



PODSTAWY PROGRAMOWANIA W CLAD

**Standard:
Wydanie:**

**CLAD-11
1**

CLAD jest językiem drabinkowym. Format ten jest najpowszechniej stosowanym językiem programowania mikro-sterowników. Drabinka jest tak powszechnie przyjętym językiem programowania, że niemal każdy programista, bez względu na kraj i gałąź przemysłu, potrafi zrozumieć i napisać w niej program. Ponieważ swoim formatem przypomina obwody elektryczne, nawet osoba, która nie jest programistą, ale posiada wiedzę z podstaw elektryki, będzie potrafiła dokonać analizy programu.

Procedury w języku CLAD zajmują kolejne komórki drabiny procedur. Program sterownika realizuje kolejne procedury znajdujące się w rzędzie, po czym przychodzi do niższego rzędu i powtarza operację.

PODSTAWOWE POJĘCIA

Aktory	- obiekty fizyczne sterowane (włączane, wyłączane bądź regulowane) przez węzły systemu automatyki. Przykładem aktorów są: oświetlenia, rolety, piece, wentylatory.
Sensory	- obiekty fizyczne dostarczające informacji od otoczenia bądź od użytkownika. Przykładem sensorów są: styczniki, wyłączniki, czujniki temperatury, oświetlenia itp.
Węzeł (Node)	- zdalne urządzenie zarządzane przez sterownik. Realizują rozkazy sterownika (włącz/wyłącz) na obiektach typu aktor oraz zbiera informację z sensorów. Węzeł może posiadać jedno lub więcej wejść i wyjść.
Numer węzła	- numer z zakresu od 1-99 identyfikujący węzeł w systemie. Numer przydzielany jest w Liście węzłów poprzez przypisanie numeru ID urządzenia do konkretnego numeru listy.
Numer portu	- numer wejścia lub wyjścia urządzenia węzłowego.
Wejście	- fizyczny punkt podłączenia aktorów lub sensorów.
Scena	- układ stanów aktorów określony przez użytkownika systemu. Przykładem sceny może być aranżacja oświetlenia w pomieszczeniu. W tym przypadku zmieniając scenę można zmieniać układ włączonych punktów świetlnych.
Timer	Wirtualny licznik odliczający nastawiony czas. W czasie odliczania stan licznika wynosi 0. Po odmierzeniu czasu stan licznika zmienia się na 1.

Log.

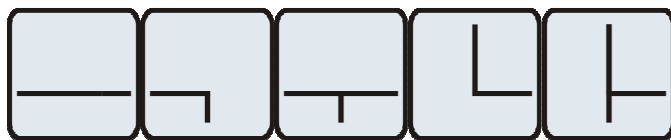
+ |

- ()



Anl.

PROCEDURY LOGICZNE



Łączniki

Procedury połączeń pomiędzy komórkami.



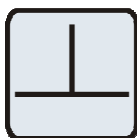
LOG1 – Procedura logicznej jedyńki

Procedura przekazuje do procedury znajdującej się po prawej stronie wartość 1. Procedura stale wpisana w pierwszej kolumnie drabinki.



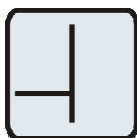
NOT – Procedura negacji

Procedura przekazuje do procedury znajdującej się po prawej stronie wartość przeciwną do stanu wejściowego procedury.



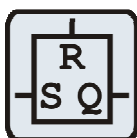
SUL– Suma logiczna

Procedura wykonuje sumę logiczną stanów lewego i górnego stopnia drabinki w stosunku do procedury i przekazuje wartość do procedury znajdującej się po prawej stronie.



SUL2– Suma logiczna

Procedura wykonuje sumę logiczną stanów lewego i górnego stopnia drabinki w stosunku do procedury i przekazuje wartość do procedury znajdującej się na niższym stopniu drabinki.



PRS - Przerzutnik RS

Zmiana stanu na wysoki na wejściu „S” ustawia stan wysoki na wyjściu „Q”. Zmian stanu na wysoki na wejściu „R” ustawia stan niski na wyjściu „Q”.



PTQ -Przerzutnik TQ

Zmiana stanu logicznego wejścia na wysoki przy niskim stanie wyjścia ustawia wyjście na logiczną jedynekę. Przy wysokim stanie wyjścia na logiczne zero.



TLI1 –Impuls jednocyklowy

Zmiana stanu logicznego wejścia na wysoki wywoła wygenerowanie stanu wysokiego na czas jednego cyklu programu.

Log.

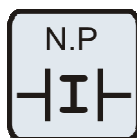
+ |

- ()



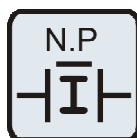
Anl.

ODCZYT STANÓW



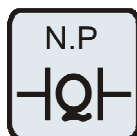
RIN - Stan wejścia węzła

Jeżeli stan wejścia numer „P” węzła „N” jest wysoki, procedura przekazuje wartość wejścia na wyjście procedury.



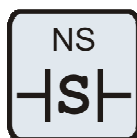
NRIN – Zanegowany stan wejścia węzła

Jeżeli stan wyjścia numer „P” węzła „N” jest wysoki, procedura przekazuje zanegowaną wartość wejścia na wyjście procedury.



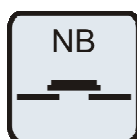
ROUT - Stan wyjścia węzła

Jeżeli stan wyjścia numer „P” węzła „N” jest wysoki, procedura przekazuje wartość wejścia na wyjście procedury.



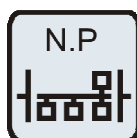
RSC - Stan sceny

Jeżeli stan sceny numer „NS” jest wysoki, procedura przekazuje wartość wejścia na wyjście procedury.



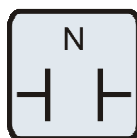
RBT - Wirtualny przycisk ekranowy

Jeżeli zostanie naciśnięty przycisk numer „NB” na ekranie dotykowym, procedura przekaże wartość wejścia na wyjście procedury. Wysoki stan logiczny na wyjściu procedury niezależnie od czasu trzymania wirtualnego przycisku trwa jeden cykl programowy.



