

SIC184 Protokół MODBUS-RTU (v1.10)

Spis treści

1.	Informacje wstępne.....	1
2.	Format danych i zaimplementowane funkcje MODBUS.....	1
3.	Opis rejestrów i funkcji.....	2
3.1	Odczyt stanu wejść/wyjść	2
3.2	Odczyt i zapis X-rejestrów (prędkość, przyśpieszenie, pozycja, parametry napędu, status napędu).....	3
3.3	Zapis pojedynczego bajtu (sterowanie napędem, wyjściami)	5
4.	Opis sterowania napędem	5
4.1	Status napędu.....	5
5.	Dodatkowe funkcje	6
5.1	Rejestry użytkownika	6
5.1.1	Odczyt rejestrów z poziomu programu wykonywanego przez sterownik.....	6
5.1.2	Odczyt rejestrów przez komendy pozycji i prędkości MODBUS	6
5.1.3	Zapis rejestrów użytkownika do pamięci nieulotnej.....	7
6.	Historia zmian	7

1. Informacje wstępne

Sterownik SIC184 został wyposażony w złącze RS485, które pozwala na komunikację z urządzeniami zewnętrznymi w protokole MODBUS-RTU.

Do połączenia RS485 między sterownikami, a urządzeniem MASTER, szczególnie przy większych dystansach i prędkościach transmisji (>38400bps, >10m) zaleca się użycie skrętki dwuprzewodowej, najlepiej dodatkowo ekranowanej.

Należy także pamiętać o dodaniu terminatora (rezystor o oporności 120Ω...470Ω włączony między linie A i B) na początku i końcu magistrali RS485.

Sterownik pozwala na komunikację z urządzeniami Master z prędkością odczytu/zapisu nie większą niż 100 ramek/sekundę dla prędkości transmisji 57600bps.



UWAGA!

Adresy rejestrów mogą ulec zmianie przy nowszych wersjach oprogramowania sterownika. Należy sprawdzić, czy wersja oprogramowania sterownika odpowiada opisanej w dokumentacji.



UWAGA!

Interfejs RS485 sterownika SIC184 nie jest optoizolowany. Należy więc zapewnić takie same potencjały między sterownikiem (sterownikami), a innymi urządzeniami na tej magistrali (wspólne masy zasilania). W przeciwnym wypadku mogą nastąpić problemy z komunikacją.

2. Format danych i zaimplementowane funkcje MODBUS

- Domyślny adres: 1
- Domyślna prędkość transmisji: **38400 b/s**
- Bity stopu: **2**, Parzystość: **brak**
- Timeout: **750µs** (maksymalny czas odstępu między kolejnymi bajtami w ramce)

Format ramki MODBUS-RTU

Znacznik początku	Adres	Funkcja	Dane	CRC	Znacznik końca
T ₁₋₂₋₃₋₄	8 bitów	8 bitów	N x 8 bitów	16 bitów	T ₁₋₂₋₃₋₄

T₁₋₂₋₃₋₄ - przerwa czasowa między kolejnymi ramkami minimum 1,750ms

Protokół MODBUS – zaimplementowane funkcje

Nr funkcji (hex)	Opis
01 (0x01)	Odczyt stanu wyjścia
02 (0x02)	Odczyt stanu wejścia
03 (0x03)	Odczyt N rejestrów
05 (0x05)	Zapis wejścia
06 (0x06)	Zapis 1 rejestru
16 (0x10)	Zapis N rejestrów

Opis typów zmiennych wykorzystywanych przez MODBUS-RTU w SIC184

Nazwa zmiennej	Opis	Rozmiar (bajty)	Zakres
BYTE	Liczba 1 bajtowa	1	0-255
WORD	Liczba 2 bajtowa	2	0-65535
DWORD	Liczba 4 bajtowa bez znaku	4	2 ³²
DINT	Liczba 4 bajtowa ze znakiem	4	-2 ³¹ ...(2 ³¹ -1)
REAL	Liczba zmiennoprzecinkowa	4	1.18*10 ⁻³⁸ ...3.40*10 ⁻³⁸ , 0, -3.40*10 ⁻³⁸ ... -1.18*10 ⁻³⁸

