Software Development Kit

Model 2246 | Version 1.5.0

SENSORAY | embedded electronics

Designed and manufactured in the U.S.A.

SENSORAY | p. 503.684.8005 | email: info@SENSORAY.com | www.SENSORAY.com

7313 SW Tech Center Drive | Portland, OR 97203

Table of Contents

Introduction

Sensoray's 2246SDK is a software development kit that has been developed to allow OEM's to build their own applications for using the 2246 video capture board without knowledge of the driver or DirectX.

Software Installation

Requirements

Minimum processor: Pentium III 600 MHz. A Pentium IV 2 GHz or faster is recommended.
Operating system: Windows 2000, or Windows XP. Windows 98 or NT are not supported.
Minimum system RAM: 128 Mbytes. 256 Mbytes or more is recommended.
Video card: A high performance video card is highly recommended. The video card must support Microsoft DirectX and should have at least 64MB of video ram
DirectX: Version 8.1 or more recent. If you do not already have DirectX installed on your system (Windows XP installations include DirectX), you must obtain a DirectX runtime package from Microsoft. The exact version you need depends on your operating system and can be downloaded directly from Microsoft's web site at the following URL:
http://www.microsoft.com/downloads/search.aspx?displaylang=en&categoryid=2

MPEG2 Decoder: Windows does not come with a default MPEG2 Decoder. For best results, use one of the supported decoders (Intervideo or FFDshow).

.NET 2003 MFC classes and runtime libraries.

2246SDK Demo Application

Description

The 2246 SDK demo or sample application is a working windows application that allows you to display in real-time multiple MPEG-1, MPEG-2, and MPEG-4 video streams from the 2246 board. Additionally, it allows for Disk Space Management, manual and automated recording of the MPEG streams, selectable video input sources, uncompressed frame capture, and playback.

The 2246SDK sample application is a simple MFC (Microsoft Foundation Classes) windows application. The source code for the application is included and demonstrates the usage of the 2246SDK API (application program interface) functions. The API consists of a DLL (mid2246.dll), two header (mid2246func.h, mid2246const.h) files, and a library file for MS Visual Studio projects (mid2246.lib).

Jpeg compression is done using freeware Independent Jpeg Group libraries.

Building an application with the 2246SDK

Files included with the SDK

The following files are distributed with the 2246SDK:

Drivers directory:

2246 drivers(in Drivers directory on the CD)

API directory:

- mid2246funct.h – contains exported API functions
  mid2246const.h -- data types, constant definitions
- mid2246.dll – dll library
- mid2246.lib – dll library
  ovlgen.dll – overlay helper library
  alleg40.dll – overlay helper library
  alfont.dll – overlay helper library

API/Filters:

These files, if installed separately (or moved), must be registered with regsvr32.
Ffmp2encoder.ax – Audio encoder
Gmfbridge.dll – Bridge function for controlling recording
libvlc.dll – VLC Muxer
Smartdump.ax – free AMM filter
SrayVlcBridge.ax – Mux for recording to file.
Wavdest.ax – DXSDK sample filter to write .wav audio clips

API/WinDir (For the windows directory)

dviparam.ini (not to be modified, except by Sensoray Inc)
s2246param.ini (may be modified by advanced users)

API/WinDir/dshow_vlc_plugins

libaccess_directory_plugin.dll
libaccess_output_file_plugin.dll
libaraw_plugin.dll
libaudio_format_plugin.dll
libdshowbridge_plugin.dll
libmemcpy_plugin.dll
libmpeg_audio_plugin.dll
libmpga_plugin.dll
libmpgv_plugin.dll
libmux_ps_plugin.dll
libpacketizer_copy_plugin.dll
libpacketizer_mpegvideo_plugin.dll
libplaylist_plugin.dll

libps_plugin.dll
libstream_out_duplicate_plugin.dll
libstream_out_es_plugin.dll
libstream_out_gather_plugin.dll
libstream_out_standard_plugin.dll
libstream_out_transcode_plugin.dll

API/Doc

2246API.doc

External dependencies:

Ffdshow:

One of the recommended decoder filters for Mpeg 2 and 1.
Mpeg2 licensing fees must be paid to use Mpeg2 and ffdshow should be configured from the start menu to use MPEG1and2.

mfc71.dll, msvcr71.dll: Usually distributed with .NET 2003 environment.

s2246param.ini

s2246param.ini in the windows directory (%WINDIR%) contains the codec specifications for decoding MPEG streams.

The [General] section contains general configuration parameters. MuteAudio specifies whether audio is played back on the host computer or not.

Following the General section is a section for each Codec([MPEG4], [MPEG2], [MPEG1]). ClockAudio specifies what source to use for synchronization of the filter graph if MuteAudio=0. It is highly recommended to leave this value and the value ClockNoAudio(MuteAudio=1) as the defaults.

StreamFilter specifies the decoding codec to use (with specified GUID) for decoding an MPEG stream from the Capture card. Specify the desired GUID.
PlayFilter specifies the decoding codec to use for file playback in the API.

DVEncodeFilter specifies the encoding codec to use (with specified GUID) for encoding video captured from the 1394 DV input.

An example ini file is shown below:

```
[General]
MuteAudio=0  ; do not mute audio


[MPEG4]
; clock -1==NULL, 0--filter alg, 1--system, 2--audio renderer, 3--demux
ClockAudio=2
ClockNoAudio=1
; Filter Choices
; INTERVIDEO  {0246CA20-776D-11D2-8010-00104B9B8592}
; FFDSHOW     {04FE9017-F873-410E-871E-AB91661A4EF7}
; Microsoft MP4S {75838A0D-B431-4C31-9487-C5A96DD39EF4} version 8?
; Microsoft MP4 {82CCD3E0-F71A-11D0-9FE5-00609778EA66}  version 7?
; Microsoft Mpg1 {FEB50740-7BEF-11CE-9BD9-0000E202599C}

StreamFilter={04FE9017-F873-410E-871E-AB91661A4EF7}
PlayFilter={75838A0D-B431-4C31-9487-C5A96DD39EF4}

[MPEG2]
ClockAudio=2
ClockNoAudio=1
StreamFilter={04FE9017-F873-410E-871E-AB91661A4EF7}
PlayFilter={04FE9017-F873-410E-871E-AB91661A4EF7}

[MPEG1]
ClockAudio=2
ClockNoAudio=1
StreamFilter={04FE9017-F873-410E-871E-AB91661A4EF7}
PlayFilter={FEB50740-7BEF-11CE-9BD9-0000E202599C}

[DV]
DVEncodeFilter={4DB2B5D9-4556-4340-B189-AD20110D953F}
```

mid2246 Data Structure Documentation

image_raw_t Struct Reference

Data Fields

- BYTE * **f0**
- int **size0**
- BYTE * **f1**
- int **size1**

Detailed Description

raw image structure

Parameters:

f0 pointer to field 0
size0 size of field 0
f1 pointer to field 1 or null if not present
size1 size of field 1

Field Documentation

BYTE* f0

BYTE* f1

int size0

int size1

The documentation for this struct was generated from the following file:

- mid2246func.h

MID2246STATS Struct Reference

Data Fields

- ULONG **iFilesize**
- TCHAR **szFilepath**[MAX_PATH]
- BOOL **bIsRecording**
- BOOL **bIsPlaying**
- TCHAR **szClipFilepath**[MAX_PATH]

Detailed Description

General Status information

Parameters:

bIsRecording true if currently recording
bIsPlaying true if video playing(for debug)
iFilesize current recorded file size and size of all clips
szFilepath current filepath being recorded
szClipFilepath is the where data is being written to currently (MPEG1 and MPEG2)

Field Documentation

ULONG iFilesize

TCHAR szFilepath[MAX_PATH]

BOOL bIsRecording

BOOL bIsPlaying

TCHAR szClipFilepath[MAX_PATH]

