

# Sprawozdanie 4

Jan Bronicki 249011

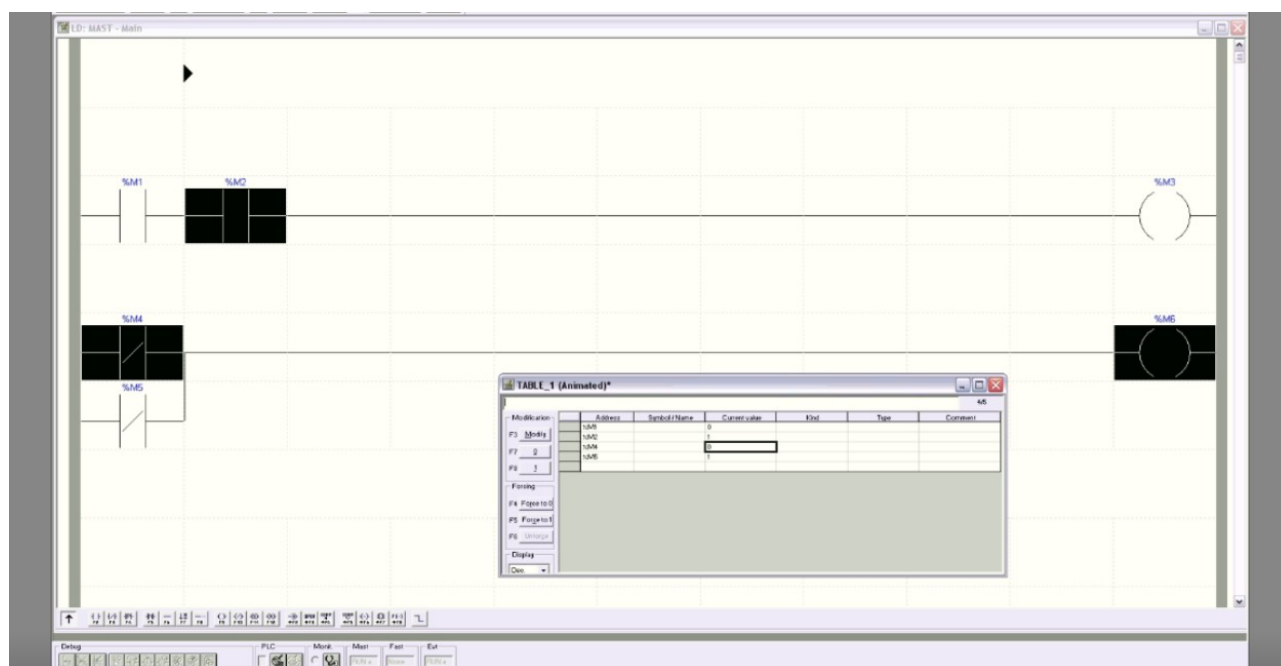
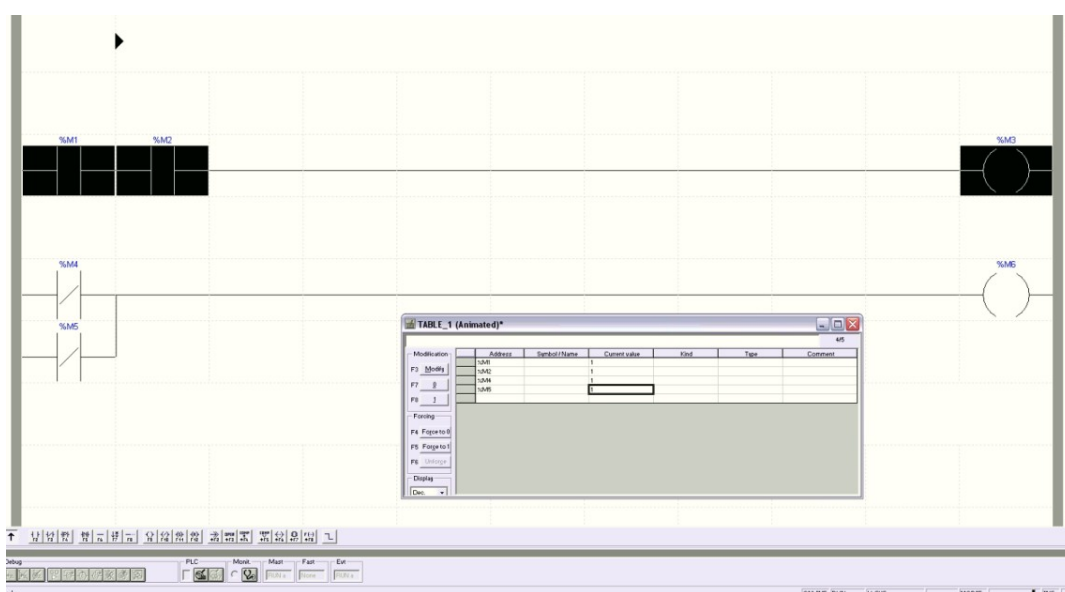
## 1 Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z językiem programowania LADDER oraz jego podstawowymi działaniami.

## 2 Schematy programów

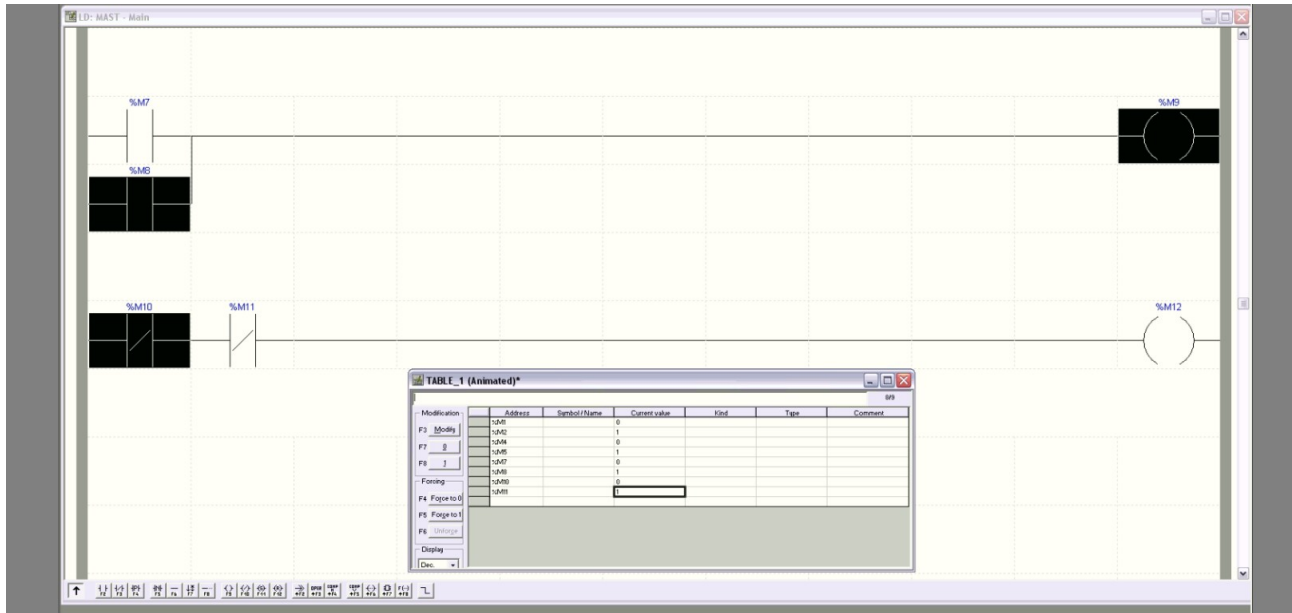
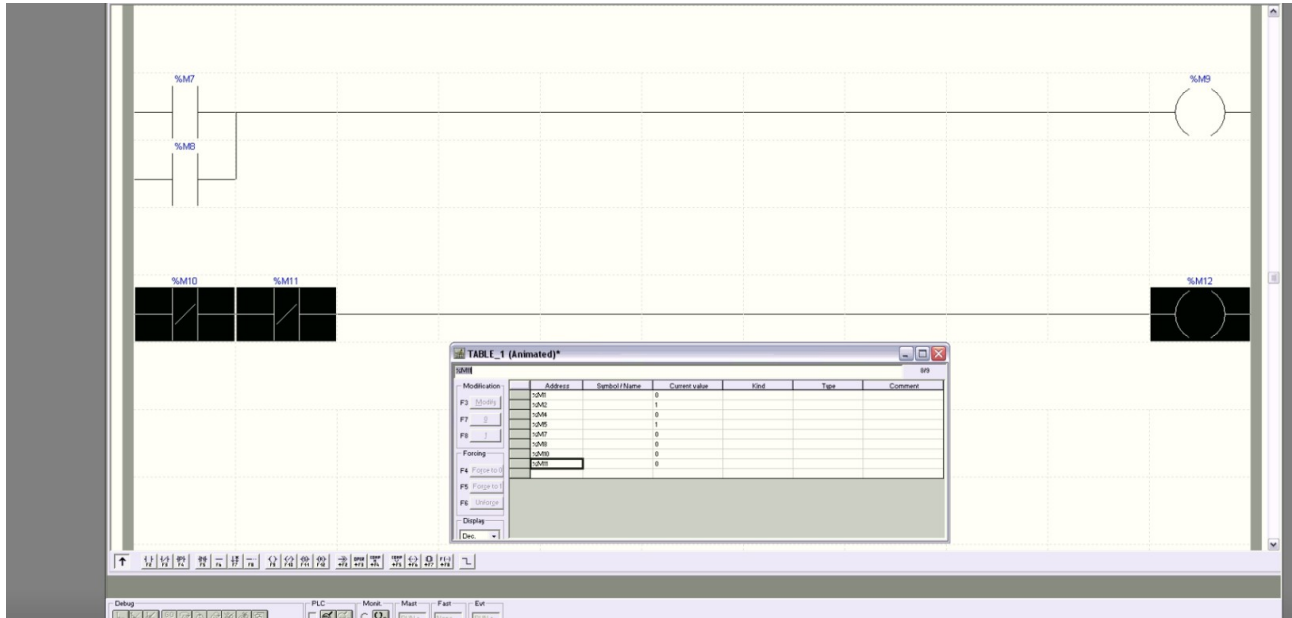
## 2.1 Program 1

