

Оглавление

Исходный код.....	1
Form1.cs	1
Form2.cs	17
Form3.cs	25
smallwindow.cs.....	31
tcpserver.cs.....	43

Исходный код

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using AForge.Video;
using AForge;
using AForge.Video.DirectShow;

namespace VideoStudio
{
    public partial class Form1 : Form
    {
        #region объявление переменных
        private smallwindow[] preview = new smallwindow[number_of_small_panels];
        // объекты предпросмотра
        private static int number_of_small_panels = 6;
        // количество маленьких панелей
        private static int number_of_small_buttons= 4;
        // количество маленьких кнопок
        private int FormWidth;
        // ширина формы
        private int FormHeight;
        // высота окна формы
        private int selected_device = 0;
        // выбранный сейчас девайс
        private int lastselected_device = 0;
        // выбранный до этого девайс (не используется тк не используется функция)
        private int counter = 0;
        // переменная счетчик при смене источников(не используется тк не используется функция)
```

```

        bool online = false;
// флаг необходимости транслировать
        private bool Record_is_work = false;
// флаг необходимости записывать
        private string way_to_folder = "";
// путь к папке назначения для хранения записанных файлов
        private int second_selected_device = -1;
// номер устройства для pip (-1 -- pip отключено)
        Bitmap result;
// результирующая картинка
        private int nWidth = 1280;
// разрешение выходного потока
        private int nHeight = 720;
// разрешение выходного потока

        //BigpictureBox
        private System.Windows.Forms.PictureBox BigpictureBox = new
System.Windows.Forms.PictureBox(); // окно вывода главного изображения
        private System.Drawing.Point BigpictureBoxLocation;
// верхний левый угол главного окна вывода изображения
        private System.Drawing.Size BigpictureBoxSize;
//размер главного окна вывода изображения

        //panel
        private System.Windows.Forms.Panel panel = new System.Windows.Forms.Panel();
// первая панель для маленьких окон
        private System.Drawing.Point panelLocation;
// верхний левый угол первой панели для маленьких окон
        private System.Drawing.Size panelSize;
//размер первой панели для маленьких окон

        //smallpanels
        private System.Windows.Forms.Panel[] smallpanels = new
System.Windows.Forms.Panel[number_of_small_panels]; // массив панелей маленьких окон
        private System.Drawing.Point[] smallpanelsLocation = new
System.Drawing.Point[number_of_small_panels]; // инициализация массива расположения
        маленьких панелей
        private System.Drawing.Size[] smallpanelsSize = new
System.Drawing.Size[number_of_small_panels]; // инициализация массива размера
        маленьких панелей

        //smallbuttons
        private System.Drawing.Point[] smallbuttonLocation = new
System.Drawing.Point[number_of_small_buttons]; // инициализация массива расположения
        кнопок маленьких панелей
        private System.Drawing.Size[] smallbuttonSize = new
System.Drawing.Size[number_of_small_panels]; // инициализация массива размера
        кнопок маленьких панелей

        //smallpictureBox
        private System.Drawing.Point smallpictureBoxLocation; // расположение маленького
pictureBox
        private System.Drawing.Size smallpictureBoxSize; // размер маленького
pictureBox

        //rec/stop button
        private System.Windows.Forms.Button recbutton = new
System.Windows.Forms.Button(); // кнопка записи
        private System.Drawing.Point recbuttonLocation;
// верхний левый угол кнопки записи
        private System.Drawing.Size recbuttonSize;
//размер кнопки записи

        //cut button

```

```

        private System.Windows.Forms.Button cutbutton = new
System.Windows.Forms.Button();           // кнопка резки видео
        private System.Drawing.Point cutbuttonLocation;
// верхний левый угол кнопки резки видео
        private System.Drawing.Size cutbuttonSize;
//размер кнопки резки видео

        //setting button
        private System.Windows.Forms.Button settingbutton = new
System.Windows.Forms.Button();           // кнопка настройки
        private System.Drawing.Point settingbuttonLocation;
// верхний левый угол кнопки настройки
        private System.Drawing.Size settingbuttonSize;
//размер кнопки настройки

        Color button_color;

#endregion

//временные переменные

string paramsforffmpeg; //строка параметров дл ffmpeg
int port = 5000; // номер порта для вывода информации
bool rec_main;
bool rec_all;
string more_settings=""; //дополнительные настройки
string ipaddress;

private tcpserver server=new tcpserver();

Form3 settings; // объект формы настроек
bool settings_had_been_opened = false; // флаг необходимости восстановить
установленные настройки в окне настроек

int videocouner;
int audiocounter;
byte[] buffer;

System.Windows.Forms.Timer time; // система сохранения
System.Windows.Forms.Timer time2; //создание картинки + сеть

private void Form1_Load(object sender, EventArgs e)
{
    selected_device = 0;
    if (Sets_sizes(System.Windows.Forms.Screen.PrimaryScreen.Bounds.Width) ==
true) // установка параметров размеров элементов и возврат состояния корректности
инициализации(true-все хорошо)
    {
        createconponents(); // создаем визуальные элементы

        for (int i = 0; i < number_of_small_panels; i++) // создаем объекты
маленьких окон
        {
            preview[i] = new smallwindow(smallpanels[i], smallbuttonLocation,
smallbuttonSize, smallpictureBoxLocation, smallpictureBoxSize, i);
        }
        button_color = recbutton.BackColor;
        Click_event(); // подключение к обработке событий

        change_onair_color();
        time = new System.Windows.Forms.Timer(); //таймер1
        time.Interval = 31;
    }
}

```

```

        time.Enabled = true;
        time.Tick += time_Tick;
        time.Start();

        time2 = new System.Windows.Forms.Timer(); //таймер1
        time2.Interval = 31;
        time2.Enabled = true;
        time2.Tick += time2_Tick;
        time2.Start();

    }

    private void time_Tick(object sender, EventArgs e)// обработка пеервого таймера
    {
        worker1(); //система записи
    }

    private void time2_Tick(object sender, EventArgs e)// обработка второго таймера
    {

        Bitmap result2 = new Bitmap(nWidth, nHeight);//зум картинки
        using (Graphics g = Graphics.FromImage(result2))
            g.DrawImage(preview[selected_device].Video_sourse_cach, 0, 0, nWidth,
nHeight);

        if (second_selected_device != -1)//pip зум и наложение
        {
            using (Graphics g = Graphics.FromImage(result2))
                g.DrawImage(preview[second_selected_device].Video_sourse_cach, nWidth
- nWidth / 3 - 20, nHeight - nHeight / 3 - 20, nWidth / 3, nHeight / 3);
        }

        result = result2;
        BigpictureBox.Image = result;// вывод в главное окно

        if (online == true)
        {
            server.sender(result); //отправка по сети
        }
    }

    private void worker1()//запись в файлы
    {
        //for (int i = 0; i < number_of_small_panels; i++) // переделать структуру
приема по сети
        //{
        //    preview[i].update_information();
        //}

        if (Record_is_work == true)// запись
        {
            for (int i = 0; i < number_of_small_panels; i++)
            {
                preview[i].videowriter();
            }
            //videocouner++;

            //if (videocouner > 30)
            //{
            //    videocouner = 0;
            //}
        }
    }

```

```

        //if (videocouner % 3 == 0)
        //{
        //    for (int i = 0; i < number_of_small_panels; i++)
        //    {
        //        preview[i].audiowriter();
        //    }
        //}

        GC.Collect(); // сбор мусора
    }

    private void Image_changer(int newselected_device, int lastselected_device1)//(не
используется)смена изображения с переходом
    {
        if (newselected_device == lastselected_device1)
        {
            BigpictureBox.Image = preview[selected_device].Video_sourse_cach;
        }
        else
        {
            if(counter<=10)
            {
                if (counter %2==0)
                    BigpictureBox.Image =
preview[lastselected_device1].Video_sourse_cach;
                else
                    BigpictureBox.Image =
preview[newselected_device].Video_sourse_cach;
                counter++;
            }
            else
            {
                counter = 0;
                lastselected_device = selected_device;
            }
        }
    }

    private void settingbutton_Click(object sender, EventArgs e)//обработка нажатия
кнопки настроек
    {
        if (settings_had_been_opened == false)
        {
            settings = new Form3();
            settings.Show();
            settings.FormClosing += settings_FormClosing;
        }
        else
        {
            settings = new Form3();
            //устанавливаем значения через свойства
            settings.Widthofpicture = nWidth;
            settings.Heightofpicture = nHeight;
            settings.rec_main = rec_main;
            settings.rec_all = rec_all;
            settings.folder = way_to_folder;
            settings.online = online;
            settings.more_settings = more_settings;
            settings.port = port;
            settings.ipaddress = ipaddress;
            // отображаем форму
            settings.Show();
            settings.FormClosing += settings_FormClosing;
        }
    }

```

