



**KATEDRA AUTOMATYKI,
BIOMECHANIKII MECHATRONIKI**



Laboratorium
Mikroautomatyki

TEMAT:

Sterowanie silnikami krokowymi 2

Aktualizacja: 23.02.2022

Opracował: mgr Maksymilian Bednarek

wersja: 1.0

Wstęp

Silniki krokowe, w odróżnieniu od silników innych silników elektrycznych wykonujących ruch obrotowy, są silnikami bezszczotkowymi. Nie ma w nich elementów mechaniczno-elektrycznych współpracujących z komutatorem w celu wymuszenia przepływu prądu w odpowiednich częściach uzwojenia wirnika.

W silniku krokowym wirnik nie obraca się ruchem ciągłym, ale wykonuje ruch obrotowy za każdym razem o ściśle określony kąt na jeden krok. Ilość kroków na jeden obrót zależy jest od rodzaju silnika (ilości cewek) oraz sterownika.

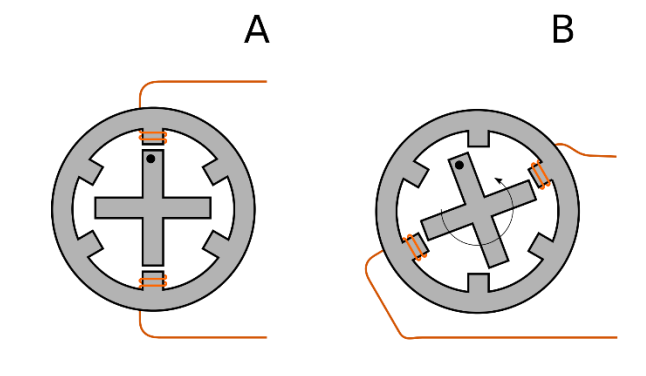
Silniki krokowe umożliwiają pracę z częstotliwością taktowania do 40 000 kroków na sekundę. Przy tak dużej prędkości taktowania kroków, ze względu na wirującą masę nie można zatrzymać wirnika w przerwach między skokami bez obawy zgubienia jakiegoś kroku.

Z uwagi na precyzję pozycjonowania, wysoki moment obrotowy przy niskich prędkościach używany jest w aplikacjach nadzorujących precyzję ruchu w drukarkach, frezarkach, układach robotycznych itp.

Wyróżniamy kilka rodzajów silników krokowych:

1) Silnik o zmiennej reluktancji VR (variable reluctance)

Wirnik wykonany jest z miękkiej stali i na swoim obwodzie posiada charakterystyczne zęby. Przepływ prądu stałego przez wybrane uzwojenie powoduje, że zęby wirnika ustawiają się naprzeciwko zasilanego uzwojenia, tak, aby zmniejszyć do minimum opór dla strumienia magnetycznego (reluktancji). Praca silnika polega na cyklicznym załączaniu odpowiednich uzwojeń stojana które wytwarzając pole magnetyczne wymusza takie ustawienia wirnika, aby ten zapewnił jak najmniejszą reluktancję dla strumienia stojana. Najmniejsza wartość kąta o jaki obróci się wirnik przy sterowaniu całym krokiem jest równa różnicy kątów rozstawienia zębów wirnika a rozstawem biegunów stojana.

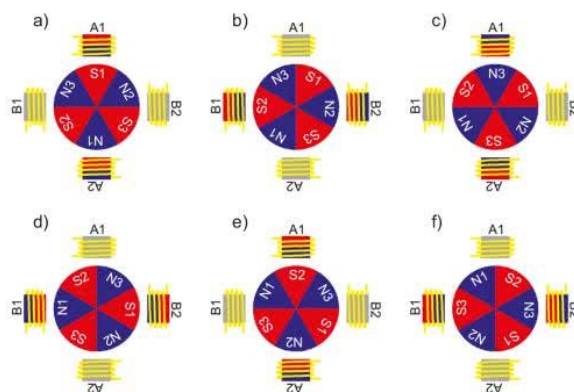


rys. 1 Schemat silnika VR

źródło: https://pl.wikipedia.org/wiki/Silnik_krokowy#/media/Plik:Stepper_motor.svg

2) Silnik z magnesem stałym PM (permanent magnet)

U podstaw działania silnika leży wzajemne oddziaływanie na siebie biegunów magnetycznych. Wirnik tego silnika ma kształt walca naprzemiennie namagnesowanego biegunami N i S. Stojan silnika posiada cewki pełniące rolę elektromagnesów. Aby wprowadzić wirnik w ruch obrotowy elektromagnesy stojana muszą wytwarzać pole magnetyczne o zmieniającym się kierunku. Wówczas na nabiegunkach pojawiać się będą na przemian bieguny N i S. Wymaga to nie tylko odpowiedniej sekwencji załączania cewek, ale również zmian kierunków prądu które przez nią przepływają.