The documentation for this struct was generated from the following file:

- mid2246func.h

MPEG_CONFIG Struct Reference

Data Fields

- int **mpeg**
- int **bitrate**
- int **vbr**
- int **max_vbr_bitrate**
- int **framerate**

Detailed Description

MPEG_CONFIG specifies the mpeg encoding parameters

Parameters:

mpeg mpeg type, 1 2 or 4
bitrate desired bitrate
vbr variable bitrate flag (0--constant bitrate, 1 VBR)
max_vbr_bitrate peak bitrate if using VBR
framecurrent current framerate (use value in mid2246const.h)

Field Documentation

int bitrate

int framerate

int max_vbr_bitrate

int mpeg

int vbr

The documentation for this struct was generated from the following file:

- mid2246const.h

overlay_text_t Struct Reference

Data Fields

- char **text** [256]
- char **fontfile** [256]
- int **fontsize**
- int **txtcolor**
- int **backcolor**
- char **reserved** [10]

Detailed Description

overlay text structure

Parameters:

text pointer to text
fontfile full name/path to font file
fontsize height of font in pixels
txtcolor text color in RGB format, example RGB(255,255,255)
backcolor Background color, zero or less means transparent. (use small value eg. RGB(0,0,1) for black)

Field Documentation

int backcolor

char fontfile[256]

int fontsize

char reserved[10]

char text[256]

int txtcolor

The documentation for this struct was generated from the following file:

- mid2246func.h

mid2246 File Documentation

mid2246const.h File Reference

Data Structures

- struct MPEG_CONFIG

Defines

- #define MAX_DEVICES 30
- #define MAX_TEXTSIZE 256
- #define MID2246_SUCCESS 0
- #define MID2246_ERR_NONE 0
- #define MID2246_ERR_UNSPECIFIED -1
- #define MID2246_ERR_CAPTURING -2
- #define MID2246_ERR_NOTCAPTURING -3
- #define MID2246_ERR_RUNNING -4
- #define MID2246_ERR_SNAPSHOT -5
- #define MID2246_ERR_INVALIDFILE -6
- #define MID2246_ERR_DIRECTX -7
- #define MID2246_ERR_NODEVICE -8
- #define MID2246_ERR_NOTINITIALIZED -9
- #define MID2246_ERR_INVALIDMODE -10
- #define MID2246_ERR_NOMEM -11
- #define MID2246_ERR_INVALIDPARAM -12
- #define MID2246_ERR_INVALIDUSERDATA -13
- #define MID2246_ERR_NONEWDATA -14
- #define MID2246_ERR_PAUSERESUME_MUX -15
- #define MID2246_FRAMERATE_DEFAULT 0
- #define MID2246_FRAMERATE_NTSC_2997 1
- #define MID2246_FRAMERATE_NTSC_15 2
- #define MID2246_FRAMERATE_NTSC_10 3
- #define MID2246_FRAMERATE_NTSC_5 4
- #define MID2246_FRAMERATE_NTSC_24 5
- #define MID2246_LEVEL_CONTRAST 1
- #define MID2246_LEVEL_BRIGHTNESS 2
- #define MID2246_LEVEL_SATURATION 4
- #define MID2246_LEVEL_HUE 8
- #define MID2246_FILE_JPEG 1
- #define MID2246_FILE_BMP 2
- #define MID2246_FILE_PPM 4
- #define MID2246_REGION_MONITOR 1
- #define MID2246_REGION_MPEG 2
- #define MID2246_REGION_STILL 4
- #define MID2246_BOARD_A 1
- #define MID2246_BOARD_H 2
- #define MID2246_BOARD_K 3

Enumerations

- enum MID2246_VIDSYS { MID2246_VIDSYS_PAL = 1, MID2246_VIDSYS_NTSC = 2 }

- enum MID2246_SOURCE { MID2246_SOURCE_COMPOSITE_0 = 0, MID2246_SOURCE_COMPOSITE_1 = 1, MID2246_SOURCE_SVIDEO_0 = 2, MID2246_SOURCE_SVIDEO_1 = 3, MID2246_SOURCE_DV = 4, MID2246_SOURCE_SDI = 5, MID2246_SOURCE_HDSDI = 6, MID2246_SOURCE_HDSDI2 = 7, MID2246_SOURCE_NUM = 8 }
- enum MID2246_AUDIO_INPUT { MID2246_AUDIO_LINE, MID2246_AUDIO_MIC }
- enum MID2246_FIELDALG { MID2246_FIELDALG_NONE = 0, MID2246_FIELDALG_DUP = 1, MID2246_FIELDALG_MERGE = 2, MID2246_FIELDALG_INTER = 3, MID2246_FIELDALG_CUST = 4 }
- enum MID2246_DVIRES { MID2246_DVIRES_W1280H1024 = 0, MID2246_DVIRES_W1600H1200 = 1, MID2246_DVIRES_W720H480 = 2, MID2246_DVIRES_W720H576 = 3, MID2246_DVIRES_W1920H1080 = 4 }
- enum MID2246_ASPECT_MODE { MID2246_ASPECT_NONE = 0, MID2246_ASPECT_CONST = 1 }
- enum MID2246_REC { MID2246_REC_MUX = 0, MID2246_REC_VES = 1, MID2246_REC_AUDIOWAVE = 2, MID2246_REC_AES = 3 }
- enum MID2246_RC { MID2246_RC_OUTSPACE = 1, MID2246_RC_RECTIME = 2, MID2246_RC_MANUALSTOP = 3, MID2246_RC_RECCLIPSIZE = 4, MID2246_RC_RECCLIPSIZEANDSTOP = 5 }
-

Define Documentation

#define MAX_DEVICES 30
#define MAX_TEXTSIZE 256
#define MID2246_BOARD_A 1
#define MID2246_BOARD_H 2
#define MID2246_BOARD_K 3
#define MID2246_ERR_CAPTURING -2
#define MID2246_ERR_DIRECTX -7
#define MID2246_ERR_INVALIDFILE -6
#define MID2246_ERR_INVALIDMODE -10
#define MID2246_ERR_INVALIDPARAM -12
#define MID2246_ERR_INVALIDUSERDATA –13
#define MID2246_ERR_NODEVICE -8
#define MID2246_ERR_NOMEM -11
#define MID2246_ERR_NONE 0
#define MID2246_ERR_NONEWDATA -14
#define MID2246_ERR_NOTCAPTURING -3
#define MID2246_ERR_NOTINITIALIZED -9
#define MID2246_ERR_PAUSERESUME_MUX -15
#define MID2246_ERR_RUNNING -4
#define MID2246_ERR_SNAPSHOT -5
#define MID2246_ERR_UNSPECIFIED -1
#define MID2246_FILE_BMP 2
#define MID2246_FILE_JPEG 1
#define MID2246_FILE_PPM 4
#define MID2246_FRAMERATE_DEFAULT 0
#define MID2246_FRAMERATE_NTSC_10 3
#define MID2246_FRAMERATE_NTSC_15 2
#define MID2246_FRAMERATE_NTSC_24 5
#define MID2246_FRAMERATE_NTSC_2997 1
#define MID2246_FRAMERATE_NTSC_5 4
VIDEO levels
#define MID2246_LEVEL_BRIGHTNESS 2
#define MID2246_LEVEL_CONTRAST 1
#define MID2246_LEVEL_HUE 8
#define MID2246_LEVEL_SATURATION 4
#define MID2246_REGION_MONITOR 1
#define MID2246_REGION_MPEG 2
#define MID2246_REGION_STILL 4
#define MID2246_SUCCESS 0

Enumeration Type Documentation

enum MID2246_ASPECT_MODE

Aspect ratio settings for MPEG decoding on host computer. Depending on Decoder, may not have an effect.

