

7. Sterowniki PLC – język FBD

Język *funkcjonalnych schematów blokowych FBD* jest wzorowany na schematach logicznych (zob. p. 2.3) i pozwala na zapis programu w postaci bloków operacji powiązanych za pomocą linii przepływu sygnałów. Oprócz bloków operacji logicznych w języku można wykorzystywać również standardowe i zdefiniowane przez użytkownika funkcje i bloki funkcjonalne.

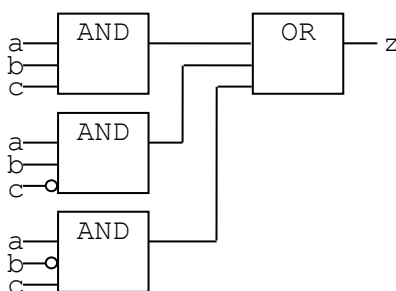
7.1. Opis języka

Program w języku *FBD* jest dzielony na obwody (ang. network). Każdy obwód to sieć prostokątów reprezentujących funkcje i bloki funkcjonalne, połączonych liniami poziomymi i pionowymi reprezentującymi kolejność przepływu sygnału. Przepływ sygnału odbywa się z wyjścia bloku poprzedzającego (lewa strona) do wejścia bloku kolejnego (prawa strona). Podczas wykonania programu napisanego w języku *FBD* obowiązują następujące zasady:

- wartość wyjściowa elementu obwodu wyznaczana jest dopiero po wyznaczeniu wszystkich wejść tego elementu,
- przetwarzanie elementu obwodu jest zakończone po wyznaczeniu wartości wszystkich jego wyjść,
- przetwarzanie obwodu jest zakończone po wyznaczeniu wyjść wszystkich jego elementów,
- porządek wykonywania obwodów zależy od producenta sterownika (może np. odbywać się w kierunku góra-dół).

Dodatkowo z punktu 6.3.1 warto przypomnieć, że w języku *FBD* funkcje i bloki funkcjonalne mogą mieć negowane wejścia lub wyjścia w sposób przedstawiony na rys. 6.2, co pozwala na wyeliminowanie konieczności użycia funkcji NOT.

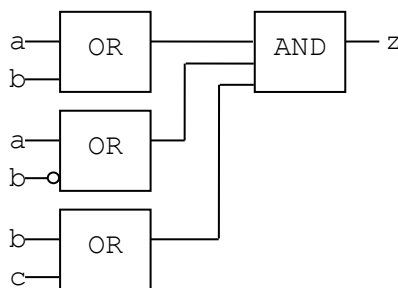
Na rys. 7.1 został przedstawiony obwód realizujący funkcję logiczną zapisaną w postaci dysjunkcyjnej: najpierw wyznaczane są wartości bloków realizujących koniunkcje, po wyznaczeniu wartości wszystkich koniunkcji wyznaczana jest ich alternatywa. Wynik alternatywy zapisywany jest w zmiennej *z*.



Rys.7.1. Obwód języka FBD realizujący funkcję

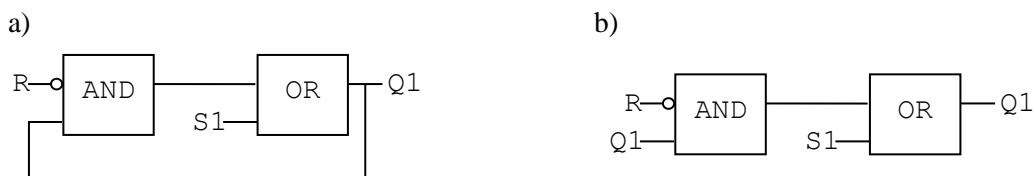
$$z = a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{c}$$

Rysunek 7.2 przedstawia tę samą funkcję, przekształconą do postaci koniunkcyjnej. W tym przypadku najpierw wyznaczane są wartości bloków realizujących alternatywy, a po ich wyznaczeniu obliczana jest ich koniunkcja. Wynik koniunkcji przypisywany jest zmiennej *z*.



Rys.7.2. Obwód języka FBD realizujący funkcję
 $z = (a+b) \cdot (a+\bar{b}) \cdot (b+c)$

W obwodach języka FBD mogą występować sprzężenia zwrotne (lub inaczej pętle sprzężenia zwrotnego). Sprzężeniem zwrotnym nazywane jest wprowadzenie sygnału z wyjścia bloku na wejście bloku występującego przed nim. Ze sprzężeniem zwrotnym związana jest zmienna, która może być wykorzystana do niejawnego odwzorowania takiego sprzężenia. Rysunek 7.3 przedstawia dwie wersje realizacji przerzutnika SR z dominującym wejściem ustawiającym w języku FBD. Na rys. 7.3a obwód został narysowany z jawną pętlą sprzężenia zwrotnego, a na rys. 7.3b pętla została ukryta z pomocą zmiennej tego sprzężenia.



