PRz – AiRA

UKŁADY CZASOWE

Programowanie z licznikiem cykli. Fala prostokątna. Standardowe czasomierze. Fala prostokątna w LD. Zabezpieczenie silnika. Drugie naciśnięcie. Minimalne układy z czasomierzami.

PROGRAMOWANIE Z LICZNIKIEM CYKLI

Podstawowym sposobem programowania układów czasowych w języku C i ewentualnie ST jest zastosowanie licznika cykli obliczeń (cykli wykonywania programu). Początkową wartość licznika ustawia się na zadany czas. Licznik jest dekrementowany w każdym cyklu, a gdy osiągnie wartość zero następuje aktywacja określonej akcji, przejście do następnego stanu itp. W poniższych przykładach licznik cykli oznaczono przez *tim*.

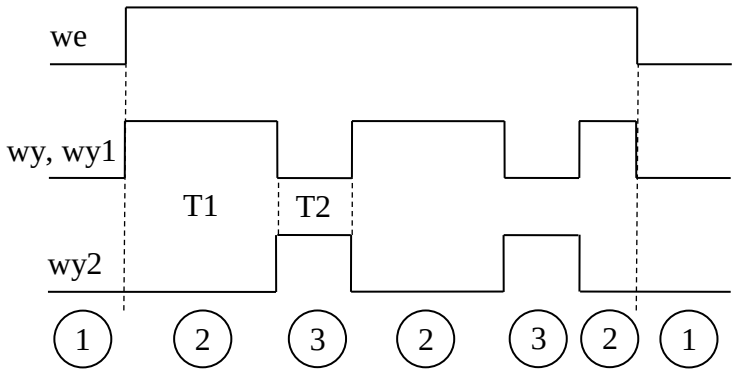
Warunkiem stosowania powyższego rozwiązania jest jednak konieczność zapewnienia stałego cyklu wykonywania programu. Tymczasem w aplikacjach PLC bardzo często wraca się do ponownego wykonania, gdy tylko poprzednie się zakończy (tzw. *PLC mode*). Kolejne czasy wykonywania programu mogą się istotnie różnić, więc nie ma mowy o stałym cyklu.

Precyzję odmierzania czasu w każdej sytuacji gwarantują specjalne bloki funkcjonalne normy PN/EN 61131–3 – czasomierze (*timery*).

FALA PROSTOKĄTNA

1. Problem

Jeżeli jest ustawione wejście *we* (np. zezwolenie), to gdy jedno z urządzeń pracuje, drugie jest wyłączone i na odwrót. Czas włączenia pierwszego urządzenia wynosi T1, a drugiego T2. Jeżeli sygnał *we* zaniknie, obydwa urządzenia zostają wyłączone.



Uwaga. Faktycznie chodzi o warunkowe wytworzenie fali prostokątnej o okresie T1+T2.

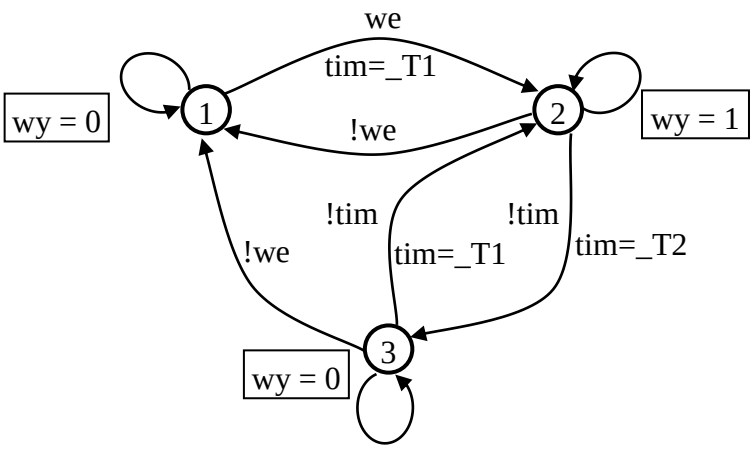
2. Stany

(1) oczekiwanie na aktywację – *we* = 0, *wy* = 0

(2) wyjście ustawione – *we* = 1, *wy* = 1

(3) wyjście nieustawione – *we* = 1, *wy* = 0

3. Graf automatu



wy1 = wy, wy2 = $\overline{wy} \cdot we$ (C: ! wy&&we)

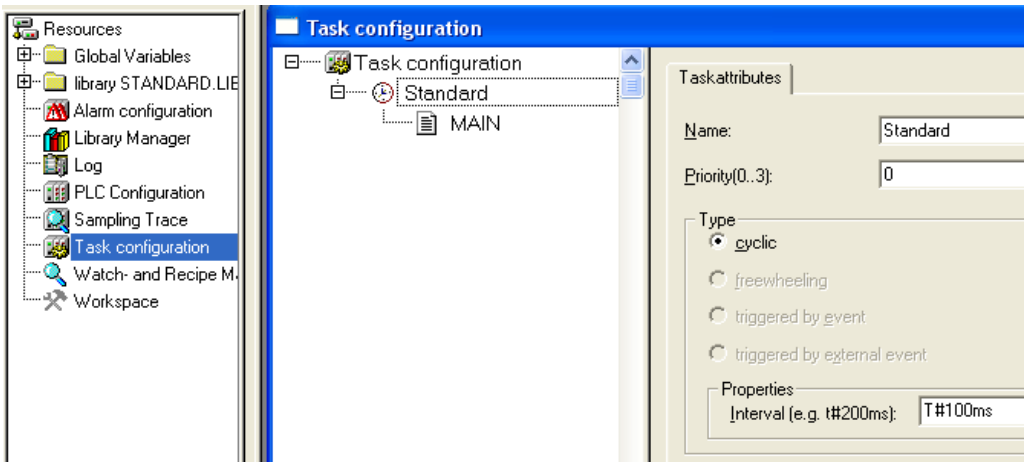
4. Kod C

```
switch(stan)
   {
    case 1: wy=0;
            if(we)  {tim=_T1; stan=2;}
            break;
    case 2: wy=1;
            if(!tim) {tim=_T2; stan=3;}
            else
                if(!we) stan=1;
            break;
    case 3: wy=0;
            if(!tim) {tim=_T1; stan=2;}
            else
                if(!we) stan=1;
            break;
    }

wy1=wy;  wy2=we && !wy;

if(tim) --tim;
```

