

基于 SDK 的视频压缩卡操作

一、实验目的：

学习在 Windows 下利用厂商提供的开发包进行视频捕捉卡开发的基本方法。

二、实验原理

由于使用厂商提供的开发包进行程序开发，下面介绍开发包中提供的函数。

与 AV800 初始化相关的函数

BOOL TS_InitDevice (HWND hWnd)

功能：初始化AV800 API 软件组件并建立视频窗。建立后AV800 处于Preview 方式。

参数：HWND hWnd 视频窗的窗口句柄

返回值：TRUE 成功，FALSE 失败

void TS_UnInitDevice()

功能：清除AV800 API 软件组件。

参数：无

返回值：无

int TS_GetState ()

功能：获得AV800 当前状态。

参数：无

返回值：DEVICE_STATE_PREVIEW, 处于Preview 状态。

DEVICE_STATE_CAPTURE, 处于采集MPEG 状态。

DEVICE_STATE_PLAYBACK, 处于播放MPEG 文件状态。

与视频显示相关的函数

void TS_SetVideoWindowPos (int x, int y, int cx, int cy)

功能：设置视频窗的位置及大小。

参数：int x 视频窗的左上角屏幕坐标，

int y 视频窗的右上角屏幕坐标，

int cx 视频窗的宽度，

int cy 视频窗的高度

返回值：无

void TS_TV Tuning()

功能：TV Tuner 自动调谐。（仅适用于AV800/TV）

参数：无

返回值: 无

int TS_GetTVChannel()

功能: 获得当前选择的电视频道号。(仅适用于AV800/TV)

参数: 无

返回值: 0 - 100

void TS_SetTVChannel(int Channel)

功能: 设置电视频道号。(仅适用于AV800/TV)

参数: int Channel: 0 -100

返回值: 无

int TS_GetVideoSource()

功能: 获得当前选择的输入视频源

参数: 无

返回值: SOURCE_TVTUNER 输入源为TV Tuner。(仅对AV800/TV 有效)

SOURCE_COMPOSITE 输入源为复合视频。

SOURCE_SVIDEO 输入源为S-Video

void TS_SetVideoSource(int VideoSource)

功能: 设置视频输入源

参数: int VideoSource 的取值分别为:

SOURCE_TVTUNER 输入源为TV Tuner。(仅对AV800/TV 有效)

SOURCE_COMPOSITE 输入源为复合视频。

SOURCE_SVIDEO 输入源为S-Video

返回值: 无

int TS_GetVideoSystem ()

功能: 获得当前输入视频制式

参数: 无

返回值: 1: PAL

2: NTSC

void TS_SetVideoSystem(int VideoSystem)

功能: 设置视频输入输出制式

参数: int VideoSystem 的取值分别为:

1: PAL

2: NTSC

返回值: 无

注: 对于输入视频制式, AV800 由硬件自动检测, 即使设置了不同制式, 在采集时仍然会根据实际输入信号的制式编码MPEG。AV800 的视频输出制式可以由此函数设置。

BOOL TS_GetDisplayUseOverlay()

功能: 获得当前视频显示是否使用了DirectX Overlay。

参数: TRUE 使用Overlay

FALSE 未使用Overlay

返回值: 无

void TS_SetDisplayUseOverlay(BOOL bOverlay)

功能: 设置视频显示是否使用DirectX Overlay。

参数: BOOL bOverlay: TRUE 使用Overlay, FALSE 不使用Overlay

返回值: 无

int TS_GetBrightness()

功能：获得当前设置的亮度值。

参数： 无

返回值： BRIGHTNESS_MIN 至 BRIGHTNESS_MAX

void TS_SetBrightness(int nBrightness)

功能：设置亮度值

参数： int nBrightness: BRIGHTNESS_MIN 至 BRIGHTNESS_MAX

返回值： 无

int TS_GetContrast()

功能：获得当前设置的对比度值。

参数： 无

返回值： CONTRAST_MIN 至 CONTRAST_MAX

void TS_SetContrast(int nContrast)

功能：设置对比度值

参数： int nContrast: CONTRAST_MIN 至 CONTRAST_MAX

返回值： 无

int TS_GetHue()

功能：获得当前设置的色度值

参数： 无

返回值： HUE_MIN 至 HUE_MAX

void TS_SetHue(int nHue)

功能：设置色度值

参数： int nHue: HUE_MIN 至 HUE_MAX

返回值： 无

int TS_GetSaturation()

