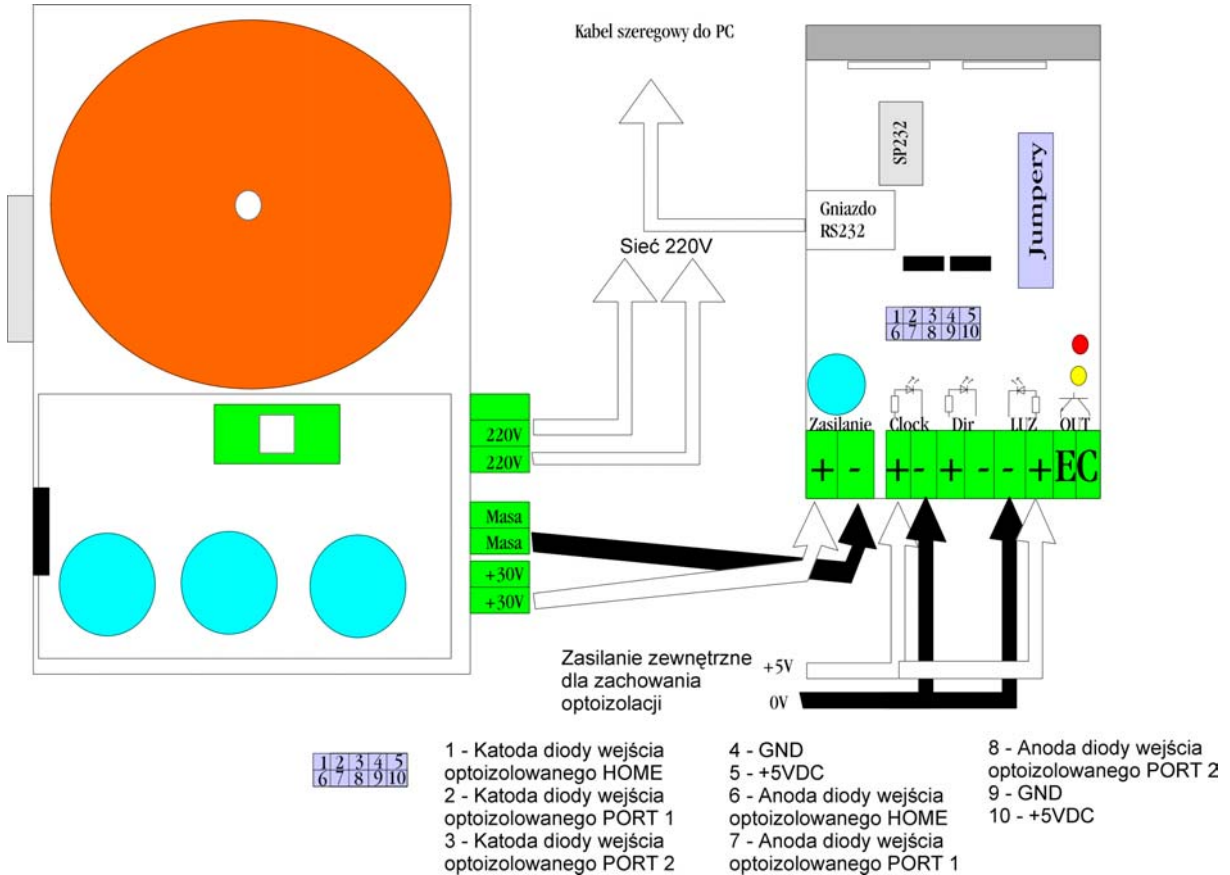


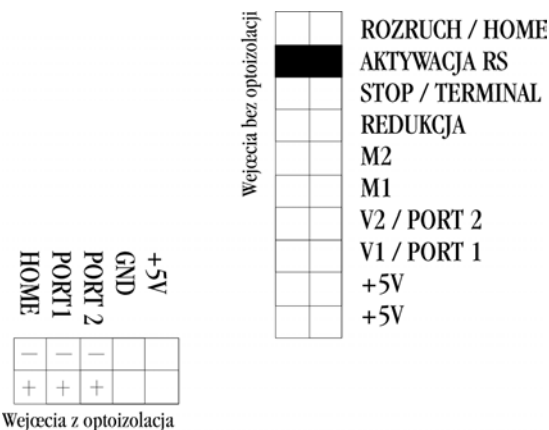
I. OPIS STANOWISKA DO BADANIA SILNIKÓW KROKOWYCH LINIOWYCH

Pracą silnika można sterować za pomocą sterownika lub przez łącze szeregowe RS485/232 z komputera.

