

Oskar Szubert

Paweł Biel

Poniedziałek TP, 10:15

Grupa D

Prowadzący: dr inż. Jan Nikodem

Urządzenia Peryferyjne

Ćwiczenie 15-Sterowaniem silnikiem krokowym za pomocą USB

Celem ćwiczeń nr 15 z Urządzeń Peryferyjnych było zapoznanie się z budową i działaniem (m.in. sposoby zasilania) silników krokowych - na przykładzie silnika M42SP-7.

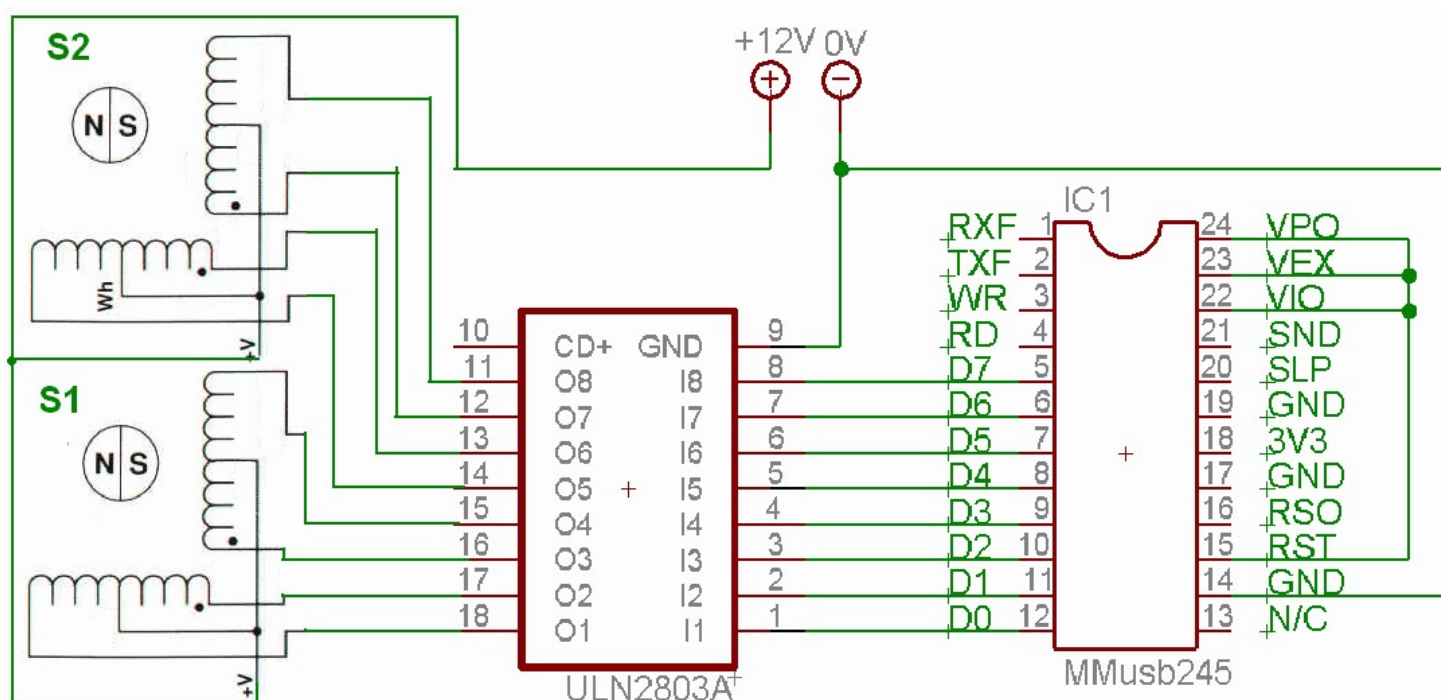
Zadaniem do wykonania w trakcie zajęć było napisanie programu kontrolującego dwa silniki unipolarne dostępne w Sali laboratoryjnej.

1. Zagadnienia teoretyczne:

Silnik krokowy – silnik elektryczny, w którym impulsowe zasilanie prądem elektrycznym powoduje, że jego wirnik nie obraca się ruchem ciągłym, lecz wykonuje za każdym razem ruch obrotowy o ściśle ustalonym kącie.

