

Wrocław, 10 stycznia 2020r.

Mikołaj Baran, 241128
Jakub Aniszewski, 241133

prowadzący: Dominik Żelazny

Laboratorium Urządzeń Peryferyjnych

Ćwiczenie 9 - Obsługa skanera płaskiego (TWAIN)

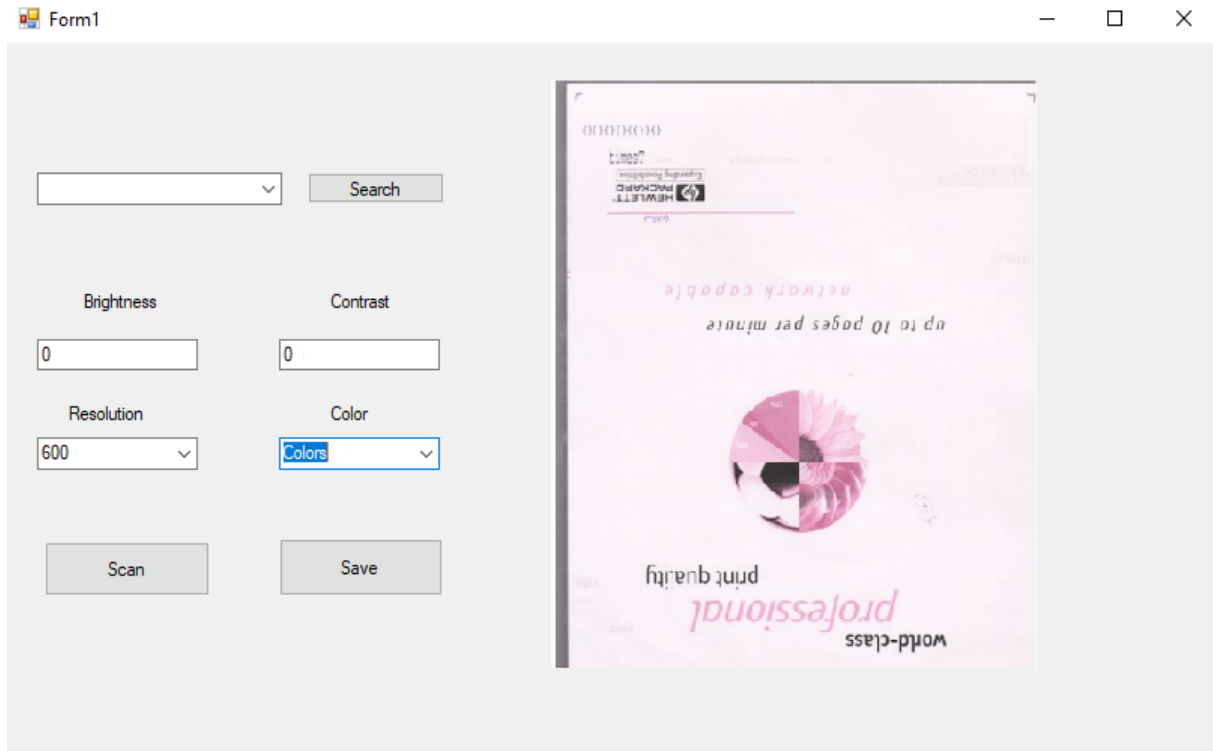
1 Cel ćwiczenia

1. Sprawdzić czy skaner działa poprawnie
2. Napisać program wykonujący skanowanie przy pomocy skanera płaskiego Twain. Możliwości programu obejmują:
 - Skanowanie z wykorzystaniem UI
 - Skanowanie bez wykorzystania UI
 - Wyświetlanie uzyskanego obrazu
 - Zmiana rozdzielczości skanera
 - Zmiana trybu skanowania
3. Rozszerzyć działanie programu:
 - Obsługa różnych trybów przesyłania danych
 - Zapis skanowanych obrazów do plików graficznych

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 przesuwaną się głowicą. Najważniejszymi elementami budowy skanera są źródło światła, elementy foto-czułe, 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 foto-czuły. 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

