

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*.

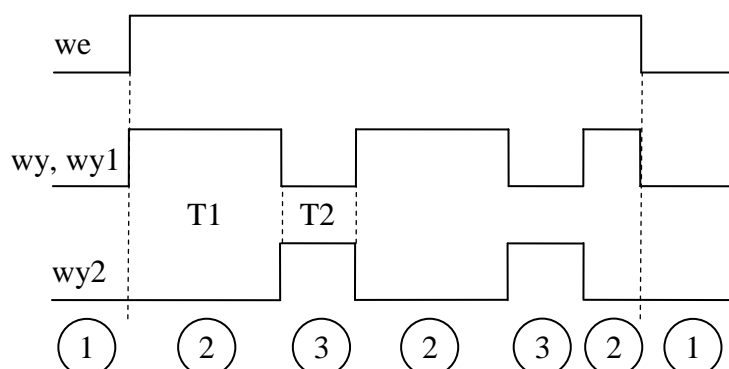
Warunkiem stosowania powyższego rozwiązania jest jednak konieczność zapewnienia stałego cyklu wykonywania programu. W aplikacjach PLC możliwy jest powrót do ponownego wykonania, gdy tylko poprzednie się zakończy (tzw. *PLC mode*), w takim przypadku kolejne czasy wykonywania programu mogą się istotnie różnić. Aby uzyskać stały cykl należy tak skonfigurować system aby zadanie PLC pracowało w tzw. trybie *Constant Sweep* (Beckhoff – Taskattributes typu cyclic z ustaloną wartością Interval – jest to domyślny typ zadania PLC w TwinCAT).

Precyzję odmierzenia 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 T_1 , a drugiego T_2 . Jeżeli sygnał *we* zaniknie, obydwa urządzenia zostają wyłączone.

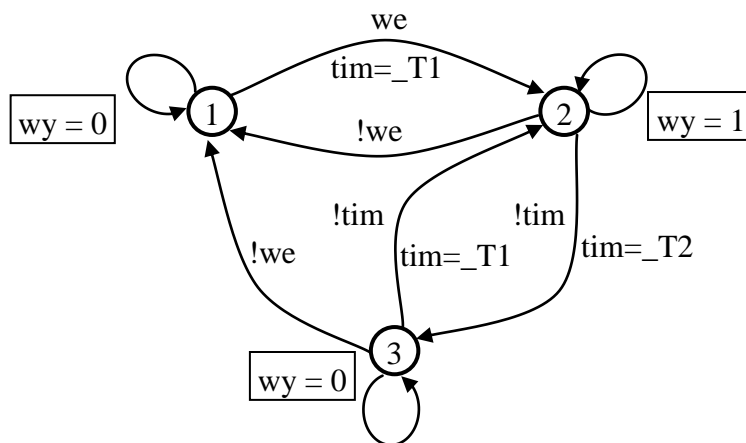


Uwaga. Faktycznie chodzi o warunkowe wytworzenie fali prostokątnej o okresie $T1+T2$.

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



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

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;