Parameters:

MID2246_ASPECT_NONE no aspect ratio change on MPEG decoding(stretch to window)
MID2246_ASPECT_CONST maintain aspect ratio as video settings

Enumerator:

MID2246_ASPECT_NONE
MID2246_ASPECT_CONST

enum MID2246_AUDIO_INPUT

MID2246_AUDIO_INPUT sets the audio input channel,

Parameters:

MID2246_AUDIO_LINE line in
MID2246_AUDIO_MIC microphone

Enumerator:

MID2246_AUDIO_LINE
MID2246_AUDIO_MIC

enum MID2246_DVIRES

DVI output resolution

Enumerator:

MID2246_DVIRES_W1280H1024
MID2246_DVIRES_W1600H1200
MID2246_DVIRES_W720H480
MID2246_DVIRES_W720H576
MID2246_DVIRES_W1920H1080

enum MID2246_FIELDALG

Interlace field merge algorithms.

Parameters:

MID2246_FIELDALG_NONE no field manipulation is done
MID2246_FIELDALG_DUP duplicate field 0
MID2246_FIELDALG_MERGE merge field 0 and 1
MID2246_FIELDALG_INTER interpolate between field 0
MID2246_FIELDALG_CUST custom algorithm

Enumerator:

MID2246_FIELDALG_NONE
MID2246_FIELDALG_DUP
MID2246_FIELDALG_MERGE
MID2246_FIELDALG_INTER

MID2246_FIELDALG_CUST

enum MID2246_RC

finished recording message values to calling application

Parameters:

MID2246_RC_OUTSPACE out of space on disk
MID2246_RC_RECTIME record time specified exceeded
MID2246_RC_MANUALSTOP manually stopped
MID2246_RC_RECCLIPSIZE clip size exceeded, starting new clip
MID2246_RC_RECCLIPSIZEANDSTOP clip size exceeded, stop recording and notify user

Enumeration values:

MID2246_RC_OUTSPACE
MID2246_RC_RECTIME
MID2246_RC_MANUALSTOP
MID2246_RC_RECCLIPSIZE
MID2246_RC_RECCLIPSIZEANDSTOP

enum MID2246_REC

recording mode for saved MPEG streams

Parameters:

MID2246_REC_MUX multiplexed(video+audio) mpeg stream
MID2246_REC_VES video elementary stream(video only)
MID2246_REC_AUDIOWAVE audio only. recorded as a WAVE file.
MID2246_REC_AES encoded audio only (AES mp2 encoding)

Enumerator:

MID2246_REC_MUX
MID2246_REC_VES
MID2246_REC_AUDIOWAVE
MID2246_REC_AES

enum MID2246_SOURCE

Video input source. Note: Changing input may stop MPEG Video Stream

Parameters:

MID2246_SOURCE_COMPOSITE_0 Composite channel 0
MID2246_SOURCE_COMPOSITE_1 Composite channel 1
MID2246_SOURCE_SVIDEO_0 S-Video Channel 0
MID2246_SOURCE_SVIDEO_1 S-Video Channel 1
MID2246_SOURCE_DV digital video firewire input (aka DV camcorder)
MID2246_SOURCE_SDI SDI input
MID2246_SOURCE_HDSDI HD-SDI input. High definition 1080i at 59.94Hz frame rate (1.485/1.001Gbps)
MID2246_SOURCE_HDSDI2 HD-SDI2 input. High definition 1080i at 60Hz frame rate (1.485 Gbps) or 50Hz frame rate.
MID2246_SOURCE_NUM maximum number of inputs

Enumerator:

MID2246_SOURCE_COMPOSITE_0
MID2246_SOURCE_COMPOSITE_1

MID2246_SOURCE_SVIDEO_0
MID2246_SOURCE_SVIDEO_1
MID2246_SOURCE_DV
MID2246_SOURCE_SDI
MID2246_SOURCE_HDSDI
MID2246_SOURCE_HDSDI2
MID2246_SOURCE_NUM

enum MID2246_VIDSYS

video systems. PAL or NTSC

Enumerator:

MID2246_VIDSYS_PAL
MID2246_VIDSYS_NTSC

mid2246func.h File Reference

Data Structures

- struct MIDSRAYSTATUS
- struct image_raw_t
- struct overlay_text_t

Defines

- #define **MID2246_API** __declspec(dllimport)

Typedefs

- typedef int(* **mergefunc_t**)(BYTE *merged, int size, BYTE *f0, int h0, BYTE *f1, int h1, int w, int *pRetW, int *pRetH)

Functions

- MID2246_API int **SN_GetVersion**(char **sMid2246Version, char **sFPGA2246Version , int *pBoardId = 0, int board = 0)
- MID2246_API int **SN_Initialize** (HWND hwnd, HWND messageHwnd=NULL, int removeMsg=0, int board = 0)
- MID2246_API int **SN_Shutdown** (, int board = -1)
- MID2246_API int **SN_UpdateClippingWindow** (HWND hwnd, int board = 0)
- MID2246_API int **SN_SetVideoPosition** (int xpos, int ypos, int xsize, int ysize, int left_clip=0, int top_clip=0, int right_clip=0, int bottom_clip=0, int board = 0)
- MID2246_API int **SN_SetSource** (**MID2246_SOURCE** input, BOOL bTest, int board = 0)
- MID2246_API int **SN_GetSource** (**MID2246_SOURCE** *pSource, int board = 0)
- MID2246_API int **SN_SetVidSys** (**MID2246_VIDSYS** vidsys, int board = 0)
- MID2246_API int **SN_GetVidSys** (, int board = 0)
- MID2246_API int **SN_SetDvi** (**MID2246_DVIRES** res, BOOL bColorBars, int DviCropLeft=0, int DviCropRight=0, int board = 0)
- MID2246_API int **SN_SetMpegCrop** (int MpegCropLeft, int MpegCropRight, int board = 0)
- MID2246_API int **SN_SetMpegConfig** (**MPEG_CONFIG** *pConfig, int board = 0)
- MID2246_API int **SN_GetMpegConfig** (**MPEG_CONFIG** *pConfig, int board = 0)
- MID2246_API int **SN_GetStatus** (**MIDSRAYSTATUS** *pStats, int board=0)
- MID2246_API int **SN_IsValidVid**(int board = 0);
- MID2246_API int **SN_IsValidSdi**(int board = 0);
- MID2246_API int **SN_IsValidDv**(int board = 0);
- MID2246_API int **SN_EnumerateInputs** (int *pStatus, int *pVidSys, int board = 0)
- MID2246_API int **SN_StartStream** (, int board = 0)
- MID2246_API int **SN_StopStream** (, int board = 0)
- MID2246_API int **SN_StartRecord** (char *filename, int size, int breakonrecsize=0, int board = 0)
- MID2246_API int SN_**SetNotifyAtRecordEnd** (HWND hNotifyApp, UINT mNotifyMessage, int board = 0);
- MID2246_API int **SN_PauseRecord** (int board = 0)
- MID2246_API int **SN_ResumeRecord** (int board = 0)
- MID2246_API int **SN_StopRecord** (int board = 0)
- MID2246_API int **SN_SetMergeMethod** (**MID2246_FIELDALG** merge_method, **mergefunc_t** custom_alg, int board = 0)
- MID2246_API int **SN_SnapshotToFile** (char *szFilename, int filetype, int freezetime, int wait, int qual=100, CustMergeSize=0, int board = 0)

