

# MIC488 Protokół MODBUS-RTU (v1.70)

## Spis treści

<b>1. Informacje wstępne .....</b>	<b>2</b>
<b>2. Parametry transmisji i funkcje MODBUS.....</b>	<b>2</b>
<b>3. Spis rejestrów .....</b>	<b>3</b>
3.1 Odczyt i zapis wejść, wyjść i rejestrów bitowych.....	3
3.2 Odczyt i zapis rejestrów liczbowych .....	4
<b>4. Opis sterowania przez Modbus.....</b>	<b>7</b>
4.1 Sterowanie programem w pamięci sterownika.....	7
4.2 Załączenie, wyłączenie i zatrzymanie napędu .....	7
4.3 Ustawienie parametrów ruchu (rampy) .....	8
4.4 Bazowanie napędu.....	8
4.5 Zadanie prędkości i pozycji .....	8
4.6 Zerowanie pozycji, odczyt pozycji i prędkości .....	9
4.7 Status sterownika.....	9
4.8 Status napędu .....	9
4.9 Tryb sterowania JOG .....	10
4.10 Odczyt / zadawanie parametrów w impulsach (dane typu DINT) .....	10
4.11 Rejestry użytkownika.....	11
<b>5. Program MIC488 ModbusTester .....</b>	<b>11</b>
<b>6. Historia zmian.....</b>	<b>12</b>

## 1. Informacje wstępne

Sterownik MIC488 został wyposażony w interfejsy szeregowo RS232 i 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.



### UWAGA!

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



### UWAGA!

Porty COM1 (RS232) i COM2 (RS485) sterownika MIC488 nie są optoizolowane. 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 lub uszkodzenie urządzenia.

## 2. Parametry transmisji i funkcje MODBUS

### Parametry transmisji

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

### Zaimplementowane funkcje MODBUS

Nr funkcji (hex)	Opis
0x01	Odczyt stanu wyjścia
0x02	Odczyt stanu wejścia
0x03	Odczyt N rejestrów (dla liczb WORD, INT, DIN, REAL)
0x05	Zapis wejścia
0x06	Zapis 1 rejestru (dla liczb WORD, INT)
0x10	Zapis N rejestrów (dla liczb DIN, REAL)

### Opis typów zmiennych wykorzystywanych przez MODBUS-RTU w MIC488

Nazwa zmiennej	Opis	Rozmiar bajty / rejestry	Zakres
BYTE	Liczba 1 bajtowa	1 / 1	0-255
WORD	Liczba 2 bajtowa	2 / 1	0...32768
INT	Liczba 2 bajtowa	2 / 1	-32768...32767
DINT	Liczba 4 bajtowa ze znakiem	4 / 2	$-2^{31} \dots (2^{31}-1)$
REAL	Liczba zmiennoprzecinkowa	4 / 2	$1.18 \cdot 10^{-38} \dots 3.40 \cdot 10^{38}, 0, -3.40 \cdot 10^{38} \dots -1.18 \cdot 10^{-38}$



Wartości wprowadzone do rejestrów nie są zapamiętywane i przyjmują wartości domyślne po ponownym włączeniu zasilania sterownika.

### 3. Spis rejestrów

#### 3.1 Odczyt i zapis wejść, wyjść i rejestrów bitowych

Adres	Nazwa	Typ zmiennej	Tryb (numer funkcji MODBUS)	Opis
6000 6000...6007	OUT	BIT	R (0x01), W (0x05)	Odczyt wyjść OUT1...OUT8 Ustawienie wyjścia OUT1...OUT8
5000...5019	IN	BIT	R (0x02)	Odczyt wejść IN1...IN20
2000	M1_EN	BIT	W (0x05)	Ustawienia wyjścia M1 EN
2001	M2_EN	BIT	W (0x05)	Ustawienia wyjścia M2 EN
2002	M3_EN	BIT	W (0x05)	Ustawienia wyjścia M3 EN
2003	M4_EN	BIT	W (0x05)	Ustawienia wyjścia M4 EN
2004	M1_STOP	BIT	W (0x05)	Zatrzymanie napędu M1 (STOP)
2005	M2_STOP	BIT	W (0x05)	Zatrzymanie napędu M2 (STOP)
2006	M3_STOP	BIT	W (0x05)	Zatrzymanie napędu M3 (STOP)
2007	M4_STOP	BIT	W (0x05)	Zatrzymanie napędu M4 (STOP)
2008	M1_JOG_PLUS	BIT	W (0x05)	Praca w trybie JOG+ napędu M1
2009	M1_JOG_MINUS	BIT	W (0x05)	Praca w trybie JOG- napędu M1
2010	M2_JOG_PLUS	BIT	W (0x05)	Praca w trybie JOG+ napędu M2
2011	M2_JOG_MINUS	BIT	W (0x05)	Praca w trybie JOG- napędu M2
2012	M3_JOG_PLUS	BIT	W (0x05)	Praca w trybie JOG+ napędu M3
2013	M3_JOG_MINUS	BIT	W (0x05)	Praca w trybie JOG- napędu M3
2014	M4_JOG_PLUS	BIT	W (0x05)	Praca w trybie JOG+ napędu M4
2015	M4_JOG_MINUS	BIT	W (0x05)	Praca w trybie JOG- napędu M4
2100	USER_REG_READ	BIT	W (0x05)	Odczytanie rejestrów użytkownika (1000..2000) z pamięci nieulotnej
2101	USER_REG_SAVE	BIT	W (0x05)	Zapisanie rejestrów użytkownika (1000..2000) do pamięci nieulotnej
2102	USER_REG_ZERO	BIT	W (0x05)	Reset rejestrów użytkownika (1000..2000) (ustawienie wartości 0)