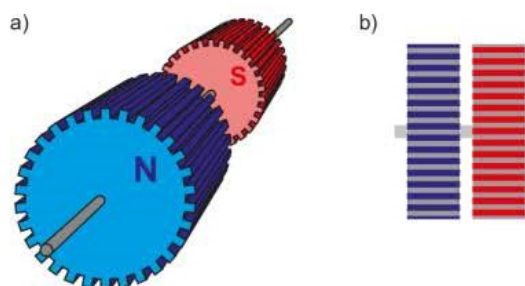


rys. 2 Schemat silnika PM

źródło: <https://ep.com.pl/rynek/wybor-konstruktor/13468-silniki-krokowe-i-serwonapedy>

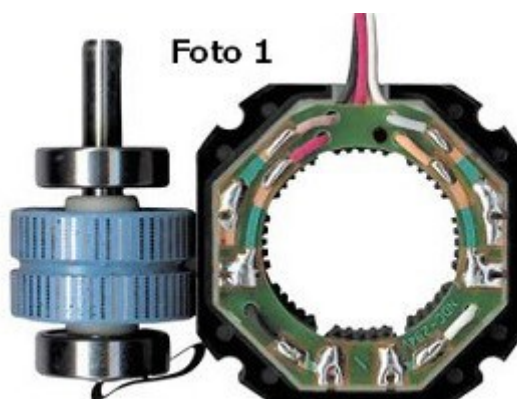
3) Silnik hybrydowy HB

Wirnik silnika hybrydowego podzielony jest na dwie osiowo (a nie promieniowo jak w przypadku silnika PM) namagnesowane części (bieguny magnesów są ustawione równoległe do osi wirnika). Boczne powierzchnie obu magnesów posiadają dużą liczbę zębów, przy czym zęby obu części są przesunięte względem siebie tak, że zęby części o polaryzacji N leżą naprzeciwko zębów o polaryzacji S.



rys. 3 Schemat namagnesowania wirnika w silniku HB

źródło: <https://ep.com.pl/rynek/wybor-konstruktor/13468-silniki-krokowe-i-serwonapedy>



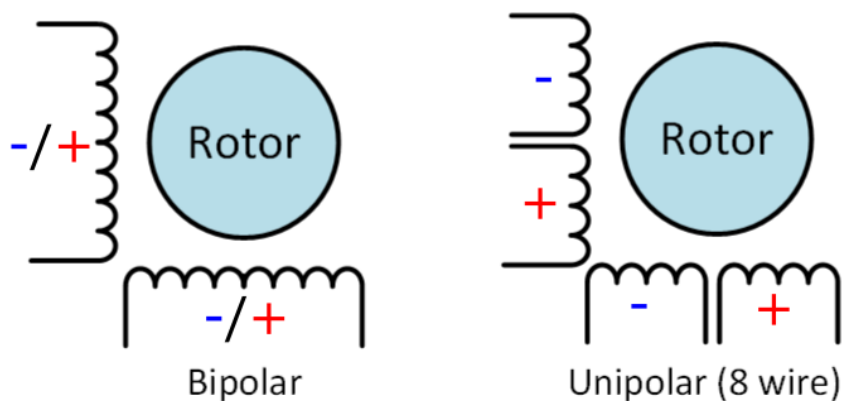
rys. 4 Zdjęcie rozmontowanego silnika HB

źródło: <https://tromil.pl/silniki-krokowe>

Połączenie cewek

Silniki krokowe wymagają do swojego działania sterowników które dostarczają energię do pasm uzwojeń silnika. Silnik krokowy może posiadać dowolną ilość cewek. Jednakże są one połączone w grupy nazywane "fazami". Wszystkie cewki w fazie są zasilane razem.

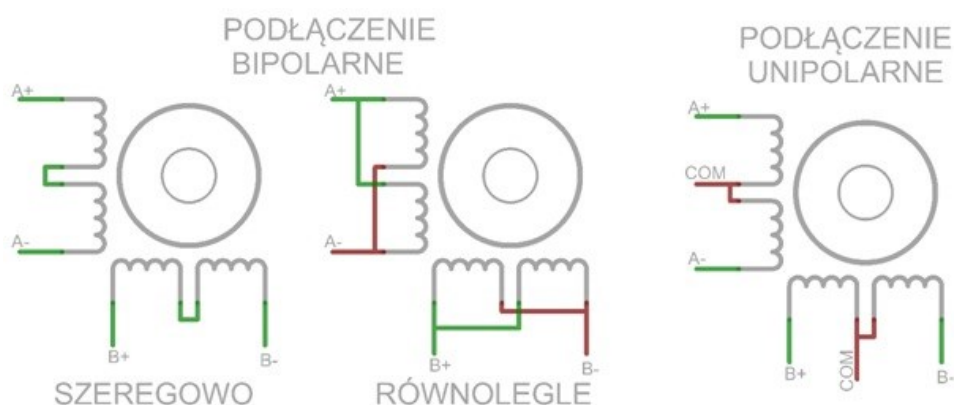
Sterownik unipolarny (silnik unipolarny) zawsze zasila fazy w ten sam sposób. Jeden "wspólny" przewód będzie zawsze z ładunkiem negatywnym. Drugi, będzie z ładunkiem pozytywnym. Sterowniki te mogą być wdrażane za pomocą prostych obwodów tranzystorowych. Jego wadą jest to, że posiada on mniej momentów obrotowych, ponieważ tylko połowa cewek może być zasilana jednocześnie.



rys. 5 Schemat połączenia cewek w silniku bipolarnym i unipolarnym

Sterownik bipolarny (silnik bipolarny) korzysta z układu mostka H, aby móc odwrócić przepływ prądu używając faz. Zasilając je naprzemienną polaryzacją, wszystkie cewki zaczną pracować, jednocześnie włączając silnik.

