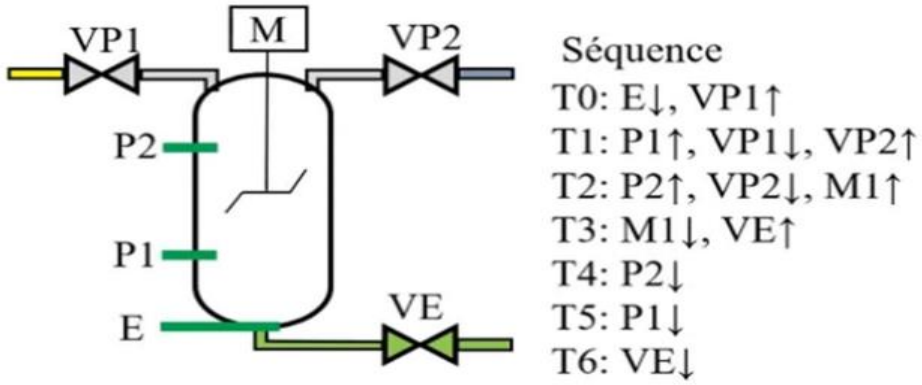
Asean-factori

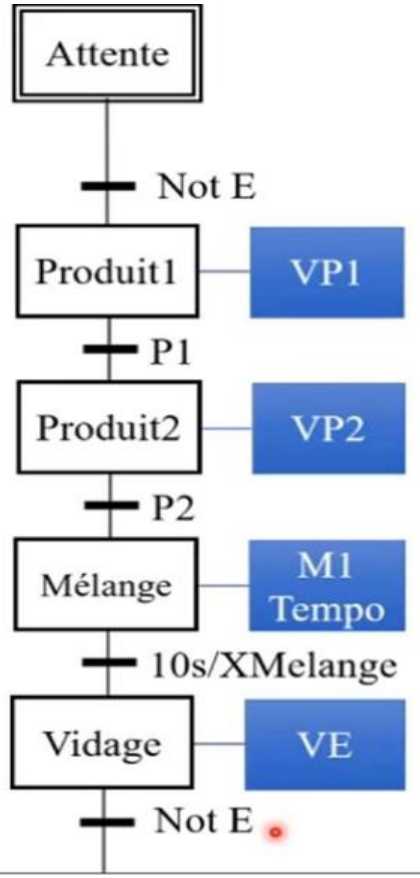
Lab Siemens (Following2.)
Grafcet Implementation

Autor : KAMMAH Oumaima
Supervised by: Jean-Marc THIRIET
Asean-Factori (grenoble-inp.fr)

- This set of slides follows the set of slides 2.PLC with the implementation of the following Grafcet



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑
T3: M1↓, VE↑
T4: P2↓
T5: P1↓
T6: VE↓



Insert inputs /Output in TIA POTAL

PLC Tags ⇨ Add new tag table

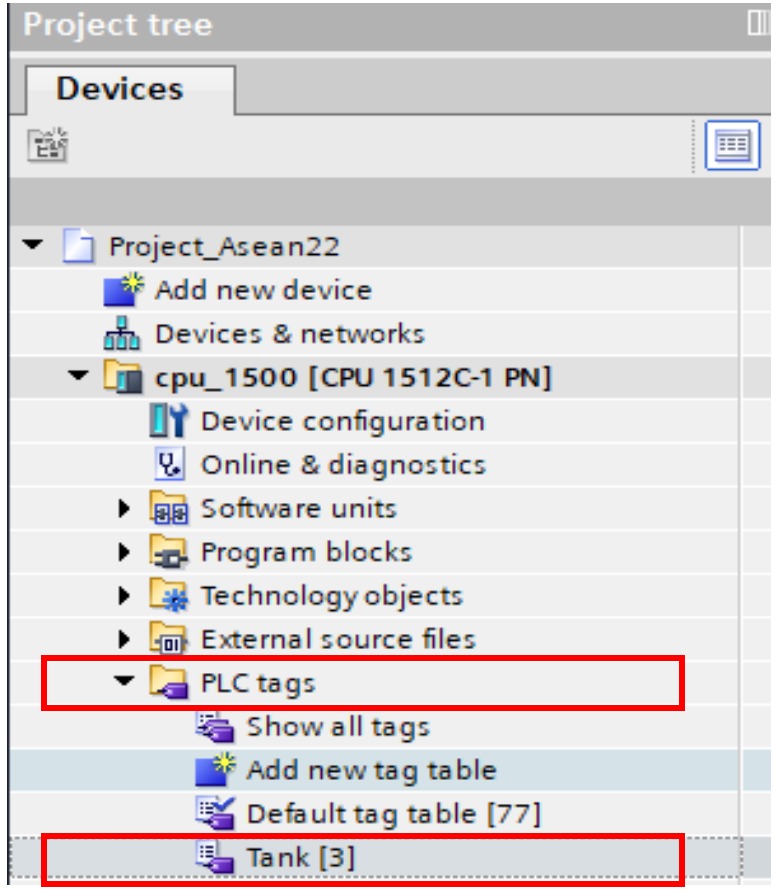
- Definitions of the 4 outputs (same manners)

Tank

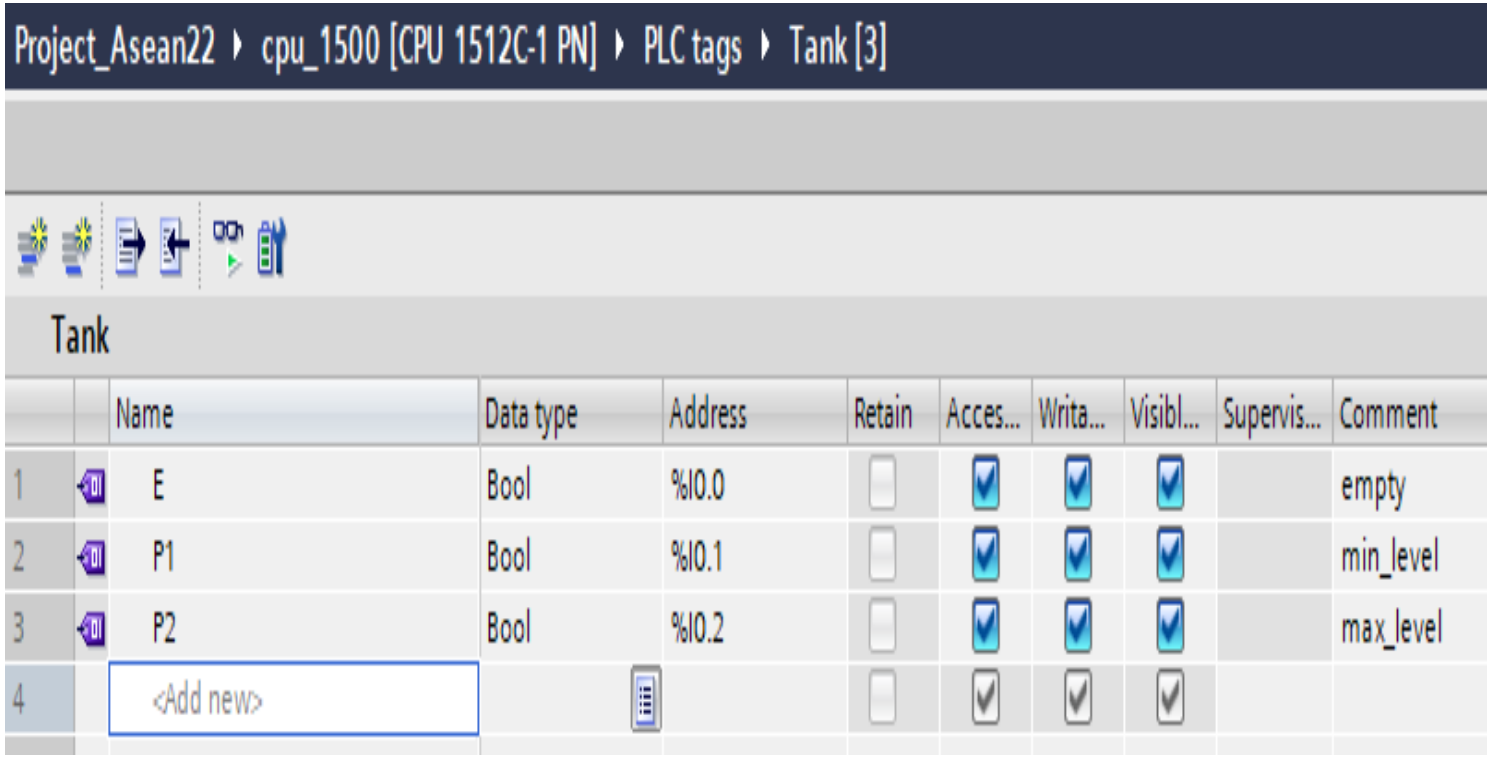
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	Comment
1	E	Bool	%I0.0		☑	☑	☑		empty
2	P1	Bool	%I0.1		☑	☑	☑		min_level
3	P2	Bool	%I0.2		☑	☑	☑		max_level
4	M	Bool	%Q0.3		☑	☑	☑		Motor
5	VP1	Bool	%Q0.4		☑	☑	☑		valve_yellow
6	VP2	Bool	%Q0.5		☑	☑	☑		valve_blue
7	VE	Bool	%Q0.7		☑	☑	☑		drain valve(green)

