



# ProLinX<sup>®</sup>

## MCM

Brama ProLinX

Modbus Master/Slave

**Autoryzowany Dystrybutor**  
**Prosoft Technology**  
**RAControls Sp. z o.o.**  
Tel: 032-788-77-20  
Email: [biuro@racontrols.pl](mailto:biuro@racontrols.pl)

Czerwiec 26, 2009

## Ważne informacje dotyczące instalacji

Podłączanie zasilania jak również Wejść oraz wyjść musi być wykonane zgodnie z metodami podłączeń dla klasy 1 dywizji 2- artykuł 501-4 (b) National Electrical Code, NFPA 70 przy instalacji na terenie USA., albo jak podano w sekcji 18-1J2 Canadian Electrical Code przy instalacji na terenie Kanady, i zgodnie z przepisami właściwych służb nadzoru. Należy przestrzegać następujących ostrzeżeń:

- A** UWAGA – NIEBEZPIECZEŃSTWO WYBUCHU – ZAMIANA KOMPONENTÓW MOŻE POGORSZYĆ WARUNKI ZGODNOŚĆ Z KLASĄ I, DYWIZJĄ, 2;
- B** UWAGA – NIEBEZPIECZEŃSTWO WYBUCHU – W WARUNKACH NIEBEZPIECZNYCH WYMIANĘ MODUŁU ORAZ PODŁĄCZANIE We/Wy NALEŻY WYKONYWAĆ PRZY WYŁĄCZONYM ZASILANIU
- C** UWAGA - NIEBEZPIECZEŃSTWO WYBUCHU –NIE NALEŻY ODŁĄCZAĆ URZĄDZEŃ CHYBA, ŻE ZASILANIE ZOSTAŁO ODŁĄCZONE LUB OTOCZENIE NIE JEST NIEBEZPIECZNE.
- D** URZĄDZENIE MUSI BYĆ ZASILANE JEDYNNIE PRZEZ ZASILACZE KLASY 2.

## Wszystkie produkty ProLinx®

UWAGA – NIEBEZPIECZEŃSTWO WYBUCHU – NIE NALEŻY ODŁĄCZAĆ URZĄDZEŃ CHYBA, ŻE ZASILANIE ZOSTAŁO ODŁĄCZONE LUB OTOCZENIE NIE JEST NIEBEZPIECZNE.

AVERTISSEMENT – RISQUE D'EXPLOSION – AVANT DE DÉCONNECTER L'EQUIPMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

### Oznaczenia

UL/Cul	ISA 12.12.01 Klasa I, Div 2 Grupy A, B, C, D
Cul	C22.2 No. 213-M1987



CL I Div 2 GPs A, B, C, D

Temp Code T5

II 3 G

Ex nA nL IIC T5 X

0°C <= Ta <= 60°C

II – Sprzęt nie jest przeznaczony do pracy pod ziemią.

3 – Urządzenie kategorii 3 testowane do standardowych warunków pracy.

G – Urządzenie chronione przed gazami wybuchowymi.

## Bramy ProLinx z portami sieci Ethernet

Seria C Bram ProLinx™ z portami Ethernetowymi **NIE** zawiera web serwera HTML. Musi on zostać zamówiony jako opcja. Opcja ta wymaga fabrycznego dołączenia części hardware. Web Server HTML obsługuje obecnie:

- 8 MB na pliki HTML oraz dołączoną grafikę (wcześniej ograniczenie wynosiło 384K)
- Maksymalny rozmiar strony wynoszący 32K (wcześniej 16K)

## Aby dokonać ulepszenia już zakupionego modułu:

Skontaktuj się z najbliższym dystrybutorem urządzeń Prosoft w celu rozpoczęcia procedury zwrotu towaru do zwrotu modułu do firmy Prosoft.

## Aby zamówić bramę ProLinx Plus z opcją -web:

Dodaj **-WEB** do standardowego numeru zamówienia modułu np. **5201-MNET-MCM-WEB**.

## **Państwa Opinia**

Chcielibyśmy, aby mieli państwo poczucie dobrze podjętej decyzji przy korzystaniu z naszych produktów. Jeżeli mają państwo uwagi, komentarze, sugestie dotyczące produktu, dokumentacji lub wsparcia, prosimy o kontakt telefoniczny lub mailowy.

### **ProSoft Technology**

5201 Truxtun Ave., 3rd Floor  
Bakersfield, CA 93309  
+1 (661) 716-5100  
+1 (661) 716-5101 (Fax)  
[www.prosoft-technology.com](http://www.prosoft-technology.com)

Copyright © ProSoft Technology, Inc. 2009. All Rights Reserved.

Instrukcja obsługi urządzenia PDPS  
Czerwiec 08, 2009

ProSoft Technology®, ProLinx®, inRAx®, ProTalk®, i RadioLinx® są zastrzeżonymi znakami towarowymi firmy ProSoft Technology, Inc. Wszystkie inne nazwy własne są lub mogą być znakami handlowymi innych producentów i są wykorzystane by zidentyfikować konkretne urządzenie.

## **Dokumentacja produktów ProSoft Technology®**

W celu oszczędzania papieru, ProSoft Technology nie dołącza drukowanych instrukcji wraz z produktami. Instrukcje obsługi, specyfikacja techniczna, przykładowe programy drabinkowe, i pliki konfiguracyjne są dostępne na załączonej płycie, oraz za darmo na stronie: [www.prosoft-technology.com](http://www.prosoft-technology.com)

Drukowana dokumentacja jest dostępna w sprzedaży. W celu zakupu dokumentacji należy skontaktować się z regionalnym biurem.

Azja, Pacyfik: +603.7724.2080  
Europa, Bliski Wschód, Afryka: +33 (0) 5.3436.87.20  
Ameryka Łacińska: +1.281.298.9109  
Ameryka Północna: +1.661.716.5100

## Spis Treści

Ważne informacje dotyczące instalacji.....	2
Państwa Opinia .....	3
Dokumentacja produktów ProSoft Technology® .....	3
<b>1 Wstęp</b>	<b>5</b>
1.1 Port(y) Modbus Master/Slave.....	5
1.2 Wewnętrzna baza danych modułu .....	6
<b>2 Fizyczny opis portu szeregowego oraz zaimplementowanego protokołu</b>	<b>8</b>
2.1 Opis Portu Modbus Master/Slave.....	8
2.2 Opis portu szeregowego .....	9
<b>3 Konfiguracja protokołu MCM</b>	<b>10</b>
3.1 [Modbus Port x] .....	11
3.2 [Modbus Port 0 Commands].....	17
<b>4 Połączenia komunikacyjne</b>	<b>22</b>
4.1 Podłączenia portów szeregowych: Jednostki wieloportowe .....	22
<b>5 Diody LED</b>	<b>26</b>
5.1 Diody modułu bazowego.....	26
5.2 Diody dla portów 0,1,2,3 .....	27
5.3 Diody w urządzeniach serii 4101 .....	27
<b>6 Błędy Modbus oraz Stany urządzeń.</b>	<b>28</b>
6.1 Oglądanie błędów oraz stanu urządzeń.....	28
6.2 Obszary danych błędów Modbus oraz stanów urządzeń .....	28
6.3 Kody błędów .....	31
<b>7 Referencje</b>	<b>34</b>
7.1 Przenoszenie danych.....	34
7.2 Plik konfiguracyjny 4102-MBS-MBM .....	35
7.3 Konfiguracja rozkazów MODBUS.....	41
7.4 Specyfikacja protokołu MODBUS.....	45
<b>8 Wsparcie, Serwis i Gwarancja</b>	<b>56</b>
8.1 Wsparcie techniczne.....	56
8.2 Warunki Procedury Zwrotu Towaru .....	57
8.3 Ograniczona Gwarancja .....	58

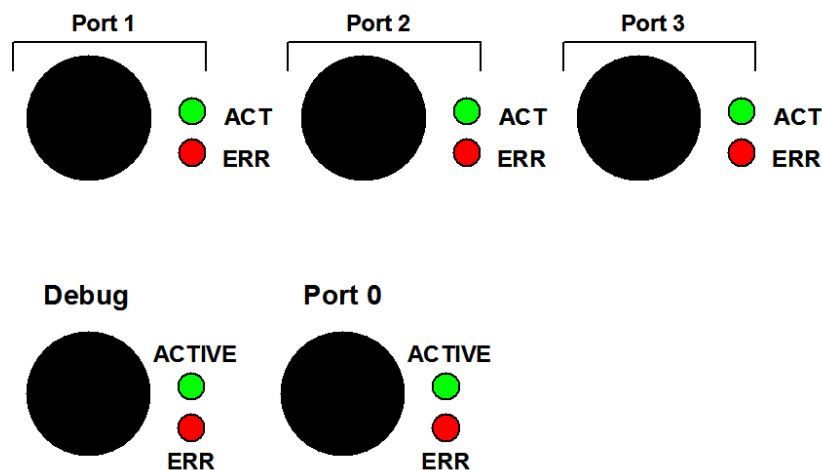
# 1 Wstęp

## Zawartość rozdziału

- ❖ Port(y) Modbus Master/Slave ..... 5
- ❖ Wewnętrzna baza danych modułu ..... 6

Urządzenie Modbus Master/Slave dostępny jest zarówno dla wersji jedno portowej (MCM) jak i wielo portowej (MCM4). W obu przypadkach każdy z portów można skonfigurować do pracy w trybie Master lub Slave.

## 1.1 Port(y) Modbus Master/Slave



Moduł ProLinX obsługuje protokół Modbus w wersji Master oraz Slave dla każdego z czterech portów. Każdy z portów obsługujących protokół Modbus jest indywidualnie konfigurowalny.

Związek pomiędzy opisem portów a ich funkcją przedstawiony jest w poniższej tabeli:

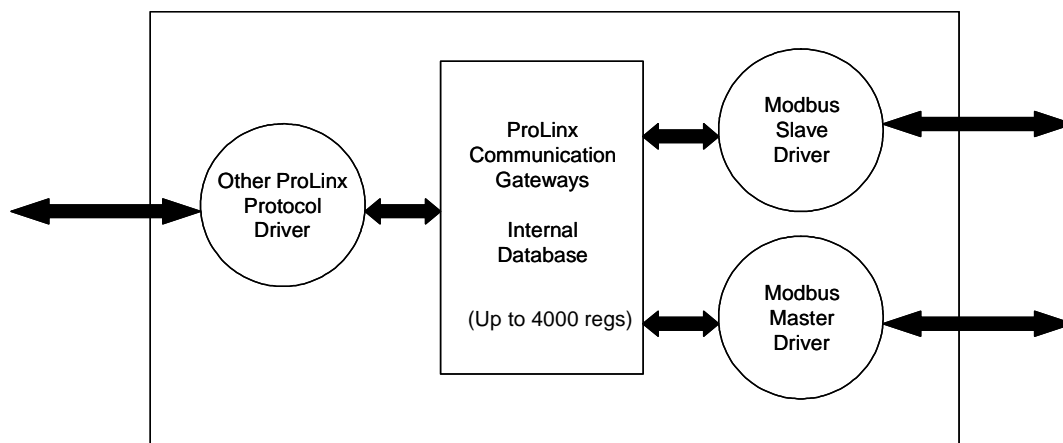
Port	Funkcja
Debug	Diagnostyka/Konfiguracja
Port 0	Modbus Port 0
Poniższe porty istnieją tylko w modułach wieloportowych	
Port 1	Modbus Port 1
Port 2	Modbus Port 2
Port 3	Modbus Port 3

Porty Modbus master mogą być użyte do ciągłej komunikacji z urządzeniami Modbus slave poprzez interfejs RS-232, RS-422 lub RS-485. Każdy port master obsługuje 100 zdefiniowanych przez użytkownika komend określających rozkazy odczytu oraz zapisu wysyłane do poszczególnych urządzeń slave podłączonych do portu.

Dodatkowo moduł może być tak skonfigurowany, aby nadawać niższe priorytety urządzeniom nieodpowiadającym na komendy. Jeśli moduł zauważy brak odpowiedzi na komendę od urządzenia typu slave to po definiowanym przez użytkownika czasie, oznaczy on urządzenie slave, jako "w błędzie komunikacji" oraz ustawi licznik opóźnienia błędu na określoną wartość.

## 1.2 Wewnętrzna baza danych modułu

Istotą funkcjonalności modułu jest wewnętrzna baza danych. Jest to obszar wymiany danych współdzielony przez wszystkie porty modułu. Służy do przepływu informacji z urządzenia w jednej sieci do jednego lub wielu urządzeń w innej sieci. Pozwala to na oglądanie i modyfikację danych z urządzenia na jednym porcie/sieci komunikacyjnej przez urządzenia z innego portu/ sieci.



Dodatkowo oprócz danych z portów master i slave, zmapowane w wewnętrznej bazie danych mogą być także wygenerowane przez moduł informacje o statusie i błędach.

### 1.2.1 Dostęp do bazy danych przez Port Modbus

Sterownik master modułu ProLinx używa bazy danych w dwojaki sposób:

- 1 Komenda odczytu z urządzenia typu slave wysłana przez sterownik typu master zwróci dane do wewnętrznej bazy danych
- 2 Komenda zapisu do urządzenia typu slave wysłana przez sterownik typu master pobiera dane z wewnętrznej bazy danych.

Sterownik slave modułu ProLinx zapewnia dostęp do danych z wewnętrznej bazy danych. Zewnętrzne urządzenia Modbus master mogą odczytywać i zapisywać dane w wewnętrznej bazie danych używając portów modułu skonfigurowanych, jako slave. Do konfiguracji portów slave wystarczy tylko plik konfiguracyjny dostarczony z modułem.

Dodatkowo oprócz danych z portów master/slave w wewnętrznej bazie danych można zmapować informacje o statusie i błędach generowane przez moduł.

## 2 Fizyczny opis portu szeregowego oraz zaimplementowanego protokołu

### Zawartość rozdziału

- ❖ Opis Portu Modbus Master/Slave ..... 8
- ❖ Opis Portu Szeregowego..... 9

### 2.1 Opis Portu Modbus Master/Slave

Parametry	Opis
Podstawowe parametry	
Wewnętrzna baza danych	Do 4000 rejestrów (słów)
Parametry komunikacyjne	Port 0: Prędkość transmisji: 110 do 38.4Kb Port 1,2,3: Prędkość transmisji: 110 do 115Kb Bity Stopu: 1 lub 2 Ilość bitów danych: 5 do 8 bitów Parzystość: None, Even, Odd, Mark, Space Opóźnienie czasowe RTS: 0 to 65535 milisekund
Tryby pracy Modbus	Tryb RTU (binarny) z CRC-16 Tryb ASCII ze kontrolą błędów LRC
Dane zmiennoprzecinkowe	Obsługiwany jest przesył danych zmiennoprzecinkowych, włącznie z konfigurowalnym wsparciem dla implementacji Enron
Kody funkcji Modbus	1: Odczyt stanu wyjścia 2: Odczyt stanu wejścia 3: Odczyt wielu rejestrów danych 4: Odczyt rejestrów wejściowych 5: Zapis pojedynczego bitu 6: Zapis pojedynczego rejestru danych 15: Zapis wielu bitów 16: Zapis wielu rejestrów danych
<b>Modbus Master</b>	
Lista rozkazów	Do 100 rozkazów na każdy z portów typu master, każdy z rozkazów w pełni konfigurowalny pod względem funkcji, adresu urządzenia slave, adresu rejestru odczytywanego/zapisywanego i liczby rejestrów/bitów



Parametry	Opis
Dane Statusu	Kody błędów dla komend. Lista stanów urządzeń typu slave obsługiwana przez aktywny port Modbus Master.
Odpytywanie według listy komend	Sposób odpytywania według listy komend można skonfigurować jako ciągły (continuous) lub wywoływany zmianą danych (on change of data)
<b>Modbus Slave</b>	
Adres węzła	1 do 247 (ustawiany programowo)
Dane statusu	Kody błędów, liczniki i status portów dla każdego skonfigurowanego portu slave dostępne są od rejestru 4400.

## 2.2 Opis portu szeregowego

Parametry	Opis
Porty szeregowo	
Kable portu szeregowego (Łącznik DB-9M)	Dla konfigurowalnego portu szeregowego dołączony jest kabel DIN na DB-9M
Typ portu	Łącznik DB-9M
Izolacja portu szeregowego	2500V RMS izolacji pomiędzy portami dla UL1577 3000V DC min. izolacja pomiędzy portem a uziemieniem i pomiędzy portem a obwodem sygnałowym.
Zabezpieczenie portu szeregowego	Dioda linii TVS interfejsu RS-485/422 portu zabezpiecza w zakresie +/- 27V Linie interfejsu RS-232 portu są zabezpieczone dla +/- 36V przy włączaniu zasilania oraz +/- 40V przy wyłączeniu.