5. Czas cyklu w TwinCAT

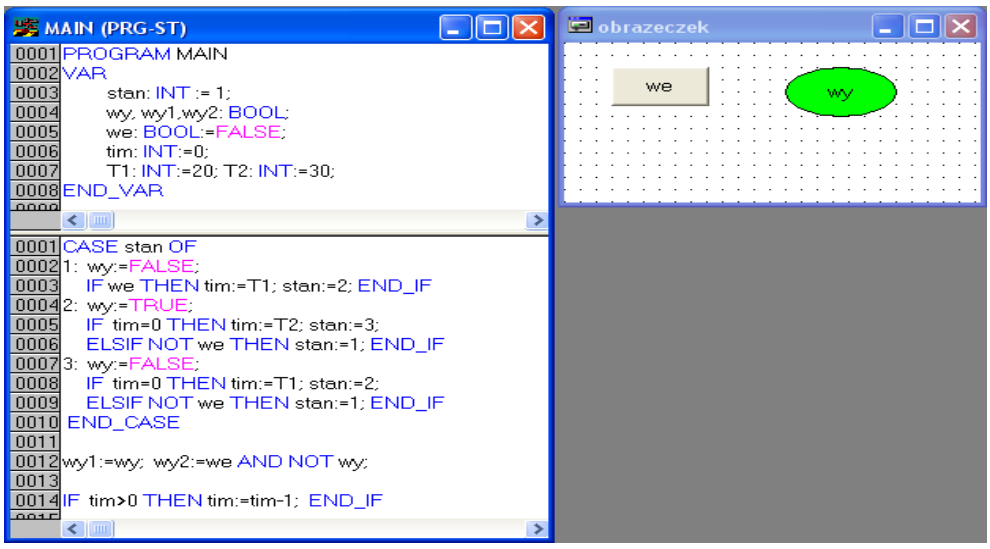
> Prawidłowe odmierzanie czasu przez TwinCAT PLC Control uruchomiony na PC wymaga uprzedniego uruchomienia programu *LowPrioProc*.

Cykl wykonywania zadania, odpowiednio do którego dobiera się początkową wartość zmiennej *tim*, ustawia się aktywując *Resources* na dole eksploratora (prawa dolna ikona) i po wybraniu *Task configuration* > *Standard* wpisując w komórce *Interval* odpowiednią wartość, np. *T#100ms* (zamiast domyślnego *T#10ms*).



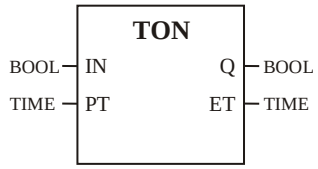
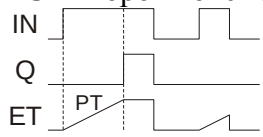
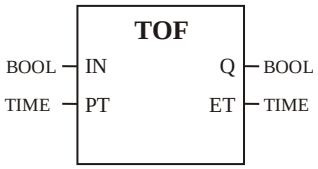
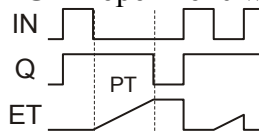
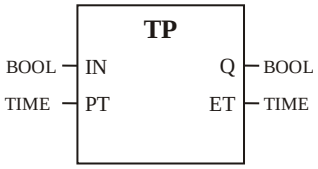
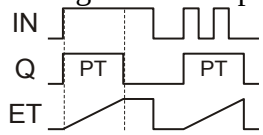
6. Kod ST

Projekt *Fala prostokątna – ST jak C*



STANDARDOWE CZASOMIERZE

1. Czasomierze normy PN/EN–61131-3

	**TON** – opóźnione załączenie (*on-delay timing*) 
	**TOF** – opóźnione wyłączenie (*off-delay timing*) 
	**TP** – generator impulsu (*pulse timing*) 

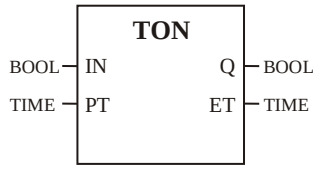
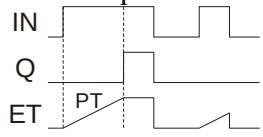
Wejścia:
- IN – wejście aktywujące
- PT – zadany czas (*preset time*)

Wyjścia:
- Q – wyjście główne
- ET – miniony czas (*elapsed time*)

Czasomierze odmierzają czas z dokładnością 1 ms (wymaganie normy PN/EN) wykorzystując liczniki obsługiwane w przerwaniu zegarowym (wysoki priorytet), a nie w programie głównym. W praktyce najczęściej stosuje się czasomierz TON.

2. Zasady zastosowania czasomierza TON

Chodzi o zastosowanie czasomierzy w programach sekwencyjno–czasowych napisanych w języku ST wykorzystując instrukcję *CASE stan OF…* Poniżej ograniczono się do czasomierza TON (z opóźnionym włączeniem), który od momentu pojawienia się TRUE na wejściu IN (narastające zbocze), po upływie czasu podanego na wejście PT, zmienia wyjście Q z FALSE na TRUE. Czasomierz ten powraca do stanu wyjściowego, gdy wejście IN staje się równe FALSE. Wyjście ET podaje upływ czasu.

	**TON** – opóźnione załączenie (*on-delay timing*) 

Zasady zastosowania czasomierza TON w programach sekwencyjno–czasowych są następujące:
- Wywołanie czasomierza następuje poza instrukcją *CASE stan OF…*
- Wejście IN musi być ustawione na FALSE przed aktywacją czasomierza przez podstawienie IN:=TRUE (narastające zbocze).

- Natychmiast po odmierzeniu czasu sygnalizowanego pojawieniem się TRUE na wyjściu Q wejście IN należy ustawić na FALSE (stan wyjściowy).

Najprościej realizuje się takie ustawianie instrukcją

IN:=(stan=n); – nawias niekonieczny,

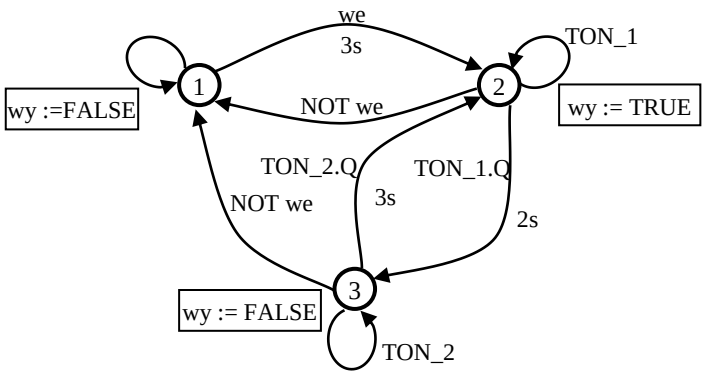
gdzie *stan=n* jest wyrażeniem boolowskim, a *n* oznacza stan, w którym czasomierz ma odmierzać czas. Jeżeli ten sam czasomierz miałby odmierzać czas w dwu stanach *n1* i *n2*, to wtedy mielibyśmy

IN:=(stan=n1 OR stan=n2);

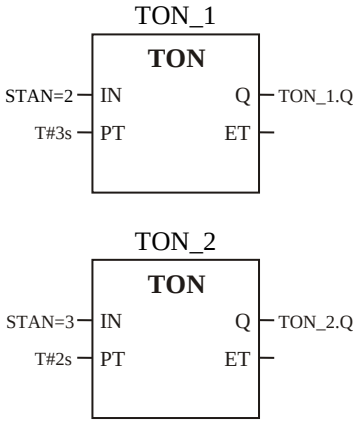
Stany *n1*, *n2* nie mogą jednak następować jeden po drugim (ewentualnie dodać IN:=FALSE; przechodząc ze stanu *n1* do *n2*).

