Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

PLC Programming languages

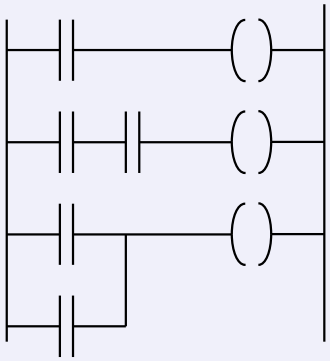
Common Programming Errors

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2223

Prof. José Gaspar, rev. 2022/2023

PLC Programming Languages (IEC 61131-3)

Ladder Diagram



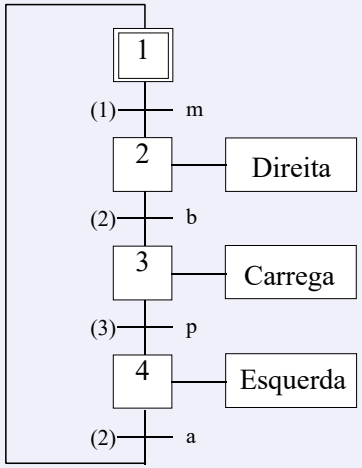
Structured Text

```
If  %I1.0  THEN
  %Q2.1 := TRUE
ELSE
  %Q2.2 := FALSE
END_IF
```

Instruction List

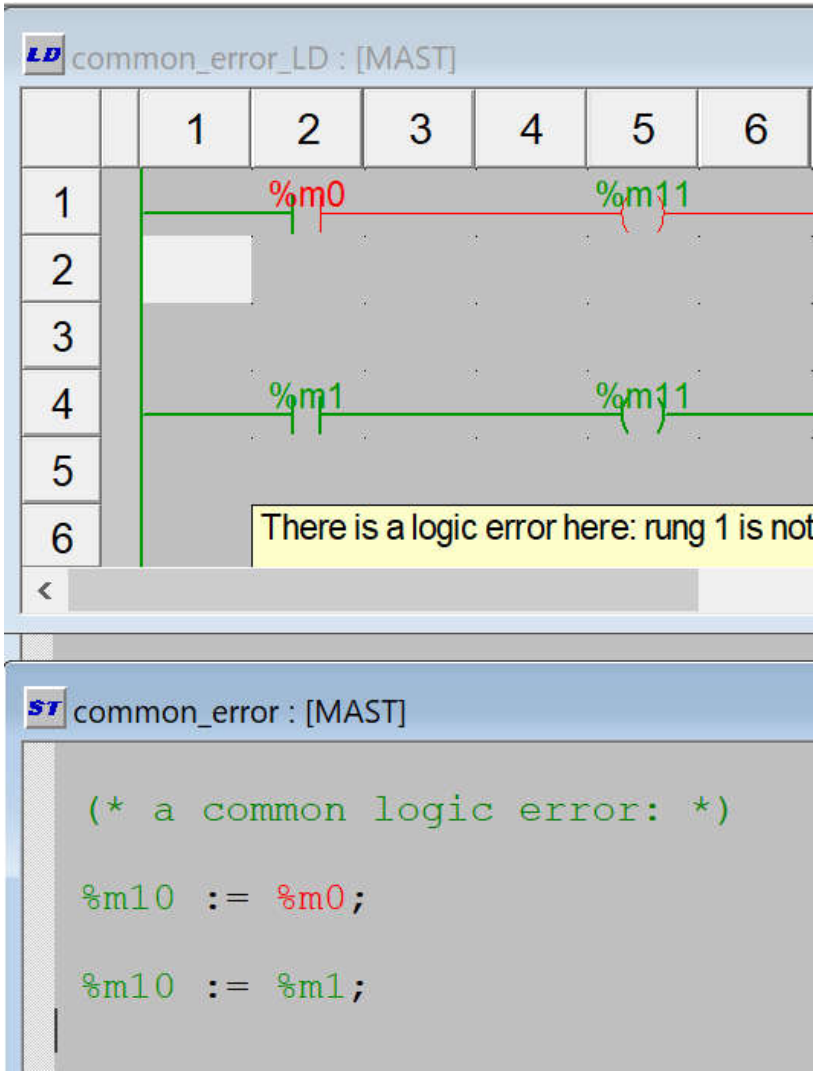
```
LD      %M12
AND     %I1.0
ANDN    %I1.1
OR      %M10
ST      %Q2.0
```

