导出相机 xml 配置文件并配置相机参数

Hello-FPGA
INFO@HELLO-FPGA.COM

目录

相机配置 xml 文件

1 说明

GenICam 定义了相机需要实施一个 xml 文件用于描述相机的功能、寄存器等详细信息，有了这些信息，host 才能利用这些信息完成对相机的配置。

对于 CoaxPress 接口的 Device 相机而言，xml 信息是如何存储的呢？
Xml 一般情形下都会存在 bootstrap 寄存器的用户区间内。

1.1 为什么要 XML

13 GenICam Support

13.1 Introduction

CoaXPress products shall support GenICam. See section 2.1 for details of the GenICam standard.

On the Device side this means an XML file shall be provided with the Device description compatible with the GenApi module of GenICam. The bootstrap registers in the Device provide a means to access the XML file. The features of the Device shall follow GenICam's SFNC if applicable, by using the name and type provided in the GenICam SFNC.

The Device vendor is free to use any suitable register layout in the Device, such as IIDC2 compliant or vendor specific, as long as an XML file is provided describing the layout and the features exposed, and providing the bootstrap registers defined in section 12.3.4 are supported.

To reduce the size of the XML file, a compressed version of the file can be stored in the Device. A compressed file shall use the DEFLATE and STORE methods of the ZIP format (section 2.2 ref 1).

Comment: This is mainly relevant to files stored in the Device, where non-volatile memory space may be limited. Note that the Device does not need to generate or understand the file, only store it.

On the Host side this means implementing a GenTL Producer to allow access to the Device's XML file, provide a control interface and a streaming interface.

1.2 XML 存在哪里

Table 54 — Bootstrap registers

Address	Name	Group	Support	Access	Length (bytes)	Register Interface
0x00000000	Standard	Support	M	R	4	Integer
0x00000004	Revision	Support	M, X	R	4	Integer
0x00000008	XmlManifestSize	Support	M	R	4	Integer
0x0000000C	XmlManifestSelector	Support	M	R/W	4	Integer
0x00000010	XmlVersion [XmlManifestSelector]	Support	M, X	R	4	Integer
0x00000014	XmlSchemaVersion [XmlManifestSelector]	Support	M	R	4	Integer
0x00000018	XmlUrlAddress [XmlManifestSelector]	Support	M	R	4	Integer

