



USB 2.0 Camera SDK Manual



Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com



Table of Contents

1.0 Construction of SDK	3
2.0 Method of API (Application Program Interface)	4
2.1 For Visual C++ 6.0 / Visual C++ .NET 2003 [MFC Application]	4
2.2 For Visual Basic 6.0	4
2.3 For Visual C# .NET 2003	5
2.4 For Visual C++ .NET 2003[Windows Form Application]	6
2.5 For Visual Basic .NET 2003	6
2.6 For Borland C++ Builder 5.0	7
3.0 Function List	8
4.0 Initialization	11
5.0 Image Information	16
6.0 Preview Image	22
7.0 Image Acquisition	43
8.0 Shutter and Gain	47
9.0 White Balance	58
10.0 Gamma	66
11.0 Sharpness	68
12.0 Hue and Saturation	70
13.0 Color Matrix	72
14.0 Mirror and Rotation	73
15.0 Save AVI File	76
16.0 Clock	80
17.0 Callback	84
18.0 Setting	93
19.0 EEPROM	95
20.0 Version Information	99
21.0 Others	103



1.0 Construction of SDK

1	StCam_SDK.chm	SDK Manual * ¹
2	StUSB.dll	USB communication DLL File * ²
3	StCamMsg.dll	Error message DLL File * ²
4	StCamD.dll	Camera control DLL File * ²
5	StCamD.h	Header file for Visual C++ 6.0 & Visual C++ .NET [MFC Application]
6	StCamD.lib	Library file for Visual C++ 6.0 & Visual C++ .NET [MFC Application]
7	StCamD.bas	for Visual Basic 6.0
8	StCamD.cs	for Visual C#. NET
9	StCamD.hpp	for Visual C++ .NET [Windows Form Application]
10	StCamD.vb	for Visual Basic .NET

Note *1: This manual is written for Visual C++ 6.0 platform
If you are using a different platform (except Visual C++ 6.0), the type of variable is different from the variables in this manual

Note *2: These files must be copied to the same folder (Directory) as application or following folder:

- a) Application directory
- b) Current directory
- c) Windows directory
- d) Entry directory on the environment variables

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

2.0 Method of API (Application Program Interface)

2.1) For Visual C++ 6.0 / Visual C++ .NET 2003[MFC Application]

A) Link “StCmaD.lib” use one of the following two methods.

(1) Add “StCamD.lib” to [Object / Library Module] of Category [General] on the [Linker] window of [Project]-[dd properties]

(2) Write “#pragma comment(lib, “StCamD.lib”)” in your own program. If “StCamD.h” exists outside of the project folder, specify the pass.

B) Include “StCamD.h.”. (For example: #include “StCamD.h”)

C) When you achieve “preview”, write the following codes, for example.

```
HANDLE      m_hCamera = StCam_Open(0);

if(m_hCamera)
{
    StCam_CreatePreviewWindow(
        m_hCamera,TEXT("Preview"),
        WS_OVERLAPPEDWINDOW | WS_VISIBLE,
        0,0,0,0,NULL,NULL,FALSE);
    StCam_StartTransfer(m_hCamera);

    MessageBox(
        NULL,
        TEXT("When stop preview please select [OK]."),
        TEXT("Stop"),MB_OK);

    StCam_StopTransfer(m_hCamera);
    StCam_Close(m_hCamera);
}
```

2.2 For Visual Basic 6.0

A) Add “StCamD.Bas” to the Project

2.2 For Visual Basic 6.0 (Continued)

B) When you achieve “preview”, write the following codes, for example:

```
Dim m_hCamera As Long
Dim lngReval As Long
Const clngFalse As Long = -1

m_hCamera = StCam_Open(0)
If m_hCamera <> 0 Then
    lngReval = StCam_CreatePreviewWindowA( _
        m_hCamera, "Preview", _
        WS_OVERLAPPEDWINDOW Or WS_VISIBLE, _
        0, 0, 0, 0, 0, 0, clngFalse)
    lngReval = StCam_StartTransfer(m_hCamera)
    MsgBox _
        "When stop preview please select [OK].", _
        vbOKOnly, "Stop"
    StCam_Close (m_hCamera)
End If
```

2.3 For Visual C# .NET 2003

A) Add “StCamD.cs” to the Project.

B) Declare the Namespace “SensorTechnology”. (For example: using namespace SensorTechnology;)

C) When you achieve “preview”, write the following codes, for example:

```
System.IntPtr m_hCamera = StCam.Open(0);
if(m_hCamera != System.IntPtr.Zero)
{
    StCam.CreatePreviewWindow(
        m_hCamera,
        "Preview",
        StCam.WS_OVERLAPPEDWINDOW | StCam.WS_VISIBLE,
        0,0,0,0,
        System.IntPtr.Zero, System.IntPtr.Zero, false);
}
```

2.4 For Visual C++ .NET 2003[Windows Form Application]

- A) Include “StCamD.hpp”. (For example: #include “StCamD.hpp”)
- B) Declare the Namespace “SensorTechnology”. (For example: using namespace SensorTechnology;)
- C) When you achieve “preview”, write the following codes, for example:

```
int m_hCamera = StCam::Open(0);
if(m_hCamera)
{
    StCam::CreatePreviewWindow(
        m_hCamera,
        "Preview",
        StCam::WS_OVERLAPPEDWINDOW | StCam::WS_VISIBLE,
        0,0,0,0,
        System::IntPtr::Zero, System::IntPtr::Zero, false);
    StCam::StartTransfer(m_hCamera);
    MessageBox::Show(
        "When stop preview please select [OK].",
        "Stop", MessageBoxButtons::OK);
    StCam::Close(m_hCamera);
}
```

2.5 For Visual Basic .NET 2003

- A) Add “StCamD.vb” to Project
- B) When you achieve “preview”, write the following codes, for example:

```
Dim m_hCamera As Integer
m_hCamera = StCam.Open(0)
If m_hCamera Then
    StCam.CreatePreviewWindow( _
        m_hCamera, "Preview", _
        StCam.WS_OVERLAPPEDWINDOW Or StCam.WS_VISIBLE, _
        0, 0, 0, 0, 0, 0, False)
    StCam.StartTransfer(m_hCamera)

    MsgBox( _
        "When stop preview please select [OK].", _
        MsgBoxStyle.OKOnly, "Stop")
    StCam.Close(m_hCamera)
End If
```

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

2.6 For Borland C++ Builder 5.0

A) Make LIB File from DLL File by using “implib.exe” that is included with Borland C++ Builder.

```
>implib StCamDBCB.lib StCamD.dll
```

B) Chose the Lib File produced in the item a) by [Project]-[Add Project].

C) Include StCmaD.h. (For example: #include “StCamD.h”)

D) When you achieve “preview”, write the following codes, for example:

```
HANDLE m_hCamera = StCam_Open(0);

if(m_hCamera)
{
    StCam_CreatePreviewWindow(
        m_hCamera,TEXT("Preview"),
        WS_OVERLAPPEDWINDOW | WS_VISIBLE,
        0,0,0,0,NULL,NULL,FALSE);
    StCam_StartTransfer(m_hCamera);

    MessageBox(
        NULL,
        TEXT("When stop preview please select [OK]"),
        TEXT("Stop"),MB_OK);

    StCam_StopTransfer(m_hCamera);
    StCam_Close(m_hCamera);
}
```

3.0 Function List

Function	Name	Description
Initialization	StCam_Open	Open the Camera and acquires the Camera Handle.
	StCam_Close	Close the Camera.
	StCam_GetLastError	The latest error number occurred in the SDK is obtained.
	StCam_CameraCount	The number of the Cameras that are connected to the PC is acquired.
	StCam_SetReceiveMsgWindow	Set the window that acquires message sent from SDK.
Information of Image	StCam_GetColorArray	The Color Order Information of Color Filter is of Image acquired.
	StCam_SetTransferBitsPerPixel	Set the number of Bits each pixel of image data from the Camera to the PC.
	StCam_GetTransferBitsPerPixel	The number of Bits each pixel of image data from the Camera to the PC is acquired.
	StCam_GetEnableTransferBitsPerPixel	The number of Bits each pixel of image data that can be set is acquired.
	StCam_SetImageSize	Set the image size.
	StCam_GetImageSize	The image size is acquired.
	StCam_GetEnableImageSize	The image size that can be set is acquired.
Preview Image	StCam_StartTransfer	Start image data transfer from the Camera.
	StCam_StopTransfer	Finish Image Data transfer from the Camera.
	StCam_SetPreviewPixelFormat	Set the Pixel Format of Preview Image Data.
	StCam_GetPreviewPixelFormat	The pixel format of preview image data is acquired.
	StCam_GetEnablePreviewPixelFormat	The pixel format of preview that can be set is acquired.
	StCam_SetColorInterpolationMethod	Set the Color Interpolation Method.
	StCam_GetColorInterpolationMethod	The Color Interpolation Method is acquired.
	StCam_CreatePreviewWindowA	Create the preview window.
	StCam_CreatePreviewWindowW	Create the preview window.
	StCam_DestroyPreviewWindow	Close the preview window
	StCam_SetPreviewWindowNameA	Set the preview window name
	StCam_GetPreviewWindowNameA	The current preview window name is acquired.
	StCam_SetPreviewWindowNameW	Set the preview window name
	StCam_GetPreviewWindowNameW	The current preview window name is acquired.
	StCam_SetPreviewMaskSize	Set the preview target image.
	StCam_GetPreviewMaskSize	The preview target image is acquired.
	StCam_SetPreviewWindowSize	Set the preview window size.
	StCam_GetPreviewWindowSize	The preview window size is acquired.
	StCam_SetPreviewWindowStyle	Set the preview window style.
	StCam_GetPreviewWindowStyle	The current preview window style is acquired.
	StCam_SetAspectMode	Set the Aspect Mode when resize preview window.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

3.0 Function List (Continued)

Function	Name	Description
Preview Image	StCam_GetAspectMode	The Aspect Mode when resize preview window is acquired.
	StCam_SetPreviewDestSize	Set the size of preview image.
	StCam_GetPreviewDestSize	The size of preview image is acquired.
	StCam_SetMagnificationMode	Set the Magnification Mode.
	StCam_GetMagnificationMode	The Magnification Mode is acquired.
Acquires the the Image	StCam_TakeRawSnapShot	The raw image data is acquired.
	StCam_TakePreviewSnapShot	The image data that before preview is acquired.
	StCam_SaveImageA	Save acquiring image data to the file.
	StCam_SaveImageW	Save acquiring image data to file.
Shutter and Gain	StCam_SetShutterSpeed	Set the shutter speed.
	StCam_GetShutterSpeed	The Shutter Speed is acquired.
	StCam_SetGain	Set the Gain.
	StCam_GetGain	The Gain is acquired.
	StCam_SetALCMode	Set the ALC Mode.
	StCam_GetALCMode	The ALC Mode is acquired.
	StCam_SetTargetBrigtness	Set the Brightness target.
	StCam_GetTargetBrigtness	The Brightness target is acquired.
	StCam_SetALCWeight	Set the weight of Brightness of ALC.
	StCam_GetALCWeight	The Weight of Brightness of ALC is acquired.
	StCam_SetShutterControlRange	Set the shutter speed range of ALC.
	StCam_GetShutterControlRange	The shutter speed range of ALC is acquired.
	StCam_SetGainControlRange	Set the Gain range of AGC.
	StCam_GetGainControlRange	The Gain range of AGC is acquired.
	StCam_SetALCControlSpeed	Set the ALC control speed.
	StCam_GetALCControlSpeed	The ALC control speed is acquired.
White Balance	StCam_SetWhiteBalanceMode	Set the White Balance Mode.
	StCam_GetWhiteBalanceMode	Current White Balance Mode is acquired.
	StCam_SetWhiteBalanceGain	Set the White Balance Gain.
	StCam_GetWhiteBalanceGain	Current White Balance Gain is acquired.
	StCam_SetWhiteBalanceTarget	Set the target of Auto White Balance.
	StCam_GetWhiteBalanceTarget	Current target of Auto White Balance is acquired.
	StCam_SetWhiteBalance Tolerance Threshold	Set the tolerance and threshold of Auto White Balance.
	StCam_GetWhiteBalance Tolerance Threshold	Current tolerance and threshold of Auto White Balance is acquired.
	StCam_SetAWBWeight	Set the weight of 16 areas of Auto White Balance.
	StCam_GetAWBWeight	Current weight of 16 Areas of Auto White Balance is acquired.
	Gamma StCam_SetGammaMode	Set the Gamma setting.
	StCam_GetGammaMode	The Gamma setting is acquired.
Sharpness	StCam_SetSharpnessMode	Set the sharpness setting.
	StCam_GetSharpnessMode	The Sharpness setting is acquired.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

3.0 Function List (Continued)

Function	Name	Description
Hue & Saturation	StCam_SetHueSaturation	Set the Hue & Saturation setting.
	StCam_GetHueSaturation	Acquire the Hue & Saturation setting.
Color Matrix	StCam_SetColorMatrix	Set the Color Matrix.
	StCam_GetColorMatrix	The Color Matrix is acquired.
Mirror and Rotation	StCam_SetMirrorMode	Set the Mirror Mode.
	StCam_GetMirrorMode	The Mirror Mode is acquired.
	StCam_SetRotationMode	Set the Rotation Mode.
	StCam_GetRotationMode	The Rotation Mode is acquired.
Save AVI Format File	StCam_SaveAVIA	Save AVI file.
	StCam_SaveAVIW	Save AVI file.
	StCam_SetAVIStatus	Set the status of making AVI file during make AVI file.
	StCam_GetAVIStatus	The status of making AVI file is acquired.
Clock	StCam_SetClock	Set the clock speed.
	StCam_GetClock	The clock speed is acquired.
	StCam_GetEnableClock	The clock speed that can be set is acquired.
	StCam_GetFrameClock	The number of Line each frame and the number of clock each line are acquired.
Callback	StCam_AddPreviewBitmapCallback	Enter the Preview Bitmap Callback function.
	StCam_RemovePreviewBitmapCallback	Delete the Preview Bitmap Callback function.
	StCam_RemoveAllPreviewBitmapCallback	Delete the all of Preview Bitmap Callback functions.
	StCam_GetPreviewBitmapCallbackCount	The number of the Preview Bitmap Callback function that already entered is acquired.
	StCam_GetPreviewBitmapCallback	The address and entry number of the Preview Bitmap Callback function already entered are acquired.
	StCam_AddPreviewGDIcallback	Enter the Preview GDI Callback function.
	StCam_RemovePreviewGDIcallback	Delete the Preview GDI Callback function.
	StCam_RemoveAllPreviewGDIcallback	Delete the all of Preview GDI Callback functions.
	StCam_GetPreviewGDIcallbackCount	The number of the Preview GDI Callback function that already entered. Is acquired.
	StCam_GetPreviewGDIcallback	The address and entry number of the Preview GDI Callback function already entered are acquired.
	StCam_AddRawCallback	Enter the Raw Callback function.
	StCam_RemoveRawCallback	Delete the Raw Callback function.
	StCam_RemoveAllRawCallback	Delete the all of Raw Callback functions.
	StCam_GetRawCallbackCount	The number of the Raw Callback function that already entered is acquired.
	StCam_GetRawCallback	The address and entry number of the Raw Callback function already entered are acquired.
Setting	StCam_SaveSettingFileA	Save current setting to the file.
	StCam_SaveSettingFileW	Save current setting to the file.
	StCam_LoadSettingFileA	Load setting from the file.
	StCam_LoadSettingFileW	Load setting from the file.
	StCam_ResetSetting	Reset all of the setup to the Camera default setup.

3.0 Function List (Continued)

Function	Name	Description
EEPROM	StCam_ReadUserMemory	Read the user data from the EEPROM of the Camera.
	StCam_WriteUserMemory	Write the user Data to the EEPROM of the Camera.
	StCam_ReadCmeraUserIDA	The Camera ID from the EEPROM of the Camera is acquired.
	StCam_ReadCmeraUserIDW	The Camera ID from the EEPROM of the Camera is acquired.
	StCam_WriteCmeraUserIDA	Set the Camera ID to the EEPROM of the Camera.
	StCam_WriteCmeraUserIDW	Set the Camera ID to the EEPROM of the Camera.
Version Information	StCam_GetCameraVersion	The version information of the Camera is acquired.
	StCam_GetDriverVersion	The version information of USB driver is acquired.
	StCam_GetUSBDIIVersion	The version information of StUSBD.dll. is acquired.
	StCam_GetCAMDIIVersion	The version information of StCamD.dll is acquired.
	StCam_GetUSBFunctionAddress	The USB function address is acquired.
Other	StCam_ConvertBitmapBGR24ToRGB24	Convert the BGR 24 Bits image data to RGB 24 Bits image data.
	StCam_ConvertRawToBGR	Color Interpolation process to the raw data from the Camera.
	StCam_SetControlArea	Set the area size use for ALC and Auto White Balance.
	StCam_GetControlArea	The area size use for ALC and Auto White Balance is acquired.

