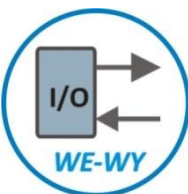


# Instrukcja obsługi SIC184 (rev2)



**Programowalny serwo-kontroler silników krokowych  
z wbudowanym kontrolerem ruchu  
i interfejsem MODBUS-RTU**



P.P.H. WObit E.J. Ober s.c.  
62-045 Pniewy, Dęboryce 16  
tel. 61 22 27 422, fax. 61 22 27 439  
e-mail: wobit@wobit.com.pl  
www.wobit.com.pl

# Spis treści

<b>1. Zasady bezpieczeństwa i montażu .....</b>	<b>4</b>
1.1 Zasady bezpieczeństwa.....	4
1.2 Zalecenia montażowe .....	4
<b>2. Wstęp .....</b>	<b>5</b>
<b>3. Opis sprzętu .....</b>	<b>7</b>
3.1 Rozmieszczenie złącz i kontrolki .....	7
3.2 Zasilanie .....	8
3.3 Podłączenie silnika .....	9
3.4 Złącze RS485 .....	10
3.5 Wyjścia .....	10
3.6 Wejścia cyfrowe .....	10
3.7 Enkoder .....	11
3.8 Wejście analogowe 0-10V .....	12
3.9 Sygnalizacja błędów .....	12
<b>4. Konfiguracja sterownika .....</b>	<b>13</b>
4.1 Informacje wstępne – parametry ruchu .....	13
4.2 Konfiguracja napędu .....	15
4.2.1 Tryby bazowania i ograniczenia ruchu .....	17
4.2.2 Kontrola pozycji z enkoderem .....	18
4.2.3 Kontrola stanu napędu .....	18
4.3 Konfiguracja wejść / komunikacji .....	19
<b>5. Programowanie sterownika .....</b>	<b>20</b>
5.1 Wstęp .....	20
5.2 Opis programu WBCprog .....	21
5.3 Opis języka WBL .....	23
<b>6. Przykłady programów .....</b>	<b>27</b>
6.1 Obsługa wejść / wyjść .....	27
6.2 Odczyt wejść analogowych (0-10V) .....	27
6.3 Sterowanie napędami .....	27
<b>7. Spis komend i rejestrów w WBL .....</b>	<b>28</b>
<b>8. Parametry techniczne .....</b>	<b>32</b>

Dziękujemy za wybór naszego produktu!

Niniejsza instrukcja ułatwi Państwu prawidłową obsługę i poprawną eksploatację opisywanego urządzenia.

Informacje zawarte w niniejszej instrukcji przygotowane zostały z najwyższą uwagą przez naszych specjalistów i służą jako opis produktu bez ponoszenia jakiegokolwiek odpowiedzialności w rozumieniu prawa handlowego. Na podstawie przedstawionych informacji nie należy wnioskować o określonych cechach lub przydatności produktu do konkretnego zastosowania.

Informacje te nie zwalniają użytkownika z obowiązku poddania produktu własnej ocenie i sprawdzenia jego właściwości. Zastrzegamy sobie możliwość zmiany parametrów produktów bez powiadomienia.

- 
- Prosimy o uważne przeczytanie instrukcji i stosowanie się do zawartych w niej zaleceń.
  - Prosimy o zwrócenie szczególnej uwagi na następujące znaki:



**UWAGA!**

Niedostosowanie się do instrukcji może spowodować uszkodzenie urządzenia albo utrudnić posługiwanie się sprzętem lub oprogramowaniem.



**UWAGA!**

Z gwarancji wyłączone są uszkodzenia mechaniczne lub elektryczne wynikające z przepięć, zwarcia oraz usterki czy awarie, których przyczyną jest wadliwa obsługa lub eksploatacja ze strony kupującego / Użytkownika.

# 1. Zasady bezpieczeństwa i montażu

## 1.1 Zasady bezpieczeństwa

- Przed pierwszym uruchomieniem urządzenia należy dokładnie zapoznać się z niniejszą instrukcją obsługi i zachować ją do późniejszego wykorzystania.
- Należy zapewnić właściwe warunki pracy, zgodne ze specyfikacją urządzenia (np.: napięcie zasilania, temperatura, maksymalny pobór prądu).
- Chronić urządzenie przed przedostaniem się do jego wnętrza jakichkolwiek przedmiotów lub płynów – grozi porażeniem i uszkodzeniem urządzenia.
- Podstawowe informacje, których znajomość i stosowanie zapewnią, aby urządzenie było użytkowane bezpiecznie i zgodnie z jego przeznaczeniem, zostaną uwidocznione na urządzeniu, a w przypadku braku takiej możliwości zostaną podane w niniejszym dokumencie.
- Urządzenie, łącznie z jego częściami składowymi, jest wykonane w taki sposób, aby zapewnić jego bezpieczny i prawidłowy montaż i przyłączenie.
- Urządzenie jest zaprojektowane i wyprodukowane w sposób zapewniający jego zgodność zasadami ochrony przed zagrożeniami, wymienionymi w ww. punktach, pod warunkiem, że urządzenie to jest użytkowane w sposób zgodny z jego przeznaczeniem i że jest odpowiednio utrzymywane.
- Urządzenie może zakłócić pracę czułych urządzeń radiowo-telewizyjnych umieszczonych w pobliżu.

## 1.2 Zalecenia montażowe

Poniżej zawarte zostały zalecenia, do których należy się stosować, by zapewnić poprawną pracę sterownika.

- Sterownik nie powinien być zasilany z tego samego źródła co sterowniki / serwonapędy silników.
- Należy zminimalizować wpływ zakłóceń pochodzących z zewnętrznych źródeł
- W celu **minimalizacji zakłóceń** przewód łączący silnik ze sterownikiem powinien być **ekranowany** lub powinien być **skręcany parami** (osobna skrętka dla fazy A i B). Zaleca się także stosowanie **pierścienia ferrytowego** na przewodzie silnika przy sterowniku.
- Przewody sygnałowe (**CLK, DIR, EN**) **nie powinny biec w pobliżu przewodów silnika** i powinny być możliwie krótkie.
- **Przewód enkodera** powinien być **ekranowany**, a jego ekran podłączony do sygnału GND sterownika.
- Dla dłuższych przewodów enkodera (powyżej 2m) zaleca się stosowanie enkodera zasilanego 12-24V z wyjściem Push-Pull.

## 2. Wstęp

**SIC184** jest programowalnym serwo-sterownikiem silników krokowych o prądzie sterowania do 4 A. Oprócz końcówki mocy posiada generator trajektorii, który pozwala na precyzyjne określenie **pozycji**, **prędkości** i **przyśpieszenia** ruchu silnika. Sterownik może zostać zaprogramowany za pomocą komend w języku skryptowym i wykonywać zadane sekwencje ruchu, reagować na stany wejść i sterować wyjściami.

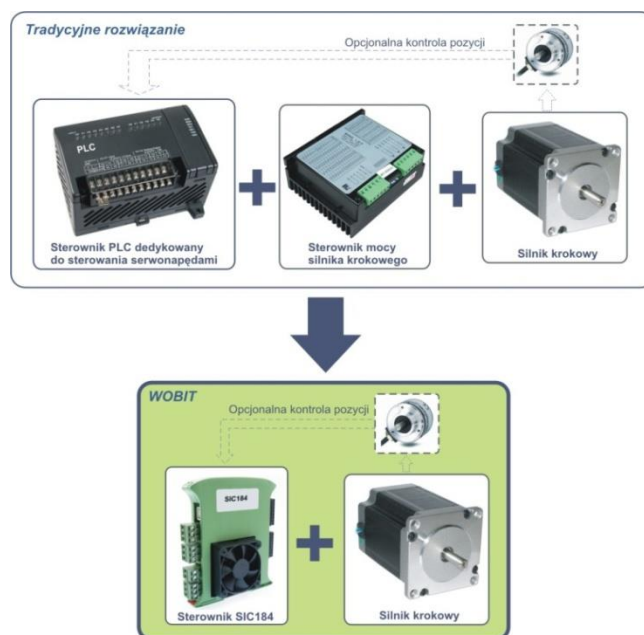
Dzięki możliwej współpracy z **enkoderem** inkrementalnym, możliwe jest zwiększenie kontroli pozycji silnika krokowego.

SIC184 posiada uniwersalne wejścia i wyjścia oraz wejście analogowe. Pozwala to na zupełne wyeliminowanie sterownika PLC w niektórych aplikacjach.

### Właściwości SIC184:

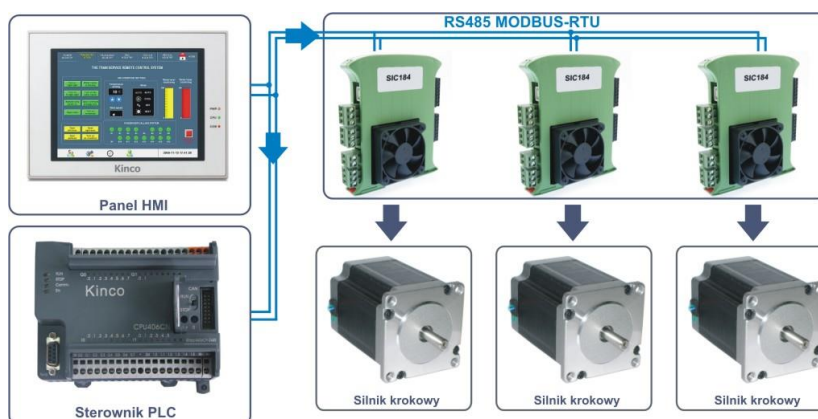
- Sterownik dla silników krokowych o prądzie do 4 A
- Wbudowany generator trajektorii (możliwość zadawania prędkości, przyśpieszenie i pozycji)
- Możliwość programowania sekwencji ruchu w prostym języku skryptowym
- Możliwość współpracy z enkoderem inkrementalnym
- Wejście analogowe 0 – 10 V
- 9 wejść (w tym 3 dedykowane dla enkodera)
- 4 wyjścia tranzystorowe PNP
- Złącze USB do konfigurowania, diagnozowania i programowania sterownika
- Złącze RS485 MODBUS-RTU do współpracy ze sterownikami PLC/panelami HMI itp.

### Sterownik SIC184 jako proste i ekonomiczne rozwiązanie w sterowaniu napędem krokowym



- SIC184 pozwala zastąpić sterownik PLC i tradycyjny sterownik mocy silnika krokowego,
- Intuicyjne oprogramowanie w języku polskim ułatwia zaprogramowanie ruchu silnika nawet przez niedoświadczonego użytkownika
- Sterownik umożliwi zapamiętanie do 2000 komend ruchu, co pozwala na realizację nawet skomplikowanych zadań ruchu, a dodatkowe wejścia i wyjścia cyfrowe pozwalają na sterowanie zewnętrznymi urządzeniami.
- SIC184 pozwala na bezpośrednie przeliczanie parametrów ruchu do jednostek np. obr. czy mm co znacznie ułatwia programowanie, a także na oszczędne zarządzanie mocą silnika, przez konfigurowalną redukcję prądu.

## Sterowniki SIC184 - sterowanie z urządzeń zewnętrznych po magistrali MODBUS-RTU



- Sterownik SIC184 może komunikować się z urządzeniami zewnętrznymi po magistrali RS485 w protokole MODBUS-RTU, będąc wówczas urządzeniem podrzędnym (SLAVE). Dzięki możliwości niezależnego zaadresowania w zakresie 1...32, możliwe jest jednoczesne sterowanie do 32 silników krokowych z urządzeń takich jak sterowniki PLC czy panele operatorskie HMI lub innych urządzeń MASTER MODBUS-RTU.

## Sterownik SIC184 - kontroler dla sterownika wysokoprądowego silnika krokowego



- Sterownik SIC184 posiada dodatkowe wyjście dla zewnętrznego sterownika mocy silnika krokowego w standardzie KROK / KIERUNEK, co pozwala na sterowanie silnikami o większych prądach, lub równoległe sterowanie kilkoma napędami.

