

## 1. Cechy

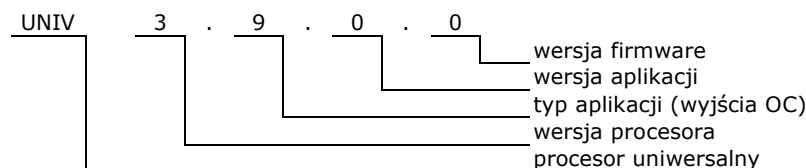
- Oprogramowanie modułu 10 wyjść typu otwarty kolektor UNIV 3.9.0.x
- 3 instrukcje sterujące wyjściami (włącz, wyłącz, zaneguj)
- 3 instrukcje blokujące.
- 10 timer-ów (1 dla każdego wyjścia) 1s-24h opóźniających wykonanie instrukcji
- Umożliwia zdefiniowanie 128 wiadomości CAN, które odebrane przez moduł mogą zmienić jego stan
- Umożliwia określenie stanów początkowych wyjść po włączeniu zasilania modułu.
- Pomiar czasu od włączenia urządzenia
- Monitorowanie stanu zdrowia modułu
- Nadawczy (42 wiadomości) i odbiorczy (42 wiadomości) bufor FIFO do i z magistrali



## 2. Kompatybilność

- Oprogramowanie dla modułu UNIV 3.9.0.x
- Firmware można wgrać do urządzeń z bootloaderem w wersji 3.1 i kompatybilnymi.

## 3. Wersja firmware



## 4. Ramki komunikacyjne (wiadomości)

### 4.1. Budowa ramki wyjść OC

Moduł wysyła na magistralę jedną wiadomość dla każdego kanału w chwili, gdy zmieni się stan jednego z wyjść. Poniższa tabela pokazuje znaczenie poszczególnych bajtów w ramce.

Tabela 1. RAMKA WYJŚĆ (0x309).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x309	3 2 1 0	Node Nr	Group Nr	0xFF	0xFF	KANAŁ	STAN	0xFF	INSTR1	0xFF	TIMER

0x309	- moduł uniwersalny, wyjścia OC	
3	-	- flaga nieużywana o wartości zawsze „0”
2	-	- flaga nieużywana o wartości zawsze „0”
1	-	- flaga nieużywana o wartości zawsze „0”
0	ODP	- flaga odpowiedzi. Jeśli ramka została wysłana w skutek zapytania o stan to ODP = 1, inaczej ODP = 0. ODP = 0 informuje, że stan modułu właśnie się zmienił.
Node Nr	- numer modułu wysyłającego wiadomość	
Group Nr	- numer grupy modułu wysyłającego wiadomość	

KANAŁ	- numer wyjścia: od 0x01 – wyjście 1do 0x0A – wyjście 10
STAN	- aktualny stan wyjścia: 0x00 - wyłączony, 0xFF - włączony
INSTR1	- instrukcja, która oczekuje na wykonanie, lub wartość 0xFF, jeśli żadna nie oczekuje
TIMER	- czas opóźnienia instrukcji oczekującej na wykonanie dla danego kanału, lub 0x00, jeśli żadna nie oczekuje

## 4.2. Pytanie o status

Stan modułu może być sprawdzony poprzez wysłanie ramki PYTANIE O STATUS (0x109) (Tabela 2).

Tabela 2. Ramka PYTANIE O STATUS (0x109).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x109	0x0	KOMP ID1	KOMP ID2	0xXX	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1090 - Ramka PYTANIE O STATUS

KOMP ID1 - identyfikator komputera (musi to być unikalny numer w sieci)

KOMP ID2 - identyfikator komputera (musi to być unikalny numer w sieci)

Node Nr - numer modułu, który jest pytany

Group Nr - numer grupy modułu, który jest pytany

0xXX - dane nieistotne, mogą być dowolnej wartości

W odpowiedzi moduł prześle ramki statusu (jedna dla każdego wyjścia OC) (Tabela 3). Znaczenie bajtów jest identyczne jak dla Tabeli 1.

Tabela 3. Odpowiedź na PYTANIE O STATUS.

