开）

int TRsetRecorderCallback(MediaRecorderHandle \*hdl,Rcallback callback);

 设置TinaRecorder消息回调

int TRsetMaxRecordTimeMs(MediaRecorderHandle \*hdl,int ms);

 设置编码最长时间，在达到最长时间后，编码器会在指定的文件路径上编码下一个文件

int TRsetVideoEncoderFormat(MediaRecorderHandle \*hdl,int Vformat);

 设置编码后视频格式

int TRsetAudioEncoderFormat(MediaRecorderHandle \*hdl,int Aformat);

 设置编码后的音频格式

int TRsetEncoderBitRate(MediaRecorderHandle \*hdl,int bitrate);

 设置编码后的比特率

int TRsetRecorderEnable(MediaRecorderHandle \*hdl,int enable);

 在Recording/Previewing状态下调用，用于丢弃编码完成的数据

int TRsetEncodeFramerate(MediaRecorderHandle \*hdl,int framerate);

 设置编码后的framerate

int TRgetRecordTime(MediaRecorderHandle \*hdl);

 在Recording/Previewing状态使用，返回当前的编码时间，如果编码通路没有打开或者其它错误会返回-1

int TRCaptureCurrent(MediaRecorderHandle \*hdl,char \*path,int format);

 在Recording/Previewing状态下使用抓图，path为图片保存位置，format为保存格式

int TRsetCameraInputFormat(MediaRecorderHandle \*hdl,int format);

 设置摄像头输入格式

int TRsetCameraFramerate(MediaRecorderHandle \*hdl,int framerate);

 设置摄像头输入帧率

int TRsetCameraCaptureSize(MediaRecorderHandle \*hdl,int width,int height);

 设置摄像头输入图像大小

int TRsetCameraDiscardRatio(MediaRecorderHandle \*hdl,int ratio);

 设置摄像头丢帧比例

int TRsetCameraWaterMarkIndex(MediaRecorderHandle \*hdl,int id);

 设置摄像头水印的ID

int TRsetCameraEnable(MediaRecorderHandle \*hdl,int enable);

 在Recording/Previewing状态使用，清空摄像头数据（填黑）

int TRsetMICInputFormat(MediaRecorderHandle \*hdl,int format);

 设置麦克风输入格式

int TRsetMICSampleRate(MediaRecorderHandle \*hdl,int sampleRate);

 设置麦克风输入sample rate

int TRsetMICChannels(MediaRecorderHandle \*hdl,int channels);

 设置麦克风输入channel数

int TRsetMICBitrate(MediaRecorderHandle \*hdl,int bitrate);

 设置麦克风输入比特率

int TRsetMICEnable(MediaRecorderHandle \*hdl,int enable);

 在Recording/Previewing状态使用，清空音频数据（填0）

int TRsetPreviewRect(MediaRecorderHandle \*hdl,dispRect \*rect);

 设置预览的窗口位置和大小

int TRsetPreviewRotate(MediaRecorderHandle \*hdl,int angle);

 设置预览窗口的旋转角度

int TRsetPreviewEnable(MediaRecorderHandle \*hdl,int enable);

 关闭或者打开preview图层