DS-4000/4100/4200/4300 系列

视音频压缩/解码板卡

系统 SDK 编程指南

(for Windows 7/2008/Vista/2003/XP)

V6.5

杭州海康威视数字技术股份有限公司
http://www.hikvision.com
技术热线：400-700-5998
UD.6L0102F0236A01

声 明

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

- 本手册适用于 DS-4000/4100/4200/4300 系列视音频压缩/解码板卡。
- 我们已尽量保证手册内容的完整性与准确性，但也不免出现技术上不准确、与产品功能及操作不相符或印刷错误等情况出现，如有任何疑问或争议，请以海康威视最终解释为准。
- 产品和手册将实时进行更新，恕不另行通知。
- 本手册中内容仅为用户提供参考指导作用，不保证与实物完全一致，请以实物为准。

0605001030121

目 录

第1章 产品简介

DS-4000/4100/4200/4300 系列是面向数码监控行业而推出的专用板卡，采用了高性能的视频压缩技术标准 H.264 及 OggVorbis、G.711 的音频编码标准，完全依靠硬件实现了视频及音频的实时编码或解码并精确同步，实现了动态码率、可控帧率、帧模式选择、动态图像质量控制，音频预览、视频丢失报警等功能，能独立调整各通道参数，性能稳定而且可靠。与 MPEG-I 产品相比，在保持同等图像质量的前提下，能大大节省存储空间、并非常适合宽带网或窄带网的传输，是新一代数字监控产品的最佳选择。

DS-4000/4100/4200/4300 系列板卡 SDK 分为三部分，分别为系统 SDK、网络 SDK、播放 SDK，本文档专门描述系统 SDK，其他 SDK 请参照我公司相关文档。系统 SDK 是专门为该系列板卡设计的本地录像及解码软件接口程序，以动态连接库的形式提供给应用软件开发者，并同时附有演示程序(System Demo、Decode File Demo)及其源码，能有效地缩短应用软件的开发周期。

在使用过程中，特别提醒软件开发人员，DS-4000/4100/4200/4300 系列编码卡可以在编码的同时修改除码流类型(复合流、纯视频流、音频流)外的所有参数，包括分辨率、码流、帧结构等。譬如在压缩过程中可改变帧率(SetIBPMode)、量化系数(SetDefaultQuant)、分辨率、码流、帧结构而无须停止、启动压缩。播放器会自动识别帧率、分辨率等参数，按当前压缩帧率、分辨率播放且声音图像播放保持正常。

通过动态修改量化系数(I、B、P)可控制压缩码率，当码率太高时，加大量化系数；当码率太低时，减少量化系数。当然，在量化系数满足的情况下，不必再降低量化系数。

DS-4000/4100/4200/4300 系列编码卡的运动检测独立于压缩，不进行压缩也可以进行运动检测。可动态改变帧率非常有价值，在无运动时按低帧率录像，运动时按高帧率录像，记录在同一个文件内，可大大节省硬盘空间。

1.1 DS-43xx 系列

DS-43xx 系列板卡包含 DS-4308HCV-E、DS-4316HCV-E、DS-4308HFV-E、DS-4316HFV-E、DS-4304HFH-E、DS-4308HW-E、DS-4316HW-E、DS-4304HD-E、DS-4308MD-E 等型号，均采用 PCI Express X1 的接口，与传统的 PCI 接口相比，大大提高了数据带宽。

DS-4308HCV-E，主通道和子通道支持设置 4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 8 路 CIF 主通道实时编码和 8 路 QCIF 子通道实时编码。

DS-4316HCV-E，主通道和子通道支持设置 4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 16 路 CIF 主通道实时编码和 16 路 QCIF 子通道实时编码。

DS-4308HFV-E，主通道和子通道都能支持设置 4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 8 路 4CIF 主通道实时编码和 8 路 CIF 子通道实时编码。

DS-4316HFV-E，主通道和子通道都能支持设置 4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 16 路 4CIF 主通道实时编码和 16 路 CIF 子通道实时编码。

DS-4304HFH-E，主通道和子通道都能支持设置编码分辨率 1920×1080、1920×540、960×540、1280×720、1280×360、640×360、4CIF、2CIF、CIF 以及 QCIF。采用双码流编码时，最高支持 4 路 1080P 主通道实时编码和 4 路 4CIF 子通道实时编码。同时支持 1 路 HDMI 视音频输出，输出分辨率支持 1080P_50HZ、

1080P_60HZ、720P_50HZ、720P_60HZ、XGA_60HZ、SXGA_960_60HZ，且支持画面分割，最多支持 16 个显示区域。

DS-4308HW-E，主通道和子通道都能支持设置 WD1/4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 8 路 WD1 主通道实时编码和 8 路 CIF 子通道实时编码。

DS-4316HW-E，主通道和子通道都能支持设置 WD1/4CIF/2CIF/CIF/QCIF 的编码分辨率。同时支持 16 路 WD1 主通道实时编码和 16 路 CIF 子通道实时编码。

DS-4304HD-E 支持 2 路 500 万（2560*1920）或者 4 路 1080P（1920*1080P）或者 8 路 720P（1280*720P）或者 16 路 4CIF（标准编码）及以下分辨率实时解码（10 路 4CIF 海康私有编码码流实时解码），同时支持 4 路 HDMI 输出(音视频)，同时还支持 4 路音频 BNC 输出。HDMI 音视频输出的分辨率支持 1080P_50HZ、1080P_60HZ、720P_50HZ、720P_60HZ、XGA_60HZ、SXGA_960_60HZ，且支持画面分割，最多支持 16 个显示区域。

DS-4308MD-E 非实时解码最大支持 16 路 WD1/ 800*600/4CIF 以及以下分辨率码流；实时解码最高支持 11 路 WD1/800*600 分辨率标准 H.264 码流、或者 13 路 4CIF 标准 H.264 码流、或者 16 路 4CIF 标准 MPEG4 码流、或者 16 路 2CIF/DCIF/CIF/QCIF 码流。

1.2 DS-42xx 系列

DS-42xx 系列板卡包含 DS-4216HC、DS-4208HFV、DS-4216HFV 3 种型号，性能稳定，功耗低。

DS-4216HC 提供 16 路 4CIF/ DCF/2CIF 非实时编码或者 16 路 CIF/QCIF 实时编码，同时支持 16 路 CIF/QCIF 子通道编码；

DS-4208HFV 提供 8 路 4CIF/DCF/2CIF/CIF/QCIF 实时编码，同时支持 8 路 CIF/QCIF 子通道编码；

DS-4216HFV 提供 16 路 4CIF/ DCF/2CIF/CIF/QCIF 实时编码，同时支持 16 路 CIF/QCIF 子通道编码。

1.3 DS-41xx 系列

DS-41xx 系列板卡包含 DS-4101HD 高清解码卡和 DS-4108HCV、DS-4116HCV 2 种型号编码卡：

DS-4101HD 是面向数码监控行业而推出的专用高清音视频解码板卡，最高支持 1 路 1080P 高清解码，提供种类丰富的输出接口(HDMI/DVI/VGA/YPbPr/BNC)，适用于多种码流格式、多种编解码算法的解码。

DS-4108HCV 包含 1 个 DSP，DS-4116HCV 包含 2 个 DSP， 每个 DSP 支持 8 路 DCIF/2CIF/CIF/QCIF，或者 4 路 4CIF 分辨率音视频压缩，每张板卡支持 1 路模拟视频矩阵输出和 1 路模拟音频矩阵输出功能，DS-4100 系列板卡的音频实时监听功能不再需用 4 针线连接板卡和声卡音频输入口。

1.4 DS-40xx 系列

DS-40xx 系列板卡包含 HC、HC+、HCS、HF、HS、MD、MD+等型号。

DS-4004HC/HC+支持 4 路的 DCIF/2CIF/CIF/QCIF 实时编码压缩，也支持 2 路的 4CIF 实时编码压缩。若需要作为 4CIF 编码录像，应用程序可以从 DS-4004HC 的 4 个编码通道中任意选取两个通道设置为 4CIF

分辨率，然后对这两个通道进行录像，此时，DS-4004HC/HC+卡的另外两个通道的图像可以作为视频预览或者不予以显示；

DS-4008HC/HC+板卡支持 8 路的 DCIF/2CIF/CIF/QCIF 实时编码压缩，也支持 4 路的 4CIF 实时编码压缩。若需要作为 4CIF 编码录像，应用程序可以从 DS-4008HC/HC+的 8 个编码通道(编码通道为 0，1，2，3，4，5，6，7)中的前面 4 个通道(0,1,2,3)任意选取两个通道设置为 4CIF 分辨率，再从后面 4 个编码通道(4,5,6,7)中任意选取两个通道设置为 4CIF 分辨率，然后对这选中的四个通道进行录像；

DS-4016HC 板卡支持 16 路的 DCIF/2CIF/CIF/QCIF 实时编码压缩，也支持 8 路的 4CIF 实时编码压缩。若需要作为 4CIF 编码录像，应用程序可以从 DS-4016HC 的 16 个编码通道(编码通道为 0~15)中的每 4 个通道(0、1、2、3 或者 4、5、6、7 或 8、9、10、11 或 12、13、14、15)中任意选取两个通道设置为 4CIF 分辨率，然后对这选中的八个通道进行录像；

对于 DS-4004HC、DS-4008HC、DS-4016HC 板卡，通过子通道编码，可以把每一个通道全部设置为 4CIF 分辨率(SetSubEncoderPictureFormat)，这样每一个通道就都可以实现 4CIF 编码，然后通过函数 StartSubVideoCapture 实现每个通道的 4CIF 分辨率录像。在一般场景下，每路图像都可以达到 15 帧以上。

DS-4016HCS：16 路视音频压缩卡，支持 16 路 CIF/QCIF 音视频实时压缩，不支持 4CIF、2CIF、DCIF 分辨率，不支持双编码。

DS-4004HF、DS-4008HF：全通道 4CIF 编码卡，每个通道均可进行 4CIF 实时编码。

DS-4008HS、DS-4016HS：一芯八路视音频压缩板卡，每个 DSP 支持 8 路 CIF/QCIF 音视频实时压缩，不支持 4CIF、2CIF、DCIF 分辨率，不支持双编码。

DS-4002MD、DS-4004MD、DS-4002MD+、DS-4004MD+系列板卡是矩阵解码板卡，实现解码的同时实现视音频矩阵功能。

第2章 SDK 版本更新

V6.5.10.507(build20130507)

新增板卡以及功能

- 新增支持 DS-4308MD-E 解码卡：
  （1） 包含 16 个解码通道和 8 个输出通道
  （2） 实时解码：
  最大支持 11 路 WD1/800*600 标准 H.264 码流，
  或者 13 路 4CIF 标准 H.264 码流，
  或者 16 路 4CIF MPEG4 码流 ，
  或者 5 路 4CIF 海康私有编码码流，
  或者 16 路 2CIF/DCIF/CIF/QCIF 码流
  （3） 非实时解码：
  最大支持 16 路 WD1/ 800*600/4CIF 以及以下分辨率码流

V6.4.10.1101(build20121101)

新增板卡以及功能

- 新增支持 DS-4308HW-E、DS-4316HW-E WD1 编码卡：
  （1） DS-4308HW-E 支持 8 路 WD1 输入，DS-4316HW-E 支持 16 路 WD1 输入
  （2） 编码、抓图以及原始码流分辨率增加 WD1 分辨率，支持 WD1/4CIF/2CIF/CIF/QCIF
  （3） 支持 1 路 CVBS 4CIF 输出
- 移动侦测：WD1 分辨率情况下，只对宽度进行缩放到 704，高度保持与原制式相同

V6.3.10.731(build20120731)

新增板卡以及功能

- 新增支持 DS-4304HD-E 高清解码卡：
  （1） 包含 16 个解码通道和 4 个 HDMI 输出通道
  （2） 支持 2 路 500 万像素(2560*1920)分辨率解码，
  或者 4 路 1080P 分辨率解码，
  或者 8 路 720P 分辨率解码，
  或者 16 路 4CIF 及以下分辨率解码
  （3） HDMI 输出支持的分辨率为：
  DISPLAY_RESOLUTION_XGA_60HZ
  DISPLAY_RESOLUTION_SXGA_960_60HZ
  DISPLAY_RESOLUTION_720P_50HZ
  DISPLAY_RESOLUTION_720P_60HZ

DISPLAY_RESOLUTION_1080P_50HZ

DISPLAY_RESOLUTION_1080P_60HZ

（4） HDMI 视频输出可以通过调用 SetDisplayVideoMode 来设置输出分辨率和帧率

- 新增支持 DS-53xx 系列 IA 嵌入式设备(DS-5316HF)

  注意：此类产品相关信息请查阅 IA 嵌入式设备相关文档

V6.2.10.0423(build20120423)

新增板卡以及功能

- 新增支持 DS-4304HFH-E 高清编码卡：

（1） 支持 4 路 HD-SDI 视频输入和 1 路 HDMI 输出

（2） 视频输入分辨率和帧率：720P_25HZ、720P_30HZ 、720P_50HZ、720P_60HZ、1080I_50HZ 、1080I_60HZ、1080P_25HZ、1080P_30HZ

（3） HDMI 输出支持的分辨率为：

DISPLAY_RESOLUTION_XGA_60HZ

DISPLAY_RESOLUTION_SXGA_960_60HZ

DISPLAY_RESOLUTION_720P_50HZ

DISPLAY_RESOLUTION_720P_60HZ

DISPLAY_RESOLUTION_1080P_50HZ

DISPLAY_RESOLUTION_1080P_60HZ

（4） 编码、抓图以及原始码流分辨率增加 HD_F、HD_H、HD_Q 三种分辨率

注：HD_F、HD_H、HD_Q 分辨率具体与视频输入制式相关，分别对应视频输入制式的原始尺寸、原始尺寸的一半、原始尺寸的 1/4。具体对应关系如下表：

分辨率 / 视频输入制式	HD_F	HD_H	HD_Q
1080P/1080I	1920*1080	1920*540	960*540
720P	1280*720	1280*360	640*360

（5） DS-43xxHFH-E 输出相关的 API SetDisplayStandard 调用无效，用 SetDisplayVideoMode 替代

（6） 新增接口 SetDefaultHDVideoStandard，设置系统默认高清视频制式

（7） DS-43xxHFH-E 编码通道和显示通道 OSD 增加支持 8x8 倍数放大

（8） 移动侦测区域的坐标值是相对于源图像为 704*576 时的位置坐标，LOGO 位置等的坐标值是相对于源图像为 704*480 时的位置坐标，DSP 会自动根据实际图像大小来调整位置。比如视频源为 1920*1080，设置坐标为（x,y），则移动侦测区域的实际坐标为（1920*x/704,1080*y/576），LOGO 的实际坐标为（1920*x/704,1080*y/480）

V6.1.10.1129(build20111129)

新增板卡以及功能

- 新增支持 DS-43xxHCV-E、DS-43xxHFV-E 编码卡
- 新增 API：

（1） 设置码流封装格式：SetStreamPackType
（2） 设置视频输入场景模式：SetSceneMode
（3） 编码去噪功能：SetDeNoise

V6.0.10.411

新增板卡及功能

- 新增支持 DS-4002MD+、DS-4004MD+解码卡：
  （1） DS-4002MD+包含 8 个解码通道和 2 个输出通道，
  可以同时解码 2 路 4CIF 分辨率码流
  或者同时解码 4 路 DCIF/2CIF 分辨率码流
  或者同时解码 8 路 CIF/QCIF 分辨率码流
  （2） DS-4004MD+包含 16 个解码通道和 4 个输出通道，
  可以同时解码 4 路 4CIF 分辨率码流
  或者同时解码 8 路 DCIF/2CIF 分辨率码流
  或者同时解码 16 路 CIF/QCIF 分辨率码流
- 新增支持 DS-4101HD 高清解码卡：
  （1） 包含 4 个解码通道和 1 个输出通道
  （2） 支持 1 路 500 万像素(2560*1920)/1080P/1080I/UXGA/XVGA 分辨率解码，
  或者 2 路 720P 分辨率解码，
  或者 3 路 SVGA 分辨率解码，
  或者 4 路 4CIF/VGA 及以下分辨率解码。
  相关函数： SetDisplayVideoMode
  GetDisplayVideoMode
  GetDisplayFormatCapability
  （3） 支持视音频矩阵输出
- 新增支持 DS-5xxx 系列 IA 嵌入式设备(DS-5116HF、DS-5216HF)
  注意：此类产品相关信息请查阅 IA 嵌入式设备相关文档

新增功能

- 新增多格式解码功能(DS-40xxMD/MD+解码卡，DS-4104HD 高清解码卡)
  （1） 码流封装格式(文件及实时流)：PS 封装、RTP 封装
  （2） 视频编码算法：H.264、MPEG4
  （3） 音频编码算法：OggVorbis、G.711_A、G.711_U、 MPEG1 Layer2 以及 MPEG2 Layer2
  （4） DS-40xxMD/MD+解码卡、DS-4101HD 高清解码卡支持的 4CI 以下非标准分辨率包含：
  160×120、160×128
  320*192；320×240(QVGA)、640×480(VGA)
  400*300、400*304(DSP 切换成 400*300)
  640*360、640*368(DSP 切成 640*360)
  其中 640×480、320×240、160×120 以及 160×128 支持输出画面 1、4、9、13 和 16 输出，其他分辨率支持画面分割不变。

- 新增编码通道指定分辨率 JPEG 格式图像抓取功能

  DS-42xxHFV、DS-41xxHCV、DS-40xxHF/HC/HC+板卡支持 4CIF/2CIF/DCIF/CIF/QCIF 分辨率 JPEG 格式抓图

  DS-4216HC、DS-40xxHS 板卡支持 CIF/QCIF 分辨率 JEFG 格式抓图。

  新增 API： GetEncoderJpegImage

- 新增编码通道指定分辨率 YUV422 格式原始图像抓取功能

  DS-42xxHFV、DS-41xxHCV、DS-40xxHF/HC/HC+板卡支持 4CIF/2CIF/DCIF/CIF/QCIF 分辨率 YUV422 格式原始图像抓图

  DS-4216HC、DS-40xxHS 板卡支持 CIF/QCIF 分辨率 YUV422 格式原始图像抓图。

  新增 API： GetEncoderOriginalImage

- 新增编码、解码和显示通道支持 Unicode OSD 功能

  新增 API： SetEncoderOsdDisplayMode
  SetDecodeOsdDisplayMode
  SetDisplayOsdDisplayMode

- 新增 DS-40xxMD/MD+、DS-41xxHCV 卡新增显示通道 LOGO 叠加功能

  LOGO 尺寸最大为 704*576

  新增 API： SetDisplayLogo
  SetDisplayLogoDisplayMode
  StopDisplayLogo

- 新增通道能力集功能

  支持根据通道号来获取通道能力集，支持根据函数名来获取该函数对应的通道能力。

  新增 API： GetChannelCapability、
  CheckFunctionChannelCapability

- 新增视频遮挡 MASK 颜色设置功能

  对 DS-41xxHCV、DS-40xxHF/HC/HC+/HCS/HS 板卡有效

  新增 API： SetupEncoderMask

- 新增编码通道预览局部放大功能

  新增 API： ZoomOutEncoderVideoPreview
  SetEncoderZoomOutAntiTearing

  相关 API： RegisterDrawFun
  StopRegisterDrawFun

- 新增解码通道预览局部放大功能

  新增 API： ZoomOutDecoderVideoPreview
  SetDecoderZoomOutAntiTearing

  相关 API： HW_RegisterDrawFun
  HW_StopRegisterDrawFun

- 新增编码通道视频锐化参数设置功能

  新增 API： SetEncoderVideoSharpness

  GetEncoderVideoSharpness

- 新增填充显示区域扩展功能

  新增 API： FillDisplayRegionEx

- 新增编码通道和解码通道 PCI 音频预览功能

  新增 API： SetDirectAudioPreview

  HW_SetDirectAudioPreview

- **DS-4200 系列板卡新增功能**

  （1） 增加获取原始码流功能，相关 API：RegisterImageStreamCallback、SetImageStream

  DS-4216HC 板卡能够获取的原始码流的分辨率最大为 CIF。

  DS-4208/4216HFV 板卡能够获取的原始码流的分辨率最大为 4CIF。

  （2） 增加编码通道视频矩阵输出功能，但仅支持单画面输出。

  （3） 增加编码通道音频矩阵输出功能。

  （4） 增加支持 DCIF 编码。

V5.1(2009-7-27)

更新

- .全面兼容 DS-42xx 系列、DS-41xx 系列、DS-40xx 系列板卡
- 显示库更新，支持多屏预览
- 当采用 Overlay 模式进行预览时，需要调用 SetPreviewOverlayMode 启用 Overlay，SDK 不再采用大画面(大于 704*576)自动切换到 Overlay 模式的方式
- 为解决某些主板上不带 cd_in 口的问题，DS- 42xx、41xx 系列板卡可以支持不接 4 针音频线直接进行音频预览功能

DS-42xxHFV 卡

- 增加本卡模拟视频输出功能，支持视频单画面输出(调用函数 SetEncoderVideoExtOutput)
- 增加本卡模拟音频输出功能(调用函数 SetEncoderAudioOutput)

DS-4216HC 卡

- 增加了 2CIF 以及 4CIF 主通道非实时编码，4216HC 卡启动 4CIF 主编码时，无子通道编码

5.0_1524 版本(2009-3-24)

更新

- 支持 DS-4004HF 板卡
- DS-4108HCV 和 DS-4116HCV 分别支持 8 路和 16 路非实时 4CIF 子通道编码
- 优化 DS-41xx 系列板卡的编码性能

修正 bug

- 修改移动侦测区域设置出错的问题

- 修改 LOGO 功能设置某些关键色无效的问题
- 修改 PC 图像显示出错的相关问题
- 修改 DS-40xx 系列板卡 JPEG 抓图有白线的问题

5.0 版本(2008-10-6)

更新

- 支持 DS-41xx 系列板卡(DS-4108HCV、DS-4116HCV)。
- 5.0 版本 SDK 兼容 DS-41xx 系列、DS-40xxHC/HC+/HCS/HF/HS/MD/系列板卡，不再兼容 DS-4000H 系列板卡。
- 新增音频矩阵输出功能，可将任意编码通道或者解码通道的音频数据输出到任意模拟音频输出口上，本功能适用于 HCV 卡和 MD 卡。
- HCV 卡视频矩阵输出功能使用函数 SetEncoderVideoExtOutput 实现，与 MD 卡本地矩阵输出功能相同，音频矩阵输出使用新增函数 SetEncoderAudioOutput 或者 SetEncoderAudioOutputExt 实现。

修正 bug

- 解决了移动侦测区域判断出错的问题
- 解决了 HS 卡全屏预览反复启停时错位的问题。

新增 API 函数

SetEncoderAudioOutput

SetEncoderAudioExtOutput

SetDecoderAudioExtOutput

4.31 版本(2009-02-18)

更新

- 支持 DS-4004HF 板卡

修正 bug

- 修改移动侦测功能中区域设置出错的问题
- 修改 LOGO 功能中设置某些关键色无效的问题
- 修改 PC 图像显示(本地预览)出错的相关问题

备注

- 4.31 版本 SDK 中，NVIDIA 显卡在 Vista 系统下不支持 Overlay

4.3 版本(2008-06-01)

更新：

- 支持一芯八路 DS-40xxHS 卡(DS-4008HS、DS-4016HS)，每块 DSP 支持 8 路 cif/qcif 编码，不支持子码流，支持 YUV 抓图、JPG 抓图、原始视频捕获、本地矩阵输出
- 新增 DS-4016HC 卡，16 路视音频压缩板卡，功能与原 DS-4004HC、4008HC 板卡相同
- 编解码性能提升
- 完善了对视频信号检测的判断

- 编解码通道上限扩充至 256 路
- 支持纯音频流编码
- 在 HC、HC+、HF 卡上增加了色度串扰的处理
- MD 卡解码延时降低
- MD 卡启动后默认音频输出改为关闭状态,之前版本为默认输出前两路音频
- 本地矩阵输出功能增加支持帧率控制，SetEncoderVideoExtOutput
- 增加 MD 卡解码视频捕获功能，HW_SetDecoderVideoCapture
- 增加 MD 卡解码图像显示的回调函数，HW_RegisterDrawFun

修正 bug

- 解决：退出应用程序时，界面已经关闭，但 SDK 可能还没有彻底退出，此时如果再启动应用程序，可能会导致死机。
- 解决：GetSoundLevel 在 HCS 卡的前 12 路上无法正确执行
- 解决抓图问题：抓 BMP 时，可能导致图像错位；抓 JPG 时，可能会返回超时，并且无法恢复。
- 解决：如果用户采用多线程来输入码流，MD 卡可能会出现多路图像混叠情况。
- 解决：MD 卡回放时文件尾部数据可能无法解码
- 解决：MD 卡解码 N 制 QCIF 花屏。
- 解决几个显卡预览相关问题。

新增 API 函数

RegisterDecoderVideoCaptureCallback

HW_SetDecoderVideoCapture

HW_RegisterDrawFun

4.2 版本(2006-07-26)

更新

- 支持 DS-4008HF 卡
- 编解码改善对噪声图像的处理，使蠕动现象不明显
- 增加对码流的 CRC 校验功能
- DS-4000MD 卡增加文件索引导入、导出功能
- DS-4000MD 卡增加视频输出亮度调整功能
- 增强 DS-4000MD 卡在流模式下的功能，设置速度、暂停、定位等功能可以在流模式下使用
- 增加 DS-4000MD 卡模拟输出视频捕获功能
- 新增隔行解码功能
- 解码后增加后处理

修正 bug

- 解决 DS-4000MD 卡无法解码某些小文件问题
- 解决 DS-4000MD 卡无法解码某些文件尾部一段数据的问题
- 解决部分 DS-4004MD 卡音频输出顺序混乱问题
- 解决 DS-4000MD 卡多路解码时，音频输出通道间可能会混乱的问题
- 解决 4.1 版本解码时音频可能有杂音的问题

新增 API 函数

SetChannelStreamCRC

SetSubChannelStreamCRC

解码 API

HW_ImportFileRef，HW_ExportFileRef

SetDisplayVideoCapture，RegisterDisplayVideoCaptureCallback

HW_SetDecoderPostProcess

SetDisplayVideoBrightness

4.1 版本(2005-10-15)

更新

- 支持全 DS-4000HC+卡
- 编码性能优化，全面提升图像质量，特别是 4CIF 的图像质量有很大提高
- DS-4000MD 卡增加文件索引功能，支持按照时间或帧号定位功能，可以获取录像文件的起止时间
- MD 卡支持抓图

修正 bug

- DS-4000MD 卡无法解码某些子通道的录像文件
- DS-4000MD 矩阵输出时可能会出现图像错误
- DS-4000MD 卡回放小文件时，可能会误报文件结束
- DS-4000HC 原始图像流的帧率控制无效(Ver:4.0)
- 录像音频的音量偏小(Ver:3.0-4.0)

4.0 版本(2005-07-25)

更新

- 支持新的板卡：DS-4016HCS、DS-4004MD
- DS-4016HCS：16 路视音频压缩卡。支持 16 路 CIF 实时压缩，支持 CIF/QCIF 分辨率，不支持 4CIF、2CIF、DCIF 分辨率，不支持双编码。新增加了 WatchDog 和报警输入、输出功能
- DS-4004MD： 8 路解码、4 路输出矩阵解码卡。产品功能和 2 块 DS-4002MD 相同;
- 视频预览帧率可调(PAL:1-25f/s,NTSC:1-30f/s);
- 增加新的移动侦测接口 SetupMotionDetectionEx，提供了更灵活的功能，并且简化了用户的工作量；对于移动侦测的操作应用程序仅需调用 3 个接口函数:
  SetupMotionDetectionEx、StartMotionDetection 和 StopMotionDetection;
- 在应用程序以新的接口函数实现移动侦测功能时，SDK 不再返回移动侦测帧，而仅仅是通过函数 SetupMotionDetectionEx 所调用的回调函数 MotionDetectionCallback 的参数 bMotionDetected 告知应用程序视频是否处于移动状态；
- 增加新的 OSD 接口 SetOsdDisplayModeEx，最多支持 8 行 OSD 字符。同时，修改 OSD 参数时无需重新启停;
- 对 SDK 的发布文件做了简化，实现所有的功能只需 ds40xxsdk.dll 一个文件;
- SDK 内部增加了异常检测、恢复机制，增强系统稳定性，无需用户干预;
- 在 DS-4016HCS 上实现了 WatchDog 功能,接口函数为 SetWatchDog，只要打开任意一块 DS-4016HCS 的 WatchDog 功能，就可以实现对上层软件和系统中所有压缩板卡的运行状态监控;
- 在 DS-4016HCS 上增加了报警输入、输出功能，当配合报警卡使用时，一块 DS-4016HCS 支持 16 路报警输入和 4 路报警输出，同时增加 RS485 串口，并提供了简单、实用的串口操作 API;

