

# SPECYFIKACJA

*HCRH-Modbus-V*



***„Przetwornik wilgotności i temperatury  
RH&T ► MODBUS RTU”***

Opracował:  
**HOTCOLD s.c.**

2017-11-29

|           |                                        |          |
|-----------|----------------------------------------|----------|
| <b>1.</b> | <b>Wprowadzenie.....</b>               | <b>3</b> |
| 1.1.      | Funkcje urządzenia.....                | 3        |
| 1.2.      | Charakterystyka urządzenia.....        | 3        |
| <b>2.</b> | <b>Dane techniczne.....</b>            | <b>4</b> |
| 2.1.      | Parametry ogólne przetwornika.....     | 4        |
| 2.2.      | Parametry pomiaru wilgotności.....     | 4        |
| 2.3.      | Parametry pomiaru temperatury.....     | 4        |
| 2.4.      | Parametry wyjścia analogowego.....     | 5        |
| 2.5.      | Parametry interfejsu szeregowego.....  | 5        |
| <b>3.</b> | <b>Instalacja.....</b>                 | <b>5</b> |
| 3.1.      | Bezpieczeństwo.....                    | 5        |
| 3.2.      | Konstrukcja urządzenia.....            | 5        |
| 3.3.      | Opis wyprowadzeń.....                  | 6        |
| 3.4.      | Konfiguracja adresu.....               | 7        |
| 3.5.      | Konfiguracja prędkości.....            | 7        |
| 3.6.      | Przywracanie ustawień fabrycznych..... | 7        |
| 3.7.      | Wytyczne.....                          | 8        |
| <b>4.</b> | <b>Protokół MODBUS.....</b>            | <b>9</b> |
| 4.1.      | Mapa rejestrów.....                    | 9        |
| 4.2.      | Funkcje protokołu.....                 | 10       |
| 4.3.      | Format danych.....                     | 11       |
| 4.4.      | Suma kontrolna CRC.....                | 12       |

# 1. Wprowadzenie

Przedmiotem niniejszego opracowania jest charakterystyka funkcjonalności przetwornika wilgotności i temperatury opartego na czujniku firmy Sensirion serii SHT, z interfejsem RS-485 z wbudowanym protokołem MODBUS RTU, z równoległym wyjściem analogowym RH w standardzie 0-10V

UWAGA: Przed przystąpieniem do uruchomienia modułu należy zapoznać się z tekstem zawartym w niniejszym opracowaniu.

## 1.1. Funkcje urządzenia

- pomiar **wilgotności względnej**
- równoległe analogowe wyjście napięciowe 0-10 [V] (w zakresie 0-100% RH)
- pomiar **temperatury**
- wyliczenie punktu rosy
- diodowa sygnalizacja pracy urządzenia
- szeregowy interfejs RS-485 (odczyt wartości pomiarowych, konfiguracja parametrów pracy)
  - protokół MODBUS RTU
  - komunikacja w trybie HALF DUPLEX
  - sprzętowo konfigurowany adres (1-127)
  - sprzętowo konfigurowana prędkość (9600, 19200, 38400, 57600, 115200)

## 1.2. Charakterystyka urządzenia

Podstawową funkcją przetwornika HCRH-Modbus jest wyznaczanie chwilowych wartości wilgotności względnej skompensowanej temperaturowo oraz równoległe chwilowych wartości temperatury. Zmierzone za pośrednictwem zintegrowanego czujnika SHT firmy Sensirion wartości, następnie przeliczone i uśrednione w mikrokontrolerze, dostępne są w jego pamięci (w rejestrach typu HOLDING REGISTERS) zgodnie ze standardem MODBUS. Odczyt rejestrów odbywa się za pomocą funkcji protokołu MODBUS przesyłanych szeregowym interfejsem RS-485. Sygnalizacja braku / błędu czujnika, realizowana jest za pośrednictwem rejestru statusowego. Równoległe wartość wilgotności względnej prezentowana jest również w postaci analogowej na wyjściu napięciowym w standardzie 0-10 [V].

