

Politechnika Wrocławska  
Wydział Elektroniki  
Urządzenia Peryferyjne

---

## DRUKOWANIE KODÓW PASKOWYCH

---

*Termin zajęć:*

WTOREK TN 7:30

*Autorzy:*

JAKUB CHMIEL 235028

TOMASZ CIEŚLAR 235652

*Prowadzący:*

dr inż. Tomasz Walkowiak

17 grudnia 2018

# Spis treści

1	Cel ćwiczenia . . . . .	2
2	Wstęp . . . . .	2
3	Założenia projektowe . . . . .	3
4	Wykorzystane narzędzia . . . . .	3
5	Implementacja programu . . . . .	3
	5.1 Interfejs aplikacji . . . . .	3
	5.2 Kod źródłowy . . . . .	4
6	Wnioski . . . . .	8
7	Bibliografia . . . . .	8

# 1 Cel ćwiczenia

1. Napisać program do drukowania kodów kreskowych w standardzie EAN13 wykorzystując drukarkę mozaikową:
  - wprowadzić kod (cyfrowo - bez ostatniej cyfry)
  - wygenerować kod jako plik graficzny
  - wydrukować cyfry kodu

# 2 Wstęp

Kod kreskowy – graficzna reprezentacja informacji poprzez kombinację ciemnych i jasnych elementów, ustaloną według symboliki reguł opisujących budowę kodu, (np. jego wymiary, zbiór kodowanych znaków, algorytm obliczania cyfry kontrolnej i inne) danego kodu. Kod kreskowy przeznaczony jest dla czytników elektronicznych. Ma na celu umożliwienie automatycznego odczytywania informacji. Głównym zastosowaniem kodu kreskowego jest automatyczna identyfikacja produktów w szeroko pojętej logistyce.

EAN (ang. European Article Number – Europejski Kod Towarowy) – rodzina kodów kreskowych (symbolika) wprowadzona w 1976 roku przez stowarzyszenie European Article Numbering. Kod został opracowany na podstawie opracowanego wcześniej w USA i Kanadzie kodu UPC. Symbolika została zaimplementowana w globalnym systemie GS1. Jest to kod ciągły, numeryczny, modularny, samosprawdzalny z dodatkową obowiązkową cyfrą kontrolną. Kod wymaga stosunkowo wysokiej precyzji wydruku, stąd nie może być stosowany na niskiej jakości papierze (np. kartonie) oraz wymaga w miarę dobrej jakości drukarek.

Kod posiada stałą długość. Stosuje się dwie wersje kodu:

- EAN-13 – zawiera 12 cyfr danych i jedną cyfrę kontrolną
- EAN-8 – zawiera 7 cyfr danych i jedną cyfrę kontrolną

W Europie symbolika ta jest powszechnie wykorzystywana do znakowania opakowań jednostkowych oraz zbiorczych (zarówno EAN-8, jak i EAN-13). Wersję EAN-13 wykorzystuje się również m.in. do kodowania numerów ISBN, ISMN czy ISSN. Kod EAN-8 przeznaczono dla małych opakowań, na których nie zmieściłby się kod EAN-13. Jednakże ze względu na wyczerpywanie się puli kodów EAN-13 przydzielonych danemu producentowi, wielu z nich stosuje indywidualnie przydzielane kody EAN-8 na dużych opakowaniach.

Przy oznaczaniu opakowań jednostkowych w kodzie EAN-13 wyróżnia się cztery grupy:

- Kod systemowy obejmujący pierwsze dwie lub trzy cyfry. Zwykle oznaczają one kod kraju (np. 590 – Polska), z wyjątkiem oznaczeń rozpoczynających się od cyfry 2 – tzw. kodów wagowych, takimi kodami oznaczane są produkty o zmiennej masie i rozmiarach, zazwyczaj konfekcjonowane w sklepach. Kod systemowy nie oznacza jednak kraju pochodzenia towaru lub przedsiębiorstwa, lecz jedynie numer organizacji krajowej, w której dany produkt jest zarejestrowany. W przypadku kodowania numerów ISSN w kodzie występuje przedrostek 977, natomiast 978 lub 979 dla ISBN (w wersji dziesięciocyfrowej) i 979 dla ISMN.

- Kod producenta składający się z czterech, pięciu lub sześciu cyfr, w zależności od długości kodu systemowego.
- Kod produktu o długości zależnej od długości kodu systemowego i kodu producenta.
- Cyfra kontrolna.

