



PLC Programmation In Phnom Penh-ITC

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



Asean-Factori 4.0 project

Phnom Penh, 15-26 August 2022

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

Grenoble: Departure of the platform



Thank to everybody for a very active implication!

Phnom Penh: Arrival of the platform



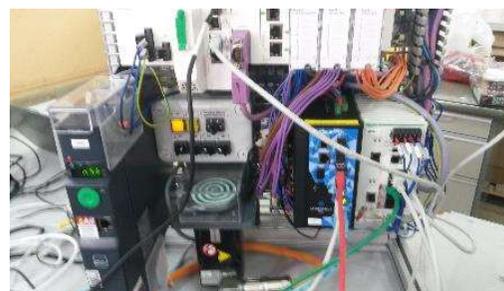
Thank to everybody for a very active implication!

The platform is working!



Of course IO are not configured yet!

The GICS simulation card is working, despite the left « fixation » has been broken during the travel!



```
envoi d'une requête "Ping" 192.168.0.50 avec 32 octets de données :  
réponse de 192.168.0.50 : octets=32 temps=2 ms TTL=255  
réponse de 192.168.0.50 : octets=32 temps=3 ms TTL=255  
réponse de 192.168.0.50 : octets=32 temps=4 ms TTL=255  
réponse de 192.168.0.50 : octets=32 temps=2 ms TTL=255
```

Now we have to begin to play with it!

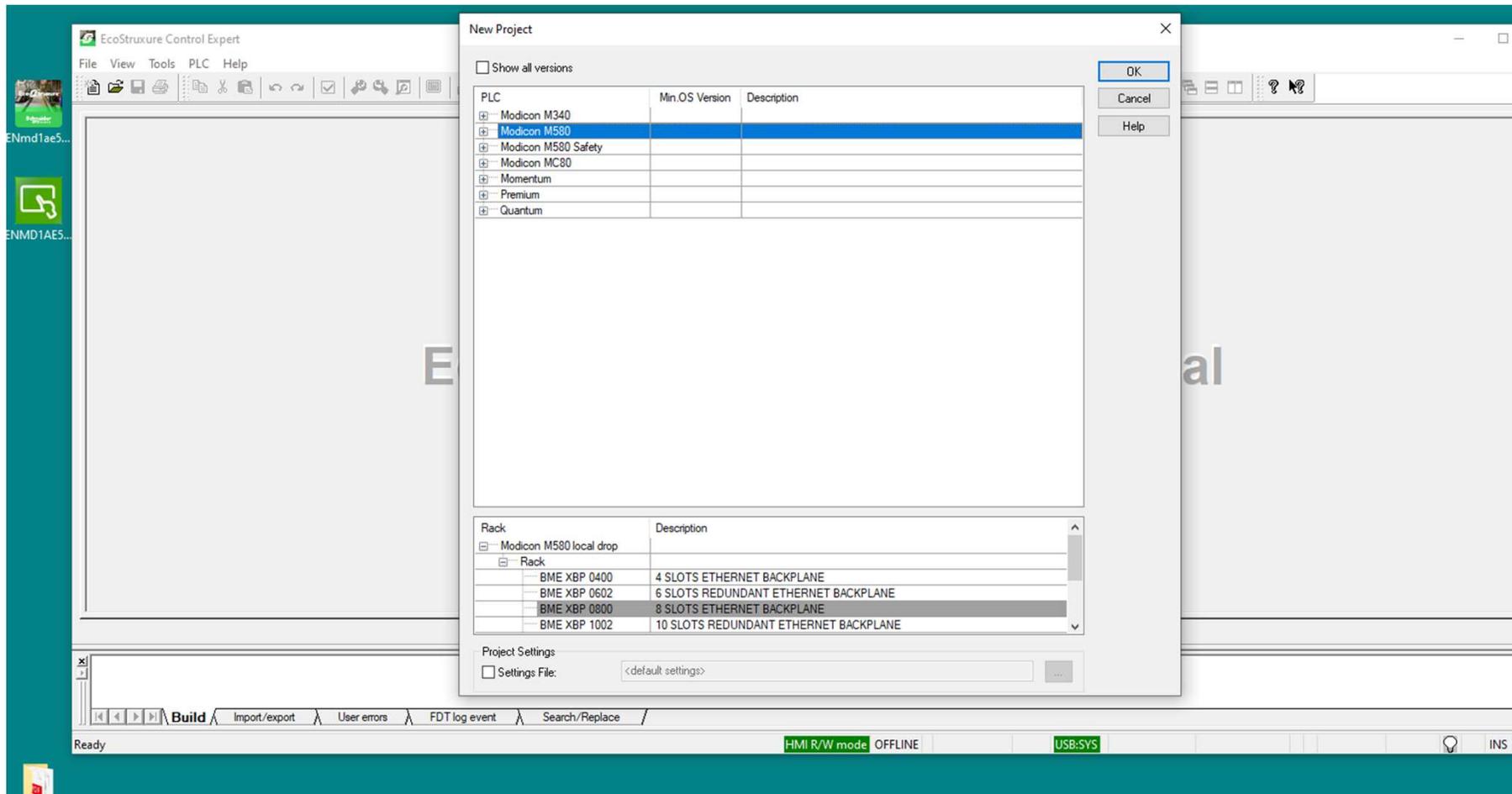
Various possible uses of the platform

- Platform
 - PLC programming (sequential) => SFC (Grafcet), Ladder => GICS card
 - Control the engine available on the platform
 - Develop a small application (lift)
 - Network applications => communication, wireshark
 - Cyber-security aspects
- Project expectations
 - Discuss: dissemination, next meetings
- Training expectations
 - 1. PLC Programming (VERY IMPORTANT)
 - 2. Networking aspects (VERY IMPORTANT)
 - 3. Cyber-security (OPTIONAL)

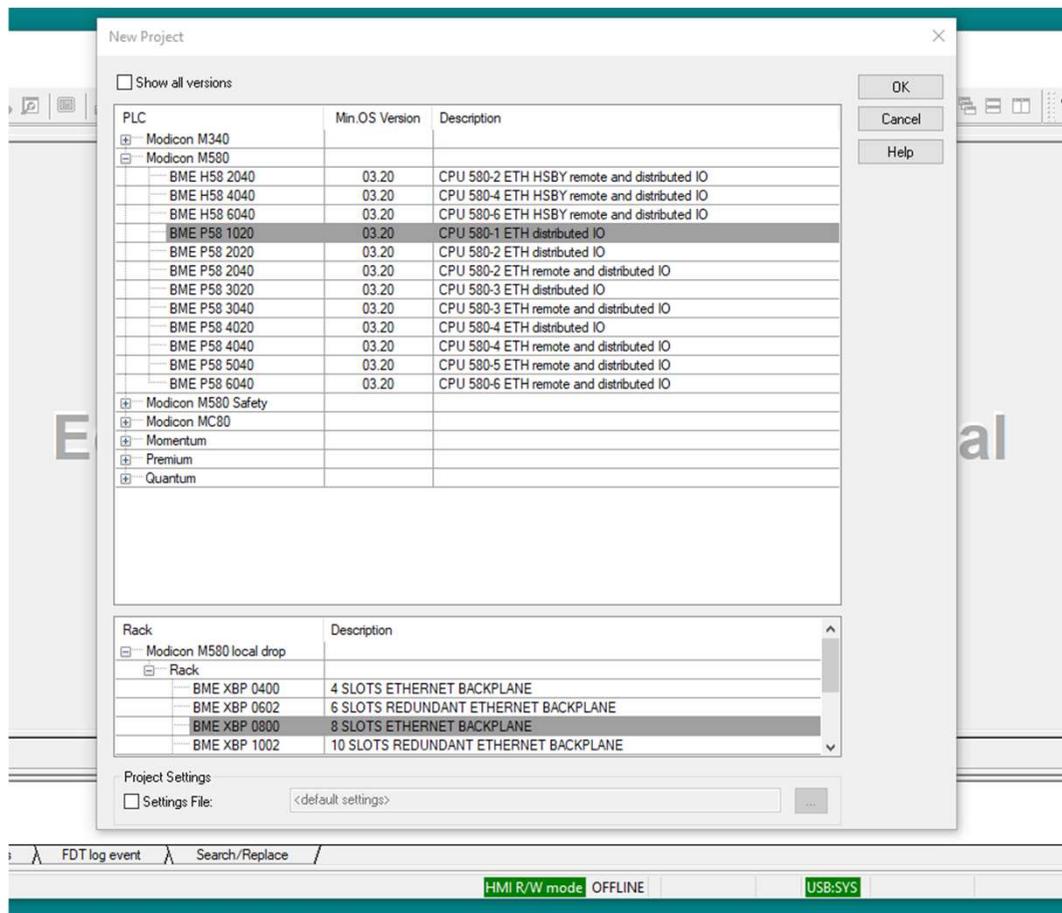
Global methodology for labs

1. Define in « Control Expert » the exact architecture, corresponding to the actual physical one, all the slots 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 GICS 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**

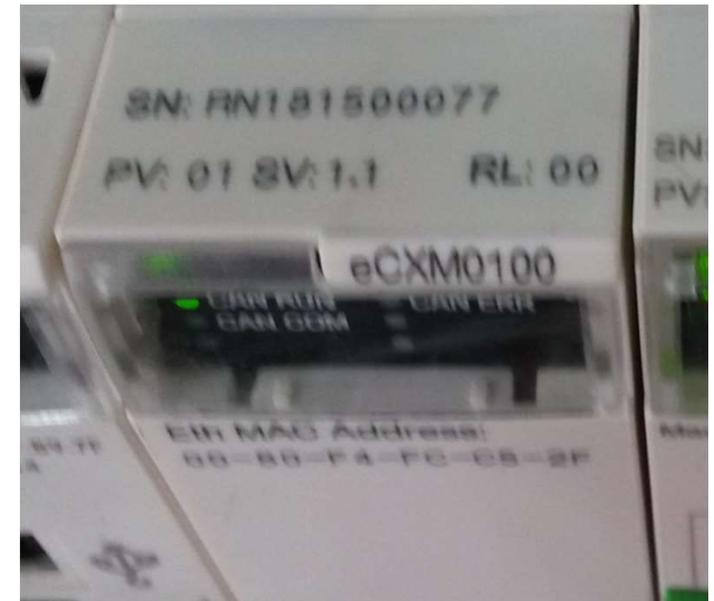
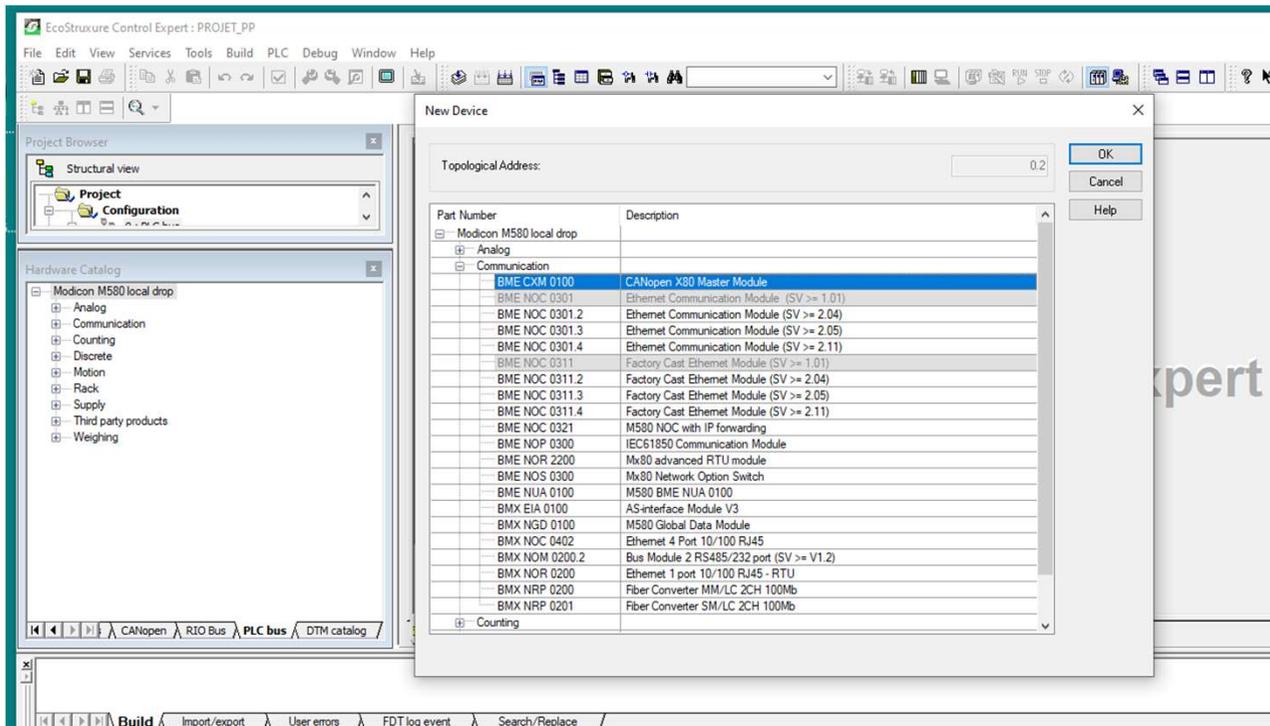
PLC Configuration