**Uwaga:** W module ProLinx z pojedynczym portem szeregowym, dane portu nie są buforowane. Pakiety są przesyłane bezpośrednio do i z chipu szeregowego procesora. Może to powodować błędy podczas komunikacji szeregowej przy prędkości powyżej 38,400 bodów. Czteroportowe moduły ProLinx posiadają oddzielne sterowniki interfejsu szeregowego dla dodatkowych 3 portów. Porty te są buforowane i obsługują komunikację do 115,200 bodów.

### 3 Konfiguracja protokołu MCM

#### Zawartość rozdziału

- ❖ [Modbus Port x].....11
- ❖ [Komendy Modbus Port 0].....17

#Przykład portu Master używającego protokołu Modbus RTU

[Modbus Port 0]

# Ustawienia wymagane zarówno dla portu typu Master jak i Slave

```
Enabled          : Yes #Flaga port aktywny Y=Tak, N=Nie
RS Interface     : 0 #0=RS-232, 1=RS-485, 2=RS-422
Type            : Master #Typ portu M=Master, S=Slave

Float Flag      : No #Dane zmiennoprzecinkowe Y=Tak, N=Nie
Float Start     : 7000 #Offset rejestru we wiadomości dla danych
                #zmiennoprzecinkowych
Float Offset    : 2000 #Adres wewnętrzny dla danych
                #zmiennoprzecinkowych

Protocol        : RTU #Typ protokołu Modbus R=RTU, A=ASCII
Baud Rate       : 384 #Prędkość transmisji
Parity          : None #Parzystość N=Brak, O=Nieparzysty, E=Parzysty
Data Bits       : 8 #5 do 8 bitów danych w ramce wiadomości
Stop Bits       : 1 #1 lub 2 bity stopu w ramce wiadomości
RTS On          : 0 #Opóźnienie (ms)wysłania wiadomości po
                #ustawieniu sygnału RTS
RTS Off         : 1 #Opóźnienie (ms)wyłączenia sygnału RTS po
                #otrzymaniu wiadomości
Minimum Response Delay : 2 #Opóźnienie odpowiedzi (ms)
Use CTS Line    : No #Obsługa sygnału CTS linii modemu(Y/N)
Response Timeout : 1000 #Czas na odpowiedź (ms)
Retry Count     : 2 #Nastawa licznika braków odpowiedzi na komendy
```

# Ustawienia wymagane dla sterownika Slave

```
Internal Slave ID : 1 #Adres Modbus portu
Bit Input Offset  : 0 #Offset wejściowych danych dyskretnych w
                #wewnętrznej bazie danych
Word Input Offset : 0 #Offset wejściowych danych typu word w
                #wewnętrznej bazie danych
Output Offset     : 0 #Offset wyjściowych danych dyskretnych w
                #wewnętrznej bazie danych
Holding Register Offset : 0 #Offset wewnętrznej bazy danych dla danych
                #rejestrów z podtrzymaniem
Use Guard Band Timer : N #Obsługa czasu przerwy pakietowej (Y/N)
Guard Band Timeout : 0 #0=nastawa domyślna, lub ustawić czas (ms)
```

```
# Ustawienia wymagane dla sterownika Master
Minimum Command Delay : 0 #Odstęp pomiędzy komendami (ms)
Error Delay Counter : 100 #0-65535 nastawa licznika pomijanych komend po
#błędzie komunikacji
Command Control Reg : -1 #Rejestr kontrolny dla komend (-1=wyłącz)

[Modbus Port 0 Commands]
# Adres Odstęp Ilość Odwrócone Adres Funkcja Adres Modbus
# Włącz Wewnętrzny Komend Rej Słowa Węzła Modbus w urządzeniu
# (Internal (Poll (Reg (Swap (Node (Modbus (MB Address
#(Enable) Address) Interval) Count) Code) Address) Func) in Device)
START
    1 0 0 10 0 1 3 0
    1 10 0 10 0 1 4 10
END
```

### 3.1 [Modbus Port x]

Sekcja [Modbus Port 0] pliku **CFG** ustawia parametry master i slave portu komunikacyjnego Modbus, definiuje ustawienia protokołu oraz parametry listy komend.

#### 3.1.1 *Enabled*

Yes lub No

Parametr określa czy Port Modbus Master/Slave będzie używany. Jeśli parametr będzie ustawiony na No, port będzie nieaktywny. Nastawa Yes oznacza, że port będzie używany.

#### 3.1.2 *RS Interface*

1 = RS-232

2 = RS-422

3 = RS-485

Parametr jest używany, gdy są obecne Porty 1, 2, 3. Służy do konfiguracji interfejsu elektrycznego dla każdego z portów.

Port 1 musi być wybrany poprzez ustawienie zworki z tyłu modułu.

#### 3.1.3 *Type*

Master lub Slave

Parametr określa czy port będzie emulował urządzenie typu master czy slave. Należy użyć ustawienia Master dla wyboru typu master a Slave dla typu slave.

### 3.1.4 *Float Flag*

Yes lub No

Parametr określa, czy będzie zaimplementowana funkcjonalność dostępu do danych zmiennoprzecinkowych. Jeśli Float Flag jest ustawione na Yes, funkcje 3, 6 oraz 16 będą interpretowały wartości zmiennoprzecinkowe dla rejestrów, jako określone przez dwa kolejne parametry.

### 3.1.5 *Float Start*

0 do 65535

Parametr określa pierwszy rejestr danych zmiennoprzecinkowych. Wszystkie żądania z wartościami rejestru większymi lub równymi tej wartości będą traktowane, jako żądania danych zmiennoprzecinkowych. Parametr używany tylko, gdy Float Flag jest ustawiona na Yes. Przykład: jeśli zostanie ustawiona wartość 7000, wszystkie żądania dla rejestrów 7000 i wyżej będą traktowana, jako dane zmiennoprzecinkowe.

### 3.1.6 *Float Offset*

0 do 9999

Parametr określa rejestr, od którego zaczynają się dane zmiennoprzecinkowe w wewnętrznej bazie danych. Parametr używany tylko, gdy Float Flag jest ustawiona na Yes. Przykład: jeśli Float Offset jest ustawiony na 3000 i Float Start na 7000, to żądanie danych dla rejestru 7000 użyje wewnętrznego rejestru Modbus 3000.

### 3.1.7 *Protocol*

RTU

ASCII

Parametr określa typ protokołu Modbus, jaki będzie użyty w porcie. Obsługiwane typy to RTU oraz ASCII.

### 3.1.8 *Baud Rate*

Parametr określa prędkość portu. Należy wprowadzić prędkość, jako wartość.

Prędkość (ilość bodów)	Wartość parametru	Port
110	110	0
150	150	1
300	300	2
600	600	3
1200	12 lub 1200	
2400	24 lub 2400	
4800	48 lub 4800	

Prędkość (ilość bodów)	Wartość parametru	Port
9600	96 lub 9600	
14,400	14, 114 lub 14400	
19,200	19, 192 lub 19200	
28,800	28, 288 lub 28800	
38,400	38, 384 lub 38400	
57,600	57 lub 576	1,2,3
115,200	115 lub 1152	

### 3.1.9 Parity

None, Odd, Even

Parzystość jest prostym algorytmem kontroli błędów używanym w komunikacji szeregowej. Parametr określa typ kontroli parzystości.

Wszystkie urządzenia komunikujące się poprzez dany port muszą używać kontroli parzystości tego samego typu.

### 3.1.10 Data Bits

7 lub 8

Parametr określa ilość bitów danych ramki protokołu. Wszystkie urządzenia komunikujące się poprzez dany port muszą używać tej samej ilości bitów danych.

### 3.1.11 Stop Bits

1 lub 2

Bit stopu określa koniec znaku w strumieniu danych. Większość aplikacji używa jednego bitu stopu. Dla wolniejszych urządzeń, które potrzebują więcej czasu na synchronizację konieczne będzie wykorzystanie dwóch bitów stopu.

Wszystkie urządzenia komunikujące się przez ten port muszą mieć identycznie ustawioną ilość bitów stopu.

### 3.1.12 RTS On

0 do 65535 milisekund

Parametr określa opóźnienie po pojawieniu się sygnału RTS do transmisji danych. Opóźnienie podane w milisekundach.

### 3.1.13 RTS Off

0 do 65535 milisekund

Ten parametr określa opóźnienie pomiędzy wysłaniem ostatniego bajtu danych a ustawieniem sygnału RTS na niski.

### **3.1.14 Minimum Response Delay**

0 do 65535

Parametr określa opóźnienie przy wysyłaniu odpowiedzi na żądanie wysłane dla danego portu. Jest on konieczny dla pracy z wolnymi urządzeniami.

### **3.1.15 Use CTS Line**

Yes lub No

Parametr określa czy ma być wykorzystana kontrola linii modemu CTS. Jeżeli parametr jest ustawione na No linia CTS nie będzie monitorowana. W przeciwnym razie linia CTS będzie monitorowana i musi być stanie wysokim zanim modem rozpocznie transmisję danych. W normalnym przypadku kontrola jest wykorzystana przy dwuprzewodowej komunikacji urządzeń half-duplex.

### **3.1.16 Response Timeout**

0 do 65535 milisekund

Parametr ten określa timeout dla wiadomości podany w milisekundach. Jeżeli w ustawionym tutaj czasie, nie nastąpi odpowiedź do portu ustawionego jako Master, nada on ponownie żądanie do urządzenia Slave. Wartość jest ustawiana w zależności od obciążenia sieci oraz oczekiwanego czasu odpowiedzi najwolniejszego urządzenia.

### **3.1.17 Retry Count**

0 do 10

Parametr określa ile razy dane żądanie będzie przesłane ponownie w przypadku błędu komunikacji.

### **3.1.18 Internal Slave ID**

0 do 255

Parametr określa adres w sieci modbus, jaki zostanie przypisany do danego portu. Każde żądanie odnoszące się do tego adresu będzie obsługane przez moduł. Każde urządzenie musi mieć unikatowy adres w sieci.

### **3.1.19 Bit Input Offset**

0 do 3999

Parametr określa offset w wewnętrznej bazie danych, który ma być wykorzystany przy żądaniu z kodem funkcji 02. Jeżeli ta wartość będzie ustawiona przykładowo na 150, żądanie adresu 0 spowoduje odesłanie bitu z rejestru 150 z wewnętrznej bazy danych.

### **3.1.20 Word Input Offset**

0 do 3999

Parametr określa offset w wewnętrznej bazie danych modułu, wykorzystywany przy realizacji żądania z kodem funkcji 04. Jeżeli ta wartość będzie ustawiona przykładowo na 150, żądanie adresu 0 spowoduje odesłanie zawartości rejestru 150 z wewnętrznej bazy danych.

### **3.1.21 Output Offset**

0 to 3999

Parametr określa offset w wewnętrznej bazie danych modułu, wykorzystywany przy realizacji żądań z kodem funkcji 1, 5 lub 15. Jeżeli ta wartość będzie ustawiona przykładowo na 100, żądanie zapisu na adres 0 spowoduje zapisanie wartości do rejestru 100.

### **3.1.22 Holding Register Offset**

0 do 3999

Parametr określa offset w wewnętrznej bazie danych modułu, wykorzystywany przy realizacji żądań z kodem funkcji 3, 6 lub 16. Jeżeli ta wartość będzie ustawiona przykładowo na 50, żądanie adresu 0 spowoduje odesłanie zawartości rejestru 50 z wewnętrznej bazy danych.

### **3.1.23 Use Guard Band Timer**

Yes lub No

Parametr określa czy wykorzystany będzie Guard Band dla pakietów.

Przy wielu urządzeniach w sieci, moduł musi umieć zdefiniować okres czasu wykorzystywanego do poprawnego rozpoznania w sekwencji danych prawidłowej wiadomości protokołu Modbus. Ten czas jest definiowany przez parametr guardband timer. Niepoprawne ustawienie tej funkcjonalności może powodować błędy komunikacyjne w sieci modbus.

Funkcjonalność ta jest wykorzystywana jedynie w trybie Modbus RTU, kiedy moduł jest skonfigurowany jako urządzenie slave w sieci MultiDrop (Okablowanie RS-422 lub RS-485). W trybie RTU, po otrzymaniu przez moduł sekwencji danych z innego urządzenia, rozpoznany zostanie koniec wiadomości po wykryciu przerwy o długości 3.5 znaku. Następny bajt po przerwie wynoszącej 3.5 znaku będzie rozpoznany, jako początek nowej wiadomości.

Przerwa 3.5 znaku zależy od ustawienia komunikacji portu, w szczególności ustawienia prędkości. Im wyższa prędkość, tym krótszy czas przeznaczony na konieczną przerwę pomiędzy ramkami

**Aby włączyć parametr Guard Band Timeout:**

- 1** Najpierw należy włączyć tę funkcję przez ustawienie:

Use Guard Band Timer = Y

- 2** Następnie należy ustawić:

Guard Band Timeout = 0

Taka konfiguracja spowoduje użycie predefiniowanych interwałów czasowych dla wiadomości Modbus w zależności od wybranej prędkości transmisji. Dla większości aplikacji zapewni to optymalną wydajność

Następujące wartości Guard Band timer'a zostaną ustawione automatycznie, gdy parametr Guard Band Timeout zostanie ustawiony na 0:

<b>Prędkości</b>	<b>Guard Band Timer (ms)</b>
110	350
150	256
300	128
600	64
1200	32
2400	16
4800	8
9600	4
19200	2
28800	2
38400	2
57600	1
115200	1

Jeżeli mimo tego pojawia się znaczna liczba błędów komunikacji, należy podnosić wartość Guard Band Timeout aż do poprawienia pracy sieci.

**3.1.24 Guard Band Timeout**

0 do 65535

Parametr ten określa czas pomiędzy pakietami w trybie pracy multi-drop. Wartość 0 powoduje użycie domyślnego czasu. Wartość z zakresu 1 do 65535 ustawia czas w milisekundach. Jeżeli ten czas upłynie, i żadne dalsze dane nie zostaną odebrane, moduł rozpocznie przetwarzanie ostatniej sekwencji danych, jako żądania Modbus.



### **3.1.25 Minimum Command Delay**

0 do 65535

Parametr ten określa liczbę milisekund, jaką należy odczekać pomiędzy wysyłaniem kolejnych żądań do sieci. Parametru można użyć by zapobiec wysyłaniu zbyt dużej ilości żądań. Parametr ten nie wpływa na powtórzenia rozkazów, jeżeli nastąpi błąd komunikacji.

### **3.1.26 Error Delay Counter**

0 do 65535

Parametr ten określa ilość zapytań, jaką należy pominąć, przed próbą kolejnego nawiązania komunikacji z urządzeniem Slave. Jeżeli urządzenie Slave nie odpowie, Master pominie tyle wysyłanych do Slave komend, ile wynosi ten parametr.

### **3.1.27 Command Control Reg**

0 do 3900

Parametr ten kontroluje wykonanie rozkazów z listy użytkownika przez ustawienie wartości 0, 1 lub 2 przy każdym rozkazie. Jeżeli użytkownik ustawi wszystkie rozkazy z kodem 0, żaden z rozkazów się nie wykona. Jeżeli wartość pierwszego rejestru kontrolnego zostanie zmieniona na 1, komenda zerowa będzie wykonywać cyklicznie. Funkcję tę można wyłączyć ustawiając parametr na -1 lub pomijając jej nazwę w pliku konfiguracyjnym. Zastosowanie tej funkcji będzie wymagało 100 rejestrów w bazie danych modułu.

## **3.2 [Modbus Port 0 Commands]**

Część pliku konfiguracyjnego określona jako [Modbus Port 0 Commands]. Jest to lista odpytywanych urządzeń Slave. Moduł obsługuje kilka rodzajów żądań. Umożliwia to modułowi interakcję z szerokim zakresem urządzeń z protokołem Modbus.

Lista urządzeń jest sformatowana inaczej niż reszta pliku konfiguracyjnego. Urządzenia są wymienione pomiędzy etykietami **START** oraz **END**. Etykiety te informują program o położeniu listy. Program będzie odpytywał kolejne urządzenia slave dopóki dojdzie do etykiety **END** lub nie zostanie osiągnięta wprowadzona wartość licznika zapytań (command Count).

Format rozkazów jest identyczny a jego zawartość zależy od typu rozkazu. W tabeli na następnej stronie znajdują się rozkazy wraz z odpowiednim formatem danych.