- MID2246_API int **SN_SnapshotToMem** (BYTE *image, int size, int freezetime, int wait, int board = 0)
- MID2246_API int **SN_SnapshotRaw** (BYTE *image, int size, **image_raw_t** *pImage, int freezetime, int wait, int board = 0)
- MID2246_API void **SN_StopSnapshot** (int board = 0)
- MID2246_API int **SN_GetTimeLeft** (int bitrate=0, int board = 0)
- MID2246_API int **SN_IsEnoughSpace** (int bitrate=0, int time=0, int board = 0)
- MID2246_API int **SN_GetVidStatus** (, int board = 0)
- MID2246_API int **SN_PlaybackVideo** (char *filename_plus_path, int board = 0)
- MID2246_API int **SN_StopPlayback** (int board = 0)
- MID2246_API int **SN_PausePlayback** (BOOL bPause, int board = 0)
- MID2246_API int **SN_SetNotifyDuringFilePlay** (HWND hNotifyApp, UINT mNotifyMessage, int board = 0)
- MID2246_API int **SN_PlaybackSetRate** (double drate, int board = 0)
- MID2246_API int **SN_PlaybackSetSeekPosition** (int percent, int range, int board = 0)
- MID2246_API int **SN_PlaybackGetSeekPosition** (int range, int board = 0)
- MID2246_API int **SN_DisplaySnapshot** (char *filename, int time, int board = 0)
- MID2246_API int SN_**StopDisplaySnapshot** (int board = 0)
- MID2246_API int **SN_SetMute** (BOOL bMute, int board = 0)
- MID2246_API int **SN_GetMute** (BOOL *bMute, int board = 0)
- MID2246_API int **SN_SetRecordMode** (**MID2246_REC** recMode, int board = 0)
- MID2246_API int **SN_SetRecordVideoStream** (BOOL bRecVid, int board = 0)
- MID2246_API int **SN_GetRecordVideoStream** (BOOL *bRecVid, int board = 0)
- MID2246_API int **SN_SetRecordAudioStream** (BOOL bRecAud, int board = 0)
- MID2246_API int **SN_GetRecordAudioStream** (BOOL *bRecAud, int board = 0)
- MID2246_API int **SN_SetAudioEncode** (BOOL bEncAud, int board = 0)
- MID2246_API int **SN_GetAudioEncode** (BOOL *bEncAud, int board = 0)
- MID2246_API int **SN_SetRenderVideo** (BOOL bDisplayVideo, int board = 0)
- MID2246_API int **SN_GetRenderVideo** (BOOL *bDisplayVideo, int board = 0)
- MID2246_API int **SN_SetAspectRatio** (MID2246_ASPECT_MODE mode, int board = 0)
- MID2246_API int **SN_GetAspectRatio** (MID2246_ASPECT_MODE *mode, int board = 0)
- MID2246_API int **SN_SetOverlayMode** (int mode, int board = 0)
- MID2246_API int **SN_OverlayText** (int xpos, int ypos, **overlay_text_t** *pOvlText, int regionmask, int board = 0)
- MID2246_API int **SN_OverlayTextIdx**(int WinIndex, int xpos, int ypos, overlay_text_t *pOvlText, int regionmask, int board = 0)
- MID2246_API int **SN_OverlayImage** (int xpos, int ypos, char *imageFile, int regionmask, int board = 0)
- MID2246_API int **SN_OverlayImageIdx**(int WinIndex, int xpos, int ypos, char *imageFile, int regionmask, int board = 0)
- MID2246_API int **SN_QuickCopyBmpToOverlayZero**(int xpos, int ypos, char *path, int board)
- MID2246_API int **SN_GetOverlayIdx**(int WinIndex, int *Type, int *Region, int *Group, int *Xpos, int *Ypos, char **Value, int board = 0)
- MID2246_API int **SN_MoveOverlay**(int WinIndex, int NewX, int NewY, int board = 0)
- MID2246_API int **SN_OverlayDelXY**(int xpos, int ypos, int board = 0)
- MID2246_API int **SN_OverlayDel**(int WinIndex, int board = 0)
- MID2246_API int **SN_OverlayBackgroundColor**(int red, int green, int blue, int board = 0)
- MID2246_API int **SN_UpdateOverlay** (int board = 0)
- MID2246_API int **SN_ClearOverlay** (int board = 0)
- MID2246_API int **SN_ClearOverlayRegion** (int regionmask, int board = 0)
- MID2246_API int **SN_SetOverlayRegion** (int regionmask, int board = 0)
- MID2246_API int **SN_SetLevels** (int param, unsigned char value, int board = 0)

- MID2246_API int **SN_SetAudioInput** (**MID2246_AUDIO_INPUT** audio, int micBoost, int board = 0)
- MID2246_API int **SN_TestDeviceRemoval** (int board = 0)
- MID2246_API int **SN_Repaint** (HDC hdc)
- MID2246_API int **SN_OnDisplayModeChanged** ()
- MID2246_API int **SN_GetNumBoards**(int *pBoards);
- MID2246_API int **SN_SetGPOutput**(int states,int board=0);
- MID2246_API int **SN_GetGPOutput**(int *states, int board=0);
- MID2246_API int **SN_GetGPInput**(int *states, int board=0);
- MID2246_API int **SN_BootD**(int board=0);
- MID2246_API int **SN_BootC**(int board=0);
- MID2246_API int **SN_LowLatency**(BOOL bON, int board);

Define Documentation

#define MID2246_API __declspec(dllimport)

MID2246 API. API functions for the 2246 board.

Author:

Sensoray Company Inc.

Version:

1.4.7

Typedef Documentation

typedef int(* mergefunc_t)(BYTE *merged, int size, BYTE *f0, int h0, BYTE *f1, int h1, int w, int *pRetW, int *pRetH)

mergefunc_t: If custom field alg used, mergefunc_t gives a callback to the raw fields to allow custom merge of the interlaced fields

Parameters:

merged pointer to allocated data for merged field(s)
size size of merged data
f0 pointer to field 0 data
h0 height of field 0
f1 pointer to field 1(not necessary used)
h1 height of field 1
w width of fields in bytes(not pixels. 3 bytes per pixel)
pRetW pointer to image width (in Pixels). Modify this to a new value if the returned image width is different. Initial value *pRetW = w/3
pRetH pointer to image height. Modify this to a new value if the returned image height is different. Initial value *pRetH = h0 + h1

Function Documentation

MID2246_API int SN_BootC(int *board*=0);

Alternate boot of Micronas video compression IC. BootC fills the initial buffer with a software-MBIST (Memory Built-In Self Test) package (OPC).

MBIST is primarily used by the manufacturer for device screening but can also be employed by the user to verify device functionality.

Parameters:

Board ID of 2246 if multiple boards are installed.

Returns:

HRESULT

MID2246_API int SN_BootD(int *board*=0);

Alternate boot of Micronas video compression IC. BootD fills the initial buffer with a Self-I/O Check Package (OPC).

Parameters:

Board ID of 2246 if multiple boards are installed.

Returns:

HRESULT

MID2246_API int SN_ClearOverlay (int *board*=0)

Clear all overlays

Parameters:

Board ID of 2246 if multiple boards are installed.

Returns:

0 on success

MID2246_API int SN_ClearOverlayRegion (int *regionmask*, int *board*=0)

Clears all overlays for specified region. Use SN_ClearOverlay to erase overlays. This function only clears the overlay by region

Parameters:

Board ID of 2246 if multiple boards are installed.

regionmask output region mask

MID2246_API int SN_DisplaySnapshot (char * *filename*, int *time*, int *board*=0)

Displays the snapshot in the video window

Parameters:

Board ID of 2246 if multiple boards installed.
filename full filename plus path plus extension
time time to display snapshot, -1 is infinite

Returns:

0 on success

MID2246_API int SN_EnumerateInputs (int * *pStatus*, int * *pVidSys*, int *board*=0)

Enumerate connected inputs. Note: This function takes over 4 seconds to complete because inputs must settle in order for the decoder to detect new video systems. Also, input channel is changed on the analog decoder.

Parameters:

Board ID of 2246 if multiple boards installed.
pStatus pointer to array of inputs of size MID2246_SOURCE_NUM. returns status for each input(0-unlocked, 1--locked, -1 unknown)
pVidSys pointer to array of inputs of size MID2246_SOURCE_NUM. returns vidsys for each input(MID2246_VIDSYS or -1 if unknown).

