DTCONTROL VS X-LIB

Revision: 1.0

1. OVERVIEW

This document includes comparison of functions and commands between DTControl library and X-Lib library. X-Lib library is used with DT's new products, which are based on the digital platform. It includes several new features, which were not supported by previous products. In addition, some features are implemented differently. Thus there isn't exact match for some of the functions or commands on DTControl library. However, the same functionality can be still implemented also with X-Lib.

Please refer to following documents for details:

"X-View5.3 User's Manual" includes DTControl.dll and related commands

"X-LIB Software Library Programmer's Manual" includes X-Lib.dll and related commands

2. CLASSES AND FUNCTIONS, DTCONTROL VS X-LIB

DTControl class	**X-LIB corresponding class**
CDTDetector	XCommand, XDevice, XSystem
CDTImage	XAcquisition, XFrameTransfer, XOffCorrect
CDTCommander	XCommand
CDTDisplay	XDisplay
CImageObject	XImage, XAnalysis

CDTDetector	**X-LIB corresponding function**	**Note**
ULONG GetBaudRate()	N/A	
Void PutBaudRate (ULONG newVal)	N/A	
ULONG GetChannelType()	N/A	
Void PutChannelType (ULONG newVal)	N/A	
ULONG GetCmdPort()	XDevice::GetCmdPort()	
Void PutCmdPort (ULONG newVal)	XDevice::SetCmdPort()	
ULONG GetCmdTimeOut()	N/A	

Void PutCmdTimeOut (ULONG newVal)	XCommand::SetTimeout()	
BOOL GetEnableLog()	N/A	
Void PutEnableLog (BOOL newVal, const char* pFileName)	XSystem::EnableLog()	
Char* GetErrMsg (LONG ErrID)	N/A	
const char* GetIPAddress()	XDevice::GetIP()	
Void PutIPAddress (const char* newVal)	XDevice::SetIP()	
BOOL GetIsOpened()	XCommand::GetIsOpen()	
ULONG GetLastErrID()	XCommand::GetLastError()	
ULONG GetSerialPort()	N/A	
Void PutSerialPort (LONG newVal)	N/A	
ULONG GetUsbPort()	N/A	
Void PutUsbPort (ULONG newVal)	N/A	
Void Close()	XCommand::Close()	
BOOL Open()	XComamnd::Open()	
BOOL SendCommandA (CHAR* Cmd,CHAR* pRT)	XCommand::SendAscCmd()	
Void SetCallback (void (*OnDTError)(LONG,CHAR*))	XCommand::RegisterEventSink()	
BOOL StartPing (ULONG nInterval)	XCommand::StartHeartbeat()	
Void StopPing()	XCommand::StopHeartbeat()	

CDTImage	**X-LIB corresponding function**	**Note**
ULONG GetBytesPerPixel()	XDevice::GetPixelDepth()	
Void PutBytesPerPixel (ULONG newVal)	N/A	
ULONG GetChannelType()	N/A	
Void PutChannelType (ULONG newVal)	N/A	
CDTDetector* GetDetectorObject()	N/A	
Void PutDetectorObject (CDTDetector* pDetector)	XAcquisition::Open()	
ULONG GetDualEnergyMode()	N/A	
Void PutDualEnergyMode (ULONG newVal)	N/A	
float GetDualRegShift()	N/A	
Void SetDualRegShift (float shift)	N/A	
ULONG GetExFrameTrigger()	N/A	
Void PutExFrameTrigger (ULONG newVal)	N/A	
ULONG GetImgHeight()	XFrameTransfer::GetLineNum()	
Void PutImgHeight (ULONG newVal)	XFrameTransfer::SetLineNum()	
ULONG GetImgWidth()	XDevie::GetPixelNumber()	
Void PutImgWidth (ULONG newVal)	N/A	
CImageObject* GetImageObject()	N/A	
ULONG GetImagePort()	XDevice::GetImgPort()	
Void PutImagePort (ULONG newVal)	XDevice::SetImgPort()	
BOOL GetIsGrabbing()	XAcquisition::GetIsGrabbing()	

