# ELECSO PRC-RH

## Moduł pomiarowy

Instrukcja wersja1.0

## Spis Treści

**1.WPROWADZENIE ..... 3**

1.1 NAJWAŻNIEJSZE CECHY ..... 3
*Warunki eksploatacji* ..... *3*
*Cechy sprzętowe* ..... *3*
*Firmware – oprogramowanie wewnętrzne urządzenia, cechy* ..... *3*
1.3 WARUNKI UŻYTKOWANIA ..... 4
1.4 GWARANCJE I SERWIS ..... 4
1.5 CERTYFIKATY I UTYLIZACJA ..... 4

**2. OPIS ELEMENTÓW MODUŁU ELECSO PRC-RH ..... 5**

2.1 MODUŁ POMIARU WILGOTNOŚCI SHTV1.0 – SCHEMAT POŁĄCZENIA DO PRCV2.0 ..... 7
2.1 WIELOPUNKTOWY POMIAR TEMPERATURY Z SYSTEMEM POMIARU WILGOTNOŚCI – SCHEMAT POŁĄCZEŃ. ..... 8

**3. KONFIGURACJA SPRZĘTOWA MODUŁU PRC-RH ..... 9**

3.1 MANUALNA ZMIANA WARTOŚCI ADRESU SPRZĘTOWEGO. ..... 9
3.2 MANUALNE WYSZUKANIE I SEGREGACJA SENSORÓW. ..... 10

**4. SCHEMAT POŁĄCZEŃ SYSTEMU POMIAROWEGO ..... 10**

**5. INTERFEJS RS485 DWUPRZEWODOWY (HALF DUPLEX) ..... 10**

5.1 PARAMETRY TRANSMISJI ..... 10

**6. PROTOKÓŁ MODBUS RTU ..... 11**

6.1 RAMKA TRANSMISJI ..... 11
6.2 KOMENDY I SPECYFIKA ODPOWIEDZI MODBUS W MODUŁACH ELECSO PRC. ..... 13
6.2.1ODCZYT WARTOŚCI TEMPERATUR (KOMENDA NR 03) ..... 13
6.2.2 ODCZYT WARTOŚCI TEMPERATURY SENSORA SHT I WILGOTNOŚCI POWIETRZA (KOMENDA NR 04) ..... 14
6.2.3 ODCZYT WARTOŚCI NAPIĘĆ ZASILANIA ORAZ NATĘŻEŃ PRĄDÓW (KOMENDA NR 06) ..... 16

# 1.Wprowadzenie

System ELECSO PRC-RH zaprojektowany został do pomiaru, rejestracji i kontroli temperatur głównie w obiektach przemysłowych magazynowych. Urządzenie ELECSO PRC-RH współpracuje z cyfrowymi czujnikami temperatury z interfejsem 1-WIRE oraz cyfrowymi sensorami wilgotności SBUS, zaś do komunikacji z systemem, zaimplementowano protokół komunikacyjny MODBUS oparty o popularny interfejs RS485. Powyższe cechy czynią urządzenie uniwersalnym w adaptacji we wszelakich układach automatyki.

## 1.1 Najważniejsze cechy

**Warunki eksploatacji**

- Temperatura pracy: od -40ºC do +75 ºC
- Wilgotność: od 5 do 90 % bez kondensacji
- Napięcie zasilania (stałe/impulsowe) : 12V

**Cechy sprzętowe**

- Możliwość pracy z maksymalnie 16 sensorami temperatury
- 1 kanał pomiarowy 1-Wire o dokładności pomiarowej od 0,0625ºC do 0.5ºC, zakres -40..+125C
- 1 kanał pomiarowy SBUS, do komunikacji z czujnikiem wilgotności o zakresie 0-100%
- Rozdzielczość pomiaru wilgotności: 0.05% RH, maksymaly błąd pomiaru 3%
- Praca w sieci, o warstwie fizycznej RS485 z protokołem MODBUS RTU
- Zabezpieczenie przed zwarciami na liniach pomiarowych
- Zabezpieczenia przepięciowe ESD linii komunikacyjnych i pomiarowych
- Kontrola napięcia zasilania
- Kontrola natężenia prądu w poszczególnych gałęziach pomiarowych
- System powiadomień wizualnych
- Hermetyczna obudowa o klasie szczelności IP65 (dławiki o klasie IP68)
- Wymiary zewnętrzne (bez dławików): 64x58x32
- Adresowanie sprzętowe (0-15) lub programowe (1-247)

## 1.2 Interfejsy komunikacyjne

- Interfejs RS485, długość 1200m, maksymalnie 32 urządzenia na linii

Urządzenia ELECSO PRC-RH, są urządzeniami programowalnymi, ich oprogramowanie (firmware) może być łatwo wymieniony na nowszy, ulepszony, gwarantujący lepszą pracę urządzenia.

**Firmware – oprogramowanie wewnętrzne urządzenia, cechy**

- Możliwość konfiguracji poszczególnych modułów w systemie, bez konieczności otwierania urządzenia
- Możliwość automatycznej konfiguracji czujników, bez konieczności podgrzewania każdego z sensora z osobna., o ile sondy pomiarowe wykonane zostały w standardzie ELECSO.
- Możliwość konfiguracji manualnej sensorów, bez udziału PC

Moduły PRC-RH są urządzeniami rozwojowymi. Jesteśmy otwarci na sugestie w zakresie pracy modułów, dopasujemy działanie do indywidualnych potrzeb klienta.

### 1.3 Warunki użytkowania

Moduły pomiarowo-kontrolne ELECSO PRC dostarczane w hermetycznej obudowie, przeznaczone są do montażu wewnątrz oraz na zewnątrz budynku/magazynu. Montaż powinien uwzględniać:
- ochronę przed wysoką temperaturą, bezpośrednim działaniem deszczu i promieni słonecznych - zakres temperaturowy działania modułu: od -40ºC do +75ºC,
- stabilność montażu,
- ochronę przed ingerencją osób postronnych.

Wszelkie czynności związane z montażem, winny być wykonane przy pomocy odpowiedniego sprzętu, zaś okablowanie powinno być doprowadzone w sposób zapewniający bezpieczeństwo pod względem mechanicznym i elektrycznym. Zabrania się wykonywania instalacji z udziałem materiałów i urządzeń (okablowania, zasilaczy, układów i urządzeń pośredniczących) uszkodzonych, mogących być przyczyną ewentualnych zwarć i porażeń elektrycznych.

### 1.4 Gwarancje i serwis

Moduły ELECSO PRC-RH dostarczane są w kartonie ochronnym zapobiegającym przypadkowym uszkodzeniom wynikających z możliwych udarów podczas transportu. Przed wysyłką, każdy moduł jest gruntownie testowany. Firma ELECSO zapewnia darmowe aktualizacje oprogramowania urządzenia (firmware) oraz daje 5-letnią gwarancję na urządzenie.

