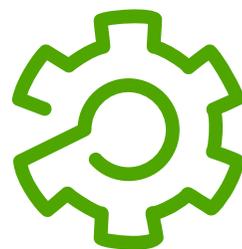




SoMachine

Solution logicielle pour votre architecture Machine
Atelier de découverte

Machine  Struxure



SoMachine

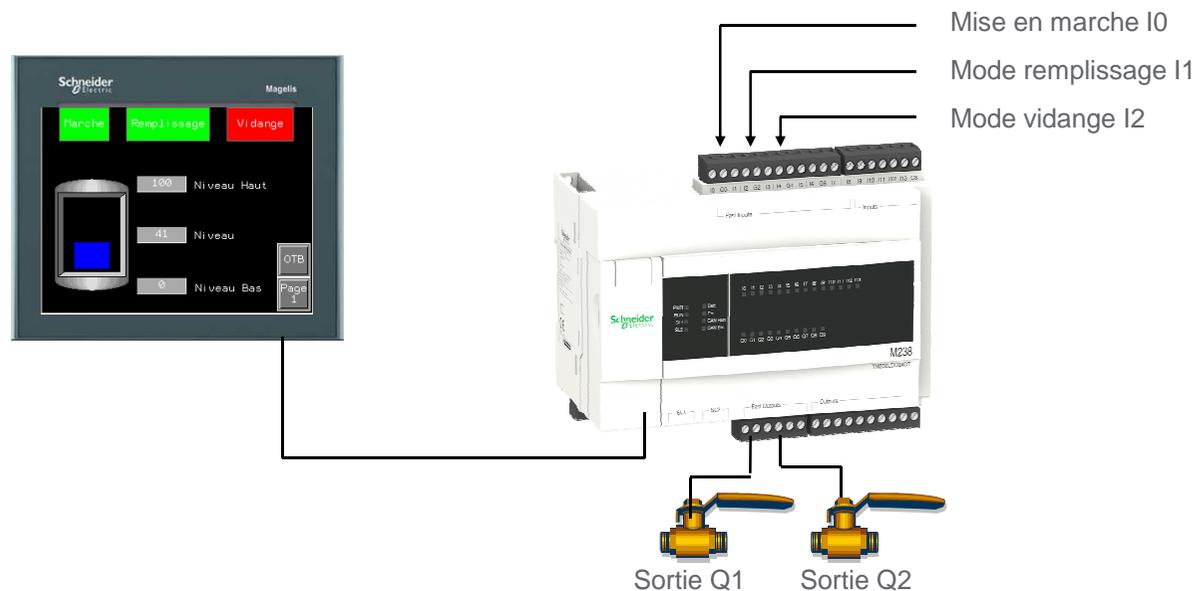
Schneider
 Electric

● Objectif de la manipulation

Le programme à réaliser va permettre de gérer le remplissage et la vidange d'une cuve en fonction d'un niveau haut et bas.

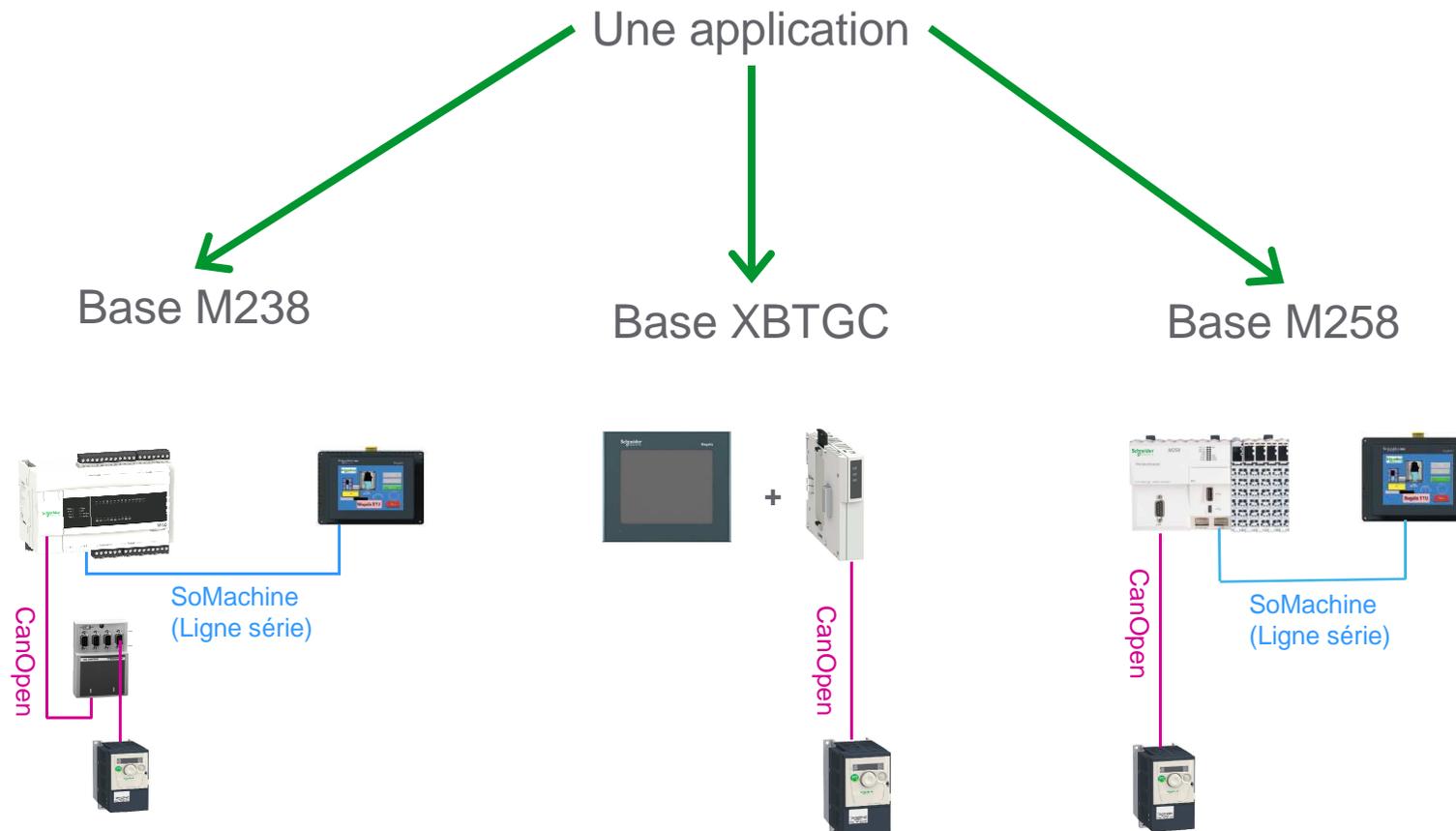
● Description du fonctionnement

- En fonction d'un niveau haut et bas de liquide dans la cuve et de l'action souhaitée (remplir la cuve ou vider la cuve), piloter les vannes permettant de réaliser cette action.
- Une fois que le niveau souhaité est atteint, désactiver l'ordre de marche des vannes



- Description des ateliers:

- 3 plateformes matérielles différentes pour une seule application



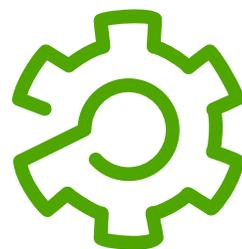


SoMachine

Découverte des valeurs du logiciel SoMachine

→ Aide à la conception

Machine  Struxure



SoMachine

Schneider
 Electric

Aide à la conception

● Lancement du SoMachine

Accès facile aux derniers projets

Aide à la création d'application

Aide à la mise en service
de la machine

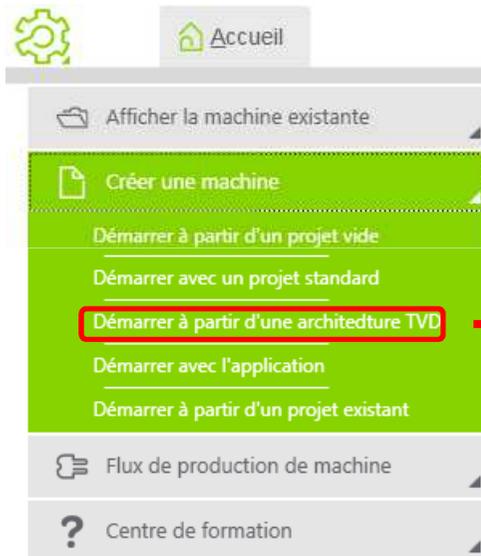
Un centre de formation

- Une aide en ligne
- Un manuel de démarrage
- Des vidéos explicatives
- Des exemples



Aide à la conception

• Une TVDA proche de votre machine



Recherche d'architecture TVD

Prévoyez-vous de n'utiliser qu'un seul coffret ou plusieurs coffrets distribués sur la machine ?
Sélectionnez

Utilisez-vous un bus de terrain sur votre machine ?
Sélectionnez

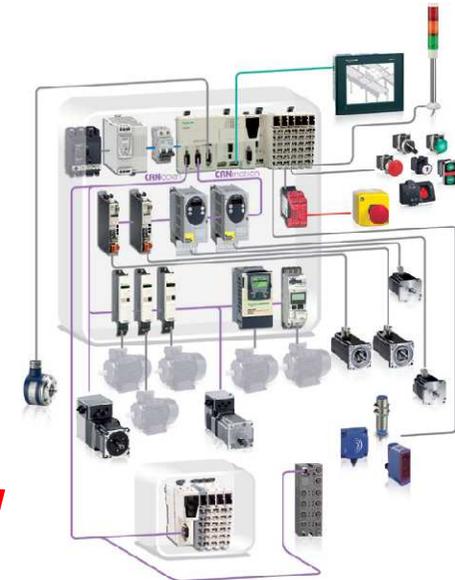
Quel type de bus de terrain utilisez-vous sur votre machine ?
Sélectionnez

Combien de commandes de moteur (comme des démarreurs progressifs, des variateurs de vitesse ou des servomoteurs) utilisez-vous ?
Sélectionnez

Quel est le nombre approximatif d'E/S utilisé par votre machine ?
Sélectionnez

Utilisez-vous le réseau EtherNet/IP ou Modbus/TCP pour superviser votre machine ?
Sélectionnez

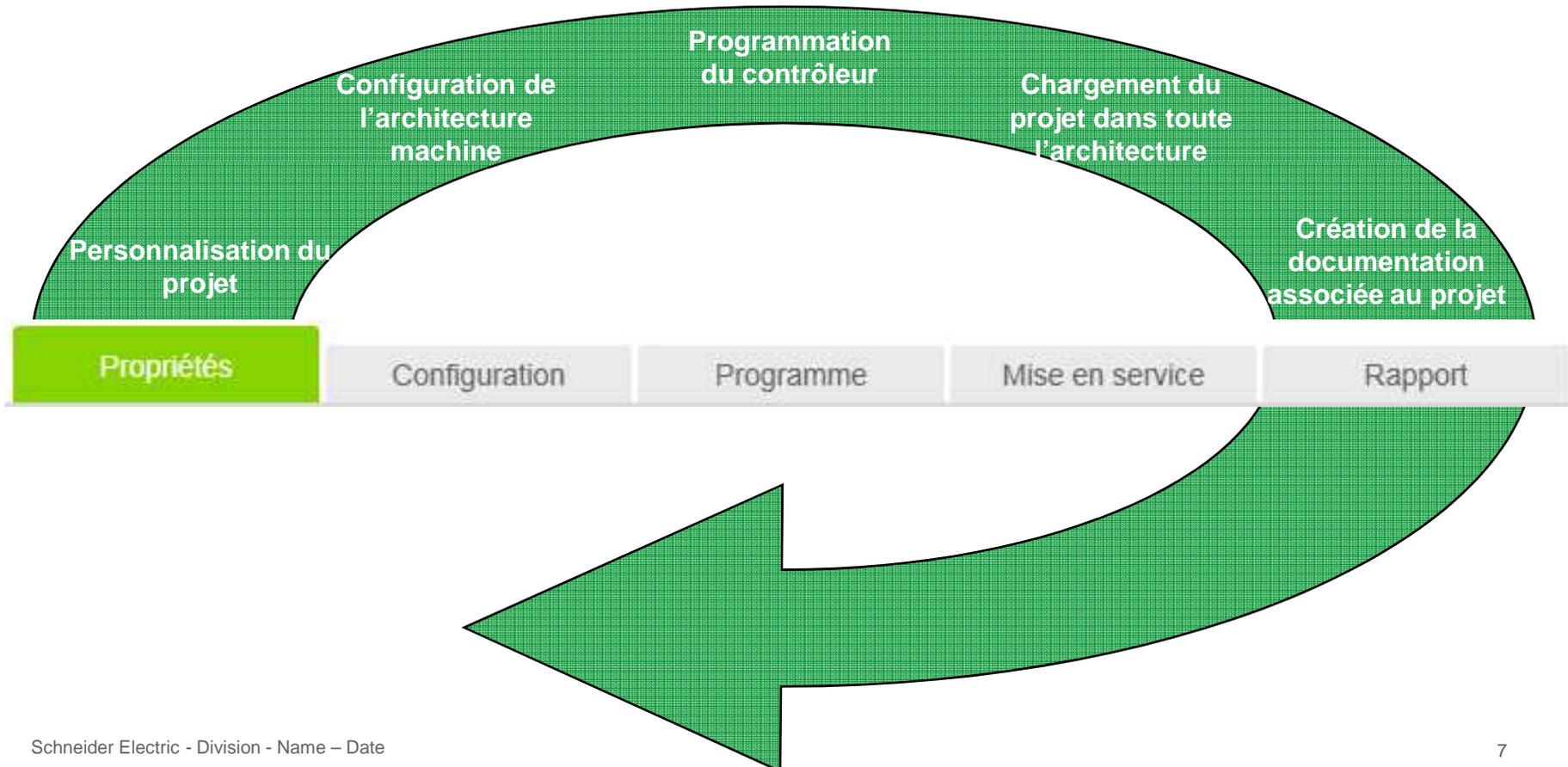
Projet recommandé



- Recommandation d'une TVD avec:
- Le code de programmation associé
 - Un document explicatif
 - Des notices de montage/câblage
 - Des préreglages