## Sterownik SIC184 - proste i intuicyjne programowanie

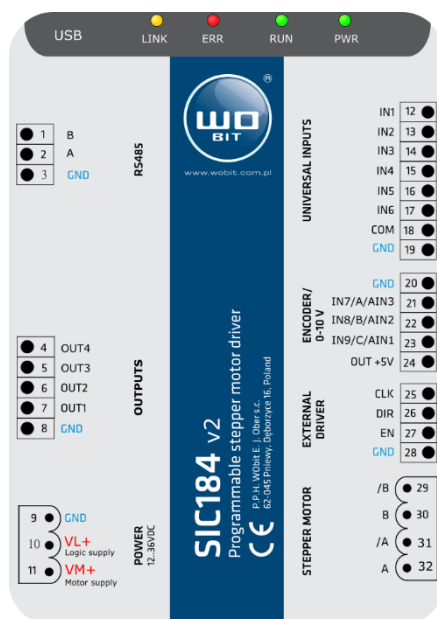


Oprogramowanie **SIC184-KONFIGURATOR** - Szybka konfiguracja i diagnostyka za pomocą złącza USB





- Złącze USB pozwala na podłączenie sterownika bezpośrednio do komputera PC. Darmowe oprogramowanie SIC-KONFIGURATOR daje możliwość konfiguracji sterownika i jego szybkiej diagnostyki (kontrola sygnałów wejść/wyjść, pracy sterownika „online” itp.), a także bezpośredniego zadawania komend ruchu (prędkości/pozycji). Oprogramowanie to pozwala także zaprogramować sterownik komendami ruchu stworzonymi w języku WBL (Wobit Basic Language).

## 3. Opis sprzętu

### 3.1 Rozmieszczenie złączy i kontroltek



Rys. 1 Opis złączy i kontroltek SIC184.

	PWR	Sygnalizacja zasilania sterownika
	RUN	Włączona/miga – błąd / przeciążenie sterownika Wyłączona – OK
	ERROR	Wyłączona – program zatrzymany Miga – program wstrzymany Włączona – program uruchomiony
	LINK	Miga – sygnalizacja komunikacja RS485

Nr	Opis	UWAGI
<b>Komunikacja (RS485)</b>		)Domyślne parametry komunikacji: Prędkość: <b>57600bps</b> , bity: 8, bit stopu: 1, parzystość: brak Adres Modbus:1, Adres wewnętrzny:0
1	RS485 B	
2	RS485 A	
3	Masa (GND)	
<b>Wyjścia uniwersalne</b>		Wyjścia tranzystorowe, maksymalny prąd <b>1A/30VDC</b> .
4	Wyjście OUT4	
5	Wyjście OUT3	
6	Wyjście OUT2	
7	Wyjście OUT1	
8	Masa (GND)	
<b>Zasilanie</b>		VL+:
9	Masa zasilania (GND)	
10	Zasilanie sterowania i wyjść (VL+)	<b>12...36 VDC</b> . Pobór prądu zależny od napięcia zasilania i ustawionego prądu sterownika.
11	Zasilanie silnika (VM+)	12...36 VDC. Pobór prądu 100mA + OUT1..4
<b>Wejścia</b>		Wszystkie wejścia są optoizolowane. Wejście jest aktywne, gdy między zacisk COM(-), a dane wejście zostanie podany sygnał min. +3V ( <b>maks.+24V</b> ).
12	IN1 – uniwersalne wejście 1	
13	IN2 – uniwersalne wejście 2	
14	IN3 – uniwersalne wejście 3	
15	IN4 – uniwersalne wejście 4	
16	IN5 – uniwersalne wejście 5	
17	IN6 – uniwersalne wejście 6	
18	COM – sygnał wspólny wejść	
19	GND – masa	
<b>Wejścia uniwersalne / enkodera / analogowe</b>		Maksymalny pobór prądu z zacisku +5V może wynosić 300mA. Dla enkoderów zasilanych z napięć 12...24V, należy zastosować
20	Masa (GND)	

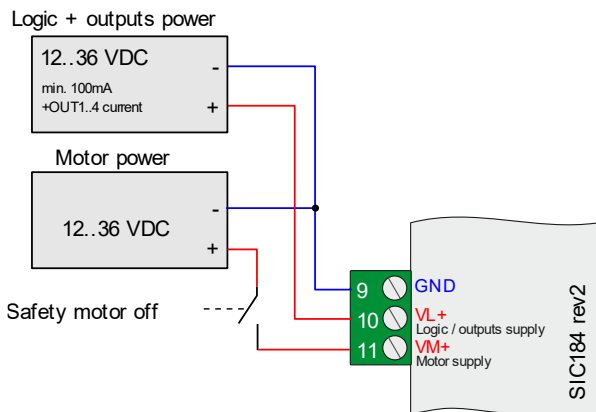


21	IN7 / ENCA - wejście uniwersalne 7 / enkoder sygnał A	zewnętrzne źródło zasilające enkoder. Maksymalne napięcie wejściowe na zaciskach A i B względem (GND) to 24V. Sygnał C dla enkodera wymaga napięcia min 12V.
22	IN8 / ENCB - wejście uniwersalne 8 / enkoder sygnał B	
23	IN9 / ENCC / AIN - wejście uniwersalne 9 / enkoder sygnał B / analogowe 0-10V	
24	Wyjście +5V do zasilania enkodera / potencjometru	
<b>Wyjście na zewnętrzny sterownik mocy</b>		Sygnały wyjściowe - Stan niski: 0V, stan wysoki: +5V Sygnał CLK – szerokość impulsu: 5µs, maks. częstotliwość sygnału: 200kHz.
25	CLK - sygnał kroku	
26	DIR – sygnał kierunku	
27	EN – sygnał aktywacji	
28	GND – masa	
<b>Silnik krokowy</b>		
29	/B	
30	B	
31	/A	
32	A	

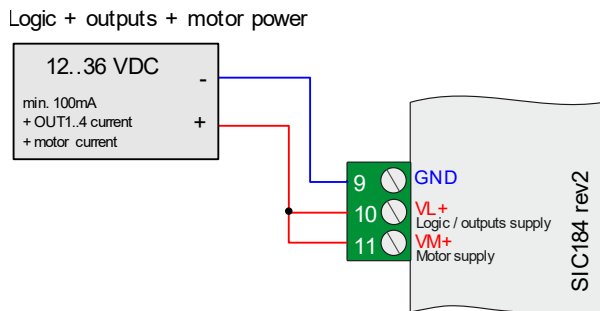
Tabela 1. Opis złączy sterownika.

### 3.2 Zasilanie

SIC184 (rev2) posiada osobne zasilanie dla części sterującej oraz silnika. Dzięki temu możliwe jest m.in. odcięcie zasilania od silnika w celu jego bezpiecznego zatrzymania przy jednoczesnym podtrzymaniu zasilania części sterującej.



Rys. 2 Osobne zasilanie części sterującej i silnika



Rys. 3 Wspólne zasilanie części sterującej i silnika

Do zasilania silnika zaleca się używać się zasilacza o napięciu wyjściowym w zakresie +12...+36 V. Zasilacz musi też odebrać energię zwrotną silnika, co umożliwiają kondensatory wyjściowe o pojemności co najmniej 4700 µF.

Minimalna wydajność prądowa źródła zasilania powinna być dobrana według poniższej zależności:

$$I_{zas} = 12 / U_{zas} * I_{ster}$$

gdzie,  $I_{zas}$  - minimalna wydajność prądowa zasilania,  $U_{zas}$  – napięcie zasilania,  $I_{ster}$  – ustawiony prąd sterownika.

Przykładowe wartości minimalnych wydajności prądowych zastosowanego źródła zasilania:

Zasilanie [V]	Prąd silnika [A]			
	1	2	3	4
12	0,8A	1,5A	2A	3,5A
24	0,5A	1A	1,6A	2,5A
36	0,4A	0,8A	1,5A	2,2A





#### UWAGA!

Odwrotna polaryzacja lub przekroczenie maksymalnego napięcia zasilania może spowodować uszkodzenie sterownika.

Zbyt małe kondensatory wyjściowe lub ich brak mogą uszkodzić sterownik lub powodować jego niewłaściwą pracę, podczas pracy sterownika z wysokim napięciem zasilania i dużym silnikiem, który pracuje dynamicznie (gwałtowne zmniejszanie jego prędkości).

Zaleca się stosowanie kondensatorów elektrolitycznych 10000  $\mu\text{F}$  przy maksymalnych prądach sterownika i większych prędkościach obrotowych (>5obr/sek.).

### Wyjście +5V

Sterownik udostępnia napięcie +5 V, które można wykorzystać do zasilania enkodera (typu TTL) lub zewnętrznego potencjometru podłączonego do wejścia AIN. Maksymalny pobór prądu 5V nie powinien przekraczać **300 mA**.

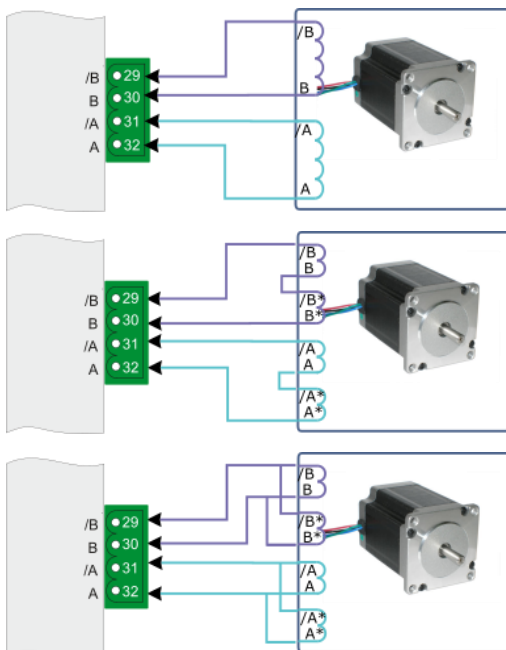


#### UWAGA!

Nie należy zwierać wyjść +5 V z masą (GND/GN), ani napięciem zasilania. Może to spowodować uszkodzenie sterownika. Należy także unikać prowadzenia przewodów z sygnałem +5 V w pobliżu innych sygnałów mogących generować zakłócenia.

## 3.3 Podłączenie silnika

Silnik należy podłączyć do sterownika za pomocą przewodu 4-żyłowego. W celu minimalizacji zakłóceń zaleca się użycia skręcanego parami przewodu (A z /A, B z /B), najlepiej dodatkowo ekranowanego. W przypadku silników unipolarnych (posiadających 6 lub 8 wyprowadzeń) należy podłączyć je według poniższych przykładów.



Rys. 4 Sposoby podłączenia silnika.

#### Podłączenie silnika bipolarnego

Przykład podłączenia silnika krokowego bipolarnego.

#### Podłączenie silnika bipolarnego równoległe

- Uzwojenia połączone równoległe w celu uzyskania większej prędkości maksymalnej silnika (większy moment przy wyższych prędkościach)
- **Zwiększony** maksymalny prąd/fazę w stosunku do nominalnego **x2**(przykładowo silnik o prądzie 2A, może być sterowany prądem maks. 4 A)

#### Podłączenie silnika bipolarnego szeregowo

- Uzwojenia połączone szeregowo w celu uzyskania większego momentu silnika (przy mniejszych prędkościach)



#### UWAGA!

Nigdy **nie należy** odłączać silnika od sterownika **podczas jego pracy**. Może to spowodować uszkodzenie sterownika.

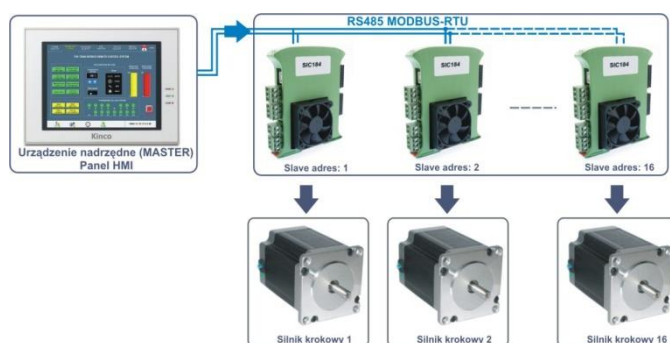
### 3.4 Złącze RS485

Sterownik SIC184 wyposażony jest w interfejs **RS485** służący do komunikacji z urządzeniami nadrzędnymi poprzez protokół **MODBUS-RTU**. Sterownik pracuje jako urządzenie SLAVE, które może być sterowane komendami ruchu przez zapis odpowiednich rejestrów. Dokładny opis rejestrów znajduje się w dokumentacji **SIC184-protokol\_MODBUS.pdf**.

