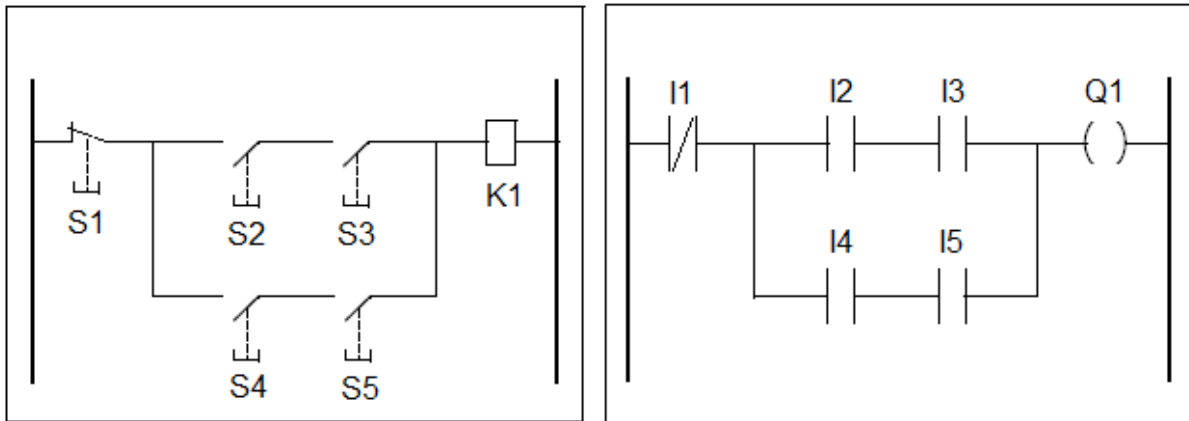


Język programowania LD – wprowadzenie

Język LD (Ladder Diagram) jest odzwierciedleniem przekaźnikowego układu sterowania, więc interpretacja działania programu napisanego w języku LD jest podobna do interpretacji działania przekaźnikowego układu sterowania.



Przekaźnikowy układ sterowania

Program w języku LD

W przekaźnikowym układzie sterowania o zasileniu cewki K1 decydują stany poszczególnych przycisków S (przycisk może być naciśnięty lub nienaciśnięty). Naciśnięcie przycisku zmienia jego stan (otwiera lub zamyka zestyk przycisku). Aby w widocznym powyżej przekaźnikowym układzie sterowania prąd doptynął do cewki K1, należy spełnić jeden z dwóch oczywistych warunków:

- 1) S1 nienaciśnięty i S2 naciśnięty i S3 naciśnięty,
- 2) S1 nienaciśnięty i S4 naciśnięty i S5 naciśnięty.

W programie napisanym w języku LD o załączeniu wyjścia Q decydują stany wejść I sterownika (do wejścia sterownika może doptywać prąd – wejście zasilone lub do wejścia może nie doptywać prąd – wejście niezasilone). **Zasilenie wejścia I sterownika zmienia stan zestyku powiązane z tym wejściem w programie napisanym w LD.** Zatem aby w widocznym powyżej programie LD wyjście Q1 zostało załączone (aktywowane), również należy spełnić jeden z dwóch warunków:

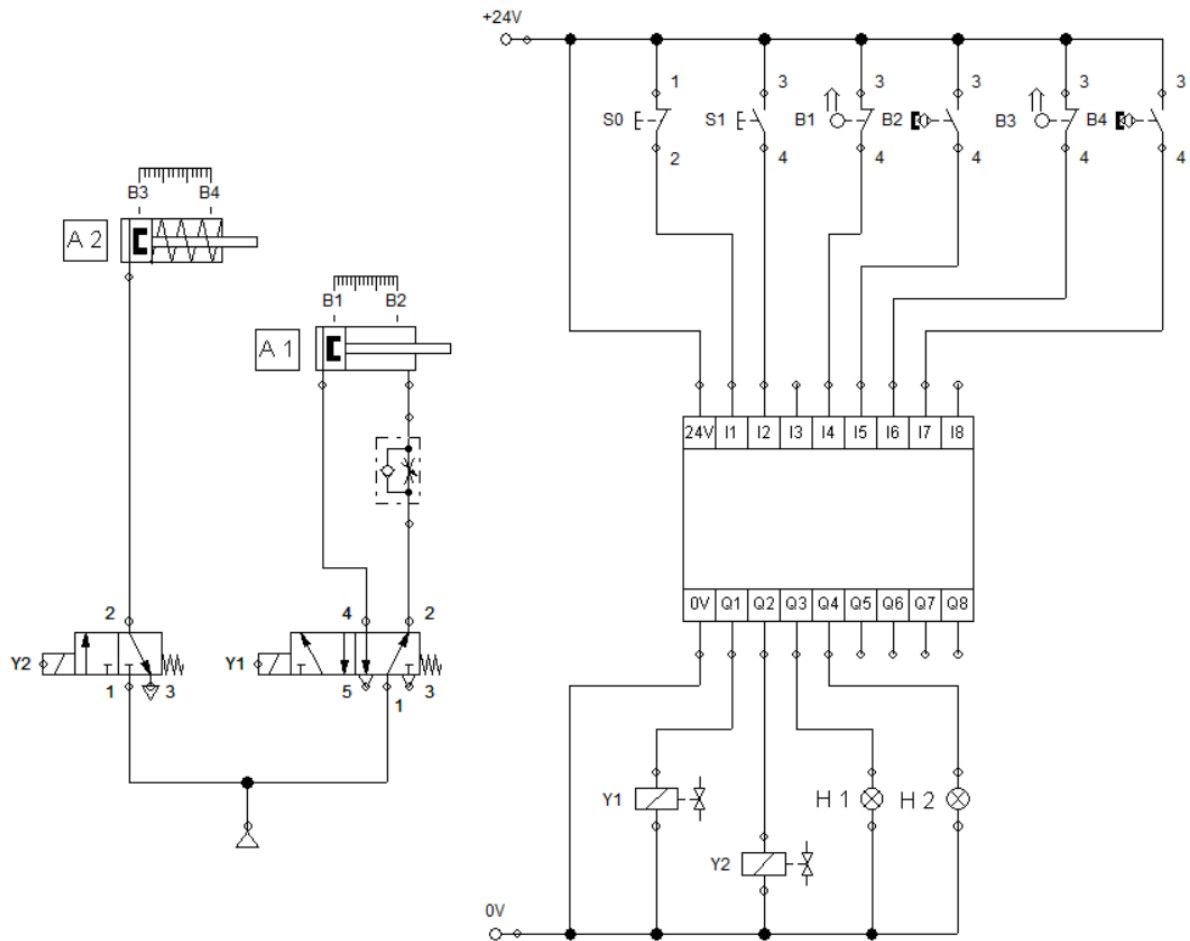
- 1) I1 niezasilone i I2 zasilone i I3 zasilone,
- 2) I1 niezasilone i I4 zasilone i I5 zasilone.

Wejście niezasilone (nieaktywne) zapisuje się jako I=0.

Wejście zasilone (aktywne) zapisuje się jako I=1.

O tym, czy dane wejście sterownika jest zasilone decyduje stan przycisku (lub czujnika) dołączonego do tego wejścia w rzeczywistym układzie oraz rodzaj zestyku użytego w tym przycisku (lub czujniku). Przykładowo, dla układu ze schematu na następnej stronie:

- 1) Do wejścia I2 sterownika dołączony jest przycisk NO – wejście I2 będzie więc zasilone dopiero po naciśnięciu przycisku.
- 2) Do wejścia I1 sterownika dołączony jest przycisk NC – wejście I1 będzie więc zasilane aż do momentu naciśnięcia przycisku (naciśnięcie przycisku NC przerywa dopływ prądu do wejścia).



Poniżej zamieszczono kilka prostych programów w języku LD z interpretacją ich działania:

1. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejście I1 będzie aktywne ($I1=1$):



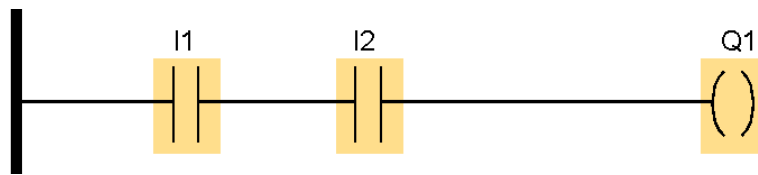
Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 **nie** będzie naciśnięty.

2. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejście I1 będzie nieaktywne ($I1=0$):



Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 będzie naciśnięty.

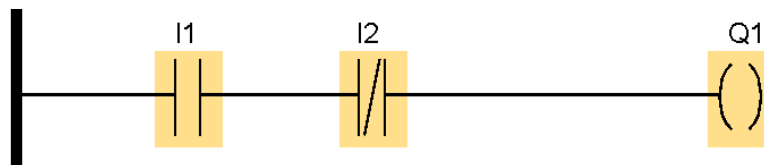
3. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejścia I1 i I2 będą aktywne ($I1=1$ i $I2=1$):



Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 **nie będzie** naciśnięty i przycisk S1 będzie naciśnięty.

