

Piotr Chłystek 226100
Michał Chojnacki 225936

Data oddania sprawozdania: 05.01.2017 r.
Termin zajęć: Poniedziałek 7:30-10:15, TP

Urządzenia peryferyjne

Ćwiczenie 9

Obsługa skanera płaskiego (TWAIN lub WIA)

Prowadzący:
dr inż. Jan Nikodem

1. Cel ćwiczenia

Celem ćwiczenia było napisanie programu, który wykryje urządzenia typu skaner, podłączone do komputera, by następnie móc wybrać urządzenie do wykonania skanu obrazu, o zadanych w programie parametrach.

2. Wstęp teoretyczny

Skaner jest urządzeniem, które realizuje między innymi odczyt obrazu, kodu kreskowego, fal magnetycznych i radiowych. Skaner płaski, któremu jest poświęcone ćwiczenie, jest najczęściej stosowanym urządzeniem do przetwarzania obrazów do formy cyfrowej. Skaner jest urządzeniem krokowym – rejestruje kolejne linie obrazu, wraz z przesuwającą się głowicą. Najważniejszymi elementami budowy skanera są źródło światła, elementy fotoczułe, układ optyczny, filtr dichroiczny, układy elektroniczne, płyta do układania dokumentów. W skanerze jest emitowane światło białe, które naświetla dokument i przesuwa się wzdłuż niego, dzięki wbudowanemu mechanizmowi napędowemu. Światło białe przyjmuje barwę fragmentu obrazu i przechodzi przez układ optyczny, pada na filtr dichroiczny, który rozdziela strumień światła na trzy strumienie, padające na układ czujników fotoelektrycznych, pokrytych filtrem RGB. Na jasność barw składowych ma wpływ prąd, generowany przez element fotoczuły. W przetworniku analogowo-cyfrowym (A/D – analog to digital) prąd jest zamieniany na sygnał cyfrowy. Parametrami skanowanego obrazu są rozdzielczość, jasność, kontrast i ilość bitów kodujących kolor.

3. Opis programu

Program został napisany w technologii C#, z wykorzystaniem biblioteki WIA 1.0

```
WIA.ICommonDialog commonDialog; //okno dialogowe
WIA.Device scanner;               //wybrany skaner
List<string> devices;              //lista urządzeń
WIA.ImageFile scannedImage;       //zeskanowany obraz
Byte[] imageBytes;                //tablica bajtów
Image image;                      //obraz w pamięci
WIA.DeviceManager manager;        //manager urządzeń

//potrzebne parametry
//https://msdn.microsoft.com/enus/library/windows/desktop/ms630196(v=vs.85).aspx
private const int WIA_DIP_FIRST = 2;
private const int WIA_DPA_FIRST = WIA_DIP_FIRST + WIA_RESERVED_FOR_NEW_PROPS;
private const int WIA_DPC_FIRST = WIA_DPA_FIRST + WIA_RESERVED_FOR_NEW_PROPS;
private const int WIA_DPS_FIRST = WIA_DPC_FIRST + WIA_RESERVED_FOR_NEW_PROPS;
private const int WIA_IPA_FIRST = WIA_DPS_FIRST + WIA_RESERVED_FOR_NEW_PROPS;
private const int WIA_IPC_FIRST = WIA_IPA_FIRST + WIA_RESERVED_FOR_NEW_PROPS;
private const int WIA_IPS_FIRST = WIA_IPC_FIRST + WIA_RESERVED_FOR_NEW_PROPS;

private const int WIA_RESERVED_FOR_NEW_PROPS = 1024;

private const int WIA_IPS_CUR_INTENT = 6146;
private const int WIA_IPS_XRES = 6147;
private const int WIA_IPS_YRES = 6148;

private const int WIA_INTENT_IMAGE_TYPE_COLOR = 0x00000001;
private const int WIA_INTENT_IMAGE_TYPE_GRAYSCALE = 0x00000002;
private const int WIA_INTENT_IMAGE_TYPE_TEXT = 0x00000004;
private const int WIA_INTENT_IMAGE_TYPE_MASK = 0x0000000F;
private const int WIA_INTENT_MINIMIZE_SIZE = 0x00010000;
private const int WIA_INTENT_MAXIMIZE_QUALITY = 0x00020000;
private const int WIA_INTENT_BEST_PREVIEW = 0x00040000;
private const int WIA_INTENT_SIZE_MASK = 0x000F0000;
private const int FEED_READY = 0x01;
```

```

//wartosc formatu pliku jako string dla formatow BMP i JPG
private const string wiaFormatBMP = "{B96B3CAB-0728-11D3-9D7B-0000F81EF32E}";
private const string wiaFormatJPEG = "{B96B3CAE-0728-11D3-9D7B-0000F81EF32E}";
public Form1()
{
    InitializeComponent();
}

private void searchDevices() //funkcja wyszukujaca urzadzenia
{
    devices = new List<string>(); //lista znalezionych urzadzen
    manager = new WIA.DeviceManager(); //menadzer urzadzen z biblioteki WIA
    foreach (WIA.DeviceInfo devInfo in manager.DeviceInfos)
        devices.Add(devInfo.DeviceID); //petla dodajaca urzadzenia do listy
}

private void buttonSearchDev_Click(object sender, EventArgs e)
{
    searchDevices(); //przycisk wywolujacy funkcje
}

private void buttonScan_Click(object sender, EventArgs e) //przycisk
//uruchamiajacy skanowanie
{
    if (scanner != null) //jezeli wybrano skaner
        scan(scanner.DeviceID); //rozpoczyna sie skanowanie
    else throw new Exception("Select scanner!");
}

private void buttonChooseScanner_Click(object sender, EventArgs e)
{
    //przycisk, który uruchamia okno dialogowe z biblioteki WIA
    commonDialog = new WIA.CommonDialog();
    scanner =
commonDialog.ShowDialog(WIA.WiaDeviceType.UnspecifiedDeviceType, true,
false); //okno, pozwalające wybrać urządzenie
}

void scan(string scannerId) //funkcja do skanowania
{
    scanner = null;
    foreach (WIA.DeviceInfo devInfo in manager.DeviceInfos)
        if (devInfo.DeviceID == scannerId) //polaczenie ze skanerem
            scanner = devInfo.Connect();
    if (scanner == null) throw new Exception("Selected scanner is
unavailable!");

    WIA.Item item = scanner.Items[1];

    try
    {
        //odczyt pliku
        commonDialog = null;
        commonDialog = new WIA.CommonDialog();
        scanInit();
        scannedImage = (WIA.ImageFile)commonDialog.ShowTransfer(item,
wiaFormatBMP, false);
        imageBytes = (byte[])scannedImage.FileData.get_BinaryData();
        MemoryStream ms = new MemoryStream(imageBytes);
        image = Image.FromStream(ms);
        pbScannedImage.Image = image; //wyświetlenie obrazu w programie
    }
}

