

# Przełączniki programowalne **NEED**

## Podręcznik użytkownika



© 2007 Relpol S.A.

## SPIS TREŚCI

|  |    |
|--|----|
| 1. WSTĘP .....   | 6  |
| 2. INFORMACJE OGÓLNE .....   | 7  |
| 2.1. Charakterystyka .....   | 7  |
| 2.2. Opis panelu czołowego przekaźnika programowalnego NEED .....          | 7  |
| 2.3. Struktura systemu i numery zamówieniowe .....                         | 9  |
| 3. INSTALACJA.....   | 12 |
| 3.1. Kolejność instalacji .....  | 13 |
| 3.2. Montaż mechaniczny .....  | 13 |
| 3.2.1. Mocowanie na szynie montażowej (DIN 35mm).....                      | 13 |
| 3.2.2. Mocowanie na śruby .....  | 14 |
| 3.3. Zaciski przyłączeniowe, przewody .....                                | 15 |
| 3.4. Podłączenie wejść dyskretnych 230V AC .....                           | 16 |
| 3.5. Podłączenie wejść dyskretnych 24V (12V) DC .....                      | 19 |
| 3.6. Podłączenie wejść analogowych AC.....                                 | 20 |
| 3.7. Podłączenie wejść analogowych 24V (12V) DC .....                      | 22 |
| 3.8. Podłączenie wyjść .....   | 26 |
| 3.9. Podłączenie zasilania AC .....  | 27 |
| 3.10. Podłączenie zasilania 24V (12V) DC .....                             | 28 |
| 4. ZASOBY PRZEKAŹNIKA .....  | 29 |
| 4.1. System przekaźnika programowalnego NEED .....                         | 29 |
| 4.2. Cykl programu .....   | 29 |
| 4.3. Zasoby przekaźnika programowalnego NEED.....                          | 31 |
| 4.4. Wejścia cyfrowe .....   | 33 |
| 4.4.1. Wejścia cyfrowe normalnie otwarte .....                             | 33 |
| 4.4.2. Wejścia cyfrowe normalnie zamknięte .....                           | 33 |
| 4.5. Wyjścia cyfrowe.....  | 33 |
| 4.5.1. Wyjścia cyfrowe normalne .....                                      | 33 |
| 4.5.2. Wyjścia cyfrowe impulsowe .....                                     | 34 |
| 4.5.3. Wyjścia cyfrowe kasujące .....                                      | 34 |
| 4.5.4. Wyjścia cyfrowe ustawiające.....                                    | 35 |
| 4.5.5. Wyjścia cyfrowe normalne wykorzystane do dalszego sterowania .....  | 35 |
| 4.5.6. Wyjścia cyfrowe zanegowane wykorzystane do dalszego sterowania..... | 35 |
| 4.6. Znaczniki .....   | 36 |
| 4.6.1. Znaczniki M1..M16.....  | 36 |
| 4.6.2. Znacznik MDIR .....   | 37 |
| 4.7. Timery .....  | 39 |
| 4.7.1. Timer „Opóźnione załączenie” (ON-DELAYD) .....                      | 40 |
| 4.7.2. Timer “Opóźnione wyłączenie” (OFF-DELAYD) .....                     | 41 |
| 4.7.3. Timer “Pojedynczy impuls” (SINGLE PULSE) .....                      | 41 |
| 4.7.4. Timer “Impulsy” (FLASHING).....                                     | 42 |
| 4.8. Liczniki .....  | 43 |
| 4.9. Zegary Tygodniowe .....   | 45 |
| 4.9.1. Działanie Zegara.....   | 45 |
| 4.9.2. Uwagi dotyczące konfiguracji Zegara.....                            | 54 |
| 4.10. Zegar Czasu Rzeczywistego .....                                      | 57 |
| 4.11. Komparator – wejścia analogowe .....                                 | 58 |
| 4.12. Potencjometr .....   | 64 |
| 4.13. Wartości remanentne przekaźnika programowalnego.....                 | 64 |
| 4.13.1. Uwagi dotyczące wartości remanentnych .....                        | 65 |
| 5. OPIS JĘZYKÓW PROGRAMOWANIA .....  | 68 |
| 5.1. Programowanie w języku tekstowym STL.....                             | 68 |
| 5.1.1. Struktura programu STL.....   | 68 |

|   |     |
|---|-----|
| 5.1.1.1. Nazwy symboliczne.....                               | 71  |
| 5.1.2. Opis Instrukcji STL.....                               | 72  |
| 5.1.2.1. Instrukcja AND.....                                  | 72  |
| 5.1.2.2. Instrukcja nawiasów AND.....                         | 72  |
| 5.1.2.3. Instrukcja AND NOT.....                              | 74  |
| 5.1.2.4. Instrukcja nawiasów AND NOT.....                     | 75  |
| 5.1.2.5. Instrukcja OR.....                                   | 76  |
| 5.1.2.6. Instrukcja nawiasów OR.....                          | 77  |
| 5.1.2.7. Instrukcja OR NOT.....                               | 78  |
| 5.1.2.8. Instrukcja nawiasów OR NOT.....                      | 79  |
| 5.1.2.9. Instrukcja XOR.....                                  | 80  |
| 5.1.2.10. Instrukcja nawiasów XOR.....                        | 80  |
| 5.1.2.11. Instrukcja XOR NOT.....                             | 81  |
| 5.1.2.12. Instrukcja nawiasów XOR NOT.....                    | 82  |
| 5.1.2.14. Instrukcja kasująca R (Reset).....                  | 83  |
| 5.1.2.15. Instrukcja przyporządkowująca =.....                | 83  |
| 5.1.2.16. Instrukcja Przekaznik impulsowy FP.....             | 84  |
| 5.1.2.17. Instrukcje Timerów.....                             | 85  |
| 5.1.2.18. Instrukcje Liczników.....                           | 90  |
| 5.1.2.19. Instrukcje Zegara.....                              | 96  |
| 5.1.2.20. Wejścia analogowe.....                              | 97  |
| 5.1.2.21. Instrukcja ładowania (LOAD).....                    | 98  |
| 5.1.2.22. Instrukcja zawsze ustawiająca SET.....              | 106 |
| 5.1.2.23. Instrukcja zawsze kasująca CLR.....                 | 107 |
| 5.2. Programowanie w języku graficznym LAD.....               | 108 |
| 5.2.1. Symbole w LAD.....                                     | 108 |
| 5.2.2. Wejścia.....   | 109 |
| 5.2.3. Wyjścia.....   | 109 |
| 5.2.4. Struktura programu w LAD.....                          | 110 |
| 5.2.5. Struktura obwodu LAD.....                              | 110 |
| 5.2.6. Opis używanych elementów.....                          | 111 |
| 5.2.7. Konfiguracja.....                                      | 113 |
| 5.2.7.1. Konfiguracja wejść.....                              | 113 |
| 5.2.7.2. Konfiguracja wyjść.....                              | 113 |
| 5.2.7.3. Konfiguracja Znaczników.....                         | 114 |
| 5.2.7.4. Konfiguracja Timerów.....                            | 115 |
| 5.2.7.5. Konfiguracja Liczników.....                          | 116 |
| 5.2.7.6. Przykłady konfiguracji.....                          | 117 |
| 5.2.8. Zasady umieszczania elementów.....                     | 118 |
| 5.2.9. Rodzaje połączeń.....                                  | 119 |
| 5.2.9.1. Odwzorowanie wejścia na wyjście.....                 | 119 |
| 5.2.9.2. Odwzorowanie zanegowanego wejścia na wyjście.....    | 119 |
| 5.2.9.3. Połączenia szeregowo.....                            | 119 |
| 5.2.9.4. Połączenia równoległe.....                           | 120 |
| 5.2.9.5. Połączenie szeregowo-równoległe.....                 | 121 |
| 5.2.10. Nazwy symboliczne.....                                | 122 |
| 5.2.11. Program w LAD.....                                    | 122 |
| 6. INSTALACJA I OPIS OPROGRAMOWANIA.....                      | 123 |
| 6.1. Wymagania sprzętowe.....                                 | 123 |
| 6.2. Instalacja oprogramowania.....                           | 123 |
| 6.3. Deinstalacja.....  | 123 |
| 6.4. Połączenie komputera z przekaźnikiem programowalnym..... | 123 |
| 6.5. Szybki start – tworzenie aplikacji.....                  | 124 |
| 6.6. Praca z programem PC Need.....                           | 129 |
| 6.6.1. Opis głównego okna programu.....                       | 129 |

|   |     |
|---|-----|
| 6.6.2. Pasek Menu.....  | 130 |
| 6.6.3. Pasek narzędzi.....  | 132 |
| 6.6.4. Skróty klawiaturowe.....   | 133 |
| 6.7. Edytor programu STL.....   | 135 |
| 6.7.1. Edytor STL.....  | 135 |
| 6.7.2. Kompilacja STL.....  | 136 |
| 6.8. Edytor programu LAD.....   | 137 |
| 6.8.1. Nowy program.....  | 137 |
| 6.8.2. Zapisanie programu.....  | 138 |
| 6.8.3. Otwarcie istniejącego programu.....                                  | 139 |
| 6.8.4. Wprowadzanie i edycja schematu połączeń.....                         | 139 |
| 6.8.5. Edycja elementu.....   | 143 |
| 6.9. Ustawienia.....  | 146 |
| 6.9.1. Rodzaje ustawień.....  | 146 |
| 6.9.2. Ustawienia Timerów.....  | 148 |
| 6.9.3. Ustawienia Zegarów.....  | 149 |
| 6.9.4. Ustawienia Liczników.....  | 149 |
| 6.9.5. Ustawienia Komparatorów.....   | 150 |
| 6.9.6. Remanencja.....  | 150 |
| 6.9.7. Ustawienia wejść.....  | 151 |
| 6.10. Podgląd zmiennych.....  | 151 |
| 6.11. Podgląd drabinki LAD.....   | 152 |
| 6.12. Hasło.....  | 153 |
| 6.12.1. Wprowadzanie hasła.....   | 154 |
| 6.12.2. Zmiana hasła.....   | 154 |
| 6.13. Zegar czasu rzeczywistego (RTC).....                                  | 155 |
| 6.14. Kod źródłowy.....   | 155 |
| 7. URUCHOMIENIE.....  | 156 |
| 7.1. Załączenie.....  | 156 |
| 7.1.1. Czynności wstępne dla wersji AC.....                                 | 156 |
| 7.1.2. Czynności wstępne dla wersji DC.....                                 | 156 |
| 7.1.3. Załączenie zasilania.....  | 156 |
| 8. INFORMACJE SPRZĘTOWE.....  | 157 |
| 8.1. Zasilanie przekaźnika.....   | 157 |
| 8.1.1. Zasilanie przekaźnika 115/230 V AC.....                              | 157 |
| 8.1.2. Zasilanie przekaźnika 24 (12) V DC.....                              | 157 |
| 8.2. Wejścia.....   | 158 |
| 8.2.1. Wejścia 230 V AC.....  | 158 |
| 8.2.2. Wejścia 24 (12) V DC.....  | 160 |
| 8.3. Wyjścia.....   | 160 |
| 8.4. Opóźnienie wejść.....  | 161 |
| 8.4.1. Opóźnienia wejść dla przekaźnika NEED-230AC-x1-.. ..                 | 161 |
| 8.4.2. Opóźnienia wejść dla przekaźnika NEED-24DC-x1-.., NEED-12DC-x1-.. .. | 163 |
| 8.5. Opóźnienie wyjść.....  | 164 |
| 9. PAMIĘĆ ZEWNĘTRZNA.....   | 165 |
| 9.1. Karta pamięci.....   | 165 |
| 9.2. Organizacja pamięci.....   | 166 |
| 9.3. Programowanie pamięci.....   | 166 |
| 9.3.1. Zapis programu.....  | 166 |
| 9.3.2. Zapis ustawień.....  | 167 |
| 9.3.3. Status pamięci EEPROM.....   | 167 |
| 9.3.4. Odczyt ustawień.....   | 167 |
| 9.4. Współpraca karty pamięci z przekaźnikiem NEED.....                     | 168 |
| 10. PRZYKŁADY ZASTOSOWAŃ.....   | 169 |
| 10.1. Ocena wysokości detalu.....   | 169 |

|   |     |
|---|-----|
| 10.2. Drzwi automatyczne .....                              | 175 |
| 10.3. Dzwonki w szkole .....                                | 183 |
| 10.4. Wykrywanie wad detali .....                           | 189 |
| 10.5. Sterowanie ruchem wózków na zakręcie taśmociągu ..... | 193 |
| 10.6. Sterownik oświetlenia i wentylacji .....              | 197 |
| 10.7. Kontrola obciążenia .....                             | 204 |
| 10.8. Sterowanie i zabezpieczenie silnika 3-fazowego .....  | 207 |
| 11. DANE TECHNICZNE .....                                   | 213 |
| 12. SŁOWNIK.....  | 224 |
| 13. INDEKS.....   | 226 |

## 1. WSTĘP

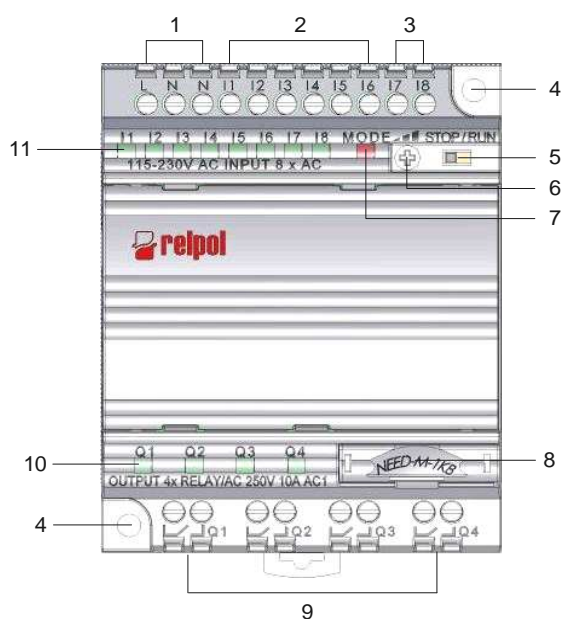
Przełącznik NEED jest przełącznikiem programowalnym, który może zastąpić skomplikowane połączenia przełącznikowe lub stycznikowe. Urządzenie to jest swobodnie programowalne, tzn. – w dowolnym czasie można korygować, zmieniać istniejący program w pamięci sterownika bez zmiany układów peryferyjnych całego otoczenia – co, w przypadku tradycyjnego sterowania przełącznikowego, praktycznie było niemożliwe. Duże możliwości i doskonałe parametry w połączeniu z funkcjonalnością przełącznika programowalnego pozwalają zaoszczędzić czas projektowania a przede wszystkim koszty wdrażanych aplikacji.

## 2. INFORMACJE OGÓLNE

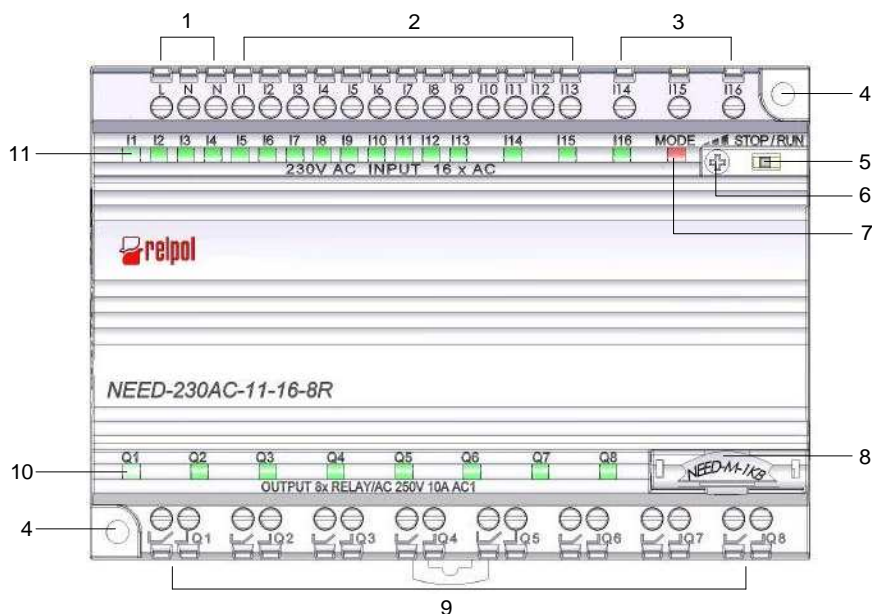
### 2.1. Charakterystyka

- 6 lub 13 wejść cyfrowych,
- 2 lub 3 wejścia cyfrowo-analogowe:  
   NEED-230AC-x1-.. : 0–255V AC,  
   NEED-24DC-x1-.., NEED-12DC-x1-.. :  
     zakres napięciowy: 0–25,5V DC (krok 0,1V) lub 0 – 12,75V (krok 0,05V),  
     zakres prądowy: 0–51mA (krok 0,2mA) lub 0 – 25,5mA (krok 0,1mA),
- 4 lub 8 wyjść przekaźnikowych (230V AC/10A),
- potencjometr do zadawania wartości analogowych,
- zegar czasu rzeczywistego,
- automatyczna zmiana czasu lato/zima w różnych strefach czasowych,
- wskaźnik trybu pracy,
- przełącznik trybu pracy RUN/STOP,
- wskaźniki stanów wejść/wyjść,
- możliwość programowania LAD i STL,
- oprogramowanie PC,
- moduł pamięci zewnętrznej,
- szybki licznik/miernik częstotliwości do 20kHz,
- detekcja prawidłowego kierunku podłączenia faz L1, L2, L3,
- pomiar asymetrii faz L1, L2, L3.

### 2.2. Opis panelu czołowego przekaźnika programowalnego NEED



Rys. 2.2.1. Opis panelu czołowego przekaźnika programowalnego NEED..-x1-08-4.



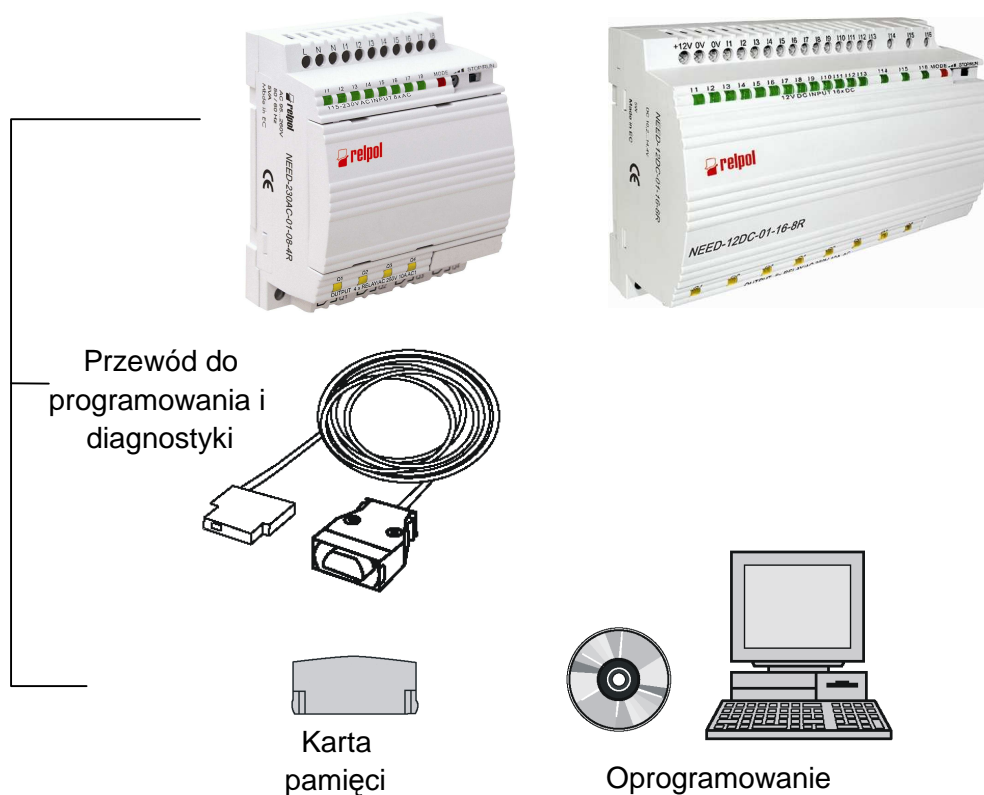
Rys. 2.2.2. Opis panelu czołowego przekaźnika programowalnego NEED..-x1-16-8.

| Oznaczenie | Opis   |
|------------|--|
| 1          | Zaciski śrubowe zasilania  |
| 2          | Zaciski śrubowe wejść cyfrowych                                  |
| 3          | Zaciski śrubowe wejść cyfrowo-analogowych                        |
| 4          | Otwory mocujące  |
| 5          | Przełącznik trybu pracy RUN–STOP                                 |
| 6          | Potencjometr do zadawania wartości analogowych                   |
| 7          | Wskaźnik LED stanu przekaźnika                                   |
| 8          | Złącze do programowania oraz dodatkowego modułu pamięci programu |
| 9          | Zaciski śrubowe wyjść  |
| 10         | Wskaźniki LED stanu wyjść  |
| 11         | Wskaźniki LED stanu wejść  |



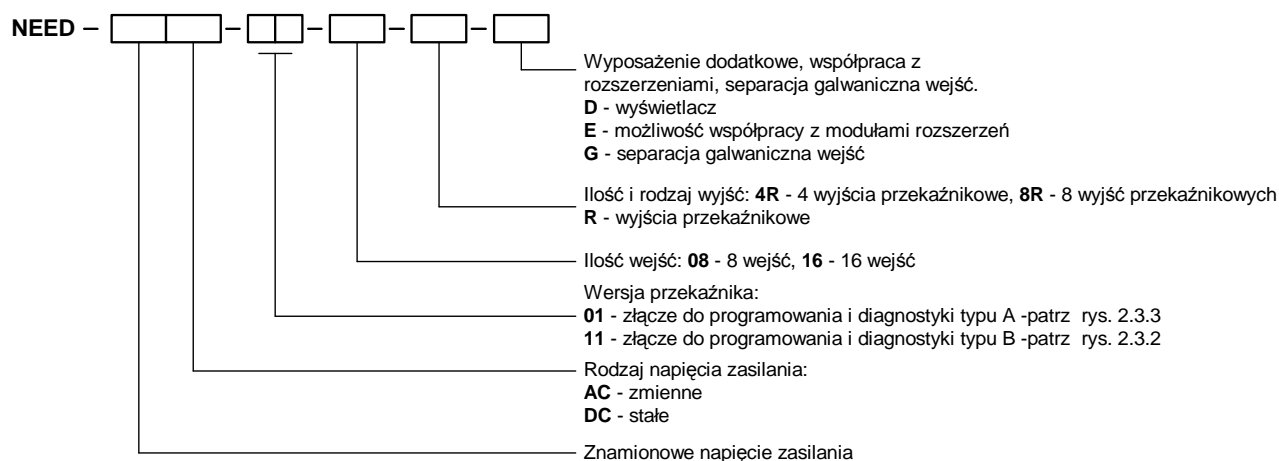
## 2.3. Struktura systemu i numery zamówieniowe

### Przełącznik programowalny NEED



Rys. 2.3.1. Struktura systemu.

### Oznaczenie typu



Przykład1:

*NEED – 230AC – 01 – 08 – 4R*

Przełącznik programowalny NEED – znamionowe napięcie zasilania 230V AC – wersja 01, ze złączem do programowania typu A – 8 wejść – 4 wyjścia przełącznikowe – bez możliwości dołączenia rozszerzeń, wyświetlacza LCD, wejścia bez separacji galwanicznej.

Przykład2:

*NEED – 24DC – 11 – 08 – 4R*

Przełącznik programowalny NEED – znamionowe napięcie zasilania 24V DC – wersja 11, ze złączem do programowania typu B – 8 wejść – 4 wyjścia przełącznikowe – bez możliwości dołączenia rozszerzeń, wyświetlacza LCD, wejścia bez separacji galwanicznej.

Przykład3:

*NEED – 24DC – 11 – 16 – 8R*

Przełącznik programowalny NEED – znamionowe napięcie zasilania 24V DC – wersja 11, ze złączem do programowania typu B – 16 wejść – 8 wyjść przełącznikowych – bez możliwości dołączenia rozszerzeń, wyświetlacza LCD, wejścia bez separacji galwanicznej.



Wersja przełącznika programowalnego bez wyświetlacza wymaga zastosowania kabla do programowania i diagnostyki oraz oprogramowania PC Need.

| Nazwa  | Oznaczenie  |
|--|---|
| Przełącznik programowalny NEED                           | Patrz oznaczenie typu                                   |
| Przewód do programowania i diagnostyki ze złączem typu A | NEED – PC – 15A   |
| Przewód do programowania i diagnostyki ze złączem typu B | NEED – PC – 15B   |
| Karta pamięci ze złączem typu A                          | NEED – M – 1K   |
| Karta pamięci ze złączem typu B                          | NEED – M – 1KB  |
| Oprogramowanie   | NEED – PC Need  |
| Podręcznik użytkownika                                   | Przełącznik programowalny NEED – podręcznik użytkownika |



NEED ze złączem do programowania i diagnostyki typu A może być programowany tylko za pomocą przewodu do programowania typu A. Karta pamięci musi posiadać także złącze typu A. To samo się odnosi do wersji przełącznika NEED ze złączem typu B.

Przykład4:

Przy zamówieniu przełącznika: *NEED – 230AC – 01 – 08 – 4R* można zamówić także:

NEED – PC – 15A – kabel do programowania i diagnostyki ze złączem typu A.

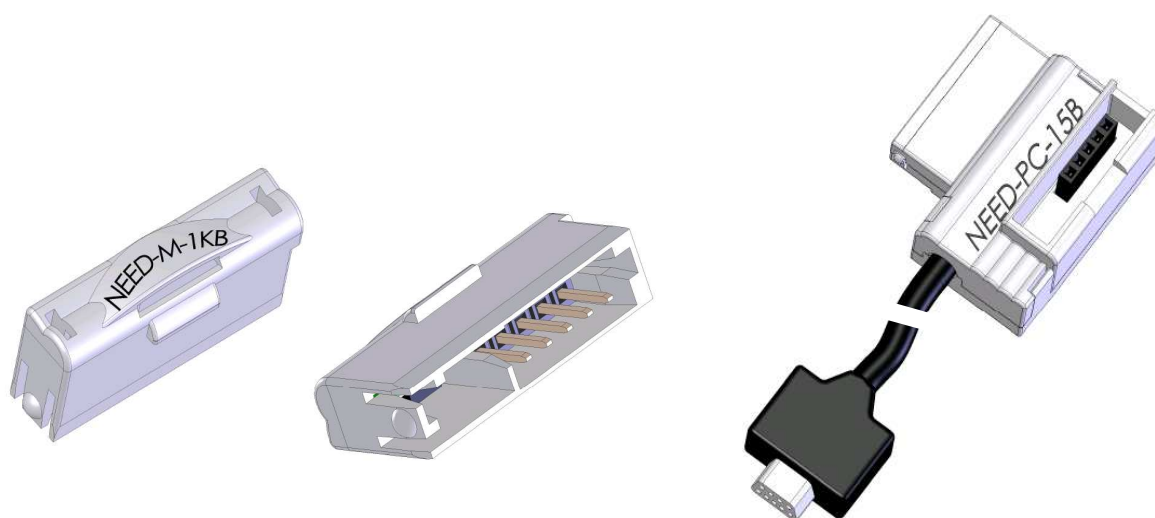
NEED – M – 1K - karta pamięci ze złączem typu A.

Przykład5:

Przy zamówieniu przełącznika: *NEED – 230AC – 11 – 08 – 4R* można zamówić także:

NEED – PC – 15B – kabel do programowania i diagnostyki ze złączem typu B.

NEED – M – 1KB - karta pamięci ze złączem typu B.



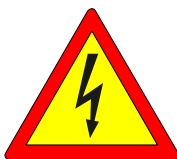
Rys. 2.3.2. Kabel do programowania i diagnostyki oraz pamięć ze złączem typu B.



Rys. 2.3.3. Kabel do programowania i diagnostyki oraz pamięć ze złączem typu A.

### 3. INSTALACJA

Stosowane oznaczenia:



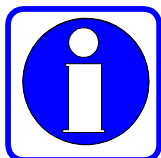
**Niebezpieczeństwo porażenia prądem!**



**Nie prowadzić żadnych prac pod napięciem!**



**Ostrzeżenie!**



**Informacje i wskazówki.**

**Przed instalacją przekaźników programowalnych zapoznaj się z poniższymi uwagami!!!**



***W przekaźniku programowalnym na jego podłączeniach występują napięcia niebezpieczne dla życia ludzkiego.***

- Wyłącz urządzenie/instalację, w którym montujesz przekaźnik programowalny.
- Zabezpiecz urządzenie/instalację przed przypadkowym załączeniem.
- Upewnij się, że żadne napięcie nie występuje w urządzeniu/instalacji.
- W przekaźniku programowalnym ustaw przełącznik w tryb STOP.
- Wykonaj wszystkie konieczne pomiary i sprawdzenia, aby nie doszło do niezamierzonego zadziałania przekaźnika programowalnego.
- Pamiętaj o odprowadzeniu ładunków elektrostatycznych przed dotknięciem przekaźnika.
- Koniecznie podłącz zabezpieczenia przeciwzwarciowe i ochronne.
- Przestrzegaj zasad i zaleceń zawartych w instrukcji użytkownika.
- Montaż przekaźnika programowalnego powinien zostać dokonany przez osobę znającą zasady montażu elektrycznego.
- Pamiętaj, że zainstalowane urządzenia muszą być zabezpieczone przed nieumyślnym uruchomieniem.
- Wszystkie podłączenia przekaźnika programowalnego muszą być zgodne z odpowiednimi normami bezpieczeństwa.
- Parametry sieci zasilającej nie powinny przekraczać granicznych progów tolerancji przedstawionych w danych technicznych przekaźnika.
- Jeżeli stosujesz przekaźnik w układach, gdzie konieczne jest zatrzymanie awaryjne, określ zachowanie się układu w czasie zadziałania i odblokowywania systemu zatrzymania awaryjnego, w celu uniknięcia nieprzewidzianych sytuacji – np. niekontrolowanego startu systemu automatyki.
- Określ poprawność zachowania układu w reakcji na wyłączenie napięcia zasilającego oraz na jego ponowne załączenie.

## Warunki bezpieczeństwa

- W celu zapewnienia bezpieczeństwa obsługi i niezawodnego działania urządzenia montaż przekaźnika programowalnego powinien zostać wykonany przez osobę znającą zasady montażu elektrycznego.
- W czasie montażu należy przestrzegać norm bezpieczeństwa pracy z urządzeniami elektrycznymi i zasad BHP.
- Przestrzegać warunków instalacji przekaźnika programowalnego.

### 3.1. Kolejność instalacji

1. Przygotowanie i zabezpieczenie miejsca instalacji.
2. Montaż mechaniczny.
3. Podłączenie przewodów:
  - podłączenie wejść
  - podłączenie wyjść.
  - podłączenie zasilania.

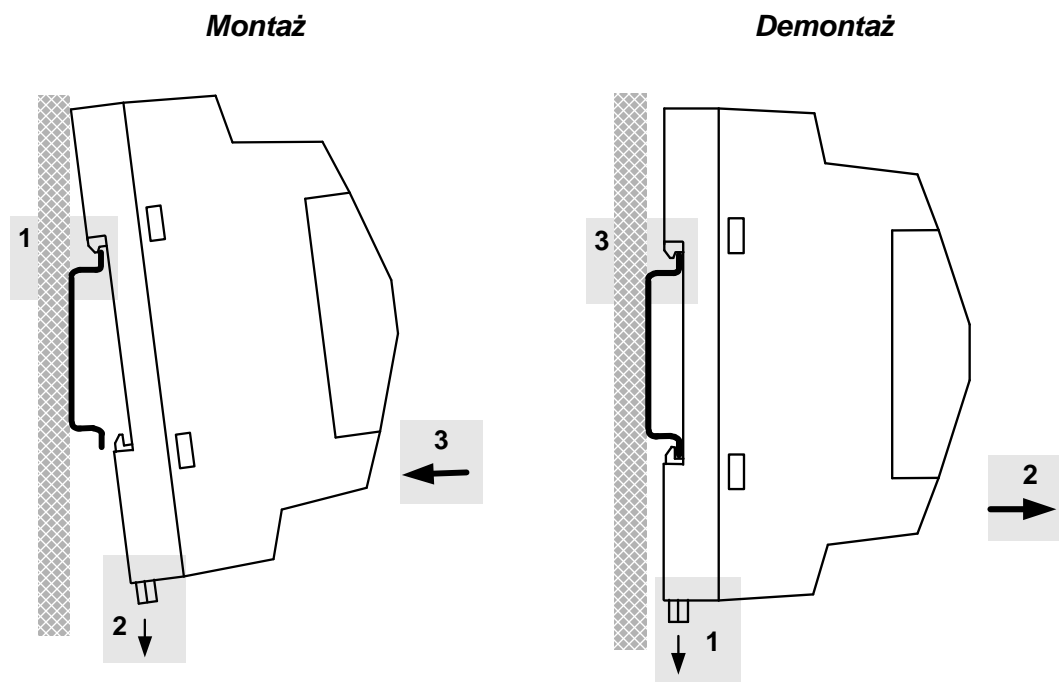
#### 3.1.1. Przygotowanie i zabezpieczenie miejsca instalacji



- Wyłącz urządzenie/installację, w którym montujesz przekaźnik programowalny.
- Uważaj, występuje niebezpieczeństwo porażenia prądem.
- Zabezpiecz urządzenie/installację przed przypadkowym załączeniem.
- Upewnij się, że żadne napięcie nie występuje w urządzeniu/installacji.
- Jeżeli nie jest możliwe całkowite odłączenie napięcia w strefie montażu, to należy zabezpieczyć miejsca występowania zagrożenia przed dotknięciem; zachowywać szczególną ostrożność!
- Sprawdź stan izolacji stosowanych przewodów.

### 3.2. Montaż mechaniczny

#### 3.2.1. Mocowanie na szynie montażowej (DIN 35mm)



Rys. 3.2.1. Mocowanie przekaźnika na szynie montażowej.

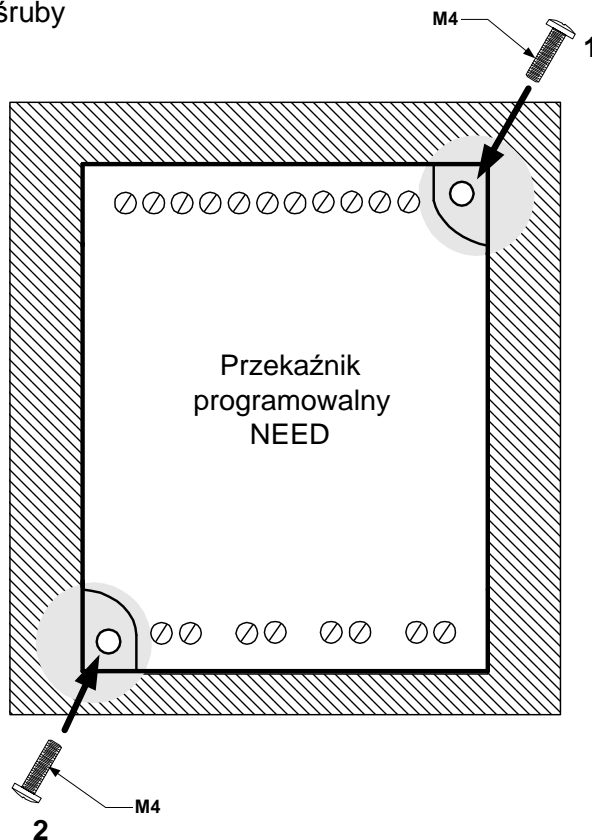
**Montaż** (rys. 3.2.1.)

1. Zaczep moduł na górnej części szyny montażowej.
2. Odciągnij dolny zaczep w dół.
3. Trzymając odciągnięty dolny zaczep dopchnij moduł od przodu i zwolnij odciągnięty zaczep.
4. Sprawdź pewność zamocowania modułu przekaźnika.

**Demontaż** (rys. 3.2.1.)

1. Odciągnij dolny zaczep w dół.
2. Trzymając odciągnięty dolny zaczep odchyl moduł od szyny.
3. Podnieś odchylony moduł i zdejmij z górnych zaczepów.

## 3.2.2. Mocowanie na śruby



Rys. 3.2.2. Mocowanie za pomocą śrub.

Montaż na 2 śruby (lub blachowkręty).

Średnica otworów przewidzianych pod montaż śrubowy: 5,5 mm.

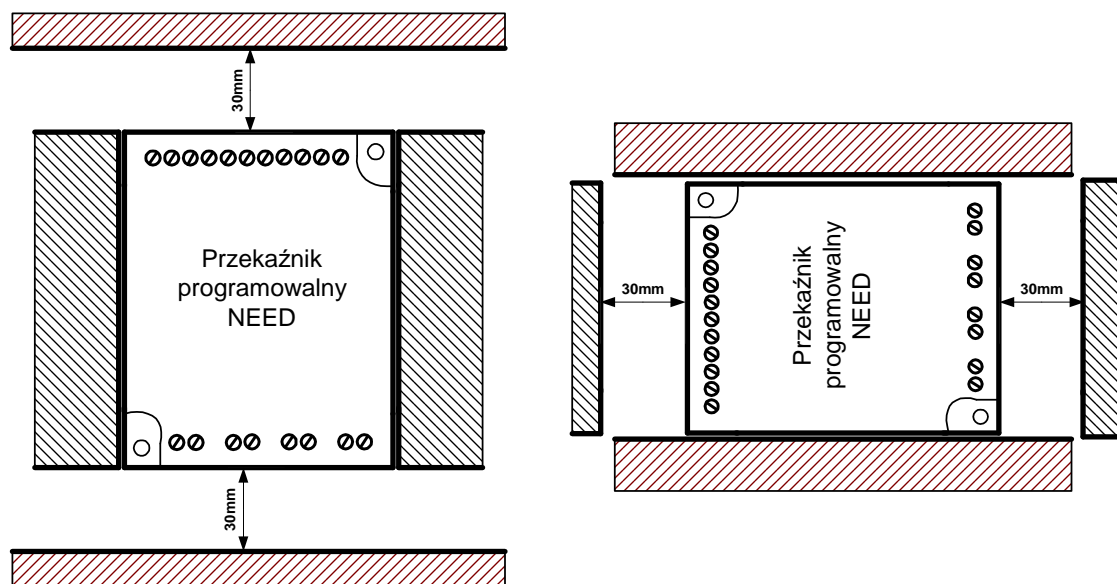
Uwaga: Do montażu na śruby nie są potrzebne żadne dodatkowe adaptory, wystarczy skorzystać z przewidzianych do tego celu otworów montażowych.

Odstępy montażowe:

Zaleca się zachować odstęp 3 cm pomiędzy krawędziami złącz wejściowych i wyjściowych przekaźnika programowalnego a innymi elementami instalacji (koryta montażowe, inne aparaty, ściana szafy montażowej itp.). Pozwoli to na wygodne okablowanie oraz umożliwi dobre chłodzenie modułu. Odstępy montażowe przedstawiono na rys. 3.2.3.

Ściany boczne mogą przylegać bezpośrednio do innych aparatów, elementów obudowy itp. Uwagi powyższe odnoszą się zarówno do montażu poziomego jak i pionowego, istotny jest odstęp od krawędzi ze złączami.

Należy pamiętać także o pozostawieniu odstępów min. 25mm od frontu urządzenia, przy montażu w zamykanej szafie.



Rys. 3.2.3. Odstępy montażowe – montaż pionowy i poziomy.

### 3.3. Zaciski przyłączeniowe, przewody

Zastosowane zaciski przyłączeniowe umożliwiają zastosowanie przewodów o przekrojach doprowadzeń:

$0,25\text{mm}^2$  do  $4\text{mm}^2$  - przewód pojedynczy

$0,25\text{mm}^2$  do  $2,5\text{mm}^2$  - przewód typu linka z końcówką tulejkową

Moment dokręcania śruby zacisku:  $0,5\text{ Nm}$  (max  $0,6\text{ Nm}$ ).

- Przewody doprowadzeniowe powinny być jak najkrótsze, ale nienaciągnięte.
- W przypadku zastosowania długich przewodów należy stosować ich ekranowanie lub skręcenie przewodów parami (przewodu fazowego L bądź sygnałowego z przewodem zerowym N lub 0V z sygnałem wejściowym +12(24)V dla wersji DC).
- Sugeruje się izolowanie obwodów prądu zmiennego i stałego oraz obwodów wytwarzających impulsy elektryczne poprzez odpowiednie prowadzenie przewodów. Można to zrealizować poprzez unikanie równoległego prowadzenia przewodów zasilających i sygnałowych, skręcanie parami, ekranowanie z jednostronnym uziemieniem ekranu.
- Przekrój przewodu dobrać uwzględniając prąd przepływający przez obciążenie.



**Uwaga:** We wszelkich nieujętych tutaj przypadkach należy stosować aktualne przepisy, standardy i regulacje urzędowe dotyczące instalacji elektrycznych.

### 3.4. Podłączenie wejść dyskretnych 230V AC



Wejścia należy łączyć do tej samej fazy, z której zasilany jest przekaźnik programowalny.

Odwrotne podłączenie zasilania tzn. zamienienie miejscami przewodów fazowego (L) i neutralnego (N) na wejściach zasilających przekaźnika programowalnego może spowodować wystąpienie niebezpiecznych napięć na jego zaciskach wejściowych, oraz brak wykrywania stanów logicznych.



Wejścia nie są izolowane galwanicznie od sieci elektrycznej zasilającej przekaźnik.



**Niebezpieczeństwo porażenia prądem – w przypadku zamiany przewodu zerowego N z fazowym L lub braku podłączenia przewodu N może występować na zaciskach napięcie o wysokości napięcia zasilającego.**

Do wejść można podłączać elementy stykowe: przyciski (zwierne, rozwierne), łączniki, przełączniki, styki przekaźników, styczników oraz fotokomórki i czujniki zbliżeniowe 2 lub 3 przewodowe 230V AC.

Zakres napięć sygnałów wejściowych zgodnie z normą PN-EN 61131:

Wejście wyłączone: 0 do 40 V AC ('0' logiczne)

Wejście załączone: 85 do 260 V AC ('1' logiczna)

Prąd wejściowy dla NEED-230AC-x1-08-4R:

I1 do I4 : 0,6 mA przy 230 V AC

I5, I6 : 8 mA przy 230 V AC – zwiększona odporność na zakłócenia, możliwość podłączenia długich przewodów – patrz specyfikacja techniczna.

I7, I8 : 0,9 mA przy 230 V AC

Prąd wejściowy dla wersji NEED-230AC-x1-16-8R:

I1 do I11 : 0,6 mA przy 230 V AC

I12, I13 : 8 mA przy 230 V AC – zwiększona odporność na zakłócenia, możliwość podłączenia długich przewodów – patrz specyfikacja techniczna.

I4, I15, I16 : 1,5 mA przy 230 V AC

Wejścia mają charakter rezystancyjny poza wejściami I5, I6 dla wersji NEED-230AC-01-08-4 lub I12, I13 dla NEED-230AC-x1-16-8 (charakter rezystancyjno-pojemnościowy), do których można podłączyć dłuższe przewody.

Nie należy stosować zbyt długich przewodów doprowadzających ze względu na ich pojemność i podatność na zakłócenia elektromagnetyczne – może to się objawić niekontrolowanymi stanami wejść logicznych np. błędne sygnalizowanie stanu załączenia wejścia.

Długości przewodów, jakie można podłączyć dla wejść zależą od układu wewnętrznego wejścia:

a) do wejść o normalnej odporności na zakłócenia można podłączyć przewód o długości do 10m – pomiar został wykonany dla najbardziej niekorzystnego przypadku prowadzenia przewodów fazowego i wejściowego równoległe obok siebie (podłączenie np. za pomocą dwużyłowego kabla sieciowego).

b) do wejść o zwiększonej odporności na zakłócenia można podłączyć przewód o długości 100m dlatego, że zawierają wewnętrzny kondensator 100nF zwiększający prąd wejściowy.

c) do wejść, które mogą pełnić rolę wejść analogowych podobnie jak do wejść zwykłych można podłączyć przewód o długości do 10m.

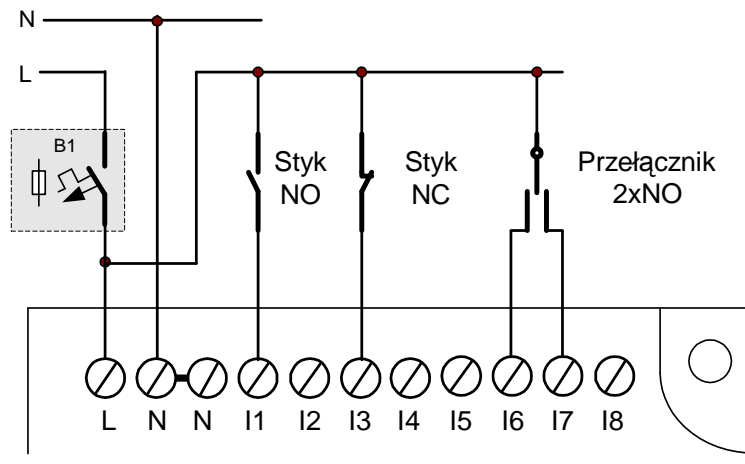




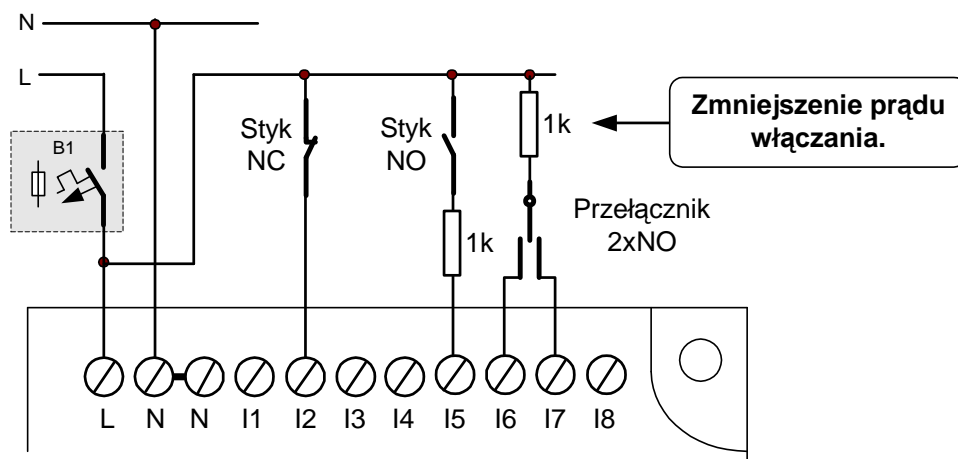
Wejścia I7 i I8 dla wersji NEED-230AC-x1-8-.. oraz I14, I15, I16 dla wersji NEED-230AC-x1-16-.. mogą być używane jako cyfrowe lub analogowe – zależy to od sposobu ich wykorzystania w programie.

Dla wejść o zwiększonej odporności na zakłócenia aby ograniczyć prąd załączania zaleca się wpięcie szeregowo z elementem stykowym rezystora o wartości ok. 1kOhm 1W (rys. 3.4.2. przedstawia podłączenie dla wersji NEED-230AC-x1-8-..).

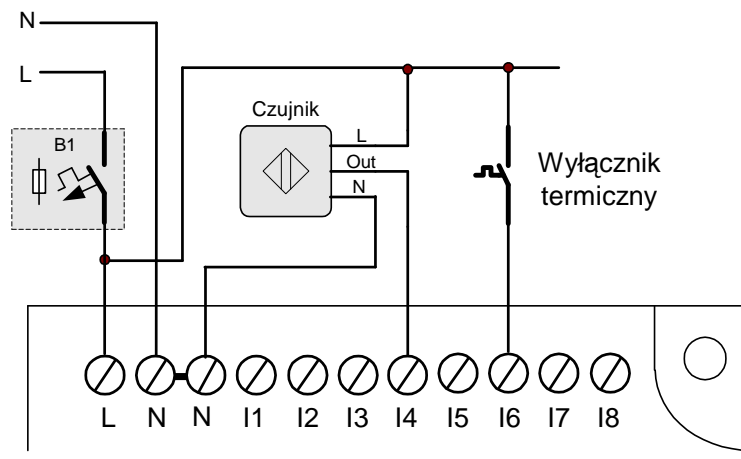
Wejścia I5, I6 dla wersji NEED-230AC-x1-8-.. oraz I12, I13 dla wersji NEED-230AC-x1-16-.. posiadające wewnętrzne kondensatory można zbocznikować zewnętrznymi rezystorami (100k Ohm) włączonymi pomiędzy wejście, a przewód N w celu szybszego rozładowania ich pojemności.



Rys. 3.4.1. Podłączenie wejść – elementy stykowe.



Rys. 3.4.2. Podłączenie wejść – elementy stykowe + rezystory zmniejszające udar prądowy wejścia.



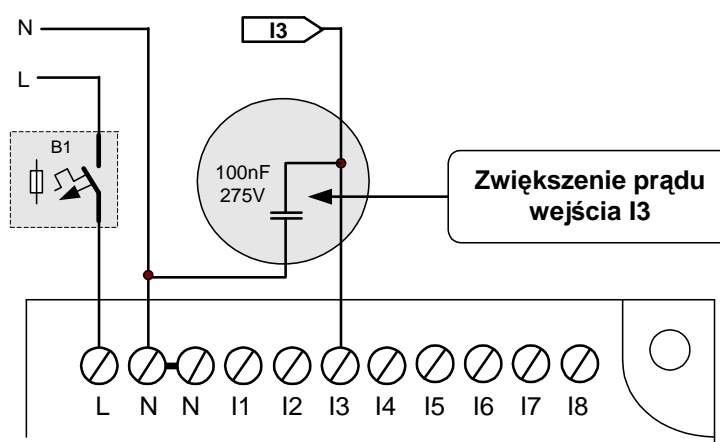
Rys. 3.4.3. Podłączenie wejść – czujnik zbliżeniowy, styk wyłącznika termicznego.



W celu zmniejszania zakłóceń na wejściach dyskretnych oraz zwiększenia długości przewodów, jakimi można podłączać do tych wejść elementy sterujące, należy zastosować elementy zewnętrzne zwiększające prąd w obwodzie wejścia oraz filtry wejściowe.

#### 1. Zwiększenie prądu w obwodzie wejścia.

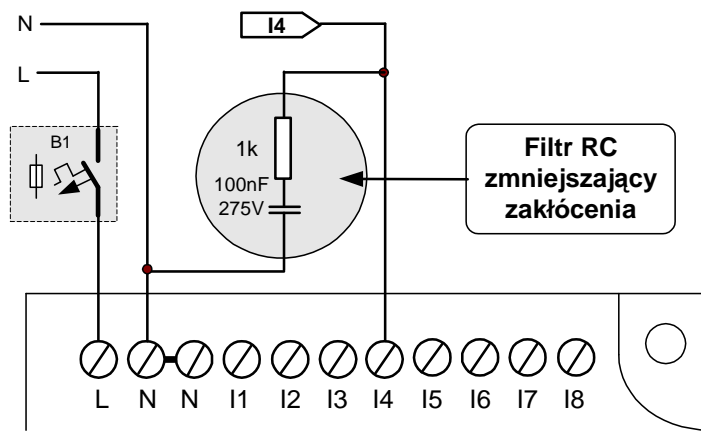
W celu zmniejszenia zakłóceń na wejściach I1..I4, I7, I8 dla NEED-230AC-x1-8-.. oraz I1..I11 i I14..I16 dla NEED-230AC-x1-16-.. można zastosować zewnętrzny kondensator, np. 100nF/275V klasy X1 lub X2 (zwiększenie prądu), podłączony pomiędzy zacisk wejściowy a zacisk N (rys. 3.4.4.).



Rys. 3.4.4. Zwiększenie prądu wejścia.

#### 2. Filtr RC

Dla zmniejszenia zakłóceń na wejściach I1..I4, I7, I8 dla NEED-230AC-x1-8-.. oraz I1..I11 i I14..I16 dla NEED-230AC-x1-16-.. można zastosować filtr RC (szeregowo połączony kondensator 100nF/275V klasy X1 lub X2 i rezystor 1k) podłączony pomiędzy zacisk wejściowy a zacisk N (rys. 3.4.5.).



Rys. 3.4.5. Wejściowy filtr RC.



**Uwaga:** Podanie pomiędzy zaciski wejściowe I1..In a N napięcia wyższego od maksymalnie dopuszczalnego może spowodować zniszczenie obwodów wejściowych przekaźnika programowalnego.

### 3.5. Podłączenie wejść dyskretnych 24V (12V) DC

Do zacisków wejściowych można podłączać elementy stykowe: przyciski (zwierne, rozwierne), łączniki, przełączniki, styki przekaźników, styczników oraz fotokomórki i czujniki zbliżeniowe 2 lub 3 przewodowe 24V (12V) DC.

Zakresy napięć sygnałów wejściowych są zgodne z normą PN-EN 61131.

Tabele 3.5. i 3.6. przedstawiają parametry wejść cyfrowych w zależności od wersji przekaźnika.

Tab. 3.5. Parametry wejść przekaźnika programowalnego **NEED-24DC-x1-8-...**

| Napięcie zasilania | Wejście | Zakres sygnałów wejściowych |              | Prąd wejściowy | Rezystancja wejściowa |
|--------------------|---------|-----------------------------|--------------|----------------|-----------------------|
|                    |         | Wej. wyłącz.                | Wej. załącz. |                |                       |
| V                  | nr      | V                           | V            | mA             | kΩ                    |
| 24 DC              | I1..I6  | -3..5                       | 15..30       | 3.3            | 7.44                  |
|                    | I7..I8  | -3..5                       | 15..30       | 2              | 12.36                 |
| 12 DC              | I1..I6  | -1..4                       | 8..26        | 3.3            | 3.65                  |
|                    | I7..I8  | -1..4                       | 8..26        | 1.1            | 10.92                 |

Tab. 3.6. Parametry wejść przekaźnika programowalnego **NEED-24DC-x1-16-....**

| Napięcie zasilania | Wejście  | Zakres sygnałów wejściowych |              | Prąd wejściowy | Rezystancja wejściowa |
|--------------------|----------|-----------------------------|--------------|----------------|-----------------------|
|                    |          | Wej. wyłącz.                | Wej. załącz. |                |                       |
| V                  | nr       | V                           | V            | mA             | kΩ                    |
| 24 DC              | I1..I13  | -3..5                       | 15..30       | 3.3            | 7.44                  |
|                    | I14..I16 | -3..5                       | 15..30       | 2              | 12.36                 |
| 12 DC              | I1..I13  | -1..4                       | 8..26        | 3.3            | 3.65                  |
|                    | I14..I16 | -1..4                       | 8..26        | 3.3            | 10.92                 |

Jeśli wejścia I14..I16 zostaną skonfigurowane jako prądowe to ich impedancja wynosi 49 Ω.



**Uwaga:** Podanie pomiędzy zaciski wejściowe I1..In a 0V napięcia wyższego od maksymalnie dopuszczalnego może spowodować zniszczenie obwodów wejściowych przekaźnika programowalnego.

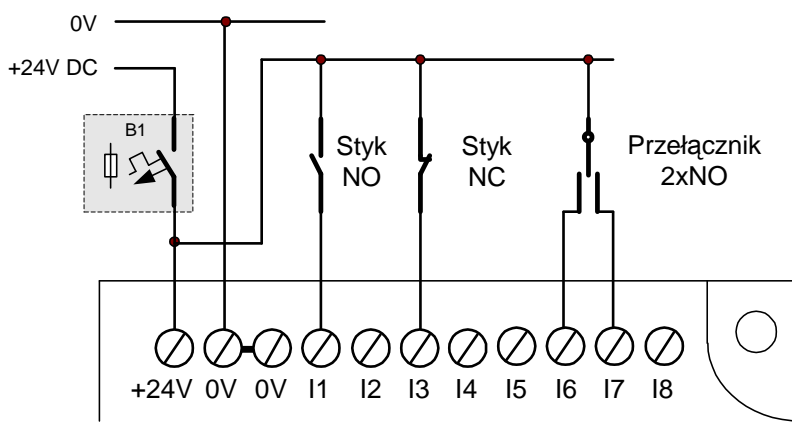


**Uwaga:** Podanie na wejścia prądowe prądu przewyższającego dopuszczalny może spowodować uszkodzenie obwodów wejściowych.

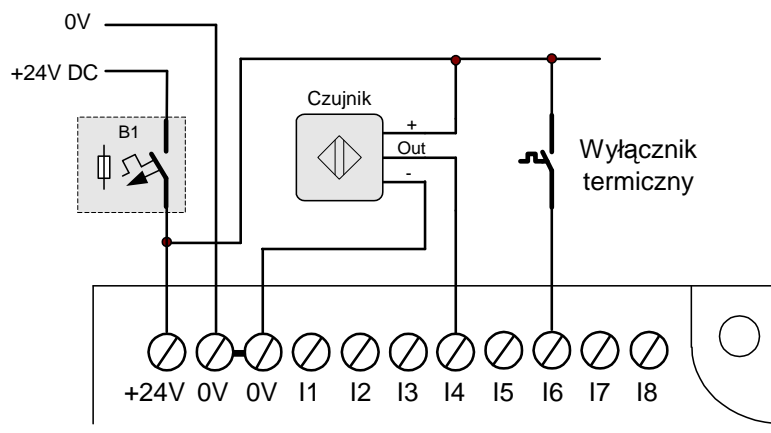


Wejścia mają charakter rezystancyjny.

Wejścia I7 i I8 dla wersji DC NEED..-x1-8-.. oraz I14, I15, I16 dla wersji NEED..-x1-16-.. mogą być wykorzystywane jako dyskretne lub analogowe – zależy to od sposobu ich wykorzystania w programie.



Rys. 3.5.1. Podłączenie wejść – elementy stykowe.



Rys. 3.5.2. Podłączenie wejść – czujnik zbliżeniowy, styk wyłącznika termicznego.

### 3.6. Podłączenie wejść analogowych AC



**Niebezpieczeństwo porażenia prądem – w przypadku zamiany przewodu zerowego N z fazowym L lub braku podłączenia przewodu N może występować na zaciskach napięcie o wysokości napięcia zasilającego.**



Wejścia analogowe nie są izolowane galwanicznie od sieci zasilającej przekaźnik.

Zakres napięć sygnałów wejściowych dla wejść analogowych wynosi 0 do 255 V AC; dokładność +/- 3% wartości zakresu pomiarowego.



Dla prawidłowego funkcjonowania pomiaru analogowego nie jest konieczna zgodność fazy i częstotliwości mierzonego przebiegu z napięciem zasilającym przełącznik.



Wejścia analogowe, które będą wykorzystywane także jako cyfrowe, należy łączyć do tej samej fazy, co napięcie zasilające przełącznik programowalny NEED.

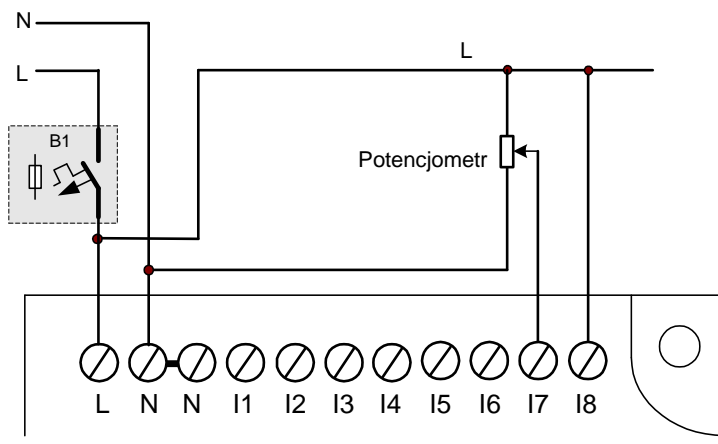
Wejścia analogowe mogą być używane jako wejścia dyskretne – wtedy stosuje się zasady podłączania jak dla wejść dyskretnych – patrz wyżej.

Pomiar analogowy dokonywany jest za pomocą układu uśredniającego. Wynik wskazywany jest w wartości skutecznej.

Wejścia analogowe, w związku z uśrednianiem, są mierzone z opóźnieniem. Napięcie wejściowe (mierzone) musi być przez chwilę stabilne, aby pomiar był dokładny.

### Wejścia analogowe dla wersji NEED-230AC-x1-8-4

W przełącznikach tego typu są to ostanie dwa wejścia I7 i I8.

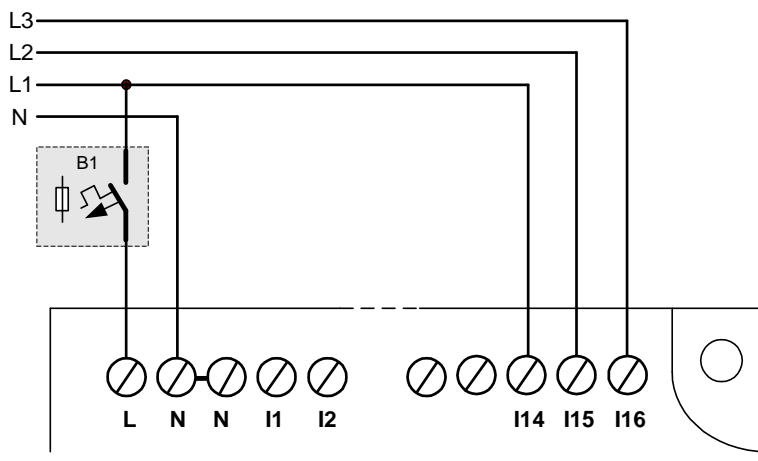


**Podłączenie sieci trójfazowej niedopuszczalne**

Rys. 3.6.1. Wejścia analogowe – potencjometr, kontrola napięcia sieci - NEED-230AC-01-8-..

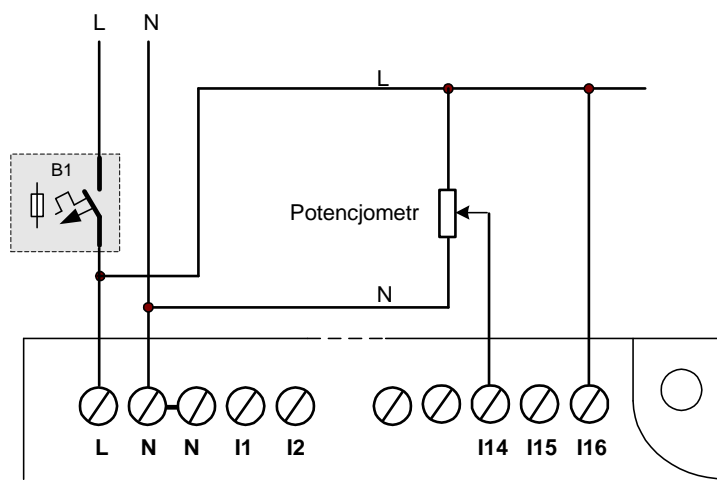
### Wejścia analogowe dla wersji NEED-230AC-x1-16-8

W przełącznikach tego typu są to ostanie trzy wejścia I14, I15, I16.



**Podłączenie sieci trójfazowej dopuszczalne**

Rys. 3.6.2. Wejścia analogowe - NEED-230AC-01-16-..



Rys. 3.6.3. Wejścia analogowe – podłączenie potencjometru.

W wersji NEED-230AC-.. niedostępne jest ładowanie czasów dla *Timerów* i progów dla *Liczników*.



**Uwaga:** Pamiętaj o odpowiednim doborze podłączanych elementów pod względem mocy oraz znamionowego napięcia pracy.



**Uwaga:** Pamiętaj, że wejście analogowe pobiera prąd. Może to znacznie zafałszować wynik pomiaru, jeżeli źródło napięcia mierzonego ma zbyt dużą impedancję wewnętrzną.



**Uwaga:** Elementy takie jak potencjometry, przełączniki itp. powinny być starannie zaizolowane ze względu na ryzyko porażenia prądem.

### 3.7. Podłączenie wejść analogowych 24V (12V) DC

Poziom sygnałów wejściowych dla wejść analogowych skonfigurowanych jako napięciowe wynosi 0–25,5V DC (krok 0,1V) lub 0 – 12,75V (krok 0,05V).

Poziom sygnałów wejściowych dla wejść analogowych skonfigurowanych jako prądowe wynosi 0–51mA (krok 0,2mA) lub 0 – 25,5mA (krok 0,1mA).

Dokładność +/- 2% wartości zakresu pomiarowego.

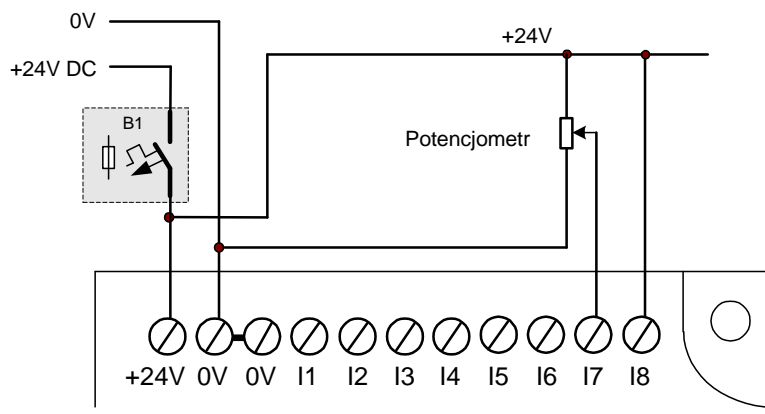
Rozdzielczość wejść analogowych: 8 bit.



Wejścia analogowe mogą być używane jako wejścia cyfrowe – wtedy stosuje się zasady podłączania jak dla wejść dyskretnych – patrz pkt. 3.4.

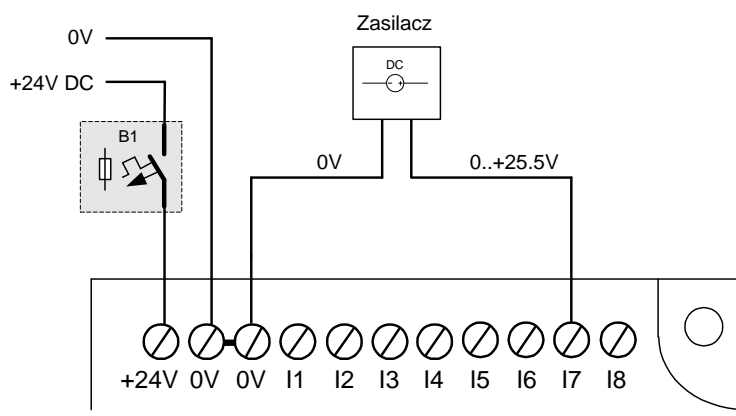
Wejścia analogowe to I7 i I8 dla wersji DC NEED..-x1-8-.. oraz I14, I15, I16 dla wersji NEED..-x1-16-...

Poniżej przedstawiono układ do zadawania napięcia na wejściu I7 oraz kontroli napięcia zasilania za pomocą wejścia I8 połączonego z „+” zasilania przełącznika dla wersji NEED-24DC-x1-8-... Przy takim połączeniu potencjometrem możemy regulować nie tylko progi komparatorów, ale także zadawać wartości czasu dla *Timerów* oraz ustawiać progi *Liczników*.



Rys. 3.7.1. Wejścia analogowe – potencjometr.

Wejścia analogowe w przekaźniku NEED pozwalają na odczyt napięcia zewnętrznego w zakresie 0V do 25.5V (lub 0V..12.5V dla wersji NEED-24DC-x1-16-..). Układ połączeń dla zewnętrznego źródła napięcia dla wersji NEED-24DC-x1-8-.. przedstawiono na rys. 3.7.2.



Rys. 3.7.2. Wejścia analogowe – zakres.



**Uwaga:** Pamiętaj o odpowiednim doborze podłączanych elementów pod względem mocy oraz znamionowego napięcia pracy.

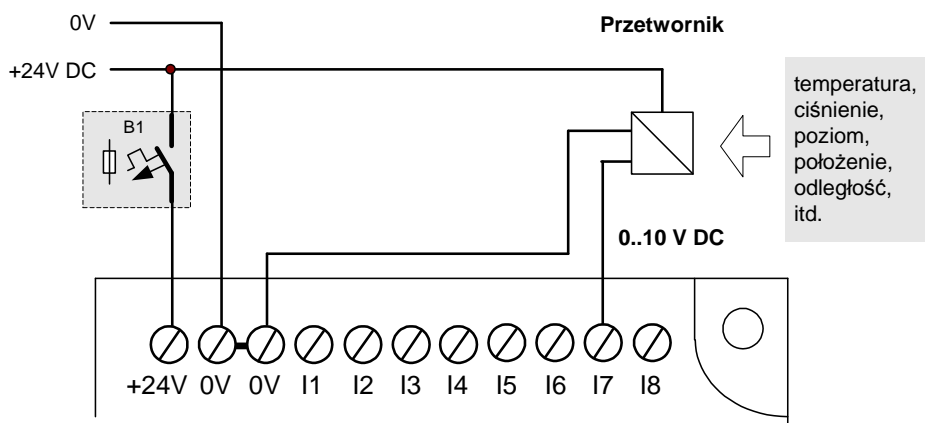


**Uwaga:** Pamiętaj, że wejście analogowe pobiera prąd. Może to znacznie zafałszować wynik pomiaru, jeżeli źródło napięcia mierzonego ma zbyt dużą impedancję wewnętrzną.

### Przetwornik 0..10 V DC.

Do wejść analogowych można podłączać różnego typu przetworniki wielkości elektrycznych (napięcie, prąd, częstotliwość) lub nieelektrycznych (temperatura, ciśnienie, odległość, siła itp.), które posiadają standardowe wyjście analogowe napięciowe lub prądowe.

Dla przetwornika napięciowego wytwarzającego napięcie od 0 do 10V dla min. i max. przetwarzanej wielkości otrzymamy 100 punktową charakterystykę przetwarzania.



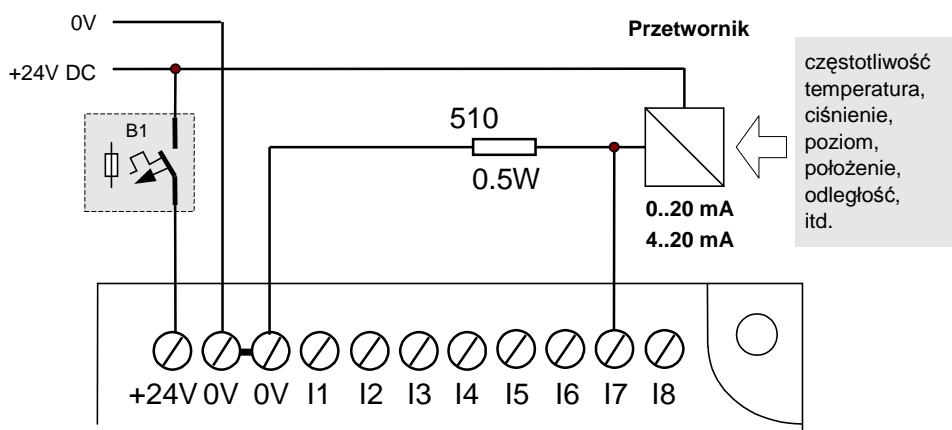
Rys. 3.7.3. Wejścia analogowe – przetwornik 0..10V dla wersji NEED-24DC-01-8-4

### Przetwornik 0..20 mA.

Wersje NEED-24DC-x1-8-4 i NEED-12DC-x1-8-4 nie posiadają wbudowanego przetwornika prąd / napięcie.

W celu wykorzystania przetwornika z wyjściem prądowym w zakresie 0..20mA lub 4..20mA należy zastosować prosty układ przetwarzający prąd na napięcie. Uzyskamy to mierząc spadek napięcia na rezystorze 510  $\Omega$ , będącym obciążeniem przetwornika. Spadek napięcia jest proporcjonalny do wielkości przepływającego prądu w zależności: 1mA = ~ 0.5V. W obliczeniach uwzględniono rezystancję wewnętrzną wejścia analogowego przekaźnika. Charakterystyczne punkty przetwarzania dla wersji 24V DC to:

- 1mA → ~0,5V
- 4mA → ~1,9V
- 10mA → ~4,9V
- 20mA → ~9,8V

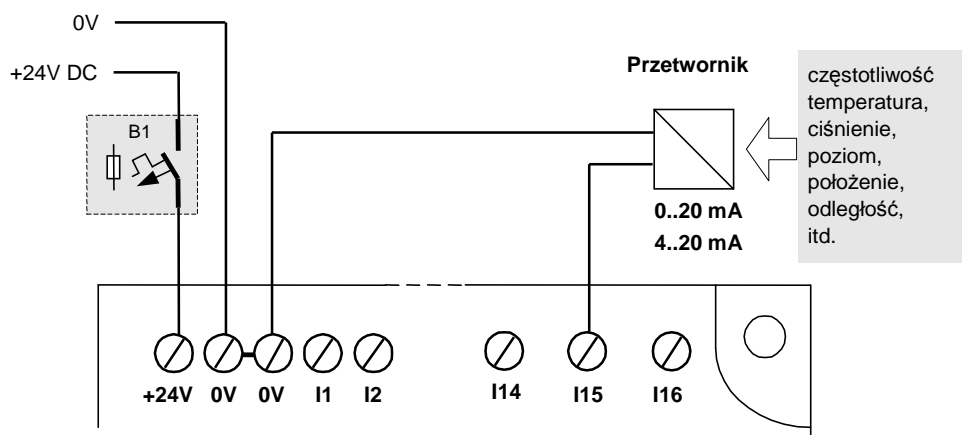


Rys. 3.7.4. Wejścia analogowe – przetwornik 0..20mA dla wersji NEED-24DC-01-8-4

Wersje NEED-24DC-x1-16-8 i NEED-12DC-x1-16-8 posiadają wbudowany przetwornik prąd / napięcie.

Przetworniki z wyjściem prądowym można podłączać bezpośrednio do wejść przekaźnika AI14, AI15, AI16 po ich uprzednim skonfigurowaniu w programie PC Need jako wejścia prądowe (I) i wgraniu ustawień do przekaźnika.





Rys. 3.7.5. Wejścia analogowe – przetwornik 20mA dla wersji NEED-24DC-x1-16-8



**Uwaga:** Podanie pomiędzy zaciski wejść analogowych a 0V napięcia wyższego od maksymalnie dopuszczalnego może spowodować zniszczenie obwodów wejściowych przekaźnika programowalnego.



**Uwaga:** Nie należy podawać napięcia na wejścia AI14, AI15, AI16 ustawione jako prądowe.



**Uwaga:** Podłączenie źródła napięcia na wejścia AI14, AI15, AI16 ustawione jako prądowe może doprowadzić do ich uszkodzenia. Maksymalne napięcie na wejściu prądowym dla  $I=51\text{mA}$  wynosi 2.5V.



**Uwaga:** Podanie napięcia ze źródła o wyższej niż 51mA wydajności na wejścia AI14, AI15, AI16 ustawione jako prądowe, spowoduje zadziałanie zabezpieczenia wewnętrznego. Przekaznik będzie cyklicznie próbował wejścia.

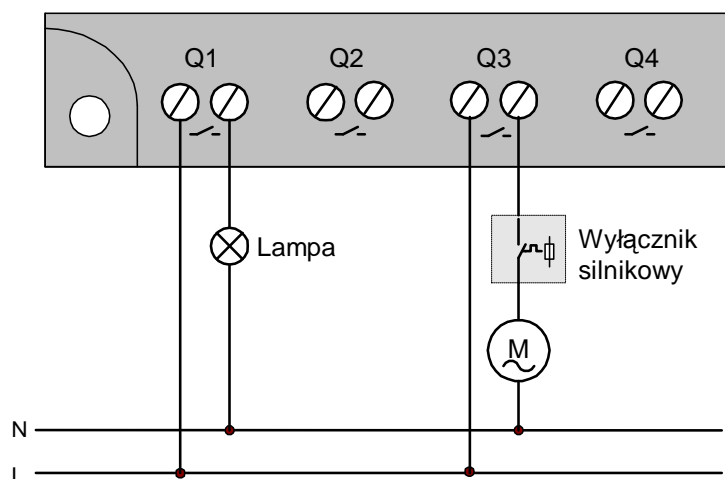
### 3.8. Podłączenie wyjść

Zaciski wyjściowe są połączone ze stykami wewnętrznych przekaźników elektromagnetycznych Q1..Qn. Wersja NEED..-x1-8-4R to 4 wyjścia przekaźnikowe. Wersja rozszerzona NEED..-x1-16-8R zawiera 8 wyjść przekaźnikowych.

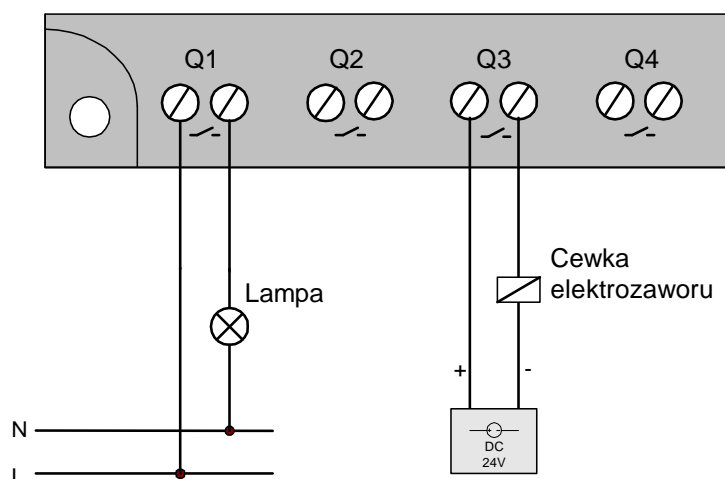
Wyjścia są bezpotencjałowe i izolowane galwanicznie od reszty układu i względem siebie – możemy zbudować niezależne układy sterowania.

Obciążalność 1 wyjścia – patrz specyfikacja techniczna – 230V, 10A przy rezystancyjnym charakterze obciążenia.

Należy pamiętać o odpowiednim zabezpieczeniu obwodów wyjściowych (bezpiecznik) sterowanych przez przekaźnik w zależności od mocy i charakteru obciążenia, aby nie przekroczyć wartości podanych w specyfikacji technicznej.



Rys. 3.8.1. Wyjścia przekaźnikowe – zasilanie sieciowe 230V AC -wersja NEED..-x1-8-4R.



Rys. 3.8.2. Wyjścia przekaźnikowe – różne obwody zewnętrzne -wersja NEED..-x1-8-4R.

### 3.9. Podłączenie zasilania AC



**Napięcie zasilające przełącznik jest niebezpieczne dla życia!**

Od jakości wykonanych połączeń zależy bezpieczeństwo użytkownika!

Należy przestrzegać prawidłowego podłączenia przewodów doprowadzających napięcie zasilające – przewód fazowy L i przewód zerowy N.



Zamienienie miejscami przewodów na wejściach zasilających tzn. podłączenie fazy L na zacisk N oraz przewodu neutralnego N na zacisk wejściowy L przełącznika programowalnego może spowodować wystąpienie niebezpiecznych napięć na zaciskach wejściowych I1..In oraz portach komunikacyjnych, a także brak wykrywania stanów logicznych.



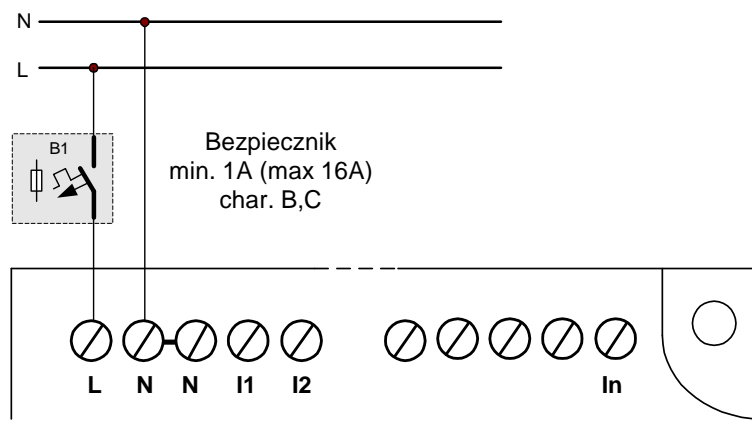
Znamionowe napięcie zasilania **115/230V AC 50/60 Hz**



**Podanie napięcia międzyfazowego 400V AC pomiędzy zaciski L i N spowoduje zniszczenie przełącznika programowalnego**

Należy zabezpieczyć przełącznik programowalny bezpiecznikiem o prądzie nominalnym 1A. Oczywiście zabezpieczenie nie może mieć zbyt dużej wartości bo nie spełni swojej roli – zalecane maksimum to 6A.

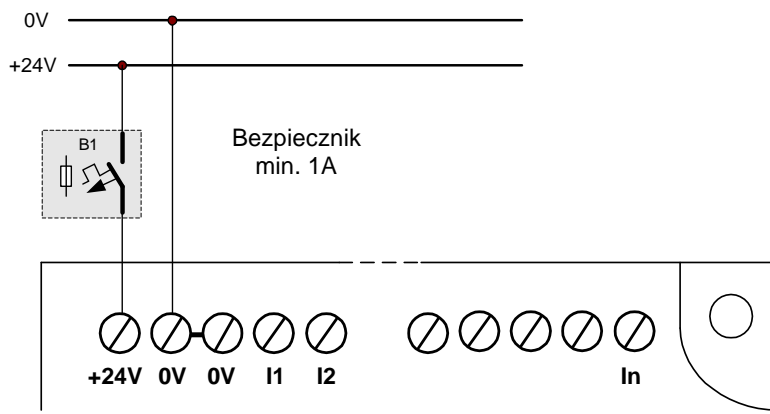
Jeśli zabezpieczenie będzie wspólne także dla wejść i wyjść, to należy uwzględnić prąd zabezpieczenia zasilania przełącznika programowalnego – min. 1A.



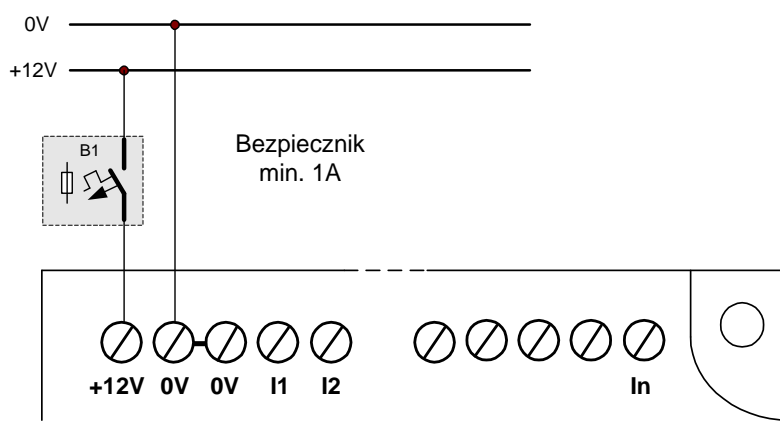
Rys. 3.9.1. Zasilanie przełącznika programowalnego NEED-230AC-x1-..”.

### 3.10. Podłączenie zasilania 24V (12V) DC

Bezpiecznik zabezpieczający przewody powinien mieć wartość powyżej 1A, gdyż w momencie włączania występuje uderzenie prądowe w wyniku ładowania wewnętrznego kondensatora w zasilaczu przekaźnika.



Rys. 3.10.1. Zasilanie przekaźnika programowalnego NEED-24DC-x1-...



Rys. 3.10.2. Zasilanie przekaźnika programowalnego NEED-12DC-x1-...



**Uwaga:** Podanie pomiędzy zaciski zasilające +24V (+12V) a 0V napięcia wyższego od maksymalnie dopuszczalnego może spowodować zniszczenie przekaźnika programowalnego.

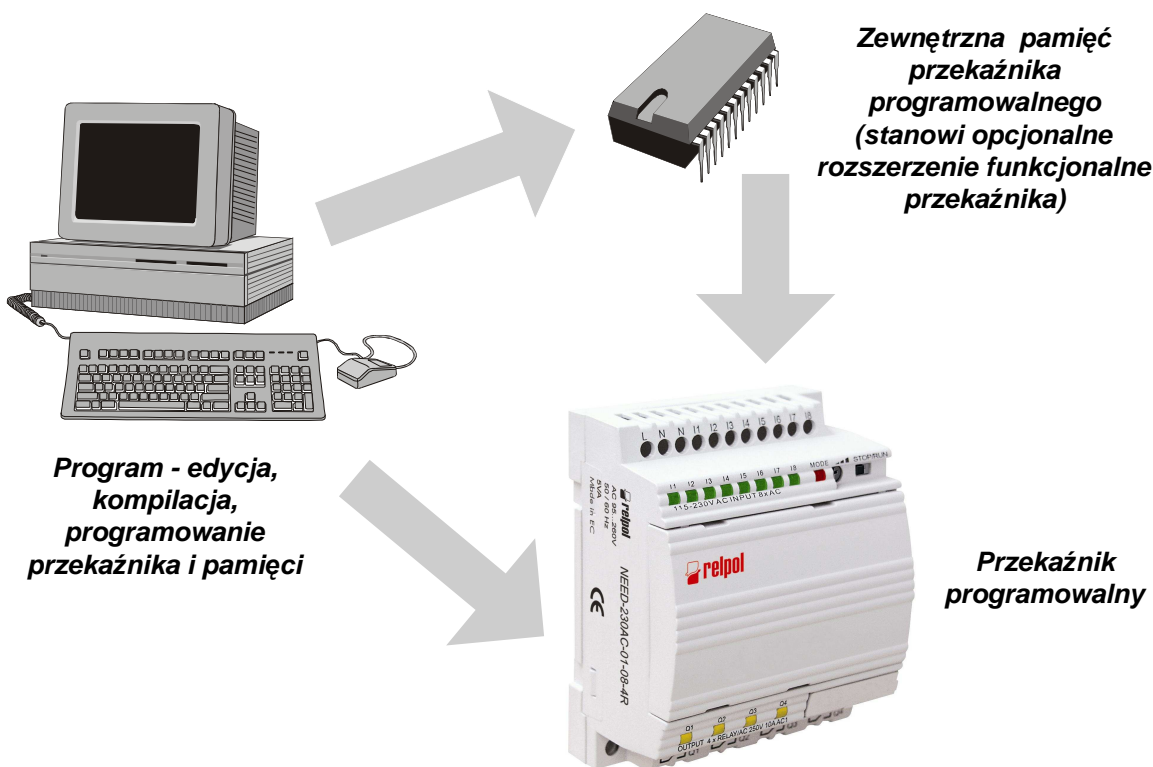
## 4. ZASOBY PRZEKAŹNIKA

Sterowniki programowalne są urządzeniami, które w swojej strukturze obejmują dwa podstawowe elementy: jednostkę centralną (CPU) wraz z pamięcią oraz peryferia wejścia/wyjścia. Oczywiście do pełnej funkcjonalności potrzebne jest również jakieś urządzenie programujące oraz kabel do komunikacji ze sterownikiem. Wszystkie te cechy posiada również przekaźnik programowalny NEED.

### 4.1. System przekaźnika programowalnego NEED

W skład systemu przekaźnika programowalnego wchodzi:

1. Program do edycji, kompilacji oraz ładowania programu do pamięci przekaźnika.
2. Zewnętrzna pamięć przekaźnika (nie jest ona wymagana - umożliwia łatwe przenoszenie programu pomiędzy PC i przekaźnikiem).
3. Przekaźnik programowalny.

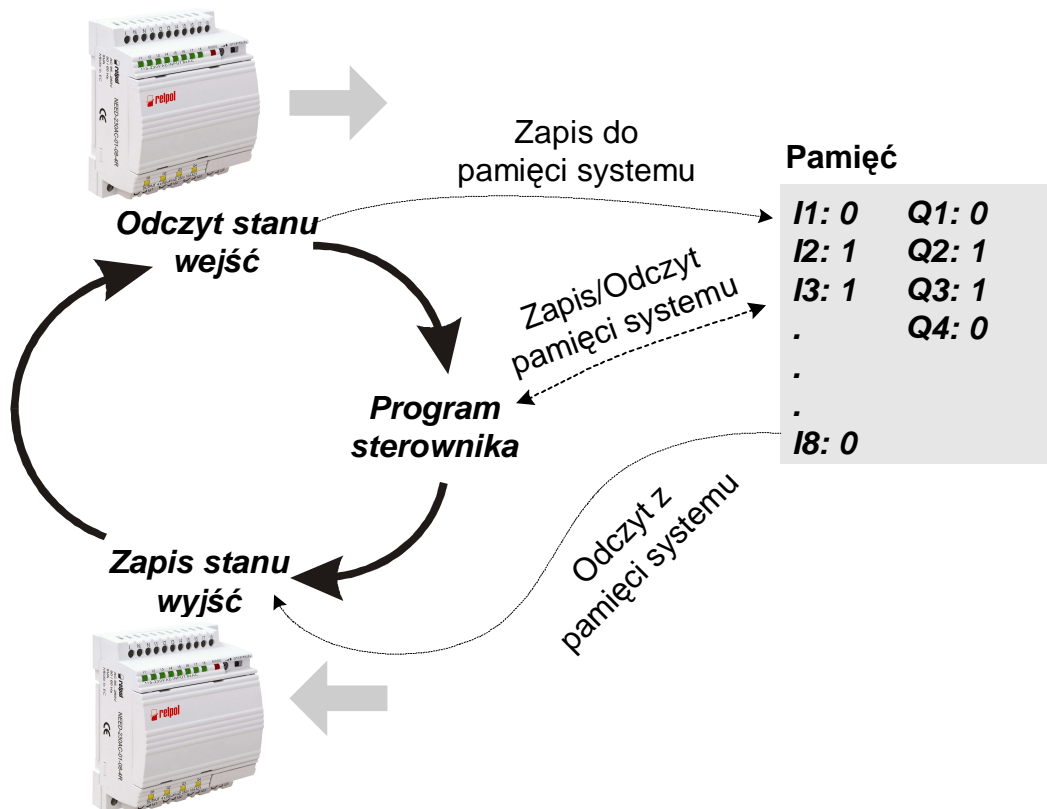


Rys. 4.1.1. System przekaźnika programowalnego NEED.

### 4.2. Cykl programu

Aby można było realizować różne aplikacje wykorzystując przekaźnik programowalny należy przede wszystkim stworzyć odpowiedni program i umieścić go w pamięci sterownika. Po uruchomieniu sterownik zaczyna przetwarzanie programu – od pierwszej instrukcji do ostatniej, po czym cykl jest powtarzany.

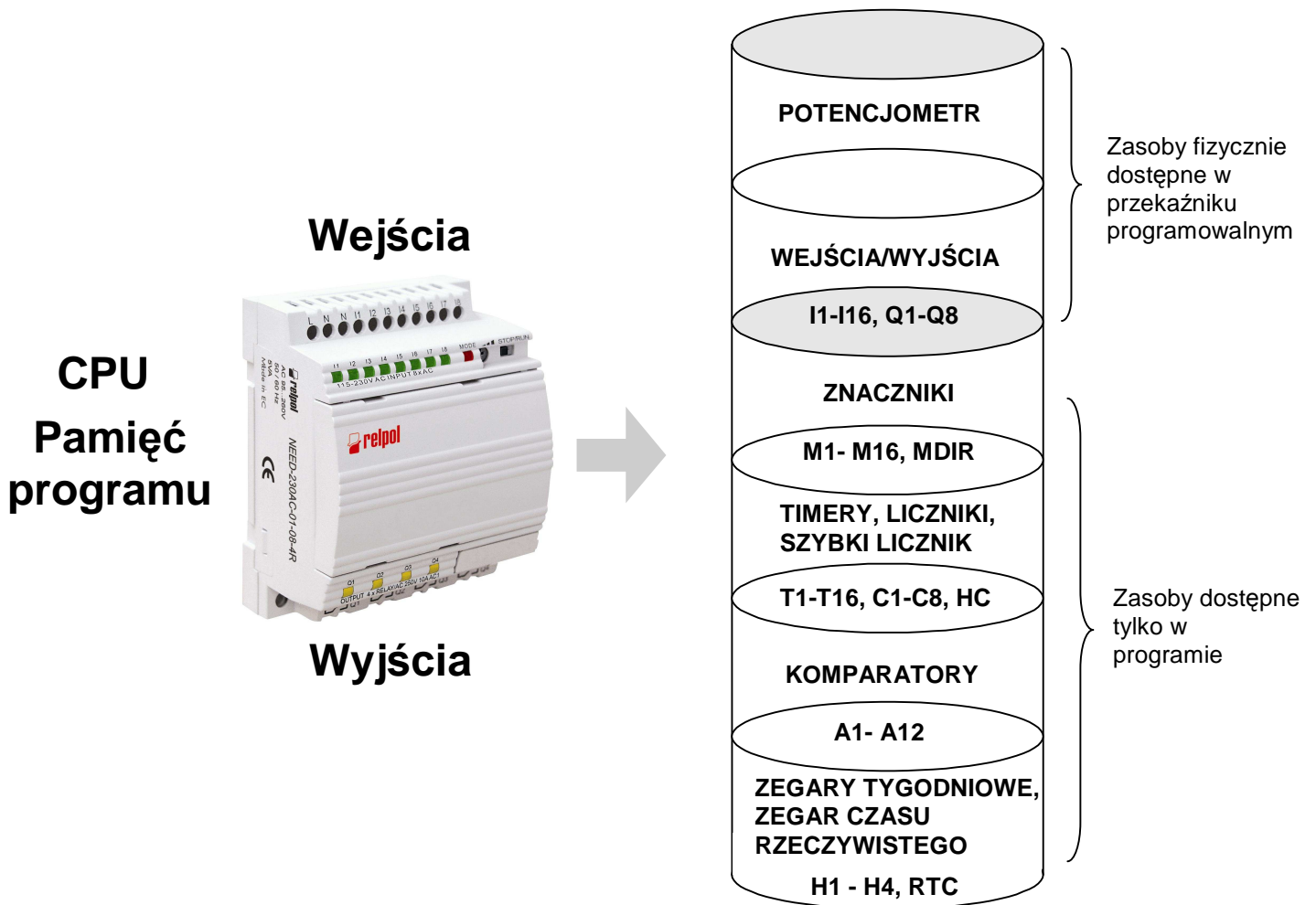
Na początku każdego cyklu stany wejść są zapisywane do specjalnych obszarów pamięci. W trakcie programu następuje odwołanie się nie do bezpośrednich stanów wejść/wyjść, ale do ich kopii umieszczonych w pamięci odwzorowującej proces. Podobnie jest z sygnałami wyjściowymi. Sterownik zapamiętuje te stany w pamięci odwzorowania procesu i dopiero z końcem każdego cyklu następuje przepisanie tychże stanów do wyjść przekaźnika. Cykl pracy sterownika przedstawiono na rys. 4.2.1.



