DLL使用方法：

先调用initDll（分配一些内存），然后调用searchDev，在DLL内部会生成一个onvif设备列表（以下简称设备列表）。

现在你可以调用getNumOfOnvifDev、getURIFromIP、getAllDevURI。

如果你添加或删除了onvif设备，再次调用searchDev，你应该能通过调用getURIFromIP和getAllDevURI来查看新设备。设备列表中本来的信息不会被清除，重复发现的设备会被忽略。searchDev会对两次search之间的列表进行比较，删除不存在的设备。

要清除列表，请调用clearDeviceList。

如果想释放内存，重置DLL，请调用resetDll；注意DLL维护的列表也会被清空。

最后别忘了调用uninitDll来释放内存。

注意：

如果存在VMware虚拟机的适配器，请禁用，否则组包（multicast package）就不知道发到哪里去了。

允许程序通过防火墙。

数据结构：

设备信息结构体：

typedef struct

{

char ip[17]; // 发现的设备ip

char URI[256]; // 发现的设备URI（RTSP）

}deviceInfoArray;

接口函数：

1.

功能：初始化，申请一些内存

返回值：0表示成功，-1表示失败

int initDll(char);

2.

功能：工作完成后释放内存

返回值：0表示成功，-1表示失败

int uninitDll(char);

3.

功能：搜索onvif设备

返回值： 0表示成功，-1表示失败

int searchDev(char);

4.

功能：得到搜索到的设备数量

返回值：设备数量，-1出错

int getNumOfOnvifDev(char);

5.

功能：根据IP得到URI（RTSP），需要授权信息

注意：你需要申请本函数所需的内存

输入：

IP：指向C风格IP字符串的指针

IPBufferLen：字符串所占的字节数

Username：指向C风格用户名字符串的指针

Password：指向C风格密码字符串的指针

URI：指向C风格URI（RTSP）字符串缓存的指针

URIBufferLen：URI（RTSP）字符串缓存所占的字节数

返回值：0表示成功，-1表示失败

int getURIFromIP(char\* IP, size\_t IPBufferLen, char\* URI, size\_t URIBufferLen, char\* username, char\* password);

6.

功能：获得所有onvif设备的URI（RTSP）

注意：你需要申请本函数所需的内存

输入：Num：所有onvif设备的数量

输出：infoArray：指向设备信息数组的指针

返回值：放入设备信息数组的信息数量，-1表示失败

int getAllDevURI(deviceInfoArray\* infoArray, size\_t Num);

7.

功能：清除本DLL维护的onvif设备列表信息

返回值：0表示成功，-1表示失败。

int clearDeviceList(void);

8.

功能：重置DLL，在使用本函数前请三思

返回值：0表示成功，-1表示失败。如果失败了，只能重新加载DLL

int resetDll(void)