BOOL GetIsOpened()	XAcquisition::GetIsOpen()	
ULONG GetLastErrID()	XAcquisition::GetLastError()	
ULONG GetLostStartID()	N/A	
CDTImage* GetObjectHandle()	N/A	
ULONG GetStreamPixelPerBytes()	N/A	
Void PutStreamPixelPerBytes (ULONG newVal)	N/A	
ULONG GetSubFrameHeight()	N/A	
Void PutSubFrameHeight (ULONG newVal)	N/A	
ULONG GetTimeOut()	N/A	
Void PutTimeOut (ULONG newVal)	XAcquisition::SetTimeout()	
Void Close()	XAcquisition::Close()	
BOOL Grab (ULONG FrameNum)	XAcquisition::Grab()	
BOOL Open()	XAcquisition::Open()	
BOOL Save (const CHAR* FilePath)	XImage::Save()	
void SetCallBack (void(*OnEvent)(BYTE, BYTE), void(*OnError)(LONG, CHAR*), void(*OnFrameReady)(), void(*OnSubFrameReady)(LONG, LONG, LONG, BOOL))	XAcquisition::RegisterEventSink()	
BOOL SetDirectFlag (ULONG Flag)	DualLineCorrect::SetDirect()	
BOOL Snap()	XAcquisition::Snap()	
BOOL StartGrabEngine()	N/A	
Void Stop()	XAcquisition::Stop()	
Void TriggerOneGrab()	N/A	

CDTDisplay	**X-LIB corresponding functions**	**Note**
Float GetGammaVal()	XDisplay::GetGama()	
Void PutGammaVal (float newVal)	XDisplay::SetGama()	
ULONG GetImageHeight()	N/A	
ULONG GetImageWidth()	N/A	
Void PutImgDataAddress (BYTE* pData)	XDisplay::Display()	
BOOL GetIsDDrawEnable()	XDisplay::GetIsDDrawEnable()	
BOOL GetIsOpened()	XDisplay::GetIsOpen()	
ULONG GetLastErrorID()	XDisplay::GetLastError()	
ULONG GetPixelBits()	N/A	
Void Close()	XDisplay::Close()	
Void Display()	XDisplay::Display()	
BOOL Open (ULONG imageWidth, ULONG imageHeight, ULONG pixelDepth, HWND hWnd, int colorMode)	XDisplay::Open()	

CImageObject	**X-LIB corresponding function**	**Note**
ULONG GetBytesPerPixel()	XImage:: _pixel_depth	
ULONG GetHeight()	XImage::_height	
Const ULONG_PTR GetImageDataAddress()	XImage:: _data_	
Const ULONG_PTR GetImageLineAddress (LONG LineIndex)	XImage::GetLineAddr()	

ULONG GetPixel (ULONG X, ULONG Y)	XImage::GetPixelVal()	
Void PutPixel (ULONG X, ULONG Y, ULONG newVal)	XImage::SetPixelVal()	
ULONG GetWidth()	XImage::_width	
Void DoStatistical (ULONG CalType)	XAnalyze::DoAnalyze()	
Void ColAverage (ULONG** pAvg)	XAnalyze::_col_avg_	
Void ColMax (ULONG** pMax)	XAnalyze::_col_max_	
Void ColMin (ULONG** pMin)	XAnalyze::_col_min_	
Void ColNoise (float** pNoise)	XAnalyze::_col_noise_	
Void RowAverage (ULONG** pAvg)	XAnalyze::_row_avg_	
Void RowMax (ULONG** pMax)	XAnalyze::_row_max_	
Void RowMin (ULONG** pMin)	XAnalyze::_row_min_	
Void RowNoise (float** pNoise)	XAnalyze::_row_noise_	

CDTCommander	**X-LIB corresponding function**	**Note**
BOOL GetAverageFilter (ULONG* pVal)	XCommand::GetPara(XPARA_AVERAGE_MODE	