2 如何读出 XML 文件

利用标准采集卡及其对应的 SDK 应该都可以导出 xml 配置文件。

2.1 华睿相机

我们以 kaya 的采集卡为例进行说明。

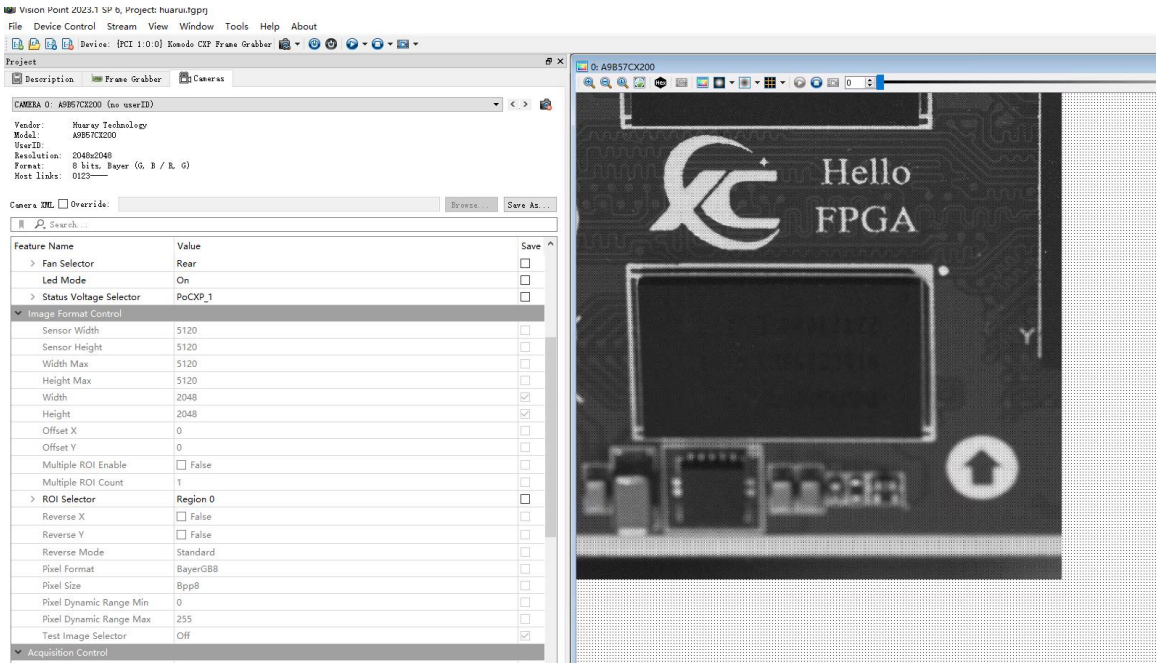

图 2-1 kaya 采集卡及其配套 SDK VISION POINT 获取相机信息

导出其 xml 文件即可。

general_bf2966e4.xml

该相机型号为华睿 A9B57CX200

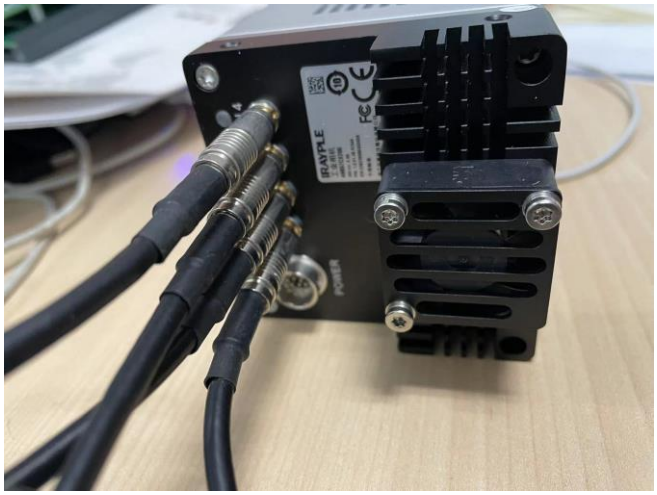

图 2-2 华睿 A9B57CX200 4 通道 6.25Gbps CoaXPress

2.2 海康相机

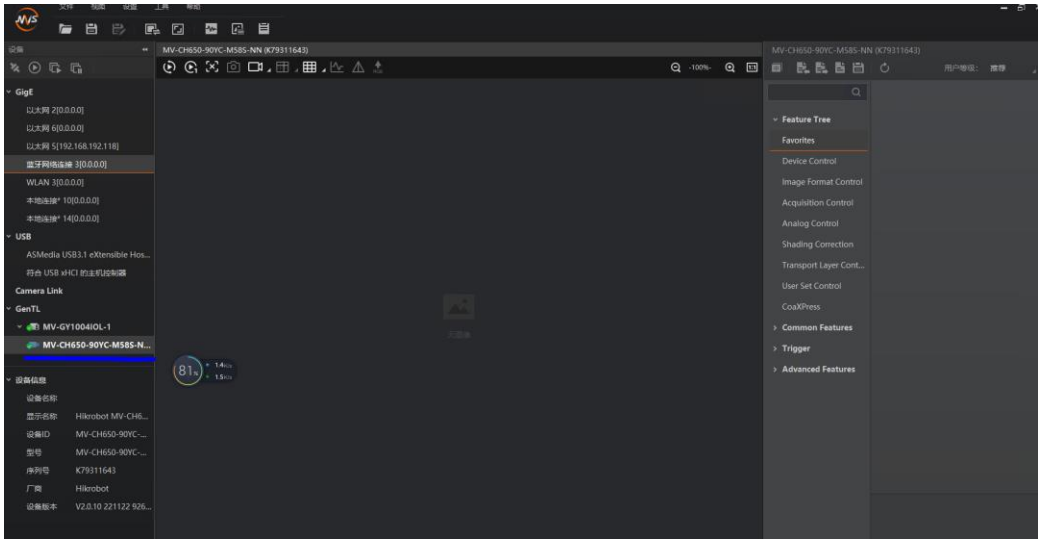

图 2-3 HK MVGY1004 采集卡 MVS 界面

