

moduł WIRE-CHIP H3.1

oprogramowanie S3.12

Instrukcja obsługi



Spis treści

Opis.....	2
Dane techniczne oprogramowania:.....	2
Zmiany w porównaniu z wersją s2.16.....	3
Mapa rejestrów MODBUS.....	4
Odczyt wartości temperatury.....	5
Podstawowa konfiguracja modułu.....	5
Użycie programu „wire-konfigurator”.....	6
Usuwanie czujników	8
Dodawanie czujników	8
Ustawianie kolejności czujników	9
Monitorowanie czujników.....	9
MAKROINSTRUKCJE.....	10
Edycja makroinstrukcji.....	11
dodawanie i zamiana.....	11
usuwanie.....	11
Lista dostępnych makroinstrukcji.....	11
Szczegółowy opis makroinstrukcji.....	12
MODB_RHR - Read Holding Registers.....	12
MODB_PHR - Preset Holding Registers.....	12
MODB_RC - Read Coils.....	12
MODB_FC - Force Coils.....	12
BIT_PULS0, BIT_PULS1, BIT_PULS2, BIT_PULS3	13
BIT_OUT0, BIT_OUT1, BIT_OUT2, BIT_OUT3.....	13
BIT_ZERO.....	13
BITY_ZERO	13
BITY_SET	13
BIT_SUM.....	13
BIT_AND.....	14
BIT_NEG.....	14
CMP	15
CMP_HIST_C	15
CMP_HIST_I	15
CMP_BC	15
CMP_AC	15
Producent.....	16

Opis

Moduł WIRE-CHIP jest bardzo zaawansowanym technologicznie urządzeniem służącym głównie do wielopunktowego pomiaru temperatury z komunikacją MODBUS RTU. Oprócz podstawowej funkcjonalności moduł może przy pomocy makroinstrukcji zarządzać portami szeregowymi, a także wykonywać operacje na zgromadzonych danych i sterować urządzeniami. W zależności od użytych makroinstrukcji, moduł może:

- wysyłać i odbierać dane do/z innych urządzeń,
- pracować jako wielokrotny termostat (kilkadziesiąt termostatów w jednym module),
- pracować w systemach monitoringu i wywoływać alarmy po przekroczeniu wartości progowych (bez nadzoru komputera lub sterownika PLC),

Moduł może pracować jako konwerter RS-232 – RS-485 w trybie SLAVE.

Dane techniczne oprogramowania:

- pomiary temperatury (czujniki firmy Dallas Semiconductor):
 - maksymalna ilość czujników: **64 szt.**
 - format i wartości danych pomiarowych – 2 miejsca po przecinku,
 - wartość błędu czujnika = -20000 (-200°C),
 - wartość dla niezainstalowanego czujnika = -30200 (-302°C),
 - zakres pomiarowy: **od -55°C do +125°C**,
 - czas konwersji temperatury: **750 ms** (nie ma znaczenia szybkość transmisji po 1-wire)
 - **maksymalny błąd pomiaru: ± 0,5°C** (w zakresie od -10°C do +80°C)
 - rozdzielczość pomiaru:
 - czujnik typu **DS18B20** (polecane) - $1/16^{\circ}\text{C} = \mathbf{0,0625^{\circ}\text{C}}$
 - czujnik typu DS18S20, DS1820 - rozdzielczość pomiaru: 0,5°C
- komunikacja (oprócz 1-wire):
 - wbudowane porty: RS-232, RS-485,
 - **pracuje jako konwerter RS-232 – RS-485**, (funkcja programu firmware),
 - do konfiguracji przez komputer PC nie wymaga konwertera,
 - szybkości transmisji: 9600, 19200, 38400, 57600, 115200 bps
 - format znaku: 8N1 (8-bitów danych, brak kontroli parzystości, 1-bit stop)
 - **protokół transmisji: MODBUS RTU – MASTER lub SLAVE**
- konfiguracja: (bezpłatne) oprogramowanie konfiguracyjne,
- wyjścia cyfrowe – 3 szt – sposób sterowania
 - ciągły – do sterowania urządzeniami (przez przekaźniki / styczniki) (grzałki, lampy, silniki, pompy, nawilżacze, dmuchawy, piece)
 - impulsowy (0,5s ON, 0,5s OFF) – do alarmowania (lampki, buzery, syreny, itp.).

Zmiany w porównaniu z wersją s2.16

Główne zmiany w porównaniu z poprzednimi wersjami (s2.xx):

1. pracuje jako SLAVE oraz **MASTER** (protokół MODBUS RTU),,
2. posiada bogaty zestaw makroinstrukcji,
3. wytworzony został program konfiguracyjny „wire-konfigurator”, dzięki czemu łatwo można przenosić ustawienia z jednego modułu na kolejny,
4. usunięty protokół „WŁASNY”.

Ad.1.

Użytkownik konfiguruje makroinstrukcje, za pomocą których WIRE-CHIP odczytuje lub zapisuje dane z własnej pamięci do zewnętrznego modułu SLAVE lub z zewnętrznego modułu SLAVE do własnej pamięci RAM.