4.0 Initialization

A) StCam_Open

1) Call
HANDLE StCam_Open (
DWORD *dwInstance*
);

2) Parameters

a) *dwInstance*
Set the 0.

3) Return Values

If the function succeeds, the return value is the Camera Handle. If the function fails, the return value is "NULL".

4) Description

Open the Camera and acquire the Camera Handle. Operating system does not recognize the Camera immediately after the Camera connects to the PC. It is required to wait a few seconds after the Camera is connected to the PC. It is necessary to close the Camera by StCam_Close function before finishing with software.

4.0 Initialization (Continued)

B) StCam_Close

- 1) Call
VOID StCam_Close (
 HANDLE *hCamera* // Camera Handle
);
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- 3) Return Values
None
- 4) Description
Close the Camera. If the Camera is opened with StCam_Open function, it is necessary to close the Camera with this function before completion.

C) StCam_GetLastError

- 1) Call
DWORD StCam_GetLastError (
 HANDLE *hCamera* // Camera Handle
);
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function. Set the "NULL" when acquisition error of StCam_Open function
- 3) Return Values
The return value is the most recent error number.

4.0 Initialization (Continued)

Error Number of this SDK

Name	Value	Description
ERRST_NOT_FOUND_CAMERA	0xE0000001	The Camera could not be opened. No Camera available to use.
ERRST_ALL_CAMARA_OPENED	0xE0000002	The Camera could not be opened. All Cameras are already opened.
ERRST_INVALID_CAMERA_HANDLE	0xE0000003	Invalid Camera Handle.
ERRST_INVALID_FUNCTION_RECEIVING	0xE0000004	Invalid function called during image transfer.
ERRST_USB_COMMAND_TRANSFER	0xE0000005	Error occurred during the commands transmission to the USB bus.
ERRST_WINDOW_NOW_CREATING	0xE0000006	Preview window is being made right now.
ERRST_WINDOW_DOES_NOT_EXISTS	0xE0000007	Preview window does not exist
ERRST_INVALID_FUNCTION_RECORDING	0xE0000008	Invalid function is called during image recording.
ERRST_AVI_STREAM	0xE0000009	Error occurred during AVI image transmission.
ERRST_AVI_NOCOMPRESSOR	0xE000000A	Designate CODEC cannot be found.
ERRST_AVI_UNSUPPORTED	0xE000000B	Image data compression of this camera is not supported by designates CODEC. Please check available CODEC on description of StCam_SaveAVIA function.
ERRST_AVI_DISK	0xE000000C	Disk error occurred during saving AVI file.
ERRST_AVI_CANCELED	0xE000000D	Saving AVI File was canceled.
ERRST_AVI_WRITE	0xE000000E	Error occurred during saving AVI file.
ERRST_INVALID_FILE_NAME	0xE000000F	Invalid file name.
ERRST_FILE_OPEN	0xE0000010	Opening file has failed.
ERRST_FILE_WRITE	0xE0000011	Writing file has failed.

4) Description

The latest error number occurred in the SDK is obtained. The error number is assigned by Operating System or SDK. If new error occurs before acquiring the previously occurred error number, the error number is over written by the new error number. When acquiring error number, this function must be called immediately after error occurs.

4.0 Initialization (Continued)

D) StCam_CameraCount

- 1) Call
 DWORD StCam_CameraCount (
 VOID
);
- 2) Parameters
 None
- 3) Return Values
 Number of the Camera connects to the PC.
- 4) Description
 The number of the Cameras that are connected to the PC is acquired. Operating System does not recognize the Camera immediately after the Camera connects to the PC. It is required to wait few seconds after the Camera is connected to the PC. It requires a few seconds after the Camera is connected to the PC to recognize the Camera connection.

E) StCam_SetReceiveMsgWindow

- 1) Call
 BOOL StCam_SetReceiveMsgWindow (
 HANDLE *hCamera* , // Camera Handle
 HWND *hWnd* // Window Handle
);
- 2) Parameters
 - a) *hCamera*
 Set the Camera Handle that is acquired by StCam_Open function.
 - b) *hWnd*
 Set the window Handle that acquires the message.

4.0 Initialization (Continued)

Message Constant of SDK

Name	Value	Description
WM_STCAM_TRANSFER_ START	0xB001	Image transfer has started (wParam:hCamera,IParam:0x0000)
WM_STCAM_TRANSFER_FINISH	0xB002	Image transfer has finished (wParam:hCamera,IParam>Error Number)
WM_STCAM_PREVIEW_WINDOW_CREATE	0xB003	Preview window was created (wParam:hCamera,IParam:hWnd)
WM_STCAM_PREVIEW_WINDOW_CLOSE	0xB004	Preview window has closed (wParam:hCamera,IParam>Error Number)
WM_STCAM_PREVIEW_WINDOW_RESIZE	0xB005	Preview window was resized (wParam:hCamera,IParam:0x0000)
WM_STCAM_PREVIEW_MASK_RESIZE	0xB006	Preview Mask was resized (wParam:hCamera,IParam:0x0000)
WM_STCAM_PREVIEW_DEST_RESIZE	0xB007	Preview image was resized (wParam:hCamera,IParam:0x0000)
WM_STCAM_AVI_FILE_START	0xB008	AVI File writing has started (wParam:hCamera,IParam:0x0000)
WM_STCAM_AVI_FILE_FINISH	0xB009	AVI File writing has finished (wParam:hCamera,IParam>Error Number)
WM_STCAM_PREVIEW_MENU_COMMAND	0xB00A	Menu of preview window was clicked (wParam:hCamera,HIWORD(IParam):0x00,LOWORD (IParam):Menue ID)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

Set the window that acquires message sent from SDK. The status change of SDK's inside can be obtained by this function.

5.0 Image Information

A) StCam_GetColorArray

1) Call

```
BOOL StCam_GetColorArray (  
    HANDLE      hCamera ,      // Camera Handle  
    PWORD      pwColorArray // Color Order Information of Color Filter  
);
```

2) Parameter

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pwColorArray*

Set the pointer to the variable that acquires the Color order Information of Color filter.

Color Order Information of Color Filter

Name	Value	Description
STCAM_COLOR_ARRAY_MONO	0x0001	Monochrome
STCAM_COLOR_ARRAY_RGBB	0x0002	RGrGbB Order
STCAM_COLOR_ARRAY_GRBG	0x0003	GrRBGb Order
STCAM_COLOR_ARRAY_GBRG	0x0004	GbBRGr Order
STCAM_COLOR_ARRAY_BGGR	0x0005	BGbGrR Order

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Color Order Information of Color Filter is acquired. Use this function when processing the raw video data before the color interpolation processing by StCam_TakeRawSnapShot or StCam_RemoveRawCallback function.

B) StCam_SetTransferBitsPerPixel

1) Call

```
BOOL StCam_SetTransferBitsPerPixel (  
    HANDLE      hCamera ,      // Camera Handle  
    DWORD      dwTransferBitsPerPixel // Number of Bits per 1 Pixel  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwTransferBitsPerPixel*

Set the number of bits of each pixel of image data from the camera to the PC. The numbers of bits of each pixel available in each camera can be acquired by StCam_GetEnableTransferBitsPerPixel function.

5.0 Image Information (Continued)

Number of Bits each pixel at Transfer

Name	Value	Description
STCAM_TRANSER_BITS_PER_PIXEL_08	0x00000001	8 Bits each pixel
STCAM_TRANSER_BITS_PER_PIXEL_16	0x00000002	16 Bits each pixel

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the number of Bits of each pixel of image data from the Camera to the PC. When this number is not set at eight, the data for preview is rounded into 8 bits. To obtain the data before rounded, use StCam_TakeRawSnapShot or RAW_Callback function.

C) StCam_GetTransferBitsPerPixel

1) Call

```
BOOL StCam_GetTransferBitsPerPixel (  
    HANDLE hCamera ,           // Camera Handle  
    PDWORD pdwTransferBitsPerPixel // Number of Bits per 1 Pixel  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pdwTransferBitsPerPixel

Set the pointer to the variable that acquires the number of bits of each pixel of image data sent from the camera to the PC.

Number of Bits each pixel at Transfer

Name	Value	Description
STCAM_TRANSER_BITS_PER_PIXEL_08	0x00000001	8 Bits each pixel
STCAM_TRANSER_BITS_PER_PIXEL_16	0x00000002	16 Bits each pixel

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The number of Bits of each pixel of image data from the Camera to the PC is acquired. When this number is not set at eight, the data for preview is rounded into 8 bits. To obtain the data before rounded, use StCam_TakeRawSnapShot or RAW_Callback Function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

5.0 Image Information (Continued)

D) StCam_GetEnableTransferBitsPerPixel

1) Call

```

    BOOL StCam_GetEnableTransferBitsPerPixel (
        HANDLE   hCamera ,           // Camera Handle
        PDWORD  pdwEnableTransferBitsPerPixel // Number of Bits per 1 Pixel
    );

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pdwEnableTransferBitsPerPixel

Set the pointer to the variable that is the result of logical OR operation of the number of Bits of each pixel of image data. This variable means the number of bits of each pixel of image data that is set.

When the Camera returns 0x00000003, the Camera supports 8 Bits and 16 Bits.

Number of Bits each pixel at Transfer

Name	Value	Description
STCAM_TRANSER_BITS_PER_PIXEL_08	0x00000001	8 Bits each pixel
STCAM_TRANSER_BITS_PER_PIXEL_16	0x00000002	16 Bits each pixel

3) Return Value

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The number of Bits each pixel of image data that can be set is acquired. When this number is not set at eight, the data for preview is rounded into 8 bits. To obtain the data before rounded, use StCam_TakeRawSnapShot or ROW Callback Function.

E) StCam_SetImageSize

1) Call

```

    BOOL StCam_SetImageSize (
        HANDLE   hCamera ,           // Camera Handle
        DWORD    dwImageSizeMode ,  // Image Size Mode
        WORD     wScanMode ,        // Scanning Mode
        DWORD    dwOffsetX ,        // Horizontal Axis Offset
        DWORD    dwOffsetY ,        // Vertical Axis Offset
        DWORD    dwWidth ,          // Width of Image
        DWORD    dwHeight           // Height of Image
    );

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) dwImageSizeMode

Set the Image Size Mode. The image sizes available for each camera can be acquired by StCam_GetEnableImageSize function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

5.0 Image Information (Continued)

E) StCam_SetImageSize (Continued)

2) Parameters (Continued)

Image Size Mode

Name	Value	Description
STCAM_IMAGE_SIZE_MODE_CUSTOM	0x00000001	Custom
STCAM_IMAGE_SIZE_MODE_VGA	0x00000008	640 x 480
STCAM_IMAGE_SIZE_MODE_XGA	0x00000020	1024 x 768

c) *wScanMode*

Set the Scanning Mode. The scan modes available for each camera can be acquired by StCam_GetEnableImageSize function.

Scanning Mode

Constant Name	Value	Description
STCAM_SCAN_MODE_NORMAL	0x0000	Normal
STCAM_SCAN_MODE_PARTIAL_2	0x0001	1/2 Partial Scanning Mode
STCAM_SCAN_MODE_PARTIAL_4	0x0002	1/4 Partial Scanning Mode

d) *dwOffsetX*

Set the horizontal axis offset of the image.

e) *dwOffsetY*

Set the vertical axis offset of the image.

f) *dwWidth*

Set the width of the image. This parameter is only enabled when STCAM_IMAGE_SIZE_MODE_CUSTOM is set in Image Size Mode (*dwImageSize*).

g) *dwHeight*

Set the height of the image. This parameter is only enabled when STCAM_IMAGE_SIZE_MODE_CUSTOM is set in Image Size Mode (*dwImageSize*).

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The image size sets by this function. The image sizes that can be set can be acquired by StCam_GetEnableImageSize function. This Function cannot be called during transferring image data from the Camera to the PC.

5.0 Image Information (Continued)

F) StCam_GetImageSize

1) Call

```
BOOL StCam_GetImageSize (  
    HANDLE      hCamera ,           // Camera Handle  
    PDWORD      pdwImageSizeMode , // Image Size Mode  
    PWORD       pwScanMode ,        // Scanning Mode  
    PDWORD      pdwOffsetX ,        // Horizontal Axis Offset  
    PDWORD      pdwOffsetY ,        // Vertical Axis Offset  
    PDWORD      pdwWidth ,          // Width of Image  
    PDWORD      pdwHeight           // Height of Image  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwImageSizeMode*

Set the pointer to the variable that acquires the Image Size Mode.

Image Size Mode

Name	Value	Description
STCAM_IMAGE_SIZE_MODE_CUSTOM	0x00000001	Custom
STCAM_IMAGE_SIZE_MODE_VGA	0x00000008	640 x 480
STCAM_IMAGE_SIZE_MODE_XGA	0x00000020	1024 x 768

c) *pwScanMode*

Set the pointer of variable that acquires the Scanning Mode.

Scanning Mode

Name	Value	Description
STCAM_SCAN_MODE_NORMAL	0x0000	Normal
STCAM_SCAN_MODE_PARTIAL_2	0x0001	1/2 Partial Scanning Mode
STCAM_SCAN_MODE_PARTIAL_4	0x0002	1/4 Partial Scanning Mode

d) *pdwOffsetX*

Set the pointer to the variable that acquires the horizontal axis offset of the image.

(0 is returned with the current version)

e) *pdwOffsetY*

Set the pointer to the variable that acquires the vertical axis offset of image

(0 is returned with the current version)

5.0 Image Information (Continued)

F) StCam_GetImageSize (Continued)

2) Parameters (Continued)

f) *dwWidth*

Set the width of the image. This parameter is only enabled when STCAM_IMAGE_SIZE_MODE_CUSTOM is set in Image Size Mode (*dwImageSize*).

g) *dwHeight*

Set the height of the image. This parameter is only enabled when STCAM_IMAGE_SIZE_MODE_CUSTOM is set in Image Size Mode (*dwImageSize*).

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The image size sets by this function. The image sizes that can be set can be acquired by StCam_GetEnableImageSize function. This Function cannot be called during transferring image data from the Camera to the PC.

G) StCam_GetEnableImageSize

1) Call

```
BOOL StCam_GetEnableImageSize (  
    HANDLE      hCamera ,           // Camera Handle  
    PDWORD     pdwEnableImageSizeMode , // Image Size Mode  
    PWORD      pwEnableScanMode      // Scanning Mode  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwEnableImageSizeMode*

Set the pointer of variable that is the result of OR operation of Image Size Mode. This variable means the image sizes that can be set.

5.0 Image Information (Continued)

G) StCam_GetEnableImageSize (Continued)

2) Parameters (Continued)

Image Size Mode

Name	Value	Description
STCAM_IMAGE_SIZE_MODE_CUSTOM	0x00000001	Custom
STCAM_IMAGE_SIZE_MODE_VGA	0x00000008	640 x 480
STCAM_IMAGE_SIZE_MODE_XGA	0x00000020	1024 x 768

c) *pwEnableScanMode*

Set the pointer of variable that is the result of OR operation of Scanning Mode. This variable means the scanning modes that can be set.

When the Camera returns 0x0002, the Camera supports Normal and 1/4 Partial Scanning Mode. When the Camera returns 0x0003, the Camera supports Normal, 1/2 Partial Scanning Mode, and 1/4 Partial Scanning Mode.

Scanning Mode

Name	Value	Description
STCAM_SCAN_MODE_NORMAL	0x0000	Normal
STCAM_SCAN_MODE_PARTIAL_2	0x0001	1/2 Partial Scanning Mode
STCAM_SCAN_MODE_PARTIAL_4	0x0002	1/4 Partial Scanning Mode

3) Return Value

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The image size that can be set is acquired.

6.0 Preview Image

A) StCam_StartTransfer

1) Call

```
BOOL StCam_StartTransfer (  
    HANDLE          hCamera    // Camera Handle  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

6.0 Preview Image

A) StCam_StartTransfer

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Start image data transfer from the Camera. If the preview window is opened, start preview. The preview window makes by StCam_CreatePreviewWindowA or StCam_CreatePreviewWindowW function. If use Visual Basic, have to use StCam_CreatePreviewWindowA to make preview window.

B) StCam_StopTransfer

1) Call

```
BOOL StCam_StopTransfer (  
    HANDLE      hCamera      // Camera Handle  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that acquired by StCam_Open function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Finish image data transfer from the Camera. During process of saving moving image, if you stop process, data will stop saving.

C) StCam_SetPreviewPixelFormat

1) Call

```
BOOL StCam_SetPreviewPixelFormat (  
    HANDLE      hCamera ,      // Camera Handle  
    DWORD      dwPreviewPixelFormat // Pixel Format of Preview  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) dwPreviewPixelFormat

Set the pixel format of preview.

6.0 Preview Image (Continued)

C) StCam_SetPreviewPixelFormat (Continued)

2) Parameters (Continued)

Pixel Format of Preview

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the pixel format of preview image data. Values that can be set can check by StCam_GetEnablePreviewPixelFormat function. This function cannot call during image data transfer.

D) StCam_GetPreviewPixelFormat

1) Call

```

BOOL StCam_GetPreviewPixelFormat (
    HANDLE   hCamera ,           // Camera Handle
    PDWORD  pdwPreviewPixelFormat // Pixel Format of Preview
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pdwPreviewPixelFormat

Set the pointer of variable that acquires the pixel format of preview.