选中相机右键即可保存相机的 xml 文件

MV-CH650-90YC-M58S-NN.xml

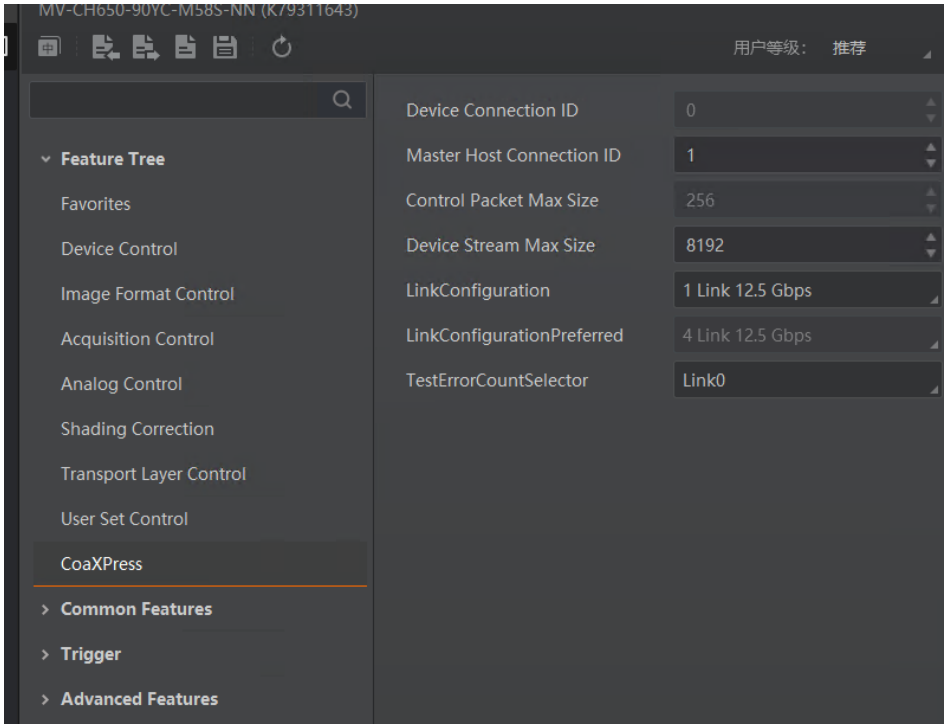

图 2-4 HK CXP 属性

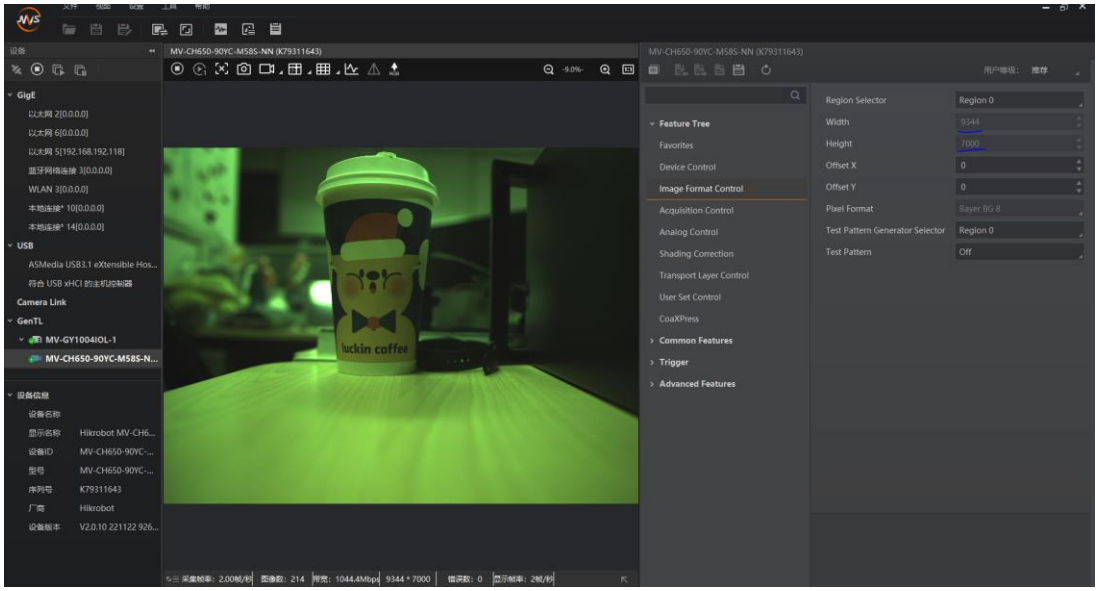

图 2-5 相机拍摄画面

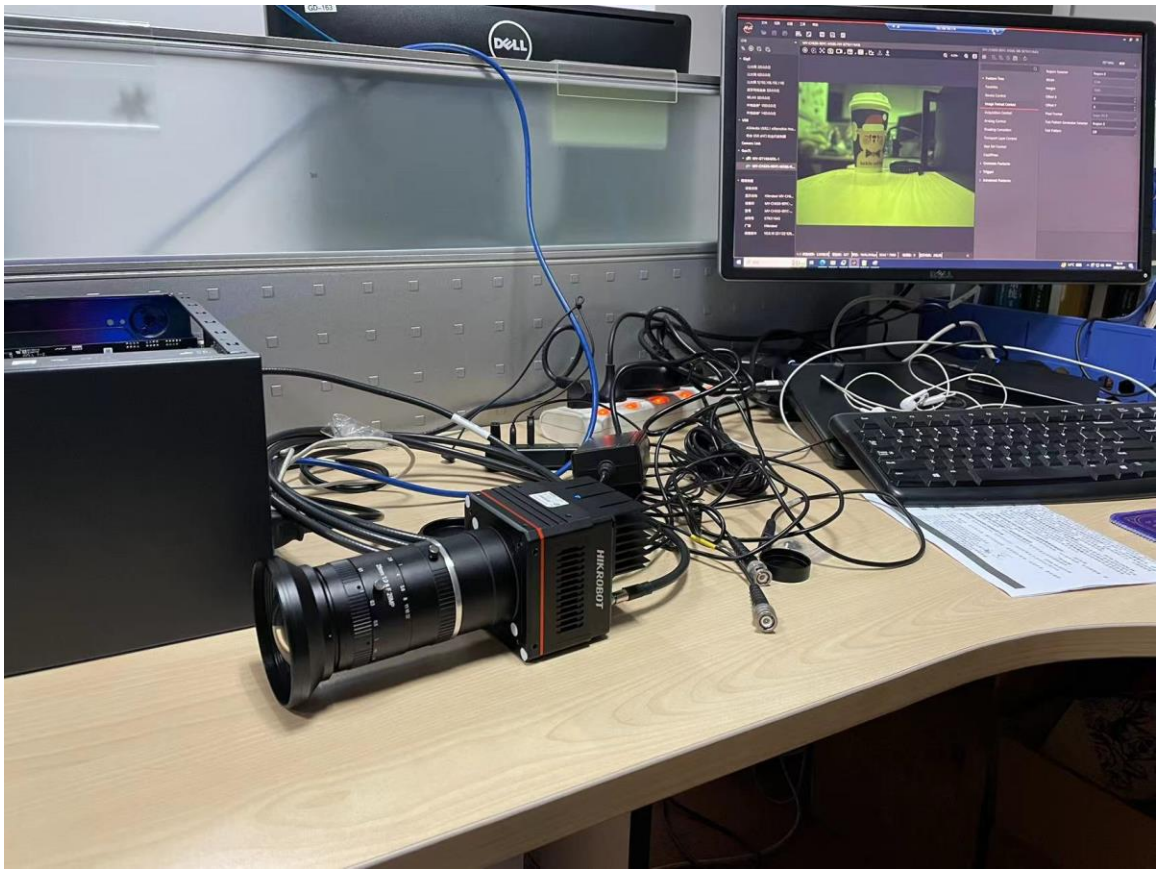

图 2-6 海康大面阵相机

2.3 ADMIC 相机

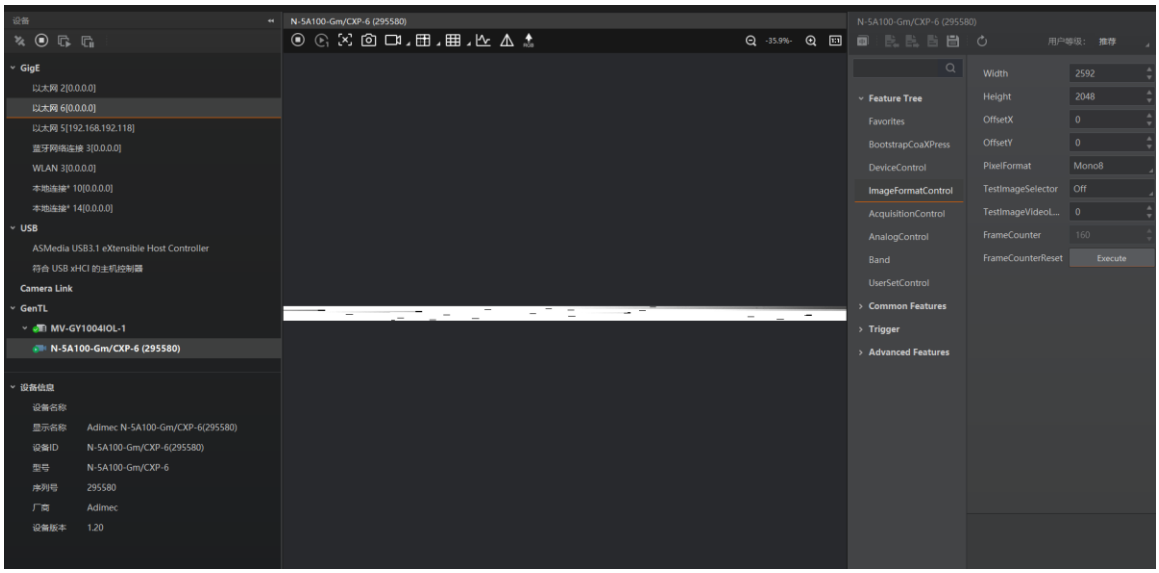

图 2-7 Admic 相机连接

Adimec_N-5A100
_Gm_CXP-6.xml

图 2-8 Admic 小相机

3 SDK DEMO 如何根据 XML 配置相机

以 huarui_A9B57CX200.xml 为示例进行说明，如何在 SDK demo 中完成相机寄存器的配置。

3.1 配置图像格式

首先查看相机格式寄存器地址。

```
1832      <IntReg Name="PixelFormatReg">
1833        <pAddress>ImageFormatControlBase</pAddress>
1834        <Address>0x0030</Address>
1835        <Length>4</Length>
1836        <AccessMode>RW</AccessMode>
1837        <pPort>Device</pPort>
1838        <Cachable>WriteThrough</Cachable>
1839        <pInvalidator>UserSetLoadReg</pInvalidator>
1840        <pInvalidator>ReverseXReg</pInvalidator>
1841        <pInvalidator>ReverseYReg</pInvalidator>
1842        <Sign>Unsigned</Sign>
1843        <Endianess>BigEndian</Endianess>
1844      </IntReg>
```

上图意思是说，PixelFormatReg 地址是在 ImageFormatControlBase 基地址基础上偏移 0x0030，对应实际地址为 ImageFormatControlBase + 0x0030, ImageFormatControlBase 为 0x30010000。

