

Politechnika Wrocławska
Wydział Elektroniki
Urządzenia Peryferyjne

OBSŁUGA KAMERY USB

Termin zajęć:

WTOREK TN 7:30

Autorzy:

JAKUB CHMIEL 235028

TOMASZ CIEŚLAR 235652

Prowadzący:

dr inż. Tomasz Walkowiak

19 listopada 2018

Spis treści

1	Cel ćwiczenia	2
2	Wstęp	2
3	Założenia projektowe	2
4	Wykorzystane narzędzia	2
5	Kod programu	3
	5.1 Interfejs aplikacji	3
	5.2 Funkcje wykonywalne	4
	5.3 Plik HTML	9
6	Wnioski	9
7	Bibliografia	9

1 Cel ćwiczenia

- Stwierdzić obecność i poprawność kamery podłączonej do portu USB komputera
- Wylistować urządzenia typu cap (kamery) i stworzyć interfejs umożliwiający wybór po nazwie urządzenia z którym chcemy się połączyć
- Zapisać obraz z kamery w dowolnym formacie
- Zapisać obraz z kamery w postaci filmu AVI
- Zmienić tak program, aby generował stronę html z odświeżanym automatycznie obrazem z kamery

2 Wstęp

AVICAP32 – Do obsługi kamery wykorzystano bibliotekę AVICAP32.DLL - bez żadnych dodatkowych modułów w formie źródeł. Łączenie się z kamerami odbywać się będzie za pomocą funkcji i komunikatów API z AVICAP32.DLL. Jest to standardowa biblioteka każdego systemu Win32, jest doinstalowywana jeśli to konieczne wraz ze sterownikami kamery.

3 Założenia projektowe

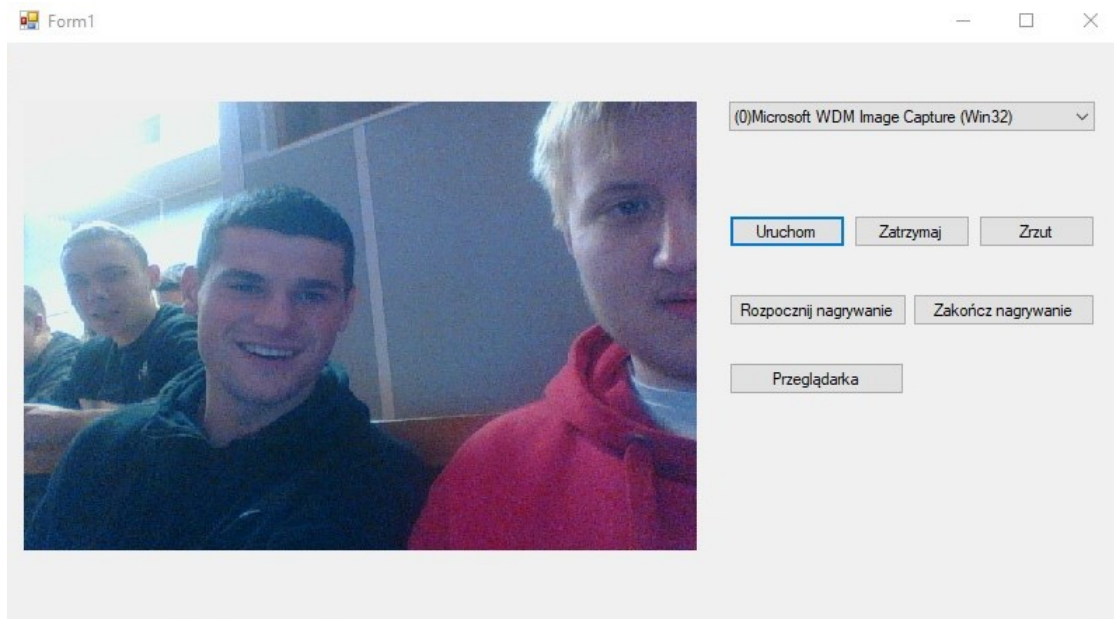
- Program był pisany w języku C#.
- Na komputerze, na którym uruchamiany był program zainstalowano system operacyjny Windows 10 w wersji 64-bitowej.
- W aplikacji użyta została kamera wbudowana w laptop

4 Wykorzystane narzędzia

- Windows Forms - API do implementacji interfejsu graficznego dla platformy .NET.
- AVICAP32.dll - biblioteka pozwalająca na nagrywanie w formacie .avi.