Informacje o module ←					→ Informacje o urządzeniu			
# kolumny	1	2	3	4	5	6	7	8
Kod Funkcji	Kod Enable	Adres wewnętrzny	Interwał Odpytania	licznik	Kod zamiany	Adres Slave	Kod funkcji	Adres w Modbus urządzeniu
fc1	Kod	Bit	sekundy	licznik	0	Adres	1	Bit
fc2	Kod	Bit	sekundy	licznik	0	Adres	2	Bit
fc3	Kod	Rejestr	sekundy	licznik	Kod	Adres	3	Rejestr
fc4	Kod	Rejestr	sekundy	licznik	0	Adres	4	Rejestr
fc5	Kod	Bit	sekundy	licznik	0	Adres	5	Bit
fc6	Kod	Rejestr	sekundy	licznik	0	Adres	6	Rejestr
fc15	Kod	Bit	sekundy	licznik	0	Adres	15	Bit
fc16	Kod	Rejestr	sekundy	licznik	0	Adres	16	Rejestr

Pierwszą część wpisu stanowią dane dotyczące modułu (w tym przypadku modułu MCM), druga część dotyczy konkretnego urządzenia Slave. Szczegółowe informacje dotyczące każdej z funkcji znajdują się w dokumentacji dotyczącej protokołu Modbus.

Przykład listy zapytań.:

```
[PORT 0 COMMANDS]
#          INTERNAL  POLL          SWAP SLAVE FUNCTION DEVICE
# ENABLE ADDRESS INTERVAL COUNT CODE NODE CODE ADDRESS
START
  1   0   0   10   0   1   3   0
  1   0   0   10   0   1  16  10
END
```

### 3.2.1 Enable

0, 1, 2 lub 3

Definiuje czy i w jakich warunkach należy wywołać żądanie.

Jeżeli parametr jest ustawiony na 0, żądanie jest wyłączone i nie zostanie wykonane w trakcie sekwencji.

Ustawienie wartości 1, spowoduje wywołanie rozkazu w każdej sekwencji zapytań o ile Interwał odpytania jest ustawiony na 0. Jeżeli nie jest, żądanie zostanie wysłane po upływie czasu.

Ustawienie parametru na 2, spowoduje wywołanie rozkazu jedynie, gdy dane wewnętrzne skojarzone z instrukcją ulegną zmianie. Stan poprawny jedynie dla rozkazów zapisu (Write).

Jeżeli parametr zostanie ustawiony na 3, komenda zostanie wywołana jedynie w chwili, gdy dane z pamięci wewnętrznej skojarzonej z rozkazem ulegną zmianie. Dodatkowo spowoduje to również wyzerowanie bitów po utworzeniu komendy write. Ta funkcjonalność działa jedynie z funkcjami 5 i 15.

**Ważne:** Nie wszystkie rewizje modułu MCM obsługują ten parametr. By określić czy moduł obsługuje parametr, skontaktuj się ze wsparciem technicznym firmy Prosoft.

### 3.2.2 *Internal Address*

0 do 4999

lub

0 do 9999

Parametr określa adres w bazie danych modułu, skojarzony z danym żądaniem. W zależności od rodzaju funkcji, jest on interpretowany, jako zakres 1-bitowy lub 16-bitowy(1 słowo).

- Dla funkcji Modbus 1,2,5 i 15 parametr jest interpretowany, jako adresowanie bitowe.
- Dla funkcji Modbus 3, 4, 6 i 16 parametr jest interpretowany, jako adresowanie przez słowa.

Jeżeli instrukcja jest funkcją odczytu, odebrane dane są zapisywane począwszy od tej lokacji.

W przypadku zapisu, dane wysyłane do zapisu pochodzą z miejsca określonego przez pierwszy bit lub słowo.

### 3.2.3 *Poll Interval*

0 do 65535

Parametr określa minimalny interwał przy odpytywaniu cyklicznym (kod enable równy 1). Parametr jest wprowadzany w sekundach, co oznacza, że jeżeli wpisana zostanie wartość 10, żądanie nie będzie wysyłane częściej niż raz na 10 sekund.

### 3.2.4 *Reg Count*

#### **Rejestry**

1 do 125

#### **Cewki**

1 do 800

Parametr określa ile rejestrów lub cewek ma być skojarzonych z danym rozkazem.

Funkcje 5 i 6 ignorują to pole ponieważ odnoszą się jedynie do pojedynczego elementu.

Dla funkcji 1, 2 i 15, parametr ten określa ilość punktów cyfrowych (wejść lub cewek) skojarzonych z instrukcją.

Dla funkcji 3, 4 i 16 jest to ilość rejestrów skojarzonych z instrukcją.

### 3.2.5 Swap Code

0,1,2,3

Parametr ten decyduje czy dane otrzymane z serwera mają zostać zapisane z zamienioną kolejnością w stosunku do tych otrzymanych z serwera. Parametr jest pomocny przy obsłudze liczb zmiennoprzecinkowych lub innych wartości zajmujących wiele rejestrów, ponieważ nie ma standardowej metody zapisu tych danych w urządzeniach Master. Parametr można ustawić tak, by dane były ustawione w kolejności przydatnej innym aplikacjom. Tabela przedstawia wartości parametru oraz odpowiadające im operacje.

Swap Code	Opis
0	Brak zmiany – Nie nastąpią żadne zmiany w kolejności danych (1234 = 1234)
1	Słowa – Zmieniona zostaje kolejność słów (1234=3412)
2	Słowa i Bajty - Zamieniana jest kolejność słów, a następnie w słowach zamieniane są bajty. (1234=4321)
3	Bajty – Zamieniane są bajty w każdym słowie (1234=2143)

Słowa powinny być zamieniane tylko w przypadku wykorzystywania parzystej liczby słów w rozkazie.

### 3.2.6 Slave Address

1 do 255 (0 oznacza tryb broadcast)

Parametr określa adres urządzenia Slave w sieci Modbus, do którego odnosi się dany rozkaz. Dopuszczalne są wartości 1-255. Większość urządzeń pracujących w protokole Modbus obsługuje jedynie adresy 1-247 więc należy zachować ostrożność. Ustawienie wartości na 0, spowoduje wiadomość typu Broadcast do wszystkich urządzeń w sieci. Protokół modbus dopuszcza tryb broadcast dla komend zapisu. Nie należy go używać dla operacji odczytu.

### 3.2.7 Funkcja Modbus

1, 2, 3, 4, 5, 6, 15, 16

Parametr określa funkcję Modbus, która ma zostać wykonana danym żądaniem. Kody funkcji są zdefiniowane w Protokole Modbus. Tabela przedstawia listę funkcji obsługiwanych przez moduł MCM. Więcej informacji na temat protokołu Modbus znajduje się na stronie [WWW.modicon.com](http://WWW.modicon.com).

Kod Funkcji Modbus	Opis
1	Read Coil Status
2	Read Input Status
3	Read Holding Register
4	Read Input Register
5	Single Coil Write
6	Single Register Write
15	Multiple Coil Write
16	Multiple Register Write

### 3.2.8 Początkowy Adres Modbus w urządzeniu

Parametr ten określa początkowy rejestr lub punkt wejściowy, który ma być początkiem danych obsługiwanych przez żądania w urządzeniu Slave. Aby określić obszar danych każdego urządzenia w sieci, konieczne jest odniesienie się do dokumentacji każdego z urządzeń.

Kod funkcji określa czy ten parametr jest rejestrem początkowym czy offsetem dla danego obszaru. Przykładowo, jeżeli rozkaz dotyczy bitów (kody funkcji 1, 2, 5, lub 15) i odczytu/zapisu cewki 0X z adresu 00001, to wprowadzona tutaj wartość powinna być 0. Dla cewki z adresu 00110, powinna być tutaj wpisana wartość 109. W przypadku operacji na rejestrach (kody funkcji 3, 4, 6, lub 16) przykładowo dla rejestrów 30001 lub 40001, wpisana tutaj wartość powinna być 0. Gdyby dotyczyło to rejestrów 31101 lub 41101, wpisana tutaj wartość wynosiłaby 1100.

## 4 Połączenia komunikacyjne

### Zawartość rozdziału

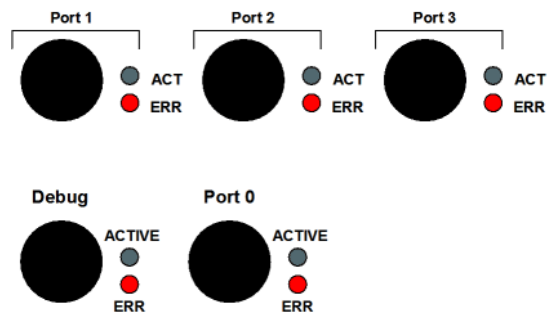
- ❖ Połączenia szeregowo – jednostki wieloportowe .....22

Ta część dokumentacji zawiera przyporządkowanie pinów w kablach oraz w portach szeregowych modułu ProLinx w zależności od protokołu (RS232/422/485).

Moduł ProLinx zostanie dostarczony z jednym do pięciu portów szeregowych w zależności od zakupionej konfiguracji. We wszystkich przypadkach, protokół dla danego portu będzie oznaczał identyczny układ pinów na wtyczkach.

**Przykład:** Moduł 4602-RIO-Modbus Master / Slave zawiera pięć portów szeregowych: cztery konfigurowalne porty Modbus Master/Slave oraz port Debug.  
Moduł 4601-RIO-Modbus Master / Slave zawiera tylko dwa porty szeregowo. Jeden konfigurowalny port Modbus Master / Slave oraz port Debug.

Każdy port szeregowy jest fizycznie złączem typu Mini-DIN. Do każdego aktywnego portu dołączane jest jeden 6-calowy kabel Mini-Din na DB-9M. Złącze DB-9M umożliwia połączenie do protokołów RS-232, RS-422 oraz RS-485. Schematy w kolejnych podrozdziałach pokazują dokładny układ kabla dla różnych połączeń fizycznych.



### 4.1 Podłączenia portów szeregowych: Jednostki wieloportowe

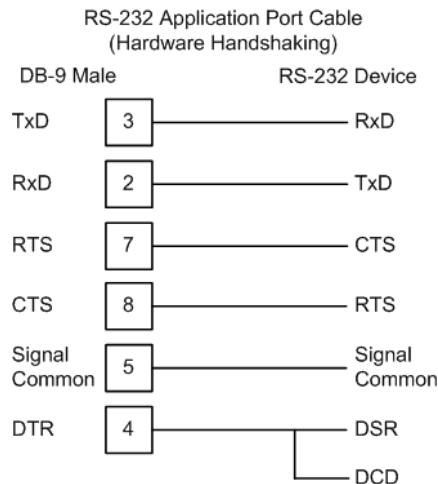
Zależność pomiędzy opisem portów z przodu modułu a aplikacją jest następująca:

**Te porty występują jedynie w jednostkach wieloportowych**

Port 1	Port Aplikacji 1
Port 2	Port Aplikacji 2
Port 3	Port Aplikacji 3

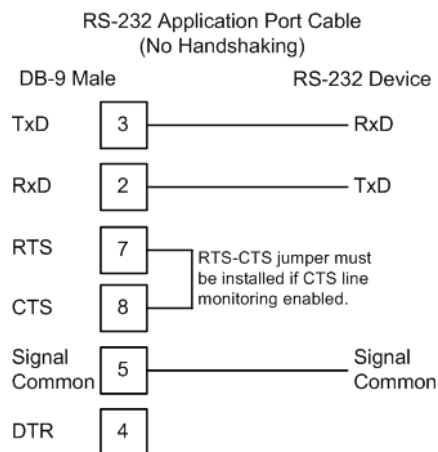
#### 4.1.1 Port 0, 1, 2, 3: RS-232 - Null Modem (z opcją handshaking)

Ten typ połączenia jest używany w momencie, gdy podłączone urządzenie wymaga opcji handshaking (kontrola i monitoring linii modemowych).



#### 4.1.2 Port 0, 1, 2, 3: RS-232 - Null Modem (bez opcji handshaking)

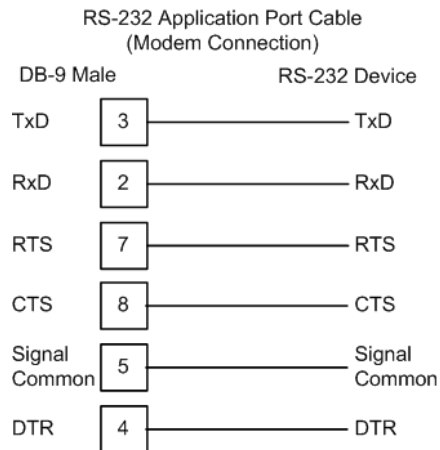
Ten typ połączenia może zostać użyty do połączenia modułu z komputerem lub portem komunikacyjnym innego urządzenia.



**Uwaga:** Jeżeli port został ustawiony z parametrem "Use CTS Line" na 'Y', konieczna jest zworka pomiędzy liniami RTS a CTS od strony modułu.

### 4.1.3 Podłączenie RS-232 do portów 0,1,2,3

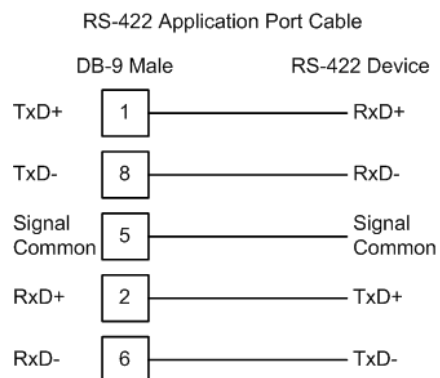
Na poniższym rysunku przedstawiony jest schemat podłączeń przy wybraniu interfejsu RS-232.



Parametr "Use CTS Line" powinien być wykorzystany do większości aplikacji modemowych.

### 4.1.4 Podłączenie RS-422 do portów 0,1,2,3

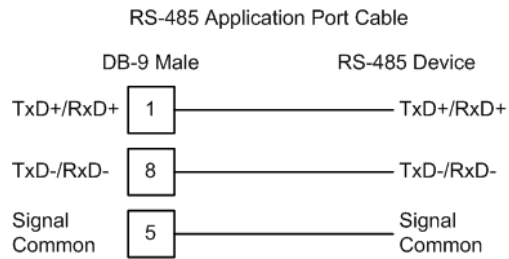
Na poniższym rysunku przedstawiony jest schemat podłączeń przy wybraniu interfejsu RS-422.





#### 4.1.5 Podłączenie RS-485 do portów 0,1,2,3

Na poniższym rysunku przedstawiony jest schemat podłączeń przy wybraniu interfejsu RS-485.



## 5 Diody LED

### Zawartość rozdziału

- ❖ Diody modułu bazowego .....26
- ❖ Diody dla portów 0,1,2,3 .....27
- ❖ Diody w urządzeniach serii 4101 .....27

Diody LED zapewniają możliwość monitorowania operacji systemu i pojedynczych portów i dlatego są bardzo przydatne do wstępnej diagnostyki. Moduł posiada diody dla każdego z portów, dodatkowo błędy konfiguracji, aplikacji oraz wskaźniki błędów są monitorowane i również sygnalizowane diodami LED by ostrzec przed możliwymi problemami.

Więcej informacji o diodach znajduje się w dokumencie *ProLinx Reference Guide*

### 5.1 Diody modułu bazowego

Dioda	Stan	Opis
Power	Wyłączony	Zasilanie modułu nie jest podłączone. Dioda jest zasilana z wewnętrznego zasilacza, więc wymaga jedynie podania napięcia.
	Stały Zielony	Zasilanie jest podłączone do modułu. Należy sprawdzić pozostałe diody by ocenić stan pracy urządzenia.
Fault	Wyłączony	Normalna Praca
	Stały czerwony	Nastąpił błąd krytyczny. Program przestał być wykonywany albo wskutek przerwania przez użytkownika albo błędu w pliku wykonywalnym. Należy nacisnąć Reset lub wyłączyć zasilanie by usunąć błąd. Jeżeli to nie pomaga należy wykorzystać procedurę Debug opisaną dalej w dokumencie.
Cfg	Wyłączony	Normalna Praca
	Stały żółty	Moduł jest w trybie konfiguracji. Plik konfiguracyjny jest odczytywany a jednostka implementuje wartości konfiguracyjne ora inicjuje hardware. Następuje to przy włączeniu zasilania lub po naciśnięciu klawisza reset. Nastąpi to również po resecie urządzenia lub przeładowaniu konfiguracji.
Err	Wyłączony	Normalna Praca
	migający	Występuje stan błędu. Należy sprawdzić konfigurację urządzenia.
	Stały czerwony	Nastąpiła znaczna ilość błędów w trakcie wykonywania aplikacji interfejsu komunikacyjnego. Flaga błędu modułu jest czyszczona na początku każdego żądania (Master/Serwer) lub po każdym otrzymanym żądaniu (slave/adapter/serwer).

