

ANALIZATOR PARAMETRÓW SIECI TYPU ND1



Instrukcja obsługi protokołu transmisji „MODBUS”

1. PRZEZNACZENIE	3
2. OPIS PROTOKOŁU MODBUS	3
2.1 RAMKA W TRYBIE ASCII	4
2.2 RAMKA W TRYBIE RTU	4
2.3 CHARAKTERYSTYKA PÓL RAMKI	4
2.4 WYZNACZENIE LRC	5
2.5 WYZNACZENIE CRC	5
2.6 FORMAT ZNAKU PRZY TRANSMISJI SZEREGOWEJ	5
2.7 PRZERWANIE TRANSAKCJI	5
3. OPIS FUNKCJI.....	6
3.1 ODCZYT N-REJESTRÓW (KOD 03).....	6
3.2 ZAPIS WARTOŚCI DO REJESTRU (KOD 06).....	6
3.3 ZAPIS DO N-REJESTRÓW (KOD 16)	7
3.4 RAPORT IDENTYFIKUJĄCY URZĄDZENIE (KOD 17)	7
4. KODY BŁĘDÓW	7
5. TABLICA REJESTRÓW	9
DODATEK A. OBLICZANIE SUMY KONTROLNEJ	14

1. PRZEZNACZENIE

Aby uzyskać wymianę informacji, przy wykorzystaniu łącza szeregowego, należy wybrać typ interfejsu i ustalić sposób interpretacji przesyłanych danych. Typ interfejsu definiuje jedynie parametry elektryczne transmisji i sposób łączenia urządzeń. Od interpretacji danych zależą takie cechy jak możliwość obsługi wielu urządzeń, sprawdzanie poprawności transmisji oraz zasady dostępu do urządzenia. Zadaniem protokołu jest określenie jaki typ danych jest interpretowany (dozwolony) i w jaki sposób są one interpretowane.

Na łączu szeregowym analizatora ND1 został zaimplementowany asynchroniczny znakowy protokół komunikacyjny MODBUS. Konfiguracja parametrów łącza szeregowego RS 485 została opisana w instrukcji obsługi analizatora ND1.

Zestawienie parametrów łącza szeregowego analizatora ND1:

- adres analizatora - 1..247
- prędkość transmisji -300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200 bit/s,
- tryby pracy - ASCII, RTU,
- jednostka informacyjna - ASCII: 8N1, 7N2, 7E1, 7O1; RTU: 8N2, 8N1, 8E1, 8O1,
- maksymalny czas odpowiedzi - 100 ms.

2. OPIS PROTOKOŁU MODBUS

Protokół MODBUS jest standardem przyjętym przez producentów sterowników przemysłowych dla asynchronicznej, znakowej wymiany informacji pomiędzy urządzeniami systemów pomiarowo kontrolnych. Posiada on takie cechy jak:

- prosta reguła dostępu do łącza oparta na zasadzie "master-slave",
- zabezpieczenie przesyłanych komunikatów przed błędami,
- potwierdzenie wykonywania rozkazów zdalnych i sygnalizacja błędów,
- skuteczne mechanizmy zabezpieczające przed zawieszeniem systemu,
- wykorzystanie asynchronicznej transmisji znakowej.

Kontrolery urządzeń pracujących w systemie **MODBUS** komunikują się ze sobą przy wykorzystaniu protokołu typu **master-slave**, w którym tylko jedno urządzenie może inicjalizować transakcje (jednostka nadrzędna-**master**), a pozostałe (jednostki podrzędne-**slave**) odpowiadają jedynie na zdalne zapytania jednostki nadrzędnej.

Transakcja składa się z polecenia wysyłanego z jednostki **master** do **slave** oraz z odpowiedzi przesyłanej w odwrotnym kierunku. Odpowiedź zawiera dane żądane przez **master** lub potwierdzenie realizacji jego polecenia.

Master może przysyłać informację do pojedynczych odbiorców lub informacje rozgłoszeniowe (broadcast), przeznaczone dla wszystkich urządzeń podrzędnych w systemie (na polecenia rozgłoszeniowe master nie otrzymuje odpowiedzi).

Format przesyłanych informacji jest następujący:

- **master⇒slave** : adres odbiorcy, kod reprezentujący żądane polecenie, dane, słowo kontrolne zabezpieczające przesyłaną wiadomość,
- **slave⇒master** : adres nadawcy, potwierdzenie realizacji rozkazu, dane żądane przez master, słowo kontrolne zabezpieczające odpowiedź przed błędami.

Jeżeli urządzenie **slave** wykryje błąd przy odbiorze wiadomości, lub nie może wykonać polecenia, przygotowuje specjalny komunikat o wystąpieniu błędu i przysyła go jako odpowiedź do **mastera**.

Urządzenia pracujące w protokole **MODBUS** mogą być ustawione na komunikację przy użyciu jednego z dwóch trybów transmisji: **ASCII** lub **RTU**. Użytkownik wybiera żądany tryb wraz z parametrami portu szeregowego (prędkość transmisji, jednostka informacyjna), podczas konfiguracji każdego urządzenia.

W systemie **MODBUS** przesyłane wiadomości są zorganizowane w ramki o określonym początku i końcu. Pozwala to urządzeniu odbierającemu na odrzucenie ramek niekompletnych i sygnalizację związanych z tym błędów.