- MD 卡矩阵功能增强;
- MD 卡完善了 9 画面分割视频输出;
- 增加新的接口函数 GetJpegImage，支持 JPEG 方式抓图，抓取的图像质量动态可调;

修正 bug

- 抓图中存在的图像质量差的问题。(增加了反隔行变换);
- OSD 时钟不准确。OSD 时钟始终以主机时钟为准，同时 SetupDateTime 函数不再有效，用户无须自行校时;
- MD 卡在频繁切换画面分割时可能产生执行失败的现象;

3.3 版本(2005-04-08)

更新

- 编码效率进一步提高;
- 优化不规则窗口预览丢帧的情况;

修正 bug

- 修正 Overlay 预览的开发在 vb、dephi 下可能存在的 bug ;

3.2 版本

HC 卡更新

- 编码质量进一步提高，在相同量化系数下，新版本 SDK 对大部分场景的压缩比比旧版本提高 10-20%，即在提供同等图象质量情况下，新版本的码流比旧版本降低 10-20%;
- 移动侦测采用全新算法，增加自适应选项(只要将 AdjustMotionDetectPrecision 函数的将运动分析灵敏度等级参数 iGrade 和 0x80000000 做“或”操作,即采用自适应分析)，在光线不足的情况下移动检测的准确率大大提高;
- 在部分显卡上实现 Overlay 预览(函数 SetPreviewOverlayMode)，提高了预览的画质和降低系统的 CPU 使用率;
- 反隔行算法优化，提高了预览图象质量;
- 取原始数据流的效率提高，运行时 CPU 的使用率下降;
- 码流控制算法优化;
- 海外板卡(DS-4000HCI)支持 PCI_X 主板;
- 驱动程序进行了更新，跟旧版本 SDK 不兼容(3.2 版本中的驱动程序和 SDK 不能与旧版本中的驱动程序和 SDK 交叉使用);
- 同一路视频编码信号支持 2 路矩阵输出;

MD 卡更新

- 优化网络解码延时;
- 完善 PCI 传输;
- 增加水平 1/3 缩小，可实现 9 画面分割视频输出，宽度需要按照 232 对齐;

修正 bug

- 修正了 HC 卡 OSD 时间错误的 bug;

3.1 版本

新增功能：

- H 卡与 HC 卡混插时，H 卡也可以在录像时动态更改分辨率；
- 解码器性能优化，与 3.0 相比提高约 50%；
- 完善 MD 卡的解码功能，提高了图像显示和音频输出的质量。
- 在 MD 卡中完善了对原 4004D 卡中已有功能的支持，绝大部分 API 和原 4004D 卡兼容；
- 完善 LOGO 配置，对 SetLogoDisplayMode 和 SetLogo 的使用，无需再考虑先后顺序；
- 完善图像处理，预览和回放的图像质量有所提高；
- HC 卡子通道可以在录像时动态更改分辨率；
- 增加了设置反隔行变换参数的接口 SetDeInterlace，用户可以设置是否执行反隔行变换，以及反隔行变换的强度，请参考相应的函数说明；

修正 bug：

- 解决了 DS-4008HC 卡启动顺序混乱的问题；
- 同时启动主通道和子通道，如果包含 4CIF 分辨率，对该路图像的反隔行变换可能会被忽略；
- 原始图像流功能在长时间运行后会停止。

3.0 版本

新增功能

- 增加 DS-4002MD(矩阵解码卡)支持，基于 DS-4002MD 可以实现视频矩阵和硬件解码功能(请参考《DS-4002MD、数字视频矩阵方案》)。
- 优化系统调度、增强编码功能。子通道录像可以设置为复合流或视频流，录像的分辨率可以任意设置，不再仅限于 QCIF。可以组成更灵活的编码方案，例如：
- 4 路 4CIF 非实时录像；主通道 4CIF＋子通道 CIF 录像等，请参考“双编码功能说明”。
- 增加 OSD 字符大小调节功能，用户可自定义 OSD 字符大小，也可以设置为根据编码分辨率自动调整 SetOsdDisplayMode。
- 增加调节视频输入信号检测灵敏度功能。避免因视频输入信号的偶然变化，使“无视频信号”提示频繁出现，影响图像 SetVideoDetectPrecision。
- 增加获取系统信息(板卡、DSP、编码通道、解码通道、矩阵输出通道……)接口，用户可以获得更全面的板卡配置。

已解决的问题

- 完善多任务处理。
- 在录像的同时修改帧结构等参数时会导致录像文件出现短时间花屏。
- 当分辨率设为 CIF 或 DCIF 时，如果 OSD 设置为半透明，在 OSD 字符(特别是中文字符)的右下部分会特别亮或特别暗(没有执行透明处理)。
- 在进行 4 路 2CIF 或 4 路 DCIF 录像时，如果视频信号频繁发生变化(PAL　NTSC)，可能会导致某一路录像停止。

DS-4002MD(矩阵解码卡)可以实现视频矩阵和硬件解码功能。

■ 解码功能

- 每块 DS-4002MD 可做 4 路解码。
- 支持的码流格式：海康威视 H、HC 系列板卡；海康威视 M、ME、ATM、HC、DVS 系列嵌入式设备。

音、视频输出：

- 音频输出：2 路，可在 4 个解码通道中任选 2 路输出。
- 视频输出：2 路，每路视频输出最多可以划分为 16 个窗口。
- 音频预览：每块 DS-4002MD 支持 1 路音频预览输出。

软件：从海康威视 3.0 版 SDK 开始提供对 DS-4002MD 的支持。

- 支持 H 卡、HC 卡和 MD 卡在 1 台 PC 内混插。
- 在一个 SDK 内同时支持 H 卡、HC 卡和 MD 卡。
- 解码部分的 API，绝大部分和原海康威视 DS-4004D 解码卡的 SDK 完全兼容(功能发生变动的 API 详见“附录”)。

目前 1 台 PC 最多支持 16 块 DS-4002MD 卡，即最多支持 64 路解码，32 路视频输出。

基本解码性能(值为每解码 1 路视频大约要占用的 DSP 资源)：

CIF：12％(512Kb)；16％(2Mb)

2CIF：30％(1Mb)

DCIF：28％(768Kb)

4CIF：50％(1.5Mb)；60％(3Mb)

※ 上述测试文件为定码率下的稳定图像。

※ 目前对解码器的进一步优化正在进行中，其性能在以后的版本中会不断的得到提升。

■ 矩阵功能

视频矩阵功能可以概括为

- 视频输入端：由 HC 卡实时采集的视频、MD 卡解码后视频(本地文件或网络实时流)。
- 视频输出端：MD 的视频输出通道。视频输出支持画面分割，每路视频输出最多可划分为 16 窗口，视频矩阵以窗口为单位进行图像切换。
- 矩阵控制：对于 1 台 PC 中的所有 HC 卡和 MD 卡，HC 卡的每个编码通道和 MD 卡的每个解码通道，都可以把本通道的视频输出到任意一块 MD 卡的任意一路显示通道中的任意一个窗口进行显示。

矩阵的基本参数

每块 DS-4002MD 支持 2 路矩阵输出，每路输出为 4CIF 分辨率。

HC 卡的每个编码通道可以同时支持 1 路显卡预览和 2 路矩阵输出。

MD 卡的每个解码通道可以同时支持 1 路显卡输出和 2 路矩阵输出。

每路视频输出都支持画中画功能，每个窗口的位置动态可调。

每路视频输出总的窗口面积之和不能超过 4CIF＋2*QCIF，即最大可以实现一个 4CIF 的全屏输出＋2 个 QCIF 的画中画输出。

新增 API

GetBoardCount

GetDspCount

GetBoardDetail

GetDspDetail

GetEncodeChannelCount

GetDecodeChannelCount

SetSubStreamType

GetSubStreamType

SetDefaultVideoStandard

SetVideoDetectPrecision

SetOsdDisplayMode(扩展)

视频输出、矩阵控制相关 API:

GetDisplayChannelCount
SetDisplayStandard
SetDisplayRegion
ClearDisplayRegion
SetDisplayRegionPosition
FillDisplayRegion
SetEncoderVideoExtOutput
SetDecoderAudioOutput
SetDecoderVideoOutput
SetDecoderVideoExtOutput

解码 API，详见原解码卡的 SDK，需要注意的事项，请参考“DS-4002MD 说明”

2.1 版本

- 修正了 2.0 版本中做 4CIF 录像时,录像文件中存在色块的 bug。
- 修正 2.0 版本中无视频信号检测失败的 bug。
- 视频预览时,Overlay 颜色的底色可以由用户自己自由设置,调用函数 SetOverlayColorKey 设置的 Overlay 颜色要与对应的预览窗口设置的颜色一样。

2.0 版本

- 改进系统调度和通讯，提高数据传输效率，减少预览丢帧，预览更流畅。
- 优化图像处理算法，编码图像与预览图像质量有所提高。
- 改进编码器，编码效率大幅提高，同时改善录像质量。在现有 4004HC 卡上可以做到两路 4CIF 实时编码，或者 4 路 2CIF(PAL:704*288 NTSC:704*240)实时编码。
- 增加新的编码分辨率：Double CIF(ENC_DCIF_FORMAT)，PAL：528*384，NTSC：528*320。
- DCIF 和 CIF 相比，在相同的码流下，图像质量和线数会有明显提高。
- 系统配置更加灵活，可以在编码的同时修改除码流类型(复合流、视频流)外的所有参数，包括分辨率、码流、帧结构等。在编码过程中，也可以检测到视频信号制式的改动，并自动切换对应的编码、预览图像的大小。
- 升级码流格式，可以支持任意改变录像分辨率而不用切换文件(需要配合新的解码库使用)。
- 增加捕获原始图象流时可以设置帧率的功能。
- 修正了旧版本的一些 BUG。

注意事项：

- 由于 DCIF 编码的计算量大，如果 4 路同时做 DCIF 编码，可能会有丢帧现象。在后续版本中，会随着软件的不断优化而得到改善。
- 由于新的版本不再限制各个通道的分辨率，因此，用户所设置的功能可能会超过了板卡所能达到的上限，此时会导致录像文件丢帧或者操作失败。例：
- 4 个通道同时做 Double CIF 编码，或者做 4 路 2CIF 编码同时还启动了 4 个子通道的双编码：此时，系统会根据图像的复杂性做出丢帧处理。
- 4 个通道同时启动 4CIF 编码：由于编码 4 路 4CIF 图像所需要的资源不够，导致启动失败，同时返回错误(ERR_NOT_SUPPORT)。

- 4 个通道在做 CIF 编码的同时，动态的把分辨率改为 4CIF：同样由于资源不足，该通道的编码会自动停止，同时返回错误(ERR_KERNEL)。

1.8 版本

- 改进小画面预览算法，小画面预览图像更清晰
- 完善板卡视频信号输入判断，在板卡初始化及长时间运行后，给予视频信号相应的检测和判断
- 完善初始化功能，减少初始化失败情况
- 改进 H 卡 OSD 显示，使 H 卡 OSD 显示位置在(0,0,703,575)内任意可调,与 HC 卡的设置相统一

1.7 版本

- 新增原始图像流捕获函数 RegisterImageStreamCallback 和 SetImageStream
- 新增视频输入位置设置函数 SetInputVideoPosition
- 新增停止画图回调功能函数 StopRegisterDrawFun
- 增加 H 卡设置屏幕遮挡的功能
- 单画面预览窗口大于或等于 704*576 时，此窗口预览自动切换为 Overlay 预览模式
- 修正了一些 BUG

1.6 版本

- 改进了在 CIF 格式下设置 OSD 和 LOGO 位置与 H 卡对齐
- 修正了 SetOsd 不能校时的 BUG
- 全面兼容即将推出的 DS-4008HC

1.5 版本

- 提高 LOGO 位图的清晰度
- 改进视频信号丢失检测机制
- 调整默认视频图像参数
- 多窗口时创建与预览窗口同等大小的 offscreen 缓冲区，可对预览窗口矩形框进行画线及图片显示等，操作更直接方便(参看 DEMO 中 DrawFun 函数)。
- 新增一个调节 OSD 时间的函数，可用于网络校时
- 修正了一些 BUG

1.0 版本

- 保持同等图像质量前提下，与 DS-400xM 系列板卡相比，压缩码流降低 30%以上，在办公室典型环境中码率仅需 20kbps~120kbps。

- 提供精确码率控制方式，无论何种情况均能输出指定码率，增加 CBR(定码率)控制方式。
- 采用新型视频采集处理芯片，极大地降低了由摄像噪声导致的图像失真、背景游动等现象，预览清晰度提高，可达 450 线。
- 采用 OggVorbis(相当于 G.722)的音频压缩算法，声音更流畅。
- 支持 PCI 2.2 接口，传输率更高，可稳定支持大路数(32 路以上，最高可达 64 路)编码与录像。
- 提供高清分辨率(4CIF(704*576))视频压缩编码功能。
- 新增一种直接取数据流方式，读写数据流的效率提高，推荐客户使用。
- 双码流功能更灵活，两路完全独立，可分别启止录像。
- 新增屏幕遮挡 MASK 函数，最多支持 32 个区域。
- 统一屏幕相对坐标(OSD、LOG、MASK、移动侦测等功能中参数)，无论采用何种编码分辨率，屏幕显示坐标均为 704*576
- 修改视频预览方式：多窗口时在显卡上创建 offscreen 表面再 BLT 到主窗口；单窗口且全屏时自动采用 OVERLAY 方式。经测试，Nvidia Tnt/Tnt2、Geforce Mx 200/400/420/440 Fx5200/5600 系列，ATI Radeon 7000/7200/7500/8500 /9000/9200 /9500/9600 系列，MatroxG450/550 系列,INTEL845G/865G 系列支持新的预览方式。注意显卡的驱动须支持硬件缩放功能， Nvidia Fx 系列显卡驱动推荐使用新版本显卡驱动(53.00 版本以上)。
- 建议使用 ATI 系列显卡以提高显示效率，屏显分辨率设置为 1024*768，颜色设为 16 位。
- SDK 接口与 DS-400xM/DS-400xH 系列板卡 SDK 原接口一致，新加其他功能(只适用于 HC 系列板卡)，成型应用软件可迅速完成移植。

第3章 错误代码及说明

3.1编码卡错误代码

错误名称	代码	说明
ERR_WAIT_TIMEOUT	0xc0000001	SDK 操作超时
ERR_INVALID_HANDLE	0xc0000002	非法句柄，在调用 SDK 函数使用了错误的句柄
ERR_INVALID_ARGUMENT	0xc0000003	参数错误，输入的参数可能超出有效范围
ERR_DDRAW_CREATE_FAILED	0xc0000004	DDRAW 返回的错误，参见 MSDN
ERR_DDRAW_CAPS_FAULT	0xc0000005	DDRAW 返回的错误，参见 MSDN
ERR_SET_COOPERATIVELEVEL_FAILED	0xc0000006	DDRAW 返回的错误，参见 MSDN
ERR_PRIMARY_SURFACE_CREATE_FAILED	0xc0000007	DDRAW 返回的错误，参见 MSDN
ERR_GET_OVERLAY_ADDRESS_FAILED	0xc0000008	DDRAW 返回的错误，参见 MSDN
ERR_OVERLAY_SURFACE_CREATE_FAILED	0xc0000009	DDRAW 返回的错误，参见 MSDN
ERR_OVERLAY_UPDATE_FAILED	0xc000000a	DDRAW 返回的错误，参见 MSDN
ERR_TMMAN_FAILURE	0xc000000b	SDK 内部错误
ERR_CHANNELMAGIC_MISMATCH	0xc000000c	通道数据毁坏
ERR_CALLBACK_REGISTERED	0xc000000d	回调函数已注册
ERR_QUEUE_OVERFLOW	0xc000000e	数据流缓存溢出
ERR__STREAM_THREAD_FAILURE	0xc000000f	无法启动流处理线程
ERR_THREAD_STOP_ERROR	0xc0000010	流处理线程停止错误
ERR_NOT_SUPPORT	0xc0000011	该功能尚不支持
ERR_OUTOF_MEMORY	0xc0000012	系统内存不足
ERR_DSP_BUSY	0xc0000013	DSP 正忙
ERR_DATA_ERROR(v2.4)	0xc0000014	严重数据错误，必须重新停启压缩
ERR_KERNEL	0xc0000016	系统核心错误
ERR_OFFSCREEN_CREATE_FAILED	0xc0000017	创建 OFFSCREEN 缓冲区错误
ERR_MULTICLOCK_FAILURE	0xc0000018	多媒体时钟错误
ERR_INVALID_DEVICE	0xc0000019	无效设备
ERR_INVALID_DRIVER	0xc000001a	无效驱动
ERR_OFFSCREEN_BLT_FAILED	0xc000001b	不支持画图回调函数
ERR_ORDER	0xc000001c	函数调用顺序错误
ERR_DDRAW_NONE	0xc000001d	系统错误，没有安装 DDRAW
ERR_DDRAW7_UNSUPPORT	0xc000001e	不支持 DDRAW7.0 版本
ERR_GLOBAL_OVE_FAILED	0xc000001f	不支持 Overlay
ERR_DDRAW_GENERAL	0xc0000020	DirectDraw 一般性错误

3.2解码卡错误代码

错误名称	代码	说明
HWERR_ALLOCATE_MEMORY	0xc1000001	内存分配错误
HWERR_INVALID_HANDLE	0xc1000002	无效句柄
HWERR_DDRAW_CREATE_FAILED	0xc1000003	创建 DirectDraw 失败
HWERR_DDRAW_CAPS_FAULT	0xc1000004	DirectDraw 表面性能检测失败
HWERR_SET_COOPERATIVELEVEL_FAILED	0xc1000005	DirectDraw 设置协作级别失败
HWERR_PRIMARY_SURFACE_CREATE_FAILED	0xc1000006	DirectDraw 创建主表面失败
HWERR_OVERLAY_CREATE_FAILED	0xc1000007	DirectDraw 创建 Overlay 表面失败
HWERR_GET_OVERLAY_ADDRESS_FAILED	0xc1000008	DirectDraw 获取 Overlay 表面地址失败
HWERR_OVERLAY_UPDATE_FAILED	0xc1000009	DirectDraw 显示 Overlay 表面失败
HWERR_SURFACE_NULL	0xc100000a	DirectDraw 表面为空
HWERR_FILEHEADER_UNKNOWN	0xc100000b	文件头未知
HWERR_CREATE_FILE_FAILED	0xc100000c	打开文件失败
HWERR_FILE_SIZE_ZERO	0xc100000d	文件长度为零
HWERR_FILE_SIZE_INVALID	0xc100000e	文件大小无效
HWERR_CREATE_OBJ_FAILED	0xc100000f	创建线程或内核对象失败
HWERR_CHANNELMAGIC_MISMATCH	0xc1000010	通道数据损坏
HWERR_PARA_OVER	0xc1000011	参数错误
HWERR_ORDER	0xc1000012	函数调用顺序错误
HWERR_COMMAND	0xc1000013	命令传递失败
HWERR_UNSUPPORTED	0xc1000014	不支持该操作
HWERR_DSPOPEN	0xc1000015	DSP 打开失败
HWERR_DSPLOAD	0xc1000016	DSP 加载错误
HWERR_ALLOCATE_DSPMEMORY	0xc1000017	DSP 内存分配错误
HWERR_DSPCHECHER	0xc1000018	DSP 校验错误
HWERR_IMGFILE_UNKNOWN	0xc1000019	未知的 IMG 文件
HWERR_INVALID_FILE	0xc100001a	无效文件
HWERR_OFFSCREEN_CREATE_FAILED	0xc100001b	创建 Offscreen 表面失败
HWERR_OFFSCREEN_BLT_FAILED	0xc100001c	不支持画图回调函数
HWERR_DDRAW_NONE	0xc100001d	系统错误，没有安装 DDRAW
HWERR_DDRAW7_UNSUPPORT	0xc100001e	不支持 DDRAW7.0 版本
HWERR_GLOBAL_OVE_FAILED	0xc100001f	不支持 Overlay
HWERR_DDRAW_GENERAL	0xc1000020	DirectDraw 一般性错误

第4章 数据类型及结构体定义

- 帧类型定义

```
typedef enum {
    PktError = 0,                       //非法帧数据
    PktIFrames = 0x0001,                //I 帧包
    PktPFrames = 0x0002,                //P 帧包
    PktBBPFrames = 0x0004,              //BBP 帧包
    PktAudioFrames = 0x0008,            //音频帧包，主通道和子通道编码共用
    PktMotionDetection = 0x00010,       //移动侦测帧
    PktSysHeader = 0x00080,             //系统头
    PktBPFrames = 0x00100,              //BP 帧包
    PktSFrames = 0x00200,               //2.0 版新增，为 I 帧捕获时传送的帧类型，目前无效
    PktSubIFrames = 0x00400,            //双编码时，子通道 I 帧
    PktSubPFrames = 0x00800,            //双编码时，子通道 P 帧
    PktSubBBPFrames = 0x01000,          //双编码时，子通道 BBP 帧
    PktSubSysHeader = 0x02000           //双编码时，子通道系统头
}FrameType_t
```

- 视频标准定义

```
typedef enum {
    StandardNone        = 0x80000000,   //无视频信号
    StandardNTSC        = 0x00000001,   //NTSC 制式
    StandardPAL         = 0x00000002,   // PAL 制式
    Standard720P_24HZ   = 0x00000003,   //720P_24HZ
    Standard720P_25HZ   = 0x00000004,   //720P_25HZ
    Standard720P_30HZ   = 0x00000005,   //720P_30HZ
    Standard720P_50HZ   = 0x00000006,   //720P_50HZ
    Standard720P_60HZ   = 0x00000007,   //720P_60HZ
    Standard1080I_50HZ  = 0x00000008,   //1080I_50HZ
    Standard1080I_60HZ  = 0x00000009,   //1080I_60HZ
    Standard1080P_24HZ  = 0x0000000a,   //1080P_24HZ
    Standard1080P_25HZ  = 0x0000000b,   //1080P_25HZ
    Standard1080P_30HZ  = 0x0000000c,   //1080P_30HZ
    Standard1080P_50HZ  = 0x0000000d,   //1080P_50HZ
    Standard1080P_60HZ  = 0x0000000e    //1080P_60HZ
}
```

- 视频分辨率定义

```
typedef enum {
    ENC_CIF_FORMAT = 0,
```

```
    ENC_QCIF_FORMAT = 1,
    ENC_2CIF_FORMAT = 2,
    ENC_4CIF_FORMAT = 3,
    ENC_QQCIF_FORMAT = 4,
    ENC_CIFQCIF_FORMAT =5,
    ENC_CIFQQCIF_FORMAT =6,
    ENC_DCIF_FORMAT =7,
    ENC_VGA_FORMAT=8,
    ENC_HD_Q_FORMAT=9,
    ENC_HD_H_FORMAT=10,
    ENC_HD_F_FORMAT=11,
    ENC_WD1_FORMAT=12
}PictureFormat_t;
```

数据结构定义

- 特殊功能能力定义

```
typedef struct tagChannelCapability{
    UCHAR bAudioPreview;        音频预览
    UCHAR bAlarmIO;             报警信号
    UCHAR bWatchDog;            看门狗
}CHANNEL_CAPABILITY, *PCHANNEL_CAPABILITY;
```

- 帧数据统计

```
typedef struct tagFramsStatistics{
    ULONG VideoFrames;          视频帧
    ULONG AudioFrames;          音频帧
    ULONG FramesLost;           丢失帧
    ULONG QueueOverflow;        缓存溢出
    ULONG   CurBps              当前码流(kb/s)
}FRAMES_STATISTICS, *PFRAMES_STATISTICS;
```

- 版本信息

```
typedef struct tagVersion{
    ULONG DspVersion, DspBuildNum;          DSP 版本及 BUILD 号
    ULONG DriverVersion, DriverBuildNum;    驱动版本及 BUILD 号
    ULONG SDKVersion, SDKBuildNum;          SDK 版本及 BUILD 号
}VERSION_INFO, *PVERSION_INFO;
```

第5章 API 调用顺序

5.1编码卡 API 调用顺序

A.

设置默认的视频制式	SetDefaultVideoStandard()
设置默认的高清视频制式	SetDefaultHDVideoStandard()

B.

初始化板卡	InitDSPs()

C.

获取编码通道总个数	GetTotalChannels()
打开通道	ChannelOpen()
注册画图回调	RegisterDrawFun()
注册获取压缩编码数据流直接读取回调	RegisterStreamDirectReadCallback()
注册读取码流消息函数	RegisterMessageNotifyHandle()
注册获取原始图像数据流的回调函数	RegisterImageStreamCallback()
设置 Overlay 关键色	SetOverlayColorKey()

D.

设置视频预览模式	SetPreviewOverlayMode()
启动视频图像预览	StartVideoPreview()

E.

//设置 OSD 方式 1	
设置 OSD 显示模式(此函数支持 2 行 OSD 显示)	SetOsdDisplayMode()
设置 OSD 显示模式(此函数最多支持 8 行 OSD 显示)	SetOsdDisplayModeEx()
设置 OSD 显示	SetOsd()
//设置 OSD 方式 2	
设置 OSD 显示（支持 OSD 前景背景色、半透明等）	SetEncoderOsdDisplayMode()
//设置 Logo	
将 24 位 bmp 文件转成 yuv 格式的数据	LoadYUVFromBmpFile()
设置 LOGO 显示模式	SetLogoDisplayMode()
设置 LOGO 图像位置及数据	SetLogo()
//设置遮挡	
设置屏幕遮挡	SetupMask()

F.

设置主通道码流封装格式	SetStreamPackType()

设置主通道的编码分辨率格式	SetEncoderPictureFormat()
设置主通道编码流类型	SetStreamType()
设置编码图像质量	SetDefaultQuant()
设置编码帧结构、帧率	SetIBPMode()
设置码流的最大比特率	SetupBitrateControl(
设置码流控制模式	SetBitrateControlMode()
设置图像亮度、对比度、饱和度	SetVideoPara()

G．移动侦测方式 1

设置移动侦测灵敏度：	AdjustMotionDetectPrecision()
设置移动侦测区域及个数	SetupMotionDetection()
启动移动侦测	StartMotionDetection()
移动侦测分析	MotionAnalyzer()

G．移动侦测方式 2

设置移动侦测	SetupMotionDetectionEx()
启动移动侦测	StartMotionDetection()

H．抓图及图像保存函数

获取原始图像	GetOriginalImage()
图像保存为 BMP 文件	SaveYUVToBmpFile()
抓取 JPEG 格式图像	GetJpegImage()

I．音频幅度获取及现场声音监听

获取现场声音音量幅度	GetSoundLevel()
设置现场声音监听	SetAudioPreview()

J．获取视频、SDK 及板卡相关信息

获取视频信号输入情况	GetVideoSignal()
获取 SDK 版本号	GetSDKVersion()
获取视频参数	GetVideoPara()
获取板卡的型号和序列号	GetBoardInfo()
获取帧统计信息	GetFramesStatistics()
获取板卡的详细信息	GetBoardDetail()
获取 DSP 的详细信息	GetDspDetail()

K．启动录像(编码压缩数据)

启动主通道数据截取	StartVideoCapture()

L．启动原始图像数据流的截取

启动获取原始图像数据流	SetImageStream()

M．子通道的参数设置以及录像

设置子通道码流封装格式	SetStreamPackType()

设置子通道编码流类型	SetSubStreamType()
设置子通道的编码分辨率格式	SetSubEncoderPictureFormat()
切换至子通道	SetupSubChannel(, 1)
//其它参数设置方式与主通道相同，可以设置与主通道不同的编码量化系数，帧率等等	
切换回主通道	SetupSubChannel(, 0)
启动子通道数据截取	StartSubVideoCapture()

N．视音频输出设置

//视频输出设置	
获取输出模式所支持的显示分辨率	GetDisplayFormatCapability
设置视频输出模式	SetDisplayVideoMode
获取视频输出模式	GetDisplayVideoMode
设置显示区域的形式及参数	SetDisplayRegion
开启编码通道视频内部输出	SetEncoderVideoOutput(, , 1, , ,)
停止编码通道视频内部输出	SetEncoderVideoOutput(, , 0, , ,)
开启编码通道视频外部输出	SetEnoderVideoExtOutput(, , 1, , ,)
停止编码通道视频外部输出	SetEncoderVideoExtOutput(, , 0, , ,)
清空显示区域	ClearDisplayRegion
//设置音频输出	
开启编码通道音频内部输出	SetEncoderAudioOutput(, 1,)
停止编码通道音频内部输出	SetEncoderAudioOutput(, 0,)
开启编码通道音频外部输出	SetEncoderAudioExtOutput(, , 1, ,)
停止编码通道音频外部输出	SeEncoderAudioExtOutput(, , 0, ,)

O．退出

停止画图回调函数	StopRegisterDrawFun()
停止获取原始图像数据流	SetImageStream()
停止移动侦测	StopMotionDetection()
停止主通道数据截取	StopVideoCapture()
停止子通道数据截取	StopSubVideoCapture()
停止视频图像预览	StopVideoPreview()
关闭通道	ChannelClose()
卸载 DSP	DeInitDSPs()

目前，SDK 函数之中除了 SetStreamType、SetSubStreamType 和 SetStreamPackType 不能在板卡编码录像过程中动态设置以外，其它视频参数，譬如 OSD、Logo、分辨率、帧率、码流、图像量化系数等参数都可以在编码录像的过程之中动态调整。

5.2解码卡 API 调用顺序

A.

