

## 实验 5.5 音视频数据获取与展示

音视频是物联网信息的重要组成部分，物联网系统开发过程中，经常需要集成摄像头的音视频信息，本实验用于学习利用 C#语言获取摄像头音视频信息的编程方法。

### 1. 实验目的

通过调用物联网操作系统提供的 API 接口，学习与操作系统接口交互及从该接口获取本机音视频设备列表以及视频流的方法，体会操作系统接口与音视频设备变化的无关性和设计该层接口的目的和作用。

### 2. 实验要求

首先学习通过 HTTP 协议进行接口调用的方法，学习 HTTP-FLV 或 RTMP 协议视频流在 C/S 软件展示方法。根据实验内容，使用 C# 编写视频流播放 C/S 软件，实验过程中认真调试、思考和记录，分析和体会通过操作系统获取音视频设备能力的流程和方法。参考学时：2 学时。

### 3. 实验内容

本实验利用物联网操作系统，提供音视频设备的管理能力，通过了解视频服务接口说明，编写程序调用相应接口实现音视频设备列表及对应视频流的获取，之后将获取到的视频流在程序中展示。

### 4. 实验环境及设备

#### 1) 开发环境

Microsoft Visual Studio-2022（.NET 桌面开发、通用 Windows 平台开发）。

#### 2) 实验设备

- （1）用来编写 C#程序的计算机一台；
- （2）安装物联网操作系统的计算机一台；
- （3）usb 摄像头或电脑自带摄像头一个。

### 5. 实验预备知识

本实验提供了封装好的通过 C# 发送 HTTP 请求以及播放视频流的必要函数，以便在后面的视频获取编程部分调用。下面对相关函数进行详细介绍。

### 1) httpreq 函数

函数声明：private string httpreq(string strUrl, string method, string data)。

函数作用：发送 HTTP 请求，调用物联网操作系统提供的视频服务接口。

接收参数说明：strUrl 为字符串类型的 HTTP 请求地址；method 为字符串类型的请求方法，可选值有 GET、POST、DELETE、UPDATE；data 为字符串类型的 JSON 格式数据。

返回结果说明：返回结果为 HTTP 请求响应的 JSON 格式的字符串数据。

使用示例：

```
string strUrl = "http://127.0.0.1:8998/isAlive";  
string result = httpreq(strUrl, "GET", "");
```

### 2) JsonConvert.DeserializeObject 函数

函数声明：classname JsonConvert.DeserializeObject<classname>(string jsonstr)。

函数作用：将 JSON 格式的字符串数据转换为 C# 对象。

接收参数说明：classname 为要转换对象的类名；jsonstr 为 classname 类型对应对象实例的 JSON 格式字符串数据。

返回结果说明：返回结果为 classname 类型的对象。

使用示例：

```
public class classname1  
{  
    public bool success { get; set; }  
    public string message { get; set; }  
}  
classname1 obj = JsonConvert.DeserializeObject<classname1>(  
    "{\"success\":true,\"message\":\"请求成功\"}"  
);
```

### 3) JsonConvert.SerializeObject 函数

函数声明：string JsonConvert.SerializeObject(classname obj)。

函数作用：将 C# 对象转换为 JSON 格式字符串数据。

接收参数说明：obj 为要转换为 JSON 格式字符串数据的 C# 对象。

返回结果说明：返回结果为 JSON 格式的字符串数据。

使用示例：

```
public class classname1
{
    public bool success { get; set; }
    public string message { get; set; }
}

classname1 obj = new classname1();
classname1.success = true;
classname1.message = "请求成功";
string jsonstr = JsonConvert.SerializeObject(classname1);
```

#### 4) createDevice 函数

函数声明：private string createDevice(string cam,string aud)。

函数作用：构造表 5-5 错误!文档中没有指定样式的文字。-3 和表 5-5 错误!文档中没有指定样式的文字。-4 接口内描述的请求参数。

接收参数说明：cam 为要打开的摄像头名称；aud 为要打开的麦克风名称（当不存在麦克风设备时，需要发送空字符串）。

返回结果说明：返回结果封装好的表 5-5 错误!文档中没有指定样式的文字。-3 和表 5-5 错误!文档中没有指定样式的文字。-4 接口内描述的请求参数。

使用示例：

```
string cam = "摄像头名称";
string aud = "";
string data = createDevice( cam , aud );
```

#### 5) \_paly 函数

函数声明：private void \_play(string url)。

函数作用：启动视频播放组件（组件启动需要 3-5 秒）。

接收参数说明：url 为要播放的视频流地址。

返回结果说明：无。

使用示例：

```
string rtmp = "rtmp://127.0.0.1:1935/myapp/92ae7616eb8d104";
_play(rtmp);
```