Wady ujawnione w okresie gwarancji usuwane będą bezpłatnie przez firmę ELECSO w możliwie krótkim terminie, nieprzekraczającym 7 dni roboczych, licząc od daty przyjęcia sprzętu do naprawy. W przypadku konieczności sprowadzenia części zamiennych z zagranicy, terminy powyższe mogą ulec przedłużeniu, o czym ELECSO poinformuje reklamującego.

Wszelkie naprawy wykonywane będą w siedzibie firmy ELECSO, zaś koszt dostarczenia i odbioru pokrywa nabywca.

Gwarancja jest tracona w wyniku:
- uszkodzeń mechanicznych,
- uszkodzeń powstałych z nieprawidłowego użytkowania,
- uszkodzeń powstałych na skutek zdarzeń losowych jak wyładowania atmosferyczne.

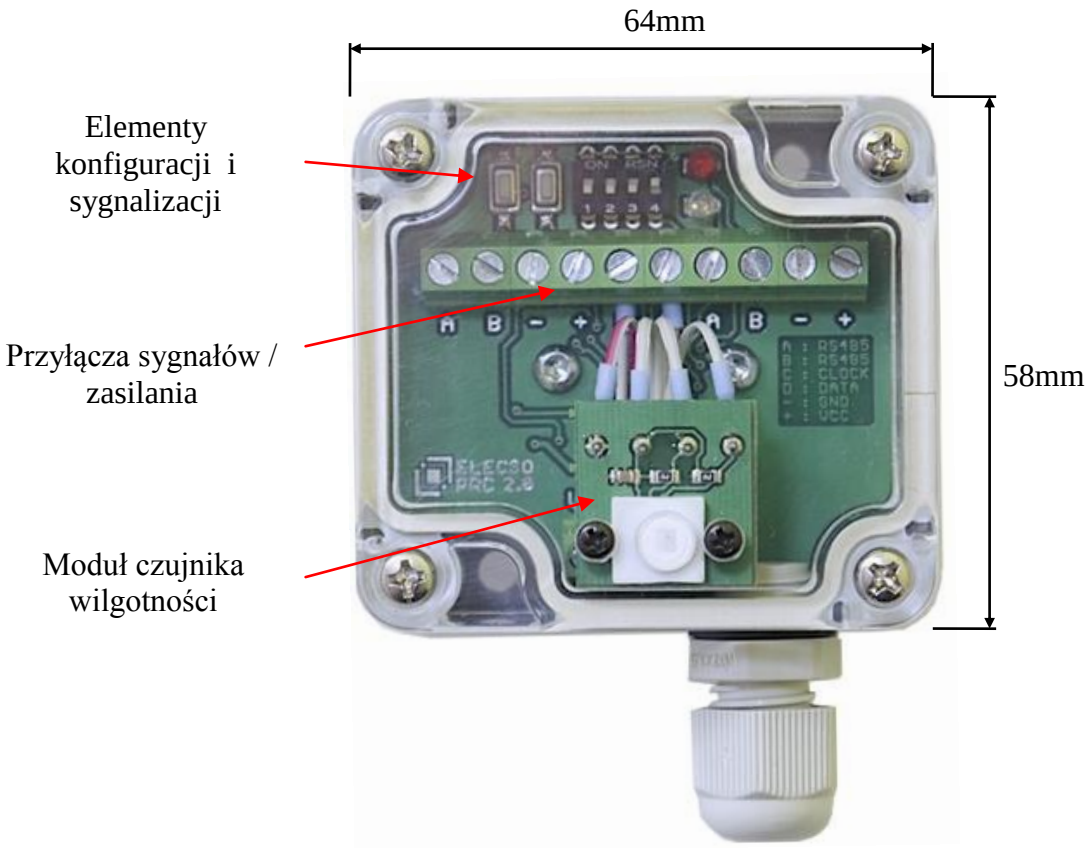
Nabywca traci wszelkie prawa wynikające z gwarancji, w przypadku stwierdzenia dokonywania nieautoryzowanych napraw lub zmian konstrukcyjnych urządzenia.

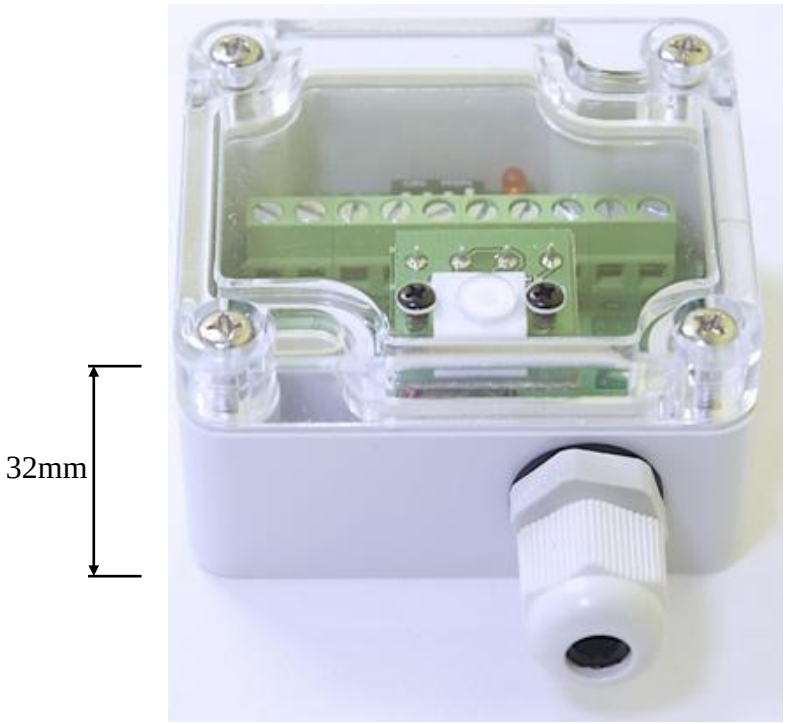
### 1.5 Certyfikaty i utylizacja

Deklaracja zgodności **CE** w zakresie użytych materiałów i elementów, oraz w zakresie kompatybilności elektromagnetycznej, wydana zostanie na życzenie nabywcy, po kontakcie z działem technicznym support@elecso.pl

Zużyte urządzenia ELECSO PRC podlegają utylizacji. Określone przez Ustawę o sprzęcie elektronicznym, zużyty moduł należy przekazać odpowiednim jednostkom lub zwrócić producentowi.