6.0 Preview Image (Continued)

D) StCam_GetPreviewPixelFormat (Continued)

2) Parameters (Continued)

Pixel Format of Preview

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The pixel format of preview image data is acquired.

E) StCam_GetEnablePreviewPixelFormat

1) Call

```

BOOL StCam_GetEnablePreviewPixelFormat(
    HANDLE hCamera,           // Camera Handle
    PDWORD pdwEnablePreviewPixelFormat // Pixel Format of Preview
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pdwEnablePreviewPixelFormat

Set the pointer of variable. Variable is result of OR operation of pixel format of preview. This variable mean is pixel format that can be set for preview.

Note: When the Camera returns 0x00000005, the Camera supports 8 Bits and 24 Bits. When the Camera returns 0x00000009, the Camera supports 8 Bits and 32 Bits. When the Camera returns 0x00000012, the Camera supports 24 Bits and 24 Bits. When the Camera returns 0x00000013, the Camera supports 8 Bits, 24 Bits, and 32 Bits.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

6.0 Preview Image (Continued)

E) StCam_GetEnablePreviewPixelFormat (Continued)

2) Parameters (Continued)

Pixel Format of Preview

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The pixel format of preview that can be set is acquired.

F) StCam_SetColorInterpolationMethod

1) Call

```
BOOL StCam_SetColorInterpolationMethod(  
    HANDLE hCamera ,           // Camera Handle  
    BYTE byteColorInterpolationMethod // Color Interpolation Method  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) byteColorInterpolationMethod

Set the Color Interpolation Method.

6.0 Preview Image (Continued)

F) StCam_SetColorInterpolationMethod (Continued)

2) Parameters (Continued)

Color Interpolation Method

Name	Value	Description
STCAM_COLOR_INTERPOLATION_NONE_MONO	0	No Color Interpolation ProcessMonochrome Image
STCAM_COLOR_INTERPOLATION_NONE_COLOR	1	No Color Interpolation ProcessColorImage
STCAM_COLOR_INTERPOLATION_NEAREST_NEIGHBOR	2	Use Nearest Neighbor ColorColor Interpolation

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Color Interpolation Method. This function proceeds on the PC. Therefore sometimes frame rate drops by Interpolation Method.

G) StCam_GetColorInterpolationMethod

1) Call

```

BOOL StCam_GetColorInterpolationMethod(
    HANDLE hCamera,           // Camera Handle
    PBYTE pbyteColorInterpolationMethod // Color Interpolation Method
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pbyteColorInterpolationMethod

Set the pointer of variable that acquires the Color Interpolation Method.

Color Interpolation Method

Name	Value	Description
STCAM_COLOR_INTERPOLATION_NONE_MONO	0	No Color Interpolation process Monochrome Image
STCAM_COLOR_INTERPOLATION_NONE_COLOR	1	No Color Interpolation process Color Image
STCAM_COLOR_INTERPOLATION_NEAREST_NEIGHBOR	2	Use Nearest Neighbor ColorColor Interpolation

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Color Interpolation Method is acquired.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

6.0 Preview Image (Continued)

H) StCam_CreatePreviewWindowA

1) Call

```
BOOL StCam_CreatePreviewWindowA(  
    HANDLE      hCamera ,           // Camera Handle  
    PCSTR       pszWindowName ,     // Window Name  
    DWORD       dwStyle ,           // Window Style  
    LONG        lngPositionX ,      // Window Horizontal Position  
    LONG        lngPositionY ,      // Window Vertical Position  
    DWORD       dwWidth ,           // Width of Window  
    DWORD       dwHeight ,          // Height of Window  
    HWND        hWndParent ,        // Parents or Owner Window Handle  
    HMENU        hMenu ,            // Menu Handle  
    BOOL        bCloseEnable        // Window Menu Close Enable  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pszWindowName*
Set the pointer of string that has window name and "NULL". String has to finish by "NULL".
- c) *dwStyle*
Set the window style. Normally set WS_OVERLAPPEDWINDOW | WS_VISIBLE
- d) *lngPositionX*
Set the window horizontal position. Set the screen vertical position when make overlap window or popup window. Set the client horizontal position when make child window.
- e) *lngPositionY*
Set the window vertical position. Set the screen vertical position when make overlap window or popup window. Set the client vertical position when make child window.
- f) *dwWidth*
Set the width of client area of window. When set 0 to *dwWidth* and *dwHeight* window size setup with current image size.
- g) *dwHeight*
Set the height of client area of window. When set 0 to *dwWidth* and *dwHeight* window size setup with current image size.
- h) *hWndParent*
Set the Handle of parents window or owner window. Set the "NULL" when doesn't have parents window or owner window.
- i) *hMenu*
Set the menu Handle of overlap window and popup window.
- j) *bCloseEnable*
Set the Enable (TRUE) or Disable (FALSE) of window menu close button.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Create the preview window. Window made by this function can be closed by StCam_DestroyPreviewWindow function. If using Visual Basic, you must use StCam_CreatePreviewWindowA to make Preview Window.

6.0 Preview Image (Continued)

I) StCam_CreatePreviewWindowW

1) Call

```
BOOL StCam_CreatePreviewWindowW (  
    HANDLE      hCamera ,           // Camera Handle  
    PCSTR       pszWindowName ,     // Window Name  
    DWORD       dwStyle ,           // Window Style  
    LONG        lngPositionX ,      // Window Horizontal Position  
    LONG        lngPositionY ,      // Window Vertical Position  
    DWORD       dwWidth ,           // Width of Window  
    DWORD       dwHeight ,          // Height of Window  
    HWND        hWndParent ,        // Parents or Owner Window Handle  
    HMENU        hMenu ,            // Menu Handle  
    BOOL        bCloseEnable        // Window Menu Close Enable  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pszWindowName*
Set the pointer of string that has window name and "NULL". String has to finish by "NULL".
- c) *dwStyle*
Set the window style. Normally set WS_OVERLAPPEDWINDOW | WS_VISIBLE
- d) *lngPositionX*
Set the window horizontal position. Set the screen vertical position to make overlap window or popup window. Set the client horizontal position to make child window.
- e) *lngPositionY*
Set the window vertical position. Set the screen vertical position to make overlap window or popup window. Set the client vertical position to make child window.
- f) *dwWidth*
Set the width of client area of window. When set 0 to *dwWidth* and *dwHeight*, window size setup with current image size.
- g) *dwHeight*
Set the height of client area of window. When set 0 to *dwWidth* and *dwHeight*, window size setup with current image size.
- h) *hWndParent*
Set the Handle of parents window or owner window. Set the "NULL" when parents window or Owner window doesn't exist.
- i) *hMenu*
Set the Menu Handle of overlap window and popup window.
- j) *bCloseEnable*
Set the Enable (TRUE) or Disable (FALSE) of window menu close button.

6.0 Preview Image (Continued)

I

I) StCam_CreatePreviewWindow (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Create the preview window. Window made by this function can be closed by StCam_DestroyPreviewWindow function. If use Visual Basic, have to use StCam_CreatePreviewWindowA to make preview window.

J) StCam_DestroyPreviewWindow

1) Call

```
BOOL StCam_DestroyPreviewWindow(  
    HANDLE    hCamera    // Camera Handle  
);
```

2) Parameter

a) *hCamera*
Set the Camera Handle that acquired by StCam_Open function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Close the preview window made by StCam_CreatePreviewWindowA or StCam_CreatePreviewWindowW function.

K) StCam_SetPreviewWindowNameA

1) Call

```
BOOL StCam_SetPreviewWindowNameA(  
    HANDLE    hCamera ,    // Camera Handle  
    PCSTR    pszWindowName    // Window Name  
);
```

2) Parameters

a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
b) *pszWindowName*
Set the pointer of string that has window name and "NULL". String has to finish by "NULL".

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the preview window name.

6.0 Preview Image (Continued)

L) StCam_GetPreviewWindowNameA

- 1) Call

```
BOOL StCam_GetPreviewWindowNameA (  
    HANDLE    hCamera ,           // Camera Handle  
    PSTR      pszWindowName ,     // Window Name  
    LONG      lngMaxCount         // Maximum Number of String  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pszWindowName*
Set the pointer of buffer that house of file name.
 - c) *lngMaxCount*
Set the size of buffer that including "NULL" at end of buffer. *pszWindowName* is the pointer of this buffer. If text is longer than buffer size, text will be cut by buffer size.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The current preview window name is acquired.

M) StCam_SetPreviewWindowNameW

- 1) Call

```
BOOL StCam_SetPreviewWindowNameW (  
    HANDLE    hCamera ,           // Camera Handle  
    PCWSTR    pszWindowName     // Window Name  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pszWindowName*
Set the pointer of string that has window name and "NULL". String has to finish by "NULL".
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Set the preview window name. If using Visual Basic, you must use StCam_SetPreviewWindowNameA

6.0 Preview Image (Continued)

N) StCam_GetPreviewWindowNameW

- 1) Call

```
BOOL StCam_GetPreviewWindowNameW (  
    HANDLE    hCamera ,           // Camera Handle  
    PWSTR     pszWindowName ,     // Window Name  
    LONG      lngMaxCount         // Maximum Number of String  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pszWindowName*
Set the pointer of buffer that house of file name.
 - c) *lngMaxCount*
Set the size of buffer including "NULL" at end of buffer. *pszWindowName* is the pointer of this buffer. If text is longer than buffer size, text will be cut by buffer size.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The current preview window name is acquired.

O) StCam_SetPreviewMaskSize

- 1) Call

```
BOOL StCam_SetPreviewMaskSize (  
    HANDLE    hCamera ,           // Camera Handle  
    DWORD     dwOffsetX ,         // Horizontal Offset of Mask  
    DWORD     dwOffsetY ,         // Vertical Offset of Mask  
    DWORD     dwWidth ,           // Width of Mask  
    DWORD     dwHeight            // Height of Mask  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *dwOffsetX*
Set the horizontal offset of preview target image
 - c) *dwOffsetY*
Set the vertical offset of preview target image
 - d) *dwWidth*
Set the width of preview target image
 - e) *dwHeight*
Set the height of preview target image

6.0 Preview Image (Continued)

O) StCam_SetPreviewMaskSize (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the preview target image. When this function uses with StCam_SetPreviewDestSize function, part of image can be magnifies.

When enable Magnification Mode by StCam_SetMagnificationMode function this parameters will be changed by mouse operation.

Magnification function proceeds on the PC. Therefore sometimes frame rate drops by Magnification function.

P) StCam_GetPreviewMaskSize

1) Call

```
BOOL StCam_GetPreviewMaskSize (  
    HANDLE      hCamera ,      // Camera Handle  
    PDWORD     pdwOffsetX ,    // Horizontal Offset of Mask  
    PDWORD     pdwOffsetY ,    // Vertical Offset of Mask  
    PDWORD     pdwWidth ,      // Width of Mask  
    PDWORD     pdwHeight      // Height of Mask  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwOffsetX*

Set the pointer of variable that acquires the horizontal offset of preview target image

c) *pdwOffsetY*

Set the pointer of variable that acquires the vertical offset of preview target image

d) *pdwWidth*

Set the pointer of variable that acquires the width of preview target image

e) *pdwHeight*

Set the pointer of variable that acquires the height of preview target image

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The preview target image is acquired.

6.0 Preview Image (Continued)

Q) StCam_SetPreviewWindowSize

1) Call

```
BOOL StCam_SetPreviewWindowSize (  
    HANDLE      hCamera ,      // Camera Handle  
    LONG       lngPositionX ,  // Horizontal Position of Window  
    LONG       lngPositionY ,  // Vertical Position of Window  
    DWORD      dwWidth ,       // Client Width of Window  
    DWORD      dwHeight      // Client Height of Window  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *lngPositionX*
Set the horizontal position of window.
- c) *lngPositionY*
Set the vertical position of window.
- d) *dwWidth*
Set the client width of window. When set 0 to *dwWidth* and *dwHeight*, window size setup with current image size.
- e) *dwHeight*
Set the client height of window. When set 0 to *dwWidth* and *dwHeight*, window size setup with current image size.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the preview window size.

R) StCam_GetPreviewWindowSize

1) Call

```
BOOL StCam_GetPreviewWindowSize (  
    HANDLE      hCamera ,      // Camera Handle  
    PLONG       plngPositionX , // Horizontal Position of Window  
    PLONG       plngPositionY , // Vertical Position of Window  
    PDWORD      pdwWidth ,       // Client Width of Window  
    PDWORD      pdwHeight      // Client Height of Window  
);
```

6.0 Preview Image (Continued)

R) StCam_GetPreviewWindowSize (Continued)

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *plngPositionX*
Set the pointer of variable that acquires the horizontal position of window.
- c) *plngPositionY*
Set the pointer of variable that acquires the vertical position of window.
- d) *pdwWidth*
Set the pointer of variable that acquires the client width of window.
- e) *pdwHeight*
Set the pointer of variable that acquires the client height of window.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The preview window size is acquired.

S) StCam_SetPreviewWindowStyle

1) Call

```
BOOL StCam_SetPreviewWindowStyle (  
    HANDLE      hCamera , // Camera Handle  
    DWORD      dwStyle   // Window Style  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *dwStyle*
Set the preview window style.

6.0 Preview Image (Continued)

S) StCam_SetPreviewWindowStyle (Continued)

2) Parameters (Continued)

Window Style (From MSDN Library)

Window Style	Description
WS_BORDER	Creates a window that has a thin-line border.
WS_CAPTION	Creates a window that has a title bar (includes the WS_BORDERstyle).
WS_CHILD	Creates a child window. A window with this style cannot have a menu bar. This style cannot be used with the WS_POPUP style.
WS_CHILDWINDOW	Same as the WS_CHILD style.
WS_CLIPCHILDREN	Excludes the area occupied by child windows when drawing occurs within the parent window. This style is used when creating the parent window.
WS_CLIPSIBLINGS	Clips child windows relative to each other; that is, when a particular child window receives a WM_PAINT message, the WS_CLIPSIBLINGS style clips all other overlapping child windows out of the region of the child window to be updated. If WS_CLIPSIBLINGS is not specified and child windows overlap, it is possible, when drawing within the client area of a child window, to draw within the client area of a neighboring child window.
WS_DISABLED	Creates a window that is initially disabled. A disabled window cannot receive input from the user. To change this after a window has been created, use EnableWindow
WS_DLGFRAME	Creates a window that has a border of a style typically used with dialog boxes. A window with this style cannot have a title bar.
WS_GROUP	Specifies the first control of a group of controls. The group consists of this first control and all controls defined after it, up to the next control with the WS_GROUP style. The first control in each group usually has the WS_TABSTOP style so that the user can move from group to group. The user can subsequently change the keyboard focus from one control in the group to the next control in the group by using the direction keys. You can turn this style on and off to change dialog box navigation. To change this style after a window has been created, use SetWindowLong.
WS_HSCROLL	Creates a window that has a horizontal scroll bar
WS_ICONIC	Creates a window that is initially minimized. Same as the WS_MINIMIZE style.
WS_MAXIMIZE	Creates a window that is initially maximized.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

6.0 Preview Image (Continued)

S) StCam_SetPreviewWindowState (Continued)

2) Parameters (Continued)

Window Style (From MSDN Library)

Window Style	Description
WS_MAXIMIZEBOX	Creates a window that has a maximize button. Cannot be combined with the WS_EX_CONTEXTHELP style. The WS_SYSMENU style must also be specified.
WS_MINIMIZE	Creates a window that is initially minimized. Same as the WS_ICONIC style.
WS_MINIMIZEBOX	Creates a window that has a minimize button. Cannot be combined with the WS_EX_CONTEXTHELP style. The WS_SYSMENU style must also be specified.
WS_OVERLAPPED	Creates an overlapped window. An overlapped window has a title bar and a border. Same as the WS_TILED style.
WS_OVERLAPPEDWINDOW	Creates an overlapped window with the WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX, and WS_MAXIMIZEBOX styles. Same as the WS_TILEDWINDOW style.
WS_POPUP	Creates a pop-up window. This style cannot be used with the WS_CHILD style.
WS_POPUPWINDOW	Creates a pop-up window with WS_BORDER, WS_POPUP, and WS_SYSMENU styles. The WS_CAPTION and WS_POPUPWINDOW styles must be combined to make the window menu visible.
WS_SIZEBOX	Creates a window that has a sizing border. Same as the WS_THICKFRAME style.
WS_SYSMENU	Creates a window that has a window menu on its title bar. The WS_CAPTION style must also be specified.
WS_TABSTOP	Specifies a control that can receive the keyboard focus when the user presses the TAB key. Pressing the TAB key changes the keyboard focus to the next control with the WS_TABSTOP style. You can turn this style on and off to change dialog box navigation. To change this style after a window has been created, use SetWindowLong .
WS_THICKFRAME	Creates a window that has a sizing border. Same as the WS_SIZEBOX style.
WS_TILED	Creates an overlapped window. An overlapped window has a title bar and a border. Same as the WS_OVERLAPPED style.
WS_TILEDWINDOW	Creates an overlapped window with the WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX, and WS_MAXIMIZEBOX styles. Same as the WS_OVERLAPPEDWINDOW style.
WS_VISIBLE	Creates a window that is initially visible. This style can be turned on and off by using ShowWindow or SetWindowPos.
WS_VSCROLL	Creates a window that has a vertical scroll bar.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

6.0 Preview Image (Continued)

S) StCam_SetPreviewWindowState (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the preview window style.

