

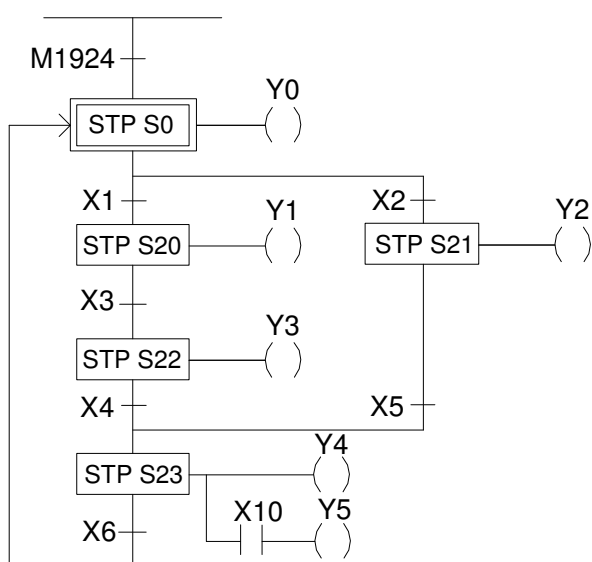
Rozdział 8 Opis programowania krokowego

Programowanie strukturalne jest wiodącym trendem w dziedzinie projektowania oprogramowania. Korzyściami takiej metody programowania są: dobra czytelność, łatwa obsługa, wygodne aktualizacje, wysoka jakość i niezawodność. Aplikacje sterujące, składające się z wielu sekwencyjnie wykonanych zadań, są trudne w konserwacji, gdy ich implementacja oparta jest na klasycznym języku drabinkowym. Dlatego też należy łączyć wykorzystywane na szeroką skalę schematy drabinkowe z elementami programowania sekwencyjnego stworzonymi specjalnie na potrzeby przebiegu pracy maszyny. Instrukcje krokowe sprawiają, że projektowanie staje się wydajniejsze, oszczędne czasowo i łatwe do opanowania. Ten rodzaj programowania, który łączy w sobie sterowanie procesem ze schematem drabinkowym nosi nazwę krokowego języka drabinkowego.

Podstawową jednostką krokowego schematu drabinkowego jest krok. Krok jest odpowiednikiem indywidualnego ruchu (kroku) podczas pracy maszyny. Całkowity proces pracy maszyny lub proces sterowania sekwencyjnego jest szeregową lub równoległą kombinacją kroków. Procedura realizowana krok po kroku pozwala na łatwiejsze zrozumienie działania maszyny tak, aby jej działanie i obsługa stały się bardziej efektywne i prostsze.

8.1 Zasada działania krokowego schematu drabinkowego

【Przykład】



【Opis】

1. **STP Sxxx** jest symbolem reprezentującym krok Sxxx z zakresu S0~S999. Podczas wykonywania kroku (stan WŁ), wykonywana zostaje operacja zapisana na schemacie drabinkowym w prawej odnodze kroku, a stan poprzedniego kroku i zapisanych w nim wyjść będzie WYŁ.
2. Po uruchomieniu programu, M1924 (marker pierwszego skanu) będzie aktywny. Nastąpi przejście do pierwszego kroku S0 (S0 WŁ), podczas gdy wszystkie pozostałe kroki będą nieaktywne, np. stany wszystkich wyjść Y1~Y5 będą WYŁ. Oznacza to, że stan M1924 WŁ→S0 WŁ→Y0 WŁ i status Y0 pozostanie WŁ do momentu, aż styki X1 lub X2 staną się aktywne.
3. Przy założeniu, że X2 będzie aktywny jako pierwszy, zostanie wykonane przejście do ścieżki S21.

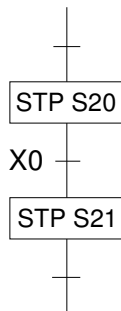
$$X2 \text{ ON} \Rightarrow \begin{cases} S21 \text{ WŁ} \\ S0 \text{ WYŁ} \end{cases} \Rightarrow \begin{cases} Y2 \text{ WŁ} \\ Y0 \text{ WYŁ} \end{cases}$$
 Y2 pozostanie WŁ do momentu, aż X5 będzie WŁ.
4. Przy założeniu, że X5 jest WŁ, proces przejdzie do kroku S23.

$$\text{tj. } X5 \text{ WŁ} \Rightarrow \begin{cases} S23 \text{ WŁ} \\ S21 \text{ WYŁ} \end{cases} \Rightarrow \begin{cases} Y4 \text{ WŁ} \\ Y2 \text{ WYŁ} \end{cases}$$
 Y4 i Y5 pozostaną WŁ do momentu aż X6 będzie WŁ.
 ※Jeżeli X10 jest WŁ, to Y5 będzie WŁ.
5. Przy założeniu, że X6 zostanie WŁ, proces przejdzie do S0.

$$\text{tj. } X6 \text{ WŁ} \Rightarrow \begin{cases} S0 \text{ WŁ} \\ S23 \text{ WYŁ} \end{cases} \Rightarrow \begin{cases} Y0 \text{ WŁ} \\ Y4, Y5 \text{ WYŁ} \end{cases}$$
 Po tym, cykl sterowania zostaje zakończony i rozpoczyna się nowy cykl.

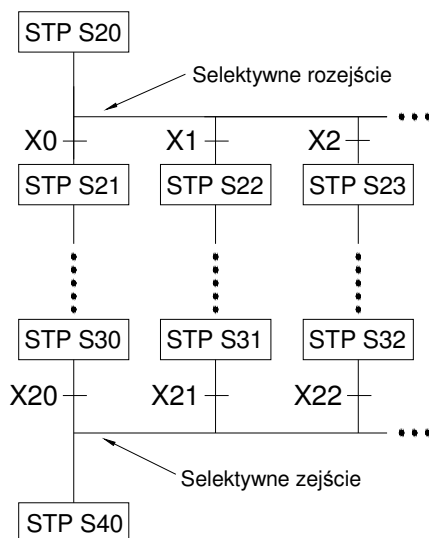
8.2 Podstawowe układy krokowego schematu drabinkowego

① Pojedyncza ścieżka



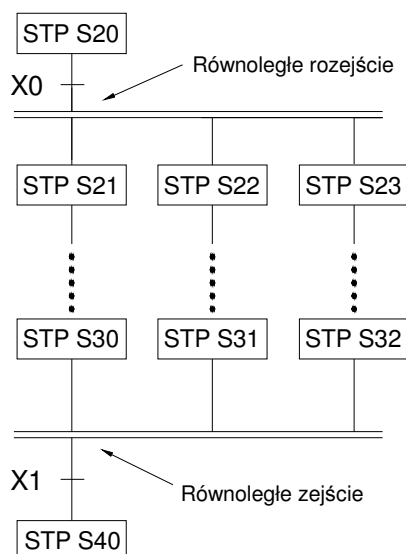
- Krok S20 przechodzi do kroku S21 gdy X0 w stanie WŁ.
- X0 może być zmieniony na inną szeregową lub równoległą kombinację styków.

② Selektywne rozejście / zejście



- Krok S20 wybiera ścieżkę, która pierwsza spełniła warunek przejścia. Np.: jeżeli X2 będzie WŁ jako pierwszym to zrealizowana zostanie jedynie ścieżka kroku S23.
- Maksymalna ilość ścieżek dla selektywnego rozejścia to 8.
- X1, X2,, X22 mogą być zastąpione szeregową lub równoległą kombinacją innych styków.

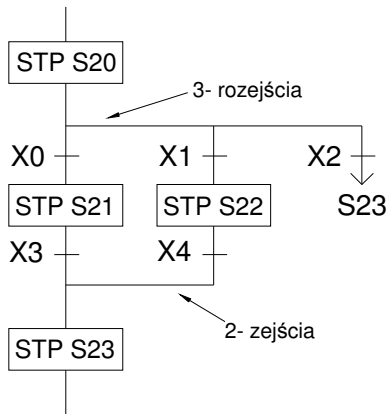
③ Równoległe rozejście/ zejście



- Po zmianie stanu X0 na WŁ, krok S20 zrealizuje równoległe przejście do wszystkich ścieżek poniżej, tj. wszystkie kroki S21, S22, S23, będą aktywne.
- Równoległe ścieżki będą zrealizowane do ostatniego kroku w zbieżnym punkcie. Dopiero gdy wszystkie równoległe ścieżki się zakończą (w przykładzie.: S30, S31 i S32 będą aktywne), jeżeli X1 jest WŁ, realizacja ścieżek może zostać przeniesiona do S40.
- Liczba rozejść ścieżek musi być identyczna jak ilość zejść ścieżek. Maksymalną liczbą ścieżek równoległych to 8.