## Funkcje AND I NAND



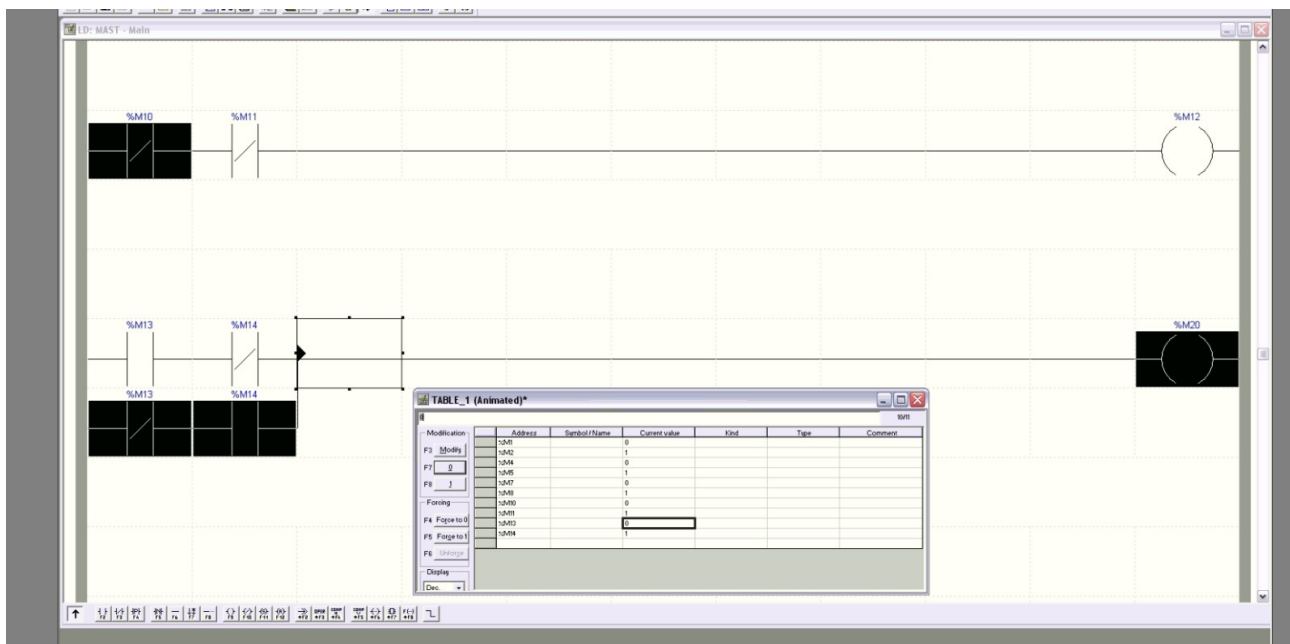
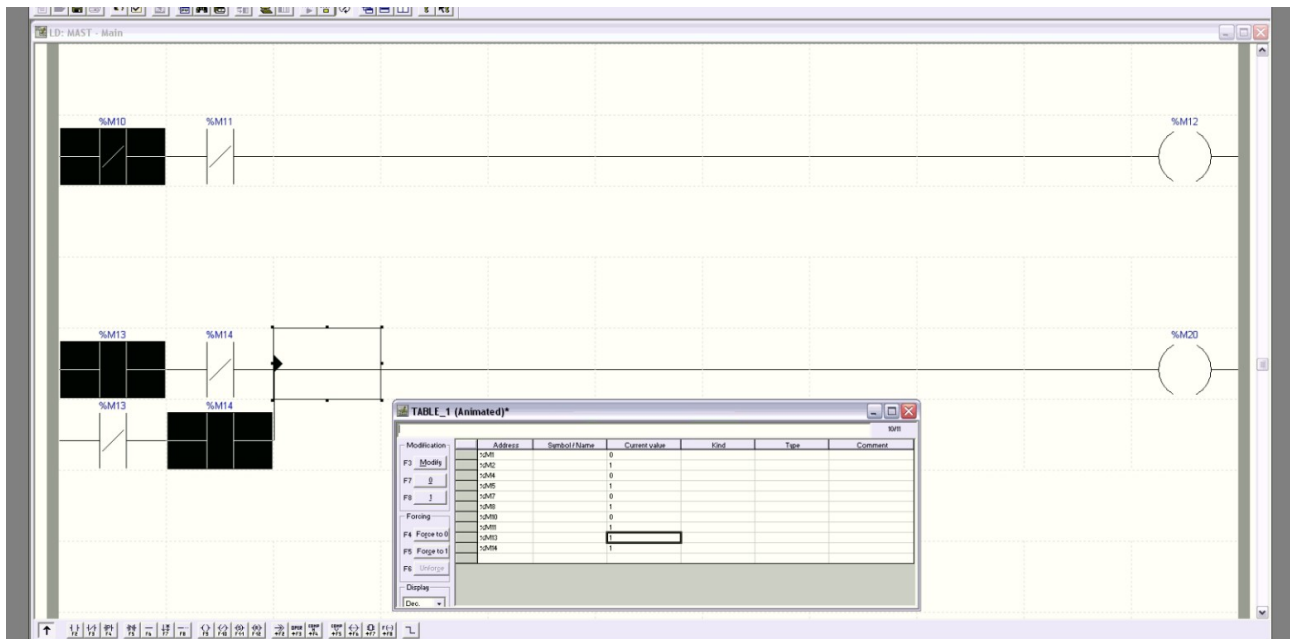
Na zdjęciach powyżej znajdują się następująco funkcje AND oraz NAND. W AND oba bity muszą być ustawione na 1 aby na wyjściu pojawiła się 1. W funkcji NAND tylko jeden z bitów może być ustawiony na 1, aby na wyjściu pojawiła się 1.

## Funkcje OR oraz NOR



Wyżej jest funkcja OR a niżej NOR. W funkcji OR wystarczy, aby 1 bit był ustawiony na 1, aby na wyjściu pojawiła się 1. W funkcji NOR oba bity muszą być ustawione na 1, aby na wyjściu pojawiła się 1.

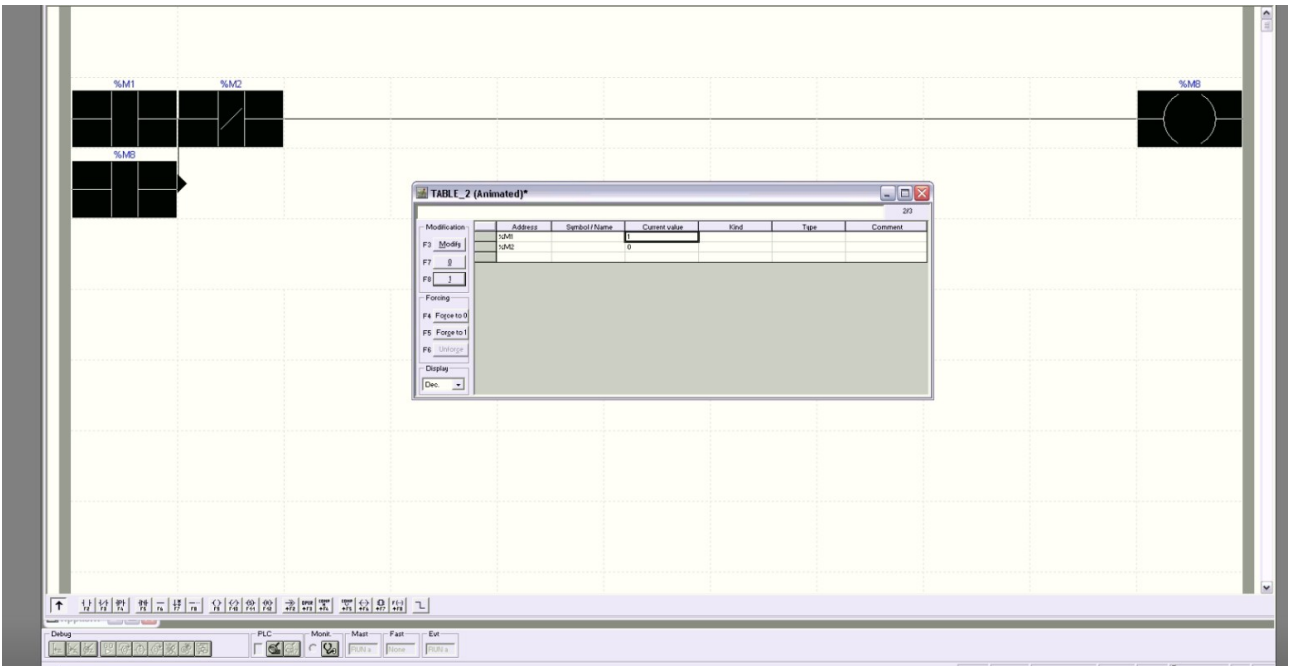
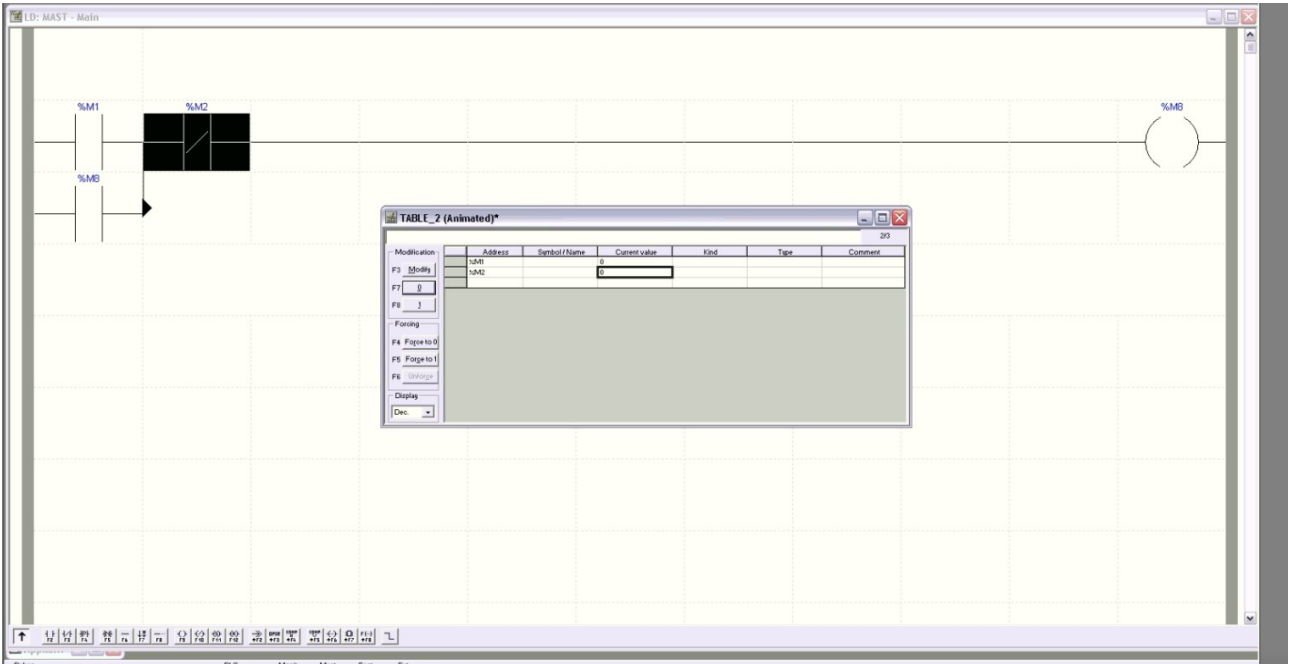
## Funkcja EXOR