Domyślne parametry transmisji:

- Prędkość: **38400bps**, bity: 8, bit stopu: 1, parzystość: brak
- Adres Modbus: 1 (dla komunikacji MODBUS-RTU)
- Adres wewnętrzny: 0 (do komunikacji i programowania poprzez program ML-PROG)

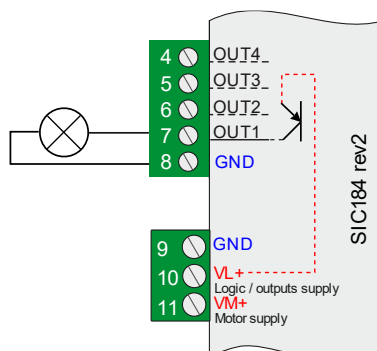
Parametry transmisji mogą być zmienione za pomocą oprogramowania SIC-KONFIGURATOR.



Rys. 5 Przykład sterowania napędów przy użyciu protokołu MODBUS-RTU.

### 3.5 Wyjścia

SIC184 (rev2) posiada 4 wyjścia tranzystorowe typu PNP o obciążalności do 5A / wyjście oraz nie więcej niż 5A łącznie na wszystkie wyjścia. Aktywne wyjście zwierane jest do zasilania VL+.



Rys. 6 Budowa wewnętrzna i sposób podłączania urządzeń do wyjść OUT1...4

### 3.6 Wejścia cyfrowe

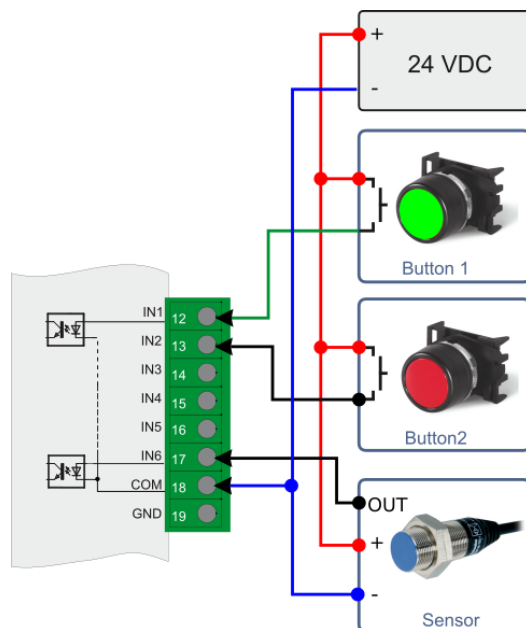
Urządzenie posiada 9 wejść, które mogą być użyte jako wejścia uniwersalne podczas tworzenia programów ruchu. Wejścia IN1...IN6 są optoizolowane. Wejścia IN7..IN9 są nieizolowane i posiadają dodatkowe funkcje:

IN7 / A, IN8/B – sygnały A i B dla enkodera inkrementalnego

IN9 / C / AIN – sygnał C dla enkodera inkrementalnego (index) lub wejście analogowe 0...10 V.

Stan wysoki dla wejść IN1...IN8 to podanie napięcia w zakresie 3...24 V.

Stan wysoki dla wejścia IN9 to podanie napięcia w zakresie 10...24 V.



Rys. 7 Przykładowy sposób podłączenia i sterowania wejść optoizolowanych IN1...IN6.

### 3.7 Enkoder

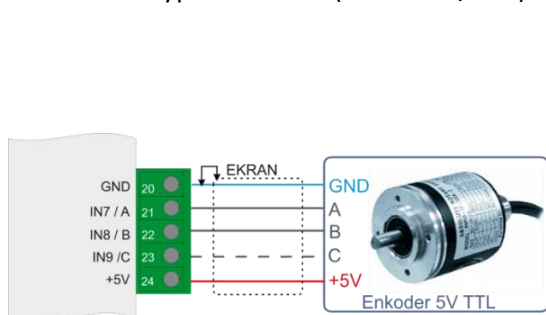
Sterownik SIC184 umożliwia podłączenie enkodera inkrementalnego. Współpracujący z silnikiem krokowym enkoder zapewnia lepszą kontrolę pozycji silnika i zapobiega gubieniu pozycji (kroków) przez silnik w przypadku jego przeciążenia lub nieoczekiwanego zatrzymania.

Enkoder może być mechanicznie podłączony z silnikiem w następujący sposób:

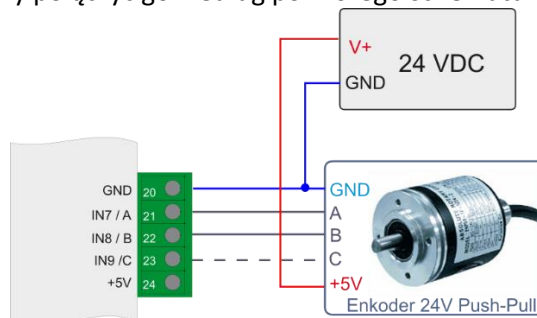
- **enkoder inkrementalny** sprzęgnięty bezpośrednio z wałem silnika krokowego,
- **enkoder inkrementalny** zamontowany pośrednio np. za przekładnią silnika krokowego,
- **liniał magnetyczny (inkrementalny)** zamontowany na elemencie wykonawczym napędzanego przez silnik krokowy

Maksymalna częstotliwość sygnału dla enkodera to **100 kHz**. Zalecany jest enkoder z wyjściem typu Push-Pull lub nadajnik linii (TTL). W przypadku enkodera typu otwarty kolektor należy dodać dodatkowe rezystory podciągające wejścia IN7A i IN8B do +5V.

W zależności od typu enkodera (Push-Pull / TTL) należy połączyć go według poniższego schematu:



Rys. 8 Podłączenie enkodera inkrementalnego 5 V TTL.



Rys. 9 Podłączenie enkodera inkrementalnego 24 V Push-Pull.



Wejście kanału C enkodera (Index) współdzielone jest z wejściem analogowym AIN. Nie jest możliwe równoczesne używanie tych dwóch sygnałów. Ponadto minimalne napięcie, które może zostać wykryte jak stan wysoki na kanale C to około 10V.

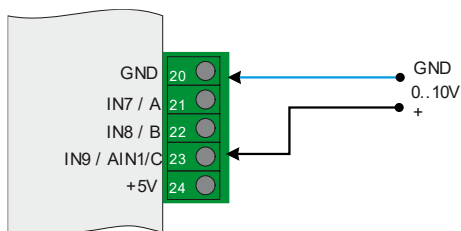


#### UWAGA!

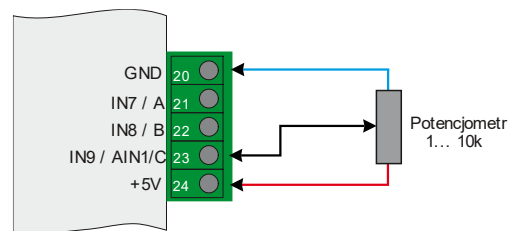
Z racji na zakłócenia, które mogą być generowane przez pracujący silnik zaleca się stosować enkoder z ekranowanym przewodem (szczególnie przy stosowaniu enkoderów typu TTL). Ekran należy podłączyć od strony sterownika do sygnału GND.

### 3.8 Wejście analogowe 0-10V

Sterownik SIC184 posiada wejście analogowe 0-10 V, które może być wykorzystane np. do sterowania prędkością lub pozycją silnika poprzez odpowiednie jego oprogramowanie podczas tworzenia programów ruchu.



Rys. 10 Sposób podłączenia sygnału 0-10V do sterownika.



Rys. 11 Przykład podłączenia potencjometru do sterownika.

### 3.9 Sygnalizacja błędów

W zależności od stanu diod SIC184 może sygnalizować następujące błędy:

Stan diod			Typ błędu	Kasowanie błędu
RUN	ERR	LINK		
OFF	ON	-	Błąd programu	Ponowny start programu
OFF	MIGA	-	Błąd pozycji napędu	-
MIGA	MIGA	-	Błąd pozycji napędu. Program w trakcie pracy zostaje wstrzymany.	Ponowny start programu
MIGA	MIGA	MIGA	Błąd krytyczny sterownika	Po kilku sekundach sterownik automatycznie zresetuje się.

Błąd programu (diody RUN wyłączona) może pojawić się gdy:

- Zapis / odczyt z nieistniejącego rejestru (np. *OUT10*, *AccM5*)
- Błąd sumy kontrolnej programu
- Uruchomienie przerwania bez etykiety przerwania

Błąd pozycji (diody RUN miga) może pojawić się gdy:

- Podczas pracy z enkoderem, gdy napęd nie może osiągnąć zadanej pozycji przez 3 sek.

## 4. Konfiguracja sterownika

Konfiguracja i programowanie sterownika odbywa się przy pomocy aplikacji SIC-KONFIGURATOR (wersja v3 lub wyższa).

Do połączenia przez USB należy użyć przewodu USB typu A – B mini. Po podłączeniu do komputera można włączyć zasilanie sterownika i uruchomić program SIC-KONFIGURATOR. Poprawna komunikacja będzie sygnalizowana informacją w górnym oknie programu.



### UWAGA!

- 1) Połączenie USB należy wykonać zawsze przed włączeniem zasilania sterownika.
- 2) Połączenie USB podatne jest na zakłócenia w sieci zasilającej oraz na zakłócenia elektromagnetyczne występujące w warunkach przemysłowych. W przypadku pojawiania się problemów z komunikacją należy zastosować dodatkowe elementy zabezpieczające w postaci:
  - Stosowania filtrów sieciowych,
  - Stosowania przewodu USB dobrej jakości, o długości < 1,5m wyposażonego w koraliki ferrytowe
  - Stosowania optoizolowanych HUBów USB po stronie komputera PC

Przy większych zakłóceniach może zdarzyć się, że komunikacja nie będzie możliwa.



SIC184 komunikuje się przy wykorzystaniu portu USB (1.1, 2.0).

Znane są problemy z kompatybilnością portu USB 3.0 (niebieski kolor gniazda) w systemie Windows 7 podczas komunikacji z urządzeniami USB HID. W przypadku problemów z komunikacją należy podłączyć sterownik do portu USB 2.0.

### 4.1 Informacje wstępne – parametry ruchu

Sterownik SIC184 pozwala na pracę sterownika w trybie **prędkości** lub **pozycji**.

**Tryb prędkości** powoduje ustawienie zadanej prędkości **VMAX** [jednostek/sek]. Zwiększenie lub zmniejszenie obrotów silnika do osiągnięcia zadanej prędkości realizowane jest z ustawionym parametrem **przyśpieszenia ACC** [jednostek /sek<sup>2</sup>].

**Tryb pozycji** powoduje ustawienie zadanej pozycji [jednostek]. Silnik rozpędza się z przyspieszeniem **ACC** [jednostek /sek<sup>2</sup>] i zwalnia z opóźnieniem **DEC**. Podczas ruchu silnik nie przekracza prędkości **VMAX** [jednostek /sek].

Prędkość i pozycja może być zadawana z wartościami ułamkowymi (np. zadanie pozycji 3,0254 jednostek.) Domyślnie **jednostką** są pełne obroty silnika. Wartość ta może być jednak przeskalowana tak by np. dostosować obroty na np. mm w przypadku napędu zamieniającego ruch obrotowy na liniowy.

Podczas zadawania prędkości lub pozycji dla napędu sterownik generują rampę (rozpędzanie/hamowanie) w celu uzyskania płynności ruchu. Uwzględniane są następujące parametry:

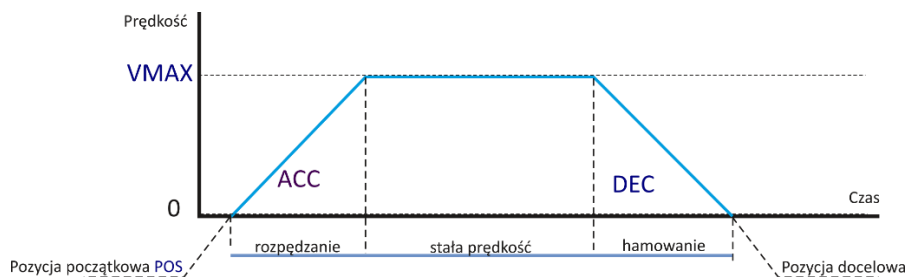
Parametr	Opis	Jednostki
<b>VMAX</b>	Prędkość maksymalna w trybie zadanej pozycji. Podczas ruchu na nową pozycję prędkość ta nie zostanie przekroczona.	[j/s]
<b>ACC</b>	Przyspieszenie dla rozpędzania w trybie zadanej pozycji. Przyspieszenie dla rozpędzania i hamowania w trybie zadanej prędkości.	[j/s <sup>2</sup> ]
<b>DEC</b>	Przyspieszenie dla hamowania w trybie zadanej pozycji.	[j/s <sup>2</sup> ]

j – jednostka ruchu zależna od konfiguracji napędu (np. obr.,mm itp.)

