GRANDSTREAM NETWORKS

Grandstream Innovative IP Voice & Video

# GSNET API
for Grandstream DVS/IP camera

Document Number: XXX
Version: XXX
Owner: Grandstream Networks, Inc.
Date: 08/20/2009

Table of Connects

Introduction

This document gives Description of the GSNET API as well as data structures.

GSNET Client API Function Description

1.1 GSNET_Startup

Purpose: Initiate the use of GSNETClient.dll

Function:

BOOL GSNET_Startup (UINT ulMessage, HWND hWnd, void (WINAPI *messagecallback)(LONG handle, int wParam, int lParam, void *context)=NULL, void* context=NULL);

Parameters:

ulMessage //reserved

hWnd //reserved

*messagecallback //Global message callback Function, it will return the callback of link state,record state, alarm outpur, and motion detect message.

*context //reserved

Return values:

TRUE //success

FALSE //failure

1.2 GSNET_Cleanup

Purpose: Terminate the use of the GSNETClient.dll

Function: GSNET_Cleanup()

1.3 GSNET_SetWaitTime

Purpose: Set the duration and time of reconnection, and duration of RTSP connection

Function: BOOL GSNET_SetWaitTime(int WaitTime=5, int TryNum=3, int TryInterval=10)

Parameter

WaitTime //timeout for RTSP Connection (second)

TryNum //times for try ing to connect, if value is -1, will connect always

TryInterval //duration for reconnect (second)

Return value

TRUE //successful

FALSE //failure

1.4 GSNET_ClientStart

Purpose: Allocate resource for device

Function: LONG GSNET_ClientStart(CHANNEL_CLIENTINFO *m_pChaninfo)

Description

This function allocates resources for destination device only, so it's a un-block Function, and device link state will reflect in messagecallback Function(in Function 1.1)

Parameter

m_pChaninfo // see CHANNEL_CLIENTINFO structure (2.2)

Return value

0	//failure
other	//successful

1.5 GSNET_ClientStop

Purpose: Stop device and release resource

Function: BOOL GSNET_ClientStop(LONG hHandle)

Description: Stop device and release resource, It's a block Function.

Parameter

hHandle //Return value of Function 1.4

Return value

TRUE	//successful
FALSE	//failure

1.6 GSNET_ClientStartView

Purpose: Show video

Function: BOOL GSNET_ClientStartView(LONG hHandle)

Description: If you call Function 1.7 to hide video, then you can call the Function show again

Parameter

hHandle //Return value of Function 1.4

Return value

TRUE	//successful
FALSE	//failure

1.7 GSNET_ClientStopView

Purpose: Hide video

Function: BOOL GSNET_ClientStopView(LONG hHandle)

Description: If you no need to view video, you can call the Function to Hideo video

Parameter

hHandle //Return value of Function 1.4

Return value

TRUE	//successful
FALSE	//failure

1.8 GSNET_ClientSetWnd

Purpose: Display on a designated window

Function: BOOL GSNET_ClientSetWnd (LONG hHandle, HWND hWnd)

Description: Display video on a designated window

Parameter

hHandle //Return value of Function 1.4

hWnd //NULL will hide video

Return value

TRUE //successful

FALSE //failure

1.9 GSNET_ClientShowcallback

Purpose: Redraw window DC

Function:

BOOL GSNET_ClientShowcallback(LONG hHandle, void(WINAPI *ShowCallBack)(BYTE *m_y, BYTE *m_u, BYTE *m_v, int stridey, int strideuv, int width, int height, void *context), void *context)

Parameter

hHandle //Return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

1.10 GSNET_GetClientState

Purpose: Get connection state

Function: LONG GSNET_GetClientState(LONG hHandle)

Parameter

hHandle //Return value of Function 1.4

Return value

See messages definition 3.1

1.11 GSNET_GetBitStreamInfo

Purpose: Get video stream Infomation

Function

BOOL GSNET_GetBitStreamInfo(LONG hHandle, ULONG *pBitRate, ULONG *pFrameRate)

Description: Get video stream Infomation

Parameter

hHandle //Return value of Function 1.4

*pBitRate //bittate

*pFrameRate //framerate

Return value

TRUE //successful

FALSE //failure

1.12 GSNET_ClientSetVideoParam

Purpose: Set Video Parameters

Function

BOOL GSNET_ClientSetVideoParam(LONG hHandle, BYTE cbBrightness, BYTE cbContrast, BYTE cbSaturation);

Description: Set Video Parameter

Parameter

hHandle	//Return value of Function 1.4
cbBrightness	// Brightness
cbContrast	//Contrast
cbSaturation	//Saturation