Rys. 4.2.1. Cykl pracy sterownika.

Do prawidłowego zrozumienia i dobrego programowania przełącznika NEED niezbędna jest znajomość jego wewnętrznych zasobów.

## 4.3. Zasoby przekaźnika programowalnego NEED



Rys. 4.3.1. Zasoby przekaźnika programowalnego NEED.

Do komunikacji ze światem zewnętrznym przekaźnik NEED używa wejść i wyjść. Praktycznie są to jedyne zasoby, które użytkownik postrzega z zewnątrz i w oparciu, o które mogą być budowane nawet skomplikowane aplikacje. Jednak o prawdziwej sile każdego sterownika stanowią także jego zasoby wewnętrzne – „niewidzialne” z zewnątrz, do których dostęp ma jedynie programista. Na rys. 4.3.1. symbolicznie przedstawiono zasoby przekaźnika programowalnego, natomiast w tabeli 4.3. wyszczególniono ilość poszczególnych elementów zawartych w strukturze systemowej przekaźnika.

Odpowiednie używanie i wykorzystywanie zasobów przekaźnika programowalnego NEED zależy od użytkownika. Poniżej przedstawiamy opis poszczególnych elementów oraz sposoby zapisu dla różnych języków programowania.

Tab. 4.3. Zasoby przekaźnika programowalnego NEED.

| Nazwa  | Ilość<br>NEED..-x1- 08- 4   | Ilość<br>NEED..-x1-16- 8   |
|--|---|--|
| Wejścia cyfrowe „I”  | I1 – I8<br>w tym wejścia I7,I8 mogą być<br>także wejściami<br>analogowymi | I1 – I16<br>w tym wejścia I14, I15, I16<br>mogą być także wejściami<br>analogowymi |
| Wyjścia cyfrowe „Q”<br>przekaźnikowe typu<br>NO                        | Q1 – Q4   | Q1 – Q8  |
| Komparatory<br>„A”   | A1 – A8   | A1 – A12   |
| Znaczniki<br>„M”   | M1 – M16  | M1 – M16   |
| Znacznik MDIR<br>określający kierunek<br>podłączenia faz L1, L2,<br>L3 |   | MDIR   |
| Timery<br>„T”  | T1 – T8   | T1 – T16   |
| Liczniki<br>„C”  | C1 – C8   | C1 – C8  |
| Szybki Licznik<br>HC do 20kHz  |   | HC1  |
| Zegar Czasu<br>Rzeczywistego   | 1   | 1<br>Automatyczna zmiana czasu<br>lato/zima w różnych strefach<br>czasowych        |
| Zegary Tygodniowe<br>„H”   | H1 – H4   | H1 – H4  |

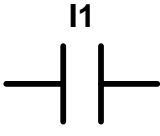


#### 4.4. Wejścia cyfrowe

Każde z wejść może być skonfigurowane jako normalnie otwarte i normalnie zamknięte. Zasoby te reprezentują fizyczne wejścia sterownika programowalnego.

##### 4.4.1. Wejścia cyfrowe normalnie otwarte

Symbole wejść cyfrowych normalnie otwartych.

| STL   | LAD  |
|---|--|
| <b>A I1</b><br>lub<br><b>O I1</b><br>lub<br><b>X I1</b> |  |

SYMBOL: **In**, gdzie n oznacza numer wejścia n=1..8 dla NEED..-x1-08-..  
n=1..16 dla NEED..-x1-16-..

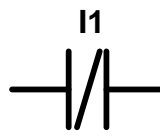
STANY LOGICZNE:

'1' – Jest napięcie zasilające na wejściu.

'0' – Brak napięcia zasilającego na wejściu.

##### 4.4.2. Wejścia cyfrowe normalnie zamknięte

Symbole wejść cyfrowych normalnie zamkniętych.

| STL  | LAD  |
|--|--|
| <b>AN I1</b><br>lub<br><b>ON I1</b><br>lub<br><b>XN I1</b> |  |

SYMBOL: **In**, gdzie n oznacza numer wejścia n=1...8 dla NEED..-01-08-..  
n=1...16 dla NEED..-01-16-..

STANY LOGICZNE:

'0' – Jest napięcie zasilające na wejściu.

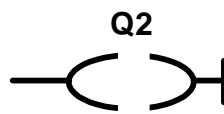
'1' – Brak napięcia zasilającego na wejściu.

#### 4.5. Wyjścia cyfrowe

Wyjścia cyfrowe mogą być różnego typu. Należy jednak zawsze pamiętać, iż mamy do dyspozycji maksymalnie 8 fizycznych wyjść!

##### 4.5.1. Wyjścia cyfrowe normalne

Symbole wyjść cyfrowych normalnych.

| STL         | LAD  |
|-------------|--|
| <b>= Q2</b> |  |

SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED..-x1-08-..  
n=1..8 dla NEED..-x1-16-..

STANY LOGICZNE:

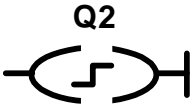
'1' – styki zwarte.

'0' – styki rozwarte.

Wyjście tego typu działa jak zwykły przekaźnik – tzn. podanie napięcia na cewkę powoduje zadziałanie przekaźnika.

#### 4.5.2. Wyjścia cyfrowe impulsowe

Symbole wyjść cyfrowych impulsowych.

| STL   | LAD  |
|-------|--|
| FP Q2 |  |

SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED..-x1-08-..  
n=1..8 dla NEED..-x1-16-..

STANY LOGICZNE:


'1' – Jeżeli poprzedni stan wynosił 0 i wystąpiło dodatnie sterujące zbocze logiczne.

'0' – Jeżeli poprzedni stan wynosił 1 i wystąpiło dodatnie sterujące zbocze logiczne.

Wyjście to działa jak przerytunek, który wyzwalany zboczem narastającym, zmienia stan swojego wyjścia na przeciwny.

#### 4.5.3. Wyjścia cyfrowe kasujące

Symbole wyjść cyfrowych kasujących.

| STL  | LAD  |
|------|--|
| R Q2 |  |

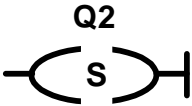
SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED..-x1-08-..  
n=1..8 dla NEED..-x1-16-..

STANY LOGICZNE:

'0' – Jeżeli wystąpił stan sterujący '1'.

#### 4.5.4. Wyjścia cyfrowe ustawiające

Symbole wyjść cyfrowych ustawiających.

| STL  | LAD  |
|------|--|
| S Q2 |  |

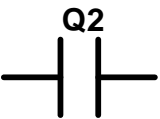
SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED...-x1-08-..  
n=1..8 dla NEED..-x1-16-..

STANY LOGICZNE:

'1' – Jeżeli wystąpił stan sterujący '1'.

#### 4.5.5. Wyjścia cyfrowe normalne wykorzystane do dalszego sterowania

Symbole wyjść cyfrowych normalnych wykorzystywanych do dalszego sterowania.

| STL                                | LAD   |
|------------------------------------|---|
| A Q2<br>lub<br>O Q2<br>lub<br>X Q2 |  |

SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED..-x1-08-..  
n=1...8 dla NEED..-x1-16-..


STANY LOGICZNE:

'1' – Jeżeli stan wyjścia fizycznego wynosi '1'.

'0' – Jeżeli stan wyjścia fizycznego wynosi '0'.

#### 4.5.6. Wyjścia cyfrowe zanegowane wykorzystane do dalszego sterowania

Symbol wyjść cyfrowych wykorzystywanych do dalszego sterowania.

| STL                                   | LAD  |
|---------------------------------------|--|
| AN Q2<br>lub<br>ON Q2<br>lub<br>XN Q2 |  |

SYMBOL: **Qn**, gdzie n oznacza numer wyjścia n=1..4 dla NEED..-x1-08-..  
n=1..8 dla NEED..-x1-16-..

STANY LOGICZNE:

'1' – Jeżeli stan wyjścia fizycznego wynosi '0'.

'0' – Jeżeli stan wyjścia fizycznego wynosi '1'.

## 4.6. Znaczniki

### 4.6.1. Znaczniki M1..M16

*Znacznik* (ang. *Marker*) jest elementem logicznym i traktowany jest jako zmienna wykorzystywana w programie. Posiada swój wewnętrzny stan '0' lub '1'.

Ze *Znacznikiem* nie możemy wiązać fizycznie określonego wejścia lub wyjścia, ale możemy go wykorzystać do łączenia logicznych struktur programu. Czyli *Znaczniki* możemy traktować jak 16 zarezerwowanych bitów, do których możemy się odnieść jak do wejść lub wyjść – tzn. podlegają one takim samym „operacjom” (instrukcjom) jak wejścia i wyjścia, ale nie mają fizycznej reprezentacji w postaci styków.

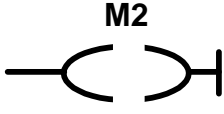
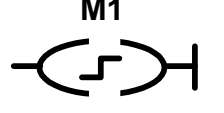
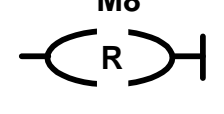
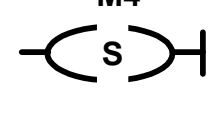
W składni instrukcji czy w symbolu graficznym zamiast litery I lub Q pojawia się symbol *Znacznika M*.

SYMBOL: **Mn**, gdzie n oznacza liczbę od 1 do 16

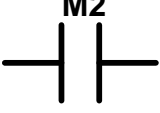

STANY LOGICZNE:

'0' lub '1' w zależności od realizowanej funkcji w programie.

Symbol *Znaczników*.

| STL   | LAD   |
|-------|---|
| = M2  |  <p>The diagram shows a coil symbol for setting M2. It consists of two parallel vertical lines connected by a horizontal line on the left, with a vertical line on the right. The label 'M2' is positioned above the coil.</p>  |
| FP M1 |  <p>The diagram shows a flip-flop symbol for M1. It consists of two parallel vertical lines connected by a horizontal line on the left, with a vertical line on the right. Inside the coil, there is a square wave symbol. The label 'M1' is positioned above the coil.</p> |
| R M8  |  <p>The diagram shows a coil symbol for resetting M8. It consists of two parallel vertical lines connected by a horizontal line on the left, with a vertical line on the right. The label 'R' is inside the coil, and 'M8' is positioned above it.</p>                      |
| S M4  |  <p>The diagram shows a coil symbol for setting M4. It consists of two parallel vertical lines connected by a horizontal line on the left, with a vertical line on the right. The label 'S' is inside the coil, and 'M4' is positioned above it.</p>                        |

Symbole Znaczników c.d.

| STL  | LAD   |
|--|---|
| <b>A M2</b><br>lub<br><b>O M2</b><br>lub<br><b>X M2</b>    | <b>M2</b><br> |
| <b>AN M9</b><br>lub<br><b>ON M9</b><br>lub<br><b>XN M9</b> | <b>M9</b><br> |

#### 4.6.2. Znacznik MDIR



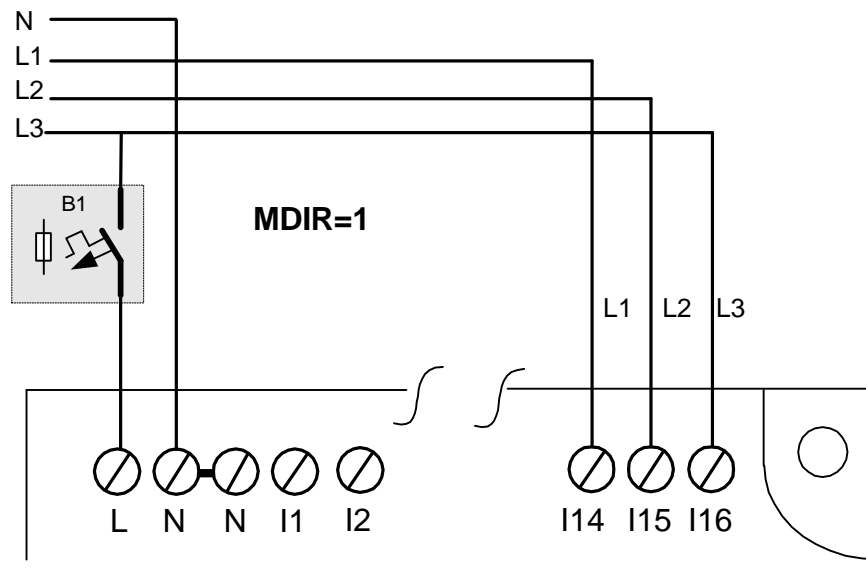
Dla wersji NEED-230VAC-x1-16-8 oprócz wspomnianych wyżej 16 Znaczników istnieje jeszcze Znacznik MDIR.

Znacznik MDIR określa kierunek faz L1, L2, L3 podłączanych do wejść I14, I15, I16. Jeśli do tych wejść zostaną podłączone kolejno fazy L1 (do I14), L2 (do I15), L3 (do I16) to MDIR przyjmuje wartość '1', w przeciwnym wypadku MDIR ma wartość '0'.

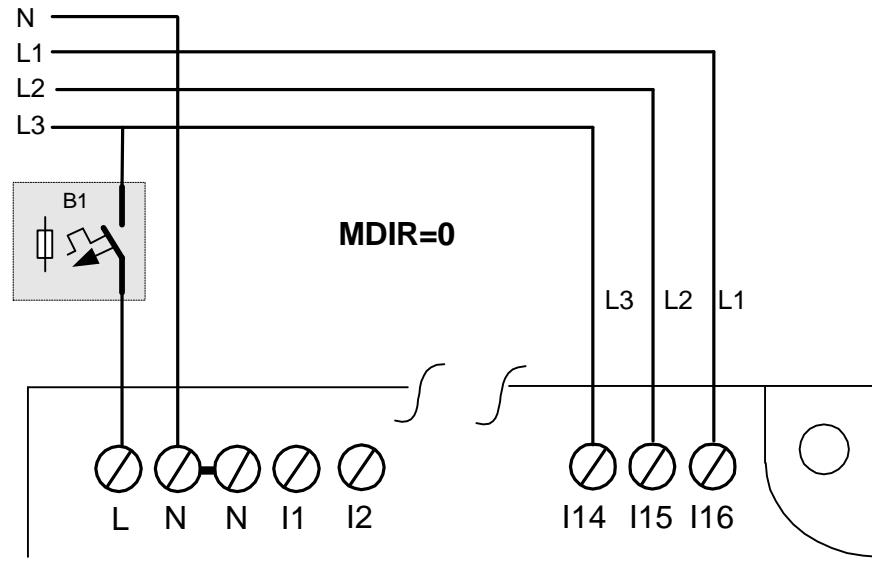
Znacznik MDIR jest tylko do odczytu.

Na rysunkach 4.6.1 i 4.6.2 przedstawiono przykładowe podłączenia, przy którym MDIR przyjmuje wartość kolejno '1' i '0'.

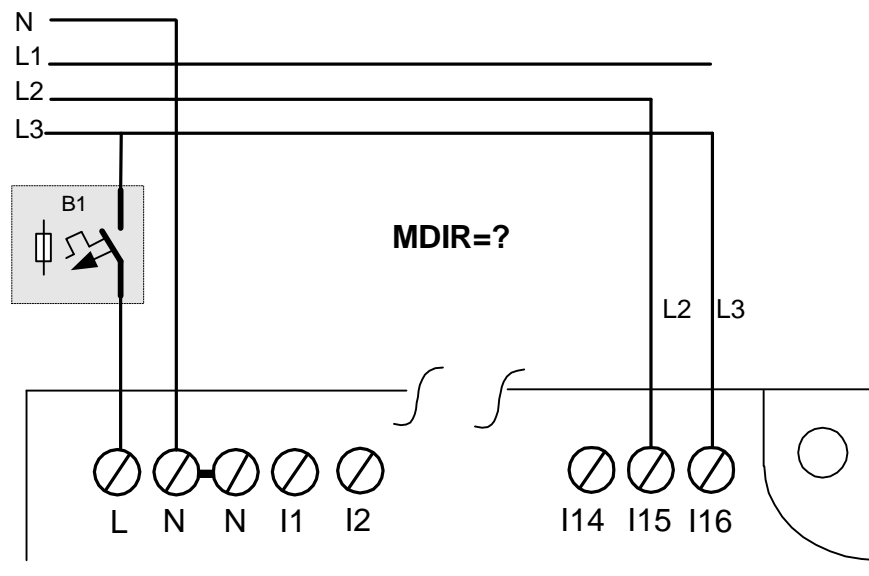
Jeśli do jednego z wejść analogowych I14, I15, I16 nie podłączymy przewodu sieciowego, wówczas Znacznik MDIR przyjmuje wartość losową – rysunek 4.6.3.



Rys.4.6.1. Podłączenie sieci trójfazowej, przy którym Znacznik MDIR przyjmuje wartość logiczną '1'.



Rys.4.6.2. Podłączenie sieci trójfazowej, przy którym Znacznik MDIR przyjmuje wartość logiczną '0'.



Rys.4.6.3. Podłączenie sieci trójfazowej, przy którym Znacznik MDIR przyjmuje wartości losowe '0' lub '1'.

#### 4.7. Timery

SYMBOL: **Tn**, gdzie n – jest numerem *Timera* : n=1..8 dla NEED..-x1-08-4  
n=1..16 dla NEED..-x1-16-8

STANY LOGICZNE WEJŚĆ **TRIGGER, RESET:**

'0' lub '1' w zależności od realizowanej funkcji w programie.

STANY LOGICZNE WYJŚCIA:

'0' lub '1' w zależności od realizowanej funkcji w programie.

ZAKRESY CZASOWE:

Wartości odmierzanych czasów przedstawione są w tabeli 4.7.

*Timer* jest elementem czasowym, dzięki któremu możemy wykorzystywać sterowanie czasowe w przekaźniku programowalnym.

Każdy z 8 *Timerów* może być używany w jednej z konfiguracji:

- Załączenie z opóźnieniem (ON-DELAYED),
- Wyłączenie z opóźnieniem (OFF-DELAYED),
- Pojedynczy impuls (SINGLE PULSE),
- Impulsy (FLASHING).

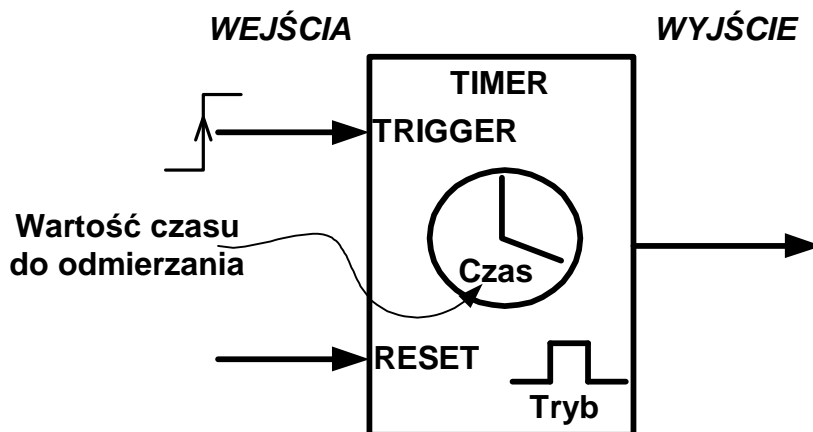
W strukturze logicznej *Timera* możemy wyróżnić wejścia, wyjście, tryb pracy oraz wartość czasu do odmierzania.

Wejścia i wyjścia *Timerów* można logicznie wiązać także poprzez sygnały bitowe (I,Q,M).

#### Wejścia.

W skład wejść wchodzi:

- wejście wyzwalające – TRIGGER (wejście to inicjuje działanie *Timera* np. rozpoczyna odmierzanie czasu)
- wejście resetujące – RESET (powoduje ustawienie wyjścia *Timera* w stan niski ('0') oraz zatrzymanie odmierzania czasu)



Rys. 4.7.1. Logiczna struktura Timera.

#### Czas do odmierzania.

Czas do odmierzania w *Timerach* ustawia się poprzez odpowiednie instrukcje ładujące (STL, LAD) lub za pomocą ustawień w pliku ustawień „\*.set”.

Zakresy czasowe *Timerów* przedstawiono w tabeli 4.7.

#### Tryb.

Rodzaj pracy *Timera* np. Załączenie z opóźnieniem, Pojedynczy impuls itp.

Tab. 4.7. Zakresy czasowe timerów.

| Format czasu               | Zakres              | Krok |
|----------------------------|---------------------|------|
| s.ms (sekundy.milisekundy) | 0s.10ms – 99s.990ms | 10ms |
| min.s (minuty.sekundy)     | 0min.1s – 99min.59s | 1s   |
| h.min (godziny.minuty)     | 0h.1min – 99h.59min | 1min |

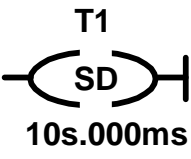
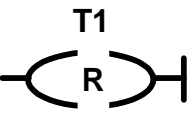
**Wyjścia.**

Wyjście *Timera* jest ustawiane lub resetowane w zależności od wyboru odpowiedniej funkcji czasowej (odpowiedniego typu *Timera*).

W programie wyjścia *Timerów* możemy używać tak, jak np. *Znaczników*, zastępując w oznaczeniu literę M literą T.

## 4.7.1. Timer „Opóźnione załączenie” (ON-DELAYD)

Symbole *Timera* SD.

| STL            | LAD   |
|----------------|---|
| L 10s<br>SD T1 |  <p>T1<br/>SD<br/>10s.000ms</p> |
| R T1           |  <p>T1<br/>R</p>               |

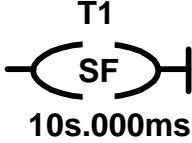
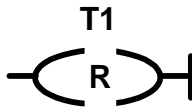
Realizowana funkcja czasowa:

Jeżeli na wejściu Trigger pojawi się zbocze narastające, a wejście Reset jest nieaktywne, to po czasie nastawionym w *Timerze*, następuje załączenie wyjścia *Timera* w stan '1' – wejście Trigger musi pozostać w stanie wysokim. Jeżeli wejście Trigger zmieni stan na '0', to nastąpi automatyczne resetowanie licznika czasu i kasowanie wyjścia.

Jeżeli wejście Reset zostanie ustawione na '1', to *Timer* zostaje skasowany w dowolnym momencie swojej pracy i przestanie odmierzać czas. Stan wyjścia powraca do stanu początkowego, czyli '0'. Ponowne rozpoczęcie liczenia czasu nastąpi dopiero po ustawieniu sygnału Reset w stan niski i pojawieniu się dodatniego zbocza na wejściu Trigger.



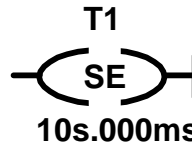
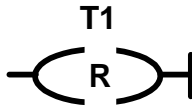
#### 4.7.2. Timer “Opóźnione wyłączenie” (OFF-DELAYD) Symbole *Timera* SF.

| STL            | LAD   |
|----------------|---|
| L 10s<br>SF T1 |  |
| R T1           |  |

Realizowana funkcja czasowa:

Jeżeli wejście Trigger ma stan ‘1’, przy stanie ‘0’ na wejściu Reset, to wyjście jest załączone. Jeśli teraz wejście Trigger przejdzie w stan niski (‘0’ – zbrocze opadające), to po czasie nastawionym w *Timerze*, następuje wyłączenie wyjścia *Timera* – ustawienie w stan ‘0’. Jeżeli wejście Trigger zmieni swój stan na ‘1’ nastąpi automatyczne resetowanie licznika czasu i ustawienie wyjścia ponownie w stan wysoki (‘1’). Jeżeli wejście Reset zostanie ustawione na ‘1’, to *Timer* zostanie skasowany i przestanie odmierzać czas, a jego wyjście przyjmie stan ‘0’. Ponowne rozpoczęcie liczenia czasu nastąpi dopiero po ustawieniu sygnału Reset w stan niski (‘0’) i pojawieniu się ujemnego zbocza wyzwalającego na wejściu Trigger.

#### 4.7.3. Timer “Pojedynczy impuls” (SINGLE PULSE) Symbole *Timera* SE.

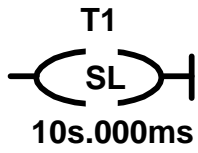
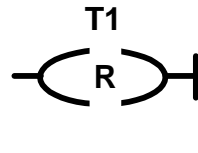
| STL            | LAD   |
|----------------|---|
| L 10s<br>SE T1 |  |
| R T1           |  |

Realizowana funkcja czasowa:

Jeżeli na wejściu wyzwalającym Trigger pojawi się sygnał wyzwalający (zbrocze narastające), a wejście Reset jest nieaktywne, to układ załącza wyjście na nastawiony czas, a następnie powraca do stanu ‘0’ – w tym czasie stan Trigger nie ma znaczenia (może przyjąć wartość ‘0’). Każde zbrocze dodatnie na wejściu Trigger przedłuży impuls o kolejny nastawiony czas. Ustawienie wejścia Reset w dowolnym momencie kasuje wyjście *Timera* do stanu ‘0’. Ponowne wyzwolenie może nastąpić po ustawieniu wejścia Reset w stan ‘0’ i po wystąpieniu kolejnego zbocza narastającego na wejściu Trigger.

#### 4.7.4. Timer "Impulsy" (FLASHING)

Symbole *Timera* SL.

| STL            | LAD   |
|----------------|---|
| L 10s<br>SL T1 |  |
| R T1           |  |

Realizowana funkcja czasowa:

*Timer* pełni funkcję generatora fali prostokątnej o wypełnieniu 50%. Układ zaczyna generować impulsy o nastawionym czasie trwania, gdy wejście wyzwalające *Trigger* przyjmuje stan '1'. Czasy trwania stanów '1' i '0' są równe, a częstotliwość pracy dla tego typu *Timera* wynosi:  $f=1/(2T)$ , gdzie T jest zadany czas do odmierzenia przez *Timer*. Ustawienie wejścia *Reset* w stan wysoki powoduje natychmiastowe ustawienie wyjścia w stan niski.

Gdy sygnał *Trigger* przyjmuje stan niski – wyjście *Timera* także przyjmuje stan '0'.

Zależności wejść *Trigger*, *Reset* i wyjścia *Timera* są następujące:

- Po podaniu '1' na *Trigger*, gdy wejście *Reset* jest w stanie '0', wyjście najpierw pozostaje w stanie '0' przez ustawiony czas, a potem ustawia się w stan '1' i cykl się powtarza.
- Jeżeli *Trigger* i *Reset* ustawione są równocześnie w stanie wysokim, to wyjście jest wyzerowane. W momencie, gdy *Reset* osiąga stan niski, a *Trigger* nadal pozostaje w stanie '1', to układ załącza wyjście na nastawiony czas, a potem je wyłącza, powtarzając cykl.

#### 4.8. Liczniki

SYMBOL: **Cn**, gdzie n – jest numerem *Licznika* od 1 do 8

STANY LOGICZNE WEJŚĆ **CU, CD, RESET**:

'0' lub '1' w zależności od realizowanej funkcji w programie.

STANY LOGICZNE WYJŚCIA:

'0' lub '1' w zależności od realizowanej funkcji w programie.

ZAKRESY WARTOŚCI ZLICZANYCH:

**0– 65535.**

*Licznik* zlicza impulsy pojawiające się w momencie wystąpienia narastającego zbocza na wejściu wyzwalającym liczenie.

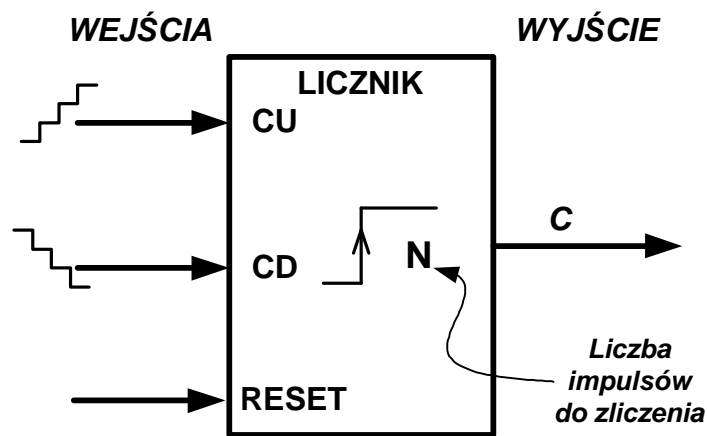
W strukturze logicznej *Licznika* możemy wyróżnić wejścia, wyjście oraz wartość liczbowa impulsów do zliczenia.

Wejścia i wyjście *Liczników* można logicznie wiązać także poprzez sygnały bitowe (I,Q,M).

#### Wejścia.

W skład wejść wchodzi:

- wejście resetujące – RESET – powoduje ustawienie wyjścia *Licznika* w stan niski ('0') oraz zatrzymuje zliczanie impulsów i zeruje *Licznik*.
- wejścia wyzwalające liczenie – CU,CD – sygnały podane na te wejścia, zbocza narastające, powodują liczenie odpowiednio w górę i w dół. Ten sam *Licznik* może więc zliczać jednocześnie i w górę, i w dół.



Rys. 4.8.1. Logiczna struktura Licznika.

#### Liczba impulsów do zliczenia.

Liczba impulsów do zliczenia od 0 do 65535 może być określona za pomocą instrukcji ładującej (STL) lub pliku ustawień („\*.set”).

#### Wyjście.

Wyjście *Licznika* jest ustawiane w stan wysoki lub resetowane (ustawiane w stan niski) w zależności od stanu *Licznika* zliczającego impulsy.

Wyjście *Licznika* przyjmuje stan '1' jeśli zachodzi relacja:

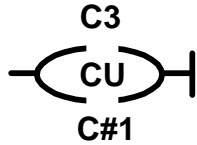
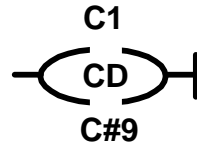
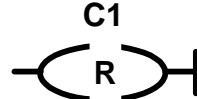
- Ilość impulsów aktualnie zliczonych jest większa lub równa od wartości zadanej do zliczenia.

Wyjście *Licznika* przyjmuje stan '0' jeśli zachodzi relacja:

- Ilość impulsów aktualnie zliczonych jest mniejsza od wartości zadanej do zliczenia lub wejście RESET jest aktywne.

Wystąpienie dodatniego zbocza na wejściu CU powoduje zwiększenie wartości zliczanej o 1. Natomiast dodatnie zbocze na wejściu CD zmniejsza wartość o 1. W programie wyjścia *Licznika* możemy używać tak, jak *Znaczników*, zastępując w oznaczeniu literę M literą C.

Symbole *Licznika*.

| STL            | LAD  |
|----------------|--|
| L C#3<br>CU C1 |   |
| L C#9<br>CD C1 |   |
| R C1           |  |

Zliczanie impulsów w górę:

- *Licznik* zlicza impulsy pojawiające się na wejściu CU.
- Zliczanie jest narastające. Jeśli wartość bieżąca Licznika jest większa lub równa zadanej liczbie impulsów, to wyjście *Licznika* przyjmuje stan '1'. Pozostaje ono tak długo w tym stanie, aż nie pojawi się na wejściu RESET stan wysoki, który kasuje wyjście i aktualny stan *Licznika* lub gdy *Licznik* zaczyna zliczać w dół i wartość bieżąca *Licznika* będzie mniejsza od wartości progowej.
- *Licznik* nigdy się nie przepełnia, jeśli liczba zliczonych impulsów osiągnie wartość 65535, to *Licznik* przestaje zliczać w górę.

Zliczanie impulsów w dół:

- *Licznik* zlicza impulsy pojawiające się na wejściu CD.
- Zliczanie jest malejące. Jeśli wartość bieżąca Licznika jest większa lub równa zadanej liczbie impulsów, to wyjście *Licznika* przyjmuje stan '1'. Pozostaje ono tak długo w tym stanie, aż nie pojawi się na wejściu RESET stan wysoki, który kasuje wyjście i aktualny stan *Licznika*.
- *Licznik* nigdy się nie przepełnia, jeśli liczba zliczonych impulsów osiągnie wartość 0, to *Licznik* przestaje zliczać w dół.



Dla wersji NEED-24DC-x1-16-8 oprócz wspomnianych wyżej 8 *Liczników* istnieje jeszcze jeden, *Szybki Licznik* HC zliczający impulsy o maksymalnej częstotliwości 20kHz. HC jest *Licznikiem* sprzętowym, zlicza impulsy pojawiające się na wejściu I11. Wejścia CU, CD przejmują wtedy, oprócz funkcji kierunku zliczania, funkcję aktywacji *Szybkiego Licznika*.

*Szybki Licznik* może pracować w trybie częstotliwościowym – zlicza ilość impulsów pojawiających się na wejściu I11 w czasie 1s.

*Szybki Licznik*, po osiągnięciu wartości maksymalnej - 65535, zaczyna liczenie od 0 po wykonaniu instrukcji Reset.



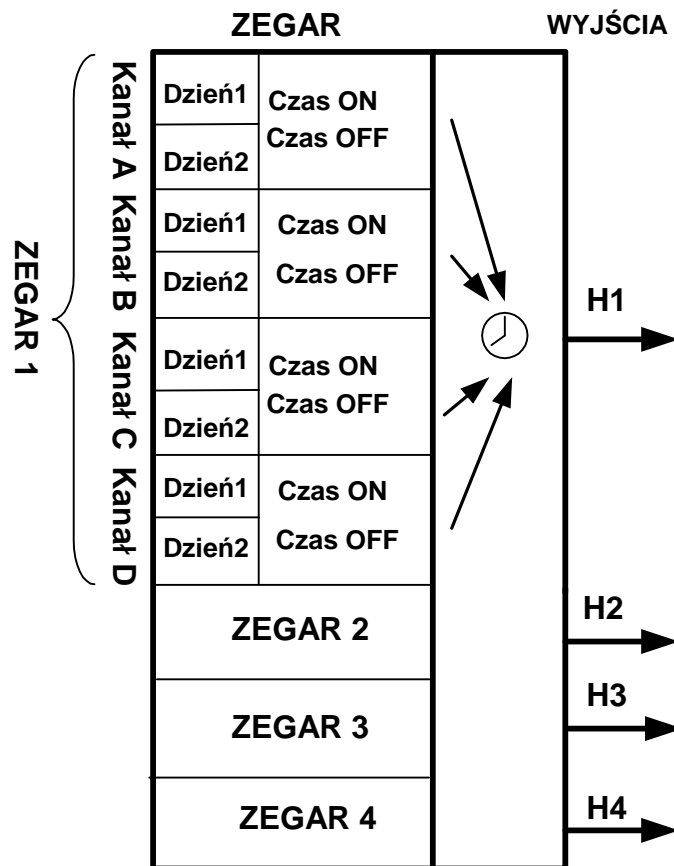
Dla wersji NEED-230VAC-x1-16-8 *Szybki Licznik* HC mierzy częstotliwość sieci (50Hz lub 60Hz) – dla trybu częstotliwościowego. Natomiast w trybie licznika zlicza impulsy sieci zasilającej co 20ms (dla częstotliwości sieci zasilającej 50Hz) lub 16,6ms (dla częstotliwości sieci zasilającej 60Hz)

#### 4.9. Zegary Tygodniowe

SYMBOL:  $H_n$ , gdzie  $n$  – jest numerem Zegara od 1 do 4.

STANY LOGICZNE WYJŚCIA:

'0' lub '1' w zależności od realizowanej funkcji w programie.

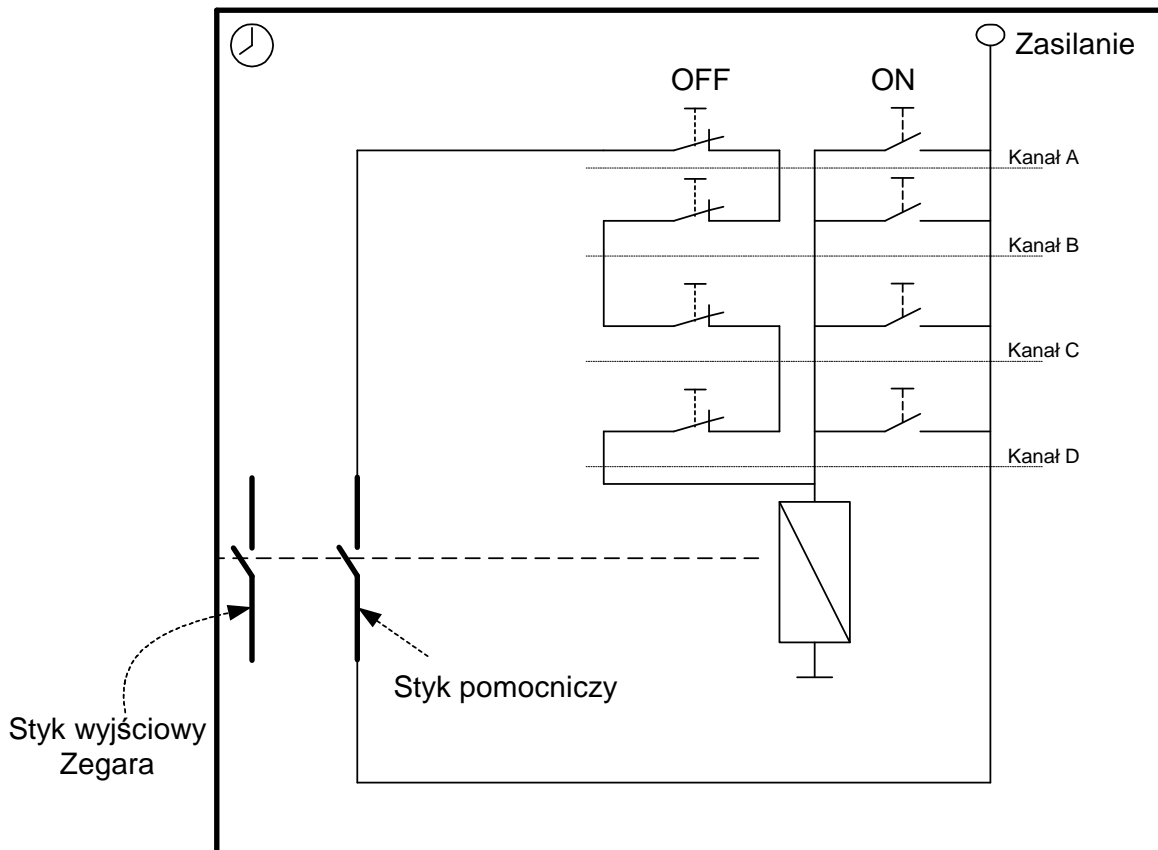


Rys. 4.9.1. Logiczna struktura Zegara.

Przy programowaniu przekaźnika NEED można użyć tygodniowe Zegary sterujące H1, H2, H3, H4. Każdy Zegar posiada cztery kanały A, B, C, D. Wyjście Zegara jest wspólne dla czterech kanałów. Na rys. 4.9.1. przedstawiono logiczną strukturę Zegarów.

##### 4.9.1. Działanie Zegara

Działanie Zegara w przekaźniku programowalnym możemy porównać do działania urządzenia, którego schemat ideowy przedstawia rys. 4.9.2. Załączenie Zegara odbywa się za pomocą przełączników „ON”, a wyłączenie za pomocą przełączników „OFF”. Ustawień Zegara dokonujemy używając programu PC Need.



Rys. 4.9.2. Schemat ideowy pojedynczego Zegara.

Przykład 1

Na rysunku 4.9.3. przedstawiono przykładowe okienko konfiguracyjne dla Zegara 1.



Rys. 4.9.3. Przykładowe okienko konfiguracyjne dla Zegara 1.

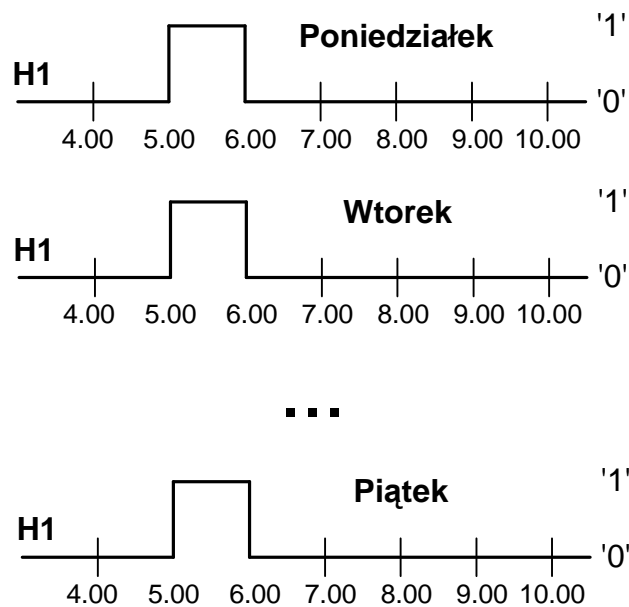
**Dzień pierwszy** – dzień pierwszy, w rozkładzie tygodniowym, w którym następuje załączenie/wyłączenie Zegara (w podanym przykładzie tym dniem jest poniedziałek).

**Ostatni dzień** – dzień ostatni, w rozkładzie tygodniowym, w którym następuje Załączenie/wyłączenie Zegara (w podanym przykładzie tym dniem jest piątek).

**Godzina załączenia** – godzina (dozwolone wartości od 0.00 do 23.59) załączenia wyjścia Zegara (w podanym przykładzie 5.00).

**Godzina wyłączenia** – godzina (dozwolone wartości od 0.00 do 23.59) wyłączenia wyjścia Zegara (w podanym przykładzie 6.00).

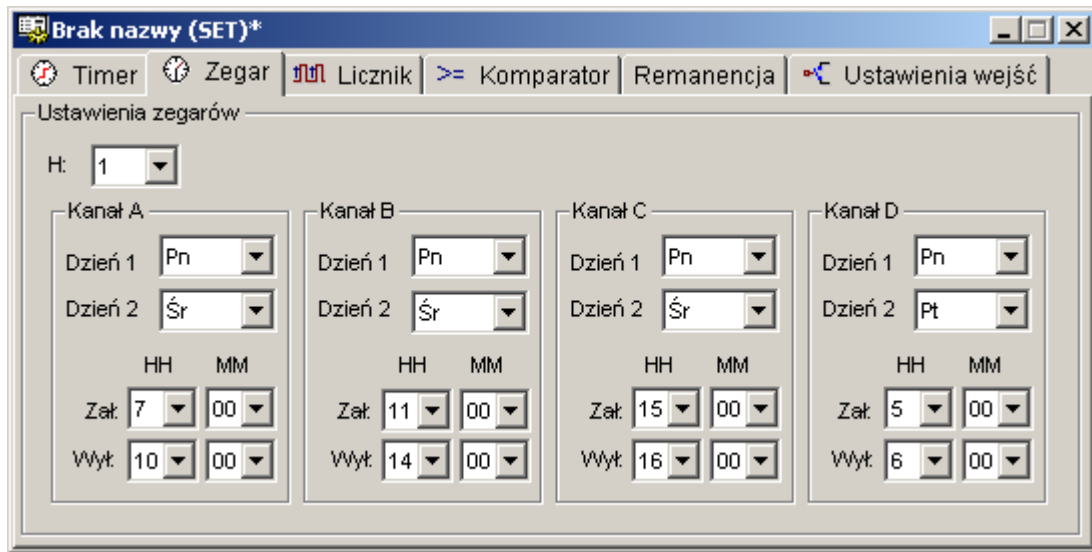
Dla przedstawionej konfiguracji Zegar 1 będzie ustawiał swoje wyjście w stan wysoki w każdy dzień od poniedziałku do piątku między 5.00 (godzina załączenia) a 6.00 (godzina wyłączenia). Przedstawia to rys. 4.9.4.



Rys. 4.9.4. Działanie Zegara 1 dla konfiguracji przedstawionej na rys. 4.9.4.

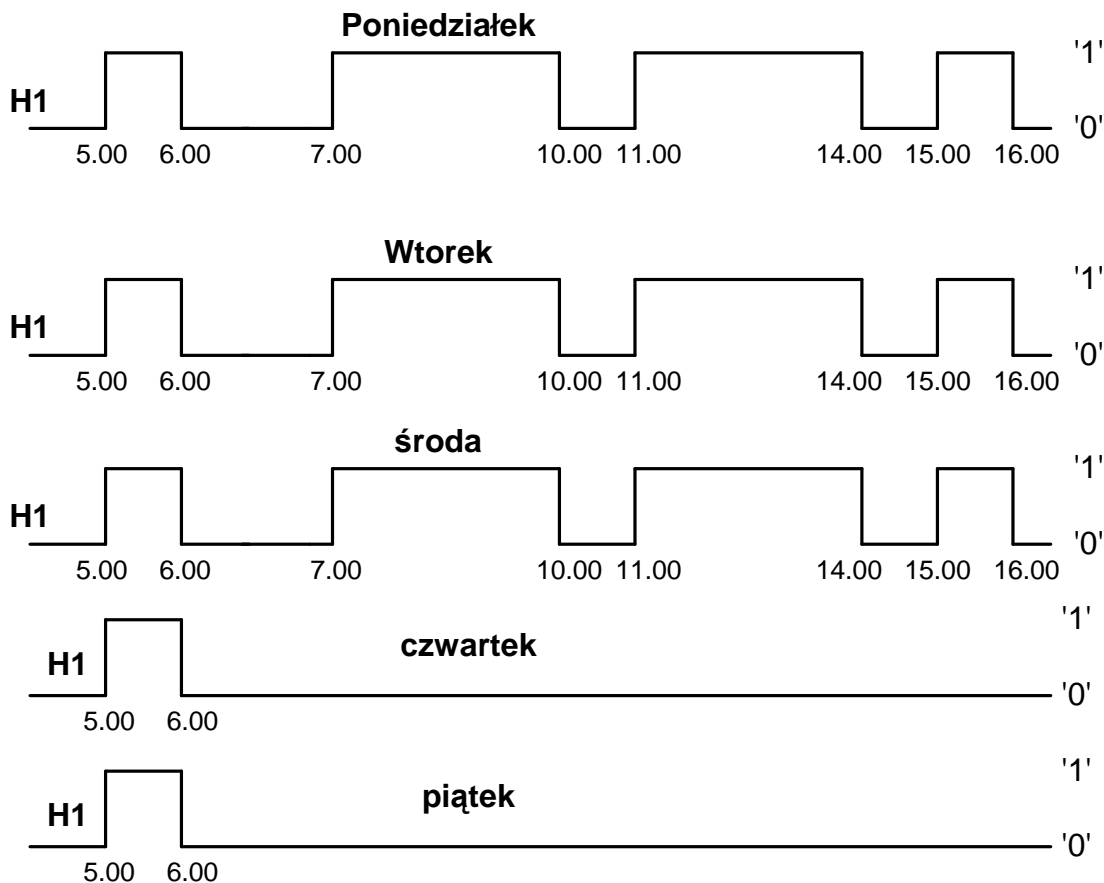
## Przykład 2

Na rysunku 4.9.5. przedstawiono przykładowe okienko konfiguracyjne dla Zegara 1.



Rys. 4.9.5. Przykładowe okienko konfiguracyjne dla Zegara 1.

Dla przedstawionej konfiguracji Zegar 1 będzie ustawiał swoje wyjście w stan wysoki w każdy dzień od poniedziałku do środy włącznie, między 7.00 (godzina załączenia) a 10.00 (godzina wyłączenia), 11.00 (godzina załączenia) a 14.00 (godzina wyłączenia), 15.00 (godzina załączenia) a 16.00 (godzina wyłączenia). Dodatkowo od poniedziałku do piątku wyjście Zegara 1 będzie ustawiane między 5.00 a 6.00. Przedstawia to rys. 4.9.6.

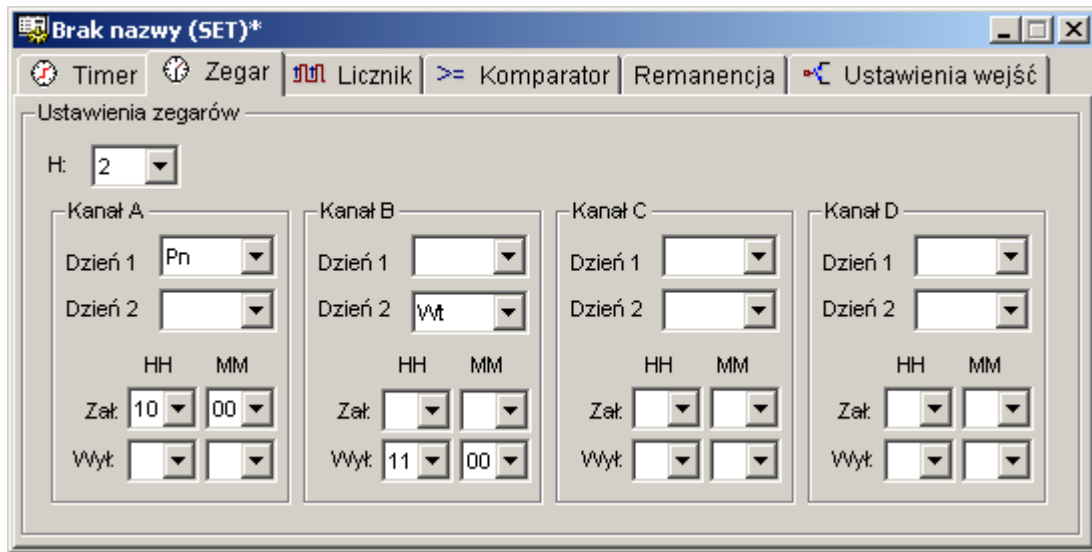


Rys. 4.9.6. Działanie Zegara 1 dla konfiguracji przedstawionej na rys.4.9.6.



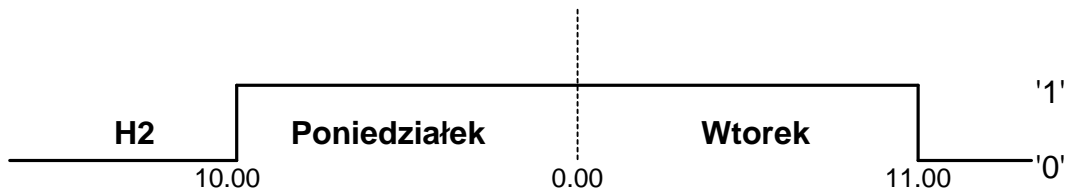
## Przykład 3

Na rysunku 4.9.7. przedstawiono przykładowe okienko konfiguracyjne dla Zegara 2.



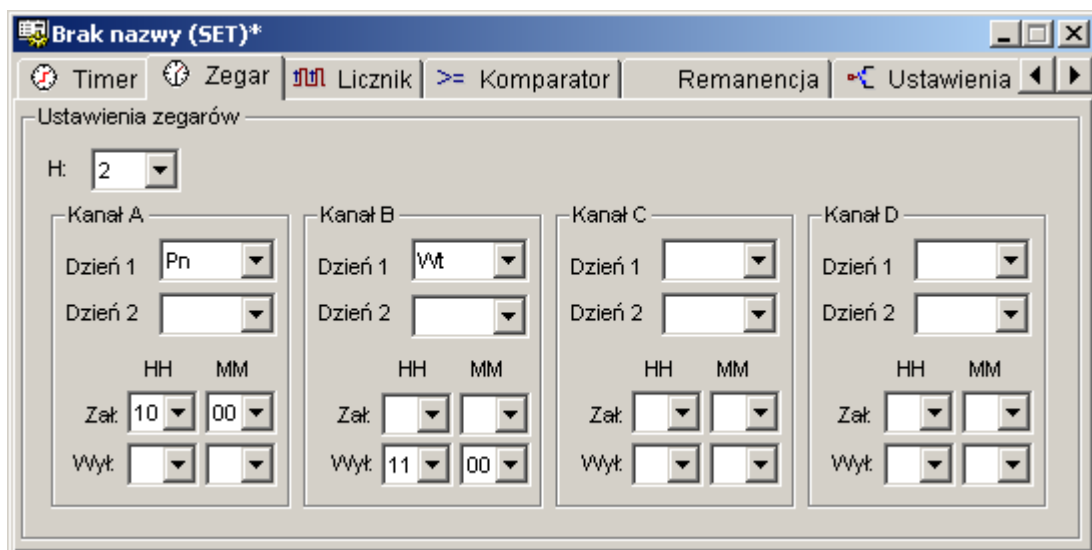
Rys.4.9.7. Przykładowe okienko konfiguracyjne dla Zegara 2.

Dla przedstawionej konfiguracji Zegar 2 będzie ustawiał swoje wyjście w stan wysoki w każdy poniedziałek o godz.10.00 (godzina załączenia) a we wtorek o godzinie 11.00 wyłączał się. Jeśli chcemy rozciągnąć sterowanie na kilka dni, to wtedy nie wypełniamy pola „Godzina załączenia” lub „Godzina wyłączenia” w odpowiednich kanałach. Rysunek 4.9.8. przedstawia diagram pracy Zegara 2.



Rys. 4.9.8. Działanie Zegara 2 dla konfiguracji przedstawionej na rys. 4.9.8.

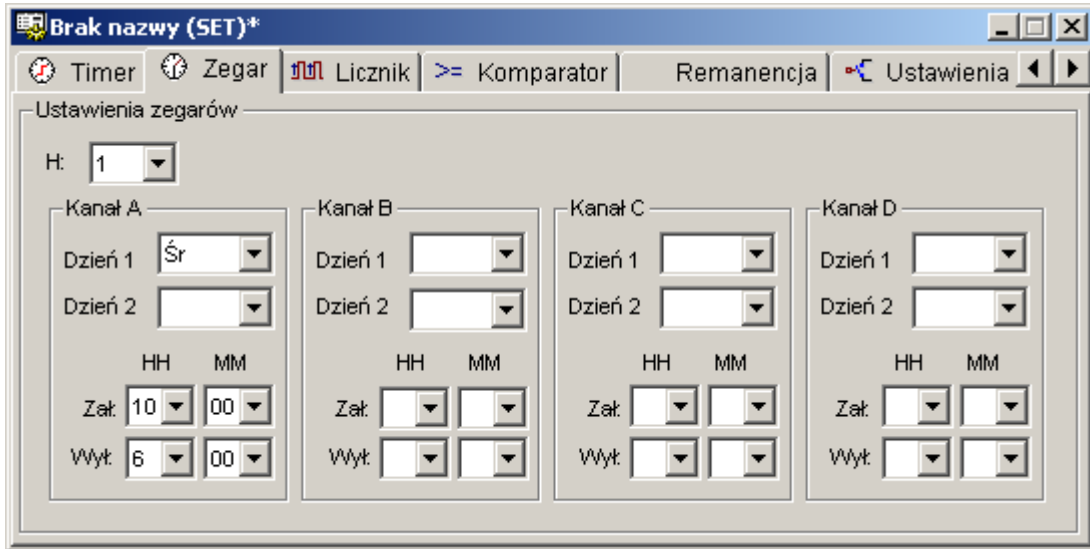
Identyczne działanie Zegara 2 uzyskamy, jeśli dokonamy konfiguracji tak, jak na rys. 4.9.9.



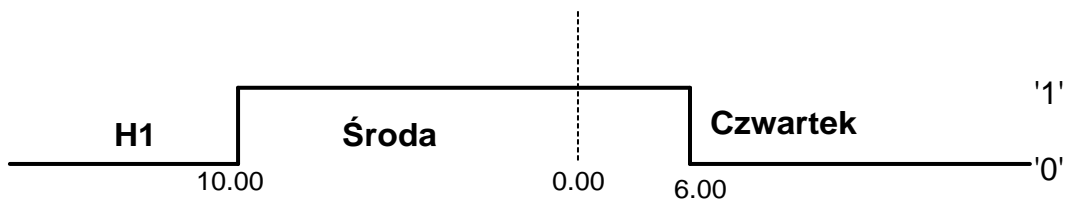
Rys. 4.9.9. Przykładowe okienko konfiguracyjne dla Zegara 2.

## Przykład 4

Jeśli godzina załączenia jest późniejsza od godziny wyłączenia, to Zegar wyłącza swoje wyjście następnego dnia – konfiguracja tak, jak na rys. 4.9.10.



Rys. 4.9.10. Przykładowe okienko konfiguracyjne dla Zegara 1.



Rys. 4.9.11. Działanie Zegara 1 dla konfiguracji przedstawionej na rys. 4.9.10.

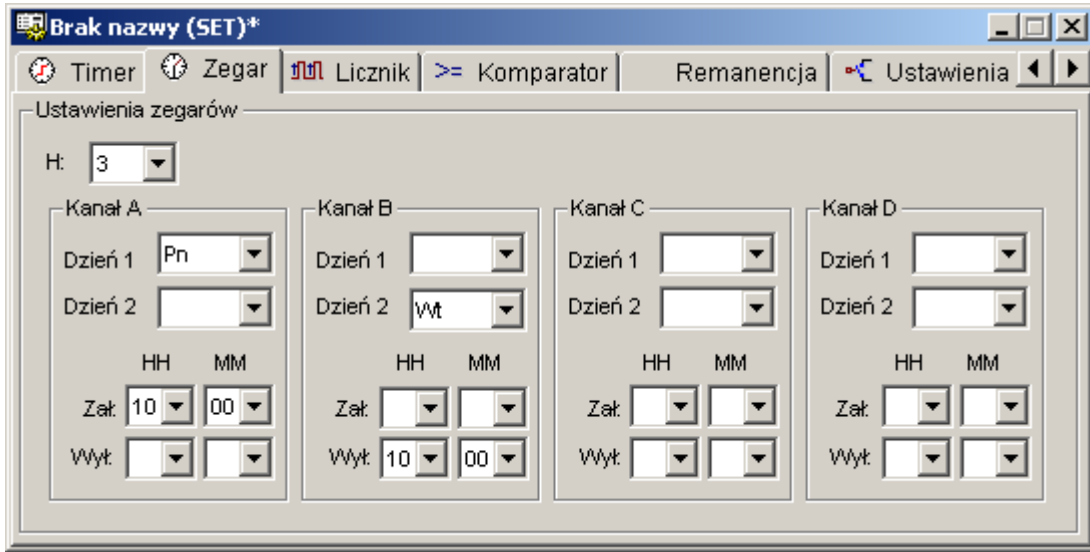


Jeśli godzina wyłączenia jest mniejsza od godziny załączenia, to przekaźnik programowalny NEED wyłącza wyjście używanego Zegara w następnym dniu.

## Przykład 5

Załączenie wyjścia Zegara na 24 godziny.

Dla pracy 24-godzinnej należy dokonać konfiguracji Zegara 3 tak, jak na rys. 4.9.12.

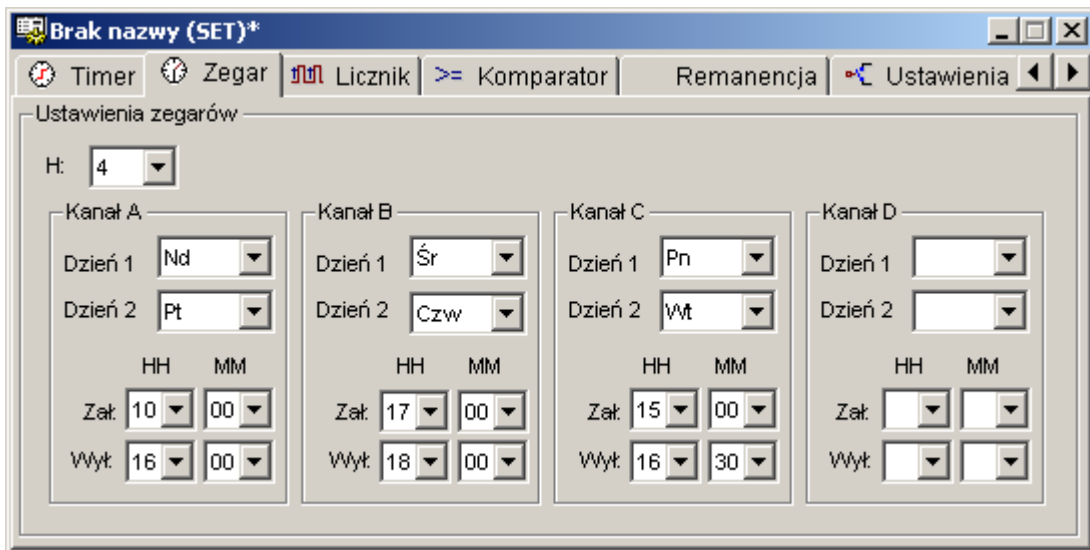


Rys. 4.9.12. Przykładowe okienko konfiguracyjne dla 24-godzinnego

## Przykład 6

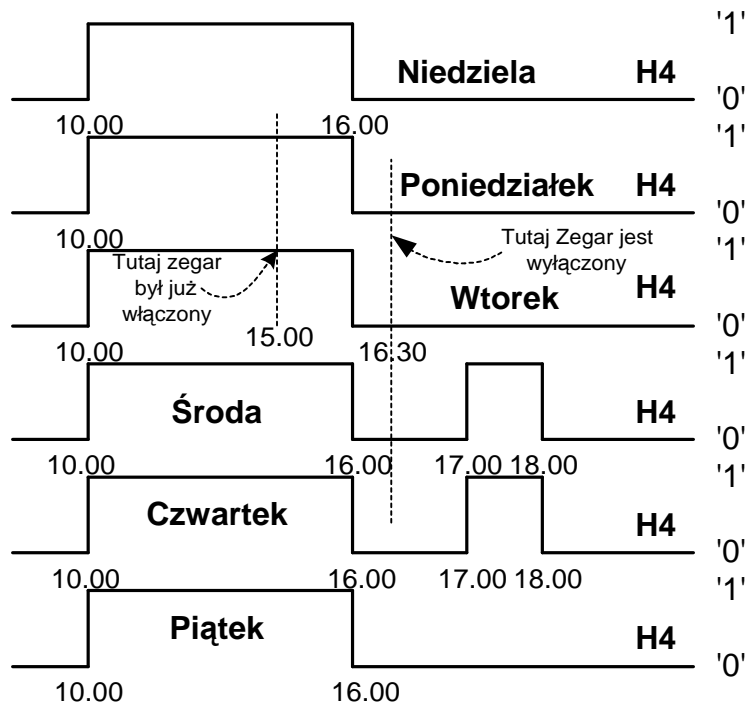
Należy pamiętać, iż stan wyjścia Zegara zależy od stanu wszystkich czterech kanałów.

Rozpatrzmy konfigurację Zegara 4 przedstawioną na rys. 4.9.13.



Rys. 4.9.13. Przykładowe okienko konfiguracyjne Zegara 4.

Zauważmy, iż czasy ustawione w kanale A i C pokrywają się – rys. 4.9.14.

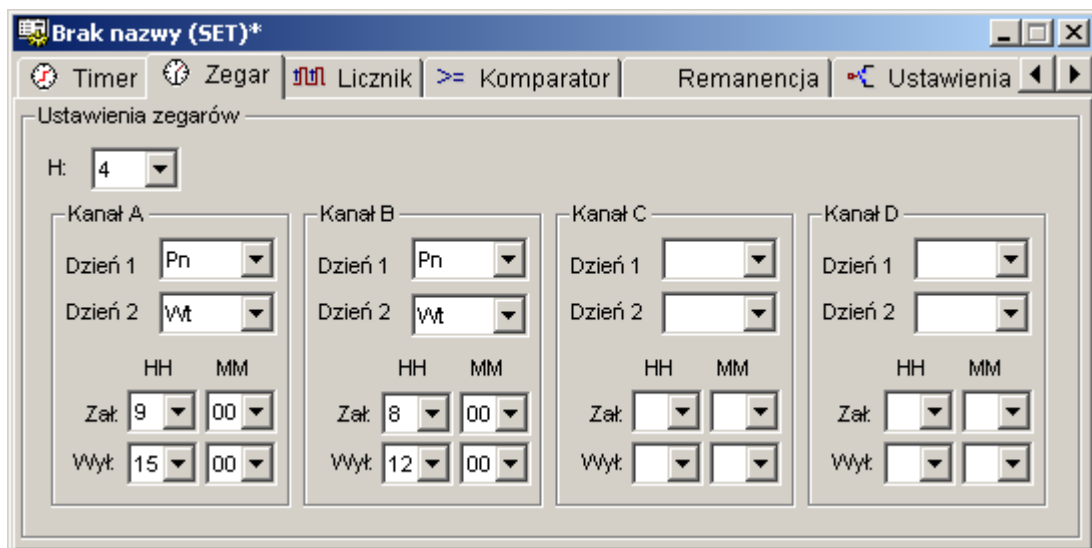


Rys. 4.9.14. Działanie Zegara 4 dla konfiguracji przedstawionej na rys. 4.9.13.

Czasy ustawień Zegara mogą nachodzić na siebie, należy zawsze pamiętać, iż wyjście Zegara załącza kanał z ustawioną wcześniejszą godziną załączenia, a wyłącza kanał z ustawioną wcześniejszą godziną wyłączenia.

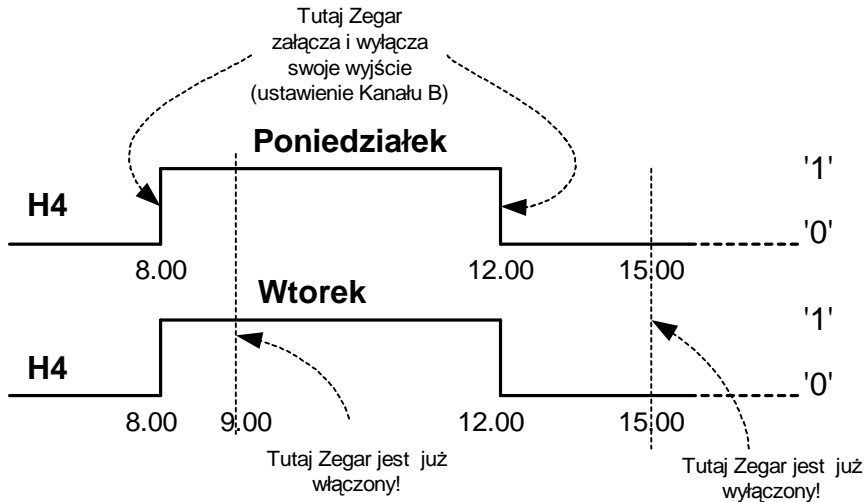
#### Przykład 7

Rozpatrzmy konfigurację Zegara 4 przedstawioną na rys. 4.9.15.



Rys. 4.9.15. Przykładowe okienko konfiguracyjne Zegara 4.

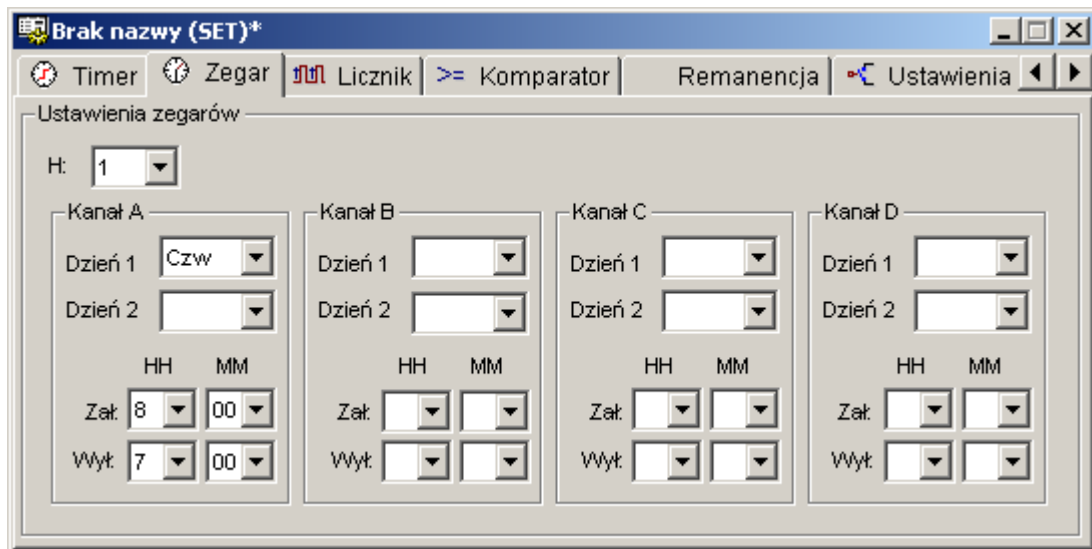
Zegar załączy swoje wyjście w poniedziałek i wtorek o godzinie 8.00, a wyłączy o godzinie 12.00 (a nie o 15.00! – stosujemy zasadę: pierwszy załącza i pierwszy wyłącza). Na rys. 4.9.16. przedstawiono diagram pracy dla Zegara 4.



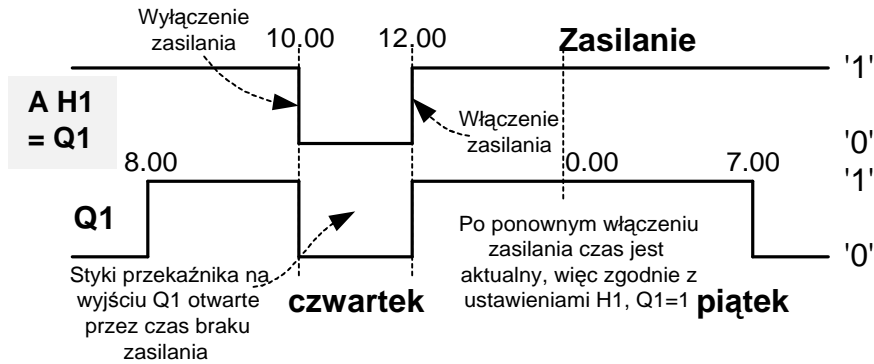
Rys. 4.9.16. Działanie Zegara 4 dla konfiguracji z rys. 4.9.16.

## Przykład 8

Na rys. 4.9.17. przedstawiona jest konfiguracja Zegara 1. Jeśli między 10.00 a 12.00 zostanie wyłączone zasilanie, wówczas styki wyjściowe przekaźnika zostaną rozwarpte, ale czas nadal będzie kontrolowany. Przy powrocie napięcia zasilającego o godz. 12.00, wyjście Zegara H1 będzie w stanie wysokim – zgodnie z konfiguracją H1. Powyższą sytuację przedstawia rys. 4.9.18.



Rys. 4.9.17. Przykładowe okienko konfiguracyjne Zegara 1.



Rys. 4.9.18. Działanie Zegara 1 dla konfiguracji z rys. 4.9.17.



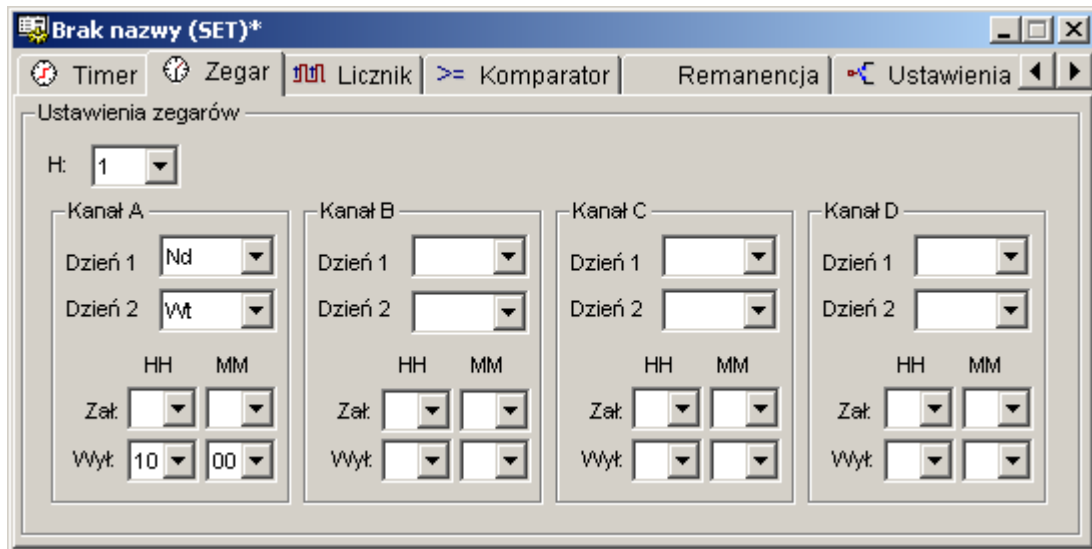
Przy zaniku zasilania czas w przekaźniku jest odmierzany dalej, jednak styki przekaźników wyjściowych nie zamykają się. Czas podtrzymywania Zegara w trakcie zaniku zasilania wynosi 64 godziny (dla 25°C).

#### 4.9.2. Uwagi dotyczące konfiguracji Zegara

##### 4.9.2.1. Jedno pole puste – 3 pola wypełnione (dla jednego kanału)

1. Wypełnione wszystkie pola, niewypełnione pole „Za”.

Przykład:



Rys. 4.9.2.1.1. Przykładowa konfiguracja Zegara – niewypełnione pole „Za”.

Od niedzieli do wtorku Zegar będzie wyłączał swoje wyjście o godz. 10.00.

2. Wypełnione wszystkie pola, niewypełnione pole „Wył”

Przykład:

Rys. 4.9.2.1.2. Przykładowa konfiguracja Zegara – wypełnione 3 pola.



Od niedzieli do wtorku Zegar będzie załączał swoje wyjście o godz. 5.00. W przypadku, gdy ustawiona jest tylko godzina załączenia, Zegar będzie zawsze załączony!

3. Wypełnione wszystkie pola, niewypełnione pole „Dzień 1”.

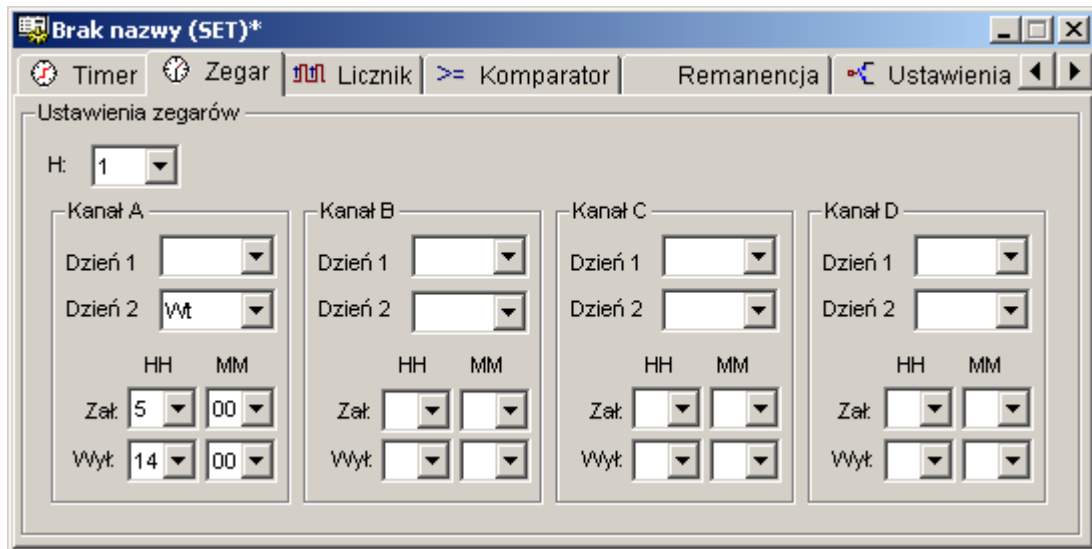
Przykład:

Rys. 4.9.2.1.3. Przykładowa konfiguracja Zegara – niewypełnione pole „Dzień 1”.

Zegar będzie załączał swoje wyjście tylko we wtorki o godz. 5.00, a wyłączał je tylko we wtorki o godz. 14.00.

4. Wypełnione wszystkie pola, niewypełnione pole „Dzień 2”.

Przykład:



Rys. 4.9.2.1.4. Przykładowa konfiguracja Zegara – niewypełnione pole „Dzień 2”.

Zegar załączy swoje wyjście tylko w niedzielę o godz. 5.00, a wyłączy je tylko w niedzielę o godz. 14.00.

4.9.2.2. Dwa pola puste – 2 pola wypełnione (dla jednego kanału)

1. Niewypełnione pola „Zał”, „Wył”.

Przykład:



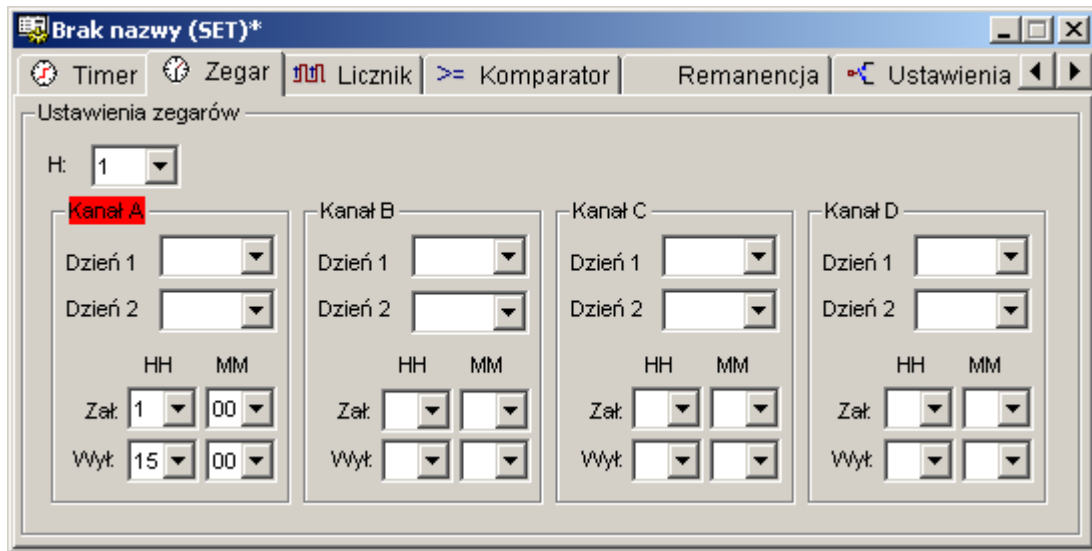
Rys. 4.9.2.2.1. Przykładowa konfiguracja Zegara – niewypełnione pola „Zał” i „Wył”.



Zegar nie działa – ustawienie niedozwolone, którego nie można przesłać do przekaźnika.

2. Niewypełnione pola „Dzień 1”, „Dzień 2”.

Przykład:



Rys. 4.9.2.2.2. Przykładowa konfiguracja Zegara – niewypełnione pola „Dzień 1” i „Dzień2”.

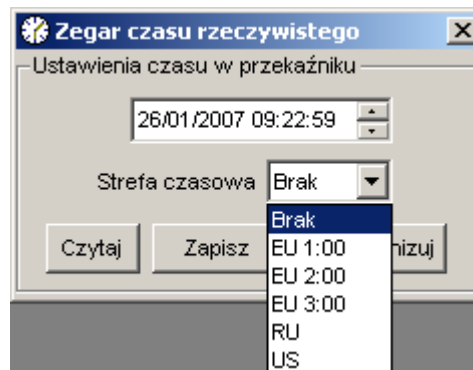
Zegar nie działa – ustawienie niedozwolone, którego nie można przesłać do przekaźnika.

4.9.2.3. Trzy pola puste (dla jednego kanału)

Zegar nie działa – ustawienie niedozwolone, którego nie można przesłać do przekaźnika.

#### 4.10. Zegar Czasu Rzeczywistego

Zegar Czasu Rzeczywistego pozwala na ustawienie w przekaźniku NEED..-x1-16-8 godziny, daty oraz strefy czasowej, dla której występuje odpowiednie przesunięcie czasu z letniego na zimowy i na odwrót. Ustawień dla Zegara Czasu Rzeczywistego dokonuje się za pomocą programu PCNeed. Na rys. 4.10.1. przedstawione jest okno, w którym dokonuje się odpowiednich ustawień.



Rys. 4.10.1. Okno ustawień dla Zegara Czasu Rzeczywistego.

Zegar Czasu Rzeczywistego pozwala na ustawienie następujących stref czasowych:

EU 1:00 – strefa, w której zmiana czasu na czas letni odbywa się w ostatnią niedzielę marca z godziny 1:00 na 2:00, a zmiana na czas zimowy w ostatnią niedzielę października z 2:00 na 1:00.

EU 2:00 – strefa, w której zmiana czasu na czas letni odbywa się w ostatnią niedzielę marca z godziny 2:00 na 3:00, a zmiana na czas zimowy w ostatnią niedzielę października z 3:00 na 2:00.

EU 3:00 – strefa, w której zmiana czasu na czas letni odbywa się w ostatnią niedzielę marca z godziny 3:00 na 4:00, a zmiana na czas zimowy w ostatnią niedzielę października z 4:00 na 3:00.

RU – strefa, w której zmiana czasu na czas letni odbywa się w ostatnią niedzielę marca z godziny 2:00 na 3:00, a zmiana na czas zimowy w ostatnią niedzielę października z 3:00 na 2:00.

US – strefa, w której zmiana czasu na czas letni odbywa się w drugą niedzielę marca z godziny 2:00 na 3:00, a zmiana na czas zimowy w pierwszą niedzielę listopada z 3:00 na 2:00.



#### 4.11. Komparator – wejścia analogowe

SYMBOL: **An**, gdzie n – jest numerem komparatora: n=1..8 dla NEED..-x1-08-..  
n=1...12 dla NEED..-x1-16-..

STANY LOGICZNE WEJŚCIA:

'0' lub '1' w zależności od wielkości napięć analogowych i ustawień konfiguracyjnych przekaźnika programowalnego.

Symbole *Komparatora*

| STL  | LAD   |
|--|---|
| <b>A A1</b><br>lub<br><b>O A1</b><br>lub<br><b>X A1</b>    | <b>A1</b><br> |
| <b>AN A1</b><br>lub<br><b>ON A1</b><br>lub<br><b>XN A1</b> | <b>A1</b><br> |

Układ przekaźnika programowalnego posiada w swojej strukturze dwa (NEED..-x1-08-4) lub trzy (NEED..-x1-16-8) wejścia analogowe. Na rys. 4.11.1. przedstawiono logiczną strukturę *Komparatora* dla przekaźnika NEED..-x1-08-4.

Sygnaly analogowe w *Komparatorze* mogą być porównywane ze sobą, z ustaloną Wartością Wzorcową oraz z nastawą zewnętrznego potencjometru. Od wyniku tego porównania zależy stan wyjść *Komparatora*. Wyjścia ustawiane zawsze w stan wysoki ('1'), jeśli warunek porównania jest spełniony. Tabele 4.11.1. oraz 4.11.2. przedstawiają dostępne porównania.

Tab. 4.11.1. Możliwe konfiguracje porównań Komparatora dla NEED.. – 08-4.

| Lp. | Typ porównania             |
|-----|----------------------------|
| 1.  | $I7 \geq$ Wartość Wzorcowa |
| 2.  | $I7 \leq$ Wartość Wzorcowa |
| 3.  | $I8 \geq$ Wartość Wzorcowa |
| 4.  | $I8 \leq$ Wartość Wzorcowa |
| 5.  | $I7 \geq$ Potencjometr     |
| 6.  | $I7 \leq$ Potencjometr     |
| 7.  | $I8 \geq$ Potencjometr     |
| 8.  | $I8 \leq$ Potencjometr     |
| 9.  | $I7 \geq I8$               |
| 10. | $I7 \leq I8$               |

Tab. 4.11.2. Możliwe konfiguracje porównań Komparatora dla NEED.. – 16-8.

| Lp.  | Typ porównania               |
|------|------------------------------|
| 1.   | $I14 \geq$ Wartość Wzorcowa  |
| 2.   | $I4 \leq$ Wartość Wzorcowa   |
| 3.   | $I15 \geq$ Wartość Wzorcowa  |
| 4.   | $I15 \leq$ Wartość Wzorcowa  |
| 5.   | $I16 \geq$ Wartość Wzorcowa  |
| 6.   | $I16 \leq$ Wartość Wzorcowa  |
| 7.   | $I14 \geq$ Potencjometr      |
| 8.   | $I14 \leq$ Potencjometr      |
| 9.   | $I15 \geq$ Potencjometr      |
| 10.  | $I15 \leq$ Potencjometr      |
| 11.  | $I16 \geq$ Potencjometr      |
| 12.  | $I16 \leq$ Potencjometr      |
| 13.  | $I14 \geq I15$               |
| 14.  | $I14 \leq I15$               |
| 15.  | $I14 \geq I16$               |
| 16.  | $I14 \leq I16$               |
| 17.  | $I15 \geq I16$               |
| 18.  | $I15 \leq I16$               |
| 19.* | $ASYM \geq$ Wartość Wzorcowa |
| 20.* | $ASYM \leq$ Wartość Wzorcowa |
| 21.* | $ASYM \geq$ Potencjometr     |
| 22.* | $ASYM \leq$ Potencjometr     |

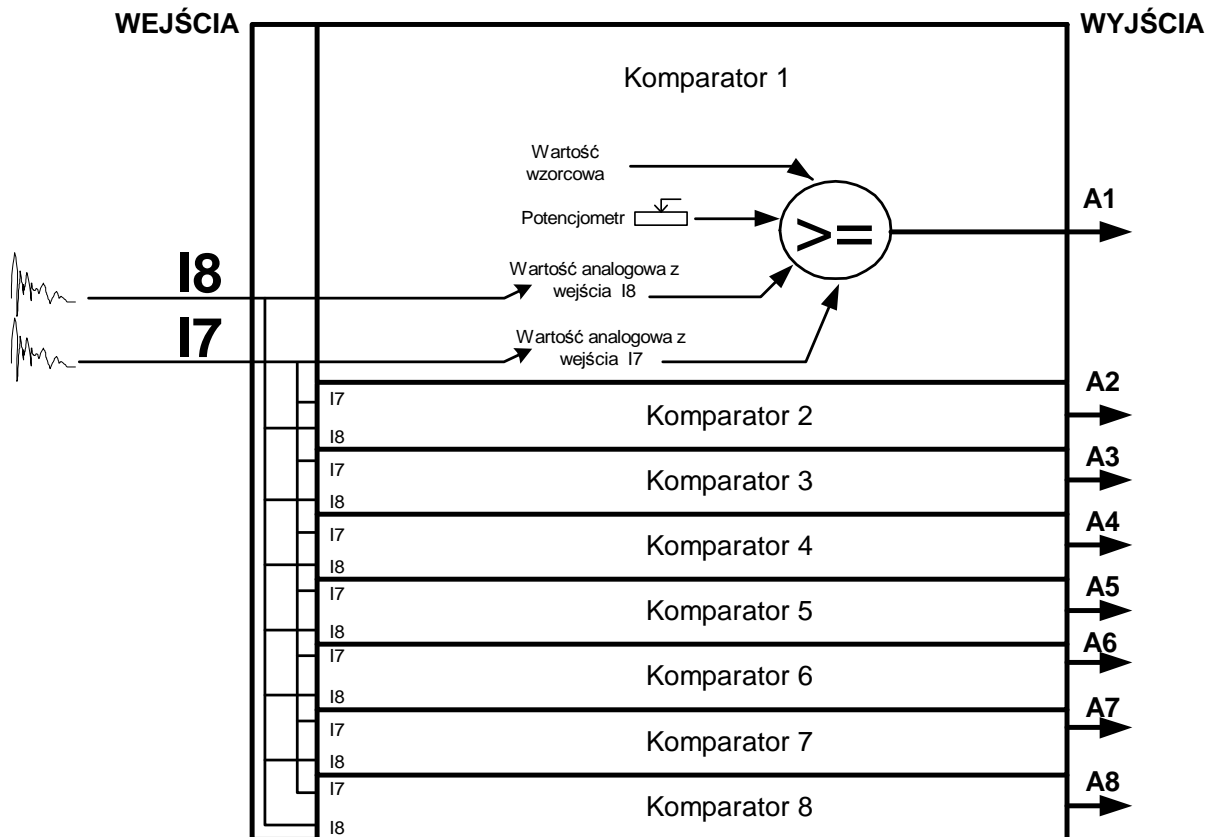
\* - możliwe tylko dla NEED-230AC-01-16-8



ASYM jest wskaźnikiem asymetrii faz (tylko w wersji NEED-230AC-x1-16-8). Pokazuje wartości skuteczne sumy faz L1, L2, L3. ASYM przyjmuje wartość 0V dla prawidłowych poziomów faz L1, L2, L3. Gdy występuje asymetria (poziom napięcia na którejś z faz jest różny od nominalnego) ASYM przyjmuje wartości większe od 0V.



ASYM wraz z Znacznikiem MDIR oraz Komparatorami sprawiają, iż NEED-230AC-x1-16-8 może pełnić funkcję przekaźnika nadzorczego monitorując asymetrię, kolejność oraz poziom napięć faz L1, L2, L3.



Rys. 4.11.1. Logiczna struktura Komparatora dla przekaźnika NEED..-08-4.

Do porównań używamy

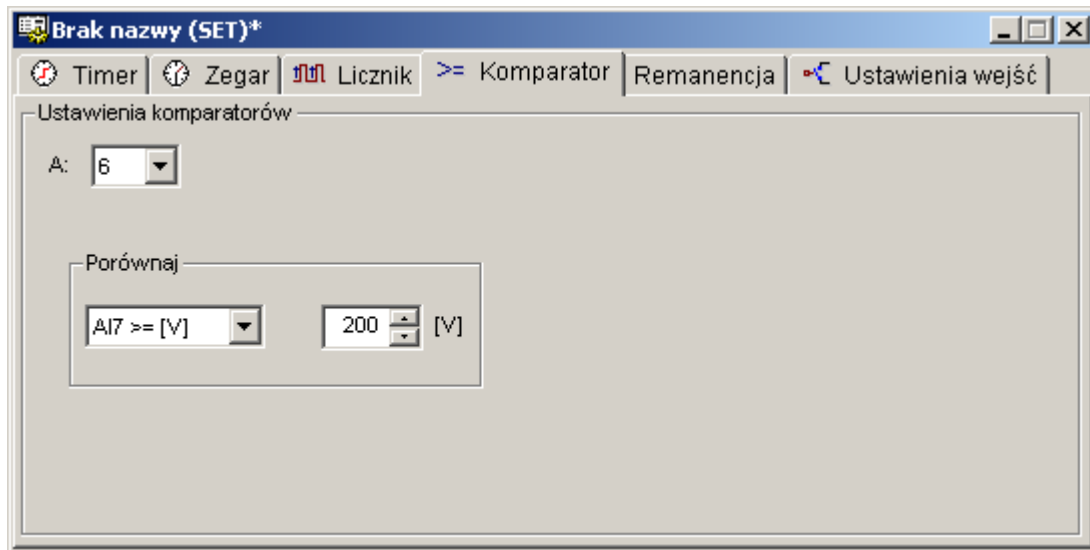
1. Wartości Wzorcowej (ustalanej na etapie konfiguracji w programie PC Need w zakresie:  
0 – 255V dla NEED-230AC-x1-..  
0 – 25,5V i 0 – 12,75V dla NEED-24DC-x1-16-8., NEED-12DC-x1-16-8.
2. Potencjometru (zakres regulacji 1 – 255 dla NEED-230AC-x1-.. oraz 0,1 – 25,5 dla NEED-24DC-x1-16-8., NEED-12DC-x1-16-8) – dostępnego na płycie czołowej przekaźnika.
3. Wartości napięcia z wejść analogowych.

W tabeli 4.11.3 przedstawiono zakresy przyjmowane przez Wartość Wzorcową.

Tab. 4.11.3. Możliwe zakresy Wartości Wzorcowej do porównań komparatorów.

| Typ               | Zakres Wartości Wzorcowej |
|-------------------|---------------------------|
| NEED-230AC-x1-..  | 0 – 255V                  |
| NEED-12DC-x1-..   | 0 – 25,5V                 |
| NEED-24DC-x1-..   | 0 – 25,5V                 |
| NEED-12DC-x1-16-8 | 0 – 12,75V                |
| NEED-24DC-x1-16-8 | 0 – 12,75V                |

Przykład konfiguracji komparatora A6 do porównania z Wartością Wzorcową przedstawiono na rys. 4.11.3.



Rys. 4.11.3. Przykład konfiguracji komparatora A6 do porównania z Wartością Wzorcową.

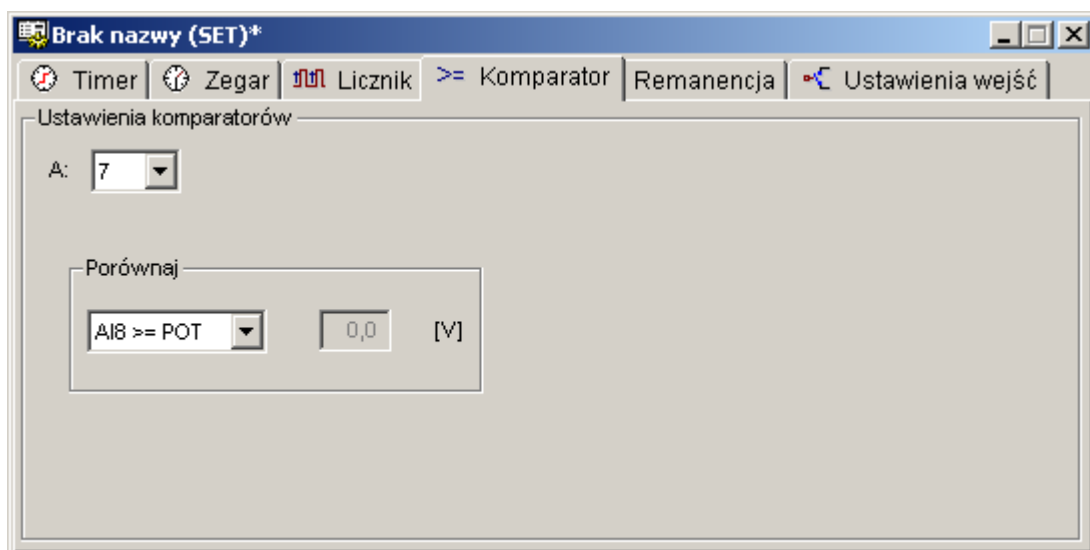
Wyjście Komparatora A6 zostanie ustawione w stan '1', gdy wartość napięcia na wejściu I7 będzie większa lub równa 200V.

