

# **UC Camera Series Software Development Kit**

*UC360.DLL (ver.2.2, CMOS WVGA Color version Camera)*  
*UC361.DLL (ver.2.2, CMOS WVGA B/W version Camera)*  
*UC1320.DLL (ver.2.2, CMOS 1.3MP Color version Camera)*  
*UC1311.DLL (ver.2.2, CMOS 1.3MP B/W version Camera)*  
*UC3010.DLL (ver.2.2, CMOS 3.1MP Color version Camera)*  
*UC5010.DLL (ver.2.2, CMOS 5.1MP Color version Camera)*  
*UC9010.DLL (ver.2.2, CMOS 9.1MP Color version Camera)*  
*UC800.DLL (ver.2.0, Sony CCD XGA Color version Camera)*  
*UC1450.DLL (ver.2.0, Sony CCD 1.4MP Color version Camera)*

**Dynamic Link Library for Windows 2000, XP and VISTA, 7**

## **User's Manual** *(Edition 1.22)*

**Copyright (C) 2009**

**Function Name:** **CameraGetCameraNum**

**Purpose:** Get number of camera connected to the PC

**Parameter:** \*CamAllNum : total camera

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** Function will return the amount of connected camera (same model) to the PC

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetMultiCameraNumber(BYTE \*CamAllNum);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraInit**

**Purpose:** Initialize Camera Settings

**Parameter:** pCallbackFunction – Callback function pointer

IMODE - Image transfer mode (display or only grab into image buffer)

*IMAGEMODE\_DISPLAY\_CALLBACK=0 (Callback with Display(draw))*

*IMAGEMODE\_CALLBACK\_ONLY=1 (Callback only)*

uiResolution - Camera Resolution

UC360	UC361	UC1311	UC1320	UC3010	UC5010	UC800(CCD)	UC1450(CCD)
0: 752 x 480	0: 752 x 480	0: 1280 x 1024	0: 1280 x 1024	0: 2048 x 1536	0: 2592 x 1944	0: 1024 x 768	0: 1360 x 1024
		1: 640 x 512	1: 640 x 480	1: 1024 x 768	1: 1280 x 960		
		2: 640 x 480 ROI	2: 640 x 480 ROI	2: 640 x 480	2: 1024 x 768		
		3: 320 x 240 ROI	3: 320x240	3: 512 x 384	3: 640 x 480		
			4: 1024x768 ROI	4: 1280 x 1024 ROI	4: 1280 x 1024 ROI		

hWndDisplay - Display Handler

lpThreadparam - Video Display Creation Address

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description :** This function must be called to start the camera operation

Callback function is to be used as event when camera finished grabbing and processing an image, in both freerun and trigger mode.

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraInit(DS\_SNAP\_PROC pCallbackFunction,  
IN DS\_IMAGEMODE IMODE,  
IN DS\_RESOLUTION uiResolution,  
IN HWND hWndDisplay,  
IN LPVOID lpThreadparam,  
BYTE CamNum);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraUnInit**

**Purpose:** Uninitialize Camera Settings

**Parameter**

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** Should be called before closing program, to release the memory and deactivate the camera.

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraUnInit(void);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraPlay**

**Purpose:** Start camera operation in freerun or trigger mode

**Parameter:** SnapMode, 0=freerun, 1=software trigger, 2=hardware trigger

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraPlay(DS_SNAP_MODE SnapMode);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSoftTriggerStart**

**Purpose:** Software trigger the camera to grab an image into the buffer.

**Parameter**

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** Camera should be set into software trigger mode beforehand (use *CameraPlay* function)

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraSoftTriggerStart(void);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraStop**

**Purpose:** Stop Camera from live/trigger mode.

**Parameter**

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraStop(void);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

**Function Name:** **CameraSaveCurrentImage**

**Purpose:** Save current Image

**Parameter** strFileName - File name(with path)

FileType - Image file format, please refer to FILE\_TYPE defination

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** Save the last streaming image from the continuous video mode.