Info:

- W urządzeniach typu HMI funkcje odczytu/zapisu wejść/wyjść bitowych oznaczane są najczęściej jako **x0**.
- Ustawienie bitu odbywa się przez wysłanie wartości 0xFF00

Przykładowa ramka danych dla odczytu stanu wejścia IN1 (adres = 5000)

Zapytanie (MASTER -> MIC488)		Odpowiedź (MIC488 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x02	Funkcja	0x02
Początkowy adres HI	<b>0x13</b>	Ilość bajtów	0x01
Początkowy adres LO	<b>0x88</b>	Status wejść	BYTE
Ilość wejść HI	0x00	CRC HI	BYTE
Ilość wejść LO	<b>0x08</b>	CRC LO	BYTE
CRC HI	FD		
CRC LO	62		

Przykładowa ramka danych dla ustawienia wyjścia OUT2 (adres = 6001)

Zapytanie (MASTER -> MIC488)		Odpowiedź (MIC488 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
Funkcja	0x05	Funkcja	0x05
Początkowy adres HI	<b>0x17</b>	Początkowy adres HI	<b>0x17</b>
Początkowy adres LO	<b>0x71</b>	Początkowy adres LO	<b>0x71</b>
Ilość wejść HI	0x00	Ilość wejść HI	0x00
Ilość wejść LO	<b>0x00</b>	Ilość wejść LO	<b>0x00</b>
CRC HI	98	CRC HI	98
CRC LO	65	CRC LO	65

## 3.2 Odczyt i zapis rejestrów liczbowych

### Rejestry (pamięć) użytkownika do dowolnego wykorzystania

Adres	Nazwa	Typ zmiennej	Tryb (numer funkcji MODBUS)	Opis
0...1999	USER_REGISTER	WORD DIN REAL	R (0x03), W (0x06) R (0x03), W (0x10) R (0x03), W (0x10)	Rejestry użytkownika. Rejestry od 1000..1999 mogą zostać zapisane w nieulotnej pamięci sterownika za pomocą rejestru 2101 (USER_REG_SAVE) oraz z niej odczytane za pomocą rejestru 2100 (USER_REG_READ)