3. Fala prostokątna z czasomierzami TON

- Automat




```
MAIN (PRG-ST)
0001 PROGRAM MAIN
0002 VAR
0003       stan: INT := 1;
0004       we: BOOL;
0005       wy, wy1,wy2: BOOL;
0006       TON_1:TON; TON_2:TON;
0007 END_VAR
0001 TON_1(IN:=STAN=2, PT:=T#3s);
0002 TON_2(IN:=STAN=3, PT:=T#2s);
0003
0004 CASE stan OF
0005 1:  wy:=FALSE;
0006     IF we THEN  stan:=2; END_IF
0007 2:  wy:=TRUE;
0008     IF  TON_1.Q THEN  stan:=3;
0009     ELSIF NOT we THEN stan:=1; END_IF
0010 3:  wy:=FALSE;
0011     IF  TON_2.Q THEN  stan:=2;
0012     ELSIF NOT we THEN stan:=1; END_IF
0013 END_CASE
0014
0015 wy1:=wy;  wy2:=we AND NOT wy;
0016
```

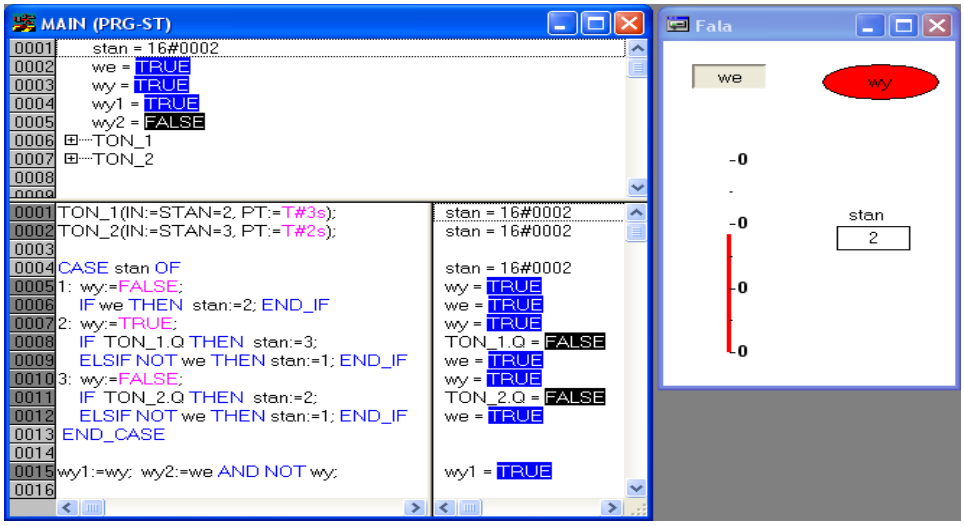


Fala prostokątna – ST TON

Uwaga. Czasomierze odmierzają czas na podstawie przerwań zegarowych, a nie cyklu zadania, który może pozostać na wartości domyślnej (T#10ms).

Norma IEC 61131 zakłada automatyczną konwersję liter małych do dużych, zatem *stan* i *STAN* oznaczają tę samą zmienną.

4. Praca układu



- Bargraf – *Bar display*
  Variable/Scale – MAIN, TON_1.ET, 0 – 3000 (ms)

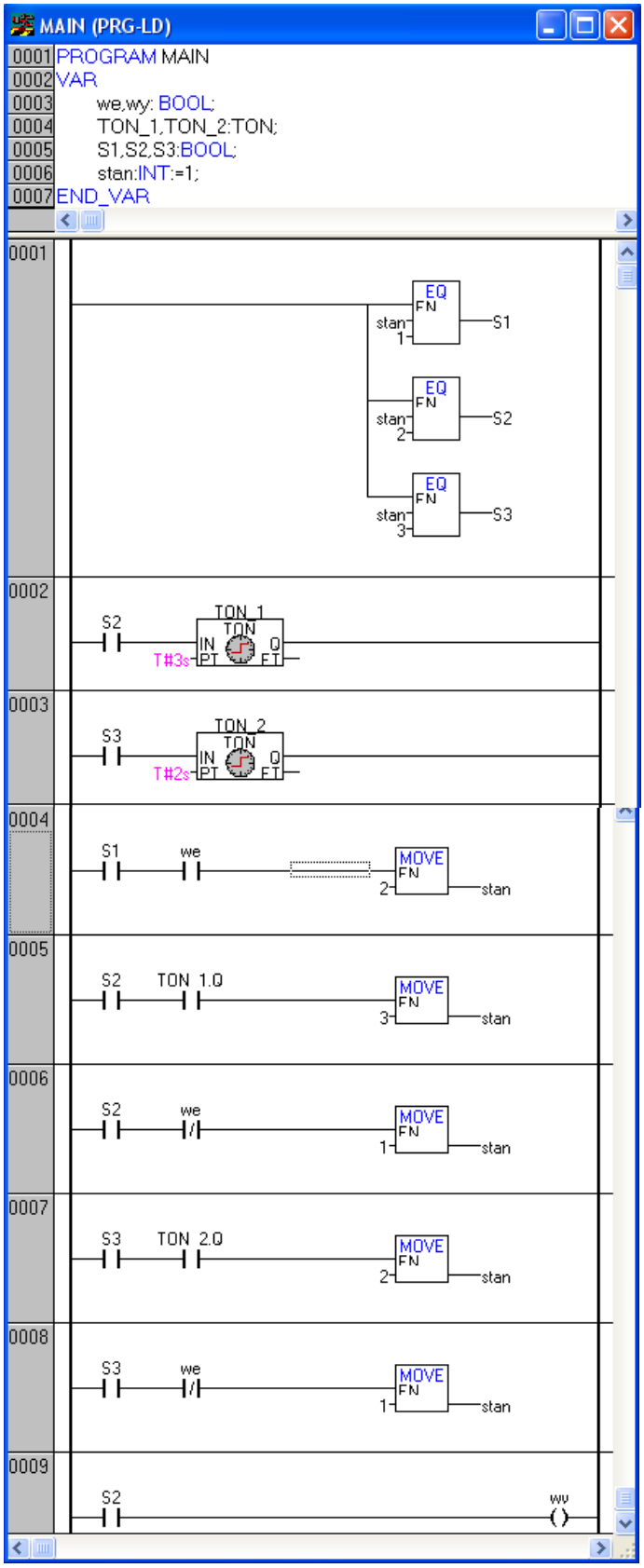
FALA PROSTOKĄTNA W LD

1. Konwersja ST → LD – zasady

- Program LD rozpoczyna się szczeblem (szczeblami) z funkcjami EQ generującymi zmienne boolowskie S1, S2, … reprezentujące stany.
- Po nich ustawia się szczeble czasomierzy aktywowane zmiennymi Si w odpowiednich stanach (jak w ST).
- Następne szczeble reprezentują przejścia miedzy stanami z funkcjami podstawienia MOVE, gdzie wyjścia czasomierzy TON.Q są stykami normalnie otwartymi.
- Końcowe szczeble ustawiają wyjścia sterujące.

  Realizacja poprzez konwersję ST → LD nie jest realizacją minimalną, tzn. niektóre zmienne, funkcje, styki i szczeble można byłoby usunąć. Jej zaletą jest jednak czytelność i łatwość wyszukania ewentualnego błędu. Skrócenie kodu nie ma praktycznie znaczenia przy realizacji programowej w PLC.