MODBUS_RC – Odczyt komórki Modbus

Jeżeli stan wejścia lub wyjścia dwustanowego (w zależności od konfiguracji procedury) numer „P” urządzenia o numerze identyfikacyjnym „N” podłączonego do magistrali Modbus jest wysoki, procedura przekazuje wartość wejścia na wyjście procedury.



LINKED – Stan połączenia z modulem

Procedura sprawdza aktualny stan komunikacji z modulem. Jeżeli sterownik jest połączony z modulem, procedura przekazuje wartość wejścia na wyjście procedury.

Procedura sygnalizuje zerwanie komunikacji po 30 sekundach od zerwania komunikacji z modulem.

Procedura jest blokowana przez 30 sekund od uruchomienia sterownika lub o ile nie nastąpiło wcześniej nawiązanie komunikacji z modulem.

Log.

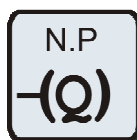
+ |

-()



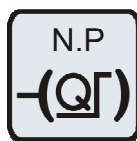
Anl.

FORSOWANIE STANÓW



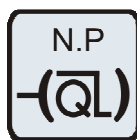
WOUT - Ustawienie wyjścia węzła

Procedura ustawia wartość z wejścia procedury (0/1) na porcie numer „P” węzła o numerze „N” przy każdej zmianie stanu wejścia. Procedura przekazuje wartość wejścia na wyjście.



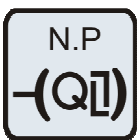
WOUTS - Ustawienie wyjścia węzła

Procedura ustawia wartość „1” na porcie numer „P” węzła o numerze „N” przy zmianie stanu wejścia z 0 na 1. Procedura przekazuje wartość wejścia na wyjście.



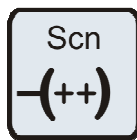
WOUTR - Ustawienie wyjścia węzła

Procedura ustawia wartość „0” na porcie numer „P” węzła o numerze „N” przy zmianie stanu wejścia z 0 na 1. Procedura przekazuje wartość wejścia na wyjście.



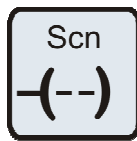
WOUTC - Ustawienie wyjścia węzła na stan przeciwny

Procedura zmienia stan na porcie numer „P” węzła o numerze „N” na przeciwny (0 na 1, 1 na 0) przy zmianie stanu wejścia z 0 na 1. Procedura przekazuje wartość wejścia na wyjście.



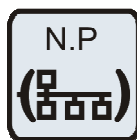
INCSC - Inkrementacja numeru sceny

Zmiana stanu logicznego na wejściu procedury z 0 na 1 dezaktywuje bieżącą scenę oraz aktywuje scenę o wyższym numerze porządkowym. Jeżeli bieżąca scena jest ostatnią, procedura nie wykonuje żadnej operacji. Procedura przekazuje wartość wejścia na wyjście.



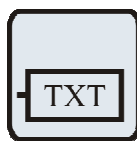
DECSC - Dekrementacja numeru sceny

Zmiana stanu logicznego na wejściu procedury z 0 na 1 dezaktywuje bieżącą scenę oraz aktywuje scenę o niższym numerze porządkowym jeżeli taka istnieje. Procedura przekazuje wartość wejścia na wyjście.



MODBUS_WC – Zapis komórki Modbus

Procedura ustawia wartość z wejścia procedury (0/1) na wyjściu numer „P” urządzenia Modbus o adresie „N” przy każdej zmianie stanu wejścia. Procedura przekazuje wartość wejścia na wyjście.



TXT – Etykieta tekstowa

Procedura wyświetla etykietę tekstową na ekranie sterownika przy zmianie stanu wejścia z 0 na 1. Łącząc procedury w szereg można stworzyć złożone etykiety tekstowe. Procedura przekazuje wartość wejścia na wyjście.

Log.

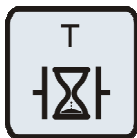
+ |

- ()



Anl.

PROCEDURY CZSOWE



TIME - Ustawienie i aktywacja timera

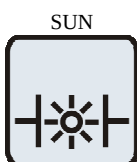
Procedura opóźnia przeniesienie wysokiego stanu wejściowego na wyjście o czas T. Licznik procedury zerowany jest logicznym stanem niskim wejścia. Parametr T określa czas w sekundach.



CLK - Alarm czasowy

Procedura przenosi stan z wejścia jeżeli aktualnie spełniony jest warunek czasowy T.A.

Symbol „T” oznacza typ procedury CLK (np.: M- alarm miesięczny).



Symbol „T”	Jednostka A	Zakres „A”
M	Miesiąc	1÷12
D	Dzień	1÷31
h	Godzina	0÷23
m	Minuta	0÷59
s	Sekunda	0÷59
-	Okres (w godzinach)	Od 0÷23 do 0÷23
Sun *	Jasno	Astronomiczny dzień
Moon *	Ciemno	Astronomiczna noc

* - do poprawnego funkcjonowania procedur Sun i Moon konieczne jest ustawienie współrzędnych geograficznych oraz poprawnego czasu i daty w konfiguracji sterownika.



DAY- Alarm tygodniowy

Procedura przenosi stan z wejścia jeżeli aktualnie spełniony jest warunek dnia tygodnia „DAY”

Log.

+ | -

- ()



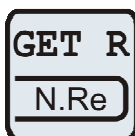
Anl.

Procedury analogowe



CONST – Analogowa wartość stała

Procedura przekazuje do prawej komórki drabinki programu stałą wartość analogową równą „VAL”. Stała może przyjmować wartości z zakresu -3000.9 do + 3000.9.