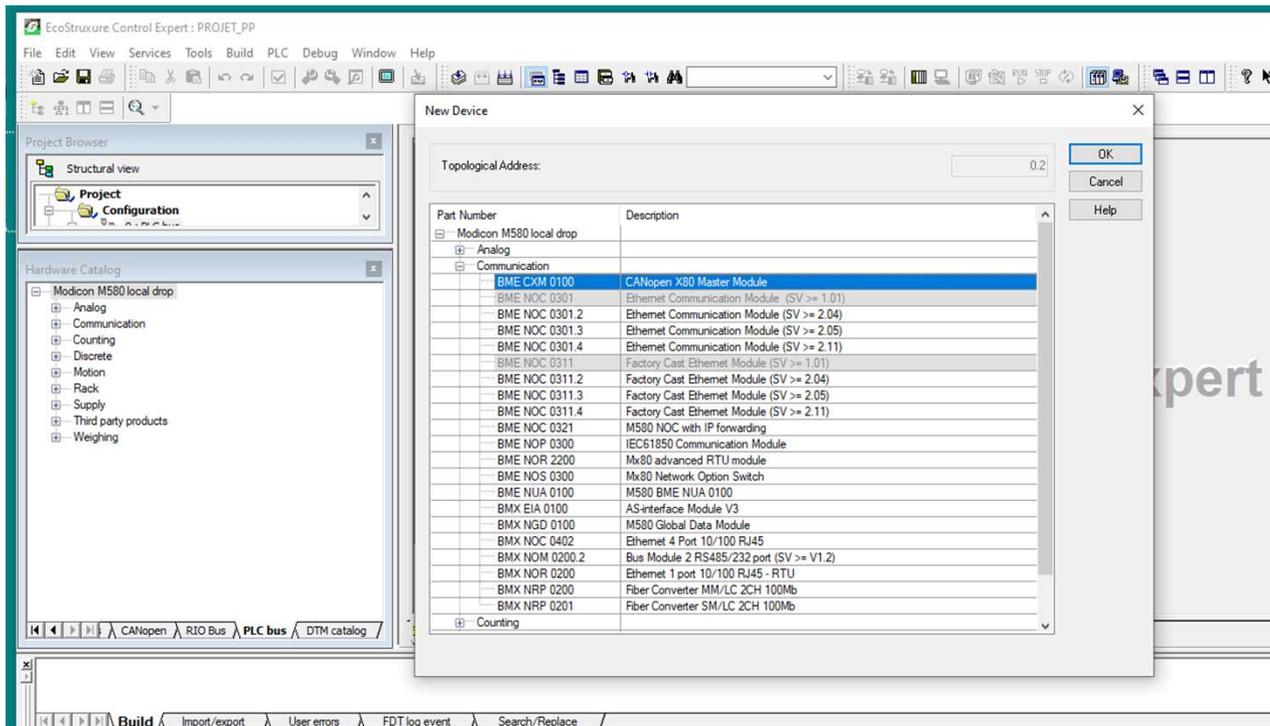
PLC Configuration



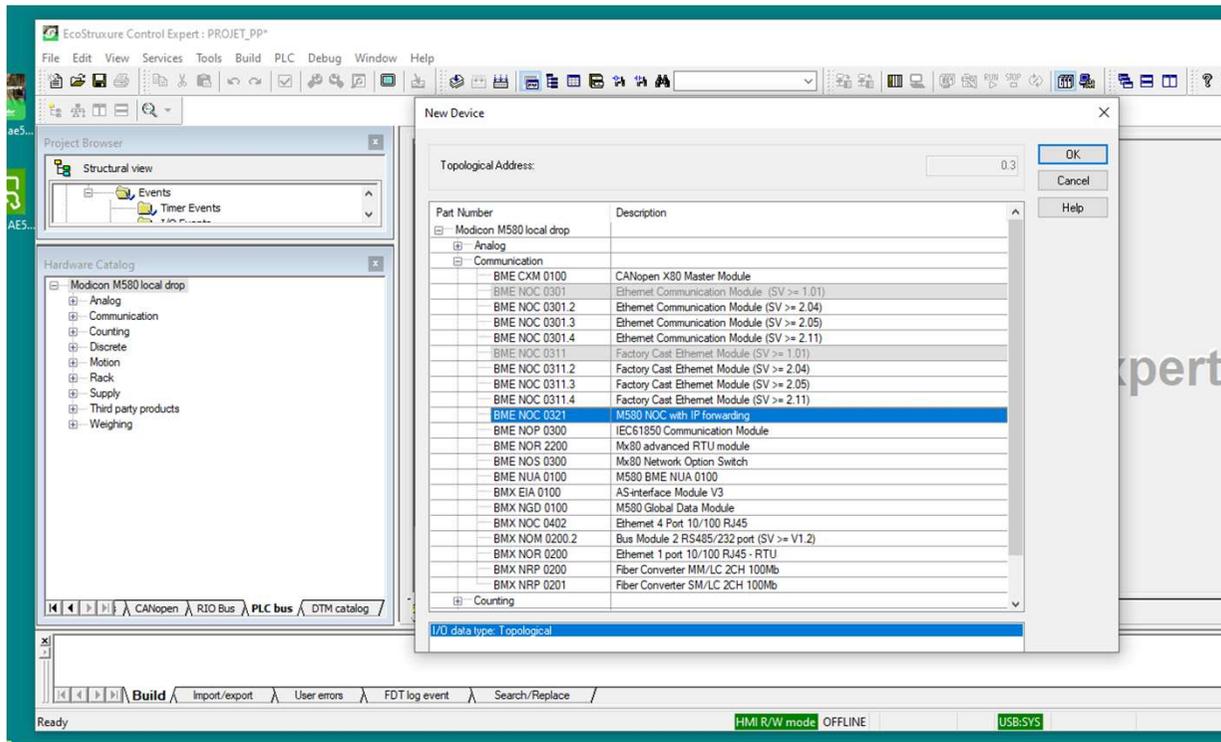
PLC Configuration



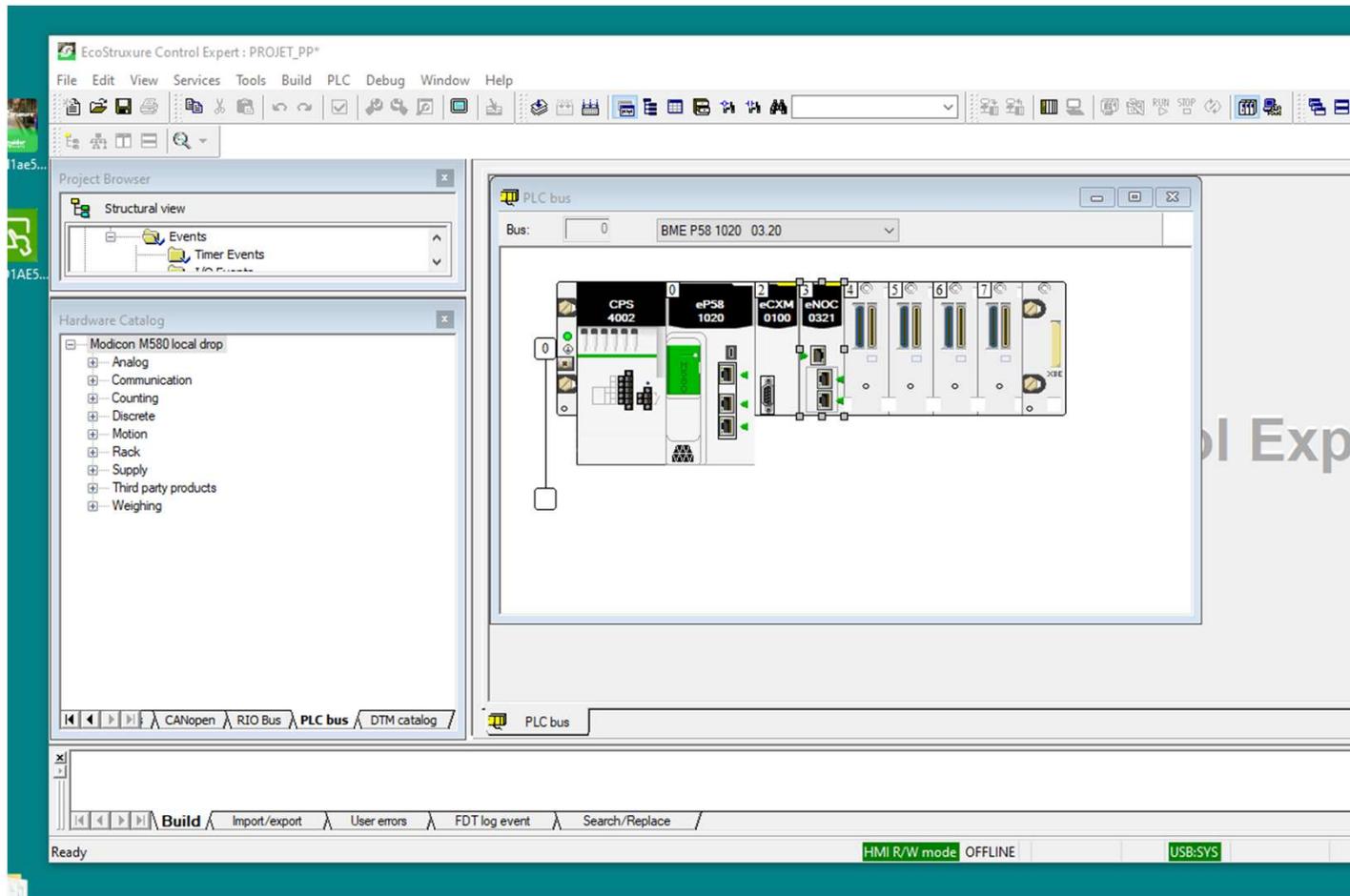
PLC Configuration



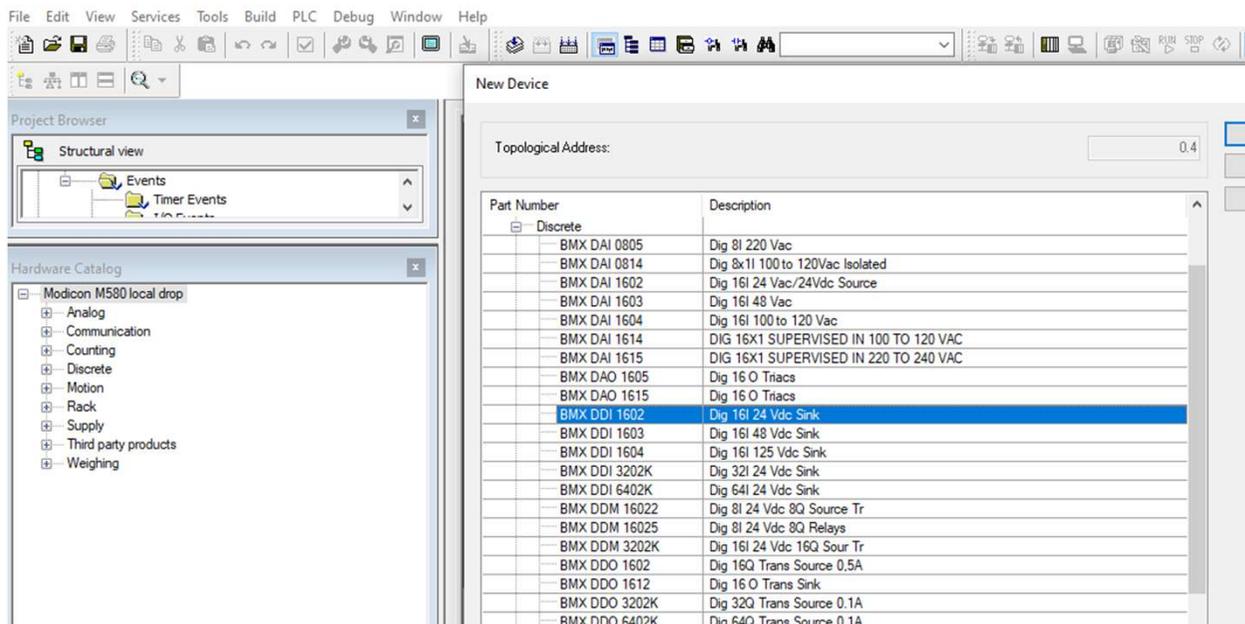
PLC Configuration



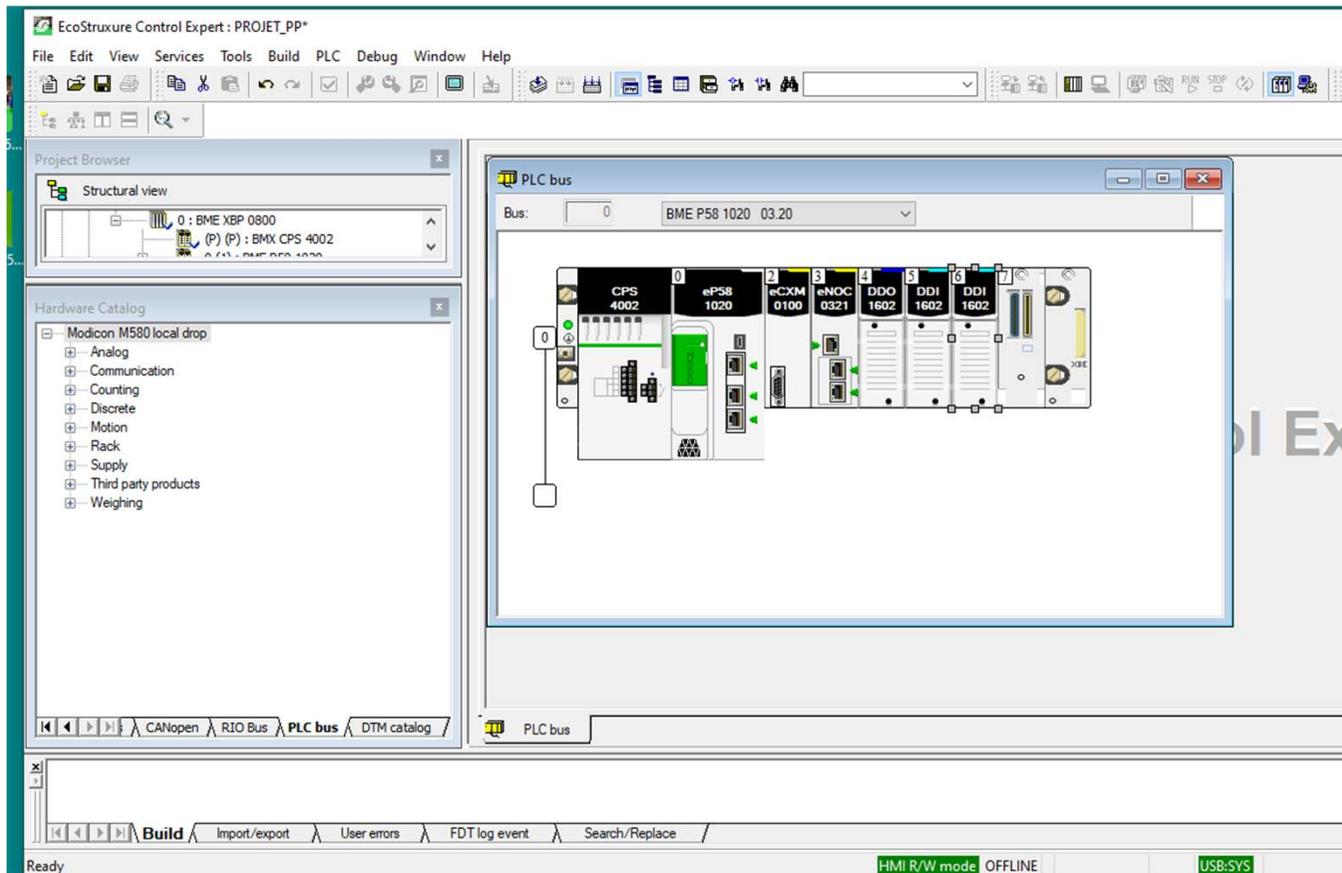
PLC Configuration



PLC Configuration

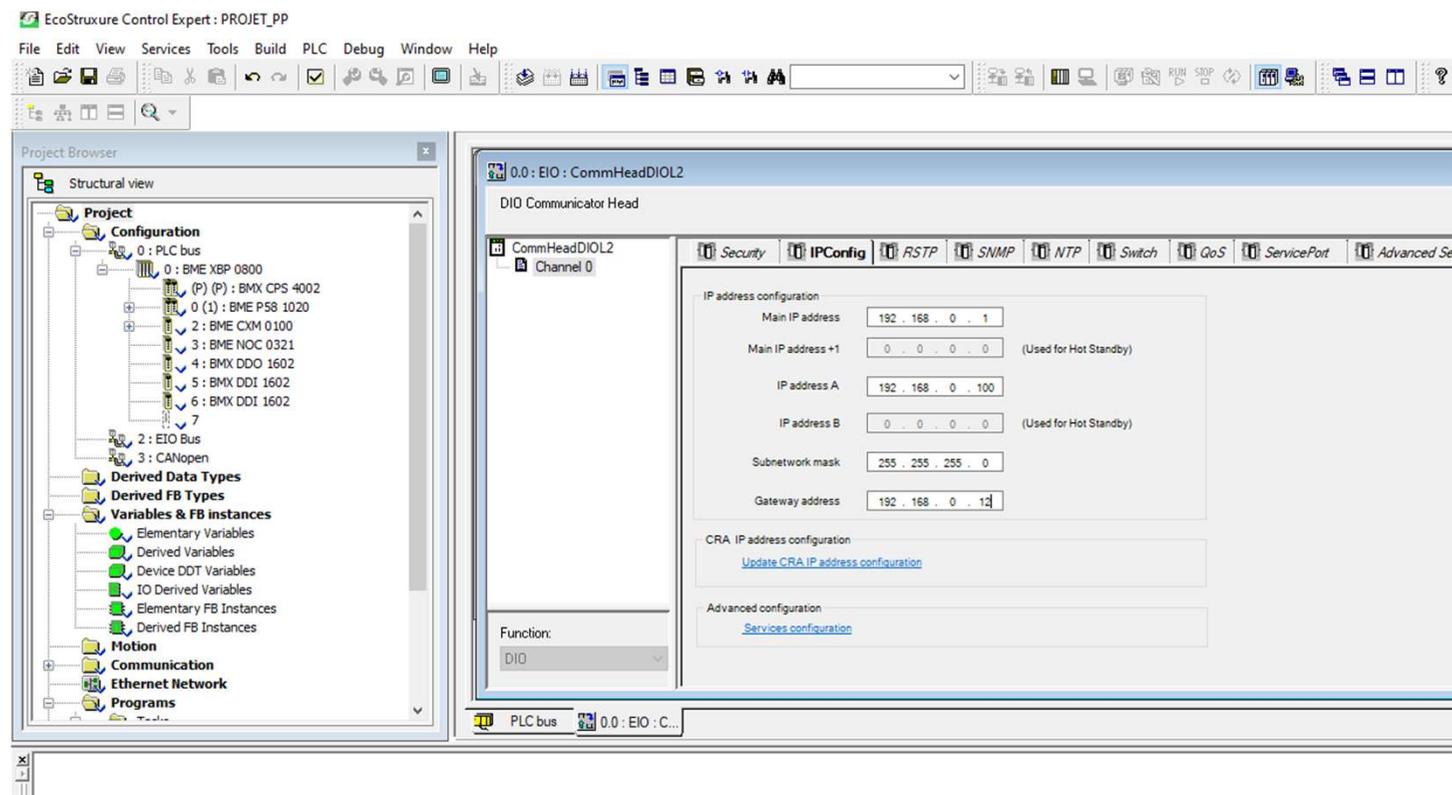


PLC Configuration



Network Configuration

- Here the gateway is 192.168.0.12.
- The choice of the gateway depends of course on the actual architecture
- If you work on a local network, you don't need any gateway !
- Several interfaces are proposed:
Main IP address is the normal address
IP address A