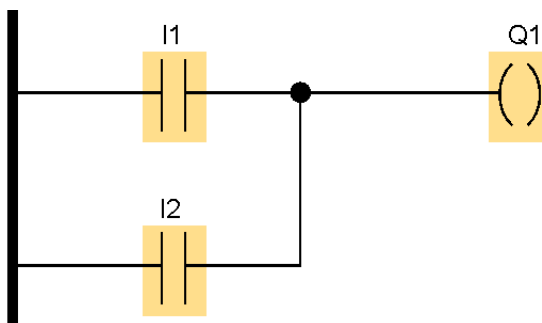
Szeregowe połączenie symboli wejść w programie oznacza realizację operacji logicznej AND (i).

4. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejście I1 będzie aktywne ($I1=1$) i wejście I2 będzie nieaktywne ($I2=0$):



Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 **nie będzie** naciśnięty i przycisk S1 **nie będzie** naciśnięty.

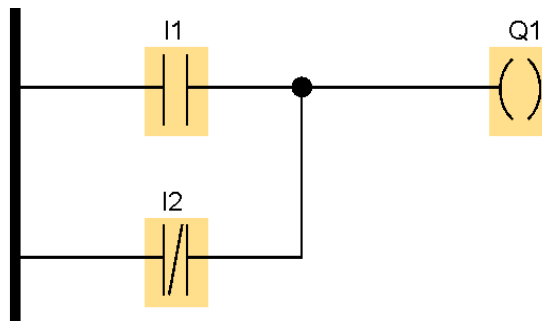
5. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejście I1 lub I2 będzie aktywne ($I1=1$ lub $I2=1$):



Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 **nie będzie** naciśnięty **lub** przycisk S1 będzie naciśnięty.

Równoległe połączenie symboli wejść w programie oznacza realizację operacji logicznej OR (lub).

6. Wyjście Q1 będzie aktywne ($Q1=1$) tylko wtedy, gdy wejście I1 będzie aktywne ($I1=1$) lub wejście I2 będzie nieaktywne ($I2=0$):

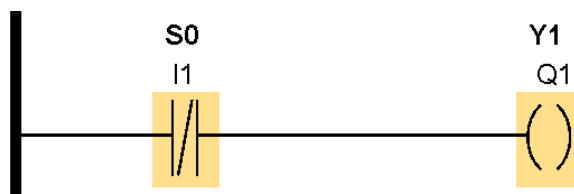


Dla układu ze schematu oznacza to, że cewka Y1 będzie zasilana, gdy przycisk S0 **nie będzie** naciśnięty lub przycisk S1 **nie będzie** naciśnięty.

W praktyce częściej mamy jednak do czynienia z sytuacją, w której wiemy jak ma działać dany układ sterowania, a naszym zadaniem jest napisanie programu realizującego opisane działanie. Dlatego w kolejnych przykładach najpierw opisano działanie układu jakie chcemy uzyskać, a następnie pokazano program realizujący to działanie. Opisy dotyczą układu z zamieszczonego wcześniej schematu.

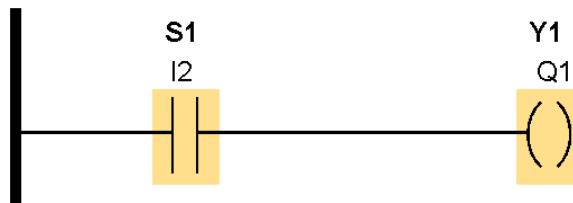
7. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S0 będzie naciśnięty.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S0 (do wejścia I1) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 tylko z wejściem I1, ponieważ stan cewki Y1 ma zależeć tylko od stanu przycisku S0. Na koniec pozostaje tylko określić rodzaj zestyku programowego dla wejścia I1. Spełniamy więc warunek zasilania cewki zawarty w treści zadania, czyli naciskamy przycisk S0 i sprawdzamy na schemacie układu stan wejścia I1 w momencie, gdy S0 jest naciśnięty. Naciśnięcie S0 spowoduje otwarcie zestyku NC przycisku S0 i odcięcie wejścia I1 od zasilania +24V. Oznacza to, że stan wejścia $I1=0$ (wejście niezasilone, nieaktywne). W takim przypadku, tzn. gdy $I1=0$ w programie umieszczamy symbol zestyku NC dla wejścia I1. W przeciwnym przypadku, tzn. gdyby po naciśnięciu S0 stan wejścia $I1=1$, należałoby wstawić w programie symbol zestyku NO dla wejścia I1.