3. Opis rejestrów i funkcji

3.1 Odczyt stanu wejść/wyjść

Funkcja 01 (x0)- odczyt stanu wyjść

Adres	Nazwa	Tryb	Typ zmiennej	Opis
5021	OUT	R	BYTE	Odczyt stanu wyjść tranzystorowych Out1 i Out2. Funkcja zwraca 1 bajt, gdzie bit 0 – stan wyjścia OUT1, bit 1 – stan wyjścia OUT2.

Funkcja 02 (x0)- odczyt stanu wejść

Adres	Nazwa	Tryb	Typ zmiennej	Opis
4000	IN	R	BYTE	Odczyt stanu wejść IN1, IN2, Home1, Home2, Start, Stop. Funkcja zwraca 1 bajt, gdzie: bit 0 – stan wyjścia IN1 bit 1 – stan wyjścia IN2 bit 2 – stan wyjścia HOME1 bit 3 – stan wyjścia HOME2 bit 4 – stan wyjścia START bit 5 – stan wyjścia STOP

Przykładowa ramka danych dla odczytu stanu wejść

Zapytanie (MASTER -> SIC184)		Odpowiedź (SIC184 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x02	Funkcja	0x02
Początkowy adres Hi	0xFA	Ilość bajtów	0x01
Początkowy adres Lo	0x00	Status wejść	BYTE
Ilość wejść Hi	0x00	CRC	16 bit
Ilość wejść Lo	0x06		
CRC	16 bitów		

3.2 Odczyt i zapis X-rejestrów (prędkość, przyspieszenie, pozycja, parametry napędu, status napędu)

Funkcja 03 (x4)- odczyt X rejestrów (R)

Funkcja 06 (x4)- zapis 1 rejestru (W1)

Funkcja 16 (x4)- zapis X rejestrów (W)

