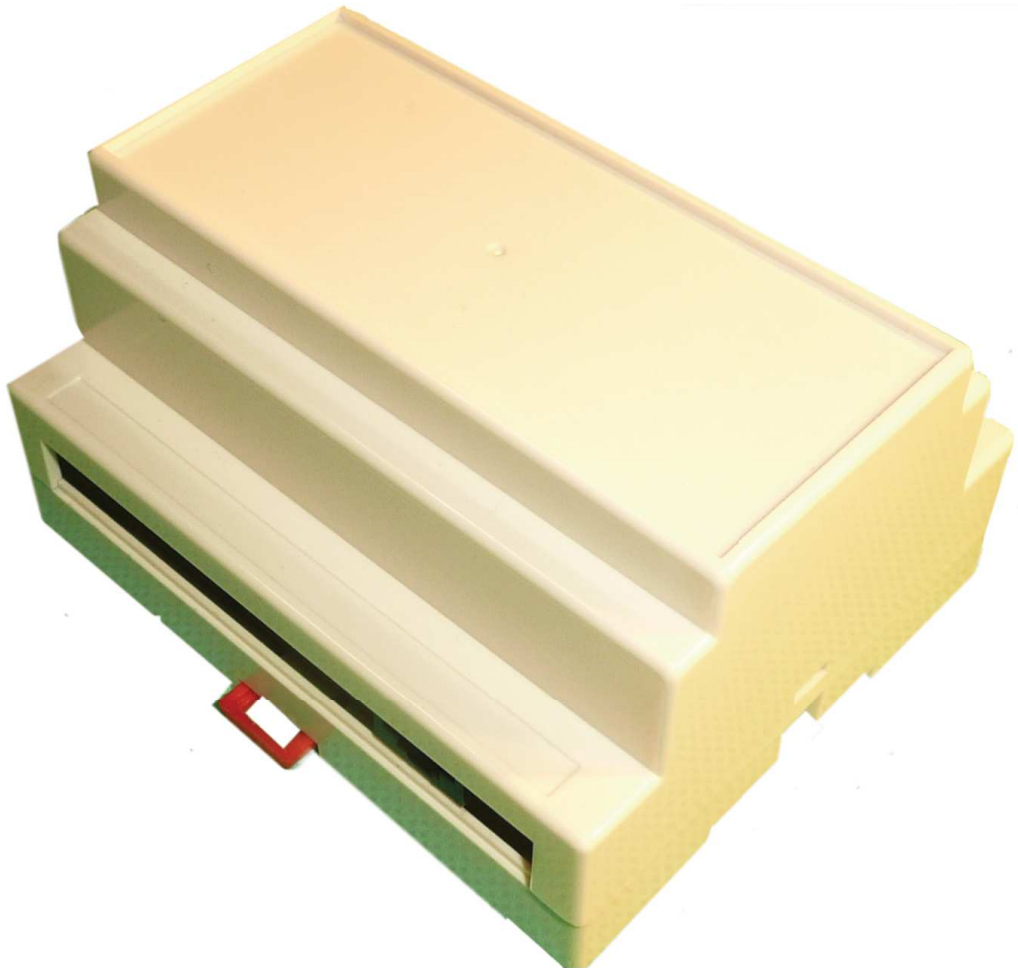


Moduł wejść i wyjść z komunikacją MODBUS

IB – Tron H3F1

PRODUKT POSIADA ZNAK CE

I ZOSTAŁ WYPRODUKOWANY ZGODNIE Z NORMĄ ISO 9001

„INSBUD”
ul. Niepodległości 16a
32-300 Olkusz
Polska
dział sprzedaży: +48 503 166 906
dział techniczny: +48 510 071 213
e-mail: insbud@insbud.net

WWW.INSBUD.NET

instrukcja: 1.0.0
firmware: H3F1V6

Spis Treści

IB-TRON H3F1

INFORMACJE OGÓLNE

H3F1 to moduł wejść i wyjść wraz z obsługą funkcji WATCHDOG do montażu na szynę DIN TH35 wyposażony został w interfejs komunikacyjny RS485 HALF DUPLEX (dwuprzewodowy) na którym został zaimplementowany protokół MODBUS RTU, przy czym H3F1 pełni funkcję SLAVE, czyli jest odpytywane przez urządzenie MASTER protokołu MODBUS RTU.