## 2. Dane techniczne

### 2.1. Parametry ogólne przetwornika

|                              |                                                    |
|------------------------------|----------------------------------------------------|
| <b>Zasilanie</b>             |                                                    |
| - napięciem stałym           | DC 24V (20...30V)                                  |
| - napięciem przemiennym      | AC 24V (20...27,6V)                                |
| <b>Pobór prądu</b>           |                                                    |
| - typowy <sup>1)</sup>       | <13,0 mA                                           |
| - maksymalny <sup>2)</sup>   | <23,0 mA                                           |
| <b>Sygnalizacja LED</b>      | 0,2 Hz                                             |
| <b>Złącze instalacyjne</b>   | śrubowe w rastrze 5.00mm ( $\leq 2,5\text{mm}^2$ ) |
| <b>Wymiary</b>               | 80 x 80 x 25 (L x H x W)                           |
| <b>Waga /Stopień ochrony</b> | 65 g / IP20                                        |
| <b>Montaż <sup>3)</sup></b>  | naścienny                                          |
| <b>Środowisko pracy</b>      | bezpyłowe, powietrze, gazy neutralne               |
| <b>Temperatura pracy</b>     | -10°C ÷ 60°C                                       |

- 1) Średni pobór prądu urządzenia w warunkach: transmisja 10 zapytań na sekundę; prędkość transmisji 9600 b/s; jednoczesny odczyt 3 rejestrów; rezystory terminujące magistralę 2 x 120Ω; zasilanie 24V DC, wyjście napięciowe obciążone rezystancją 10k;
- 2) Maksymalny chwilowy pobór prądu urządzenia w warunkach jak w punkcie 1) + wyjście napięciowe obciążone rezystancją 1k;
- 3) Instalacji urządzenia powinien dokonywać wykwalifikowany personel; Orientacja pionowa zgodnie z oznakowaniem UP – góra, DOWN – dół;

### 2.2. Parametry pomiaru wilgotności

|                                      |                     |
|--------------------------------------|---------------------|
| <b>Typ czujnika</b>                  | Seria SHT           |
| <b>Zakres pomiarowy</b>              | 0 ÷ 100 %RH         |
| <b>Rozdzielczość</b>                 | 12 bitów (0,04 %RH) |
| <b>Dokładność dla T=25°C</b>         |                     |
| - w zakresie 20 ÷ 80 %RH             | ±1,8 %RH            |
| - w pozostałym zakresie              | ±( 1,8 ÷ 3 ) %RH    |
| <b>Histereza</b>                     | ±1 %RH              |
| <b>Częstotliwość próbkowania</b>     | 1 Hz                |
| <b>Czas odpowiedzi <sup>1)</sup></b> | 8 s                 |

- 1) Warunkiem uzyskania podanych czasów odpowiedzi jest przepływ powietrza > 1m/s w temperaturze 25°C; podany czas odpowiedzi jest równy jednej stałej czasowej odpowiadającej 63% wartości ustalonej;

### 2.3. Parametry pomiaru temperatury

|                                      |                    |
|--------------------------------------|--------------------|
| <b>Typ czujnika</b>                  | Seria SHT          |
| <b>Rozdzielczość</b>                 | 14 bitów (0,01 °C) |
| <b>Zakres pomiarowy</b>              | -40°C ÷ 123,8°C    |
| <b>Dokładność</b>                    |                    |
| - w zakresie 10 ÷ 50 °C              | ±0,2 °C            |
| - w zakresie 0 ÷ 60 °C               | ±0,25 °C           |
| - w pozostałym zakresie              | ±(0,25 ÷ 1,0) °C   |
| <b>Częstotliwość próbkowania</b>     | 1 Hz               |
| <b>Czas odpowiedzi <sup>1)</sup></b> | 5 ÷ 30 s           |

- 1) Warunkiem uzyskania podanych czasów odpowiedzi jest przepływ powietrza > 1m/s; podany czas odpowiedzi jest równy jednej stałej czasowej odpowiadającej 63% wartości ustalonej;

## 2.4. Parametry wyjścia analogowego

|                           |                          |
|---------------------------|--------------------------|
| Typ wyjścia               | napięciowe               |
| Zakres wyjściowy          | 10 V                     |
| Rozdzielczość             | 12 bitów (5 mV)          |
| Obciążalność              | $R_L > 1\text{ k}\Omega$ |
| Częstotliwość odświeżania | 1 Hz                     |

## 2.5. Parametry interfejsu szeregowego

|                                     |                                           |
|-------------------------------------|-------------------------------------------|
| Warstwa fizyczna                    | RS-485                                    |
| Protokół komunikacji                | MODBUS RTU                                |
| Konfiguracje połączeń <sup>1)</sup> | HALF DUPLEX                               |
| Prędkości transmisji                | 9600 / 19200 / 38400 / 57600 / 115200 b/s |

