

See3CAM_24CUG

Extension Unit SDK API Manual



Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

<u>INTRODUCTION TO SEE3CAM_24CUG</u>	3
PREREQUISITES	3
DESCRIPTION	3
<u>SUPPORTED APIs</u>	5
BOOL INITEXTENSIONUNIT(TCHAR *USBINSTANCEID)	5
BOOL DEINITEXTENSIONUNIT()	5
BOOL GETDEVICESCOUNT(UINT32 *Cnt)	6
BOOL GETDEVICEPATHS(TCHAR **DEVICEPATHS)	7
BOOL GETDEVICENAME(TCHAR *NAME);	7
BOOL READFIRMWAREVERSION(UINT8 *PMAJORVERSION, UINT8 *PMINORVERSION1, UINT16 *PMINORVERSION2, UINT16 *PMINORVERSION3)	8
BOOL GETCAMERAUNIQUEID(TCHAR szUNIQUEID)	9
BOOL GETSPECIALEFFECTS24CUG(UINT8 *UEFFECTSMODE)	9
BOOL SETSPECIALEFFECTS24CUG(UINT8 ueFFECTSMODE)	10
BOOL GETDENOISEVALUE24CUG(UINT8 *UDENOISEVALUE)	11
BOOL SETDENOISEVALUE24CUG(UINT8 udENOISEVALUE)	11
BOOL GETEXPROID24CUG(UINT8 uExpRoiMode, UINT8 uxPos, UINT8 uyPos, UINT8 uWinSize)	12
BOOL SETEXPROID24CUG(UINT8 uExpRoiMode, UINT8 uxPos, UINT8 uyPos, UINT8 uWinSize)	13
BOOL GETQFACTOR24CUG(UINT8 *uQFactorValue)	13
BOOL SETQFACTOR24CUG(UINT8 uQFactorValue)	14
BOOL RESTOREDEFAULT24CUG()	15
BOOL GETFLIPMODE24CUG(UINT8 *uFlipMode)	15
BOOL SETFLIPMODE24CUG(UINT8 iFlipMode)	16
BOOL GETFACEDETECTIONRECT24CUG(UINT8 *uState, UINT8 *uEmbedDataState, UINT8 *uOverlayRectState)	17
BOOL SETFACEDETECTIONRECT24CUG(UINT8 uState, UINT8 uEmbedDataState, UINT8 uOverlayRectState)	17
BOOL GETEXPOSURECOMPENSATION24CUG(UINT32 *iExposureValue)	18
BOOL SETEXPOSURECOMPENSATION24CUG(UINT32 iExposureValue)	19
BOOL GETFRAMERATEVALUE24CUG(UINT8 *uFrameRate)	19
BOOL SETFRAMERATEVALUE24CUG(UINT8 uFrameRate)	20
BOOL GETSTROBEMODE24CUG(UINT8 *uStrobeMode)	20
BOOL SETSTROBEMODE24CUG(UINT8 uStrobeValue)	21
BOOL GETANTIFLICKERMODE24CUG(UINT8 *uMode)	21
BOOL SETANTIFLICKERMODE24CUG(UINT8 uMode)	22
BOOL GETSTREAMMODE24CUG(UINT8 iStreamMode, UINT8 *iAutoFunctionsLockStatus)	22
BOOL SETSTREAMMODE24CUG(UINT8 iStreamMode, UINT8 iAutoFunctionsLockStatus)	23
BOOL RESETDEVICE()	24
<u>SUPPORT</u>	25

Introduction to See3CAM_24CUG

See3CAM_24CUG is a 2.3 MP, color, global shutter, UVC compliant, USB 3.2 Gen 1 SuperSpeed camera from e-con Systems, a company with over two decades of experience in designing, developing, and manufacturing OEM cameras. It is the latest member of the See3CAM family of USB 3.2 Gen 1 SuperSpeed camera products.