Insert inputs /Output in TIA POTAL

PLC Tags ⇨ Add new tag table



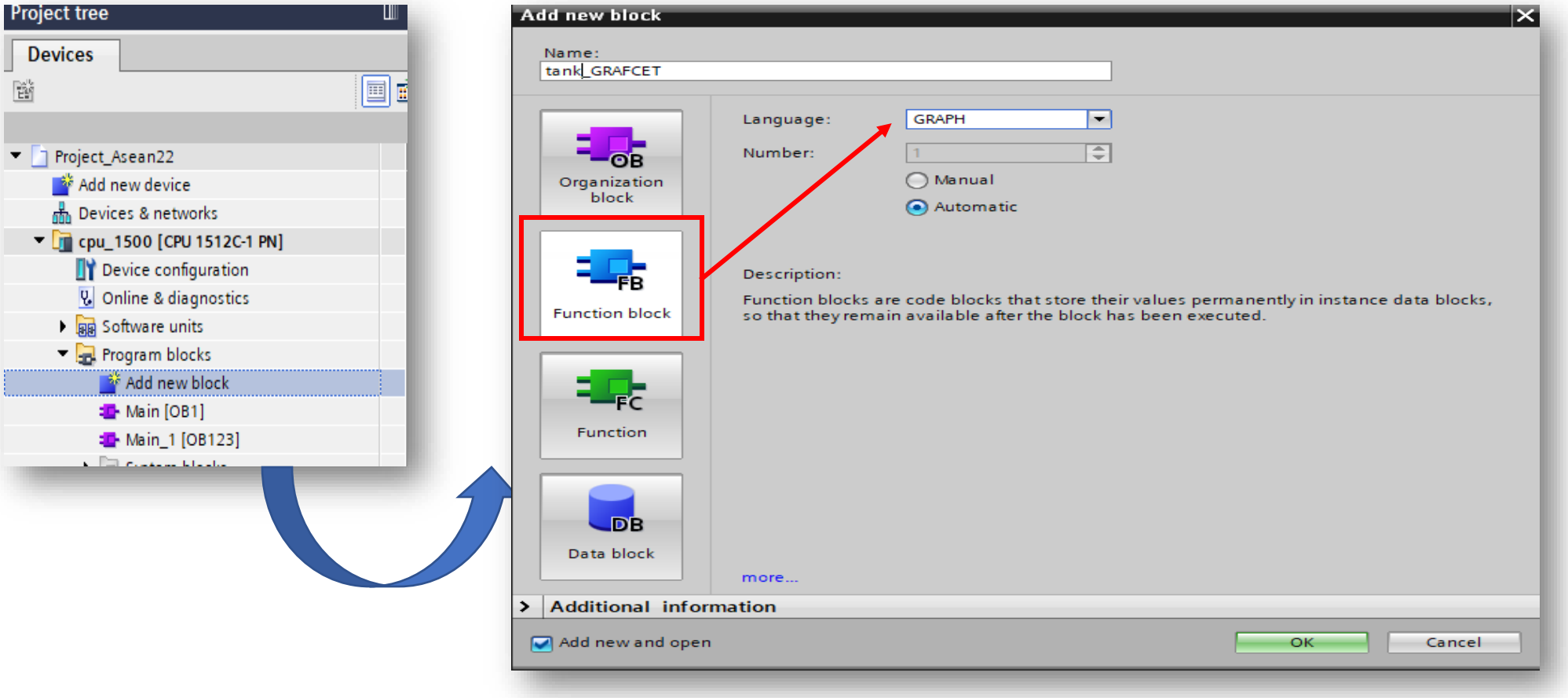
- Add Oue 3 inputs



Implementation of the Grafcet programmes

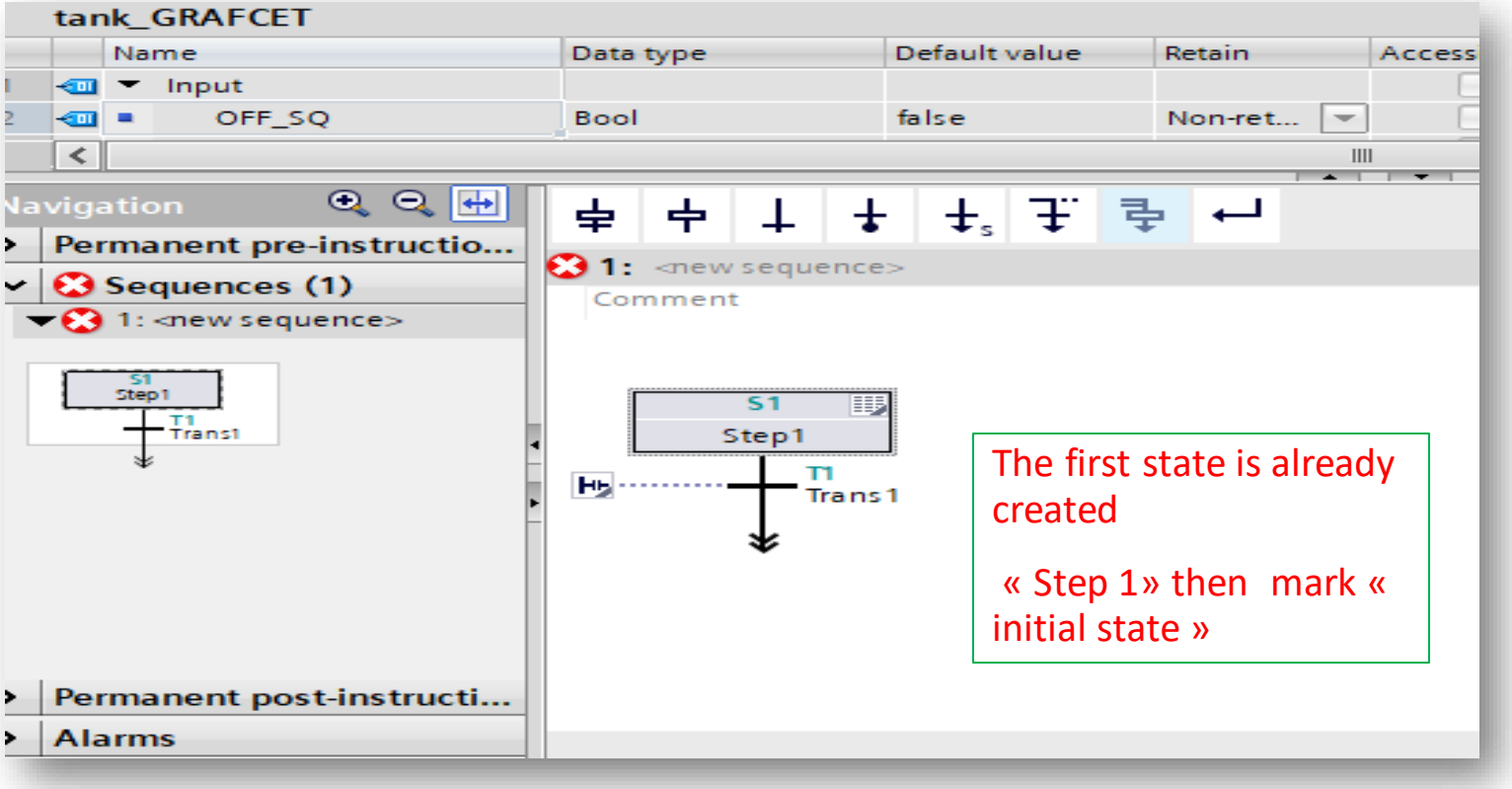
To create our Grafcet on TIA Portal v15(Siemens)

- Click on Programs blocks --> Add new block --> Define Name--> Select SCL Languge



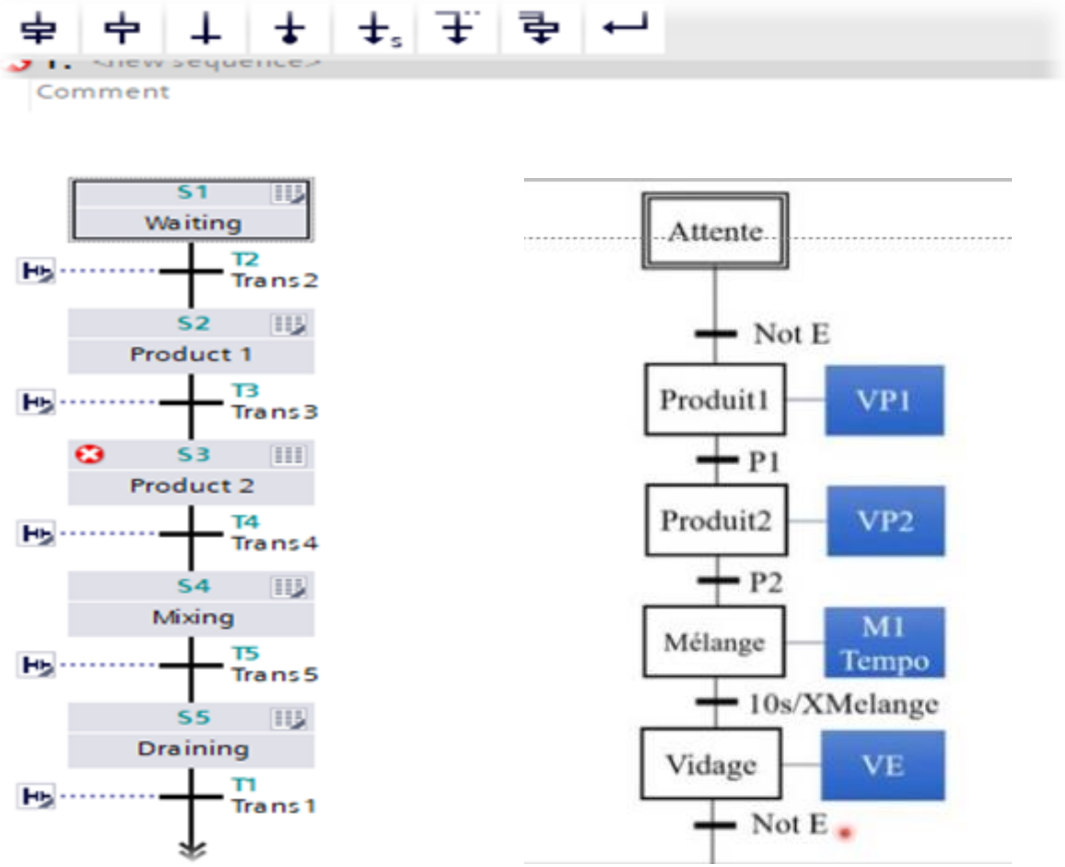
Implementation of the Grafcet programmes

- Click on Programs blocks --> Add new block --> Define Name--> Select SCL Language
- Click OK--> The editing area for the steps and transitions is opened-->We start drawing our behavioural part of the Grafcet



Implementation of the Grafcet programmes

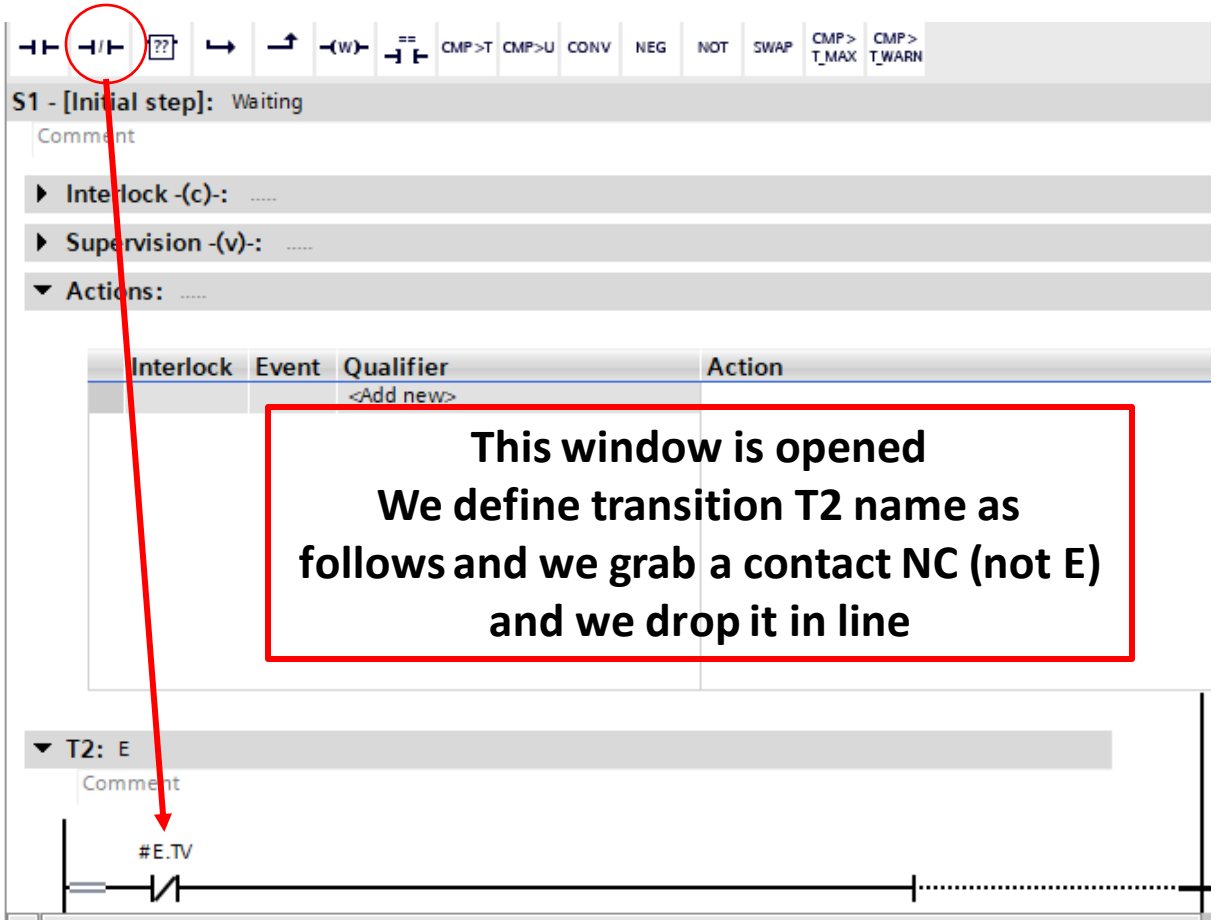
- Click on Programs blocks --> Add new block --> Define Name--> Select SCL Language
- Click OK--> The editing area for the steps and transitions is opened-->We start drawing our behavioural part of the Grafcet



•Creation in the chart of the states equivalent states of « Produit1 »,« Produit2 », « Melange » and « **Vidange** »