Aide à la conception

- Démarrage à partir d'un projet vide:
 - Navigation à travers les différentes étapes de conception de votre machine





SoMachine

Découverte des valeurs du logiciel SoMachine

→ Logiciel unique et intuitif

Machine  Structure



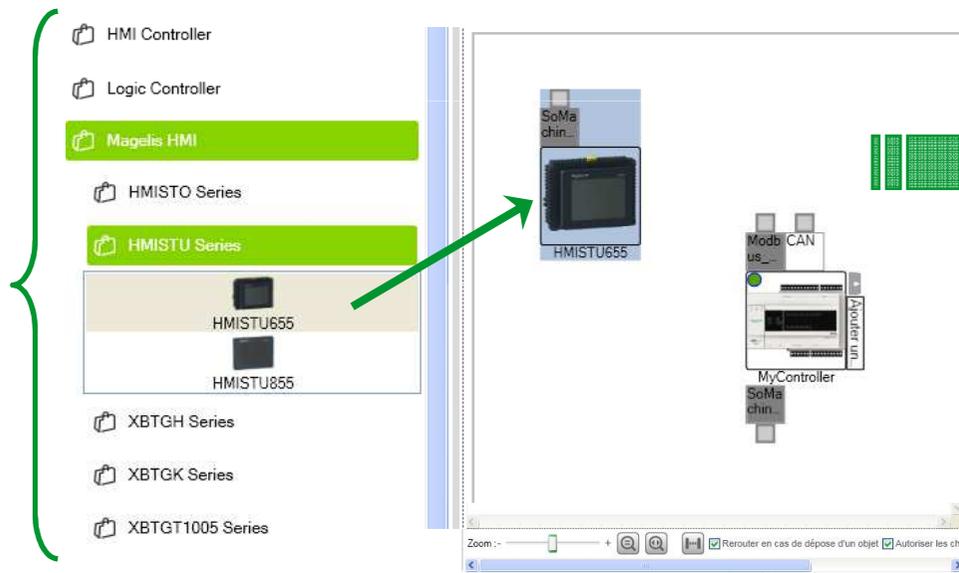
SoMachine

Schneider
 Electric

Logiciel unique et intuitif

- Exercice 1: Configuration de l'architecture Machine par « Drag&Drop »
 - ➔ Objectif: Configuration de l'architecture de gestion de la cuve

Ensemble des plateformes programmables avec le logiciel unique SoMachine

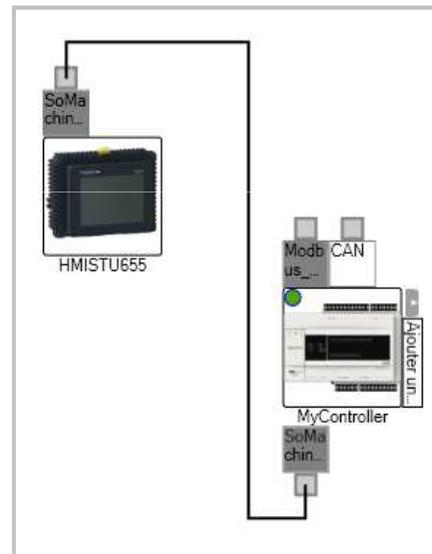


Lancement automatique de l'éditeur HMI intégré

Logiciel unique et intuitif

- Exercice 2: Configuration des réseaux

➔ Objectif: Configuration de la liaison HMI de la machine



- Exercice 3: Sauvegarde du projet unique et Retour à l'écran d'accueil

Logiciel unique et intuitif

- **Exercice 4: Ouverture du projet correspondant à sa plateforme matérielle**

Note: ce projet inclus une interface HMI développée pour le scénario de ce Clic'N Start

Nom du projet : Base exercice_xxxx_xxxx_SoMV3_Beta5

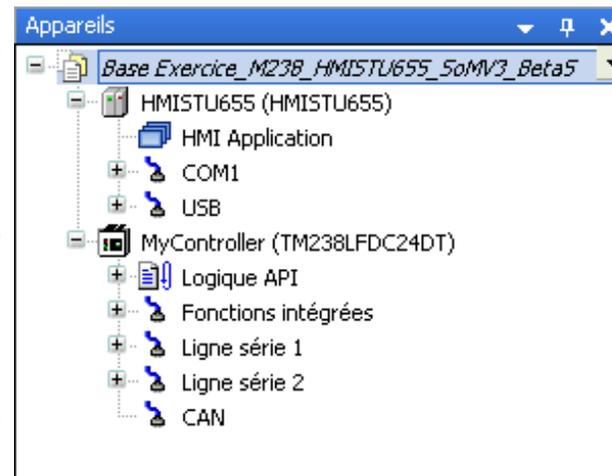
Exemple : Base Exercice_M238_HMISTU655_SoMV3_Beta5

- **Exercice 5: Découverte de la partie programmation**

➔ Objectif: Découverte de l'arborescence du projet

Partie HMI

Partie Contrôleur
 -Logique
 -E/S et fonctions intégrées
 -Communication



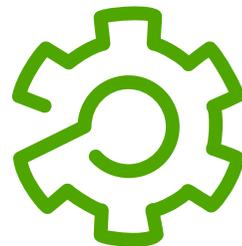


SoMachine

Découverte des valeurs du logiciel SoMachine

→ Simplicité de programmation

Machine  Struxure



SoMachine

Schneider
 Electric

Simplicité de programmation

- **Exercice 1 : Créer une section de programmation (POU) en Ladder**
 - ➔ Objectif : Définir le mode de marche des vannes selon l'opération voulue
 - **Etape 1 : Création du POU (Mode_de_marche) en Ladder**
 - **Etape 2 : Piloter une bobine « Etat_Machine » lorsque la machine est en marche**
 - **Etape 3 : Sur sélection du mode (Remplissage ou vidange) activer des bobines de sortie**
 - **Etape 4 : Ajouter le POU « Mode_de_marche » dans la tache MAST**
 - **Etape 5 : Compiler l'application** et vérifier le résultat de compilation dans la fenêtre « Messages ». Corriger les erreurs si nécessaire.

NOTE:

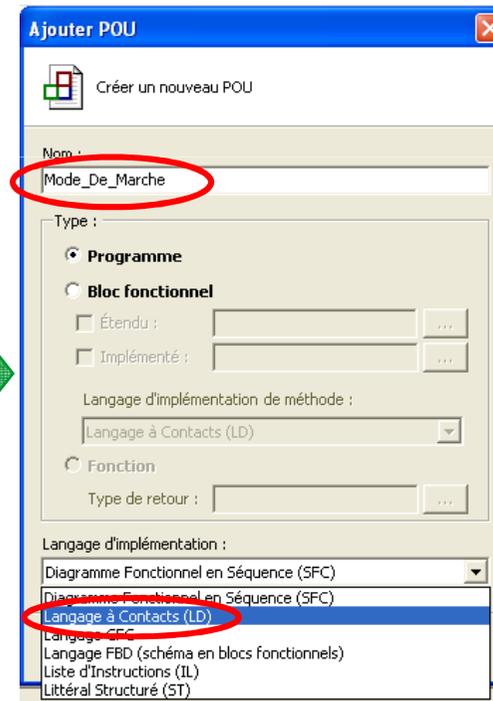
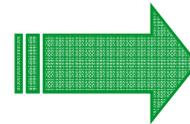
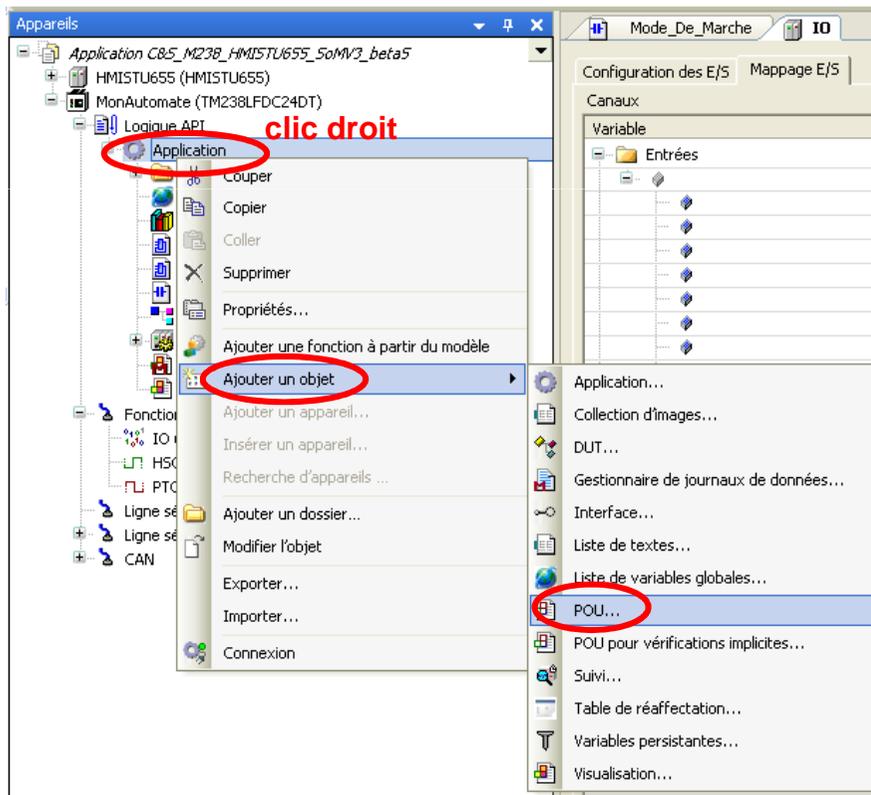
L'activation des bobines de sortie ne doit pas être possible si la machine n'est pas en marche.

