所有的接口函数均在CCHVAPI.cpp和CCHVAPI.h中，用户只需要使用这两个文件，其他的文件不需要涉及。

在CCHVAPI.h的最下面，包括了所有的dll导出函数列表，以CCT\_API为前缀。

在CCHVAPI.cpp最底部，有函数名以cs为前缀的c# 的示例包裹函数。一些比较复杂的业务逻辑，特别是涉及到指针操作的，一般在C++中完成，最后用更加简单的形式导出到C#中。

## 发现设备,discovery：

void CUsbControlDlg::OnBnClickedBtnSearch()

{

//search camera

listbox.ResetContent();

try

{

GigEsearchCamera(cameralist);

}

catch(exception &e)

{

CString error(e.what());

SetDlgItemText(IDC\_STATIC\_TEXT, error);

}

map\_camera ::iterator itr;

for(itr=cameralist->begin();itr!=cameralist->end();++itr)

{

std::string temp=itr->first;

CString tempc(temp.c\_str());

listbox.AddString(tempc);

}

UpdateData(TRUE);

}

其中GigESearchCamera(cameralist);函数为发现设备。cameralist是创建的一个自定义map\_camera型，其原型为：

typedef std::map<std::string, CCHCamera\*> map\_camera;

它用来存储搜索到的设备，第一个string为相机的序列号，第二个值为相机的属性。map型必须用一个iterator来遍历里面的所有值，或者是使用key值来查找，它没有类似于数组的下标功能(如char list[2])，在下面的连接按键的功能中需要注意。

## 设备连接connect功能：

void CUsbControlDlg::OnBnClickedBtnConnect()

{

//get info from list selection

int idx = listbox.GetCurSel();

CString str;

if (idx <= -1)

{

map\_camera::iterator itr;

itr = cameralist->begin();

if (itr != cameralist->end())

{

CCHCamera \*c0 = itr->second;

board1 = GigEaddInstance((LPVOID\*)this, RawCallBack, c0);

itr++;

}

if (itr != cameralist->end())

{

CCHCamera \*c1 = itr->second;

board2 = GigEaddInstance((LPVOID\*)this, RawCallBack2, c1);

}

if (board1&&board2>0)

{

str.Format(L"Device connected");

SetDlgItemText(IDC\_STATIC\_TEXT, str);

gb\_imgctrl.EnableWindow(1);

}

else

{

str.Format(L"connect error %d", board1);

SetDlgItemText(IDC\_STATIC\_TEXT, str);

}

return;

}

////////////////////////////以上为双设备示例/////////////////////////////////

else

{

CString strtemp;

listbox.GetText(idx, strtemp);

CT2CA pszConvertedAnsiString(strtemp);

std::string strtemp2(pszConvertedAnsiString);

cameralist->find(strtemp2);

CCHCamera\* c = cameralist->find(strtemp2)->second;

//connect button

board1 = GigEaddInstance((LPVOID\*)this, RawCallBack, c);

if (board1 > 0)

{

str.Format(L"Device connected");

SetDlgItemText(IDC\_STATIC\_TEXT, str);

gb\_imgctrl.EnableWindow(1);

}

else

{

str.Format(L"connect error %d", board1);

SetDlgItemText(IDC\_STATIC\_TEXT, str);

}

}

上半部分为一个双设备连接的示例程序，即如果搜索到2个设备，如果不在设备列表里面选中，则2个设备都连接上。

首先开一个itr，取cameralist中的第一个设备，使用

board1 = GigEaddInstance((LPVOID\*)this, RawCallBack, c0);

RawCallBack为回调函数，c0为从cameralist中取得的一个相机实例。

开启第一个相机实例后，使用itr++，指向cameralist中的第二个设备，再重复开启即可。

## ForceIp

forceIP功能主要用在装机之前烧写ip地址或烧写ee上的参数用，它可以临时的将相机的ip地址更改，保证相机可以与电脑连上进行一些操作，而不用更改电脑的ip地址。

举例：

电脑ip地址为192.168.2.10，

相机ip地址为192.168.1.2,

这时点discovery可以搜索到相机，将相机的ip改为192.168.2.3,点击forceIP即可。

这时可以点击connect，更改setIP下的ip地址，例如可以将相机的ip更改为192.168.3.1（该相机最终将用在192.168.3.1的网卡上）。

videoCapture等按钮这时也可以使用。在重新上电后，上一步设的新的ip地址才会使用。

## 回调函数RawCallBack

在RawCallBack里面，相机传上来的数据保存在thisFrame中，数据的长度为height\*width\*camsize,

例如，假如您的系统为6个相机，每个相机的分辨率为1280\*960，那么传上来的数据总长度就是1280\*960\*6，

各个相机的图像数据从0号相机到5号相机依次排列，例如2号相机的图像就位于1280\*960\*2,长度为1280\*960,

只需要对该数据块按照分辨率进行切割，就是对应的相机的图像。

## 其他设置函数

其他设置函数主要形式为参数+摄像头号的形式来实现的，例如

int GigEsetIP(CCHCamera \*devinfo,int camNum)

第一个参数CCHCamera\*为包含ip地址的设备信息的结构体

第二个参数camNum为设备号，即在Connect按键中通过调用GigeaddInstance得到的。