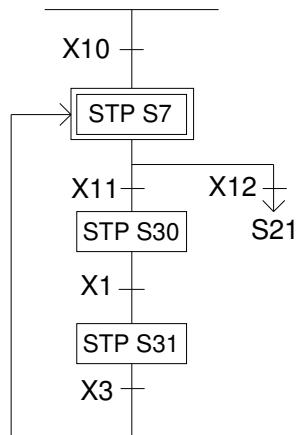
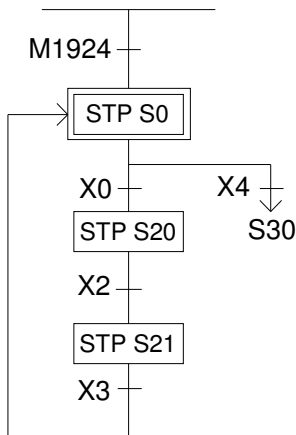
④ Skok

a. Do tej samej pętli krokowej



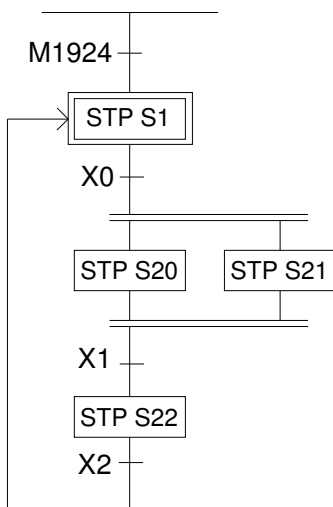
- Jak pokazano na przykładzie po lewej, istnieją 3 ścieżki dla kroku S20. Przy założeniu, że X2 jest WŁ, proces może przeskoczyć bezpośrednio do kroku S23 w celu zrealizowania go bez przechodzenia przez proces selektywnego zejścia (S21,S22).
- Skok jest niemożliwy w przypadku równoległego wykonywania ścieżek.

b. Do innej pętli krokowej

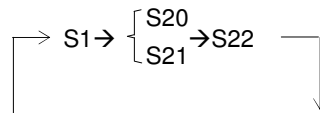


⑤ Zamknięta pętla i pojedynczy cykl

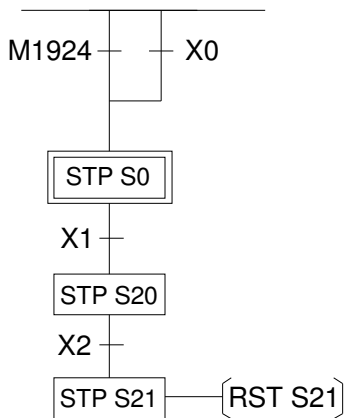
a. Zamknięta pętla



- Po załączeniu sterownika, krok S1 jest aktywny i cykl kroków będzie wykonywany w nieskończoność

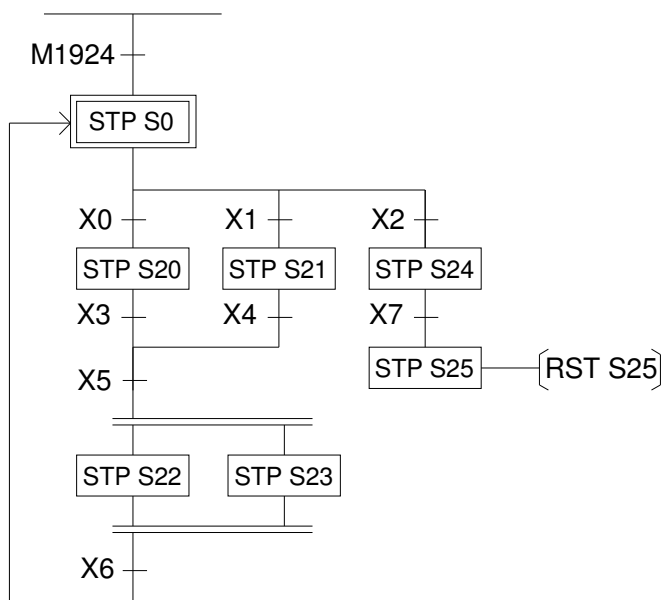


b. Pojedynczy cykl

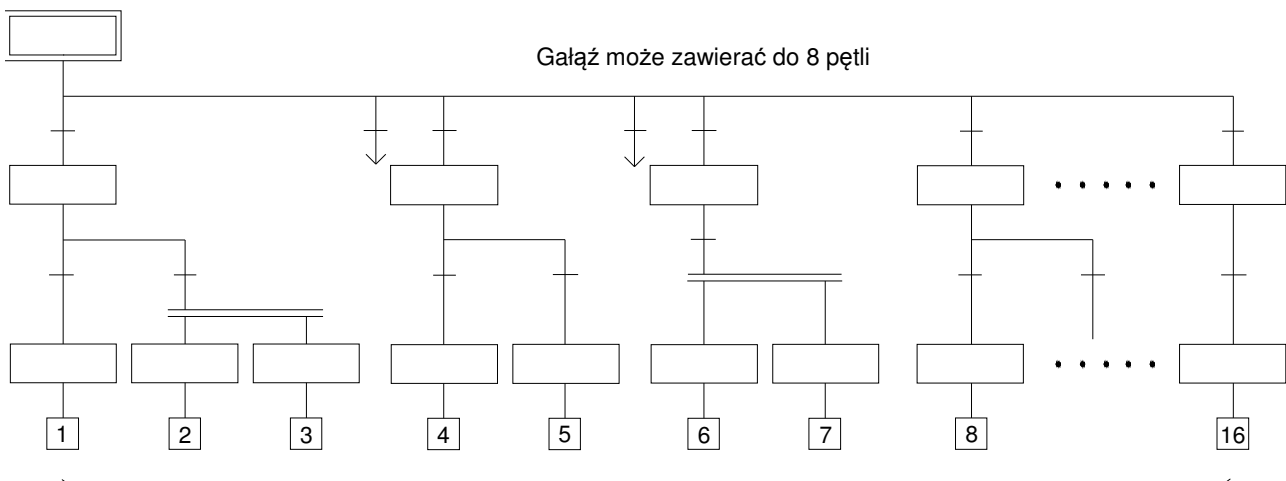


- Jeżeli krok S20 będzie aktywny i X2 będzie WŁ, to funkcja „RST S21” wyłączy S21, co zakończy wykonywanie całego procesu.

c. Proces mieszany



⑥ Złożona aplikacja



Gałąź może zawierać do 8 pętli

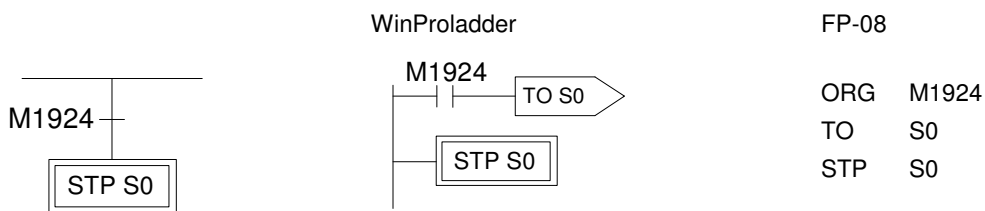
Maksymalna ilość pętli dla dolnej poziomej gałęzi kroku inicjującego wynosi 16

8.3 Wprowadzenie do instrukcji krokowych: STP, FROM, TO i STPEND

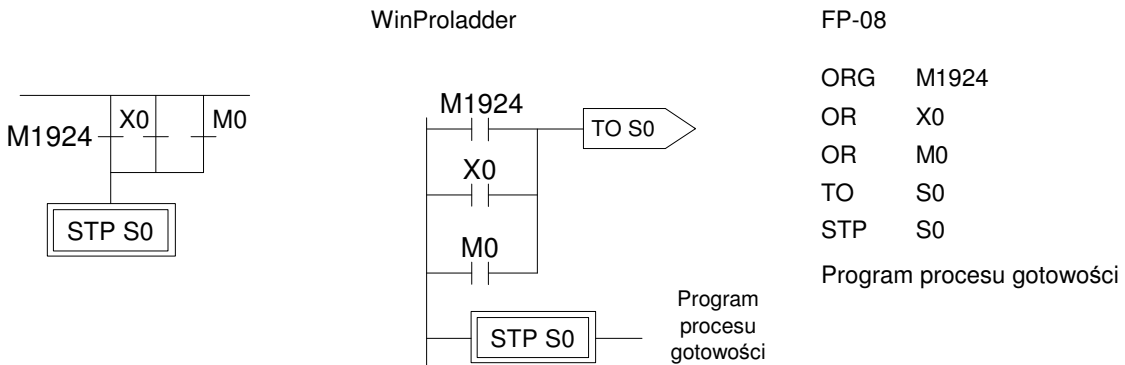
- **STP Sx** : $S0 \leq Sx \leq S7$ (Wyświetlane w WinProladder)
lub
STP Sx : $S0 \leq Sx \leq S7$ (Wyświetlane w FP-08)

Instrukcja ta jest krokiem inicjującym, od którego rozpoczyna się sterowanie krokowe każdego procesu maszyny. W serii FBs istnieje 8 kroków inicjujących (S0~S7), tj. PLC może zrealizować kontrolę do 8 procesów równocześnie. Każdy proces może działać niezależnie lub generować wyniki będące wykorzystywane w pozostałych procesach.

【Przykład 1】 Po każdym uruchomieniu sterownia przejdź do kroku inicjującego S0



【Przykład 2】 Za każdym razem gdy urządzenie jest uruchamiane, naciskany jest przycisk lub urządzenie działa nieprawidłowo, następuje automatyczne przejście do kroku inicjującego S0 (gotowości).



【Opis】 X0: Ręczny przycisk, M0: marker informujący o nieprawidłowej pracy maszyny.