Implementation of the Grafcet programmes

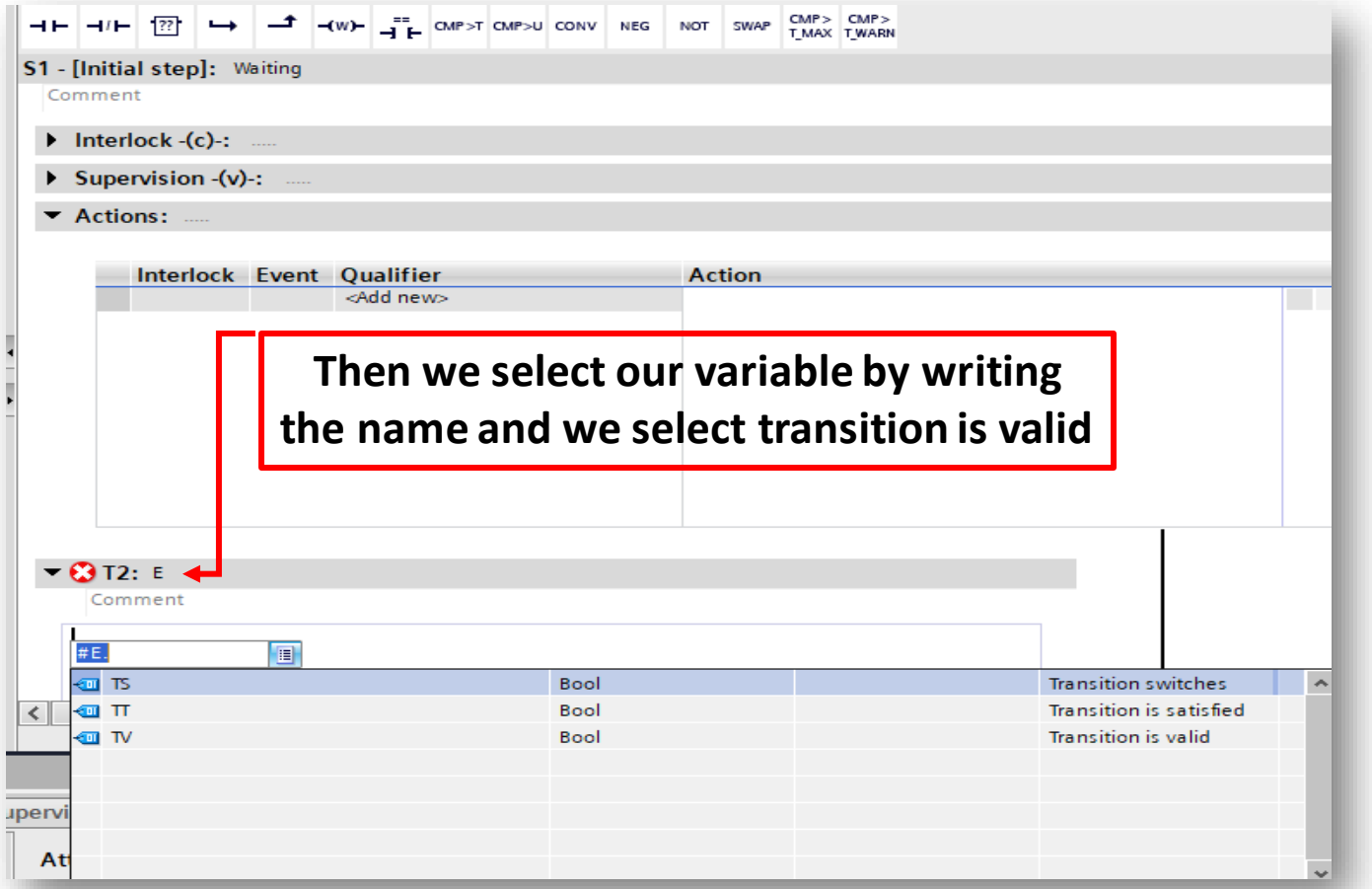
•Implementation of the transitions by double clicking between two states on transition



Implementation of the Grafcet programmes

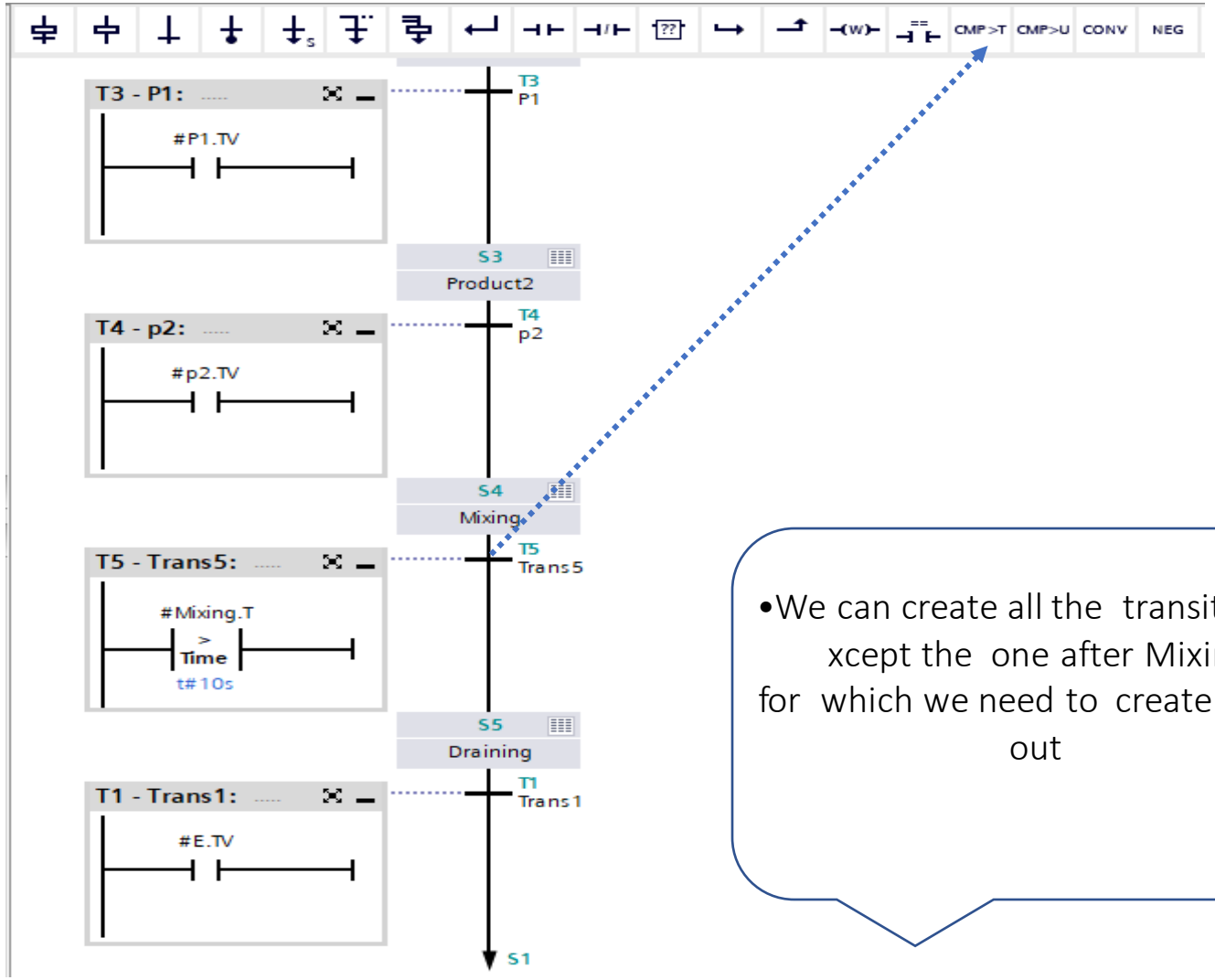
•Implementation of the transitions by double clicking btween two states on transition

•First transition is « E »
•It should be « inversed » because this is actually « not E »



Implementation of the Grafcet programmes

•Implementation of the transitions by double clicking between two states on transition

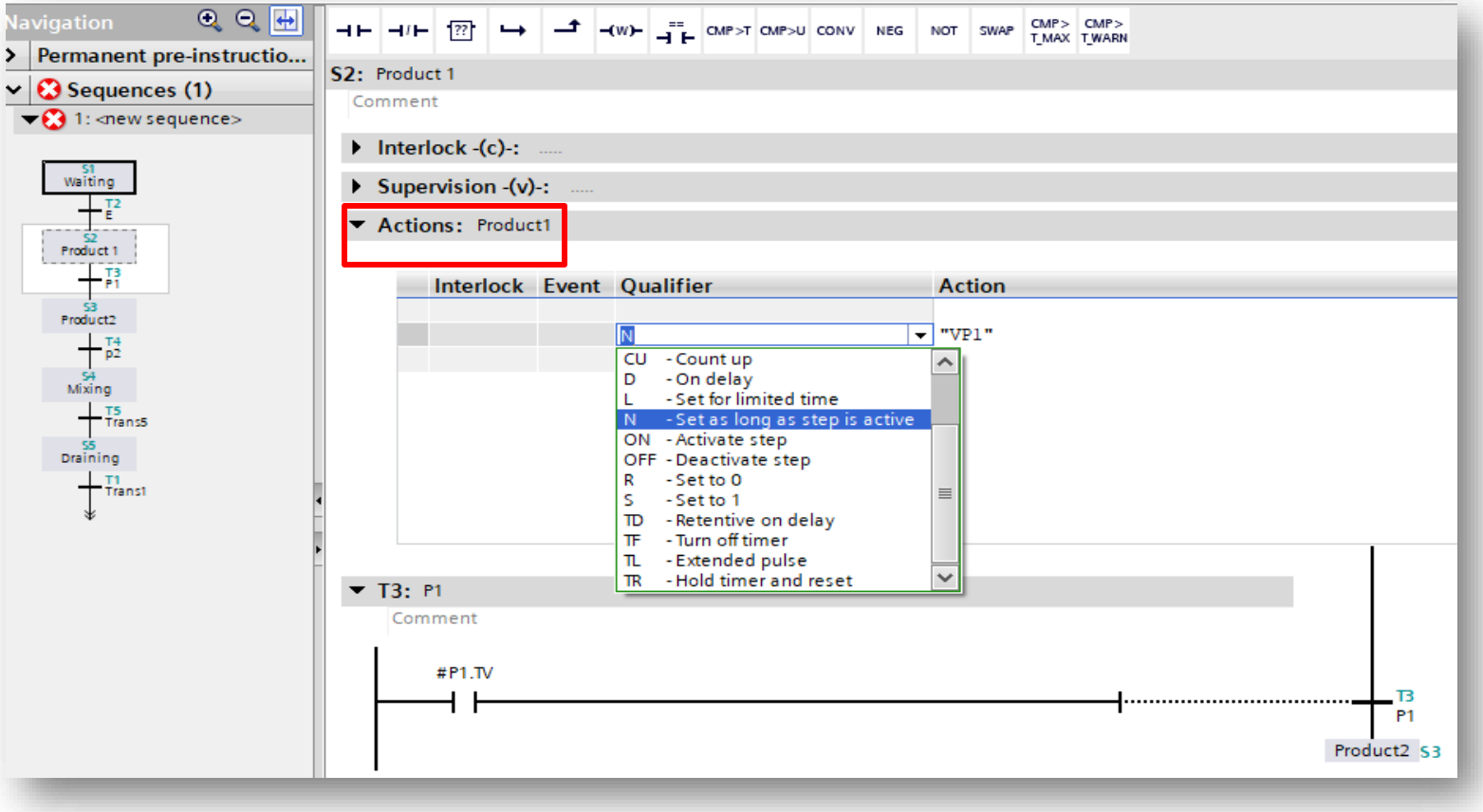


Mixing is running for 10s as soon as time is bigger than 10s mixing is desactivate

•We can create all the transitions, e xcept the one after Mixing for which we need to create a time out

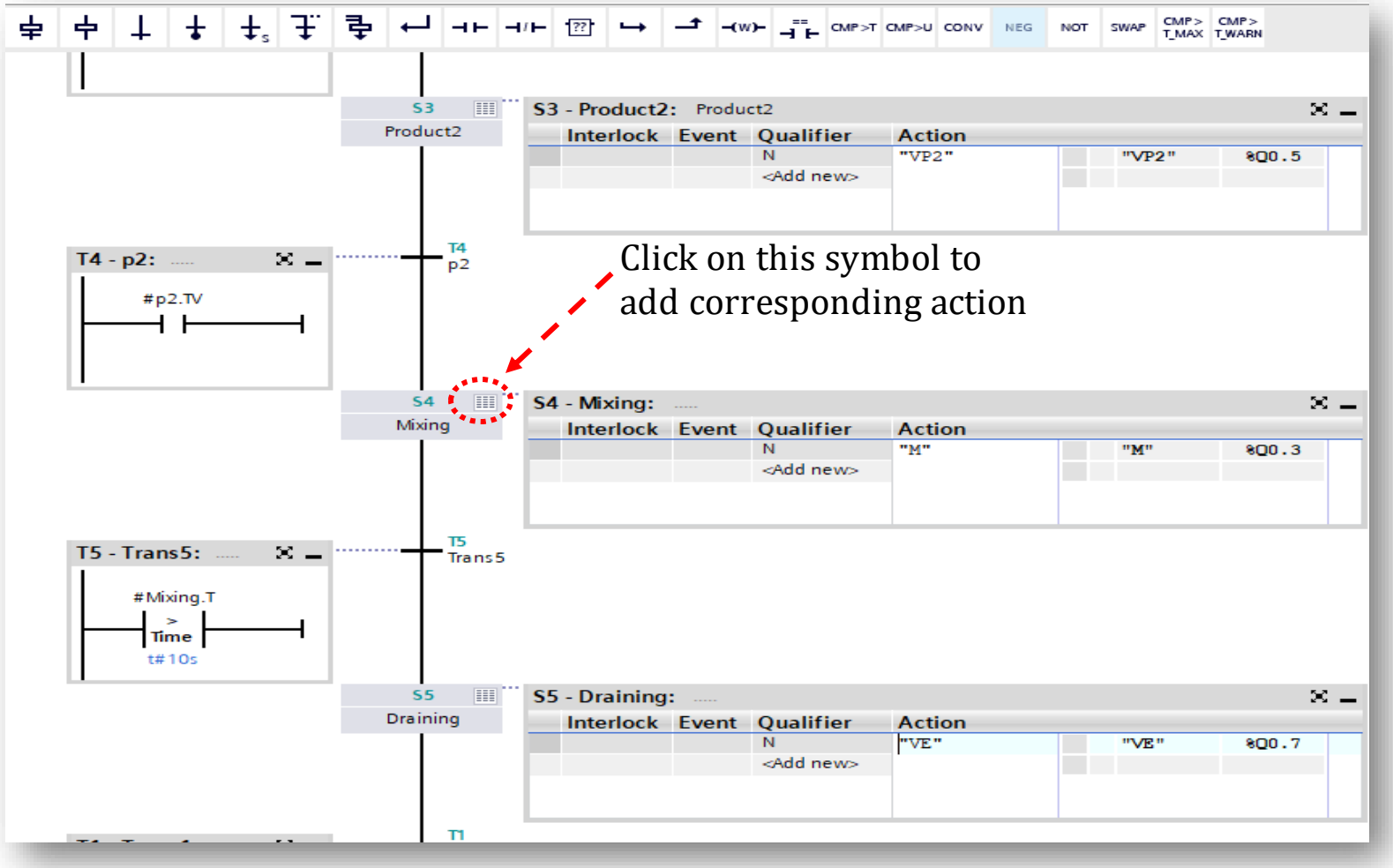
Implementation of the Grafcet programmes

