UGA
Université
Grenoble Alpes

FACTORI
4.0
Erasmus +

Asean-factori

Lab Schneider Implementation (following 2. Schneider Lab Implementation)

Jean-Marc THIRIET

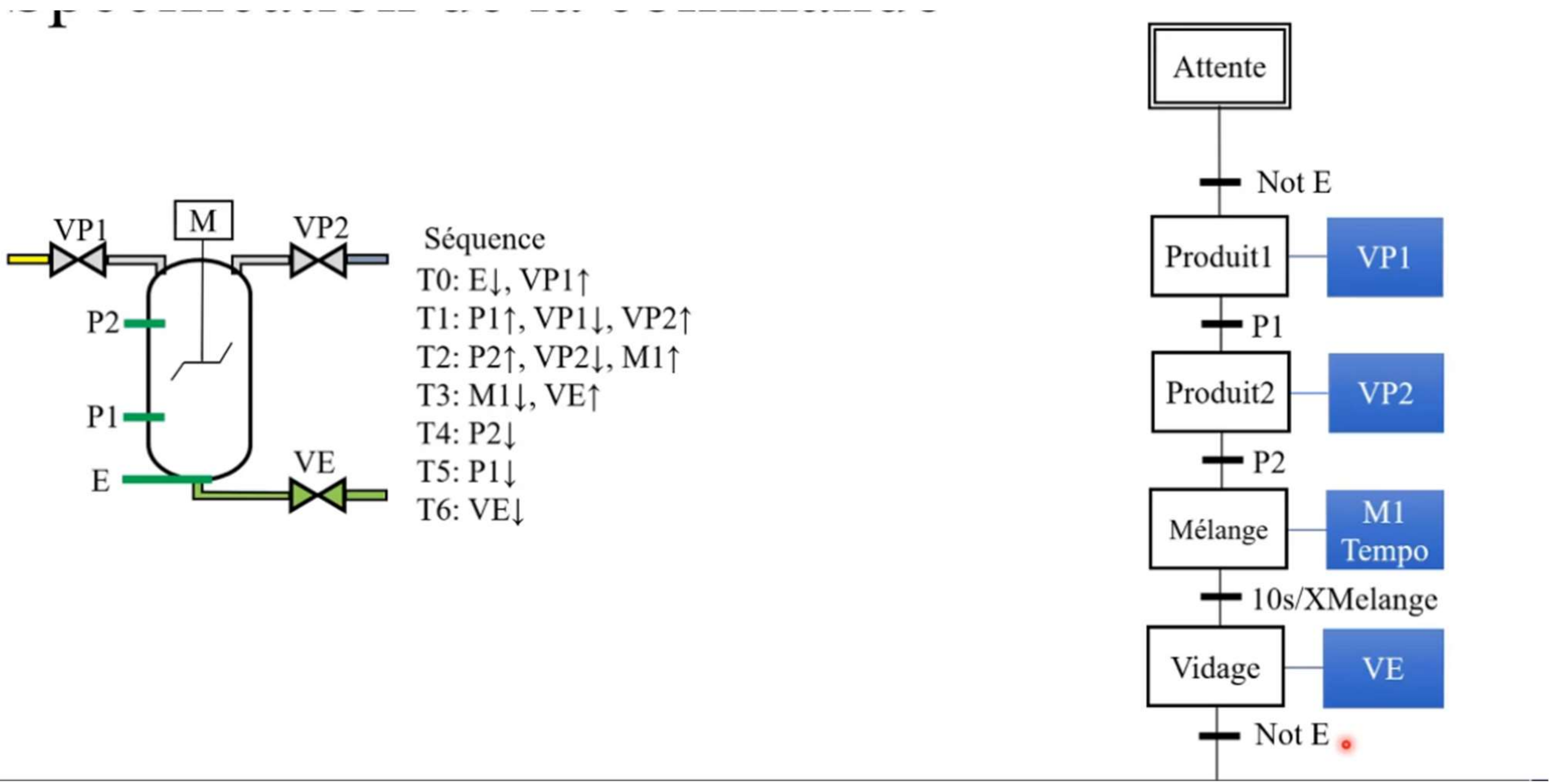
jean-marc.thiriet@univ-grenoble-alpes.fr

http://www.gipsa-lab.grenoble-inp.fr/~jean-marc.thiriet/asean/asean.html

http://www.gipsa-lab.grenoble-inp.fr/~jean-marc
.thiriet/asean/asean.html

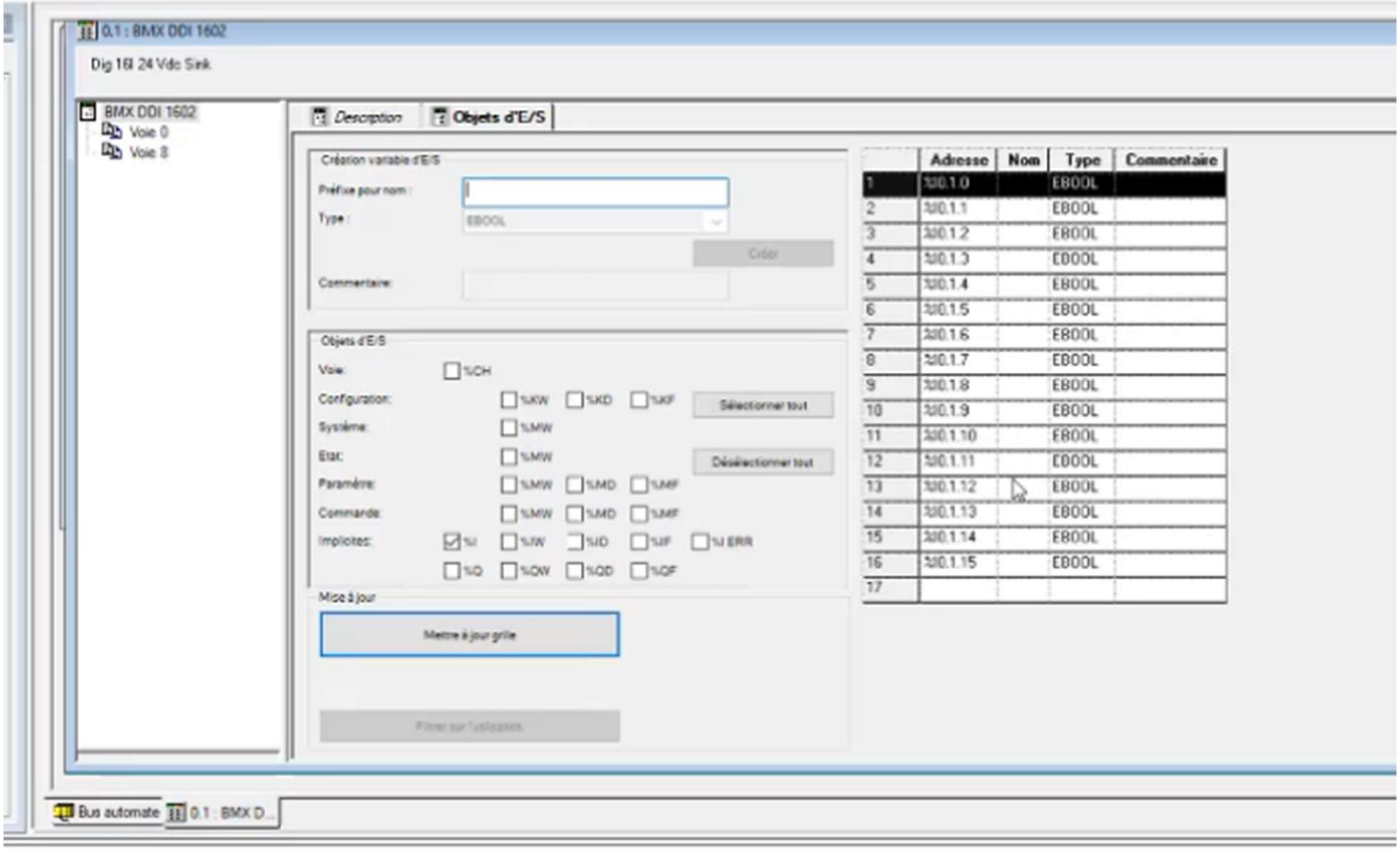
Grafcet

- 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↓

Inputs and Outputs



0.1 : BMX DDI 1602

Dig 16I 24 Vdc Sink

Création variable d'E/S

Préfixe pour nom :

Type : EBOOL

Commentaire:

Objets d'E/S

Voie: □ %CH

Configuration: □ %KW □ %KD □ %KF

Système: □ %MW

Etat: □ %MW

Paramètre: □ %MW □ %MD □ %MF

Commande: □ %MW □ %MD □ %MF

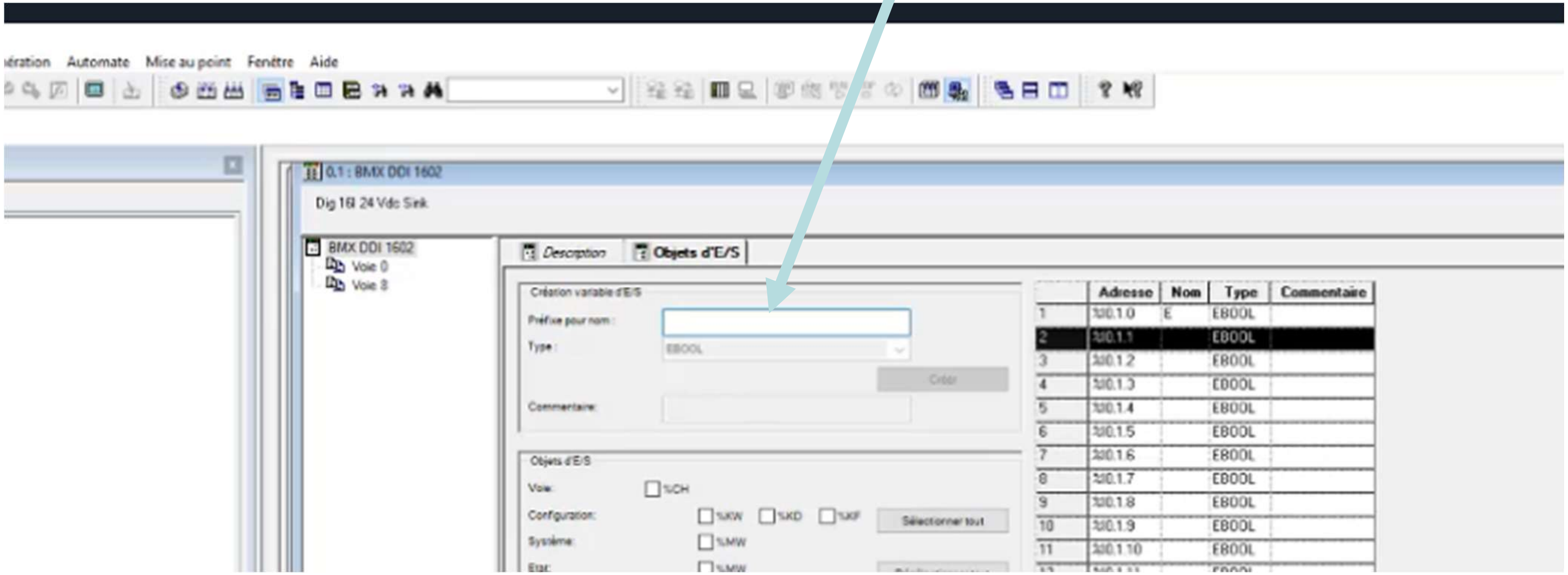
Implicites: ☑ %I □ %IW □ %ID □ %IF □ %I ERR

□ %Q □ %QW □ %QD □ %QF

Mise à jour

	Adresse	Nom	Type	Commentaire
1	%I0.1.0		EBOOL	
2	%I0.1.1		EBOOL	
3	%I0.1.2		EBOOL	
4	%I0.1.3		EBOOL	
5	%I0.1.4		EBOOL	
6	%I0.1.5		EBOOL	
7	%I0.1.6		EBOOL	
8	%I0.1.7		EBOOL	
9	%I0.1.8		EBOOL	
10	%I0.1.9		EBOOL	
11	%I0.1.10		EBOOL	
12	%I0.1.11		EBOOL	
13	%I0.1.12		EBOOL	
14	%I0.1.13		EBOOL	
15	%I0.1.14		EBOOL	
16	%I0.1.15		EBOOL	
17				

Possibility to create directly a variable



0.1 : BMX DDI 1602

Dig 16 24 Vdc Sink

- BMX DDI 1602
  - Voie 0
  - Voie 8

Description | Objets d'E/S

Création variable d'E/S

Préfixe pour nom :

Type : EBOOL

Créer

Commentaire:

Objets d'E/S

Voie: □ %CH

Configuration: □ %KW □ %KD □ %KF

Sélectionner tout

Système: □ %MW

Etat: □ %MW

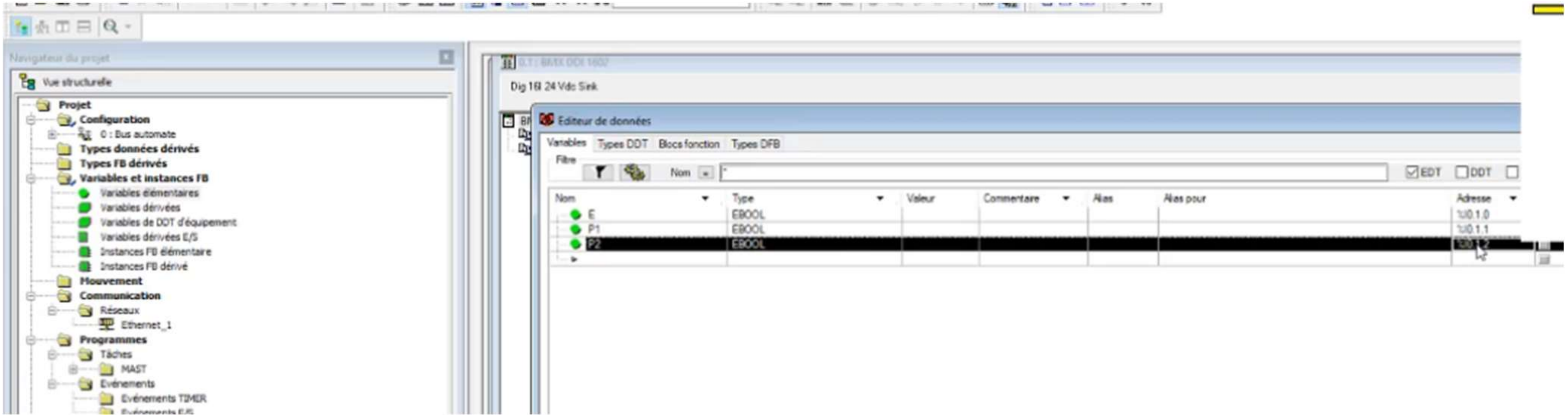
	Adresse	Nom	Type	Commentaire
1	%I0.1.0	E	EBOOL	
2	%I0.1.1		EBOOL	
3	%I0.1.2		EBOOL	
4	%I0.1.3		EBOOL	
5	%I0.1.4		EBOOL	
6	%I0.1.5		EBOOL	
7	%I0.1.6		EBOOL	
8	%I0.1.7		EBOOL	
9	%I0.1.8		EBOOL	
10	%I0.1.9		EBOOL	
11	%I0.1.10		EBOOL	
12	%I0.1.11		EBOOL	

Our variables

- ! EBOOL better than BOOL!
  - – Contains the present value
  - – The previous value
  - – One information indicating if the bit has been forced or not

Possibility to add the variables in « Elementary variables »

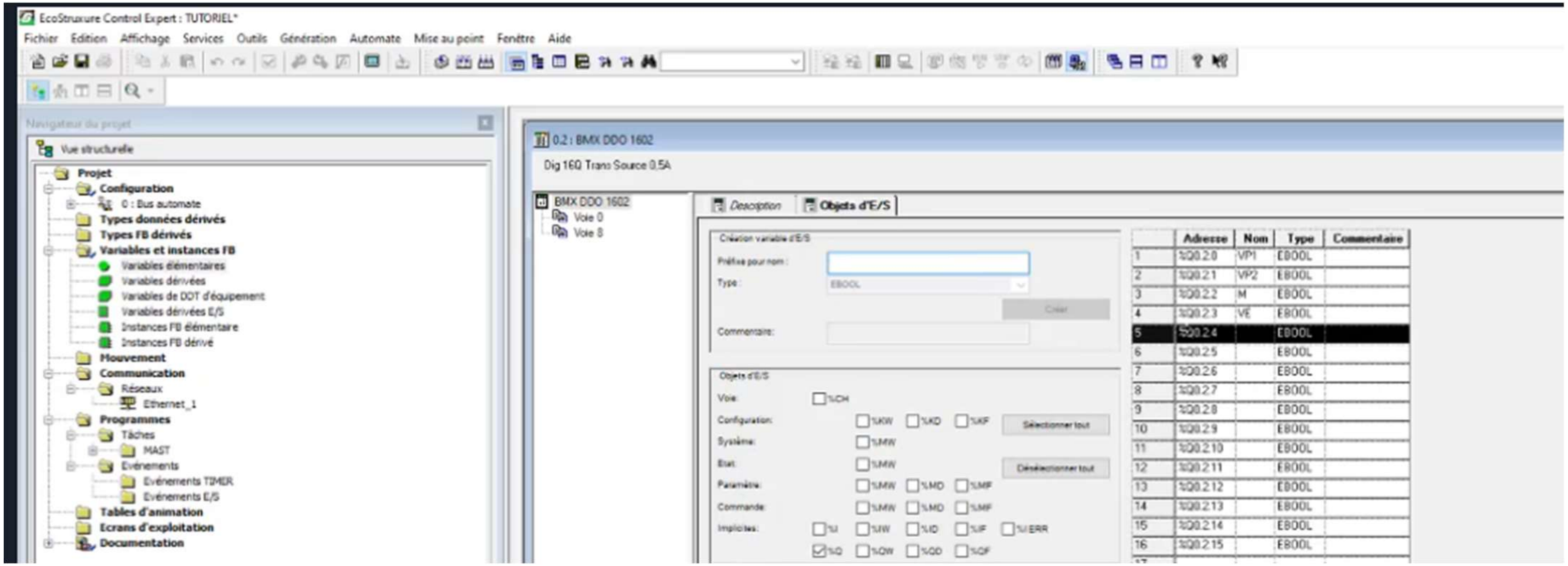
- 3 Inputs



Nom: *

Nom	Type	Valeur	Commentaire	Alias	Alias pour	Adresse
E	EBOOL					%I0.1.0
P1	EBOOL					%I0.1.1
P2	EBOOL					%I0.1.2

Definitions of the 4 outputs (same manners)