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x01	STAN	0xFF	INSTR1	0xFF	TIMER1
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x02	STAN	0xFF	INSTR1	0xFF	TIMER2
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x03	STAN	0xFF	INSTR1	0xFF	TIMER3
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x04	STAN	0xFF	INSTR1	0xFF	TIMER4
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x05	STAN	0xFF	INSTR1	0xFF	TIMER5
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x06	STAN	0xFF	INSTR1	0xFF	TIMER6
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x07	STAN	0xFF	INSTR1	0xFF	TIMER7
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x08	STAN	0xFF	INSTR1	0xFF	TIMER8
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x09	STAN	0xFF	INSTR1	0xFF	TIMER9
0x309	0x1	Node Nr	Group Nr	0xFF	0xFF	0x0A	STAN	0xFF	INSTR1	0xFF	TIMER10

## 4.3. Pytanie o UPTIME

Moduł odpowiada na pytanie o czas jaki upłynął od startu (up time).

Tabela 4. Ramka PYTANIE O UPTIME (0x113).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x113	0x0	KOMP ID1	KOMP ID2	0xXX	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1130 - Ramka PYTANIE O UPTIME

KOMP ID1 - identyfikator komputera (musi to być unikalny numer w sieci)

KOMP ID2 - identyfikator komputera (musi to być unikalny numer w sieci)

Node Nr - numer modułu, który jest pytany

Group Nr - numer grupy modułu, który jest pytany

0xXX - dane nieistotne, mogą być dowolnej wartości

Tabela 5. Ramka odpowiedzi na PYTANIE O UPTIME (0x113).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x113	0x1	Node Nr	Group Nr	0xFF	0xFF	0xFF	0xFF	UPTIME3	UPTIME2	UPTIME1	UPTIME0

0x1131 - Ramka odpowiedzi na PYTANIE O UPTIME

Node Nr - numer modułu sterownika

Group Nr - numer grupy modułu sterownika

UPTIME -  $(UPTIME3*256^3+UPTIME2*256^2+UPTIME1*256^1+UPTIME0*256^0)$  w sekundach

#### 4.4. Pytanie o STAN ZDROWIA modułu

Moduł odpowiada na pytanie o jego kondycję.

Tabela 6. Ramka STAN ZDROWIA – PYTANIE O STATUS (0x115).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x0	KOMP ID1	KOMP ID2	0x01	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1150 – Ramka STAN ZDROWIA

KOMP ID1 – identyfikator komputera (musi to być unikalny numer w sieci)  
KOMP ID2 – identyfikator komputera (musi to być unikalny numer w sieci)

- 0x01 – pytanie o status
- Node Nr – numer modułu, który jest pytany
- Group Nr – numer grupy modułu, który jest pytany
- 0xXX – dane nieistotne, mogą być dowolnej wartości

Tabela 7. Ramki odpowiedzi na STAN ZDROWIA – PYTANIE O STATUS (0x115).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x1	Node Nr	Group Nr	0x01	RXCNT	TXCNT	RXCNTMX	TXCNTMX	CANINTCNT	RXERRCNT	TXERRCNT

0x1151 – Ramka odpowiedzi na STAN ZDROWIA – PYTANIE O STATUS

Node Nr – numer modułu sterownika  
Group Nr – numer grupy modułu sterownika

- 0x01 – ramka 1 (stany od ostatniego włączenia)
- RXCNT – aktualny poziom zajętości bufora odbiorczego FIFO
- TXCNT – aktualny poziom zajętości bufora nadawczego FIFO
- RXCNTMX – maksymalny poziom zajętości bufora odbiorczego FIFO od czasu włączenia modułu
- TXCNTMX – maksymalny poziom zajętości bufora nadawczego FIFO od czasu włączenia modułu
- CANINTCNT – ilość restartów interfejsu CAN
- RXERRCNT – aktualny rejestr błędów odbioru interfejsu CAN
- TXERRCNT – aktualny rejestr błędów nadawania interfejsu CAN

