See3CAM_CU27

Extension Unit SDK API Manual

Version 1.2

e-con Systems

06/12/2021

e-con Systems

Your Product Development Partner

Disclaimer

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

Contents

Introduction to See3CAM_CU27

See3CAM_CU27 is a 16.0 Megapixel, color, UVC compliant, USB 3.1 Gen 1 SuperSpeed Autofocus camera with Type-C connector from e-con Systems, a leading embedded Product Design Services Company which specializes in the advanced camera solutions. It is a USB 3.1 Gen1 SuperSpeed autofocus camera product with reversible plug and play Type-C connector interface.

See3CAM_CU27 is a 16.0 Megapixel color camera with Autofocus module. It is a one-board solution containing autofocus camera module with 1/2.8" IMX298 CMOS image sensor from SONY along with USB 3.0 interface to the host PC, through Type-A to Type-C legacy cable with reversible cable detection.

See3CAM_CU27 is a UVC compliant USB 3.1 GEN1 SuperSpeed camera with autofocus that is also backward compatible with USB 2.0 host ports and it does not require any special camera drivers to be installed in the host PC. When connected to USB 2.0 host ports, See3CAM_CU27 supports fewer resolutions and at lower frame rates.

See3CAM_CU27 is a UVC compliant camera and it 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 for See3CAM_CU27 .

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_CU27 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_CU27 .

This extension unit can be linked to your application and can access the See3CAM_CU27 extension unit controls.

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

BOOL InitExtensionUnit(TCHAR *USBInstanceID)

This function initializes extension unit of See3CAM_CU27 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_CU27 and store it in a buffer.

Parameters	Description	Return Values
TCHAR *USBInstanceID	Pointer of the See3CAM_CU27 InstanceID is stored.	TRUE on Success FALSE on Failure

Sample Code

```
void InitHID()
{
	TCHAR tzUSBInstanceID[MAX_PATH];
	if(EnumerateImageDevice(tzUSBInstanceID))
	{
		if(!InitExtensionUnit(szInstanceID))
		{
		printf("InitExtensionUnit failed\r\n")
		}
	}
}
BOOL EnumerateImageDevice(TCHAR *USBInstanceID)
{
	TCHAR devicePath[MAX_PATH]=_T("");
	HRESULT hr = S_OK;
	Bool result = false;
	ULONG cFetched;
	IMoniker *pM = NULL;
	ICreateDevEnum *pCreateDevEnum = NULL;
	IEnumMoniker *pEm = NULL;
	//Create the system device enumerator
```

```
      if(FAILED==CoCreateInstance(CLSID_SystemDeviceEnum,N
ULL,CLSCTX_INPROC_SERVER,IID_ICreateDevEnum,
(void**)&pCreateDevEnum))
      return result;
      // Obtain a class enumerator for the video
compressor category.
      if(FAILED==pCreateDevEnum->CreateClassEnumerator
(CLSID_VideoInputDeviceCategory, &pEm, 0))
      return result;
      pEm->Reset();
      //Enumerate the monikers
      while(hr = pEm->Next(1, &pM, &cFetched), hr==S_OK)
      {
            IPropertyBag *pBag=0;
            if(SUCCEEDED == pM->BindToStorage(0, 0,
IID_IPropertyBag, (void **)&pBag))
            {
                  VARIANT var;
                  var.vt = VT_BSTR;
                  //To retrieve the device path
                  if(SUCCEEDED == pBag->Read(L"DevicePath",
&var, 0))
            {
                        if(devicePath != NULL)
                        {
                        StringCbPrintf(USBInstanceID,
MAX_PATH,L"%s",devicePath);
                        OutputDebugString(USBInstanceID);
                        result = true;
                        }
                  }
                  SysFreeString(var.bstrVal);
                  pM->AddRef();
            }
            else
            {
                  result = false;
                  break;
            }
            pM->Release();
```

```
        }
        pEm->Release();
        return result;
}
```

BOOL DeinitExtensionUnit()

This function will de-initialize the extension unit of See3CAM_CU27 . 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 ReadFirmwareVersion(UINT8 *pMajorVersion, UINT8 *pMinorVersion1, UINT16 *pMinorVersion2, UINT16 *pMinorVersion3)

