

Urządzenia peryferyjne

Autor:

Tymon Tobolski (181037)

Jacek Wieczorek (181043)

Prowadzący:

Dr inż. Jacek Mazurkiewicz

Wydział Elektroniki

III rok

Pn 8.15 - 11.00

14 stycznia 2012

1 Cel laboratorium

Celem laboratorium było zapoznanie się z zasadą pracy i obsługą skanera płaskiego. W tym celu wykorzystana została biblioteka WIA 1.0 i język programowania *C#*

2 Parametry skanowania

Poza samym zeskanowaniem obrazu, udało nam się również zmieniać podstawowe parametry skanowania, takie jak:

- czarnobiałe lub kolorowe
- rozdzielczość (DPI)
- kontrast
- jasność

2.1 Rozdzielczość

DPI (dots per inch) - liczba plamek przypadająca na cal, używane jako miara rozdzielczości drukarek, ploterów, skanerów itp.

2.2 Kontrast

TODO

2.3 Jasność

TODO

3 Implementacja

Program do obsługi skanera płaskiego napisany został w języku *C#* z wykorzystaniem biblioteki WIA 1.0 .

W celu wyboru urządzenia skanującego skorzystaliśmy ze standardowego okna dialogowego dostępnego w bibliotece.

```

1 private bool chooseDevice()
{
    try
    {
        Scanner = Dialog1.ShowSelectDevice(WIA.WiaDeviceType.ScannerDeviceType, true, true);
        if (Scanner != null) return true;
    }
    catch (Exception ex)
    {
        //Message
11 }
    return false;
}

```

Do inicjalizacji urządzenia i określenia parametrów skanowania :

```

private void InitializeITEM()
{
    Object Object1 = null;
    Object Object2 = null;
    Int32 DPI = Convert.ToInt32(textBox1.Text) ;
6    Int32 C = Convert.ToInt32(textBox2.Text);
    Int32 B = Convert.ToInt32(textBox3.Text);
    try
    {
        Object1 = (Object)WIA_IPS_CUR_INTENT.ToString(); // "6146";

        if (radioButton1.Checked)
            Object2 = (Object)WIA_INTENT_IMAGE_TYPE.COLOR; //WIA_INTENT_IMAGE_TYPE.GRAYSCALE;
        else
            Object2 = (Object)WIA_INTENT_IMAGE_TYPE.GRAYSCALE;
16    Scanner.Items[1].Properties.GetItem(ref Object1).setValue(ref Object2);

        Object1 = (Object)"6147";
        Object2 = (Object)DPI; //rozdzielczosc
        Scanner.Items[1].Properties.GetItem(ref Object1).setValue(ref Object2);

        Object1 = (Object)"6148";
        Object2 = (Object)DPI;
        Scanner.Items[1].Properties.GetItem(ref Object1).setValue(ref Object2);

26    Object1 = (Object)(WIA_IPS_FIRST + 9).ToString();
        Object2 = (Object)C; //kontrast
        Scanner.Items[1].Properties.GetItem(ref Object1).setValue(ref Object2);

        Object1 = (Object)(WIA_IPS_FIRST + 8).ToString();
        Object2 = (Object)B; //jasnosc
        Scanner.Items[1].Properties.GetItem(ref Object1).setValue(ref Object2);

    }
    catch
36 {
        //Message
    }
    Object1 = null;
    Object2 = null;
}

```

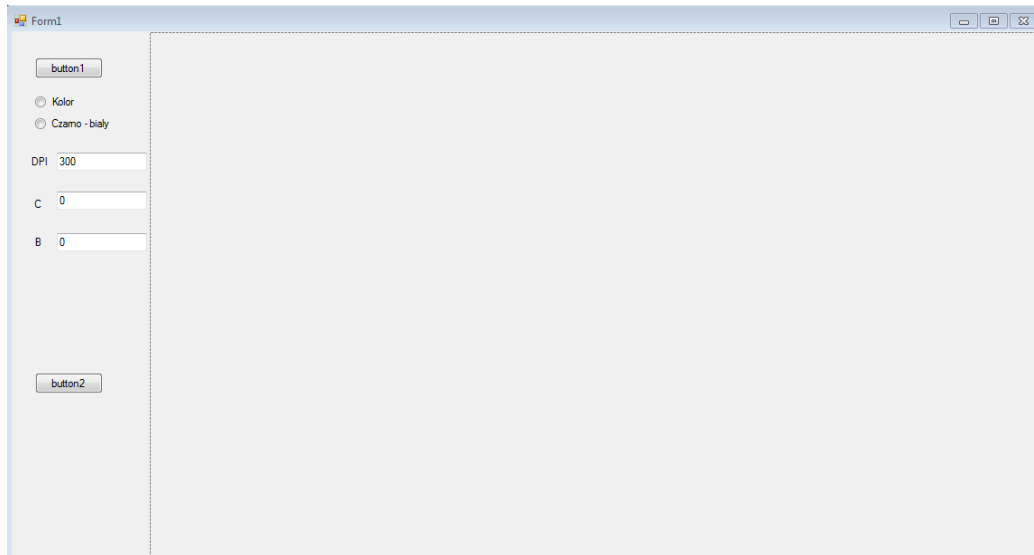
Proces skanowania :

```
...
InitializeITEM(); //inicjuje ustawienia skanera
Img = (ImageFile)Dialog1.ShowTransfer(Scanner.Items[1], wiaFormatBMP, true);
imageBytes = (byte[])Img.FileData.get_BinaryData();
MemoryStream ms = new MemoryStream(imageBytes);
Image image = Image.FromStream(ms);
pictureBox1.Image = image;
...
```

Zapis obrazu do pliku :

```
private void button2_Click(object sender, EventArgs e)
2 {
    if (pictureBox1.Image != null)
    {
        FileStream file = File.OpenWrite("d.bmp");
        file.Write(imageBytes, 0, imageBytes.Length);
        file.Close();
        MessageBox.Show("zapisano");
    }
}
```

Przykładowe okno programu :



Rysunek 1: Przykładowe okno programu

4 Wnioski

Napisanie programu do obsługi skanera płaskiego w wykorzystaniem biblioteki WIA nie jest zadaniem trudnym. Jedynym problemem z jakim spotkaliśmy się podczas implementacji było znalezienie odpowiednich wartości stałych by ustawić parametry skanowania.