Ad.2.

Oprócz funkcji sterujących pracą portu (w trybie MASTER) posiada także makroinstrukcje analizujące dane, sterujące wyjściami cyfrowymi

Ad.3.

Program komputerowy sprawia, że konfiguracja modułu WIRE-CHIP jest o wiele prostsza niż gdyby konfiguracja odbywała się przez tryb konfiguracyjny w oknie terminala (jak dotychczas – wersje firmware s2.xx).

Kolejne części instrukcji zawierają informacje:

Mapa rejestrów

Okno terminala

- uzyskanie informacji o module
- możliwości konfiguracji z poziomu terminala
- przejście do trybu „MODBUS CONFIG”

Program konfiguracyjny „wire-konfigurator”

- ustawienia podstawowe programu,
- odczyt zapis konfiguracji modułu WIRE-CHIP,
- karta wszystkie rejestry,
- karta ustawienia,
- karta czujniki,
- karta makroinstrukcje

Mapa rejestrów MODBUS

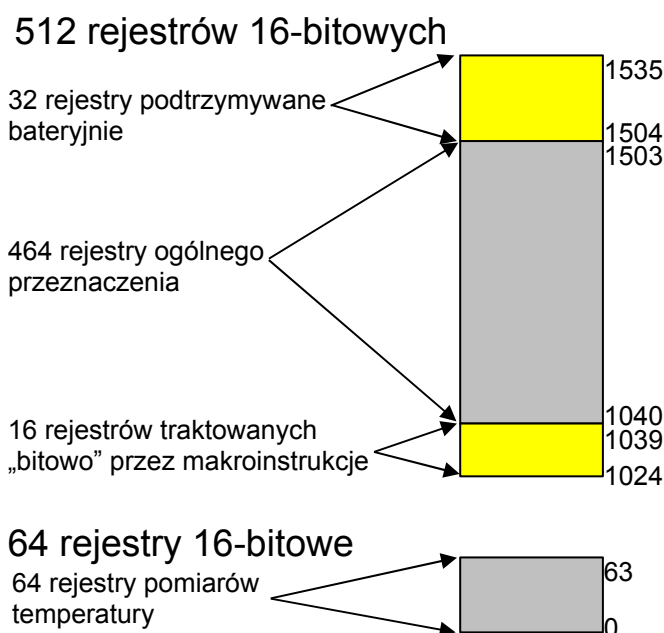
Mapa rejestrów (16-bitowych) modułu WIRE-CHIP (dla obecnej wersji firmware):
dostępne w czasie normalnej pracy:

- **0...63 = 0x0000 ... 0x003F** - mierzone wartości temperatury (z własnych czujników 1-wire DS18B20, DS18S20, DS1820),
- **1023 = 0x03FF** - status, tryb, komendy
- **1024..1535 = 0x0400.. 0x05FF** - pamięć RAM dowolnego użytku (przechowywanie danych tymczasowych).
 - **1024..1039** – rejestry traktowane „bitowo” (16szt) - ich bity używane są przez makroinstrukcje (bity 0..255),
 - **1504..1535** – zawartość tych 32-rejestrów jest **podtrzymywana bateryjnie**

dostępne w czasie konfiguracji poprzez „MODBUS CONFIG”:

- **0...63 = 0x0000 ... 0x003F** - mierzone wartości temperatury (z własnych czujników DS),
- **1023 = 0x03FF** - status, tryb, komendy,
- **1024..1279 = 0x0400.. 0x04FF** - numery seryjne czujników DS (na 1-WIRE)
- **1280 = 0x0500**
 - starszy bajt (MSB) = ilość czujników;
 - młodszy bajt (LSB) = własny ID dla Modbus,
- **1281 = 0x0501**
 - MSB = baudrate dla RS485 (0-9600 bps, 1-19200, 2-38400, 3-57600, 4-115200);
 - LSB=okres pomiaru (min 1s, max 99s)
- **1282..1535 = 0x0400.. 0x05FF** - pamięć kodów makroinstrukcji (508 Bajtów)

Mapa rejestrów dostępnych podczas normalnej pracy:



Odczyt wartości temperatury

Wartości mierzonej temperatury dostępne są w rejestrach 0..63 (holding registers – funkcja 03). Aby uzyskać wartość w stopniach Celsjusza [°C] należy odczytaną liczbę podzielić przez 100. Reszta z dzielenia (2 najmniej znaczące cyfry) są częścią po przecinku.

$$\text{Temperatura [°C]} = \text{zawartość rejestru} / 100$$

Zakres prawidłowych wartości to -55,00 ... +125,00.