Rys.7.3. Sprzężenie zwrotne w wyrażeniu $Q1 = \bar{R} \cdot Q1 + S1$
 a) pętla jawna, b) pętla ukryta

Rysując program w języku FBD, należy dbać o jego czytelność, starając się, żeby w rysowanych obwodach występowało jak najmniej linii przecinających się. W następnych punktach omówione zostały przykładowe programy, w których oprócz funkcji boolowskich wykorzystane zostały standardowe bloki funkcjonalne: przerzutniki, detektory zbrocza, liczniki i timery.

7.2. Programy z przerzutnikami

W punkcie 4.5.2 została omówiona metoda pozwalająca na zastosowanie przerzutników SR do realizacji bloku pamięci układu sekwencyjnego. W najprostszym przypadku, kiedy blok pamięci układu zbudowany jest tylko z jednego elementu pamięci, zastosowanie metody sprowadza się właściwie do ustalenia warunków ustawiającego i zerującego ten element. Warunki te są następnie wykorzystywane jako funkcje wzbudzeń przerzutnika SR.

7.2.1. Przykład 1a

Na początek rozważone zostanie zadanie sterowania pracą urządzenia za pomocą dwóch łączników bistabilnych rozważana we wcześniejszych przykładach jako układ włącz-wyłącz II (zob. p. 4.2.2, 4.4.1 i 4.5.2). Pierwszy łącznik odpowiada za włączanie urządzenia, a drugi za jego wyłączanie. Łącznik wyłączający ma wyższy priorytet, a więc przy włączonych obydwu łącznikach urządzenie powinno zostać wyłączone.

Zakładając, że zostały zadeklarowane trzy zmienne boolowskie:

- zmienna w odpowiadająca stanowi łącznika wykorzystywanego do włączania urządzenia,
- zmienna z odpowiadająca stanowi łącznika wykorzystywanego do wyłączania (zatrzymywania) urządzenia,
- zmienna u skojarzona z wyjściem sterownika wykorzystywanym do sterowania pracą urządzenia,

korzystając z przedstawionych warunków pracy urządzenia, można zapisać warunki dla wejścia ustawiającego i zerującego przerzutnika SR.

Uwzględniając fakt, że wyłączanie ma wyższy priorytet, warunek dla wejścia ustawiającego można słownie opisać jako: włączony łącznik pierwszy przy wyłączonym drugim, co w postaci wyrażenia można zapisać jako:

$$w \cdot \bar{z}.$$

Wyłączanie jest realizowane po włączeniu drugiego łącznika, niezależnie od stanu pierwszego łącznika, co można zapisać w postaci wyrażenia:

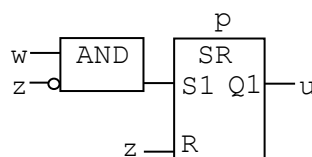
$$\bar{w} \cdot z + w \cdot z,$$

które, po zastosowaniu reguły sklejania, może być skrócone do postaci:

$$z.$$

Ostatecznie, sterowanie pracą urządzenia można zrealizować, wykorzystując przerzutnik SR z dominującym wejściem ustawiającym, tak jak pokazano to na rys. 7.4 (por. rys. 4.57).

```
PROGRAM Przyklad1a
VAR
  w AT %IX1.1: BOOL;
  z AT %IX1.2: BOOL;
  u AT %QX2.1: BOOL;
  p: SR;
END_VAR
```

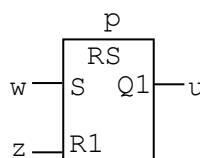


Rys.7.4. Sterowanie pracą urządzenia wersja 1

7.2.2. Przykład 1b

Zastosowanie przerzutnika SR z dominującym wejściem zerującym umożliwia dodatkowe uproszczenie warunku dla wejścia ustawiającego, z którego można wyeliminować sprawdzanie stanu łącznika zatrzymującego pracę urządzenia. Ze względu na dominację zerowania, urządzenie jest zatrzymywane po włączeniu drugiego łącznika, więc ustawianie przerzutnika może nastąpić jedynie, gdy drugi łącznik jest wyłączony. Ta wersja programu została przedstawiona na rys. 7.5.

```
PROGRAM Przyklad1b
VAR
  w AT %IX1.1: BOOL;
  z AT %IX1.2: BOOL;
  u AT %QX2.1: BOOL;
  p: RS;
END_VAR
```



Rys.7.5. Sterowanie pracą urządzenia wersja 2

7.2.3. Przykład 1c

W kolejnym przykładzie rozważone zostało sterowanie pracą dwóch urządzeń. Każde urządzenie posiada swój zestaw łączników bistabilnych używanych do włączania i wyłączania. Dodatkowo wprowadzone zostało ograniczenie niepozwalające na jednoczesną pracę obydwu urządzeń.