Le remplissage et la vidange de la cuve ne peuvent pas se faire en même temps (Utiliser des contacts inversés et les variables globales « Moteur_remplissage » et « moteur_vidange »)

Simplicité de programmation

- Etape 1 : Création du POU 'Mode_De_Marche' en Ladder

➔ Solution



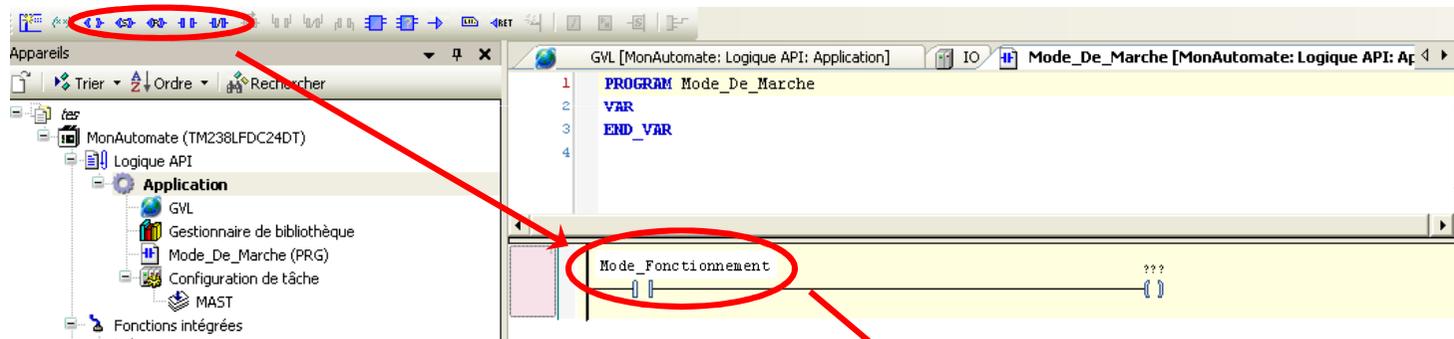
Assistant de création de programme ou de bloc fonction

Simplicité de programmation

• Etape 2 : Piloter une bobine Etat_Machine lorsque la machine est en marche

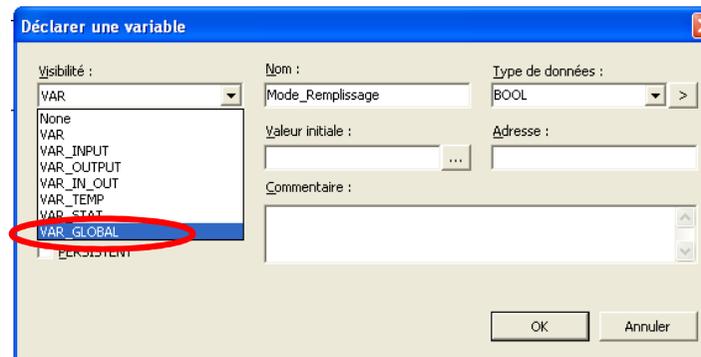
- Nom des variables : Mode_Fonctionnement, Etat_machine
- Variables de type BOOL qui doivent être déclarées en GVL (Variable Globales) car partagées dans l'application

→ Solution



Assistant de déclaration de variables:

- Nom de la variable
- Visibilité
- Valeur initiale
- Adresse
- Type de données



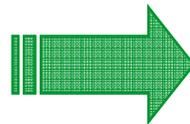
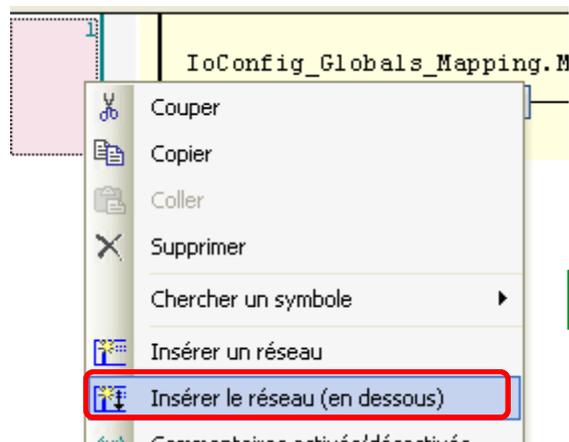
Simplicité de programmation

• Etape 3: Sur sélection du mode, activer des bobines de sortie

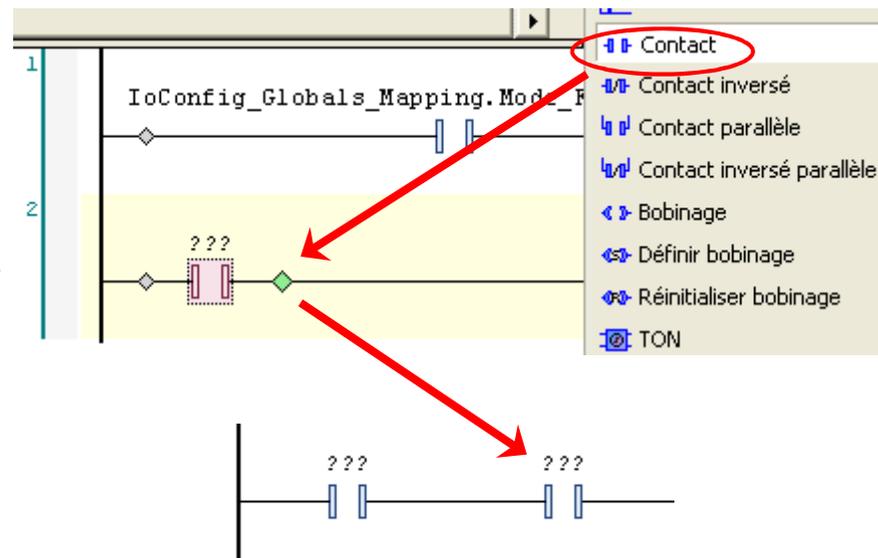
- Sélection du mode :
 - Nom des variables : Mode_Remplissage et Mode_vidange
 - Variables de type BOOL qui doivent être déclarées en GVL (Variable Globales)
- Bobines de sortie
 - Nom des variables : Commande_remplissage et Commande_vidange
 - Variables de type BOOL qui doivent être déclarées en GVL (Variable Globales)

→ Solution

Ajouter un Réseau



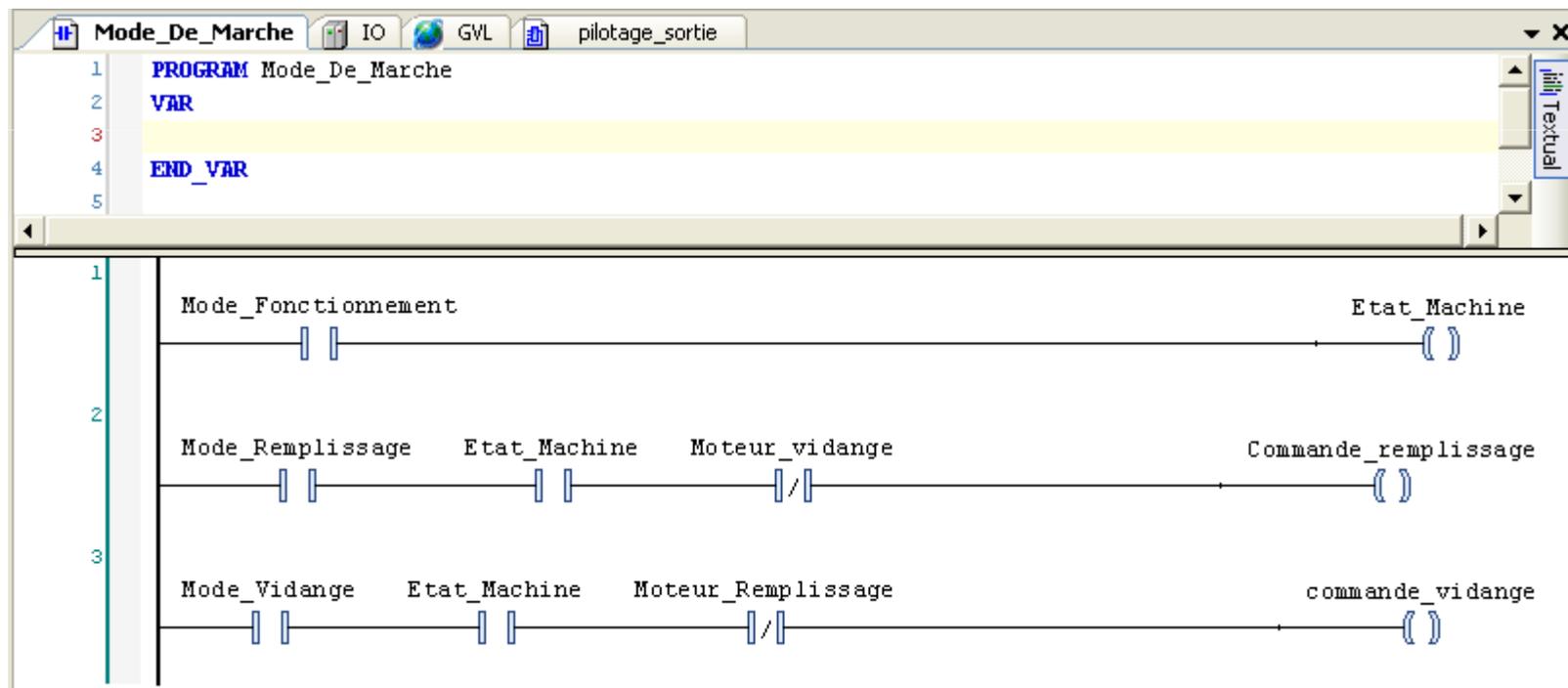
Ajouter un contact (drag & drop)



Simplicité de programmation

- Etape 3 : Sur sélection du mode, activer des bobines de sortie

→ Résultat



Simplicité de programmation

- Etape 4 : Ajouter le POU « Mode_De_Marche » dans la tâche MAST

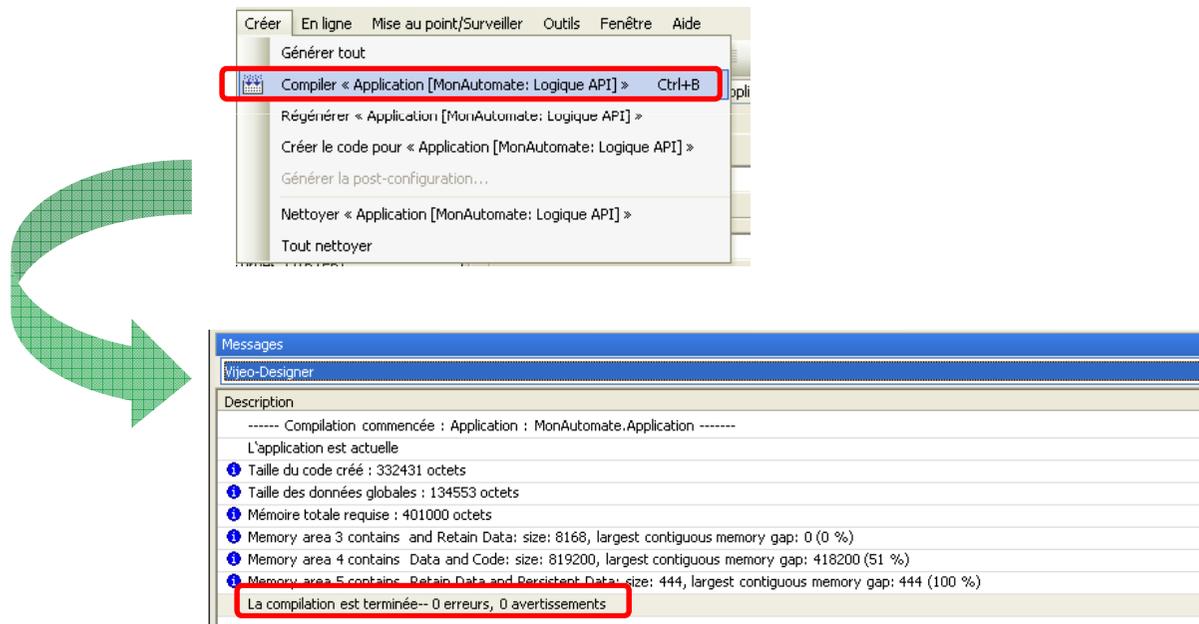
→ Résultat

The screenshot displays the SoMachine software interface. On the left, the project tree shows the hierarchy: MonAutomate (TM238LFDC24DT) > Logique API > Application > MAST. The 'MAST' task is circled in red. In the center, the 'Configuration' window for the 'MAST' task is visible, showing settings like 'Type: Cyclique', 'Intervalle: 20 ms', and 'Horloge de surveillance: Activer'. Below this, a 'POU' (Program Object User) list is shown with 'Ajouter l'appel' circled in red. On the right, the 'Aide à la saisie' (Help to enter) dialog box is open, showing a list of programs. 'Mode_De_Marche' is selected and circled in red. The dialog also shows 'PROGRAM Mode_De_Marche' in the documentation field.