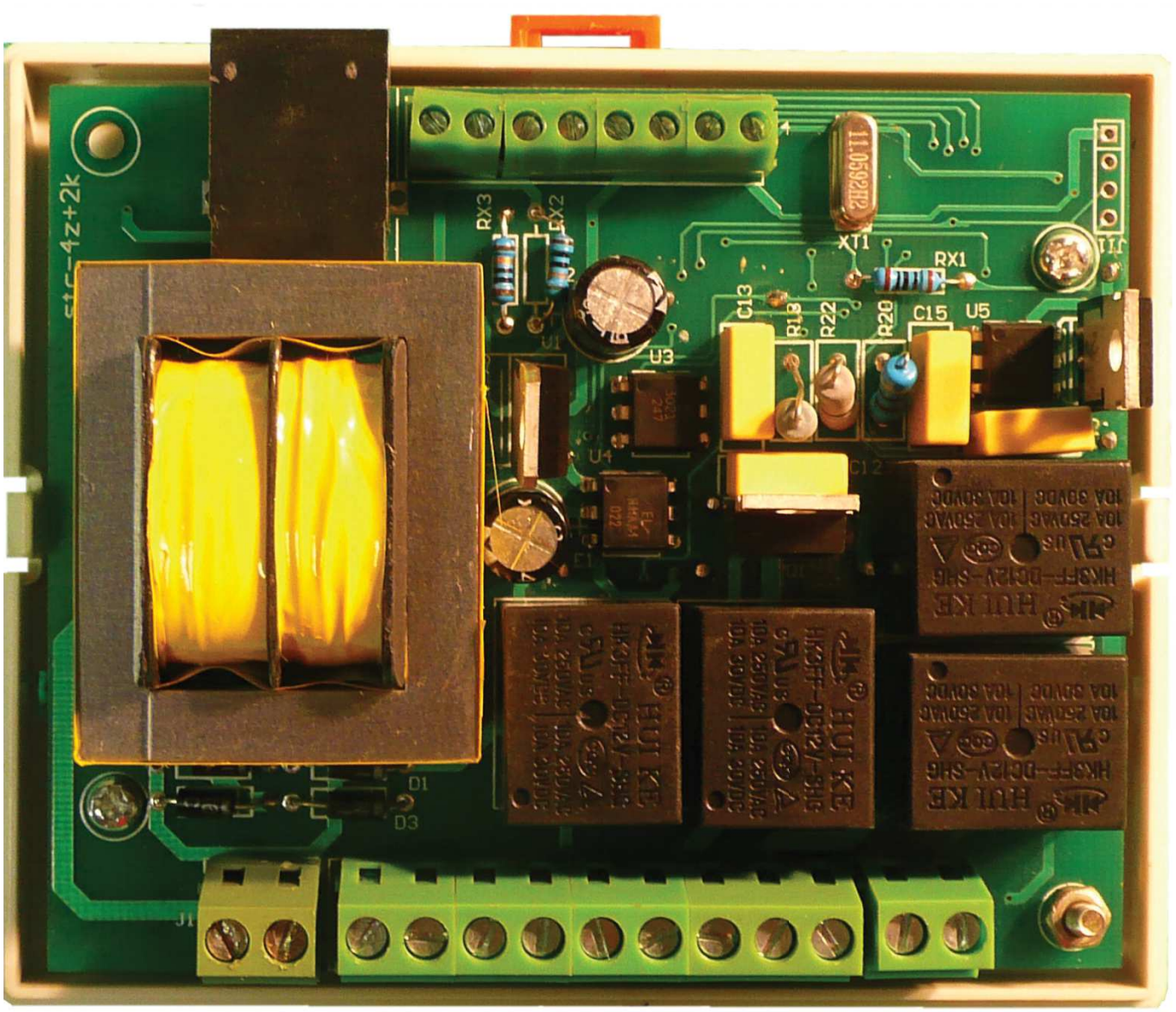
WŁASCIWOŚCI OGÓLNE

- Montaż na szynie DIN TH35
- Zasilanie 230V AC
- Podtrzymywanie pamięci
- Obsługa 4 wejść temperaturowych
- Obsługa 4 wyjść przekaźnikowych 230V
- Obsługa 2 wyjść proporcjonalnych PWM (triak) 230V
- Kalibracja torów pomiarowych
- Funkcja zewnętrznego WATCHDOGA
- Komunikacja RS-485 zgodna z protokołem MODBUS RTU

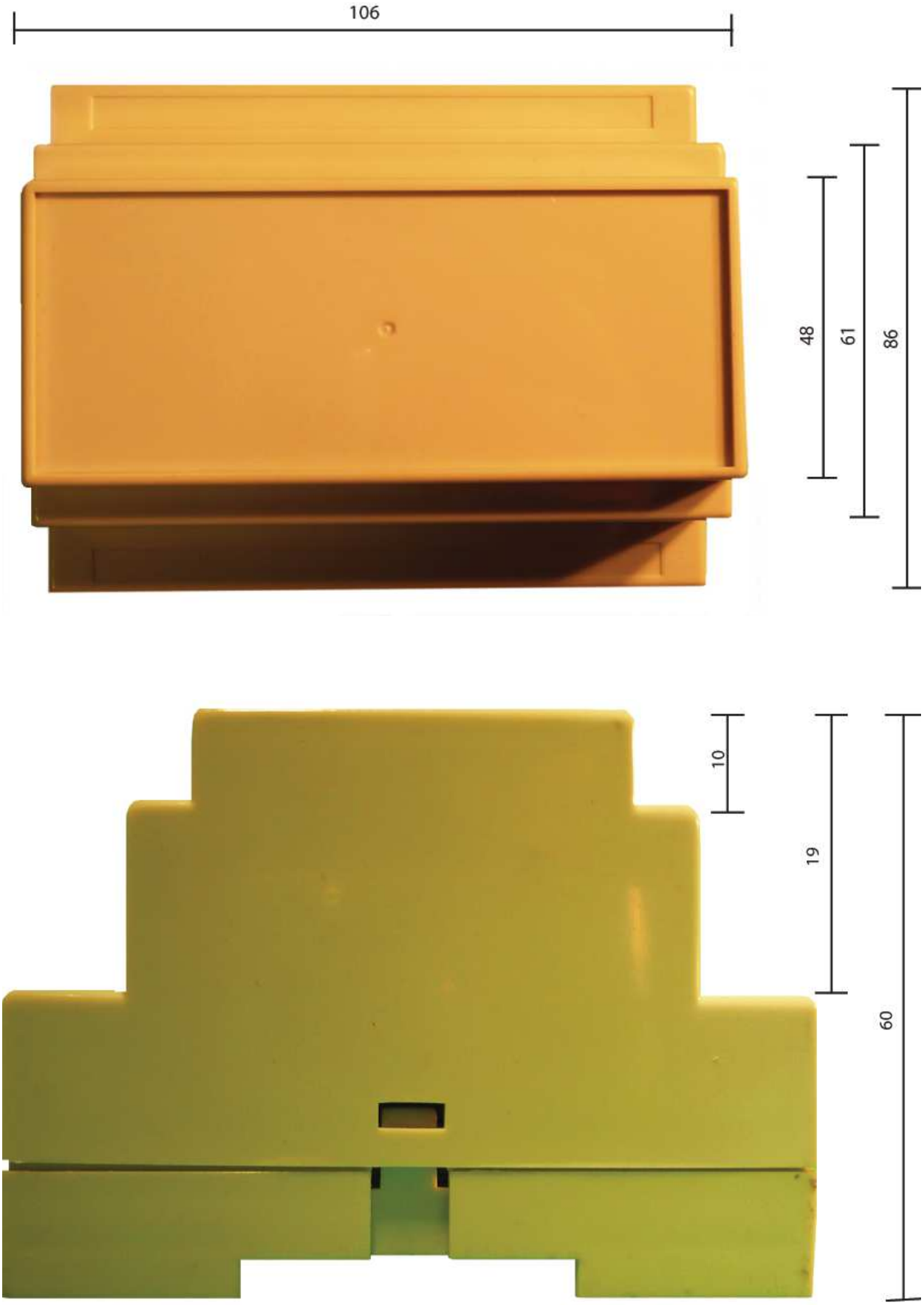
DANE TECHNICZNE

- Zużycie energii: < 2 W
- Temp. składowania: -20 ÷ 50 °C
- Zasilanie: 230V AC
- Rozmiary [mm]: 106 x 86 x 60
- Warunki wilgotności: 5 ÷ 90%
- Obudowa: ABS
- Stopień ochrony: IP30
- Montaż: szyna DIN TH35
- Wyjścia:
  - 4x przekaźnikowe 230V
  - 2x proporcjonalnych PWM (triak) 230V
- Maksymalne obciążenie: 2A dla przekaźnika na kanał. Sumarycznie nie więcej niż 3A
- Wejścia:
  - 4x temperaturowe NTC10kOhm z możliwością interpretacji jako wejście cyfrowe (zwarte/rozwarte)
- Komunikacja: RS-485
- Parametry komunikacji: 9800 8 N 1
- Protokół: MODBUS RTU
- Obsługiwane funkcje:
  - Read Holding Registers (0x03)
  - Preset Single Register (0x06)
  - Write Multiple Registers (0x10)