Return value

TRUE	//successful
FALSE	//failure

1.13 GSNET_ClientGetVideoParam

Purpose: Get the video Parameters

Function

BOOL GSNET_ClientGetVideoParam(LONG hHandle, BYTE *pcbBrightness, BYTE *pcbContrast, BYTE *pcbSaturation);

Description: Get the video Parameters

Parameter

hHandle	//Return value of Function 1.4
cbBrightness	//Brightness
cbContrast	//Contrast
cbSaturation	//Saturation

Return value

TRUE	//successful
FALSE	//failure

1.14 GSNET_ClientVerticalFlip

Purpose: flip the video180° vertically

Function: BOOL GSNET_ClientVerticalFlip(LONG hHandle, BOOL bVFlip);

Description: flip the video 180° vertically

Parameter

hHandle	//Return value of Function 1.4
bVFlip	//if the value is TRUE, then flip; if the value is FALSE, then do not flip

Return value

TRUE	//successful
FALSE	//failure

1.15 GSNET_ClientRotate180

Purpose: Rotate video 180°

Function: void GSNET_ClientRotate180(LONG hHandle, BOOL bRotate180);

Description: Rotate video180°

Parameter

hHandle //Return value of Function 1.4

bRotate180 //if the value is TRUE, then Rotate; if the value is FALSE, then do not Rotate

1.16 GSNET_ClientPlayAudio

Purpose: Play audio

Function: BOOL GSNET_ClientPlayAudio(LONG hHandle);

Description: Play audio

Parameter

hHandle //return value of Function 1.4. If to replay files this should be the return value of function1.29

Return value

TRUE //successful

FALSE //failure

1.17 GSNET_ClientStopAudio

Purpose: Stop audio

Function: BOOL GSNET_ClientStopAudio (LONG hHandle);

Description: Stop audio

Parameter

hHandle //return value of Function 1.4. If to replay files this should be the return value of function1.29

Return value

TRUE //successful

FALSE //failure

1.18 GSNET_ClientCapturePicture

Purpose: take a snapshot

Function: BOOL GSNET_ClientCapturePicture(LONG hHandle, LPCTSTR filename);

Description: take a snapshot. And also can be used to replay files

Parameter

hHandle //return value of Function 1.4. If to replay files this should be the return value of function1.29

name //BMP format

Return value

TRUE //successful

FALSE //failure

Remarks

The filename must be ended with ".bmp", eg: c:\123.bmp

1.19 GSNET_ClientStartRecord

Purpose: Start to record

Function

BOOL GSNET_ClientStartRecord(LONG hHandle, LPCTSTR filename, DWORD dwDurationSeconds=0);

Description: Start to record

Parameter

hHandle //return value of Function 1.4

filename //record path

dwDurationSeconds //duration for files(unit :second)，if 0 record always

Return value

TRUE //successful

FALSE //failure

1.20 GSNET_ClientStopRecord

Purpose: Stop recording

Function: BOOL GSNET_ClientStopRecord(LONG hHandle)

Description: Stop recording

Parameter

hHandle //return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

1.21 GSNET_ClientGetMDInfo

Purpose: Get motion detection regions

Function

BOOL GSNET_ClientGetMDInfo(LONG hHandle, MOTION_DETECT_INFO* pMDInfo/*out*/)

Description: The motion detection region settings are based on resolution 704*576

Parameter

hHandle //return value of Function 1.4

* pMDInfo //see struct MOTION_DETECT_INFO

Return value

TRUE //successful

FALSE //failure

1.22 GSNET_ClientSaveMDInfo

Purpose: Save motion detection region

Function

BOOL GSNET_ClientSaveMDInfo(LONG hHandle, MOTION_DETECT_INFO* pMDInfo/*out*/)

Description: The motion detection region settings are based on resolution 704*576

Parameter

hHandle //return value of Function 1.4

* pMDInfo //see struct MOTION_DETECT_INFO

Return value

TRUE //successful

FALSE //failure

1.23 GSNET_ClientStartMD

Purpose: Start motion detection

Function: BOOL GSNET_ClientStartMD(LONG hHandle)

Description: Start motion detection

Parameter

hHandle //return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

1.24 GSNET_ClientStopMD

Purpose: Close motion detection

Function: BOOL GSNET_ClientStopMD(LONG hHandle)

Description: Close motion detection

Parameter

hHandle //return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

1.25 GSNET_ClientStopAlarm

Purpose: Stop alarm output

Function: BOOL GSNET_ClientStopAlarm(LONG hHandle, ULONG ulDeviceNum);

Description: Stop alarm output

Parameter

hHandle //return value of Function 1.4

ulDeviceNum //I/O index

Return value

TRUE //successful

FALSE //failure

1.26 GSNET_ClientShowMDRegion

Purpose: Show motion detect regions

Function: BOOL GSNET_ClientShowMDRegion(LONG hHandle, ULONG ulShow);