See3CAM_24CUG is a 2.3 MP color camera with the S-mount (also known as M12 board lens) lens holder. The S-mount is one of the most commonly used small form-factor lens mounts for board cameras. See3CAM_24CUG is a single board solution containing the 1/2.6" AR0234 CMOS image sensor from onsemi™ with USB 3.2 Gen 1 interface. It is also backward compatible with the USB 2.0 high speed interface, albeit at lower frame rates.

See3CAM_24CUG does not require any drivers to be installed on the PC. The native UVC drivers of Windows and Linux Operating Systems (OS) will be compatible with this camera. e-con Systems also provides the sample application that demonstrates some of the features of this camera. However, this camera can utilize any DirectShow application such as Skype and so on.

This document highlights the Extension Unit APIs that are currently used in the sample application of See3CAM_24CUG.

Prerequisites

The Visual C++ Redistributable Packages install runtime components of Visual C++ libraries that are required to run applications developed using Visual Studio 2017 on a computer. These packages install runtime components of the C Runtime (CRT) and Standard C++.

- Visual C++ Redistributable for Visual Studio 2017.
- Build Environment support from Visual Studio 2005 and higher Versions.

Description

See3CAM_24CUG supports the extension unit for the custom controls which are not the part of UVC controls. e-con Systems provides an extension unit in library form and header file which contains the API declaration supported for See3CAM_24CUG. This extension unit can be linked to your application and can access the See3CAM_24CUG extension unit controls.

Note: See3CAM_24CUG is available in two variants, which are, with enclosure (See3CAM_24CUG_CHL_TC_BX) and without enclosure (See3CAM_24CUG_CHL_TC).

The side view of See3CAM_24CUG with enclosure (See3CAM_24CUG_CHL_TC_BX) and without enclosure (See3CAM_24CUG_CHL_TC) is shown in below figures.



Figure 1: Side View of See3CAM_24CUG without Enclosure



Figure 2: Side View of See3CAM_24CUG with Enclosure

Supported APIs

The details regarding the supported APIs are explained in this section.

BOOL InitExtensionUnit(TCHAR *USBInstanceId)

This function initializes extension unit of See3CAM_24CUG and it must be called first before calling any other extension unit related APIs. Before calling this API, you must get the InstanceID of the See3CAM_24CUG and store it in a buffer.

Parameters	Description	Return Values
TCHAR *USBInstanceId	Pointer of the See3CAM_24CUG InstanceID is stored.	TRUE on Success FALSE on Failure

Sample Code

```
void InitExtensionUnitCAM()
{
    TCHAR **szInstanceID;
    UINT32 DevicesCnt = 0;
    szInstanceID = (TCHAR**)malloc(DevicesCnt);
    if (InitExtensionUnit(*szInstanceID))
    {
        printf("InitExtensionUnit success");
    }
    else
    {
        printf("InitExtensionUnit Failed");
    }
}

return result;
}
```

BOOL DeinitExtensionUnit()

This function will de-initialize the extension unit of See3CAM_24CUG. If this function is called no other API will work.

Parameters	Description	Return Values
None	N/A	TRUE on Success FALSE on Failure

Sample Code

```
void DeInitHID()
{
    if (!DeinitExtensionUnit())
    {
        printf("DeinitExtensionUnit failed\r\n");
    }
}
```

BOOL GetDevicesCount(UINT32 *Cnt)

This function provides the connected number of devices

Parameters	Description	Return Values
UINT32 *Cnt	Output parameter to store Count value	TRUE on Success FALSE on Failure

Sample Code

```
void GetDevicesCount ()
{
    TCHAR **szInstanceID;
    UINT32 DevicesCnt = 0;
    if (GetDevicesCount (&DevicesCnt))
    {
        printf("GetDevicesCount success\r\n");
    }
    else
    {
        printf("GetDevicesCount failed\r\n");
    }
}
```

BOOL GetDevicePaths(TCHAR **DevicePaths)

This function provides the connected devices path

Parameters	Description	Return Values
TCHAR **DevicePaths	To get the device path lists	TRUE on Success FALSE on Failure