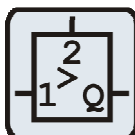
GETR – Odczyt wartości analogowej z węzła

Procedura pobiera wartość analogową z rejestru numer „Re” węzła „N” i przekazuje do prawej komórki drabinki programu.



SETR – Zapis wartości analogowej do węzła

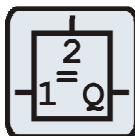
Procedura przesyła do rejestru numer „Re” węzła „N” wartość analogową z wejścia procedury (lewej komórki drabinki programu).



KOMP – Warunek porównania 1>2

Procedura wystawia na wyjście Q wartość logiczną „1” jeżeli wartość wejścia 1 jest większa od wartości wejścia 2. W przeciwnym przypadku Q=0.

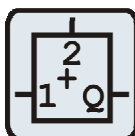
Q – wartość przekazywana do prawej komórki drabinki programu,
1 – wartość pobierana z lewej komórki drabinki programu,
2 – wartość pobierana z wyższej komórki drabinki programu.



EQU – Warunek równości

Procedura wystawia na wyjście Q wartość logiczną „1” jeżeli wartości wejść 1 i 2 są sobie równe. W przeciwnym przypadku Q=0.

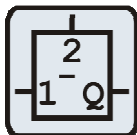
Q – wartość przekazywana do prawej komórki drabinki programu,
1 – wartość pobierana z lewej komórki drabinki programu,
2 – wartość pobierana z wyższej komórki drabinki programu.



SUM – Suma

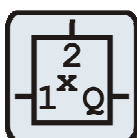
Procedura wykonuje operacji dodawania wartości wejść 1 i 2. Wynik przekazywany jest do wyjścia Q.

Q – wartość przekazywana do prawej komórki drabinki programu,
1 – wartość pobierana z lewej komórki drabinki programu,
2 – wartość pobierana z wyższej komórki drabinki programu.

**SUB – Różnica**

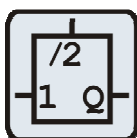
Procedura wykonuje operacji odejmowania od wartości wejścia 1 wartości wejścia 2. Wynik przekazywany jest do wyjścia Q.

Q – wartość przekazywana do prawej komórki drabinki programu,
 1 – wartość pobierana z lewej komórki drabinki programu,
 2 – wartość pobierana z wyższej komórki drabinki programu.

**MUL – Mnożenie**

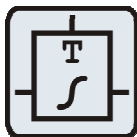
Procedura wykonuje operacji mnożenia wartości wejść 1 i 2. Wynik przekazywany jest do wyjścia Q.

Q – wartość przekazywana do prawej komórki drabinki programu,
 1 – wartość pobierana z lewej komórki drabinki programu,
 2 – wartość pobierana z wyższej komórki drabinki programu.

**DIV – Dzielenie**

Procedura wykonuje operacji dzielenia wartości wejścia 1 przez wartości wejścia 2. Wynik przekazywany jest do wyjścia Q.

Q – wartość przekazywana do prawej komórki drabinki programu,
 1 – wartość pobierana z lewej komórki drabinki programu,
 2 – wartość pobierana z wyższej komórki drabinki programu.

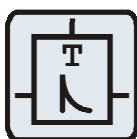
**INT – Całkowanie**

Procedura wykonuje operację całkowania wartości wejścia względem czasu T (w sekundach). Wynik przekazywany jest do wyjścia.

Procedura realizuje algorytm:

$$y = y_{Tn} = y_{Tn-1} + (x - y_{Tn-1}) \Delta t / T$$

- gdzie y_{Tn} i y_{Tn-1} wartości "pamiętane" odpowiednio w bieżącym i poprzednim cyklu, a Δt różnica czasu między bieżącym a poprzednim cyklem obliczeń

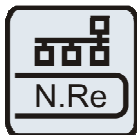
**DIF – Różniczkowanie**

Procedura wykonuje operację różniczkowania wejścia względem czasu T (w sekundach). Wynik przekazywany jest do wyjścia.

Procedura realizuje algorytm:

$$y_{pn} = (x - y_{pn-1}) * \Delta t / T$$

- gdzie: y_{pn-1} - wartość pamiętana w cyklu poprzednim, a Δt - różnica między bieżącym a poprzednim cyklem obliczeń.

**MODBUS_RR – Odczyt rejestru Modbus**

Procedura pobiera wartość analogową z rejestru wejściowego lub wyjściowego (w zależności od konfiguracji procedury) numer „Re” urządzenia Modbus o adresie „N” i przekazuje do prawej komórki drabinki programu.

Uwaga!

Wartość rejestru Modbus równa 1 odpowiada w systemie Ceuron wartości 0,1. Wartość 100 odpowiada wartości 10 w systemie Ceuron itd.

**MODBUS_WR – Zapis rejestru Modbus**

Procedura przesyła do rejestru numer „Re” urządzenia Modbus o adresie „N” wartość analogową z wejścia procedury (lewej komórki drabinki programu).

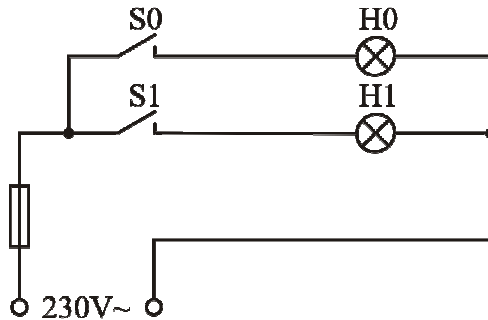
Uwaga.!

Wartość rejestru Modbus równa 1 odpowiada w systemie Ceuron wartości 0,1. Wartość 100 odpowiada wartości 10 w systemie Ceuron itd.