## 5.2 Diody dla portów 0,1,2,3

Niektóre z modułów ProLinX mają dodatkowe cztery porty szeregowy. Każdy z tych portów ma dwie diody diagnostyczne

Dioda	Kolor	Opis
Port 0 - ACT	Wyłączony	Brak aktywności na porcie.
Port 1 - ACT	Migający zielony	Port albo odbiera albo wysyła dane
Port 2 - ACT		
Port 3 - ACT		
Port 0 - ERR	Wyłączony	Normalny stan. Jeżeli dioda jest wyłączona a dioda Port ACT wykazuje aktywność, oznacza to brak błędów komunikacyjnych.
Port 1 - ERR		
Port 2 - ERR	Żółty stały lub migający	
Port 3 - ERR		

## 5.3 Diody w urządzeniach serii 4101

Dioda	Stan	Opis
Power	Wyłączony	Zasilanie modułu nie jest podłączone. Dioda jest zasilana z wewnętrznego zasilacza, więc wymaga jedynie podania napięcia.
	Stały Zielony	Zasilanie jest podłączone do modułu. Należy sprawdzić pozostałe diody by ocenić stan pracy urządzenia.
Fault	Wyłączony	Normalna Praca
	Stały czerwony	Włączony jest tryb Configuration/Debug (odnosi się, do modułów obsługujących pass-through na porcie Debug jak np. DFCM moduły). Jeżeli dioda CFG LED nie jest włączona, nastąpił błąd krytyczny. Program przestał być wykonywany albo wskutek przerwania przez użytkownika albo błędu w pliku wykonywalnym. Należy nacisnąć Reset lub wyłączyć zasilanie by usunąć błąd. Jeżeli to nie pomaga należy wykorzystać procedurę Debug opisaną dalej w dokumencie.
CFG	Wyłączony	Normalna Praca
	Stały żółty	Jeżeli zapalona jest dioda Fault aktywny jest tryb Configuration/ Debug (o ile moduł obsługuje tryb pass-through na porcie Debug jak np. Moduły DFCM). Jeżeli dioda Fault jest wyłączona moduł jest w trybie konfiguracji. Plik konfiguracyjny jest odczytywany a jednostka implementuje wartości konfiguracyjne ora inicjuje hardware. Następuje to przy włączeniu zasilania lub po naciśnięciu klawisza reset. Nastąpi to również po resecie urządzenia lub przeładowaniu konfiguracji.
ERR	Wyłączony	Normalna Praca
	migający	Występuje stan błędu. Należy sprawdzić konfigurację urządzenia.
	Stały czerwony	Nastąpiła znaczna ilość błędów w trakcie wykonywania aplikacji interfejsu komunikacyjnego. Flaga błędu modułu jest czyszczona na początku każdego żądania (Master/Serwer) lub po każdym otrzymanym żądaniu (slave/adapter/serwer).

## 6 Błędy Modbus oraz Stany urządzeń.

### Zawartość rozdziału

- ❖ Oglądanie błędów oraz stanu urządzeń.....28
- ❖ Obszary danych błędów Modbus oraz stanów urządzeń.....28
- ❖ Kody błędów .....31

Drugą z i najbardziej przekrojową metodą diagnostyki problemów w module MCM jest wykorzystanie portu DEBUG znajdującego się na module. Daje to znacznie szerszy dostęp do wewnętrznych operacji oraz stanów modułu. Dostęp do możliwości portu Debug daje podłączenie komputera do portu oraz uruchomienie programu takiego jak Hyperterminal lub Prosoft Configuration Builder..

**Uwaga:** moduły z serii 4101 wykorzystujące obydwa porty szeregowy (Debug i Port x) jako porty aplikacyjne (np. 4101-MCM-DFCM) musi zostać zresetowany, lub mieć przełączone zasilania w celu włączenia funkcjonalności porty Debug. Aby włączyć port w tryb debug, należy podłączyć się do niego wykorzystując kabel null-modem oraz trzymać klawisz A w trakcie resetowania modułu karty. Nie należy zwalniać tego klawisza dopóki typer terminal nie pokaże menu portu debug. Tryb debug może zostać włączony na każdym z portów. Komenda upload/download jest dostępna jedynie przy podłączeniu do portu Debug.

### 6.1 Oglądanie błędów oraz stanu urządzeń.

Kolejne tematy w tym rozdziale przedstawiają rejestry adresowe zawierające dane o błędach i stanach. Wykorzystanie w głównym menu modułu ProLinx opcji Database View umożliwia podejrzenie zawartości każdego rejestru. Dodatkowe informacje o tej opcji znajdują się w dokumencie *ProLinx Reference Guide*.

### 6.2 Obszary danych błędów Modbus oraz stanów urządzeń

Błędy Modbus oraz dane statusowe zapisywane są w rejestrach w zależności od konfiguracji portów w module. Początkowe rejestry znajdują się w tabeli.

Port Modbus	Adres początkowy
0	4400
1	4800
2	5200
3	5600

**Uwaga:** Żaden z tych adresów nie jest dostępny przez standardowe adresowanie Modbus. W celu wykorzystania tych informacji w aplikacji należy przenieść je do zakresu adresów od 0 do 4000 za pomocą sekcji Data Map w pliku konfiguracyjnym. Przenoszenie danych jest opisane w odpowiednim rozdziale i przedstawia przykład do przeglądania takich danych.

Kolejne tematy opisują zakresy adresów dla błędów oraz stanów modułów..

### 6.2.1 Porty Modbus: Błędy i stany

W tej części dokumentacji omówione są błędy oraz stany portu szeregowego (Modbus Master/Slave).

Obszar danych jest inicjowany z zerami za każdym startem modułu. Dzieje się tak przy włączeniu modułu, resecie modułu, oraz przeładowaniu konfiguracji (ciepły restart).

Przykładowy adres w wewnętrznej bazie	Offset	Opis
4400	0	Ilość żądań z listy zapytań
4401	1	Ilość odpowiedzi z listy zapytań.
4402	2	Ilość błędów żądań command
4403	3	Ilość żądań
4404	4	Ilość odpowiedzi
4405	5	Ilość błędów wysłanych
4406	6	Ilość błędów odebranych
4407	7	Kod błędu konfiguracji
4408	8	Aktualny błąd/Indeks
4409	9	Ostatni błąd/Indeks

W dalszej części dokumentacji znajduje się omówienie kodów błędów, aby możliwe było odpowiednie zinterpretowanie kodów znajdujących się w obszarze danych

### 6.2.2 Master Port: Command List Errors

Pojedyncze błędy (command errors) są zwracane do adresów z poniższej tabeli.

Port Modbus	Obszar adresowy
0	4410 do 4509
1	4810 do 4909
2	5210 do 5309
3	5610 do 5709

Pierwsze słowo w zdefiniowanym obszarze rejestrów zawiera kod błędu/stanu pierwszego żądania w liście żądań portu. Każde kolejne słowo, odpowiada kolejnemu rozkazowi z pliku konfiguracyjnego.

Na stronie 31 znajdują się wyjaśnienia kodów błędów z obszaru danych.

*Przykładowa lista błędów dla portu 0*

Adres w bazie danych (przykład)	Offset	opis
4410	0	Stan błędu żądania #0
4411	1	Stan błędu żądania #1
4412	2	Stan błędu żądania #2
4413	3	Stan błędu żądania #3
4414	4	Stan błędu żądania #4
.	.	.
.	.	.
.	.	.
4507	97	Stan błędu żądania #97
4508	98	Stan błędu żądania #98
4509	99	Stan błędu żądania #99

Należy zwrócić uwagę, że wartości w powyższej tabeli są inicjowane na 3 w trakcie uruchomienia, resetu albo przeładowania konfiguracji.

### 6.2.3 Port Master: Stan urządzeń Slave

Lista stanu urządzeń Slave umożliwia sprawdzenie komunikacji z każdym urządzeniem Slave podłączonym do portu Master. Urządzenia Slave mogą mieć jeden z trzech stanów:

0	Slave jest nieaktywny i niezdefiniowany w liście portu Master
1	Slave jest z powodzeniem odpytywany przez urządzenie Master, komunikacja jest poprawna.
2	Port Master nie nawiązał komunikacji z urządzeniem Slave. Komunikacja jest zawieszona przez zdefiniowany przez użytkownika czas oparty o skanowanie listy rozkazów.

Urządzenia Slave są definiowane w systemie przy inicjalizacji listy żądań w urządzeniu Master. Stan każdego urządzenia Slave zostanie ustawiony na 1 w pierwszym kroku. Jeżeli port master nie nawiąże z nim komunikacji (upłynie liczba prób ustawiona w retry counter), urządzenie master ustawi stan Slave na 2 w tabeli stanów. Powoduje to zawieszenie komunikacji z danym urządzeniem na czas ustawiony w **Error Delay Counter**. Za każdym razem, gdy ma zostać wykonane żądanie dotyczące urządzenia Slave z zawieszoną komunikacją, wartość licznika zostanie zmniejszona o 1. Gdy dojdzie do 0 stan urządzenia Slave zostanie zmieniony na 1. Spowoduje to ponowne odpytywanie urządzenia Slave.

Pierwsze słowo w lokacji rejestrów zawiera kod stanu dla urządzenia z adresu 1. Każde kolejne słowo jest związane z kolejnymi urządzeniami aż do adresu 255.

Dla każdego portu, modbus zarezerwowany jest oddzielny obszar adresowy, w którym znajdują się Stany urządzeń Slave. Obszary znajdują się w poniższej tabeli.

Port Modbus	Obszar adresowy
0	4510 do 4765
1	4910 do 5165
2	5310 do 5565
3	5710 do 5965

Przykład listy stanów portu 0 (Slave)

Adres w bazie danych (przykład)	Offset	Opis
4510	0	Stan urządzenia Slave #1
4511	1	Stan urządzenia Slave #2
4512	2	Stan urządzenia Slave #3
4513	3	Stan urządzenia Slave #4
4514	4	Stan urządzenia Slave #5

Przykładowe adresy z tabeli zakładają korzystanie z portu 0. Należy pamiętać, że każdy port typu Master, będzie miał oddzielny obszar stanów dla urządzeń slave z własną lokacją oraz adresami.

Należy pamiętać, że lista urządzeń slave jest inicjalizowana (przy włączeniu zasilania, zimnym restarcie) z domyślnymi wartościami 0.

## 6.3 Kody błędów

W tym rozdziale znajdują się kody błędów protokołu Modbus lub rozszerzone kody dostępne jedynie dla modułu MCM.

### 6.3.1 Kody błędów Modbus

Znajdujące się w tabeli kody błędów są generowane zarówno przez porty master jak i slave. Są to standardowe błędy protokołu Modbus.

Kod	Opis
1	Nieprawidłowa funkcja
2	Nieprawidłowy adres danej
3	Nieprawidłowa wartość danej
4	Błąd w skojarzonym urządzeniu
5	Potwierdzenie
6	Zajęty, wiadomość odrzucona.

### 6.3.2 Kody błędów komunikacji modułu.

Kody błędów zwracane przy przetwarzaniu listy rozkazów są zapisane w obszarze pamięci dla błędów listy rozkazów. Dla każdego rozkazu alokowane jest jedno słowo w pamięci. Poniżej znajduje się tabela z możliwymi błędami.

Kod	Opis
-1	Linia kontrolna modemu (CTS) nieustawiona przed transmisją
-2	Timeout w trakcie nadawania wiadomości
-11	Timeout w oczekiwaniu na odpowiedź po nadaniu żądania
253	Nieprawidłowy adres urządzenia Slave w odpowiedzi
254	Nieprawidłowy kod funkcji w odpowiedzi
255	Nieprawidłowa wartość CRC/LRC w odpowiedzi.

Kody pomagają w określeniu czy wykonanie rozkazu zakończyło się sukcesem. W przypadku, gdy rozkaz nie zostanie wykonany kody błędów pomagają określić przyczynę niepowodzenia.

### 6.3.3 Kody błędów listy rozkazów.

Kody błędów zwracane przy przetwarzaniu listy rozkazów (wykryte w trakcie uruchomienia modułu lub jego resetu) są zapisane w obszarze pamięci dla błędów listy rozkazów. Dla każdego rozkazu jest w pamięci alokowane jedno słowo. Poniżej znajduje się lista kodów błędów:

Kod	Opis
-41	Nieprawidłowy kod enable
-42	Wewnętrzny adres > maksymalny adres
-43	Nieprawidłowy adres urządzenia (<0 lub > 255)
-44	Parametr licznika ustawiony na 0
-45	Nieprawidłowy kod funkcji
-46	Wszystkie parametry ustawione na 0
-47	Wszystkie parametry ustawione na -1

Kodów błędów należy użyć do wykrycia błędnych wartości konfiguracji w liście rozkazów pliku konfiguracyjnego.



### 6.3.4 Słowo błędów konfiguracji protokołu Modbus

Słowa błędów konfiguracji protokołu Modbus są zapisywane w specyficznych dla protokołu rejestrach. W tabeli znajduje się lista rejestrów z odniesieniem do odpowiedniego portu komunikacyjnego.

Port Modbus	Rejestr słów błędów konfiguracji
0	4407
1	4807
2	5207
3	5607

Rejestr zawierający kod z problemem konfiguracyjnym. Poniższa tabela przedstawia kody błędów, opis błędów oraz konieczny do poprawienia parametr w pliku konfiguracyjnym.

Bit	Kod	Opis
0	0x0001	Nieprawidłowy parametr Enable (Yes lub No)
1	0x0002	Nieprawidłowy parametr interfejsu szeregowego (0 do 2)
2	0x0004	Nieprawidłowy typ (Master lub Slave)
3	0x0008	Nieprawidłowy protokół (RTU lub ASCII)
4	0x0010	Nieprawidłowa prędkość transmisji
5	0x0020	Nieprawidłowa kontrola parzystości (None, Odd, Even)
6	0x0040	Nieprawidłowa ilość bitów danych (7 lub 8 bitów)
7	0x0080	Nieprawidłowa ilość bitów stopu (1 lub 2)
8	0x0100	Nieprawidłowe użycie CTS (Yes lub No)
9	0x0200	Nieprawidłowe ustawienie licznika retry count (0 do 10)
10	0x0400	Nieprawidłowe dane dla operacji zmiennoprzecinkowych: Flaga Float w stanie innym niż Yes lub No Float Start mniejszy od 0 lub nieprawidłowy Float Offset
11	0x0800	Nieprawidłowy adres wewnętrznego Slave (1 do 255) (Slave)
12	0x1000	Nieprawidłowy wpis w polu Register Offset Data (Slave)
13	0x2000	Zarezerwowany
14	0x4000	Zarezerwowany
15	0x8000	Zarezerwowany

## 7 Referencje

### Zawartość Rozdziału

❖ Przenoszenie danych .....	34
❖ Plik konfiguracyjny 4102-MBS-MBM .....	34
❖ Konfiguracja Rozkazów Modbus.....	41
❖ Specyfikacja Protokołu Modbus.....	45

### 7.1 Przenoszenie danych

Poniżej znajduje się przykład sekcji mapowania danych w pliku konfiguracyjnym. Sekcja ta umożliwi użytkownikowi przenoszenie danych wewnątrz bazy danych w celu stworzenia podobnych żądań danych i kontroli.

```
# Ta sekcja przenosi dane wewnątrz bazy danych aby skupić dane dla stworzenia
# prostszych zapytań i kontroli. From Address wskazuje lokacją początkową w
# bazie danych z której należy skopiować ilość rejestrów zapisanych w Register
# Count do adresów wskazanych w To Address. Po skopiowaniu danych kolejność
# bajtów może zostać zmieniona przy użyciu pola Swap Code jak poniżej:
# SWAP CODE      DEFINITION
0 Bytes left in original order (1234 -> 1234)
1 Words are swapped (1234 -> 3412)
2 Words and bytes are swapped (1234 -> 4321)
3 Bytes in each word are swapped (1234 -> 2143)
[DATA MAP]
# From      To      Register  Swap  Delay
# Address   Address  Count    Code  Preset
START
  4000      1000      9         0     1000
  6300      1010      20        0     1002
  4400      1030      20        0     1003
END
```

## 7.2 Plik konfiguracyjny 4102-MBS-MBM

### 7.2.1 [Modbus Master Port]

#### Internal Slave ID

0 do 255

Ten parametr definiuje wirtualny adres w sieci Modbus dla urządzenia Slave w wewnętrznej bazie danych. Każde żądanie z tym adresem otrzymane przez port będzie przetworzone przez moduł. Każde urządzenie musi mieć unikatowy adres w sieci.