## 6. 视频服务接口说明

本实验视频服务接口由物联网操作系统提供, 需要保证编写程序的计算机和安装有物联网操作系统的计算机在同一个局域网内, 视频服务的 IP 地址为安装有物联网操作系统计算机的局域网 IP 地址, 端口号为 8998。

### 1) 判断视频服务接口

该接口用来判断视频服务是否可用, 返回结果内 success 为 true 即为可用。  
接口详细内容如表 5-5 错误!文档中没有指定样式的文字。-1。

表 5-5 错误!文档中没有指定样式的文字。-1 判断视频服务是否可用接口

接口说明	判断视频服务是否可用		
协议	HTTP	方法	GET
URL 地址	[serverIP]:8998/isAlive		
请求参数	参数类型	无	
	参数说明	无	
	参数示例	无	
响应参数	参数类型	Response (json 报文)	
	参数说明	请求返回的数据, 格式如下	
	参数示例	{ "success":true, //请求返回状态 "message":"获取成功", //请求返回消息 "token":""," "code":1, "data": [] }	

### 2) 获取音视频设备列表接口

该接口用来获取与物联网操作系统连接的音视频设备列表。接口详细内容如表 5-5 错误!文档中没有指定样式的文字。-2。

表 5-5 错误!文档中没有指定样式的文字。-2 音视频设备列表获取接口

接口说明	音视频设备列表获取接口		
协议	HTTP	方法	GET
URL 地址	[serverIP]:8998/nvr_management/getCameraList		
请求参数	参数类型	application/x-www-form-urlencoded	
	参数说明	本地用户号, 此处统一发送 userId=1	
	参数示例	?userId=1	
响应参数	参数类型	Response (json 报文)	
	参数说明	请求返回的数据, 格式如下	
	参数示例	{ "success":true, //请求返回状态	

		<pre> "message": "获取成功", //请求返回消息 "token": "", "code": 1, "data": { //返回的摄像头名称列表 "cameraDeviceArray": ["\\\"HIK 1080P Camera\\\""], //返回的麦克风名称列表 "audioDeviceArray": ["\\\"HIK 1080P Camera Audio\\\""] } } </pre>
--	--	--

### 3) 获取视频流链接接口

该接口用来获取视频设备的 HTTP-FLV 和 RTMP 协议的视频流。接口详细内容如表 5-5

表 5-5 获取视频流链接接口

接口说明	获取视频流链接接口		
协议	HTTP	方法	POST
URL 地址	[serverIP]:8998/nvr_management/startPush		
请求参数	参数类型	application/json	
	参数说明	要打开的音视频设备信息	
	参数示例	<pre> { "device_info":{ //选择的摄像头名称 "currentCam": "HIK 1080P Camera", //选择的麦克风 "currentAud": "HIK 1080P Camera Audio", //安装操作系统的 IP (只有 IP 地址和端口) "serverIp": "127.0.0.1:8998", //设备类型，使用固定值"CAM" "device_type": "CAM", //视频流类型，使用固定值"flv" "stream_type": "flv", //固定值 } } </pre>	
响应参数	参数类型	Response (json 报文)	
	参数说明	请求返回的数据，格式如下	
	参数示例	<pre> { "success": true, //请求返回状态 "message": "获取成功", //请求返回消息 "token": "", </pre>	

		<pre>"code":1, "data":{     //返回的 HTTP-FLV 视频流链接     "http":"http://127.0.0.1:8010/live?port=1935&amp;app=myapp&amp;stream=92a1d6a2fb987e3cb6ea62de95568e7616eb80da51c3ff9f096e5a0ffdbed104",     //返回的 RTMP 视频流链接     "rtmp":"rtmp://127.0.0.1:1935/myapp/92a1d6a2fb987e3cb6ea62de95568e7616eb80da51c3ff9f096e5a0ffdbed104"   } }</pre>
	参数示例	<pre>{   success: false, //请求返回状态   message: '流出错', //请求返回消息   token: "",   code: 1,   data: {} }</pre>

4) 关闭视频流接口

该接口用来关闭视频流。接口详细内容如表 5-5 错误!文档中没有指定样式的文字。-4。

表 5-5 错误!文档中没有指定样式的文字。-4 关闭视频流接口

接口说明	关闭视频推流接口		
协议	HTTP	方法	DELETE
URL 地址	[serverIP]:8998/nvr_management/stopPush		
请求参数	参数类型	application/json	
	参数说明	要打开的摄像头信息	
	参数示例	<pre>{   "device_info":{     //选择的摄像头名称     "currentCam":"HIK 1080P Camera",     //选择的麦克风     "currentAud": "HIK 1080P Camera Audio",     //安装操作系统的 IP (只有 IP 地址和端口)     "serverIp": "127.0.0.1:8998",     //设备类型，使用固定值"CAM"     "device_type": "CAM",     //视频流类型，使用固定值"flv"     "stream_type": "flv", //固定值   } }</pre>	