This function is used to get the firmware version of See3CAM_CU27. 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.	TRUE on Success FALSE on Failure
`UINT8 *pMinorVersion1`	Pointer to store the minor version1 of firmware.	
`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.%d\r\n\n",MajorVersion, MinorVersion1,
MinorVersion2,MinorVersion3);
	}
	else
	{
		printf("ReadFirmwareVersion Failed\r\n\n");
	}
}
```

BOOL ReadISPFirmwareVersionCU27(UINT8 *pMajorVersion, UINT8 *pMinorVersion1, UINT16 *pMinorVersion2, UINT16 *pMinorVersion3)

This function is used to get the ISP firmware version of See3CAM_CU27. The firmware version will be stored in the respective variables. The ISP 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.	TRUE on Success FALSE on Failure
UINT8 *pMinorVersion1	Pointer to store the minor version1 of firmware.	
UINT16 *pMinorVersion2	Pointer to store the minor version2 of firmware.	
UINT16 *pMinorVersion3	Pointer to store the minor version3 of firmware.	

Sample Code

```
void GetISPFirmwareVersionCU27()
{
	UINT8 MajorVersion = 0,MinorVersion1 = 0;
	UINT16 MinorVersion2 = 0,MinorVersion3 = 0;
	if(ReadFirmwareVersionCU27(&MajorVersion,&MinorVersi
on1,&MinorVersion2,&MinorVersion3))
	{
	printf("ISP ReadFirmwareVersion success
%d.%d.%d.%d\r\n\n",MajorVersion, MinorVersion1,
MinorVersion2,MinorVersion3);
```

```
	}
	else
	{
		printf("ISP ReadFirmwareVersion
Failed\r\n\n");
	}
}
```

BOOL GetCameraUniqueID(TCHAR szUniqueID)

This function is used to get the unique ID of See3CAM_CU27 .

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 ResetDevice()

This function Sends the Extension Unit command to Reset the See3CAM_CU27 .

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

```
void ResettheDevice()
{
	if(!ResetDevice ())
	{
```

```
      printf("ResetDevice failed\r\n");
      }
}
```

BOOL GetAWBPresetModeCU27 (UINT8 *iAWBPresentMode)

This function Sends the Extension Unit command to get the Auto White Balance Mode in See3CAM_CU27 .

0x01 – Cloudy

0x02 – Daylight

0x03 – Flash

0x04 – Cool white fluorescent

0x05 – Tungsten

0x06 – Candlelight

0x07 – Horizon

0x08 – Custom

0x09 – Auto

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

Sample Code

```
Void  GetAWBPresentMode()
{
      UINT8 iAWBPresentMode =0;
      if(!GetAWBPresentModeCU27 (&iAWBPresentMode))
```

```
        {
            Printf("GetAWBPresentModeCU27  is failed\r\n");
        }
}
```

BOOL SetAWBPresetModeCU27 (UINT8 iAWBPresentMode)

This function Sends the Extension Unit command to set the Auto White Balanace in See3CAM_CU27 .

0x01 – Cloudy

0x02 – Daylight

0x03 – Flash

0x04 – Cool white fluorescent

0x05 – Tungsten

0x06 – Candlelight

0x07 – Horizon

0x08 – Custom

0x09 – Auto

Parameters	Description	Return Values
`UINT8 iAWBPresentMode`	AWB mode to be set.	TRUE on Success FALSE on Failure

Sample Code

```
void SetAWBPresentMode()
{
      UINT8 iAWBPresentMode = 0x01      //cloudy
      if(!SetAWBPresentModeCU27 (iAWBPresentMode))
      {
            printf("SetAWBPresentModeCU27 failed\r\n");
      }
}
```

BOOL GetAEMModeCU27 (UINT8 *iAEMMode)

This Function sends the Extension Unit Command to get the Auto Exposure Metering Mode of See3CAM_CU27 .

0x00 – Auto exposure off

0x01 – Center – weighted average mode

0x02 – All block integral mode

0x05 – Small area mode

0x06 – Large area mode

Parameters	Description	Return Values
UINT8 * iAEMMode	Pointer to store the AEM Mode.	TRUE on Success FALSE on Failure

Sample Code