T) StCam_GetPreviewWindowState

1) Call

```
BOOL StCam_GetPreviewWindowState (  
    HANDLE hCamera ,      // Camera Handle  
    PDWORD pdwStyle       // Window Style  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that acquired by StCam_Open function.

b) pdwStyle

Set the pointer of variable that acquires preview window style.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The current preview window style is acquired.

U) StCam_SetAspectMode

1) Call

```
BOOL StCam_SetAspectMode(  
    HANDLE hCamera ,      // Camera Handle  
    BYTE byteAspectMode   // Aspect Mode  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that acquired by StCam_Open function.

b) byteAspectMode

Set the Aspect Mode.

6.0 Preview Image (Continued)

U) StCam_SetAspectMode (Continued)

2) Parameters (Continued)

Aspect Mode

Name	Value	Description
STCAM_ASPECT_MODE_FIXED	0	Fixed Aspect Ratio
STCAM_ASPECT_MODE_KEEP_ASPECT	1	Maximum size with keep Aspect Ratio
STCAM_ASPECT_MODE_ADJUST_WINDOW	2	Adjust with window size
STCAM_ASPECT_MODE_CUSTOM	255	Custom size with StCam_SetPreviewDestSize function

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Aspect Mode when resize preview window. This function proceeds on the PC. Sometimes reduce frame rate.

V) StCam_GetAspectMode

1) Call

```

BOOL StCam_GetAspectMode (
    HANDLE hCamera ,           // Camera Handle
    PBYTE pbyteAspectMode     // Aspect Mode
);

```

2) Parameters

- hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- pbyteAspectMode*
Set the pointer of variable that acquires the Aspect Mode.

Aspect Mode

Name	Value	Description
STCAM_ASPECT_MODE_FIXED	0	Fixed Aspect Ratio
STCAM_ASPECT_MODE_KEEP_ASPECT	1	Maximum size with keep Aspect Ratio
STCAM_ASPECT_MODE_ADJUST_WINDOW	2	Adjust with window size
STCAM_ASPECT_MODE_CUSTOM	255	Custom size with StCam_SetPreviewDestSize function

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Aspect Mode when resize preview window is acquired.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

6.0 Preview Image (Continued)

W) StCam_SetPreviewDestSize

1) Call

```
BOOL StCam_SetPreviewDestSize (  
    HANDLE      hCamera ,      // Camera Handle  
    DWORD       dwOffsetX ,    // Horizontal Position of the Image  
    DWORD       dwOffsetY ,    // Vertical Position of the Image  
    DWORD       dwWidth ,      // Width of Preview Image  
    DWORD       dwHeight      // Height of Preview Image  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *dwOffsetX*
Set the horizontal position of the image on the preview window.
- c) *dwOffsetY*
Set the vertical position of the image on the preview window.
- d) *dwWidth*
Set the width of preview image.
- e) *dwHeight*
Set the height of preview image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

Set the size of preview image. If this size does not match with size of StCam_SetPreviewMaskSize, image will be magnified.
When resize preview window, position will be renewed automatically Magnification function proceeds on the PC. Therefore sometimes frame rate drops by Magnification function.

X) StCam_GetPreviewDestSize

1) Call

```
BOOL StCam_GetPreviewDestSize (  
    HANDLE      hCamera ,      // Camera Handle  
    PDWORD      pdwOffsetX ,    // Horizontal Position of the Image  
    PDWORD      pdwOffsetY ,    // Vertical Position of the Image  
    PDWORD      pdwWidth ,      // Width of Preview Image  
    PDWORD      pdwHeight      // Height of Preview Image  
);
```

6.0 Preview Image (Continued)

X) StCam_GetPreviewDestSize (Continued)

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pdwOffsetX*
Set the pointer of variable that acquires the horizontal position of the image on the preview window.
- c) *pdwOffsetY*
Set the pointer of variable that acquires the vertical position of the image on the preview window.
- d) *pdwWidth*
Set the pointer of variable that acquires the width of preview image.
- e) *pdwHeight*
Set the pointer of variable that acquires the height of preview image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The size of preview image is acquired.

Y) StCam_SetMagnificationMode

1) Call

```
BOOL StCam_SeMagnificationMode (  
    HANDLE      hCamera ,           // Camera Handle  
    BYTE        byteMagnificationMode // Magnification Mode  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *byteMagnificationMode*
Set the Magnification Mode.

Magnification Mode

Name	Value	Description
STCAM_MAGNIFICATION_MODE_OFF	0	Magnification Mode OFF
STCAM_MAGNIFICATION_MODE_ON	1	Magnification Mode ON

6.0 Preview Image (Continued)

Y) StCam_SetMagnificationMode (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Magnification Mode. When the Magnification Mode is ON, cursor of mouse changes from the arrow to the magnifying glass.

During the Magnification Mode is ON, left click then image magnifies. During the Magnification Mode is ON, press "Alt" key and left click then image shrinks.

This function proceeds on the PC. Therefore sometimes frame rate drops by this function.

Z) StCam_GetMagnificationMode

1) Call

```
BOOL StCam_GeMagnificationMode (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE      pbyteMagnificationMode // Magnification Mode  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteMagnificationMode*

Set the pointer of variable that acquires the Magnification Mode.

Magnification Mode Table

Name	Value	Description
STCAM_MAGNIFICATION_MODE_OFF	0	Magnification OFF
STCAM_MAGNIFICATION_MODE_ON	1	Magnification ON

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Magnification Mode is acquired.

7.0 Image Acquisition

A) StCam_TakeRawSnapShot

1) Call

```
BOOL StCam_TakeRawSnapShot (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE      pbyteBuffer ,       // Buffer for Image Data  
    DWORD      dwBufferSize ,      // Buffer Size  
    PDWORD     pdwNumberOfByteTrans , // Total Number of transfer Byte  
    PDWORD     pdwFrameNo ,        // Frame Number  
    DWORD      dwMilliseconds      // Time-out Time  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pbyteBuffer*
Set the pointer of buffer that acquires the image data.
- c) *dwBufferSize*
Set the total number of bytes of image data buffer. Multiple image data acquired when set more than 2 image data size.
- d) *pdwNumberOfByteTrans*
Set the pointer of variable that acquires the total number of bytes of image data.
- e) *pdwFrameNo*
Set the pointer of variable (or string) that acquires the frame number of image. When acquires multiple image, need number of string as same as number of frame.
- f) *dwMilliseconds*
Set the Time-Out time. Value unit is mill second

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The raw image data is acquired.. Buffer size can acquire by StCam_GetImageSize and StCam_GetTransferBitsPerPixel function.

When image is 8Bits each pixel, buffer needs size of [Width of Image x Height of Image]. When image is 16Bits each pixel, buffer needs size of [Width of Image x Height of Image x 2].

Color information can acquire by StCam_GetColorArray function.

7.0 Image Acquisition (Continued)

B) StCam_TakePreviewSnapShot

1) Call

```
BOOL StCam_TakePreviewSnapShot(  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE       pbyteBuffer ,       // Buffer for Image Data  
    DWORD       dwBufferSize ,      // Buffer Size  
    PDWORD      pdwNumberOfByteTrans , // Total Number of transfer Byte  
    PDWORD      pdwFrameNo ,        // Frame Number  
    DWORD       dwMilliseconds      // Time-out Time  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteBuffer*

Set the pointer of buffer that acquires the image data.

c) *dwBufferSize*

Set the total number of bytes of image data buffer. Multiple image data acquire when set more than 2 image data size.

d) *pdwNumberOfByteTrans*

Set the pointer of variable that acquires the total number of bytes of image data.

e) *pdwFrameNo*

Set the pointer of variable (or string) that acquires the frame number of image. When acquires multiple image, need number of string as same as number of frame.

f) *dwMilliseconds*

Set the Time-Out time. Value unit is mill second

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Image data before preview is acquired. Buffer size can be acquired by StCam_GetImageSize and StCam_GetPreviewPixelFormat function.

When image is 8 Bits each pixel, buffer needs size of [Width of Image x Height of Image]. When image is 24 Bits each pixel, buffer needs size of [Width of Image x Height of Image x 3].

7.0 Image Acquisition (Continued)

C) StCam_SaveImageA

1) Call

```

BOOL StCam_SaveImageA(
    HANDLE    hCamera ,           // Camera Handle
    DWORD     dwWidth ,           // Width of Image
    DWORD     dwHeight ,          // Height of Image
    DWORD     dwPreviewPixelFormat , // Pixel Format of Preview
    PBYTE     pbyteData ,         // Buffer for Image Data
    PCSTR     pszFileName ,       // File Name
    DWORD     dwParam             // Parameter
);

```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *dwWidth*
Set the width of image.
- c) *dwHeight*
Set the height of image.
- d) *dwPreviewPixelFormat*
Set the pixel format of preview image data.

Pixel Format of Preview Image Data

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel Data (Order by Blue, Green, Red)stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red)stores to the buffer from the upper left of the image.

- e) *pbyteData*
Set the pointer of buffer that has image data.
- f) *pszFileName*
Set the pointer of strings that has file name and "NULL". Strings have to finish by "NULL". File format will be changed by extension (bmp, jpg, tif, png, raw)
- g) *dwParam*
This parameter enables when save JPEG file. Set the Compression rate of JPEG file. The value range is 0 to 100 (0: Low Compression, 100: High Compression). Set 0 when save file except JPEG.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
 Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

7.0 Image Acquisition (Continued)

C) StCam_SaveImageA (Continued)

4) Description

Save acquiring image data to the file. Save file format can select from BMP, JPEG, TIFF, PNG, and RAW. Save the image data that acquire by StCam_TakePreviewSnapShot, StCam_TakeRawSnapShot, PreviewBitmap, and PreviewRaw functions. IJG Code uses to make JPEG file.

D) StCam_SaveImageW

1) Call

```

BOOL StCam_SaveImageW(
    HANDLE      hCamera ,           // Camera Handle
    DWORD       dwWidth ,           // Width of Image
    DWORD       dwHeight ,          // Height of Image
    DWORD       dwPreviewPixelFormat , // Pixel Format of Preview
    PBYTE       pbyteData ,         // Buffer for Image Data
    PCSTR       pszFileName ,       // File Name
    DWORD       dwParam             // Parameter
);
    
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *dwWidth*
Set the width of image.
- c) *dwHeight*
Set the height of image.
- d) *dwPreviewPixelFormat*
Set the pixel format of preview image data.

Pixel Format of Preview Image Data

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel Data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.

e) *pbyteData*

Set the pointer of buffer that has image data.

f) *pszFileName*

Set the pointer of strings that has file name and "NULL". Strings have to finish by "NULL". File format will be changed by extension (bmp, jpg, tif, png, raw)

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

7.0 Image Acquisition (Continued)

D) StCam_SaveImageW (Continued)

2) Parameters (Continued)

g) *dwParam*

This parameter enables when saving JPEG file. Set the Compression rate of JPEG file. The value range is 0 to 100 (0: Low Compression, 100: High Compression). Set 0 when save file except JPEG.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Save acquiring image data to file. Save file format can select from BMP, JPEG, TIFF, PNG, and RAW. Save the image data that acquire by StCam_TakePreviewSnapShot, StCam_TakeRawSnapShot, PreviewBitmap, and PreviewRaw functions. IJG Code uses to make JPEG file. If use Visual Basic, have to use StCam_SaveImageA to save Image to File.

8.0 Shutter and Gain

A) StCam_SetShutterSpeed

1) Call

```
BOOL StCam_SetShutterSpeed (  
    HANDLE    hCamera ,           // Camera Handle  
    WORD      wShutterLine ,      // Shutter Speed (Number of Lines)  
    WORD      wShutterClock       // Shutter Speed (Number of Clocks)  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *wShutterLine*

Set the shutter speed (Number of lines). The value range is 0 to ("Number of lines each frame" - 1).

c) *wShutterClock*

Set the shutter speed (Number of clocks). The value range is 0 to ("Number of clocks each line" - 1).

8.0 Shutter and Gain (Continued)

A) StCam_SetShutterSpeed (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the shutter speed. Shutter speed set by number of lines and number of clocks.

Shutter Speed

Shutter Speed [second]	Value	Total Number of Clock
Total number of clock / Clock frequency	Number of line = Number of clock = 0	(Number of lines each frame) x (Number of clocks each line)
	Other	(wshutterLine) x (Number of clocks each line) + (wShutterClock)

Number of lines each frame and number of clocks each line acquire by StCam_GetFrameClock function. Clock frequency acquire by StCam_GetClock function.

B) StCam_GetShutterSpeed

1) Call

```

BOOL StCam_GetShutterSpeed (
    HANDLE    hCamera ,           // Camera Handle
    PWORD     pwShutterLine ,     // Shutter Speed (Number of Lines)
    PWORD     pwShutterClock     // Shutter Speed (Number of Clocks)
);
    
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pwShutterLine*
Set the pointer of variable that acquires the shutter speed (Number of lines). The value range is 0 to (Number of lines each frame - 1).
- c) *pwShutterClock*
Set the pointer of variable that acquires the shutter speed (Number of clocks). The value range is 0 to ("Number of clocks each line" - 1).

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The shutter speed is acquired. Shutter speed sets by number of lines and number of clocks.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
 Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

8.0 Shutter and Gain (Continued)

B) StCam_GetShutterSpeed (Continued)

4) Description (Continued)

Shutter Speed

Shutter Speed [second]	Value	Total Number of Clock
Total number of clock / Clock frequency	Number of line = Number of clock = 0	(Number of lines each frame) x (Number of clocks each line)
	Other	(wshutterLine) x (Number of clocks each line) + (wShutterClock)

Number of lines each frame and number of clocks each line acquire by StCam_GetFrameClock function. Clock frequency acquire by StCam_GetClock function.

C) StCam_SetGain

1) Call

```

BOOL StCam_SetGain (
    HANDLE      hCamera , // Camera Handle
    WORD        wGain     // Gain
);

```

2) Parameters

a) hCamera

Set the Camera Handle that IS acquired by StCam_Open function.

b) wGain

Set the Gain value. The value range is 0 to 255 (Factory Default is 128.). Gain formula is following:

$$\text{Gain [dB]} = 2.36 \text{ dB} + 0.132 \text{ dB} \times w\text{Gain}$$

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Gain.

D) StCam_GetGain

1) Call

```

BOOL StCam_GetGain (
    HANDLE      hCamera , // Camera Handle
    PWORD       pwGain     // Gain
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pwGain

Set the pointer of variable that acquires the Gain value

8.0 Shutter and Gain (Continued)

D) StCam_GetGain (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Gain is acquired.

E) StCam_SetALCMode

1) Call

```
BOOL StCam_SetALCMode (
    HANDLE hCamera, // Camera Handle
    BYTE byteAlcMode // ALC Mode
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) byteAlcMode

Set the ALC Mode.

ALC Mode

Name	Value	Description
STCAM_ALCMODE_ALC_FIXED_AGC_OFF	0	Fixed shutter & Manual Gain
STCAM_ALCMODE_ALC_FULLAUTO_AGC_ON	1	ALC & AGC
STCAM_ALCMODE_ALC_FULLAUTO_AGC_OFF	2	ALC & Manual Gain
STCAM_ALCMODE_ALC_FIXED_AGC_ON	3	Fixed shutter & AGC
STCAM_ALCMODE_ALCAGC_ONESHOT	4	One-Shot ALC & AGC(When reach the target brightness, ALCMode changes to the fixed shutter & manual Gain)
STCAM_ALCMODE_ALC_ONESHOT_AGC_OFF	5	One-Shot ALC & Manual Gain (When reach the target brightness, ALC Mode changes to the fixed shutter & manual Gain)
STCAM_ALCMODE_ALC_FIXED_AGC_ONESHOT	6	Fixed shutter & On-Shot (When reach the target brightness, ALC Mode changes to the fixed shutter & manual Gain)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the ALC Mode.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

8.0 Shutter and Gain (Continued)

F) StCam_GetALCMode

1) Call

```

    BOOL StCam_GetALCMode (
        HANDLE      hCamera ,      // Camera Handle
        PBYTE      pbyteAlcMode // ALC Mode
    );

```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteAlcMode*

Set the pointer of variable that acquires the ALC Mode.