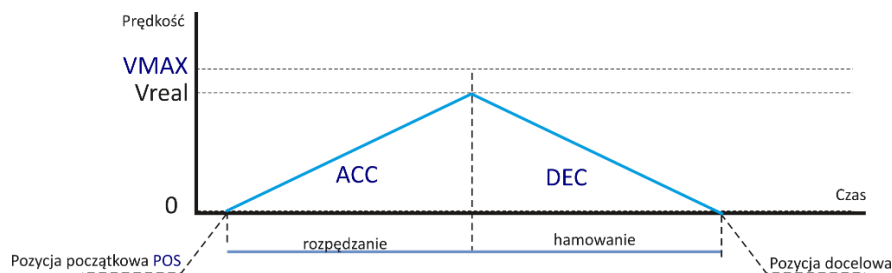
## Rampa dla zadanej pozycji

- **Rozpędzanie:** rozpoczęcie ruchu następuje od prędkości 0 do prędkości określonej parametrem **VMAX** (prędkość maksymalna dla pozycji) z przyspieszeniem **ACC**.
- **Stała prędkość VMAX** (tylko wówczas jeśli docelowa droga będzie dłuższa niż droga potrzebna na rozpędzenie i wyhamowanie z zadanymi parametrami i **DEC**).
- **Hamowanie:** zmniejszanie prędkości do 0 z opóźnieniem **DEC**, aż do osiągnięcia pozycji docelowej.

Podczas zadawania pozycji w zależności od wartości przyspieszeń dla rozpędzania i hamowania napęd może osiągnąć prędkość maksymalną jeśli łączna droga potrzebna na rozpędzenie i wyhamowanie będzie mniejsza niż droga zadana do przebycia. Rys. 12 pokazuje przypadek osiągnięcia prędkości maks., natomiast Rys. 13 pokazuje rampę z ograniczoną prędkością do prędkości **Vreal**.



Rys. 12 Przykład rampy pozycji – prędkość Vmax możliwa do osiągnięcia.



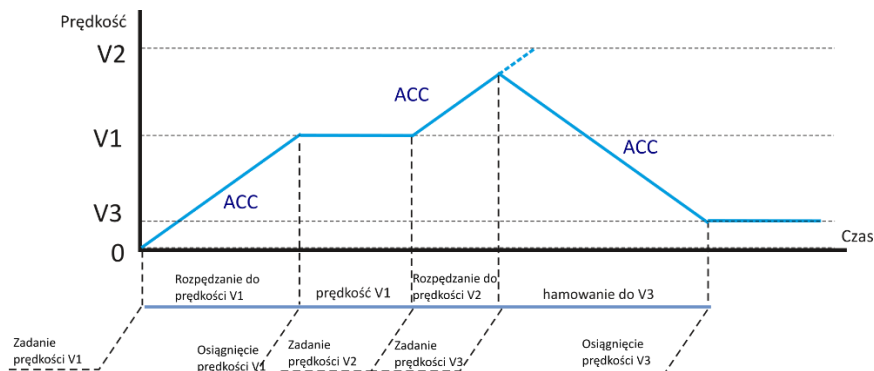
Rys. 13 Przykład rampy pozycji – prędkość VMAX niemożliwa do osiągnięcia.

Przykładowo dla parametrów ruchu **VMAX** = 1, **ACC** = 1, jednostek = obroty, zadanej pozycji 10 obrotów, silnik rozpędzi się do prędkości 1 obr/sek. w czasie 1 sek., wykona kolejne 8 obrotów w 8 sekund, a następnie wyhamuje do prędkości 0 w ciągu 1 sekundy.

### Uwagi:

- Zadanie pozycji powoduje ruch zawsze od prędkości zerowej.
- Zadanie nowej pozycji większej od aktualnej podczas trwającego ruchu spowoduje ustawienie nowej pozycji z prędkością **VMAX** = aktualnej prędkości silnika.
- Zadanie nowej pozycji mniejszej od aktualnej podczas trwającego ruchu spowoduje nagłe zatrzymanie silnika i ruch w drugą stronę. **Taka sytuacja może spowodować utratę pozycji dla napędu krokowego podczas pracy bez enkodera.**
- Zadanie prędkości podczas trwania ruchu na pozycję spowoduje ustawienie nowej prędkości z rampą ACC (dla wyhamowania lub przyspieszenia do prędkości zadanej).

## Rampa dla zadanej prędkości

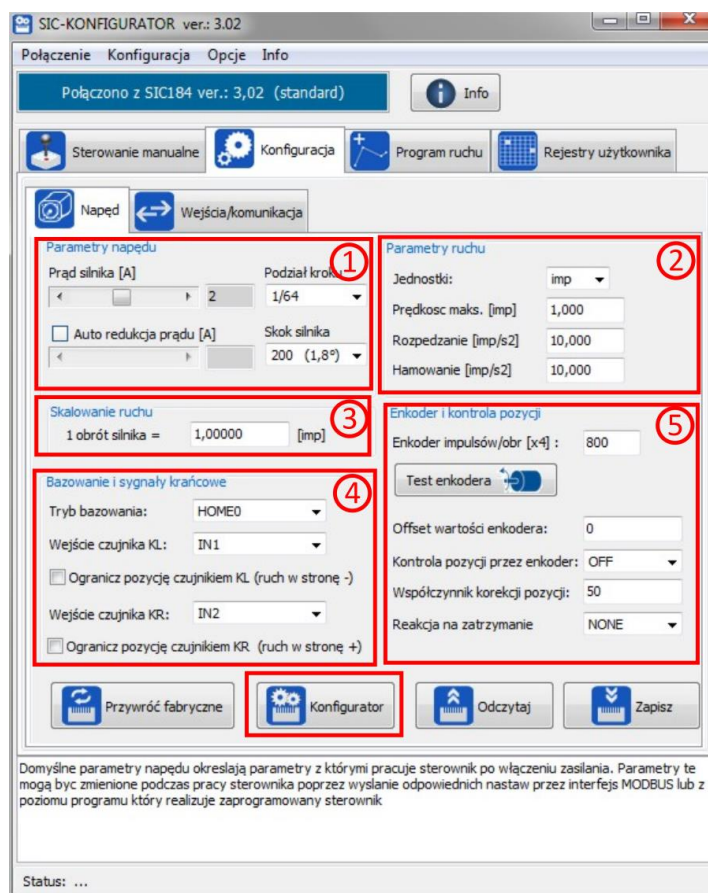


Rys. 14 Przykład rampy dla zadanej prędkości.

#### Uwagi:

- Rampa dla prędkości (dla rozpędzania i hamowania) realizowana jest zawsze z parametrem **ACC**.
- **Podczas trwania ruchu nie jest możliwa zmiana parametru ACC**. Nowa wartość **ACC** uwzględniana jest podczas ruchu od prędkości zerowej.
- Zadanie nowej prędkości podczas ruchu spowoduje osiągnięcie nowej prędkości z uwzględnieniem rampy **ACC**.
- Zadanie pozycji większej od aktualnej pozycji silnika spowoduje ruch do tej pozycji z aktualną prędkością VMAX.
- Zadanie pozycji mniejszej od aktualnej pozycji silnika spowoduje nagłe zatrzymanie silnika i ruch w drugą stronę. **Taka sytuacja może spowodować utratę pozycji dla napędu krokowego podczas pracy bez enkodera.**
- 

## 4.2 Konfiguracja napędu



Rys. 15 Okno konfiguracji napędu w programie SIC-KONFIGURATOR.



## 1) Parametry napędu:

Są to domyślne parametry napędu, z którymi będzie pracował sterownik po włączeniu zasilania. Zapisywane są w pamięci nieulotnej sterownika. Parametry te mogą być zmieniane podczas pracy sterownika poprzez protokołu MODBUS lub odpowiednie komendy podczas programu realizowanego z pamięci sterownika.

- **Prąd silnika** należy ustawić w zależności od zastosowanego silnika krokowego. Ustawienie zbyt dużego prądu dla słabszego silnika spowoduje jego szybsze nagrzewanie się lub może spowodować uszkodzenie.
- **Automatyczna redukcja prądu** powoduje automatyczne ograniczenie prądu do ustawionej w redukcji wartości, gdy silnik nie znajduje się w ruchu.
- **Podział kroku** wbudowanego sterownika mocy umożliwia podzielenie standardowego kroku (200 kroków / obrót silnika) na mniejsze części, co zapewnia płynniejszy ruch i dokładniejsze pozycjonowanie. Przykładowo wybranie podziału 1/16 pozwala ustawić wał silnika krokowego z dokładnością  $1/200/16 = 0,0003125$  obrotu.
- **Skok silnika** należy ustawić w zależności od zastosowanego silnika. Najczęściej jest to 200.

## 2) Parametry ruchu:

Są to domyślne parametry ruchu, z którymi będzie pracował sterownik po włączeniu zasilania. Zapisywane są w pamięci nieulotnej sterownika. Parametry te mogą być zmieniane podczas pracy sterownika poprzez protokołu MODBUS lub odpowiednie komendy podczas programu realizowanego z pamięci sterownika.

- **Jednostki** – określają jednostki dla parametrów i zadanych wartości ruchu
- **Prędkość maksymalna VMAX** (opisane w rozdziale 4.1)
- **Przyśpieszenie ACC** (opisane w rozdziale 4.1)
- **Hamowanie DEC** (opisane w rozdziale 4.1)

## 3) Skalowanie ruchu:

Parametr ten pozwala przeliczyć domyślne jednostki (jakimi są obroty napędu) na inne jednostki np. mm.  
Przykład:

Silnik krokowy napędza śrubę toczną o skoku 4 mm/obrot. By przeskalować wartość w obrotach na mm należy wprowadzić „1 obrót silnika = 4”

## 4) Bazowanie i sygnały krańcowe:

Pozwala określić numery wejść służące do bazowania napędu/kontroli skrajnych pozycji oraz wybrać tryb bazowania.


- **Tryb bazowania** – pozwala wybrać tryb bazowania napędu (opisane w 4.2.1)
- **Wejście czujnika KL/KR** – pozwala wybrać numer wejścia dla sygnału bazującego/krańcowego lewego/prawego.
- **Ogranicz pozycję czujnikiem KL/KR** – określa, czy wejścia KL/KR mają pełnić rolę sygnałów krańcowych, zatrzymujących ruch napędu po ich aktywowaniu.


## 5) Enkoder i kontrola pozycji

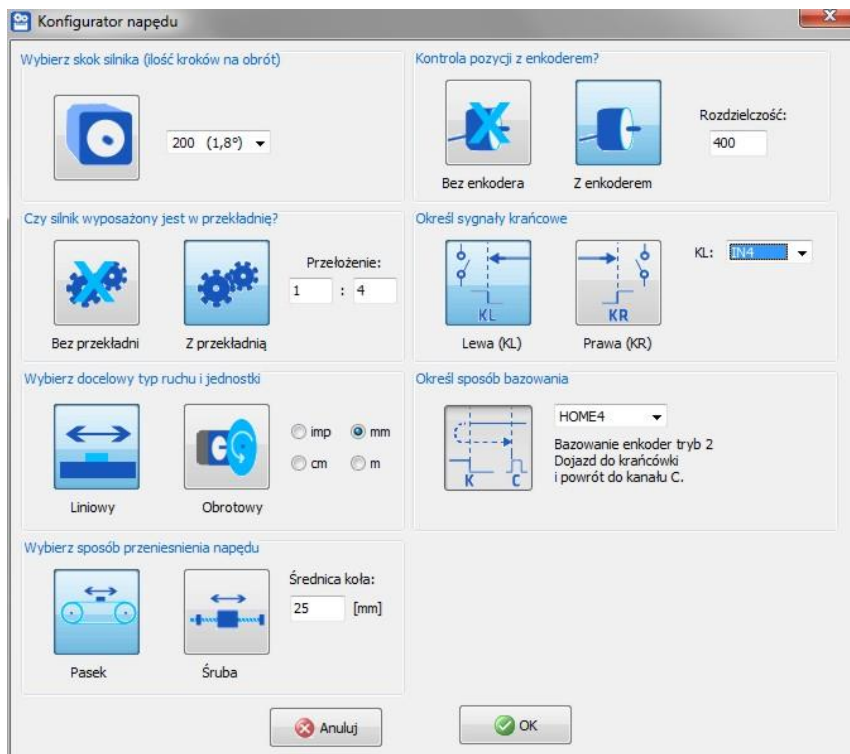
Ustawienia związane z enkoderem

- **Enkoder impulsów / obr [x4]** – należy wprowadzić rozdzielczość enkodera z kwadraturą (x4) (opisane w rozdziale 4.2.2)
- **TEST ENKODERA** – pozwala przetestować poprawność podłączenia enkodera.
- **Offset wartości enkodera** – domyślna wartość enkodera po resecie.
- **Kontrola pozycji przez enkoder** – włączenie kontroli pozycji napędu za pomocą enkodera.
- Podłączony do sterownika enkoder wymaga jego skonfigurowania przez wprowadzenie **rozdzielczości enkodera** (ilości impulsów na obrót).
- **Współczynnik korekcji pozycji** - określa wzmocnienie regulatora pozycji dla trybu kontroli pozycji z enkodera. Im większa wartość tym szybsze ustalenie pozycji po utracie kroku.
- **Reakcja na zatrzymanie:** Sposób zachowania sterownika w przypadku wykrycia utraty pozycji. Opcja dostępna tylko dla trybu kontroli pozycji z enkodera:  
NONE - brak reakcji