0.2 : BMX DDO 1602

Dig 16Q Trans Source 0.5A

Création variable d'E/S

Préfixe pour nom :

Type : EBOOL

Commentaire:

Objets d'E/S

Voie: □ %CH

Configuration: □ %KW □ %KD □ %KF

Système: □ %MW

Etat: □ %MW

Paramètre: □ %MW □ %MD □ %MF

Commande: □ %MW □ %MD □ %MF

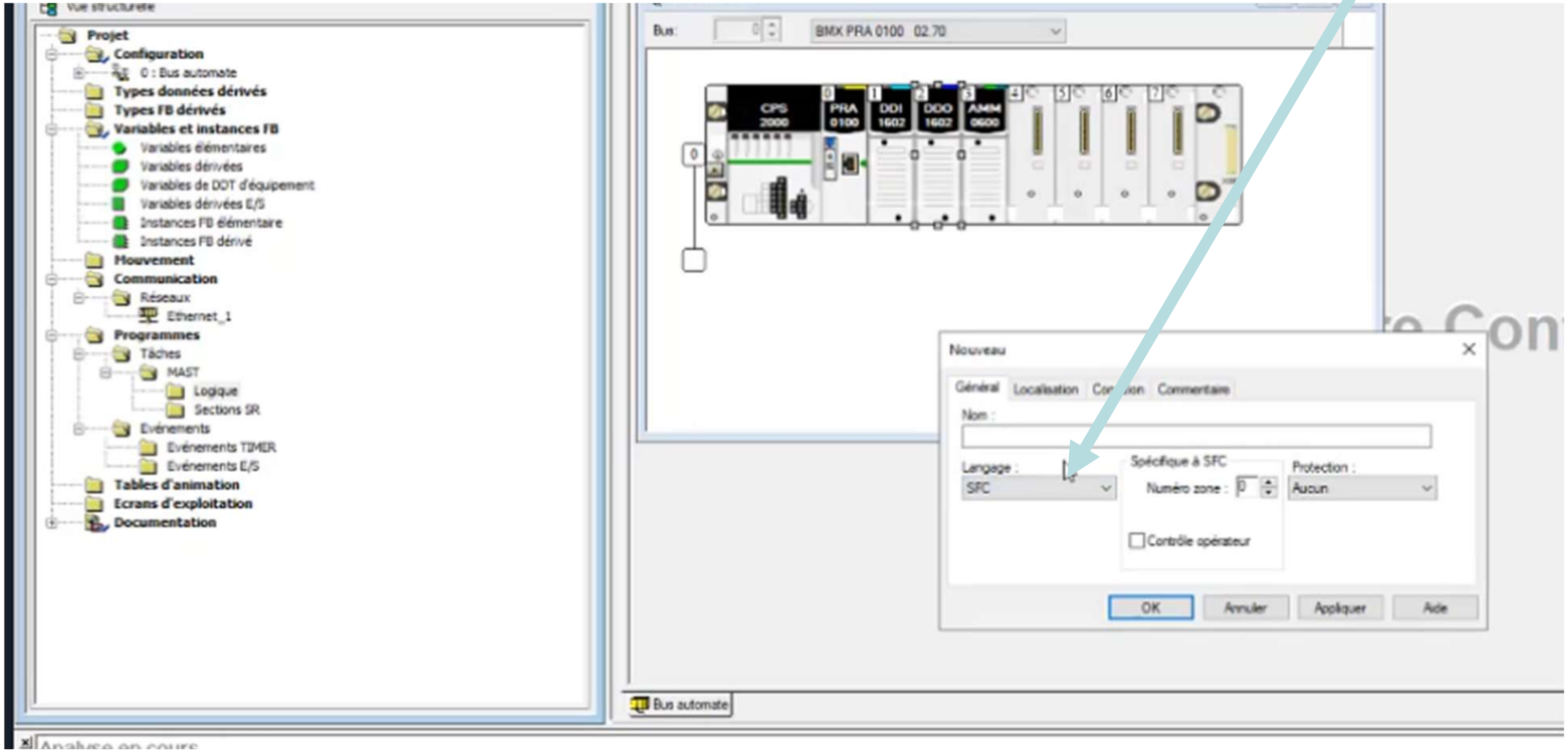
Implicites: □ %I □ %IW □ %ID □ %IF □ %I ERR

☑ %Q □ %QW □ %QD □ %QF

	Adresse	Nom	Type	Commentaire
1	%Q0.2.0	VP1	EBOOL	
2	%Q0.2.1	VP2	EBOOL	
3	%Q0.2.2	M	EBOOL	
4	%Q0.2.3	VE	EBOOL	
5	%Q0.2.4		EBOOL	
6	%Q0.2.5		EBOOL	
7	%Q0.2.6		EBOOL	
8	%Q0.2.7		EBOOL	
9	%Q0.2.8		EBOOL	
10	%Q0.2.9		EBOOL	
11	%Q0.2.10		EBOOL	
12	%Q0.2.11		EBOOL	
13	%Q0.2.12		EBOOL	
14	%Q0.2.13		EBOOL	
15	%Q0.2.14		EBOOL	
16	%Q0.2.15		EBOOL	

Implementation of the Grafcet programmes

- 2 sections
  - One section of the finite state machine
  - One section for the actions
- Click on Programs then Task then MAST then « new section » then SFC
- Then give a name (for example « main »)



Nouveau

Nom :

