

2246
Software
Development Kit
Version 1.7.6

© Sensoray 2013
7313 SW Tech Center Dr.
Tigard, OR 97223
Phone 503.684.8005 • Fax 503.684.8164
sales@sensoray.com
www.sensoray.com



Table of Contents

Table of Contents.....	2
Introduction.....	6
Software Installation.....	7
Requirements.....	7
2246SDK Demo Application.....	7
Description.....	7
Building an application with the 2246SDK.....	8
mid2246 Data Structure Documentation.....	11
image_raw_t Struct Reference.....	11
Public Attributes.....	11
Detailed Description.....	11
Member Data Documentation.....	11
BYTE* image_raw_t::f0.....	11
BYTE* image_raw_t::f1.....	11
int image_raw_t::size0.....	11
int image_raw_t::size1.....	11
The documentation for this struct was generated from the following file:.....	11
MIDSRAYSTATUS Struct Reference.....	12
Public Attributes.....	12
Detailed Description.....	12
Member Data Documentation.....	12
BOOL MIDSRAYSTATUS::bIsPlaying.....	12
BOOL MIDSRAYSTATUS::bIsRecording.....	12
ULONG MIDSRAYSTATUS::iFilesize.....	12
char MIDSRAYSTATUS::szFilepath[MAX_PATH].....	12
char MIDSRAYSTATUS::szReserved[MAX_PATH].....	12
The documentation for this struct was generated from the following file:.....	12
MIDSRAYSTATUS_UNICODE Struct Reference.....	13
Public Attributes.....	13
Member Data Documentation.....	13
BOOL MIDSRAYSTATUS_UNICODE::bIsPlaying.....	13
BOOL MIDSRAYSTATUS_UNICODE::bIsRecording.....	13
ULONG MIDSRAYSTATUS_UNICODE::iFilesize.....	13
WCHAR MIDSRAYSTATUS_UNICODE::szFilepath[MAX_PATH].....	13
char MIDSRAYSTATUS_UNICODE::szReserved[MAX_PATH].....	13
The documentation for this struct was generated from the following file:.....	13
File Documentation.....	16
mid2246const.h File Reference.....	16
Classes.....	16
Defines.....	16
Error constants.....	16
Defines.....	16
Snapshot error constants.....	16
Defines.....	16
Frame rate Constants.....	17
Defines.....	17
Maximum count of MID2246_SOURCE_xxx enum's.....	17
Defines.....	17
Video levels.....	17
Defines.....	17
Snapshot and Overlay bitmasks for file types.....	17

Defines.....	17
Caption region display mask.....	17
Defines.....	17
Revision moniker for SDK backward capability.....	18
Defines.....	18
Enumerations.....	18
Define Documentation.....	19
#define MAX_DEVICES 30.....	19
#define MAX_TEXTSIZE 256.....	19
#define MID2246_SUCCESS 0.....	19
#define MID2246_ERR_NONE 0.....	19
#define MID2246_ERR_UNSPECIFIED -1.....	19
#define MID2246_ERR_CAPTURING -2.....	19
#define MID2246_ERR_NOTCAPTURING -3.....	19
#define MID2246_ERR_RUNNING -4.....	19
#define MID2246_ERR_SNAPSHOT -5.....	19
#define MID2246_ERR_INVALIDFILE -6.....	19
#define MID2246_ERR_DIRECTX -7.....	19
#define MID2246_ERR_NODEVICE -8.....	19
#define MID2246_ERR_NOTINITIALIZED -9.....	19
#define MID2246_ERR_INVALIDMODE -10.....	19
#define MID2246_ERR_NOMEM -11.....	19
#define MID2246_ERR_INVALIDPARAM -12.....	19
#define MID2246_ERR_INVALIDUSERDATA -13.....	19
#define MID2246_ERR_NONEWDATA -14.....	19
#define MID2246_ERR_PAUSERESUME_MUX -15.....	19
#define MID2246_ERR_INVALID_DVIPARAM -16.....	19
#define MID2246_WARN_NOPROGSDI_JITTER -17.....	19
#define MID2246_ERR_DV_COM_ERROR -18.....	19
#define MID2246_ERR_STRLEN -19.....	19
#define MID2246_SNAP_ERR_UNSPECIFIED -1.....	19
#define MID2246_SNAP_ERR_BUFSIZE -2.....	19
#define MID2246_SNAP_ERR_BIST -3.....	19
#define MID2246_SNAP_ERR_SYNCH -4.....	19
#define MID2246_SNAP_ERR_CRITSEC -5.....	19
#define MID2246_SNAP_ERR_PARTBUF -6.....	19
#define MID2246_SNAP_ERR_DVBUF -7.....	19
#define MID2246_SNAP_ERR_DVGRAB -8.....	19
#define MID2246_SNAP_ERR_USERERR -9.....	19
#define MID2246_FRAMERATE_DEFAULT 0.....	19
#define MID2246_FRAMERATE_NTSC_2997 1.....	19
#define MID2246_FRAMERATE_NTSC_15 2.....	19
#define MID2246_FRAMERATE_NTSC_10 3.....	19
#define MID2246_FRAMERATE_NTSC_5 4.....	19
#define MID2246_FRAMERATE_NTSC_24 5.....	19
#define MID2246_SOURCE_LASTSDI MID2246_SOURCE_HDSDI720P2.....	19
#define MID2246_LEVEL_CONTRAST 1.....	19
#define MID2246_LEVEL_BRIGHTNESS 2.....	19
#define MID2246_LEVEL_SATURATION 4.....	19
#define MID2246_LEVEL_HUE 8.....	19
#define MID2246_LEVEL_SATURATION_CR 4.....	20
#define MID2246_LEVEL_SATURATION_CB 8.....	20
#define MID2246_FILE_JPEG 1.....	20
#define MID2246_FILE_BMP 2.....	20
#define MID2246_FILE_PPM 4.....	20
#define MID2246_REGION_MONITOR 1.....	20