ERRMODE 0 = zatrzymanie napędu, status napędu (rejestr STATUS) = M\_POS\_ERROR  
 ERRMODE 1 = ERRMODE 0 + zatrzymanie programu  
 ERRMODE 2 = ERRMODE 1 + włączenie wyjścia OUT1

Po ustawieniu parametrów należy zapisać je w sterowniku za pomocą przycisku „Zapisz” 

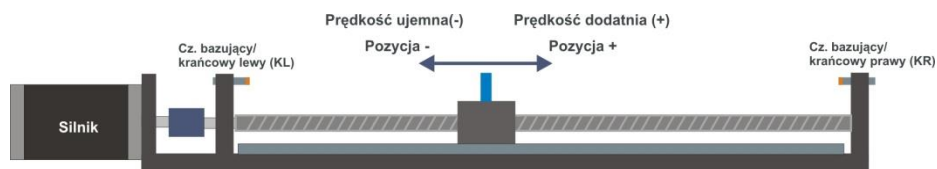
Przycisk „Konfigurator”  pozwala w prosty sposób skonfigurować użyty napęd.



Rys. 16 Okno konfiguratora napędu.

#### 4.2.1 Tryby bazowania i ograniczenia ruchu

Sterownik pozwala na ustawienie dwóch sygnałów wejściowych, które mogą służyć do bazowania i/lub ograniczenia skrajnych pozycji podczas ruchu. Sygnały nazwane są jako **KL** (bazowanie lub ograniczenie ruchu podczas ruchu w kierunku ujemnym (zmniejszanie pozycji)) oraz **KR** (w kierunku przeciwnym).



Rys. 17 Sygnały krańcowe i kierunki ruchu napędu.

Bazowanie napędu może odbywać się w jednym z poniższych trybów, ustawianych w nastawie **HOME**:

- **HOME0**: Bazowanie proste. Zatrzymanie napędu następuje od razu po dojeździe do czujnika krańcowego.
- **HOME1**: Bazowanie precyzyjne tryb 1. Napęd dojeżdża do czujnika krańcowego, zatrzymuje się, a następnie powoli cofa aż do zaniku sygnału z czujnika (cofnięcie przed czujnik).
- **HOME2**: Bazowanie precyzyjne tryb 2. Napęd dojeżdża do czujnika krańcowego, zatrzymuje się, a następnie rusza powoli, aż do zaniku sygnału z czujnika (przejazd za czujnik).

- **HOME3:** Bazowanie enkoder tryb 1. Napęd dojeżdża do mechanicznej blokady, zatrzymuje się, a następnie powoli cofa, aż do wykrycia sygnału z kanału C enkodera (INDEX).
- **HOME4:** Bazowanie enkoder tryb 2. Napęd dojeżdża do czujnika krańcowego, zatrzymuje się, a następnie powoli cofa, aż do wykrycia sygnału z kanału C enkodera (INDEX).

Dla trybów HOME1 do HOME4 ruch powrotny odbywa się z prędkością 20% prędkości bazowania.

Ograniczenie pozycji przez sygnały krańcowe konfiguruje się za pomocą nastawy **LIMIT**. Napęd zatrzymuje się, gdy wykryty zostanie sygnał krańcowy. Możliwy jest wówczas ruch tylko w przeciwną stronę. Nie należy ustawiać tego samego wejścia gdy aktywne są dwa sygnały krańcowe (tryb KL+KR) – spowoduje to zablokowanie pracy napędu po aktywacji czujnika.

Możliwe jest także programowe ograniczenie pozycji (rejstry **LIML** oraz **LIMR** dostępne przez Modbus lub program WBCprog). Domyślnie rejestry te przechowują maksymalne zakresy pozycji.

## 4.2.2 Kontrola pozycji z enkoderem

Sterownik może kontrolować pozycję silnika w oparciu o zewnętrzny enkoder podłączany do wejść IN7/A oraz IN8/B.

Do prawidłowej współpracy enkodera z silnikiem konieczne jest wprowadzenie poprawnych parametrów „Skok silnika” określającego ilość impulsów silnika na pełen obrót oraz „Enkoder impulsów / obr [x4]” określającego ilość impulsów z enkodera na pełen obrót z uwzględnieniem kwadratury.

Przykładowo dla enkodera o rozdzielczości podanej na obudowie 400, powyższy parametr będzie równy  $4 * 400 = 1600$ .

Po wprowadzeniu parametrów należy przetestować napęd zadając np. pozycję absolutną równą 1. Możliwe są wówczas następujące sytuacje:

Zachowanie napędu	Opis
Wykonanie pełnego obrotu.	Parametry i podłączenie enkodera poprawne.
Napęd wykona pełen obrót, a następnie cofnie się.	Wprowadzona za mała rozdzielczość enkodera (np. nie uwzględniona została kwadratura) lub za duża rozdzielczość silnika.
Napęd wykona więcej jak pełen obrót i zatrzyma się.	Wprowadzona za duża rozdzielczość enkodera lub za mała rozdzielczość silnika.
Napęd wykona pełen obrót i będzie kontynuował pracę bez zatrzymania.	Błędnie podłączony enkoder lub silnik. Należy zamienić jedną z faz silnika (np. A z /A) lub kanały z enkodera (A z B)
Napęd wykona pełen obrót, ale podczas utraty pozycji wolno ją koryguje.	Należy zwiększyć wartość regulatora pozycji KP.

## 4.2.3 Kontrola stanu napędu

Stan napędu może być kontrolowany za pomocą:

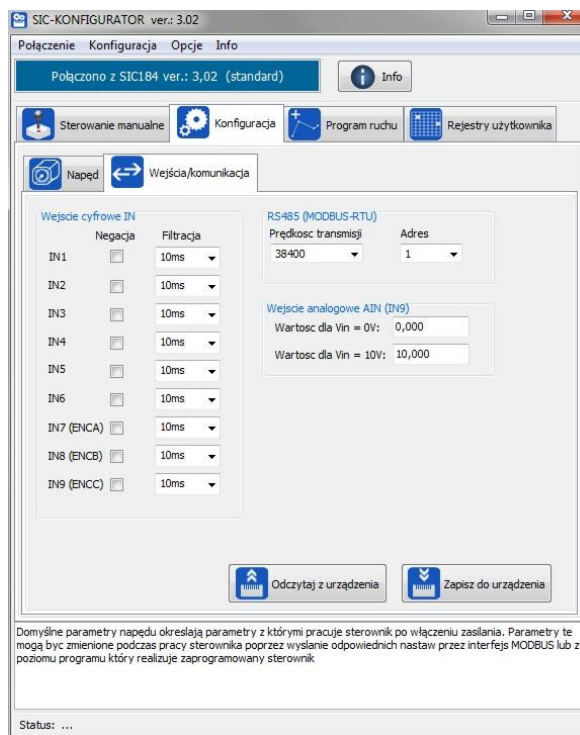
- Informacji w oknie sterowania manualnego,
- Rejestrów MODBUS: *STATUS (1010)*
- Rejestrów w programie WBCprog: *STATUS*

W tabeli poniżej zestawiono wartości rejestru statusowego i odpowiadające im stany napędu.

Wartość rejestru	Opis
0	Napęd wyłączony (sygnał EN = 0)
1	Napęd włączony, ale nie znajduje się w ruchu (sygnał EN aktywny)
2	Napęd znajduje się w trybie zadanej prędkości
3	Napęd znajduje się w trybie ruchu do zadanej pozycji
4	Napęd dojechał do zadanej pozycji

5	Błąd dojazdu do zadanej pozycji (dla pracy z enkoderem)
6	Napęd w trybie bazowania
7	-
8	Napęd w trybie korekcji pozycji (dla pracy z enkoderem)
9	Napęd osiągnął krańcową pozycję L podczas ruchu w stronę zmniejszającą pozycję (programową, lub przez aktywację sygnału krańcowego KL)
10	Napęd osiągnął krańcową pozycję R podczas ruchu w stronę zwiększającą pozycję (programową, lub przez aktywację sygnału krańcowego KL)

### 4.3 Konfiguracja wejść / komunikacji



Rys. 18 Okno ustawień wejść i komunikacji.

#### Wejście cyfrowe IN

- **Negacja** - gdy zaznaczone, brak sygnału na wejściu oznacza stan wysoki
- **Filtracja** - określa minimalną długość sygnału jaka musi pojawić się na wejściu, by został on uznany za stan wysoki. Filtracja dotyczy się tylko odczytu wejść w programie ruchu. Nie obowiązuje dla sygnałów bazujących/krańcowych oraz enkodera.

#### RS485 (MODBUS-RTU)

- **Prędkość transmisji** - prędkość transmisji RS485
- **Adres** - adres sterownika SIC184 w komunikacji MODBUS.

#### Ustawienia wejścia analogowego AIN

- **Wartość dla Vin = 0V** – określa wartość rejestru AIN1 dla napięcia = 0 V.
- **Wartość dla Vin = 10V** – określa wartość rejestru AIN1 dla napięcia = 10 V.

## 5. Programowanie sterownika

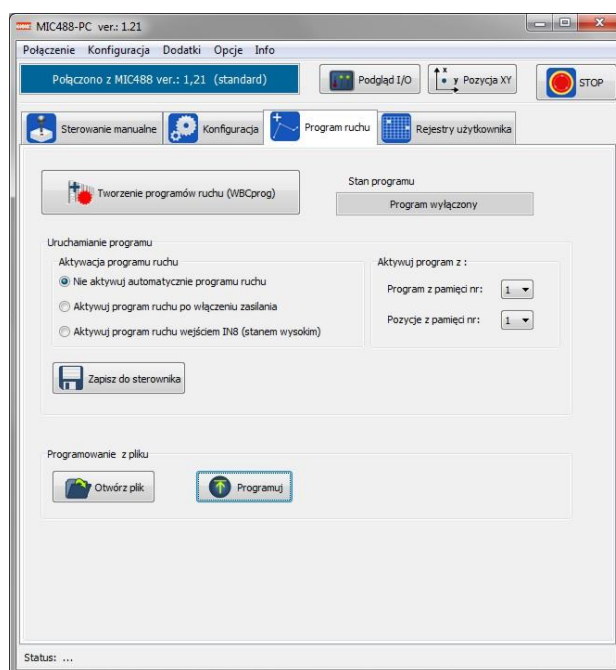
### 5.1 Wstęp

Programowanie sterownika odbywa się przy wykorzystaniu wbudowanej w SIC-KONFIGURATOR aplikacji **WBCprog**. Aplikację uruchamia się z zakładki „**Program ruchu**” przyciskiem „**Tworzenie programów ruchu (WBCprog)**”.

Programowanie polega na wprowadzaniu w języku tekstowym, zwanym **WBL** (Wobit Basic Language), prostych komend np.: „**PABS 10**” (ruch na pozycję absolutną 10) lub „**SET OUT2= ON**” (włączenie wyjścia OUT2).

Język ten dzięki prostym komendom tekstowym pozwala w intuicyjny i szybki sposób tworzyć programy dla sterownika. Z poziomu stworzonego programu możliwe jest dowolne sterowanie ruchem napędu, sterowanie uniwersalnymi wyjściami, reakcja na wejścia, zliczanie impulsów z enkodera, funkcje opóźnień czasowych, proste operacje matematyczne, operacje na zmiennych dostępnych przez rejestry MODBUS itp.

