Lab 4: Timers on the SIMATIC S7-1200

Lab Objective

The purpose of this lab is to demonstrate how to declare and use different types of timers on the SIMATIC S7-1200 PLC. You will learn:

- How to use extended instructions to implement timing functions
- How to use the **Bit Logic Operations** library in TIA Portal
- How to simulate a program using the **PLCSIM** simulator

1. IEC Timers

There are three types of IEC-compliant timers:

- TON **(On-delay timer)**
- TOF **(Off-delay timer)**
- TP **(Pulse timer)**

2.1. On-Delay Timer (TON)

The **TON** instruction (On-delay timer) allows you to delay the activation of the output **Q** by the programmed time duration **PT**.

The instruction is triggered by a **rising edge** at the input **IN**. Once the instruction starts, the timer begins counting the duration **PT**. After this time has elapsed, the output **Q** is set to **1**.

The output **Q** remains at **1** as long as the input **IN** stays at **1**. If the input **IN** changes from **1** to **0**, the output **Q** resets to **0**.

The timing process will restart when a new rising edge is detected at the **IN** input.

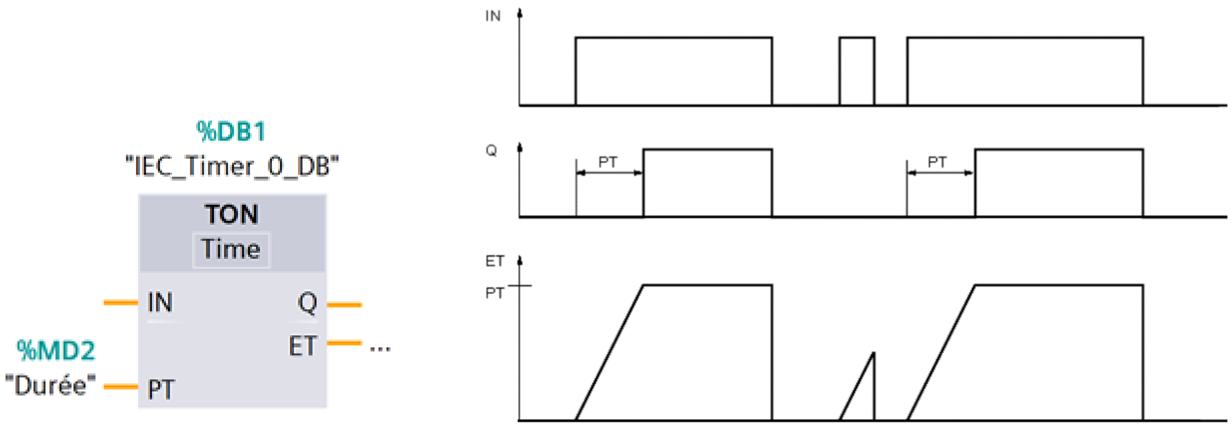


Figure.1. Operating principle of the TON timer.

2.2. Off-Delay Timer (TOF):

The **TOF** instruction (Off-delay timer) allows you to delay the reset of the output **Q** by the programmed time duration **PT**.

The output **Q** is set to **1** when the input **IN** transitions from **0** to **1** (rising edge). When the logical state at **IN** returns to **0**, the programmed time **PT** begins counting.

The output **Q** remains at **1** while the **PT** duration elapses. Once the time has elapsed, **Q** is reset to **0**. If the input **IN** goes back to **1** before the **PT** time has elapsed, the timer is reset, and the output **Q** remains at **1**.