ALC Mode

Name	Value	Description
STCAM_ALCMODE_ALC_FIXED_AGC_OFF	0	Fixed shutter & Manual Gain
STCAM_ALCMODE_ALC_FULLAUTO_AGC_ON	1	ALC & AGC
STCAM_ALCMODE_ALC_FULLAUTO_AGC_OFF	2	ALC & Manual Gain
STCAM_ALCMODE_ALC_FIXED_AGC_ON	3	Fixed shutter & AGC
STCAM_ALCMODE_ALCAGC_ONESHOT	4	One-Shot ALC & AGC(When reach the target brightness, ALC Mode changes to the fixed shutter & manual Gain)
STCAM_ALCMODE_ALC_ONESHOT_AGC_OFF	5	One-Shot ALC & Manual Gain (When reach the target brightness, ALC Mode changes to the fixed shutter & manual Gain)
STCAM_ALCMODE_ALC_FIXED_AGC_ONESHOT	6	Fixed Shutter & On-Shot (When reach the target brightness, ALC Mode changes to the fixed shutter & manual Gain)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The ALC Mode is acquired.

8.0 Shutter and Gain (Continued)

G) StCam_SetTargetBrigtness

- 1) Call

```
BOOL StCam_SetTargetBrigtness (  
    HANDLE    hCamera ,           // Camera Handle  
    BYTE      byteTargetBrigtness , // Brightness Target  
    BYTE      byteTolerance ,      // Brightness Tolerance  
    BYTE      byteThreshold       // Brightness Threshold  
);
```
- 2) Parameters
 - a) *hcamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *byteTargetBrigtness*
Set the target Brightness of ALC
 - c) *byteTolerance*
Set the tolerance of deactivates ALC. When (*byteTargetBrigtness* – Current Brightness) becomes smaller than *ByteTolerance* then deactivates ALC.
 - d) *byteThreshold*
Set the threshold of activates ALC. When Current Brightness becomes bigger than (*byteTargetBrigtness* + *byteThreshold*) then activates ALC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Set the Brightness target.

H) StCam_GetTargetBrigtness

- 1) Call

```
BOOL StCam_GetTargetBrigtness (  
    HANDLE    hCamera ,           // Camera Handle  
    PBYTE     pbyteTargetBrigtness , // Brightness Target  
    PBYTE     pbyteTolerance ,      // Brightness Tolerance  
    PBYTE     pbyteThreshold       // Brightness Threshold  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteTargetBrigtness*
Set the pointer of variable that acquires the target Brightness of ALC
 - c) *pbyteTolerance*
Set the pointer of variable that acquires the tolerance of deactivates ALC.
 - d) *pbyteThreshold*
Set the pointer of variable that acquires the threshold of activates ALC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

8.0 Shutter and Gain (Continued)

H) StCam_GetTargetBrigtness (Continued)

- 4) Description
The Brightness target is acquired.

I) StCam_SetALCWeight

- 1) Call

```
BOOL StCam_SetALCWeight (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE      pbyteALCWeight      // Brightness Weight  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteALCWeight*
Set the pointer of string for 16 bytes that acquires the Brightness weight of 16 areas of ALC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Set the weight of Brightness of ALC. Each area size can change by StCam_SetControlArea function.

J) StCam_GetALCWeight

- 1) Call

```
BOOL StCam_GetALCWeight (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE      pbyteALCWeight      // Brightness Weight  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteALCWeight*
Set the pointer of string for 16 bytes that acquires the Brightness weight of 16 areas of ALC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The weight of Brightness of ALC is acquired.. Each area size can change by StCam_SetControlArea function.

8.0 Shutter and Gain (Continued)

K) StCam_SetShutterControlRange

1) Call

```

BOOL StCam_SetShutterControlRange (
    HANDLE   hCamera ,           // Camera Handle
    WORD     wMinShutterLine ,   // Min. Shutter Speed (Number of Lines)
    WORD     wMinShutterClock ,  // Min. Shutter Speed (Number of Lines)
    WORD     wMaxShutterLine ,   // Max. Shutter Speed (Number of Clocks)
    WORD     wMaxShutterClock   // Max. Shutter Speed (Number of Clocks)
);
    
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) wMinShutterLine

Set the minimum shutter speed (Number of lines) of ALC. No minimum shutter speed when set 0 to wMinShutterLine and set 1 to wMinShutterClock

c) wMinShutterClock

Set the minimum shutter speed (Number of clocks) of ALC. No minimum shutter speed when set 0 to wMinShutterLine and set 1 to wMinShutterClock

d) wMaxShutterLine

Set the maximum shutter speed (Number of lines) of ALC. No maximum shutter speed when set 0 to wMinShutterLine and wMinShutterClock

e) wMaxShutterClock

Set the maximum shutter speed (Number of clocks) of ALC. No maximum shutter speed when set 0 to wMinShutterLine and wMinShutterClock

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the shutter speed range of ALC. Shutter speed set by number of lines and number of clocks.

Shutter Speed

Shutter Speed [second]	Value	Total Number of Clock
Total number of clock / Clock	Number of line = Number of clock = 0	(Number of lines each frame) x frequency (Number of clocks each line)
	Other	(wshutterLine) x (Number of clocks each line) + (wShutterClock)

Number of lines each frame and number of clocks each line acquire by StCam_GetFrameClock function. Clock frequency acquire by StCam_GetClock function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

8.0 Shutter and Gain (Continued)

L) StCam_GetShutterControlRange

1) Call

```

BOOL StCam_GetShutterControlRange (
    HANDLE      hCamera ,           // Camera Handle
    PWORD       pwMinShutterLine ,  // Min. Shutter Speed (Number of Lines)
    PWORD       pwMinShutterClock , // Min. Shutter Speed (Number of Clocks)
    PWORD       pwMaxShutterLine ,  // Max. Shutter Speed (Number of Lines)
    PWORD       pwMaxShutterClock , // Max. Shutter Speed (Number of Clocks)
);
    
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pwMinShutterLine*
Set the pointer of variable that acquires the minimum shutter speed (Number of lines) of ALC
- c) *pwMinShutterClock*
Set the pointer of variable that acquires the minimum shutter speed (Number of clocks) of ALC
- d) *pwMaxShutterLine*
Set the pointer of variable that acquires the maximum shutter speed (Number of lines) of ALC
- e) *pwMaxShutterClock*
Set the pointer of variable that acquires the maximum shutter speed (Number of lines) of ALC

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The shutter speed range of ALC is acquired.. Shutter speed set by number of lines and number of clocks.

Shutter Speed

Shutter Speed [second]	Value	Total Number of Clock
Total number of clock / Clock frequency	Number of line = Number of clock = 0	(Number of lines each frame) x (Number of clocks each line)
	Other	(wshutterLine) x (Number of clocks each line) + (wShutterClock)

Number of lines each frame and number of clocks each line acquire by StCam_GetFrameClock function.
Clock frequency acquire by StCam_GetClock function.

8.0 Shutter and Gain (Continued)

M) StCam_SetGainControlRange

- 1) Call

```
BOOL StCam_SetGainControlRange (  
    HANDLE    hCamera , // Camera Handle  
    WORD      wMinGain , // AGC Minimum Gain  
    WORD      wMaxGain // AGC Maximum Gain  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *wMinGain*
Set the minimum Gain of AGC.
 - c) *wMaxGain*
Set the maximum Gain of AGC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.
- 4) Description
Set the Gain range of AGC.

N) StCam_GetGainControlRange

- 1) Call

```
BOOL StCam_GetGainControlRange (  
    HANDLE    hCamera , // Camera Handle  
    PWORD     pwMinGain , // AGC Minimum Gain  
    PWORD     pwMaxGain , // AGC Maximum Gain  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pwMinGain*
Set the pointer of variable that acquires the minimum Gain of AGC.
 - c) *pwMaxGain*
Set the pointer of variable that acquires the maximum Gain of AGC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.
- 4) Description
The Gain range of AGC is acquired.

8.0 Shutter and Gain (Continued)

O) StCam_SetALCControlSpeed

1) Call

```
BOOL StCam_SetALCControlSpeed (
    HANDLE hCamera ,           // Camera Handle
    BYTE byteShutterCtrlSpeedLimit , // Limit of Shutter Control Speed
    BYTE byteGainCtrlSpeedLimit ,   // Limit of Gain Control Speed
    BYTE byteSkipFrameCount ,       // Number of Skip Frame
    BYTE byteAverageFrameCount      // Number of Frame for Control
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteShutterCtrlSpeedLimit*

Set the limit of the shutter control speed of ALC. When the smaller number sets to this value, the control speed becomes slow. When the bigger number sets to this value, the control speed becomes fast.

c) *byteGainCtrlSpeedLimit*

Set the limit of the Gain control speed of ALC. When the smaller number sets to this value, the control speed becomes slow. When the bigger number sets to this value, the control speed becomes fast.

d) *byteSkipFrameCount*

Set the number of frame to control ALC / AGC function. If set 3 to the *byteSkipFrameCount*, ALC / AGC function maintain Brightness each 3 frames. The value range is 3 to 255.

e) *byteAverageFrameCount*

Set the number of the frame for calculates Brightness of ALC and AGC.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the ALC control speed.

8.0 Shutter and Gain (Continued)

P) StCam_GetALCControlSpeed

- 1) Call

```
    BOOL StCam_GetALCControlSpeed (
        HANDLE    hCamera ,           // Camera Handle
        PBYTE     pbyteShutterCtrlSpeedLimit , // Limit of Shutter Control Speed
        PBYTE     pbyteGainCtrlSpeedLimit ,   // Limit of Gain Control Speed
        PBYTE     pbyteSkipFrameCount ,       // Number of Skip Frame
        PBYTE     pbyteAverageFrameCount     // Number of Frame for Control
    );
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteShutterCtrlSpeedLimit*
Set the pointer of variable that acquires the limit of shutter control speed of ALC.
 - c) *pbyteGainCtrlSpeedLimit*
Set the pointer of variable that acquires the limit of Gain control speed of ALC.
 - d) *pbyteSkipFrameCount*
Set the pointer of variable that the number of frame to control ALC / AGC function.
 - e) *pbyteAverageFrameCount*
Set the pointer of variable that acquires the number of frame for calculating Brightness of ALC and AGC.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The ALC control speed is acquired.

9.0 White Balance

A) StCam_SetWhiteBalanceMode

- 1) Call

```
    BOOL StCam_SetWhiteBalanceMode (
        HANDLE    hCamera ,           // Camera Handle
        BYTE      byteWBMode          // White Balance Mode
    );
```

9.0 White Balance (Continued)

A) StCam_SetWhiteBalanceMode (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteWBMode*

Set the White Balance Mode

White Balance Mode

Name	Value	Description
STCAM_WB_OFF	0	White Balance OFF
STCAM_WB_MANUAL	1	Manual White Balance
STCAM_WB_FULLAUTO	2	Auto White Balance
STCAM_WB_ONESHOT	3	One-Shot White Balance(When reach the target White Balance, White BalanceMode changes to the Manual White Balance)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

Set the White Balance Mode.

B) StCam_GetWhiteBalanceMode

1) Call

```
BOOL StCam_GetWhiteBalanceMode (  
    HANDLE      hCamera ,           // Camera Handle  
    BYTE        pbyteWBMode        // White Balance Mode  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteWBMode*

Set the pointer of variable that acquires White Balance Mode.

9.0 White Balance (Continued)

B) StCam_GetWhiteBalanceMode (Continued)

2) Parameters (Continued)

White Balance Mode

Name	Value	Description
STCAM_WB_OFF	0	White Balance OFF
STCAM_WB_MANUAL	1	Manual White Balance
STCAM_WB_FULLAUTO	2	Auto White Balance
STCAM_WB_ONESHOT	3	One-Shot White Balance(When reach the target White Balance, White BalanceMode changes to the Manual White Balance)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Current White Balance Mode is acquired.

C) StCam_SetWhiteBalanceGain

1) Call

```

BOOL StCam_SetWhiteBalanceGain (
    HANDLE hCamera ,    // Camera Handle
    WORD wWBGainR ,    // White Balance R Gain
    WORD wWBGainGr ,   // White Balance Gr Gain
    WORD wWBGainGb ,   // White Balance Gb Gain
    WORD wWBGainB      // White Balance B Gain
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) wWBGainR

Set the R Gain of White Balance. The value range is 128 to 640. (128: 1x Gain, 640: 5x Gain)

c) wWBGainGr

Set the Gr Gain of White Balance. The value range is 128 to 640. (128: 1x Gain, 640: 5x Gain)

9.0 White Balance (Continued)

C) StCam_SetWhiteBalanceGain (Continued)

2) Parameters (Continued)

d) *wWBGainGb*

Set the Gb Gain of White Balance. The value range is 128 to 640. (128: 1x Gain, 640: 5x Gain)

e) *wWBGainB*

Set the B Gain of White Balance. The value range is 128 to 640. (128: 1x Gain, 640: 5x Gain)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the White Balance Gain.

D) StCam_GetWhiteBalanceGain

1) Call

```
BOOL StCam_GetWhiteBalanceGain (  
    HANDLE      hCamera ,           // Camera Handle  
    PWORD       pwWBGainR ,         // White Balance R Gain  
    PWORD       pwWBGainGr ,        // White Balance Gr Gain  
    PWORD       pwWBGainGb ,        // White Balance Gb Gain  
    PWORD       pwWBGainB           // White Balance B Gain  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pwWBGainR*

Set the pointer of variable that acquires the R Gain of White Balance.

c) *pwWBGainGr*

Set the pointer of variable that acquires the Gr Gain of White Balance.

d) *pwWBGainGb*

Set the pointer of variable that acquires the Gb Gain of White Balance.

e) *pwWBGainB*

Set the pointer of variable that acquires the B Gain of White Balance.

9.0 White Balance (Continued)

D) StCam_GetWhiteBalanceGain (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The current White Balance Gain is acquired.

E) StCam_SetWhiteBalanceTarget

1) Call

```
BOOL StCam_SetWhiteBalanceTarget (  
    HANDLE      hCamera ,      // Camera Handle  
    WORD        wAWBTargetR ,  // R Target of Auto White Balance  
    WORD        wAWBTargetB   // B Target of Auto White Balance  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) wAWBTargetR

Set the R axis target of Auto White Balance. The value range is 0 to 512 (Factory Default is 100). When the smaller number sets to this value, image will be more Cyan image. When the bigger number sets to this value, image will be more Red image.

c) wAWBTargetB

Set the B axis target of Auto White Balance. The value range is 0 to 512 (Factory Default is 100). When the smaller number sets to this value, image will be a more Yellow image. When the bigger number sets to this value, image will be a more Blue image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the target of Auto White Balance.

9.0 White Balance (Continued)

F) StCam_GetWhiteBalanceTarget

1) Call

```
BOOL StCam_GetWhiteBalanceTarget (  
    HANDLE      hCamera ,           // Camera Handle  
    PWORD      pwAWBTargetR ,       // R Target of Auto White Balance  
    PWORD      pwAWBTargetB        // B Target of Auto White Balance  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pwAWBTargetR*
Set the pointer of variable that acquires the R axis target of Auto White Balance.
- c) *pwAWBTargetB*
Set the pointer of variable that acquires the B axis target of Auto White Balance.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Current target of Auto White Balance is acquired.

G) StCam_SetWhiteBalanceToleranceThreshold

1) Call

```
BOOL StCam_SetWhiteBalanceToleranceThreshold (  
    HANDLE      hCamera ,           // Camera Handle  
    WORD        wAWBTolerance ,     // Tolerance of Auto White Balance  
    WORD        wAWBThreshold       // Threshold of Auto White Balance  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *wAWBTolerance*
Set the tolerance of Auto White Balance. When (Current average Color – Target Auto White Balance) in RGB space becomes smaller than *wAWBTolerance* then deactivates Auto White Balance. Target Auto White Balance can acquire by StCam_GetWhiteBalanceTarget function.
- c) *wAWBThreshold*
Set the threshold of Auto White Balance. When (Current average Color – Target Auto White Balance) in RGB space becomes bigger than *wAWBThreshold* then activates Auto White Balance. Target Auto White Balance can acquire by StCam_GetWhiteBalanceTarget function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

9.0 White Balance (Continued)

G) StCam_SetWhiteBalanceToleranceThreshold (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the tolerance and threshold of Auto White Balance.

H) StCam_GetWhiteBalanceToleranceThreshold

1) Call

```
BOOL StCam_GetWhiteBalanceToleranceThreshold (  
    HANDLE    hCamera ,           // Camera Handle  
    PWORD     pwAWBTolerance ,    // Tolerance of Auto White Balance  
    PWORD     pwAWBThreshold      // Threshold of Auto White Balance  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pwAWBTolerance*

Set the pointer of variable that acquires the tolerance of Auto White Balance.

c) *pwAWBThreshold*

Set the pointer of variable that acquires the threshold of Auto White Balance.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Current tolerance and threshold of Auto White Balance are acquired.

