**Лабораторная работа 4.**

**Разработка программных модулей**

**Элементы сетевого программирования**

Цель: Ознакомиться с теорией и практикой создания элементов сетевого программирования в среде .Net.

**Теория**

Рассмотрим примеры передачи видеопотока по протоколу UDP.

**Протокол UDP**

**Задание 1**: Реализовать ниже приведенный пример проекта (Webcam proj) по передаче видеопотока с помощью протокола UDP:

**Пример приложения UDP**

Программа будет состоять из двух отдельных программ:

Первый модуль или проект назовем Consumer – этот проект будет получать информацию и отображать.

Второй модуль или проект назовем Produser – этот проект будет отправлять информацию с web камеры (видеопоток).

Решение или пространство имен назовем Webcam proj.

За основу был взять проект на базе платформы Windows Forms.

**Первый модуль** создадим проект Consumer, Решение или пространство имен назовем Webcam proj. За основу возьмем платформу Windows Forms. На форму добавим объект PictureBox.

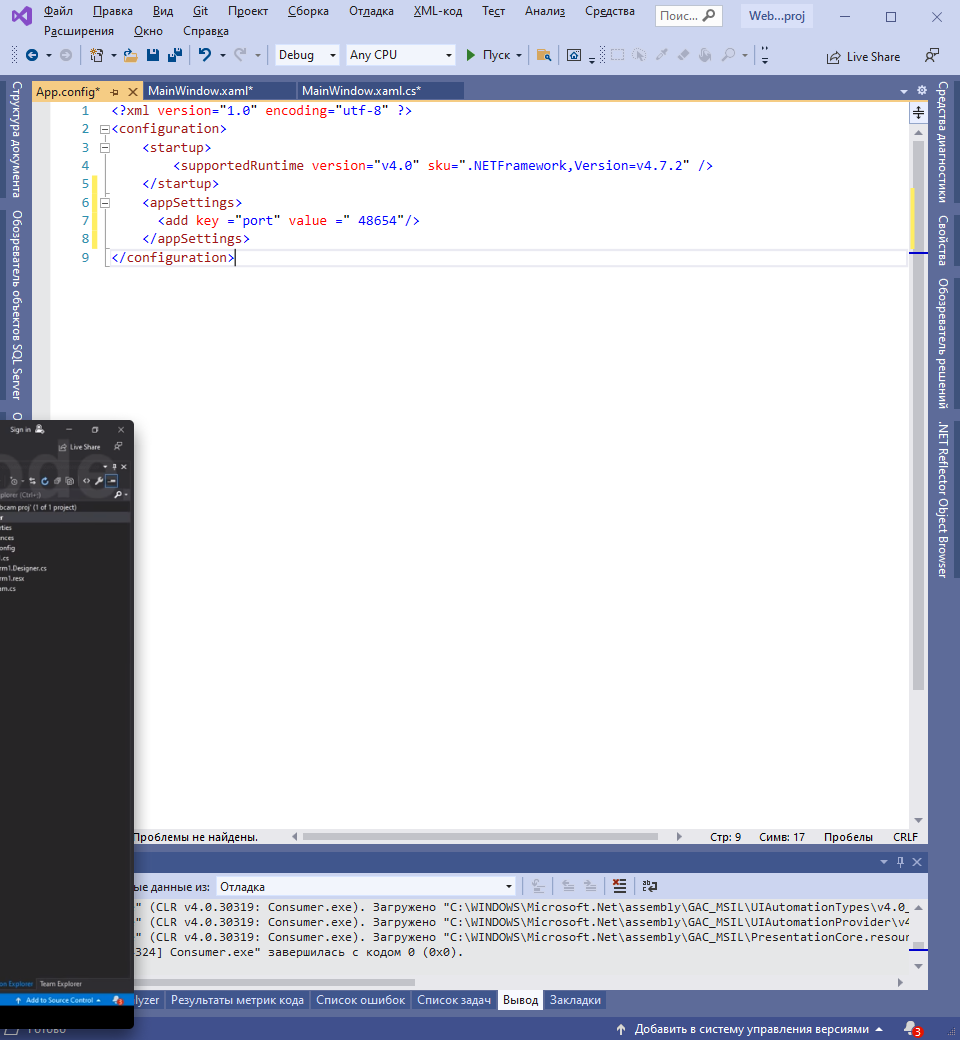
В обозревателе объектов откроем файл App.config и добавим туда информацию о подключении к источнику видео.