Simplicité de programmation

- **Etape 5 : Compiler l'application** et vérifier le résultat de compilation dans la fenêtre « Messages »

➔ Résultat

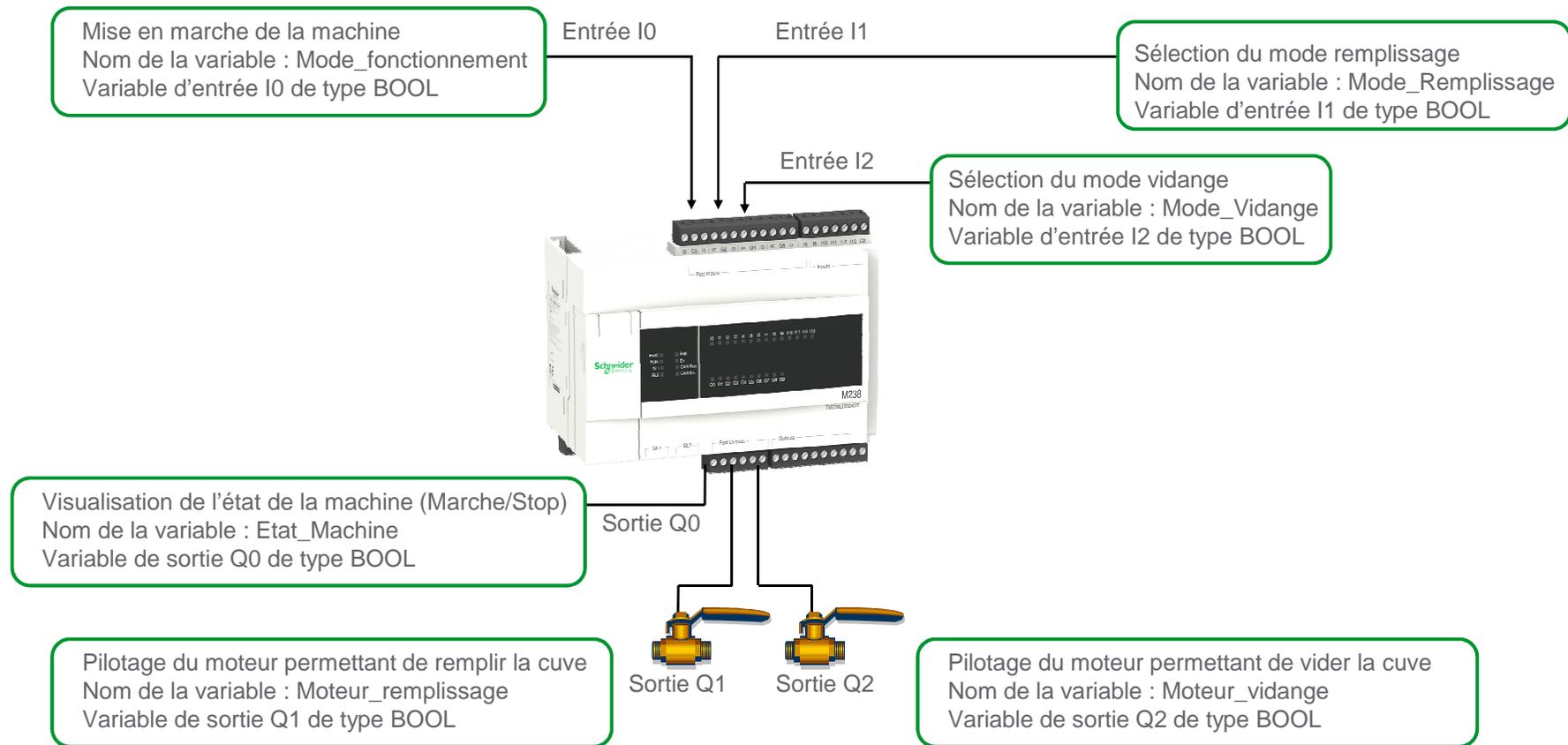


The image shows two screenshots from the SoMachine software. The top screenshot displays the 'Créer' (Create) menu with the 'Compiler « Application [MonAutomate: Logique API] »' option highlighted by a red box. The bottom screenshot shows the 'Messages' window with the text 'La compilation est terminée-- 0 erreurs, 0 avertissements' (Compilation is finished-- 0 errors, 0 warnings) highlighted by a red box. A large green arrow points from the menu item to the Messages window.

Pour afficher la fenêtre « Message », sélectionner le menu « Affichage » puis « Messages »

Simplicité de programmation

- **Exercice 2 : Configuration des entrées/sorties physiques du contrôleur**
 - ➔ Objectif : Affecter des noms de variables aux entrées/sorties physiques



Simplicité de programmation

- Exercice 2 : Configuration des entrées/sorties physiques du contrôleur

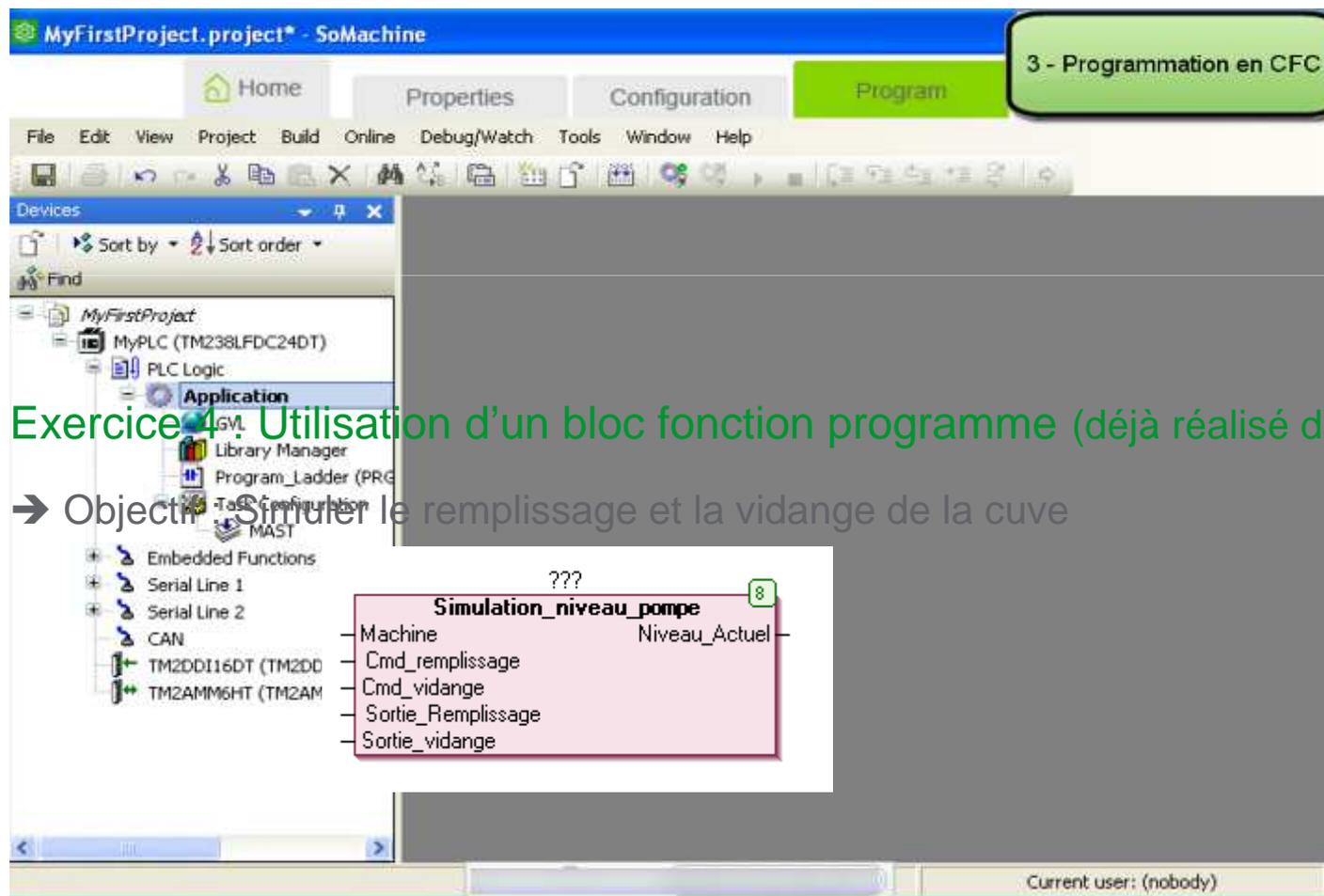
➔ Solution: Ouvrir la fenêtre de paramétrage des entrées/sorties et leur affecter des noms de variables.

The screenshot shows the SoMachine software interface. On the left, a tree view shows the project structure with 'IO (IO)' highlighted under 'Fonctions intégrées'. The main window displays the 'Configuration des E/S' (IO Mapping) dialog, which is titled 'Mappage E/S'. It contains a table with columns for 'Variable', 'Mappage', 'Canal', 'Adresse', and 'Type'. The 'Entrées' (Inputs) section is expanded, showing a list of input channels (I0 to I13) with their corresponding addresses and types. A red box highlights the 'Mode Remplissage' variable in the 'Entrées' list. To the right, a 'Aide à la saisie' (Input Assistance) dialog is open, showing a list of available variables. The 'Mode_Fonctionnement' variable is highlighted in red in this dialog, and a red arrow points from the 'Mode Remplissage' variable in the main table to this dialog, indicating the selection process.

Variable	Mappage	Canal	Adresse	Type
Entrées				
		IW0	%IW0	W
		I0	%IX0.0	BO
		I1	%IX0.1	BO
		I2	%IX0.2	BO
		I3	%IX0.3	BO
		I4	%IX0.4	BO
		I5	%IX0.5	BO
		I6	%IX0.6	BO
		I7	%IX0.7	BO
		I8	%IX1.0	BO
		I9	%IX1.1	BO
		I10	%IX1.2	BO
		I11	%IX1.3	BO
		I12	%IX1.4	BO
		I13	%IX1.5	BO
Sorties				
		QW0	%QW0	W
		Q0	%QX0.0	BO
		Q1	%QX0.1	BO
		Q2	%QX0.2	BO
		Q3	%QX0.3	BO
		Q4	%QX0.4	BO
		Q5	%QX0.5	BO
		Q6	%QX0.6	BO
		Q7	%QX0.7	BO
		Q8	%QX1.0	BO
		Q9	%QX1.1	BO

Simplicité de programmation

- Exercice 3 : Créer un Pou en CFC (déjà réalisé dans l'application)



- Exercice 4 : Utilisation d'un bloc fonction programme (déjà réalisé dans l'application)

→ Objectif : Simuler le remplissage et la vidange de la cuve

Simplicité de programmation

● Exercice 3 : Créer un Pou en CFC

→ Objectif : Piloter les vannes en fonction de:

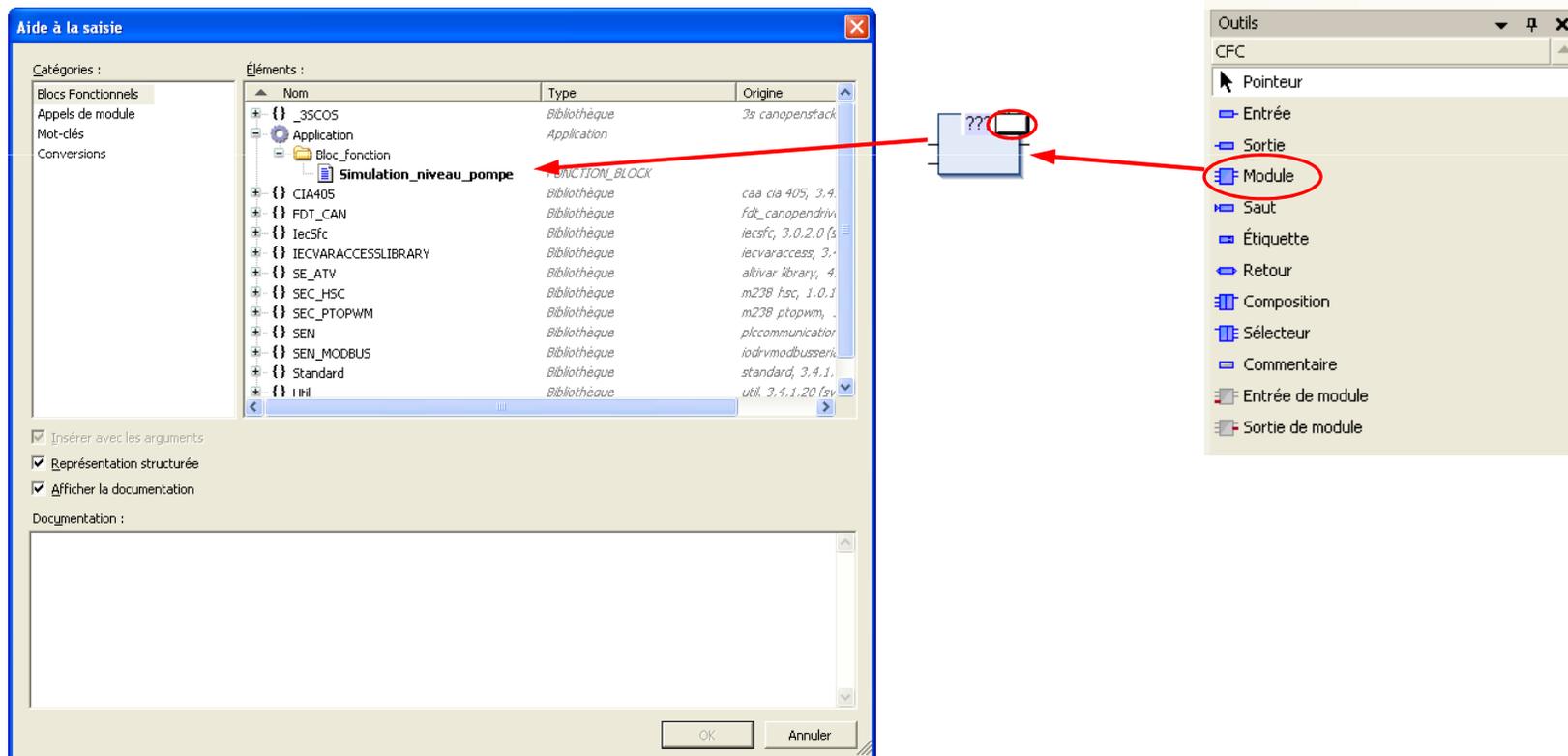
- La commande souhaitée (remplir la cuve, vider la cuve)
- Le niveau actuel de la cuve
- Le niveau haut et le niveau bas de la cuve

● Exemple: Je souhaite remplir la cuve au niveau haut qui est de 100l. Actuellement le niveau dans la cuve est de 50l. Je pilote la vanne permettant de remplir la cuve jusqu'à ce que le niveau haut soit atteint.