Adres	Nazwa	Tryb	Typ zmiennej	Opis
Rejestry z wartościami całkowitymi (DWORD, DINT)				
0-1	VMIN	R/W	DWORD	Prędkość minimalna (imp/sek)
2-3	VMAX	R/W	DWORD	Prędkość maksymalna (imp/sek)
4-5	AMAX	R/W	DWORD	Przyspieszenie (imp /s ²)
6-7	XDEST	R/W	DINT	Pozycja docelowa (imp). Powoduje ruch do pozycji X [imp] względem pozycji 0.
8-9	XMOVE	R/W	DINT	Przesunięcie (imp). Powoduje ruch o X [imp] względem pozycji aktualnej (XACT).
10-11	XACT	R	DINT	Pozycji aktualna (imp)
12-13	VDEST	R/W	DINT	Prędkość docelowa (imp /s)
14-15	VACT	R	DINT	Prędkość aktualna (imp /s)
16-17	MOTOR_HOME	R/W	DINT	Bazowanie pozycji (prędkość bazowania imp/s)
18-19	ENC_ACT	R	DINT	Wartość pozycji odczytana z enkodera przeliczona na impulsy wygenerowane dla silnika (imp).
20	REG_SPEED	W	WORD	Wybór rejestru użytkownik, spod którego ma zostać odczytana i zadana prędkość (50...98 lub 0xFF00 + 50...98) Gdy starszy bajt rejestru jest ustawiony (rejestr + 0xFF00) wartość z rejestru użytkownika dekodowana jest na liczbę REAL. W przeciwnym wypadku na DINT
21	REG_POS	W	WORD	Wybór rejestru użytkownik, spod którego ma zostać odczytana i zadana pozycja (50...98 lub 0xFF00 + 50...98) Gdy starszy bajt rejestru jest ustawiony (rejestr + 0xFF00) wartość z rejestru użytkownika dekodowana jest na liczbę REAL. W przeciwnym wypadku na DINT
22	MOTOR_STEPS	R/W1	WORD	Podział kroku sterownika (2/8/10/16/20/32/40/64)
23	MOTOR_CURRENT	R/W1	WORD	Prąd sterownika (10..100)
24	MOTOR_IS_GET_HOME	R	WORD	Status bazowania napędu: 0 – brak bazowania, 1 – rozpoczęcie, 2 – w toku, 3 – odjazd (tylko dla bazowania precyzyjnego), 4 – bazowanie zakończone
25	MOTOR_STATUS	R	WORD	Rejestr statusu napędu: 0 – napęd wyłączony; 1- napęd włączony, 2 – tryb zadanej prędkości, 3- tryb poszukiwania pozycji, 4- pozycja osiągnięta, 5- błąd pozycji, 6 – bazowanie, 7 – przeciążenie, 8 – korekcja pozycji. Dokładny opis znajduje się w rozdziale 4.1
Rejestry z wartościami zmiennoprzecinkowymi (REAL)				
26-27	REAL_VMIN	R/W	REAL	Prędkość minimalna (obr/sek)
28-29	REAL_VMAX	R/W	REAL	Prędkość maksymalna (obr /sek)
30-31	REAL_AMAX	R/W	REAL	Przyspieszenie (obr /s ²)
32-33	REAL_XDEST	R/W	REAL	Pozycja docelowa (imp). Powoduje ruch do pozycji X [obr] względem pozycji 0.
34-35	REAL_XMOVE	R/W	REAL	Przesunięcie (obr). Powoduje ruch o X [obr] względem pozycji aktualnej (XACT).
36-37	REAL_XACT	R	REAL	Pozycja aktualna (obr)
38-39	REAL_VDEST	R/W	REAL	Prędkość docelowa (obr/s)
40-41	REAL_VACT	R	REAL	Prędkość aktualna (obr /s)
42-43	REAL_MOTOR_HOME	R/W	REAL	Bazowanie pozycji (prędkość bazowania obr /s)
44-45	REAL_ENC_ACT	R	REAL	Wartość pozycji odczytana z enkodera przeliczona na ilość obrotów silnika.
46	M_VDEST	R/W	INT	Prędkość docelowa (obr/min)
47	M_VACT	R	INT	Prędkość aktualna (obr/min)
50-100	USER_REGISTER	R/W		Rejestry użytkownika. Mogą zostać zapamiętane w pamięci nieulotnej sterownika poprzez rejestr 5100 (SAVE_USER_REG)

**UWAGA!**

Adresy rejestrów rozpoczynają się od 0. Dla sterowników, w których adresowanie rozpoczyna się od 1 należy podawać wartości rejestrów z przesunięciem o 1 np.: rejestr VMIN: 1-2, VMAX 3-4 itd.

Rejestry typu **DWORD** i **DINT** dotyczące parametrów ruchu silnika (pozycja, prędkość, przyspieszenie) przechowują parametry ruch w jednostkach "impulsy", które są równoważne ilościom kroków wygenerowanych przez sterownik.

Przykładowo:

- rzeczywista prędkość silnika = 2 obr/sek.
- podział kroku sterownika silnika = 16,
- silnik standardowy 200 kroków / obrót

Rejestr VACT będzie zawierał liczbę: $2 * 16 * 200 = 6400$ [imp]

Analogicznie dla zapisu rejestru, chcąc zadać pozycję np. 14,2 obrotów do rejestru XDEST należy wpisać wartość (dla parametrów napędu jak wyżej): $14,2 * 16 * 200 = 45440$ [imp]

Rejestry typu **REAL** przechowują wartości prędkości, pozycji i przyspieszenia w odniesieniu do rzeczywistych obrotów silnika. Przykładowo by zadać pozycję 10,25 obr. do rejestru REAL_XDEST należy przesłać właśnie tą wartość.