BOOL PutAverageFilter (ULONG newVal)	XCommand::SetPara(XPARA_AVERAGE_MODE	
BOOL GetBaseline (ULONG* pVal)	XCommand::GetPara(XPARA_BASE_LINE	
BOOL PutBaseline (ULONG newVal)	XCommand::SetPara (XPARA_BASE_LINE	
BOOL GetCorrectionBaseline (BOOL *pVal)	XCommand::GetPara(XPARA_EN_BASELINE_CORRECT	
BOOL PutCorrectionBaseline (BOOL newVal)	XCommand::SetPara(XPARA_EN_BASELINE_CORRECT	
BOOL GetCorrectionGain (BOOL* pVal)	XCommand::GetPara(XPARA_EN_GAIN_CORRECT	
BOOL PutCorrectionGain (BOOL newVal)	XCommand::SetPara(XPARA_EN_GAIN_CORRECT	
BOOL GetCorrectionOffset (BOOL* pVal)	XCommand::GetPara(XPARA_EN_OFFSET_CORRECT	
BOOL PutCorrectionOffset (BOOL newVal)	XCommand::SetPara(XPARA_EN_OFFSET_CORRECT	
BOOL GetDataPattern (BOOL *pVal)	XCommand::GetPara(XPARA_GCU_TEST_MODE	
BOOL PutDataPattern (BOOL newVal)	XCommand::SetPara(XPARA_GCU_TEST_MODE	

BOOL GetDataPatternMode (ULONG *pVal)	XCommand::GetPara(XPARA_GCU_TEST_MODE	
BOOL PutDataPatternMode (ULONG newVal)	XCommand::SetPara(XPARA_GCU_TEST_MODE	
CDTDetector* GetDetectorObject()	N/A	
Void PutDetectorObject (CDTDetector* pDetector)	N/A	
BOOL GetEndPixel (ULONG* pVal)	Not support	XOffCorrect class provides the corresponding function of off-board correction
BOOL PutEndPixel (ULONG newVal)	Not support	XOffCorrect class provides the corresponding function of off-board correction
BOOL GetExFrameTriggerDelay (ULONG* pVal)	XCommand::GetPara(XPARA_FRAME_TRIGGER_DELAY	
BOOL PutExFrameTriggerDelay (ULONG newVal)	XCommand::SetPara(XPARA_FRAME_TRIGGER_DELAY	
BOOL GetFECardsNum (ULONG* pVal)	XCommand::GetPara(XPARA_CH_NUM	
BOOL GetFPGABuild (ULONG* pVal)	N/A	
BOOL GetFPGAVersion (ULONG* pVal)	XCommand::GetPara(XPARA_GCU_FIRM_VER	
BOOL GetGain	N/A	

(ULONG Index, FLOAT* pVal)		
BOOL PutGain (ULONG Index, FLOAT newVal)	N/A	
BOOL GetIntegrationTime (ULONG* pVal)	XCommand::GetPara(XPARA_INT_TIME	
BOOL PutIntegrationTime (ULONG newVal)	XCommand::SetPara(XPARA_INT_TIME	
BOOL GetIsOpened()	XCommand::GetIsOpen()	
BOOL GetLineTrigger (BOOL* pVal)	XCommand::GetPara(XPARA_EN_LINE_TRIGGER	
BOOL PutLineTrigger (BOOL newVal)	XCommand::SetPara(XPARA_EN_LINE_TRIGGER	
BOOL GetMaxIntTime (ULONG* pVal)	XCommand::GetPara(XPARA_MAXMIN_INTTIME	
BOOL GetMCUVersion (ULONG* pVal)	N/A	
BOOL GetMinIntTime (ULONG pVal)	XCommand::GetPara(XPARA_MAXMIN_INTTIME	
BOOL GetModuleType (ULONG* pVal)	XCommand::GetPara(XPARA_GCU_TYPE	
BOOL GetModuleTypeName (char** pVal)	N/A	
BOOL GetOffset	N/A	