Description: Show motion detect regions

Parameter

hHandle //return value of Function 1.4

ulShow //from low to high bit ,0 bit show all set region(wite), 1 bit show motion detected region（red）,default ulShow = 0x00000002

Return value

TRUE //successful

FALSE //failure

Remarks

ulShow

31 bit	30 bit	…	1 bit	0 bit
			If 0, not show motion detected region	If 0, not show setting regions
			If 1, show motion detected region(red)	If 1, show setting regions(white)

1.27 GSNET_ClientStartTalk

Purpose: Open talking

Function: BOOL GSNET_ClientStartTalk(LONG hHandle)

Description: Open talking

Parameter

hHandle //return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

Remarks

For the 4-channel DVS, only one channel can open talking at one time

1.28 GSNET_ClientStopTalk

Purpose: Stop talking

Function: BOOL GSNET_ClientStopTalk(LONG hHandle);

Description: Stop talking

Parameter

hHandle //return value of Function 1.4

Return value

TRUE //successful

FALSE //failure

1.29 GSNET_OpenFile

Purpose: Play the recorded file

Function: LONG GSNET_OpenFile(char* filename, HWND hWnd, BOOL bPause=FALSE)

Description: Play the recorded file

Parameter

filename //record file path

hWnd //display window hanndle

bPause

Return value

0 // failure

Nonzero //successful

1.30 GSNET_CloseFile

Purpose: Stop replaying the recorded file

Function: void GSNET_CloseFile(LONG hHandle)

Description: Stop replaying the recorded file and destroy resourse

Parameter

hHandle //return value of Function 1.29

1.31 GSNET_ReplayPause

Purpose: Pause the playing file

Function: BOOL GSNET_ReplayPause(LONG hHandle);

Parameter

hHandle //return value of Function 1.29

Return value

TRUE //successful

FALSE //failure

1.32 GSNET_ReplayContinue

Purpose: Continue to play recorded file

Function: BOOL GSNET_ReplayContinue(LONG hHandle);

Description: If you use Function 1.36 to pause, you can use it to continue

Parameter

hHandle //return value of Function 1.29

Return value

TRUE //successful

FALSE //failure

1.33 GSNET_ReplayStepByStep

Purpose: Repaly a frame step by step

Function: BOOL GSNET_ReplayStepByStep(LONG hHandle);

Description: Repaly a frame step by step

Parameter

hHandle //return value of Function 1.29

Return value

TRUE	//successful
FALSE	//failure

1.34 GSNET_SpeedNormal

Purpose: Go back to replay in normal speed

Function: BOOL GSNET_SpeedNormal(LONG hHandle)

Description: Go back to replay in normal speed

Parameter

hHandle //return value of Function 1.29

Return value

TRUE	//successful
FALSE	//failure

Remarks

1.35 GSNET_SpeedFast

Purpose: Replay in faster speed

Function: BOOL GSNET_SpeedFast(LONG hHandle);

Description: Replay 2 times faster

Parameter

hHandle //return value of Function 1.29

Return value

TRUE	//successful
FALSE	//failure

Remarks

The fastest speed is 4 times the normal speed

1.36 GSNET_SpeedSlow

Purpose: Replay in slower speed

Function: ULONG GSNET_SpeedSlow(LONG hHandle)

Description: Replay two times slower

Parameter

hHandle //return value of Function 1.29

Return value

TRUE	//successful
FALSE	//failure

Remarks: The slowest speed is ¼ of the normal speed

1.37 GSNET_ReplayTotalTime

Purpose: Get the duration of file

Function: BOOL GSNET_ReplayTotalTime(LONG hHandle)

Description: Get the duration of file

Parameter

hHandle //return value of Function 1.29

Return value

Uint: second

1.38 GSNET_ReplayCurTime

Purpose: Current location of replay file

Function: BOOL GSNET_ReplayCurTime(LONG hHandle)

Description: Current location of replay file

Parameter

hHandle //return value of Function 1.29

Return value

Uint: second

1.39 GSNET_ReplaySeek

Purpose: Designcurrent play time

Function: BOOL GSNET_ReplaySeek(LONG hHandle, ULONG ulSeconds);

Parameter

hHandle //return value of Function 1.29

ulSeconds //uint: second

Return value

TRUE //success

FALSE //failure

1.40 GSNET_ReplaySeek

Purpose: Designate current play time

Function: BOOL GSNET_ClientPTZCtrl(LONG hHandle, int type, int param);