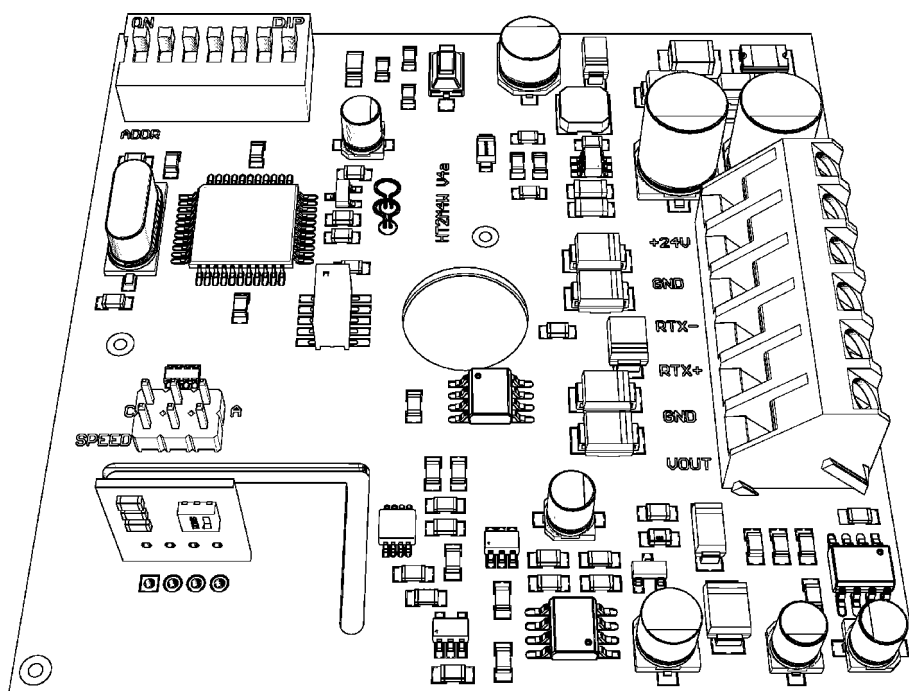
1) HALF DUPLEX – komunikacja dwukierunkowa jedną parą przewodów;

## 3. Instalacja

### 3.1. Bezpieczeństwo

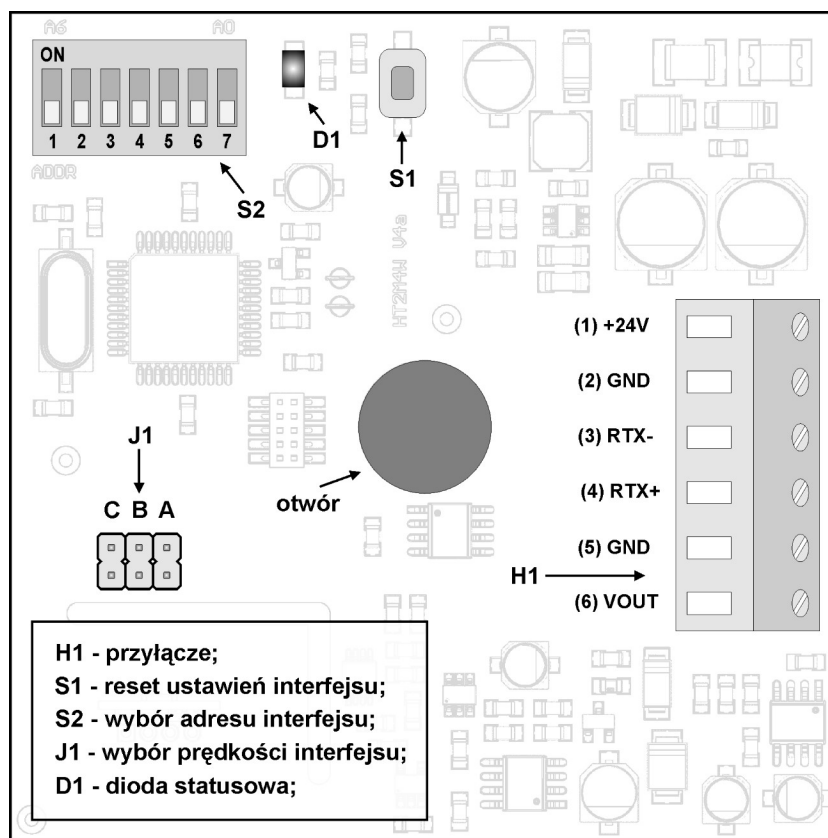
- Instalacji urządzenia powinien dokonywać wykwalifikowany personel!
- Wszystkie podłączenia należy wykonać zgodnie ze schematami elektrycznymi przedstawionymi w niniejszej specyfikacji!
- Przed przystąpieniem do uruchomienia należy sprawdzić wszystkie podłączenia elektryczne!

