

# **STEROWANIE MASZYN I URZĄDZEŃ I**

## **Laboratorium**

### **2. Układy kombinacyjne**

Opracował: dr hab. inż. Cezary Orlikowski

Instytut Politechniczny



**PWSZ**  
w ELBLĄGU

W tym ćwiczeniu będą realizowane programy sterujące zawierające tylko elementy z grup *CONTACTS* i *COILS* czyli styki i przekaźniki. Przy pomocy tych elementów można budować układy kombinacyjne i sekwencyjne bez uzależnień czasowych. Przy czym w tym ćwiczeniu będą realizowane tylko układy kombinacyjne.

#### STYKI I PRZEKAŹNIKI PROGRAMU *CIMPLICITY*

Następujące typy styków dostępnych w programie *CIMPLICITY* będą używane do budowy programów sterujących:

Styk otwarty – | | –

Styk taki działa jak wyłącznik, który przewodzi sygnał (zwiera styki), gdy wartość logiczna przypisanej mu zmiennej wynosi 1.

Styk zamknięty – | / | –

Styk taki działa jak wyłącznik, którego styki pozostają zwarte (przewodzi sygnał), gdy wartość logiczna przypisanej mu zmiennej wynosi 0.

Typ przekaźnika zależy od celu jego zastosowania. Jeśli stan przekaźnika ma być zachowany po wyłączeniu sterownika należy zastosować przekaźnik z pamięcią. Przekaźniki bez pamięci w takiej sytuacji zostają wyzerowane. Następujące typy przekaźników dostępnych w programie *CIMPLICITY* będą używane do budowy programów sterujących:

Przekaźnik o stykach otwartych – ( ) –

Przekaźnik taki ustawia wartość przypisanej zmiennej na jeden, gdy dopłynie do niego sygnał (równy 1). Jest to przekaźnik bez pamięci.

Przekaźnik o stykach zamkniętych – ( / ) –

Przekaźnik taki ustawia wartość przypisanej zmiennej na zero, gdy dopłynie do niego sygnał. Jest to przekaźnik bez pamięci.

Przekaźnik o stykach otwartych , z pamięcią – (M) –

Działa podobnie jak zwykły przekaźnik o stykach otwartych, lecz stan przekaźnika zostaje podtrzymany w przypadku zaniku zasilania sterownika.

Przekaźnik o stykach zamkniętych , z pamięcią – (/M) –

Działa podobnie jak zwykły przekaźnik o stykach zamkniętych. Stan przekaźnika zostaje podtrzymany w przypadku zaniku zasilania sterownika.

Działanie styków i przekaźników ilustrują wykresy czasowe pokazane na rys. 1.

#### PRZYKŁADY PROGRAMÓW STERUJĄCYCH

Poniżej podano przykłady programów sterujących wykorzystujących styki i przekaźniki opisane powyżej.

##### Przykład 1

##### *Zadanie*

Należy zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów *WE1*, *WE2*, *WE3* (rys. 2a). Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem *WY1*. Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2. Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem *WY3*.

##### *Rozwiązanie*

Na podstawie opisu słownego problemu można sformułować następujące funkcje logiczne opisujące działanie układu:

$$\begin{aligned}
 WY1 &= \overline{WE1} \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot \overline{WE3} \\
 WY2 &= \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3} \\
 WY3 &= WE1 \cdot WE2 \cdot WE3
 \end{aligned}$$

Odpowiadający im schemat drabinkowy pokazano na rys. 2b.

### Przykład 2

#### Zadanie

Zaprojektować układ automatycznego sterowania dwóch linii zasilających zbiornik (rys. 3). Zbiornik jest opróżniany nierównomiernie, ale zawsze napełniany za pomocą dwóch linii (zaworów). Jeżeli zbiornik jest napełniony mniej niż w połowie, to obydwa zawory napełnią zbiornik. Jeżeli zbiornik nie jest pełny ale jego napełnienie przekracza połowę, to otwarty jest tylko zawór pierwszy. Jeżeli zbiornik jest pełny, to obydwa zawory są zamknięte.