Parameter

hHandle //Return value of Function 1.07

type //PTZ definition; see 3.1 defintion

param //PTZ speed

Return value

TRUE //success

FALSE //failure

Data Structure

2.1 Transfer protocol

```
enum{
        PROTOCOL_RTSP_UDP     = 0,            //UDP
        PROTOCOL_RTSP_TCP     = 1,       //TCP (Recommend)
        PROTOCOL_RTSP_MCAST = 2,      //reserved
};
```

2.2 CHANNEL_CLIENTINFO

```
typedef struct{
        WORD m_protocol;           // Transfer protocol
        WORD m_playstart;            //show video 1 show; 0 not show
        BYTE m_ch;                       //channel number 0-3 primary stream 4-7 secondary stream
        HWND m_hVideohWnd;      //window handle for video shown
        HWND m_hChMsgWnd;      //reserved
        UINT m_nChMsgID;          // reserved
        int    m_buffnum;               // reserved
        int    m_useoverlay;           // reserved
        COLORREF nColorKey;    //color key(reserved)

        Void *lpReserved                 //reserved
        void *callbackContext;                     //context for callback Function

        char url128;                      // hostname:port    eg: 192.168.83.254:554
        char m_sernameMAX_SERVER_NAME_LEN+1;    //reserved
        char m_usernameMAX_USER_NAME_LEN+1;      //user name
        char m_passwordMAX_PASSWORD_LEN+1;       //password
}CHANNEL_CLIENTINFO;
```

2.3 structure for motion detect region

```
typedef struct{
        BYTE      cbSensitiveMAX_MOTION_REGION;// Sensitive    0 -100
        RECT      rcRegionMAX_MOTION_REGION; //acording to 704*576
}MOTION_DETECT_INFO;
```

Definition

3.1 Definition

```
#define MAX_MOTION_REGION          16
#define MAX_SERVER_NAME_LEN        32
#define MAX_USER_NAME_LEN          32
#define MAX_PASSWORD_LEN                64
#define MAX_FILENAME_LEN           255

/******************** callback Parameter****************/

/********************link message************************/
/*wParam*/
#define GSMSG_LINKMSG              1          //link to device

/*lParam*/
#define GSMSG_LINKMSG_OK                     0    //OK
#define GSMSG_LINKMSG_CONNECTING              1    //connecting
#define GSMSG_LINKMSG_FAILURE                 2    //connect failure
#define GSMSG_LINKMSG_DISCONNECT              3    //disconnect
#define GSMSG_LINKMSG_RECONNECT                   4     //reconnect
#define GSMSG_LINKMSG_PLAYFAILURE             5    //play failure
/*******************************************************/

/********************motion detect*********************/
/*wParam*/
#define GSMSG_RECORD               6          //record state
/*lParm*/
#define GSMSG_RECORD_BEGIN_NORMAL_RECORD          0
#define GSMSG_RECORD_END_NORMAL_RECORD            1
#define GSMSG_RECORD_BEGIN_ALARM_RECORD           2
#define GSMSG_RECORD_END_ALARM_RECORD             3
#define GSMSG_RECORD_NORMAL_PACKET_FINISH         4
#define GSMSG_RECORD_ALARM_PACKET_FINISH      5
/*******************************************************/


/*wParam*/
#define GSMSG_VIDEOMOTION          2          //motion detect

/*lParam*/
lParam:    from 0-15 bit dentify 0-15 motion detect region torespectively
/*********************************************************/
/*wParam*/
#define GSMSG_VIDEOLOST            3          //video lost alarm reserved
#define GSMSG_ALARM                4          // Probe alarm
```

```
#define GSMSG_OUTPUTSTATUS       5         //out put alarm reserved
/***************************PTZ control******************************/
#define PTZ_STOP                 0    /*    stop*/

#define TILT_UP             1    /* up*/
#define TILT_DOWN           2    /* down*/
#define PAN_LEFT            3    /* left*/
#define PAN_RIGHT                4    /* right*/

#define PT_LEFT_UP               5    /* left-up*/
#define PT_LEFT_DOWN             6    /* left -down*/
#define PT_RIGHT_UP                   7    /* right-up*/
#define PT_RIGHT_DOWN                 8    /* right-down*/

#define PTZ_ZOOM_IN              9    /* zoom in */
#define PTZ_ZOOM_OUT             10   /* zoom out*/
#define FOCUS_NEAR                    11        /* focus near*/
#define FOCUS_FAR                12   /* focus far*/
#define IRIS_OPEN                13   /* iris open*/
#define IRIS_CLOSE               14   /* iris close*/

#define GOTO_PRESET         15   /* goto pre-set*/
#define CLE_PRESET          16   /* clesr pre-set*/
#define SET_PRESET          17   /* set pre-set*/

#define PAN_AUTO            18   /* auto scan start*/
#define PAN_AUTO_STOP 19    /* auto scan stop*/
```