I) StCam_SetAWBWeight

1) Call

```
BOOL StCam_SetAWBWeight (  
    HANDLE    hCamera ,           // Camera Handle  
    PBYTE     pbyteAWBWeight      // Weight of Auto White Balance  
);
```

9.0 White Balance (Continued)

I) StCam_SetAWBWeight (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteAWBWeight*

Set the pointer of string for 16 bytes that acquires the Brightness weight of 16 areas of ALC.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the weight of 16 areas of Auto White Balance.

J) StCam_GetAWBWeight

1) Call

```
BOOL StCam_SetAWBWeight (  
    HANDLE   hCamera ,           // Camera Handle  
    PBYTE    pbyteAWBWeight      // Weight of Auto White Balance  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteAWBWeight*

Set the pointer of string for 16 bytes that acquires the current weight of 16 areas of Auto White Balance.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Current weight of 16 areas of Auto White Balance is acquired

10.0 Gamma

A) StCam_SetGammaMode

1) Call

```

BOOL StCam_SetGammaMode (
    HANDLE    hCamera ,           // Camera Handle
    BYTE      byteGammaTarget ,   // Gamma Target
    BYTE      byteGammaMode ,     // Gamma Mode
    WORD      wGamma ,           // Gamma Value
    PBYTE     pbyteGammaTable    // Gamma Table
);
    
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteGammaTarget*

Set the Gamma target.

Gamma Target

Name	Value	Description
STCAM_GAMMA_TARGET_Y	0	Gamma Setup for Brightness
STCAM_GAMMA_TARGET_R	1	Gamma Setup for R only
STCAM_GAMMA_TARGET_GR	2	Gamma Setup for Gr only
STCAM_GAMMA_TARGET_GB	3	Gamma Setup for Gb only
STCAM_GAMMA_TARGET_B	4	Gamma Setup for B only

c) *byteGammaMode*

Set the Gamma Mode

Gamma Mode

Name	Value	Description
STCAM_GAMMA_OFF	0	Gamma Correct OFF
STCAM_GAMMA_ON	1	Gamma Correct ON
STCAM_GAMMA_REVERSE	2	Gamma Correct ON (Invert Brightness)
STCAM_GAMMA_TABLE	255	Gamma Correct ON (Use user Table)

d) *wGamma*

Set the Gamma Value. The value range is 1 to 500. (Factory default is 100) (1: Gamma = 0.01, 100: Gamma = 1, 500: Gamma = 5). This value is only activated when Gamma Mode is chosen STCAM_GAMMA_On or STCAM_GAMMA_REVERSE

e) *pbyteGammaTable*

Set the pointer of string (BYTE *pbyteGammaTable*[256]) that has a Table value for use with Gamma convert. This value is only activated when Gamma Mode is chosen STCAM_GAMMA_TABLE. When in STCAM_GAMMA_OFF, STCAM_GAMMA_ON, STCAM_GAMMA_REVERSE software must be set to "NULL".

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

10.0 Gamma

A) StCam_SetGammaMode

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Gamma setting. This function proceeds on the PC. Therefore sometimes frame rate drops by Gamma Mode.

B) StCam_GetGammaMode

1) Call

```
BOOL StCam_GetGammaMode (  
    HANDLE hCamera ,           // Camera Handle  
    BYTE byteGammaTarget ,     // Gamma Target  
    PBYTE pbyteGammaMode ,     // Gamma Mode  
    PWORD pwGamma ,           // Gamma Value  
    PBYTE pbyteGammaTable     // Gamma Table  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteGammaTarget*

Set the pointer of variable that acquires the Gamma Target.

Gamma Target

Name	Value	Description
STCAM_GAMMA_TARGET_Y	0	Gamma Setup for Brightness
STCAM_GAMMA_TARGET_R	1	Gamma Setup for R only
STCAM_GAMMA_TARGET_GR	2	Gamma Setup for Gr only
STCAM_GAMMA_TARGET_GB	3	Gamma Setup for Gb only
STCAM_GAMMA_TARGET_B	4	Gamma Setup for B only

c) *pbyteGammaMode*

Set the pointer of variable that acquires the Gamma Mode.

Gamma Mode

Name	Value	Description
STCAM_GAMMA_OFF	0	Gamma Correct OFF
STCAM_GAMMA_ON	1	Gamma Correct ON
STCAM_GAMMA_REVERSE	2	Gamma Correct ON(Invert Brightness)
STCAM_GAMMA_TABLE	255	Gamma Correct ON(Use user Table)

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

10.0 Gamma (Continued)

2) Parameters

B) StCam_GetGammaMode (Continued)

d) *pwGamma*

Acquire the Gamma value. When Gamma Mode chose STCAM_GAMMA_TABLE invalid value is return.

e) *pbyteGammaTable*

Set the pointer of string (BYTE *pbyteGammaTable*[256]) that have table value use with Gamma convert

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Gamma setting is acquired.

11.0 Sharpness

A) StCam_SetSharpnessMode

1) Call

```
BOOL StCam_SetSharpnessMode (  
    HANDLE    hCamera ,           // Camera Handle  
    BYTE      byteSharpnessMode , // Sharpness Mode  
    WORD      wSharpnessGain ,    // Sharpness Gain  
    BYTE      byteSharpnessCoring // Sharpness Coring  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteSharpnessMode*

Set the sharpness Mode.

Sharpness Mode

Name	Value	Description
STCAM_SHARPNESS_OFF	0	Sharpness OFF
STCAM_SHARPNESS_ON	1	Sharpness ON

11.0 Sharpness

A) StCam_SetSharpnessMode (Continued)

2) Parameters

c) *wSharpnessGain*

Set the sharpness Gain value. The value range is 0 to 500. The value range is 0 to 500. (Factory default is 0) When the value increases the Sharpness increases.

d) *byteSharpnessCoring*

Set the sharpness Coring value. The value range is 0 to 255. (Factory default is 0) When the value increases, sharpness of low edge (including Noise) is suppressed.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the sharpness setting. This function proceeds on the PC. Therefore sometimes frame rate drops by Sharpness Mode.

B) StCam_GetSharpnessMode

1) Call

```
BOOL StCam_GetSharpnessMode (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE       pbyteSharpnessMode , // Sharpness Mode  
    PWORD       pwSharpnessGain ,   // Sharpness Gain  
    PBYTE       pbyteSharpnessCoring // Sharpness Coring  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteSharpnessMode*

Set the pointer of variable that acquires the Sharpness Mode.

Sharpness Mode

Name	Value	Description
STCAM_SHARPNESS_OFF	0	Sharpness OFF
STCAM_SHARPNESS_ON	1	Sharpness ON

11.0 Sharpness (Continued)

B) StCam_GetSharpnessMode (Continued)

2) Parameters (Continued)

c) *pwSharpnessGain*

Set the pointer of variable that acquires the sharpness Gain.

d) *pbyteSharpnessCoring*

Set the pointer of variable that acquires the sharpness Coring.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

The sharpness setting is acquired.

12.0 Hue and Saturation

A) StCam_SetHueSaturation

1) Call

```
BOOL StCam_SetSharpnessMode (  
    HANDLE hCamera ,           // Camera Handle  
    BYTE byteHueSaturationMode , // Hue & Saturation Mode  
    SHORT shtHue ,             // Hue  
    WORD wSaturation            // Saturation  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteHueSaturationMode*

Set the Hue & Saturation Mode.

Hue & Saturation Mode

Name	Value	Description
STCAM_HUE_SATURATION_OFF	0	Hue & Saturation OFF
STCAM_HUE_SATURATION_ON	1	Hue & Saturation ON

c) *sthHue*

Set the Hue. The value range is - 1800 to 1800. (-1800: -180 deg., 1800: 180 deg.) When set the 0 to *sthHue*, Hue procedure is OFF.

d) *wSaturation*

Set the Saturation. The value range is 0 to 200. When set the 0 to *wSaturation*, Image will be Monochrome Image.

12.0 Hue and Saturation

A) StCam_SetHueSaturation

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Hue & Saturation setting. This function proceeds on the PC. Therefore sometimes frame rate drops by Hue & Saturation Mode.

B) StCam_GetHueSaturation

1) Call

```
BOOL StCam_GetSharpnessMode (  
    HANDLE      hCamera ,           // Camera Handle  
    PBYTE      pbyteHueSaturationMode , // Hue & Saturation Mode  
    PSHORT     pshtHue ,           // Hue  
    PWORD      pwSaturation         // Saturation  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that acquired by StCam_Open function.

b) *pbyteHueSaturationMode*

Set the pointer of variable that acquires the Hue & Saturation Mode.

Hue & Saturation Mode

Name	Value	Description
STCAM_HUE_SATURATION_OFF	0	Hue & Saturation OFF
STCAM_HUE_SATURATION_ON	1	Hue & Saturation ON

c) *pshtHue*

Set the pointer of variable that acquires the Hue.

d) *pwSaturation*

Set the pointer of variable that acquires the Saturation.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Hue & Saturation setting are acquired.

13.0 Color Matrix

A) StCam_SetColorMatrix

1) Call

```

BOOL StCam_SetColorMatrix (
    HANDLE    hCamera ,           // Camera Handle
    BYTE      byteColorMatrixMode , // Color Matrix Mode
    PSHORT    pshtColorMatrix     // Color Matrix
);
    
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that acquired by StCam_Open function.
- b) *byteColorMatrixMode*
Set the Color Matrix Mode.

Color Matrix Mode

Name	Value	Description
STCAM_COLOR_MATRIX_OFF	0x00	OFF
STCAM_COLOR_MATRIX_CUSTOM	0xFF	Custom (Use Color Matrix)

- c) *pshtColorMatrix*
Set the pointer of string (SHORT *pshtColorMatrix*[12]) that acquires the Color Matrix

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Color Matrix. Relationship between R'B'G(data after procedure) and RGB(data Before procedure) with m (Color Matrix).

$$\begin{aligned}
 R' &= (Rxm[0] + Gxm[1] + Bxm[2] + m[3]) / 100 \\
 G' &= (Rxm[4] + Gxm[5] + Bxm[6] + m[7]) / 100 \\
 B' &= (Rxm[8] + Gxm[9] + Bxm[10] + m[11]) / 100
 \end{aligned}$$

This function proceeds on the PC. Therefore sometimes frame rate drops by Color Matrix Mode.

B) StCam_GetColorMatrix

1) Call

```

BOOL StCam_GetColorMatrix (
    HANDLE    hCamera ,           // Camera Handle
    PBYTE      pbyteColorMatrixMode , // Color Matrix Mode
    PSHORT    pshtColorMatrix     // Color Matrix
);
    
```

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
 Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

13.0 Color Matrix (Continued)

B) StCam_GetColorMatrix (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteColorMatrixMode*

Set the pointer of variable that acquires the Color Matrix Mode.

Color Matrix Mode

Name	Value	Description
STCAM_COLOR_MATRIX_OFF	0x00	OFF
STCAM_COLOR_MATRIX_CUSTOM	0xFF	Custom (Use Color Matrix)

c) *pshtColorMatrix*

Set the pointer of string (SHORT *pshtColorMatrix*[12]) that acquires the Color Matrix.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Color Matrix is acquired.

14.0 Mirror and Rotation

A) StCam_SetMirrorMode

1) Call

```
BOOL StCam_SetMirrorMode (  
    HANDLE hCamera , // Camera Handle  
    BYTE byteMirrorMode // Mirror Mode  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteMirrorMode*

Set the Mirror Mode.

14.0 Mirror and Rotation (Continued)

A) StCam_SetMirrorMode

2) Parameters

Mirror Mode

Name	Value	Description
STCAM_MIRROR_OFF	0	Normal Image
STCAM_MIRROR_HORIZONTAL	1	Mirror Image (Horizontal)
STCAM_MIRROR_VERTICAL	2	Mirror Image (Vertical)
STCAM_MIRROR_HORIZONTAL_VERTICAL	3	Mirror Image(Horizontal & Vertical)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Mirror Mode. This function proceeds on the PC. Therefore sometimes frame rate drops by Mirror Mode.

B) StCam_GetMirrorMode

1) Call

```

BOOL StCam_GetMirrorMode (
    HANDLE   hCamera ,           // Camera Handle
    PBYTE    pbyteMirrorMode    // Mirror Mode
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pbyteMirrorMode

Set the pointer of variable that acquires the Mirror Mode.

Mirror Mode Table

Name	Value	Description
STCAM_MIRROR_OFF	0	Normal Image
STCAM_MIRROR_HORIZONTAL	1	Mirror Image (Horizontal)
STCAM_MIRROR_VERTICAL	2	Mirror Image (Vertical)
STCAM_MIRROR_HORIZONTAL_VERTICAL	3	Mirror Image(Horizontal & Vertical)

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

14.0 Mirror and Rotation (Continued)

B) StCam_GetMirrorMode (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Mirror Mode is acquired.

C) StCam_SetRotationMode

1) Call

```
BOOL StCam_SetRotationMode (  
    HANDLE hCamera ,           // Camera Handle  
    BYTE   byteRotationMode    // Rotation Mode  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *byteRotationMode*

Set the Rotation Mode.

Rotation Mode

Name	Value	Description
STCAM_ROTATION_OFF	0	Rotation OFF
STCAM_ROTATION_CLOCKWISE_90	1	90 deg. Clock wise Rotation
STCAM_ROTATION_COUNTERCLOCKWISE_90	2	90 deg. Counter Clock wise Rotation

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Rotation Mode. When the Rotation Mode changes during preview, window size has to be renewed. This function cannot call during save moving image.

14.0 Mirror and Rotation (Continued)

D) StCam_GetRotationMode

- 1) Call

```
BOOL StCam_GetRotationMode (  
    HANDLE    hCamera ,           // Camera Handle  
    PBYTE     pbyteRotationMode // Rotation Mode  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteRotationMode*
Set the pointer of variable that acquires the Rotation Mode.

Rotation Mode

Name	Value	Description
STCAM_ROTATION_OFF	0	Rotation OFF
STCAM_ROTATION_CLOCKWISE_90	1	90 deg. Clock wise rotation
STCAM_ROTATION_COUNTERCLOCKWISE_90	2	90 deg. Counter clock wise rotation

- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.
- 4) Description
The Rotation Mode is acquired.

15.0 Save AVI File

A) StCam_SaveAVIA

- 1) Call

```
BOOL StCam_SaveAVIA (  
    HANDLE    hCamera ,           // Camera Handle  
    PCSTR     pszFileName ,       // AVI File Name  
    DWORD     dwCompressor ,      // VFW CODEC  
    DWORD     dwLength ,          // Number of Frame  
    LPVOID    lpReserved         // Reserved  
);
```

15.0 Save AVI File (Continued)

A) StCam_SaveAVIA (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pszFileName*

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

c) *dwCompressor*

Set the Compression Method of VFW (Video For Windows).

Compression Method

Name	Value	Description
STCAM_AVI_COMPRESSOR_UNCOMPRESSED	0x00000000	Uncompressed
STCAM_AVI_COMPRESSOR_MP42	0x3234706D	Microsoft MPEG-4 Video Codec V2mpg4c32.dll is necessary to use this CODEC.mpg4c32.dll can get from Windows Media Codecs 8.0 for IT Professionals or Windows Media Tool 4.1
STCAM_AVI_COMPRESSOR_MPV4	0x3467706D	Microsoft MPEG-4 Video Codec V1mpg4c32.dll is necessary to use this CODEC.mpg4c32.dll can get from Windows Media Codecs 8.0 for IT Professionals or Windows Media Tool 4.1
STCAM_AVI_COMPRESSOR_DIALOG_BOX	0xFFFFFFFF	Popup dialog

d) *dwLength*

Set the number of frame.

e) *lpReserved*

Set the "NULL" (This parameter does not use at current SDK).

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Save AVI file. Please reduce frame rate by StCam_SetClock function when playing AVI file is not smooth because of missing frame or save file size.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

15.0 Save AVI File (Continued)

B) StCam_SaveAVIW

1) Call

```

BOOL StCam_SaveAVIW (
    HANDLE   hCamera ,           // Camera Handle
    PCSTR    pszFileName ,       // AVI File Name
    DWORD    dwCompressor ,      // VFW CODEC
    DWORD    dwLength ,          // Number of Frame
    LPVOID   lpReserved          // Reserved
);

```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pszFileName

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

c) dwCompressor

Set the Compression Method of VFW (Video For Windows).

Compression Method

Name	Value	Description
STCAM_AVI_COMPRESSOR_UNCOMPRESSED	0x00000000	Uncompressed
STCAM_AVI_COMPRESSOR_MP42	0x3234706D	Microsoft MPEG-4 Video Codec V2mpg4c32.dll is necessary to use this CODEC.mpg4c32.dll can get from Windows Media Codecs 8.0 for IT Professionals or Windows Media Tool 4.1
STCAM_AVI_COMPRESSOR_MPV4	0x3467706D	Microsoft MPEG-4 Video Codec V1mpg4c32.dll is necessary to use this CODEC.mpg4c32.dll can get from Windows Media Codecs 8.0 for IT Professionals or Windows Media Tool 4.1
STCAM_AVI_COMPRESSOR_DIALOG_BOX	0xFFFFFFFF	Popup dialog

d) dwLength

Set the number of frame.

e) lpReserved

Set the "NULL" (This parameter does not use at current SDK).