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSaveCurrentImage(LPCTSTR strFileName, BYTE FileType);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

**Function Name:** **CameraCaptureFile**

**Purpose:** Grab an image from camera and save to file

**Parameter** strFileName - File name(with path)

(if strFileName is left empty, the "file save" dialog box will be display)

FileType - Image file format, please refer to FILE\_TYPE defination

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** This function can only be called during live video (freerun)

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraCaptureFile(IN LPCTSTR strFileName, IN BYTE FileType);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetImageSize**

**Purpose:** To get the current image size

**Parameter** \*pWidth - pointer of image width

\*pHeight - pointer of image height

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetImageSize(int \*pWidth, int \*pHeight);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetAeState**

**Purpose:** Set Auto Exposure On or Off

**Parameter** bState - TRUE (Auto)  
- FALSE (Manual)

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** Auto Exposure value target (self adjust exposure value target) can be set by *CameraSetAeTarget*

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraSetAeState(IN BOOL bState);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetAeState**

**Purpose:** Get Auto Exposure State(On or Off)

**Parameter** \*pAeState - pointer to store state

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraGetAeState(BOOL *pAeState);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetAWBState**

**Purpose:** One Push Auto White Balance

**Parameter** bAWBState - TRUE (Auto White Balance)

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** This function is to obtain the white balance automatically and fixed the value.

It's not usable for setting the camera into auto white balance mode but for one-off white balance setting and fixed.

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraSetAWBState(BOOL bAWBState);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetAeTarget**

**Purpose:** Set Auto Exposure Target Value

**Parameter** uiAeTarget - AE Target Value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetAeTarget(IN BYTE uiAeTarget);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetAeTarget**

**Purpose:** Get Auto Exposure Target Value

**Parameter** \*pAeTarget - pointer to store Auto Exposure Target Value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetAeTarget(IN OUT BYTE \*pAeTarget);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetExposureTime**

**Purpose:** Set Sensor Exposure Time (Integration Time)

**Parameter** uiExposureTime - Exposure time value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetExposureTime(IN int uiExposureTime);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetExposureTime**

**Purpose:** Set Sensor Exposure Time (Integration)

**Parameter** \*pExposureTime - pointer to store Exposure Time Value

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetExposureTime(IN int \*pExposureTime);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetExposureTime**

**Purpose:** Get Sensor Exposure Time (Integration)

**Parameter** \*pExposureTime - pointer to store Exposure Time Value

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetExposureTime(IN int \*pExposureTime);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetAnalogGain**

**Purpose:** Set Hardware Gain

**Parameter** usAnalogGain - Hardware Gain Value

Return Values:STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetAnalogGain(IN USHORT usAnalogGain);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetAnalogGain**

**Purpose:** Get Hardware Gain

**Parameter** \*pAnalogGain - pointer to store Gain Value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API_DS_CAMERA_STATUS __stdcall CameraGetAnalogGain(IN USHORT *pAnalogGain);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetGain**

**Purpose:** Set RGB Channel Gain Values

**Parameter**    RGain - Red Gain  
                 GGain - Green Gain  
                 BGain - Blue Gain

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API_DS_CAMERA_STATUS __stdcall CameraSetGain(IN USHORT RGain, USHORT GGain, USHORT BGain);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetGain**

