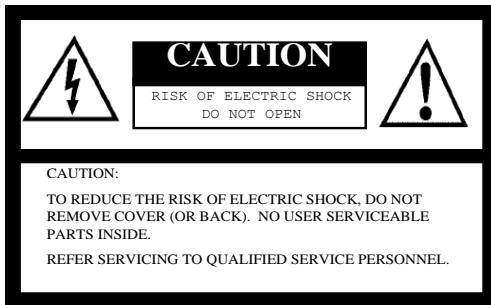




Sentech USB Camera Trigger Function SDK

Sentech USB 2.0
Software Development Kit

Safety Precautions



The lightning flash with arrowhead symbol, within an equilateral triangle, is intended to alert the user to the presence of uninsulated “dangerous voltage” within the product’s enclosure that may be of sufficient magnitude to constitute a risk of electric shock to persons.



The exclamation point within an equilateral triangle is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the appliance.

For U.S.A.

Warning:

This equipment generates and uses radio frequency energy and if not installed and used properly, I.e., in strict accordance with the instruction manual, may cause harmful interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment.

For Canada

Warning:

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

WARNING:

TO PREVENT FIRE OR SHOCK HAZARD, DO NOT EXPOSE THIS APPLIANCE TO RAIN OR MOISTURE.

Product Precautions

- Handle the camera with care. Do not abuse the camera. Avoid striking or shaking it. Improper handling or storage could damage the camera.
- Do not pull or damage the camera cable.
- During camera use, do not wrap the unit in any material. This will cause the internal temperature of the unit to increase.
- Do not expose the camera to moisture, or do not try to operate it in wet areas.
- Do not operate the camera beyond its temperature, humidity and power source ratings.
- While the camera is not being used, keep the lens or lens cap on the camera to prevent dust or contamination from getting in the CCD or filter area and scratching or damaging this area.
- Do not keep the camera under the following conditions:
 - In wet, moist, and high humidity areas
 - Under hot direct sunlight
 - In high temperature areas
 - Near an object that releases a strong magnetic or electric field
 - Areas with strong vibrations
- Use a soft cloth to clean the camera. Use pressured air spray to clean the surface of the glass. DO not scratch the surface of the glass.

Copyright & Disclaimer

Sensor Technologies America, Inc. (DBA Sentech America) believes the contents and specifications of its website, catalog, documentation and ads are correct; however, Sentech America provides no representation or warranty regarding such information or product(s) contained therein. It is requested that Sentech America be given appropriate acknowledgement in any subsequent use of such work by a third party.

While every effort has been made to ensure that the details contained in Sentech America's website and all documentation are correct and up-to-date, Sentech America assumes no liability, legal or otherwise for any errors in listings, specifications, part numbers, process, software or model applications. Sentech America reserves the right to change specifications, product descriptions, product quality, pricing and application at any time without prior written or oral notice. Any party using such information assumes all risk for any and all damaged caused to themselves, a third party and/or property by virtue of incorrect information and/or failure of these products. By installing and/or using a Sentech America software development kit or other similar product and/or information obtained from Sentech America's website, catalog, documentation or ads, you hereby accept and understand these stated terms and conditions.

Contents

I. Notes	5
II. Trigger SDK Applicable Cameras	5
III. Structure of the Trigger Function SDK	5
IV. Features of the Trigger Function	6
V. Windows Version Differences (X86 / X64)	6
VI. Methods for Image Capture	6
VII. Function List of the Trigger Function SDK	7-9
VIII. Function Descriptions of the Trigger Function SDK	10-56
Setting 1: Initialization	10
Setting 2: Camera Information.....	11-14
Setting 3: Setting	15
Setting 4: Camera Settings	16-23
Setting 5: Trigger Settings	24-30
Setting 6: IO Signals.....	31-35
Setting 7: Time Out	36-37
Setting 8: Callback Functions	38-39
Setting 9: Get Raw Image.....	40
Setting 10: Transfer Control.....	41
Setting 11: Noise Reduction.....	42
Setting 12: White Balance.....	43-44
Setting 13: Image Process.....	45-53
Setting 14: Save Imaging Process.....	54
Setting 15: Display Image.....	55
Setting 16: Clear Buffer	56
IX. StTrgTest Sample Software (Command Prompt Base) Operating	57
X. StTrgDisplay Sample Software (Window base) Operating	58-60
XI. StTakeRawSnapShot Sample Software (Take 10 Snap Shot) Operating	60
XII. Frequently Asked Questions	61-62
XIII. Example Operating the Trigger Function SDK	63-64

I. Notes

It is important to have measures for error messages as transferring errors may occur due to increasing load of PC or other reasons.

The following operating conditions may induce errors during the use of this product:

- Operating with a low- performance PC.
- Operating with a low quality USB cable.
- Operating with a long or extended USB cable.
- Operating with a USB camera that connects through a USB HUB.
- Operating with a USB interface board with the early version of USB2.0.
- Operating with a USB interface board (Recommended to use PC using a chipset with a USB function).
- Operating without the latest version of Windows service pack.
- Operating with a USB camera connected to the PC, while other USB equipment is connected.

II. Trigger SDK Applicable Cameras

This trigger SDK is for the following Sentech USB cameras:

USB Camera	The Software Trigger	The Hardware and Software Trigger	The Compact Case
VGA (640 x 480)	STC-TC33USB-AS/AT/BS/BT STC-TB33USB-AS/AT/BS/BT	STC-TC33USB-ASH STC-TB33USB-ASH	STC-MC33USB STC-MB33USB
XGA (1024 x 768)	STC-TC83USB-AS/AT/BS/BT STC-TB83USB-AS/AT/BS/BT	STC-TC83USB-ASH STC-TB83USB-ASH	STC-MC83USB STC-MB83USB
QVGA (1280 x 960)	STC-TC133USB-AS/AT/BS/BT STC-TB133USB-AS/AT/BS/BT	STC-TC133USB-ASH STC-TB133USB-ASH	STC-MC133USB STC-MB133USB
SXGA (1360 x 1024)	STC-TC152USB-AS/AT/BS/BT STC-TB152USB-AS/AT/BS/BT	STC-TC152USB-ASH STC-TB152USB-ASH	STC-MC152USB STC-MB152USB
UXGA (1600 x 1200)	STC-TC202USB-AS/AT/BS/BT STC-TB202USB-AS/AT/BS/BT	STC-TC202USB-ASH STC-TB202USB-ASH	STC-MC202USB STC-MB202USB

Note: Many functions of this SDK do not supported by the non-Trigger USB cameras, which are not listed in the chart above.

III. Structure of the Trigger Function SDK

The trigger function SDK is composed of the following files:

1. Library files (StTrgApi.dll, StTrgApi.lib, StTrgAPI.h, StTrg.cs)
2. USB Driver files (StUSB.sys, StUSB.inf)
3. Sample software for the trigger function (StTrgTest.exe, StTrgDisplay.exe, StTakeRawSnapShot.exe)
4. The source codes of the sample software for Visual C++ 6.0
(For StTrgText.exe: StTrgText.cpp, StTrg.dsp, StTrgTest.dsw)
(For StTrgDisplay.exe: CameraSelectDlg.cpp, FPS.cpp, InputCameraDDlg.cpp, MainFrm.cpp, StdAfx.cpp, StTrgDisplay.cpp, StTrgDisplayDoc.cpp, StTrgDisplayView.cpp, TabChildBase.cpp, TabIODlg.cpp, TabOtherDlg.cpp, TabParentDlg.cpp, TabShutterGainDlg.cpp, TabTriggerModeDlg.cpp, TabTriggerTimingDlg.cpp, TabWBDDlg.cpp, StTrgDisplay.dsp, StTrgDisplay.dsw, CameraSelectDlg.h, FPS.h, InputCameraDDlg.h, Mainrm.h, resource.h, StdAfx.h, StTrgDisplay.h, StTrgDisplayDoc.h, StTrgDisplayView.h, TabChildBase.h, TabIODlg.h, tabparentdlg.h, TabShutterGainDlg.h, TabTriggerTimingDlg.h, TabWBD.h, StTrgDisplay.rc, StTrgDisplay.reg)
(For StTrgRawSnapShot.exe: StTakeRawSnapShot.cpp, StTakeSnapShot.dsp, StTakeRawSnapShot.dsw)
5. The source codes of the sample software for Visual C#

IV. Features of the Trigger Function

The following are additional notes on the trigger function:

1. Frame Numbers

The frame numbers in the camera can be obtained with the image data. These frame numbers can be used for understanding the number of dropped frames.

2. Strobe Control

The external strobe can be controlled by the IO function of the camera, while using a hardware trigger cameras.

3. Exposure End Timing

When using the exposure end callback function, the object can move even if the image data transfer is not finished after exposure is done. Also, with this function, the next exposure can start even if the image data transfer is not finished after exposure is done.

4. Start / Stop Trigger Function

The extended exposure can be done with this function; it is the start /stop timing of the exposure which are controlled from the PC.

5. Memory Function

By having memory function on the camera, the number of dropped frames is reduced.

V. Windows Version Differences (X86 / X64)

The explanation of the combinations of the DLL file, the driver and the operating system is as follows:

	x86 operating system (32bit)		x64 operating system (64bit)		
Application (Specified the CPU)	x86	Any CPU	x86	x64	Any CPU
DLL file (StTrgApi.dll)	StTrgAPI.dll for x86		StTrgAPI.dll for x86	StTrgAPI.dll for x64	
Driver (StUSB.sys)	Driver for x86		Driver for x64	Driver for x64	

Note: The trigger function SDK has the 2 different StUSB.sys and StTrgApi.dll (StTrgApi.lib) files for x86 system (32bit system) and x64 system (64bit system). Using the chart above, please select the proper files based on the operating system and the applications.

VI. Methods for Image Capture

There are two methods for the image capturing:

1. Using the Callback Function

The callback function, which is already entered with "StTrg_SetTransferEndCallback" in advanced, is activated when 1 frame image transfer is completed by the camera. This method cannot be used with Visual Basic.

2. Using the Take Image Function

An image that is already transferred or transferred within the specified time-out time can be obtained by the "StTrg_TakeRawSnapshot" function during the image transfer.

This method can be used with Visual Basic. Therefore, to use this function it is necessary to make a sub-class for procedure for the message at the proper timing.

Before using the "StTrg_TakeRawSnapshot" function, the transferred image is stocked to the buffer inside of SDK. The stock frame number has to be specified with the "StTrg_SetRawSnapshotBufferCount" function.

Delete starting from the oldest image, when the frame numbers is bigger than the stock frame number that is specified by the "StTrg_SetRawSnapshotBufferCount" function and the "StTrg_TakeRawSnapshot" function.

VII. Function List of the Trigger Function SDK

Sections 1	Initialization	
A.	StTrg_Open	Makes the camera available.
B.	StTrg_Close	Makes the camera unavailable.
Section 2	Camera Information	
A.	StTrg_GetDllVersion	Obtains the version information of the DLL file.
B.	StTrg_GetCameraVersion	Obtains the version information of the camera.
C.	StTrg_GetColorArray	Obtains the CCD color array information of the camera.
D.	StTrg_ReadCameraUserIDA	Reads the camera ID (This is for Visual Basic)..
E.	StTrg_ReadCameraUserIDW	Reads the camera ID.
F.	StTrg_WriteCameraUserIDA	Writes the camera ID (This is for Visual Basic)..
G.	StTrg_WriteCameraUserIDW	Writes the camera ID.
H.	StTrg_HasFunction	Verifies the function of the camera.
I.	StTrg_ResetCounter	Resets the frame counter in the camera.
Section 3	Setting	
A.	StTrg_WriteSettingFileA	Saves the current settings to a file (This is for Visual Basic).
B.	StTrg_WriteSettingFileW	Saves the current settings to a file.
C.	StTrg_ReadSettingFileA	Reads the settings from a file (This is for Visual Basic).
D.	StTrg_ReadSettingFileW	Reads the settings from a file.
Section 4	Camera Settings	
A.	StTrg_GetScanMode	Obtains the scanning mode from the camera.
B.	StTrg_SetScanMode	Sets a scanning mode for the camera.
C.	StTrg_GetEnableTransferBitsPerPixel	Obtains the number of the bits that can be represented in each pixel of the image.
D.	StTrg_GetTransferBitsPerPixel	Obtains the number of the bits represented in each pixel of the image.
E.	StTrg_SetTransferBitsPerPixel	Sets the number of the bits represented in each pixel of the image.
F.	StTrg_ConvTo8BitsImage	Converts to the 8bits represented in each pixel of the image.
G.	StTrg_GetClock	Obtains a clock speed for the camera.
H.	StTrg_SetClock	Sets the clock speed for the camera.
I.	StTrg_GetFrameClock	Obtains the total lines and the horizontal clocks.
J.	StTrg_GetExposureClock	Obtains the exposure time.
K.	StTrg_SetExposureClock	Sets the exposure time.
L.	StTrg_GetGain	Obtains the gain.
M.	StTrg_SetGain	Sets the gain.
N.	StTrg_GetDigitalGain	Obtains the digital gain.
O.	StTrg_SetDigitalGain	Sets the digital gain.
Section 5	Trigger Settings	
A.	StTrg_GetTriggerMode	Obtains the trigger mode for the camera.
B.	StTrg_SetTriggerMode	Sets the trigger mode for the camera.
C.	StTrg_GetTriggerTiming	Obtains the timings for the trigger function.
D.	StTrg_SetTriggerTiming	Sets the timings for the trigger function.
E.	StTrg_SoftTrigger	Sends the software trigger signal to the camera.
F.	StTrg_SoftSubTrigger	Sends the second trigger signal to the camera for the extended exposure.
G.	StTrg_TriggerReadOut	Sends the image out signal to the camera.
Section 6	IO Signals	
A.	StTrg_GetIOPinDirection	Obtains the signal direction for the IO port.
B.	StTrg_SetIOPinDirection	Sets the signal direction for the IO port.
C.	StTrg_SetIOPinPolarity	Sets the IO port polarity.
D.	StTrg_GetIOPinPolarity	Obtains the IO port polarity.
E.	StTrg_SetIOPinMode	Sets the signal mode for the IO port.
F.	StTrg_GetIOPinMode	Obtains the signal mode for the IO port.
G.	StTrg_SetIOPinStatus	Sets the output signal status of the output port.
H.	StTrg_GetIOPinStatus	Obtains the output signal status of the output port.
I.	StTrg_GetSwStatus	Obtains DIP Switch Status.
J.	StTrg_GetLEDStatus	Obtains the status of LED.
K.	StTrg_SetLEDStatus	Sets the status of LED.

Section 7	Time Out	
A.	StTrg_GetTimeOut	Obtains the timeout period for the exposure end detects and the image out.
B.	StTrg_SetTimeOut	Sets the timeout period for the exposure end detection and the image out.
Section 8	Callback Functions	
A.	StTrg_SetExposureEndCallback	Sets the callback function that is called at the exposure end.
B.	StTrg_SetTransferEndCallback	Sets the callback function that is called at the completed image out.
C.	StTrg_SetRcvErrorCallback	Sets the callback function that is called when an error occurs.
Section 9	Get Raw Image	
A.	StTrg_SetRcvMsgWnd	Sets the window for the message from the SDK
C.	StTrg_SetRawSnapshotBufferCount	Sets the buffer for the raw image.
B.	StTrg_TakeRawSnapShot	Obtains the raw image.
Section 10	Transfer Control	
A.	StTrg_StartTransfer	Starts the image output.
B.	StTrg_StopTransfer	Stops the image output.
Section 11	Noise Reduction	
A.	StTrg_NoiseReduction	Processes the noise reduction for the extended exposure.
Section 12	White Balance	
A.	StTrg_GetWhiteBalanceMode	Obtains the white balance mode.
B.	StTrg_SetWhiteBalanceMode	Sets the white balance mode.
C.	StTrg_GetWhiteBalanceGain	Obtains the gain for the white balance.
D.	StTrg_SetWhiteBalanceGain	Sets the gain for the white balance.
E.	StTrg_WhiteBalance	Processes white balance.
Section 13	Image Process	
A.	StTrg_GetGammaModeEx	Obtains the gamma mode.
B.	StTrg_SetGammaModeEx	Sets the gamma mode.
C.	StTrg_RawColorGamma	Processes the gamma with raw data.
D.	StTrg_BGRGamma	Processes the gamma with BGR data.
E.	StTrg_MirrorRotation	Processes the mirror and rotation of the image.
F.	StTrg_ColorInterpolation	Processes the color interpolation.
G.	StTrg_GetHueSaturationMode	Obtains the hue and saturation mode.
H.	StTrg_SetHueSaturationMode	Sets the hue and saturation mode.
I.	StTrg_GetColorMatrix	Obtains the color matrix.
J.	StTrg_SetColorMatrix	Sets the color matrix.
K.	StTrg_HueSaturationColorMatrix	Process the color.
L.	StTrg_GetSharpnessMode	Obtains the sharpness mode.
M.	StTrg_SetSharpnessMode	Sets the sharpness mode.
N.	StTrg_Sharpness	Process the sharpness.
Section 14	Image Saving Process	
A.	StTrg_SaveImageA	Saves the image file into a BMP format (For Visual Basic).
B.	StTrg_SaveImageW	Saves the image file into a BMP format.
Section 15	Display Image	
A.	StTrg_Draw	Displays the image with the device context.
Section 16	Clear Buffer	
A.	StTrg_ClearBuffer	Clears the buffer in the camera.