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

15.0 Save AVI File (Continued)

B) StCam_SaveAVIW

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Save AVI file. It is necessary to install each CODEC DLL File to make compressed AVI file. Please reduce frame Rate by StCam_SetClock function when playing AVI file is not smooth because of missing frame or save file size. If use Visual Basic, have to use StCam_SaveAVIA function to save AVI file.

C) StCam_SetAVIStatus

1) Call

```
BOOL StCam_SetAVIStatus (  
    HANDLE hCamera ,    // Camera Handle  
    BYTE   byteAVIStatus // Status of AVI File  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) byteAVIStatus

Set the status of making AVI file.

Status of making AVI File

Name	Value	Description
STCAM_AVI_STATUS_STOP	0x00	Stop making AVI file
STCAM_AVI_STATUS_START	0x01	Making AVI file
STCAM_AVI_STATUS_PAUSE	0x02	Temporarily Stop making AVI file

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the status of making AVI file during make AVI file. Status is "Temporary Stop", "Resume", "Stop". If this function calls while not making AVI file, error message is return.

15.0 Save AVI File (Continued)

D) StCam_GetAVIStatus

- 1) Call

```
BOOL StCam_GetAVIStatus (  
    HANDLE    hCamera ,           // Camera Handle  
    PBYTE     pbyteAVIStatus ,    // Status of AVI File  
    PDWORD    pdwTotalFrameCounts , // Number of Total Frame  
    PDWORD    pdwCurrentFrameCounts // Number of current Frame  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pbyteAVIStatus*
Set the pointer of variable that acquires the status of making AVI file.

Status of making AVI File

Name	Value	Description
STCAM_AVI_STATUS_STOP	0x00	Stop making AVI file
STCAM_AVI_STATUS_START	0x01	Making AVI file
STCAM_AVI_STATUS_PAUSE	0x02	Temporary Stop making AVI file

- c) *pdwTotalFrameCounts*
Set the pointer of variable that acquires the number of total frame.
 - d) *pdwCurrentFrameCounts*
Set the pointer of variable that acquires the number of current frame.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.
- 4) Description
The status of making AVI file is acquired.

16.0 Clock

A) StCam_SetClock

- 1) Call

```
BOOL StCam_SetClock (  
    HANDLE    hCamera ,           // Camera Handle  
    DWORD     dwClockMode ,       // Clock Mode  
    DWORD     dwClock             // Clock Frequency  
);
```

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006
Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)
www.SentechAmerica.com

16.0 Clock (Continued)

A) StCam_SetClock (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwClockMode*

Set the Clock Mode. The base clock can acquire by StCam_GetEnableClock function.

Clock Mode

Name	Value	Description
STCAM_CLOCK_MODE_NORMAL	0x00000000	Base Clock
STCAM_CLOCK_MODE_DIV_2	0x00000001	1/2 of Base Clock
STCAM_CLOCK_MODE_DIV_4	0x00000002	1/4 of Base Clock
STCAM_CLOCK_MODE_DIV_8	0x00000004	1/8 of Base Clock
STCAM_CLOCK_MODE_CUSTOM	0x80000000	Custom

c) *dwClock*

Set the clock. The value is only activates when the Clock Mode chose STCAM_CLOCK_MODE_CUSTOM

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the clock speed. This function cannot call during the creation of AVI file. B) StCam_GetClock

B) StCam_GetClock

1) Call

```

BOOL StCam_GetClock (
    HANDLE    hCamera ,    // Camera Handle
    PWORD     pdwClockMode , // Clock Mode
    PWORD     pdwClock      // Clock Frequency
);
    
```

2) Parameters

a) *hCamera*

Set the Camera Handle that acquired is by StCam_Open function.

b) *pdwClockMode*

Set the pointer of variable that acquires the Clock Mode.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

16.0 Clock (Continued)

B) StCam_GetClock (Continued)

2) Parameters (Continued)

Clock Mode

Name	Value	Description
STCAM_CLOCK_MODE_NORMAL	0x00000000	Base Clock
STCAM_CLOCK_MODE_DIV_2	0x00000001	1/2 of Base Clock
STCAM_CLOCK_MODE_DIV_4	0x00000002	1/4 of Base Clock
STCAM_CLOCK_MODE_DIV_8	0x00000004	1/8 of Base Clock
STCAM_CLOCK_MODE_CUSTOM	0x80000000	Custom

c) *pdwClock*

Set the pointer of variable that acquires the clock frequency.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

The clock speed is acquired.

C) StCam_GetEnableClock

1) Call

```
BOOL StCam_GetEnableClock (  
    HANDLE hCamera ,           // Camera Handle  
    PDWORD pdwEnableClockMode , // Clock Mode  
    PDWORD pdwStandardClock ,  // Standard Clock  
    PDWORD pdwMinimumClock ,   // Minimum Speed Clock  
    PDWORD pdwMaximumClock     // Maximum Speed Clock  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwEnableClockMode*

Set the pointer of variable. Variable is result of OR operation of Clock Mode. This variable mean is Clock Mode that can be set for preview. (Current camera has to return 0x00000003).

16.0 Clock (Continued)

C) StCam_GetEnableClock

2) Parameters

Clock Mode

Name	Value	Description
STCAM_CLOCK_MODE_NORMAL	0x00000000	Base Clock
STCAM_CLOCK_MODE_DIV_2	0x00000001	1/2 of Base Clock
STCAM_CLOCK_MODE_DIV_4	0x00000002	1/4 of Base Clock
STCAM_CLOCK_MODE_DIV_8	0x00000004	1/8 of Base Clock
STCAM_CLOCK_MODE_CUSTOM	0x80000000	Custom

c) *pdwStandardClock*

Set the pointer of variable that acquires the base clock speed.

d) *pdwMinimumClock*

Set the pointer of variable that acquires the minimum clock speed that can be set.

e) *pdwMaximumClock*

Set the pointer of variable that acquires the maximum clock speed that can be set.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The clock speed that can be set is acquired.

D) StCam_GetFrameClock

1) Call

```
BOOL StCam_GetFrameClock (  
    HANDLE      hCamera ,           // Camera Handle  
    PWORD      pwCurrentLinePerFrame , // Number of Line per 1 Frame  
    PWORD      pwCurrentClockPerLine  // Number of Clock per 1 Line  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwCurrentLinePerFrame*

Set the pointer of variable that acquires the number of line each frame.

c) *pdwCurrentClockPerLine*

Set the pointer of variable that acquires the number of clock each line.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

16.0 Clock (Continued)

D) StCam_GetFrameClock (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The number of line each frame and the number of clock each line are acquired.

17.0 Callback

A) StCam_AddPreviewBitmapCallback

1) Call

```
BOOL StCam_AddPreviewBitmapCallback (  
    HANDLE                hCamera ,           // Camera Handle  
    fStCamPreviewBitmapCallbackFunc pPreviewBitmapCallbackFunc ,// Callback Function  
    LPVOID                pContext ,         // Context  
    PDWORD                pdwPreiewBitmapCallbackNo // Number of  
                                                    // Preview Bitmap  
                                                    // Callback Function  
);
```

2) Parameters

a) hCamera

Set the Camera Handle that is acquired by StCam_Open function.

b) pPreviewBitmapCallbackFunc

Set the pointer of Preview Bitmap Callback function.

c) pContext

Set the pointer of data that use inside of Callback function.

d) pdwPreiewBitmapCallbackNo

Set the pointer of variable that acquires the entry number of Preview Bitmap Callback function. This number uses at StCam_RemovePreviewBitmapCallback function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Enter the Preview Bitmap Callback function. This function is for Visual C++ only, cannot use for Visual Basic.

17.0 Callback (Continued)

B) StCam_RemovePreviewBitmapCallback

- 1) Call

```
BOOL StCam_RemovePreviewBitmapCallback (  
    HANDLE      hCamera ,           // Camera Handle  
    DWORD      dwPreiewBitmapCallbackNo , // Number of Preview Bitmap Callback  
                                           // Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *dwPreiewBitmapCallbackNo*
Set the entry number of Preview Bitmap Callback function. Set the value that acquire by StCam_AddPreviewBitmapCallback function.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Delete the Preview Bitmap Callback function entered by StCam_AddPreviewBitmapCallback function. This function is for Visual C++ only, cannot use for Visual Basic.

C) StCam_RemoveAllPreviewBitmapCallback

- 1) Call

```
BOOL StCam_RemoveAllPreviewBitmapCallback (  
    HANDLE      hCamera           // Camera Handle  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Delete the all of Preview Bitmap Callback functions that enter by StCam_AddPreviewBitmapCallback function. This function is for Visual C++ only, cannot use for Visual Basic.

17.0 Callback (Continued)

D) StCam_GetPreviewBitmapCallbackCount

- 1) Call

```
BOOL StCam_GetPreviewBitmapCallbackCount (  
    HANDLE      hCamera ,           // Camera Handle  
    PDWORD     pdwListCount       // Number of Callback Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pdwListCount*
Set the pointer of variable that acquires the number of the Preview Bitmap Callback function already entered.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The number of the Preview Bitmap Callback function that already entered is acquired. This function is for Visual C++ only, cannot use for Visual Basic.

E) StCam_GetPreviewBitmapCallback

- 1) Call

```
BOOL StCam_GetPreviewBitmapCallback (  
    HANDLE      hCamera ,           // Camera Handle  
    DWORD      dwCallbackIndex ,   // Callback Function  
                                           // Index  
    fStCamPreviewBitmapCallbackFunc* ppPreviewBitmapCallbackFunc ,  
                                           // Preview Bitmap  
                                           // Callback Function  
    PDWORD     pdwCallbackFunctionNo // Number of  
                                           // Preview Bitmap  
                                           // Callback Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *dwCallbackIndex*
Set the index of Callback function. The value range is 0 to "entry number of function - 1"
 - c) *ppPreviewBitmapCallbackFunc*
Set the pointer of variable that acquires the address of the Preview Bitmap Callback function already entered.
 - d) *pdwCallbackFunctionNo*
Set the pointer of variable that acquires the entry number of the Preview Bitmap Callback function.

17.0 Callback (Continued)

E) StCam_GetPreviewBitmapCallback (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The address and entry number of the Preview Bitmap Callback function already entered by StCam_AddPreviewBitmapCallback function are acquired. This function is for Visual C++ only, cannot use for Visual Basic.

F) StCam_AddPreviewGDIcallback

1) Call

```
BOOL StCam_AddPreviewGDIcallback (  
    HANDLE                hCamera ,           // Camera Handle  
    fStCamPreviewGDIcallbackFunc pPreviewGDIcallbackFunc , // Callback Function  
    LPVOID                pContext ,          // Context  
    PWORD                 pdwPreiewGDIcallbackNo // Number of Preview  
                                                // GDI Callback Function  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pPreviewGDIcallbackFunc*

Set the pointer of Preview GDI (Graphic Device Interface) Callback function.

c) *pContext*

Set the pointer of data that use inside of Callback function.

d) *pdwPreiewGDIcallbackNo*

Set the pointer of variable that acquires the entry number of Preview GDI Callback function. This number uses at StCam_RemovePreviewGDIcallback function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Enter the Preview GDI Callback function. This function is for Visual C++ only, cannot use for Visual Basic.

17.0 Callback (Continued)

G) StCam_RemovePreviewGDICallback

1) Call

```
BOOL StCam_RemovePreviewGDICallback (  
    HANDLE      hCamera ,           // Camera Handle  
    DWORD      dwPreiewGDICallbackNo // Number of Preview GDI Callback Function  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwPreiewGDICallbackNo*

Set the entry number of Preview GDI Callback function. Set the value that acquire by StCam_AddPreviewGDICallback function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Delete the Preview GDI Callback function enter by StCam_AddPreviewGDICallback function. This function is for Visual C++ only, Cannot use for Visual Basic.

H) StCam_RemoveAllPreviewGDICallback

1) Call

```
BOOL StCam_RemoveAllPreviewGDICallback (  
    HANDLE      hCamera           // Camera Handle  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Delete the all of Preview GDI Callback functions enter by StCam_AddPreviewGDICallback function. This function is for Visual C++ only, cannot use for Visual Basic.

17.0 Callback (Continued)

I) StCam_GetPreviewGDICallbackCount

- 1) Call

```
BOOL StCam_GetPreviewGDICallbackCount (  
    HANDLE    hCamera ,           // Camera Handle  
    PDWORD    pdwListCount       // Number of Callback Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *pdwListCount*
Set the pointer of variable that acquires the number of the Preview GDI Callback function already entered.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
The number of the Preview GDI Callback function that already entered is acquired. This function is for Visual C++ only, cannot use for Visual Basic.

J) StCam_GetPreviewGDICallback

- 1) Call

```
BOOL StCam_GetPreviewGDICallback (  
    HANDLE    hCamera ,           // Camera Handle  
    DWORD     dwCallbackIndex ,   // Callback Function  
                                // Index  
    fStCamPreviewGDICallbackFunc* ppPreviewGDICallbackFunc , // Preview GDI  
                                // Callback Function  
    PDWORD    pdwCallbackFunctionNo // Number of  
                                // Preview GDI  
                                // Callback Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *dwCallbackIndex*
Set the index of Callback function. The value range is 0 to "entry number of function - 1"
 - c) *ppPreviewGDICallbackFunc*
Set the pointer of variable that acquires the address of the Preview GDI Callback function already entered.
 - d) *pdwCallbackFunctionNo*
Set the pointer of variable that acquires the entry number of the Preview GDI Callback function.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

17.0 Callback (Continued)

J) StCam_GetPreviewGDICallback (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The address and entry number of the Preview GDI Callback function already entered by StCam_AddPreviewGDICallback function are acquired. This function is for Visual C++ only, cannot use for Visual Basic.

K) StCam_AddRawCallback

1) Call

```
BOOL StCam_AddPreviewGDICallback (  
    HANDLE          hCamera ,           // Camera Handle  
    fStCamRawCallbackFunc pRawCallbackFunc , // Callback Function  
    LPVOID         pContext ,          // Context  
    DWORD          pdwRawCallbackNo    // Number of Raw Callback Function  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pRawCallbackFunc*

Set the pointer of Raw Callback function.

c) *pContext*

Set the pointer of data that use inside of Callback function.

d) *pdwRawCallbackNo*

Set the pointer of variable that acquires the entry number of Raw Callback function. This number uses at StCam_RemoveRawCallback function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Enter the Raw Callback function. This function is for Visual C++ only, Cannot use for Visual Basic.

17.0 Callback (Continued)

L) StCam_RemoveRawCallback

- 1) Call

```
BOOL StCam_RemoveRawCallback (  
    HANDLE      hCamera ,           // Camera Handle  
    DWORD      dwRawCallbackNo      // Number of Raw Callback Function  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
 - b) *dwRawCallbackNo*
Set the entry number of Raw Callback function. Set the value that acquire by StCam_AddRawCallback function.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Delete the Raw Callback function entered by StCam_AddRawCallback function. This function is for Visual C++ only, Cannot use for Visual Basic.

M) StCam_RemoveAllRawCallback

- 1) Call

```
BOOL StCam_RemoveAllRawCallback (  
    HANDLE      hCamera           // Camera Handle  
);
```
- 2) Parameters
 - a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- 3) Return Values
If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.
- 4) Description
Delete the all of Raw Callback functions enter by StCam_AddRawGDICallback function. This function is for Visual C++ only, Cannot use for Visual Basic.

17.0 Callback (Continued)

N) StCam_GetRawCallbackCount

1) Call

```
BOOL StCam_GetRawCallbackCount (  
    HANDLE   hCamera ,           // Camera Handle  
    PDWORD  pdwListCount       // Number of Callback Function  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwListCount*

Set the pointer of variable that acquires the number of the Raw Callback function already entered.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The number of the Raw Callback function that already entered is acquired.. This function is for Visual C++ only, Cannot use for Visual Basic.

O) StCam_GetRawCallback

1) Call

```
BOOL StCam_GetRawCallback (  
    HANDLE   hCamera ,           // Camera Handle  
    DWORD    dwCallbackIndex ,   // Callback Function Index  
    fStCamRawCallbackFunc* ppRawCallbackFunc , // RAW Callback Function  
    PDWORD  pdwCallbackFunctionNo // Number of RAW Callback  
                                           // Function  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwCallbackIndex*

Set the index of Callback function. The value range is 0 to "entry number of function - 1"

c) *ppRawCallbackFunc*

Set the pointer of variable that acquires the address of the Raw Callback function already entered.

d) *pdwCallbackFunctionNo*

Set the pointer of variable that acquires the entry number of the Raw Callback function.

17.0 Callback (Continued)

O) StCam_GetRawCallback (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The address and entry number of the Raw Callback function already entered by StCam_AddRawCallback function are acquired.. This function is for Visual C++ only, Cannot use for Visual Basic.

18.0 Setting

A) StCam_SaveSettingFileA

1) Call

```
BOOL StCam_SaveSettingFileA (  
    HANDLE      hCamera ,      // Camera Handle  
    PCSTR      pszFileName    // Setting File Name  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pszFileName*

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Save current setting to the file. This function cannot call during image data transfer.