Silnikami z 4 wyprowadzeniami możemy sterować tylko bipolarnie, natomiast o tym jak sterować silnikiem z 6 lub 8 wyprowadzeniami możemy już zdecydować sami (rysi. 6). Silnik krokowy możemy podłączyć szeregowo, gdzie musimy go zasilić wyższym napięciem, ale przy tym będzie pobierał mniejszy prąd, co się przełoży na spadek mocy przy większych prędkościach. Możemy również jego podłączyć równolegle i zasiląć jego niższym napięciem, ale za to wyższym prądem. Co da nam mniejsze spadki mocy przy wyższych prędkościach. Przy niskich prędkościach zarówno przy podłączeniu szeregowym jak i równoległym silnik będzie miał ten sam moment.

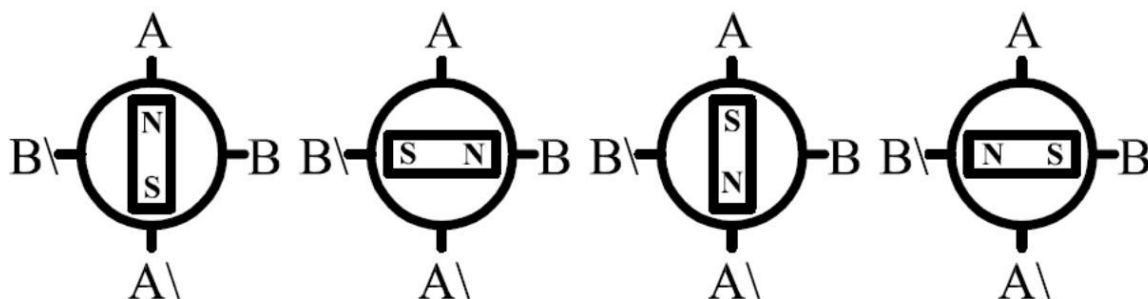


rys. 6 Schemat łączenia wyprowadzeń cewek

Podział sterowania uwzględniający rodzaj kroku wykonywanego przez silnik

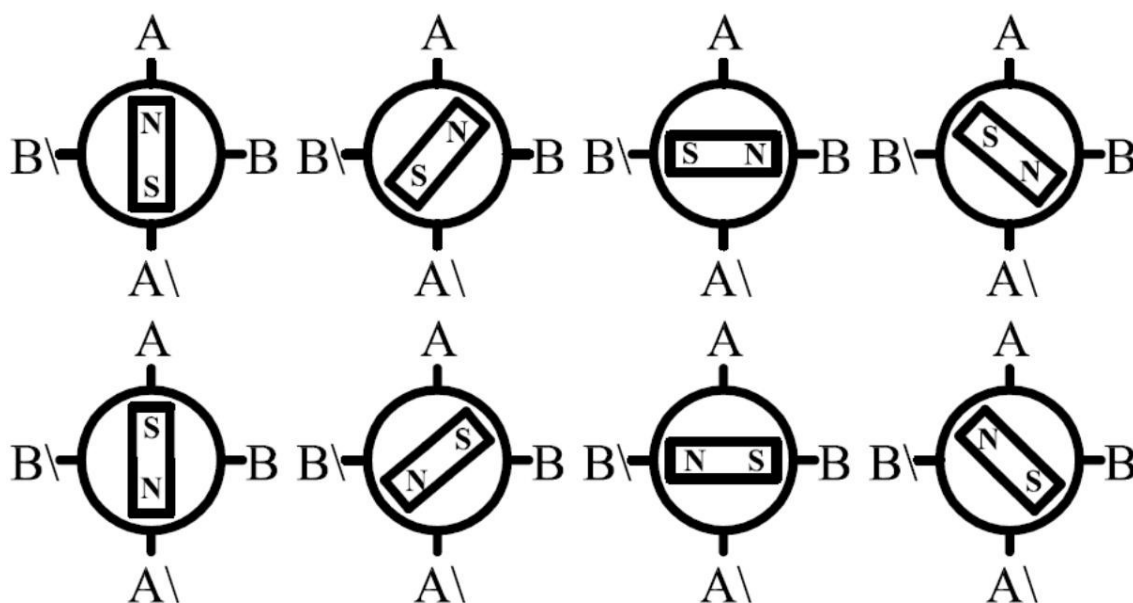
Sterowanie mikrokrokami polega na przełączaniu uzwojeń w taki sposób, aby ich pola elektromagnetyczne częściowo się znosiły lub wzmacniały. Powoduje to wyznaczenie równowagi dla wirnika w pośredniej pozycji pomiędzy skrajnymi sąsiednimi położeniami, w których pola te działają z pełną siłą.

Zastosowanie mikrokroku zwiększa liczbę kroków silnika na jeden obrót, co wpływa na zwiększenie rozdzielczości pozycjonowania oraz poprawienie jakości pracy (przez zmniejszenie wpływu rezonansów silnika), natomiast może zmniejszać maksymalną prędkość.