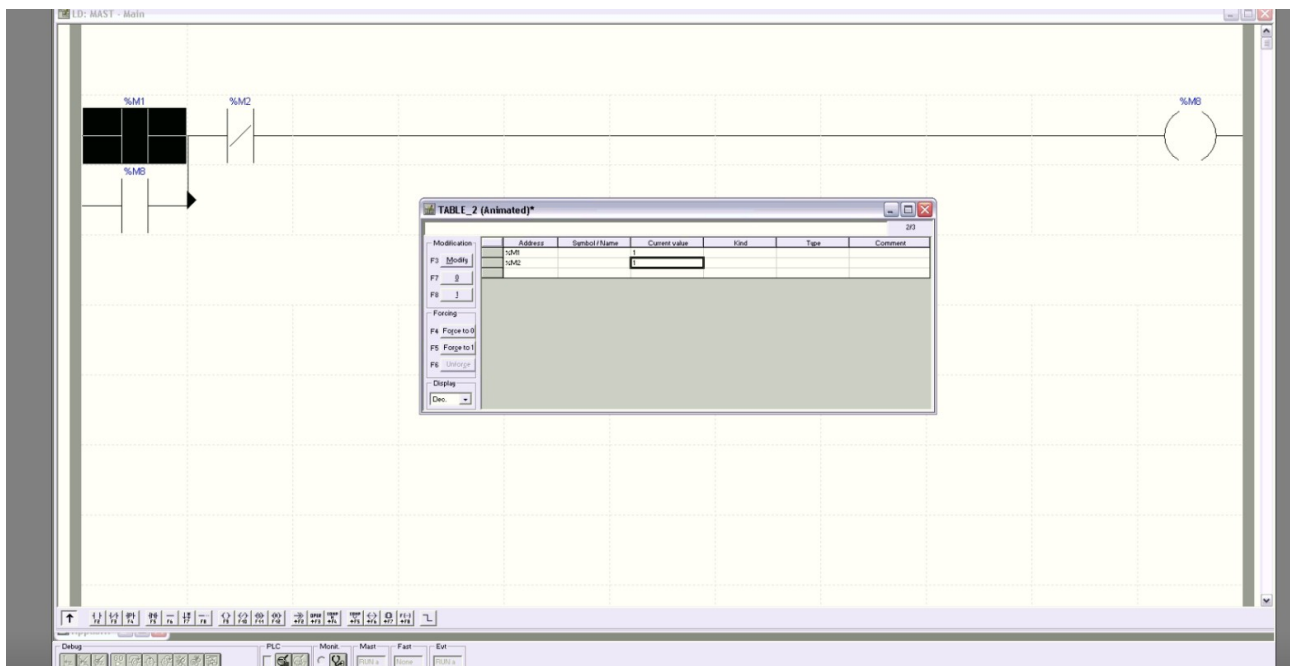


na powyższej funkcji musimy ustawić różne wartości bitów wejściowych, aby na wyjściu pojawiła się 1.