Please **DO NOT** use the commands listed in the chart below during the image out or inside of the Callback function.

Function Name	During image out	Inside Callback Function
StTrg_Close		Prohibited
StTrg_StartTransfer		Prohibited
StTrg_StopTransfer		Prohibited
StTrg_SetExposureEndCallback	Prohibited	Prohibited
StTrg_SetTransferEndCallback	Prohibited	Prohibited
StTrg_SetRcvErrorCallback	Prohibited	Prohibited
StTrg_ScanMode	Prohibited	Prohibited
StTrg_SetTriggerMode	Prohibited	Prohibited
StTrg_SetTimeOut	Prohibited	Prohibited
StTrg_ClearBuffer	Prohibited	Prohibited
StTrg_SetClock	Prohibited	Prohibited
StTrg_WriteSettingFileA	Prohibited	Prohibited
StTrg_ReadSettingFileA	Prohibited	Prohibited
StTrg_WriteSettingFileA	Prohibited	Prohibited
StTrg_ReadSettingFileA	Prohibited	Prohibited

VIII. Function Descriptions of the Trigger Function SDK

Section 1: Initialization

A. StTrg_Open

1. Description:

This command initializes the camera, making it available for receiving commands. After this command, the camera can be controlled through the software. Cameras must be initialized by this command prior to being issued additional commands. Issue this command according to the number of cameras being used.

2. Syntax:

HANDLE StTrg_Open(void)

3. Argument:

N/A

4. Return Code:

HANDLE: The return code uses for other functions.

INVALID_HANDLE_VALUE: The camera cannot open.

B. StTrg_Close

1. Description:

This command closes the camera, making it unavailable for receiving commands. If there is a need to control the camera after this command, it will be necessary to reopen (StTrg_Open) the camera.

2. Syntax:

void StTrg_Close(HANDLE hcamera)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

N/A

Section 2: Camera Information

A. StTrg_GetCameraVersion

1. Description:

This command obtains the version information (Vendor ID, Product ID, FPGA's version, Firmware's version) of the camera.

2. Syntax:

BOOL Stg_GetCameraVersion(HANDLE hCamera, PWORD pwUSBVendorID, PWORD pwUSBProductID, PWORD pwFPGAVersion, PWORD pwFirmVersion)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwUSBVendorID: This parameter obtains the Vendor ID.

*pwUSBProductID: This parameter obtains the Product ID, which is the one of the USB camera product ID listed in the chart below.

USB product ID	Descriptions	Support Trigger Functions
STCAM_USBPID_STC_B33USB	Monochrome VGA	-
STCAM_USBPID_STC_C33USB	Color VGA	-
STCAM_USBPID_STC_B83USB	Monochrome XGA	-
STCAM_USBPID_STC_C83USB	Color XGA	-
STCAM_USBPID_STC_TB33USB	Monochrome VGA	X
STCAM_USBPID_STC_TC33USB	Color VGA	X
STCAM_USBPID_STC_TB83USB	Monochrome XGA	X
STCAM_USBPID_STC_TC83USB	Color XGA	X
STCAM_USBPID_STC_TB133USB	Monochrome QVGA	X
STCAM_USBPID_STC_TC133USB	Color QVGA	X
STCAM_USBPID_STC_TB152USB	Monochrome SXGA	X
STCAM_USBPID_STC_TC152USB	Color SXGA	X
STCAM_USBPID_STC_TB202USB	Monochrome UXGA	X
STCAM_USBPID_STC_TC202USB	Color UXGA	X
STCAM_USBPID_STC_MB33USB	Monochrome VGA	X
STCAM_USBPID_STC_MC33USB	Color VGA	X
STCAM_USBPID_STC_MB83USB	Monochrome XGA	X
STCAM_USBPID_STC_MC83USB	Color XGA	X
STCAM_USBPID_STC_MB133USB	Monochrome QVGA	X
STCAM_USBPID_STC_MC133USB	Color QVGA	X
STCAM_USBPID_STC_MB152USB	Monochrome SXGA	X
STCAM_USBPID_STC_MC152USB	Color SXGA	X
STCAM_USBPID_STC_MB202USB	Monochrome UXGA	X
STCAM_USBPID_STC_MC202USB	Color UXGA	X

Note 1: The trigger function is only available for the STC-TXXXUSB camera series.

Note 2: Many functions of this SDK are not supported by the non-trigger USB cameras.

*pwFPGAVersion: This parameter obtains the FPGA code version.

*pwFirmVersion: This parameter obtains the camera firmware version.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

B. StTrg_GetDllVersion

1. Description:

This command obtains the version information of the SDK.

2. Syntax:

```
BOOL StTrg_GetDllVersion(PDWORD pdwFileVersionMS, PDWORD pdwFileVersionLS, PDWORD  
pdwProductVersionMS, PDWORD pdwProductVersionLS)
```

3. Arguments:

*pdwFileVersionMS: This parameter obtains the first 4 bytes of the file version.

*pdwFileVersionLS: This parameter obtains the second 4 bytes of the file version.

*pdwProductVersionMS: This parameter obtains the first 4 bytes of the product version.

*pdwProductVersionLS: This parameter obtains the second 4 bytes of the product version.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

C. StTrg_GetColorArray

1. Description:

This command obtains the CCD color array information of the camera.

2. Syntax:

```
BOOL StTrg_GetColorArray(HANDLE hCamera, PWORD pwColorArray)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwColorArray: This parameter obtains a CCD type for the camera, which is the one of the CCD type listed in the chart below.

CCD type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RGGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BgBGrR

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

D. StTrg_WriteCameraUserIDA

1. Description:

This command writes the camera ID (Number and / or Name) of the camera (This is for Visual Basic).

Setting the camera ID in advance will allow for easier product identification when there are multiple cameras in the system.

2. Syntax:

```
BOOL StTrg_WriteCameraUserID(HANDLE hCamera, DWORD dwCameraNo, LPCTSTR lpCameraName, DWORD  
dwBufferSize)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwCameraNo: This parameter sets the camera number.

*lpCameraName: This parameter sets the camera name (Maximum is 250 byte including NULL).

dwBufferSize: This parameter sets the buffer size (Unite is byte).

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

E. StTrg_WriteCameraUserIDW

1. Description:

This command writes the camera ID (Number and / or Name) of the camera.

Setting the camera ID in advance will allow for easier product identification when there are multiple cameras in the system.

2. Syntax:

```
BOOL StTrg_WriteCameraUserID(HANDLE hCamera, DWORD dwCameraNo, LPCTSTR IpCameraName, DWORD dwBufferSize)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwCameraNo: This parameter sets the camera number.

*IpCameraName: This parameter sets the camera name (Maximum is 250 byte including NULL).

dwBufferSize: This parameter sets the buffer size (Unite is byte).

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

F. StTrg_ReadCameraUserIDA

1. Description:

This command reads the camera ID (Number and / or Name) of the camera (This is for Visual Basic).

2. Syntax:

```
BOOL StTrg_ReadCameraUserID(HANDLE hCamera, PDWORD pdwCameraNo, LPTSTR IpCameraName, DWORD dwBufferSize)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwCameraNo: This parameter obtains the camera number.

*IpCameraName: This parameter obtains the camera name (Maximum is 250 byte including NULL).

dwBufferSize: This parameter sets the buffer size (Unite is byte).

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

G. StTrg_ReadCameraUserIDW

1. Description:

This command reads the camera ID (Number and / or Name) of the camera.

This command is for the Visual Basic.