Returns:

0 on success

MID2246_API int SN_GetAspectRatio (MID2246_ASPECT_MODE * *mode*, int *board*=0)

Get the aspect ratio mode

Parameters:

Board ID of 2246 if multiple boards installed.
mode returned aspect ratio mode

Returns:

0 on success

MID2246_API int SN_GetAudioEncode (BOOL * *bEncAud*, int *board*=0)

Get the audio encode setting

Parameters:

Board ID of 2246 if multiple boards installed.
bEncAud pointer to audio encode setting

Returns:

0 on success

MID2246_API int SN_GetGPInput (int * *states*, int *board*=0)

Get the value present on the general purpose input pins.

Parameters:

Board ID of 2246 if multiple boards installed.
States pointer to an integer that receives the input value.

Returns:

Same value that is put in states.

MID2246_API int SN_GetGPOutput (int * *states*, int *board*=0)

Reads the value that was previously written to the general purpose output pins via **SN_SetGPOutput**.

Parameters:

Board ID of 2246 if multiple boards installed.
States pointer to an integer that receives the value in the register driving the outputs.

Returns:

Same value that is put in states variable location.

MID2246_API int SN_SetGPOutput (int *states*, int *board*=0)

Drives the general purpose output pins with the value in states.

Parameters:

Board ID of 2246 if multiple boards installed.

States integer that drives the general purpose outputs.

Returns:

0 on success.

MID2246_API int SN_GetMpegConfig (MPEG_CONFIG * *pConfig*, int *board*=0)

Retrieves the current mpeg configuration parameters

Parameters:

Board ID of 2246 if multiple boards installed.
pConfig pointer to retrieved **MPEG_CONFIG**

Returns:

0 on success, negative otherwise (see errors in mid2246const.h)

MID2246_API int SN_GetMute (BOOL * *bMute*, int *board*=0)

Gets the mute setting

Parameters:

Board ID of 2246 if multiple boards installed.
bMute pointer to retrieved mute setting

Returns:

0 on success

MID2246_API int SN_GetNumBoards (int * *pBoards*=0)

Gets the number of 2246 boards detected.

Parameters:

pBoards Number of 2246 boards detected if multiple boards are installed.

Returns:

0 on success

MID2246_API int SN_GetRecordAudioStream (BOOL * *bRecAud*, int *board*=0)

Gets the audio recording setting

Parameters:

Board ID of 2246 if multiple boards installed.
bRecAud pointer to record video setting

Returns:

0 on success

MID2246_API int SN_GetRecordVideoStream (BOOL * *bRecVid*, int *board*=0)

Gets the video recording setting

Parameters:

Board ID of 2246 if multiple boards installed.
bRecVid pointer to record video setting

Returns:

0 on success

MID2246_API int SN_GetRenderVideo (BOOL * *bDisplayVideo*, int *board*=0)

Gets the display video setting

Parameters:

Board ID of 2246 if multiple boards installed.
bDisplayVideo pointer to display video setting

Returns:

0 on success

MID2246_API int SN_GetSource (MID2246_SOURCE * *pSource*, int *board*=0)

Gets the current video source

Parameters:

Board ID of 2246 if multiple boards installed.
pSource pointer to returned value of current input source

Returns:

0 on success

MID2246_API int SN_GetStatus (MIDSRAYSTATUS * *pStats*, int *board*=0)

Retrieves current API statistics and file information (see MIDSRAYSTATUS structure above)

Parameters:

Board ID of 2246 if multiple boards installed.
pStats Pointer to MIDSRAYSTATUS structure that receives results.

Returns:

0 on success, negative otherwise (see errors in mid2246const.h)

MID2246_API int SN_GetTimeLeft (int *bitrate* = 0, int *board*=0)

Returns estimated recording time left on disk

Parameters:

Board ID of 2246 if multiple boards installed.
bitrate is optional bitrate(otherwise uses current bitrate)

Returns:

time in seconds

MID2246_API int SN_GetVidStatus (int *board*=0)

GetVidStatus returns current lock status of the current input

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

TRUE(1) if video locked else FALSE(0)

MID2246_API int SN_GetVersion(char ***s*Mid2246Version*, char ***s*FPGA2246Version,* int ***pBoardId* , int *board*=0)

Returns strings indicating the current versions of the Middleware and Firmware that are

Loaded and running.

Parameters:

Board ID of 2246 if multiple boards installed.
**sMid2246Version* is set to point to an internal string containing the version of the middleware that is currently loaded and running.

**sFPGA2246Version* is set to point to an internal string containing the version of the firmware that is currently loaded and running in the hardware.
**pBoardId* Pointer to integer receiving value of:

- S2246_BOARD_A for Board Rev A-G
- S2246_BOARD_H for Board Rev H-J
- S2246_BOARD_K for Board Rev K-L

board Board ID of 2246 if multiple boards are installed.

Returns:

Returns 0 on success.

MID2246_API int SN_GetVidSys (int *board*=0)

Get the current video system setting

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

vidsys The current video system type MID2246_VIDSYS cast to int on success.

MID2246_API int SN_Initialize (HWND *hwnd*, HWND *messageHwnd* = `NULL`, int *removeMsg* = `0`, int *board*=0)

Opens and Initializes the middleware.

Parameters:

Board ID of 2246 if multiple boards installed.
hwnd pointer to window in which to display video
messageHwnd call back window when device removed. removeMsg is sent when device removed(unplug USB cable unexpectantly). set to NULL is not wanted.
removeMsg value of remove message to send when device removed. see TestDeviceRemove below and demo app for usage of the remove message

Returns:

0 on success

MID2246_API int SN_IsEnoughSpace (int *bitrate* = `0`, int *time* = `0`, int *board*=0)

IsEnoughSpace returns true if enough space to record at specified bitrate for length of time specified. If 0, uses current configuration

Parameters:

Board ID of 2246 if multiple boards installed.
bitrate current bitrate
time time to record

MID2246_API int SN_IsValidDv(int *board*=0)

Test to see if there are any valid DV (1394/Firewire) inputs.

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

TRUE(1) if video locked else FALSE(0)

MID2246_API int SN_IsValidSdi(int *board*=0)

Tests to see if there are any valid High Definition or Standard Definition SDI video streams present.

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

TRUE(1) if video locked else FALSE(0)

MID2246_API int SN_IsValidVid(int *board*=0)

Check to see if there are any valid Composite or S-Video inputs.

Note: This function takes over 4 seconds to complete because the inputs must settle in order for the decoder to detect new video systems. Also, the input channel is changed to the analog decoder.

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

TRUE(1) if video locked else FALSE(0)

MID2246_API int SN_LowLatency(BOOL *bON*, int *board*=0)

Low Latency mode runs the direct show graph with sync clock = -1

The video will go through the graph at the maximum speed without pacing. The tradeoff is the audio/video may not be synched and the video may not be completely smooth.

Call this function before starting the stream if desired.

Parameters:

Board ID of 2246 if multiple boards installed.
bON Enable or disable low latency mode

Returns:

0 on success

MID2246_API int SN_OnDisplayModeChanged (int *board*=0)

Informs the video renderer that a WM_DISPLAYCHANGE message has been received by the application This callback should be called when the application has a WM_DISPLAYCHANGE event

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_SetOverlayMode (int *mode*, int *board*=0)

Set overlay mode.

Parameters:

Board ID of 2246 if multiple boards installed.
mode

- 0 : Use 16 bit color, 32k pixel, single buffered overlay.

- 1 : Use 9 bit color, 32k pixel double buffered overlay or
  Use 9 bit color, 64k pixel single buffered overlay.