#### Rozwiązanie

Ze słownego sformułowania warunków pracy układu wynika, że zbiornik może znajdować się w jednym z trzech stanów:

1. od stanu „pusty” do „zapełniony w połowie”
2. od stanu „zapełniony w połowie” do „pełny” lecz nie „pełny”
3. w stanie „pełny”

Do opisanie tych stanów potrzebne są dwie zmienne binarne, co pokazano w tabl. 1.

Tablica do przykładu 2 zbiornik	<b>Tablica 1</b>	
	<i>D</i>	<i>G</i>
od stanu „pusty” do „zapełniony w połowie”	0	0
od stanu „zapełniony w połowie” do „pełny” lecz (nie „pełny”)	1	0
„pełny”	1	1

Zmienne *D*, *G* są sygnałami wejściowymi szukanego układu automatycznego sterowania i pochodzą z czujników umieszczonych w zbiorniku i kontrolujących poziom środkowy (*DS*) i górny (*G*). Z powyższej tablicy wynika, że przyjęto iż sygnał z każdego czujnika poziomu wynosi 0, gdy poziom wody jest niższy od poziomu przez ten czujnik kontrolowanego, a wynosi 1, gdy poziom wody jest wyższy od poziomu kontrolowanego. Oczywiście, można przyjąć inaczej i wówczas tablica będzie miała inną postać.

Stany, w których może znajdować się każda z pomp określone są przez zmienne  $Z_1$  (pierwszy zawór) i  $Z_2$  (drugi zawór). Założymy, że  $Z_1=Z_2=0$  oznacza, że pompy nie pracują, a  $Z_1=Z_2=1$  – pompy pracują. Po tych wstępnych ustaleniach możemy sporządzić tablicę (tabl. 2) funkcji logicznej realizowanej przez projektowany układ.

**Tablica 2**

Działanie układu sterowania (przykład 2)

D	G	Z <sub>1</sub>	Z <sub>2</sub>
0	0	1	1
0	1	nie występuje	
1	0	1	0
1	1	0	0

Stan  $D=0, G=1$  nie występuje w poprawnie działającym układzie, zatem wartości  $Z_1$  oraz  $Z_2$  dla tego stanu możemy wybrać dowolne – najlepiej takie, które ułatwią dalsze działania.

Ponieważ w naszym wypadku posługiwać się będziemy postacią *ZNPD* funkcji logicznych, wygodnie jest przyjąć te wartości równe zero (tabl. 3).

**Tablica 3**

Tablica funkcji logicznych (przykład 2)

D	G	Z <sub>1</sub>	Z <sub>2</sub>
0	0	1	1
0	1	0	0
1	0	1	0
1	1	0	0

Projektowany układ sterowania posiada więc dwa wejścia  $D$  i  $G$  oraz dwa wyjścia  $Z_1, Z_2$ . Realizuje więc dwie funkcje logiczne  $Z_1(D,G)$  oraz  $Z_2(D,G)$  określone ostatnią tablicą, z której otrzymujemy:

$$Z_1 = \bar{G}$$

$$Z_2 = \bar{D}\bar{G},$$

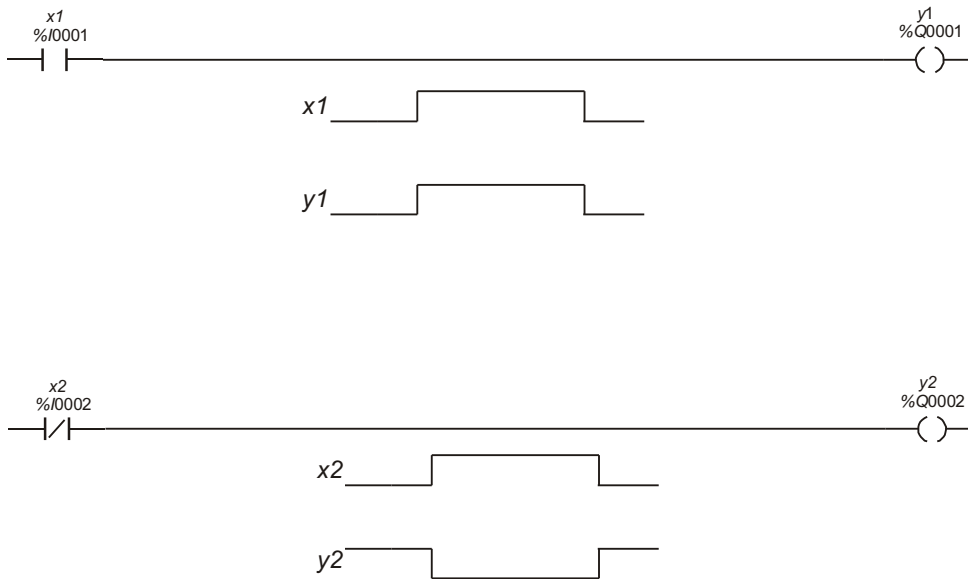
a schemat układu realizującego obydwie funkcje jest pokazany na rys. 4.

## CEL I PRZEBIEG ZAJĘĆ

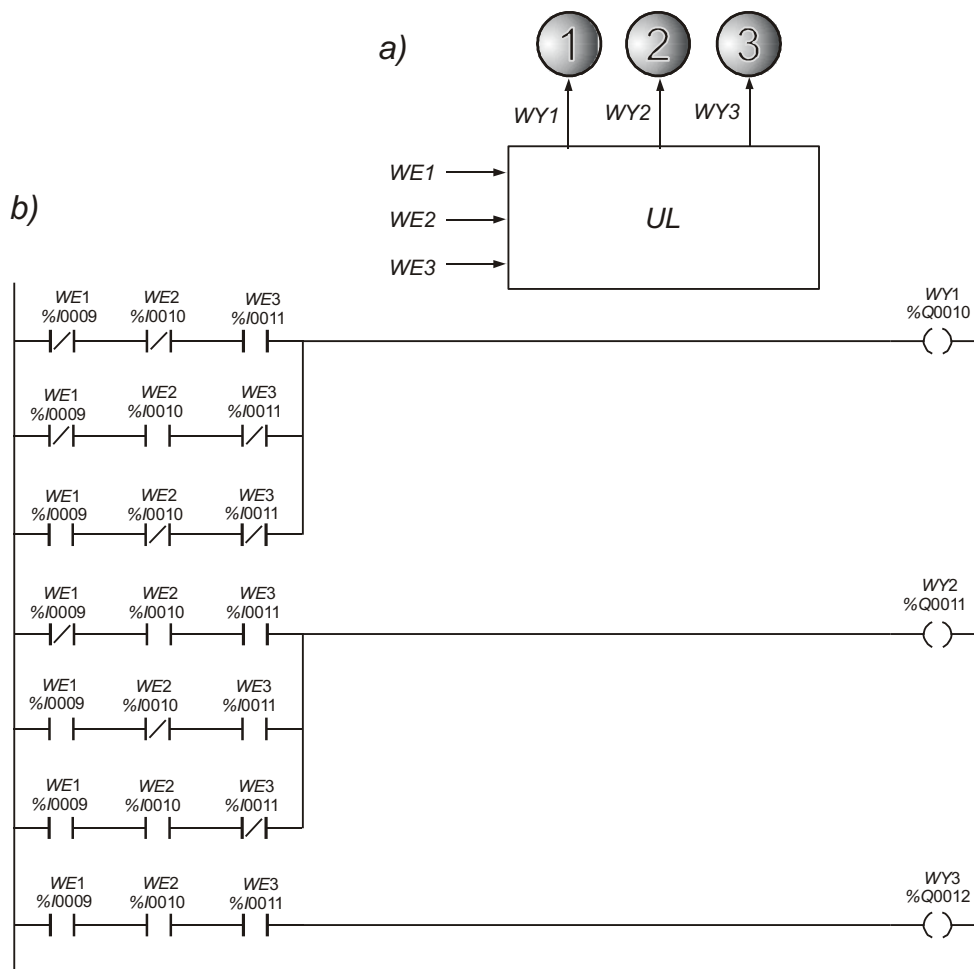
Cel: Programowanie sterownika *GE FANUC* do realizacji układów sterowania kombinacyjnych.