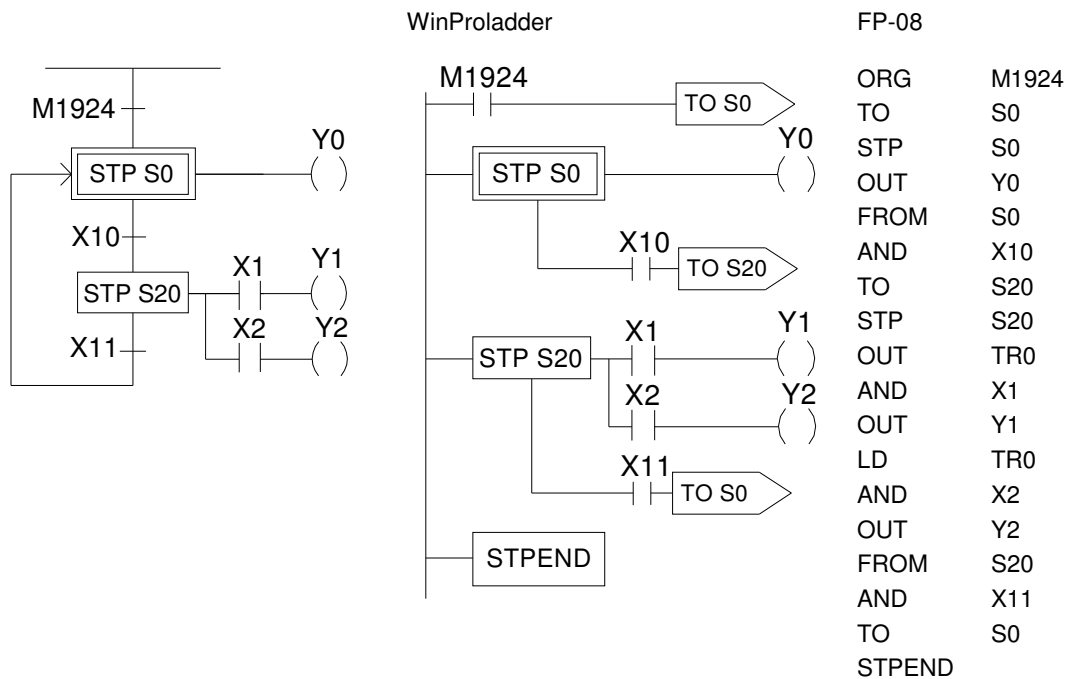
- **STP Sxxx** : S20≤Sxxx≤S999 (Wyświetlane w WinProladder)

lub

STP Sxxx : S20≤Sxxx≤S999 (Wyświetlane w FP-08)

Instrukcja ta jest krokiem. Każdemu kroku procesu odpowiada jedna instrukcja kroku w sekwencji. Jeżeli stan kroku jest WŁ, to krok jest aktywny i realizowany jest program drabinkowy odpowiadający temu krokowi.

【Przykład】



【Opis】 1. Po załączeniu sterownika, inicjujący krok S0 i Y0 są WŁ.

2. Przy pojawieniu się warunku przejścia X10 WŁ (w rzeczywistym zastosowaniu warunek przejścia może być tworzony przez szeregową lub równoległą kombinację styków X, Y, M, T i C), aktywowany jest krok S20. System automatycznie wyłączy krok S0 w tym samym skanie programu, co spowoduje przejście Y0 w stan WYŁ..

$$\text{tj. } X10 \text{ WŁ} \Rightarrow \begin{cases} S20 \text{ WŁ} \\ S0 \text{ WYŁ} \end{cases} \Rightarrow \begin{cases} X1 \text{ WŁ} \rightarrow Y1 \text{ WŁ} \\ X2 \text{ WŁ} \rightarrow Y2 \text{ WŁ} \\ Y0 \text{ WYŁ} \end{cases}$$

3. Jeżeli warunek przejścia X11 jest WŁ, to krok S0 jest WŁ, Y0 jest WŁ, natomiast S20, Y1 i Y2 będą WYŁ w tej samej chwili.

$$\text{tj. } X11 \text{ ON} \Rightarrow \begin{cases} S0 \text{ WŁ} \\ S20 \text{ WYŁ} \end{cases} \Rightarrow \begin{cases} Y0 \text{ WŁ} \\ Y1 \text{ WYŁ} \\ Y2 \text{ WYŁ} \end{cases}$$

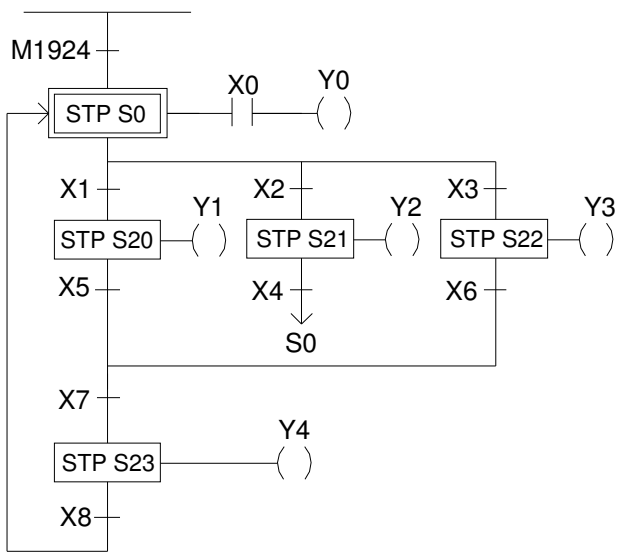
● FROM Sxxx : $S0 \leq Sxxx \leq S999$ (Wyświetlane w WinProladder)

lub

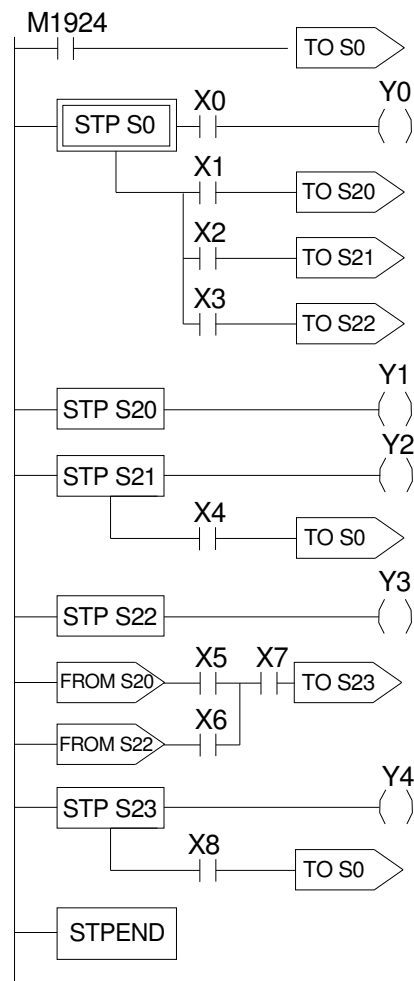
FROM Sxxx : $S0 \leq Sxxx \leq S999$ (Wyświetlane w FP-08)

Instrukcja opisuje krok źródłowy przejścia, tj. przejścia z kroku Sxxx do kolejnego kroku, w skorelowaniu z warunkiem przejścia.

【Przykład】



WinProladder



FP-08

```

ORG      M1924
TO       S0
STP      S0
AND      X0
OUT      Y0
FROM     S0
OUT TR   0
AND      X1
TO       S20
LD TR   0
AND      X2
TO       S21
LD TR   0
AND      X3
TO       S22
LD TR   0
AND      X4
TO       S23
LD TR   0
AND      X5
TO       S20
LD TR   0
AND      X6
TO       S21
LD TR   0
AND      X7
TO       S22
LD TR   0
AND      X8
TO       S23
LD TR   0
STPEND

```

【Opis】 : 1. Po załączeniu sterownika, inicjujący krok S0 jest WŁ. Jeżeli X0 jest WŁ, to Y0 będzie WŁ.

2. Jeżeli S0 jest WŁ: a. jeżeli X1 jest WŁ, to krok S20 i Y0 będą WŁ.

b. jeżeli X2 jest WŁ, to krok S21 i Y2 będą WŁ

c. jeżeli X3 jest WŁ, to krok S22 i Y3 będą WŁ

d. jeżeli X1, X2 i X3 są jednocześnie WŁ, to najpierw krok S20 będzie WŁ, a krok S21 lub S22 nie będzie WŁ.

e. jeżeli X2 i X3 są jednocześnie WŁ, to najpierw krok S21 będzie WŁ, a krok S22 nie będzie WŁ