(ULONG Index, ULONG* pVal)		
BOOL PutOffset (ULONG Index, ULONG newVal)	N/A	
BOOL GetOutputBits (ULONG* pVal)	XCommand::GetPara(XPARA_OUTPUT_SCALE	
BOOL PutOutputBits (ULONG newVal)	XCommand::SetPara(XPARA_OUTPUT_SCALE	
BOOL GetOverallGain (FLOAT* pVal)	N/A	
BOOL PutOverallGain (FLOAT* newVal)	N/A	
BOOL GetPixelNumber (ULONG* pVal)	XCommand::GetPara(XPARA_PIXEL_NUMBER	
BOOL GetPixelSize (FLOAT* pVal)	XCommand::GetPara(XPARA_PIXEL_SIZE	
BOOL GetSensitivityLevel (ULONG* pVal)	XCommand::GetPara(XPARA_DM_GAIN	
BOOL PutSensitivityLevel (ULONG newVal)	XCommand::SetPara(XPARA_DM_GAIN	
BOOL GetSerialNumber (char** pVal)	XCommand::GetPara(XPARA_GCU_SERIAL	
BOOL GetStartPixel	not support	XOffCorrect class provides the corresponding function of off-board

(ULONG* pVal)		correction
BOOL PutStartPixel (ULONG newVal)	not support	XOffCorrect class provides the corresponding function of off-board correction
BOOL GetSumLines (ULONG* pVal)	XCommand::GetPara(XPARA_SUMMING_MODE	
BOOL PutSumLines (ULONG newVal)	XCommand::SetPara(XPARA_SUMMING_MODE	
BOOL GetTemperature (ULONG* pVal)	XCommand::GetPara (XPARA_GCU_HEALTH	
BOOL EnOffGainOffset (BOOL bGain, BOOL bOffset)	XOffCorrect::DoCorrect()	
BOOL Initialize (BOOL Mode)	XCommand::ExecutePara(XPARA_INIT_PARA XCommand::ExecutePara(XPARA_INIT1_PARA	
BOOL LoadGain (ULONG Index)	XOffCorrect::LoadFlash()	
BOOL LoadOffGainOffset (char* pFile, CDTImage* pDTImage)	XOffCorrect::LoadFile()	
BOOL LoadOffset()	XOffCorrect::LoadFlash()	
BOOL OffBoardCalibration (ULONG CalibrationType, CDTImage* pDTImage, ULONG StartPixel, ULONG EndPixel,	XOffCorrect::CalculatePara()	

ULONG TargetVal)		
BOOL OnBoardGainCalibration (ULONG TargetValue)	not support	XOffCorrect class provides the corresponding function of off-board correction
BOOL OnBoardOffsetCalibration ()	not support	XOffCorrect class provides the corresponding function of off-board correction
BOOL ResetGain()	XCommand::ExecutePara(XPARA_RESET_GAIN	
BOOL ResetOffset()	XCommand::ExecutePara(XPARA_RESET_OFFSET	
BOOL SaveCurrentStatus()	XCommand::ExecutePara(XPARA_SAVE_PARA	
BOOL SaveGain (ULONG Index)	XOffCorrect::SaveFlash()	
BOOL SaveOffGainOffset (char* pFile, CDTImage* pDTImage)	XOffCorrect::SaveFile()	
BOOL SaveOffset()	XOffCorrect::SaveFlash()	
BOOL SendExFrameTrigger()	XCommand::ExecutePara(XPARA_SIMULATE_FRAME_TRIGGER	