Przykład ramka zadania pozycji w impulsach (Funkcja: 16, Adres rejestru: 6 (XDEST))

Zapis (MASTER -> SIC184)		Odpowiedź (SIC184 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x10	Funkcja	0x10
Adres rejestru Hi	0x00	Początkowy adres Hi	0x00
Adres rejestru Lo	0x06	Początkowy adres Lo	0x06
Ilość rejestrów Hi	0x00	Ilość rejestrów Hi	0x00
Ilość rejestrów Lo	0x02	Ilość rejestrów Lo	0x02
Ilość bajtów	0x04	CRC	16 bit
Rejestr 0x06 Hi	DINT (Bajt 1)		
Rejestr 0x06 Lo	DINT (Bajt 0)		
Rejestr 0x06 +1 Hi	DINT (Bajt 3)		
Rejestr 0x06 +1 Lo	DINT (Bajt 2)		
CRC	16 bit		

Bajt 0, Bajt 1 – młodsza część liczby double integer

Bajt 2, Bajt 3 – starsza część liczby double integer

Przykładowa ramka odczytu pozycji aktualnej w impulsach (Funkcja: 03, Adres rejestru: 10 (XACT))

Zapytanie (MASTER -> SIC184)		Odpowiedź (SIC184 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x03	Funkcja	0x03
Adres rejestru Hi	0x00	Ilość bajtów	0x04
Adres rejestru Lo	0x0A	Rejestr 0x0A Hi	DINT (Bajt 1)
Ilość rejestrów Hi	0x00	Rejestr 0x0A Lo	DINT (Bajt 0)
Ilość rejestrów Lo	0x02	Rejestr 0x0A+1 Hi	DINT (Bajt 3)
CRC	16 bitów	Rejestr 0x0A+1 Lo	DINT (Bajt 2)
		CRC	16 bit

**UWAGA!**

Wszystkie liczby 4 bajtowe typu **DINT**, **DWORD**, **FLOAT** zawierają się zawsze w **dwóch rejestrach**. Pierwszy rejestr zawiera młodszą część liczby, drugi starszą. Przykładowo by odczytać pozycję aktualną XACT należy odczytać rejestr o numerach 10 oraz 11, a następnie wykonać odpowiednią konwersję.

Konwersja 2 rejestrów (4 bajty) na liczbę 32 bitową (DINT, DWORD, FLOAT).

- RejestrX HI <-> Bajt1
- RejestrX LO <-> Bajt0
- RejestrX+1 HI <-> Bajt3
- RejestrX+1 LO <-> Bajt2

Liczba_32_bit = Bajt3<<24 + Bajt2<<16 + Bajt1<<8 + Bajt0

Lub **Liczba_32_bit** = RejestrX + Rejestr(X + 1)<<16

3.3 Zapis pojedynczego bajtu (sterowanie napędem, wyjściami)

Funkcja 0x05 – zapis pojedynczego bajtu (W)

Adres	Nazwa	Tryb	Opis
5000	MOTOR_DISABLE	W	Wyłączenie napędu
5001	MOTOR_ENABLE	W	Włączenie napędu
5002	MOTOR_STOP	W	Zatrzymanie napędu
5003	MOTOR_RESET	W	Zatrzymanie napędu, wyzerowanie licznika pozycji (XACT, ENC_ACT)
5010	START_PROG	W	Uruchomienie programu z pamięci sterownika
5011	STOP_PROG	W	Zatrzymanie programu
5012	PAUSE_PROG	W	Spauzowanie programu
5021	OUT1	W	Ustawienie wyjścia tranzystorowego 1
5022	OUT2	W	Ustawienie wyjścia tranzystorowego 2
5100	SAVE_USER_REG	W	Zapis rejestrów użytkownika (50..100) w pamięci nieulotnej SIC184