3. Przy S20 WŁ, jeżeli X5 i X7 są jednocześnie WŁ, to krok S23 będzie WŁ, Y4 będzie WŁ, a S20 i Y1 będą WYŁ.

4. Przy S21 WŁ, jeżeli X4 jest WŁ, to krok S0 będzie WŁ, a S21 i Y1 będą WYŁ

5. Przy S22 WŁ, jeżeli X6 i X7 są jednocześnie WŁ, to krok S23 będzie WŁ, Y4 będzie WŁ, a S22 i Y3 będą WYŁ

6. Przy S23 WŁ, jeżeli X8 jest WŁ, to krok S0 będzie WŁ, a S23 i Y4 będą WYŁ.

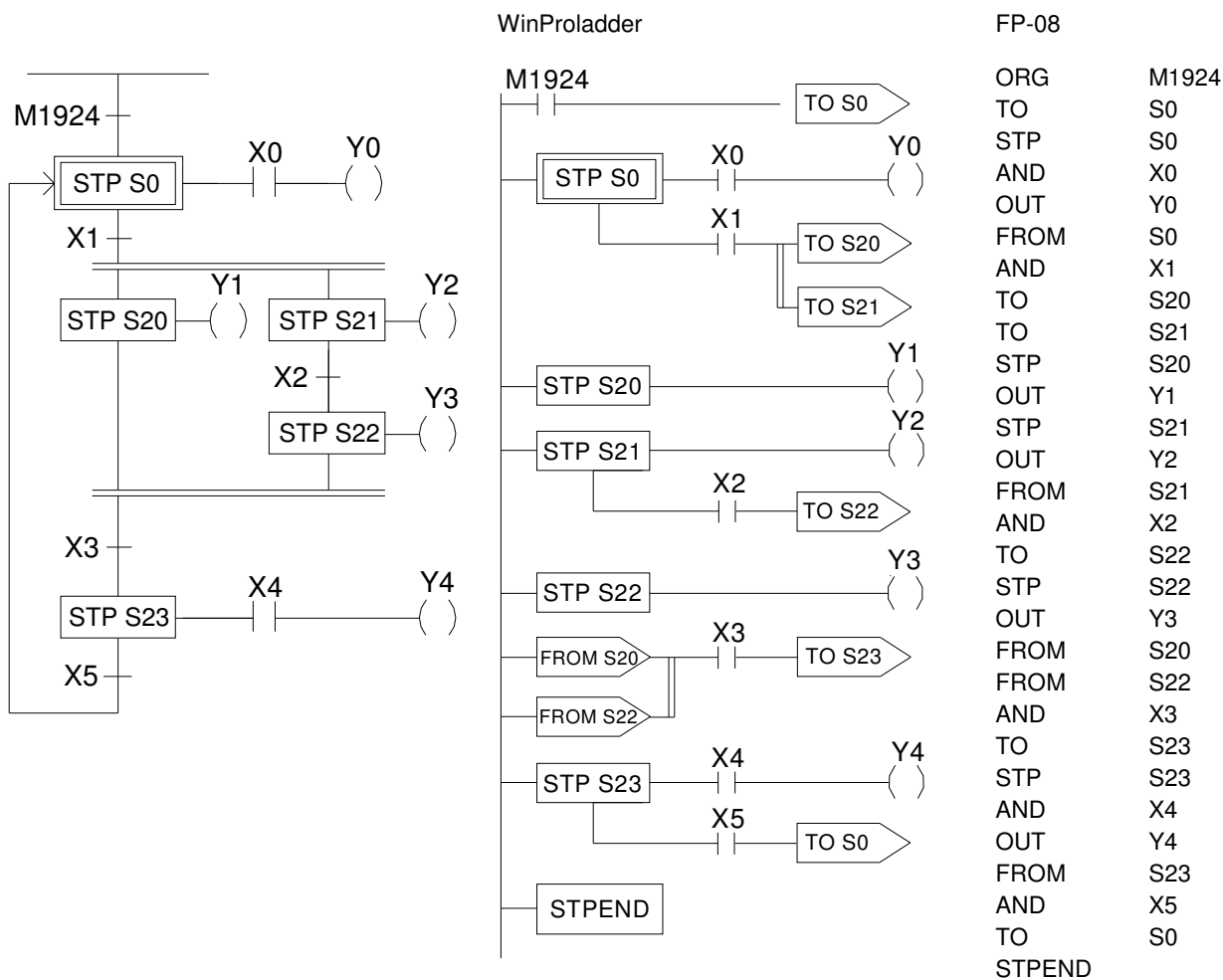
●  : $S0 \leq Sxxx \leq S999$ (Wyświetlane w WinProladder)

lub

TO Sxxx : $S0 \leq Sxxx \leq S999$ (Wyświetlane w FP-08)

Poniższa instrukcja określa krok, do którego ma być wykonane przejście.

【Przykład】



【Opis】 : 1. Po załączeniu sterownika, inicjujący krok S0 jest WŁ. Jeżeli X0 jest WŁ, to Y0 będzie WŁ

2. Jeżeli S0 jest WŁ i jeżeli X1 jest WŁ, to kroki S20 i S21 będą WŁ jednocześnie. Y1 i Y2 także będą WŁ.
3. Przy S21 WŁ, jeżeli X2 jest WŁ, to krok S22 będzie WŁ, Y3 będzie WŁ, a S21 i Y2 będą WYŁ.
4. Jeżeli S20 i S22 są WŁ jednocześnie i warunek transferu X3 jest WŁ, to krok S23 będzie WŁ (jeżeli X4 jest WŁ, to Y4 będzie WŁ), a S20 i S22 oraz Y1 i Y3 zostaną automatycznie wyłączone.
5. Przy S23 WŁ, jeżeli X5 jest WŁ, to proces przejdzie do początkowego kroku, tj. S0 będzie WŁ, a S23 i Y4 będą WYŁ.
6. Przy S23 WŁ, jeżeli X8 jest WŁ, to krok S0 będzie WŁ, a S23 i Y4 będą WYŁ.

STPEND : (Wyświetlane w WinProladder)

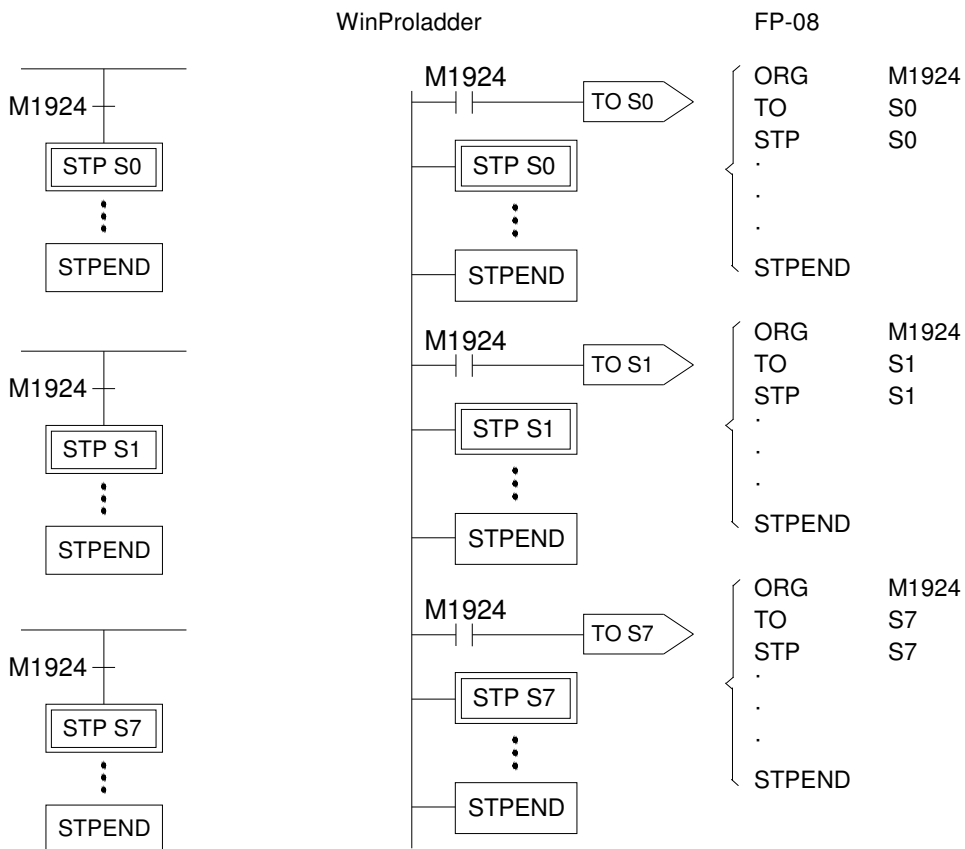
lub

STPEND : (Wyświetlane w FP-08)

Instrukcja ta reprezentuje zakończenie procesu. Aby procesy działały prawidłowo należy umieścić tę instrukcję w programie, na końcu każdego procesu rozpoczynającego się krokiem inicjującym.

PLC może kontrolować do 8 procesów krokowych (inicjowanych krokami S0~S7) równocześnie. W związku z tym można wprowadzić do 8-miu instrukcji STPEND.

【Przykład】

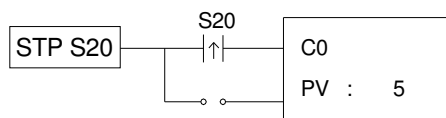


【Opis】 Po załączeniu sterownika, 8 procesów krokowych będzie aktywnych równocześnie.

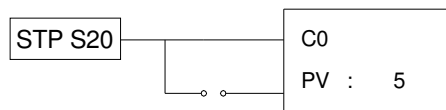
8.4 Uwagi dotyczące tworzenia krokowego schematu drabinkowego