#### Protokół

RTU lub ASCII

Parametr określa typ protokołu Modbus wykorzystanego na porcie. Dostępne protokoły to RTU i ASCII.

#### Prędkość Transmisji

Prędkość, z jaką pracuje port. Należy tutaj wprowadzić prędkość, jako wartość.

Prędkość	Wartość Parametru	Port
110	110	0
150	150	1
300	300	2
600	600	3
1200	12 lub 1200	
2400	24 lub 2400	
4800	48 lub 4800	
9600	96 lub 9600	
14,400	14, 114 lub 14400	
19,200	19, 192 lub 19200	
28,800	28, 288 lub 28800	
38,400	38, 384 lub 38400	
57,600	57 lub 576	1,2,3
115,200	115 lub 1152	

#### Parity

None, Odd, Even

Parzystość to prosty algorytm kontroli wykorzystywany w komunikacji szeregowej. Określa on typ użytej kontroli parzystości. Wszystkie urządzenia w sieci muszą mieć identycznie ustawioną kontrolę parzystości.

### Data Bits

7 lub 8

Parametr ten określa, jaką ilość bitów w każdym słowie wykorzystuje protokół. Wszystkie urządzenia w sieci muszą mieć identycznie ustawione bity danych.

### Stop Bits

1 lub 2

Bit stopu określa koniec znaku w strumieniu danych. Większość aplikacji używa jednego bitu stopu. Dla wolniejszych urządzeń, które potrzebują więcej czasu na synchronizację konieczne będzie wykorzystanie dwóch bitów stopu.

Wszystkie urządzenia w sieci muszą mieć identycznie ustawioną ilość bitów stopu.

### RTS On

0 do 65535 milisekund

Parametr określa opóźnienie po pojawieniu się sygnału RTS do transmisji danych. Opóźnienie podane w milisekundach.

### RTS Off

0 do 65535 milisekund

Ten parametr określa opóźnienie pomiędzy wysłaniem ostatniego bajtu danych a ustawieniem sygnału RTS na niski.

### Minimum Response Delay

0 do 65535

Parametr określa opóźnienie przy wysyłaniu odpowiedzi na żądanie wysłane dla danego portu. Jest on konieczny dla pracy z wolnymi urządzeniami.

### Use CTS Line

Yes lub No

Parametr określa czy ma być wykorzystana kontrola linii modemu CTS. Jeżeli parametr jest ustawione na No linia CTS nie będzie monitorowana. W przeciwnym razie linia CTS będzie monitorowana i musi być stanie wysokim zanim modem rozpocznie transmisję danych. W normalnym przypadku kontrola jest wykorzystana przy dwuprzewodowej komunikacji urządzeń half-duplex.

### Response Timeout

0 do 65535 milisekund

Parametr określa timeout odpowiedzi na żądanie w inkrementach 1 milisekundowych. Jest to czas, jaki urządzenie master będzie oczekiwało przed ponowną transmisją żądania, jeżeli nie zostanie odebrana odpowiedź od urządzenia Slave. Wartość należy ustawiać w zależności od obciążenia sieci i od czasu odpowiedzi najwolniejszego urządzenia w sieci.

### Retry Count

0 do 10

Parametr określa ilość retransmisji żądania w przypadku wykrycia Timeout'u.

## **7.2.2 [Modbus Slave Port x]**

### Enabled

Yes lub No

Flaga określa czy wykorzystany będzie port Modbus Master modułu. Jeżeli parametr zostanie ustawiony na No, port nie będzie używany. Ustawienie na Yes, oznacza, że port będzie używany.

### RS Interface

1 = RS-232

2 = RS-422

3 = RS-485

Parametr określa elektryczny interfejs poszczególnych portów modułu. Port 1 musi zostać skonfigurowany za pomocą zworki umieszczonej z tyłu modułu.

### Protocol

RTU lub ASCII

Parametr określa typ protokołu Modbus wykorzystanego na porcie. Poprawne protokoły to RTU i ASCII.

### Baud Rate

Parametr określa prędkość transmisji portu. Należy ustawić ją, jako wartość.

Prędkość	Wartość Parametru	Port
110	110	0
150	150	1
300	300	2
600	600	3
1200	12 lub 1200	
2400	24 lub 2400	
4800	48 lub 4800	
9600	96 lub 9600	
14,400	14, 114 lub 14400	
19,200	19, 192 lub 19200	
28,800	28, 288 lub 28800	
38,400	38, 384 lub 38400	
57,600	57 lub 576	1,2,3
115,200	115 lub 1152	

### Parity

None, Odd, Even

Parzystość to prosty algorytm kontroli wykorzystywany w komunikacji szeregowej. Określa on typ użytej kontroli parzystości. Wszystkie urządzenia komunikujące się na danym porcie muszą mieć identycznie ustawioną kontrolę parzystości.

### Data Bits

7 lub 8

Parametr określa ilość bitów danych wykorzystanych przez protokół. Wszystkie urządzenia komunikujące się przez ten port muszą mieć identycznie ustawione bity danych.

### Stop Bits

1 lub 2

Bit stopu określa koniec znaku w strumieniu danych. Większość aplikacji używa jednego bitu stopu. Dla wolniejszych urządzeń, które potrzebują więcej czasu na synchronizację konieczne będzie wykorzystanie dwóch bitów stopu.

Wszystkie urządzenia komunikujące się przez ten port muszą mieć identycznie ustawioną ilość bitów stopu.

### RTS On

0 do 65535 milisekund

Parametr określa opóźnienie po pojawieniu się sygnału RTS do transmisji danych. Opóźnienie podane w milisekundach.

### RTS Off

0 do 65535 milisekund

Ten parametr określa opóźnienie pomiędzy wysłaniem ostatniego bajtu danych a ustawieniem sygnału RTS na niski.

### Minimum Response Delay

0 do 65535

Parametr określa opóźnienie przy wysyłaniu odpowiedzi na żądanie wysłane dla danego portu. Jest on konieczny dla pracy z wolnymi urządzeniami.

### Use CTS Line

Yes lub No

Parametr określa czy ma być wykorzystana kontrola linii modemu CTS. Jeżeli parametr jest ustawione na No linia CTS nie będzie monitorowana. W przeciwnym razie linia CTS będzie monitorowana i musi być stanie wysokim zanim modem rozpocznie transmisję danych. W normalnym przypadku kontrola jest wykorzystana przy dwuprzewodowej komunikacji urządzeń half-duplex.

### Use Guard Band Timer

Yes lub No

Parametr określa czy wykorzystany będzie Guard Band dla pakietów.

Przy wielu urządzeniach w sieci, moduł musi umieć zdefiniować okres czasu wykorzystywanego do poprawnego rozpoznania w sekwencji danych prawidłowej wiadomości protokołu Modbus. Ten czas jest definiowany przez parametr guardband timer. Niepoprawne ustawienie tej funkcjonalności może powodować błędy komunikacyjne w sieci modbus.

Funkcjonalność ta jest wykorzystywana jedynie w trybie Modbus RTU, kiedy moduł jest skonfigurowany, jako urządzenie slave w sieci MultiDrop (Okablowanie RS-422 lub RS-485). W trybie RTU, po otrzymaniu przez moduł sekwencji danych z innego urządzenia, rozpoznany zostanie koniec wiadomości po wykryciu przerwy o długości 3.5 znaku. Następny bajt po przerwie wynoszącej 3.5 znaku będzie rozpoznany, jako początek nowej wiadomości.

Przerwa 3.5 znaku zależy od ustawienia komunikacji portu, w szczególności ustawienia prędkości. Im wyższa prędkość, tym krótszy czas przeznaczony na konieczną przerwę pomiędzy ramkami.

**Aby włączyć parametr Guard Band Timeout:**

**3** Najpierw należy włączyć tę funkcję przez ustawienie:

Use Guard Band Timer = Y

**4** Następnie należy ustawić:

Guard Band Timeout = 0

Taka konfiguracja spowoduje użycie predefiniowanych interwałów czasowych dla wiadomości Modbus w zależności od wybranej prędkości transmisji. Dla większości aplikacji zapewni to optymalną wydajność

Następujące wartości Guard Band timer'a zostaną ustawione automatycznie, gdy parametr Guard Band Timeout zostanie ustawiony na 0:

Prędkości	Guard Band Timer (ms)
110	350
150	256
300	128
600	64
1200	32
2400	16
4800	8
9600	4
19200	2
28800	2
38400	2
57600	1
115200	1

Jeżeli mimo tego pojawia się znaczna liczba błędów komunikacji, należy podnosić wartość Guard Band Timeout aż do poprawienia pracy sieci.

**Guard Band Timeout**

0 do 65535

Parametr ten określa czas pomiędzy pakietami w trybie pracy multi-drop. Wartość 0 powoduje użycie domyślnego czasu. Wartość z zakresu 1 do 65535 ustawia czas w milisekundach. Jeżeli ten czas upłynie, i żadne dalsze dane nie zostaną odebrane, moduł rozpocznie przetwarzanie ostatniej sekwencji danych, jako żądania Modbus.

**Request Timeout**

Najlepiej wykorzystać domyślną wartość timeout'u, ustawioną w tym miejscu.



### Busy Timeout

0 do 65535

Parametr określa liczbę milisekund, jaką należy odczekać zanim port DF1 Master będzie dostępny. Jeżeli port DF1 master przetwarza aktualnie rozkaz, zostanie ustawiona flaga zajęty. Flaga pozostanie ustawiona do zakończenia operacji. Jeżeli port nie będzie dostępny po tym czasie wiadomość zostanie przerwana. Jeżeli port Master będzie dostępny przed upływem określonego w tym parametrze czasu, wiadomość zostanie przekazana do tego portu master.

## 7.3 Konfiguracja rozkazów MODBUS

Urządzenia MCM Modbus Master/Slave firmy Prosoft obsługują znaczną ilość rozkazów odczytu, zapisu. Przy konfiguracji portu Master typ użytego rozkazu jest wybierany na podstawie rodzaju danych, które mają zostać zaadresowane i poziomu obsługi protokołu Modbus w urządzeniu Slave. Przy konfiguracji portu Slave ważne może się okazać zrozumienie funkcjonowania rozkazów Modbus w celu dobrania odpowiedniej struktury danych.

Do dokumentacji załączono wyciąg ze specyfikacji protokołu MODBUS (strona 34) by pomóc w zrozumieniu funkcjonalności protokołu MODBUS.

### 7.3.1 Rozkazy MCM

Moduł MCM obsługuje rozkazy protokołu Modbus związane przede wszystkim z odczytem i zapisem danych. Ten rozdział przedstawia rozkazy obsługiwane przez moduł. Większość informacji będzie przydatna dla osób implementujących protokół Master, ale część informacji będzie również przydatna dla implementujących protokół Modbus Slave.

Kod Funkcji	Rozkaz	Zakres Adresów	Komentarz do protokołu Slave	Komentarz do protokołu Master
1	Read Output Status	Cewki 0001 do 9999	Moduł Zwraca binarne dane z obszaru 'Output Status'. Moduł obsługuje zapytania do 125 słów w jednym rozkazie.	<b>Device Address:</b> Początkowy bit w urządzeniu slave, od którego należy zacząć odczyt <b>Count:</b> ilość bitów do odczytu (maksymalnie 125 słów) <b>Internal Address:</b> Początkowy adres w pamięci urządzenia gdzie dane mają zostać zapisane.

Kod Funkcji	Rozkaz	Zakres Adresów	Komentarz do protokołu Slave	Komentarz do protokołu Master
2	Read Input Status	Bit 10001 do 29999	Moduł Zwraca binarne dane z obszaru 'Input Status'. Obsługuje zapytania do 125 słów w jednym rozkazie.	<p><b>Device Address:</b> : Początkowy bit w urządzeniu slave, od którego należy zacząć odczyt. Wprowadzenie 0 oznacza adres 10001</p> <p><b>Count:</b> ilość bitów do odczytu (maksymalnie 125 słów)</p> <p><b>Internal Address:</b> Początkowy adres w pamięci urządzenia gdzie dane mają zostać zapisane..</p>
3	Read Multiple Registers	Rejestry 40001 do 47999	Moduł zwraca słowa z obszaru rejestrów. Adresowalny obszar rejestrów w module jest złożony z 5000 słów. Słowo 0 odpowiada adresowi 40001 protokołu Modbus. Moduł obsługuje zapytania do 125 słów w jednym rozkazie.	<p><b>Device Address:</b> : Początkowy rejestr w urządzeniu slave, od którego należy zacząć odczyt. Wprowadzenie 0 oznacza adres 40001</p> <p><b>Count:</b> Ilość słów, jaką należy odczytać. (maksymalnie 125 słów)</p> <p><b>Internal Address:</b> Początkowy adres w pamięci urządzenia gdzie dane mają zostać zapisane.</p> <p><b>Type:</b> kontroluje zamianę bajtów oraz słów dla odczytów liczb zmiennoprzecinkowych.</p>
4	Read Input Registers	Rejestry 30001 do 39999	Moduł zwraca rejestry z obszaru 'Input Register' Moduł obsługuje zapytania do 125 słów w jednym rozkazie.	<p><b>Device Address:</b> Początkowy rejestr w urządzeniu slave, od którego należy zacząć odczyt. Wprowadzenie 0 oznacza adres 30001</p> <p><b>Count:</b> Ilość słów, jaką należy odczytać. (Maksymalnie 125 słów)</p> <p><b>Internal Address:</b> Początkowy adres w pamięci urządzenia gdzie dane mają zostać zapisane.</p>
5	Single Bit/Coil Write	Cewki 0001 do 9999	<p><b>Tryb Normalny:</b> Bit zostanie zapisany w obszarze danych modułu.</p> <p><b>Tryb Pass-Through:</b> Bit zostanie przesłany do sterownika, PLC/SLC do obsłużenia przez logikę.</p>	<p><b>Internal Address:</b> Adres bitu w module MCM, który ma zostać użyty do określenia typu rozkazu (set/reset)</p> <p><b>Count:</b> Nieużywany, domyślnie 1</p> <p><b>Device Address:</b> Adres bitu w urządzeniu Slave, na którym ma zostać wykonana operacja. Wprowadzenie 0 oznacza cewkę 0001 w urządzeniu slave.</p>

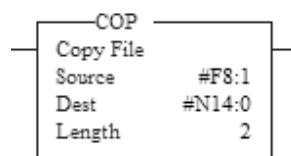
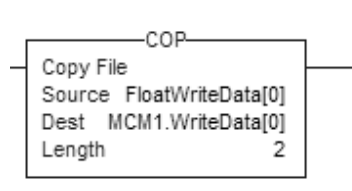
Kod Funkcji	Rozkaz	Zakres Adresów	Komentarz do protokołu Slave	Komentarz do protokołu Master
6	Single Register Write	Rejestry 40001 do 47999	<p><b>Tryb Normal:</b> Bit zostanie zapisany w obszarze danych modułu.</p> <p><b>Tryb Pass-Through:</b> Bit zostanie przesłany do sterownika, PLC/SLC do obsłużenia przez logikę.</p>	<p><b>Internal Address:</b> Adres rejestru w module MCM, który ma być źródłem danych.</p> <p><b>Count:</b> Nieużywany, domyślnie 1</p> <p><b>Device Address:</b> Rejestr w urządzeniu Slave do którego mają zostać zapisane dane. Wprowadzenie 0 oznacza rejestr 40001 w urządzeniu slave</p>
15	Multiple Bit/Coil Write		<p><b>Tryb Normal:</b> Zapisane bity zostaną umieszczone w pamięci modułu. Maksymalnie 125 słów</p> <p><b>Tryb Pass-Through:</b> Bity zostaną przesłane dalej do PLC/SLC do obsłużenia przez logikę. Maksymalnie 30 słów danych binarnych.</p>	<p><b>Internal Address:</b> Adres pierwszego bitu w module MCM, który ma zostać zapisany do urządzenia Slave</p> <p><b>Count:</b> ilość bitów do zapisania (maksymalnie 125 słów)</p> <p><b>Device Address:</b> Adres pierwszego bitu w urządzeniu slave do którego należy zapisać dane. Wprowadzenie 0 oznacza cewkę 0001 w urządzeniu slave.</p>
16	Multiple Register Write		<p><b>Tryb Normal:</b> Zapisane rejestry zostaną umieszczone w pamięci modułu. Maksymalnie 125 słów.</p> <p><b>Tryb Pass-Through:</b> Rejestry zostaną przesłane do PLC/SLC do obsłużenia przez logikę. Maksymalnie 60 słów.</p>	<p><b>Internal Address:</b> Adres pierwszego rejestru w module MCM, który ma zostać zapisany do urządzenia Slave</p> <p><b>Count:</b> ilość słów do zapisania (maksymalnie 125 słów)</p> <p><b>Dest Addr:</b> Adres pierwszego rejestru w urządzeniu slave do którego należy zapisać dane. Wprowadzenie 0 oznacza rejestr 40001 w urządzeniu slave.</p>

### 7.3.2 Obsługa wartości zmiennoprzecinkowych

Przenoszenie danych zmiennoprzecinkowych pomiędzy modulem MCM a innymi urządzeniami jest łatwe o ile urządzenia obsługują liczby rzeczywiste w formacie IEEE 754. Format ten jest formatem dla 32-bitowych liczb zmiennoprzecinkowych.