B) StCam_SaveSettingFileW

1) Call

```
BOOL StCam_SaveSettingFileW (  
    HANDLE      hCamera ,      // Camera Handle  
    PCSTR      pszFileName    // Setting File Name  
);
```

18.0 Setting (Continued)

B) StCam_SaveSettingFileW (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pszFileName*

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Save current setting to the file. This function cannot call during image data transfer. If use Visual Basic, have to use StCam_SaveSettingFileA to save setting.

C) StCam_LoadSettingFileA

1) Call

```
BOOL StCam_LoadSettingFileA (  
    HANDLE   hCamera ,      // Camera Handle  
    PCSTR    pszFileName    // Setting File Name  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pszFileName*

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Load setting from the file. This function cannot call during image data transfer.

D) StCam_LoadSettingFileW

1) Call

```
BOOL StCam_LoadSettingFileW (  
    HANDLE   hCamera ,      // Camera Handle  
    PCSTR    pszFileName    // Setting File Name  
);
```

18.0 Setting (Continued)

D) StCam_LoadSettingFileW (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pszFileName*

Set the pointer of string that has file name and "NULL". String has to finish by "NULL".

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Load setting from the file. This function cannot call during image data transfer. If use Visual Basic, have to use StCam_LoadSettingFileA to load setting.

E) StCam_ResetSetting

1) Call

```
BOOL StCam_ResetSetting (  
    HANDLE    hCamera    // Camera Handle  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Reset all of the setup to the Camera default setup. This function cannot call during image data transfer.

19.0 EEPROM

A) StCam_ReadUserMemory

1) Call

```
BOOL StCam_ReadUserMemory (  
    HANDLE    hCamera ,    // Camera Handle  
    PBYTE    pbyteBuffer ,    // Buffer for read Data  
    WORD     wOffset ,    // Start Position  
    WORD     wLength      // Size of read Data  
);
```

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

19.0 EEPROM (Continued)

A) StCam_ReadUserMemory (Continued)

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pbyteBuffer*
Set the Pointer of Buffer that acquires the Data.
- c) *wOffset*
Set the start position of data read from the EEPROM. Start position range is 0 to 1023.
- d) *wLength*
Set the size of data read from the EEPROM. Size range is 1 to 1024.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Read the user data from the EEPROM of the Camera.

B) StCam_WriteUserMemory

1) Call

```
BOOL StCam_WriteUserMemory (  
    HANDLE    hCamera ,           // Camera Handle  
    PBYTE     pbyteBuffer ,       // Buffer for write Data  
    WORD      wOffset ,           // Start Position  
    WORD      wLength             // Size of write Data  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pbyteBuffer*
Set the pointer of buffer that writes data.
- c) *wOffset*
Set start position of data writes to the EEPROM. Start position range is 0 to 1023.
- d) *wLength*
Set the size of the data write to the EEPROM. Size range is 1 to 1024.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Write the user data to the EEPROM of the Camera. The EEPROM of the Camera can stores maximum 1024 Bits of data.

19.0 EEPROM (Continued)

C) StCam_ReadCmeraUserIDA

1) Call

```
BOOL StCam_ReadCmeraUserIDA (  
    HANDLE      hCamera ,      // Camera Handle  
    PDWORD     pdwCameraID ,   // User Camera ID  
    PSTR       pszBuffer ,     // Buffer for read Data  
    DWORD      dwBufferSize    // Buffer Size  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pdwCameraID*
Set the pointer of variable that acquires the user Camera ID (4 Bytes of numbers).
- c) *pszBuffer*
Set the pointer of buffer that acquires the user Camera ID (4 Bytes of Numbers) .
- d) *dwBufferSize*
Set the buffer size for data read from the EEPROM.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Camera ID from the EEPROM of the Camera is acquired. Set the Camera ID by StCam_WriteCameraUserIDA function.

D) StCam_ReadCmeraUserIDW

1) Call

```
BOOL StCam_ReadCmeraUserIDW (  
    HANDLE      hCamera ,      // Camera Handle  
    PDWORD     pdwCameraID ,   // Camera ID  
    PWSTR      pszBuffer ,     // Buffer for read Data  
    DWORD      dwBufferSize    // Buffer Size  
);
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pdwCameraID*
Set the pointer of variable that acquires the user Camera ID (4 Bytes of numbers).
- c) *pszBuffer*
Set the pointer of buffer that acquires the user Camera ID (4 Bytes of numbers).
- d) *dwBufferSize*
Set the buffer size for data read from the EEPROM.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

19.0 EEPROM (Continued)

D) StCam_ReadCmeraUserIDW (CContinued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The Camera ID from the EEPROM of the Camera is acquired. Set the Camera ID by StCam_WriteCameraUserIDW function. If use Visual Basic, have to use StCam_WriteCameraUserIDA to set the Camera ID.

E) StCam_WriteCmeraUserIDA

1) Call

```
BOOL StCam_WriteCmeraUserIDA (  
    HANDLE      hCamera ,      // Camera Handle  
    DWORD      dwCameraID ,    // User Camera ID  
    PCSTR      pszBuffer ,     // Buffer for write Data  
    DWORD      dwBufferSize    // Buffer Size  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwCameraID*

Set the user Camera ID (4 Bytes of Numbers).

c) *pszBuffer*

Set the pointer of string that has user Camera ID to write and "NULL". String has to finish by "NULL".

d) *dwBufferSize*

Set the buffer size for data read from the EEPROM.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Camera ID to the EEPROM of the Camera. Acquire the Camera ID by StCam_ReadCameraUserIDA function.

19.0 EEPROM (Continued)

F) StCam_WriteCmeraUserIDW

1) Call

```
BOOL StCam_WriteCmeraUserIDW (  
    HANDLE      hCamera ,           // Camera Handle  
    DWORD       dwCameraID ,        // Camera ID  
    PCWSTR      pszBuffer ,         // Buffer for write Data  
    DWORD       dwBufferSize       // Buffer Size  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwCameraID*

Set the user Camera ID (4 Bytes of Numbers).

c) *pszBuffer*

Set the pointer of string that has user Camera ID to write and "NULL". String has to finish by "NULL".

d) *dwBufferSize*

Set the buffer size for write data from the EEPROM.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

Set the Camera ID to the EEPROM of the Camera. 4 Bytes of numeric value and 250 Bytes of string can sets to the Camera ID. Acquire the Camera ID by StCam_ReadCameraUserIDW function. If use Visual Basic, have to use StCam_ReadCameraUserIDA to acquires the Camera ID.

20.0 Version Information

A) StCam_GetCameraVersion

1) Call

```
BOOL StCam_GetCameraVersion (  
    HANDLE      hCamera ,           // Camera Handle  
    WORD        pwUSBVendorID ,     // USB Vendor ID  
    WORD        pwUSBProductID ,    // USB Product ID  
    WORD        pwCameraType ,      // Camera Type  
    WORD        pwFirmVersion      // Firmware Version  
);
```

20.0 Version Information (Continued)

A) StCam_GetCameraVersion (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pwUSBVendorID*

Set the pointer of variable that acquires the USB vender ID of the Camera.

c) *pwUSBProductID*

Set the pointer of variable that acquires the USB product ID of the Camera.

USB Product ID

Name	Value	Description
STCAM_USBPID_STC_C33USB	0x0305 STC-C33USB	(VGA, Color Model)
STCAM_USBPID_STC_C83USB	0x0605 STC-C83USB	(XGA, Color Model)
STCAM_USBPID_STC_33USB	0x0705 STC-B33USB	(VGA, Monochrome Model)
STCAM_USBPID_STC_83USB	0x0805 STC-B83USB	(XGA, Monochrome Model)

d) *pwCameraType*

Set the pointer of variable that acquires the Camera type of the Camera.

e) *pwFirmVersion*

Set the pointer of variable that acquires the firmware version of the Camera.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE. When acquiring error number, use StCam_GetLastError function.

4) Description

The version information of the Camera is acquired.

B) StCam_GetDriverVersion

1) Call

```

BOOL StCam_GetDriverVersion (
    HANDLE      hCamera ,           // Camera Handle
    PDWORD      pdwFileVersionMS ,  // File Version (First 4 Bytes)
    PDWORD      pdwFileVersionLS ,  // File Version (Second 4 Bytes)
    PDWORD      pdwProductVersionMS , // Product Version (First 4 Bytes)
    PDWORD      pdwProductVersionLS // Product Version (Second 4 Bytes)
);
    
```

20.0 Version Information (Continued)

B) StCam_GetDriverVersion (Continued)

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *pdwFileVersionMS*
Set the pointer of variable that acquires the first 4 Bytes of file version.
- c) *pdwFileVersionLS*
Set the pointer of variable that acquires the second 4 Bytes of file version.
- d) *pdwProductVersionMS*
Set the pointer of variable that acquires the first 4 Bytes of product version.
- e) *pdwProductVersionLS*
Set the pointer of variable that acquires the second 4 Bytes of product version.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The version information of USB driver is acquired.

C) StCam_GetUSBDeviceVersion

1) Call

```
BOOL StCam_GetUSBDeviceVersion (  
    PDWORD pdwFileVersionMS ,           // File Version (First 4 Bytes)  
    PDWORD pdwFileVersionLS ,           // File Version (Second 4 Bytes)  
    PDWORD pdwProductVersionMS ,        // Product Version (First 4 Bytes)  
    PDWORD pdwProductVersionLS         // Product Version (Second 4 Bytes)  
);
```

2) Parameters

- a) *pdwFileVersionMS*
Set the pointer of variable that acquires the first 4 Bytes of file version.
- b) *pdwFileVersionLS*
Set the pointer of variable that acquires the second 4 Bytes of file version.
- c) *pdwProductVersionMS*
Set the pointer of variable that acquires the first 4 Bytes of product version.
- d) *pdwProductVersionLS*
Set the pointer of variable that acquires the second 4 Bytes of product version.

20.0 Version Information (Continued)

C) StCam_GetUSBDllVersion (Continued)

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The version information of StUSB.Dll is acquired.

D) StCam_GetCAMDllVersion

1) Call

```
BOOL StCam_GetCAMDllVersion (  
    PDWORD pdwFileVersionMS ,           // File Version (First 4 Bytes)  
    PDWORD pdwFileVersionLS ,           // File Version (Second 4 Bytes)  
    PDWORD pdwProductVersionMS ,        // Product Version (First 4 Bytes)  
    PDWORD pdwProductVersionLS         // Product Version (Second 4 Bytes)  
);
```

2) Parameters

a) *pdwFileVersionMS*

Set the pointer of variable that acquires the first 4 Bytes of file version.

b) *pdwFileVersionLS*

Set the pointer of variable that acquires the second 4 Bytes of file version.

c) *pdwProductVersionMS*

Set the pointer of variable that acquires the first 4 Bytes of product version.

d) *pdwProductVersionLS*

Set the pointer of variable that acquires the second 4 Bytes of product version.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

The version of StCamD.dll is acquired.

E) StCam_GetUSBFunctionAddress

1) Call

```
BOOL StCam_GetUSBFunctionAddress (  
    HANDLE hCamera ,           // Camera Handle  
    PBYTE pbyteUSBFunctionAddress // USB function address  
);
```

20.0 Version Information (Continued)

E) StCam_GetUSBFunctionAddress (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pbyteUSBFunctionAddress*

Set the pointer variable that acquires the USB function address. The value range is 1 to 127.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

The USB function address is acquired.

21.0 Others

A) StCam_ConvertBitmapBGR24ToRGB24

1) Call

```
BOOL StCam_ConvertBitmapBGR24ToRGB24 (  
    HANDLE   hCamera ,           // Camera Handle  
    DWORD   dwWidth ,           // Image Width  
    DWORD   dwHeight ,         // Image Height  
    PBYTE   pbyteBitmap        // Buffer for Image  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *dwWidth*

Set the width of image.

c) *dwHeight*

Set the height of image.

d) *pbyteBitmap*

Set the pointer of buffer that has image data that before convert. image data that after convert will be overwrite this buffer.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

Convert the BGR 24 Bits image data to RGB 24 Bits image data.

Sensor Technologies America, Inc.

1345 Valwood Parkway, Suite 320 :: Carrollton, Texas 75006

Phone: (972) 481-9223 :: Fax: (972) 481-9209 :: Toll Free: 1 (877) SENTECH (736-8324)

www.SentechAmerica.com

21.0 Others (Continued)

B) StCam_ConvertRawToBGR

1) Call

```

BOOL StCam_ConvertRawToBGR (
    HANDLE   hCamera ,           // Camera Handle
    DWORD    dwWidth ,           // Image Width
    DWORD    dwHeight ,         // Image Height
    PBYTE    pbyteSrcRaw ,       // Buffer for RAW Image
    PBYTE    pbyteDstBGR ,      // Buffer for BGR Image
    BYTE     byteColorInterpolationMethod , // Color Interpolation Method
    DWORD    dwPreviewPixelFormat // Pixel Format of Preview Image
);
    
```

2) Parameters

- a) *hCamera*
Set the Camera Handle that is acquired by StCam_Open function.
- b) *dwWidth*
Set the width of image.
- c) *dwHeight*
Set the height of image.
- d) *pbyteSrcRaw*
Set the pointer of buffer that has image data that before Color Interpolation.
- e) *pbyteDstBGR*
Set the pointer of buffer that has image data that after Color Interpolation.
- f) *byteColorInterpolationMethod*
Set the Color Interpolation Method.

Color Interpolation Method

Name	Value	Description
STCAM_COLOR_INTERPOLATION_NONE_MONO	0	No Color Interpolation ProcessMonochrome Image
STCAM_COLOR_INTERPOLATION_NONE_COLOR	1	No Color Interpolation ProcessColor Image
STCAM_COLOR_INTERPOLATION_NEAREST_NEIGHBOR	2	Use Nearest Neighbor Color Interpolation Method
STCAM_COLOR_INTERPOLATION_BILINEAR	3	Use Bilinear Color Interpolation Method
STCAM_COLOR_INTERPOLATION_BICUBIC	4	Use Bicubic Color Interpolation Method

21.0 Others (Continued)

B) StCam_ConvertRawToBGR (Continued)

2) Parameters (Continued)

- g) *dwPreviewPixelFormat*
Set the pixel format of preview image.

Pixel Format of Preview Image

Name	Value	Description
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	0x00000001	8 Bits each pixel
STCAM_PIXEL_FORMAT_24_BGR	0x00000004	24 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.
STCAM_PIXEL_FORMAT_32_BGR	0x00000008	32 Bits each pixel data (Order by Blue, Green, Red) stores to the buffer from the upper left of the image.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.
When acquiring error number, use StCam_GetLastError function.

4) Description

Color Interpolation process to the raw data from the Camera

C) StCam_SetControlArea

1) Call

```

BOOL StCam_SetControlArea (
    HANDLE      hCamera ,           // Camera Handle
    DWORD       pdwSepalateX ,      // Horizontal Separate Position
    DWORD       pdwSepalateY       // Vertical Separate Position
);
    
```

21.0 Others (Continued)

C) StCam_SetControlArea (Continued)

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pdwSepalateX*

Set the pointer of string (DWORD *pdwSepalateX*[3]) that acquires the horizontal separate position of weight areas for ALC and Auto White Balance.

Set value range is % x 100 (Example: 2500 = 25%, 5000=50%, 7500=75%).

c) *pdwSepalateY*

Set the pointer of string (DWORD *pdwSepalateY*[3]) that acquires the vertical separate position of weight areas for ALC and Auto White Balance.

Set value range is % x 100 (Example: 2500 = 25%, 5000=50%, 7500=75%).

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

Set the area size use for ALC and Auto White Balance.

D) StCam_GetControlArea

1) Call

```
BOOL StCam_GetControlArea (  
    HANDLE      hCamera ,           // Camera Handle  
    PWORD      pwSepalateX ,       // Horizontal Separate Position  
    PWORD      pwSepalateY       // Vertical Separate Position  
);
```

2) Parameters

a) *hCamera*

Set the Camera Handle that is acquired by StCam_Open function.

b) *pwSepalateX*

Set the pointer of string (WORD *pwSepalateX*[3]) that acquires the horizontal separate position of weight areas for ALC and Auto White Balance.

c) *pwSepalateY*

Set the pointer of string (WORD *pwSepalateY*[3]) that acquires the vertical separate position of weight areas for ALC and Auto White Balance.

3) Return Values

If the function succeeds, the return value is TRUE. If the function fails, the return value is FALSE.

When acquiring error number, use StCam_GetLastError function.

4) Description

The area size use for ALC and Auto White Balance is acquired.