### Rejestry sterujące typu WORD

Adres	Nazwa	Typ zmiennej	Tryb (numer funkcji MODBUS)	Opis
10000	DATA_TYPE	WORD	R (0x03), W (0x06)	Typ danych wysyłanych przez MODBUS: 0 – REAL, 1 - DINT
10001	PROG_BANK_SEL	WORD	R (0x03), W (0x06)	Wybór numeru banku programu do uruchomienia
10002	POS_BANK_SEL	WORD	R (0x03), W (0x06)	Wybór numeru banku pozycji do uruchomienia
10003	PROGRAM_CTRL	WORD	R (0x03), W (0x06)	Sterowanie programem: 0 – STOP, 1 – START, 2 - PAUZA
10004	AXIS_CTR_MODE	WORD	R (0x03), W (0x06)	Tryb zadawania prędkości/pozycji dla napędów: 0 (DIRECT) - prędkość/pozycja wykonana zaraz po wpisaniu do danego rejestru 1 (TRIGGER) - ruch wykonywany jest po zapisaniu do rejestru AXIS_POS_TRIG (wyzwolenie pozycji) lub AXIS_VEL_TRIG (wyzwolenie prędkości)
10005	AXIS_VEL_ABS_TRIG	WORD	R (0x03), W (0x06)	Dla trybu AXIS_CTR_MODE = 1 Wyzwolenie ruchu z rejestrów prędkości absolutnej. Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10006	AXIS_VEL_REL_TRIG	WORD	R (0x03), W (0x06)	Dla trybu AXIS_CTR_MODE = 1 Wyzwolenie ruchu z rejestrów prędkości względnej Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10007	AXIS_POS_ABS_TRIG	WORD	R (0x03), W (0x06)	Dla trybu AXIS_CTR_MODE = 1 Wyzwolenie ruchu z rejestrów pozycji absolutnej Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10008	AXIS_POS_REL_TRIG	WORD	R (0x03), W (0x06)	Dla trybu AXIS_CTR_MODE = 1 Wyzwolenie ruchu z rejestrów pozycji względnej Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10009	M_ALL_STATUS	WORD	R (0x03)	Niedostępne
10010	M1_STATUS	WORD	R (0x03)	Status napędu:
10011	M2_STATUS			0 - napęd wyłączony (sygnał EN nieaktywny)
10012	M3_STATUS			1 - napęd włączony, brak ruchu (sygnał EN aktywny)
10013	M4_STATUS			2 - napęd w trybie zadanej prędkości
				3 - napęd w trybie ruchu do zadanej pozycji
				4 - napęd dojechał do zadanej pozycji
				5 - błąd dojazdu do zadanej pozycji (enkoderem)
				6 - napęd w trybie bazowania
				8 - napęd w trybie korekcji pozycji (enkoderem)
10014	M_ENABLE	WORD	W (0x06)	Włączenie wyjścia EN napędu Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10015	M_DISABLE	WORD	W (0x06)	Wyłączenie wyjścia EN napędu Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4
10016	M_STOP	WORD	W (0x06)	Zatrzymanie napędu Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4

#### Info:

- W urządzeniach typu HMI funkcje odczytu/zapisu wartości liczbowych oznaczane są najczęściej jako **x4**.
- Dla rejestrów sterujących jednocześnie 4 napędami (AXIS\_VEL\_ABS\_TRIG, AXIS\_VEL\_REL\_TRIG, AXIS\_POS\_ABS\_TRIG, AXIS\_POS\_REL\_TRIG, M\_ENABLE, M\_DISABLE, M\_STOP) najmłodsze 4 bity rejestru określają numer napędu który ma byćysterowany: Bit 0 – M1, Bit 1 – M2, Bit 2 – M3, Bit 4 – M4. Przykładowo do jednoczesnego zatrzymania napędów M1, M2, M3, M4 należy wysłać do rejestru M\_STOP wartość 15 (ustawione bity 0,1,3,4).