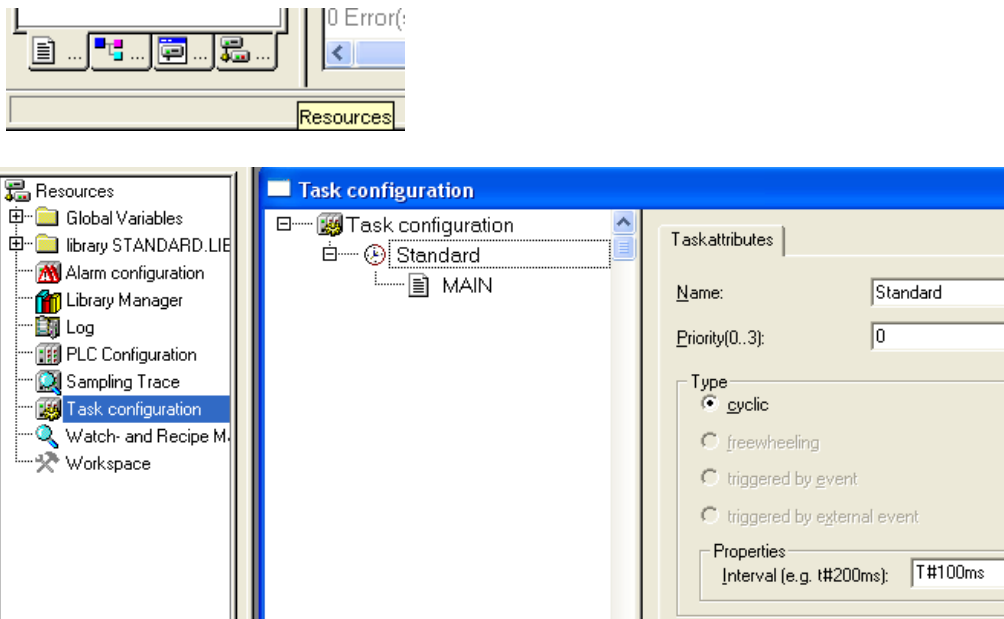
if(tim) --tim;

```

5. Czas cyklu w TwinCAT

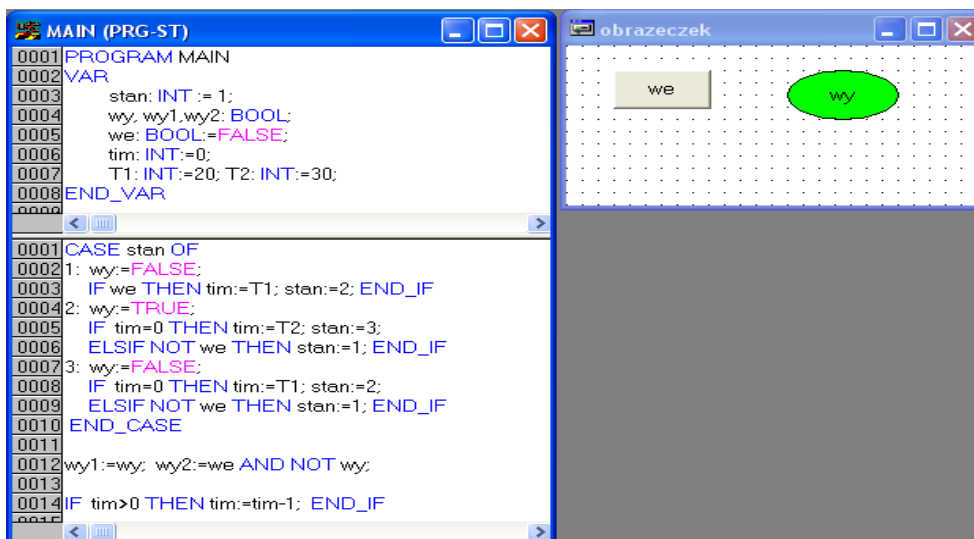
Prawidłowe odmierzenie czasu przez TwinCAT PLC Control uruchomiony na PC wymaga uprzedniego uruchomienia programu *LowPrioProc*.

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*).



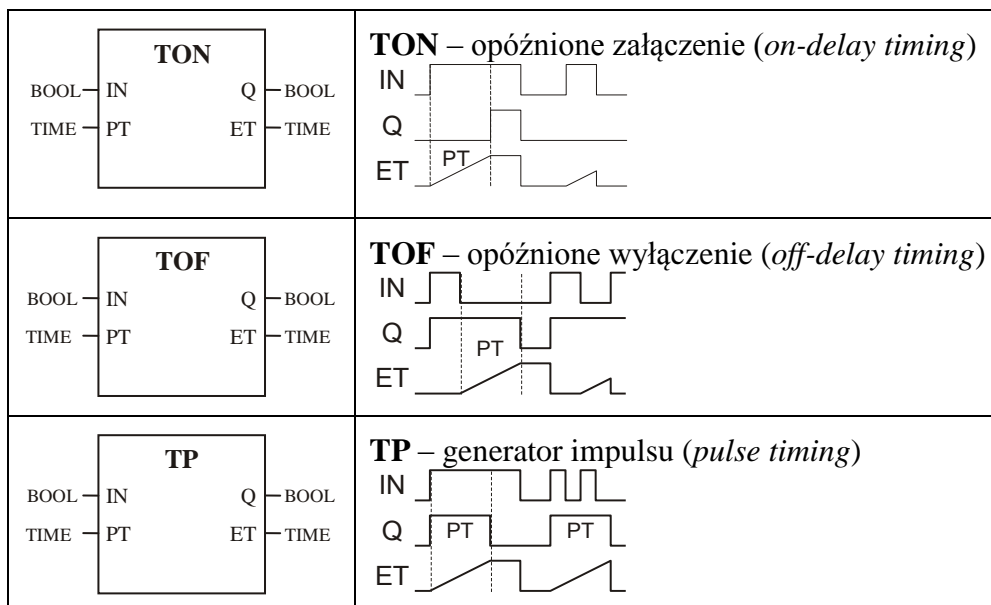
6. Kod ST

Projekt *Fala prostokątna – ST jak C*



STANDARDOWE CZASOMIERZE

1. Czasomierze normy PN/EN–61131-3



Wejścia:

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

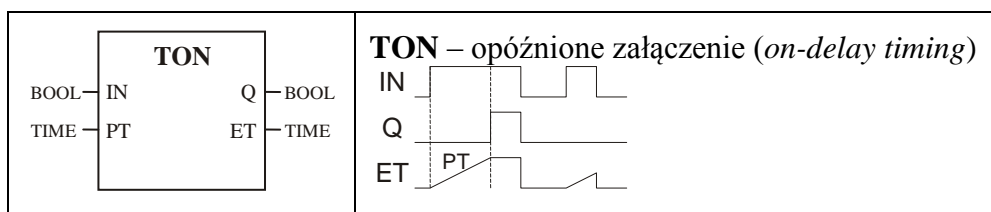
Wyjścia:

- 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 czasomierza 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.



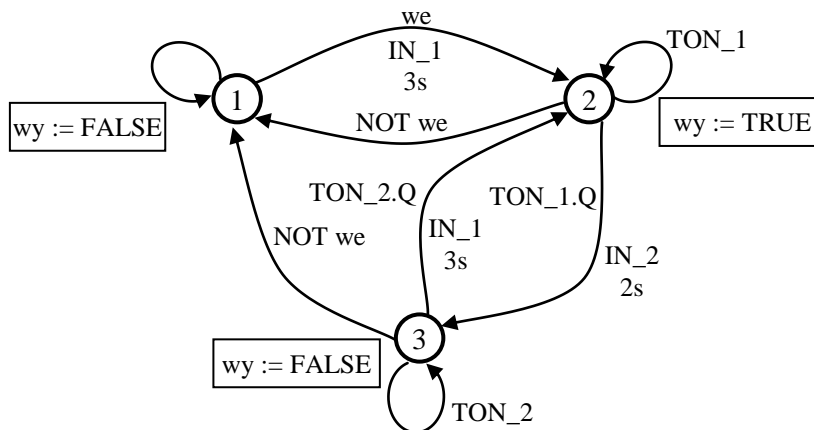
Zasady zastosowania czasomierza TON w programach sekwencyjno–czasowych są następujące:

- 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).
- Ustawienie IN na FALSE w stanie 1 (lub w inicjacji).

3. Fala prostokątna z czasomierzami TON

- Automat

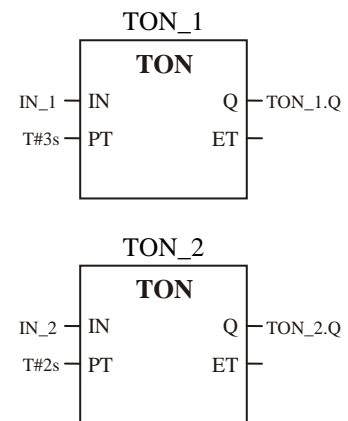


- Kod ST