初始化解码卡	HW_InitDecDevice()
初始化 DirectDraw	HW_InitDirectDraw()

B.

打开解码通道	HW_ChannelOpen()

C．文件解码

设置文件索引	HW_SetFileRef()
打开文件	HW_OpenFile()
设置视频预览	HW_SetDisplayPara()
启动视频解码	HW_Play()
启动音频解码	HW_PlaySound()
设置音频预览(开启)	HW_SetAudioPreview()
设置音频预览(关闭)	HW_SetAudioPreview()
停止音频解码	HW_StopSound()
停止视频解码	HW_Stop()
关闭文件	HW_CloseFile()

C．实时流解码

打开实时流	HW_OpenStream()
设置视频预览	HW_SetDisplayPara()
启动视频解码	HW_Play()
调节流模式	HW_SetStreamOpenMode()
启动音频解码	HW_PlaySound()
设置音频预览(开启)	HW_SetAudioPreview()
输入数据流	HW_InputData()
设置音频预览(关闭)	HW_SetAudioPreview()
停止音频解码	HW_StopSound()
停止视频解码	HW_Stop()
关闭实时流	HW_CloseStream()

D．视音频输出设置

//视频输出设置	
获取输出模式所支持的显示分辨率	GetDisplayFormatCapability
设置视频输出模式	SetDisplayVideoMode
获取视频输出模式	GetDisplayVideoMode
设置显示区域的形式及参数	SetDisplayRegion
开启解码通道视频内部输出	SetDecoderVideoOutput(, , 1, , ,)
停止解码通道视频内部输出	SetDecoderVideoOutput(, , 0, , ,)

开启解码通道视频外部输出	SetDecoderVideoExtOutput(, , 1, , ,)
停止解码通道视频外部输出	SetDecoderVideoExtOutput(, , 0, , ,)
清空显示区域	ClearDisplayRegion
//音频输出设置	
开启解码通道音频内部输出	SetDecoderAudioOutput(, 1,)
停止解码通道音频内部输出	SetDecoderAudioOutput(, 0,)
开启解码通道音频外部输出	SetDecoderAudioExtOutput(, , 1, ,)
停止解码通道音频外部输出	SetDecoderAudioExtOutput(, , 0, ,)

E．退出

关闭解码通道	HW_ChannelClose()
释放 DirectDraw	HW_ReleaseDirectDraw
释放解码卡	HW_ReleaseDecDevice

1、 文件解码。

- 如果需要调用 HW_LocateByAbsoluteTime(按绝对时间定位文件播放位置)和 HW_LocateByFrameNumber(按帧号定位文件播放位置)来定位文件时，必须在打开文件之前调用 HW_SetFileRef 来设置文件索引。通过调用 HW_SetSpeed 可以调节播放速度。

2、 实时流解码。

- 需要调节实时性和流畅性时，可以调用 HW_SetStreamOpenMode，将模式设置为 1~5。除了模式为 1 时与接口调用顺序有关外，其他模式都没有关系。
- 如果设置模式为 0，将实时流按照文件模式来解码：
  首先调用 HW_OpenStream 打开流，调用 HW_Play 启动解码，再调用 HW_SetStreamOpenMode 设置模式为 1 时，启动“快速模式”，“快速模式”延迟最小，播放速度最快；如果 HW_SetStreamOpenMode 在 HW_Play 之前调用，设置模式为 1 时播放速度仅次于“快速模式”。 然后调用 HW_InputData 输入流数据进行解码
- 总的来说，“快速模式”播放速度最快，模式 1~5 取值越大，流畅性越好但延迟越大。

第6章 函数说明

6.1板卡初始化及卸载

6.1.1 初始化 DSP InitDSPs

函　数：　int __stdcall InitDSPs()
参　数：　无
返回值：　系统内可用的编码通道个数。
说　明：　初始化系统中每一块板卡，应在应用软件程序启动时完成。如果返回值为 0 则表明初始化失败，可能没有找到相应的 DSP 软件模块。

6.1.2 卸载 DSP DeInitDSPs

函　数：　int __stdcall DeInitDSPs()
参　数：　无
返回值：　0
说　明：　关闭每一块板卡上的功能，应在应用软件程序退出时调用。

6.2板卡信息获取

6.2.1 获取系统中板卡的张数 GetBoardCount

函　数：　unsigned int __stdcall GetBoardCount()
参　数：　无
返回值：　系统中板卡的总张数。
说　明：　获取系统中所有板卡的张数，包含编码卡和解码卡。

6.2.2 获取系统中 DSP 的个数 GetDspCount

函　数：　unsigned int __stdcall GetDspCount()
参　数：　无
返回值：　系统中 DSP 的总个数
说　明：　获取系统中所有板卡的 DSP 的个数。

6.2.3 获取系统中编码通道的个数 GetEncodeChannelCount

函　数：　unsigned int __stdcall GetEncodeChannelCount()
参　数：　无
返回值：　系统中编码通道的个数
说　明：　获取系统中所有编码通道总个数。

6.2.4 获取系统中解码通道的个数 GetDecodeChannelCount

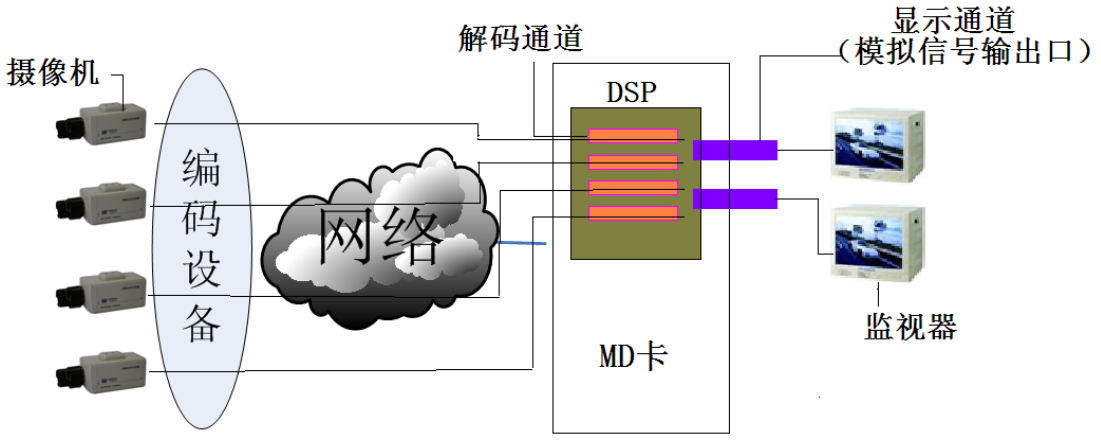
函　数：　unsigned int __stdcall GetDecodeChannelCount()
参　数：　无
返回值：　系统中解码通道的个数
说　明：　获取系统中所有解码通道个数

6.2.5 获取系统中显示通道的个数 GetDisplayChannelCount

函　数：　unsigned int __stdcall GetDisplayChannelCount()
参　数：　无
返回值：　系统中显示通道的个数
说　明：　获取系统中所有显示通道即输出通道的个数。

释　义：　解码通道、显示通道

MD、MD+、HD 卡集成解码、矩阵、画面分割 3 项功能。下面以 DS-40xxMD 卡为例，每块 DSP 包含 4 个解码通道和 2 个显示通道，解码通道解码的图像可以输出到任意一个 DSP 的任意一个显示通道上，实现矩阵功能，DS-40xxMD 每个显示通道最多支持 16 画面分割，即在同一个显示通道上最多可以同时显示前端 16 路视频信号。板卡的具体功能参数请参考板卡产品说明文档。



6.2.6 获取板卡详细信息 GetBoardDetail

函　数：　int __stdcall GetBoardDetail(UINT boardNum,DS_BOARD_DETAIL *pBoardDetail)

参　数：

UINT boardNum；板卡索引

DS_BOARD_DETAIL *pBoardDetail；板卡信息

返回值：　成功返回 0；失败返回错误号

说　明：　获取某张板卡的详细信息

板卡信息结构体

typedef struct{

BOARD_TYPE_DS type；板卡型号

BYTE sn[16]；序列号

UINT dspCount；此张板卡所包含的 DSP 个数

UINT firstDspIndex；此张板卡上第一个 DSP 在所有 DSP 中的索引

UINT encodeChannelCount；此张板卡所包含的编码通道个数

UINT firstEncodeChannelIndex；此张板卡上第一个编码通道在所有编码通道中的索引

UINT decodeChannelCount；此张板卡所包含的解码通道个数

UINT firstDecodeChannelIndex；此张板卡上第一个解码通道在所有解码通道中的索引

UINT displayChannelCount；此张板卡所包含的显示通道个数

UINT firstDisplayChannelIndex；此张板卡上第一个显示通道在所有显示通道中的索引

UINT reserved1；

UINT reserved2；

UINT initInfo；初始化信息，格式：初始化是否失败标志位:bit31,为 1 表示失败，为 0 表示成功；插槽号为 0；　总线号:bit 8-15；设备号:bit0-7，板卡没有设备号，因此设备号为 0

UINT version；硬件版本，格式：major.minor.build，major:bit 16-19，minor:bit8-15，build:0-7

}DS_BOARD_DETAIL

6.2.7 获取 DSP 详细信息 GetDspDetail

函　数：　int __stdcall GetDspDetail(UINT dspNum,DSP_DETAIL *pDspDetail)

参　数：

UINT dspNum；DSP 索引

DSP_DETAIL *pDspDetail；DSP 信息

返回值：　成功返回 0；失败返回错误号

说　明：　获取某个 DSP 的详细信息

DSP 信息结构体

typedef struct{

UINT encodeChannelCount；此 DSP 所包含的编码通道个数

UINT firstEncodeChannelIndex；此 DSP 上第一个编码通道在所有编码通道中的索引

UINT decodeChannelCount；此 DSP 所包含的解码通道个数

UINT firstDecodeChannelIndex；此 DSP 上第一个解码通道在所有解码通道中的索引

UINT displayChannelCount；此 DSP 包含的显示通道个数

UINT firstDisplayChannelIndex；此 DSP 上第一个显示通道在所有显示通道中的索引

UINT reserved1;

UINT reserved2;

UINT reserved3;

UINT initInfo; 初始化信息，格式：初始化是否失败标志位:bit31,为 1 表示失败，为 0 表示成功；插槽号为 0； 总线号:bit 8-15；设备号:bit0-7。板卡没有设备号，因此设备号为 0

}DSP_DETAIL

6.2.8 获取板卡型号及序列号信息 GetBoardInfo

函 数： int __stdcall GetBoardInfo(HANDLE hChannelHandle, ULONG *BoardType, UCHAR *SerialNo)

参 数：

HANDLE hChannelHandle；通道句柄

ULONG *BoardType；板卡型号

UCHAR *SerialNo；板卡 ID 号，内容为板卡序列号的 ASCII 的数值，次序为 SerialNo[0] 对应最高位，SerialNo[11]对应最低位。比如卡号为“４０００００００２３４５”的值对应为 4,0,0,0,0,1,0,0,2,3,4,5 的整形数组。

返回值： 成功为 0；失败返回错误号

说 明： 获取板卡的型号及序列号信息

板卡型号结构体

```
typedef enum {
    DS400XM          =0,  //M 卡，已停产
    DS400XH          =1,  //H 卡，已停产
    DS4004HC         =2,  //DS-4004HC
    DS4008HC         =3,  //DS-4008HC
    DS4016HC         =4,  //DS-4016HC
    DS4001HF         =5,  //保留，无对应产品
    DS4004HF         =6,  //DS-4004HF
    DS4002MD         =7,  //DS-4002MD
    DS4004MD         =8,  //DS-4004MD
    DS4016HCS        =9,  //DS-4016HCS
    DS4002HT         =10, //保留，无对应产品
    DS4004HT         =11, //保留，无对应产品
    DS4008HT         =12, //保留，无对应产品
    DS4004HC_PLUS    =13, //DS-4004HC+
    DS4008HC_PLUS    =14, //DS-4008HC+
    DS4016HC_PLUS    =15, //保留，无对应产品
    DS4008HF         =16, //DS-4008HF
    DS4008MD         =17, //保留，无对应产品
    DS4008HS         =18, //DS-4008HS
    DS4016HS         =19, //DS-4016HS
    DS4108HCV        =20, //DS-4108HCV
    DS4116HCV        =21, //DS-4116HCV
```

```
        DS5016HC            =22, //5016HC，已停产
        DS4208HFV           =23, //DS-4208HFV
        DS4216HC            =24, //DS-4216HC
        DS4216HFV           =25, //DS-4216HFV
        DS5008HF            =26, //保留，无对应产品
        DS5116HF            =27, //DS-5116HF
        DS5216HC            =28, //保留，无对应产品
        DS5208HF            =29, //保留，无对应产品
        DS5216HF            =30, //DS-5216HF
        DS4101HD            =31, //DS-4101HD
        DS4102HD            =32, //保留，无对应产品
        DS4104HD            =33, //保留，无对应产品
        DS4002MD_PLUS       =34, //DS-4002MD+
        DS4004MD_PLUS       =35, //DS-4004MD+
        DS4204HFV           =36, // DS-4204HFV
        DS4308HCV           =37, //DS-4308HCV-E
        DS4308HFV           =38, // DS-4308HFV-E
        DS4316HCV           =39, // DS-4316HCV-E
        DS4316HFV           =40, // DS-4316HFV-E
        DS4304HD            =41,  //DS-4304HD-E
        DS4304HFH           =42,  //DS-4304HFH-E
        DS4304HFV           =43,  //DS-4304HFV-E
        DS4302HFH           =44,  //DS-4302HFH-E
        DS4308HW            =46,  // DS-4308HW-E
        DS4316HW            =47,  // DS-4316HW-E
        DS4308MD            =48,  // DS-4308MD-E
        INVALID_BOARD_TYPE  =0xffffffff,
    }BOARD_TYPE_DS;
```

6.2.9 获取板卡 SDK 信息 GetSDKVersion

函　数：　int __stdcall GetSDKVersion(PVERSION_INFO VersionInfo)

参　数：　PVERSION_INFO VersionInfo；版本信息

返回值：　成功返回 0；失败返回错误号。

说　明：　获取当前所使用的 DSP、Driver、SDK 版本号

版本信息结构体

```
typedef struct tagVersion{
    ULONG DspVersion, DspBuildNum;
    DSP 版本号，DSP 的 BUILD 号，用于软件升级时标明该版本的最后修改时间
    ULONG DriverVersion, DriverBuildNum;
    Driver 版本号，Driver 的 BUILD 号，用于软件升级时标明该版本的最后修改时间
    ULONG SDKVersion, SDKBuildNum;
    SDK 版本号，SDK 的 BUILD 号，用于软件升级时标明该版本的最后修改时间
```

}VERSION_INFO, *PVERSION_INFO

6.3编码卡 API

通道打开及关闭

6.3.1 打开通道 ChannelOpen

函　数：　HANDLE __stdcall ChannelOpen(int ChannelNum)
参　数：　int ChannelNum；通道号(从 0 开始)
返回值：　成功返回有效句柄(值可能为 0)；失败返回 0xFFFFFFFF 或者错误号。
说　明：　打开通道，获取编码通道的操作句柄，与通道相关的操作需使用相对应的句柄。

6.3.2 关闭通道 ChannelClose

函　数：　int __stdcall ChannelClose(HANDLE hChannelHandle)
参　数：　HANDLE　hChannelHandle；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　关闭通道，释放相关资源

视频预览

Overlay 预览模式

释　义：　Overlay 预览模式

Overlay 通常被称为重叠页面或者是覆盖层，是一种需要特定的硬件支持的页面，通常被用于显示实时视频于主页面之上，而不需要 Blit 操作到主页面或用任何方法改变主页面的内容。使用该方式进行预览可以提高预览的画质和降低 CPU 利用率。

6.3.3 设置视频预览模式 SetPreviewOverlayMode

函　数：　int __stdcall SetPreviewOverlayMode(BOOL bTrue)
参　数：　BOOL bTrue；是否设置 Overlay 预览方式，也适用于 MD 卡
返回值：　0 表示显卡支持板卡的 Overlay 预览方式；其他值表示显卡不支持
说　明：　SDK 自 3.2 版本起在部分显卡中实现了 HC 卡以 overlay 方式预览的功能(此功能不支持与 H 卡混插的状态下)，可以提高预览的画质和降低 CPU 利用率。需要调用该函数来启动 Overlay 模式，如不设置则默认采用 Offscreen 模式进行预览显示。

6.3.4 设置视频预览模式扩展 SetPreviewOverlayModeEx

函　数：　int __stdcall SetPreviewOverlayModeEx(BOOL bTrue)
参　数：　BOOL bTrue；TRUE，如果显卡硬件支持则开启全局 Overlay；FALSE，关闭全局 Overlay
返回值：　0 表示显卡支持板卡的 Overlay 预览方式；其他值表示显卡不支持
说　明：　开启或者关闭全局 Overlay 扩展模式，增加了对 NV12 的检测

释　义：Overlay 关键色

Overlay 关键色相当于一层透视膜，显示的画面只能穿过这种颜色，而其他的颜色将挡住显示的画面。用户应该在显示窗口中涂上这种颜色，那样才能看到显示画面。一般应该使用一种不常用的颜色作为 Overlay 关键色。这是一个双字节值 0x00rrggbb,最高字节为 0，后三个字节分别表示 r,g,b 的值

6.3.5 设置 Overlay 关键色 SetOverlayColorKey

函　数：　int __stdcall SetOverlayColorKey(COLORREF DestColorKey)
参　数：　COLORREF DestColorKey；overlay 关键色参数(RGB(*，*，*))
返回值：　成功返回 0；失败返回错误号
说　明：　调用 SetPreviewOverlayMode 可以开启 Overlay 预览模式，关键色默认设置为 RGB (10，10，10)，用户也可以通过调用 SetOverlayColorKey 修改关键色。应用程序上只需将显示界面的底色也设置为关键色颜色即可看到显示画面
注　意：　需要在 StartVideoPreview 前调用该函数。

6.3.6 恢复当前丢失的表面 RestoreOverlay

函　数：　int __stdcall RestoreOverlay()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　恢复当前丢失的表面，例如：当系统按下 CTRL+ALT+DEL 时系统的 OVERLAY 表面会被强制关闭，调用该函数可以恢复 OVERLAY 表面

开启及停止视频预览

6.3.7 开启视频预览 StartVideoPreview

函　数：　int __stdcall StartVideoPreview(HANDLE hChannelHandle,HWND WndHandle,
RECT *rect,BOOLEAN bOverlay, int VideoFormat, int FrameRate)
参　数：
HANDLE hChannelHandle；通道句柄
HWND WndHandle；显示窗口对应父窗口的句柄
RECT *rect；显示窗口相对父窗口客户区的矩形区域坐标

BOOLEAN bOverlay；是否启用 Overlay 预览模式(目前无效)

int VideoFormat；视频预览格式(目前无效)

int FrameRate；视频预览帧率，1~30

返回值： 成功返回 0；失败返回错误号

说 明： 启动视频预览，调用 SetPreviewOverlayMode 后，可进行 Overlay 模式预览，否则将默认采用 Offscreen 模式预览。

注 意： 预览帧率 FrameRate 取值范围应该小于视频源所对应的帧率并且小于 30，例如，PAL 制的取值范围为 1~25，NTSC 制的为 1~30，高清卡编码卡 720P_60HZ 的视频输入则帧率取值范围为 1~30。

示 例： 客户区转换过程：

父窗口句柄：hParentWnd

显示窗口句柄：hWnd

通过如下操作可以获得显示窗口相对于父窗口客户区的矩形区域坐标：

```
RECT rect;
GetWindowRect(hWnd, &rect);
ScreenToClient(hParentWnd, (LPPOINT)&rect);
ScreenToClient(hParentWnd, (LPPOINT)&rect+1);
```

6.3.8 停止视频预览 StopVideoPreview

函 数： int __stdcall StopVideoPreview(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说 明： 停止视频预览

6.3.9 编码视频预览局部放大 ZoomOutEncoderVideoPreview

函 数：int __stdcall ZoomOutEncoderVideoPreview(HANDLE hChannelHandle, UINT nRectIdx, BOOL bEnable, HWND hWnd, ZOOM_RECT_NORMALIZE* pZoomRect, RECT* pRect);

参 数：

hChannelHandle ；通道句柄

nRectIdx；放大区域的序号,0 表示放大后在当前编码通道视频预览的窗口显示

bEnable；开始或者关闭局部放大功能

hWnd；显示放大区域的窗口句柄，nRectIdx 为 0 时无效

pZoomRect ；局部放大区域相对于编码通道视频预览区域的坐标，取值归一化，见 ZOOM_RECT_NORMALIZE 结构

pRect；显示局部放大视频预览的目标区域，一般为 hWnd 的客户区坐标。nRectIdx 为 0 时无效

局部放大区域归一化结构体，区域归一化,浮点数值为原始视频宽高的百分比大小,精度为 0.001

```
typedef struct _ZOOM_RECT_NORMALIZE_
{
    float left；放大区域左上角点的 X 轴坐标
    float top；放大区域左上角点的 Y 轴坐标
    float width；放大区域的宽度
```

float height；放大区域的高度

}ZOOM_RECT_NORMALIZE;

返回值： 成功返回 0；失败返回错误号

说 明： 6.0 版本新增函数

注 意： 1 路编码通道最多支持 16 个局部放大区域。nRectIdx 的取值范围是[0，16]，取 0 时表示放大后在当前编码通道视频预览的窗口显示，此时 hWnd 参数和 pRect 参数无效；取 1 到 16 时分别表示设置对应的局部放大区域的参数。

6.3.10 注册编码视频预览画图回调函数 RegisterDrawFun

函 数：int __stdcall RegisterDrawFun(DWORD nport, DRAWFUN(DrawFun),LONG nUser);

参 数：

nport；通道号。高 16 位为局部放大区域的序号，从 1 开始计，为 0 表示原始的编码视频预览图像；低 16 位为编码通道号。例如，nport 为 0x00010000，表示第 0 路编码通道第 1 个区域进行放大； nport 为 0x00000000，表示第 0 路编码通道的视频预览图像。

DrawFun；用户回调函数

nUser；保留参数

返回值： 成功返回 0；失败返回错误号

说 明： 获取当前 Offscreen 表面的 device context，HC 系列板卡采用创建 offscreen 的方式，所以在窗口客户区中的 DC 中无法画图或者写字，必须使用 DirectDraw 里的 offscreen 表面的 DC。此函数 6.0 版本起新增注册编码预览局部放大功能画图回调功能

画图回调函数

#define DRAWFUN(x) void (CALLBACK* x)(long nPort,HDC hDc,LONG nUser)

LONG nPort；通道号

HDC hDc；Offscreen 表面设备上下文，相当于显示窗口中的 DC

LONG nUser；用户数据

注 意：

如果采用 Overlay 预览模式，则直接在 Overlay 表面画图即可，无需调用此函数。

调用该接口注册编码通道的某个局部放大区域的画图回调之前，需要先调用 ZoomOutEncoderVideoPreview 接口开启该局部放大区域的局部放大功能。

6.3.11 停止注册编码视频预览画图回调函数 StopRegisterDrawFun

函 数：int __stdcall StopRegisterDrawFun(DWORD nport);

参 数：

nport；通道号。同 RegisterDrawFun 的 nport。

返回值： 成功返回 0；失败返回错误号

说 明： 停止画图回调。在某些显卡上进行画图回调，会使得 CPU 的利用率变高，所以可以在适当的时候(画图回调结束)停止调用。

6.3.12 设置编码通道局部放大的预览垂直同步

SetEncoderZoomOutAntiTearing

函　数：int __stdcall　SetEncoderZoomOutAntiTearing(HANDLE hChannelHandle, UINT nRectIdx, BOOL bAntiTearing, DWORD reserved);

参　数：

hChannelHandle:；通道句柄

nRectIdx；放大区域的序号，必须大于等于 1

bAntiTearing；是否开启垂直同步

reserved；保留参数

返回值：　成功返回 0；失败返回错误号

说　明：　6.0 版本新增函数

注　意：调用该接口开启编码通道的某个局部放大区域的预览垂直同步之前，需要先调用 ZoomOutEncoderVideoPreview 接口开启该局部放大区域的局部放大功能。且调用该接口开启局部放大的预览垂直同步后，若调用 ZoomOutEncoderVideoPreview 接口重新开启该局部放大区域的局部放大功能后，该局部放大区域的预览垂直同步功能失效，如果需要打开该局部放大区域的预览垂直同步，要再次调用该接口开启。

视频参数的设置及获取

6.3.13 设置视频参数 SetVideoPara

函　数：　int __stdcall SetVideoPara(HANDLE hChannelHandle, int Brightness, int Contrast,int Saturation, int Hue)

参　数：

HANDLE hChannelHandle；通道句柄

int Brightness；亮度值(0-255)

int Contrast；对比度(0-127)

int Saturation；饱和度(0-127)

int Hue；色调(0-255)

返回值：　成功返回 0；失败返回错误号

说　明：　设置视频参数，DS-43xxHFH-E 不支持此接口。

6.3.14 获取视频参数 GetVideoPara

函　数：　int __stdcall GetVideoPara(HANDLE hChannelHandle, VideoStandard_t *VideoStandard, int*Brightness, int *Contrast, int *Saturation, int *Hue)

参　数：

HANDLE hChannelHandle；通道句柄

VideoStandard_t *VideoStandard；视频制式

int *Brightness；亮度指针值(0-255)

int *Contrast；对比度指针值(0-127)

int *Saturation；饱和度指针值(0-127)

int *Hue；色调指针值(0-255)

返回值： 成功返回 0；失败返回错误号

说 明： 获取视频参数，对于 DS-43xxHFH-E，仅视频制式项有效，其他视频参数项无效

VideoStandard_t 视频制式

StandardNone； 无视频信号

StandardNTSC； NTSC 制式

StandardPAL； PAL 制式

Standard720P_25HZ； 720P_25HZ

Standard720P_30HZ； 720P_30HZ

Standard720P_50HZ； 720P_50HZ

Standard720P_60HZ； 720P_60HZ

Standard1080I_50HZ； 1080I_50HZ

Standard1080I_60HZ； 1080I_60HZ

Standard1080P_25HZ； 1080P_25HZ

Standard1080P_30HZ； 1080P_30HZ

6.3.15 设置编码通道的视频锐化参数

SetEncoderVideoSharpness

函 数：int __stdcall SetEncoderVideoSharpness(HANDLE hChannelHandle, int sharpness);

参 数：

hChannelHandle ；通道句柄

sharpness；锐化参数，取值范围[0,3]，值越大锐度越大。

返回值： 成功返回 0；失败返回错误号

说 明： 6.0 版本新增函数。DS-4016HCS、DS-43xxHFH-E 不支持此项功能。

6.3.16 获取编码通道的视频锐化参数

GetEncoderVideoSharpness

函 数：int __stdcall GetEncoderVideoSharpness(HANDLE hChannelHandle, int* pSharpness);

参 数：

hChannel；通道句柄

pSharpness ；锐化参数

返回值： 成功返回 0；失败返回错误号

说 明： 6.0 版本新增函数。DS-4016HCS、DS-43xxHFH-E 不支持此项功能。

视频信号设置(制式、状况、输入位置等)

6.3.17 设置系统默认标清视频制式 SetDefaultVideoStandard

函　数：　int __stdcall SetDefaultVideoStandard(VideoStandard_t VideoStandard)
参　数：　VideoStandard_t VideoStandard；视频制式，默认为 PAL
返回值：　成功返回 0；失败返回错误号
说　明：　设置系统默认的标清视频制式，系统中所有的标清视频输入通道如果无视频输入或者在系统启动的时候，通道会按照所设置的系统默认标清视频制式进行处理。
注　意：　该函数只能在系统初始化(InitDSPs)之前运行，否则无效。

6.3.18 设置系统默认高清视频制式 SetDefaultHDVideoStandard

函　数：　int __stdcall SetDefaultHDVideoStandard(VideoStandard_t VideoStandard)
参　数：　VideoStandard_t VideoStandard；视频制式，默认为 Standard1080P_25HZ
返回值：　成功返回 0；失败返回错误号
说　明：　设置系统默认的高清视频制式，系统中所有的高清视频输入通道如果无视频输入或者在系统启动的时候，通道会按照所设置的系统默认高清视频制式进行处理。
注　意：　该函数只能在系统初始化(InitDSPs)之前运行，否则无效。