功能：获得当前设置的色饱和度值

参数： 无

返回值： SATURATION_MIN 至 SATURATION_MAX

void TS_SetSaturation(int nSaturation)

功能：设置色饱和度值

参数： int nSaturation: SATURATION_MIN 至 SATURATION_MAX

返回值： 无

int TS_GetSharpness()

功能：获得当前设置的色锐度值

参数： 无

返回值： SHARPNESS_MIN 至 SHARPNESS_MAX

void TS_SetSharpness(int nSharpness)

功能：设置色锐度值

参数： int nSharpness: SHARPNESS_MIN 至 SHARPNESS_MAX

返回值： 无

与 MPEG 编码相关的函数

AV800 的MPEG 编码参数包括：

如果您在阅读过程中发现疏漏和错误，请您尽快和编者取得联系 network@ustc.edu.cn cxh@ustc.edu.cn

Video Bitrate: 视频位速率, 可以通过**TS_SetVideoBitRate** 和 **TS_GetVideoBitRate** 改变或查询。有效值为512000 - 15000000。

GOP Size: MPEG GOP (Group Of Picture)的I 帧间隔, 通过**TS_SetGOPSize** 和**TS_GetGOPSize** 改变或查询。有效值为1 - 15 并且必须是P 帧间隔的整倍数, 一般取值为12 或15。

P Frame Distance: MPEG GOP 中的P 帧间隔, 通过**TS_SetPDistance** 和 **TS_SetPDistance** 改变或查询。有效值为1 - 3。

Rate Control: 这是关于MPEG 的编码方式, 可供选择的是CBR(Constant Bitrate) 和VBR(Variable BitRate), 通过**TS_SetRateCtrl** 和 **TS_GetRateCtrl** 改变或查询。有效值为0: CBR, 1: VBR。

TopFieldFirst: 用于在解码时通知解码器首先解码哪一场图像, 有效值为:TRUE 表示先解码A Field,FALSE 表示先解码B Field。可以通过 **TS_SetTopFieldFirst** 和**TS_GetTopFieldFirst** 改变或查询, 一般情况下应设置为TRUE。

HHRfilter: HHR(Half Horizontal Resolution) 用于指出当水平分辨率为352 或480 时对输入图像分辨率进行变换, 可以通过**TS_SetHHR** 和 **TS_GetHHR** 改变或查询。一般说, 当采集DVD 时禁止HHR, 当采集 VCD, SVCD 时应允许HHR。有效值为TRUE:允许, FALSE:禁止。

OffsetField1: 第一场第一条扫描线的位移量。有效值为/0 - 31, 目前固定设置为2。尽管这里提供了**TS_SetOffsetField1** 和 **TS_SetOffsetField1** 但用户最好不要改变它, 否则将影响到视频播放时的图像稳定性。

OffsetField2: 第二场第一条扫描线的位移量。有效值为/0 - 31, 目前固定设置为2。尽管这里提供了**TS_SetOffsetField2** 和 **TS_SetOffsetField2** 但用户最好不要改变它, 否则将影响到视频播放时的图像稳定性。

Resolution: MPEG 编码分辨率。通过**TS_SetResolution** 和**TS_SetResolution** 改变或查询, 有效值为:
0: CIF(SIF) 352 x 288 或352 x 240, 用于VCD/MPEG-1 编码
1: 1/2 D1 352 x 576 或352 x 480, 用于CVD/MPEG-2 编码
2: 2/3 D1 480 x 576 或480 x 480, 用于SVCD/MPEG-2 编码
3: Full D1 704 x 576 或704 x 480, 用于DVD/MPEG-2 编码

Prefilterno: 预处理滤波器编号。通过**TS_SetPreFilterNo** 和 **TS_GetPreFilterNo** 改变或查询, 有效值为0 - 7。

Int TS_GetTotalEncodeMode()
功能: 获得预定义编码参数组个数。为了方便用户操作目前AV800 D11 预置了5 组MPEG 编码参数。分别为: VCD, SVCD, DVD, MPEG-1, MPEG-2。在下一个版本将增加用户自定义参数组功能。
参数: 无
返回值: 预定义编码参数组个数

Int TS_GetEncodeMode()
功能: 获得当前的参数组索引。
参数: 无

返回值: 当前参数组索引

void TS_SetEncodeMode(int nIndex)

功能: 改变参数组。

参数: int nIndex: 参数组索引, 应大于0 并小于**TS_GetTotalEncodeMode** 的返回值。

返回值: 无

void TS_GetEncodeModeName(int nIndex, CString & EncModeName)