那么向寄存器写入什么值呢？如下图所示，如果要配置成 Mono8，那么 PixelFormatReg 就要写入 0x01080001.

```
1622    <Enumeration Name="PixelFormat" NameSpace="Standard">
1623      <ToolTip>Format of the pixel provided by the device.</ToolTip>
1624      <Description>Format of the pixel provided by the device. It represents all the informations provided by
          PixelCoding, PixelSize, PixelColorFilter but combined in one single value.</Description>
1625      <DisplayName>Pixel Format</DisplayName>
1626      <Visibility>Beginner</Visibility>
1627      <pIsLocked>TLParamsLocked</pIsLocked>
1628      <EnumEntry Name="Mono8">
1629        <pIsAvailable>Mono8AvailExpr</pIsAvailable>
1630        <Value>0x01080001</Value>
1631      </EnumEntry>
1632      <EnumEntry Name="Mono10">
1633        <pIsAvailable>Mono10AvailExpr</pIsAvailable>
1634        <Value>0x01100003</Value>
1635      </EnumEntry>
1636      <EnumEntry Name="Mono12">
1637        <pIsAvailable>Mono12AvailExpr</pIsAvailable>
1638        <Value>0x01100005</Value>
1639      </EnumEntry>
1640      <EnumEntry Name="BayerGR8">
1641        <pIsAvailable>BayerGR8AvailExpr</pIsAvailable>
1642        <Value>0x01080008</Value>
1643      </EnumEntry>
1644      <EnumEntry Name="BayerGR10">
1645        <pIsAvailable>BayerGR10AvailExpr</pIsAvailable>
1646        <Value>0x0110000C</Value>
1647      </EnumEntry>
1648      <EnumEntry Name="BayerGR12">
1649        <pIsAvailable>BayerGR12AvailExpr</pIsAvailable>
1650        <Value>0x01100010</Value>
1651      </EnumEntry>
```

那么怎么写入呢？
在 SDK Demo 中，如何完成这个配置呢？
调用 WRITE_DEVICE_REG (int address, int value) 即可，如果要读出数据，调用 int read_value = READ_DEVICE_REG(int address) 即可。

在什么位置完成相机的设置呢？
应该在设备链接建立之后配置，建议在 CameraInit()内增加。

3.2 配置曝光时间

根据附录示例，曝光时间参数地址 AcquisitionControlBase + 0x0058,//int Addr_ExposureTime;
那么对应增加 WRITE_DEVICE_REG (0x30020000+ 0X58, exposure_time)，通常相机单位是 us。
曝光模式也是类似的设置。

4 XML 寄存器类型说明

XML 寄存器类型来自 GenICam 标准，这里简单说明常见的类型

4.1 StringReg

String 类型，通常存储厂商信息

```
<StringReg Name="DeviceModelName" NameSpace="Standard">
<ToolTip>This feature provides the model of the device.</ToolTip>
<Description>This feature provides the model of the device.</Description>
<DisplayName>Device Model Name</DisplayName>
<Visibility>Beginner</Visibility>
<ImposedAccessMode>RO</ImposedAccessMode>
<Address>0x2020</Address>
<Length>32</Length>
<AccessMode>RO</AccessMode>
<pPort>Device</pPort>
<Cachable>NoCache</Cachable>
</StringReg>
```

4.2 IntReg

32 位 int 有符号，通常存储控制信息

```
<IntReg Name="HeightReg">
<Visibility>Invisible</Visibility>
<Address>0x6004</Address>
<Length>4</Length>
<AccessMode>RW</AccessMode>
<pPort>Device</pPort>
<Cachable>NoCache</Cachable>
<Sign>Unsigned</Sign>
<Endianess>BigEndian</Endianess>
</IntReg>
```

4.3 MaskedIntReg

MaskedIntReg – 从寄存器的一段取出一个整数，例如，从第 24 位到第 31 位，芬腾 S710 相机的部分寄存器就是这样定义的。

```
<MaskedIntReg Name="TriggerModeReg">
<Visibility>Invisible</Visibility>
<Address>0x6128</Address>
<Length>4</Length>
<AccessMode>RW</AccessMode>
<pPort>Device</pPort>
<Cachable>NoCache</Cachable>
<LSB>31</LSB>
<MSB>24</MSB>
<Sign>Unsigned</Sign>
<Endianess>BigEndian</Endianess>
</MaskedIntReg>
```

