

Urządzenia peryferyjne - laboratorium

Karol Kulawiec 241281

Jakub Kalina 241346

25.11.2019

1 Wstęp

Celem wykonanego ćwiczenia było napisanie programu, za pomocą którego będzie można zeskanować obraz ze skanera płaskiego CanoScan LiDE 100.

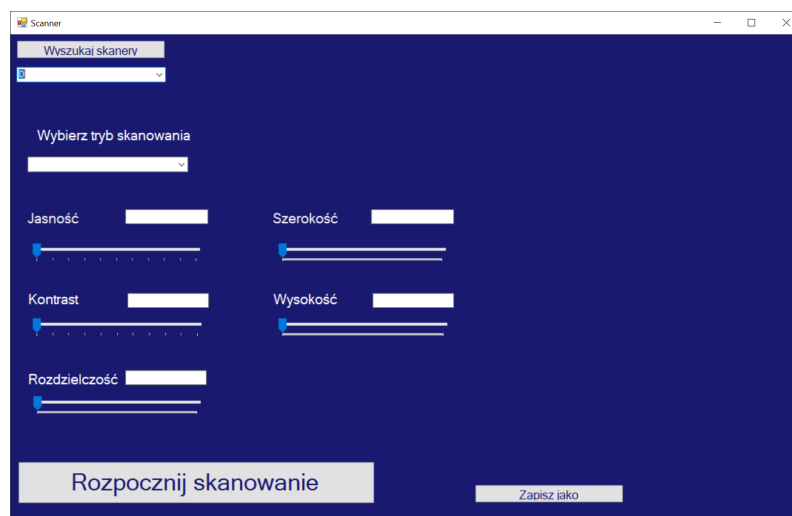
Skaner płaski CanoScan LiDE 100 to urządzenie służące do skanowania dokumentów. Jest to czytnik krokowy, czyli głowica czytająca skaner lub inne medium pod nią, nie rejestruje całego obrazu na raz (tak jak jest to w przypadku aparatów fotograficznych), a przesuwa się, krok po kroku. Skaner badany podczas zajęć charakteryzuje się rozdzielczością optyczną wynoszącą 2400x4800 dpi. Skaner skanuje, ponieważ wydzielane światło odbija się od dokumentu, a następnie dociera do optycznych elementów głowicy skanera i jest analizowane przez procesor, który rekonstruuje obraz w postaci elektronicznej.

Do zamiany dokumentu z papierowego na jego obraz elektroniczny skaner potrzebuje sześciu podstawowych komponentów: źródła światła białego, systemu luster do odbicia światła, soczewek do skupienia światła, elementów CCD (Charge-Coupled Device) przechwytyjących odbite światło, konwertera ADC (Analog-Digital Converter) zamieniającego sygnał z elementów CCD na cyfrowy i silnika krokowego, który przesuwa wszystkie powyższe elementy wzdłuż skanowanego dokumentu.

2 Opis aplikacji

Program został napisany w technologii *C#*. Najpierw wykorzystano bibliotekę Vintasoft.Twain, która pozwala kontrolować skaner płaski i ADF, kamerę internetową i cyfrową oraz dowolne inne urządzenie kompatybilne ze standardem TWAIN. Wykorzystanie tej biblioteki spowodowało, że aplikacja działała prawidłowo, ale okazało się, że własnoręczne zmiany obrazu nie mają sensu, ponieważ biblioteka ma wbudowaną własną "aplikację", która pojawiała się, po wyrażeniu chęci skanowania i w której można było ustawiać parametry obrazu. Dlatego zdecydowaliśmy się napisać drugą wersję programu wykorzystując mniej automatyczną bibliotekę WIA.

Aplikacja prezentuje się w następujący sposób:



Rysunek 1: Interface aplikacji

Po lewej stronie aplikacji, znajdują się następujące funkcjonalności:

- przycisk do przeszukiwania komputera w celu znalezienia dostępnych skanerów,
- rozwijana lista, prezentująca dostępne skanery,
- rozwijana lista, prezentująca dostępne tryby skanowania (mapa bitowa, monochromatyczny, model RGB),
- suwak zintegrowany z polem do wpisywania liczb, które przedstawiają wybraną jasność,
- suwak zintegrowany z polem do wpisywania liczb, które przedstawiają wybrany kontrast,
- suwak zintegrowany z polem do wpisywania liczb, które przedstawiają wybraną rozdzielczość,
- suwaki zintegrowany z polami do wpisywania liczb, które przedstawiają wybraną szerokość i wysokość,
- przycisk rozpoczynający skanowanie dokumentu.

Po prawej stronie, znajduje się pole, w którym, po zakończeniu skanowania, wyświetli się skanowany obraz. Pod polem znajduje się przycisk, który umożliwi nam zapisanie zeskanowanego obrazu do pliku.