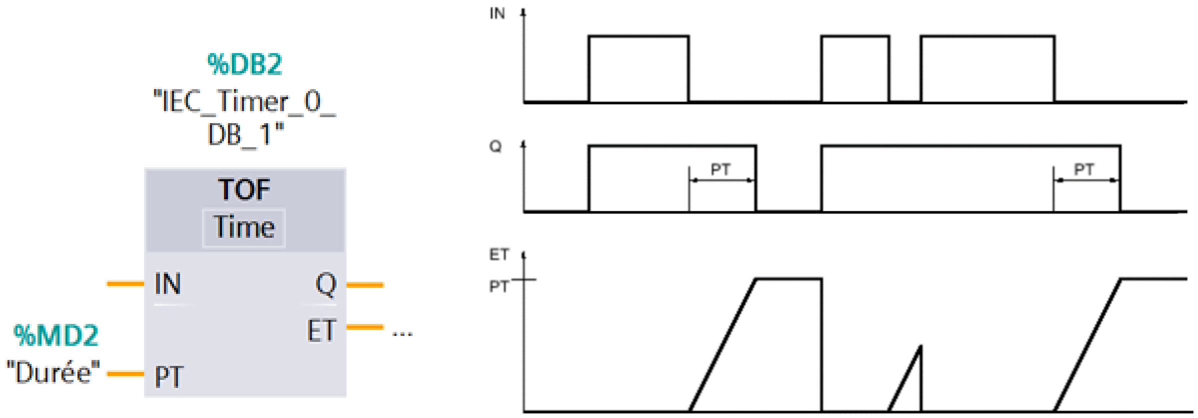


Figure.2. Operating principle of the TOF timer

2.3. Pulse Generation (TP)

The **TP** instruction (Pulse timer) sets the output **Q** to **1** for a programmed time duration. The instruction is triggered when the input **IN** changes from **0** to **1** (rising edge). The programmed duration **PT** begins counting at the moment the instruction is triggered.

The output **Q** remains at **1** for the entire **PT** duration, regardless of any changes at the input. Even if a new rising edge is detected during this time, it has no effect on the output state **Q** until the full **PT** period has elapsed.

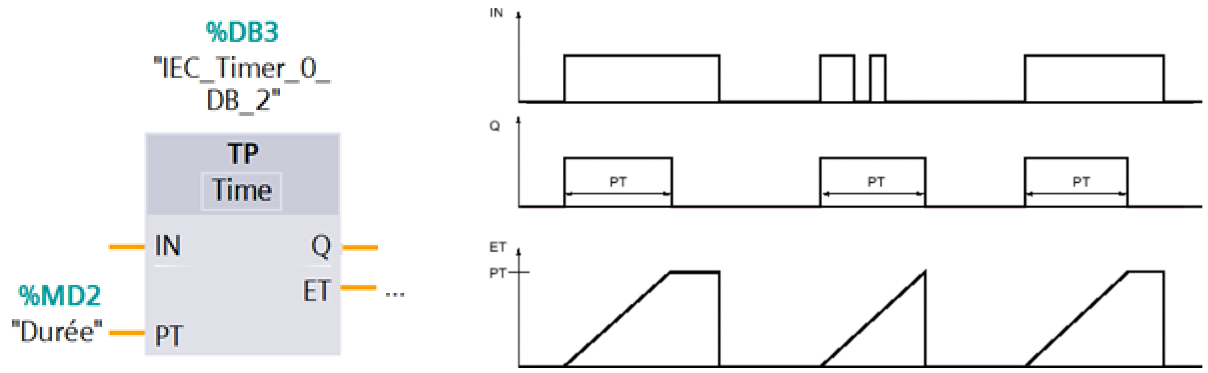


Figure.3. Operating principle of the TP timer.

The figure below illustrates the difference between the TON, TOF, and TP timers. As an example, a time duration **PT** of 2 seconds has been used.