```

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   IN_1, IN_2: BOOL;
0008 END_VAR
0009
0001 TON_1(IN:=IN_1, PT:=T#3s);
0002 TON_2(IN:=IN_2, PT:=T#2s);
0003
0004 CASE stan OF
0005 1: wy:=FALSE;
0006   IN_1:=FALSE; IN_2:=FALSE;
0007 IF we THEN IN_1:=TRUE; stan:=2; END_IF
0008 2: wy:=TRUE;
0009 IF TON_1.Q THEN IN_2:=TRUE; IN_1:=FALSE; stan:=3;
0010 ELSIF NOT we THEN stan:=1; END_IF
0011 3: wy:=FALSE;
0012 IF TON_2.Q THEN IN_1:=TRUE; IN_2:=FALSE; stan:=2;
0013 ELSIF NOT we THEN stan:=1; END_IF
0014 END_CASE
0015
0016 wy1:=wy; wy2:=we AND NOT wy;

```



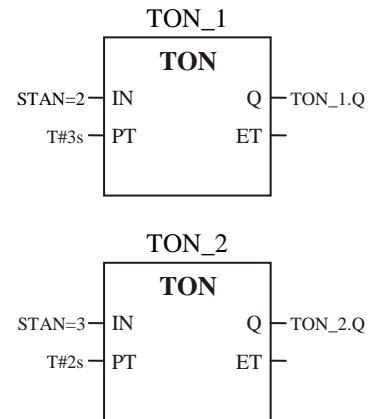
Fala prostokątna – TON - IN

Uwaga. Ponieważ czasomierze TON_1, TON_2 mają wejścia ustawiane na TRUE odpowiednio w stanach 2 i 3, więc przełączanie wejść IN_1, IN_2 można zastąpić podstawieniami z wyrażeniami logicznymi IN_1:=STAN=2 i IN_2:=STAN=3. Powrót na FALSE nastąpi wtedy automatycznie.

```

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
0008
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 – TON - STAN

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

4. Praca układu

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

FALA PROSTOKĄTNA W LD

1. Konwersja ST → LD – zasady

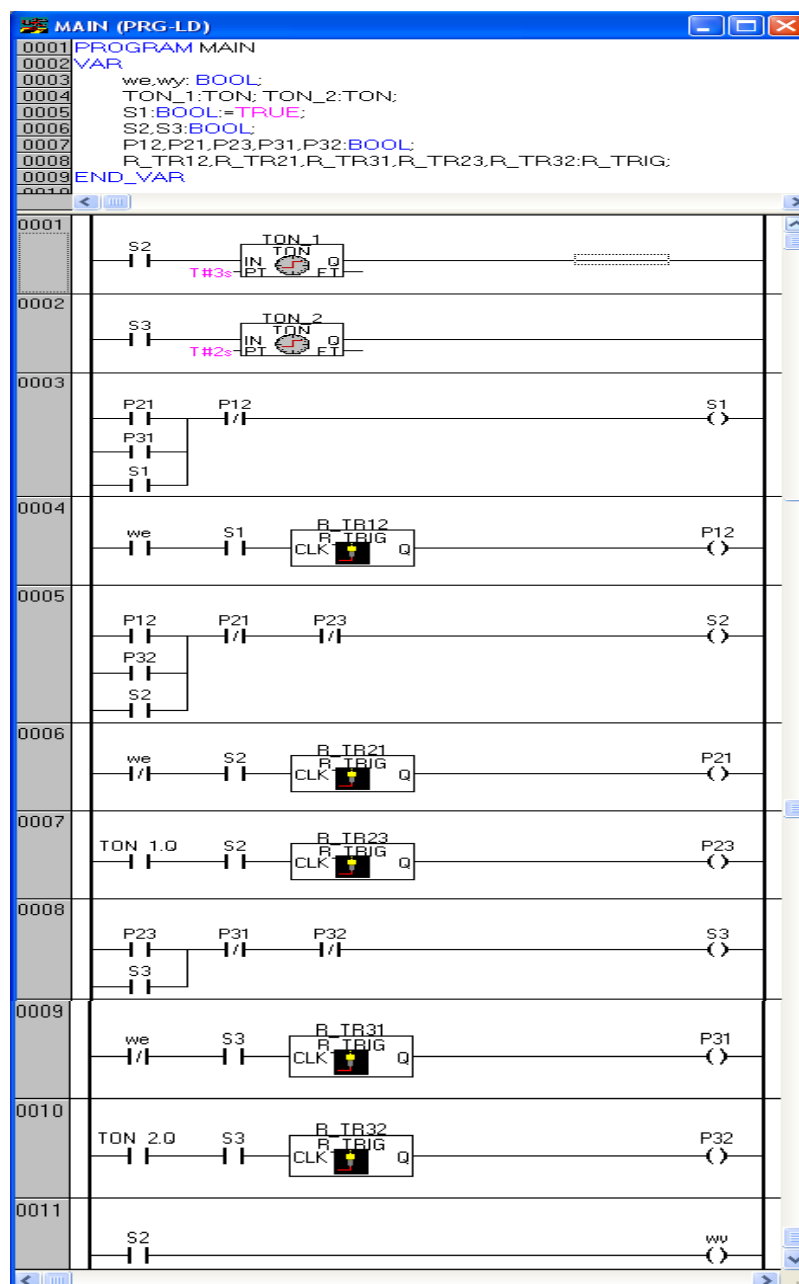
- Szczeble czasomierzy aktywowanych stykami stanów S_i rozpoczynają program.
- Ogólna struktura programu jest taka jak w układach sekwencyjnych, tzn.:
 - szczebel stanu S_i jest układem z podtrzymaniem aktywowanym przejściami dochodzącymi, a zerowanymi przejściami wychodzącymi,