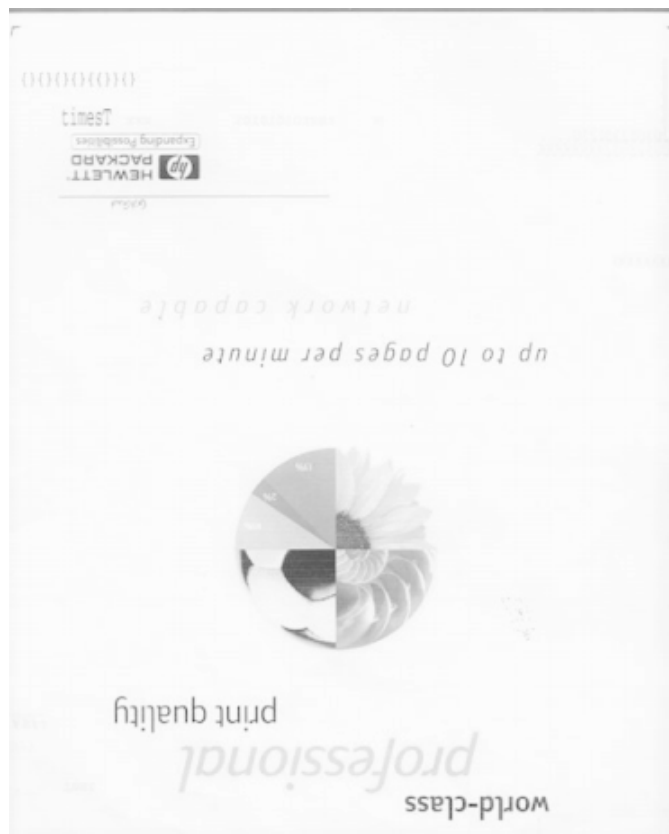


Rysunek 1: Przykładowy wyniki skanowania wyświetlony w programie.

Obsługa naszego programu jest bardzo prosta. Na początku musimy znaleźć urządzenie którego chcemy użyć. Następnie możemy podać takie parametry dla zeskanowanego obrazu jak:

- Jasność - wartość zero oznacza brak zmiany w parametrze
- Kontrast - wartość zero oznacza brak zmiany w parametrze
- Rozdzielczość - zmiana wartości skanuje powiększone fragmenty skanowanego obrazka, działa dokładnie tak jak narzędzie z systemu Windows.
- Tryb koloru - dostępne są dwa tryby, kolorowy oraz czarno-biały.

Po wybraniu parametru możemy nacisnąć przycisk **Scan** i rozpocznie się skanowanie. Stworzony obraz możemy zapisać w wybranej lokalizacji przez naciśnięcie przycisku **Save**.



Rysunek 2: Przykład zapisanego skanu w trybie koloru czarno-biały oraz ze zwiększoną jasnością.

4 Najważniejsze funkcje w programie

Funkcja do wyszukania skanera.

```
private void findScanners(object sender, EventArgs e)
{
    for (int i = 1; i <= deviceManager.DeviceInfos.Count; i++)
    {
        if (deviceManager.DeviceInfos[i].Type !=
            WiaDeviceType.ScannerDeviceType)

            continue;
        else
        {
            searchCB.Items.Add(deviceManager.DeviceInfos[i].
                               Properties["Name"].get_Value());

            firstDevice = deviceManager.DeviceInfos[i];
            break;
        }
    }
}
```

Korzystając z obiektu **DeviceManager** w języku C# wyszukujemy wszystkie urządzenia typu **ScannerDeviceType**. Jeżeli nie trafimy na takie urządzenie kontynuujemy pętlę. Jeżeli natomiast natrafimy na skaner dodajemy jego nazwę do naszego ComboBox'a z wyborami oraz ustawiamy ostatni znaleziony skaner jako domyślnie wybrany.

Funkcja do skanowania.

```
private void scan(object sender, EventArgs e)
{
    device = firstDevice.Connect();
    scannerItem = device.Items[1];
    brightness = Int32.Parse(brightnessText.Text);
    contrast = Int32.Parse(contrastText.Text);
    AdjustScannerSettings(scannerItem, resolution, 0, 0, width_pixel,
        height_pixel, brightness, contrast, color_mode);

    var path = "temp"+index.ToString()+".jpeg";
    index++;
    imageFile = (ImageFile)scannerItem.Transfer(FormatID.wiaFormatJPEG);

    imageFile.SaveFile(path);

    this.bit.SizeMode = PictureBoxSizeMode.Zoom;
    bit.Image = new Bitmap(path);
}
```

Na początku musimy połączyć się z wybranym skanerem oraz zapisać obiekt tego rzędu w zmiennej **scannerItem**. Następne trzy linijki odpowiadają za czytanie wybranych jasności oraz kontrastu a następnie ustawienie wybranych parametrów w skanerze. Dalej zapisujemy wczytany obraz w domyślnej ścieżce, zmienna **index** pomaga nam gdy skanujemy wiele obrazów, nie powtarzamy wtedy jednej nazwy. Na koniec umieszczamy obraz w bitMapie aby był wyświetlony w naszym programie.