**Rysunek przedstawiający sposób podłączenia sterownika:**



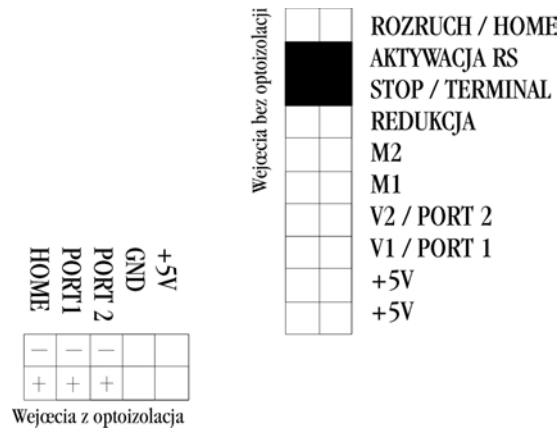
**Zwory na płycie z łączem szeregowym – ustawienie zworek dla programowania.**



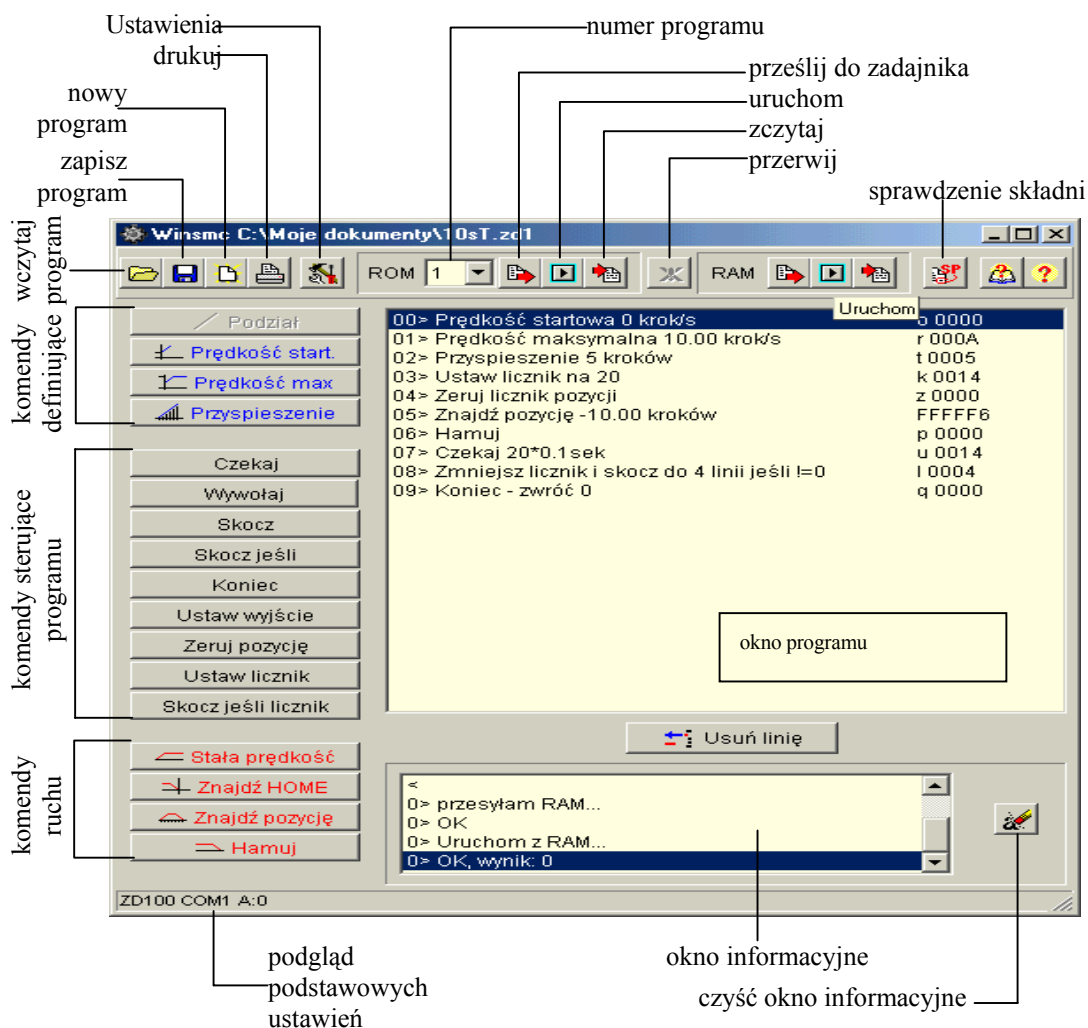
Aktywacja komunikacji poprzez moduł łącza szeregowego następuje gdy podczas załączania zasilania sterownika jest zwarta zwora **aktywacja RS**. Jednocześnie gdy nie ma zwory **terminal** sterownik przechodzi do terminalowego trybu pracy. Sterownik nie wykonuje wewnętrznego programu lecz oczekuje na komendy z łącza szeregowego wysyłane z dowolnego terminala (2400 bodów, 8 bitów, bez parzystości, brak kontroli przepływu) lub z wykorzystaniem programu SMC dla Windows. Jeżeli natomiast podczas załączania zasilania zwarta jest zwora terminal sterownik przechodzi bezpośrednio do wykonywania

wewnętrznego programu. Program uruchamiany jest aktywnym sygnałem **clock** (podanie na wejście CLK +5V) i działa w pętli – tzn. gdy sterownik dojdzie do końca programu, zaczyna go od początku.

