

## Notatka służbowa nr 1

Temat:	Sterowniki PLC (TSX Micro 37)
Wykonanie:	Zuzanna Mejer, 259382
Termin zajęć:	poniedziałek TP, 10:55

## 1 Cel ćwiczenia

Celem ćwiczenia było zaznajomienie się z obsługą sterownika PLC - konfiguracją oraz programowaniem w języku drabinkowym. Podczas ćwiczenia używany był sterownik TSX Micro 3721 firmy Schneider Automation z oprogramowaniem w wersji V1.6. Do zapoznawania się z podstawami języka drabinkowego korzystano z programu PL7PRO.

## 2 Uruchomienie oprogramowania i konfiguracja sterownika

Przed rozpoczęciem pracy ze sterownikiem, należało uruchomić i skonfigurować oprogramowanie i sterownik. W tym celu wykonano poniższe czynności.

1. Uruchomiono program PL7PRO i utworzono nowy projekt. Pojawiło się okno, w którym należało wybrać odpowiedni sterownik (TSX Micro), model wraz z wersją oprogramowania (TSX 3721 V1.5 - ze względu na brak wersji V1.6) oraz zaznaczyć „No” przy Grafcet. Przedstawia to poniższe zdjęcie.

tu foto

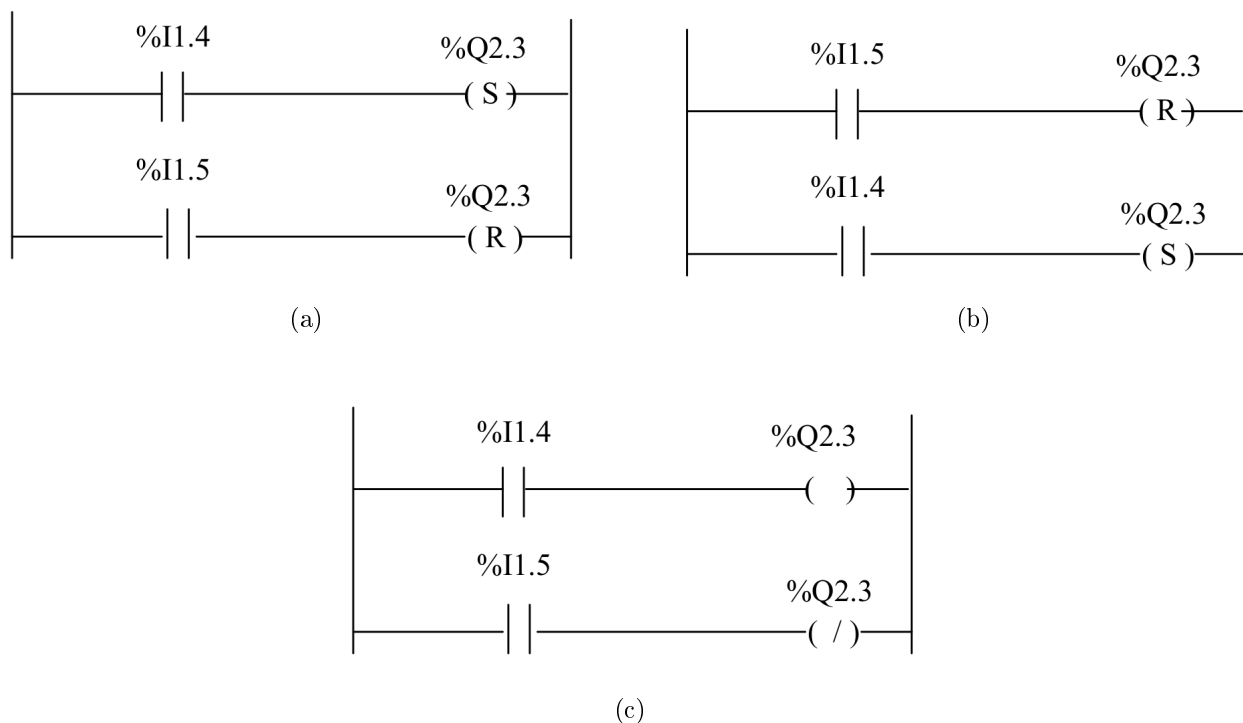
2. Następnie zdefiniowano typy modułów sterownika. Wybrano opcję *Configuration* → *Hardware configuration* i wpisano odpowiednie typy modułów. W przypadku stanowiska pierwszego były one następujące:
  - moduł wejść i wyjść cyfrowych - TSX DMZ 28DR
  - moduł wejść analogowych - TSX AEZ 414
  - moduł wyjść analogowych - TSX ASZ 200
3. Wybrano język programowania drabinkowy i przećwiczone podstawy obsługi programu oraz przesyłanie programu do sterownika i uruchamianie go.

## 3 Podstawowe funkcje logiczne i układy czasowe

Po zakończonej konfiguracji programu i sterownika, na fragmencie linii produkcyjnej wykonywano zadania dostępne w instrukcji laboratoryjnej.

### 3.1 Zadanie wprowadzające

Dane były trzy układy do przetestowania i przeanalizowania:



Rysunek 1: Układy do przeanalizowania

W programowaniu w języku drabinkowym należy pamiętać o tym, że sterownik wykonuje działania kolejno od górnego szczebla w kierunku dolnego. Różnica pomiędzy układami 1a a 1b polega na zamienieniu kolejności szczebli. Układ 1b nie zadziała poprawnie, ze względu na to, że program najpierw zrobi reset cewki  $\%Q2.3$ , a dopiero potem set.