```

    }

}

void settings_FormClosing(object sender, FormClosingEventArgs e)// получение
параметров с формы настроек
{
    if (settings.allgood == true)
    {
        nWidth = settings.Widthofpicture;
        nHeight = settings.Heightofpicture;
        rec_main = settings.rec_main;
        rec_all = settings.rec_all;
        way_to_folder = settings.folder;

        if (settings.online == true)
        {
            ipaddress = settings.ipaddress;
            more_settings = settings.more_settings;
            port = settings.port;
            online = settings.online;
            paramsforffmpeg = settings.mainparams;

        }
        else
        {
            ipaddress = "";
            more_settings = "";
            port = 5000;
            online = settings.online;
        }
        settings_had_been_opened = true;
        settings.Dispose();

        try
        {
            server.closing();
        }
        catch
        {
        }

        if (settings.online == true)
        {
            server = new tcpserver(port+1,paramsforffmpeg);// используется порт+1
для отправки в ffmpeg
            Thread.Sleep(10);
            server.ffmpeg();
        }

    }

}

#region Стабильно работает

#region обработка событий

#region Click on pip

private void preview0button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
{

```

```

        pip(0);
    }

    private void preview1button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
    {
        pip(1);
    }

    private void preview2button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
    {
        pip(2);
    }

    private void preview3button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
    {
        pip(3);
    }

    private void preview4button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
    {
        pip(4);
    }

    private void preview5button3_Click(object sender, EventArgs e)//обработка нажатия
кнопки on pip
    {
        pip(5);
    }

    #endregion

    #region Click on Air

    private void preview0button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
    {
        selected_device = 0;
        change_onair_color();
    }

    private void preview1button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
    {
        selected_device = 1;
        change_onair_color();
    }

    private void preview2button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
    {
        selected_device = 2;
        change_onair_color();
    }

    private void preview3button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
    {
        selected_device = 3;
        change_onair_color();
    }
}

```

```

        private void preview4button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
        {
            selected_device = 4;
            change_onair_color();
        }

        private void preview5button1_Click(object sender, EventArgs e)//обработка нажатия
кнопки on air
        {
            selected_device = 5;
            change_onair_color();
        }

        #endregion

        #region DoubleClick on preview

        private void preview0pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 0;
            change_onair_color();
        }

        private void preview1pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 1;
            change_onair_color();
        }

        private void preview2pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 2;
            change_onair_color();
        }

        private void preview3pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 3;
            change_onair_color();
        }

        private void preview4pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 4;
            change_onair_color();
        }

        private void preview5pictureBox_DoubleClick(object sender, EventArgs e)//
обработка двойного щелчка на окно превью
        {
            selected_device = 5;
            change_onair_color();
        }

        #endregion

        private void Form1_KeyDown(object sender, KeyEventArgs e) //обработка нажатия
клавиш на клавише

```



```

    {
        // recbutton.Text = Convert.ToString(e.KeyValue);
        switch (Convert.ToString(e.KeyValue))
        {
            // numlock
            case ("97"): selected_device = e.KeyValue - 97; change_onair_color();
            break;//1
            case ("98"): selected_device = e.KeyValue - 97; change_onair_color();
            break;
            case ("99"): selected_device = e.KeyValue - 97; change_onair_color();
            break;
            case ("100"): selected_device = e.KeyValue - 97; change_onair_color();
            break;
            case ("101"): selected_device = e.KeyValue - 97; change_onair_color();
            break;
            case ("102"): selected_device = e.KeyValue - 97; change_onair_color();
            break;//6

            //обычные клавиши
            case ("49"): selected_device = e.KeyValue - 49; change_onair_color();
            break;//1
            case ("50"): selected_device = e.KeyValue - 49; change_onair_color();
            break;
            case ("51"): selected_device = e.KeyValue - 49; change_onair_color();
            break;
            case ("52"): selected_device = e.KeyValue - 49; change_onair_color();
            break;
            case ("53"): selected_device = e.KeyValue - 49; change_onair_color();
            break;
            case ("54"): selected_device = e.KeyValue - 49; change_onair_color();
            break;//6

            //qwerty for pic
            case ("81"): pip(0); break;//1
            case ("87"): pip(1); break;
            case ("69"): pip(2); break;
            case ("82"): pip(3); break;
            case ("84"): pip(4); break;
            case ("89"): pip(5); break;//6
        }
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)// обработка
    закрытия формы (завершение всех потоков)
    {
        for (int i = 0; i < preview.Length; i++)
        {
            preview[i].CloseCurrentVideoSource();
            preview[i].stoprec();
        }
    }

    #region pictureboxchanger

    void preview0_pictureboxchanger()
    {
        preview[0].pictureBox.Click -= new
        System.EventHandler(preview0pictureBox_DoubleClick); //отписываемся от старых
        preview[0].pictureBox.DoubleClick -= new
        System.EventHandler(preview0pictureBox_DoubleClick); //отписываемся от старых
        preview[0].pictureBox.Click += new
        System.EventHandler(preview0pictureBox_DoubleClick); //подписываемся на новые
    }

```

```

        preview[0].pictureBox.DoubleClick += new
System.EventHandler(preview0pictureBox_DoubleClick); //подписываемся на новые
    }

    void preview1_pictureboxchanger()
    {
        preview[1].pictureBox.Click -= new
System.EventHandler(preview1pictureBox_DoubleClick);
        preview[1].pictureBox.DoubleClick -= new
System.EventHandler(preview1pictureBox_DoubleClick);
        preview[1].pictureBox.Click += new
System.EventHandler(preview1pictureBox_DoubleClick);
        preview[1].pictureBox.DoubleClick += new
System.EventHandler(preview1pictureBox_DoubleClick);
    }

    void preview2_pictureboxchanger()
    {
        preview[2].pictureBox.Click -= new
System.EventHandler(preview2pictureBox_DoubleClick);
        preview[2].pictureBox.DoubleClick -= new
System.EventHandler(preview2pictureBox_DoubleClick);
        preview[2].pictureBox.Click += new
System.EventHandler(preview2pictureBox_DoubleClick);
        preview[2].pictureBox.DoubleClick += new
System.EventHandler(preview2pictureBox_DoubleClick);
    }

    void preview3_pictureboxchanger()
    {
        preview[3].pictureBox.Click -= new
System.EventHandler(preview3pictureBox_DoubleClick);
        preview[3].pictureBox.DoubleClick -= new
System.EventHandler(preview3pictureBox_DoubleClick);
        preview[3].pictureBox.Click += new
System.EventHandler(preview3pictureBox_DoubleClick);
        preview[3].pictureBox.DoubleClick += new
System.EventHandler(preview3pictureBox_DoubleClick);
    }

    void preview4_pictureboxchanger()
    {
        preview[4].pictureBox.Click -= new
System.EventHandler(preview4pictureBox_DoubleClick);
        preview[4].pictureBox.DoubleClick -= new
System.EventHandler(preview4pictureBox_DoubleClick);
        preview[4].pictureBox.Click += new
System.EventHandler(preview4pictureBox_DoubleClick);
        preview[4].pictureBox.DoubleClick += new
System.EventHandler(preview4pictureBox_DoubleClick);
    }

    void preview5_pictureboxchanger()
    {
        preview[5].pictureBox.Click -= new
System.EventHandler(preview5pictureBox_DoubleClick);
        preview[5].pictureBox.DoubleClick -= new
System.EventHandler(preview5pictureBox_DoubleClick);
        preview[5].pictureBox.Click += new
System.EventHandler(preview5pictureBox_DoubleClick);
        preview[5].pictureBox.DoubleClick += new
System.EventHandler(preview5pictureBox_DoubleClick);
    }

    #endregion

```

```

#endregion

private void change_onair_color()// изменение цвета текущего источника
{
    for (int i = 0; i < number_of_small_panels; i++)
    {
        preview[i].button1.BackColor = button_color;
    }
    preview[selected_device].button1.BackColor = Color.Red;
}

private void change_pip_color()// обнуление цвета клавиш pip
{
    for (int i = 0; i < number_of_small_panels; i++)
    {
        preview[i].button3.BackColor = button_color;
    }
}

private void pip(int new_second_dev)// смена видеовхода картинка в картинке и
смена цвета кнопки pip
{
    if (second_selected_device != new_second_dev)
    {
        second_selected_device = new_second_dev;
        change_pip_color();
        preview[second_selected_device].button3.BackColor = Color.Red;
    }
    else
    {
        second_selected_device = -1;
        change_pip_color();
    }
}

private void Click_event()// подключение к обработке событий
{
    this.KeyDown += new System.Windows.Forms.KeyEventHandler(Form1_KeyDown);//
нажатие клавиш на клавише
    this.FormClosing += Form1_FormClosing; //
закрытие формы

    // click on air
    preview[0].button1.Click += new System.EventHandler(preview0button1_Click);
    preview[1].button1.Click += new System.EventHandler(preview1button1_Click);
    preview[2].button1.Click += new System.EventHandler(preview2button1_Click);
    preview[3].button1.Click += new System.EventHandler(preview3button1_Click);
    preview[4].button1.Click += new System.EventHandler(preview4button1_Click);
    preview[5].button1.Click += new System.EventHandler(preview5button1_Click);