### 3.2. Konstrukcja urządzenia

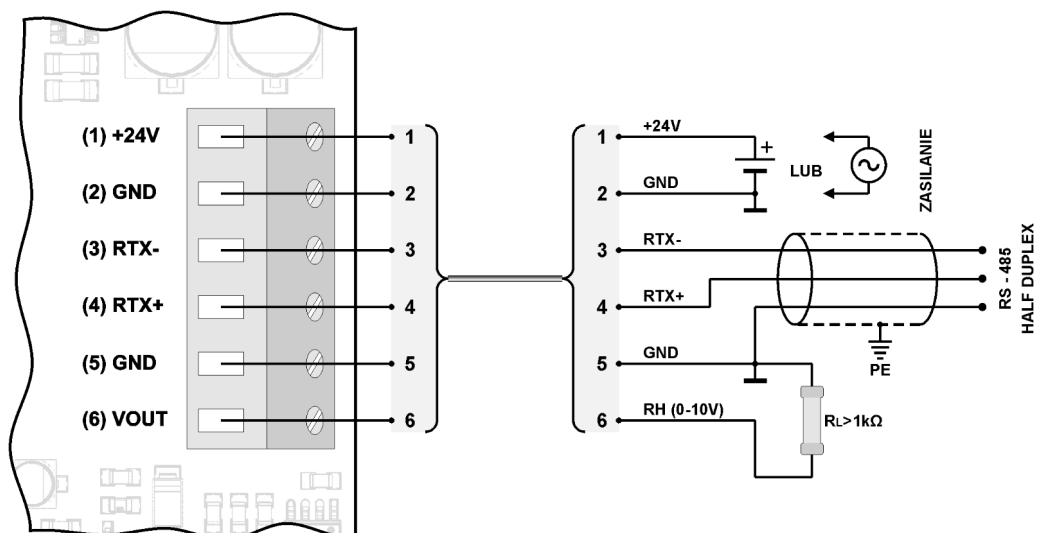


Rysunek 1. Widok obwodu drukowanego wersji **naściennego** przetwornika.

### 3.3. Opis wyprowadzeń



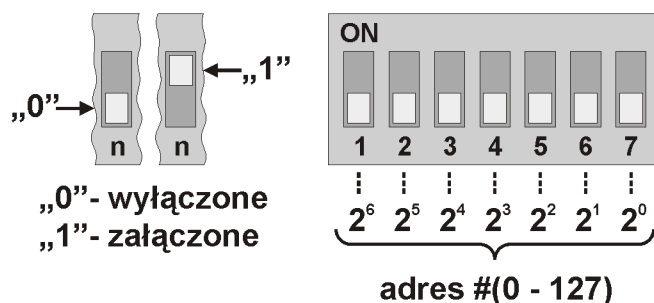
Rysunek 2. Opis wyprowadzeń przetwornika w wersji **naściennej**.



Rysunek 3. Schemat podłączenia przetwornika w wersji **naściennej**.

### 3.4. Konfiguracja adresu

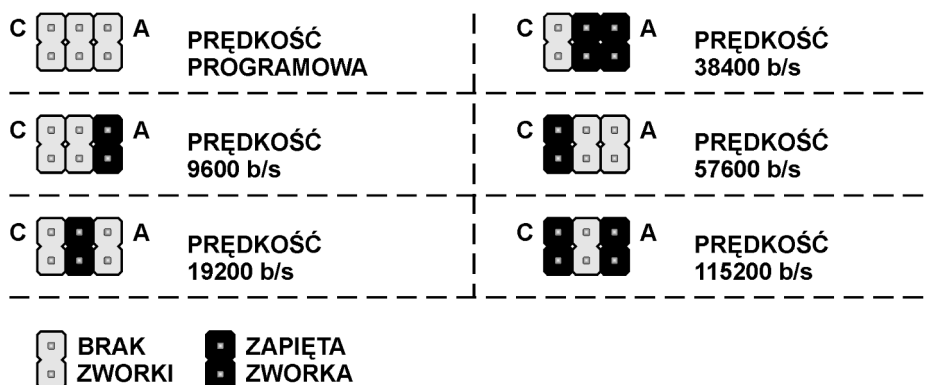
Urządzenie wyposażono w przełącznik typu DIP-SWITCH (5, 6 lub 7 pozycyjny) do sprzętowego ustalenia adresu (od „1” do max „127”). Ustawienie adresu „0” na przełączniku spowoduje użycie adresu zapisanego w urządzeniu za pośrednictwem protokołu MODBUS (domyślnie „1”).