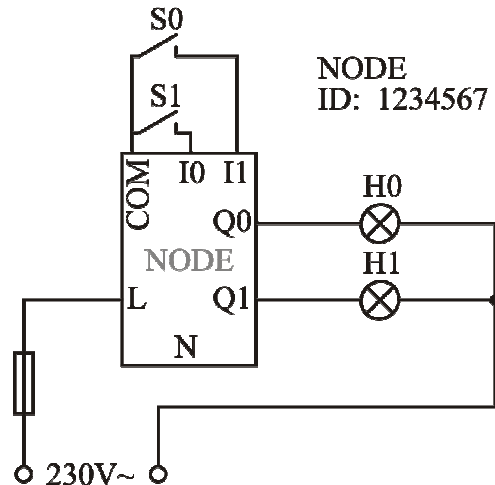
PRZYKŁAD NR 1

Załączenie oświetlenia podwójnym włącznikiem.

Założmy, że potrzebujemy zrealizować przedstawiony na **rysunku 1.1** układ załączania żarówek H0 oraz H1. Układ często spotykany jest jako podwójny włącznik (S1 i S2) zapalający dwie lampy.



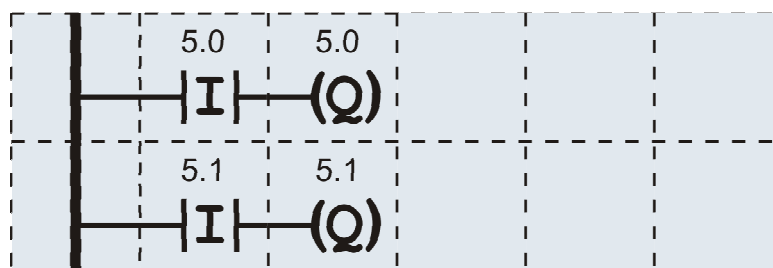
Rys. 1. 1 Schemat elektryczny załączenia żarówek H0 i H1.



Rys. 1. 2 Schemat elektryczny załączenia żarówek H0 i H1 z zastosowaniem węzła Ceuron.

Oprogramowanie sterownika

Pierwszym etapem zaprogramowania nowego węzła jest wprowadzenie ID węzła do „NODE listy”. W naszym przykładzie przypiszemy ID (1234567) do numeru 5 „NODE listy”. Kolejnym etapem będzie wprowadzenie kodu programu.



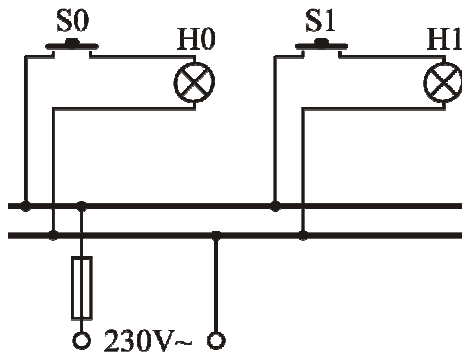
Rys. 1. 3 Program realizujący algorytm przykładu nr 1.

Załączenie przełącznika S0 spowoduje przekazanie stanu z lewej komórki do prawej (w tym przypadku przekazanie stanu logicznej jedynki) co z kolei spowoduje przekazanie wysokiego stanu do węzła 5 (ID 1234567) na port 0 i załączenie żarówki H0. Analogicznie wygląda sytuacja styku S1.

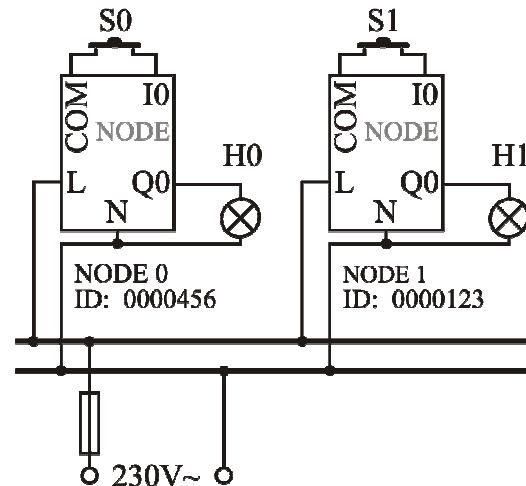
PRZYKŁAD NR 2

Załączenie oświetlenia łącznikami monostabilnymi.

Założmy, że potrzebujemy zrealizować przedstawiony na **rysunku 2.1** układ załączania żarówek H0 oraz H1. Punkty świetlne w odróżnieniu od poprzedniego przykładu załączane są za pomocą włączników monostabilnych (takich jak włącznik dzwonkowy). W tym przykładzie włączniki znajdują się w różnych puszkach instalacyjnych.



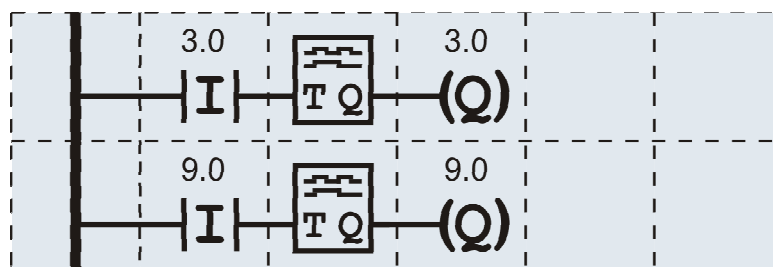
Rys. 2. 1 Schemat elektryczny załączenia żarówek H0 i H1 łącznikami monostabilnymi.



Rys. 2. 2 Schemat elektryczny załączenia żarówek H0 i H1 z zastosowaniem dwóch węzłów Ceuron.

Oprogramowanie sterownika

Pierwszym etapem zaprogramowania nowego węzła jest wprowadzenie ID węzłów do „NODE listy”. W naszym przykładzie przypiszemy ID (0000456) do numeru 3 a ID (0000123) do węzła nr 9 „NODE listy”. Kod programu wygląda następująco:



Rys. 2. 3 Program realizujący algorytm przykładu nr 2.