    //click on pip
    preview[0].button3.Click += new System.EventHandler(preview0button3_Click);
    preview[1].button3.Click += new System.EventHandler(preview1button3_Click);
    preview[2].button3.Click += new System.EventHandler(preview2button3_Click);
    preview[3].button3.Click += new System.EventHandler(preview3button3_Click);
    preview[4].button3.Click += new System.EventHandler(preview4button3_Click);
    preview[5].button3.Click += new System.EventHandler(preview5button3_Click);

    // Click to air
    preview[0].pictureBox.Click += new
System.EventHandler(preview0pictureBox_DoubleClick);
    preview[1].pictureBox.Click += new
System.EventHandler(preview1pictureBox_DoubleClick);

```

```

        preview[2].pictureBox.Click += new
System.EventHandler(preview2pictureBox_DoubleClick);
        preview[3].pictureBox.Click += new
System.EventHandler(preview3pictureBox_DoubleClick);
        preview[4].pictureBox.Click += new
System.EventHandler(preview4pictureBox_DoubleClick);
        preview[5].pictureBox.Click += new
System.EventHandler(preview5pictureBox_DoubleClick);

        // DoubleClick to air
        preview[0].pictureBox.DoubleClick += new
System.EventHandler(preview0pictureBox_DoubleClick);
        preview[1].pictureBox.DoubleClick += new
System.EventHandler(preview1pictureBox_DoubleClick);
        preview[2].pictureBox.DoubleClick += new
System.EventHandler(preview2pictureBox_DoubleClick);
        preview[3].pictureBox.DoubleClick += new
System.EventHandler(preview3pictureBox_DoubleClick);
        preview[4].pictureBox.DoubleClick += new
System.EventHandler(preview4pictureBox_DoubleClick);
        preview[5].pictureBox.DoubleClick += new
System.EventHandler(preview5pictureBox_DoubleClick);

        // клавиша настроек
        settingbutton.Click += new System.EventHandler(settingbutton_Click);

        // клавиша записи
        recbutton.Click += new System.EventHandler(recbutton_Click);

        // клавиша нарезки записи видео
        cutbutton.Click += new System.EventHandler(cutbutton_Click);
        // подписка на событие переинициализации
        preview[0].pictureboxchanger += preview0_pictureboxchanger;
        preview[1].pictureboxchanger += preview1_pictureboxchanger;
        preview[2].pictureboxchanger += preview2_pictureboxchanger;
        preview[3].pictureboxchanger += preview3_pictureboxchanger;
        preview[4].pictureboxchanger += preview4_pictureboxchanger;
        preview[5].pictureboxchanger += preview5_pictureboxchanger;
    }

private void createcomponents()// создание объектов с нужными размерами
{
    // Form

    this.Size = new Size(FormWidth, FormHeight);// задаем размеры формы

    // pictureBox

    BigpictureBox.BackColor = System.Drawing.SystemColors.ActiveCaptionText;//
цвет
    BigpictureBox.Location = BigpictureBoxLocation;
    BigpictureBox.Size = BigpictureBoxSize;
    // pictureBox.TabIndex = 0;
    BigpictureBox.SizeMode = PictureBoxSizeMode.Zoom;
    this.Controls.Add(BigpictureBox);

    // panel

    panel.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));

```

```

panel.BackColor = Form1.DefaultBackColor;
panel.Location = panelLocation;
panel.Size = panelSize;
this.Controls.Add(panel);

// smallpanels from panel

for (int i = 0; i < number_of_small_panels - 2; i++)
{
    Panel pan2 = new Panel();
    pan2.Location = smallpanelsLocation[i];
    pan2.BackColor = System.Drawing.Color.Gray;
    pan2.Size = smallpanelsSize[i];
    smallpanels[i] = pan2;
    panel.Controls.Add(smallpanels[i]);
}

// smallpanels from form

for (int i = number_of_small_panels - 2; i < number_of_small_panels; i++)
{
    Panel pan2 = new Panel();
    pan2.Location = smallpanelsLocation[i];
    pan2.BackColor = System.Drawing.Color.Gray;
    pan2.Size = smallpanelsSize[i];
    smallpanels[i] = pan2;
    this.Controls.Add(smallpanels[i]);
}

//rec/stop button

recbutton.Location = recbuttonLocation;
recbutton.Size = recbuttonSize;
recbutton.Text = "Rec/Stop";
recbutton.TabIndex = 1;
this.Controls.Add(recbutton);

//cut button

cutbutton.Location = cutbuttonLocation;
cutbutton.Size = cutbuttonSize;
cutbutton.Text = "Cut";
cutbutton.TabIndex = 3;
this.Controls.Add(cutbutton);

//setting button

settingbutton.Location = settingbuttonLocation;
settingbutton.Size = settingbuttonSize;
settingbutton.Text = "Setup";
settingbutton.TabIndex = 2;
this.Controls.Add(settingbutton);
}

private bool Sets_sizes(int DisplayWidth)// установка параметров размеров
элементов (ширина экрана)
{
    //DisplayWidth = 1920;
    if (DisplayWidth < 1900)
    {
        // переменные размеров

```

```

//form
FormWidth = 1366;// ширина формы
FormHeight = 730;// высота окна формы
this.MinimumSize = new Size(1350, 725);
//BigpictureBox
BigpictureBoxLocation = new System.Drawing.Point(125, 5);// верхний левый
угол главного окна вывода изображения
BigpictureBoxSize = new System.Drawing.Size(767, 420);//размер главного
окна вывода изображения

//panel
panelLocation = new System.Drawing.Point(12, 457);// верхний левый угол
первой панели для маленьких окон
panelSize = new System.Drawing.Size(1326, 231);//размер первой панели для
маленьких окон

//smallpanel 1
smallpanelsLocation[0] = new System.Drawing.Point(3, 3);
smallpanelsSize[0] = new System.Drawing.Size(324, 225);

//smallpanel 2
smallpanelsLocation[1] = new System.Drawing.Point(333, 3);
smallpanelsSize[1] = new System.Drawing.Size(324, 225);

//smallpanel 3
smallpanelsLocation[2] = new System.Drawing.Point(663, 3);
smallpanelsSize[2] = new System.Drawing.Size(324, 225);

//smallpanel 4
smallpanelsLocation[3] = new System.Drawing.Point(993, 3);
smallpanelsSize[3] = new System.Drawing.Size(324, 225);

//smallpanel 5
smallpanelsLocation[4] = new System.Drawing.Point(1005, 230);
smallpanelsSize[4] = new System.Drawing.Size(324, 225);

//smallpanel 6
smallpanelsLocation[5] = new System.Drawing.Point(1005, 2);
smallpanelsSize[5] = new System.Drawing.Size(324, 225);

//smallbutton 1
smallbuttonLocation[0] = new System.Drawing.Point(34, 184);
smallbuttonSize[0] = new System.Drawing.Size(70, 41);

//smallbutton 2
smallbuttonLocation[1] = new System.Drawing.Point(106, 184);
smallbuttonSize[1] = new System.Drawing.Size(70, 41);

//smallbutton 3
smallbuttonLocation[2] = new System.Drawing.Point(178, 184);
smallbuttonSize[2] = new System.Drawing.Size(70, 41);

//smallbutton 4
smallbuttonLocation[3] = new System.Drawing.Point(250, 184);
smallbuttonSize[3] = new System.Drawing.Size(70, 41);

//smallpictureBox
smallpictureBoxLocation = new System.Drawing.Point(1, 3);
smallpictureBoxSize = new System.Drawing.Size(320, 180);

//rec/stop button
recbuttonLocation = new System.Drawing.Point(12, 5);
recbuttonSize = new System.Drawing.Size(107, 124);

```

```

        //cut button
        cutbuttonLocation = new System.Drawing.Point(12, 229);
        cutbuttonSize = new System.Drawing.Size(107, 100);

        //setting button
        settingbuttonLocation = new System.Drawing.Point(15, 153);
        settingbuttonSize = new System.Drawing.Size(107, 47);

        return true;
    }
    else
    {

        //form
        FormWidth = 1920;// ширина формы
        FormHeight = 1080;// высота окна формы
        this.MinimumSize = new Size(1900, 1070);

        //BigpictureBox
        BigpictureBoxLocation = new System.Drawing.Point(175, 7);// верхний левый
угол главного окна вывода изображения
        BigpictureBoxSize = new System.Drawing.Size(1078, 617);//размер главного
окна вывода изображения

        //panel
        panelLocation = new System.Drawing.Point(17, 672);// верхний левый угол
первой панели для маленьких окон
        panelSize = new System.Drawing.Size(1856, 339);//размер первой панели для
маленьких окон

        //smallpanel 1
        smallpanelsLocation[0] = new System.Drawing.Point(4, 4);
        smallpanelsSize[0] = new System.Drawing.Size(453, 320);