Użytkownik ma możliwość zapisania do pamięci sterownika 6 niezależnych programów składających się z maks. 1000 komend każdy. Wybrany program z pamięci sterownika może być uruchomiony automatycznie po włączeniu zasilania sterownika lub po aktywacji wejściem IN1. Takiej konfiguracji dokonuje się w oknie zakładki „**Program ruchu**”.

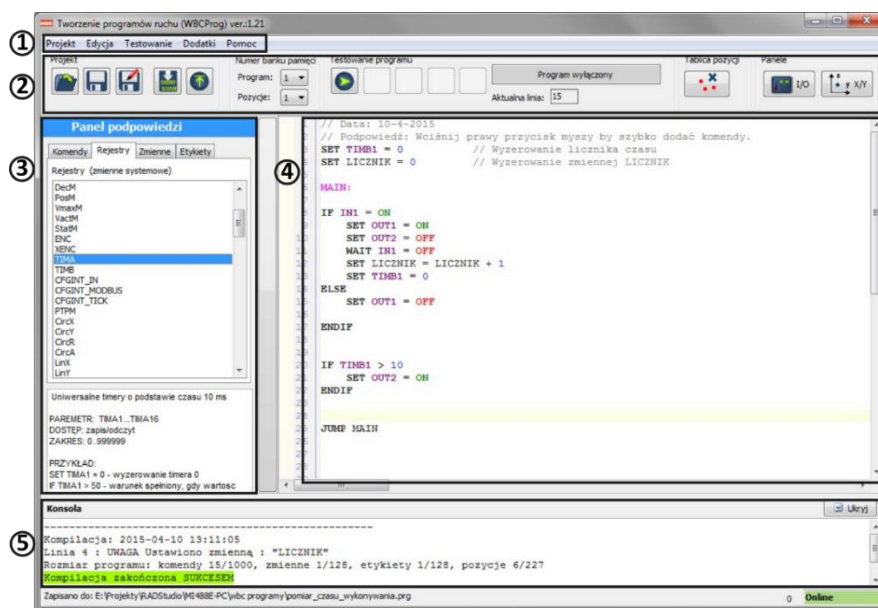


Rys. 19 Okno z ustawieniami uruchamiania programu z pamięci sterownika.

Ponadto możliwe jest uruchomienie wybranego programu za pomocą odpowiedniego rejestru MODBUS.

## 5.2 Opis programu WBCprog

### Okno główne



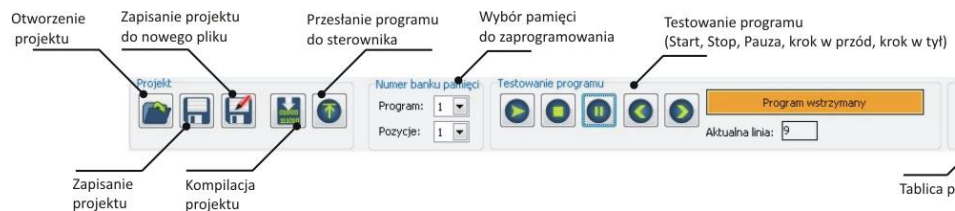
Rys. 20 Główne okno programu WBCprog.

#### 1) Pasek narzędzi:

Zawiera dostęp do wszystkich funkcji programu.

#### 2) Pasek skrótów:

Zawiera skróty do najważniejszych funkcji programu. W części „**testowanie programu**” znajdują się przyciski pozwalające na uruchomienie programu w celu przetestowania jego pracy. Obok przycisków wyświetlana jest także informacja o aktualnym stanie programu oraz ewentualnych błędach.



Rys. 21 Pasek skrótów.

#### 3) Panel odpowiedzi:

Zawiera spis wszystkich komend oraz rejestrów kompilatora WBCprog (zakładki **Komendy** i **Rejestry**). Pokazuje także użyte w programie własne zmienne (zakładka **Zmienne**) oraz etykiety (zakładka **Etykiety**).


#### 4) Okno edytora kodu:

Edytor tekstowy służący do wprowadzania komend. Edytor posiada funkcję wyróżniania wprowadzanych komend/zmiennych ułatwiając pisanie programu.

#### 5) Okno informacji o kompilacji programu:

Pokazuje informacje o skompilowanym programie (rozmiar programu itp.) lub informacje o błędach.

### Zapisywanie i otwieranie projektu


Przycisk **Zapisz**  powoduje zapisanie zmian w projekcie.

Przycisk **Zapisz jako**  powoduje zapisanie projektu do nowego pliku.

Przy zapisywaniu projektu tworzone są dwa pliki (o takiej samej nazwie, ale innym rozszerzeniu):



- Plik główny z programem (rozszerzenie **.prg**).
- Plik z aktualnymi ustawieniami sterownika (rozszerzenie **.cfg**).

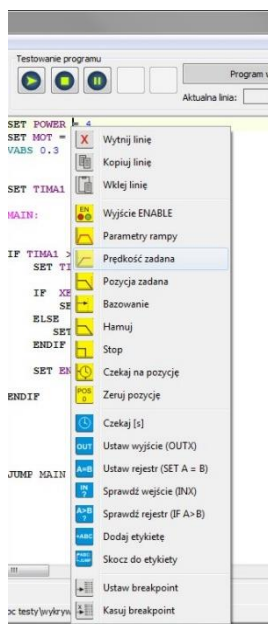
Przycisk **Otwórz**  powoduje otworenie projektu (otwieramy plik z rozszerzeniem **.prg**).



Jeśli w katalogu z projektem nie ma pliku konfiguracyjnego (**\*.cfg**) program wyświetli komunikat o braku tego plików. Zapisanie projektu spowoduje automatyczne utworzenie pliku konfiguracji z aktualnymi ustawieniami sterownika.


### Menu szybkich komend

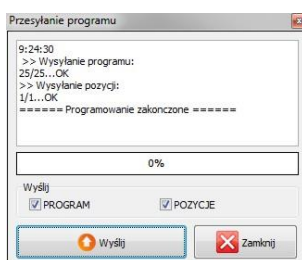
Kliknięcie prawym przyciskiem myszy w wybranej linii programu w oknie edytora (5) powoduje pojawienie się menu, które pozwala na dodanie do programu wybranych komend.



Rys. 22 Menu szybkich komend.

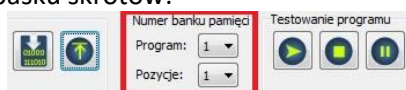
### Przesyłanie programu do sterownika


W celu zaprogramowania sterownika należy wcisnąć przycisk **Wyślij do sterownika** .



Rys. 23 Okno przesyłania programu do sterownika.






Po wciśnięciu przycisku **Wyślij** nastąpi przesłanie programu do sterownika. Program zostanie zapisany w pamięci sterownika o numerze ustawionym na pasku skrótów:



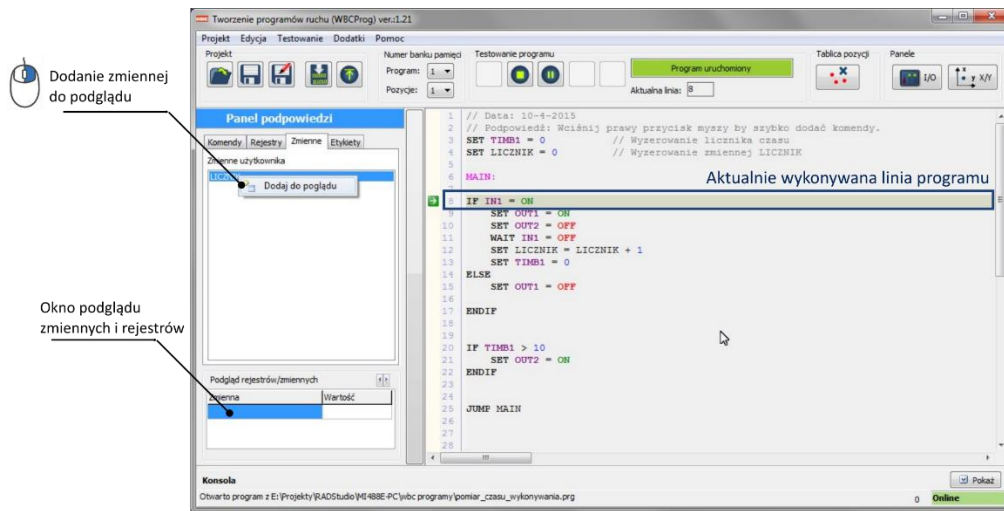
Przed wysłaniem do sterownika program jest automatycznie kompilowany. Nie ma więc konieczności jego wcześniejszej kompilacji za pomocą przycisku .



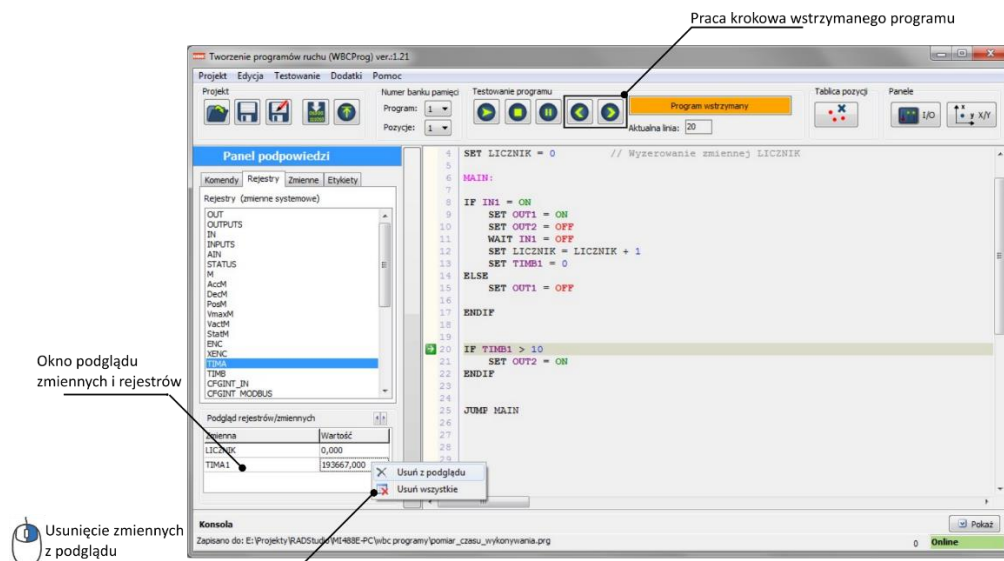
## Uruchamianie i testowanie programu

Przesłany do sterownika program może zostać uruchomiony ręcznie za pomocą przycisku . Po uruchomieniu okno edytora zostaje zablokowane. Za pomocą przycisku  można wstrzymać pracę programu, a następnie przyciskami   sterować krokowo jego pracą. Aktualnie wykonywana linia oznaczana jest obrazkiem .

Podczas działania programu możliwe jest podejrzenie aktualnych wartości wybranych rejestrów i zmiennych (maks. 12). W celu dodania do podglądu na **Panelu odpowiedzi** należy kliknąć prawym przyciskiem myszy na interesujący nas rejestr (w zakładce **Rejestry**) lub zmienną (w zakładce **Zmienne**) i wcisnąć „**Dodaj do podglądu**”.



Rys. 24 Sposób dodawania rejestrów / zmiennych do podglądu.



Rys. 25 Usunięcie z podglądu.

## 5.3 Opis języka WBL

Język **WBL** (Wobit Basic Language) to prosty język tekstowy zbliżony do BASIC. Ułatwia on tworzenie programów realizujących ruch napędów oraz daje większą elastyczność w stosunku do innych standardowych języków.

Poniżej znajdują się najważniejsze informacje związane z programowaniem w języku **WBL** za pomocą programu **WBCprog**.

## Kompilacja programu

Kompilacja programu powoduje wygenerowanie danych na podstawie wprowadzonych komend, które są zrozumiałe dla sterownika. Skompilowany program, jeśli nie zawiera błędów, może zostać przesłany do sterownika.