# 2. Opis elementów modułu ELECSO PRC-RH

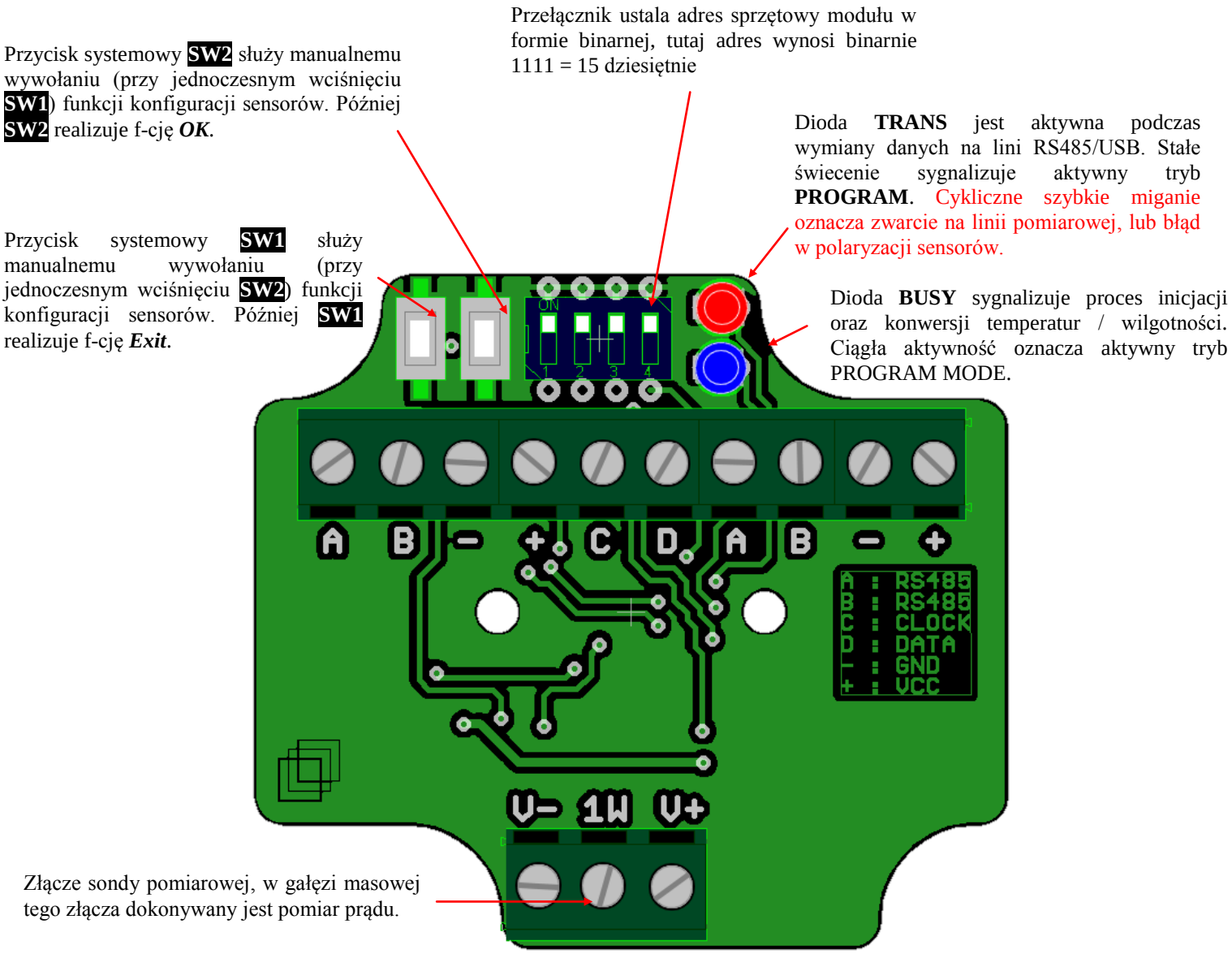


Poniżej przedstawiono moduł pomiarowo-kontrolny PRCv2.0. Zawarto objaśnienia dotyczące sygnałów wyprowadzonych na złącza modularne oraz objaśnienia kontrolek i elementów konfiguracyjnych.

Sygnał - znaczenie sygnału

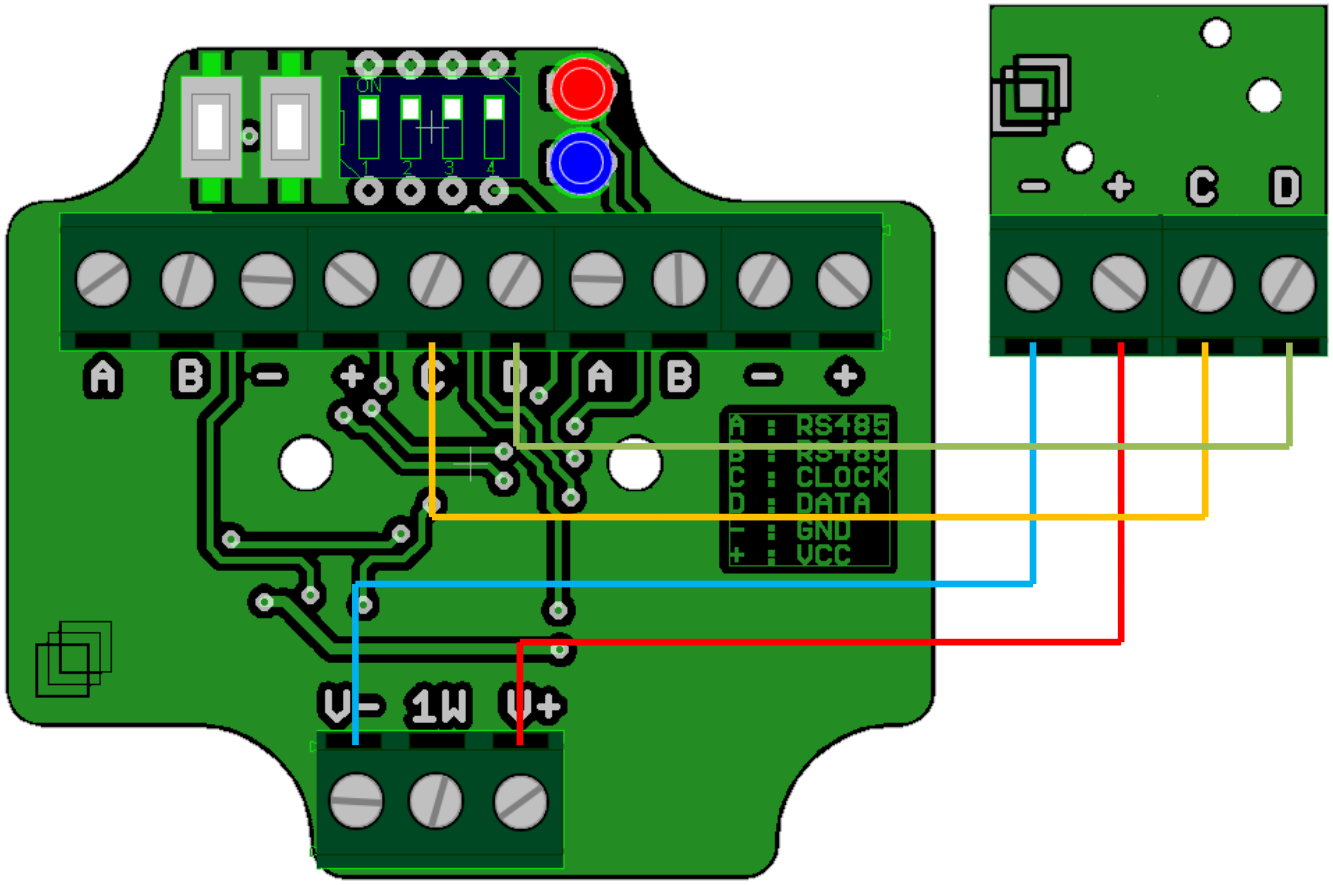
- \- masa zasilania (zdublowany)
- \+ plus zasilania (+12V) (zdublowany)
- **A, B** – zdublowane sygnały A i B interfejsu RS485 dwuprzewodowego (Half Duplex)
- **C, D** – sygnały interfejsu SBUS pomiaru wilgotności powietrza
- 1W – linia sygnałowa sond sensorów serii DS18B20
- **SW1, SW2** – przyciski systemowe
- **DSW –** switch adresu sprzętowego