Rysunek 4. Adresacja przetwornika.

### 3.5. Konfiguracja prędkości

Urządzenie wyposażono w układ 3 zwerek do sprzętowego ustalenia prędkości interfejsu RS-485 (zgodnie z poniższą tabelą). Brak zwerek spowoduje użycie wartości prędkości zapisanej w urządzeniu za pośrednictwem protokołu MODBUS (domyślnie „9600 b/s”).



Rysunek 5. Konfiguracja prędkości interfejsu RS-485.

### 3.6. Przywracanie ustawień fabrycznych

Funkcja przywracania ustawień fabrycznych dotyczy wyłącznie parametrów transmisji interfejsu RS-485 (w tym adresu i prędkości). Aby przywrócić ustawienia należy nacisnąć i przytrzymać przycisk S1 przez około 2 sekundy (zabezpieczenie przed przypadkowym naciśnięciem). Gdy dioda D1 zacznie mrugać należy zwolnić przycisk. Urządzenie zacznie działać z nowymi ustawieniami automatycznie.

### 3.7. Wytyczne

- W przypadku pracy w otoczeniu dużych zakłóceń, należy zastosować przewody ekranowane.
- Ekran przewodu należy podłączyć do najbliższego punktu PE od strony zasilacza.

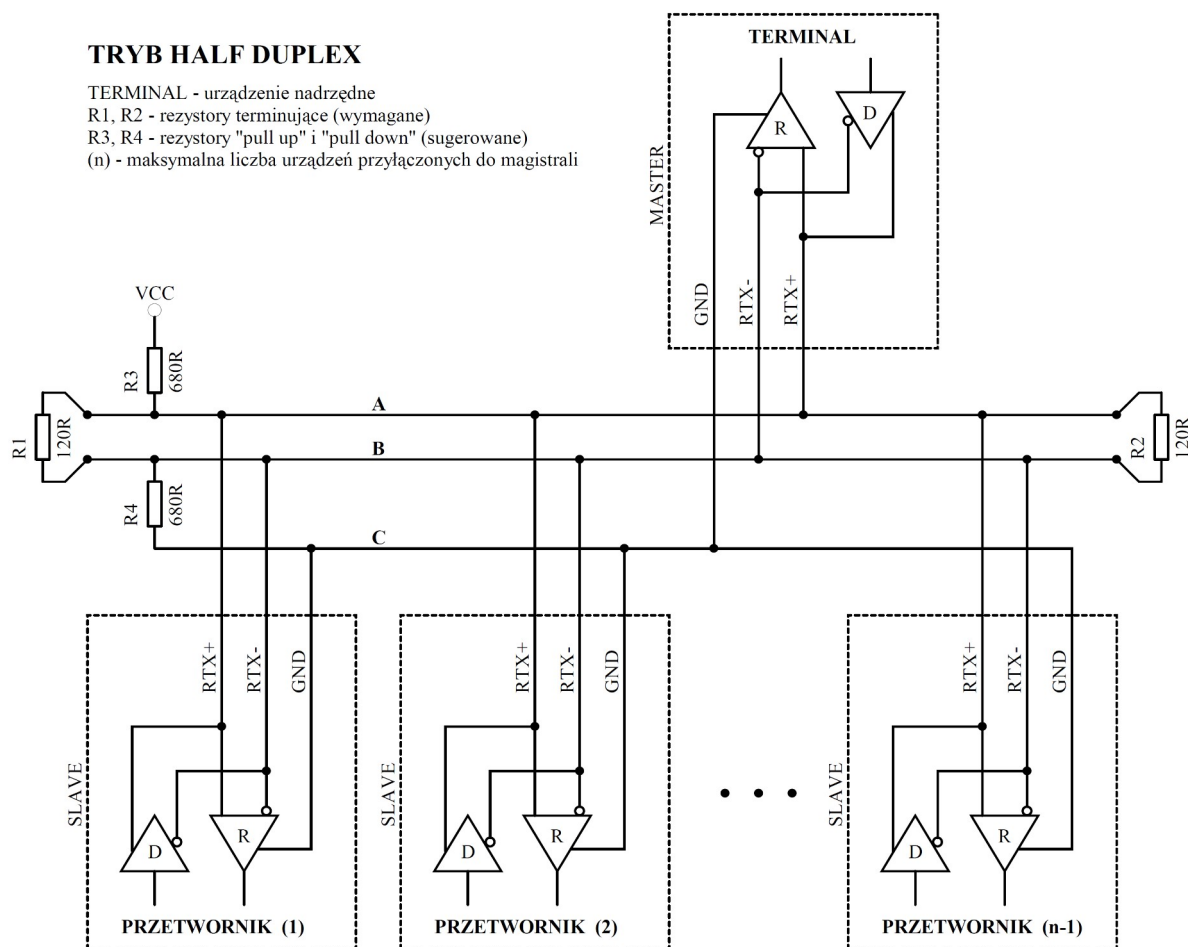
#### TRYB HALF DUPLEX

TERMINAL - urządzenie nadrzędne

R1, R2 - rezystory terminujące (wymagane)

R3, R4 - rezystory "pull up" i "pull down" (sugerowane)

(n) - maksymalna liczba urządzeń przyłączonych do magistrali



**Rysunek 6.** Sposób podłączenia przetwornika do magistrali RS-485 pracującej w trybie HALF DUPLEX.

## 4. Protokół MODBUS

### 4.1. Mapa rejestrów

Tabela rejestrów:

| Nr rejestru | Wartości          | Opis                                                               |
|-------------|-------------------|--------------------------------------------------------------------|
| 1           | 1 – 1000          | Wilgotność względna ( 1 = 0,1%; 1000 = 100% )                      |
| 2           | -4000 – 12380     | Temperatura [ °C ] ( 1 = 0,01 °C ) ze znakiem                      |
| 3           | -4000 – 12380     | Punkt rosy [ °C ] ( 1 = 0,01 °C ) ze znakiem                       |
| 4           | 1234              | Rejestr hasła                                                      |
| 5           | 1 / 2 / 3         | Rejestr poleceń                                                    |
| 6           | wg tabeli poleceń | Rejestr parametru                                                  |