Ze względu na możliwość pracy w jednym z dwóch różnych trybów transmisji (**ASCII** lub **RTU**), definiuje się dwie ramki.

2.1 Ramka w trybie ASCII

W trybie ASCII każdy bajt wiadomości przesyłany jest w postaci dwóch znaków ASCII. Podstawową zaletą tego trybu jest to, iż pozwala on na długie odstępy między znakami (do 1s) bez powodowania błędów.

Format ramki przedstawiono poniżej:

Znacznik początku	Adres	Funkcja	Dane	Kontrola LRC	Znacznik końca
1 znak ":'	2 znaki	2 znaki	n znaków	2 znaki	2 znaki CR LF

Znacznikiem początku jest znak dwukropka (":"- ASCII 3Ah), natomiast znacznikiem końca dwa znaki CR i LF. Część informacyjną ramki zabezpiecza się kodem LRC (Longitudinal Redundancy Check).

2.2 Ramka w trybie RTU

W trybie RTU wiadomości rozpoczynają i kończą się odstępem trwającym minimum 3.5 x (czas trwania pojedynczego znaku), w którym panuje cisza na łączu. Najprostszą implementacją wymienionego interwału czasowego jest wielokrotne odmierzenie czasu trwania znaku przy zadanej szybkości bodowej przyjętej na łączu.

Format ramki przedstawiono poniżej:

Znacznik początku	Adres	Funkcja	Dane	Kontrola CRC	Znacznik końca
T1-T2-T3-T4	8 bitów	8 bitów	n x 8 bitów	16 bitów	T1-T2-T3-T4

Znaczniki początku i końca zaznaczono symbolicznie jako odstęp równy czterem długościom znaku (jednostki informacyjnej). Słowo kontrolne jest 16 bitowe i powstaje jako rezultat obliczenia CRC (Cyclical Redundancy Check) na zawartości ramki.

2.3 Charakterystyka pól ramki

Pole adresowe

Pole adresowe w ramce zawiera dwa znaki (w trybie ASCII) lub osiem bitów (w trybie RTU). Zakres adresów jednostek slave wynosi 0 - 247. Master adresuje jednostki slave umieszczając jej adres na polu adresowym ramki. Kiedy jednostka slave wysyła odpowiedź, umieszcza swój własny adres na polu adresowym ramki, co pozwala masterowi sprawdzić, z którą jednostką realizowana jest transakcja.

Adres 0 jest wykorzystywany jako adres rozgłoszeniowy, rozpoznawany przez wszystkie jednostki slave podłączone do magistrali.

Pole funkcji

Pole funkcji zawiera dwa znaki w trybie ASCII lub 8-bitów w trybie RTU. Zakres kodów funkcji od 1 - 255.

Przy transmisji polecenia z jednostki master do slave, pole funkcji zawiera kod rozkazu, określający działanie, które ma podjąć jednostka slave na żądanie mastera.

Kiedy jednostka slave odpowiada masterowi, pole funkcji wykorzystuje do potwierdzenia wykonania polecenia lub sygnalizacji błędu, jeżeli z jakichś przyczyn nie może wykonać polecenia. Potwierdzenie pozytywne realizowane jest poprzez umieszczenie na polu funkcji kodu wykonanego rozkazu.

W przypadku stwierdzenia błędu, jednostka slave umieszcza na polu funkcji szczególną odpowiedź, którą stanowi kod funkcji z ustawionym na 1 najstarszym bitem.

Kod błędu umieszczany jest na polu danych ramki odpowiedzi

Pole danych

Pole danych tworzy zestaw dwucyfrowych liczb heksadecymalnych, o zakresie 00-FF. Liczby te przy transmisji w trybie ASCII reprezentowane są dwoma znakami, a przy transmisji w trybie RTU jednym. Pole danych ramki polecenia zawiera dodatkowe informacje potrzebne jednostce slave do wykonania rozkazu określonego kodem funkcji. Mogą to być adresy rejestrów, liczba bajtów w polu danych, dane itp.

W niektórych ramach pole danych może posiadać zerową długość. Tak jest zawsze, gdy operacja określona kodem nie wymaga żadnych parametrów.

Pole kontrolne

W protokole MODBUS słowo kontrolne zabezpieczające część informacyjną zależy od zastosowanego trybu transmisji.

W trybie ASCII pole kontrolne składa się z dwóch znaków ASCII, które są rezultatem obliczenia Longitudinal Redundancy Check (LRC) na zawartości części informacyjnej ramki (bez znaczników początku i końca). Znaki LRC są dołączane do wiadomości jako ostatnie pole ramki, bezpośrednio przed znacznikiem końca (CR,LF).

W trybie RTU słowo kontrolne jest 16-bitowe i powstaje jako rezultat obliczenia Cyclical Redundancy Check (CRC) na zawartości ramki. Pole kontrolne zajmuje dwa bajty dołączane na końcu ramki. Jako pierwszy przesyłany jest mniej znaczący bajt, jako ostatni starszy bajt, który jest jednocześnie znakiem kończącym ramkę.

2.4 Wyznaczenie LRC

Obliczanie LRC polega na sumowaniu kolejnych 8-bitowych bajtów wiadomości, odrzuceniu przeniesień i na koniec wyznaczeniu uzupełnienia dwójkowego wyniku. Sumowanie obejmuje całą wiadomość za wyjątkiem znaczników początku i końca ramki. Wartość 8-bitowa sumy LRC jest umieszczana na końcu ramki w postaci dwóch znaków ASCII, najpierw znak zawierający starszą tetradę, a za nim znak zawierający młodszą tetradę LRC.