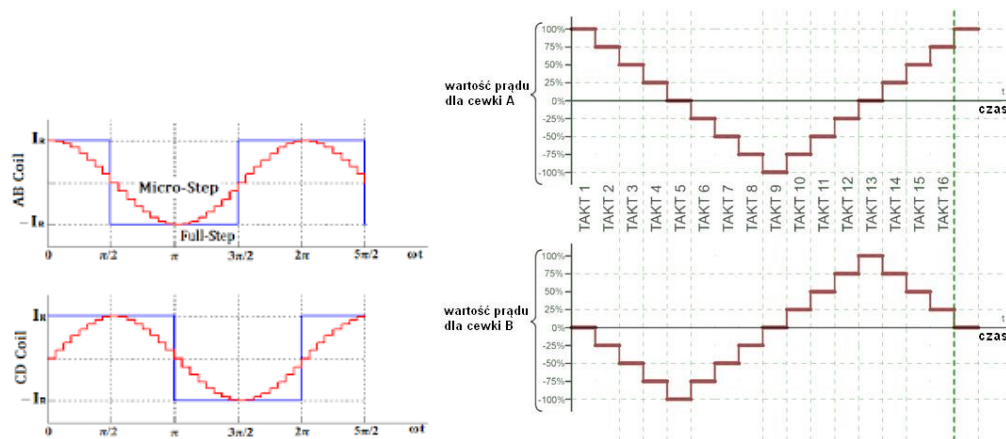
rys. 7 Schemat pracy w trybie pełnokrokowym

źródło: <https://www.ebmia.pl/wiedza/porady/budowa-i-sterowanie-maszyn-cnc/silnik-krokowy-budowa-dzialanie-zastosowanie/>



rys. 8 Schemat pracy w trybie półkrokowym

źródło: <https://www.ebmia.pl/wiedza/porady/budowa-i-sterowanie-maszyn-cnc/silnik-krokowy-budowa-dzialanie-zastosowanie/>



rys. 9 Schemat podziału prądu sterującego na mikrokroki

źródło: Variable Rate Based Microstepping of Stepper Motor Using MATLAB GUI, D.Gaan, M.Kumar, C.Majumder, IEEE Xplore, 2017(lewy obrazek), <http://silniki-krokowe.com.pl/wp-content/uploads/2017/01/16.jpg> (prawy obrazek)

Obliczanie prędkości silnika w zależności od sterowania

Znając jaką ilość kroków na obrót ma konkretny silnik oraz częstotliwość impulsów podawaną na sterownik możemy w bardzo prosty sposób obliczyć ilość obrotów na minutę czy sekundę.

$$d = \frac{a}{b \cdot \frac{1}{c}}$$

Gdzie:

a – częstotliwość impulsów sterujących (np. 2000 Hz),

b – ilość kroków na obrót silnika (najczęściej $200 \frac{\text{kroków}}{\text{obrot}}$),

c – wielkość kroku (np. $1, \frac{1}{2}, \frac{1}{8}$),

d – ilość obrotów na sekundę.

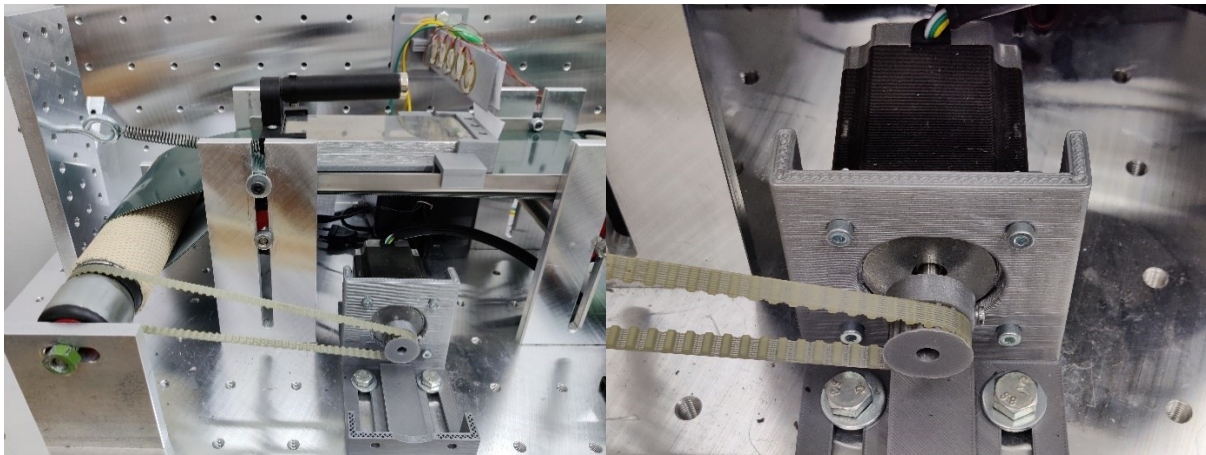
Laboratorium

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się ze sposobami sterowania silnikiem krokowym.

Przedstawione zostaną 2 sposoby sterowania silnikiem krokowym poprzez:

- Arduino
- kartę pomiarową NI (labview) + generator



rys. 10 Zdjęcie stanowiska laboratoryjnego oraz zbliżenie na silnik.