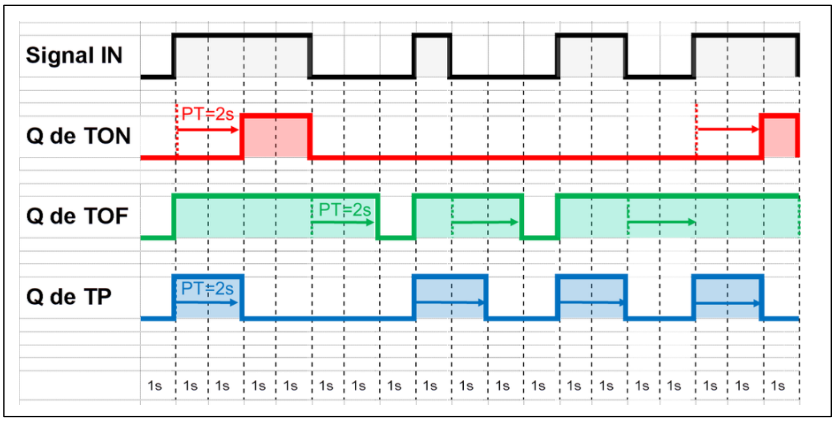
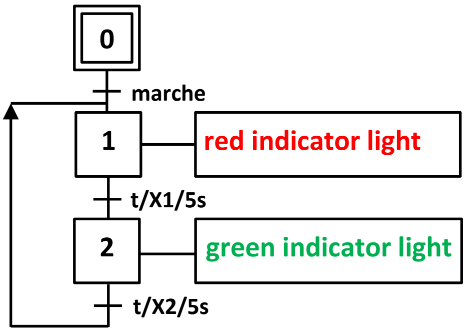


Figure.2. Operating principle of the three timer types

Example:

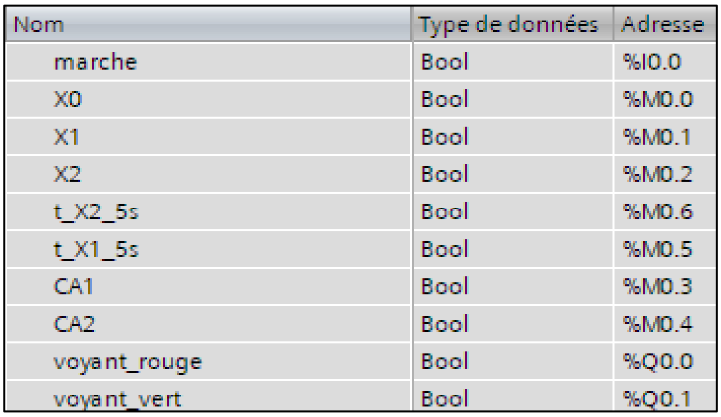
After pressing the **marche** button, a **red indicator light** turns on for **5 seconds**, then turns off, followed by a **green indicator light** that turns on for **5 seconds** and then turns off. This cycle repeats **indefinitely**.

Translate the GRAFCET below into Ladder Logic.



In this GRAFCET, a **5-second timer** is used. There are several possible ways to implement this timing function. Before proceeding with the programming, we will first declare the **variable table**, and then provide the **step activation/deactivation table**.

Variable Table



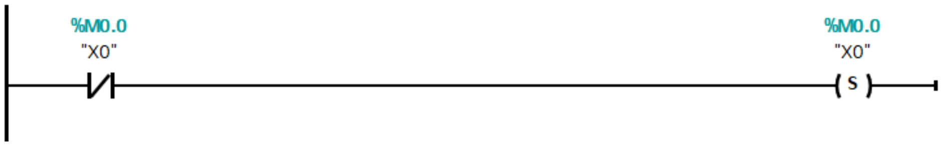
Nom	Type de données	Adresse
marche	Bool	%I0.0
X0	Bool	%M0.0
X1	Bool	%M0.1
X2	Bool	%M0.2
t_X2_5s	Bool	%M0.6
t_X1_5s	Bool	%M0.5
CA1	Bool	%M0.3
CA2	Bool	%M0.4
voyant_rouge	Bool	%Q0.0
voyant_vert	Bool	%Q0.1

Step Activation and Deactivation Table

Etape $(\mathrm{X_i})$	Condition d'activation (CA_i)	Condition de désactivation (CD_i)
X_0	*	X_1
X_1	$X_0 \cdot marche + X_2 \cdot t/X_2/5s$	X_2
X_2	$X_1 \cdot t/X_1/5s$	X_1