Przykładowa ramka załączająca napęd (Funkcja: 05, Adres rejestru: 5001)

Zapytanie (MASTER -> SIC184)		Odpowiedź (SIC184 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x05	Funkcja	0x05
Adres rejestru Hi	0x13	Adres rejestru Hi	0x13
Adres rejestru Lo	0x89	Adres rejestru Lo	0x89
Rejestr 0x00 Hi	0xFF	Rejestr 0x00 Hi	0xFF
Rejestr 0x00 Lo	0x00	Rejestr 0x00 Lo	0x00
CRC	16 bitów	CRC	16 bitów

4. Opis sterowania napędem

Po włączeniu zasilania napęd jest gotowy do pracy z domyślnymi parametrami ruchu (prędkość minimalna, maksymalna, przyspieszenie).

Przed rozpoczęciem pracy w trybie zadawania pozycji napęd należy zbazować (wymagane do określania pozycji względem pozycji zerowej). Służy do tego rejestr **MOTOR_HOME** (lub **REAL_MOTOR_HOME**). Proces bazowania może być kontrolowany przez sprawdzanie rejestru **MOTOR_IS_GET_HOME**, który zawiera informacje o postępie bazowania. Gdy bazowanie zostanie zakończone rejestr będzie zawierał wartość 4, w przeciwnym wypadku 1,2 lub 3, w zależności od aktualnego stanu i trybu bazowania.

Po zakończeniu bazowania można zadać pozycję bezwzględną za pomocą wpisania wartości pozycji do rejestru **XDEST** (lub **REAL_XDEST**).

Nie jest wymagane uruchamianie napędu komendą **MOTOR_ENABLE** przed zadaniem pozycji lub prędkości, gdyż jest to realizowane automatycznie.

Napęd nie jest wyłączany automatycznie po osiągnięciu zadanej pozycji. Można to zrealizować przez rejestr **MOTOR_DISABLE**.

4.1 Status napędu

Stan napędu może być kontrolowany przez odczyt rejestru statusowego **MOTOR_STATUS (R.25)**. Rejestr może przyjmować następujące wartości:

Wartość rejestru	Opis
0	Napęd wyłączony (silnik w spoczynku).
1	Napęd włączony (silnik w stanie postoju, trzymanie momentu).
2	Napęd w trybie zadanej prędkości (silnik kręci się z ustaloną prędkością). Gdy prędkość zadana wynosi 0 – status zostaje zmieniony na 1 .
3	Napęd w trybie zadanej pozycji (dojazd do zadanej pozycji). Gdy silnik osiągnie zadaną pozycję status zmienia się na 4 .
4	Osiągnięcie zadanej pozycji. Zakończenie bazowania także sygnalizowane jest tym statusem.
5	Błąd pozycji – silnik nie może osiągnąć zadanej pozycji – np. podczas mechanicznego zablokowania lub utraty synchronizmu. (Status dostępny tylko podczas pracy z enkoderem.)
6	Bazowanie w toku. Po zakończeniu bazowania status zmienia się na 4 .
7	Przeciążenie napędu – przekroczenie prądu maksymalnego, przeciążenie termiczne.
8	Korekcja pozycji – napęd koryguje pozycję aktualną względem rzeczywistej pozycji odczytanej z enkodera. Po skorygowaniu pozycji status zmienia się na 4 . (Status dostępny tylko podczas pracy z enkoderem.)

5. Dodatkowe funkcje

5.1 Rejestry użytkownika

SIC184 posiada 50 rejestrów dowolnego przeznaczenia (**adresy 50-98**). Użytkownik może przechowywać w tych rejestrach dowolne wartości, np. parametry ruchu silnika (prędkości, przyśpieszenia, zadane pozycje). Zapisane przez użytkownika rejestry mogą być odczytywane przez program wykonywany w sterowniku lub przez komendy modbus REG_SPEED oraz REG_POS. Rejestry użytkownika mogą być dodatkowo zapamiętane w pamięci nieulotnej sterownika – wówczas przesłane do nich wartości mogą być dostępne po ponownym włączeniu urządzenia.