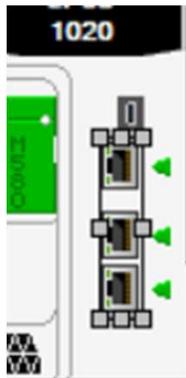
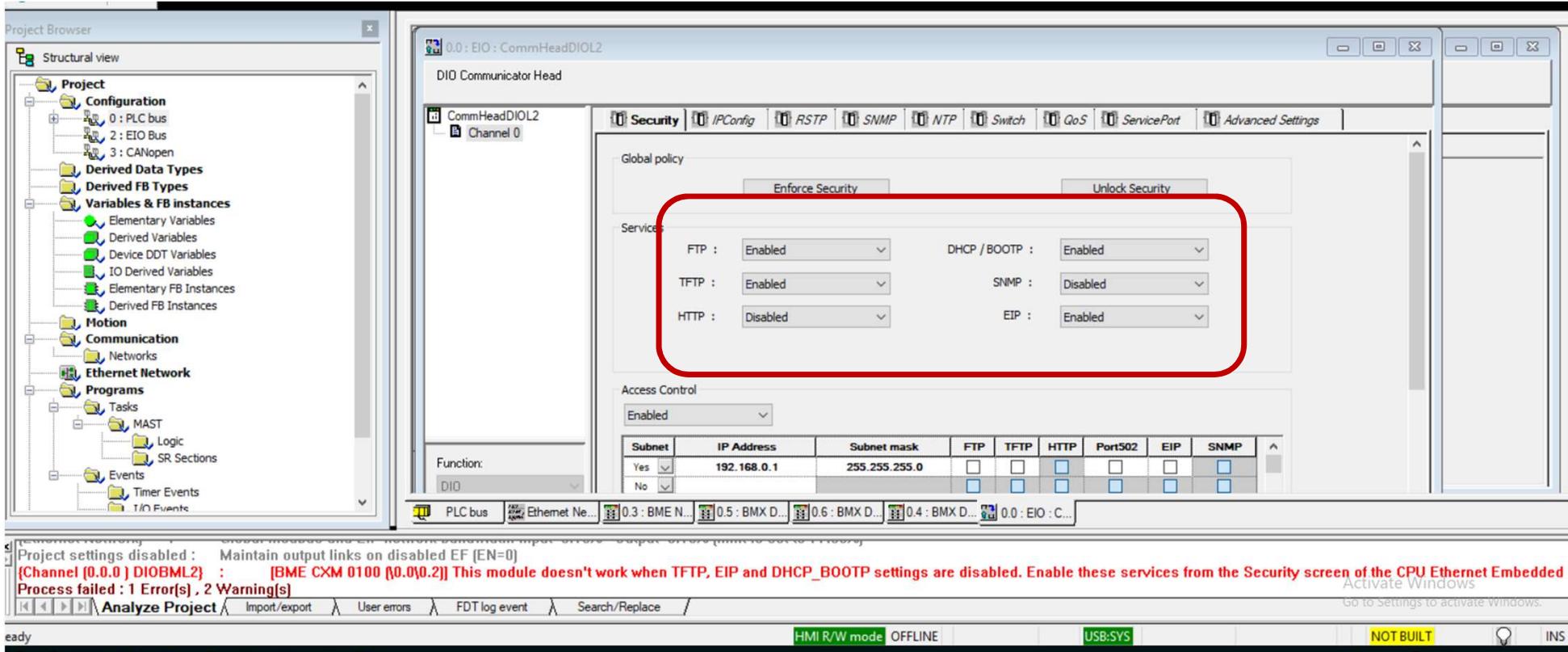
Network Configuration: M580 Addresses

For the M580, the Main IP Address is the address of the CPU and the IP Address A (& B) is the IP address of the EIO scanner service (IP address B – used for Hot Standby).

NOTE: If you change IP address A, the system may recalculate all IP addresses (including those of the drops) to keep all devices in the same subnetwork.

For the M580 Hot Standby, the Main IP address will always be assigned to the Primary PAC and the Main IP +1 will be the Standby PAC, this is an automatic switchover. The A and B IP addresses however will always be assigned to the A and B specific hardware (switch setting), they do not change on switchover

Schneider website: Last Modified on:9/29/2021

Project settings disabled : Maintain output links on disabled EF (EN=0)
[Channel (0.0.0) DIOBML2] : [BME CXM 0100 (0.0|0.2)] This module doesn't work when TFTP, EIP and DHCP_BOOTP settings are disabled. Enable these services from the Security screen of the CPU Ethernet Embedded
 Process failed : 1 Error(s) , 2 Warning(s)

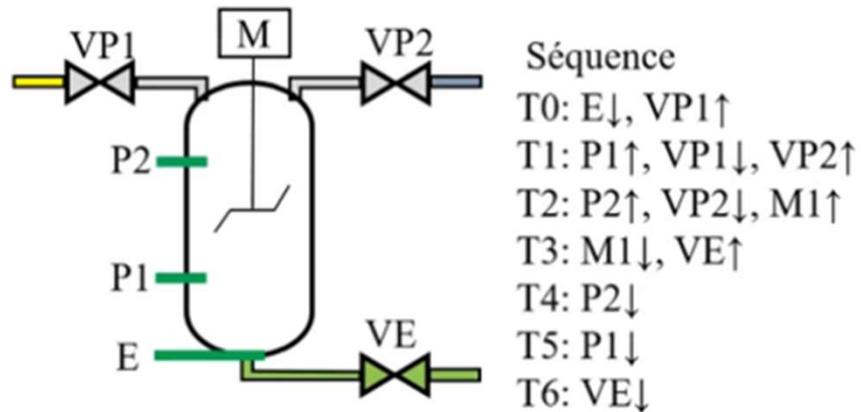
Subnet	IP Address	Subnet mask	FTP	TFTP	HTTP	Port502	EIP	SNMP
Yes	192.168.0.1	255.255.255.0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
No			<input checked="" type="checkbox"/>					

Maybe you may have the red error, in this case you should go to the Ethernet interface, and Enable the requested services, as it is done on this screen.

Network Configuration

Name	Type	Subtype	Profiles	Topo address	Configuration Mode	DHCP Enable	IP Address	Subnet Mask	Gateway Address	Identified By	Identifier	SNMP	NTP State	NTP Configuration
▼ BMEP58_ECPU_EXT	Scanner	Scanner DIO	Distributed Remote	0.0/0.0	Static	Yes	A: 192.168.0.100 Main : 192.168.0.1	255.255.255.0 255.255.255.0	192.168.0.254			IP1: 192.168.0.200 IP2: 192.168.0.201	Disabled	
CANopen_Master	Module	Gateway Ready	Distributed	0.0/0.2		Yes	192.168.0.110	255.255.255.0	192.168.0.254	Device Name	Mx80_02_BMECXM01			
BMENOC0321	Scanner	Scanner CTRL	Distributed	0.0/0.3	Static	Yes	Main : 172.16.12.1	255.255.255.0	172.16.12.254			IP1: 0.0.0.0 IP2: 0.0.0.0	Client	Primary: 0.0.0.0 Secondary: 0.0.0.0
Router->1				0.0/0.0	Static	No	192.168.0.12	255.255.255.0	0.0.0.0				Disabled	

Configuration of the variables



If we take as an example this application:

- We have 3 sensors P1, P2 and E, which are **Inputs** for the PLC
- We have four actuators VP1, VP2, VE and M which are **Outputs** for the PLC

Configuration of the variables

When using the EIO rack, it is not possible to use directly the topological addresses %I and %Q. The topological address is possible only on the "main rack".

The addressing for the EIO bus is in "Device DDT Variables". The name of the variables will be for instance something like DDM_MOD3_16.DIS_CH_OUT[16].VALUE

In order to use more easily the variable names, we can create "aliases". These aliases are actually names which can be used in programmes.

See in the next slides how to configure these variables.

Configuration of the variables (original French version, as a reminder)

L'automate **M580** ne permet **pas d'utiliser directement** des adresse **topologiques** d'entrées /sorties telles que **%I et %Q** des **racks déportés** sur le **bus EIO**.

L'adressage topologique n'est en effet possible que sur le rack principal.

L'adressage sur le bus EIO est en DDT d'équipement uniquement. Les variables ont alors des noms" à rallonge", exemple : DDM_MOD3_16.DIS_CH_OUT[16].VALUE

Pour pouvoir utiliser des variables de façon simplifiée dans le programme, il est nécessaire de créer des "Alias", ces alias consiste à leur donner un nom qu'il sera possible d'utiliser dans le programme.

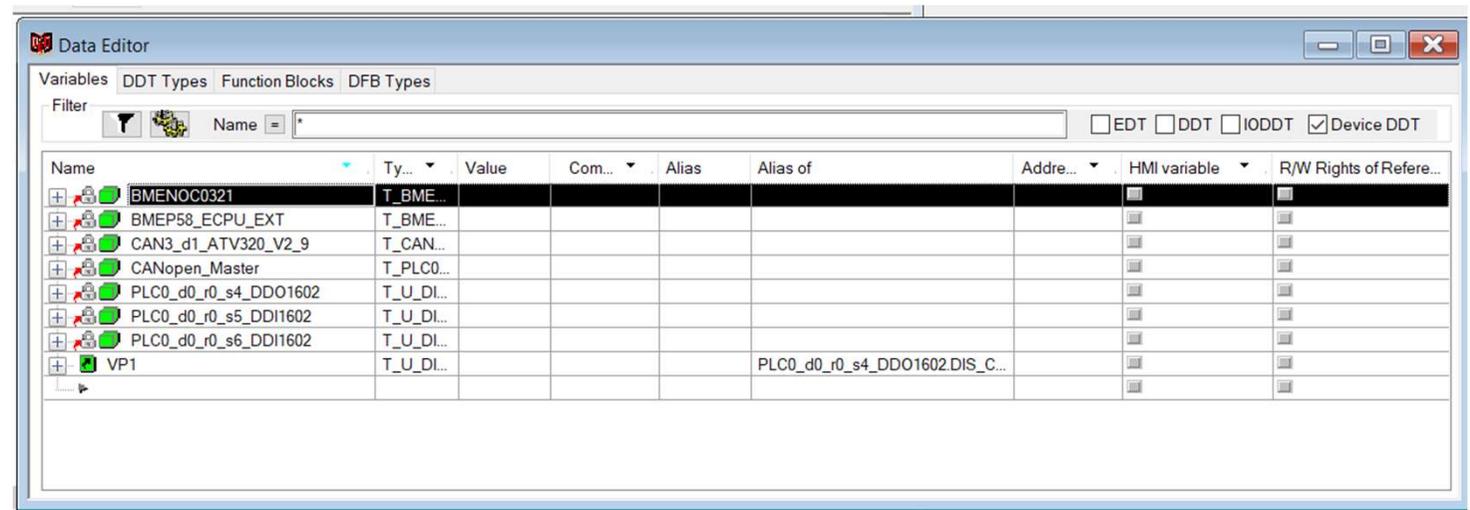
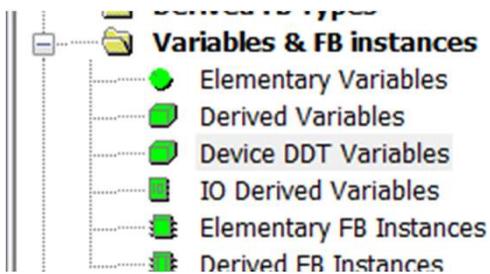
Les alias apparaitront dans la liste des variables en types EDT et non DDT d'équipement. Dans la colonne "Alias pour" vous pourrez retrouver le nom complet pour mémoire.

Consulter le document joint pour exemple

Configuration of the variables

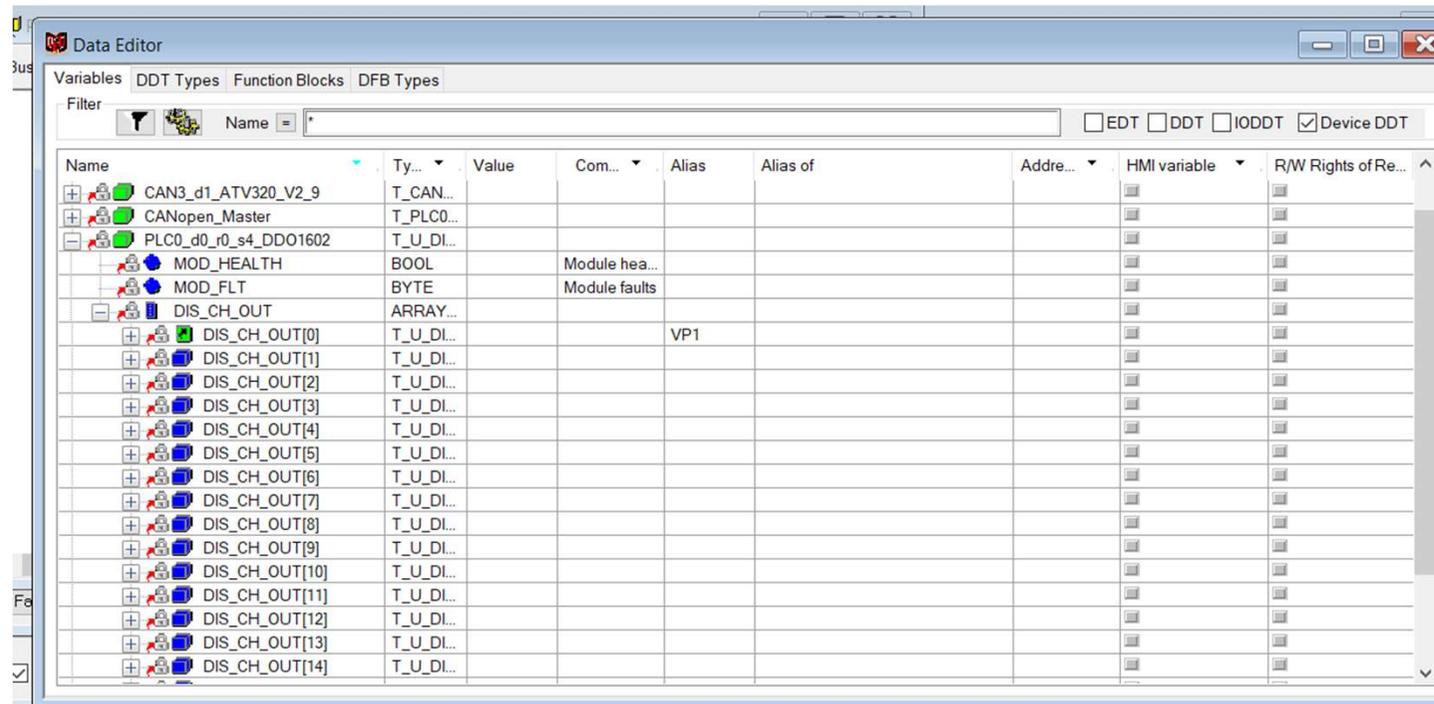
So the IO variables are not configured directly. There are « structured variables ». We should proceed through an alias.