## Rejestry sterujące typu DINT, REAL

Adres	Nazwa	Typ zmiennej	Tryb (numer funkcji MODBUS)	Opis
Wartości aktualne oraz parametry rampy				
10017	JOG_SPEED	REAL	W (0x06)	Prędkość dla trybu ręcznego ruchu napędami
10020 10022 10024 10026	M1_POS_ACT M2_POS_ACT M3_POS_ACT M4_POS_ACT	REAL / DINT*	R (0x03), W (0x10)	Odczyt pozycji aktualnej napędu Zapis pozycji aktualnej do napędu. <b>Wpisanie wartości 0 powoduje wyzerowanie aktualnej pozycji.</b>
10028 10030 10032 10034	M1_VEL_ACT M2_VEL_ACT M3_VEL_ACT M4_VEL_ACT	REAL / DINT*	R (0x03)	Odczyt aktualnej prędkości.
10036 10038 10040 10042	M1_ACC M2_ACC M3_ACC M4_ACC	REAL / DINT*	R (0x03), W (0x10)	Odczyt / zapis przyspieszenia w trybie pozycji oraz przyspieszenia i hamowania w trybie prędkości.
10044 10046 10048 10050	M1_DEC M2_DEC M3_DEC M4_DEC	REAL / DINT*	R (0x03), W (0x10)	Odczyt / zapis hamowania w trybie prędkości.
10052 10054 10056 10058	M1_VMAX M2_VMAX M3_VMAX M4_VMAX	REAL / DINT*	R (0x03), W (0x10)	Odczyt / zapis prędkości maksymalnej dla trybu pozycji.
Zadawanie ruchu.				
UWAGA: Gdy rejestr AXIS_CTR_MODE (10004) = 0 (tryb DIRECT) wykonanie ruchu następuje odrazu po zapisaniu do danego rejestru. Gdy rejestr AXIS_CTR_MODE (10004) = 1 (tryb TRIGGER) wykonanie ruchu następuje dopiero po zapisaniu do rejestru wyzwalającego (odpowiednio rejestry 1005..1008)				
10060 10062 10064 10066	M1_HOME M2_HOME M3_HOME M4_HOME	REAL / DINT*	W (0x10)	Wykonanie bazowania. Wartość rejestru określa prędkość bazowania.
10068 10070 10072 10074	M1_VEL_ABS M2_VEL_ABS M3_VEL_ABS M4_VEL_ABS	REAL / DINT*	W (0x10)	Zadanie prędkości absolutnej (prędkość napędu będzie równa wprowadzonej wartości).
10076 10078 10080 10082	M1_VEL_REL M2_VEL_REL M3_VEL_REL M4_VEL_REL	REAL / DINT*	W (0x10)	Zadanie prędkości względnej (prędkość napędu będzie równa prędkości aktualnej + wprowadzonej wartości).
10084 10086 10088 10090	M1_POS_ABS M2_POS_ABS M3_POS_ABS M4_POS_ABS	REAL / DINT*	W (0x10)	Zadanie pozycji absolutnej (nastąpi ruch napędu do osiągnięcia zadanej pozycji).
10092 10094 10096 10098	M1_POS_REL M2_POS_REL M3_POS_REL M4_POS_REL	REAL / DINT*	W (0x10)	Zadanie pozycji względnej (nastąpi ruch napędu do pozycji równej aktualnej + wprowadzonej wartości).
Rejestry dla enkoderów				
10100 10102 10104 10106	ENC1_IMP ENC2_IMP ENC3_IMP ENC4_IMP	DINT	R (0x03), W (0x10)	Wartość licznika enkodera w impulsach.
10108 10110 10112 10114	ENC1_XPOS ENC2_XPOS ENC3_XPOS ENC4_XPOS	REAL / DINT*	R (0x03), W (0x10)	Wartość licznika enkodera w impulsach przeliczonych na obroty napędu.
10120 10122 10124 10126	M1_POSLIM_L M2_POSLIM_L M3_POSLIM_L M4_POSLIM_L	REAL / DINT*	R (0x03), W (0x10)	Programowe ograniczenie pozycji w stronę ujemną (L)
10128 10130	M1_POSLIM_R M2_POSLIM_R	REAL / DINT*	R (0x03), W (0x10)	Programowe ograniczenie pozycji w stronę dodatnią (R)

10132	M3_POSLIM_R		
10134	M4_POSLIM_R		



MIC488 adresuje rejestry od 0. Dla urządzeń MASTER, w których adresowanie rozpoczyna się od 1 należy podawać wartości rejestrów z przesunięciem o 1 np.: JOG\_SPEED = 10017 + 1 = 10018

\* Typ zmiennej zależy od ustawionego typu danych w rejestrze DATA\_TYPE (10000). Domyślnie są to wartości typu REAL, przekalowane na odpowiednie jednostki ruchu.

Po wpisaniu do rejestru **DATA\_TYPE** wartości 1 sterownik przyjmuje i odsyła wartości w impulsach (liczby typu DINT), które nie są przekalowane. Wartości w impulsach są równoważne ilościom kroków wygenerowanych przez sterownik.

Przykładowa ramka zadania prędkości absolutnej dla napędu M1. Funkcja: 0x10, Adres rejestru: 10068 (M1\_VEL\_ABS)

Zapis (MASTER -> MIC488)		Odpowiedź (MIC488 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
<b>Funkcja</b>	<b>0x10</b>	Funkcja	0x10
<b>Adres rejestru Hi</b>	<b>0x27</b>	Początkowy adres Hi	0x27
<b>Adres rejestru Lo</b>	<b>0x54</b>	Początkowy adres Lo	<b>0x54</b>
Ilość rejestrów Hi	0x00	Ilość rejestrów Hi	0x00
Ilość rejestrów Lo	0x02	Ilość rejestrów Lo	<b>0x02</b>
Ilość bajtów	0x04	CRC	16 bit
Rejestr 0x06 Hi	REAL/DINT* (Bajt 1)		
Rejestr 0x06 Lo	REAL/DINT* (Bajt 0)		
Rejestr 0x06 +1 Hi	REAL/DINT* (Bajt 3)		
Rejestr 0x06 +1 Lo	REAL/DINT* (Bajt 2)		
CRC	<b>16 bit</b>		

Przykładowa ramka odczytu pozycji aktualnej dla napędu M1. Funkcja: 0x03, Adres rejestru: 1020 (M1\_POS\_ACT)