- Implementation of the Actions by double clicking on corresponding step

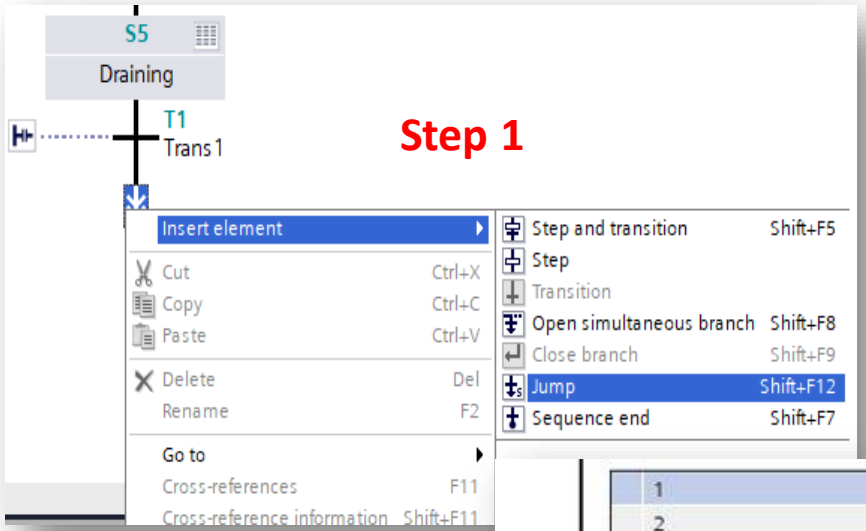


Implementation of the Grafcet programmes

- Implementation of the Actions directly by clicking right on the top of state

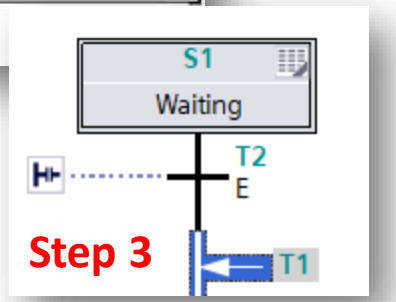


Implementation of the Grafcet programmes



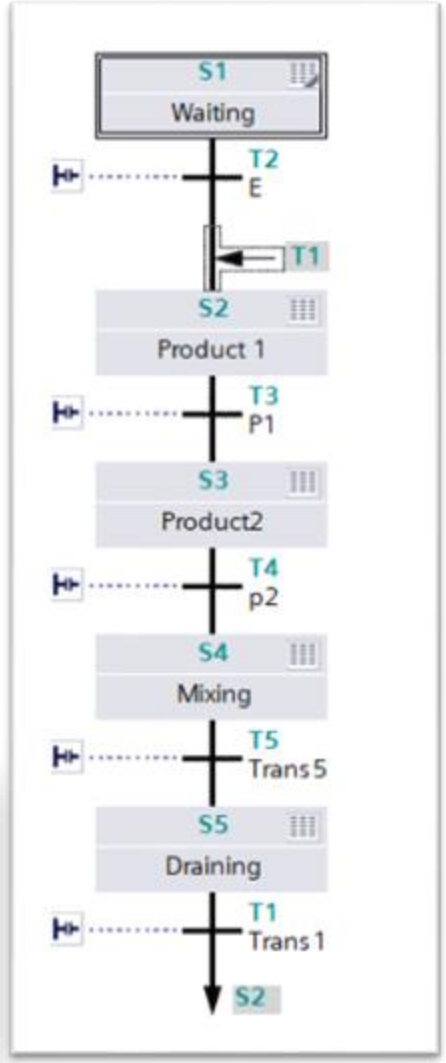
Step 1

Step 2

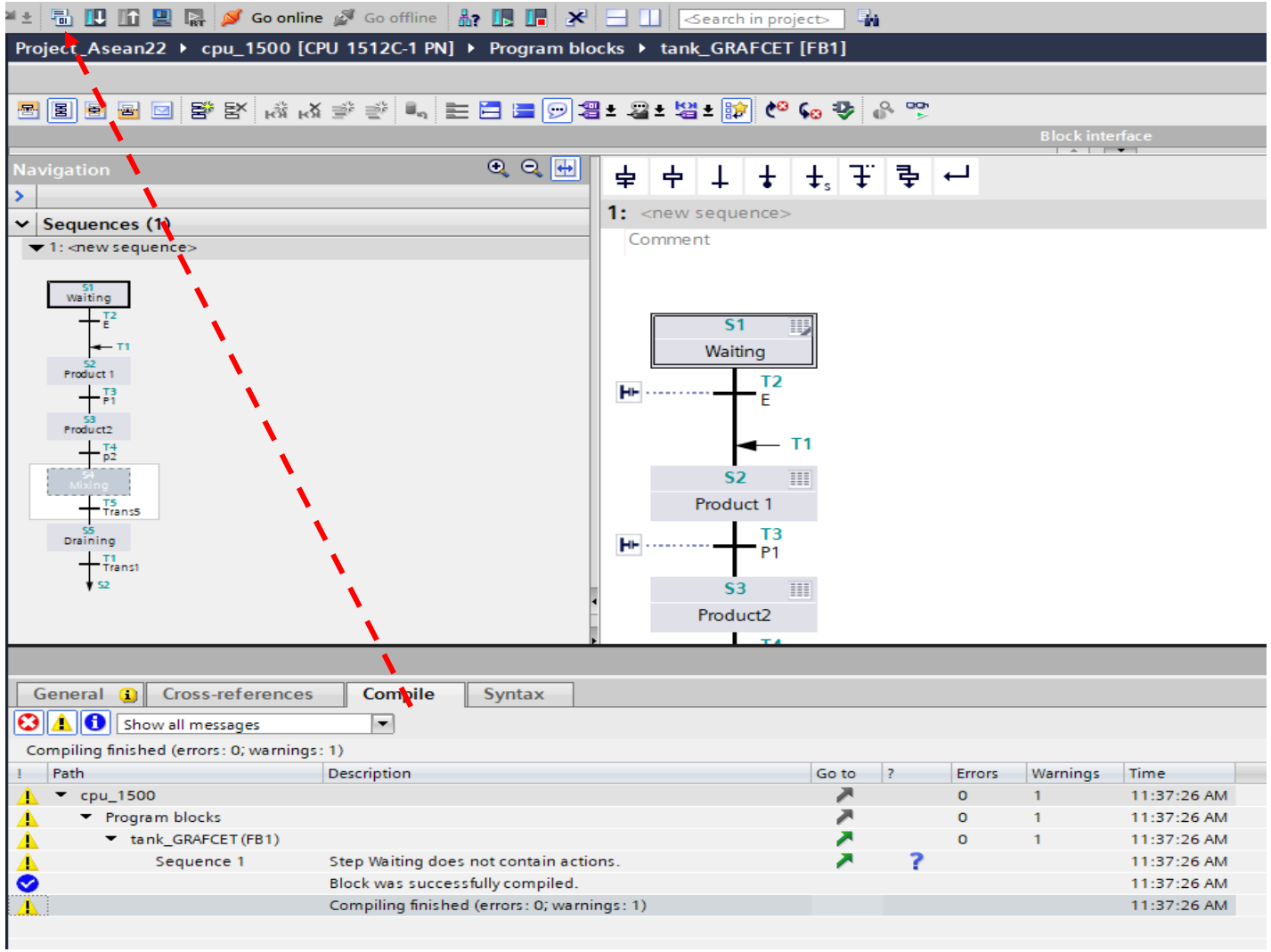


Step 3

•We close the Grafcet by using the tool « Jump to»as follows;