【Uwagi】

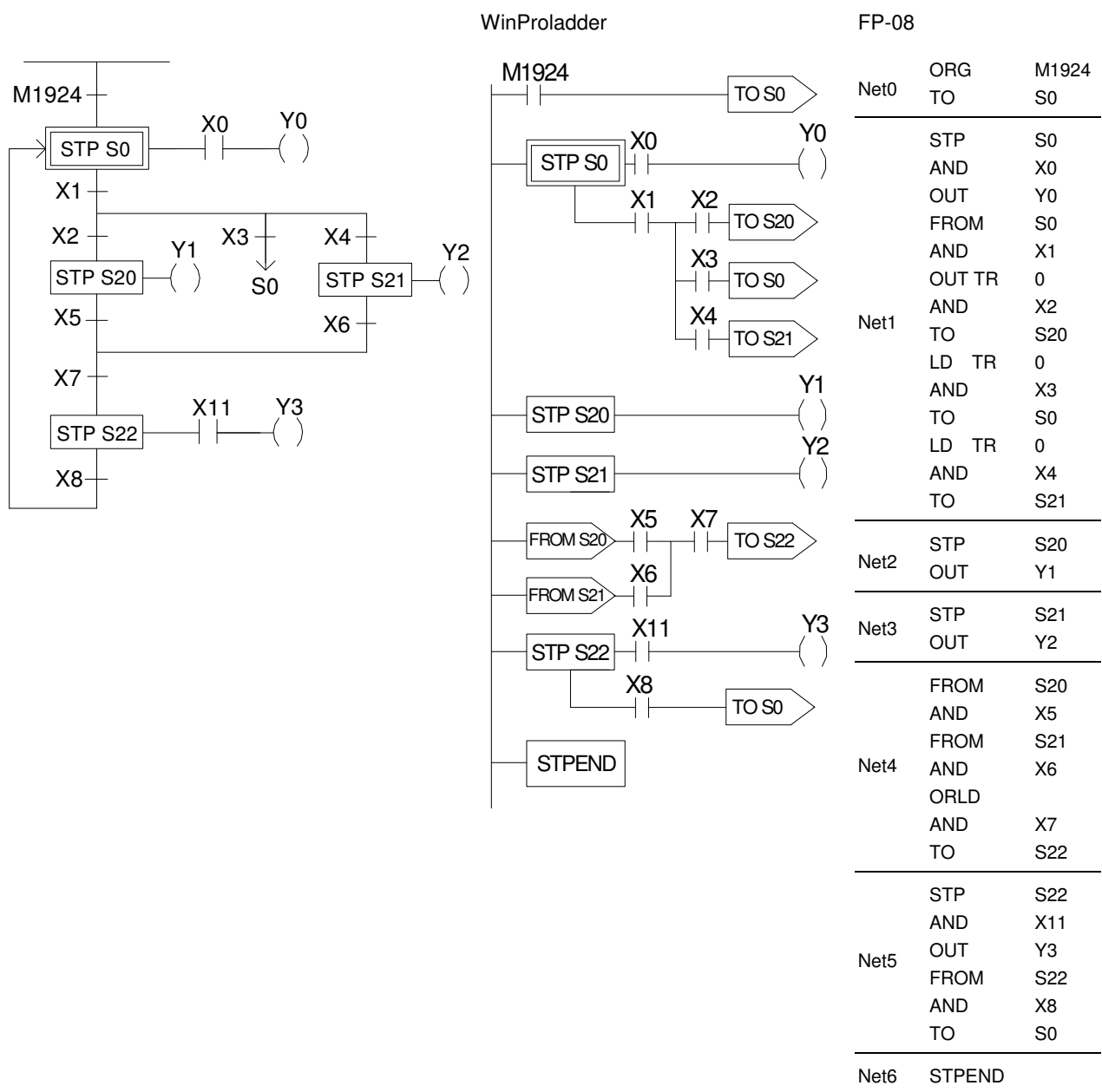
- Język drabinkowy i programowanie krokowe mogą być wykorzystywane w programie sterownika równocześnie.
- Istnieje 8 kroków, S0~S7, które mogą być wykorzystane jako punkt początkowy i nazywają się je „krokami inicjującymi”.
- Kiedy PLC zaczyna pracę, konieczna jest aktywacja kroku inicjującego. Do aktywowania kroku inicjującego może być wykorzystany marker systemowy M1924 (pierwszy skan programu).
- Z wyjątkiem kroków inicjujących, aktywacja pozostałych kroków musi być wywołana poprzednim krokiem.
- Aby program krokowy przeszedł proces sprawdzenia składni, musi rozpoczynać się krokiem inicjującym a kończyć instrukcją STPEND.
- Istnieje 980 kroków, S20~S999, które mogą być używane dowolnie. Jednakże wykorzystane już numery kroków nie mogą być użyte po raz kolejny. Kroki S500~S999 są podtrzymywane (zakres ten może być zmieniany przez użytkowników) - mogą być użyte w przypadku konieczności kontynuowania procesu po wyłączeniu maszyny.
- Generalnie, krok powinien składać się z trzech elementów: wyjścia sterującego, warunków przejścia i kroku docelowego przejścia.
- Instrukcje MC i SKP nie mogą być zastosowane w programie krokowym ani w podprogramach. Zaleca się unikania instrukcji JMP.
- Jeżeli wymagane jest, aby punkt wyjściowy kroku został w stanie WŁ po wyjściu z tego kroku, należy zastosować wyjście SET w celu ustawienia tego wyjścia na stałe, a po tym, odpowiednio wyjścia RST, w celu ustawienia go ponownie w stan WYŁ.
- Maksymalna ilość poziomych ścieżek, patrząc w dół od kroku inicjującego, wynosi 16. Jednakże do jednego kroku może być przypisanych do 8 rozgałęzień.
- Jeżeli M1918=0, a w pętli sterującej master (FUN 0) lub w programie krokowym wykorzystywana są funkcje reagujące na zbocze narastające (PULSE), to przed takimi funkcjami musi znaleźć się instrukcja TU. Na przykład:



Jeżeli M1918=1, to instrukcja TU nie jest wymagana, np.:

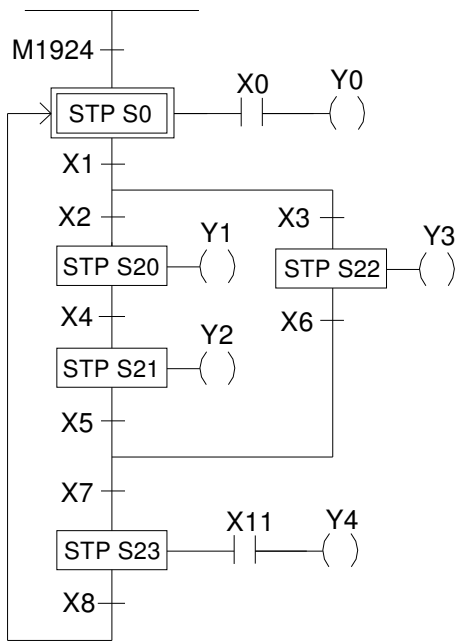


Przykład 1

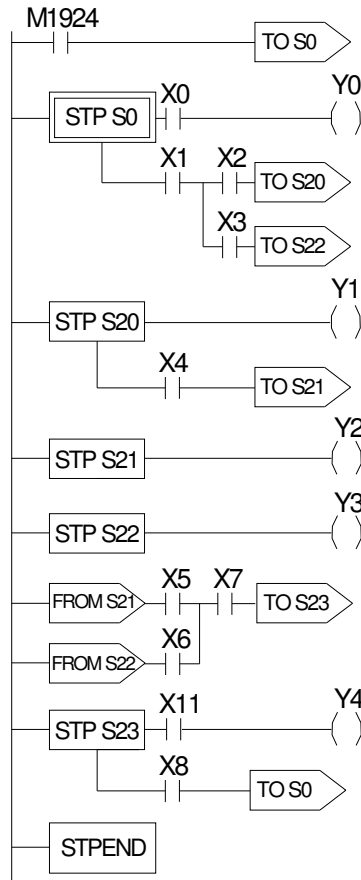


- Opis**
1. Wprowadź warunek przejścia do kroku inicjującego S0
 2. Wprowadź S0 i warunki rozejścia do S20, S0 i S21
 3. Wprowadź S20
 4. Wprowadź S21
 5. Wprowadź zejście z S20 i S21
 6. Wprowadź S22

Przykład 2



WinProLadder



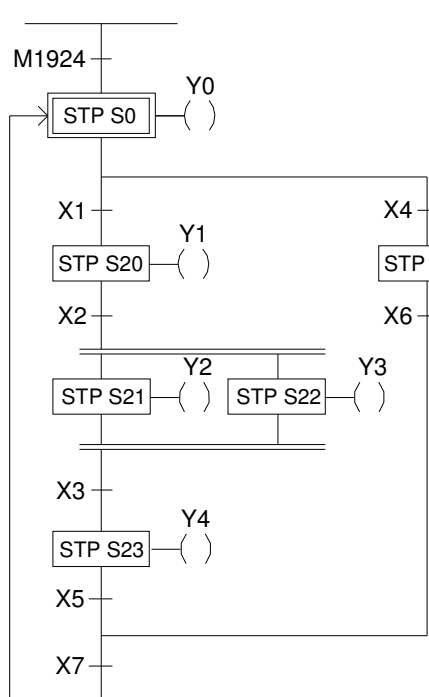
FP-08

Net0	{	ORG	M1924
		TO	S0
Net1	{	STP	S0
		AND	X0
		OUT	Y0
		FROM	S0
		AND	X1
		OUT TR	0
		AND	X2
		TO	S20
		LD TR	0
		AND	X3
		TO	S22
Net2	{	STP	S20
		OUT	Y1
		FROM	S20
		AND	X4
		TO	S21
Net3	{	STP	S21
		OUT	Y2
Net4	{	STP	S22
		OUT	Y3
Net5	{	FROM	S21
		AND	X5
		FROM	S22
		AND	X6
		ORLD	
		AND	X7
		TO	S23
Net6	{	STP	S23
		AND	X11
		OUT	Y4
		FROM	S23
		AND	X8
		TO	S0
Net7	{	STPEND	