DTControl	**New functions of X-LIB**	**Note**
N/A	XPiecewiseCorrect class	The XPiecewiseCorrect object takes three air images corresponding to low, middle and high X-ray to build a more accurate piecewise correcting function.
N/A	XReferenceCorrect class	The XReferenceCorrect object provides function to correct the

		signal variations caused by LINAC X-ray source.
N/A	XPixelCorrect class	The XPixelCorrect object corrects the bad pixels.
N/A	XOffHLCorrect class	The XOffCorrect class is in charge of the off-board correction functions only in non-continuous hi/lo trigger mode.
N/A	XMultiTransfer class	The XMultiTransfer class synchronizes multi-FrameTransfer objects while connecting with more than one X-GCUs. It combines the sub-frames into one frame. The XAcquisition object should enable the line info mode while working with the XMultiTransfer object.
N/A	XSystem class	The XSystem object communicates with the X-GCU by broadcast. It enumerates the IP, MAC and port info of the X-GCU and also configures the network settings.
N/A	Camlink wrapper functions	Wraps Camlink frame grabber card API.

3. COMMAND COMPARISON

Old Control Unit		**X-GCU**	
Command(ASCII)	**Description**	**Command(HEX/ASCII)**	**Note**
AC	Calibration Area No Save and Load	N/A	No on-board correction
AF	Average filter size property	0x41/AF	
AS	Set AD Sampling Point	0x74/AP	
BS	Baseline property	0x35/BS	
CN	Set DM number in each channel	0x25/CN	
CS	Calibration switch property	0x30/EG 0x31/EO 0x32/EB	
DA	Set DM Averaging Function (Pixel binning function)	0x40/DA	
DC	Set high low operation mode in single energy operation mode	0x24/DC	
DM	Set Detector and ASIC type	0x6E/CT	
DS	Set Trigger Delay Sweep Mode	N/A	
DT	Set DM board Diagnostic Test Mode	0x6B/DT	
ED	Enable data pattern. Set pattern mode	0x6A/ED	
EF	Enable/Disable external frame trigger	0x54/FM 0x55/EF 0x56/FD	
EL	External line trigger Enable/Disable property	0x51/EL	
EM	Single/dual energy mode property	0x6C/DP	
FI	Get the FPGA version info	0x68/GF 0x69/DF	
GC	Gain Calibration	N/A	No on-board correction
GS	Save and load Gain	0x33/GC,L	
IN	Initialize	0x10/IN 0x11/IN1 0x12/IN2	

IT	Set IF board Test Mode	0x6B/DT	
LC	Enable/Disable LED	0x75/LC	
NA	Set image channel listen port	0x01	Broadcast
NG	gateway property	N/A	
NI	IP address property	0x01	Broadcast
NK	Mask property	N/A	
NM	Mac address	0x01	Broadcast
NP	Command listen port	0x01	Broadcast
NW	Save/Load network parameters	0x01	Broadcast
NT	Non-Continues Mode Integration time setting	0x21/NT	
OA	Offset Moving Average Filter	0x44/OA	
OC	Offset calibration	N/A	No on-board correction
OM	Operation Mode: Continues/ Non-Cont Mode, Dual/Single Energy Mode.	0x22/OM	
OS	Offset save/load	0x34/OC,L	
PA	operation of average filter	0x41/AF	
PN	Test Mode Pulse Number	N/A	
RC	Reset gain and offset	0x37/RG 0x38/RO	
RG	Set overall gain	N/A	
RI	Get the detector info	0x64/PN 0x65/PS 0x67/IR 0x61/GT 0x62/GS	
SA	Access FPGA SRAM	N/A	
SB	Output scaling	0x43/SB	
SC	Set gain or offset value for a pixel	N/A	
SD	Set integration mode; you can refer to function description in remark;	0x22/OM	
SE	Send one external trigger pulse	0x57/SE	

