

Kurs:
Urządzenia Peryferyjne - Laboratorium

STEROWANIE SILNIKIEM KROKOWYM

Autor:

ALEKSANDRA CHRUSTEK, 263900

JONASZ TROCHA, 263951

Prowadzący:
dr inż. Dariusz Caban

18.10.2023

1 Wstęp

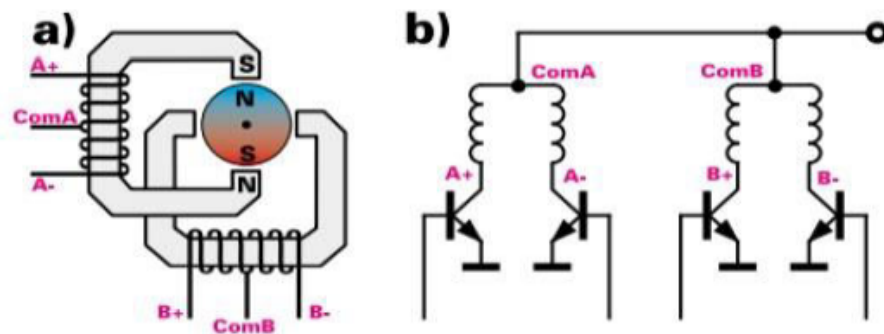
Celem ćwiczenia było zapoznanie się z zasadami działania silnika krokowego oraz napisanie oprogramowania sterującego unipolarnym silnikiem krokowym - dostępnym w sali laboratoryjnej.

2 Informacje teoretyczne

Silnik krokowy jest urządzeniem sterowanym dyskretnymi impulsami elektrycznymi. Jego sposób zasilania sprawia, że wirnik silnika jest w stanie obrócić się o dowolny kąt. Stąd kąt obrotu wirnika zależy od liczby impulsów elektrycznych dostarczanych do silnika. Sterowanie silnikiem krokowym odbywa się poprzez bezpośrednie podłączenie - kablem od silnika do interfejsu USB komputera osobistego. Silniki krokowe dzielimy ze względu na sposób sterowania, liczbę uzwojeń czy rodzaj materiału, z którego są wykonane. Trzy większe grupy to:

1. Silniki z magnesem stałym oraz hybrydowe.
2. Silniki o zmiennej reluktancji (VR).
3. Silniki bipolarne i unipolarne – silniki z dzielonymi uzwojeniami.

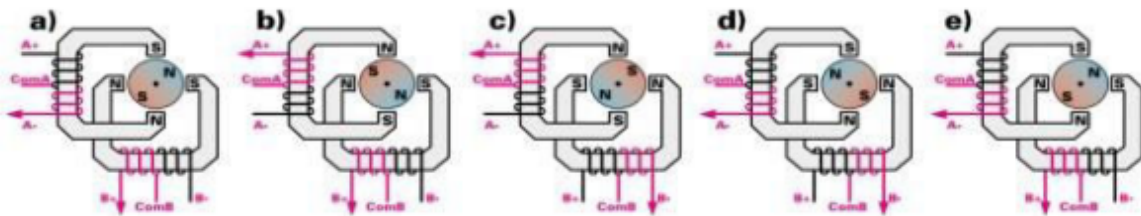
Na zajęciach pracowaliśmy z silnikiem unipolarnym. Jest to silnik, który posiada uzwojenia z odczepem, który pozwala na łatwiejszy sposób sterowania, a więc łatwiejszy sposób implementacji oprogramowania (sterownika).



Rysunek 1: Schemat silnika unipolarnego

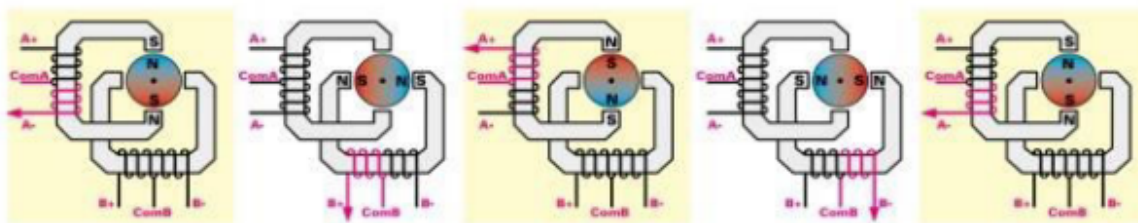
Tryby pracy silnika możemy podzielić na trzy przedstawione poniżej.