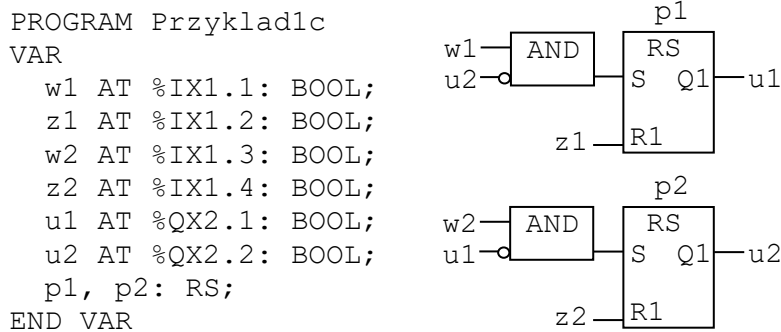
Do zrealizowania programu w tej wersji wykorzystane zostały dwa przerzutniki SR z dominującym wejściem zerującym. Zakładając, że stanom łączników sterujących pracą pierwszego urządzenia odpowiadają zmienne w_1 i z_1 , a stanom łączników drugiego urządzenia zmienne w_2 i z_2 , warunki na wejściach ustawiających przerzutników można zapisać podobnie jak w poprzednim przykładzie, trzeba jednak uwzględnić wprowadzone ograniczenie blokujące jednoczesną pracę urządzeń. W przypadku urządzenia pierwszego warunek na wejściu ustawiającym przerzutnika nie może pozwolić na jego włączenie, w przypadku gdy włączone jest drugie urządzenie:

$$w1 \cdot \overline{u2},$$

w przypadku drugiego urządzenia warunek nie może pozwolić na włączenie, gdy włączone jest urządzenie pierwsze:

$$w2 \cdot \overline{u1}.$$

Warunki na wejściach zerujących konstruowane są na podstawie stanów łączników zatrzymujących urządzenia. Program realizujący tak postawione zadanie został przedstawiony na rys. 7.6.



Rys.7.6. Sterowanie pracą dwóch urządzeń wersja 1

Zastosowanie przerzutników *SR* z *dominującym wejściem zerującym* pozwala tak jak poprzednio na uproszczenie warunków dla wejść ustawiających obydwu przerzutników, w których nie jest konieczne sprawdzanie stanu łączników wyłączających. W przypadku zastosowania *przerzutników z dominującym wejściem ustawiającym* warunki te należałoby zapisać jako: $w1 \cdot z1 \cdot \overline{u2}$ dla pierwszego przerzutnika i $w2 \cdot z2 \cdot \overline{u1}$ dla drugiego przerzutnika.

7.3. Programy z detekcją zbrocza

7.3.1. Przykład 2a

W przypadku sterowania pracą urządzenia w sposób przedstawiony na rys. 7.4 czy 7.5, jeżeli po włączeniu obydwu łączników zwolniony zostanie łącznik wyłączający, to urządzenie natychmiast zostanie włączone. W kolejnej wersji programu uwzględnione zostanie dodatkowe ograniczenie blokujące włączanie urządzenia w takim przypadku. Zgodnie z przyjętymi założeniami, w celu zwiększenia bezpieczeństwa pracy, do włączenia urządzenia niezbędne są dodatkowe działania operatora, który w takiej sytuacji powinien wyłączyć i ponownie włączyć pierwszy łącznik. W ten sposób urządzenie zostaje zabezpieczone przed przypadkowym włączeniem przez zwolnienie łącznika wyłączającego.

Wprowadzone zmiany wymagają zmodyfikowania warunku dla wejścia ustawiającego przerzutnika, który powinien pozwalać na pracę urządzenia, tylko gdy łącznik włączający zmieni swój stan z *wyłączonego* na *włączony*. W ten sposób, jeżeli obydwa łączniki były włączone i urządzenie nie pracowało, a operator zwolnił łącznik wyłączający to włączenie urządzenia nastąpi, tylko gdy operator wyłączy i ponownie włączy łącznik pierwszy – zmieni on wtedy swój stan z *wyłączonego* na *włączony*.