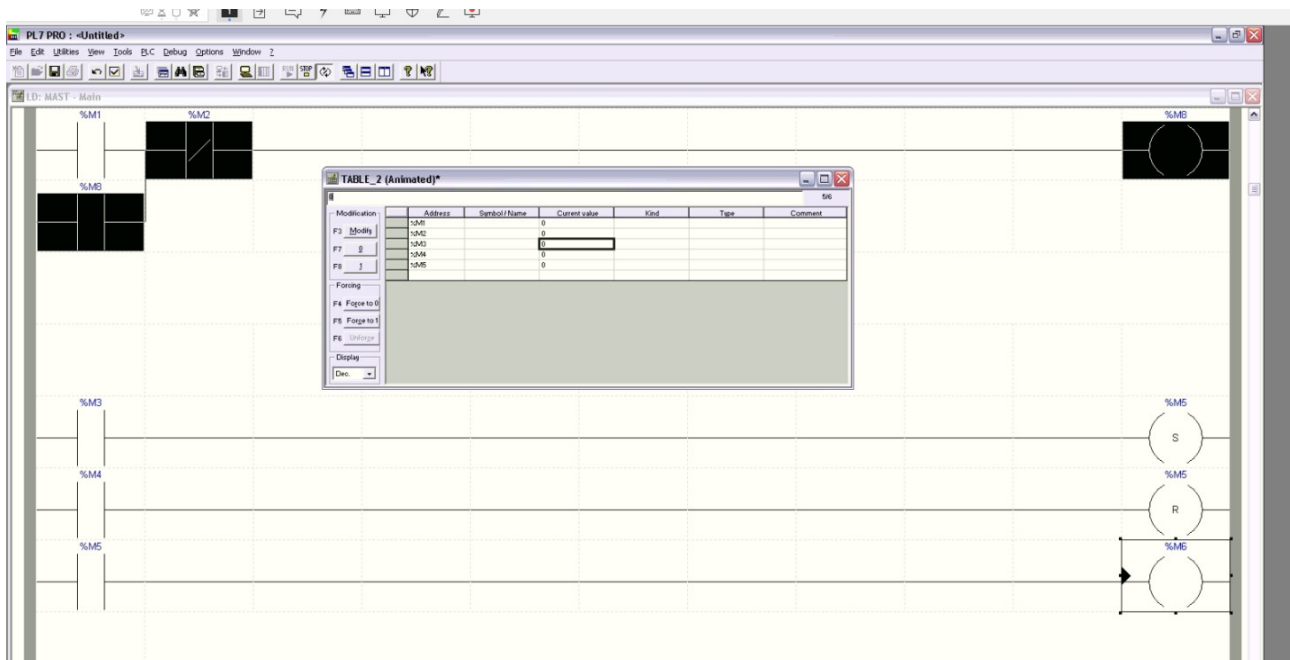
2.2 Program 2

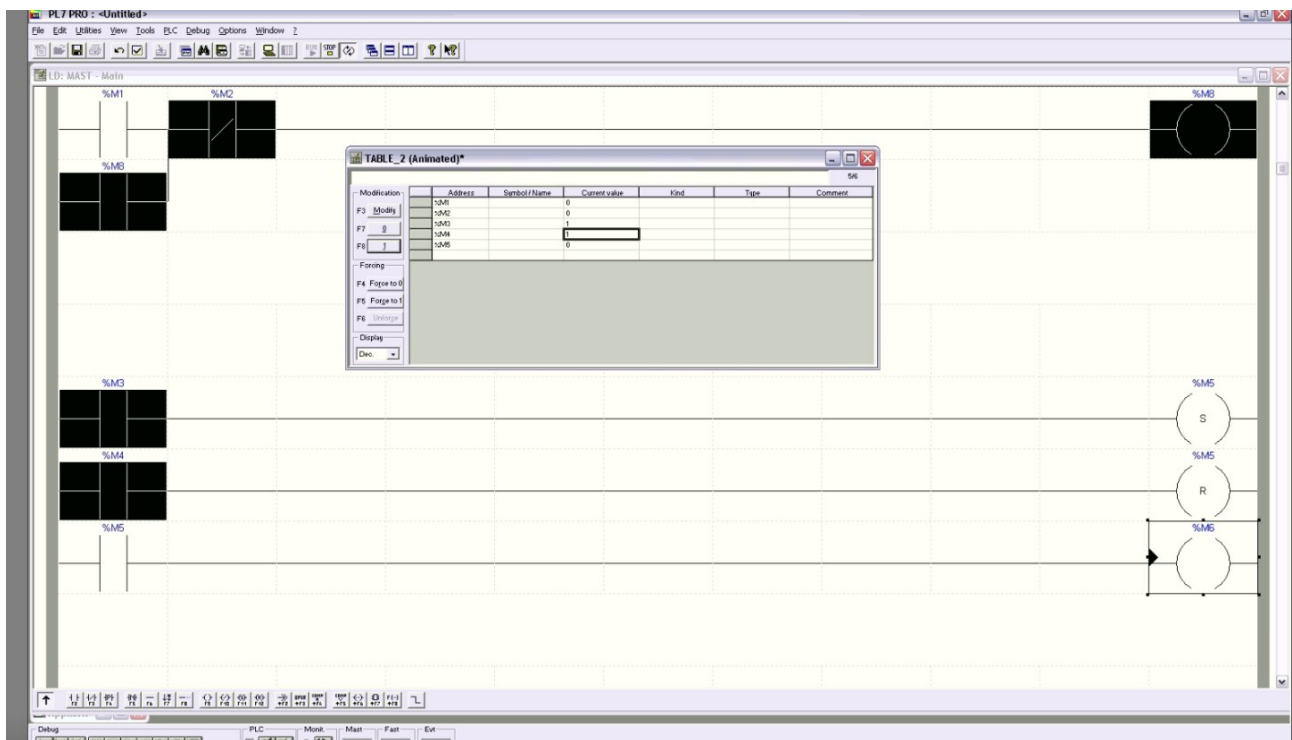
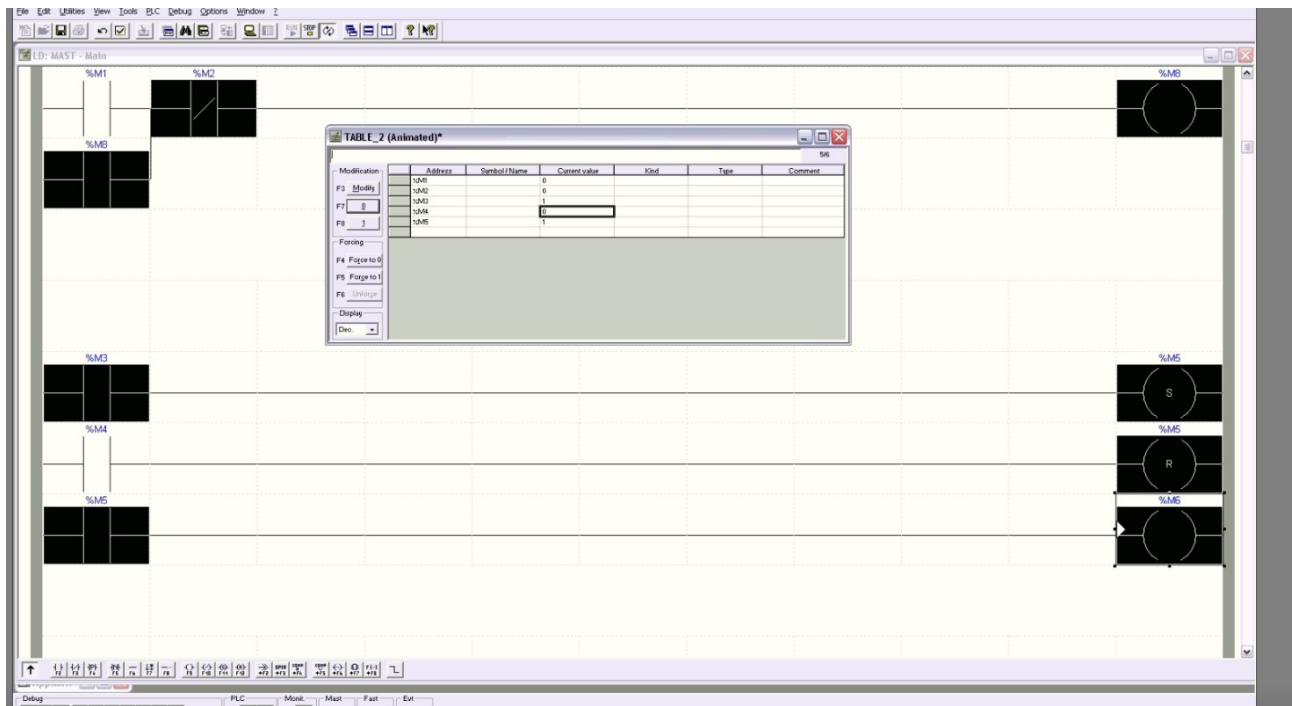
2.2.a)





2.2.b)

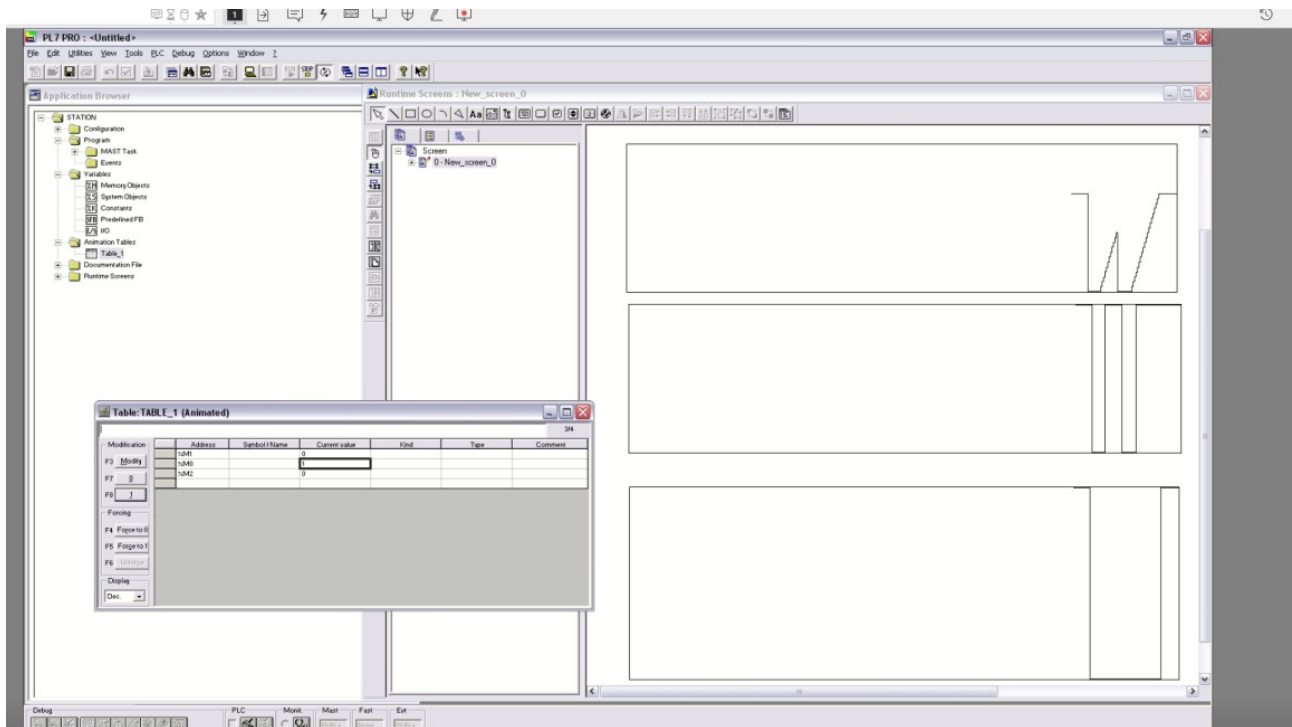




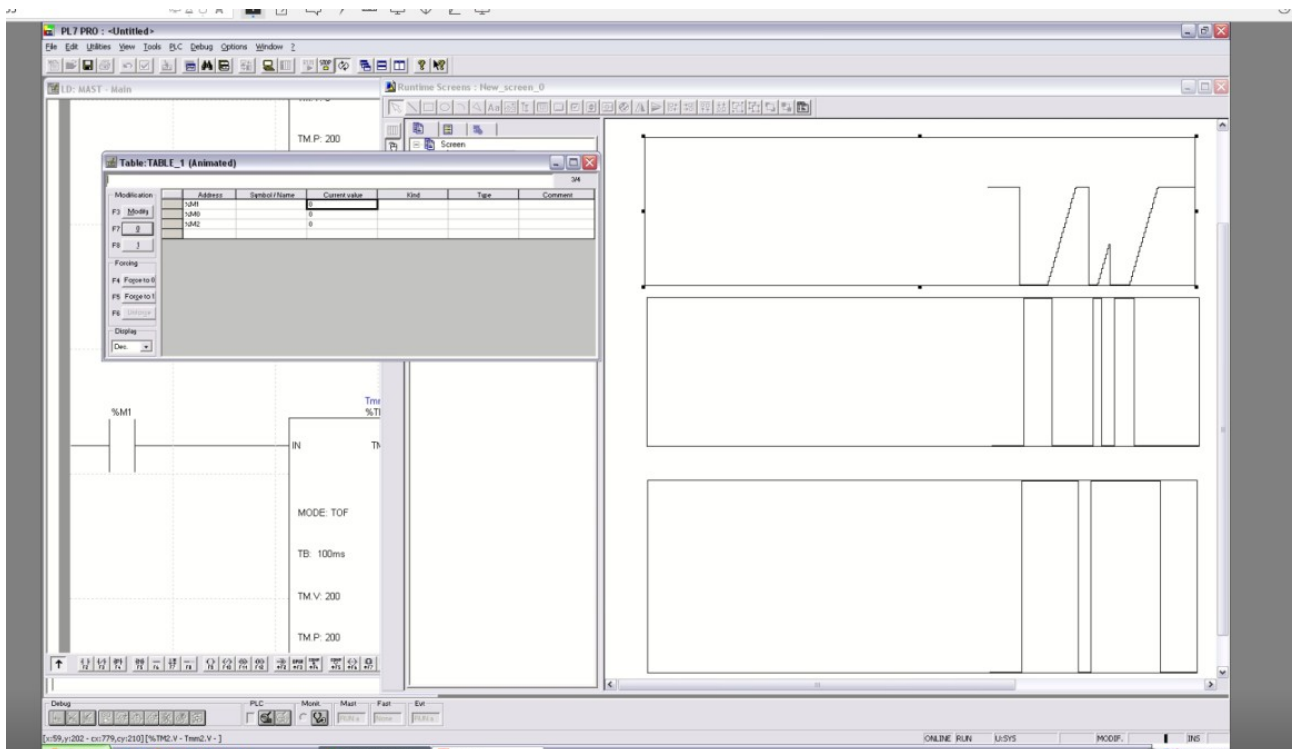
## 2.3 Program 3

Timer TON sprawia że powstaje zbocze narastające. Timer TOF powoduje, że powstaje zbocze opadające, natomiast timer TP powoduje, że na wyjściu jest impuls.

