

Piotr Chłystek 226100
Michał Chojnacki 225936

Data oddania sprawozdania: 15.01.2018 r.
Termin zajęć: Poniedziałek 7:30-10:15, TP

Urządzenia peryferyjne

Ćwiczenie 8

Drukowanie kodów paskowych

Prowadzący:
dr inż. Jan Nikodem

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z zasadami działania kodów kreskowych i ich standardami, by w następnej kolejności napisać program generujący taki kod i umożliwiający jego wydrukowanie.

2. Wstęp teoretyczny

Kody kreskowe są graficzną formą reprezentacji informacji czytelnej dla czytników elektronicznych. Najszersze stosowane są w logistyce, jako forma identyfikacji produktów. Proces odczytu jednowymiarowego, dwukolorowego kodu przez skaner odbywa się przez naświetlenie ciągu pasków, białe elementy, czyli przerwy odbijają światło, podczas gdy czarne pochłaniają to światło. Odbite przez białe elementy światło trafia do elementu światłoczułego, które generuje silniejsze impulsy, podczas interpretacji białych elementów i słabsze podczas odczytu czarnych. Grubość pasków również ma znaczenie, ponieważ wpływa na czas trwania impulsu. Sygnały elektryczne zostają przetłumaczone na cyfry i przesłane do pamięci komputera. W ćwiczeniu zajmowaliśmy się głównie standardem EAN-13, zaimplementowanego przez GS1 (Global Standards). Pierwsze 3 cyfry kodu wskazują na organizację krajową, która przydzieliła producentowi kod. Następny jest kod producenta i kod produktu (nadawany produktowi przez producenta). Znaki w kodzie EAN-13 są kodowane na 3 sposoby, 2 sposoby przypadają na lewą stronę kodu i 1 sposób na prawą stronę. Pierwsza cyfra kodu jest zakodowana za pomocą alfabetów A i B lewej strony kodu. Kod kończy się cyfrą kontrolną, obliczaną według algorytmu modulo 10.

3. Opis programu

Program został napisany w technologii C#, z użyciem biblioteki Barcode.dll

```
private void buttonGenerate_Click(object sender, EventArgs e)
{
    BaseBarcode barcode = BarcodeFactory.GetBarcode(Symbology.EAN13);
    //deklaracja zmiennej, która będzie przetrzymać kod EAN-13
    barcode.Number = tbEan.Text;
    //numer zostanie wprowadzony przez użytkownika
    if (cbControlSum.Checked == true)
        barcode.ChecksumAdd = true;
    else barcode.ChecksumAdd = false;
    //jeżeli użytkownik wprowadził kod bez sumy kontrolnej (12 znaków),
    //program wygeneruje sumę kontrolną sam, jeżeli wprowadził sumę kontrolną
    //(13 znaków) program działa dalej
    pbBarcode.Image = barcode.Render();
    //program generuje obraz z kodem i zapisuje go do katalogu z plikami
    //projektu
    barcode.Save("barcode.png", ImageType.Png);
}

private void buttonPrint_Click(object sender, EventArgs e)
{
    //przycisk do drukowania kodu uruchamia funkcję drukującą
    PrintDocument printDocument = new PrintDocument();
    printDocument.PrintPage += PrintGeneratedBarcode;
    printDocument.Print();
}
```

```
private static void PrintGeneratedBarcode(object o, PrintPageEventArgs e)
{
    //funkcja, drukująca kod kreskowy
    System.Drawing.Image image = System.Drawing.Image.FromFile("barcode.png");
    Point loc = new Point(100, 100);
    e.Graphics.DrawImage(image, loc);
}
```

4. Wnioski

Program napisany w trakcie zajęć działa poprawnie. Jedynym zastrzeżeniem jakie możemy wskazać jest to, że program w przypadku kiedy użytkownik wprowadza 13 a nie 12 znaków, nie sprawdza czy wprowadzona przez użytkownika suma kontrolna jest poprawna. Nie zdołaliśmy się przekonać czy wydrukowany kod jest „czytelny” dla czytnika kodów kreskowych, ponieważ w trakcie zajęć pojawiły się problemy z komunikacją między czytnikiem a komputerem.