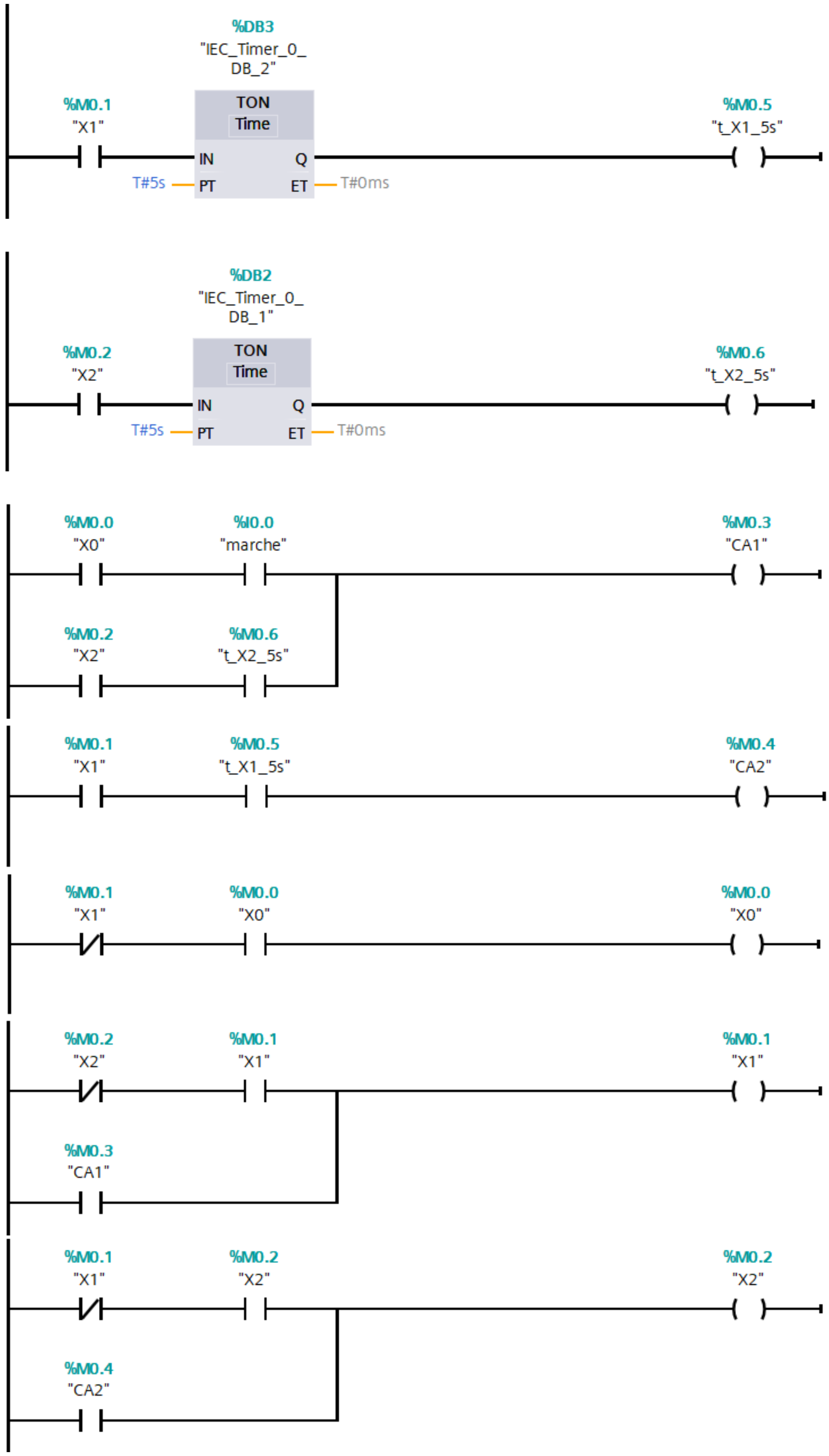
* The activation of the initial step is performed in Organization Block OB100 (Startup Block).