        //smallpanel 2
        smallpanelsLocation[1] = new System.Drawing.Point(466, 4);
        smallpanelsSize[1] = new System.Drawing.Size(453, 320);

        //smallpanel 3
        smallpanelsLocation[2] = new System.Drawing.Point(928, 4);
        smallpanelsSize[2] = new System.Drawing.Size(453, 320);

        //smallpanel 4
        smallpanelsLocation[3] = new System.Drawing.Point(1390, 4);
        smallpanelsSize[3] = new System.Drawing.Size(453, 320);

        //smallpanel 5
        smallpanelsLocation[4] = new System.Drawing.Point(1407, 328); ;
        smallpanelsSize[4] = new System.Drawing.Size(453, 320);

        //smallpanel 6
        smallpanelsLocation[5] = new System.Drawing.Point(1407, 3);
        smallpanelsSize[5] = new System.Drawing.Size(453, 320);

        //smallbutton 1
        smallbuttonLocation[0] = new System.Drawing.Point(40, 260);
        smallbuttonSize[0] = new System.Drawing.Size(98, 60);

        //smallbutton 2
        smallbuttonLocation[1] = new System.Drawing.Point(141, 260);
        smallbuttonSize[1] = new System.Drawing.Size(98, 60);

        //smallbutton 3
        smallbuttonLocation[2] = new System.Drawing.Point(243, 260);

```

```

        smallbuttonSize[2] = new System.Drawing.Size(98, 60);

        //smallbutton 4
        smallbuttonLocation[3] = new System.Drawing.Point(343, 260);
        smallbuttonSize[3] = new System.Drawing.Size(98, 60);

        //smallpictureBox
        smallpictureBoxLocation = new System.Drawing.Point(10, 2);
        smallpictureBoxSize = new System.Drawing.Size(430, 255);

        //rec/stop button
        recbuttonLocation = new System.Drawing.Point(12, 5);
        recbuttonSize = new System.Drawing.Size(107, 124);

        //cut button
        cutbuttonLocation = new System.Drawing.Point(12, 229);
        cutbuttonSize = new System.Drawing.Size(107, 100);

        //setting button
        settingbuttonLocation = new System.Drawing.Point(15, 153);
        settingbuttonSize = new System.Drawing.Size(107, 47); ;

        return true;
    }

}

public Form1()
{
    InitializeComponent();
    KeyPreview = true; //переличаем клавиатуру на форму
}

private void recbutton_Click(object sender, EventArgs e)//обработка нажатия
кнопки on rec
{
    if (Record_is_work == false)
    {
        Record_is_work = false;
        string new_way = System.IO.Path.Combine(way_to_folder,
Convert.ToString(DateTime.Now.ToString().Replace(':', '-'))); //генерация нового пути
        System.IO.Directory.CreateDirectory(new_way);
        for (int i = 0; i < preview.Length; i++)
        {
            preview[i].Cutter(new_way);
        }
        Record_is_work = true;
        recbutton.BackColor = Color.Red;
    }
    else
    {
        Record_is_work = false;
        for (int i = 0; i < preview.Length; i++)
        {
            preview[i].stoprec();
        }
        recbutton.BackColor = cutbutton.BackColor;
    }
}

private void cutbutton_Click(object sender, EventArgs e)//обработка нажатия
кнопки on cut
{

```



```

        Record_is_work = false;
        string new_way = System.IO.Path.Combine(way_to_folder,
Convert.ToString(DateTime.Now.ToString().Replace(':', '-')));
        for (int i = 0; i < preview.Length; i++)
        {
            System.IO.Directory.CreateDirectory(new_way);
            preview[i].Cutter(new_way);
        }
        Record_is_work = true;
        recbutton.BackColor = Color.Red;

    }

    #endregion

}
}

```

Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AForge;
using AForge.Video;
using AForge.Video.DirectShow;

namespace VideoStudio
{
    public partial class Form2 : Form
    {
        private VideoCaptureDeviceForm smallform_of_camera; // панель настройки камеры
        private IVideoSource videosource; //переменная потока видео
        public bool all_right; // переменная правильности
        введенных данных(см обработку button1)
        private bool datacopping; // true когда мы
        восстанавливаем данные в нашу форму, для временного открытия

        public Form2()
        {
            all_right = false;
            InitializeComponent();
            //установка параметров по умолчанию
            comboBox1.Visible = false;
            comboBox2.Visible = false;
            comboBox3.Visible = false;
            label1.Visible = false;
            textBox1.Visible = false;
        }
    }
}

```

```

    public Form2(bool checkBox1_Checked, bool checkBox2_Checked, int
index_of_combobox1, int index_of_combobox3, string text_of_combobox2, string textbox,
bool checkbox3)// восстановление значений после пересоздания формы
    {
        all_right = false;
        InitializeComponent();
        datacopping = true;
        try
        {
            if (checkBox1_Checked == true)// обработка видеовхода
            {
                checkBox1.Checked = true;
                comboBox1.Items.Clear();
                comboBox1.Items.Add("Видеокамера");
                comboBox1.Items.Add("Захват монитора ПК");
                comboBox1.Items.Add("Видеофайл");
                comboBox1.Items.Add("JPEGStream");
                comboBox1.Items.Add("MJPEGStream");
                //comboBox1.Items.Add("Другой ПК");
                comboBox1.Visible = true;
                comboBox2.Visible = true;

                if (index_of_combobox1 < 3)
                {
                    comboBox2.Enabled = false;
                }
                else
                {
                    comboBox2.Enabled = true;
                }
                try
                {
                    comboBox1.SelectedIndex = index_of_combobox1;
                }
                catch
                {
                    comboBox1.SelectedIndex = 0;
                }
                comboBox2.Text = text_of_combobox2;
            }
            else
            {
                comboBox1.Visible = false;
                comboBox2.Visible = false;
            }

            if (checkBox2_Checked == true)// работа с аудио выбором
            {
                checkBox2.Checked = true;
                comboBox3.Visible = true;
                label1.Visible = true;
                textBox1.Visible = true;
                textBox1.Text = textbox;
                comboBox3.Items.Clear();

                for (int i = 0; i < NAudio.Wave.WaveIn.DeviceCount; i++)
                {
                    comboBox3.Items.Add(NAudio.Wave.WaveIn.GetCapabilities(i).ProductName);
                }
                try
                {
                    comboBox3.SelectedIndex = index_of_combobox3;
                }
            }
        }
    }

```

```

        catch
        {
            comboBox3.SelectedIndex = 0;
        }
    }
    else
    {
        comboBox3.Visible = false;
        label1.Visible = false;
        textBox1.Visible = false;
    }
}
catch
{
    MessageBox.Show("Ошибка: 706 (ошибка восстановления данных)");
}
checkBox3.Checked = checkBox3;
datacopping = false;
}

#region обработка событий

private void checkBox1_CheckedChanged(object sender, EventArgs e) //
постановка/снятия галочки видео
{
    if (datacopping == false)//
    {
        if (checkBox1.Checked == true)// видео часть
        {
            comboBox1.Items.Clear();
            comboBox1.Items.Add("Видеокамера");
            comboBox1.Items.Add("Захват монитора ПК");
            comboBox1.Items.Add("Видеофайл");
            comboBox1.Items.Add("JPEGStream");
            comboBox1.Items.Add("MJPEGStream");
            // comboBox1.Items.Add("Другой ПК");

            comboBox1.Visible = true;
            checkBox3.Checked = true;
        }
        else
        {
            comboBox1.Items.Clear();
            comboBox1.Visible = false;
            comboBox2.Visible = false;

            if (checkBox2.Checked == false)
            {
                checkBox3.Checked = false;
            }
            else
            {
                checkBox3.Checked = true;
            }
        }
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)// выбор
типа видео устройства
{
    try

```

```

{
    if (datacopping == false)
    {
        if (comboBox1.SelectedIndex == 0)// выбираем камеру
        {
            comboBox2.Enabled = false;// блокируем изменение значения
            comboBox2.Items.Clear(); // очищаем

            smallform_of_camera = new VideoCaptureDeviceForm();// обновляем
            список доступных камер
            if (smallform_of_camera.ShowDialog(this) == DialogResult.OK)//
            открытие и настройка камер
            {
                comboBox2.Items.Add(smallform_of_camera.VideoDeviceMoniker);// выводим значение камеры
                comboBox2.SelectedIndex = 0;
                comboBox2.Visible = true;//открываем путь к источнику
            }
            else //если окно закрыли без сохранения
            {
            }
        }

        if (comboBox1.SelectedIndex == 1)// выбрали монитор
        {
            comboBox2.Enabled = true;
            comboBox2.Items.Clear();
            for (int i=0; i< Screen.AllScreens.Length; i++)
                comboBox2.Items.Add(Screen.AllScreens[i].DeviceName);

            comboBox2.SelectedIndex = 0;
            comboBox2.Visible = true;
        }

        if (comboBox1.SelectedIndex == 2)
        {
            comboBox2.Enabled = false;
            comboBox2.Items.Clear();
            comboBox2.Text = "";
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                comboBox2.Text = openFileDialog1.FileName;
                comboBox2.Visible = true;
            }
            else
            {
            }
        }

        if (comboBox1.SelectedIndex == 3)// url
        {
            comboBox2.Enabled = true;
            comboBox2.Items.Clear();
            comboBox2.Text = "";
            comboBox2.Visible = true;
        }

        if (comboBox1.SelectedIndex == 4)// murl
        {
            comboBox2.Enabled = true;
            comboBox2.Items.Clear();
            comboBox2.Text = "";
            comboBox2.Visible = true;
        }
    }
}