#define MID2246_REGION_MPEG_2	20
#define MID2246_REGION_STILL_4	20
#define MID2246_BOARD_A_1	20
#define MID2246_BOARD_H_2	20
#define MID2246_BOARD_K_3	20
Enumeration Type Documentation	20
enum MID2246_VIDSYS	20
enum MID2246_SOURCE	20
enum MID2246_AUDIO_INPUT	21
enum MID2246_AUDIO_OUTPUT	21
enum MID2246_FIELDALG	21
enum MID2246_DVIREs	22
enum MID2246_ASPECT_MODE	22
enum MID2246_REC	22
enum MID2246_RC	22
mid2246func.h File Reference	24
Data Structures	24
Defines	24
Typedefs	24
Functions	24
Define Documentation	26
#define MAX_PATH_260	26
#define MID2246_API__declspec(dllimport)	26
#define MID2246_MERGEAPI_int	27
#define MAX_PATH_260	27
#define MID2246_MAX_TEXT_256	27
#define MID2246_MAX_FONTFILE_256	27
Typedef Documentation	27
typedef int(* mergefunc_t)(BYTE *merged, int size, BYTE *f0, int h0, BYTE *f1, int h1, int w, int *pRetW, int *pRetH)	27
Function Documentation - Configuration and Setup functions	27
MID2246_API int SN_GetNumBoards (int * pBoards)	27
MID2246_API int SN_GetVersion (char ** sMid2246Version, char ** sFPGA2246Version, int * pBoardId, int board)	28
MID2246_API int SN_Initialize (HWND hwnd, HWND messageHwnd, int removeMsg, int board)	28
MID2246_API int SN_Shutdown (int board)	28
MID2246_API int SN_UpdateClippingWindow (HWND hwnd, int board)	28
MID2246_API int SN_SetVideoPosition (int xpos, int ypos, int xsize, int ysize, int left_clip, int top_clip, int right_clip, int bottom_clip, int board)	28
MID2246_API int SN_SetSource (MID2246_SOURCE input, BOOL bTest, int board)	29
MID2246_API int SN_GetSource (MID2246_SOURCE * pSource, int board)	29
MID2246_API int SN_SetVidSys (MID2246_VIDSYS vidsys, int board)	29
MID2246_API int SN_GetVidSys (int board)	29
MID2246_API int SN_SetDvi (MID2246_DVIREs res, BOOL bColorBars, int DviCropLeft, int DviCropTop, int DviCropRight, int DviCropBottom, int board)	30
MID2246_API int SN_SetMpegCrop (int MpegCropLeft, int MpegCropTop, int MpegCropRight, int MpegCropBottom, int board)	30
MID2246_API int SN_SetMpegConfig (MPEG_CONFIG * pConfig, int board)	30
MID2246_API int SN_GetMpegConfig (MPEG_CONFIG * pConfig, int board)	30
MID2246_API int SN_SetLevels (int param, unsigned char value, int board)	30
MID2246_API int SN_SetAudioInput (MID2246_AUDIO_INPUT audio, int micBoost, int board)	31
0 on success	31
MID2246_API int SN_SetAudioOutput (MID2246_AUDIO_OUTPUT audio, int board)	31
MID2246_API int SN_GetStatus (MIDSRAYSTATUS * pStats, int board)	31

MID2246_API int SN_GetFrameSync (int * sync, int board)	31
MID2246_API int SN_IsValidVid (MID2246_SOURCE checksource, int board)	32
MID2246_API int SN_IsValidSdi (MID2246_SOURCE checksource, int board)	32
MID2246_API int SN_IsValidDv (int board)	32
MID2246_API int SN_EnumerateInputs (int * pStatus, int * pVidSys, int board)	32
MID2246_API int SN_GetVidStatus (int board)	32
MID2246_API int SN_GetVidStatusNoBIST (int board)	33
Function Documentation - Callback functions from application	33
MID2246_API int SN_TestDeviceRemoval (int board)	33
MID2246_API int SN_Repaint (HDC hdc)	33
MID2246_API int SN_OnDisplayModeChanged (int board)	33
Function Documentation - Recording functions	33
MID2246_API int SN_StartStream (int board = 0)	33
MID2246_API int SN_StopStream (int board = 0)	34
MID2246_API int SN_LowLatency (BOOL bON, int board)	34
MID2246_API int SN_StartRecord (char * filename, int size, int breakonrecsize = 0, int board = 0)	34
MID2246_API int SN_StartRecordW (LPWSTR filename, int size, int breakonrecsize = 0, int board = 0)	34
MID2246_API int SN_SetNotifyAtRecordEnd (HWND hNotifyApp, UINT mNotifyMessage, int board = 0)	35
MID2246_API int SN_SetNotifyAtRecordEndW (HWND hNotifyApp, UINT mNotifyMessage, int board = 0)	35
MID2246_API int SN_PauseRecord (int board = 0)	36
MID2246_API int SN_ResumeRecord (int board = 0)	36
MID2246_API int SN_StopRecord (int board = 0)	36
MID2246_API int SN_SetRecordMode (MID2246_REC recMode, int board = 0)	36
MID2246_API int SN_SetRecordVideoStream (BOOL bRecVid, int board = 0)	36
MID2246_API int SN_GetRecordVideoStream (BOOL * bRecVid, int board = 0)	37
MID2246_API int SN_SetRecordAudioStream (BOOL bRecAud, int board = 0)	37
MID2246_API int SN_GetRecordAudioStream (BOOL * bRecAud, int board = 0)	37
MID2246_API int SN_SetAudioEncode (BOOL bEncAud, int board = 0)	37
MID2246_API int SN_GetAudioEncode (BOOL * bEncAud, int board = 0)	37
MID2246_API int SN_SetRenderVideo (BOOL bDisplayVideo, int board = 0)	38
MID2246_API int SN_GetRenderVideo (BOOL * bDisplayVideo, int board = 0)	38
MID2246_API int SN_SetAspectRatio (MID2246_ASPECT_MODE mode, int board = 0)	38
MID2246_API int SN_GetAspectRatio (MID2246_ASPECT_MODE * mode, int board = 0)	38
MID2246_API int SN_GetTimeLeft (int bitrate, int board)	38
Function Documentation - Playback functions	39
MID2246_API int SN_PlaybackVideo (char * filename_plus_path, int board = 0)	39
MID2246_API int SN_PlaybackVideoW (LPWSTR filename_plus_path, int board = 0)	39
MID2246_API int SN_StopPlayback (int board = 0)	39
MID2246_API int SN_PausePlayback (BOOL bPause, int board = 0)	39
MID2246_API int SN_SetNotifyDuringFilePlay (HWND hNotifyApp, UINT mNotifyMessage, int board = 0)	39
MID2246_API int SN_PlaybackSetRate (double drate, int board = 0)	40
MID2246_API int SN_PlaybackSetSeekPosition (int percent, int range, int board = 0)	40
MID2246_API int SN_PlaybackGetSeekPosition (int range, int board = 0)	40
MID2246_API int SN_DisplaySnapshot (char * filename, int time, int board = 0)	40
MID2246_API int SN_DisplaySnapshotW (LPWSTR filename, int time, int board = 0)	41
MID2246_API int SN_StopDisplaySnapshot (int board = 0)	41
Function Documentation - General configuration functions	41
MID2246_API int SN_SetMute (BOOL bMute, int board = 0)	41
MID2246_API int SN_GetMute (BOOL * bMute, int board = 0)	41
Function Documentation - Image capture functions	42