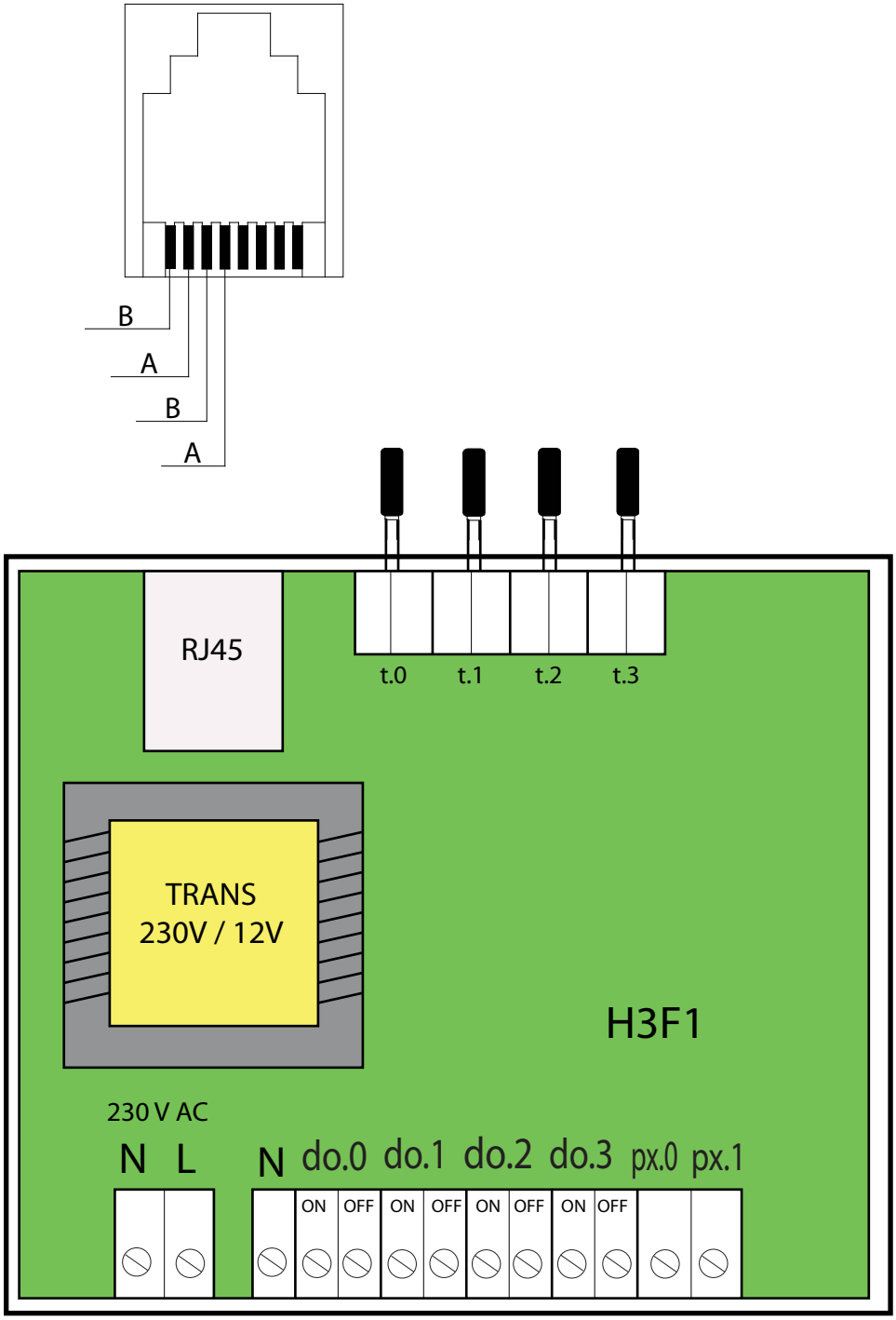
Budowa Sterownika



Wymiary

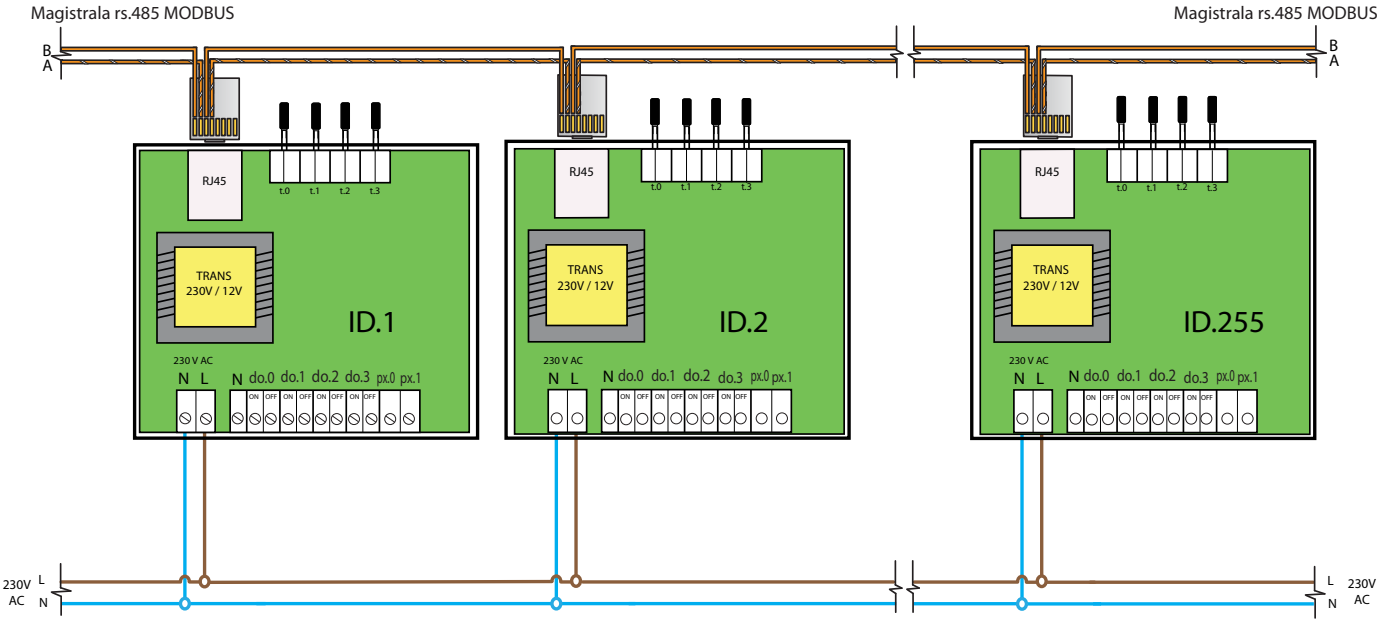


Podłączenie

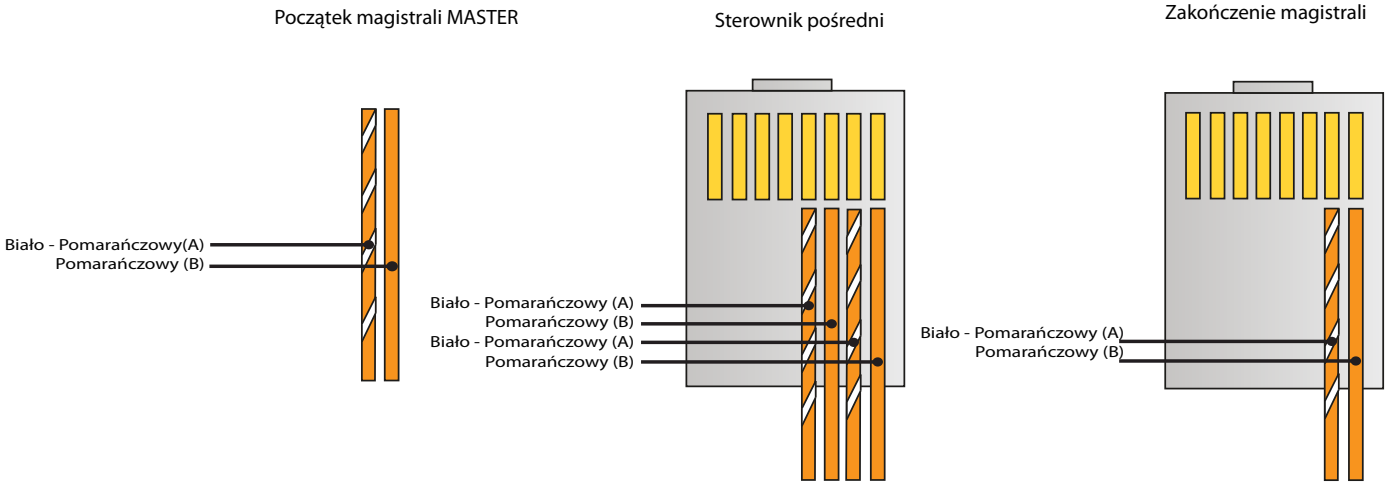


- ☞ Jeżeli wyjście do.N jest włączone (stan 1) to faza znajduje się na złączu ON
- ☞ Jeżeli wyjście do.N jest wyłączone (stan 0) to faza znajduje się na złączu OFF
- ☞ Wyjścia px.N to wyjścia PWM (triak) 230V. Wyjście fazowe.
- ☞ Wejścia t.N to wejścia temperaturowe lub zamiennie cyfrowe (zwarte/rozwarte)

PRZYKŁAD PODŁĄCZENIA



Magistrala rs



Czujniki Temperatury

Urządzenie jest kompatybilny z czujnikami NTC 10kΩ o następującej charakterystyce:

Temperatura [°C]	Oporność [Ω]
-40	346 405
-30	181 628
-20	99 084
-10	56 140
0	32 960
10	20 000
20	12 510
25	10 000
30	8 047
40	5 310
50	3 588
60	2 476
70	1 743
80	1 249
90	911
100	647

Funkcjonalność

H3F1 umożliwia zdalne sterowanie wyjściami oraz odczyt wejść temperaturowych lub zamiennie wejść cyfrowych przy pomocy protokołu komunikacyjnego MODBUS RTU.

Urządzenie zostało wyposażone w funkcję WATCHDOG, która umożliwia reset zewnętrznego urządzenia w przypadku utraty komunikacji.

Zmiana adresu MODBUS odbywa się z poziomu protokołu komunikacyjnego MODBUS RTU.

Watchdog

Jeżeli przez zadany okres czasu, który jest zapisany w rejestrze setting.hwd.period nie nastąpi pobudzenie rejestru setting.hwd.heartbit, wówczas wyjście przekaźnikowe do.0 zostanie ustawione jako wyłączone (stan 0) na wskazany okres czasu po czym zostanie włączone (stan 1).