2. Syntax:

```
BOOL StTrg_ReadCameraUserID(HANDLE hCamera, PDWORD pdwCameraNo, LPTSTR IpCameraName, DWORD dwBufferSize)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwCameraNo: This parameter obtains the camera number.

*IpCameraName: This parameter obtains the camera name (Maximum is 250 byte including NULL).

dwBufferSize: This parameter sets the buffer size (Unite is byte).

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

H. StTrg_HasFunction

1. Description:

This command verifies the function of the camera.

2. Syntax:

BOOL StTrg_HasFunction(HANDLE hCamera, DWORD dwCameraFunctionID, BOOL* pbHasFunction)

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

dwCameraFunctionID: This parameter sets the camera function ID, which is the one of the following Camera Function IDs listed in the chart below:

Camera function ID	Descriptions
STCAM_CAMERA_FUNCTION_STARTSTOP	Start / stop exposure control function
STCAM_CAMERA_FUNCTION_MEMORY	Memory function
STCAM_CAMERA_FUNCTION_IO_CHANGE_DIRECTION	Change the direction of IO pin function
STCAM_CAMERA_FUNCTION_LED	LED control function
STCAM_CAMERA_FUNCTION_DISABLE_DIP_SW	DISABLE DIP Switch function
STCAM_CAMERA_FUNCTION_TRIGGER	Trigger function
STCAM_CAMERA_FUNCTION_DIGITAL_GAIN	Digital gain function
STCAM_CAMERA_FUNCTION_VARIABLE_PARTIAL	Variable partial scanning method
STCAM_CAMERA_FUNCTION_BINNING_PARTIAL	Binning partial scanning method
STCAM_CAMERA_FUNCTION_IO	IO signal port
STCAM_CAMERA_FUNCTION_REST_FRAME_COUNTER	Frame counter reset function

*pbHasFunction: This parameter obtains the existence of the function.

TRUE: The camera has the function.

FALSE: The camera does NOT have the function.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

I. StTrg_ResetCounter

1. Description:

This command resets the frame counter in the camera. The frame counter value can be obtained by the "ExposureEndCallbackFunc" or "TransferEndCallbackFunc" function.

2. Syntax:

BOOL StTrg_ResetCounter(HANDLE hCamera)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

Section 3: Setting

A. StTrg_WriteSettingFileA

1. Description:

This command saves the setting to a file (This is for Visual Basic).

2. Syntax:

BOOL StTrg_WriteSettingFileA(HANDLE hCamera, PCSTR pszFileName)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pszFileName: This parameter sets a file name of the setting file.

4. Return Code:

TRUE: The function was successful

FALSE: The function failed

B. StTrg_ReadSettingFileA

1. Description:

This command loads the settings from a file (This is for Visual Basic).

2. Syntax:

BOOL StTrg_ReadSettingFileA(HANDLE hCamera, PCSTR pszFileName)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pszFileName: This parameter sets a file name of the setting file.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

C. StTrg_WriteSettingFileW

1. Description:

This command saves the setting to a file.

2. Syntax:

BOOL StTrg_WriteSettingFileW(HANDLE hCamera, PCWSTR pszFileName)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pszFileName: This parameter sets a file name of the setting file.

4. Return Code:

TRUE: The function was successful

FALSE: The function failed

D. StTrg_ReadSettingFileW

1. Description:

This command loads the settings from a file.

2. Syntax:

BOOL StTrg_ReadSettingFileW(HANDLE hCamera, PCWSTR pszFileName)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pszFileName: This parameter sets a file name of the setting file.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

Section 4 Camera Settings

A. StTrg_GetScanMode

1. Description:

This command obtains the scanning mode for the camera.

2. Syntax:

BOOL StTrg_GetScanMode(HANDLE hCamera, PWORD pwScanMode, PDWORD pdwOffsetX, PDWORD pdwOffsetY, PDWORD pdwWidth, PDWORD pdwHeight)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwScanMode: This parameter obtains the scanning mode, which is the one of the scanning mode listed in the chart below:

Scanning mode	STC- XX33USB	STC- XX83USB	STC- XX133USB	STC- XX152USB	STC- XX202USB
STCAM_SCAN_MODE_NORMAL	640 x 480	1024 x 768	1280 x 960	1360 x 1024	1600 x 1200
STCAM_SCAN_MODE_PARTIAL_2	640 x 224	1024 x 344	1280 x 440	1360 x 472	1600 x 544
STCAM_SCAN_MODE_PARTIAL_4	640 x 80	1024 x 136	1280 x 168	1360 x 176	1600 x 208
STCAM_SCAN_MODE_PARTIAL_1	640 x 480	1024 x 768	1280 x 960	1360 x 1024	1600 x 1200
STCAM_SCAN_MODE_VARIABLE_PARTIAL	640 x -	1024 x -	1280 x -	1360 x -	1600 x -
STCAM_SCAN_MODE_BINNING	640 x 240	1024 x 384	1280 x 480	1360 x 512	1600 x 600
STCAM_SCAN_MODE_BINNING_PARTIAL_1	640 x 240	1024 x 384	1280 x 480	1360 x 512	1600 x 600
STCAM_SCAN_MODE_BINNING_PARTIAL_2	640 x 112	1024 x 172	1280 x 220	1360 x 236	1600 x 272
STCAM_SCAN_MODE_BINNING_PARTIAL_4	640 x 40	1024 x 68	1280 x 84	1360 x 88	1600 x 104
STCAM_SCAN_MODE_BINNING_VARIABLE_PARTIAL	640 x -	1024 x -	1280 x -	1360 x -	1600 x -

*pdwOffsetX: This parameter obtains 0.

*pdwOffsetY: This parameter obtains the start line of the image when selected the variable partial scanning. Obtains 0 when select the scanning mode except the variable partial scanning.

*pdwWidth: This parameter obtains the current image width.

*pdwHeight: This parameter obtains the current image height.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

B. StTrg_SetScanMode

1. Description:

This command sets the scanning mode for the camera.

2. Syntax:

BOOL StTrg_SetScanMode(HANDLE hCamera, WORD wScanMode, DWORD dwOffsetX, DWORD dwOffsetY, DWORD dwWidth, DWORD dwHeight)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

wScanMode: This parameter sets the scanning mode, which is the one of the scanning mode listed in the chart below.

The resolution of the camera will change after changing the scanning mode.

Scanning mode	STC-XX33USB	STC-XX83USB	STC-XX133USB	STC-XX152USB	STC-XX202USB
STCAM_SCAN_MODE_NORMAL	640 x 480	1024 x 768	1280 x 960	1360 x 1024	1600 x 1200
STCAM_SCAN_MODE_PARTIAL_2	640 x 224	1024 x 344	1280 x 440	1360 x 472	1600 x 544
STCAM_SCAN_MODE_PARTIAL_4	640 x 80	1024 x 136	1280 x 168	1360 x 176	1600 x 208
STCAM_SCAN_MODE_PARTIAL_1	640 x 480	1024 x 768	1280 x 960	1360 x 1024	1600 x 1200
STCAM_SCAN_MODE_VARIABLE_PARTIAL	640 x -	1024 x -	1280 x -	1360 x -	1600 x -
STCAM_SCAN_MODE_BINNING	640 x 240	1024 x 384	1280 x 480	1360 x 512	1600 x 600
STCAM_SCAN_MODE_BINNING_PARTIAL_1	640 x 240	1024 x 384	1280 x 480	1360 x 512	1600 x 600
STCAM_SCAN_MODE_BINNING_PARTIAL_2	640 x 112	1024 x 172	1280 x 220	1360 x 236	1600 x 272
STCAM_SCAN_MODE_BINNING_PARTIAL_4	640 x 40	1024 x 68	1280 x 84	1360 x 88	1600 x 104
STCAM_SCAN_MODE_BINNING_VARIABLE_PARTIAL	640 x -	1024 x -	1280 x -	1360 x -	1600 x -

dwOffsetX: This parameter sets to 0.

dwOffsetY: This parameter sets the start line for the variable partial scanning. Sets to the multiple numbers of the eight when selected the variable partial scanning. Sets to 0 when selected besides the variable partial scanning.

dwWidth: This parameter sets to 0.

dwHeight: This parameter sets the image height for the variable partial scanning. Sets to the multiple numbers of the eight when selected the variable partial scanning. Sets to 0 when selected besides the variable partial scanning.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

C. StTrg_GetEnableTransferBitsPerPixel

1. Description:

This command obtains the number of bits that can be represented for each pixel of the transfer image.

2. Syntax:

BOOL StTrg_GetEnableTransferBitsPerPixel(HANDLE hCamera, PDWORD pdwEnableTransferBitsPerPixel)

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

*pdwEnableTransferBitsPerPixel: This parameter obtains the number of the bits can be represent for each pixel of the transfer image, the pixel formats are listed in the chart below:

Pixel format	Descriptions
STCAM_TRANSFER_BITS_PER_PIXEL_08	1byte is represented in each pixel.
STCAM_TRANSFER_BITS_PER_PIXEL_10	2bytes are represented in each pixel. Lower 10bits are valid data. Higher 6bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.
STCAM_TRANSFER_BITS_PER_PIXEL_12	2bytes are represented in each pixel. Lower 12bits are valid data. Higher 4bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

D. StTrg_GetTransferBitsPerPixel

1. Description:

This command obtains the number of the bits represented in each pixel of the image.

2. Syntax:

BOOL StTrg_GetTransferBitsPerPixel(HANDLE hCamera, PDWORD pdwTransferBitsPerPixel)

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

*pdwTransferBitPerPixel: This parameter obtains the number of bits represented in each pixel of the image, the pixel formats are listed in the chart below:

Pixel format	Descriptions
STCAM_TRANSFER_BITS_PER_PIXEL_08	1byte is represented in each pixel.
STCAM_TRANSFER_BITS_PER_PIXEL_10	2bytes are represented in each pixel. Lower 10bits are valid data. Higher 6bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.
STCAM_TRANSFER_BITS_PER_PIXEL_12	2bytes are represented in each pixel. Lower 12bits are valid data. Higher 4bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

E. StTrg_SetTransferBitsPerPixel

1. Description:

This command sets the number of the bits represented in each pixel of the image.

2. Syntax:

BOOL StTrg_SetTransferBitsPerPixel(HANDLE hCamera, DWORD dwTransferBitsPerPixel)

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

dwTransferBitsPerPixel: This parameter sets the number of the bits represented in each pixel of the image, the pixel formats are listed in the chart below:

Pixel format	Descriptions
STCAM_TRANSFER_BITS_PER_PIXEL_08	1byte is represented in each pixel.
STCAM_TRANSFER_BITS_PER_PIXEL_10	2bytes are represented in each pixel. Lower 10bits are valid data. Higher 6bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.
STCAM_TRANSFER_BITS_PER_PIXEL_12	2bytes are represented in each pixel. Lower 12bits are valid data. Higher 4bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

F. StTrg_ConvTo8BitsImage

1. Description:

This command sets the number of the bits represented in each pixel of the image.

2. Syntax:

BOOL StTrg_ConvTo8BitsImage(DWORD dwWidth, DWORD dwHeight, DWORD dwTransferBitsPerPixel, PWORD pwRaw, PBYTE pbyteRaw)

3. Arguments:

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

dwTransferBitsPerPixel: This parameter sets the number of the bits represented in each pixel of the image, the pixel formats are listed in the chart below:

Pixel format	Descriptions
STCAM_TRANSFER_BITS_PER_PIXEL_08	1byte is represented in each pixel.
STCAM_TRANSFER_BITS_PER_PIXEL_10	2bytes are represented in each pixel. Lower 10bits are valid data. Higher 6bits are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.
STCAM_TRANSFER_BITS_PER_PIXEL_12	2bytes are represented in each pixel. Lower 12bits are valid data. Higher 4bis are always 0. By selecting this option, the transfer data will increase; therefore, it is necessary to slow down the clock speed.

*pwRaw: This parameter sets the input image.

*pbyteRaw: This parameter obtains the output image, which is the 8bits converted image.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

G. StTrg_GetClock

1. Description:

This command obtains the clock speed for the camera.

2. Syntax:

BOOL StTrg_GetClock(HANDLE hCamera, PDWORD pdwClockMode, PDWORD pdwClock)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwClockMode: This parameter obtains the clock mode, which is the one of the clock mode listed in the chart below.

Clock mode (Clock Speed)	STC-XX33USB	STC-XX83USB	STC-XX133USB	STC-XX152USB	STC-XX202USB
STCAM_CLOCK_MODE_NORMAL	25.5454 MHz	29.5 MHz	36.818 MHz	36.818 MHz	36.818 MHz
STCAM_CLOCK_MODE_DIV_2	12.2727 MHz	14.75 MHz	18.409 MHz	18.409 MHz	18.409 MHz
STCAM_CLOCK_MODE_DIV_4	6.13635 MHz	7.3756 MHz	9.204 MHz	9.204 MHz	9.204 MHz

Clock mode (Clock Speed)	STC-MX33USB
STCAM_CLOCK_MODE_NORMAL	25.5454 MHz
STCAM_CLOCK_MODE_DIV_2	12.2727 MHz
STCAM_CLOCK_MODE_DIV_4	6.13635 MHz
STCAM_CLOCK_MODE_VGA_90FPS	36.818 MHz

*pdwClock: This parameter obtains the clock speed of the camera. Unit is Hz.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

H. StTrg_SetClock

1. Description:

This command sets the clock speed for the camera.

2. Syntax:

BOOL StTrg_SetClock(HANDLE hCamera, DWORD dwClockMode, DWORD dwClock)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwClockMode: This parameter sets the clock mode, which is the one of the clock mode listed in the chart below. When the clock speed is slowed down, it results in the exposure time possibly extending further than the previous clock speed. Also, the frame rate of the image will reduce, but there is a possibility of an image out error.

Clock mode (Clock Speed)	STC-XX33USB	STC-XX83USB	STC-XX133USB	STC-XX152USB	STC-XX202USB
STCAM_CLOCK_MODE_NORMAL	25.5454 MHz	29.5 MHz	36.818 MHz	36.818 MHz	36.818 MHz
STCAM_CLOCK_MODE_DIV_2	12.2727 MHz	14.75 MHz	18.409 MHz	18.409 MHz	18.409 MHz
STCAM_CLOCK_MODE_DIV_4	6.13635 MHz	7.3756 MHz	9.204 MHz	9.204 MHz	9.204 MHz

Clock mode (Clock Speed)	STC-MX33USB
STCAM_CLOCK_MODE_NORMAL	25.5454 MHz
STCAM_CLOCK_MODE_DIV_2	12.2727 MHz
STCAM_CLOCK_MODE_DIV_4	6.13635 MHz
STCAM_CLOCK_MODE_VGA_90FPS	36.818 MHz

dwClock: This parameter sets to 0.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

I. StTrg_GetFrameClock

1. Description:

This command obtains the total lines and the horizontal clocks of the image.

2. Syntax:

BOOL StTrg_GetFrameClock(HANDLE hCamera, PWORD pwTotalLine, PWORD pwClockPerLine)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwTotalLine: This parameter obtains the total lines of the image including the blanking. The total lines of the image will change when the scanning mode is changed.

*pwClockPerLine: This parameter obtains the number of the horizontal clocks.

1V period of time and 1H period of the time can be obtained with the following formulas:

1V period of time[s] = *pwTotalLine x *pwClockPerLine / *pdwClock

1H period of time[s] = *pwClockPerLine / *pdwClock

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

J. StTrg_GetExposureClock

1. Description:

This command obtains the exposure time.

2. Syntax:

BOOL StTrg_GetExposureClock(HANDLE hCamera, PDWORD pdwExposureClock)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwExposureClock: This parameter obtains the exposure time [s].

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

K. StTrg_SetExposureClock

1. Description:

This command sets the exposure time.

2. Syntax:

BOOL StTrg_SetExposureClock(HANDLE hCamera, DWORD dwExposureClock)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwExposureClock: This parameter sets the exposure time[s] x *pdwClock. The camera works with the shutter OFF when dwExposureClock sets to 0. The maximum value to set is *pwClockPerLine x 4090

4. Return Code:

TRUE: the function is successful.

FALSE: the function failed.

L. StTrg_GetGain

1. Description:

This command obtains the gain.

2. Syntax:

BOOL StTrg_GetGain(HANDLE hCamera, PWORD pwGain)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwGain: This parameter obtains the gain of the camera.

4. Return Code:

TRUE: the function is successful.

FALSE: the function failed.

M. StTrg_SetGain

1. Description:

This command sets the gain.

2. Syntax:

BOOL StTrg_SetGain(HANDLE hCamera, WORD wGain)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

wGain: This parameter sets the gain of the camera. The relation between “set value N” and “the value of dB” is shown in the following formula.

$$DB[dB] = -2.36 \text{ dB} + 0.132 \text{ dB} \times N$$

The set range of N is 0 to 255.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

N. StTrg_GetDigitalGain

1. Description:

This command obtains the digital gain of the camera.

2. Syntax:

BOOL StTrg_GetDigitalGain(HANDLE hCamera, PWORD pwDigitalGain)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwDigitalGain: This parameter obtains the digital gain of the camera.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

O. StTrg_SetDigitalGain

1. Description:

This command sets the digital gain, which is controlled at FPGA.

2. Syntax:

BOOL StTrg_SetDigitalGain(HANDLE hCamera, WORD wDigitalGain)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

wDigitalGain: This parameter sets the digital gain of the camera. The pixel value becomes “set value” / 64 multiplied by the digital gain. The set range of the digital gain is 0 to 511.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

Section 5: Trigger Settings

A. StTrg_GetTriggerMode

1. Description:

This command obtains the trigger mode for the camera.

2. Syntax:

BOOL StTrg_GetTriggerMode(HANDLE hCamera, PDWORD pdwTriggerMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwTriggerMode: This parameter obtains the logical disjunction value, which is listed in the chart below.

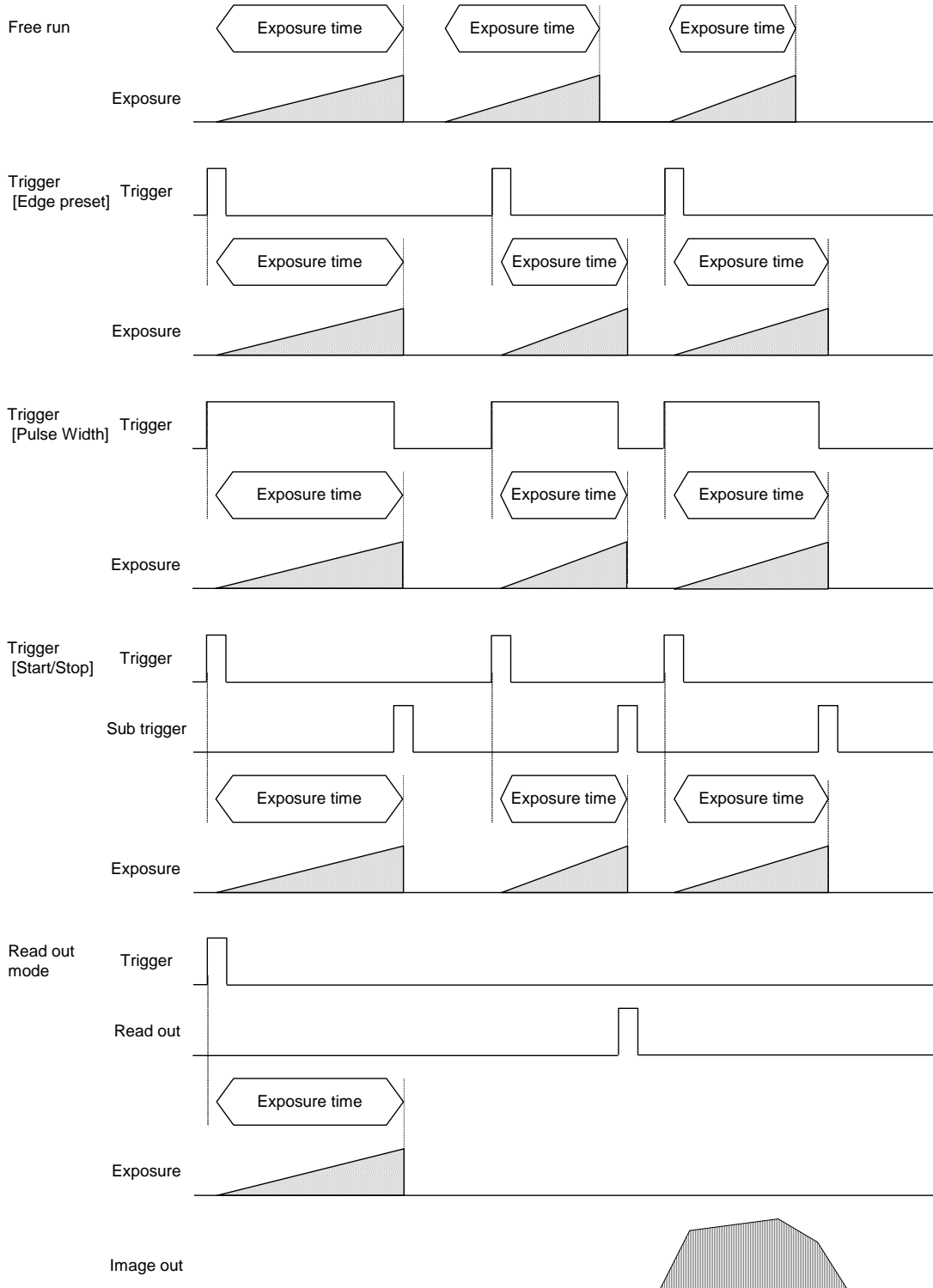
Trigger mode	Descriptions
Operation mode (Select one of the following three modes)	
STCAM_TRIGGER_MODE_TYPE_FREE_RUN	Free running (continuous) mode. This mode has continuous image out without the trigger signal.
STCAM_TRIGGER_MODE_TYPE_TRIGGER	Trigger mode. The exposure starts with the software or the hardware trigger signal then the image out after the exposure is finished.
STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO	Trigger with the video readout mode. The exposure starts with the software or the hardware trigger signal. The image out starts with the software or the hardware image readout signal. If a long period of time elapses while sending the image readout signal after the end of the trigger signal, the image quality may be negatively affected. In this mode, it is necessary to send the image readout signal after sending the trigger signal.
Exposure end detection (Select one of the following two modes)	
STCAM_TRIGGER_MODE_EXPEND_DISABLE	This disables the exposure end detection.
STCAM_TRIGGER_MODE_EXPEND_ENABLE	This enables the exposure end detection.
Trigger signal source (Select one of the following three sources)	
STCAM_TRIGGER_MODE_SOURCE_NONE	This disables the software and the hardware trigger signal input.
STCAM_TRIGGER_MODE_SOURCE_SOFTWARE	This enables the software trigger signal input.
STCAM_TRIGGER_MODE_SOURCE_HARDWARE	This enables the hardware trigger signal input.
Trigger readout signal source (Select one of the following two sources)	
STCAM_TRIGGER_MODE_READOUT_SOFTWARE	This enables the software image readout signal input.
STCAM_TRIGGER_MODE_READOUT_HARDWARE	This enables the hardware image readout signal input.
Trigger mode (Select one of the following two modes)	
STCAM_TRIGGER_MDOE_EXPTIME_EDGE_PRESET	Set the edge preset trigger mode. The exposure time is defined by settings of the StTrg_SetExposureClock function.
STCAM_TRIGGER_MDOE_EXPTIME_PULSE_WIDTH	Set the pulse width trigger mode. The exposure time is the pulse width of the trigger signal.
STCAM_TRIGGER_MODE_EXPTIME_START_STOP	Set the extended exposure (the start and stop signal trigger) mode. The exposure starts with the first trigger signal and the exposure ends with the second trigger signal. [When the trigger signal source is selected the hardware] The first trigger signal is entered to "Trigger Input" and the second trigger signal is entered to "Sub Trigger Input". [When the trigger signal source is selected the software] "StTrg_SoftTrigger" generates the first trigger signal and "StTrg_SoftSubTrigger" generates the second trigger signal.
Exposure wait HD (Select one of the following two settings)	
STCAM_TRIGGER_MDOE_EXPOSURE_WAIT_HD_OFF	The exposure starts immediately after the trigger signal input
STCAM_TRIGGER_MDOE_EXPOSURE_WAIT_HD_ON	The exposure starts with next HD after the trigger signal input.
Exposure wait readout (Select one of the following two settings)	
STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_OFF	This is only for "Edge preset trigger mode". The exposure starts immediately.
STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON	This is only for "Edge preset trigger mode". The exposure starts adjusts with the judgment, whether the previous image out is finished when the exposure is finished.

Trigger mode	Descriptions
Memory type (Select one of the following three types)	
STCAM_TRIGGER_MODE_CAMERA_MEMORY_OFF	No memory function. This mode has continuous image out, without the trigger signal. Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.
STCAM_TRIGGER_MODE_CAMERA_MEMORY_TYPE_A	The number of dropped frames in this memory type is less than that of the previously listed memory type (...MEMORY_OFF). This memory type can be used with "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON". Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.
STCAM_TRIGGER_MODE_CAMERA_MEMORY_TYPE_B	The number of dropped frames in memory type B is less than that of memory type A. This memory type CANNOT be used with "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON". The memory type will automatically change from type B to type A when "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON" is selected. Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.

4. Return Code:

- TRUE: The function is successful.
- FALSE: The function failed.

The explanation of the trigger modes and the exposures are below:



B. StTrg_SetTriggerMode

1. Description:

This command sets the trigger mode for the camera.

2. Syntax:

BOOL StTrg_SetTriggerMode(HANDLE hCamera, DWORD dwTriggerMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwTriggerMode: This parameter sets the logical disjunction value, which is listed in the chart below.

Trigger mode	Descriptions
Operation mode (Select one of the following three modes)	
STCAM_TRIGGER_MODE_TYPE_FREE_RUN	Free running (continuous) mode. This mode has continuous image out without the trigger signal.
STCAM_TRIGGER_MODE_TYPE_TRIGGER	Trigger mode. The exposure starts with the software or the hardware trigger signal then the image out after the exposure is finished.
STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO	Trigger with the video readout mode. The exposure starts with the software or the hardware trigger signal. The image out starts with the software or the hardware image readout signal. If a long period of time elapses while sending the image readout signal after the end of the trigger signal, the image quality may be negatively affected. In this mode, it is necessary to send the image readout signal after sending the trigger signal.
Exposure end detection (Select one of the following two modes)	
STCAM_TRIGGER_MODE_EXPEND_DISABLE	This disables the exposure end detection.
STCAM_TRIGGER_MODE_EXPEND_ENABLE	This enables the exposure end detection.
Trigger signal source (Select one of the following three sources)	
STCAM_TRIGGER_MODE_SOURCE_NONE	This disables the software and the hardware trigger signal input.
STCAM_TRIGGER_MODE_SOURCE_SOFTWARE	This enables the software trigger signal input.
STCAM_TRIGGER_MODE_SOURCE_HARDWARE	This enables the hardware trigger signal input.
Trigger readout signal source (Select one of the following two sources)	
STCAM_TRIGGER_MODE_READOUT_SOFTWARE	This enables the software image readout signal input.
STCAM_TRIGGER_MODE_READOUT_HARDWARE	This enables the hardware image readout signal input.
Trigger mode (Select one of the following two modes)	
STCAM_TRIGGER_MDOE_EXPTIME_EDGE_PRESET	Set the edge preset trigger mode. The exposure time is defined by settings of the StTrg_SetExposureClock function.
STCAM_TRIGGER_MDOE_EXPTIME_PULSE_WIDTH	Set the pulse width trigger mode. The exposure time is the pulse width of the trigger signal.
STCAM_TRIGGER_MODE_EXPTIME_START_STOP	Set the extended exposure (the start and stop signal trigger) mode. The exposure starts with the first trigger signal and the exposure ends with the second trigger signal. [When the trigger signal source is selected the hardware] The first trigger signal is entered to "Trigger Input" and the second trigger signal is entered to "Sub Trigger Input". [When the trigger signal source is selected the software] "StTrg_SoftTrigger" generates the first trigger signal and "StTrg_SoftSubTrigger" generates the second trigger signal.
Exposure wait HD (Select one of the following two settings)	
STCAM_TRIGGER_MDOE_EXPOSURE_WAIT_HD_OFF	The exposure starts immediately after the trigger signal input
STCAM_TRIGGER_MDOE_EXPOSURE_WAIT_HD_ON	The exposure starts with next HD after the trigger signal input.
Exposure wait readout (Select one of the following two settings)	
STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_OFF	This is only for "Edge preset trigger mode". The exposure starts immediately.
STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON	This is only for "Edge preset trigger mode". The exposure starts adjusts with the judgment, whether the previous image out is finished when the exposure is finished.

Trigger mode	Descriptions
Memory type (Select one of the following three types)	
STCAM_TRIGGER_MODE_CAMERA_MEMORY_OFF	No memory function. This mode has continuous image out, without the trigger signal. Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.
STCAM_TRIGGER_MODE_CAMERA_MEMORY_TYPE_A	The number of dropped frames in this memory type is less than that of the previously listed memory type (...MEMORY_OFF). This memory type can be used with "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON". Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.
STCAM_TRIGGER_MODE_CAMERA_MEMORY_TYPE_B	The number of dropped frames in memory type B is less than that of memory type A. This memory type CANNOT be used with "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON". The memory type will automatically change from type B to type A when "STCAM_TRIGGER_MODE_EXPOSURE_WAIT_READOUT_ON" is selected. Please check whether or not the camera supports this function with "StTrg_HasFunction" before applying the function.

Note: Send the trigger signal more than 1V period of time after changing to the trigger mode.

4. Return Code:

TRUE: The function is successful.
FALSE: The function failed.

C. StTrg_GetTriggerTiming

1. Description:

This command obtains the timings for the trigger function.

2. Syntax:

BOOL StTrg_GetTriggerTiming(HANDLE hCamera, DWORD dwTriggerTimingType, PDWORD pdwValue)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwTriggerTimingType: This parameter obtains the type of timing to be used with the camera, which is the one of the timing listed in the chart below.

Timing	Descriptions
STCAM_TRIGGER_TIMING_EXPOSURE_DELAY	The delay of the exposure start
STCAM_TRIGGER_TIMING_STROBE_START_DELAY	The output delay of the output strobe control signal
STCAM_TRIGGER_TIMING_STOBE_END_DELAY	The pulse duration of the output strobe control signal.
STCAM_TRIGGER_TIMING_TRIGGER_PULSE_DELAY	The output delay of the output trigger signal.
STCAM_TRIGGER_TIMING_TRIGGER_PULSE_DURATION	The pulse duration of the output trigger signal.
STCAM_TRIGGER_TIMING_READOUT_DELAY	The image readout delay. The period of time between the exposure end and the image readout starts.

*pdwValue: This parameter obtains the value, which is time[s] x **pdeClock gets by StTrg_GetClock function.

4. Return Code:

TRUE: The function is successful.
FALSE: The function failed.

D. StTrg_SetTriggerTiming

1. Description:

This command sets the timings for the trigger function.

2. Syntax:

BOOL StTrg_SetTriggerTiming(HANDLE hCamera, DWORD dwTriggerTimingType, DWORD dwValue)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwTriggerTimingType: This parameter sets the type of timing to be used with the camera, which is the one of the timing listed in the chart below.

Timing	Descriptions
STCAM_TRIGGER_TIMING_EXPOSURE_DELAY	The delay of the exposure start
STCAM_TRIGGER_TIMING_STROBE_START_DELAY	The output delay of the output strobe control signal
STCAM_TRIGGER_TIMING_STROBE_END_DELAY	The pulse duration of the output strobe control signal.
STCAM_TRIGGER_TIMING_TRIGGER_PULSE_DELAY	The output delay of the output trigger signal.
STCAM_TRIGGER_TIMING_TRIGGER_PULSE_DURATION	The pulse duration of the output trigger signal.
STCAM_TRIGGER_TIMING_READOUT_DELAY	The image readout delay. The period of time between the exposure end and the image readout starts.

dwValue: This parameter sets the value, which is $\text{time[s]} \times \text{pdwClock}$ gets by StTrg_GetClock function. Set the value is less than 4094. Time sets by the unit of HD when sets the timing of the image out delay. The set value is less than 65535 to dwValue. The image out start immediate after the exposure is finished when sets to 0.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

D. StTrg_SoftTrigger

1. Description:

This command sends the software trigger signal to the camera. While send the software trigger signal continuously, if sending the software trigger before the previous image out is finished an error may occur.

2. Syntax:

BOOL StTrg_SoftTrigger(HANDLE hCamera)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

F. StTrg_SoftSubTrigger

1. Description:

This command sends the second software trigger signal to the camera to stop the exposure. This command is activated when the camera works with the extended exposure mode.

2. Syntax:

BOOL StTrg_SoftSubTrigger(HANDLE hCamera)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

G. StTrg_TriggerReadOut

1. Description:

This command sends the image out signal to the camera. Send the image out signal after the exposure is finished.

2. Syntax:

```
BOOL StTrg_TriggerReadOut(HANDLE hCamera)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

Section 6: IO Signals

A. StTrg_GetIOPinDirection

1. Description:

This command obtains the direction for the IO port (IO0, IO1, IO2, and IO3).

2. Syntax:

```
BOOL StTrg_GetIOPinDirection(HANDLE hCamera, PDWORD pdwDirection)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwDirection: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The IO port is the input signal port when obtains 0 and the output port when obtains 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

B. StTrg_SetIOPinDirection

1. Description:

This command sets the IO port (IO0, IO1, IO2, and IO3) information. Please check whether on not the camera has this function with "StTrg_HasFunction" before applying the command.

2. Syntax:

```
BOOL StTrg_SetIOPinDirection(HANDLE hCamera, DWORD dwDirection)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwDirection: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The IO port is the input signal port when sets 0 and the output port when sets 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

C. StTrg_GetIOPinPolarity

1. Description:

This command obtains the polarity of the signal through IO port (IO0, IO1, IO2 and IO3).

2. Syntax:

```
BOOL StTrg_GetIOPinPolarity(HANDLE hCamera, PDWORD pdwPolarity)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwPolarity: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The positive polarity signal through the IO port when obtains 0 and the negative polarity signal through the IO port when obtains 1.

4. Return Code:

TRUE: the function is successful.

FALSE: the function failed.

D. StTrg_SetIOPinPolarity

1. Description:

This command sets the polarity of the signal through IO port (IO0, IO1, IO2 and IO3). This setting is disregarded for the general signal.

2. Syntax:

```
BOOL StTrg_SetIOPinPolarity(HANDLE hCamera, DWORD dwPolarity)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwPolarity: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The positive polarity signal through the IO port when sets to 0 and the negative polarity signal through the IO port when sets to 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

E. StTrg_GetIOPinMode

1. Description:

This command obtains the signal mode for the IO port (IO0, IO1, IO2 and IO3).

2. Syntax:

BOOL StTrg_GetIOPinMode(HANDLE hCamera, BYTE bytePinNo, PDWORD pdwMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

bytePinNo: This parameter sets the IO port number. The IO port number is 0 when sets to 0. The IO port number is 3 when sets to 3.

*pdwMode: This parameter obtains the signal mode, which is the one of the signal mode listed in the chart below.

Signal mode	In/Out	Descriptions
STCAM_OUT_PIN_MODE_DISABLE	Output	Disables output port
STCAM_OUT_PIN_MODE_GENERAL_OUTPUT	Output	Enables output port for the general signal output. Set the output signal status by StTrg_SetIOPinStatus function.
STCAM_OUT_PIN_MODE_TRIGGER_OUTPUT_PROGRAMABLE	Output	Enables output port for the trigger signal output.
STCAM_OUT_PIN_MODE_TRIGGER_LOOP_THROUGH	Output	Enables output port for the trigger signal output. Input trigger signal is output as is.
STCAM_OUT_PIN_MODE_EXPOSURE_END	Output	Enables output port for the exposure end signal output.
STCAM_OUT_PIN_MODE_CCD_READ_END_OUTPUT	Output	Enables output port for the image readout end signal output.
STCAM_OUT_PIN_MODE_STROBE_OUTPUT_PROGRAMABLE	Output	Enables output port for the strobe control signal output. The strobe control signal set by StTrg_SetTriggerTiming function.
STCAM_IN_PIN_MODE_STROBE_OUTPUT_EXPOSURE	Output	Enables output port for the strobe control signal output. The strobe control signal is the same as the exposure time and timing.
STCAM_IN_PIN_MODE_DISABLE	Input	Disables input port
STCAM_IN_PIN_MODE_GENERAL_INPUT	Input	Enables input port for the general signal input. Get the input signal status by StTrg_GetIOPinStatus function.
STCAM_IN_PIN_MODE_TRIGGER_INPUT	Input	Enables input port for the trigger signal input.
STCAM_IN_PIN_MODE_READOUT_INPUT	Input	Enables input port for the image readout signal input.
STCAM_IN_PIN_MODE_SUB_TRIGGER_INPUT	Input	Enables input port for the second trigger signal input for the extended trigger mode.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

F. StTrg_SetIOPinMode

1. Description:

This command sets the signal mode for the IO port (IO0, IO1, IO2 and IO3).

2. Syntax:

BOOL StTrg_SetIOPinMode(HANDLE hCamera, BYTE bytePinNo, DWORD dwMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

bytePinNo: This parameter sets the IO port number. The IO port number is 0 when sets to 0. The IO port number is 3 when sets to 3.

dwMode: This parameter sets signal mode, which is the one of the signal mode listed in the chart below:

Signal mode	In/Out	Descriptions
STCAM_OUT_PIN_MODE_DISABLE	Output	Disables output port
STCAM_OUT_PIN_MODE_GENERAL_OUTPUT	Output	Enables output port for the general signal output. Set the output signal status by StTrg_SetIOPinStatus function.
STCAM_OUT_PIN_MODE_TRIGGER_OUTPUT_PROGRAMABLE	Output	Enables output port for the trigger signal output.
STCAM_OUT_PIN_MODE_TRIGGER_LOOP_THROUGH	Output	Enables output port for the trigger signal output. Input trigger signal is output as is.
STCAM_OUT_PIN_MODE_EXPOSURE_END	Output	Enables output port for the exposure end signal output.
STCAM_OUT_PIN_MODE_CCD_READ_END_OUTPUT	Output	Enables output port for the image readout end signal output.
STCAM_OUT_PIN_MODE_STROBE_OUTPUT_PROGRAMABLE	Output	Enables output port for the strobe control signal output. The strobe control signal set by StTrg_SetTriggerTiming function.
STCAM_IN_PIN_MODE_STROBE_OUTPUT_EXPOSURE	Output	Enables output port for the strobe control signal output. The strobe control signal is the same as the exposure time and timing.
STCAM_IN_PIN_MODE_DISABLE	Input	Disables input port
STCAM_IN_PIN_MODE_GENERAL_INPUT	Input	Enables input port for the general signal input. Get the input signal status by StTrg_GetIOPinStatus function.
STCAM_IN_PIN_MODE_TRIGGER_INPUT	Input	Enables input port for the trigger signal input.
STCAM_IN_PIN_MODE_READOUT_INPUT	Input	Enables input port for the image readout signal input.
STCAM_IN_PIN_MODE_SUB_TRIGGER_INPUT	Input	Enables input port for the second trigger signal input for the extended trigger mode.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

G. StTrg_GetIOPinStatus

1. Description:

This control obtains the output signal status of the general signal.

