

LAN KONTROLER



RESTARTER, MONITOR, WATCHDOG, STEROWNIK

MOŻLIWOŚCI:

- zarządzanie przez WWW lub SNMP v2.
- upgrade firmware zdalnie przez TFTP
- odczyt danych w czasie rzeczywistym bez konieczności odświeżania strony
- możliwość przełączania do 5-ciu przekaźników bezpośrednio ze strony WWW
- tablica zdarzeń dla każdego wejścia i wyjścia do samodzielnego zaprogramowania przez użytkownika
- Scheduler (załączanie wyjść o określonych godzinach w ciągu tygodnia)
- Watchdog IP do 5 urządzeń IP
- monitoring dodatkowych urządzeń np. czujek, stanów położenia
- pomiar temperatury otoczenia i napięcia zasilania urządzenia
- pomiar napięcia, temperatury, prądu z podłączonych czujników
- pomiar mocy i energii dla napięcia stałego
- możliwość dołączenia dodatkowej płytki z 4 przekaźnikami lub 4-ma odłączanymi portami PoE
- ustawianie czasu ręcznie lub wg serwera NTP
- możliwość kalibracji wskazań czujników
- sterowanie częstotliwością i wypełnieniem przebiegu PWM
- powiadamianie mailem o zaprogramowanych zdarzeniach
- powiadamianie SNMP TRAP o zaprogramowanych zdarzeniach
- automatyczne wysyłanie SNMP TRAP z wartością lub stanem wejścia
- obsługiwane protokoły: HTTP, SNMP, SMTP, SNTP, ICMP, DNS, DHCP.
- obsługiwane czujniki temperatury: NTC10K B=3950, KTY-84, PT1000

Domyślny użytkownik i hasło to „admin”, adres IP modułu to

192.168.1.100.

SPECYFIKACJA TECHNICZNA

- napięcie zasilania: 8-36V (od 6V jeśli nie używamy pomiaru prądu i PT1000)
- pobór mocy : 1W
- zasilanie PoE: TAK, pasywne
- zabezpieczenie przed odwrotną polaryzacją zasilania: TAK
- interfejsy: ethernet 10Mbit/s
- przekaźnik: 255VAC 10A
- temperatura pracy: -20 do +85 st. C
- waga 45g (bez obudowy)
- typ obudowy Z-67 (nie dołączona)
- wymiary 57x67mm

WEJSCIA/WYJSCIA:

- 1 PRZEKAZNIK do załączania/odłączania innych urządzeń, dostępny styk normalnie otwarty i normalnie zamknięty
- 4 WYJŚCIA do załączania przekaźników, portów PoE lub innych urządzeń;
- 1 STEROWANE WYJŚCIE PWM od 2,6 KHz do 4Mhz;
- 2 WEJŚCIA LOGICZNE (max 12V): do monitoringu innych urządzeń, np. zasilaczy buforowych. Współpraca z wyjściami typu: OC,NO,
- 5 WEJŚĆ ANALOGOWYCH:

INP1: pomiar temperatury przy pomocy termistora 10K z stałą B=3950(od -40 do +120 °C) lub termistora KTY-84-130 (od -40 do +300°C), dokładność 1 °C (zależy od NTC)