| 7           | 0-65535           | Licznik poprawnych ramek                                           |
| 8           | 0-65535           | Licznik wyjątków                                                   |
| 9           | 0-65535           | Licznik błędnych CRC                                               |
| 10          | 0-65535           | Licznik błędnych bajtów                                            |
| 11          | -                 | nie używany                                                        |
| 12          | 0 / 1 / 2         | Rejestr statusowy ( 0: "NO SENSOR", 1: "SENSOR OK", 2: "ERROR" (*) |
| 13          | 1000<br>(0x03E8)  | Wartość testowa - do weryfikacji poprawności odczytu rejestrów     |

(\*) "NO SENSOR" – brak czujnika; "SENSOR OK" – prawidłowe działanie czujnika; "ERROR" – błąd działania czujnika;

Tabela poleceń:

| Nr polecenia | Funkcja                   | Parametry                                                                                                      |
|--------------|---------------------------|----------------------------------------------------------------------------------------------------------------|
| 1            | Ustaw adres urządzenia    | 1 – 247 (1-wartość domyślna)                                                                                   |
| 2            | Ustaw prędkość transmisji | 96 – 9600 b/s (wartość domyślna)<br>192 – 19200 b/s<br>384 – 38400 b/s<br>576 – 57600 b/s<br>1152 – 115200 b/s |
| 3            | Ustaw bity parzystości    | 0 – NO PARITY; brak bitu parzystości<br>1 – EVEN PARITY; (wartość domyślna)<br>2 – ODD PARITY,                 |
| 4            | Ustaw bity Stopu          | 1 – 1 x STOP; 1 bit stopu (wartość domyślna)<br>2 – 2 x STOP; 2 bity stopu                                     |
| 5            | Reset urządzenia          | 1 – programowy reset urządzenia                                                                                |

Uwagi:

- Podanie błędnej lub spoza zakresu wartości parametru, skutkuje wpisaniem do rejestru poleceń wartości 0xEEEE.
- Każdorazowemu wywołaniu polecenia musi towarzyszyć wpisanie hasła (1234 decymalnie).

- Wywołanie polecenia poprzez pojedyncze wpisy do rejestrów, musi zostać zakończone wpisaniem hasła.