Rys.1 Widok modułu PRCv2.0

### 2.1 Moduł pomiaru wilgotności SHTv1.0 – schemat połączenia do PRCv2.0

Moduł pomiarowy PRC-RH dostarczany jest z zamontowanym sensorem wilgotności SHT. Poniżej znajduje się schemat podłączeń sygnałów sensora ze złączem modułu PRC.

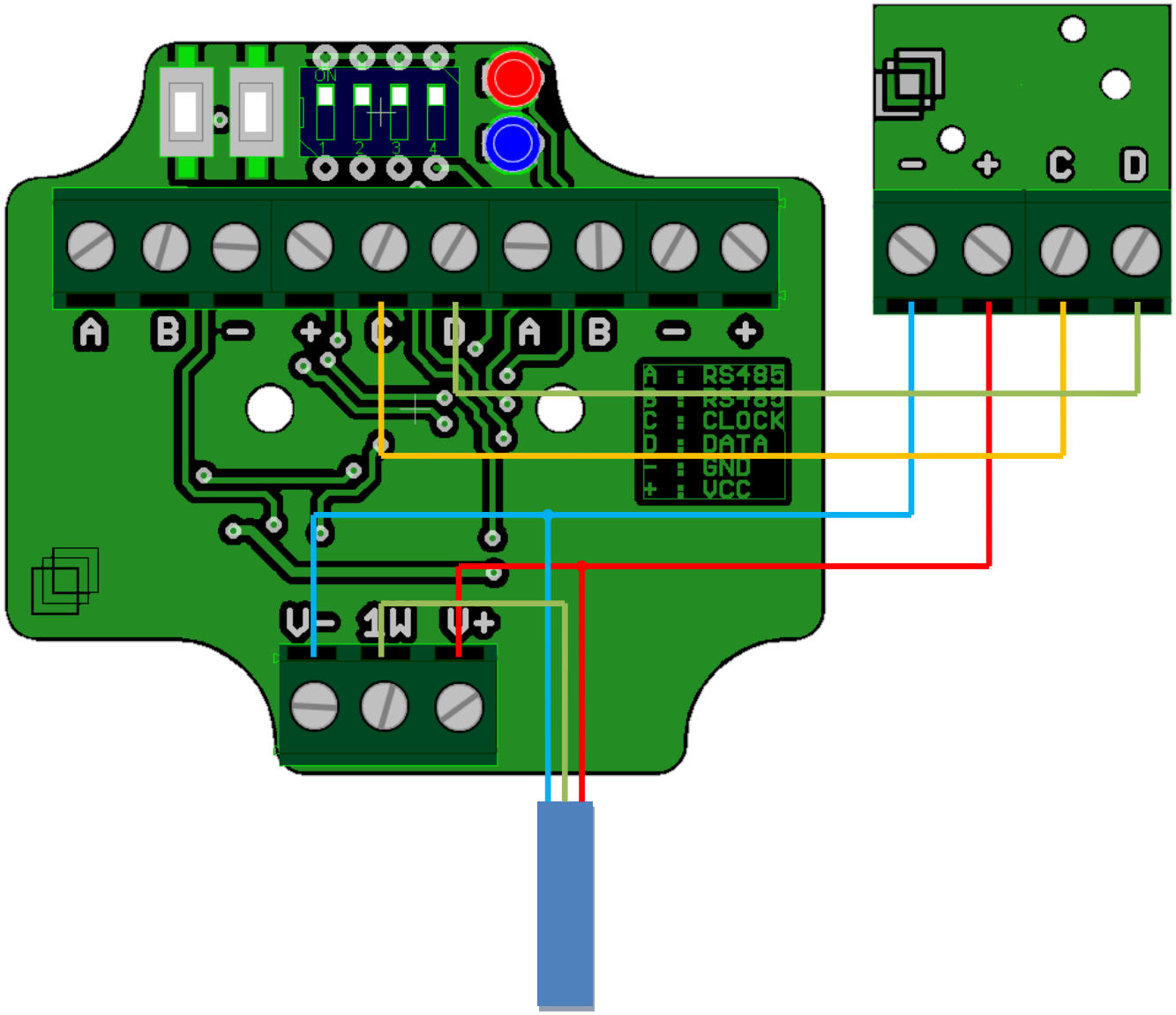



Rys.2 Schemat połączeń modułu pomiaru wilgotności SHT.

| Opis sgnału | Moduł PRC | Moduł SHT |
|---|---|---|
| Zasilanie sensora - | V- | - |
| Zasilanie sensora + | V+ | + |
| Sygnał CLK | C | C |
| Sygnał DATA | D | D |

### 2.1 Wielopunktowy pomiar temperatury z systemem pomiaru wilgotności – schemat połączeń.

Moduł pomiarowy PRC-RH prócz pomiaru wilgotności może dokonywać wielopunktowego pomiaru temperatury. Dla przykładu można dokonać pomiaru temperatury zewnętrznej w kilku punktach: przy gruncie, w miejscu zacienionym, nasłonecznionym itp. Moduł PRC obsługuje do 16 czujników DS18B20 oraz 1 sensor pomiaru wilgotności.