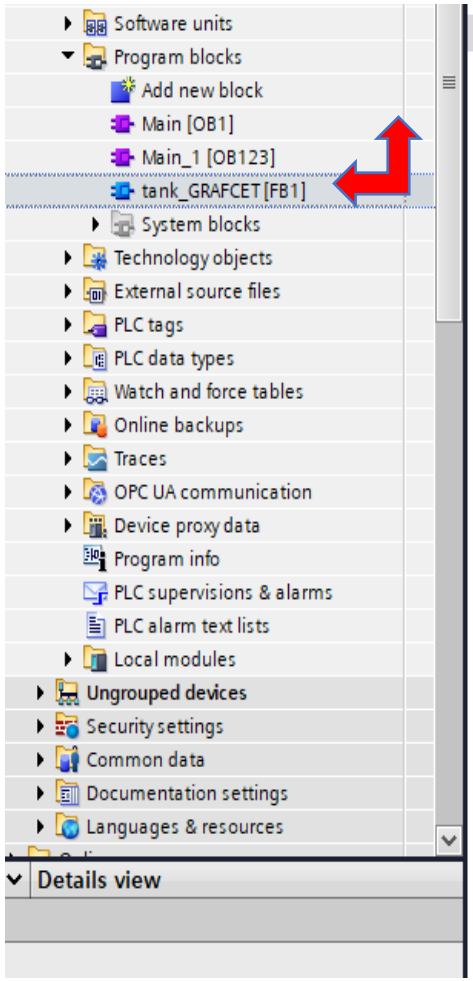


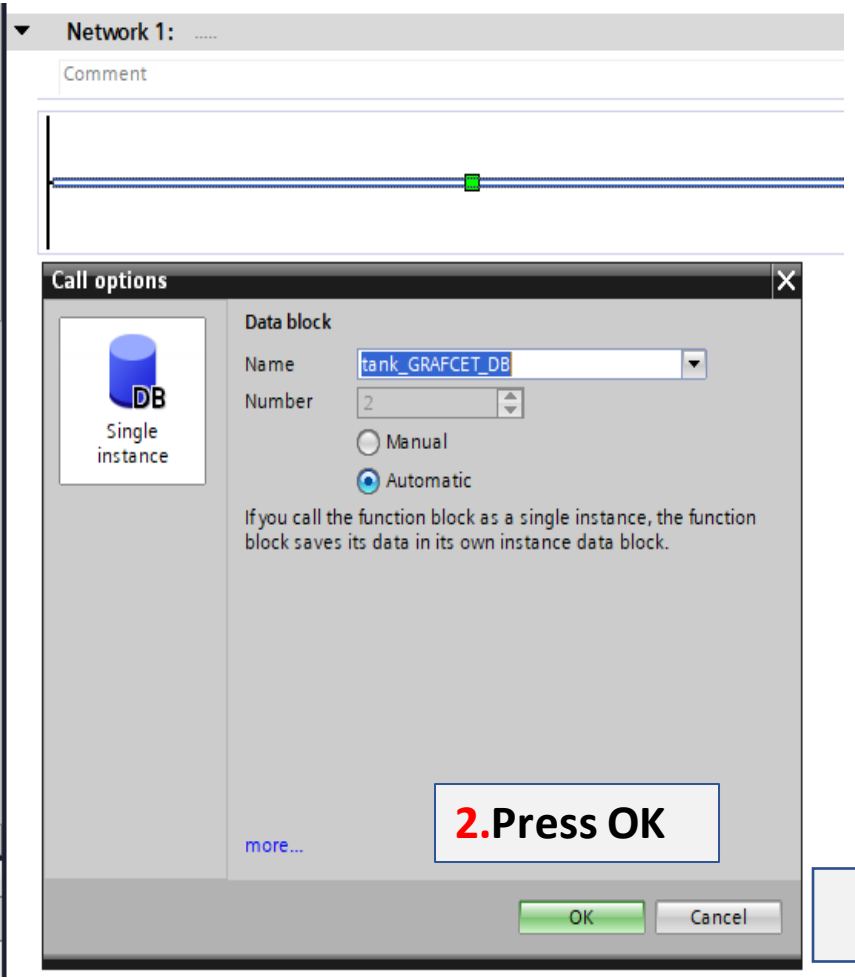
Compiling Our Grafcet



Zero error ==>Compilation done correctly

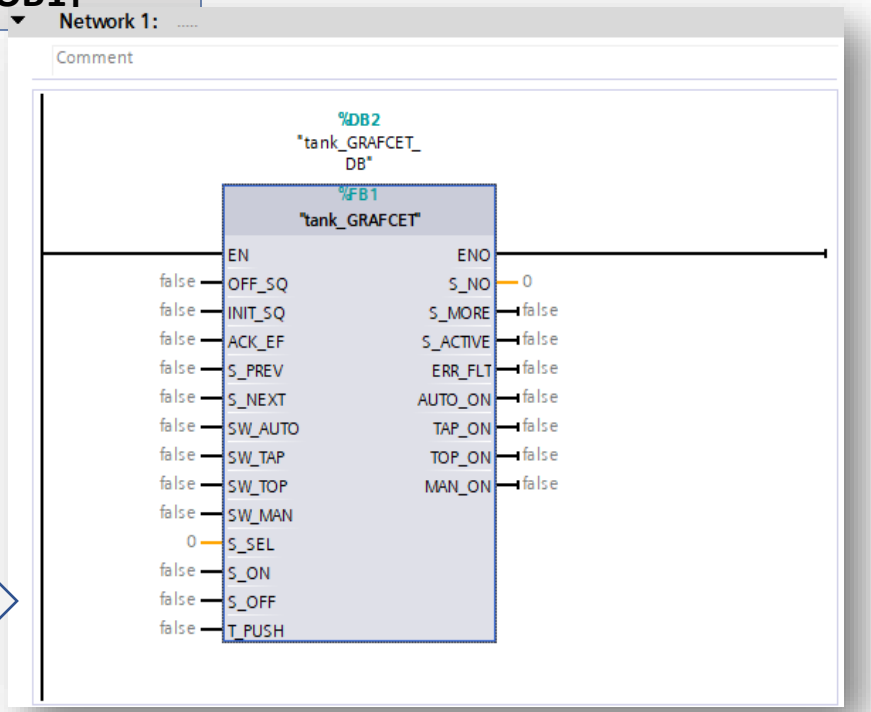
Implementation of the actions in ladder



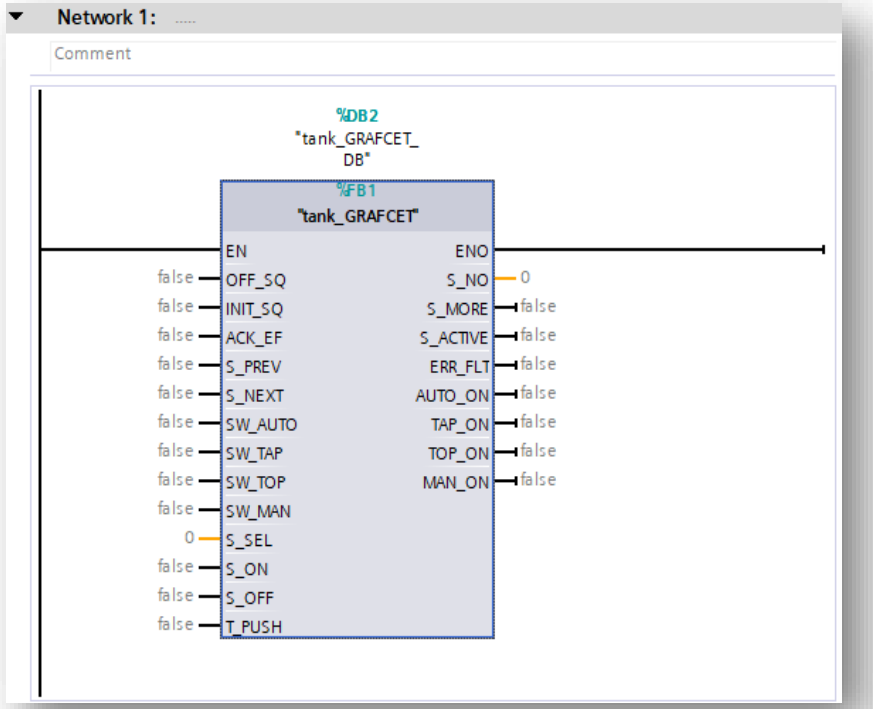


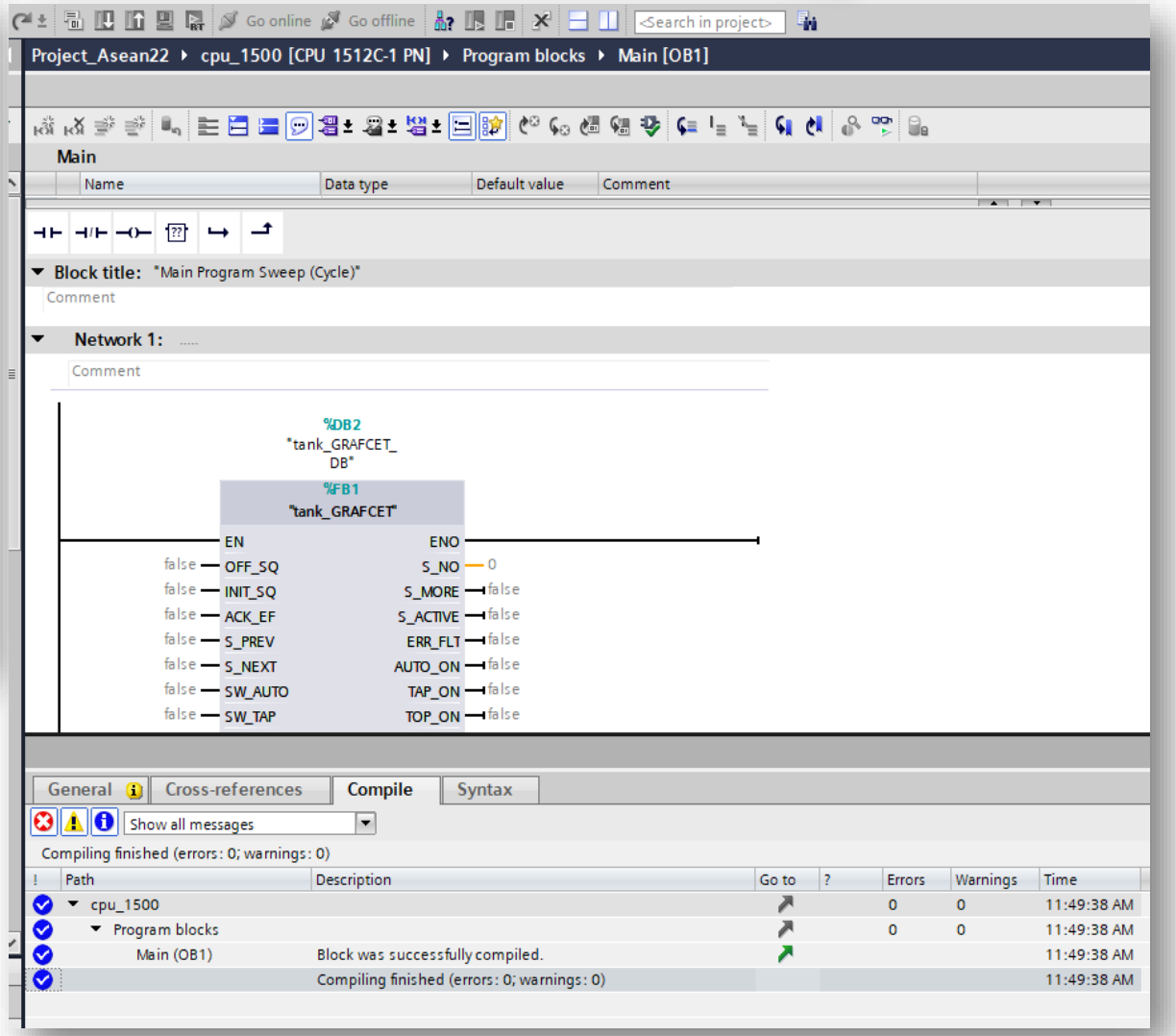
1 .Drag and drop our tnak_GRAFCET into network 1 of Main[OB1]