## 4.2. Funkcje protokołu

W przetworniku zaimplementowano następujące funkcje standardu MODBUS:

| KOD       | ZNACZENIE                        |
|-----------|----------------------------------|
| 03 (0x03) | Odczyt N x 16-bitowych rejestrów |
| 16 (0x10) | Zapis N x 16-bitowych rejestrów  |

### 4.2.1. Odczyt zawartości grupy rejestrów wyjściowych (0x03)

Format żądania:

| Opis                 | Rozmiar | Wartości        |
|----------------------|---------|-----------------|
| Adres urządzenia     | 1 bajt  | 1 – 247 (0xF7)  |
| Kod funkcji          | 1 bajt  | 0x03            |
| Adres bloku danych   | 2 bajty | 0x0000 – 0xFFFF |
| Liczba rejestrów (N) | 2 bajty | 1 – 125 (0x7D)  |
| Suma kontrolna CRC   | 2 bajty | wg obliczeń     |

Format odpowiedzi:

| Opis               | Rozmiar     | Wartości          |
|--------------------|-------------|-------------------|
| Adres urządzenia   | 1 bajt      | 1 – 247 (0xF7)    |
| Kod funkcji        | 1 bajt      | 0x03              |
| Licznik bajtów     | 1 bajty     | 2 x N             |
| Wartości rejestrów | N x 2 bajty | wg mapy rejestrów |
| Suma kontrolna CRC | 2 bajty     | wg obliczeń       |

Format błędu:

| Opis               | Rozmiar | Wartości                  |
|--------------------|---------|---------------------------|
| Adres urządzenia   | 1 bajt  | 1 – 247 (0xF7)            |
| Kod funkcji        | 1 bajt  | 0x83                      |
| Kod błędu          | 1 bajt  | 0x01 / 0x02 / 0x03 / 0x04 |
| Suma kontrolna CRC | 2 bajty | wg obliczeń               |

### 4.2.2. Zapis do grupy rejestrów wyjściowych (0x10)

Format żądania:

| Opis                 | Rozmiar     | Wartości        |
|----------------------|-------------|-----------------|
| Adres urządzenia     | 1 bajt      | 1 – 247 (0xF7)  |
| Kod funkcji          | 1 bajt      | 0x10            |
| Adres bloku danych   | 2 bajty     | 0x0000 – 0xFFFF |
| Liczba rejestrów (N) | 2 bajty     | 1 – 123 (0x7B)  |
| Licznik bajtów       | 1 bajt      | 2 x N           |
| Wartości             | N x 2 bajty | użytkownika     |
| Suma kontrolna CRC   | 2 bajty     | wg obliczeń     |

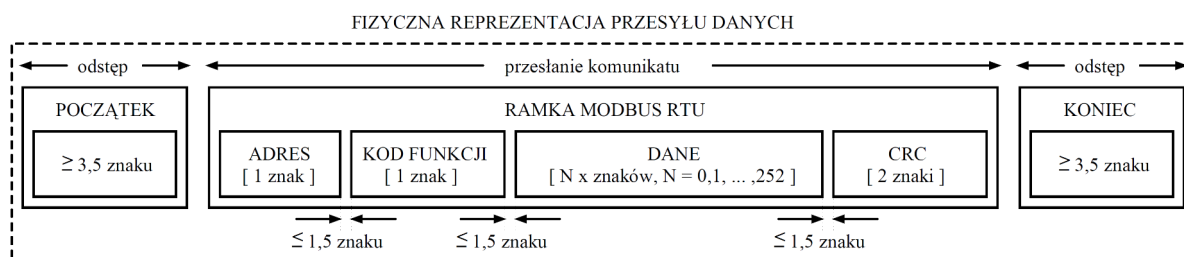
Format odpowiedzi:

| Opis                 | Rozmiar | Wartości        |
|----------------------|---------|-----------------|
| Adres urządzenia     | 1 bajt  | 1 – 247 (0xF7)  |
| Kod funkcji          | 1 bajt  | 0x10            |
| Adres bloku danych   | 2 bajty | 0x0000 – 0xFFFF |
| Liczba rejestrów (N) | 2 bajty | 1 – 123 (0x7B)  |
| Suma kontrolna CRC   | 2 bajty | wg obliczeń     |

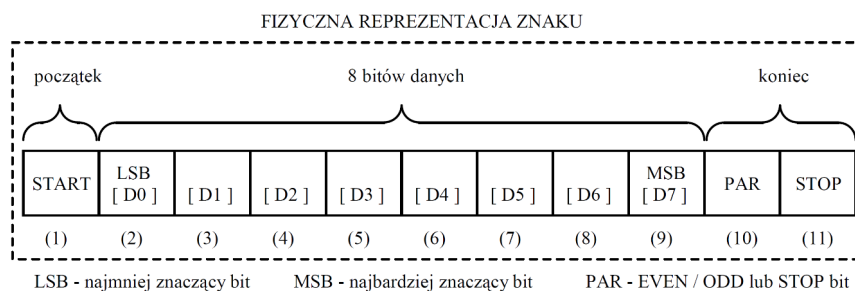
Format błędu:

| Opis               | Rozmiar | Wartości                  |
|--------------------|---------|---------------------------|
| Adres urządzenia   | 1 bajt  | 1 – 247 (0xF7)            |
| Kod funkcji        | 1 bajt  | 0x90                      |
| Kod błędu          | 1 bajt  | 0x01 / 0x02 / 0x03 / 0x04 |
| Suma kontrolna CRC | 2 bajty | wg obliczeń               |