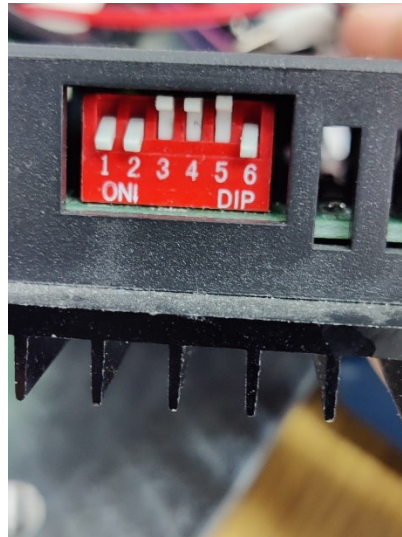
W ćwiczeniu wykorzystamy stanowisko do badania dynamiki masy na taśmie pokazane na rys. 9. będziemy posługiwać się silnikiem krokowym bipolarnym oraz sterownikiem TB6600. Aby sterować prędkością oraz kierunkiem obrotów silnika należy wysyłać odpowiednie sygnały na sterownik. Jak możemy zobaczyć na rys. 10 wejścia i wyjścia sterownika są podzielone na dwie części: „Signal” oraz „High Voltage”. W części „High Voltage” podłączamy zasilanie silników oraz wyjście zasilające uzwojenia silnika. W części „Signal” znajdują się wejścia sterujące. Możemy wykorzystać generator, kartę pomiarową NI, Arduino bądź inne urządzenia do wysyłania odpowiednich sygnałów logicznych do sterowania silnikiem. Aby ustawić kierunek obrotu silnika należy na zaciski DIR podłączyć stałe napięcie wysokie (obrót w lewo) lub niskie (obrót w prawo). Chcąc wprawić wirnik silnika w ruch należy impulsowo załączać oraz wyłączać napięcie na zaciskach PUL. Doskonale do tego nada się sygnał prostokątny wysyłany np. przez generator.



rys. 11 Zdjęcie sterownika silnika krokowego TB6600

Zmiana ustawień kroku

Możemy zmienić wielkość kroku, jeśli chcemy np. zwiększyć precyzję pozycjonowania. Odbywa się to poprzez ustawienie odpowiednich przełączników ukazanych na rys. 12 oraz odczytaniu legendy z wierzchniej warstwy sterownika (rys. 11). Przełącznik ustawiony w pozycji dolnej jest włączony (ON). Zgodnie z legendą na rys. 12 ustawione jest sterowanie pełnokrokowe.



Rys.12 Zdjęcie przełączników do ustawiania sterowania krokowego oraz natężenia prądu

Sterowanie poprzez Arduino

Jednym z najtańszych, najprostszych i najszybszych rozwiązań do uruchomienia silnika krokowego jest podłączenie go do Arduino. Bez dodatkowych modułów nie będziemy w stanie sterować prędkością silnika w czasie działania programu, natomiast w prosty sposób nastawimy pewną stałą prędkość oraz funkcję rosnąco-malejącą. Na początku musimy połączyć wyjścia odpowiednich kanałów Arduino ze sterownikiem (patrz rys. 13). Następnie tworzymy prosty program, w którym musimy zadbać o to, aby zdefiniować których wyjść będziemy używać,

```
#define dirPin 2
```

```
#define stepPin 3
```

```
pinMode(stepPin, OUTPUT);
```

```
pinMode(dirPin, OUTPUT);
```

ustawić na stałe kierunek obrotu,

```
digitalWrite(dirPin, HIGH);
```

oraz w pętli załączać na zmianę stan wysoki i niski co pewien okres.

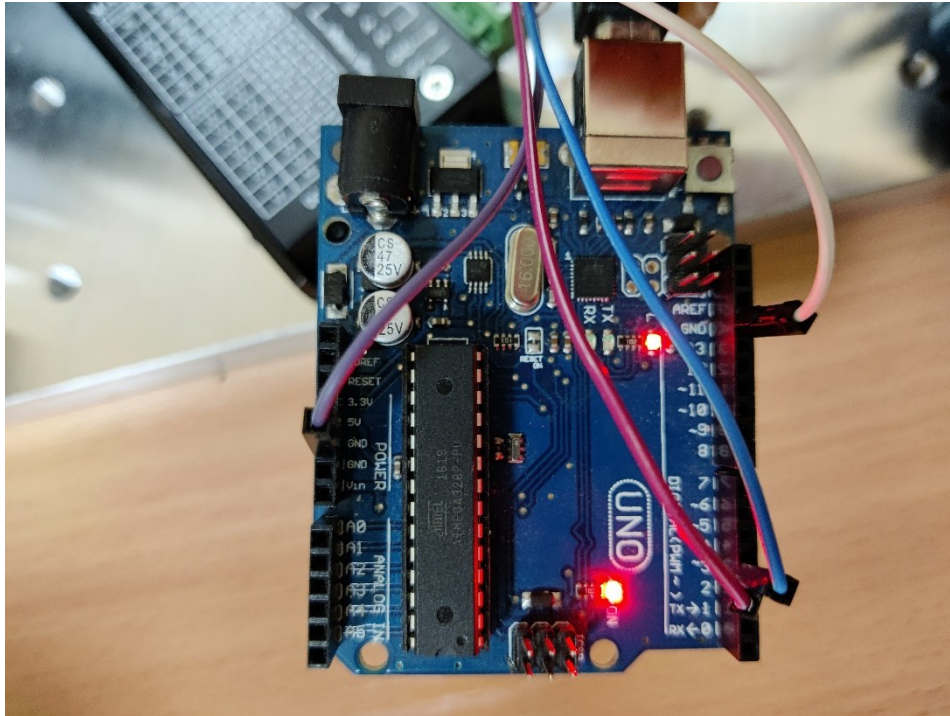
```
digitalWrite(stepPin, HIGH);
```

```
delayMicroseconds(500);
```

```
digitalWrite(stepPin, LOW);
```

```
delayMicroseconds(500);
```


Aby wgrać utworzony program na płytkę należy podłączyć ją do komputera za pomocą przewodu USB, wejść w narzędzia -> Port -> wybrać odpowiedni port (np. COM3) a następnie przyciski „zweryfikuj” i „wgraj” (patrz rys.15).