SF	Start/Stop frame data out	0x27/SF	
SG	Operation of DM Gain Setting	0x23/SG	
SI	Set image output port	N/A	
SK	Set constant integration time you can refer to function description in remark;	0x22/OM 0x21/NT	
SM	Sending times of SPI Command in XCU	N/A	
SP	Set 16/20bit system	N/A	
SR	Save current status	0x10/IN,S	
SS	Sensitivity level	0x23/SG	
ST	Integration Time. Unit is μs	0x20/ST	
SU	Set sum line function	0x42/SU	
TD	Trigger delay	0x52/TD	
TM	Test Mode	0x6B/DT	
TP	Ping period	0x60/TP	
TR	Retrieve the MCU board temperature	0x72/GI	
TS	Select trigger source in IF2 board	N/A	
TT	Set Trigger Raw Delay	0x53/RD	
WR	Write/Read register	N/A	
N/A		0x50/LM	Set external line trigger mode
N/A		0x58/LE	Set Hi/Lo line trigger with hi/lo edge level detection
N/A		0x59/SP	Set Hi/Lo line trigger sampling position
N/A		0x63/DS	Set DM serial number
N/A		0x66/PD	Set Pixel depth
N/A		0x6D/DN	Set card number per DFE
N/A		0x73/DI	Read DM’s Voltage and temperature
N/A		0x76/TR	Enable terminal resistor

4. CALIBRATION PROCEDURES COMPARISON

DTControl	**X-LIB corresponding functions**	**Note**
Commander.ResetOffset(); *Commander.ResetGain();*	XOffCorrect::Reset(); XCommand::ExecutePara (XPARA_RESET_GAIN); XCommand::ExecutePara (XPARA_RESET_OFFSET);	
Commander.put_StartPixel(0); *Commander.put_EndPixel(863)*	N/A	Start, end position is wrapped in XOffCorrect::Calculate()
//Shut down X-ray *//Do offset calibration.[OC]* *Commander.OnBoardOffsetCalibration();* *//Wait 3s or more* *Sleep(3000);*	//Power off X-ray, calculate offset XOffCorrect::Calculate(1,…)	
Commander.put_CorrectionOffset(1); *//Power on X-ray* *//Do gain calibration, and use mean level as target signal value. [GC,1]* *Commander.OnBoardGainCalibration(1);* *Sleep(3000);*	//Power on X-ray, calculate gain XOffCorrect::Calculate(0,….)	
Commander.put_BaseLine(80);	XCommand::SetPara (XPARA_BASE_LINE , 80)	
//Save gain and offset into flash *Commander.SaveGain(0);* *Commander.SaveOffset();*	XOffCorrect::SaveFlash(cmd,0)	
//Enable gain, offset and base line correction. [CS,W,G,1] [CS,W,O,1] [CS,W,B,1] *Commander.put_CorrectionGain(1);* *Commander.put_CorrectionOffset(1);* *Commander.put_CorrectionBaselline (1);*	XCommand ::SetPara (XPARA_EN_GAIN_CORRECT ,1); XCommand ::SetPara (XPARA_EN_Offset_CORRECT ,1); XCommand ::SetPara (XPARA_EN_BASELINE_CORRECT ,1);	
//After the process finished, just load the parameters while restarting the system. *Commander.Loadgain(0);* *Commander.LoadOffset();* *Commander.put_CorrectionGain(1);* *Commander.put_CorrectionOffset(1);* *Commander.put_BaseLine(80);* *Commander.put_CorrectionBaselline*	//After the process finished, just load the parameters while restarting the system. XCommand::ExecutePara (XPARA_LOAD_GAIN, 0); XCommand::ExecutePara	

(1);	(XPARA_LOAD_OFFSET); XCommand::SetPara (XPARA_BASE_LINE , 80) XCommand ::SetPara (XPARA_EN_GAIN_CORRECT ,1); XCommand ::SetPara (XPARA_EN_Offset_CORRECT ,1); XCommand ::SetPara (XPARA_EN_BASELINE_CORRECT ,1);	