2.5 Wyznaczenie CRC

Obliczanie CRC realizowane jest według następującego algorytmu:

1. Załadowanie FFFFh do 16-bitowego rejestru CRC.
2. Pobranie bajtu z bloku danych i wykonanie operacji EXOR z młodszym bajtem rejestru CRC. Umieszczenie rezultatu w rejestrze CRC.
3. Przesunięcie zawartości rejestru CRC w prawo o jeden bit połączone z wpisaniem 0 na najbardziej znaczący bit (MSB=0).
4. Sprawdzenie stanu najmłodszego bitu (LSB) wysuniętego z rejestru CRC w poprzednim kroku. Jeżeli jego stan równa się 0, to następuje powrót do kroku 3 (kolejne przesunięcie), jeżeli 1, to wykonywana jest operacja EXOR rejestru CRC ze stałą A001h.
5. Powtórzenie kroków 3 i 4 osiem razy, co odpowiada przetworzeniu całego bajtu.
6. Powtórzenie sekwencji 2,3,4,5 dla kolejnego bajtu wiadomości. Kontynuacja tego procesu aż do przetworzenia wszystkich bajtów wiadomości.
7. Zawartość CRC po wykonaniu wymienionych operacji jest poszukiwaną wartością CRC.
8. Wartość CRC jest umieszczana na końcu ramki najpierw mniej znaczący bajt, a za nim bardziej znaczący bajt.

2.6 Format znaku przy transmisji szeregowej

W protokole MODBUS znaki są przesyłane od najmłodszego do najstarszego bitu.

Organizacja jednostki informacyjnej w trybie ASCII:

- 1 bit startu,
- 7 bitów pola danych,
- 1 bit kontroli parzystości (nieparzystości) lub brak bitu kontroli parzystości,
- 1 bit stopu przy kontroli parzystości lub 2 bity stopu przy braku kontroli parzystości

Organizacja jednostki informacyjnej w trybie RTU:

- 1 bit startu,
- 8 bitów pola danych,
- 1 bit kontroli parzystości (nieparzystości) lub brak bitu kontroli parzystości,
- 1 bit stopu przy kontroli parzystości lub 2 bity stopu przy braku kontroli parzystości.

2.7 Przerwanie transakcji

W jednostce master użytkownik ustawia ważny parametr jakim jest "maksymalny czas odpowiedzi na ramkę zapytania", po którego przekroczeniu transakcja jest przerywana. Czas ten dobiera się tak, aby każda jednostka slave pracująca w systemie (nawet ta najwolniejsza) zdążyła normalnie odpowiedzieć na ramkę zapytania. Przekroczenie tego czasu świadczy zatem o błędzie i tak jest traktowane przez jednostkę master.

Jeżeli jednostka slave wykryje błąd transmisji, nie wykonuje polecenia oraz nie wysyła żadnej odpowiedzi. Spowoduje to przekroczenie czasu oczekiwania na ramkę odpowiedzi i przerwanie transakcji.

3. OPIS FUNKCJI

W analizatorze ND1 zaimplementowane zostały następujące funkcje protokołu:

Kod	znaczenie
03	odczyt n-rejestrów
17	identyfikacja urządzenia slave

3.1 Odczyt n-rejestrów (kod 03)

Żądanie:

Funkcja umożliwi odczyt wartości zawartych w rejestrach w zaadresowanym urządzeniu slave. **Rejestry są 16 lub 32-bitowymi jednostkami, które mogą zawierać wartości numeryczne związane ze zmiennymi procesowymi itp.**

Ramka żądania określa 16-bitowy adres początkowy rejestru oraz liczbę rejestrów do odczytania. Znaczenie zawartości rejestrów o danych adresach może być różne dla różnych typów urządzeń.

Funkcja nie jest dostępna w trybie rozgłoszeniowym.

Przykład. Odczyt 3 rejestrów zaczynając od rejestru o adresie 6Bh

adres	funkcja	adres rejestru Hi	adres rejestru Lo	liczba rejestrów Hi	liczba rejestrów Lo	suma kontrolna
11	03	00	6B	00	03	7E

LRC

Odpowiedź:

Dane rejestrów są pakowane począwszy od najmniejszego adresu: najpierw starszy bajt, potem młodszy bajt rejestru.

Przykład. Ramka odpowiedzi

adres	funkcja	liczba bajtów	wart. w rej.107 Hi	wart. w rej.107 Lo	wart. w rej.108 Hi	wart. w rej.108 Lo	wart. w rej.109 Hi	wart. w rej.109 Lo	suma kontrolna
11	03	06	02	2B	00	00	00	64	55

LRC

3.2 Zapis wartości do rejestru (kod 06)

Żądanie

Funkcja umożliwia modyfikację zawartości rejestru. Jest dostępna w trybie rozgłoszeniowym.

Przykład.

adres	funkcja	adres rejestru Hi	adres rejestru Lo	wartość Hi	wartość Lo	suma kontrolna
11	06	00	87	03	9E	C1

LRC

Odpowiedź:

Prawidłową odpowiedzią na żądanie zapisu wartości do rejestru jest retransmisja komunikatu po wykonaniu operacji.