Sequential Function Chart (GRAFCET)

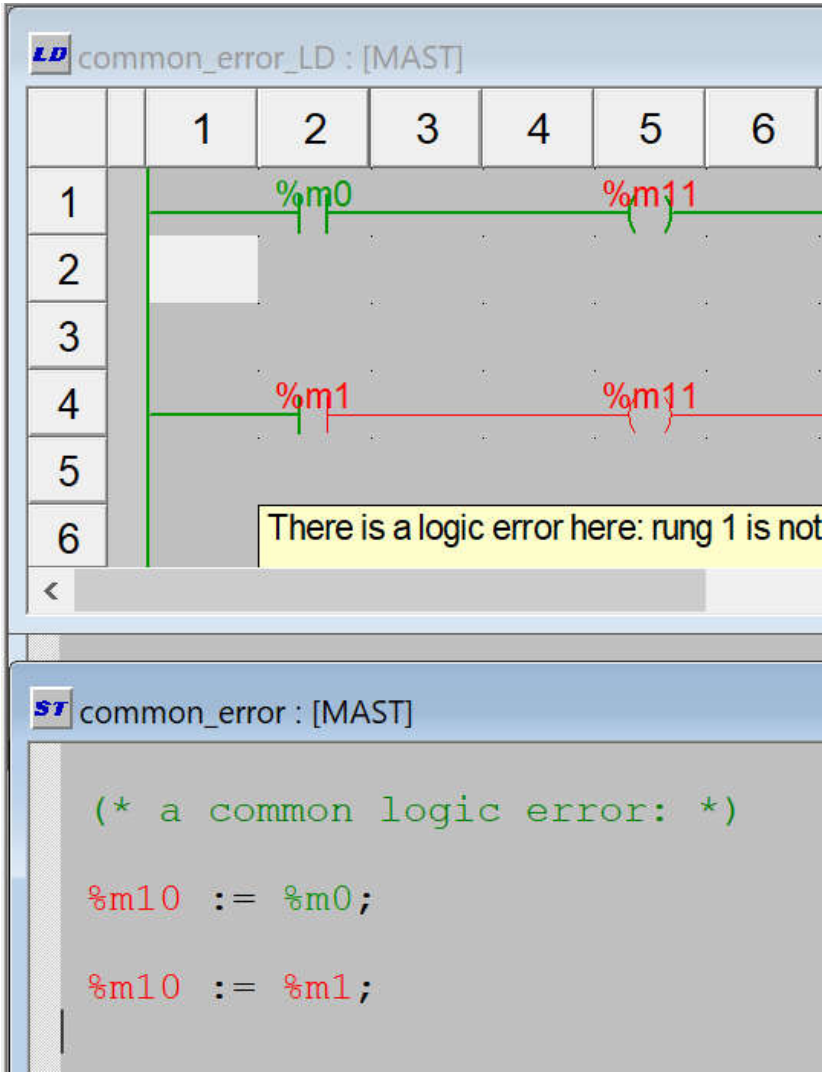


1. Multiple writes to one output in the same scan cycle

A very common programming error:

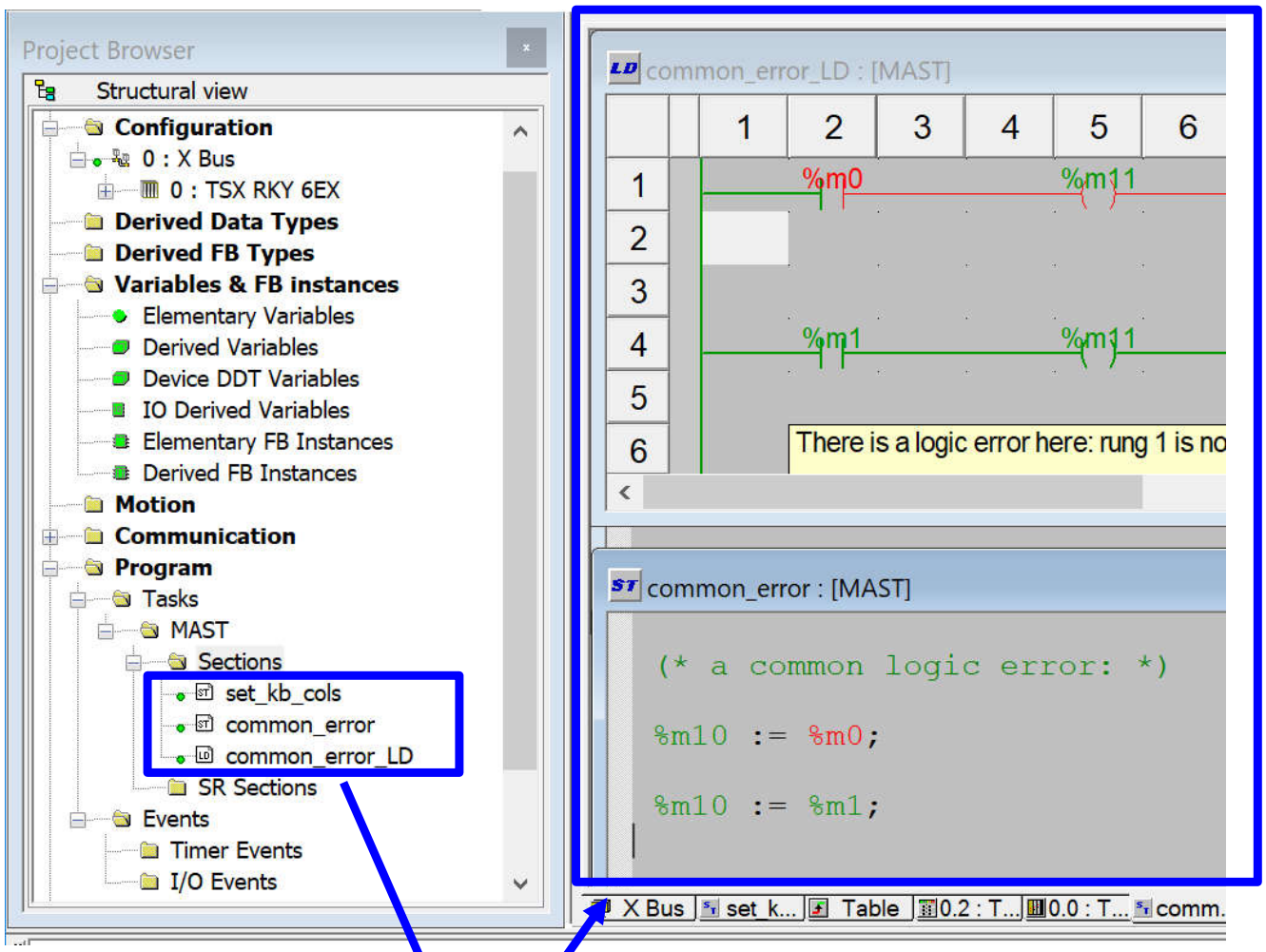


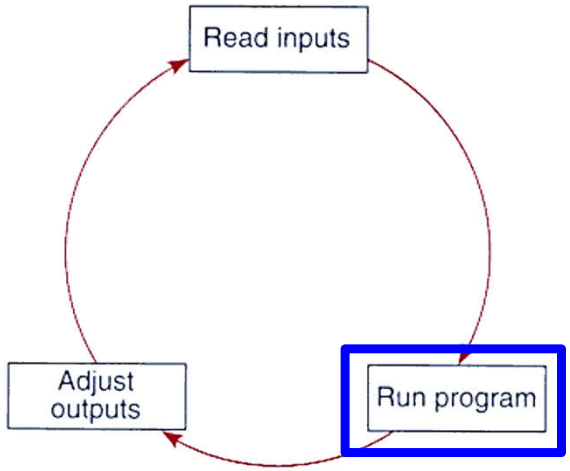
Noting %m0 is FALSE
why do we have %m11 and %m10 = TRUE?



Noting %m0 is TRUE
why do we have %m10 and %m11 = FALSE?