Załączenie przełącznika S0 spowoduje załączenie przerzutnika TQ (procedura PTQ) i przekazanie stanu wysokiego do procedury WOUT co z kolei spowoduje załączenie na portu 0 węzła 3 (ID 0000456) i załączenie żarówki H0. Analogicznie wygląda sytuacja styku S1.

PRZYKŁAD NR 3

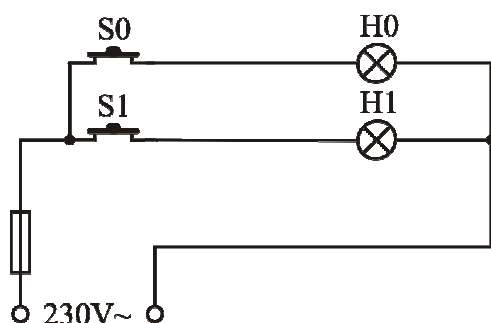
Sceny świetlne

Założmy, że potrzebujemy zrealizować przedstawiony na **rysunku 3.1** układ załączania żarówek H0 oraz H1 w różnych konfiguracjach scen świetlnych:

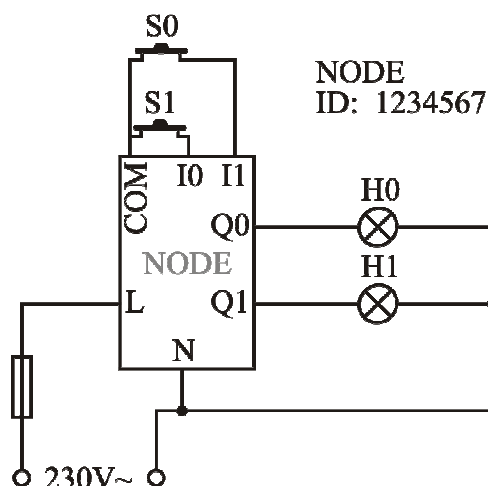
Scena 1 – Załączona żarówka H0.

Scena 2 – Załączone żarówki H0 oraz H1.

Scena 3 – Załączona żarówka H1.



Rys. 3. 1 Schemat elektryczny załączenia żarówek H0 i H1 podwójnym łącznikiem monostabilnym przed zastosowaniem węzła Ceuron.



Rys. 3. 2 Schemat elektryczny załączenia żarówek H0 i H1 z zastosowaniem węzła Ceuron.

Oprogramowanie sterownika

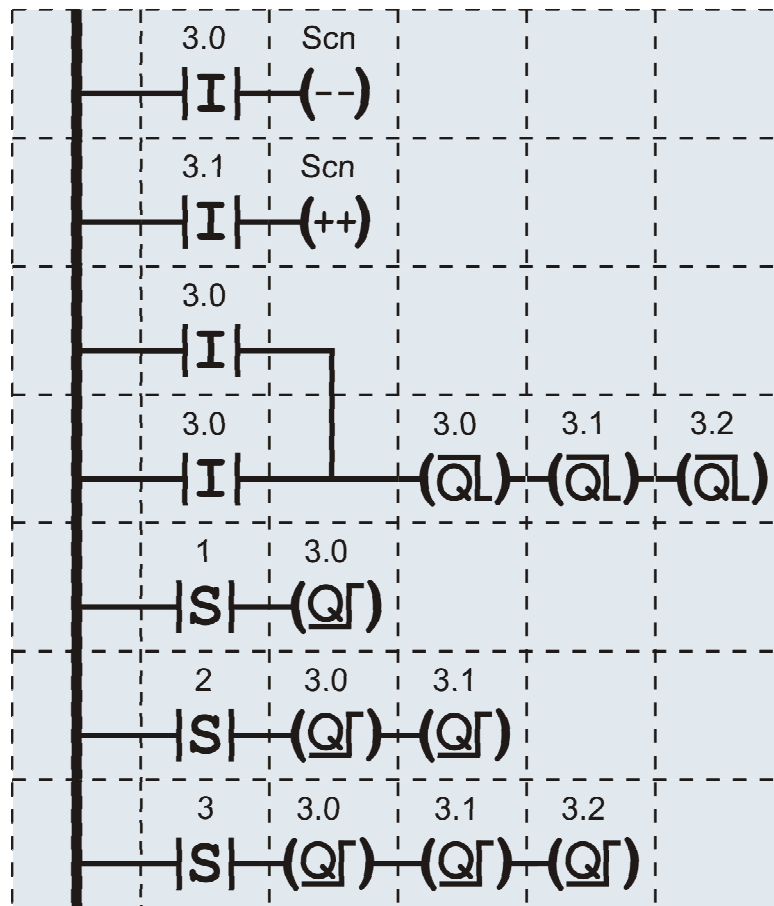
W pierwszym etapie wprowadzamy ID węzła do „NODE listy”. W naszym przykładzie przypiszemy ID (1234567) do numeru 3 „NODE listy”. Kolejnym etapem będzie wprowadzenie kodu programu.

Inkrementacja i dekrementacja sceny przy użyciu łącznika.

Włącznik S0 reprezentowany przez procedurę RIN powoduje dekrementację numeru sceny (procedura DECSC). Naciśnięcie włącznika S1 wywołuje inkrementację numeru sceny.

Komponowanie scen.

Załączenie sceny numer 1 powoduje przekazanie stanu z lewej komórki odpowiedniej procedury RSC do prawej. Pozostaje tylko dołączyć odpowiednio ustawione procedury WOUT i otrzymujemy gotową scenę świetlną. Program został pokazany na **rysunku 3.3**.



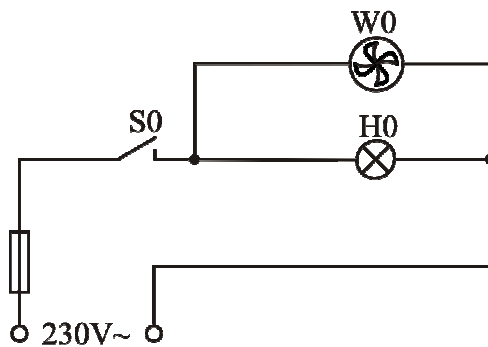
Rys. 3. 3 Program realizujący algorytm przykładu nr 3.