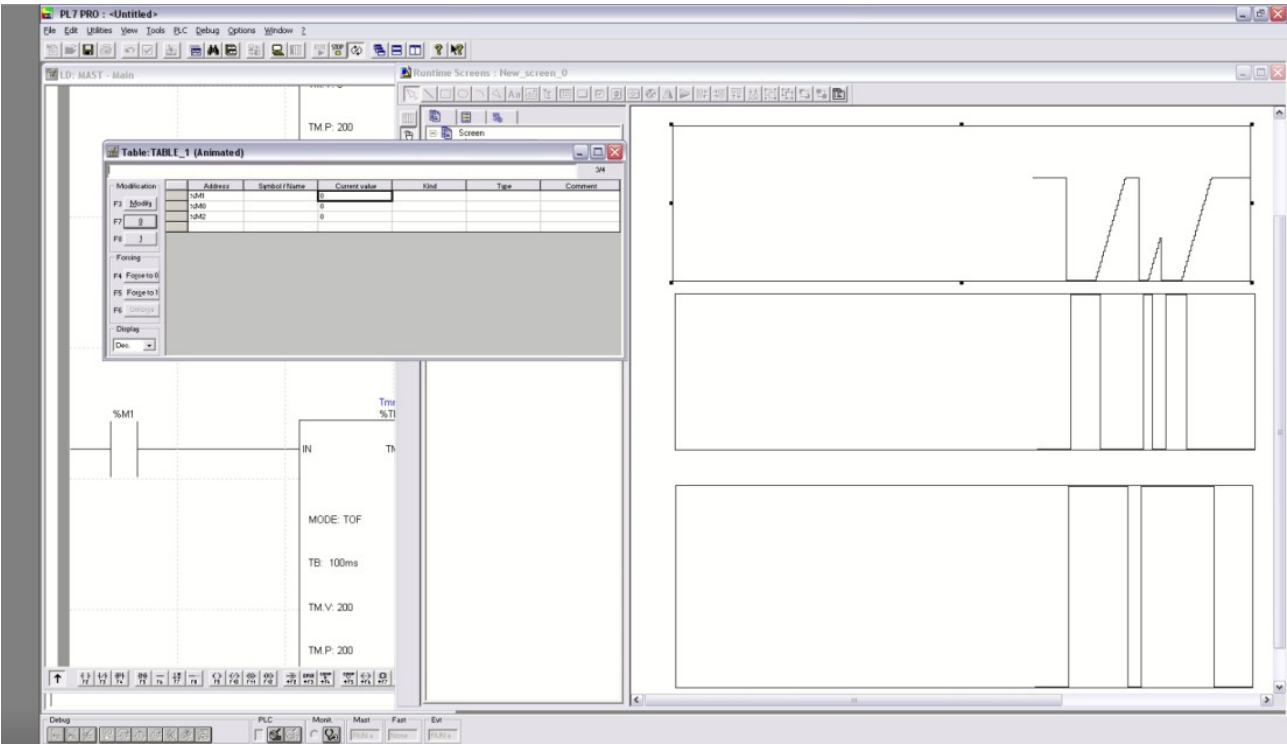
### Timer TON



### Timer TOF

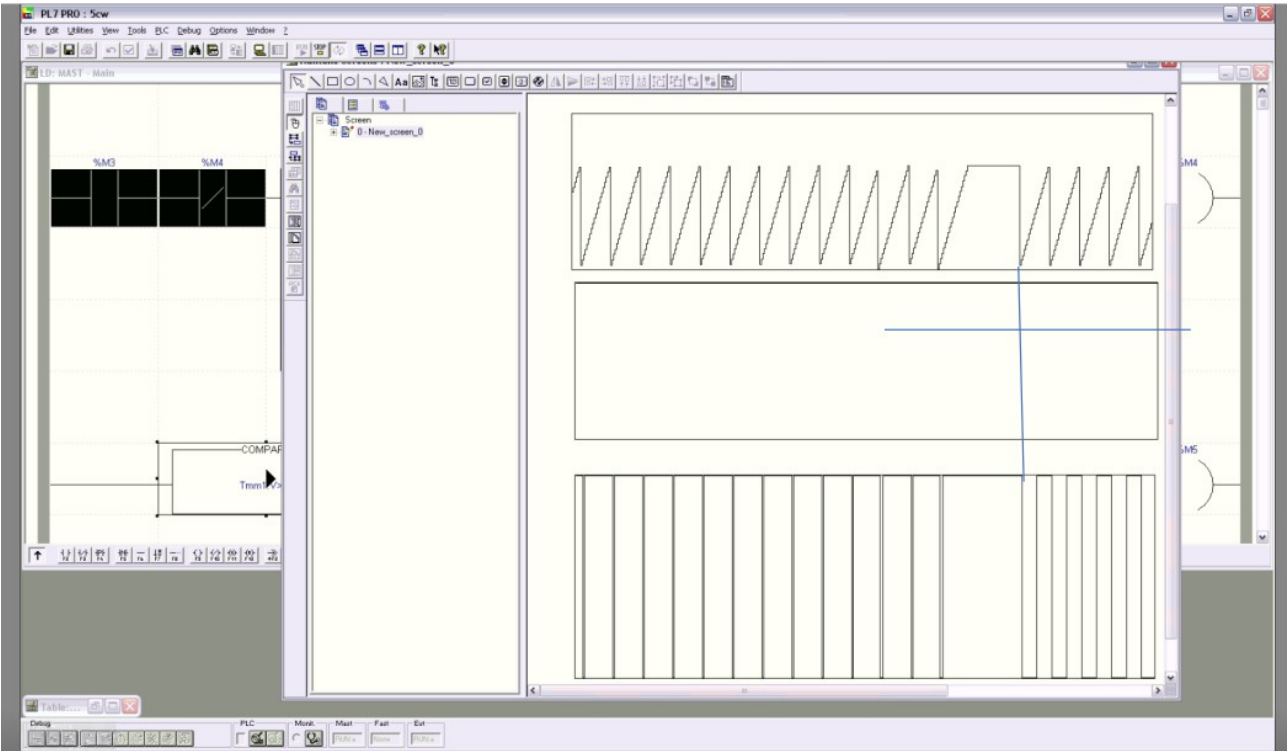


Timer TP

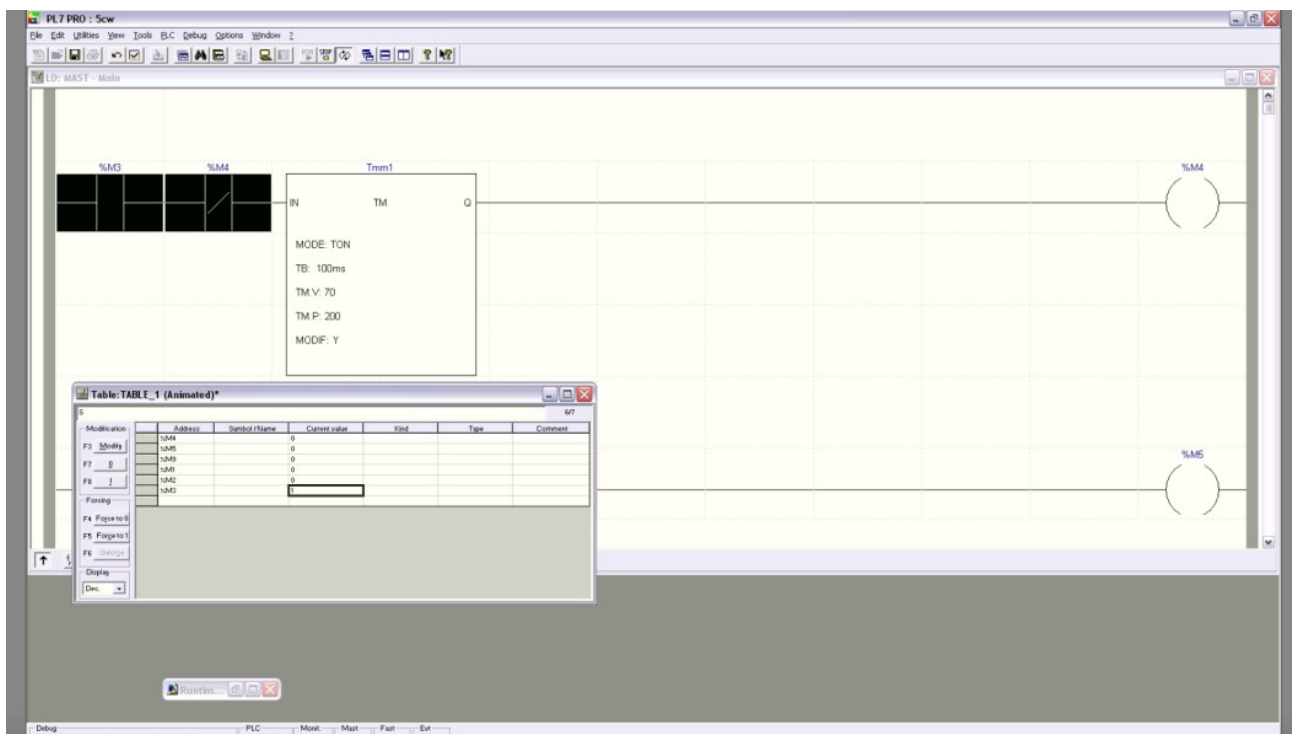
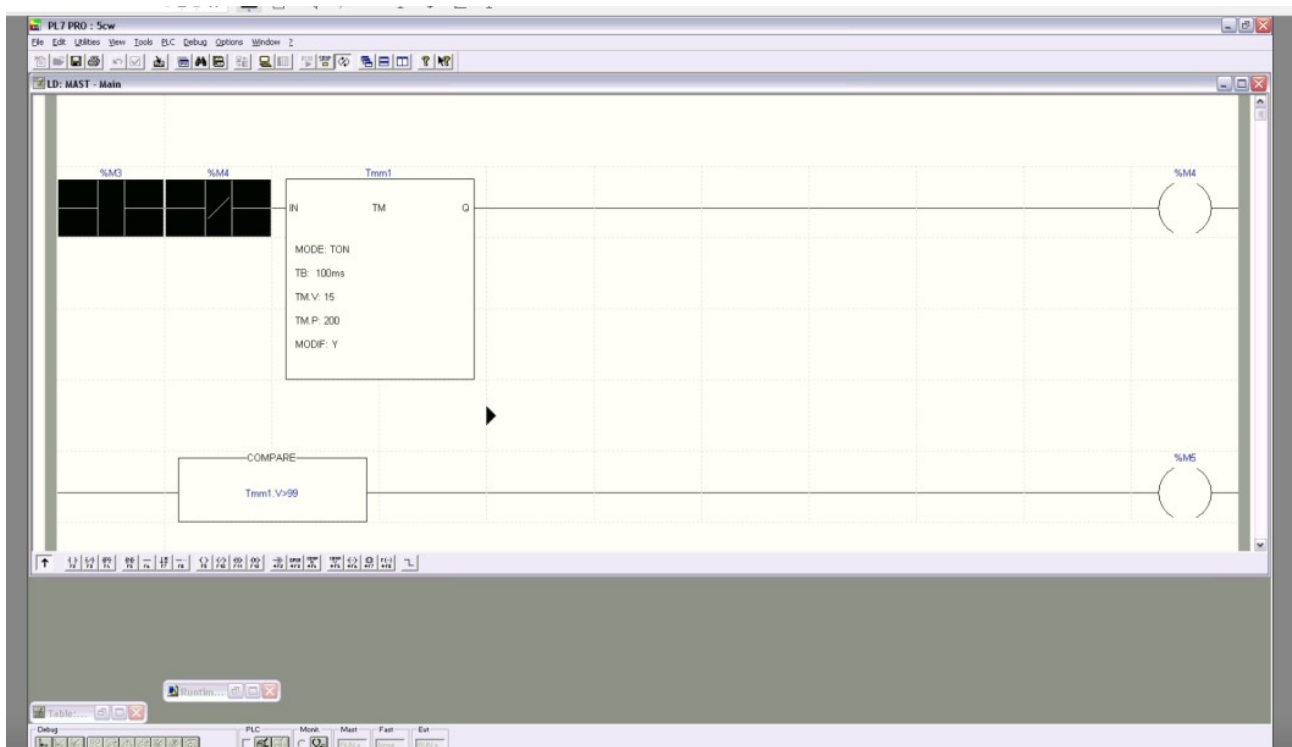