**Ustawienie zwrotek dla pracy terminalowej – bez podłączenia do PC, a tylko z zapisanym programem:**



Programy sterujące pisze się i testuje na komputerze. Wykorzystuje się do tego celu program WINSMC pracujący w systemie operacyjnym Windows. Widoczny na ekranie monitora formularz główny programu pokazano na rys. 12.

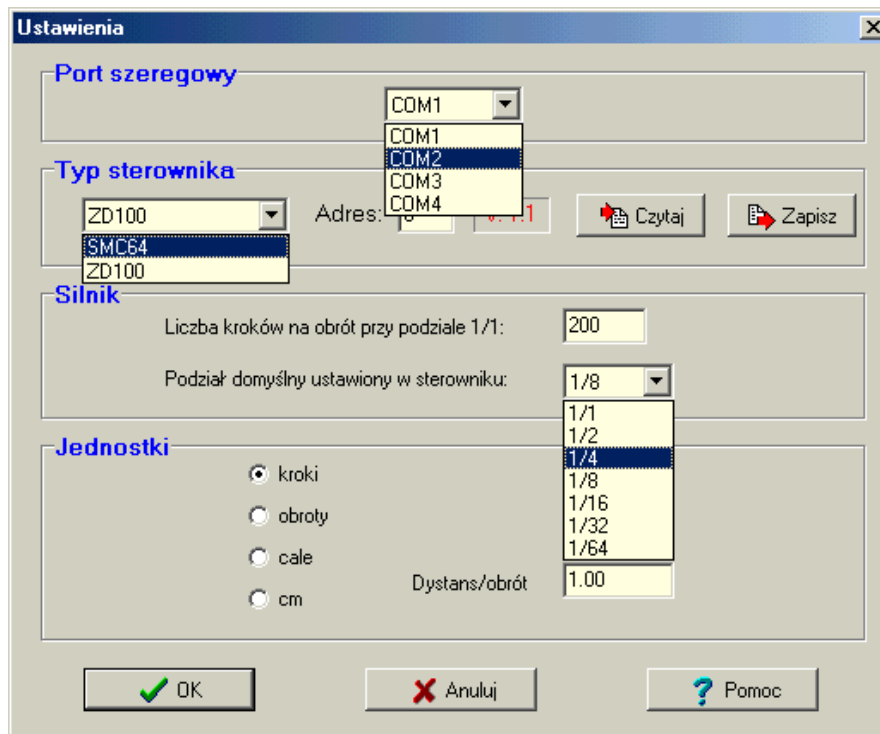


Rys. 12. Okno programu WINSMC

Wyróżnić w nim można:

- pasek narzędziowy,
- okno programu, zawierające treść tworzonego programu,
- okno informacyjne, w którym ukazują się komunikaty,
- przyciski komend.

Przed przystąpieniem do pisania programu sterującego należy ustawić parametry transmisji danych pomiędzy komputerem a sterownikiem bądź zadajnikiem. Po uaktywnieniu z paska narzędzi przycisku *ustawienia* (rys. 13) definiuje się port szeregowy, typ sterownika lub zadajnika.



Rys. 13. Okno ustawień programu WINSMC

*Okno programu* zawiera instrukcje tworzonego programu. Składają się one z numeru linii, opisu instrukcji oraz komendy w formacie docelowym. Nowa instrukcja dopisywana jest przed instrukcją aktualnie podświetloną po wybraniu odpowiedniego *przycisku komend*. Do kasowania linii służy klawisz *usuń linię*.

W *oknie informacyjnym* wyświetlane są komunikaty programu o poprawności jego wykonania bądź o stwierdzonych błędach.

### 1. Komendy uruchomieniowe

Zestaw trzech komend uruchomieniowych *prześlij*, *uruchom*, *zczytaj* w pasku narzędziowym występuje w dwóch wersjach dla pamięci ROM i RAM. Korzystanie z pamięci nieulotnej ROM, powoduje, iż programy ładowane są do zadajnika na stałe tzn. jest możliwe późniejsze ich uruchamianie bez pomocy komputera. Natomiast program załadowany do pamięci RAM jest kasowany z chwilą wyłączenia napięcia. Zaletą korzystania bezpośrednio z pamięci RAM jest szybszy dostęp. Uruchamianie z RAM należy jednak traktować jako opcję testowania programu przed jego ostatecznym załadowaniem do ROM. W programie WINSMC aktualny numer programu

Jednostki- jednostki, w których opisywane będą komendy ruchu silnika.