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x1	Node Nr	Group Nr	0x02	0xFF	0xFF	RXCNTMXE	TXCNTMXE	CANINTCNTE	RXERRCNTE	TXERRCNTE

0x1151 – Ramka odpowiedzi na PYTANIE O STAN ZDROWIA

Node Nr – numer modułu sterownika  
Group Nr – numer grupy modułu sterownika

- 0x02 – ramka 2 (maksymalne zarejestrowane stany i zapisane w pamięci eeprom)
- RXCNTMXE – maksymalny zarejestrowany w pamięci nieulotnej poziom zajętości bufora odbiorczego FIFO
- TXCNTMXE – maksymalny zarejestrowany w pamięci nieulotnej poziom zajętości bufora nadawczego FIFO od czasu włączenia modułu
- CANINTCNTE – maksymalna zarejestrowana w pamięci nieulotnej ilość restartów interfejsu CAN
- RXERRCNTE – maksymalny zarejestrowany w pamięci nieulotnej rejestr błędów odbioru interfejsu CAN
- TXERRCNTE – maksymalny zarejestrowany w pamięci nieulotnej rejestr błędów nadawania interfejsu CAN

Aby wyzerować wartości maksymalne zapisane w pamięci eeprom modułu, należy wysłać ramkę jak w Tabeli 8. Moduł nie odpowiada na tę ramkę.

Tabela 8. Ramka STAN ZDROWIA – PYTANIE O WYZEROWANIE (0x115).

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x0	KOMP ID1	KOMP ID2	0x02	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1150 – Ramka STAN ZDROWIA

KOMP ID1 – identyfikator komputera (musi to być unikalny numer w sieci)  
KOMP ID2 – identyfikator komputera (musi to być unikalny numer w sieci)

- 0x02 – pytanie o wyzerowanie
- Node Nr – numer modułu, który jest pytany
- Group Nr – numer grupy modułu, który jest pytany
- 0xXX – dane nieistotne, mogą być dowolnej wartości

## 5. Sterowanie

Moduł może być sterowany bezpośrednio z komputera, lub pośrednio - przez inne moduły. W obu przypadkach można wykorzystać opisane poniżej instrukcje sterujące. Instrukcje blokujące 0xDD-0xDF można wykorzystać tylko przy sterowaniu pośrednim.

### 5.1. Instrukcje sterujące

W tabeli zebrano instrukcje wykonywane przez moduł.

Tabela 9. Kodowanie instrukcji sterujących modułem

Instrukcja	Kod Instrukcji								Opis	Sterowanie	
	INSTR1	INSTR2	INSTR3	INSTR4	INSTR5	INSTR6	INSTR7	INSTR8		Bezpośr.	Pośred.
WYŁĄCZ	0x00	WYJŚCIE 1	WYJŚCIE 2	TIMER	0xXX	0xXX	0xXX	0xXX	Wyłączy wybrane wyjścia, pozostałe pozostaną bez zmian.		
WŁĄCZ	0x01	WYJŚCIE 1	WYJŚCIE 2	TIMER	0xXX	0xXX	0xXX	0xXX	Włączy wybrane wyjścia, pozostałe pozostaną bez zmian.	√	√
ZANEGUJ	0x02	WYJŚCIE 1	WYJŚCIE 2	TIMER	0xXX	0xXX	0xXX	0xXX	Ustawi stan wybranych wyjść na przeciwny, pozostałe pozostaną bez zmian.		

0xXX – bajt może mieć dowolną wartość

WYJŚCIE 1	WYJŚCIE 2	Opis
0x01	0x00	- <00000001> <00000000> - tylko wyjście OC1
0x02	0x00	- <00000010> <00000000> - tylko wyjście OC2
0x03	0x00	- <00000011> <00000000> - wyjście OC1 i OC2
0x04	0x00	- <00000100> <00000000> - tylko wyjście OC3
...	...	
0xFF	0x03	- <11111111> <00000011> - wyjścia OC1-OC10

#### WYJŚCIE1

bit <0> - wyjście OC1, bit <1> - wyjście OC2, bit <2> - wyjście OC3, bit <3> - wyjście OC4, bit <4> - wyjście OC5, bit <5> - wyjście OC6, bit <6> - wyjście OC7, bit <7> - wyjście OC8