Przykład.

adres	funkcja	adres rejestru Hi	adres rejestru Lo	wartość Hi	wartość Lo	suma kontrolna
11	06	00	87	03	9E	C1

LRC

3.3 Zapis do n-rejestrów (kod 16)

Żądanie:

Funkcja dostępna w trybie rozgłoszeniowym. Umożliwia modyfikacje zawartości rejestrów.

Przykład. Zapis dwóch rejestrów począwszy od rejestru o adresie 136

adres	funkcja	adres rej. Hi	adres rej. Lo	liczba rej. Hi	liczba rej. Lo	liczba bajtów	dane Hi	dane Lo	dane Hi	dane Lo	suma kontrolna
11	10	00	87	00	02	04	00	0A	01	02	45

 LRC

Odpowiedź:

Prawidłowa odpowiedź zawiera adres jednostki slave, kod funkcji, adres początkowy oraz liczbę zapisanych rejestrów.

Przykład.

adres	funkcja	adres rej. Hi	adres rej. Lo	liczba rej. Hi	liczba rej. Lo	suma kontrolna
11	10	00	87	00	02	56

 LRC

3.4 Raport identyfikujący urządzenie (kod 17)

Żądanie:

Funkcja pozwala użytkownikowi uzyskać informacje o typie urządzenia, statusie i zależnej od tego konfiguracji.

Przykład.

Adres	funkcja	suma kontrolna	
11	11	CD	EC

Odpowiedź:

Pole „identyfikator urządzenia” w ramce odpowiedzi oznacza unikalny identyfikator danej klasy urządzeń, natomiast pozostałe pola zawierają parametry zależne od typu urządzenia.

Przykład dla analizatora ND1.

Adres slave	funkcja	liczba bajtów	identyfikator urządzenia	stan urządzenia	suma kontrolna	
11	11	02	BD	FF	4D	EF

4. KODY BŁĘDÓW

Gdy urządzenie master wysła żądanie do urządzenia slave, to za wyjątkiem komunikatów w trybie rozgłoszeniowym, oczekuje prawidłowej odpowiedzi. Po wysłaniu żądania jednostki master może wystąpić jedno z czterech możliwych zdarzeń:

- Jeżeli jednostka slave odbiera żądanie bez błędu transmisji oraz może je wykonać prawidłowo, wówczas zwraca prawidłową odpowiedź.
- Jeżeli jednostka slave nie odbiera żądania, żadna odpowiedź nie jest zwracana. W programie urządzenia master zostaną spełnione warunki timeout dla żądania.
- Jeżeli jednostka slave odbiera żądanie, ale z błędami transmisji (błąd parzystości, sumy kontrolnej LRC lub CRC), żadna odpowiedź nie jest zwracana. W programie urządzenia master zostaną spełnione warunki timeout dla żądania.
- Jeżeli jednostka slave odbiera żądanie bez błędu transmisji, ale nie może go wykonać prawidłowo (np. jeżeli żądaniem jest odczyt nie istniejącego wyjścia bitowego lub rejestru), wówczas zwraca odpowiedź zawierającą kod błędu, informujący urządzenie master o przyczynie błędu.

Komunikat z błędną odpowiedzią zawiera dwa pola odróżniające go od prawidłowej odpowiedzi:

1. Pole kodu funkcji:

W prawidłowej odpowiedzi, jednostka slave retransmituje kod funkcji z komunikatu żądania na polu kodu funkcji odpowiedzi. Wszystkie kody funkcji mają najbardziej znaczący bit (MSB) równy 0 (wartości kodów są poniżej 80h). W błędnej odpowiedzi urządzenie slave ustawia bit MSB kodu funkcji na 1. To powoduje, że wartość kodu funkcji w błędnej odpowiedzi jest dokładnie o 80h większa niż byłaby w prawidłowej odpowiedzi.

Na podstawie kodu funkcji z ustawionym bitem MSB program urządzenia master może rozpoznać błędną odpowiedź i może sprawdzić na polu danych kod błędu.

2. Pole danych:

W prawidłowej odpowiedzi, urządzenie slave może zwrócić dane na polu danych (pewne informacje żądane przez jednostkę master). W błędnej odpowiedzi, urządzenie slave zwraca kod błędu na polu danych. Określa on warunki urządzenia slave, które spowodowały błąd.

Poniżej przedstawiono przykład żądania urządzenia master i błędną odpowiedź urządzenia slave. Dane są w postaci heksadecymalnej.

Przykład: żądanie

adres slave	funkcja	adres zmiennej Hi	adres zmiennej Lo	liczba zmiennych Hi	liczba zmiennych Lo	suma kontrolna
0A	01	04	A1	00	01	4F

 LRC

Przykład: błędna odpowiedź

adres slave	funkcja	kod błędu	suma kontrolna
0A	81	02	73

 LRC

W tym przykładzie urządzenie master adresuje żądanie do jednostki slave o numerze 10 (0Ah). Kod funkcji (01) służy do operacji odczytu stanu wyjścia bitowego. Ta ramka oznacza więc żądanie odczytu statusu jednego wyjścia bitowego o adresie 1245 (04A1h).

Jeżeli w urządzeniu slave nie ma wyjścia bitowego o podanym adresie, wówczas urządzenie zwróci błędną odpowiedź z kodem błędu nr 02. Oznacza on niedozwolony adres danych w urządzeniu slave.