Zakładając, że zadanie zostanie zrealizowane z pomocą *przerzutnika SR z dominującym wejściem zerującym*, warunek na wejściu ustawiającym przerzutnika nie musi uwzględniać stanu łącznika wyłączającego (zob. rys. 7.5), a jedynie zmianę wartości z 0 na 1 zmiennej *w*, co może być symbolicznie zapisane jako:

$$\uparrow w.$$

Zmianę taką można wykryć z pomocą *detektora zbrocza narastającego* (zob. p. 6.4.2). Program uwzględniający wprowadzoną modyfikację został przedstawiony na rys. 7.7.

PROGRAM Przyklad2a

VAR

w AT %IX1.1: BOOL;

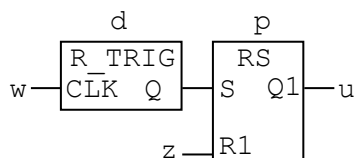
z AT %IX1.2: BOOL;

u AT %QX2.1: BOOL;

p: RS;

d: R_TRIG;

END_VAR



Rys. 7.7. Sterowanie pracą urządzenia wersja 3

7.3.2. Przykład 2b

W kolejnym przykładzie omówione zostanie wykorzystanie detekcji zbocza narastającego do sterowania pracą dwóch urządzeń. Teraz, w odróżnieniu od rozwiązania przedstawionego na rys. 7.6, założono, że sterowanie odbywa się za pomocą jednego zestawu łączników wykorzystywanych do włączania i wyłączania obydwu urządzeń. Pierwsze włączenie łącznika włączającego uruchamia urządzenie pierwsze, po wyłączeniu i ponownym włączeniu łącznika urządzenie pierwsze jest zatrzymywane, a uruchamiane jest urządzenie drugie, kolejne wyłączenie i włączenie łącznika zatrzymuje urządzenie drugie i uruchamia pierwsze itd.

Podobnie jak w przykładzie 1c, wykorzystano tu dwa przerzutniki SR z dominującym wejściem zerującym. Warunki na wejściach ustawiających przerzutników blokują włączanie, gdy drugie z urządzeń jest uruchomione (tab. 7.1). Warunki na wejściach zerujących pozwalają na wyłączenie urządzenia albo po włączeniu łącznika wyłączającego, albo po wyłączeniu i ponownym włączeniu łącznika włączającego (pod warunkiem, że urządzenie pracuje).

Tab. 7.1. Warunki w przykładzie 2b

	wejście ustawiające	wejście zerujące
przerzutnik 1	$w \cdot \bar{u}2$	$z + \uparrow w \cdot u1$
przerzutnik 2	$w \cdot \bar{u}1$	$z + \uparrow w \cdot u2$

Ze względu na konieczność detekcji zbocza $\uparrow w$ w dwóch warunkach, w celu skrócenia zapisu, wprowadzona została pomocnicza zmienna wn , która dostaje wartość 1 w przypadku wykrycia zbocza narastającego sygnału w . Program uwzględniający nowe warunki pracy urządzeń został przedstawiony na rys. 7.8.

PROGRAM Przyklad2b

VAR

w AT %IX1.1: BOOL;

z AT %IX1.2: BOOL;

u1 AT %QX2.1: BOOL;

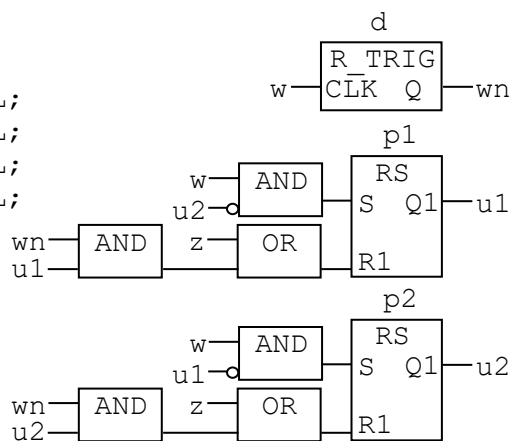
u2 AT %QX2.2: BOOL;

p1, p2: RS;

d: R_TRIG;

wn: BOOL;

END_VAR



Rys. 7.8. Sterowanie pracą dwóch urządzeń wersja 2

7.4. Program z licznikiem

7.4.1. Przykład 3

Liczniki umożliwiają zliczanie zboczy narastających. W przedstawionym poniżej zastosowaniu licznik umożliwił wprowadzenie dodatkowej blokady włączania urządzenia. Założono, że urządzenie sterowane programem z rys. 7.5 może być uruchomione maksymalnie 3 razy, a kolejne próby uruchomienia kończą się niepowodzeniem.