Rys.3 Schemat połączeń modułu pomiaru wilgotności SHT oraz sondy czujników temperatury.

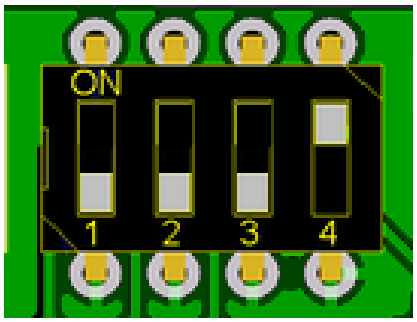
| Opis sygnału | Moduł PRC | Moduł SHT | Sonda temperatury |
|---|---|---|---|
| Zasilanie sensora - | V- | - | Przewód niebieski |
| Zasilanie sensora + | V+ | + | Przewód czerwony |
| Sygnał CLK | C | C | |
| Sygnał DATA | D | D | |
| Sygnał 1-Wire | 1W | | Przewód zielony |

# 3. Konfiguracja sprzętowa modułu PRC-RH

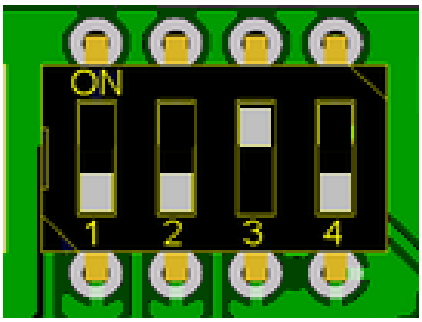
Moduły pomiarowe PRC zawsze dostarczane są z następującymi ustawieniami:
- sprzętowy tryb adresowania (konfiguracja za pomocą przełącznika DIPSwitch (patrz Rys.1)
- ustawiony sprzętowy adres: 1
- ustawiony programowy adres: 2
- ustawiony interwał pomiarowy modułu: 4s
- wyzerowana pamięć adresów sensorów.

### 3.1 Manualna zmiana wartości adresu sprzętowego.

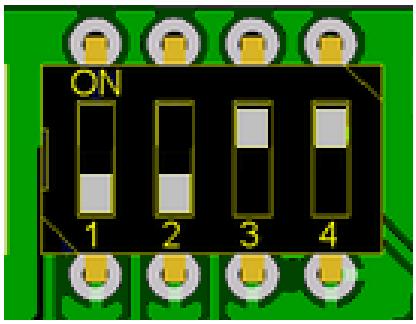
Adres sprzętowy ustawia się za pomocą 4-ro bitowego przełącznika DIPSwitch. Wartość zmienia się traktując poszczególne sekcje jako młodszą część bajtu. Poniżej, dla przykładu umieszczono konfigurację adresu, kolejno: 1 (binarnie: 0001) , 2 (binarnie: 0010) i 3 (binarnie: 0011).




Adres : 1




Adres : 2




Adres : 3

W ten sposób możliwa konfiguracja następujących adresów:

| Ustawienie przełącznika adresu DIPSwitch (binarnie) | Wartość adresu (dziesiętnie) |
|---|---|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | 10 |
| 1011 | 11 |
| 1100 | 12 |
| 1101 | 13 |
| 1110 | 14 |
| 1111 | 15 |

### 3.2 Manualne wyszukanie i segregacja sensorów.

Omawiana funkcja działa dla sond wykonanych w standardzie ELECSO (przy udziale modułu SGT0.1 lub 2.0). Wynikiem operacji jest zapis do pamięci modułu PRC wszystkich sensorów wchodzących w skład podłączonej sondy, przy czym sensory zostają posegregowane sposób chronologiczny – od najwyżej do najniżej zamontowanego w sondzie. Operacja zabiera krótką chwilę, zaś cały tak wykonany system gwarantuje sprawną i bezproblemową obsługę. Wywołanie i obsługę procesu wykonujemy przyciskami ***SW1*** i ***SW2*** (patrz Rys.1).

Poniżej przedstawiono w punktach przebieg całego procesu wyszukania i segregacji sensorów w kanale.

1. Wciśnij jednocześnie ***SW1*** i ***SW2*** przez 2 sekundy.
*Wywołanie funkcji* (rezultat końcowy LED: dioda niebieska świeci, czerwona pulsuje co 1 sekundę)

2. Mamy 2 możliwości:

2a. Wciśniecie ***SW1*** – wyjście z procedury.
*Moduł będzie pracował ze starymi ustawieniami* (rezultat końcowy LED: wykonywanie programu głównego)

2b. Wciśnięcie ***SW2*** – wyszukanie sensorów i segregacja.
*Moduł zgasi niebieski LED, zaś czerwony mrugnie tyle razy, ile znalazł sensorów…*
(rezultat końcowy LED: dioda niebieska mruga 2 razy na sekundę, czerwona zgaszona)

3. Mamy 2 możliwości:

3a. Wciśniecie ***SW1*** – powrót do punktu 2.

3b. Wciśnięcie ***SW2*** – zapis do trwałej pamięci modułu wyników wyszukania sensorów i segregacji.
*Moduł zapali niebieski oraz czerwony LED na czas zapisu. Całość zakończy się zresetowaniem modułu i powrotem do programu głównego. Proces zakończony Wyszukania i segregacji sensorów – zakończony.*

# 4. Schemat połączeń systemu pomiarowego

W załączniku  przedstawiony został schemat systemu pomiarowego.

# 5. Interfejs RS485 dwuprzewodowy (Half Duplex)

### 5.1 Parametry transmisji

- Prędkość transmisji: 9600
- Rozmiar danych: 8
- Parzystość:  bez kontroli parzystości
- Bit stopu: 1 bit

# 6. Protokół MODBUS RTU

Najczęściej stosowaną i używaną konfiguracją w automatyce przemysłowej jest protokół MODBUS współpracujący z interfejsem RS485, gdzie występuje jedno urządzenie nadrzędne (Master) inicjalizujące transakcje (wysyłające polecenie), natomiast pozostałe urządzenia są podrzędne (Slaves), wykonują polecenia Master-a i odsyłają odpowiedź. W danej chwili tylko jeden Slave może odpowiadać na zdalne zapytanie Master-a, natomiast nie ma możliwości komunikacji pomiędzy urządzeniami podrzędnymi. Typowym Masterem jest urządzenie z procesorem głównym (host procesor), zawierające programowalny panel na przykład komputer PC lub nadrzędny sterownik logiczny, a typowy Slave to programowalny sterownik logiczny. Węzły podrzędne (Slaves) są wykorzystywane do sterowania oraz zbierania danych z urządzeń peryferyjnych takich jak: mierników, liczników, przetworników A/C i C/A, czujników, przekaźników, sygnalizatorów itp. Jako interfejs komunikacyjny dla protokołu MODBUS zastosowano magistralę RS485, pozwalającą pracować w warunkach silnych zakłóceń (np. w przemyśle) oraz na uzyskanie znacznych zasięgów transmisji.