Procedury RIN inkrementujące i dekrementujące numer sceny mogą być zastąpione przez procedury RBT. Sterowanie odbywać się wtedy będzie przez wirtualne przyciski ekranowe sterownika.

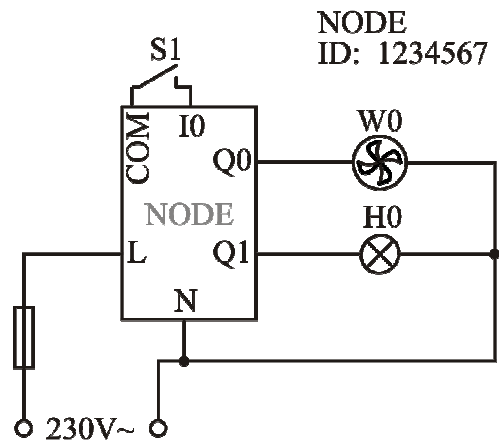
PRZYKŁAD NR 4

Wykorzystanie timera do sterowania wentylatorem.

Założmy, że potrzebujemy zrealizować przedstawiony na **rysunku 4.1** układ załączania żarówki H0 oraz wentylatora W0. Wentylator będzie włączany 120 sekund po załączeniu oświetlenia a wyłączany 240 sekund po wyłączeniu oświetlenia. Układ stosowany do inteligentnego uruchamiania wentylatora teoalety.



Rys. 4. 1 Schemat elektryczny załączenia żarówki oraz wentylatora przed zastosowaniem systemu Ceuron.



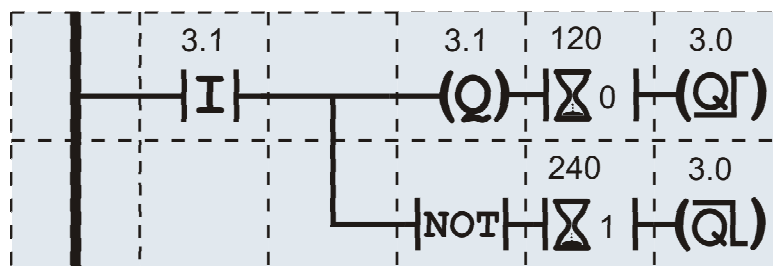
Rys. 4. 2 Schemat elektryczny załączenia żarówki H0 i wentylatora W1 z zastosowaniem wężła Ceuron.

Oprogramowanie sterownika

Pierwszym etapem zaprogramowania nowego wężła jest wprowadzenie ID wężła do „NODE listy”. Przypisujemy ID (1234567) do numeru 3 „NODE listy”. Kolejnym etapem będzie wprowadzenie kodu programu z **rysunku 4.3**.

Załączenie przełącznika S0 spowoduje włączenie portu numer 1 wężła numer 3 (załączenie oświetlenia), oraz odliczania timera numer 0 na 120 sekund. Po odliczeniu 120 sekund stan timera zostanie ustawiony na logiczną jedynkę i przekazany na wyjście uruchamiając procedurę WOUT. W rezultacie wysterowane zostanie wejście numer 0 wężła numer 3 i zostanie włączony wentylator.

Wyłączenie przełącznika S0 spowoduje wyłączenie portu numer 1 wężła 3 (wyłączenie oświetlenia) oraz przekazanie wartość zero na wejście procedury NOT która dokona inwersji i uruchomi odliczanie timera numer 1 na czas 240 sekund. Efektem odliczenia ustawionego czasu przez timer T2 będzie wyłączenie portu numer 0 wężła 3 (wyłączenie wentylatora).



Rys. 4. 3 Program realizujący algorytm przykładu wykorzystania timera.

PRZYKŁAD NR 5

Zmienne dwustanowe i analogowe

Zmienne dwustanowe służą w do tymczasowego przechowywania informacji bądź przekazywania stanu w inne miejsce programu. W systemie Ceuron zmienne dwustanowe oznaczane są przedrostkiem „v” przed numerem zmiennej.

Rysunek 5.1 pokazuje wykorzystanie zmiennych dwustanowych do przekazania informacji o stanie wejścia. W przykładzie port numer 0 wężła numer 5 za pośrednictwem zmiennej numer 8 łączy wyjście 0 wężła 17.