W tabeli 4.11.4 przedstawiono zakres Potencjometru do porównań komparatorów.

Tab. 4.11.4. Możliwe wartości Potencjometru do porównań komparatorów.

| Typ              | Zakres Potencjometru |
|------------------|----------------------|
| NEED-230AC-x1-.. | 1 - 255              |
| NEED-12DC-x1-..  |                      |
| NEED-24DC-x1-..  |                      |

Przykład konfiguracji komparatora A7 do porównania z Potencjometrem przedstawiono na rys. 4.11.4.



Rys. 4.11.4. Przykład konfiguracji komparatora A7 do porównania z Potencjometrem.

Wyjście *Komparatora A7* zostanie ustawione w stan '1', gdy wartość napięcia na wejściu I8 będzie większa lub równa wartości ustalonej za pomocą Potencjometru.



Dla NEED-24DC-x1.., NEED-12DC-x1.. Potencjometr przyjmuje wartości od 0,1 do 25,5. Należy zwrócić uwagę na to, iż w okienku „Podglądu zmiennych” POT przyjmuje wartości 1 – 255, ale do porównań komparatora używane są wartości od 1/10 (czyli 0,1) do 255/10 (czyli 25,5)!

Na przykład dla relacji „A17<=POT” przy napięciu na A17=5V, wyjście komparatora będzie ustawiane w stan wysoki np. dla ustawień zmiennej POT=50 (czyli 50/10) w „Podglądzie zmiennych”.

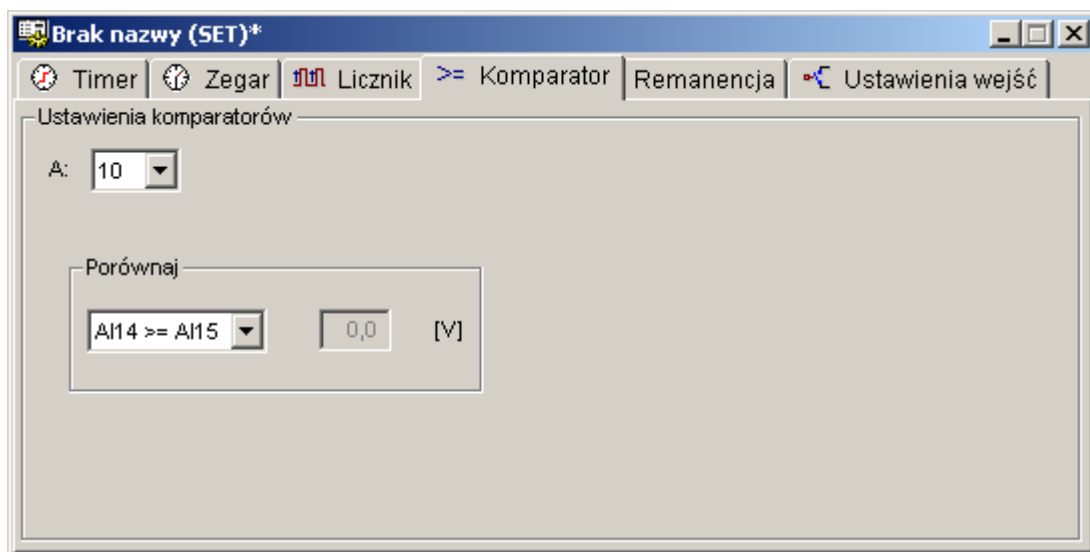
Aby więc dobrze ustawić Potencjometr dla przekaźników NEED-24DC-x1.., NEED-12DC-x1.. należy wskazać zmienną POT, w „Podglądzie zmiennej”, zawsze dzielić przez 10.

W tabeli 4.11.5. przedstawiono zakresy wejść analogowych w przekaźniku NEED.

Tab. 4.11.5. Zakresy wejść analogowych w przekaźniku NEED.

| Typ               | Rodzaj wejścia analogowego | Zakres     | Rozdzielczość |
|-------------------|----------------------------|------------|---------------|
| NEED-230AC-x1-..  | Napięciowe                 | 0 – 255V   | 1V            |
| NEED-12DC-x1-..   | Napięciowe                 | 0 – 25,5V  | 0,1V          |
| NEED-24DC-x1-..   | Napięciowe                 | 0 – 25,5V  | 0,1V          |
| NEED-12DC-x1-16-8 | Napięciowe                 | 0 – 12,75V | 0,05V         |
| NEED-24DC-x1-16-8 | Napięciowe                 | 0 – 12,75V | 0,05V         |
| NEED-12DC-x1-16-8 | Prądowe                    | 0 – 51mA   | 0,2mA         |
| NEED-24DC-x1-16-8 | Prądowe                    | 0 – 51mA   | 0,2mA         |
| NEED-12DC-x1-16-8 | Prądowe                    | 0 – 25,5mA | 0,1mA         |
| NEED-24DC-x1-16-8 | Prądowe                    | 0 – 25,5mA | 0,1mA         |

Przykład konfiguracji *komparatora A10* porównującego wartości sygnałów na wejściach analogowych I14, I15 przedstawiono na rys. 4.11.5.



Rys. 4.11.5. Przykład konfiguracji komparatora A10 do porównania dwóch wartości analogowych

Wyjście *Komparatora* A10 zostanie ustawione w stan '1', gdy wartość napięcia na wejściu analogowym I14 będzie większa lub równa wartości napięcia na wejściu analogowym I15.



Jeśli jakieś wejście analogowe przekaźnika NEED skonfigurowane jest jako prądowe, to dla porównań komparatora brane są tylko wartości **napięcia** według następującego wzoru:

$$\text{Wartość napięcia dla komparatora [V]} = 0,5 * \text{wartość prądu zmierzonego na wejściu [mA]}$$

Jest to liniowe skalowanie, gdzie 20mA odpowiada 10V.

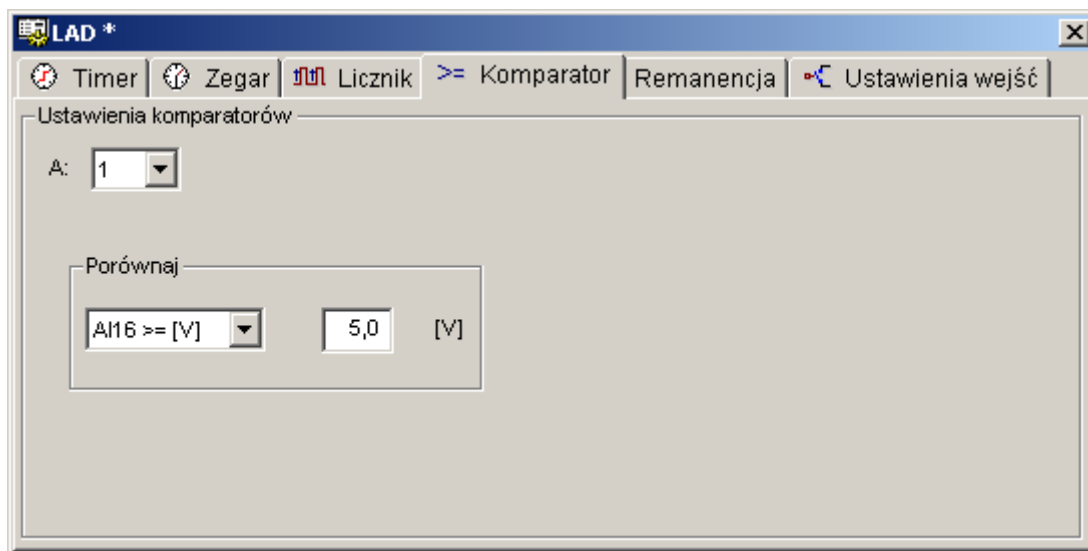
Przykład:

Do wejścia analogowego I16 podłączamy czujnik o wyjściu prądowym. Chcemy, aby mierzona wartość analogowa (np. ciśnienie), w „przeliczeniu” na prąd, nie przekroczyła 10mA.

Wtedy komparator A1 należy skonfigurować tak, jak na rys. 4.11.6. – zgodnie z przytoczoną powyżej formułą:

$$0,5 * 10\text{mA} = 5\text{V}$$

Do pola z lewej strony należy wpisać wartość 5V.



Rys. 4.11.6. Przykład konfiguracji komparatora A1 w programie PC Need, dla wejścia analogowego AI16 skonfigurowanego jako wejście prądowe.



Dla przekaźnika NEED-230AC-x1-.. wejścia analogowe są czytane co 4ms. Opóźnienie to nie zależy od ustawienia opóźnień dla wejść I7, I8 lub I14, I15, I16 przy konfiguracji w programie PC Need – patrz rozdział „8.4. Opóźnienie wejść”.

Dla przekaźników: NEED-12DC-x1-.. oraz NEED-24DC-x1-.. ustawienie opóźnienia wejść analogowych I7, I8 lub I14, I15, I16 spowoduje uśrednienie odczytywanych wartości mierzonych według poniższej reguły:

$$\text{Wartość bieżąca} = (\text{wartość poprzednia} + \text{wartość odczytana z wejścia analogowego}) / 2$$

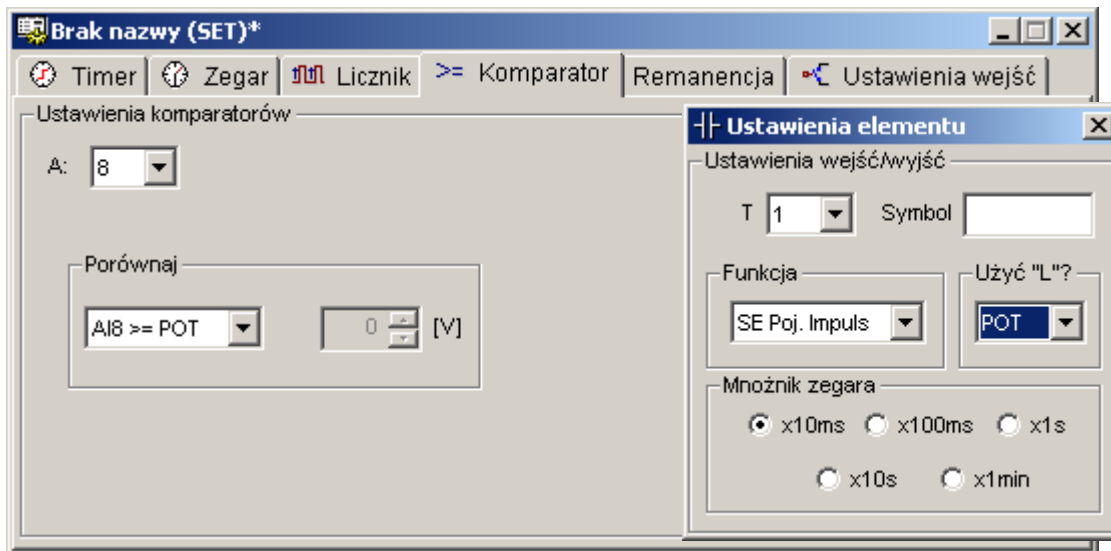
Wejścia analogowe, dla wykonań przekaźnika NEED-12DC-x1-.. oraz NEED-24DC-x1-.. czytane są co 4ms.

#### 4.12. Potencjometr

Potencjometr należy do zasobów typowo sprzętowych i można go używać do:

- ustawiania czasów dla *Timerów*,
- ustawiania wartości do zliczania dla *Liczników*,
- ustawiania progu przełączania *Komparatora*.

Pełny obrót Potencjometru odpowiada wartościom od 1 do 255. Do ustawiania odpowiednich wartości zadawanych z Potencjometru służy instrukcja „L” (STL – patrz rozdział 5.1.2.21., LAD – patrz rozdział 5.2.), w której można programowo modyfikować zakresy Potencjometru dla lepszego dopasowania się do oczekiwanej wartości mierzonej. Przykład wykorzystania Potencjometru dla NEED-230AC-01-.. przedstawiono na rys. 4.12.1.



Rys. 4.12.1. Przykład wykorzystania Potencjometru.

W powyższym przykładzie *Timer* T1 będzie odmierzał czas o długości równej: ustawienie Potencjometru x 10ms (np. 12\*10=120ms).

Natomiast wyjście *Komparatora* A8 będzie w stanie wysokim, gdy wartość napięcia na wejściu analogowym I8 będzie większa od wartości ustawionej za pomocą Potencjometru (1 – 255).

#### 4.13. Wartości remanentne przekaźnika programowalnego

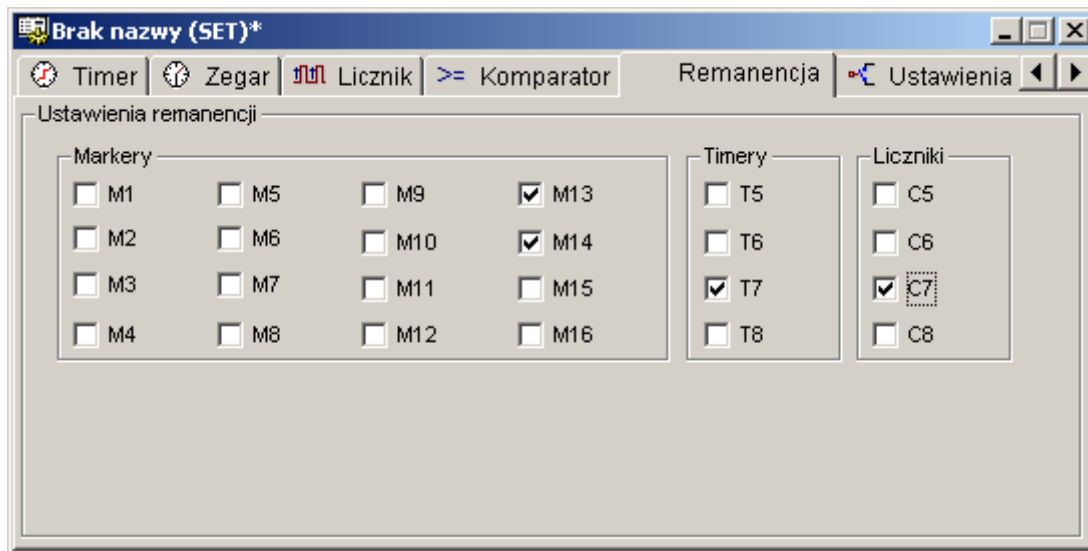
W procesach sterowania często zdarza się, iż potrzebne jest zachowanie danych po wyłączeniu zasilania. W przekaźniku programowalnym NEED można ustalić pewne „obszary” zasobów przekaźnika – zasoby remanentne, które mogą być podtrzymywane przy wyłączonym napięciu zasilającym lub po przełączeniu przekaźnika w tryb STOP. W tabeli 4.13.1. przedstawiono zasoby, które mogą być zdefiniowane jako remanentne.

Tab. 4.13.1. Zasoby remanentne w przekaźniku programowalnym NEED.

| Zasoby remanentne | Zakres   |
|-------------------|----------|
| Znaczniki         | M1 – M16 |
| Timery            | T5 – T8  |
| Liczniki          | C5 – C8  |

Aby zdefiniować zasoby przekaźnika programowalnego jako remanentne należy w programie konfiguracyjnym PC Need zaznaczyć odpowiednie pola. Przykładowa konfiguracja wartości remanentnych przekaźnika została przedstawiona na rys. 4.13.2.





Rys. 4.13.2. Przykładowa konfiguracja zasobów remanentnych.

W powyższym przykładzie *Znaczniki* M13, M14, *Timer* 7 oraz *Licznik* 7 zostały skonfigurowane jako remanentne.

Ustalenie tej konfiguracji powinno się odbyć w trybie pracy STOP przekaźnika. Fabrycznie lub po operacji *RESET* zasoby remanentne są nieustawione.



Ustawienie remanencji może być przyczyną nieoczekiwanego działania programu ze względu na nieokreślone warunki początkowe. Przeładowanie programu powoduje wyzerowanie wartości remanentnych.

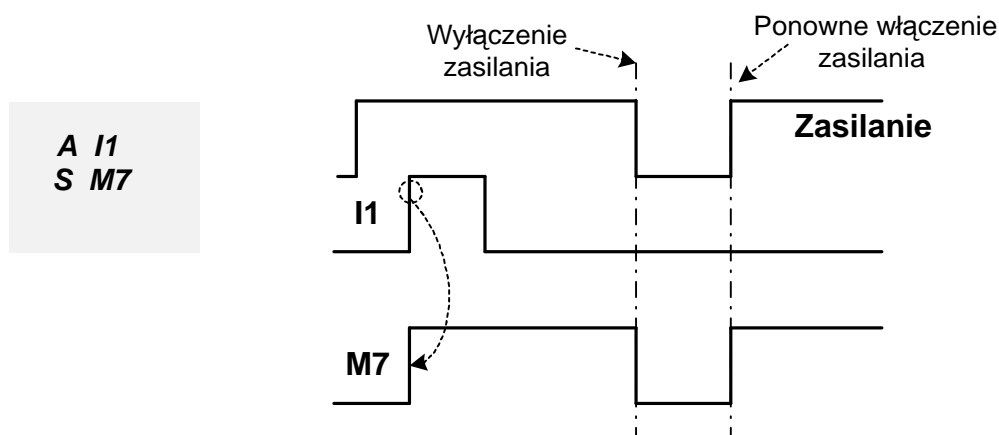
#### 4.13.1. Uwagi dotyczące wartości remanentnych

##### 1. Znaczniki.

Jeśli ustawimy dany *Znacznik* jako remanentny, to po wyłączeniu i ponownym załączeniu zasilania lub przejściu przekaźnika RUN → STOP → RUN, będzie on pamiętał swój stan logiczny sprzed wyłączenia.

Przykład:

Ustawmy *Znacznik* M7 jako remanentny.



Rys. 4.13.1.1. Działanie remanencji dla Znacznika M7.

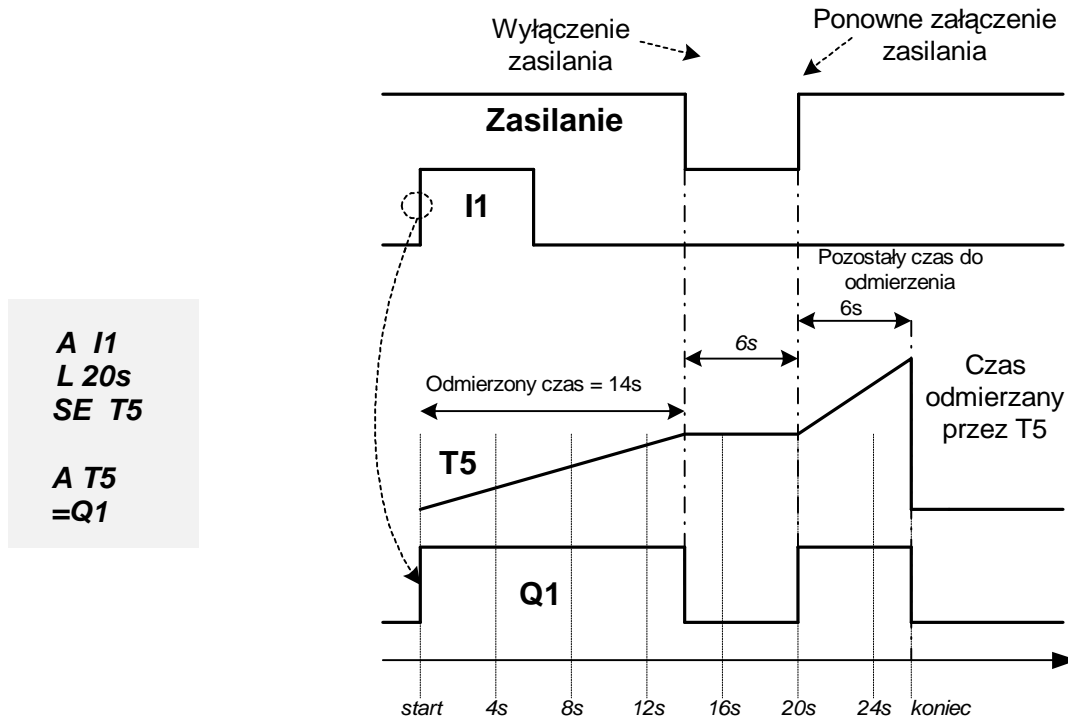
Po wyłączeniu i ponownym załączeniu zasilania, M7 ma stan wysoki, mimo, że I1='0'.

## 2. Timery.

Jeśli ustawimy dany *Timer* jako remanentny, to po wyłączeniu i ponownym załączeniu zasilania lub przejściu przekaźnika RUN → STOP → RUN, będzie on pamiętał swój stan logiczny oraz wartość czasu sprzed wyłączenia.

Przykład:

Ustawmy *Timer* T5 jako remanentny.



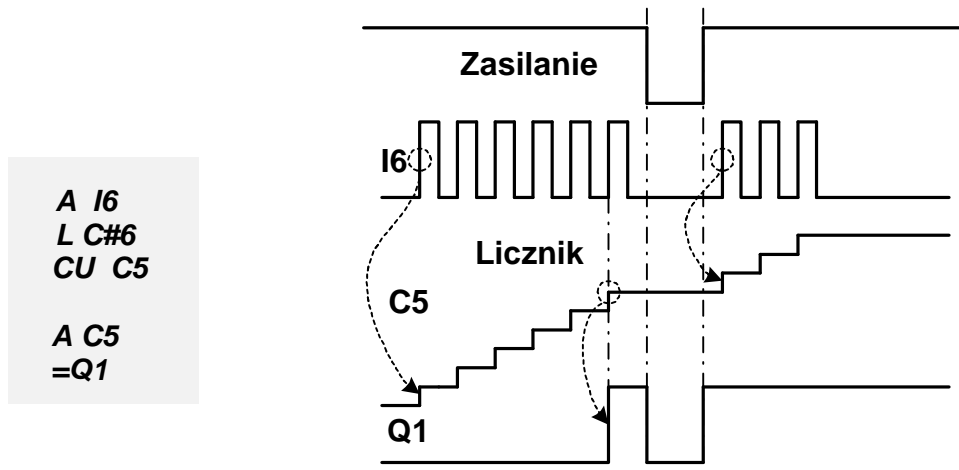
Rys.4.13.1.2. Działanie remanencji dla Timera T5.

Po wyzwoleniu *Timer* T5 odmierza czas. Po wyłączeniu zasilania, w 14s odmierzony czas jest pamiętany, a po ponownym załączeniu zasilania, *Timer* kończy odmierzenie 20s – ustawia swoje wyjście w stan wysoki na pozostały czas 6s.

## 3. Liczniki.

Jeśli ustawimy dany *Licznik* jako remanentny, to po, np. wyłączeniu i ponownym załączeniu zasilania, będzie on pamiętał swój stan logiczny oraz ilość zliczonych impulsów przed wyłączeniem.

Przykład:  
Ustawmy Licznik C5 jako remanentny.



Rys. 4.13.1.3. Działanie remanencji dla Licznika C5.

Impulsy, które pojawiają się na I6 zwiększają wartość Licznika C5. Po wyłączeniu i ponownym załączeniu zasilania, Licznik pamięta swoją wartość oraz stan wyjściowy sprzed wyłączenia. Gdy pojawią się następne impulsy wyzwalające, Licznik nie liczy od zera, ale od wartości zapamiętanej sprzed wyłączenia zasilania.



Licznik remanentny zliczy impuls, jeżeli stan na wejściu przed wyłączeniem zasilania był '0', a po załączeniu '1'. Licznik nieremanentny nie zliczy takiego impulsu.

Jeżeli przed wyłączeniem zasilania na wejściu Licznika był stan '1' i po załączeniu zasilania nadal utrzymywał się stan wysoki, to Licznik remanentny nie zliczy impulsu.

## 5. OPIS JĘZYKÓW PROGRAMOWANIA

Przełącznik NEED możemy programować używając dwóch języków programowania. Zdefiniowane one zostały w taki sposób, aby programowanie przełącznika było jak najbardziej efektywne oraz, aby użytkownik miał możliwość wyboru wygodnego dla siebie języka programowania. I tak do opisywania zadań sterowania możemy korzystać z:

- języka tekstowego – Lista Instrukcji (STL)
- języka graficznego – Schemat drabinkowy (LAD)

### 5.1. Programowanie w języku tekstowym STL

Język tekstowy STL (ang. *Statement List*) jest zbiorem instrukcji obejmujących operacje logiczne, relacje, jak również funkcje przerzutników, *Timerów*, *Liczników* itp., które umożliwiają odpowiednie zaprogramowanie przełącznika. Używanie języka tekstowego do programowania przełącznika NEED jest bardzo efektywne i najbardziej przybliża kod wynikowy do struktury wewnętrznej programu.

#### 5.1.1. Struktura programu STL

Program w STL jest ciągiem kolejno wykonywanych instrukcji.

Każda instrukcja składa się z dwóch elementów:

- 1) Symbolu Instrukcji – identyfikatora (kodu), który w języku STL pełni rolę tzw. słowa kluczowego.
- 2) Argumentu – czyli zmiennej (wyjątek stanowi instrukcja SET i CLR).

| <b>&lt;kod&gt;</b>   | <b>&lt;argument&gt;</b>  |
|--|--|
| A, A(, AN, AN(, O,O(, ON, ON(,<br>X, X(, XN, XN(<br>S, R, =, FP<br>SD, SF, SE, SL<br>CU, CD<br>L, SET, CLR | I,Q,M, MDIR, H, A,<br>T, C, HC1, H<br>L–Licznik <sup>1)</sup><br>L–Timer <sup>2)</sup> |

1- L–Licznik – jest liczbą zadanych zliczeń dla Licznika.

2- L–Timer – jest zadany czasem do odmierzenia dla Timera.

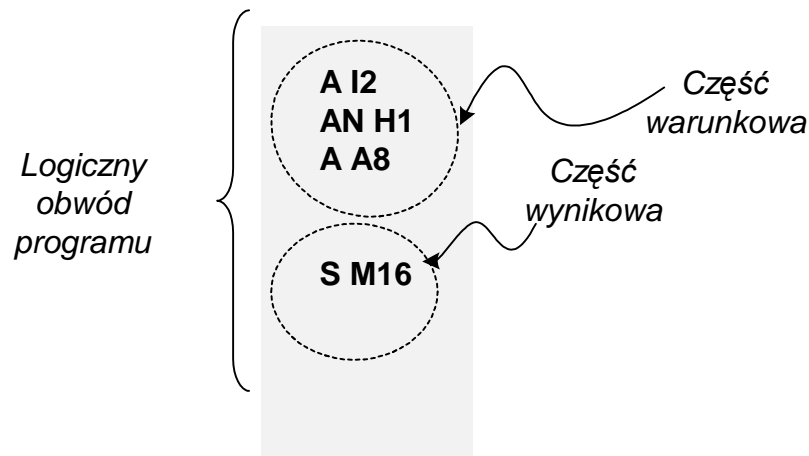
W zapisie logicznym pewnych sekwencji, z których składa się program możemy wyróżnić część warunkową (poprzedzającą) i część wynikową (następującą). Innymi słowy: jeśli spełnione są pewne warunki zapisane za pomocą określonych instrukcji i zmiennych, to ta sytuacja wywoła pewien skutek, który także jest określony za pomocą odpowiednich instrukcji i zmiennych. Taki zestaw części warunkowej i wynikowej nazywamy obwodem. Każda linia programu powinna zaczynać się od dozwolonej instrukcji, po której powinien występować argument. Dozwolony więc jest zapis typu:

```
A I1
A I2
S Q5
```

Zabroniony natomiast jest zapis typu:

```
I1
A I2
S Q5
```

W pierwszej linii przykładu powyżej brak jest nazwy instrukcji - może nią być instrukcja A, A(, AN, AN(, O, O(, ON, ON(, X, X(, XN, XN(.



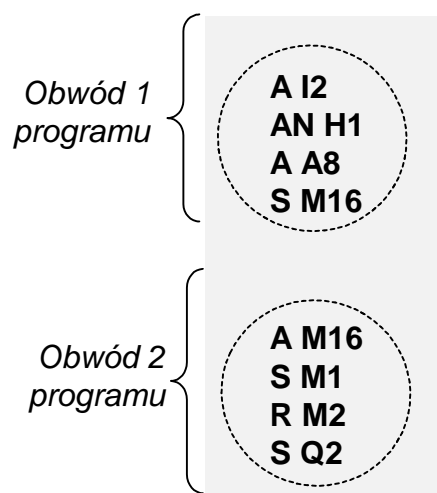
Rys. 5.1.1.1. Struktura programu STL.

Przykład:

|    |      |   |   |
|----|------|---|---|
| 1) | A I1 | } | Część warunkowa (sprawdza relacje: I1 AND A2) |
| 2) | A A2 |   |   |
| 3) | S Q4 | } | Część wynikowa                                |
| 4) | R Q1 |   |   |

Część warunkową w tym przykładzie będą stanowiły instrukcje zapisane w liniach 1,2. Jeśli na wejściu I1 i wyjściu *Komparatora* A2 będzie stan wysoki, to wyjście Q4 zostanie ustawione (stan '1'), a wyjście Q1 zostanie zresetowane (stan '0'). Instrukcje 'S Q4' oraz 'R Q1' stanowią część wynikową.

Instrukcje: A, A(, AN, AN(, O, O(, ON, ON(, X, X(, XN, XN( składają się na część warunkową obwodu a instrukcje S, R, =, FP, SD, SF, SL, SE, CD, CU tworzą część wynikową obwodu. Każdy oddzielny obwód powinien zaczynać się częścią warunkową a kończyć częścią wynikową.

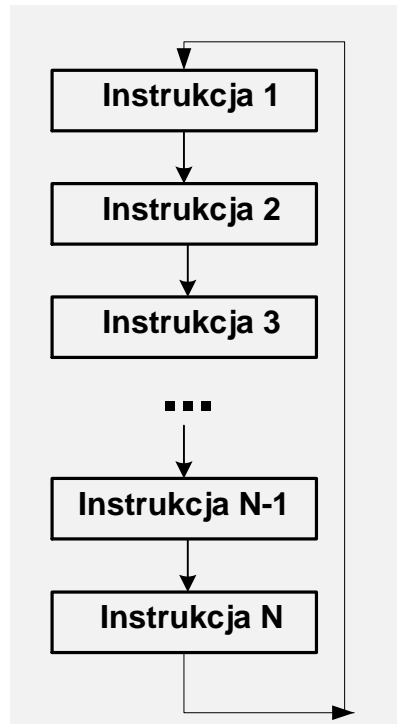


Rys.5.1.1.2. Przykład dwóch obwodów w STL.

W przekaźniku programowalnym znajduje się tylko jeden program, którego nie można dzielić na żadne wywoływane podprogramy.

Procesor w sterowniku realizuje poszczególne instrukcje kolejno, poczynając od pierwszej a kończąc na ostatniej. Po wykonaniu ostatniej instrukcji cykl programu powtarza się.

Przetwarzanie programu sterownika przedstawiono na rys. 5.1.1.3.



Rys. 5.1.1.3. Cykliczne przetwarzanie programu STL.

W tab. 5.1.1. przedstawiono wykaz wszystkich dostępnych instrukcji STL.

Tab. 5.1.1. Instrukcje STL.

| STL Instrukcja | Opis                          | Operandy                |
|----------------|-------------------------------|-------------------------|
| A              | Instrukcja AND                | I,Q,M,MDIR,A,H,HC1,C,T  |
| A(             | Instrukcja nawiasów AND       |                         |
| AN             | Instrukcja AND NOT            | I,Q,M,MDIR,A,H,HC1,C,T  |
| AN(            | Instrukcja nawiasów AND NOT   |                         |
| O              | Instrukcja OR                 | I,Q,M,MDIR,A,H,HC1, C,T |
| O(             | Instrukcja nawiasów OR        |                         |
| ON             | Instrukcja OR NOT             | I,Q,M,MDIR,A,H,HC1,C,T  |
| ON(            | Instrukcja nawiasów OR NOT    |                         |
| X              | Instrukcja XOR                | I,Q,M,MDIR,A,H,HC1,C,T  |
| X(             | Instrukcja nawiasów XOR       |                         |
| XN             | Instrukcja XOR NOT            | I,Q,M,MDIR,A,H,HC1,C,T  |
| XN(            | Instrukcja nawiasów XOR NOT   |                         |
| S              | Instrukcja ustawiająca        | Q,M                     |
| R              | Instrukcja resetująca         | Q,M,T,C,HC1             |
| =              | Instrukcja przyporządkowująca | Q,M                     |
| FP             | Przekaźnik impulsowy          | Q,M                     |
| L              | Instrukcja ładująca           | Stała wartość operandu  |

Tab. 5.1.1. Instrukcje STL- c.d.

| STL Instrukcja | Opis                          | Operandy |
|----------------|-------------------------------|----------|
| SD             | Timer – Opóźnione załączenie  | T        |
| SE             | Timer – Opóźnione wyłączenie  | T        |
| SF             | Timer – Pojedynczy impuls     | T        |
| SL             | Timer – Impulsy               | T        |
| CU             | Licznik – Zliczanie w górę    | C, HC1   |
| CD             | Licznik – Zliczanie w dół     | C, HC1   |
| SET            | Instrukcja zawsze ustawiająca |          |
| CLR            | Instrukcja zawsze kasująca    |          |

#### 5.1.1.1. Nazwy symboliczne

Dla przekaźników NEED istnieje możliwość przypisania w projekcie nazw symbolicznych do zmiennych. Ułatwia to analizę programu i czyni go bardziej przejrzystym.

Aby skojarzyć zmienną z nazwą symboliczną należy użyć wyrażenia o następującej składni:

**. DEFINE < nazwa symboliczna > = < zmienna >**

Potem można używać zamiast adresu zmiennej np. Q1, I11 nazwy symbolicznej, która musi być poprzedzona znakiem „%” np.:

```
.DEFINE Pompa = Q1
.DEFINE Awaria = I11
```

```
A %Awaria
R %Pompa
```

W nazwach symbolicznych nie są rozróżniane małe i duże litery.

Nazwami symbolicznymi nie mogą być także nazwy elementów zasobów przekaźnika oraz instrukcji. Nazwy symboliczne muszą zaczynać się od litery lub znaku podkreślenia ( ), po którym następuje ciąg liter lub cyfr. Nie można używać w nazwie znaku spacji.

Nazwy symboliczne nie mogą zaczynać się cyfrą. Mogą zawierać maksymalnie 30 znaków.

## 5.1.2. Opis Instrukcji STL

## 5.1.2.1. Instrukcja AND

SYMBOL – **A**

Instrukcja 'A' jest logiczną instrukcją typu AND.

SKŁADNIA:

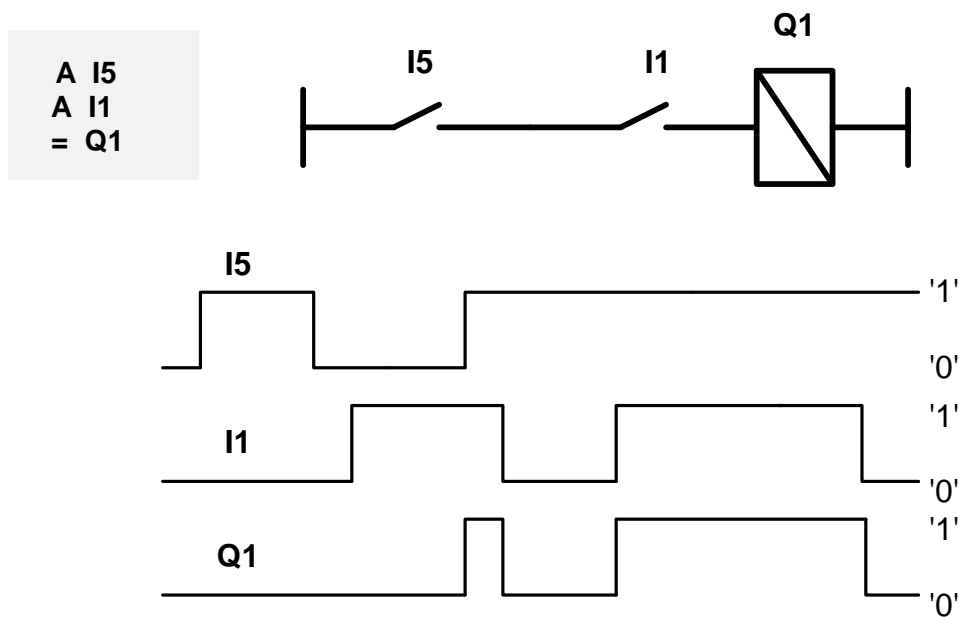
**A < I, Q, M, MDIR, A, H, C, HC1, T >**

Czas wykonania instrukcji: 6µs.

Przykład:

STL

Schemat przekaźnikowy



Rys. 5.1.2.1. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Powyższy przykład realizuje połączenie szeregowe.

wyjście Q1 zostanie ustawione (stan '1'), gdy oba wejścia będą miały stan wysoki – zgodnie z zasadami działania funkcji AND.

## 5.1.2.2. Instrukcja nawiasów AND

SYMBOL – **A(**

Instrukcja 'A(' jest logiczną instrukcją typu AND, której operandem jest wynik operacji logicznych w nawiasie.

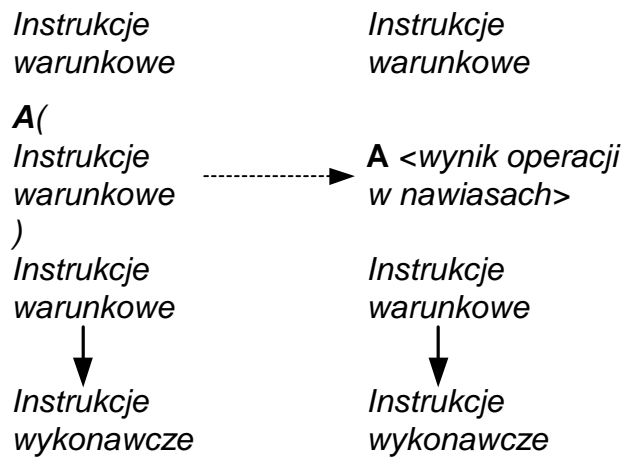
SKŁADNIA:

**A(****Instrukcje warunkowe****)**

Czas wykonania instrukcji: 6µs.



Na rys. 5.1.2.2.1. przedstawiono zasadę działania instrukcji 'A(' – wszystkie inne instrukcje nawiasów działają na tej samej zasadzie.



Rys. 5.1.2.2.1. Zasada działania instrukcji nawiasów 'A('.

Wykonujemy operacje w nawiasach. W wyniku tych operacji logicznych otrzymujemy jakiś rezultat ('0' lub '1'), którego używamy do następnych operacji logicznych np. dla programu:

```
A I1
A(
O M1
O M2
)
=Q1 //Jest to równoważne operacji logicznej I1 AND (M1 OR M2) = Q1
```

oraz stanów logicznych: M1='0', M2='0', I1='1'.

Można zapisać:

```
A I1      A '1'
A(        A '0' //bo '0' O '0' = '0'
O M1
O M2
)
=Q1      = '0'
```

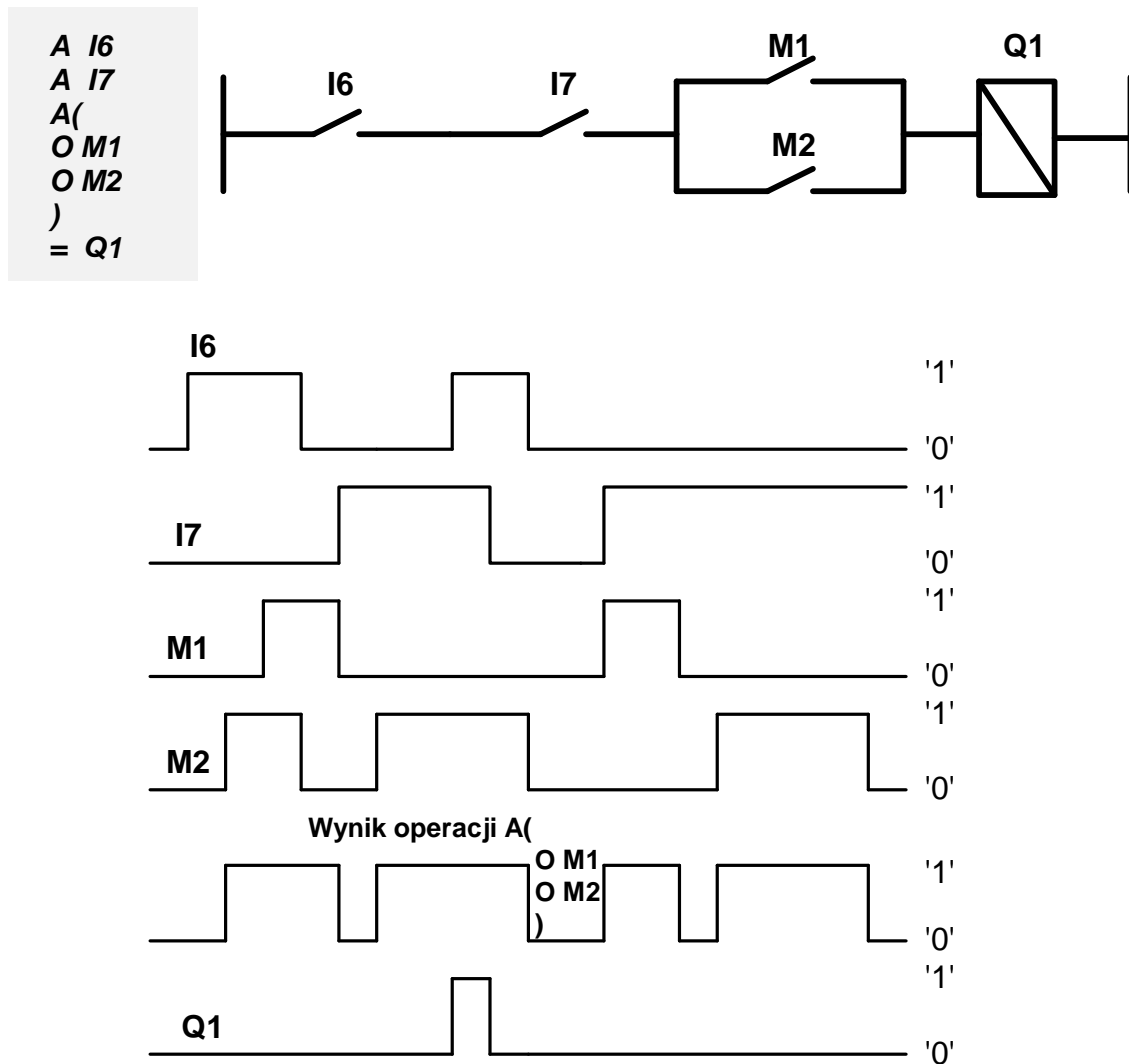
Czyli dla analizowanych stanów wyjście Q1 będzie w stanie '0'. Natomiast dla stanów M1='1', M2='0', I1='1' otrzymujemy:

```
A I1      A '1'
A(        A '1' //bo '1' O '0' = '1'
O M1
O M2
)
=Q1      = '1'
```

Przykład:

STL

Schemat przekaźnikowy



Rys. 5.1.2.2.2. Przykładowe przebiegi czasowe na wejściach I6, I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy wejścia I6, I7 będą miały stan wysoki oraz gdy przynajmniej jeden ze Znaczników M1 lub M2, będzie w stanie '1'.

### 5.1.2.3. Instrukcja AND NOT

SYMBOL – **AN**

Instrukcja 'AN' jest logiczną instrukcją typu AND NOT (instrukcja AND z zanegowanym stanem operandu).

SKŁADNIA:

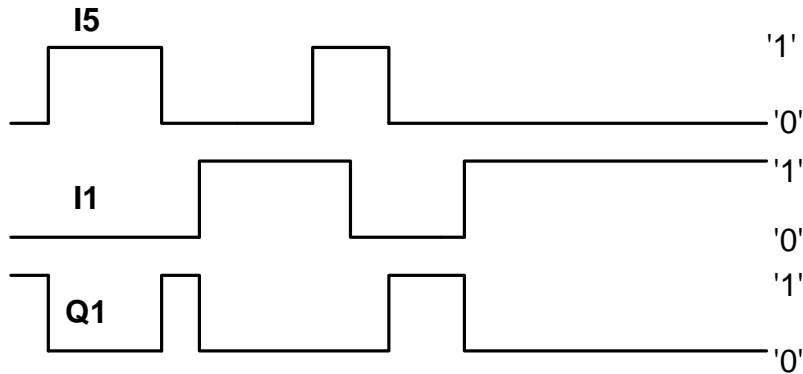
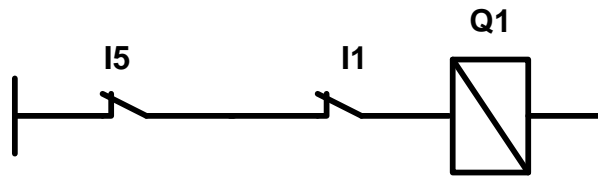
**AN** < I, Q, M, MDIR, A, H, HC1, C, T >

Czas wykonania instrukcji: 6µs.

Przykład:  
STL

```
AN I5
AN I1
= Q1
```

Schemat przekaźnikowy



Rys. 5.1.2.3. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy oba wejścia będą miały stan niski ('0').

#### 5.1.2.4. Instrukcja nawiasów *AND NOT*

SYMBOL – **AN(**

Instrukcja 'AN(' jest logiczną instrukcją typu *AND NOT*, której operandem jest wynik operacji logicznych w nawiasie.

SKŁADNIA:

**AN(**

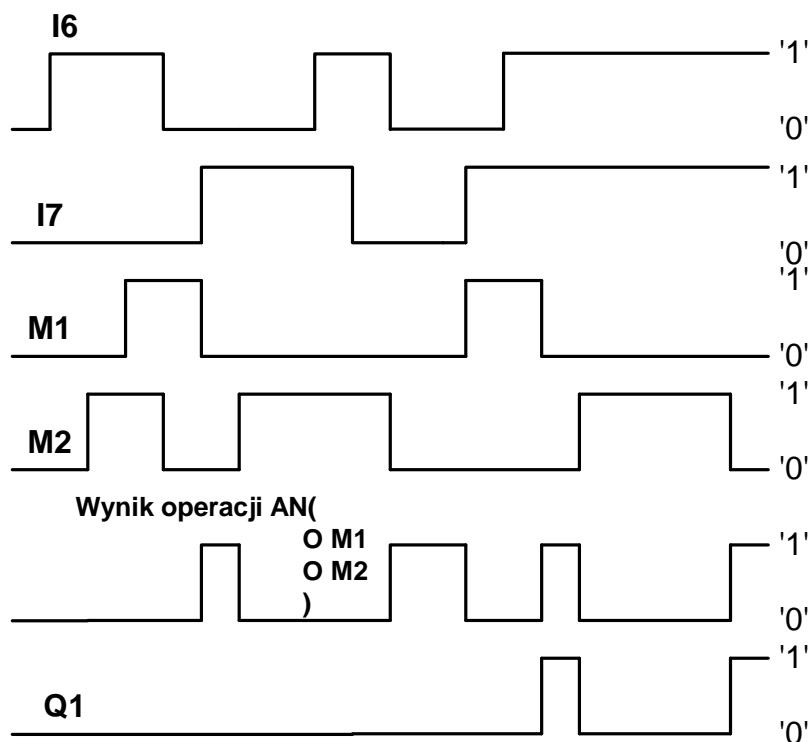
**Instrukcje warunkowe**

**)**

Czas wykonania instrukcji: 6µs.

Przykład:  
STL

```
A I6
A I7
AN(
O M1
O M2
)
= Q1
```



Rys. 5.1.2.4. Przykładowe przebiegi czasowe na wejściach I6, I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy wejścia I6, I7 będą miały stan wysoki oraz gdy oba Znaczniki M1 i M2 będą w stanie '0'. Jest to równoważne operacji logicznej  $I6 \text{ AND } I7 \text{ AND } (M1 \text{ OR } M2) = Q1$

#### 5.1.2.5. Instrukcja OR

SYMBOL – **O**

Instrukcja 'O' jest logiczną instrukcją typu OR

SKŁADNIA:

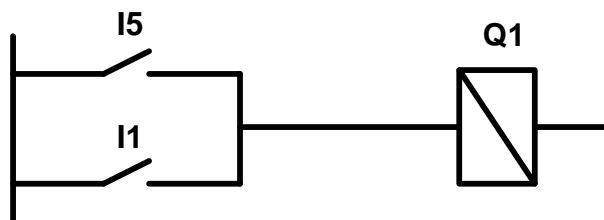
**O < I, Q, M, MDIR, A, H, HC1, C, T >**

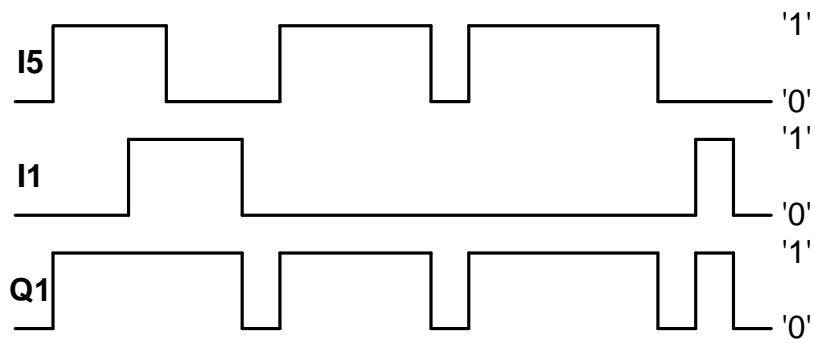
Czas wykonania instrukcji: 6µs.

Przykład:  
STL

```
O I5
O I1
= Q1
```

Schemat przekaźnikowy





Rys. 5.1.2.5. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy jedno z wejść będzie miało stan wysoki ('1'). Realizacja połączenia równoległego.

#### 5.1.2.6. Instrukcja nawiasów OR

SYMBOL – **O(**

Instrukcja 'O(' jest logiczną instrukcją typu OR, której operandem jest wynik operacji logicznych w nawiasie.

SKŁADNIA:

**O(**

**Instrukcje warunkowe**

**)**

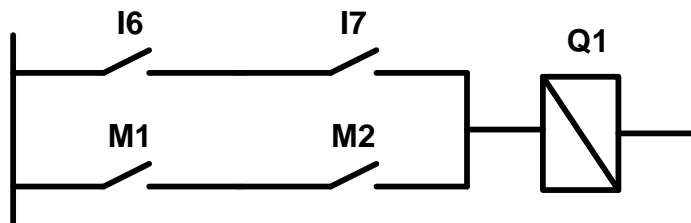
Czas wykonania instrukcji: 6µs.

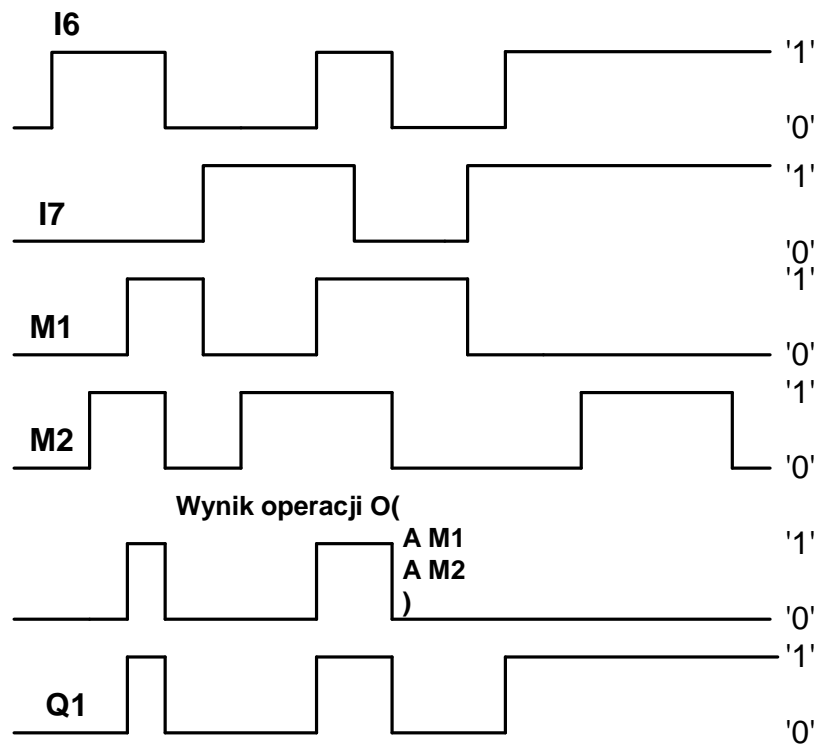
Przykład:

STL

Schemat przekaźnikowy

```
A I6
A I7
O(
A M1
A M2
)
= Q1
```





Rys. 5.1.2.6. Przykładowe przebiegi czasowe na I6, I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy wejścia I6, I7 będą miały stan wysoki lub, gdy oba Znaczniki M1 i M2 będą w stanie '1'. Jest to równoważne z

#### 5.1.2.7. Instrukcja OR NOT

SYMBOL – **ON**

Instrukcja 'O' jest logiczną instrukcją typu OR NOT (instrukcja OR z zanegowanym stanem operandu).

SKŁADNIA:

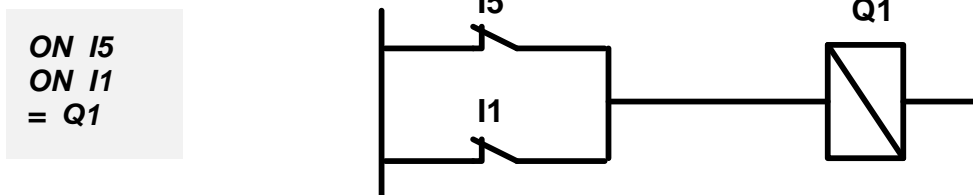
**ON < I, Q, M, MDIR, A, H, HC1, C, T >**

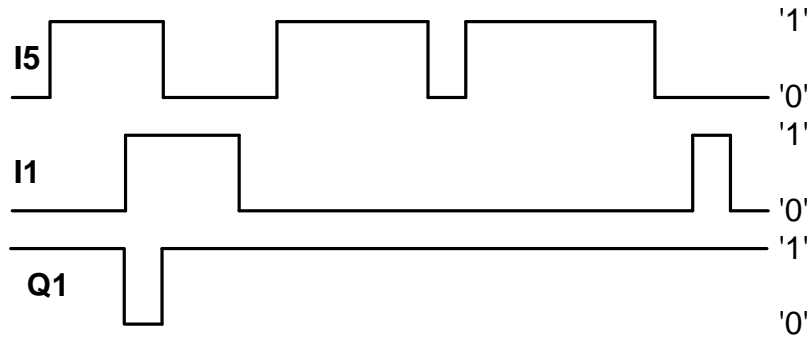
Czas wykonania instrukcji: 6µs.

Przykład:

STL

Schemat przekaźnikowy





Rys. 5.1.2.7. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy przynajmniej jedno z wejść będzie miało stan niski ('0').

#### 5.1.2.8. Instrukcja nawiasów *OR NOT*

SYMBOL – **ON(**

Instrukcja 'ON(' jest logiczną instrukcją typu *OR NOT* wyniku operacji logicznych w nawiasie.

SKŁADNIA:

**ON(**

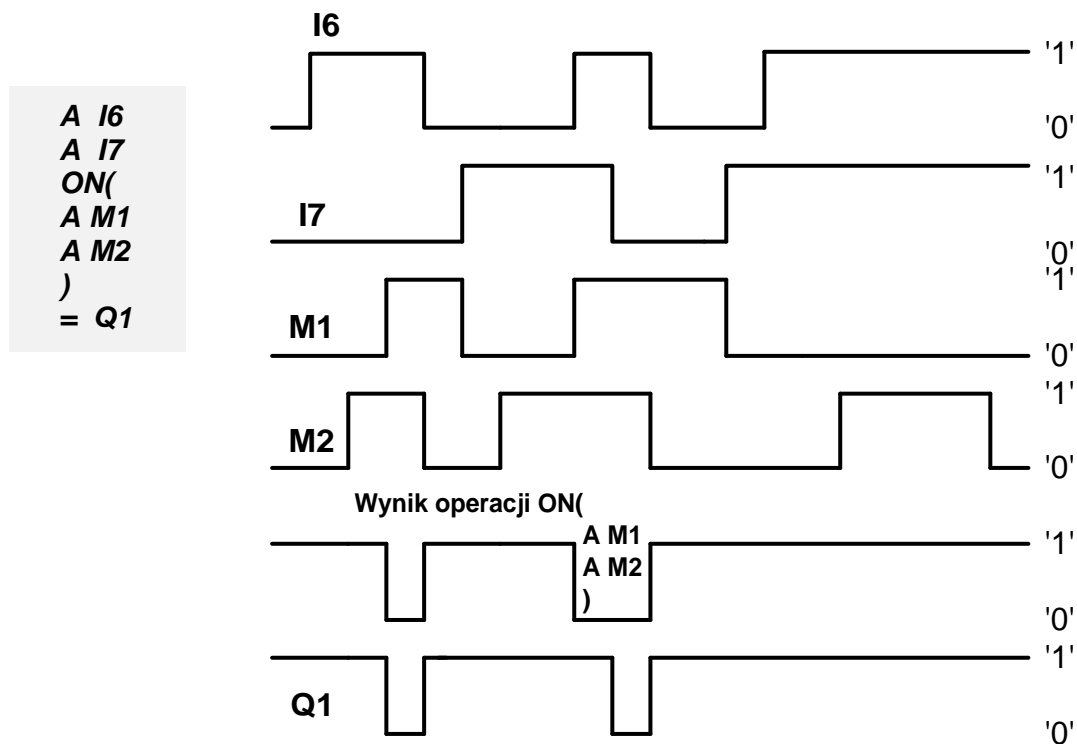
**Instrukcje warunkowe**

**)**

Czas wykonania instrukcji: 6µs.

Przykład:

STL



Rys. 5.1.2.8. Przykładowe przebiegi czasowe na I6, I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy wejścia I6, I7 będą miały stan wysoki lub, gdy

jeden ze Znaczników M1 lub M2 będzie w stanie '0'.

#### 5.1.2.9. Instrukcja XOR

SYMBOL – X

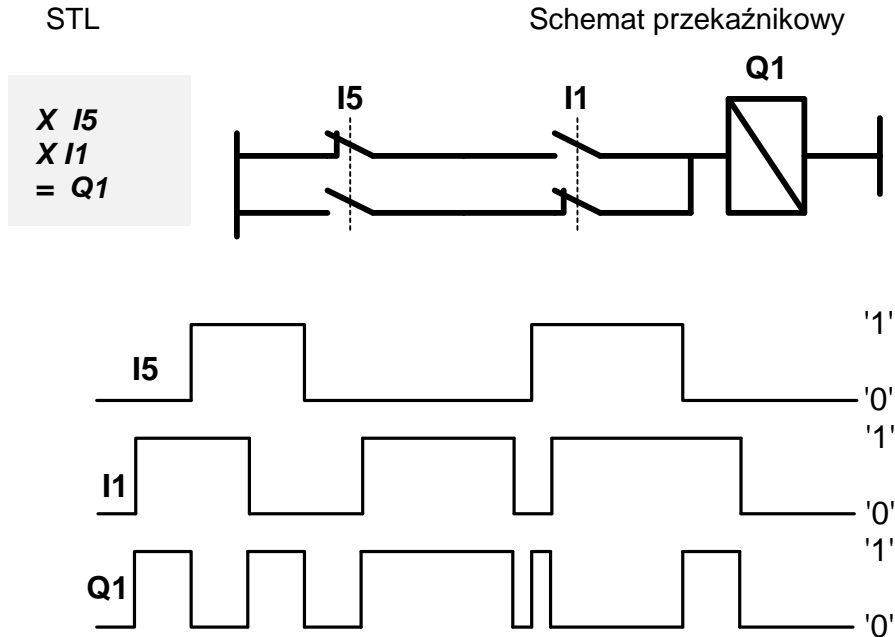
Instrukcja 'X' jest logiczną instrukcją typu XOR.

SKŁADNIA:

**X <I, Q, M, MDIR, A, H, HC1, C, T >**

Czas wykonania instrukcji: 6µs.

Przykład:



Rys. 5.1.2.9. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy na wejściach I5 oraz I1 będą panowały stany przeciwne (I5='1' i I1='0' lub I5='0' i I1='1').

#### 5.1.2.10. Instrukcja nawiasów XOR

SYMBOL – X(

Instrukcja 'X(' jest logiczną instrukcją typu XOR, której operandem jest wynik operacji logicznych w nawiasie.

SKŁADNIA:

**X(**

**Instrukcje warunkowe**

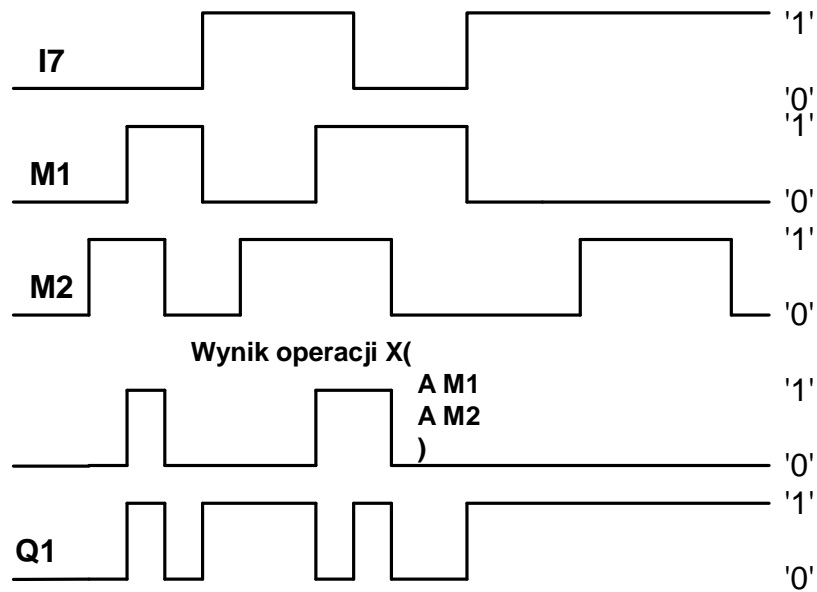
**)**

Czas wykonania instrukcji: 6µs.



Przykład:  
STL

```
A I7
X(
A M1
A M2
)
= Q1
```



Rys. 5.1.2.10. Przykładowe przebiegi czasowe na I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1') zgodnie z działaniem funkcji XOR tzn.:  
 $Q1=1$  dla  $I7=1$  i któryś ze Znaczników ustawiony jest w stanie '0'.  
 $Q1=1$  dla  $I7=0$  i oba Znaczniki są ustawione w stan wysoki ('1').

#### 5.1.2.11. Instrukcja XOR NOT

SYMBOL – **XN**

Instrukcja 'XN' jest logiczną instrukcją typu XOR NOT.

SKŁADNIA:

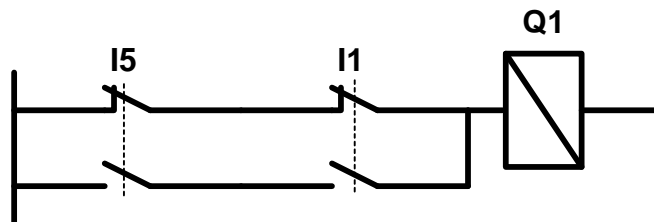
**X <I, Q, M, MDIR, A, H, HC1, C, T >**

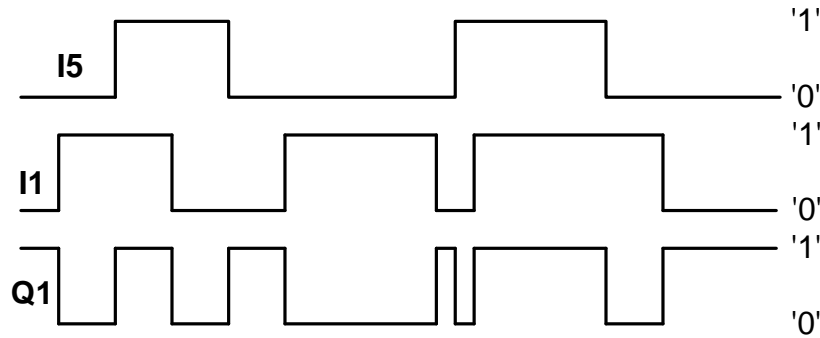
Czas wykonania instrukcji: 6µs.

Przykład:  
STL

Schemat przekaźnikowy

```
X I5
XN I1
= Q1
```





Rys. 5.1.2.11. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy na wejściach I5 oraz I1 będą panowały takie same stany logiczne (I5='0' i I1='0' lub I5='1' i I1='1').

#### 5.1.2.12. Instrukcja nawiasów *XOR NOT*

SYMBOL – ***XN(***

Instrukcja '*XN(*' jest logiczną instrukcją typu *XOR NOT* wyniku operacji logicznych w nawiasie.

SKŁADNIA:

***XN(***

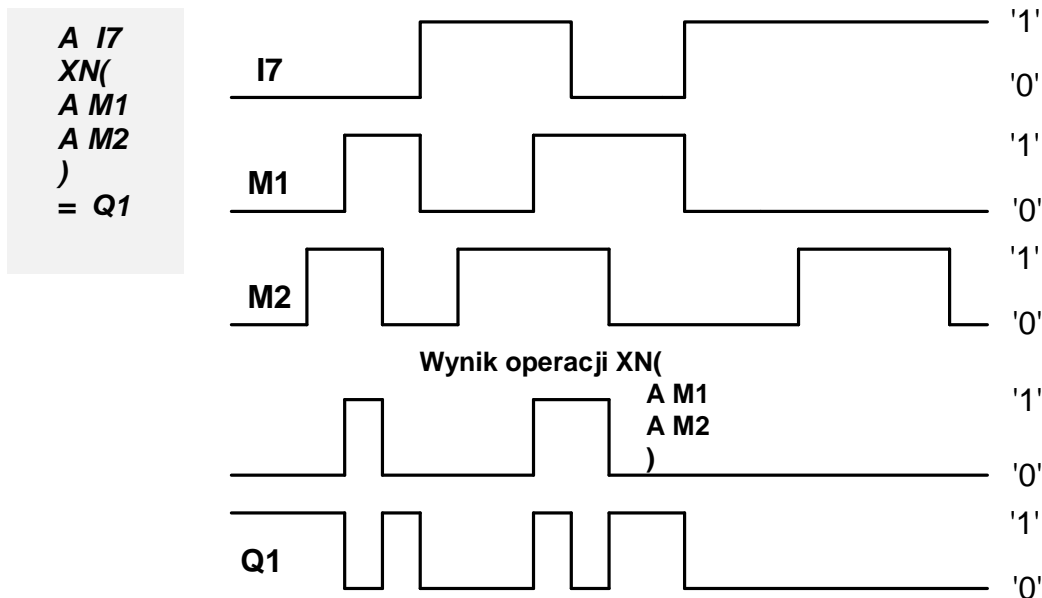
***Instrukcje warunkowe***

***)***

Czas wykonania instrukcji: 6μs.

Przykład:

STL



Rys. 5.1.2.12. Przykładowe przebiegi czasowe na I7, M1, M2 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1') zgodnie z działaniem funkcji *XOR NOT* tzn.:

Q1=1 dla I7=1 i oba Znaczniki M1, M2 są w stanie wysokim ('1').

Q1=1 dla I7=0 i któryś ze Znaczników jest w stanie niskim ('0').

## 5.1.2.13. Instrukcja ustawiająca S

SYMBOL – **S**

Instrukcja 'S' jest logiczną instrukcją ustawiającą argument w stan wysoki ('1').

SKŁADNIA:

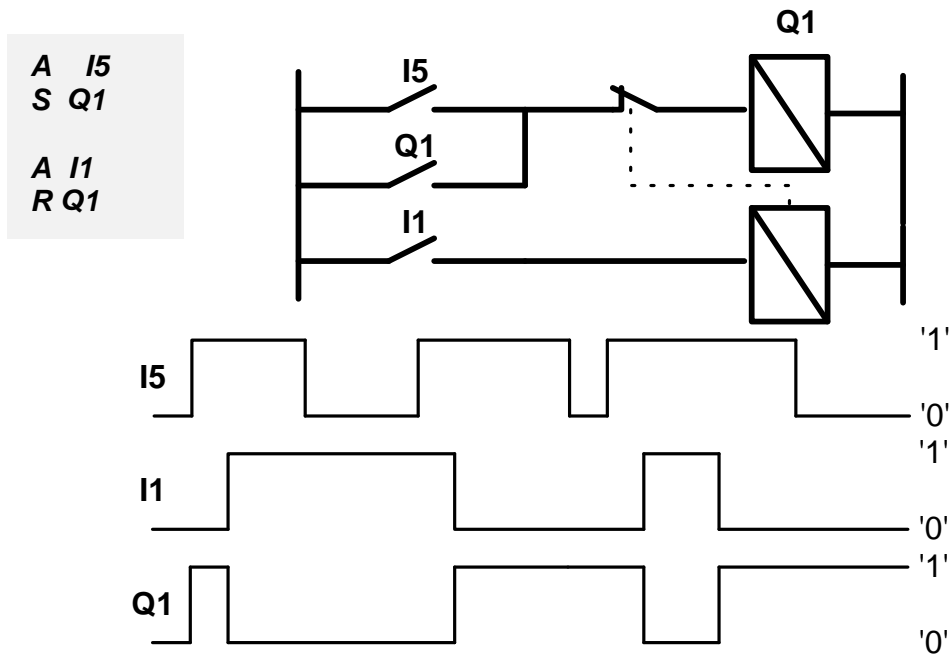
**S < Q, M >**

Czas wykonania instrukcji: 6,5µs.

Przykład:

STL

Schemat przekaźnikowy



Rys. 5.1.2.13. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Wyjście Q1 zostanie ustawione (stan '1'), gdy wejście I5 będzie miało stan wysoki ('1'). Będzie ono pozostawało w stanie wysokim tak długo, aż nie zostanie ustawiony stan niski ('0') instrukcją 'R' – wejście I1.

## 5.1.2.14. Instrukcja kasująca R (Reset)

SYMBOL – **R**

Instrukcja 'R' jest logiczną instrukcją ustawiającą argument w stan niski ('0').

SKŁADNIA:

**R < Q, M, T, C, HC1 >**

Czas wykonania instrukcji: 6,5µs.

Przykład: patrz instrukcja „S”.

## 5.1.2.15. Instrukcja przyporządkowująca =

SYMBOL – **=**

Instrukcja '=' jest logiczną instrukcją, w której argument przyjmuje wartość (stan '0' lub '1') zależną od wyniku wcześniejszych operacji logicznych.

SKŁADNIA:

**= < Q, M >**

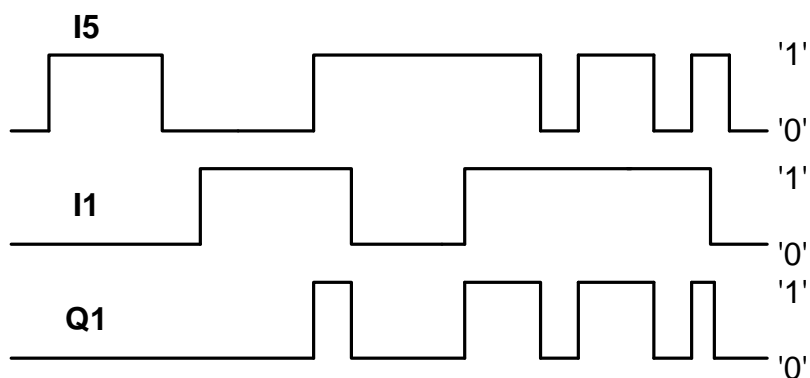
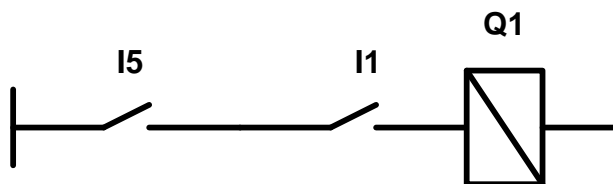
Czas wykonania instrukcji: 6,7µs.

Przykład:

STL

Schemat przekaźnikowy

```
A I5
A I1
= Q1
```



Rys. 5.1.2.15. Przykładowe przebiegi czasowe na wejściach I5, I1 oraz na wyjściu Q1.

Stan wyjścia Q1 zależy jest od wcześniejszych operacji logicznych tzn. przyjmuje stan '0', gdy stan któregoś z wejść jest '0' lub przyjmuje stan '1', gdy stany obu wejść są równe '1'.

#### 5.1.2.16. Instrukcja Przełącznik impulsowy FP

Przełącznik impulsowy pełni rolę przerzutnika wyzwalanego zboczem narastającym. Każdy narastający impuls zmienia stan wyjścia na przeciwny.

SYMBOL – **FP**

SKŁADNIA:

**FP <Q, M>**

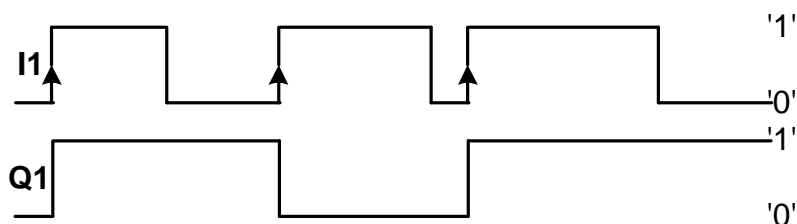
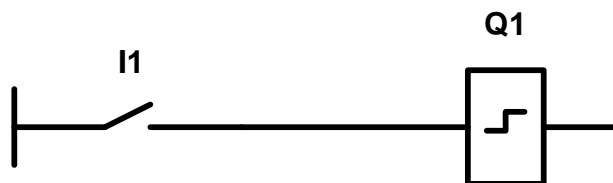
Czas wykonania instrukcji: 5,9µs.

Przykład:

STL

Schemat przekaźnikowy

```
A I1
FP Q1
```



Rys. 5.1.2.16. Przykładowe przebiegi czasowe na wejściu I1 oraz na wyjściu Q1.

Jeżeli wyjście Q1 pozostaje w stanie niskim i na wejściu I1 pojawi się dodatnie zbocze sterujące, to wyjście Q1 zostanie ustawione w stan wysoki.

Jeżeli wyjście Q1 pozostaje w stanie wysokim i na Wejściu I1 pojawi się dodatnie zbocze sterujący, to wyjście Q1 zostanie ustawione w stan niski.

### 5.1.2.17. Instrukcje Timerów

#### 5.1.2.17.1. Timer Opóźnione załączenie (ON-DELAYED)

Timer realizuje funkcję opóźnione załączenie.

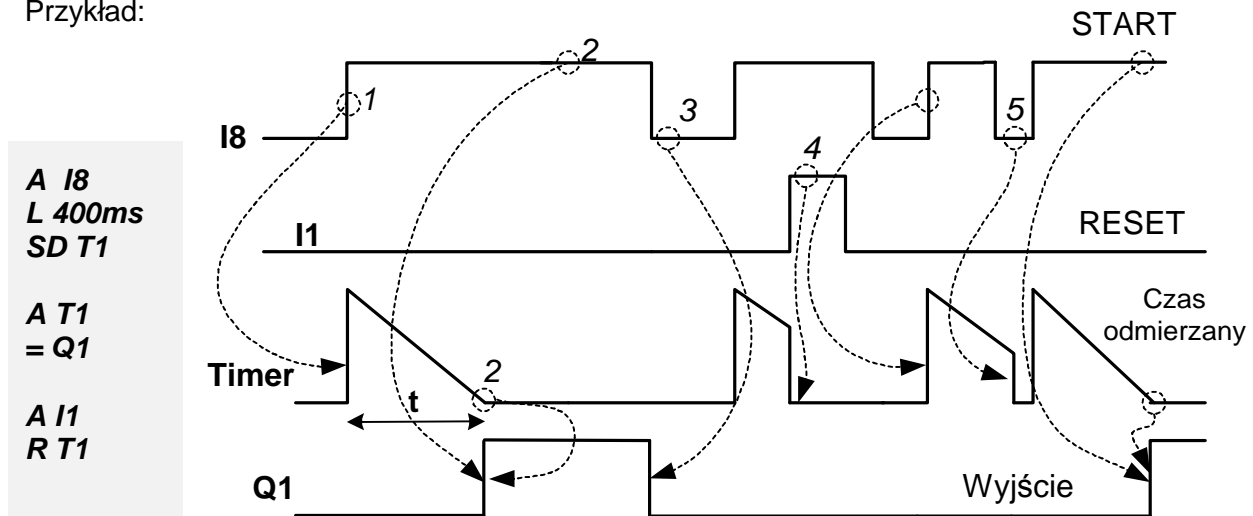
SYMBOL – **SD**

SKŁADNIA:

SD <T>

Czas wykonania instrukcji: 14,1µs.

Przykład:



Rys. 5.1.2.17.1. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Timera SD.

1.

Wejście I8 pełni rolę wejścia wyzwalającego (Triggera). Instrukcja ('L') ładująca określoną wartość czasu do odmierzenia, powinna się znaleźć bezpośrednio przed instrukcją Timera (SD).

Odmierzanie czasu następuje po wykonaniu instrukcji aktywacji Timera SD (zbocze narastające na wejściu I8).

2.

Po upływie czasu  $t=400\text{ms}$  następuje ustawienie wyjścia Q1 w stan wysoki ('1').

Jednocześnie na wejściu wyzwalającym I8 powinien utrzymywać się sygnał wysoki ('1').

3.

Jeśli na wejściu Triggera I8 pojawi się stan niski, następuje skasowanie licznika odmierzanego czasu Timera T1, a wyjście Q1 zostanie ustawione w stan niski ('0').

4.

Jeśli na wejściu I1 resetującym T1 pojawi się stan wysoki następuje automatyczne skasowanie licznika odmierzanego czasu Timera T1, a wyjście Q1 zostanie ustawione w stan niski ('0').

Jeżeli instrukcja „L” nie zostanie użyta, to czas do odmierzenia przez T1 zostanie ustalony na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).

## 5.1.2.17.2. Timer Opóźnione wyłączenie (OFF-DELAYED)

Timer realizuje funkcję opóźnione wyłączenie.

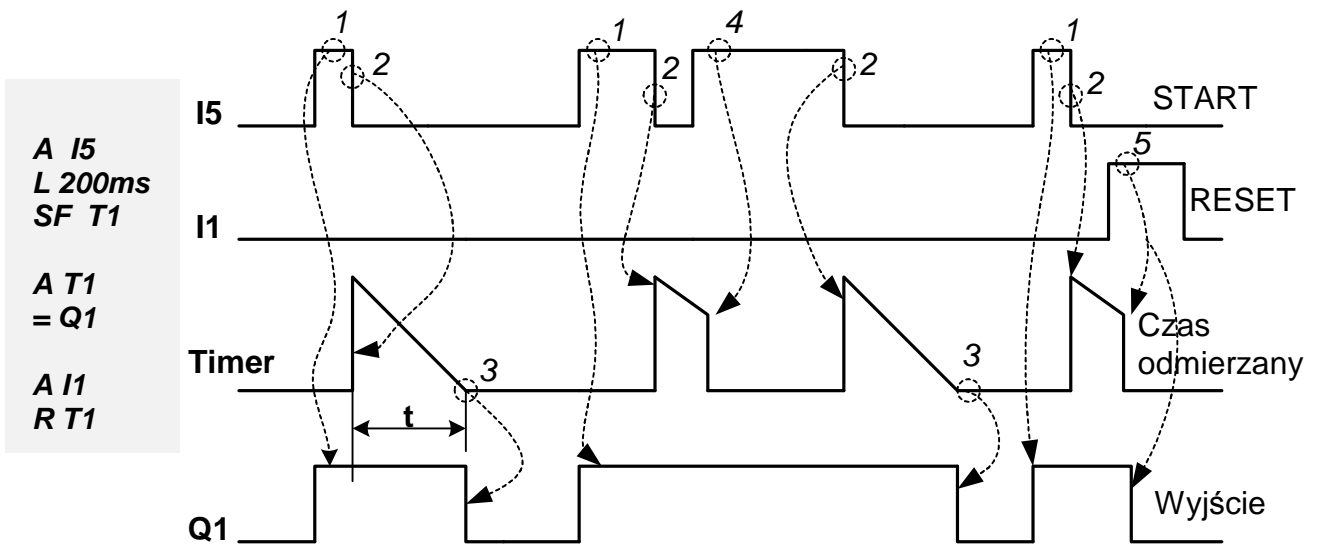
SYMBOL – **SF**

SKŁADNIA:

**SF <T>**

Czas wykonania instrukcji: 18,7µs.

Przykład:



Rys. 5.1.2.17.2. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Timera SF.

1. Wejście I5 pełni rolę wejścia wyzwalającego (Triggera). Instrukcja ('L') ładująca określoną wartość czasu do odmierzenia, powinna się znaleźć bezpośrednio przed instrukcją Timera (SF). Ustawienie wejścia I5 powoduje natychmiastowe ustawienie wyjścia Timera T1.
2. Odmierzanie czasu następuje po wykonaniu instrukcji aktywacji Timera SF (zbrocze opadające na wejściu I5).
3. Po upływie czasu  $t=200\text{ms}$  następuje ustawienie wyjścia Q1 w stan niski ('0'), czyli wyłączenie Q1.
4. Jeżeli podczas odmierzenia czasu Timera pojawi się, na jego wejściu wyzwalającym, wysoki poziom, to następuje skasowanie licznika odmierzanego czasu. Timer zostanie ponownie wyzwolony po pojawieniu się opadającego zbocza na wejściu I5.
5. Jeśli na wejściu I1 resetującym T1 pojawi się stan wysoki następuje skasowanie licznika odmierzanego czasu i wyjścia Timera T1. Jeżeli instrukcja „L” nie zostanie użyta, to czas do odmierzenia przez T1 zostanie ustalony na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).

## 5.1.2.17.3. Timer Pojedynczy impuls (SINGLE PULSE)

Timer realizuje funkcję pojedynczego impulsu.

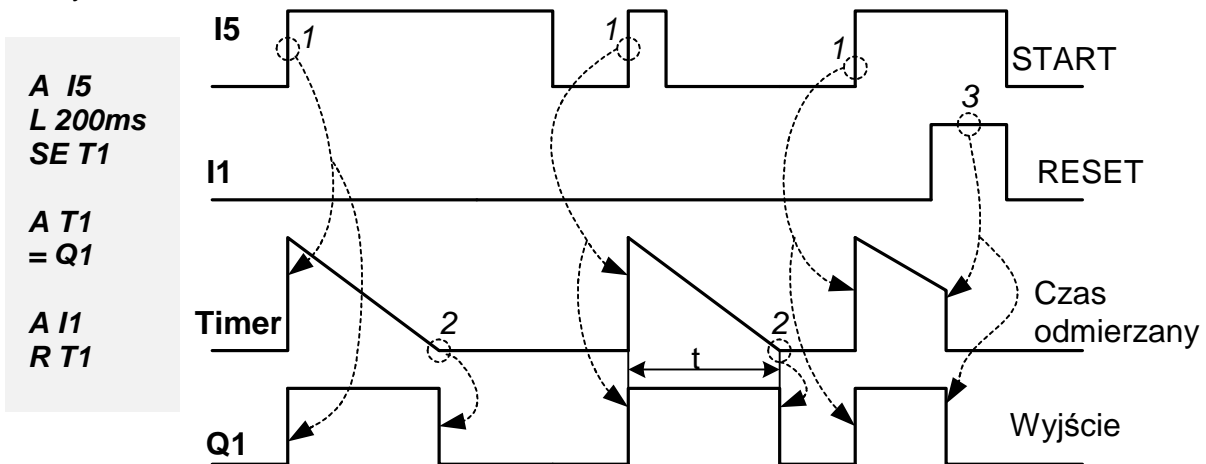
SYMBOL – SE

SKŁADNIA:

SE <T>

Czas wykonania instrukcji: 18,7µs.

Przykład:



Rys. 5.1.2.17.3. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Timera SE.

1.

Wejście I5 pełni rolę wejścia wyzwalającego (Triggera). Instrukcja ('L') ładująca określoną wartość czasu do odmierzenia powinna się znaleźć bezpośrednio przed instrukcją Timera SE.

Odmierzanie czasu następuje po wykonaniu instrukcji aktywacji Timera (narastające zbocze na wejściu I5).

2.

Przez  $t=200\text{ms}$  wyjście Q1 będzie ustawione w stanie wysokim ('1'). Stan ten może zostać przedłużony, gdy nastąpi kolejne wyzwolenie na wejściu Trigger. Po odmierzeniu zadanej wartości czasu wyjście Timera powraca do stanu niskiego ('0'), czyli Q1 przyjmuje stan niski.

3.

Jeśli na wejściu I1 resetującym T1 pojawi się stan wysoki następuje skasowanie licznika odmierzanego czasu i wyjścia Timera T1.

Jeżeli instrukcja „L” nie zostanie użyta, to czas do odmierzenia przez T1 zostanie ustalony na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).

## 5.1.2.17.4. Timer Impulsy (FLASHING)

Timer realizuje funkcję generatora fali prostokątnej o wypełnieniu 50 %.

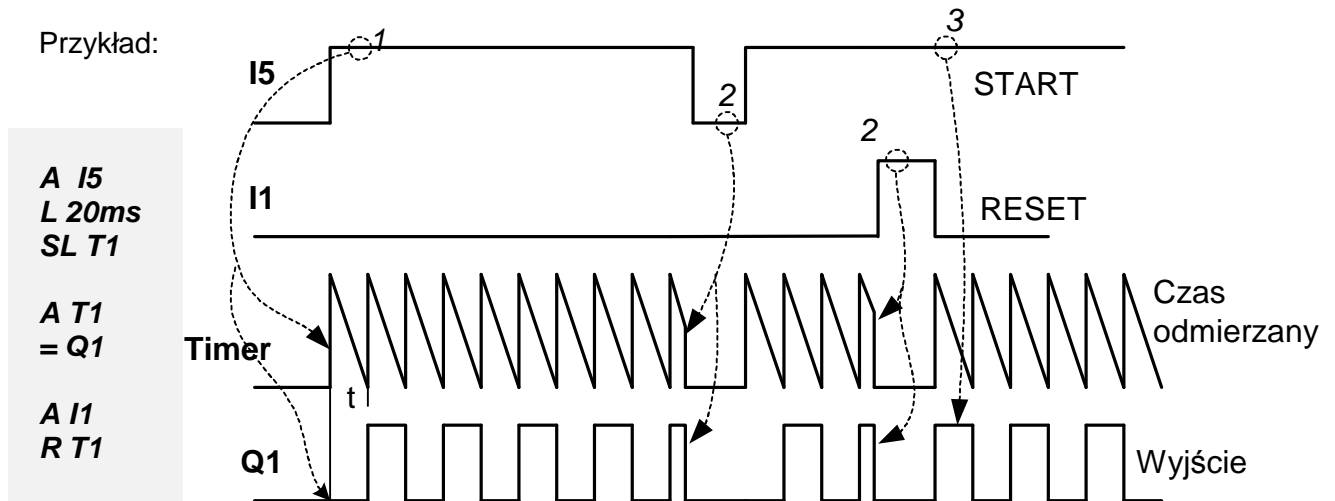
SYMBOL – **SL**

SKŁADNIA:

**SL <T>**

Czas wykonania instrukcji: 18,7µs.

Przykład:



Rys. 5.1.2.17.4. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Timera SL.

1.

Wejście I5 pełni rolę wejścia wyzwalającego (Triggera). Instrukcja ładująca określoną wartość czasu do odmierzenia powinna się znaleźć bezpośrednio przed instrukcją Timera SL.

Odmierzanie czasu następuje po wykonaniu instrukcji aktywacji Timera (poziom wysoki '1' na wejściu wyzwalającym I5).

Przez  $t=20\text{ms}$  wyjście Q1 będzie ustawione w stanie niskim ('0'), a następnie przez kolejne 20ms będzie w stanie wysokim ('1'). Sytuacja ta będzie się powtarzała tak długo, jak długo na wejściu I5 będzie stan wysoki lub do momentu wystąpienia wysokiego stanu na wejściu resetującym I1.

2.

Jeśli na wejściu I5 (Trigger) pojawi się stan niski ('0') lub na wejściu I1 (Reset) stan wysoki ('1'), następuje zerowanie licznika odmierzanego czasu oraz wyjścia Timera.

3.

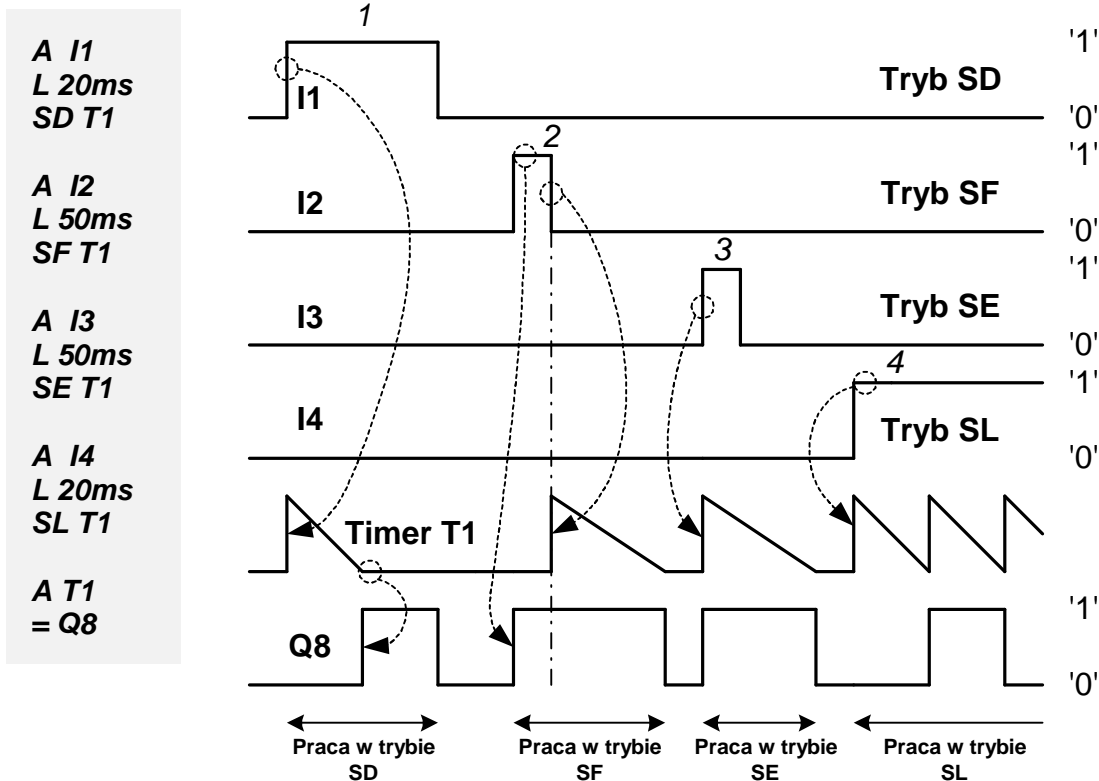
Jeśli na wejścia Reset i Trigger podane są jednocześnie stany wysokie ('1') i po jakimś czasie poziom sygnału Reset zmieni się na niski ('0'), to następuje załączenie wyjścia Timera na  $t=20\text{ms}$ , potem wyłączenie wyjścia na 20ms, załączenie itd. Timer generuje na swoim wyjściu falę prostokątną, przesuniętą o  $180^\circ$  w stosunku do przebiegu z punktu 1.

Jeżeli instrukcja „L” nie zostanie użyta, to czas do odmierzenia przez T1 zostanie ustalony na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).



## 5.1.2.17.5. Uwagi dotyczące wykorzystywania Timerów

Ten sam *Timer* można używać wiele razy, w różnych trybach.



Rys. 5.1.2.17.5. Przykładowe przebiegi czasowe sygnałów ilustrujące wielokrotne użycie Timera T1.

W powyższym przykładzie, jeśli na wejściu I1 pojawi się zbocze narastające, wówczas zostanie wyzwolony *Timer* T1 w trybie SD z czasem 20ms (1).

Jeśli na wejściu I2 pojawi się zbocze opadające, wówczas zostanie wyzwolony *Timer* T1 w trybie SF z czasem 50ms (2).

Jeśli na wejściu I3 pojawi się zbocze narastające, wówczas zostanie wyzwolony *Timer* T1 w trybie SE z czasem 50ms (3).

Jeśli na wejściu I4 pojawi się stan wysoki, wówczas zostanie wyzwolony *Timer* T1 w trybie SL z czasem 20ms (4). Na rys. 5.1.2.17.5. przedstawiono przykładowe przebiegi czasowe I1, I2, I3, I4 oraz Q8.