Returns:

0 on success

MID2246_API int SN_SetOverlayRegion (int *regionmask*, int *board*=0)

Set all overlays for specified region. This function only sets regionmask for all overlays

Parameters:

Board ID of 2246 if multiple boards installed.
regionmask output region mask

Returns:

0 on success

MID2246_API int SN_OverlayImage (int *xpos*, int *ypos*, char * *imageFile*, int *regionmask*, int *board*=0)

Adds overlay image.

If overlay already exists at that x,y position, deletes window before adding.

Overlay active on regions defined by regionmask.

Parameters:

Board ID of 2246 if multiple boards installed.
xpos -- start X position
ypos -- start y position
imageFile -- pointer to overlay image(full path) 24bit BMP or PCX files supported only
regionmask -- output region mask

Returns:

0 on success, -1 on too many regions, other negative value on failure

MID2246_API int SN_OverlayImageIdx (int *WinIndex*, int *xpos*, int *ypos*, char * *imageFile*, int *regionmask*, int *board*=0)

Adds/Update overlay image.

If overlay already exists at index WinIndex, deletes window before adding.

Overlay active on regions defined by regionmask.

Parameters:

Board ID of 2246 if multiple boards installed.
WinIndex -- sub-window position to update/add image
xpos -- start X position
ypos -- start y position
imageFile -- pointer to overlay image(full path) 24bit BMP or PCX files supported only
regionmask -- output region mask

Returns:

0 on success, -1 on too many regions, other negative value on failure

MID2246_API int SN_QuickCopyBmpToOverlayZero (int *xpos*, int *ypos*, char * *imageFile*, int *board*=0)

Blit new sub-window to image in overlay zero.

- Assumes Overlay at Index[0] is loaded into current memory AND double buffer. (Loaded twice)

-Assumes Overlay at Index [0] is first overlay in memory (Address offset 0)

-Assumes Overlay at Index [0] has even number of lines

-Assumes Overlay at Index [0] has a width that is divisible by 128

-Assumes all other overlays (Overlay at Index [1-7] are contained within the boundaries of Overlay at Index [0]

- The region will be the same as the Overlay at index[0].

Parameters:

Board ID of 2246 if multiple boards installed.
xpos -- start X position (relative to Overlay 0)
ypos -- start y position (relative to Overlay 0)
imageFile -- pointer to overlay image(full path) 24bit BMP or PCX files supported only

Returns:

0 on success, -1 on too many regions, other negative value on failure

MID2246_API int SN_OverlayText (int *xpos*, int *ypos*, overlay_text_t * *pOvlText*, int *regionmask*, int *board*=0)

Adds/Update overlay text.

If overlay already exists that x,y position, deletes window before adding.

Overlay active on regions defined by regionmask.

If text contains embedded newline characters (\n = 10 dec), then each line of text will be created on in a new window, AtIndex sub-window position, directly below the preceding line. Each sub-window position only consumes enough overlay memory needed to hold the individual line.

If text contains embedded character 30 dec (entered programmatically or by holding down Alt- and typing "030" on the numeric keypad), then each line following a char(30) will be on a new line. Multi-line text created this way will be created as one large graphic at one AtIndex location.

Parameters:

Board ID of 2246 if multiple boards installed.
xpos -- start X position
ypos -- start y position
pOvlText -- pointer to overlay text
regionmask -- region mask (MID2246_REGION_MONITOR, MID2246_REGION_MPEG, MID2246_REGION_STILL)

Returns:

0 on success, -1 on too many regions, other negative value failure

MID2246_API int SN_OverlayTextIdx (int *WinIndex*, int *xpos*, int *ypos*, overlay_text_t * *pOvlText*, int *regionmask*, int *board*=0)

Adds overlay text.

If overlay already exists at index WinIndex, deletes window before adding.

Overlay active on regions defined by regionmask.

Multi-lines created as defined in **SN_OverlayText**

Parameters:

Board ID of 2246 if multiple boards installed.
WinIndex – sub-window position to update/add text
xpos -- start X position
ypos -- start y position
pOvlText -- pointer to overlay text
regionmask -- region mask (MID2246_REGION_MONITOR, MID2246_REGION_MPEG, MID2246_REGION_STILL)

Returns:

0 on success, -1 on too many regions, other negative value failure

MID2246_API int SN_GetOverlayIdx (int *WinIndex* , int **Type*, int **Region*, int **Group*, int **Xpos*, int **Ypos*, char ***Value*, int *board*=0)

Get basic parameters of an overlay at WinIndex.

Parameters:

Board ID of 2246 if multiple boards installed.
WinIndex -- sub-window position to get parameters from
Type -- Type: 1=Image 0=Text
Region -- Overlay Region, MID2246_REGION_MONITOR | MID2246_REGION_MPEG | MID2246_REGION_STILL
Group -- Text with common group number is kept together. (-1 indicates not used)
Xpos -- start X position
Ypos -- start y position
Value -- pointer to text or image file path

Returns:

0 on success

MID2246_API int SN_MoveOverlay (int *WinIndex*, int *NewX*, int *NewY*, int *board*=0)

Change position of an overlay sub-window at index WinIndex.

Does not reload or download the new image. (Use SN_UpdateOverlay)

Parameters:

Board ID of 2246 if multiple boards installed.
WinIndex -- index of sub-window to move
xpos -- new X position
ypos -- new y position

Returns:

0 on success , -1 overlay does not exist yet.

MID2246_API int SN_OverlayDelXY (int *xpos*, int *ypos*, int *board*=0)

Delete an overlay at x,y location.

Does not reload or download the new image. (Use SN_UpdateOverlay)

Parameters:

Board ID of 2246 if multiple boards installed.
xpos -- X position of sub-window to delete
ypos -- y position of sub-window to delete

Returns:

return 0 on success, -1 overlay does not exist yet, -2 if that sub-window already unused.

MID2246_API int SN_OverlayDel (int *WinIndex*, int *board*=0)

Delete an overlay at a given index.

Does not reload or download the new image. (Use SN_UpdateOverlay)

Parameters:

Board ID of 2246 if multiple boards installed.
WinIndex -- index of sub-window to delete

Returns:

return 0 on success, -1 overlay does not exist yet.

MID2246_API int SN_OverlayBackgroundColor (int *red*, int *green*, int *blue*, int *board*=0)

Set the color used for the background regions

Parameters:

Board ID of 2246 if multiple boards installed.
red -- 0 min, 255 max red
green -- 0 min, 255 max green
blue -- 0 min, 255 max blue

Returns:

0 on success, -1 on too many regions, other negative value failure

MID2246_API int SN_PausePlayback (BOOL *bPause*, int *board*=0)

Pause/resume playback

Parameters:

Board ID of 2246 if multiple boards installed.
bPause bPause = TRUE(pause), bPause=FALSE(resume)

Returns:

0 on success

MID2246_API int SN_PauseRecord (int *board*=0)

Pause recording

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_PlaybackGetSeekPosition (int *range*, int *board*=0)

Retrieves current position(percent of total stream) in the playback stream

Parameters:

Board ID of 2246 if multiple boards installed.
range defines the percentage granularity

Returns:

percent, an integer from 0(start) to *range*(end) or -1 on error

MID2246_API int SN_PlaybackSetRate (double *drate*, int *board*=0)

Change playback rate

Parameters:

Board ID of 2246 if multiple boards installed.
drate is a double specifying playback speed. 0.5 == half, 2.0 == double minimum speed is 0.5

Returns:

return 0 on success, -value otherwise

MID2246_API int SN_PlaybackSetSeekPosition (int *percent*, int *range*, int *board*=0)

Seek to a position relative to the percent of total stream length.

Parameters:

Board ID of 2246 if multiple boards installed.
percent is an integer from 0(start) to *range*(end)
range defines the percentage granularity

Returns:

0 on success -1 otherwise

MID2246_API int SN_PlaybackVideo (char * *filename_plus_path*, int *board*=0)

plays back the specified video clip in the current window default is first

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_Repaint (HDC *hdc*)

Repaint callback should be called when application receives a WM_PAINT event. This is necessary to notify the video renderer of a repaint event.