1. Double-click on « Device DDT Variables »



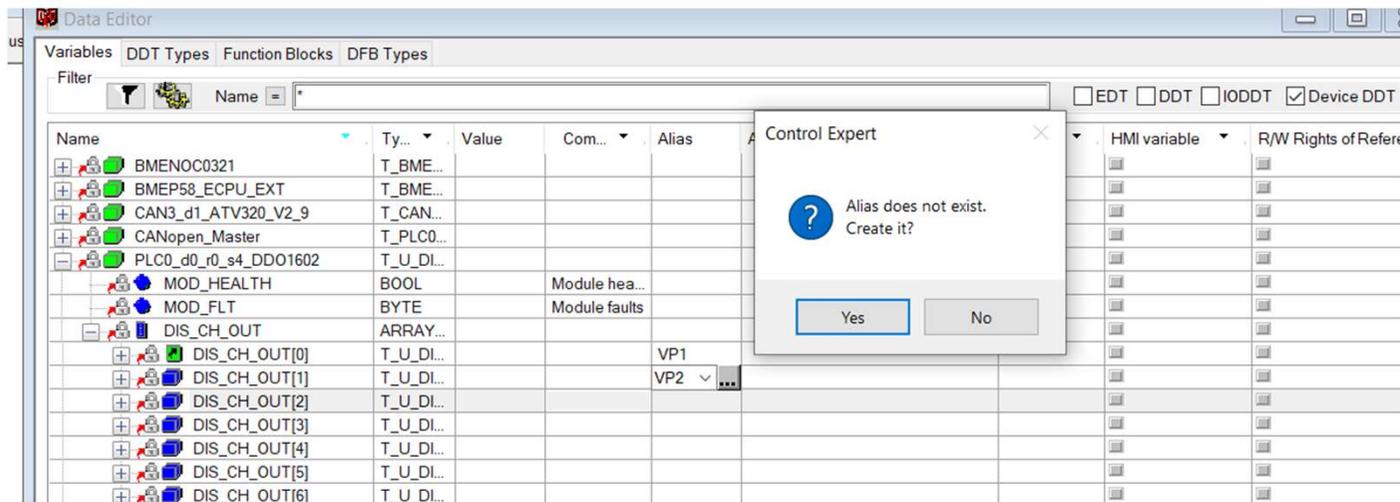
Configuration of the variables

2. Then open DDO1602 for example (Digital Output slot)
3. Here we see that VP1 has been configured on the Channel_Out 0
4. Let's configure VP2 on the channel_out 1



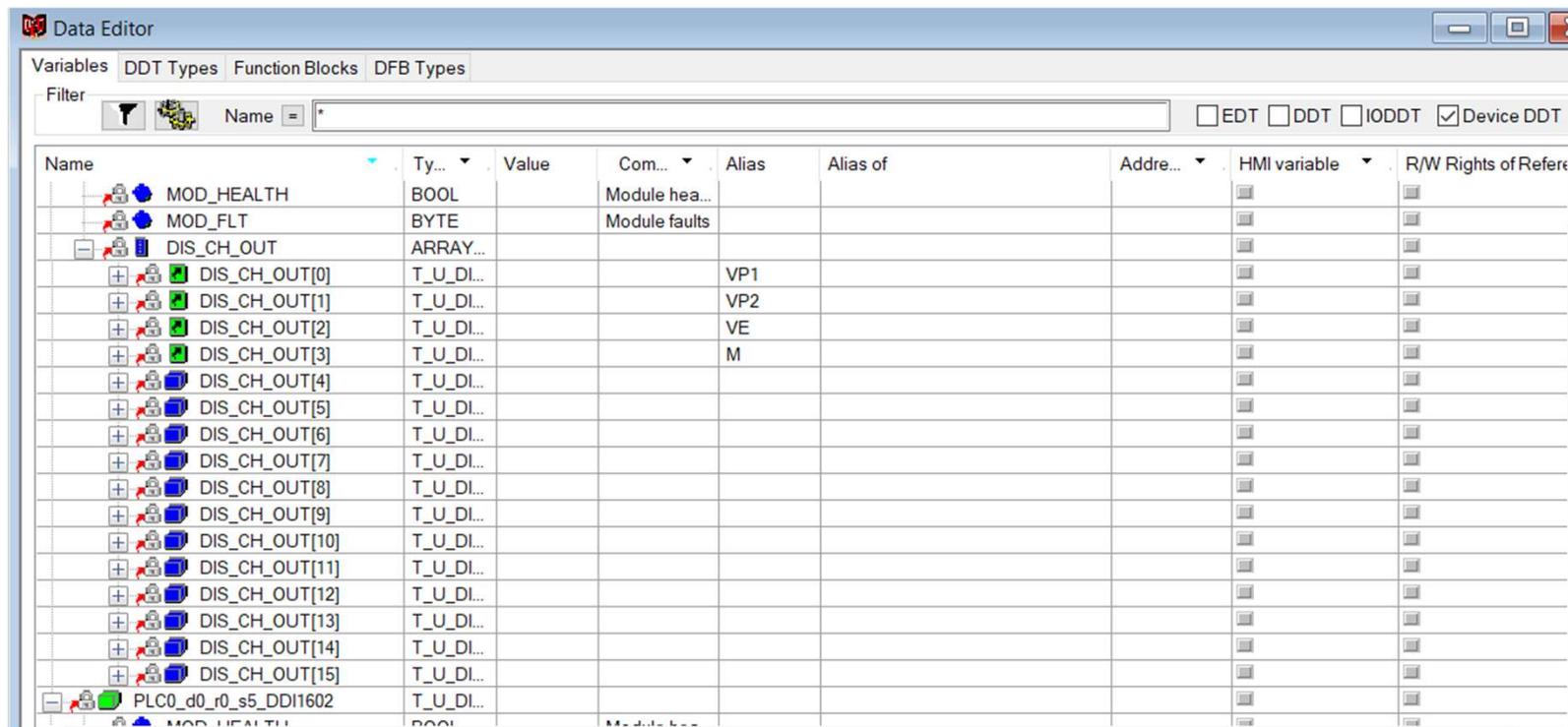
Configuration of the variables

4. Let's configure VP2 on the channel_out 1
5. We just add VP2 as an alias of Channel_out 1 (DIS_CH_OUT[1])
6. You need to confirm that you add an alias



Configuration of the variables

7. Do the same for the other outputs, you will obtain that:

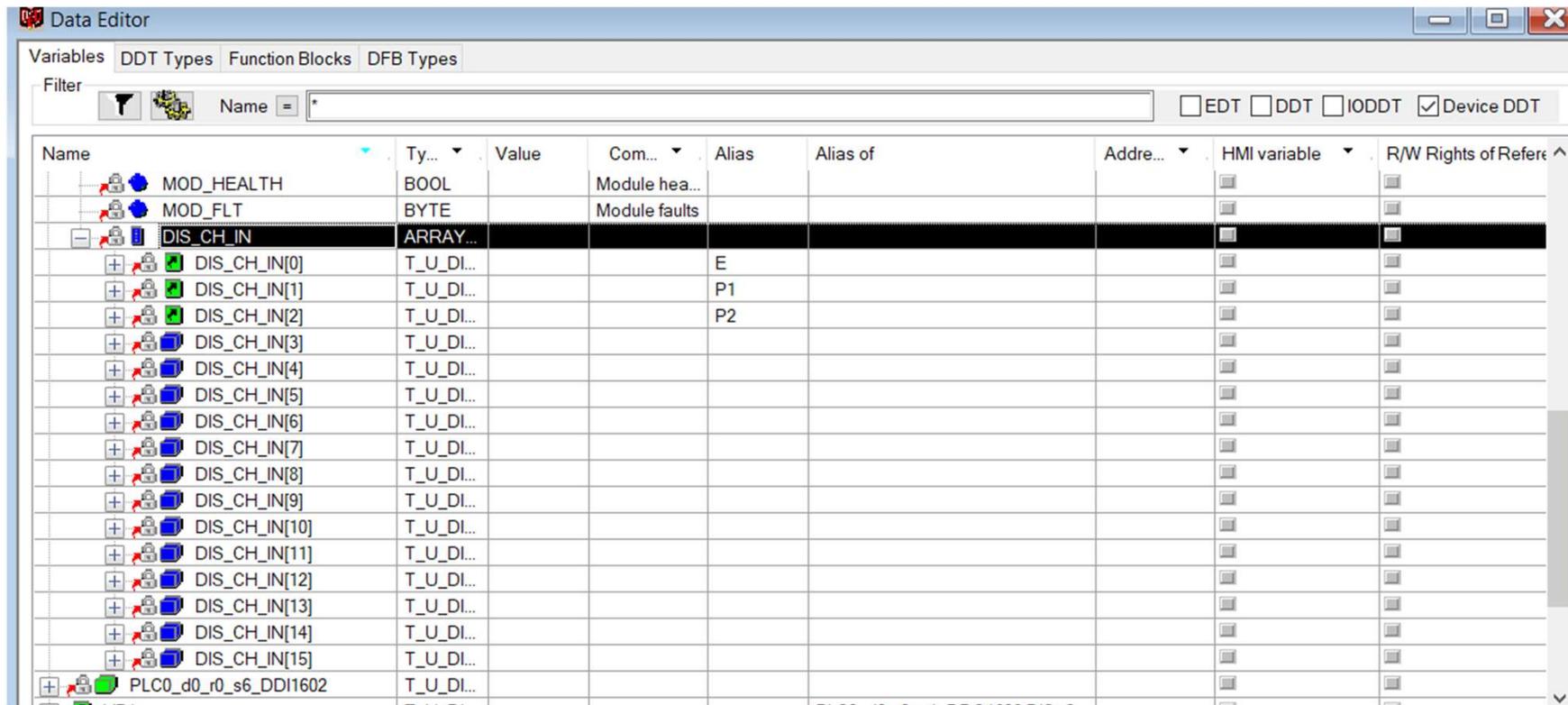


The screenshot shows the 'Data Editor' window with the 'Variables' tab selected. The 'Filter' field is empty. The 'Device DDT' checkbox is checked. The table below represents the data shown in the window.

Name	Ty...	Value	Com...	Alias	Alias of	Adresse...	HMI variable	R/W Rights of Refer...
MOD_HEALTH	BOOL		Module hea...					
MOD_FLT	BYTE		Module faults					
DIS_CH_OUT	ARRAY...							
DIS_CH_OUT[0]	T_U_DI...			VP1				
DIS_CH_OUT[1]	T_U_DI...			VP2				
DIS_CH_OUT[2]	T_U_DI...			VE				
DIS_CH_OUT[3]	T_U_DI...			M				
DIS_CH_OUT[4]	T_U_DI...							
DIS_CH_OUT[5]	T_U_DI...							
DIS_CH_OUT[6]	T_U_DI...							
DIS_CH_OUT[7]	T_U_DI...							
DIS_CH_OUT[8]	T_U_DI...							
DIS_CH_OUT[9]	T_U_DI...							
DIS_CH_OUT[10]	T_U_DI...							
DIS_CH_OUT[11]	T_U_DI...							
DIS_CH_OUT[12]	T_U_DI...							
DIS_CH_OUT[13]	T_U_DI...							
DIS_CH_OUT[14]	T_U_DI...							
DIS_CH_OUT[15]	T_U_DI...							
PLC0_d0_r0_s5_DDI1602	T_U_DI...							

Configuration of the variables

8. Do the same for the inputs on the Digital input (the first one, slot5 (the one on slot6 is not connected yet), you will obtain that!

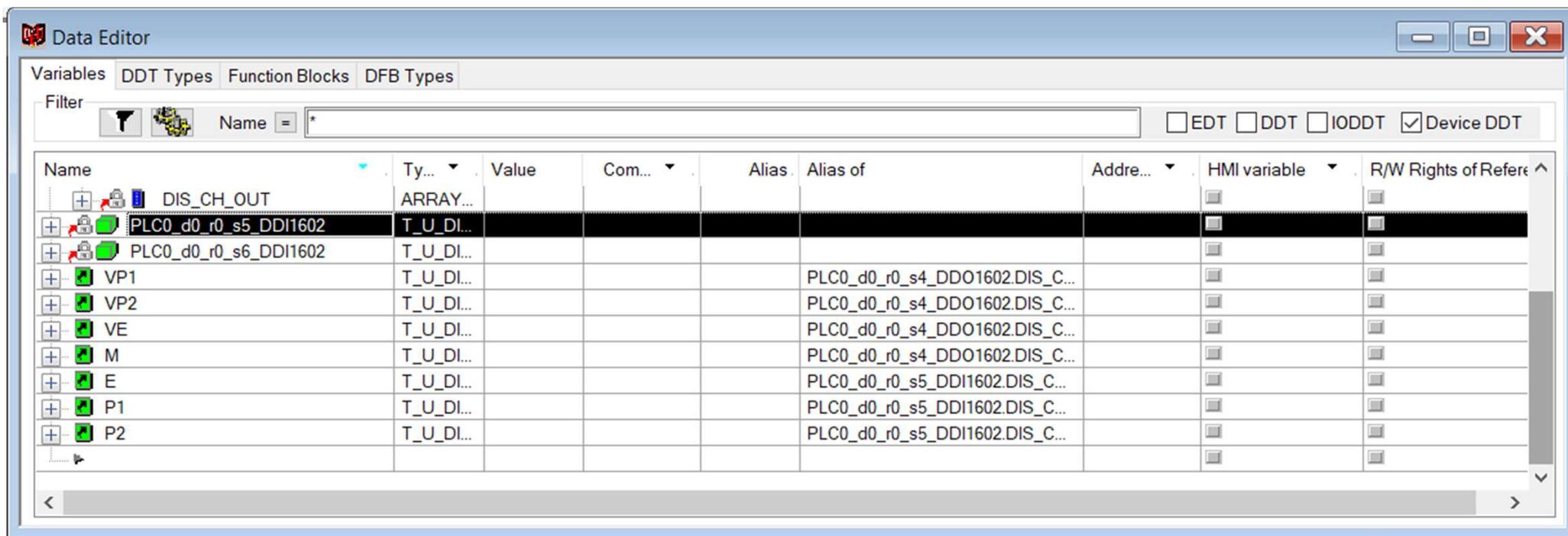


The screenshot shows the 'Data Editor' window with the 'Variables' tab selected. The table below represents the data shown in the interface.