Główne zalety protokołu MODBUS to:

- prostota zastosowanych w nim rozwiązań;
- jawność specyfikacji protokołu;
- zabezpieczenie przesyłanych komunikatów przed błędami;
- potwierdzanie wykonania rozkazów zdalnych i sygnalizacja błędów;
- stały format ramki i zestaw standardowych funkcji służących wymianie danych,
- mechanizmy zabezpieczające przed zawieszeniem systemu.

### 6.1 Ramka transmisji

Ramka protokołu MODBUS określa format przesyłanych wiadomości i zawiera: adres odbiorcy, kod funkcji reprezentujący żądane polecenie, dane dotyczące funkcji oraz słowo kontrolne zabezpieczające przesyłaną wiadomość. Postać ramki zapytania wysyłanego przez jednostkę Master i ramki odpowiedzi jednostki podrzędnej Slave jest podobna. Różnica polega na tym, że w polu danych ramki odpowiedzi występują dane, których dostarczenia żądała jednostka Master (PC lub sterownik).

| Adres [1B] | Kod polecenia [1B] | Dane[x Bajtów] | Suma kontrolna [2 Bajty] |
|---|---|---|---|

Rys. 6 Budowa ramki transmisyjnej MODBUS

Opis poszczególnych pól ramki:
- **Adres** SLAVE: liczba z zakresu 1 – 247, 0 – adres rozgłoszeniowy;
- **Kod polecenia**: jest liczbą z zakresu 1...127;
- Pole **Danych**: jego długość zależy od rodzaju wiadomości i może zawierać:
  - w przypadku zapytania – argumenty funkcji;
  - w przypadku pozytywnej odpowiedzi - argumenty funkcji;
  - w przypadku szczególnej odpowiedzi - kod błędu;
  - w niektórych przypadkach może być równa 0;
- **Suma kontrolna**: wyznaczana z zawartości przesyłanego komunikatu. W protokole MODBUS RTU jako zabezpieczenie ramki wiadomości stosuje się sumę kontrolną CRC16 [CRCL, CRCH). Jej wartość wyznacza urządzenie nadające dla zawartości przesyłanego komunikatu i umieszcza w ramce po części informacyjnej.

Węzeł odbiorczy oblicza sumę kontrolną dla odebranego komunikatu i porównuje jej wartość z wartością otrzymaną. Niezgodność sum świadczy o wystąpieniu błędu.

W węźle nadrzędnym ustalany jest pewien maksymalny czas oczekiwania na odbiór ramki zawierającej komunikat z odpowiedzią (Timeout). Jego wartość musi być tak dobrana, aby nawet najwolniejszy z węzłów podrzędnych zdążył odesłać odpowiedź. Przekroczenie tego czasu powoduje przerwanie transakcji. Przyczyn braku odpowiedzi może być kilka.

Najczęstszymi przyczynami braku odpowiedzi są:

- wystąpienie błędu transmisji ramki polecenia,
- zaadresowanie nieistniejącego węzła podrzędnego.

Spis błędów i ich opisy znajdziecie Państwo w rozdziale 6 - *Spis błędów ich opisy i sposoby usuwania*.

Typowe zachowanie urządzenia podrzędnego Slave, po odbiorze zapytania i zweryfikowaniu adresu może wyglądać następująco:

- w przypadku poprawnego odbioru i bezbłędnej interpretacji ramki zapytania wysyłanej przez Master odpowiada również zgodnie z formatem zdefiniowanym w protokole MODBUS,

- gdy wystąpi błąd przy odbiorze wiadomości lub Slave nie jest w stanie wykonać polecenia wysyła tzw. szczególną odpowiedź (Exception Response), w ramce odpowiedzi wysyłanej przez Slave w polu kodu funkcji ustawiany jest najstarszy bit, natomiast w polu danych umieszczany jest kod błędu, umożliwiający węzłowi nadrzędnemu określenie przyczyny jego wystąpienia.

Dla urządzeń ELECSO PRC odpowiedź w takim przypadku wygląda następująco:

| **Dla wartości adresu spoza zakresu** | `02 83 02 30 F1` |
|---|---|

Gdzie:

`0x02` - adres pytanego urządzenia [1Bajt]
`0x83` – Sygnalizacja błędu [1Bajt]
`0x02` – Rodzaj błędu, w tym przypadku wartości adresu spoza zakresu [1Bajt]
`0x30F1` - suma CRC16 pakietu zapytania [2Bajty: CRCL,CRCH]

| **Dla wartości danych spoza zakresu** | `02 83 03 30 F1` |
|---|---|

Gdzie:

`0x02` - adres pytanego urządzenia [1Bajt]
`0x83` – Sygnalizacja błędu [1Bajt]
`0x03` – Rodzaj błędu, w tym przypadku wartości danej spoza zakresu [1Bajt]
`0xF131` - suma CRC16 pakietu zapytania [2Bajty: CRCL,CRCH]

- w przypadku błędu sumy kontrolnej lub innych błędów w ramce zapytania nie zdefiniowanych w kodach błędów, układ podrzędny Slave nie wysyła odpowiedzi.

### 6.2 Komendy i Specyfika odpowiedzi MODBUS w modułach ELECSO PRC.

Urządzenia ELECSO PRC komunikują się z systemem za pomocą komend zgodnych z MODBUS RTU. Układ nadrzędny Master, chcąc wywołać odpowiednią reakcję modułu (np. pobrać wyniki konwersji temperatury) wysyła komendę konkretnemu układowi. Identyfikacja modułów następuje po ich indywidualnych adresach. Ramka transmisyjna została przedstawiona na Rys.6.