W poniższej tabeli przedstawione są możliwe kody błędów i ich znaczenie

Kod	Znaczenie
01	niedozwolona funkcja
02	niedozwolony adres danych
03	niedozwolona wartość danej
04	uszkodzenie w przyłączonym urządzeniu
05	potwierdzenie
06	zajęty, komunikat usunięty
07	negatywne potwierdzenie
08	błąd parzystości pamięci

5. Tablica rejestrów

- Identyfikator analizatora ND1 (wysyłany w odpowiedzi na funkcję identyfikacji) : 0xBD
- Typy rejestrów:
 - float – liczba zmiennoprzecinkowa (patrz opis poniżej),
 - sfloat – liczba zmiennoprzecinkowa (patrz opis poniżej),
 - double – liczba zmiennoprzecinkowa podwójnej precyzji (patrz opis poniżej),
 - sdouble – liczba zmiennoprzecinkowa podwójnej precyzji (patrz opis poniżej),
 - long – liczba stałoprzecinkowa 32 bitowa (patrz opis poniżej)
 - slong – liczba stałoprzecinkowa 32 bitowa (patrz opis poniżej)
- Tryby dostępu do rejestrów:
 - RO – tylko do odczytu,
- Reprezentacja liczb stało przecinkowych 32 bitowych (long)

StarszeSłowo	MłodszeSłowo
--------------	--------------

- Reprezentacja liczb zmiennoprzecinkowych (float IEEE 754)

bajt 4	bajt 3	bajt 2	bajt 1
SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM

- Reprezentacja liczb zmiennoprzecinkowych podwójnej precyzji (double IEEE 754)

bajt 8	bajt 7	bajt 6	bajt 5	bajt 4	bajt 3	bajt 2	bajt 1
SEEEEEEE	EEEEEMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM

S – bit znaku (Sign bit)
 E – wykładnik (Exponent)
 M – mantysa

Bajty rejestrów typu **long** przesyłane są w kolejności StarszeSłowo MłodszeSłowo
 Bajty rejestrów typu **slong** przesyłane są w kolejności MłodszeSłowo StarszeSłowo

Bajty rejestrów typu **float** przesyłane są w kolejności 4321
 Bajty rejestrów typu **sfloat** przesyłane są w kolejności 2143

Bajty rejestrów typu **double** przesyłane są w kolejności 87654321
 Bajty rejestrów typu **sdouble** przesyłane są w kolejności 21436587

Tabela 1 Tablica rejestrów ND1

Adres	Typ	Dostęp	Opis
2000	int	RO	Alarmy - kolejne bity to stan kolejnych alarmów
			Rejestry typu float adresowane 16 bitowo
4000..4237	float	RO	Parametry sieci (patrz Tabela 2)
4300..4347	float	RO	Wejścia Modbus Master
4500..4523	float	RO	Wejścia Binarne
4550..4573	float	RO	Alarmy
4600..4639	float	RO	Liczniki energii w kWh (patrz Tabela 3)
4700..4739	float	RO	Liczniki energii w 100 Wh (patrz Tabela 4)
10000..10357	float	RO	Harmoniczne napięcie
10000..10101	float	RO	Harmoniczne napięcie L1
10128..10229	float	RO	Harmoniczne napięcie L2
10256..10357	float	RO	Harmoniczne napięcie L3
10400..10757	float	RO	Harmoniczne prądów
10400..10501	float	RO	Harmoniczne prądów L1
10528..10629	float	RO	Harmoniczne prądów L2
10656..10757	float	RO	Harmoniczne prądów L3
			Rejestry typu sfloat adresowane 16 bitowo

5000..5237	sfloat	RO	Patrz rejestry 4000..4237
5300..5347	sfloat	RO	Patrz rejestry 4300..4347
5500..5523	sfloat	RO	Patrz rejestry 4500..4523
5550..5573	float	RO	Patrz rejestry 4550..4573
5600..5639	sfloat	RO	Patrz rejestry 4600..4639
5700..5739	sfloat	RO	Patrz rejestry 4700..4739
15000..15357	sfloat	RO	Patrz rejestry 10000..10357
15400..15757	sfloat	RO	Patrz rejestry 10400..10757
			Rejestry typu float adresowane 32 bitowo
7000..7118	float	RO	Patrz rejestry 4000..4237
7200..7223	float	RO	Patrz rejestry 4300..4347
7300..7311	float	RO	Patrz rejestry 4500..4523
7350..7361	float	RO	Patrz rejestry 4550..4573
7400..7419	float	RO	Patrz rejestry 4600..4639
7450..7469	float	RO	Patrz rejestry 4700..4739
20000..20178	float	RO	Harmoniczne napięć (patrz rejestry 10000..10357)
20000..20050	float	RO	Harmoniczne napięć L1
20064..20114	float	RO	Harmoniczne napięć L2
20128..20178	float	RO	Harmoniczne napięć L3
20200..20378	float	RO	Harmoniczne prądów (patrz rejestry 10400..10757)
20200..20250	float	RO	Harmoniczne prądów L1
20264..20314	float	RO	Harmoniczne prądów L2
20328..20378	float	RO	Harmoniczne prądów L3
			Rejestry typu sfloat adresowane 32 bitowo
8000..8118	sfloat	RO	Patrz rejestry 4000..4237
8200..8223	sfloat	RO	Patrz rejestry 4300..4347
8300..8311	sfloat	RO	Patrz rejestry 4500..4523
8350..8361	sfloat	RO	Patrz rejestry 4550..4573
8400..8419	sfloat	RO	Patrz rejestry 4600..4639
8450..8469	sfloat	RO	Patrz rejestry 4700..4739
25000..25178	sfloat	RO	Patrz rejestry 20000..20178
25200..25378	sfloat	RO	Patrz rejestry 20200..20378
			Rejestry liczników energii typu double, sdouble, long i slong
6000..6039	double	RO	Liczniki energii w kWh (patrz Tabela 3)
6100..6139	sdouble	RO	Liczniki energii w kWh (patrz Tabela 3)
6200..6219	long	RO	Liczniki energii w kWh (patrz Tabela 3)
6300..6339	long	RO	Liczniki energii w 100 Wh (patrz Tabela 4)
6400..6419	slong	RO	Liczniki energii w kWh (patrz Tabela 3)
6500..6539	slong	RO	Liczniki energii w 100 Wh (patrz Tabela 4)