Name	Ty...	Value	Com...	Alias	Alias of	Adresse	HMI variable	R/W Rights of Refer...
MOD_HEALTH	BOOL		Module hea...					
MOD_FLT	BYTE		Module faults					
DIS_CH_IN	ARRAY...							
DIS_CH_IN[0]	T_U_DI...			E				
DIS_CH_IN[1]	T_U_DI...			P1				
DIS_CH_IN[2]	T_U_DI...			P2				
DIS_CH_IN[3]	T_U_DI...							
DIS_CH_IN[4]	T_U_DI...							
DIS_CH_IN[5]	T_U_DI...							
DIS_CH_IN[6]	T_U_DI...							
DIS_CH_IN[7]	T_U_DI...							
DIS_CH_IN[8]	T_U_DI...							
DIS_CH_IN[9]	T_U_DI...							
DIS_CH_IN[10]	T_U_DI...							
DIS_CH_IN[11]	T_U_DI...							
DIS_CH_IN[12]	T_U_DI...							
DIS_CH_IN[13]	T_U_DI...							
DIS_CH_IN[14]	T_U_DI...							
DIS_CH_IN[15]	T_U_DI...							
PLC0_d0_r0_s6_DDI1602	T_U_DI...							

Configuration of the variables

9. Now all the variables are configured !



The screenshot shows the 'Data Editor' window with the 'Variables' tab selected. The 'Filter' field is empty, and the 'Name' field contains an asterisk (*). The 'Device DDT' checkbox is checked. The table below lists the configured variables:

Name	Ty...	Value	Com...	Alias	Alias of	Addre...	HMI variable	R/W Rights of Refer...
DIS_CH_OUT	ARRAY...							
PLC0_d0_r0_s5_DDI1602	T_U_DI...							
PLC0_d0_r0_s6_DDI1602	T_U_DI...							
VP1	T_U_DI...				PLC0_d0_r0_s4_DDO1602.DIS_C...			
VP2	T_U_DI...				PLC0_d0_r0_s4_DDO1602.DIS_C...			
VE	T_U_DI...				PLC0_d0_r0_s4_DDO1602.DIS_C...			
M	T_U_DI...				PLC0_d0_r0_s4_DDO1602.DIS_C...			
E	T_U_DI...				PLC0_d0_r0_s5_DDI1602.DIS_C...			
P1	T_U_DI...				PLC0_d0_r0_s5_DDI1602.DIS_C...			
P2	T_U_DI...				PLC0_d0_r0_s5_DDI1602.DIS_C...			

Configuration of the variables

10. These variables (like E or M for instance on the screen below) are actually « structured variables ». A structured variable is composed of several « elementary variables »: here E.CH_HEALTH to use for diagnosis, and **E.VALUE which is the actual variable we are interested in !**

11. Just for information, you can see that the « Discrete output » itself is EBOOL

- ! EBOOL better than BOOL!

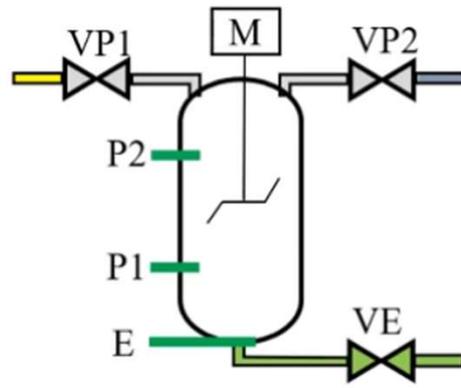
Contains the present value

The previous value

One information indicating if the bit has been forced or not

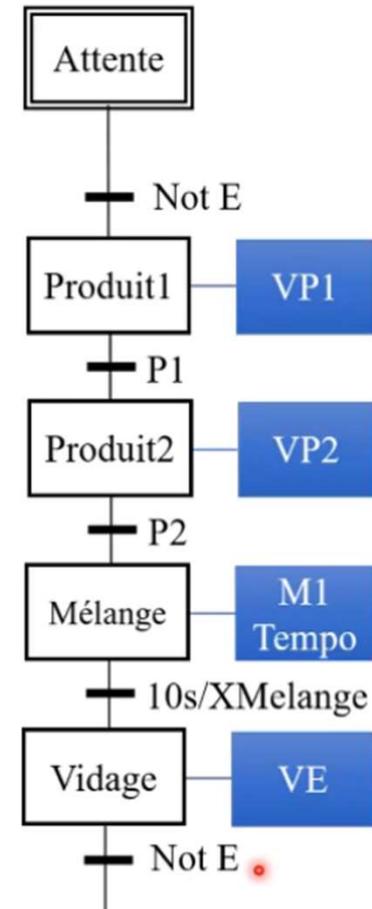
DIS_CH_OUT[0]	T_U_DI...		VP1			
CH_HEALTH	BOOL	Channel he...				
VALUE	EBOOL	Discrete out...				
DIS_CH_OUT[1]	T_U_DI...		VP2			
CANopen_Master	T_PLCO...					
E	T_U_DI...			PLC0_d0_r0_s5_DDI1602.DIS_C...		
CH_HEALTH	BOOL	Channel he...				
VALUE	EBOOL	Discrete inp...				
M	T_U_DI...			PLC0_d0_r0_s4_DDO1602.DIS_C...		
CH_HEALTH	BOOL	Channel he...				
VALUE	EBOOL	Discrete out...				
P1	T_U_DI...			PLC0_d0_r0_s5_DDI1602.DIS_C...		
P2	T_U_DI...			PLC0_d0_r0_s5_DDI1602.DIS_C...		

SFC (Grafcet) to be programmed



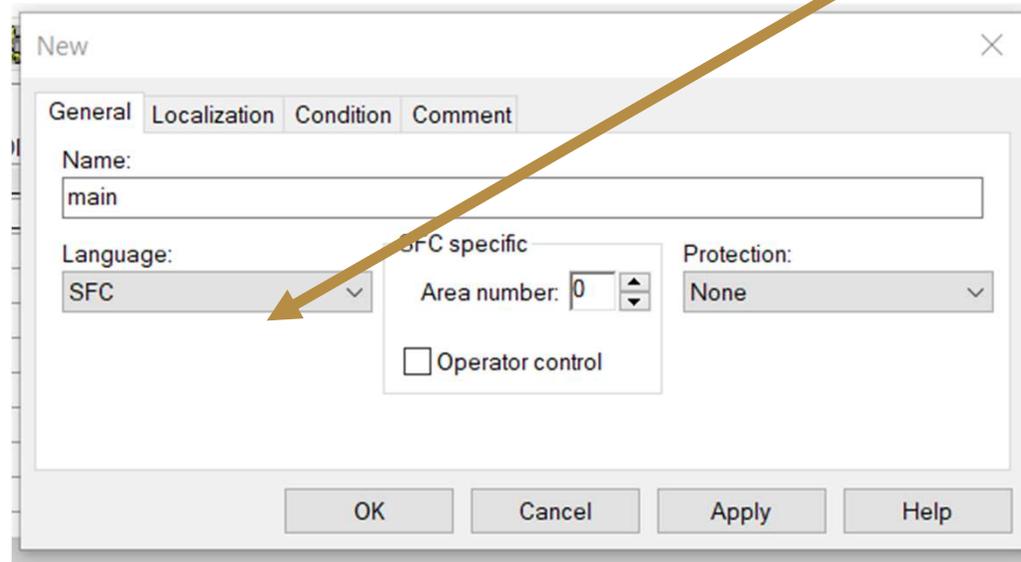
Séquence

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



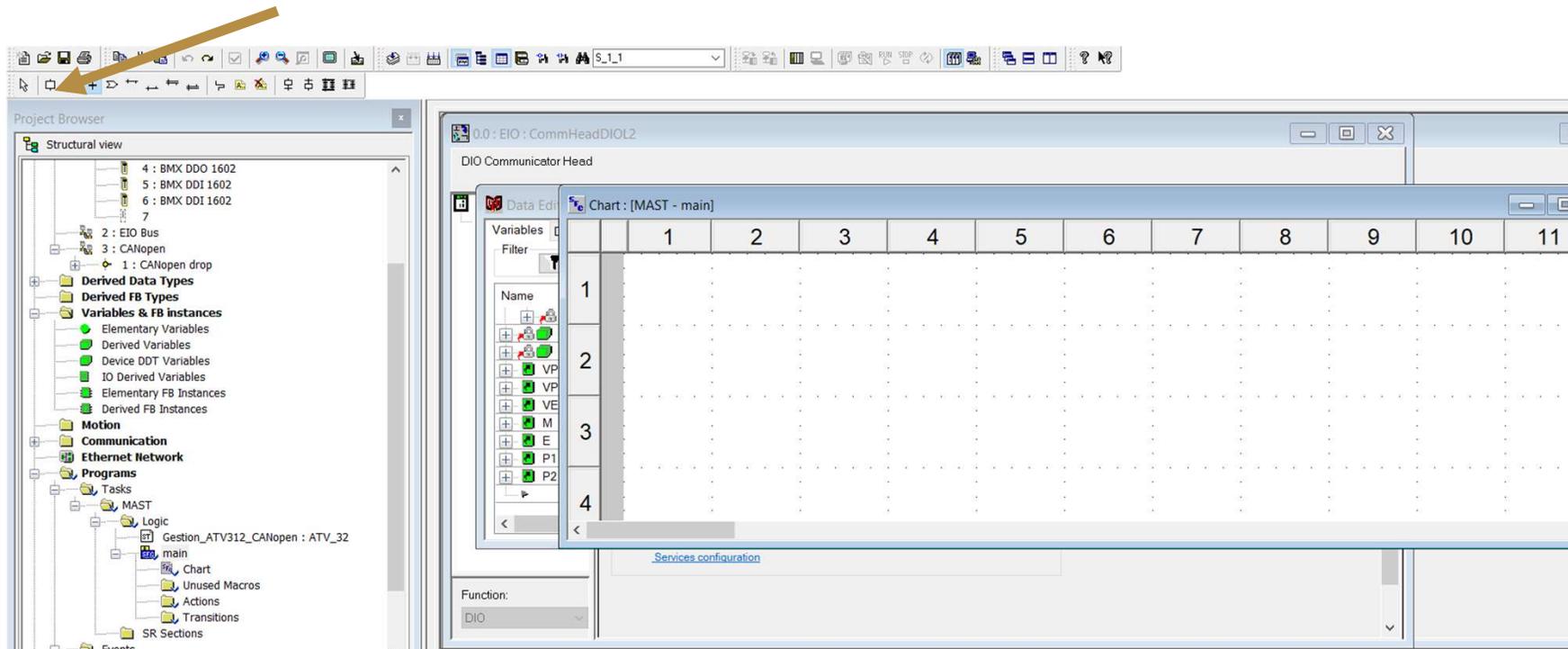
Implementation of the SFC programme

- 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 »)



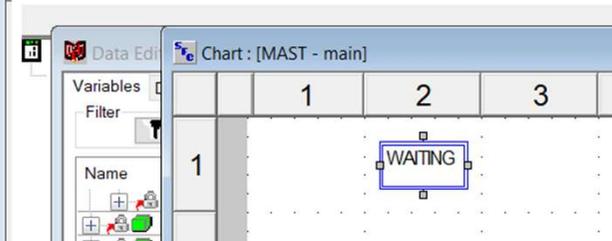
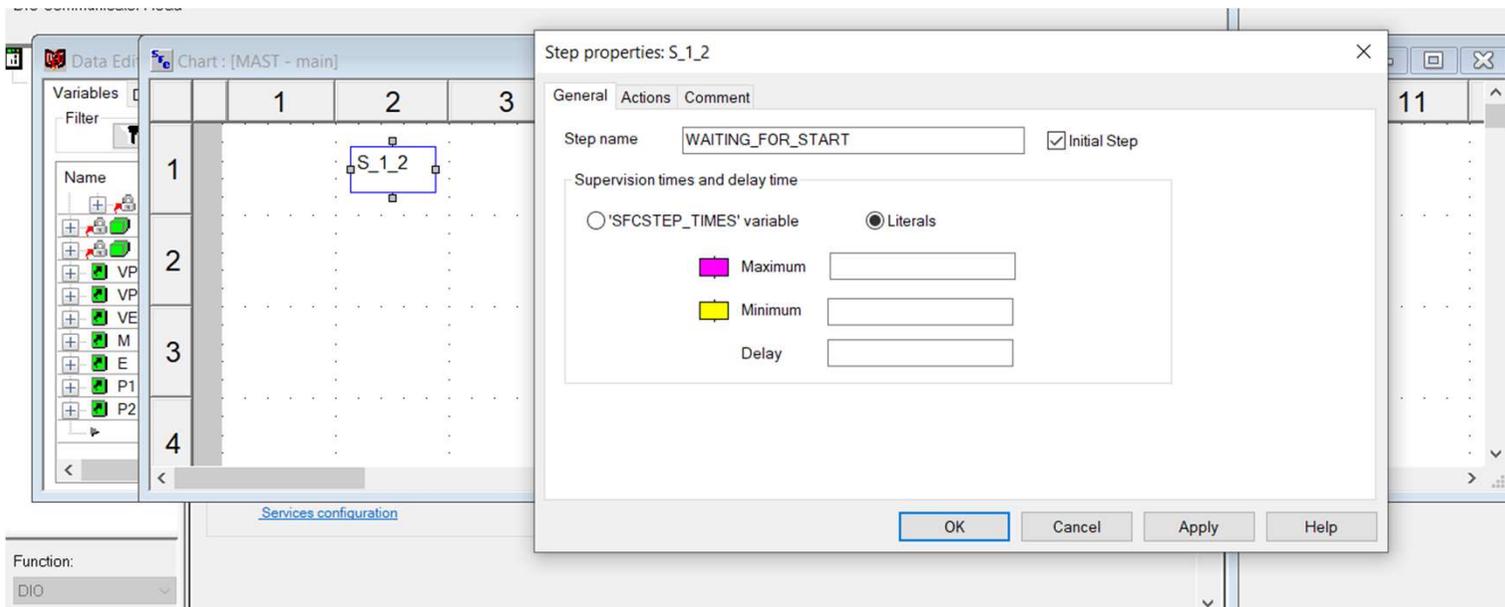
Implementation of the SFC programme