2. Syntax:

BOOL StTrg_GetIOPinStatus(HANDLE hCamera, PDWORD pdwStatus)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwStatus: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The low signal through the IO port when obtains 0 and the high signal through the IO port when obtains 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

H. StTrg_SetIOPinStatus

1. Description:

This control sets the output signal status of the general signal.

2. Syntax:

```
BOOL StTrg_SetIOPinStatus(HANDLE hCamera, DWORD dwStatus)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwStatus: The bit0 represents the IO0, the bit1 represents the IO1, the bit2 represents the IO2 and the bit3 represents the IO3. The low signal through the IO port when sets to 0 and the high signal through the IO port when sets to 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

I. StTrg_GetSwStatus

1. Description:

This command obtains DIP Switch status of the camera.

2. Syntax:

```
BOOL APIENTRY StTrg_GetSwStatus(HANDLE hCamera, PDWORD pdwSwStatus)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pdwSwStatus: The bit0 represents the SW1, the bit1 represents the SW2, the bit2 represents the SW3 and the bit3 represents the SW4. The switch is OFF position when obtains 0 and the switch is ON position when obtains 1.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

J. StTrg_GetLEDStatus

1. Description:

This command obtains the status of the LED. This command is NOT available for all cameras. Please verify whether the camera has this function before applying the command.

2. Syntax:

```
BOOL StTrg_GetLEDStatus(HANDLE hCamera, PDWORD pdwLEDStatus)
```

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

*pdwLEDStatus: This parameter obtains the logical disjunction value the LED status, which is the combination of LED status listed in the chart below.

LED status	Descriptions
STCAM_LED_GREEN_ON	Green LED is on
STCAM_LED_RED_ON	Red LED is on

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

K. StTrg_SetLEDStatus

1. Description:

This command sets the status of the LED. This command is NOT available for all cameras. Please verify whether the camera has an LED function before applying the command.

2. Syntax:

BOOL StTrg_SetLEDStatus(HANDLE hCamera, DWORD dwLEDStatus)

3. Arguments:

hCamera: This parameter sets the control handle of the camera.

dwLEDStatus: This parameter sets the logical disjunction value the LED status, which is the combination of LED status listed in the chart below:

LED status	Descriptions
STCAM_LED_GREEN_ON	Green LED is on
STCAM_LED_RED_ON	Red LED is on

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

Section 7: Time Out

A. StTrg_GetTimeOut

1. Description:

This command obtains the time [ms] of timeout for the exposure end detects or the image out.

2. Syntax:

BOOL StTrg_GetTimeOut(HANDLE hCamera, DWORD dwTimeOutType, PDWORD pdwTimeOutMS)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwTimeOutType: This parameter obtains the time out type, which is the one of the time out type listed in the chart below.

Time out type	Descriptions
STCAM_TIMEOUT_ST2EE	This is for the software trigger mode. Set time of the timeout from send the trigger signal to the exposure end detects. Set time more than the exposure time. This timeout disable when set "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" or "STCAM_TRIGGER_MODE_EXPEND_DISABLE" for the trigger mode. Sets the time of the timeout from sending the second trigger signal to the exposure end detects when the trigger mode is set to "STCAM_TRIGGER_MODE_EXPTIME_START_STOP".
STCAM_TIMEOUT_TE2EE	This is for the hardware trigger mode. Set time of the timeout from the previous image out (or the image out start by StTrg_StartTransfer function) to the exposure end detects. This timeout error is occurred periodically if no hardware trigger signal input. Set time is more than "the interval from the exposure end to next trigger signal input" + "the exposure time".
STCAM_TIMEOUT_EE2TE	Set time of the timeout from the exposure end detects to the image out end. Set time more than 1V period of time. This timeout disable when set "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" for the trigger mode.
STCAM_TIMEOUT_RO2TE	Set the time of timeout from send the image out signal to the image out end. Set time more than 1V period of time This timeout disabled when set "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" for the trigger mode.

*pdwTimeOutMS: This parameter obtains the time out [ms]. The timeout error occurs when the communication is not finished within the set time. The timeout setting is disabled when this parameter is set to 0xFFFFFFFF. When the timeout setting is disabled, there is possibility to have no response from the camera.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

B. StTrg_SetTimeOut

1. Description:

This command sets the time [ms] of timeout for the exposure end detects or the image out.

2. Syntax:

BOOL StTrg_SetTimeOut(HANDLE hCamera, DWORD dwTimeOutType, DWORD dwTimeOutMS)

3. Argument:

hCamera: This parameter sets the control handle of the camera

dwTimeOutType: This parameter sets the time out type, which is the one of the time out type listed in the chart below.

Time out type	Descriptions
STCAM_TIMEOUT_ST2EE	This is for the software trigger mode. Sets time of the timeout from send the trigger signal to the exposure end detects. Sets time more than the exposure time. When the trigger mode is set to "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" or "STCAM_TRIGGER_MODE_EXPEND_DISABLE" the timeout is disabled. Sets the time of the timeout from sending the second trigger signal to the exposure end detects when the trigger mode is set to "STCAM_TRIGGER_MODE_EXPTIME_START_STOP".
STCAM_TIMEOUT_TE2EE	This is for the hardware trigger mode. Sets time of the timeout from the previous image out (or the image out start by StTrg_StartTransfer function) to the exposure end detects. This timeout error is occurred periodically if no hardware trigger signal input. Sets time is more than "the interval from the exposure end to next trigger signal input" + "the exposure time".
STCAM_TIMEOUT_EE2TE	Sets time of the timeout from the exposure end detects to the image out end. Sets time more than 1V period of time. This timeout disable when set "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" for the trigger mode.
STCAM_TIMEOUT_RO2TE	Sets the time of timeout from send the image out signal to the image out end. Sets time more than 1V period of time. This timeout disabled when set "STCAM_TRIGGER_MODE_TYPE_TRIGGER_RO" for the trigger mode.

dwTimeOutMS: This parameter sets the time out [ms]. The timeout error occurs when the communication is not finished within the set time. The timeout setting is disabled when this parameter is set to 0xFFFFFFFF. When the timeout setting is disabled, there is possibility to have no response from the camera.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

Section 8: Callback Functions

A. StTrg_SetExposureEndCallback

1. Description:

This command sets the callback function that uses after the exposure is finished.

This function is disabled when set "STCAM_TRIGGER_MODE_EXPEND_DISABLE" to the trigger mode.

2. Syntax:

BOOL StTrg_SetExposureEndCallback(HANDLE hCamera, funcExposureEndCallback func1, PVOID pvContext)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

func1: This function is disabled when set to "NULL".

The exposure end callback function is as below.

void ExposureEndCallbackFunc(HANDLE hCamera, DWORD dwFrameNo, PVOID pvContext)

dwFrameNo: Frame No. Reference to **StTrg_ResetCounter** function.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

B. StTrg_SetTransferEndCallback

1. Description:

This command sets the callback function that uses after the image out is finished.

2. Syntax:

BOOL StTrg_SetTransferEndCallback(HANDLE hCamera, funcTransgerEndCallback func2, PVOID pvContext)

3. Argument:

hCamera: This parameter sets control handle of the camera.

func2: This function is disabled when set "NULL"

The transfer end callback function is not processed until the exposure end callback is finished, if still the exposure end callback function is processed when the image out is finished.

The transfer end callback function is below.

void TransferEndCallbackFunc(HANDLE hCamera, DWORD dwFrameNo, DWORD dwWidth, DWORD dwHeight, WORD wColorArray, PBYTE pbyteRaw, PVOID pvContext)

dwFrameNo: This parameter sets Frame No. reference to StTrg_ResetCoutner function.

dwWidth: This parameter obtains the image width

dwHeight: This parameter obtains the image height

wColorArray: This parameter sets a CCD type for the camera, which is one of the CCD type listed in the chart below:

CCD type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RRGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BGrGrR

***pbyteRaw:** This parameter obtains the raw image. If it is necessary to process the white balance or the color interpolation, use this raw image. It is necessary to make a copy of the raw image at pbyteRaw before processes the callback function because this image is overwrite after the callback function.

*pvContext: This parameter sets the return value of the callback function.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

C. StTrg_SetRcvErrorCallback

1. Description:

This command sets the callback function that uses when the error is occurred.

2. Syntax:

```
BOOL StTrg_SetRcvErrorCallback(HANDLE hCamera, funcRcvErrorCallback func3, PVOID pvContext)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

func3: This function is disabled when set "NULL".

The error receives callback function is as below.

void RcvErrorCallbackFnc(HANDLE hCamera, DWORD dwErrorCode, PVOID pvContext)

dwErrorCode: This parameter obtains the error code, which is the one of the error code listed in the chart below:

Error code	Descriptions
ERROR_ACCESS_DENIED	Could not access to the camera because of the camera was disconnected to the PC. It is necessary to close the camera at outside of the callback function when get "ERROR_ACCESS_DENIED".
ERR_EXPOSURE_END_DROPPED	The error was occurred while receive the exposure end signal.
ERR_IMAGE_DATA_DROPPED	The error was occurred while received the image
ERR_TIMEOUT_ST2EE	The timeout error was occurred between send the software trigger signal and the exposure end detects.
ERR_TIMEOUT_TE2EE	The timeout error was occurred between the previous image out end the exposure end detects.
ERR_TIMEOUT_EE2TE	The timeout error was occurred while the image out after the exposure end detects.
ERR_TIMEOUT_RO2TE	The timeout error was occurred while the image out after send the image out signal.

Note 1: Possibility to not detect all error.

Note 2: If send another trigger signal while the error is occurred, the error is occurred again. The image can get by send new trigger signal when get the value except "ERROR_ACCESS_DENIED", which is ready to send image. The cause of the error is "drop the image output speed" or "The image out starts before the previous image out is not finished" or others. Set good enough time for the interval from the exposure end to the trigger input. When drop the image is happened, may get "ERR_XX_DROPPED" or "ERR_TIMEOUT_XX" at error timing.

*pvContext: This parameter sets the return value of the callback function.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

Section 9: Get Raw Image

A. StTrg_SetRcvMsgWnd

1. Description:

This command sets the window that obtains the message from the SDK.

2. Syntax:

BOOL StTrg_SetRcvMsgWnd(HANDLE hCamera, HWND hWnd)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

hWnd: This parameter sets the handle of the window that obtains the message from the SDK, which is the one of the following handles listed in the chart below:

Handle	Descriptions
WM_STCAM_TRANSFER_END	The image out finished. (wParam: hCamera, lParam: frame number)
WM_STCAM_EXPOSURE_END	The exposure finished. (wParam: hCamera, lParam: frame number)
WM_STCAM_ERROR	An error occurred. (wParam: hCamera, lParam: error number)

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

B. StTrg_SetRawSnapShotBufferCount

1. Description:

This command sets the buffer count (buffer size) for "StTrg_TakeRawSnapShot" function. It is necessary to setup the buffer count (buffer size) before using the "StTrg_TakeRawSnapShot" function.

2. Syntax:

BOOL StTrg_SetRawSnapShotBufferCount(HANDLE hCamera, DWORD dwBufferCount)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwBufferCount: This parameter sets the number of the frames for "StTrg_TakeRawSnapShot".

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

C. StTrg_TakeRawSnapShot

1. Description:

This command obtains the raw image while the image out without the Callback functions. "StTrg_TakeRawSnapShot" and "StTrg_SetTransferEndCallback" functions cannot be used because "StTrg_TakeRawSnapShot" function calls "StTrg_SetTransferEndCallback" in the SDK. It is necessary to setup the buffer count (buffer size) with "StTrg_SetRawSnapShotBufferCount" before using this function.

2. Syntax:

BOOL StTrg_TakeRawSnapShot(HANDLE hCamera, PBYTE pbyteBuffer, DWORD dwBufferSize, PDWORD pdwNumberOfByteTranse, PDWORD pdwFrameNo, DWORD dwMilliseconds)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pbyteBuffer: This parameter sets the first address of the buffer.

*pbyteBuffer: This parameter obtains the image data.

dwBufferSize: This parameter set the buffer size (unit is byte).

*pdwNumberOfByteTrans: This parameter obtains the image size (unit is byte).

*pdwFrameNo: This parameter obtains the frame number.

dwMilliseconds: This parameter sets the time out time (unit is milliseconds).

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

Section 10: Transfer Control

A. StTrg_StartTransfer

1. Description:

This command starts the image out. It is necessary to start the image out before sending another trigger signal.

2. Syntax:

```
BOOL StTrg_StartTransfer(HANDLE hCamera)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

B. StTrg_StopTransfer

1. Description:

This command stops the image out.

2. Syntax:

```
BOOL StTrg_StopTransfer(HANDLE hCamera)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

Section 11: Noise Reduction

A. StTrg_NoiseReduction

1. Description:

This command processes the noise reduction of the image.

2. Syntax:

BOOL StTrg_NoiseReduction(HANDLE hCamera, DWORD dwReductionMode, DWORD dwWidth, DWORD dwHeight, WORD wColorArray, PBYTE pbyteRaw)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwReductionMode: This parameter sets the noise reduction mode, which is the one of the noise reduction modes listed in the chart below:

Noise Reduction Mode	Descriptions
STCAM_NR_OFF	No noise reduction process
STCAM_NR_EASY	Noise reduction process with the current image. Advantage: The random noises can be reduced and it is unnecessary to calibrate the shade image. Disadvantage: The resolution of the image is reduced and does not reduce the adjoining noises.
STCAM_NR_COMPREX	Noise reduction process with the shade image. It is necessary to calibrate the shade image with STCAM_NR_DARK_CL mode. Advantage: The adjoining noises can be reduced and the resolution of the image may not reduce. Disadvantage: The random noises do not reduce and it is necessary to calibrate the shade image.
STCAM_NR_DARK_CL	Calibrates the shade image for the noise reduction process. Gets the shade image with the same condition of the camera (gain and the exposure time) for taking the image.

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

wColorArray: This parameter obtains a CCD type for the camera, which is the one of the CCD type listed in the chart below:

CCD Type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RGGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BGbGrR

*pbyteRaw: This parameter obtains the raw image by TransferEndCallbackFunc function. *pbyteRaw is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

Section 12: White Balance

A. StTrg_GetWhiteBalanceMode

1. Description:

This command obtains the white balance mode.

2. Syntax:

BOOL StTrg_GetWhiteBalanceMode(HANDLE hCamera, PBYTE pbyteWBMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pbyteWBMode: This parameter obtains the white balance mode, which is the one of the white balance mode listed in the chart below.

White balance mode	Descriptions
STCAM_WB_OFF	No white balance process.
STCAM_WB_MANUAL	Manual white balance process with the gain settings already set.
STCAM_WB_ONESHOT	Auto white balance process once then change automatically to manual white balance after that with gain setting that gets before change manual white balance.
STCAM_WB_FULLAUTO	Auto white balance process.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

B. StTrg_SetWhiteBalanceMode

1. Description:

This command sets the white balance mode.

2. Syntax:

BOOL StTrg_SetWhiteBalanceMode(HANDLE hCamera, BYTE byteWBMode)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteWBMode: This parameter sets the white balance mode, which is the one of the white balance mode listed in the chart below.

White balance mode	Descriptions
STCAM_WB_OFF	No white balance process.
STCAM_WB_MANUAL	Manual white balance process with the gain settings already set.
STCAM_WB_ONESHOT	Auto white balance process once then change automatically to manual white balance after that with gain setting that gets before change manual white balance.
STCAM_WB_FULLAUTO	Auto white balance process.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

C. StTrg_GetWhiteBalanceGain

1. Description:

This command obtains the gain for the white balance.

2. Syntax:

```
BOOL StTrg_SetWhiteBalanceGain(HANDLE hCamera, PWORD pwWBGainR, PWORD pwWBGainGr, PWORD pwWBGainGb, PWORD pwWBGainB)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pwWBGainR: This parameter obtains the R gain for the white balance. The multiplier for R is "sets value" / 128 (The gain is time 1 when obtains 128).