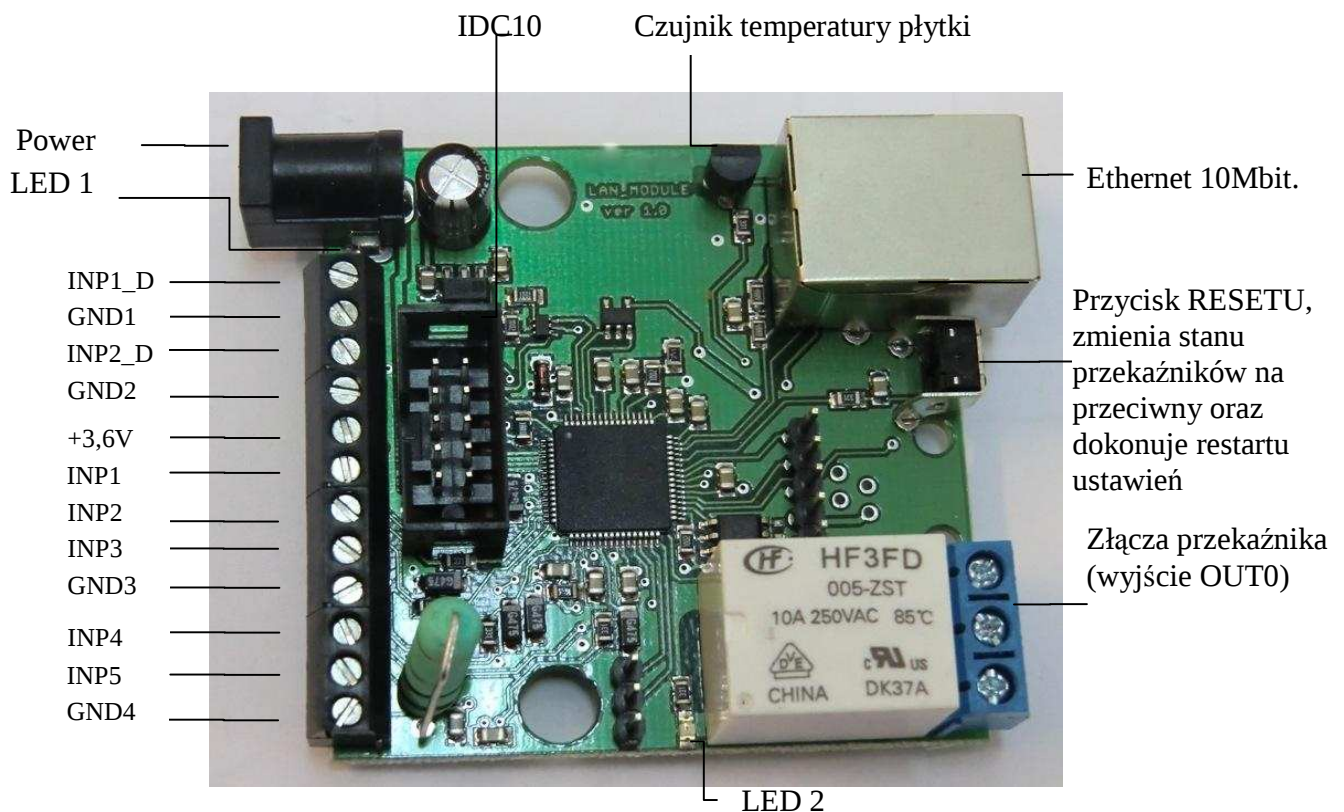
INP2: termistor 10K lub pomiar napięcia stałego do 3,6V, przy użyciu dodatkowego dzielnika zwiększenie zakresu..

INP3: napięcie stałe do 35V, z dokładnością +/- 0,1V;

INP4: pomiar temperatury na PT1000 (od -20 do +850 °C) dokładność +/- 2°C;

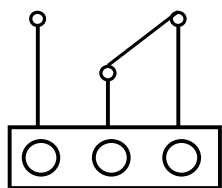
INP5: pomiar prądu stałego do 3A z dokładnością +/- 10mA;

OPIS WYPROWADZEŃ I ELEMENTÓW



Złącze/Element	Opis
Power	Zasilanie urządzenia 8V-35VDC lub przez PoE
ZŁACZA przekaźnika	Do podpięcia urządzeń zewnętrznych, szczegółowy opis poniżej
LED1	Dioda świecąca oznacza podpięcie zasilania do płytki
LED2	Dioda świecąca oznacza załączenie przekaźnika
IDC10	Do podpięcia dodatkowych płytek , szczegółowy opis poniżej
INP1_D	Wejście logiczne pierwsze
GND1	Masa odniesienia dla INP1_D i INP2_D lub masa ogólna
INP2_D	Wejście logiczne drugie
GND2	Masa odniesienia ogólna
+3,6V	Napięcie dla czujników NTC-10K i KTY-84 podpinanych do INP1 i INP2
INP1	Wejście dla czujnika NTC-10K lub pomiar napięcia max (bez dzielnika) 3,6V
INP2	Wejście dla czujnika NTC-10K lub KTY-84-130
INP3	Wejście do pomiaru napięcia max 35V
GND3	Masa odniesienia ogólna
INP4	Wejście na czujnik PT1000 lub podpięcie pod taki czujnik w sterowniku solara
INP5	Wejście na pomiar prądu
GND4	Masa dla pomiaru prądu lub masa ogólna jak nie mierzymy prądu