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



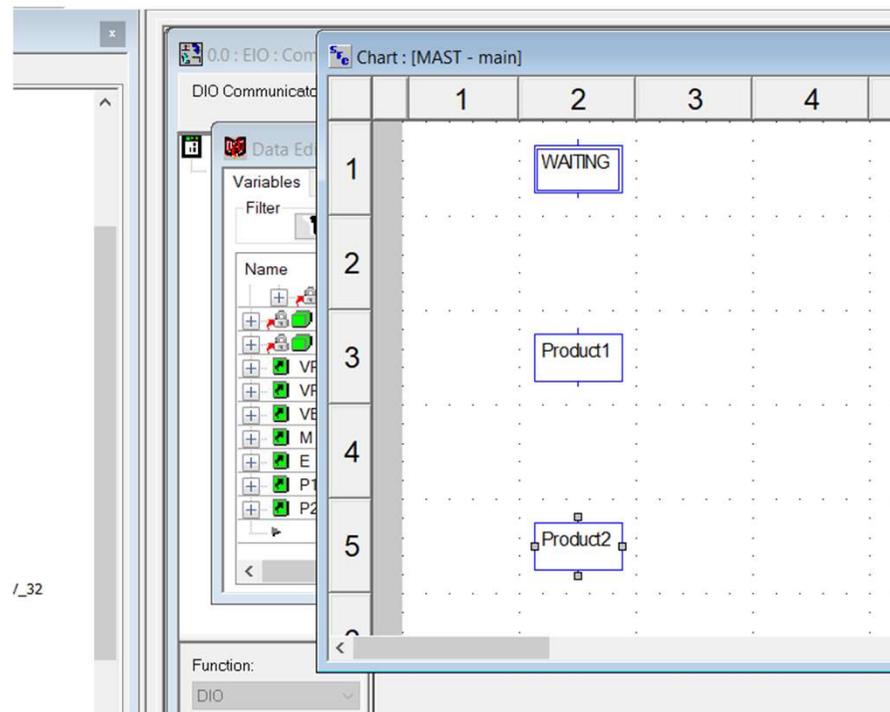
Implementation of the SFC programme

- Creation of the first state, let's call it « WAITING » for example then select « initial step »
- Validate the choice
- Now this first step, called « WAITING » is ready



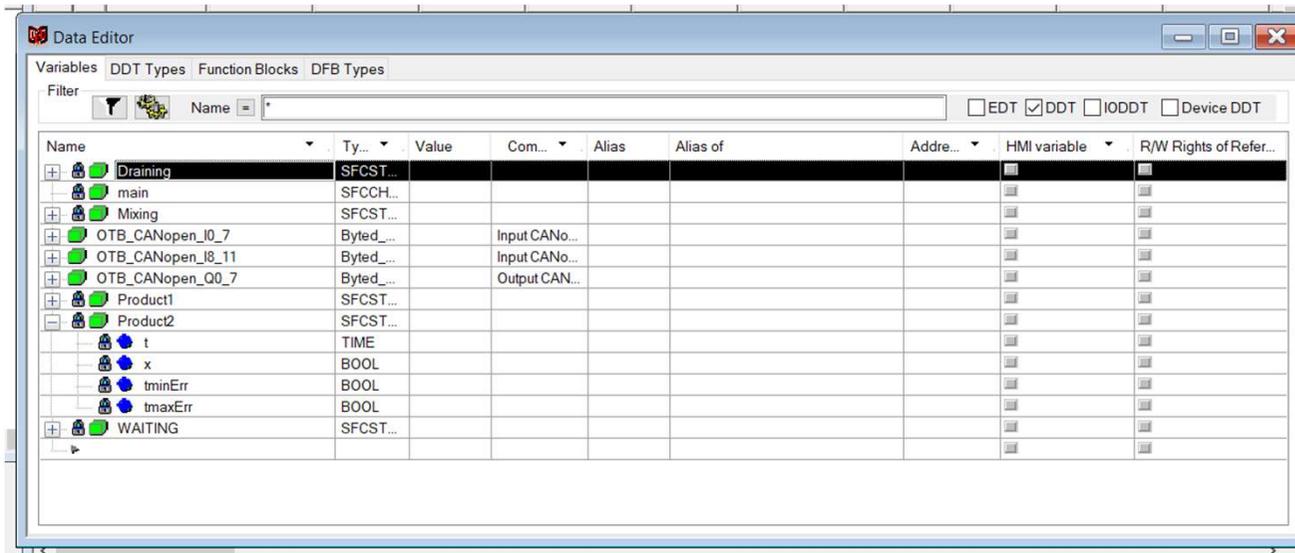
Implementation of the SFC programme

- Creation in the chart of the states « Product1 », « Product2 », « Mixing » and « Draining »



Implementation of the SFC programme

- 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 limiting variables of the residence time in the state

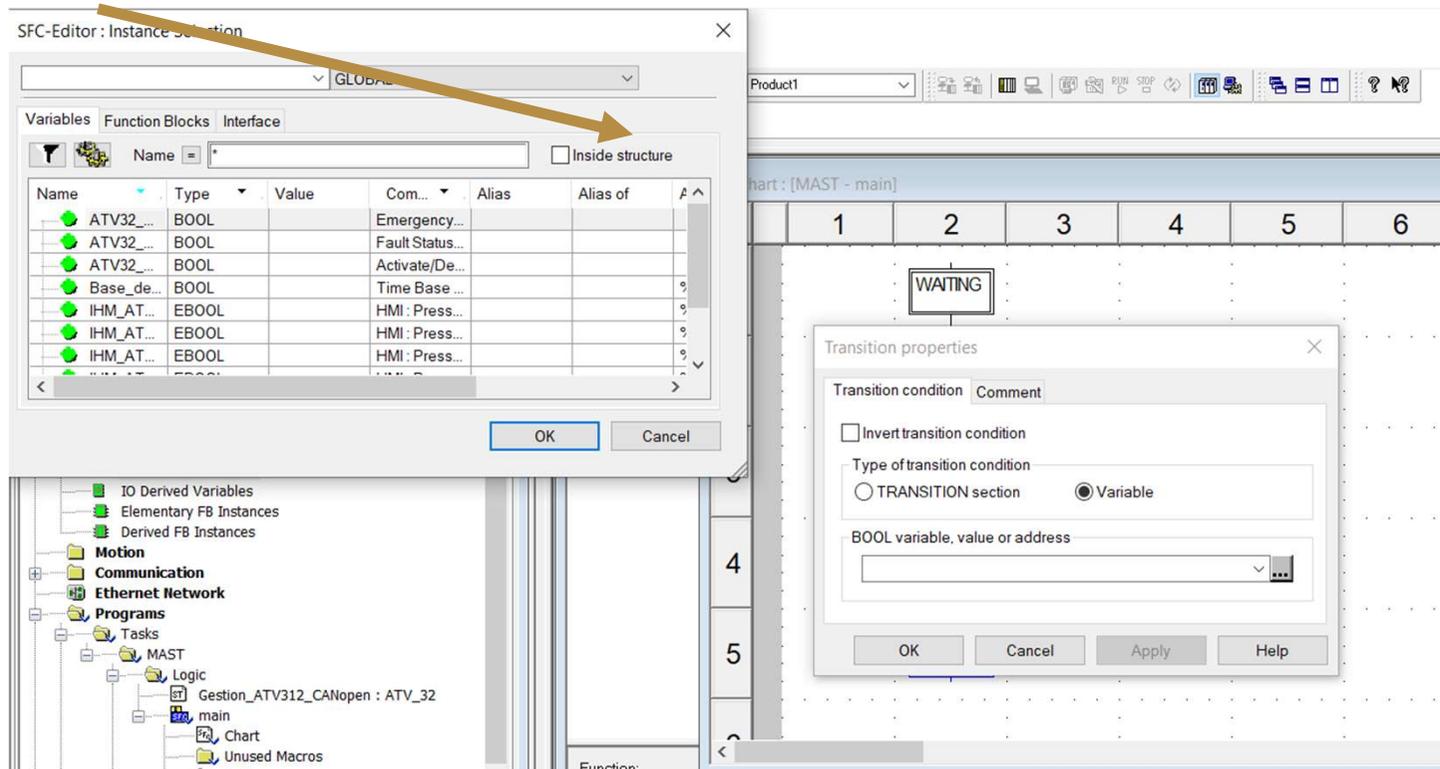


The screenshot shows the 'Data Editor' window with a table of variables. The table has columns for Name, Type, Value, Comment, Alias, Alias of, Address, HMI variable, and R/W Rights of Reference. The variables listed include states like Draining, main, Mixing, Product1, Product2, and WAITING, along with time variables (t, TminErr, TmaxErr) and CAN-related variables (OTB_CANopen_I0_7, I0_11, Q0_7).

Name	Ty...	Value	Com...	Alias	Alias of	Adre...	HMI variable	R/W Rights of Refer...
Draining	SFCST...							
main	SFCCH...							
Mixing	SFCST...							
OTB_CANopen_I0_7	Byted_...		Input CANo...					
OTB_CANopen_I0_11	Byted_...		Input CANo...					
OTB_CANopen_Q0_7	Byted_...		Output CAN...					
Product1	SFCST...							
Product2	SFCST...							
t	TIME							
x	BOOL							
tminErr	BOOL							
tmaxErr	BOOL							
WAITING	SFCST...							

Implementation of the SFC programme

- Implementation of the transitions by clicking btween two states
- Our variables are Structure-type based, so not visible at first
- So let's click on « Inside structure »

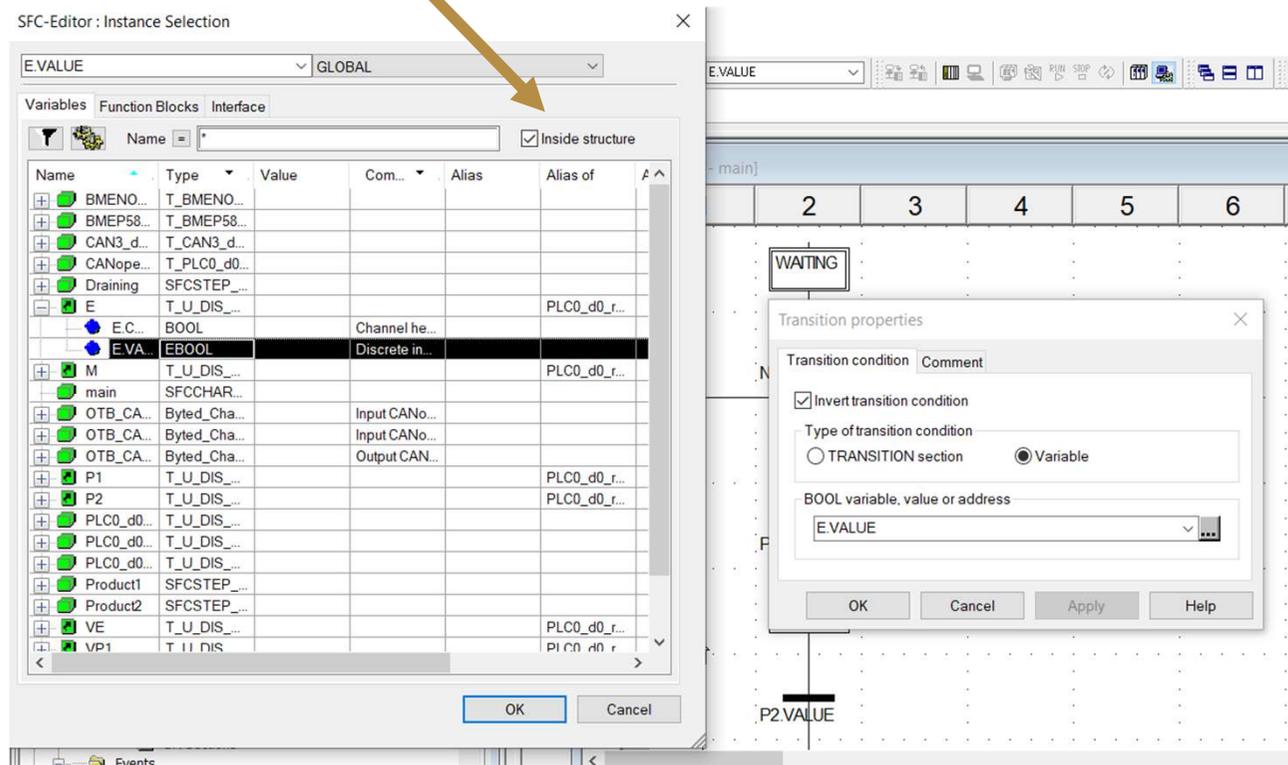


The screenshot shows the SFC-Editor interface. The 'Instance Selection' dialog box is open, displaying a table of variables. A yellow arrow points to the 'Inside structure' checkbox, which is checked. The 'Transition properties' dialog box is also open, showing the 'Transition condition' section with the 'Variable' radio button selected. The background shows a state transition diagram with a 'WAITING' state and a transition between states 4 and 5.

Name	Type	Value	Com...	Alias	Alias of
ATV32...	BOOL		Emergency...		
ATV32...	BOOL		Fault Status...		
ATV32...	BOOL		Activate/De...		
Base_de...	BOOL		Time Base ...		
IHM_AT...	EBOOL		HMI: Press...		
IHM_AT...	EBOOL		HMI: Press...		
IHM_AT...	EBOOL		HMI: Press...		

Implementation of the SFC programme

- Now we see our structured variables E, M, P1, P2, VE, VP1, VP2
- First transition is « E » or more precisely « E.VALUE »
- It should be « inversed » because this is actually « not E.VALUE »



The image shows two overlapping windows from the SFC-Editor software. The background window is the 'Instance Selection' dialog, which lists various variables. The foreground window is the 'Transition properties' dialog, which is used to configure the transition condition.

SFC-Editor : Instance Selection

Name	Type	Value	Com...	Alias	Alias of
BMENO...	T_BMENO...				
BMEP58...	T_BMEP58...				
CAN3_d...	T_CAN3_d...				
CANope...	T_PLCO_d0...				
Draining	SFCSTEP...				
E	T_U_DIS...				PLCO_d0_r...
E.C...	BOOL		Channel he...		
E.VA...	EBOOL		Discrete in...		PLCO_d0_r...
M	T_U_DIS...				PLCO_d0_r...
main	SFCCHAR...				
OTB_CA...	Byted_Cha...		Input CANo...		
OTB_CA...	Byted_Cha...		Input CANo...		
OTB_CA...	Byted_Cha...		Output CAN...		
P1	T_U_DIS...				PLCO_d0_r...
P2	T_U_DIS...				PLCO_d0_r...
PLCO_d0...	T_U_DIS...				
PLCO_d0...	T_U_DIS...				
PLCO_d0...	T_U_DIS...				
Product1	SFCSTEP...				
Product2	SFCSTEP...				
VE	T_U_DIS...				PLCO_d0_r...
VP1	T_U_DIS...				PLCO_d0_r...