Zapytanie (MASTER -> MIC488)		Odpowiedź (MIC488 -> MASTER)	
Adres urządzenia	0x01	Adres urządzenia	0x01
<b>Funkcja</b>	<b>0x03</b>	Funkcja	0x03
<b>Adres rejestru Hi</b>	<b>0x27</b>	Ilość bajtów	<b>0x04</b>
<b>Adres rejestru Lo</b>	<b>0x24</b>	Rejestr 0x03 Hi	REAL/DINT* (Bajt 1)
Ilość rejestrów Hi	0x00	Rejestr 0xFC Lo	REAL/DINT* (Bajt 0)
Ilość rejestrów Lo	0x02	Rejestr 0x03+1 Hi	REAL/DINT* (Bajt 3)
CRC HI	0x04	Rejestr 0xFC+1 Lo	REAL/DINT* (Bajt 2)
CRC LO	0x7F	CRC	16 bit



Wszystkie liczby 4 bajtowe typu **DINT**, **DWORD**, **REAL** zawierają się zawsze w **dwóch rejestrach**. Ponadto dla liczb DINT pierwszy rejestr zawiera jej młodszą część, drugi starszą. Przykładowo by odczytać pozycję aktualną napędu M1 należy odczytać rejestry o numerach 1020 oraz 1021, a następnie wykonać odpowiednią konwersję (jeśli nie ma odpowiedniej funkcji modbus w sterowniku MASTER).

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**

## 4. Opis sterowania przez Modbus

### 4.1 Sterowanie programem w pamięci sterownika

Za pomocą rejestru **PROG\_BANK\_SEL** (10001) możliwy jest wybór programu z pamięci sterownika który ma być kontrolowany (uruchomiony, zatrzymany, wstrzymany). Do rejestru należy wpisać wartość odpowiadającą wybranemu programowi (wartość 0...5).

Za pomocą rejestru **POS\_BANK\_SEL** (10002) można wybrać bank z pozycjami, które będą wykorzystywane przez program (o ile w program używa pozycji z tablicy pozycji).

Rejestr **PROGRAM\_CTRL** (10003) steruje pracą programu:

- Zapisanie wartości 1 powoduje uruchomienie programu
- Zapisanie wartości 2 powoduje wstrzymanie wykonywanego programu
- Zapisanie wartości 0 powoduje zatrzymanie programu



Jeśli uruchomiony zostanie program z pustego banku programu lub pozycji, sterownik zasygnalizuje błąd zapalając czerwoną diodę ERR.

### 4.2 Załączenie, wyłączenie i zatrzymanie napędu

#### Zapisywanie do rejestrów kontrolujących wybrane napędy jednocześnie

Rejestry takie jak:

- **AXIS\_VEL\_ABS\_TRIG** (10005)
- **AXIS\_VEL\_REL\_TRIG** (10006),
- **AXIS\_POS\_ABS\_TRIG** (10007),
- **AXIS\_POS\_REL\_TRIG** (10008),
- **M\_ENABLE** (10014)
- **M\_DISABLE** (10015)
- **M\_STOP** (10016)

pozwalają na jednoczesne sterowanie wybranymi napędami (np. włączenie wybranych napędów, zatrzymanie itp., wyzwolenie ruchu dla trybu TRIGGER) . Do wybranego rejestru należy zapisać wartość odpowiadającą bitowemu ustawieniu wybranych napędów, gdzie najmłodszy bit 0 – napęd M1, bit 1 – M2, bit 2 – M3, bit 3 – M4. Poniższa tabela zawiera zestawienie wszystkich kombinacji bitowych. Znak „+” oznacza, że dany napęd jest wybrany.

wybrany napęd				wartość wpisana do rejestru
M1	M2	M3	M4	
-	-	-	-	0
+	-	-	-	1
-	+	-	-	2
+	+	-	-	3
-	-	+	-	4
+	-	+	-	5
-	+	+	-	6
+	+	+	-	7
-	-	-	+	8
+	-	-	+	9
-	+	-	+	10
+	+	-	+	11
-	-	+	+	12
+	-	+	+	13
-	+	+	+	14
+	+	+	+	15

## Sterowanie wyjściem ENABLE

- Niezależne sterowanie każdym z wyjść przy wykorzystaniu rejestrów bitowych **M1\_EN...M4\_EN** (2000...2003). Jeden rejestr steruje tylko jednym wyjściem EN. Zapisanie do rejestru wartości 0xFF00 włącza wyjście EN, zapisanie wartości 0x00 wyłącza wyjście.
- Jednoczesne sterowanie wybranymi wyjściami za pomocą rejestru **M\_ENABLE** (1014) włączającego wybrane wyjścia EN oraz **M\_DISABLE** (1015) wyłączającego.