**Kroki** - pod pojęciem kroku rozumiemy pojedyncze przesunięcie wirnika silnika pod wpływem podania impulsu taktującego na wejściu sterownika. Fizyczna wielkość kroku zależy od aktualnego podziału w jakim pracuje sterownik. Zakładając, że silnik wykonuje 200 kroków na pełen obrót w podziale 1/1 to w podziale 1/2 wykona 400 kroków na obrót, w podziale 1/4 800, 1/8 1600 itd. Jest to o tyle istotne, iż w przypadku korzystania z jednostek kroki fizyczny efekt komend ruchu zależy od podziału np. Stała prędkość 200 kroków/s oznacza w podziale 1/1 jeden obrót/s, zaś w podziale 1/2 pół obrotu/s.

**Obroty** - jednostką jest obrót silnika wg parametrów z części ustawień: silnik.

**Cale, Cm** -jednostka liniowa wg schematu dystans/obrot, Oznacza odległość jaka jest pokonywana podczas jednego obrotu silnika.

Przy korzystaniu z jednostek obroty, cale, cm, fizyczny efekt komend nie zależy od podziału tzn. Stała prędkość 1 obrót/s oznacza 1 obrót /s w każdym zastosowanym podziale - program automatycznie przelicza wartości wg stosowanego podziału.

## **2. Sprawdzenie składni**

Komenda umożliwiająca sprawdzenie poprawności logicznej niektórych funkcji programu, w szczególności wykrywane są następujące nieprawidłowości mogące mieć wpływ na niepoprawne działanie programu:

- Brak definicji prędkości startowej.
- Brak definicji prędkości maksymalnej.
- Niepoprawne ustawienie wartości prędkości maksymalnej, startowej lub przyspieszenia np.
  - Prędkość maksymalna =0 ;
  - Prędkość startowa > Prędkość maksymalna;
- Zbyt duża wartość przyspieszenia tzn (Prędkość maksymalna - Prędkość startowa) < przyspieszenie;
- Brak komend ruchu - silnik nie wykona żadnego ruchu.
- Brak zerowania licznika pozycji przy jednoczesnym wykorzystywaniu funkcji szukania pozycji.

## **3. Komendy sterowania**

Komendy sterowania dzielimy na trzy zasadnicze grupy:

Komendy definiujące - instrukcje definiujące parametry ruchu silnika:

*Prędkość startowa, prędkość maksymalna, przyspieszenie* - parametry definiujące rozpoczęcie ruchu, ruch, hamowanie silnika.

Komendy sterujące programem - instrukcje niezwiązane bezpośrednio z ruchem silnika, umożliwiające poruszanie się po programie, kontrolujące dostępne porty wejścia, wyjścia zadajnika.

*Czekaj* - czekaj określony czas;

*Skocz* - skok bezwarunkowy;

*Skocz jeśli* - skok warunkowy;

*Ustaw wyjście* - kontrola dostępnych portów wyjściowych;

*Zeruj pozycję* - zerowanie licznika pozycji;

*Wywołaj* - wywołaj podprogram

*Koniec* - koniec programu;

Komendy ruchu - komendy bezpośrednio oddziałujące na ruch silnika:

*Stała prędkość* - ruch ze stałą prędkością;

*Znajdź HOME* - szukaj punktu odniesienia HOME;

*Znajdź pozycję* - przesuń silnik do zadanej pozycji.

*Hamuj* - hamuj silnik poruszający się ze stałą prędkością.

### 3.1. Prędkość startowa, maksymalna, przyspieszenie - komendy definiujące

**Uwaga. Komendy definiujące powinny znaleźć się na początku każdego programu**

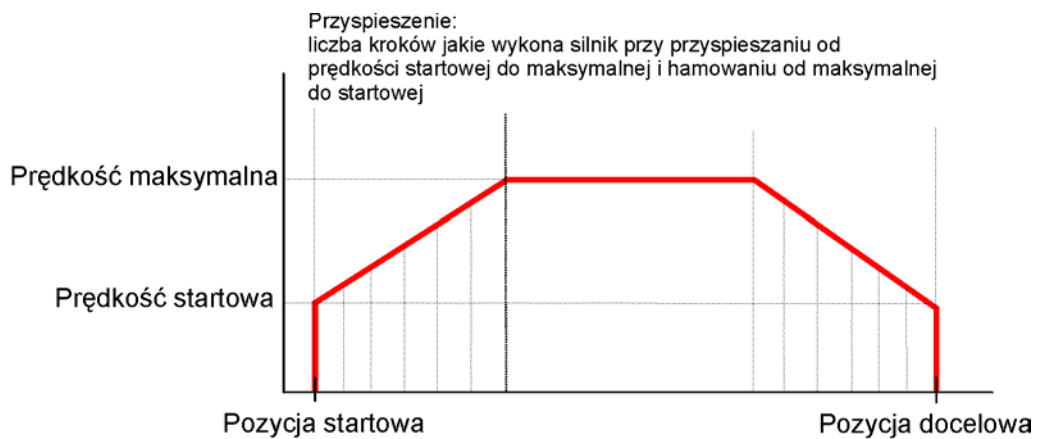
Komendy definiujące ustalają podstawowe parametry ruchu silnika:

*Prędkość startowa* - każdy ruch silnika (komendy Znajdź pozycję, Znajdź HOME, Stała prędkość) zaczynają się od prędkości startowej (pierwszy krok).