**Purpose:** Get RGB Channel Gain Values

**Parameter**    \*pRGain - pointer to store Red Gain Value  
                 \*pGGain - pointer to store Green Gain Value  
                 \*pBGain - pointer to store Blue Value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

```
UC_API_DS_CAMERA_STATUS __stdcall CameraGetGain(IN int *pRGain, int *pGGain, int *pBGain);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---



**Function Name:** **CameraSetGamma**

**Purpose:** Set GAMMA Value

**Parameter** uiGamma

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetGamma(IN BYTE uiGamma);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetGamma**

**Purpose:** Get GAMMA Value

**Parameter** \*pGamma - pointer to store Gamma Value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetGamma(IN BYTE \*pGamma);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetColorEnhancement**

**Purpose:** Enable Software Color Enhancement Feature

**Parameter** bEnable - TRUE  
- FALSE

**Return Values** :STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetColorEnhancement(IN BOOL bEnable);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetColorEnhancement**

**Purpose:** Get Software Color Enhancement Feature Status

**Parameter** \*pEnable - pointer to store Software Color Enhancement Feature Status

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetColorEnhancement(IN BOOL \*pEnable);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetSaturation**

**Purpose:** Set Saturation Value

**Parameter** uiSaturation

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** This function is only working when the *CameraColorSetEnhancement* is ON

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetSaturation(IN BYTE uiSaturation);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetSaturation**

**Purpose:** Get Saturation Value

**Parameter** \*pSaturation - pointer to store saturation value

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

**Function Declaration in VC:**

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetSaturation(IN BYTE \*pSaturation);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:**     **CameraSetContrast**

**Purpose:** Set Contrast Value

**Parameter** uiContrast

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetContrast(IN BYTE uiContrast);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:**     **CameraGetContrast**

**Purpose:** Get Contrast Value

**Parameter** \*pContrast - pointer to store Contrast value

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetContrast(IN BYTE \*pContrast);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:**     **CameraSetMirror**

**Purpose:** Set Image Mirror (Flipping)

**Parameter** uiDir - Direction, please refer to DS\_MIRROR\_DIRECTION defination

    bEnable - TRUE

        - FALSE

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetMirror(IN DS\_MIRROR\_DIRECTION uiDir, IN BOOL bEnable);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetMonochrome**

**Purpose:** Set Camera to Monochrome Mode

**Parameter** bEnable - TRUE (Monochrome)  
- FALSE (Color)

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetMonochrome(IN BOOL bEnable);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetFrameSpeed**

**Purpose:** Set Frame rate mode (high or low)

**Parameter** FrameSpeed - frame rate, please refer to DS\_FRAME\_SPEED

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Decription:** Low frame rate mode makes sensor to be high sensitivity

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraSetFrameSpeed(IN DS\_FRAME\_SPEED FrameSpeed);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetFrameSpeed**

**Purpose:** Get Frame rate mode (high or low)

**Parameter** FrameSpeed - pointer to store Frame rate mode

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API DS\_CAMERA\_STATUS \_\_stdcall CameraGetFrameSpeed(IN BYTE \*pFrameSpeed);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSaveParameter**

**Purpose:** Save current camera settings

**Parameter** Team - settings mode from 0-3

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API\_DS\_CAMERA\_STATUS \_\_stdcall CameraSaveParameter(IN DS\_PARAMETER\_TEAM Team);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraReadParameter**

**Purpose:** Load camera settings

**Parameter** Team - settings mode from 0-3

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API\_DS\_CAMERA\_STATUS \_\_stdcall CameraReadParameter(IN DS\_PARAMETER\_TEAM Team);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

**Function Name:** **CameraGetCurrentParameterTeam**

**Purpose:** Read current settings mode number

**Parameter** \*pTeam - pointer to store settings mode number

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API\_DS\_CAMERA\_STATUS \_\_stdcall CameraGetCurrentParameterTeam(IN BYTE \*pTeam);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraSetStrobeMode**

**Purpose:** Set strobe output delay time and duration

**Parameter** Delay : Strobe Delay Time

Duration : Strobe Duration

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

*UC\_API\_DS\_CAMERA\_STATUS \_\_stdcall CameraSetStrobe(UINT Delay, UINT Duration);*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

- This function is only available for UC cameras based on CCD sensor. CMOS sensor uses GPIO function to Generate the strobe signal.
-

**Function Name:** **CameraSetStrobeMode**

**Purpose:** Set strobe output delay time and duration

**Parameter** \*Delay : Strobe Delay Time

\*Duration : Strobe Duration

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:**

*Function Declaration in VC:*

```
UC_API DS_CAMERA_STATUS __stdcall CameraGetStrobe(UINT *Delay, UINT *Duration);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

- This function is only available for UC cameras based on CCD sensor. CMOS sensor uses GPIO function to Generate the strobe signal.

---

**Function Name:** **CameraSetTriggerDelay**

**Purpose:** Set delay time after receiving trigger signal before grabbing an image

**Parameter** DelayTimeMs : trigger delay time in milisecond

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:** To set the delay (waiting) time before trigger

*Function Declaration in VC:*

```
UC_API DS_CAMERA_STATUS __stdcall CameraSetTriggerDelay(USHORT DelayTimeMs );
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

---

**Function Name:** **CameraSetTriggerDelay**

**Purpose:** Set delay time after receiving trigger signal before grabbing an image

**Parameter** \*pDelayTimeMs : trigger delay time in milisecond

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Description:** To set the delay (waiting) time before trigger

*Function Declaration in VC:*

```
UC_API DS_CAMERA_STATUS __stdcall CameraGetTriggerDelay(USHORT *pDelayTimeMs );
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

---

**Function Name:** **CameraGetBitmap**

**Purpose:** Copy camera image data into user image buffer (picData)

**Parameter** \*picData - pointer to store image data  
FlipMode - Flip the image (upside down: 0-normal, 1-flip)

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** This should be used when not in freerun mode, instead, please use CameraCopyBitmap function below.