2. Kod LD

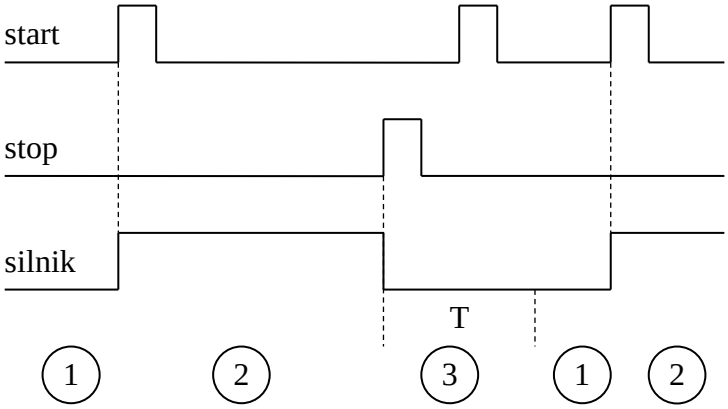


Fala prostokątna – LD TON

ZABEZPIECZENIE SILNIKA

1. Problem

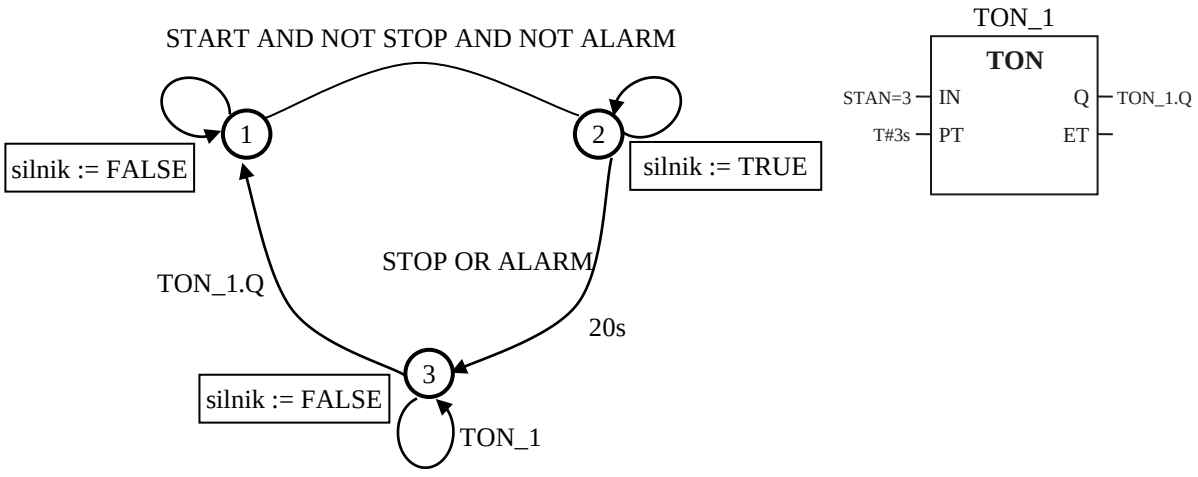
Silnik jest załączany/wyłączany przyciskami *start*, *stop*, z tym że po naciśnięciu *stop* ponowne włączenie może nastąpić dopiero po czasie *T*. Na sygnał *alarm* (zabezpieczenie termiczne) silnik reaguje podobnie jak na *stop*.



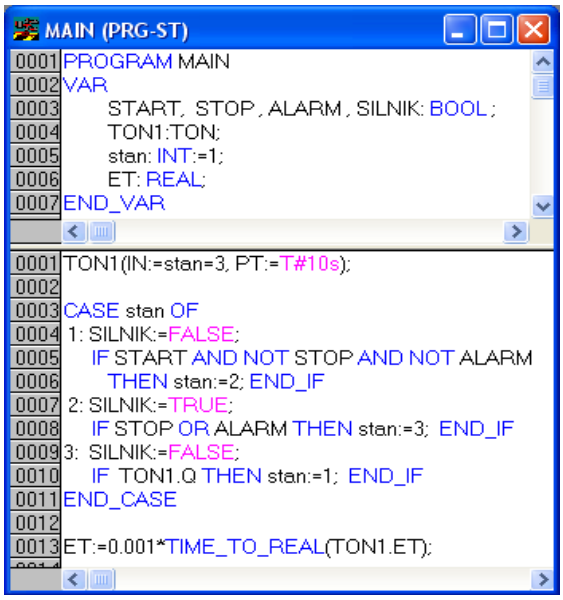
2. Stany

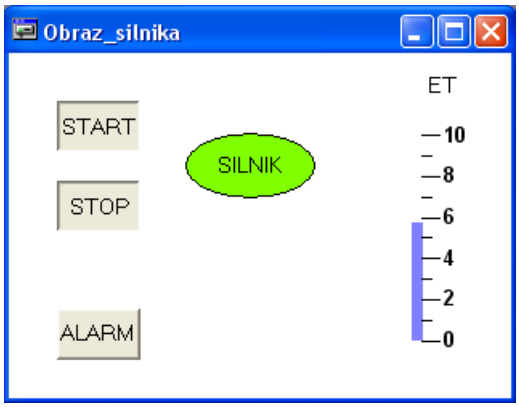
1. silnik wyłączony
2. silnik włączony
3. odmierzanie czasu po *stop* (lub *alarm*)