- szczebel przejścia P_{ij} zawiera połączone szeregowo: warunek przejścia, stan S_i , z którego przejście wychodzi oraz blok R_TRIG generujący jednorazowy impuls,
- wyjście jest równe logicznej sumie stanów S_i , w których ma być ustawione na TRUE (w przypadku zerowania FALSE byłyby to iloczyn zanegowanych styków S_i).

Uwagi. Indeksy i, j przejścia P_{ij} są odpowiednio numerami stanu początkowego i oraz końcowego j . Na przykład P_{23} oznacza przejście ze stanu 2 do stanu 3.

Realizacja poprzez konwersję ST \rightarrow LD nie jest realizacją minimalną, tzn. niektóre zmienne, 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.

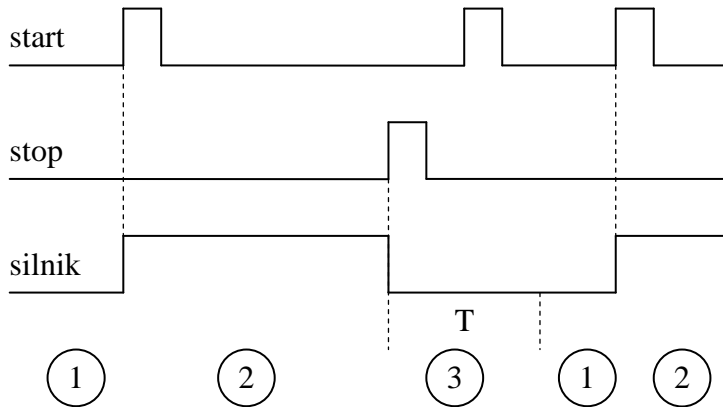
2. Kod LD



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*.

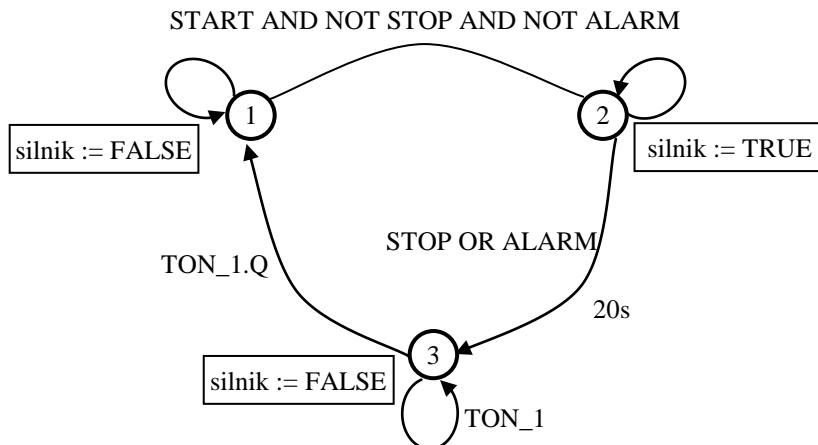


2. Stany

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

3. Automat

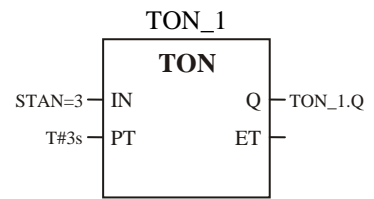
- Automat



- Kod ST

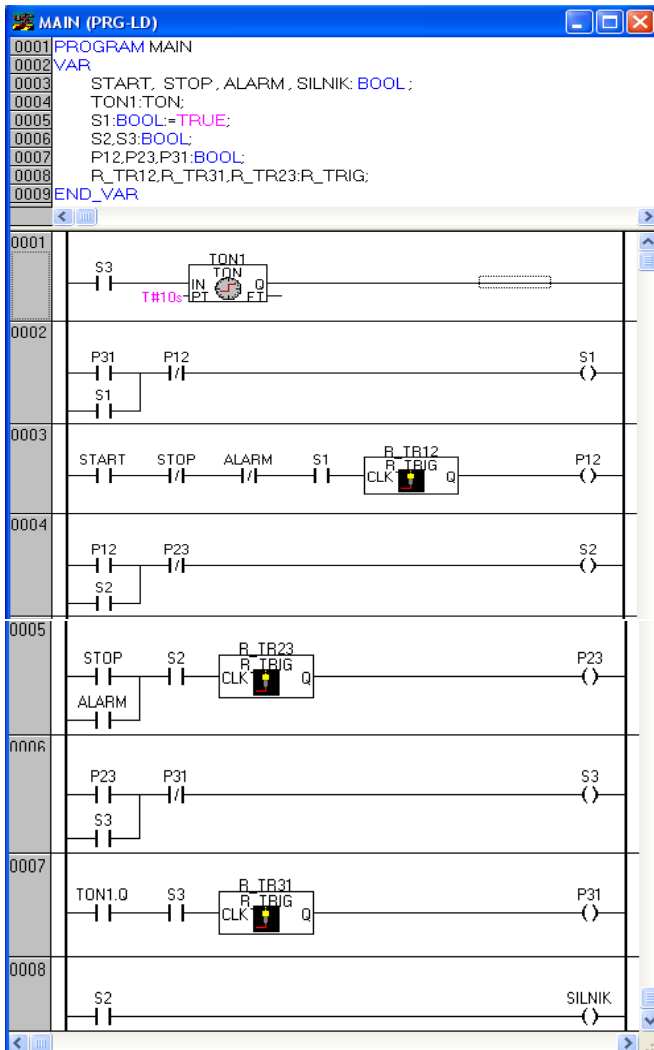
```

MAIN (PRG-ST)
0001 PROGRAM MAIN
0002 VAR
0003     START, STOP, ALARM, SILNIK: BOOL;
0004     TON_1:TON;
0005     stan:INT:=1;
0006 END_VAR
0007
0001 TON_1(IN:=stan=3, PT:=T#10s);
0002
0003 CASE stan OF
0004 1: SILNIK:=FALSE;
0005     IF START AND NOT STOP AND NOT ALARM
0006     THEN stan:=2; END_IF
0007 2: SILNIK:=TRUE;
0008     IF STOP OR ALARM THEN stan:=3; END_IF
0009 3: SILNIK:=FALSE;
0010     IF TON_1.Q THEN stan:=1; END_IF
0011 END_CASE
  