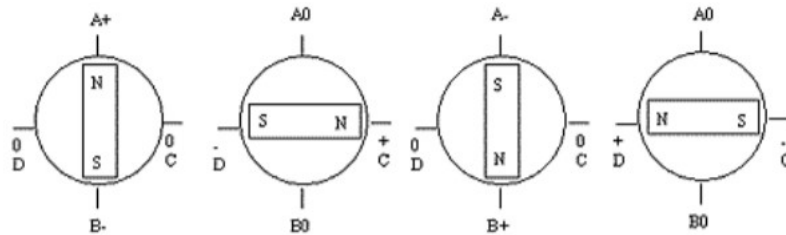
Dzięki temu kąt obrotu wirnika jest ściśle zależny od liczby dostarczonych impulsów prądowych, a prędkość kątowna wirnika jest dokładnie równa częstotliwości impulsów pomnożonej przez wartość kąta obrotu wirnika w jednym cyklu pracy silnika.

Sterowanie odbywa się za pomocą portu USB komputera PC. Za konwersję standardu USB na 8 bitową linię odpowiedzialny jest układ firmy FTDI o symbolu FT245BM. Do wyjścia tego układu, które ma postać 8 bitowej szyny dołączony jest układ ULN2803A. Jest to 8 układów Darlington'a z zanegowanym wyjściem, pozwalających sterować uzwojeniami 2 silników unipolarnych. Dla silnika M42SP-7 wartość obrotu dla jednego kroku wynosi 7.5° .



Za pomocą odpowiednich pobudzeń pinów jesteśmy w stanie sterować silnikami na trzy sposoby:

- Jednofazowo – zasilana jest tylko jedna faza w danym momencie.



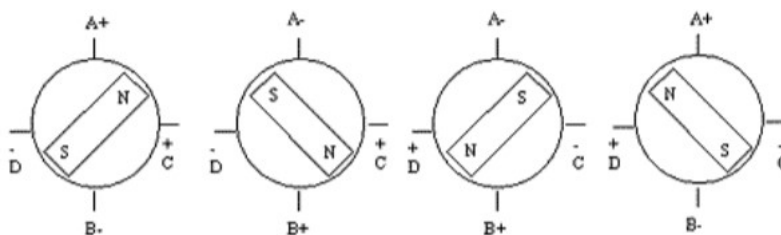
Sygnały powodujące obrót w lewo wykorzystując zasilanie jednofazowe:

{0x08, 0x02, 0x04, 0x01}* – w postaci binarnej (w takiej postaci wysyłana do sterownika), możemy zauważyć jedną jedynkę w każdej liczbie 8=1000, 2=0010, 4=0100, 1=0001. '1' powoduje więc zasilenie fazy.

*-zapis w postaci heksadecymalnej dla wygody– w naszym sprzęcie młodsze bity sterują silnikiem w pozycji poziomej(lewo, prawo), starsze w pozycji pionowej (górze, dół). Z powodu ryzyka uszkodzenia silnika poziomego(brak swobody obrotu) w programie nie był używany.

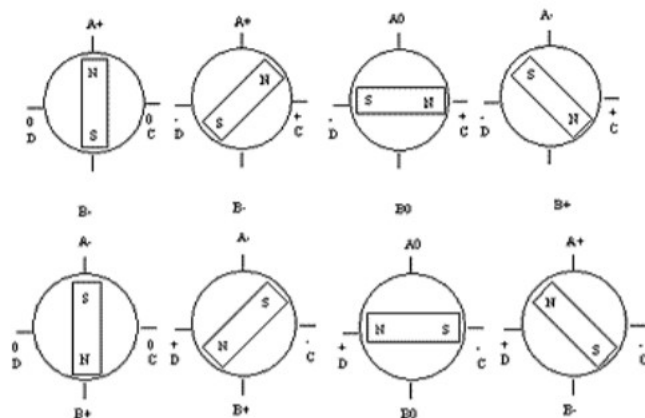
Aby wykonać obrót w drugą stronę należy podać odwrotną kombinację sygnałów, tak więc obrót w prawo to **{ 0x01, 0x04, 0x02, 0x08 } .**

- Dwufazowo – zasilenie dwóch faz w danym momencie. Pozycja wirnika jest przesunięta o pół kroku



Sygnały powodujące obrót w lewo wykorzystując zasilanie dwufazowe: **{ 0x09, 0x0A, 0x06, 0x05 }**, oraz w prawo: **{ 0x05, 0x06, 0x0A, 0x09 }** – i tak jak w punkcie wyżej możemy zważyć, że po dwie jedynki w sygnale.

- Półkrokowe –naprzemienne zasilanie jednej fazy oraz dwóch. Silnik wykonuje jedynie połowę kroku.



Sygnały powodujące obrót w :

- Lewo { 0x09, 0x08, 0x0A, 0x02, 0x06, 0x04, 0x05, 0x01 };
- Prawo { 0x01, 0x05, 0x04, 0x06, 0x02, 0x0A, 0x08, 0x09 };

Ważne jest aby po zakończeniu przesyłania sygnałów powodujących ruch przestać zasilać fazy. W tym celu należy przestać sygnał 0x00. Utrzymywanie przez dłuższy czas zasilania któregośkolwiek z uzwojeń silnika może spowodować jego uszkodzenie.

