

# SPECYFIKACJA

*HCEM-1*



***„Przetwornik stężenia pyłu  
zawieszonego, temperatury  
i wilgotności”***

Opracował:  
***HOTCOLD s.c.***

2022-04-28

<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 stężenia pyłów.....	5
2.3.	Parametry pomiaru temperatury i wilgotności.....	5
2.4.	Parametry interfejsu szeregowego.....	5
<b>3.</b>	<b>Instalacja.....</b>	<b>6</b>
3.1.	Bezpieczeństwo.....	6
3.2.	Konstrukcja urządzenia.....	6
3.3.	Schemat podłączenia.....	7
3.4.	Konfiguracja adresu.....	7
3.5.	Konfiguracja prędkości.....	8
3.6.	Przywracanie ustawień fabrycznych.....	8
3.7.	Wytyczne.....	9
<b>4.</b>	<b>Protokół MODBUS.....</b>	<b>10</b>
4.1.	Mapa rejestrów.....	10
4.2.	Funkcje protokołu.....	12
4.3.	Format danych.....	14
4.4.	Suma kontrolna CRC.....	16

# 1. Wprowadzenie

Przedmiotem niniejszego opracowania jest charakterystyka funkcjonalności przetwornika stężenia pyłu zawieszonego opartego na czujniku firmy HoneyWell serii HPM, z interfejsem RS-485 z wbudowanym protokołem MODBUS RTU oraz opcjonalnym wyświetlaczem

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

## 1.1. Funkcje urządzenia

- pomiar stężenia pyłów zawieszonych PM1.0; PM2.5; PM4.0; PM10 w zakresie 0-1000 $\mu\text{g}/\text{m}^3$
- pomiar temperatury i wilgotności powietrza z wyliczeniem punktu rosy
- opcjonalny wyświetlacz graficzny OLED o rozdzielczości 256x64
- RS-485 (odczyt wartości pomiarowych, konfiguracja parametrów pracy)
  - protokół MODBUS RTU
  - komunikacja w trybie HALF DUPLEX
  - sprzętowo (1-127) lub programowo konfigurowany adres (1-247)
  - sprzętowo lub programowo konfigurowana prędkość interfejsu

## 1.2. Charakterystyka urządzenia

Podstawową funkcją przetwornika stężenia pyłu zawieszonego jest wyznaczanie chwilowych wartości stężenia pyłów zawieszonych w otaczającym powietrzu. Dodatkowo mierzone są: temperatura i wilgotność, jest wyliczany punkt rosy. Zmierzone za pośrednictwem zintegrowanego czujnika serii HPM firmy HoneyWell wartości, następnie przeliczone w mikrokontrolerze, dostępne są w jego pamięci (w rejestrach typu HOLDING REGISTERS) zgodnie ze standardem MODBUS RTU oraz wyświetlane na wyświetlaczu urządzenia. Odczyt rejestrów przesyłanych szeregowym interfejsem RS-485, odbywa się za pomocą funkcji protokołu MODBUS. Sygnalizacja braku / błędu czujnika, realizowana jest za pośrednictwem rejestru statusowego.

## 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>	<15,0 mA
- maksymalny <sup>2)</sup>	<30,0 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>	120 x 80 x 25 (L x H x W)
<b>Waga</b>	-
<b>Montaż <sup>3)</sup></b>	-
<b>Środowisko pracy</b>	powietrze, gazy neutralne
<b>Temperatura pracy</b>	-20°C ÷ 50°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

2) pobór prądu w trakcie przetwarzania + warunki z pkt 1)

3) Instalacji urządzenia powinien dokonywać wykwalifikowany personel; Orientacja pionowa zgodnie z oznakowaniem UP – góra, DOWN – dół na płytkach urządzenia; nie zasłaniać otworów wentylacyjnych urządzenia.