- Etape 1 : Création d'un POU CFC (Nom du Pou: pilotage_sortie)
- Etape 2 : Réalisation du programme à l'aide de fonctions de la bibliothèque
- Etape 3 : Affecter les variables aux blocs fonction

Simplicité de programmation

- Réalisation du programme à l'aide de fonctions de la bibliothèque:
 - ➔ Solution : Utilisation de la barre d'outil CFC



Simplicité de programmation

- Affecter les variables aux blocs fonction

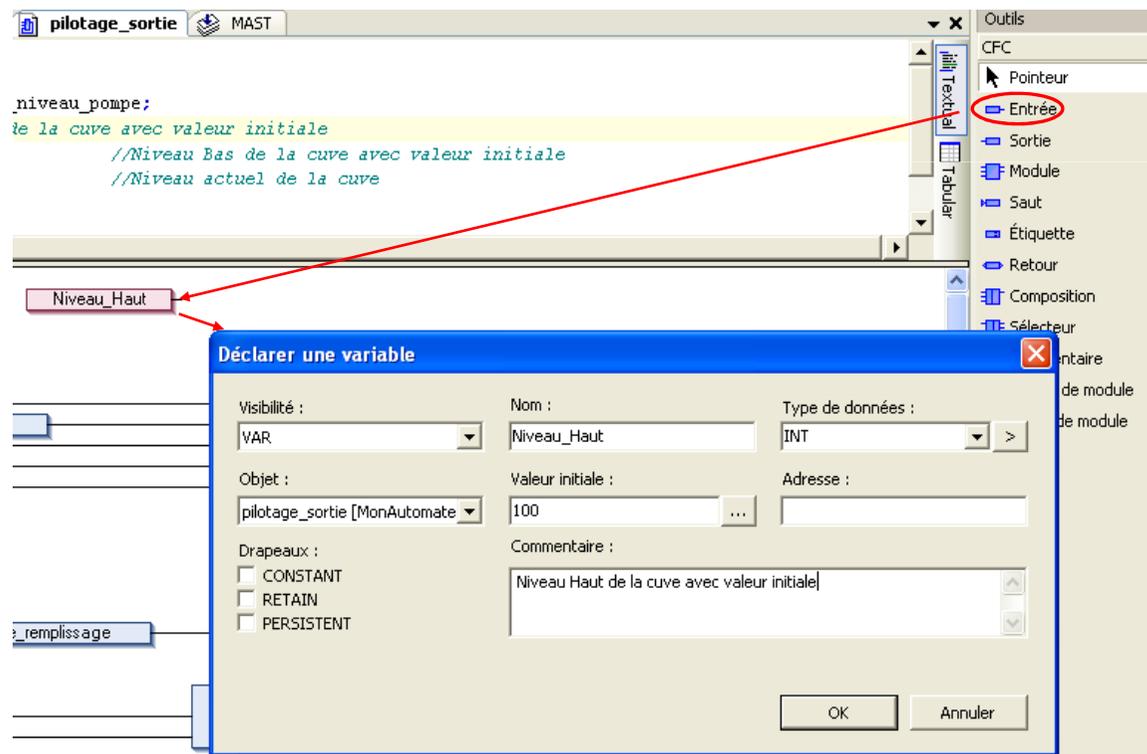
→ Solution : Utilisation et déclaration d'une entrée (avec valeur initiale)

3 variables à créer:

Nom de la variable : Niveau_Haut
Type de variable : INT
Valeur initiale : 100

Nom de la variable : Niveau_Bas
Type de variable : INT
Valeur initiale : 0

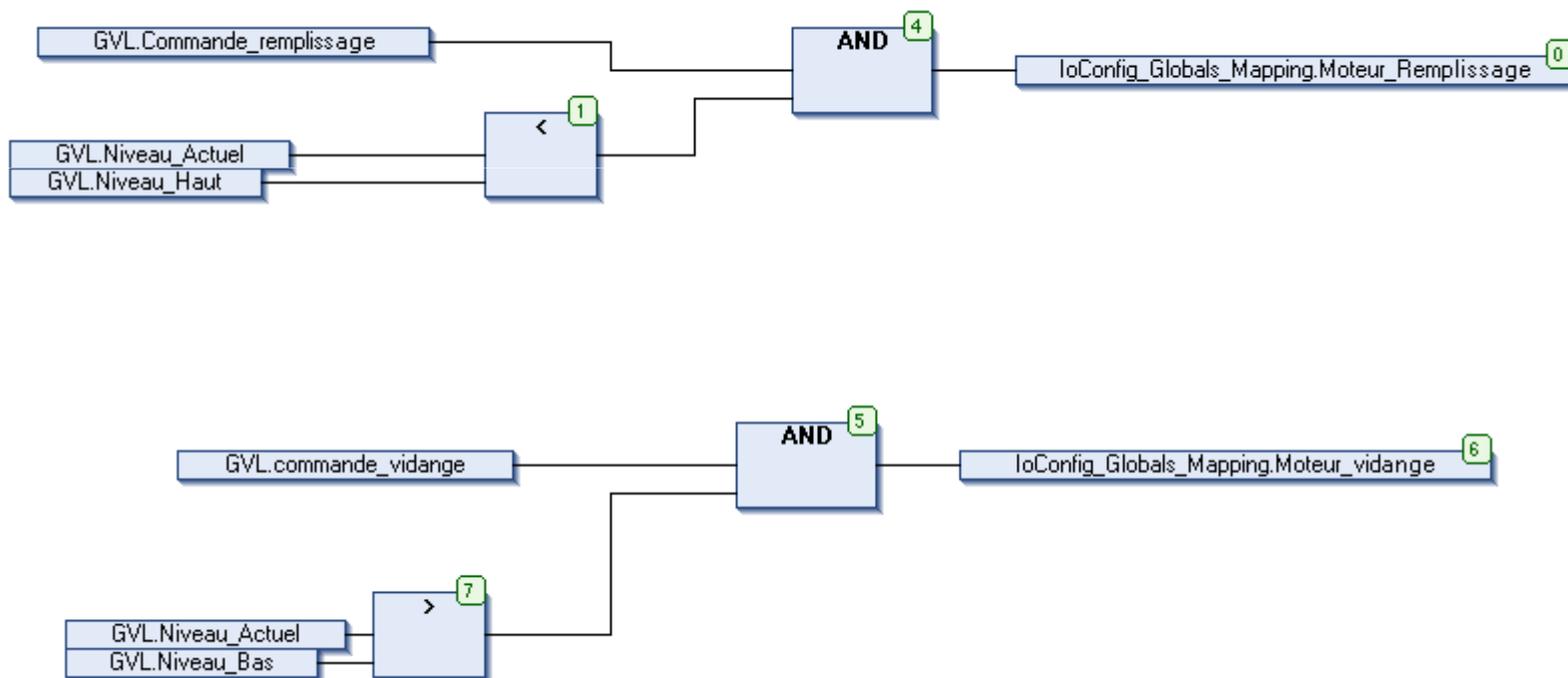
Nom de la variable : Niveau_Actuel
Type de variable : INT
Valeur initiale : 0



Simplicité de programmation

- Exercice 3 : Créer un Pou en CFC pour piloter les vannes

→ Solution



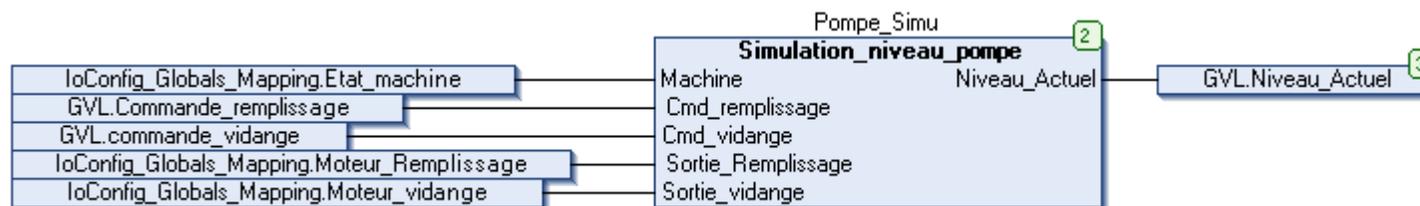
Simplicité de programmation

- **Exercice 4 : Utilisation d'un bloc fonction programme**

➔ Objectif : Placer dans la section CFC, le bloc fonction 'simulation_niveau_pompe'

Note : Le bloc fonction est déjà crée dans l'application fournie.

- **Etape 1 :** A l'aide de la barre d'outil CFC, sélectionner un module et aller chercher dans la bibliothèque le bloc fonction « simulation_niveau_pompe »
- **Etape 2 :** Donner au bloc fonction le nom d'instance « Pompe_Simu »
- **Etape 3 :** Affecter les Entrée/Sorties au bloc fonction