Tabela 2 Rejestry parametrów sieci

Indeks	Rejestry adresowane 16 bitowo		Rejestry adresowane 32 bitowo		Opis
	float	sfloat	float	sfloat	
0	4000	5000	7000	8000	Urms L1
1	4002	5002	7001	8001	Urms L2
2	4004	5004	7002	8002	Urms L3
3	4006	5006	7003	8003	U L1-2
4	4008	5008	7004	8004	U L2-3
5	4010	5010	7005	8005	U L3-1
6	4012	5012	7006	8006	Upeak- L1
7	4014	5014	7007	8007	Upeak- L2
8	4016	5016	7008	8008	Upeak- L3
9	4018	5018	7009	8009	Upeak+ L1
10	4020	5020	7010	8010	Upeak+ L2
11	4022	5022	7011	8011	Upeak+ L3
12	4024	5024	7012	8012	Ucf L1
13	4026	5026	7013	8013	Ucf L2
14	4028	5028	7014	8014	Ucf L3
15	4030	5030	7015	8015	Irms L1

16	4032	5032	7016	8016	Irms L2
17	4034	5034	7017	8017	Irms L3
18	4036	5036	7018	8018	Icf L1
19	4038	5038	7019	8019	Icf L2
20	4040	5040	7020	8020	Icf L3
21	4042	5042	7021	8021	INC
22	4044	5044	7022	8022	INM
23	4046	5046	7023	8023	Reserved
24	4048	5048	7024	8024	IPeak- L1
25	4050	5050	7025	8025	IPeak- L2
26	4052	5052	7026	8026	IPeak- L3
27	4054	5054	7027	8027	IPeak+ L1
28	4056	5056	7028	8028	IPeak+ L2
29	4058	5058	7029	8029	IPeak+ L3
30	4060	5060	7030	8030	φ U-I L1 [rad]
31	4062	5062	7031	8031	φ U-I L2 [rad]
32	4064	5064	7032	8032	φ U-I L3 [rad]
33	4066	5066	7033	8033	φ U L1-2 [rad]
34	4068	5068	7034	8034	φ U L2-3 [rad]
35	4070	5070	7035	8035	φ U L3-1 [rad]
36	4072	5072	7036	8036	φ I L1-2 [rad]
37	4074	5074	7037	8037	φ I L2-3 [rad]
38	4076	5076	7038	8038	φ I L3-1 [rad]
39	4078	5078	7039	8039	φ U L1 [rad]
40	4080	5080	7040	8040	φ U L2 [rad]
41	4082	5082	7041	8041	φ U L3 [rad]
42	4084	5084	7042	8042	φ I L1 [rad]
43	4086	5086	7043	8043	φ I L2 [rad]
44	4088	5088	7044	8044	φ I L3 [rad]
45	4090	5090	7045	8045	φ U-I L1 [°]
46	4092	5092	7046	8046	φ U-I L2 [°]
47	4094	5094	7047	8047	φ U-I L3 [°]
48	4096	5096	7048	8048	φ U L1-2 [°]
49	4098	5098	7049	8049	φ U L2-3 [°]
50	4100	5100	7050	8050	φ U L3-1 [°]
51	4102	5102	7051	8051	φ I L1-2 [°]
52	4104	5104	7052	8052	φ I L2-3 [°]
53	4106	5106	7053	8053	φ I L3-1 [°]
54	4108	5108	7054	8054	φ U L1 [°]
55	4110	5110	7055	8055	φ U L2 [°]
56	4112	5112	7056	8056	φ U L3 [°]
57	4114	5114	7057	8057	φ I L1 [°]
58	4116	5116	7058	8058	φ I L2 [°]
59	4118	5118	7059	8059	φ I L3 [°]
60	4120	5120	7060	8060	P L1
61	4122	5122	7061	8061	P L2
62	4124	5124	7062	8062	P L3
63	4126	5126	7063	8063	S L1
64	4128	5128	7064	8064	S L2
65	4130	5130	7065	8065	S L3
66	4132	5132	7066	8066	Q L1
67	4134	5134	7067	8067	Q L2
68	4136	5136	7068	8068	Q L3
69	4138	5138	7069	8069	Tg φ L1
70	4140	5140	7070	8070	Tg φ L2
71	4142	5142	7071	8071	Tg φ L3
72	4144	5144	7072	8072	PF L1
73	4146	5146	7073	8073	PF L2
74	4148	5148	7074	8074	PF L3
75	4150	5150	7075	8075	Reserved
76	4152	5152	7076	8076	Reserved