3. Automat

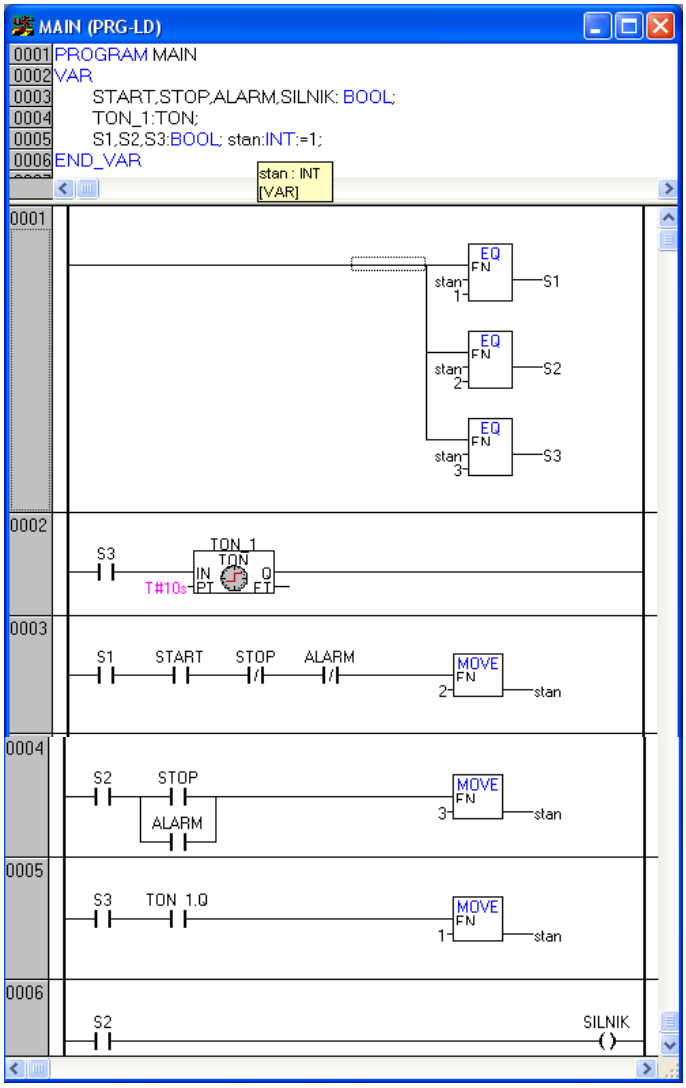


- Kod ST i obraz (*Run*) – *Zabezpieczenie silnika – ST*



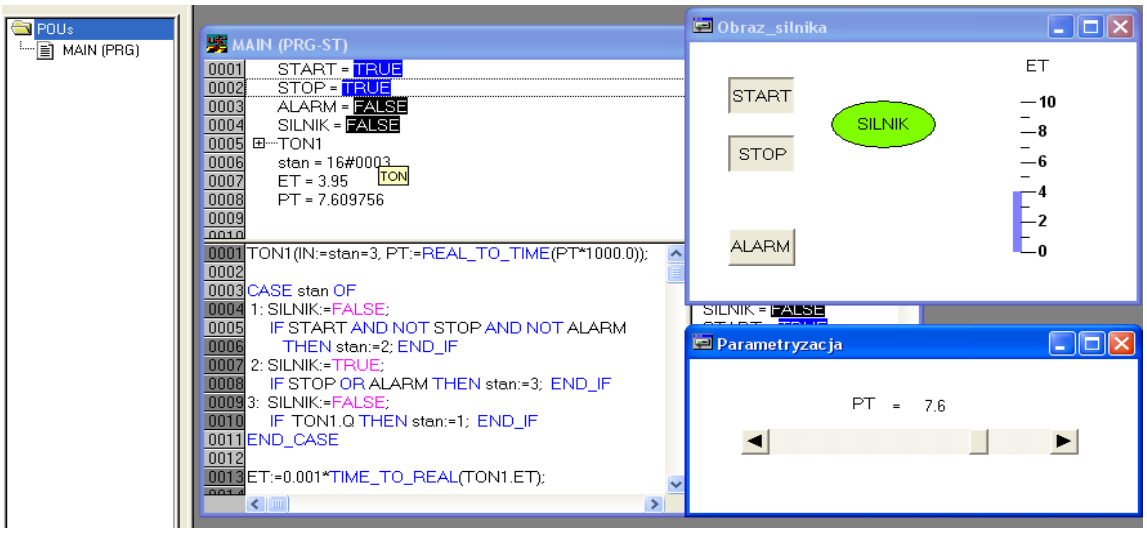


- Kod LD – Zabezpieczenie silnika – LD



4. Parametryzacja *on-line*

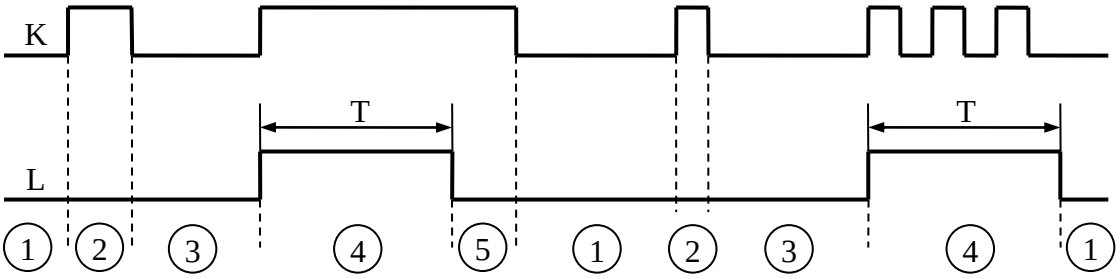
- Chodzi o to, aby podczas pracy układu sterowania, bez wyłączania, można było zmieniać niektóre parametry programu. Funkcjonalność taka jest nazywana zwykle parametryzacją *on-line*.
- Przypuśćmy, że w przypadku zabezpieczenia silnika parametrem *on-line* ma być czas *Preset Time* na wejściu *TON1.PT* czasomierza. Jest on wyrażony w milisekundach. W dodatkowym oknie *Parametryzacja* (zob. niżej) jest ustawiona suwakiem i prezentowana na wskaźniku pomocnicza zmienna *PT* typu REAL reprezentująca sekundy. Po pomnożeniu przez 1000.0 i konwersji *REAL_TO_TIME ( )* staje się ona wejściem czasomierza (linia 0001 kodu ST).



DRUGIE NACIŚNIĘCIE

1. Problem

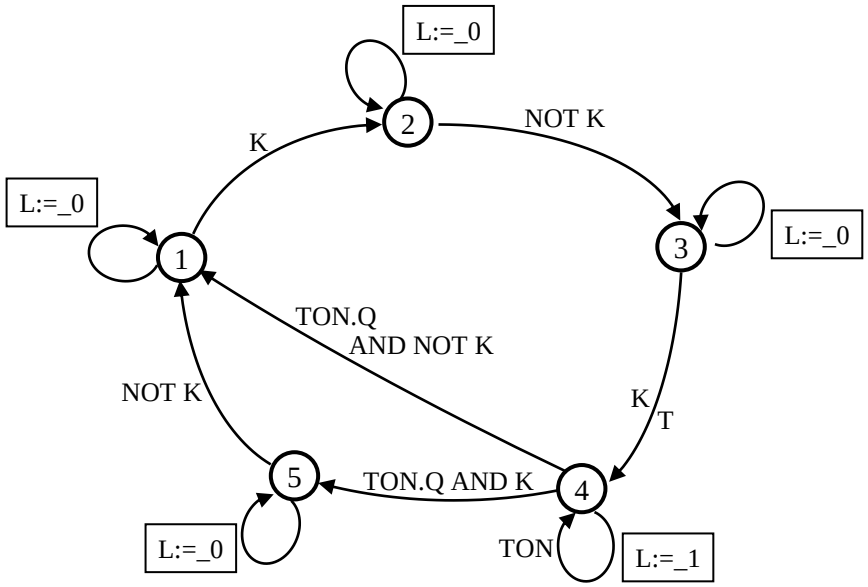
Jeżeli sygnał K pojawił się po raz drugi, wówczas wyjście L zostaje ustawione na czas T. Zanik sygnału K w tym czasie, ani jego zmiany, nie mają znaczenia. Ilustrują to poniższe przebiegi.