5.1.1 Odczyt rejestrów z poziomu programu wykonywanego przez sterownik

Programista ma możliwość odwoływania się do rejestrów użytkownika z poziomu języka ML-PROG. Podczas tworzenia komend ruchu w programie ML-PROG używając komend takich jak: *SetVmin*, *SetVmax*, *SetAmax*, *SetVdest*, *SetXdest*, *SetXmove*, *WaitTime*, *GoHome* jako parametr można podać adres rejestru np.:

SetVdest(0, \$50) - zadanie prędkości przechowywanej w rejestrze 50 (pierwszy rejestr użytkownika)
SetXmove(0, \$98) - zadanie prędkości przechowywanej w rejestrze 98 (ostatni rejestr użytkownika)

Rejestry można podawać tylko jako wartości **parzyste** (50,52,54...98), gdyż parametry ruchów przechowywane są zawsze w **dwóch sąsiednich rejestrach**. Np. podając adres rejestru 50, program odwoła się do rejestrów 50-51 przechowujących liczbę 4-bajtową.

Dane zapisywane w rejestrach 50...98 muszą zawierać liczby typu **REAL** (DWORD dla komendy *WaitTime*) i zajmować zawsze 2 kolejne rejestry, np. rejestry 50-51 → paramer A, 52-53 → paramer B itd...

5.1.2 Odczyt rejestrów przez komendy pozycji i prędkości MODBUS

Za pomocą rejestrów *REG_SPEED* lub *REG_POS* użytkownik może zadać wartość prędkości lub pozycji przechowywanej w rejestrach użytkownika. Jako wartość przesyłaną do rejestru *REG_SPEED* lub *REG_POS* należy podać:

- adres rejestru użytkownika (50..98), wówczas zadana zostanie prędkość lub pozycja w impulsach, a rejestr użytkownika do którego odnosi się komenda powinien przechowywać wartość typu **DINT**,
- adres rejestru użytkownika (50..98) + 0xFF00, wówczas zadana zostanie prędkość lub pozycja w

obrotach, a rejestr do którego odnosi się komenda powinien przechowywać wartość typu **REAL**,

5.1.3 Zapis rejestrów użytkownika do pamięci nieulotnej

Zapamiętanie wartości rejestrów użytkownika następuje przez wysłanie wartości 0xFF00 do rejestru **SAVE_USER_REG (5100)**. Przykład zapisu pojedynczego rejestru znajduje się w rozdziale 3.3.

6. Historia zmian

v1.08 (dla wersji **firmware SIC184 1.8** lub wyższej):

- dodany rejestr **46 - M_VDEST** (zadawaniem prędkości w obr/min)
- dodany rejestr **47 - M_VACT** (aktualna prędkość w obr/min)
- usunięte rejestry **20-21 - REAL_TEST_REG**

v1.07 (dla wersji **firmware SIC184 1.7** lub wyższej):

- rejestr statusu pozycji **25 MOTOR_IS_GET_POS** zastąpiony przez rejestr **MOTOR_STATUS**, który zawiera **więcej informacji o stanie napędu i trybu pracy**.

v1.03 (dla wersji **firmware SIC184 1.6** lub wyższej):

- usunięte rejestry **20-21 TEST_REG**
- dodane rejestry **20 -REG_SPEED** (zadawanie prędkości przechowywanej w rejestrach użytkownika)
21 -REG_POS (zadawanie pozycji przechowywanej w rejestrach użytkownika)
- dodane rejestry **50-100 - USER_REGISTER** (rejestry użytkownika)
- dodany rejestr **5100 -SAVE_USER_REG** (zapis rejestrów użytkownika do pamięci nieulotnej)