## 2.2. Parametry pomiaru stężenia pyłów

Typ czujnika	LASEROWY
Zakres pomiarowy	0-1000 $\mu\text{g}/\text{m}^3$
Rozdzielczość	1 $\mu\text{g}$
Dokładność 25°C $\pm 5^\circ\text{C}$	
- w zakresie 0-100 $\mu\text{g}/\text{m}^3$	PM2.5: $\pm 15 \mu\text{g}/\text{m}^3$ ; PM1.0, PM4.0, PM10: $\pm 25 \mu\text{g}/\text{m}^3$
- w zakresie 100-1000 $\mu\text{g}/\text{m}^3$	PM2.5: $\pm 15 \%$ ; PM1.0, PM4.0, PM10: $\pm 25 \%$
Częstotliwość próbkowania <sup>1)</sup>	Min: 60s Max: 12h
Czas pomiaru <sup>1)</sup>	Min: 6s Max: 3min

1) Od czasu pomiaru i jego częstotliwości zależy żywotność czujnika pomiarowego. Domyślna wartość czasu pomiaru wynosi 30 sekund a częstotliwość próbkowania 2min.

## 2.3. Parametry pomiaru temperatury i wilgotności

Typ czujnika	CMOSens
Zakres pomiarowy temperatury	-40 ÷ +125°C
Rozdzielczość	0,1°C
Dokładność	
- w zakresie 0÷60°C	$\pm 0,2^\circ\text{C}$
- w zakresie -40÷120°C	$\pm 0,6^\circ\text{C}$
Częstotliwość próbkowania	5s
Czas pomiaru	<1s
Zakres pomiarowy wilgotności	0 ÷ 100%
Rozdzielczość	0,1°C
Dokładność	
- w zakresie 30÷70%	$\pm 1,8$
- w zakresie 0÷100%	$\pm 3$
Częstotliwość próbkowania	5s
Czas pomiaru	<1s

## 2.4. Parametry interfejsu szeregowego

Warstwa fizyczna	RS-485
Protokół komunikacji	MODBUS RTU
Konfiguracje połączeń <sup>1)</sup>	HALF DUPLEX
Prędkości transmisji	2400 4800 9600 14400 19200 28800 38400 57600 115200 b/s

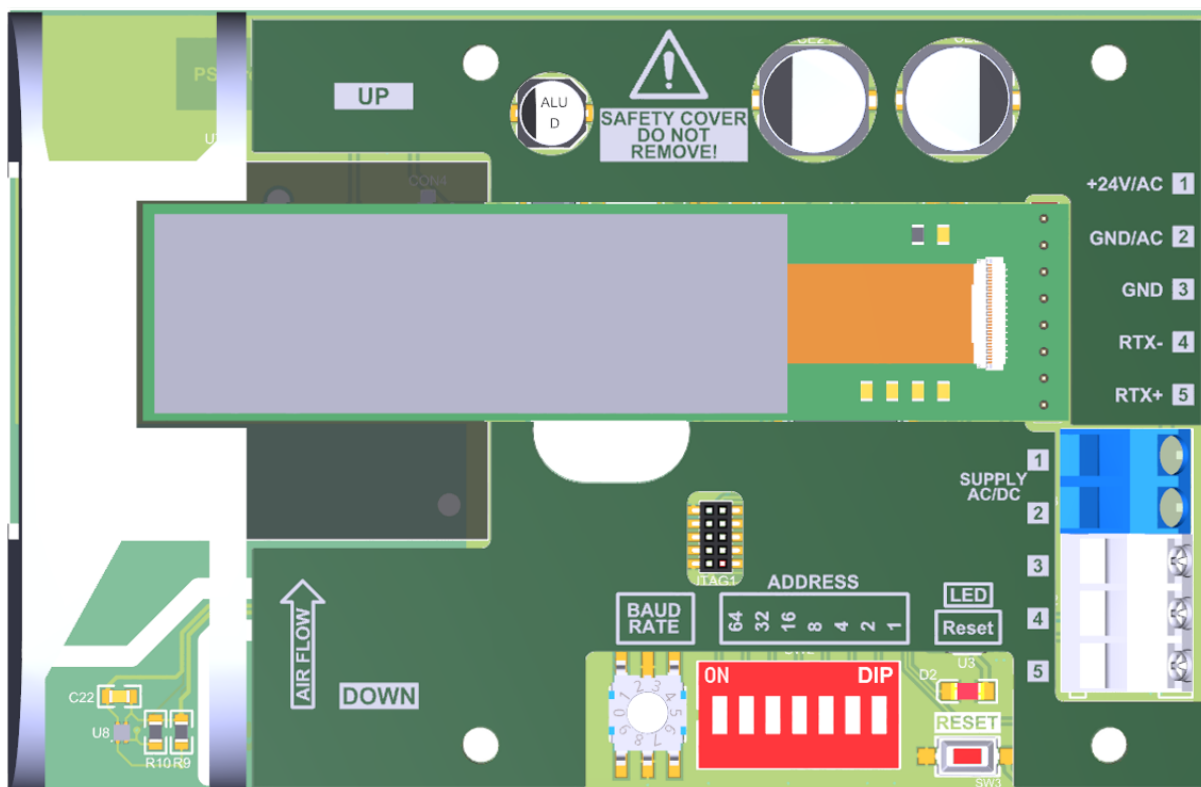
1) HALF DUPLEX – komunikacja dwukierunkowa jedną parą przewodów;