Funkcja do zapisywania.

```
private void save(object sender, EventArgs e)
{
    Bitmap picture = (Bitmap)bit.Image;
    saveFileDialog.Title = "Save an Image File";
    saveFileDialog.ShowDialog();
    System.IO.FileStream fs = (System.IO.FileStream)saveFileDialog.OpenFile();
    picture.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);
    fs.Close();
}
```

Bardzo prosta funkcja, wczytujemy nasz obrazek z wyświetlanej bitMapy a następnie korzystając z rozwiązań języka C# oraz jego integracji z systemem Windows otwieramy okno dialogowe służące do zapisu.

Funkcje do zmiany parametrów.

```
private static void SetWIAProperty(IProperties properties ,
                                   object propName, object propValue)
{
    Property prop = properties.get_Item(ref propName);
    prop.set_Value(ref propValue);
}

private static void AdjustScannerSettings(IItem scannerItem ,
int scanResolutionDPI , int scanStartLeftPixel ,
int scanStartTopPixel , int scanWidthPixels ,
int scanHeightPixels , int brightnessPercents ,
int contrastPercents , int colorMode)
{
    const string WIA_SCAN_COLOR_MODE = "6146";
    const string WIA_HORIZONTAL_SCAN_RESOLUTION_DPI = "6147";
    const string WIA_VERTICAL_SCAN_RESOLUTION_DPI = "6148";
    const string WIA_HORIZONTAL_SCAN_START_PIXEL = "6149";
    const string WIA_VERTICAL_SCAN_START_PIXEL = "6150";
    const string WIA_HORIZONTAL_SCAN_SIZE_PIXELS = "6151";
    const string WIA_VERTICAL_SCAN_SIZE_PIXELS = "6152";
    const string WIA_SCAN_BRIGHTNESS_PERCENTS = "6154";
    const string WIA_SCAN_CONTRAST_PERCENTS = "6155";

    SetWIAProperty(scannerItem.Properties ,
WIA_HORIZONTAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
    SetWIAProperty(scannerItem.Properties ,
WIA_VERTICAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
    SetWIAProperty(scannerItem.Properties ,
WIA_HORIZONTAL_SCAN_START_PIXEL, scanStartLeftPixel);
    SetWIAProperty(scannerItem.Properties ,
WIA_VERTICAL_SCAN_START_PIXEL, scanStartTopPixel);
    SetWIAProperty(scannerItem.Properties ,
WIA_HORIZONTAL_SCAN_SIZE_PIXELS, scanWidthPixels);
    SetWIAProperty(scannerItem.Properties ,
WIA_VERTICAL_SCAN_SIZE_PIXELS, scanHeightPixels);
    SetWIAProperty(scannerItem.Properties ,
WIA_SCAN_BRIGHTNESS_PERCENTS, brightnessPercents);
    SetWIAProperty(scannerItem.Properties ,
WIA_SCAN_CONTRAST_PERCENTS, contrastPercents);
    SetWIAProperty(scannerItem.Properties ,
WIA_SCAN_COLOR_MODE, colorMode);
}
```

Ponieważ korzystamy z biblioteki **WIA** ustawianie parametrów naszego skanera również jest bardzo proste. Każdy z tych parametrów ma swoją unikalną nazwę po której jest rozpoznawany. Stworzyliśmy więc funkcję **SetWIAProperty** przyjmującą wybrany parametr który wyszukuje i przypisuje do zmiennej **prop** oraz wartość którą ustawia. Funkcja ta działa dla pojedynczych opcji. **AdjustScannerSettings** to funkcja do której przesyłamy wszystkie parametry na raz i ustawiamy je przy pomocy opisanej wcześniej metody.

5 Wnioski

Dzięki zastosowaniu biblioteki **WIA** oraz bardzo dobrej współpracy języka C# oraz Windows'a obsługa skanera okazała się bardzo prosta. Trudnością mogłoby być jedynie wyszukiwanie nazw parametrów skanera które chcemy zaimplementować jednak dokumentacja jest bardzo pomocna i zrozumiała nawet dla początkujących.

Wszystkie z podstawowych założeń ćwiczenia zostały zaimplementowane oraz działały poprawnie. Z dodatkowych opcji program mógł zapisywać zeskanowane zdjęcie natomiast funkcjonalność służąca do obsługi różnych trybów przesyłania danych nie została dodana.