Inne możliwe wartości:

-200,00 – błąd odczytu czujnika (brak spodziewanego czujnika lub błąd CRC)

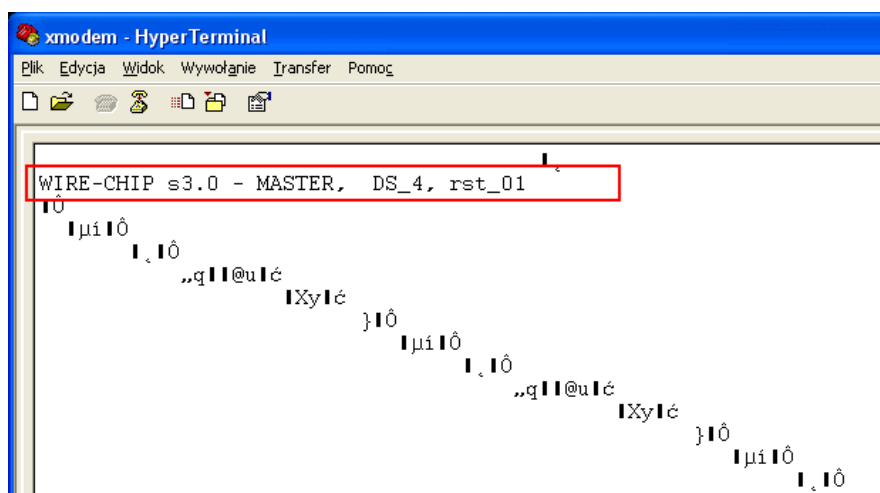
-302,00 – czujnik nie zainstalowany

Podstawowa konfiguracja modułu

Okno terminala

Konfigurację modułu należy rozpocząć od uruchomienia programu terminalowego np. HyperTerminal firmy Microsoft lub PCLite firmy Moxa.

1. Należy otworzyć port komunikacyjny, którym komputer PC jest połączony z modułem WIRE-CHIP.
2. Zresetować moduł WIRE-CHIP – jeśli komunikacja w kierunku do komputera jest poprawna – pojawi się komunikat zaczynający się od: **WIRE-CHIP** (s3.X – wersja oprogramowania, DS_X – ilość zainstalowanych czujników)



3. Przejść do trybu CONFIG (trzymając wciśnięty przycisk CONFIG, na chwilę wcisnąć przycisk RESET, przycisk CONFIG puścić po krótkim sygnale dźwiękowym od modułu). W oknie terminala pojawi się następujące menu trybu konfiguracyjnego:

```

xmodem male okno - HyperTerminal
Plik Edycja Widok Wywołanie Transfer Pomoc
WIRE-CHIP h3.1 bootloader
=====
WIRE-CHIP - config
-----

MENU:
n - numer modulu (ID) {0, 1, 2, ..., 255} |
b - baud rate {9600, 19200, 38400, 57600, 115200 bps}
t - tryb - SLAVE/MASTER/TX
m - zapisz i konfiguruj przez MODBUS
s - zapisz i wyjdź
x - anuluj i wyjdź
-

```

4. Należy ustawić parametry dostępne w menu konfiguracyjnym numer modułu (ID), baudrate (szybkość transmisji), tryb,
5. Aby móc skonfigurować moduł za pomocą oprogramowania wire-konfigurator, należy teraz wybrać opcję „m – zapisz i konfiguruj przez MODBUS” lub wcisnąć przycisk CONFIG na module. Zostanie wówczas wyświetlony komunikat „WIRE-CHIP s3... ID=... ..” oraz komunikat „MODBUS CONFIG”

```

xmodem male okno - HyperTerminal
Plik Edycja Widok Wywołanie Transfer Pomoc
WIRE-CHIP h3.1 bootloader
WIRE-CHIP s3.11- SLAVE, ID=222 DS_5, rst_10
MODBUS CONFIG

```

6. Należy wówczas wyłączyć program HyperTerminal (lub inny program terminalowy), aby nie blokował portu programowi wire-konfigurator.

Użycie programu „wire-konfigurator”

7. Uruchomić program wire-konfigurator.
8. W górnej części okna programu należy wybrać odpowiedni port COM (taki sam jak w programie terminalowym), szybkość transmisji, ID modułu do konfiguracji (w tym przypadku ID=222). Te ustawienia są zapamiętywane przy zamykaniu aplikacji i ponownie przywracane przy jej uruchamianiu.
9. Aby obejrzeć aktualną konfigurację modułu należy wcisnąć przycisk „**Czytaj z WIRE-CHIP**”, wówczas w karcie „wszystkie rejestry” pokaże się aktualna zawartość rejestrów konfiguracyjnych. W kolejnych kartach - „ustawienia”, „czujniki” i „makro instrukcje” znajdują się informacje wynikające z zawartości rejestrów pokazywanych w karcie „wszystkie rejestry”.