### 3. Instalacja

#### 3.1. Bezpieczeństwo

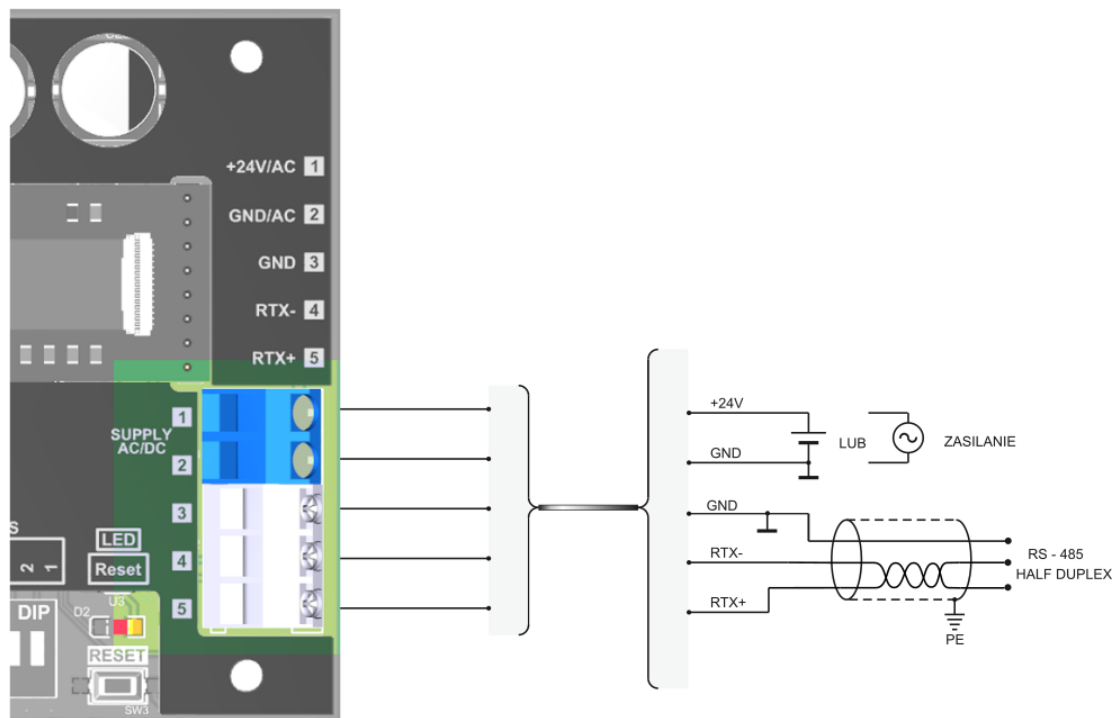
- Instalacji urządzenia powinien dokonywać wykwalifikowany personel!
- Wszystkie połączenia należy wykonać zgodnie ze schematami elektrycznymi przedstawionymi w niniejszej specyfikacji!
- Przed przystąpieniem do uruchomienia należy sprawdzić wszystkie połączenia elektryczne!

#### 3.2. Konstrukcja urządzenia



Rysunek 1. Widok obwodu drukowanego przetwornika.