2.4 Program 4

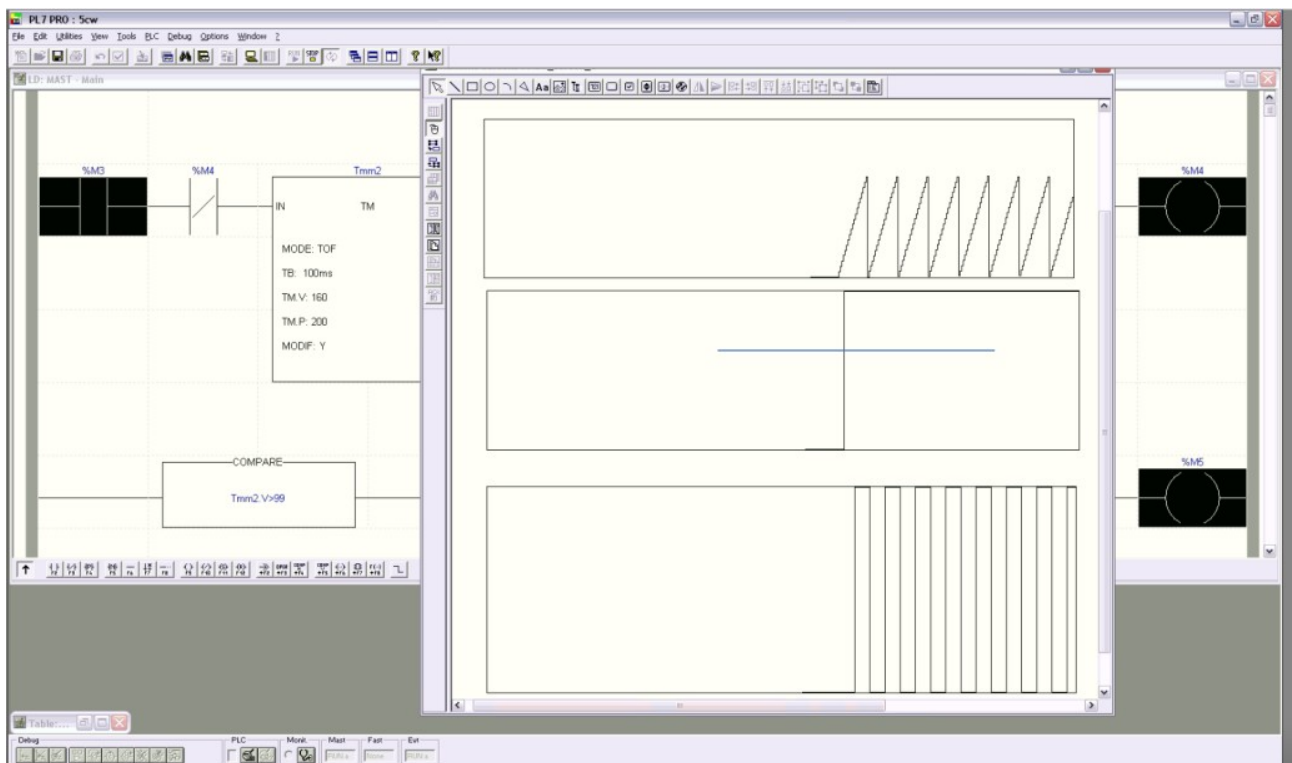
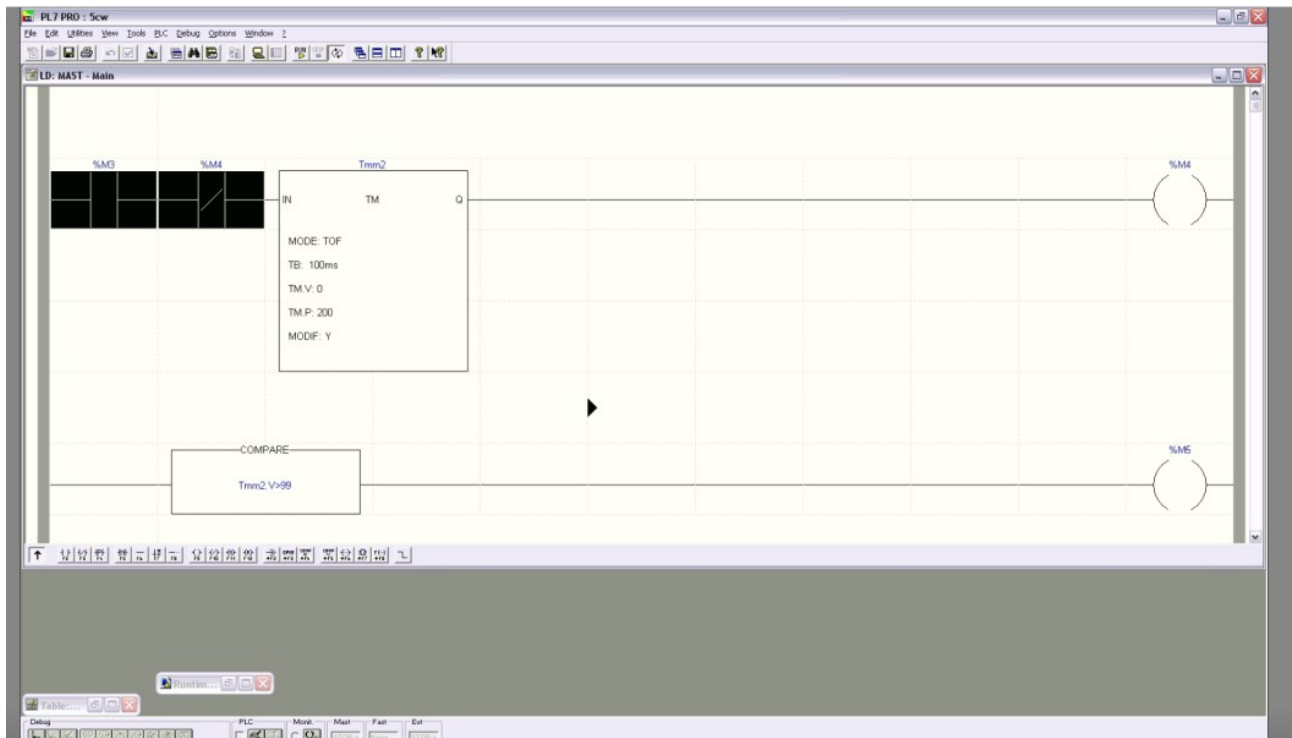
TON