OPIS ZŁĄCZA PRZekaźNIKA:



NO CON NZ

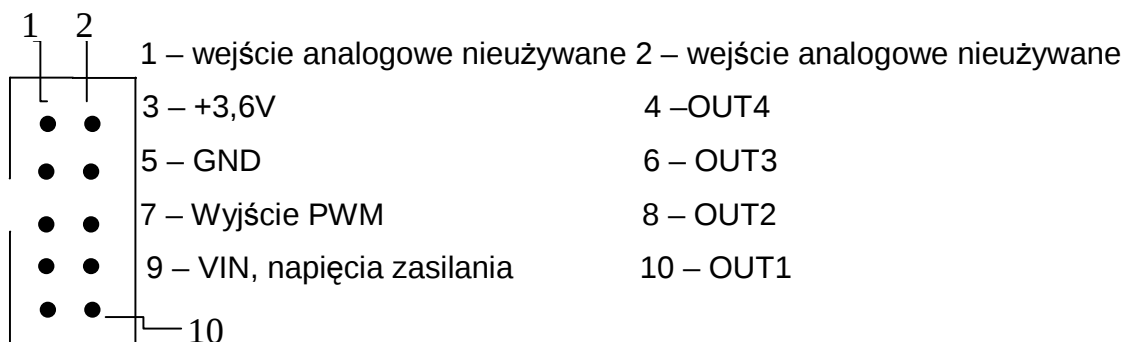
NO – styk normalnie otwarty

CON – styk wspólny

NZ – styk normalnie zamknięty

UWAGA: Pomimo że przekaźniki są w stanie przełączać napięcie zmienne 255VAC 10A, to sama płytką nie spełnia wymogów bezpieczeństwa (brak obudowy, uziemienia). Dlatego takie odbiorniki należy podłączać przy pomocy bezpiecznych zewnętrznych przekaźników np. na szynie DIN, sterowanych z przekaźnika znajdującego się na płytce.

OPIS ZŁĄCZA IDC10



PRZYCISK RESETU

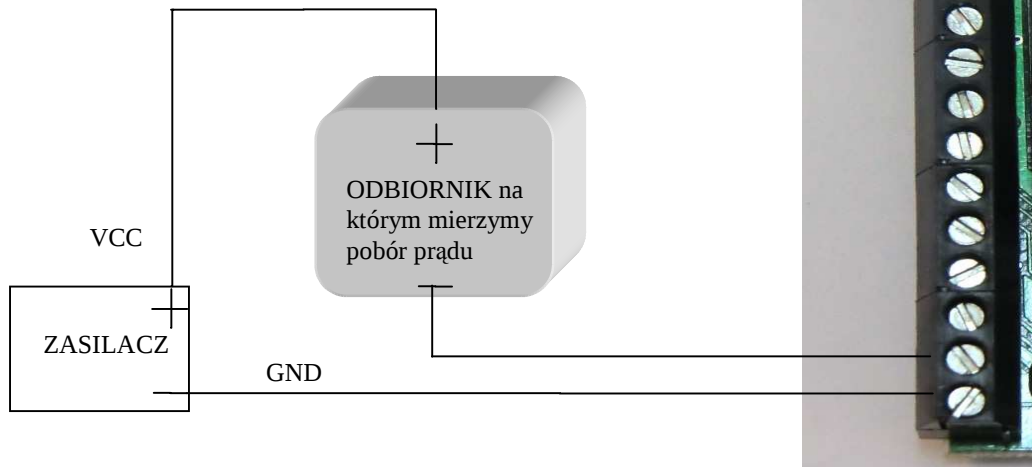
Wciśnięcie na około 0,5 sekundy powoduje zmianę stanu przekaźników na przeciwny, przetrzymanie dłużej do około 5 sekund (gdy nie jesteśmy zalogowani przez WWW na moduł) powoduje reset modułu, dalsze przetrzymanie na około 10 sekund powoduje zmianę wszystkich ustawień (zarówno sieciowych jak i konfiguracyjnych) na domyślne, potwierdzeniem resetu ustawień jest szybkie załączenie i wyłączenie przekaźnika(pyk-pyk), nie mylić z zmianą stanu i wyłączeniem przekaźnika po restarcie.