*pwWBGainGr: This parameter obtains the Gr (which is the G on the row in a R) gain for the white balance. The multiplier for Gr is "obtains value" / 128 (The gain is time 1 when obtains 128).

*pwWBGainGb: This parameter obtains the Gb (which is the G on the row in a B) gain for the white balance. The multiplier for Gb is "obtains value" / 128 (The gain is time 1 when obtains 128).

*pwWBGainB: This parameter obtains the B gain for the white balance. The multiplier for B is "sets value" / 128 (The gain is time 1 when obtains 128).

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

D. StTrg_SetWhiteBalanceGain

1. Description:

This command sets the gain for the white balance.

2. Syntax:

```
BOOL StTrg_SetWhiteBalanceGain(HANDLE hCamera, WORD wWBGainR, WORD wWBGainGr, WORD wWBGainGb, WORD wWBGainB)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera

wWBGainR: This parameter sets the R gain for the white balance. The multiplier for R is "sets value" / 128 (The gain is time 1 when sets to 128). Sets range is 0 to 640.

wWBGainGr: This parameter sets the Gr (which is the G on the row in a R) gain for the white balance. The multiplier for Gr is "sets value" / 128 (The gain is time 1 when sets to 128). Sets range is 0 to 640.

wWBGainGb: This parameter sets the Gb (which is the G on the row in a B) gain for the white balance. The multiplier for Gb is "sets value" / 128 (The gain is time 1 when sets to 128). Sets range is 0 to 640.

wWBGainB: This parameter sets the B gain for the white balance. The multiplier for B is "sets value" / 128 (The gain is time 1 when sets to 128). Sets range is 0 to 640.

4. Return Code:

TRUE: The function was successful.

FALSE: The function failed.

E. StTrg_WhiteBalance

1. Description:

This command processes the white balance. When this function is used with a monochrome type the white balance is not processed.

2. Syntax:

```
BOOL StTrg_WhiteBalance(HANDLE hCamera, PBYTE pbyteRaw)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pbyteRaw: This parameter obtains the raw image by TransferEndCallbackFunc function. *pbyteRaw is overwrite after this process.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

Section 13: Image Process

A. StTrg_GetGammaModeEx

1. Description:

This command obtains the gamma mode of the image.

2. Syntax:

BOOL StTrg_GetGammaModeEx(HANDLE hCamera, BYTE byteGammaTarget, PBYTE pbyteGammaMode, PWORD pwGamma, PSHORT pshtBrightness, PBYTE pbyteContrast, PBYTE pbyteGammaTable)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteGammaTarget: This parameter obtains the gamma target, which is the one of the noise reduction modes listed in the chart below:

Gamma Target	Descriptions
STCAM_GAMMA_TARGET_Y	Gamma adjusts for the luminance
STCAM_GAMMA_TARGET_R	Gamma adjusts for R
STCAM_GAMMA_TARGET_GR	Gamma adjusts for Gr
STCAM_GAMMA_TARGET_GB	Gamma adjusts for Gb
STCAM_GAMMA_TARGET_B	Gamma adjusts for B

*pbyteGammaMode: This parameter obtains the gamma mode, which is the one of the noise reduction modes listed in the chart below:

Gamma Mode	Descriptions
STCAM_GAMMA_OFF	No gamma adjustment
STCAM_GAMMA_ON	Gamma adjusts with "*pbyteGamma"
STCAM_GAMMA_REVERSE	Gamma adjusts with "*pbyteGamma" (Reverse luminance)
STCAM_GAMMA_TABLE	Gamma adjusts with "*pbyteGammaTable"

*pwGamma: This parameter obtains the gamma value. The setting value is specified as "actual gamma value multiplied by 100." This is activated only when "STCAM_GAMM_ON" or "STCAM_GAMMA_REVERSE" is selected for the gamma mode.

*pshtBrightness: This parameter obtains the offset.

*pbyteContrast: This parameter obtains the contrast.

*pbyteGammaTable: This parameter obtains the gamma table for the gamma table mode. This is only activated when "STCAM_GAMMA_TABLE" is selected for the gamma mode.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

B. StTrg_SetGammaModeEx

1. Description:

This command sets the gamma mode of the image.

2. Syntax:

BOOL StTrg_SetGammaModeEx(HANDLE hCamera, BYTE byteGammaTarget, BYTE byteGammaMode, WORD wGamma, SHORT shtBrightness, BYTE byteContrast, PBYTE pbyteGammaTable)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteGammaTarget: This parameter sets the gamma target, which is the one of the noise reduction modes listed in the following chart:

Gamma Target	Descriptions
STCAM_GAMMA_TARGET_Y	Gamma adjusts for the luminance
STCAM_GAMMA_TARGET_R	Gamma adjusts for R
STCAM_GAMMA_TARGET_GR	Gamma adjusts for Gr
STCAM_GAMMA_TARGET_GB	Gamma adjusts for Gb
STCAM_GAMMA_TARGET_B	Gamma adjusts for B

byteGammaMode: This parameter sets the gamma mode, which is the one of the noise reduction modes listed in the following chart:

Gamma Mode	Descriptions
STCAM_GAMMA_OFF	No gamma adjustment
STCAM_GAMMA_ON	Gamma adjusts with <code>**pbyteGamma</code>
STCAM_GAMMA_REVERSE	Gamma adjusts with <code>**pbyteGamma</code> (Reverse luminance)
STCAM_GAMMA_TABLE	Gamma adjusts with <code>**pbyteGammaTable</code>

wGamma: This parameter sets the gamma value. The setting value is specified as "actual gamma value multiplied by 100." This is activated only when "STCAM_GAMM_ON" or "STCAM_GAMMA_REVERSE" is selected for the gamma mode. The set range of the gamma is from 1 to 500.

shtBrightness: This parameter sets the offset. The set range is from -255 to 255.

byteContrast: This parameter sets the contrast. The set range is from 0 to 127

*pbyteGammaTable: This parameter sets the gamma table for the gamma table mode. This is activates only select "STCAM_GAMMA_TABLE" for the gamma mode. Set to "NULL" when not selecting STCAM_GAMMA_TABLE.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

C. StTrg_RawColorGamma

1. Description:

This command processes the gamma adjustment for the raw color image. It is necessary to setup the gamma mode with "StTrg_SetGammaModeEx" before using this process.

2. Syntax:

BOOL StTrg_RawColorGamma(HANDLE hCamera, DWORD dwWidth, DWORD dwHeight, WORD wColorArray, PBYTE pbyteRaw)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

wColorArray: This parameter sets a CCD type for the camera, which is the one of the CCD type listed in the chart below:

CCD Type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RGGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BGrGrR

*pbyteRaw: This parameter obtains the raw image. *pbyteRaw is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

D. StTrg_BGRGamma

1. Description:

This command processes the gamma adjustment for the BGR color image. It is necessary to setup the gamma mode with "StTrg_SetGammaModeEx" before using this process.

2. Syntax:

BOOL StTrg_BGRGamma(HANDLE hCamera, DWORD dwReductionMode, DWORD dwWidth, DWORD dwHeight, DWORD dwPixelFormat, PBYTE pbyteBGR)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

*pbyteBGR: This parameter obtains the BGR image. *pbyteBGR is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

E. StTrg_MirrorRotation

1. Description:

This command processes the mirror image and the rotation of the image. The image size or the color order may change after this process.

2. Syntax:

BOOL StTrg_MirrorRotation(BYTE byteMirrorMode, BYTE byteRotationMode, PDWORD pdwWidth, PDWORD pdwHeight, PWORD pwColorArray, PBYTE pbyteRaw)

3. Argument:

byteMirrorMode: This parameter sets the mirror mode, which is the one of the mirror modes listed in the chart below:

Mirror Mode	Descriptions
STCAM_MIRROR_OFF	Normal image
STCAM_MIRROR_HORIZONTAL	Horizontally flipped image
STCAM_MIRROR_VERTICAL	Vertically flipped image
STCAM_MIRROR_HORIZONTAL_VERTICAL	Horizontally and vertically flipped image 180 deg. rotated image from normal image

byteRotationMode: This parameter sets the rotation mode, which is the one of the rotation modes listed in the chart below:

Rotation Mode	Descriptions
STCAM_ROTATION_OFF	Normal image
STCAM_ROTATION_CLOCKWISE_90	90 degrees clockwise rotation of the normal image
STCAM_ROTATION_COUNTERCLOCKWISE_90	90 degrees counter clockwise rotation of the normal image

*pdwWidth: This parameter sets the width of the image.

*pdwHeight: This parameter sets the height of the image.

*pwColorArray: This parameter sets a CCD type for the camera, which is the one of the CCD type listed in the chart below:

CCD Type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RGGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BGrGrR

*pbyteRaw: This parameter obtains the raw image. *pbyteRaw is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

F. StTrg_ColorInterpolation

1. Description:

This command processes the color interpolation.

2. Syntax:

BOOL StTrg_ColorInterpolation(DWORD dwWidth, DWORD dwHeight, WORD wColorArray, PBYTE pbyteRaw, PBYTE pbyteBGR, BYTE byteColorInterpolationMethod, DWORD dwPixelFormat)

3. Argument:

dwWidth: This parameter sets the image width.

dwHeight: This parameter sets the image height.

wColorArray: This parameter sets the CCD type for the camera, which is one of the CCD type listed in the chart below:

CCD type	Descriptions
STCAM_COLOR_ARRAY_MONO	Monochrome
STCAM_COLOR_ARRAY_RGGB	Color: Bayer in order RGrGbB
STCAM_COLOR_ARRAY_GRBG	Color: Bayer in order GrRBGb
STCAM_COLOR_ARRAY_GBRG	Color: Bayer in order GbBRGr
STACM_COLOR_ARRAY_BGGR	Color: Bayer in order BGbGrR

*pbyteRaw: This parameter sets the raw image, which is obtains by TransferEnd Callback Func function.

*pbyteBGR: This parameter obtains the image after the color interpolation process. It is necessary to set the good enough buffer size for set format to pbyteBGR.

byteColorInterpolationMethod: This parameter sets the color interpolation method, which is the one of the color interpolation method listed in the chart below:

Color interpolation	Descriptions
STCAM_COLOR_INTERPOLATION_NONE_MONO	Monochrome image with no color interpolation process
STCAM_COLOR_INTERPOLATION_NONE_COLOR	Color image with no color interpolation process
STCAM_COLOR_INTERPOLATION_NEAREST_NEIGHBOR	Nearest neighbor color interpolation process. Copy the nearest pixel's information.
STCAM_COLOR_INTERPOLATION_BILINEAR	Bilinear color interpolation process. Color interpolation with 8 surrounding pixel's information.
STCAM_COLOR_INTERPOLATION_BICUBIC	Bicubic color interpolation process. Color interpolation with 16 surrounding pixel's information.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below:

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

G. StTrg_GetHueSaturationMode

1. Description:

This command obtains the hue and the saturation mode of the image.

2. Syntax:

```
BOOL StTrg_GetHueSaturationMode(HANDLE hCamera, PBYTE pbyteHueSaturationMode, PSHORT pshtHue,
    PWORD pwSaturation)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pbyteHueSaturationMode: This parameter obtains the hue and the saturation mode, which is the one of the hue and the saturation modes listed in the chart below:

Hue and saturation Mode	Descriptions
STCAM_HUE_SATURATION_OFF	No hue and saturation adjustment
STCAM_HUE_SATURATION_ON	Hue and saturation adjusts with hue gain and saturation gain

*pshtHue: This parameter obtains the hue value.

*pwSaturation: This parameter obtains the saturation of the color.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

H. StTrg_SetHueSaturationMode

1. Description:

This command sets the hue and the saturation mode of the image.

2. Syntax:

```
BOOL StTrg_SetHueSaturationMode(HANDLE hCamera, BYTE byteHueSaturationMode, SHORT shtHue, WORD
    wSaturation)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteHueSaturationMode: This parameter sets the hue and the saturation mode, which is the one of the hue and the saturation modes listed in the chart below:

Hue and saturation Mode	Descriptions
STCAM_HUE_SATURATION_OFF	No hue and saturation adjustment
STCAM_HUE_SATURATION_ON	Hue and saturation adjusts with hue gain and saturation gain

shtHue: This parameter sets the hue value change amount in "number of degree X 10". (i.e., When the hue phase change is 180 degrees, the set value should be 1800. The range of the set value is between -1800 and 1800. Set "0" if there is no change of hue phase.

wSaturation: This parameter sets the changing amount of color saturation. The range of the set value is from 0 to 200.

Set "100" if there is no change of color saturation.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

I. StTrg_GetColorMatrix

1. Description:

This command obtains the color matrix of the image.

2. Syntax:

BOOL StTrg_GetColorMatrix(HANDLE hCamera, PBYTE pbyteColorMatrixMode, PSHORT pshtColorMatrix)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

*pbyteColorMatrixMode: This parameter obtains the color matrix mode, which is the one of the color matrix modes listed in the chart below:

Color Matrix Mode	Descriptions
STCAM_COLOR_MATRIX_OFF	No color matrix adjustment
STCAM_COLOR_MATRIX_CUSTOM	Color matrix adjusts with customer's table

*pshtColorMatrix: This parameter obtains the color matrix for the image; it is "set value" / 100.

The color matrix formulas are as follows:

$$R' = (R \times (\text{shtColorMatrix}[0]) + G \times (\text{shtColorMatrix}[1]) + B \times (\text{shtColorMatrix}[2]) + (\text{shtColorMatrix}[3])) / 100$$

$$G' = (R \times (\text{shtColorMatrix}[4]) + G \times (\text{shtColorMatrix}[5]) + B \times (\text{shtColorMatrix}[6]) + (\text{shtColorMatrix}[7])) / 100$$

$$B' = (R \times (\text{shtColorMatrix}[8]) + G \times (\text{shtColorMatrix}[9]) + B \times (\text{shtColorMatrix}[10]) + (\text{shtColorMatrix}[11])) / 100$$

(Note.1: R', B', and G' are RGB after color matrix process)

(Note.2: R, B, and G are RGB before color matrix process)

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed

J. StTrg_SetColorMatrix

1. Description:

This command sets the color matrix of the image.

2. Syntax:

BOOL StTrg_SetColorMatrix(HANDLE hCamera, BYTE byteColorMatrixMode, PSHORT pshtColorMatrix)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteColorMatrixMode: This parameter sets the color matrix mode, which is the one of the color matrix modes listed in the chart below:

Color Matrix Mode	Descriptions
STCAM_COLOR_MATRIX_OFF	No color matrix adjustment
STCAM_COLOR_MATRIX_CUSTOM	Color matrix adjusts with customer's table

*pshtColorMatrix: This parameter sets the color matrix for the image; it is "set value" / 100.

The color matrix formulas are as follows:

$$R' = (R \times (\text{shtColorMatrix}[0]) + G \times (\text{shtColorMatrix}[1]) + B \times (\text{shtColorMatrix}[2]) + (\text{shtColorMatrix}[3])) / 100$$

$$G' = (R \times (\text{shtColorMatrix}[4]) + G \times (\text{shtColorMatrix}[5]) + B \times (\text{shtColorMatrix}[6]) + (\text{shtColorMatrix}[7])) / 100$$

$$B' = (R \times (\text{shtColorMatrix}[8]) + G \times (\text{shtColorMatrix}[9]) + B \times (\text{shtColorMatrix}[10]) + (\text{shtColorMatrix}[11])) / 100$$

(Note.1: R', B', and G' are RGB after color matrix process)

(Note.2: R, B, and G are RGB before color matrix process)

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

K. StTrg_HueSaturationColorMatrix

1. Description:

This command processes the hue, the saturation and the color matrix of the image.