Rys. 5. 1 Program wykorzystujący zmienną dwustanową

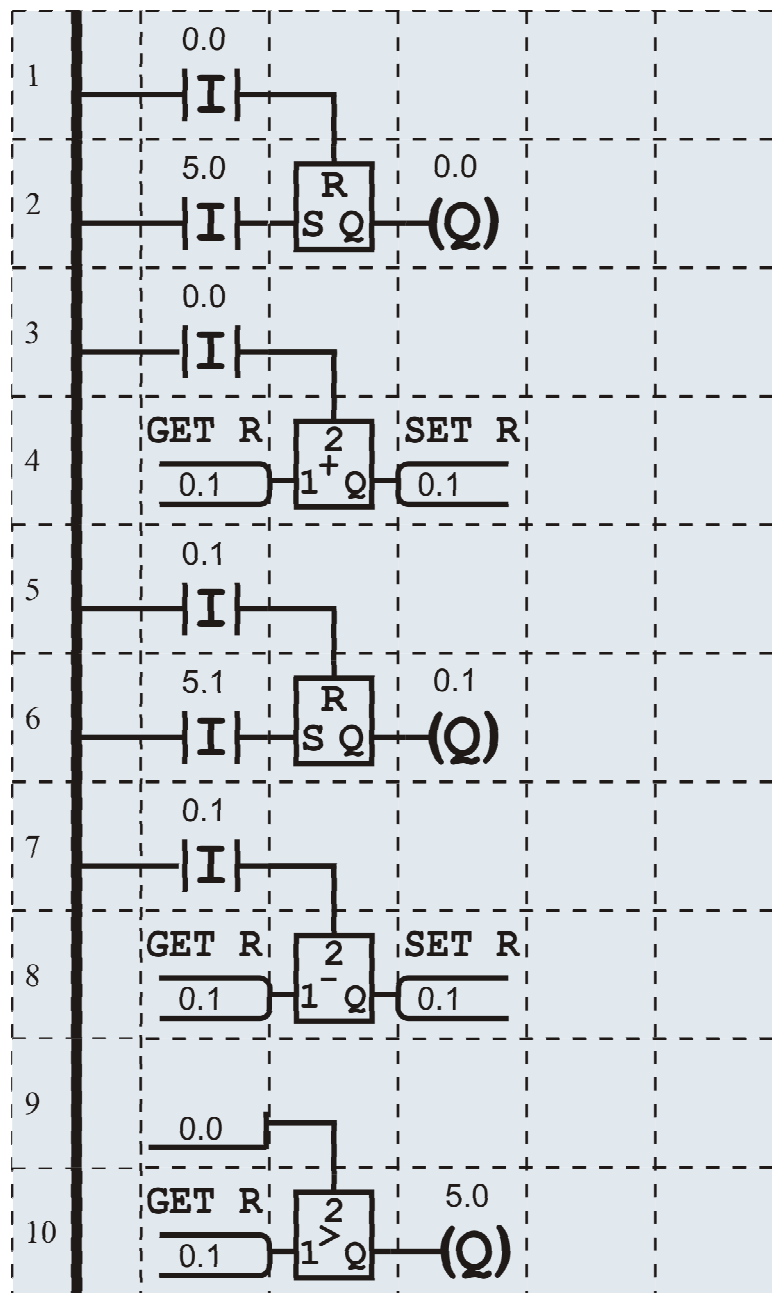
PRZYKŁAD NR 6

Licznik wejść – inteligentny włącznik światła

Założmy, że potrzebujemy zrealizować inteligentny włącznik światła uzależniony od obecności osób w pomieszczeniu. Światło włącza się kiedy do pomieszczenia wejdzie pierwsza osoba a wyłącza kiedy wyjdzie ostatnia. Osoby wchodzące wykrywa fotokomórka A podłączona do wejścia 0 węzła 5 a osoby wychodzące wykrywa fotokomórka B podłączona do wejścia 1 węzła 5. Poniżej przedstawione są trzy sposoby wykonania zadania.

Sposób 1.

Algorytm wykorzystujący przerzutnik RS oraz procedurę KOMP przedstawiony został na **rysunek 6.1**.



Rys. 6.1 Program wykorzystujący przerzutnik RS.

Rzędy 1, 2 pokazują sposób generowania impulsu o czasie trwania jednego cyklu programu za pomocą procedury przerzutnika. Pojawienie się stanu wysokiego na wejściu przerzutnika RS powoduje ustawienie jego wyjścia na wartości „1” i jej zapis to zmiennej numer 0.

Przy kolejnym cyklu programu, stan wysoki zmiennej numer 0 wywoła reset przerzutnika i ustawienie zmiennej numer 0 na logiczne „0”. Kolejne ustawienie „1” na wyjściu przerzutnika uwarunkowane jest przejściem wejścia procedury przerzutnika przez stan niski ponieważ jest on inicjowany zmianą stanu z niskiego na wysoki a nie poziomem.

W ten sposób uzyskany został generator impulsu na zmiennej numer 0 zależny od zmiany wejścia 0 wężła 5. Algorytm został powtórzony w rzędach 5, 6 dla generacji impulsu na zmiennej numer 1 zależnego od zmiany wejścia 1 wężła 5.

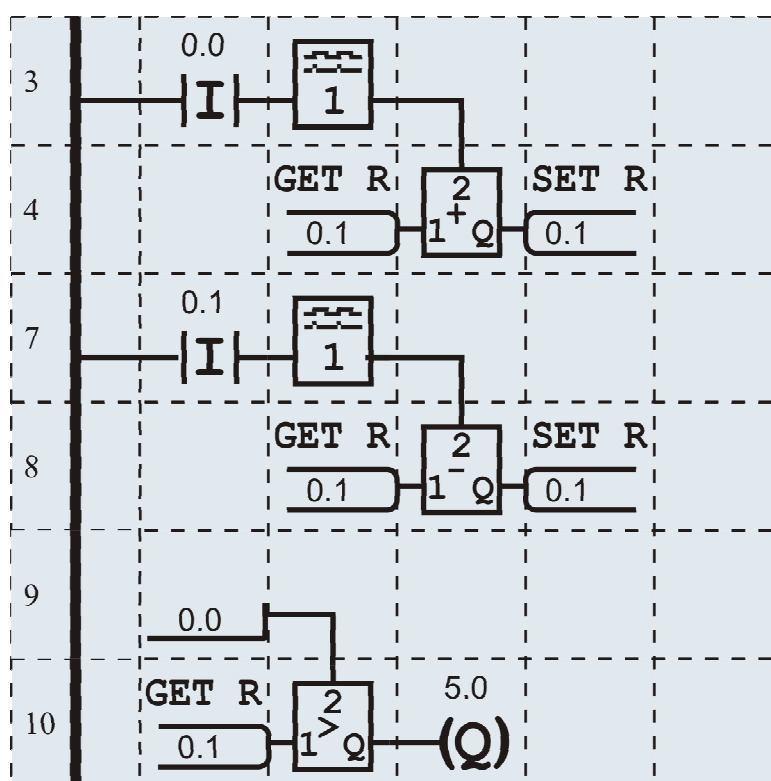
W rzędach 3,4 oraz 7,8 utworzone zostały liczniki impulsów. Impulsy dwustanowej zmiennej 0 dodawane są do zmiennej analogowej numer 1 w (rzędach 3,4) a w rzędach 7,8 impulsy zmiennej dwustanowej numer 1 odejmowane są od zmiennej analogowej numer.