### 3 Założenia projektowe

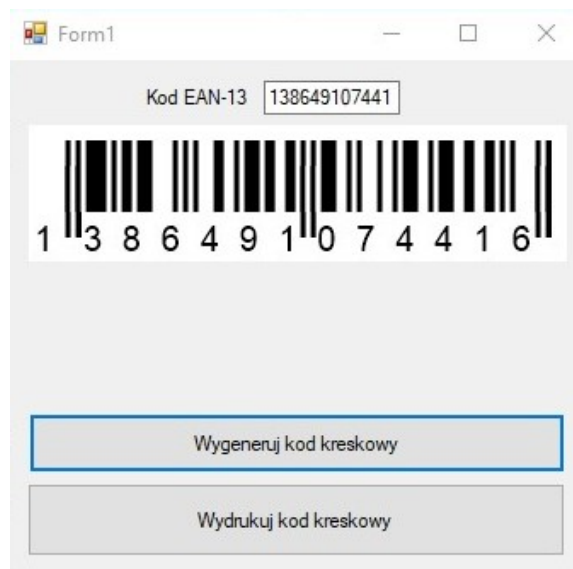
- Program był pisany w języku C#.
- Na komputerze, na którym uruchamiany był program zainstalowano system operacyjny Windows 10 w wersji 64-bitowej.
- Kod kreskowy należało wygenerować bez użycia gotowych bibliotek

### 4 Wykorzystane narzędzia

- Windows Forms - API do implementacji interfejsu graficznego dla platformy .NET.

## 5 Implementacja programu

### 5.1 Interfejs aplikacji



*Rysunek 1. Interfejs aplikacji*

## 5.2 Kod źródłowy

```
using [...]  
  
namespace UP_Kody  
{  
    public partial class Form1 : Form  
    {  
        //Tablice przechowujące schematy kodowania EAN-13  
        private string[] odd = { "0001101", "0011001",  
            "0010011", "0111101", "0100011", "0110001",  
            "0101111", "0111011", "0110111", "0001011" };  
        private string[] even = { "0100111", "0110011",  
            "0011011", "0100001", "0011101", "0111001",  
            "0000101", "0010001", "0001001", "0010111" };  
        private string[] parity = { "000000", "000001",  
            "000010", "000011", "000100", "000101",  
            "000110", "000111", "001000", "001001",  
            "001010", "001011", "001100", "001101",  
            "001110", "001111", "010000", "010001",  
            "010010", "010011", "010100", "010101",  
            "010110", "010111", "011000", "011001",  
            "011010", "011011", "011100", "011101",  
            "011110", "011111", "100000", "100001",  
            "100010", "100011", "100100", "100101",  
            "100110", "100111", "101000", "101001",  
            "101010", "101011", "101100", "101101",  
            "101110", "101111", "110000", "110001",  
            "110010", "110011", "110100", "110101",  
            "110110", "110111", "111000", "111001",  
            "111010", "111011", "111100", "111101",  
            "111110", "111111" };  
  
        public Form1()  
        {  
            InitializeComponent();  
            maskedTextBox1.Select();  
        }  
  
        private void button2_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(@"C:\Users\DELL\Desktop\myimage.jpg");  
            Point loc = new Point(100, 100);  
            e.Graphics.DrawImage(image, loc);  
        }  
  
        private void button3_Click(object sender, EventArgs e)  
        {  
            var eanCode = maskedTextBox1.Text;  
            var controlSum = 0;  
            //Wyliczanie sumy kontrolnej, na zmianę *1 i *3  
            for (var i = 0; i < eanCode.Length - 1; i += 2)  
            {
```

```

        controlSum += Int32.Parse(eanCode[i] + "");
        controlSum += 3 * Int32.Parse(eanCode[i + 1] +
            "");
    }
    //Reszta z dzielenia przez 10
    controlSum = controlSum % 10;
    controlSum = 10 - controlSum;
    if (controlSum == 10)
    {
        controlSum = 0;
    }
    //StringBuilder przechowujący 0 i 1 kodu kreskowego
    var barcode = new StringBuilder();
    //Pierwszy guard
    barcode.Append("101");
    //Przepisujemy pierwsza cyfre
    var first = Convert.ToInt32(maskedTextBox1.Text[0])
        - 48;
    var scheme = parity[first];
    //Kodujemy zgodnie z maska odd/even
    for (var i = 1; i < 7; i++)
    {
        if (scheme[i - 1] == 'o')
        {
            barcode.Append(odd[Convert.
                ToInt32(maskedTextBox1.Text[i]) - 48]);
        }
        if (scheme[i - 1] == 'e')
        {
            barcode.Append(even[Convert.
                ToInt32(maskedTextBox1.Text[i]) - 48]);
        }
    }
    //Drugi guard
    barcode.Append("01010");
    //Kodujemy kolejne cyfry zgodnie z tablica right
    for (var i = 7; i < 12; i++)
    {
        barcode.Append(right[Convert.
            ToInt32(maskedTextBox1.Text[i]) - 48]);
    }

    //Dodajemy dume kontrolna
    barcode.Append(right[controlSum]);
    //Ostatni guard
    barcode.Append("101");
    //Czcionki do grafiki
    Font barCodeF = new Font("Free 3 of 9", 45,
        FontStyle.Regular, GraphicsUnit.Pixel);
    Font plainTextF = new Font("Arial", 20,
        FontStyle.Regular, GraphicsUnit.Pixel);