## 5.1.2.18. Instrukcje Liczników

## 5.1.2.18.1. Zliczanie w górę

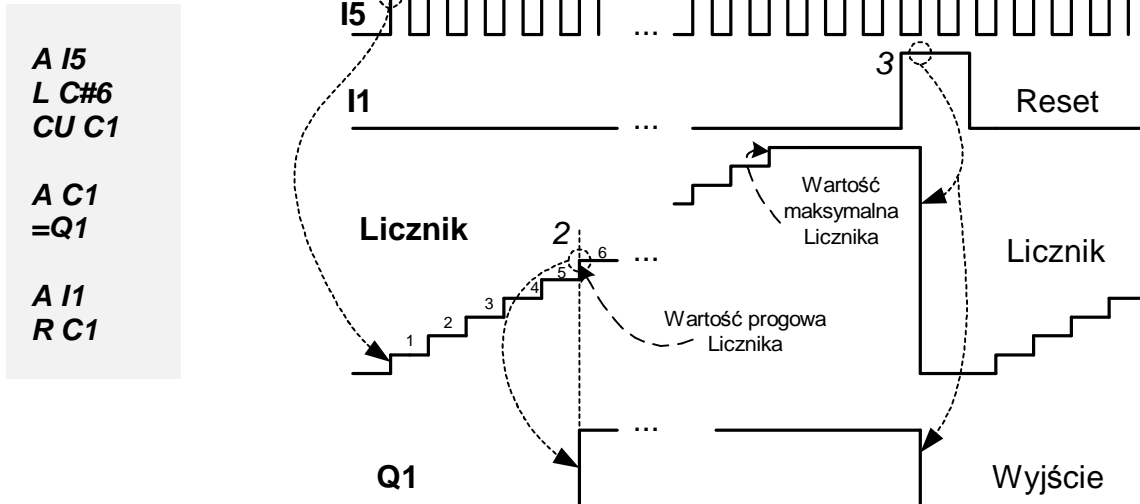
SYMBOL – **CU**

SKŁADNIA:

**CU <C>**

Czas wykonania instrukcji: 6,1µs.

Przykład:



Rys. 5.1.2.18.1. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Licznika CU.

1.

Po wystąpieniu narastającego zbocza na wejściu wyzwalającym I5 nastąpi zwiększenie o 1 bieżącej wartości *Licznika* C1.

2.

Gdy wartość bieżąca *Licznika* osiągnie wartość progową (6), to wyjście Q1 zostanie ustawione w stan wysoki.

Jeśli na wejściu wyzwalającym nadal będą pojawiały się impulsy, *Licznik* będzie je zliczał, aż do osiągnięcia wartości maksymalnej 65535, pozostawiając swoje wyjście w stanie wysokim. *Licznik* nigdy się nie przepełnia – w momencie osiągnięcia maksymalnej wartości *Licznik* przestaje reagować na impulsy wyzwalające.

3.

Jeśli na wejściu resetującym I1 pojawi się stan wysoki – nastąpi skasowanie bieżącej wartości *Licznika* C1 i jego wyjścia. Po osiągnięciu stanu niskiego na tym wejściu możliwa jest dalsza praca *Licznika*.

Jeżeli instrukcja „L” nie zostanie użyta, to wartość progę, po osiągnięciu której *Licznik* C1 ustawia swoje wyjście w stan wysoki, zostanie ustalona na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).

## 5.1.2.18.2. Zliczanie w dół

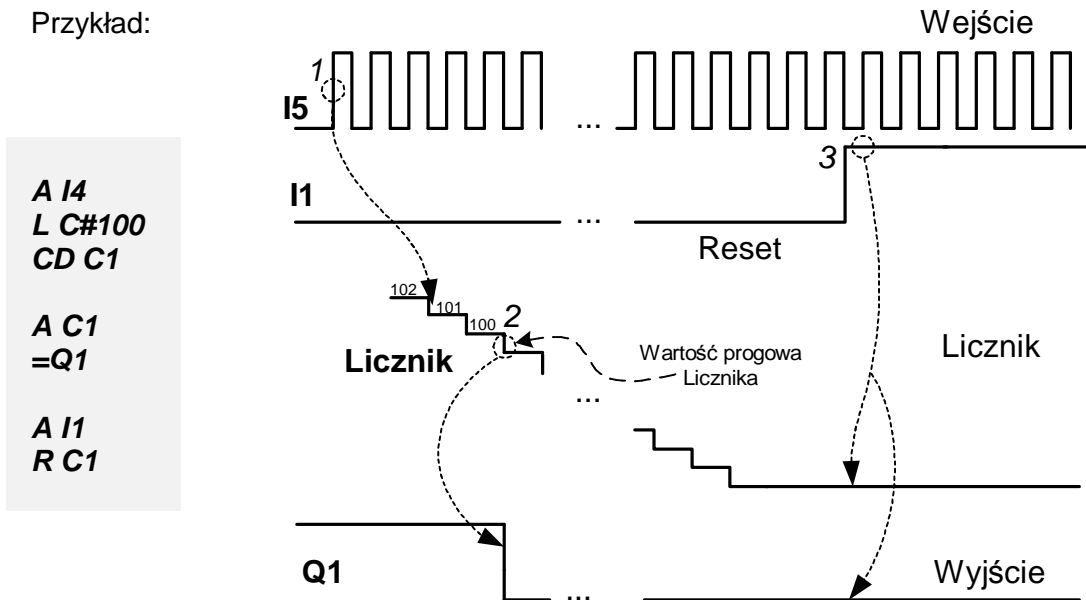
SYMBOL – **CD**

SKŁADNIA:

**CD <C>**

Czas wykonania instrukcji: 6,1µs.

Przykład:



Rys. 5.1.2.18.2. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Licznika CD.

1.

Po wystąpieniu narastającego zbocza na wejściu wyzwalającym I4 nastąpi zmniejszenie o 1 bieżącej wartości Licznika C1.

2.

Gdy wartość bieżąca Licznika impulsów spadnie poniżej wartości progowej (100), to wyjście Q1 zostanie ustawione w stan niski.

Jeśli na wejściu wyzwalającym I4 nadal będą pojawiały się impulsy, Licznik będzie je zliczał, aż do osiągnięcia wartości minimalnej 0.

Licznik nigdy się nie przepęlnia – w momencie osiągnięcia minimalnej wartości, Licznik przestaje reagować na impulsy wyzwalające.

3.

Jeśli na wejściu I1 resetującym C1 pojawi się stan wysoki – nastąpi skasowanie bieżącej wartości Licznika C1 i jego wyjścia. Dalsza praca Licznika możliwa jest po osiągnięciu stanu niskiego na wejściu Reset.

Jeżeli instrukcja „L” nie zostanie użyta, to wartość progę, po osiągnięciu którego Licznik C1 ustawia swoje wyjście w stan wysoki, zostanie ustalona na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).



Maksymalna częstotliwość impulsów zliczających zależy od czasu wykonywanego programu. Stan wejścia liczącego musi być stabilny przynajmniej przez jeden cykl obiegu pętli programu.

## 5.1.2.18.3. Uwagi dotyczące wykorzystywania Liczników

1. Wykorzystanie *Szybkiego Licznika* HC

Aby korzystać z *Szybkiego Licznika* należy:

- podłączyć do wejścia I11 sygnał wyzwalający *Licznik*
- uaktywnić *Szybki Licznik* używając instrukcji CU lub CD, np.:

```
A I1
L # 25000
CU HC1
```

W powyższym przykładzie *Szybki Licznik* ustawi swoje wyjście w stan wysoki, jeżeli zliczanie zostanie uaktywnione stanem wysokim na wejściu I1 oraz wartość bieżąca *Licznika* będzie większa lub równa 25000.

```
A I1
L # 100
CD HC1
```

W powyższym przykładzie *Szybki Licznik* ustawi swoje wyjście w stan wysoki, jeżeli zliczanie zostanie uaktywnione stanem wysokim na wejściu I1 oraz wartość bieżąca *Licznika* będzie większa lub równa 100.

Jeżeli instrukcja „L” nie zostanie użyta, to wartość progów, po osiągnięciu którego *Szybki Licznik* ustawia swoje wyjście w stan wysoki, zostanie ustalony na podstawie pliku konfiguracyjnego „\*.set” (okna Ustawień w programie PC Need).

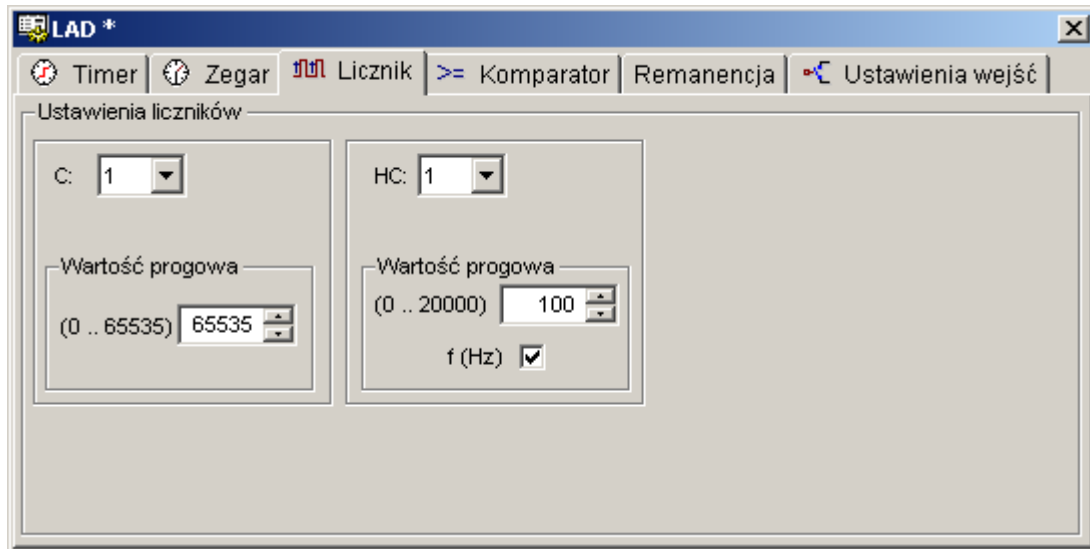
*Szybki Licznik* zlicza w górę i w dół. Po osiągnięciu wartości maksymalnej - 65535, zaczyna liczenie od 0 po wykonaniu instrukcji Reset.

*Szybki Licznik* może mierzyć także częstotliwość – odpowiedni tryb pracy zostaje ustalony za pomocą konfiguracji w programie PCNeed.



Maksymalna gwarantowana częstotliwość pracy *Szybkiego Licznika* wynosi 20kHz.

Na rys. 5.1.2.18.3.1. przedstawiono przykładowe okno ustawień *Szybkiego Licznika* HC1



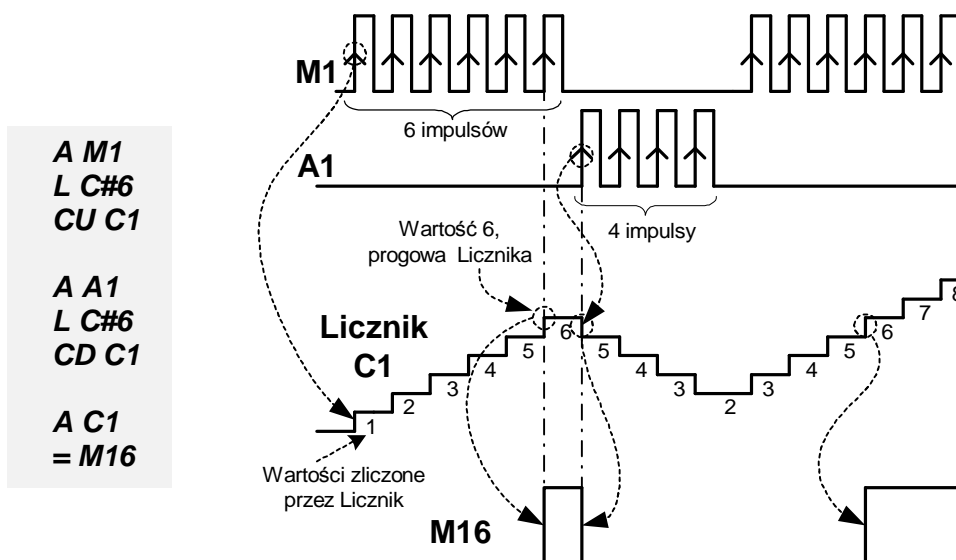
Rys. 5.1.2.18.3.1. Przykładowe okno konfiguracji Szybkiego Licznika HC1.

W podanym wyżej przykładzie *Szybki Licznik* ustawi swoje wyjście w stan wysoki, jeżeli ilość zliczanych impulsów w ciągu 1s będzie większa lub równa 100.

## 2. Jeden próg przełączający.

Aby ustawić jeden próg, przełączający wyjście *Licznika* w stan wysoki, należy używać tych samych argumentów (wartości do zliczania) w instrukcji *Load* dla CU i CD – rys. 5.1.2.18.3.2. Zbocza narastające, pojawiające się na M1 powodują zliczanie *Licznika* C1 w górę. Jeśli wartość zliczona przez C1 będzie większa lub równa 6 wówczas zostanie ustawione wyjście C1.

Zbocza narastające, pojawiające się na A1, powodują zliczanie *Licznika* C1 w dół. Jeśli wartość zliczona przez C1 będzie mniejsza od 6 wówczas wyjście C1 zostanie ustawione w stan niski.



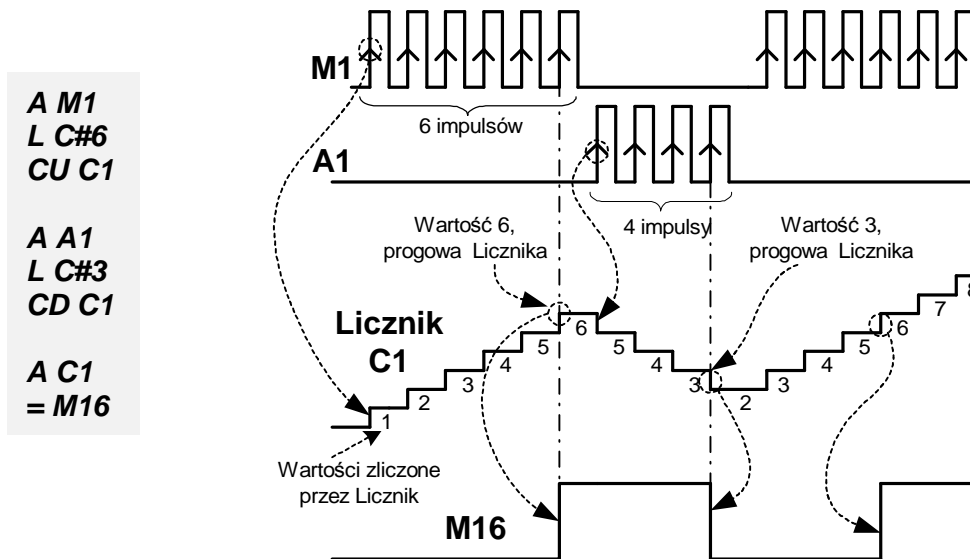
Rys. 5.1.2.18.3.2. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Licznika dla dwóch jednakowych progów przełączania.

## 3. Dwa progi przełączające (zakres).

Jeśli instrukcje *Load Liczników* używają różnych argumentów (wartości do zliczania), to zostają ustawione dwa progi przełączające – rys. 5.1.2.18.3.3.

Zbocza narastające, pojawiające się na M1 powodują zliczanie *Licznika* C1 w górę. Jeśli wartość zliczona przez C1 będzie większa lub równa 6 wówczas zostanie ustawione wyjście C1.

Zbocza narastające, pojawiające się na A1, powodują zliczanie *Licznika* C1 w dół. Dopiero, gdy wartość zliczona przez C1 będzie mniejsza od 3 – wyjście C1 zostanie ustawione w stan niski. Tak więc, przy zliczaniu w dół, wyjście C1 ustawione jest w stanie wysokim, gdy wartości zliczane przez *Licznik* będą pomiędzy 6 a 3.



Rys. 5.1.2.18.3.3. Przykładowe przebiegi czasowe sygnałów ilustrujące działanie Licznika dla dwóch różnych progów przełączania.



## 5.1.2.19. Instrukcje Zegara

Zegar jest zegarem czasu rzeczywistego i dokładna jego konfiguracja powinna być przeprowadzona przy użyciu programu PC Need, patrz rozdział 6. Dokładny opis działania Zegara przedstawiono w rozdziale 4.9. „Zegary”.

SYMBOL – *H*

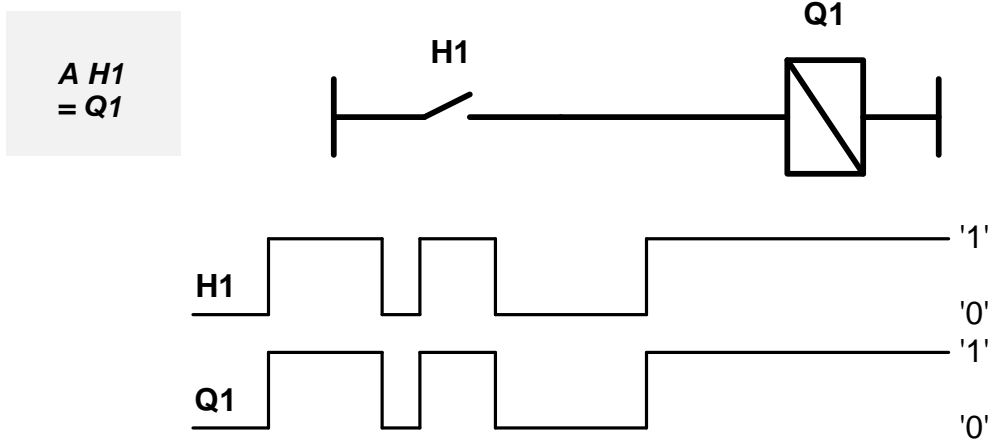
SKŁADNIA:

**<instrukcje warunkowe> H <numer Zegara>**

Przykład:

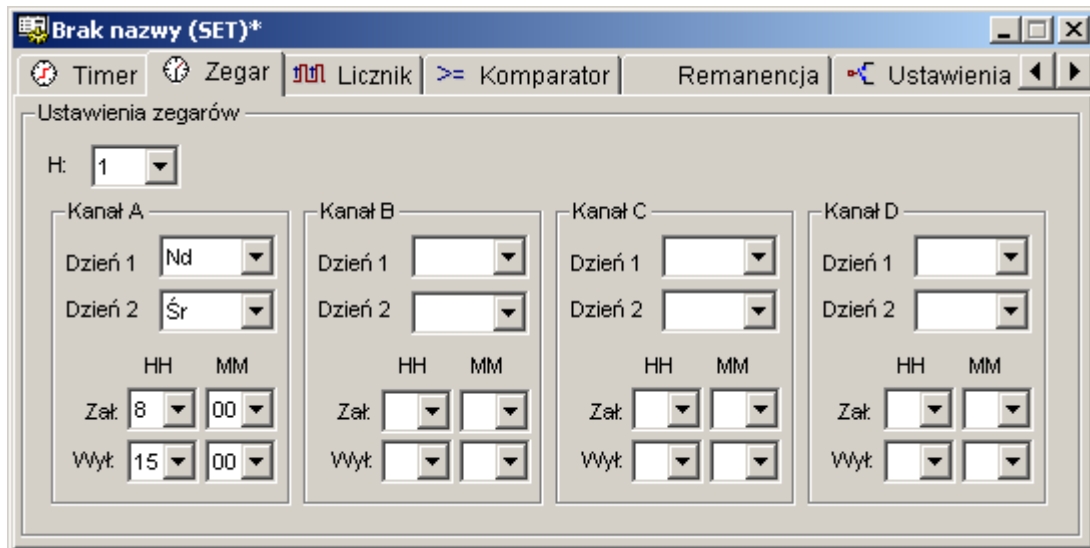
STL

Schemat przekaźnikowy



Rys. 5.1.2.19.1. Przykładowe przebiegi czasowe na styku H1 oraz na wyjściu Q1.

Konfigurujemy odpowiednio Zegar H1 za pomocą programu PC Need – patrz rozdział 6. Na rys. 5.1.2.19.2. przedstawiono przykładową konfigurację Zegara H1.



Rys. 5.1.2.19.2. Przykładowa konfiguracja Zegara H1.

Wyjście Q1 będzie ustawiane w takt zmian wyjścia zegara H1– od niedzieli do środy w godzinach od 8.00 do 15.00.



## 5.1.2.20. Wejścia analogowe

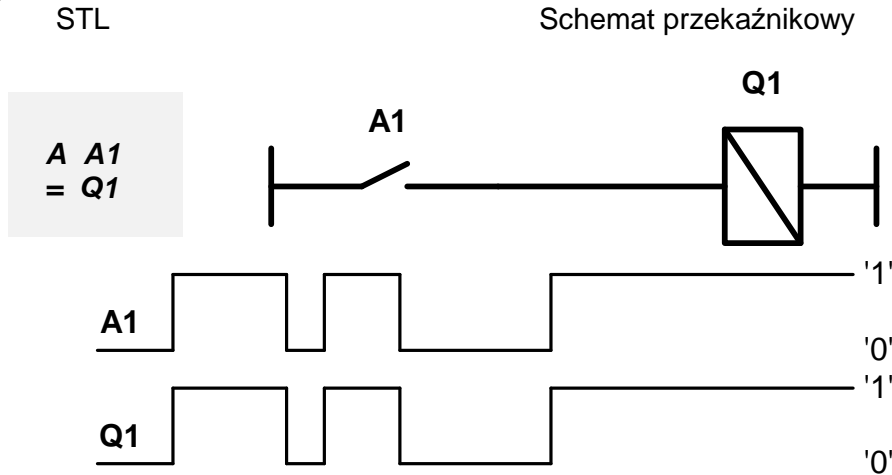
Dokładny opis działania wejść analogowych znajduje się w rozdziale 4.10. "Komparator – wejścia analogowe".

SYMBOL – **A**

SKŁADNIA:

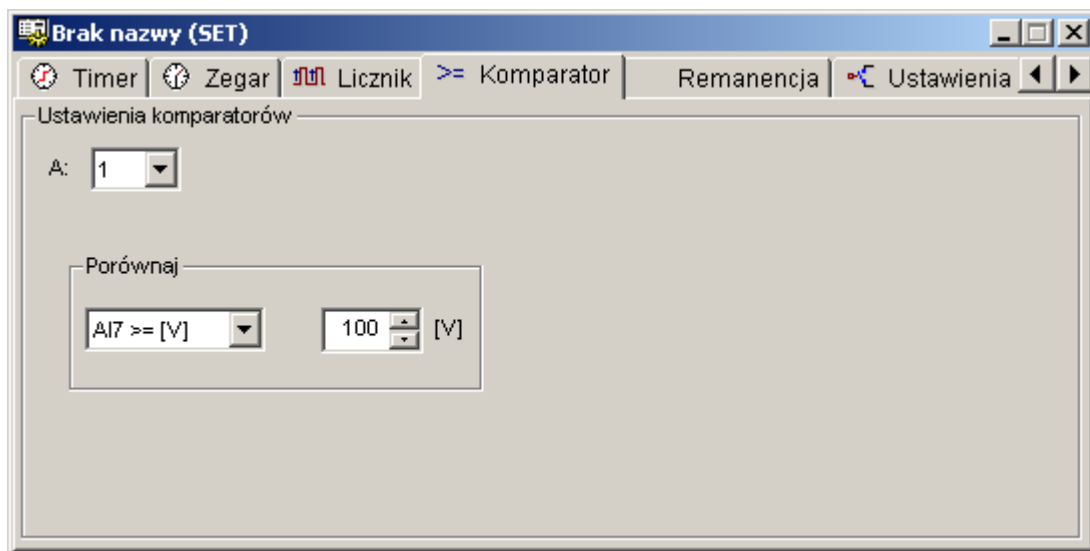
**<instrukcje warunkowe> A <numer Komparatora>**

Przykład:



Rys.5.1.2.20.1. Przykładowe przebiegi czasowe na styku A1 oraz na wyjściu Q1.

Konfigurujemy odpowiednio wejścia analogowe za pomocą programu PC Need – patrz rozdział 6. Na rys. 5.1.2.20.2. przedstawiono przykładową konfigurację *Komparatora A1*.



Rys.5.1.2.20.2. Przykładowa konfiguracja komparatora A1.

*Komparator* porównuje wartość zadaną –100 z wartością analogową na wejściu I7. Jeśli wartość napięcia na wejściu I7 będzie większa lub równa od 100V, to *Komparator* przyjmie stan '1', w przeciwnym wypadku wyjście *Komparatora* będzie w stanie '0'. Wyjście Q1 podąża za zmianami na wyjściu *Komparatora A1*.

## 5.1.2.21. Instrukcja 'L'

Instrukcja 'L' służy do określenia odpowiednich czasów dla *Timerów* oraz wartości progowych (do zliczania) dla *Liczników*.

SYMBOL – **L**

SKŁADNIA:

**L <wartość>**

## 5.1.2.21.1. Instrukcja 'L' dla Timerów

## 5.1.2.21.1.1. Wartości stałe czasów dla Timerów

Czas wykonania instrukcji: 8,3µs

Parametr **<wartość>** dla instrukcji 'L' przyjmuje odpowiednie stałe wartości czasów z zakresów podanych w tab. 5.1.2.21.1. np.:

**L 100ms** //Do Timera T1 pracującego w trybie SL zostanie załadowana wartość 100ms  
**SL T1**

**L 10min** //Do Timera T2 pracującego w trybie SD zostanie załadowana wartość 10 min  
**SD T2**

**L 1h.34min** //Do Timera T8 pracującego w trybie SF zostanie załadowana wartość  
**SF T8** //1h.34min

Tabela 5.1.2.21.1. przedstawia dostępne wartości czasów, które mogą być używane z instrukcją 'L'.

Tab .5.1.2.21.1. Argumenty instrukcji 'L' dla Timerów.

| Format czasu               | Zakres              | Krok | Przykładowe wartości   |
|----------------------------|---------------------|------|------------------------|
| s.ms (sekundy.milisekundy) | 0s.10ms – 99s.990ms | 10ms | 0.50ms, 24s, 50s.120ms |
| min.s (minuty.sekundy)     | 0min.1s – 99min.59s | 1s   | 2min, 32min, 98min.24s |
| h.min (godziny.minuty)     | 0h.1min – 99h.59min | 1min | 1h, 5h.18min           |

## 5.1.2.21.1.2. Wartości czasów dla Timerów określane na podstawie ustawienia Potencjometru

Czas wykonania instrukcji: 10,3µs.

Można również używać wartości czytanej z Potencjometru jako zadanego czasu do odmierzenia przez *Timery*, wtedy argument **<wartość>** instrukcji 'L' może przyjmować następujące wartości (patrz Tabela 5.1.2.21.2.):

1. x10ms

**L Pot x10ms** //Czas do odmierzenia = aktualna wartość Potencjometru (1 – 255)x10ms,  
//np. gdy wartość ustawiona Potencjometru = 25, to czas do odmierzenia =  
//25x10ms = 250ms.

2. x100ms

**L Pot x100ms** //Czas do odmierzenia = aktualna wartość Potencjometru (1 – 255) x 100ms,  
//np. wartość ustawiona Potencjometru = 15, to czas do odmierzenia =  
//15x100ms = 1500ms = 1,5s

## 3. x1s

**L Pot x1s** //Czas do odmierzenia = aktualna wartość Potencjometru (1 – 255) x 1s,  
//np. wartość ustawiona Potencjometru = 10, to czas do odmierzenia =  
//10x1s = 10s

## 4. x10s

**L Pot x10s** //Czas do odmierzenia = aktualna wartość Potencjometru (1 – 255) x 10s,  
//np. wartość ustawiona Potencjometru = 8, to czas do odmierzenia =  
//8x10s = 80s

## 5. x1min

**L Pot x1min** //Czas do odmierzenia = aktualna wartość Potencjometru (1 – 255) x 1min,  
//np. wartość ustawiona Potencjometru = 255, to czas do odmierzenia =  
//255x1min = 255min

Tab.5.1.2.21.1.2. Argumenty instrukcji 'L' dla Timerów uzyskiwane poprzez nastawy Potencjometru.

| Zakres Potencjometru | Mnożnik | Zakres czasu    |
|----------------------|---------|-----------------|
| 1 – 255              | x 10ms  | 10ms – 2,55s    |
|                      | x 100ms | 100ms – 25,50s  |
|                      | x 1s    | 1s – 4min15s    |
|                      | x 10s   | 10s – 42min30s  |
|                      | x 1min  | 1min – 255min0s |

5.1.2.21.1.3. Wartości czasów dla *Timerów* określane na podstawie wartości napięć na wejściach analogowych napięciowych

Czas wykonania instrukcji: 10,3µs.



Istnieje możliwość używania, do odmierzenia czasów dla *Timerów*, wartości napięć czytanych z wejść analogowych I7, I8 dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub I14, I15, I16 dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8.

Dla wejść analogowych napięciowych argument **<wartość>** instrukcji 'L' może przyjmować wartości czasów przedstawionych w tabeli 5.1.2.21.1.3.

Tab.5.1.2.21.1.3. Argumenty instrukcji 'L' dla Timerów uzyskiwane poprzez odczyt wejść analogowych napięciowych.

| Zakres napięcia mierzonego na wejściu analogowym [V] | Mnożnik zakresu | Mnożnik ogólny | Zakres czasu     |
|--|-----------------|----------------|------------------|
| 0,10 – 25,50<br>(krok 0,10)                          | x 10ms          | x 10           | 10ms – 2s550ms   |
|  | x 100ms         |                | 100ms – 25s500ms |
|  | x 1s            |                | 1s – 4min15s     |
|  | x 10s           |                | 10s – 42min30s   |
|  | x 1min          |                | 1min – 255min0s  |
| 0,05 – 12,75<br>(krok 0,05)                          | x 10ms          | x 20           | 10ms – 2s550ms   |
|  | x 100ms         |                | 100ms – 25s500s  |
|  | x 1s            |                | 1s – 4min15s     |
|  | x 10s           |                | 10s – 42min30s   |
|  | x 1min          |                | 1min – 255min0s  |

Odmierzany czas dla przekaźników NEED-24DC-x1-..., NEED-12DC-x1.. obliczamy:

*Wartość napięcia na wejściu analogowym [V] x Mnożnik zakresu x Mnożnik ogólny = Czas odmierzanym*

W składni języka STL używamy symboli **AI7** lub **AI8** dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub **AI14**, **AI15**, **AI16** dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8 np.:

**L AI7 x1min** //Czas do odmierzenia = aktualna wartość napięcia na wejściu analogowym  
//AI7[V] x 1min x 10, (zakres 0,1V - 25,50V)  
//np. wartość napięcia na wejściu analogowym AI7 = 20V,  
//to czas do odmierzenia = 20V x 1min x 10 = 20min x 10 = 200min

**L AI14 x100ms** //Czas do odmierzenia = aktualna wartość napięcia na wejściu  
//analogowym AI14[V] (zakres 0,05V - 12,75V) x 100ms x 20,  
//np. wartość napięcia analogowego AI14 = 10V,  
//to czas do odmierzenia = 10V x 100ms x 20 = 1000ms x 20 = 20s



Zwiększoną rozdzielczość wejść analogowych (zakres pracy 0,05V – 12,75V) możemy stosować tylko dla przekaźników NEED-12DC-x1-16-8 lub NEED-24DC-x1-16-8.

#### 5.1.2.21.1.4. Wartości czasów dla Timerów określane na podstawie wartości prądów na wejściach analogowych prądowych

Czas wykonania instrukcji: 10,3µs.

Dla wejść analogowych prądowych (tylko dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8) argument **<wartość>** instrukcji 'L' może przyjmować wartości czasów przedstawionych w tabeli 5.1.2.21.1.4.

Tab.5.1.2.21.1.4. Argumenty instrukcji 'L' dla Timerów uzyskiwane poprzez odczyt wejść analogowych prądowych.

| Zakres prądu mierzonego na wejściu analogowym [mA] | Mnożnik zakresu | Mnożnik ogólny | Zakres czasu     |
|--|-----------------|----------------|------------------|
| 0,2 – 51,0<br>(krok 0,2)                           | x 10ms          | x 5            | 10ms – 2s550ms   |
|  | x 100ms         |                | 100ms – 25s500ms |
|  | x 1s            |                | 1s – 4min15s     |
|  | x 10s           |                | 10s – 42min30s   |
|  | x 1min          |                | 1min – 255min0s  |
| 0,1 – 25,50<br>(krok 0,1)                          | x 10ms          | x 10           | 10ms – 2s550ms   |
|  | x 100ms         |                | 100ms – 25s500s  |
|  | x 1s            |                | 1s – 4min15s     |
|  | x 10s           |                | 10s – 42min30s   |
|  | x 1min          |                | 1min – 255min0s  |

Odmierzany czas dla przekaźników NEED-24DC-x1-16-8, NEED-12DC-x1-16-8 obliczamy:

*Wartość prądu na wejściu analogowym [mA] x Mnożnik zakresu x Mnożnik ogólny = Czas odmierzanym*

**L AI16 x1min** //Czas do odmierzenia = aktualna wartość prądu w [mA] (zakres 0,2 – 51) x //1min x 5  
//np. wartość prądu AI16 = 10mA  
//to czas do odmierzenia = 10mA x 1min x 5 = 10minx5=50min

**L AI14 x1s** //Czas do odmierzenia = aktualna wartość prądu w [mA] (zakres 0,1 – 25,5) x //1s x 10  
//np. wartość prądu AI16 = 5mA  
//to czas do odmierzenia = 5mA x 1s x 10 = 5s x 10 = 50s



Zwiększoną rozdzielczość wejść analogowych (zakres pracy 0,1– 25,50mA) możemy stosować tylko dla przekaźników NEED-12DC-x1-16-8 lub NEED-24DC-x1-16-8.

### 5.1.2.21.2. Instrukcja 'L' dla Liczników

#### 5.1.2.21.2.1. Wartości stałe, progowe dla Liczników

Czas wykonania instrukcji: 8,3µs

Parametr **<wartość>** instrukcji 'L' przyjmuje odpowiednie stałe liczby, poprzedzone znakami **C#**, do zliczenia dla *Liczników* z zakresu 0–65535 np.:

**L C#10**  
**CU C1** //Ustawienie wartości 10 do zliczania przez Licznik C1 liczący w górę

**L C#1000**  
**CD C8** //Ustawienie wartości 1000 do zliczania przez Licznik C8 liczący w dół

#### 5.1.2.21.2.2. Wartości progowe dla Liczników określone na podstawie ustawienia Potencjometru

Czas wykonania instrukcji: 10,3µs.

Można również używać wartości czytanej z Potencjometru jako zadanej wartości do zliczania przez *Liczniki*, wtedy format instrukcji 'L' może przyjmować następującą postać:

1.

**L Pot x1** //Wartość do zliczania z zakresu (1 – 255)x1 (np. Potencjometr ma  
**CU C1** //ustawioną wartość 23 – to wartość do zliczania przez C1 będzie  
 //równa 23x1=23)

2.

**L Pot x10** //Wartość do zliczania z zakresu (1 – 255)x10 (np. Potencjometr  
**CD C2** //ma ustawioną wartość 23 – to wartość do zliczania przez C2 będzie  
 //równa 23x10=230)

3.

**L Pot x100** //Wartość do zliczania z zakresu (1 – 255)x100 (np. Potencjometr  
 //ma ustawioną wartość 23 – to wartość do zliczania przez C3 będzie  
 //równa 23x100=2300)

Tab.5.1.2.21.2.1. Argumenty instrukcji 'L' dla Liczników uzyskiwane poprzez nastawy potencjometru.

| Zakres Potencjometru | Mnożnik zakresu | Zakres liczb |
|----------------------|-----------------|--------------|
| 1 – 255              | x 1             | 1 – 255      |
|                      | x 10            | 10 – 2550    |
|                      | x 100           | 100 – 25500  |

#### 5.1.2.21.2.3. Wartości progowe dla *Liczników* określone na podstawie wartości napięć na wejściach analogowych napięciowych

Czas wykonania instrukcji: 10,3µs.



Istnieje możliwość używania, do określania progów dla *Licznika*, wartości napięć czytanych z wejść analogowych I7, I8 dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub I14, I15, I16 dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8. W takim przypadku argument **<wartość>** instrukcji 'L' może przyjmować wartości progów przedstawionych w tabeli 5.1.2.21.2.3.

Tab.5.1.2.21.2.3. Argumenty instrukcji 'L' dla Liczników uzyskiwane poprzez odczyt wejść analogowych.

| Zakres napięcia na wejściu analogowym [V] | Mnożnik zakresu | Mnożnik ogólny | Zakres liczb |
|---|-----------------|----------------|--------------|
| 0,1 – 25,5<br>(krok 0,1)                  | x 1             | x 10           | 1– 255       |
|   | x 10            |                | 10 – 2550    |
|   | x 100           |                | 100 – 25500  |
| 0,05 – 12,75<br>(krok 0,05)               | x 1             | x 20           | 1– 255       |
|   | x 10            |                | 10 – 2550    |
|   | x 100           |                | 100 – 25500  |

Ustawiony próg dla przekaźników NEED-24DC-x1-..., NEED-12DC-x1-.. obliczamy:

*Wartość napięcia na wejściu analogowym [V] x Mnożnik zakresu x Mnożnik ogólny = próg dla Licznika*

W składni języka STL używamy symboli **A17** lub **A18** dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub **A14**, **A15**, **A16** dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8 np.:

**L A17 x100** //Wartość do zliczania = aktualna wartość napięcia na wejściu analogowym  
 //(zakres 0,1V – 25,5V) A17[V] x 100 x 10,  
 //np. wartość napięcia na A17=10V  
 //to wartość do zliczenia = 10V x 100 x 10 = 10V x 1000 = 10 000

**L A15 x10** // Wartość do zliczania = aktualna wartość napięcia na wejściu analogowym  
 //(zakres 0,05V – 12,75V) A15[V] x 10 x 20,  
 //np. wartość napięcia na A15 = 1V  
 //to wartość do zliczania = 1V x 10 x 20 = 1V x 200 = 200



Zwiększoną rozdzielczość wejść analogowych (zakres pracy 0,05V – 12,75V) możemy stosować tylko dla przekaźników NEED-12DC-x1-16-8 lub NEED-24DC-x1-16-8.

5.1.2.21.2.4. Wartości progowe dla *Liczników* określane na podstawie wartości prądów na wejściach analogowych prądowych

Czas wykonania instrukcji: 10,3µs.



Istnieje możliwość używania, do określania progów dla *Licznika*, wartości prądów mierzonych na wejściach analogowych I14, I15, I16 dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8. W takim przypadku argument **<wartość>** instrukcji 'L' może przyjmować wartości progów przedstawionych w tabeli 5.1.2.21.2.4.

Tab.5.1.2.21.2.4. Argumenty instrukcji 'L' dla Liczników uzyskiwane poprzez odczyt wejść analogowych.

| Zakres prądu na wejściu analogowym [mA] | Mnożnik zakresu | Mnożnik ogólny | Zakres liczb |
|---|-----------------|----------------|--------------|
| 0,2 – 51,0<br>(krok 0,2)                | x 1             | x 5            | 10– 255      |
|   | x 10            |                | 100 – 2550   |
|   | x 100           |                | 1000 – 25500 |
| 0,1 – 25,5<br>(krok 0,1)                | x 1             | x 10           | 10– 255      |
|   | x 10            |                | 100 – 2550   |
|   | x 100           |                | 1000 – 25500 |

Ustawiony próg dla przekaźników NEED-24DC-x1-..., NEED-12DC-x1-.. obliczamy:

*Wartość prądu na wejściu analogowym [mA] x Mnożnik zakresu x Mnożnik ogólny = próg dla Licznika*

**L AI16 x100** //Wartość do zliczania = aktualna wartość prądu na wejściu analogowym  
 //(zakres 0,2mA – 51mA) AI16 [mA] x 100 x 5,  
 //np. wartość prądu AI16 = 1mA  
 //to wartość do zliczenia = 1mA x 100 x 5 = 1mA x 500 = 500

**L AI15 x10** // Wartość do zliczania = aktualna wartość prądu na wejściu analogowym  
 //(zakres 0,1mA – 25,5mA) AI15[mA] x 10 x 10,  
 //np. wartość prądu AI15 = 10mA  
 //to wartość do zliczenia = 10mA x 10 x 10 = 10mA x 100 = 1000



Zwiększoną rozdzielczość wejść analogowych (zakres pracy 0,1– 25,50mA) możemy stosować tylko dla przekaźników NEED-12DC-x1-16-8 lub NEED-24DC-x1-16-8.

Przykład zastosowania instrukcji 'L':

**A I5**  
**L 20s**  
**SF T1**

**A I5**  
**L C#10**  
**CU C8**

**A I8**  
**L Pot x1s**  
**SE T2**

**A I5**  
**L AI16 x10**  
**CU C1**

Do Timera T1 zostaje załadowana wartość 20s.

Dla Licznika C8 zostaje ustalona stałą wartością progowa 10 przełączająca jego stan wyjściowy z niskiego ('0') na wysoki ('1').



Do *Timera* T2 zostaje załadowana wartość z Potencjometru pomnożona przez 1s  
 Dla *Licznika* C1 zostaje ustalona wartość progowa określona za pomocą wartości napięcia analogowego występującego na AI16, pomnożona przez 10 (Mnożnik zakres 0,1 – 25,5V) x 10 (mnożnik ogólny), przełączająca jego stan wyjściowy z niskiego ('0') na wysoki ('1').

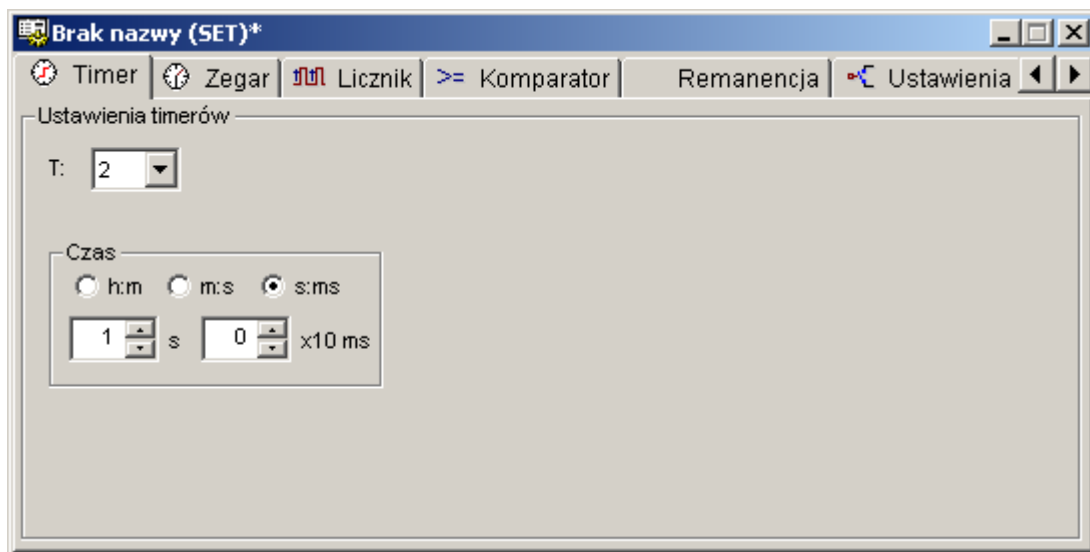
#### 5.1.2.21.3. Uwagi dotyczące używania instrukcji 'L'

1. Jeśli w programie nie została wykonana żadna instrukcja *Load*, to wartości czasów odmierzanym przez *Timery* oraz wartości progowe dla *Liczników* określone są w programie PC Need, w pliku konfiguracyjnym „\*.set” np.:

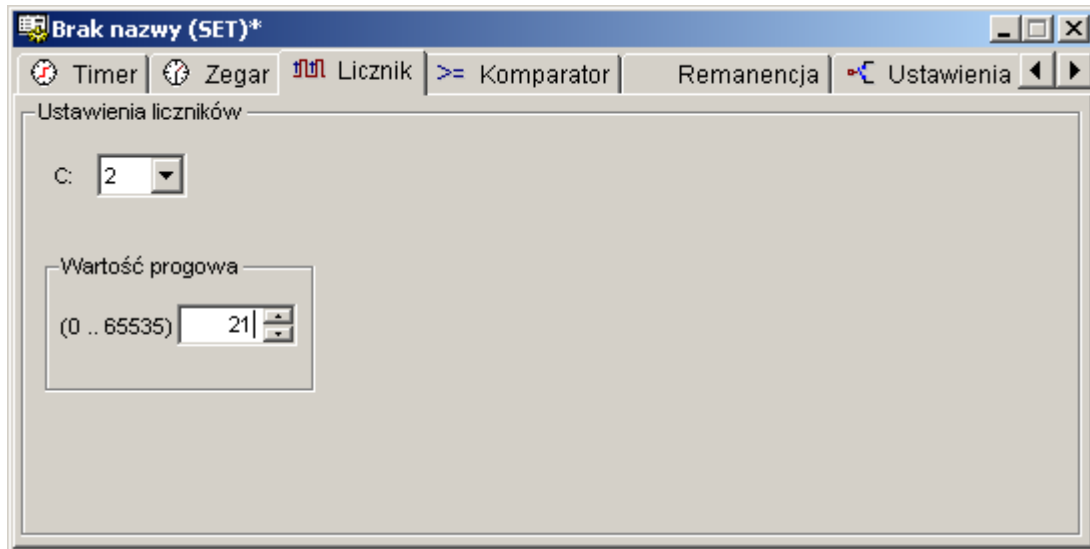
**A I3  
SE T2**

**A I5  
CD C2**

W powyższym przykładzie *Timer* T2 będzie odmierzał czas 1s, ustawiony w programie PC Need, natomiast *Licznik* będzie ustawiał/kasował swoje wyjście dla progu 21. Poniższe konfiguracje przedstawione są na rys. 5.1.2.21.3.1. oraz rys. 5.1.2.21.3.2.



Rys.5.1.2.21.3.1. Ustawienie czasu dla Timera T2.



Rys.5.1.2.21.3.2. Ustawienie czasu dla Licznika C2.

2. Jeśli w programie została wykonana instrukcja *Load*, to wszystkie wartości czasów do odmierzenia dla *Timerów* i wartości do zliczenia dla *Liczników*, występujących bezpośrednio po *Load*, określa ta instrukcja.

**A I3**  
**SE T2**

**A I8**  
**L 1min**  
**SE T3**

**A I8**  
**SE T4**

W powyższym przykładzie wyzwolenie *Timera* T2 narastającym zboczem na wejściu I3 spowoduje, iż T2 będzie odmierzał czas określony w programie PC Need, w pliku konfiguracyjnym.

Jeśli na wejściu I8 pojawi się zbocze narastające, to *Timer* T3 będzie odmierzał czas określony w instrukcji *Load* – 1min, a *Timer* T4 będzie odmierzał czas określony w pliku ustawień „\*.set”.

#### 5.1.2.22. Instrukcja zawsze ustawiająca SET

Instrukcja ‘*SET*’ ustawia na stałe stan wysoki ‘1’.

SYMBOL – **SET**

SKŁADNIA:

**SET**

Czas wykonania instrukcji: 8,9µs.

Instrukcja ‘*SET*’ jest instrukcją bezwarunkową (wykonywaną zawsze), ustawiającą na stałe stan logiczny ‘1’ w części warunkowej obwodu.

Przykład:

```
SET  
= Q4  
SL T1  
S M16
```

Wyjście Q4, Znacznik M16 po wykonaniu tej instrukcji, na stałe będą ustawione w stan wysoki '1'. Natomiast *Timer* T1 zostanie na stałe wyzwolony i będzie pracował w trybie generatora impulsów.

#### 5.1.2.23. Instrukcja zawsze kasująca CLR

Instrukcja 'CLR' ustawia na stałe stan niski '0'.

SYMBOL – **CLR**

SKŁADNIA:

**CLR**

Czas wykonania instrukcji: 8,9µs.

Instrukcja 'CLR' jest instrukcją bezwarunkową (wykonywaną zawsze), ustawiającą na stałe stan logiczny '0' w części warunkowej obwodu.

Przykład:

```
CLR  
= Q4  
= M1  
SL T1
```

Znacznik M1 i wyjście Q1, po wykonaniu instrukcji 'CLR', na stałe będą ustawione w stan niski '0' natomiast *Timer* T1 nigdy nie wystartuje.

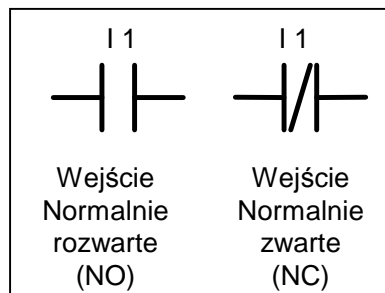
## 5.2. Programowanie w języku graficznym LAD

LAD (ang. *Ladder Diagram*) to prosty sposób programowania używany do edycji programów w sterownikach PLC. Dzięki zachowaniu podstawowych kanonów standardu języka, nie powinien on sprawić żadnych problemów użytkownikom, którzy mieli już do czynienia z podobnym sposobem tworzenia programów. Użytkownicy przełączników NEED, którzy spotykają się z nim po raz pierwszy, będą mieli możliwość zapoznania się i zastosowania tego sposobu programowania, który nawiązuje do „rysowania” schematów połączeń elektrycznych.

### 5.2.1. Symbole w LAD

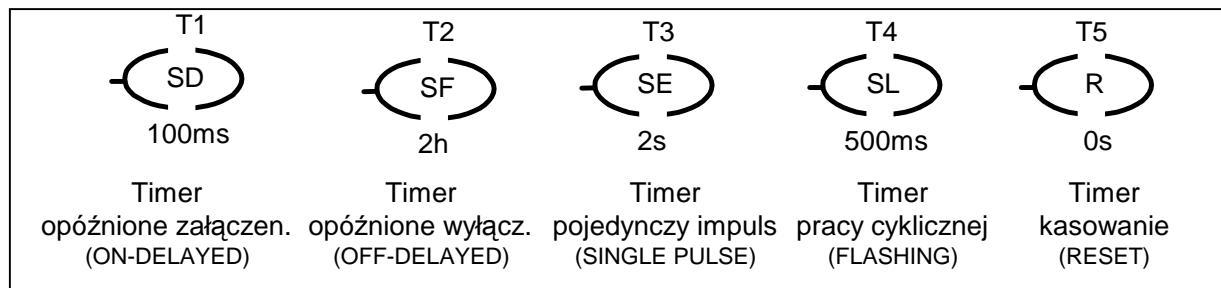
Język schematów drabinkowych LAD – bazuje na symbolach logiki stykowo-przełącznikowej. W ten sposób mogą być reprezentowane styki (elementy wejściowe), wyjścia dwustanowe (odzwierciedlenie cewek przełącznika) oraz wyjścia funkcyjne.

Podstawowe symbole języka LAD dla odzwierciedlenia wejść są przedstawione na rys. 5.2.1.1.

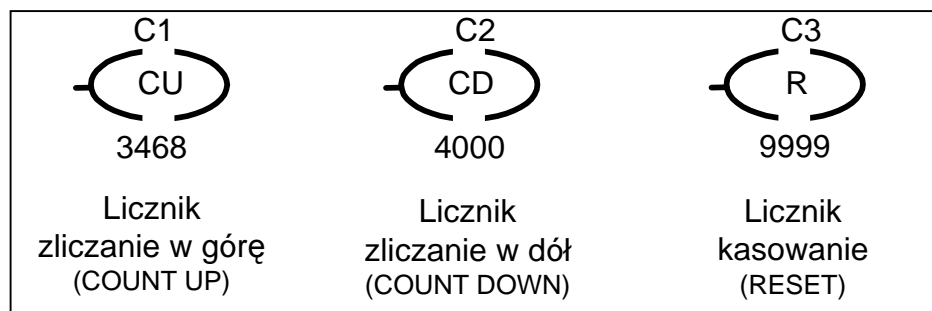


Rys. 5.2.1.1. Podstawowe elementy języka LAD – wejścia (styki)

Wyjścia funkcyjne to *Timery* rys. 5.2.1.2. oraz *Liczniki* rys. 5.2.1.3.

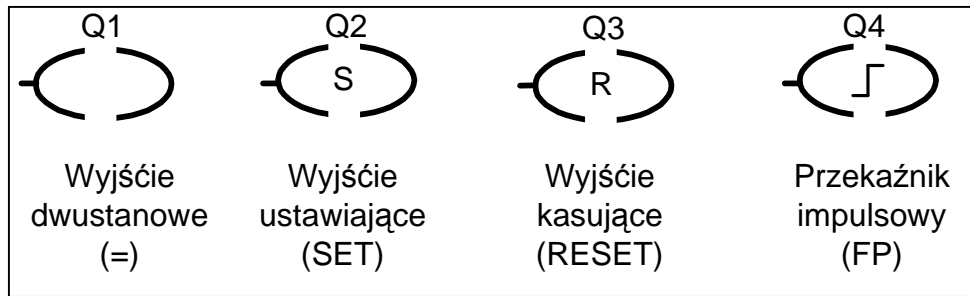


Rys. 5.2.1.2. Elementy języka LAD – Timery.



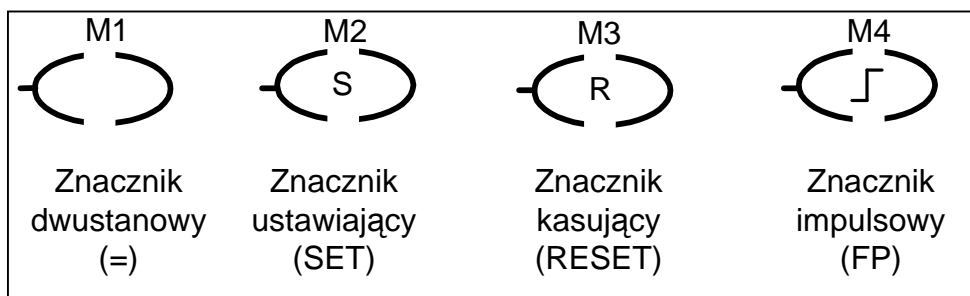
Rys. 5.2.1.3. Elementy języka LAD – Liczniki.

Symbole języka LAD dla odzwierciedlenia wyjść są przedstawione na rys. 5.2.1.4.



Rys.5.2.1.4. Elementy języka LAD – wyjścia.

Symbole języka LAD dla odzwierciedlenia Znaczników są przedstawione na rys. 5.2.1.5.



Rys.5.2.1.5. Elementy języka LAD – Znaczniki

### 5.2.2. Wejścia

Z punktu widzenia programu LAD wejściem może być nie tylko fizyczny styk elementu elektrycznego (wejście dyskretne), ale także stan (poziom logiczny) *Timera*, *Licznika*, *Zegara*, *Znacznika* lub nawet wyjścia. Ponieważ elementom tym w trakcie ich działania przypisywane są wartości dwustanowe ('0' lub '1') możemy je sprawdzać i uzależniać od nich działanie innych elementów obwodu.



**Uwaga:** Sprawdzenie wyjść polega jedynie na uzyskaniu informacji programowej o stanie rejestru, który steruje danym wyjściem fizycznym, tzn. nie uwzględnia się sprawności przełącznika i układu wykonawczego danego wyjścia.

### 5.2.3. Wyjścia

W najprostszym przypadku mamy do czynienia z dwustanowym elementem, takim jak przełącznik, którego cewka jest zasilana bądź nie. Możemy wtedy mówić, że przełącznik działa (jest aktywny), jeśli jego cewka jest zasilana, – czyli przyjmujemy dla niej określony stan logiczny. W naszym wypadku będziemy stosować logikę dodatnią, czyli będzie to stan '1' dla wyjścia załączonego, a dla niedziałającego wyjścia stan logiczny '0'.

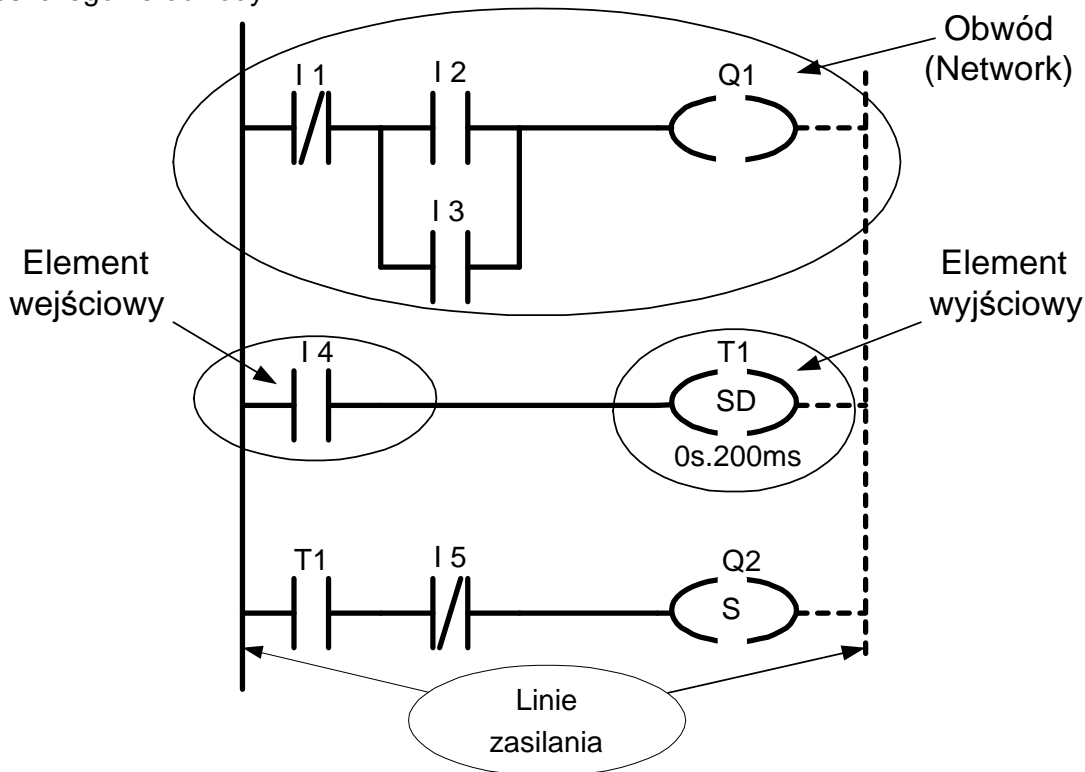
W zależności od przypisanej funkcji (patrz tab. 5.2.6.) wyjście może być ustawiane w ciągłej zależności od wejść (instrukcja „=”) – analogia do przełącznika działającego, jeśli cewka jest zasilana. Inaczej zachowują się wyjścia typu SET i RESET, gdzie po spełnieniu warunków następuje ustawienie na stałe (instrukcja „S”) stanu logicznego '1'. Stan taki pozostaje aż do momentu wykonania operacji kasującej (R) – odpowiada to zachowaniu się przełącznika z podtrzymaniem.

Wyjścia w LAD również mogą nie posiadać swoich fizycznych odpowiedników w strukturze przełącznikowej, są to tzw. wyjścia funkcyjne, które umożliwiają użycie takich elementów jak *Timer*, *Licznik*, *Zegar*, *Znacznik*. Elementy te są ustawione podobnie jak fizyczne wyjścia (przyjmują stan '0' lub '1'), w zależności od przypisanej im funkcji (patrz tab. 5.2.6.).

### 5.2.4. Struktura programu w LAD

Symbole umieszczane są w obwodach (ang. *Network*). Obwody umieszczone są w sposób podobny do szczebli (ang. *Rungs*) w drabince. Kolejne obwody (szczeble drabiny) odczytywane są kolejno od góry do dołu. Po dojściu do ostatniego szczebla, proces śledzenia programu rozpoczyna się od początku.

Obwód ograniczony jest z lewej i prawej strony przez szyny prądowe. Prawa szyna może być widoczna, ale także może być pominięta na rysunku. Ze względu na analogię z schematem przekaźnikowym programy w LAD mogą być czytane jako przepływ prądu od lewej do prawej linii pionowej (np. lewa strona zasilanie, prawa potencjał masy) poprzez poszczególne obwody.



Rys. 5.2.3. Przykładowa aplikacja zrealizowana w języku LAD.

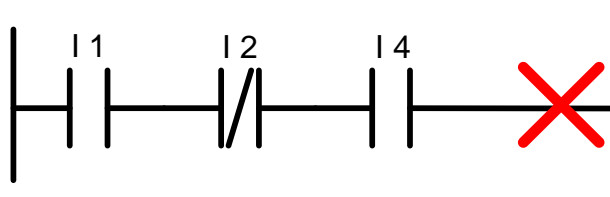
### 5.2.5. Struktura obwodu LAD

Obwód musi posiadać odpowiedni format i składnię. Oto kilka najważniejszych zasad:

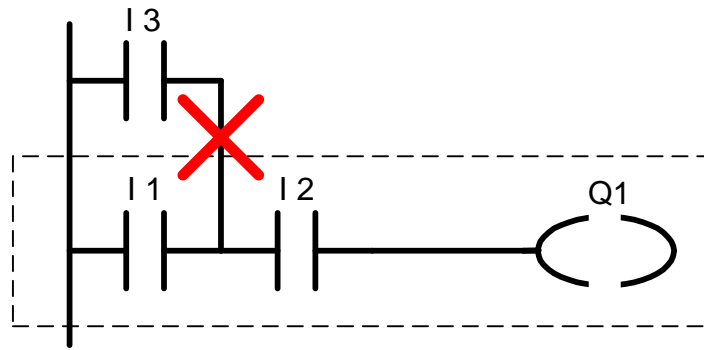
- każdy obwód może posiadać do 16 linii równoległych,
- ostatnim elementem szeregowego połączenia w danym obwodzie musi być jeden z elementów wykonawczych (wyjście dwustanowe lub funkcyjne),
- obwód może posiadać maksymalnie do 16 elementów wyjściowych,
- nie może wystąpić rozgałęzienie mające początek lub koniec wewnątrz innego odgałęzienia, które łączy się z „linią zasilającą” lub wyjściami.

Poniżej przedstawiono przykłady niedozwolonych połączeń:

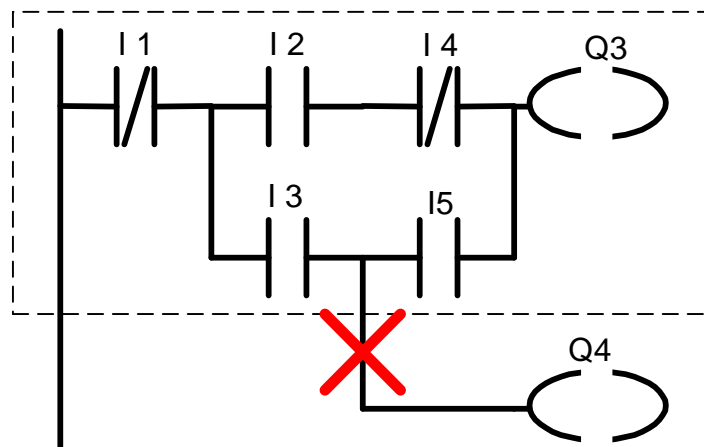
- Brak elementu wyjściowego



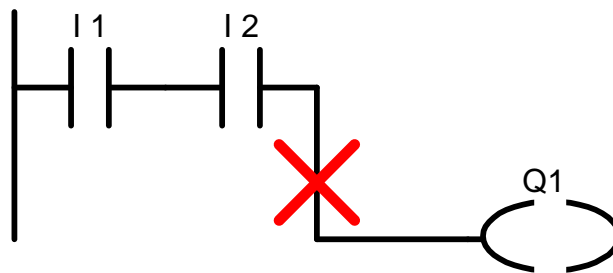
- Połączenie I3 przed obwodem



- Rozgałęzienie wewnątrz innego obwodu, którego jeden koniec (lub początek) łączy się z „linią zasilającą” lub wyjściami ( w poniższym przykładzie wyjście Q4 nie może być podłączone do rozgałęzienia I3 i I5).



- Brak elementu wyjściowego w linii I1, I2.



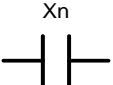
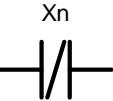
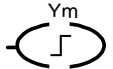
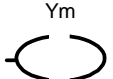
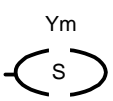
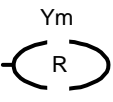
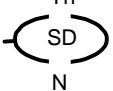
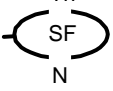
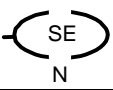
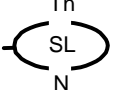
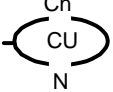
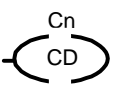
#### 5.2.6. Opis używanych elementów

Elementowi logicznemu (symbolowi – patrz tab. 5.2.6.), który pełni w języku LAD funkcję wejścia bądź wyjścia sygnału, mogą być przyporządkowane różne zmienne tzn. wejściem sygnału może być nie tylko napięcie podawane na sprzętowe wejścia (ozn. I1..I8), ale także wejściem może być stan *Timera*, *Licznika*, *Zegara* oraz stan *Wyjścia*. Decyduje o tym opis na symbolu elementu. Cyfra w oznaczeniu wskazuje numer wejścia, które ma być sprawdzane. Analogicznie możemy ustawiać (bądź kasować) nie tylko fizyczne wyjścia, ale także *Znaczniki* (wyjścia bez fizycznego wyprowadzenia) oraz stan *Timerów*, *Liczników* itp. W tabeli 5.2.6 przedstawiono symbole języka LAD z opisem i dopuszczalnymi zmiennymi XY sygnału dla danego elementu (X – wejścia, Y – wyjścia).

Wejście aktywne – wejście, którego stan pozwala na przepływ sygnału. (Logiczne ‘1’ dla wejścia NO, logiczne ‘0’ dla wejścia NC).

Wyjście aktywne – wyjście, które posiada sygnał logiczny ‘1’.

Tab. 5.2.6. Podstawowe symbole języka LAD.

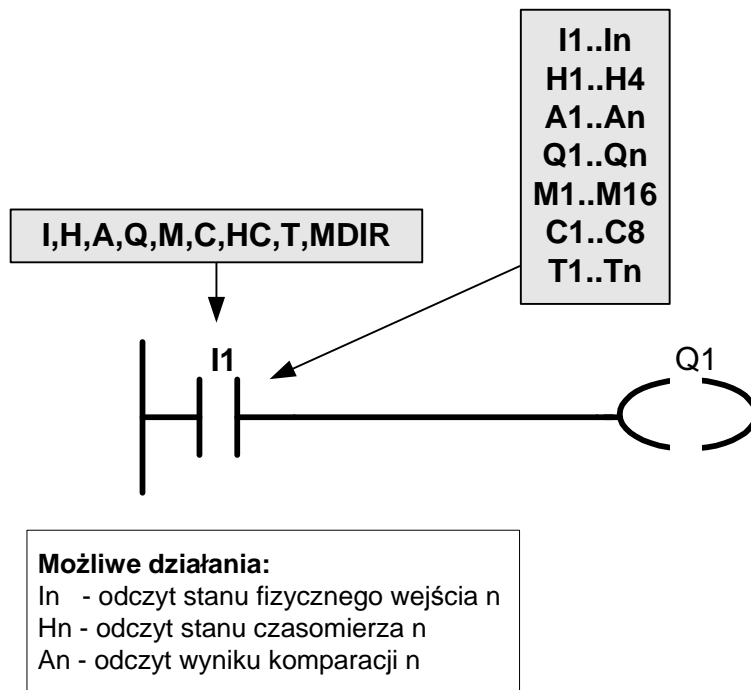
| LAD   | Opis  | Zmienna  |
|---|---|--|
|    | Wejście normalnie otwarte – NO.<br>Wejście aktywne (styk zwarty), gdy wartość logiczna przypisanej zmiennej wynosi '1'.<br><b>(Normally Open)</b>   | X: I, A, H, Q, M,<br>T, C<br>n: numer wejścia danego rodzaju |
|    | Wejście normalnie zamknięte – NC.<br>Wejście aktywne (styk rozarty), gdy wartość logiczna przypisanej zmiennej wynosi '0'.<br><b>(Normally Closed)</b>  |  |
|    | Przełącznik impulsowy – pełni rolę przerywacza wyzwanego zboczem narastającym. Każdy narastający impuls zmienia stan wyjścia na przeciwny. <b>(FP)</b>  | Y: Q, M<br>m: numer wyjścia danego rodzaju                   |
|    | Wyjście Przyporządkowujące<br>Ustawia wartość przypisanej zmiennej na '1', gdy podany zostanie do niego sygnał. Odpowiednik przełącznika o stykach otwartych (przepisanie stanu wejściowego na wyjście) <b>(Assign)</b>                                   | Y: Q, M<br>m: numer wyjścia danego rodzaju                   |
|    | Wyjście Set<br>Ustawia wartość przypisanej zmiennej na '1', gdy podany zostanie do niego sygnał i utrzymuje ten stan aż do momentu wykonania instrukcji 'Reset' lub do wyłączenia zasilania przełącznika programowalnego (przełącznik z podtrzymaniem)    | Y: Q, M<br>m: numer wyjścia danego rodzaju                   |
|  | Wyjście Reset<br>Ustawia wartość przypisanej zmiennej na '0', gdy podany zostanie do niego sygnał i utrzymuje ten stan aż do momentu wykonania instrukcji 'Set (S-STL)' lub do wyłączenia zasilania przełącznika programowalnego (kasowanie wyjścia)      |  |
|  | <i>Timer</i> Opóźnione załączenie<br>Ustawia wartość Tn = '1' po upływie zadanego czasu „N” od momentu uaktywnienia.  |  |
|  | <i>Timer</i> Opóźnione wyłączenie<br>Utrzymuje wartość Tn = '1' przez zadany czas „N” po zdjęciu sygnału aktywacji.   |  |
|  | <i>Timer</i> Pojedynczy impuls<br>Po uaktywnieniu zostaje wygenerowany pojedynczy impuls o czasie trwania „N”.  |  |
|  | <i>Timer</i> Impulsy<br>Jeśli aktywny – to generowana jest fala prostokątna (impulsy) o wypełnieniu 50% (czas trwania „N” stanu wysokiego i czas trwania „N” stanu niskiego impulsów).  |  |
|  | <i>Licznik w górę</i><br>Po uaktywnieniu zliczane są impulsy – stan <i>Licznika</i> jest zwiększany na, przypisanym do danego <i>Licznika</i> wejściu. Po osiągnięciu progu „N”, <i>Licznik</i> przyjmuje wartość '1'                                     |  |
|  | <i>Licznik w dół</i><br>Po uaktywnieniu zliczane są impulsy – stan <i>Licznika</i> jest zmniejszany na, przypisanym do danego <i>Licznika</i> wejściu. Po zmniejszeniu licznika wartości bieżącej poniżej progu „N”, <i>Licznik</i> przyjmuje wartość '0' |  |



## 5.2.7. Konfiguracja

## 5.2.7.1. Konfiguracja wejść

Każdemu wejściu w programie (obwodzie) musimy przyporządkować rodzaj i zmienną. Rodzaj przyporządkowujemy w sposób graficzny – wybierając styk normalnie otwarty lub styk normalnie zwarty, zmienną umieszczamy nad symbolem graficznym. Zmienna, która określa typ wejścia składa się z oznaczenia literowego i numeru.



Rys. 5.2.7.1. Konfiguracja wejść.

Do dyspozycji mamy następujące zmienne:

I – wejścia,

H – Zegary,

A – komparacje analogowe,

Q – stany wyjść,

M – stany Znaczników,

C – stany Liczników,

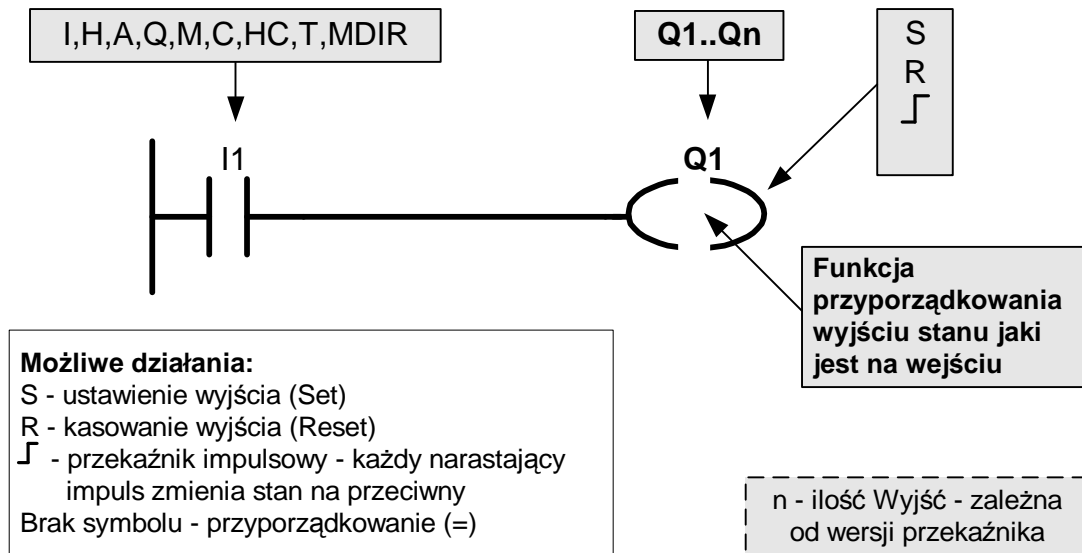
T – stany Timerów.

HC – szybki licznik/miernik częstotliwości 0-20 kHz.

MDIR – systemowy znacznik kierunku faz

## 5.2.7.2. Konfiguracja wyjść

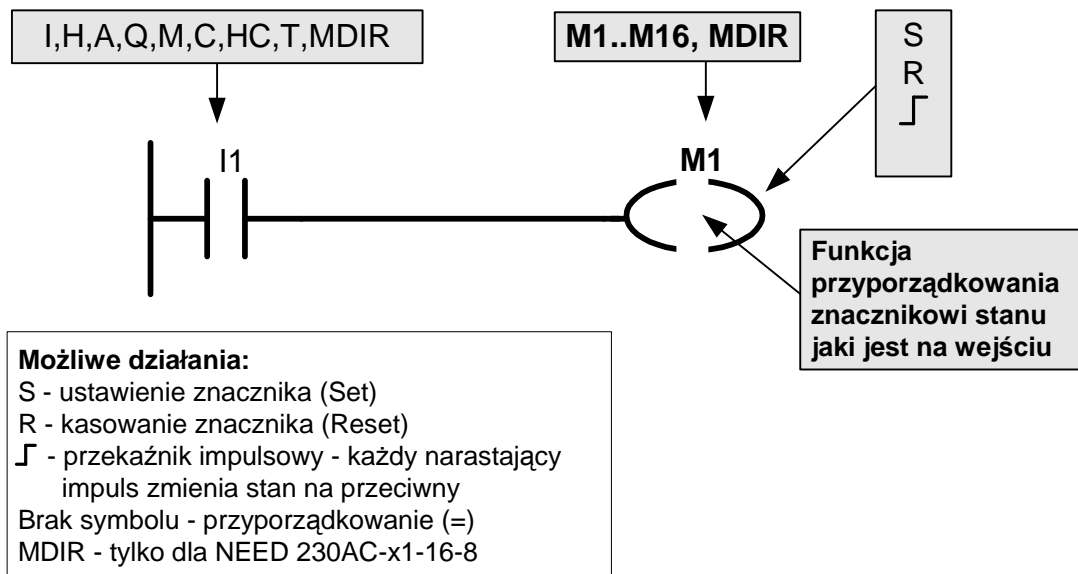
Fizyczne wyjścia, przedstawiamy za pomocą symbolu graficznego jak na rys. 5.2.7.2. W zależności od oczekiwanego zachowania wyjścia, stosujemy odpowiedni symbol graficzny. Nad symbolem graficznym umieszczamy literę Q, która oznacza wyjście oraz numer tegoż wyjścia.



Rys. 5.2.7.2. Konfiguracja wyjść.

## 5.2.7.3. Konfiguracja Znaczników

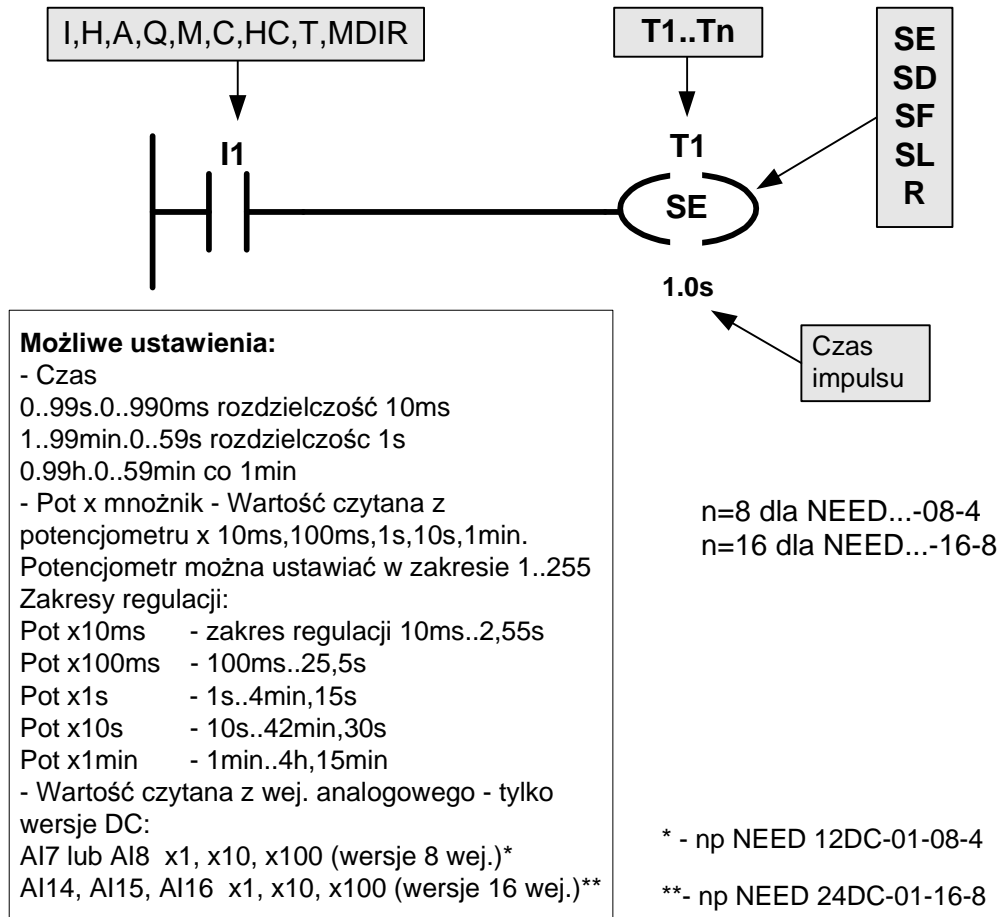
*Znaczniki*, podobnie jak wyjścia przedstawiamy za pomocą tego samego symbolu graficznego zamieniając Q na M (rys. 5.2.7.3). W zależności od oczekiwanej reakcji *Znacznika* stosujemy odpowiedni symbol wewnątrz graficznego oznaczenia. Nad symbolem graficznym umieszczamy literę M (ang. *Marker*) oraz numer.



Rys. 5.2.7.3. Konfiguracja Znaczników.

## 5.2.7.4. Konfiguracja Timerów

*Timery*, przedstawiamy za pomocą tego samego symbolu graficznego jak wyjścia – rys. 5.2.7.4. W zależności od oczekiwanego sposobu działania *Timera* stosujemy odpowiedni symbol wewnątrz graficznego oznaczenia *Timera*. Nad symbolem graficznym umieszczamy literę T oraz numer.



Rys. 5.2.7.4. Konfiguracja Timerów.

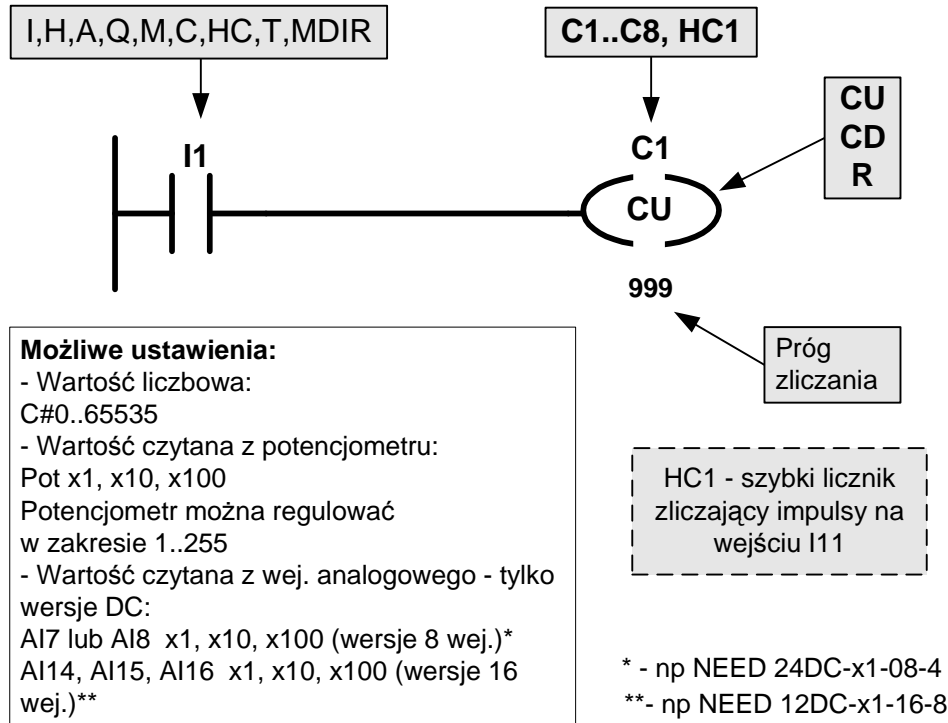


Istnieje możliwość używania, do odmierzenia czasów dla *Timerów*, wartości napięć czytanych z wejść analogowych I7, I8 dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub I14, I15, I16 dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8. Opisane dokładnie jest w rozdziale 5.1.2.21.1. Instrukcja 'L' dla Timerów.

## 5.2.7.5. Konfiguracja Liczników

*Licznik*, przedstawiamy za pomocą tego samego symbolu graficznego jak wyjścia – rys.

5.2.7.5. W zależności od oczekiwanego sposobu działania *Licznika* stosujemy odpowiednie symbole wewnątrz graficznego oznaczenia *Licznika*. Nad symbolem graficznym umieszczamy literę C, która oznacza *Licznik* (ang. Counter) oraz numer *Licznika*.



Rys. 5.2.7.5. Konfiguracja Liczników.



Istnieje możliwość używania, do ustawiania progu dla *Liczników*, wartości napięć czytanych z wejść analogowych I7, I8 dla NEED-12DC-x1-08-4, NEED-24DC-x1-08-4 lub I14, I15, I16 dla NEED-12DC-x1-16-8, NEED-24DC-x1-16-8. Opisane dokładniej jest w rozdziale 5.1.2.21.2. Instrukcja 'L' dla Liczników.



Wersje DC NEED..-x1-16-8 posiadają *Szybki Licznik* HC1 zliczający impulsy o maksymalnej częstotliwości 20kHz. HC1 jest *Licznikiem* sprzętowym, zlicza impulsy pojawiające się na wejściu I11. Wejścia CU, CD przejmują wtedy, oprócz funkcji kierunku zliczania, funkcję aktywacji *Szybkiego Licznika*.

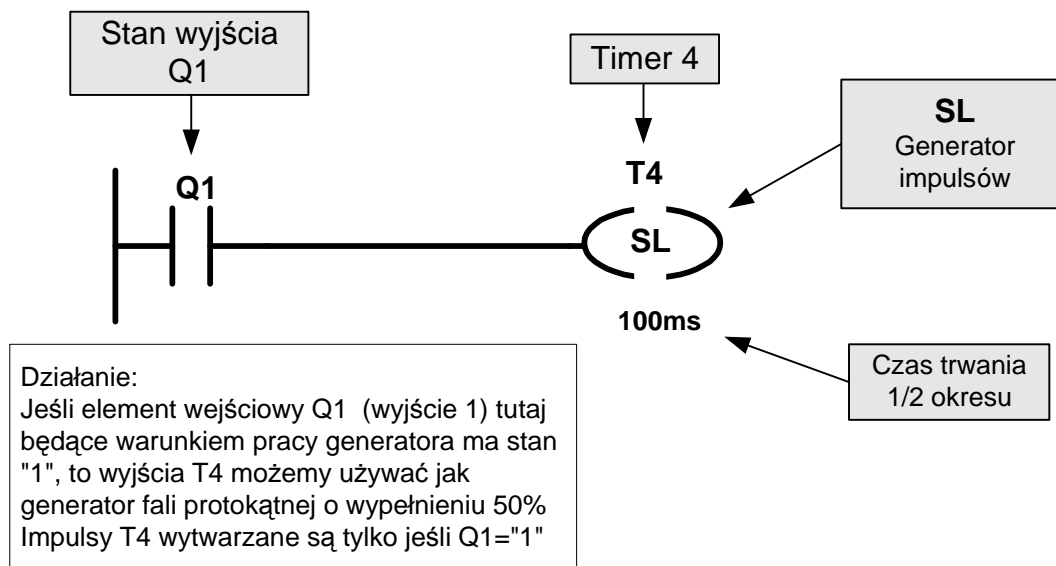
*Szybki Licznik* może pracować w trybie częstotliwościowym – zlicza ilość impulsów pojawiających się na wejściu I11 w czasie 1s.

*Szybki Licznik* nigdy się nie przepętnia. Próg do zliczania może być ustawiany w zakresie 0..65535. Wykonanie operacji Reset na *Szybkim Liczniku* kasuje stan i ilość zliczonych impulsów.

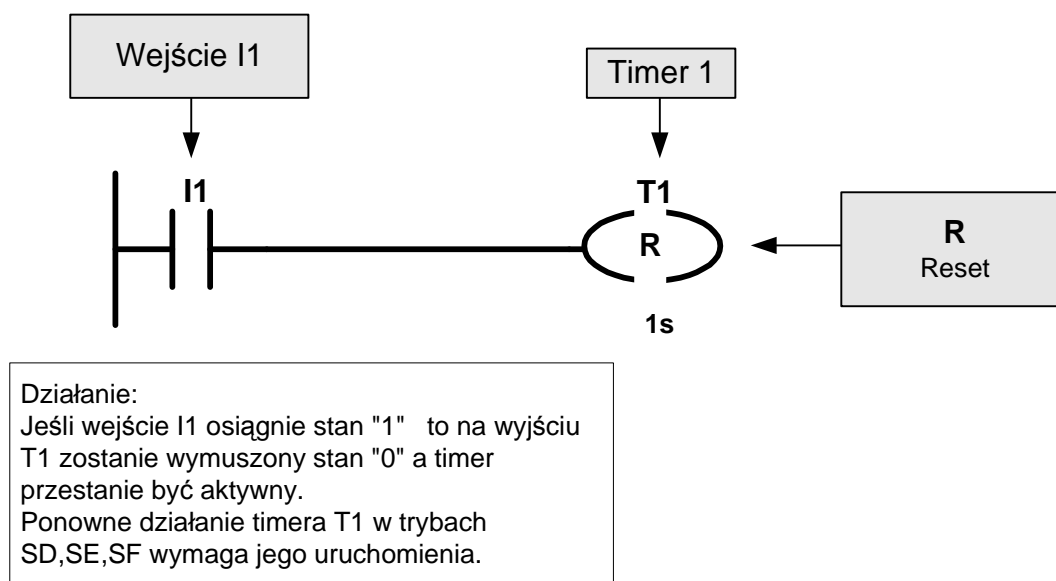


Dla wersji NEED-230AC-x1-16-8 *Szybki Licznik* HC1 mierzy częstotliwość sieci (50Hz lub 60Hz) jeżeli wejście I11 jest aktywne. Ponieważ częstotliwość sieci jest stała i znana można wykorzystać szybki licznik jako dodatkowy timer. Jeżeli próg = 1000 to dla 50Hz licznik przełączy się po  $100 \times 20\text{ms} = 20\text{s}$ .

## 5.2.7.6. Przykłady konfiguracji

Przykład1: *Timer SL Impulsy* (Generator impulsów)

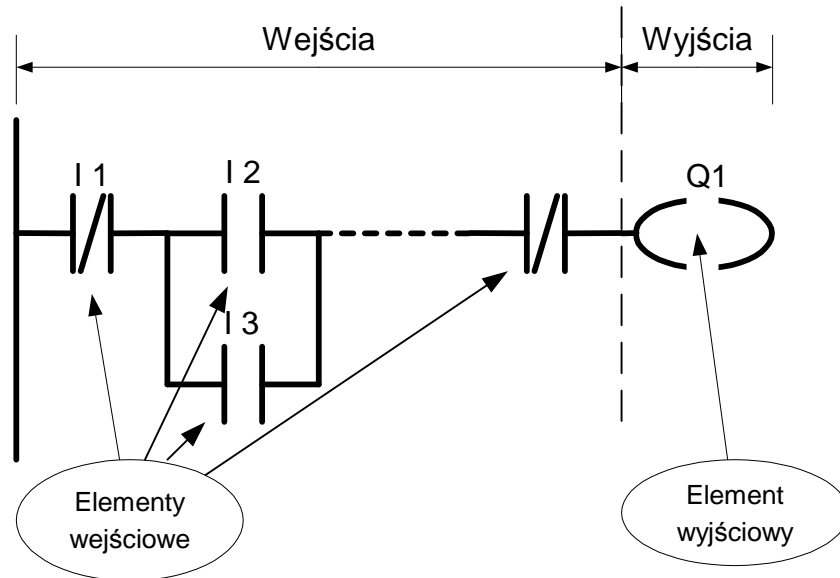
Rys. 5.2.7.6. Przykład konfiguracji Timera SL.

Przykład2: *Reset Timera*

Rys. 5.2.7.7. Przykład resetu Timera.

## 5.2.8. Zasady umieszczania elementów

Na rys 5.2.8.1. przedstawiono bardzo prosty obwód programu z rozmieszczeniem elementów zgodnie ze strukturą opisaną wcześniej. Dla ułatwienia na przykładach przedstawiane są dyskretne wejścia i wyjścia.

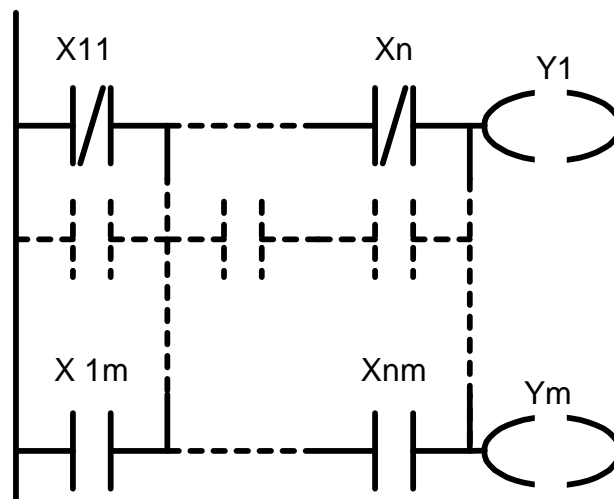


Rys. 5.2.8.1. Obwód w LAD.

Ogólnie obwód składa się z części wejściowej (warunkowej – poprzedzającej) i wyjściowej (wykonawczej – następującej). Pierwsza część określa warunki, jakie muszą być spełnione by zostało uaktywnione wyjście (element wykonawczy).

Elementy wejściowe mogą występować w różnych wzajemnych połączeniach, ich liczba podlega jedynie ograniczeniom ze względu na czytelność programu i możliwości edycyjne.

**Uwaga:** W przekaźniku NEED maksymalnie może być 150 wierszy (maksymalnie 862 bajty po kompilacji).



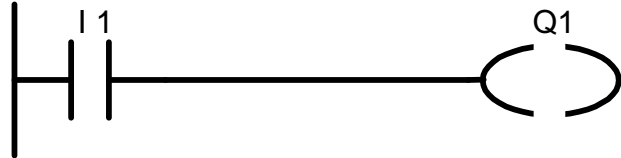
Rys. 5.2.8.2. Maksymalna ilość elementów jednego obwodu.

## 5.2.9. Rodzaje połączeń

Rozwiązanie układu sterowania wymaga stworzenia programu, który odpowiednio połączy zależności pomiędzy sygnałami wejściowymi i wyjściowymi.

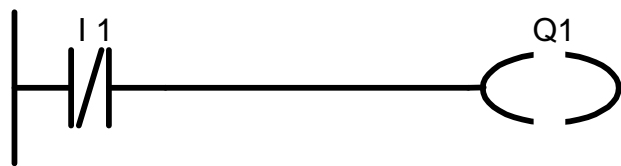
Podstawowe rodzaje połączeń przedstawiono poniżej.

## 5.2.9.1. Odwzorowanie wejścia na wyjście



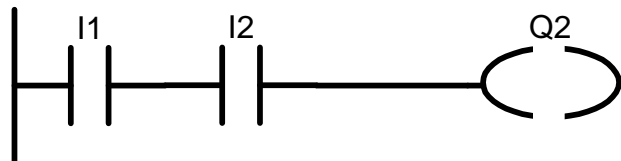
Stan wejścia I1 będzie przepisywany na wyjście Q1. Wyjście Q1 będzie aktywne ( $Q1='1'$ ), jeśli wejście I1 będzie w stanie logicznym '1'.

## 5.2.9.2. Odwzorowanie zanegowanego wejścia na wyjście



Zanegowany stan wejścia I1 będzie przepisywany na wyjście Q1. Wyjście Q1 będzie aktywne ( $Q1='1'$ ), jeśli wejście I1 będzie w stanie logicznym '0'.

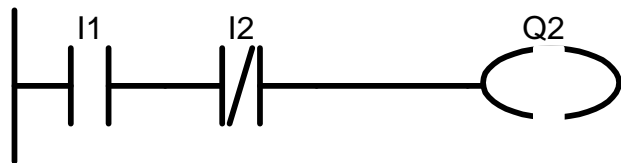
## 5.2.9.3. Połączenia szeregowe



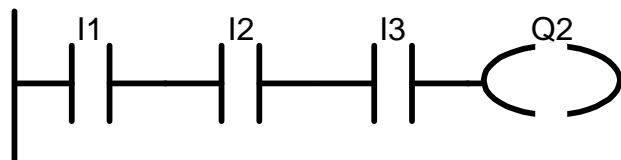
Przedstawiony powyżej układ realizuje funkcję iloczynu logicznego. Wyjście Q2 będzie aktywne ( $Q2='1'$ ), jeśli oba wejścia I1 i I2 będą w stanie logicznym '1'.