Funkcjonalność ta umożliwia reset np. zewnętrznych bramek MODBUS lub wprost urządzenia MASTER w przypadku zaniku poprawnej komunikacji ze sterownikiem.

Interfejs Komunikacyjny

Urządzenie jest wyposażone w interfejs komunikacyjny RS485 HALF DUPLEX, o parametrach 9600 8 N 1. Na fizycznym interfejsie RS485 został zaimplementowany protokół komunikacyjny MODBUS RTU. Urządzenie pełni rolę układu SLAVE, który jest odpytywany przez urządzenie nadrzędne MASTER protokołu. Maksymalny odstęp czasowy pomiędzy wysyłanymi do urządzenia bajtami w ramce nie powinien wynosić więcej jak Tbt = 8m s, natomiast maksymalny czas przetwarzania ramki (od momentu odebrania ostatniego bajtu od urządzenia master do chwili wysłania pierwszego bajtu odpowiedzi do urządzenia master) wynosi Tresp = 2 0 m s. Ponadto po wysłaniu ostatniego bajtu odpowiedzi, gdy doszło do uaktualnienia EEPROM, urządzenie potrzebuje ok. Tprep = 30ms na przygotowanie się do odbioru następnej ramki danych. Żeby wyznaczyć maksymalną częstotliwość wymiany ramek, wówczas do czasów Tresp oraz Tprep należy doliczyć czas potrzebny na transmisję ramek z mastera do urządzenia oraz zwrotnej odpowiedzi uwzględniając przy tym rozmiary ramek oraz prędkość transmisji.