Transition properties

Transition condition: Comment

Invert transition condition

Type of transition condition

TRANSITION section Variable

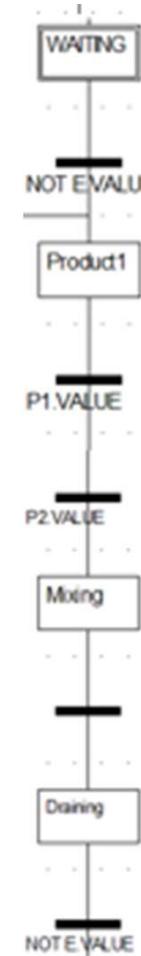
BOOL variable, value or address

E.VALUE

OK Cancel Apply Help

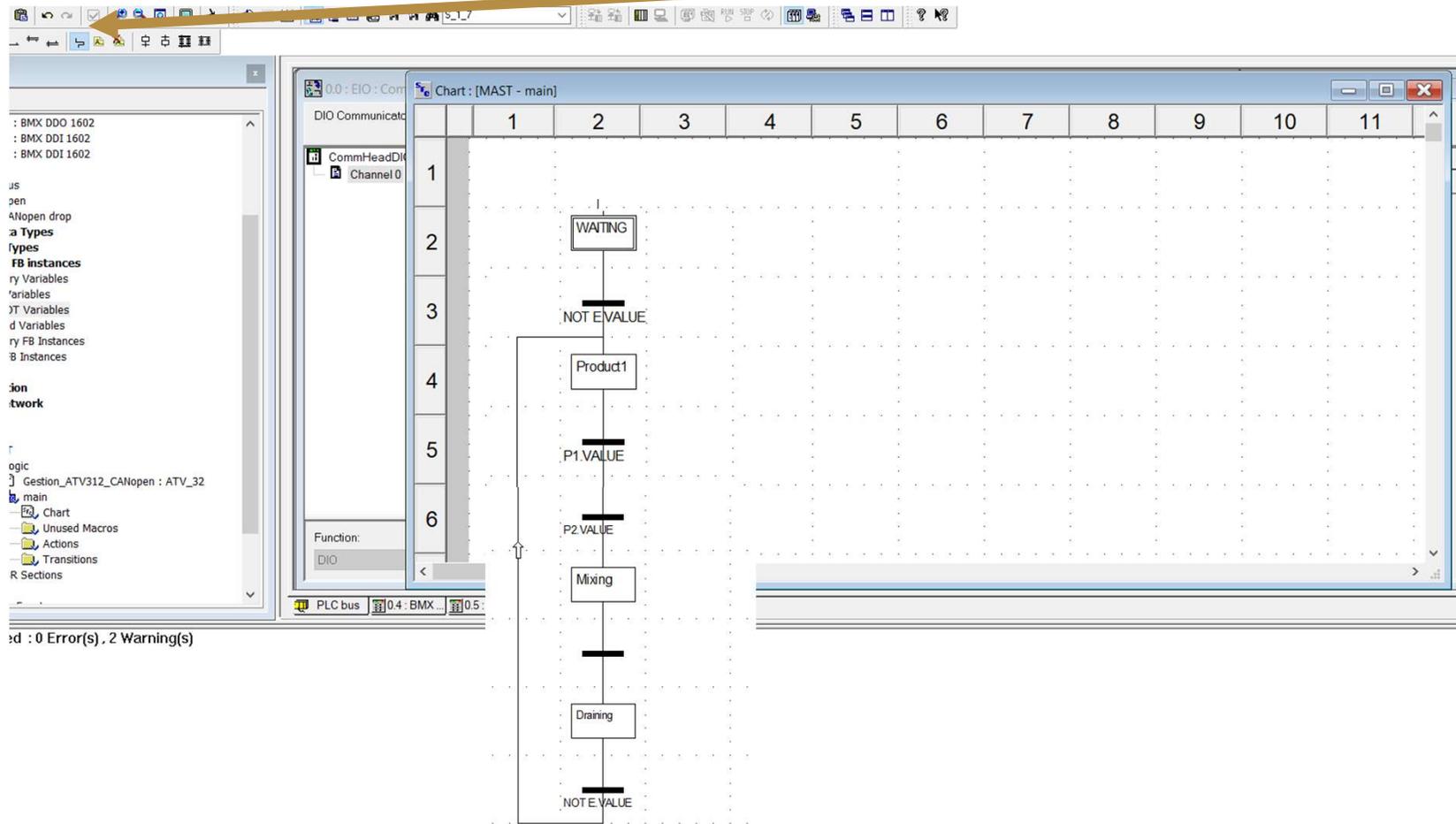
Implementation of the SFC programme

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



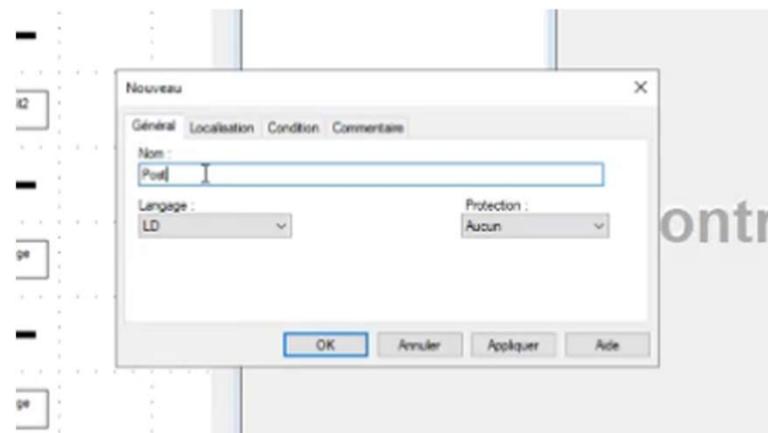
Implementation of the SFC programme

- We close the SFC by using the tool « Link »



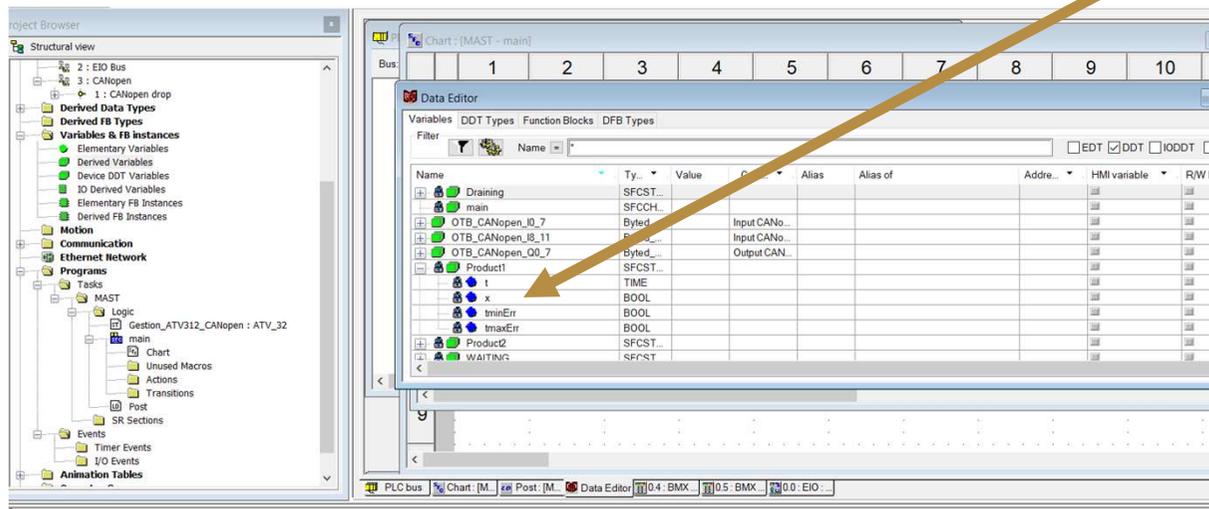
Implementation of the actions in ladder

- 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, because 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 » (SFC programme)



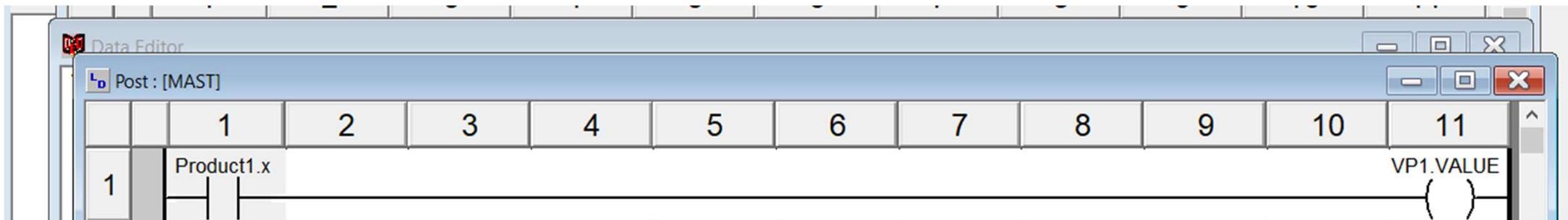
Implementation of the actions in ladder

- For the state Product1, there is one action => VP1.VALUE
- We add a ladder contact
- The condition (click on the contact) will be Product1.X (we will find it in the structure)



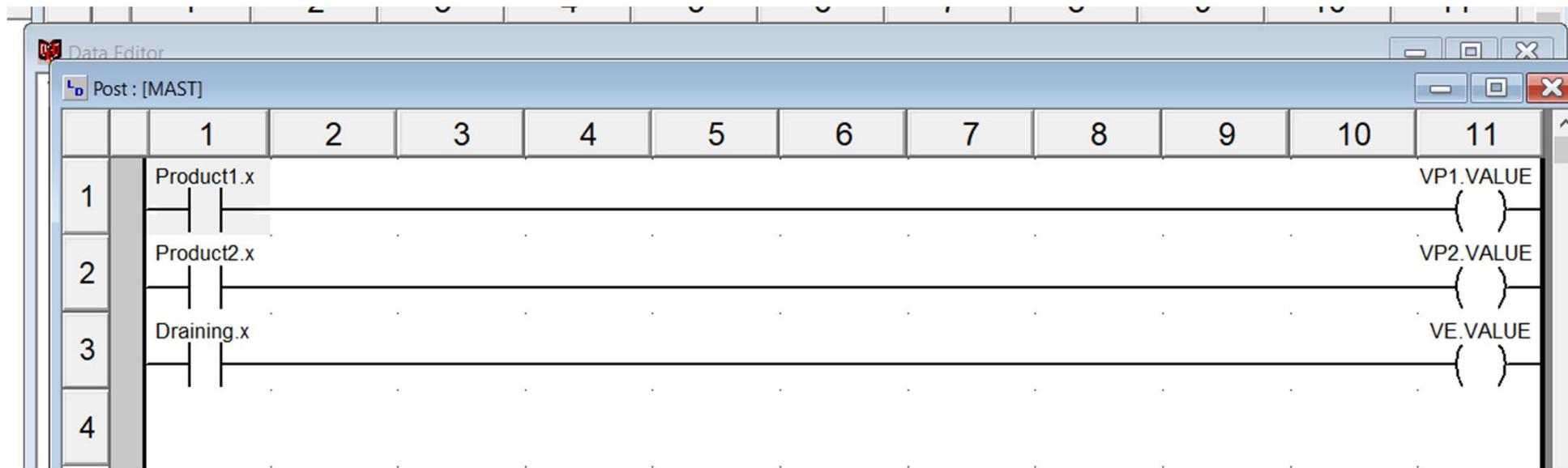
Implementation of the actions in ladder

- The output is VP1.VALUE
- This ladder means that VP1.VALUE is On when the step Product1 is On (actually when the variable Product1.X is On)
- The Coil VP1.X here is activated where Product1.X is On, and then deactivated (monostable)



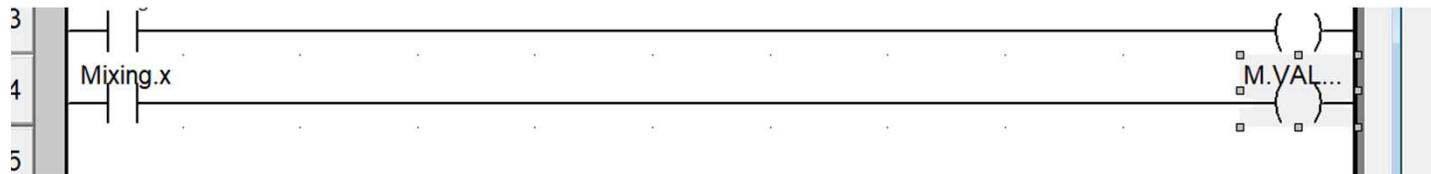
Implementation of the actions in ladder

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



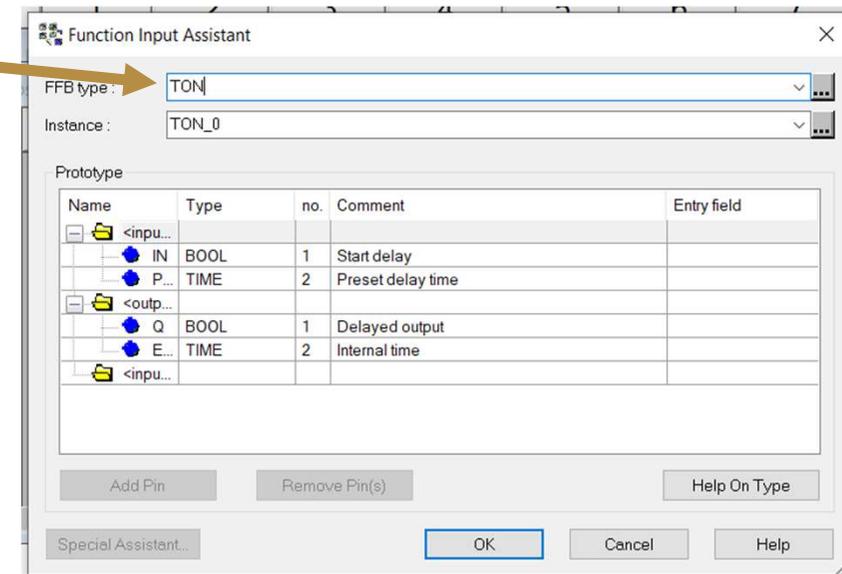
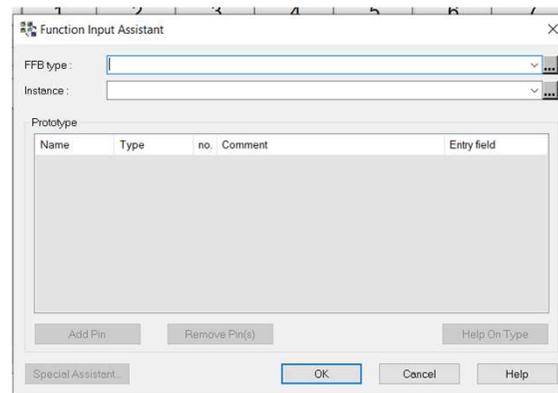
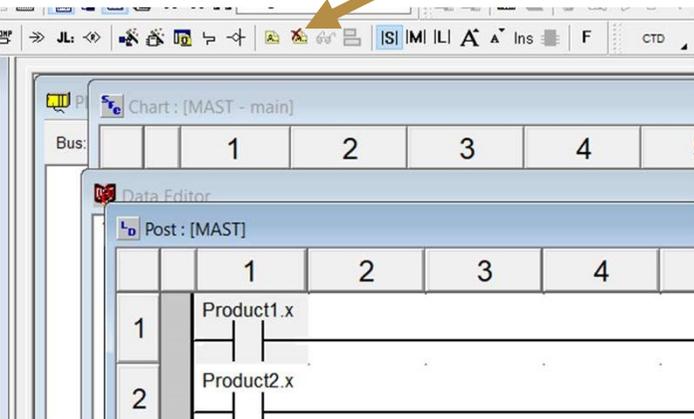
Implementation of the actions in ladder

- You need to configure M.VALUE as a function of Mixing.x (validation of the engine from the step Mixing)



