

Industrial Automation

(Automação de Processos Industriais)

PLC Programming languages

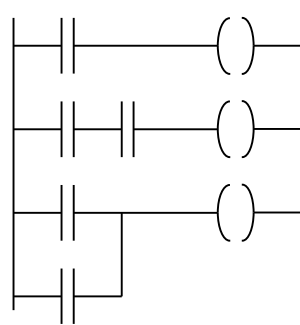
Common Programming Errors

<http://www.isr.tecnico.ulisboa.pt/~jag/courses/api20b/api2021.html>

Prof. José Gaspar, rev. 2020/2021

PLC Programming Languages (IEC 61131-3)

Ladder Diagram



Structured Text

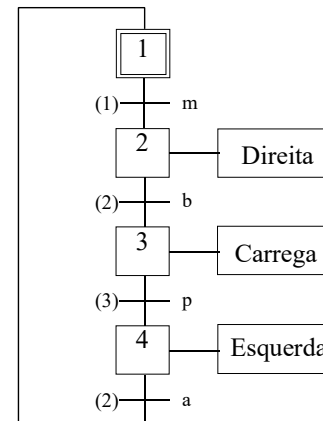
```

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:

LD common_error_LD : [MAST]

	1	2	3	4	5	6
1	%m0		%m11			
2						
3						
4	%m1		%m11			
5						
6						

There is a logic error here: rung 1 is not

ST common_error : [MAST]

```
(* a common logic error: *)
%m10 := %m0;
%m10 := %m1;
```

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

LD common_error_LD : [MAST]

	1	2	3	4	5	6
1	%m0		%m11			
2						
3						
4	%m1		%m11			
5						
6						

There is a logic error here: rung 1 is not

ST common_error : [MAST]

```
(* a common logic error: *)
%m10 := %m0;
%m10 := %m1;
```

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

A very common programming error:

Project Browser

Structural view

- Configuration
 - 0 : X Bus
 - 0 : TSX RKY 6EX
- Derived Data Types
- Derived FB Types
- Variables & FB instances
 - Elementary Variables
 - Derived Variables
 - Device DDT Variables
 - IO Derived Variables
 - Elementary FB Instances
 - Derived FB Instances
- Motion
- Communication
- Program
 - Tasks
 - MAST
 - Sections
 - set_kb_cols
 - common_error
 - common_error_LD
 - SR Sections
 - Events
 - Timer Events
 - I/O Events

LD common_error_LD : [MAST]

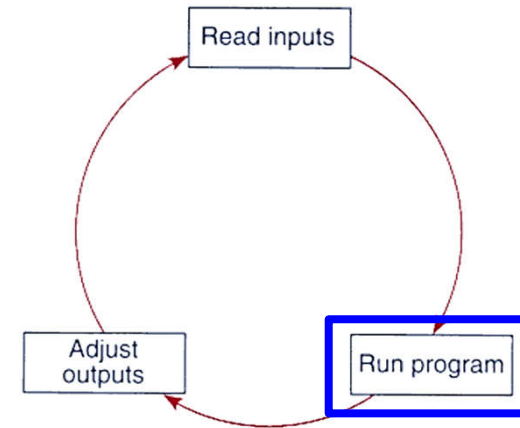
	1	2	3	4	5	6
1	%m0		%m1			
2						
3						
4	%m1		%m1			
5						
6	There is a logic error here: rung 1 is no					

ST common_error : [MAST]

```
(* a common logic error: *)  
%m10 := %m0;  
%m10 := %m1;
```

X Bus | set k... | Table | 0.2 : T... | 0.0 : T... | comm.

Code to run in 1 single scan cycle



Adjusted outputs is what the hardware connected to the PLC see as IO and also what you see on screen.

Note: The first assignment %m10 := %m0; is overwritten by the second %m10 := %m1;

***2. Timers in subroutines not running
are not reset***

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;
```