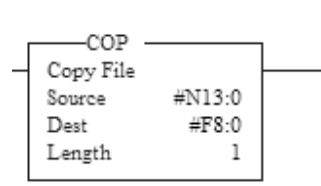
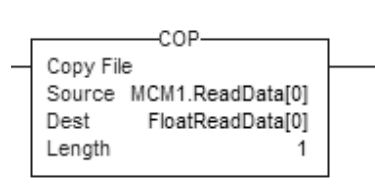
Programowanie konieczne do przeniesienia liczby zmiennoprzecinkowej polega na wykorzystaniu zalet istniejącej w sterowniku SLC oraz Logix instrukcji COP. Unikatowość tej instrukcji polega na tym, że przy przenoszeniu danych nie dokonuje się żadnej konwersji pomiędzy typami danych (kopiowany jest obraz danych a nie wartości)

Struktura instrukcji COP do przenoszenia danych z liczby zmiennoprzecinkowej do liczby całkowitej (operacja którą trzeba wykonać aby liczbę zmiennoprzecinkową zapisać do modułu):



Instrukcja spowoduje przeniesienie jednej wartości 32-bitowej do dwóch liczb 16-bitowych w pliku Integer. Przy przenoszeniu większej liczby danych należy w parametrze Length ustawić wielokrotność liczby 2.

Struktura instrukcji COP do przenoszenia danych z liczb całkowitych do liczby zmiennoprzecinkowej (operacja którą trzeba wykonać aby liczbę zmiennoprzecinkową odczytać z modułu):



Ta komenda przenosi dwie liczby 16-bitowe całkowite do jednej 32-bitowej liczby zmiennie przecinkowej.

### Obsługa liczb zmiennoprzecinkowych ENRON

Wielu producentów zaimplementowało specjalną wersją protokołu MODBUS znaną jako wersja ENRON. W tej implementacji rejestry adresowe większe niż 7000 są uważane za wartości zmiennoprzecinkowe. Ma to o tyle istotne znaczenie, że w tym wykonaniu pole **Count** staje się polem ilości wartości gdzie każda wartość reprezentuje dwa słowa 16-bitowe.

## 7.4 Specyfikacja protokołu MODBUS

### 7.4.1 Read Output Status (Kod funkcji 01)

#### Żądanie

Funkcja umożliwia odczytanie stan cewek logicznych używanych do kontroli wyjść dyskretnych jedynie z zaadresowanego urządzenia Slave. Nie jest obsługiwany tryb broadcast. Oprócz adresu urządzenia Slave oraz kodu funkcji wiadomość wymaga podania adresu od którego należy rozpocząć odczyt cewek, oraz długość obszaru jaki będzie odczytany.

Adresowanie umożliwia odczytanie do 2000 cewek w każdym żądaniu. Niektóre urządzenia mogą mieć dodatkowe ograniczenia prowadzące do zmniejszenia ogólnej liczby cewek. Cewki są numerowane od zerowej (numer cewki 1 = zerowa, numer cewki 2 = pierwsza, numer cewki 3 = druga, itd.).

Poniższa tabela jest przykładowym żądaniem odczytu cewek 0020 do 0056 z urządzenia z adresu 11.

Adres	Funkcja	Początek danych bajt wysoki	Początek danych bajt niski	Ilość danych bajt wysoki	Ilość danych bajt niski	Pole kontroli błędów
11	01	00	13	00	25	CRC

#### Odpowiedź

Poniżej znajduje się przykładowa odpowiedź na żądanie Read Output Status. Na każdą cewkę przypada jeden bit. Odpowiedź zawiera adres urządzenia Slave, kod funkcji, ilość znaków z danymi, dane oraz kontrolę błędów. Najniższy bit zawiera cewkę która została bezpośrednio zaadresowana. Dla ilości cewek niebędącej wielokrotnością liczby osiem, ostatni znak będzie uzupełniony zerami. Ilość znaków z danymi jest określona przez ilość znaków RTU niezależnie czy wykorzystaniu jest protokół RTU czy ASCII.

Ponieważ interfejs Slave jest obsługiwany na końcu skanu procesora, dane będą przedstawiać stan cewek na końcu skanu procesora. Niektóre urządzenia Slave ograniczają ilość dostępnych na końcu skanu cewek; stąd przy znacznej liczbie cewek, konieczne są wielokrotne żądania odczytujące stan po poszczególnych skanach programu.

Adres	Funkcja	Ilość Bajtów	Stan cewek 20 do 27	Stan cewek 28 do 35	Stan cewek 36 do 43	Stan cewek 44 do 51	Stan cewek 52 do 56	Pole kontroli błędów
11	01	05	CD	6B	B2	OE	1B	CRC

Stan cewek 20 do 27 jest pokazany jako CD(HEX) = 1100 1101 (Binarnie). Czytając od lewej do prawej widać że cewki 27, 26, 23, 22 i 20 są w stanie wysokim. Pozostałe dane dla cewek są dekodowane podobnie. Ze względu na ilość żądanych statusów, ostatnie pole danych, 1B (HEX) = 0001 1011 (Binarnie), zawiera stan jedynie 5 cewek (52 do 56) zamiast 8 cewek. 3 Bity z lewej strony są ustawione na 0 aby dopełnić format 8-bitowy.

#### 7.4.2 Read Input Status (Kod funkcji 02)

##### Żądanie

Funkcja ta umożliwia odczytanie stanu wejść dyskretnych w urządzeniu Slave. Nieobsługiwany jest tryb broadcast. Oprócz adresu urządzenia Slave, kodu funkcji wiadomość musi zawierać początkowy adres do odczytania oraz ilość bitów jaka ma zostać odczytana.

Adresowanie umożliwia odczytanie do 2000 wejść w każdym żądaniu. Niektóre urządzenia mogą ograniczać maksymalną ilość wejść. Wejścia są numerowane od zerowego (wejście 10001 = zerowe, wejście 10002 = pierwsze, wejście 10003 = drugie, itd., dla procesora 584).

W tabeli znajduje się przykładowe żądanie "read input status" mające na celu odczytanie wejść od 10197 do 10218 z urządzenia Slave o adresie 11.

Adres	Funkcja	Początek danych bajt wysoki	Początek danych bajt niski	Ilość danych bajt wysoki	Ilość danych bajt niski	Pole kontroli błędów
11	02	00	C4	00	16	CRC

##### Odpowiedź

Przykładowa odpowiedź na żądanie znajduje się w tabeli poniżej. Każdemu wejściu odpowiada jeden bit danych. Odpowiedź zawiera adres urządzenia Slave, kod funkcji, ilość znaków z danymi, dane oraz kontrolę błędów. Najniższy bit zawiera zaadresowane bezpośrednio wejście. Dla ilości wejść która nie jest wielokrotnością liczby 8, brakujące bity będą ustawiane na zero od najbardziej znaczącego. Ilość znaków z danymi zawsze jest podawana w liczbie znaków RTU, niezależnie czy wykorzystywany jest protokół RTU czy ASCII.

Ponieważ interfejs slave jest obsługiwany na końcu skanu programu, dane będą odzwierciedlały stan wejść z końca skanu programu. Niektóre urządzenia slave ograniczają dostępną na końcu skanu liczbę wejść; stąd przy znacznej liczbie wejść, konieczne są wielokrotne żądania, odczytujące stan po każdym Skanie programu.

Adres	Funkcja	Ilość Bajtów	Wejścia dyskretne 10197 do 10204	Wejścia dyskretne 10205 do 10212	Wejścia dyskretne 10213 do 10218	Pole kontroli błędów
11	02	03	AC	DB	35	CRC

Stan wejść 10197 do 10204 jest pokazany jako AC (HEX) = 10101 1100 (binarnie). Czytając od lewej do prawej oznacza to że wejścia 10204, 10202, i 10199 są w stanie wysokim. Pozostałe wejścia dekoduje się identycznie.

Ze względu na ilość żądanych stanów wejść, ostatni znak pokazany jako 35 HEX = 0011 0101 (binarnie) zawiera stan jedynie 6 wejść (10213 do 10218) zamiast 8. Dwa bity z lewej strony mają wartość 0 aby uzupełnić format 8-bitowy

### 7.4.3 Read Holding Registers (Kod funkcji 03)

#### Żądanie

Funkcja Read holding registers (03) umożliwia użytkownikowi odczytanie rejestrów adresowanych 4xxxx w urządzeniu Slave. Rejestry mogą zawierać wartości liczbowe dla skojarzonych z nimi timerów i liczników. Adresowanie umożliwia odczytanie do 125 rejestrów w każdym żądaniu; Niektóre urządzenia mogą ograniczać maksymalną ilość dostępnych rejestrów. Rejestry są numerowane od zerowego (40001 = zerowy, 40002 = pierwszy, itd.) . Tryb broadcast jest niedopuszczalny.

Przykład przedstawia odczyt rejestrów 40108 do 40110 z urządzenia slave o adresie 11.

Adres	Funkcja	Początek danych bajt wysoki	Początek danych bajt niski	Ilość danych bajt wysoki	Ilość danych bajt niski	Pole kontroli błędów
11	03	00	6B	00	03	CRC

#### Odpowiedź

Slave będący odbiorcą żądania odpowiada przesyłając swój adres, kod funkcji oraz pole danych. Pole danych zawiera jeden bajt mówiący o ilości danych wysyłanych. Dane z 16-bitowych rejestrów umieszczane są w dwóch bajtach. Pierwszy bajt zawiera bity bardziej znaczące a drugi bajt, bity mniej znaczące.

Ponieważ interfejs Slave jest obsługiwany na końcu skanu, dane będą pokazywały stan rejestrów na końcu skanu. Niektóre urządzenia slave ograniczają ilość rejestrów dostępnych na końcu skanu; stąd przy znacznej liczbie rejestrów konieczne będą wielokrotne żądania, odczytujące stan po każdym Skanie programu.

W załączonym przykładzie rejestry 40108 do 40110 mają wartość dziesiętną równą odpowiednio 555, 0 i 100.

Adres	Funkcja	Ilość bajłów	Bajt wysoki	Bajt niski	Bajt wysoki	Bajt niski	Bajt wysoki	Bajt niski	Pole kontroli błędów
11	03	06	02	2B	00	00	00	64	CRC

#### 7.4.4 Read Input Registers (Kod funkcji 04)

##### Żądanie

Kod funkcji 04 umożliwia odczytanie rejestrów wejściowych z adresów 3xxxx. Rejestry te zawierają dane od urządzeń wejściowych podłączonych do struktury We/Wy I mogą być od strony sterownika jedynie odczytywane a niezmiennie. Adresowanie umożliwia odczytanie do 125 rejestrów w każdym żądaniu. Niektóre urządzenia Slave mogą ograniczać maksymalną ilość rejestrów. Rejestry są numerowane od zerowego (30001 = zerowy, 30002 = pierwszy, itd.). Tryb broadcast jest niedozwolony.

Przykład pokazuje żądanie zawartości rejestru 3009 w urządzeniu z adresu 11.

Adres	Funkcja	Początek danych bajt wysoki	Początek danych bajt niski	Ilość danych bajt wysoki	Ilość danych bajt niski	Pole kontroli błędów
11	04	00	08	00	01	CRC

##### Odpowiedź

Slave będący odbiorcą żądania odpowiada swoim adresem, kodem funkcji po którym znajduje się pole danych. W polu danych znajduje się jeden bajt wskazujący długość danych wysłanych. Każdy z żądanych rejestrów znajduje się w dwóch bajtach. Pierwszy bajt zawiera bardziej znaczące bity a drugi bajt, mniej znaczące bity.

Ponieważ interfejs slave jest obsługiwany na końcu skanu programu, dane odzwierciedlają stan rejestrów na końcu skanu. Każde urządzenie ogranicza maksymalną liczbę dostępnych rejestrów; stąd przy znacznych ilościach rejestrów konieczne są wielokrotne żądania aby odczytać komplet danych po każdym Skanie.

W następującym przykładzie rejestr 3009 zawiera dziesiętną wartość 0.

Adres	Funkcja	Ilość bajtów	Bajt wysoki Rejestru	Bajt niski rejestru	Pole kontroli błędów
11	04	02	00	00	E9

#### 7.4.5 Force Single Coil (Kod Funkcji 05)

##### Żądanie

Ta funkcja wymusza stan pojedynczej cewki albo w stan wysoki albo niski. Każda cewka może mieć wymuszony stan wysoki lub niski. Sterownik cały czas wykonuje program, wobec tego może on zmieniać cyklicznie stan cewki chyba że dana cewka będzie wyłączona. Cewki są numerowane od zerowej (cewka 0001 = zerowa, cewka 0002 = pierwsza, itd.). Wartość 65,280 (FF00 HEX) ustawi cewkę w stan wysoki, a wartość 0 w stan niski. Inne wartości są nieprawidłowe i zostaną przez cewkę zignorowane.

Wykorzystanie adresu 00( Tryb Broadcast) wymusi tę cewkę we wszystkich urządzeniach Slave.



**Uwaga:** Funkcje 5, 6, 15 i 16 są jedynymi funkcjami obsługującymi tryb broadcast.

Na przykładzie pokazano żądanie aby Slave z adresu 11 włączył cewkę 0173.

Adres	Funkcja	Numer cewki bajt wysoki	Numer cewki bajt niski	Stan	Dane	Pole kontroli błędów
11	05	00	AC	FF	00	CRC

### Odpowiedź

Normalna odpowiedź na takie żądanie polega na retransmisji wiadomości, gdy stan cewki zostanie zmieniony.

Adres	Funkcja	Numer cewki bajt wysoki	Numer cewki bajt niski	Stan	Dane	Pole kontroli błędów
11	05	00	AC	FF	00	CRC

Wymuszenie stanu cewki kodem funkcji 5 będzie wykonane niezależnie od tego czy cewka jest włączona czy nie. (W produktach Prosoftu cewka jest tylko zmieniana gdy zaimplementowana jest odpowiednia logika).

**Uwaga:** Protokół Modbus nie zawiera instrukcji do Testu oraz zmiany stanu DISABLE wejść lub wyjść cyfrowych. Tam gdzie jest to konieczne, może być to wykonane za pomocą specjalnych komend programowania (w urządzeniach Prosoftu może to być tylko wykonane za pomocą logiki).

Cewki które są ustawiane w logice programu, nie są zerowane przy włączeniu zasilania. Wobec tego jeżeli taka cewka zostanie włączona z pomocą funkcji 5 pozostanie ona włączona przez cały czas a wyjście podłączone do tej cewki będzie w stanie wysokim.

## 7.4.6 Preset Single Register (Kod Funkcji 06)

### Żądanie

Funkcja (06) umożliwia użytkownikowi modyfikację pojedynczego rejestru typu holding register. Każdy rejestr obecny w sterowniku może zostać zmieniony za pomocą takiego żądania. Ponieważ jednak sterownik wykonuje program cyklicznie, możliwe jest zmodyfikowanie dowolnego rejestru w dowolnym czasie. Wartości są w formacie binarnym aż do maksymalnego zakresu sterownika. Nieużywane bity muszą być ustawione na 0. Przy użyciu w trybie broadcast (adres slave równy 0) wszystkie urządzenia slave załadują zadaną wartość do zadanego rejestru.