```



Zabezpieczenie silnika – STAN

- Kod LD

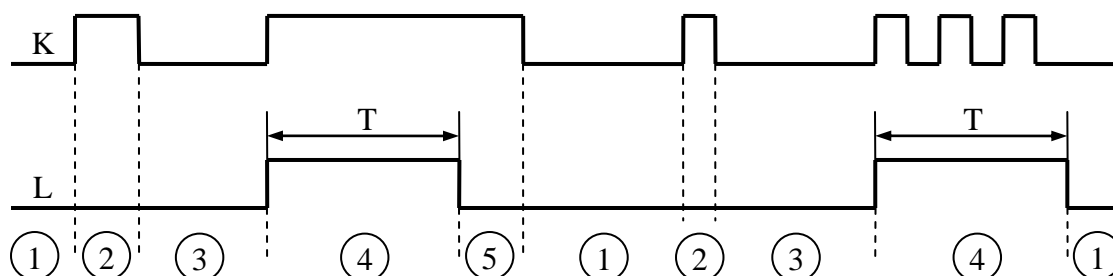


Zabezpieczenie silnika – LD

DRUGIE NACIŚNIĘCIE

1. Problem

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

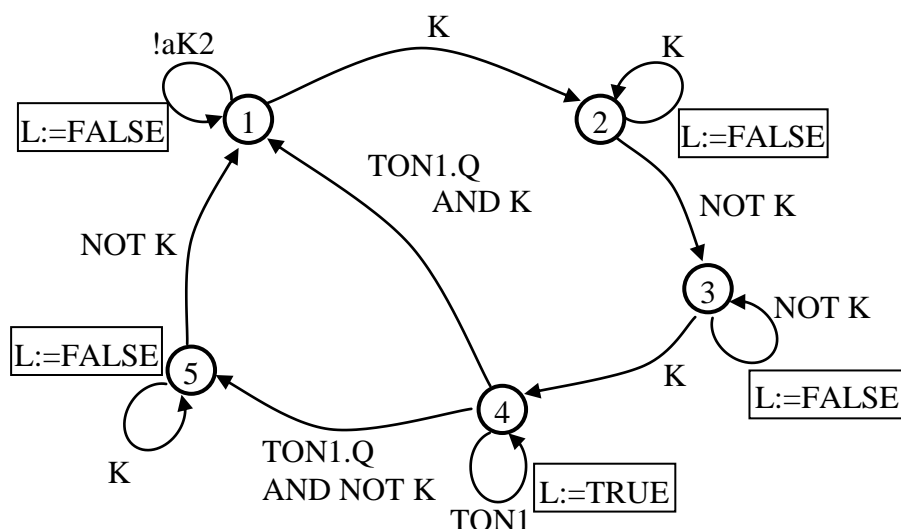


Dane: $T = 3s$

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 odmierzenie czasu T, ustawienie wyjścia L
- ⑤ oczekiwanie na zanik sygnału wejściowego, jeżeli był ustawiony na końcu stanu 4