Użytkownik i hasło: admin

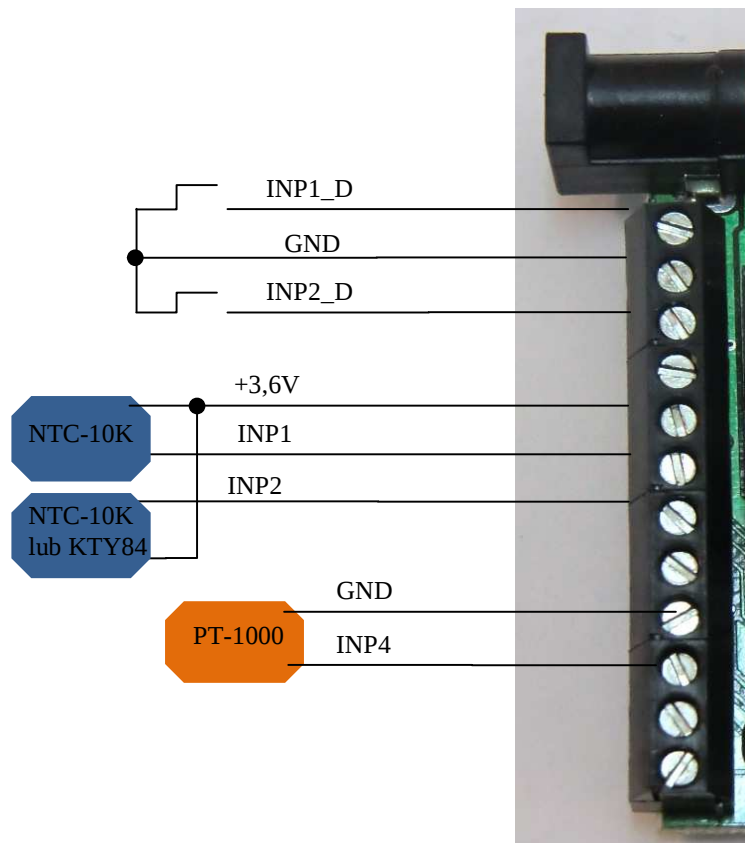
IP: 192.168.1.100

Podpięcie czujników

1. Pomiar prądu.

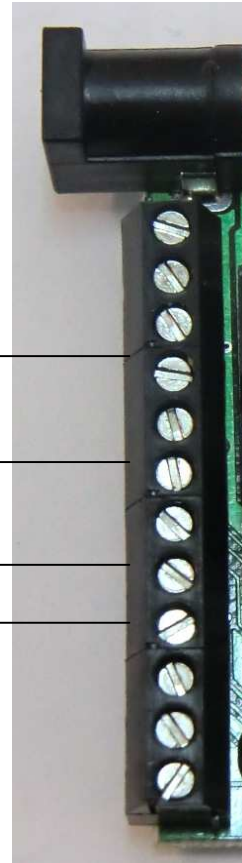
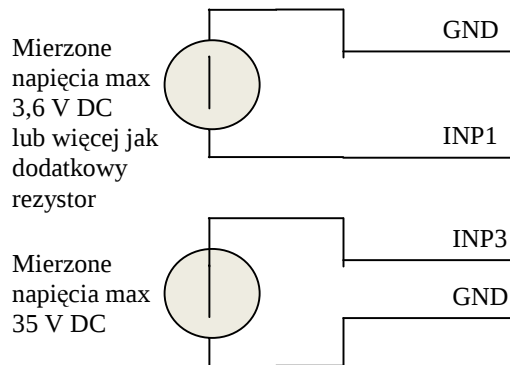