2.Press OK



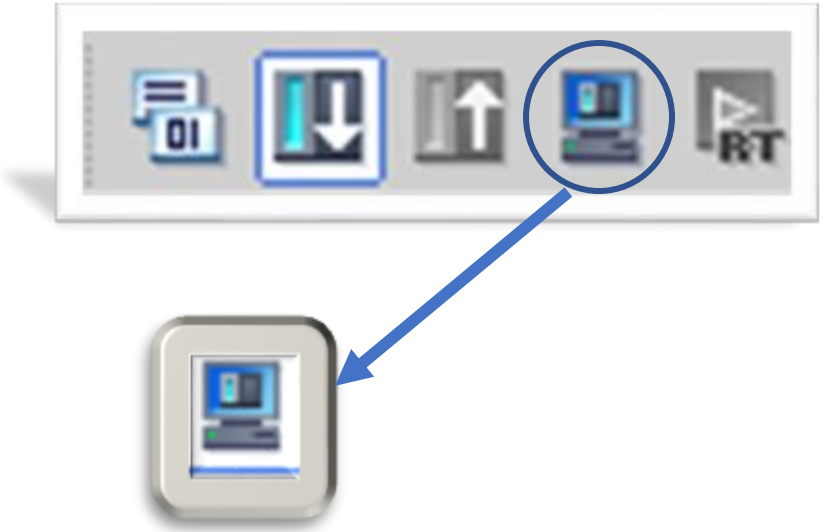
1.Compile block created



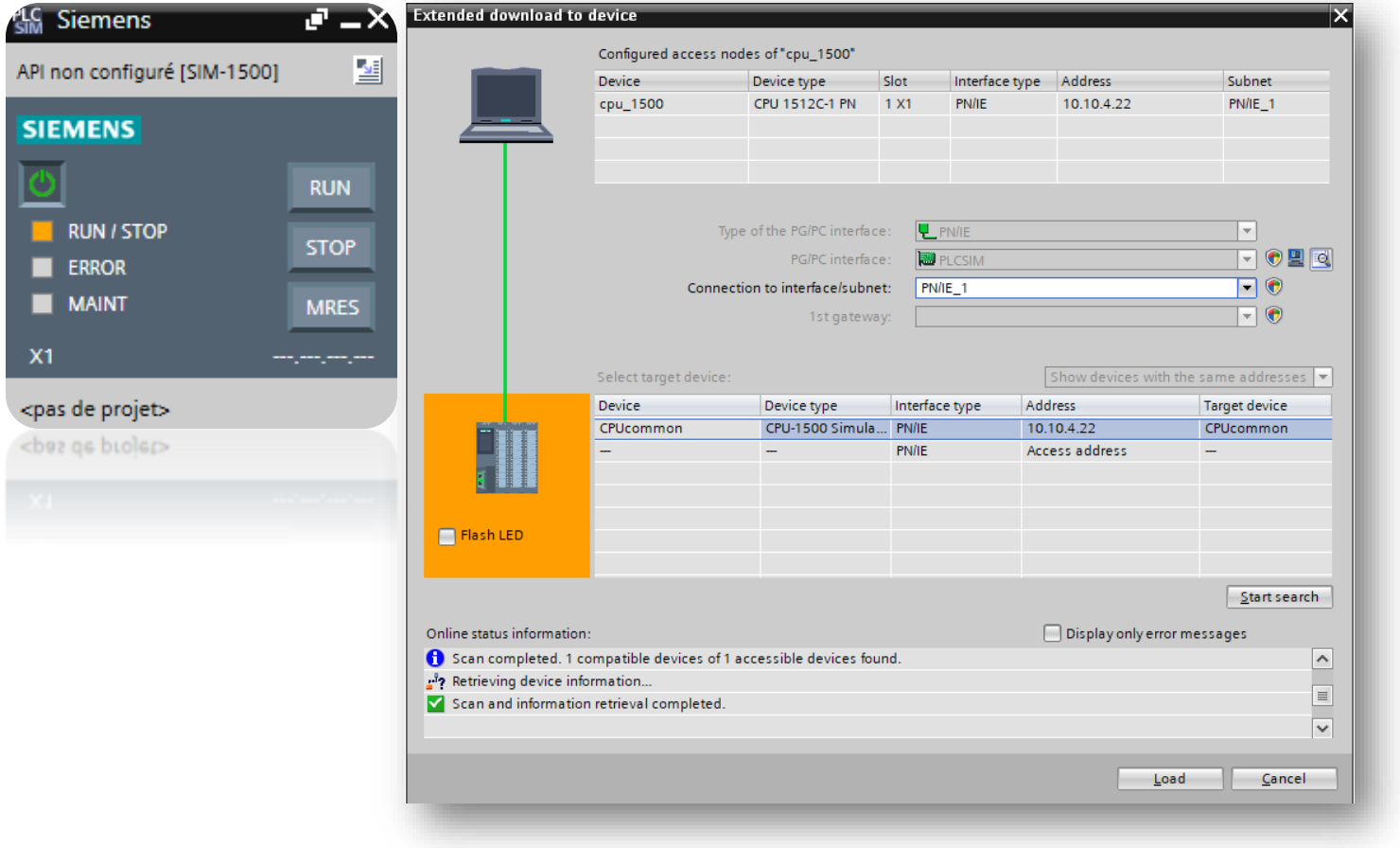


Zero error ==>Compilation done correctly

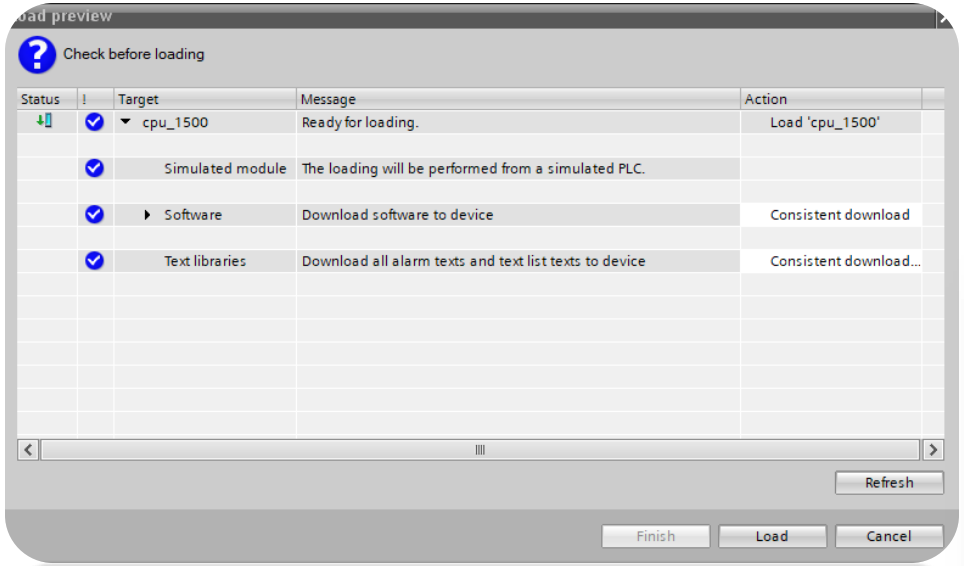
2.1)Simulation by PLCSIM (corresponding virtual PLC)



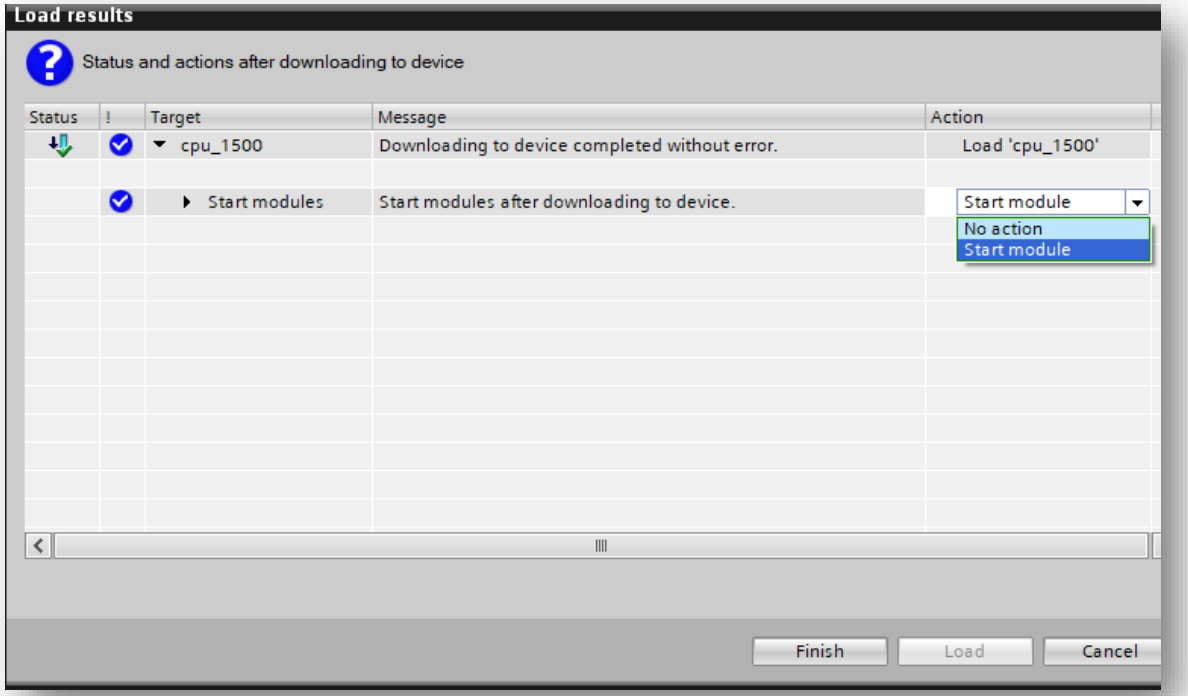
=> Click on Start simulation icon
=>Select PLCSIM in PG/PC interface
=>Start search
=>Select Device 10.10.100.113
=>Load



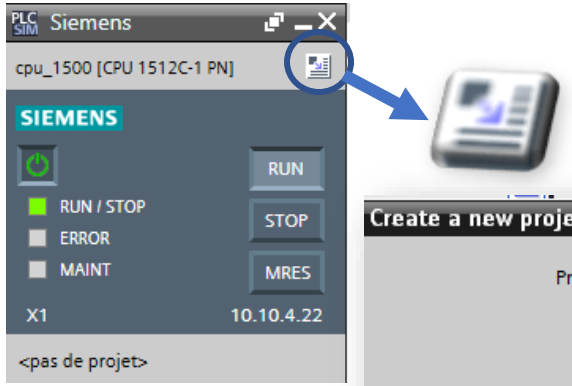
2.2)Simulation by PLCSIM (corresponding virtual PLC)

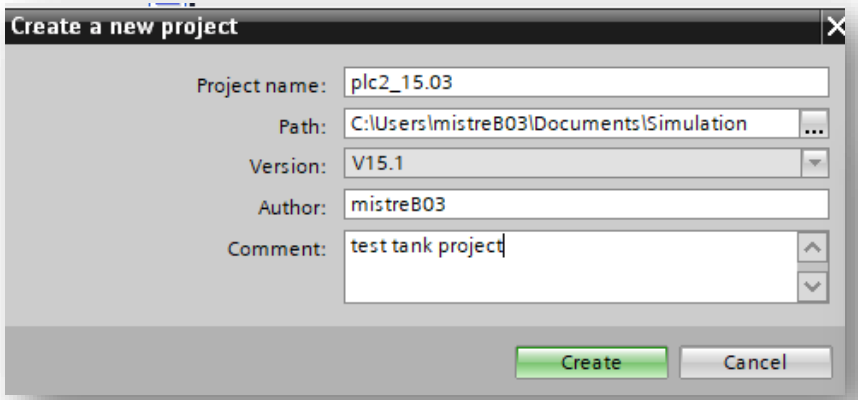


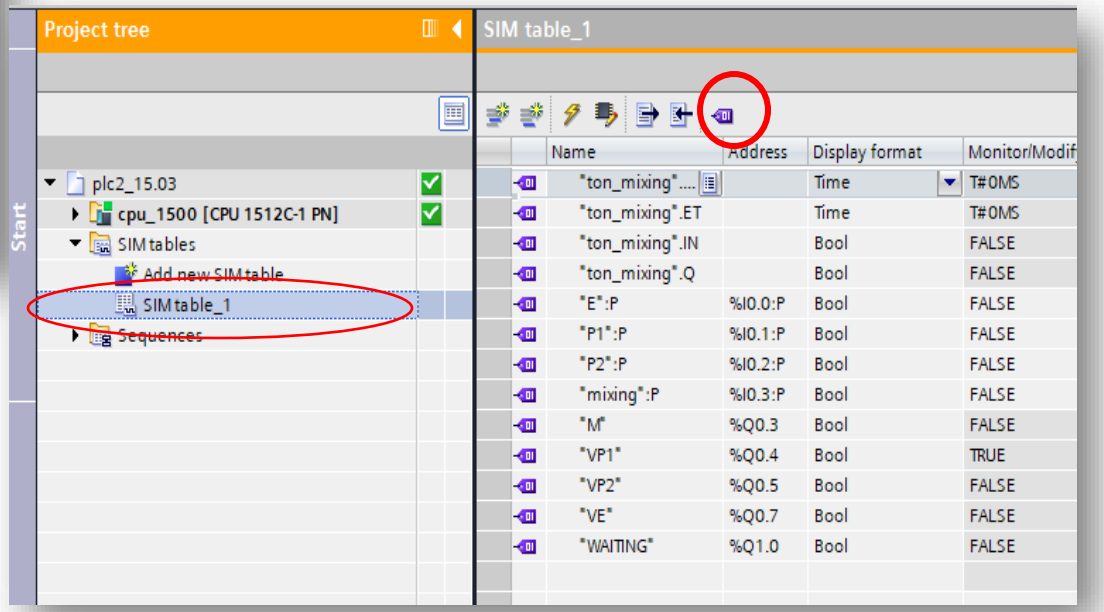
=> Load preview window is open
=>Click on Load button
=>Load results window is open
=>Select Start module
=>Click on Finish



2.3)Simulation by PLCSIM (corresponding virtual PLC)



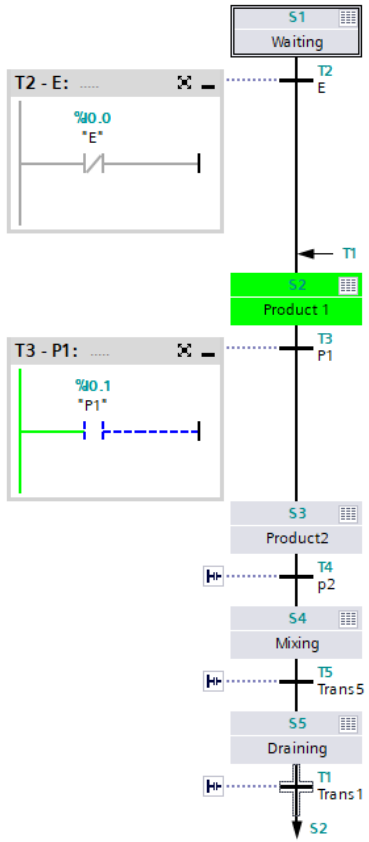


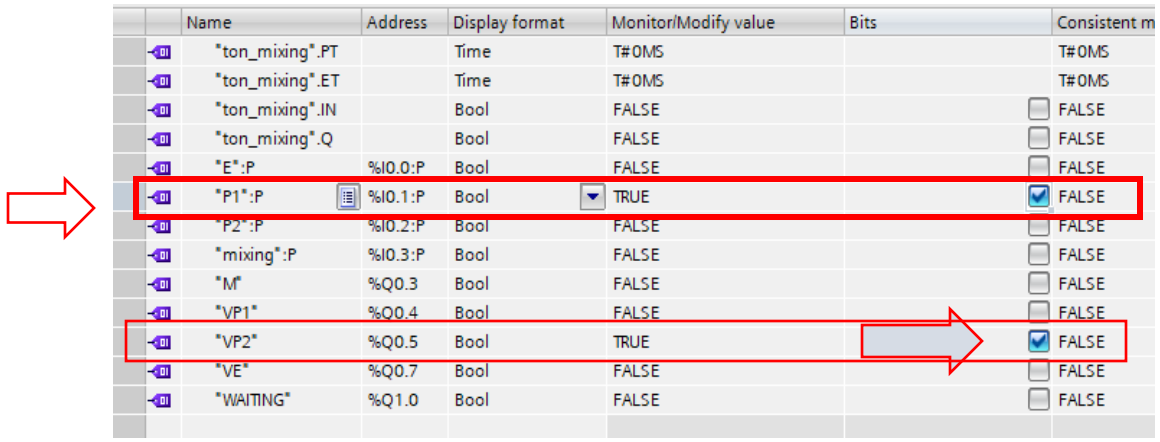


=> Then click on the icon right on the top
=> PLCSIM is open
=> Create a new project
=>load tags