3 Opis ważniejszych funkcji

- Zasada działań suwaków z zintegrowanymi polami do wpisywania wartości, na podstawie jasności:

```
private void trackBarScanningBrightness_Scroll(object sender,
    EventArgs e)
{
    brightness = trackBarScanningBrightness.Value;
    textBoxScanningBrightness.Text = brightness.ToString();
    textBoxScanningBrightness.Update();
}
private void textBoxScanningBrightness_TextChanged(object
    sender, EventArgs e)
{
    if (!Int32.TryParse(textBoxScanningBrightness.Text, out
        brightness))
    {
        MessageBox.Show("Wprowadzono nieprawidłowa jasność,
            spróbuj ponownie!");
    }
    else
    {
        try
        {
            trackBarScanningBrightness.Value = brightness;
            trackBarScanningBrightness.Update();
        }
        catch (Exception)
        {
            MessageBox.Show("Wprowadzono wartość poza
                zakresem, spróbuj ponownie!");
            brightness = 0;
            trackBarScanningBrightness.Value = brightness;
            trackBarScanningBrightness.Update();
            textBoxScanningBrightness.Text =
                brightness.ToString();
        }
    }
}
```

Pierwsza funkcja pobiera wartość z suwaka oraz daną wartość zamienia na string i wyświetla ją w polu. Druga funkcja najpierw sprawdza czy wprowadzony tekst to liczba, jeżeli tak, to próbuje ustawić suwak na podaną wartość, jeżeli wartość będzie poza limitem, zmienia ją na 0 i taką ustawia.

- Wybór trybu skanowania:

```
private void comboBoxScanningMode_SelectedIndexChanged(object sender, EventArgs e)
{
    string chosenScanningMode = comboBoxScanningMode.Text;
    if (chosenScanningMode.Equals("Kolorowe")) colorMode = 1;
    if (chosenScanningMode.Equals("Czarno biale")) colorMode = 2;
    if (chosenScanningMode.Equals("Szare")) colorMode = 4;
}
```

Wybór z listy dostępnego trybu skanowania, powoduje ustawienie ScanningMode na odpowiadającą mu wartość.

- Rozpoczęcie skanowania:

```
private void buttonStartScanning_Click(object sender, EventArgs e)
{
    try
    {
        // Polaczenie z wybranym skanerem
        var connectedDevice = firstScannerAvailable.Connect();

        // Wybranie obiektu skanowania
        var scannerItem = connectedDevice.Items[1];

        // Ustawienie opcji skanera
        AdjustScannerSettings(scannerItem, resolution, 0, 0,
            documentWidth, documentHeight, brightness,
            contrast, colorMode);

        // Wykonanie skanowania
        scannedDocument =
            (ImageFile)scannerItem.Transfer("{B96B3CAE-0728-11D3-9D7B-0000F81EF32E}");

        filePath += "\\scan.jpeg";

        // Sprawdzenie czy sciezka jest wolna
        if (File.Exists(filePath))
        {
            File.Delete(filePath);
        }

        // Zapis do pliku
        scannedDocument.SaveFile(filePath);

        MessageBox.Show("Zeskanowany dokument zostal zapisany do pliku");

        // Wyświetlenie zeskanowanego dokumentu w formie
        pictureBoxScannedDocument.ImageLocation = filePath;
    }
}
```

```

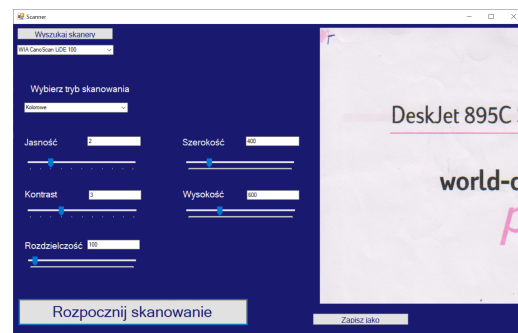
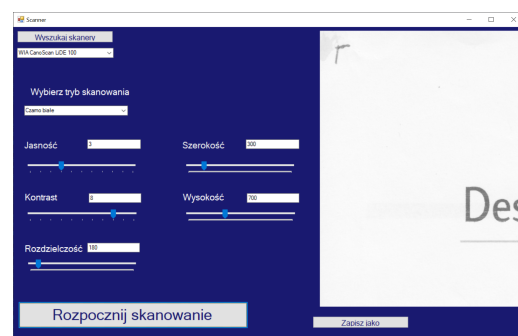
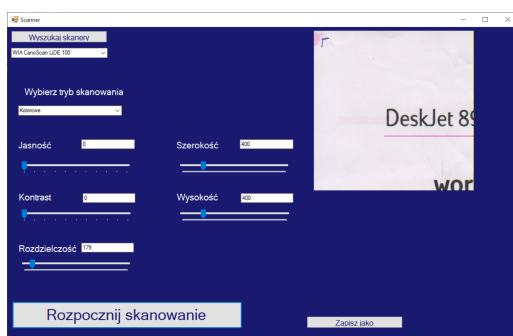
catch(Exception)
{
    MessageBox.Show("Wystąpił błąd podczas skanowania,  
upewnij się, że odpowiedni skaner jest wybrany  
oraz podłączony");
}
}

```

Na początku ustawiamy parametry skanowania wg wcześniej wybranych parametrów (skaner, rozdzielczość, szerokość i wysokość, jasność, kontrast i tryb druku). Następnie skanowany jest dokument (podaje się odpowiedni ciąg znaków), a następnie, w wybranym przez nas folderze, zapisywany jest plik pod nazwą "scan.jpg".

4 Różne opcje skanowania

Poniżej przedstawiono różne efekty skanowanych dokumentów:



5 Wnioski

Biblioteka Vintasoft.Twain posiada wiele ciekawych funkcjonalności, umożliwiających zarządzać skanerem w naprawdę łatwo sposób, ponieważ zawiera wiele wbudowanych opcji.

Biblioteka WIA również posiada wiele ciekawych funkcjonalności, jednak aby wydrukować wg własnych potrzeb, należy manualnie pobrać oraz ustawić dane parametry. Korzystanie z biblioteki również nie stanowiło problemu.

Link do repozytorium, w którym znajduje się program napisany na zajęciach: <https://github.com/JakubKalina/UP-Skaner>.