Sample Code

```
void GetDevicePaths ()
{
    UINT32 DevicesCnt = 0;
    szInstanceID = (TCHAR**)malloc(DevicesCnt);
    if (GetDevicePaths(szInstanceID))
    {
        printf("GetDevicePaths success \r\n");
    }
    else
    {
        printf("GetDevicePaths failed \r\n");
    }
}
```

BOOL GetDeviceName(TCHAR *Name);

This function provides the connected devices name

Parameters	Description	Return Values
TCHAR *Name	To get the connected devices name	TRUE on Success FALSE on Failure

Sample Code

```
void GetDeviceName()
{
    DeviceName = (TCHAR*)malloc(MAX_PATH * sizeof(TCHAR));
    if (GetDeviceName(DeviceName))
    {
        printf("GetDeviceName success\r\n");
    }
    else
    {
        printf("GetDeviceName failed\r\n");
    }
}
```

BOOL ReadFirmwareVersion(UINT8 *pMajorVersion, UINT8 *pMinorVersion1, UINT16 *pMinorVersion2, UINT16 *pMinorVersion3)

This function is used to get the firmware version of See3CAM_24CUG. The firmware version will be stored in the respective variables. The firmware version is displayed as **pMajorVersion. pMinorVersion1. pMinorVersion2. pMinorVersion3**, for example it will be displayed as **1.17.122.232**.

Parameters	Description	Return Values
UINT8 *pMajorVersion	Pointer to store the major version of firmware.	
UINT8 *pMinorVersion1	Pointer to store the minor version1 of firmware.	TRUE on Success FALSE on Failure
UINT16 *pMinorVersion2	Pointer to store the minor version2 of firmware.	
UINT16 *pMinorVersion3	Pointer to store the minor version3 of firmware.	

Sample Code

```
void GetFirmwareVersion()
{
    UINT8 MajorVersion = 0, MinorVersion1 = 0;
```

```

    UINT16 MinorVersion2 = 0,MinorVersion3 = 0;
    if(ReadFirmwareVersion(&MajorVersion,&MinorVersion1,
    &MinorVersion2,&MinorVersion3))
    {
        printf("ReadFirmwareVersion success
%d.%d.%d\r\n",MajorVersion, MinorVersion1,
MinorVersion2,MinorVersion3);
    }
    else
    {
        printf("ReadFirmwareVersion Failed\r\n");
    }
}

```

BOOL GetCameraUniqueID(TCHAR szUniqueID)

This function is used to get the unique ID of See3CAM_24CUG.

Parameters	Description	Return Values
TCHAR szUniqueID	Pointer to store the camera unique ID.	TRUE on Success FALSE on Failure

Sample Code

```

void GetCamUniqueID()
{
    TCHAR szUniqueID[50];
    if(!GetCameraUniqueID(szUniqueID))
    {
        printf("GetCameraUniqueID failed\r\n");
    }
}

```

BOOL GetSpecialEffects24CUG(UINT8 *uEffectsMode)

This function sends the extension unit command to get the effects mode of See3CAM_24CUG.

0x01 - Normal Mode - In this mode, the UYVY or MJPEG image stream is unprocessed from the camera.

0x04 - Black and White Effect - In this mode, thresholding is applied to the normal preview and the image stream is composed of black and white pixels.

0x07 - Grayscale Effect - In this mode, the normal preview is desaturated, and the image stream is composed of gray shades.

0x08 - Negative Effect - In this mode, the normal preview is color inversed.

0x10 - Sketch Effect - In this mode, an effect of edge dominant image stream is produced which is useful for edge detection.

Parameters	Description	Return Values
UINT8 *uEffectsMode	Pointer to store the effects mode.	TRUE on Success FALSE on Failure

Sample Code

```
void GetEffectsModeValue()
{
    UINT8 uEffectsMode = 0;
    if (!GetSpecialEffects24CUG(&uEffectsMode))
    {
        printf("GetSpecialEffects24CUG failed\r\n");
    }
}
```

BOOL SetSpecialEffects24CUG(UINT8 uEffectsMode)

This function sends the extension unit command to set the effects mode of See3CAM_24CUG.