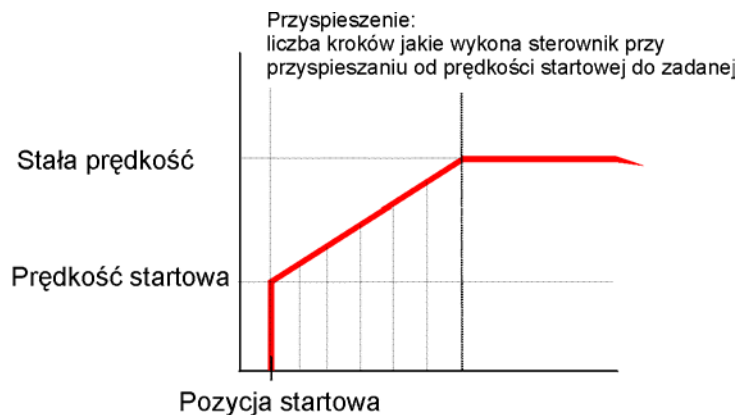
*Prędkość maksymalna* - ruch silnika podczas wykonywania komend typu znajdź odbywa się z prędkością zdefiniowaną jako maksymalną.

*Przyspieszenie* - liczba kroków, w których silnik rozpędza się od prędkości startowej do maksymalnej (lub zadanej dla stałej prędkości) oraz hamuje od prędkości maksymalnej (stałej) do startowej.

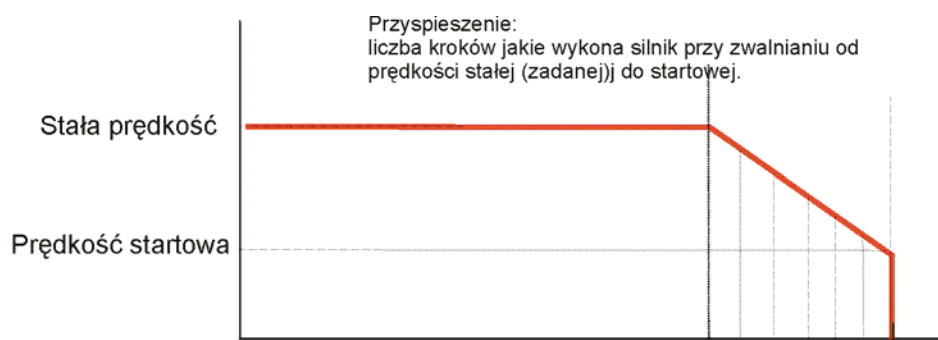
- Dla komendy Znajdź pozycję



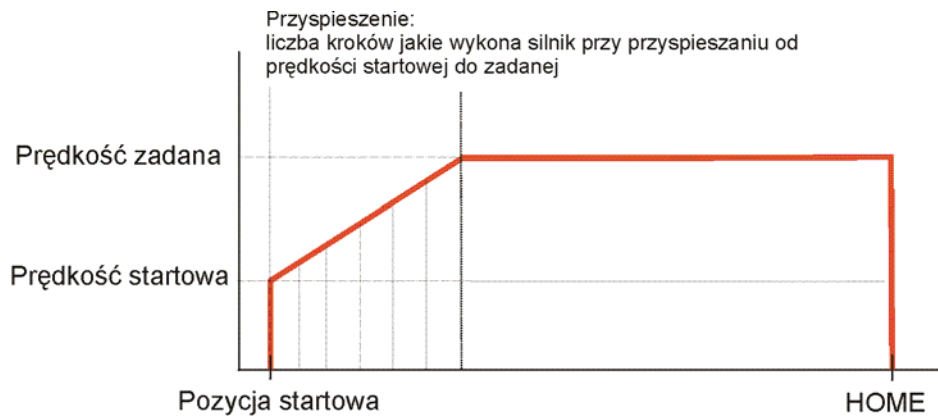
- Dla komendy Stała prędkość



- Dla komendy Hamuj



- Dla komendy Znajdź HOME



Prędkość startowa może wynosić 0 . W takim przypadku silnik rozpędza się liniowo od 0 do prędkości maksymalnej (stałej).

### 3.2. Czekaaj - komenda sterująca

Komenda powodująca zwłokę czasową o określonej wartości. Ruch silnika pozostaje taki jaki został określony poprzedzającymi komendami. Program przechodzi do wykonywania następnej komendy dopiero po upływie zadanego czasu.