2. Stany

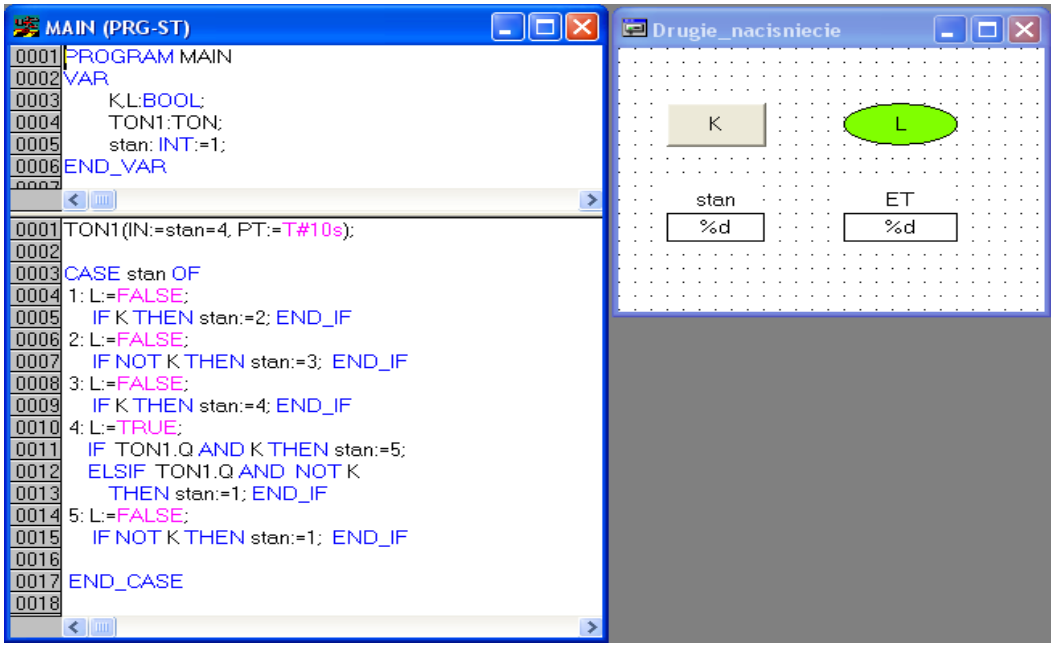
1. oczekiwanie na 1-sze pojawienie się sygnału K (1-sze naciśnięcie)
2. trwa pierwsze pojawienie się sygnału
3. oczekiwanie na 2-gie pojawienie się sygnału
4. odmierzanie czasu T, ustawienie wyjścia L
5. oczekiwanie na zanik sygnału wejściowego, jeżeli był ustawiony na końcu stanu 4

3. Graf automatu

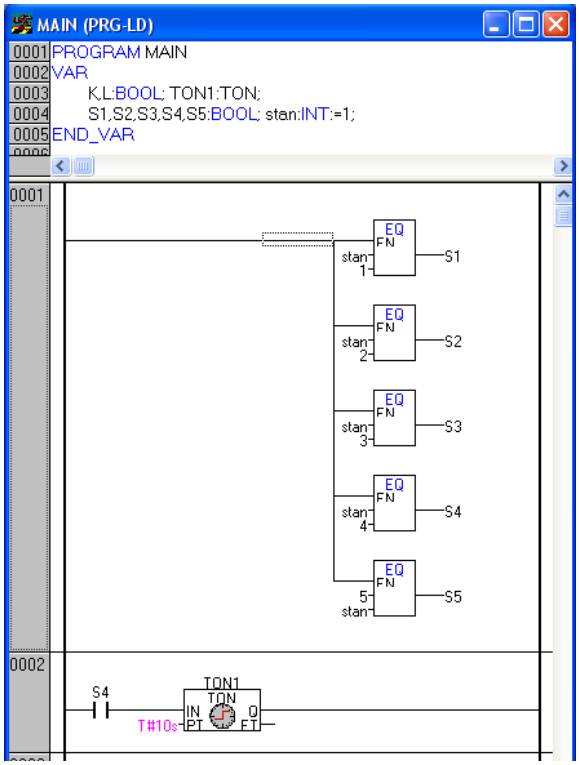


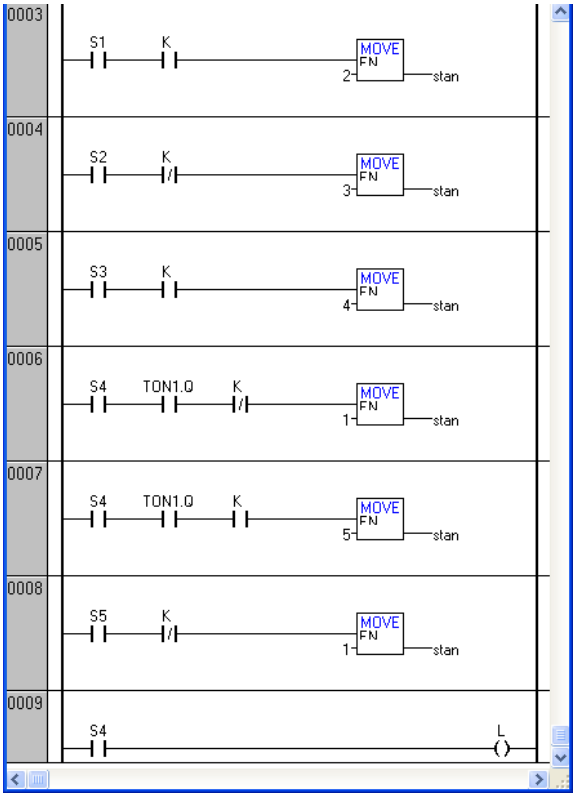
Dane: T = 10s

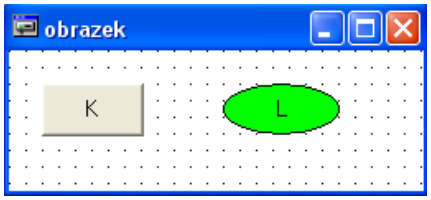
4. Kod ST i obraz – *Drugie naciśnięcie – ST*



5. Kod LD – *Drugie naciśnięcie – LD*





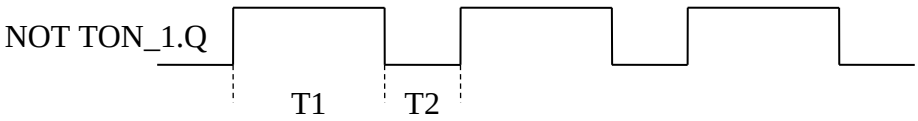


MINIMALNE UKŁADY Z CZASOMIERZAMI

Programy podane niżej są wzorowane na dawniejszych układach przekaźnikowych, gdzie chodziło o minimalną liczbę elementów sprzętowych. Analiza ich działania nie jest jednak tak oczywista, jak w przypadku realizacji automatowych.

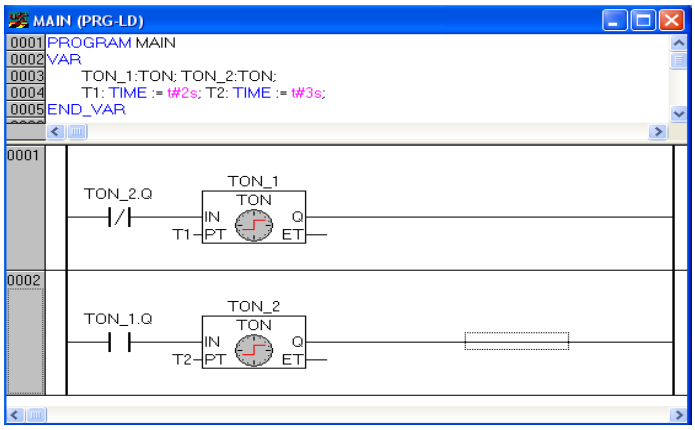
1. Oscylator

- Dwa połączone ze sobą czasomierze generują falę prostokątną o okresie T1+T2. Oscylator taki pracuje nieprzerwanie. Negacja NOT TON_1.Q ma wartość TRUE przez czas T1.