0x01 - Normal Mode - In this mode, the UYVY or MJPEG image stream is unprocessed from the camera.

0x04 - Black and White Effect - In this mode, thresholding is applied to the normal preview and the image stream is composed of black and white pixels.

0x07 - Grayscale Effect - In this mode, the normal preview is desaturated, and the image stream is composed of gray shades.

0x08 - Negative Effect - In this mode, the normal preview is color inversed.

0x10 - Sketch Effect - In this mode, an effect of edge dominant image stream is produced which is useful for edge detection.

Parameters	Description	Return Values
UINT8 EffectsMode	The effects mode to be set in the camera.	TRUE on Success FALSE on Failure

Sample Code

```
void SetEffectsModeValue()
{
    if(!SetSpecialEffects24CUG(0x01))          //Normal mode
    {
        printf("SetSpecialEffects24CUG failed\r\n");
    }
}
```

BOOL GetDenoiseValue24CUG(UINT8 *uDenoiseValue)

This function sends the extension unit command to get the denoise value in See3CAM_24CUG.

Parameters	Description	Return Values
UINT8 *uDenoiseValue	Pointer to get the denoise value from 0x00 to 0x0F.	TRUE on Success FALSE on Failure

Sample Code

```
void GetDenoiseModeValue()
{
    UINT8 uDenoiseValue = 0;
    if(!GetDenoiseValue24CUG(&uDenoiseValue ))
    {
        printf("GetDenoiseValue24CUG failed\r\n");
    }
}
```

BOOL SetDenoiseValue24CUG(UINT8 uDenoiseValue)

This function sends the extension unit command to set the denoise value in See3CAM_24CUG.

Parameters	Description	Return Values
UINT8 uDenoiseValue	Denoise value to be set from 0x00 to 0x0F.	TRUE on Success FALSE on Failure

Sample Code

```
void SetDenoiseValue()
{
    //Range from 0x00 - 0x0F
```

```

if (!SetDenoiseValue24CUG(0x0A))
{
    printf("SetDenoiseValue24CUGfailed \r\n");
}
}

```

BOOL GetExpRoiMode24CUG(UINT8 uExpRoiMode, UINT8 uXPos, UINT8 uYPos, UINT8 uWinSize)

This function sends the extension unit command to get the exposure Region of Interest (ROI) mode and value in See3CAM_24CUG.

0x01 - Full ROI Mode - In this mode, the full region-based exposure value will be applied to the frame.

0x02 - Manual ROI Mode - In this mode, you can select the ROI and for that region, the exposure value will be applied to the entire frame. The focus window size is configured as 1-8.

0x03 - Disabled ROI Mode - In this mode, the auto exposure ROI is disabled.

Parameters	Description	Return Values
UINT8 uExpRoiMode	Pointer to store exposure ROI mode.	
UINT8 uXPos	Pointer to store X coordinate of exposure ROI.	TRUE on Success FALSE on Failure
UINT8 uYPos	Pointer to store Y coordinate of exposure ROI.	
UINT8 uWinSize	Pointer to store window size.	

Note: This function parameter returns the disable command when the auto focus is disabled through UVC.

Sample Code

```

void GetExposureRoiModeValue()
{
    UINT8 uExposureMode = 0, uXPos = 0, uYPos = 0,
uWinSize = 0;

    if (!GetExpRoiMode24CUG(&uExposureMode,
&uXPos, &uYPos, &uWinSize))

    {
        printf("GetExpRoiMode24CUGfailed \r\n");
    }
}

```

BOOL SetExpRoiMode24CUG(UINT8 uExpRoiMode, UINT8 uXPos, UINT8 uYPos, UINT8 uWinSize)

This function sends the extension unit command to set the exposure ROI mode and value in See3CAM_24CUG.