Adres	Funkcja	Adres rejestru bajt wysoki	Adres rejestru bajt niski	Nowa wartość bajt wysoki	Nowa wartość bajt niski	Pole kontroli błędów
11	06	00	01	00	03	CRC

### Odpowiedź

Odpowiedź na żądanie ustawienia pojedynczego rejestru polega na powtórny transmitowaniu wiadomości z żądaniem

Adres	Funkcja	Bajt wysoki adresu	Bajt niski adresu	Wartość rejestru bajt wysoki	Wartość rejestru bajt niski	Pole kontroli błędów
11	06	00	01	00	03	CRC

### 7.4.7 Diagnostics (Kod funkcji 08)

Kod funkcji 08 w protokole MODBUS zapewni serię testów sprawdzających komunikację między klientem a serwerem lub do sprawdzenia różnych stanów błędów wewnętrznych serwera.

Funkcja wykorzystuje dwu-bajtowy kod by zdefiniować jaki rodzaj testów ma zostać przeprowadzony. Serwer w normalnej odpowiedzi odsyła kod funkcji oraz kod testu. Niektóre z operacji diagnostycznych powodują odesłanie informacji o wynikach testów z serwera do zdalnego urządzenia.

W ogólnym przypadku, wysłanie żądania diagnostyki, nie wpływa na normalne wykonanie aplikacji w zdalnym urządzeniu. Logika użytkownika oraz rejestry z danymi nie są sprawdzane przez diagnostykę. Niektóre z funkcji mogą zerować liczniki w zdalnych urządzeniach.

Serwer może zostać wprowadzony w tryb 'Listen Only ' w którym będzie on monitorował przychodzące wiadomości ale nie będzie na nie odpowiadał. Może to wpłynąć na działanie aplikacji końcowej jeżeli jej działanie zależy od danych ze zdalnego urządzenia. W ogólnym przypadku, tryb ten jest wykorzystywany do usunięcia wadliwego urządzenia z sieci. Następujące funkcje diagnostyczne są dostępne jedynie dla urządzeń z portem szeregowym.

Normalna odpowiedź na żądanie Return Query Data to odesłanie tych samych danych. Również odsyłany jest kod funkcji oraz kod testu.

### Żądanie

Kod funkcji	1 Bajt	0x08
Kod-testu	2 Bajty	
Dane	N x 2 Bajty	

### Odpowiedź

Kod funkcji	1 Bajt	0x08
Kod-testu	2 Bajty	
Dane	N x 2 Bajty	

### Błąd

Kod błędu	1 Bajt	0x88
Kod wyjątku	1 Bajt	01 lub 03 lub 04

Kody testu dostępne dla urządzeń szeregowych.

Każdy kod testu jest pokazany z odpowiadającym mu polem danych. Kody testu obsługiwane przez moduł MSM są pogrubione.

Kod Testu		Nazwa
Hex	Dec	
<b>00</b>	<b>00</b>	<b>Return Query Data</b>
01	0	Restart Communications Option
02	02	Return Diagnostic Register
03	03	Change ASCII Input Delimiter
04	04	Force Listen Only Mode
05..09	05..09	Zarezerwowane
0A	10	Clear Counters and Diagnostic Register
0B	11	Return Bus Message Mount
0C	12	Return Bus Communication Error Mount
0D	13	Return Bus Exception Error Mount
0E	14	Return Slave Message Mount
0F	15	Return Slave No Response Mount
10	16	Return Slave NAK Mount
11	17	Return Slave Busy Mount
12	18	Return Bus Character Overrun Count
13	19	Zarezerwowane
14	20	Clear Overrun Counter and Flag
N.A.	21..65535	Zarezerwowane

**00 Return Query Data**

Dane przesłane w polu danych mają zostać przesłane z powrotem do urządzenia. Cała odpowiedź powinna być identyczna z żądaniem.

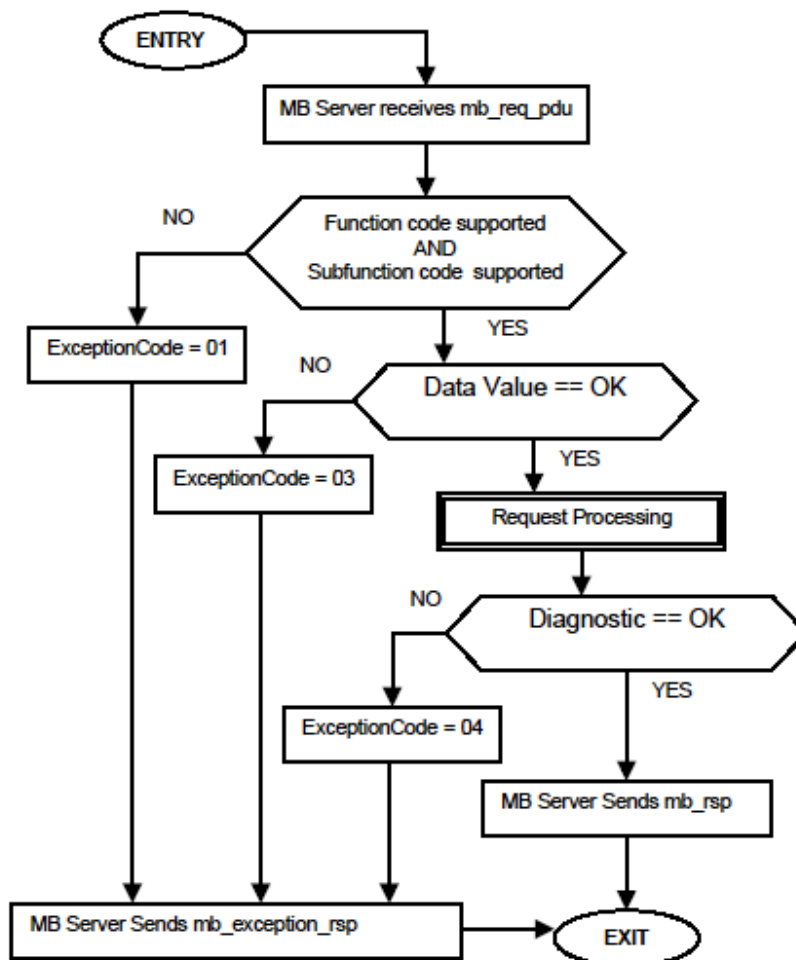
Kod Testu	Pole Danych (żądanie)	Pole danych (odpowieź)
00 00	Dowolne	Powtórzone dane z żądania

**Przykład i diagram stanów.**

Poniżej znajduje się przykładowe żądanie typu Return Query Data do zdalnego urządzenia. Wykorzystywany jest kod testu 0 (00 00 w dwóch bajtach zapisanych w HEX). Dane do odesłania znajdują się w polu danych (A5 37 hex) złożonym z dwóch bajtów.

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	08	Function	08
Sub-function Hi	00	Sub-function Hi	00
Sub-function Lo	00	Sub-function Lo	00
Data Hi	A5	Data Hi	A5
Data Lo	37	Data Lo	27

Pole danych w odpowiedzi na inne żądania może zawierać liczniki błędów lub inne dane zależne od kodu testu.



#### 7.4.8 Force Multiple Coils (Kod Funkcji 15)

##### Żądanie

Ta wiadomość wymusza stan wysoki lub niski w bloku cewek. Każda z występujących w sterowniku cewek może zostać wystawiona w stan wysoki lub niski. Ponieważ sterownik aktywnie wykonuje program, to cewki które nie są wyłączone mogą ulec zmianie w wyniku logiki programu. Cewki są numerowane od zerowej (cewka 00001 = zerowa, cewka 00002 = pierwsza, itd.) Żądaniu stanowi każdej cewki odpowiada jeden bit w polu danych. Wykorzystanie trybu broadcast spowoduje wymuszenie stanu bloku cewek we wszystkich urządzeniach Slave znajdujących się w sieci.

**Uwaga:** Funkcje 5, 6, 15, i 16 są jedynymi funkcjami mogącymi być wywoływane przy komunikacji typu broadcast.

Następujący przykład wymusza stan 10 cewek zaczynając od 20 (13HEX). Dwa pola danych CD = 1100 i 00 = 0000 000, oznaczają że cewki 27, 26, 23, 22 i 20 mają mieć zawsze stan wysoki..

Adres	Funkcja	Wysoki bajt adresu	Niski bajt adresu	ilość	Ilość bajtów	Stan cewek 20 do 27	Stan cewek 28 do 29	Pole kontroli błędów
11	0F	00	13	00	0A	02	CD	00 CRC

#### Odpowiedź

Normalna odpowiedź zawiera adres urządzenia slave, kod funkcji, początkowy adres oraz ilość wymuszonych cewek.

Adres	Funkcja	Wysoki bajt adresu	Niski bajt adresu	ilość	Pole kontroli błędów
11	0F	00	13	00	0A CRC

Zapisanie cewek za pomocą funkcji 15 zostanie wykonane niezależnie od tego czy cewki są włączone czy nie.

Cewki które nie są wykorzystane w logice procesora, nie są zerowane przy uruchomieniu systemu. W związku z tym taka cewka jeżeli zostanie włączona kodem funkcji 15, to w żadnym momencie zmiany stanu sterownika, cewka nie wróci do stanu 0.

### 7.4.9 Preset Multiple Registers (Kod funkcji 16)

#### Żądanie

Wartości istniejących w sterowniku rejestrów typu „Holding registers” mogą mieć zmienianą wartość za pomocą tej funkcji (maksymalnie 60 rejestrów). Ponieważ jednak sterownik cały czas wykonuje program zmiany mogą ulegać dowolne rejestry w dowolnej chwili czasu. Wartości są wysyłane binarnie aż do maksymalnej dostępnej dla sterownika (16-bitów dla 184/384 oraz 584); nieużywane bity wyższego rzędu muszą zostać ustawione na 0.

**Uwaga:** Kody funkcji 5, 6, 15 i 16 są jedynymi kodami które zadziałają przy komunikacji typu broadcast.

Adres	Funkcja	Adres wysoki	Adres niski	Ilość słów	Ilość bajtów	Dane wysoki	Dane niski	Dane wysoki	Dane niski	Pole kontroli błędów
11	10	00	87	00	02 04	00	0A	01	02	CRC

#### Odpowiedź

Normalna odpowiedź na żądanie numer 16 to odesłanie adresu, kodu funkcji początkowego adresu oraz ilości rejestrów do zapisania.

Adres	Funkcja	Wysoki bajt adresu	Niski bajt adresu	ilość	Pole kontroli błędów.
11	10	00	87	00 02	56

### 7.4.10 Kody wyjątków protokołu MODBUS

Kiedy klient wysyła żądanie do serwera oczekuje normalnej odpowiedzi. Możliwy jest jeden z czterech scenariuszy od strony urządzenia Master:

- Kiedy serwer odbierze zapytanie bez błędów komunikacji i może normalnie je obsłużyć, wysyła normalną odpowiedź.
- Jeżeli serwer nie otrzyma zapytania w związku z błędem komunikacji, żadna odpowiedź nie jest wysyłana. Klient po pewnym czasie ustawi dla żądania stan timeout.
- Jeżeli serwer otrzyma żądanie ale wykryje w nim błędy (parzystość, LRC, CRC itp. ) żadna odpowiedź nie jest wysyłana. Klient po pewnym czasie ustawi dla żądania stan timeout.
- Jeżeli serwer odbierze żądanie poprawnie ale nie jest w stanie go obsłużyć (np. Odczyt nieistniejącego rejestru), zwróci on wiadomość o wyjątku informującą klienta o rodzaju błędu.

Wiadomość o wyjątku ma dwa pola które odróżniają ją od normalnej odpowiedzi:

**Pole kodu funkcji :** W przypadku normalnej odpowiedzi, w polu tym serwer powtarza kod funkcji z żądania. Wszystkie kody funkcji mają najbardziej znaczący bit (MSB) równy 0 (ich wartości są poniżej 80 szesnastkowo). W odpowiedzi z wyjątkiem MSB jest ustawiany na 1. Powoduje to, że wartość kodu funkcji wynosi powyżej 80 szesnastkowo i jest to wartość wyższa niż dla normalnej odpowiedzi.

Z ustawionym MSB, aplikacja klienta może rozpoznać odpowiedź z wyjątkiem i sprawdzić pole danych pod kątem kodu błędu.

**Pole danych:** W normalnej odpowiedzi, serwer zwraca dane lub statystykę w polu danych (w zależności od żądanych informacji). W odpowiedzi o wyjątku, serwer zwraca kod wyjątku w tym polu. Identyfikuje on błąd który spowodował wysłanie wyjątku.

Poniższa tabela przedstawia przykładowe żądanie oraz odpowiedź z wyjątkiem.

Żądanie		Odpowiedź	
Nazwa pola	(Hex)	Nazwa pola	(Hex)
Funkcja	01	Funkcja	81
Adres początkowy Hi	04	Kod wyjątku	02
Adres początkowy Lo	A1		
Ilość wyjść bajt Hi	00		
Ilość wyjść bajt Lo	01		

W tym przykładzie klient wysyła żądanie do serwera. Kod funkcji (01) oznacza odczyt statusów wyjść. Żądany jest status wyjść z adresu 1245 (04A1 hex). Jest to jedyny żądany status gdyż taka ilość jest określona w polu ilości wyjść (0001).

Jeżeli adres wyjściowy nie istnieje, serwer zwróci odpowiedź z wyjątkiem I kodem wyjątku jak pokazano w przykładzie (02). Oznacza to zapytanie o nieistniejący adres.

Kod wyjątków MODBUS

Kod	Nazwa	Znaczenie
01	Nieprawidłowa Funkcja	Kod funkcji otrzymany w żądaniu nie jest dopuszczalną funkcją dla serwera. Może to być efektem tego, że kod jest dostępny jedynie w nowych urządzeniach i nie został zaimplementowany w wybranym urządzeniu. Może to również znaczyć, że serwer jest w nieprawidłowym trybie do przetworzenia żądania tego typu np. gdy urządzenie nie jest skonfigurowane a otrzyma żądanie „return register values”.
02	Nieprawidłowy Adres danych	Adres danych otrzymany w zapytaniu, nie jest dopuszczalnym adresem dla serwera. Dokładniej rzecz biorąc kombinacja adresu początkowego oraz ilości rejestrów jest nieprawidłowa. Dla urządzenia o 100 rejestrach żądanie z offsetem 96 i długością 4 skończy się powodzeniem. To samo żądanie z długością 5 spowoduje wygenerowanie wyjątku 02.
03	Nieprawidłowa wartość danych	Wartość wpisana w pytaniu nie jest prawidłową wartością dla serwera. Wskazuje to na błąd w strukturze w złożonym żądaniu (np. Nieprawidłowa długość). Nie oznacza to że dane przeznaczone do zapisu mają wartość spoza zakresu oczekiwanego przez aplikację, ponieważ protokół Modbus nie rozpoznaje znaczenia poszczególnych typów.
04	Awaria urządzenia Slave	Nastąpił błąd krytyczny urządzenia w chwili gdy serwer próbował wykonać żadaną operację.
05	Potwierdzenie	Specjalne użycie w połączeniu z komendami programowania. Serwer przyjął żądanie i przetwarza je ale czas przetwarzania może być długi. Wiadomość ta jest generowana aby zapobiec błędowi timeout po stronie klienta. Klient może w następnej kolejności wysłać zapytanie o zakończenie przetwarzania.
06	Urządzenie Slave zajęte	Specjalne użycie w połączeniu z komendami programowania. Serwer jest zaangażowany w przetwarzanie długotrwałego żądania. Po otrzymaniu tego wyjątku klient powinien przestać żądać ponownie po upływie pewnego czasu.
08	Błąd parzystości pamięci	Specjalne użycie w połączeniu z kodami funkcji 20 i 21 i typem odniesienia 6, do wskazania że rozszerzony obszar pliku nie przeszedł testu spójności. Serwer próbował odczytać wskazany obszar ale wykrył błąd parzystości w pamięci. Klient może powtórzyć żądanie ale konieczny może okazać się serwis urządzenia Serwer.
0a	Ścieżka do bramy niedostępna	Specjalne użycie w połączeniu z Bramami, wskazuje że brama nie była w stanie ustawić ścieżki z portu wejściowego do portu wyjściowego dla zadanego żądania. Zwykle znaczy to że brama jest źle skonfigurowana lub przeładowana.
0b	Docelowa brama nie odpowiedziała.	Specjalne użycie w połączeniu z bramami, oznacza że nie otrzymano odpowiedzi od urządzenia docelowego. Zwykle oznacza to że urządzenie nie jest obecne w sieci.