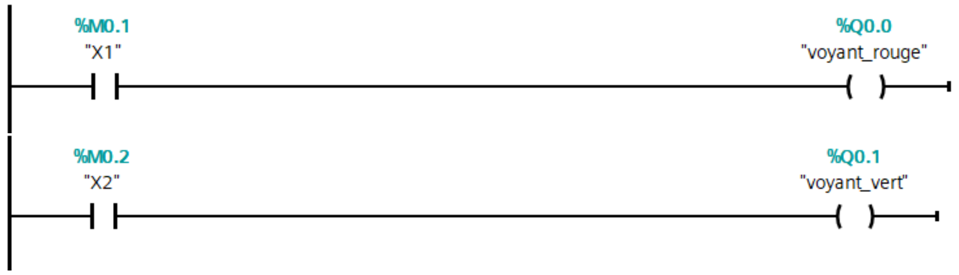
GRAFCET programming using the TON timer:
The startup organization block OB100 is used to initialize the process.



The Cyclic Organization Block OB1

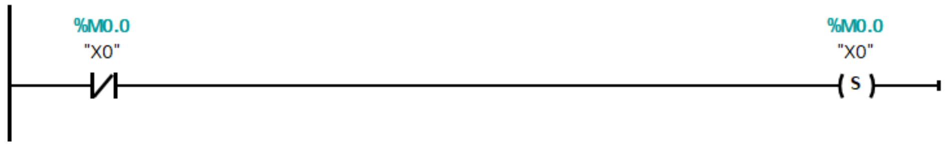
It is recommended to start by declaring the two transitions (**t/X_1/5s** and **t/X_2/5s**) using the memory variables **t_X1_5s (M0.5)** and **t_X2_5s (M0.6)**, respectively.