## Zatrzymanie napędu

- Niezależne zatrzymanie każdego napędu przy wykorzystaniu rejestrów bitowych **M1\_STOP...M4\_STOP** (2004...2007). Zapisanie do rejestru wartości 0xFF00 zatrzymuje napęd.
- Jednoczesne zatrzymanie wybranych napędów za pomocą rejestru **M\_STOP** (1016). Przykładowo zapisując wartość 15 zatrzymujemy wszystkie napędy.

## 4.3 Ustawienie parametrów ruchu (rampy)

Ustawienie parametrów ruchu kontrolujących rampę odbywa się za pomocą rejestrów:

- **M1\_ACC...M4\_ACC**(10036...10042) – przyspieszenie rozpędzania dla trybu pozycji i prędkości
- **M1\_DEC...M4\_DEC** (10044...10050) – przyspieszenie zwalniania dla trybu pozycji
- **M1\_VMAX...M4\_VMAX** (10052...10058) – prędkość maksymalna dla zadanej pozycji



Parametry ruchu powinny być wprowadzane jako wartości dodatnie, większe od zera!

## 4.4 Bazowanie napędu

Bazowanie napędów odbywa się za pomocą rejestrów **M1\_HOME...M4\_HOME** (10060...10066). Wpisana do rejestru wartość określa prędkość bazowania. Bazowanie następuje od razu po zapisaniu do rejestru. Po zakończeniu bazowania pozycja napędu jest automatycznie zerowana.



Bazowanie w stronę **krańcówki lewej** (KL) odbywa się przez zapisanie prędkości **ze znakiem ujemnym**.

## 4.5 Zadanie prędkości i pozycji

### Prędkość/pozycja absolutna i względna (relatywna)

Prędkość i pozycja może być zadana **absolutnie** (przykładowo napęd osiągnie prędkość/pozycję równą wpisanej do rejestru), lub **względnie** (napęd zwiększy lub zmniejszy swoją prędkość/pozycję o wartość wpisaną do rejestru). Rejestry posiadające w nazwie **ABS** zadają wartości absolutnie, a **REL** – względnie (relatywnie).

### Rejestry ruchu

- **M1\_VEL\_ABS... M4\_VEL\_ABS** (10068...10074) – zadanie prędkości absolutnie
- **M1\_VEL\_REL ... M1\_VEL\_REL** (10076...10082) – zadanie prędkości względnie
- **M1\_POS\_ABS ... M4\_POS\_ABS** (10084...10090) – zadanie pozycji absolutnie
- **M1\_POS\_REL ... M1\_POS\_REL** (10092...10098) – zadanie pozycji względnie

### Tryby zadawania prędkości / pozycji

Zmiana trybu następuje przez zapisanie do rejestru **AXIS\_CTR\_MODE** (10004) wartości 1 (tryb TRIGGER) lub 0 (tryb DIRECT).



**Tryb bezpośredni (DIRECT)**, (gdy rejestr `AXIS_CTR_MODE = 0`)

Domyślnie sterownik pracuje w trybie bezpośredniego zadawania prędkości/pozycji (tryb **DIRECT**). Sterownik realizuje ruch od razu po zapisaniu do odpowiedniego rejestru ruchu.

**Tryb wyzwalania (TRIGGER)**, (gdy rejestr `AXIS_CTR_MODE = 1`)

W tym trybie zapisanie do rejestrów ruchu nie powoduje wykonania ruchu przez sterownik. Wykonanie ruchu następuje dopiero po zapisaniu wartości odpowiadającej wybranym numerom napędów do rejestrów wyzwalających:

- `AXIS_VEL_ABS_TRIG` (10005) – dla wyzwolenia prędkości absolutnych
- `AXIS_VEL_REL_TRIG` (10006) – dla wyzwolenia prędkości relatywnych
- `AXIS_POS_ABS_TRIG` (10007) – dla wyzwolenia pozycji absolutnych
- `AXIS_POS_REL_TRIG` (10008) – dla wyzwolenia pozycji relatywnych

Jest to przydatne, gdy należy jednocześnie uruchomić kilka napędów w celu uzyskania synchronicznych ruchów.

Przykładowo dla napędów M1, M2 i M3 należy zadać pozycję absolutną odpowiednio 10, 15 i 20:

- 1) Ustawiamy tryb TRIGGER przez zapisanie do rejestru `AXIS_CTR_MODE` (10004) wartości 1 (by wrócić do trybu DIRECT należy zapisać 0).
- 2) Zapisujemy do rejestrów `M1_POS_ABS`, `M2_POS_ABS` i `M3_POS_ABS` odpowiednio wartości 10, 15 i 20
- 3) Zapisujemy do rejestru wyzwalającego `AXIS_POS_ABS_TRIG` (10007) wartość 7 (odpowiadającą binarnej wartości 0111, bo wyzwalamy ruch dla napędów M1, M2 i M3)

UWAGA: Sygnały EN muszą być wcześniej załączone jeśli sterują pracą napędów.