<appSettings>

<add key =**"port"** value =**" 48654"**/>

</appSettings>

Тогда у вас этот файл примет вид



Для окна Window Forms добавим обработчик событий Load.

private async void Form1\_Load(object sender, EventArgs e) **{**

**}**

Так как событие будем обрабатывать асинхронно, добавим **async**.

В обработчике события добавим объявления переменной port которой присвоим приведенное к типу int значение переменной port из конфигурационного файла:

var port **=** int**.**Parse**(**ConfigurationManager**.**AppSettings**.**Get**(**"port"**));**

Для адекватного подключения класса

ConfigurationManager

подключим с помощью директивы using пространство имен

**using** System**.**Configuration**;**

Далее добавим переменную класса UdpClient передав в качестве аргумента в конструктор класса параметр port.

var client **=** **new** UdpClient**(**port**);**

Далее в бесконечном цикле while будет получать данные асинхронно, обращаясь к асинхронному методу ReceiveAsync**()** объекта client

**while** **(true)**

**{**

var data **=** **await** client**.**ReceiveAsync**();**

**}**

Метод ReceiveAsync асинхронно UDP-датаграмму отправленное удаленным узлом.

С помощью await из курса многопоточного программирования дожидаемся результата выполнения метода ReceiveAsync. Далее этот результат помещается в объект data.

Создадим объект ms типа MemoryStream в конструктор которого передаем Bufer с байтами которые получили через UDP.

using (var ms = new MemoryStream(data.Buffer))

{

pictureBox1.Image = new Bitmap(ms);

}

Добавим текстовое сопровождение в заголовке Windows Forms о количестве байт переданных в буфер

Text = $"Bytes received: {data.Buffer.Length \* sizeof(byte)}";

Отдельно добавим еще один обработчик событий двойного нажатия на объект pictureBox, которое будет отображать сообщение о текущем IP адресе.

private void pictureBox1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

var host = Dns.GetHostEntry(Dns.GetHostName());

MessageBox.Show(string.Join("\n", host.AddressList.Where(i => i.AddressFamily == AddressFamily.InterNetwork)));

}

**Второй модуль** создадим проект Produser, Решение или пространство имен назовем Webcam proj. За основу возьмем платформу Console. На форму добавим объект PictureBox.

В обозревателе объектов откроем файл App.config и добавим туда информацию о подключении к источнику видео.

<appSettings>

<add key="consumerIp" value="192.168.9.168"/>

<add key="consumerPort" value="48654"/>

</appSettings>

Соответственно значение IP у вас будет свое.

Код представленного проекта

using System;

using System.Configuration;

using System.Net;

using AForge.Video.DirectShow;

using System.Drawing;

using System.IO;

using System.Drawing.Imaging;

using System.Net.Sockets;

using System.Runtime.InteropServices;

class Program

{

const int SW\_HIDE = 0;

const int SW\_SHOW = 5;

[DllImport("kernel32.dll")]

static extern IntPtr GetConsoleWindow();

[DllImport("user32.dll")]

static extern bool ShowWindow(IntPtr hWdn, int nCmdShow);

private static IPEndPoint consumerEndPoint;

private static UdpClient udpClient = new UdpClient();

static void Main(string[] args)

{

var consumerIp = ConfigurationManager.AppSettings.Get("consumerIp");

var consumerPort = int.Parse(ConfigurationManager.AppSettings.Get("consumerPort"));

consumerEndPoint = new IPEndPoint(IPAddress.Parse(consumerIp), consumerPort);

Console.WriteLine($"consumer: {consumerEndPoint}");

FilterInfoCollection videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);

VideoCaptureDevice videoSource = new VideoCaptureDevice(videoDevices[0].MonikerString);

//videoSource.VideoResolution = videoSource.VideoCapabilities

videoSource.NewFrame += videoSource\_NewFrame;

videoSource.Start();