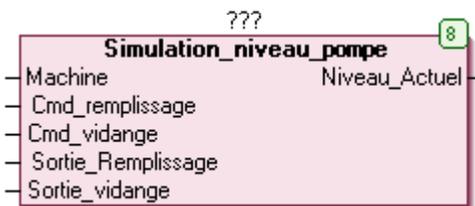


- **Etape 4 :** Ajouter le POU dans la tache MAST puis compiler

Simplicité de programmation

- Explication sur bloc fonction

- Le bloc fonction permet de simuler le remplissage et la vidange de la cuve



```

M3LT Simulation_niveau_pompe [MonAutomate: Logique API: Application] Modbus_IOScanner pilotag
1  FUNCTION_BLOCK Simulation_niveau_pompe
2  VAR_INPUT
3      Machine: BOOL;
4      Cmd_remplissage: BOOL;
5      Cmd_vidange: BOOL;
6      Sortie_Remplissage: BOOL;
7      Sortie_vidange: BOOL;
8  END_VAR
9  VAR_OUTPUT
10     Niveau_Actuel: INT;
11 END_VAR
12 VAR
13     Time_ls: BLINK;
14     tempo: BOOL;
15     test: BOOL;
16     Front_l: R_TRIG;
17     F_tempo: BOOL;
18 END_VAR
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Simplicité de programmation

- Exercice 5 : Simulation de l'application

→ Objectif : Utiliser le simulateur pour simuler le mode de marche de l'application.

- Etape 1 : Passer en mode Simulation
- Etape 2 : Transférer l'application
- Etape 3 : Ouvrir un écran « surveiller » et y ajouter les variables (Mode Fonctionnement, Mode Remplissage, Mode_vidange)
- Etape 4 : Commander les entrées dans la fenêtre « surveiller 1 » et vérifier le comportement dans le POU « mode_de_marche »

Simplicité de programmation

- Etape 1 : Passer en mode Simulation

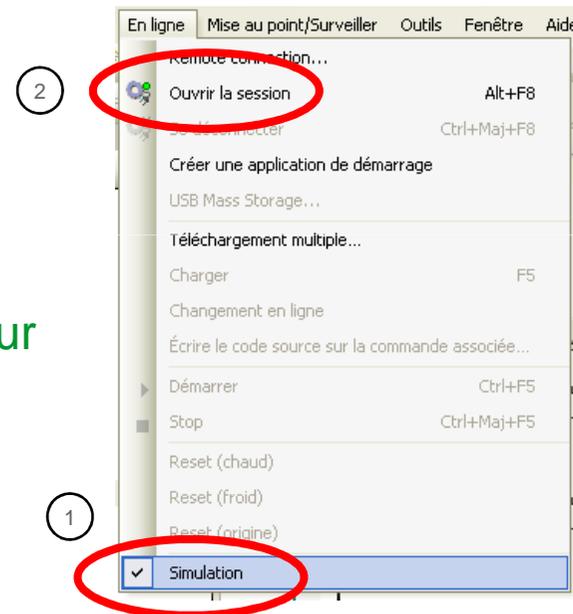
- Solution:

- ① Menu « En Ligne », « Simulation »

- Etape 2 : Transférer l'application dans le simulateur

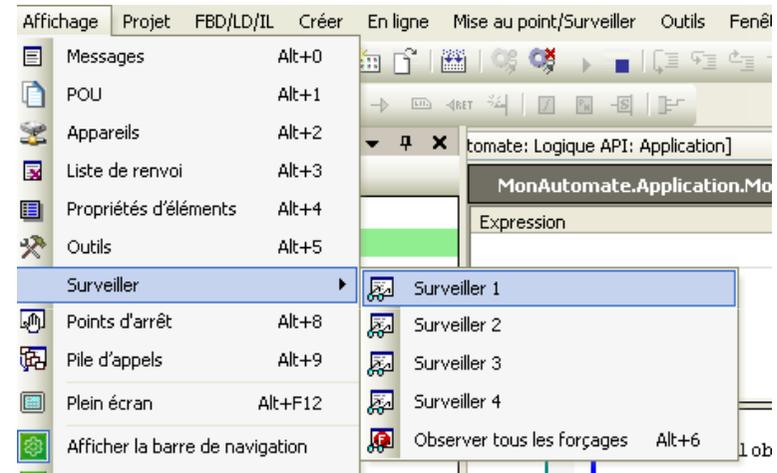
- Solution:

- ② Menu « En Ligne », « Ouvrir la session »



Simplicité de programmation

- Etape 3 : Ouvrir un écran « surveiller »
→ Solution



- Ajouter les variables (Mode Fonctionnement, Mode Remplissage, Mode_vidange)

Surveiller 1				
Expression	Comment	Type de données	Valeur	Valeur préparée
MonAutomate.Application.Mode_Fonctionnement		BOOL	FALSE	
MonAutomate.Application.Mode_replissage		BOOL	FALSE	
MonAutomate.Application.Mode_vidange		BOOL	FALSE	

Simplicité de programmation

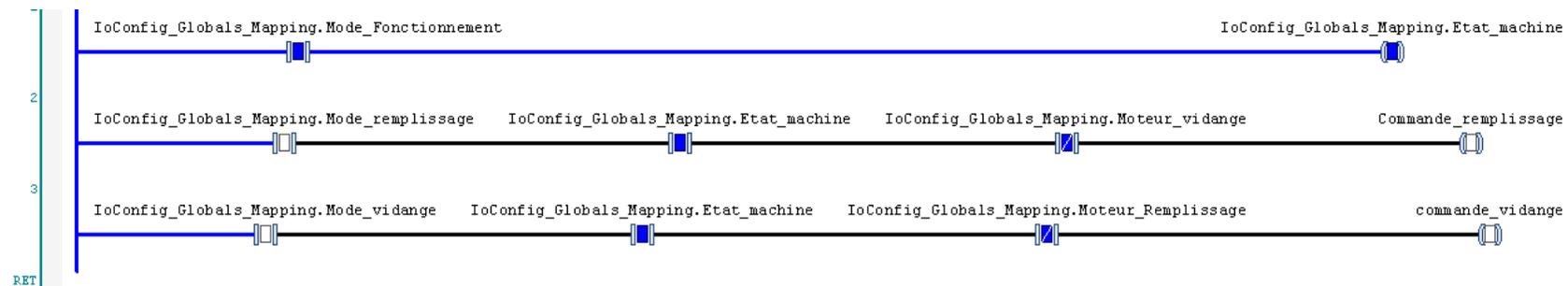
- Etape 4 : Commander les entrées

➔ Solution: Commander les entrées dans la fenêtre « surveiller 1 » et vérifier le comportement dans le POU « mode_de_marche »

Surveiller 1				
Expression	Comment	Type de données	Valeur	Valeur préparée
MonAutomate.Application.Mode_Fonctionnement		BOOL	TRUE	
MonAutomate.Application.Mode_replissage		BOOL	FALSE	TRUE
MonAutomate.Application.Mode_vidange		BOOL	FALSE	

Pour écrire les valeurs, aller dans 'Mise au point/surveiller' puis 'Ecrire les valeurs'

Note: raccourci Ctrl + F7 pour modifier l'état de la variable



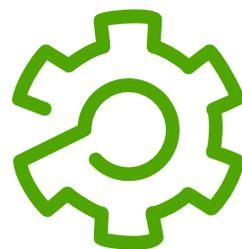


SoMachine

Découverte des valeurs du logiciel SoMachine

→ Transparence des échanges

Machine  Struxure



SoMachine

Schneider
 Electric

Transparence des échanges

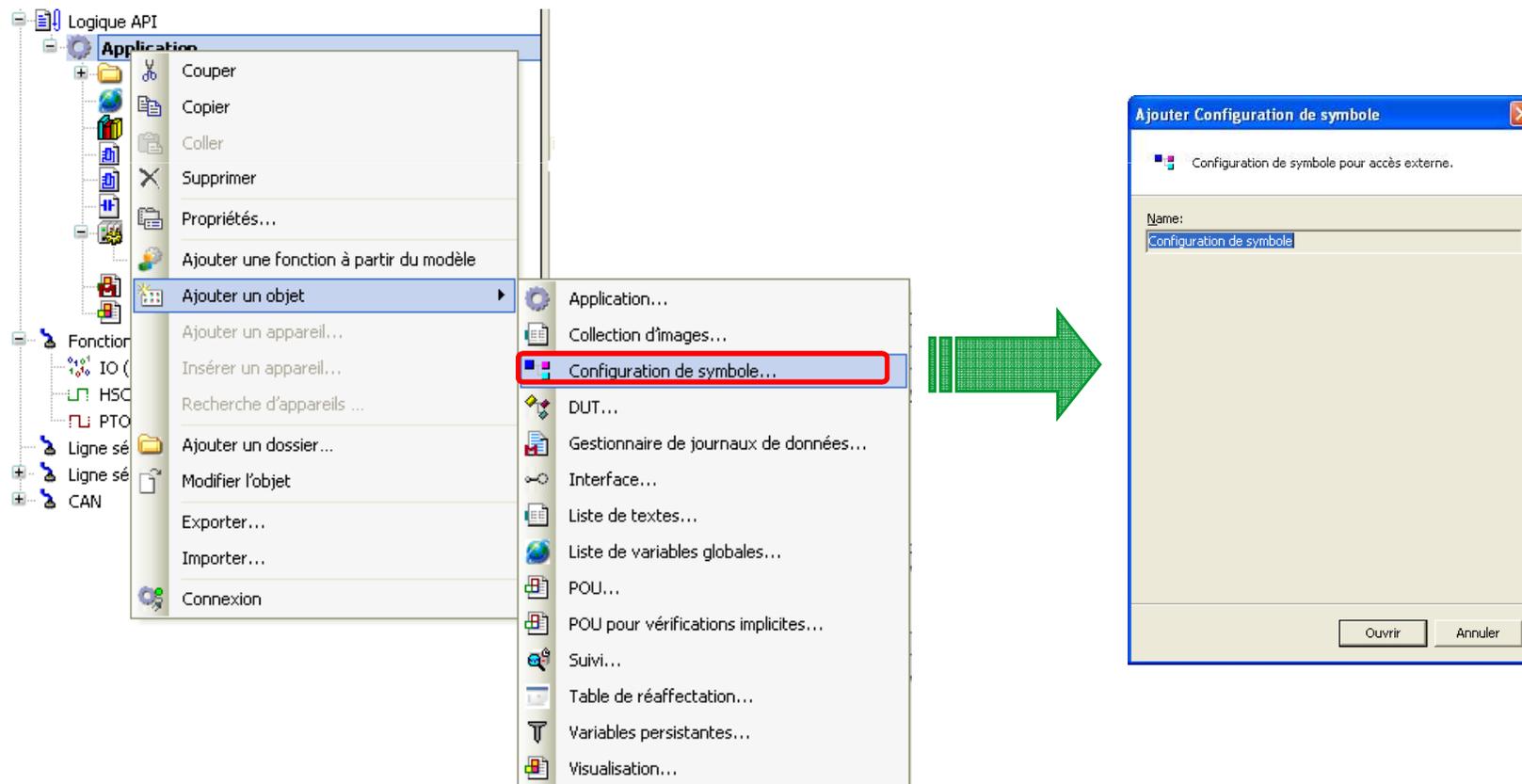
- **Exercice 1: Echange des variables avec l'afficheur**

➔ Objectif : Exploiter les variables du programme pour visualiser le volume de liquide dans la cuve

- **Etape 1:** Publier les variables à partager avec l'afficheur HMI
 - Ajouter un objet « configuration de symbole » et partager les variables désirées