### 3.3. Skocz - komenda sterująca

Skok bezwarunkowy do określonej linii. Umożliwia tworzenie pętli programowych.

```
<.>
<3> .....
....
<.> skocz do 3 linii      - skok do 3 linii
```

### 3.4. Skocz jeśli - komenda sterująca

Skok warunkowy. Wykonanie skoku do określonej linii programu pod warunkiem istnienia zdefiniowanego stanu logicznego na porcie wejściowym sterownika. Liczba portów wejściowych zależy od typu podłączonego sterownika. Instrukcja umożliwia tworzenie rozgałęzień programowych reagujących na sygnały zewnętrzne dla sterownika.

```
<.>
<3> ....
....
<6> Skocz do linii 3 jeśli PORT 0 ON  - jeśli port PORT 0 jest w stanie ON '0' skocz do linii
3, jeśli jest OFF '1' idź dalej do linii 7
<7>
```

Warunek sprawdzany jest jednokrotnie tzn nie ma oczekiwania na jego spełnienie.

### 3.5. Ustaw wyjście - komenda sterująca

Komenda umożliwiającą ustawianie zadanych wartości na wyjściach zadajnika.

```
<.>
<.> Ustaw wyjście OUT 1 - LOW '0' - ustaw wyjście OUT 1 w stan niski '0'
<.>
```

### **Uwaga:**

**Komendy wywołania można stosować tylko w programie głównym. Podprogramy nie mogą wywoływać kolejnych podprogramów.**

#### 3.6. Wywołaj - komenda sterująca

Do zadajnika ZD100 możemy zapisać 9 różnych programów. Mogą one być traktowane jako osobne programy uruchamiane oddzielnie, lub jako podprogramy uruchamiane z programu głównego komendą *Wywołaj podprogram*.

#### Program 2:

<0> Ustaw wyjście OUT 1 - LOW '0'

<1> Koniec

#### Program 1:

<.>

<.> Wywołaj program1

<.>

#### 3.7. Zeruj pozycję - komenda sterująca

Zerowanie licznika pozycji tj. przypisanie mu wartości 0.

W przypadku korzystania z komendy *Znajdź pozycję* zerowanie pozycji musi zostać przynajmniej raz wykonane w celu ustalenia wartości początkowej licznika (punktu odniesienia).

#### 3.8. Koniec - komenda sterująca

Koniec działania programu. Silnik zostaje zatrzymany. W przypadku uruchamiania programu z komputera zwracana wartość pojawia się w oknie informacyjnym. Umożliwia to identyfikację punktu zakończenia programu.

**Osiągnięcie instrukcji Koniec nie powoduje rozpoczęcia działania programu od początku. Jeśli chcemy wykonywać program cyklicznie należy użyć instrukcji skoku bezwarunkowego.**

#### **Jednokrotne wykonanie:**

<2 >

...

<10> Koniec - zwróć 1 - po osiągnięciu linii 10 program przestanie działać

#### **Wielokrotne wykonanie:**

<2>

...

<9> skocz do linii 2 - wstawiamy pętlę - program będzie wykonywany cyklicznie

<10> Koniec zwróć 1

#### 3.9. Ustaw licznik, Skocz jeśli licznik - komendy sterujące

Licznik programowy umożliwia wykonanie określonej liczby pętli programowych.

Komenda „ustaw licznik” wpisuje wartość początkową licznika.

Komenda „skocz jeśli licznik” zmniejsza zawartość licznika o 1 (pod warunkiem, że licznik > 0) i jeśli licznik przyjmie wartość różną od 0 następuje skok do określonej linii.

...

<10> Ustaw licznik na wartość 10

<11>

...

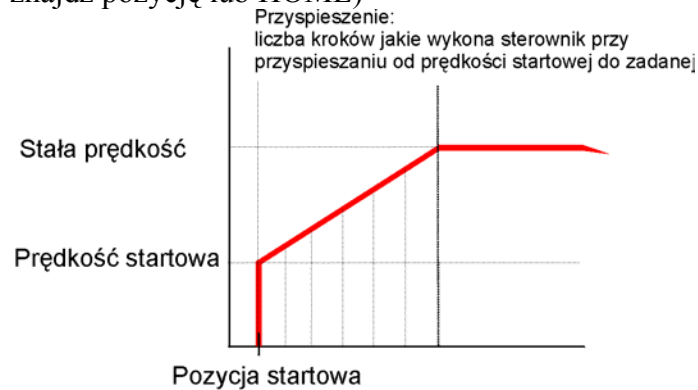
<n> Skocz jeśli licznik !=0 do linii 11

<n+1>

...Program wykona 10 razy pętlę programową od linii 11 do n. Następnie przejdzie do linii n+1;

### 3.10. Stała prędkość - komenda ruchu

Rozpędzenie silnika do zadanej prędkości. Silnik będzie pozostawał w ruchu aż do komendy hamuj (ewentualnie znajdź pozycję lub HOME)



Kolejna instrukcja stała prędkość dla silnika będącego w ruchu spowoduje odpowiednie zwiększenie lub zmniejszenie prędkości również zgodnie ze zdefiniowanym przyspieszeniem.