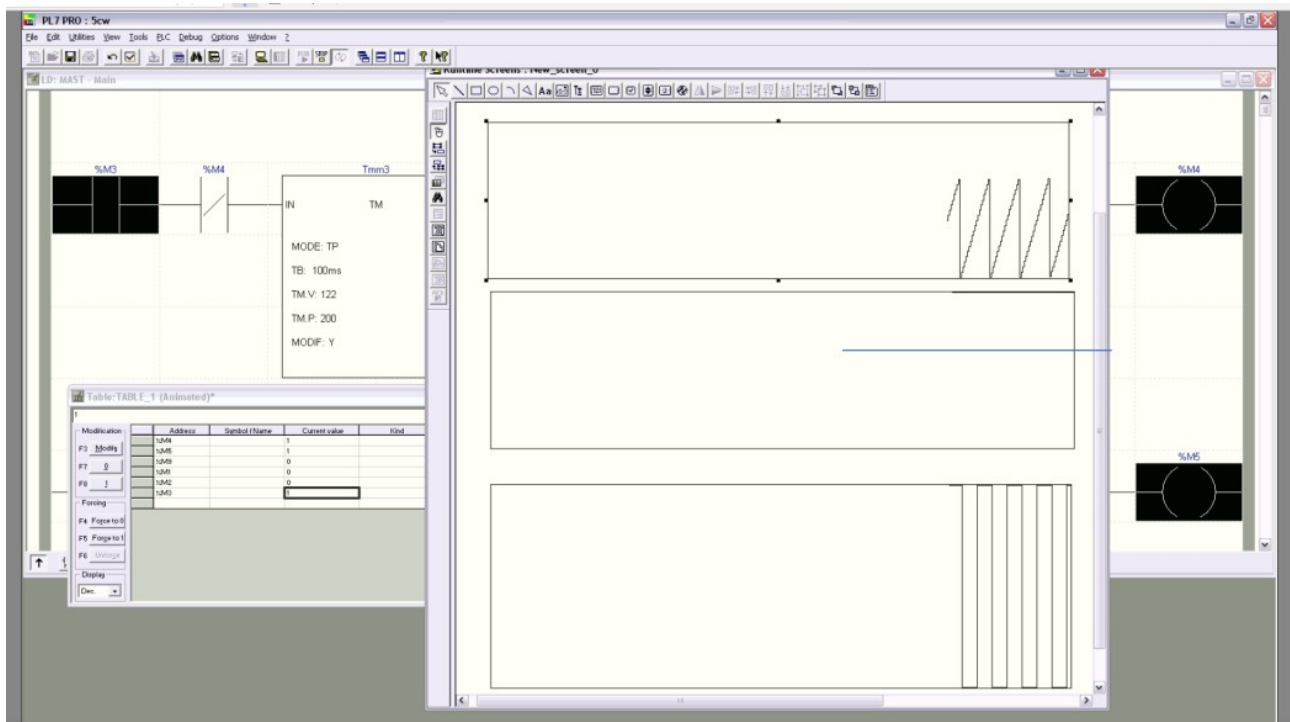
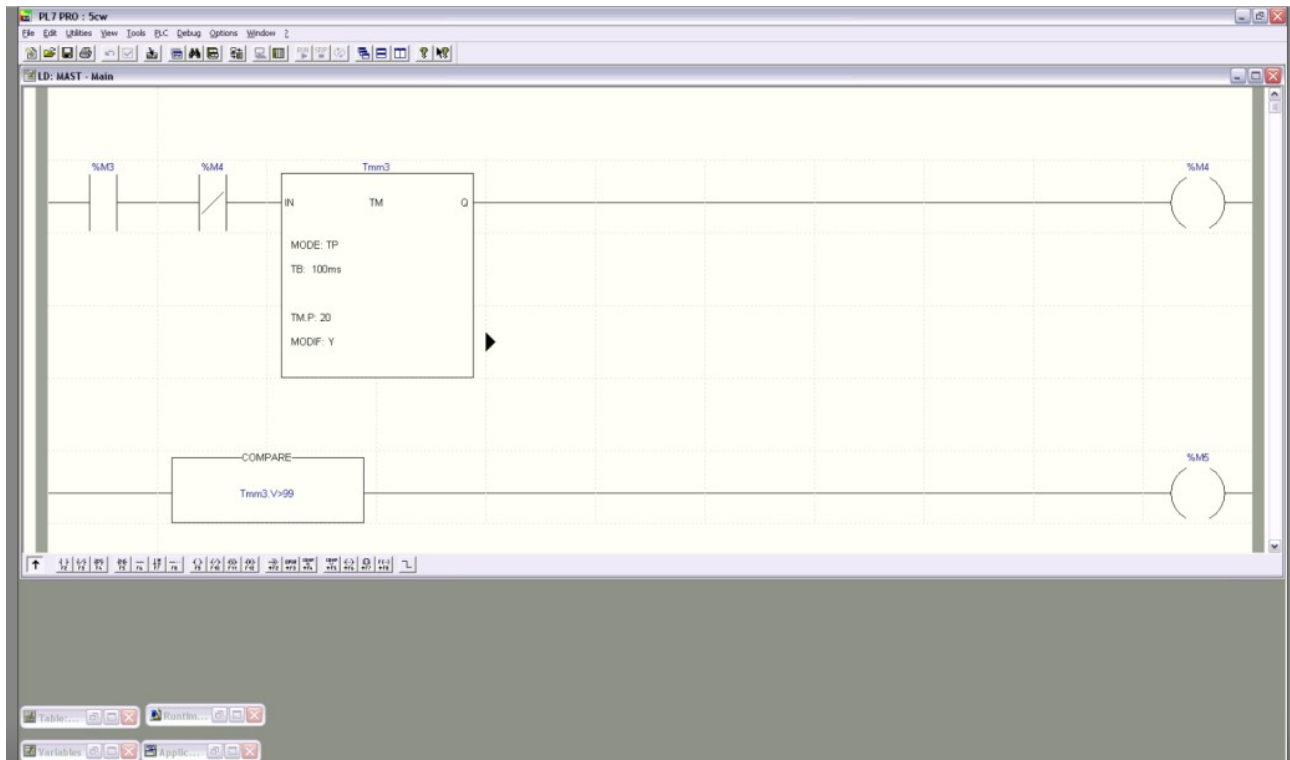




TOF



TP



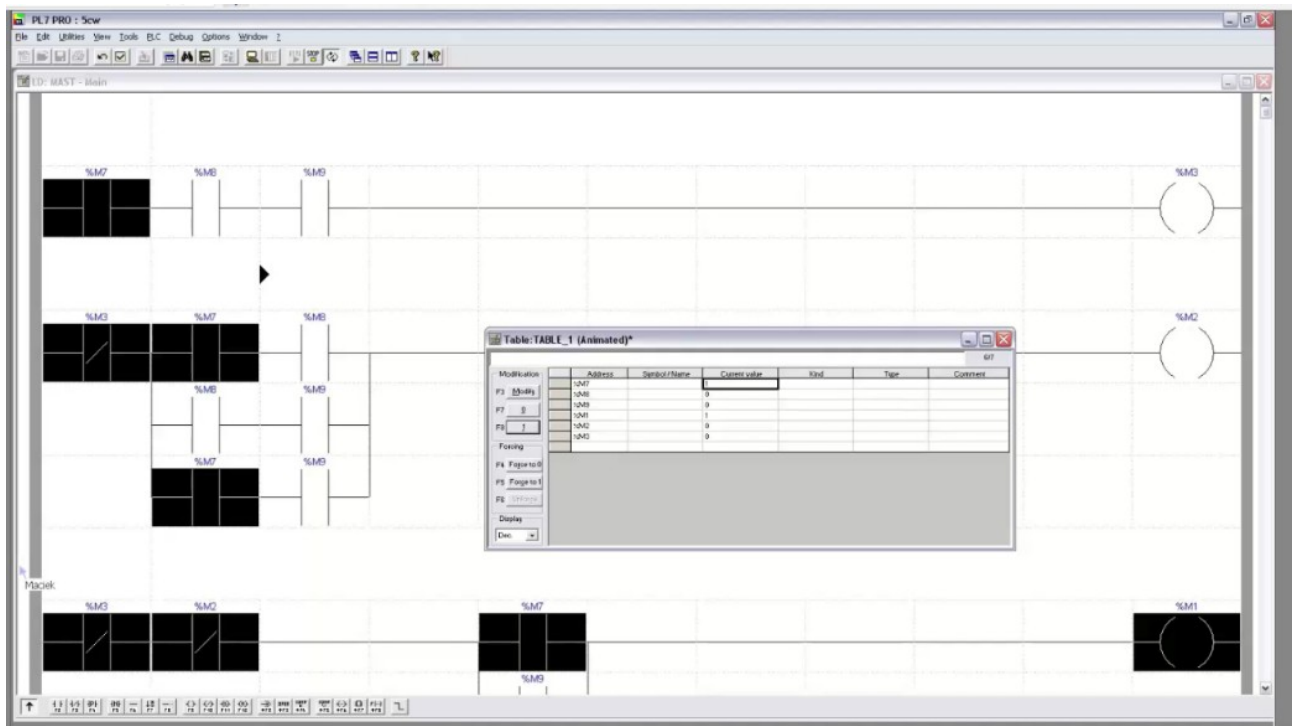
Układy z Timerem typu TON, TOF i TP generują sygnał prostokątny 0,5Hz o wypełnieniu 50%. Układ przełącza wyjście na stan wysoki przy osiągnięciu połowy wartości maksymalnej (50%) przez licznik timera po czym przy wartości 0 osiąga stan niski i cykl zaczyna się od początku.