W chwilę pisania niniejszej instrukcji, aktualny firmware urządzenia (wersja: 00.02) w zakresie komend MODBUS umożliwiał:
- Odczyt wartości temperatur
- Odczyt wartości napięć zasilania i systemowych oraz natężeń prądów płynących w kanałach pomiarowych

Objaśnienia każdej transakcji omówione zostały na przykładach dla modułu ELECSO PRC o adresie **02.**

#### 6.2.1Odczyt wartości temperatur (komenda nr 03)

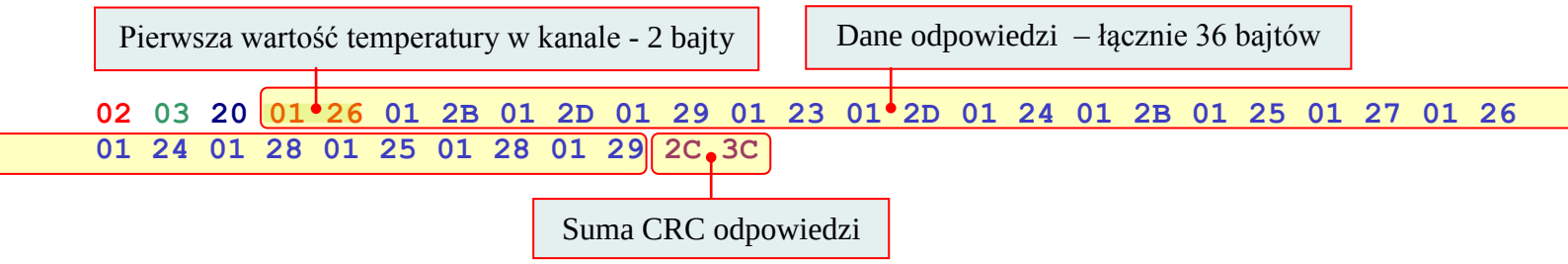
Komenda układu nadrzędnego Master, przedstawiona w systemie szesnastkowym, ma postać:

```
02 03 00 00 00 10 44 35
```

gdzie kolejno:

0x**02** - adres pytanego urządzenia [1Bajt]
0x**03** - kod polecenia pobrania wyników konwersji temperatury [1Bajt]
0x**0000** - adres początkowy tablicy z danymi, możliwy zakres od 0x0000 do 0x000F (od 0 do 15) [2Bajty]
0x**0010** – ilość danych (ilość sensorów), możliwy zakres od 0x0000 do 0x0010 (od 0 do 16) [2Bajty]
0x**4435** - suma CRC16 pakietu zapytania liczona z pierwszych 6 bajtów komendy [2Bajty: CRCL,CRCH]

Odpowiedź modułu ELECSO PRC Slave:




0x**02** - adres pytanego urządzenia [1Bajt]
0x**03** - kod polecenia pobrania wyników konwersji [1Bajt]
0x**CA** - ilość bajtów danych odpowiedzi, w tym przypadku 0xCA (dziesiętnie 202) [1Bajt]
0x**0000**… **01E5**– dane, w tym przypadku 202 bajty
0x**A9A5** - suma CRC16 pakietu odpowiedzi [2Bajty: CRCL,CRCH]

Szerszego omówienia wymaga blok **Danych** odpowiedzi, który zawiera wyniki konwersji temperatur. Wartość temperatury każdego sensora reprezentowana jest dwubajtowym słowem ***T.HI*** i ***T.LO***. W powyższym przykładzie temperatura najwyżej zamontowanego sensora reprezentowana jest słowem 0x0126, sensor poniżej zamontowany – 0x012B. Ostatni (16) najniższy pomiar reprezentowany jest słowem heksalnym: 0x0129.

Formuła przeliczająca wartości przekazywane przez moduł na skalę Celsjusza jest następująca:

$$T_C = \frac{(T.HI \times 256 + T.LO) \times 125}{2000}$$

Dla powyższego przykładu, temperatura pierwszego sensora w sondzie z kanału będzie wynosić:
T.HI = 0x01 = 1 dziesiętnie
T.LO = 0x26 = 38 dziesiętnie

$$T_C = \frac{(1\times 256+38)\times 125}{2000} = 18{,}375^{o}C$$

Funkcja przeliczająca wartości oraz uwzględniające wartości ujemnie, w języku C może wyglądać następująco:

```
float Konwertuj_Temp(unsigned char Temp_H, unsigned char Temp_L)
{
    float Wynik = 0;
    unsigned char uc_1 = 0;
    unsigned int ui_1 = 0; unsigned int ui_2 = 0; unsigned int ui_3 = 0;
    //--------------------------------------------------------------------

     uc_1 = Temp_H & 0xF0;

     if(uc_1 != 0xF0) // dla dodatniej wartości temperatury
     {
        ui_1 = Temp_H;
        ui_1 = (ui_1<<8) + Temp_L;
        Wynik = ui_1;
        Wynik = (Wynik*125)/2000;
     }
     else  // dla ujemnej wartości temperatury
        {
           ui_3 = Temp_H;
           ui_3 = (ui_3<<8) + Temp_L;
           ui_2 = 0xFFFF - ui_3 + 1;
           Wynik = ui_2;
           Wynik = (-1)*(Wynik*125)/2000;
        }
     return(Wynik);
}
```

W razie niejasności lub problemów prosimy o kontakt lub odsyłamy do noty aplikacyjnej układu DS18B20.

### 6.2.2 Odczyt wartości temperatury sensora SHT i wilgotności powietrza (komenda nr 04)

Komenda układu nadrzędnego Master, przedstawiona w systemie szesnastkowym, ma postać:

**02 04 00 00 00 02 71 F8**

gdzie kolejno:

0x**02** - adres pytanego urządzenia [1Bajt]
0x**04** - kod polecenia pobrania temperatury i wilgotności [1Bajt]
0x**0000** - adres początkowy tablicy z danymi, możliwy zakres od 0x0000 do 0x0002 (od 0 do 2) [2Bajty]
0x**0002** – ilość danych (ilość wielkości), możliwy zakres od 0x0001 do 0x0002 (od 1 do 2) [2Bajty]
0x**71F8** - suma CRC16 pakietu zapytania liczona z pierwszych 6 bajtów komendy [2Bajty: CRCL,CRCH]

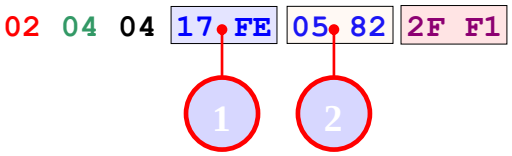
Odpowiedź modułu ELECSO PRC Slave:

**02 04 04 17 FE 05 82 2F F1**

gdzie kolejno:

0x**02** - adres pytanego urządzenia [1Bajt]
0x**04** - kod polecenia pobrania wyników konwersji przetworników analogowo-cyfrowych [1Bajt]
0x**04** - ilość bajtów danych odpowiedzi, w tym przypadku 0x04 (dziesiętnie 4) [1Bajt]
0x**17FE,0582** – dane, w tym przypadku 4 bajty
0x**2FF1** - suma CRC16 pakietu odpowiedzi [2Bajty: CRCL,CRCH]