INTERFEJS KOMUNIKACYJNY

Należy również brać pod uwagę opóźnienia wprowadzane przez urządzenia i protokoły występujące w torze transmisji (np. konwerter RS489/TCPIP). Uwzględniony powinien również czas ciszy MODBUS, który wynosi czas transmisji 4 bajtów Tslnt = time(4bytes) przed pierwszym i za ostatnim bajtem ramki, co sumarycznie daje czas 8 przetransmitowanych bajtów.

REJESTRY MODBUS

Rejestr	0
Nazwa	dev.hardware
Wartości	3
Typ	R

Identyfikator sprzętu

Rejestr	1
Nazwa	dev.firmware
Wartości	1
Typ	R

Identyfikator oprogramowania

Rejestr	2
Nazwa	dev.version
Wartości	6
Typ	R

Wersja oprogramowania

Rejestr	3
Nazwa	dev.uid.0
Wartości	0..65535
Typ	R

Rejestr	4
Nazwa	dev.uid.1
Wartości	0..65535
Typ	R

Rejestr	5
Nazwa	dev.uid.2
Wartości	0..65535
Typ	R

Rejestr	6
Nazwa	dev.uid.3

Rejestry MODBUS

Wartości	0..65535
Typ	R

Unikalny identyfikator urządzenia.

Rejestr	7
Nazwa	dev.reset
Wartości	1
Typ	RW

Wpis wartości 1 do tego rejestru powoduje przywrócenie ustawień fabrycznych urządzenia.

UWAGA: Adres MODBUS zostanie również przywrócony do wartości fabrycznej.

Rejestr	8
Nazwa	dev.restart
Wartości	1
Typ	RW

Ustawienie na jeden tej flagi powoduje restart urządzenia

Rejestr	9
Nazwa	modbus.address
Wartości	1..255
Typ	RW

Adres MODBUS urządzenia - wartość domyślna to 255

Rejestr	10
Nazwa	input.t.0.value
Wartości	-32768..32767
Typ	R

Rejestr	11
Nazwa	input.t.1.value
Wartości	-32768..32767
Typ	R

Rejestr	12
Nazwa	input.t.2.value
Wartości	-32768..32767
Typ	R

Rejestr	13
Nazwa	input.t.3.value
Wartości	-32768..32767
Typ	R

Wartości temperatur zmierzonych odpowiednio na kanałach 0, 1, 2, 3 w dziesiętnych częściach stopnia Celsjusza. Mogą one być również wykorzystane jako wejścia dwustanowe.

Rejestr	14
Nazwa	input.t.0.err
Wartości	0..5
Typ	R

Rejestr	15
Nazwa	input.t.1.err
Wartości	0..5
Typ	R

Rejestr	16
Nazwa	input.t.2.err
Wartości	0..5
Typ	R

Rejestr	17
Nazwa	input.t.3.err
Wartości	0..5

REJESTRY MODBUS

Typ	R

Statusy wejść temperaturowych:
0 - pomiar prawidłowy
1 - brak czujnika
2 - zwarcie na wejściu
3 - zbyt wysoka temperatura
4 - zbyt niska temperatura
5 - inny błąd

Kody błędów np. 1 i 2 można wykorzystac do interpretacji wejść cyfrowych (zwarte/rozwarte).

Rejestr	18
Nazwa	output.px.0
Wartości	0..100
Typ	RW

Rejestr	19
Nazwa	output.px.1
Wartości	0..100
Typ	RW

Wartości dla wyjść analogowych PWM (triak). Po uruchomieniu urządzenia rejestry te ładowane są wartościami domyślnymi, które zapisane są w rejestrach odpowiednio setting.default.px.0 oraz setting.default.px.0.

Rejestr	20
Nazwa	output.do.0
Wartości	0..1
Typ	RW

Wyjście dwustanowe przekaźnikowe. Wyjście to jest również wykorzystywane przez funkcję watchdoga, jeżeli została ona aktywowana - patrz opis. Jeżeli funkcja watchdoga nie jest aktywna, wówczas po uruchomieniu urządzenia, do rejestru tego przepisywana jest wartość z rejestru setting.default.do.0. Odczyt tego rejestru zwraca zawsze faktyczny stan przekaźnika. Ustawienie tego rejestru, powoduje automatyczne ustawienie powiązanego z nim przekaźnika.

Rejestr	21
Nazwa	output.do.1
Wartości	0..1
Typ	RW

Rejestr	22
Nazwa	output.do.2
Wartości	0..1
Typ	RW

Rejestr	23
Nazwa	output.do.3
Wartości	0..1
Typ	RW

Rejestr	24
Nazwa	output.do.4
Wartości	0..1
Typ	RW

Wyjścia dwustanowe przekaźnikowe. Podczas uruchomienia, do rejestrów tych przepisywane są wartości rejestrów odpowiednio: setting.default.do.1, setting.default.do.2, setting.default.do.3, setting.default.do.4. Odczyt ich zwraca zawsze faktyczny stan danego przekaźnika. Ustawienie ich, powoduje automatyczne ustawienie przekaźnika związanego z danym rejestrem. Wyjście do.4 jest wyjściem wirtualnym i nie jest powiązane z fizycznym przekaźnikiem w urządzeniu.

Rejestr	25

Rejestry MODBUS

Nazwa	setting.default.px.0
Wartości	0..100
Typ	RW

Rejestr	26
Nazwa	setting.default.px.1
Wartości	0..100
Typ	RW

Wartości domyślne wyjść PWM. Wartości tych rejestrów są przepisywane do rejestrów odpowiednio output.px.0 i output.px.1 podczas startu urządzenia

Rejestr	27
Nazwa	setting.default.do.0
Wartości	0..1
Typ	RW

Rejestr	28
Nazwa	setting.default.do.1
Wartości	0..1
Typ	RW

Rejestr	29
Nazwa	setting.default.do.2
Wartości	0..1
Typ	RW

Rejestr	30
Nazwa	setting.default.do.3
Wartości	0..1
Typ	RW

Rejestr	31
Nazwa	setting.default.do.4
Wartości	0..1
Typ	RW

Wartości domyślne wyjść przekaźnikowych. Wartości tych rejestrów są przepisywane do rejestrów odpowiednio output.do.0, output.do.1, output.do.2, output.do.3 oraz output.do.4 podczas startu urządzenia o ile nie została załączona funkcja watchdoga do.0.

