

Urządzenia peryferyjne laboratorium	
Temat:	Kamera
Data zajęć:	22.12.2021
Grupa zajęciowa:	TN środa 11:15
Członkowie zespołu:	Aleksandra Ciężka 252701 Mateusz Chalik 252735
Grupa	C
Prowadzący:	dr inż. Tomasz Walkowiak

## 1. Wstęp

Celem ćwiczenia było zapoznanie się z obsługą kamery oraz zapisywanie obrazów i filmów. Dodatkowo należało rozbudować program o generowanie strony html z obrazem z kamery.

## 2. Realizacja zadania

### 2.1 Teoria

Kamera cyfrowa to urządzeniem, które przekształca filmowany obraz do postaci sygnału cyfrowego, łatwego do obrobienia i przesyłania. Kamera może transmitować obraz statycznie, po przez odświeżanie co pewien czas pojedynczych obrazów, lub transmitować w sposób ciągły.

Przykładowe metody obsługi kamery:

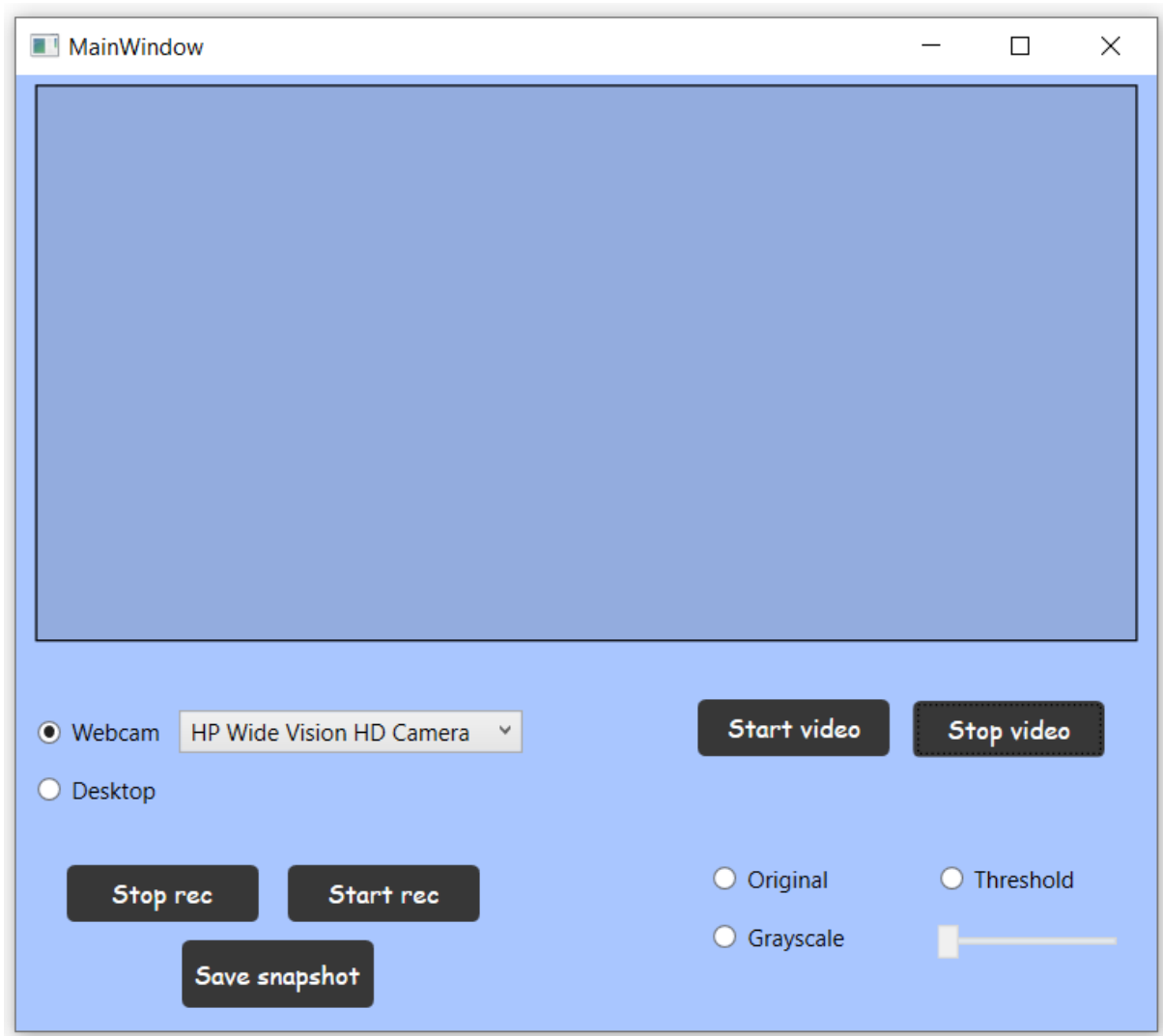
- avicap32.dll
- TWAIN
- WIA 1.0
- WIA 2.0