功能: 获得某参数组名称。

参数: int nIndex: 参数组索引。

CString & EncModeName: 通过此参数返回指定参数组名称。

返回值: 无

void TS_SetEncodeModeName(int nIndex, CString EncModeName)

功能:

参数: int nIndex: 参数组索引。

CString & EncModeName: 参数组名称。

返回值: 无

void TS_ReadEncodeParameters(int nIndex)

功能: 获得指定参数组的参数值。在调用此函数后可以通过**TS_GetVideoBitRate** 等函数获得每一个参数的当前值。在调用**TS_InitDevice** 时**TS_InitDevice** 会以最近一次调用**TS_SetEncodeMode** 设置的参数组索引调用**TS_ReadEncodeParameters**, 所以在调用**TS_InitDevice** 后可以直接通过**TS_GetVideoBitRate** 等函数获得每一个参数的当前值。

参数: int nIndex: 参数组索引。

返回值: 无

void TS_UpdateEncodeParameter()

功能: 在调用**TS_GetVideoBitRate** 等函数改变了MPEG 编码参数后, **TS_SetEncodeMode** 通知

AV800 D11 更新参数值。

参数: 无

返回值: 无

int TS_GetGOPSize()

功能: 参考前面关于AV800 的MPEG 编码参数说明。

参数:

返回值:

void TS_SetGOPSize(int nGOPSize)

功能: 参考前面关于AV800 的MPEG 编码参数说明。

参数:

返回值:

BOOL TS_GetHHR()

功能: 参考前面关于AV800 的MPEG 编码参数说明。

参数:

返回值:

void TS_SetHHR(BOOL bHHR)

功能: 参考前面关于AV800 的MPEG 编码参数说明。

参数:

返回值:

int TS_GetOffsetField1()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetOffsetField1(int nField1Offset)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetOffsetField2()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetOffsetField2(int nField2Offset)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetPDistance()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetPDistance(int nPDistance)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetPreFilterNo()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetPreFilterNo(int nPreFilterNo)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetRateCtrl()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetRateCtrl(int nRateCtrl)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetResolution()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetResolution(int nResolution)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

BOOL TS_GetTopFieldFirst()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetTopFieldFirst(BOOL bNewValue)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

long TS_GetVideoBitRate()

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

void TS_SetVideoBitRate(long nVideoBitrate)

功能：参考前面关于AV800 的MPEG 编码参数说明。

参数：

返回值：

int TS_GetAudioBitRate()

功能：获得当前设置的MPEG Audio Bitrate。

参数： 无

返回值：当前AV800 支持的MPEG 音频位速率为128KHz, 224KHz, 384KHz。

TS_GetAudioBitRate()

返回的是音频为速率的索引值，即：0 表示128K, 1 表示224K, 2 表示384K。

void TS_SetAudioBitRate(long nAudioBitrateIndex)

功能：设置MPEG 音频编码位速率。

参数： long nAudioBitrateIndex：有效值为：

0 表示128KHz

1 表示224KHz

2 表示384KHz

返回值：无

采集、播放控制函数

BOOL TS_StartCapture(int StreamMode, LPCTSTR FilePath, long CallbackAddress)

功能：调用此函数启动采集MPEG，可以通过此函数采集MPEG 文件或MPEG 数据流，或二者同时。

参数：int StreamMode： STREAM_TO_FILE 表示采集到文件，第二个参数为文件名，第三个参数为NULL。

STREAM_TO_BUFFER 表示采集MPEG 流。这时第二个参数为NULL, 第三个参数为回调函数地址。

STREAM_TO_FILE | STREAM_TO_BUFFER 表示同时采集文件和MPEG

流。

LPCTSTR FilePath: 文件路径。

long CallbackAddress: 回调函数地址

返回值: TRUE:成功, FALSE:出错。

*注: 回调函数定义为: typedef DWORD (CALLBACK *MD_CALLBACK) (BYTE bMsg, DWORD dwParam1,*

DWORD dwParam2)。第一参数hMsg 为回调时传递的消息, 有效的消息有三个, 分别为: MD_CODE_CREATE 这是第一次回调时传递的消息, 表明数据开始, 这时dwParam1 和 dwParam2 无意义

MD_CODE_WRITE 此消息表明传递给回调函数的第二个参数为MPEG 数据缓冲区地址, 第三个参数为数据长度

MD_CODE_CLOSE 当停止采集(通过调用TS_StopCapture)时传递此消息。

BOOL TS_StartVidCapture(int StreamMode, LPCTSTR FilePath, long CallbackAddress)