Algorytm ten wykorzystuje własności języka CLAD w którym logiczne „0” zmiennej dwustanowej przy połączeniu ze zmienną analogową rzutowane jest na wartość 0,0 a logiczne „1” na analogową wartość 1,0.

Rzędy 9 i 10 realizują algorytm progu przejść od którego załączany będzie port 0 wężła 5 (w naszym przykładzie oświetlenie).

Sposób 2.

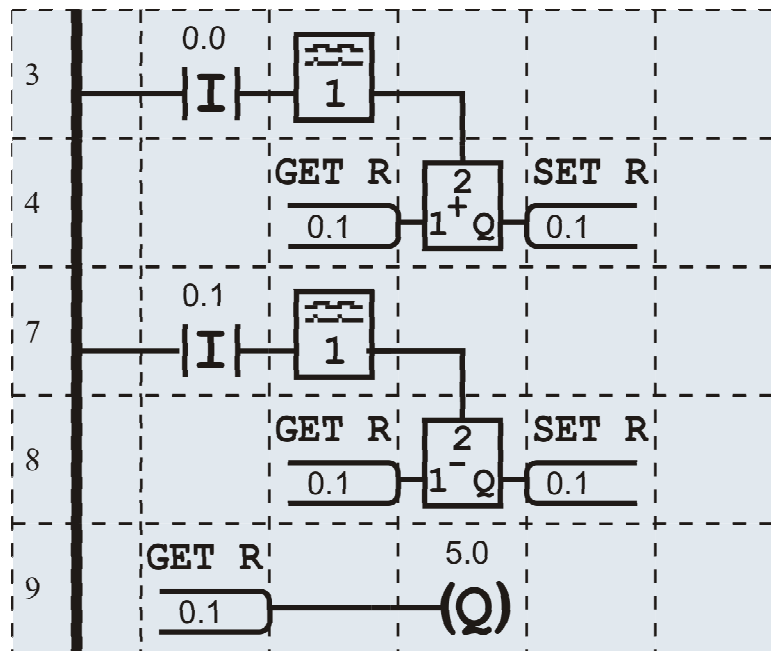
Zamiast skomplikowanego algorytmu generatora impulsów można zastosować procedurę TL11. Algorytm z tym rozwiązaniem został pokazany na **rysunku 6. 2**. Rozwiązanie to jest znacznie prostsze i czytelniejsze.



Rys. 6.1 Program wykorzystujący generator TL11.

Sposób 3.

Nasz przykład wymaga reakcji (włączenia oświetlenia) przy pierwszej wchodzącej osobie czyli przy przekroczeniu wartości 0 na zmiennej analogowej numer 1. Możemy więc wykorzystać kolejną własność rzutowania zmiennych. Zmienna dwustanowa przyjmuje wartość logicznej jedynki jeżeli rzutowana jest na nią wartość analogowa większa od 0,0. Pokazane to zostało w dziewiątym rzędzie **rysunku 6.3**. Nie jest to rozwiązanie efektywne ale może mieć zastosowanie kiedy wymagane jest skrócenie kodu programu.



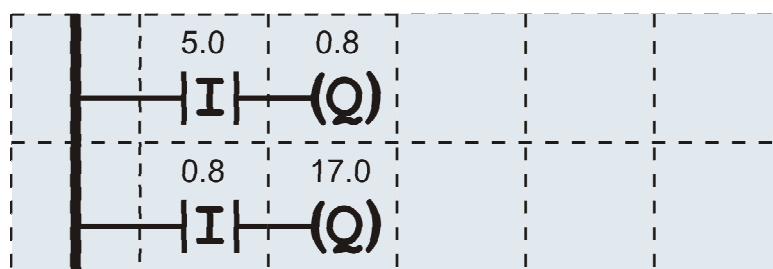
Rys. 6.3 Program wykorzystujący właściwości rzutowania analogu.

PRZYKŁAD NR 7

Przekazywanie informacji pomiędzy sterownikami

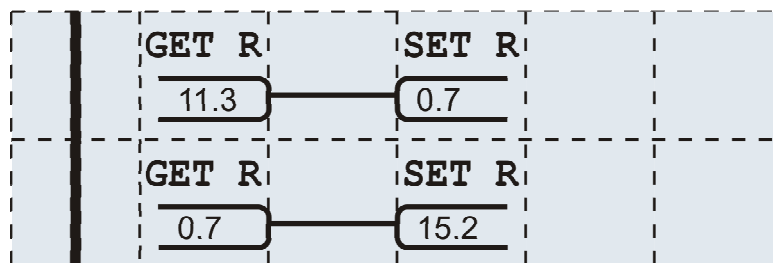
Węzeł numer 0 traktowany jest w systemie Ceuron jako bufor informacyjny do przechowywania informacji wewnętrznych programu oraz do przekazywania informacji pomiędzy sterownikami.

Rysunek 5.1 pokazuje sposób przekazania informacji o stanie wejścia. W przykładzie port numer 0 węzła numer 5 za pośrednictwem węzła 0, portu 8 przekazuje stan do systemu nadrzędnego. Na porcie wejściowym nr 8 węzła 0 pobierane są informacje z systemu nadrzędnego.



Rys. 7. 1 Program wymieniający dane dwustanowe z systemem nadrzędnym

Analogicznie postępuje się przy przekazywaniu danych analogowych. Prezentuje to rysunek 7.2.



Rys. 5.2 Program wymieniający dane analogowe z systemem nadrzędnym

Rejestry analogowe węzłów są wykorzystywane głównie do odczytu informacji typu temperatura, naświetlenie, wilgotność itp.