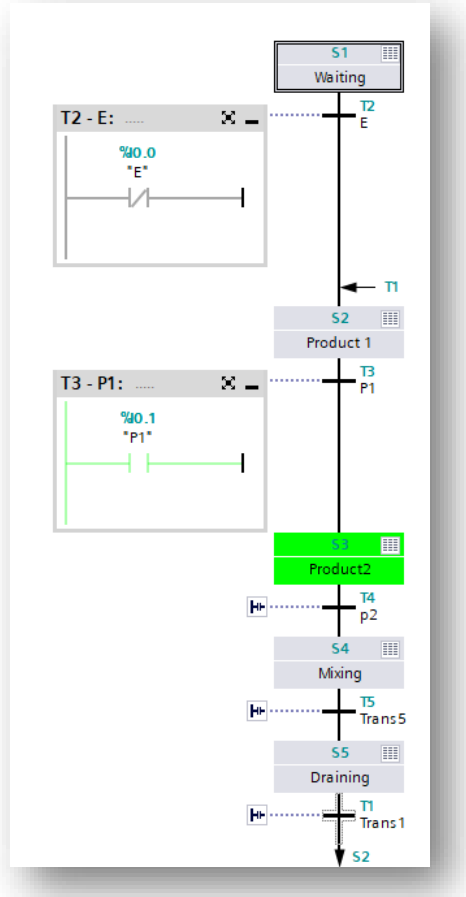
2.4)Go online

=> Click on go online icon
=> Force P1 to true
=>valve of Product 1 switch OFF => valve Product2 switch ON



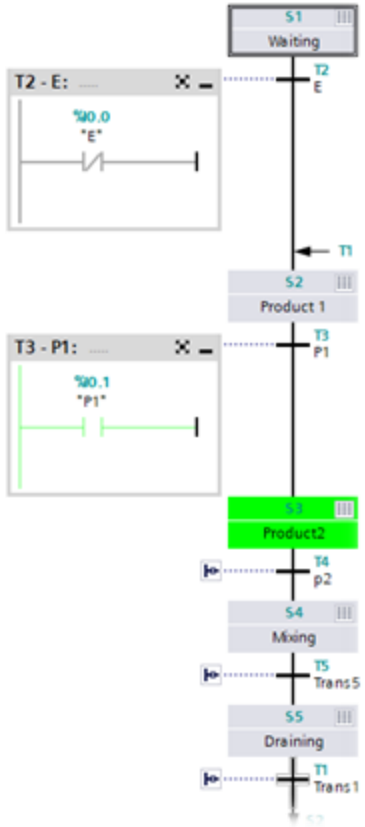


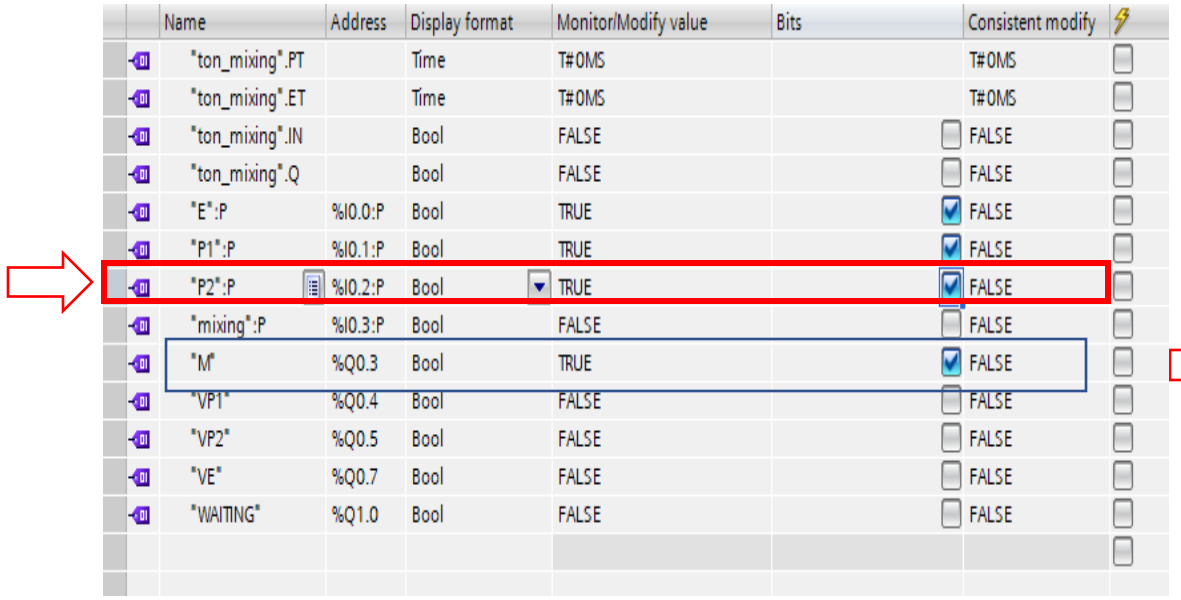
	Name	Address	Display format	Monitor/Modify value	Bits	Consistent m
	"ton_mixing".PT		Time	T#0MS		T#0MS
	"ton_mixing".ET		Time	T#0MS		T#0MS
	"ton_mixing".IN		Bool	FALSE		FALSE
	"ton_mixing".Q		Bool	FALSE		FALSE
	"E":P	%I0.0:P	Bool	FALSE		FALSE
	"P1":P	%I0.1:P	Bool	TRUE		FALSE
	"P2":P	%I0.2:P	Bool	FALSE		FALSE
	"mixing":P	%I0.3:P	Bool	FALSE		FALSE
	"M"	%Q0.3	Bool	FALSE		FALSE
	"VP1"	%Q0.4	Bool	FALSE		FALSE
	"VP2"	%Q0.5	Bool	TRUE		FALSE
	"VE"	%Q0.7	Bool	FALSE		FALSE
	"WAITING"	%Q1.0	Bool	FALSE		FALSE



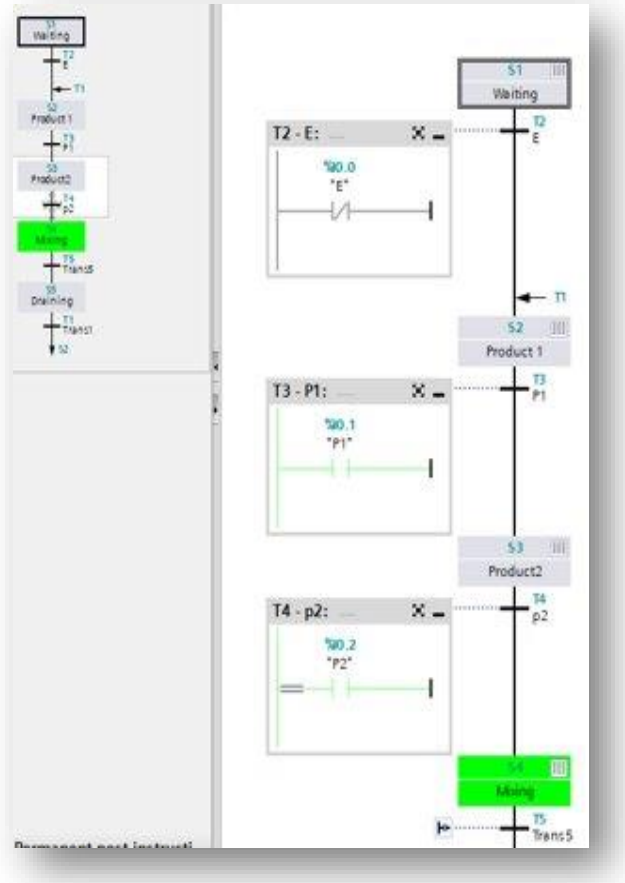
2.4)Go online

=> Force P2 to true to pass from state adding product 2 to mixing state



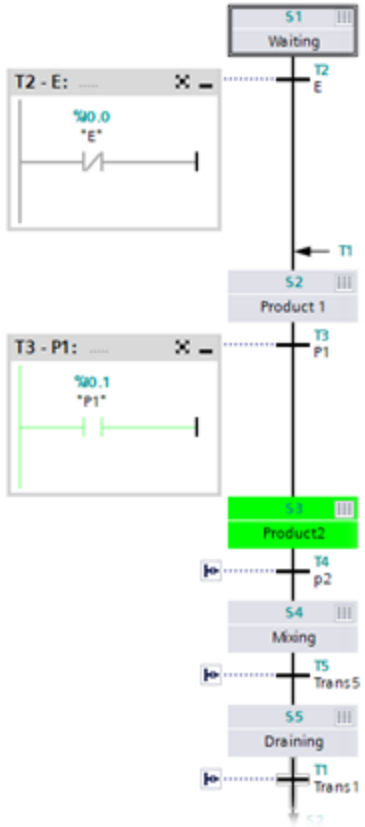


	Name	Address	Display format	Monitor/Modify value	Bits	Consistent modify
	"ton_mixing".PT		Time	T#0MS		T#0MS
	"ton_mixing".ET		Time	T#0MS		T#0MS
	"ton_mixing".IN		Bool	FALSE		FALSE
	"ton_mixing".Q		Bool	FALSE		FALSE
	"E":P	%I0.0:P	Bool	TRUE		FALSE
	"P1":P	%I0.1:P	Bool	TRUE		FALSE
	"P2":P	%I0.2:P	Bool	TRUE		FALSE
	"mixing":P	%I0.3:P	Bool	FALSE		FALSE
	"M"	%Q0.3	Bool	TRUE		FALSE
	"VP1"	%Q0.4	Bool	FALSE		FALSE
	"VP2"	%Q0.5	Bool	FALSE		FALSE
	"VE"	%Q0.7	Bool	FALSE		FALSE
	"WAITING"	%Q1.0	Bool	FALSE		FALSE

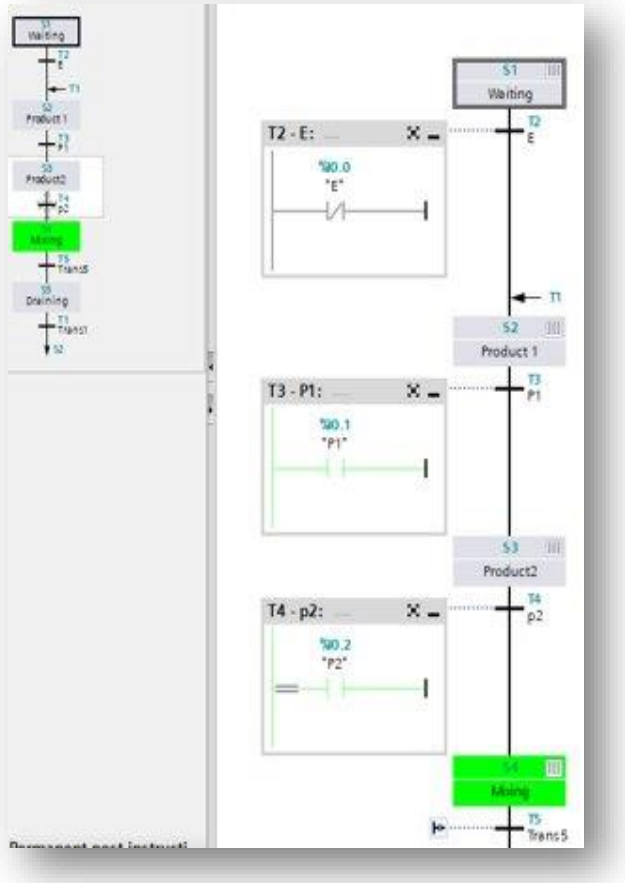


2.4)Go online

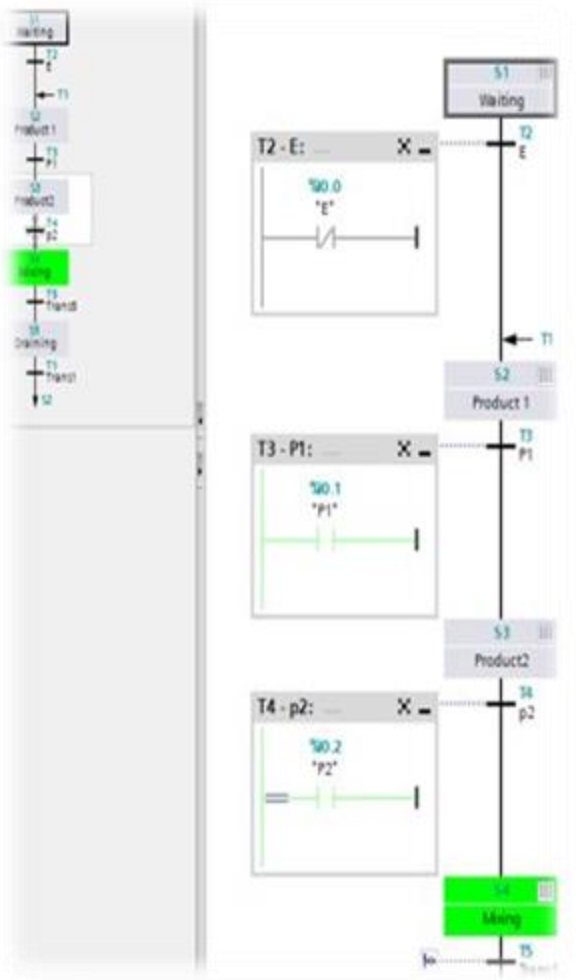
=> Force P2 to true to pass from state adding product 2 to mixing state