#### WYJŚCIE2

bit <0> - wyjście OC9, bit <1> - wyjście OC10, bit <2> - nieużywany, bit <3> - nieużywany, bit <4> - nieużywany, bit <5> - nieużywany, bit <6> - nieużywany, bit <7> - nieużywany

TIMER	Opis
0x00	- instrukcja zostanie wykonana natychmiast
0x01	- instrukcja zostanie wykonana z opóźnieniem 1s
...	
0xFF	- instrukcja zostanie wykonana z opóźnieniem 24h

Instrukcja	Kod	BoksX	BoksY	INSTR1	INSTR2	INSTR3	INSTR4	INSTR5	INSTR6	INSTR7	INSTR8	Opis	Bezpośr.	Pośred.
ODBLOKUJ BOKS	0xDD	BoksX	BoksY	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	Odblokuje wybrane boksy – boksy zostaną sprawdzone przy odebraniu kolejnej wiadomości z magistrali		√
ZABLOKUJ BOKS	0xDE	BoksX	BoksY	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	Zablokuje wybrane boksy – boksy zostaną pominięte przy odebraniu kolejnej wiadomości z magistrali		√
ZANEGUJ BLOKADĘ	0xDF	BoksX	BoksY	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	Zaneguje blokadę tzn. zablokuje boksy jeśli były odblokowane i vice versa		√

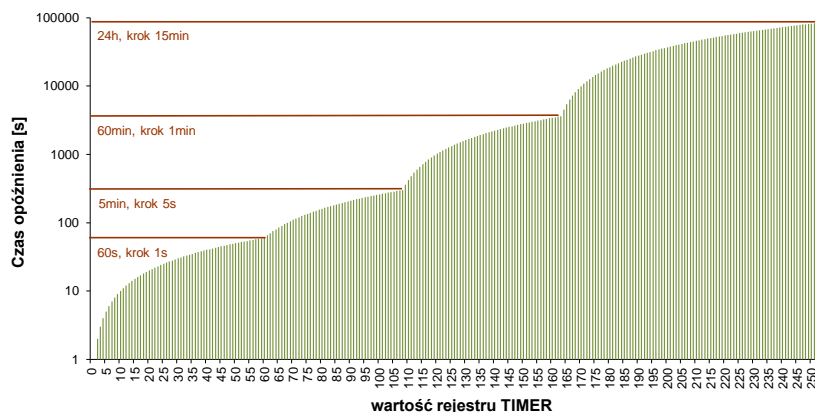
0xXX – bajt może mieć dowolną wartość

BoksX	Opis
0x00	- od Boku 1
0x01	- od Boku 2
...	
0x7F	- od Boku 128

BoksY	Opis
0x00	+ 0 -(i żaden inny)
0x01	+ 1 -(i jeden następny)
...	
0x7F	+127 -(i 127 następnych)

## 5.2. Timer

Wykonanie wszystkich instrukcji (oprócz blokujących 0xDD-0xDF) może być opóźnione wykorzystując timer-y. Każdy kanał posiada własny niezależny timer. Czas opóźnienia może być ustawiony w zakresie 1s - 24h. Poniższy wykres pokazuje zależność czasu opóźnienia od wartości rejestru TIMER.



Rysunek 1. Zależność czasu opóźnienia od wartości rejestru TIMER.

## 5.3. Sterowanie bezpośrednie

Sterując bezpośrednio należy wysłać na magistralę, ciąg bajtów jak w Tabeli 10. W odpowiedzi moduł wyśle ramki (0x109) aktualnych stanów.