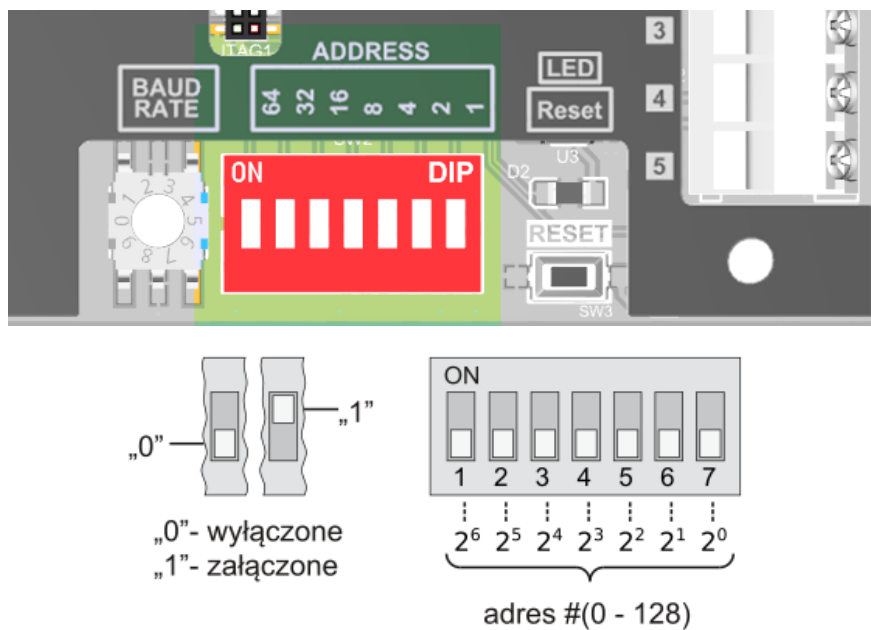
### 3.3. Schemat podłączenia



Rysunek 2. Schemat podłączenia przetwornika.

### 3.4. Konfiguracja adresu

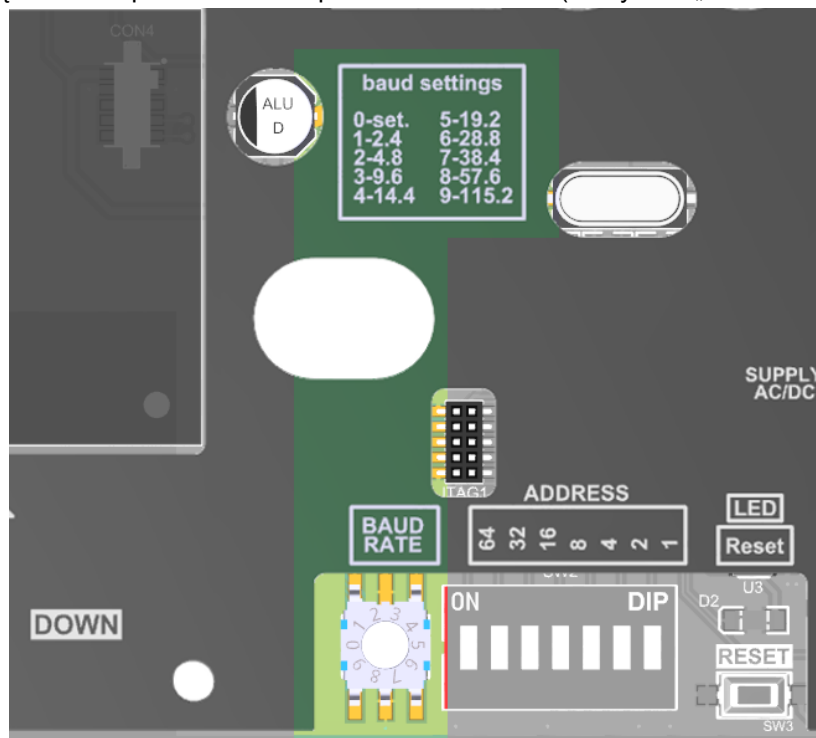
Urządzenie wyposażono w przełącznik 7-pozycyjny do sprzętowego ustalenia adresu (od „1” do „128”). Ustawienie adresu „0” na przełączniku spowoduje użycie adresu zapisanego w urządzeniu za pośrednictwem protokołu MODBUS (domyślnie „1”).



Rysunek 3. Adresacja przetwornika.