77	4154	5154	7077	8077	Reserved
78	4156	5156	7078	8078	Reserved
79	4158	5158	7079	8079	Reserved
80	4160	5160	7080	8080	Reserved
81	4162	5162	7081	8081	Pst(l min)
82	4164	5164	7082	8082	Pst
83	4166	5166	7083	8083	Plt
84	4168	5168	7084	8084	Uavg
85	4170	5170	7085	8085	Iavg
86	4172	5172	7086	8086	Reserved
87	4174	5174	7087	8087	Reserved
88	4176	5176	7088	8088	Reserved
89	4178	5178	7089	8089	Reserved
90	4180	5180	7090	8090	ΣP
91	4182	5182	7091	8091	ΣS
92	4184	5184	7092	8092	ΣQ
93	4186	5186	7093	8093	Tg ϕ av
94	4188	5188	7094	8094	PFav
95	4190	5190	7095	8095	Uunb
96	4192	5192	7096	8096	Reserved
97	4194	5194	7097	8097	Reserved
98	4196	5196	7098	8098	f
99	4198	5198	7099	8099	Pav
100	4200	5200	7100	8100	Uav
101	4202	5202	7101	8101	UavQTotal
102	4204	5204	7102	8102	UavQPart
103	4206	5206	7103	8103	fav
104	4208	5208	7104	8104	favQTotal
105	4210	5210	7105	8105	favQPart
106	4212	5212	7106	8106	Reserved
107	4214	5214	7107	8107	Reserved
108	4216	5216	7108	8108	Reserved
109	4218	5218	7109	8109	Reserved
110	4220	5220	7110	8110	Reserved
111	4222	5222	7111	8111	Reserved
112	4224	5224	7112	8112	Reserved
113	4226	5226	7113	8113	THD U L1
114	4228	5228	7114	8114	THD U L2
115	4230	5230	7115	8115	THD U L3
116	4232	5232	7116	8116	THD I L1
117	4234	5234	7117	8117	THD I L2
118	4236	5236	7118	8118	THD I L3

Tabela 3 Rejestry liczników energii w kWh i kvarh

Indeks	Rejestry adresowane 16 bitowo						Rejestry adresowane 32 bitowo		Opis
	long	slong	double	sdouble	float	sfloat	float	sfloat	
0	-	-	-	-	4600	5600	7400	8400	EnP [100 MWh]
1	6200	6400	6000	6100	4602	5602	7401	8401	EnP [kWh]
2	-	-	-	-	4604	5604	7402	8402	EnP Taryfa 1 [100 MWh]
3	6202	6402	6004	6104	4606	5606	7403	8403	EnP Taryfa 1 [kWh]
4	-	-	-	-	4608	5608	7404	8404	EnP Taryfa 2 [100 MWh]
5	6204	6404	6008	6108	4610	5610	7405	8405	EnP Taryfa 2 [kWh]
6	-	-	-	-	4612	5612	7406	8406	EnP Taryfa 3 [100 MWh]
7	6206	6406	6012	6112	4614	5614	7407	8407	EnP Taryfa 3 [kWh]
8	-	-	-	-	4616	5616	7408	8408	EnP Taryfa 4 [100 MWh]
9	6208	6408	6016	6116	4618	5618	7409	8409	EnP Taryfa 4 [kWh]
10	-	-	-	-	4620	5620	7410	8410	EnQ [100 Mvarh]
11	6210	6410	6020	6120	4622	5622	7411	8411	EnQ [kvarh]
12	-	-	-	-	4624	5624	7412	8412	EnQ Taryfa 1 [100 Mvarh]
13	6212	6412	6024	6124	4626	5626	7413	8413	EnQ Taryfa 1 [kvarh]
14	-	-	-	-	4628	5628	7414	8414	EnQ Taryfa 2 [100 Mvarh]
15	6214	6414	6028	6128	4630	5630	7415	8415	EnQ Taryfa 2 [kvarh]
16	-	-	-	-	4632	5632	7416	8416	EnQ Taryfa 3 [100 Mvarh]
17	6216	6416	6032	6132	4634	5634	7417	8417	EnQ Taryfa 3 [kvarh]
18	-	-	-	-	4636	5636	7418	8418	EnQ Taryfa 4 [100 Mvarh]
19	6218	6418	6036	6136	4638	5638	7419	8419	EnQ Taryfa 4 [kvarh]

Wartości rejestrów energii czynnej typu float i sfloat w kWh są podawane w zakresie do 999999 [kWh], wartości powyżej 999999 [kWh] są podawane w rejestrach opisanych jako 100 MWh (to samo dotyczy rejestrów energii biernej).