Poniżej przedstawiono inne rodzaje połączeń szeregowych

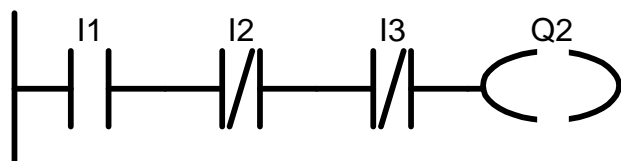
Wyjście Q2 będzie aktywne ( $Q2='1'$ ), jeśli wejście I1 będzie w stanie logicznym '1' a wejście I2 w stanie logicznym '0'.



Połączenie szeregowe 3 elementów. Wyjście Q2 będzie aktywne ( $Q2='1'$ ), jeśli wszystkie wejścia I1..I3 będą w stanie logicznym '1'.

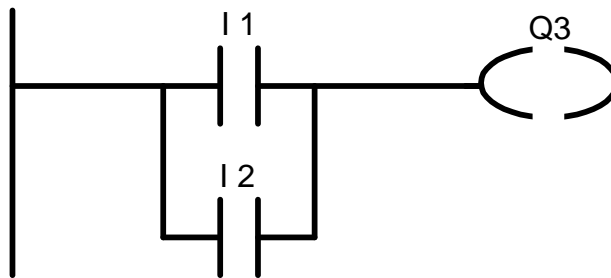


Połączenie szeregowe 3 elementów. Wyjście Q2 będzie aktywne ( $Q2='1'$ ), jeśli wejście I1 będzie w stanie logicznym '1' a wejścia I2, I3 będą w stanie logicznym '0'.



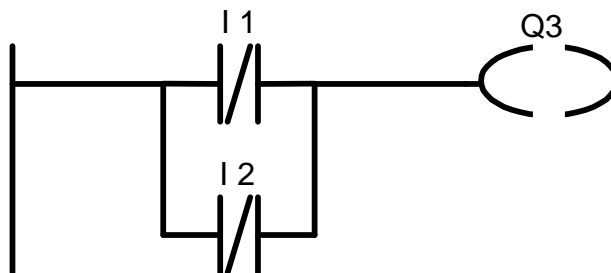
## 5.2.9.4. Połączenia równoległe

Przedstawiony obok układ realizuje funkcję sumy logicznej. Wyjście Q3 będzie aktywne ( $Q3=1$ ), jeśli jedno z wejść I1, I2 (lub oba) będzie w stanie logicznym '1'.

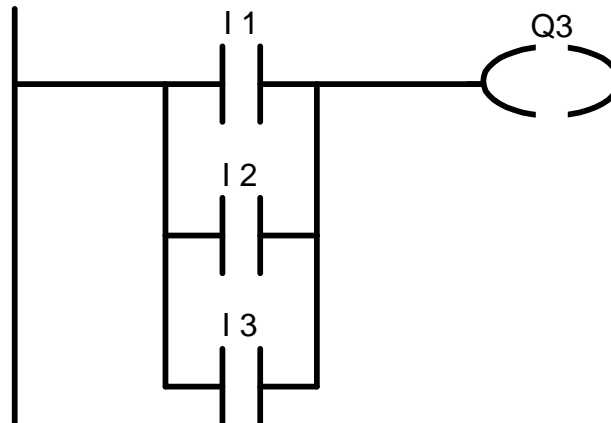


Poniżej przedstawiono inne rodzaje połączeń równoległych

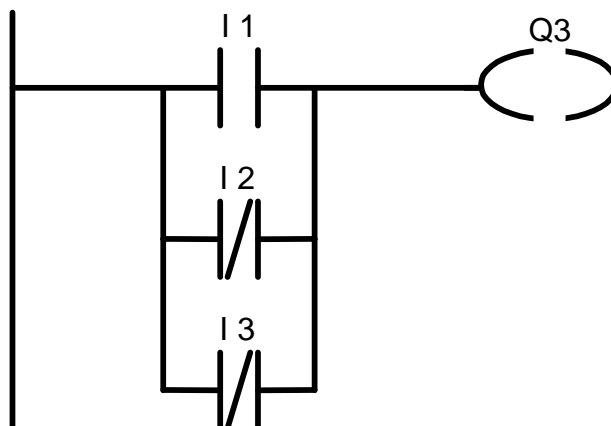
Wyjście Q3 będzie aktywne ( $Q3=1$ ), jeśli jedno z wejść I1, I2 (lub oba) będzie w stanie logicznym '0'.



Przedstawiony obok układ realizuje funkcję sumy logicznej 3 elementów. Wyjście Q3 będzie aktywne ( $Q3=1$ ), jeśli przynajmniej jedno z wejść I1, I2, I3 będzie w stanie logicznym '1'.



Suma logiczna 3 elementów. Wyjście Q3 będzie aktywne ( $Q3=1$ ), jeśli wejście I1 będzie aktywne (stan '1') lub gdy jedno z wejść I2, I3 (lub oba) będzie w stanie logicznym '0'.





## 5.2.9.5. Połączenie szeregowo-równoległe

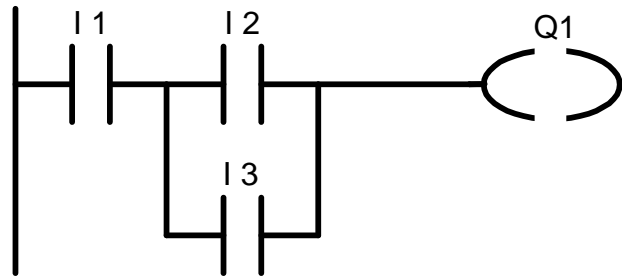
W celu przedstawienia układu sterowania opisane powyżej połączenia podstawowe można łączyć ze sobą w ramach dopuszczalnej ilości elementów wejściowych poziomych (3) i pionowych (150) zgodnie z zasadami tworzenia połączeń.

Jeśli algorytm wymaga użycia, dla wysterowania wyjścia, większej ilości elementów wejściowych, to należy odpowiednio zmodyfikować drabinkę połączeń używając do tego celu *Znaczników* tzn. podzielić zadania na mniejsze.

Poniżej przedstawiono przykłady obwodów wykorzystujących kombinacje połączeń szeregowo-równoległych wraz z interpretacją działania.

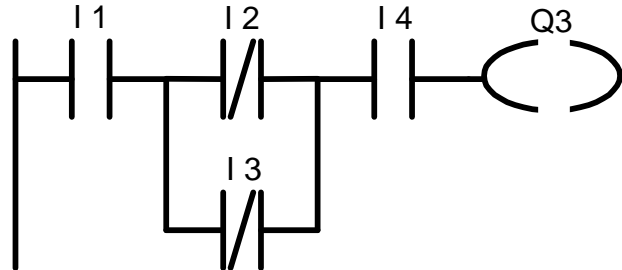
Układ realizujący połączenie szeregowe elementu I1 z równoległe połączonymi elementami I2, I3.

Działanie wyjścia Q1 jest następujące:  $Q1='1'$ , jeśli I1 jest aktywne (stan '1') i jedno z wejść I2, I3 (lub oba) jest w stanie logicznym '1'.

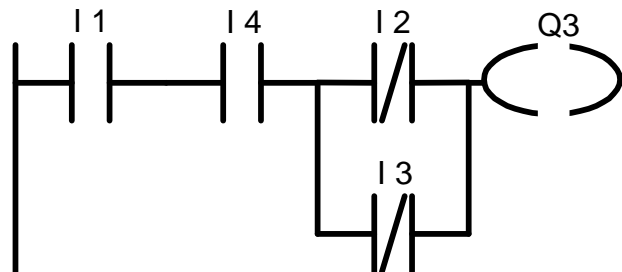


Układ realizujący połączenie szeregowe elementu I1 z równoległe połączonymi elementami I2, I3 oraz dalej szeregowo I4.

Działanie wyjścia Q3 jest następujące:  $Q3='1'$ , jeśli I1 i I4 jest aktywne (stan '1') i jedno z wejść I2, I3 (lub oba) jest nieaktywne (stan '0').



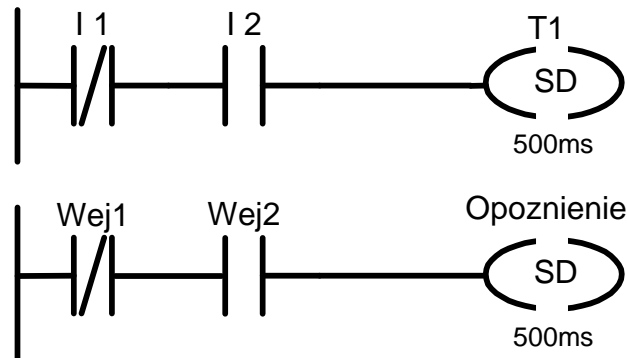
Równoważny powyżej układ przedstawiony jest w innej postaci: najpierw połączenie szeregowe I1, I4, a następnie połączenie równoległe I2 i I3.



## 5.2.10. Nazwy symboliczne

Dla przekaźników NEED istnieje możliwość przypisania nazw symbolicznych do zmiennych. Ułatwia to analizę programu i czyni go bardziej przejrzystym.

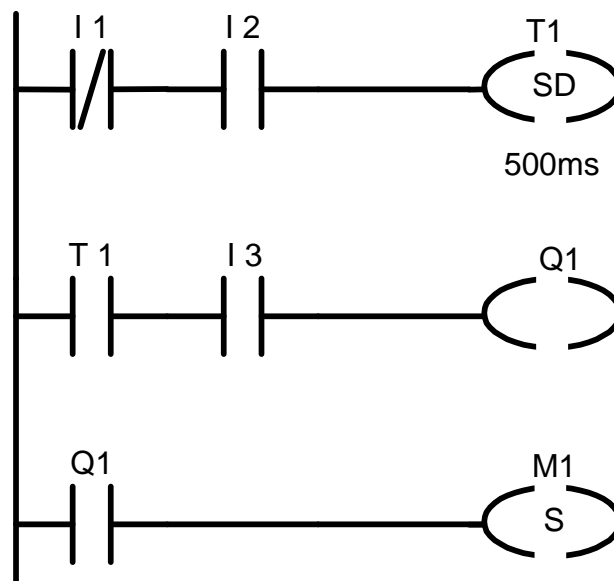
Możliwe jest przełączanie widoku zmienna/nazwa symboliczna. Na rys 5.2.10. przedstawiono obwód w zapisie zwykłym i poniżej z użyciem nazw symbolicznych.



Rys. 5.2.10. Przykład użycia symboli w LAD.

## 5.2.11. Program w LAD

Program składa się z obwodów, najprostszy program zawierać może tylko 1 obwód (linię programu). Poniżej przedstawiono program składający się z trzech obwodów.



Rys. 5.2.11. Przykładowy program zapisany w LAD.

Opis programu:

W pierwszym obwodzie jak na rys. 5.2.11. użyto wejść podłączonych bezpośrednio do przekaźnika programowalnego. Pierwsze wejście (I1) jest typu NC (normalnie zamknięte), drugie (I2) typu NO (normalnie otwarte), czyli załączenie *Timera* T1 nastąpi, jeśli I1 = '0' i I2 = '1'.

W drugim obwodzie (T1, I3, Q1) sprawdzamy stan *Timera* T1 (ustawianego w obwodzie 1) i wejścia I3. Jeśli *Timer* zostanie załączony (po czasie 500ms od spełnienia warunku I1='0' i I2='1') i wejście I3 będzie aktywne (I3 = '1'), to wyjście Q1 będzie w stanie wysokim (zasilane). Gdy wejście I3 zostanie wyłączone (I3 = '0') wyjście Q1 zostanie wyłączone.

Obwód 3 służy do zapamiętania załączenia wejścia Q1. Jeśli tylko wejście Q1 przyjmie stan „1”, to nastąpi ustawienie (na stałe) *Znacznika* M1 (M1 = '1').

Zauważmy, że powyższy program w zasadzie kończy się na ustawieniu *Znacznika* M1, gdyż dalsze operacje na *Znaczniku* M1 (np. resetowanie) nie są wykonywane.

## 6. INSTALACJA I OPIS OPROGRAMOWANIA

PC Need jest programem komputerowym, za pomocą którego można edytować, kompilować oraz ładować programy do pamięci przekaźnika programowalnego. Dodatkowo można także monitorować w trakcie pracy zasoby przekaźnika, dzięki czemu użytkownik może być na bieżąco informowany o stanach wejść, wyjść, *Timerach*, *Licznikach* itp. Pozwala to na całkowitą kontrolę aktualnie wykonywanego programu.

Prostota i różnorodność edycji programu (tekst lub grafika) sprawiają, iż PC Need jest bardzo wygodnym narzędziem, dzięki któremu nawet skomplikowane aplikacje powstają bardzo szybko, a czas ich uruchomienia jest krótki.

### 6.1. Wymagania sprzętowe

Dowolny komputer klasy PC ze złączem RS232 oraz z kartą graficzną VGA.

System operacyjny: Windows NT<sup>®</sup>, Windows 98<sup>®</sup>, Windows 2000<sup>®</sup>, Windows XP<sup>®</sup>.

### 6.2. Instalacja oprogramowania

1. Włóż płytę instalacyjną do napędu CD-ROM w komputerze.
2. Jeżeli program instalacyjny nie otworzy się automatycznie, to odszukaj na płycie plik „*setup.exe*” i dwa razy kliknij, aby rozpocząć instalację.
3. W czasie instalacji wskaż odpowiedni katalog, w którym zostanie zainstalowany PC Need. Jeżeli podczas instalacji została wybrana opcja umieszczenia ikon na pulpicie, to po właściwym zakończeniu instalacji na pulpicie powinna znaleźć się ikona programu. PC Need można także uruchomić z paska zadań.

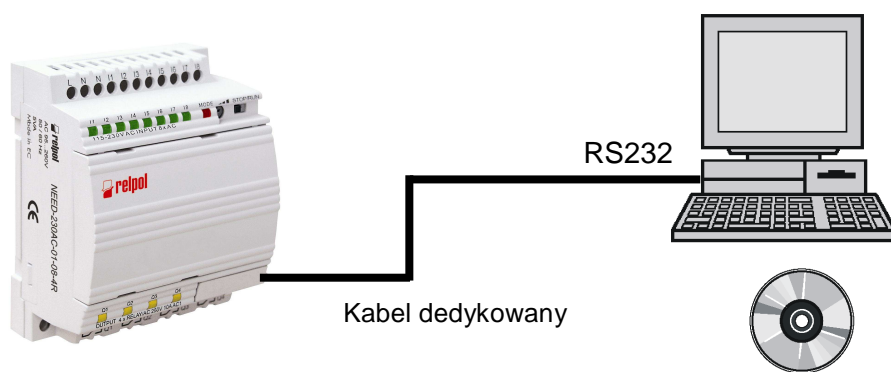
### 6.3. Deinstalacja

Aby usunąć program, należy użyć automatycznej deinstalacji: przycisk **Start > Programy > Relpol > Deinstalacja programu PC Need**.

Po wybraniu tej opcji program PC Need zostanie automatycznie usunięty z systemu.

### 6.4. Połączenie komputera z przekaźnikiem programowalnym

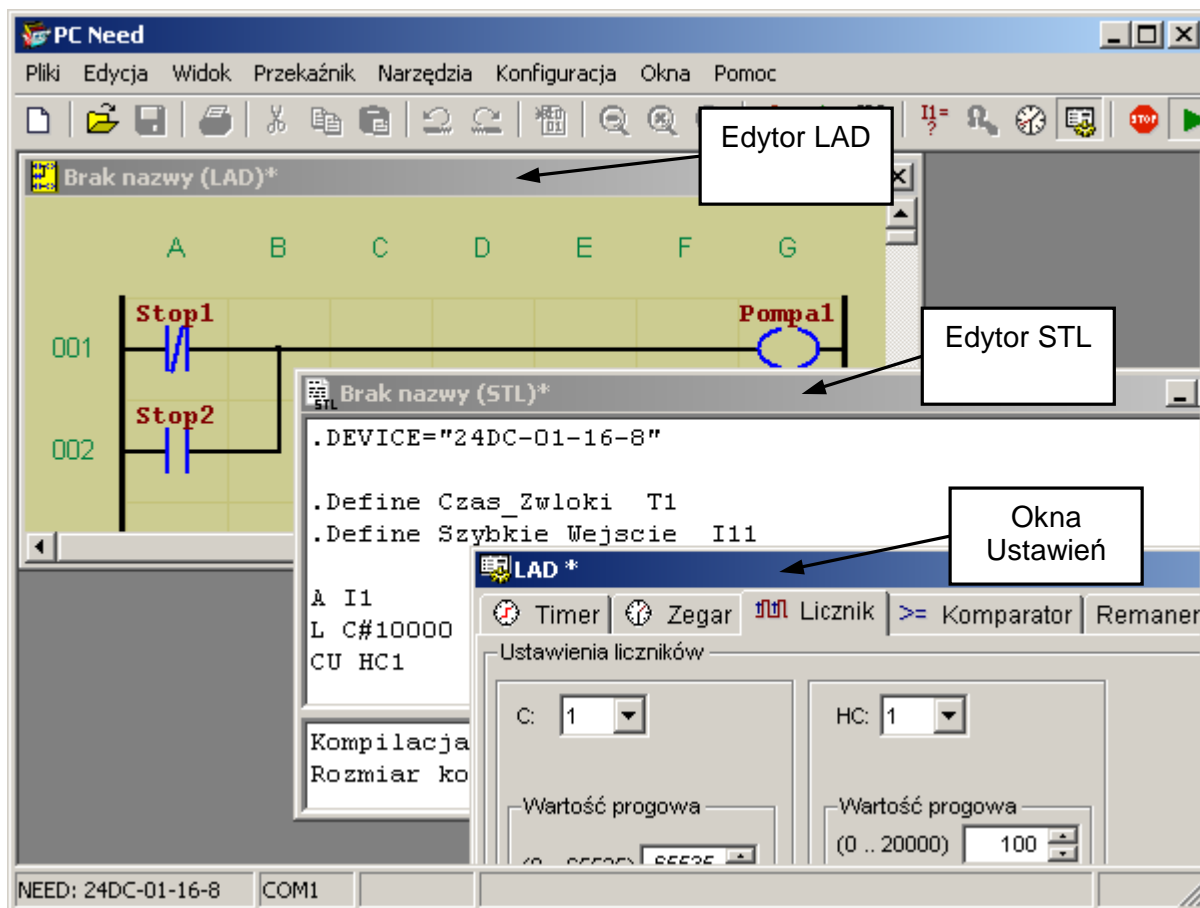
Komputer należy połączyć z przekaźnikiem programowalnym za pomocą dedykowanego kabla.



Rys. 6.4. Połączenie przekaźnika programowalnego NEED z komputerem.



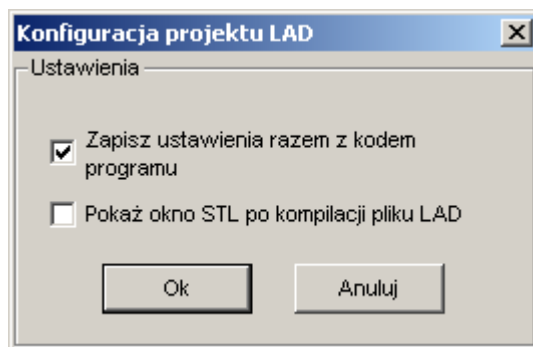
Odwrotne podłączenie zasilania tzn. zamienienie miejscami przewodów fazowego (L) i neutralnego (N) na wejściach zasilających przekaźnika programowalnego może spowodować wystąpienie niebezpiecznych napięć na porcie komunikacyjnym przekaźnika.



Rys. 6.5.1. Okna programu PC Need.

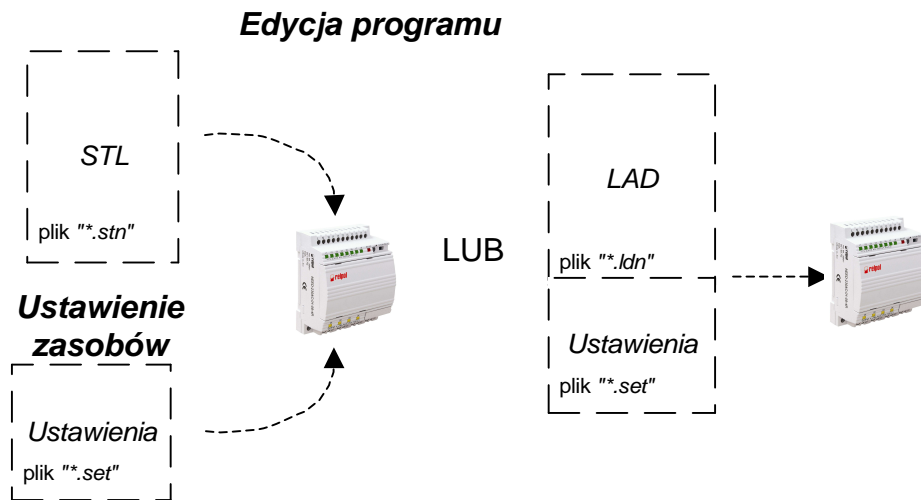
Projekt dla przekaźnika NEED powinien zawierać przynajmniej jeden plik z rozszerzeniem „\*.stn” lub „\*.ldn” (program użytkownika). Jeżeli programista korzysta z takich zasobów przekaźnika programowalnego jak *Zegary* czy *Komparatory*, wtedy oprócz pliku źródłowego (program STL lub LAD) do pamięci przekaźnika NEED należy załadować także ustawienia przekaźnika, edytowane w oknie ustawień – rys. 6.5.1.

Dla edytora LAD można w opcji, **Konfiguracja > Projekt LAD**, ustawić opcję **Zapisz ustawienia razem z kodem programu** (domyślnie opcja ta jest załączona – rys. 6.5.2.). Po zaznaczeniu tej opcji do pamięci przekaźnika załadowany zostanie plik programu „\*.ldn” oraz okno ustawień.



Rys. 6.5.2. Okna konfiguracji projektu LAD.

Na rys. 6.5.3. przedstawiono schematycznie zawartość projektu dla przekaźnika programowalnego NEED.



Rys.6.5.3. Pliki zawarte w projekcie dla przekaźnika programowalnego NEED.

Jeśli programista korzysta z *Zegarów*, *Komparatorów*, remanencji to, ustawienia powinny być zawsze ładowane do pamięci przekaźnika.

Przykład:

Projekt: program STL bez używania takich zasobów przekaźnika jak *Zegary*, remanencje, *Komparatory* itp. – rys. 6.5.4.

```

STL D:\Programy\RelpolPC\Fast Start.stn*
A I1
S Q1
R M1

A Q1
FP Q2
//Koniec programu

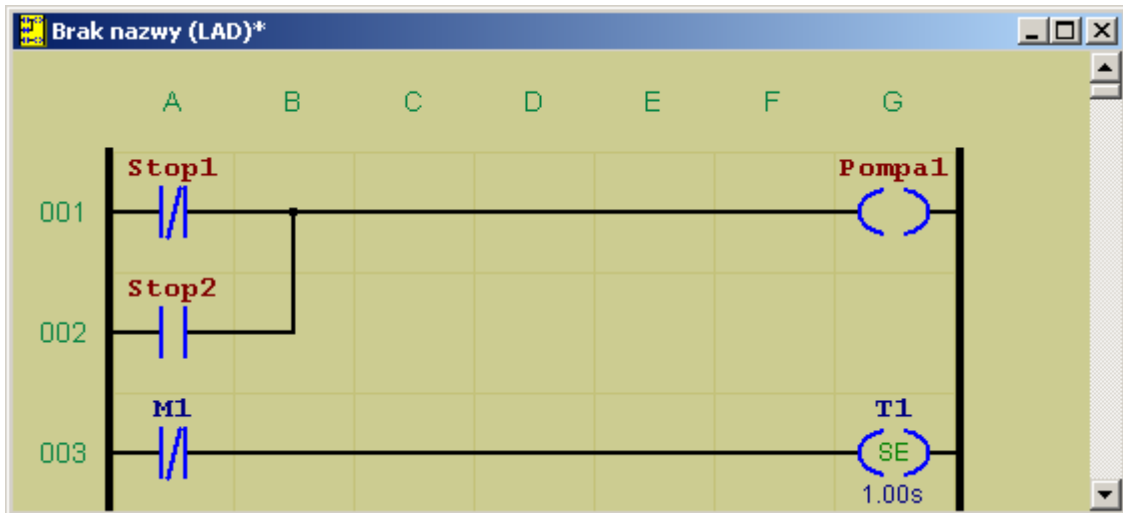
Kompiluje plik: D:\Programy\RelpolPC\Fast Start.stn...
Kompilacja zakończona
Rozmiar kodu 11 (max 862). Zajętość pamięci 1.3%
    
```

Rys.6.5.4. Program STL.

Do załadowania tylko plik „\*.stn”.

Przykład:

Projekt: program LAD. Zaznaczona opcja „**Zapisz ustawienia razem z kodem programu**”



Rys.6.5.5. Program LAD.

Po wyborze z Menu: **Transmisja > Zapis do przekaźnika** do pamięci przekaźnika programowalnego zostanie zapisany program LAD (plik „\*.ldn”) oraz ustawienia.

Przykład:

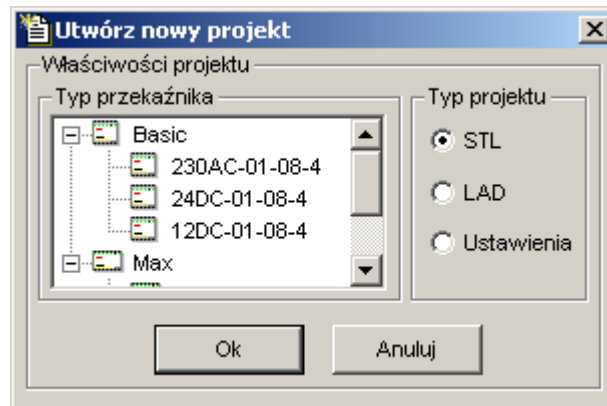
Projekt: program STL , w którym wykorzystany jest Zegar i Komparator.

Rys.6.5.6. Program STL z wykorzystaniem Zegara i Komparatora.

Do załadowania plik „Przykład2.stn” oraz Ustawienia „Przykład2.set” przekaźnika.

## 6.5.1. Przykład projektu – program STL

1. Uruchom program PC Need.
2. Utwórz nowy projekt np.: **Plik > Nowy** – w okienku wyboru (rysunek poniżej) wybierz odpowiedni typ przekaźnika oraz zaznacz w polu „Typ projektu” – **STL**.



3. Napisz program np. taki jak poniżej:

```

C:\NEED_Source\Szybki_Start.stn
.DEVICE="24DC-01-16-8"

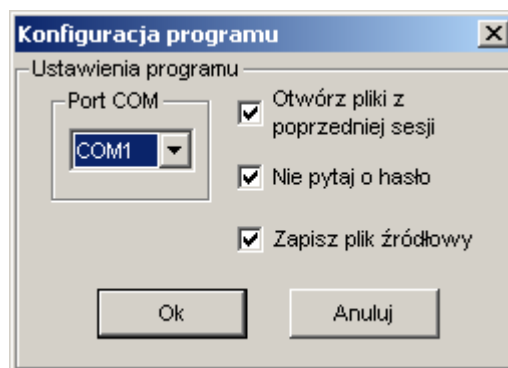
A I1
L 1S
SL T1

A T1
=Q1

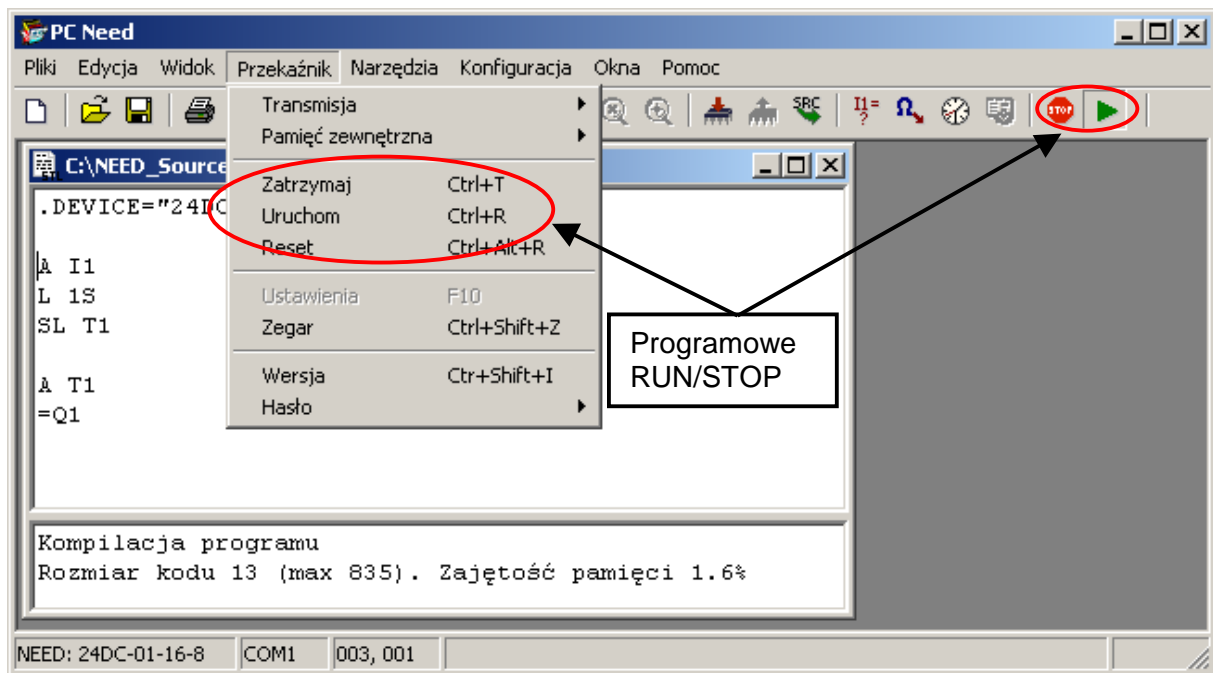
Kompilacja programu
Rozmiar kodu 13 (max 835). Zajętość pamięci 1.6%

```

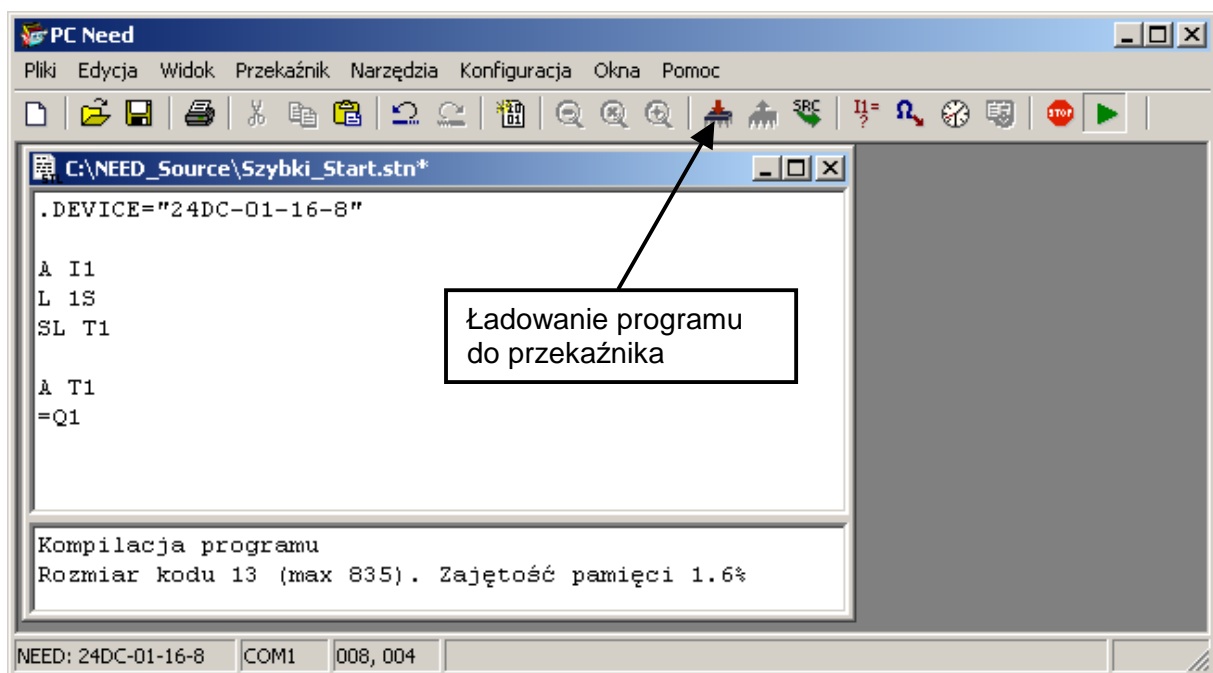
4. Połącz komputer z przekaźnikiem oraz dokonaj konfiguracji portu szeregowego RS232: **Konfiguracja → Program → COM** – wybierz odpowiedni (wolny) port. Operację tą należy powtarzać tylko przy pierwszym uruchomieniu lub, gdy chcemy zmienić port komunikacyjny.



5. Ustaw przełącznik w tryb STOP: (przełącznik lub **Przełącznik > Zatrzymaj**).



6. Załaduj program do pamięci przełącznika: **Przełącznik > Transmisja > Zapis do przełącznika**.



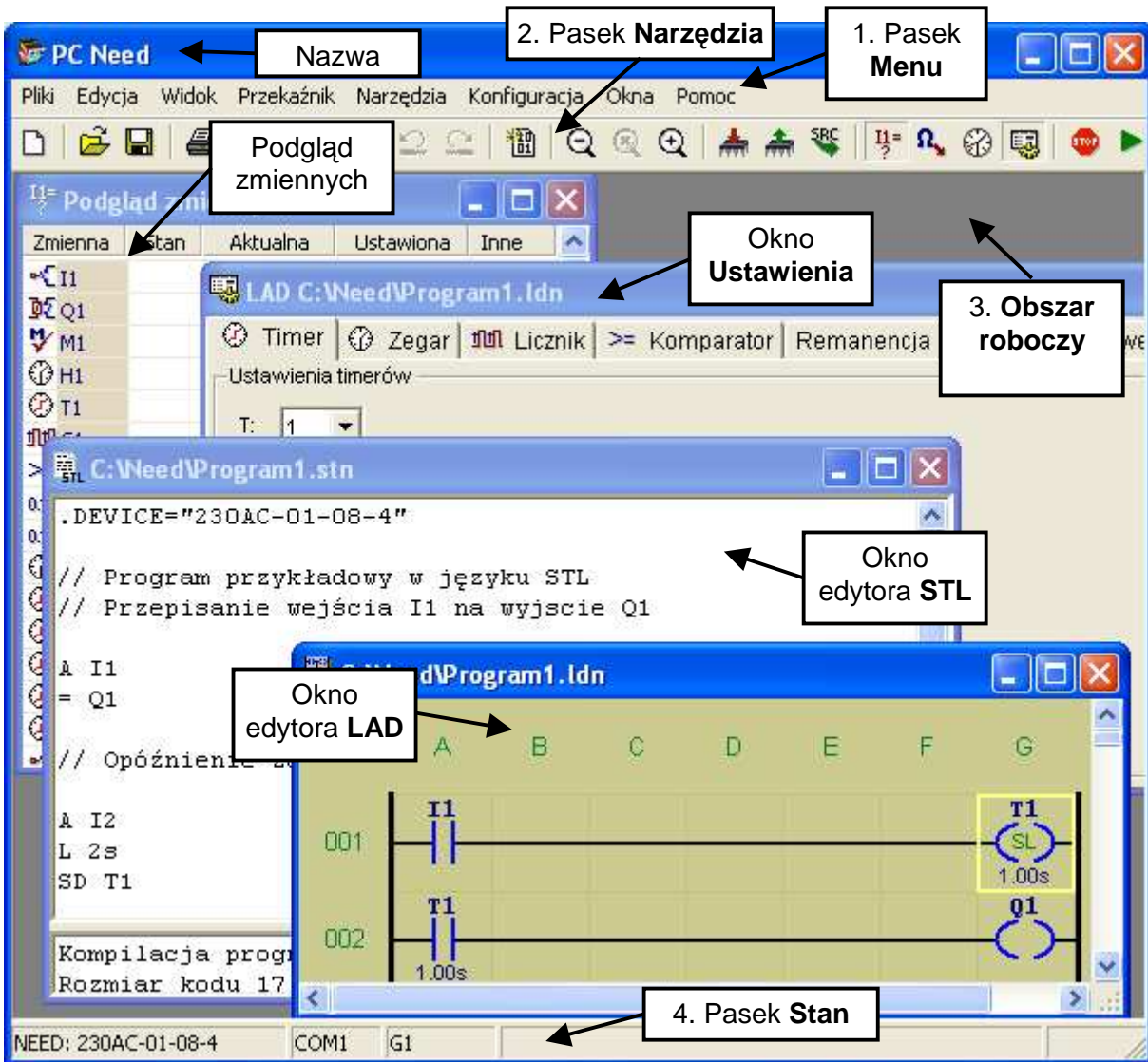
7. Przełącz przełącznik w tryb RUN (przełącznik lub **Przełącznik > Uruchom**) i podaj na wejście I1 sygnał (stan wysoki). Wyjście Q1 powinno migać (cykliczne 1-sekundowe załączenie i 1-sekundowe wyłączenie).



## 6.6. Praca z programem PC Need

### 6.6.1. Opis głównego okna programu

Po uruchomieniu programu PC Need otwiera się okno – interfejs użytkownika.



Rys. 6.6.1. Interfejs użytkownika PC Need.

Interfejs użytkownika tworzą:

#### 1. Pasek Menu.

Pliki Edycja Widok Przekaznik Narzędzia Konfiguracja Okna Pomoc

#### 2. Pasek Narzędzia.



3. Obszar roboczy – w nim otwierane okna, między innymi: **Edytor LAD, Edytor STL, Ustawienia, Podgląd zmiennych, Konfiguracja, Ustawienia elementu.**

#### 4. Pasek Stan.

NEED: 24DC-01-08-4 COM1 A1 Rozmiar kodu 1 (max 862). Zajętość pamięci 0.1%

## 6.6.2. Pasek Menu

**Pliki** – zarządzanie operacjami na plikach

- > **Nowy** – otwarcie okna wyboru typu przekaźnika i projektu: STL, LAD, Ustawienia
  - >> **STL** – utworzenie nowego pliku w edytorze języka STL
  - >> **LAD** – utworzenie nowego projektu w edytorze języka LAD
  - >> **Ustawienia** – utworzenie nowego pliku ustawień SET
- > **Otwórz** – otwarcie istniejącego pliku do edycji lub zmiany ustawień; otwierane pliki:
  - \*.stn – pliki zapisane w języku tekstowym STL
  - \*.ldn – pliki zapisane w języku drabinkowym LAD
  - \*.set – pliki ustawień (SET)
  - („\*” - nazwa pliku; .stn rozszerzenie - rodzaj pliku)
- > **Zapisz** – zapisanie pliku na dysku
- > **Zapisz jako** – zapisanie pliku na dysku poprzez utworzenie nowego pliku
- > **Dokument** – informacje o tworzonej programie (do tabeli wydruku)
- > **Drukuj** – wydruk dokumentu
- > **Zakończ** – zakończenie pracy z programem NEED

**Edycja** – polecenia edycji programu

- > **Cofnij**: cofnięcie ostatniej operacji
- > **Powtórz**: powtórzenie ostatnio wykonanego cofnięcia
- > **Wytnij**: wycięcie zaznaczonej zawartości
- > **Kopiuuj**: skopiowanie zaznaczonej zawartości
- > **Wklej**: wstawienie zawartości w zaznaczone miejsce
- > **Wyczyść**: usunięcie zaznaczonej zawartości
- > **Szukaj**: okno wyszukiwania (STL, LAD)

**Widok** – ustawianie parametrów okna programu NEED – aktywne, jeśli edytowany (otwarty) program w języku drabinkowym LAD

- > **Powiększenie**: dostosowanie obszaru w oknie edytora LAD
  - >> **Powiększ** – powiększenie rozmiaru
  - >> **Pomniejsz** – pomniejszenie rozmiaru
  - >> **Normalny** – rozmiar domyślny
  - >> **Okno STL** – wyświetla okno STL z wynikowym kodem kompilacji języka LAD.

**Przekaźnik** – zestaw funkcji do obsługi przekaźnika

- > **Transmisja** – obsługa komunikacji z przekaźnikiem
  - >> **Zapis do przekaźnika** – kompilacja programu i przesłanie kodu wykonawczego programu do przekaźnika lub zapis nowych ustawień (w zależności od aktywnego aktualnie okna)
  - >> **Odczyt z przekaźnika** – odczyt ustawień z przekaźnika, lub odczyt wartości dla podglądu zmiennych (w zależności od aktywnego aktualnie okna)
  - >> **Odczyt kodu programu** – odczyt kodu źródłowego z przekaźnika
  - >> **Porównaj** – porównanie kodu aktualnego programu zapisanego na dysku, z programem zapisanym w przekaźniku (porównanie z aktywnym otwartym programem w edytorze)
- > **Pamięć zewnętrzna** – obsługa modułu pamięci
  - >> **Zapis** – zapis aktualnego programu lub ustawień do modułu pamięci. Zapisywany jest aktualnie otwarty plik programu lub ustawień. Jeżeli chcemy wgrać program i ustawienia, to należy przeprowadzić zapis dwukrotnie, raz dla aktywnego okna programu, raz dla aktywnego okna pliku ustawień

- >> **Odczyt** – odczyt ustawień z modułu pamięci
- >> **Status** – informacje o stanie partycji pamięci oraz deaktywacja partycji
- > **Zatrzymaj** – polecenie wprowadzające przełącznik w tryb STOP
- > **Uruchom** – polecenie wprowadzające przełącznik w tryb RUN
- > **Reset** – kasowanie pamięci programu, ustawień i hasła w przełączniku
- > **Ustawienia** – otwarcie okna edycji ustawień
- > **Zegar** – otwarcie okna zarządzania czasem w przełączniku
- > **Wersja** – informacja o typie i wersji oprogramowania przełącznika
- > **Hasło** – zabezpieczenie przed odczytem i zapisem programu w przełączniku programowalnym
  - >> **Wprowadź** – wprowadzanie hasła do weryfikacji z hasłem w przełączniku
  - >> **Zmień** – zmiana istniejącego hasła z jego weryfikacją

**Narzędzia** – zestaw funkcji do uruchamiania aplikacji

- > **Kompilacja** – kompilacja programu
- > **Ustawienia elementu** – otwarcie okna z parametrami styku (edytor LAD)
- > **Podgląd zmiennych** – otwarcie okna do odczytu aktualnych wartości zmiennych w przełączniku. Uaktywnienie podglądu **Przełącznik > Uruchom**. W tabeli należy wpisać, które zmienne mają podlegać odczytowi.
- > **Nazwy symboliczne** – zmiana sposobu wyświetlania zmiennych - rejestry/nazwy symboliczne

**Konfiguracja** – opcje programu PC Need

- > **Program** – otwarcie okna – wybór portu komunikacyjnego, decyzja o otwieraniu plików z poprzedniej sesji, rezygnacja z zabezpieczania hasłem.
- > **Projekt LAD** – otwarcie okna – opcje zapisu kodu programu i ustawień oraz decyzja o otwieraniu okna STL po kompilacji programu LAD
- > **Język** – otwarcie okna wyboru języka interfejsu programu.

**Okna** – zarządzanie otwartymi oknami w obszarze roboczym programu NEED























- > **Kaskadowo** – rozmieszczenie okien jedno nad drugim
- > **Sąsiadująco** – rozmieszczenie okien obok siebie
- > **Informacja o oknach**

**Pomoc** – plik pomocy i informacje o programie

- > **Indeks** – tematy pomocy
- > **PC Need – informacje** – o wersji programu i producencie.

## 6.6.3. Pasek narzędzi

Dla najczęściej używanych opcji menu dostępne są przyciski na pasku narzędzi umożliwiające szybsze otwarcie danej funkcji programu po kliknięciu na wybrany przycisk. Poniżej w skrócie opis przycisków na pasku narzędzi.

|   |                          |   |
|---|--------------------------|---|
|    | <b>Nowy</b>              | Otwiera nowy projekt (plik)   |
|    | <b>Otwórz</b>            | Otwiera istniejący projekt (plik)   |
|    | <b>Zapisz</b>            | Zapisuje aktywny projekt  |
|    | <b>Drukuj</b>            | Drukuje aktywny projekt   |
|    | <b>Wytnij</b>            | Wycina zaznaczenie  |
|    | <b>Kopiuj</b>            | Kopiuje zaznaczenie   |
|    | <b>Wklej</b>             | Wkleja zaznaczenie  |
|    | <b>Cofnij</b>            | Cofa ostatnią operację  |
|    | <b>Ponów</b>             | Ponawia cofniętą operację   |
|   | <b>Kompilacja</b>        | Kompilacja aktywnego projektu   |
|  | <b>Pomniejsz</b>         | Pomniejsza rozmiar zawartości okna (LAD, Podgląd zmiennych)                       |
|  | <b>Widok normalny</b>    | Ustawia widok domyślny (LAD, Podgląd zmiennych)                                   |
|  | <b>Powiększ</b>          | Powiększa rozmiar zawartości okna (LAD, Podgląd zmiennych)                        |
|  | <b>Zapis</b>             | Zapis (transmisja) do przekaźnika (LAD, STL, SET)                                 |
|  | <b>Odczyt</b>            | Odczyt (transmisja) z przekaźnika (LAD, SET, Podgląd drabinki, Podgląd zmiennych) |
|  | <b>Odczyt źródła</b>     | Odczyt z przekaźnika programu źródłowego  |
|  | <b>Podgląd zmiennych</b> | Odczyt stanu wybranych zmiennych z przekaźnika                                    |
|  | <b>Nazwy symboliczne</b> | Przełączanie widoku adres/symbol  |
|  | <b>Zegar</b>             | Otwiera okno zegara czasu rzeczywistego (RTC)                                     |
|  | <b>Ustawienia</b>        | Otwiera okno ustawień   |
|  | <b>Stop</b>              | Ustawienie trybu STOP w przekaźniku (Stop programu)                               |
|  | <b>Run</b>               | Ustawienie trybu RUN w przekaźniku (Start programu)                               |

## 6.6.4. Skróty klawiaturowe

PC Need umożliwia wykonywanie większości poleceń za pomocą klawiatury. Poniżej przypisane klawiszom bądź ich kombinacjom funkcje.

**PC Need**

|                    |   |
|--------------------|---|
| F1                 | Wyświetl "Help'a"   |
| F3                 | Szukaj następnego wystąpienia słowa (w STL) lub elementu w LAD. Wcześniej powinno być wyświetlone okno „Szukaj” |
| F5                 | Zapisz dane do przekaźnik (STL lub LAD lub/i SET)   |
| F6                 | Czytaj dane z przekaźnika (SET)   |
| F7                 | Kompilacja  |
| F10                | Wyświetl okno ustawień (dla aktywnego okna LAD)   |
| F12                | Wyświetl okno podglądu zmiennych  |
| CTRL + "N"         | Okno "Nowy projekt"   |
| CTRL + "O"         | Okno "Otwórz plik"  |
| CTRL + "S"         | Zapisz plik w aktywnym oknie  |
| CTRL + "P"         | Print - Wydruk dokumentu  |
| ALT + F4           | Zakończenie pracy z programem NEED  |
| CTRL + "Z"         | Operacja "Cofnij"   |
| CTRL + "Y"         | Operacja "Powtórz"  |
| CTRL + "X"         | Operacja "Wytnij"   |
| CTRL + "C"         | Operacja "Copy"   |
| CTRL + INSERT      | Operacja "Copy"   |
| CTRL + "V"         | Operacja "Wklej"  |
| SHIFT + INSERT     | Operacja "Wklej"  |
| CTRL + NUM+        | Powiększenie rozmiaru zawartości okna (Zwiększ zoom)  |
| CTRL + NUM-        | Pomniejszenie rozmiaru zawartości okna (Zmniejsz zoom)  |
| CTRL + NUM*        | Rozmiar domyślny zawartości okna (Zoom "normalny")  |
| CTRL + T           | Ustaw tryb STOP w przekaźniku   |
| CTRL + R           | Ustaw tryb RUN w przekaźniku  |
| CTRL + ALT + "R"   | Reset przekaźnika   |
| SHIFT + CTRL + "Z" | Otwarcie okna zarządzania czasem w przekaźniku  |
| CTRL + SHIFT + "I" | Informacja o typie i wersji oprogramowania w przekaźniku  |

|                    |  |
|--------------------|--|
| SHIFT + CTRL + "A" | Wyświetl okno "Ustawienie elementu" (dla aktywnego okna LAD) |
| ALT + "/"          | Wyświetl okno "PCNeed - Informacje"                          |

**Edytor LAD**

|              |   |
|--------------|---|
| A            | Wstaw Komparator  |
| C            | Wstaw Licznik   |
| D            | Wstaw MDIR  |
| H            | Wstaw Zegar   |
| I            | Wstaw Wejście   |
| M            | Wstaw Znacznik  |
| Q            | Wstaw Wyjście   |
| T            | Wstaw Timer   |
| SHIFT + "C"  | Wstaw Szybki licznik (HC)   |
| CTRL + "A"   | Zaznacz cały schemat LAD  |
| SPACJA       | Rysuj "linię" poziomą   |
| ENTER        | Pokaż okno "Ustawienia elementu"  |
| ALT lewy + ← | Jeśli aktualnie wybrany jest jakiś element to strzałki LEWO i PRAWO zmieniają funkcje NO/NC FP,SE, SD, SF,...   |
| ALT lewy + → |   |
| ALT lewy + ↑ | Jeśli aktualnie wybrany jest jakiś element to strzałki GÓRA i DÓŁ zmieniają numer rejestru.<br>Jeśli aktualnie nie jest wybrany żaden element to strzałki umożliwiają rysowanie połączeń. |
| ALT lewy + ↓ |   |
| HOME         | Działają tak jak w edytorach tekstowych.<br>Można dodatkowo używać je razem SHIFT'em do zaznaczania.  |
| END          |   |
| Page Up      |   |
| Page Down    |   |

Objaśnienie:

F2 – naciśnięcie klawisza funkcyjnego F2

CTRL+S – jednoczesne naciśnięcie klawisza Ctrl i Klawisza S

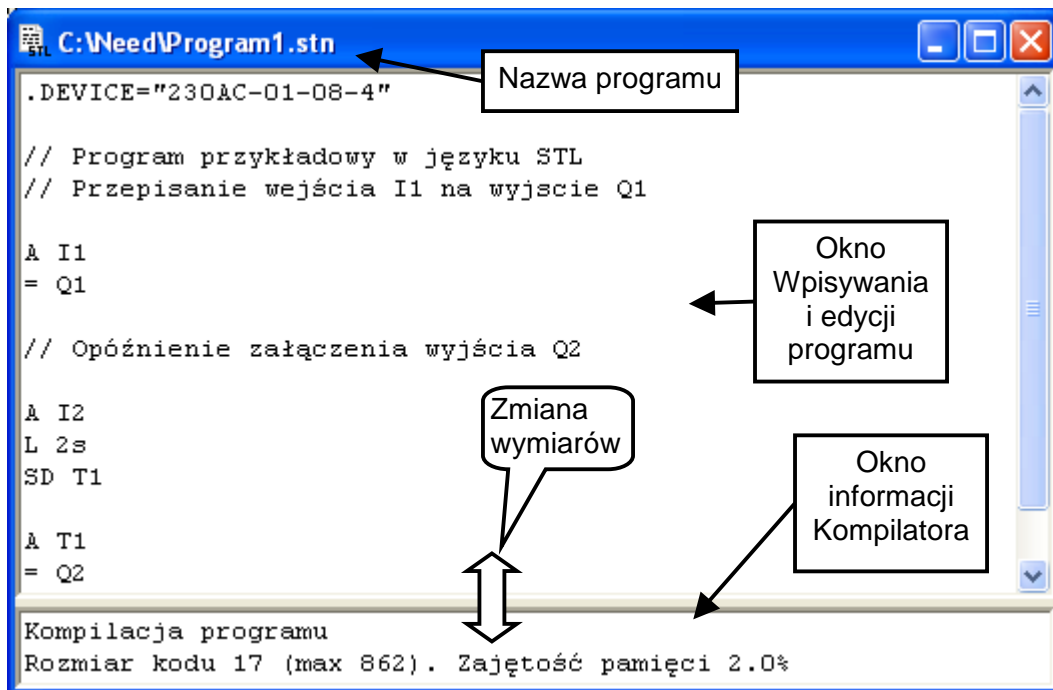
CTRL+NUM+ – jednoczesne naciśnięcie klawisza Ctrl i Klawisza + sekcji numerycznej klawiatury

CTRL+ALT+ "R" – jednoczesne naciśnięcie klawiszy Ctrl i Alt (lewy) oraz klawisza R

CTRL+SHIFT+ "Z" – jednoczesne naciśnięcie klawiszy Ctrl i SHIFT oraz klawisza Z

## 6.7. Edytor programu STL

Edycja programu w języku STL odbywa się w edytorze STL – okno poniżej.



Rys. 6.7.1. Edytor STL.

### 6.7.1. Edytor STL

Okno edytora otwierane jest w obszarze roboczym programu PC Need i podlega zachowaniom standardu Windows® – zmiana rozmiarów i położenia, zamykanie. Obsługa edytora jest podobna do obsługi prostego notatnika. Tekst wprowadzamy z klawiatury, stosując zasady składni podane w rozdziale 5.1. Programowanie w języku tekstowym STL.

Możliwe są operacje usuwania, przenoszenia, kopiowania zaznaczonego fragmentu bądź całego tekstu.

- **Zaznaczanie** – za pomocą myszki lub klawiatury (klawisze SHIFT+ strzałki) podświetli wybrany tekst.
- **Wycinanie** – zaznacz tekst, który ma być wycięty a następnie wykonaj polecenie *Wytnij*, czyli kombinację klawiszy Ctrl+X (jednoczesne naciśnięcie klawiszy Ctrl i X). Wycięty tekst będzie zachowany w schowku.
- **Usuwanie** – zaznacz tekst, który ma być usunięty a następnie wykonaj polecenie *Usuń* – klawisz Del.
- **Wklejanie** – ustaw kursor w miejscu, gdzie ma być początek wklejanego tekstu, wykonaj polecenie *Wklej* – kombinacja klawiszy Ctrl+V (jednoczesne naciśnięcie klawiszy Ctrl i V). Po wykonaniu operacji następuje wstawienie zawartości schowka systemowego.
- **Przenoszenie** – polega na wykonaniu operacji zaznaczenia (*Zaznaczanie*) wybranego tekstu, wycięcia go (*Wytnij* Ctrl+X) a następnie wklejenia go w żądanym miejscu (*Wklej* Ctrl+V).
- **Kopiowanie** – zaznacz tekst, który ma być skopiowany (patrz: *Zaznaczanie*) a następnie wykonaj polecenie *Kopiuj*, czyli kombinację klawiszy Ctrl+C (jednoczesne naciśnięcie klawiszy Ctrl i C).
- **Szukanie** – otwórz okno *Szukaj* (Ctrl+F) i wpisz poszukiwany tekst.

## Komentarze

Edytowany program w celu zwiększenia czytelności możemy opatrzyć komentarzami. Tekst objęty komentarzem nie jest analizowany przy tworzeniu kodu wykonawczego. Komentarz, który rozpoczyna się znakiem //lub; obowiązuje do końca linii. Taki komentarz możemy zacząć od początku linii lub po wpisaniu instrukcji.

Przykłady:

```
// To jest komentarz od początku linii.
; To też jest komentarz od początku linii.
A I1 ;To jest komentarz po instrukcji.
```

W celu pominięcia w trakcie kompilacji większej liczby linii możemy zastosować komentarz typu: /\* tekst \*/. W tym typie komentarza określamy obowiązkowo początek i koniec tekstu, który nie będzie należał do kodu programu.

Przykład:

```
/* A I1
A I2
= Q1
*/
A I3
=Q2
```

W powyższym przykładzie trzy pierwsze instrukcje będą pominięte przy kompilacji programu. Dopiero od instrukcji A I3 będzie tworzony kod wynikowy.

### 6.7.2. Kompilacja STL

Dolne okno początkowo jest puste, po wykonaniu komendy kompilacja (klawisz F7) pojawia się raport z kompilacji. Jeśli program jest poprawny, to otrzymamy komunikat jak na rys. 6.7.1. o zakończeniu kompilacji naszego programu oraz o rozmiarze kodu i procentowej zajętości pamięci w przekaźniku.

Jeśli program będzie zawierał błędy, to otrzymamy komunikat o typie błędu i jego położeniu [numer wiersza, numer kolumny] – rys. 6.7.2. wraz z komentarzem.

```
C:\NeedVProgram1.stn
// Program przykładowy w języku STL
// Przepisanie wejścia I1 na wyjście Q1

A I1
= Q1

// Opóźnienie załączenia wyjścia Q2

A I2
L 2s
SD T1

T1
= Q2

Kompilacja programu
Błąd 4 [15, 1]: "T1" Niewłaściwa instrukcja na początku obwodu
Błąd 4 [16, 1]: "=" Niewłaściwa instrukcja na początku obwodu
Liczba błędów: 2
```

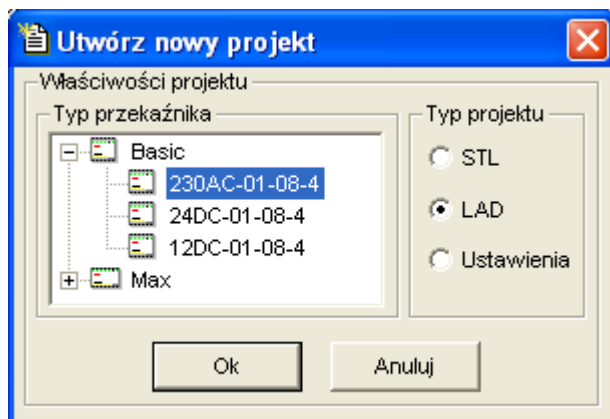
Rys. 6.7.2. Błąd kompilacji.



## 6.8. Edytor programu LAD

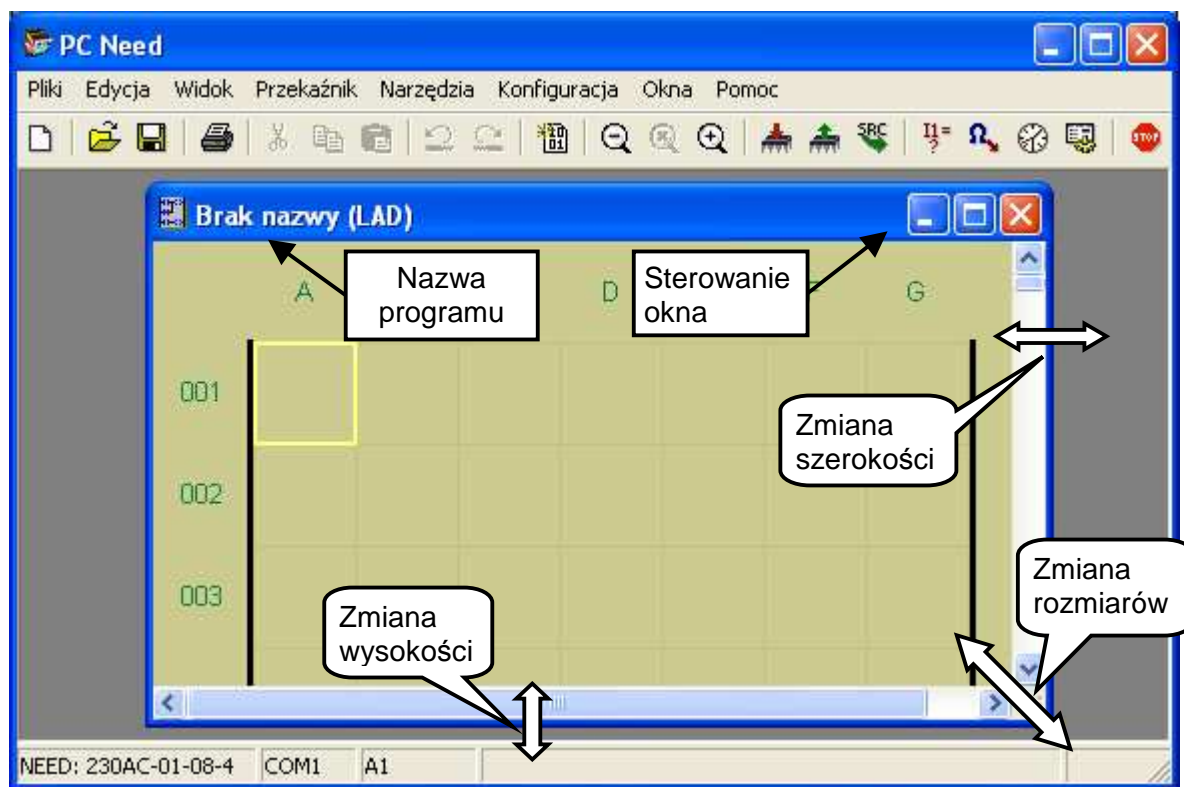
### 6.8.1. Nowy program

Po uruchomieniu programu PC Need, aby utworzyć program w języku LAD, wybieramy Menu **Pliki > Nowy > LAD**



Rys. 6.8.1.1. Nowy projekt LAD.

Wybór zatwierdzamy klawiszem *Enter* lub lewym przyciskiem myszy. W obszarze roboczym utworzy się okno **edytora LAD**.

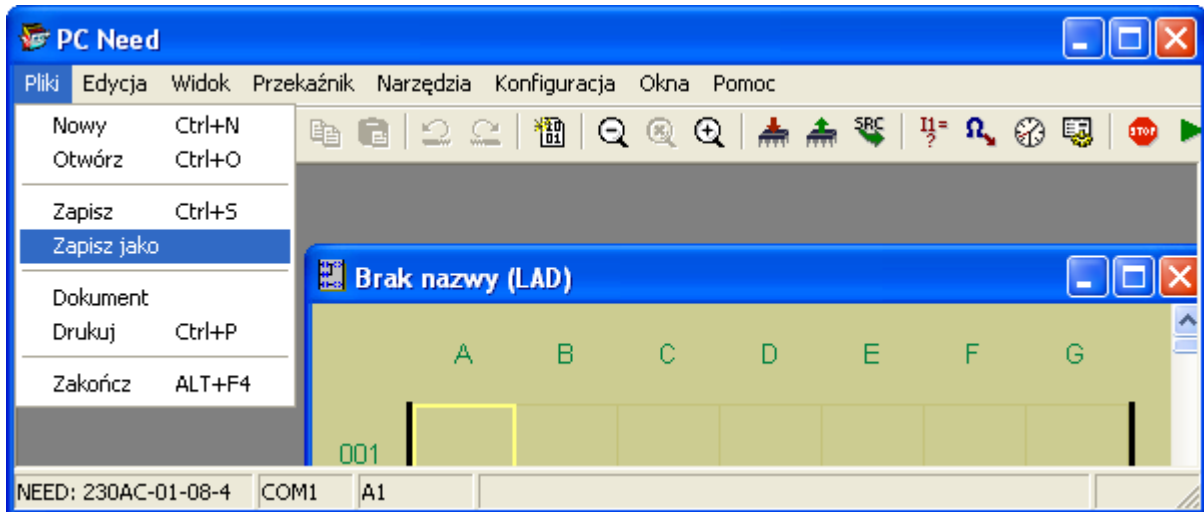


Rys 6.8.1.2. Edytor Lad – nowy program.

Za pomocą myszki można dostosować rozmiar okna do własnych potrzeb i preferencji. Korzystając ze standardowych przycisków sterujących oknami można okno edytora LAD zmaksymalizować, zminimalizować, lub zamknąć (x).

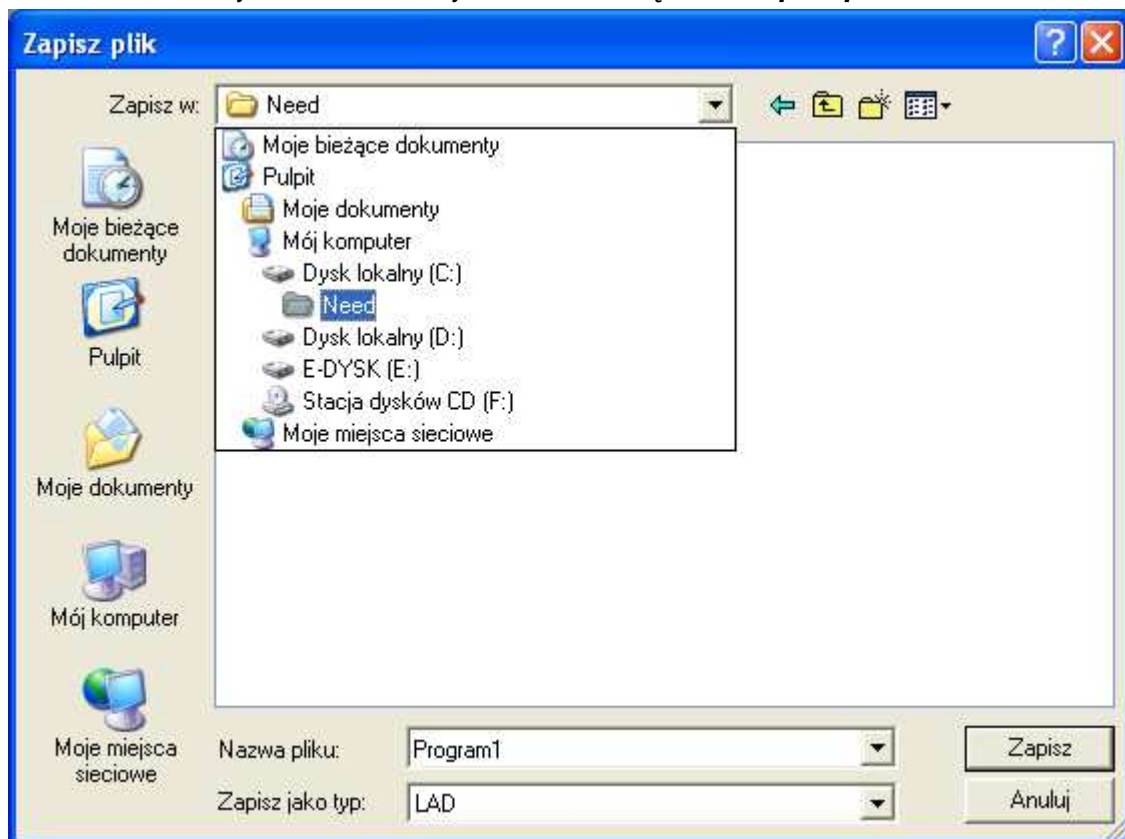
## 6.8.2. Zapisanie programu

Ponieważ nowo otwarty program nie ma nazwy (*Brak nazwy (LAD)*) należy go zapisać nadając mu odpowiednią nazwę. W tym celu w Menu Pliki wybieramy opcję *Zapisz jako*:



Rys. 6.8.2.1. Okno Zapisz jako.

Po zatwierdzeniu lewym klawiszem myszki otwiera się okno **Zapisz plik**:



Rys. 6.8.2.2. Okno Zapisz plik LAD.

Wybieramy położenie pliku (ścieżka dostępu)– w tym wypadku:

Zapisz w: *Need*;


Nazwa pliku: wpisujemy np. *Program1*;

Zapisz jako typ: *LAD* (*domyślne rozszerzenie nazwy pliku – .ldn*)

i zatwierdzamy operację klawiszem *Zapisz*

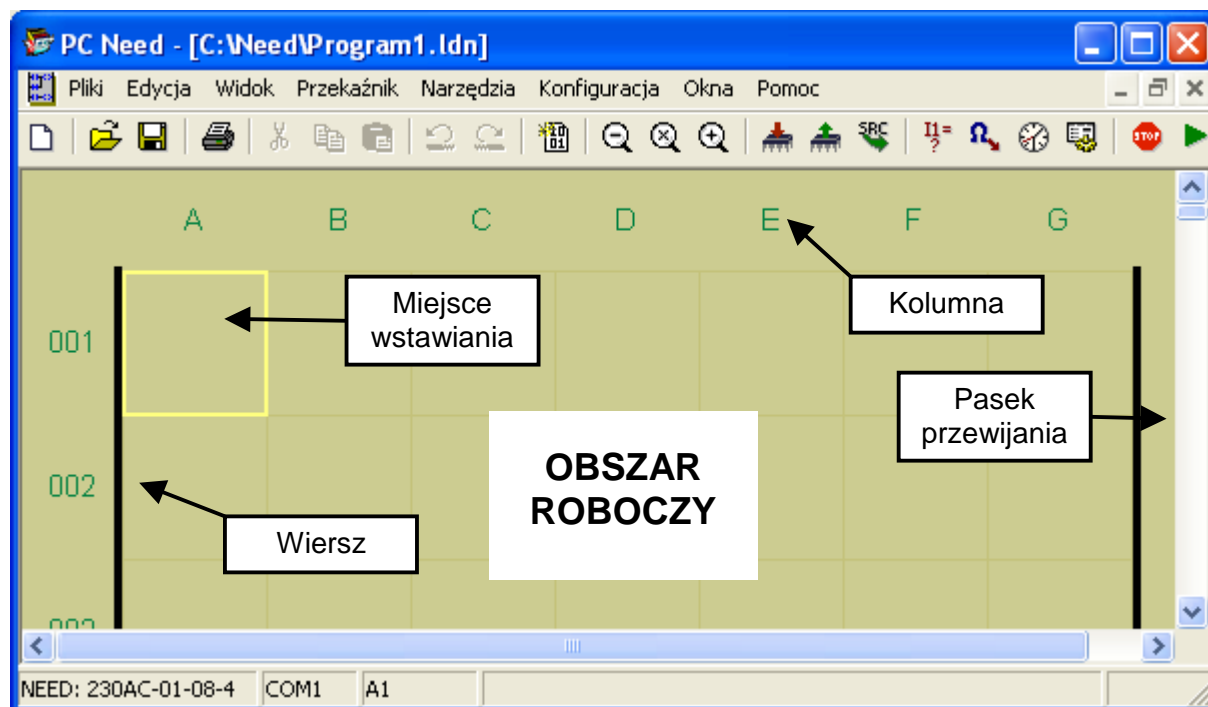
### 6.8.3. Otwarcie istniejącego programu

Jeśli chcemy otworzyć już istniejący dokument, to przy pomocy Menu **Pliki** i wybraniu **Otwórz** – otwiera się standardowe okno menadżera plików jak powyżej, w którym wybieramy plik z rozszerzeniem „\*.ldn”.

Podobny efekt uzyskamy wykorzystując przycisk  z paska narzędzi lub skrót klawiaturowy **Ctrl+O**.

### 6.8.4. Wprowadzanie i edycja schematu połączeń

W Menu **Pliki** tworzymy nowy program LAD (patrz p. 6.8.1.) i nadajemy mu nazwę np. *Program1.ldn* (patrz p. 6.8.2). Dostosowujemy rozmiar okna i otrzymujemy okno edytora LAD:



Rys. 6.8.4.1. Okno edytora LAD.

Obszar roboczy, to siatka oparta na kwadratach, których położenie określa litera kolumny **A, C, E...** oraz numer wiersz **001..150**.

Kolumny **A, C, E...** są przeznaczone do wstawiania elementów wejściowych programu (Wejścia fizyczne, stan wyjść, *Znaczników, Timerów, Liczników, Zegarów, Komparatorów*) lub połączeń.

Kolumny **B, D, F...** są przeznaczone do wstawiania połączeń pomiędzy elementami. Ostatnia kolumna, to miejsce wstawiania elementów wyjściowych (fizyczne wyjścia, *Znaczniki Timery, Liczniki*).

#### Rysowanie schematu połączeń

Wewnątrz okna edytora LAD za pomocą myszki przemieszczamy się po kwadratowych polach siatki, aktualnie wybrane pole jest zaznaczone szarą obwódką. Dodatkowo na pasku stanu, w lewym dolnym rogu, podawane są współrzędne pola (wiersz, kolumna).

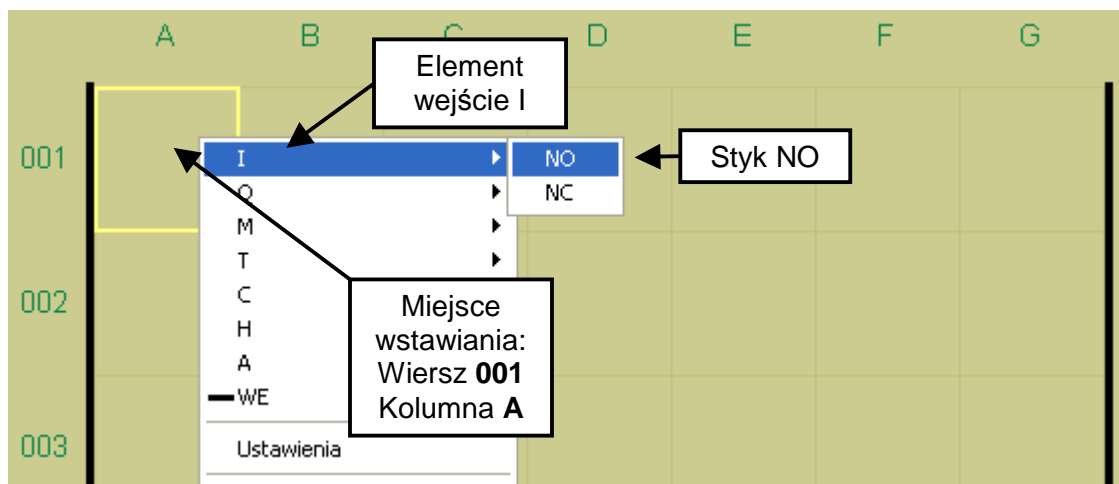
Pole wybieramy ustawiając nad nim kursor i klikając lewym klawiszem myszki.

Po wybraniu pola (zaznaczeniu), za pomocą prawego klawisza myszki, wybieramy z rozwijanego menu element lub połączenie (w zależności od bieżącej kolumny).

Stosowane symbole są zgodne z opisem języka LAD (rozdział 5.2.).

**Wstawianie elementu wejściowego:**

Klikając prawym klawiszem w kolumnie A, C lub E (poniżej jest to pole 001 A) wywołujemy rozwijane menu jak na rys. 6.8.4.2.

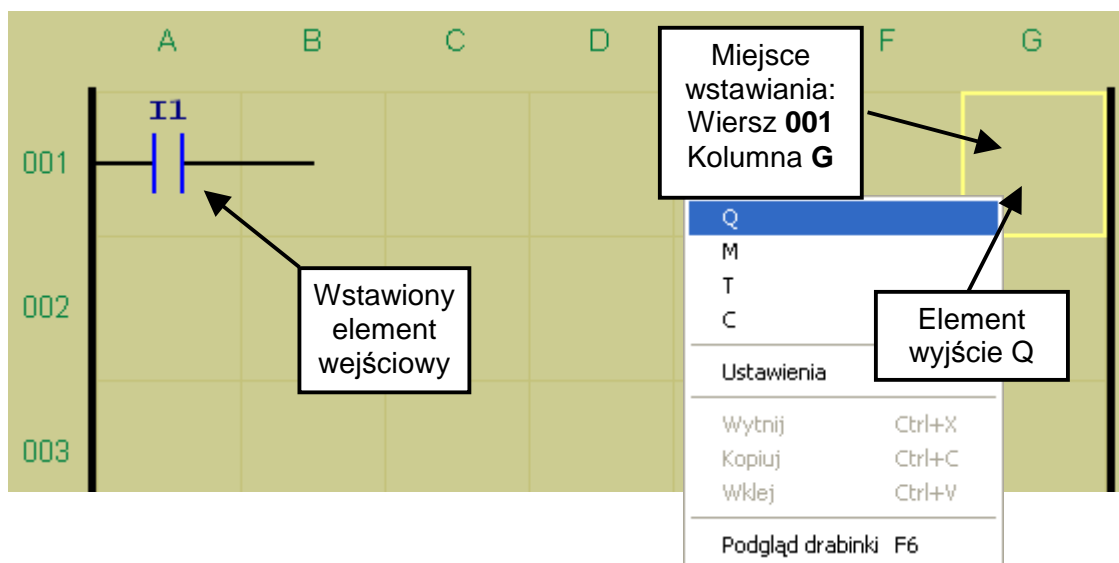


Rys. 6.8.4.2. Wstawianie elementu wejściowego.

Zatwierdzamy wybór lewym przyciskiem myszki lub klawiszem Enter.

**Wstawianie elementu wyjściowego**

Klikając prawym klawiszem w kolumnie G (na rys. poniżej jest to pole 001 G) wywołujemy rozwijane menu jak na rys 6.8.4.3.

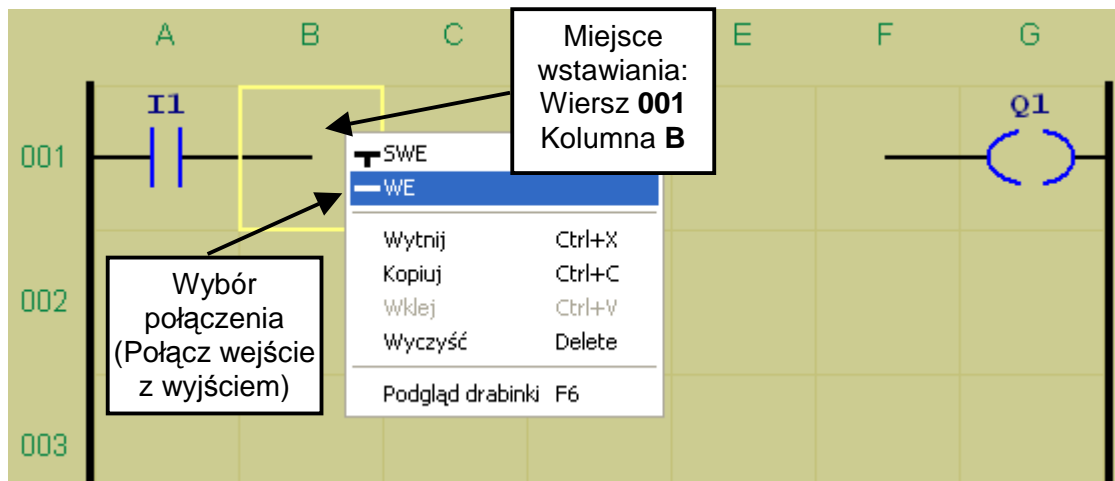


Rys. 6.8.4.3. Wstawianie elementu wyjściowego.

Zatwierdzamy wybór lewym przyciskiem myszki lub klawiszem klawiszem *Enter*.

**Usuwanie elementu**

Możliwe jest także usunięcie elementu – po wybraniu (zaznaczeniu) pola w którym znajduje się element, z rozwijanego menu (prawym klawiszem myszy) wybieramy (podświetlamy) **Wyczyść** – po zatwierdzeniu lewym klawiszem element zostanie usunięty. Alternatywnie działanie uzyskamy używając klawisza Delete z klawiatury.

**Wstawianie połączenia:**

Rys. 6.8.4.4. Wstawianie połączenia.

Po wybraniu pola połączeń i naciśnięciu prawego klawisza myszy wyświetlane są, w rozwijanym menu, aktualnie możliwe do wykorzystania połączenia. Oprócz graficznego symbolu podany jest skrótowy opis kierunków będący kombinacją liter S, W, N, E.

**S** – South (dół)

**W** – West (lewo)

**N** – North (górze)

**E** – East (prawo)

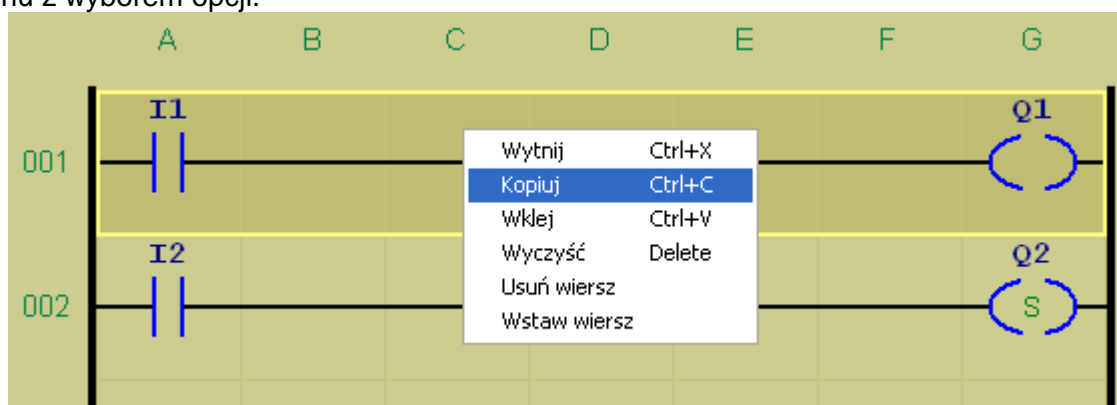
Możliwe jest także usunięcie połączenia – po wybraniu (zaznaczeniu) pola, w którym istnieje połączenie, z rozwijanego menu, (prawy klawisz myszy) wybieramy (podświetlamy)

**Wyczyść** – po zatwierdzeniu lewym klawiszem, połączenie zostanie usunięte.

**Edycja obszaru**

Program PC Need umożliwia edycję projektu LAD poprzez operacje usuwania, przenoszenia, kopiowania zaznaczonego obszaru. Obszar do kopiowania jest prostokątem opartym na kwadratach siatki połączeń.

Zaznaczenie wykonujemy lewym klawiszem myszki, prawym klawiszem otwieramy rozwijane menu z wyborem opcji.

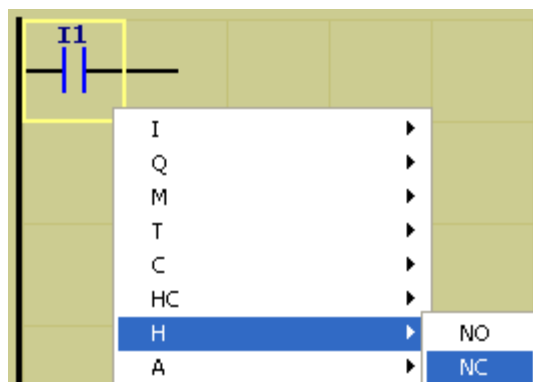


Rys. 6.8.4.5. Zaznaczenie wiersza i wybór opcji.

- **Wytnij** - Zaznaczony (ciemniejsze tło) wiersz lub obszar możemy przemieścić w inne miejsce korzystając z opcji *Wytnij*, a następnie wskazując początkowy lewy kwadrat siatki miejsca przeznaczenia i operację *Wklej*.
- **Kopiuj** - Zaznaczony obszar możemy skopiować w inne miejsce korzystając z opcji *Kopiuj*, a następnie wskazując początkowy lewy kwadrat siatki miejsca przeznaczenia i operację *Wklej*.

- **Wyczyść** -możemy wyczyścić zaznaczony obszar – otrzymamy czystą siatkę
- **Usuń wiersz** – zaznaczony wiersz zostanie usunięty
- **Wstaw wiersz** – w zaznaczone miejsce wstawiony zostanie czysty wiersz

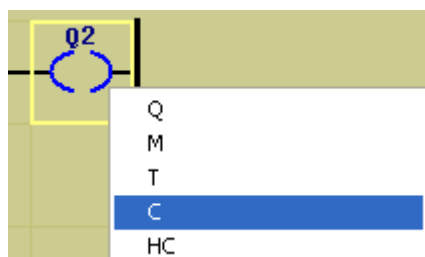
### Zmiana rodzaju wejścia.



Klikamy lewym klawiszem na polu elementu, który chcemy zmienić (I1).  
Prawym klawiszem rozwijamy menu, w którym wybieramy nowy rodzaj elementu (H) i typ styku (NO lub NC).  
Zatwierdzamy wybór lewym klawiszem myszy.

Rys. 6.8.4.6. Zmiana rodzaju wejścia.

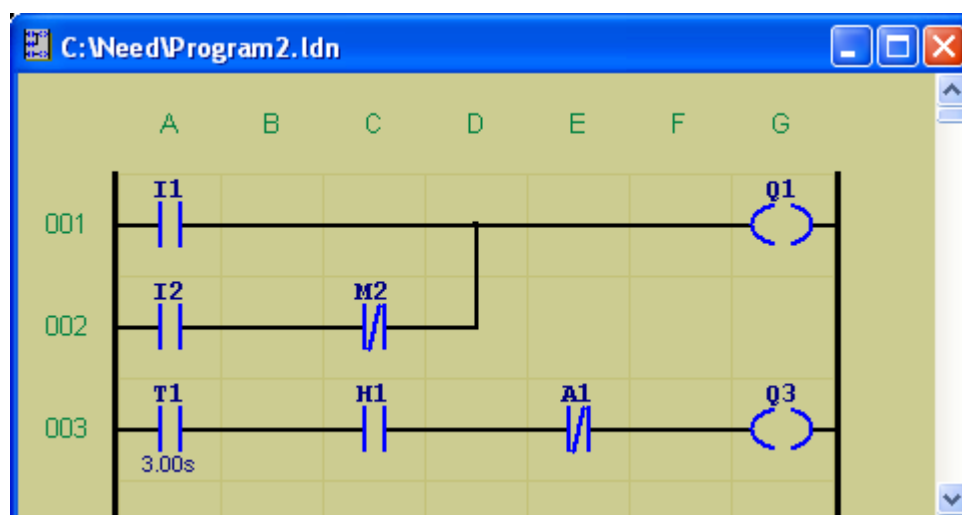
### Zmiana rodzaju wyjścia.



Klikamy lewym klawiszem na polu elementu, który chcemy zmienić.  
Prawym klawiszem rozwijamy menu, w którym wybieramy nowy rodzaj (M).  
Zatwierdzamy wybór lewym klawiszem myszy.

Rys. 6.8.4.7. Zmiana rodzaju wyjścia.

Postępując z powyższymi zasadami możemy utworzyć program jak na rys. 6.8.4.13. (poniżej)



Rys. 6.8.4.8. Program w LAD.

Z plikiem *Program2.lad* związane jest okno Ustawienia (*Program2.ladn*), które jest niezbędne do konfiguracji takich elementów jak **Timer**, **Zegar**, **Licznik**, **Komparator**, **Remanencja** oraz **Opóźnienia wejść**.

Wprowadzone w Ustawieniach wartości *Timerów* i *Liczników* są widoczne na schemacie LAD.

### 6.8.5. Edycja elementu

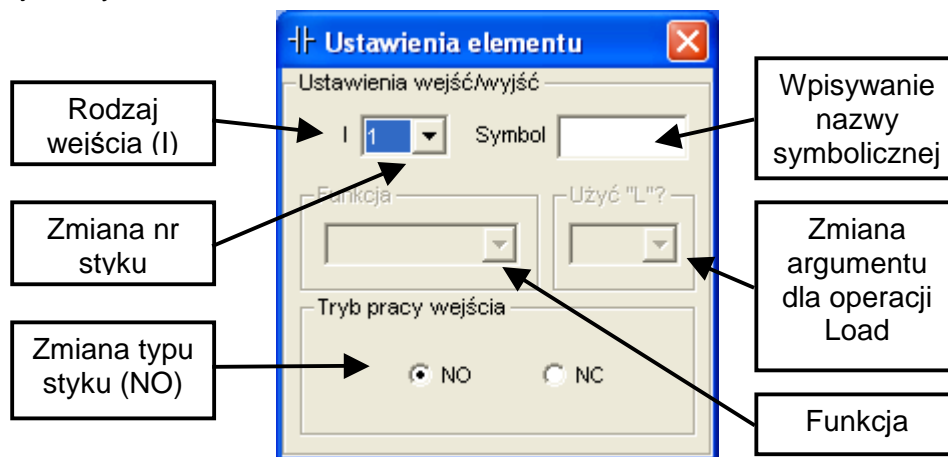
Każdy element umieszczony na schemacie (siatce połączeń) możemy poddać edycji: zmienić parametry, rodzaj i numer wejścia, wyjścia, typ styku.

#### Ustawienia elementu

Po dwukrotnym kliknięciu (lewym przyciskiem) na element umieszczony na schemacie wywołujemy okno *Ustawienia elementu*.

W zależności od rodzaju elementu dostępne są odpowiednie pola do zmiany.

Element wejściowy:



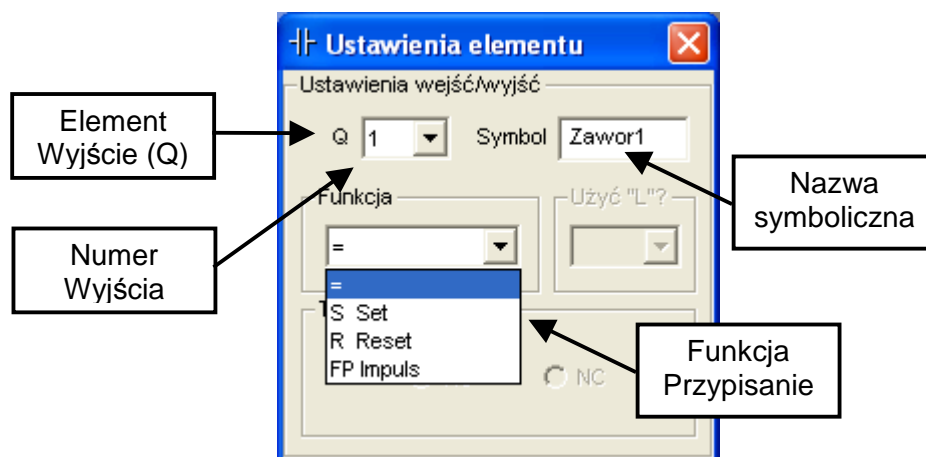
Rys. 6.8.5.1. Okno Ustawienia elementu.

Na rys. 6.8.5.1. przedstawiono *Ustawienia elementu* dla wejścia I (fizyczne wejście), podobnie wyglądają okna dla innych rodzajów wejść, A, H, Q, M, T, C. W zależności od rodzaju wejścia dostępny jest wybór numeru zależny od zasobów przekaźnika np. dla I jest to 1..8(16), dla M 1..16. Każde wejście ma wybierany typ styku NO lub NC.

Dla wejść, okno elementu ma nieaktywne pole *Funkcja*, które jest używane tylko dla elementów wyjściowych.

#### Ustawienia elementu dla wyjścia Q i Znacznika M.

Możemy wybrać numer wyjścia (1..4(8) lub, jeśli wybrano M to 1..16) oraz w zależności od rodzaju wyjścia dostępną funkcję. Dla wyjścia Q i *Znacznika M* będą to operandy: =, S, R, FP.



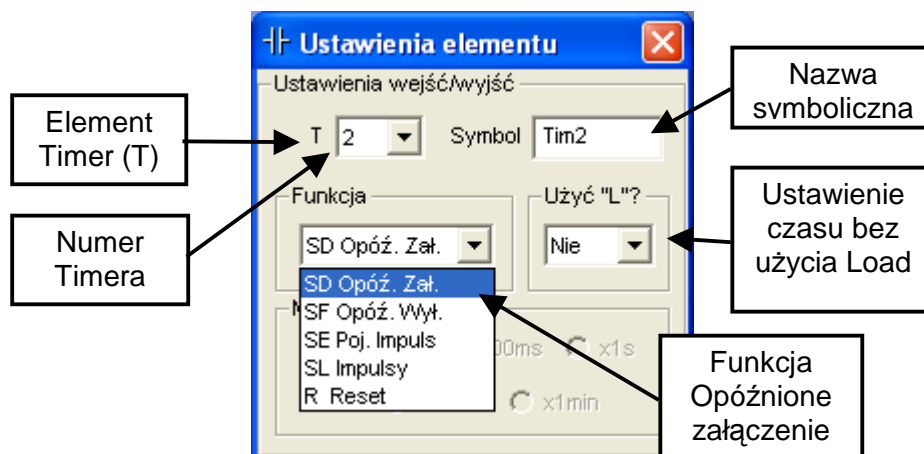
Rys. 6.8.5.2. Okno Ustawienia elementu Wyjście.

Dla wyjść okno *Ustawienia elementu* ma nieaktywne pole wyboru styku, natomiast aktywne jest pole *Funkcja*.



**Ustawienia elementu dla wyjścia Timer.**

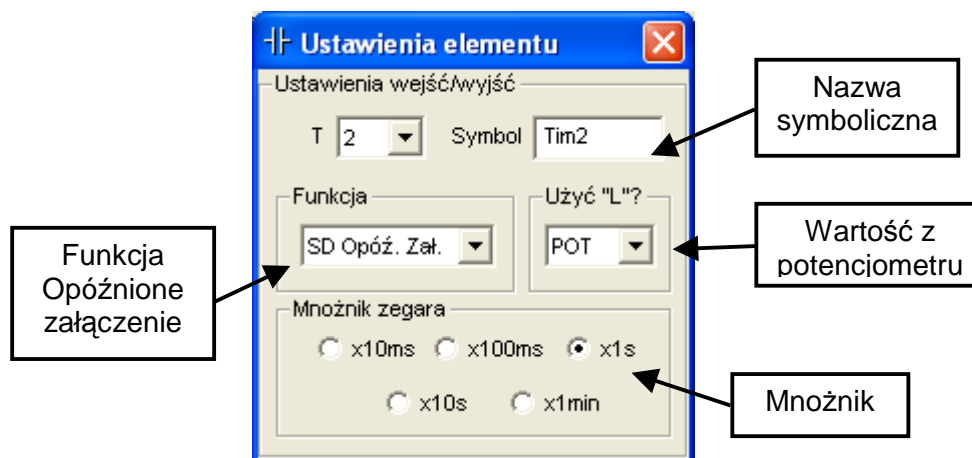
Dla wyjścia typu *Timer* (T) możemy wybrać parametry: numer 1..n oraz operandy: SD, SF, SE, SL, R.