4.4 FloatReg

单精度浮点数，32bit 表示

```
<FloatReg Name="pFrameRateReg">
<Visibility>Invisible</Visibility>
<Address>0x60C0</Address>
<Length>4</Length>
<AccessMode>RW</AccessMode>
<pPort>Device</pPort>
<Cachable>NoCache</Cachable>
<Endianess>BigEndian</Endianess>
</FloatReg>
```

比如该寄存器读出 16 进制数据为 0x41c00000，对应小数为 24

https://www.h-schmidt.net/FloatConverter/IEEE754.html

5 附录

以 huarui_A9B57CX200.xml 为示例

```
//huarui A9B57CX200 5000*5000
const CameraRegisters CameraRegistersVendorHuarui = {
#define AcquisitionControlBase              0x30020000
    AcquisitionControlBase + 0x0000,//int Addr_AcquisitionMode;
    AcquisitionControlBase + 0x0004,    //int Addr_AcquisitionStart;// 1
    AcquisitionControlBase + 0x0008,    //int Addr_AcquisitionStop  ;// 1
    AcquisitionControlBase + 0x000c,    //int Addr_AcquisitionBurstFrameCount;// 1
    AcquisitionControlBase + 0x0010,    //int Addr_AcquisitionFrameRate;
```

```
    AcquisitionControlBase + 0x0014,//int AcquisitionStatusAcq
        //int Addr_AcquisitionFrameRateEnable;
        //int Addr_AcquisitionFrameRateMax;
        ////#define ResetAll                    0x00008300//resets the camera to its power
up state
    AcquisitionControlBase + 0x0020,//int Addr_TriggerMode    ; //sel==1 ¹ì¶¨ÉèÖÃ
    AcquisitionControlBase + 0x0030,    //int Addr_TriggerSource;
    AcquisitionControlBase + 0x0038,    //int Addr_TriggerActivation;
    AcquisitionControlBase + 0x0028,//int Addr_TriggerSoftware;
    AcquisitionControlBase + 0x003c,    //Addr_TriggerDelay

    AcquisitionControlBase + 0x0048,    //int Addr_ExposureMode;
    AcquisitionControlBase + 0x0050,    //int Addr_ExposureTimeMinReg;
    AcquisitionControlBase + 0x0054,    //int Addr_ExposureTimeMaxReg;
    AcquisitionControlBase + 0x0058,    //int Addr_ExposureTime;

#define AnalogControlBase               0x30030000
        AnalogControlBase + 0x001C,//int Addr_BlackLevel;//Controls the analog black level
0-255
        AnalogControlBase + 0x000C,//int Addr_AnalogGain;//Defines the analog Gain in
discrete steps.enum x1 x2 x4
        //
        //
#define ImageFormatControlBase          0x30010000
    ImageFormatControlBase + 0x0010,//int Addr_Width          ;
    ImageFormatControlBase + 0x001c,//int Addr_Height         ;
    ImageFormatControlBase + 0x0028,//int Addr_OffsetX        ;
    ImageFormatControlBase + 0x002C,//int Addr_OffsetY        ;
    ImageFormatControlBase + 0x0030,//int Addr_PixelFormat    ;
    ImageFormatControlBase + 0x0004,//int Addr_SensorHeight ;
    ImageFormatControlBase + 0x0674,//int Addr_SensorWidth  ;
    ImageFormatControlBase + 0x003c,//int Addr_TEST_IMAGE_SLECTOR;//TestImageSelector
        //
#define TransportLayerControlBase       0x30040000
    TransportLayerControlBase + 0x4,//int Addr_TapGeometry    ; //0x0004
    TransportLayerControlBase + 0x301C,//int Addr_Image1StreamID;
};
```