Uwzględnienie takiego ograniczenia wymaga wprowadzenia zmian do warunku ustawiającego przerzutnik, który oprócz sprawdzenia wymagania podstawowego (włączenie łącznika włączającego), powinien dodatkowo sprawdzać czy liczba uruchomień nie przekracza założonych trzech. Do zliczania wykorzystany został *licznik zliczający w górę* (zob. p. 6.4.3). Licznik ten jest włączany (na jego wyjściu pojawia się wartość 1) po wykryciu zadanej liczby zboczy narastających. Sygnał wyjściowy licznika w warunku ustawiającym przerzutnika musi być więc negowany, ponieważ urządzenie jest włączane, tylko gdy liczba uruchomień nie przekracza założonej. Program uwzględniający wprowadzone ograniczenie został przedstawiony na rys. 7.9.

```
PROGRAM Przyklad3
```

```
VAR
```

```
  w AT %IX1.1: BOOL;
```

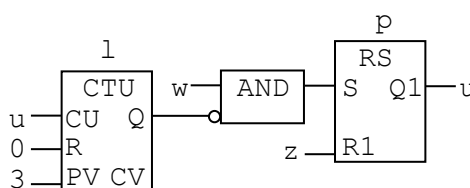
```
  z AT %IX1.2: BOOL;
```

```
  u AT %QX2.1: BOOL;
```

```
  p: RS;
```

```
  l: CTU;
```

```
END_VAR
```



Rys.7.9. Sterowanie pracą urządzenia wersja 4

7.5. Programy z timerami

7.5.1. Przykład 4a

W kolejnej wersji programu sterującego pracą urządzenia obsługiwanego z pomocą dwóch łączników bistabilnych założono, że urządzenie dodatkowo powinno być automatycznie wyłączane po 5 sekundach od uruchomienia (może być wyłączone wcześniej łącznikiem wyłączającym).

W celu uwzględnienia takiego ograniczenia konieczna jest modyfikacja warunku ustawianego na wejściu zerującym przerzutnika. W warunku tym, oprócz sprawdzenia podstawowego wymagania (włączenie łącznika wyłączającego), należy także sprawdzać czy nie został przekroczony maksymalny czas działania urządzenia. Do mierzenia czasu, który upłynął od momentu uruchomienia urządzenia wykorzystany został *timer TON* (zob. p. 6.4.4). Na wejście IN timera wprowadzony jest sygnał określający aktualny stan urządzenia. Wyjście Q timera jest włączane po upływie czasu zadanego na wejściu PT, sygnał o wartości 1 na tym wyjściu jest jednym z dwóch powodów (oprócz stanu łącznika wyłączającego) wyłączania urządzenia. Program z rys. 7.5 ze zmodyfikowanym warunkiem wyłączającym został pokazany na rys. 7.10.

```
PROGRAM Przyklad4a
```

```
VAR
```

```
  w AT %IX1.1: BOOL;
```

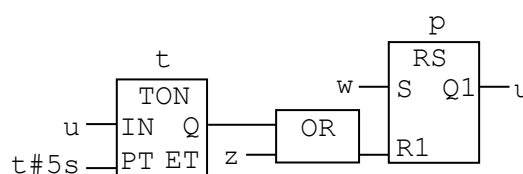
```
  z AT %IX1.2: BOOL;
```

```
  u AT %QX2.1: BOOL;
```

```
  p: RS;
```

```
  t: TON;
```

```
END_VAR
```

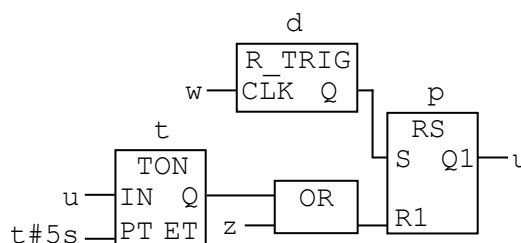


Rys.7.10. Sterowanie pracą urządzenia wersja 5

7.5.2. Przykład 4b

Program z poprzedniego przykładu działa zgodnie z przyjętymi założeniami jednak wyłączenie urządzenia po upływie założonego czasu, w sytuacji gdy włączony jest łącznik włączający zajmuje tylko jeden cykl pracy sterownika. W kolejnym cyklu urządzenie jest ponownie włączane. W celu rozwiązania tego problemu można zastosować podejście przedstawione w punkcie 7.3 i wykorzystać zbocze narastające sygnału odpowiadającego stanowi łącznika włączającego. Po wyłączeniu urządzenia niezbędne będzie w takim przypadku włączenie łącznika (jeśli łącznik był włączony, konieczne będzie jego wyłączenie i ponowne włączenie). Program uwzględniający wprowadzoną modyfikację został przedstawiony na rys. 7.11.

```
PROGRAM Przyklad4b
VAR
  w AT %IX1.1: BOOL;
  z AT %IX1.2: BOOL;
  u AT %QX2.1: BOOL;
  p: RS; t: TON;
  d: R_TRIG;
END_VAR
```