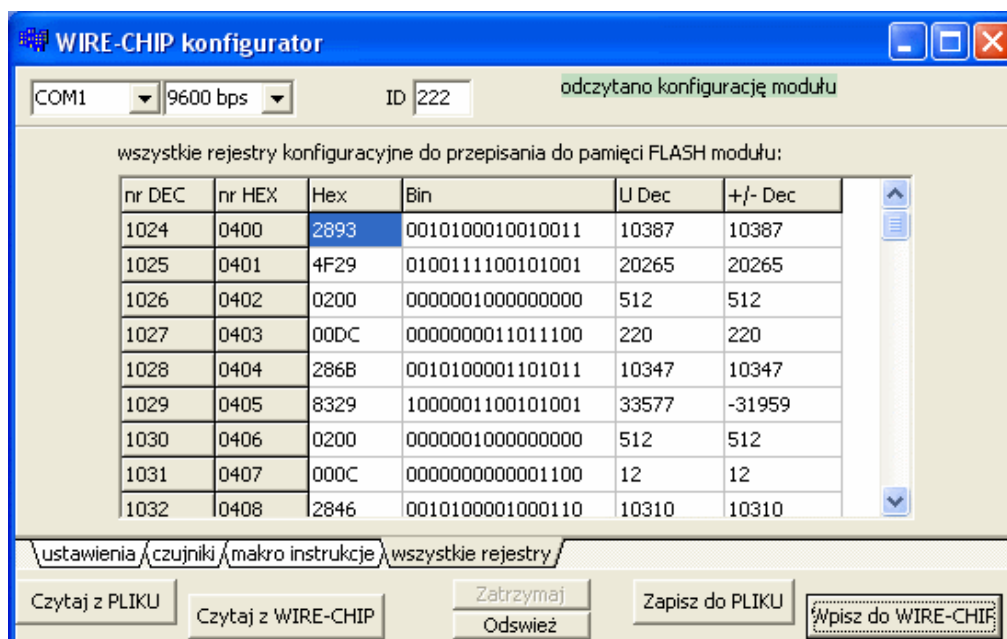
Aby zapisać konfigurację modułu w celu późniejszego jej wykorzystania, należy wcisnąć przycisk „**Zapisz do PLIKU**”.

Aby odczytać ustawienia konfiguracyjne z pliku należy wcisnąć przycisk „**Czytaj z PLIKU**”

Jeśli użytkownik zmodyfikował ustawienia konfiguracyjne i chce je wpisać do modułu, wówczas powinien wcisnąć przycisk „**Wpisz do WIRE-CHIP**”.

Karta „wszystkie rejestry” umożliwia edycję wartości rejestrów w kodzie szesnastkowym (Hex), binarnym (Bin), dziesiętnym bez znaku (U Dec) [0..65535], dziesiętnym ze znakiem (+/- Dec) [-32768..32767]. Numer rejestru pokazywany jest w postaci dziesiętnej (DEC), jak również szesnastkowej (HEX).

10. Po odczytaniu zawartości rejestrów modułu ekran wygląda podobnie jak na rysunku:



W prawym górnym rogu ekranu znajduje się informacja o poprawności wykonania komendy. Jeśli komenda zostanie wykonana niepoprawnie, komunikat będzie na to wskazywać.

Jeśli program nie będzie mógł otworzyć portu COM (błędny numer, zajęty przez inny program, awaria portu), wówczas nazwa portu podświetli się na **czerwono**.

Wszelkie zmiany ustawień odniosą skutek po zapisaniu ich do modułu (przycisk „Wpisz do WIRE-CHIP”) i zresetowaniu modułu (odłączyć i podłączyć zasilanie lub wcisnąć przycisk RESET).

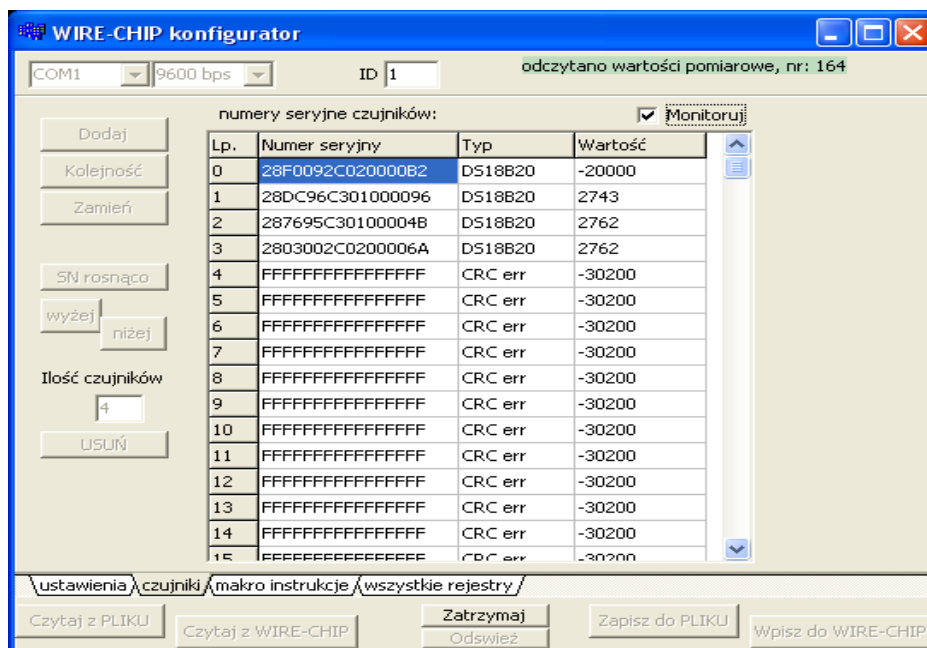
11. Aby zmienić podstawowe ustawienia modułu należy przejść do karty „ustawienia”.