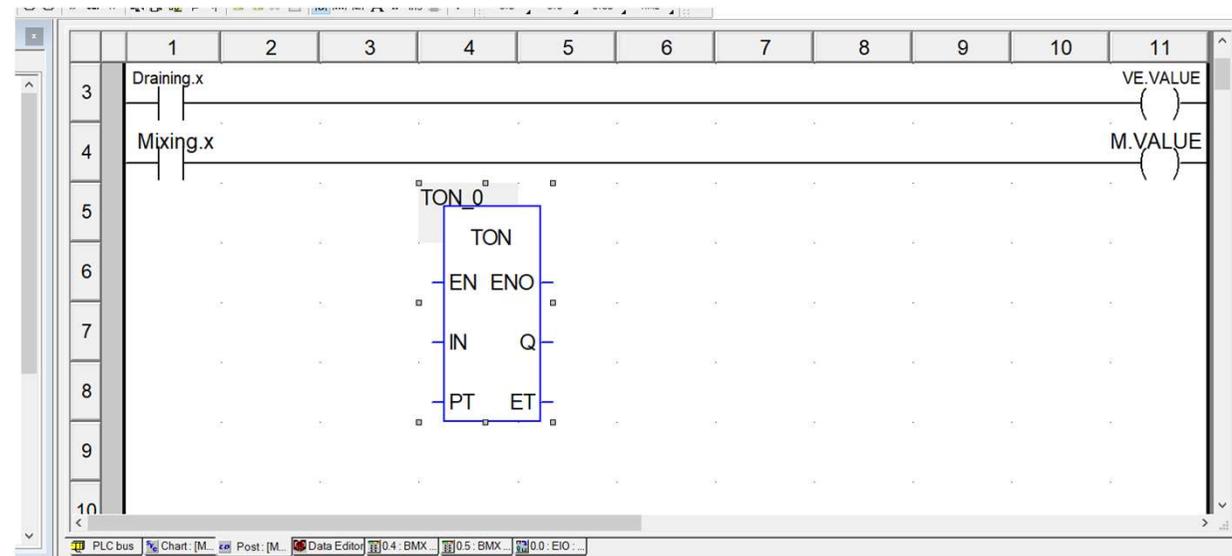
Implementation of the timing on the engine

- Each actuator activation should appear once only.
- For « Mixing », we need to activate both the engine (M1) and the timeout.
- Timeout are function blocks
- We will use the tool « FFB Input Assistant », you should click in the « Post » window (the ladder programme), to add the « TON » function on the ladder
- Timeout are objects with the type TON

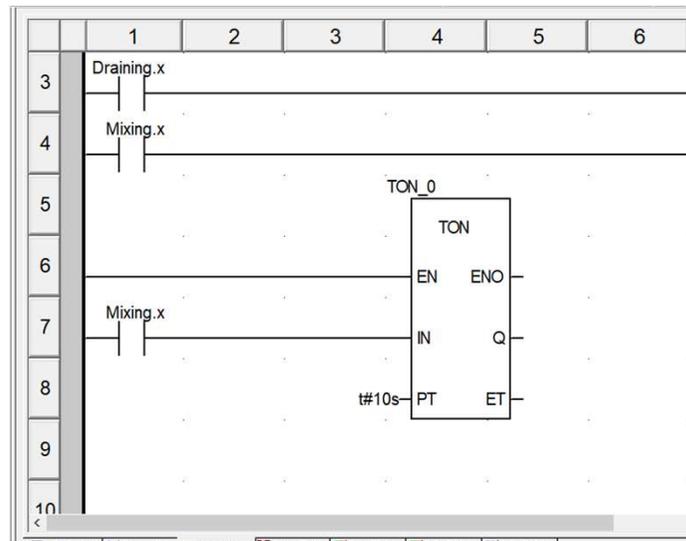
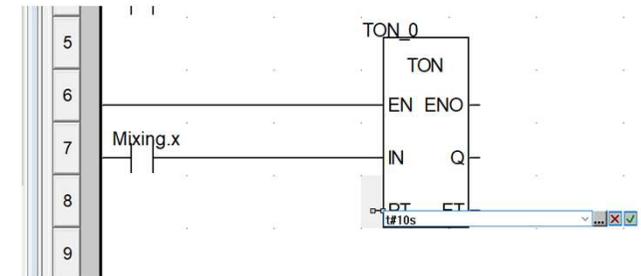
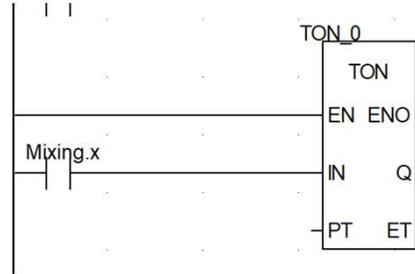
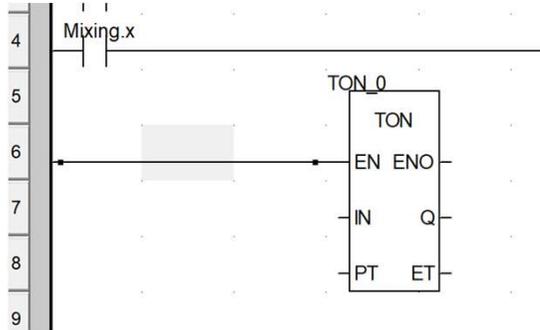


Implementation of the timing on the engine

- EN means Enable (should be used for the block to be executed)
- ENO will be activated at the end of the execution 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 active at the end of the timeout. It will be used for our SFC

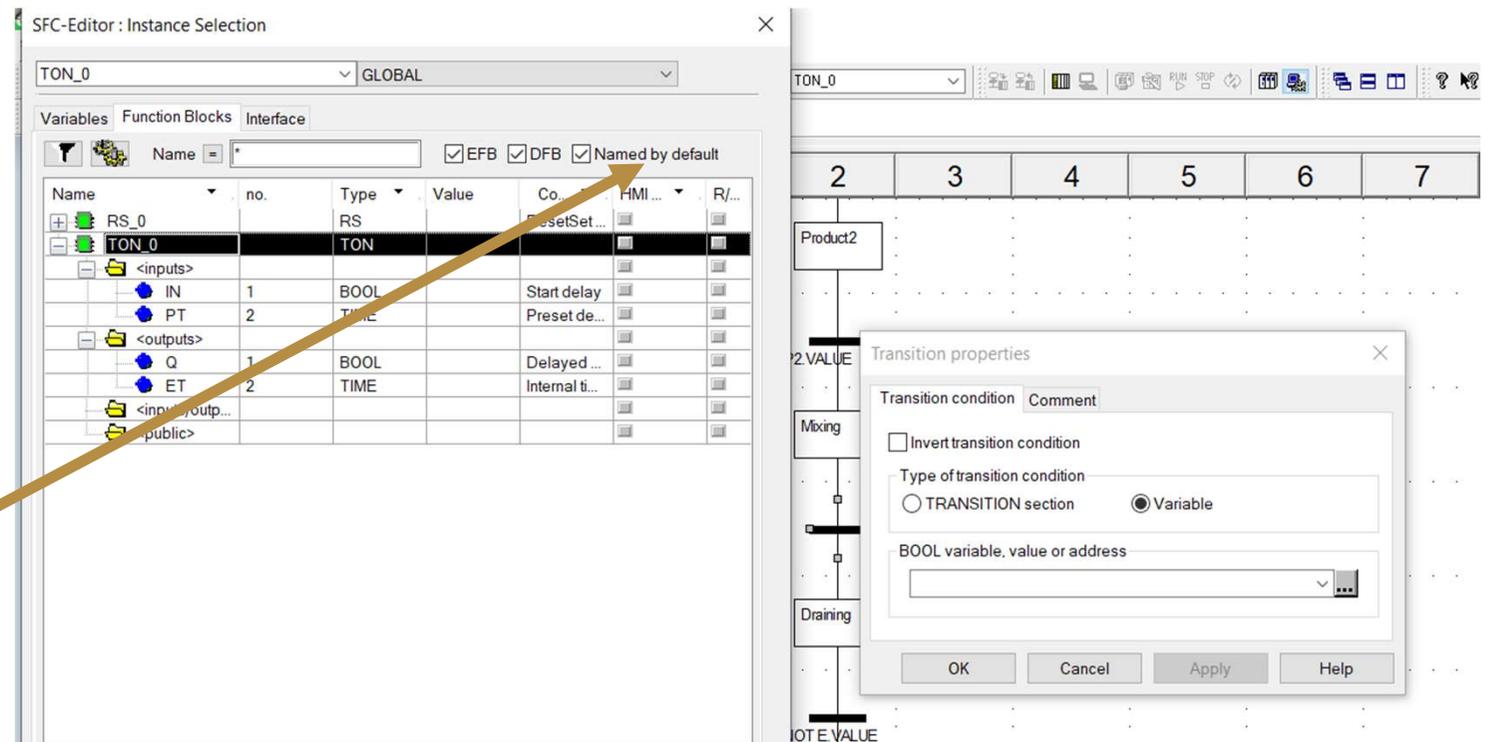


Implementation of the timing on the engine



Implementation of the timing on the engine

- In order to create the transition with the « timer » you need:
- To look for the variable you are interested in: TON_0.Q (which means the Q output of the TON_0 timer)
- To find it, you need to look for in « Function blocks », and also if necessary you need to click on « Named by default »



SFC-Editor : Instance Selection

TON_0 GLOBAL

Variables Function Blocks Interface

Name * EFB DFB Named by default

Name	no.	Type	Value	Co...	HMI ...	R/...
RS_0		RS		ResetSet...		
TON_0		TON				
<inputs>						
IN	1	BOOL		Start delay		
PT	2	TIME		Preset de...		
<outputs>						
Q	1	BOOL		Delayed ...		
ET	2	TIME		Internal ti...		
<input/output...>						
public>						

Transition properties

Transition condition Comment

Invert transition condition

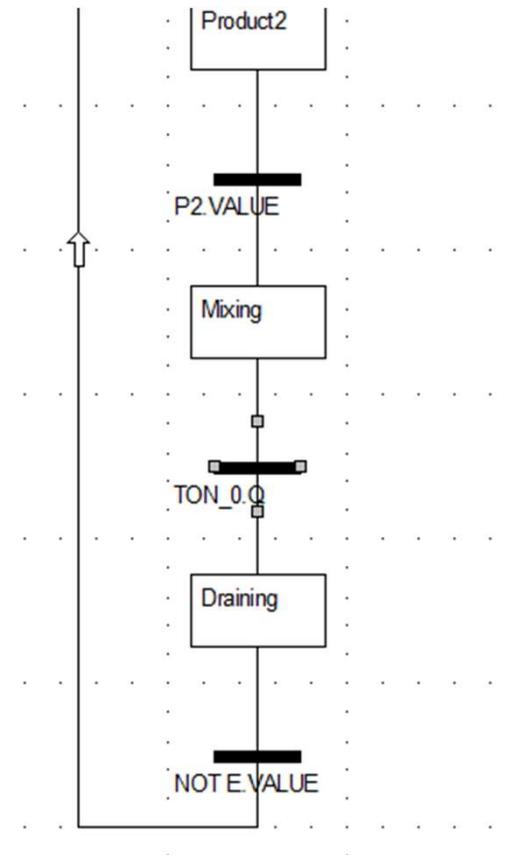
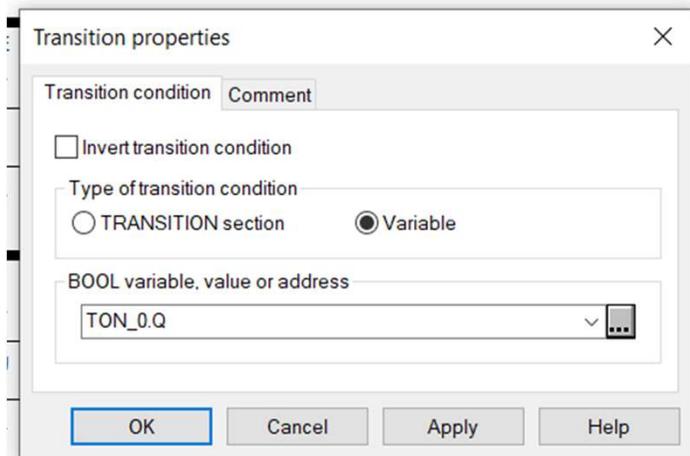
Type of transition condition

TRANSITION section Variable

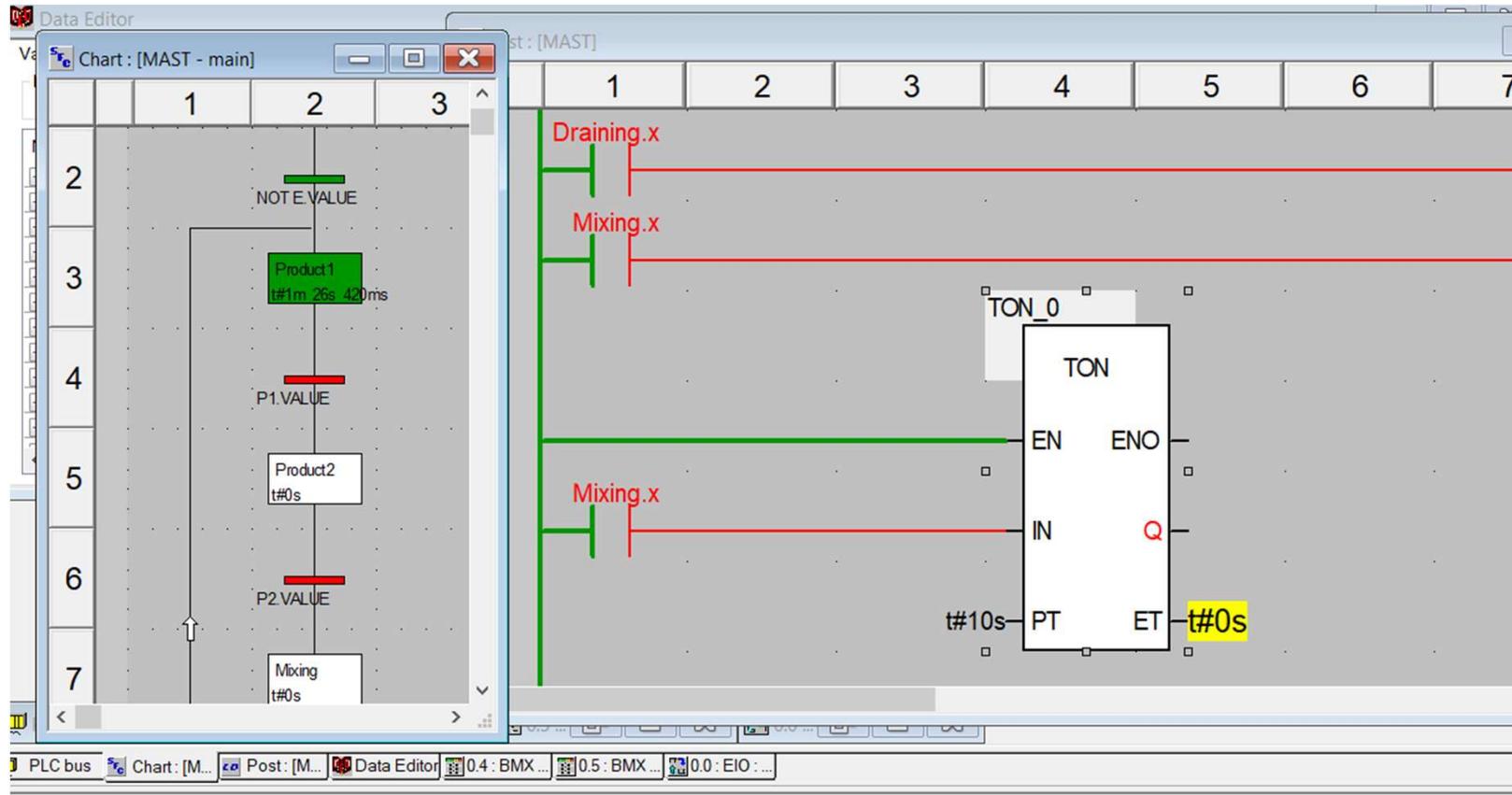
BOOL variable, value or address

OK Cancel Apply Help

Implementation of the timing on the engine



A running programme



Merci pour votre attention
Merci pour votre attention



សូមអរគុណចំពោះការ
យកចិត្តទុកដាក់របស់អ្នក
។

**Merci pour votre
attention**

**Thank you for
your attention**