### 3.5. Konfiguracja prędkości

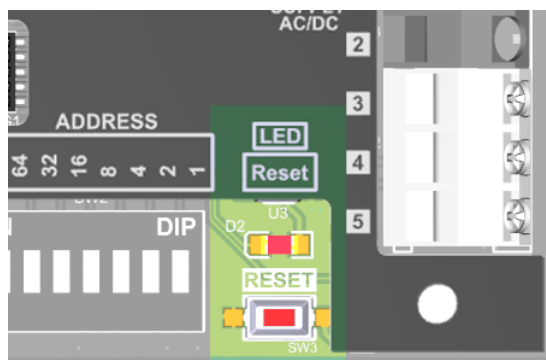
Urządzenie wyposażono w układ przełącznik obrotowy do sprzętowego ustalenia prędkości interfejsu RS-485 (zgodnie z tabelą na pokrywie). Ustawienie 0 spowoduje użycie wartości prędkości zapisanej w urządzeniu za pośrednictwem protokołu MODBUS (domyślnie „9600 b/s”).



Rysunek 4. 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 migać należy zwolnić przycisk. Urządzenie zacznie działać z nowymi ustawieniami automatycznie.



Rysunek 5. Reset interfejsu.



### 3.7. Wyświetlacz

W trybie normalnej pracy na wyświetlaczu pomiary wyświetlane są naprzemiennie: zawartość pyłów oraz temperatura i wilgotność wraz ze statusem przetwornika.

W trakcie uruchamiania urządzenia wyświetlane są dodatkowe informacje. Jak na poniższych rysunkach.



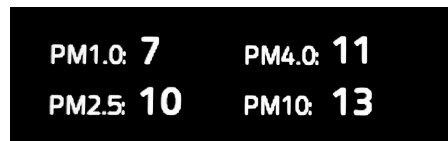
Rysunek 6. Informacje o wersjach



Rysunek 7. Ustawienia



Rysunek 8. Uruchamianie



Rysunek 9. Wyniki pomiaru pyłów



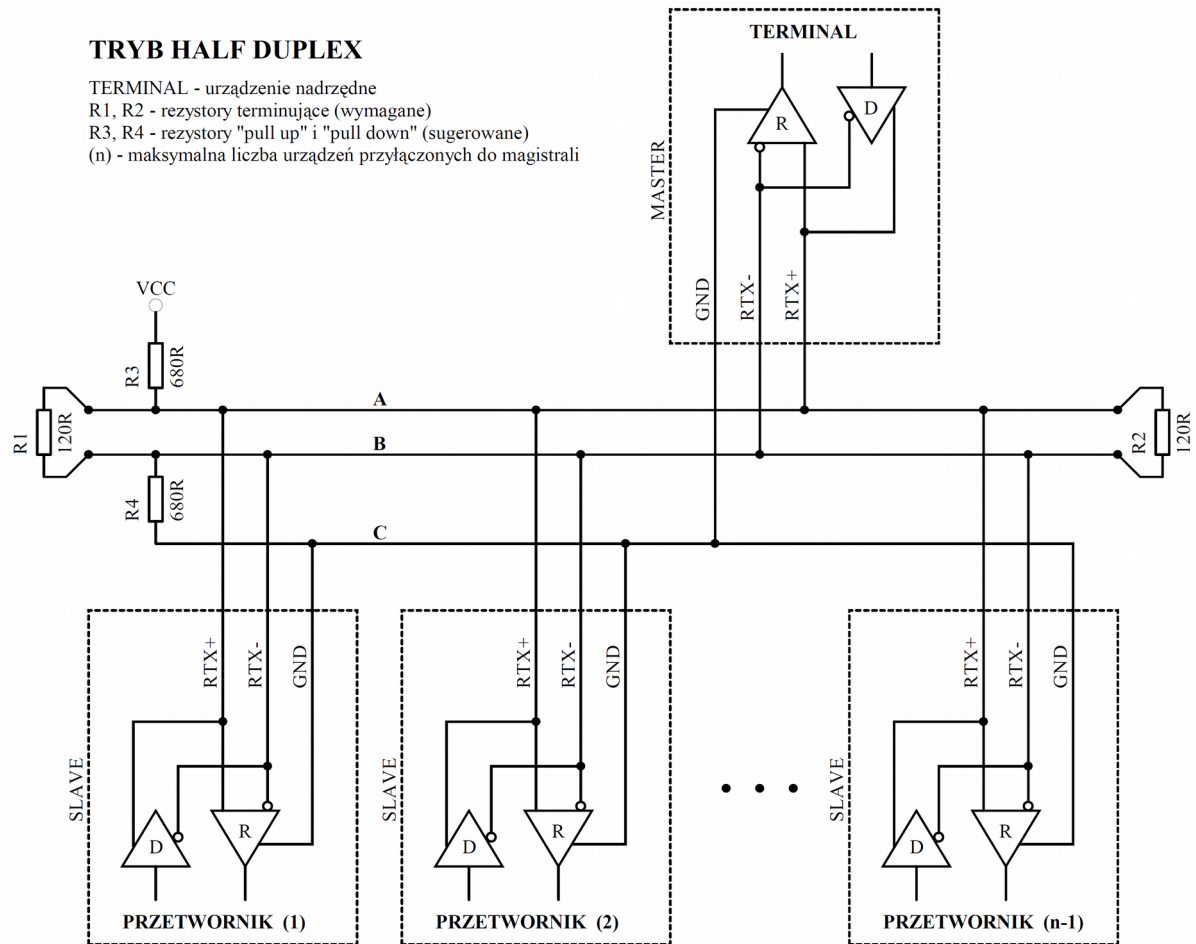
Rysunek 10. Wyniki pomiaru temperatury i wilgotności, punktu rosy, status przetwornika PM: trwa pomiar