MID2246_API int SN_SetMergeMethod (MID2246_FIELDALG merge_method, mergefunc_t * custom_alg, int board).....	42
MID2246_API int SN_SnapshotToFile (char * szFilename, int filetype, int freetime, int wait, int qual, int CustMergeSize, int board).....	42
MID2246_API int SN_SnapshotToFileW (LPWSTR pFile, int filetype, int freetime, int wait, int qual, int CustMergeSize, int board).....	42
MID2246_API int SN_SnapshotToMem (BYTE * image, int size, int freetime, int wait, int board).....	43
MID2246_API int SN_SnapshotRaw (BYTE * image, int size, image_raw_t * pImage, int freetime, int wait, int board).....	43
MID2246_API int SN_FreezeFrame (int bFreeze, int board = 0).....	44
Function Documentation - Overlay functions.....	44
MID2246_API int SN_OverlayText (int xpos, int ypos, overlay_text_t * pOvlText, int regionmask, int board).....	44
MID2246_API int SN_OverlayTextIdx (int AtIndex, int xpos, int ypos, overlay_text_t * pOvlText, int regionmask, int board).....	44
MID2246_API int SN_OverlayImage (int xpos, int ypos, char * imageFile, int regionmask, int board).....	45
MID2246_API int SN_OverlayImageIdx (int AtIndex, int xpos, int ypos, char * imageFile, int regionmask, int board).....	45
MID2246_API int SetOverlayImageRaw (BYTE * image, int xPos, int yPos, int xSize, int ySize, int AtIndex, int AtIndex, int xpos, int ypos, int regionmask, int board).....	45
MID2246_API int SN_OverlayQuad (int WinIndex, char * imageFile, int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4, int regionmask, int board).....	46
MID2246_API int SN_CopyBmpToOverlayZero (char * ImageFile, char * ImageRGBptr, int xPos, int yPos, int xSize, int ySize, int Backgnd_ForegndN, int board).....	46
MID2246_API int SN_GetOverlayIdx (int WinIndex, int * Type, int * Region, int * Group, int * Xpos, int * Ypos, char ** Value, int board).....	46
MID2246_API int SN_OverlayBackgroundColor (int red, int green, int blue, int board).....	47
MID2246_API int SN_MoveOverlay (int WinIndex, int NewX, int NewY, int board).....	47
MID2246_API int SN_OverlayDelXY (int xpos, int ypos, int board).....	47
MID2246_API int SN_OverlayDel (int WinIndex, int board).....	47
MID2246_API int SN_UpdateOverlay (int board).....	47
MID2246_API int SN_ClearOverlay (int board).....	48
MID2246_API int SN_ClearOverlayRegion (int regionmask, int board).....	48
MID2246_API int SN_SetOverlayRegion (int regionmask, int board).....	48
MID2246_API int SN_SetOverlayMode (int mode, int board).....	48
Function Documentation - General Purpose IO.....	48
MID2246_API int SN_SetGPOOutput (int states, int board).....	48
MID2246_API int SN_GetGPOOutput (int * states, int board).....	48
MID2246_API int SN_GetGPIInput (int * states, int board).....	49
Function Documentation - Other.....	49
MID2246_API int SN_BootC (int board).....	49
MID2246_API int SN_BootD (int board).....	49
Sample code.....	50
Setting Custom Merge Method for interlace reconstruction.....	50

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
mid2246_vb.dll – dll library for Visual Basic
mid2246_vb.lib – dll library for Visual Basic
ovlgen.dll – overlay helper library
alleg42.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
Smartdump.ax – free AMM filter
Wavdest.ax – DXSDK sample filter to write .wav audio clips
M2TSMuxer.ax
M2TSMuxerR3.ax
mp2lame.ax
SRAYBridge.ax
sraywrite.ax
DSKernel2.dll
LMMpg1Mx2.dll
LMMpg2Mx2.dll
ffmp2encoder.ax

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

```
#include <mid2246func.h>
```

Public Attributes

- 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
-

Member Data Documentation

BYTE* image_raw_t::f0

BYTE* image_raw_t::f1

int image_raw_t::size0

int image_raw_t::size1

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

- C:/quartet/mid-2246/code/mid2246func.h

MIDSRAYSTATUS Struct Reference

```
#include <mid2246func.h>
```

Public Attributes

- ULONG **iFilesize**
 - char **szFilepath** [MAX_PATH]
 - BOOL **bIsRecording**
 - BOOL **bIsPlaying**
 - char **szReserved** [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

Member Data Documentation

BOOL MIDSRAYSTATUS::bIsPlaying

BOOL MIDSRAYSTATUS::bIsRecording

ULONG MIDSRAYSTATUS::iFilesize

char MIDSRAYSTATUS::szFilepath[MAX_PATH]

char MIDSRAYSTATUS::szReserved[MAX_PATH]

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

- C:/quartet/mid-2246/code/mid2246func.h

MIDSRAYSTATUS_UNICODE Struct Reference

```
#include <mid2246func.h>
```

Public Attributes

- ULONG **iFileSize**
 - WCHAR **szFilepath** [MAX_PATH]
 - BOOL **bIsRecording**
 - BOOL **bIsPlaying**
 - char **szReserved** [MAX_PATH]
-

Member Data Documentation

BOOL MIDSRAYSTATUS_UNICODE::bIsPlaying

BOOL MIDSRAYSTATUS_UNICODE::bIsRecording

ULONG MIDSRAYSTATUS_UNICODE::iFileSize

WCHAR MIDSRAYSTATUS_UNICODE::szFilepath[MAX_PATH]

char MIDSRAYSTATUS_UNICODE::szReserved[MAX_PATH]

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

- C:/quartet/mid-2246/code/mid2246func.h

MPEG_CONFIG Struct Reference

```
#include <mid2246const.h>
```

Public Attributes

- 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**)

Member Data Documentation

int MPEG_CONFIG::bitrate

int MPEG_CONFIG::framerate

int MPEG_CONFIG::max_vbr_bitrate

int MPEG_CONFIG::mpeg

int MPEG_CONFIG::vbr

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

- C:/quartet/mid-2246/code/**mid2246const.h**

overlay_text_t Struct Reference

```
#include <mid2246func.h>
```

Public Attributes

- 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)

Member Data Documentation

int overlay_text_t::backcolor

char overlay_text_t::fontfile[256]

int overlay_text_t::fontsize

char overlay_text_t::reserved[10]

char overlay_text_t::text[256]

int overlay_text_t::txtcolor

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

- C:/quartet/mid-2246/code/**mid2246func.h**

File Documentation

mid2246const.h File Reference

Classes

- `struct MPEG_CONFIG`

Defines

- `#define MAX_DEVICES 30`
- `#define MAX_TEXTSIZE 256`

Error constants

Defines

- `#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_ERR_INVALID_DVIPARAM -16`
- `#define MID2246_WARN_NOPROGSDI_JITTER -17`
- `#define MID2246_ERR_DV_COM_ERROR -18`
- `#define MID2246_ERR_STRLEN -19`

Snapshot error constants

Defines

- `#define MID2246_SNAP_ERR_UNSPECIFIED -1`
- `#define MID2246_SNAP_ERR_BUFSIZE -2`
- `#define MID2246_SNAP_ERR_BIST -3`
- `#define MID2246_SNAP_ERR_SYNCH -4`
- `#define MID2246_SNAP_ERR_CRITSEC -5`
- `#define MID2246_SNAP_ERR_PARTBUF -6`

- #define MID2246_SNAP_ERR_DVBUF -7
- #define MID2246_SNAP_ERR_DVGRAB -8
- #define MID2246_SNAP_ERR_USERERR -9

Frame rate Constants.