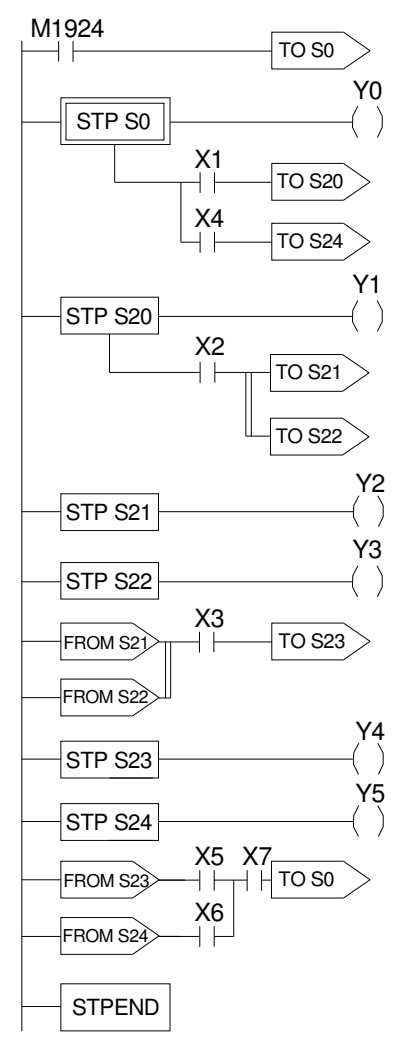
Opis

1. Wprowadź warunek przejścia do kroku inicjującego S0
2. Wprowadź S0 i warunki rozejścia do S20 i S22
3. Wprowadź S20
4. Wprowadź S21
5. Wprowadź S22
6. Wprowadź zejście z S21 i S22
7. Wprowadź S23

Przykład 3



WinProLadder



FP-08

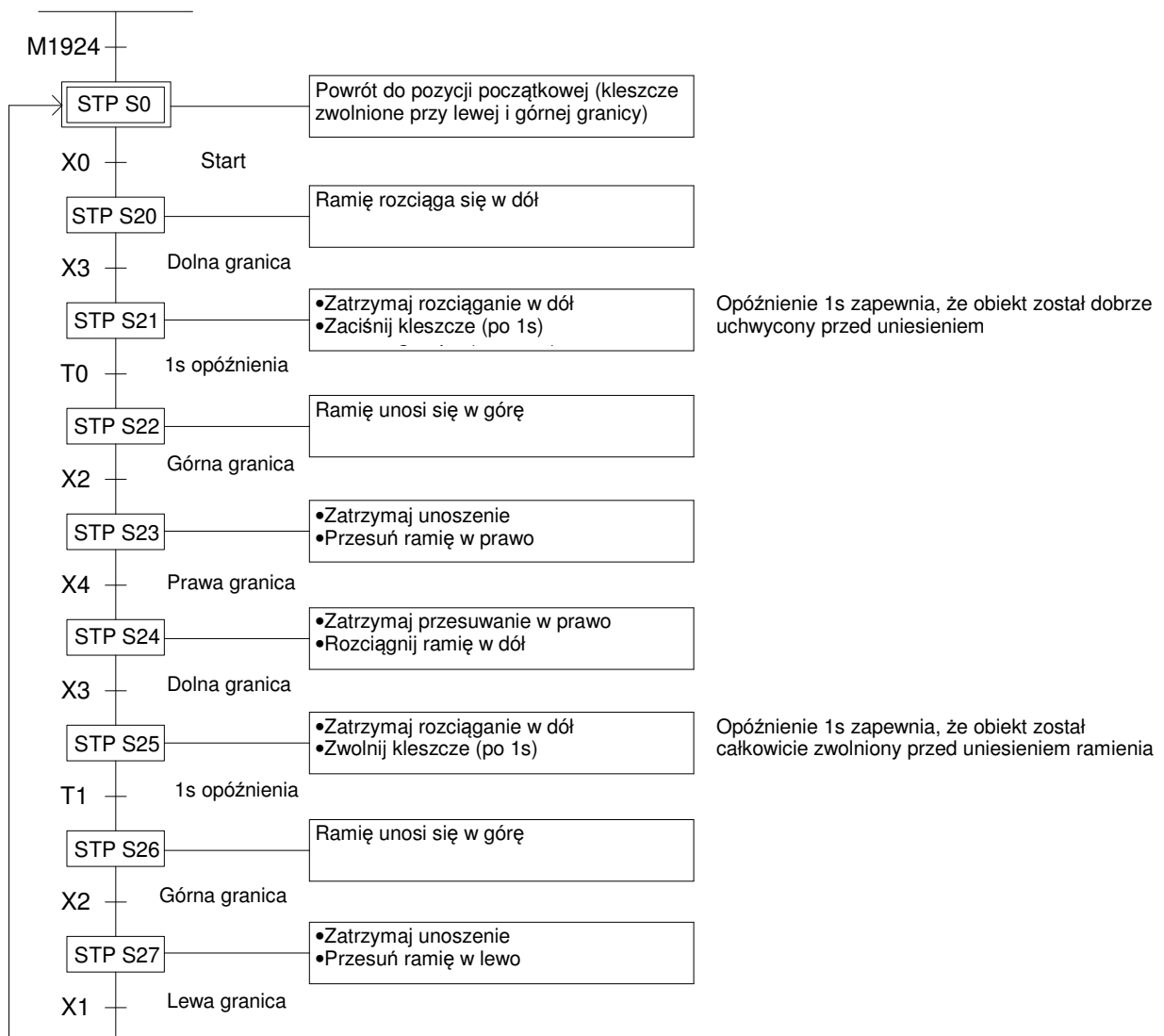
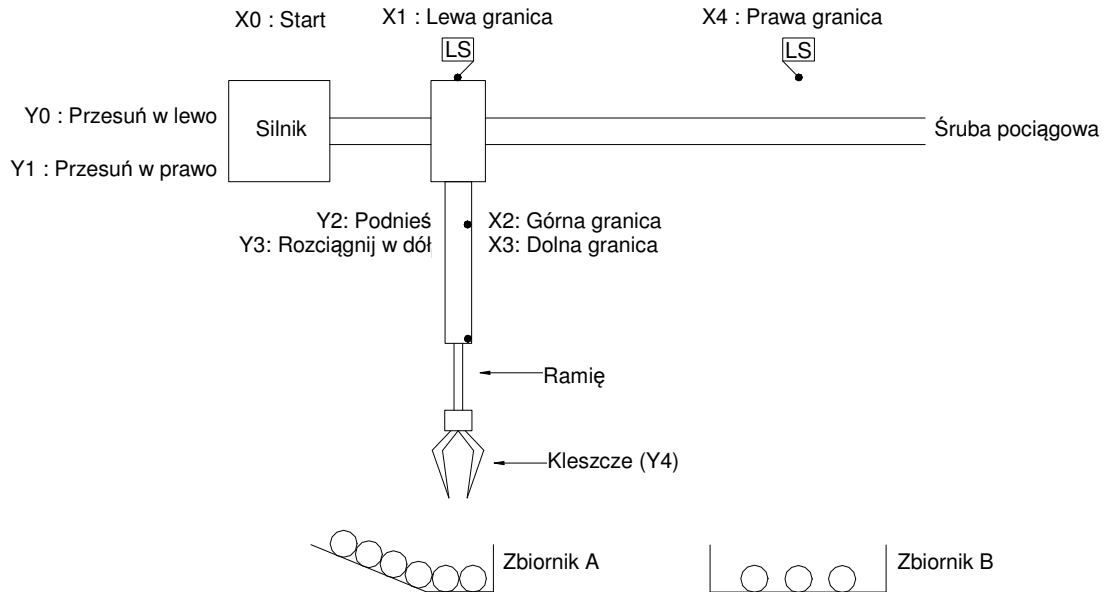
Net0	ORG TO	M1924 S0
Net1	STP OUT FROM OUT TR AND TO LD TR AND TO	S0 Y0 S0 0 X1 S20 0 X4 S24
Net2	STP OUT FROM AND TO TO	S20 Y1 S20 X2 S21 S22
Net3	STP OUT	S21 Y2
Net4	STP OUT	S22 Y3
Net5	FROM FROM AND TO	S21 S22 X3 S23
Net6	STP OUT	S23 Y4
Net7	STP OUT	S24 Y5
Net8	FROM AND FROM AND ORLD AND TO	S23 X5 S24 X6 X7 S0
Net9	STPEND	