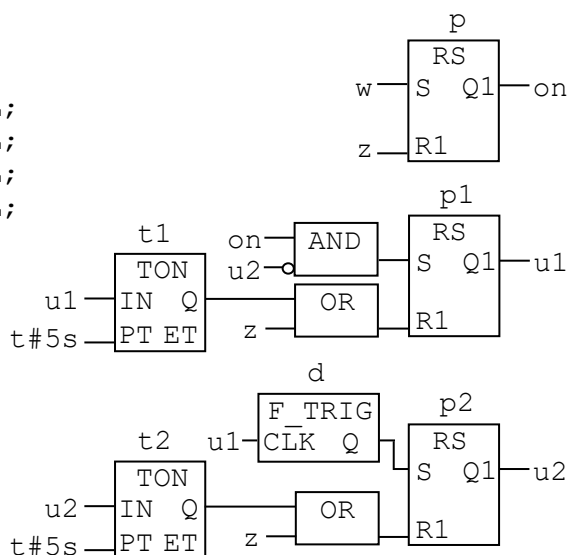
Rys.7.11. Sterowanie pracą urządzenia wersja 6

7.5.3. Przykład 4c

W kolejnym zadaniu ponownie rozważono sterowanie pracą dwóch urządzeń. Podobnie jak w rozwiązaniu przedstawionym na rys. 7.8, założono, że praca układu zależy od stanu łącznika włączającego i wyłączającego. Łącznik włączający uruchamia pierwsze urządzenie, po 5 sekundach urządzenie jest wyłączane i uruchamiane jest urządzenie drugie, po kolejnych 5 sekundach urządzenie drugie jest wyłączane i uruchamiane jest ponownie pierwsze urządzenie. Wyłączanie i włączanie urządzeń trwa do momentu włączenia łącznika wyłączającego, który zatrzymuje cały proces.

Do kontroli czasu aktywności obydwu urządzeń wykorzystane zostały, podobnie jak w przykładzie poprzednim, *timery TON* (zob. p. 6.4.4). Warunek włączający drugie urządzenie został napisany z wykorzystaniem *detektora zbocza opadającego* – urządzenie drugie jest uruchamiane po wyłączeniu urządzenia pierwszego. Dodatkowo w celu uproszczenia postaci warunku włączającego urządzenie pierwsze wykorzystany został dodatkowy *przerzutnik SR z dominującym wejściem zerującym*, którego zadaniem jest ustalenie czy aktualny stan łączników pozwala na działanie całego układu.

```
PROGRAM Przyklad4c
VAR
  w AT %IX1.1: BOOL;
  z AT %IX1.2: BOOL;
  u1 AT %QX2.1: BOOL;
  u2 AT %QX2.2: BOOL;
  on: BOOL;
  p: RS;
  p1: RS;
  p2: RS;
  t1: TON;
  t2: TON;
  d: F_TRIG;
END_VAR
```

