## 多相机使用方式：

在应用软件中包含头文件：

Include “cctapi.h”

在应用软件头文件中开指针，有几个相机就开几个这种类

CCCTAPIAppUSB \*h\_cctapi;

CCCTAPIAppUSB \*h\_cctapi2;

然后在合适的地方进行初始化：

h\_cctapi=new CCCTAPIAppUSB(this);

h\_cctapi2=new CCCTAPIAppUSB(this);

即可在后面使用，例如开启采集中：

void CUsbControlDlg::OnBnClickedBtnVideocapture()

{

if (!m\_bUsbOpen)

{

SetDlgItemText(IDC\_STATIC\_TEXT, L"USB未打开。");

return;

}

GetDlgItem(IDC\_RADIO1)->EnableWindow(false);

GetDlgItem(IDC\_RADIO2)->EnableWindow(false);

cv::namedWindow("disp");

cv::namedWindow("disp1");

HWND hWnd = (HWND)cvGetWindowHandle("disp");//获取子窗口的HWND

HWND hParentWnd = ::GetParent(hWnd);//获取父窗口HWND。父窗口是我们要用的

//隐藏窗口标题栏

long style = GetWindowLong(hParentWnd, GWL\_STYLE);

style &= ~(WS\_SYSMENU);

SetWindowLong(hParentWnd, GWL\_STYLE, style);

SetDlgItemText(IDC\_STATIC\_TEXT,L"采集中...");

if(h\_cctapi->startCap(g\_height,g\_width,RawCallBack,0)<0)

{

SetDlgItemText(IDC\_STATIC\_TEXT,L"USB设备打开失败！");

GetDlgItem(IDC\_RADIO1)->EnableWindow(true);

GetDlgItem(IDC\_RADIO2)->EnableWindow(true);

GetDlgItem(IDC\_CHECK\_SAVE\_VEDIO)->EnableWindow(true);

return;

}

if(h\_cctapi2->startCap(g\_height,g\_width,RawCallBack2,1)<0)

{

SetDlgItemText(IDC\_STATIC\_TEXT,L"USB2设备打开失败！");

GetDlgItem(IDC\_RADIO1)->EnableWindow(true);

GetDlgItem(IDC\_RADIO2)->EnableWindow(true);

GetDlgItem(IDC\_CHECK\_SAVE\_VEDIO)->EnableWindow(true);

return;

}

}

### 每个类CCCTAPIUSB,可以使用h\_cctapi->调用以下功能：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： startCap

输入参数： int height ：高度

int width：宽度

LPMV\_CALLBACK2 CallBackFunc： 回调函数

返回值： 成功时返回0，否则返回-1.

功能：开始采集。该函数在内部打开USB，开始读取下位机数据，并将回调函数的地址传入。

函数原型： int startCap(int height,int width,LPMV\_CALLBACK2 CallBackFunc);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： stopCap

输入参数：

返回值： 成功时返回0，否则返回-1.

功能：停止采集

函数原型： int stopCap ();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： WrSensorReg

输入参数： unsigned short iAddr: 2字节Sensor寄存器地址， unsigned short iValue: 2字节Sensor寄存器值

返回值： 无.

功能：写Sensor寄存器。

函数原型： void WrSensorReg(unsigned short iAddr, unsigned short iValue);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： RdSensorReg

输入参数： unsigned short iAddr: 2字节Sensor寄存器地址

返回值： unsigned short： 2字节寄存器值

功能：读Sensor寄存器。

函数原型： unsigned short RdSensorReg(unsigned short);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： InitSensor

输入参数： 无

返回值： 无

功能：初始化Sensor

函数原型：void InitSensor(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： GetFpsMbs

输入参数： int &FCount：每秒钟帧数， long &lBytePerSecond 以MB为单位的每秒钟数据量

返回值： 无

功能：获取每秒钟帧数和数据量

函数原型：void GetFpsMbs(int &FCount,long &lBytePerSecond );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： CyUsb\_Init

输入参数： 无

返回值： 成功时返回0，否则返回-1.