Parameters:

hdc device context handle

Returns:

HRESULT S_OK on success.

MID2246_API int SN_ResumeRecord (int *board*=0)

Resume recording

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_SetAspectRatio (MID2246_ASPECT_MODE *mode*, int *board*=0)

Set aspect ratio flag

Parameters:

Board ID of 2246 if multiple boards installed.
mode MID2246_ASPECT_NONE--stretched, MID2246_ASPECT_CONST--maintain aspect ratio

Returns:

0 on success, -value on error

MID2246_API int SN_SetAudioEncode (BOOL *bEncAud*, int *board*=0)

Sets the audio encoder setting

Parameters:

Board ID of 2246 if multiple boards installed.
bEncAud encode audio or not.

Returns:

0 on success

MID2246_API int SN_SetAudioInput (MID2246_AUDIO_INPUT *audio*, int *micBoost*, int *board*=0)

Set audio input channel (default is LINE in)

Parameters:

Board ID of 2246 if multiple boards installed.
audio is the audio channel input line-in, MID2246_AUDIO_LINE, MID2246_AUDIO_MIC
micBoost 0--normal, 1--+20dB

Returns:

0 on success

MID2246_API int SN_SetDvi (MID2246_DVIRES *res*, BOOL *bColorBars,* int *DviCropLeft* = 0, int *DviCropRight* = 0, int *board*=0)

Set the DVI resolution and optional built-in colorbar test mode

Parameters:

Board ID of 2246 if multiple boards installed.
res : enumerated resolution (MID2246_DVIRES)
bColorBars,: Set Colorbar test pattern on or off
DviCropLeft: Clip left of image by this value then stretch to fit.
DviCropRight: Clip right of image by this value then stretch to fit.

Returns:

0 on success

MID2246_API int SN_SetLevels (int *param*, unsigned char *value*, int *board*=0)

Set the brightness, contrast and saturation

Parameters:

Board ID of 2246 if multiple boards installed.
param is the parameter to change (MID2246_LEVEL_CONTRAST, MID2246_LEVEL_BRIGHTNESS, MID2246_LEVEL_SATURATION, MID2246_LEVEL_HUE)
value is the value of the parameter (0--mininum, 128 nominal, 255 max)

Returns:

0 on success

MID2246_API int SN_SetMergeMethod (MID2246_FIELDALG *merge_method*, mergefunc_t *custom_alg*, int *board*=0)

Set the interlaced field merging algorithm

Parameters:

Board ID of 2246 if multiple boards installed.
merge_method is the method in mid2246const.h. if custom, specify function custom_alg

custom_alg is the custom merge algorithm. (see above mergefunc_t description)

MID2246_API int SN_SetMpegConfig (MPEG_CONFIG * *pConfig*, int *board*=0)

Sets the mpeg configuration parameters(see MPEG_CONFIG above).

Parameters:

Board ID of 2246 if multiple boards installed.
pConfig pointer to **MPEG_CONFIG** structure

Returns:

0 on success, negative otherwise (see errors in mid2246const.h)

MID2246_API int SN_SetMpegCrop (int *MpegCropLeft*, int *MpegCropRight*, int *board*=0)

Set the MPEG Hardware Cropping

Parameters:

Board ID of 2246 if multiple boards installed.
MpegCropLeft Clip left of image by this value then stretch to fit.
MpegCropRight Clip right of image by this value then stretch to fit.

Returns:

0 on success

MID2246_API int SN_SetMute (BOOL *bMute*, int *board*=0)

Mutes the audio on the host computer. Audio will still be recorded. Note: if currently streaming(StartStream used), this function will stop the stream and restart it.

Parameters:

Board ID of 2246 if multiple boards installed.
bMute whether to mute audio on PC

Returns:

0 on success

MID2246_API int SN_SetNotifyAtRecordEnd (HWND *hNotifyApp*, UINT *mNotifyMessage*, int *board*=0)

Enables or Disables sending of message indicating that recording has stopped.

Parameters:

Board ID of 2246 if multiple boards installed.
hNotifyApp Handle of application to send notification message to.
mNotifyMessage message to send calling application when recording done

- The WPARAM parameter of the message will return one of the following reasons for why the recording was stopped.
  - MID2246_RC_OUTSPACE out of space on disk
  - MID2246_RC_RECTIME record time specified exceeded
  - MID2246_RC_MANUALSTOP manually stopped
  - MID2246_RC_RECCLIPSIZE clip size exceeded, starting new clip
  - MID2246_RC_RECCLIPSIZEANDSTOP clip size exceeded, stop recording and notify user
- The LPARAM of the message will contain return a pointer to a string containing the name of the recorded file that was just closed.

Returns:

none

MID2246_API int SN_SetNotifyDuringFilePlay (HWND *hNotifyApp*, UINT *mNotifyMessage*, int *board*=0)

Enable the SN_PlaybackVideo function to send messages to an application indicating the status of a playing video file.

To support this callback the called back application must include the following:

- #include "control.h" // for definition of IMediaEventEx
- #include "evcode.h" // for EC_COMPLETE (EC_COMPLETE = 0x01)

In addition, the call back function must call IMediaEventEx->FreeEventParams(...) each time it accesses the event via IMediaEventEx->GetEvent(...) to free allocated memory. (See the example in DemoDlg.cpp)

Parameters:

Board ID of 2246 if multiple boards installed.
hNotifyApp Handle of application to send notification message to.
mNotifyMessage message to send calling application when recording done

- The WPARAM parameter of the called message is undefined.
- The LPARAM parameter of the message will return a pointer to the IMediaEvent interface that is connected to the file play graph.

Returns:

none

MID2246_API int SN_SetRecordAudioStream (BOOL *bRecAud*, int *board*=0)

Sets whether audio is recorded or not. NOTE: this function is different from Mute. SetMute only turns off audio playback on the PC.

Parameters:

Board ID of 2246 if multiple boards installed.
bRecAud whether to record audio or not

Returns:

0 on success

MID2246_API int SN_SetRecordMode (MID2246_REC *recMode*, int *board*=0)

SetRecordMode encapsulates the SetRecord functions. It is the preferred method of setting the parameters. In which case, the individual functions SN_SetRecordVideoStream, SN_SetRecordAudioStream and SN_SetRecordAudioEncode are NOT required.

Parameters:

Board ID of 2246 if multiple boards installed.
recMode MID2246_REC setting (see mid2246const.h)

Returns:

0 on success

MID2246_API int SN_SetRecordVideoStream (BOOL *bRecVid*, int *board*=0)

Sets video recording. NOTE: this function is different from startrecord. If RecordVideoStream is turned off, no video stream is sent to the recording module

Parameters:

bRecVid whether to record video or not
Re *Board* ID of 2246 if multiple boards installed.

turns:

0 on success

MID2246_API int SN_SetRenderVideo (BOOL *bDisplayVideo*, int *board*=0)

SetRenderVideo allows one to turn off the video window but still stream the video after SN_StartStream. Video can also be recording with this setting off, but won't be displayed.

Parameters:

Board ID of 2246 if multiple boards installed.
bDisplayVideo whether to display video or not

Returns:

0 on success

MID2246_API int SN_SetSource (MID2246_SOURCE *input*, BOOL *bTest*, int *board*=0)

Sets the video input source. Note: will stop the stream when input changed.