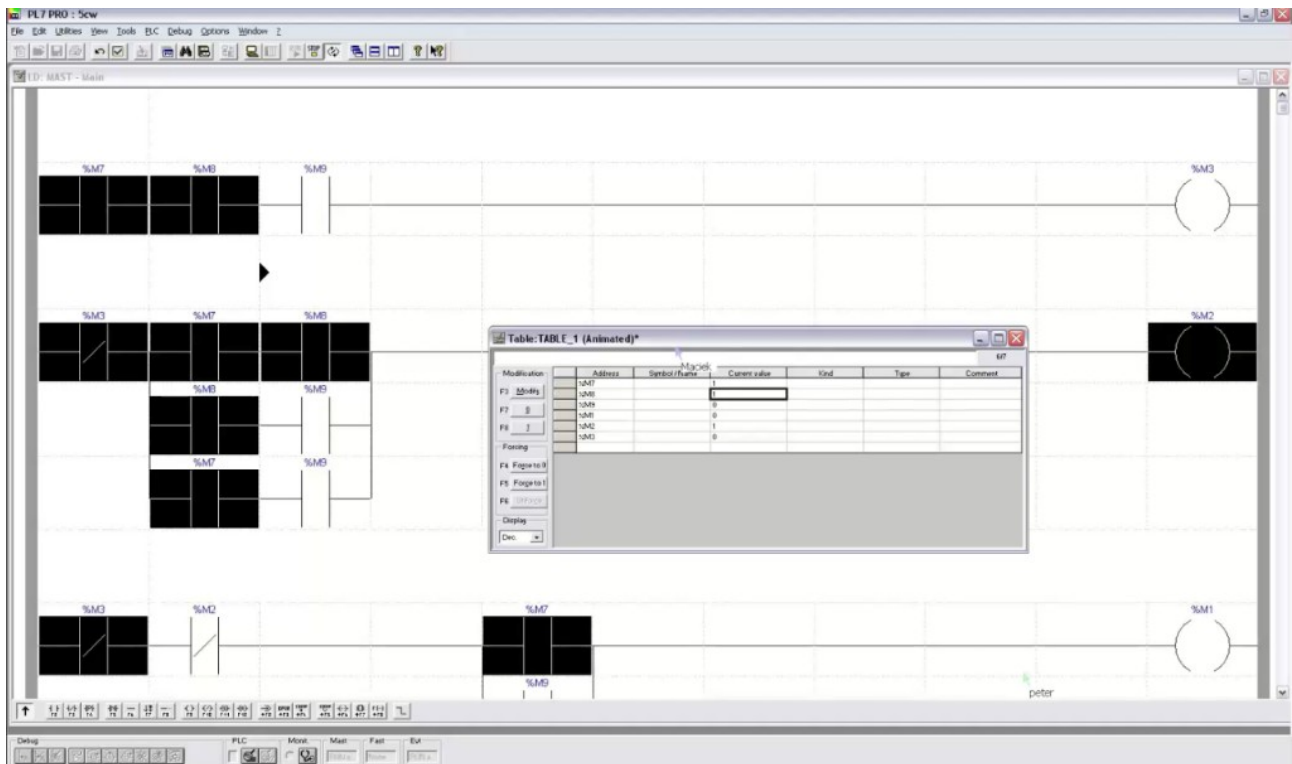
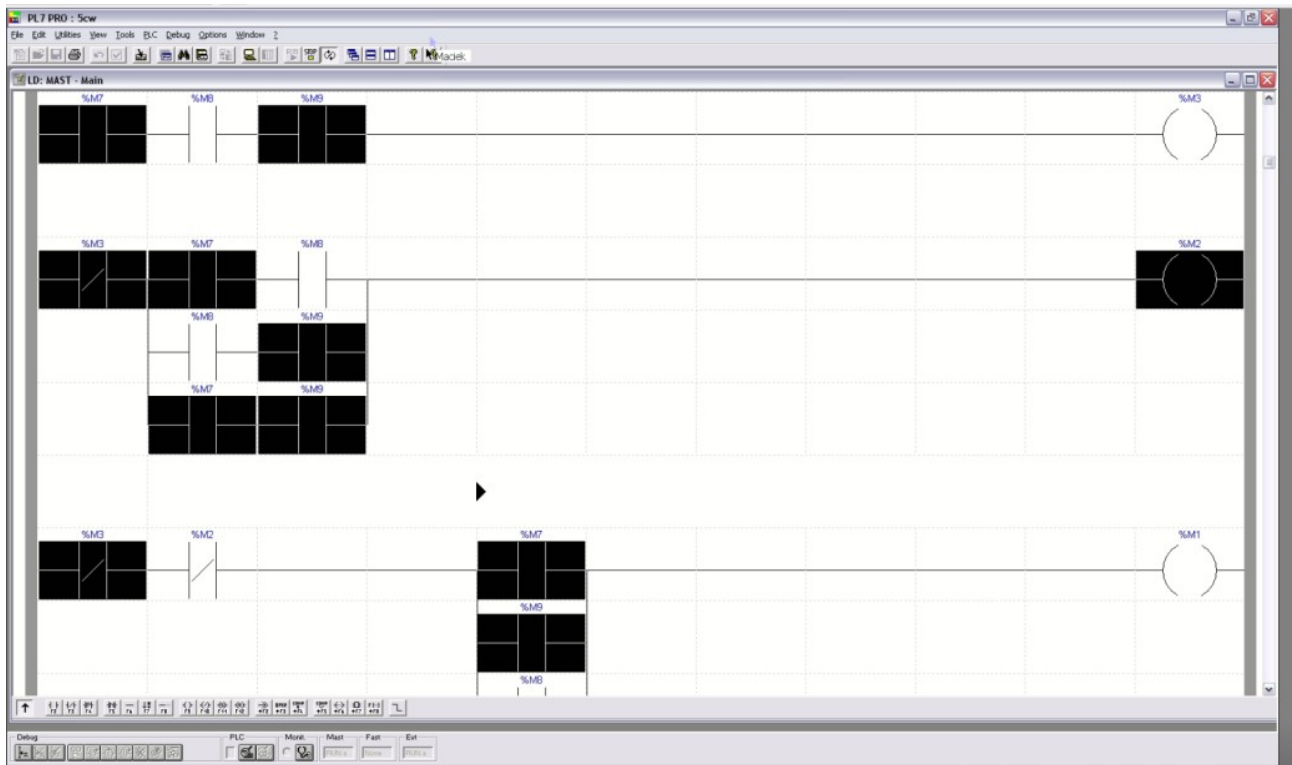
The screenshot displays the PL7 PRO software interface. The main workspace shows a ladder logic diagram with several rungs. The first rung contains three normally open contacts labeled %M7, %M8, and %M9, followed by a coil labeled %M0. The second rung contains a normally open contact labeled %M3, followed by two parallel branches: one with a normally open contact labeled %M7 and a coil labeled %M8, and another with a normally open contact labeled %M8 and a coil labeled %M9. The third rung contains two parallel branches: one with a normally open contact labeled %M7 and a coil labeled %M8, and another with a normally open contact labeled %M8 and a coil labeled %M9. The fourth rung contains three normally open contacts labeled %M3, %M8, and %M9, followed by a coil labeled %M1. The fifth rung contains a normally open contact labeled %M7 and a coil labeled %M9. On the right side, there are two circular symbols labeled %M0 and %M2, and a circular symbol labeled %M1 at the bottom right.

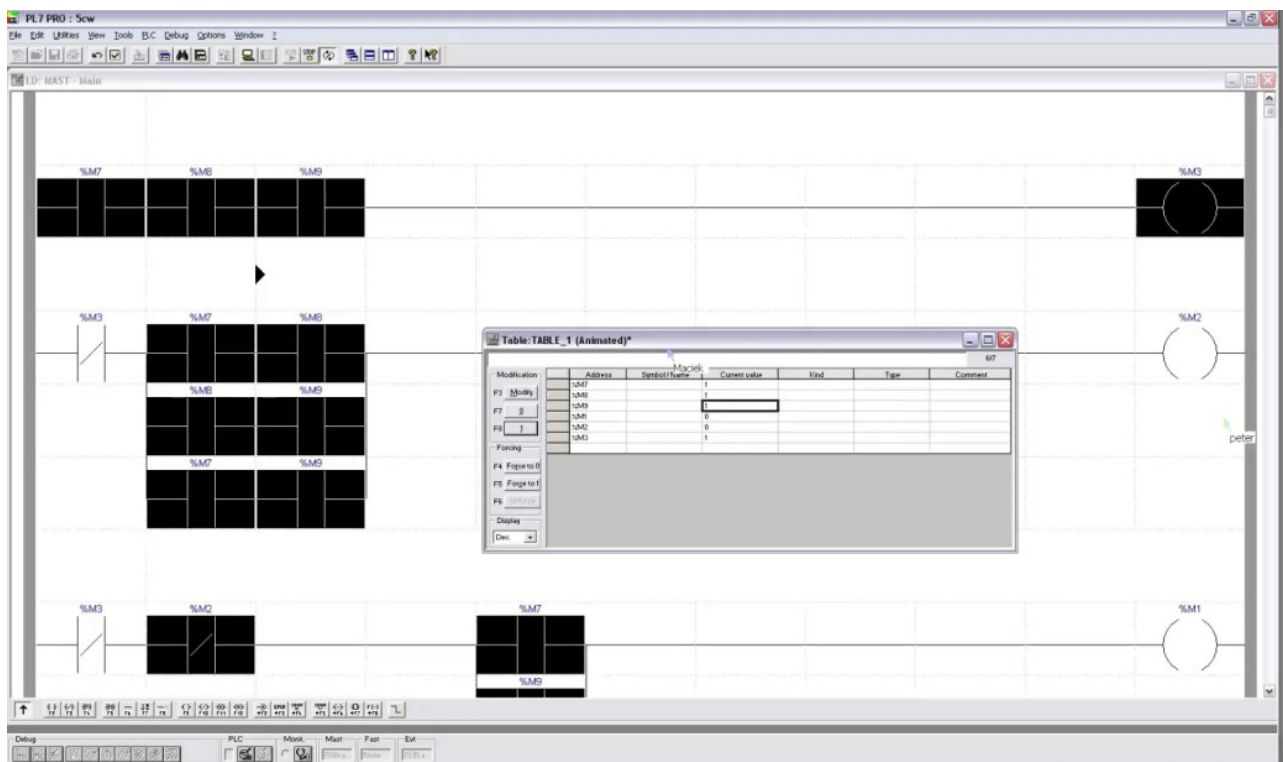
A dialog box titled "Table: TABLE\_1 (Animated)\*" is open in the foreground. It contains a table with the following data:

Address	Symbol/Name	Current value	Unit	Type	Comment
M0		0	Boolean		
M8		0			
M9		0			
M8		0			
M2		0			
M0		0			

The dialog box also includes a "Modification" section with a list of variables (M0, M8, M9, M8, M2, M0) and a "Display" section with a "On" button.







Zmiany stanów na wejściach M7, M8, M9 zmieniają w odpowiedniej konfiguracji stany na wyjściach – M3, M2, M1 co widać na załączonych grafikach. Załączenie M7 (jednego bitu) załącza M1 (pierwszy bit). Załączenie M7+M8 (dwóch bitów) załącza M2 (drugi bit). Załączenie M7+M8+M9 (trzech bitów) załącza M3 (trzeci bit).

### 3 Wnioski

Zmontowano programy w języku programowania LADDER w sposób przedstawiony wyżej. Programy działają zgodnie z założeniami postawionymi nam w poszczególnych ćwiczeniach, co pozwala nam postawić tezę, że ćwiczenia zostały wykonane w sposób poprawny.