Przy programowaniu kamery możemy wpływać na jej opcje, takie jak rozdzielczość obrazu, jego nasycenie, kontrast, zoom. Na oryginalny obraz kamery można nałożyć filtry, np. grayscale.

Obraz z kamery może być także zapisywany m.in. jako JPEG czy PNG, lub w postaci filmu. Ułatwiają to gotowe funkcje z odpowiednich bibliotek.

## 2.2 Własna aplikacja okienkowa

Kolejnym etapem zadnienia było napisanie aplikacji okienkowej służącej do obsługi kamery.



*Rysunek 1 Wgląd graficznego interfejsu użytkownika*

W aplikacji zaimplementowano następujące funkcje:

### 1. Zapisanie obrazu jako bitmapy

```
public static BitmapImage ToBitmapImage(this Bitmap bitmap)
{
    BitmapImage bi = new BitmapImage();
    bi.BeginInit();
    MemoryStream ms = new MemoryStream();
    bitmap.Save(ms, ImageFormat.Bmp);
    ms.Seek(0, SeekOrigin.Begin);
    bi.StreamSource = ms;
    bi.EndInit();
    return bi;
}
```

### 2. Pobranie dostępnych urządzeń

```
private void GetVideoDevices()
{
    var devices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
    foreach (FilterInfo device in devices)
    {
        VideoDevices.Add(device);
    }
    if (VideoDevices.Any())
    {
        CurrentDevice = VideoDevices[0];
    }
    else
    {
        MessageBox.Show("No webcam found");
    }
}
```

### 3. Włączenie kamery

```
private void StartCamera()
{
    if (IsDesktopSource)
    {
        var rectangle = new Rectangle();
        foreach (var screen in System.Windows.Forms.Screen.AllScreens)
        {
            rectangle = Rectangle.Union(rectangle, screen.Bounds);
        }
        _videoSource = new ScreenCaptureStream(rectangle);
        _videoSource.NewFrame += video_NewFrame;
        _videoSource.Start();
    }
    else if (IsWebcamSource)
    {
        if (CurrentDevice != null)
        {
            _videoSource = new VideoCaptureDevice(CurrentDevice.MonikerString);
            _videoSource.NewFrame += video_NewFrame;
            _videoSource.Start();
        }
        else
        {
            MessageBox.Show("Current device can't be null");
        }
    }
}
```

### 4. Wyłączenie kamery

```
private void StopCamera()
{
    if (_videoSource != null && _videoSource.IsRunning)
    {
        _videoSource.SignalToStop();
        _videoSource.NewFrame -= video_NewFrame;
    }
    Image = null;
}
```