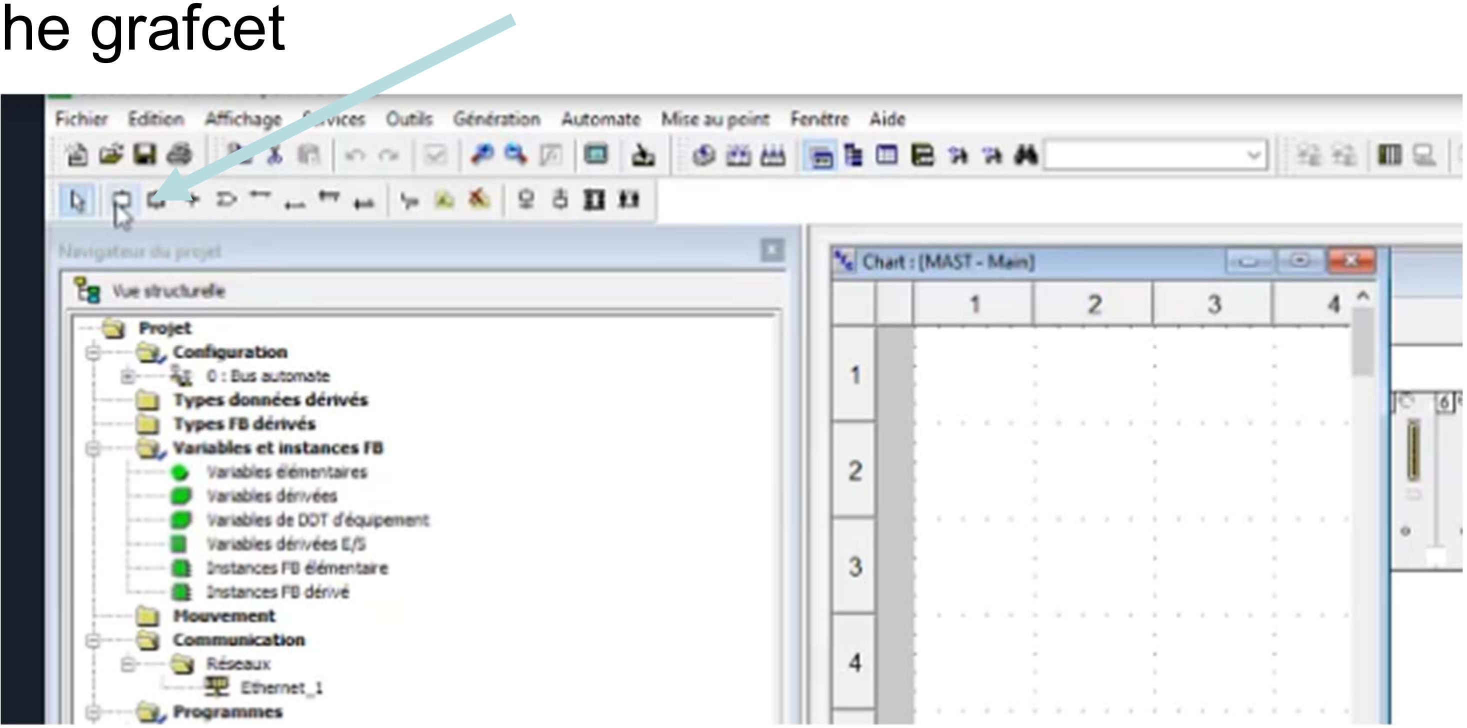
Langage : SFC

Spécifique à SFC — Numéro zone : 0

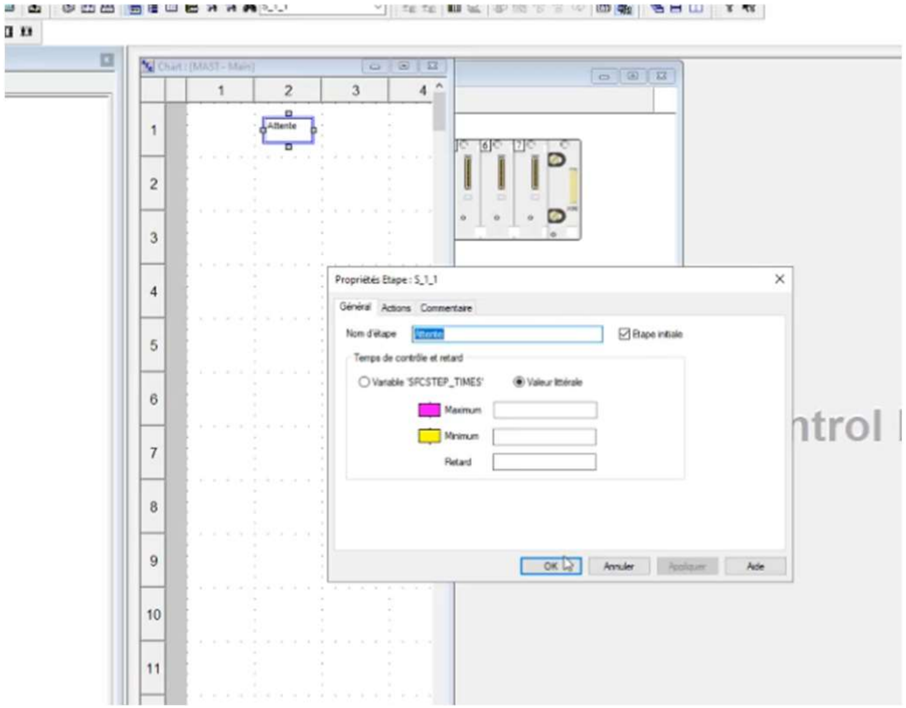
Protection : Aucun

□ Contrôle opérateur

- Then open the « chart »
- We will draw the behavioural part of the Graftcet
- There is an editing area for the steps and transitions of the grafcet



- Creation of the first state « Attente » then mark « initial state »



Propriétés Etape : S_1_1

Général | Actions | Commentaire

Nom d'étape: Attente ☑ Etape initiale

Temps de contrôle et retard

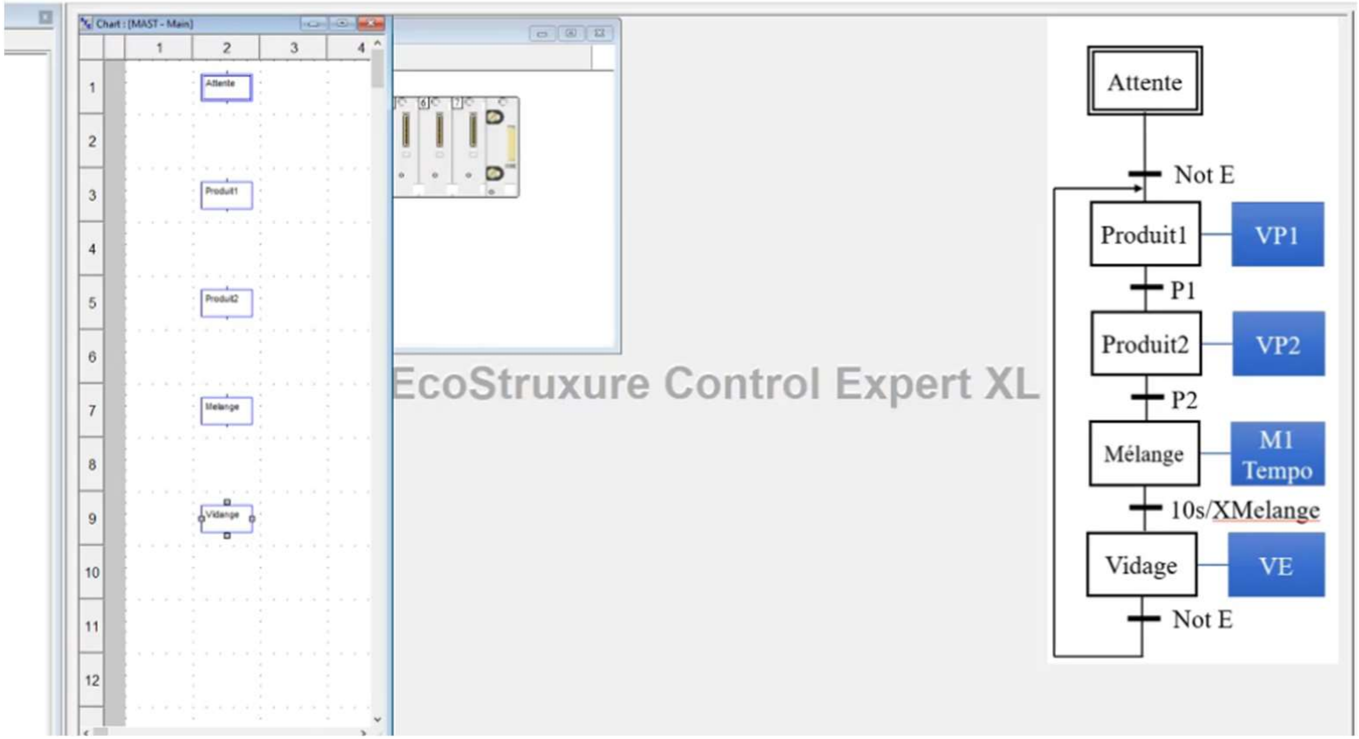
○ Variable 'SFCSTEP_TIMES' ● Valeur littérale

Maximum

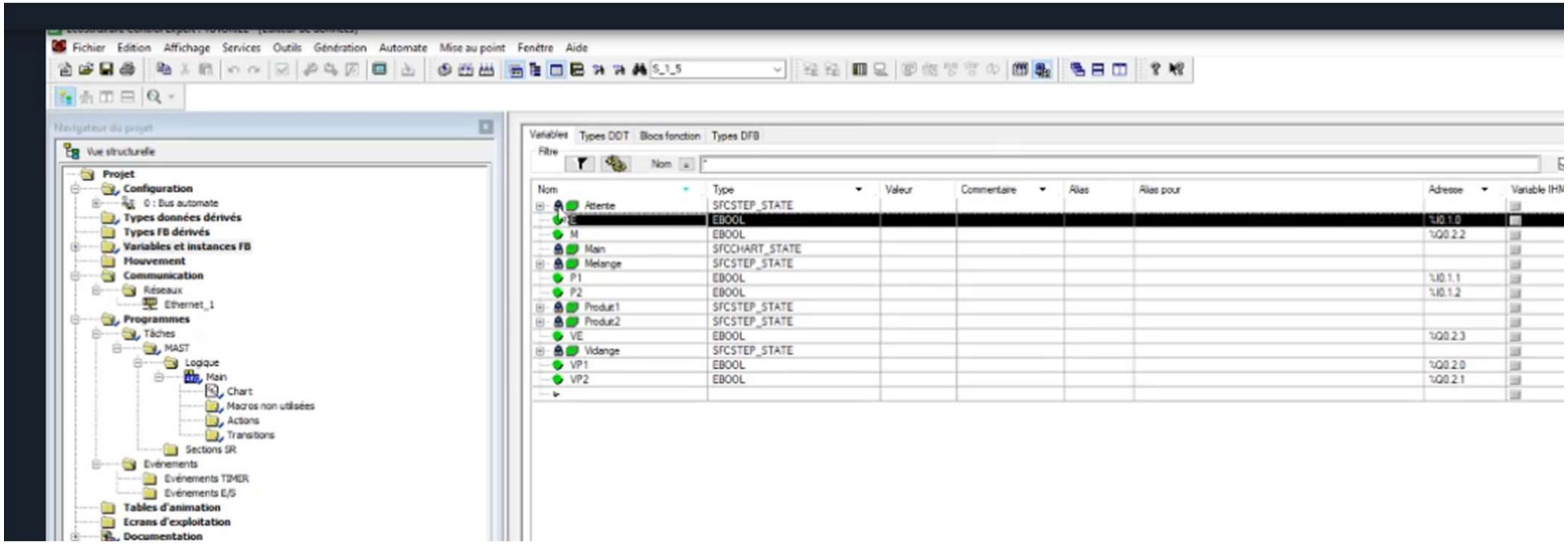
Minimum

Retard

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

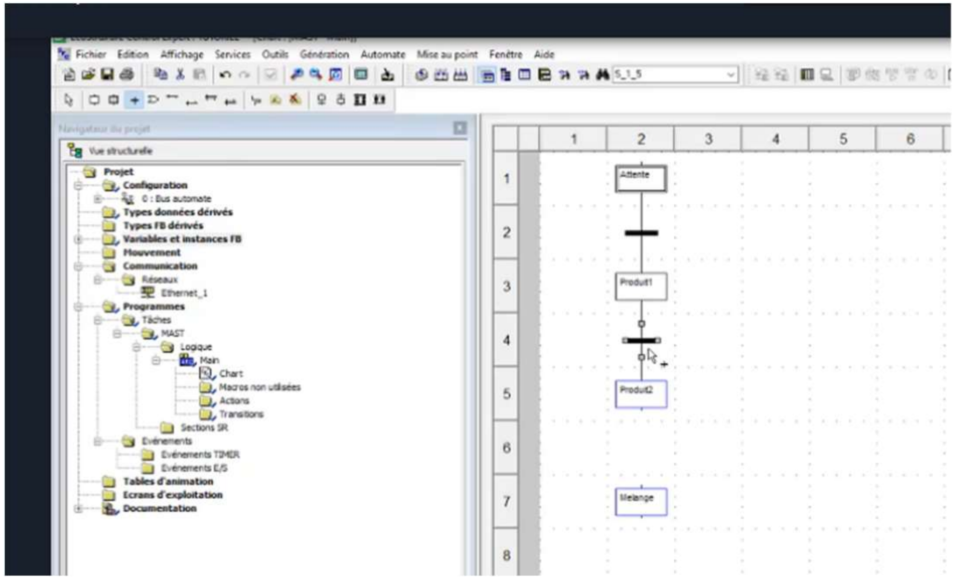


- Following the creation of the states, some variables (type « Structure ») have been created, corresponding to the states
- X corresponds to the fact that the state is activated or not
- T indicates the time spent within the state
- Tmin and Tmax are two two limiting variables of the residence time in the state