	Name	Address	Display format	Monitor/Modify value	Bits	Consistent modify
	"ton_mixing".PT		Time	T#0MS		T#0MS
	"ton_mixing".ET		Time	T#0MS		T#0MS
	"ton_mixing".IN		Bool	FALSE		FALSE
	"ton_mixing".Q		Bool	FALSE		FALSE
	"E":P	%I0.0:P	Bool	TRUE		FALSE
	"P1":P	%I0.1:P	Bool	TRUE		FALSE
	"P2":P	%I0.2:P	Bool	TRUE		FALSE
	"mixing":P	%I0.3:P	Bool	FALSE		FALSE
	"M"	%Q0.3	Bool	TRUE		FALSE
	"VP1"	%Q0.4	Bool	FALSE		FALSE
	"VP2"	%Q0.5	Bool	FALSE		FALSE
	"VE"	%Q0.7	Bool	FALSE		FALSE
	"WAITING"	%Q1.0	Bool	FALSE		FALSE

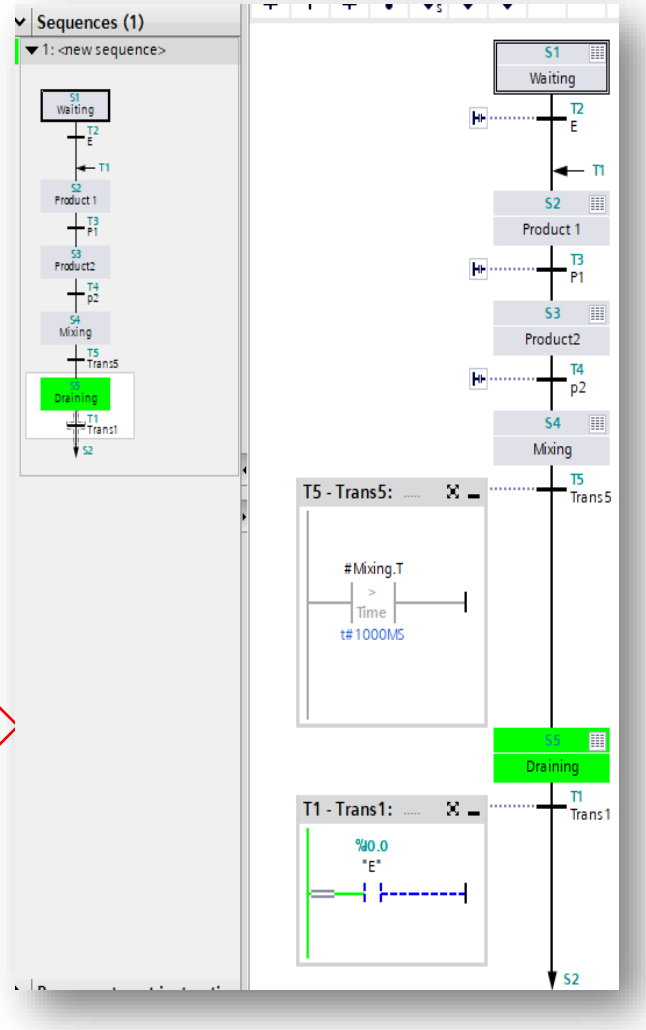


2.4)Go online



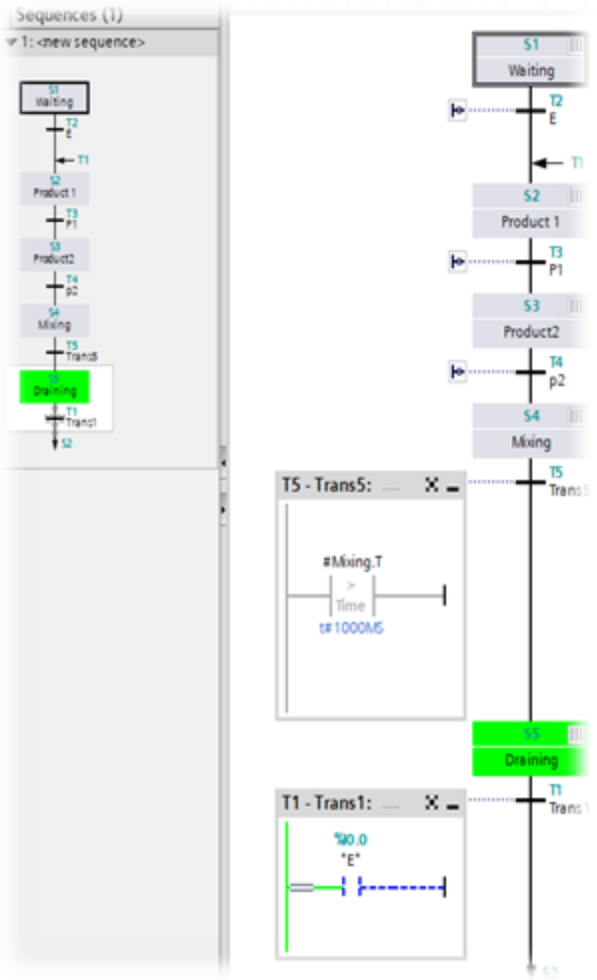
=> Mixing state last 1s then without forcing pass to draining state VE TRUE

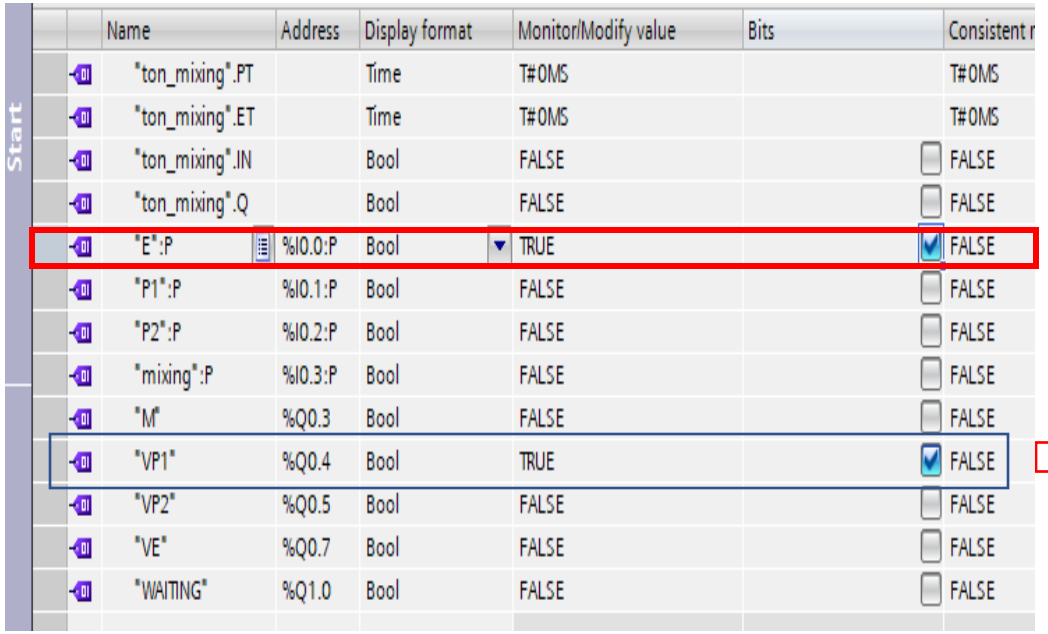
	Name	Address	Display format	Monitor/Modify value	Bits	Consistent modify
	"ton_mixing".PT		Time	T#0MS		T#0MS
	"ton_mixing".ET		Time	T#0MS		T#0MS
	"ton_mixing".IN		Bool	FALSE		FALSE
	"ton_mixing".Q		Bool	FALSE		FALSE
	"E":P	%I0.0:P	Bool	FALSE		FALSE
	"P1":P	%I0.1:P	Bool	FALSE		FALSE
	"P2":P	%I0.2:P	Bool	FALSE		FALSE
	"mixing":P	%I0.3:P	Bool	FALSE		FALSE
	"M"	%Q0.3	Bool	FALSE		FALSE
	"VP1"	%Q0.4	Bool	FALSE		FALSE
	"VP2"	%Q0.5	Bool	FALSE		FALSE
	"VE"	%Q0.7	Bool	TRUE		FALSE
	"WAITING"	%Q1.0	Bool	FALSE		FALSE



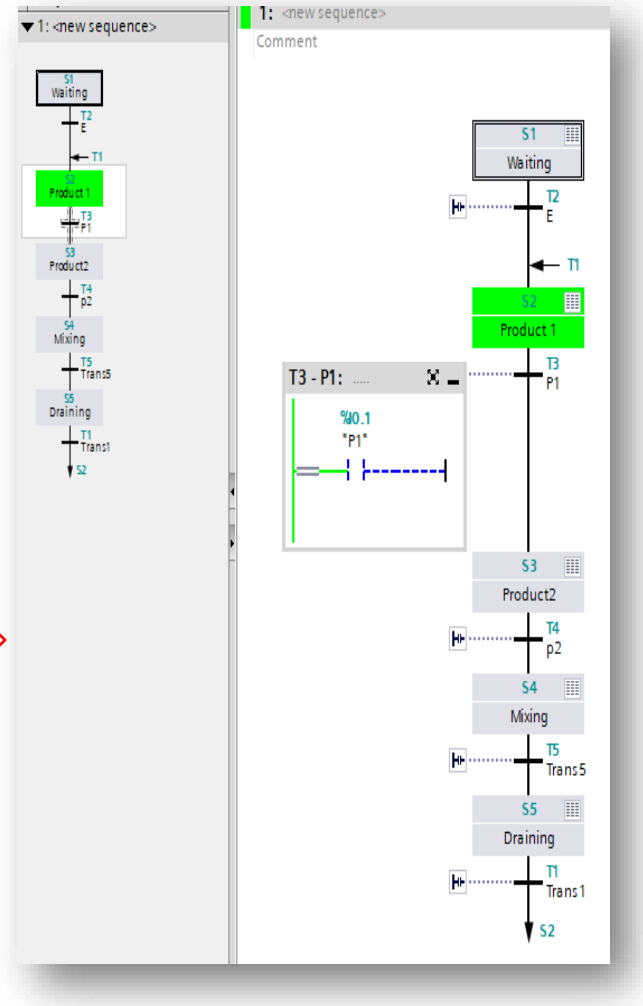
2.4)Go online

=> We force E (empty tank) to true to back into product 1 state





	Name	Address	Display format	Monitor/Modify value	Bits	Consistent r
	"ton_mixing".PT		Time	T#0MS		T#0MS
	"ton_mixing".ET		Time	T#0MS		T#0MS
	"ton_mixing".IN		Bool	FALSE		FALSE
	"ton_mixing".Q		Bool	FALSE		FALSE
	"E":P	%I0.0:P	Bool	TRUE		FALSE
	"P1":P	%I0.1:P	Bool	FALSE		FALSE
	"P2":P	%I0.2:P	Bool	FALSE		FALSE
	"mixing":P	%I0.3:P	Bool	FALSE		FALSE
	"M"	%Q0.3	Bool	FALSE		FALSE
	"VP1"	%Q0.4	Bool	TRUE		FALSE
	"VP2"	%Q0.5	Bool	FALSE		FALSE
	"VE"	%Q0.7	Bool	FALSE		FALSE
	"WAITING"	%Q1.0	Bool	FALSE		FALSE



Global methodology for labs

1. Define in TIA Portal/Unity the exact architecture, corresponding to the actual physical one, all the lots should be configured (it is equivalent to add the required libraries in IT)
2. Configure all the variables (I, O, M…)
3. Configure precisely the network interface (crucial step!), external clock for synchronization
4. Write a programme, the actions, possibly specific memories for Modbus frames
5. You can first work in simulation
    1. Choose the simulation
    2. Compile your programme (HW and SW)
    3. Send your programme to the PLC (here in simulation)
    4. Then run
6. You can then work with the real PLC
    1. Choose the actual PLC
    2. Compile your programme (HW and SW)
    3. Send your programme to the PLC (here on the actual PLC)
    4. Then run, you will need to interact using the simulation card
    5. You can also observe what's going on on the PLC using the debug mode
7. **Never forget to save from time to time what you do !**
8. **Always stop running and the debug mode when you want to send a new programme to the PLC**


Thank you