功能: 调用此函数启动采集MPEG 视频, 可以通过此函数采集MPEG 视频文件或MPEG 视频数据

流, 或二者同时。

参数: int StreamMode: STREAM_TO_FILE 表示采集到文件, 第二个参数为文件名, 第三个参数为NULL。

STREAM_TO_BUFFER 表示采集MPEG 流。这时第二个参数为NULL, 第

三个参数为回调函数地址。

STREAM_TO_FILE | STREAM_TO_BUFFER 表示同时采集文件和MPEG

流。

LPCTSTR FilePath: 文件路径。

long CallbackAddress: 回调函数地址

返回值: TRUE:成功, FALSE:出错。

注: 此函数仅仅采集MPEG 视频流。

BOOL TS_StartAudCapture(int StreamMode, LPCTSTR FilePath, long CallbackAddress)

功能: 调用此函数启动采集MPEG 音频, 可以通过此函数采集MPEG 音频文件或MPEG 音频数据

流, 或二者同时。

参数: int StreamMode: STREAM_TO_FILE 表示采集到文件, 第二个参数为文件名, 第三个参数为NULL。

STREAM_TO_BUFFER 表示采集MPEG 流。这时第二个参数为NULL, 第

三个参数为回调函数地址。

STREAM_TO_FILE | STREAM_TO_BUFFER 表示同时采集文件和MPEG

流。

LPCTSTR FilePath: 文件路径。

long CallbackAddress: 回调函数地址

返回值: TRUE:成功, FALSE:出错。

注: 此函数仅仅采集MPEG 音频流, 编码方式为MPEG-1 Audio Layer-II。

BOOL TS_StartAVCapture(LPCTSTR VidFilePath, LPCTSTR AudFilePath)

功能: 调用此函数启动采集MPEG 并分离为两个文件, 一个为MPEG 视频文件, 一个为MPEG 音频文件。

参数: LPCTSTR VidFilePath: MPEG 视频文件路径。

LPCTSTR AudFilePath: MPEG 音频文件路径。

返回值: TRUE:成功, FALSE:出错。

注: 此函数不支持回调。

BOOL TS_StopCapture()

功能: 停止采集。

参数: 无

返回值: TRUE:成功, FALSE:出错。

BOOL TS_StartPlayback(LPCTSTR FilePath)

功能: 播放MPEG 文件。

参数: LPCTSTR FilePath: MPEG 文件名路径。

返回值: TRUE:成功, FALSE:出错。

BOOL TS_PausePlayback()

功能: 暂停播放MPEG 文件。

参数:

返回值: TRUE:成功, FALSE:出错。

BOOL TS_ResumePlayback()

功能: 恢复播放MPEG 文件。

参数:

返回值: TRUE:成功, FALSE:出错。

BOOL TS_StopPlayback()

功能: 停止播放MPEG 文件。

参数:

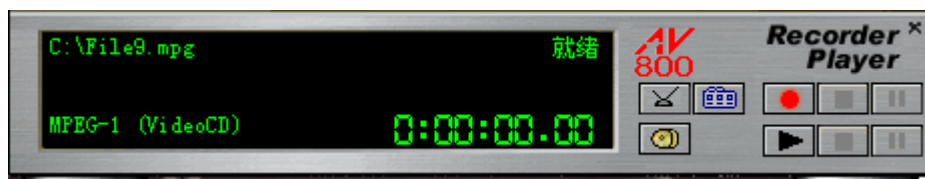
返回值: TRUE:成功, FALSE:出错。

三、实验要求:

1 操作视频捕捉卡

在菜单中选择“Video Catcher 3.0”程序并运行。采集一段压缩图像, 并改变以下 MPEG 压缩设置, 对不同参数设置的情况进行对比。





2 视频压缩卡编程

请同学们参照样本程序，学习利用开发包进行视频设备编程的基本方法。要求自己制作一个程序（可以用 VC、Delphi），显示由摄像头摄入的影像，并能够将抓取的图象和动画保存到硬盘上。本实验的编程很简单，请不要抄袭样本程序。对具有独特风格的程序，将给予加分。

四、思考题

- 1、 视频采集卡和视频压缩卡的主要区别是什么？
- 2、 在介绍开发包函数的时候，进行了分类，谈谈个人对此型号压缩卡和提供的开发包有什么建议？