```

```

        catch (Exception e)
        {
            throw e;
        }
    finally
    {
        Marshal.ReleaseComObject(scannedImage); //zwalnia obiekt COM
    }
}

private void scanInit()
{
    Object Object1 = null; //zmienne typu Object przechowujące
    Object Object2 = null; //właściwości skanera
    Int32 DPI = Convert.ToInt32(textBoxDpi.Text);
    Int32 C = Convert.ToInt32(trackBarContrast.Value);
    Int32 B = Convert.ToInt32(trackBarBrightness.Value);
    try
    {
        Object1 = (Object)WIA_IPS_CUR_INTENT.ToString(); // "6146";

        if (radioButtonColor.Checked) //obraz kolorowy
            Object2 = (Object)WIA_INTENT_IMAGE_TYPE_COLOR;
        else
            //obraz czarno-biały
            Object2 = (Object)WIA_INTENT_IMAGE_TYPE_GRAYSCALE;
        scanner.Items[1].Properties.get_Item(ref Object1).set_Value(ref
            Object2);

        Object1 = (Object)"6147";
        Object2 = (Object)DPI; //rozdzielczosc
        scanner.Items[1].Properties.get_Item(ref Object1).set_Value(ref
            Object2);

        Object1 = (Object)"6148";
        Object2 = (Object)DPI;
        scanner.Items[1].Properties.get_Item(ref Object1).set_Value(ref
            Object2);

        Object1 = (Object)(WIA_IPS_FIRST + 9).ToString();
        Object2 = (Object)C; //kontrast
        scanner.Items[1].Properties.get_Item(ref Object1).set_Value(ref
            Object2);

        Object1 = (Object)(WIA_IPS_FIRST + 8).ToString();
        Object2 = (Object)B; //jasnosc
        scanner.Items[1].Properties.get_Item(ref Object1).set_Value(ref
            Object2);
    }
    catch
    {
        MessageBox.Show("Scanner is not ready or it does not suport this
function.\r\n\r\n Setting default settings.", "Initialization",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    Object1 = null;
    Object2 = null;
}
//przycisk kolorowego obrazu
private void radioButtonColor_Checked(object sender, EventArgs e)
{
    if (radioButtonColor.Checked == true)

```

```

        {
            radioButtonColor.CheckedChanged -= new
                EventHandler(this.radioButtonColor_Checked);
            radioButtonGrayscale.CheckedChanged -= new
                EventHandler(this.radSelCols_Checked);
            radioButtonColor.Checked = false;
            radioButtonGrayscale.Checked = true;
            radioButtonColor.CheckedChanged += new
                EventHandler(this.radioButtonColor_Checked);
            radioButtonGrayscale.CheckedChanged += new
                EventHandler(this.radSelCols_Checked);
        }
    }
    //przycisk do trybu czarno-bialego
    private void radSelCols_Checked(object sender, EventArgs e)
    {
        if (radioButtonGrayscale.Checked == true)
        {
            radioButtonColor.CheckedChanged -= new
                EventHandler(this.radioButtonColor_Checked);
            radioButtonGrayscale.CheckedChanged -= new
                EventHandler(this.radSelCols_Checked);
            radioButtonGrayscale.Checked = false;
            radioButtonColor.Checked = true;
            radioButtonColor.CheckedChanged += new
                EventHandler(this.radioButtonColor_Checked);
            radioButtonGrayscale.CheckedChanged += new
                EventHandler(this.radSelCols_Checked);
        }
    }
    //przycisk do uruchomienia okna dialogowego do zapisu zeskanowanego obrazu
    private void buttonSave_Click(object sender, EventArgs e)
    {
        if (pbScannedImage.Image != null)
        {
            Bitmap image = (Bitmap) pbScannedImage.Image;
            saveFileDialog.Filter = "Bitmap Image|*.bmp";
            saveFileDialog.Title = "Save an Image File";
            saveFileDialog.ShowDialog();
            System.IO.FileStream fs = (System.IO.FileStream)
                saveFileDialog.OpenFile();
            image.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);
            fs.Close();
            MessageBox.Show("Saved!");
        }
    }
}

```

4. Wnioski

Dzięki funkcjonalności biblioteki WIA implementacja programu, obsługującego skaner okazała się być łatwiejsza niż się spodziewaliśmy. Program działa poprawnie i dzięki zmianie odpowiednich parametrów obrazu można uzyskać różne efekty, leżące w zakresie możliwości sprzętowych skanera.