5 Kod programu

5.1 Interfejs aplikacji



```
using [...]\n\nnamespace CameraForms\n{\n    public partial class Form1 : Form\n    {\n        WebCam oWebCam;\n        public Form1()\n        {\n            InitializeComponent();\n        }\n\n        private void Form1_Load(object sender, EventArgs e)\n        {\n            oWebCam = new WebCam();\n            oWebCam.Container = pictureBox1;\n            oWebCam.ComboBox = comboBox1;\n            oWebCam.Load();\n        }\n\n        //Rozpoczenie polaczenia\n        private void button1_Click(object sender, EventArgs e)\n        {\n            oWebCam.OpenConnection();\n        }\n\n        //Wlaczenie kamerki w przegladarce, krzystajac z pliku\n        HTML
```

```

private void button2_Click(object sender, EventArgs e)
{
    Process.Start("file:///C:/Users/DELL/Desktop/
        Politechnika/obraz.html");
}
//
private void button3_Click(object sender, EventArgs e)
{
    oWebCam.Dispose();
}
//Zrobienie screenshota
private void button4_Click(object sender, EventArgs e)
{
    oWebCam.SaveImage();
}
//Rozpoczecie nagrywania
private void button5_Click(object sender, EventArgs e)
{
    oWebCam.StartRecording();
}
//Zakonczenie nagrywania
private void button6_Click(object sender, EventArgs e)
{
    oWebCam.StopRecording();
}
}
}

```

5.2 Funkcje wykonywalne

```

using [...]

namespace CameraForms
{
    public class WebCam : IDisposable
    {
        // Stale sluzace przeciazaniu niezarządzanych kodow
        // Kazda stala reprezentuje stan

        private const short WM_CAP = 0x400;
        private const int WM_CAP_DRIVER_CONNECT = 0x40a;
        private const int WM_CAP_DRIVER_DISCONNECT = 0x40b;
        private const int WM_CAP_EDIT_COPY = 0x41e;
        private const int WM_CAP_SET_PREVIEW = 0x432;
        private const int WM_CAP_SET_OVERLAY = 0x433;
        private const int WM_CAP_SET_PREVIEWRATE = 0x434;
        private const int WM_CAP_SET_SCALE = 0x435;
        private const int WS_CHILD = 0x40000000;
        private const int WS_VISIBLE = 0x10000000;
        const int WM_CAP_DLG_VIDEOFORMAT = WM_CAP + 41;
    }
}

```

```

const int WM_CAP_DLG_VIDEOSOURCE = WM_CAP + 42;
private const int WM_CAP_FILE_SET_CAPTURE_FILE = WM_CAP
    + 20;
private const int WM_CAP_SEQUENCE = WM_CAP + 62;
private const int WM_CAP_STOP = WM_CAP + 68;
private const int WM_CAP_FILE_SAVEAS = WM_CAP + 23;

//zapis bitmapy
const int WM_CAP_SAVEDIB = WM_CAP + 25;
private const short SWP_NOMOVE = 0x2;
private short SWP_NOZORDER = 0x4;
private short HWND_BOTTOM = 1;

private Timer _timer;

//Ta funkcja umożliwia wylistowanie urządzeń kamery
internetowej
[DllImport("avicap32.dll")]
protected static extern bool
    capGetDriverDescriptionA(short wDriverIndex,
        [MarshalAs(UnmanagedType.VBByRefStr)]ref String
            lp.szName,
        int cbName, [MarshalAs(UnmanagedType.VBByRefStr)]
            ref String lp.szVer, int cbVer);

//Ta funkcja umożliwia utworzenie okna, aby można było
je wyświetlić np w PictureBox
[DllImport("avicap32.dll")]
protected static extern IntPtr
    capCreateCaptureWindowA([MarshalAs(UnmanagedType.
        VBByRefStr)] ref string
        lp.szWindowName,
        int dwStyle, int x, int y, int nWidth, int nHeight,
        int hWndParent, int nID);

//Ta funkcja umożliwia zmianę pozycji okna
[DllImport("user32")]
protected static extern int SetWindowPos(IntPtr hwnd,
    int hWndInsertAfter, int x, int y, int cx, int cy,
    int wFlags);

//Ta funkcja umożliwia wysłanie odpowiedniej wiadomości
do okna
[DllImport("user32", EntryPoint = "SendMessageA")]
protected static extern int SendMessage(IntPtr hwnd,
    int wMsg, int wParam,
    [MarshalAs(UnmanagedType.AsAny)] object
        lParam);

[DllImport("user32", EntryPoint = "SendMessageA")]

```

```

protected static extern int SendMessage(IntPtr hwnd,
    int wMsg, bool wParam,
    [MarshalAs(UnmanagedType.AsAny)] object
    lParam);

//Ta funkcja pozwala zniszczyc okno
[DllImport("user32")]
protected static extern bool DestroyWindow(IntPtr hwnd);

// ID urzadzenia
int DeviceID = 0;
// Wskaznik uchwytu dla okna podgladu
private IntPtr hHwnd;
//Lista urzadzen
ArrayList ListOfDevices = new ArrayList();

//Wstawienie obrazka
public PictureBox Container { get; set; }
public ComboBox ComboBox { get; set; }

// Polaczenie z urzadzeniem .
/// Zaladowanie listy urzadzen
public void Load()
{
    string Name = String.Empty.PadRight(100);
    string Version = String.Empty.PadRight(100);
    bool EndOfDeviceList = false;
    short index = 0;
    _timer = new Timer { Interval = (200) };
    _timer.Elapsed += new
        ElapsedEventHandler(TimerTick);
    _timer.Enabled = true;
    _timer.Start();
    // ladowanie wszyskich dostepnych urzadzen do listy
    do
    {
        // pobranie nazwy i wersji
        EndOfDeviceList =
            capGetDriverDescriptionA(index, ref Name,
            100, ref Version, 100);
        // Jesli jest urzadzenie, to dodanie jego nazwy
        do listy
        if (EndOfDeviceList) ListOfDevices.Add("(" +
            index + ")" + Name.Trim());
        index += 1;
    }
    while (!(EndOfDeviceList == false));

    ComboBox.Items.AddRange(ListOfDevices.ToArray());
    ComboBox.SelectedIndex = 0;
}