Rejestr	32
Nazwa	setting.px.0.min
Wartości	0..100
Typ	RW

Rejestr	33
Nazwa	setting.px.1.min
Wartości	0..100
Typ	RW

Rejestry kalibrujące wyjścia PWM odpowiednio output.px.0 oraz output.px.1. Wartość ta jest podawana do odpowiedniego kanału PWM, jeżeli na odpowiednim wejściu output.px.N wpisano wartość mniejszą niż zadaną w tym rejestrze. Wartość ta nie może być większa od odpowiedniego rejestru setting.px.N.max.

Rejestr	34
Nazwa	setting.px.0.max
Wartości	0..100
Typ	RW

Rejestr	35
Nazwa	setting.px.1.max
Wartości	0..100
Typ	RW

Rejestry kalibrujące wyjścia PWM odpowie-

REJESTRY MODBUS

dio output.px.0 oraz output.px.1. Wartość ta jest podawana do odpowiedniego kanału PWM, jeżeli na odpowiednim wejściu output.px.N wpisano wartość większą niż zadaną w tym rejestrze. Wartość ta nie może być mniejsza od odpowiedniego rejestru setting.px.N.min

Rejestr	36
Nazwa	setting.t.0.calib
Wartości	-50..50
Typ	RW

Rejestr	37
Nazwa	setting.t.1.calib
Wartości	-50..50
Typ	RW

Rejestr	38
Nazwa	setting.t.2.calib
Wartości	-50..50
Typ	RW

Rejestr	39
Nazwa	setting.t.3.calib
Wartości	-50..50
Typ	RW

Rejestry kalibrujące wskazania temperatur w jednostkach dziesiętnych stopni Celsjusza. Wartości tych rejestrów każdorazowo są dodawane do pomiaru temperatury na odpowiednim kanale i wynik zapisywany jest w odpowiednim rejestrze input.t.N.value.

Rejestr	40
Nazwa	counter.system.work_time
Wartości	0..65535
Typ	R

Czas pracy urządzenia wyrażony w minutach.

Rejestr	41
Nazwa	setting.hwd.period
Wartości	0..65535
Typ	RW

Rejestr funkcji watchdoga. Jeżeli jest wpisana wartość zero to funkcja watchdoga jest wyłączona. Każda inna wartość powoduje, że funkcja watchdoga jest załączona i oznacza okres w sekundach, maksymalnie przez który musi zostać wpisana wartość 1 do rejestru setting.hwd.heartbit, żeby wyjście output.do.0 nie zostało ustawione na 0.

Rejestr	42
Nazwa	setting.hwd.heartbit
Wartości	1..1
Typ	RW

Rejestr watchdoga, który przy aktywnej funkcji cyklicznie musi być pobudzany żeby nie zostało ustawione 0 na wyjściu output.do.0 - patrz opis setting.hwd.period.

Rejestr	43
Nazwa	setting.hwd.reset.time
Wartości	1..65535
Typ	RW

Rejestr watchdoga oznaczający czas impulsu resetującego wyjście output.do.0 wyrażony w sekundach. Jeżeli przez okres czasu, który jest zapisany w rejestrze setting.hwd.period nie nastąpi pobudzenie rejestru setting.hwd.heartbit, wówczas wyjście output.do.0 zostanie ustawione na 0 na ten okres po czym zostanie ustawione z powrotem na 1.

Rejestry MODBUS

Rejestr	44
Nazwa	counter.hwd.resets_no
Wartości	0..65535
Typ	R

Liczba resetów wykonanych przez watchdog. Resety spowodowane ręcznie nie są zliczane.

Rejestr	45
Nazwa	setting.hwd.clr_resets_no
Wartości	1..1
Typ	RW

Wpisanie jedynki do tego rejestru powoduje wyczyszczenie rejestru counter.hwd.resets_no.

Funkcje MODBUS

Urządzenie obsługuje trzy funkcje standardu MODBUS:

- ☞ Read Holding Registers (Function Code 0x03)
- ☞ Preset Single Register (Function Code 0x06)
- ☞ Write Multiple Registers (Function Code 0x10)

W odpowiedzi wysyła dane, potwierdzenie wykonanych zapisów lub zwraca błąd, opisany jednym z następujących kodów wyjątków:

- ☞ Illegal Function (Exception Code 0x01)
- ☞ Illegal Data Address (Exception Code 0x02)
- ☞ Illegal Data Value (Exception Code 0x03)
- ☞ Slave Device Failure (Exception Code 0x04)

⚠ UWAGA: Poniższe przykłady mają za zadanie pokazać przykładową komunikację MODBUS i nie muszą odnosić się do rzeczywistych rejestrów w urządzeniu. W przykładach komunikacji urządzenie HxFy ma adres domyślny 255 (0xff)

1. Read Holding Registers (Function Code 0x03)

Funkcja odczytuje określoną liczbę rejestrów, począwszy od danego adresu

Rozkaz:

ADDRESS	FUN_CODE	FUN_CODE	REG_ADDR_MSB

REG_ADDR_LSB	REGS_NO_MSB	REGS_NO_LSB

FUNKCJE MODBUS

CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_CODE	kod funkcji MODBUS – w tym przypadku 0x03
REG_ADDR_MSB	starszy bajt adresu pierwszego rejestru do odczytu.
REG_ADDR_LSB	młodszy bajt adresu pierwszego rejestru do odczytu.
REGS_NO_MSB	starszy bajt ilości rejestrów do odczytu.
REGS_NO_LSB	młodszy bajt ilości rejestrów do odczytu.
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REGS_NO_LSB) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REGS_NO_LSB) – starszy bajt

Odpowiedź zwracająca wartości rejestrów:

ADDRESS	FUN_CODE	BYTES	VAL_0_MSB

VAL_0_LSB	...	VAL_N_MSB	VAL_N_LSB

CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_CODE	kod funkcji MODBUS – w tym przypadku 0x03
BYTES	liczba bajtów zajętych przez przesyłane wartości rejestrów
VAL_N_MSB	starszy bajt wartości rejestru N
VAL_N_LSB	młodszy bajt wartości rejestru N
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do VAL_N_LSB) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do VAL_N_LSB) – starszy bajt

Przykład: odczyt dwóch rejestrów (addr 0x0000) oraz (addr 0x0001)

Zapytanie: MASTER->HxFy

0xff	0x03	0x00	0x00	0x00	0x02	0xd1	0xd5

Odpowiedź: HxFy->MASTER

0xff	0x03	0x04	0x00	0x04	0x00	0x02	0x25	0xfc

addr	fcod	byts	reg val 0	reg val 1	crc

HxFy zwróciło wartości dwóch rejestrów. (addr 0x0000) = 4 oraz (addr 0x0001) = 2.