Parameters:

Board ID of 2246 if multiple boards installed.
input is the enumerated input MID2246_SOURCE enum(composite, svideo, DV) (see mid2246const.h)
bTest whether to include colorbars or not

Returns:

0 on success

MID2246_API int SN_SetVideoPosition (int *xpos*, int *ypos*, int *xsize*, int *ysize*, int *left_clip* = 0, int *top_clip* = 0, int *right_clip* = 0, int *bottom_clip* = 0, int *board*=0)

Set the video position in the clipping window

Parameters:

Board ID of 2246 if multiple boards installed.
xpos x position of video
ypos y position of video
xsize size of video display
ysize size of video display
left_clip number of pixels at left of source video to clip
top_clip y number of pixels at top of source video to clip
right_clip number of pixels at right of source video to clip
bottom_clip number of pixels at bottom of source video to clip

Returns:

0 on success

MID2246_API int SN_SetVidSys (MID2246_VIDSYS *vidsys*, int *board*=0)

Set the current video system

Parameters:

Board ID of 2246 if multiple boards installed.
vidsys is the video system type

Returns:

0 on success

MID2246_API int SN_Shutdown (int *board*=-1)

Cleans up and shuts down the API gracefully

Parameters:

Board ID of 2246 if multiple boards installed, -1 for all boards

Returns:

0 on success

MID2246_API int SN_SnapshotRaw (BYTE * *image*, int *size*, image_raw_t * *pImage*, int *freezetime*, int *wait*, int *board*=0)

Get snapshot data only. This function does no processing on the image, it just retrieves a pointer to the raw fields(field 1 will be null for 720p format) in YCrCb format

Parameters:

Board ID of 2246 if multiple boards installed.
image is a pointer to pre-allocated space for image
size is the size of image array above
pImage is a pointer to the returned raw image fields
freezetime is the time, in milli-sec to freeze the image on the monitor. (a minimum value greater than the video frame rate is required)
wait wait =1 will wait if operations pending(board locked), wait=0 returns error if board busy

Returns:

number of bytes read (image size) or -1 on error.

MID2246_API int SN_SnapshotToFile (char * *szFilename*, int *filetype*, int *freezetime*, int *wait*, int *qual* = `100, int` *CustMergeSize*, int *board*=0)

takes a snapshot and saves to file in filename

Parameters:

Board ID of 2246 if multiple boards installed.
szFilename is the fully qualified file + path (without extension)
filetype is the file type to save(see mid2246const.h: MID2246_FILE_JPEG and/or MID2246_FILE_BMP)
freezetime is the time, in milli-sec to freeze the image. (minimum value greater than the video frame rate is required)
wait wait =1 will wait if operations pending(board locked), wait=0 returns error if board busy
qual JPEG quality, (use 25-100, default 100).
CustMergeSize Size in bytes of the buffer sent to the custom image merge call back set by SN_SetMergeMethod(). Default value 0 indicates that a buffer that is the same as the raw image size will be created when calling the merge callback routine.

Returns:

Number of bytes read (image size) or -1 on error.

MID2246_API int SN_SnapshotToMem (BYTE * *image*, int *size*, int *freezetime*, int *wait*, int *board*=0)

Get snapshot to memory function. Retrieves processed image into memory. Image is converted to YCrCB and merged using the algorithm in SetMergeMethod.

Parameters:

Board ID of 2246 if multiple boards installed.
image IN/OUT pointer to retrieved snapshot.
size is the size of the array above (input)
freezetime is the time, in milli-sec to freeze the image on the monitor. (a minimum value greater than the video frame rate is required)
wait wait =1 will wait if operations pending(board locked), wait=0 returns error if board busy

Returns:

number of bytes read (image size) or -1 on error.

MID2246_API int SN_StartRecord (char * *filename*, int *size*, int *breakonrecsize*=0, int *board*=0)

Starts recording to file using filter graph previously set up.

Parameters:

Board ID of 2246 if multiple boards installed.
filename is the filename to record to. Use the full path, no extension
size is the size of the files to record in Megabytes if size full, then a new file started with an incremental number appended to the filename.
breakonrecsize flag indicating whether to, 1=end recording when filesize reaches size, or 0=continue recording to new file.

Returns:

0 on success.

MID2246_API int SN_StartStream (int *board*=0)

starts the streaming of video and audio

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success.

MID2246_API int SN_StopDisplaySnapshot (int *board*=0)

Stops displaying current snapshot in rendering window

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_StopPlayback (int *board*=0)

stops video playback

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_StopRecord (int *board*=0)

stops recording to file. video/sound preview window will continue

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API void SN_StopSnapshot (int *board*=0)

stops pending snapshots

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_StopStream (int *board*=0)

stops video/audio stream and stops recording if recording

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_TestDeviceRemoval (int *board*=0)

Test whether device was removed or not When SN_Initialize is called, a message window handle and remove message is set. When remove message is received by the application, this function should be called to see if the device was removed. DLL should be shut down if device terminated

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 if not removed, 1 if removed

MID2246_API int SN_UpdateClippingWindow (HWND *hwnd*, int *board*=0)

update the clipping window for the video

Parameters:

hwnd pointer to video display window
Board ID of 2246 if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_UpdateOverlay (int *board*=0)

refresh or update the overlay to the hardware

Parameters:

Board ID of 2246 if multiple boards installed.

Returns:

0 on success

Sample code

See 2246demo code

Setting Custom Merge Method for interlace reconstruction

First set the merge method:

```
SN_SetMergeMethod( MID2246_FIELDALG_CUST, (mergefunc_t ) custmerge_fields);
```

Where custmerge_fields follows the mergefunc_t declaration. Finally, define a merge function. Two examples of a custom merge algorithm are shown below:

```
/** merge fields
 *  @param newf pointer to allocated space for new field
 *  @param size size of newf
 *  @param f0 pointer to field 0
 *  @param h0 height of field 0
 *  @param f1 pointer to field 1 (may be unused in some cases)
 *  @param h1 height of field 1  (may be unused in some cases)
 *  @param w  width of fields in bytes
 *  @param pRetW pointer to image width (in Pixels).  Modify this to a new value if
 *  the returned image width is different. Initial value *pRetW = w/3
 *  @param pRetH pointer to image height. Modify this to a new value if
 *  the returned image height is different. Initial value *pRetH = h0 + h1
 *  @return 0 on success, -1 otherwise
 */
int custmerge_fields( BYTE *newf, int size,  BYTE *f0, int h0, BYTE *f1,
                      int h1, int w, int *pRetW, int *pRetH)
{
    int i,j,h;
    j = 0;
    h = (h0 < h1) ? h0 : h1;
    for( i=0; i< h; i++, j+=2) {
        memcpy( newf + j*w,     &f0[i*w], w);
        memcpy( newf + (j+1)*w, &f1[i*w], w);
    }
    return 0;
}

int custmerge_fields_with_clip( BYTE *newf, int size,  BYTE *f0, int h0, BYTE *f1,
                                int h1, int w, int *pRetW, int *pRetH)
{
    int i,h,h;
    int TrimXLeft   = 4; // Require Trimxxx to be multiples of 2
    int TrimXRight  = 8; // to make line calculations easier.
    int TrimYTop    = 4;
    int TrimYBottom = 6;
    int TotalLines;
    i = TrimYTop/2;
    h = (h0 < h1) ? h0 : h1;
    TotalLines = 2*(h - ((TrimYTop+TrimYBottom)/2));
    for( j=0; j < TotalLines; i++, j+=2) {
        memcpy( newf + ( j   *(w - 3*(TrimXLeft+TrimXRight))) ,
                &f0[(i*w) + (3*TrimXLeft)],
                w - (3*(TrimXLeft+TrimXRight)) ); // 3 bytes per pixel
        memcpy( newf + ((j+1)*(w - 3*(TrimXLeft+TrimXRight))) ,
                &f1[(i*w) + (3*TrimXLeft)],
                w - (3*(TrimXLeft+TrimXRight)) ); // 3 bytes per pixel
    }
    *pRetW = *pRetW - (TrimXLeft+TrimXRight);  // In pixels
    *pRetH = TotalLines;                       // In pixels
    return 0;
}
```

NOTE: The size of BYTE *newf buffer set by size parameter in SN_SnapshotToMem() or CustMergeSize parameter in SN_SnapshotToFile()