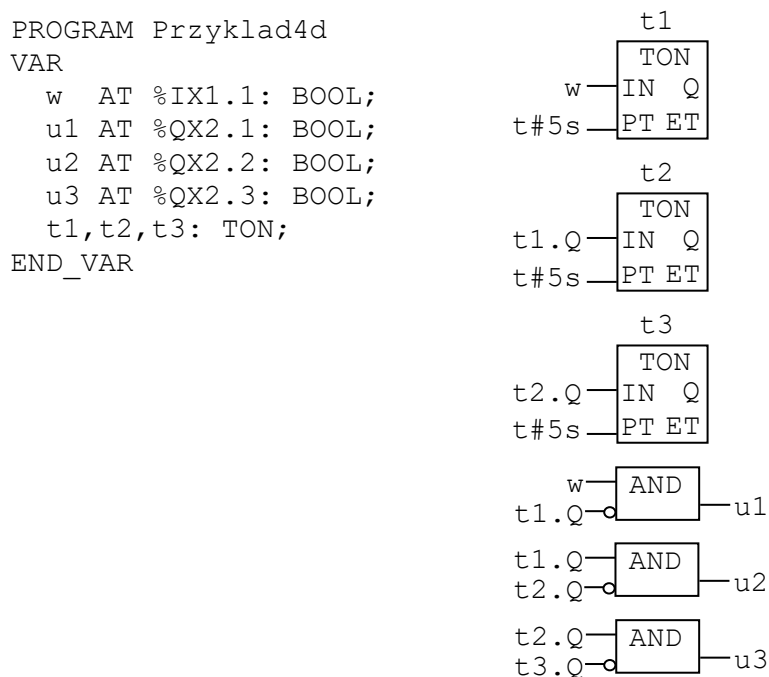


Rys.7.12. Sterowanie pracą dwóch urządzeń wersja 3

Przedstawione rozwiązanie nie jest jedynym możliwym. Zadanie kaskadowego włączania urządzeń można alternatywnie rozwiązać bez wykorzystywania detektora zbrocza. Zanim jednak przedstawione zostanie takie rozwiązanie, rozważone zostaną jeszcze dwa prostsze zadania z timerami.

7.5.4. Przykład 4d

W przykładzie omówione zostanie wykorzystanie *timerów* do sterowania pracą trzech urządzeń. Pierwsze urządzenie powinno zostać włączone po włączeniu łącznika bistabilnego włączającego pracę całego układu. Urządzenie pierwsze jest wyłączane po upływie 5 sekund i automatycznie jest uruchamiane urządzenie drugie. Po kolejnych 5 sekundach urządzenie drugie jest zatrzymywane, a uruchamiane jest urządzenie trzecie, które po upływie kolejnych 5 sekund jest wyłączane. Rozpoczęty proces włączania urządzeń może być przerwany po wyłączeniu łącznika, a kolejne uruchamianie urządzeń inicjowane jest po ponownym włączeniu tego łącznika (urządzenia zawsze uruchamiane są od pierwszego niezależnie od tego czy przy poprzednim uruchomieniu wszystkie urządzenia były uruchomione czy nie).



Rys.7.13. Sterowanie pracą trzech urządzeń wersja 1

Z analizy programu wynika, że:

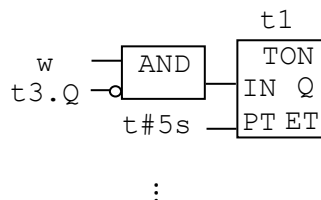
- timer t_1 jest uruchamiany po włączeniu łącznika włączającego,
- timer t_2 jest uruchamiany po włączeniu wyjścia Q timera t_1 ,
- timer t_3 jest uruchamiany po włączeniu wyjścia Q timera t_2 ,
- urządzenie pierwsze pracuje, gdy włączony jest łącznik włączający i nie upłynął czas niezbędny do włączenia wyjścia timera t_1 ,
- urządzenie drugie pracuje, gdy upłynął czas niezbędny do włączenia wyjścia timera t_1 i nie upłynął czas niezbędny do włączenia wyjścia timera t_2 ,
- urządzenie trzecie pracuje, gdy upłynął czas niezbędny do włączenia wyjścia timera t_2 i nie upłynął czas niezbędny do włączenia wyjścia timera t_3 .

Warto zauważyć, że po uruchomieniu wszystkich urządzeń wyjścia Q wszystkich timerów są włączone. Wyłączenie łącznika włączającego powoduje wyzerowanie wyjścia timera t_1 , co w konsekwencji zeruje wejście, a więc i wyjście timera t_2 i ostatecznie również wejście i wyjście timera t_3 . Zerowanie timerów jest niezależne od aktualnego stanu pracy urządzeń. Po włączeniu łącznika włączającego proces uruchamiania urządzeń jest wznowiany.

7.5.5. Przykład 4e

Układ sterujący pracą trzech urządzeń z poprzedniego przykładu wymaga od operatora wyłączenia i włączenia łącznika włączającego w celu ponownego rozpoczęcia cyklu. W nowej wersji proces jest wznawiany automatycznie: po uruchomieniu ostatniego urządzenia ponownie jest uruchamiane urządzenie pierwsze. Proces cyklicznego włączania i wyłączania urządzeń jest przerywany tylko po wyłączeniu łącznika.

```
PROGRAM Przyklad4e
VAR
  w AT %IX1.1: BOOL;
  u1 AT %QX2.1: BOOL;
  u2 AT %QX2.2: BOOL;
  u3 AT %QX2.3: BOOL;
  t1, t2, t3: TON;
END_VAR
```