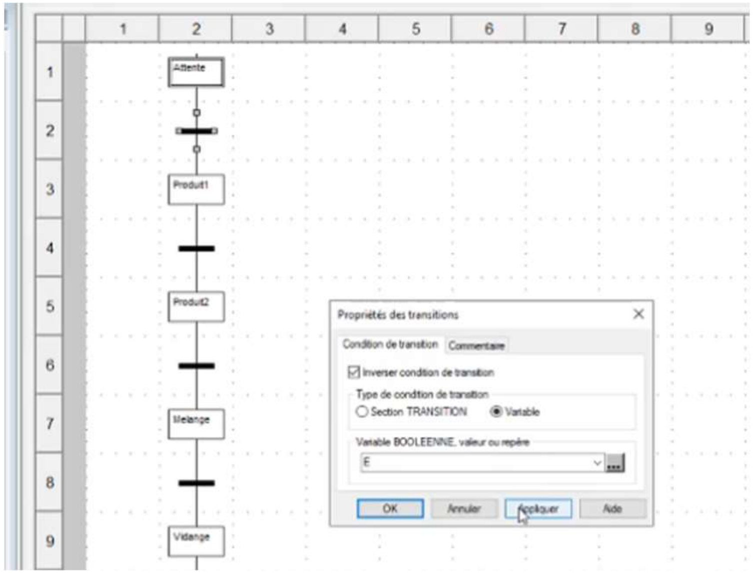


Nom	Type	Valeur	Commentaire	Alias	Alias pour	Adresse	Variable IHM
Attente	SFCSTEP_STATE						
E	EBOOL					%I0.1.0	
M	EBOOL					%Q0.2.2	
Main	SFCCHART_STATE						
Melange	SFCSTEP_STATE						
P1	EBOOL					%I0.1.1	
P2	EBOOL					%I0.1.2	
Produit1	SFCSTEP_STATE						
Produit2	SFCSTEP_STATE						
VE	EBOOL					%Q0.2.3	
Vidange	SFCSTEP_STATE						
VP1	EBOOL					%Q0.2.0	
VP2	EBOOL					%Q0.2.1	

- Implementation of the transitions by clicking btween two states



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



Propriétés des transitions

Condition de transition | Commentaire

☑ Inverser condition de transition

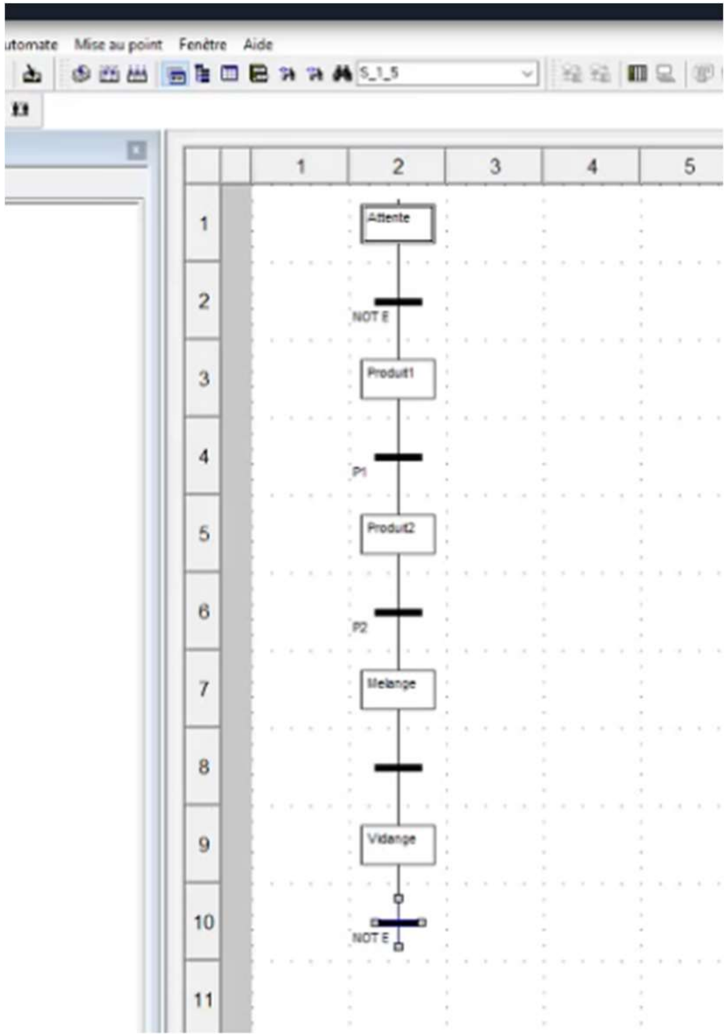
Type de condition de transition

○ Section TRANSITION ● Variable

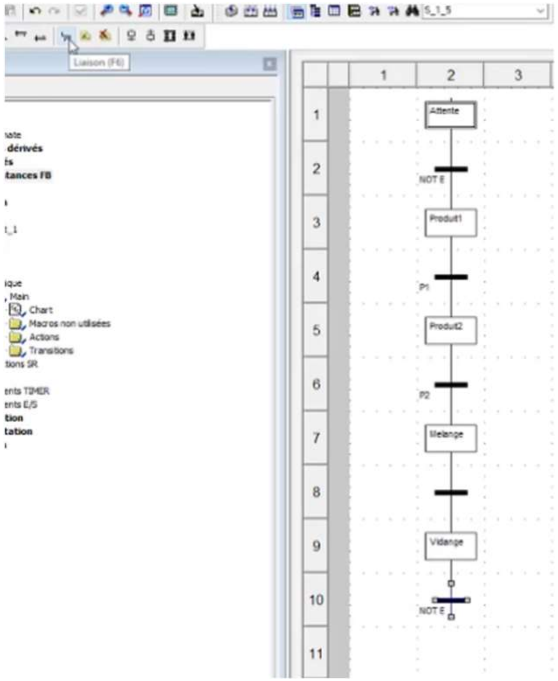
Variable BOOLEENNE, valeur ou repère

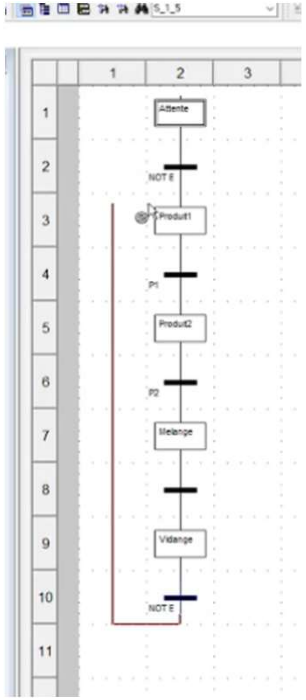
E

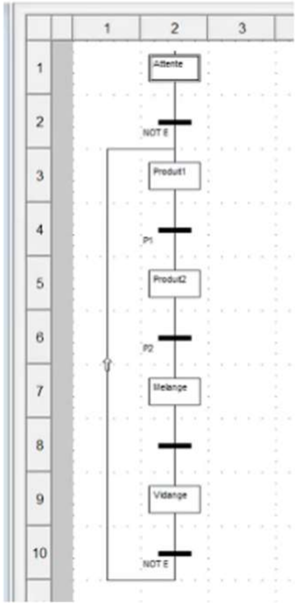
- We can create all the transitions, except the one after Melange for which we need to create a timeout (it will be done later)



- We close the Grafcet by using the tool « Link » (Liaison)

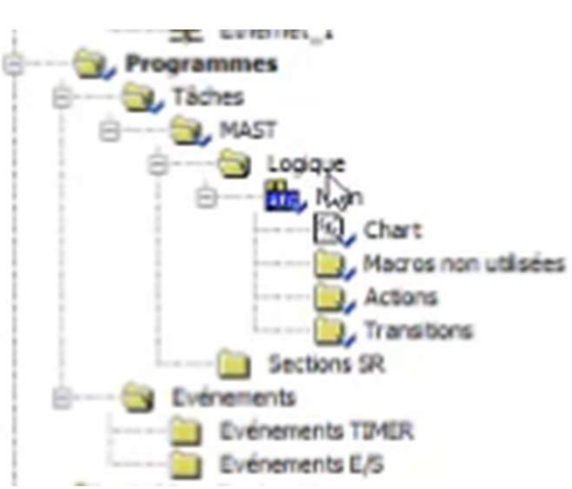


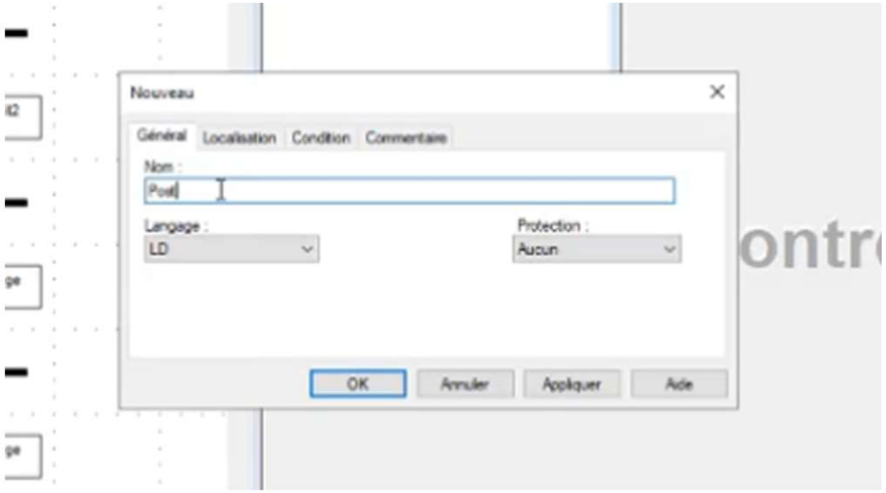




Implementation of the actions in ladder

- Definition of a new section for the programme in ladder
- This section is in Ladder (LD) and is called « Post »
- « Post » is a traditional name, which means post-processing, bacause the actuators are activated after the programme
- Another important aspect is that the sections are run in the alphabetic order ; so « Post » is after « Main » (Grafcet programme)