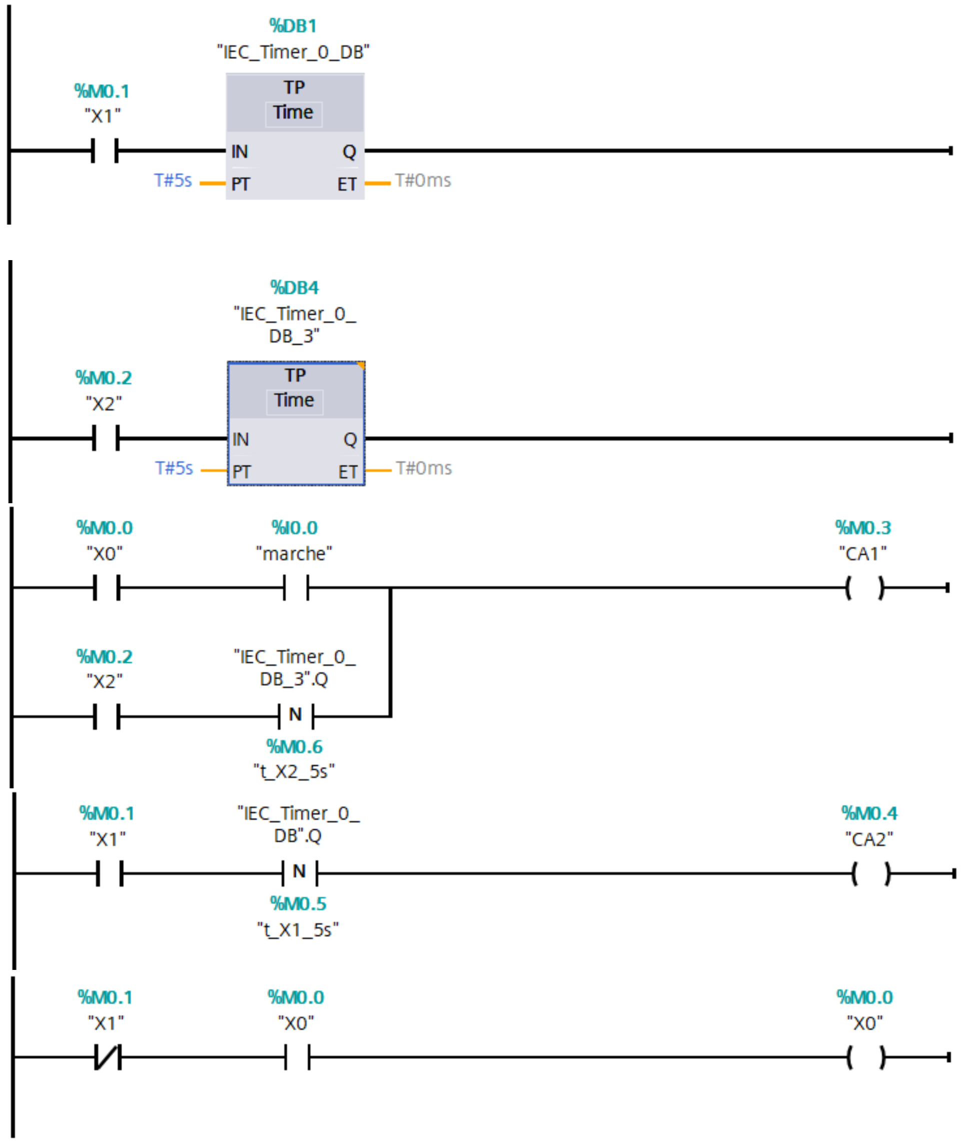
GRAFCET Programming Using the TP Timer

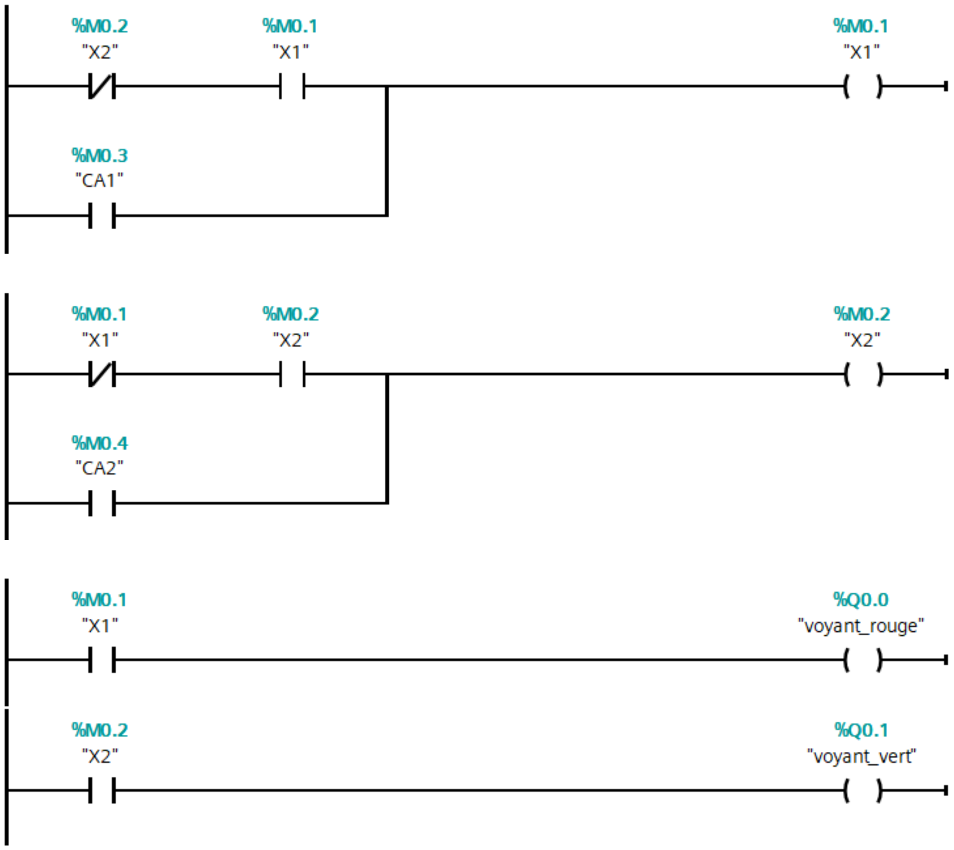
The startup organization block **OB100** is used to activate the initial step.