3. Graf automatu

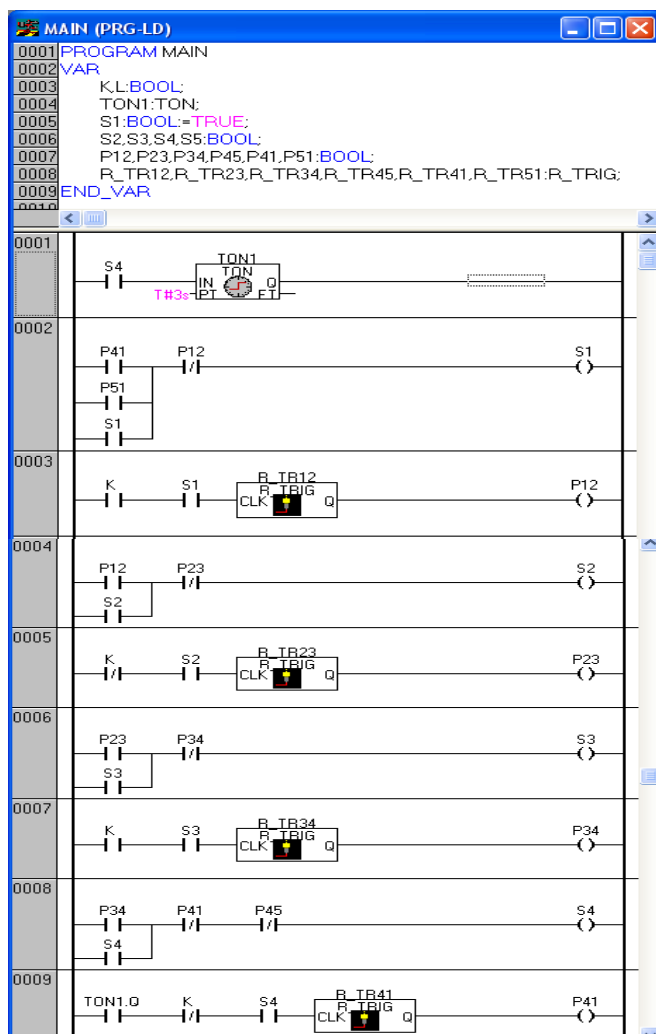


4. Kod ST

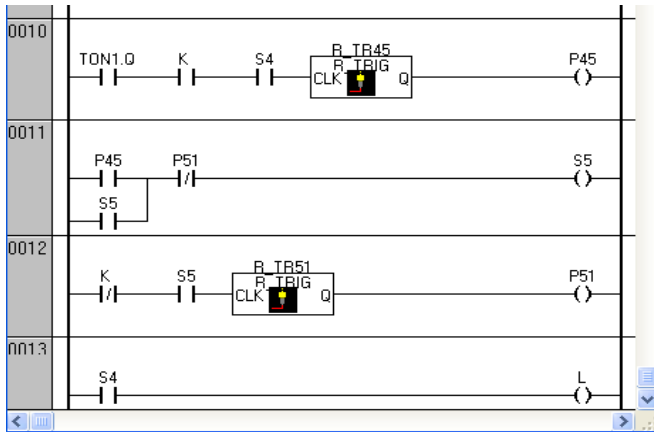
```
MAIN (PRG-ST)
0001 PROGRAM MAIN
0002 VAR
0003   K:L:BOOL;
0004   TON1:TON;
0005   stan:INT:=1;
0006 END_VAR
0007
0008 TON1(IN:=stan=4, PT:=T#10s);
0009
0010 CASE stan OF
0011 1: L:=FALSE;
0012   IF K THEN stan:=2; END_IF
0013 2: L:=FALSE;
0014   IF NOT K THEN stan:=3; END_IF
0015 3: L:=FALSE;
0016   IF K THEN stan:=4; END_IF
0017 4: L:=TRUE;
0018   IF TON1.Q AND K THEN stan:=5;
0019   ELSIF TON1.Q AND NOT K
0020   THEN stan:=1; END_IF
0021 5: L:=FALSE;
0022   IF NOT K THEN stan:=1; END_IF
0023 END_CASE
```

Drugie naciśnięcie – ST

5. Kod LD



Drugie naciśnięcie – LD

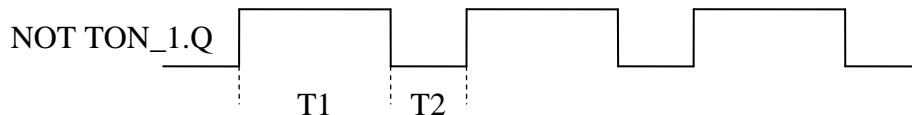


MINIMALNE UKŁADY Z CZASOMIERZAMI

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.

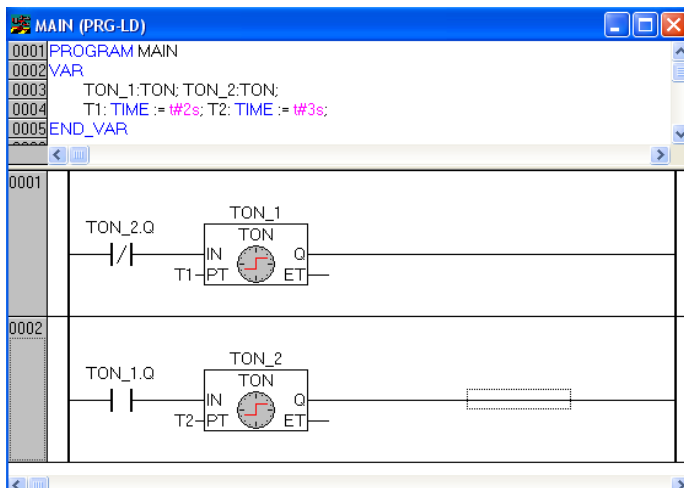
1. Oscylator

- 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 $T1$.

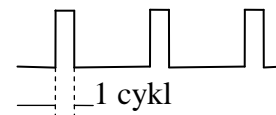


- Kod

LD



TON_2.Q zmienia się tylko na jeden cykl.