0x01 - Full ROI Mode - In this mode, full region-based exposure value will be applied to the frame.

0x02 - Manual ROI Mode - In this mode, you can select the ROI and for that region, the exposure value will be applied to the entire frame. The focus window size is configured as 1-8.

Parameters	Description	Return Values
UINT8 uExpRoiMode	Exposure ROI mode to be set.	
UINT8 uXPos	X coordinate of exposure ROI to be set from 0x00 to 0xFF.	TRUE on Success FALSE on Failure
UINT8 uYPos	Y coordinate of exposure ROI to be set from 0x00 to 0xFF.	
UINT8 uWinSize	Size of the window for which exposure ROI is to be set from 1 to 8.	

Sample Code

```
void SetExposureROIModeValue()
{
    // Manual ROI mode
    UINT8 iExpROIMode = 0x02;

    // Retrieve and send preview co-ordinates
    UINT8 uXPos = 125, uYPos = 78;

    UINT8 uWinSize = 6;           // Range 1 to 8
    if(!SetExpRoiMode24CUG(uExpROIMode ,uXPos, uYPos,
    uWinSize))
    {
        printf("SetExpRoiMode24CUG failed \r\n");
    }
}
```

BOOL GetQFactor24CUG(UINT8 *uQFactorValue)

This function sends the extension unit command to get the Q-Factor value in See3CAM_24CUG.

Parameters	Description	Return Values
<code>UINT8 *uQFactorValue</code>	The value is used as a scale factor for the quantization table. Pointer to store the Q-Factor value from 10 to 96 with default value being 96 in USB 3.0 and 84 in USB 2.0.	TRUE on Success FALSE on Failure

Sample Code

```
void GetQFactorValue()
{
    UINT8 uQFactorValue = 0;
    if (!GetQFactor24CUG(&uQFactorValue))
    {
        printf("GetQFactor24CUG failed\r\n");
    }
}
```

BOOL SetQFactor24CUG(UINT8 uQFactorValue)

This function sends the extension unit command to set the Q-Factor value in See3CAM_24CUG.

Parameters	Description	Return Values
<code>UINT8 uQFactorValue</code>	The value is used as a scale factor for the quantization table. The values can be varied from 10 to 96 with default value being 96 in USB 3.0 and 84 in USB 2.0.	TRUE on Success FALSE on Failure

Sample Code

```
void SetQFactorValue()
{
    UINT8 uQFactorValue = 14;
    if (!SetQFactor24CUG(uQFactorValue))
    {
        printf("SetQFactor24CUGfailed\r\n");
    }
}
```

BOOL RestoreDefault24CUG()

This function sends the extension unit command to restore default settings in See3CAM_24CUG.

Parameters	Description	Return Values
None	N/A	TRUE on Success FALSE on Failure

After invoking this function, the default values will be updated for the modes as shown in below table.

Table 1: See3CAM_24CUG Controls and Modes

Controls	Modes
Scene Mode	Normal
Special Effect	Normal
Auto Exposure ROI	Full ROI Mode
De-Noise	0
Q-Factor	96
Face Detection	Disabled
Flip	Disabled
Burst Length	1
Smile Detection	Disabled
Exposure Compensation	Default value will be obtained based on the resolution
Frame Rate control	Frame rate of the current resolution

Sample Code

```
void SetDefaultSettings()
{
    if (!RestoreDefault24CUG())
    {
        printf("RestoreDefault24CUG failed\r\n");
    }
}
```

BOOL GetFlipMode24CUG(UINT8 *uFlipMode)

This function sends the extension unit command to get the flip mode in See3CAM_24CUG.

0x00 - Disable Flip.

0x01 - Enable Horizontal Flip.

0x02 - Enable Vertical Flip.

0x03 - Enable both Horizontal and Vertical Flips.