响应参数	参数类型	Response (json 报文)
	参数说明	请求返回的数据，格式如下
	参数示例	<pre>{   success: true,   message: '成功',   token: "",   code: 1, }</pre>

## 7. 实验步骤

### 1) 摄像头设备连通性测试

在使用摄像头设备之前通过操作系统提供的能力测试摄像头设备的连通性。

#### (1) 进入视频设备配置

在系统首页点击设备管理内的视频设备配置，如图 5-5-1 所示。



图 5-5-1 操作系统首页

#### (2) 进入本机摄像头选项卡

进入视频设备配置页面后，点击本机摄像头选项卡进入音视频设备选择页面。

如图 5-5-2 所示。



图 5-5-2 本机摄像头选项卡

### (3) 选择摄像头

通过下拉框列表选择需要测试的摄像头和麦克风，之后页面中便会展示相应的摄像头画面。如图 5-5-3 所示。



图 5-5-3 选择摄像头

### 2) 视频获取编程

本实验提供搭建好的视频开发项目框架，编写程序时需要在项目的特定文件内根据需求在相应位置填充代码。

打开计算机” 璘雾实验系统\实验 5.5” 文件夹，项目文件结构如图 5-5-4 所示。



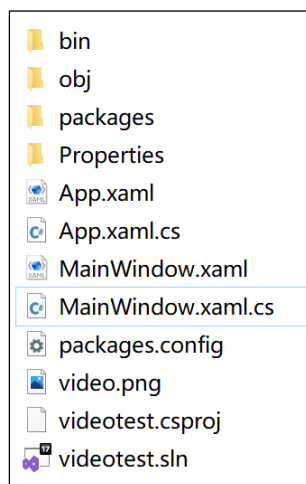


图 5-5-4 项目文件结构

本实验需要操作的文件有 videotest.sln、MainWindow.xaml 以及 MainWindow.xaml.cs 这三个文件。

videotest.sln 为导入项目时选择的项目入口文件。

MainWindow.xaml 为本实验提供的软件的界面结构文件，里面包含了：视频服务地址输入框、获取音视频设备按钮、摄像头列表下拉框、麦克风列表下拉框、播放按钮、停止按钮。界面效果如图 5-5-5 所示。

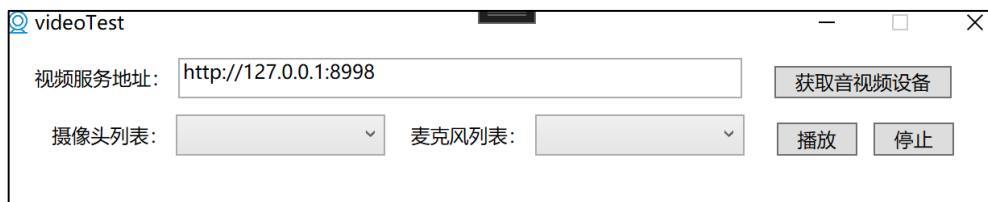


图 5-5-5 视频实验软件界面

MainWindow.xaml.cs 为对应于 MainWindow.xaml 的交互逻辑文件，除了在 5.实验预备知识中介绍的封装方法外，此文件还包含了 `getDeviceList`、`play` 和 `stop` 这三个需要编程填充代码的函数，这三个函数均需要调用物联网操作系统视频服务，在调用服务前需要先访问表 5-5 错误!文档中没有指定样式的文字。-1 判断视频服务是否可用接口描述的接口判断视频服务是否可用。

`getDeviceList` 函数对应于获取音视频设备按钮，需要在里面调用表 5-5 错误!文档中没有指定样式的文字。-2 描述的接口获取物联网操作系统连接的音视频设备列表，获取到音视频设备列表后，将设备动态的在摄像头列表和麦克风列表对应的下拉框控件中展示。

`play` 函数对应于播放按钮，点击播放按钮时判断是否选择了摄像头，否则提

示用户请选择摄像头。麦克风为可选值，若没有麦克风设备发送请求时需将麦克风字段置为空字符串。符合条件后调用表 5-5 错误!文档中没有指定样式的文字。-3 接口获取摄像头 HTTP-FLV 和 RTMP 协议的视频流并播放，成功播放后将除停止按钮以外的所有控件（输入框、按钮、下拉框）设置不可用。

stop 函数对应于停止按钮，点击停止按钮时调用表 5-5 错误!文档中没有指定样式的文字。-4 描述的接口关闭视频流并将设置为禁用的控件设置为可用。

实验最终效果如图 5-5-6 所示。



图 5-5-6 视频实验软件界面

## 8. 实验报告

根据如下要求完成实验报告：

- （1）总结本实验中获取音视频数据的实验步骤；
- （2）打印播放按钮对应 play 函数中相关编程语句；
- （3）描述实验过程中遇到的问题和解决办法；
- （4）总结实验的收获与体会。