2. Kod programu:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using FTD2XX_NET;

namespace upKrokowy
{
    class Program
    {
        FTD2XX_NET.FTDI.FT_STATUS ftstatus;
        FTD2XX_NET.FTDI.FT_DEVICE_INFO_NODE[] devicelist = new
FTD2XX_NET.FTDI.FT_DEVICE_INFO_NODE[1];
        FTDI device = new FTDI();

        byte[] LeftP = { 0x08, 0x02, 0x04, 0x01 }; // jednofazowe krokowe
        byte[] RightP = { 0x01, 0x04, 0x02, 0x08 };

        byte[] LeftD = { 0x09, 0x0A, 0x06, 0x05 }; //dwufazowe krokowe
        byte[] RightD = { 0x05, 0x06, 0x0A, 0x09 };

        byte[] LeftU = { 0x09, 0x08, 0x0A, 0x02, 0x06, 0x04, 0x05, 0x01 }; // półkrokowe
        byte[] RightU = { 0x01, 0x05, 0x04, 0x06, 0x02, 0x0A, 0x08, 0x09 };

        byte[] stop = { 0x00 }; //po zakończeniu przesyłania sygnalow sterujacych

        int degree = 360; //obrot o odpowiedni kat
        int speed = 60; //szybkosc przesyłania

        public void rote(int skoki, byte[] buff)
        {
            int loops = (degree * 10) / 50 * skoki; //7.5 stopnia na skok //jezeli półkrokowe do
2*50 -> 360/7.5=50
            int steps = 4 * skoki; //wysokie na wyjsci RS232 potrzebne do wykonania obrotu, w
przypadku krokowego przesyłane jest 4 a półkrokowego przesyłanie jest 8 sygnalow
            Int32 bytesToWrite = 1;
            UInt32 bytesWritten = 0;

            byte[][] controlByte = new byte[steps][];

            for (int i = 0; i < steps; i++)
                controlByte[i] = new byte[] { buff[i] };

            for (int k = 0, i = steps - 1; k < loops; k++)
            {
                ftstatus = device.Write(controlByte[i], bytesToWrite, ref bytesWritten);
                //przesłanie odpowiedniej kombinacji bitowej do sterownika
                System.Threading.Thread.Sleep(1000 / speed);

                if (i == 0)
```

```

        i = steps - 1;
    else
        i--;
    }

    ftstatus = device.Write(stop, 1, ref bytesWritten); //zaprzestanie wysylania stanów
wysokich do silnika krokowego
}

public void Connect()
{
    ftstatus = device.GetDeviceList(devicelist);
    try
    {
        //sprawdzenie czy urzadzenie zostalo podlaczone
        ftstatus = device.OpenByDescription(devicelist[0].Description);
        ftstatus = device.SetBitMode(0xff, 1); //pozwala na przesylanie danych do ukkladu
        Console.WriteLine("Status: " + ftstatus.ToString());
    }

    catch (Exception e)
    {
        Console.WriteLine("Urzadzenie nie zostao podlaczone!\nNacisnij przycisk aby zamknac
program");
        Environment.Exit(0);
    }
}

public Program()
{
    string choise;
    Connect();
    do
    {
        Console.WriteLine("*****STEROWANIE SILNIKIEM KROKOWY*****\nSzybkosc
obrotu:");
        //speed = Console.ReadLine();
        speed = Convert.ToInt32(Console.ReadLine());
        try
        {
            Console.WriteLine("Kat obrotu:");
        }
        catch(Exception e)
        {
            Console.WriteLine("Nie podales liczby!");
        }

        try
        {
            degree = Convert.ToInt32(Console.ReadLine());
        }
        catch (Exception e)
        {
            Console.WriteLine("Nie podales liczby!");
        }
        Console.WriteLine("1 ->Lewo Krokowe-jednofazowe\n2 ->Prawo Krokowe-jednofazowe\n3 -
>Lewo polkrokowe-jednofazowe + dwufazowe\n4 -> Prawo polkrokowe-jednofazowe + dwufazowe\nInne
klawisze zakoncza dzialanie programu");
        choise = Console.ReadLine();

        switch (choise[0])
        {
            case '1':

```

```
        rote(1, LeftP);
        break;
    case '2':
        rote(1, RightP);
        break;
    case '3':
        rote(2, LeftU);
        break;
    case '4':
        rote(2, RightU);
        break;
    default:
        return;
    }
} while (true);

Console.ReadLine();
}

static void Main(string[] args)
{
    new Program();
}
}
```