- **Etape 2:** Ouvrir l'écran n°2 de l'application Vijeo Designer et affecter les variables suivantes aux objets.
 - Variables : Etat_Machine, Moteur_remplissage, Moteur_Vidange, Niveau_haut, Niveau_bas, Niveau_actuel
- **Etape 3:** Coté Vijeo Designer, configurer l'adresse du contrôleur

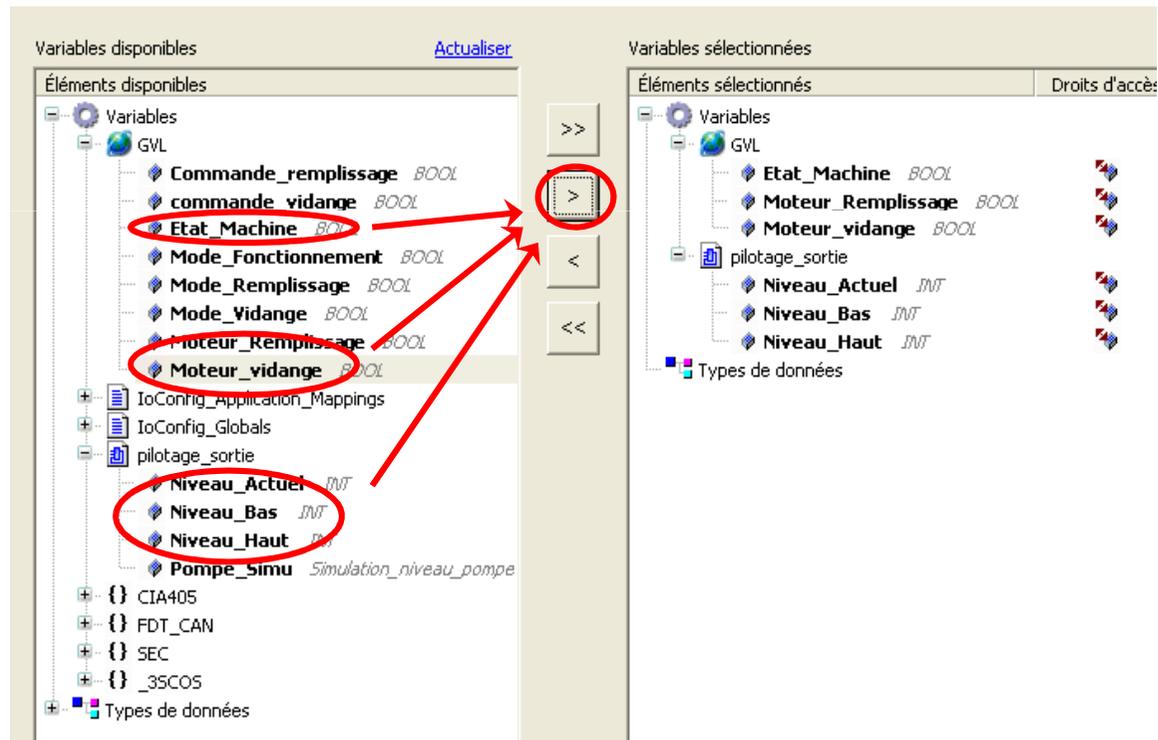
Transparence des échanges

- Etape 1: Publier les variables à partager
 - Solution: Ouverture du configurateur de symboles



Transparence des échanges

- Etape 1: Publier les variables à partager
 - ➔ Solution: Publication des variables



Variables du contrôleur

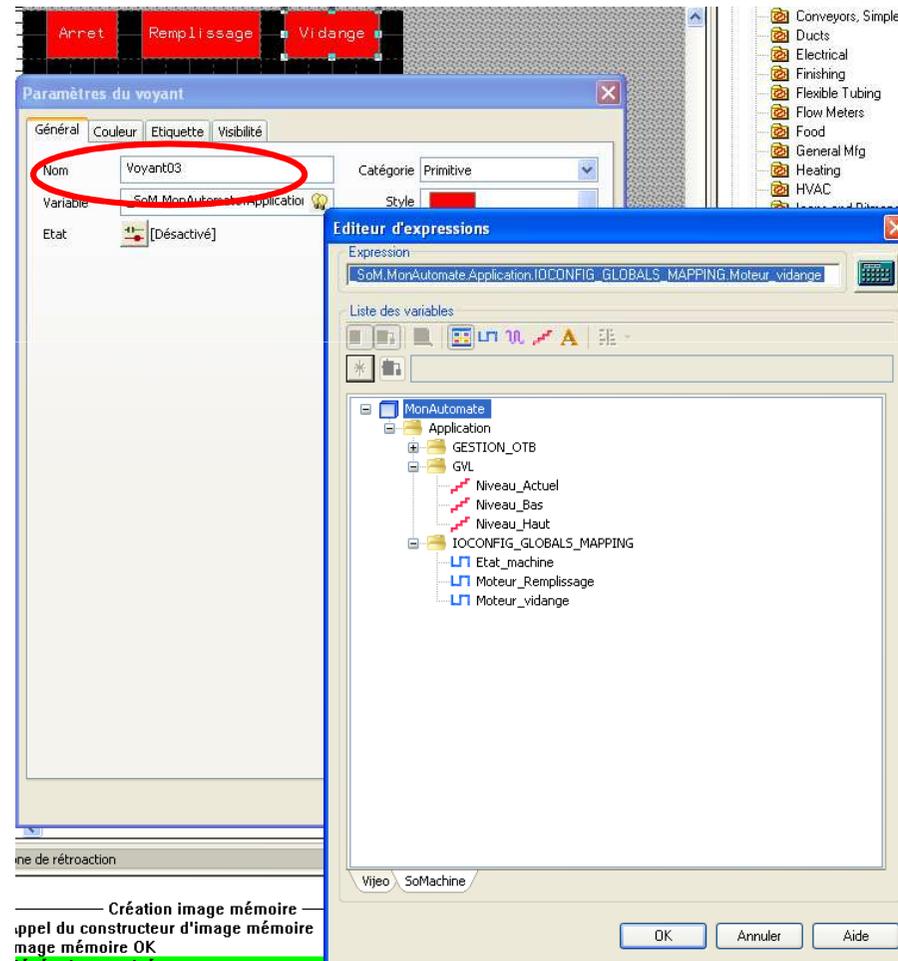


Variables partagées avec l'afficheur

Transparence des échanges

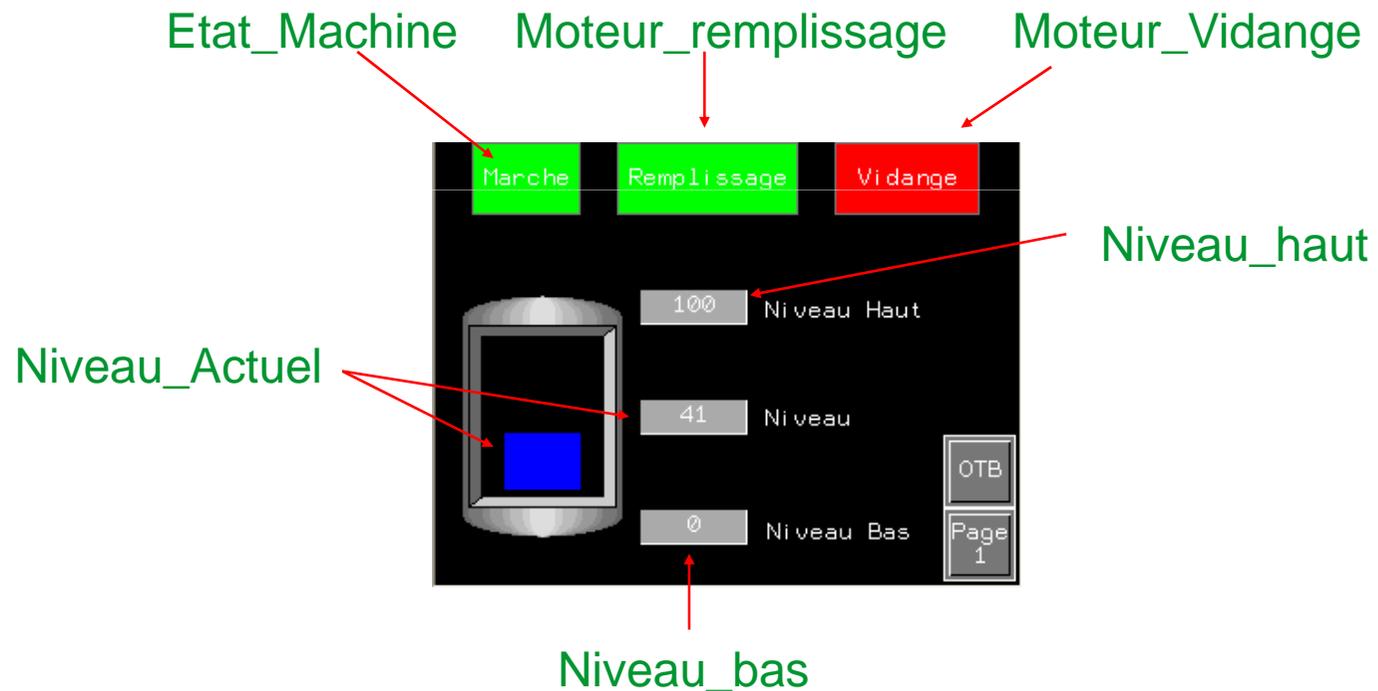
- Etape 2: Affecter les variables dans Vijeo Designer

→ Solution: Utilisation des variables partagées dans Vijeo Designer



Transparence des échanges

- Etape 2: Affecter les variables dans Vijeo Designer
 - Solution: Utilisation des variables partagées dans Vijeo Designer



Transparence des échanges

- Exercice 2 : Recherche des variables dans tout le projet

- Ouvrir une liste de renvoi et saisir par exemple la variable Moteur_Replissage. La liste de renvoi indique l'ensemble des actions effectuées sur cette variable.

The screenshot shows the 'Liste de renvoi' window with the following table:

POU	Variable	Accès	Type	Domaine	Adresse	Position
IO	Moteur_Replissage	Lire	BOOL	Global	AT %QX0.7	
MODE_DE_MARCHE	Moteur_Replissage	Lire	BOOL	Global	AT %QX0.7	Network 3 / Operand 'IoConfig_Globals_Mapp
PILOTAGE_SORTIE	Moteur_Replissage	Lire	BOOL	Global	AT %QX0.7	
PILOTAGE_SORTIE	Moteur_Replissage	Écrire	BOOL	Global	AT %QX0.7	Élément 0 (Impl)
XBTGT2220	Moteur_Replissage	Écrire	BOOL	Global	AT %QX0.7	\XBTGT2220\Ecrans graphiques\Ecrans de ba
XBTGT2220	Moteur_Replissage	Lire	BOOL	Global	AT %QX0.7	\XBTGT2220\Ecrans graphiques\Ecrans de ba

The 'Paramètres du voyant' dialog box shows the following configuration:

- Nom: Voyant02
- Catégorie: Primitive
- Variable: MAPPING.Moteur_Replissage
- Style: 00001
- Etat: [Désactive]

- Si on double clique sur la dernière ligne, Somachine ouvre la fenêtre de configuration de l'écran Vijeo Designer correspondant.

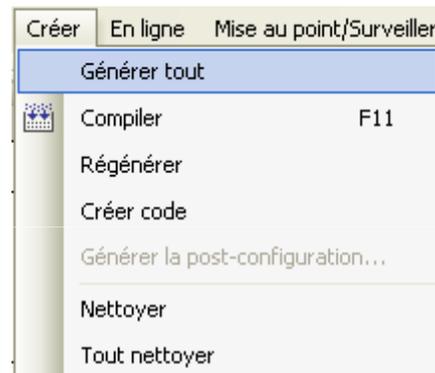
Transparence des échanges

- Exercice 3 : Chargement du projet en un seul point de connexion
 - ➔ Objectif : Générer et transférer un projet unique
 - Etape 1 : Générer le projet
 - Etape 2 : Configuration de la Gateway
 - Etape 3 : Transférer le projet (Contrôler + Afficheur)

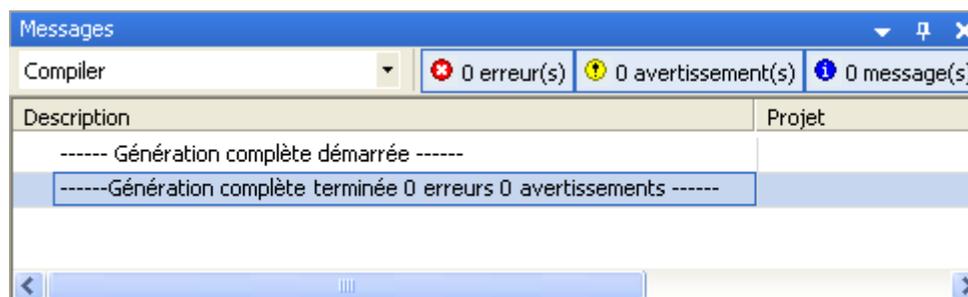
Transparence des échanges

● Etape 1 : Générer le projet

- ➔ Menu « Créer »
- ➔ « Générer tout »



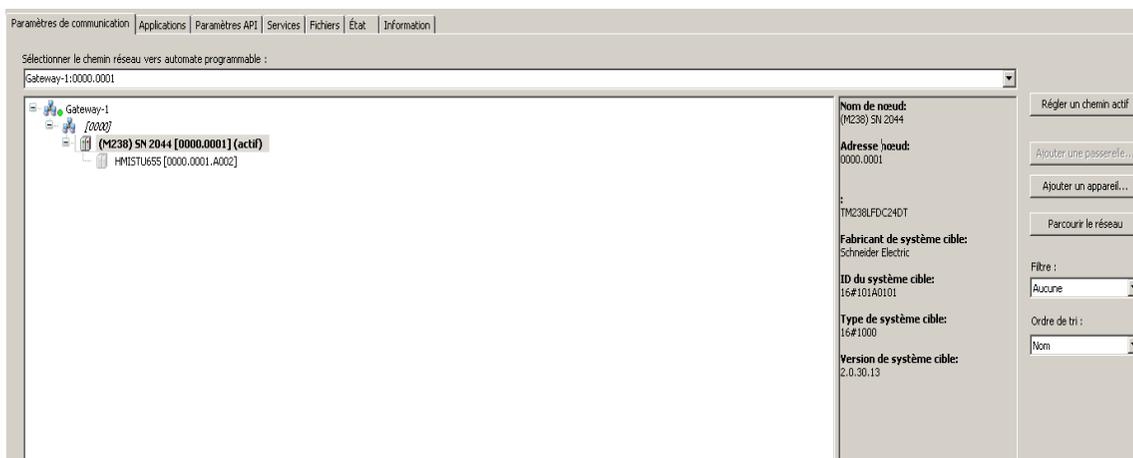
- ➔ Vérification du résultat de la génération



Transparence des échanges

● Etape 2 : Configuration de la Gateway

- Double cliquer sur « MonAutomate » pour ouvrir l'écran de configuration de la gateway.



- Mettre le filtre sur « Aucun »
- Cliquer sur « parcourir le réseau »
- Cliquer sur la gateway et sélectionner l'équipement
- Cliquer sur « régler le chemin actif »

Transparence des échanges