```
void GetAEMMode()
{
      UINT8 iAEMMode = 0;
      if(!GetAEMModeCU27 (&iAEMMode))
      {
            printf("GetAEMModeCU27  Failed\r\n\n");
      }
}
```

BOOL SetAEMModeCU27 (UINT8 iAEMMode)

This function is used to set the Auto Exposure Metering Mode in See3CAM_CU27 .

0x00 – Auto exposure off

0x01 – Center – weighted average mode

0x02 – All block integral mode

0x05 – Small area mode

0x06 – Large area mode

Parameters	Description	Return Values
UINT8 iAEMMode	Auto Exposure Mode to be set.	TRUE on Success FALSE on Failure

Sample Code

```
void SetAEMMode()
{
      UINT8  iAEMMode = 0x00;       //AEM OFF
      if(!SetAEMModeCU27 (iAEMMode))
      {
            printf("SetAEMModeCU27  failed\r\n");
      }
}
```

BOOL GetFlickerModeCU27 (UINT8 *iFlickerMode)

This function sends the Extension Unit command to get the Flicker Mode in See3CAM_CU27 .

0x00 – Auto

0x01 – 50 HZ

0x02 – 60 HZ

0x04 – Disable

Parameters	Description	Return Values
UINT8 *iFlickerMode	Pointer to store the Flicker Mode.	TRUE on Success FALSE on Failure

Sample Code

```
void GetFlickerMode()
{
      UINT8 *iFlickerMode = 0;
      if(!GetFlickerModeCU27 (&iFlickerMode))
```

```
      {
            printf("GetFlickerModeCU27  failed\r\n");
      }
}
```

BOOL SetFlickerModeCU27 (UINT8 iFlickerMode)

This function sends the extension unit command to set the Flicker Mode in See3CAM_CU27 .

0x00 – Auto

0x01 – 50 HZ

0x02 – 60 HZ

0x04 – Disable

Parameters	Description	Return Values
UINT8 iFlickerMode	To store the Flicker Mode.	TRUE on Success FALSE on Failure

Sample Code

```
void GetFlickerMode ()
{
      UINT8 iFlickerMode = 0x01;   //50HZ
      if(GetFlickerModeCU27 (iFlickerMode))
      {
            printf("GetFlickerModeCU27  failed\r\n");
      }
}
```

BOOL GetJPEGQvalueCU27 (UINT8*iJPEGQValue)

This function sends the extension unit command to get the JPEG Q value of See3CAM_CU27 .

Parameters	Description	Return Values
UINT8 *iJPEGQValue	Pointer to store the JPEG Q value ranges from 0 to 100.	TRUE on Success FALSE on Failure

Sample Code

```
void GetJPEGQValue()
{
      UINT8 iJPEGQValue = 0;
      if(!GetJPEGQValueCU27  (&iJPEGQValue))
      {
            printf("GetJPEGQValueCU27  failed\r\n");
      }
}
```

BOOL SetJPEGQvalueCU27 (UINT8 iJPEGQValue)

This function sends the extension unit command to set the JPEG Value in See3CAM_CU27 .

Parameters	Description	Return Values
UINT8 iJPEGQValue	To store the JPEG Q value ranges from 0 to 100.	TRUE on Success FALSE on Failure

Sample Code

```
void SetJPEGQValue ()
{
      UINT8 iJPEGQValue = 70;
      if(!SetJPEGQValueCU27  (iJPEGQValue))
      {
            printf("SetJPEGQValueCU27  failed\r\n");
      }
}
```

BOOL RestoreDefaultCU27 ()

This function sends the Extension unit command to set our See3CAM_CU27 to restore to its default initial values.

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

Sample Code

```
void RestoreDefault()
{
      if (!RestoreDefaultCU27 ())
      {
            printf("RestoreDefaultCU27  failed\r\n");
      }
}
```

BOOL GetAELockStatusCU27 (UINT8 *iAELockMode)

This function sends the Extension unit command to get the Auto Exposure Lock status of See3CAM_CU27 .

0x00 – OFF

0x01 – ON

Parameters	Description	Return Values
UINT8 *iAELockMode	To store the AE lock status.	TRUE on Success FALSE on Failure

Sample Code

```
void GetAELockStatus()
{
      UINT8 *iAELockMode = 0;
      if(!GetAELockStatusCU27 (&iAELockMode))
      {
            printf("GetAELockStatusCU27  failed\r\n");
      }
}
```

BOOL SetAELockStatusCU27 (UINT8 iAELockMode)

This function sends the Extension unit command to Set the Lock for the Auto Exposure in See3CAM_CU27 .

0x00 – OFF

0x01 – ON

Parameters	Description	Return Values
UINT8 iAELockMode	To store the lock status of Auto Exposure.	TRUE on Success FALSE on Failure

Sample Code