Są tam dostępne takie ustawienia jak:

1. ID (dec) – podawane dziesiętnie ID,
2. Szybkość transmisji,
3. Buzzer w przypadku błędów
4. Okres pomiaru
5. Tryb pracy

12. Aby dokonywać operacji na czujnikach należy przejść do karty „czujniki”.

Karta „czujniki” – umożliwia edycję konfiguracji oraz monitorowanie stanu czujników. Poniżej wygląd karty „czujniki” podczas monitorowania czujników:



Usuwanie czujników

odbywa się najpierw w programie wire-konfigurator, a fakt ich usunięcia jest przekazywany do modułu po wciśnięciu przycisku „Wpisz do WIRE-CHIP”.

Dodawanie czujników

Po wciśnięciu przycisku „Dodaj”, program przekazuje komendę do modułu WIRE-CHIP, który odnajduje wszystkie podłączone do magistrali 1-WIRE czujniki. Jeśli Jakiś czujnik był już zainstalowany wcześniej, wówczas moduł nie zmienia jego numeru. Jeśli jakiś czujnik nie był wcześniej zainstalowany, moduł przyporządkowuje mu nowy numer.

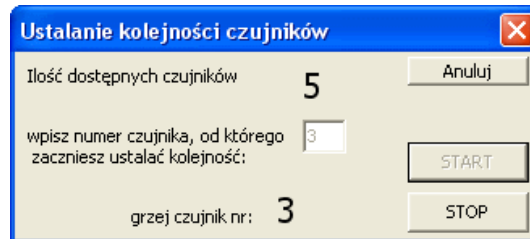
Np.: W module były zainstalowane 4 czujniki (numery 0..3), użytkownik podłączył dodatkowo 3 czujniki. Nowe czujniki będą mieć numery 4, 5, 6 (wszystkich będzie 7).

UWAGA

Do wyszukiwania (dodawania i ustawiania kolejności) czujników moduł potrzebuje użyć całą dostępną pamięć RAM (także tą przeznaczoną w czasie konfiguracji „MODBUS CONFIG” na ustawienia podstawowe i makroinstrukcje). Z tego powodu, aby nie utracić ustawień podstawowych i makroinstrukcji, zalecane jest najpierw dokonać odczytu danych z modułu, a dopiero później wykonać właściwe działania (dodania czujników, ustalenie kolejności).

Ustawianie kolejności czujników

dodanie nowych czujników spowoduje, że będą one wszystkie zainstalowane w systemie, natomiast nie będą znane ich numery (jeśli dodano więcej niż 1 czujnik jednocześnie). Aby ustawić kolejność czujników, należy wcisnąć przycisk „Kolejność”. Zostanie pokazane okno dialogowe umożliwiające ustalenie kolejności. Poniżej widok okna.



Jeśli użytkownik chce ustalić zmienić kolejność ostatnich 2 czujników (z 5 zainstalowanych), w tym przypadku musi w polu edycyjnym wpisać liczbę 3 (ustalenia kolejności czujników będą dotyczyć kolejnych czujników: 3 i 4 w tym przypadku). Ustalanie kolejności czujników rozpocznie się po wciśnięciu przycisku „START”. Należy wówczas **podgrzać lub ochłodzić czujnik**, który ma mieć numer 3. Aby zmiana temperatury czujnika została wykryta, jego temperatura musi w czasie okresu pomiarowego zmienić się o 1 °C. Po wykryciu podgrzania / ochłodzenia czujnika, moduł WIRE-CHIP daje krótki sygnał dźwiękowy, przypisuje czujnikowi numer 3 i prosi o podgrzanie kolejnego czujnika (aż do ostatniego zainstalowanego). Jeśli użytkownik zamierza zakończyć (nie anulować) ustalanie kolejności czujników, wówczas powinien wcisnąć przycisk „STOP” i poczekać na komunikat o zakończeniu ustalania kolejności (program musi pobrać dane o zmianach kolejności z modułu).

Po zakończeniu (STOP, normalne zakończenie) operacji dodawania lub ustalania kolejności czujników, do modułu zostaje automatycznie przepisana poprzednia zawartość pamięci konfiguracyjnej.

Jeśli ustalanie kolejności było anulowane lub awaryjnie przerwane wówczas, aby dalej pracować z modułem, trzeba wykonać jego restart (i przejście do trybu MODBUS CONFIG) lub wpisać do modułu (przycisk Wpisz do WIRE-CHIP) zawartość aktualnie przechowywanych w programie wire-konfigurator ustawień.

