

Konrad Kieda 250997  
Dmytro Matviichuk 240545

Termin: pt TN 17:15-20:15

**Urządzenia peryferyjne**  
**Temat: Drukowanie kodów paskowych**

Kod źródłowy(GitHub ): [https://github.com/dmatviichuk/Urzadzenia\\_Peryferyjne](https://github.com/dmatviichuk/Urzadzenia_Peryferyjne)

## 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 do pliku PDF.

## 2. Wstęp teoretyczny

Kody kreskowe są graficzną formą reprezentacji informacji czytelnej dla czytników elektronicznych. Najszerszej stosowane są w logistyce, jako forma identyfikacji produktów. Proces odczytu jednowymiarowego, dwukolorowego kodu przez skaner odbywa 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 organizacji krajowej, która przydzieliła producentowi kod. Następny właśnie 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. Informacja o programie oraz fragmenty kodu

Program został napisany w technologii C#, z użyciem biblioteki Limilabs.Barcode.dll. Jako drukarkę użyliśmy "Microsoft to PDF".

### 3.1 Biblioteki które zostały użyte w naszej aplikacji:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using Limilabs.Barcode;  
using System.Drawing.Printing;
```

### 3.2 Generowanie kodu paskowego:

```
private void buttonGenerate_Click(object sender, EventArgs e)  
{  
    BaseBarcode barcode = BarcodeFactory.GetBarcode(Symbology.EAN13);  
    barcode.Number = tbEan.Text;  
    if (cbControlSum.Checked == true)  
        barcode.ChecksumAdd = true;  
    else barcode.ChecksumAdd = false;  
    pbBarCode.Image = barcode.Render();  
    barcode.Save("barcode.png", ImageType.Png);  
}
```

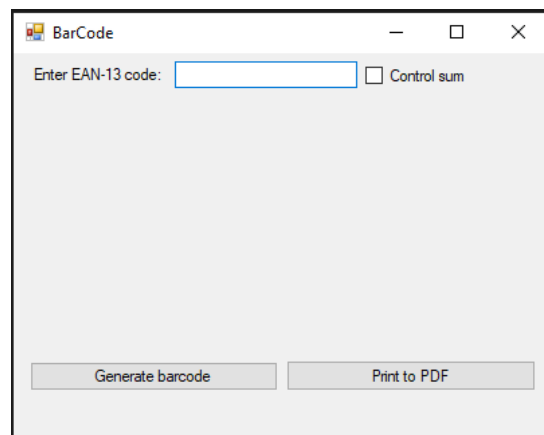
3.4 Wydruk obrazka w aplikacji:

```
private void buttonPrint_Click(object sender, EventArgs e)
{
    PrintDocument printDocument = new PrintDocument();
    printDocument.PrintPage += PrintGeneratedBarcode;
    printDocument.Print();
}

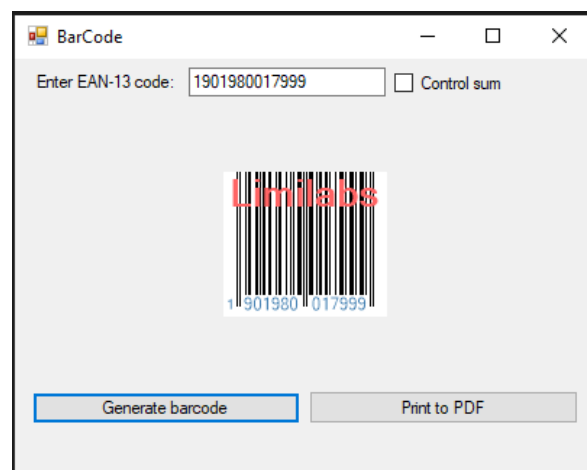
private static void PrintGeneratedBarcode(object o, PrintPageEventArgs e)
{
    System.Drawing.Image image = System.Drawing.Image.FromFile("barcode.png");
    Point loc = new Point(100, 100);
    e.Graphics.DrawImage(image, loc);
}
```

## 4. Działanie aplikacji

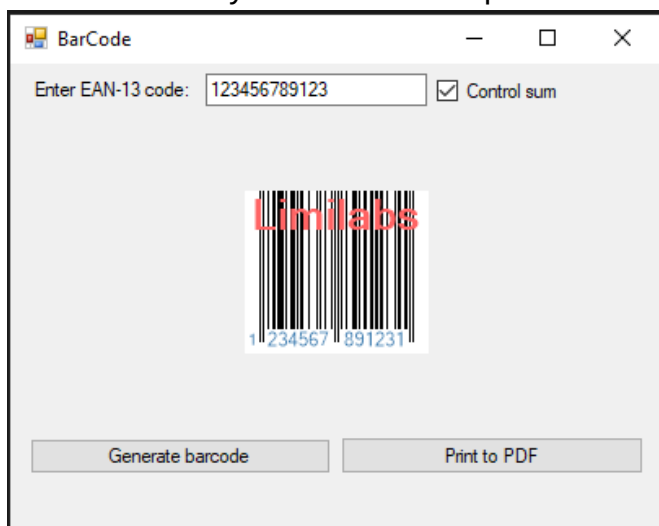
4.1 Menu główne:



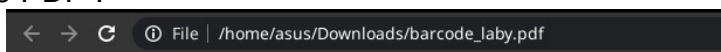
4.2 Wprowadzamy kod oraz klikamy "Generate Barcode":



4.3 Wprowadzamy kod oraz klikamy "Control sum" a potem "Generate Barcode":



4.4 Klikamy "Print to PDF":



## 5. Wnioski

Podczas laboratoriów nauczyliśmy się podstawowej obsługi drukowania kodów paskowych oraz ich generowania. Aplikacja działa poprawnie, zgodnie ze wszystkimi założeniami.