Note: for PAL, only the default(25fps) is available

Defines

- #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

Maximum count of MID2246_SOURCE_xxx enum's

Defines

- #define MID2246_SOURCE_LASTSDI *MID2246_SOURCE_HDSDI720P2*

Video levels

Defines

- #define MID2246_LEVEL_CONTRAST 1
- #define MID2246_LEVEL_BRIGHTNESS 2
- #define MID2246_LEVEL_SATURATION 4
- #define MID2246_LEVEL_HUE 8
- #define MID2246_LEVEL_SATURATION_CR 4
- #define MID2246_LEVEL_SATURATION_CB 8

Snapshot and Overlay bitmasks for file types

Defines

- #define MID2246_FILE_JPEG 1
- #define MID2246_FILE_BMP 2
- #define MID2246_FILE_PPM 4

Caption region display mask

Defines

- #define MID2246_REGION_MONITOR 1
- #define MID2246_REGION_MPEG 2
- #define MID2246_REGION_STILL 4

Revision moniker for SDK backward capability

Defines

- #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_HDSI = 6, MID2246_SOURCE_HDSI2 = 7, MID2246_SOURCE_HDSI720P = 8, MID2246_SOURCE_HDSI720P2 = 9, MID2246_SOURCE_NUM = 10 }
 - enum MID2246_AUDIO_INPUT { MID2246_AUDIO_LINE, MID2246_AUDIO_MIC, MID2246_AUDIO_EMB_AUD, MID2246_AUDIO_EMB_TST }
 - enum MID2246_AUDIO_OUTPUT { MID2246_OAUDIO_EMB_AUD, MID2246_OAUDIO_LINE_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_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_ERR_INVALID_DVIPARAM -16
#define MID2246_WARN_NOPROGSDI_JITTER -17
#define MID2246_ERR_DV_COM_ERROR -18
#define MID2246_ERR_STRLLEN -19
#define MID2246_SNAP_ERR_UNSPECIFIED -1
#define MID2246_SNAP_ERR_BUFSIZE -2
#define MID2246_SNAP_ERR_BIST -3
#define MID2246_SNAP_ERR_SYNCH -4
#define MID2246_SNAP_ERR_CRITSEC -5
#define MID2246_SNAP_ERR_PARTBUF -6
#define MID2246_SNAP_ERR_DVBUF -7
#define MID2246_SNAP_ERR_DVGRAB -8
#define MID2246_SNAP_ERR_USERERR -9
#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_SOURCE_LASTSDI MID2246_SOURCE_HDSI720P2
#define MID2246_LEVEL_CONTRAST 1
#define MID2246_LEVEL_BRIGHTNESS 2
#define MID2246_LEVEL_SATURATION 4
#define MID2246_LEVEL_HUE 8
```

```
#define MID2246_LEVEL_SATURATION_CR 4
#define MID2246_LEVEL_SATURATION_CB 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
```

Enumeration Type Documentation

enum MID2246_VIDSYS

Video systems. PAL or NTSC

Enumerator:

MID2246_VIDSYS_PAL
MID2246_VIDSYS_NTSC

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-SDI input. High definition 1080i at 60Hz frame rate (1.485 Gbps)
MID2246_SOURCE_HDSDI720P 720p, 1.485/1.001Gbps, Vclk=74.1758 Mhz (59.94Hz frame rate)
MID2246_SOURCE_HDSDI720P2 720p, 1.485 Gbps, Vclk=74.25 Mhz (60Hz 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_HDSDI720P
MID2246_SOURCE_HDSDI720P2
MID2246_SOURCE_NUM

enum MID2246_AUDIO_INPUT

MID2246_AUDIO_INPUT sets the audio input channel,

Parameters:

MID2246_AUDIO_LINE line in
MID2246_AUDIO_MIC microphone
MID2246_AUDIO_EMB_AUD Embedded Audio from the input video source selected.

Enumerator:

MID2246_AUDIO_LINE
MID2246_AUDIO_MIC
MID2246_AUDIO_EMB_AUD
MID2246_AUDIO_EMB_TST

enum MID2246_AUDIO_OUTPUT

MID2246_AUDIO_OUTPUT sets the audio source feeding the SDI-OUT embedded audio.

Parameters:

MID2246_AUDIO_LINEMIC Audio from the Line or Microphone inputs. (selected by *SN_SetAudioInput()*)
MID2246_AUDIO_EMB_AUD Embedded Audio from the input video source selected.

Enumerator:

MID2246_OAUDIO_EMB_AUD
MID2246_OAUDIO_LINE_MIC

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_DVIRES

DVI output resolution These correspond to the [RESx] lines in \windows\dviparam.ini where x is called out by the MID2246_DVIRES_xxx parameter.

Parameters:

MID2246_DVIRES_W1280H1024 Scale to/and set DVI output to 1280 x 1024 resolution.

MID2246_DVIRES_W1600H1200 Scale to/and set DVI output to 1600 x 1200 resolution.

MID2246_DVIRES_W720H480 Scale to/and set DVI output to 720 x 480 resolution.

MID2246_DVIRES_W720H576 Scale to/and set DVI output to 720 x 576 resolution.

MID2246_DVIRES_W1920H1080 Scale to/and set DVI output to 1920 x 1080 resolution.

Enumerator:

MID2246_DVIRES_W1280H1024

MID2246_DVIRES_W1600H1200

MID2246_DVIRES_W720H480

MID2246_DVIRES_W720H576

MID2246_DVIRES_W1920H1080

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_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_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

Enumerator:

MID2246_RC_OUTSPACE

MID2246_RC_RECTIME
MID2246_RC_MANUALSTOP
MID2246_RC_RECCLIPSIZE
MID2246_RC_RECCLIPSIZEANDSTOP

mid2246func.h File Reference