Console.WriteLine("\n Нажмите Enter чтобы спрятать консоль . . .");

Console.ReadLine();

ShowWindow(GetConsoleWindow(),SW\_HIDE);

}

private static void videoSource\_NewFrame(object sender, AForge.Video.NewFrameEventArgs EventArgs)

{

var bmp = new Bitmap(EventArgs.Frame, 800, 600);

try

{

using (var ms = new MemoryStream())

{

bmp.Save(ms, ImageFormat.Jpeg);

var bytes = ms.ToArray();

udpClient.Send(bytes,bytes.Length, consumerEndPoint);

}

}

catch(Exception e)

{

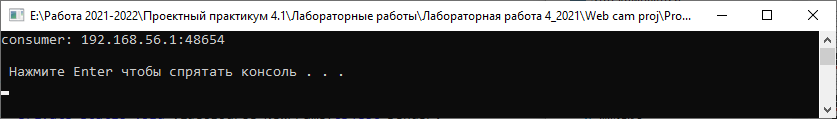
Console.WriteLine(e);

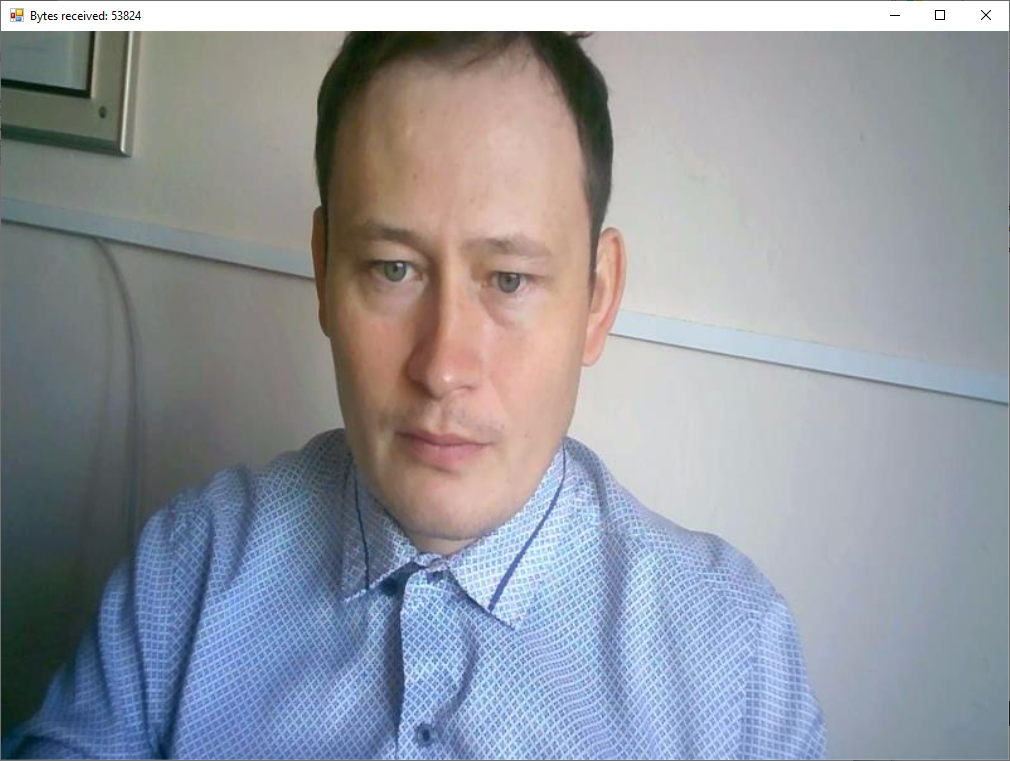
}

}

}

Результат работы программы:





**Задание 2**: Реализовать выше приведенный проект на платформе по передаче видеопотока с помощью протокола UDP платформе WPF.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. **Фленов М. Е.**Библия C#. — 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2019. — 512 с.: ил.
2. URL: https://www.interestprograms.ru/sources/csharp/os-windows/desktop/setevye-prilozheniya (дата обращения 10.04.2021)
3. URL: https://www.youtube.com/watch?v=m0PpxahUmMk (дата обращения 10.04.2021)