2. Syntax:

```
BOOL StTrg_HueSaturationColorMatrix(HANDLE hCamera, DWORD dwWidth, DWORD dwHeight, DWORD dwPixelFormat, PBYTE pbyteBGR)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

*pbyteBGR: This parameter obtains the BGR image. *pbyteBGR is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

L. StTrg_GetSharpnessMode

1. Description:

This command obtains the edge enhancement (sharpness) mode of the image.

2. Syntax:

```
BOOL StTrg_GetSharpnessMode(HANDLE hCamera, PBYTE pbyteSharpnessMode, PWORD pwSharpnessGain, PBYTE pbyteSharpnessCoring)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteSharpnessMode: This parameter gets the sharpness mode, which is the one of the sharpness modes listed in the chart below:

Sharpness Mode	Descriptions
STCAM_SHARPNESS_OFF	No edge enhancement (sharpness) adjustment
STCAM_SHARPNESS_ON	Edge enhancement (sharpness) adjusts with sharpness gain and coring

*pwSharpnessGain: This parameter obtains the sharpness gain of the image.

*pbyteSharpnessCoring: This parameter obtains the sharpness coring of the image.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

M. StTrg_SetSharpnessMode

1. Description:

This command sets the edge enhancement (sharpness) mode of the image.

2. Syntax:

BOOL StTrg_SetSharpnessMode(HANDLE hCamera, BYTE byteSharpnessMode, WORD wSharpnessGain, BYTE byteSharpnessCoring)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

byteSharpnessMode: This parameter sets the sharpness mode, which is the one of the sharpness modes listed in the chart below:

Sharpness Mode	Descriptions
STCAM_SHARPNESS_OFF	No edge enhancement (sharpness) adjustment
STCAM_SHARPNESS_ON	Edge enhancement (sharpness) adjusts with sharpness gain and coring

wSharpnessGain: This parameter sets the sharpness gain of the image. The set range of the sharpness gain is 0 to 500. The edge enhancement becomes stronger when this parameter is increased.

byteSharpnessCoring: This parameter sets the sharpness coring of the image. The set range of the sharpness coring is 0 to 255. The small edge (including the noise) enhancement becomes suppress when this parameter is increased.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

N. StTrg_Sharpness

1. Description:

This command processes the edge enhancement (sharpness) of the image. It is necessary to setup the sharpness mode with "StTrg_SetSharpnessMode" before using this process.

2. Syntax:

BOOL StTrg_Sharpness(HANDLE hCamera, DWORD dwWidth, DWORD dwHeight, DWORD dwPixelFormat, PBYTE pbyteGrayOrBGR)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

dwWidth: This parameter sets the width of the image.

dwHeight: This parameter sets the height of the image.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

*pbyteGrayOrBGR: This parameter obtains the BGR (or monochrome) image. *pbyteGrayOrBGR is overwritten after this process.

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

Section 14: Image Saving Process

A. StTrg_SaveImageA

1. Description:

This command saves the image file to a BMP format (This is for Visual Basic).

2. Syntax:

BOOL StTrg_SaveImageA(DWORD dwWidth, DWORD dwHeight, DWORD dwPixelFormat, PBYTE pbyteGrayOrBGR, PCSTR pszFileName, DWORD dwParam)

3. Argument:

dwWidth: This parameter sets the image width.

dwHeight: This parameter sets the image height.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

*pbyteGrayOrBGR: This parameter sets the image data.

*pszFileName: This parameter sets the file name which includes a “bmp” extension

dwParam: This parameter sets to 0.

4. Return Code:

TRUE: The function is successful.

FALSE: The function failed.

A. StTrg_SaveImageW

1. Description:

This command saves the image file to a BMP format.

2. Syntax:

BOOL StTrg_SaveImageW(DWORD dwWidth, DWORD dwHeight, DWORD dwPixelFormat, PBYTE pbyteGrayOrBGR, PCWSTR pszFileName, DWORD dwParam)

3. Argument:

dwWidth: This parameter sets the image width.

dwHeight: This parameter sets the image height.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

*pbyteGrayOrBGR: This parameter sets the image data.

*pszFileName: This parameter sets the file name which includes a “bmp” extension.

dwParam: This parameter sets to 0.

4. Return Code:

TRUE: The function is successful.

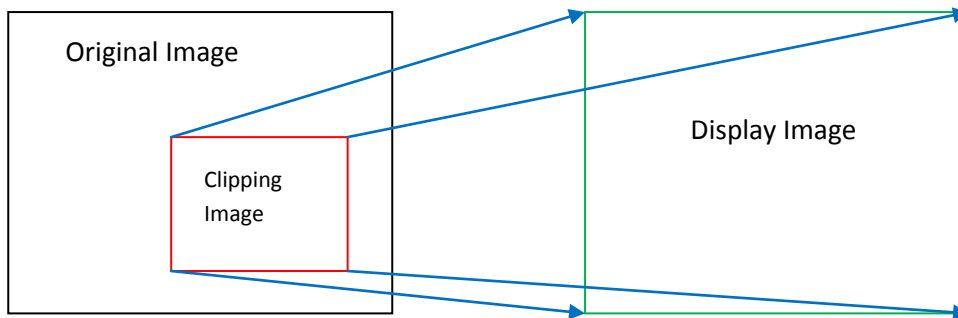
FALSE: The function failed.

Section 15: Display Image

A. StTrg_Draw

1. Description:

This command displays the image with the device context. The set values of dwOrgWidth and dwOrgHeight compose the size of the original image. The set values of SrcOffsetX, SrcOffsetY, dwSrcWidth and dwSrcHeight compose the size of the clipping image from original image. The set values of DestOffsetX, DestOffsetY, dwDestWidth and dwDestHeight compose the size of the display image for the clipping image. If the values to “dwDestWidth” and “dwDestHeight” (from display image) are larger than “dWSrcWidth” and “dWSrcHeight” (from clipping image), the display image will be an expansion of the clipping image.



2. Syntax:

```
BOOL StTrg_Draw(HANDLE hCamera, HDC hdc, int DestOffsetX, int DestOffsetY, DWORD dwDestWidth, DWORD dwDestHeight, int SrcOffsetX, int SrcOffsetY, DWORD dwSrcWidth, DWORD dwSrcHeight, DWORD dwOrgWidth, DWORD dwOrgHeight, PBYTE pbyteImage, DWORD dwPixelFormat)
```

3. Argument:

hCamera: This parameter sets the control handle of the camera.

hdc: This parameter sets the device context.

DestOffsetX: This parameter sets the offset of X direction for the display image from the clipping image.

DestOffsetY: This parameter sets the offset of Y direction for the display image from the clipping image.

dwDestWidth: This parameter sets the width of the display image for the clipping image.

dwDestHeight: This parameter sets the width of the display image for the clipping image.

SrcOffsetX: This parameter sets the offset of X direction for the clipping image from the original image.

SrcOffsetY: This parameter sets the offset of Y direction for the clipping image from the original image.

dwSrcWidth: This parameter sets the width of the clipping image from the original image.

dwSrcHeight: This parameter sets the height of the clipping image from the original image.

dwOrgWidth: This parameter sets the original image width.

dwOrgHeight: This parameter sets the original image height.

*pbyteImage: This parameter sets the original image data.

dwPixelFormat: This parameter sets the pixel format, which is the one of the pixel format listed in the chart below.

Pixel format	Descriptions
STCAM_PIXEL_FORMAT_08_MONO_OR_RAW	8 bit for 1 pixel
STCAM_PIXEL_FORMAT_24_BGR	24 bit (in order B, G, R) for 1 pixel
STCAM_PIXEL_FORMAT_32_BGR	31 bit (in order B, G, R, dummy) for 1 pixel

4. Return Code:

TRUE: the function was successful.

FALSE: the function failed.

Section 16: Clear Buffer Process

A. StTrg_ClearBuffer

1. Description:

This command clears the buffer in the camera after stop the image out by StTrg_StopTransfer function. The wrong procedure (like “send the trigger signal before the image out is finished” or “send the image out signal before the exposure is finished”) is occurred while the trigger function, the wrong data remains on the buffer of the camera.

2. Syntax:

BOOL StTrg_ClearBuffer(HANDLE hCamera)

3. Argument:

hCamera: This parameter sets the control handle of the camera.

4: Return Code:

TRUE: The function is successful.

FALSE: The function failed.

IX. StTrgTest Sample Software (Command Prompt Base) Operation

This sample software is available for Visual C++6.0 (MFC) and Visual C++ 2005 (MFC)

1. Connect the USB camera to the PC and install the USB driver.
2. Double-click "StTrgTest.exe" to start the sample software.
3. Enter the key, which corresponds to the trigger mode (F: Free Running, S: Software Trigger, H: Hardware Trigger) then hit the "Enter" key.
4. Enter "B" then hit the "Enter" key, save the image that appears next time to a BMP format file.
5. Enter "W" then hit the "Enter" key, the white balance will change to "One shot white balance" and the gain for the white balance will adjust next time the image is retrieved.
6. To display the current gain for the white balance, enter "X" then hit the "Enter" key.
7. To send the trigger signal continuously during the software trigger mode, enter "S" then hit the "Enter" key.
To send a single trigger signal during the software trigger mode, enter "O" then hit the "Enter" key.
To end the software trigger mode during the software trigger mode, enter "F" then hit the "Enter" key.
8. to send the image out signal, enter "R" then hit the "Enter" key.
9. The following messages are displayed when the exposure is finished or the image out is finished.

EE	12	-	17.0ms
(A)	(B)		(C)

- (A) "EE" mean is the exposure end Callback function
- (B) Frame number
- (C) Time from sent the software trigger (or the previous image out end)

TE	12	-	16.1ms	-	Drop0	-	Ave.: 25	-	435.1ms
(D)	(E)		(F)		(G)		(H)		(I)

- (D) "TE" mean is the image out end Callback function
- (E) Frame number
- (F) Time from the exposure end
- (G) Detected the number of the drop the image
- (H) Average pixel value of the image
- (I) Time from got the first frame

10. To close the sample software, enter "E" then hit the "Enter" Key.

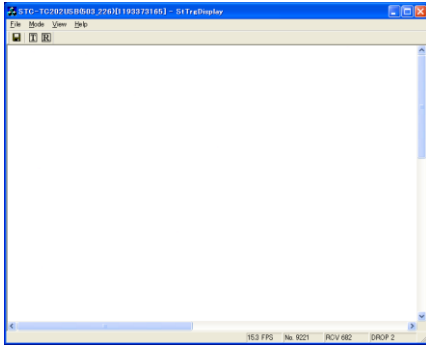
Note 1: To display the information of the commands, enter "H" then hit the "Enter" key.

Note 2: Accept the capital letter and lower-case letter for the command entering.

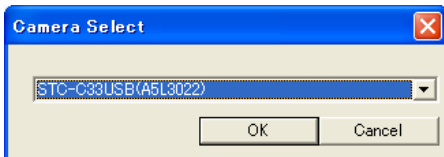
X. StTrgDisplay Sample Software (Widows base) Operation

This sample software is available for Visual C++ 6.0 (MFC), Visual C++ 2005 (MFC), Visual C# 2005, Visual Basic 2005 and Visual Basic 6.0

1. Connect the USB camera to the PC and install the USB driver.
2. Double-click “StTrgDisplay.exe” to start the sample software.



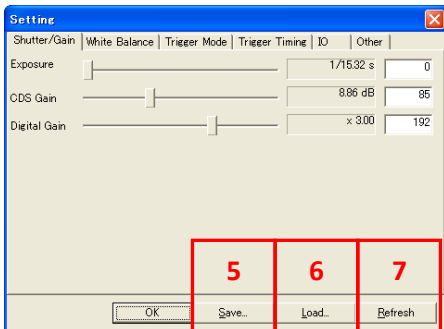
3. When the following window appears select the being used.



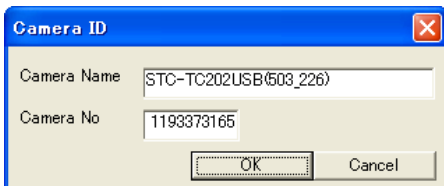
4. To change the camera's settings, select “Setting” under “Mode” in the menu bar.




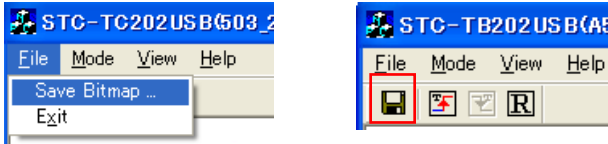
5. To save the settings to a file, select the “Save” button at the bottom right-side of the Setting window.




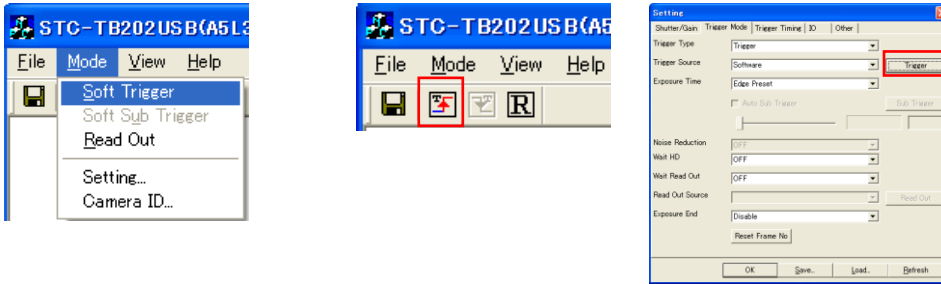
6. To load the settings from a file to the camera, select the “Load” button at the bottom right-side of the Setting window.
7. To update the current settings from the camera, select the “Refresh” button at the bottom right-side of the Setting window.
8. To change the camera ID, select “Camera ID” under “Mode” in the menu. The following window will appear:



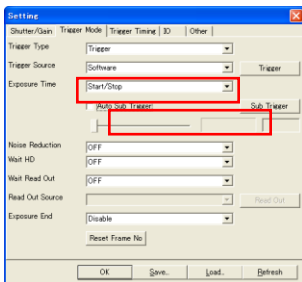
9. To save current image to a Bitmap format file, select “Save Bitmap” under “File” in the menu or select the “Save” button  in the tool bar.



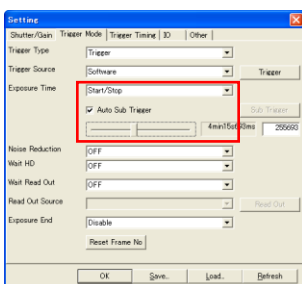
10. To send the Trigger signal to the camera, select “Soft Trigger” under “Mode” in the menu or select the trigger button  in the tool bar. Also the Trigger signal can be sends by selecting the “Trigger” button on the “Trigger Mode” tab of the setting window.




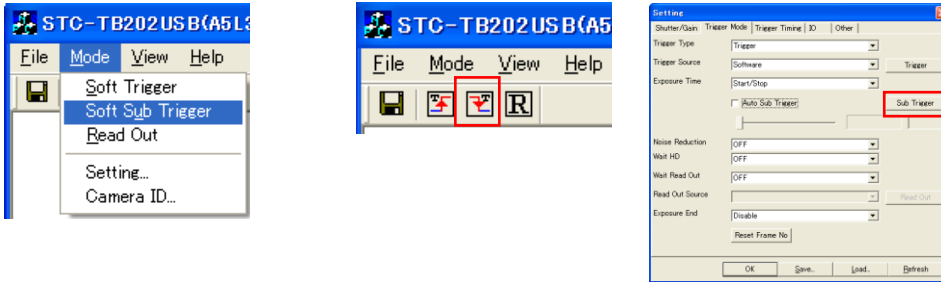
11. To use the extended exposure, select “Start/Stop” at “Exposure Time” on the “Trigger Mode” tab.