Odpowiedź informująca o błędzie:

ADDRESS	FUN_ERR_CODE	EXCEPTION_CODE	CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_ERR_CODE	suma logiczna kodu funkcji MODBUS – w tym przypadku 0x03 z ustawionym bitem błędu 0x80. Co daje 0x83
EXCEPTION_CODE	kod błędu MODBUS.
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – starszy bajt

Przykład: próba odczytu dwóch rejestrów spoza zakresu adresów.

Adres startowy: 0x1234, liczba rejestrów: 2.

Zapytanie: MASTER->HxFy

0xff	0x03	0x12	0x34	0x00	0x02	0x95	0x63

Odpowiedź: HxFy->MASTER

0xff	0x83	0x02	0xa1	0x01

zwrócony błąd to 0x02 Illegal Data Address.

2. Preset Single Register (Function Code 0x06)

Funkcja wpisuje wartość do pojedynczego rejestru.

Rozkaz:

ADDRESS	FUN_CODE	REG_ADDR_MSB	REG_ADDR_LSB

REG_VAL_MSB	REG_VAL_LSB	CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_CODE	kod funkcji MODBUS – w tym przypadku 0x06
REG_ADDR_MSB	starszy bajt adresu rejestru, do którego ma nastąpić zapis wartości
REG_ADDR_LSB	młodszy bajt adresu rejestru, do którego ma nastąpić zapis wartości
REG_VAL_MSB	starszy bajt wartości, która ma zostać zapisana w rejestrze
REG_VAL_LSB	młodszy bajt wartości, która ma zostać zapisana w rejestrze
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REG_VAL_LSB) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REG_VAL_LSB) – starszy bajt

Odpowiedź potwierdzająca wpis (w tym wypadku jest dokładnie powtórzeniem rozkazu):

ADDRESS	FUN_CODE	REG_ADDR_MSB	REG_ADDR_LSB

REG_VAL_MSB	REG_VAL_LSB	CRC_LSB	CRC_MSB

Przykład: Wpisane do rejestru o adresie 65 (0x0041) wartości 20 (0x0041)

Zapytanie: MASTER->HxFy

0xff	0x06	0x00	0x41	0x00	0x14	0xcc	0x0f

Odpowiedź: HxFy->MASTER

0xff	0x06	0x00	0x41	0x00	0x14	0xcc	0x0f

Odpowiedź informująca o błędzie:

ADDRESS	FUN_ERR_CODE	EXCEPTION_CODE	CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_ERR_CODE	suma logiczna kodu funkcji MODBUS – w tym przypadku 0x06 z ustawionym bitem błędu 0x80. Co daje 0x86
EXCEPTION_CODE	kod błędu MODBUS.
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – starszy bajt

Przykład: próba wpisania do rejestru o adresie 1 (0x001) niedozwolonej wartości 12 (0x000c)

Zapytaie: MASTER->HxFy

0xff	0x06	0x00	0x01	0x00	0x0c	0xcd	0xd1

Odpowiedź: HxFy->MASTER

FUNKCJE MODBUS

0xff	0x86	0x04	0x22	0x53

W tym przypadku urządzenie zwróciło błąd z kodem wyjątku„Slave Device Failure - 0x04”

3. Write Multiple Registers (Function Code 0x10)

Funkcja wpisuje wartości do wybranych rejestrów, począwszy od danego adresu

Rozkaz:

ADDRESS	FUN_CODE	REG_ADDR_MSB	REG_ADDR_LSB

REGS_NO_MSB	REGS_NO_LSB	BYTES_NO	VAL_0_MSB

VAL_0_LSB	...	VAL_N_MSB	VAL_N_LSB	CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_CODE	kod funkcji MODBUS – w tym przypadku 0x10
REG_ADDR_MSB	starszy bajt adresu pierwszego rejestru, do którego ma nastąpić zapis wartości
REG_ADDR_LSB	młodszy bajt adresu pierwszego rejestru, do którego ma nastąpić zapis wartości
REGS_NO_MSB	numer rejestrów do zapisu – starszy bajt
REGS_NO_LSB	numer rejestrów do zapisu – młodszy bajt
BYTES_NO	liczba bajtów, którą zajmują przesyłane wartości rejestrów
VAL_0_MSB	starszy bajt wartości pierwszego rejestru do zapisu
VAL_0_LSB	młodszy bajt wartości pierwszego rejestru do zapisu
VAL_N_MSB	starszy bajt wartości N-tego rejestru do zapisu
VAL_N_LSB	młodszy bajt wartości N-tego rejestru do zapisu
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do VAL_N_LSB) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do VAL_N_LSB) – starszy bajt

Odpowiedź potwierdzająca zapis:

ADDRESS	FUN_CODE	REG_ADDR_MSB	REG_ADDR_LSB

REGS_NO_MSB	REGS_NO_LSB	CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_CODE	kod funkcji MODBUS – w tym przypadku 0x10
REG_ADDR_MSB	starszy bajt adresu pierwszego zapisanego rejestru.
REG_ADDR_LSB	młodszy bajt adresu pierwszego zapisanego rejestru.
REGS_NO_MSB	ilość zapisanych rejestrów – starszy bajt
REGS_NO_LSB	ilość zapisanych rejestrów – młodszy bajt
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REGS_NO_LSB) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do REGS_NO_LSB) – starszy bajt

Przykład: zapis do dwóch rejestrów: do rejestru o adresie 64 (0x0040) wartość 1 (0x0001) oraz do rejestru o adresie 65 (0x0041) wartość 7 (0x0007)