2. Pomiar temperatury, wejścia logiczne.



3. Pomiar napięcia.

INP1 służy do pomiaru temperatury lub napięcia, w panelu sterowania należy wybrać co będzie mierzone



Dodatkowy rezystor do zwiększenia zakresu pomiarowego INP1

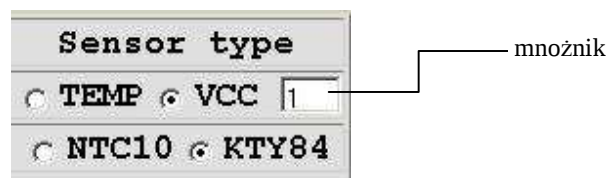


R = 10K zwiększa zakres pomiarowy razy 2 (mnożnik 2) czyli $3,6V \times 2 = 7,2V$

R = 20 K zwiększa zakres pomiarowy razy 3 (mnożnik 3)

R = 30 K zwiększa zakres pomiarowy razy 4 (mnożnik 4) itd.

Dodanie rezystora musi wiązać się z ustawieniem odpowiedniego mnożnika w polu VCC.



Zarządzanie przez WWW.

1.Control Panel

CONTROL PANEL

VCC SUPPLY = 9.4 VBoard Temperature= 25.7°C

Digital Outputs Control

Out0Out1Out2Out3Out4

OFFOFFOFFOFFOFF

1Off2Off3Off4Off5Off

Set State

PWM Output

OFF

Frequency= 5008 Hz 5008

Duty= 50.0 % 50.0

Start

ANALOG Inputs State

Input	Value	Unit	kal	Sensor type
Inp1	0.0	V	0.0	TEMPVCC1
Inp2	N/A	°C	0.0	NTC10KTY84
Inp3	0.0	V	0.0	
Inp4	N/A	°C	0.0	PTSOLAR
Inp5	0.00	A	0.00	
Power measure				
I3*I5	0.00	W		
P*t	0.000	Wh		StartReset

DIGITAL Inputs State

Input1	Input2
HIGH	HIGH

• Pusck red or green circle to change output state or chose "Set State" button.

Soft Version: v1.0

Wciśnięcie przycisku powoduje zmianę stanu wyjścia (OUT0 przekaźnik na module)

Ustawia równocześnie stan wszystkich wyjść zgodnie z ustawieniami w polach wyboru

Załącza generator PWM, przy zmianie częstotliwości lub wypełnienia nie trzeba wyłączać generatora

Wartość kalibracji danego wejścia, dodaje się lub odejmuje żadaną wartość

Pomiar mocy i energii z wejścia INP3 (napięcie) oraz INP5 (prąd)

Wybór rodzaju podłączonego czujnika do odpowiednich wejść

2.Events Config (Tablica zdarzeń)

Events Config									
INPUTS		OUTPUTS/ACTION							
HYSTERESIS		OUT0	OUT1	OUT2	OUT3	OUT4	PWM	E-MAIL	SNMP TRAP
TEMP	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
VCC SUPPLY	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP1 °C	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP2 °C	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP3 V	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP4 °C	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP5 A	0.0	0.0	0.0	0.0	0.0	0.0	0.0 Hz 0.0 %	text	0.0
INP1 DIG		☐	☐	☐	☐	☐	☐ Hz ☐ %	text	☐
INP2 DIG		☐	☐	☐	☐	☐	☐ Hz ☐ %	text	☐

Save Config

- Włączenie danego wejścia
- Wartość danego wejścia przy której nastąpi załączenie danego wyjścia lub generatora PWM lub wysłanie e-maila lub SNMP Trap
- Jeśli zaznaczone to wyjście zostanie załączone przy przekroczeniu wartości w górę (narastanie np. temperatury), w przeciwnym przypadku przełączenie jest przy opadaniu wartości
- Zapisuje ustawienia (włączenia i wyłączenia danego wejścia nie trzeba zapisywać)
- Jeśli zaznaczone to reaguje na zmianę stanu, w przeciwnym przypadku brak reakcji (wyłączone)
- Treść wiadomości, która zostanie wysłana email'em przy wystąpieniu zdarzenia, maksymalna ilość znaków to 89, znak „=„ i „&” nie jest dozwolony