<..>

<..> Stała prędkość 200 kroków/s -nastąpi rozpędzenie silnika do prędkości 200 kroków/s.

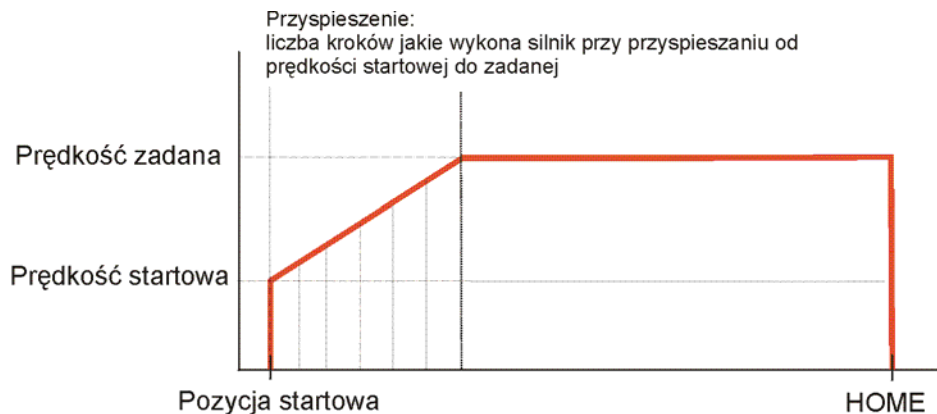
**Po osiągnięciu w/w prędkości program przechodzi do wykonywania następnej komendy.**

<..>

### 3.11. Znajdź HOME - komenda ruchu

Wprowadzenie silnika w ruch, który będzie trwał tak długo aż osiągniemy pkt HOME tj stan logiczny LOW '0' na dedykowanym porcie wejściowym zadajnika.

Podanie stanu LOW na wejście HOME powoduje **natychmiastowe** zatrzymanie silnika.



**Program przechodzi do wykonywania następnej instrukcji dopiero po osiągnięciu punktu HOME.** Komenda oczekuje na spełnienie warunku.



### 3.12. Znajdź pozycję - komenda ruchu

#### Pozycja

Położenie silnika może być identyfikowane za pomocą licznika pozycji. Jest to wewnętrzny licznik, którego wartość jest zwiększana lub zmniejszana (w zależności od kierunku obrotu silnika) co każdy krok (mikrokrok).

Oznacza to, iż zastosowany podział krokowy ma wpływ na fizyczne odwzorowanie pozycji np.

Pozycja = 10 dla podziału 1/1 odpowiada fizycznie Pozycja=20 dla podział 1/2.

- ◇ Zeruj licznik pozycji
- ◇ Podział 1/1
- ◇ Znajdź pozycję 200 kroków - silnik wykona jeden obrót
- ◇ Zeruj licznik pozycji
- ◇ Podział 1/2
- ◇ Znajdź pozycję 200 kroków - silnik wykona pół obrotu

Licznik ma pojemność +/- 7340031 i ma charakter cykliczny tj.

$$7340031+1 = -7340031$$

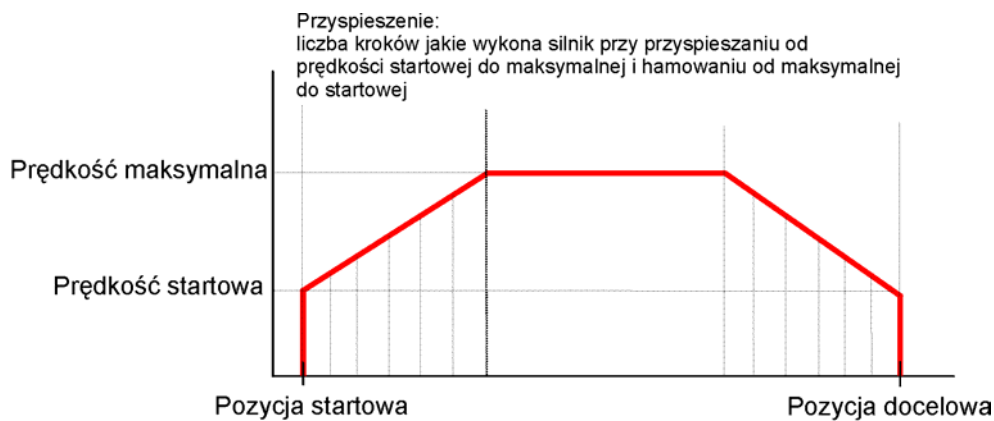
$$-7340031-1 = 7340031$$

Komenda powoduje wprawienie silnika w ruch i przemieszczenie do zadanej pozycji. W zależności od aktualnego położenia sterownik automatycznie wybierze kierunek obrotów - wg krótszej drogi.