```

```

        if (comboBox1.SelectedIndex == 5)// наша программа
        {
            comboBox2.Enabled = true;
            comboBox2.Items.Clear();
            comboBox2.Text = "";
            comboBox2.Visible = true;
        }
    }
}
catch
{
    MessageBox.Show("Ошибка: 707( проблема с выбором видео источника)");
}
}

private void checkBox2_CheckedChanged(object sender, EventArgs e)//
постановка/снятия галочки аудио
{
    if (datacopping == false)
    {
        if (checkBox2.Checked == true)
        {
            comboBox3.Items.Clear();
            for (int i = 0; i < NAudio.Wave.WaveIn.DeviceCount; i++)
            {
                comboBox3.Items.Add(NAudio.Wave.WaveIn.GetCapabilities(i).ProductName);
            }
            comboBox3.Visible = true;
            comboBox3.SelectedIndex = 0;
            label1.Visible = true;
            textBox1.Visible = true;
            textBox1.Text = "44100";
            checkBox3.Checked = true;
        }
        else
        {
            comboBox3.Items.Clear();
            comboBox3.Visible = false;
            label1.Visible = false;
            textBox1.Visible = false;
            if (checkBox1.Checked == false)
            {
                checkBox3.Checked = false;
            }
            else
            {
                checkBox3.Checked = true;
            }
        }
    }
}

private void button1_Click(object sender, EventArgs e)// проверка на ошибки перед
передачей параметров родительскому объекту
{
    try
    {
        all_right = true;
        if (checkBox1.Checked == true)
        {
            if (comboBox2.Text == null)
            {
                all_right = false;
            }
        }
    }
}

```

```

        MessageBox.Show("Ошибка с форматом Видео источника");
    }
    if (comboBox2.Text == "")
    {
        all_right = false;
        MessageBox.Show("Ошибка с форматом Видео источника");
    }
}
else
{
    videosource = null;
}

if (checkBox2.Checked == true)
{
    if (comboBox3.Text == null)
    {
        all_right = false;
        MessageBox.Show("Ошибка с форматом Аудио источника");
    }
    if (comboBox3.Text == "")
    {
        all_right = false;
        MessageBox.Show("Ошибка с форматом Аудио источника");
    }
    if (textBox1.Text == "")
    {
        all_right = false;
        MessageBox.Show("Ошибка с форматом Аудио источника");
    }
    if (textBox1.Text == null)
    {
        all_right = false;
        MessageBox.Show("Ошибка с форматом Аудио источника");
    }
}
else
{
}
}
catch
{
    MessageBox.Show("Ошибка: 708");
}

if (all_right == true)// если все верно форма закроется
{
    this.Close();
}

}

private void button2_Click(object sender, EventArgs e)// нажатие отмены
{
    all_right = false;// поднимаем флаг, что не все параметры корректны =>
    переинициализация не произойдет
    this.Close();// закрываем форму
}

#endregion

```

```

#region Свойства

#region Свойства передачи выбранных потоков

public NAudio.Wave.WaveIn Audio// свойство для передачи аудиопотока
{
    get
    {
        if (checkBox2.Checked == true)
        {
            NAudio.Wave.WaveIn sourceStream = new NAudio.Wave.WaveIn();
            sourceStream.DeviceNumber =comboBox3.SelectedIndex;
            sourceStream.WaveFormat = new
NAudio.Wave.WaveFormat(Convert.ToInt32(textBox1.Text),
NAudio.Wave.WaveIn.GetCapabilities(index_combobox3).Channels);
            return sourceStream;
        }
        else
        {
            return null;
        }
    }
}

public IVideoSource Video// свойство для передачи видеопотока
{
    get
    {
        try
        {
            if (checkBox1.Checked == true)
            {
                if (comboBox1.SelectedIndex == 0)// поток с камеры
                {
                    videosource = smallform_of_camera.VideoDevice;
                }

                if (comboBox1.SelectedIndex == 1)// поток с монитора
                {
                    videosource = new
ScreenCaptureStream(Screen.AllScreens[comboBox2.SelectedIndex].Bounds, 10);
                }

                if (comboBox1.SelectedIndex == 2)// поток из файла
                {
                    videosource = new FileVideoSource(comboBox2.Text);
                }
                if (comboBox1.SelectedIndex == 3)// поток по url ссылке
                {
                    videosource = new JPEGStream(comboBox2.Text);// создаем новый
поток с ссылкой
                }
                if (comboBox1.SelectedIndex == 4)// поток по url ссылке
                {
                    videosource = new MJPEGStream(comboBox2.Text);// создаем
новый поток с ссылкой
                }
                if (comboBox1.SelectedIndex == 5)// если мы выбрали дорогой
компьютер с нашей программой,
// обычный видео поток нам не подходит
                {
                    videosource = null;
                }
            }
        }
    }
}

```

```

        }

        return videosource;
    }
    else
    {
        return null;
    }
}
catch
{
    return null;
}
}
}

#endregion

//свойства настроек аудио и видео для сохранения
//информации о состоянии формы в родительский объект
#region Свойства настроек видео

public bool checkBox1_was_Checked // выбран ли видео вход
{
    get
    {
        return checkBox1.Checked;
    }
}

public int index_combobox1// тип выбранного видео устройства
{
    get
    {
        return comboBox1.SelectedIndex;
    }
}

public string text_of_combobox2// моникер видео устройства
{
    get
    {
        return comboBox2.Text;
    }
}

#endregion

#region Свойства настроек звука

public bool checkBox2_was_Checked // выбран ли аудио вход
{
    get
    {
        return checkBox2.Checked;
    }
}

public int index_combobox3// выбранное аудио устройство
{
    get
    {
        return comboBox3.SelectedIndex;
    }
}

```



```

    }

    public string text_of_textbox// частота дискретизации звука
    {
        get
        {
            return textBox1.Text;
        }
    }

    #endregion

    #region Свойства настроек записи

    public bool CheckBox3_checked// необходимо ли производить запись
    {
        get
        {
            return checkBox3.Checked;
        }
    }

    #endregion

    #endregion
}
}

```

Form3.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.IO;
using System.Windows.Forms;

namespace VideoStudio
{
    public partial class Form3 : Form
    {
        private bool all_good = false;
        private string param1, param2, param7, param4, param6;
        public string mainparam;

        public Form3()//основные окна настройки
        {
            InitializeComponent();
            textBox4.Visible = true;
            textBox5.Visible = true;
            checkBox3.Checked = false;
            textBox7.Text = "-b:v 2000k -vcodec mpeg2video -f mpegts";
            textBox1_TextChanged(null, null);//принудительный вызов обработки событий
        }
    }
}

```

```

        textBox2_TextChanged(null, null);
        textBox7_TextChanged(null, null);
        textBox4_TextChanged(null, null);
        textBox6_TextChanged(null, null);
        textBox7_TextChanged(null, null);
        checkBox3_CheckedChanged(null, null);
        command_update();//обновление строки для ffmpeg
    }

    private void Form3_Load(object sender, EventArgs e)
    {
        textBox3.Enabled = false;
        command_update();//обновление строки для ffmpeg
    }

    private void button1_Click(object sender, EventArgs e)// открытие диалогового
окна выбора папки
    {
        if (folderBrowserDialog1.ShowDialog(this) == DialogResult.OK)
        {
            textBox3.Text = folderBrowserDialog1.SelectedPath;
        }
    }

    private void checkBox3_CheckedChanged(object sender, EventArgs e)// выбор онлайн
трансляции
    {
        if (checkBox3.Checked == true)
        {
            textBox4.Visible = true;
            textBox5.Visible = true;
            label4.Visible = true;

            label5.Visible = true;
            label6.Visible = true;
            textBox6.Visible = true;
            textBox7.Visible = true;
            label7.Visible = true;
            textBox4.Text = "127.0.0.1";
            textBox6.Text = "5000";
        }
        else
        {
            textBox4.Visible = false;
            textBox5.Visible = false;
            label4.Visible = false;
            label5.Visible = false;
            label6.Visible = false;
            textBox6.Visible = false;
            textBox7.Visible = false;
            label7.Visible = false;
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)//ширина
    {
        try
        {
            if (textBox1.Text != "")
            {
                int a = int.Parse(textBox1.Text);
            }
        }
    }

```