Tabela 4 Rejestry liczników energii w 100 Wh i 100 varh

Indeks	Rejestry adresowane 16 bitowo				Rejestry adresowane 32 bitowo		Opis
	float	sfloat	long	slong	float	sfloat	
0	4700	5700	6300	6500	7450	8450	EnP [10 MWh]
1	4702	5702	6302	6502	7451	8451	EnP [100 Wh]
2	4704	5704	6304	6504	7452	8452	EnP Taryfa 1 [10 MWh]
3	4706	5706	6306	6506	7453	8453	EnP Taryfa 1 [100 Wh]
4	4708	5708	6308	6508	7454	8454	EnP Taryfa 2 [10 MWh]
5	4710	5710	6310	6510	7455	8455	EnP Taryfa 2 [100 Wh]
6	4712	5712	6312	6512	7456	8456	EnP Taryfa 3 [10 MWh]
7	4714	5714	6314	6514	7457	8457	EnP Taryfa 3 [100 Wh]
8	4716	5716	6316	6516	7458	8458	EnP Taryfa 4 [10 MWh]
9	4718	5718	6318	6518	7459	8459	EnP Taryfa 4 [100 Wh]
10	4720	5720	6320	6520	7460	8460	EnQ [10 Mvarh]
11	4722	5722	6322	6522	7461	8461	EnQ [100 varh]
12	4724	5724	6324	6524	7462	8462	EnQ Taryfa 1 [10 Mvarh]
13	4726	5726	6326	6526	7463	8463	EnQ Taryfa 1 [100 varh]
14	4728	5728	6328	6528	7464	8464	EnQ Taryfa 2 [10 Mvarh]
15	4730	5730	6330	6530	7465	8465	EnQ Taryfa 2 [100 varh]
16	4732	5732	6332	6532	7466	8466	EnQ Taryfa 3 [10 Mvarh]
17	4734	5734	6334	6534	7467	8467	EnQ Taryfa 3 [100 varh]
18	4736	5736	6336	6536	7468	8468	EnQ Taryfa 4 [10 Mvarh]
19	4738	5738	6338	6538	7469	8469	EnQ Taryfa 4 [100 varh]

Wartości rejestrów energii czynnej w 100 Wh są podawane w zakresie do 999999 [100 Wh] (99999.9 [kWh]), wartości powyżej 999999 [100 Wh] są podawane w rejestrach opisanych jako 10 MWh (to samo dotyczy rejestrów energii biernej).

DODATEK A. OBLICZANIE SUMY KONTROLNEJ

W dodatku tym przedstawiono przykłady funkcji w języku C, obliczające sumę kontrolną LRC dla trybu ASCII oraz CRC dla trybu RTU.

Funkcja do obliczenia LRC ma dwa argumenty:

<i>unsigned char *outMsg;</i>	Wskaźnik do bufora komunikacyjnego, zawierającego dane binarne, z których należy obliczyć LRC
<i>unsigned short usDataLen;</i>	Liczba bajtów w buforze komunikacyjnym

Funkcja zwraca LRC typu *unsigned char*.

static unsigned char LRC(outMsg, usDataLen)

```
unsigned char *outMsg;                /* bufor do obliczenia LRC */
unsigned short usDataLen;           /* liczba bajtów w buforze */
{
    unsigned char uchLRC = 0;          /* inicjalizacja LRC */
    while (usDataLen--)
        uchLRC += *outMsg++;        /* dodaj bajt bufora bez przeniesienia */
    return ((unsigned char)-(char uchLRC)); /* zwraca sumę w kodzie uzupełnienia do dwóch */
}
```

Poniżej przedstawiono przykład funkcji w języku C obliczającej sumę CRC. Wszystkie możliwe wartości sumy CRC są umieszczone w dwóch tablicach. Pierwsza tablica zawiera starszy bajt wszystkich z 256 możliwych wartości 16-bitowego pola CRC, natomiast druga tablica młodszy bajt.

Wyznaczenie sumy CRC poprzez indeksowanie tablic jest o wiele szybsze niż obliczenie nowej wartości CRC dla każdego znaku z bufora komunikacyjnego.

Uwaga: Poniższa funkcja przestawia bajty sumy CRC starszy/młodszy, tak że wartość CRC zwracana przez funkcję może być bezpośrednio umieszczona w buforze komunikacyjnym.

Funkcja do obliczenia CRC ma dwa argumenty:

<i>unsigned char *puchMsg;</i>	Wskaźnik do bufora komunikacyjnego, zawierającego dane binarne, z których należy obliczyć CRC
<i>unsigned short usDataLen;</i>	Liczba bajtów w buforze komunikacyjnym

Funkcja zwraca CRC typu *unsigned short*.

unsigned short CRC16(puchMsg, usDataLen)

```
unsigned char *puchMsg;              /* bufor do obliczenia CRC */
unsigned short usDataLen;           /* liczba bajtów w buforze */
{
    unsigned char uchCRChi = 0xFF;     /* inicjalizacja starszego bajtu CRC */
    unsigned char uchCRClo = 0xFF;     /* inicjalizacja młodszego bajtu CRC */
    while (usDataLen--)
    {
        uIndex = uchCRChi ^ *puchMsg++; /* obliczenie CRC */
        uchCRChi = uchCRClo ^ crc_hi[uIndex];
        uchCRClo = crc_lo[uIndex];
    }
    return(uchCRChi<<8 | uchCRClo);
}
```

```
//tablica starszego bajtu CRC
```

```
const unsigned char crc_hi[]={
```

```
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
```

```
};
```

```
//tablica mlodsze bajtu CRC
```

```
const unsigned char crc_lo[]={
```

```
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
```

```
0x40
```

```
};
```

© 2010 LZAE Lumel S.A., Wszelkie prawa zastrzeżone.
ND1-07C/2 (22.06.11)

Lubuskie Zakłady Aparatów Elektrycznych LUMEL S.A.
ul. Sulechowska 1,
65-022 Zielona Góra