**Po osiągnięciu zadanej pozycji silnik zatrzymuje się.**

Poniższe wykresy przedstawiają możliwe przypadki dochodzenia do pozycji docelowej:

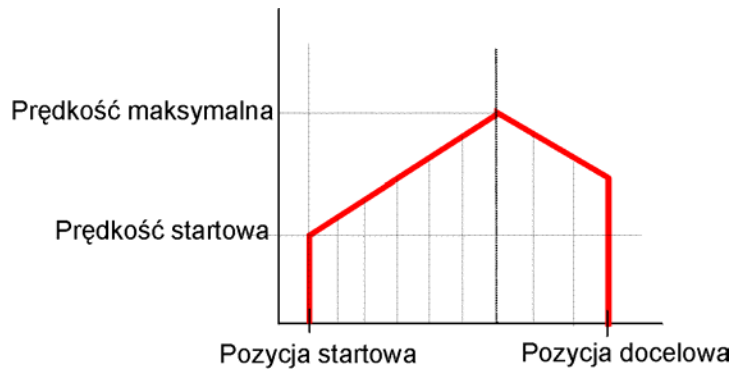
- Pozycja docelowa - pozycja startowa > 2\* przyspieszenie:



- Pozycja docelowa - pozycja startowa < przyspieszenie:

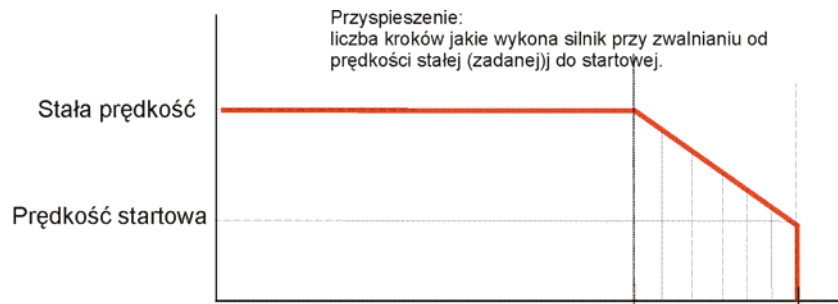


- Pozycja docelowa - pozycja startowa < 2\* przyspieszenie:



### 6.3.13. Hamuj - komenda ruchu

Hamowanie silnika wg zdefiniowanych parametrów. Komenda ma sens jedynie wtedy gdy silnik został uprzednio wprowadzony w ruch komendą Stała prędkość.



**Program przechodzi do następnej instrukcji dopiero po wyhamowaniu tj. zatrzymaniu silnika.**

## 7.0. Przykłady

### Program 1 (smc2.smc)

Silnik cyklicznie co 2 sekundy wykonuje ruch w prawo o 200 kroków w podziale 1/4, co odpowiada 1/4 pełnego obrotu (silnik o parametrze 200 kroków na obrót przy pełnym podziale). Jednocześnie ruch silnika sygnalizowany jest niskim stanem na wyjściu OUT 1.

- 00> Podział 1/4
- 01> Ustaw wyjście OUT 1 - HIGH '1'
- 02> Prędkość startowa 20.00 krok/s
- 03> Prędkość maksymalna 1500.00 krok/s
- 04> Przyspieszenie 25 kroków
- 05> Zeruj licznik pozycji
- 06> Ustaw wyjście OUT 1 - LOW '0'
- 07> Znajdź pozycję 200.00 kroków
- 08> Ustaw wyjście OUT 1 - HIGH '1'

- 09> Czekaj 20\*0.1sek
- 10> Skocz do 5 linii
- 11> Koniec - zwróć 0

*Program 2 (smc1.smc)*

Modyfikacja programu 1. Silnik cyklicznie co 2 sekundy wykonuje ruch w prawo lub w lewo o 200 kroków w podziale 1/4 , co odpowiada 1/4 pełnego obrotu w zależności od stanu na wejściu PORT 1. Ruch silnika sygnalizowany jest niskim stanem na wyjściu OUT 1. Jednocześnie sprawdzany jest stan portu wejściowego PORT 2. Podanie stanu ON powoduje zakończenie działania programu.

- 00> Podział 1/4
- 01> Ustaw wyjście OUT 1 - HIGH '1'
- 02> Prędkość startowa 20.00 krok/s
- 03> Prędkość maksymalna 1500.00 krok/s
- 04> Przyspieszenie 25 kroków
- 05> Zeruj licznik pozycji
- 06> Skok do 15 jeśli PORT 2 ON
- 07> Ustaw wyjście OUT 1 - LOW '0'
- 08> Skok do 11 jeśli PORT 1 ON
- 09> Znajdź pozycję 200.00 kroków
- 10> Skocz do 12 linii
- 11> Znajdź pozycję -200.00 kroków
- 12> Ustaw wyjście OUT 1 - HIGH '1'
- 13> Czekaj 20\*0.1sek
- 14> Skocz do 5 linii
- 15> Koniec - zwróć 0