Parameters	Description	Return Values
UINT8 *uFlipMode	Pointer to store the flip mode.	TRUE on Success FALSE on Failure

Sample Code

```
void GetFlipModeValue()
{
    UINT8 uFlipMode = 0;
    if (!GetFlipMode24CUG(&uFlipMode ))
    {
        printf("GetFlipMode24CUG failed\r\n");
    }
}
```

BOOL SetFlipMode24CUG(UINT8 iFlipMode)

This function sends the extension unit command to set the flip mode in See3CAM_24CUG.

0x00 - Disable Flip.

0x01 - Enable Horizontal Flip.

0x02 - Enable Vertical Flip.

0x03 - Enable both Horizontal and Vertical Flips.

Parameters	Description	Return Values
UINT8 uFlipMode	Flip mode value to be set.	TRUE on Success FALSE on Failure

Sample Code

```
void SetFlipModeValue()
{
    UINT8 uFlipMode = 0x01;
    if(!SetFlipMode24CUG(uFlipMode ))
    {
        printf("SetFlipMode24CUG failed\r\n");
    }
}
```

BOOL GetFaceDetectionRect24CUG(UINT8 *uState, UINT8 *uEmbedDataState, UINT8 *uOverlayRectState)

This function sends the extension unit command to get the face detection rectangle, embed data and overlay rectangle states in See3CAM_24CUG.

0x00 - Disable.

0x01 - Enable.

Parameters	Description	Return Values
UINT8 *uState	Pointer to store the face detection rectangle state.	
UINT8 *uEmbedDataState	Pointer to store the embed data state which can be replaced with face details at the last part of the frame.	TRUE on Success FALSE on Failure
UINT8 *uOverlayRectState	Pointer to store the overlay rectangle state. It allows you to enable or disable the overlay rectangle around the faces during face detection.	

Sample Code

```
void GetFaceDetectionRecValue()
{
    UINT8 uState = 0, uEmbedDataState = 0; UINT8
uOverlayRectState = 0;

    if(!GetFaceDetectionRect24CUG(&uState, &uEmbedDataSta
te, &uOverlayRectState))
    {
        printf("GetFaceDetectionRect24CUG
failed\r\n");
    }
}
```

BOOL SetFaceDetectionRect24CUG(UINT8 uState, UINT8 uEmbedDataState, UINT8 uOverlayRectState)

This function sends the extension unit command to set the face detection rectangle, embed data and overlay rectangle states in See3CAM_24CUG.

0x00 - Disable.

0x01 - Enable.

Parameters	Description	Return Values
UINT8 uState	Face detection state to be set.	
UINT8 uEmbedDataState	Embed data state to be set to replace with face details in the last part of the frame.	TRUE on Success FALSE on Failure
UINT8 uOverlayRectState	Overlay rectangle state to be set to enable or disable the overlay rectangle around the faces during face detection.	

Sample Code

```
void SetFaceDetectionRecValue()
{
    UINT8 uState = 0x00;
    UINT8 uEmbedDataState = 0x00, uOverlayRectState = 0x00;

    //If the State set to disable, device will discard Embed & OverlayRect value whatever you set
    if(!SetFaceDetectionRect24CUG(uState, uEmbedDataState, uOverlayRectState))
    {
        printf("SetFaceDetectionRect24CUG failed\r\n");
    }
}
```

BOOL GetExposureCompensation24CUG(UINT32 *iExposureValue)

This function sends the extension unit command to get the exposure compensation value in See3CAM_24CUG.

Parameters	Description	Return Values
UINT32 *iExposureValue	Pointer to store the exposure value from 50 to 10,00,000µs to adjust the automatically calculated exposure compensation value.	TRUE on Success FALSE on Failure

Sample Code

```
void GetExposureCompensationValue()
{
    UINT32 iExposureVal = 0;
```

```

if (!GetExposureCompensation24CUG(&iExposureVal ))
{
    printf("GetExposureCompensation24CUG failed\r\n");
}
}

```

BOOL SetExposureCompensation24CUG(UINT32 iExposureValue)

This function sends the extension unit command to set the exposure compensation value in See3CAM_24CUG.