```
#include "windows.h"  
#include "mid2246const.h"
```

Data Structures

- struct **MIDSRAYSTATUS**
- struct **MIDSRAYSTATUS_UNICODE**
- struct **image_raw_t**
- struct **overlay_text_t**

Defines

- #define **MID2246_API** __declspec(dllimport)
- #define **MID2246_MERGEAPI** int
- #define **MAX_PATH** 260
- #define **MID2246_MAX_TEXT** 256
- #define **MID2246_MAX_FONTFILE** 256

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_GetNumBoards** (int *pBoards)
- MID2246_API int **SN_GetVersion** (char **sMid2246Version, char **sFPGA2246Version, int *pBoardId, int board)
- MID2246_API int **SN_Initialize** (HWND hwnd, HWND messageHwnd, int removeMsg, int board)
- MID2246_API int **SN_Shutdown** (int board)
- MID2246_API int **SN_UpdateClippingWindow** (HWND hwnd, int board)
- MID2246_API int **SN_SetVideoPosition** (int xpos, int ypos, int xsize, int ysize, int left_clip, int top_clip, int right_clip, int bottom_clip, int board)
- MID2246_API int **SN_SetSource** (**MID2246_SOURCE** input, BOOL bTest, int board)
- MID2246_API int **SN_GetSource** (**MID2246_SOURCE** *pSource, int board)
- MID2246_API int **SN_SetVidSys** (**MID2246_VIDSYS** vidsys, int board)
- MID2246_API int **SN_GetVidSys** (int board)
- MID2246_API int **SN_SetDvi** (**MID2246_DVIRES** res, BOOL bColorBars, int DviCropLeft, int DviCropTop, int DviCropRight, int DviCropBottom, int board)
- MID2246_API int **SN_SetMpegCrop** (int MpegCropLeft, int MpegCropTop, int MpegCropRight, int MpegCropBottom, int board)
- MID2246_API int **SN_SetMpegConfig** (**MPEG_CONFIG** *pConfig, int board)
- MID2246_API int **SN_GetMpegConfig** (**MPEG_CONFIG** *pConfig, int board)
- MID2246_API int **SN_SetLevels** (int param, unsigned char value, int board)
- MID2246_API int **SN_SetAudioInput** (**MID2246_AUDIO_INPUT** audio, int micBoost, int board)
- MID2246_API int **SN_SetAudioOutput** (**MID2246_AUDIO_OUTPUT** audio, int board)
- MID2246_API int **SN_GetStatus** (**MIDSRAYSTATUS** *pStats, int board)
- MID2246_API int **SN_GetFrameSync** (int *sync, int board)

- MID2246_API int **SN_IsValidVid** (**MID2246_SOURCE** checksource, int board)
- MID2246_API int **SN_IsValidSdi** (**MID2246_SOURCE** checksource, int board)
- MID2246_API int **SN_IsValidDv** (int board)
- MID2246_API int **SN_EnumerateInputs** (int *pStatus, int *pVidSys, int board)
- MID2246_API int **SN_GetVidStatus** (int board)
- MID2246_API int **SN_GetVidStatusNoBIST** (int board)
- MID2246_API int **SN_TestDeviceRemoval** (int board)
- MID2246_API int **SN_Repaint** (HDC hdc)
- MID2246_API int **SN_OnDisplayModeChanged** (int board)
- MID2246_API int **SN_StartStream** (int board=0)
- MID2246_API int **SN_StopStream** (int board=0)
- MID2246_API int **SN_LowLatency** (BOOL bON, int board)
- MID2246_API int **SN_StartRecord** (char *filename, int size, int breakonrecsize=0, int board=0)
- MID2246_API int **SN_StartRecordW** (LPWSTR 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_SetNotifyAtRecordEndW** (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_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_GetTimeLeft** (int bitrate, int board)
- MID2246_API int **SN_PlaybackVideo** (char *filename_plus_path, int board=0)
- MID2246_API int **SN_PlaybackVideoW** (LPWSTR 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_DisplaySnapshotW** (LPWSTR 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_SetMergeMethod** (**MID2246_FIELDALG** merge_method, **mergefunc_t** *custom_alg, int board)

- MID2246_API int **SN_SnapshotToFile** (char *szFilename, int filetype, int freezetime, int wait, int qual, int CustMergeSize, int board)
- MID2246_API int **SN_SnapshotToFileW** (LPWSTR pFile, int filetype, int freezetime, int wait, int qual, int CustMergeSize, int board)
- MID2246_API int **SN_SnapshotToMem** (BYTE *image, int size, int freezetime, int wait, int board)
- MID2246_API int **SN_SnapshotRaw** (BYTE *image, int size, **image_raw_t** *pImage, int freezetime, int wait, int board)
- MID2246_API int **SN_FreezeFrame** (int bFreeze, int board=0)
- MID2246_API int **SN_OverlayText** (int xpos, int ypos, **overlay_text_t** *pOvlText, int regionmask, int board)
- MID2246_API int **SN_OverlayTextIdx** (int AtIndex, int xpos, int ypos, **overlay_text_t** *pOvlText, int regionmask, int board)
- MID2246_API int **SN_OverlayImage** (int xpos, int ypos, char *imageFile, int regionmask, int board)
- MID2246_API int **SN_OverlayImageIdx** (int AtIndex, int xpos, int ypos, char *imageFile, int regionmask, int board)
- MID2246_API int **SN_OverlayImageRaw** (BYTE *image, int xPos, int yPos, int xSize, int ySize, int AtIndex, int regionmask, int board)
- MID2246_API int **SN_OverlayQuad** (int WinIndex, char *imageFile, int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4, int regionmask, int board)
- MID2246_API int **SN_CopyBmpToOverlayZero** (char *ImageFile, char *ImageRGBptr, int xPos, int yPos, int xSize, int ySize, int Backgnd_ForegndN, int board)
- MID2246_API int **SN_GetOverlayIdx** (int WinIndex, int *Type, int *Region, int *Group, int *Xpos, int *Ypos, char **Value, int board)
- MID2246_API int **SN_OverlayBackgroundColor** (int red, int green, int blue, int board)
- MID2246_API int **SN_MoveOverlay** (int WinIndex, int NewX, int NewY, int board)
- MID2246_API int **SN_OverlayDelXY** (int xpos, int ypos, int board)
- MID2246_API int **SN_OverlayDel** (int WinIndex, int board)
- MID2246_API int **SN_UpdateOverlay** (int board)
- MID2246_API int **SN_ClearOverlay** (int board)
- MID2246_API int **SN_ClearOverlayRegion** (int regionmask, int board)
- MID2246_API int **SN_SetOverlayRegion** (int regionmask, int board)
- MID2246_API int **SN_SetOverlayMode** (int mode, int board)
- MID2246_API int **SN_SetGPOutput** (int states, int board)
- MID2246_API int **SN_GetGPOutput** (int *states, int board)
- MID2246_API int **SN_GetGPInput** (int *states, int board)
- MID2246_API int **SN_BootC** (int board)
- MID2246_API int **SN_BootD** (int board)

Define Documentation

#define MAX_PATH 260

#define MID2246_API __declspec(dllimport)

MID2246 API. API functions for the 2246 board.

Author:

Sensaray Company Inc.

```
#define MID2246_MERGEAPI int
```

```
#define MAX_PATH 260
```

```
#define MID2246_MAX_TEXT 256
```

Overlay text structure

Parameters:

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

```
#define MID2246_MAX_FONTFILE 256
```

Typedef Documentation

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

Defines a callback function prototype that is used set by **SN_SetMergeMethod()** If field *custom_alg* is set, *mergefunc_t* defines a callbacks raw fields to allow the user to merge the interlaced fields with a custom algorithm.

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 - Configuration and Setup functions