Rysunek 11. Wyniki pomiaru temperatury i wilgotności, punktu rosy, status przetwornika PM: oczekiwanie

### 3.8. Wtyczne

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



Rysunek 12. 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	0 - 1000	PM 1.0
2	0 - 1000	PM 2.5
3	0 - 1000	PM 4.0
4	0 - 1000	PM 10
5	0 lub 1	Status danych (PM) 0 - nie poprawne, 1 - ok, 2 - błąd
6	-400 do +1250	Temperatura [deciCelsius]
7	0 do 100	Wilgotność względna [%]
8	-400 do +1250	Punkt rosy [deciCelsius]
9	0 lub 1	Status danych (temp) 0 - nie poprawne, 1 - ok, 2 - błąd
10	1234	Rejestr hasła
11	wg tabeli poleceń	Rejestr poleceń
12	wg tabeli poleceń	Rejestr parametru
20	0-65535	Licznik poprawnych ramek
21	0-65535	Licznik wyjątków
22	0-65535	Licznik błędnych CRC
23	0-65535	Licznik błędnych bajtów
24	1000	Wartość testowa - do weryfikacji poprawności odczytu rejestrów
25	0-65535	Licznik błędów sensora
26	0-65535	Czas do następnego pomiaru
27	0-65535	Licznik błędów komunikacji z sensorem

Tabela poleceń:

Nr polecenia	Funkcja	Parametry
1	Ustaw adres urządzenia	1 – 247 (1-wartość domyślna)
2	Ustaw prędkość transmisji	24 – 2400 b/s 48 – 4800 b/s 96 – 9600 b/s (wartość domyślna) 144 – 14400 b/s 192 – 19200 b/s 228 – 28800 b/s 384 – 38400 b/s 576 – 57600 b/s 1152 – 115200 b/s
3	Ustaw bity parzystości	0 – NO PARITY; brak bitu parzystości 1 – EVEN PARITY; (wartość domyślna) 2 – ODD PARITY,
4	Ustaw bity Stopu	1 – 1 x STOP; 1 bit stopu (wartość domyślna) 2 – 2 x STOP; 2 bity stopu
6	Czas pomiaru	6-180 (30 - wartość domyślna) w sekundach
106	Odczyt parametru 6	brak (odczytana wartość dostępna przy następnym odczycie rejestru parametru)
7	Częstotliwość próbkowania	60-43200 (600 - wartość domyślna) w sekundach
107	Odczyt parametru 7	brak (odczytana wartość dostępna przy następnym odczycie rejestru parametru)
8	Reset	brak , 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
06 (0x06)	Zapis pojedynczego 16-bitowego rej.

Na pozostałe zapytania zostanie zwrócona odpowiedź `ILLEGAL_FUNCTION`.