Rys.13 Zdjęcie połączeń przewodów pomiędzy Arduino i sterownikiem silnika krokowego

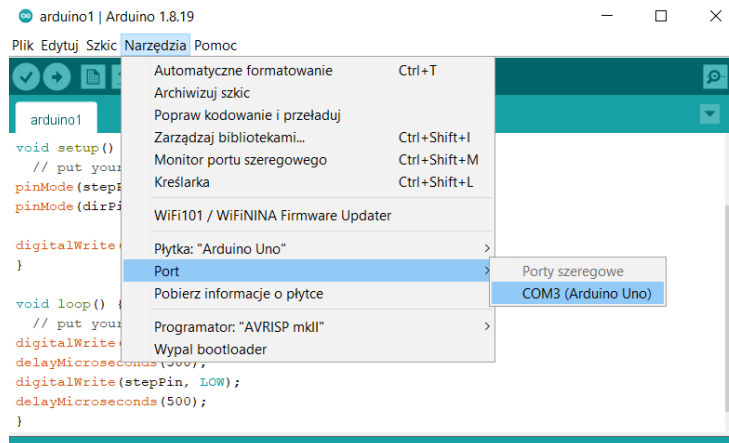
```
arduino1 §  arduino1-LAPTOP-VSN1LFDJ
#define dirPin 2
#define stepPin 3

void setup() {
    // put your setup code here, to run once:
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);

    digitalWrite(dirPin, HIGH);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
```

Rys.14 Kod źródłowy programu do wysyłania sygnału napięciowego prostokątnego



Rys.15 Ustawianie portu, na który ma zostać wgrany program.

Ustawianie silnika od razu na wysokie obroty może powodować przeskoki kroków bądź nawet uszkodzenie silnika. Zmodyfikujmy powyższy kod w ten sposób, aby prędkość płynnie się zwiększała, a następnie malała po osiągnięciu pewnej maksymalnej wartości. Rys. 16 pokazuje implementację takiego zachowania. Definiujemy pewną wartość kroków (w tym przypadku 1600) która posłuży nam do kontroli ilości powtórzeń pętli. Pętle są dwie. W pierwszej przyspieszamy silnik poprzez zmniejszanie odstępów czasowych pomiędzy zboczami sygnału prostokątnego. W drugiej sytuacja jest odwrotna – zwiększamy odstęp pomiędzy załączeniami stanów wysokiego i niskiego. Po zakończeniu cyklu sprawdzamy czy w którą stronę obracał się silnik i zmieniamy jego kierunek na przeciwny.

```

arduino2$
#define dirPin 2
#define stepPin 3
#define steps 1600

int stan = LOW;
void setup() {

    // put your setup code here, to run once:
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);

    digitalWrite(dirPin, LOW);
}

void loop() {
    // put your main code here, to run repeatedly:

    for(int i = steps; i>0;i--)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(200 +i);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(200+i);
    }

    for(int i = 0; i<steps;i++)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(200 +i);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(200+i);
    }

    if (stan == HIGH)
    {
        digitalWrite(dirPin, LOW);
        stan =LOW;
    }
    else
    {
        digitalWrite(dirPin, HIGH);
        stan = HIGH;
    }
    delay(1000);
}

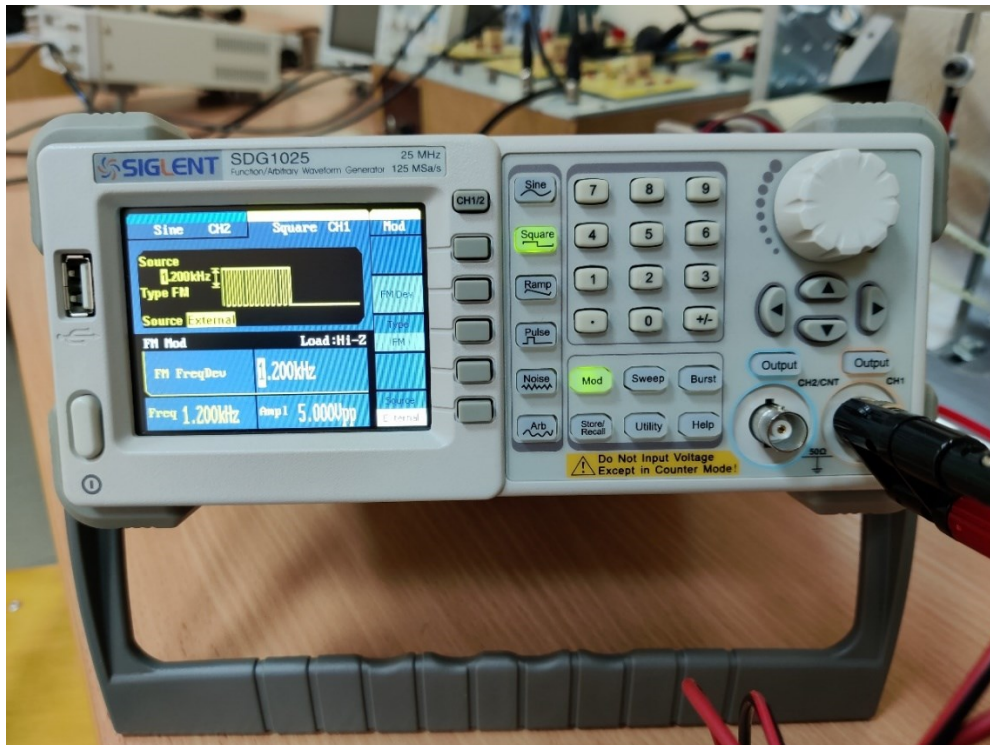
```

Rys.16 Kod źródłowy programu, który przyspiesza i zwalnia obroty silnika.