6.3.19 设置视频信号灵敏度 SetVideoDetectPrecision

函　数：　int __stdcall SetVideoDetectPrecision(HANDLE hChannel,unsigned int value)
参　数：
　　HANDLE hChannel；通道句柄
　　unsigned int value；灵敏度。取值范围：0-100，默认为 20
返回值：　成功返回 0；失败返回错误号
说　明：　设置视频信号检测的灵敏度。如果视频信号的强度比较弱，或者信号通断的切换比较频繁，会出现“无视频信号”的提示字样，为了避免提示字样影响图像，可以更改视频信号检测的灵敏度。灵敏度取值越大，检测精度越低，出现“无视频信号”提示字样的频率越低。当将 value 值设置为 0xffffffff 时，将不会再出现“无视频信号”的提示字样。

6.3.20 获取视频信号输入情况 GetVideoSignal

函　数：　int __stdcall GetVideoSignal(HANDLE　hChannelHandle)
参　数：　HANDLE　hChannelHandle；通道句柄
返回值：　信号正常返回 0；返回其他值说明信号异常或有错误
说　明：　获取视频信号的输入情况，用于视频丢失报警

6.3.21 调整视频信号输入位置 SetInputVideoPosition

函　数：　int __stdcall SetInputVideoPosition(HANDLE hChannel,UINT x,UINT y)

参　数：

HANDLE hChannelHandle；通道句柄

UINT x；X 轴坐标，默认值为 8

UINT y；Y 轴坐标，默认值为 2

返回值：　成功返回 0；失败返回错误号

说　明：　设置视频信号的输入位置。(x，y)为系统处理图像的左上角在摄像机输入的原始图像中的坐标，某些摄像机输入的图像在预览时可能在左边会有黑边，可以通过此函数进行调节，x 必须设置为 2 的整数倍。(x，y)的取值和摄像机的型号有关，如果指定的值和摄像机的输入参数不匹配，可能会导致图像静止、水平垂直方向滚动或者黑屏，请谨慎使用。

注　意： DS-43xxHFH-E、DS-52xx 系列和 DS-42xx 系列板卡不支持输入位置的调整，DS-43xxHW-E 系列板卡不支持 Y 方向调整。

6.3.22 设置反隔行变换及强度 SetDeInterlace

函　数：　int __stdcall SetDeInterlace(HANDLE hChannelHandle,UINT mode,UINT level)

参　数：

HANDLE hChannelHandle；通道句柄

UINT mode；mode 为 0 表示关闭反隔行变换，此时 level 参数无效；mode 为 1 时对 DS-40xx 卡有效，通过设置 level 可以调节反隔行强度；mode 为 2 时适用于所有板卡，此时 level 无效，使用默认算法(系统默认值)。

UINT level；当 mode＝1 且为 DS-40xx 板卡时有效，其它时无效。0－10，反隔行变换的强度逐渐加强，0 最弱，对图像的损失最小，10 最强，对图像的损失最大。

返回值：　成功返回 0；失败返回错误号

说　明：　设置是否采用反隔行算法以及采用反隔行时的强度

释　义： **反隔行变换**

如果该通道的图像需要进行 4CIF 的预览或编码，此时的图像中会同时包含奇、偶两场的数据，由于奇场图像和偶场图像不同步，导致图像中运动的部分发生错位、边缘模糊，此时需要对图像进行反隔行变换来去掉这种现象。如果用户能够确定使用的是逐行扫描格式的摄像机，或者主要应用在静止场景，此时可以关掉反隔行变换功能，或者降低强度，这样可以提高系统运行效率，并降低反隔行变换对图像质量带来的损失。

6.3.23 设置视频输入场景模式 SetSceneMode

函　数：　int __stdcall SetSceneMode(HANDLE hChannelHandle,UINT mode)

参　数：

HANDLE　hChannelHandle；通道句柄

UINT　mode；　视频输入场景模式，取值范围为 0-3。

返回值：　成功返回 0；失败返回错误号

说　明：　6.1 新增接口，适用于 DS-43xxHCV-E、DS-43xxHFV-E 和 DS-43xxHW-E 编码卡。设置视频输入场景模式，可选标准/室内/夜间/户外。

场景模式宏定义

#define VIDEO_MODE_STANDARD　　0 ；　标准

#define VIDEO_MODE_INDOOR　　1 ；　室内

#define VIDEO_MODE_DIM_LIGHT　　2 ；　夜间

#define VIDEO_MODE_OUTDOOR　　3 ；　户外

视频编码参数设置

释　义：　双编码功能(主、子通道)

对一路视频图像进行两路视频编码，两路视频可以有不同的码流类型、不同分辨率、不同码率等。3.0 版本对双编码功能做了增强，子通道的所有参数都可以任意设置。

双编码中主通道和子通道唯一的区别在于：子通道占用的系统资源比主通道少，优先级比主通道低。当系统忙时，会尽量保证主通道编码，并先从子通道开始丢帧。由于占用资源少，因此可以利用子通道来实现多路高分辨率的非实时编码。例如：可以把 DS-40xxHC 中的每个子通道全部设置为 4CIF 分辨率(SetSubStreamType)，而不使用主通道编码，这样就可以实现全部通道的 4CIF 编码。在一般场景下，每路图像都可以达到 15 帧以上。

6.3.24 主、子通道切换 SetupSubChannel

函　数：　int __stdcall SetupSubChannel(HANDLE hChannelHandle, int iSubChannel)

参　数：

HANDLE hChannelHandle；通道句柄

int iSubChannel；子通道号(0 表示主通道，1 表示子通道)

返回值：　成功返回 0；失败返回错误号

说　明：　配合双编码模式使用。当设置某个通道为双编码模式时，如主通道编码 CIF，子通道编码 QCIF，这时可对主/子通道分别设置某些参数。关键帧间隔、OSD、LOGO 等参数对主/子通道是一样的；在设置帧率、量化系数、变码流/定码流模式、码流大小等参数时应调用此函数分别对主/子通道进行设置，缺省是对主通道进行设置

6.3.25 获取双编码时数据流类型 GetSubChannelStreamType

函　数：　int __stdcall GetSubChannelStreamType(void　*DataBuf, int FrameType)

参　数：

void　*DataBuf；数据缓存区，此参数暂时无效，

int FrameType；帧类型

返回值：　0 其他数据

1 主通道数据流的文件头

2 子通道数据流的文件头

3 主通道数据流的视频帧类型

4 子通道数据流的视频帧类型

5 数据流的音频帧

说　明：　配合双编码模式使用，当设置双编码模式时，启动录像后，DSP 会向上送两种数据流，调用此函数得到主通道和子通道的数据流类型，供应用程序使用。

编码流类型的设置及获取(不支持动态修改)

6.3.26 设置主通道编码流类型 SetStreamType

函　数：　int __stdcall SetStreamType(HANDLE hChannelHandle, USHORT Type)

参　数：

HANDLE hChannelHandle；通道句柄

USHORT Type；流类型

返回值：　成功返回 0；失败返回错误号

说　明：　设置主通道编码流类型。此函数需在启动编码前进行设置。

流类型宏定义

#define　STREAM_TYPE_VIDEO　1；视频流

#define　STREAM_TYPE_AVSYNC　3；音视频复合流

6.3.27 获取主通道编码流类型 GetStreamType

函　数：　int __stdcall GetStreamType(HANDLE hChannelHandle, USHORT *StreamType)

参　数：

HANDLE hChannelHandle,；通道句柄

USHORT *StreamType；流类型

返回值：　成功返回 0；失败返回错误号

说　明：　获取主通道编码流类型

6.3.28 设置子通道编码流类型 SetSubStreamType

函　数：　int __stdcall SetSubStreamType(HANDLE hChannelHandle, USHORT Type)

参　数：

HANDLE hChannelHandle；通道句柄

USHORT Type；流类型

返回值：　成功返回 0；失败返回错误号

说　明：　设置子通道编码流类型，此函数需在启动编码前进行设置

6.3.29 获取子通道编码流类型 GetSubStreamType

函　数：　int __stdcall GetSubStreamType(HANDLE hChannelHandle, USHORT *StreamType)
参　数：
　　HANDLE hChannelHandle；通道句柄
　　USHORT *StreamType；流类型
返回值：　成功返回 0；失败返回错误号
说　明：　获取子通道编码流类型

(支持动态修改)的编码参数设置

6.3.30 设置编码图像质量 SetDefaultQuant

函　数：　int __stdcall SetDefaultQuant(HANDLE hChannelHandle, int IQuantVal, int PQuantVal, int BQuantVal)
参　数：
　　HANDLE hChannelHandle；通道句柄
　　int IQuantVal；I 帧量化系数，取值范围：12-30
　　int PQuantVal；P 帧量化系数。取值范围：12-30(目前无效)
　　int BQuantVal；B 帧量化系数。取值范围：12-30(目前无效)
返回值：　成功返回 0；失败返回错误号
说　明：　设置图像量化系数，用于调整图像质量。量化系数越小图像质量越高。系统默认量化系数值为 18，18，23。
释　义：量化系数
　　量化系数是强烈影响 MPEG 和 H.264 标准中编码图像质量和码率的参数，当量化系数越低，图像质量就会越高，码率也就越高，反之，图形质量就会越低，码率也就越低

6.3.31 设置编码帧结构、帧率 SetIBPMode

函　数：　int __stdcall SetIBPMode(HANDLE hChannelHandle, int KeyFrameIntervals,int BFrames, int PFrames, int FrameRate)
参　数：
　　HANDLE hChannelHandle；通道句柄
　　int KeyFrameIntervals；关键帧间隔。取值范围 1 -400，系统默认为 100
　　int BFrames；B 帧数量，取值为 0 或者 2，系统默认为 2
　　int PFrames；P 帧数量。目前暂取值无效
　　int FrameRate；帧率，帧率范围 1-25(PAL)、1-30(NTSC)
返回值：　成功返回 0；失败返回错误号
说　明：　设置编码帧结构和帧率，支持动态修改。其中 B 帧数量，对于 DS-40xx 和 DS-41xx 系列，取值为 0 或 2，系统默认为 2；对于 DS-42xx 和 DS-43xx 系列，取值为 0，系统默认为 0。

释　义：关键帧间隔

关键帧为编码码流中采用帧内压缩的图像帧，其特点是图像清晰度好，但数据量大，通常作为帧间编码的原始参考帧。关键帧间隔是连续的帧间编码的帧个数，因 H264 和 MPEG4 编码是有损压缩，关键帧的个数会影响图像质量，因此关键帧的间隔需要合理设计。

设置编码分辨率

6.3.32 设置主通道编码分辨率 SetEncoderPictureFormat

函　数：　int __stdcall SetEncoderPictureFormat(HANDLE hChannelHandle, PictureFormat_t PictureFormat)

参　数：

HANDLE hChannelHandle；通道句柄

PictureForamt_t PictureFormat；编码图像分辨率

返回值：　成功返回 0；失败返回错误号

说　明：　设置主通道编码分辨率。支持动态修改。

注　意：

DS-43xxHFV/HCV-E 系列支持 4CIF/2CIF/CIF/QCIF 编码

DS-43xxHW-E 系列支持 WD1/4CIF/2CIF/CIF/QCIF 编码

DS-43xxHFH-E 系列板卡除了支持 WD1/4CIF/2CIF/CIF/QCIF 编码，新增 HD_F、HD_H、HD_Q 三种分辨率，具体与视频输入制式相关，分别对应视频输入制式的原始尺寸、原始尺寸的一半、原始尺寸的 1/4，对应关系如下表：

分辨率 / 视频输入制式	HD_F	HD_H	HD_Q
1080P/1080I	1920*1080	1920*540	960*540
720P	1280*720	1280*360	640*360

其他系列板卡或者设备主通道支持 4CIF/DCIF/2CIF/CIF/QCIF 编码

6.3.33 设置子通道编码分辨率 SetSubEncoderPictureFormat

函　数：　int __stdcall SetSubEncoderPictureFormat(HANDLE hChannelHandle, PictureFormat_t PictureFormat)

参　数：

HANDLE hChannelHandle；子通道句柄

PictureFormat_t PictureFormat；子通道编码图像分辨率

返回值：　成功返回 0；失败返回错误号

说　明：　设置双编码模式时子通道的编码分辨率，支持动态修改。

注　意：

DS-52xx 系列设备、DS-42xx 系列板卡子通道支持 CIF/QCIF 编码。

DS-43xxHFV/HCV-E 系列板卡子通道支持 4CIF/2CIF/CIF/QCIF 编码

DS-43xxHW-E 系列板卡子通道支持 WD1/4CIF/2CIF/CIF/QCIF 编码

DS-43xxHFH-E 系列板卡除了支持 WD1/4CIF/2CIF/CIF/QCIF 编码，新增 HD_F、HD_H、HD_Q 三种

分辨率，具体与视频输入制式相关，分别对应视频输入制式的原始尺寸、原始尺寸的一半、原始尺寸的 1/4，对应关系如下表：

分辨率 / 视频输入制式	HD_F	HD_H	HD_Q
1080P/1080I	1920*1080	1920*540	960*540
720P	1280*720	1280*360	640*360

其他系列板卡子通道支持 4CIF/DCIF/2CIF/CIF/QCIF 编码

设置码率及码流控制模式

6.3.34 设置码流最大比特率 SetupBitrateControl

函　数：　int __stdcall SetupBitrateControl(HANDLE hChannelHandle, ULONG MaxBps)

参　数：

HANDLE hChannelHandle；通道句柄

ULONG Maxbps；最大比特率，单位 bps。取值：10000 以上

返回值：　成功返回 0；失败返回错误号

说　明：　设置编码的最大比特率。设置为 0 时码流控制无效，设置为某一最大比特率时，当编码码流超过该值时，DSP 会自动调整编码参数来保证不超过最大比特率，当编码码流低于最大比特率时，DSP 不进行干涉。调整误差<10%。

6.3.35 设置码流控制方式 SetBitrateControlMode

函　数：　int __stdcall SetBitrateControlMode(HANDLE hChannelHandle,BitrateControlType_t brc)

参　数：

HANDLE hChannelHandle；通道句柄

BitrateControlType_t brc；码流控制方式，分为变码率(brVBR)和恒定码率(brCBR)两种方式

返回值：　成功返回 0；失败返回错误号

说　明：　设置编码码流控制方式。配合 SetupBitrateControl 使用。当设置为变码率(brVBR)时，最大比特率将作为编码码流上限，由 DSP 在码流上限下自动控制码率，一般会自动回落到最低的状态(由设定的图像质量参数和关键帧间隔决定)，能最大程度地降低带宽和存储空间，但存储容量一般难以估算；当设置为定码率(brCBR)时，以最大比特率作为编码码率参数恒定输出码流，不会自动回落到低码流状态，存储容量可根据设定码率的大小进行估算。

6.3.36 强制设定 I 帧 CaptureIFrame

函　数：　int __stdcall CaptureIFrame(HANDLE hChannelHandle)

参　数：　HANDLE hChannelHandle；通道句柄

返回值：　成功返回 0；失败返回错误号

说　明：　将当前编码帧强制设定为 I 帧模式，可从码流中提取该帧单独用于网络传送。

6.3.37 获取帧统计信息 GetFramesStatistics

函　数：　int __stdcall GetFramesStatistics(HANDLE hChannelHandle, PFRAMES_STATISTICS framesStatistics)

参　数：

HANDLE hChannelHandle；通道句柄

PFRAMES_STATISTICS framesStatistics；帧统计信息

返回值：　成功返回 0；失败返回错误号

说　明：　获取帧统计信息

帧统计信息结构体

typedef struct tagFramsStatistics{

ULONG VideoFrames；开启编码后已经编码的视频帧数

ULONG AudioFrames；开启编码后已经编码的音频帧数

ULONG FramesLost；丢失帧

ULONG QueueOverflow；丢失的码流(字节)

ULONG CurBps；当前的码率(bps)

}FRAMES_STATISTICS, *PFRAMES_STATISTICS

设置码流封装格式

6.3.38 设置码流封装格式 SetStreamPackType

函　数：　int __stdcall SetStreamPackType(HANDLE hChannelHandle, USHORT Type)

参　数：

HANDLE hChannelHandle；通道句柄

USHORT Type；码流封装类型

返回值：　成功返回 0；失败返回错误号

说　明：　6.1 新增接口，适用于 DS-43xx 系列编码卡。设置主通道和子通道的码流封装类型，需在启动编码前进行设置。主通道和子通道的码流封装类型相同，主通道和子通道编码均没有启动编码时设置生效，否则设置不成功。默认为海康封装。

码流封装类型宏定义

#define STREAM_PACK_TYPE_HIKVISION　2 ；　海康封装

#define STREAM_PACK_TYPE_PS　3 ；　PS 封装

数据捕获

抓图(获取单帧图像数据)

抓取 BMP 格式图像

6.3.39 抓取固定分辨率的 YUV422 格式原始图像

GetOriginalImage

函　数：　int __stdcall GetOriginalImage(HANDLE hChannelHandle, UCHAR *ImageBuf, ULONG *Size)

参　数：

HANDLE hChannelHandle；通道句柄

UCHAR *ImageBuf；原始 yuv422 格式图像指针

ULONG *Size；原始 yuv422 格式图像尺寸，函数调用前是 ImageBuf 的大小，调用后是实际图像所占用的字节数

返回值：　成功返回 0，失败返回错误号

说　明：　抓取固定分辨率的 yuv422 格式图像。

注　意：

原始图像是 HD_F 格式：

DS-43xxHFH-E

原始图像是 WD1 格式:

DS-43xxHW-E

原始图像是 4CIF 格式：

DS-43xxHCV-E、DS-43xxHFV-E 板卡

DS-42xxHFV、DS-41xxHCV、DS-40xxHC/HC+/HF 板卡；

DS-5216HF、DS-5116HF 设备

原始图像是 CIF 格式：

DS-4216HC、DS-40xxHS 板卡

6.3.40 抓取编码通道指定分辨率的 YUV422 格式原始图像

GetEncoderOriginalImage

函 数：int __stdcall GetEncoderOriginalImage(HANDLE hChannelHandle, UCHAR *ImageBuf, ULONG *Size, PictureFormat_t picFormat);

参　数：

hChannelHandle ；通道句柄

ImageBuf；图像数据存放缓冲

Siz；缓冲大小

picFormat；图像分辨率

返回值：函数调用成功返回 0，失败则返回错误号。

说　明：　6.0 版本 SDK 新增功能函数，抓取编码通道指定分辨率的的 YUV422 格式原始图像

注　意：

下列型号的板卡支持 WD1/4CIF/2CIF/CIF/QCIF 以及 HD_F、HD_H、HD_Q 分辨率（详见第 2 章 SDK 版本更新）的抓图：

DS-43xxHFH-E

下列型号的板卡支持 WD1/4CIF/2CIF/CIF/QCIF 分辨率的抓图：

DS-43xxHW-E

下列型号的板卡支持 4CIF/2CIF/CIF/QCIF 分辨率的抓图：

DS-43xxHCV-E、DS-43xxHFV-E

下列型号的板卡（设备）支持 4CIF/2CIF/DCIF/CIF/QCIF/分辨率的 YUV422 抓图图像：

DS-42xxHFV、DS-41xxHCV、DS-40xxHF/HC/HC+板卡

DS-5216HF、DS-5116HF 设备。

下列型号的板卡支持 CIF/QCIF 分辨率的 YUV422 抓图图像：

DS-4216HC、DS-40xxHS 卡.

6.3.41 图像格式转换 YUVtoBMP SaveYUVToBmpFile

函　数：　int __stdcall SaveYUVToBmpFile(char *FileName, unsigned char *yuv, int Width,int Height)

参　数：

char *FileName；文件名

unsigned char *yuv；yuv422 格式图像指针

int Width；图像宽度

int Height；图像高度

返回值：　成功返回 0 ，失败返回错误号

说　明：　用户程序可调用此函数来生成 24 位的 bmp 文件

抓取 JPEG 格式图像

6.3.42 抓取编码通道固定分辨率的 JPEG 格式图像

GetJpegImage

函　数：　int __stdcall GetJpegImage(HANDLE hChannelHandle,UCHAR *ImageBuf,ULONG *Size, UINT nQuality)

参　数：

HANDLE hChannelHandle；通道句柄

UCHAR *ImageBuf；JPEG 图像指针

ULONG *Size；JPEG 图像尺寸，函数调用前是 ImageBuf 的大小，调用后是实际图像所占用的字节数

UINT nQuality；JPEG 图像质量，取值范围 1-100，取值 100 时质量最好

返回值：　成功返回 0，失败返回错误值

说　明：　抓取编码通道通道固定分辨率的 JPEG 格式图像

注　意：

JPEG 抓图图像是 HD_F 分辨率：

DS-43xxHFH-E

JPEG 抓图图像是 WD1 分辨率：

DS-43xxHW-E

JPEG 抓图图像是 4CIF 分辨率：

DS-43xxHCV-E、DS-43xxHFV-E 板卡

DS-42xxHFV、DS-41xxHCV、DS-40xxHC/HC+/HF 板卡；

DS-5216HF、DS-5116HF 设备。

JPEG 抓图图像是 CIF 分辨率：

DS-4216HC、DS-40xxHS 板卡

6.3.43 抓取编码通道指定分辨率的 JPEG 格式图像

GetEncoderJpegImage

函　数：　int __stdcall GetEncoderJpegImage(HANDLE hChannelHandle,UCHAR *ImageBuf,ULONG *Size, UINT nQuality,PictureFormat_t picFormat);

输入参数：

hChannelHandle；通道句柄

ImageBuf；图像数据存放缓冲

Size；缓冲大小

nQuality；图像质量

picFormat；图像分辨率

输出参数：

Size；JPEG 图像的大小

返回值：　成功返回 0，失败返回错误值

说　明：　6.0 版本 SDK 新增功能函数，抓取编码通道指定分辨率的的 JPEG 格式图像

注　意：

下列型号的板卡支持 WD1/4CIF/2CIF/CIF/QCIF 以及 HD_F、HD_H、HD_Q 分辨率（详见第 2 章 SDK 版本更新）的抓图：

DS-43xxHFH-E

下列型号的板卡支持 WD1/4CIF/2CIF/CIF/QCIF 分辨率的抓图：

DS-43xxHW-E

下列型号的板卡支持 4CIF/2CIF/CIF/QCIF 分辨率的抓图：

DS-43xxHCV-E、DS-43xxHFV-E

下列型号的板卡（设备）支持 4CIF/2CIF/DCIF/CIF/QCIF 分辨率的 JPEG 抓图：

DS-42xxHFV、DS-41xxHCV、DS-40xxHF/HC/HC+板卡、

DS-5216HF、DS-5116HF 设备

下列型号的板卡支持 CIF/QCIF 分辨率的 JPEG 抓图：

DS-4216HC、DS-40xxHS 卡.

原始图像数据流捕获(获取 YUV420 格式数据流)

6.3.44 注册原始图像数据流回调函数

RegisterImageStreamCallback

函　数：　int __stdcall RegisterImageStreamCallback
(IMAGE_STREAM_CALLBACK ImageStreamCallback,void *context)
参　数：
　　IMAGE_STREAM_CALLBACK；原始图像数据流回调函数
　　void *context；设备上下文
返回值：　成功返回 0；失败返回错误号
说　明：　注册获取原始图像数据流函数，用户可以获取实时的 YUV420 格式的预览数据

原始图像数据流回调函数

typedef void (*IMAGE_STREAM_CALLBACK)(UINT channelNumber,void *context)
　　UINT channelNumber；通道号
　　void *context；设备上下文

6.3.45 开启及停止原始数据流捕获 SetImageStream

函　数：　int __stdcall SetImageStream(HANDLE hChannel,BOOL bStart,UINT fps,UINT width,
UINT height,unsigned char *imageBuffer)
参　数：
　　HANDLE hChannelHandle；通道句柄
　　BOOL bStart；是否启动捕获
　　UINT fps；帧率
　　UINT width；图像宽度
　　UINT heigh；图象高度
　　unsigned char *imageBuffer；数据存储缓存
返回值：　成功返回 0；失败返回错误号
说　明：　开启或停止原始图像数据流捕获，此函数依赖主机的处理速度。

DS-43xxHFH-E：如果视频输入为 1080P/1080I，支持 1920*1080、960*540、WD1 或者 4CIF 以及 4CIF 的 1/8、1/4、1/2 的原始图像捕获；如果视频输入为 720P，支持 1280*720、640*360、WD1 或者 4CIF 以及 4CIF 的 1/8、1/4、1/2 的原始图像捕获。

DS-43xxHW-E：支持 WD1、4CIF 以及 4CIF 的 1/8、1/4、1/2 的原始图像捕获。

DS-40xxHS、DS-4216HC 板卡只能捕获不大于 CIF 格式的数据流，宽高须为 CIF 宽高的 1/8、1/4、1/2、1 倍。

其他 DS-4xxx 系列板卡、DS-5xxx 系列设备最大捕获支持 4CIF 格式数据流，宽高须为 4CIF 宽高的 1/8、1/4、1/2、1 倍

编码数据流捕获即录像(获取编码后 H.264 格式数据流)

编码数据流捕获方式设置

注　意：注册直接回调或者消息回调后，需要启动编码数据流捕获函数才能进行数据回调。三种数据回调方式，只需选取其中一种使用即可。对于 DS-4xxx 系列板卡和 DS-5xxx 系列设备，推荐使用第一种读取方式。对于 H 系列板卡，只能使用后两种读取方式。

方式一、直接读取方式

6.3.46 注册编码图像数据流直接读取回调函数

RegisterStreamDirectReadCallback

函　数：　int __stdcall RegisterStreamDirectReadCallback
(STREAM_DIRECT_READ_CALLBACK StreamDirectReadCallback,void *Context)
参　数：
　　STREAM_DIRECT_READ_CALLBACK StreamDirectReadCallback；编码数据流直接读取回调函数
　　void* Context；设备上下文
返回值：　成功返回 0；失败返回错误号
说　明：　DS-40xxHC 系列板卡新增的一种数据流读取方式，当启动数据捕获后，StreamDirectReadCallback 会提供数据流的地址、长度、帧类型等，供用户程序直接处理。

编码数据流直接读取回调函数

typedef int (*STREAM_DIRECT_READ_CALLBACK)(ULONG channelNumber,void *DataBuf,
DWORD Length,int FrameType,void *context)
　　ULONG channelNumber；通道号
　　void* DataBuf；缓冲区地址
　　DWORD Length；缓冲区长度
　　int FrameType；缓冲区数据帧类型
　　void* context；设备上下文

方式二、消息读取方式

6.3.47 设置消息读取阈值 SetupNotifyThreshold*

此函数只对 H 卡有效
函　数：　int __stdcall SetupNotifyThreshold(HANDLE hChannelHandle, int iFramesThreshold)
参　数：

HANDLE hChannelHandle；通道句柄

int iFramesThreshold；读取消息阈值，范围 1-10

返回值： 成功返回 0；失败返回错误号

说 明： 设置消息读取的阈值，可以将缓冲区的数据在 OnDataReady 中一次性取走

6.3.48 注册消息读取码流函数 RegisterMessageNotifyHandle

函 数： int __stdcall RegisterMessageNotifyHandle(HWND hWnd, UINT MessageId)

参 数：

HWND hWnd；通道句柄

UINT MessageId；自定义消息

返回值： 成功返回 0；失败返回错误号

说 明： 当数据准备好时，SDK 会向 hWnd 窗口发送 MessageId 消息，目标窗口收到 Message 后调用 ReadStreamData 读取一帧数据。如果其他编码卡与 H 卡混插，可以先调用 RegisterStreamDirectReadCallback 函数来注册其他编码卡取码流回调函数，再调用 RegisterMessageNotifyHandle 函数来注册 H 卡取码流消息函数。

DS-4xxx 系列板卡和 DS-5xxx 系列设备建议使用方式一进行数据捕获。

5.1 以上 SDK 不支持 H 卡

方式三、另一种直接读取方式

6.3.49 注册直接读取码流回调函数 RegisterStreamReadCallback

函 数： int __stdcall RegisterStreamReadCallback(STREAM_READ_CALLBACK StreamReadCallback, void *Context)

参 数：

STREAM_READ_CALLBACK StreamReadCallback；直接读取码流回调函数

void *Context；设备上下文

返回值： 成功返回 0；失败返回错误号

说 明： 另一种数据流读取方式

直接读取码流回调函数

typedef int (*STREAM_READ_CALLBACK)(ULONG channelNumber, void *context)

ULONG channelNumber；通道号

void *context；设备上下文

6.3.50 读取码流函数 ReadStreamData

函 数： int __stdcall ReadStreamData(HANDLE hChannelHandle, void *DataBuf, DWORD *Length, int *FrameType)

参 数：

HANDLE hChannelHandle；通道句柄

void *DataBuf；自定义的数据缓存区

DWORD * Length；输入：缓存区的大小；输出：一帧数据的大小

int *FrameType；帧类型

返回值： 成功返回 0；失败返回错误号

说 明： 读指定长度的数据流，适用于方式二及方式三。当调用 StartVideoCapture 或 StartMotionDetection 后，SDK 线程会向已注册的用户窗口消息处理函数发送指定的消息，并提供消息来源的通道号。当用户程序收到该消息时，可调用本函数来读取数据，Length 在作为输入时必须提供缓冲的大小，ReadStreamData 会判断缓冲是否足够，如果缓冲足够大，则返回值为当前的读取的帧长度，否则返回错误。