Dla wejść INP1D i INP2D, przy zaznaczonym polu wyboru e-mail i SNMP Trap, powiadomienia przez emial jak i snmp są wysyłane zarówno przy zmianie stanu z wysokiego na niski jak i z niskiego na wysoki, dodatkowo do treści (na końcu) emaila dodawana będzie liczba 1 lub 0 oznaczająca aktualny stan wejścia.

Opis działania tablicy zdarzeń:

Przykład:

Przy wejściu INP1 w polu OUT0 jest ustawiona wartość 20 i zaznaczony checkbox, takie same wartości są ustawione w polu E_MAIL. Wartość początkowa INP1 wynosi 10, po pewnym czasie rośnie temperatura i wartość INP1 wynosi 20 lub więcej, w tym momencie zostanie załączone wyjście OUT0 (jeśli było wyłączone, w przeciwnym wypadku nic się nie zmienia) oraz zostanie wysłana jedna wiadomość e-mail. Po pewnym czasie temperatura zaczyna spadać i jak spadnie poniżej 20, to OUT0 zostanie wyłączone, jeśli temperatura znowu przekroczy 20, to ponownie zostanie załączone OUT0 i wysłany kolejny e_mail, analogicznie dla wszystkich wejść i wyjść.

Jeśli mamy załączone sprawdzanie stanu z kilku czujników to wymuszenie stanu na wyjściach OUTX oraz ustawienie generatora PWM będzie takie jakie było ostatnie zarejestrowane zdarzenie.

3.Watchdog

WatchDog

☐ Enable IP0
IP0 192.168.1.10 OUT0: ☒ ON | ☐ OFF | RESET= 10 s | PING Failures 3

☐ Enable IP1
IP1 192.168.1.10 OUT1: ☒ ON | ☐ OFF | RESET= 10 s | PING Failures 3

☐ Enable IP2
IP2 192.168.1.10 OUT2: ☒ ON | ☐ OFF | RESET= 10 s | PING Failures 3

☐ Enable IP3
IP3 192.168.1.10 OUT3: ☒ ON | ☐ OFF | RESET= 10 s | PING Failures 3

☐ Enable IP4
IP4 192.168.1.10 OUT4: ☒ ON | ☐ OFF | RESET= 10 s | PING Failures 3

Time to wait for respond is 4s

Czas oczekiwania na odpowiedź wynosi 4 sekundy po tym czasie jest naliczany jeden nieodebrany ping. W momencie oczekiwania na odpowiedź inne adresy IP nie są pingowane, co może wydłużyć czas stwierdzenia że dany adres jest nieosiągalny.

Ilość nieodebranych pingów po, których nastąpi jedno ze zdarzeń, w zależności od ustawień będzie to: włączenie (ON) danego wyjścia, wyłączenie (OFF) lub reset (ON/OFF) na określony czas w sekundach (max 65535s).

4.Scheduler

Scheduler

DATE and TIME:Th-1970-01-01;00:00:10

☐ Enable S0
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S1
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S2
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S3
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S4
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S5
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S6
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S7
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S8
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

☐ Enable S9
0,Mo,00:00:00 ☒ ON| ☐ OFF| ☐ RESET= 10

Save Config

Format wpisywania momentu zdarzenia jest następujący, numer wyjścia (od 0 do 4) na którym ma wystąpić zdarzenie, dzień lub dni tygodnia oddzielone przecinkami, oraz czas w formacie xx:xx:xx, zamiast dni tygodnia można wpisać krzyżyki '##' (dwa krzyżyki) i wtedy zdarzenie następuje każdego dnia o zadanej godzinie.