Rys. 6.8.5.3. Okno Ustawienia elementu Timer.

Po zaznaczeniu opcji *Użyć "L" – Pot*, *Timer* jako wartość do zliczania, bierze mnożnik zegara i wartość ustawioną za pomocą Potencjometru. W przykładzie poniżej dla mnożnika x1s wartość do zliczenia może być ustawiona w zakresie 1s .. 255s ((1-255) x 1s).

Po zaznaczeniu opcji *Użyć "L" – AIn*, *Timer* jako wartość do zliczania, bierze mnożnik zegara i wartość odczytaną z wejścia analogowego In. Więcej na temat zastosowania użycia wartości analogowych do odmierzania czasów dla Timerów można znaleźć w rozdziale 5.1.2.21. Instrukcja ładowania (LOAD).



Rys. 6.8.5.4. Okno Ustawienia elementu Timer POT.



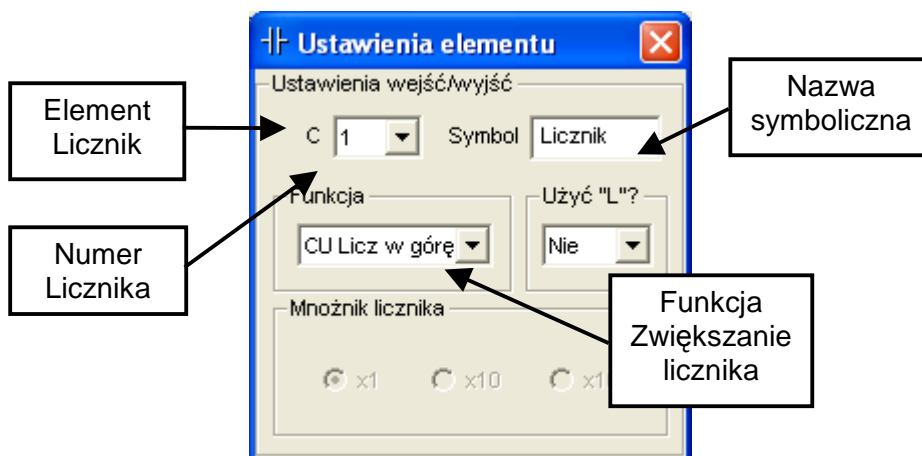
**Uwaga:** Zmiana wartości czasu odbywa się w oknie – *Ustawienia*.

AIn – to np. AI7, AI8 dla NEED..-x1-08-... i AI14, AI15, AI16 dla NEED..-x1-16-...



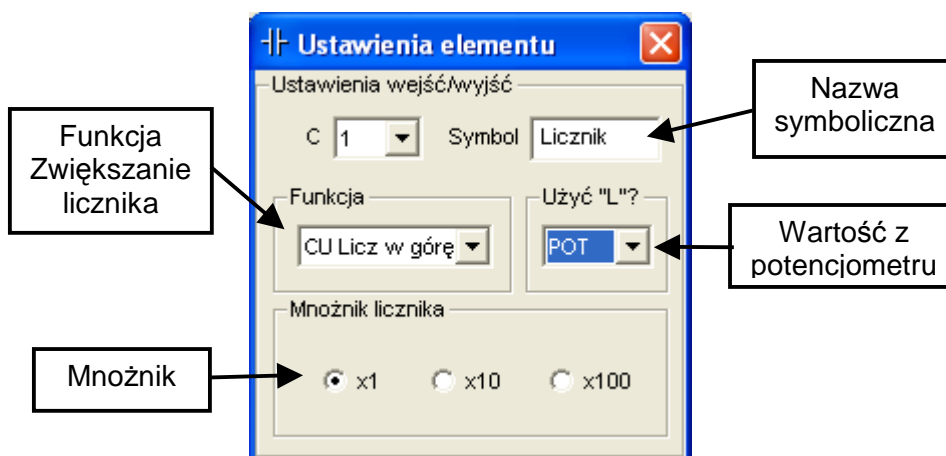
### Ustawienia elementu dla wyjścia Licznik

Dla wyjścia typu *Licznik* (C, HC) możemy wybrać parametry: numer 1..8 oraz operandy: CU, CD, R.



Rys. 6.8.5.5. Okno Ustawienia elementu Licznik.

Po zaznaczeniu opcji *Użyć "L" – Pot*, *Licznik* jako wartość do zliczania, bierze mnożnik licznika i wartość ustawioną za pomocą Potencjometru. W przykładzie poniżej dla mnożnika x1 wartość do zliczenia może być ustawiona w zakresie 1..255 ((1-255) x 1).



Rys. 6.8.5.6. Okno Ustawienia elementu Licznik POT.

Po zaznaczeniu opcji *Użyć "L" – Ain*, *Licznik* jako wartość do zliczania, bierze mnożnik zegara i wartość odczytaną z wejścia analogowego In . Wartości czytane z wejść analogowych przyjmowane są podobnie jak odczyt Potencjometru w zakresie 1..255. Ain – to odpowiednio AI7, AI8 dla NEED..DC-x1-08-.. i AI14, AI15, AI16 dla NEED..DC-x1-16-...



**Uwaga:** Zmiana wartości do zliczania odbywa się w oknie – *Ustawienia*.

## 6.9. Ustawienia

### 6.9.1. Rodzaje ustawień

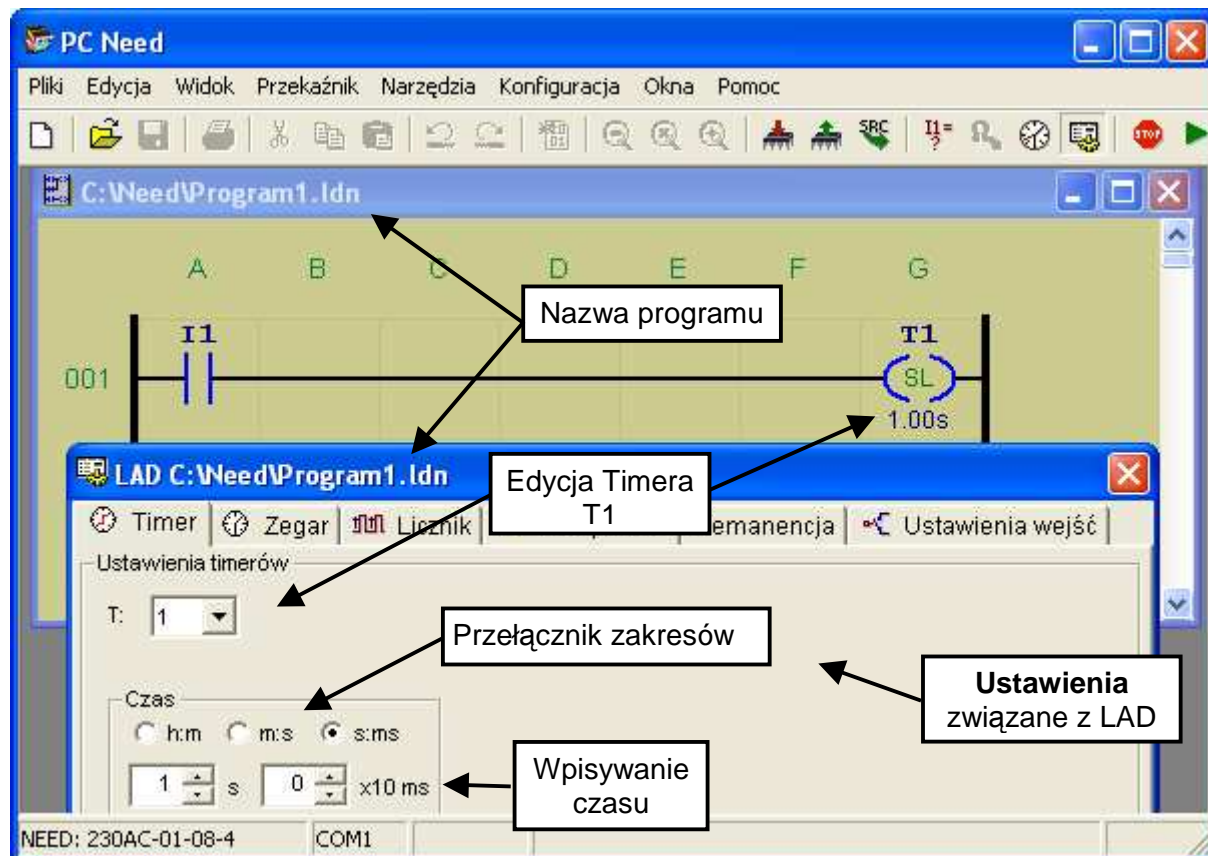
Wyróżniamy dwa typy ustawień:

1. Okno Ustawienia związane z programem LAD.
2. Plik Ustawienia niezależny.

W pierwszym przypadku dla aktywnego okna programu LAD tworzone są ustawienia przechowujące dane dla tego programu.

W drugim przypadku możemy utworzyć samodzielny plik (lub pliki) o różnych nazwach, które można niezależnie wgrać do przekaźnika, zastępując dotychczasowe ustawienia.

Ma to sens np., jeśli chcemy nie zmieniając programu zmienić czasy *Timerów*. Bez szukania w programie, tylko edytując plik ustawienia, możemy wprowadzić nowe wartości.



Rys. 6.9.1.1. Ustawienia związane z programem LAD.

Jeśli plik Ustawienia, związany z programem LAD, nie zostanie załadowany, to program w przekaźniku będzie wykonywany z ustawieniami, jakie ostatnio były w przekaźniku.

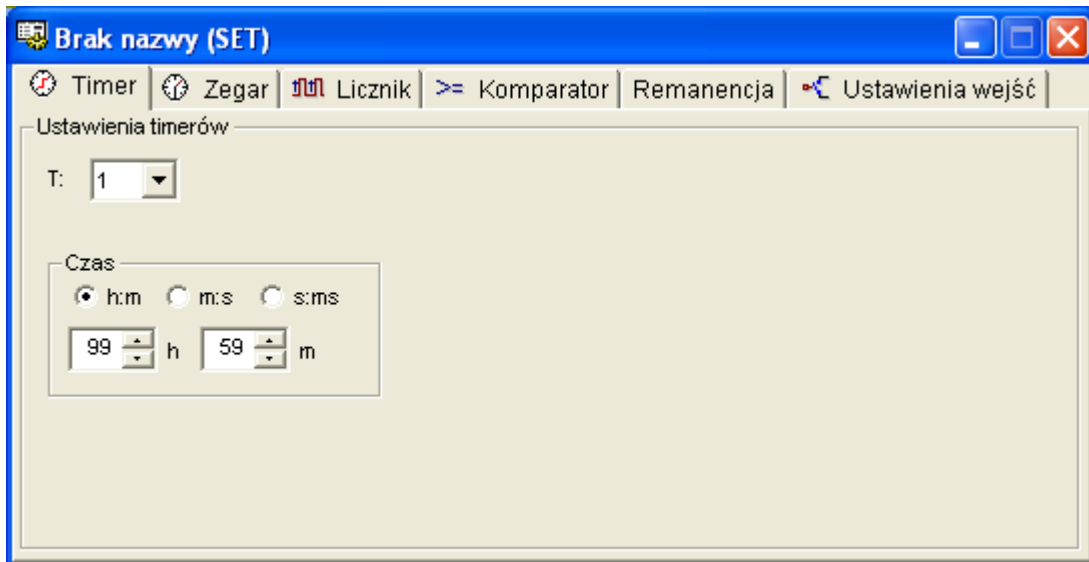
Po resecie będą to wartości maksymalne *Timerów* i *Liczników*, wyzerowane *Zegary*, ustawiony brak remanencji i załączone opóźnienia wejść.

Aby przejść do Ustawień w przypadku edycji programu LAD należy kliknąć ikonkę na pasku narzędzia lub alternatywnie wybrać w Menu: **Przekaźnik > Ustawienia** (ewentualnie klawisz skrót F10). Na rys. 6.9.1.2. przedstawiono efekt wykonania tego polecenia i edycji *Timera 1* (tryb SE, czas 1s).

Aby utworzyć nowy plik SET należy w Menu **Plik** wybrać **Nowy** a następnie w oknie *Utwórz nowy projekt* zaznaczyć **Ustawienia** oraz wybrać wersję przekaźnika.

Ustawienia związane z programem LAD są zapisywane automatycznie przy zapisie tego programu, jeżeli opcja ta jest aktywna w konfiguracji projektu LAD.

Ustawienia utworzone samodzielnie należy zapisać podobnie, jak program LAD czy STL nadając im nazwę.



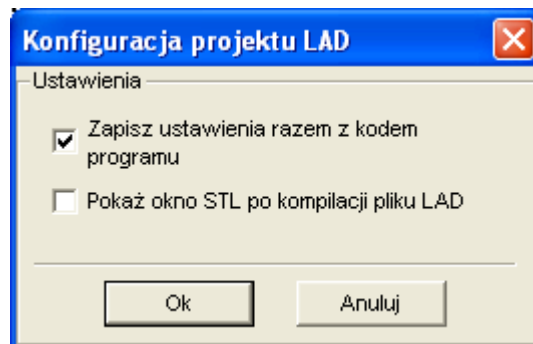
Rys. 6.9.1.2. Nowo utworzony plik ustawień (SET).

Domyślne rozszerzenie zapisywanego na dysku pliku Ustawienia to „\*.set”.  
Poniżej przedstawiono zasadnicze różnice dla ustawień związanych z programem LAD oraz dla niezależnego pliku *Ustawienia1.set*.

**i Uwaga:** Plik Ustawienia jest ładowany niezależnie od programu. Domyślnie w programie NEED PC, dla edytora LAD, ładowany jest program a następnie ustawienia z nim związane.

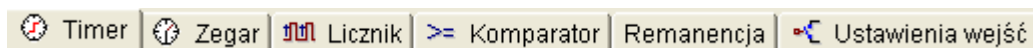
Automatyczne ładowanie ustawień dla projektu LAD można wyłączyć korzystając z opcji Menu:



**Konfiguracja > Projekt LAD.**



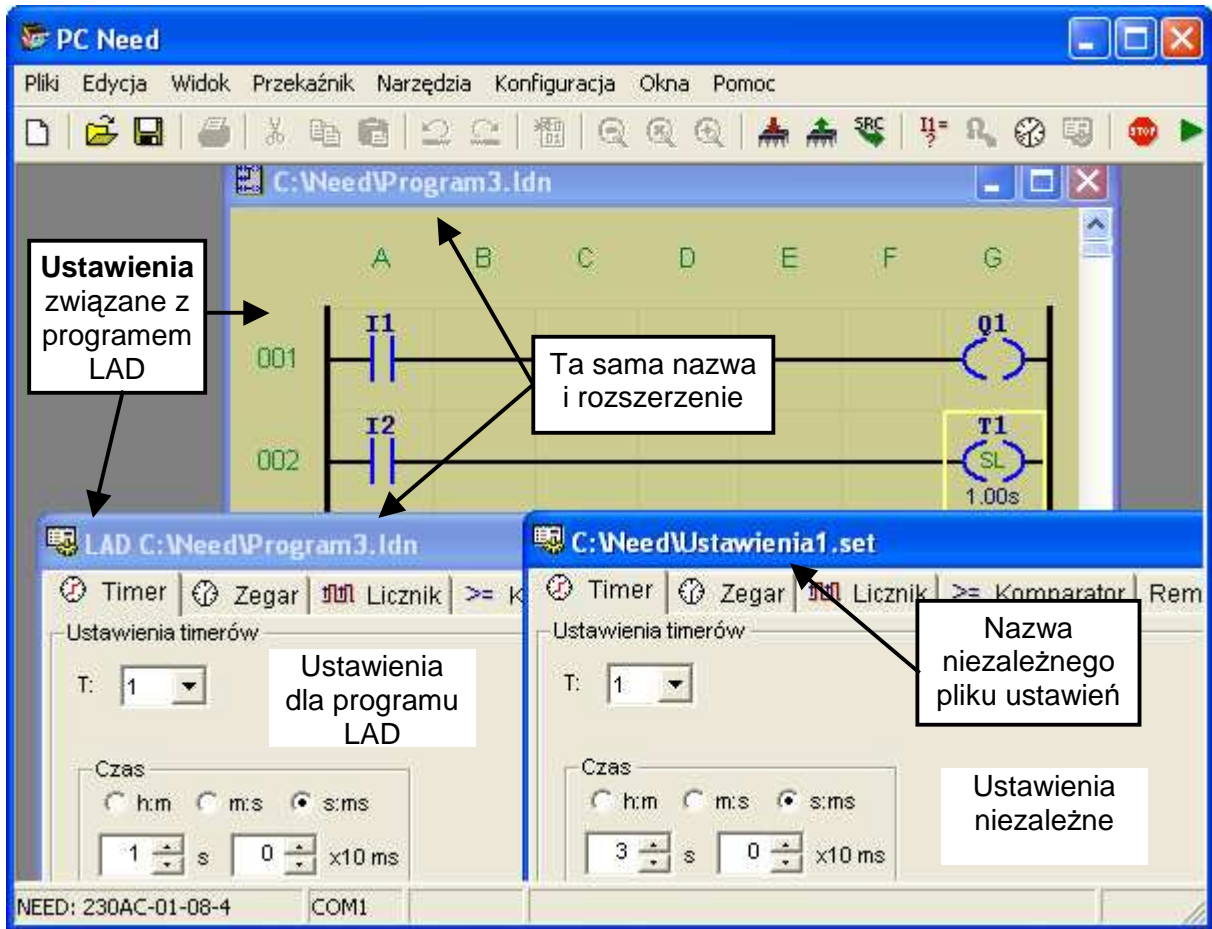
Rys. 6.9.1.3. Konfiguracja projektu LAD.

Posługując się zakładkami wybieramy typ zmiennych do ustawienia.



Okno *Ustawienia* skojarzone z programem LAD ma tylko możliwość zamknięcia  natomiast plik SET ma jeszcze ikonizowanie .

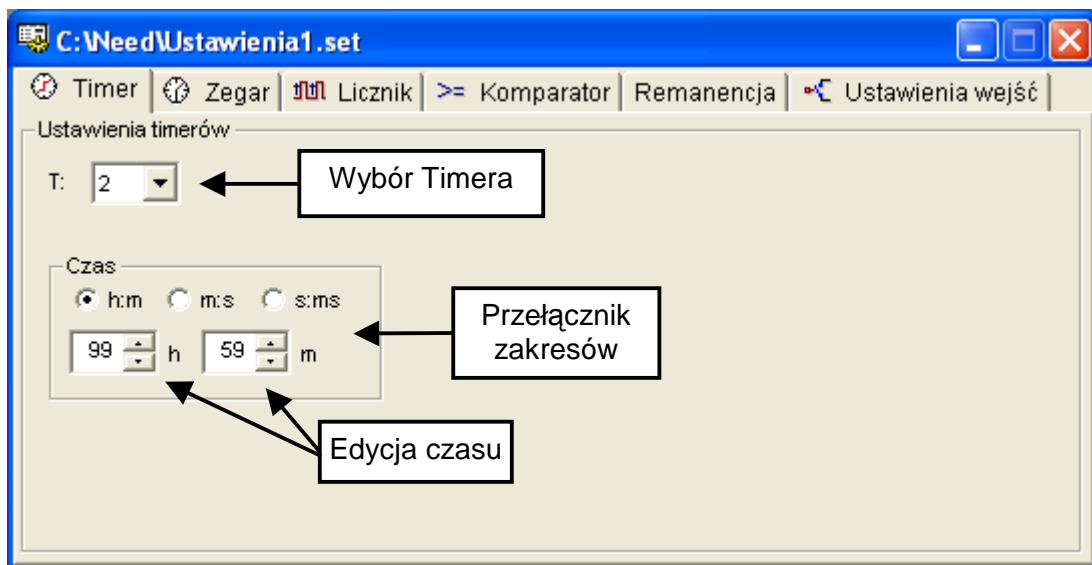
Poza nazwą i skojarzeniem edycja ustawień jest identyczna, polega na wypełnianiu opisanych pól lub wyborze wartości z rozwijanych menu.



Rys. 6.9.1.4. Różnice w ustawieniach.

### 6.9.2. Ustawienia Timerów

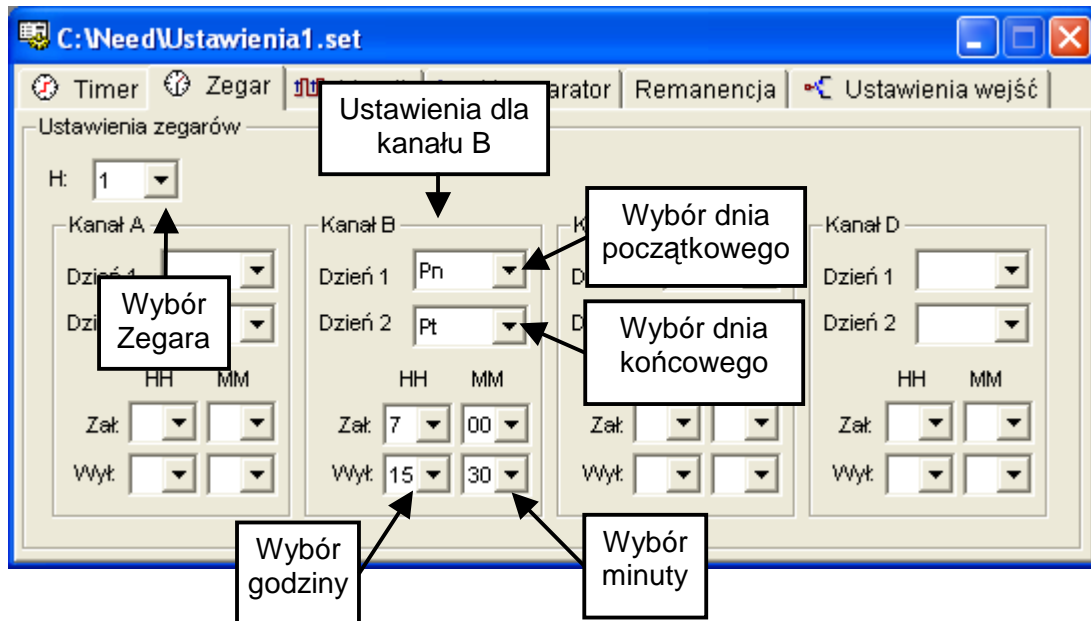
Wybieramy numer *Timera* a następnie przyporządkowujemy mu zakres (godz:minuty, minuty:sekundy, sekundy:milisekundy x 10). Potem wpisujemy zadaną wartość czasu w pola do edycji.



Rys. 6.9.2.1. Ustawienia Timerów.

### 6.9.3. Ustawienia Zegarów

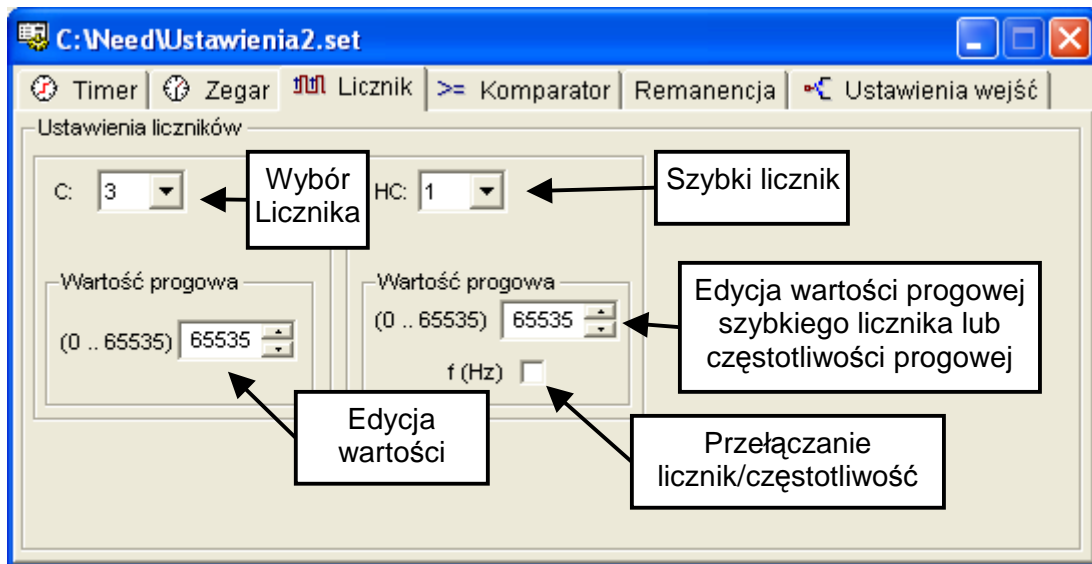
Wybieramy numer zegara (H1..H4) a następnie edytujemy wybrane kanały A..D ustawiając dni tygodnia, godziny i minuty.



Rys. 6.9.3.1. Ustawienia Zegarów.

### 6.9.4. Ustawienia Liczników

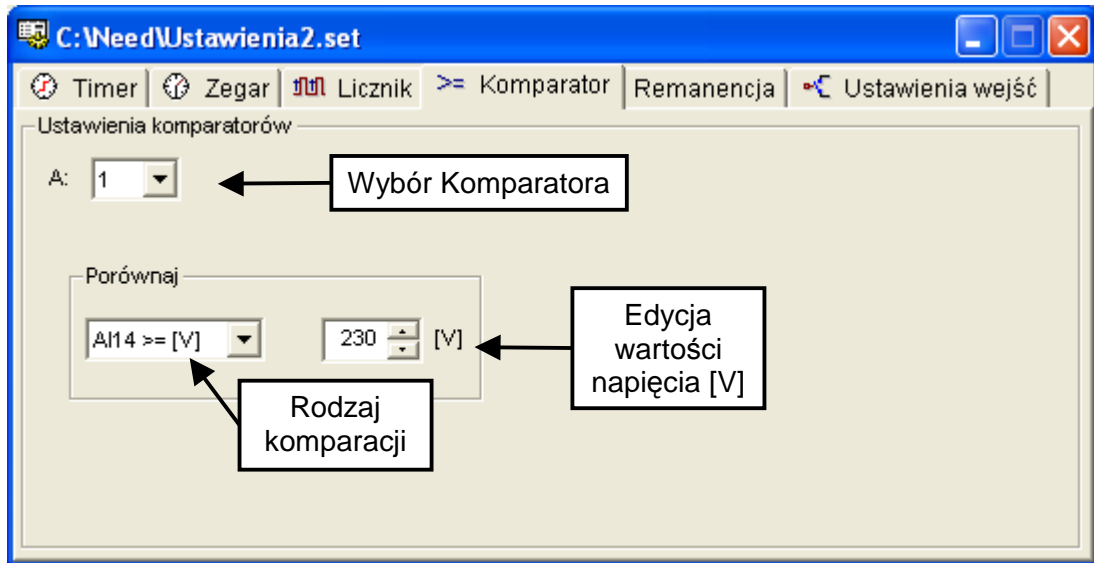
Wybieramy numer *Licznika*, a następnie wpisujemy wartość do zliczenia.



Rys. 6.9.4.1. Ustawienia Liczników.

### 6.9.5. Ustawienia Komparatorów

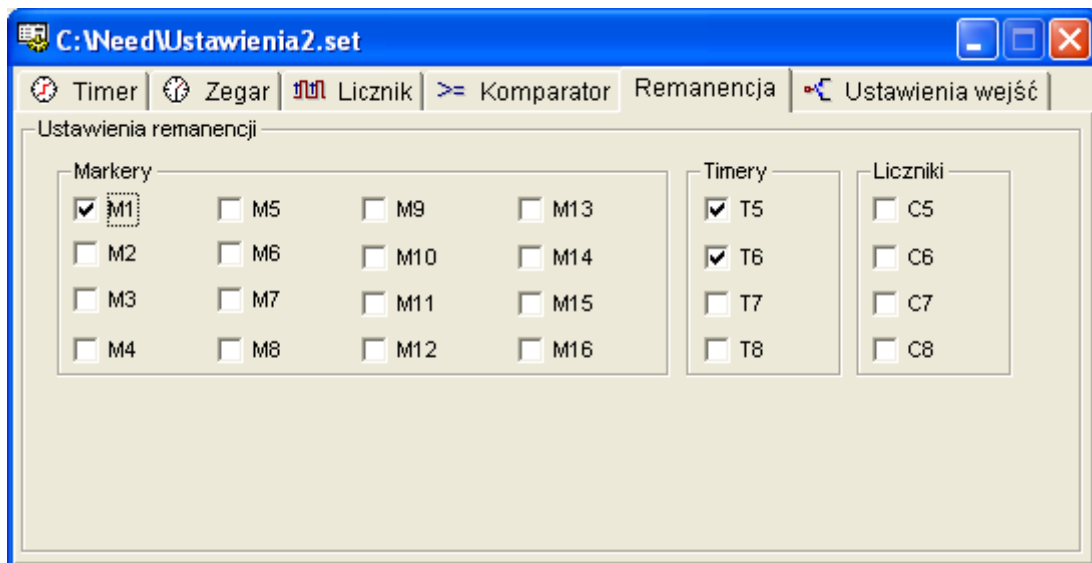
Wybieramy numer *Komparatora* i przypisujemy mu rodzaj porównania, a dla porównań z wartością stałą wpisujemy wartość napięcia w woltach.



Rys. 6.9.5.1. Ustawienia Komparatorów.

### 6.9.6. Remanencja

Wyboru zmiennej, która ma być remanentna dokonuje się poprzez zaznaczenie pola przy danej zmiennej (rys. 6.9.6.1 wybrano jako remanentne M1, T5 i T6).



Rys. 6.9.6.1. Ustawienia Remanencji

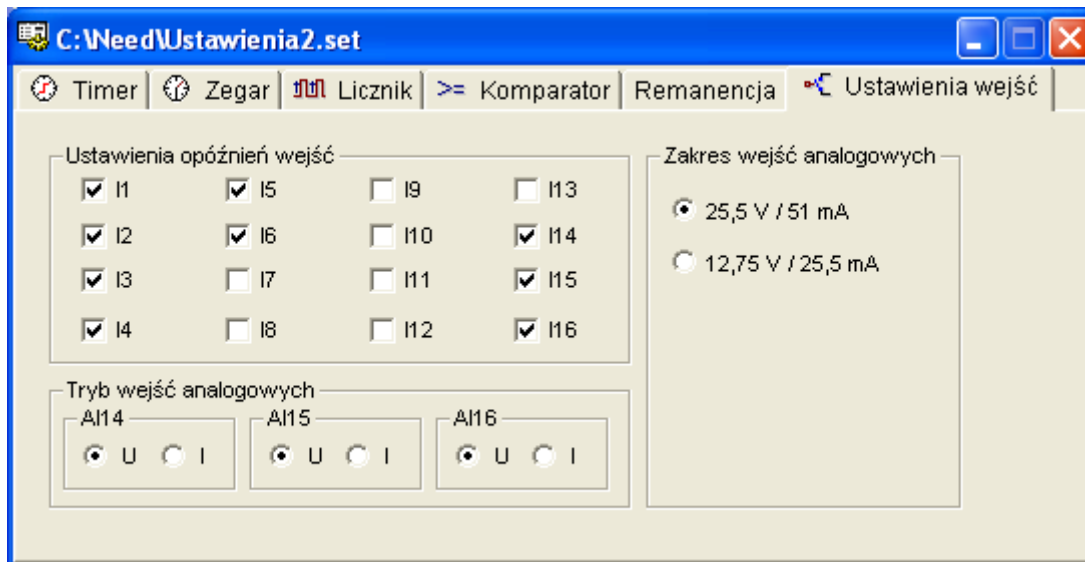


### 6.9.7. Ustawienia wejść

Jeśli wejście ma być skanowane bez opóźnienia, to należy odznaczyć jego pole wyboru. (na rys. 6.9.7.1. wejścia od I7 do I13 nie będą opóźniane).

Domyślnie wejścia mają zaznaczone pola wyboru – jest ustawione opóźnienie wejść.


Dla wersji NEED ..DC-x1-16-8 dodatkowe pola służą do konfiguracji zakresu i trybu wejść analogowych AI14..AI16.




Rys. 6.9.7.1. Ustawienia Opóźniania wejść

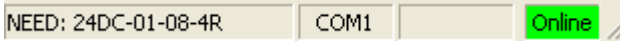
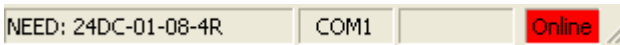
### 6.10. Podgląd zmiennych

Program PC Need został wyposażony w narzędzie do monitorowania wszystkich zmiennych w przekaźniku.

Okno Podgląd zmiennych wywołujemy poprzez Menu: **Narzędzia > Podgląd zmiennych** lub korzystając z przycisku . Skrót klawiaturowy F12.

Jeśli mamy połączenie z przekaźnikiem, to możemy korzystając z przycisku  (Odczyt) lub Menu: **Przekaźnik > Transmisja > Odczyt z Przekaźnika** (skrót klawiaturowy F6) uruchomić ciągły podgląd wpisanych zmiennych.

Połączenie z przekaźnikiem jest sygnalizowane na pasku stanu:

- Dla aktywnego podglądu zmiennych i trybu RUN 
- Dla aktywnego podglądu zmiennych i trybu STOP 

Wyboru zmiennych podlegających podglądowi dokonuje się poprzez wpisanie ich w kolumnie *Zmienna*. Aby wpisać nową zmienną należy kliknąć dwukrotnie lewym klawiszem myszy lub nacisnąć *Enter* po ustawieniu kursora w wybranym wierszu, w kolumnie *Zmienna*. Korzystamy z mnemoniki, jakiej używamy do pisania programów oraz dodatkowo:

POT – Potencjometr

AI7, AI8 – wartości napięć na zaciskach wejściowych odpowiednio I7, I8.

AI14, AI15, AI16 – wartości napięć na zaciskach wejściowych odpowiednio I14, I15, I16.

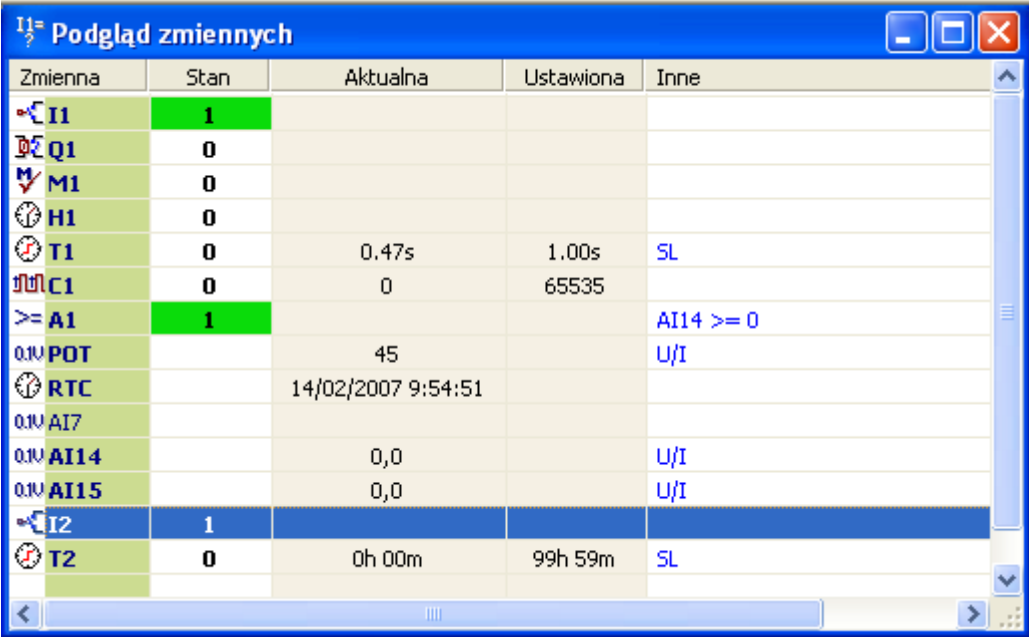
RTC – Zegar Czasu Rzeczywistego.

ASYM – asymetria faz

Zielony kolor podświetlenia w kolumnie *Zmienna* oraz napis Online na pasku stanu podświetlany na zielono oznacza pracę przekaźnika w trybie RUN.

Czerwony kolor podświetlenia sygnalizuje tryb STOP (zatrzymanie) przekaźnika.

W kolumnie *Stan* zmienne binarne o wartości „1” są podświetlane na zielono.



| Zmienna | Stan | Aktualna           | Ustawiona | Inne      |
|---------|------|--------------------|-----------|-----------|
| I1      | 1    |                    |           |           |
| Q1      | 0    |                    |           |           |
| M1      | 0    |                    |           |           |
| H1      | 0    |                    |           |           |
| T1      | 0    | 0,47s              | 1,00s     | SL        |
| C1      | 0    | 0                  | 65535     |           |
| A1      | 1    |                    |           | AI14 >= 0 |
| POT     |      | 45                 |           | U/I       |
| RTC     |      | 14/02/2007 9:54:51 |           |           |
| AI7     |      |                    |           |           |
| AI14    |      | 0,0                |           | U/I       |
| AI15    |      | 0,0                |           | U/I       |
| I2      | 1    |                    |           |           |
| T2      | 0    | 0h 00m             | 99h 59m   | SL        |

Rys. 6.10.1. Podgląd zmiennych.

Wpisaną zmienną możemy skasować lub zastąpić inną.


W kolumnie *Stan* przedstawiany jest odczytana wartość 0 lub 1 dla zmiennych binarynych.

W kolumnie *Aktualna* przedstawione są bieżące (aktualne) wartości *Timerów* i *Liczników* oraz wartość liczbowa dla POT i AI7, AI8, AI14, AI15, AI16

Dla RTC jest podawana data i czas w formacie: dzień/miesiąc/rok godzina:minuta:sekunda.

W kolumnie *Ustawiona* przedstawione są zadane (ustawione) wartości *Timerów* i *Liczników*.  
W kolumnie *Inne* podawane są dodatkowe informacje takie, jak tryb dla *Timera*, rodzaj komparacji dla *Komparatora*, itp.

W tabeli Podglądu jak na rys. 6.10.1. w wierszu, w którym wpisany jest T1 mamy informację, że *Timer* ma stan 0, aktualnie pozostało do odliczenia 0.47 s, ustawiony czas to 1 sekunda *Timer* jest ustawiony w trybie generacji impulsów (SL).



W kolumnie *Zmienna* można wpisywać także zmienne symboliczne, jeżeli ikonka  jest aktywna.

### 6.11. Podgląd drabinki LAD


Program PC Need umożliwia podgląd działania programu LAD w przekaźniku.

Jeśli mamy aktywne połączenie z przekaźnikiem, to możemy klikając prawym klawiszem myszy, w aktywnym oknie programu LAD rozwinąć menu z którego wybieramy opcje *Podgląd drabinki*.

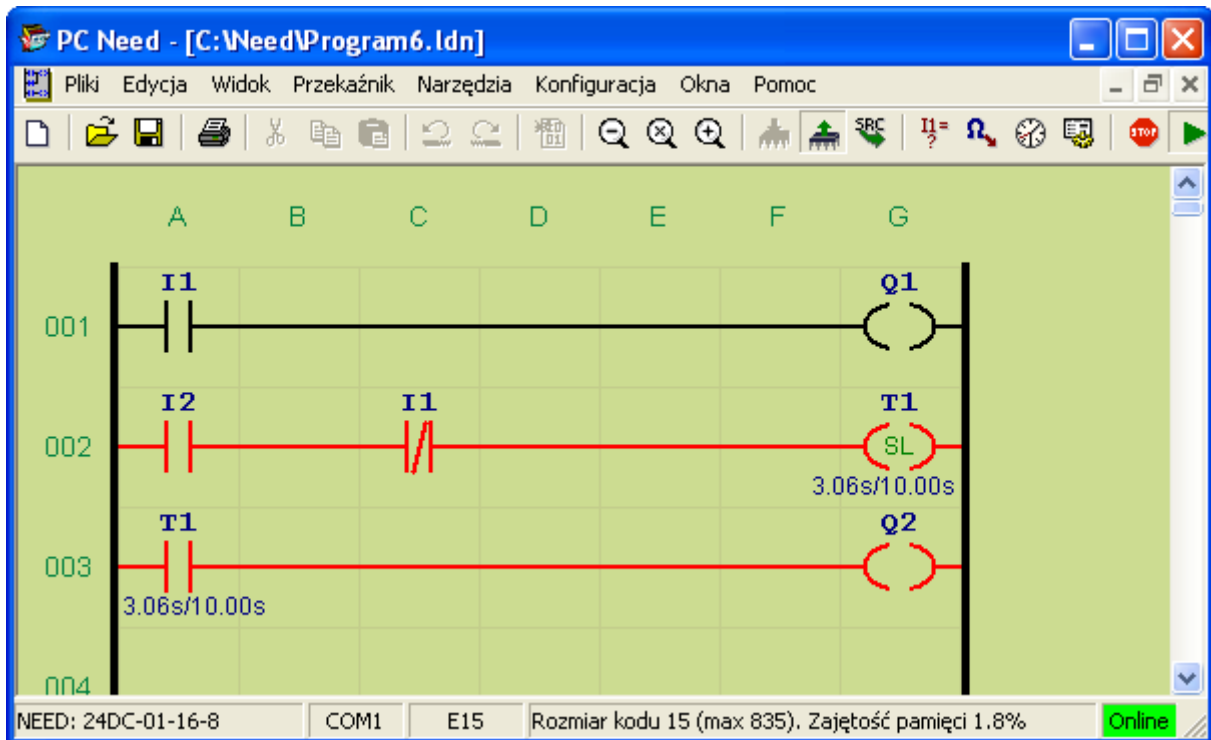
Połączenie z przekaźnikiem jest sygnalizowane pulsującym napisem *Online* na pasku stanu:

- Dla aktywnego podglądu drabinki i trybu RUN 
- Dla aktywnego podglądu drabinki i trybu STOP 

Przy braku lub wyłączonej komunikacji wyświetlany jest napis *Offline*.

W celu zakończenia podglądu drabinki należy nacisnąć prawy klawisz myszy na aktywnej drabince i wybrać . Alternatywą jest naciśnięcie klawisza F6.





Rys. 6.11.1. Okno - Podgląd drabinki.

W podglądzie drabinki na czerwono podświetlane są aktywne obwody lub elementy, a czarno nieaktywne. Na rys. 6.11.1 aktywne obwody to 002, 003. Nieaktywny jest obwód 001 gdyż stan wejścia I1 wynosi „0”. W obwodzie 002 wejście I1 jest typu NC, jego stan wynosi „0”. Z punktu widzenia LAD jest ono aktywne. Dodatkowo wyświetlane są informacje o wartości czasu *Timera* bieżącego i ustawionego. Dla *Liczników* podawany jest wartość zliczona i ustawiony próg.

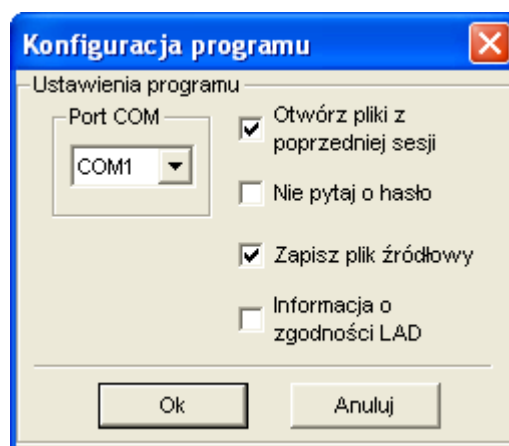
W trybie STOP drabinka jest nieaktywna, nie ma możliwości podglądu.

## 6.12. Hasło

W celu zablokowania dostępu osobom niepowołanym przekaźnik programowalny NEED może być zabezpieczony czterocyfrowym hasłem (0 do 9999).

Jeżeli chcemy używać hasła przy programowaniu Needa, to należy w **Menu > Konfiguracja > Program** odznaczyć opcję „Nie pytaj o hasło” rys 6.12.1.

Domyślne hasło to cyfra 0.



Rys. 6.12.1. Uruchomienie pytania o hasło

Hasło jest przechowywane w pamięci EEPROM przekaźnika. Reset przekaźnika przywraca domyślne hasło.

#### 6.12.1. Wprowadzanie hasła

Po uruchomieniu programu PC Need, jeśli nie była zaznaczona opcja „Nie pytaj o hasło”, w celu nawiązania komunikacji z przekaźnikiem NEED należy jednokrotnie podać hasło. Hasło podajemy korzystając z **Menu > Przekaźnik > Hasło > Wprowadź** lub wpisując hasło na żądanie przy obsłudze polecenia związanego z połączeniem z przekaźnikiem. Jeśli hasło nie było ustawione (hasło: 0) to wystarczy zaakceptować polecenie – będzie ono wykonane.

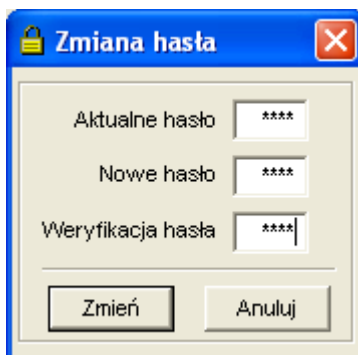


Rys. 6.12.2. Okno Wprowadzanie hasła.

#### 6.12.2. Zmiana hasła

W celu ustawienia lub zmiany istniejącego hasła należy wybrać polecenie: **Menu > Przekaźnik > Hasło > Zmień**. W oknie jak na rys. 6.12.2. wprowadzamy obowiązujące hasło (Aktualne hasło) oraz hasło jakie zamierzamy zastosować (Nowe hasło). Dodatkowo w polu „Weryfikacja hasła” powtarzamy nowe hasło w celu uniknięcia pomyłki przy wpisywaniu hasła.

Hasło jest zapisywane w pamięci przekaźnika.



Rys. 6.12.3. Okno Zmiana hasła.



**Uwaga:** Hasło jest przechowywane w pamięci przekaźnika. Reset przekaźnika kasuje hasło i ustawia hasło na domyślne (hasło = 0). Jednocześnie kasowane są ustawienia i pamięć programu.




**Uwaga:** W przypadku użycia do zaprogramowania przekaźnika pamięci zewnętrznej hasło w niej zawarte musi być takie jak w przekaźniku. W przypadku, gdy hasła będą różne przekaźnik nie będzie reagował na obecność pamięci zewnętrznej.

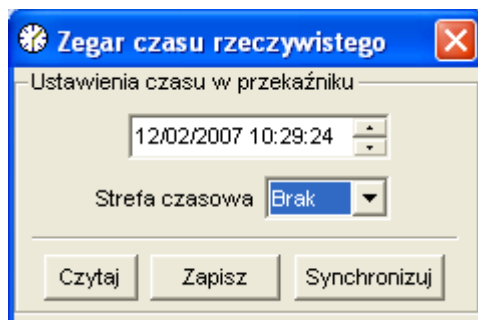
Hasło w pamięci zewnętrznej jest takie, jakie było aktualnie ustawione w programie PC Need podczas jej programowania.

### 6.13. Zegar czasu rzeczywistego (RTC)

Okno obsługi zegara RTC wywołujemy korzystając z **Menu > Przekaznik > Zegar**

(Ctrl+Shift+Z z klawiatury) lub korzystając z ikonki  na pasku narzędzi.

Możliwy jest podgląd aktualnego czasu w sterowniku – przycisk *Czytaj*, ustawienie dowolnej daty i czasu i skorzystanie z klawisza *Zapisz* oraz ustawienie czasu takiego jaki aktualnie ma urządzenie programujące (PC) przy pomocy opcji *Synchronizuj*.



Rys. 6.13.1 Okno Zegar czasu rzeczywistego.

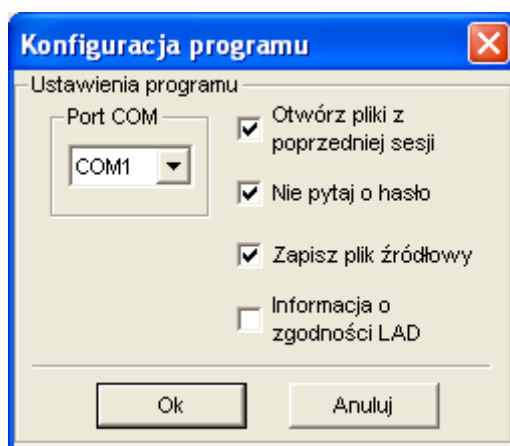
Pole wyboru *Strefa czasowa* umożliwia ustawienie automatycznej zmiany czasu zimowego na letni i odwrotnie w zależności od położenia geograficznego. Wybór *Brak* oznacza, że przekaznik nie będzie uwzględniał zmiany czasu letniego na zimowy i odwrotnie.

### 6.14. Kod źródłowy

Przekazniki w wersji NEED ...-16-8 umożliwiają zachowanie kodu źródłowego bezpośrednio w przekazniku. Niemożliwe jest załadowanie kodu do pamięci zewnętrznej.

Ładowanie programu źródłowego następuje w trakcie transmisji do przekaznika bezpośrednio po załadowaniu kodu wykonawczego i ustawień.

Ładowanie kodu źródłowego jest domyślną ustawioną opcją. Możemy zrezygnować z przechowywania kodu źródłowego w przekazniku odznaczając opcję „Zapisz plik źródłowy” w menu: **Konfiguracja > Program**.



Rys. 6.14.1. Okno Konfiguracja Programu.

Odczyt kodu źródłowego z przekaznika wykonujemy poprzez menu: **Przekaznik >**

**Transmisja > Odczyt kodu programu** lub korzystając z przycisku .

Po wczytaniu zawartości projektu PC Need automatycznie utworzy nowe okno w edytorze (LAD, STL) w jakim napisany był program umieszczony w przekazniku.

## 7. URUCHOMIENIE

### 7.1. Załączenie

#### 7.1.1. Czynności wstępne dla wersji AC



1. Sprawdzić prawidłowość podłączenia napięcia zasilającego:  
zacisk L: przewód fazowy 230V AC  
zacisk N: przewód zerowy
2. Sprawdzić prawidłowość podłączenia wejść i wyjść przełącznika;  
Uwaga: wejścia I1.. In sterowane przez przewód fazowy L
3. Ustawić przełącznik RUN/STOP w pozycję STOP.
4. Układy sterowane przez przełącznik programowalny zabezpieczyć przed dostępem niepowołanych osób – przy pierwszym uruchomieniu istnieje ryzyko niekontrolowanego działania maszyn (napędy, pompy, wentylatory) i urządzeń lub wystąpienia niebezpiecznych napięć na wejściach. Może to być spowodowane np. błędem w programie lub okablowaniu.

#### 7.1.2. Czynności wstępne dla wersji DC

1. Sprawdzić prawidłowość podłączenia napięcia zasilającego:
  - zacisk +24V DC: przewód zasilający dodatni zasilania 24V DC dla wersji 24DC
  - zacisk +12V DC: przewód zasilający dodatni zasilania 12V DC dla wersji 12DC
  - zacisk 0V: masa zasilania
2. Sprawdzić prawidłowość podłączenia wejść i wyjść przełącznika;  
Uwaga: wejścia I1.. In sterowane przez napięcie dodatnie względem zacisku 0V.
3. Ustawić przełącznik RUN/STOP w pozycję STOP.
4. Układy sterowane przez przełącznik programowalny zabezpieczyć przed dostępem niepowołanych osób – przy pierwszym uruchomieniu istnieje ryzyko niekontrolowanego działania maszyn (napędy, pompy, wentylatory) i urządzeń lub wystąpienia niebezpiecznych napięć na wejściach. Może to być spowodowane np. błędem w programie lub okablowaniu.

#### 7.1.3. Załączenie zasilania

1. Dołączyć zewnętrzne zasilanie do zacisków przełącznika programowalnego.
2. Sprawdzić działanie niezależnych elementów bezpieczeństwa (jeżeli są) – np. wyłącznik awaryjny zasilania.
3. Ocenić prawidłowość sygnalizacji diodami LED wejść przełącznika programowalnego.
4. Przeszawić przełącznik RUN/STOP w pozycję RUN.

Obserwować zachowanie układu – w wypadku nieprawidłowego działania sprawdzić układ połączeń i program sterujący.



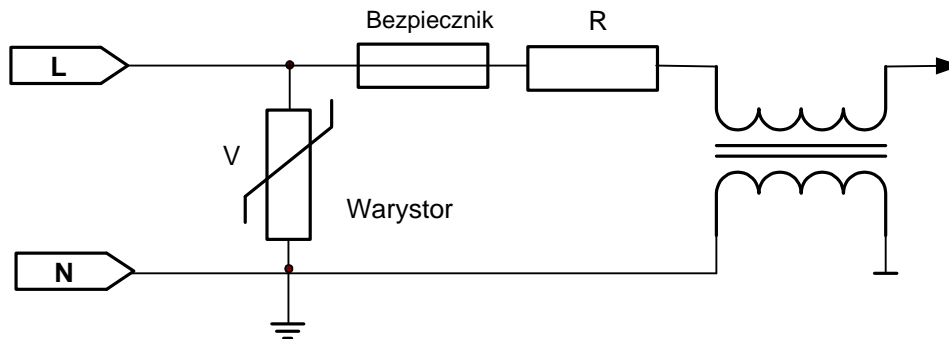
Od momentu przestawienia w tryb RUN, natychmiast uruchamiany jest program, który przejmuje kontrolę nad wyjściami.

## 8. INFORMACJE SPRZĘTOWE

### 8.1. Zasilanie przekaźnika

#### 8.1.1. Zasilanie przekaźnika 115/230 V AC

Schemat ideowy obwodu zasilania przekaźnika NEED przedstawiono na rys. 8.1.1



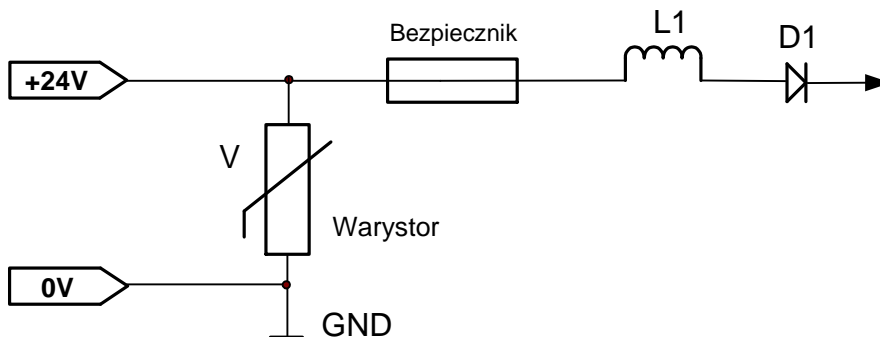
Rys.8.1.1. Schemat ideowy zasilania przekaźnika NEED AC



Układ zasilacza przekaźnika NEED nie jest odseparowany galwanicznie od zasilania sieci elektrycznej. Oznacza to, że w przypadku zamiany przewodów na zaciskach L i N na złączu komunikacyjnym mogą wystąpić napięcia groźne dla życia.

#### 8.1.2. Zasilanie przekaźnika 24 (12) V DC

Schemat ideowy obwodu zasilania przekaźnika NEED dla wersji 24V DC przedstawiono na rys. 8.1. Wersja 12 V DC różni się doбором elementów.



Rys.8.1.2. Schemat ideowy zasilania przekaźnika NEED DC



Układy zasilania przekaźnika w wersji DC są zabezpieczone przed odwrotnym podłączeniem napięcia zasilającego.

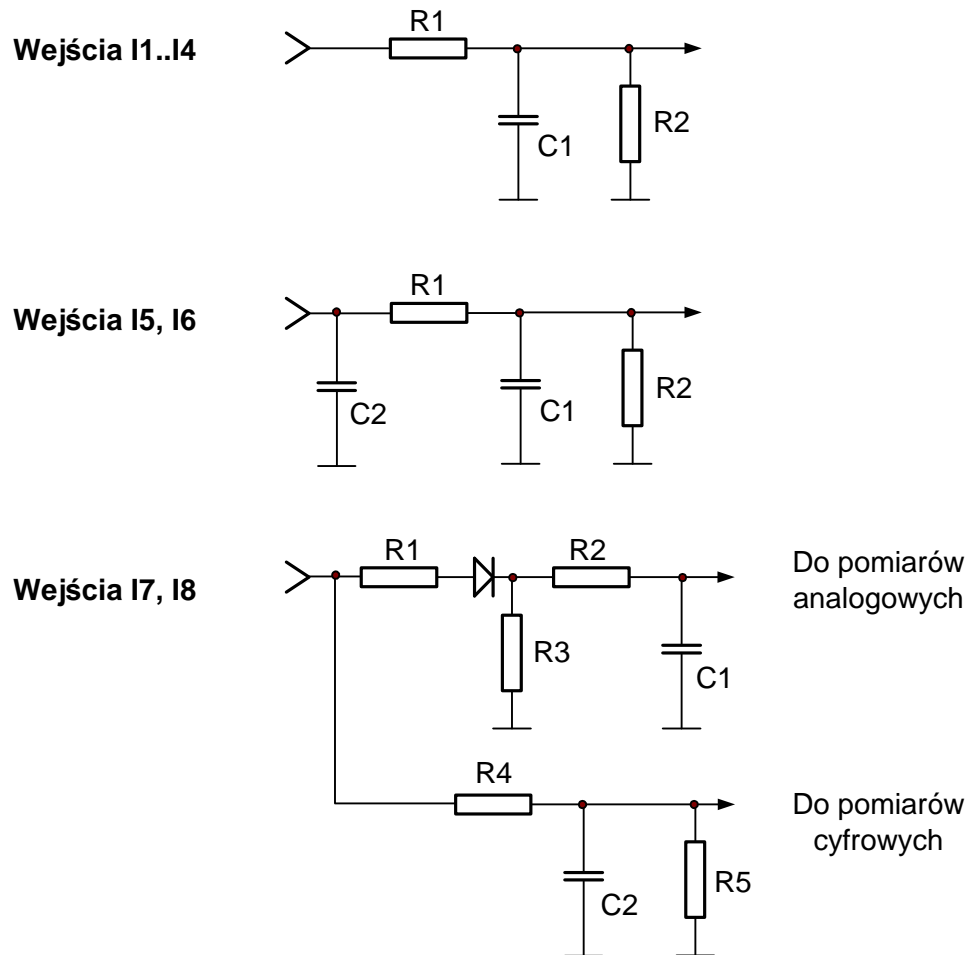
## 8.2. Wejścia

### 8.2.1. Wejścia 230 V AC

Schemat ideowy układów wejściowych przekaźnika NEED-230AC-x1-8-.. został przedstawiony na rys. 8.2.1 a NEED-230AC-x1-16-.. na rys. 8.2.2.

Wejścia analogowe to I7 i I8 dla wersji DC NEED..-x1-8-.. oraz I14, I15, I16 dla wersji NEED..-x1-16-...

Wejścia o zwiększonej odporności na zakłócenia to I5 i I6 dla wersji AC NEED..-x1-8-.. oraz I12, I13 dla wersji AC NEED..-x1-16-...



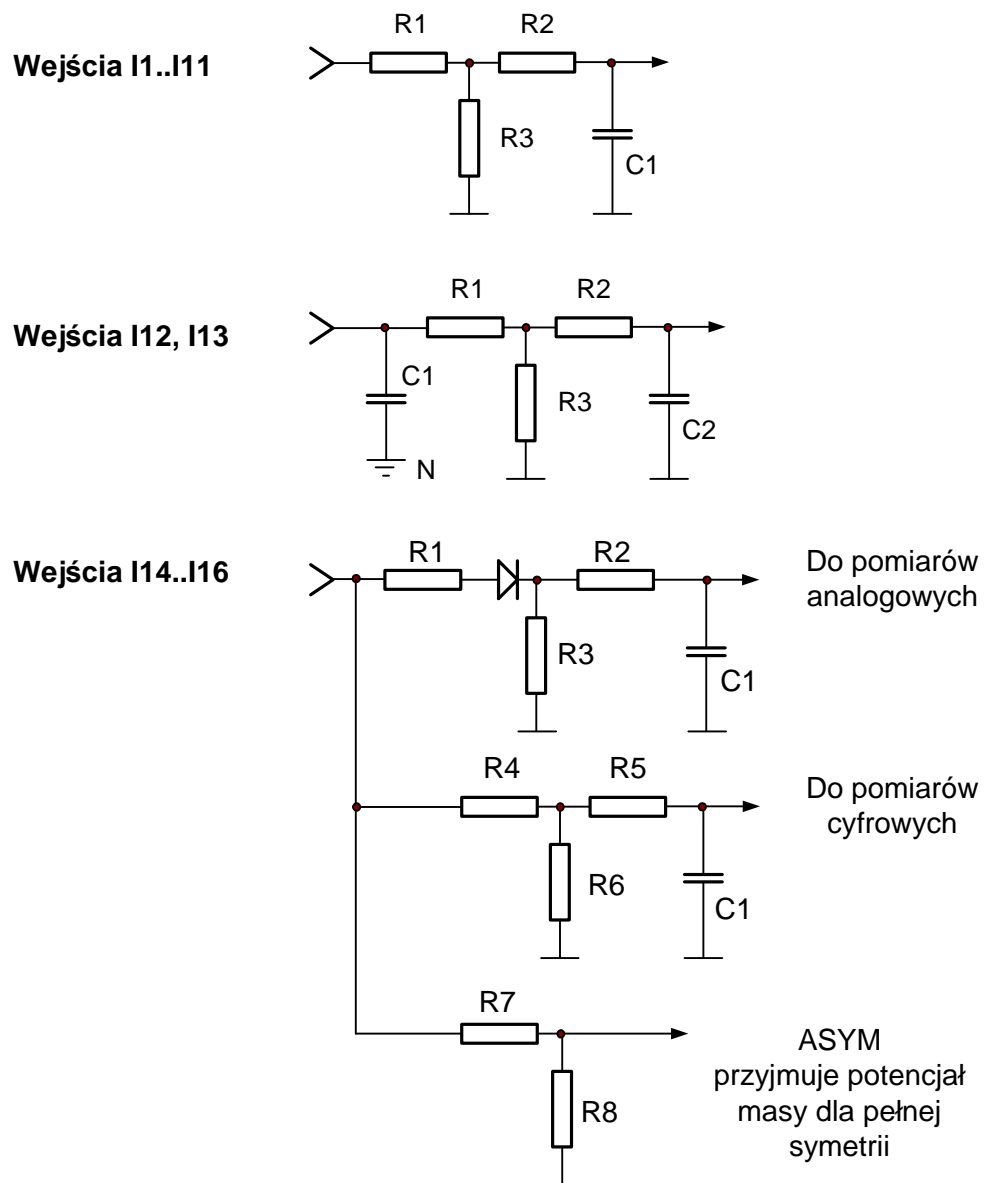
Rys. 8.2.1. Schemat ideowy wejść przekaźnika NEED-230AC-x1-8-4.



Wejścia o zwiększonej odporności na zakłócenia posiadają kondensator (filtr przeciwzakłócenia), który pozwala dołączać do nich długie przewody. Wejścia I7, I8 pełnią rolę wejść cyfrowych i analogowych – patrz rozdział „4.11. Komparator – Wejście analogowe”.



Wejścia przekaźnika NEED nie są odseparowane galwanicznie od zasilania sieci elektrycznej.



Rys. 8.2.2. Schemat ideowy wejść przekaźnika NEED-230AC-x1-16-8.



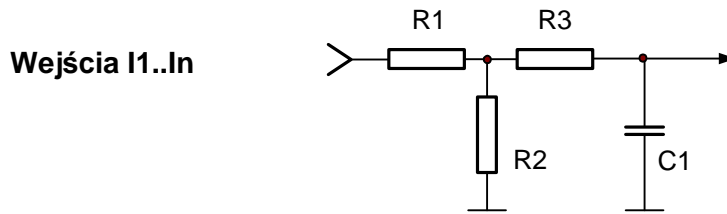
Wejścia o zwiększonej odporności (I11, I12) na zakłócenia posiadają kondensator (filtr przeciwzakłóceńowy), który pozwala dołączać do nich długie przewody. Wejścia I14, I15, I16 pełnią rolę wejść cyfrowych i analogowych – patrz rozdział „4.11. Komparator – Wejście analogowe”.



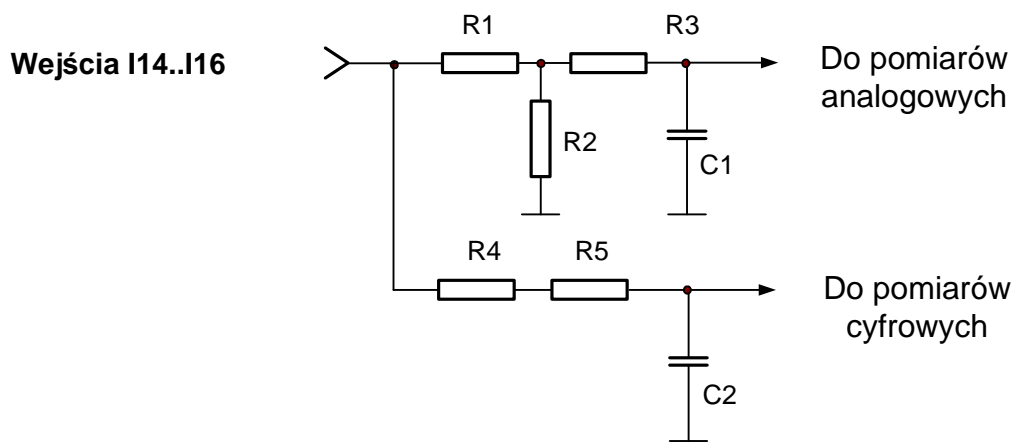
Wejścia przekaźnika NEED nie są odseparowane galwanicznie od zasilania sieci elektrycznej.

### 8.2.2. Wejścia 24 (12) V DC

Schemat ideowy układów wejściowych przekaźnika NEED DC został przedstawiony na rys. 8.2.3. Wszystkie wejścia cyfrowe i cyfrowo-analogowe dla wersji NEED-..DC-x1-8-4 mają podobny układ połączeń. Wejścia analogowo-cyfrowe mają inny układ połączeń dla wersji NEED-..DC-x1-16-8 – pokazano to na rys. 8.2.4.



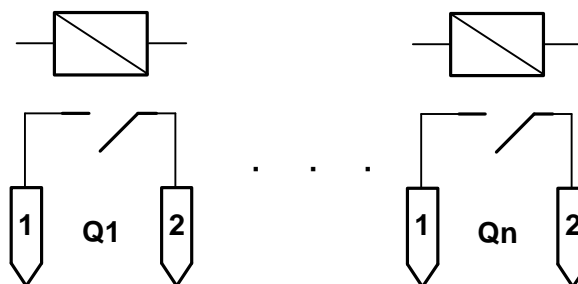
Rys. 8.2.3. Schemat ideowy wejść przekaźnika NEED DC.



Rys. 8.2.4. Schemat ideowy wejść analogowo-cyfrowych przekaźnika NEED-24DC-x1-16-8.

### 8.3. Wyjścia

Schemat ideowy układów wyjściowych przekaźnika NEED został przedstawiony na rys. 8.3.



Rys. 8.3. Schemat ideowy wyjść przekaźnika NEED.

Wyjścia przekaźnika NEED to bezpotencjałowe styki przekaźników.

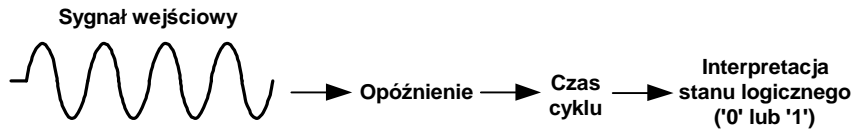


Wyjścia przekaźnika NEED są odseparowane galwanicznie od wejść oraz zasilania sieci.



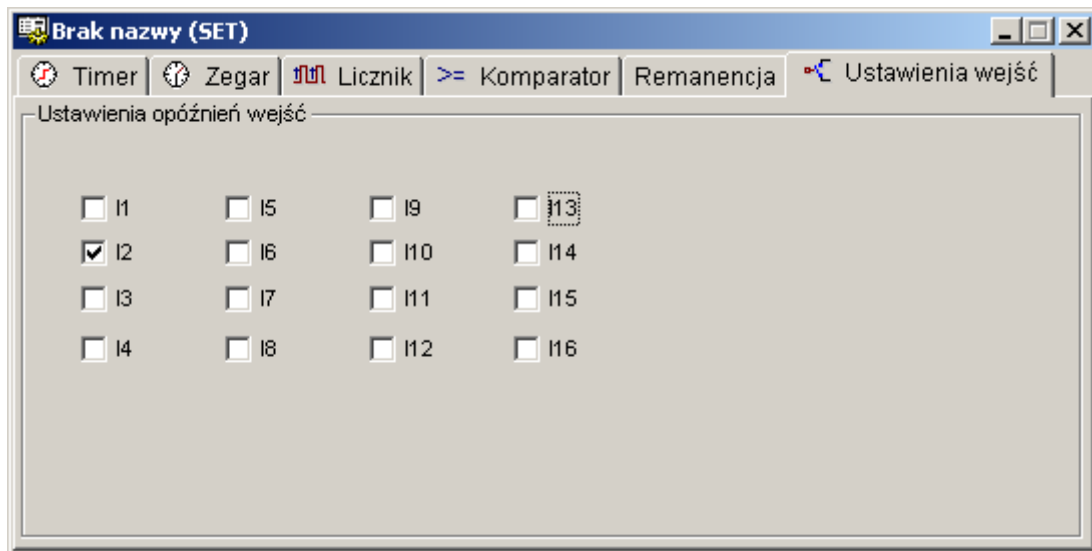
#### 8.4. Opóźnienie wejść

W wielu problemach sterowania bardzo kłopotliwą rzeczą jest drganie styków np. przełącznika. W przekaźniku programowalnym NEED można ustawić odpowiednie opóźnienie wejść tak, aby wyeliminować te niekorzystne zjawiska. Przetwarzanie sygnałów wejściowych w przekaźniku NEED pokazuje rys. 8.4.1.



Rys. 8.4.1. Przetwarzanie sygnałów wejściowych w przekaźniku NEED.

W przekaźniku NEED opóźnienie wejść można ustalać za pomocą konfiguracji programowej (patrz rozdział „6. Instalacja i opis oprogramowania”). Przykładowe okienko konfiguracyjne programu PC Need z ustawieniem opóźnienia dla wejścia I2 przedstawiono na rys. 8.4.2.



Rys. 8.4.2. Przykładowa konfiguracja opóźnienia dla wejść.

##### 8.4.1. Opóźnienia wejść dla przekaźnika NEED-230AC-x1-..

Jeśli wejścia ustawimy bez opóźnienia, to przekaźnik NEED, co 20ms (jedna, dodatnia połówka przebiegu sinusoidalnego dla częstotliwości napięcia zasilającego 50Hz) – będzie sprawdzał stan sygnału wejściowego i od razu po sprawdzeniu, będzie dokonywał interpretacji czy poziom napięcia pojawiający się na wejściu zaliczyć do stanu wysokiego lub niskiego. Czyli maksymalny czas interpretacji stanu sygnału wejściowego bez opóźnienia wynosi:

20ms + czas cyklu programu.

W maksymalnie tak długim czasie, sygnał na wejściu bez opóźnienia, może być „zauważony” i prawidłowo zinterpretowany przez przekaźnik.

Jeśli wejścia ustawimy z opóźnieniem, to przekaźnik NEED co ok. 20ms (dla częstotliwości napięcia zasilającego 50Hz) będzie dokonywał interpretacji sygnału wejściowego. Jeśli po trzecim sprawdzeniu stan na badanym wejściu nie zmieni się, to przekaźnik dokona odpowiedniej interpretacji, zaliczając poziom napięcia do stanu niskiego lub wysokiego. Czyli maksymalny czas interpretacji stanu sygnału wejściowego z opóźnieniem wynosi:

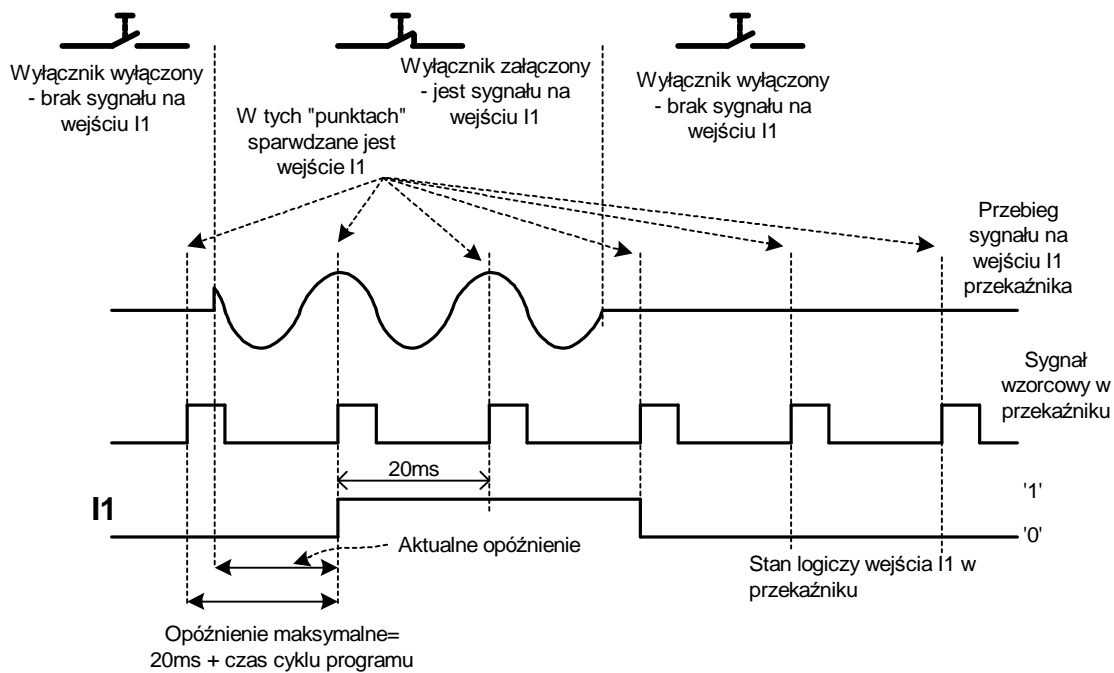
60ms + czas cyklu programu.

W maksymalnym czasie 60ms (jednak nie krótszym niż 40ms), sygnał na wejściu NEED'a,

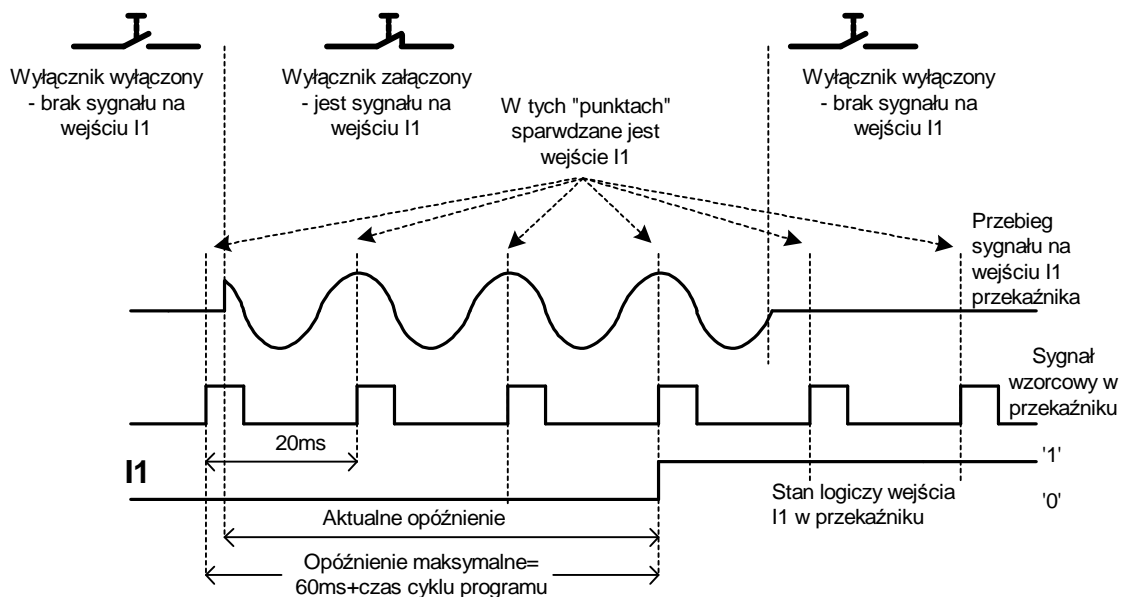
może być „zauważony” i prawidłowo zinterpretowany przez przełącznik.  
 W tabeli 8.1. przedstawiono czasy opóźnień dla wejść przełącznika programowalnego.  
 Tab.8.1. Czasy opóźnień wejść przełącznika NEED.

| Częstotliwość napięcia zasilającego | Maksymalny czas opóźnienia sygnału wejściowego |                      |
|-------------------------------------|--|----------------------|
|                                     | Opóźnienie załączone                           | Opóźnienie wyłączone |
| f=50Hz                              | 60ms+czas cyklu                                | 20ms+czas cyklu      |
| f=60Hz                              | 49,8ms+czas cyklu                              | 16,6ms+czas cyklu    |

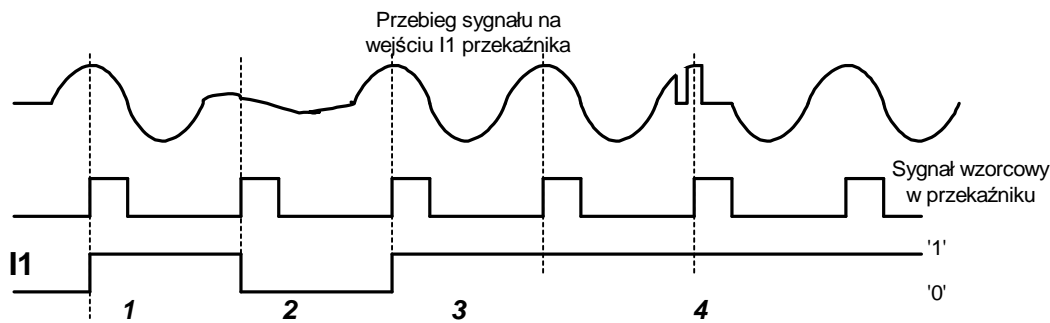
Na rys. 8.4.3. oraz rys. 8.4.4. przedstawiono interpretację stanów logicznych wejść przełącznika NEED-230AC-x1-..



Rys.8.4.3. Interpretacja stanu logicznego wejścia I1 przełącznika dla nieustawionego czasu opóźnienia – NEED-230AC-x1-..



Rys.8.4.4. Interpretacja stanu logicznego wejścia I1 przełącznika dla ustawionego czasu opóźnienia – NEED-230AC-x1-..



Rys.8.4.5. Przykładowa interpretacja stanu logicznego wejścia I1 przekaźnika dla nieustawionego czasu opóźnienia – NEED-230AC-x1-..

Dla przebiegu sygnału wejściowego, jak na rys. 8.4.5., możemy wyróżnić następujące etapy: sygnał wejściowy jest w stanie wysokim (1), więc przekaźnik zinterpretuje go jako logiczną jedynkę. Jednak, jeśli przekaźnik nie wykryje w następnych 20ms prawidłowego poziomu sinusoidy, wtedy zmieni stan swojego wejścia na zero logiczne (2). Po kolejnych 20ms sygnał wejściowy zostaje zinterpretowany jako stan wysoki (3). Impulsy krótkie mogą być detekowane prawidłowo, jeśli „natrafiają” na odpowiedni moment w interpretacji sygnału wejściowego przez przekaźnik (4).

#### 8.4.2. Opóźnienia wejść dla przekaźnika NEED-24DC-x1-..., NEED-12DC-x1-..

Jeśli wejścia ustawimy bez opóźnienia, to przekaźnik NEED, raz na obieg pętli programu, będzie sprawdzał stan sygnału wejściowego i od razu po sprawdzeniu, będzie dokonywał interpretacji czy poziom napięcia pojawiający się na wejściu zaliczyć do stanu wysokiego lub niskiego. Czyli maksymalny czas interpretacji stanu sygnału wejściowego bez opóźnienia wynosi:

maksymalny czas cyklu programu.

W maksymalnie tak długim czasie, sygnał na wejściu bez opóźnienia, może być „zauważony” i prawidłowo zinterpretowany przez przekaźnik.

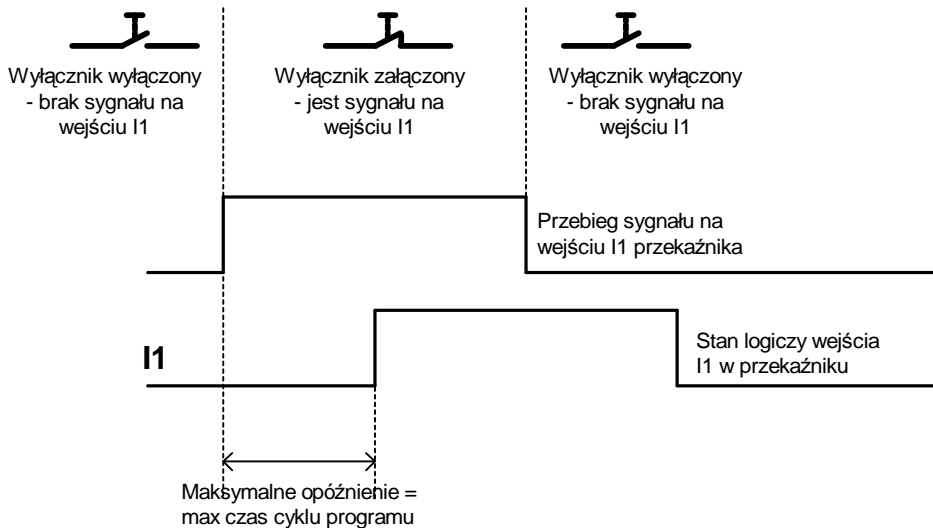
Jeśli wejścia ustawimy z opóźnieniem, to przekaźnik NEED co, 21ms będzie dokonywał interpretacji sygnału wejściowego. Czyli maksymalny czas interpretacji stanu sygnału wejściowego z opóźnieniem wynosi:

21ms + czas cyklu programu.

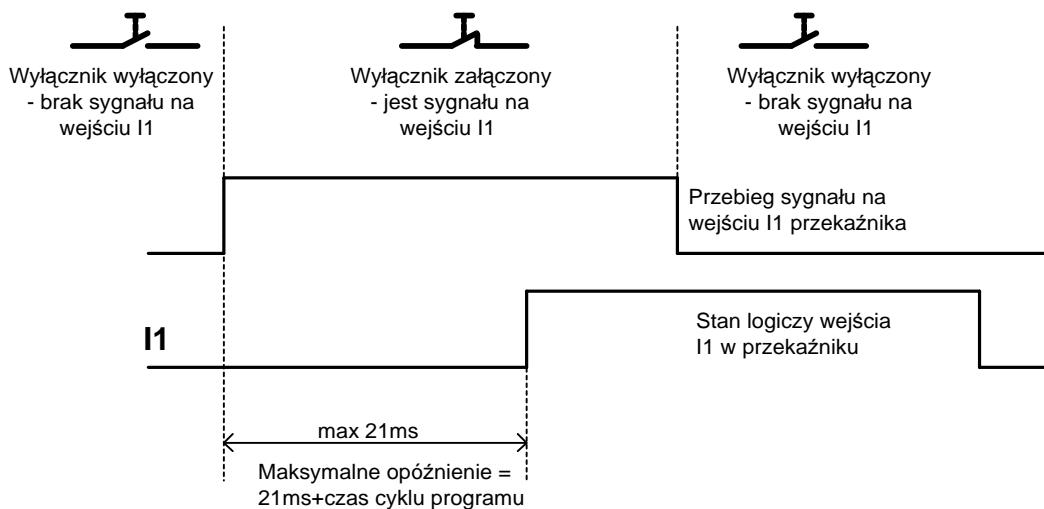
W maksymalnym czasie 21ms sygnał na wejściu NEED’a, może być „zauważony” i prawidłowo zinterpretowany przez przekaźnik.



Jeżeli w trakcie odmierzenia czasu opóźnienia, dla przekaźników NEED-24DC-x1-..., NEED-12DC-x1-.., sygnał wejściowy ulegnie zmianie, to liczenie czasu opóźnienia rozpoczyna się od początku.



Rys.8.4.6. Przykładowa interpretacja stanu logicznego wejścia I1 przekaźnika dla nieustawionego czasu opóźnienia – NEED-24DC-x1-..., NEED-12DC-x1-...



Rys.8.4.7. Przykładowa interpretacja stanu logicznego wejścia I1 przekaźnika dla ustawionego czasu opóźnienia – NEED-24DC-x1-..., NEED-12DC-x1-...



Przetwarzanie sygnałów wejściowych w przekaźniku NEED odbywa się z opóźnieniem.

### 8.5. Opóźnienie wyjść

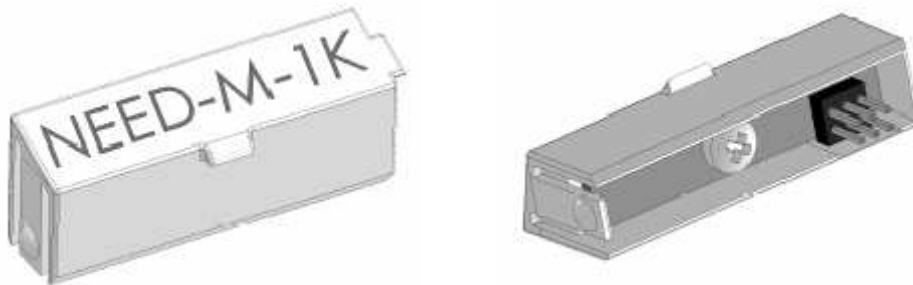
Wyjścia w przekaźniku NEED nie są opóźniane – są ustawiane tak szybko, jak jest to możliwe. Jednakże należy wziąć pod uwagę opóźnienia wynikające z zastosowanych elementów wyjściowych, sterujących np. dla wersji *NEED-230AC-x1-08-4* czas zadziałania wyjść to:

Czas zadziałania przekaźnika wyjściowego + czas cyklu.

## 9. PAMIĘĆ ZEWNĘTRZNA

### 9.1. Karta pamięci

Dla zwiększenia funkcjonalności przekaźnika NEED dostępna jest karta pamięci zewnętrznej NEED-M-1K. Jest to moduł EEPROM o pojemności 1 KBajt. Pamięć ta może być wykorzystywana do przepisywania programu do przekaźnika NEED bez używania komputera.

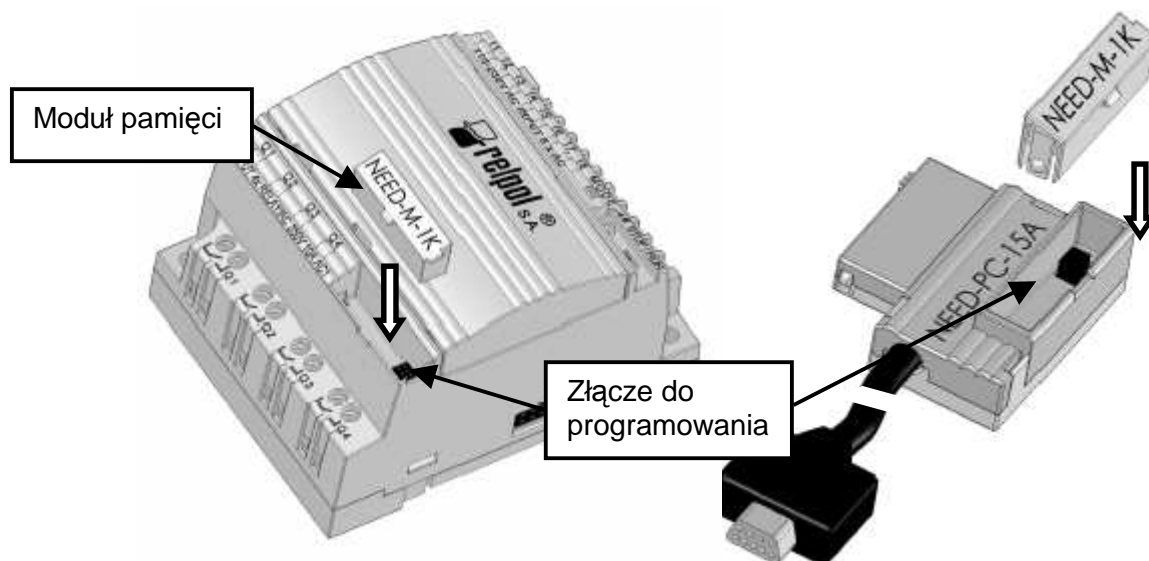


Rys. 9.1.1. Moduł pamięci zewnętrznej – widok z góry i z dołu.

Moduł programujemy korzystając z przewodu do programowania przekaźnika NEED. W tym celu umieszczamy moduł w przeznaczonym do tego złączu, we wtyczce kabla programującego (rys. 9.1.2).

Możliwy jest także odczyt z partycji pamięci ustawień tam zapisanych.

Zaprogramowaną kartę pamięci po wyjęciu z wtyczki kabla programującego umieszczamy w przekaźniku zamiast wtyku przewodu programującego – przewód programujący i pamięć korzystają z tego samego złącza w przekaźniku NEED.



Rys. 9.1.2. Miejsce instalacji modułu pamięci zewnętrznej w przekaźniku i złączu kabla.



**Uwaga:** Przewodu z włożonym modułem pamięci nie należy podłączać do złącza programującego przekaźnika i używać do programowania przekaźnika.



**Uwaga:** Używanie przewodu do programowania z włożonym modułem pamięci może doprowadzić do niekontrolowanego przesycenia danych do przekaźnika lub modułu pamięci.

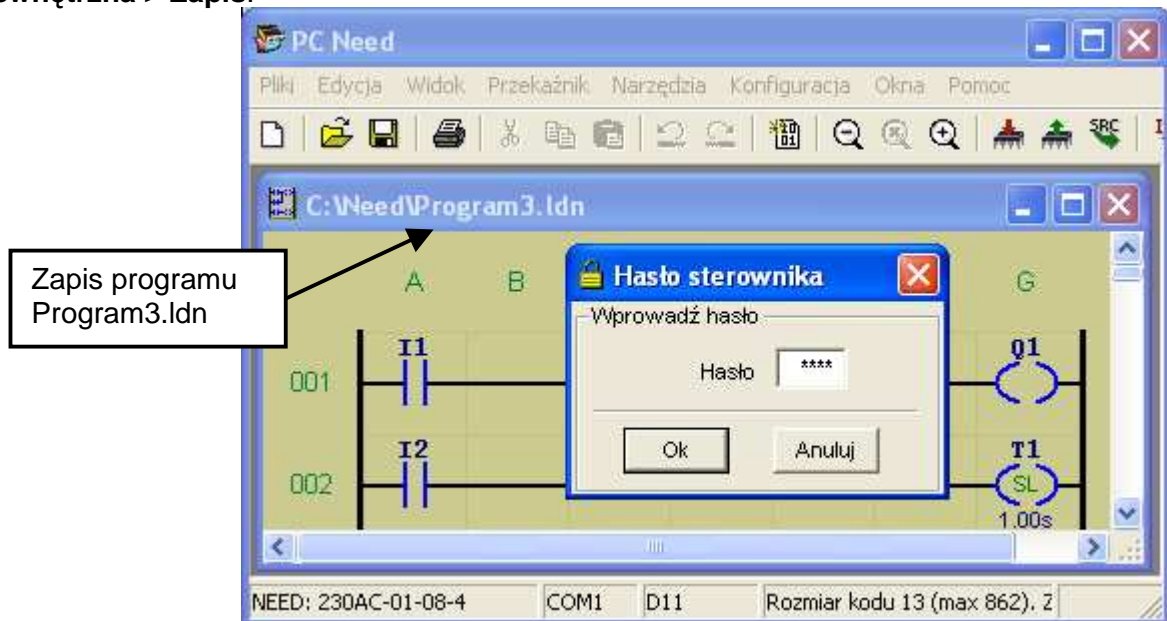
## 9.2. Organizacja pamięci

Pamięć na karcie została podzielona na 2 partycje – jedna służy do przechowywania kodu programu, druga do przechowywania ustawień. Przy programowaniu karty pamięci możemy dokonać wyboru, które partycje są aktywne. Jeśli partycja jest aktywna, to zapisane w niej dane są przepisywane do pamięci przekaźnika NEED. Możliwe są więc sytuacje, że wgrywamy tylko kod programu, wgrywamy tylko nowe ustawienia, wgrywamy kod programu i ustawienia. Jeśli obie partycje będą nieaktywne, to przekaźnik nie załaduje żadnych danych do swojej wewnętrznej pamięci.

## 9.3. Programowanie pamięci

### 9.3.1. Zapis programu

Jeśli mamy podłączoną kartę pamięci do złącza przewodu programującego, to ustawiamy jako aktywne (na wierzchu, niebieskie podświetlenie nagłówka) okno programu, którego kod chcemy zapisać do pamięci. Następnie wykonujemy komendę **Przekaźnik > Pamięć zewnętrzna > Zapis**.



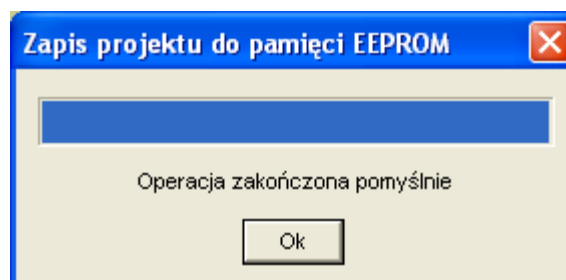
Rys. 9.3.1.1. Zapis Programu.