The Cyclic Organization Block OB1

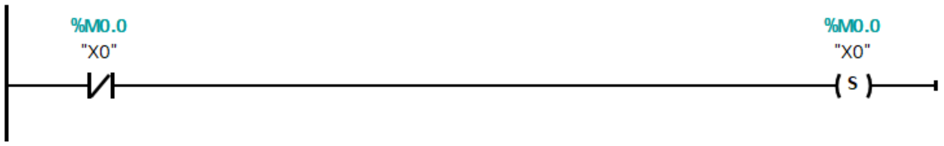
In our example, when using the **TP timer**, it is necessary to detect its end — that is, the **falling edge** of its output. To do this, the program uses the logical operation **"Detect falling edge of an operand"** from the **Bit Logic Operations** library, represented by the symbol **--|N|--**.



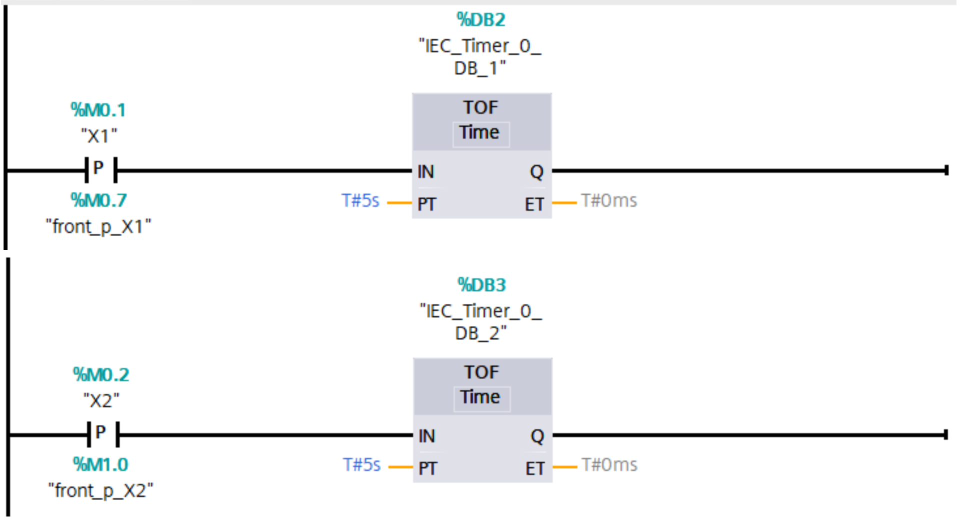


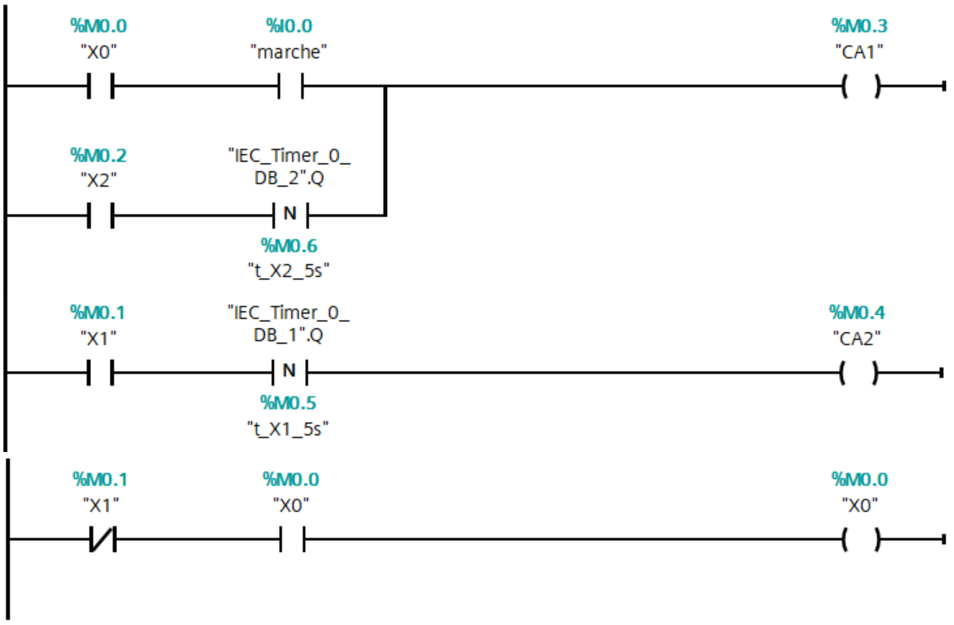
GRAFCET Programming Using the TOF Timer