Monitorowanie czujników

Program umożliwia odczyt stanu czujników (bieżące wartości temperatury, błędy). Aby uaktywnić funkcję, należy zaznaczyć funkcję przy słowie „Monitoruj”. W czasie monitorowania pozostałe funkcje są niedostępne. Aby wykonać inne czynności w programie, należy wyłączyć funkcję monitorowania. Jeśli program napotka na problem z odczytem rejestrów, funkcja monitorowania wyłączy się automatycznie po kilkunastu sekundach prób. Funkcja monitorowania pokazuje wartości odczytane z rejestru, aby uzyskać temperaturę w [°C], należy wynik podzielić przez 100.

MAKROINSTRUKCJE

Karta „makro instrukcje” umożliwia edycję komend wykonywanych przez moduł WIRE-CHIP.

Makroinstrukcje mają do dyspozycji pamięć RAM modułu WIRE-CHIP w zakresie 1024..1535 (512 rejestry 16-bitowe), jak również pamięć wyników pomiarów temperatury – zakres 0..63 (64 rejestry 16-bitowe).

Makroinstrukcje nie są wykonywane w trybie konfiguracyjnym.

Jeśli moduł pracuje w trybie SLAVE lub TX – nie są wykonywane makroinstrukcje dotyczące portu szeregowego (odbieranie wysyłanie danych przez port, protokół modbus rtu).

Na kod makroinstrukcji przewidzianych jest 508 bajtów pamięci FLASH. Poszczególne makroinstrukcje mogą zajmować od 1 do 11 bajtów.

Wynik wykonania makroinstrukcji (sukces / porażka) wpisywany jest do własnej pamięci RAM. Jeśli makroinstrukcja zakończona jest sukcesem, odpowiedni bit jest ustawiany, jeśli zakończyła się porażką – odpowiedni bit jest zerowany. Kolejnym makroinstrukcjom przyporządkowane są kolejne bity rejestrów poczynając od rejestru 1024, bitu 0. Druga makroinstrukcja zmienia bit 1, trzecia zmienia bit 2, ..., 19-ta zmienia bit 2 rejestru 1025, itd. .

Wygląd karty „makro instrukcje” z przykładowymi makroinstrukcjami pokazany jest poniżej:

WIRE-CHIP konfigurator

COM1 9600 bps ID 6 Status Komunikacji

komparator: bit = A < B; A jest podaną wartością stałą

bit_out A ptr_B

CMP_AC 101 -5500 0 0 0

Dopisz Dopisz Na Początku Dopisz Na Końcu Zamień Usuń 1 Usuń Wszystkie

Lp.	Adres	Nazwa	Szczegóły
1	0	CMP_HIST_C	bit_out=100, ptr_A=0, ptr_B=1504, HIST=100
2	8	CMP_AC	bit_out=101, A=-5500, ptr_B=0
3	14	BIT_AND	bit_out=130, bit_start=100, bit_stop=101
4	18	CMP_HIST_C	bit_out=102, ptr_A=1, ptr_B=1505, HIST=100
5	26	CMP_AC	bit_out=103, A=-5500, ptr_B=1
6	32	BIT_AND	bit_out=131, bit_start=102, bit_stop=103
7	36	CMP_HIST_C	bit_out=104, ptr_A=3, ptr_B=1506, HIST=100
8	44	CMP_AC	bit_out=105, A=-5500, ptr_B=3
9	50	BIT_AND	bit_out=132, bit_start=104, bit_stop=105
10	54	CMP_HIST_C	bit_out=106, ptr_A=5, ptr_B=1507, HIST=100
11	62	CMP_AC	bit_out=107, A=-5500, ptr_B=5
12	68	BIT_AND	bit_out=133, bit_start=106, bit_stop=107

ustawienia / czujniki / makro instrukcje / wszystkie rejestry

Czytaj z PLIKU Czytaj z WIRE-CHIP Zatrzymaj Zapisz do PLIKU Wpisz do WIRE-CHIP

Odśwież

Edycja makroinstrukcji

dodawanie i zamiana

Należy wybrać odpowiedni rodzaj makroinstrukcji, wpisać odpowiednie wartości parametrów, a następnie wybrać jedną z opcji „dopisz”, „dopisz na początku”, „dopisz na końcu” lub „zamień”. Po wybraniu odpowiedniego typu makroinstrukcji, zostają aktywowane odpowiednie pola edycji parametrów (tyle pól, ile parametrów wymaganych jest przez makroinstrukcję). Nad polami edycji parametrów przedstawiane są ich nazwy. Ustawiając kursor myszki nad typem makroinstrukcji lub nad nazwą parametru, program wyświetla duszka (hint) z opisem parametru. Ustawiając kursor nad polem edycji parametru, program wyświetla duszka z zakresem przyjmowanych wartości.

Użycie opcji „**dopisz na początku**” spowoduje przesunięcie wszystkich innych zdefiniowanych makroinstrukcji o jedną pozycję w dół listy i dodanie aktualnie konfigurowanej makroinstrukcji na początku listy.