```

    }
    catch
    {
        MessageBox.Show("Некорректные данные");
    }

    if (textBox1.Text != "")
    {
        param1 = " -video_size " + textBox1.Text;
    }
    else
    {
        param1 = "";
    }
    command_update();
}

private void textBox2_TextChanged(object sender, EventArgs e)//высота
{
    try
    {
        if (textBox2.Text != "")
        {
            int a = int.Parse(textBox2.Text);

            if (textBox2.Text != "")
            {
                param2 = "x" + textBox2.Text;
            }
            else
            {
                param2 = "";
            }
            command_update();
        }
        catch
        {
            MessageBox.Show("Некорректные данные");
        }
    }

    private void button2_Click(object sender, EventArgs e)//сохранить
    {
        // сохранить
        all_good = true;

        //textbox1
        try
        {
            if (textBox1.Text != "")
            {
                int a = int.Parse(textBox1.Text);
            }
        }
        catch
        {
            all_good = false;
            MessageBox.Show("Некорректные данные в поле: Ширина выходной картинки");
        }

        //textbox2
        try
        {

```

```

        if (textBox2.Text != "")
        {
            int a = int.Parse(textBox2.Text);
        }
    }
    catch
    {
        all_good = false;
        MessageBox.Show("Некорректные данные в поле: Высота выходной картинки");
    }

    // проверка пути
    if (textBox3.Text != "")
    {
        if (Directory.Exists(textBox3.Text) == true)
        {
        }
        else
        {
            all_good = false;
            MessageBox.Show("Заданного пути не существует");
        }
    }

    try
    {
        int a = int.Parse(textBox6.Text);
    }
    catch
    {
        all_good = false;
        MessageBox.Show("Некорректные данные в поле: Порт");
    }

    if(all_good==true)//закрытие
    {
        this.Close();
    }

}

private void button3_Click(object sender, EventArgs e)//отмена
{
    //отмена
    all_good = false;
    this.Close();
}

#region Свойства для сохранения настроек

public bool allgood
{
    get
    {
        return all_good;
    }
}

public int Widthofpicture
{
    get

```

```

        {
            return Convert.ToInt32( textBox1.Text);
        }
        set
        {
            textBox1.Text =Convert.ToString( value);
        }
    }

    public int Heightofpicture
    {
        get
        {
            return Convert.ToInt32( textBox2.Text);
        }
        set
        {
            textBox2.Text = Convert.ToString(value);
        }
    }

    public bool rec_main
    {
        get
        {
            return checkBox1.Checked;
        }
        set
        {
            checkBox1.Checked = value;
        }
    }

    public bool rec_all
    {
        get
        {
            return checkBox2.Checked;
        }
        set
        {
            checkBox2.Checked = value;
        }
    }

    public string folder
    {
        get
        {
            return textBox3.Text;
        }
        set
        {
            textBox3.Text = value;
        }
    }

    public bool online
    {
        get
        {
            return checkBox3.Checked;
        }
        set
        {

```

```

        checkBox3.Checked = value;
    }
}

public string ipaddress
{
    get
    {
        return textBox4.Text;
    }
    set
    {
        textBox4.Text = value;
    }
}

public int port
{
    get
    {if(textBox6.Text=="")
    {
        return 5000;
    }
    return Convert.ToInt32(textBox6.Text);
    }
    set
    {
        textBox6.Text = Convert.ToString(value);
    }
}

public string more_settings
{
    get
    {
        return textBox7.Text;
    }
    set
    {
        textBox7.Text = value;
    }
}

public string mainparams
{
    get
    {
        return textBox5.Text;
    }
}
#endregion

private void textBox6_TextChanged(object sender, EventArgs e)
{
    try
    {
        int a = int.Parse(textBox6.Text);

    }
    catch
    {
        MessageBox.Show("Некорректные данные в поле: Порт");
    }
}

```

```

        if (textBox6.Text != "")
        {
            param6 = ":" + textBox6.Text;
        }
        else
        {
            param6 = "";
        }
        command_update();
    }

    private void textBox4_TextChanged(object sender, EventArgs e)
    {
        if (textBox4.Text != "")
        {
            param4 = " udp:" + textBox4.Text;
        }
        else
        {
            param4 = "";
        }
        command_update();
    }

    private void textBox7_TextChanged(object sender, EventArgs e)
    {
        if (textBox7.Text != "")
        {
            param7 = " " + textBox7.Text;
        }
        else
        {
            param7 = "";
        }
        command_update();
    }
    public void command_update()
    {
        textBox5.Text = param1 + param2 + param7 + param4 + param6;
    }
}
}

```

smallwindow.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using AForge;
using AForge.Video;
using AForge.Video.FFMPEG;
using AForge.Video.DirectShow;

```

```

using System.Threading;
using NAudio.Wave;
using NAudio.CoreAudioApi;

namespace VideoStudio
{
    class smallwindow
    {

        #region Объявление переменных

        #region объекты формы

            public System.Windows.Forms.PictureBox pictureBox = new
System.Windows.Forms.PictureBox(); // окно вывода картинки
            public System.Windows.Forms.Button button1 = new System.Windows.Forms.Button();
// кнопка on air
            private System.Windows.Forms.Button button2 = new System.Windows.Forms.Button();
// кнопка работы со звуком
            public System.Windows.Forms.Button button3 = new System.Windows.Forms.Button();
// кнопка картинка в картинке
            private System.Windows.Forms.Button button4 = new System.Windows.Forms.Button();
// кнопка настройки
            private System.Windows.Forms.Label label = new System.Windows.Forms.Label();
// надпись с номером потока
            private Panel outpanel;
// панель на которой настроятся 4 кнопки и pictureBox
            private System.Drawing.Point[] smallbuttonLocation;
// массив расположения кнопок маленьких панелей
            private System.Drawing.Size[] smallbuttonSize;
// массив размера кнопок маленьких панелей
            private System.Drawing.Point smallpictureBoxLocation;
// расположение pictureBox
            private System.Drawing.Size smallpictureBoxSize;
// размер pictureBox
            private int id;
//номер потока/камеры

        #endregion

        #region данные окна маленьких настроек

            private Form2 forms_of_selection;
// вспомогательная форма для настройка водных параметров
            private bool checkBox1_Checked;
// выбрано ли использование видео устройства
            private bool checkBox2_Checked;
// выбрано ли использование аудио устройства
            private int index_of_combobox1=0;
// номер выбранного типа видеоустройства
            private int index_of_combobox3;
// номер выбранного аудио устройства
            private string text_of_combobox2;
// выбранное видео устройство
            private string textbox;
// частота дискретизации
            private bool form_had_been_opened;
// флаг на использование ранее сохраненных данных true- испльзовать
            private bool checkbox3;

        # endregion
    }
}

```



```

        private IVideoSource videosource;
// видео поток
        private IVideoSource cashvideosource;
// временная переменная для смены видеопотоков
        private NAudio.Wave.WaveIn AudiosourceStream;
// аудио поток
        private NAudio.Wave.WaveIn cashAudiosourceStream;
// временная переменная для смены аудиопотоков
        private NAudio.Wave.DirectSoundOut waveOut = null;
// объект воспроизведения звука
        private NAudio.Wave.WaveFileWriter waveWriter = null;
// объект записи в файл
        private byte[] Audiosourcebuffer;
// буфер аудио
        private int AudiosourceBytesRecorded;
// количество полученного звука
        private int Audiosourceoffset;
//
        private string AudioOutputfile;
// переменная для хранения пути к файлу записи аудио
        private string VideoOutputfile;
// переменная для хранения пути к файлу записи видео
        private Bitmap Videosourcecach=null;
// переменная для хранения текущего кадра
        private tcpclient client= new tcpclient();
// подключение к другому ПК
        private VideoFileWriter Videowriter1;
// запись видео в файл
        Bitmap img;
// переменная хранит заставку и выдает ее на выход если нет видео входа

        private bool Isrecordworkingnow=false;
// переменная хранит значение работает ли сейчас кнопка записи (по умолчанию false)

        #endregion

        public delegate void MethodContainer();

        //Событие OnCount с типом делегата MethodContainer.
        public event MethodContainer pictureboxchanger;

        public void CloseCurrentVideoSource() //завершаем все потоки, если он не
остановлены
        {
            try
            {
                client.flag = true;
            }
            catch
            {
            }

            try
            {
                waveOut.Stop();
            }
            catch
            {
            }

            try

```

```

        {
            waveWriter.Close();
        }
    catch
    {
    }

    try
    {
        videosource.SignalToStop();
    }
    catch
    {
    }

    try
    {
        videosource.WaitForStop();
    }
    catch
    {
    }

    try
    {
        videosource.Stop();
    }
    catch
    {
    }

    try
    {
        AudiosourceStream.StopRecording();
    }
    catch
    {
    }

    try
    {
        AudiosourceStream.Dispose();
    }
    catch
    {
    }

    try
    {
        client.Closing();
    }
    catch
    {
    }
}

private void button2_Click(object sender, EventArgs e)