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraGetBitmap(BYTE *picData, BYTE FlipMode);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraCopyBitmap**

**Purpose:** Copy camera image buffer data (srcData) into user image buffer (picData)

**Parameter** \*srcData - pointer to camera image buffer  
\*picData - pointer to store image data

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** This function can be used in any mode, freerun or trigger, and should be called in *callback* function which works as an event when camera finished grabbing and processing an image, in both freerun and trigger mode.

**Function Declaration in VC:**

```
UC_API DS_CAMERA_STATUS __stdcall CameraCopyBitmap(BYTE *srcData, BYTE *picData);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

---

**Function Name:** **CameraGetRowTime**

**Purpose:** Read the current pixels row scanning time

**Parameter**

**Return Values:** STATUS\_OK if success or else return error code accordingly

**Description:** unit in micro second (us); used for exposure time calculation

**Function Declaration in VC**

```
UC_API DS_CAMERA_STATUS __stdcall CameraGetRowTime(IN UINT *pRowTime);
```

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC, C#, Delphi, VB, VB.Net

**Function Name:** **CameraSetROI**

**Purpose:** Define the ROI(Region Of Interest) of the sensor.

Smaller ROI will speed up the Camera Frame Rate

**Parameter:** ROI\_OffX : Number of pixels Offset from the left for origin pixel

ROI\_OffY : Number of pixels Offset from the top for origin pixel

ROI\_Width : ROI Width in number of pixel

ROI\_Height : ROI Height in number of pixel

**Return Values:**STATUS\_OK if success or else return error code accordingly

**Decription:** ROI Size (height and width) should be always in multiply of 4

***Function Declaration in VC***

UC\_API\_DS\_CAMERA\_STATUS \_\_stdcall CameraSetROI(USHORT ROI\_OffX, USHORT ROI\_OffY, USHORT ROI\_Width, USHORT ROI\_Height);

*\* ROI function is not available for CCD Sensor models due to the CCD limitation for ROI.*

Compatible for platform: VC, C#, Delphi, BCB, VB, VB.Net

Function used in sample program available in: VC

---



## **Parameters (constant) declaration**

```
typedef enum tagDS_IMAGEMODE
```

```
{  
    IMAGEMODE_DISPLAY_CALLBACK=0,  
    IMAGEMODE_CALLBACK_ONLY,  
}DS_IMAGEMODE;
```

```
typedef enum tagDS_CAMERA_STATUS
```

```
{  
    STATUS_OK = 1, //Operation Success  
    STATUS_INTERNAL_ERROR = 0, //Internal Error  
    STATUS_NO_DEVICE_FIND = -1, //Camera Not Found  
    STATUS_NOT_ENOUGH_SYSTEM_MEMORY = -2, //Out of Memory  
    STATUS_HW_IO_ERROR = -3, //Hardware IO Error  
    STATUS_PARAMETER_INVALID = -4, //Invalid Parameter  
    STATUS_PARAMETER_OUT_OF_BOUND = -5, //Out of Bound Parameter  
    STATUS_FILE_CREATE_ERROR = -6, //Error Creating File  
    STATUS_FILE_INVALID = -7, //Invalid File Format  
  
}DS_CAMERA_STATUS;
```

```
typedef enum tagDS_FRAME_SPEED
```

```
{  
    FRAME_SPEED_NORMAL = 0,  
    FRAME_SPEED_HIGH = 1,  
}DS_FRAME_SPEED;
```

```
typedef enum tagDS_FILE_TYPE
```

```
{  
    FILE_JPG = 1,  
    FILE_BMP = 2,  
    FILE_RAW = 4,  
}DS_FILE_TYPE;
```

```
typedef enum tagDS_SNAP_MODE {
```

```
    SNAP_MODE_CONTINUOUS = 0,  
    SNAP_MODE_SOFT_TRIGGER = 1,  
    SNAP_MODE_EXTERNAL_TRIGGER = 2,  
} DS_SNAP_MODE;
```

```
typedef enum tagDS_PARAMETER_TEAM{
```

```
    PARAMETER_TEAM_A = 0,  
    PARAMETER_TEAM_B = 1,  
    PARAMETER_TEAM_C = 2,  
    PARAMETER_TEAM_D = 3,  
}DS_PARAMETER_TEAM;
```

```
typedef enum tagDS_STROBE_MODE{
```

```
    STROBE_MODE_AUTO = 0,  
    STROBE_MODE_OFF = 1,  
}DS_STROBE_MODE;
```