## 4.6 Zerowanie pozycji, odczyt pozycji i prędkości

### Odczyt aktualnej prędkości i pozycji

Aktualna prędkość dostępna jest w rejestrach `M1_VEL_ACT`... `M4_VEL_ACT`.

Aktualna pozycja dostępna jest w rejestrach `M1_POS_ACT`... `M4_POS_ACT`.

### Zerowanie aktualnej pozycji

W celu wyzerowania aktualnej pozycji dla wybranego napędu należy zapisać 0 do odpowiedniego rejestru `M1_POS_ACT`... `M4_POS_ACT`. Zapisanie wartości różnej od 0 powoduje nadpisanie aktualnej pozycji przesłaną wartością.

## 4.7 Status sterownika

Rejestr `M_STATUS` (10009) informuje o globalnym stanie pracy i błędach sterownika.

Wartość rejestru <code>M_STATUS</code>	Opis
0	Status OK
1	Błąd pozycji napędu (podczas pracy z enkoderem)
2	Błąd programu (niedozwolona instrukcja, dzielenie przez 0 itp.)
3	Błąd komunikacji Modbus Master (urządzenie slave nie odpowiedziało)

## 4.8 Status napędu

Za pomocą rejestrów statusowych `M1_STATUS` (10010)...`M4_STATUS` (10013) możliwa jest kontrola działania napędu. W zależności od wartości rejestru

Wartość rejestru <code>MX_STATUS</code>	Opis
0	Napęd wyłączony (sygnał EN zdjęty)

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)

Status napędu może być wykorzystany m.in. do określenia czy napęd osiągnął zadaną pozycję przed zadaniem kolejnej.

## 4.9 Tryb sterowania JOG

Tryb JOG może być wykorzystany do ręcznej zmiany pozycji napędów poprzez np. panel HMI. Przez ustawienie odpowiedniego rejestru JOG następuje ruch w danym napędzie. Prędkość ruchu w trybie JOG określana jest w rejestrze **JOG\_SPEED** (1017) (liczba typu REAL).

Ruch dla poszczególnych napędów wyzwalany jest poprzez rejestry bitowe:

- **M1\_JOG\_PLUS** (2008) - ruch w stronę dodatnią napędu M1
- **M1\_JOG\_MINUS** (2009) - ruch w stronę ujemną napędu M1
- **M2\_JOG\_PLUS** (2010) - ruch w stronę dodatnią napędu M2
- **M2\_JOG\_MINUS** (2011) - ruch w stronę ujemną napędu M2
- **M3\_JOG\_PLUS** (2012) - ruch w stronę dodatnią napędu M3
- **M3\_JOG\_MINUS** (2013) - ruch w stronę ujemną napędu M3
- **M4\_JOG\_PLUS** (2014) - ruch w stronę dodatnią napędu M4
- **M4\_JOG\_MINUS** (2015) - ruch w stronę ujemną napędu M4

## 4.10 Odczyt / zadawanie parametrów w impulsach (dane typu DINT)

Sterownik pozwala sterować napędami bez przeliczeń jednostek (konfigurowanych w sterowniku) za pomocą wartości typu **DINT**. Jest to przydatne, gdy wymagane jest bezpośrednie zadawanie pozycji w impulsach lub sterownik nadrzędny nie wspiera konwersji rejestrów do liczb zmiennoprzecinkowych typu **REAL**.

By przełączyć typ danych na DINT należy do rejestru **DATA\_TYPE** (10000) zapisać 1. Wówczas wszystkie rejestry, których typ oznaczony jest jako **REAL / DINT\*** będą przyjmowały/odszyły wartości typu DINT.

### Przykład

Napęd: silnik krokowy 200 imp./obrót z podziałem kroku 1/64.

Przeliczenie jednostek:  $200 * 64 = 12800$  impulsów / obrót silnika.

### Zadanie prędkości 2,5 obr./sek dla napędu M1:

Do rejestru **M1\_VEL\_ABS** (10068) należy zapisać wartość  $2,5 * 12800 = 32000$  [imp]

### Odczyt aktualnej pozycji napędu M1:

Odczytujemy rejestr **M1\_POS\_ACT** (10020), który przykładowo odsyła wartość 57600.