Użycie opcji „**dopisz na końcu**” powoduje dopisanie aktualnie konfigurowanej makroinstrukcji na końcu listy.

Użycie opcji „**dopisz**” rozpoczyna oczekiwanie programu na kliknięcie w odpowiednim wierszu tabeli z makroinstrukcjami. Makroinstrukcja dodawana jest we wskazanym wierszu. Jeśli we wskazanym i następujących wierszach są zdefiniowane makroinstrukcje, wówczas zostaną one przeniesione o 1 pozycję w dół listy.

Użycie opcji „**zamień**” zamienia makroinstrukcję aktualnie wybranego wiersza tabeli, obecnie definiowaną makroinstrukcją.

usuwanie

Możliwe jest usunięcie wszystkich makroinstrukcji jednocześnie lub pojedynczej wskazanej makroinstrukcji. Aby usunąć wszystkie makroinstrukcje, należy kliknąć przycisk „**usuń wszystkie**”. Aby usunąć tylko 1 makroinstrukcję, należy kliknąć przycisk „**usuń 1**”, a następnie wskazać kliknięciem wiersz tabeli w którym znajduje się makroinstrukcja do usunięcia.

Lista dostępnych makroinstrukcji

Oprogramowanie firmware w obecnej wersji posiada następujące makroinstrukcje:

MODB_RHR - ReadHoldingRegisters - odczyt rejestrów poprzez MODBUS z urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

MODB_PHR - PresetHoldingRegisters - zapis rejestrów poprzez MODBUS do urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

MODB_RC – Read Coils – odczyt bitów poprzez MODBUS z urządzenia SLAVE

MODB_FC – Force Coils – zapis bitów poprzez MODBUS do urządzenia SLAVE

BIT_PULS0, BIT_PULS1, BIT_PULS2, BIT_PULS3 - przekazanie wartości pulsującego (1Hz) bitu (nr) do wyjścia cyfrowego 0, 1, 2, 3=buzzer

BIT_OUT0, BIT_OUT1, BIT_OUT2, BIT_OUT3 - przekazanie wartości (bez pulsowania) bitu (nr) do wyjścia cyfrowego 0, 1, 2, 3=buzzer

BIT_ZERO - zerowanie pojedynczego bitu (nr)

BITY_ZERO - zerowanie wielu bitów (nr od, do)

BITY_SET - ustawianie (1) wielu bitów (nr od, do)
BIT_SUM - suma logiczna wielu bitów (nr od, do)
BIT_AND - iloczyn logiczny wielu bitów
BIT_NEG - zanegowanie jednego bitu

CMP - prosty komparator 2-zmiennych (ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq B$)
CMP_HIST_C - komparator z histerezą, która ma wartość stałą (podaną w makroinstrukcji) (ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq (B+H)$)
CMP_HIST_I - komparator z histerezą, która ma wartość w rejestrze wskazanym przez parametr
CMP_BC - komparator jak wyżej, ale $B = \text{const}$ (wartość podana jako parametr)
CMP_AC - komparator jak wyżej, ale $A = \text{const}$ (wartość podana jako parametr)

Szczegółowy opis makroinstrukcji

MODB_RHR - Read Holding Registers

odczyt rejestrów poprzez MODBUS z urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

- zajętość kodu w pamięci FLASH: 7B
- parametry:
 - ID – adres urządzenia SLAVE na magistrali MODBUS RTU [0..255],
 - ilość - ilość rejestrów do przesłania [1..64],
 - ADR_MA – adres wewnątrz opisywanego modułu (MASTER) [0..1535]
 - ADR_SL – adres rejestrów zewnętrznego modułu (SLAVE) [0.65535]

MODB_PHR - Preset Holding Registers

zapis rejestrów poprzez MODBUS do urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

- zajętość kodu w pamięci FLASH: 7B
- parametry:
 - ID – adres urządzenia SLAVE na magistrali MODBUS RTU [0..255],
 - ilość - ilość rejestrów do przesłania [1..64],
 - ADR_MA – adres wewnątrz opisywanego modułu (MASTER) [0..1535]
 - ADR_SL – adres rejestrów zewnętrznego modułu (SLAVE) [0.65535]

MODB_RC - Read Coils

odczyt bitów poprzez MODBUS z urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

- zajętość kodu w pamięci FLASH: 7B
- parametry:
 - ID – adres urządzenia SLAVE na magistrali MODBUS RTU [0..255],
 - ilość - ilość bitów do przesłania [1..64],
 - ADR_MA – adres bitu wewnątrz opisywanego modułu (MASTER) [0..255]
 - ADR_SL – adres bitów zewnętrznego modułu (SLAVE) [0.65535]

MODB_FC - Force Coils

zapis bitów poprzez MODBUS do urządzenia SLAVE (WIRE-CHIP w trybie MASTER)

- zajętość kodu w pamięci FLASH: 7B

- parametry:
 - ID – adres urządzenia SLAVE na magistrali MODBUS RTU [0..255],
 - ilość - ilość bitów do przesłania [1..64],
 - ADR_MA – adres bitu wewnątrz opisywanego modułu (MASTER) [0..255]
 - ADR_SL – adres bitów zewnętrznego modułu (SLAVE) [0.65535]

BIT_PULS0, BIT_PULS1, BIT_PULS2, BIT_PULS3

przekazanie wartości pulsującego (1Hz) bitu (nr) do wyjścia cyfrowego 0, 1, 2, 3=buzzer

- zajętość kodu w pamięci FLASH: 2B
- parametry:
 - bit_nr – nr bitu [0..255] (rejstry traktowane „bitowo”), którego wartość „1” powoduje pulsowanie wybranego wyjścia (0,1,2,3).