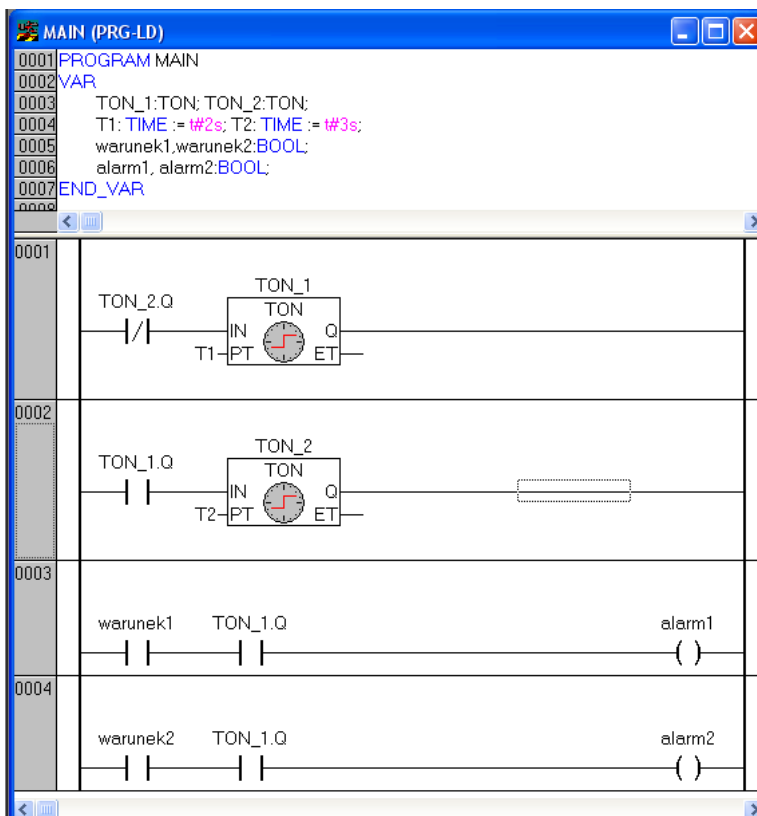


ST

```
MAIN (PRG-ST)
0001 PROGRAM MAIN
0002 VAR
0003     TON_1:TON; TON_2:TON;
0004     T1: TIME := #2s; T2: TIME := #3s;
0005 END_VAR
0001 TON_1(IN:=NOT TON_2.Q, PT:=T1);
0002 TON_2(IN:=TON_1.Q, PT:=T2);
0003
```

2. Blyskające 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

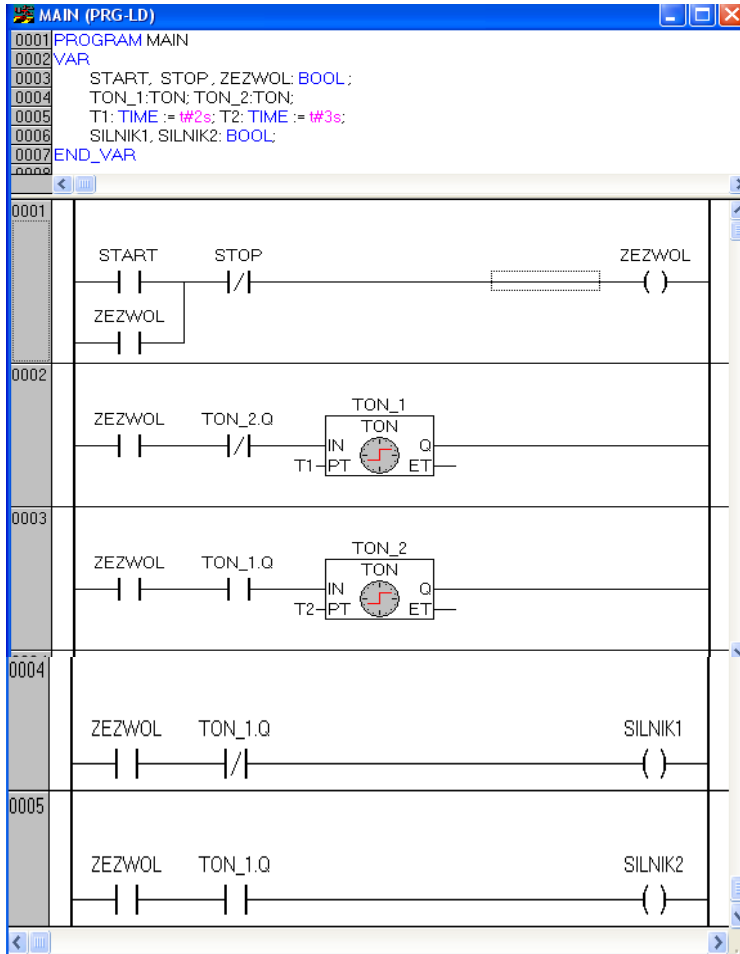


3. Naprzemienne załączanie/wyłączanie urządzeń

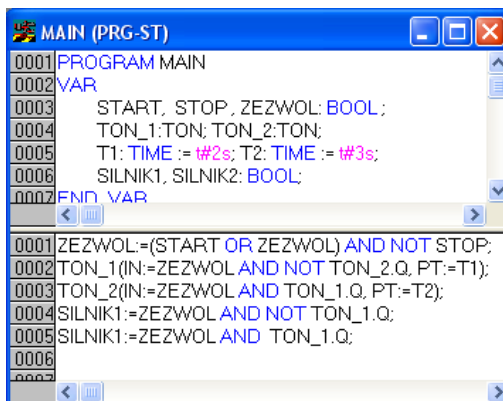
- Jeżeli sygnał *zezwoł(enie)* jest ustawiony, wówczas dwa urządzenia zostają na przemian załączone i wyłączone na czasy odpowiednio T1 i T2 (jak poprzednio). Zezwolenie jest ustawione przyciskiem *start*, a kasowane przyciskiem *stop*.

- Kod

LD



ST

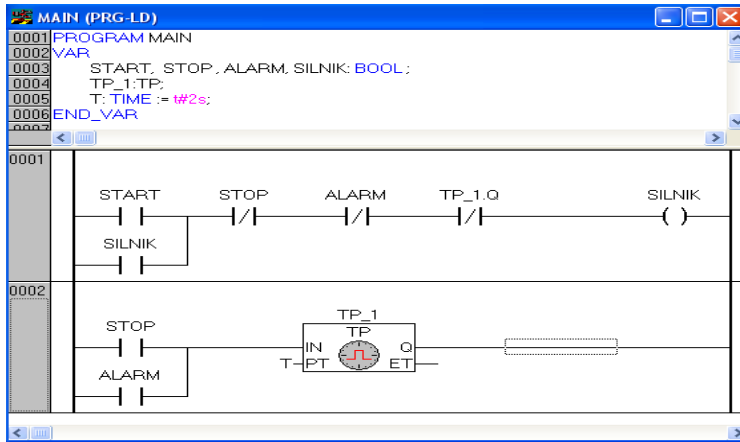


4. Zabezpieczenie silnika przed natychmiastowym ponownym włączeniem

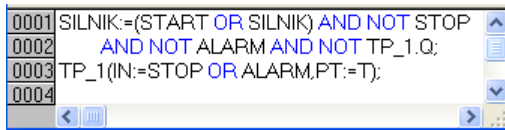
- Poprzednia realizacja wykorzystująca licznik *tim* jest niżej zastąpiona układem z czasomierzem TP.