A very common programming error:





*Real outputs are what the hardware connected to the PLC see and what **you see on screen**.*

Detail: The first assignment

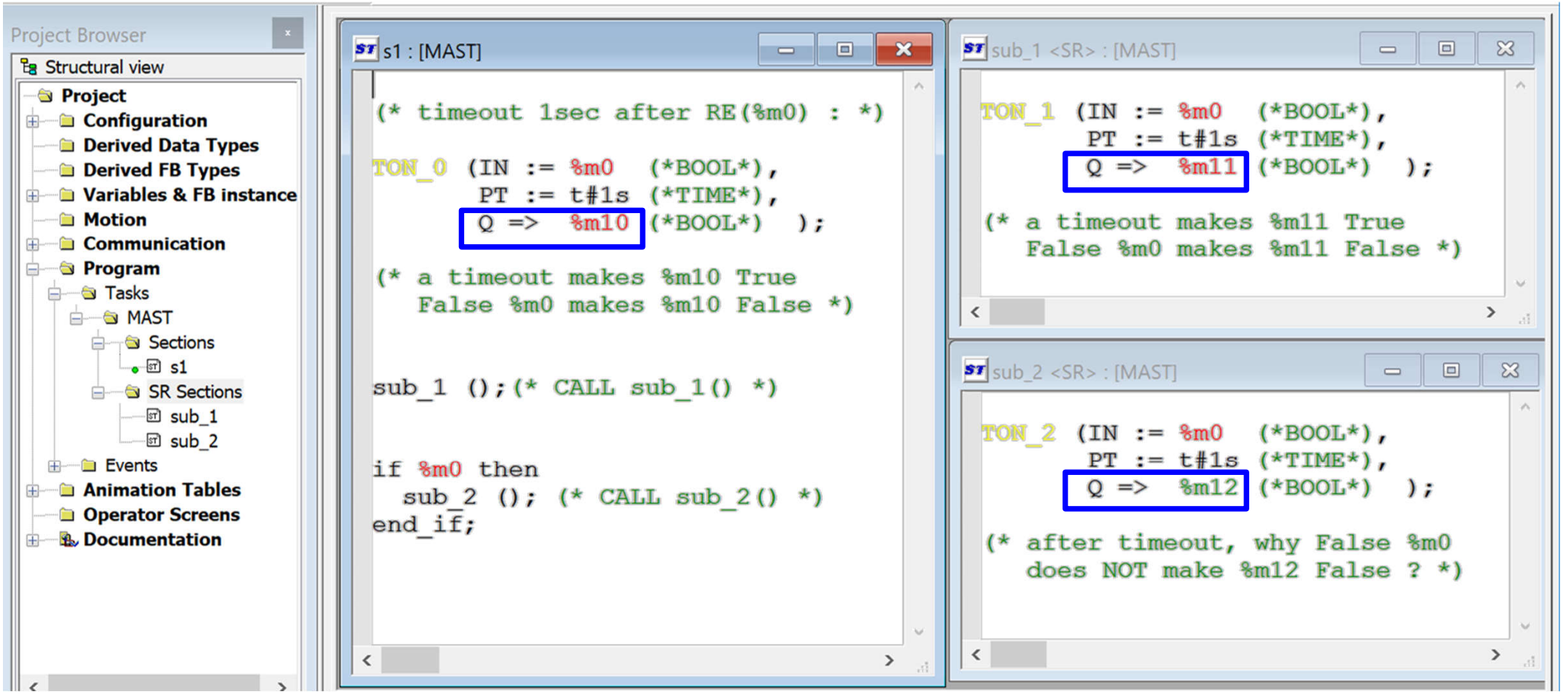
```
%m10 := %m0;
```

is overwritten by the second

```
%m10 := %m1;
```

All this code is executed in 1 single scan cycle

2. Timers in subroutines not running are not reset



False %m0 is making false %m11, as expected.

However, we also see
False %m0 and True %m12. Why is %m12 True?

One timer can be called multiple times

```
(*
   After declaring a timer in FB instances, PT needs to be set.
   Setup timer to start on %M0 and timeout on %M10.
   Note: no need to include arg ET of type TIME.
*)
TON_0 (IN := %m0  (*BOOL*),
       PT := t#3s (*TIME*),
       Q  => %m10 (*BOOL*));

(* Use timer to timeout also on %m11 *)
TON_0 (Q => %m11 (*BOOL*));

(* Use timer name "as a structure" *)
%m12 := TON_0.Q;

(* Use a separate call to reset timer *)
if %m1 then
  TON_0( IN := False );
end_if;

(* Auto reset if %m3 is True. Use it to toggle %M13. *)
if %m3 AND %m10 then
  TON_0( IN := False );
  %m13 := NOT(%m13);
end_if;

(* Use a separate call to redefine PT *)
if %m2 then
  TON_0( PT := t#1s );
end_if;
```