### 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:

<b>Opis</b>	<b>Rozmiar</b>	<b>Wartości</b>
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	<b>0x10</b>
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:

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

Format błędu:

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

### 4.2.3. Zapis do pojedynczego rejestru wyjściowych (0x06)

Format żądania:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	<b>0x06</b>
Adres rejestru	2 bajty	0x0000 – 0xFFFF
Wartość rejestru	2 bajty	0x0000 - 0xFFFF
Suma kontrolna CRC	2 bajty	wg obliczeń

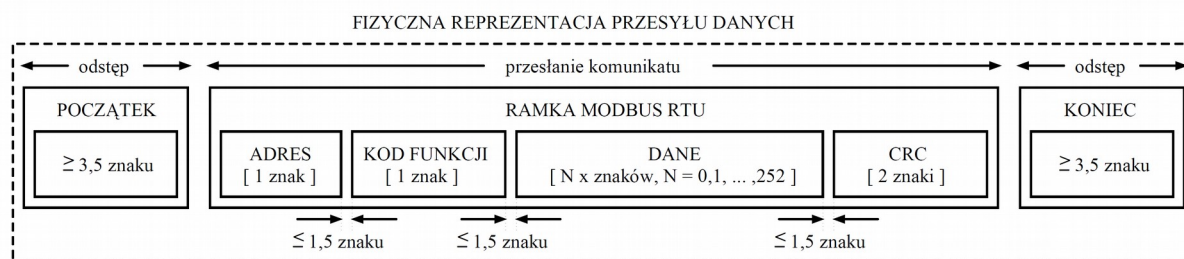
Format odpowiedzi:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	<b>0x06</b>
Adres rejestru	2 bajty	0x0000 – 0xFFFF
Wartość rejestru	2 bajty	0x0000 - 0xFFFF
Suma kontrolna CRC	2 bajty	wg obliczeń

Format błędu:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	<b>0x86</b>
Kod błędu	1 bajt	0x01 / 0x02 / 0x03 / 0x04
Suma kontrolna CRC	2 bajty	wg obliczeń

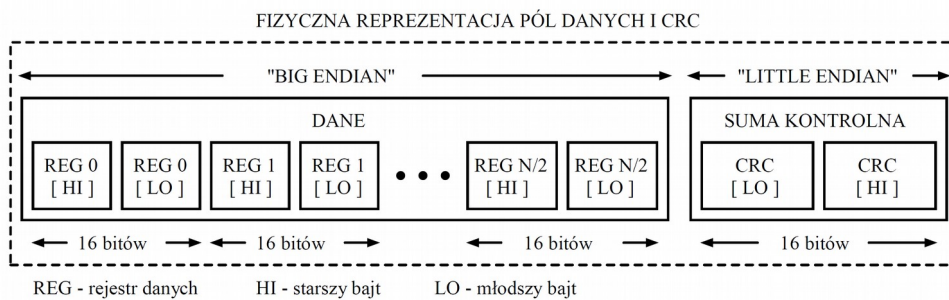
### 4.3. Format danych



Rysunek 13. Przesył danych w standardzie MODBUS RTU zaimplementowany w przetworniku.



Rysunek 14. Format znaku w standardzie MODBUS RTU zastosowany w przetworniku.



**Rysunek 15.** 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:

- a) załadowanie wartości 0xFFFF do 16-bitowego rejestru CRC;
- b) pobranie pierwszego bajta z bloku danych i wykonanie operacji EX-OR z młodszym bajtem rejestru CRC, umieszczenie rezultatu w rejestrze;
- c) przesunięcie zawartości rejestru CRC w prawo o jeden bit w kierunku najmniej znaczącego bitu (LSB), wyzerowanie najbardziej znaczącego bitu (MSB);
- d) 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;
- e) powtórzenie punktów c i d do ośmiu razy, co odpowiada przetworzeniu całego bajta;
- f) powtórzenie sekwencji b, c, d, e dla kolejnego bajta wiadomości, kontynuacja tego procesu aż do przetworzenia wszystkich bajtów wiadomości;
- g) zawartość rejestru CRC po wykonaniu wymienionych operacji jest poszukiwaną wartością sumy kontrolnej CRC;
- h) 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, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0x40
};

```

```

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 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
};

```