Tabela 10. Ramka STEROWANIE BEZPOŚREDNIE (0x10A)

Typ ramki	Flagi	Moduł	Grupa	D0	D1	D2	D3	D4	D5	D6	D7
0x10A	0x0	KOMP ID1	KOMP ID2	INSTR1	INSTR2	Node Nr	Group Nr	INSTR3	INSTR4	INSTR5	INSTR6

0x10A0 - ramka zawierająca instrukcję do wykonania

KOMP ID1 - identyfikator komputera (musi to być unikalny numer w sieci)

KOMP ID2 - identyfikator komputera (musi to być unikalny numer w sieci)

Node Nr - numer modułu, który jest sterowany

Group Nr - numer grupy modułu, który jest sterowany

INSTR1-6 - 6 bajtów instrukcji do wykonania

## 5.4. Sterowanie pośrednie

Przy sterowaniu pośrednim moduł będzie reagował na wiadomości wysłane przez inne moduły w sieci. To, które wiadomości mają oddziaływać na moduł, określa się podczas konfiguracji zależności komunikacyjnych. Zależności te zdefiniowane są w 128 boksach (komórkach pamięci).

Sterowanie pośrednie posiada możliwość ustawienia prostych warunków wykonania instrukcji. Wykorzystuje się do tego instrukcje blokujące jak w Tabeli 9. Instrukcje blokujących nie można użyć w sterowaniu bezpośrednim.

## 6. Konfiguracja

Poniższe parametry modułu mogą być skonfigurowane w tej wersji firmware. Konfigurację wykonuje się używając oprogramowania HAPCAN Programator.

### 6.1. Identyfikator modułu

Każdemu modułowi w sieci HAPCAN musi być przypisany unikalny numer. Numer ten składa się z dwóch bajtów, numeru modułu (1 bajt) i numeru grupy (1 bajt). Przynależność modułu do konkretnej grupy może mieć znaczenie przy programowaniu systemu. Na przykład niektóre urządzenia mogą reagować na wiadomość wysłaną przez dowolny moduł w danej grupie.

### 6.2. Opis modułu

W celu ułatwienia użytkownikowi identyfikowania modułu w sieci, może on zostać opisany 16 znakami.

### 6.3. Nazwy wyjść

Każde wyjście może być opisane 32 znakami.

### 6.4. Stany początkowe wyjść po włączeniu zasilania

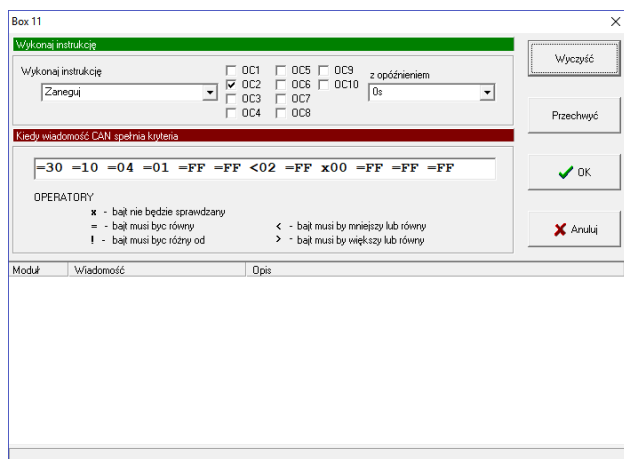
Dla każdego wyjścia indywidualnie można określić stan, jaki przyjmie po włączeniu modułu (np. po zaniku zasilania). Wyjście może być wyłączone, włączone lub przyjąć ostatni stan zapisany w nieulotnej pamięci. Zapis stanu wyjść do nieulotnej pamięci następuje po 6s od wykonania instrukcji. Moduł wysyła na magistralę wiadomości o stanie początkowym wyjść.

## 6.5. Notatka tekstowa.

Do 1024 znaków tekstowych może być zapisanych do pamięci procesora. Mogą to być np. uwagi do konfiguracji modułu

## 6.6. Zależności komunikacyjne

Do programowania zależności pomiędzy modułami służy sterowanie pośrednie, czyli sterownie wiadomościami pochodzącymi od innych modułów. Każdy moduł podczas zmiany jego stanu (np. wciśnięcia lub zwolnienia przycisku w module przycisk) wysyła informację, która może być wykorzystana do sterowania tym samym modulem, a także do sterowania innymi modułami znajdującymi się na magistrali. W tym celu moduł posiada 128 boksów (komórek pamięci), do których można wpisać wiadomości, na jakie moduł ma reagować, kiedy odbierze je z magistrali. Każdy boks zawiera informację o tym, jaka wiadomość ma zainicjować działanie oraz jaka instrukcja ma być wykonana kiedy ta wiadomość zostanie odebrana.