dalsza część programu jak na rys. 7.13

Rys.7.14. Sterowanie pracą trzech urządzeń wersja 2

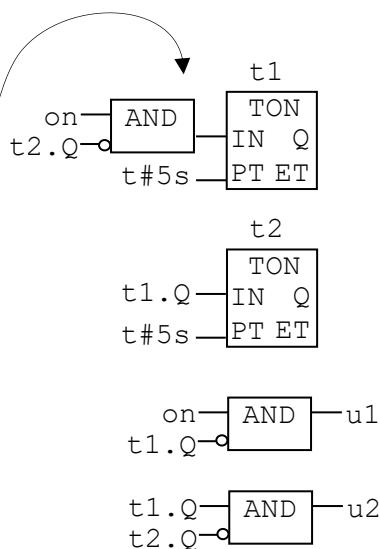
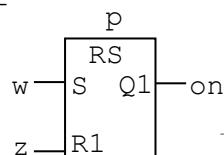
Wprowadzenie opisanej modyfikacji zasad pracy trzech urządzeń wymaga jedynie poprawienia warunku uruchamiającego timer $t1$. Timer ten jest uruchamiany teraz pod warunkiem, że włączony jest łącznik włączający i wyjście ostatniego timera jest wyłączone. Takie rozwiązanie powoduje, że:

- timer $t1$ jest uruchamiany po starcie programu (wyjście timera $t3$ nie jest włączone),
- po włączeniu timera $t3$ timer $t1$ jest zerowany, co w konsekwencji powoduje wyzerowanie pozostałych timerów, a wyzerowanie ostatniego z nich (przy włączonym łączniku włączającym) powoduje rozpoczęcie kolejnego cyklu pracy urządzeń.

7.5.6. Przykład 4f

Rozwiązanie wykorzystane w przykładzie poprzednim może być zaadaptowane do alternatywnego rozwiązania zadania sterowania pracą dwóch urządzeń z pomocą dwóch łączników bistabilnych z przykładu 4c. Zezwolenie na pracę układu zależy od stanu obydwu łączników i może być ustalone w taki sam sposób, jak w przykładzie 4c, w którym wykorzystany został przerzutnik SR ustawiający wartość pomocniczej zmiennej on . Zmienna ta może być wykorzystana w taki sam sposób, jak zmienna w w przykładzie poprzednim.

```
PROGRAM Przyklad4f
VAR
  w AT %IX1.1: BOOL;
  z AT %IX1.2: BOOL;
  u1 AT %QX2.1: BOOL;
  u2 AT %QX2.2: BOOL;
  on: BOOL;
  p: RS;
  t1: TON;
  t2: TON;
END_VAR
```



Rys.7.15. Sterowanie pracą dwóch urządzeń wersja 4

7.6. Zadania

- 7.1. Napisz program, którego zadaniem jest dwuręczne sterowanie ruchem urządzenia. Dla bezpieczeństwa, w celu uruchomienia urządzenia, należy wymusić położenie obu rąk na przyciskach sterowniczych – tzn. urządzenie powinno działać tylko wtedy, gdy jednocześnie wciśnięte są dwa przyciski służące do jego włączania.
- 7.2. Zmodyfikuj program z zadania 7.1 tak, aby uniemożliwić włączanie urządzenia za pomocą jednej ręki poprzez zablokowanie jednego z przycisków w stanie wciśniętym i operowanie tylko drugim. Należy założyć, że obydwa przyciski muszą być włączone w czasie nie dłuższym niż 500 ms.
Wskazówka: urządzenie jest włączane tylko wtedy, gdy wciśnięte są obydwa przyciski i nie upłynął czas dłuższy od 500 ms od wciśnięcia jednego z nich.
- 7.3. Napisz program, którego zadaniem jest sterowanie ruchem siłownika jednostronnego działania. Siłownik powinien wykonać jeden ruch roboczy po wciśnięciu przez operatora przycisku włączającego (przycisk to łącznik monostabilny, należy założyć, że operator nie musi go trzymać przez cały czas pracy siłownika). Po całkowitym wysunięciu tłoczyska (sygnalizowane stanem wysokim czujnika siłownika) należy rozpocząć ruch powrotny.
- 7.4. Zmodyfikuj program z zadania 7.3 tak, aby po całkowitym wysunięciu tłoczyska siłownik pozostawał w tym stanie przez określony wcześniej czas.
- 7.5. Zmodyfikuj program z zadania 7.4 tak, aby siłownik cyklicznie wykonywał ruchy pomiędzy skrajnymi położeniami sygnalizowanymi przez 2 czujniki, w które jest wyposażony. Ruch roboczy należy rozpocząć, jeżeli tłoczysko siłownika jest całkowicie wsunięte (sygnalizowane stanem wysokim pierwszego czujnika). Po całkowitym wysunięciu tłoczyska (sygnalizowane stanem wysokim drugiego czujnika) należy rozpocząć ruch powrotny.
- 7.6. Zmodyfikuj program z zadania 7.5, dodając możliwość włączania i wyłączania pracy cyklicznej za pomocą przycisków START i STOP.