### Przebieg

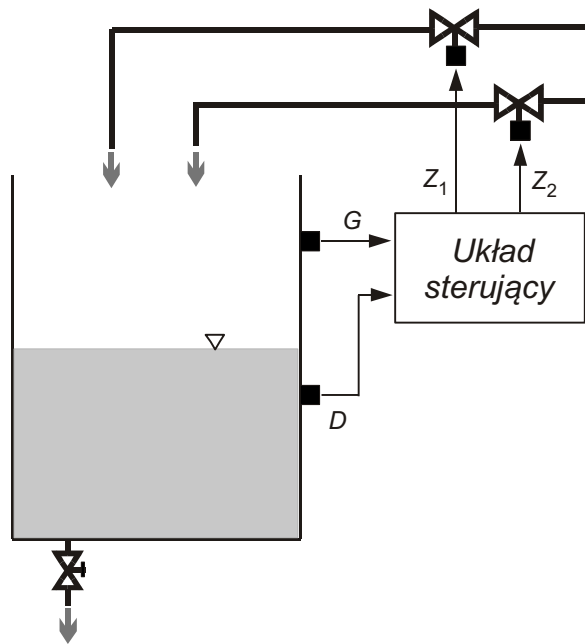
1. Zapoznać się z przykładami programów sterujących przedstawionych we wprowadzeniu do ćwiczenia.
  - przeanalizować rozwiązanie zadania 1 oraz sporządzić tabele funkcji logicznych
  - przeanalizować rozwiązanie zadania 2.
2. Zaprogramować sterownik do realizacji programów sterujących przedstawionych we wprowadzeniu do ćwiczenia. Sprawdzić ich działanie.
3. Zmodyfikować powyższe programy według wskazań prowadzącego.
4. Zbudować nowe programy sterujące zgodnie z poleceniem prowadzącego ćwiczenia. Sprawdzić ich działanie.



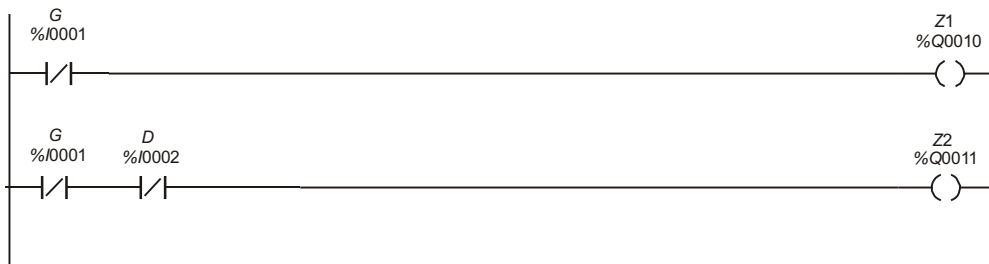
Rys. 1. Najprostsze układy stykowo - przekaźnikowe i ich wykresy czasowe



Rys. 2. Przykład 1: a - schemat układu, b - układ sterujący



Rys. 3. Schemat obiektu do przykładu 2



Rys. 4. Układ realizujący funkcje  $Z_1$  i  $Z_2$