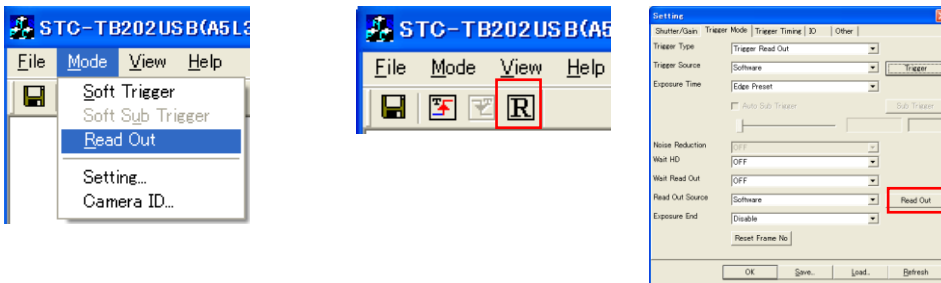
12. To use the auto extended exposure, select “Start/Stop” at “Exposure Time” and “Auto Sub Trigger” on the “Trigger Mode” tab. The exposure time can be adjusted with the slide bar.



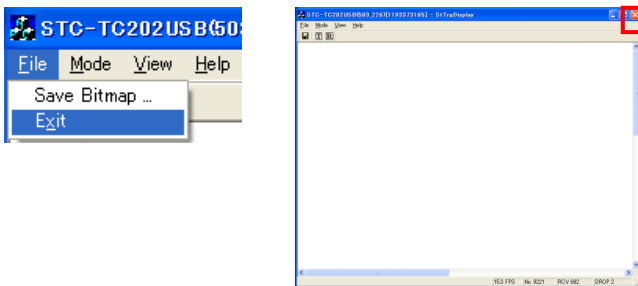
13. To send the second trigger signal to the camera to stop the exposure while in the extended exposure mode, select “Soft Sub Trigger” under “Mode” in the menu or select readout button  in the tool bar. Also, the second trigger signal can be sent by selecting the “Sub Trigger” button on the “Trigger Mode” tab of the setting window.



14. To send the Read out signal to the camera, select “Read out” under “Mode” in the menu or select readout button  in the tool bar. Also the Readout signal can be sends by selecting the “Trigger” button on the “Trigger Mode” tab of the setting window.



15. To close the sample software, select “Exit” under “File” in the menu or select the “X” button at the top of the viewing screen.



XI. StTakeRawSnapShot Sample Software (Take 10 Snap Shot) Operation

This sample software is available for Visual C++6.0 (MFC) and Visual C++ 2005 (MFC)

1. Connect the USB camera to the PC and install the USB driver.
2. Double-click “StTakeRawSnapShot.exe” to start the sample software then create 10 bitmap format files, which is 10 frame image files into the folder that has “StTrgDisplay.exe”.

XII. Frequently Asked Questions

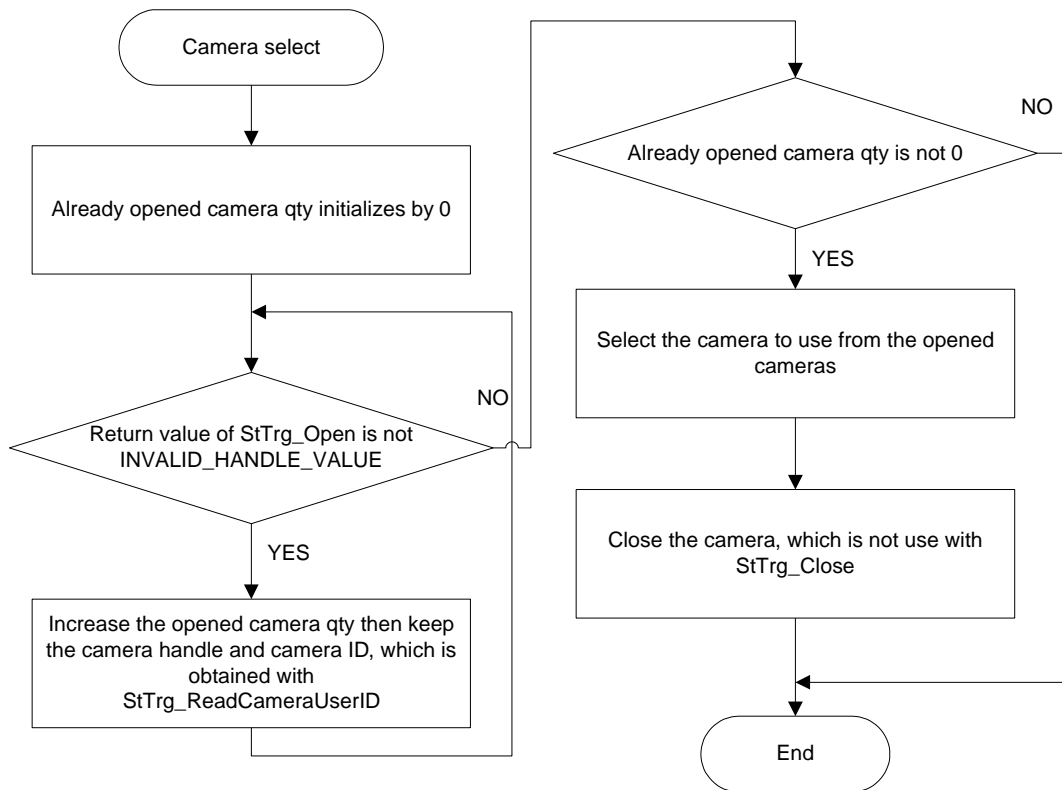
1. Got the error number, which is not defined by this SDK.

Sometimes unknown error number, which is not defined by this SDK, gets with StTrg_SetRcvErrorCallback function or the return value of the GetLastError. These error numbers, which are not defined by this SDK, are error codes of Win32 API. These errors can get by the errorlook.exe, which is attachment of the Visual Studio.

2. Select the camera from the connected multiple cameras.

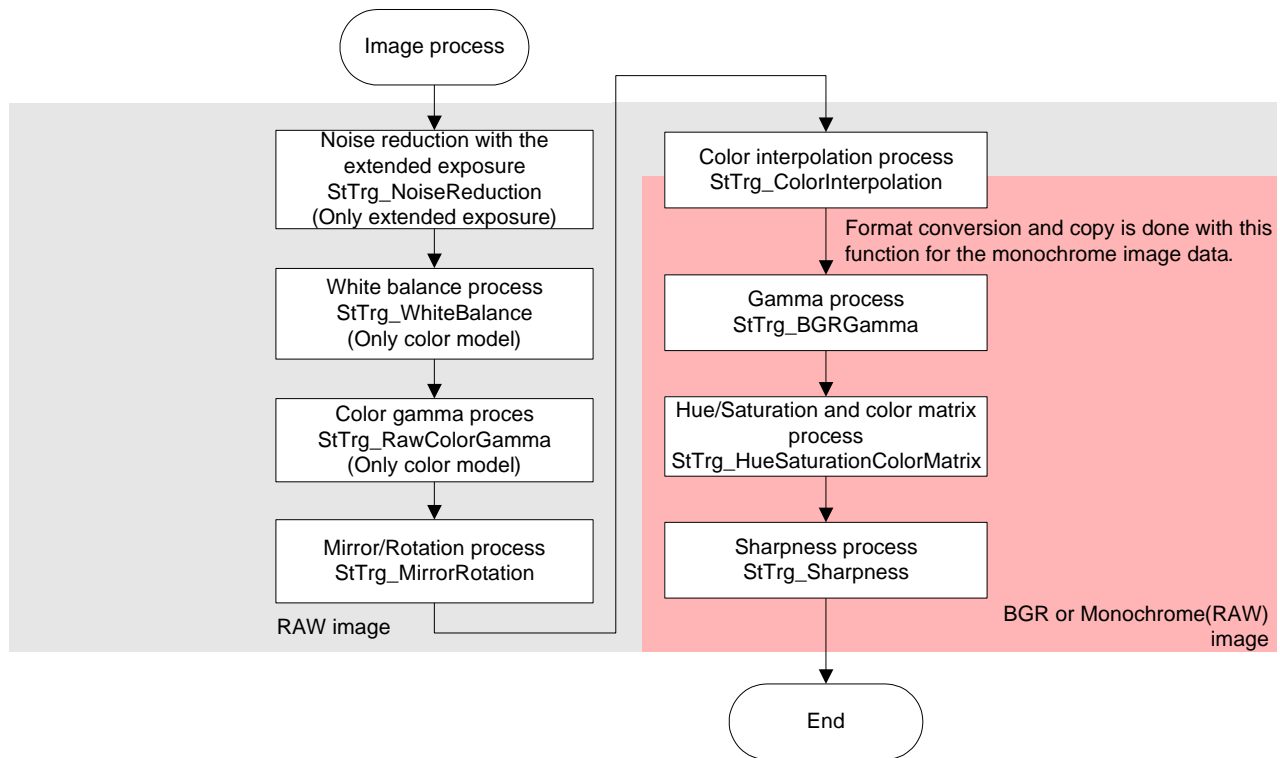
It is necessary to open all cameras that are connected to the PC then close the cameras that are not use when the multiple cameras are connected to the PC.

The sample program "StTrgDisplay" is following this order to select the camera.



3. Sequence of the image process.

Please refer the below diagram for the sequence of the image process.
The image size may change after the color interpolation process.



4. If I receive an error that is not explained in this document, how do I know what kind of error occurred?

StCam_GetLatestError obtains the error number which is received from the SDK and the OS.
Error numbers from the OS are not explained in this document. Therefore, please refer to "Errlook.h" in the Microsoft SDK for information of the error number from the OS.
You can get the error information with "Errlook.exe" in the Visual C++.

You can find "Errlook.exe" in the following locations:

- For Visual C++ 6.0: C:\Program Files\Microsoft Visual Studio\Common\Tools\errlook.exe
- For Visual C++ 2003: C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\Tools\errlook.exe
- For Visual C++ 2005: C:\Program Files\Microsoft Visual Studio 8\Common7\Tools\errlook.exe
- For Visual C++ 2008: C:\Program Files\Microsoft Visual Studio 9\Common7\Tools\errlook.exe
- For Visual C++ 2010: C:\Program Files\Microsoft Visual Studio 10\Common7\Tools\errlook.exe

XIII. The trigger function SDK example operating

Camera open: (1-A) **StTrg_Open**

↓

Camera Information:

DLL version information: (2-A) **StTrg_GetDllVersion**

Version information: (2-B) **StTrg_GetCameraVersion**

Camera Functions: (2-H) **StTrg_HasFunction**

CCD color array information: (2-C) **StTrg_GetColorArray**

Camera ID: (2-D) **StTrg_ReadCameraUserIDA**, (2-E) **StTrg_ReadCameraUserIDW**, (2-F) **StTrg_WriteCameraUserIDA**,
(2-G) **StTrg_WriteCameraUserIDW**

Settings

Save and load settings: (3-C) **StTrg_WriteSettingFileW**, (3-D) **StTrg_ReadSettingFileW**

Save and load settings (For Visual Basic): (3-A) **StTrg_WriteSettingFileA**, (3-B) **StTrg_ReadSettingFileA**

Scanning mode: (4-A) **StTrg_GetScanMode**, (4-B) **StTrg_SetScanMode**

Bit setting for the pixel: (4-C) **StTrg_GetEnableTransferBitsPerPixel**, (4-D) **StTrg_GetTransferBitsPerPixel**,
(4-E) **StTrg_SetTransferBitsPerPixel**

Clock speed: (4-G) **StTrg_GetClock**, (4-H) **StTrg_SetClock**

Exposure time: (4-I) **StTrg_GetFrameClock**, (4-J) **StTrg_GetExposureClock**, (4-K) **StTrg_SetExposureClock**

Gain: (4-L) **StTrg_GetGain**, (4-M) **StTrg_SetGain**

Digital gain: (4-N) **StTrg_GetDigitalGain**, (4-O) **StTrg_SetDigitalGain**

Trigger mode: (5-A) **StTrg_GetTriggerMode**, (5-B) **StTrg_SetTriggerMode**

IO port signal direction: (6-A) **StTrg_GetIOPinDirection**, (6-B) **StTrg_SetIOPinDirection**

IO port polarity: (6-C) **StTrg_GetIOPinPolarity**, (6-D) **StTrg_SetIOPinPolarity**

IO port mode: (6-E) **StTrg_GetIOPinMode**, (6-F) **StTrg_SetIOPinStatus**

IO port status: (6-G) **StTrg_GetIOPinStatus**, (6-H) **StTrg_SetIOPinStatus**

DIP Switch status: (6-I) **StTrg_GetSwStatus**

LED status: (6-J) **StTrg_GetLEDStatus**, (6-K) **StTrg_SetLEDStatus**

Timing: (5-C) **StTrg_GetTriggerTiming**, (5-D) **StTrg_SetTriggerTiming**

Reset frame counter: (2-I) **StTrg_ResetCounter**

White balance: (12-A) **StTrg_GetWhiteBalanceMode**, (12-B) **StTrg_SetWhiteBalanceMode**,
(12-C) **StTrg_GetWhiteBalanceGain**, (12-D) **StTrg_SetWhiteBalanceGain**

Gamma: (13-A) **StTrg_GetGammaModeEx**, (13-B) **StTrg_SetGammaModeEx**

Hue and saturation: (13-G) **StTrg_GetHueSaturationMode**, (13-H) **StTrg_SetHueSaturationMode**

Color matrix: (13-I) **StTrg_GetColorMatrix**, (13-J) **StTrg_SetColorMatrix**

Sharpness: (13-L) **StTrg_GetSharpnessMode**, (13-M) **StTrg_SetSharpnessMode**

Timeout: (7-A) **StTrg_SetTimeOut**, (7-B) **StTrg_SetTimeOut**

Set the exposure end Callback function: (8-A) **StTrg_SetExposureEndCallback**

Set the image out end Callback function: (8-B) **StTrg_SetTransferEndCallback**

Set the error receives callback function: (8-C) **StTrg_SetRcvErrorCallback**

Set the buffer for raw image: (9-B) **StTrg_SetRawSnapshotBufferCount**

Set the window for the message from the SDK: (9-C) **StTrg_SetRcvMsgWnd**

↓

The image out start: (10-A) **StTrg_StartTransfer**

↓

Send the trigger signal: (5-E) **StTrg_SoftTrigger**

Send the second trigger signal: (5-F) **StTrg_SoftSubTrigger**

Send the image out signal: (5-G) **StTrg_TriggerReadOut**

Use the callback function

Exposure end detects

The image out end

8Bit image convert process: (4-F) **StTrg_NoiseReduction**

Noise reduction process: (11-A) **StTrg_NoiseReduction**

White balance process: (12-E) **StTrg_WhiteBalance**

Color gamma process: (13-C) **StTrg_RawColorGamma**

Mirror and rotation process: (13-E) **StTrg_MirrorRotation**

Color interpolation process: (13-F) **StTrg_ColorInterpolation**

Gamma process: (13-D) **StTrg_BRGamma**

Hue, saturation and color matrix process: (13-K) **StTrg_HueSaturationColorMatrix**

Sharpness process: (13-N) **StTrg_Sharpness**

Save the image process (For Visual Basic): (14-A) **StTrg_SaveImageA**

Save the image process: (14-B) **StTrg_SaveImageW**

Error receives

Get raw image: (12-A) **StTrg_TakeRawSnapShot**

Display image: (13-A) **StTrg_Draw**

↓

The image out stop: (10-B) **StTrg_StopTransfer**

↓

Camera close: (1-B) **StTrg_Close**

Other: (16-A) **StTrg_ClearBuffer**

Senor Technologies America, Inc.

1345 Valwood Parkway, Suite 320
Carrollton, Texas 75006-6891
TEL (972)481-9223 FAX(972) 481-9209
URL <http://www.sentechamerica.com/>

Sensor Technology Co., Ltd.

7F, Harada Center Building
9-17, Naka cho 4 chome
Atsugi-city, Kanagawa
243-0018 Japan
TEL +81-46-295-7061 FAX +81-46-295-7066
URL <http://www.sentech.co.jp/>

Taiwan Sensor Technology, Inc.

3F-6, No. 9, Aiguo W, Rd., Jhong Jheng District
Taipei City 100, Taiwan, R.O.C.
TEL 886-2-2383-2331 FAX 886-2-2370-8775
EMAIL sentech0501@yahoo.com.tw