```

```

}

/// Funkcja do wyswietlania danych wejsciowych z
///  urzadzenia przechwytujacego wideo
///  nalezy utworzyc okno przechwytywania.
public void OpenConnection()
{
    string DeviceIndex = Convert.ToString(DeviceID);
    IntPtr oHandle = Container.Handle;

    // Otwarcie podgladu w PictureBox .
    // Tworzenie podrzednego okna, aby mozna bylo je
    // wyswietlic w PictureBox

    hWndd = capCreateCaptureWindowA(ref DeviceIndex,
        WS_VISIBLE | WS_CHILD, 0, 0, 1280, 720,
        oHandle.ToInt32(), 0);

    // Polaczenie z urzadzeniem
    if (SendMessage(hWndd, WM_CAP_DRIVER_CONNECT, 0, 0)
        != 0)
    {
        // Ustawienie skali podgladu
        SendMessage(hWndd, WM_CAP_SET_SCALE, true, 0);
        // Ustawienie czestotliwosci odswiezania
        // podgladu
        SendMessage(hWndd, WM_CAP_SET_PREVIEWRATE, 66,
            0);

        // Rozpoczecie przegladanie obrazu z kamery
        SendMessage(hWndd, WM_CAP_SET_PREVIEW, true, 0);

        // Dopasowanie okna do ramki graficznej
        SetWindowPos(hWndd, HWND_BOTTOM, 0, 0,
            Container.Width, Container.Height,
            SWP_NOMOVE | SWP_NOZORDER);
    }
    else
    {
        // Blad przy polaczeniu urzadzenia
        DestroyWindow(hWndd);
    }
}

public void Settings()
{
    SendMessage(hWndd, WM_CAP_DLG_VIDEOFORMAT,
        DeviceID, 0);
}

```



```

public void ImageSettings()
{
    SendMessage(hHwnd, WM_CAP_DLG_VIDEOSOURCE,
        DeviceID, 0);
}

void CloseConnection()
{
    SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT,
        DeviceID, 0);
    // zamkniecie okna
    DestroyWindow(hHwnd);
}

public void StartRecording()
{
    SendMessage(hHwnd, WM_CAP_FILE_SET_CAPTURE_FILE, 0,
        @"C:\Users\DELL\Desktop\Politechnika\Semestr
        5\Urządzenia
        peryferyjne\LAB\UP_Kamera\nagranie.avi");
    SendMessage(hHwnd, WM_CAP_SEQUENCE, DeviceID, 0);
}

public void StopRecording()
{
    SendMessage(hHwnd, WM_CAP_FILE_SAVEAS, DeviceID,
        @"C:\Users\DELL\Desktop\Politechnika\Semestr
        5\Urządzenia
        peryferyjne\LAB\UP_Kamera\nagranie.avi");
    SendMessage(hHwnd, WM_CAP_STOP, DeviceID,
        @"C:\Users\DELL\Desktop\Politechnika\Semestr
        5\Urządzenia
        peryferyjne\LAB\UP_Kamera\nagranie.avi");
}

public void SaveWebImg()
{
    SendMessage(hHwnd, WM_CAP_SAVEDIB, DeviceID,
        @"C:\Users\DELL\Desktop\Politechnika\doWebu.jpg");
}

public void SaveImage()
{
    SendMessage(hHwnd, WM_CAP_SAVEDIB, DeviceID,
        @"C:\Users\DELL\Desktop\Politechnika\Semestr
        5\Urządzenia
        peryferyjne\LAB\UP_Kamera\zdjecie.jpg");
}

void TimerTick(object sender, ElapsedEventArgs e)

```

```

    {
        SaveWebImg();
    }
    // Ta funkcja konczy polaczenie z urzadzeniem
    #region IDisposable Members

    public void Dispose()
    {
        CloseConnection();
    }
    #endregion
}
}

```

5.3 Plik HTML

```

<html>
    <head>
    </head>
    <body>
        
        <meta http-equiv="refresh" content="0.2">
    </body>
</html>

```

6 Wnioski

Podczas ćwiczenia poznaliśmy podstawową obsługę kamery USB i funkcje takie jak nagrywanie, przechwytywanie obrazu, czy wyświetlanie obrazu na stronie internetowej. AVICAP32.dll mimo, iż jest bardzo starą biblioteką, umożliwił nam sprawne wykonanie wyżej wymienionych zadań.

7 Bibliografia

- Dokumentacja Microsoftu:
<https://docs.microsoft.com/en-us/windows/desktop/api/vfw/>