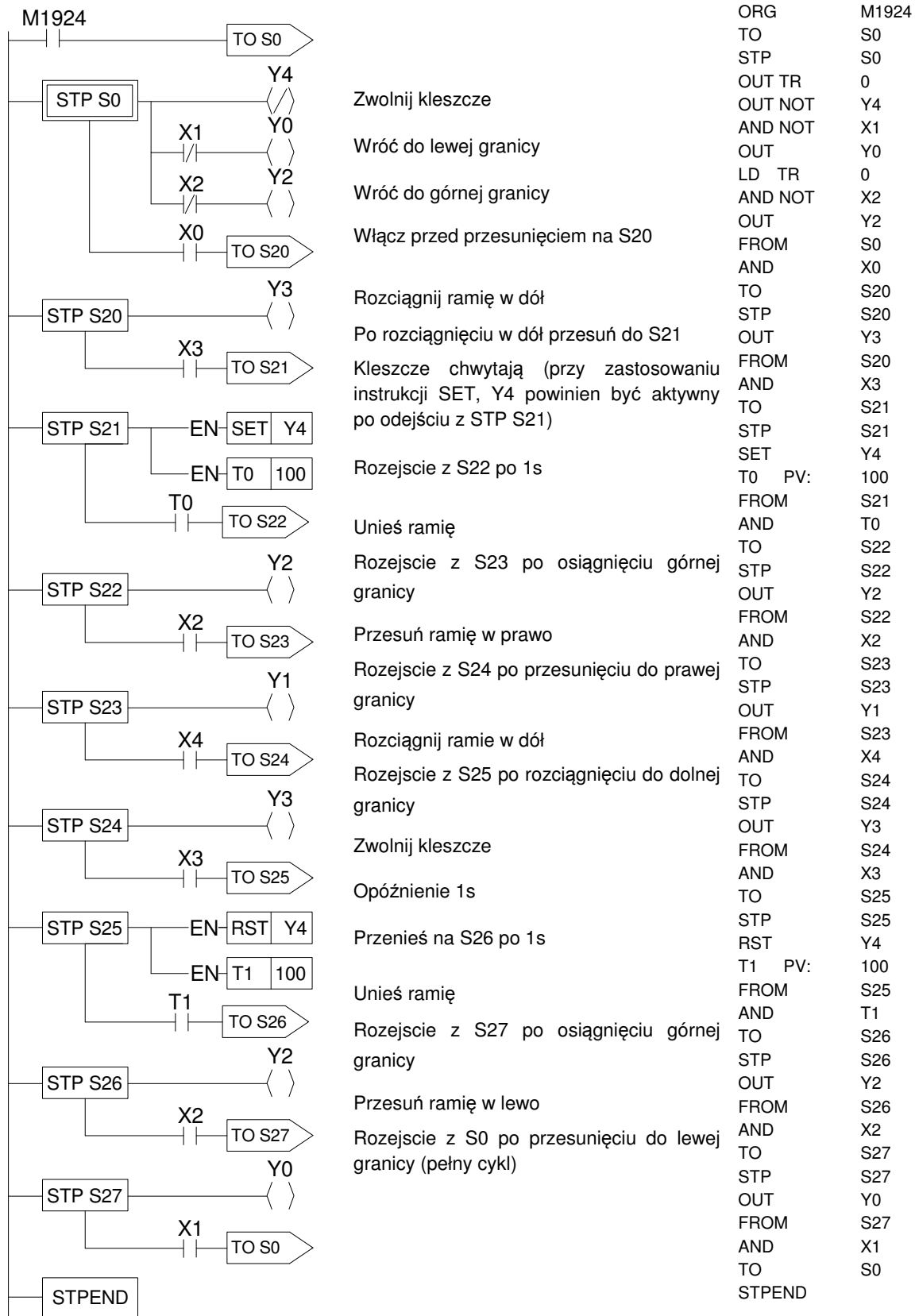
Opis

1. Wprowadź warunek przejścia do kroku inicjującego S0
2. Wprowadź S0 i warunki rozejścia do S20 i S24
3. Wprowadź S20
4. Wprowadź S20 i rozejście równoległe do S21 i S22
5. Wprowadź S21
6. Wprowadź S22
7. Wprowadź zejście równoległe z S21 i S22
8. Wprowadź S23
9. Wprowadź S24
10. Wprowadź zejście z S23 i S24