Po otwarciu okna *Zapis programu* ustawiamy hasło zgodne z ustawionym w przekaźniku a następnie naciskamy przycisk *Start*. Pytanie o hasło ma miejsce tylko wtedy, gdy nie była ustawiona opcja „Nie pytaj o hasło” w konfiguracji programu.



**Uwaga:** Jeśli wpisane hasło przy programowaniu pamięci jest inne niż hasło w przekaźniku NEED, wtedy program nie jest przepisywany do wewnętrznej pamięci przekaźnika programowalnego.

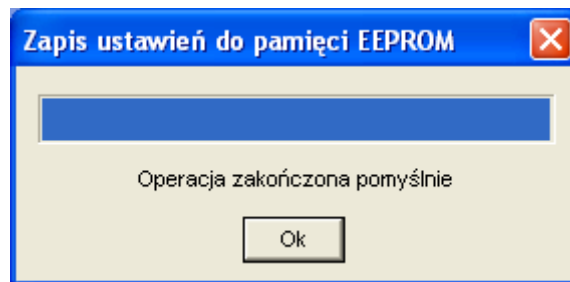
Po wykonaniu operacji otrzymamy komunikat o zakończeniu zapisu programu.



Rys. 9.3.1.2. Okno Zapis programu.

### 9.3.2. Zapis ustawień

Jeśli mamy podłączoną kartę pamięci do złącza przewodu programującego, to ustawiamy jako aktywne (na wierzchu, niebieskie podświetlenie nagłówka) okno Ustawienia. Następnie wykonujemy komendę **Przełącznik > Pamięć zewnętrzna > Zapis**.

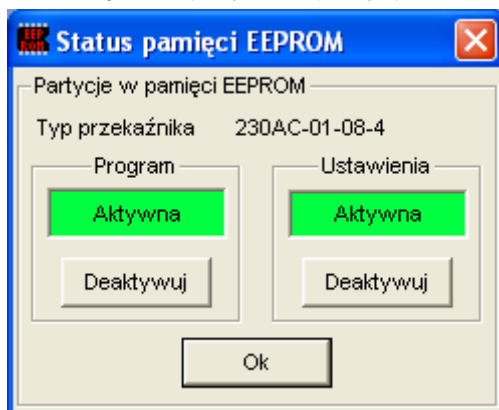


Rys. 9.3.2.1. Okno zapis Ustawień.

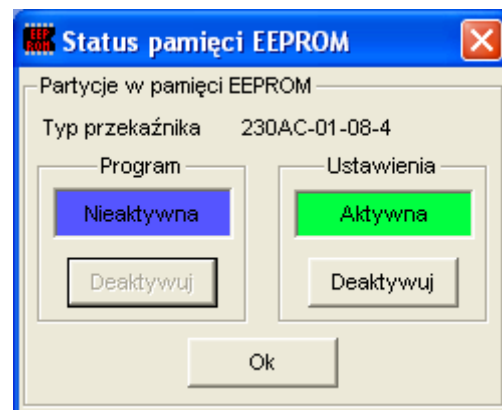
Po otwarciu okna *Zapis ustawienia przełącznika* decydujemy się czy zabezpieczamy ustawienia hasłem a następnie naciskamy przycisk *Start*. Podane hasło musi być zgodne z hasłem wpisanym do przełącznika. W innym przypadku pamięć nie zostanie skopiowana. Po wykonaniu zapisu otrzymamy komunikat o zakończeniu zapisu ustawień.

### 9.3.3. Status pamięci EEPROM

Opcję kontroli statusu pamięci wywołujemy: **Przełącznik > Pamięć zewnętrzna > Status**. Możemy uczynić nieaktywnymi – przycisk *Zdeaktywuj* – partycje *Program* lub *Ustawienia* lub obie. Po deaktywacji np. partycji *Program* (rys. 9.3.3.2) do przełącznika NEED skopiowane zostaną dane jedynie z partycji *Ustawienia*.



Rys. 9.3.3.1. Okno Aktywne partycje.



Rys. 9.3.3.2. Okno Nieaktywna partycja.



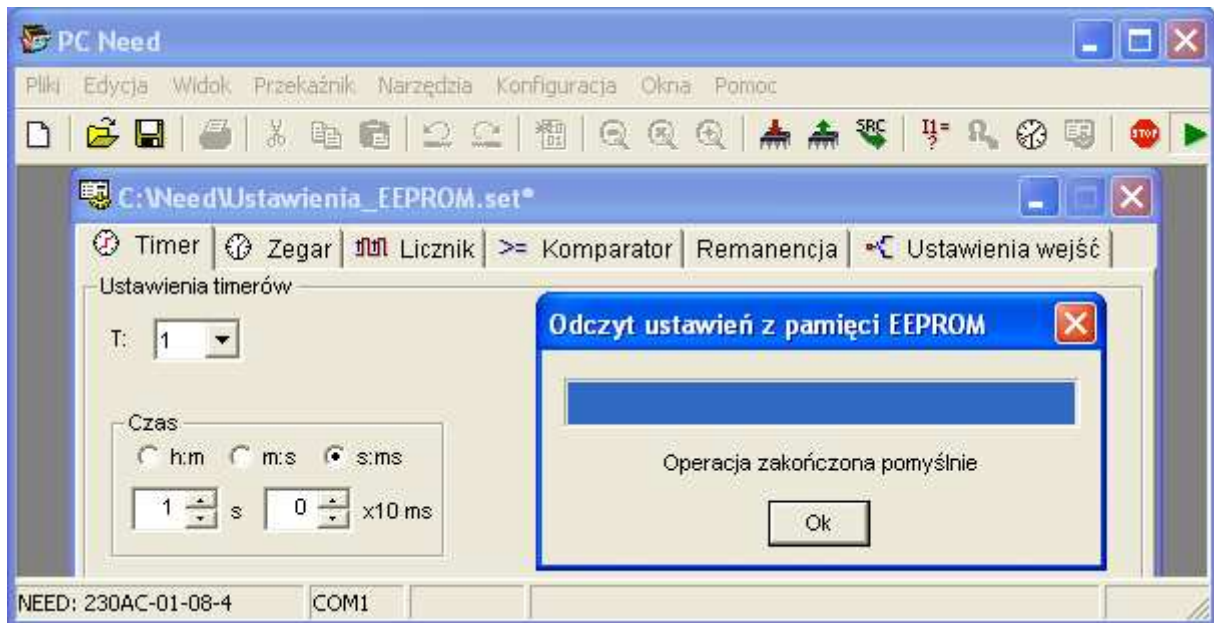
**Uwaga:** Partycję aktywuje się poprzez wgranie nowej zawartości.

### 9.3.4. Odczyt ustawień

Możliwy jest odczyt partycji *Ustawienia* z karty pamięci.

W tym celu należy ustawić w obszarze roboczym programu PC Need jako aktywne okno Ustawienia, do którego odczytane dane mają trafić. W przykładzie poniżej utworzony został do tego celu plik *Ustawienia\_EEPROM.set*. Po wykonaniu komendy **Przełącznik > Pamięć zewnętrzna > Odczyt** – okno poniżej – dane z partycji zostaną skopiowane do pliku z rozszerzeniem *.set*.





Rys. 9.3.4.1. Okno Odczyt ustawień.

#### 9.4. Współpraca karty pamięci z przełącznikiem NEED



Kartę pamięci należy wkładać przy odłączonym napięciu zasilającym przełącznik NEED.



Na porcie komunikacyjnym dla wersji 115/230V AC mogą występować napięcia niebezpieczne dla zdrowia i życia.



Wkładanie karty przy załączonym zasilaniu przełącznika grozi uszkodzeniem układu pamięci i przełącznika

1. Przy odłączonym zasilaniu należy umieścić moduł karty pamięci w złączu programującym przełącznika NEED.
2. Po załączeniu zasilania modułu przełącznika następuje kopiowanie zawartości pamięci (aktywnych partycji) do pamięci przełącznika NEED. O kopiowaniu informuje migająca na czerwono dioda MODE.
3. Po skopiowaniu zawartości, przełącznik ustawia tryb pracy zależnie od położenia przełącznika trybu pracy. Jeśli ustawiony jest tryb RUN, to przechodzi automatycznie do wykonywania programu.



**Uwaga:** Zawartość pamięci jest kopiowana jednokrotnie w momencie załączenia zasilania. Po przekopiowaniu moduł pamięci może być usunięty ze złącza przełącznika.



**Uwaga:** Przepisanie danych z pamięci do sterownika jest możliwe gdy:

- hasło jest prawidłowe,
- pamięć posiada dane dla właściwego typu przełącznika,
- partycje są aktywne,
- dane w przełączniku są inne niż w pamięci.



**Uwaga:** Pamięć zewnętrzna nie umożliwia przenoszenia kodu źródłowego LAD/STL do przełącznika.

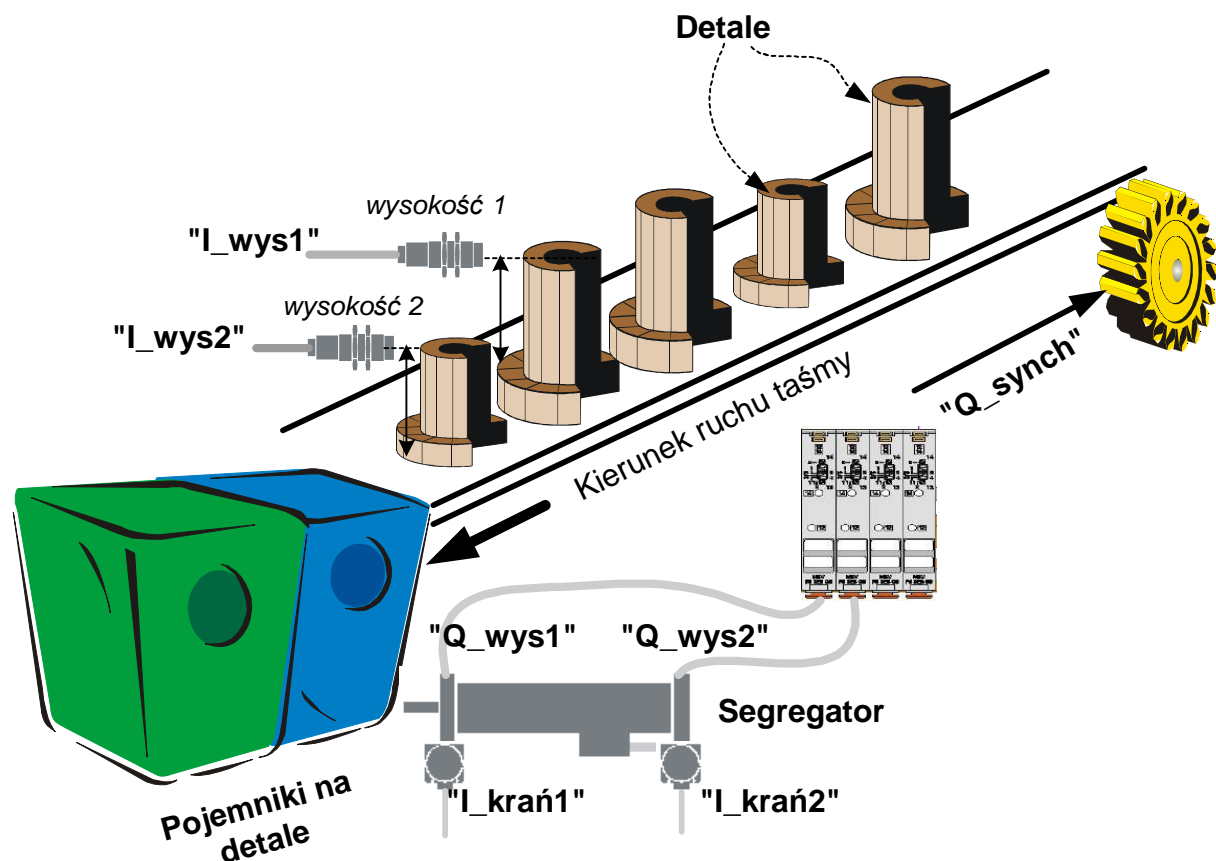


## 10. PRZYKŁADY ZASTOSOWAŃ

### 10.1. Ocena wysokości detalu

Często w procesie produkcyjnym zachodzi konieczność segregacji detali ze względu na ich wymiary geometryczne. Można to robić ręcznie, mierząc jakiś wymiar lub automatycznie – stosując przełącznik programowalny NEED oraz kilka czujników zewnętrznych.

Jeśli przyjmimy np. dwie kategorie wysokości detali, to dla skompletowania całego sterowania wystarczy przełącznik programowalny NEED oraz dwa czujniki detekujące w odpowiedni sposób zadany wymiar geometryczny.



Rys. 10.1.1. Detekcja wysokości detali.

Opis zadania:

Należy dokonać segregacji detali, przesuujących się na podajniku taktowym, ze względu na ich wysokość.

Dobór sprzętu:

- 1) Dla prawidłowego wykonania zadania należy dobrać dwa czujniki o odpowiednim zasięgu. Do detekcji wysokości (jeśli są to detale metalowe) można zastosować czujniki indukcyjne. Nazwijmy czujnik detekujący wysokość 1 (detal wysoki) „I\_wys1”, a czujnik detekujący wysokość 2 (detal niski) „I\_wys2”. Zakładamy, że ruch taśmy jest synchronizowany, więc potrzebny jest jeszcze sygnał zezwalający na ponowny przesuw taśmy – nazwijmy go „Q\_synch”.
- 2) Segregator – może to być siłownik sterowany zaworem elektromagnetycznym impulsowym (po podaniu sygnału sterującego na jedną cewkę elektromagnesu zawór pozostaje w zajmowanym położeniu również po zaniku tego sygnału, aż do chwili podania sygnału na drugą cewkę), na którym zamocowane będą dwa pojemniki.

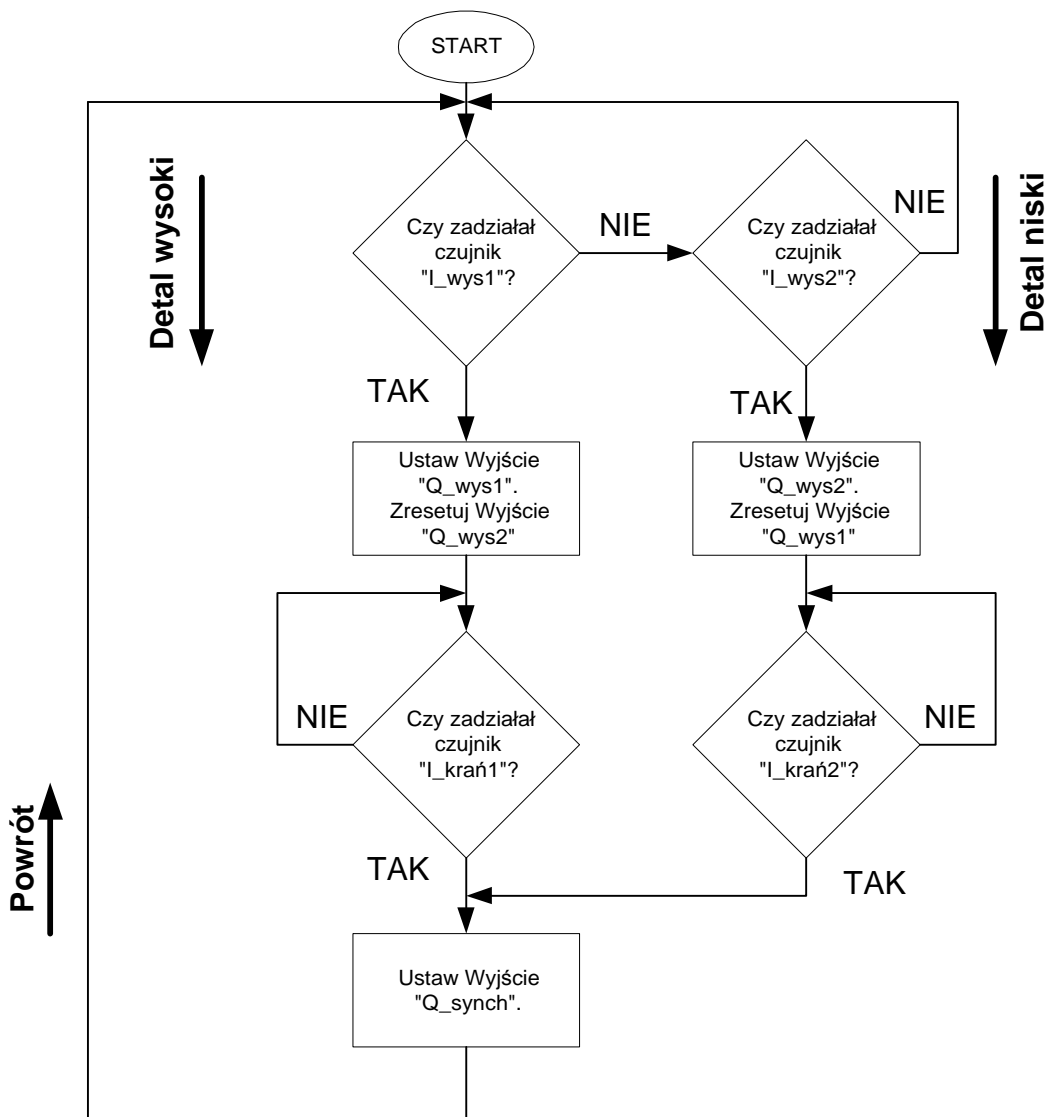
Pozycje krańcowe siłownika oznaczają będą: „Wysokość 1 detali”, „Wysokość 2 detali”. Nazwijmy wyjścia sterujące elektrozaworami – „Q\_wys1” oraz „Q\_wys1”. Odpowiadające im czujniki krańcowych pozycji „I\_krań\_1”, „I\_krań\_2”.

3) Przekaznik programowalny: potrzebujemy 4 Wejścia, 3 wyjścia.

Na rys. 10.1.1. przedstawiono idee analizowanego zadania, a na rys. 10.1.2. przykładowy schemat połączeń elektrycznych.

Algorytm:

Na początek będziemy sprawdzać wysokość detali. Jeśli zadziała czujnik „I\_wys1” to wiemy na pewno, że jest to detal wyższy. Jeśli zadziała czujnik „I\_wys2” i nie zadziała czujnik „I\_wys1”, to będzie to detal niższy. W zależności od wysokości detalu przesuwamy (załączamy wyjście połączone do elektrozaworu) pojemnik albo na detale wyższe, albo na detale niższe. Po całej operacji (gdy pojemnik jest ustawiony na właściwy typ detalu) dajemy sygnał synchronizacji do przesuwu taśmy. Dokładny algorytm działania przedstawiono poniżej.



Zauważmy, że nasz program „nigdzie się nie zatrzymuje”! Sterownik nie czeka na żaden sygnał startu – program jest po prostu przetwarzany przez procesor od pierwszej linii do ostatniej.

Program

Uporządkujmy naszą konfigurację sprzętową:

| Adres | Wejścia   | Adres | Wyjścia   |
|-------|-----------|-------|-----------|
| I1    | „I_wys1”  | Q1    | „Q_wys1”  |
| I2    | „I_wys2”  | Q2    | „Q_wys2”  |
| I3    | „I_krań1” | Q3    | „Q_synch” |
| I4    | „I_krań2” |       |           |

Spróbujmy przełożyć teraz algorytm na język programowania.

## STL

```
//Detekcja detalu wysokiego
A I1          //Jeżeli I1=1, to ustaw Q1. Jeżeli I1=0 to nic nie rób
AN T1        //Timer T1 musi być w stanie niskim
S Q1         //Przestaw pojemnik na wysokość „wysoki”
R Q2         //Q2=0, Q1=1

//Detekcja detalu niskiego
AN I1        //Czujnik do detekcji wyższych detali nie działa I1=0 oraz
A I2         //Czujnik do detekcji detali niższych I2=1
AN T1        //Timer T1 musi być w stanie niskim
S Q2         // Przestaw pojemnik na wysokość „niski”
R Q1         //Q2=1, Q1=0

//Detal wysoki
//Detekcja czujnika krańcowego I3 na siłowniku
A I3         //Jeśli pojawi się zbocze narastające na I3
A Q1         //oraz będzie ustawione Q1
=M1          //to ustaw Znacznik M1

//Detal niski
//Detekcja czujnika I4 na siłowniku
A I4         //Jeśli pojawi się zbocze narastające na I4
A Q2         //oraz będzie ustawione Q2
=M2          //to ustaw Znacznik M2

//Detekcja przerwy między detalami
AN I1        //Brak detalu „wysokiego”
AN I2        //Brak detalu niskiego
R M1         //Kasowanie Znaczników pomocniczych M1,M2
R M2

//Wyzwolenie Timera T1 do synchronizacji
O M1         //Jeśli Znacznik M1 lub
O M2         //Znacznik M2 są w stanie ‘1’
L 1s        //to wywołaj Timer1 w trybie Pojedynczy impuls
SE T1       //o czasie trwania 1s

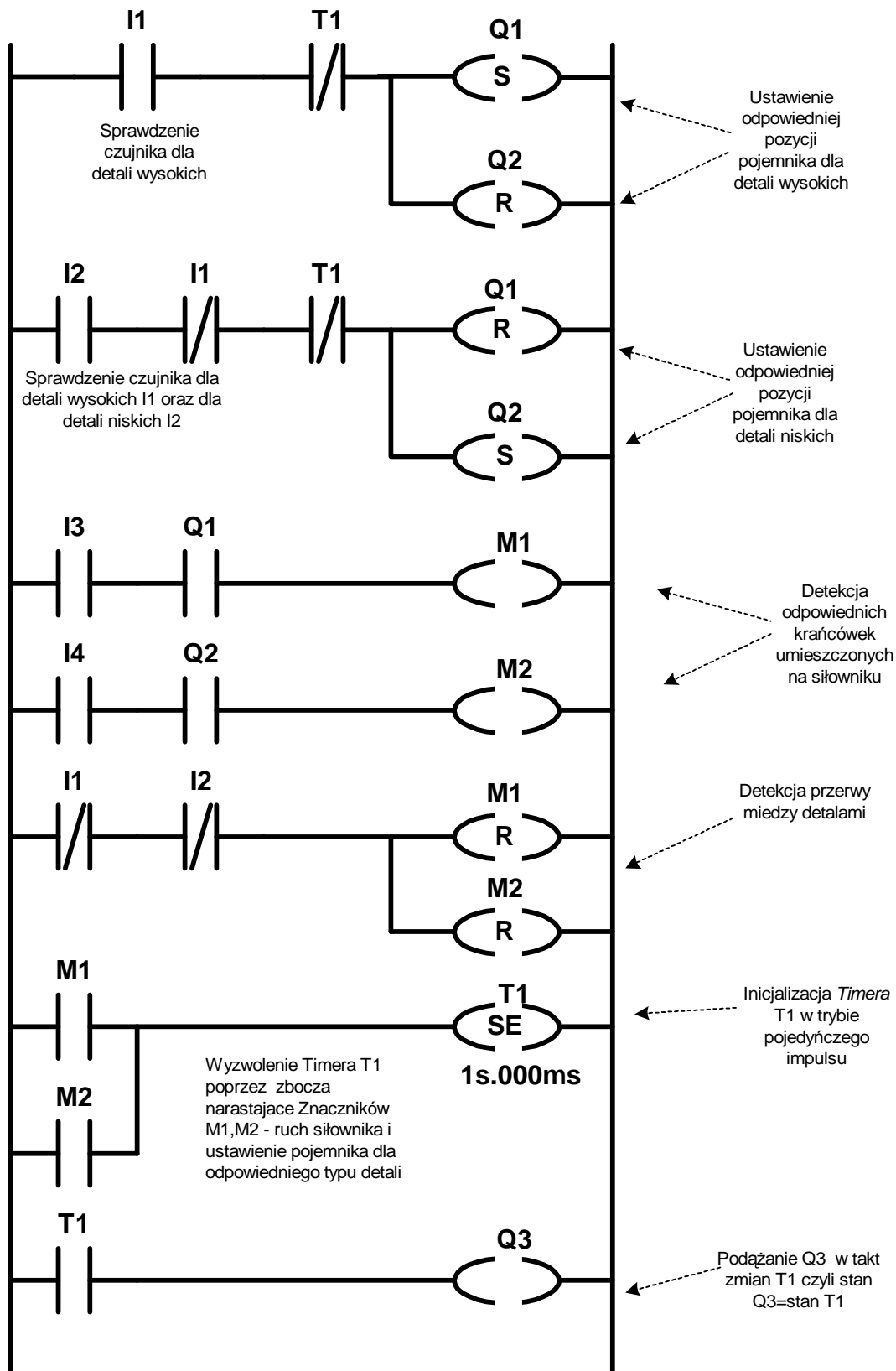
A T1         //Ustaw Q3 tak jak T1
=Q3
```

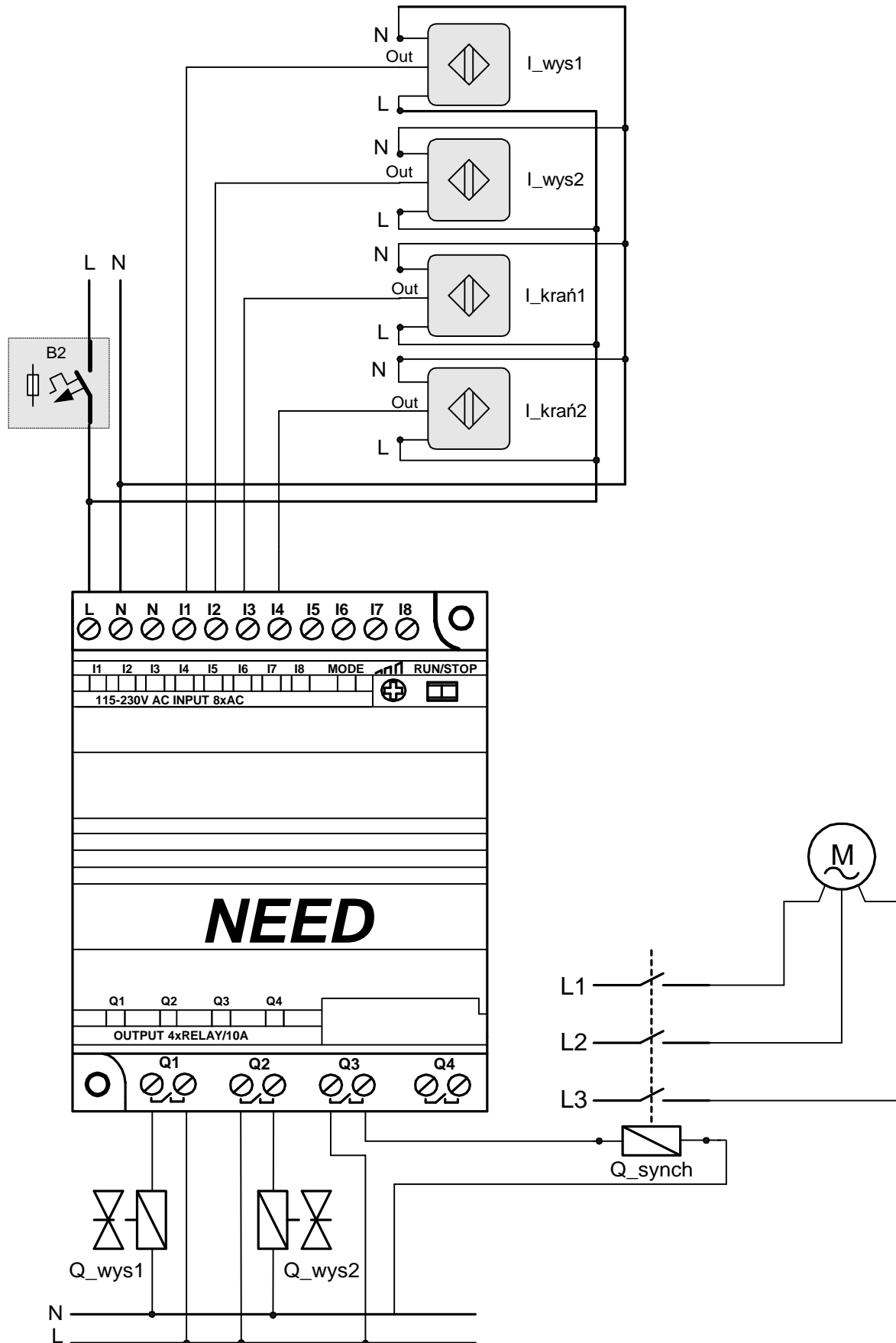
Do generacji impulsu wykorzystaliśmy *Timer* T1 w trybie Pojedynczego Impulsu. Czyli

pojawienie się zbocza narastającego na wejściu I3 lub I4 spowoduje wygenerowanie pojedynczego impulsu synchronizacji na Q3.

Poniżej przedstawiono wersję programu w języku graficznym LAD.

## LAD

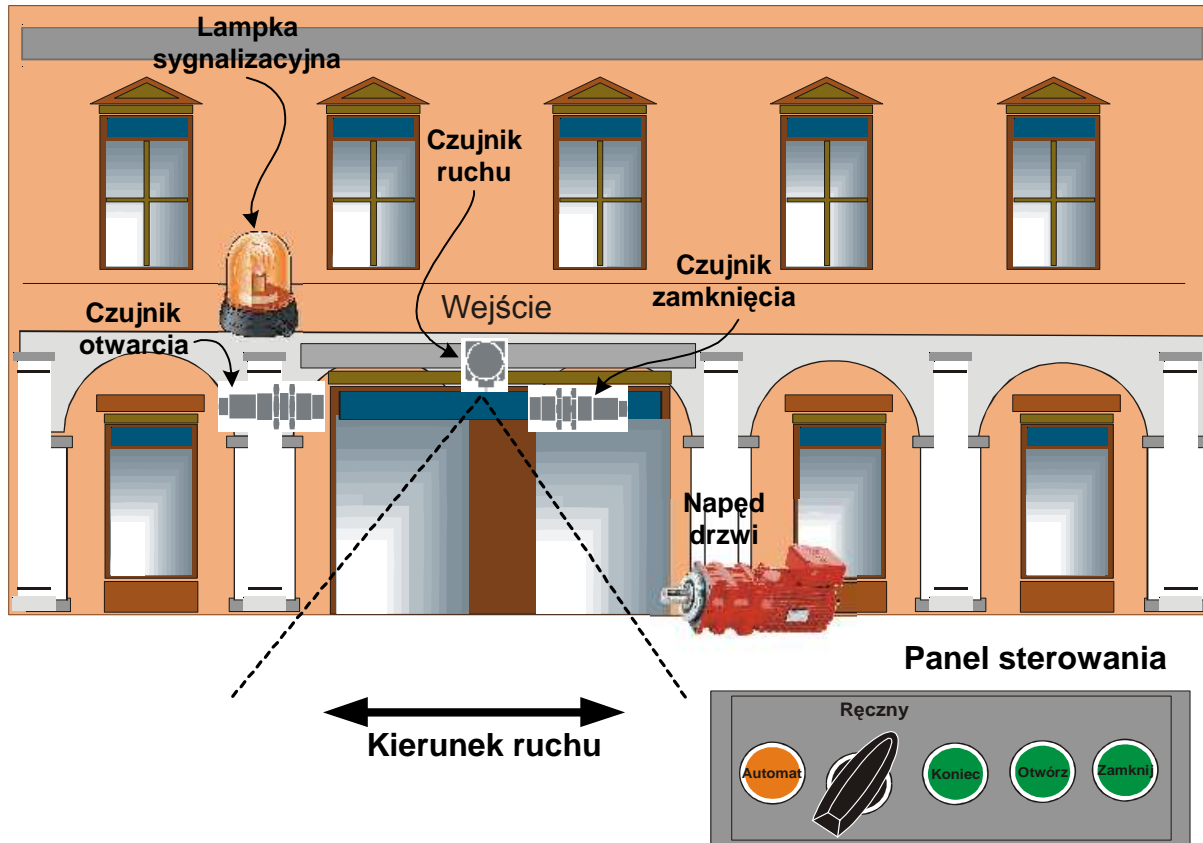




Rys. 10.1.2. Przykładowy schemat połączeń elektrycznych układu do detekcji wysokości detalu.

## 10.2. Drzwi automatyczne

Sterowanie drzwiami automatycznymi jest chyba wszystkim dobrze znane. Sklepy, urzędy, banki itp. bardzo często wykorzystują drzwi automatyczne, ale przy pomocy przekaźnika programowalnego NEED można tradycyjne sterowanie ubogacić o nowe funkcje, znacznie usprawniające nie tylko ruch klientów, ale i funkcjonalność całego budynku.



Rys. 10.2.1. Sterowanie drzwiami automatycznymi.

Opis zadania:

Należy sterować otwieraniem i zamykaniem drzwi automatycznych w budynku.

Dobór sprzętu:

- 1) Należy dobrać odpowiedni czujnik do detekcji ruchu. Zasięg tego czujnika powinien być taki, aby czas reakcji drzwi nie tamował ruchu. Czyli, aby otwarcie drzwi odbywało się odpowiednio wcześniej, przed dojściem klienta. Czujnik zewnętrzny nazwijmy „I\_zew”, a wewnętrzny „I\_wew”. Aby detekować położenie drzwi należy także zainstalować czujniki położenia krańcowych. Nazwijmy je „I\_otwarte” dla otwartych drzwi i „I\_zamknięte” dla zamkniętych. Dla zwiększenia funkcjonalności drzwi dołóżmy jeszcze przełącznik, którym będziemy mogli ustawiać 3 tryb pracy: *Automat* – wszystko funkcjonuje tak, jak w czasie normalnych godzin pracy, *Koniec* – drzwi zostają otwarte tylko dla osób wychodzących z budynku. Dobrą regułą przy takiego typu sterowaniu jest także sterowanie ręczne – wyposażmy więc nasz układ w przyciski „Otwórz” oraz „Zamknij” służące do ręcznego otwierania i zamykania drzwi – w trybie pracy *Ręczny* (nie jest załączone ani *Zamykanie*, ani *Automat*). Wszystkie przyciski zbierzmy w jednym miejscu – w panelu sterowania.
- 2) Drzwi powinny być napędzane silnikiem ze sprzęgłem, które zabezpiecza przed zaciśnięciem. Oznaczmy sygnał sterujący ruchem silnika „Q\_zamknij” (ruch do

- przodu – zamykanie – stycznik załączony, ruch do tyłu – otwieranie – stycznik wyłączony) oraz „Q\_silnik” – wyjście załączające silnik. W cały układ włączmy jeszcze lampkę sygnalizacyjną „Q\_syg”, której miganie będzie oznaczało „Zamykanie” sklepu.
- 3) Przekaznik programowalny NEED: będziemy potrzebować 8 wejść i 3 wyjścia.

#### Algorytm

Na początku ustalamy tryb pracy, który sygnalizowany jest świeceniem lampki (w tym przykładzie lampka miga dla trybu *Zamykanie*). Drzwi mają otwierać się, gdy sygnał na wyjściu detektora ruchu jest wysoki. Aby uniknąć reakcji otwierania na przypadkowe wyzwolenia, układ reaguje dopiero po 200ms tzn., jeśli po 200ms od wyzwolenia nadal czujnik ruchu wskazuje na przemieszczenie, wówczas drzwi się otwierają. Czas zwłoki musi oczywiście być dobrany w taki sposób, aby osoba wchodząca lub wychodząca nie musiała czekać na otwarcie drzwi (także odpowiednie ustawienie oraz czułość detektorów ruchu).

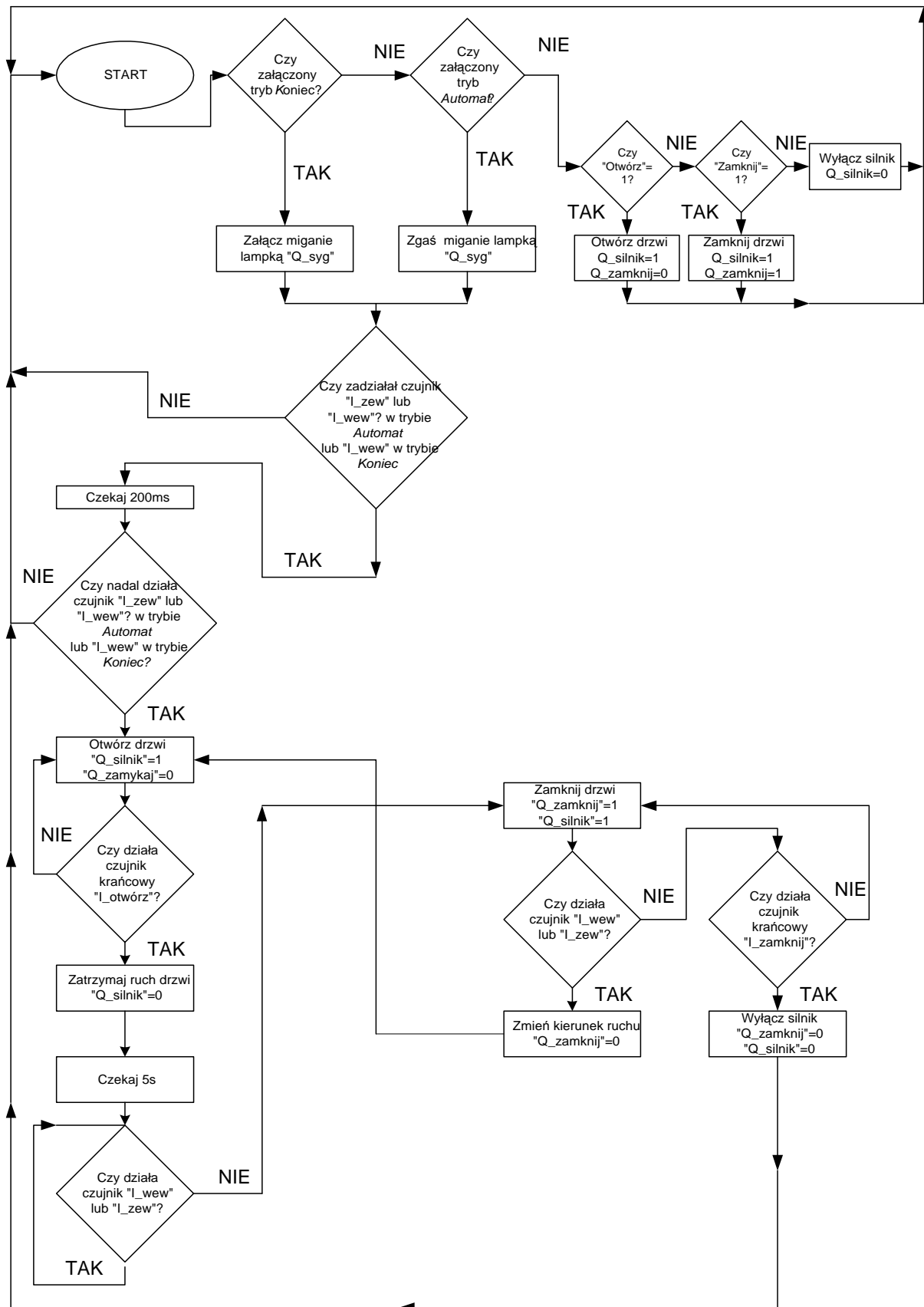
Po otwarciu, drzwi pozostają w takim stanie przez ok. 5s, a następnie zamykają się. Każdy wykryty ruch przez czujnik podczas zamykania powoduje ponowne otwarcie drzwi.

Do zatrzymania drzwi służą czujniki krańcowe.

Jeśli zostanie ustalony tryb pracy *Ręczny*, otwieranie drzwi odbywa się za pomocą przycisku „Otwórz”



Algorytm



Program  
Konfiguracja sprzętowa:

| Adres | Wejścia       | Adres | Wyjścia     |
|-------|---------------|-------|-------------|
| I1    | „I_wew”       | Q1    | „Q_zamknij” |
| I2    | „I_zew”       | Q2    | „Q_silnik”  |
| I3    | „I_otwarte”   | Q3    | „Q_syg”     |
| I4    | „I_zamkniete” |       |             |
| I5    | „Automat”     |       |             |
| I6    | „Koniec”      |       |             |
| I7    | „Zamknij”     |       |             |
| I8    | „Otwórz”      |       |             |

### STL

```

O I1 //Wprowadzamy Znacznik M1, który będzie informował nas
O I2 // o zadziałaniu któregoś z czujników ruchu: „I_wew”, „I_zew”.
=M1

A M1 // Znacznik M2 jest ustawiany, gdy mamy tryb „Automat” oraz
A I5 // nastąpiła detekcja ruchu wewnątrz i na zewnątrz budynku
=M2

A I1 // Znacznik M3 jest ustawiany, gdy mamy tryb „Zamykamy”
A I6 // oraz nastąpiła detekcja ruchu wewnątrz budynku
=M3

AN I5 //Ustalenie trybu pracy “Ręczny”
AN I6
=M13 //Znacznik trybu pracy “Ręczny”

A I6 // I6 jako sygnał wyzwalający T3
L 1s //Ustawienie Timera 3 w trybie migania (długość impulsu 1s)
SL T3

A T3 //Zapalenie lampki sygnalizacyjnej dla trybu pracy „Zamykanie”
=Q3

O M2 // Znacznik M4 jest ustawiany, gdy nastąpiła detekcja ruchu
O M3 // po którejś ze stron, w którymś z trybów
= M4

A M4
L 200ms //Opóźnienie 200ms generowane przez Timer T1 w trybie
SD T1 //Pojedynczy Impuls

A T1 //Ustawienie Znacznika pomocniczego M6 po czasie 200ms od
S M6 //wyzwolenia – od zadziałania jakiegoś czujnika ruchu

A T1 //Sprawdzenie po 200ms czy po którejś ze stron czujnik detekuje ruch
A M4
R M5 //Ustalenie kierunku ruchu drzwi – otwieranie

```

A M6 //Działanie silnika drzwi wejściowych  
AN I3 //do momentu zadziałania czujnika I3  
= M16

A M6 //Działanie silnika drzwi wejściowych  
AN I4 //do momentu zadziałania czujnika I3  
= M15

O M15 //Załączenie lub wyłączenie silnika drzwi  
O M16  
=Q2

A I3 //Wyzwolenie Timera T2 pracującego w trybie Opóźnione Załączenie  
L 5s //czyli ustawienie "sztywnego" czasu otwarcia drzwi  
SD T2  
R M6 //Kasowanie Znacznika M6

A T2 //Ustawienie pomocniczego Znacznika M5  
AN M4  
S M5

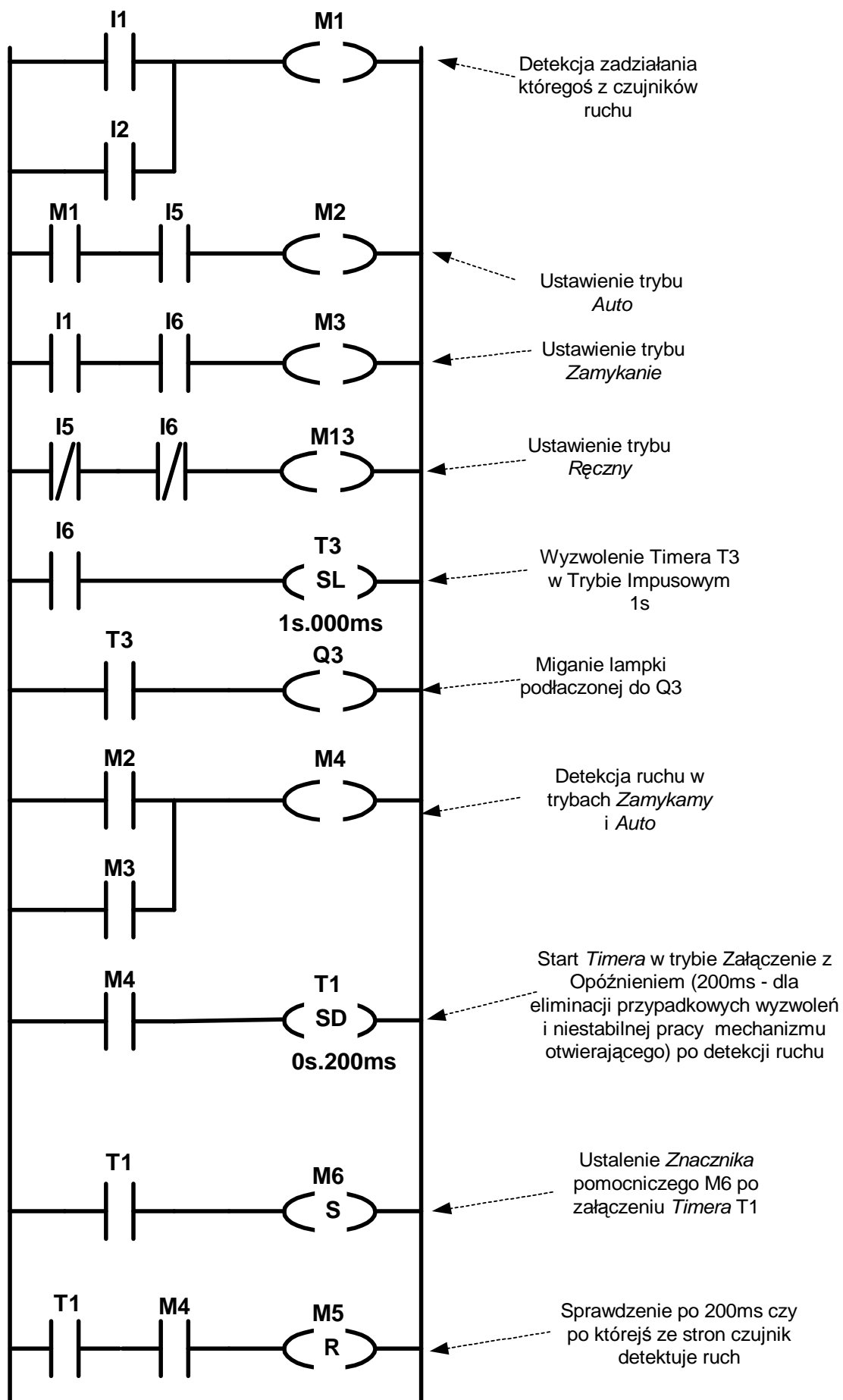
A M5 //Wyłączenie Q1 po osiągnięciu  
AN I4 //pozycji krańcowej sygnalizowanej przez I4  
=Q1

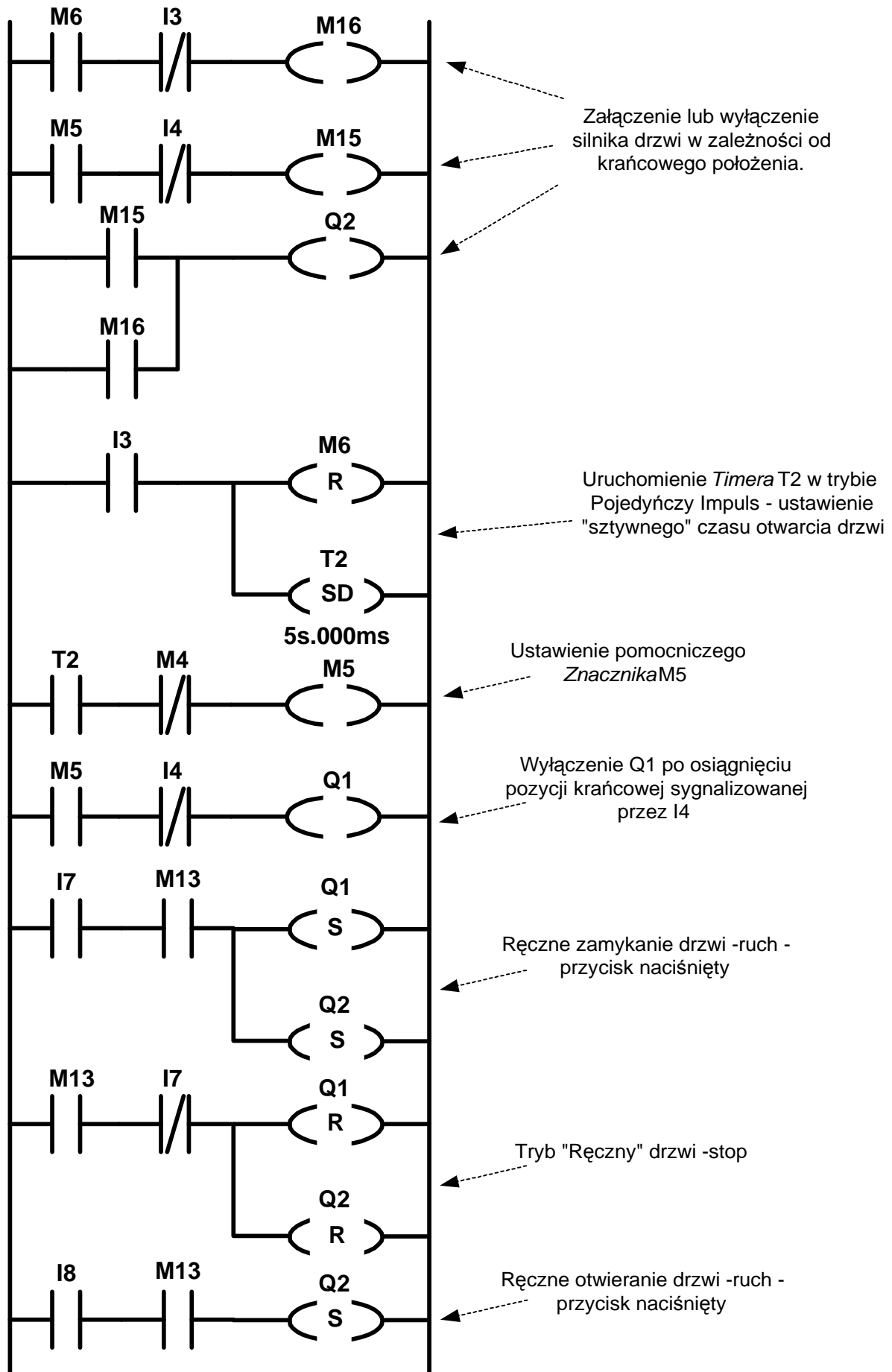
A I7 //Ręczne zamykanie drzwi –ruch – przycisk naciśnięty  
A M13  
S Q1  
S Q2

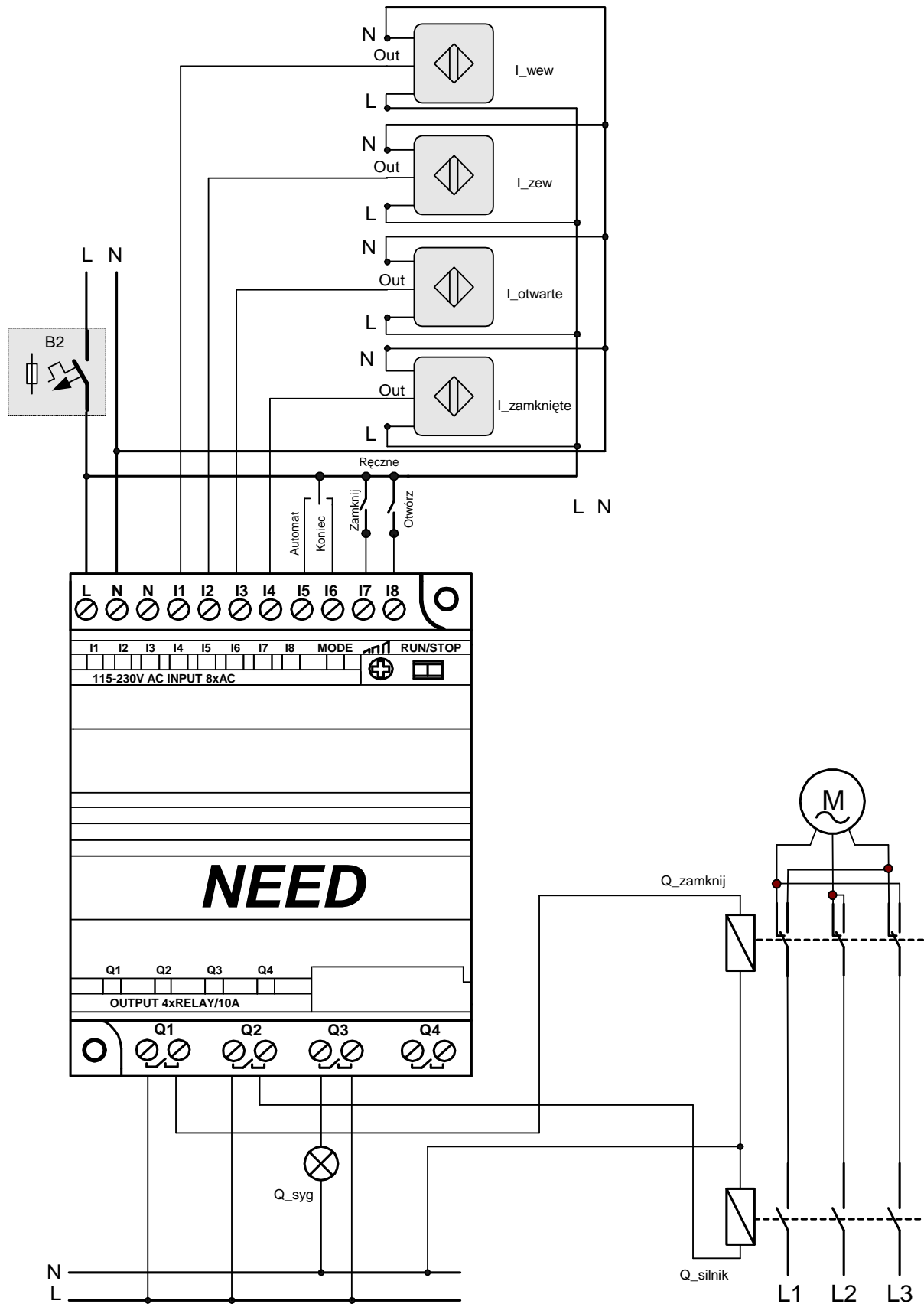
AN I7 //Tryb „Ręczny” drzwi –stop.  
A M13  
R Q1  
R Q2

A M13 //Ręczne otwieranie drzwi –ruch – przycisk naciśnięty  
A I8  
S Q2

LAD







Rys. 10.2.2. Przykład połączeń elektrycznych do sterowania drzwiami automatycznymi.

### 10.3. Dzwonki w szkole

Często zdarza się, iż w szkołach, zakładach pracy instalowane są czasomierze-zegary załączające w określonych godzinach różne urządzenia (dzwonki, alarmy, grzałki itp.) Używając przekaźnika programowalnego NEED można stworzyć własny, prosty, czasowy system sterowania – bardziej przystosowany do lokalnych wymogów i potrzeb.

Opis zadania:

Należy stworzyć system dzwonienia oparty o następujący rozkład zajęć w szkole:

| Lekcja   | Czas trwania  | Dzwonek na przerwę  |                     | Dzwonek na lekcję   |                     |
|----------|---------------|---------------------|---------------------|---------------------|---------------------|
|          |               | Początek załączenia | Początek wyłączenia | Początek załączenia | Początek wyłączenia |
| Lekcja 1 | 8.00 – 8.45   | 8.45                | 8.46                | 8.49                | 8.50                |
| Lekcja 2 | 8.50 – 9.35   | 9.35                | 9.36                | 9.39                | 9.40                |
| Lekcja 3 | 9.40 – 10.25  | 10.25               | 10.26               | 10.34               | 10.35               |
| Lekcja 4 | 10.35 – 11.20 | 11.20               | 11.21               | 11.49               | 11.50               |
| Lekcja 5 | 11.50 – 12.35 | 12.35               | 12.36               | 12.44               | 12.45               |
| Lekcja 6 | 12.45 – 13.30 | 13.30               | 13.31               | 13.39               | 13.40               |
| Lekcja 7 | 13.40 – 14.25 | 14.25               | 14.26               | 14.34               | 14.35               |
| Lekcja 8 | 14.35 – 15.20 | 15.20               | 15.21               | 15.29               | 15.30               |

Dobór sprzętu:

- 1) Należy dobrać odpowiedni panel z przyciskami, które zapewniałyby ręczny tryb sterowania dzwonekami oraz wyłączenie dzwonek w wyznaczonych okresach np. w czasie wakacji, ferii świątecznych, soboty itp.

Nazwijmy:

- Przełącznik – trybu ręcznego – „I\_ręczny” (w tym trybie możliwe tylko „dzwonienie ręczne”),
- Przełącznik – trybu automatyczny – „I\_auto”,
- Przycisk – załączający dzwonek w trybie ręcznym – „I\_włącz”,

- 2) Przełącznik programowalny NEED: 3 wejścia, 1 wyjście.

Program

Konfiguracja sprzętowa:

| Adres | Wejścia    | Adres | Wyjścia     |
|-------|------------|-------|-------------|
| I1    | „I_ręczny” | Q1    | „Q_dzwonek” |
| I2    | „I_auto”   |       |             |
| I3    | „I_włącz”  |       |             |

Algorytm

Do załączeń i wyłączeń dzwonek użyjemy *Zegarów*, w następującej konfiguracji:

## Zegar 1

Brak nazwy (SET)\*

Timer Zegar Licznik >= Komparator Remanencja Ustawienia wejść

Ustawienia zegarów

H: 1

| Kanał A      | Kanał B      | Kanał C      | Kanał D      |
|--------------|--------------|--------------|--------------|
| Dzień 1: Pn  | Dzień 1: Pn  | Dzień 1: Pn  | Dzień 1: Pn  |
| Dzień 2: Sob | Dzień 2: Sob | Dzień 2: Sob | Dzień 2: Sob |
| Zak: 8:45    | Zak: 8:49    | Zak: 9:35    | Zak: 9:39    |
| Wyt: 8:46    | Wyt: 8:50    | Wyt: 9:36    | Wyt: 9:40    |

## Zegar 2

Brak nazwy (SET)\*

Timer Zegar Licznik >= Komparator Remanencja Ustawienia wejść

Ustawienia zegarów

H: 2

| Kanał A      | Kanał B      | Kanał C      | Kanał D      |
|--------------|--------------|--------------|--------------|
| Dzień 1: Pn  | Dzień 1: Pn  | Dzień 1: Pn  | Dzień 1: Pn  |
| Dzień 2: Sob | Dzień 2: Sob | Dzień 2: Sob | Dzień 2: Sob |
| Zak: 10:25   | Zak: 10:34   | Zak: 11:20   | Zak: 11:49   |
| Wyt: 10:26   | Wyt: 10:35   | Wyt: 11:21   | Wyt: 11:50   |



## Zegar 3

Brak nazwy (SET)\*

Timer Zegar Licznik >= Komparator Remanencja Ustawienia wejść

Ustawienia zegarów

H: 3

| Kanał   | Dzień 1 | Dzień 2 | Zał. HH | Zał. MM | Wył. HH | Wył. MM |
|---------|---------|---------|---------|---------|---------|---------|
| Kanał A | Pn      | Sob     | 12      | 35      | 12      | 36      |
| Kanał B | Pn      | Sob     | 12      | 44      | 12      | 45      |
| Kanał C | Pn      | Sob     | 13      | 30      | 13      | 31      |
| Kanał D | Pn      | Sob     | 13      | 39      | 13      | 40      |

## Zegar 4

Brak nazwy (SET)\*

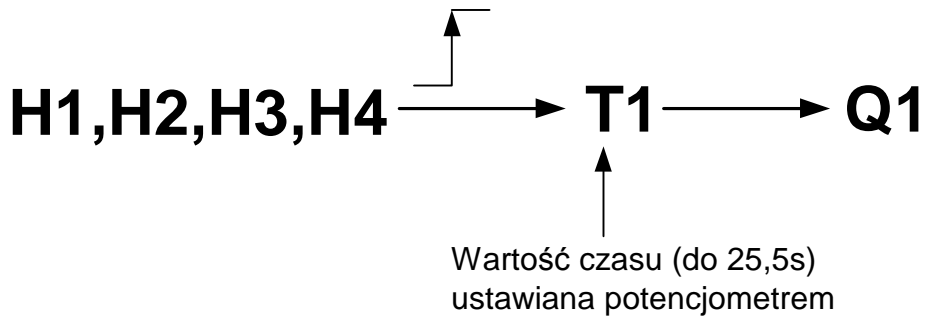
Timer Zegar Licznik >= Komparator Remanencja Ustawienia wejść

Ustawienia zegarów

H: 4

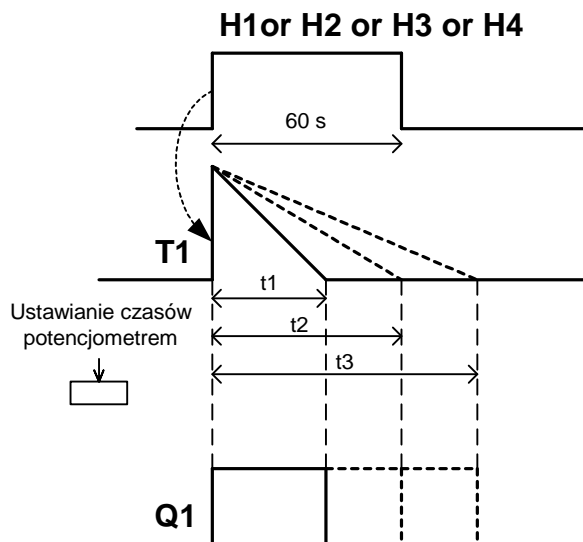
| Kanał   | Dzień 1 | Dzień 2 | Zał. HH | Zał. MM | Wył. HH | Wył. MM |
|---------|---------|---------|---------|---------|---------|---------|
| Kanał A | Pn      | Sob     | 14      | 25      | 14      | 26      |
| Kanał B | Pn      | Sob     | 14      | 34      | 14      | 35      |
| Kanał C | Pn      | Sob     | 15      | 20      | 15      | 21      |
| Kanał D | Pn      | Sob     | 15      | 29      | 15      | 30      |

Zauważmy jedną niedogodność – Zegary możemy ustawiać z dokładnością do 1 minuty. Jak w takim razie poradzić sobie z dzwonekami trwającymi np. tylko 7s (czas dzwonięcia 1 minuty jest zbyt długi). Możemy zastosować *Timer* w trybie Pojedynczy impuls, dzięki któremu będzie możliwa regulacja czasów dzwonięcia – rys. 10.3.2. przedstawia sposób załączania i wyłączania dzwonka.



Rys. 10.3.1. Sposób sterowania dzwonkiem.

Oczywiście nastawy Potencjometru określają tylko w bardzo przybliżony sposób wartości czasów, ale dla ustalenia czasu dzwonienia, tego typu szacowania dają zadawalające rezultaty. Czyli wyjścia Zegarów H1, H2, H3, H4 są załączone przez 1 minutę (najmniejszy możliwy przedział czasowy do ustalenia dla Zegarów). Zegary te wyzwalają Timer 1 (wartość czasu do odmierzenia regulowana Potencjometrem), który z kolei ustawia wyjście Q1. Na rys. 10.3.2. przedstawiony jest sposób „kształtowania” długości czasu dzwonków.



Rys. 10.3.2. Sposób „kształtowania” czasu

O H1 //Załączenie H1 albo,  
 O H2 //załączenie H2 albo,  
 O H3 //Załączenie H3 albo,  
 O H4 //Załączenie H4  
 L Pot x100ms //Powoduje wyzwolenie Timera 1 w trybie Pojedynczy impuls  
 SE T1 //o czasie trwania ustalonym za pomocą Potencjometru

//Tryb AUTO

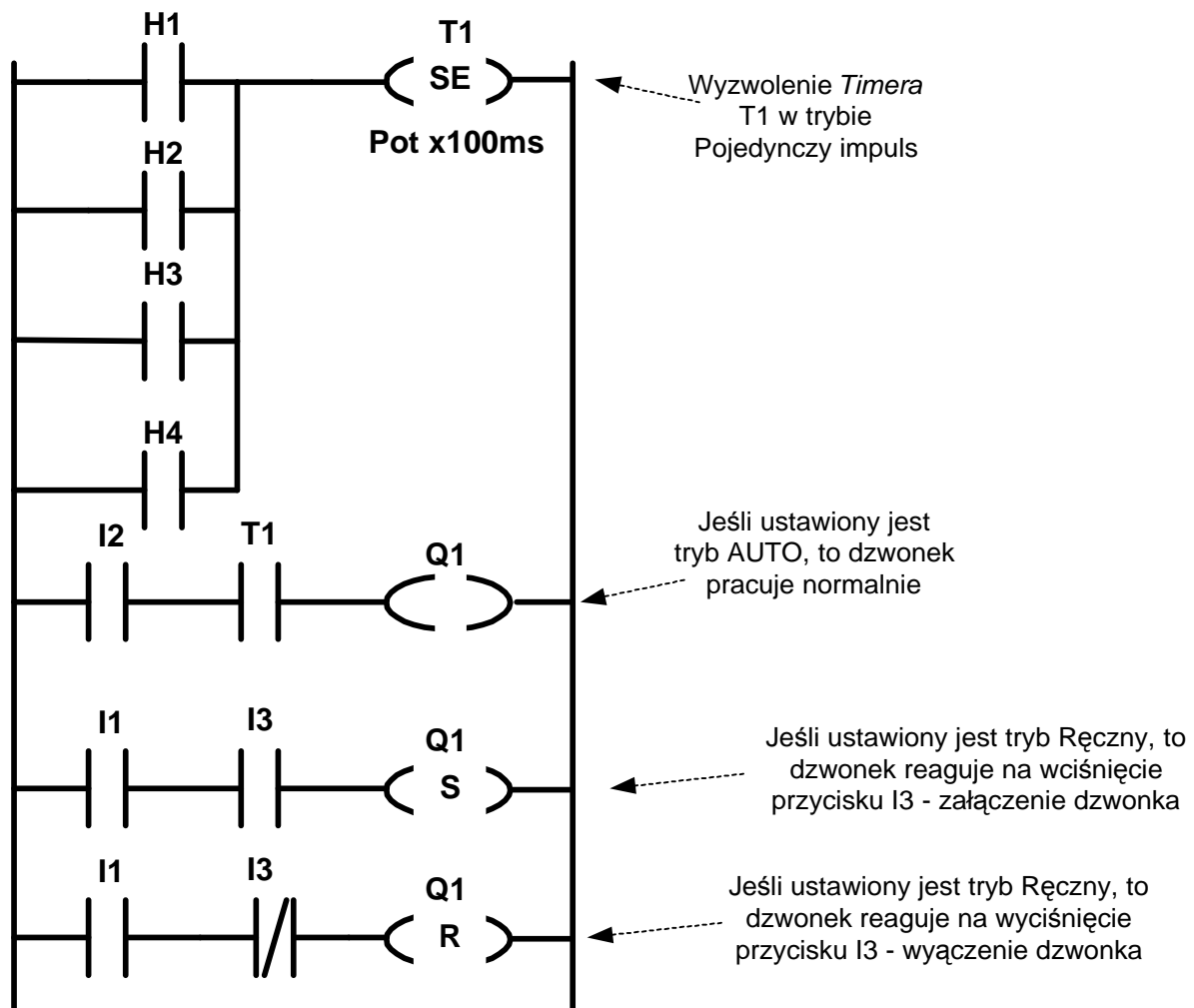
A I2 //Jeśli ustawiony jest tryb AUTO, to dzwonek pracuje normalnie  
 A T1  
 =Q1

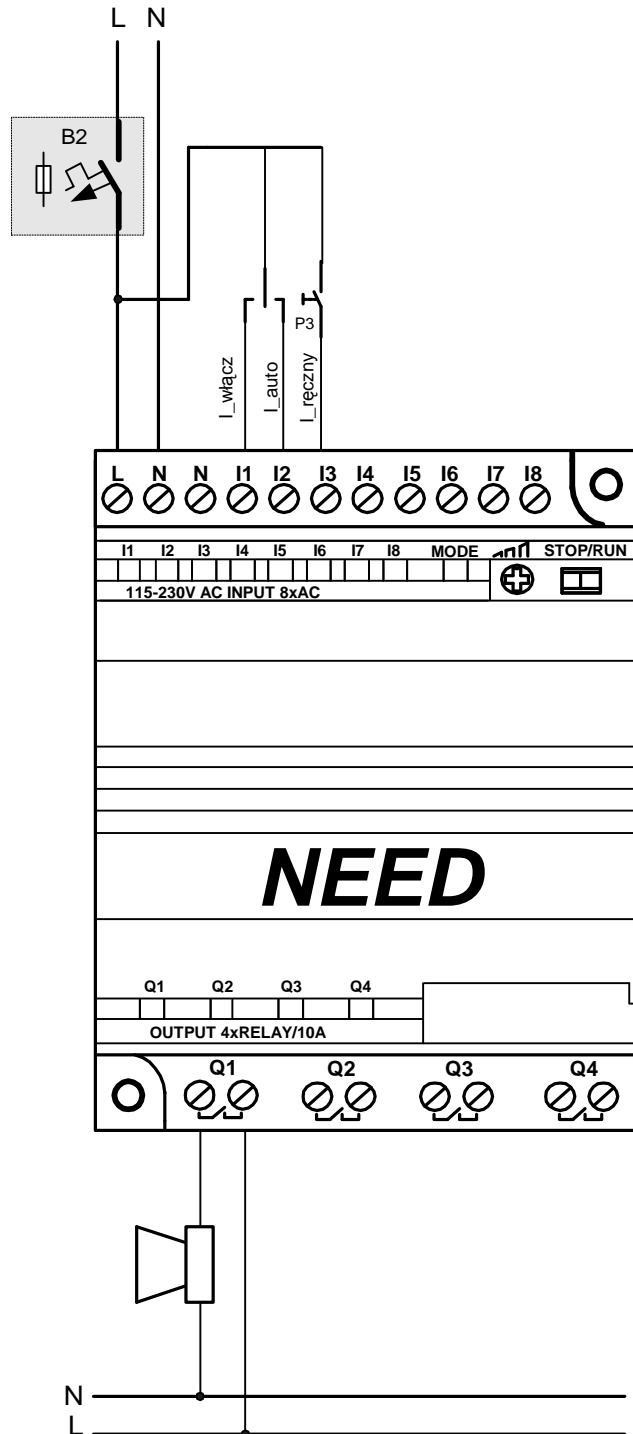
//Tryb Ręczny

A I1 //Jeśli ustawiony jest tryb Ręczny, to dzwonek reaguje na  
 A I3 //wciśnięcie przycisku I3 – załączenie dzwonka  
 S Q1

A I1 //Jeśli ustawiony jest tryb Ręczny, to dzwonek reaguje na  
 AN I3 //wyciśnięcie przycisku I3 – wyłączenie dzwonka  
 R Q1

LAD

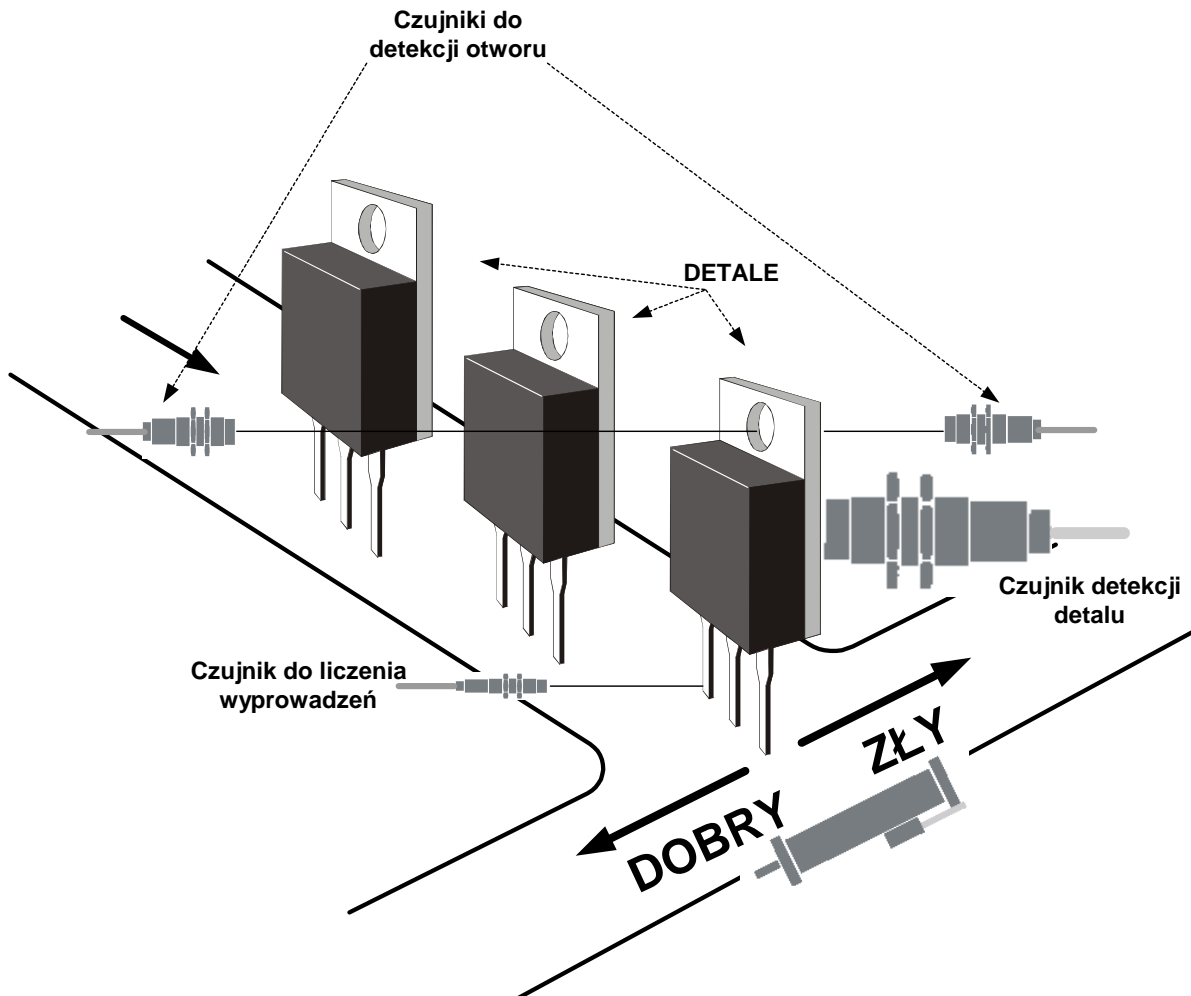




Rys. 10.3.3. Przykład połączeń elektrycznych dla automatycznych dzwonek w szkole.

#### 10.4. Wykrywanie wad detali

W procesie produkcyjnym często zachodzi potrzeba wykrywania wadliwych detali. Za pomocą przekaźnika programowalnego NEED można w prosty i w tani sposób stworzyć system kontrolujący jakość produkowanych detali.



Rys.10.4.1. Wykrywanie wadliwych detali.

Opis zadania:

Należy stworzyć system pozwalający sprawdzać detale (otwór w obudowie tranzystora i liczba wyprowadzeń). Po detekcji złego detalu należy odsegregować go od pozostałych.

Dobór sprzętu:

- 4) Do detekcji otworu potrzebna jest para czujników optycznych (nadajnik i odbiornik), natomiast zliczać wyprowadzenia tranzystora może czujnik laserowy o małej średnicy plamki światła. Dołożymy jeszcze czujnik detekujący transportowany detal – uprości to znacznie nasz późniejszy program.
- 5) Segregator – może to być siłownik sterowany zaworem elektromagnetycznym impulsowym (po podaniu sygnału sterującego na jedną cewkę elektromagnesu zawór pozostaje w zajmowanym położeniu również po zaniku tego sygnału, aż do chwili podania sygnału na drugą cewkę), na którym zamocowane będą np. przegrody mechaniczne do przekierowania ruchu złych detali.
- 6) Przekaźnik programowalny NEED: 3 wejścia, 2 wyjścia.

## Algorytm

W momencie pojawienia się detalu (działa czujnik) na linii transportowej powinna nastąpić aktywacja liczenia wyprowadzeń. W tym samym czasie powinien także być sprawdzony otwór w obudowie tranzystora.

Program:

Konfiguracja sprzętowa:

| Adres | Wejścia   | Adres | Wyjścia                             |
|-------|---|-------|-------------------------------------|
| I1    | Czujnik do detekcji otworu w obudowie tranzystora | Q1    | Pozycja segregatora<br><i>DOBRE</i> |
| I2    | Czujnik liczący wyprowadzenia                     | Q2    | Pozycja segregatora<br><i>ZŁE</i>   |
| I3    | Czujnik detekujący detal                          |       |                                     |

## STL

```

A I3          //Detekcja obecności detalu, zapisanie stanu
= M1         //czujnika I3 do M1
R M3         //Kasowanie Znacznika „Dobry detal” – w ten sposób przegroda
             //zostaje zawsze w ostatniej pozycji i nie jest przestawiana
             //za każdym razem, gdy pojawia się detal

A M1         //Sprawdzenie otworu, przy obecności detalu
A I1
S M2

A I2         //Ustawienie Licznika C1 do zliczania 3 wyprowadzeń
L C#3        //tranzystora
CU C1

AN M1        //Sprawdzenie czy jest otwór i czy zostały zliczone
A M2         //trzy wyprowadzenia tranzystora, gdy detal przestał
A C1         //być “widziany” przez czujnik I3
S M3         //Ustawienie Znacznika „Dobry detal”

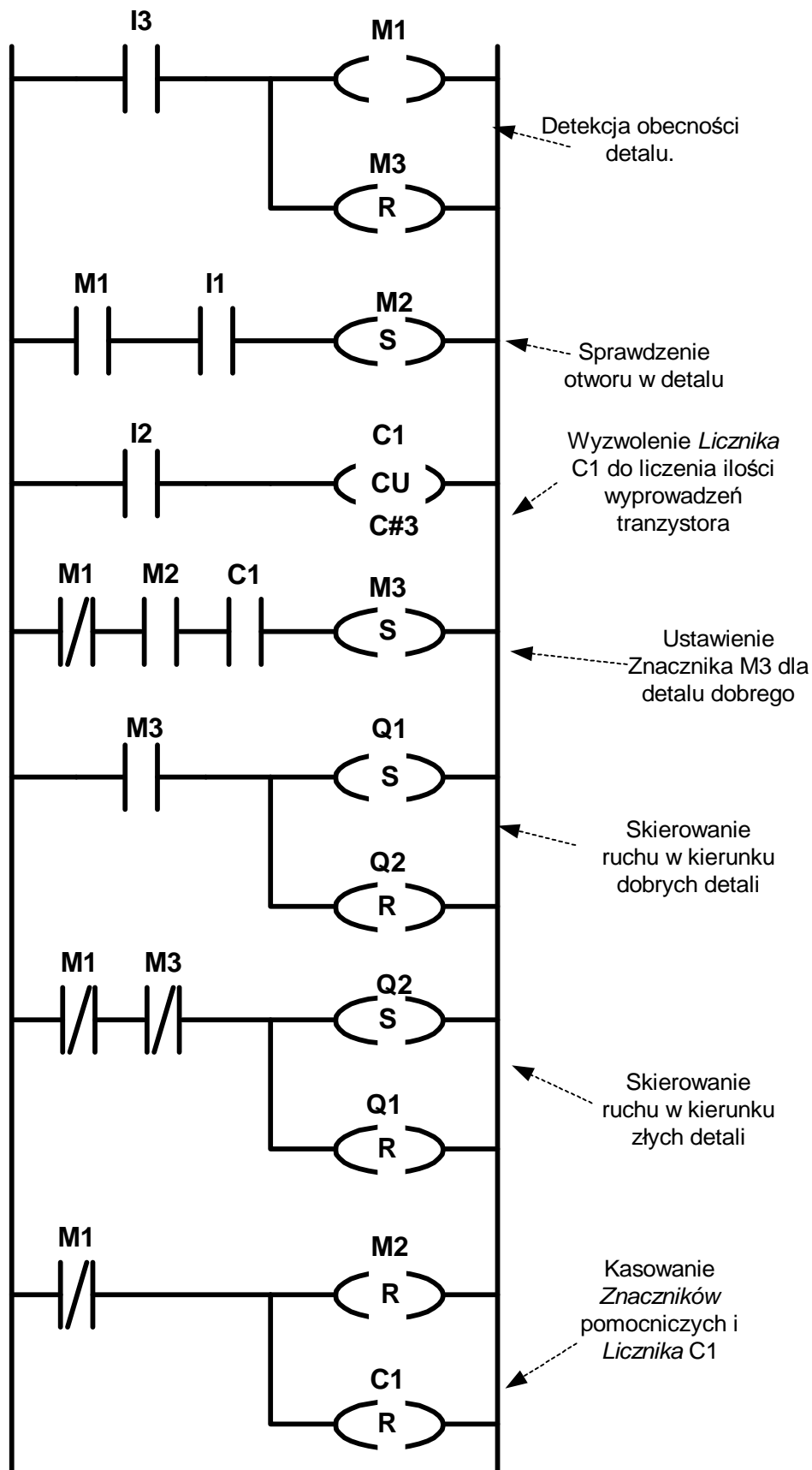
A M3         //Jeśli tranzystor dobry to zostaje przestawiona przegroda
S Q1         //uwalniająca ruch w kierunku detali dobrych
R Q2

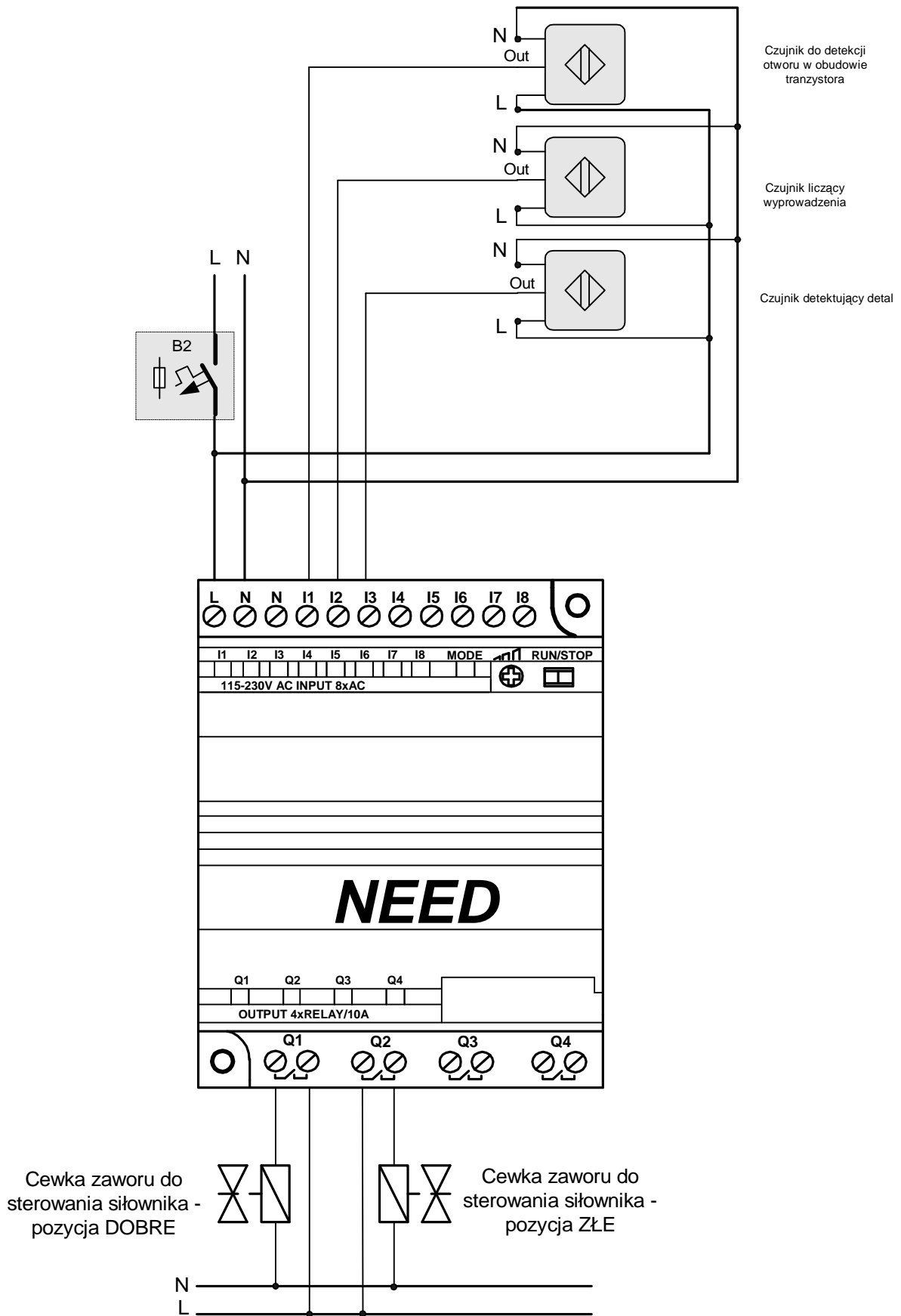
AN M1        //Jeśli tranzystor zły to zostaje przestawiona przegroda
AN M3        // uwalniająca ruch w kierunku detali złych
R Q1
S Q2

AN M1        //Kasowanie Znaczników pomocniczych i Licznika C1
R M2
R C1

```

## LAD





Rys. 10.4.2. Przykład połączeń elektrycznych dla wykrywania wad detali.



### 10.5. Sterowanie ruchem wózków na zakręcie taśmociągu

Opis zadania.

Należy zrealizować przemieszczenie wózków z jednej strony taśmociągu na drugą.

Realizowane to jest przez obracający się talerz napędzany silnikiem M1.

Tylko jeden wózek może się znajdować na obrotowym talerzu.

Nie można wpuszczać następnego wózka, jeśli poprzedni nie opuścił bieżni talerza, lub taśmociąg jest zapełniony (kolejka wózków za zakrętem).

Dodatkowo na zakręcie wózki mogą być zdejmowane, ale muszą być z powrotem odłożone.

Do zrealizowania zadania potrzebne będą elementy sterowania, które poglądowo przedstawiono na rys. 10.5. – czujniki (wejścia) I1 i I2, wyjścia Q1, Q2 i Q3.

Wyjścia czujników podłączamy pod wejścia przełącznika programowalnego następująco:

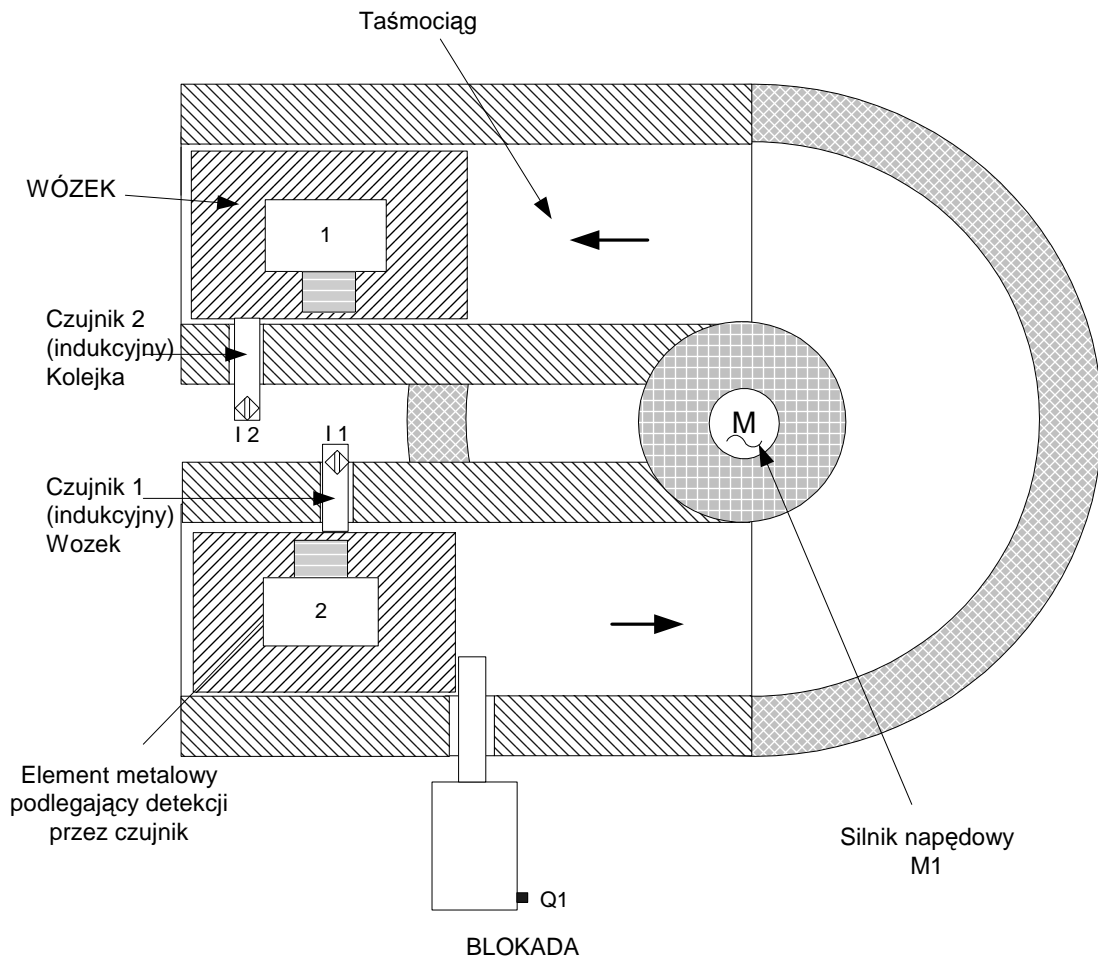
Wej. I1 – czujnik indukcyjny obecności wózka (230 V AC PNP)

Wej. I2 – czujnik indukcyjny kolejki, jednocześnie przejazdu wózka na drugą stronę taśmociągu (230 V AC PNP).

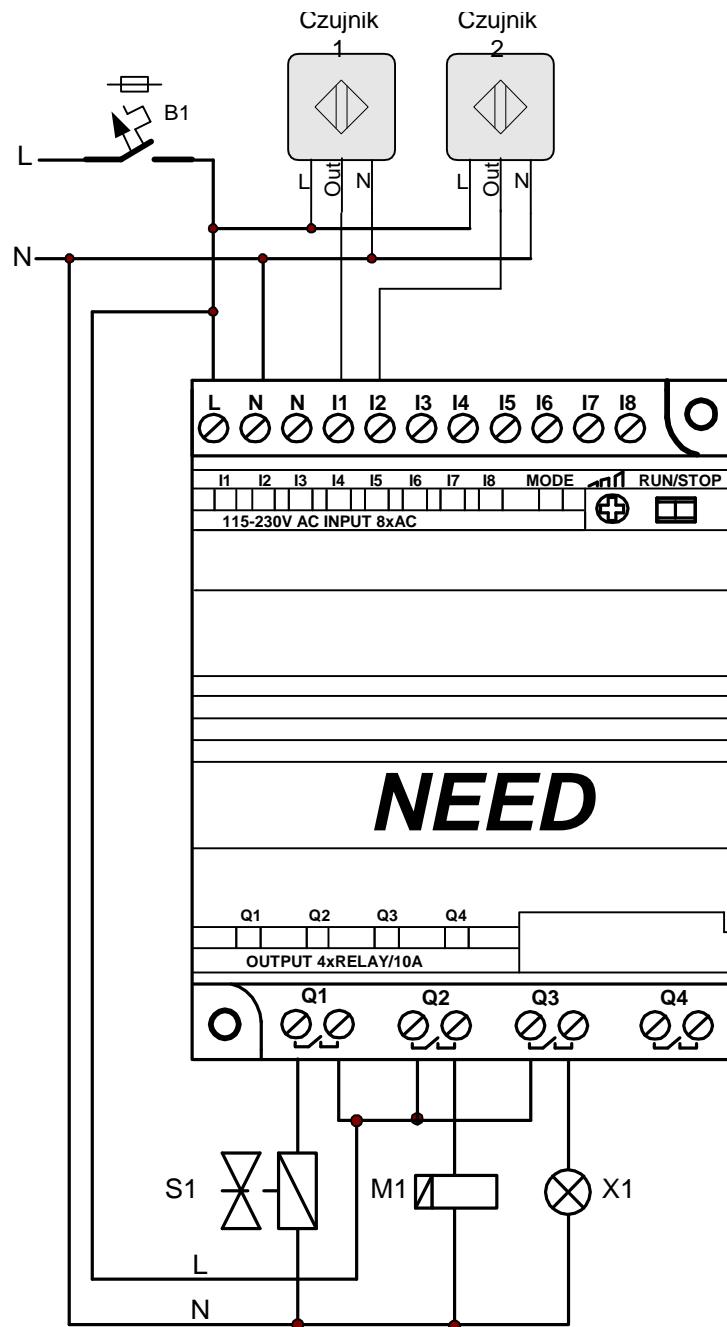
Wyj. Q1 – cewka elektrozaworu sterującego siłownikiem pneumatycznym S1 (230V AC).

Wyj. Q2 – załączanie silnika M1.

Wyj. Q3 – lampka informująca o zgodności liczby wózków wjeżdżających i wyjeżdżających.



Rys. 10.5.1. Sterowanie zakrętem taśmociągu.