在 DS-4xxx 系列板卡和 DS-5xxx 系列设备中，如果已经先调用了 RegisterStreamDirectReadCallback() 函数，则不需调用 ReadStreamData 来读取数据，因为音视频数据会在 RegisterStreamDirectReadCallback 所注册的回调函数中直接返回。

开启及停止录像

6.3.51 启动主通道编码数据流捕获 StartVideoCapture

函 数： int __stdcall StartVideoCapture(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说 明： 启动主通道编码数据流捕获。用户程序可以使用直接读取方式，使用 StreamDirectReadCallback 回调函数直接对数据流进行处理；也可以通过消息读取方式，等 SDK 向用户程序发送在 RegisterMessageNotifyHandle 中注册的消息，用户程序使用 ReadStreamData 来读取数据流。

6.3.52 停止主通道编码数据流捕获 StopVideoCapture

函 数： int __stdcall StopVideoCapture(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说 明： 停止主通道编码数据流捕获

6.3.53 启动子通道编码数据流捕获 StartSubVideoCapture

函 数： int __stdcall StartSubVideoCapture(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说 明： 启动子通道编码数据流捕获

6.3.54 停止子通道编码数据流捕获 StopSubVideoCapture

函　数：　int __stdcall StopSubVideoCapture(HANDLE hChannelHandle)
参　数：　HANDLE hChannelHandle；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止子通道编码数据流捕获

移动侦测

释　义：　移动侦测

DS-40xxHC 提供运动强度信息来处理运动检测，设置移动侦测区域时以 32*32 像素块为单位，按 4CIF(704*576)分辨率计算，每行有 22 个块(704/32)，PAL 时 18 行(576/32)，NTSC 时 15 行(480/32)，与编码格式无关。经过测试，这种方法比 H 卡提高了灵敏度和可靠性，并简化了返回的数据，返回的值是 18 个 DWORD，PAL 制时对应屏幕高度 576/32=18 行，NTSC 制时对应屏幕高度 480/32=15 行，,每个 DWORD 的 0-21 位对应屏幕宽度 704/32=22, 其中 1 为运动,0 为静止，也可以调用原有 MotionAnalyzer 分析结果

4.0 版本的 SDK 新增了接口函数 SetupMotionDetectionEx，提供了更灵活的功能，并且简化了用户的工作量。

设置方式一

设置移动侦测相关参数并启动移动侦测后，运动检测信息会通过数据流传送，用户程序发现 PktMotionDetection 帧类型时，可调用 MotionAnalyzer 来处理运动信息，结果由 MotionAnalyzer 在 iResult 中返回。也可以按照 SDK 提供的数据格式来自己分析，运动信息格式参见移动侦测释义。

6.3.55 设置移动侦测灵敏度 AdjustMotionDetectPrecision

函　数：　int __stdcall AdjustMotionDetectPrecision(HANDLE hChannelHandle, int iGrade, int iFastMotionDetectFps, int iSlowMotionDetectFps)
参　数：
HANDLE hChannelHandle；通道句柄
int iGrade；运动分析灵敏度等级，取值范围 0-6，0 级最灵敏，6 级最迟钝，推荐值为 2。将 iGrade 和“0x80000000“做”或“操作，会对移动侦测启用自适应分析。
int iFastMotionDetectFps；高速运动检测的帧间隔，取值范围 0-99，通常值取为 2
int iSlowMotionDetectFps；低速运动检测的帧间隔，取值范围 1-100，且取值必须要大于 iFastMotionDetectFps
返回值：　成功返回 0；失败返回错误号
说　明：　调整运动分析的灵敏度，支持动态调整，此函数决定 DSP 全局运动分析的灵敏度。而 MotionAnalyzer 的 iThreshold 则主要用于分析指定区域的运动统计结果。

6.3.56 设置移动侦测区域范围及个数 SetupMotionDetection

函　数：　int __stdcall SetupMotionDetection(HANDLE hChannelHandle, RECT *RectList,int iAreas)
参　数：
HANDLE hChannelHandle；通道句柄
RECT *rectList；矩形框数组
int numberOfAreas；矩形框个数，最大值为 100
返回值：　成功返回 0；失败返回错误号
说　明：　设置运动检测区域；当收到运动信息的数据帧(PktMotionDetection)时,调用 MotionAnalyzer；MotionAnalyzer 会根据在 SetupMotionDetection 中的设置来分析每个需要检测的区域，当某区域的阀值(MotionAnalyzer 的 iThreshold)到达时，会在返回的区域数组(MotionAnalyzer 的 iResult)标明最后的判断；矩形框范围是(0，0，703，575)。

6.3.57 移动侦测分析 MotionAnalyzer

函　数：　int __stdcall MotionAnalyzer(HANDLE hChannelHandle, char *MotionData, int iThreshold, int *iResult)
参　数：
HANDLE hChannelHandle；通道句柄
char *MotionData；运动矢量指针
int iThreshold；判断运动程度的阈值
int *iResult；按照阈值指定的强度分析后的结果数组。数组的大小在 SetupMotionDetection 的 numberOfAreas 指定，如果某数组单元的值大于零则表明在该区域有该值表明的运动强度
返回值：　成功返回 0；失败返回错误号
说　明：　动态监测分析，移动侦测由 DSP 完成，DSP 送出的 IPktMotionData 帧就是已经分析好的运动信息，区域的运动分析由主机完成，数据源由码流中的 PktMotionData 帧提供，结果在 iResult 中说明，2.0 以上版本的运动分析基于 DSP 提供的运动强度，不再使用运动矢量，其灵敏度及可靠性有大幅提高，用户软件可使用由码流提供的运动强度信息来自己分析或调用该函数来进行区域分析

设置方式二

6.3.58 设置移动侦测(扩展)SetupMotionDetectionEx

函　数：　int __stdcall SetupMotionDetectionEx(HANDLE hChannelHandle,int iGrade, int iFastMotionDetectFps, int iSlowMotionDetectFps,UINT delay,RECT *RectList, int iAreas, MOTION_DETECTION_CALLBACK MotionDetectionCallback,int reserved)
参　数：
HANDLE hChannelHandle；通道句柄
int iGrade；运动分析灵敏度等级，取值范围 0-6，0 级最灵敏，6 级最迟钝，推荐值 2。将 iGrade 和“0x80000000“做”或“操作，会对移动侦测启用自适应分析。

int iFastMotionDetectFps；高速运动检测的帧间隔，取值范围 0-99，通常值取为 2

int iSlowMotionDetectFps；低速运动检测的帧间隔，取值范围 1-100，且取值必须要大于 iFastMotionDetectFps

UINT delay；前一次移动帧测产生后的延时时间

RECT *RectList；进行移动侦测的矩形框数组

int iAreas；矩形框个数(最大个数为 100)

MOTION_DETECTION_CALLBACK MotionDetectionCallback；移动侦测结果回调函数

int reserved；保留参数

返回值： 成功返回 0；失败返回错误号

说 明： 设置移动侦测，这种设置方式可替代设置方式一中 3 个函数的共同作用，在这种情况下，DSP 将不再反馈移动侦测帧。

UINT delay：画面静止之后的延时时间，单位为秒，若在该延时时间内没有产生移动侦测，则将回调函数 MotionDetectionCallback 之中的参数 bMotionDetected 标志为 False，若在该延时时间之内，在当前所设置的区域内产生移动侦测，则 bMotionDetected 被标志为 True，并且在产生移动侦测之后的 delay 时间内，DSP 不会对在此时间段之内的视频帧进行移动侦测分析，因此 DSP 和主机都省却了在此时间段对产生的视频运动进行频繁判断和分析。直至超过了此 delay 秒延时时间，DSP 才会对此时刻的视频进行判断，若产生了移动侦测，则回调函数中的 bMotionDetected 被再次标志为 True，否则标志为 False。

移动侦测结果回调函数

typedef void (*MOTION_DETECTION_CALLBACK)(ULONG channelNumber,BOOL bMotionDetected, void *context)

ULONG channelNumber；通道号

BOOL bMotionDetected；移动侦测发生标志，如果当前通道所设置的移动侦测区域内产生了移动侦测，则被置为 True；如果当前通道所设置的移动侦测区域内自上一次产生移动侦测后 delay 秒内没有发生移动侦测，则被置为 False。

Void *context；设备上下文

启动及停止移动侦测

6.3.59 启动移动侦测 StartMotionDetection

函 数： int __stdcall StartMotionDetection(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说 明： 启动移动侦测。

注 意： 移动侦测和编码相互独立，用户程序可在不启动编码的情况下进行运动检测

6.3.60 停止移动侦测 StopMotionDetection

函 数： int __stdcall StopMotionDetection(HANDLE hChannelHandle)

参 数： HANDLE hChannelHandle；通道句柄

返回值： 成功返回 0；失败返回错误号

说　明：　停止移动侦测

视频信息叠加

信息叠入视频编码(OSD、LOGO、MASK)

注 意：使用此部分函数时，在录像文件中，包含所叠加的信息

OSD

6.3.61 设置 OSD 显示模式 SetOsdDisplayMode

函　数：　int __stdcall SetOsdDisplayMode(HANDLE hChannelHandle, int Brightness, BOOL Translucent, int parameter, USHORT *Format1, USHORT *Format2)

参　数：

HANDLE hChannelHandle；通道句柄

int Brightness；OSD 显示亮度，取值范围 0~255。对于 DS-52xx 系列设备或者 DS-42xx 系列板卡，亮度值≥128 时为白色(235)，<128 时为黑色(16)；对于其他设备或板卡，可以调节亮度，0 最暗，255 最亮。

BOOL Translucent；　OSD 图像是否做半透明处理

int param；Bit0：是否自动进行颜色翻转 Bit16-23 垂直放大倍数 Bit24-31 水平放大倍数

USHORT *Format1；　描述字符叠加的位置和次序的格式

USHORT *Format2；　描述字符叠加的位置和次序的格式

返回值：　成功返回 0；失败返回错误号

说　明：

OSD 字符中，ASCII 字符的标准分辨率为 8×16，汉字的标准分辨率为 16×16。由于在编码之前需要对原始图像进行缩小才能产生编码所需的分辨率，此时为了保证在缩小后的编码图像上能够看清 OSD 字符，就需要先把 OSD 字符放大以后再叠加在 4CIF 的原始图像上。

如果不指定放大倍数(采用默认设置)，则系统会根据该通道录像的分辨率自动设置，这样在任何分辨率下都可以保证回放时能够看清 OSD 内容，但是这会导致 OSD 的大小和位置在原始图像中不固定。

为了避免上面的现象，用户可以指定 OSD 的大小。例如，如果应用程序想以 CIF、DCIF、2CIF、4CIF 的分辨率录像，这时候可以将放大系数设为 2、2，此时 OSD 的位置始终固定，但在不同的编码分辨率下，OSD 字符的分辨率也不同，所以需要特别注意。如果此时使用 QCIF 录像，则 OSD 字符会变得模糊不清(因为 QCIF 要对图像进行 1/4 缩小，而对 OSD 字符只进行了 2 倍的放大)。具体配置详见下表：

水平放大倍数	垂直放大倍数	适合的录像分辨率	说明
1	1	4CIF、WD1	其它分辨率下会模糊
1	2	2CIF	小于 2CIF 时无法分辨
2	2	CIF、DCIF	QCIF 时无法分辨
4	4	QCIF	在其它分辨率下字符会很大
任意系数为 0		自动设置(默认值)	
其它无效值		按水平 2、垂直 2 处理	

DS-43xxHFH-E 新增 8*8 放大倍数，具体推荐配置如下所示：

水平放大倍数	垂直放大倍数	适合的录像分辨率
1	1	HD_H
1	2	HD_F
2	2	4CIF、HD_Q
2	4	2CIF
4	4	CIF
8	8	QCIF
任意系数为 0		自动设置(默认值)
其它无效值		按水平 2、垂直 2 处理

注意：因为字符的位置会随着不同的录像分辨率而改变，在位置改变后，某些 OSD 字符的位置可能会超出图像的范围，此时这些字符将无法显示，但系统并不会返回错误。

USHORT *Forma1, *Format2

描述字符叠加的位置和次序的格式串，具体定义如下：

USHORT X， USHORT Y， CHAR0， CHAR1， CHAR2，… CHARN， NULL

其中 X，Y 是该字串在标准 4CIF 图象的起始位置，X 必须是 16 的倍数，Y 可以在图象高度内取值即(0-575)PAL 、(0-479)NTSC，DS-52xx 系列设备、DS-42xx 系列板卡的 Y 值需要按照 16 对齐，如果取值未对齐会作自动调整；CHARN 也是 USHORT 型的参数，可以是 ASCII 码也可以是汉字，当想要显示当前时间时，可以指定为固定的时间定义值，其值如下：

_OSD_YEAR4 四位的年显示，如 2004
_OSD_YEAR2 两位的年显示，如 02
_OSD_MONTH3 英文的月显示，如 Jan
_OSD_MONTH2 两位阿拉伯数字的月显示，如 07
_OSD_DAY 两位的阿拉伯数字的日显示，如 31
_OSD_WEEK3 英文的星期显示，如 Tue
_OSD_CWEEK1 中文的星期显示，如星期二
_OSD_HOUR24 24 小时的时钟显示，如 18
_OSD_HOUR12 12 小时的时钟显示，如 AM09 或 PM09
_OSD_MINUTE 两位分钟的显示
_OSD_SECOND 两位秒的显示
_OSD_APM 两位上下午，AM 和 PM

在格式字符串的最后必须以 NULL(0)结尾，否则会显示错误的内容。

字符串和时间显示可以在 FORMAT1 也可以在 FORMAT2，也可以混合在一起，但不得超过一行 4CIF 图象的宽度。

例如：

要显示位置在 16，19 的字符串“办公室”的格式字符串如下：USHORT Format[] = {16, 19, ‘办’,’公’,’室’, ‘\0’};

要显示位置在 8，3 的时间字符串可以如下：USHORT Format[]={8, 3, _OSD_YEAR4, ‘年’,_OSD_MONTH2,’月’,_OSD_DAY, ‘日’,_OSD_HOUR24,’:’, _OSD_MINUTE, ‘:’, _OSD_SECOND, ‘\0’};

如果只想显示其中一行，则将起始的字符串定义如下：USHORT FormatNoDisplay[]={0, 0, ‘\0’};

6.3.62 设置 OSD 显示模式(扩展)SetOsdDisplayModeEx

函　数：　int __stdcall SetOsdDisplayModeEx(HANDLE hChannelHandle,int color,BOOL Translucent, int param,int nLineCount,USHORT **Format)

参　数：

HANDLE hChannelHandle；通道句柄

int Brightness；OSD 显示亮度，取值范围 0~255。对于 DS-52xx 系列设备或者 DS-42xx 系列板卡，亮度值≥128 时为白色(235)，<128 时为黑色(16)；对于其他设备或板卡，可以调节亮度，0 最暗，255 最亮。

BOOL Translucent；OSD 图像是否做半透明处理

int param；Bit0：是否自动进行颜色翻转 Bit16-23 垂直放大倍数 Bit24-31 水平放大倍数

int nLineCount；OSD 显示的行数，最多为 8 行

USHORT **Format；多行字符显示

返回值：　成功返回 0；失败返回错误号。

说　明：　此函数为 SetOsdDisplayMode 的扩展，SetOsdDisplayModeEx 函数支持最多 8 行 OSD 字符串的显示。

USHORT **Format；多行字符显示，描述字符叠加的位置和次序的格式串，其中每一行的第一元素 X 和第二元素 Y 是该字串在标准 4CIF 图象的起始位置，X 必须是 16 的倍数，Y 可以在图象高度内取值即(0-575)PAL 、(0-479)NTSC，DS-52xx 系列设备、DS-42xx 系列板卡的 Y 值需要按照 16 对齐，如果取值未对齐会作自动调整；可以是 ASCII 码也可以是汉字，当想要显示当前时间时，可以指定为固定的时间定义值，其值如下：

_OSD_YEAR4	四位的年显示，如 2004
_OSD_YEAR2	两位的年显示，如 02
_OSD_MONTH3	英文的月显示，如 Jan
_OSD_MONTH2	两位阿拉伯数字的月显示，如 07
_OSD_DAY	两位的阿拉伯数字的日显示，如 31
_OSD_WEEK3	英文的星期显示，如 Tue
_OSD_CWEEK1	中文的星期显示，如星期二
_OSD_HOUR24	24 小时的时钟显示，如 18
_OSD_HOUR12	12 小时的时钟显示，如 AM09 或 PM09
_OSD_MINUTE	两位分钟的显示
_OSD_SECOND	两位秒的显示
_OSD_APM	两位上下午，AM 和 PM

在格式字符串的每一行最后一个元素必须以 NULL(0)结尾，否则会显示错误的内容。

6.3.63 设置编码通道的 OSD 显示模式 SetEncoderOsdDisplayMode

函　数：int __stdcall　SetEncoderOsdDisplayMode(HANDLE hChannelHandle, BOOL bEnableOsd, UINT nLuminance, UINT nFlash, COLORREF CharacterColor, COLORREF BackGroundColor, UINT nTranslucentMode, BOOL bAutoAdjustLum, int *param, int nLineCount, UINT **Format);

参　数：

hChannelHandle ；通道句柄

bEnableOsd；开启或关闭 OSD

nLuminance；OSD 亮度值，取值范围[0,255]

nFlash；OSD 闪烁，Bit0-7 停止秒数，Bit8-15 显示秒数；例如，flash＝(2<<8)|1 表示 Logo 显示 2 秒，停止 1 秒

CharacterColor；OSD 字符的颜色，支持 RGB 格式；如果值为-1 则为普通模式，OSD 字符用白色

BackGroundColor；OSD 背景的颜色，支持 RGB 格式；如果值为-1 则为普通模式，背景无颜色

nTranslucentMode；是否半透明,0 前景和背景都不透明，1 前景半透明且背景不透明，2 前景不透明且背景半透明，3 前景和背景都半透明

bAutoAdjustLum ；是否自动调整亮度，自动调整时，指定的亮度值无效

param；int 型的数组，设定每一行 OSD 的水平和垂直放大倍数，Bit0-7 垂直放大倍数 Bit8-15 水平放大倍数

nLineCount；OSD 显示的行数，最多为 8 行

Format；描述多行字符的叠加位置和次序的格式，其中每一行的第一元素 X 和第二元素 Y 是该字串在标准 4CIF 图象的起始位置，X 必须是 16 的倍数，Y 可以在图象高度内取值即(0-575)PAL 、(0-479)NTSC，DS-52xx 系列设备、DS-42xx 系列板卡 Y 值需要按照 16 对齐，如果取值未对齐会作自动调整；可以是 ASCII 码也可以是汉字，当想要显示当前时间时，可以指定为固定的时间定义值，其值如下：

_OSD_YEAR4_EX	四位的年显示，如 2004
_OSD_YEAR2_EX	两位的年显示，如 02
_OSD_MONTH3_EX	英文的月显示，JAN~DEC
_OSD_MONTH2_EX	两位阿拉伯数字的月显示，01~12
_OSD_DAY_EX	两位的阿拉伯数字的日显示，01~31
_OSD_WEEK3_EX	英文的星期显示，MON~SUN
_OSD_CWEEK1_EX	中文的星期显示，星期一~星期日
_OSD_HOUR24_EX	24 小时的时钟显示，00~23
_OSD_HOUR12_EX	12 小时的时钟显示，00~12
_OSD_MINUTE_EX	两位分钟的显示，00~59
_OSD_SECOND_EX	两位秒的显示，00~59
_OSD_MILISECOND_EX	3 位毫秒的显示，000~999
_OSD_APM_EX	2 位上下午，AM 和 PM

在格式字符串的每一行最后一个元素必须以 NULL(0)结尾，否则会显示错误的内容。

返回值：　成功返回 0；失败返回错误号。

说　明：　6.0 版本新增函数，支持闪烁，可以设置 OSD 字符颜色和背景颜色，字符和背景支持半透明，支持 8 行字符且每行可以独立设置放大倍数，支持自动调整亮度。

OSD 字符中，ASCII 字符的标准分辨率为 8×16，汉字的标准分辨率为 16×16。由于在编码之前需要对原始图像进行缩小才能产生编码所需的分辨率，此时为了保证在缩小后的编码图像上能够看清 OSD 字符，就需要先把 OSD 字符放大以后再叠加在 4CIF 的原始图像上。

如果不指定放大倍数(采用默认设置)，则系统会根据该通道录像的分辨率自动设置，这样在任何分辨率下都可以保证回放时能够看清 OSD 内容，但是这会导致 OSD 的大小和位置在原始图像中不固定。

为了避免上面的现象，用户可以指定 OSD 的大小。例如，如果应用程序想以 CIF、DCIF、2CIF、4CIF 的分辨率录像，这时候可以将放大系数设为 2、2，此时 OSD 的位置始终固定，但在不同的编码分辨率下，

OSD 字符的分辨率也不同，所以需要特别注意。如果此时使用 QCIF 录像，则 OSD 字符会变得模糊不清(因为 QCIF 要对图像进行 1/4 缩小，而对 OSD 字符只进行了 2 倍的放大)。具体配置详见下表：

水平放大倍数	垂直放大倍数	适合的录像分辨率	说明
1	1	4CIF、WD1	其它分辨率下会模糊
1	2	2CIF	小于 2CIF 时无法分辨
2	2	CIF、DCIF	QCIF 时无法分辨
4	4	QCIF	在其它分辨率下字符会很大
任意系数为 0		自动设置(默认值)	
其它无效值		按水平 2、垂直 2 处理	

DS-43xxHFH-E 新增 8*8 放大倍数，具体推荐配置如下所示：

水平放大倍数	垂直放大倍数	适合的录像分辨率
1	1	HD_H
1	2	HD_F
2	2	4CIF、HD_Q
2	4	2CIF
4	4	CIF
8	8	QCIF
任意系数为 0		自动设置(默认值)
其它无效值		按水平 2、垂直 2 处理

注　意：　DS-4200 系列板卡不支持显示彩色 OSD 字符和背景色。

6.3.64 设置 OSD 显示 SetOsd

函　数：　int __stdcall SetOsd(HANDLE hChannelHandle, BOOL Enable)

参　数：

HANDLE hChannelHandle；通道句柄

BOOL Enable；OSD 是否显示

返回值：　成功返回 0；失败返回错误号

说　明：　设置 OSD 显示，将当前的系统时间年月日星期时分秒等信息叠加在视频之上，并可作透明处理。

LOGO

6.3.65 数据格式转换(bmp 转 yuv422)LoadYUVFromBmpFile

函　数：　int __stdcall LoadYUVFromBmpFile(char *FileName, unsigned char *yuv, int BufLen, int *Width, int *Height)

参　数：

char *FileName；文件名

unsigned char *yuv；YUV422 图像指针

int BufLen；YUV422 图像缓存大小

int *Width；返回的 YUV422 图像的宽度

int *Height；返回的 YUV422 图像的高度

返回值： 成功返回 0；失败返回错误号

说 明： 将 24 位 bmp 格式图像转换为 yuv422 格式图像。

注 意： bmp 位图的长宽必须为 16 的倍数，图像尺寸最大支持 128*128，

4.3 版本 SDK 起图像尺寸扩大为 256*128；

6.0 版本起 SDK，

Logo 最大支持 4CIF：

DS-43xx、DS-42xx、DS-41xx、DS-40xxHC/HC+/HF/HCS/HS 板卡

DS-5216HF、DS-5116HF 设备

Logo 最大支持 CIF：

DS-4216HC 卡

6.3.66 设置 LOGO 显示模式 SetLogoDisplayMode

函 数： int __stdcall SetLogoDisplayMode(HANDLE hChannelHandle, COLORREF ColorKey, BOOL Translucent, int TwinkleInterval)

参 数：

HANDLE hChannelHandle；通道句柄

COLORREF ColorKey；LOGO 图像中该颜色在显示时将会全透明

BOOL Translucent；LOGO 图像是否作半透明处理

int Twinkle；闪烁的时间设置，由 16 进制数表示为 0xXXYY，其中 XX 为显示的秒数，YY 为停止的秒数，XXYY 均为 0 时正常显示。

返回值： 成功返回 0；失败返回错误号

说 明： 设置 LOGO 显示模式

6.3.67 设置 LOGO 显示位置及数据 SetLogo

函 数： int __stdcall SetLogo(HANDLE hChannelHandle, int x, int y, int w, int h, unsigned char *yuv)

参 数：

HANDLE hChannelHandle；通道句柄

int x；LOGO 左上角 x 坐标位置，取值范围 0-703，需按 16 对齐

int y；LOGO 左上角 y 坐标位置，取值范围 0-575，需按 8 对齐

int w；LOGO 宽度，最大值为 256，需按 16 对齐，此宽度必须和 LOGO 图片的宽度相一致

int h；LOGO 高度，最大值为 128，需按 8 对齐

unsigned char *yuv；LOGO 图片指针(yuv422 格式)

返回值： 成功返回 0；失败返回错误号

说 明： 设置 LOGO 图像位置及数据，用户程序可先调用 LoadYUVFromBmpFile 将 24 位 bmp 文件中转化为 YUV422 格式数据，透明处理由 DSP 完成。

注 意： DS-40xxHS 卡的 x 需要 16 对齐，w 需要 32 对齐，y，h 需要 8 对齐；DS-52xx 系列设备、DS-42xx 系列板卡 Logo 的宽高需要按照 16*16 对齐。

6.3.68 停止 LOGO 显示 StopLogo

函　数：　int __stdcall StopLogo(HANDLE hChannelHandle)
参　数：　HANDLE hChannelHandle；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止 LOGO 显示

视频遮挡 MASK

6.3.69 设置屏幕遮挡 SetupMask

函　数：　int __stdcall SetupMask(HANDLE hChannelHandle, RECT *RectList, int iAreas)
参　数：
HANDLE hChannelHandle；通道句柄
RECT *rectList；矩形框数组，宽度范围 0-704，按 16 对齐；高度范围 0-576，按 8 对齐。
int iAreas；矩形框个数
返回值：　成功返回 0；失败返回错误号
说　明：　设置屏幕遮挡，最多可以设置 32 个
注　意：　DS-52xx 系列设备、DS-42xx 系列板卡 Mask 的宽高需要按照 16*16 对齐。

6.3.70 设置视频遮挡,可以针对每个遮挡区域设置颜色 SetupEncoderMask

函　数：　int __stdcall SetupEncoderMask(HANDLE hChannelHandle, int iAreaCount,MASK_AREA_PARAM* pAreaParam);
参　数：
hChannelHandle ；通道句柄
iAreaCount；遮挡的个数,为 0 时关闭视频遮挡功能
pAreaParam；遮挡区域参数，见 MASK_AREA_PARAM 结构
返回值：　成功返回 0；失败返回错误号
说　明：　6.0 版本新增函数，最多可以设置 32 个
注　意：　对 DS-41xxHCV、DS-40xxHF/HC/HC+/HCS/HS 板卡有效；对 DS-42xx 系列板卡无效。

遮挡区域结构体

```
typedef struct
{
    UINT left;          遮挡区域左上角 X 轴坐标
    UINT top;           遮挡区域左上角 Y 轴坐标
    UINT width;         遮挡区域宽度
    UINT height;        遮挡区域高度
    COLORREF color;     遮挡区域颜色
```

}MASK_AREA_PARAM;

6.3.71 停止屏幕遮挡 StopMask

函　数：　int __stdcall StopMask(HANDLE hChannelHandle)
参　数：　HANDLE hChannelHandle；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止屏幕遮挡

音频

6.3.72 设置音频预览 SetAudioPreview

函　数：　int __stdcall SetAudioPreview(HANDLE hChannelHandle, BOOL bEnable)
参　数：
　　HANDLE hChannelHandle；通道句柄
　　BOOL bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　设置音频预览与否，同一时间，系统只支持一路音频预览。使用该接口时，DS-40xx 系列板卡需要将 4 针线和主板上的声卡音频输入口连接；其他板卡或设备不需要连线，可以直接预览输出。对于 DS-40xx 系列板卡，如果不连线到主板，可以调用 SetDirectAudioPreview；对于其他板卡或设备 SetAudioPreview 和 SetDirectAudioPreview 具有完全一样的功能。

6.3.73 获取音频输入音量幅度 GetSoundLevel

函　数：　int __stdcall GetSoundLevel(HANDLE hChannelHandle)
参　数：　HANDLE hChannelHandle；通道句柄
返回值：　当前通道的音频输入幅度
说　明：　获取当前通道的现场声音幅度
注　意：　当无声音输入时因背景噪声的原因返回值并不为 0。