Dni wpisujemy skrótem dwuliterowym z angielskiego, pierwsza litera musi być duża a druga mała, Mo – poniedziałek, Tu- wtorek, We-środa, Th- czwartek, Fr- piątek, Sa- sobota, Su- niedziela. Przykład:

0,Mo,12:23:00 – zadziałanie co poniedziałek o 12:23

1,Sa;Fi,23:22:03 – zadziałanie co sobotę i piątek o 23:22:03

1,##,12:01:30 – zadziałanie w każdy dzień o 12:01:30

Efektom zadziałania może być włączenie przekaźnika, wyłączenie, lub reset (włączenie i wyłączenie) na określony czas w sekundach (max 65535).

4. Network Configuration.

Network Configuration

Email client settings

SMTP Server: Port:

User Name:

Password:

To:

From:

Subject:

When you change setting press "Save Config" before Test

Ustawienia parametrów klienta Email. Po zmianie ustawień i chęci przetestowania klienta należy zapisać ustawienia, przycisk „Save Config”

Network settings

MAC Address:

Host Name:

☐ Enable DHCP

IP Address:

Gateway:

Subnet Mask:

Primary DNS:

Secondary DNS:

ACCESS settings

User:

Password:

Max char 8

NTP settings

NTP Server: Port:

Time Interval

Time Zone

SNMP settings

Read Comm1 :

Read Comm2 :

Read Comm3 :

Write Comm1:

Write Comm2:

Write Comm3:

☒ TRAP Enable

Trap Reciver IP

Trap Comm

Nazwa użytkownika i hasło dostępu do modułu.

Ustawienia serwera NTP, Time Interval - okres w minutach co jaki będzie synchronizowany czas z serwerem.

TRAP Enable – włączenie możliwości wysyłania komunikatów TRAP przez SNMP.

AUTO SEND TRAP settings

☐ Enable Automatic Send TRAP

☒ TEMP

☒ VCC

☒ INP1

☒ INP2

☒ INP3

☒ INP4

☒ INP5

☒ INP1D

☒ INP2D

Time Interval * 10s = 0.17m

Date and Time

NTP ☒

Set Manual ☐

Enable Automatic Send TRAP – włączenie (powyższe TRAP Enable musi być włączone) automatycznego wysyłania komunikatów TRAP (wartość lub stan) z danego wejścia.

Time Interval – okres co jaki będą wysyłane komunikaty, rozdzielczość co 10 sekund (max 10555)

Czas ustawiany indywidualnie lub z serwera NTP, w przypadku ustawienia ręcznego po każdym restarcie urządzenia konieczne ustawienie czasu.

Upgrade oprogramowania.

W przypadku gdy pojawi się nowa wersja oprogramowania lub wersja pod specjalne zastosowanie istnieje możliwość załadowania takiego oprogramowania do urządzenia. Można to zrobić zdalnie przez sieć przy pomocy protokołu TFTP.

W celu załadowania oprogramowania należy zrestartować (opcja „Save config and Reboot” w Network configuration) lub wyłączyć (z zasilania) i ponownie włączyć urządzenie, następnie mamy 5 sekund (miga zielona dioda w gnieździe RJ45) na rozpoczęcie transmisji przez TFTP, jeśli transmisja nie nastąpi urządzenie uruchamia się normalnie (zielona dioda w RJ45 świeci). W przypadku gdy transmisja pliku upgrade nastąpi należy poczekać około 90 sekund na załadowanie oprogramowania. Poprawne załadowanie kończy się komunikatem „Przesłano pomyślnie”. **Plik musi być przesyłany w trybie binarnym**, dla windowsowego tftp wymagana opcja -i, przykład:
tftp -i 192.168.1.100 put „plik_upgrade.bin”.



```
C:\C:\SYSWXP\system32\cmd.exe

C:\>tftp -i 192.168.1.100 put "firmware_lan_1.0.bin"
Przesłano pomyślnie: bajtów: 321664 w 79 ss, bajtów/s: 4071
C:\>_
```

Po poprawnym załadowaniu, urządzenie zrestartuje się i będzie gotowe do pracy.