功能：初始化USB

函数原型：CyUsb\_Init();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： CloseUsb

输入参数： 无

返回值： 成功时返回0，否则返回-1.

功能：关闭USB

函数原型：CloseUsb();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： OpenUsb

输入参数： 无

返回值： 成功时返回0，否则返回-1.

功能：打开USB

函数原型：OpenUsb()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： SetTrigMode

输入参数： enum TRIGMODE m

返回值： 无

功能：设置触发模式。

typedef enum TRIGMODE{TgMd\_Auto,TgMd\_FPGA,TgMd\_Soft,TgMd\_OutSig};

分别为自动触发，fpga触发，软触发，外部信号触发。

函数原型： void SetTrigModAuto(TRIGMODE );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setFpgaFreq

输入参数： unsigned char fpgafreq： 触发频率

返回值： 无

功能：设置fpga触发模式频率

函数原型： void setFpgaFreq(unsigned char fpgafreq);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setGainValue

输入参数： unsigned short GainValue： 增益值

返回值： 无

功能：设置增益值

函数原型： void setGainValue(unsigned short GainValue);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setAutoGainExpo

输入参数： bool isAutoGain： 自动增益，bool isAutoExpo ：自动曝光

返回值： 无

功能：设置增益和曝光模式。

isAutoGain=true; isAutoExpo=true: 自动增益, 自动曝光

isAutoGain=true; isAutoExpo=false: 自动增益, 手动曝光

isAutoGain=false; isAutoExpo=true: 手动增益, 自动曝光

isAutoGain=false; isAutoExpo=false: 手动增益, 手动曝光

函数原型： void setAutoGainExpo(bool isAutoGain, bool isAutoExpo);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setExpoValue

输入参数： unsigned short ExpoValue： 曝光值

返回值： 无

功能：设置曝光值

函数原型： void setExpoValue(unsigned short ExpoValue);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setResolution

输入参数： unsigned char resNo： 分辨率模式号

返回值： 无

功能：设置曝分辨率

resNo=0： 752\*480

resNo=1： 640\*480

函数原型： void setResolution(unsigned char resNo);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setNormal

输入参数： 无

返回值： 无

功能：设置正常模式，即无镜像

函数原型： void setNormal(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setXmirror

输入参数： 无

返回值： 无

功能：设置X镜像

函数原型： void setXmirror(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： setYmirror

输入参数： 无

返回值： 无

功能：设置Y像

函数原型： void setYmirror(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： SendUsbSpeed2Fpga

输入参数： unsigned char speedType

返回值： 无

功能：将USB接口类型（2.0或者3.0）发送给FPGA

SpeedType=0: USB3.0

SpeedType=1: USB2.0

函数原型： void SendUsbSpeed2Fpga(unsigned char speedType);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： RdDeviceID

输入参数：

unsigned char\* buff 存储读取到的数据的buff的指针

int& len，存储读取到数据的长度

返回值： int读出数字的长度，正常值为1

功能：读取设备ID，长度为1个字节。

函数原型： int RdDeviceID(unsigned char\* buff,int& len);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： WrDeviceID

输入参数：

unsigned char\* buff 存储写入到的设备的buff的指针

int& len，写入数据的长度

返回值： int写入数字的长度，正常值为1

功能：写入设备ID，长度为1个字节。

函数原型： int WrDeviceID(unsigned char\* buff,int& len);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： RdDeviceSN

输入参数：

unsigned char\* buff 存储读取到的设备的buff的指针

int& len，读取数据的长度

返回值： int读取数字的长度，正常值为4

功能：读取设备SN，长度为4个字节。

函数原型： int RdDeviceSN (unsigned char\* buff,int& len);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： WrDeviceSN

输入参数：

unsigned char\* buff 存储写入到的设备的buff的指针

int& len，写入数据的长度

返回值： int写入数字的长度，正常值为4

功能：写入设备SN，长度为4个字节。

函数原型： int RdDeviceSN (unsigned char\* buff,int& len);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名： softTrigOnce

输入参数：

返回值： 0

功能：进行一次软触发

函数原型： void softTrigOnce();