## 5. Nowa klatka

```
private void video_NewFrame(object sender, NewFrameEventArgs eventArgs)
{
    try
    {
        if (_recording)
        {
            using (var bitmap = (Bitmap)eventArgs.Frame.Clone())
            {
                _writer.WriteVideoFrame(bitmap);

                if (_firstFrameTime != null)
                {
                    _writer.WriteVideoFrame(bitmap, DateTime.Now - _firstFrameTime.Value);
                }
                else
                {
                    _writer.WriteVideoFrame(bitmap);
                    _firstFrameTime = DateTime.Now;
                }
            }
        }

        // filtrowanie obrazu
        BitmapImage bi;
        using (var bitmap = (Bitmap)eventArgs.Frame.Clone())
        {
            if (Grayscaled)
            {
                using (var grayscaledBitmap = Grayscale.CommonAlgorithms.BT709.Apply(bitmap))
                {
                    bi = grayscaledBitmap.ToBitmapImage();
                }
            }
            else if (Thresholded)
            {
                using (var grayscaledBitmap = Grayscale.CommonAlgorithms.BT709.Apply(bitmap))
                using (var thresholdedBitmap = new Threshold(Threshold).Apply(grayscaledBitmap))
                {
                    bi = thresholdedBitmap.ToBitmapImage();
                }
            }
            else // (Original)
            {
                bi = bitmap.ToBitmapImage();
            }
        }
        bi.Freeze();
        Dispatcher.CurrentDispatcher.Invoke(() => Image = bi);
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);
        StopCamera();
    }
}
```

## 6. Włączenie nagrywania

```
private void StartRecording()
{
    var dialog = new SaveFileDialog();
    dialog.FileName = "Video1";
    dialog.DefaultExt = ".avi";
    dialog.AddExtension = true;
    var dialogresult = dialog.ShowDialog();
    if (dialogresult != true)
    {
        return;
    }
    _firstFrameTime = null;
    _writer = new VideoFileWriter();
    _writer.Open(dialog.FileName, (int)Math.Round(Image.Width, 0), (int)Math.Round(Image.Height, 0));
    _recording = true;
}
```

## 7. Zapisanie obrazu

```
private void SaveSnapshot()
{
    var dialog = new SaveFileDialog();
    dialog.FileName = "Snapshot1";
    dialog.DefaultExt = ".png";
    var dialogresult = dialog.ShowDialog();
    if (dialogresult != true)
    {
        return;
    }
    var encoder = new PngBitmapEncoder();
    encoder.Frames.Add(BitmapFrame.Create(Image));
    using (var filestream = new FileStream(dialog.FileName, FileMode.Create))
    {
        encoder.Save(filestream);
    }
}
```

### 3. Wnioski

Przy obsłudze kamery pomocne jest korzystanie z gotowych bibliotek. Do obsługi kamery przydatne są funkcje zawarte w takich bibliotekach jak AForge.Video czy Accord.Video.

Aplikacje służącą do komunikacji z modemem napisano w języku C#. W aplikacji udało się zaimplementować funkcje służące połączenia się z kamerą i wyświetlania obrazu, dodania filtrów do obrazu z kamery, zapisu obrazu z kamery w formacie JPG. Nie udało się zaimplementować się prawidłowo działających funkcji zapisu obrazu z kamery w formie filmu oraz wygenerowania strony html z odświeżanym automatycznie obrazem z kamery.

### Bibliografia

- [1] Strona internetowa: [http://accord-framework.net/docs/html/T\\_Accord\\_Video\\_FFMPEG\\_VideoFileReader.htm](http://accord-framework.net/docs/html/T_Accord_Video_FFMPEG_VideoFileReader.htm)
- [2] Strona internetowa: <https://programmerall.com/article/89941037611/>
- [3] Strona internetowa: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.media.imaging.bitmapimage?view=windowsdesktop-6.0>
- [4] Strona internetowa: <https://fileinfo.com/extension/avi>
- [5] Strona internetowa: [http://www-old.wemif.pwr.wroc.pl/wslowko/oo/10%27cw10\\_opis.pdf](http://www-old.wemif.pwr.wroc.pl/wslowko/oo/10%27cw10_opis.pdf)