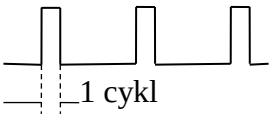


- Kod

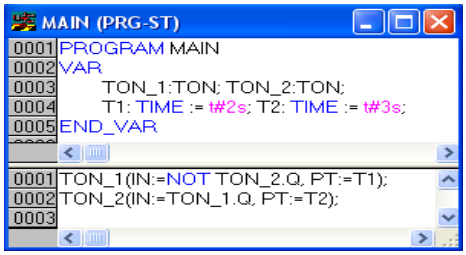
LD



TON_2.Q zmienia się tylko na jeden cykl.



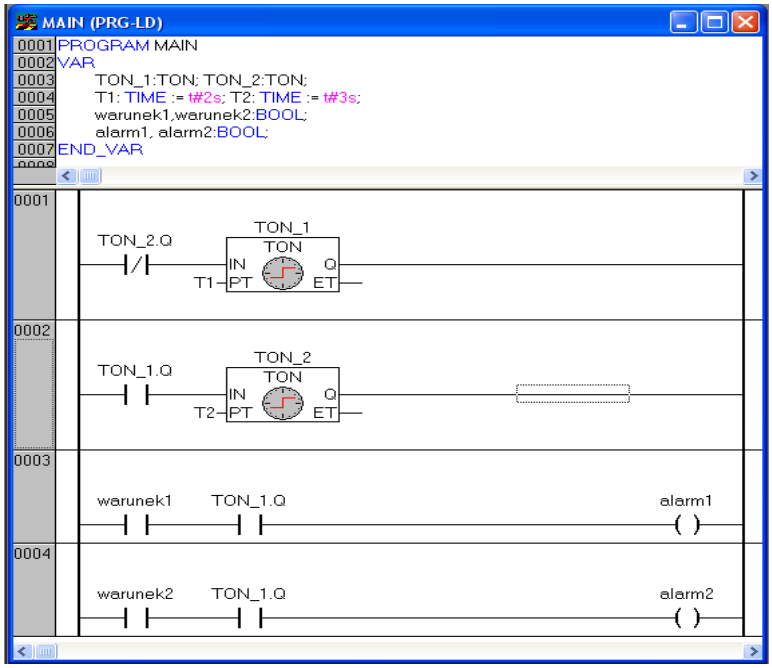
ST



2. Błyskające światła alarmowe

- Oscylator pokazany wyżej można wykorzystać w programie błyskania światłami alarmowymi. Jeżeli warunek jest spełniony (np. przekroczenie dopuszczalnej temperatury), wówczas aktywowany jest odpowiedni alarm. W niektórych sytuacjach bywa potrzebny alarm dźwiękowy.

- LD

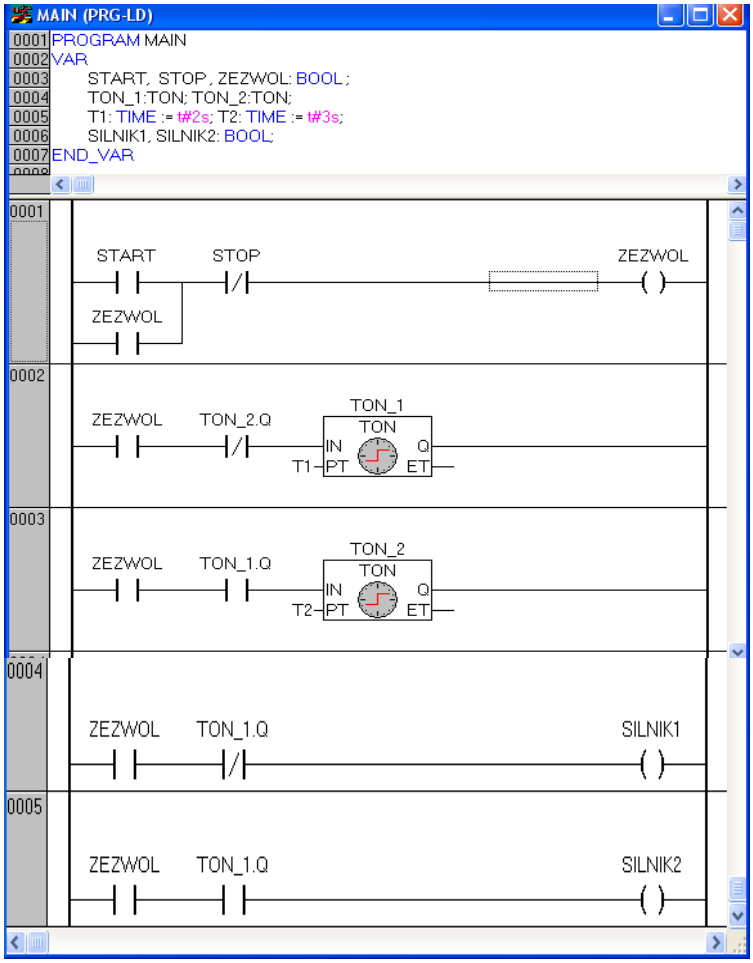


3. Naprzemienne załączanie/wyłączanie urządzeń

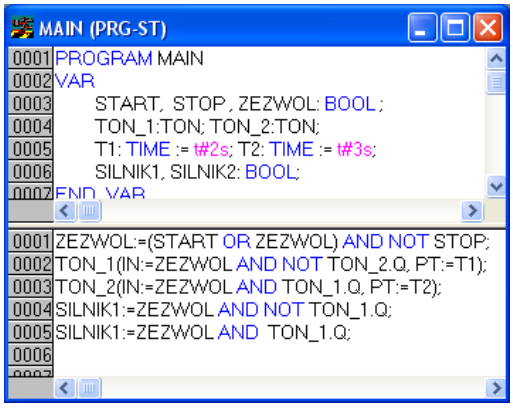
- Jeżeli sygnał *zezwol*(enie) jest ustawiony, wówczas dwa urządzenia zostają na przemian załączone i wyłączone na czasy odpowiednio T1 i T2 (jak poprzednio). Zezwolenie jest ustawione przyciskiem *start*, a kasowane przyciskiem *stop*.