```

```

{
    //MessageBox.Show("2");
}

#region запуск захвата аудио и видео

private void OpenAudioSource()// запуск захвата аудио
{
    AudiosourceStream.StartRecording();// получение потока

    AudiosourceStream.DataAvailable += new
EventHandler<NAudio.Wave.WaveInEventArgs>(sourceStream_DataAvailable); // подписываемся
на событие получения звука
}

private void OpenVideoSource()// запуск захвата видео
{
    CloseCurrentVideoSource(); // останавливаем все предыдущие действия
    videosource.NewFrame += new NewFrameEventHandler(player_NewFrame); //
подписываемся на событие прихода нового видео кадра
    videosource.Start();

}

#endregion

void CreateBitmap() //создание битмапа если нет изображения в папке
{
    System.Drawing.Bitmap flag = new System.Drawing.Bitmap(720, 720);
    for (int x = 0; x < flag.Height; ++x)
        for (int y = 0; y < flag.Width; ++y)
            flag.SetPixel(x, y, Color.Black);

    img= flag;
}

public smallwindow(Panel outpanel, System.Drawing.Point[] smallbuttonLocation,
System.Drawing.Size[] smallbuttonSize, System.Drawing.Point smallpictureBoxLocation,
System.Drawing.Size smallpictureBoxSize, int id)
{
    try
    {
        //попытка получить картинку заставку
        System.IO.FileStream fs = new System.IO.FileStream("foto.jpg",
System.IO.FileMode.Open);
        img = (Bitmap)System.Drawing.Image.FromStream(fs);
        fs.Close();
    }
    catch
    {
        CreateBitmap();// создаем черный битмап
    }
    // передача параметров конструктора в поля
    this.outpanel = outpanel;
    this.smallbuttonLocation = smallbuttonLocation;
    this.smallbuttonSize = smallbuttonSize;
    this.smallpictureBoxLocation = smallpictureBoxLocation;
    this.smallpictureBoxSize = smallpictureBoxSize;
    this.id = id;
    videosource = new VideoCaptureDevice();
    AudiosourceStream = new NAudio.Wave.WaveIn();
    try
    {

```

```

        drawing();// отрисовка 4 кнопок, номера камеры, pictureBox
        button2.Enabled = false;

    }
    catch
    {
        MessageBox.Show("Ошибка: 703( не возможно создать элементы интерфейса");
    }
    // подписываемся на события кликов мышью (первую и третью клавишу
    обрабатывает form1)
    button2.Click += new System.EventHandler(button2_Click);
    button4.Click += new System.EventHandler(button4_Click);
}

    public smallwindow()
    {

    }

    //public void update_information()//обновление полученной информации если выбран
    удаленный ПК
    //{
    //    if (index_of_combobox1 == 5)
    //    {
    //        Videosourcecach = client.Image;
    //        pictureBox.Image = client.Image;
    //        Audiosourcebuffer = client.bytedata;
    //        Audiosourceoffset = client.return_offset;
    //        AudiosourceBytesRecorded = client.return_recv;
    //    }
    //}

    private void forms_of_selection_FormClosing(object sender, FormClosingEventArgs
e) // обработка закрытия формы настройки
    {
        if (forms_of_selection.all_right == true)                //(all_right==true если
пользователь нажал Ok и все параметры действительны)
        {
            copy_data_from_form2(); // сохранение информации из form2

            forms_of_selection.Dispose();//уничтожение form2

            if(checkBox2_Checked==true)
            {
                change_audio();                // смена аудио источника
            }

            if (index_of_combobox1 != 5)// если не выбран показ с другого ПК
            {
                bool flag = checkBox3;        //блокировка записи
                stoprec();                    // остановка записи
                change_video();                // смена видео источника

                if (form_had_been_opened == true)
                {
                    if (flag == true)
                    {
                        MessageBox.Show("Запись с этого источника начнется только
после команды CUT или REC");
                    }
                }
            }
        }
    }
}

```

```

        form_had_been_opened = true; // поднимаем флаг, что мы уже открывали
форму
    }
    else
    {
        try
        {
            if (videosource.IsRunning == true) //останавливаем видео поток
            {
                videosource.SignalToStop();
                videosource.WaitForStop();
                videosource.Stop();
            }
            try
            {
                AudiosourceStream.StopRecording();
                AudiosourceStream.Dispose();
            }
            catch
            {
            }
        }
        catch
        {
            MessageBox.Show("Ошибка остановки захвата с источников");
        }

        ////////// запуск получения картинки с друго компа

    }
    forms_of_selection.all_right = false; // меняем значение на
false, чтобы при дальнейшем закрытии формы, мы не вернулись сюда
    forms_of_selection.Close(); // закрываем form2
}

private void copy_data_from_form2() // сохранение информации из form2
{
    try
    {
        cashvideosource = forms_of_selection.Video; // устанавливаем
значение будущего видео потока
        cashAudiosourceStream = forms_of_selection.Audio; // устанавливаем
значение будущего аудио потока

        // сохранение информации для повторной инициализации формы
        checkBox1_Checked = forms_of_selection.checkBox1_was_Checked;
        checkBox2_Checked = forms_of_selection.checkBox2_was_Checked;
        index_of_combobox1 = forms_of_selection.index_combobox1;
        index_of_combobox3 = forms_of_selection.index_combobox3;
        text_of_combobox2 = forms_of_selection.text_of_combobox2;
        textbox = forms_of_selection.text_of_textbox;
        checkBox3 = forms_of_selection.CheckBox3_checked;
    }
    catch
    {
    }
}

private void change_video()
{
    try
    {

```

```

        if (checkBox1_Checked == true)                // запуск процедуры захвата
видео с нового потока
        {
            if (videosource.IsRunning == true) //останавливаем видео поток
            {
                videosource.SignalToStop();
                videosource.WaitForStop();
                videosource.Stop();
            }
            pictureBox.Dispose();
            pictureBox = null;
            pictureBox = new PictureBox();
            pictureBox.BackColor = System.Drawing.Color.Black;
            pictureBox.Location = smallpictureBoxLocation;
            pictureBox.Size = smallpictureBoxSize;
            // pictureBox.BorderStyle = BorderStyle.Fixed3D;
            pictureBox.SizeMode = PictureBoxSizeMode.Zoom;
            outpanel.Controls.Add(pictureBox);
            pictureBoxchanger();
            videosource = cashvideosource; //меняем девайс
            OpenVideoSource();// запускаем
        }
        else
        {
            if (videosource.IsRunning == true) //останавливаем видео поток
            {
                videosource.SignalToStop();
                videosource.WaitForStop();
                videosource.Stop();
            }
        }
    }
    catch
    {
        MessageBox.Show("Ошибка: 705 (Ошибка смены видеовхода)");
    }
} // смена видеовхода

private void change_audio()
{
    try
    {
        if (checkBox2_Checked == true)                // запуск процедуры захвата с
нового потока аудио
        {
            AudiosourceStream.StopRecording();
            AudiosourceStream.Dispose();
            AudiosourceStream = cashAudiosourceStream;
            OpenAudioSource();
        }
        else
        {
            AudiosourceStream.StopRecording();
            AudiosourceStream.Dispose();
        }
    }
    catch
    {
        MessageBox.Show("Ошибка: 704 (Ошибка смены аудиовхода)");
    }
} // смена аудиовхода

```

```

        private void player_NewFrame(object sender, NewFrameEventArgs eventArgs)//
событие связанное с появлением нового изображения в потоке
        {
            try
            {
                pictureBox.Image = (Bitmap)eventArgs.Frame.Clone(); //вывод в превью

                Videosourcecach = (Bitmap)eventArgs.Frame.Clone();

                GC.Collect(); // сбор мусора, т.к clone засоряет оперативную память
            }
            catch
            {
            }
        }

        private void sourceStream_DataAvailable(object sender,
NAudio.Wave.WaveInEventArgs e)// захват аудио потока и сохранения в буфера
        {
            AudiosourceBytesRecorded = e.BytesRecorded;
            Audiosourcebuffer = e.Buffer;
            Audiosourceoffset = 0;

            if(Isrecordworkingnow == true)
            {
                audiowriter();
            }
        }

        # region Свойства

        # region свойства аудио

        public byte[] Audiosourcebuffer// свойство возвращающее аудио буфер
        {
            get
            {
                return Audiosourcebuffer;
            }
        }

        public int Audio_source_Bytes_Recorded// свойство возвращающее количество
элементов аудио буфера
        {
            get
            {
                return AudiosourceBytesRecorded;
            }
        }

        public int Audio_source_offset// свойство возвращающее количество элементов аудио
        {
            get
            {
                return Audiosourceoffset;
            }
        }

        #endregion

        public Bitmap Video_source_cach// свойство возвращающее текущий кадр видео
        {
            get
            {