6.3.74 启动编码通道 PCI 音频预览 SetDirectAudioPreview

函 数：int __stdcall　SetDirectAudioPreview(HANDLE hChannelHandle,BOOL bEnable);
参 数：
　　hChannelHandle ；通道句柄
　　bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　6.0 版本新增函数。使用该接口时。所有系列的板卡和设备，都可以不连线到主板，直接预览输出。对于 DS-40xx 系列板卡，如果调用 SetAudioPreview，需要将 4 针线和主板上的声卡音频输入口连接，

才能有预览输出。

码流数字水印校验

DS-52xx 系列设备、DS-43xx 系列板卡、DS-42xx 系列板卡不支持校验功能

6.3.75 设置主通道数字水印校验 SetChannelStreamCRC

函　数：　int __stdcall SetChannelStreamCRC(HANDLE hChannel,BOOL bEnable)
参　数：
　　HANDLE hChannel；通道句柄
　　BOOL bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　此函数不支持动态设置，设置后会在下一次启动录像后生效。

6.3.76 设置子通道数字水印校验 SetSubChannelStreamCRC

函　数：　int __stdcall SetSubChannelStreamCRC(HANDLE hChannel,BOOL bEnable)
参　数：
　　HANDLE hChannel；通道句柄
　　BOOL bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　此函数不支持动态设置，设置后会在下一次启动录像后生效。

去噪

6.3.77 设置编码通道去噪 SetDeNoise

函　数：　int __stdcall SetDeNoise(HANDLE hChannelHandle,UINT level)
参　数：
　　HANDLE　hChannelHandle；通道句柄
　　UINT　level；　去噪等级取值，取值范围为 0-3。
返回值：　成功返回 0；失败返回错误号
说　明：　6.1 新增接口，适用于 DS-43xxHW-E、DS-43xxHFV-E 系列板卡和 DS-5316HF 设备。设置编码通道去噪，去噪等级：3 最高，0 最低，默认为 0。

6.4解码卡 API

解码卡初始化及释放

6.4.1 初始化解码卡 HW_InitDecDevice

函　数：　int __stdcall HW_InitDecDevice(long *pDeviceTotal)
参　数：　long *pDeviceTotal；输出参数，输出初始化成功的解码通道个数
返回值：　成功返回 0；失败返回错误号
说　明：　初始化解码卡，输出解码通道个数

6.4.2 释放解码卡 HW_ReleaseDecDevice

函　数：　int __stdcall HW_ReleaseDecDevice()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　释放解码卡，应在程序退出时调用

6.4.3 初始化 DirectDraw HW_InitDirectDraw

函　数：　int __stdcall HW_InitDirectDraw(HWND hParent,COLORREF colorKey)
参　数：
HWND hParent；窗口句柄，显示区域必须在此窗口内。显示区域的坐标使用此窗口的客户区坐标
COLORREF colorKey；Overlay 关键色
返回值：　成功返回 0；失败返回错误号
说　明：　初始化 DirectDraw。

6.4.4 释放 DirectDraw HW_ReleaseDirectDraw

函　数：　int __stdcall HW_ReleaseDirectDraw()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　释放 DirectDraw

打开及关闭解码通道

6.4.5 打开解码通道 HW_ChannelOpen

函　数：　int __stdcall HW_ChannelOpen(long nChannelNum,HANDLE* phChannel)
参　数：
　　long nChannelNum；通道号，自 0 开始
　　HANDLE* phChannel；输出参数，输出解码卡的操作句柄，与通道相关的操作需使用相对应的通道操作句柄
返回值：　成功返回 0；失败返回错误号
说　明：　打开解码通道，获取通过操作句柄

6.4.6 关闭解码通道 HW_ChannelClose

函　数：　int __stdcall HW_ChannelClose(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　关闭通道，释放相关资源

解码卡信息获取

6.4.7 版本信息获取 HW_GetVersion

函　数：　int __stdcall HW_GetVersion(PHW_VERSION pVersion)
参　数：　PHW_VERSION pVersion；版本信息
返回值：　成功返回 0；失败返回错误号
说　明：　获取版本信息

版本信息结构体

typedef struct{
　　ULONG DspVersion, DspBuildNum；DSP 程序的版本号和 Build 号
　　ULONG DriverVersion, DriverBuildNum；驱动程序的版本号和 Build 号
　　ULONG SDKVersion, SDKBuildNum；SDK 的版本号和 Build 号
}HW_VERSION, *PHW_VERSION

解码卡音视频信号输出设置

音频预览设置

6.4.8 音频预览设置 HW_SetAudioPreview

函　数：　int __stdcall HW_SetAudioPreview(HANDLE hChannel, BOOL bEnable)
参　数：
　　HANDLE hChannel；通道句柄
　　BOOL bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　通过 PC 声卡对音频进行预览，接线方式和编码卡相同。同一时间只能打开一个通道的音频预览，且会自动关闭原先已经打开的通道声音
注　意：　启动音频预览功能前必须要打开通道 HW_ChannelOpen 及打开通道声音 HW_PlaySound，先前 SDK 版本默认每个 DSP 输出前 2 路音频，无需调用 SetDecoderAudioOutput 即可输出此 2 路音频预览，4.3 版本起调整为解码卡启动后默认的音频输出为关闭状态，必须先调用 SetDecoderAudioOutput 设置音频输出后才能调用 HW_SetAudioPreview 预览相应的音频通道

6.4.9 启动解码通道 PCI 音频预览 HW_SetDirectAudioPreview

函　数：int __stdcall HW_SetDirectAudioPreview(HANDLE hChannel,BOOL bEnable);
参　数：
　　hChannel；通道句柄
　　bEnable；使能
返回值：　成功返回 0；失败返回错误号
说　明：　6.0 版本新增函数

视频 VGA 预览设置

6.4.10 设置视频显示参数 HW_SetDisplayPara

函　数：　int __stdcall HW_SetDisplayPara(HANDLE hChannel,DISPLAY_PARA *pPara)
参　数：
　　HANDLE hChannel；通道句柄
　　DISPLAY_PARA *pPara；视频显示参数
返回值：　成功返回 0；失败返回错误号
说　明：　设置视频显示参数
视频显示参数结构体
typedef struct{

long bToScreen；是否输出到 PC 屏幕，1 是 0 否
long bToVideoOut；是否输出到监视器，1 是 0 否(目前无效)
long nLeft；输出到屏幕上的范围，相对 hParent 客户区坐标
long nTop；
long nWidth；
long nHeight；
long nReserved；保留参数
}DISPLAY_PARA,*PDISPLAY_PARA

6.4.11 刷新 DirectDraw 表面 HW_RefreshSurface

函　数：　int __stdcall HW_RefreshSurface()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　刷新显示区域，当窗口(hParent)的位置发生改变后，需要刷新 DirectDraw 表面，适应新的区域。

6.4.12 重载 DirectDraw 表面 HW_RestoreSurface

函　数：　int __stdcall HW_RestoreSurface()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　表面在使用过程中可能被其他的程序占用，这时需要调用这个接口，以便重新获得表面。使用 MFC 开发时最方便的做法是在 OnPaint 中调用这个接口。

6.4.13 清除 DirectDraw 表面中的数据 HW_ClearSurface

函　数：　int __stdcall HW_ClearSurface()
参　数：　无
返回值：　成功返回 0；失败返回错误号
说　明：　清除表面中的数据，在窗口切换时，表面会保留上一帧图像内容。如果不想显示这个内容，可以调用此接口。

6.4.14 缩放 DirectDraw 表面的显示区域 HW_ZoomOverlay

函　数：　int __stdcall HW_ZoomOverlay(RECT* pSrcClientRect, RECT* pDecScreenRect)
参　数：
RECT* pSrcClientRect；源数据，窗口(hParent)客户区坐标
RECT* pDecScreenRect；目标区域，屏幕坐标
返回值：　成功返回 0；失败返回错误号
说　明：　放大或缩小显示表面上的某块区域。这个接口也是实现全屏显示的方法之一。目前有两种方法可实现全屏显示：一是使用显卡放大功能调用这个接口，将要放大的某块区域

pSrcClientRec(DISPLAY_PARA 中指定的范围，客户区坐标)，放大到全屏 pDecScreenRect(屏幕坐标)，注意，pSrcClientRect 作为源应该不小于原始图像大小(PAL 制下 CIF 格式图像是 352*288)，否则效果很差，在附带的 demo 程序中可以比较；二是使用卡上的放大功能，在设置 DISPLAY_PARA 中的现实范围时，会自动调用卡上的缩放功能，

注　意：　如果要使用这种方法，初始化 DirectDraw 时指定的 hParent 窗口客户区必须覆盖整个屏幕，因为 DISPLAY_PARA 中指定的显示区域不能超出 hParent 窗口的客户区。

6.4.15 预览去闪烁功能 HW_SetDecoderPostProcess

函　数：　int __stdcall HW_SetDecoderPostProcess(HANDLE hChannel,UINT param)

参　数：

HANDLE hChannel；通道句柄

UINT param；bit0 设置为 1 则执行去闪烁功能，设置为 0 则不执行

返回值：　成功返回 0；失败返回错误号

说　明：　静止图像区域有噪声情况下,图像会经常闪烁(或称刷新)，启动去闪烁功能后，闪烁效果可消除或减轻

6.4.16 解码视频回放局部放大 ZoomOutDecoderVideoPreview

函　数：int __stdcall　ZoomOutDecoderVideoPreview(HANDLE hChannel, UINT nRectIdx, BOOL bEnable, HWND hWnd, ZOOM_RECT_NORMALIZE* pZoomRect, RECT* pRect);

参　数：

hChannel；通道句柄

nRectIdx；放大区域的序号,0 表示放大后在当前解码通道视频回放的窗口显示

bEnable；开始或者关闭局部放大功能

hWnd；显示放大区域的窗口句柄，nRectIdx 为 0 时无效

pZoomRect；局部放大区域相对于解码通道视频回放区域的坐标，取值归一化，见

ZOOM_RECT_NORMALIZE 结构

pRect；显示局部放大视频回放的目标区域，一般为 hWnd 的客户区坐标。nRectIdx 为 0 时无效

局部放大区域归一化结构体，区域归一化,浮点数值为原始视频宽高的百分比大小,精度为 0.001

```
typedef struct _ZOOM_RECT_NORMALIZE_
{
    float left；放大区域左上角点的 X 轴坐标
    float top；放大区域左上角点的 Y 轴坐标
    float width；放大区域的宽度
    float height；放大区域的高度
}ZOOM_RECT_NORMALIZE;
```

返回值：　成功返回 0；失败返回错误号

说　明：　6.0 版本新增函数

注　意：1 路解码通道最多支持 16 个局部放大区域。nRectIdx 的取值范围是[0，16]，取 0 时表示放大后在当前解码通道视频预览的窗口显示，此时 hWnd 参数和 pRect 参数无效；取 1 到 16 时分别表示设置对应的局部放大区域的参数。

6.4.17 注册解码视频回放画图回调函数 HW_RegisterDrawFun

函　数：int __stdcall　HW_RegisterDrawFun(DWORD nport, DRAWFUN(DrawFun),LONG nUser);

参　数：

nport；通道号。高 16 位为局部放大区域的序号，从 1 开始计，为 0 表示原始的解码视频回放图像；低 16 位为解码通道号。例如，nport 为 0x00010000，表示第 0 路解码通道第 1 个区域进行放大； nport 为 0x00000000，表示第 0 路解码通道的视频回放图像。

DrawFun；用户回调函数

nUser；保留参数

返回值：　成功返回 0；失败返回错误号

说　明：　在 Offscreen 预览模式下可通过此函数在预览视频画面上叠加字符等信息。6.0 版本新增注册解码通道的某个局部放大区域功能

画图回调函数

#define DRAWFUN(x)　　void　(CALLBACK* x)(long nPort,HDC hDc,LONG nUser)

LONG nPort；通道号

HDC hDc；表面设置上下文，相当于显示窗口的 DC

LONG nUser；用户数据

注　意：调用该接口注册解码通道的某个局部放大区域的画图回调之前，需要先调用 ZoomOutDecoderVideoPreview 接口开启该局部放大区域的局部放大功能。

6.4.18 停止注册解码视频回放画图回调函数 HW_StopRegisterDrawFun

函　数：int __stdcall HW_StopRegisterDrawFun(DWORD nport);

参　数：

nport；通道号。同 HW_RegisterDrawFun 的 nport。

返回值：　成功返回 0；失败返回错误号

说　明：　停止注册回调。

6.4.19 设置解码通道局部放大的预览垂直同步 SetDecoderZoomOutAntiTearing

函　数：int __stdcall　SetDecoderZoomOutAntiTearing(HANDLE hChannel, UINT nRectIdx, BOOL bAntiTearing, DWORD reserved);

参　数：

hChannel；通道句柄

nRectIdx；放大区域的序号，必须大于等于 1

bAntiTearing；是否开启垂直同步

reserved；保留参数

返回值： 成功返回 0；失败返回错误号

说 明： 6.0 版本新增函数

注 意：调用该接口开启解码通道的某个局部放大区域的预览垂直同步之前，需要先调用 ZoomOutDecoderVideoPreview 接口开启该局部放大区域的局部放大功能。且调用该接口开启局部放大的预览垂直同步后，若调用 ZoomOutDecoderVideoPreview 接口重新开启该局部放大区域的局部放大功能后，该局部放大区域的预览垂直同步功能失效，如果需要打开该局部放大区域的预览垂直同步，要再次调用该接口开启。

视频输出设置

6.4.20 设置视频显示通道的视频制式 SetDisplayStandard

函 数： int __stdcall SetDisplayStandard(UINT nDisplayChannel,VideoStandard_t VideoStandard)

参 数：

UINT nDisplayChannel；显示通道索引

VideoStandard_t VideoStandard；视频制式

返回值： 成功返回 0；失败返回错误号

说 明： 设置视频显示通道的视频制式。DS-43xx 系列板卡的视频输出模式设置通过 SetDisplayVideoMode 实现。

6.4.21 设置视频输出模式 SetDisplayVideoMode

函 数： int _stdcall SetDisplayVideoMode(UINT nDisplayChannel, DISPLAY_ FORMAT displayFormat, DISPLAY_RESOLUTION displayResolution)

参 数：

nDisplayChannel；输出通道

DISPLAY_ FORMAT displayFormat；输出接口类型

DISPLAY_RESOLUTION displayResolution；输出分辨率和帧率

返回值：成功返回 0，失败返回错误号。

说 明：6.0 版本新增函数，可设置输出接口、输出分辨率和刷新频率

注 意：此函数适用于 DS-4101HD 和 DS-43xx 系列编解码卡。

输出模式

```
typedef enum
{
    DISPLAY_FORMAT_INVALID      = 0x00000000,
    DISPLAY_FORMAT_CVBS         = 0x00000001,
    DISPLAY_FORMAT_DVI          = 0x00000002,
    DISPLAY_FORMAT_VGA          = 0x00000004,
    DISPLAY_FORMAT_HDMI         = 0x00000008,
    DISPLAY_FORMAT_YPbPr        = 0x00000010
}DISPLAY_FORMAT;
```

输出分辨率和频率

```
typedef enum
{
    DISPLAY_RESOLUTION_INVALID              = 0x00000000,
    DISPLAY_RESOLUTION_D1                   = 0x00000001,
    DISPLAY_RESOLUTION_XGA_60HZ             = 0x00000002,
    DISPLAY_RESOLUTION_SXGA_60HZ            = 0x00000004,
    DISPLAY_RESOLUTION_SXGA_960_60HZ        = 0x00000008,
    DISPLAY_RESOLUTION_720P_50HZ            = 0x00000010,
    DISPLAY_RESOLUTION_720P_60HZ            = 0x00000020,
    DISPLAY_RESOLUTION_1080I_50HZ           = 0x00000040,
    DISPLAY_RESOLUTION_1080I_60HZ           = 0x00000080,
    DISPLAY_RESOLUTION_1080P_24HZ           = 0x00000100,
    DISPLAY_RESOLUTION_1080P_25HZ           = 0x00000200,
    DISPLAY_RESOLUTION_1080P_30HZ           = 0x00000400,
}DISPLAY_RESOLUTION;
```

6.4.22 获取视频输出模式 GetDisplayVideoMode

函　数：int _stdcall　GetDisplayVideoMode(UINT nDisplayChannel,DWORD *dwDisplayFormat, DWORD *dwDisplayResolution)

参　数：

nDisplayChannel；输出通道

dwDisplayFormat；输出接口

dwDisplayResolution；输出分辨率和刷新频率

返回值：正确为 0，其他为错误号

说　明：6.0 版本新增函数，可获取输出接口、输出分辨率和刷新频率

注　意：此函数适用于 DS-4101HD 和 DS-43xx 系列编解码卡。

6.4.23 获取输出模式所支持的显示分辨率 GetDisplayFormatCapability

函　数： int __stdcall GetDisplayFormatCapability(UINT nDisplayChannel,DISPLAY_FORMAT displayFormat, DWORD *dwResolutionCapabiltiy);

输入参数：

nDisplayChannel；输出通道

DISPLAY_FORMAT displayFormat；输出模式

输出参数：

DWORD *dwResolutionCapabiltiy；返回该模式所支持的分辨率，按位表示，

具体定义见 DISPLAY_RESOLUTION。

说　明：6.0 版本新增函数，

注　意：此函数适用于 DS-4101HD 和 DS-43xx 系列编解码卡。

DISPLAY_FORMAT_CVBS 支持分辨率：

DISPLAY_RESOLUTION_D1

DISPLAY_FORMAT_VGA 支持分辨率：

DISPLAY_RESOLUTION_XGA_60HZ

DISPLAY_RESOLUTION_SXGA_60HZ

DISPLAY_RESOLUTION_SXGA_960_60HZ

DISPLAY_RESOLUTION_720P_60HZ

DISPLAY_FORMAT_DVI，DISPLAY_FORMAT_HDMI 支持分辨率：

DISPLAY_RESOLUTION_XGA_60HZ

DISPLAY_RESOLUTION_SXGA_60HZ

DISPLAY_RESOLUTION_SXGA_960_60HZ

DISPLAY_RESOLUTION_720P_50HZ

DISPLAY_RESOLUTION_720P_60HZ

DISPLAY_RESOLUTION_1080I_50HZ

DISPLAY_RESOLUTION_1080I_60HZ

DISPLAY_RESOLUTION_1080P_24HZ

DISPLAY_RESOLUTION_1080P_25HZ

DISPLAY_RESOLUTION_1080P_30HZ

DISPLAY_FORMAT_YPbPr 支持分辨率：

DISPLAY_RESOLUTION_720P_60HZ

DISPLAY_RESOLUTION_1080I_60HZ

视频输出显示区域设置

6.4.24 设置显示区域的形式及参数(视频输出的画面分割情况)SetDisplayRegion

函　数：　int __stdcall SetDisplayRegion(UINT nDisplayChannel,UINT nRegionCount, REGION_PARAM *pParam,UINT nReserved)

参　数：

UINT nDisplayChannel；显示通道索引

UINT nRegionCount；画面分割区域个数，每个显示通道最多划分为 16 个显示区域

REGION_PARAM *pParam；区域参数

UINT nReserved；保留参数

返回值：

ERR_NOT_SUPPORT：DSP 资源不足，无法划分窗口

ERR_NOT_SUPPORT：MD/MD+卡每一个显示通道最大支持 1 个 4cif＋2 个 qcif 窗口，如果该卡的某个显示通道的总面积超过该限制，则返回此错误

ERR_INVALID_DEVICE：nDisplayChannel 溢出

ERR_INVALID_ARGUMENT：nRegionCount 溢出，参数对齐错误，区域超出范围

ERR_KERNEL：其它情况

说　明：　将某个显示通道的输出划分为多个区域，实现矩阵输出功能。

区域参数结构体

typedef struct{

UINT left；区域左边界，16 对齐

UINT top；区域上边界，8 对齐

UINT width；区域宽度，16 对齐

UINT height；区域高度，8 对齐

COLORREF color；区域背景色

UINT param；区域扩展参数

}REGION_PARAM

6.4.25 改变某个显示区域的位置 SetDisplayRegionPosition

函　数：　int __stdcall SetDisplayRegionPosition(UINT nDisplayChannel,UINT nRegion,UINT nLeft, UINT nTop)

参　数：

UINT nDisplayChannel；显示通道索引

UINT nRegion；需要调整位置的区域

UINT nLeft,调整后的左边界

UINT nTop；调整后的上边界

返回值：

ERR_INVALID_DEVICE：nDisplayChannel 溢出，nRegion 超过该显示通道已划分的区域个数。

ERR_KERNEL：其它情况

说　明：　调整某个显示区域的位置。

6.4.26 用自定义的图像填充显示区域 FillDisplayRegion

函　数：　int __stdcall FillDisplayRegion(UINT nDisplayChannel,UINT nRegion,unsigned char *pImage)

参　数：

UINT nDisplayChannel；显示通道索引

UINT nRegion；需要填充的区域

unsigned char *pImage；YUV420 格式图像指针

返回值：

ERR_INVALID_DEVICE：nDisplayChannel 溢出，nRegion 超过该显示通道已划分的区域个数。

ERR_KERNEL：其它情况

说　明：　用自定义的 yuv420 格式图像填充某个显示区域。pImage 所指向图象的大小必须和 SetDisplayRegion 中设置的图像大小相同，否则图像无法正常显示。

注　意：　使用此函数前应当停止该区域解码，如果该区域当前有图像正在显示，则该命令无效。

6.4.27 填充显示区域(扩展) FillDisplayRegionEx

函　数：int __stdcall FillDisplayRegionEx(UINT nDisplayChannel,UINT nRegion,unsigned char *pImageBuf , UINT nWidth, UINT nHeight);

参　数：

nDisplayChannel；显示通道

nRegion；显示区域

pImageBuf ；进行填充的图像数据

nWidth；进行填充的图像宽度

nHeight；进行填充的图像高度

返回值：　成功反馈 0，失败返回错误好

说　明：　6.0 版本新增函数，　与 FillDisplayRegion 区别在于，FillDisplayRegion 需要填充和显示区域一样大的图片，且显示区域宽度不能大于 704 个像素，高度不能大于 576 个像素；FillDisplayRegionEx 能够支持缩放，如果图片比较大，内部会自动调整为显示区域大小(如果显示区域宽度大于 704 个像素，则图像宽度缩放成 704 个像素；如果显示区域高度大于 576（PAL 制）或者 480（NTSC 制），则图像高度缩放成 576（PAL 制）或者 480（NTSC 制）。

6.4.28 清空显示区域 ClearDisplayRegion

函　数：　int __stdcall ClearDisplayRegion(UINT nDisplayChannel,UINT nRegionFlag)

参　数：

UINT nDisplayChannel；显示通道索引

UINT nRegionFlag；Bit0－Bit15，对应区域 1－16，对应位置 1，则将该区域清空。

返回值：　ERR_INVALID_DEVICE：nDisplayChannel 溢出。ERR_KERNEL：其他情况

说　明：　清空显示区域，显示 SetDisplayRegion 中所设置的背景色

注　意：　使用此函数前应当停止该区域解码，如果该区域当前有图像正在显示，则该命令无效。

视频输出亮度调节

6.4.29 设置视频输出亮度 SetDisplayVideoBrightness

函　数：　int __stdcall SetDisplayVideoBrightness(UINT chan,int Brightness)

参　数：

UINT chan；显示通道索引

int Brightness；亮度值，取值范围 0-255

返回值：　成功返回 0；失败返回错误号

说　明：　设置板卡视频输出口的亮度值

解码卡解码及播放

当对解码卡的视频、音频设置完成后，需启动相应的播放功能函数才能进行图像的显示或者音频的播放

解码卡解码数据流

6.4.30 设置流播放模式及参数 HW_SetStreamOpenMode

函　数：　int __stdcall HW_SetStreamOpenMode(HANDLE hChannel,ULONG nMode)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG nMode；流调节参数，取值范围 0-5。0 表示不作任何调节，适用于文件播放模式；1-5 调节实时流的流程性和延迟性，取值越大流畅性越好但延迟越大。
返回值：　成功返回 0；失败返回错误号
说　明：　设置流播放模式下的流畅性和延时性.

6.4.31 获取流播放模式及参数 HW_GetStreamOpenMode

函　数：　int __stdcall HW_GetStreamOpenMode(HANDLE hChannel,ULONG *pMode)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG *pMode；获取流调节参数 0-5
返回值：　成功返回 0；失败返回错误号
说　明：　获取当前流的调节参数

6.4.32 打开数据流 HW_OpenStream

函　数：　int __stdcall HW_OpenStream(HANDLE hChannel,PBYTE pFileHeadBuf,DWORD nSize)
参　数：
　　HANDLE hChannel；通道句柄
　　BYTE* pFileHead；文件头数据
　　DWORD nHeadSize；文件头长度
返回值：　成功返回 0；失败返回错误号
说　明：　打开数据流，类似于打开录像文件。

6.4.33 关闭数据流 HW_CloseStream

函　数：　int __stdcall HW_CloseStream(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄

返回值： 成功返回 0；失败返回错误号
说 明： 关闭数据流

6.4.34 输入数据流 HW_InputData

函 数： int __stdcall HW_InputData(HANDLE hChannel,PBYTE pBuf,DWORD nSize)
参 数：
HANDLE hChannel,；通道句柄
PBYTE pBuf,；数据缓冲区
DWORD nSize；输入数据的大小，取值需按 4 字节对齐
返回值： 实际成功输入的数据大小，异常返回 0 或错误号
说 明： 输入数据流，需要在打开数据流后进行数据输入。

6.4.35 流模式下重启解码器 HW_ResetStream

函 数： int __stdcall HW_ResetStream(HANDLE hChannel)
参 数： HANDLE hChannel；通道句柄
返回值： 成功返回 0；失败返回错误号
说 明： 在流模式下重启解码器

解码卡解码数据流功能扩展(以视、音频分开的形式)

6.4.36 打开数据流 HW_OpenStreamEx

函 数： int __stdcall HW_OpenStreamEx(HANDLE hChannel,PBYTE pFileHeadBuf,DWORD nSize)
参 数：
HANDLE hChannel；通道句柄
PBYTE pFileHeadBuf；文件头数据
DWORD nSize；文件头长度
返回值： 成功返回 0；失败返回错误号
说 明： 以视、音频分开的方式打开数据流

6.4.37 关闭数据流 HW_CloseStreamEx

函 数： int __stdcall HW_CloseStreamEx(HANDLE hChannel)
参 数： HANDLE hChannel；通道句柄
返回值： 成功返回 0；失败返回错误号
说 明： 关闭以视、音频分开的方式打开的数据流

6.4.38 输入视频数据流 HW_InputVideoData

函　数：　int __stdcall HW_InputVideoData(HANDLE hChannel,PBYTE pBuf,DWORD nSize)
参　数：
　　HANDLE hChannel；通道句柄
　　PBYTE pBuf,；数据缓冲区
　　DWORD nSize；输入数据的大小，取值需按 4 字节对齐
返回值：　实际成功输入的数据大小，异常返回 0 或错误号
说　明：　输入视频数据流，需要在打开数据流后进行数据输入。

6.4.39 输入音频数据流 HW_InputAudioData

函　数：　int __stdcall HW_InputAudioData(HANDLE hChannel,PBYTE pBuf,DWORD nSize)
参　数：
　　HANDLE hChannel；通道句柄
　　PBYTE pBuf；数据缓冲区
　　DWORD nSize；缓冲区大小
返回值：　实际成功输入的数据大小，异常返回 0 或错误号
说　明：　输入音频数据流，需要在打开数据流后进行数据输入

解码卡解码录像文件

6.4.40 打开录像文件 HW_OpenFile

函　数：　int __stdcall HW_OpenFile(HANDLE hChannel,LPTSTR sFileName)
参　数：
　　HANDLE hChannel；通道句柄
　　LPTSTR sFileName；文件名
返回值：　成功返回 0；失败返回错误号
说　明：　打开录像文件

6.4.41 关闭录像文件 HW_CloseFile

函　数：　int __stdcall HW_CloseFile(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　关闭录像文件

6.4.42 文件结束标志 HW_SetFileEndMsg

函　数：　int __stdcall HW_SetFileEndMsg(HANDLE hChannel,HWND hWnd,UINT nMsg)
参　数：
　　HANDLE hChannel；通道句柄
　　HWND hWnd；接收消息的窗口句柄
　　UINT nMsg；自定义 windows 消息
返回值：　成功返回 0；失败返回错误号
说　明：　注册一个自定义的 windows 消息，当文件结束时将被 post 到指定窗口。

视音频播放

视频播放

6.4.43 开始视频播放 HW_Play

函　数：　int __stdcall HW_Play(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　开始播放

6.4.44 停止视频播放 HW_Stop

函　数：　int __stdcall HW_Stop(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止播放

音频播放

6.4.45 打开声音 HW_PlaySound

函　数：　int __stdcall HW_PlaySound(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　打开该通道的声音，默认情况下声音是被关闭的

6.4.46 关闭声音 HW_StopSound

函　数：　int __stdcall HW_StopSound(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止该通道的声音