5. WORKING PROCESS COMPARISON

DTControl	**X-LIB corresponding functions**	**Note**
CDTDetector Detector; *//Set the command channel properties* *Detector.PutChannelType(2); //TCP* *Detector.PutIPAddress("192.168.1.2"); //IP* *Detector.PutCmdPort(3000); //Port*	XSystem xsystem("192.168.1.10"); //Bind local IP address xsystem.Open(); int dev_num = xsystem.FindDevice(); //Find detector XDevice* dev_= xsystem.GetDevice(0); //Get IP, port etc	Get properties of detector
Detector.SetCallback(&OnDTError); *Detector.Open()*	XGigFactory xfactory; XCommand xcommand(&xfactory); xcommand.RegisterEventSink(&cmd_sink); xcommand.Open(dev_);	Open command channel
CDTImage Image; *Image.PutChannelType (6) // UDP* *Image.PutDetectorObject (&Detector) //Image control need call detector to send* *Image.PutImgHeight (512);* *Image.PutImgWidth (1024); //Total pixel number* *Image.PutImagePort (4001); //Port* *Image.PutBytesPerPixel(2); //For 16-bit or 14-bit series* *Image.PutStreamPixelPerBytes (2); //For 16-bit or 14-bit series* *Image.SetCallBack(&OnDTEvent, &OnDTError, &OnFrameReady);*	XFrameTransfer xtransfer(512); //Frame has 512 lines xtransfer.RegisterEventSink(&img_sink); XAcquisition xacq(&xfactory); xacq.RegisterEventSink(&img_sink); xacq.RegisterFrameTransfer(&xtransfer);	Set properties of image channel
Image.Open ()	xacq.Open(dev_, &xcommand);	Open image channel
//Send command, the pRTBuffer will get the returned info *Detector.SendCommandA("[ST,R]",pRTBuffer);*	xcommand.SendAscCmd("[ST,R,0]", recv_str);	Send command to detector
//ImageObject to get image data *pImageObject = Image.GetImageObject();*	xacq.Grab(0);	Start scanning

//Call grab function, the on frame ready callback function will respond *Image.Grab (0)*		
Image.Stop();	Xacq.Stop()	Stop scanning
Image.Close(); *Detector.Close();*	xacq.Close(); xcommand.Close(); xsystem.Close();	Close objects
//Define error callback function *void OnDTError(LONG ID, CHAR* Info)* *{* *printf("Error happens, error ID %d, error info %s \n", ID, Info);* *}* *//Define event callback function* *void OnDTEvent(BYTE ID, BYTE Num)* *{* *switch(ID)* *{* *case 1:* *printf("Data lost happens, %d lines lost\n ", Num);* *break;* *case 2:* *printf("Frame buffer overflow \n");* *break;* *default:* *break;* *}* *}* *//Define frame ready callback function* *void OnFrameReady()* *{* *//Get the pointer of image data* *BYTE* pImgData = (BYTE*)pImageObject->GetImageDataAddress();* *//Get pixel value* *ULONG PixelValue = pImageObject->GetPixel(X,Y);* *}*	class CmdSink :public IXCmdSink { public: void OnXError(uint32_t err_id, const char* err_msg_) { printf("OnXERROR: %u, %s\n", err_id, err_msg_); } void OnXEvent(uint32_t event_id, float data) { // switch(event_id) // { // case XEVENT_CMD_HEARTBEAT_TEMPRA: // printf("X-GCU's temperature: %f\n", data); // break; // case XEVENT_CMD_HEARTBEAT_HUMIDITY: // printf("X-GCU's humidity: %f\n",data); // break; // } } }; class ImgSink : public IXImgSink { void OnXError(uint32_t err_id, const char* err_msg_) { printf("OnXERROR: %u, %s\n", err_id, err_msg_); } void OnXEvent(uint32_t event_id, uint32_t data) { if(XEVENT_IMG_PARSE_DATA_LOST == event_id || XEVENT_IMG_PARSE_PAC_LOST == event_id) { lost_count += data; } } void OnFrameReady(XImage* image_) { printf("Frame %u ready, lost line %u\n", frame_count++, lost_count); printf("Firset pixel value: %u\n", image_->GetPixelVal(0,0));	Call back function

	} }; CmdSink cmd_sink; ImgSink img_sink;	

Change record

Rev	Date issued	Person	Description of change
1.0	2016-11-30	Zhang Xu	Initial release