```

```

        if(Videosourcecach!=null)
        {
            return Videosourcecach;
        }
        else
        {
            return img;
        }
    }
}

#endregion

#region Запись

public void audiowriter() // запись звука
{
    if (checkbox3 == true)
    {
        if (waveWriter != null)
        {
            if (Audiosourcebuffer!=null)
                waveWriter.Write(Audiosourcebuffer, Audiosourceoffset,
AudiosourceBytesRecorded);
            waveWriter.Flush();
        }
    }
}

public void videowriter() // запись видео
{
    if (checkbox3 == true)
    {
        if (Videowriter1 != null)
        {
            if (Videowriter1.IsOpen != false)
            {
                Videowriter1.WriteVideoFrame(Videosourcecach);
            }
        }
    }
}

public void Cutter(string new_folder)// Перезалка записи
{
    stoprec();
    startrec(new_folder);
}

public void startrec(string new_folder)//запуск записи
{
    Isrecordworkingnow = true;
    if (checkbox3 == true)
    {
        if (checkBox1_Checked == true)
        {
            VideoOutputfile = System.IO.Path.Combine(new_folder, "camera" + id +
".avi");// генерация имени файла

            if (Videowriter1 != null)
                Videowriter1.Close();

```



```

        Videowriter1 = new VideoFileWriter();

        try
        {
            Videowriter1.Open(VideoOutputfile, Videosourcecach.Width,
Videosourcecach.Height, 30, VideoCodec.MPEG2, 45000000);
        }
        catch
        {
            MessageBox.Show("Ошибка создания файла для записи видео" + id);
        }

        Isrecordworkingnow = true;
    }

    if (checkBox2_Checked == true)
    {
        AudioOutputfile = System.IO.Path.Combine(new_folder, "audio" + id +
".wav"); // генерация имени файла
        try
        {
            if (waveWriter != null)
                waveWriter.Close();
            waveWriter = null;
            waveWriter = new NAudio.Wave.WaveFileWriter(AudioOutputfile,
AudiosourceStream.WaveFormat);
        }
        catch
        {
            MessageBox.Show("Ошибка создания файла для записи аудио" + id);
        }
    }
}

public void stoprec()//остановка записи
{
    Isrecordworkingnow = false;
    try
    {
        if (waveWriter != null)
            waveWriter.Close();
        waveWriter = null;
        if (Videowriter1 != null)
            Videowriter1.Close();
    }
    catch
    {
        MessageBox.Show("Ошибка остановки записи потоков " + id);
    }
}

#endregion

private void button4_Click(object sender, EventArgs e)// нажатие клавиши настроек
{
    try
    {
        if (form_had_been_opened == true)// если ранее уже настраивали
        {
            forms_of_selection = new Form2(checkBox1_Checked, checkBox2_Checked,
index_of_combobox1, index_of_combobox3, text_of_combobox2, textbox, checkbox3);//
инициализируем новую форму с старыми параметрами

```

```

        forms_of_selection.FormClosing += forms_of_selection_FormClosing;
// подписываемся на событие ее закрытия
        forms_of_selection.Show();
// открываем форму
    }
    else
    {
        forms_of_selection = new Form2();
// инициализируем новую форму
        forms_of_selection.FormClosing += forms_of_selection_FormClosing;
// подписываемся на событие ее закрытия
        forms_of_selection.Show();
// открываем форму
    }
    GC.Collect();// сборщик мусора
}
catch
{
    MessageBox.Show("Ошибка 702(открытие окна настроек источника)");
}
}

private void drawing() // добавление и отрисовка элементов маленькой формы
{
    //smallbutton 1
    button1.Location = smallbuttonLocation[0];
    button1.Size = smallbuttonSize[0];
    button1.UseVisualStyleBackColor = true;
    button1.Text = "On air";
    outpanel.Controls.Add(button1);

    //smallbutton 2
    button2.Location = smallbuttonLocation[1];
    button2.Size = smallbuttonSize[1];
    button2.UseVisualStyleBackColor = true;
    button2.Text = "Sound";
    outpanel.Controls.Add(button2);

    //smallbutton 3
    button3.Location = smallbuttonLocation[2];
    button3.Size = smallbuttonSize[2];
    button3.Text = "Pic-in-Pic";
    button3.UseVisualStyleBackColor = true;
    outpanel.Controls.Add(button3);

    //smallbutton 4
    button4.Location = smallbuttonLocation[3];
    button4.Size = smallbuttonSize[3];
    button4.Text = "Setup";
    button4.UseVisualStyleBackColor = true;
    outpanel.Controls.Add(button4);

    //smallpictureBox
    pictureBox.BackColor = System.Drawing.Color.Black;
    pictureBox.Location = smallpictureBoxLocation;
    pictureBox.Size = smallpictureBoxSize;
    // pictureBox.BorderStyle = BorderStyle.Fixed3D;
    pictureBox.SizeMode = PictureBoxSizeMode.Zoom;
    outpanel.Controls.Add(pictureBox);

    // label
    if(System.Windows.Forms.Screen.PrimaryScreen.Bounds.Width<1810)
    {
        label.Location = new System.Drawing.Point(2, 190);
    }
}

```

```

    }
    else
    {
        label.Location = new System.Drawing.Point(4, 290);
    }
    label.Font = new Font("Arial", 12, FontStyle.Bold);
    switch(id)
    {
        case (0): label.Text = Convert.ToString("1 q"); break;
        case (1): label.Text = Convert.ToString("2 w"); break;
        case (2): label.Text = Convert.ToString("3 e"); break;
        case (3): label.Text = Convert.ToString("4 r"); break;
        case (4): label.Text = Convert.ToString("5 t"); break;
        case (5): label.Text = Convert.ToString("6 y"); break;
    }

    outpanel.Controls.Add(label);
}
}
}

```

tcpserver.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Net.Sockets;
using System.IO;
using System.Drawing.Imaging;
using System.Threading;
using System.Net;
using AForge.Video;
using AForge.Video.DirectShow;
using NAudio.Wave;
using NAudio.CoreAudioApi;
using System.Diagnostics;
using System.Reflection;

namespace VideoStudio
{
    class tcpserver
    {
        private Socket connection;
        private NetworkStream socketStream;
        private BinaryWriter writer;
        TcpListener listener;
    }
}

```

```

IPAddress local;
Process process_ffmpeg;
private string command;
int port;

bool flag = false; // флаг работы функции sender

public tcpserver()
{
}

public tcpserver(int port, string command)
{
    this.command = command;
    this.port = port;

    local = IPAddress.Any;
    listener = new TcpListener(local, port);
    Thread client;
    try
    {
        listener.Start();
        client = new Thread(connector); // запускаем поток в котором мы ожидаем
подключения
        client.Start();
    }
    catch
    {
        MessageBox.Show("Ошибка создания подключения");
        try
        {
            listener.Stop();
            flag = false;
        }
        catch
        {
        }
    }
}

private void connector()
{
    while (connection == null)
    {
        // Принимаем новых клиентов
        connection = listener.AcceptSocket();
    }
    socketStream = new NetworkStream(connection);
    writer = new BinaryWriter(socketStream);
    flag = true;
}

public void ffmpeg()
{
    try
    {
        process_ffmpeg = new Process();
    }
}

```

```

        //ProcessStartInfo startInfo = new
ProcessStartInfo(@"ffmpeg\bin\ffplay.exe", "-i tcp:127.0.0.1:" +
int.Parse(textBox1.Text));

        string arg = "-i tcp:127.0.0.1:" + port + " " + command;// формируем
команду
        ProcessStartInfo startInfo = new
ProcessStartInfo(@"ffmpeg\bin\ffmpeg.exe", arg);
        startInfo.CreateNoWindow = true;// показ/непоказ консоли
        startInfo.UseShellExecute = false;
        //startInfo.RedirectStandardError = true;
        //startInfo.RedirectStandardOutput = true;
        process_ffmpeg.StartInfo = startInfo;
        process_ffmpeg.Start();//запуск

        //StreamReader myStreamReader = process_ffmpeg.StandardOutput;
        //// Read the standard error of net.exe and write it on to console.
        //MessageBox.Show(myStreamReader.ReadLine());

        flag = true;
    }
    catch
    {
        MessageBox.Show("Ошибка запуска ffmpeg");
        try
        {
            listener.Stop();
            flag = false;
        }
        catch
        {
        }
    }
}

public void sender(Bitmap image)
{
    if (flag == true)
    {
        MemoryStream ms = new MemoryStream();
        image.Save(ms, ImageFormat.Bmp);

        byte[] arrImage = ms.GetBuffer();

        ms.Close();
        ms.Dispose();
        ms = null;
        try// если не возможно отправить данные абоненту, то мы все выключаем
        {

            writer.Write(arrImage);
            writer.Flush();
        }
        catch
        {
            closing();
        }
        GC.Collect();
    }
}

public void closing() //остановка всего
{
    try

```

```
{
    flag = false;
    writer.Flush();
    writer.Close();
    socketStream.Close();
    connection.Close();
    listener.Stop();
    process_ffmpeg.Kill();
    MessageBox.Show("Поток онлайн трансляции остановлен");
}
catch
{
}

}

}
```