6.4.47 音量调节 HW_SetVolume

函　数：　int __stdcall HW_SetVolume(HANDLE hChannel,ULONG nVolume)
参　数：
HANDLE hChannel；通道句柄
ULONG nVlome；音量，取值范围 0-0xffff
返回值：　成功返回 0；失败返回错误号
说　明：　音量大小调节

6.4.48 暂停播放 HW_Pause

函　数：　int __stdcall HW_Pause(HANDLE hChannel,ULONG bPause)
参　数：
HANDLE hChannel；通道句柄
ULONG bPause；1 暂停播放，0 继续播放
返回值：　成功返回 0；失败返回错误号
说　明：　暂停播放

解码播放速度设置及获取

6.4.49 设置播放速度 HW_SetSpeed

函　数：　int __stdcall HW_SetSpeed(HANDLE hChannel,long nSpeed)
参　数：
HANDLE hChannel；通道句柄
long nSpeed；播放速度，取值范围-4~4
返回值：　成功返回 0；失败返回错误号
说　明：　设置播放速度。-4 时停止播放，调用 HW_Pause(hChannel,0)可单帧播放，每调用一次播放一帧。-3 表示 1/8 速，-2 表示 1/4 速，-1 表示 1/2 速，0 表示正常播放，1 表示 2 倍速，2 表示 4 倍速，3 表示 8 倍速，4 表示最大倍速。

6.4.50 获取播放速度 HW_GetSpeed

函　数：　int __stdcall HW_GetSpeed(HANDLE hChannel,long *pSpeed)
参　数：
　　HANDLE hChannel；通道句柄
　　long *pSpeed；输出参数，播放速度
返回值：　成功返回 0；失败返回错误号
说　明：　获取播放速度

解码播放位置设置及获取

6.4.51 设置播放位置 HW_SetPlayPos

函　数：　int __stdcall HW_SetPlayPos(HANDLE hChannel,ULONG nPos)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG nPos；播放位置的百分数，取值范围 0-100
返回值：　成功返回 0；失败返回错误号
说　明：　设置录像文件播放位置

6.4.52 获取播放位置 HW_GetPlayPos

函　数：　int __stdcall HW_GetPlayPos(HANDLE hChannel,ULONG* pPos)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG* pPos；输出参数，播放位置的百分数
返回值：　成功返回 0；失败返回错误号
说　明：　获取录像文件的播放位置

设置解码播放跳跃

6.4.53 设置播放跳跃时间间隔 HW_SetJumpInterval

函　数：　int __stdcall HW_SetJumpInterval(HANDLE hChannel,ULONG nSecond)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG nSecond；跳跃时间间隔，单位：秒
返回值：　成功返回 0；失败返回错误号

说　明：　设置跳跃时间的时间间隔

6.4.54 设置播放跳跃方向 HW_Jump

函　数：　int __stdcall HW_Jump(HANDLE hChannel,ULONG nDirection)
参　数：
　　HANDLE hChannel；通道句柄
ULONG nDirection；录像文件播放跳跃方向，JUMP_FORWARD 向前跳跃；JUMP_BACKWARD 向后跳跃；
返回值：　成功返回 0；失败返回错误号
说　明：　设置录像文件播放的跳跃方向

解码时间及帧信息获取

时间信息

6.4.55 获取文件总时间 HW_GetFileTime

函　数：　int __stdcall HW_GetFileTime(HANDLE hChannel, ULONG* pFileTime)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG* pFileTime；文件总时间，单位：毫秒
返回值：　成功返回 0；失败返回错误号
说　明：　获取文件的总时间

6.4.56 获取当前播放帧的时间(相对时间)HW_GetCurrentFrameTime

函　数：　int __stdcall HW_GetCurrentFrameTime(HANDLE hChannel,ULONG* pFrameTime)
参　数：
　　HANDLE hChannel；通道句柄
　　ULONG* pFrameTime；当前播放帧的时间，单位：毫秒
返回值：　成功返回 0；失败返回错误号
说　明：　获取当前播放帧的时间

6.4.57 获取文件的起止的绝对时间 HW_GetFileAbsoluteTime

函　数：　int __stdcall HW_GetFileAbsoluteTime(HANDLE hChannel,

SYSTEMTIME *pStartTime,SYSTEMTIME *pEndTime)

参　数：

HANDLE hChannel；通道句柄

SYSTEMTIME *pStartTime；文件开始绝对时间(毫秒参数无效，始终为 0)

SYSTEMTIME *pEndTime；文件结束绝对时间(毫秒参数无效，始终为 0)

返回值：　成功返回 0；失败返回错误号

说　明：　获取文件的绝对起止时间

6.4.58 获取文件当前播放的绝对时间

HW_GetCurrentAbsoluteTime

函　数：　int __stdcall HW_GetCurrentAbsoluteTime(HANDLE hChannel,SYSTEMTIME *pTime)

参　数：

HANDLE hChannel；通道句柄

SYSTEMTIME *pTime；文件当前播放的绝对时间(毫秒参数无效，始终为 0)

返回值：　成功返回 0；失败返回错误号

说　明：　获取文件当前播放的绝对时间

6.4.59 按照绝对时间定位文件播放位置

HW_LocateByAbsoluteTime

函　数：　int __stdcall HW_LocateByAbsoluteTime(HANDLE hChannel,SYSTEMTIME time)

参　数：

HANDLE hChannel 通道句柄

SYSTEMTIME time；定位时间(毫秒参数无效)

返回值：　成功返回 0；失败返回错误号

说　明：　按照绝对时间定位文件播放位置。只在回放录像文件、打开索引后有效

帧信息

6.4.60 获取文件总帧数 HW_GetFileTotalFrames

函　数：　int __stdcall HW_GetFileTotalFrames(HANDLE hChannel,ULONG* pTotalFrames)

参　数：

HANDLE hChannel；通道句柄

ULONG* pTotalFrames；总帧数

返回值：　成功返回 0；失败返回错误号

说　明：　获取文件总帧数

6.4.61 获取已解码的视频帧数 HW_GetPlayedFrames

函　数：　int __stdcall HW_GetPlayedFrames(HANDLE hChannel,ULONG *pDecVFrames)
参　数：
HANDLE hChannel；通道句柄
ULONG *pDecVFrames；已解码的帧数
返回值：　成功返回 0；失败返回错误号
说　明：　获取已经解码的视频帧数

6.4.62 获取当前播放帧率 HW_GetCurrentFrameRate

函　数：　int __stdcall HW_GetCurrentFrameRate(HANDLE hChannel,ULONG* pFrameRate)
参　数：
HANDLE hChannel；通道句柄
ULONG* pFrameRate；帧率
返回值：　成功返回 0；失败返回错误号
说　明：　获取当前播放的帧率

6.4.63 获取当前播放帧序号 HW_GetCurrentFrameNum

函　数：　int __stdcall HW_GetCurrentFrameNum(HANDLE hChannel,ULONG* pFrameNum)
参　数：
HANDLE hChannel；通道句柄
ULONG* pFrameNum；当前播放的帧序号
返回值：　成功返回 0；失败返回错误号
说　明：　获取当前播放的帧序号

6.4.64 按照帧号定位文件播放位置 HW_LocateByFrameNumber

函　数：　int __stdcall HW_LocateByFrameNumber(HANDLE hChannel,UINT frmNum)
参　数：　HANDLE hChannel；通道句柄
UINT frmNum；定位帧号
返回值：　成功返回 0；失败返回错误号
说　明：　按照帧号定位文件播放那个位置。只在回放录像文件、打开索引后有效

数据捕获

抓图

6.4.65 抓取解码卡解码后 YV12 格式图像 HW_GetYV12Image

函　数：　int __stdcall HW_GetYV12Image(HANDLE hChannel, PBYTE pBuffer, ULONG nSize)
参　数：
　　HANDLE hChannel；解码通道句柄
　　PBYTE pBuffer；保存图像的缓冲区
　　ULONG nSize；缓冲区的尺寸，大小应大于或等于 yv12 格式图像的尺寸
返回值：　成功返回 0；失败返回错误号
说　明：　抓取解码卡当前解码的 yv12 格式图像。yv12 格式图像的存储大小为：(*pWidth)*(*pHeight)*3/2，YV12 格式每个像素占用 3/2 个 BYTE。DS-4101HD 暂不支持。

6.4.66 图像格式转换(YV12 转为 BMP)HW_ConvertToBmpFile

函　数：　int __stdcall HW_ConvertToBmpFile(BYTE * pBuf,ULONG nSize,ULONG nWidth, ULONG nHeight,char *sFileName,ULONG nReserved)
参　数：
　　BYTE * pBuf；YV12 格式图像缓冲区
　　ULONG nSize；缓冲区大小
　　ULONG nWidth；图像宽度
　　ULONG nHeight；图像高度
　　char *sFileName；保存为 BMP 格式的文件名
　　ULONG nReserved；保留参数
返回值：　成功返回 0；失败返回错误号
说　明：　将 YV12 格式的图像转化为 BMP 格式

录像

6.4.67 启动码流捕获 HW_StartCapFile

函　数：　int __stdcall HW_StartCapFile(HANDLE hChannel,LPTSTR sFileName)
参　数：
　　HANDLE hChannel；通道号
　　LPTSTR sFileName；录像文件名
返回值：　成功返回 0；失败返回错误号
说　明：　启动码流捕获，存成录像文件

6.4.68 停止码流捕获 HW_StopCapFile

函　数：　int __stdcall HW_StopCapFile(HANDLE hChannel)
参　数：　HANDLE hChannel；通道句柄
返回值：　成功返回 0；失败返回错误号
说　明：　停止码流捕获

6.4.69 获取码流中图像尺寸 HW_GetPictureSize

函　数：　int __stdcall HW_GetPictureSize(HANDLE hChannel,ULONG* pWidth, ULONG* pHeight)
参　数：
HANDLE hChannel；通道句柄
ULONG* pWidth；输出参数，图像宽
ULONG* pHeight；输出参数，图像高
返回值：　成功返回 0；失败返回错误号
说　明：　获取当前码流中的原始图像尺寸

解码后原始数据流捕获(YUV420 格式)

解码卡解码通道原始图像数据回调

6.4.70 注册解码通道数据流捕获回调函数 RegisterDecoderVideoCaptureCallback

函　数：　int__stdcall RegisterDecoderVideoCaptureCallback (DECODER_VIDEO_CAPTURE_CALLBACK DecoderVideoCaptureCallback,void *context)
参　数：
DECODER_VIDEO_CAPTURE_CALLBACK DecoderVideoCaptureCallback；解码回调函数
void *context；设备上下文
返回值：　成功返回 0；失败返回错误号
说　明：　注册解码通道原始数据流捕获回调函数

解码回调函数

typedef void (*DECODER_VIDEO_CAPTURE_CALLBACK)(UINT nChannelNumber,void *DataBuf, UINT width,UINT height,UINT nFrameNum,UINT nFrameTime,SYSTEMTIME *pFrameAbsoluteTime,void *context)

UINT nChannelNumber；解码通道句柄
void *DataBuf；缓冲区地址
UINT width；图像宽度
UINT height；图像高度

UINT nFrameNum；捕获的当前帧的序号

UINT nFrameTime；捕获的当前帧的相对时间，单位：毫秒

SYSTEMTIME *pFrameAbsoluteTime；捕获的当前帧的绝对时间

void *context；设备上下文

6.4.71 设置解码通道数据流捕获函数

HW_SetDecoderVideoCapture

函　数：　int __stdcall HW_SetDecoderVideoCapture(HANDLE hChannel,BOOL bStart,UINT param)

参　数：

HANDLE hChannel；解码通道句柄

BOOL bStart；使能

UINT param；保留参数，取值为 0

返回值：　成功返回 0；失败返回错误号

说　明：　捕获解码卡解码后解码通道输出的 yuv420 数据。DS-4101HD 不支持解码通道的原始视频数据获取。

解码卡显示通道原始图像数据回调

6.4.72 注册显示通道数据流捕获回调函数

RegisterDisplayVideoCaptureCallback

函　数：　int __stdcall RegisterDisplayVideoCaptureCallback
(IMAGE_STREAM_CALLBACK DisplayVideoCaptureCallback,void *context)

参　数：

IMAGE_STREAM_CALLBACK　DisplayVideoCaptureCallback；显示回调函数

void *context；设备上下文

返回值：　成功返回 0；失败返回错误号

说　明：　注册显示通道数据流捕获回调。

显示回调函数

typedef void (*IMAGE_STREAM_CALLBACK)(UINT channelNumber,void *context)

UINT nDisplayChannel；显示通道索引

void *context；设备上下文

6.4.73 设置显示通道数据流捕获函数 SetDisplayVideoCapture

函　数：　int __stdcall SetDisplayVideoCapture(UINT nDisplayChannel,BOOL bStart,UINT fps,
UINT width,UINT height,unsigned char *imageBuffer)

参　数：

UINT nDisplayChannel；显示通道索引

BOOL bStart；使能

UINT fps；帧率

UINT width；图像宽度(暂时只支持 4CIF 宽度 704，不支持缩放)

UINT height；图像高度(暂时只支持 4CIF 高度 PAL576/NTSC480，不支持缩放)

unsigned char *imageBuffer；yuv420 数据缓存

返回值： 成功返回 0；失败返回错误号

说 明： 捕获解码后显示通道的 yuv420 数据流；DS-4101HD 不支持显示通道的原始视频数据获取。

文件索引

6.4.74 创建文件索引 HW_SetFileRef

函 数： int __stdcall HW_SetFileRef(HANDLE hChannel,BOOL bEnable,
FILE_REF_DONE_CALLBACK FileRefDoneCallback)

参 数：

HANDLE hChannel；通道句柄

BOOL bEnable；使能

FILE_REF_DONE_CALLBACK FileRefDoneCallback；创建索引完成后回调函数

返回值： 成功返回 0；失败返回错误号

说 明： 设置文件索引

创建索引完成回调函数

typedef void (*FILE_REF_DONE_CALLBACK)(UINT nChannel,UINT nSize)

UINT nChannel；通道号

UINT nSize；索引大小(暂时无效，以后可以增加索引导出、导入功能)

6.4.75 文件索引导入 HW_ImportFileRef

函 数： int __stdcall HW_ImportFileRef(HANDLE hChannel,char *pBuffer,UINT nSize)

参 数：

HANDLE hChannel；通道句柄

char *pBuffer；索引数据缓冲

UINT nSize；缓冲区大小

返回值： 成功返回 0；失败返回错误号

说 明： 文件索引导入。要使用此功能，必须先在 HW_SetFileRef 中关掉创建文件索引功能，然后在 HW_OpenFile 之后，再导入索引。

6.4.76 文件索引导出 HW_ExportFileRef

函 数： int __stdcall HW_ExportFileRef(HANDLE hChannel,char *pBuffer,UINT nSize)

参 数：

HANDLE hChannel；通道句柄

char *pBuffer；索引导出数据缓冲

UINT nSize；缓冲区大小，不能小于索引数据的长度，索引长度可以在生成索引的回调函数中获得

返回值： 成功返回 0；失败返回错误号

说 明： 索引文件导出。

OSD

6.4.77 设置解码通道的 OSD 显示模式

SetDecoderOsdDisplayMode

函 数：int __stdcall SetDecoderOsdDisplayMode(HANDLE hChannel, BOOL bEnableOsd, UINT nLuminance, UINT nFlash, COLORREF CharacterColor, COLORREF BackGroundColor, UINT nTranslucentMode, BOOL bAutoAdjustLum, UINT timeMode, int *param, int nLineCount, UINT **Format);

参 数：

hChannelHandle ；通道句柄

bEnableOsd；开启或关闭 OSD

nLuminance；OSD 亮度值，取值范围[0,255]

nFlash；OSD 闪烁，Bit0-7 停止秒数，Bit8-15 显示秒数；例如，flash＝(2<<8)|1 表示 Logo 显示 2 秒，停止 1 秒

CharacterColor；OSD 字符的颜色，支持 RGB 格式；如果值为-1 则为普通模式，OSD 字符用白色

BackGroundColor；OSD 背景的颜色，支持 RGB 格式；如果值为-1 则为普通模式，背景无颜色

nTranslucentMode；是否半透明,0 前景和背景都不透明，1 前景半透明且背景不透明，2 前景不透明且背景半透明，3 前景和背景都半透明

bAutoAdjustLum；是否自动调整亮度，自动调整时，指定的亮度值无效

timeMode ；0，系统时间；1，解码时间。

Param；int 型的数组，设定每一行 OSD 的水平和垂直放大倍数，Bit0-7 垂直放大倍数 Bit8-15 水平放大倍数

nLineCount ；OSD 显示的行数，最多为 8 行

Format；描述多行字符的叠加位置和次序的格式，同 SetEncoderOsdDisplayMode 的 Format 参数

返 回 值：函数调用成功返回 0，失败则返回错误号。

说 明： 6.0 版本新增函数，支持闪烁，可以设置 OSD 字符颜色和背景颜色，字符和背景支持半透明，支持 8 行字符且每行可以独立设置放大倍数，支持自动调整亮度。

6.4.78 设置显示通道的 OSD 显示模式

SetDisplayOsdDisplayMode

函 数：int __stdcall SetDisplayOsdDisplayMode(UINT nDisplayChannel, BOOL bEnableOsd, UINT nLuminance, UINT nFlash, COLORREF CharacterColor, COLORREF BackGroundColor, UINT nTranslucentMode, BOOL bAutoAdjustLum, int *param, int nLineCount, UINT **Format);

参　数：

nDisplayChannel；显示通道序号

bEnableOsd；开启或关闭 OSD

nLuminance；OSD 亮度值

nFlash；OSD 闪烁，Bit0-7 停止秒数，Bit8-15 显示秒数；例如，flash＝(2<<8)|1 表示 Logo 显示 2 秒，停止 1 秒

CharacterColor；OSD 字符的颜色，支持 RGB 格式；如果值为-1 则为普通模式，OSD 字符用白色

BackGroundColor；OSD 背景的颜色，支持 RGB 格式；如果值为-1 则为普通模式，背景无颜色

nTranslucentMode；是否半透明,0 前景和背景都不透明，1 前景半透明且背景不透明，2 前景不透明且背景半透明，3 前景和背景都半透明

bAutoAdjustLum；是否自动调整亮度，自动调整时，指定的亮度值无效

param；int 型的数组，设定每一行 OSD 的水平和垂直放大倍数，Bit0-7 垂直放大倍数 Bit8-15 水平放大倍数

nLineCount；OSD 显示的行数，最多为 8 行

Format；描述多行字符的叠加位置和次序的格式。其中每一行的第一元素 X 和第二元素 Y 是该字串在标准 4CIF 图象的起始位置，X 必须是 16 的倍数，Y 可以在图象高度内取值即(0-575)PAL 、(0-479)NTSC，DS-52xx 系列设备、DS-42xx 系列板卡 Y 值需要按照 16 对齐，如果取值未对齐会作自动调整；可以是 ASCII 码也可以是汉字，当想要显示当前时间时，可以指定为固定的时间定义值，其值如下：

_OSD_YEAR4_EX	四位的年显示，如 2004
_OSD_YEAR2_EX	两位的年显示，如 02
_OSD_MONTH3_EX	英文的月显示，JAN~DEC
_OSD_MONTH2_EX	两位阿拉伯数字的月显示，01~12
_OSD_DAY_EX	两位的阿拉伯数字的日显示，01~31
_OSD_WEEK3_EX	英文的星期显示，MON~SUN
_OSD_CWEEK1_EX	中文的星期显示，星期一~星期日
_OSD_HOUR24_EX	24 小时的时钟显示，00~23
_OSD_HOUR12_EX	12 小时的时钟显示，00~12
_OSD_MINUTE_EX	两位分钟的显示，00~59
_OSD_SECOND_EX	两位秒的显示，00~59
_OSD_MILISECOND_EX	3 位毫秒的显示，000~999
_OSD_APM_EX	2 位上下午，AM 和 PM

在格式字符串的每一行最后一个元素必须以 NULL(0)结尾，否则会显示错误的内容。

返 回 值：函数调用成功返回 0，失败则返回错误号。

说　明：　6.0 版本新增函数，支持闪烁，可以设置 OSD 字符和背景颜色，字符和背景支持半透明，支持 8 行字符且每行可以独立设置放大倍数，支持自动调整亮度。

6.4.79 设置 OSD 显示 SetOsd

函　数：　int __stdcall SetOsd(HANDLE hChannelHandle, BOOL Enable)

参　数：

HANDLE hChannelHandle；通道句柄

BOOL Enable；OSD 是否显示

返回值： 成功返回 0；失败返回错误号

说 明： 设置 OSD 显示，将当前的系统时间年月日星期时分秒等信息叠加在视频之上，并可作透明处理。

LOGO

6.4.80 数据格式转换(bmp 转 yuv422)LoadYUVFromBmpFile

函 数： int __stdcall LoadYUVFromBmpFile(char *FileName, unsigned char *yuv, int BufLen, int *Width, int *Height)

参 数：

char *FileName；文件名

unsigned char *yuv；YUV422 图像指针

int BufLen；YUV422 图像缓存大小

int *Width；返回的 YUV422 图像的宽度

int *Height；返回的 YUV422 图像的高度

返回值： 成功返回 0；失败返回错误号

说 明： 将 24 位 bmp 格式图像转换为 yuv422 格式图像。

注 意： bmp 位图的长宽必须为 16 的倍数，图像尺寸最大支持 4CIF。

6.4.81 设置显示通道 LOGO 显示位置及数据 SetDisplayLogo

函 数：int __stdcall SetDisplayLogo(UINT nDisplayChannel, int x, int y, int w, int h, unsigned char *yuv);

参 数：

nDisplayChannel；显示通道号

x,y,w,h；LOGO 位图的坐标

YUV；LOGO 图片指针，UYVY (yuv422)格式

返回值：成功返回 0，失败返回错误号。

说 明： 6.0 版本新增函数，DS-40xxMD/MD+卡、DS-4101HD 卡、DS-41xxHCV 卡、DS-5116HF 设备显示通道支持叠加 LOGO 功能

注 意： x 和 w 要 16 对齐，y 和 h 要 8 对齐。

6.4.82 设置显示通道 LOGO 显示模式

SetDisplayLogoDisplayMode

函 数：int __stdcall SetDisplayLogoDisplayMode(UINT nDisplayChannel, COLORREF ColorKey, BOOL Translucent, int Twinkle);

参 数：

nDisplayChannel；显示通道号

ColorKey；关键色

Translucent；是否透明

Twinkle；是否闪烁，闪烁时间

返回值：成功返回 0，失败返回错误号。

说　明：　6.0 版本新增函数，DS-400xMD/MD+卡、DS-4101HD 卡、DS-41xxHCV 卡、DS-5116HF 设备显示通道 LOGO 叠加功能

6.4.83 停止显示通道 LOGO 显示 StopDisplayLogo

函　数：int __stdcall StopDisplayLogo(UINT nDisplayChannel);

参　数：

nDisplayChannel；显示通道号

返回值：成功返回 0，失败返回错误号。

说　明：　6.0 版本新增函数，DS-4000MD/MD+、DS-4100HCV 卡显示通道 LOGO 叠加功能

6.5 板卡视音频输出 API

视音频输出功能是指能将系统中编码通道或者解码通道的数据输出到系统中的显示通道即输出通道上的功能，一般用于本地或者远程模拟信号上墙的场合。

DS-40xxMD/MD+矩阵解码卡、DS-4101HD 卡、DS-4308MD-E 单独使用的时候，可以将获取的网络实时流或者文件通过解码通道解码，然后输出到任意输出输出口上，实现矩阵上墙的功能。

DS-40xx/41xx/42xx 系列编码板卡和 DS-40xxMD/MD+、DS-4101HD、DS-4308MD-E 解码卡混插的时候，编码板卡的输入数据可以通过解码卡的输出口直接进行模拟矩阵输出，实现本地视音频模拟矩阵输出功能；

DS-41xxHCV、DS-42xxHFV 型号编码板卡上，每张板卡均支持 1 路视频与 1 路音频输出功能，可以将系统中的编码通道或者解码通道数据输出到输出口上，实现本地视音频输出功能。

DS-43xxHCV-E、DS-43xxHFV-E 型号编码板卡，每张板卡均支持 1 路视频与 1 路音频输出功能，可以将系统中的编码通道或者解码通道数据输出到输出口上，实现本地视音频输出功能。

释　义：

内部输出

指(编码通道或解码通道)与(输出通道)必须是位于同一个处理芯片中

外部输出

指(编码通道或解码通道)与(输出通道)在或者不在同一个处理芯片中都可以

注　意：6.0 版本 SDK 中 DS-42xx 卡的编码通道经过其他板卡的输出通道输出或者其他板卡的编码/解码通道通过 DS-42xx 卡的输出通道输出，均只支持单画面输出。DS-43xx 系列板卡为 PCI-E 接口，跨卡输出是否支持与具体主板相关，跨卡的音视频来源和输出目标两块板卡所在的 PCI-E 插槽支持跨卡传输则调用成功，反之则调用失败。

	DS-42xxHFV **支持 1 路输出通道**	**DS-41xxHCV** **支持 1 路输出通道**	**DS-40xxMD/MD+** **支持 x 路输出通道**	**DS-4101HD** **支持 1 路输出通道**
视频输出	5.1 版本 SDK 支持 1 路**单画面**单卡内部输出 6.0 版本 SDK 支持 1 路**单画面**内部或外部输出，且仅支持一次输出	最高支持 1 路 **16 画面**内部或外部输出。 6.0 版本 SDK 当 DS-42xx 卡编码通道码流通过 DS-41xx 卡输出时，仅支持**单画面**	最高支持**x**路**16 画面**内部或外部输出 6.0 版本 SDK 当 DS-42xx 卡编码通道码流通过 DS-40xxMD/MD+卡输出	最高支持**1**路**16 画面**内部或外部输出 6.0 版本 SDK 当 DS-42xx 卡编码通道码流通过 DS-4101HD 卡输出时，仅支

			时，仅支持**单画面**	持**单画面**
视频来源	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道
视频输出 API	SetEncoderVideoExtOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoOutput SetDecoderVideoExtOutput
音频输出	支持 1 路内部或外部输出	支持 1 路内部或外部输出	支持 x 路内部或外部输出	支持 1 路内部输出
音频来源	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道	DS-40xx、DS-41xx、DS-42xx 系列的编码通道 DS-40xxMD/MD+、DS-4101HD 的解码通道
音频输出 API	SetEncoderAudioOutput SetEncoderAudioExtOutput SetDecoderAudioExtOutput	SetEncoderAudioOutput SetEncoderAudioExtOutput SetDecoderAudioExtOutput	SetEncoderAudioExtOutput SetDecoderAudioOutput SetDecoderAudioExtOutput	SetDecoderAudioOutput SetDecoderAudioExtOutput

	DS-43xxHW/HFV/HCV-E 支持 1 路输出通道	**DS-43xxHFH-E 支持 1 路输出通道**	**DS-4308MD-E 支持 8 路输出通道**	**DS-4304HD-E 支持 4 路输出通道**
视频输出	最高支持 1 路 **16 画面**输出。	最高支持 1 路 **16 画面**输出	最高支持 16 画面输出	最高支持 4 路 16 画面输出
视频来源	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道
视频输出 API	SetEncoderVideoExtOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoOutput SetDecoderVideoExtOutput	SetEncoderVideoExtOutput SetDecoderVideoOutput SetDecoderVideoExtOutput
音频输出	支持 1 路输出	支持 1 路输出	支持 8 路输出	支持 4 路输出

音频来源	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道	DS-43xx 系列的编码通道 DS-43xx 系列的解码通道
音频输出 API	SetEncoderAudioOutput SetEncoderAudioExtOutput SetDecoderAudioExtOutput	SetEncoderAudioOutput SetEncoderAudioExtOutput SetDecoderAudioExtOutput	SetEncoderAudioExtOutput SetDecoderAudioOutput SetDecoderAudioExtOutput	SetEncoderAudioExtOutput SetDecoderAudioOutput SetDecoderAudioExtOutput

注：DS-43xx 系列板卡为 PCIe 接口，跨卡输出是否支持与具体主板相关。

6.5.1 解码通道音频内部输出 SetDecoderAudioOutput

函　数：　int __stdcall SetDecoderAudioOutput(UINT nDecodeChannel,BOOL bOpen,UINT nOutputChannel)

参　数：

UINT nDecodeChannel；解码通道索引

BOOL bOpen；使能

UINT nOutputChannel；音频输出通道索引

返回值：

ERR_INVALID_DEVICE：nDecodeChannel 溢出、nOutputChannel 取值错误；

ERR_KERNEL：其他情况

说　明：　设置解码通道音频内部输出。将第 nDecodeChannel 个解码通道解码的音频输出到第 nOutputChannel 路输出通道。如果已有其他解码通道解码的音频在 nOutputChannel 路输出通道上输出，则系统会自动先将其停止。