```
void SetAELockStatus()
{
      UINT8 iAELockMode = 0x00;
      if(!SetAELockStatusCU27 (iAELockMode))
      {
            printf("SetAELockStatusCU27  failed\r\n");
      }
}
```

BOOL GetAWBLockStatusCU27 (UINT8 *iAWBLockState)

This function sends the Extension unit command to get the Lock status of the Auto White Balance of See3CAM_CU27 .

0x00 – Off

0x01 – On

Parameters	Description	Return Values
UINT8 *iAWBLockstate	To get the AWB lock status.	TRUE on Success FALSE on Failure

Sample Code

```
void GetAWBLockStatus()
{
      UINT8 *iAWBLockState=0;
      if(!GetAWBLockStatusCU27 (&iAWBLockState))
      {
            printf("GetAWBLockStatusCU27  failed\r\n");
      }
}
```

BOOL SetAWBLockStatusCU27 (UINT8 iAWBLockState)

This function sends the Extension unit command to set the Lock status for the Auto white Balance in See3CAM_CU27 .

0x00 – Off

0x01 – On

Parameters	Description	Return Values
`UINT8 iAWBLockState`	Lock status of AWB mode to be set.	TRUE on Success FALSE on Failure

Sample Code

```
void SetAWBLockStatus()
{
      UINT8 iAWBLockState=0x00;
      if(!SetAWBLockStatusCU27 (iAWBLockState))
      {
            printf("SetAWBLockStatusCU27  failed\r\n");
      }
}
```

BOOL GetBurstLengthCU27 (UINT8 *BurstLength)

This function sends the Extension unit command to get the Burst Length of See3CAM_CU27 .

Parameters	Description	Return Values
UINT8 *Burstlength	Pointer to store the Burst Length Value.	TRUE on Success FALSE on Failure

Sample Code

```
void GetBurstLength()
{
      UINT8 * Burstlength =0;
      if(!GetBurstLengthCU27 (&Burstlength))
      {
            printf(" GetBurstLengthCU27 failed\r\n");
      }
}
```

BOOL SetBurstLengthCU27 (UINT8 BurstLength)

This function sends the Extension unit command to set the Burst Length for See3CAM_CU27.

Parameters	Description	Range	Return Values
UINT8 BurstLength	The Burst length value can be set from 1 to 5 and the corresponding number of images will be captured when you trigger the still capture.	1 to 5	TRUE on Success FALSE on Failure

Sample Code

```
void SetBurstLength()
{
      UINT8 BurstLength =0x1B;
      if(!SetAutoFocusOFFCU27(BurstLength))
      {
            printf("SetAutoFocusOFFCU27 failed\r\n");
      }
}
```

BOOL GetDenoiseValueCU27(UINT8 *uDenoiseValue)

This function sends the Extension unit command to get the Denoise value of See3CAM_CU27 .

Parameters	Description	Return Values
`UINT8 * uDenoiseValue`	Pointer to store the denoise Value.	TRUE on Success FALSE on Failure

Sample Code

```
void GetBurstLength()
{
      UINT8 * uDenoiseValue =0;
      if(!GetDenoiseValueCU27 (&uDenoiseValue))
      {
            printf(" GetDenoiseValueCU27 failed\r\n");
      }
}
```

BOOL SetDenoiseValueCU27 (UINT8 uDenoiseValue)

This function sends the Extension unit command to set the Denoise value for See3CAM_CU27.

Parameters	Description	Return Values
UINT8 uDenoiseValue	Denoise value to be set 0x00 -Disable 0x01 -Enable	TRUE on Success FALSE on Failure

Sample Code

```
void SetDenoiseValue ()
{
      if(!SetDenoiseValueCU27 (uDenoiseValue))
      {
            printf("SetDenoiseValueCU27 failed\r\n");
      }
}
```

Support

Contact Us

If you need any support on See3CAM_CU27 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	17-AUG-2021	Initial Draft	Application Team
1.1	13-OCT-2021	Added Reset API	Application Team
1.2	06-DEC-2021	Added Denoise API	Application Team