Parameters	Description	Return Values
UINT32 iExposureValue	You can adjust the automatically calculated exposure compensation value when set value is from 50 to 10, 00,000µs.	TRUE on Success FALSE on Failure

Sample Code

```

void SetExposureCompensationValue()
{
    UINT32 iExposureVal = 15500;
    if (!SetExposureCompensation24CUG(iExposureVal ))
    {
        printf("SetExposureCompensation24CUG failed\r\n");
    }
}

```

BOOL GetFrameRateValue24CUG(UINT8 *uFrameRate)

This function sends the extension unit command to get the frame rate in See3CAM_24CUG.

Parameters	Description	Return Values
UINT8 *uFrameRate	Pointer to store the frame rate value from 1 to 120.	TRUE on Success FALSE on Failure

Sample Code

```

void GetFrameRateControlValue()
{
    UINT8 uFrameRate = 0;
    if (!GetFrameRateValue24CUG(&uFrameRate ))
}

```

```

    {
        printf("GetFrameRateValue24CUG failed\r\n");
    }
}

```

BOOL SetFrameRateValue24CUG(UINT8 uFrameRate)

This function sends the extension unit command to set the frame rate in See3CAM_24CUG.

Parameters	Description	Return Values
UINT8 uFrameRate	Frame rate value to be set from 1 to 120.	TRUE on Success FALSE on Failure

Sample Code

```

void SetFrameRateControlValue()
{
    UINT8 uFrameRate = 110;
    If(!SetFrameRateValue24CUG(uFrameRate ))
    {
        printf("SetFrameRateValue24CUG failed\r\n");
    }
}

```

BOOL GetStrobeMode24CUG(UINT8 *uStrobeMode)

This function sends the extension unit command to get the strobe mode value in See3CAM_24CUG.

0x00 - OFF.

0x01 - ON.

Parameters	Description	Return Values
UINT8 *uStrobeMode	Pointer to store the Strobe mode value.	TRUE on Success FALSE on Failure

Sample Code

```

void GetStrobeModeValue()
{
    UINT8 uStrobeValue = 0;
    if(!GetStrobeMode24CUG(&uStrobeValue))
}

```

```

    {
        printf("GetStrobeMode24CUG failed\r\n");
    }
}

```

BOOL SetStrobeMode24CUG(UINT8 uStrobeValue)

This function sends the extension unit command to set the strobe mode value in See3CAM_24CUG.

0x00 - OFF.

0x01 - ON.

Parameters	Description	Return Values
UINT8 uStrobeValue	The strobe mode value to be set.	TRUE on Success FALSE on Failure

Sample Code

```

void SetFlashControlValue()
{
    UINT8 uStrobeValue =0x01;//strobe mode ON
    if(!SetStrobeMode24CUG (uFlashValue))
    {
        printf("SetStrobeMode24CUG failed\r\n");
    }
}

```

BOOL GetAntiFlickerMode24CUG(UINT8 *uMode)

This function sends the extension unit command to get anti flicker mode in See3CAM_24CUG.

00 - Auto Flicker Mode.

01 - 50 Hz.

02 - 60 Hz.

03 - Disable/OFF Flicker Mode.

Parameters	Description	Range	Return Values
UINT8 uMode [Output]	To store the flicker mode.	uMode value must be 0 to 3.	TRUE on Success FALSE on Failure

Sample Code

```
void GettingFlickerMode()
{
    UINT8 uMode ;
    if(!GetAntiFlickerMode24CUG (&uMode) )
    {
        printf("GetAntiFlickerMode24CUG failed\r\n");
    }
}
```

BOOL SetAntiFlickerMode24CUG(UINT8 uMode)

This function sends the extension unit command to set anti flicker mode in See3CAM_24CUG.

00 - Auto Flicker Mode.

01 - 50 Hz.