- Kod

LD



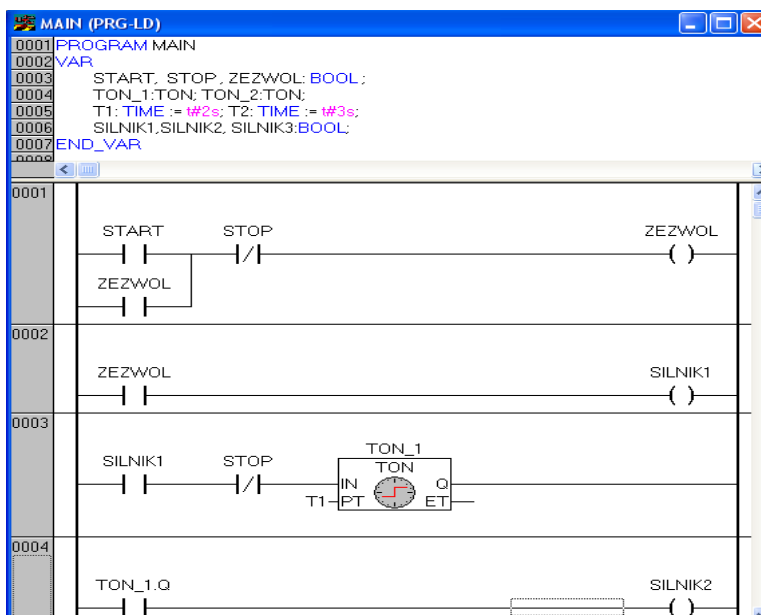
ST

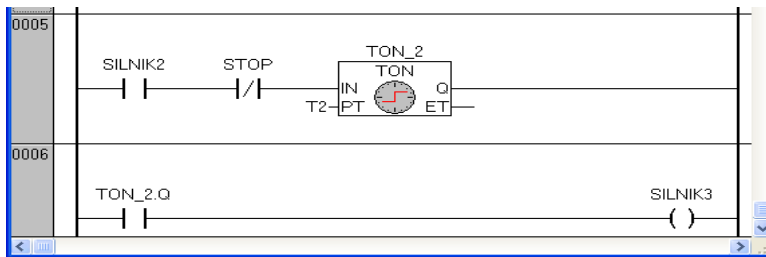


5. Sekwencyjne uruchamianie silników

- 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.

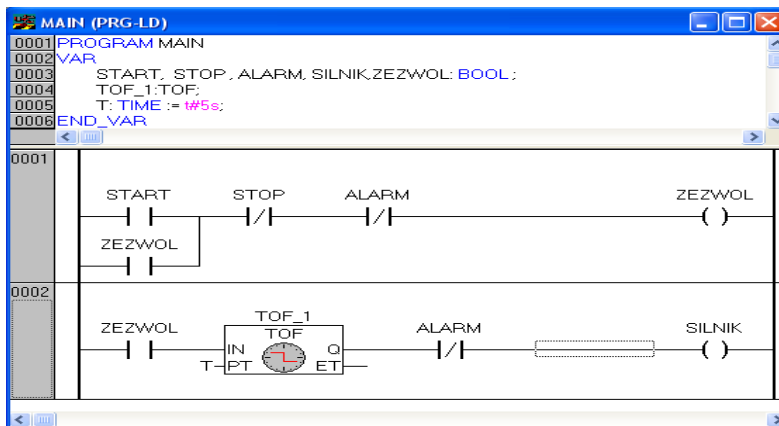
- LD





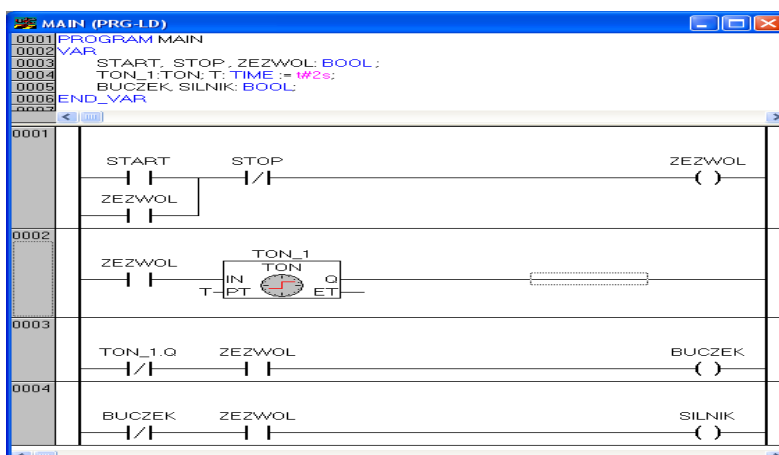
6. Wyłączanie z opóźnieniem

- 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



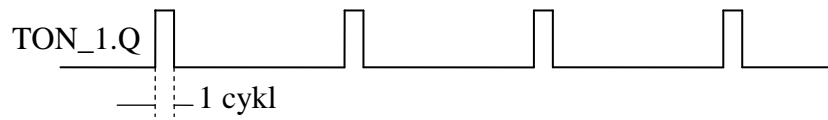
7. Buczek ostrzegawczy – włączenie z opóźnieniem

- 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 buczonek, 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