1. Tryb pełnokrokowy: uzwojenia są zasilane parami w odpowiedniej sekwencji.



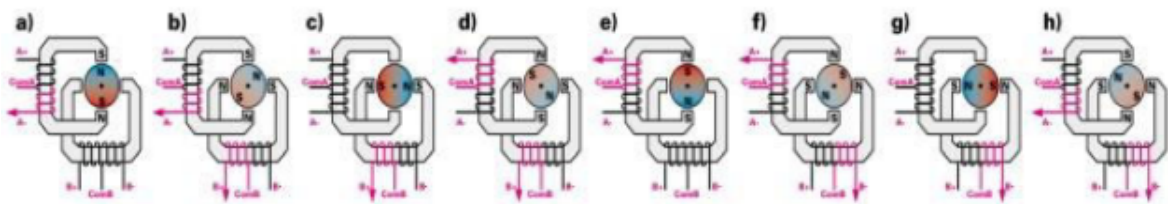
Rysunek 2: Tryb pełnokrokowy

2. Tryb falowy: w każdej fazie zasilania pracuje jedynie 25% wszystkich uzwojeń - jest to szczególny przypadek trybu pełnokrokowego.



Rysunek 3: Tryb falowy

3. Tryb półkrokowy: połączenie trybu falowego i pełnokrokowego. Naprzemiennie zasilane jest jedno oraz dwa uzwojenia. Taka sekwencja zasilania sprawia, że bieguny wirnika ustawiają się przeciwnie do biegunów stojana lub w połowie między tymi biegunami.



Rysunek 4: Tryb półkrokowy

3 Realizacja zadania

Silnik unipolarny składa się z dwóch uzwojeń. Dla pracy falowej mamy najmniej przypadków zasilania uzwojeń silnika. Poniższa tabela przedstawia odpowiednie sekwencje bitowe dostarczane do uzwojeń silnika:

1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

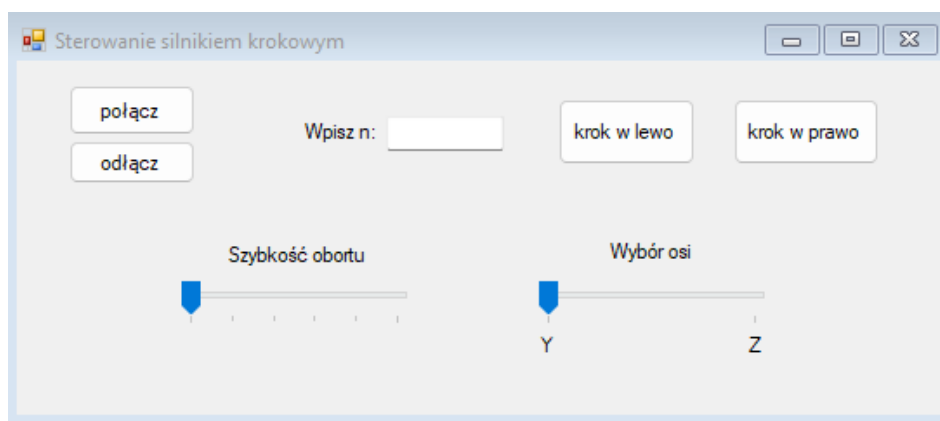
W programie reprezentacja tych liczb binarnych została zapisana do tablicy bajtowej i przyjmuje ona postać:

```
byte[] LeftP = { 0x08, 0x02, 0x04, 0x01 };  
byte[] RightP = { 0x01, 0x04, 0x02, 0x08 };
```

Rysunek 5: Tablica bajtowa przechowująca reprezentację liczb binarnych

Jak widać, aby obracać wirnikiem w drugą stronę, wystarczy zasilać uzwojenia elementami tablicy w odwrotnej kolejności.

W celu stworzenia oprogramowania sterującego silnikiem zostały zaimplementowane wyżej przedstawione tablice bajtowe. Komunikacja silnika przez interfejs USB została rozwiązana przy pomocy biblioteki FTD2XX_NET. Poniżej widoczny jest interfejs programu:



Rysunek 6: Interfejs programu

Przyciski połącz oraz odłącz służą do nawiązania lub zerwania połączenia z silnikiem. Obok widoczne jest pole do wpisania liczby n, która jest ilością kroków do wykonania,

które następnie wykonujemy klikając jeden z przycisków "krok w lewo" lub "krok w prawo" w zależności od kierunku w jakim chcemy, aby krok został wykonany. Dodatkowo niżej mamy możliwość wybrania osi, po której będzie poruszał się silnik oraz szybkość kroku.