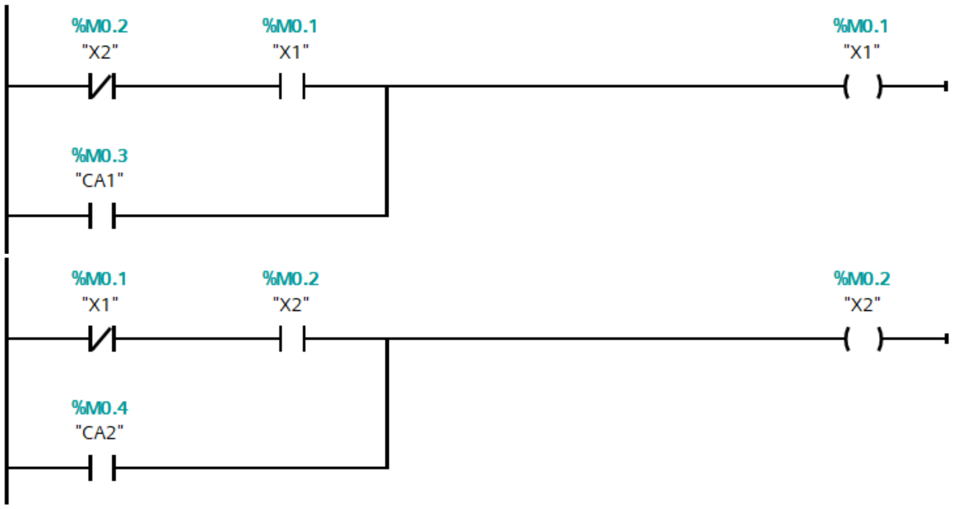
The startup organization block **OB100** is used to activate the initial step.

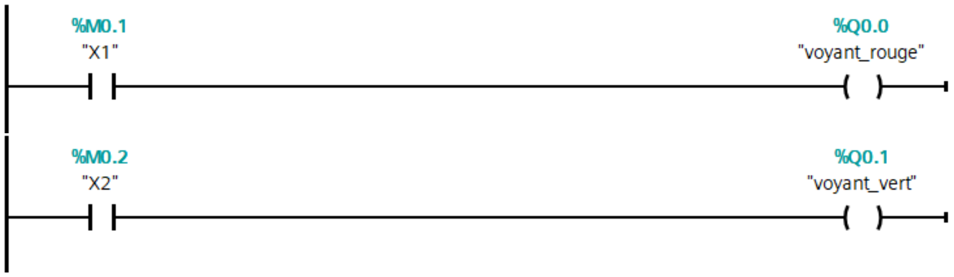


The Cyclic Organization Block OB1




%M0.0
"X0"
%I0.0
"marche"
%M0.3
"CA1"
%M0.2
"X2"
"IEC_Timer_0_DB_2".Q
N
%M0.6
"t_X2_5s"
%M0.1
"X1"
"IEC_Timer_0_DB_1".Q
N
%M0.5
"t_X1_5s"
%M0.4
"CA2"
%M0.1
"X1"
%M0.0
"X0"
%M0.0
"X0"


%M0.2
"X2"
%M0.1
"X1"
%M0.1
"X1"
%M0.3
"CA1"
%M0.1
"X1"
%M0.2
"X2"
%M0.2
"X2"
%M0.4
"CA2"


%M0.1
"X1"
%Q0.0
"voyant_rouge"
%M0.2
"X2"
%Q0.1
"voyant_vert"