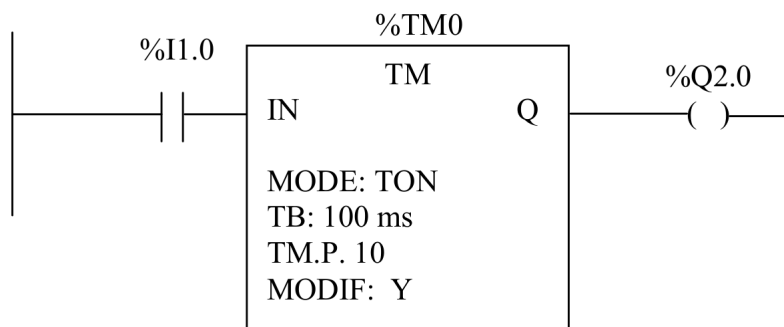
W układzie 1c wciśnięcie przycisku  $\%I1.4$  zaktywuje zmienną  $\%Q2.3$  do momentu, aż przycisk nie zostanie puszczony. Wciśnięcie przycisku  $\%I1.5$  zaneguje zmienną  $\%Q2.3$ . W praktyce na fragmencie linii produkcyjnej należało przycisnąć guziki wystarczająco długo, aby wykonał się pełen ruch.

## 3.2 Timery

W języku Ladder są dostępne trzy układy czasowe: TON, TOF oraz TP. Ich działanie zostało przetestowane oraz opisane poniżej.

### 3.2.1 Timer On-Delay (TON) - opóźnione zadziałanie

Układ TON został przedstawiony poniżej:

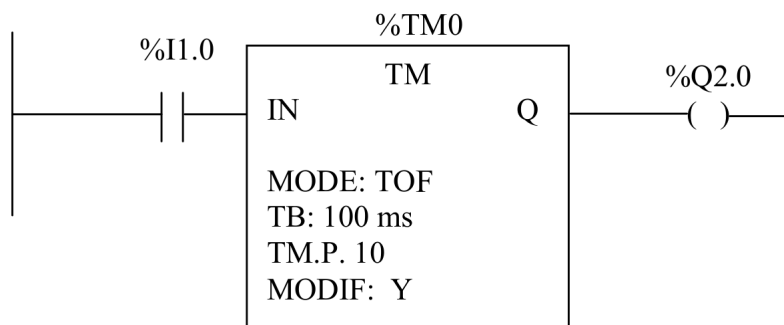


Rysunek 2: Timer On-Delay

Parametr  $TB$  to czas bazowy, ustawiony tutaj na 100 ms.  $TM.P$  to wartość zadana. Na podstawie tych parametrów obliczany jest czas opóźnienia:  $TB \cdot TM.P = 100ms \cdot 10 = 1000ms = 1s$ . W tym przypadku opóźnienie ustawiono na 1s. Timer On-Delay opóźnia działanie o podany czas. To oznacza, że po wciśnięciu przycisku  $\%I1.0$ , timer odmierzył 1 sekundę i dopiero po tym czasie zaktywował zmienną  $\%Q2.0$ .

### 3.2.2 Timer Off-Delay (TOF) - opóźnione wyłączenie

Układ TOF został przedstawiony poniżej:



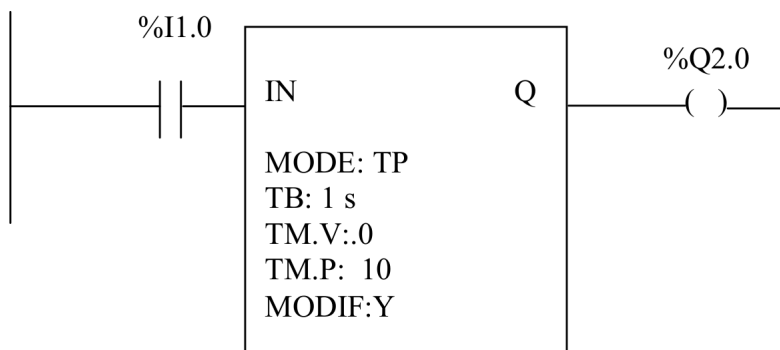
Rysunek 3: Timer Off-Delay

Po wciśnięciu przycisku  $\%I1.0$  program od razu aktywuje zmienną  $\%Q2.0$  i wyłącza program po odliczeniu czasu wyliczonego jako  $TB \cdot TM.P$ .

przebiegi?

### 3.2.3 Timer Pulse (TP) - puls

Układ TP został przedstawiony poniżej:



Rysunek 4: Timer Pulse

Parametr  $TMi.V$  to bieżąca wartość timera. Timer ten działa pulsacyjnie, to znaczy, że każda operacja trwa określoną długość (w tym przypadku 10s) i następnie przełącza się na wykonywanie kolejnej operacji.

### 3.3 Przykład zmiany stałej PRESET Timera przy pomocy funkcji *operate*

Zbudowany został następujący układ:

tutaj foto

tutaj opis

### 3.4 Liczniki

tutaj foto

tutaj opis

## 4 Podsumowanie