Omówienie Bloku Danych odpowiedzi




Wyniki konwersji reprezentują wartości zapisane dwubajtowo

| Adres | Kanał, ilość danych |
|---|---|
| **0…1 [Hi/Lo]** | **(1)** temperatura sensora wilgotności |
| **2…3 [Hi/Lo]** | **(2)** wilgotność względna powietrza |

Formuła przeliczająca wartości przekazywane przez moduł na temperaturę sensora wilgotności:

$$T = -40.1 + 0.01 * Conv$$

Dla powyższych danych:
Conv = 0x17FE = 6142 dziesiętnie

$$T = -40.1 + 0.01 * 6142 = 21{,}32\,'C$$

Funkcja w C może wyglądać następująco:

```
float Przelicz_T_SHT(unsigned char Byte_Hi, unsigned char Byte_Lo)
{
      unsigned int ui_1 = 0;
      float Wynik = 0;

      //-----------------

      ui_1 = Byte_Hi;
      ui_1 = ui_1<<8;
      ui_1 = ui_1 + Byte_Lo;

      Wynik = -40,1 + (0,01*ui_1);

      return(Wynik);
}
```

Formuła przeliczająca wartości przekazywane przez moduł na wilgotność względną jest następująca:

$$RH = -2.0458 + (0.0367 * CONV) + ((-1.5955 / 1000000) * CONV^2)$$

Dla powyższych danych:
Conv = 0x0582 =1410 dziesiętnie

$$RH = -2.0458 + (0.0367 * 1410) + ((-1.5955 / 1000000) * 1410^2) = 46{,}53\%$$

Funkcja w C może wyglądać następująco:

```
float Przelicz_RH(unsigned char Byte_Hi, unsigned char Byte_Lo)
{
     unsigned int ui_1 = 0;
     float Wynik = 0;

     //-----------------

     ui_1 = Byte_Hi;
     ui_1 = ui_1<<8;
     ui_1 = ui_1 + Byte_Lo;

     Wynik =  -2.0458+(0.0367*CONV)+((-1.5955/1000000)*CONV*CONV);

     return(Wynik);
}
```

### 6.2.3 Odczyt wartości napięć zasilania oraz natężeń prądów (komenda nr 06)

Komenda układu nadrzędnego Master, przedstawiona w systemie szesnastkowym, ma postać:

**02 06 00 00 00 07 C8 3B**

gdzie kolejno:

0x**02** - adres pytanego urządzenia [1Bajt]
0x**06** - kod polecenia pobrania wyników konwersji przetworników analogowo-cyfrowych [1Bajt]
0x**0000** - adres początkowy tablicy z danymi, możliwy zakres od 0x0000 do 0x0007 (od 0 do 7) [2Bajty]
0x**0007** – ilość danych (ilość wielkości), możliwy zakres od 0x0000 do 0x0007 (od 1 do 7) [2Bajty]
0x**C83B** - suma CRC16 pakietu zapytania liczona z pierwszych 6 bajtów komendy [2Bajty: CRCL,CRCH]

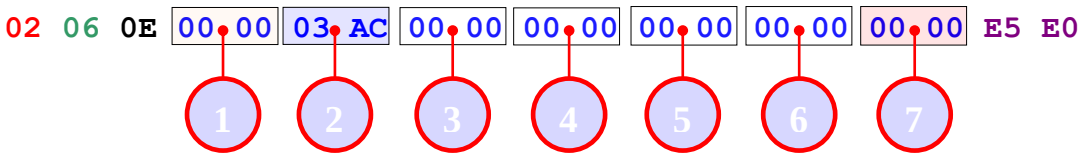
Odpowiedź modułu ELECSO PRC Slave:

**02 06 0E 00 00 03 AC 00 00 00 00 00 00 00 00 00 00 E5 E0**

gdzie kolejno:

0x**02** - adres pytanego urządzenia [1Bajt]
0x**06** - kod polecenia pobrania wyników konwersji przetworników analogowo-cyfrowych [1Bajt]
0x**0E** - ilość bajtów danych odpowiedzi, w tym przypadku 0x0E (dziesiętnie 14) [1Bajt]
0x**0000,03AB**... **0000** – dane, w tym przypadku 14 bajtów
0x**E5E0** - suma CRC16 pakietu odpowiedzi [2Bajty: CRCL,CRCH]

Omówienie Bloku Danych odpowiedzi




Wyniki konwersji reprezentują wartości zapisane dwubajtowo

| Adres | Kanał, ilość danych |
|---|---|
| **0…1 [Hi/Lo]** | **(1)** Zarezerwowane… |
| **2…3 [Hi/Lo]** | **(2)** Wynik pomiaru napięcia systemowego +5V |
| **4…5 [Hi/Lo]** | **(3)** Zarezerwowane… |
| **6…7 [Hi/Lo]** | **(4)** Zarezerwowane… |
| **8…9 [Hi/Lo]** | **(5)** Zarezerwowane… |
| **10…11 [Hi/Lo]** | **(6)** Zarezerwowane… |
| **12…13 [Hi/Lo]** | **(7)** Wynik pomiaru natężenia prądu w gałęzi masy sondy pomiarowej |

Formuła przeliczająca wartości przekazywane przez moduł na napięcie systemowe jest następująca:

$$U_{5V} = \left( \frac{(B.HI \times 256 + B.LO) \times 1.1}{1024} \right) \times 5$$

Dla powyższych danych:
B.HI = 0x03 = 3 dziesiętnie
B.LO = 0xAC = 172 dziesiętnie

$$U_{5V} = \left( \frac{(3 \times 256 + 172) \times 1.1}{1024} \right) \times 5 = 5.048V$$

Funkcja w C może wyglądać następująco:

```
float Przelicz_U5V(unsigned char Byte_Hi, unsigned char Byte_Lo)
{
      unsigned int ui_1 = 0;
      float Wynik = 0;

      //-----------------

      ui_1 = Byte_Hi;
      ui_1 = ui_1<<8;
      ui_1 = ui_1 + Byte_Lo;

      Wynik = ((ui_1*1.1)/1024)*5;

      return(Wynik);
}
```

Formuła przeliczająca wartości przekazywane przez moduł na natężenie prądu jest następująca:

$$I = \left( \frac{(B.HI \times 256 + B.LO) \times 1.1}{1024} \right) \div 2.7$$

Funkcja w C może wyglądać następująco:

```
float Przelicz_Prad(unsigned char Byte_Hi, unsigned char Byte_Lo)
{
      unsigned int ui_1 = 0;
      float Wynik = 0;

      //-----------------

      ui_1 = Byte_Hi;
      ui_1 = ui_1<<8;
      ui_1 = ui_1 + Byte_Lo;

      Wynik =((us_1*1.1)/1024)/2.7;

      return(Wynik);
}
```