W przypadku próby wysłania złego pliku dostaniemy komunikat o błędzie „invalid file”



```
C:\C:\SYSWXP\system32\cmd.exe

C:\>tftp -i 192.168.1.100 put "firmware_lan_1.1.bin"
Błąd na serwerze : invalid file
C:\>
```

PLIK MIB do SNMP

LAN_MODULE DEFINITIONS ::= BEGIN

IMPORTS

enterprises, IpAddress, Gauge, TimeTicks FROM RFC1155-SMI
DisplayString FROM RFC1213-MIB
OBJECT-TYPE FROM RFC-1212
TRAP-TYPE FROM RFC-1215;

lan_module OBJECT IDENTIFIER ::= { enterprises 17095 }

product OBJECT IDENTIFIER ::= { lan_module 1 }

setup OBJECT IDENTIFIER ::= { lan_module 2 }

control OBJECT IDENTIFIER ::= { lan_module 3 }

ON-OFF ::= INTEGER { ON(1), OFF(0) }

name OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Name of product. e.g. PICDEM.net etc."

::= { product 1 }

version OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Version string. e.g. 1.0"

::= { product 2 }

date OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Date of version"

::= { product 3 }

out0 OBJECT-TYPE

SYNTAX INTEGER { OFF(1), ON(0) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"OUT0 output set or read"
::= { control 1 }

out1 OBJECT-TYPE
SYNTAX INTEGER { OFF(1), ON(0) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"OUT1 output set or read"
::= { control 2 }

out2 OBJECT-TYPE
SYNTAX INTEGER { OFF(1), ON(0) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"OUT2 output set or read"
::= { control 3 }

out3 OBJECT-TYPE
SYNTAX INTEGER { OFF(1), ON(0) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"OUT3 output set or read"
::= { control 4 }

out4 OBJECT-TYPE
SYNTAX INTEGER { OFF(1), ON(0) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"OUT4 output set or read"
::= { control 5 }

temp OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..15))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Temp value: Air temperature around board"
::= { control 6 }

vcc OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..15))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"VCC value: Input VCC supply board"
::= { control 7 }

inp1 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Inp1 value: temperature from NTC10 or voltage"

::= { control 8 }

inp2 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Inp2 value: temperature from NTC10 or KTY-84"

::= { control 9 }

inp3 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Inp3 value: voltage max 35V DC"

::= { control 10 }

inp4 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Inp2 value: temperature from PT1000 or PT1000 connected to Solar System Controller"

::= { control 11 }

inp5 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Inp2 value: DC current, max 3A"

::= { control 12 }

i3xi5 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"I3xI5 value: Power[W] from Input3 [V] and Input5[A]"

::= { control 13 }

Pxt OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Pxt value: Energy[Wh] from Input3 and Input5 "
::= { control 14 }

inp1Digital OBJECT-TYPE
SYNTAX INTEGER { HIGH(1), LOW(0) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"INP1D Digital state"
::= { control 15 }

inp2Digital OBJECT-TYPE
SYNTAX INTEGER { HIGH(1), LOW(0) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"INP2D Digital state"
::= { control 16 }

traps OBJECT-TYPE
SYNTAX SEQUENCE OF TrapEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Trap table"
::= { setup 1 }

trapEntry OBJECT-TYPE
SYNTAX TrapEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Single trap entry containing trap receiver info."
INDEX { trapReceiverNumber }
::= { traps 1 }

trapEntry ::=

```
SEQUENCE {
    trapReceiverNumber
        INTEGER,
    trapEnabled
        INTEGER,
    trapReceiverIPAddress
        IpAddress,
    trapCommunity
        DisplayString
}
```

trapReceiverNumber OBJECT-TYPE
SYNTAX INTEGER (0.. 4)
ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Index of trap receiver"

::= { trapEntry 1 }

trapEnabled OBJECT-TYPE

SYNTAX INTEGER { Yes(1), No(0) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Indicates if this trap entry is enabled or not."

::= { trapEntry 2 }

trapReceiverIPAddress OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Trap receiver IP address"

::= { trapEntry 3 }

trapCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..7))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Trap community to be used by agent to send trap"

::= { trapEntry 4 }

END