Zapytanie: MASTER->HxFy

0xff	0x10	0x00	0x40	0x00	0x02	0x04	0x00

0x01	0x00	0x07	0xd0	0x76

Funkcje MODBUS

Odpowiedź: HxFy->MASTER

0xff	0x10	0x00	0x40	0x00	0x02	0x55	0xc2

Odpowiedź informująca o błędzie:

ADDRESS	FUN_ERR_CODE	EXCEPTION_CODE

CRC_LSB	CRC_MSB

Gdzie:

ADDRESS	adres MODBUS urządzenia SLAVE – HxFy
FUN_ERR_CODE	suma logiczna kodu funkcji MODBUS – w tym przypadku 0x10 z ustawionym bitem błędu 0x80. Co daje 0x90
EXCEPTION_CODE	kod błędu MODBUS.
CRC_LSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – młodszy bajt
CRC_MSB	suma kontrolna liczona dla całej ramki (począwszy od ADDRESS do EXCEPTION_CODE) – starszy bajt

Przykład: zapis do dwóch rejestrów gdzie do jednego z nich następuje próba zapisu niedozwolonej wartości. Do rejestru o adresie 64 (0x0040) wartość 1 (0x0001) oraz do rejestru o adresie 65 (0x0041) wartość 11 (0x000b)

Zapytanie: MASTER->HxFy

0xff	0x10	0x00	0x40	0x00	0x02	0x04	0x00
0x01	0x00	0x0b	0xd0	0x73			

Odpowiedź: HxFy->MASTER

0xff	0x90	0x04	0x2c	0x33

W tym przypadku urządzenie zwróciło błąd z kodem wyjątku „Slave Device Failure – 0x04" ponieważ do jednego z rejestrów próbowano wpisać wartość poza zakresem.

Przykładowa funkcja licząca MODBUS CRC.

```
/**
* calculates modbus crc
* @param points to the first element
* of modbus frame
* @param points to the first after last
* element of modbus frame
* @tparam IT iterator or pointer type
* @return calculated crc
* */
template <typename IT>
unsigned short calculateCrc(IT first, IT
last)
{
  unsigned int crc = 0xffff;
  while (first != last)
  {
   crc ^= *first++;
   for(int j = 0; j < 8; ++j)
   {
     if(crc & 0x0001)
      crc = (crc >> 1) ^ 0xa001;
     else
      crc >>= 1;
   }
  }
  return crc;
}
```

Wersje Oprogramowania

- 6 - Adres MODBUS jest przechowywany na oddzielnej stronie pamięci flash przy użyciu sprzętu watchdoga. Rejestry UUID. Poprawione PWM.
- 5 - Dodano rejestr dev.restart.
- 4 - Refaktoryzacja.
- 3 - Dodano dodatkowe DO. zmieniono obsługę PWM - zero wykrywania.
- 2 - Skorygowano obliczenie temperatury NTC.
- 1 - Pierwsze wydanie oprogramowania

WARUNKI GWARANCJI

- Gwarancji udziela się na okres 24 miesięcy licząc od dnia zakupu towaru.
- Ujawnione w okresie gwarancji wady będą usuwane w terminie nie dłuższym niż 21 dni roboczych, licząc od daty przyjęcia sprzętu do serwisu.
- W przypadku zaistnienia konieczności importu towaru lub części z zagranicy, czas naprawy ulega wydłużeniu o czas niezbędny do ich sprowadzenia.
- Klient dostarcza towar do serwisu na własny koszt. Towar wysyłany na koszt serwisu nie będzie odebrany.
- Na czas naprawy serwis nie ma obowiązku dostarczenia nabywcy zastępczego towaru.
- Naprawa w ramach gwarancji będzie dokonywana po przedstawieniu poprawnie i czytelnie wypełnionej karty gwarancyjnej reklamowanego sprzętu, podpisanej przez gwaranta i klienta oraz dokumentu sprzedaży.
- Gwarancja obejmuje tylko wady powstałe z przyczyn tkwiących w sprzedanej rzeczy. Nie są objęte gwarancją uszkodzenia powstałe z przyczyn zewnętrznych takich jak: urazy mechaniczne, zanieczyszczenia, zalania, zjawiska atmosferyczne, niewłaściwa instalacja lub obsługa, jak również eksploatacja niezgodna z przeznaczeniem i instrukcją obsługi. Gwarancja nie ma też zastosowania w przypadku dokonania przez Klienta nieautoryzowanych napraw, zmiany oprogramowania (firmwaru) oraz formatowania urządzenia
- Ze względu na naturalne zużycie materiałów eksploatacyjnych, niektóre z nich nie są objęte gwarancją (dotyczy np. kabli, baterii, ładowarek, mikro-styków, przycisków itp).

WARUNKI GWARANCJI

- W przypadku nieuzasadnionego roszczenia w zakresie naprawy gwarancyjnej, koszty przesłania sprzętu do i z serwisu ponosi Klient.
- Serwis ma prawo odmówić wykonania naprawy gwarancyjnej w przypadku: stwierdzenia sprzeczności pomiędzy danymi wynikającymi z dokumentów a znajdującymi się na sprzęcie, dokonania napraw we własnym zakresie, zmian konstrukcyjnych sprzętu.
- Odmowa wykonania naprawy gwarancyjnej jest równoznaczna z utratą gwarancji.
- W przypadku braku możliwości testowana towaru przed jego zakupem (dotyczy sprzedaży na odległość), dopuszcza się możliwość zwrotu towaru w ciągu 14 dni od daty jego otrzymania (decyduje data nadania). Zwracany towar nie może nosić znamion eksploatacji, koniecznie musi zawierać wszystkie elementy, z którymi był dostarczony.
- W przypadku rezygnacji z zakupionego towaru koszt przesyłki ponosi kupujący. Do przesyłki należy dołączyć dokument zakupu oraz podać dokładne dane Nabywcy wraz z numerem konta bankowego, na które zostanie zwrócona kwota równa wartości zwróconego towaru, nie później niż 21 dni roboczych od dnia dostarczenia towaru. Kwota ta jest pomniejszona o koszty wysyłki do Klienta, jeżeli koszty te zostały poniesione przez Sprzedawcę. Warunkiem koniecznym do zwrotu pieniędzy jest dostarczenie podpisanej kopii korekty dokumentu zakupu. Korektę dokumentu zakupu Klient otrzymuje po wcześniejszym kontakcie ze sprzedającym.