Na rysunku obok zaprogramowano instrukcję negowania aktualnego stanu wyjścia OC2 w momencie odebrania wiadomości pochodzącej od modułu przycisk o ID (4,1) – patrz na opis ramki przycisk w dokumencie „Opis oprogramowania” dla modułu przycisk.

Bajty wiadomości zapisane są w formacie szesnastkowym.

Operatory przed bajtami informują, że moduł zareaguje na wiadomość odebraną z magistrali, w której ten bajt:

- x - może mieć dowolną wartość
- = - będzie identyczny z tutaj podanym
- ! - będzie różny od tutaj podanego
- < - będzie mniejszy lub równy tutaj podanemu
- > - będzie większy lub równy tutaj podanemu

W tym przykładzie:

- =30 =10 - wiadomość od modułu przycisk
- =04 =01 - ID modułu to (4,1)
- =FF =FF - te bajty nie są wykorzystywane w module przycisk i są zawsze równe 0xFF
- <02 - numer przycisku mniejszy lub równy 2, czyli 1 lub 2
- =FF - przycisk wciśnięty
- x00 - jest to bajt informujący o stanie diody LED w module przycisk. Ponieważ nie chcemy uzależnić wykonania instrukcji od tego bajtu, więc może on mieć dowolną wartość.
- =FF =FF =FF - te bajty nie są wykorzystywane w module przycisk i są zawsze równe 0xFF

W konsekwencji moduł wykona instrukcje kiedy odbierze wiadomość od modułu przycisk o ID (4,1), że przycisk 1 lub 2 został wciśnięty. Instrukcja wykona się niezależnie od tego czy dioda LED w module przycisk świeci się, czy nie.

Każda własna i odebrana z magistrali wiadomość informująca o zmianie stanu modułu jest sprawdzana z każdym aktywnym (odblokowanym) boksem w kolejności od boku nr 1 do 128. W związku z tym można ustawić reakcję na tę samą wiadomość w dwóch lub więcej boksach. Na przykład w pierwszym instrukcja może wykonać się bez opóźnienia, a w drugim kolejna instrukcja po definiowanym czasie. Ważne jest zachowanie logicznej kolejności wykonywania instrukcji, aby uzyskać pożądane działanie modułu. Wywołanie instrukcji, która ma wykonać się bez opóźnienia anuluje ustawioną dotychczas instrukcję z opóźnieniem.

Boksy można dynamicznie aktywować (odblokowywać) i dezaktywować (zablokować). Służą do tego instrukcje blokujące (Tabela 9). W ten sposób można kontrolować jakie boksy mają być sprawdzane po odebraniu wiadomości.

## 7. Licencja



Oprogramowanie sprzętowe Projektu Automatyki Domowej HAPCAN, Copyright (C) 2018 [hapcan.com](http://hapcan.com)

open source

To jest wolne oprogramowanie. Możesz je modyfikować i rozprowadzać na warunkach licencji GNU General Public License, opublikowanej przez Free Software Foundation, w wersji 3 lub (według Twojego wyboru) w jakiegokolwiek wersji późniejszej.



Niniejsza oprogramowanie rozpowszechniane jest z nadzieją, iż będzie one użyteczne, jednak BEZ JAKIEJKOLWIEK GWARANCJI, również bez gwarancji PRZYDATNOŚCI HANDLOWEJ albo PRZYDATNOŚCI DO OKREŚLONYCH CELÓW. W celu uzyskania bliższych informacji zobacz pełną treść licencji GNU GPL.

Powinieneś być otrzymać licencję GNU GPL razem z oprogramowaniem. Jeśli nie spójrz na <<http://www.gnu.org/licenses/>>.

## 8. Wersja dokumentu

Plik	Opis	Data
univ_3-9-0-0a_pl.pdf	Wersja oryginalna	Marzec 2018