BIT_OUT0, BIT_OUT1, BIT_OUT2, BIT_OUT3

przekazanie wartości (bez pulsowania) bitu (nr) do wyjścia cyfrowego 0, 1, 2, 3=buzzer

- zajętość kodu w pamięci FLASH: 2B
- parametry:
 - bit_nr – nr bitu [0..255] (rejstry traktowane „bitowo”), którego wartość przepisać do wybranego wyjścia (0,1,2,3).

BIT_ZERO

zerowanie pojedynczego bitu (nr)

- zajętość kodu w pamięci FLASH: 2B
- parametry:
 - bit_nr – nr bitu [0..255] (rejstry traktowane „bitowo”), którego wartość zerować,

BITY_ZERO

zerowanie wielu bitów (nr od, do)

- zajętość kodu w pamięci FLASH: 3B
- parametry:
 - bit_start – nr pierwszego bitu do zerowania [0..255] (rejstry traktowane „bitowo”),
 - bit_stop – nr ostatniego bitu do zerowania.

BITY_SET

ustawianie (1) wielu bitów (nr od, do)

- zajętość kodu w pamięci FLASH: 3B
- parametry:
 - bit_start – nr pierwszego bitu do ustawienia [0..255] (rejstry traktowane „bitowo”),
 - bit_stop – nr ostatniego bitu do ustawienia.

BIT_SUM

suma logiczna wielu bitów (nr od, do)

- zajętość kodu w pamięci FLASH: 4B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],

- bit_start – nr pierwszego bitu do sumowania [0..255],
- bit_stop – nr ostatniego bitu do sumowania [0..255].

BIT_AND

iloczyn logiczny wielu bitów

- zajętość kodu w pamięci FLASH: 4B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - bit_start – nr pierwszego bitu iloczynu [0..255],
 - bit_stop – nr ostatniego bitu iloczynu [0..255].

BIT_NEG

zanegowanie jednego bitu

- zajętość kodu w pamięci FLASH: 2B
- parametry:
 - bit_nr – nr bitu [0..255] (rejstry traktowane „bitowo”), którego wartość negować,

CMP

prosty komparator 2-zmiennych (ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq B$)

- zajętość kodu w pamięci FLASH: 6B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - ptr_A – nr rejestru przechowującego wartość A [0..1535],
 - ptr_B – nr rejestru przechowującego wartość B [0..1535].

CMP_HIST_C

komparator z histerezą, która ma wartość stałą (podaną w makroinstrukcji) (ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq (B+H)$)

- zajętość kodu w pamięci FLASH: 8B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - ptr_A – nr rejestru przechowującego wartość A [0..1535],
 - ptr_B – nr rejestru przechowującego wartość B [0..1535],
 - HIST – wartość histerezy [-32768..+32767].

CMP_HIST_I

komparator z histerezą, która ma wartość w rejestrze wskazanym przez parametr

- zajętość kodu w pamięci FLASH: 8B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - ptr_A – nr rejestru przechowującego wartość A [0..1535],
 - ptr_B – nr rejestru przechowującego wartość B [0..1535],
 - ptr_H – nr rejestru przechowującego wartość histerezy [0..1535].

CMP_BC

komparator jak wyżej, ale $B = \text{const}$ (wartość podana jako parametr)(ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq B$)

- zajętość kodu w pamięci FLASH: 6B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - ptr_A – nr rejestru przechowującego wartość A [0..1535],
 - B – wartość B [-32768..+32767].

CMP_AC

komparator jak wyżej, ale $A = \text{const}$ (wartość podana jako parametr)(ustawia bit, jeśli $A < B$, zeruje jeśli $A \geq B$)

- zajętość kodu w pamięci FLASH: 6B
- parametry:
 - bit_out – nr bitu do którego wpisać wynik [0..255],
 - A – wartość A [-32768..+32767].
 - ptr_B – nr rejestru przechowującego wartość B [0..1535],

Przykładów użycia makroinstrukcji należy szukać na naszej stronie internetowej.

Producent

SolidChip
ul. Olszowa 4, Niedziałka Druga,
05-306 Jakubów / obok Mińska Mazowieckiego
www.SolidChip.eu
biuro@solidchip.eu
tel./ faks 025-759-28-80
GSM 693-367-323