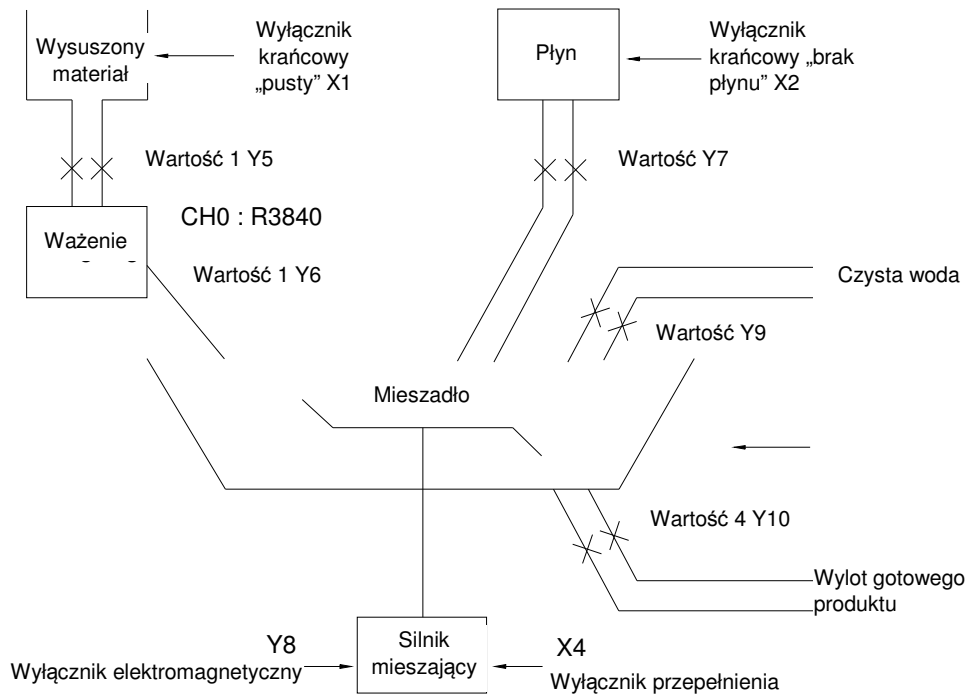
8.5 Przykłady zastosowań

Przykład 1 Chwyć obiekt ze zbiornika A i włóż go do zbiornika B



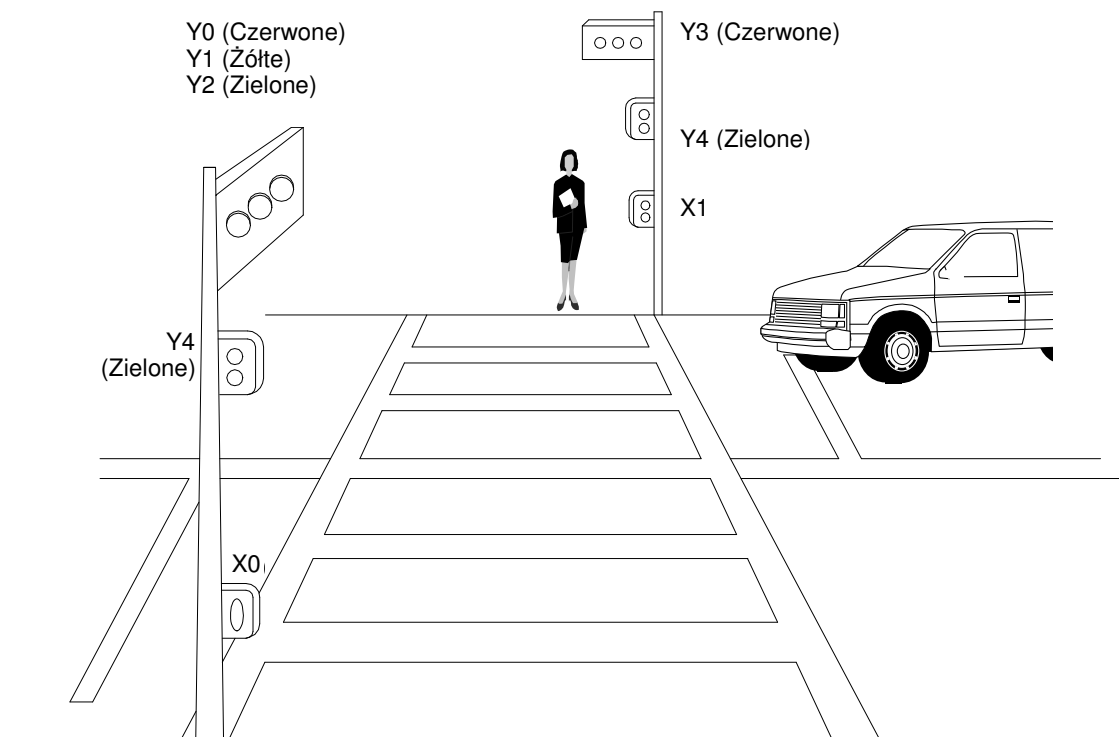


Przykład 2 Proces mieszania płynów



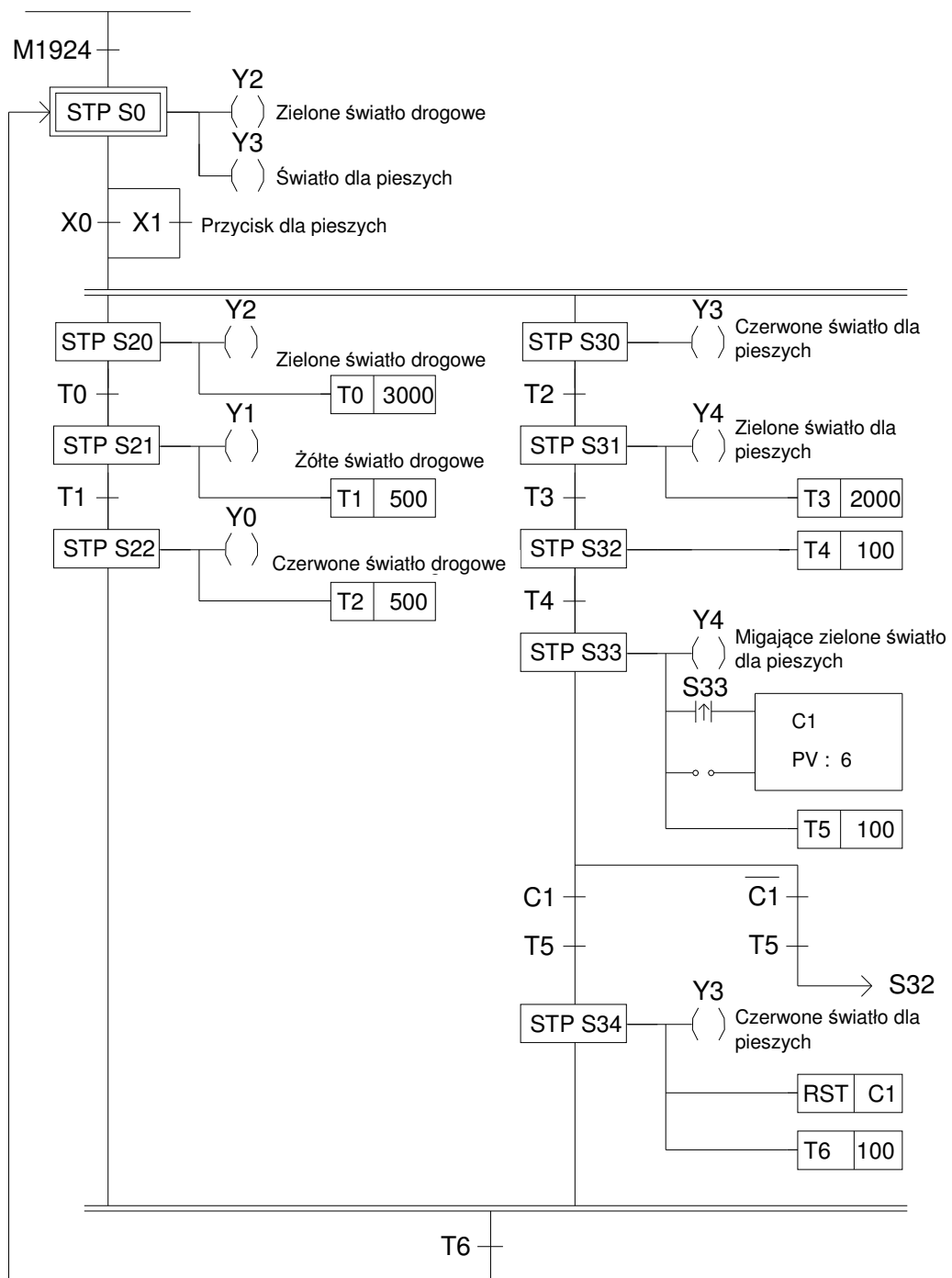
- Punkty wejściowe: Wyłącznik krańcowy „pusty” X1
 Wyłącznik krańcowy „brak płynu” X2
 Wyłącznik krańcowy „pusty” X3
 Wyłącznik przepiętnienia X4
 Przycisk ostrzegawczy X5
 Przycisk start X6
 Przycisk mycia wodą X7
- Wskaźniki ostrzegawcze: Opróżnij wysuszony materiał Y1
 Nieodpowiedni płyn Y2
 Opróżnij jednostkę mieszającą Y3
 Przeciążony silnik Y4
- Punkty wyjściowe: Zawór wlotowy osuszonego materiału Y5
 Zawór wlotowy osuszonego materiału Y6
 Zawór wlotowy płynu Y7
 Zawór elektromagnetyczny uruchamiający silnik Y8
 Zawór wlotowy czystej wody Y9
 Zawór wylotowy końcowego produktu Y10
- Wyjście ważące: CH0 (R3840)
- M1918=0

Przykład 3 Światła przy przejściu dla pieszych



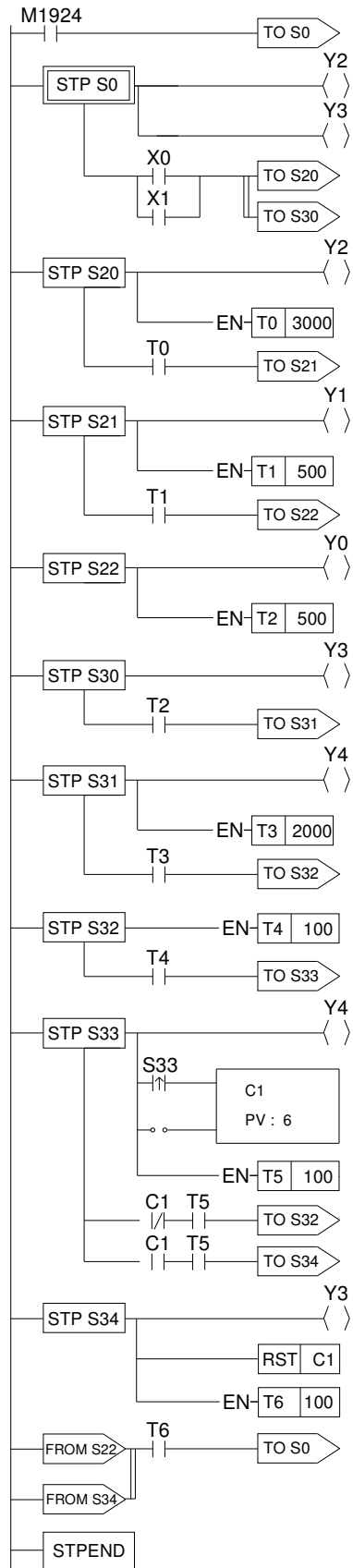
- ♦ Punkty wejściowe: Przycisk dla pieszych X0
- ♦ Punkty wyjściowe: Przycisk dla pieszych X1
- ♦ Punkty wyjściowe: Czerwone światło drogowe Y0
- ♦ Punkty wyjściowe: Żółte światło drogowe Y1
- ♦ Punkty wyjściowe: Zielone światło drogowe Y2
- ♦ Punkty wyjściowe: Czerwone światło dla pieszych Y3
- ♦ Punkty wyjściowe: Zielone światło dla pieszych Y4
- ♦ M1918=0

● Schemat sterowania światłami przy przejściu dla pieszych



● Program sterowania światłami przy przejściu dla pieszych

WinProLadder



FP-08

ORG	M1924	STP	S32
TO	S0	T4 PV:	100
STP	S0	FROM	S32
OUT	Y2	AND	T4
OUT	Y3	TO	S33
FROM	S0	STP	S33
LD	X0	OUT TR	0
OR	X1	OUT	Y4
ANDLD		LD TR	0
TO	S20	AND TU	S33
TO	S30	LD	OPEN
STP	S20	C1 PV:	6
OUT	Y2	LD TR	0
T0 PV:	3000	T5 PV:	100
FROM	S20	FROM	S33
AND	T0	OUT TR	1
TO	S21	AND NOT	C1
STP	S21	AND	T5
OUT	Y1	TO	S32
T1 PV:	500	LD TR	1
FROM	S21	AND	C1
AND	T1	AND	T5
TO	S22	TO	S34
STP	S22	STP	S34
OUT	Y0	OUT	Y3
T2 PV:	500	RST	C1
STP	S30	T6 PV:	100
OUT	Y3	FROM	S22
FROM	S30	FROM	S34
AND	T2	AND	T6
TO	S31	TO	S0
STP	S31	STPEND	
OUT	Y4		
T3 PV:	2000		
FROM	S31		
AND	T3		
TO	S32		

8.6 Kody błędów składni w programowaniu krokowym

Kody błędów związane z programowaniem krokowym są następujące:

- E51 : TO(S0-S7) musi rozpoczynać się instrukcją ORG.
- E52 : TO(S20-S999) nie może rozpoczynać się instrukcją ORG.
- E53 : Instrukcja TO bez odpowiedniej instrukcji FROM.
- E54 : Instrukcja TO musi występować po instrukcji TO, AND, OR, ANDLD lub ORLD.
- E56 : Przed instrukcją FROM muszą znajdować się instrukcje AND, OR, ANDLD lub ORLD.
- E57 : Instrukcja po FROM nie może być cewką ani funkcją.
- E58 : Cewka lub funkcja musi znajdować się przed FROM w linii STEP.
- E59 : Ponad 8 TO# w jednej linii.
- E60 : Ponad 8 FROM# w jednej linii.
- E61 : TO(S0-S19) musi znajdować się w pierwszym wierszu linii.
- E62 : Styk zajmuje miejsce dla instrukcji TO.
- E72 : Podwojona instrukcja TO Sxx.
- E73 : Podwojona instrukcja STP sxx.
- E74 : Podwojona instrukcja FROM sxx.
- E76 : STP(S0~S19) bez dopasowanej instrukcji STPEND lub STPEND bez dopasowanej instrukcji STP(S0~S19).
- E78 : Instrukcje TO(S20~S999), STP (S20~S999) lub FROM znajdują się przed STP(S0~S19) lub brak STP (S0~S19).
- E79 : Instrukcje STP Sxx lub FROM Sxx znajdują się przed TO Sxx lub brak TO Sxx.
- E80 : Instrukcja FROM Sxx znajduje się przed STP Sxx lub brak STP Sxx.
- E81 : Maksymalna liczba odgałęzień musi być ≤ 16 .
- E82 : Maksymalna liczba odgałęzień na tym samym poziomie musi być ≤ 16 .
- E83 : Nie wprowadzono instrukcji STP w sekwencji TO->STP->FROM.
- E84 : Definicja sekwencji STP# nie jest poprzedzona sekwencją TO#.
- E85 : Zejście nie odpowiada odpowiedniemu rozejściu.
- E86 : Niedozwolone zastosowanie STP lub FROM przed zejściem instrukcją TO.
- E87 : STP# lub FROM# znajdują się przed odpowiednią instrukcją TO#.
- E88 : W tej gałęzi, STP# lub FROM# znajduje się przed odpowiednią instrukcją TO#.
- E89 : FROM# znajduje się przed odpowiadającą mu instrukcją TO# lub STP#.
- E90 : Nieprawidłowe zastosowanie To# w równoległej gałęzi.
- E91 : Funkcja sterowania przepływem programu nie może być użyta w obrębie programu krokowego.