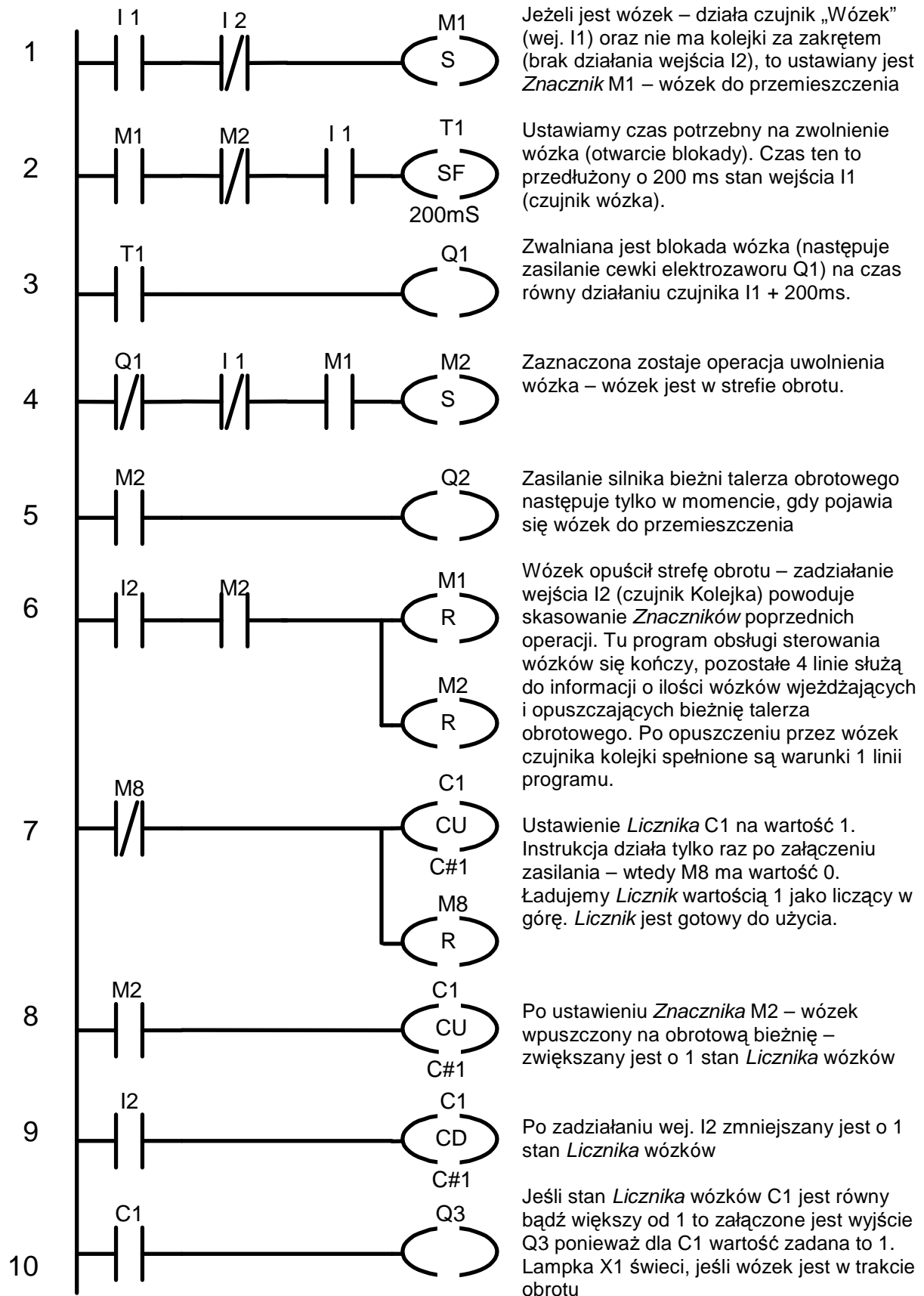


Rys. 10.5.2 Sterowanie zakrętem taśmociągu – schemat elektryczny

Poniżej przedstawiono programy napisane w języku LAD i STL. Numery w pierwszej kolumnie służą do oznaczenia poszczególnych obwodów programu w celu porównania zapisu LAD i STL. Nie są one częścią programu.

W normalnym zapisie STL przerwy między instrukcjami nie muszą występować. Zwiększają one jednak przejrzystość programu. Dodatkowo można wprowadzić komentarze, które pomagają analizować (śledzić) program oraz później pozwalają na łatwą modyfikację.

## Program LAD



## Program STL

|    |   |  |
|----|---|--|
| 1  | A I1<br>AN I2<br>S M1                     | Jeżeli jest wózek – działa czujnik „Wózek” (wej. I1) oraz nie ma kolejki za zakrętem (brak działania wejścia I2), to ustawiany jest <i>Znacznik</i> M1 – wózek do przemieszczenia  |
| 2  | A M1<br>AN M2<br>A I1<br>L 200mS<br>SF T1 | Ustawiamy czas potrzebny na zwolnienie wózka (otwarcie blokady). Czas ten to przedłużony o 200 ms stan wejścia I1 (czujnik wózka). <i>Znacznikiem</i> M1 i M2 zapobiegają ponownemu zadziałaniu <i>Timera</i> w wypadku pojawienia się kolejnego wózka przy czujniku I1(M1) i zadziałaniu przed opuszczeniem wózka strefy obrotu (M2).   |
| 3  | A T1<br>= Q1                              | Zwalniana jest blokada wózka (następuje zasilanie cewki elektrozaworu Q1) na czas równy działaniu czujnika I1 + 200ms.   |
| 4  | AN Q1<br>AN I1<br>A M1<br>S M2            | Zaznaczona zostaje operacja uwolnienia wózka – wózek jest w strefie obrotu.  |
| 5  | A M2<br>= Q2                              | Zasilanie silnika bieżni talerza obrotowego następuje tylko w momencie, gdy pojawia się wózek do przemieszczenia   |
| 6  | A I2<br>A M2<br>R M1<br>R M2              | Wózek opuścił strefę obrotu – zadziałanie wejścia I2 (czujnik Kolejka) powoduje skasowanie <i>Znaczników</i> poprzednich operacji. Tu program obsługi sterowania przemieszczaniem wózków się kończy, pozostałe 4 linie służą do informacji o ilości wózków wjeżdżających i opuszczających bieżnię talerza obrotowego. Po opuszczeniu przez wózek czujnika kolejki spełnione są warunki 1 linii programu. |
| 7  | AN M8<br>L C#1<br>CU C2<br>S M8           | Ustawienie <i>Licznika</i> C2 na wartość 1. Instrukcja działa tylko raz po załączeniu zasilania – wtedy M8 ma wartość 0. Ładujemy <i>Licznik</i> wartością 1 jako liczący w górę. <i>Licznik</i> jest gotowy do użycia. Ustawiamy M8 na '1' co zapewnia, że do momentu wyłączenia zasilania ten obwód (6) nie będzie miał wpływu na działanie programu.  |
| 8  | A M2<br>L C#1<br>CU C2                    | Po ustawieniu <i>Znacznika</i> M2 – wózek wpuszczony na obrotową bieżnię – zwiększany jest o 1 stan <i>Licznika</i> wózków   |
| 9  | A I2<br>L C#1<br>CD C2                    | Po zadziałaniu (wej. I2) zmniejszany jest o 1 stan <i>Licznika</i> wózków  |
| 10 | A C2<br>= Q3                              | Jeśli stan <i>Licznika</i> wózków C2 jest równy bądź większy od 1 to załączone jest wyjście Q3, ponieważ dla C1 wartość zadana to 1. Lampka X1 świeci, jeśli wózek jest w trakcie obrotu   |

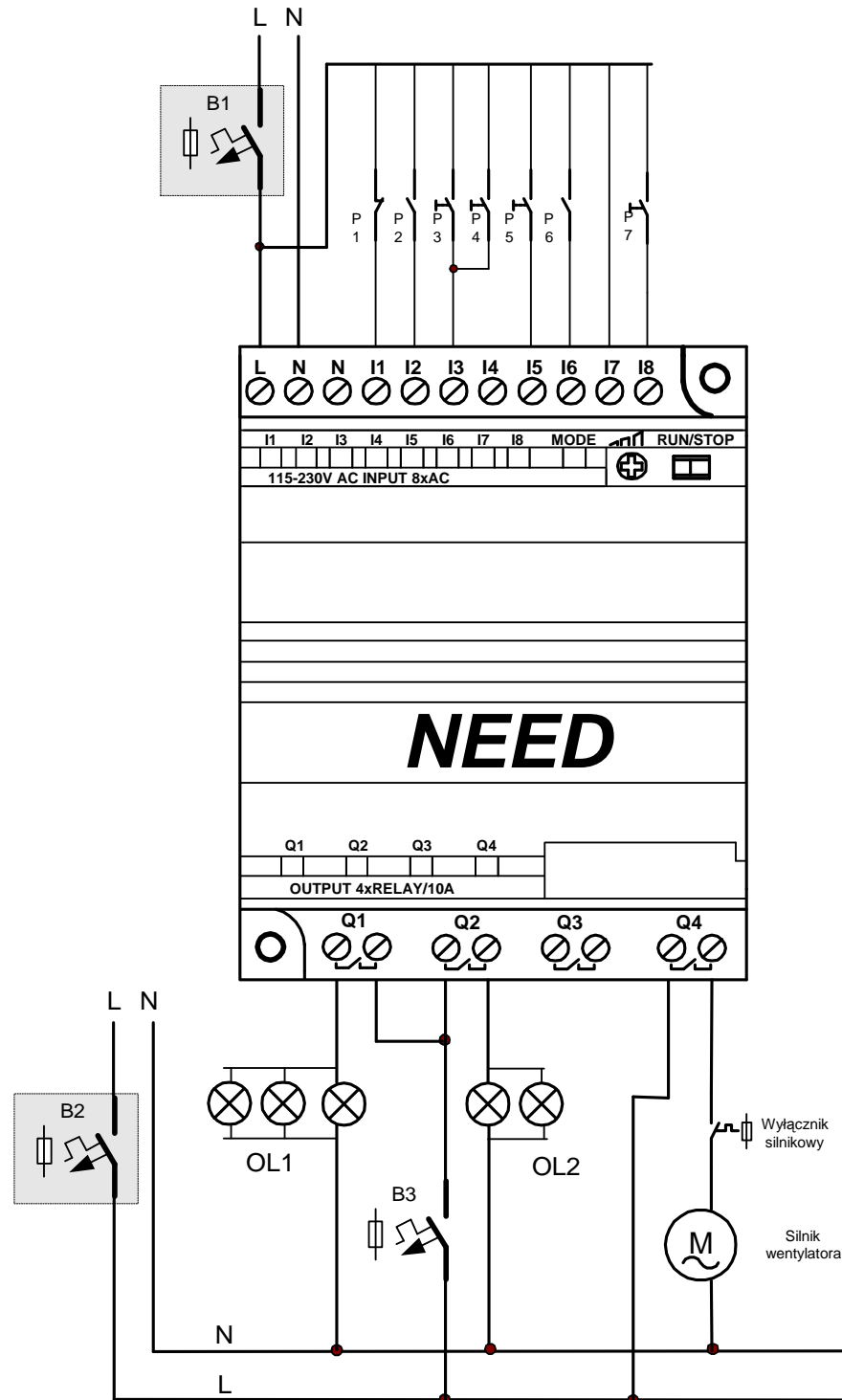
## Komentarz do programu

Sytuacja wyjściowa (przed uruchomieniem programu w przekaźniku programowalnym) jest następująca – siłownik S1 BLOKADA (sterowany elektrozaworem Q1) jest ciągle wysunięty. Po uruchomieniu przekaźnika programowalnego (START) analizowany jest stan wejść i wyjść układu. Następnie linia po linii wykonywane są instrukcje programu – opis w komentarzu obok programu.

## 10.6. Sterownik oświetlenia i wentylacji

### Opis zadania

Zadaniem przedstawionego układu jest kontrola oświetlenia np. biura, hali produkcyjnej, sklepu itp. Często się zdarza, że wychodząc do domu zapominamy o wyłączeniu zbędnego oświetlenia lub o załączeniu oświetlenia tzw. nocnego, które jest niezbędne dla ochrony obiektu. Dodatkowo, dzięki sygnalizacji diod LED umieszczonych na przekaźniku programowalnym, mamy informację o załączonych obwodach i działaniu przycisków. Układ umożliwia centralne wyłączenie i załączenie napięcia, ręcznie lub automatycznie o określonej godzinie lub np. po załączeniu/wyłączeniu zewnętrznego alarmu.



Rys. 10.6.1. Schemat układu połączeń.

## Realizacja.

Do załączania użyjemy łączników chwilowych, tzn. takich, które przewodzą prąd jedynie w chwili naciskania. Dzięki możliwości ich oprogramowania możemy ich użyć w ten sposób, że pierwsze naciśnięcie załączy obwód, drugie go rozłączy, nie blokując możliwości automatycznego wyłączenia/załączenia. Dodatkowy przełącznik (posiadający dla odmiany dwa stany stabilne) umożliwi zmianę rodzaju pracy z automatycznej na ręczną i odwrotnie. W trybie ręcznym układ nie reaguje na centralne wyłączenie.

Dzięki wykorzystaniu przekaźnika programowalnego uzyskujemy komfort obsługi, oszczędność energii oraz możliwość modyfikacji układu bez wykonywania zmian w instalacji. Przedstawione rozwiązanie pokazuje jak bardzo elastycznie można „kształtować” funkcje oświetlenia każdego pomieszczenia.

W układzie jak na rys. 10.6. zastosowano elementy:

P1 – styk wyłącznika awaryjnego.

P2 – przełącznik trybu pracy.

P3, P4 – łączniki dla obwodu lamp L1.

P5 – łącznik dla obwodu lamp L2.

P6 – styk od układu alarmowego (niezależnie działający układ alarmowy).

P7 – załączanie wentylatora – łącznik chwilowy.

wejście I7 – kontrola napięcia zasilającego.

Q1 – sterowanie obwodu OL1.

Q2 – sterowanie obwodu OL2.

Q4 – załączanie silnika wentylatora.

Działanie:

Rozwarcie styku P1 unieruchamia wszystkie obwody wyjściowe.

P2 rozarty – tryb ręczny, P2 zwarty tryb automatyczny.

P3 lub P4 – pierwsze naciśnięcie załącza układ Obwód Lamp 1, kolejne rozłącza.

P5 – działanie jak P3 i P4 tyle, że dla obwodu Obwód Lamp 2.

P6 – styk informujący o uzbrojeniu zewnętrznego alarmu.

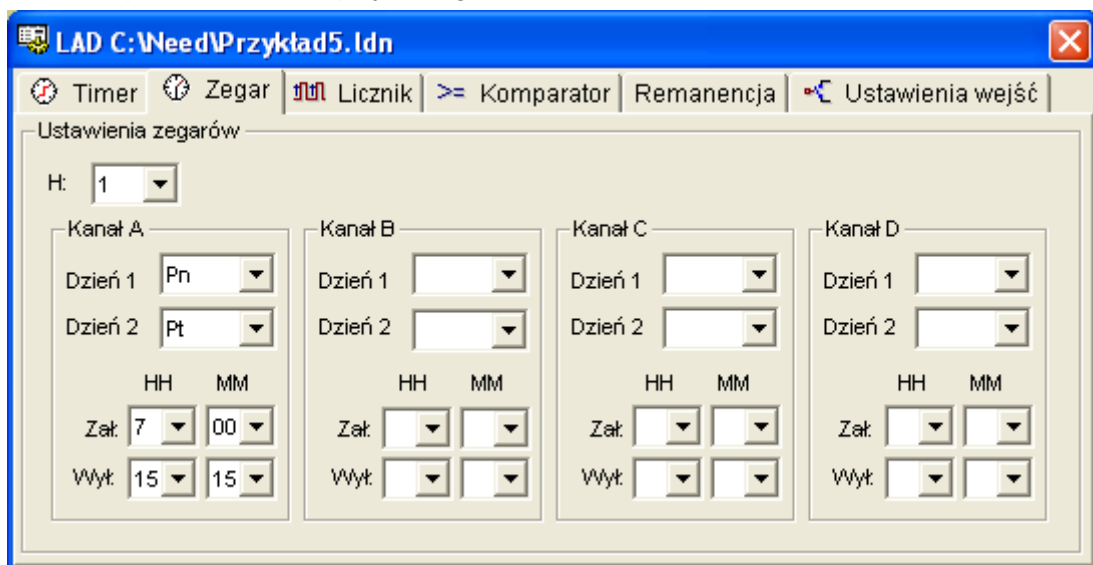
P7 – wł/wył wentylatora.

W trybie automatycznym oświetlenie jest wyłączane/załączane stykiem alarmu lub zgodnie z ustawionym *Zegarem* lub przyciskami P3, P4, P5.

W trybie ręcznym na oświetlenie działają jedynie przyciski P3, P4, P5.

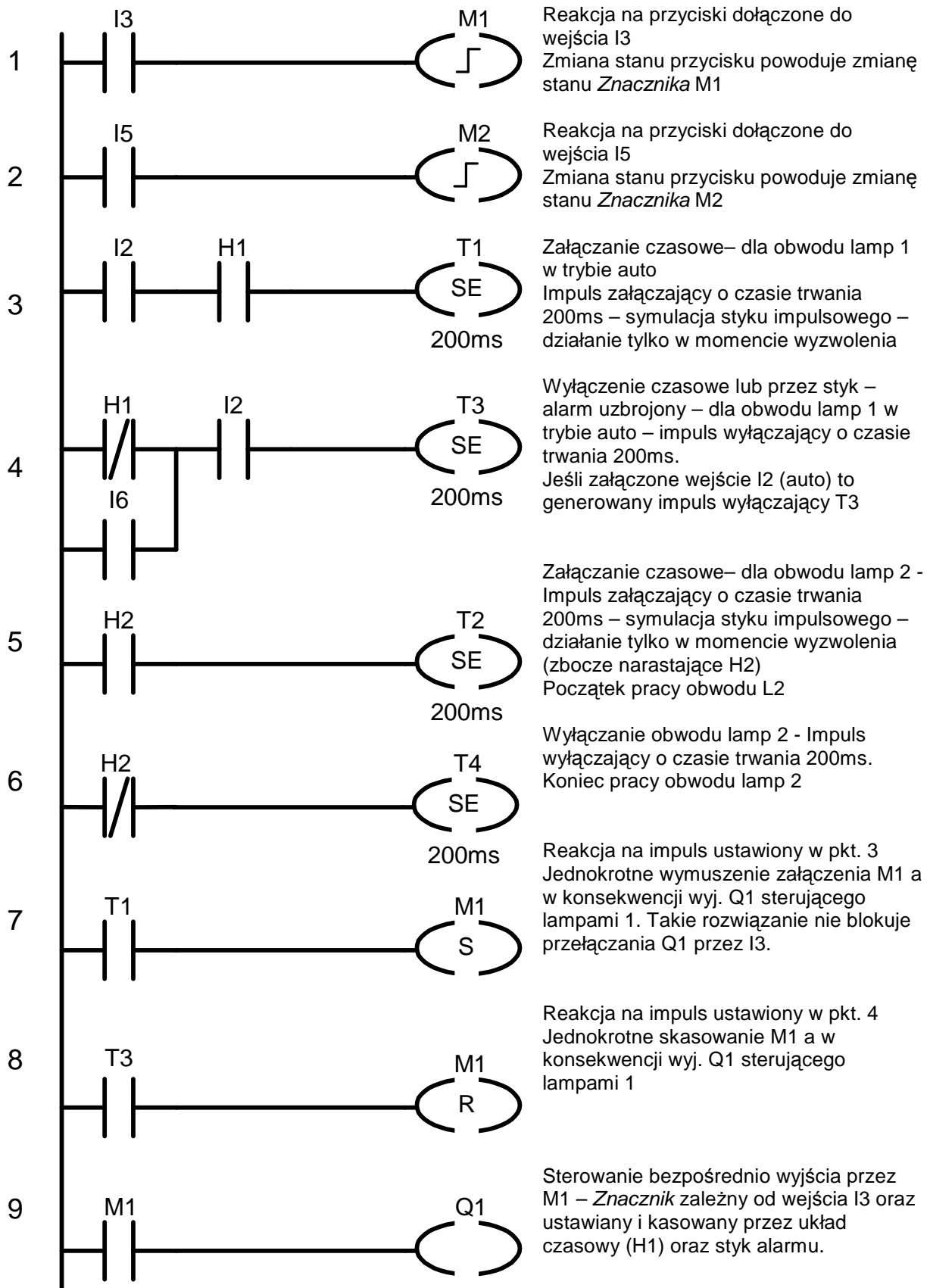
Wentylator wyciągowy działa w ustawionych godzinach pracy.

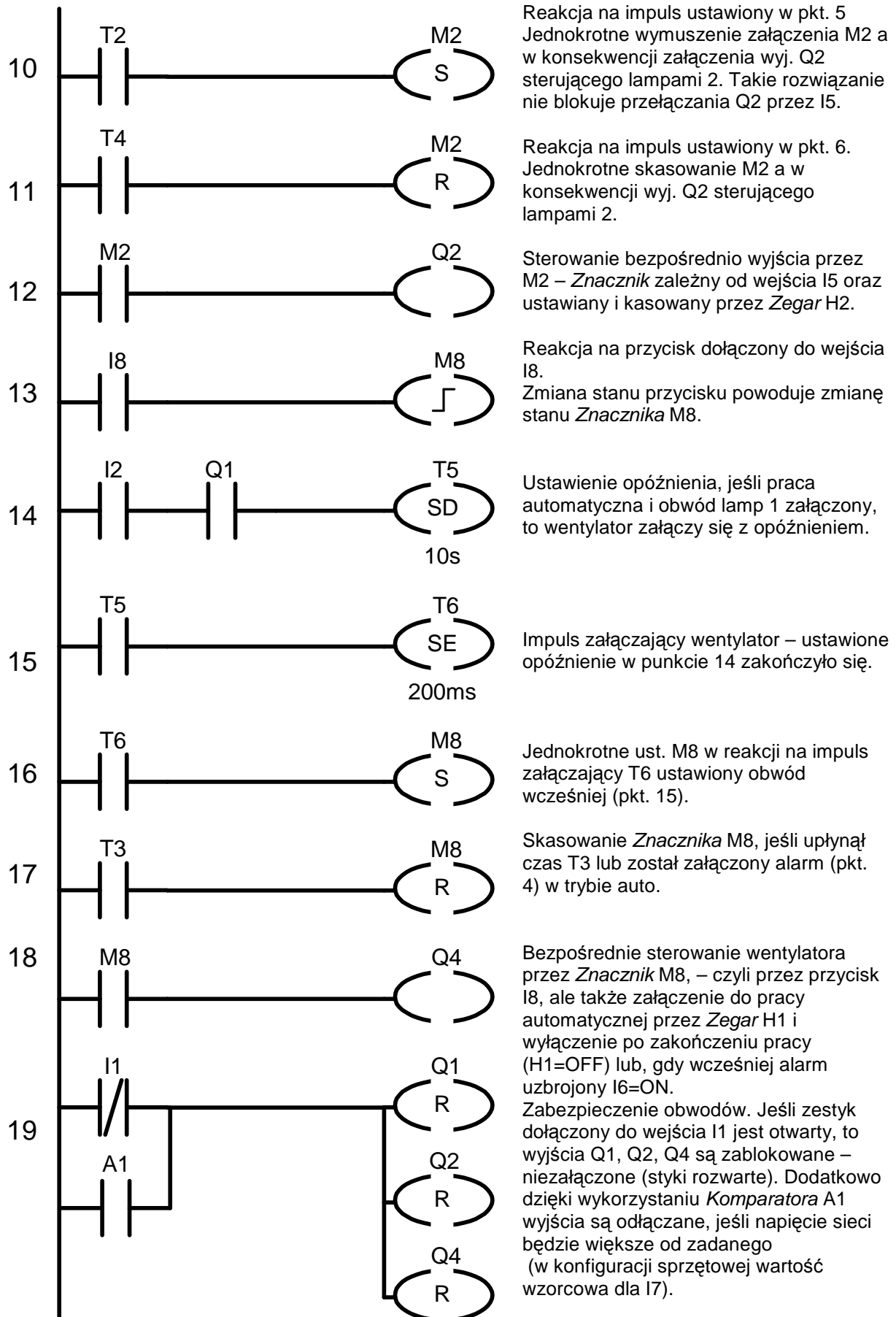
Poniżej przedstawiono przykładową konfigurację *Zegara* ustawionego na załączanie codziennie od poniedziałku do piątku w godz. 7.00 do 15.15.



Rys. 10.6.2. Konfiguracja Zegara

## Program w LAD





Reakcja na impuls ustawiony w pkt. 5  
Jednokrotne wymuszenie załączenia M2 a  
w konsekwencji załączenia wyj. Q2  
sterującego lampami 2. Takie rozwiązanie  
nie blokuje przełączania Q2 przez I5.

Reakcja na impuls ustawiony w pkt. 6.  
Jednokrotne skasowanie M2 a w  
konsekwencji wyj. Q2 sterującego  
lampami 2.

Sterowanie bezpośrednio wyjścia przez  
M2 – *Znacznik* zależny od wejścia I5 oraz  
ustawiany i kasowany przez *Zegar* H2.

Reakcja na przycisk dołączony do wejścia  
I8.  
Zmiana stanu przycisku powoduje zmianę  
stanu *Znacznika* M8.

Ustawienie opóźnienia, jeśli praca  
automatyczna i obwód lamp 1 załączony,  
to wentylator załączy się z opóźnieniem.

Impuls załączający wentylator – ustawione  
opóźnienie w punkcie 14 zakończyło się.

Jednokrotne ust. M8 w reakcji na impuls  
załączający T6 ustawiony obwód  
wcześniej (pkt. 15).

Skasowanie *Znacznika* M8, jeśli upłynął  
czas T3 lub został załączony alarm (pkt.  
4) w trybie auto.

Bezpośrednie sterowanie wentylatora  
przez *Znacznik* M8, – czyli przez przycisk  
I8, ale także załączenie do pracy  
automatycznej przez *Zegar* H1 i  
wyłączenie po zakończeniu pracy  
(H1=OFF) lub, gdy wcześniej alarm  
uzbrojony I6=ON.

Zabezpieczenie obwodów. Jeśli zestyk  
dołączony do wejścia I1 jest otwarty, to  
wyjścia Q1, Q2, Q4 są zablokowane –  
niezałączone (styki rozwarte). Dodatkowo  
dzięki wykorzystaniu *Komparatora* A1  
wyjścia są odłączane, jeśli napięcie sieci  
będzie większe od zadanego  
(w konfiguracji sprzętowej wartość  
wzorcowa dla I7).



## Program w STL

|   |   |  |
|---|---|--|
| 1 | <pre>// Zał/wył obwodu 1 - przyciski A I3 FP M1</pre>   | <p>Reakcja na przyciski dołączone do wejścia I3.<br/>Zmiana stanu przycisku powoduje zmianę stanu<br/><i>Znacznika</i> M1.</p>   |
| 2 | <pre>// Zał/wył obwodu 2 - przycisk A I5 FP M2</pre>  | <p>Reakcja na przyciski dołączone do wejścia I5.<br/>Zmiana stanu przycisku powoduje zmianę stanu<br/><i>Znacznika</i> M2.</p>   |
| 3 | <pre>// Załączanie czasowe automat. // Obwód L1 - Impuls załączający A I2 A H1 L 200ms SE T1</pre>                    | <p>Załączanie czasowe– dla obwodu lamp 1 w trybie auto.<br/>Impuls załączający o czasie trwania 200ms – symulacja styku impulsowego – działanie tylko w momencie wyzwolenia.</p>   |
| 4 | <pre>// Wyłączanie czasowe lub przez alarm // Obwód L1 w trybie auto AN H1 O I6 = M10  A I2 A M10 L 200ms SE T3</pre> | <p>Wyłączenie czasowe lub przez styk – alarm uzbrojony – dla obwodu lamp 1 w trybie auto – impuls wyłączający o czasie trwania 200ms.<br/>M10 – suma logiczna czasu wyłączenia (H1) i I6 – wcześniej ktoś uzbroił alarm.</p> <p>Jeśli załączone wejście I2 (auto) to generowany impuls wyłączający T3.</p> |
| 5 | <pre>// Załączanie czasowe // Obwód L2 - Impuls załączający A H2 L 200ms SE T2</pre>                                  | <p>Załączanie czasowe– dla obwodu lamp 2 – Impuls załączający o czasie trwania 200ms – symulacja styku impulsowego – działanie tylko w momencie wyzwolenia → H2 =1.<br/>Początek pracy obwodu L2.</p>  |
| 6 | <pre>// Wyłączanie czasowe // Obwód L2 AN H2 L 200ms SE T4</pre>  | <p>Wyłączanie obwodu lamp 2 – Impuls wyłączający o czasie trwania 200ms.<br/>Koniec pracy obwodu lamp 2.</p>   |
| 7 | <pre>// Załączanie obwodu 1 A T1 S M1</pre>   | <p>Reakcja na impuls ustawiony w pkt. 3.<br/>Jednokrotne wymuszenie załączenia M1a w konsekwencji Wyj. Q1 sterującego lampami 1.<br/>Takie rozwiązanie nie blokuje przełączania Q1 przez I3.</p>   |
| 8 | <pre>// Wyłączanie obwodu 1 A T3 R M1</pre>   | <p>Reakcja na impuls ustawiony w pkt. 4.<br/>Jednokrotne skasowanie M1 a w konsekwencji wyj. Q1 sterującego lampami 1.</p>   |
| 9 | <pre>// Wyjście Q1 O M1 = Q1</pre>  | <p>Sterowanie bezpośrednio wyjścia przez M1 – <i>Znacznik</i> zależny od wejścia I3 oraz ustawiany i kasowany przez <i>Zegar</i> H1 oraz styk alarmu.</p>  |

|    |  |  |
|----|--|--|
| 10 | // Załączanie obwodu 2<br>A T2<br>S M2   | Reakcja na impuls ustawiony w pkt. 5.<br>Jednokrotne wymuszenie załączenia M2 a w konsekwencji wyj. Q2 sterującego lampami 2.<br>Takie rozwiązanie nie blokuje przełączania Q2 przez I5.   |
| 11 | // Wyłączanie obwodu 2<br>A T4<br>R M2   | Reakcja na impuls ustawiony w pkt. 6.<br>Jednokrotne skasowanie M2 a w konsekwencji wyj. Q2 sterującego lampami 2.   |
| 12 | // Wyjście Q2<br>A M2<br>= Q2  | Sterowanie bezpośrednio wyjścia przez M2 –<br><i>Znacznik</i> zależny od wejścia I5 oraz ustawiany i kasowany przez <i>Zegar</i> H2.   |
| 13 | // Załączenie przycisku P8<br>// - Wentylator<br>A I8<br>FP M8                           | Reakcja na przycisk dołączony do wejścia I8.<br>Zmiana stanu przycisku powoduje zmianę stanu <i>Znacznika</i> M8.  |
| 14 | // Załączenie - Wentylator<br>// Ustawienie opóźnienia<br>A I2<br>A Q1<br>L 10s<br>SD T5 | Ustawienie opóźnienia, jeśli praca automatyczna i obwód lamp 1 załączony, to wentylator załączy się z opóźnieniem.   |
| 15 | // Impuls załączający<br>A T5<br>I 200ms<br>SE T6  | Impuls załączający wentylator – ustawione opóźnienie zakończyło się.   |
| 16 | // Zał. wentylatora automatyczne<br>A T6<br>S M8   | Jednokrotne ust. M8 w reakcji na impuls załączający T6 ustawiony powyżej (pkt. 15).  |
| 17 | // Wyłączenie wentylatora<br>// Automatyczne czasowe lub przez alarm<br>A T3<br>R M8     | Jednokrotne skasowanie <i>Znacznika</i> M8, jeśli upłynął czas H1 lub został załączony alarm (pkt. 4) w trybie auto.   |
| 18 | // Wyjście Q4<br>A M8<br>= Q4  | Bezpośrednie sterowanie wentylatora przez <i>Znacznik</i> M8 – czyli przez przycisk I8, ale także załączenie do pracy automatyczne przez <i>Zegar</i> H1 i wyłączenie po zakończeniu pracy (H1=OFF) lub gdy wcześniej alarm uzbrojony I6=ON.   |
| 19 | // Zabezpieczenie/wyłączenie<br>AN I1<br>O A1<br>R Q1<br>R Q2<br>R Q4                    | Zabezpieczenie obwodów. Jeśli zestyk dołączony do wejścia I1 jest otwarty, to wyjścia Q1, Q2, Q4 są zablokowane – niezałączone (styki rozwarte). Dodatkowo dzięki wykorzystaniu <i>Komparatora</i> A1 wyjścia są odłączane, jeśli napięcie sieci będzie większe od zadanego (w konfiguracji sprzętowej wartość wzorcowa dla I7). |

Komentarz do programu.

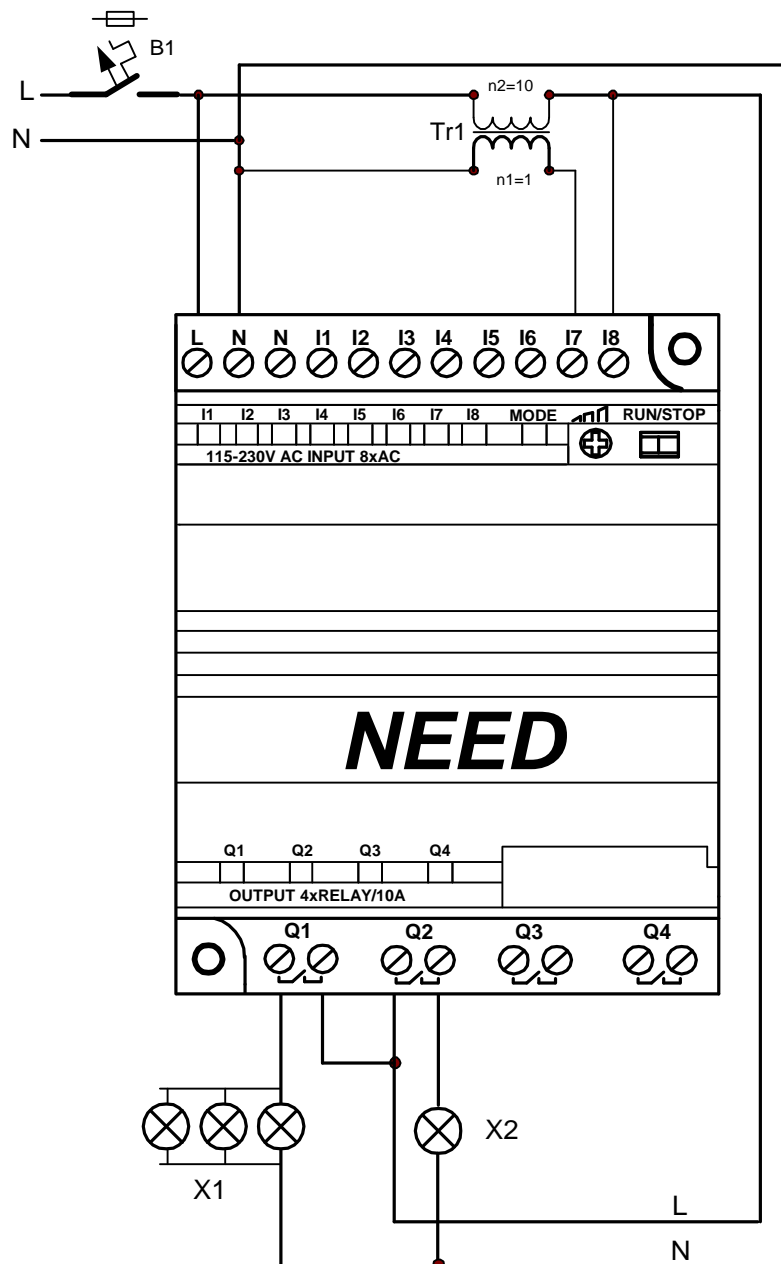
Jest to tylko jeden z możliwych wariantów użycia przekaźnika programowalnego NEED, mający pokazać użycie różnych instrukcji. np. funkcja FP pozwala na naturalne posługiwanie się łącznikami chwilowymi jako załącznikami/wyłącznikami oświetlenia.

Użycie wbudowanego zegara stwarza szereg możliwości czasowego sterowania układami. Wykorzystanie wejścia analogowego pozwala zabezpieczyć obwody sterowane przed skutkami niewłaściwego poziomu napięć zasilających (pod warunkiem zasilania obwodów wykonawczych z tej samej fazy co napięcie zasilające przekaźnik programowalny).

## 10.7. Kontrola obciążenia

Opis zadania.

Wykorzystując możliwości wejścia analogowego można dosyć dokładnie kontrolować pobór mocy pobierany przez obciążenie i odpowiednio zareagować na jego przekroczenie np. odłączając obciążenie od źródła zasilania. W prosty sposób można więc do układu sterowania oświetleniem np. klatki schodowej dołożyć funkcję zabezpieczenia instalacji przed niekontrolowanym poborem prądu (kradzieżą). Dodatkowo tak powstały ogranicznik mocy, nie będzie mógł być przestawiony na wyższą moc bez ingerencji w oprogramowanie. Przedstawiony program może być częścią kompleksowego rozwiązania oświetlenia klatki schodowej lub korytarza, do wolnych wejść można podłączyć łączniki przycisków, czujnika otwarcia drzwi zewnętrznych lub sygnału otwierania zamka elektromagnetycznego przez sieć domofonową. Dodatkowo wykorzystując wbudowany zegar/ kalendarz można uzależnić działanie pewnych funkcji od pory dnia/tygodnia.



Rys. 10.7. Schemat połączeń elektrycznych.

Dla uproszczenia założymy, że zastosowany dodatkowy element przekładnika prądowego, który będzie przetwarzał prąd płynący w obwodzie głównym na proporcjonalne napięcie. Im wyższy pobór prądu tym wyższe napięcie po stronie wtórnej. Pamiętając o ograniczeniach sprzętowych możemy kontrolować moc do 2300W ( $I=10A$ ,  $U_z = 230VAC$ ). Zakładając, że napięciu 100V na uzwojeniu wtórnym przekładnika odpowiada przepływ 10A w obwodzie kontrolowanym, możemy ustawić wartość prądu np. 5A poprzez wpisanie jako wartości wzorcowej 50. Zamiast wartości wzorcowej możemy użyć wbudowanego Potencjometru i wtedy ręcznie ustawić żadaną wartość. Korzystając z funkcji *Komparatora* analogowego A2 ( $I7 \geq$  wartość wzorcowa) możemy sterować wyjściem Q1 w zależności od pobieranego prądu. Po przekroczeniu żadanego prądu ( $I7 \geq 50V$ ) wyjście odłącza obciążenie, co jest sygnalizowane wyjściem Q2. Ponowne załączenie jest możliwe po 10 sekundach od wyłączenia (oczywiście można ustawić dowolny czas).

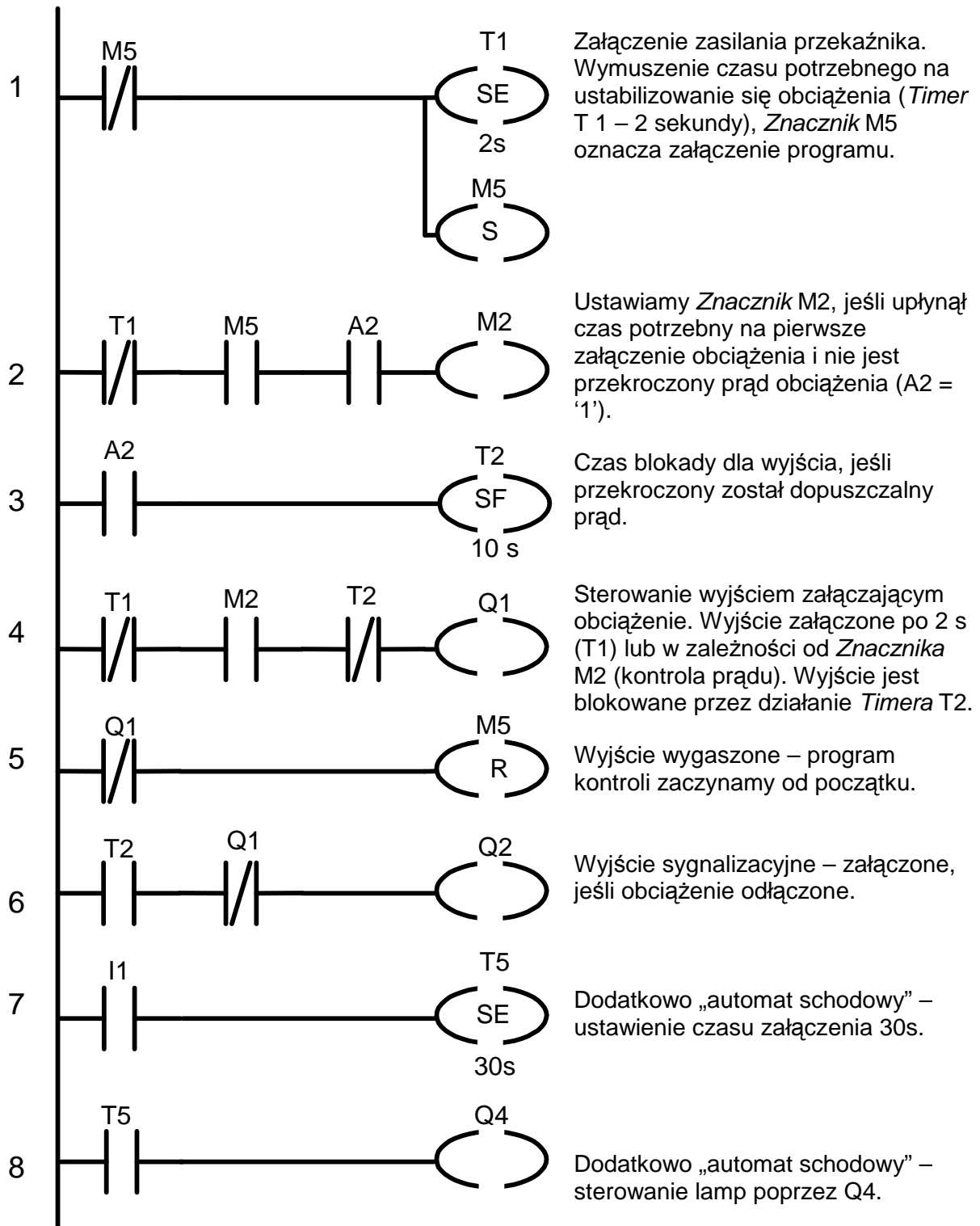
### Program STL

|   |                                |  |
|---|--------------------------------|--|
| 1 | AN M5<br>L 2S<br>SE T1<br>S M5 | Załączenie zasilania przełącznika. Wymuszenie czasu potrzebnego na ustabilizowanie się obciążenia ( <i>Timer</i> T 1 – 2 sekundy), <i>Znacznik</i> M5 oznacza załączenie programu.   |
| 2 | AN T1<br>A M5<br>A A2<br>= M2  | Ustawiamy <i>Znacznik</i> M2, jeśli upłynął czas potrzebny na pierwsze załączenie obciążenia i nie jest przekroczony prąd obciążenia ( $A2 = '1'$ ).   |
| 3 | A A2<br>L 10s<br>SF T2         | Czas blokady dla wyjścia, jeśli przekroczony został dopuszczalny prąd.   |
| 4 | AN T1<br>A M2<br>AN T2<br>= Q1 | Sterowanie wyjściem załączającym obciążenie. Wyjście załączone po 2 s od załączenia (T1) lub w zależności od <i>Znacznika</i> M2 (kontrola prądu). Wyjście jest blokowane przez działanie zwłoki czasowej (T2) po przekroczeniu warunku <i>Komparacji</i> A2 (przekroczona wartość prądu). |
| 5 | AN Q1<br>R M5                  | Wyjście wygaszone – program kontroli zaczynamy od początku.  |
| 6 | A T2<br>AN Q1<br>=Q2           | Wyjście sygnalizacyjne – załączone, jeśli obciążenie odłączone.  |
| 7 | A I1<br>L 30s<br>SE T5         | Dodatkowo „automat schodowy” – ustawienie czasu załączenia 30s.  |
| 8 | A T5<br>= Q4                   | Dodatkowo „automat schodowy” – sterowanie lamp poprzez Q4.   |

#### Uwaga do programu

Należy pamiętać, aby w konfiguracji *Znacznik* M5 nie był *Znacznikiem* remanentnym – po załączeniu zasilania nie powinien być zapamiętywany jego stan.

## Program LAD



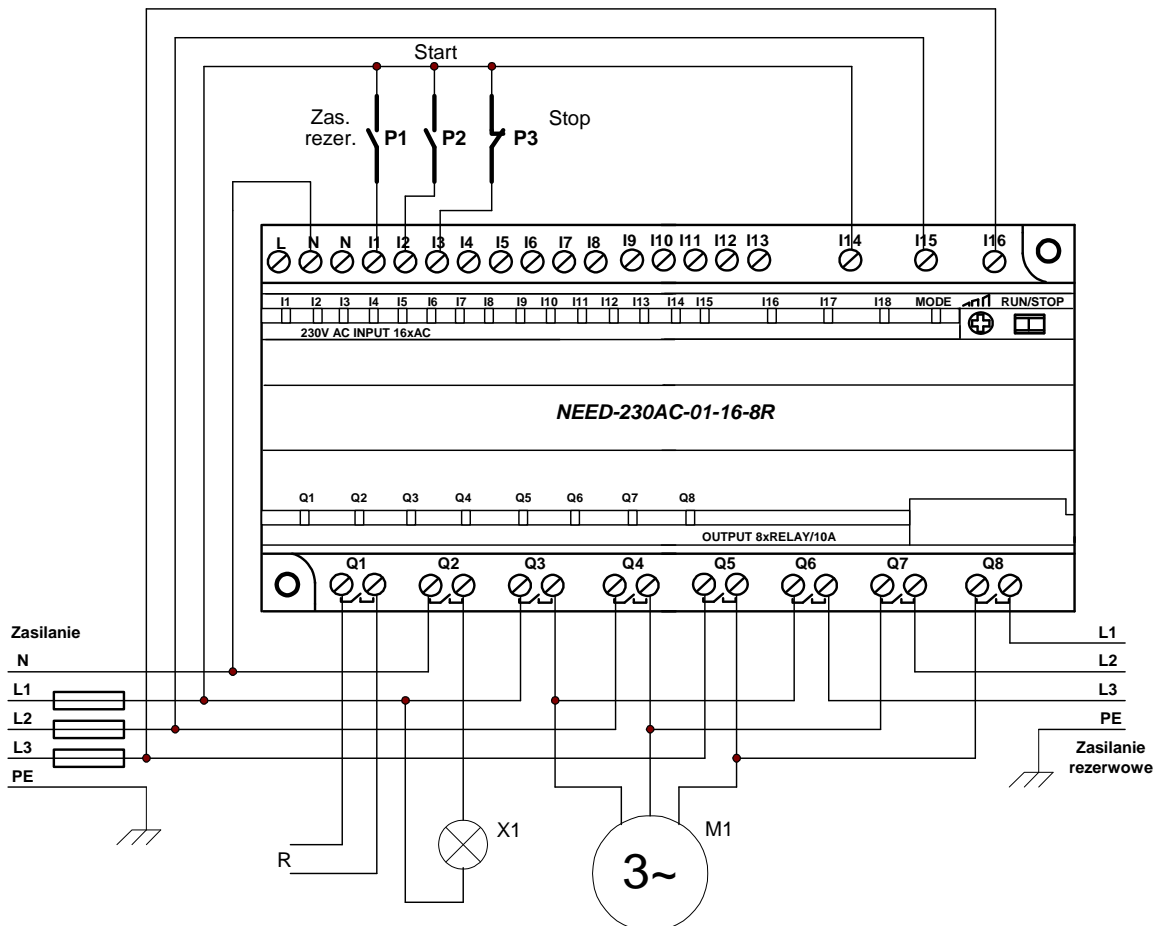
## Komentarz do programu

Obwody 1..6 odnoszą się do schematu na rys. 10.7. Obwody 7 i 8 pokazują dalsze możliwości wykorzystania wolnych wejść/wyjść do zrealizowania prostego sterowania czasowego oświetleniem klatki schodowej.

## 10.8. Sterowanie i zabezpieczenie silnika 3-fazowego

Opis zadania.

Zadaniem przedstawionego układu jest sterowanie silnika i nadzór nad parametrami sieci zasilającej silnik 3-fazowy małej mocy. Dzięki możliwościom przekaźnika NEED 230AC-x1-16-8 możemy bez żadnych dodatkowych elementów zrealizować załączanie/wyłączenie silnika 3-fazowego, kontrolować parametry zasilającej i w razie potrzeby przełączyć na zasilanie rezerwowe.



Rys. 10.8.1. Schemat podłączeń elektrycznych.

Włączanie / wyłączenie silnika odbywa się za pomocą przycisków START / STOP.

Przełącznik „Zasilanie rezerwowe” umożliwia wybór trybu pracy:

- nie załączony – w razie nieprawidłowości sieci zasilającej silnik zostanie odłączony.
- załączony - w razie nieprawidłowości sieci zasilającej silnik będzie pracował na zasilaniu rezerwowym i powróci automatycznie do pracy na zasilaniu sieciowym, jeśli parametry sieci zasilającej wrócą do zadanych granic.

Wbudowana w przekaźnik funkcjonalność umożliwia wykonanie przekaźnika nadzorczego sieci zasilającej spełniającego następujące funkcje:

- a) Kontrola poziomów napięć poszczególnych faz (minimalne  $U_{min}$  i maksymalne napięcie zasilające  $U_{max}$ )
- b) Nadzór kolejności faz
- c) Nadzór zaniku fazy
- d) Nadzór asymetrii –  $U_{asym}$

Powyższe parametry mogą być dowolnie konfigurowane w ustawieniach programu.

Możemy ustawić minimalne i maksymalne napięcie oddzielnie dla każdej fazy, minimalny i maksymalny poziom asymetrii. Modyfikując program możemy zrezygnować z kontroli wybranych parametrów, jeśli jest to nieistotne dla sterowanego układu.

**Program STL**

Program zbudowany jest z trzech części:

1. Określenie typu przekaźnika
2. Definicja nazw symbolicznych
3. Program sterujący (kod źródłowy)

```

1 // Typ przekaźnika
.DEVICE="230AC-X1-16-8"

2 // Zmienne symboliczne
.DEFINE A1_L1min=A1
.DEFINE A2_L2min=A2
.DEFINE A3_L3min=A3
.DEFINE A4_L1max=A4
.DEFINE A5_L2max=A5
.DEFINE A6_L3max=A6
.DEFINE A7_Asym=A7
.DEFINE A8_Asym=A8
.DEFINE I1_Zal_rez=I1
.DEFINE I2_START=I2
.DEFINE I3_STOP=I3
.DEFINE M1_Umin=M1
.DEFINE M2_Umax=M2
.DEFINE M3_Uzas=M3
.DEFINE M4_Asym=M4
.DEFINE M5_Zas_OK=M5
.DEFINE M6_ZALACZ=M6
.DEFINE Q1_R=Q1
.DEFINE Q2_Sygn=Q2
.DEFINE Q3_L1=Q3
.DEFINE Q4_L2=Q4
.DEFINE Q5_L3=Q5
.DEFINE Q6_L1rez=Q6
.DEFINE Q7_L2rez=Q7
.DEFINE Q8_L3rez=Q8
.DEFINE T1_Opozn=T1
.DEFINE T2_sygn=T2

```

```

3 // Program
A %A1_L1min
A %A2_L2min
A %A3_L3min
= %M1_Umin

```

Znacznik M1 ustawiony jeśli napięcie każdej fazy jest większe od Umin.

```

A %A4_L1max
A %A5_L2max
A %A6_L3max
= %M2_Umax

```

Znacznik M2 ustawiony jeśli napięcie każdej fazy jest mniejsze od Umax.

```

A %M1_Umin
A %M2_Umax
= %M3_Uzas

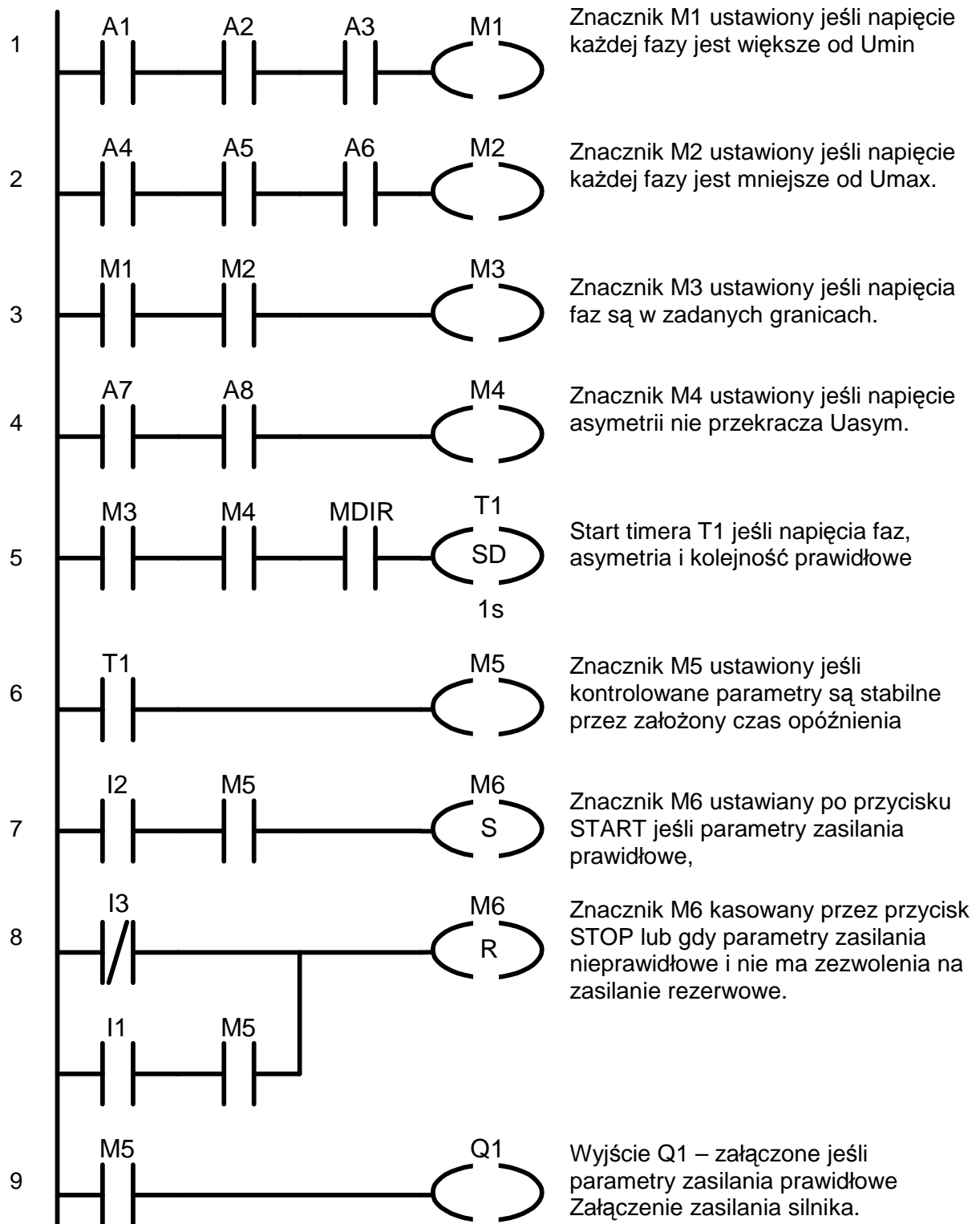
```

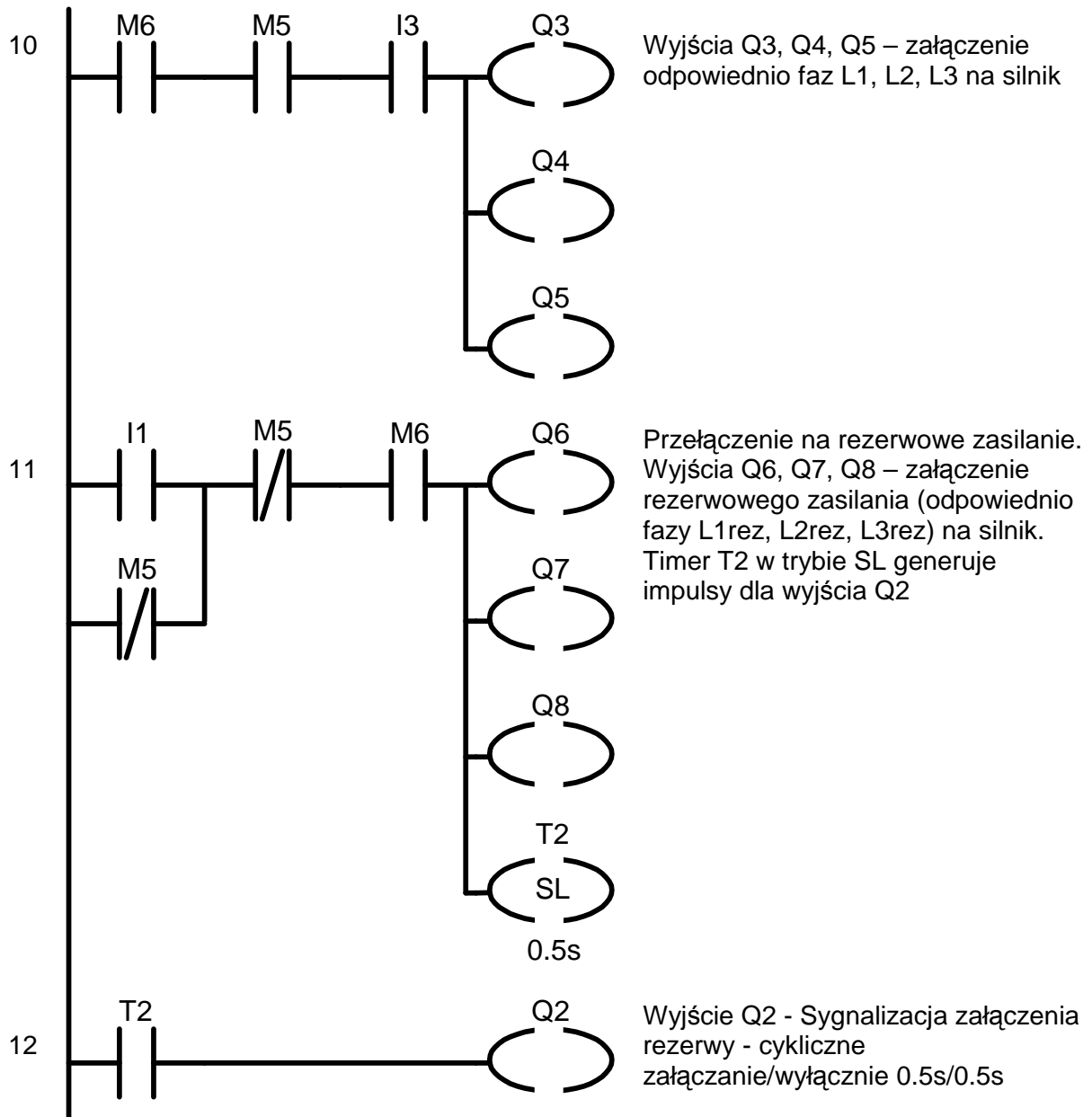
Znacznik M3 ustawiony jeśli napięcia faz są w zadanych granicach.



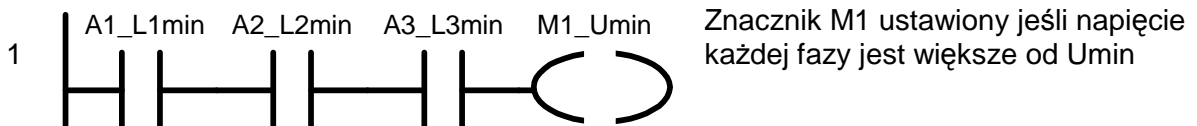
|   |   |  |
|---|---|--|
| 3 | <pre> A %A7_Asym A %A8_Asym = %M4_Asym  A %M3_Uzas A %M4_Asym A MDIR SD %T1_Opozn // 1.00s  A %T1_Opozn = %M5_Zas_OK  A %I2_START A %M5_Zas_OK S %M6_ZALACZ  A(   ON %I3_STOP   O(     AN %I1_Zal_rez     AN %M5_Zas_OK   ) ) R %M6_ZALACZ  A %M5_Zas_OK = %Q1_R  A %M6_ZALACZ A %M5_Zas_OK A %I3_STOP = %Q3_L1 = %Q4_L2 = %Q5_L3  A(   O %I1_Zal_rez   ON %M5_Zas_OK ) AN %M5_Zas_OK A %M6_ZALACZ = %Q6_L1rez = %Q7_L2rez = %Q8_L3rez SL %T2_sygn // 0.50s  A %T2_sygn = %Q2_Sygn </pre> | <p>Znacznik M4 ustawiony jeśli napięcie asymetrii nie przekracza Uasym.</p> <p>Start timera T1 jeśli napięcia faz, asymetria i kolejność prawidłowe</p> <p>Znacznik M5 ustawiony jeśli kontrolowane parametry są stabilne przez założony czas opóźnienia</p> <p>Znacznik M6 ustawiany po przycisku START jeśli parametry zasilania prawidłowe, kasowany przez przycisk STOP lub gdy parametry zasilania nieprawidłowe i nie ma zezwolenia na zasilanie rezerwowe.</p> <p>Wyjście Q1 – załączone jeśli parametry zasilania prawidłowe</p> <p>Załączenie zasilania silnika.<br/>Wyjścia Q3, Q4, Q5 – załączenie odpowiednio faz L1, L2, L3 na silnik</p> <p>Przełączenie na rezerwowe zasilanie. Wyjścia Q6, Q7, Q8 – załączenie rezerwowego zasilania (odpowiednio fazy L1rez, L2rez, L3rez) na silnik. Timer T2 w trybie SL generuje impulsy dla wyjścia Q2</p> <p>Wyjście Q2 - Sygnalizacja załączenia rezerwy - cykliczne załączanie/wyłączenie 0.5s/0.5s.</p> |
|---|---|--|

## Program LAD





Program LAD można również zaopatrzyć w nazwy symboliczne (podobnie jak w STL). Jeśli przypiszemy symbole rejestrów i przełączymy widok na nazwy symboliczne to zamiast nazwy rejestru będzie widoczna nazwa symboliczna. Poniżej przykład dla pierwszej linii.



## Ustawienia

Parametry sieci zasilającej są kontrolowane poprzez wejścia analogowe I14, I15, I16. Napięcia minimalne  $U_{min}$  i maksymalne  $U_{max}$  dla poszczególnych faz są ustawiane w komparatorach A1..A6 odpowiednio:

**A1:** AI14  $\geq$  200V

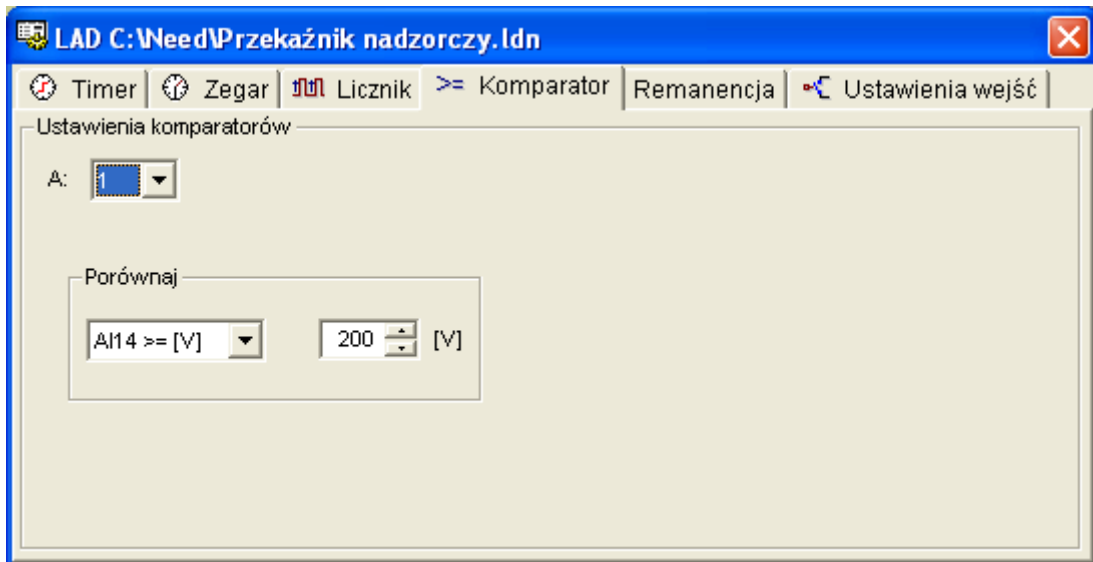
**A2:** AI15  $\geq$  200V

**A3:** AI16  $\geq$  200V

**A4:** AI14  $\leq$  240V

**A5:** AI14  $\leq$  240V

**A6:** AI14  $\leq$  240V



Rys. 10.8.1. Ustawianie komparatora.

Wszystkie fazy mają ustawione takie samo napięcie minimalne 200V i maksymalne 240V. Można oczywiście ustawić je dowolnie.

Komparatory A7 i A8 służą do zadania minimalnego i maksymalnego napięcia asymetrii 0..10V.

**A7:** ASYM  $\geq$  0V

**A8:** ASYM  $\leq$  10V

Dodatkowo ustawić możemy czas opóźnienia timera T1, czyli minimalny czas stabilności prawidłowych parametrów sieci oraz czas impulsów sygnalizacji T2.

**11. DANE TECHNICZNE****Dane ogólne**

|  |   |
|--|---|
| Opis przekaźnika<br>NEED..-x1-08-4R  | 6 wejść cyfrowych AC<br>2 wejścia analogowo - cyfrowe<br>4 wyjścia cyfrowe przekaźnikowe NO<br>Zegar Czasu Rzeczywistego  |
| NEED..-x1-16-8R  | 13 wejść cyfrowych AC<br>3 wejścia analogowo - cyfrowe<br>8 wyjść cyfrowych przekaźnikowych NO<br>Zegar Czasu Rzeczywistego<br>Szybki Licznik<br>Kontrola sieci trójfazowej (wejścia analogowe) |
| Użytkowanie  | W instalacjach niskiego napięcia  |
| Napięcie zasilania<br>NEED-230AC-x1-..<br>NEED-24DC-x1-..<br>NEED-12DC-x1-..   | 95V ÷ 260V AC, 50Hz/60Hz<br>19,6V ÷ 28,8V DC<br>10,2V ÷ 14,4V DC  |
| Montaż:<br>Położenie<br>Mocowanie<br>Miejsce pracy   | Dowolne<br>Na szynie montażowej o szerokości 35mm<br>lub 2 wkręty Ø 4mm<br>W szafie sterowniczej, rozdzielnic<br>instalacyjnej zgodnej z PN-EN 61131-2  |
| Instalacja   | Urządzenie może być montowane tylko<br>przez osobę znającą montaż instalacji<br>elektrycznych   |
| Przewody łączeniowe  | 1×2,5mm <sup>2</sup><br>2×1mm <sup>2</sup>  |
| Maksymalny moment dokręcania śrub<br>zacisków przyłączeniowych   | 0,6Nm   |
| Zgodność z normą   | PN-EN 61131-2   |
| Certyfikaty  | CE, B, UL, VDE, GOST  |
| Wymiary:<br>NEED..-x1-08-4R:<br>Szerokość<br>Długość<br>Wysokość<br>Masa<br>NEED..-x1-16-8R:<br>Szerokość<br>Długość<br>Wysokość | <br><br>72mm<br>90mm<br>55mm<br>210g<br><br>132mm<br>90mm<br>55mm   |

|      |      |
|------|------|
| Masa | 400g |
|------|------|

### Warunki i wymagania klimatyczne i mechaniczne

|   |  |
|---|--|
| Temperatura robocza   | Od -20°C do +55°C  |
| Wilgotność względna   | Od 10% do 95%, bez kondensacji   |
| Ciśnienie atmosferyczne   | 795hPa do 1080hPa  |
| Stopień zanieczyszczenia  | 2  |
| Drgania dopuszczalne w stanie roboczym (PN-EN 60068-2-6)  | 5Hz do 9Hz (stała amplituda 3,5mm)<br>9Hz do 150Hz (stałe przyspieszenie 1g) |
| Udary (PN-EN 60068-2-27)  | 6 uderzeń (półsinusoidea 15g/11ms)   |
| Upadki płaskie i podtrzymywane (PN-EN 60068-2-31)   | 100mm, 2 próby<br>30°, 2 próby   |
| Temperatura składowania/transportu  | -40°C do +70°C   |
| Swobodne spadki (PN-EN 60068-2-32):<br>wyrób w opakowaniu transportowym<br>wyrób w opakowaniu sprzedażnym | 1 000mm<br>300mm   |

### Wymagania bezpieczeństwa

|                                       |          |
|---------------------------------------|----------|
| Znamionowe napięcie izolacji:         | 300V AC  |
| Znamionowe napięcie udarowe           | 2 500V   |
| Kategoria przepięciowa:               | Klasa II |
| Stopień ochrony obudowy (PN-EN 60529) | IP 20    |
| Klasa odporności ogniowej (UL94)      | V0       |

**Wymagania kompatybilności elektromagnetycznej (EMC)**

|  |   |
|--|---|
| Emisja zaburzeń promieniowych (PN-EN 55011)<br>NEED..-x1-08-4R                           | Klasa wartości granicznej A, grupa 1                              |
| NEED..-x1-16-8R  | Klasa wartości granicznej B, grupa 1                              |
| Emisja zaburzeń przewodzonych (PN-EN 55011)<br>NEED..-x1-08-4R                           | Klasa wartości granicznej A, grupa 1                              |
| NEED..-x1-16-8R  | Klasa wartości granicznej B, grupa 1                              |
| Odporność na wyładowanie elektrostatyczne (PN-EN 61000-4-2)                              | 8kV – wyładowanie powietrzne,<br>4kV – wyładowanie powierzchniowe |
| Podatność na pola elektromagnetyczne o częstotliwości radiowej (PN-EN 61000-4-3)         | 10V/m,<br>80MHz – 1 000MHz<br>800MHz – 960MHz<br>1,4GHz – 2,0GHz  |
| Seria szybkich elektrycznych stanów przejściowych (PN-EN 61000-4-4):<br>NEED-230AC-x1-.. | 2kV – przewody zasilające<br>2kV – przewody sygnałowe             |
| NEED-24DC-x1..., NEED-12DC-x1-..   | 2kV – przewody zasilające<br>1kV – przewody sygnałowe             |
| Udar wysokiej energii (PN-EN 61000-4-5):<br>NEED-230AC-x1-..<br>port zasilania           | 2kV – sygnał niesymetryczny<br>1kV – sygnał symetryczny           |
| porty obwodów wejściowych  | 2kV – sygnał niesymetryczny<br>1kV – sygnał symetryczny           |
| NEED-24DC-x1-..., NEED-12DC-x1-..<br>port zasilania                                      | 1kV – sygnał niesymetryczny<br>0,5kV – sygnał symetryczny         |
| porty obwodów wejściowych (linie nieekranowane)  | 0,5kV – sygnał niesymetryczny<br>0,5kV – sygnał symetryczny       |
| Odporność na zaburzenia o częstotliwości radiowej  | 3V,<br>26 – 80MHz   |

**Obwód zasilania**

|  |   |
|--|---|
| Napięcie zasilania:<br>NEED-230AC-x1-..<br>wartość znamionowa<br>zakres pracy  | 115V / 230V AC, 60Hz/50Hz<br>95V – 260V               |
| NEED-24DC-x1-..<br>wartość znamionowa<br>zakres pracy  | 24V DC<br>19,6V – 28,8V DC                            |
| NEED-12DC-x1-..<br>wartość znamionowa<br>zakres pracy  | 12V DC<br>10,2V – 14,4V DC                            |
| Prąd znamionowy (dla wysokich stanów na<br>wszystkich wejściach i wyjściach)<br>NEED-230AC-x1-08-4R<br>NEED-24DC-x1-08-4R<br>NEED-12DC-x1-08-4R, | 19mA<br>70mA<br>120mA                                 |
| NEED-230AC-x1-16-8R<br>NEED-24DC-x1-16-8R<br>NEED-12DC-x1-16-8R,   | 40mA<br>160mA<br>260mA                                |
| Maksymalny pobór mocy:<br>NEED-230AC-x1-08-4R<br>NEED-24DC-x1-08-4R<br>NEED-12DC-x1-08-4R  | < 5VA<br>< 3W<br>< 3W                                 |
| NEED-230AC-x1-16-8R<br>NEED-24DC-x1-16-8R<br>NEED-12DC-x1-16-8R  | <10VA<br><5W<br><5W                                   |
| Wyższe harmoniczne w sygnale zasilania<br>NEED-230AC-x1-..   | < 10% wartości napięcia składowej<br>podstawowej      |
| Zabezpieczenie prądowe w obwodzie zasilania<br>NEED-24DC-x1-.., NEED-12DC-x1-..  | 500mA<br><br>Zabezpieczenie przed zamianą<br>biegunów |
| Zabezpieczenie prądowe w obwodzie zasilania<br>NEED-230AC-x1-..  | 600mA   |
| Odporność na łagodną i szybką zmianę napięcia<br>zasilania   | PN-EN 61131-2   |
| Przerwy napięcia zasilania (PN-EN 61131-2)   | 20ms  |
| Podtrzymania Zegara Czasu Rzeczywistego  | 64h w T=+25°C<br>24h w T=+55°C                        |



## Specyfikacja obwodów wejściowych

|  |  |
|--|--|
| Typ wejść cyfrowych (PN-EN 61131-2)  | typ 1 (wejścia odbierające prąd)                           |
| Liczba:<br>NEED...x1-08-4R   | 8 (I1-I8)  |
| NEED...x1-16-8R  | 16 (I1-I16)  |
| Wizualizacja stanu logicznego  | diody LED  |
| Napięcie znamionowe:<br>NEED-230AC-x1-08-4<br>dla stanu logicznego '1'<br>dla stanu logicznego '0' | 85V – 260V<br>0V – 40V                                     |
| NEED-230AC-x1-16-8<br>dla stanu logicznego '1'<br>dla stanu logicznego '0'                         | 85V – 260V<br>0V – 32V                                     |
| NEED-24DC-x1-..<br>dla stanu logicznego '1'<br>dla stanu logicznego '0'                            | 15 – 40V<br>-3V – 5V                                       |
| NEED-12DC-x1-..<br>dla stanu logicznego '1'<br>dla stanu logicznego '0'                            | 8V – 26V<br>-1,5V – 4V                                     |
| Prąd wejściowy dla stanu logicznego '1':   |  |
| NEED-230AC-x1-08-4 (dla 230V AC)   | 0,6mA (I1 – I4)<br>8,0mA (I5 – I6)<br>0,9mA (I7 – I8)      |
| NEED-24DC-x1-08-4 (dla 24V DC)   | 3,3mA (I1 – I6)<br>2,0mA (I7 – I8)                         |
| NEED-12DC-x1-08-4 (dla 12V DC)   | 3,3mA (I1 – I6)<br>1,1mA (I7 – I8)                         |
| NEED-230AC-x1-16-8 (dla 230V AC)   | 0,6mA (I1 – I11)<br>8,0mA (I12 – I13)<br>1,5mA (I14 – I16) |
| NEED-24DC-x1-16-8 (dla 24V DC)   | 3,3mA (I1 – I13)<br>2,0mA (I14 – I16)                      |
| NEED-12DC-x1-16-8 (dla 12V DC)   | 3,3mA (I1 – I13)<br>1,1mA (I14 – I16)                      |
| Maksymalny czas opóźnienia przy przejściu ze stanu logicznego '0' na '1':                          |  |
| NEED-230AC-x1-..<br>eliminacja odbić styków ZAŁ<br>eliminacja odbić styków WYŁ                     | 60ms<br>20ms   |
| NEED-24DC-x1-..., NEED-12DC-x1-..<br>eliminacja odbić styków ZAŁ                                   | 21ms   |

|   |                                      |
|---|--------------------------------------|
| eliminacja odbić styków WYŁ   | 0,20ms + czas cyklu programu         |
| Maksymalny czas opóźnienia przy przejściu ze stanu logicznego '1' na '0' :  |                                      |
| NEED-230AC-x1-..<br>eliminacja odbić styków ZAŁ<br>eliminacja odbić styków WYŁ  | 60ms<br>20ms                         |
| NEED-24DC-x1-..., NEED-12DC-x1-..<br>eliminacja odbić styków ZAŁ<br>eliminacja odbić styków WYŁ                       | 21ms<br>0,25ms + czas cyklu programu |
| Znamionowe napięcie izolacji:   | 300V AC                              |
| Separacja galwaniczna:<br>od napięcia zasilania<br>względem siebie<br>względem wyjść                                  | nie<br>nie<br>tak                    |
| Maksymalna dopuszczalna długość przewodów (przewód L i sygnałowy prowadzone razem):                                   |                                      |
| NEED-230AC-x1-08-4:<br>dla wejść cyfrowych I1 – I4<br>dla wejść cyfrowych I5 – I6<br>dla wejść cyfrowych I7 – I8      | 10m<br>100m<br>10m                   |
| NEED-230AC-x1-16-8:<br>dla wejść cyfrowych I1 – I11<br>dla wejść cyfrowych I12 – I13<br>dla wejść cyfrowych I14 – I16 | 10m<br>100m<br>10m                   |
| NEED-24DC-x1-08-4, NEED-12DC-x1-08-4<br>dla wejść cyfrowych I1 – I8   | 100m                                 |
| NEED-24DC-x1-16-8, NEED-12DC-x1-16-8<br>dla wejść cyfrowych I1 – I16  | 100m                                 |

### Specyfikacja obwodów wyjściowych

|                                     |   |
|-------------------------------------|---|
| Typ wyjść cyfrowych (PN-EN 61131-2) | przełącznikowe – styki NO,<br>niezabezpieczone<br>(wyjścia cyfrowe AC dostarczające prąd) |
| Liczba:<br>NEED..-x1-08-4R          | 4   |
| NEED..-x1-16-8R                     | 8   |
| Wizualizacja stanu logicznego       | diody LED   |

|   |  |
|---|--|
| Równoległe łączenie wyjść dla zwiększenia obciążalności   | niedopuszczalne  |
| Zewnętrzne zabezpieczenie obwodu wyjść  | 16A (wyłącznik instalacyjny B16)                                       |
| Znamionowy prąd obciążenia w kat. AC1   | 10A AC   |
| Znamionowe napięcie obciążenia w kat. AC1   | 250V AC  |
| Minimalny prąd zestyków   | 10mA   |
| Minimalne napięcie zestyków   | 10V  |
| Rezystancja styków  | <100mΩ   |
| Całkowity prąd wyjściowy (PN-EN 61131-2):<br>NEED...x1-08-4R  | 40A (4x10A)  |
| NEED...x1-16-8R   | 80A (8x10A)  |
| Znamionowe napięcie izolacji<br>wzmocniona<br>podstawowa  | 300V<br>między wejściami a wyjściami<br>między wyjściami               |
| Napięcie probiercze przerwy stykowej  | 1 000V AC  |
| Czas zadziałania  | 7ms  |
| Czas powrotu  | 3ms  |
| Maksymalna częstotaść łączeń:<br>- przy obciążeniu znamionowym<br>(kategoria AC1)<br>- bez obciążenia       | 600cykli/h<br>72 000cykli/h  |
| Trwałość łączeniowa:<br>- w kategorii AC1<br>- w zależności od stałej czasowej T<br>(L/R=40ms)              | >1,7×10 <sup>5</sup> (10A 250V AC)<br>>10 <sup>5</sup> (0,15A 220V DC) |
| Trwałość mechaniczna  | 3×10 <sup>7</sup> cykli łączeniowych                                   |
| Separacja galwaniczna względem:<br>- napięcia zasilania<br>- wejść cyfrowych<br>- złącza PC i karty pamięci | tak<br>tak<br>tak  |

## Specyfikacja wejść analogowych

| Typ wejścia  | wejścia analogowe  |
|--|--|
| Liczba:<br>NEED..-x1-08-4  | 2 (I7 – I8)  |
| NEED..-x1-16-8   | 3 (I14 – I16)  |
| Rodzaj wejść:<br>NEED-230AC-x1..<br>NEED-24DC-x1.., NEED-12DC-x1..<br>NEED-24DC-x1-16-8, NEED-12DC-x1-16-8 | napięciowe, sygnał naprzemienny<br>napięciowe, sygnał stały<br>prądowe, sygnał stały |
| Impedancja wejściowa:  |  |
| NEED-230AC-x1-08-4 dla I7 – I8   | 200k $\Omega$ dla półfali dodatniej<br>400k $\Omega$ dla półfali ujemnej             |
| NEED-230AC-x1-16-8 dla I14 – I16   | 200k $\Omega$ dla półfali dodatniej<br>400k $\Omega$ dla półfali ujemnej             |
| NEED-230AC-x1-08-4 dla I5 – I6   | 28,75k $\Omega$  |
| NEED-230AC-x1-16-8 dla I12 – I13   | 28,75k $\Omega$  |
| NEED-230AC-x1-08-4 dla I1 – I4   | 400k $\Omega$  |
| NEED-230AC-x1-16-8 dla I1 – I11  | 400k $\Omega$  |
| NEED-24DC-x1..   | 12,36 k $\Omega$   |
| NEED-12DC-x1-..  | 10,92 k $\Omega$   |
| NEED-24DC-x1-16-8, NEED-12DC-x1-16-8<br>dla I14 – I16 na zakresie prądowym                                 | 49 $\Omega$  |
| Zakres sygnałów wejściowych:   |  |
| NEED-230AC-x1-..   | 0V - 255V AC   |
| NEED-24DC-x1-..  | -3V - 25,5V DC   |
| NEED-12DC-x1-..  | -1,5V - 14,4V DC   |
| NEED-24DC-x1-16-8  | 0 – 51mA (zakres prądowy)  |
| NEED-12DC-x1-16-8  | 0 – 51mA (zakres prądowy)  |
| Prąd wejściowy:  |  |
| NEED-230AC-x1-08-4 (dla 230V AC)   | 0,9mA  |
| NEED-24DC-x1-08-4 (dla 24V DC)   | 2,0mA  |
| NEED-12DC-x1-08-4 (dla 12V DC)   | 1,1mA  |
| NEED-230AC-x1-16-8 (dla 230V AC)   | 1,5mA  |
| NEED-24DC-x1-16-8 (dla 24V DC)   | 2,0mA  |
| NEED-12DC-x1-16-8 (dla 12V DC)   | 1,1mA  |
| Czas konwersji   | 1ms  |
| Rozdzielczość cyfrowa:   |  |
| NEED-230AC-x1-..   | 1V AC  |
| NEED-12DC-x1-08-4 (zakres napięciowy)  | 0,1V   |
| NEED-12DC-x1-16-8 (zakres napięciowy)  | 0,1V lub 0,05V   |
| NEED-24DC-x1-08-4 (zakres napięciowy)  | 0,1V   |
| NEED-24DC-x1-16-8 (zakres napięciowy)  | 0,1V lub 0,05V   |
| NEED-12DC-x1-16-8 (zakres prądowy)   | 0,2mA lub 0,1mA  |

|   |  |
|---|--|
| NEED-24DC-x1-16-8 (zakres prądowy)  | 0,2mA lub 0,1mA                                    |
| Maksymalne dopuszczalne przeciążenie ciągłe:<br>NEED-230AC-x1-..<br>NEED-24DC-x1-..<br>NEED-12DC-x1..   | 300V AC<br>40V DC<br>26V DC                        |
| Błąd wejścia analogowego:<br>błąd maksymalny przy 25°C<br>NEED-230AC-x1-..<br>NEED-24DC-x1-..., NEED-12DC-x1-..                                   | ±3% zakresu pomiarowego<br>±2% zakresu pomiarowego |
| Przesłuch między kanałami   | 36dB   |
| Nieliniowość  | ±3%  |
| Długość przewodów (ekranowane)  | 40m  |
| Separacja galwaniczna:<br>- z napięciem zasilania<br>- względem wejść cyfrowych<br>- względem wyjść cyfrowych<br>- względem złącza programującego | nie<br>nie<br>tak<br>nie                           |

### Jednostka centralna i pamięć

|  |                                   |
|--|-----------------------------------|
| Pojemność pamięci programu użytkownika<br>NEED..-x1-08-4 | 862 bajty                         |
| NEED..-x1-16-8   | 835 bajty                         |
| Rodzaj dostępnej pamięci                                 | EEPROM                            |
| Języki programowania (PN-EN 61131-3)                     | Tekstowy (STL)<br>Graficzny (LAD) |

|  |  |
|--|--|
| Zasoby programowe<br>NEED..-x1-08-4:<br>Znaczniki<br>Timery<br>dokładność Timerów<br>Liczniki (zliczanie w górę/dół)<br>Wartości zliczane<br>Komparatory<br>Ilość możliwych porównań<br>Zegar tygodniowy<br>Zegar Czasu Rzeczywistego<br>Dokładność Zegara Czasu Rzeczywistego   | 16<br>8<br>$\pm 1\%$ wartości ustawionej + (0 - 1)ms<br>8<br>0-65535<br>8<br>10<br>4x4 kanały<br>1<br>$\pm 3$ s/dzień        |
| NEED..-x1-16-8:<br>Znaczniki<br>Znacznik kierunku faz<br>Timery<br>Dokładność Timerów<br>Liczniki (zliczanie w górę/dół)<br>Wartości zliczane<br>Komparatory<br>Ilość możliwych porównań<br>Zegar tygodniowy<br>Zegar Czasu Rzeczywistego<br>(automatyczna zmiana czasu w różnych strefach czasowych)<br>Dokładność Zegara Czasu Rzeczywistego | 16<br>1<br>16<br>$\pm 1\%$ wartości ustawionej + (0 - 1)ms<br>8<br>0-65535<br>12<br>22<br>4x4 kanały<br>1<br>$\pm 3$ s/dzień |
| Remanencja:<br>Czas podtrzymania Zegara<br><br>Znaczniki<br>Liczniki<br>Timery   | 64h (przy 25°C)<br>24h (przy 40°C)<br>M1 – M16<br>C5 – C8<br>T5 – T8   |

**Zewnętrzna karta pamięci**

|  |                     |
|--|---------------------|
| Opis złącza<br>NEED-M-1K (złącze typu A) | Złącze dwurzędowe   |
| NEED-M-1KB (złącze typu B)               | Złącze jednorzędowe |
| Wymiary (długość × wysokość × głębokość) | 30mm x 11mm x 5mm   |
| Masa                                     | 2g                  |
| Typ pamięci                              | EEPROM              |
| Pojemność pamięci                        | 1KB                 |
| Typ interfejsu złącza                    | I <sup>2</sup> C    |

**Kabel dedykowany**

|             |  |
|-------------|--|
| Opis złącza |  |
|-------------|--|

|  |                                   |
|--|-----------------------------------|
| NEED-PC-15A (złącze typu A)                        | Złącze dwurzędowe                 |
| NEED-PC-15B (złącze typu B)                        | Złącze jednorzędowe               |
| Długość kabla                                      | 2m                                |
| Masa   | 100g                              |
| Sposób podłączenia:<br>- z PC<br>- z przekaźnikiem | 9 pin D-Sub<br>Gniazdo dedykowane |
| Typ kabla  | RS 232                            |
| Szybkość transmisji danych cyfrowych               | 19200bit/s                        |
| Kontrola danych                                    | Sumy kontrolne                    |

## 12. SŁOWNIK

|                                  |   |
|----------------------------------|---|
| <b>Czas cyklu</b>                | – czas przetwarzania wszystkich instrukcji programu   |
| <b>Hasło</b>                     | – zabezpieczenie przed kopiowaniem programu znajdującego się w sterowniku   |
| <b>Karta pamięci</b>             | – zewnętrzna pamięć przekaźnika programowalnego, z której można przepisywać program do wewnętrznej pamięci przekaźnika  |
| <b>Kompilacja</b>                | – sprawdzanie poprawności programu i generowanie kodu zrozumiałego dla procesora przekaźnika programowalnego            |
| <b>Konfiguracja</b>              | – ustawienie odpowiednich wartości parametrów dla przekaźnika programowalnego   |
| <b>LAD</b>                       | – język graficzny programowania przekaźnika   |
| <b>Licznik</b>                   | – element logiczny zasobów wewnętrznych przekaźnika używany w programie przy realizacji zliczających funkcji sterowania |
| <b>Ładowanie programu</b>        | – zapis skompilowanego programu z PC do pamięci przekaźnika programowalnego   |
| <b>Pamięć programu</b>           | – obszar pamięci przekaźnika przeznaczony do zapisywania programu sterowania przez użytkownika                          |
| <b>Program</b>                   | – zapis określonego procesu sterowania za pomocą odpowiedniego języka programowania                                     |
| <b>Przełącznik programowalny</b> | – przełącznik posiadający wejścia (styki), wyjścia (cewki) oraz programowalne zasoby logiczne wraz z pamięcią           |
| <b>RUN</b>                       | – jeden z trybów pracy przekaźnika, w którym następuje normalne przetwarzanie programu                                  |
| <b>STL</b>                       | – język tekstowy programowania przekaźnika  |
| <b>STOP</b>                      | – jeden z trybów pracy przekaźnika, w którym przekaźnik nie wykonuje programu – wyjścia przekaźnika są odłączone        |
| <b>Timer</b>                     | – element logiczny zasobów wewnętrznych przekaźnika używany w programie przy realizacji czasowych funkcji sterowania    |



---

|                          |  |
|--------------------------|--|
| <b>Trigger</b>           | – wejście wyzwalające odmierzenie czasu w Timerach   |
| <b>Wejście</b>           | – fizyczne wejście przekaźnika, do którego dołącza się zewnętrzne sygnały z czujników, styków itp.   |
| <b>Wejście analogowe</b> | – fizyczne wejście przekaźnika, do którego dołącza się sygnały analogowe   |
| <b>Wyjście</b>           | – fizyczne wyjście przekaźnika, do którego podłącza się urządzenia sterowane: lampki, styczniki, zawory elektromagnetyczne itp.              |
| <b>Zasoby remanentne</b> | – elementy logiczne przekaźnika, które pamiętają swój stan po wyłączeniu napięcia zasilającego   |
| <b>Zegar</b>             | – element logiczny zasobów wewnętrznych przekaźnika używany w programie przy realizacji funkcji sterowania wykorzystującego czas rzeczywisty |
| <b>Znacznik</b>          | – element logiczny zasobów wewnętrznych przekaźnika używany w programie przy realizacji funkcji sterowania                                   |

### 13. INDEKS

- =, instrukcja, 79
- AND -A, 68
- AND NOT -AN, 70
- AND NOT( -AN(, 71
- AND( -A(, 68
- ASYM, 56
- CD, instrukcja Licznika, 87
- CLR, instrukcja, 99
- CU, instrukcja Licznika, 86
- Cykl programu, 26
- Deinstalacja, 115
- Działanie, Licznik CD, 41
- Działanie, Licznik CU, 41, 86
- Działanie, Timer SD, 37, 81
- Działanie, Timer SE, 38
- Działanie, Timer SF, 38, 82
- Działanie, Timer SL, 39, 84
- Działanie, Zegar, 42
- Edytor, LAD, 127
- Edytor, STL, 125
- FP, instrukcja, 80
- Instalacja, 115
- Instrukcje dla wejść analogowych, 91
- Instrukcje Zegara, 90
- Komparator, porównania, 56
- Konfiguracja, Zegar, 43
- LAD, 100
- LAD, Liczniki, 108
- LAD, obwód, 102
- LAD, program, 102
- LAD, Timery, 107
- LAD, wejścia, 101, 105
- LAD, wyjścia, 101, 105
- LAD, zasady, 110
- LAD, Znaczniki, 106
- Liczniki, 39
- Liczniki, liczba impulsów do zliczenia, 40
- Liczniki, wejścia, 40
- Liczniki, wyjście, 40
- LOAD – L,, 92
- Menu, 121
- Menu, opis, 122
- Mocowanie na szynie DIN, 11
- Mocowanie na śruby, 12
- Narzędzia, 121
- Opóźnienie wejść, 149
- Opóźnienie wyjść, 152
- OR NOT( -ON(, 74
- OR -O, 72
- OR( -O(, 73
- Pamięć programowanie, 154
- Pamięć zewnętrzna, 153
- Pamięć, kopiowanie, 156
- Pamięć, partycje, 153
- pliki, rodzaje, 136
- Podgląd zmiennych, 141
- Podłączenie, wejścia cyfrowe, 14, 17
- Podłączenie, wejścia analogowe, 18, 20
- Podłączenie, wyjścia, 23
- Podłączenie, zasilanie, 24, 25
- Połączenie, z komputerem, 115
- Potencjometr, 60
- Projekt, 116
- Przykłady, zastosowania, 157
- Remanencje, 60
- Reset -R, 79
- S, instrukcja, 79
- SE, instrukcja Timera, 83
- SET, instrukcja, 98
- SF, instrukcja Timera, 82
- SL, instrukcja Timera, 84, 85
- STL, 64
- Symbole, Komparator, 55
- Symbole, LAD, 100, 103
- Symbole, Timer SD - Opóźnione  
załączenie, 37
- Symbole, Timer SE – Pojedynczy impuls,  
38
- Symbole, Timer SF - Opóźnione  
wyłączenie, 37
- Symbole, Timer SL - Impulsy, 38
- Symbole, wejścia cyfrowe, 30
- Symbole, wyjścia cyfrowe impulsowe, 31
- Symbole, wyjścia cyfrowe kasujące, 31
- Symbole, wyjścia cyfrowe normalne, 30
- Symbole, wyjścia cyfrowe sterujące, 32
- Symbole, wyjścia cyfrowe ustawiające, 32
- Symbole, wyjścia cyfrowe zanegowane  
sterujące, 32
- Symbole, Znaczniki, 33
- Szybki Licznik, 41
- Timer, czas do odmierzenia, 36
- Timer, wejścia, 36
- Timer, wyjścia, 37
- Timer, zakresy czasowe, 36
- Timery, 36
- Typ, oznaczenie, 9
- Uruchomienie, 145
- Ustawienia, 136
- Ustawienia, Komparatory, 140
- Ustawienia, Liczniki, 139
- Ustawienia, opóźnienia wejść, 141
- Ustawienia, Remanencje, 140
- Ustawienia, Timery, 138
- Ustawienia, Zegary, 139
- XOR NOT -XN, 77

XOR NOT( -XN(, 78

XOR -X, 75

XOR( -X(, 76

Zaciski, przewody - rodzaj, 13

Zasilanie, 146

Zasoby przekaźnika, 28

Zegar Czasu Rzeczywistego, 54

Zegar Tygodniowy, 42

Znacznik MDIR, 34

Znaczniki, 33