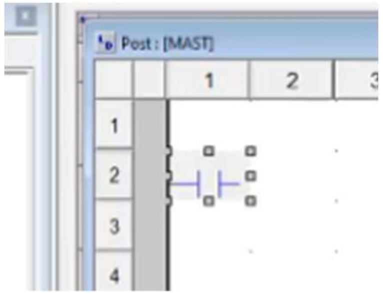


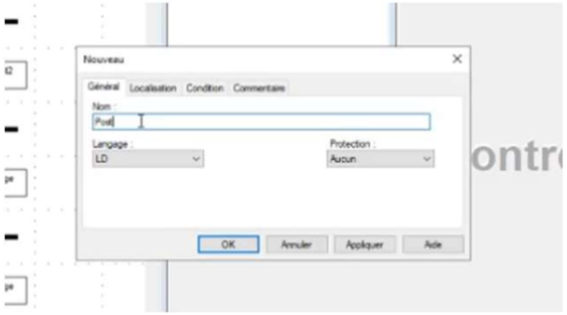
Nom : Post

Langage : LD

Protection : Aucun

- 1. For state Produit1, there is one action => VP1
- We add a ladder contact
- The condition (click on the contact) will be Produit1.X (we will find it in the structure)

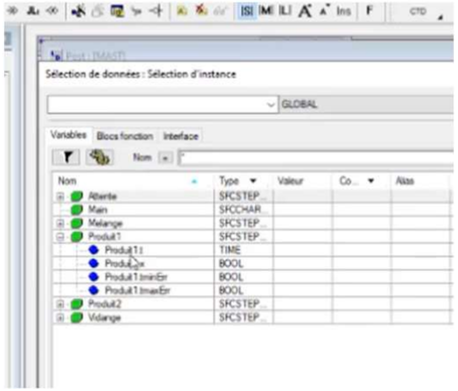



Nouveau

Nom : Post

Langage : LD

Protection : Aucun



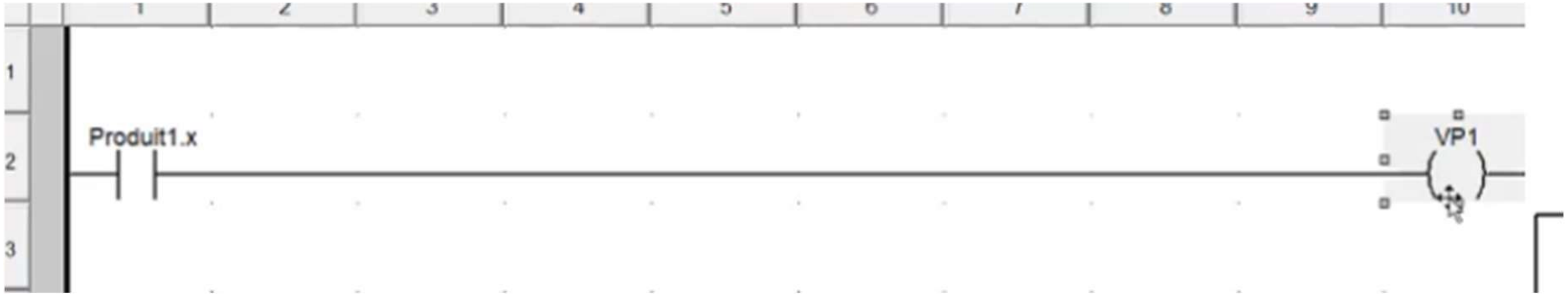
Sélection de données : Sélection d'instance

GLOBAL

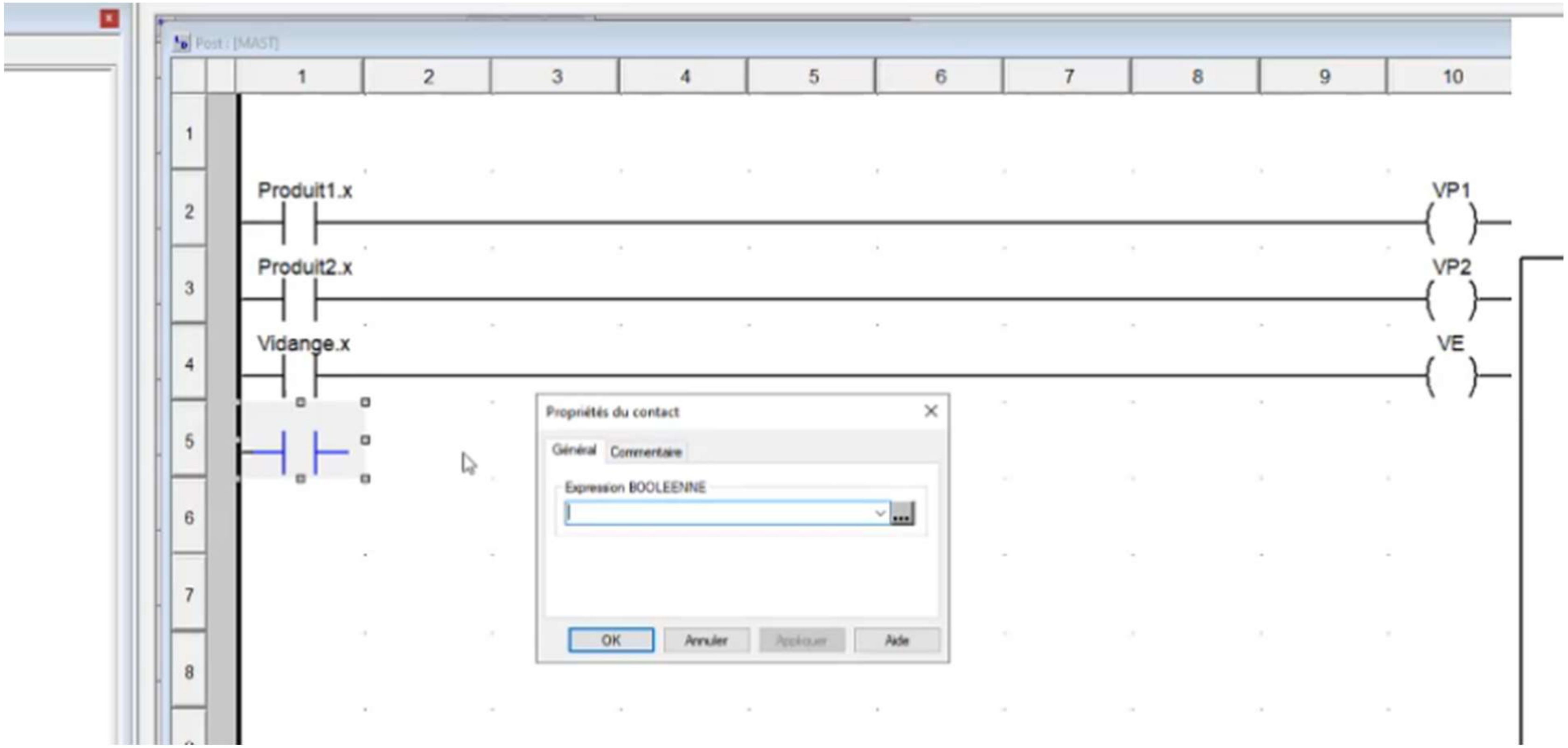
Nom = *

Nom	Type	Valeur	Co...	Alias
Attente	SFCSTEP...			
Main	SFCCHAR...			
Melange	SFCSTEP...			
Produit1	SFCSTEP...			
Produit1.t	TIME			
Produ...x	BOOL			
Produit1.tminErr	BOOL			
Produit1.tmaxErr	BOOL			
Produit2	SFCSTEP...			
Vidange	SFCSTEP...			