Aktualna pozycja napędu w [obr] =  $57600/12800 = 4,5$  [obr]

## 4.11 Rejestry użytkownika

MIC488 posiada 2000 rejestrów dowolnego przeznaczenia (**adresy 0-1999**). Użytkownik może przechowywać w tych rejestrach wartości, które mogą być odczytywane lub zapisywane przez program wykonywany w sterowniku. Do rejestrów mogą być zapisywane wartości typu INT, DINT oraz REAL. W przypadku zapisu wartości typu DINT i REAL, które zawierają się zawsze w dwóch sąsiednich rejestrach należy zapisywać je pod adresy parzyste.

Rejestr	0	1	2	3	...	1000	1001	1996	1997
	INT 0	INT 1	INT 2	INT 3		INT 1000	INT 1001	INT 1996	INT 1997
	DINT 0		DIN 2			DIN 1000		DIN 1996	
	REAL 0		REAL 2			REAL 1000		REAL 1996	
	możliwość zapisu do pamięci nieulotnej								

W celu odwołania się do rejestrów użytkownika w języku WBC należy użyć komend:

\$IX - odwołanie się do rejestru z wartością typu INT

\$DX - odwołanie się do rejestru z wartością typu DINT (X tylko parzyste)

\$RX - odwołanie się do rejestru z wartością typu REAL (X tylko parzyste)

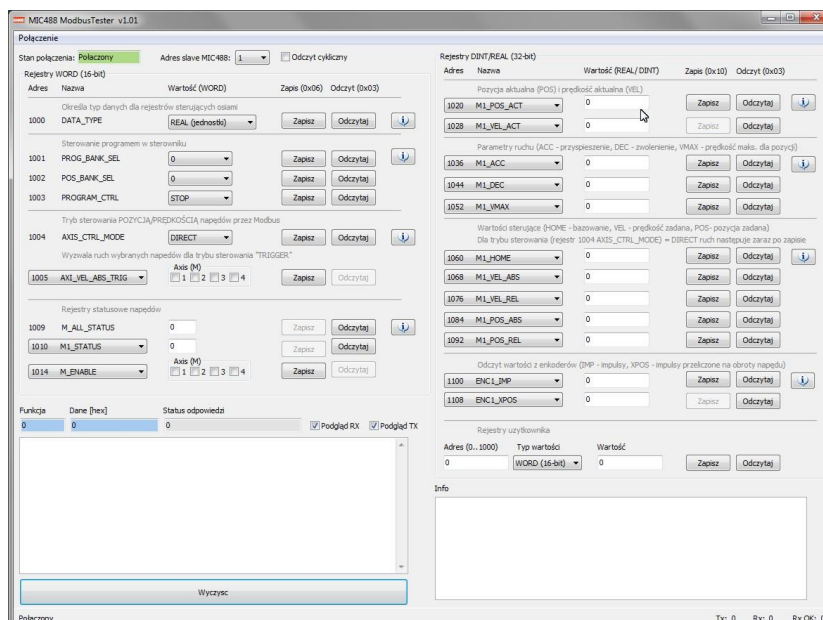
,gdzie X oznacza adres rejestru 0..1996

Rejestry od numeru 1000 do 1999 mogą zostać zapisane w nieulotnej pamięci sterownika. Zapis / odczyt / kasowanie odbywa się poprzez następujące rejestry bitowe:

- USER\_REG\_READ (2100) – odczyt z pamięci
- USER\_REG\_SAVE (2101) – zapis do pamięci
- USER\_REG\_RESET (2102) – ustawienie wartości 0

## 5. Program MIC488 ModbusTester

Aplikacja MIC488 ModbusTester pozwala przetestować poszczególne rejestry sterownika oraz podejrzeć ramki protokołu Modbus. Do komunikacji programu z MIC488 należy wykorzystać konwerter USB<->RS232/RS485 oparty o układ FTDI.



Program może nie zawierać wszystkich rejestrów dostępnych w urządzeniu.

## 6. Historia zmian

### **v1.23:**

- dodane rejestry pozycji granicznych MX\_POSLIM\_L/ MX\_POSLIM\_R

### **v1.40:**

- zmiana wszystkich offsetów adresów na 10000 ( dla oprogramowania <1.40 było 1000)
- zwiększona pamięć rejestrów użytkownika do 2000

### **V1.70:**

- dodamy rejestr M\_STATUS (10009) – globalny status pracy/błędów sterownika
- dodane rejestry USER\_REG\_READ (2100) , USER\_REG\_SAVE (2101) , USER\_REG\_RESET (2102) zapisu rejestrów użytkownika do pamięci nieulotnej