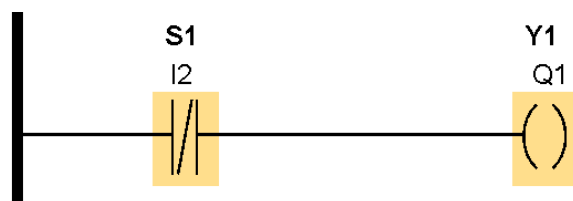
8. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S1 będzie naciśnięty.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S1 (do wejścia I2) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 tylko z wejściem I2, ponieważ stan cewki Y1 ma zależeć tylko od stanu przycisku S1. Na koniec pozostaje tylko określić rodzaj zestyku programowego dla wejścia I2. Spełniamy więc warunek zasilenia cewki zawarty w treści zadania, czyli naciskamy przycisk S1 i sprawdzamy na schemacie układu stan wejścia I2 w momencie, gdy S1 jest naciśnięty. Naciśnięcie S1 spowoduje zamknięcie zestyku NO przycisku S1 i dołączenie wejścia I2 od zasilania +24V. Oznacza to, że stan wejścia I2=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I2=1 w programie umieszczamy symbol zestyku NO dla wejścia I2. W przeciwnym przypadku, tzn. gdyby po naciśnięciu S1 stan wejścia I2=0, należałoby wstawić w programie symbol zestyku NC dla wejścia I2.



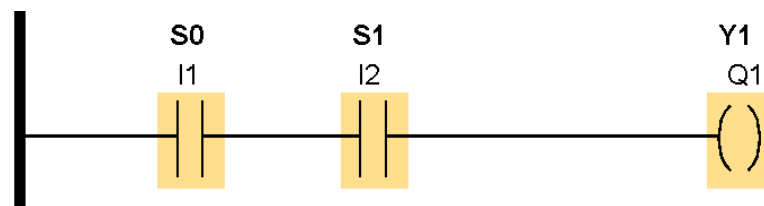
9. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S1 nie będzie naciśnięty.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S1 (do wejścia I2) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 tylko z wejściem I2, ponieważ stan cewki Y1 ma zależeć tylko od stanu przycisku S1. Na koniec pozostaje tylko określić rodzaj zestyku programowego dla wejścia I2. Spełniamy więc warunek zasilenia cewki zawarty w treści zadania, czyli **nie naciskamy** przycisku S1 i sprawdzamy na schemacie układu stan wejścia I2 w momencie, gdy S1 nie jest naciśnięty. Gdy S1 nie jest naciśnięty, to zestyk NO przycisku S1 pozostaje otwarty i odcina wejście I2 od zasilania +24V. Oznacza to, że stan wejścia I1=0 (wejście niezasilone, nieaktywne). W takim przypadku, tzn. gdy I1=0 w programie umieszczamy symbol zestyku NC dla wejścia I1.



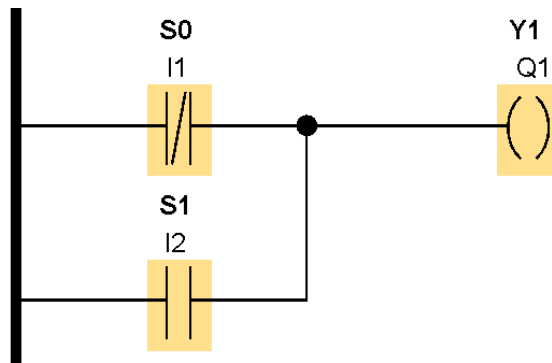
10. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S0 nie będzie naciśnięty i przycisk S1 będzie naciśnięty.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S0 (do wejścia I1), gdzie jest podłączony przycisk S1 (do wejścia I2) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 z wejściami I1 i I2, ponieważ stan cewki Y1 ma zależeć od stanu dwóch przycisków S0 i S1. Wejścia I1 i I2 muszą być przy tym połączone ze sobą **szeregowo**, ponieważ wyjście Q1 ma być aktywowane tylko w momencie, gdy przycisk S0 nie będzie naciśnięty i przycisk S1 będzie naciśnięty – program ma więc realizować funkcję logiczną **AND**. Na koniec pozostaje tylko określić rodzaj zestyków programowych dla wejść I1 i I2. Spełniamy więc warunek zasilenia cewki zawarty w treści zadania, czyli nie naciskamy przycisku S0, naciskamy przycisk S1 i sprawdzamy na schemacie układu stan wejść I1 i I2 w tym momencie. Gdy S0 nie jest naciśnięty, to zestyk NC przycisku S0 pozostaje zamknięty i dołącza wejście I1 od zasilania +24V. Oznacza to, że stan wejścia I1=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I1=1 w programie umieszczamy symbol zestyku NO dla wejścia I1. Natomiast naciśnięcie S1 spowoduje zamknięcie zestyku NO przycisku S1 i dołączenie wejścia I2 od zasilania +24V. Oznacza to, że stan wejścia I2=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I2=1 w programie umieszczamy symbol zestyku NO dla wejścia I2.



11. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S0 będzie naciśnięty lub przycisk S1 będzie naciśnięty.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S0 (do wejścia I1), gdzie jest podłączony przycisk S1 (do wejścia I2) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 z wejściami I1 i I2, ponieważ stan cewki Y1 ma zależeć od stanu dwóch przycisków S0 i S1. Wejścia I1 i I2 muszą być przy tym połączone ze sobą **równolegle**, ponieważ wyjście Q1 ma być aktywowane tylko w momencie naciśnięcia przycisku S0 **lub** S1 – program ma więc realizować funkcję logiczną **OR**. Na koniec pozostaje tylko określić rodzaj zestyków programowych dla wejść I1 i I2. Spełniamy więc warunek zasilenia cewki zawarty w treści zadania, czyli naciskamy przycisk S0, naciskamy przycisk S1 i sprawdzamy na schemacie układu stan wejść I1 i I2 w tym momencie. Naciśnięcie S0 spowoduje otwarcie zestyku NC przycisku S0 i odcięcie wejścia I1 od zasilania +24V. Oznacza to, że stan wejścia I1=0 (wejście niezasilone, nieaktywne). W takim przypadku, tzn. gdy I1=0 w programie umieszczamy symbol zestyku NC dla wejścia I1. Natomiast naciśnięcie S1 spowoduje zamknięcie zestyku NO przycisku S1 i dołączenie wejścia I2 od zasilania +24V. Oznacza to, że stan wejścia I2=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I2=1 w programie umieszczamy symbol zestyku NO dla wejścia I2.



12. Cewka Y1 będzie zasilana tylko wtedy, gdy przycisk S0 lub przycisk S1 będzie naciśnięty i dodatkowo czujnik B1 będzie aktywowany.

Rozwiązanie: Sprawdzamy, gdzie jest podłączony przycisk S0 (do wejścia I1), gdzie jest podłączony przycisk S1 (do wejścia I2), gdzie jest podłączony czujnik B1 (do wejścia I4) i gdzie jest podłączona cewka Y1 (do wyjścia Q1). W programie łączymy zatem wyjście Q1 z wejściami I1, I2 i I3, ponieważ stan cewki Y1 ma zależeć od stanu dwóch przycisków S0 i S1 oraz czujnika B1. Wejścia I1 i I2 muszą być przy tym połączone ze sobą **równolegle**, ponieważ wyjście Q1 ma być aktywowane w momencie naciśnięcia przycisku S0 **lub** S1 – program ma więc realizować funkcję logiczną **OR** w odniesieniu do przycisków S0 i S1. Natomiast wejście I4 musi być połączone szeregowo względem wejść I1 i I2, ponieważ wyjście Q1 ma być aktywowane w momencie naciśnięcia jednego z przycisków i aktywacji czujnika B1 – program ma więc realizować funkcję logiczną **AND** w odniesieniu do stanu przycisków i stanu czujnika. Na koniec pozostaje tylko określić rodzaj zestyków programowych dla wejść I1, I2 i I4. Spełniamy więc kolejne warunki zasilania cewki zawarte w treści zadania i sprawdzamy na schemacie układu stany wejść I1, I2 i I4 w tych momentach. Naciśnięcie S0 spowoduje otwarcie zestyku NC przycisku S0 i odcięcie wejścia I1 od zasilania +24V. Oznacza to, że stan wejścia I1=0 (wejście niezasilone, nieaktywne). W takim przypadku, tzn. gdy I1=0 w programie umieszczamy symbol zestyku NC dla wejścia I1. Naciśnięcie S1 spowoduje zamknięcie zestyku NO przycisku S1 i dołączenie wejścia I2 od zasilania +24V. Oznacza to, że stan wejścia I2=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I2=1 w programie umieszczamy symbol zestyku NO dla wejścia I2. Aktywacja czujnika B1 spowoduje zamknięcie zestyku NO czujnika B1 i dołączenie wejścia I4 od zasilania +24V. Oznacza to, że stan wejścia I4=1 (wejście zasilone, aktywne). W takim przypadku, tzn. gdy I4=1 w programie umieszczamy symbol zestyku NO dla wejścia I4.