## Działanie programu w sterowniku

Stworzony program wykonywany jest przez sterownik komenda po komendzie. Program powinien być „zapętłony” za pomocą funkcji skoków. Jeśli program po ostatniej komendzie nie wykona skoku do wcześniejszych komend zostanie zakończony. **Zakończony program powoduje wyłączenie wszystkich wyjść oraz zatrzymanie i wyłączenie napędów.**

## Komentarze w programie

Wprowadzenie znaków „//” przed dowolną komendą lub opisem spowoduje, że będzie on pominięty podczas kompilacji programu. Przykład:

```
SET OUT1 = ON
//SET OUT2 = ON           // Ta linia programu zostanie pominięta
```

Umieszczenie fragmentu programu między znakami „/\*” oraz „\*/” spowoduje, że zostanie on ominięty na etapie kompilowania programu. Przykład:

```
SET OUT1 = ON
/*
SET OUT1 = ON           // Te linie programu zostaną pominięte
SET OUT2 = ON
*/
```

## Wartości zmiennoprzecinkowe

Wartości liczbowe z przecinkiem należy wprowadzać używając znaku kropki „.” Symbol przecinka spowoduje błąd podczas kompilacji.

```
SET VMAX = 1.45           // Wartość zmiennoprzecinkowa
```

## Komendy

Komenda to linia w programie, która zostaje odpowiednio zinterpretowana przez sterownik i wykonana. Komenda może zawierać dodatkowe parametry lub nie.

```
SET VMAX = 2             // Komenda SET użyta do zapisania wartości do rejestru prędkości maks. napędu
PABS -5                  // Komenda PABS zadająca pozycję absolutną
WAITPOS                  // Komenda WAITPOS oczekiwania na dojazd do pozycji (bez parametrów)
```

Spis wszystkich komend znajduje się w rozdziale **Błąd! Nie można odnaleźć źródła odwołania.**

## Definicje

Definicje pozwalają nadać własną nazwę dla stałej wartości liczbowej lub rejestru. By stworzyć definicje należy przed jej nazwą dodać znak „#”, a za nazwą wprowadzić wartość lub rejestr, któremu ma odpowiadać np.:

```
#POWTORZEN_X 10           // Stała wartość 10 jako nazwa "PREDKOSC_MAX"
#PREDKOSC_MAX 15         // Rejestr AccM1 jako nazwa "PREDKOSC_M1"
#HMI_PREDKOSC $R50      // Rejestr MODBUS 50 jako nazwa "HMI_PREDKOSC"
...
IF HMI_PREDKOSC < PREDKOSC_MAX
SET VMAX = HMI_PREDKOSC
```

Ponadto program posiada kilka zdefiniowanych na stałe wartości:

Definicja	Wartość	Opis
OFF	0	Stan wyłączony
ON	1	Stan włączony
M_OFF	0	Napęd wyłączony
M_ON	1	Napęd włączony
M_SPEED	2	Zadana prędkość
M_POS_SEARCH	3	Ruch do zadanej pozycji
M_POS_OK	4	Zadana pozycja osiągnięta
M_POS_ERROR	5	Błąd pozycji
M_POS_HOMING	6	Bazowanie
M_POS_CORRECTION	8	Korekcja pozycji
RIS	1	Zbocze rosnące przerwania
FAL	2	Zbocze malejące przerwania
RISFALL	3	Zbocze rosnące i malejące

## Etykiety i funkcje skoków

Skoki między komendami pozwalają na realizację bardziej skomplikowanych funkcji sterowania np.: wykonywania określonej ilości powtórzeń, ponowienia wybranego fragmentu programu czy jego realizacji w zależności od spełnienia warunku. Skoki w programie realizowane są do tzw. **etykiet** – czyli nazw dodanych w dowolnych fragmentach programu, zakończonych znakiem dwukropka „:” np.:

```

FUNKCJA_1: // Etykieta o nazwie "FUNKCJA_1"
SET OUT1 = ON
SET OUT2 = OFF
RETURN
.. // zrób coś...
JUMP FUNKCJA_1 // Skok do etykiety „FUNKCJA_1”

```

## Nazwy etykiet i własnych zmiennych

Użytkownik może wprowadzać własne zmienne, nazwy etykiet itp. zbudowane z dowolnych znaków będących literami i cyframi. Utworzone nazwy nie mogą być takie same jak nazwy komendy i rejestrów. Spis zarezerwowanych nazw znajduje się w rozdziale Błąd! Nie można odnaleźć źródła odwołania..

## Rejestry

Rejestry to zmienne, które używane są przez sterownik do kontroli napędów, odczytu wejść, sterowania wyjść itp. Dostępne są także uniwersalne rejestry użytkownika, pozwalające na komunikację w protokole MODBUS-RTU z urządzeniami zewnętrznymi.

Spis wszystkich rejestrów znajduje się w rozdziale Błąd! Nie można odnaleźć źródła odwołania..

## Rejestry MODBUS użytkownika

Użytkownik ma dostęp do 500 rejestrów (komórek pamięci) do których może zapisywać lub odczytywać dowolne wartości. Dostęp do tych rejestrów jest możliwy także przez interfejsy RS232/RS485 w protokole MODBUS-RTU. Wartości zapisywane/odczytywane przez MODBUS mogą być typu INT, DINT, REAL. By poprawnie interpretować różne typy danych przed adresem rejestru należy użyć przedrostka:

**\$I** – dla wartości typu INT  
**\$D** – dla wartości typu DINT  
**\$R** – dla wartości typu REAL

```

SET $R0 = 10.2 // Zapisanie do rejestru 0 (typu REAL) wartości 10,2
...
IF $I200 > 50 // Sprawdzenie, czy wartość (INT) w rejestrze 200 > 50
...
PABS M1 $R20 // Zadanie pozycji absolutnej (typu REAL) z rejestru 20

```

## Przerwania

Przerwania umożliwiają przerwanie aktualnie wykonywanej linii programu (np. na skutek zmiany sygnału na wybranym wejściu IN) i skok do zdefiniowanej etykiety. Pozwala to na szybką reakcję sterownika na sygnały zewnętrzne.

Przerwanie może być wykonane dla:

- zmiany stanu wejść IN1...INX
- zapisu przez Modbus rejestrów użytkownika (0...500).
- cykliczne przerwanie realizowane z okresem 10ms, 100ms i 1000ms

Przerwanie powoduje skok do określonej etykiety, która musi być dodana do programu:

Źródło przerwania	Nazwa etykiety
Zmiana stanu na wejściu IN1 ... INX	INT_IN1 ... INT_INX
Zapis przez Modbus do rejestru 0...500	INT_MODBUS
Przerwanie cykliczne 10ms	INT_TICK
Przerwanie cykliczne 100ms	INT_TICK_100MS
Przerwanie cykliczne 1000ms	INT_TICK_1000MS

Etykieta przerwania może kończyć się komendą RETURN albo JUMP. Jeśli kończy się komendą JUMP skok musi nastąpić do etykiety, która nie jest zakończona komendą RETURN (np. pętla główna programu). Podczas trwania przerwania (dopóki nie pojawi się komenda RETURN albo JUMP) inne przerwania są zablokowane.

By aktywować przerwanie od danego sygnału należy zapisać do odpowiedniego rejestru konfiguracyjnego np.:

```
SET CFGINT_IN1 = RIS // Przerwanie od zmiany stanu z 0 na 1 na wejściu IN1
SET CFGINT_MODBUS = HIGH // Przerwanie od zapisu do rejestrów Modbus (0...500)
SET CFGINT_TICK = HIGH // Przerwanie czasowe wykonywane co 10 ms
```

Do konfiguracji przerwania można użyć następujących wartości:

Przerwania od wejść IN1..INX:

**0 (OFF)** – przerwania wyłączone

**1 (RIS)** – przerwanie włączone na zbocze rosnące (zmiana sygnału z 0 na 1 na wejściu)

**2 (FALL)** – przerwanie włączone na zbocze opadające (zmiana sygnału z 1 na 0 na wejściu)

**3 (RISFALL)** – przerwanie włączone na zbocze rosnące i opadające (zmiana sygnału z 0 na 1 albo 1 na 0 na wejściu)

Przerwania od zapisu rejestrów użytkownika przez MODBUS, przerwanie cykliczne

**0 (OFF)** – przerwania wyłączone

**4 (HIGH)** – przerwanie włączone

## 6. Przykłady programów

Poniżej zestawiono przykłady wykorzystania poszczególnych funkcji sterownika SIC184. Dodatkowe przykłady można znaleźć w katalogu programu o nazwie **WBC przykłady**.

### 6.1 Obsługa wejść / wyjść

Wykorzystywane rejestry:

- **OUTX** – pojedyncze wyjście
- **OUTPUTS** – wszystkie wyjścia
- **INX** – pojedyncze wejście
- **INPUTS** – wszystkie wejścia
- **MX\_EN** – sygnał ENABLE sterowania napędem

#### Sterowanie wyjściami

```
SET OUT1 = ON           // Włączenie wyjścia O1
SET OUT1 = OFF          // Wyłączenie wyjścia O1
SET OUTPUTS = 0        // Wyłączenie wszystkich wyjść
SET MOT = ON           // Załączenie napędu
```

#### Sprawdzenie stanu wejścia

```
IF IN2 = ON            // Jeśli wejście I2 aktywne
...                   // ...
ENDIF                 // Koniec warunku
```

#### Oczekiwanie na wejście

```
WAIT IN2 = ON          // Czekaj, aż wejście I2 będzie aktywne
```

### 6.2 Odczyt wejść analogowych (0-10V)

Wykorzystywane rejestry: **AINX** – wartość z wejścia analogowego

```
VABS AIN1              // Zadanie prędkości dla napędu = wartości z wejścia AIN1
...
IF AIN1 < 5,0          // Sprawdzenie czy napięcie na AIN2 mniejsze od 5.
...                   // zrób coś...
ENDIF
```



Wartość z rejestrów AIN1/AIN2 może być przeskalowana w ustawieniach (zakładka **Konfiguracja** -> **We 0-10V**)

### 6.3 Sterowanie napędami

Wykorzystywane rejestry:

- **ACC, DEC, VMAX** – parametry rampy (przyspieszenie, hamowanie, prędkość maks.)
- **POS** – pozycja aktualna
- **VACT** – prędkość aktualna
- **MOT** – sterowanie danym napędem (załączenie wyjścia ENABLE, funkcja STOP i BRAKE)

#### Ustawienie parametrów ruchu (rampy)

```
SET ACC = 10           // Ustawienie przyspieszenia dla napędu
SET DEC = 10           // Ustawienie hamowania dla napędu
SET VMAX = 50          // Ustawienie prędkości maks. dla napędu
```

## Załączenie napędu

```
SET MOT = ON // Włączenie napędu
```

## Zatrzymanie/zahamowanie napędu w ruchu

```
STOP // Zatrzymanie napędu (nagle)  
BRAKE // Zahamowanie napędu (wyhamowanie do prędkości 0)
```

## Bazowanie napędu

```
...  
HOME -2 // Rozpoczęcie bazowania napędu w kierunku ujemnym (do  
WAITPOS // krańcówki KL)
```

## Zadanie prędkości

```
VABS 5 // Zadanie prędkości 5 dla napędu  
VABS -2,5 // Zadanie prędkości -2,5 dla napędu (ruch w stronę „ujemną”)  
VREL -0,5 // Zmniejszenie prędkości napędu o 0,5  
VREL 0,5 // Zwiększenie prędkości napędu o 0,5
```

## Zadanie pozycji

```
PABS 10 // Zadanie pozycji 10 dla napędu M1  
WAITPOS // Oczekiwanie na osiągnięcie zadanych pozycji  
  
PREL -1 // Zmniejszenie pozycji napędu M2 o 1
```

## Oczekiwanie na osiągnięcie zadanej pozycji

Sposób 1 – komenda **WAITPOS** oczekująca na osiągnięcie pozycji dla wszystkich napędów

```
PABS 10 // Zadanie pozycji 10 dla napędu M1  
WAITPOS // Oczekiwanie na osiągnięcie pozycji
```

Sposób 2 – sprawdzanie rejestru **MOT**. Gdy **MOT = 0** to napęd nie znajduje się w ruchu.

```
PABS 10 // Zadanie pozycji 10 dla napędu  
WAIT MOT = 0 // Oczekiwanie, aż napęd nie będzie w ruchu
```

Sposób 3 – sprawdzenie rejestru statusowego **STATUS**.

```
PABS 10 // Zadanie pozycji 10 dla napędu M1  
WAIT STATUS = M_POS_OK // Oczekiwanie, aż napęd osiągnie zadana pozycję
```



Jeśli dany napęd jest wyłączony (sygnał EN wyłączony) to komenda **WAITPOS** pomija sprawdzanie jego pozycji.