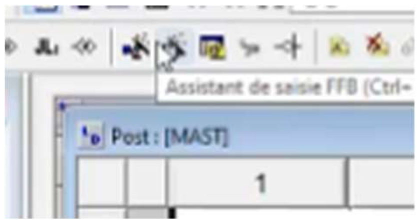
- The output is VP1
- This ladder means that VP1 is On when the state Produit1 os On (actually when the variable Produit1.X is On)
- The Coil VP1 here is activated where Produit1.X is On, and then deactivated (monostable)

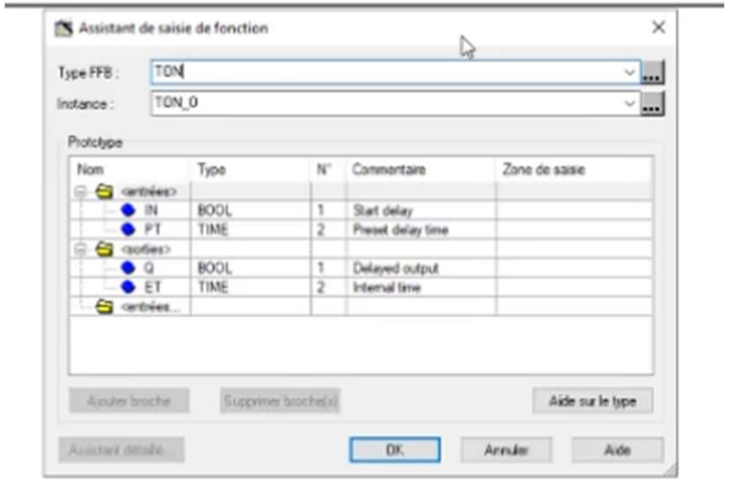


- We can then create Product2.X => VP2 and vidange.X => VE



- Each actuator activation should appear once only.
- For « Melange », we need to activate both the engne (M1) and the timeout.
- Timeout are function blocks
- We will use the tool « Assistant for function edition »
- Timeout are objects with the type TON




Assistant de saisie de fonction

Type FFB : TON

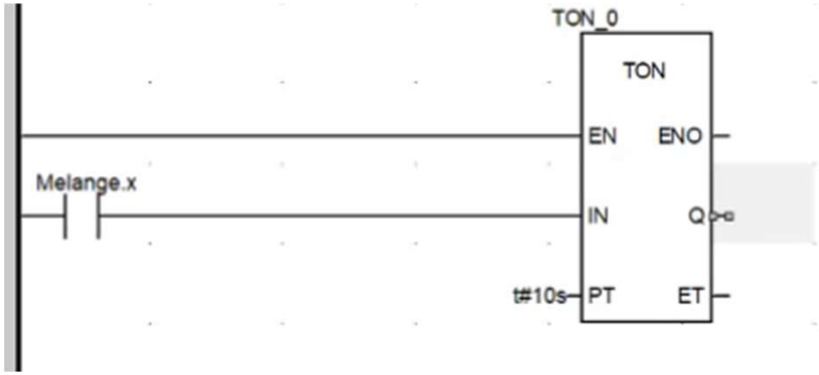
Instance : TON_0

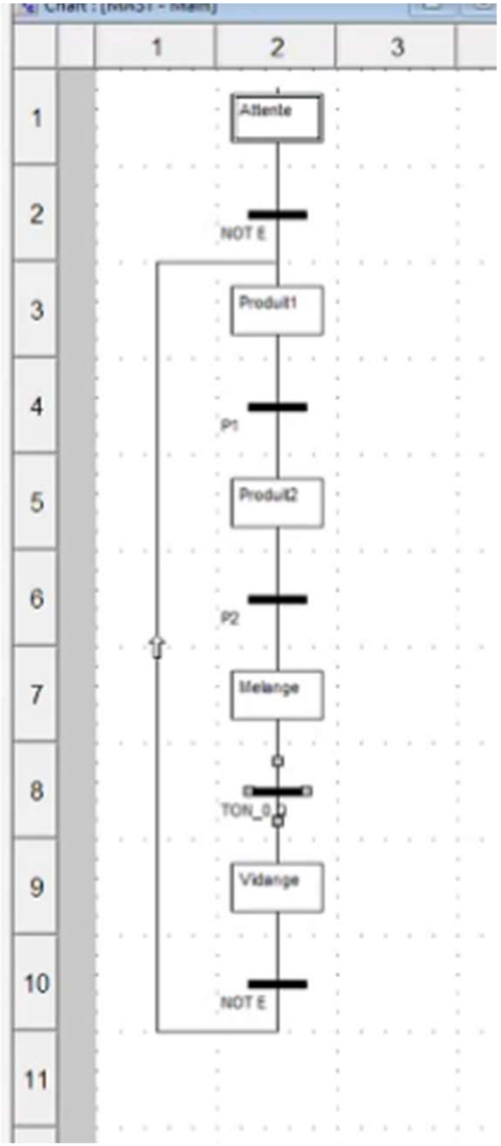
Prototype

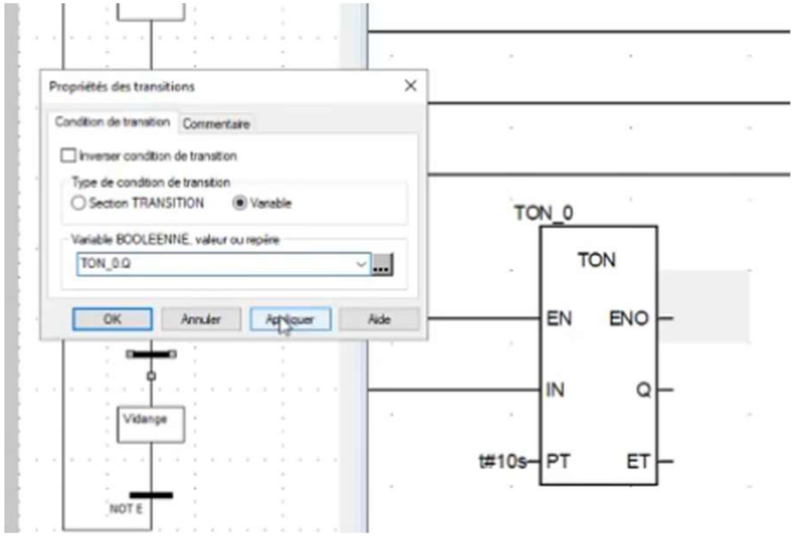
Nom	Type	N°	Commentaire	Zone de saisie
<entrées>				
IN	BOOL	1	Start delay	
PT	TIME	2	Preset delay time	
<sorties>				
Q	BOOL	1	Delayed output	
ET	TIME	2	Internal time	
<entrées...				

Function blocks

- EN means Enable (should be used for the block to be executed)
- ENO will be activated at the end of the excution of the function code. Very useful if we need some sequences (chain) of functions (for instance ADD before a multiplication…)
- IN is the input of the function : for a timeout, it is here we should put the signal for the beginning of the timeout
- PT (Preset Time), here 10s (t# means this is a variable with the type « Time »)
- Q is the input, which is acative at the end of the timeout. It will be used for our Grafcet







Propriétés des transitions

Condition de transition | Commentaire

☐ Inverser condition de transition

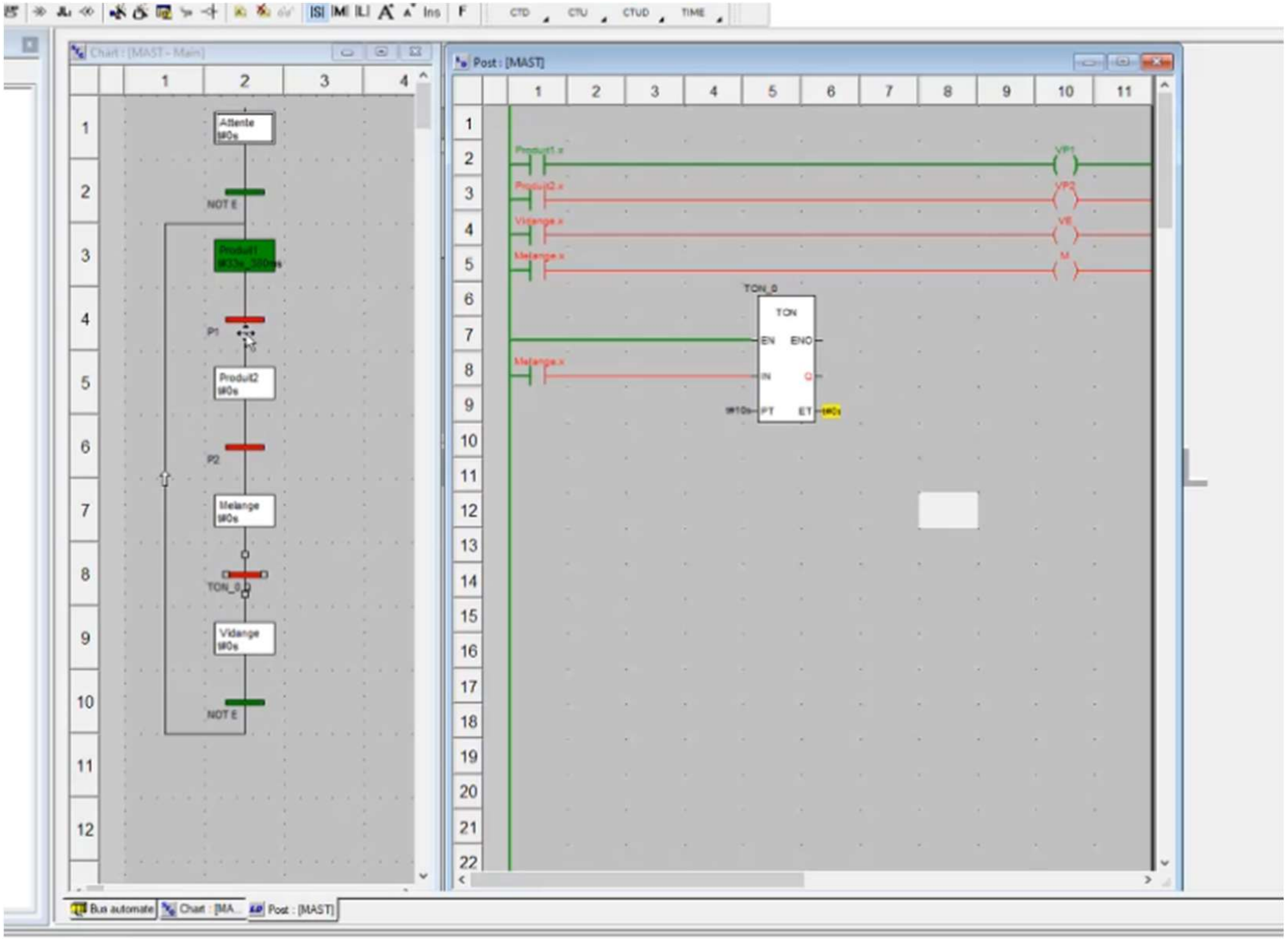
Type de condition de transition
○ Section TRANSITION ● Variable