Poniżej przedstawione zostały poszczególne metody do obsługi silnika krokowego:

1. Metoda `enablebtn_Click()` jest wywoływana w przypadku kliknięcia przycisku połącz - nawiązanie połączenia silnika krokowego z komputerem.

```
private void enablebtn_Click(object sender, EventArgs e) {
    try{
        UInt32 ftdiDeviceCount = 0;
        device.GetNumberOfDevices(ref ftdiDeviceCount);
        FTD2XX_NET.FTDI.FT_DEVICE_INFO_NODE[] devicelist =
            new FTD2XX_NET.FTDI.FT_DEVICE_INFO_NODE[ftdiDeviceCount];
        device.GetDeviceList(devicelist);

        ftstatus = device.OpenBySerialNumber(devicelist[0].SerialNumber);
        ftstatus = device.SetBitMode(0xff, 1);
        Console.WriteLine("Urządzenie: " + ftstatus.ToString());
        stepLeftbtn.Enabled = true;
        stepRightbtn.Enabled = true;
    }
    catch (Exception ee){
        Console.WriteLine("Urządzenie nie zostało podłączone!
        \nNacisnij [ENTER] aby zamknąć program");
        Console.ReadLine();
        Environment.Exit(0); }
}
```

2. Metoda `disablebtn_Click()` jest wywoływana w przypadku kliknięcia przycisku odłącz - zerwanie połączenia silnika krokowego z komputerem.

```
private void disablebtn_Click(object sender, EventArgs e){
    stepLeftbtn.Enabled = false;
    stepRightbtn.Enabled = false;
    ftstatus = device.Close();
}
```

3. Metoda `stepLeftbtn_Click` jest wywoływana, gdy zostanie wciśnięty przycisk kroku w lewo.

```

private void stepLeftbtn_Click(object sender, EventArgs e){
    int value;

    if(textBox1.Text == ""){
        value = 0;}
    else{
        value = int.Parse(textBox1.Text);
    }

    Console.WriteLine("Wartosc n: " + value);
    rotate(value, -1);
}

```

4. Metoda stepRightbtn_Click jest wywoływana, gdy zostanie wciśnięty przycisk kroku w prawo.

```

private void stepRightbtn_Click(object sender, EventArgs e) {
    int value;

    if (textBox1.Text == ""){
        value = 0;}
    else{
        value = int.Parse(textBox1.Text);
    }

    Console.WriteLine("Wartosc n: " + value);
    rotate(value, 1);
}

```

5. Metoda rotate jest wywoływana w obu funkcjach stepLeftbtn_Click oraz stepRightbtn_Click.

```

private void rotate(int count, int direction) {
    numBytesWritten = 0;
    Int32 bytesToWrite = 1;

    for (int i = 0; i < count; i++){
        index += direction;
        if (index < 0){
            index = 3;}
        else if (index > 3){

```

```

        index = 0;}
if (trackBar2.Value == 1){
    byte[] y = { (byte)(LeftP[index] << 4) };
    device.Write(y, bytesToWrite, ref numBytesWritten);}
else{
    byte[] x = { LeftP[index] };
    device.Write(x, bytesToWrite, ref numBytesWritten);}

Thread.Sleep(speed - trackBar1.Value * 28);}
}

```

4 Wnioski

Zadanie miało na celu zapoznanie z programowaniem urządzeń peryferyjnych. W tym przypadku był to silnik krokowy. Realizowane zadanie było trudne, ponieważ wymagało zaznajomienia się z dokumentacją układu FT245BM składającego się z układu Mmusb245 oraz ULN2803. Wymagało także zrozumienia samej idei pracy silnika krokowego. Udało się to dzięki dokumentacji zawartej w instrukcji laboratoryjnej. Laboratorium przebiegło pomyślnie, a metody przez nas zaimplementowane zadziałały poprawnie.

Umieszczenie pomyślnie 18.10.2023 Janasz Trach
zadanie 4 - silnik krokowy - raport Aleksandra
grupa 6B Anesteti

Kolejne etapy:

1. Otworzenie aplikacji otwierającej iamy-
tącej urządzenie. (płytek potężer/możesz).
2. Dodanie pomyślnie kroków w lewo/w prawo.
3. Dodanie okna do wpisania ilości kroków
przez użytkownika ("wpisz n").
4. Dodanie suwaka do wyboru szybkości (czas
trwania) kroku.
5. Dodanie suwaka do wyboru osi poruszanie.
6. Próba implementacji sposobu sterowania
pełnokrokowego i półkrokowego.

pełnowe Left $\{0 \times 08, 0 \times 02, 0 \times 04, 0 \times 01\}$
Right $\{0 \times 01, 0 \times 04, 0 \times 02, 0 \times 08\}$

pełnokrokowe Left $\{0 \times 06, 0 \times 0A, 0 \times 09, 0 \times 05\}$
Right $\{0 \times 05, 0 \times 09, 0 \times 0A, 0 \times 06\}$

Wszystkie przesyłane są na magistralę w
odpowiednich sekwencjach, w celu konmu-
tacji w silniku. Aby spowodować ruch
silnika w drugą stronę, należy aktywować sekwencję

Wykonano jedynie sterowanie
pełnokrokowe.

18.10.2023