```

```

Bitmap bmp = new Bitmap(1, 1);
Graphics graphics = Graphics.FromImage(bmp);

int barCodewidth =
    (int)graphics.MeasureString(maskedTextBox1.Text,
    barCodeF).Width + 100;
int barCodeHeight =
    (int)graphics.MeasureString(maskedTextBox1.Text,
    barCodeF).Height;

int plainTextWidth =
    (int)graphics.MeasureString(maskedTextBox1.Text,
    plainTextF).Width + 100;
int plainTextHeight =
    (int)graphics.MeasureString(maskedTextBox1.Text,
    plainTextF).Height;

    bmp = new Bitmap(bmp,
                      new Size(plainTextWidth,
                                barCodeHeight +
                                plainTextHeight));
}
graphics = Graphics.FromImage(bmp);

graphics.Clear(Color.White);
graphics.SmoothingMode = SmoothingMode.AntiAlias;
graphics.TextRenderingHint =
    TextRenderingHint.AntiAlias;

for (var i = 0; i < 95; i++)
{
    if (i < 3 || (i > 45 && i < 50) || i > 91)
    {
        if (barcode[i] == '0')
        {
            graphics.DrawString("|",
                                barCodeF,
                                new SolidBrush(Color.White),
                                10 + i * 3,
                                15);
        }
        else
        {
            graphics.DrawString("|",
                                barCodeF,
                                new SolidBrush(Color.Black),
                                10 + i * 3,
                                15);
        }
    }
}

```

```

        }
    }
    if (barcode[i] == '0')
    {
        graphics.DrawString("|",
            barCodeF,
            new SolidBrush(Color.White),
            10 + i * 3,
            0);
    }
    else
    {
        graphics.DrawString("|",
            barCodeF,
            new SolidBrush(Color.Black),
            10 + i * 3,
            0);
    }
}
var numbers = new StringBuilder();
numbers.Append(maskedTextBox1.Text[0] - 48);
numbers.Append(" ");
for (var i = 1; i < 7; i++)
{
    numbers.Append(Convert.ToInt32
        (maskedTextBox1.Text[i]) - 48);
    numbers.Append(" ");
}
numbers.Append(" ");
for (var i = 7; i < 12; i++)
{
    numbers.Append(Convert.ToInt32
        (maskedTextBox1.Text[i]) - 48);
    numbers.Append(" ");
}
numbers.Append(controlSum);
graphics.DrawString(Convert.ToString(numbers),
    plainTextF,
    new SolidBrush(Color.Black),
    0,
    barCodeHeight);

bmp.Save(@"C:\Users\DELL\Desktop\myimage.jpg",
    ImageFormat.Jpeg);
pictureBox1.Image = bmp;
}
}

```



## 6 Wnioski

Nie udało nam się wykonać całości zadania, ponieważ nie potrafiliśmy wydrukować kodu kreskowego, używając drukarki mozaikowej. Pomyślnie wykonaliśmy natomiast generowanie pliku graficznego kodu kreskowego.

## 7 Bibliografia

- International Article Number:  
*[https://en.wikipedia.org/wiki/International\\_Article\\_Number](https://en.wikipedia.org/wiki/International_Article_Number)*
- Encoding EAN-13:  
*<http://www.gomaro.ch/Specifications/EAN13e.htm>*