例如：MD 卡每个 DSP 包含 4 个解码通道及 2 个音频输出通道，则 nOutputChannel 值只能取为 0 或 1。DS4308MD-E 每个 DSP 有 8 个显示通道，所以该参数可以取值 0、1、2、3、4、5、6、7。

6.5.2 解码通道音频外部输出 SetDecoderAudioExtOutput

函　数：　int __stdcall SetDecoderAudioExtOutput(UINT nDecodeChannel,UINT nPort,BOOL bOpen, UINT nOutChannel,UINT nReserved);

参　数：

UINT nDecodeChannel；解码通道索引

UINT nPort；二次输出，取值为 0 或 1，每个解码通道最多可供音频输出 2 次

BOOL bOpen；使能

UINT nOutChannel；音频输出通道索引，以系统中所有输出通道顺序排列，取值为 0、1、2、3、4……

UINT nReserved；保留参数

返回值：　成功返回 0；失败返回错误号

说　明：　设置音频矩阵输出功能，将音频数据从第 nDecodeChannel 个解码通道的第 nPort 路，输出到系统中任意第 nOutChannel 个输出通道上。

6.5.3 编码通道音频内部输出 SetEncoderAudioOutput

函　数：　int __stdcall SetEncoderAudioOutput(UINT nEncodeChannel,BOOL bEnable,UINT nOutputChannel);

参　数：

UINT nEncodeChannel；编码通道索引

BOOL bEnable；使能

UINT nOutputChannel；音频输出通道索引。DS-41xxHCV、DS-42xxHFV 卡每张板卡只有一个音频输出口，取值只能为 0。

返回值：　成功返回 0；失败返回错误号

说　明：　编码通道音频内部输出功能。

6.5.4 编码通道音频外部输出 SetEncoderAudioExtOutput

函　数：　int __stdcall SetEncoderAudioExtOutput(UINT nEncodeChannel,UINT nPort,BOOL bOpen, UINT nOutChannel,UINT nReserved);

参　数：

UINT nEncodeChannel；编码通道索引

UINT nPort；二次输出，取值为 0 或 1，每个编码通道最多可供输出 2 次

BOOL bOpen；使能

UINT nOutChannel；音频输出通道索引，以系统中所有音频输出通道顺序排列，取值为 0、1、2、3、4……

UINT nReserved；保留参数

返回值：　成功返回 0；失败返回错误号

说　明：　编码通道音频矩阵输出功能，将音频数据从第 nEncodeChannel 个编码通道的第 nPort 路，输出到系统中任意第 nOutChannel 个音频输出口上。

6.5.5 解码通道视频内部输出 SetDecoderVideoOutput

函　数：　int __stdcall SetDecoderVideoOutput(UINT nDecodeChannel,UINT nPort,BOOL bOpen, UINT nDisplayChannel,UINT nDisplayRegion,UINT nReserved)

参　数：

UINT nDecodeChannel；解码通道索引

UINT nPort；二次输出，取值为 0 或 1，每个解码通道最多可供输出 2 次

BOOL bOpen；使能，当 bOpen 为 0，则 nDisplayChannel nDisplayRegion 无意义

UINT nDisplayChannel；显示通道号

UINT nDisplayRegion；显示区域

UINT nReserved；保留参数

返回值：

ERR_INVALID_DEVICE：nDecodeChannel 溢出、nPort 大于 1、nDisplayChannel 溢出，nDisplayRegion 超过该显示通道已划分的区域个数。

ERR_KERNEL：其它情况

说　明：　设置解码通道的视频输出功能。将视频图像从第 nDecodeChannel 个解码通道的第 nPort 路，输

出到此解码通道所在的 DSP 的第 nDisplayChannel 个显示通道的第 nDisplayRegion 个显示区域上。

DS-40xxMD/MD+卡每个 DSP 只有 2 个显示通道，所以 nDisplayChannel 取 0 或 1；DS-4308MD-E 卡每个 DSP 有 8 个显示通道，nDisplayChannel 取值范围为 0、1、2、3、4、5、6、7。

6.5.6 解码通道视频外部输出 SetDecoderVideoExtOutput

函　数：　int __stdcall SetDecoderVideoExtOutput(UINT nDecodeChannel,UINT nPort,BOOL bOpen, UINT nDisplayChannel,UINT nDisplayRegion,UINT nReserved)

参　数：

UINT nDecodeChannel；解码通道索引

UINT nPort；二次输出，取值为 0 或 1，每个解码通道最多可供视频输出 2 次

BOOL bOpen：使能，当 bOpen 为 0，则 nDisplayChannel nDisplayRegion 无意义

UINT nDisplayChannel；显示通道索引，以系统中所有显示通道顺序排列，取值为 0、1、2、3、4……

UINT nDisplayRegion；显示区域

UINT nReserved；保留参数

返回值：

ERR_INVALID_DEVICE：nDecodeChannel 溢出、nPort 大于 1、nDisplayChannel 溢出，nDisplayRegion 超过该显示通道已划分的区域个数。

ERR_KERNEL：其它情况

说　明：　设置解码通道视频矩阵输出功能。将视频图像从第 nDecodeChannel 个解码通道的第 nPort 路，输出到系统中第 nDisplayChannel 个显示通道的第 nDisplayRegion 个显示区域上。

6.5.7 编码通道视频输出 SetEncoderVideoExtOutput

函　数：　int __stdcall SetEncoderVideoExtOutput(UINT nEncodeChannel,UINT nPort,BOOL bOpen, UINT nDisplayChannel,UINT nDisplayRegion,UINT nFrameRate)

参　数：

UINT nEncodeChannel；编码通道索引

UINT nPort；二次输出，取值为 0 或 1，每个编码通道做多可供视频输出 2 次

BOOL bOpen；使能，当 bOpen 为 0，则 nDisplayChannel nDisplayRegion 无意义

UINT nDisplayChannel；显示通道索引，以系统中所有显示通道顺序排列，取值为 0、1、2、3、4……

UINT nDisplayRegion；显示区域

UINT nFrameRate；帧率

返回值：

ERR_INVALID_DEVICE：nDecodeChannel 溢出、nPort 大于 1、nDisplayChannel 溢出，nDisplayRegion 超过该显示通道已划分的区域个数。

ERR_KERNEL：其它情况

说　明：　此函数适用于 DS-4xxx 系列编码卡和 MD/MD+/HD 卡混插的情况，或者在具备视频输出功能的板卡上(如 DS-41xx 系列、DS-42xx 系列、DS-43xx 系列)，可将编码通道的视频数据直接输出至 DS-40xxMD/MD+卡、DS-4101HD 卡或者 DS-41xx 系列、DS-42xx 系列、DS-43xx 系列卡的视频输出口上，实现视频矩阵输出功能。将视频图像从第 nEncodeChannel 个编码通道的第 nPort 路，输出到系统中第 nDisplayChannel 个显示通道的第 nDisplayRegion 个显示区域上。

6.6获取通道能力集

智能能力

```
typedef enum _VCA_CAPABILITY_
{
    NONE_CAPABILITY          = 0x0    不支持任何智能能力
    BEHAVIOR_CAPABILITY      = 0x1    支持行为分析
    PLATE_CAPABILITY         = 0x2    支持车牌识别
    FACE_CAPABILITY          = 0x4    支持人脸识别
}VCA_CAPABILITY;
```

智能通道类型

```
typedef enum _VCA_CHANNEL_TYPE_
{
    NORMAL_CHAN                 = 0    普通通道
    VCA_CHAN_ON_VCA_BOARD       = 1    智能板卡上的智能通道
    NORMAL_CHAN_ON_VCA_BOARD    = 2    智能板卡上的非智能通道
}VCA_CHANNEL_TYPE;
```

通道类型

```
typedef enum _CHANNEL_TYPE_
{
    ENCODE_CHANNEL          = 1    编码通道
    DECODE_CHANNEL          = 2    解码通道
    DISPLAY_CHANNEL         = 3    显示通道
}CHANNEL_TYPE;
```

反隔行模式

```
typedef enum _DEINTERLACE_MODE_
{
    NO_DEINTERLACE          = 0x0    不支持反隔行
    DEINTERLACE_MODE1       = 0x1    模式 1
    DEINTERLACE_MODE2       = 0x2    模式 2
    DEINTERLACE_MODE3       = 0x4    模式 3
} DEINTERLACE_MODE;
```

音频预览

```
typedef enum _AUDIO_PREVIEW_
{
    AUDIO_PREVIEW_NOT_SUPPORT   =0x00000000    不支持音频预览
    AUDIO_PREVIEW_HARD          =0x00000001    硬件连线音频预览
    AUDIO_PREVIEW_SOFT          =0x00000002    软件模式音频预览
} AUIDO_PREVIEW;
```

图像分辨率

typedef enum _CAP_PICTURE_FORMAT_
{
CAP_PICTURE_FORMAT_INVALID = 0x00000000,
CAP_PICTURE_FORMAT_QCIF = 0x00000001
QCIF 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_CIF = 0x00000002
CIF 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_2CIF = 0x00000004
2CIF 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_DCIF = 0x00000008
DCIF 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_4CIF = 0x00000010
4CIF 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_VGA = 0x00000020
VGA 分辨率，该定义只适用于能力集
}CAP_PICTURE_FORMAT;

图像分辨率(扩展)

typedef enum _CAP_PICTURE_FORMAT_EX_
{
CAP_PICTURE_FORMAT_XGA = 0x00000001 1024*768 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_SXGA = 0x00000002 1280*1024 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_SXGA_960= 0x00000004 1280*960 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_720 = 0x00000008 1280*720 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_1080 = 0x00000010 1920*1080 分辨率，该定义只适用于能力集
CAP_PICTURE_FORMAT_UXGA = 0x00000020 1600*1200 分辨率，该定义只适用于能力集
}CAP_PICTURE_FORMAT_EX;

通道能力集

typedef struct _CHANNEL_CAPABILITY_EX_
{
BOARD_TYPE_DS boardType 通道所属板卡类型
CHANNEL_TYPE channelType 通道类型
VCA_CHANNEL_TYPE vcaChanType 智能通道类型
DWORD dwVcaCapability 智能能力，见 VCA_CAPABILITY
DWORD dwStreamPackType 主通道支持的码流打包格式，暂时没有使用
DWORD dwSubStreamPackType 子通道支持的码流打包格式，暂时没有使用
DWORD dwVideoCodec 主通道支持的视频算法，暂时没有使用
DWORD dwSubVideoCodec 子通道支持的视频算法，暂时没有使用
DWORD dwAudioCodec 主通道支持的音频算法，暂时没有使用
DWORD dwSubAudioCodec 子通道支持的音频算法，暂时没有使用
DWORD dwInputVideoPosition; 是否支持调整视频输入的位置，DS-42xx 系列板卡暂时不

支持

DWORD dwMbcsOsdMode; 多字节字符集 OSD 支持的相关功能，0 支持 0~255 的 8bit 灰度调节的字符，1 只支持黑色和白色字符， 2 表示不支持该 OSD 模式(只有编码通道支持该 OSD 模式)。DS-42xx 系列板卡亮度调节暂时只支持黑色和白色

DWORD dwUnicodeOsdMode Unicode OSD 支持的相关功能，0 支持字符亮度调节，支持字符和背景可设置颜色，支持字符和背景的半透明； 1 只支持字符的黑色和白色，不支持字符亮度调节，不支持字符和背景可设置颜色，只支持字符的半透明。DS-42xx 系列板卡亮度调节暂时只支持黑色和白色

DWORD dwImageStream 是否支持原始码流获取

DWORD dwEncoderPIP 是否支持编码预览画中画功能

DWORD dwAdaptiveMotionDetection 是否支持移动侦测自适应，DS-42xx 系列板卡暂时不支持

DWORD dwEncoderAudioOutput 是否支持编码通道音频输出，非矩阵，DS-40xx 系列板卡不支持，DS-4101HD 卡不支持

DWORD dwEncoderAudioExtOutput 是否支持编码通道音频矩阵输出

DWORD dwAudioPreview 支持音频预览的类型，0 不支持，1 支持硬件连线，2 支持 PCI 音频预览，3 两者都支持

DWORD dwSubVideoCapture 是否支持子通道编码

DWORD dwEncoderPictureFormat 支持的主通道编码分辨率，DS-4016HCS 卡和非扩展模式下的 DS-40xxHS 卡只支持 CIF 以及小于 CIF 的编码分辨率，海外版本 DS-40xxHS 卡在扩展模式下可以支持到 4cif，DS-4216HC 不支持大于 CIF 的分辨率编码，DS-42xxHFV 卡支持 4CIF/DCIF/2CIF/CIF/QCIF。

DWORD dwEncoderPictureFormatEx 支持的主通道编码分辨率，DS-4016HCS 卡和非扩展模式下的 DS-40xxHS 卡只支持 CIF 以及小于 CIF 的编码分辨率，海外版本 DS-40xxHS 卡在扩展模式下可以支持到 4cif，4216HC 不支持大于 CIF 的分辨率编码，DS-42xxHFV 卡支持 4CIF/DCIF/2CIF/CIF/QCIF。对应 PICTURE_FORMAT_EX 中的类型.

注 意:dwEncoderPictureFormat 对应 CAP_PICTURE_FORMAT 结构,描述了 QCIF 到 VGA 的分辨率，主要用于描述标清分辨率的编码；dwEncoderPictureFormatEx 对应 CAP_PICTURE_FORMAT_EX 结构，描述了 SVGA 到 UXGA 的分辨率，主要用于描述 VGA 以上高清分辨率的编码，暂时没有用到，

DWORD dwSubEncoderPictureFormat 支持的子通道编码分辨率,DS-42xx 系列板卡暂时支持 CIF 以及小于 CIF 的编码分辨率。对应 PICTURE_FORMAT 中的类型

DWORD dwSubEncoderPictureFormatEx 支持的子通道编码分辨率,DS-42xx 系列板卡暂时支持 CIF 以及小于 CIF 的编码分辨率。对应 PICTURE_FORMAT_EX 中的类型

DWORD dwWatermark 是否支持水印，42xx 板卡暂时不支持

DWORD dwStreamCRC 是否支持 CRC 校验，42xx 板卡暂时不支持

DWORD dwStreamEncrypt 是否支持码流加密

DWORD dwEncoderParam 是否支持调整主通道的编码框架和编码复杂度

DWORD dwSubEncoderParam 是否支持调整子通道的编码框架和编码复杂度

DWORD dwEncoderVideoOutput 是否支持编码通道视频输出，非矩阵，DS-40xx 系列和 DS-42xx 系列板卡不支持，DS-4116HCV 第二颗 dsp 的通道不支持

DWORD dwEncoderVideoExtOutput 是否支持编码通道视频矩阵输出

DWORD dwWatchDog 是否支持 watchdog，只有 DS-4016HCS 卡支持

DWORD dwAlarmIn 是否支持报警输入，只有 DS-4016HCS 卡支持
DWORD dwAlarmOut 是否支持报警输出，只有 DS-4016HCS 卡支持
DWORD dwDeInterlace 支持的反隔行模式，见 DEINTERLACE_MODE
DWORD dwDrawEncoderLineRect 是否支持解码通道 DSP 画线画框
DWORD dwDisplayFormat 支持的视频输出的格式，具体定义参考 DISPLAY_FORMAT
DWORD dwReserved[20] 保留字
}CHANNEL_CAPABILITY_EX;

通道能力集类型，和 CHANNEL_CAPABILITY_EX 结构体中各个成员对应

typedef enum _CHANNEL_CAPABILITY_TYPE_
{
NormalCapability = 1 CHANNEL_CAPABILITY_EX 结构体中的能力对应的函数外的其它函数对应的能力
VcaChanType = 2 智能通道类型，对应 vcaChanType
VcaCapability = 3 智能能力，对应 dwVcaCapability
StreamPackType = 4 主通道支持的码流打包格式，对应 dwStreamPackType
SubStreamPackType = 5 子通道支持的码流打包格式，对应 dwSubStreamPackType
VideoCodec = 6 主通道支持的视频算法，对应 dwVideoCodec
SubVideoCodec = 7 子通道支持的视频算法，对应 dwSubVideoCodec
AudioCodec = 8 主通道支持的音频算法，对应 dwAudioCodec
SubAudioCodec = 9 子通道支持的音频算法，对应 dwSubAudioCodec
InputVideoPosition = 10 是否支持调整视频输入的位置，对应 dwInputVideoPosition
MbcsOsdMode = 11 多字节字符集 OSD 支持的相关功能，对应 dwMbcsOsdMode
UnicodeOsdMode = 12 Unicode OSD 支持的相关功能，对应 dwUnicodeOsdMode
ImageStream = 13 是否支持原始码流获取，对应 dwImageStream
EncoderPIP = 14 是否支持画中画功能，对应 dwEncoderPIP
AdaptiveMotionDetection = 15 是否支持移动侦测自适应，对应 dwAdaptiveMotionDetection
EncoderAudioOutput = 16 是否支持编码通道音频输出，非矩阵，对应 dwEncoderAudioOutput
EncoderAudioExtOutput = 17 是否支持编码通道音频矩阵输出，对应 dwEncoderAudioExtOutput
AudioPreview = 18 支持音频预览的类型，对应 dwAudioPreview
SubVideoCapture = 19 是否支持子通道编码，对应 dwSubVideoCapture
EncoderPictureFormat = 20 支持的主通道编码分辨率，对应 dwEncoderPictureFormat
EncoderPictureFormatEx = 21 支持的主通道编码分辨率，对应 dwEncoderPictureFormatEx
SubEncoderPictureFormat = 22 支持的子通道编码分辨率，对应 dwSubEncoderPictureFormat
SubEncoderPictureFormatEx = 23 支持的子通道编码分辨率，对应 dwSubEncoderPictureFormatEx
Watermark = 24 是否支持水印，对应 dwWatermark
StreamCRC = 25 是否支持 CRC 校验，对应 dwStreamCRC
StreamEncrypt = 26 是否支持码流加密，对应 dwStreamEncrypt
EncoderParam = 27 是否支持调整主通道的编码框架和编码复杂度，对应 dwEncoderParam
SubEncoderParam = 28 是否支持调整子通道的编码框架和编码复杂度，对应 dwSubEncoderParam
EncoderVideoOutput = 29 是否支持编码通道视频输出，非矩阵，对应 dwEncoderVideoOutput
EncoderVideoExtOutput= 30 否支持编码通道视频矩阵输出， 对应 dwEncoderVideoExtOutput
WatchDog = 31 是否支持 watchdog，对应 dwWatchDog

```
    AlarmIn                 = 32  是否支持报警输入，对应 dwAlarmIn
    AlarmOut                = 33  是否支持报警输出，对应 dwAlarmOut
    DeInterlace             = 34  支持的反隔行模式，对应 dwDeInterlace
    DrawEncoderLineRect     = 35  是否支持解码通道 DSP 画线画框,对应 dwDrawEncoderLineRect
    DisplayFormat           = 36  支持的视频输出格式，对应 dwDisplayFormat
} CHANNEL_CAPABILITY_TYPE;
```

6.6.1 获取通道的能力集 GetChannelCapability

函　数：　int __stdcall GetChannelCapability(CHANNEL_TYPE chanType, UINT chan, CHANNEL_CAPABILITY_EX * pCapability);

输入参数：

CHANNEL_TYPE chanType；通道类型

UINT chan；通道序号

输出参数：

CHANNEL_CAPABILITY_EX * pCapability；通道能力集

返回值：成功返回 0，失败返回 ERR_KERNEL

6.6.2 根据函数名获取对应的通道能力

CheckFunctionChannelCapability

函数对应的通道能力结构, CheckFunctionChannelCapability 接口函数返回

```
typedef struct _FUNCTION_CHANNEL_CAPABILITY_
{
    CHANNEL_CAPABILITY_TYPE dwChanCapType1；通道能力类型 1
    CHANNEL_CAPABILITY_TYPE dwChanCapType2；通道能力类型 2
    DWORD                   dwChanCapValue1；通道能力值 1
    DWORD                   dwChanCapValue2；通道能力值 2
    DWORD                   dwReserved[8];
}FUNCTION_CHANNEL_CAPABILITY;
```

函　数：　int __stdcall CheckFunctionChannelCapability(CHANNEL_TYPE chanType, UINT chan,
const char *functionName, FUNCTION_CHANNEL_CAPABILITY *pFuncChanCap);

输入参数：

CHANNEL_TYPE chanType；通道类型

UINT chan；通道序号

const char *functionName；接口函数名

输出参数：

FUNCTION_CHANNEL_CAPABILITY *pFuncChanCap；函数对应的通道能力结构

chanCapType1；　输入函数名对应的通道能力类型

chanCapType2； 输入函数名对应的通道能力类型

pCapValue1； 输入函数名对应的通道能力值

pCapValue2； 输入函数名对应的通道能力值

返回值：成功返回 0，失败返回 ERR_KERNEL

注 意：在 FUNCTION_CHANNEL_CAPABILITY 结构中 chanCapType1 对应 pCapValue1，chanCapType2 对应 pCapValue2。

附录

可以用新版函数替代功能或者无效的 API

GetTotalChannels：可用 GetEncodeChannelCount 替代

GetTotalDSPs：可用 GetDspCount 替代

SetupDateTime：4.0 版本起无效

HW_GetChannelNum：无效，请使用 GetBoardDetail

HW_GetDeviceSerialNo：无效，请使用 GetBoardDetail

HW_SetVideoOutStandard：无效，请使用 SetDisplayStandard

HW_SetDspDeadlockMsg：无效

HW_ResetDsp：无效

HW_SetDisplayPara：DISPLAY_PARA 结构中 bToVideoOut 无效，解码卡模拟视频输出功能已经整合到视频矩阵之中。

函数说明

获取总的编码通道个数：GetTotalChannels

函数： int GetTotalChannels()

返回： 获取系统内可使用的通道个数，如果返回小于系统中安装的通道数，表明有一 DSP 初始化失败

获取系统内正确安装的编码通道个数：GetTotalDSPs

函数： int GetTotalDSPs()

返回： 获取系统内正确安装的编码通道个数，如果返回小于系统中安装的通道数，表明有一 DSP 初始化失败；

注意： 此函数返回系统中所有的编码通道个数，若要获取 DSP 个数请使用函数 GetDspCount

注意：在原 H 卡、D 卡 SDK 中，因为当时 DSP 和编、解码通道存在着一一对应的关系，所以可以使用 GetTotalDSPs 来取得系统中所有的编码和解码通道个数，但是在新的 HC 卡、MD 系统中，DSP 个数不再和通道个数相等，使用 GetTotalDSPs 会带来歧异，因此在 3.0 版本的 SDK 中做了完善，分别增加了获取板卡个数 GetBoardCount、DSP 个数 GetDspCount、编码通道个数 GetEncodeChannelCount、解码通道个数 GetDecodeChannelCount、显示通道个数 GetDisplayChannelCount 的 API，建议用户使用新提供的 API，不再使用原来的 GetTotalDSPs，同时为了保持兼容性，GetTotalDSPs 仍然返回系统中所有的编码通道个数，其功能和 GetEncodeChannelCount 相同，并不代表 DSP 个数，需要特别注意

设置 OSD 时间(用于网络校时)：SetupDateTime

函数： int SetupDateTime(HANDLE hChannelHandle, SYSTEMTIME *now)

参数：

HANDLE hChannelHandle 通道句柄

SYSTEMTIME *now 需要设置的时间指针值，如为 NULL，即本地校时

返回： 正确为 0，其他为第 4 节定义的错误号；

说明： 设置 OSD 中的时间，可用于网络校时。设置了此函数后，SetOsd( )的默认本地校时的功能被屏蔽。

注意：此接口函数自 v4.0 版本 SDK 开始，不再有效。

获得某个 DSP 对应的通道号 HW_GetChannelNum

函数： int__stdcall HW_GetChannelNum(long nDspNum,long *pChannelNum,
ULONG nNumsToGet,ULONG * pNumsGotten)

输入参数：

nDspNum DSP 号。

nNumsToGet 要得到通道号的个数。即 pChannelNum 分配的个数。

输出参数：

pChannelNum 返回的通道号，一个 DSP 可能对应不只一个通道。但目前(1.0ver)是一一对应的，这里只是为以后扩展做一个保留。

pNumsGotten 获得实际得到的通道号个数。可以先指定 pChannelNum=NULL, nNumsToGet=0, 调用这个函数来获得实际的通道个数，然后为 pChannelNum 分配合适内存大小，再调用这个函数。

返回值： 错误码

说 明： 此接口函数无效，请使用 GetBoardDetail。

获取卡的序列号 HW_GetDeviceSerialNo

函数：int__stdcall HW_GetDeviceSerialNo(HANDLE hChannel,ULONG *pDeviceSerialNo);

输入参数： hChannel 通道句柄；

输出参数：*pDeviceSerialNo 卡的序列号。

返回值： 错误码

说 明： 此接口函数无效，请使用 GetBoardDetail。

设置 VideoOut 输出制式 HW_SetVideoOutStandard

函数： int __stdcall HW_SetVideoOutStandard(HANDLE hChannel,ULONG nStandard);

输入参数：

hChannel 通道句柄

nStandard HW_PAL 为 PAL 制；HW_NTSC 为 NTSC 制；

返回值： 错误码

说 明： 此接口函数无效，请使用 SetDisplayStandard。

设置 DSP 死掉后向主机发送的消息 HW_SetDspDeadlockMsg

函数： int __stdcall HW_SetDspDeadlockMsg(HWND hWnd,UINT nMsg);

输入参数：

hWnd 接收消息窗口的句柄。

nMsg dsp 死掉后向主机发送的消息。wParam 返回 DSP 号。

返回值： 错误码

说 明： 此接口函数无效。

重置 DSP HW_ResetDsp

函数： int __stdcall HW_ResetDsp(long nDspNum)

输入参数：nDspNum 要重置的 DSP 号。

返回值： 错误码

说明： 重置 DSP。当用户收到 DSP 死掉后发送的消息后可以调用这个接口来重置 DSP。用户应该保存当前的工作状态，当重置 DSP 后恢复当前工作状态，即所有的操作需要重新

来一边(如打开文件，播放等)。用户可以通过 HW_GetChannelNum 来获得当前 DSP 对应的通道号。

说　明：　此接口函数无效。

获取板卡特殊功能信息 GetCapability

函　数：　int __stdcall GetCapability(HANDLE hChannelHandle, CHANNEL_CAPABILITY *Capability)

参　数：

HANDLE　hChannelHandle；通道句柄

CHANNEL_CAPABILITY *Capability；特殊功能

返回值：　成功返回 0；失败返回错误号

说　明：　获取板卡特殊功能信息

特殊功能结构体

typedef struct tagChannelCapability{

UCHAR bAudioPreview；音频预览

UCHAR bAlarmIO；报警信号，不支持

UCHAR bWatchDog；看门狗，不支持

}CHANNEL_CAPABILITY, *PCHANNEL_CAPABILITY

获取板卡 SDK 及 DSP 错误报告 GetLastErrorNum*

此函数只对 H 卡有效

函　数：　int __stdcall GetLastErrorNum(HANDLE hChannelHandle, ULONG *DspError, ULONG *SdkError)

参　数：

HANDLE hChannelHandle；通道句柄

ULONG *DspError；DSP 错误

ULONG *SdkError；SDK 错误

返回值：　DSP 错误信息、SDK 错误信息

说　明：　获取 SDK 及 DSP 错误报告。此函数只对 H 卡有效，用于 HC 卡上返回 0 且无效

设置视频制式 SetVideoStandard*

此函数只对 H 卡有效

函　数：　int __stdcall SetVideoStandard(HANDLE hChannelHandle, VideoStandard_t VideoStandard)

参　数：

HANDLE hChannelHandle；通道句柄

VideoStandard_t VideoStandard；视频制式

返回值：　成功返回 0；失败返回错误号

说　明：　设置视频制式，在某一制式的摄像头已经接好的情况下启动系统时可不必调用该函数，如果没有接摄像头的情况下启动系统然后再接 NTSC 制式的摄像头则必须调用该函数，或者中途调换不同制式的摄像头也必须调用该函数。

复位 DSP ResetDSP**

此函数目前无效

函　数：　int __stdcall ResetDSP(int DspNumber);

参　数：　int DspNumber；DSP 索引号

返回值：　成功返回 0；失败返回错误号

说　明：　复位某个 DSP，注意请谨慎调用该函数，请确定 DSP 故障无法软件恢复时再关闭相关的资源后复位 DSP。

设置看门狗 SetWatchDog

函　数：　int __stdcall SetWatchDog(UINT boardNumber,BOOL bEnable)

参　数：

UINT boardNumber；板卡索引

BOOL bEnable；使能

返回值：　成功返回 0；失败返回错误号

说　明：　设置看门狗。DS-4016HCS 提供 4pin 复位接口，用户需要把主机机箱的 Reset 接线连接到板卡上相邻的 2pin 复位接口，板卡上的另外相邻的 2pin 接口连接到主板的 Reset，这样就可以实现对上层软件和系统中所有压缩板卡的运行状态监控。

科技呵护未来

First Choice for Security Professionals