Sterowanie poprzez Labview + generator

Generator Siglent SDG1025 jest w stanie nie tylko wysłać stałą zadaną częstotliwość impulsów, ale również na podstawie zewnętrznego sygnału napięciowego zmieniać częstotliwość podczas działania urządzenia. Użyjemy tej właściwości, aby płynnie zmieniać prędkość silnika za pomocą suwaka w utworzonym programie Labview. Na sterowniku stawimy wartość środkową żądanej częstotliwości oraz przesunięcie, które będzie ustalało maksymalną i minimalną częstotliwość możliwą do zmiany. Przykładowo, jeśli chcemy ustawić zakres częstotliwości od 500 Hz do 1500 Hz, to nastawiamy na początku 1000 Hz oraz przesunięcie 500 Hz. Wartości graniczne będą odpowiadały odpowiednio wartościom napięcia 5 oraz -5 V. Przekazując na generator napięcie -5 V uzyskamy maksymalną częstotliwość z nastawionego zakresu. Analogicznie, ustawiając wartość 5 V uzyskamy wartość minimalną. Zmieniając napięcie możemy w płynny sposób sterować częstotliwością silnika. Szczegółowe informacje na temat generatora można znaleźć w dołączonej instrukcji. Ustawienie trybu opisanego trybu jest przedstawione poniżej.

Należy wybrać sygnał prostokątny, ustawić połowę maksymalnej częstotliwości, której będziemy używać. Następnie wejść w tryb „Mod”, wybrać Shape – square, Type – Fm oraz Source – External. W polu FM FreqDev ustawiamy taką samą częstotliwość jak na początku w sygnale prostokątnym. Przykładowo, ustawiając „Freq” na 1000 Hz oraz FM FreqDev również na 1000 Hz uzyskujemy zakres od 0 do 2000 Hz. Na rys. 16 i 17 możemy zobaczyć, jak powinien wyglądać poprawnie ustawiony interfejs generatora oraz podłączenie przewodów.

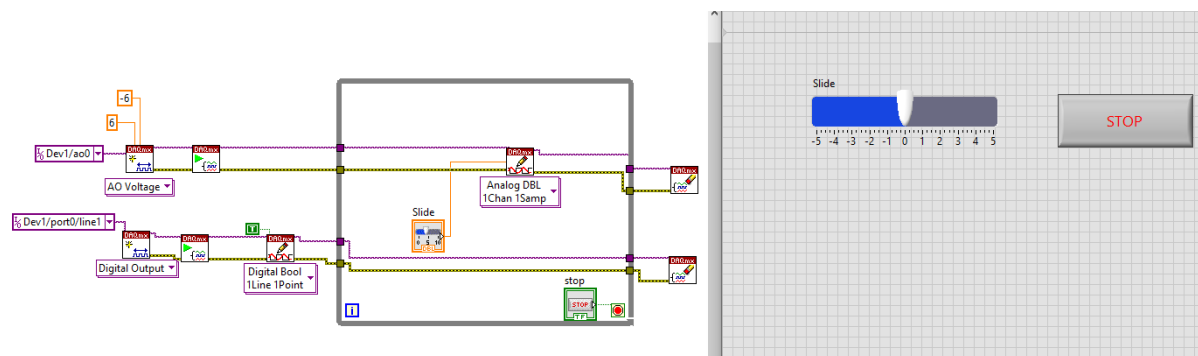


Rys.16 Zdjęcie ustawień interfejsu generatora używanego w ćwiczeniu.

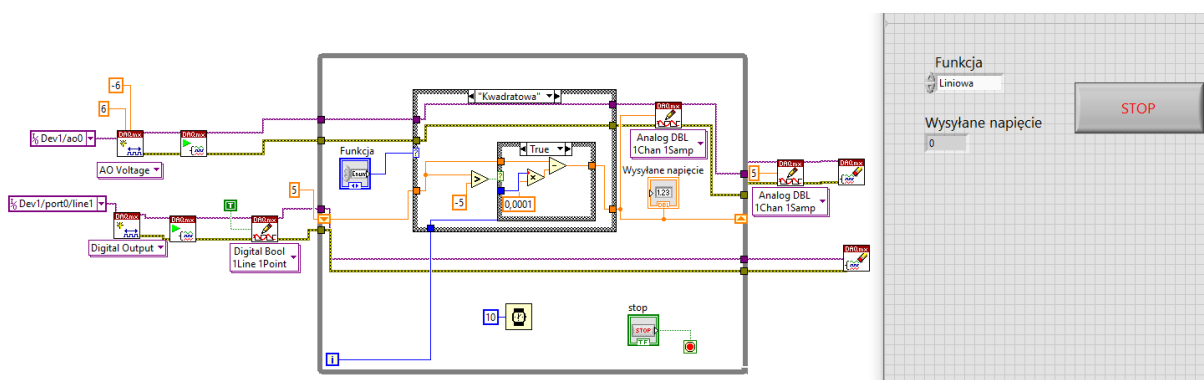


Rys.17 Podłączenie przewodów sterujących wychodzących z karty pomiarowej do generatora.