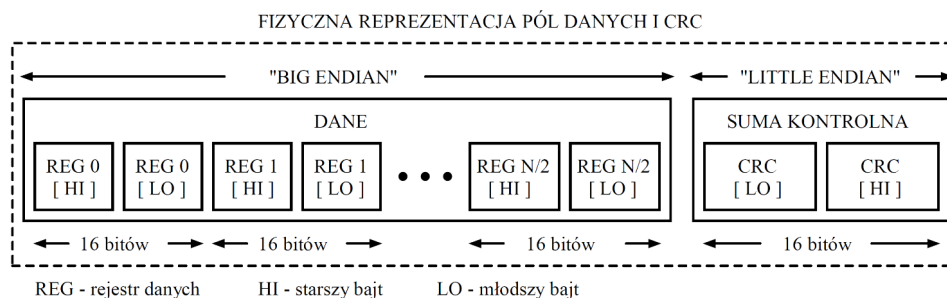
### 4.3. Format danych



Rysunek 7. Przesył danych w standardzie MODBUS RTU zaimplementowany w przetworniku.



Rysunek 8. Format znaku w standardzie MODBUS RTU zastosowany w przetworniku.



Rysunek 9. Format pól danych i CRC w standardzie MODBUS RTU zastosowany w przetworniku.

## 4.4. Suma kontrolna CRC

Zgodnie ze standardem MODBUS, do wyliczenia sumy kontrolnej CRC wykorzystano wielomian:  
 $X^{16} + X^{15} + X^2 + 1$ .

### 4.4.1. Bitowy algorytm liczenia CRC:

Procedura wyznaczania sumy kontrolnej CRC metodą bitową:

- załadowanie wartości 0xFFFF do 16-bitowego rejestru CRC;
- pobranie pierwszego bajta z bloku danych i wykonanie operacji EX-OR z młodszym bajtem rejestru CRC, umieszczenie rezultatu w rejestrze;
- przesunięcie zawartości rejestru CRC w prawo o jeden bit w kierunku najmniej znaczącego bitu (LSB), wyzerowanie najbardziej znaczącego bitu (MSB);
- sprawdzenie stanu najmłodszego bitu (LSB) w rejestrze CRC, jeżeli jego stan równa się 0, to następuje powrót do punktu c, jeżeli 1, to wykonywana jest operacja EX-OR rejestru CRC ze stałą 0xA001;
- powtórzenie punktów c i d do ośmiu razy, co odpowiada przetworzeniu całego bajta;
- powtórzenie sekwencji b, c, d, e dla kolejnego bajta wiadomości, kontynuacja tego procesu aż do przetworzenia wszystkich bajtów wiadomości;
- zawartość rejestru CRC po wykonaniu wymienionych operacji jest poszukiwaną wartością sumy kontrolnej CRC;
- dopisanie sumy kontrolnej CRC do ramki MODBUS RTU musi zostać poprzedzone zamianą miejscami starszego i młodszego bajta rejestru CRC.

### 4.4.2. Tablicowy algorytm liczenia CRC:

Przykład implementacji procedury wyznaczania sumy kontrolnej CRC metodą tablicową:

```
/* The function returns the CRC as a unsigned short type */
unsigned short CRC16 ( puchMsg, usDataLen )
/* message to calculate CRC upon */
unsigned char *puchMsg ;
/* quantity of bytes in message */
unsigned short usDataLen ;

{
    /* high byte of CRC initialized */
    unsigned char uchCRCHi = 0xFF ;
    /* low byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ;
    /* will index into CRC lookup table */
    unsigned uIndex ;

    /* pass through message buffer */
    while (usDataLen--)
    {
        /* calculate the CRC */
        uIndex = uchCRCLo ^ *puchMsg++ ;
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex] ;
        uchCRCHi = auchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}
```

```

/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
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, 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, 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
0x40
} ;

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
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
};

```