```
MID2246_API int SN_GetNumBoards (int * pBoards)
```

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_GetVersion (char ** *sMid2246Version*, char ** *sFPGA2246Version*, int * *pBoardId*, int *board*)

Get middleware version number and FPGA version number.

Parameters:

sMid2246Version String pointer set to mid2246 middleware version.

sFPGA2246Version String pointer set to 2246 FPGA version.

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 ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_Initialize (HWND *hwnd*, HWND *messageHwnd*, int *removeMsg*, int *board*)

Opens and Initializes the middleware.

Parameters:

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

board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_Shutdown (int *board*)

Cleans up and shuts down the API gracefully

Parameters:

board A -1 indicates all boards will be closed.

Returns:

0 on success

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

Update the clipping window for the video

Parameters:

hwnd pointer to video display window

board ID if multiple boards installed.

Returns:

0 on success

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

Set the video position in the clipping window

Parameters:

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
board ID if multiple boards installed.

Returns:

0 on success

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

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

Parameters:

input is the enumerated input MID2246_SOURCE enum(composite, svideo, DV...) (see mid22xxconst.h)
bTest whether to include colorbars or not
board ID if multiple boards installed.

Returns:

0 on success

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

Gets the current video source

Parameters:

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

Returns:

0 on success

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

Set the current video system

Parameters:

vidsys is the video system type
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_GetVidSys (int *board*)

Get the current video system

Parameters:

board ID if multiple boards installed.

Returns:

the video system type on success

MID2246_API int SN_SetDvi (MID2246_DVIRES *res*, BOOL *bColorBars*, int *DviCropLeft*, int *DviCropTop*, int *DviCropRight*, int *DviCropBottom*, int *board*)

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

Parameters:

res : Enumerated resolution (MID2246_DVIRES)
bColorBars,: Turn Colorbars on or off
DviCropLeft clip left of scaled image by this value then stretch to fit.
DviCropTop clip top of scaled image by this value then stretch to fit.
DviCropRight clip right of scaled image by this value then stretch to fit.
DviCropBottom clip bottom of scaled image by this value then stretch to fit.
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_SetMpegCrop (int *MpegCropLeft*, int *MpegCropTop*, int *MpegCropRight*, int *MpegCropBottom*, int *board*)

Set the MPEG Hardware Cropping

Parameters:

MpegCropLeft clip left of scaled image by this value then stretch to fit.
MpegCropTop clip top of scaled image by this value then stretch to fit.
MpegCropRight clip right of scaled image by this value then stretch to fit.
MpegCropBottom clip bottom of scaled image by this value then stretch to fit.
board ID if multiple boards installed.

Returns:

0 on success

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

Sets the mpeg configuration parameters(see MPEG_CONFIG above).

Parameters:

pConfig pointer to MPEG_CONFIG structure
board ID if multiple boards installed.

Returns:

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

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

Retrieves the current mpeg configuration parameters

Parameters:

pConfig pointer MIDSRAYSTATUS Status structure.
board ID if multiple boards installed.

Returns:

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

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

Set the brightness, contrast and saturation

Parameters:

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--minimum, 128 nominal, 255 max)
board ID if multiple boards installed.

Returns:

0 on success

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

Set audio input channel (default is MID2246_AUDIO_LINE)

Parameters:

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

Returns:

0 on success

MID2246_API int SN_SetAudioOutput (MID2246_AUDIO_OUTPUT *audio*, int *board*)

Set audio input channel sent to SDI-OUT embedded audio (default is MID2246_OAUDIO_EMB_AUD)

Parameters:

<i>audio</i>	: MID2246_AUDIO_OUTPUT Embedded Audio on selected input or line/mic-in: MID2246_AUDIO_EMB_AUD, MID2246_AUDIO_LINE_MIC.
<i>board</i>	ID if multiple boards installed.

Returns:

0 on success

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

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

Parameters:

pStats pointer to
board ID if multiple boards installed.

Returns:

0 on success, negative otherwise (see errors in mid22xxconst.h)

MID2246_API int SN_GetFrameSync (int * *sync*, int *board*)

Check whether a video sync pulse has occurred.

Parameters:

sync pointer to an integer that receives '1' if a video sync has occurred since the last call.
board ID if multiple boards installed.

Returns:

Same value that is put in states variable location.

MID2246_API int SN_IsValidVid (MID2246_SOURCE *checksource*, int *board*)

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:

checksource Set to MID2246_SOURCE_COMPOSITE_0, MID2246_SOURCE_COMPOSITE_1, MID2246_SOURCE_SVIDEO_0 or MID2246_SOURCE_SVIDEO_1 to check if one specific video source is valid. Any other value will return true if any of the above sources are valid.
board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_IsValidSdi (MID2246_SOURCE *checksource*, int *board*)

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

Parameters:

checksource Set to MID2246_SOURCE_SDI, MID2246_SOURCE_HDSDI, MID2246_SOURCE_HDSDI2, MID2246_SOURCE_HDSDI720P, or MID2246_SOURCE_HDSDI720P2 to check if one specific video source is valid. Any other value will return true if any of the above sources are valid.
board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_IsValidDv (int *board*)

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

Parameters:

board ID if multiple boards installed.

Returns:

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

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

Enumerate connected inputs. Note: This function 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:

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).
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_GetVidStatus (int *board*)

GetVidStatus Checks and waits for status of the Built In Self Test (BIST) of the DDR memory and then returns current lock status of the current input.

Parameters:

board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_GetVidStatusNoBIST (int *board*)

GetVidStatus Checks and waits for status of the Built In Self Test (BIST) of the DDR memory and then returns current lock status of the current input.

Parameters:

board ID if multiple boards installed.

Returns:

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

Function Documentation - Callback functions from application

MID2246_API int SN_TestDeviceRemoval (int *board*)

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 if multiple boards installed.

Returns:

0 if not removed, 1 if removed

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:

0 on success

MID2246_API int SN_OnDisplayModeChanged (int *board*)

Notifies 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

Returns:

0 on success

Function Documentation - Recording functions

MID2246_API int SN_StartStream (int *board* = 0)

Starts the streaming of video and audio

Parameters:

board ID 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 if multiple boards installed.

Returns:

0 on success

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

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 (with **SN_StartStream()**) if desired.

Parameters:

bON Enable or disable low latency mode.
board ID if multiple boards installed.

Returns:

0 on success

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:

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.
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_StartRecordW (LPWSTR *filename*, int *size*, int *breakonrecsize* = 0, int *board* = 0)

Unicode version of **SN_StartRecord()** Starts recording to file using filter graph previously set up. Note that function **SN_StartRecord()** will try to detect if a unicode string passed to it, but this is not guaranteed (see Microsoft MSDN notes on IsUnicodeText.) If using Unicode filenames, please use the proper function.

Parameters:

filename Full filename plus path and no extension in unicode (Windows WideChar)
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

breakonresize flag indicating whether to, 1=end recording when filesize reaches size, or 0=continue recording to new file.
board ID if multiple boards installed.

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. Sets the call back filename as Ascii

Parameters:

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 ASCII name of the recorded file that was just closed.

board ID if multiple boards installed.

Returns:

0 on success

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

Enables or Disables sending of message indicating that recording has stopped. Sets the call back filename as Unicode (Windows Wide Characters WCHAR)

Parameters:

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 wide character (WCHAR) name of the recorded file that was just closed.

board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_PauseRecord (int *board* = 0)

Pause recording.

Parameters:

board ID if multiple boards installed.

Returns:

HRESULT = S_OK on success

MID2246_API int SN_ResumeRecord (int *board* = 0)

Resume recording.

Parameters:

board ID if multiple boards installed.

Returns:

HRESULT = S_OK on success

MID2246_API int SN_StopRecord (int *board* = 0)

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

Parameters:

board ID if multiple boards installed.

Returns:

HRESULT = S_OK 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:

recMode MID2246_REC setting (see **mid2246const.h**)

board ID if multiple boards installed.

Returns:

0 on success

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

Turn on/off video recording NOTE: this is different from **SN_StartRecord()**, etc. **SN_SetRecordVideoStream()** and **SN_SetRecordAudioStream()** are to allow configuration for either Video or Audio Elementary stream while recording. If RecordVideoStream is turned off, no video stream is sent to the recording module. If RecordAudioStream is turned off, no audio is sent to the recording module.(different from mute)

Parameters:

bRecVid whether to record video or not.

board ID if multiple boards installed.

Returns:

0 on success

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

Gets the video recording setting

Parameters:

bRecVid pointer to record video setting
board ID if multiple boards installed.

Returns:

0 on success

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. **SN_SetRecordVideoStream()** and **SN_SetRecordAudioStream()** are to allow configuration for either Video or Audio Elementary stream while recording. If RecordVideoStream is turned off, no video stream is sent to the recording module. If RecordAudioStream is turned off, no audio is sent to the recording module.(different from mute)

Parameters:

bRecAud whether to record audio or not
board ID if multiple boards installed.

Returns:

0 on success

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

Gets the audio recording setting

Parameters:

bRecAud pointer to record video setting
board ID if multiple boards installed.

Returns:

0 on success

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

Sets the audio encoder setting

Parameters:

bEncAud encode audio or not.
board ID if multiple boards installed.

Returns:

0 on success

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

Get the audio encode setting

Parameters:

bEncAud pointer to audio encode setting
board ID if multiple boards installed.

Returns:

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:

bDisplayVideo whether to display video or not
board ID if multiple boards installed.

Returns:

0 on success

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

Gets the display video setting

Parameters:

bDisplayVideo pointer to display video setting
board ID if multiple boards installed.

Returns:

0 on success

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

Set aspect ratio flag

Parameters:

mode MID2246_ASPECT_NONE--stretched, MID2246_ASPECT_CONST--maintain aspect ratio
board ID if multiple boards installed.

Returns:

0 on success, -ve on error

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

Get the aspect ratio mode

Parameters:

mode returned aspect ratio mode
board ID if multiple boards installed.

Returns:

0 on success

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

Returns estimated recording time left on disk

Parameters:

bitrate is optional bitrate(otherwise uses current bitrate)
board ID if multiple boards installed.

Returns:

time in seconds

Function Documentation - Playback functions

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:

filename_plus_path A UTF-8 (standard ASCII) zero terminated string containing path of the file to record to.
board ID if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_PlaybackVideoW (LPWSTR *filename_plus_path*, int *board* = 0)

Unicode version of **SN_PlaybackVideo()** Plays back the specified video clip in the current window Default is first.

Parameters:

filename_plus_path Full filename plus path and no extension in unicode (Windows WideChar) of the file to record to.
board ID if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_StopPlayback (int *board* = 0)

Stops video playback

Parameters:

board ID if multiple boards installed.

Returns:

0 on success

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

Pause/resume playback

Parameters:

bPause *bPause* = TRUE(pause), *bPause*=FALSE(resume)
board ID if multiple boards installed.

Returns:

0 on success

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:

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.

board ID if multiple boards installed.

Returns:

none

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

Change playback rate

Parameters:

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

Returns:

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.

Parameters:

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

Returns:

0 on success -1 otherwise

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

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

Parameters:

range defines the percentage granularity
board ID if multiple boards installed.

Returns:

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

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

Displays the snapshot in the video window

Parameters:

filename full filename plus path plus extension coded in UTF-8. (standard ASCII.)
time time to display snapshot, -1 is infinite
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_DisplaySnapshotW (LPWSTR *filename*, int *time*, int *board* = 0)

Unicode version of **SN_DisplaySnapshot()** Displays the snapshot in the video window using unicode for the filename.

Parameters:

filename Full filename plus path extension in unicode (Windows WideChar)
time Time to display snapshot, -1 is infinite
board ID 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 if multiple boards installed.

Returns:

0 on success

Function Documentation - General configuration functions

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:

bMute whether to mute audio on PC
board ID if multiple boards installed.

Returns:

0 on success

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

Gets the mute setting

Parameters:

bMute pointer to retrieved mute setting
board ID if multiple boards installed.

Returns:

0 on success

Function Documentation - Image capture functions

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

Set the interlaced field merging algorithm

Parameters:

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)
board ID if multiple boards installed.

Returns:

0 on success

**MID2246_API int SN_SnapshotToFile (char * *szFilename*, int *filetype*, int *freezetime*, int
wait, int *qual*, int *CustMergeSize*, int *board*)**

Takes a snapshot and saves to file in filename

Parameters:

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 Sets JPEG quality, (use 25-100, default 100).
CustMergeSize Size in bytes of the buffer sent to the custom image merge call back.
board ID if multiple boards installed.

Returns:

number of bytes read (image size) or MID2246_SNAP_ERR_UNSPECIFIED -1
MID2246_SNAP_ERR_BUFSIZE -2 'Size' of buffer provided is too small for snapshot.
MID2246_SNAP_ERR_BIST -3 Built in self test failed. Hardware not ready.
MID2246_SNAP_ERR_SYNCH -4 Aquisition's synchronization loop timed out.
MID2246_SNAP_ERR_CRITSEC -5 Aquisition's critical section already locked, and user doesn't
want to wait for it. MID2246_SNAP_ERR_PARTBUF -6 Didn't recieve a full frame of data.
MID2246_SNAP_ERR_DVBUF -7 Failed to get current frame buffer from DV
MID2246_SNAP_ERR_DVGRAB -8 The DV frame grabber was not setup
MID2246_SNAP_ERR_USERERR -9 Error returned from user's merge function

**MID2246_API int SN_SnapshotToFileW (LPWSTR *pFile*, int *filetype*, int *freezetime*, int *wait*,
int *qual*, int *CustMergeSize*, int *board*)**

Unicode version of SN_SnapshotToFile. Takes a snapshot and saves to file.

Parameters:

pFile Full filename plus path and no extension in unicode (Windows WideChar)
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 Sets JPEG quality, (use 25-100, default 100).
CustMergeSize Size in bytes of the buffer sent to the custom image merge call back.
board ID if multiple boards installed.

Returns:

number of bytes read (image size) or MID2246_SNAP_ERR_UNSPECIFIED -1
MID2246_SNAP_ERR_BUFSIZE -2 'Size' of buffer provided is too small for snapshot.
MID2246_SNAP_ERR_BIST -3 Built in self test failed. Hardware not ready.
MID2246_SNAP_ERR_SYNCH -4 Acquisition's synchronization loop timed out.
MID2246_SNAP_ERR_CRITSEC -5 Acquisition's critical section already locked, and user doesn't want to wait for it. MID2246_SNAP_ERR_PARTBUF -6 Didn't receive a full frame of data.
MID2246_SNAP_ERR_DVBUF -7 Failed to get current frame buffer from DV
MID2246_SNAP_ERR_DVGRAB -8 The DV frame grabber was not setup
MID2246_SNAP_ERR_USERERR -9 Error returned from user's merge function

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

Get snapshot to memory function. Retrieves processed image into memory. Image is converted to RGB (1 byte per color) and then the fields are merged using the algorithm set by **SN_SetMergeMethod()**.

Parameters:

image IN/OUT pointer to retrieved snapshot.
size is the size of the image buffer.
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
board ID if multiple boards installed.

Returns:

number of bytes read (image size) or MID2246_SNAP_ERR_UNSPECIFIED -1
MID2246_SNAP_ERR_BUFSIZE -2 'Size' of buffer provided is too small for snapshot.
MID2246_SNAP_ERR_BIST -3 Built in self test failed. Hardware not ready.
MID2246_SNAP_ERR_SYNCH -4 Acquisition's synchronization loop timed out.
MID2246_SNAP_ERR_CRITSEC -5 Acquisition's critical section already locked, and user doesn't want to wait for it. MID2246_SNAP_ERR_PARTBUF -6 Didn't receive a full frame of data.
MID2246_SNAP_ERR_DVBUF -7 Failed to get current frame buffer from DV
MID2246_SNAP_ERR_DVGRAB -8 The DV frame grabber was not setup
MID2246_SNAP_ERR_USERERR -9 Error returned from user's merge function

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

Get snapshot data from hardware 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:

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
board ID if multiple boards installed.

Returns:

number of bytes read (image size) or MID2246_SNAP_ERR_UNSPECIFIED -1
MID2246_SNAP_ERR_BUFSIZE -2 'Size' of buffer provided is too small for snapshot.
MID2246_SNAP_ERR_BIST -3 Built in self test failed. Hardware not ready.
MID2246_SNAP_ERR_SYNCH -4 Acquisition's synchronization loop timed out.

MID2246_SNAP_ERR_CRITSEC -5 Aquisition's critical section already locked, and user doesn't want to wait for it. MID2246_SNAP_ERR_PARTBUF -6 Didn't recieve a full frame of data.

MID2246_API int SN_FreezeFrame (int *bFreeze*, int *board* = 0)

Freezes and Unfreezes the output frame.

Parameters:

<i>bFreeze</i>	= 0 unfreezes the frame, <i>bFreeze</i> = 1 freezes the frame
<i>board</i>	ID if multiple boards installed.

Returns:

0 on success

Function Documentation - Overlay functions

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

Adds overlay text. If overlay text already exists at that x,y position, deletes windows 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, at index 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:

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)

board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_OverlayTextIdx (int *AtIndex*, int *xpos*, int *ypos*, overlay_text_t * *pOvlText*, int *regionmask*, int *board*)

Adds/Update 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:

AtIndex -- sub-window position to update/add text

xpos -- new X position

ypos -- new y position

pOvlText -- pointer to overlay text

regionmask -- region mask (MID2246_REGION_MONITOR, MID2246_REGION_MPEG, MID2246_REGION_STILL)
board ID if multiple boards installed.

Returns:

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

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

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

Parameters:

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
board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_OverlayImageIdx (int *AtIndex*, int *xpos*, int *ypos*, char * *imageFile*, int *regionmask*, int *board*)

Adds/Update overlay image. If overlay already exists at index WinIndex, deletes window before adding.

Parameters:

AtIndex -- sub-window position to update/add text
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
board ID if multiple boards installed.

Returns:

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

MID2246_API int SetOverlayImageRaw (BYTE * *image*, int *xPos*, int *yPos*, int *xSize*, int *ySize*, int *AtIndex*, int *AtIndex*, int *xpos*, int *ypos*, int *regionmask*, int *board*)

Adds/Update overlay image. If overlay already exists at index WinIndex, deletes window before adding.

Parameters:

image -- Pointer to memory containing 24-bit RGB data Row 0 first
xpos -- Start X position
ypos -- Start Y position
xSize -- Size of horizontal line in pixels (Each pixel is 3 bytes: LSB=Red,Green,MSB=Blue)
ySize -- Number of Lines
AtIndex -- sub-window position to update/add image
regionmask -- output region mask
board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_OverlayQuad (int *WinIndex*, char * *imageFile*, int *x1*, int *y1*, int *x2*, int *y2*, int *x3*, int *y3*, int *x4*, int *y4*, int *regionmask*, int *board*)

Use an image to texture map a 4-sided polygon. The original text or image in the overlay is lost.

Parameters:

WinIndex -- index of sub-window to delete
imageFile -- pointer to overlay image(full path) 24bit BMP or PCX files supported only
x1 x-value of coordinate 1, the upper left corner of the texture
y1 y-value of coordinate 1
x2 x-value of coordinate 2, the upper right corner of the texture
y2 y-value of coordinate 2
x3 x-value of coordinate 3, the lower right corner of the texture
y3 y-value of coordinate 3
x4 x-value of coordinate 4, the lower left corner of the texture
y4 y-value of coordinate 4
regionmask -- output region mask
board ID if multiple boards installed.

Returns:

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

MID2246_API int SN_CopyBmpToOverlayZero (char * *ImageFile*, char * *ImageRGBptr*, int *xPos*, int *yPos*, int *xSize*, int *ySize*, int *Backgnd_ForegndN*, int *board*)

Copy given image to region within overlay zero. Assumes Overlay zero is loaded into current memory Assumes Overlay zero has a width that is divisible by 8 Assumes all other overlays (1-7) are contained within the boundaries of overlay zero

(It is assumed that the Overlay at Index-0 will be the same size as the input resolution)

Parameters:

ImageFile - Full name/path to image file. 24-bit bmp and pcx files are supported. If NULL, use Raw 24-bit bitmap pointer Image
ImageRGBptr - Pointer to memory containing 24-bit RGB data Row 0 first
xPos - Horizontal Destination pixel position from left of Bitmap-0 to copy to.
yPos - Vertical Destination pixel position from top of Bitmap-0 to copy to.
xSize - Size of horizontal line in pixels (NOTE: Each pixel is 3 bytes: LSB=Red,Green,MSB=Blue) Used with Image pointer only.
ySize - Number of Lines. Used with Image pointer only.
Backgnd_ForegndN - Copy directly to currently displayed overlay image when 0. Otherwise copy to overlay's double buffered background.
board ID if multiple boards installed.
Returns 0 on success, -1 on failure.

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

Get basic parameters of an overlay index

Parameters:

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
board ID if multiple boards installed.

Returns:

0 on success

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

Set the color used for the background regions

Parameters:

red -- 0 min, 255 max red
green -- 0 min, 255 max green
blue -- 0 min, 255 max blue
board ID if multiple boards installed.

Returns:

0 on success

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

Change position of an overlay sub-window Does not reload or download the new image

Parameters:

WinIndex -- index of sub-window to move
NewX -- new X position
NewY -- new y position
board ID if multiple boards installed.

Returns:

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

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

Delete an overlay at x,y location. Does not reload or download the new image

Parameters:

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

Returns:

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*)

Delete an overlay at a given index. Does not reload or download the new image

Parameters:

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

Returns:

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

MID2246_API int SN_UpdateOverlay (int *board*)

Refresh or update the overlay to the hardware

Parameters:

board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_ClearOverlay (int board)

Clear all overlays

Parameters:

board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_ClearOverlayRegion (int regionmask, int board)

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

Parameters:

regionmask output region mask
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_SetOverlayRegion (int regionmask, int board)

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

Parameters:

regionmask output region mask
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_SetOverlayMode (int mode, int board)

Set overlay mode. No longer has any effect. All current SDK/FPGA releases support full screen (at source input resolution) double buffered text and graphic overlays with 16-bit color.

Function Documentation - General Purpose IO

MID2246_API int SN_SetGPOOutput (int states, int board)

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

Parameters:

states integer value to drive the general purpose outputs with.
board ID if multiple boards installed.

Returns:

0 on success

MID2246_API int SN_GetGPOOutput (int * states, int board)

Reads the value that was previously written to the general purpose output pins via SN_SetGPOOutput

Parameters:

states pointer to an integer that receives the value in the register driving the outputs.
board ID if multiple boards installed.

Returns:

Same value that is put in *states* variable location.

MID2246_API int SN_GetGPIInput (int * *states*, int *board*)

Get the value present on the general purpose input pins.

Parameters:

states pointer to an integer that receives the input value.
board ID if multiple boards installed.

Returns:

Same value that is put in *states* variable location.

Function Documentation - Other

MID2246_API int SN_BootC (int *board*)

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 if multiple boards installed.

Returns:

HRESULT

MID2246_API int SN_BootD (int *board*)

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

Parameters:

board ID if multiple boards installed.

Returns:

HRESULT

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()`