## 8 Wsparcie, Serwis i Gwarancja

### W tym rozdziale

- ❖ Wsparcie techniczne.....56
- ❖ Warunki Procedury Zwrotu Towaru .....57
- ❖ Ograniczona Gwarancja .....58

ProSoft Technology, Inc. (ProSoft) stara się zapewnić możliwie najbardziej skuteczne wsparcie techniczne. Przed wykonaniem telefonu prosimy o zebranie następujących informacji w celu przyspieszenia procesu:

- 1 Numer wersji urządzenia
- 2 Architektura systemu
- 3 Konstrukcja sieci

Jeżeli problem jest związany z urządzeniem będą również potrzebne:

- 1 Konfiguracja modułu i zawartość pliku
  - o Praca modułu
  - o Informacje o konfiguracji i błędach modułu
  - o Stan diod LED
- 2 Informacja o procesorze oraz pliku danych użytkownika oraz stan diod na urządzeniu.
- 3 Szczegóły dotyczące urządzeń podłączonych przez port szeregowy.

### 8.1 Wsparcie techniczne

---

<b>Internet</b>	witryna: <a href="http://www.prosoft-technology.com/support">www.prosoft-technology.com/support</a> Adres Email: <a href="mailto:support@prosoft-technology.com">support@prosoft-technology.com</a>
-----------------	--

---

#### **Azja, Australia, Oceania**

+603.7724.2080, [support.asia@prosoft-technology.com](mailto:support.asia@prosoft-technology.com)  
Obsługa w językach chińskim i angielskim.

#### **Europa (Toulouse, Francja)**

+33 (0) 5.34.36.87.20, [support.EMEA@prosoft-technology.com](mailto:support.EMEA@prosoft-technology.com)  
Obsługa w językach francuskim i angielskim

#### **Ameryka Północna/Południowa (oprócz Brazylii) (Kalifornia)**

+1.661.716.5100, [support@prosoft-technology.com](mailto:support@prosoft-technology.com)  
Obsługa w językach angielskim i hiszpańskim

*W przypadku problemów technicznych na terenie USA system odpowiadania po godzinach umożliwi dostęp przez pager do jednego z naszych specjalistów w każdej chwili.*

#### **Brazylia (Sao Paulo)**

+55-11-5084-5178, [eduardo@prosoft-technology.com](mailto:eduardo@prosoft-technology.com)  
Obsługa w językach angielskim i portugalskim.



## **8.2 Warunki Procedury Zwrotu Towaru**

Następujące warunki Procedury Zwrotu towaru dotyczą każdego produktu zwracanego firmie Prosoft. Warunki mogą ulec zmianie bez informowania o nich klientów. Warunki gwarancji znajdują się w rozdziale Ograniczona Gwarancja. W przypadku niezgodności pomiędzy warunkami gwarancji a Procedurą zwrotu towaru, za obowiązujące uznaje się warunki gwarancji.

### **8.2.1 Zwrot dowolnego produktu:**

- a) W celu zwrotu dowolnego produktu do naprawy, wymiany lub w jakimkolwiek innym celu klient musi otrzymać od firmy Prosoft numer RMA(Return Material Authorization ) i postępować zgodnie z instrukcjami dotyczącymi wysyłki.
- b) W przypadku problemu z dowolnym produktem należy skontaktować się z pomocą techniczną(numery tel. Na stronie 50).Inżynier wsparcia technicznego poprosi o przeprowadzenie kilku testów w celu zlokalizowania problemu. Jeżeli problemem okaże się urządzenie wystawiony zostanie numer RMA.
- c) Każdy zwracany produkt musi zostać wysłany opłaconą przesyłką w opakowaniu oryginalnym lub podobnym do miejsca wskazanego przez Prosoft. Do przesyłki należy dołączyć dowód zakupu. Numer RMS musi zostać umieszczony na opakowaniu w widoczny sposób. Klient zobowiązuje się ubezpieczyć przesyłkę lub przyjąć ryzyko utraty przesyłki w transporcie. Produkt wysłany do firmy prosoft w sposób inny niż określono, lub bez numeru RMA zostanie odesłany do klienta z pobraniem.
- d) W przypadku zwrotu zakupionego urządzenia które okazało się niepotrzebne, nadmiarowe itp., pobierana jest opłata w wysokości 10% ceny katalogowej.

### **8.2.2 Procedury zwrotu urządzeń objętych gwarancją:**

Inżynier wsparcia technicznego musi zatwierdzić zwrot urządzenia na gwarancji:

- a) Zostanie wysłane urządzenie zastępcze. Konieczne będzie nowe zamówienie.
- b) Po odebraniu produktu przez punkt wskazany przez Prosoft, zostanie wystawiony dokument otrzymania urządzenia.

### **8.2.3 Procedura zwrotu towarów po okresie gwarancyjnym:**

- a) Klient przysyła moduł do oceny
- b) Jeżeli nie zostanie stwierdzona usterka, Klient zostanie obciążony kosztami wynoszącymi 100 USD plus koszty transportu, podatki i cła (o ile dotyczy). Konieczne będzie nowe zamówienie.

- c) Jeżeli urządzenie zostanie naprawione, klient zostanie obciążony kosztami naprawy wynoszącymi 30% aktualnej ceny katalogowej plus koszty transportu, podatki i cła (o ile dotyczy). Konieczne będzie również nowe zamówienie, lub zgoda na wykorzystanie oryginalnego zamówienia w celu oszacowania kosztów.

Poniżej znajduje się lista urządzeń nienaprawialnych:

- 3150 - wszystkie
- 3750
- 3600 - wszystkie
- 3700
- 3170 - wszystkie
- 3250
- 1560 – może zostać naprawiony jedynie gdy wada dotyczy zasilacza
- 1550 - może zostać naprawiony jedynie gdy wada dotyczy zasilacza
- 3350
- 3300
- 1500 - wszystkie

#### **8.2.4 Zakup przedłużenia gwarancji:**

- a) Standardowy okres gwarancji na produkty Prosoft wynosi 3 lata od daty dostarczenia zgodnie z warunkami zawartymi w punkcie „Ograniczona Gwarancja “. Okres gwarancji może zostać wydłużony w chwili zakupu urządzenia za dodatkową opłatą według specyfikacji:
- Dodatkowy 1 rok = 10% ceny katalogowej
  - Dodatkowe 2 lata = 20% ceny katalogowej
  - Dodatkowe 3 lata = 30% ceny katalogowej

### **8.3 Ograniczona Gwarancja**

Ograniczona Gwarancja ("Gwarancja") dotyczy sprzedaży urządzeń, oprogramowania, i innych produktów wyprodukowanych i/lub oferowanych do sprzedaży przez firmę Prosoft jak również wszystkich usług oferowanych przez Prosoft włączając w to utrzymanie, naprawy, wymianę gwarancyjną. Kupując lub korzystając produktu lub usług Klient zgadza się ze wszelkimi warunkami ograniczonej gwarancji. Każde sprzedane oprogramowanie lub inna własność intelektualna, jest przedmiotem licencji towarzyszącej oprogramowaniu lub innej własności intelektualnej.

### **8.3.1 Zakres gwarancji**

- a) *Gwarancja na nowe produkty.* Prosoft gwarantuje kupującemu, że produkt objęty sprzedażą, pracuje zgodnie z opublikowaną specyfikacją, która została przygotowana, zatwierdzona i upubliczniona przez firmę Prosoft, oraz że jest wolny od wad materiałowych oraz w wykonaniu przy założeniu że produkt jest sprzedawany jako nowy. Gwarancja ta upływa w terminie 3 lat od daty sprzedaży. Jeżeli w trakcie trwania gwarancji klient zauważy uszkodzenie produktu, wadę w wykonaniu lub działanie niezgodnie ze specyfikacją, musi On poinformować Prosoft lub dystrybutora telefonicznie, pocztą elektroniczną lub faksem. W żadnym wypadku zgłoszenie nie może być wykonane później niż w terminie 39 miesięcy od daty sprzedaży. W możliwie najkrótszym czasie po zgłoszeniu awarii, Prosoft zobowiązuje się usunąć niezgodność ze specyfikacją czy to poprzez wymianę urządzenia na nowe czy naprawę. Koszty naprawy wliczając w to koszt części zamiennych i koszt pracy pokrywa firma Prosoft. Naprawa zostanie dokonana w centrach serwisowych wskazanych przez firmę Prosoft.
- b) *Gwarancja na usługi.* Elementy oraz naprawa uszkodzonego urządzenia posiadają gwarancję jak nowe urządzenie, przy czym gwarancja zachowuje ważność do końca okresu gwarancyjnego nowego urządzenia, lub, w przypadku gdy okres gwarancji dla urządzenia zakończył się, 90 dni od daty naprawy.

### **8.3.2 Zakres nieobjęty gwarancją**

- c) Prosoft nie gwarantuje ani w sposób jawny ani ukryty, że oprogramowanie zakupione w firmie Prosoft będzie działało bezbłędnie, ani że funkcjonalność oprogramowania spełni oczekiwania, stawiane przez klienta.; Klient przejmuje pełną odpowiedzialność za decyzje podejmowane w oparciu o informacje uzyskane w trakcie korzystania z oprogramowania firmy Prosoft.
- d) Gwarancja nie obejmuje sytuacji w której urządzenie nie wykonuje określonych zadań lub jest uszkodzone wskutek: transportu(I); nieprawidłowego montażu lub nieprzestrzegania przez klienta innych instrukcji dostarczonych przez Prosoft(II); nieautoryzowanej naprawy (III); modyfikacji mechanicznych lub programowych wykonanych za pomocą narzędzi niezatwierdzanych przez firmę Prosoft (m.in. przy pomocy programów napisanych w językach zgodnych z IEC 61131-3 lub jakiegokolwiek wersji języka C)(IV); wykorzystywania produktu niezgodnie z jego przeznaczeniem(V); jakiegokolwiek innego niedopatrzenia ze strony użytkownika (VI), wypadku, nieprawidłowego testowania lub innej przyczyny niezależnej od produktu jak np. działanie wysokich temperatur, wilgotności, awaria zasilania lub zwarcie zasilania(VII); katastrof naturalnych jak powódź, trzęsienie ziemi lub pożar(VIII).

- e) Informacja zawarta w tej umowie może zostać zmieniona bez powiadomienia. Prosoft nie bierze odpowiedzialności za błędy edytorskie, lub braki w dokumentacji jak również za przypadkowe uszkodzenia który były wynikiem korzystania z tej dokumentacji. Instrukcja dostarczona z produktem zawiera informacje objęte prawem autorskim. Żadna część tej instrukcji nie może być rozpowszechniana, kopiowana bez pisemnej zgody firmy Prosoft.

### **8.3.3 Dementi dotyczące aplikacji wysokiego ryzyka.**

Urządzenia produkowane przez firmę Prosoft nie są urządzeniami typu Fault Tolerant i nie są przeznaczone do pracy w niebezpiecznych warunkach wymagających pracy typu fail-safe. Produkt nie jest przeznaczony w szczególności do: Instalacji jądrowych, nawigacji samolotowej, systemów kontroli lotów, urządzeń podtrzymujących życie lub systemów obronnych w których wada produktu mogłaby pośrednio lub bezpośrednio spowodować śmierć, kalectwo, ciężkie obrażenia fizyczne lub zanieczyszczenie środowiska. Prosoft zrzeka się gwarancji na produkt dla aplikacji wysokiego ryzyka.

### **8.3.4 Ochrona własności intelektualnej**

Kupujący zobowiązuje się że nie będzie żądał od firmy Prosoft ani od jej pracowników jakiegokolwiek zadośćuczynienia za straty które klient poniósł wskutek korzystania z produktu. Niezależnie od tego czy straty wynikały z błędu klienta czy z jakiegokolwiek innego powodu dotyczącego urządzenia. Klient zobowiązuje się na własny koszt chronić firmę Prosoft w wypadku gdy produkt wykonany na zamówienie klienta narusza prawa autorskie strony trzeciej. Prosoft nie gwarantuje że dostarczony produkt nie stanie się przedmiotem w sporze o prawa autorskie, naruszenie patentu itp. Kupujący bierze na siebie ryzyko (włączając w to ryzyko pozwu) że produkt lub jakiegokolwiek użycie produktu może naruszać czyjeś patenty, znaki towarowe lub prawa autorskie

- a) Każda dokumentacja dostarczona z produktem jest chroniona prawem autorskim i nie może podlegać rozpowszechnianiu lub kopiowaniu bez pisemnej zgody firmy Prosoft.
- b) Specyfikacja techniczna modułu jak również dostarczona do niego dokumentacja może zostać zmieniona bez informowania użytkownika.
- c) Przeniesienie prawa własności nie daje klientowi żadnych praw produkowania urządzeń dostarczanych przez Prosoft.
- d) Klient nie uzyskuje prawa do użytkowania oprogramowania lub innej własności intelektualnej w sposób inny niż zostało to określone w umowie licencyjnej dotyczącej danego oprogramowania lub innej własności intelektualnej.

- e) Klient zobowiązuje się że nie będzie kopiował ani zlecał kopiowania oprogramowania firmy Prosoft (o ile nie zezwala na to osobna umowa licencyjna towarzysząca oprogramowaniu); przekazywał oprogramowanie bez produktu, modyfikował, zmieniał, tłumaczył, dekodował, dekompilował, ani w żaden inny sposób próbował odczytać kod źródłowy oprogramowania lub urządzenia bazującego na oprogramowaniu. Klient zobowiązuje się również nie eksportować oprogramowania wbrew międzynarodowym zasadom dotyczącym eksportu. Dodatkowo klient zobowiązuje się wykorzystywać oprogramowanie tylko zgodnie z jego przeznaczeniem do współpracy z dedykowanym urządzeniem.

f) **Dodatkowe ograniczenia związane z oprogramowaniem i inną własnością intelektualną**

Oprócz zachowania warunków gwarancji, klient dokonujący zakupu oprogramowania lub innej własności intelektualnej zobowiązuje się przestrzegać umów licencyjnych dostarczanych wraz z oprogramowaniem. Niedotrzymanie tego zobowiązania może spowodować anulowanie gwarancji wraz z gwarancją na inne urządzenia będące własnością intelektualną.

### **8.3.5 Zastąpienie innych gwarancji**

Warunki gwarancji wymienione w punkcie *Zakres gwarancji (strona 54)* zastępują wszelkie inne gwarancje jawne lub ukryte włącznie z gwarancją spełniania przez produkt określonych oczekiwań aplikacyjnych.

### **8.3.6 Ograniczenie odpowiedzialności \*\***

W żadnym wypadku ani Prosoft ani jego dystrybutor nie będą odpowiedzialni za jakiegokolwiek szkody, zerwanie kontraktu, oraz inne zdarzenia spowodowane uszkodzeniem urządzenia na gwarancji. Lista zdarzeń za które ani Prosoft ani jego dystrybutor nie biorą odpowiedzialności obejmuje: utratę zysków, utratę oszczędności, utratę funkcjonalności przez produkt końcowy lub urządzenie, utratę danych, utratę kapitału, straty w wyniku przestoju produkcji, koszt zakupu zamienników, roszczenia firm trzecich – klientów kupującego.

\*\* niektóre obszary nie zezwalają na ograniczenie czasowe gwarancji, lub umożliwiają wykluczenie z warunków odpowiedzialność za uszkodzenia przypadkowe lub będące konsekwencją uszkodzenia urządzenia. Wymienione ograniczenia mogą nie dotyczyć takich sytuacji. Gwarancja daje pewne możliwości prawne. W zależności od miejsca możliwe jest przysługiwanie dodatkowych uprawnień związanych z gwarancją.

### **8.3.7 Ograniczenie czasowe dla zgłoszenia usterki**

Każda akcja związana z naprawą gwarancyjną musi być rozpoczęta maksymalnie w 39 miesięcy od daty sprzedaży.

### **8.3.8 Żadnej dodatkowej gwarancji**

Gwarancja ta jest jedynym i kompletnym porozumieniem stron dotyczącym przedmiotu gwarancji o ile zmiana nie została pisemnie potwierdzona przez obydwie strony. Tym samym wszelkie inne umowy czy to zawarte ustnie czy pisemnie, włączając w to deklaracje osoby sprzedającej, uznaje się za nieważne. Żaden z pracowników firmy Prosoft ani innego podmiotu nie ma prawa dodawać czegokolwiek do zakresu gwarancji. Klient jest zobowiązany zapoznać się z warunkami gwarancji i sprawdzić czy warunki te odpowiadają jego potrzebom..

### **8.3.9 Alokacja ryzyka**

Gwarancja ta lokuje ryzyko uszkodzenia produktu pomiędzy firmą Prosoft a klientem. Jest ona zaakceptowana przez obydwie strony i odzwierciedlona ceną urządzenia. Klient potwierdza że przeczytał tę gwarancję i zgadza się z jej warunkami..