- Kod

  LD



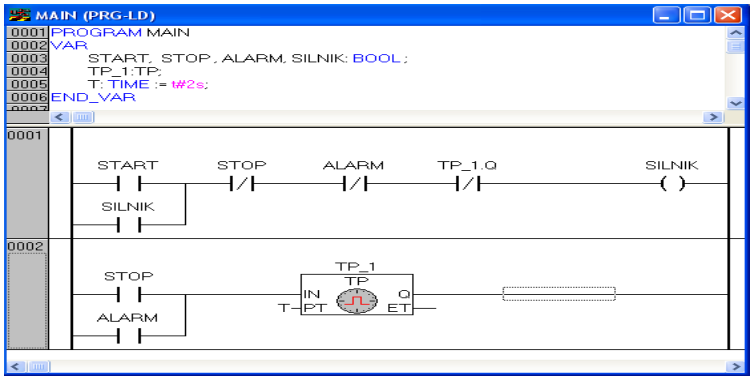
  ST



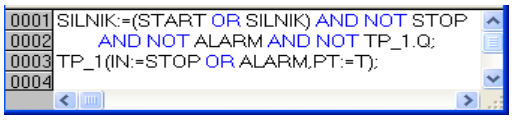
4. Zabezpieczenie silnika przed natychmiastowym ponownym włączeniem

- Poprzednią realizację można uprościć stosując czasomierz TP.
- Kod

LD

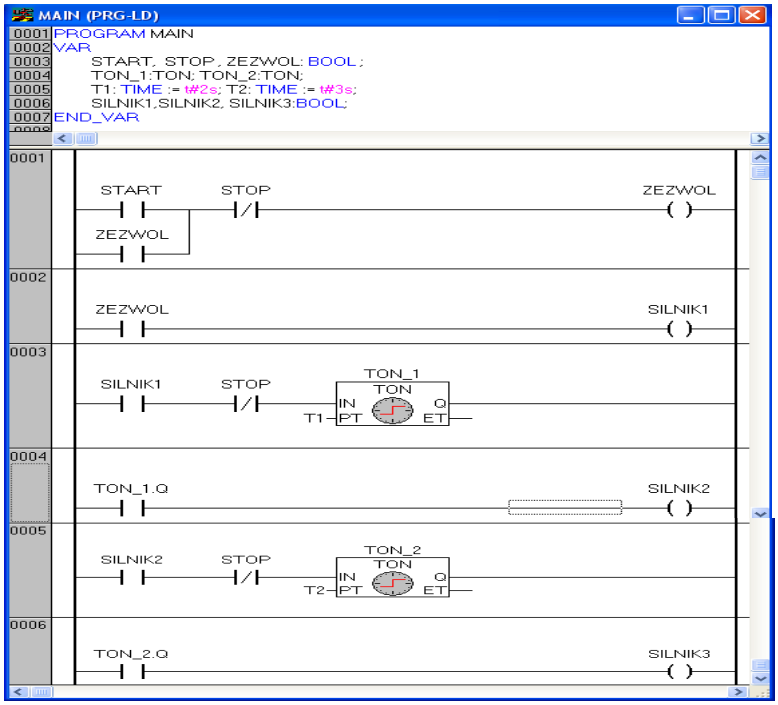


ST



5. Sekwencyjne uruchamianie silników

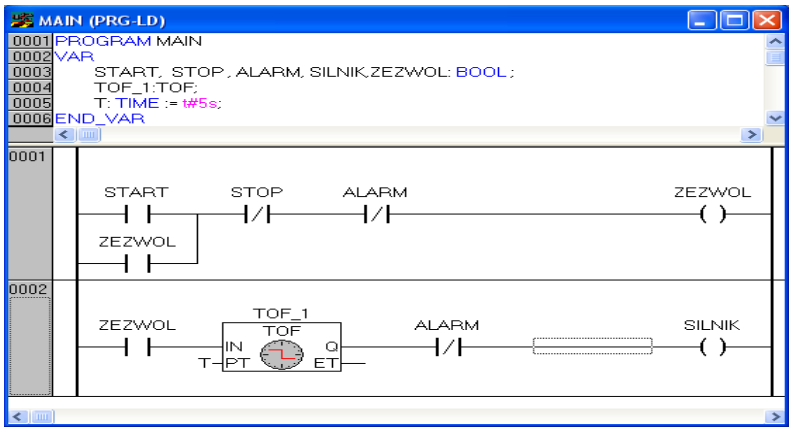
- Jednoczesne załączenie wszystkich silników instalacji technologicznej jest zwykle niewskazane ze względu na przeciążenie rozdzielni zasilającej. W układzie pokazanym niżej pierwszy silnik jest uruchamiany natychmiast po naciśnięciu *start*, drugi po czasie T1, a trzeci po czasie T2 (dla uproszczenia pominięto sygnały *alarm* pochodzące od zabezpieczeń). Sygnał *stop* natychmiast zatrzymuje wszystkie silniki.
- LD



6. Wyłączanie z opóźnieniem

- Silnik jest załączony natychmiast przyciskiem *start*, a wyłączany przyciskiem *stop*, ale dopiero po upływie czasu T. Sygnał *alarm* wyłącza silnik natychmiast. Naturalnym rozwiązaniem jest zastosowanie czasomierza TOF (opóźnione wyłączenie).

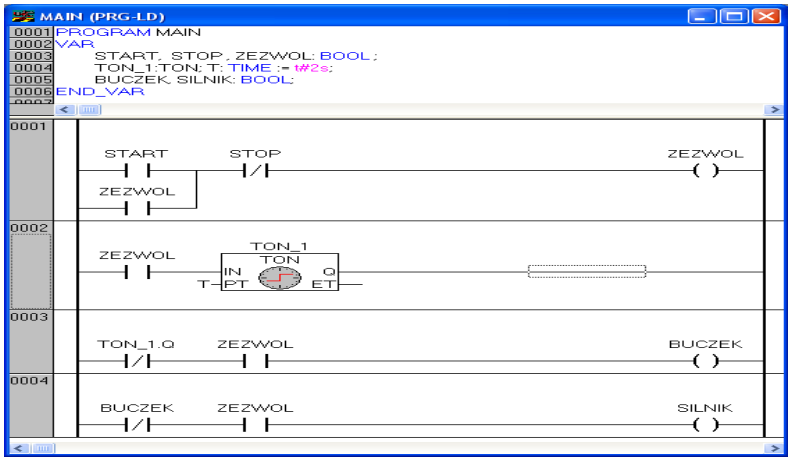
- LD



7. Buczek ostrzegawczy – włączenie z opóźnieniem

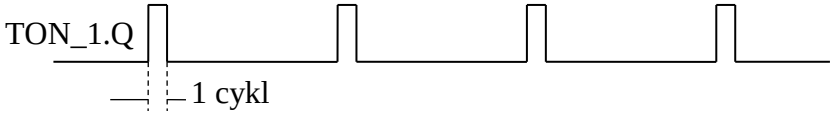
- Buczek ma ostrzegać, że zamierzamy włączyć ruchome urządzenie, np. robot, linię produkcyjną, podajnik wielkogabarytowych elementów, itp. W układzie pokazanym niżej po naciśnięciu *start* najpierw na czas T załączany jest buczek, a dopiero potem silnik uruchamiający urządzenie.

- LD



8. Generator impulsów

- Układ z samoresetującym się czasomierzem TON generuje impulsy trwające jeden cykl.



W poniższym układzie impulsy zlicza licznik CPU (*up–counter*).

- LD