Program napisany w Labview jest bardzo prosty. Opiera się na wysyłaniu napięcia na generator w zależności od pozycji suwaka. Dodatkowo wysyła sygnał wysoki do wyjścia DIR sterownika ustawiając obroty zgodnie z ruchem wskazówek zegara.

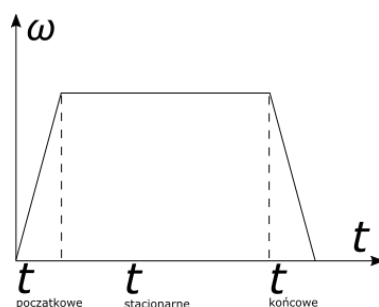


rys. 18 Schemat blokowy (po lewej) oraz interfejs użytkownika (po prawej) programu do sterowania obrotami silnika w zależności od pozycji suwaka.



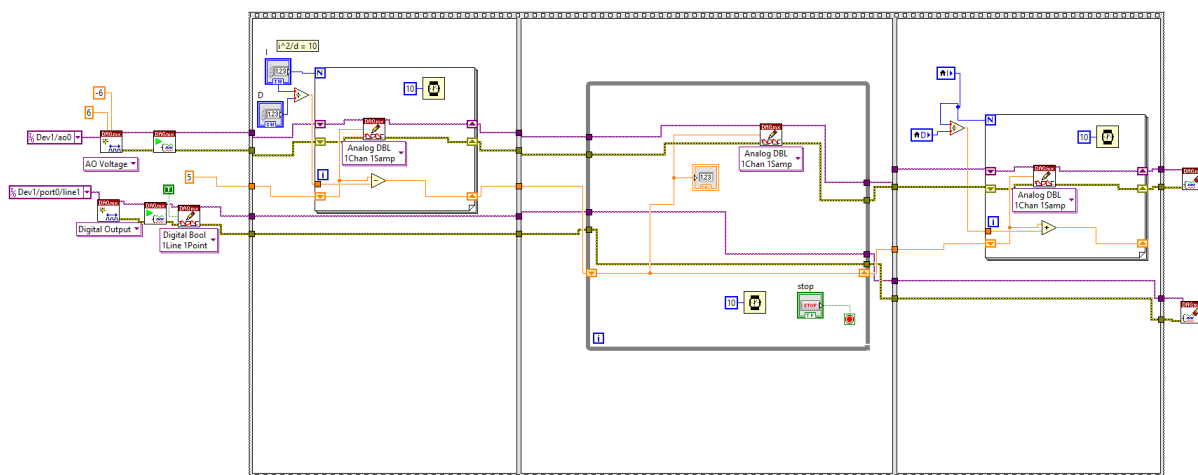
rys. 19 Schemat blokowy (po lewej) oraz interfejs użytkownika (po prawej) programu, który wysyła sygnał napięciowy według podanej przez użytkownika funkcji

W faktycznie działających układach, gdy do silnika podłączone są np. jakieś rozrządzone masy nagłe zatrzymanie silnika może narazić na uszkodzenia wał silnika i elementy do niego podłączone. Analogicznie, nagłe rozrządanie układu może powodować wiele problemów włączając sytuację, w której silnik po prostu nie ruszy. Zmodyfikujemy więc program w taki sposób, aby silnik na początku rozrządzał się do wartości zadanej, pracował na niej przez pewien czas, a następnie zwalniał aż do zatrzymania (rys. 20).



rys. 20 Schemat zasady sterowania prędkością silnika w czasie.

Rys. 21 ukazuje przykładowe użycie struktury flat sequence oraz trzech pętli w celu uzyskania powyżej opisanego zachowania. W pierwszej części wykonywana jest pętla for która liniowo zwiększa napięcie wysyłane na generator. Od parametrów I oraz D zależy szybkość wzrostu funkcji, natomiast należy pamiętać, aby było spełnione równanie $\frac{I^2}{D} = 10$. Parametr I odpowiada za ilość iteracji pętli, natomiast iloczyn I oraz D odpowiada za przyrost, który jest dodawany do każdej iteracji pętli.



rys. 21 Schemat blokowy programu, który realizuje 3 stany przyspieszeń silnika.

Bibliografia

Mechatronika Podręcznik, REA, Kaufmann, Paetzold, Baumann, Schmid, Zippel, 2002
ISBN: 978-83-7141-425-1

Variable Rate Based Microstepping of Stepper Motor Using MATLAB GUI, D.Gaan, M.Kumar,
C.Majumder, IEEE Xplore, 2017

<https://amgautomatyka.pl/>

https://pl.wikipedia.org/wiki/Silnik_krokowy

<https://wobit.com.pl/>

<https://tromil.pl/silniki-krokowe/>

<https://botland.com.pl/content/220-wszystko-o-silnikach-krokowych>

https://www.youtube.com/watch?v=fVY-QF_NmGc&ab_channel=ElektroROBBIT

<https://www.ebmia.pl/wiedza/porady/budowa-i-sterowanie-maszyn-cnc/silnik-krokowy-budowa-dzialanie-zastosowanie/>