## Zerowanie pozycji

```
SET POS = 0 // Wyzerowanie pozycji napędu
```

# 7. Spis komend i rejestrów w WBL

## Komendy podstawowe

Komenda	Opis	Składnia
<b>SET</b>	Ustawia wyjście, zmienną, rejestr, wykonuje operacje matematyczne.	<b>SET X = Y</b> <b>SET X = Y operacja Z</b> <b>SET X = funkcja(Y)</b> <b>operacja</b> – dostępne operacje opisane niżej.
<b>IF / ELSE /ENDIF</b>	Porównuje stan wejść, wyjść, zmiennych, rejestrów. Gdy warunek spełniony wykonanie kolejnych linii programu, aż do komendy ENDIF (do ELSE jeśli istnieje). Jeśli nie przeskoczenie do ENDIF (ELSE jeśli istnieje)	<b>IF X warunek Y</b> .. gdy warunek spełniony <b>ENDIF</b> lub <b>IF X warunek Y</b> .. gdy warunek spełniony <b>ELSE</b> ... gdy warunek niespełniony <b>ENDIF</b> <b>warunek</b> – dostępne warunki opisane niżej
<b>WHILE / ENDWHILE</b>	Porównuje stan wejść, wyjść, zmiennych, rejestrów. Dopóki warunek spełniony występuje zapętlenie między WHILE, a ENDWHILE. Jeśli warunek niespełniony następuj wyjście z ENDWHILE.	<b>WHILE X warunek Y</b> .. wykonywanie dopóki warunek spełniony <b>ENDWHILE</b> <b>warunek</b> – dostępne warunki opisane niżej
<b>WAIT</b>	Oczekuje na stan wejść, wyjść, zmiennych, rejestrów. Jeśli warunek spełniony przejście do następnej linii. Jeśli nie oczekuje, aż zostanie spełniony.	<b>WAIT X warunek Y</b> <b>warunek</b> – dostępne warunki opisane niżej
<b>DELAY</b>	Wprowadza opóźnienie czasowe w sekundach.	<b>DELAY X</b>
<b>JUMP</b>	Skok do istniejącej etykiety programu.	<b>JUMP NAZWA_ETYKIETY</b>
<b>RETURN</b>	Powrót do linii za ostatnią komendą skoku.	<b>RETURN</b>
<b>END</b>	Zakończenie działania programu	<b>END</b>
<b>PULSE</b>	Zmiana stanu wyjścia po upływie określonego czasu	<b>PULSE(REJESTR_WYJSCIA, CZAS)</b>

Parametry X, Y, Z mogą być zmiennymi użytkownika, rejestrami urządzenia, rejestrami modbus lub wartościami stałymi.

#### Spis dostępnych operacji i warunków matematycznych

Operacje matematyczne	Funkcje matematyczne	Warunki
<ul style="list-style-type: none"> <li>➤ + (dodawanie),</li> <li>➤ - (odejmowanie),</li> <li>➤ * (mnożenie),</li> <li>➤ / (dzielenie),</li> <li>➤ % (reszta z dzielenia),</li> <li>➤   (suma bitowa),</li> <li>➤ &amp; (iloczyn bitowy),</li> <li>➤    (suma logiczna),</li> <li>➤ &amp;&amp; (iloczyn logiczny),</li> <li>➤ &gt;&gt; (przesunięcie bitowe w prawo),</li> <li>➤ &lt;&lt; (przesunięcie bitowe w lewo),</li> </ul>	<ul style="list-style-type: none"> <li>➤ <b>sin</b> (sinus),</li> <li>➤ <b>cos</b> (cosinus),</li> <li>➤ <b>tg</b> (tangens),</li> <li>➤ <b>ctg</b> (cotangens),</li> <li>➤ <b>asin</b> (arcus sinus),</li> <li>➤ <b>acos</b> (arcus cosinus),</li> <li>➤ <b>sqrt</b> (pierwiastek kwadratowy),</li> <li>➤ <b>min</b> (wartość min),</li> <li>➤ <b>max</b> (wartość maks),</li> </ul> <p>Funkcje trygonometryczne przyjmują wartości w stopniach.</p>	<ul style="list-style-type: none"> <li>➤ = (równy),</li> <li>➤ &lt; (mniejszy),</li> <li>➤ &gt; (większy),</li> <li>➤ &lt;= (mniejszy równy),</li> <li>➤ &gt;= (większy równy)</li> </ul>

#### Komendy ruchu

Komenda	Opis	Składnia
<b>HOME</b>	Wykonuje bazowanie napędu.	<b>HOME Y</b> Y – prędkość bazowania
<b>VABS</b>	Zadanie prędkości absolutnej (równej wprowadzonej wartości).	<b>VABS Y</b> Y – prędkość absolutna
<b>VREL</b>	Zwiększenie (zmniejszenie) aktualnej prędkości.	<b>VREL Y</b> Y – prędkość relatywna



<b>PABS</b>	Zadanie pozycji absolutnej (równej wprowadzonej wartości).	<b>PABS Y</b> Y – pozycja absolutna
<b>PREL</b>	Zwiększenie (zmniejszenie) aktualnej pozycji napędu.	<b>PREL Y</b> Y – pozycja relatywna
<b>BRAKE</b>	Zatrzymanie napędu (ustawienie prędkości 0).	<b>BRAKE</b>
<b>STOP</b>	Zatrzymanie napędu (szybkie).	<b>STOP</b>
<b>WAITPOS</b>	Oczekiwanie na osiągnięcie pozycji we wszystkich napędach. UWAGA: Jeśli dany napęd nie jest aktywny (SET MOT = OFF) to jest on pomijany przy sprawdzaniu pozycji.	<b>WAITPOS</b>

## Rejestry ogólne

Rejestr	Opis	Dostęp	Zakres wartości
OUTX	Dostęp do wybranego wyjścia uniwersalnego sterownika.	R/W	0 – wyjście wyłączone, > 0 – wyjście włączone
OUTPUTS	Binarny dostęp do wszystkich wyjść uniwersalnych	R/W	0 ... 65535
INX	Dostęp do wybranego wejścia uniwersalnego sterownika.	R	0 – wyjście nieaktywne > 0 – wyjście aktywne
INPUTS	Binarny dostęp do wszystkich wejść uniwersalnych.	R	0 ... 65535
AINX	Dostęp do wejść analogowych sterownika	R	REAL
TIMAX	Uniwersalny timer o podstawie czasu 10 ms	R/W	0 - 999999
TIMBX	Uniwersalny timer o podstawie czasu 1 sek	R/W	0 - 999999
CFGINT_INX	Konfiguracja przerwania dla wejść IN1...IN8 X = 1...8	R/W	0 (OFF) – przerwanie wyłączone 1 (RIS) – przerwanie na zbocze rosnące 2 (FALL) – przerwanie na zbocze opadające 1 (RISFALL) – przerwanie na oba zbocza
CFGINT_MODBUS	Konfiguracja przerwania dla zapisu danych do rejestrów użytkownika Modbus.	R/W	0 (INT_OFF) - wyłączone 4 (INT_HIGH) - włączone
CFGINT_USER_REGX	Konfiguracja przerwania od zapisu wartości !=0 do rejestru użytkownika o adresie 0,1,2 lub 3. Po wystąpieniu przerwania rejestr jest automatycznie zerowany. X = 0...3	R/W	0 (INT_OFF) - wyłączone 4 (INT_HIGH) - włączone
CFGINT_TICK CFGINT_TICK_100MS CFGINT_TICK_1000MS	Konfiguracja przerwania cyklicznego, wykonywanego co 10ms, 100ms lub 1000ms (1s)	R/W	0 (INT_OFF) - wyłączone 4 (INT_HIGH) - włączone

## Rejestry napędu

Rejestr	Opis	Dostęp	Zakres wartości
MOT	Sterowanie wyjściem ENABLE. Włączenie napędu, sprawdzenie czy napęd jest w ruchu.	R/W	Zapis: 0 (OFF) – wyłączenie napędu 1 (ON) – włączenie napędu Odczyt: 0 – napęd w spoczynku 1 – napęd w ruchu
STATUS	Status pracy napędu.	R	0 (M_OFF) – napęd wyłączony 1 (M_ON) – napęd włączony 2 (M_SPEED) – tryb prędkości 3 (M_POS_SEARCH) – tryb pozycji 4 (M_POS_OK) – pozycja osiągnięta 5 (M_POS_ERROR) – błąd pozycji 6 (M_POS_HOMING) – bazowanie 8 (M_POS_CORRECTION) – korekcja pozycji 9 (M_POS_LIM_L) – osiągnięta skrajna pozycja L 10 (M_POS_LIM_R) – osiągnięta skrajna pozycja R
POWER	Moc napędu (%) odpowiadająca procentowej wartości prądu sterownika	R/W	2..100
POWRED	Redukcja mocy sterownika przy bezczynności.	R/W	2...100
VMAX	Prędkość maksymalna dla trybu pozycji.	R/W	REAL
VACT	Prędkość aktualna napędu.	R	REAL
ACC	Przyspieszenie dla rozpędzania w trybie zadanej pozycji. Przyspieszenie dla rozpędzania i hamowania w trybie zadanej prędkości.	R/W	REAL
DEC	Przyspieszenie dla hamowania w trybie zadanej pozycji.	R/W	REAL
POS	Pozycja aktualna napędu.	R/W	REAL
POSLIML	Programowe ograniczenie pozycji w stronę ujemną (L)	R/W	REAL
POSLIMR	Programowe ograniczenie pozycji w stronę dodatnią (R)	R/W	REAL
ENC	Impulsy z enkodera.	R/W	DINT
XENC	Impulsy z enkodera przeliczone na pozycję napędu.	R	REAL

## Rejestry MODBUS użytkownika

Rejestr	Opis	Dostęp	Zakres wartości
\$I0 - \$I499	Rejestry użytkownika (wartości typu INT)	R/W	INT (-32768...32767)
\$D0 - \$D498	Rejestry użytkownika (wartości typu DINT)	R/W	DINT (-2147483648...2147483647)
\$R0 - \$R498	Rejestry użytkownika (wartości typu REAL)	R/W	REAL (zmiennoprzecinkowa)

UWAGA: rejestry \$I, \$D i \$R zajmują tę samą przestrzeń pamięci.

UWAGA: rejestry nie są domyślnie zapamiętywane po wyłączeniu zasilania. By zapisać rejestry do pamięci nielotnej należy zapisać bit o adresie 4000 przez Modbus.

## 8. Parametry techniczne

Opis	Parametr
Zasilanie	VL+ (zasilanie sterowania i wyjść): 12 ... 36 VDC, min 100mA VM+ (zasilanie silnika): 12 ... 36 VDC, maks. 2,5A
Wyjścia tranzystorowe PNP OC	Maks. 5 A / wyjście oraz 5 A na wszystkie wyjścia
Wejścia optoizolowane (IN1... IN6)	5 ... 24 VDC (stan niski – maksymalnie 1 V) Filtracja wejścia: minimalny czas trwania sygnału 10ms, konfigurowana programowo.
Wejście enkodera (IN7..IN9)	Sygnały A,B: stan niski <1V, stan wysoki 3-24V. Maks. częstotliwość wejściowa: 100Khz Sygnał C: stan niski <5V, stan wysoki >10V
Wejście analogowe AIN1 0..10V	Rozdzielczość: 12bit Rezystancja wejściowa 22k
Prąd silnika na fazę dla wbudowanego sterownika mocy	maks. 4 A (płynnie regulowany)
Podział kroku dla wbudowanego sterownika mocy	½, 1/5, 1/8, 1/10, 1/16, 1/20, 1/32, 1/40, 1/64
Generator trajektorii	Kontrola pozycji, prędkości, przyspieszenia
Sygnały wyjściowe CLK, DIR, EN dla zewnętrznego sterownika mocy	Stan niski: 0V, stan wysoki: 5V Maks. częstotliwość: 200khz, szerokość impulsu: 5us
Pamięć programu	2000 komend
Pamięć użytkownika z dostępem przez modbus	500 słów 16-bitowych z możliwością zapisu do pamięci nieulotnej
Komunikacja	<b>RS485</b> , domyślnie parametry 38400bps, 8:n:1, protokół wewnętrzny, protokół MODBUS-RTU <b>USB: 1.1 2.0</b>
Zakres temperatur pracy	-5°C...+50°C
Obudowa	Wymiary: 120 x 101 x 23 mm Stopień ochrony: IP20 Mocowanie: uchwyt na szynę DIN