Variable BOOLEENNE, valeur ou repère
TON_0.Q

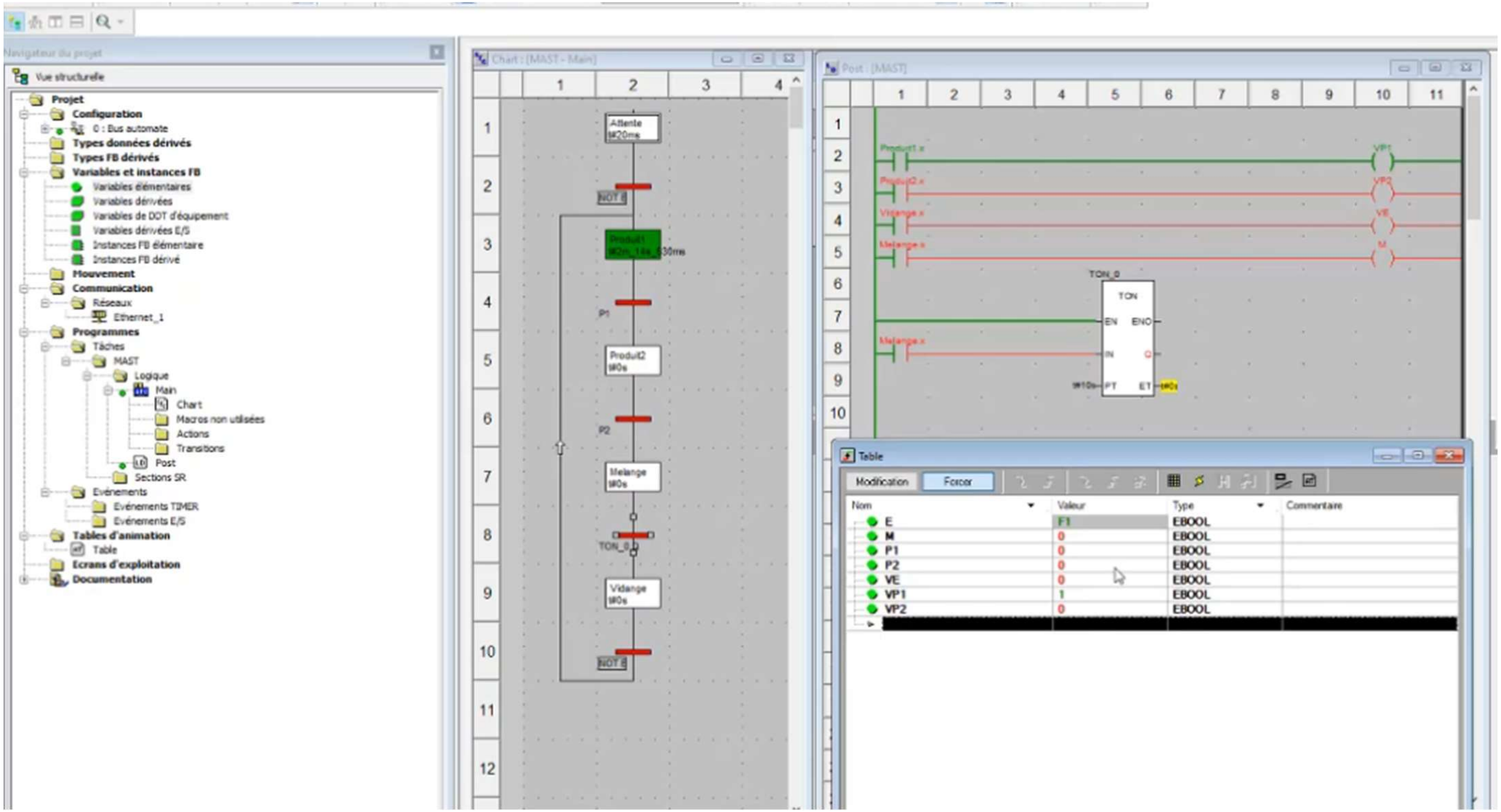
Global methodology for labs

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**

A running programme

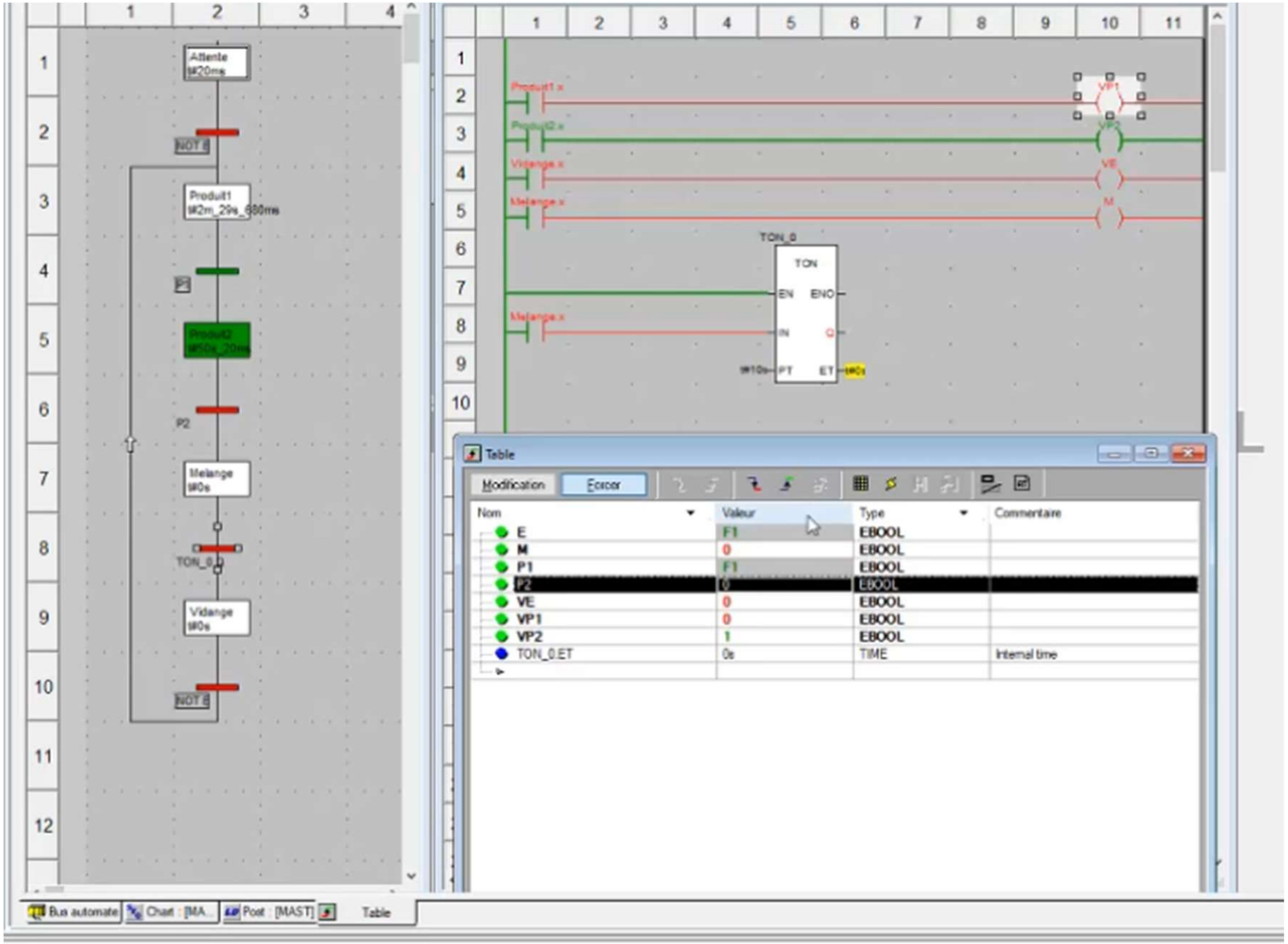


Using of the animation table (in simulation) to force variables



Nom	Valeur	Type	Commentaire
E	F1	EBOOL	
M	0	EBOOL	
P1	0	EBOOL	
P2	0	EBOOL	
VE	0	EBOOL	
VP1	1	EBOOL	
VP2	0	EBOOL	

Timeout can also be visualised



Nom	Valeur	Type	Commentaire
E	F1	EBOOL	
M	0	EBOOL	
P1	F1	EBOOL	
P2	0	EBOOL	
VE	0	EBOOL	
VP1	0	EBOOL	
VP2	1	EBOOL	
TON_0.ET	0s	TIME	Internal time

Thank you !