02 - 60 Hz.

03 - Disable/OFF Flicker Mode.

Parameters	Description	Range	Return Values
UINT8 uMode [Input]	To store the flicker mode.	uMode value must be 0 to 3.	TRUE on Success FALSE on Failure

Sample Code

```
void SettingFlickerMode()
{
    UINT8 uMode = 1; //50 Hz Flicker mode.
    if(!SetAntiFlickerMode24CUG (uMode) )
    {
        printf("SetAntiFlickerMode24CUG failed\r\n");
    }
}
```

BOOL GetStreamMode24CUG(UINT8 *iStreamMode , UINT8 *iAutoFunctionsLockStatus)

This function sends the extension unit command to get the Stream mode in See3CAM_24CUG.

0x00 - Master mode - In this mode, you can capture still image with streaming.

0x01 - Trigger mode - In this mode, you can capture still image without streaming using the Hardware Pulse.

Parameters	Description	Return Values
UINT8 *iStreamMode	Pointer to store the stream mode.	TRUE on Success FALSE on Failure
UINT8 * iAutoFunctionsLockStatus	Pointer to store the auto functions lock status.	

Sample Code

```
Void GetStreamModeValue()
{
    UINT8 iStreamMode = 0;
    UINT8 iAutoFunctionsLockStatus = 0;
    If (!GetStreamMode24CUG(&iStreamMode, &iAutoFunctionsLockStatus))
    {
        Printf("GetStreamMode24CUG failed\r\n");
    }
}
```

BOOL SetStreamMode24CUG(UINT8 iStreamMode, UINT8 iAutoFunctionsLockStatus)

This function sends the extension unit command to set the stream mode in See3CAM_24CUG.

0x00 - Master mode – In this mode, you can capture still image with streaming.

0x01 - Trigger mode – In this mode, you can capture still image without streaming using the Hardware Pulse.

Parameters	Description	Return Values
UINT8 iStreamMode	Stream mode to be set.	TRUE on Success FALSE on Failure
UINT8 iAutoFunctionsLockStatus	Auto functions lock status to be set.	

Sample Code

```
Void SetStreamModeValue()
{
    UINT8 iStreamMode = 0x01;           //Trigger mode
```

```

UINT8 iAutoFunctionsLockStatus = 0x01;//Set the AFL
If(!SetStreamMode24CUG(iStreamMode,
iAutoFunctionsLockStatus))
{
    Printf("SetStreamMode24CUG failed\r\n");
}
}

```

BOOL ResetDevice()

This function will Reset the Connected Device.

Parameters	Description	Return Values
None	N/A	TRUE on Success FALSE on Failure

Sample Code

```

void ResetDevice()
{
    if(ResetDevice())
    {
        printf("ResetDevice Success\r\n");
    }
    else
    {
        printf("ResetDevice failed\r\n");
    }
}

```

Support

Contact Us

If you need any support on See3CAM_24CUG product, please contact us using the Live Chat option available on our website - <https://www.e-consystems.com/>

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - <https://www.e-consystems.com/create-ticket.asp>

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - <https://www.e-consystems.com/RMA-Policy.asp>

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - <https://www.e-consystems.com/warranty.asp>

Revision History

Rev	Date	Description	Author
1.0	10-Oct-2020	Initial Draft	Murali Mohan M
1.1	23-Nov-2020	Added Stream mode API, Modified Flash mode to Strobe Mode	Murali Mohan M
1.2	27-Nov-2020	Modified Exposure compensation minimum range to 100us	Murali Mohan M
1.3	19-Jan-2021	Removed the Scene mode control, modified the Stream Mode control API.	Murali Mohan M
1.4	27-Feb-2021	Modified Exposure compensation minimum range to 50us	Camera Team
1.5	01-Sep-2021	Added Box Camera Variant Details	Camera Team
1.6	20-June-2022	Added ResetDevice API	Trisha Kanna S
1.7	06-Nov-2023	Added Changes	Camera Team