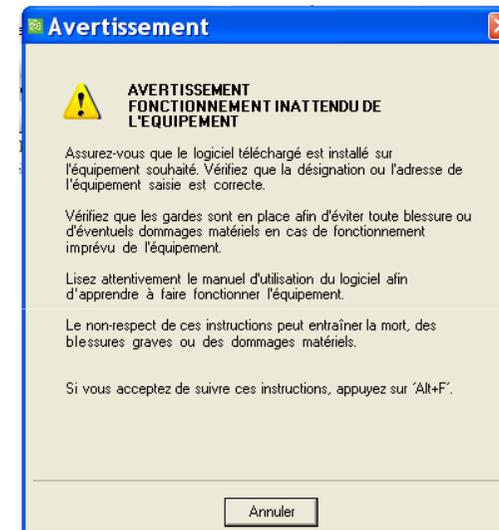
● Etape 2 : Configuration de la Gateway

- Confirmer en faisant Alt + F

Note: La suite ne concerne pas l'XBTGC

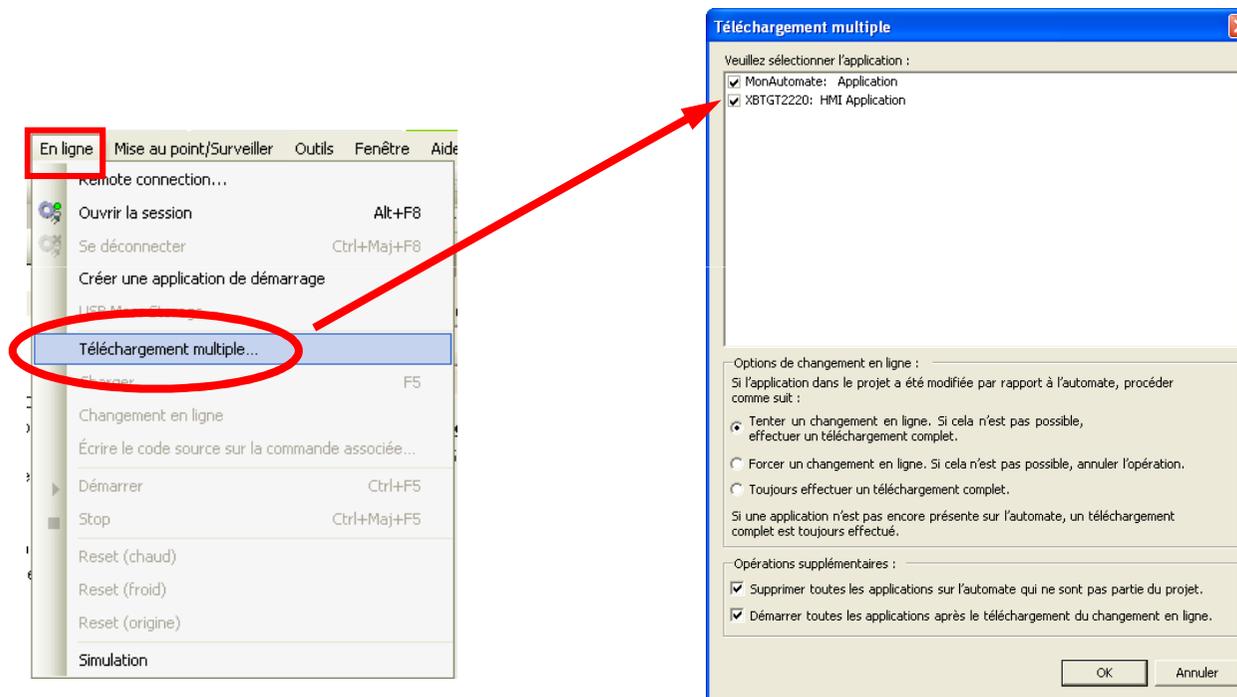
- Double cliquer sur le HMI pour découvrir la gateway puis régler le chemin actif comme précédemment.

- La gateway est prête pour le téléchargement multiple



Transparence des échanges

- Etape 3 : Transférer le projet dans toute l'architecture



➔ Votre machine est prête à fonctionner!



SoMachine

Découverte des valeurs du logiciel SoMachine

→ Pour aller plus loin

Machine  Struxure



SoMachine

Schneider
 Electric

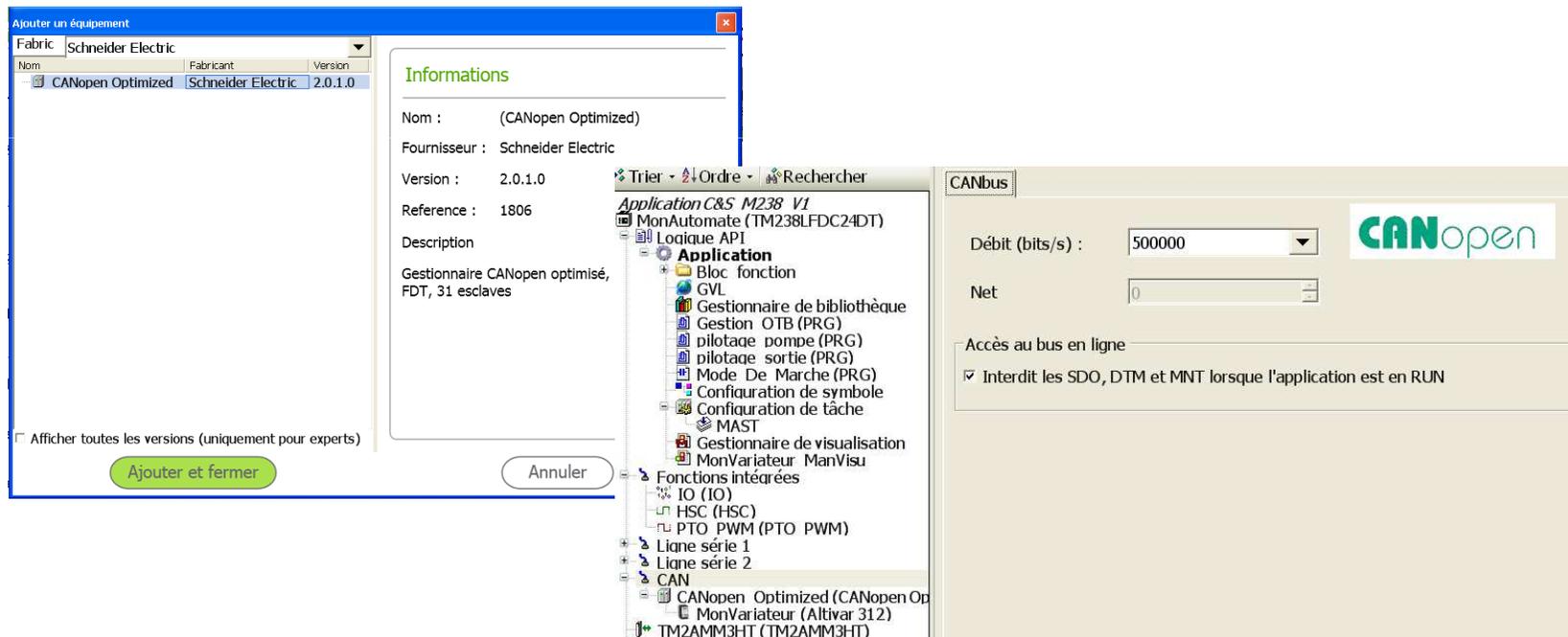
Pour aller plus loin

- Pour remplir le réservoir plus vite, on décide d'y ajouter une pompe à vitesse variable qui sera pilotée par CANopen
- Exercice 1 : Ajout d'un équipement CANopen à partir d'un modèle
 - Etape 1 : Définir le port CANopen puis configurer sa vitesse (500 kb/s)
 - Etape 2 : Créer un nouveau POU vide en langage CFC et l'appeler dans la tâche MAST
 - Etape 3 : Créer un nouvel esclave CANopen à partir d'un modèle.
 - Etape 4 : Tester la fenêtre de visualisation obtenue
 - Etape 5 : Créer les variables locales d'entrées/sorties du bloc de pilotage ATV puis piloter ce bloc à l'aide des variables.

Pour aller plus loin

- Etape 1 : Définir le port CANopen

- Dans l'onglet configuration, définir le port CANopen puis dans l'onglet programmation (double clic sur CAN) configurer sa vitesse (500 kb/s)



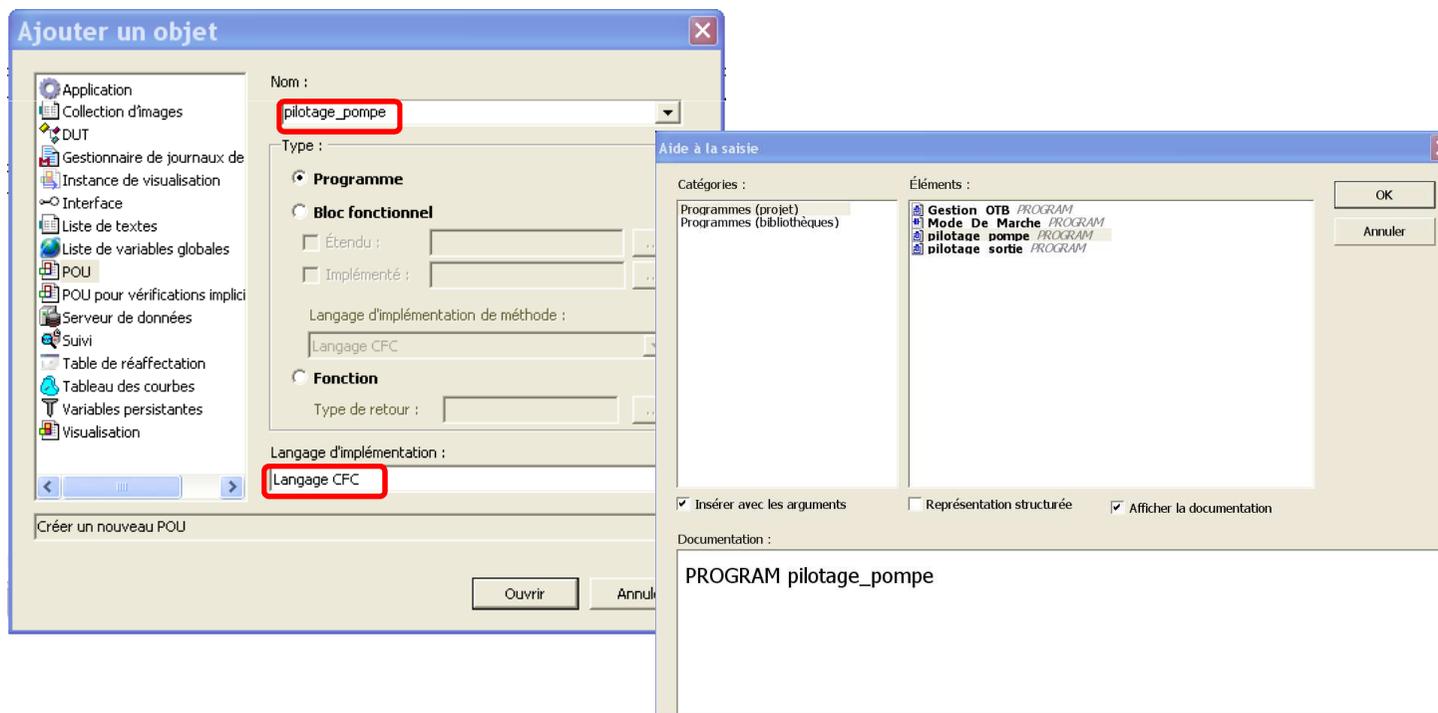
The screenshot displays the software interface for configuring a CANopen port. It is divided into several key areas:

- Ajouter un équipement (Add Equipment):** A dialog box with a table listing available equipment. The selected item is 'CANopen Optimized' by 'Schneider Electric' with version '2.0.1.0'. Below the table is an 'Ajouter et fermer' button.
- Informations (Information):** A panel showing details for the selected equipment:
 - Nom : (CANopen Optimized)
 - Fournisseur : Schneider Electric
 - Version : 2.0.1.0
 - Reference : 1806
 - Description
 - Gestionnaire CANopen optimisé, FDT, 31 esclaves
- Project Tree:** A hierarchical view of the application components. The 'CAN' port is highlighted in blue. Other components include 'Application C&S M238 V1', 'MonAutomate (TM238LFDC24DT)', 'Logique API', 'Application', 'Bloc fonction', 'GVL', 'Gestionnaire de bibliothèque', 'Gestion OTB (PRG)', 'pilotage pompe (PRG)', 'pilotage sortie (PRG)', 'Mode De Marche (PRG)', 'Configuration de symbole', 'Configuration de tâche', 'MAST', 'Gestionnaire de visualisation', 'MonVariateur ManVisu', 'Fonctions intégrées', 'IO (IO)', 'HSC (HSC)', 'PTO PWM (PTO PWM)', 'Ligne série 1', 'Ligne série 2', 'CANopen Optimized (CANopen Op)', 'MonVariateur (Altivar 312)', and 'TM2AMM3HT (TM2AMM3HT)'. Buttons for 'Trier', 'Ordre', and 'Rechercher' are visible above the tree.
- CANbus Configuration:** A panel for configuring the CANopen port. It includes:
 - Debit (bits/s): 500000
 - Net: 0
 - Accès au bus en ligne: Interdit les SDO, DTM et MNT lorsque l'application est en RUN

Pour aller plus loin

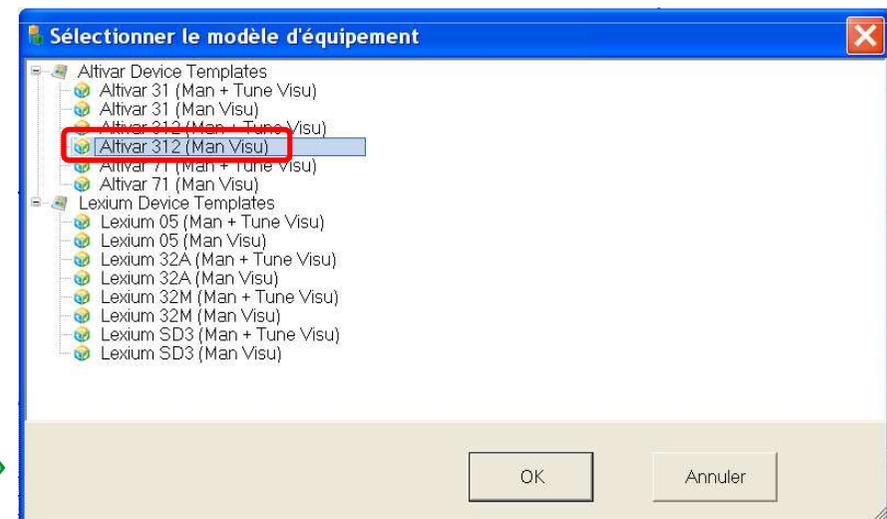
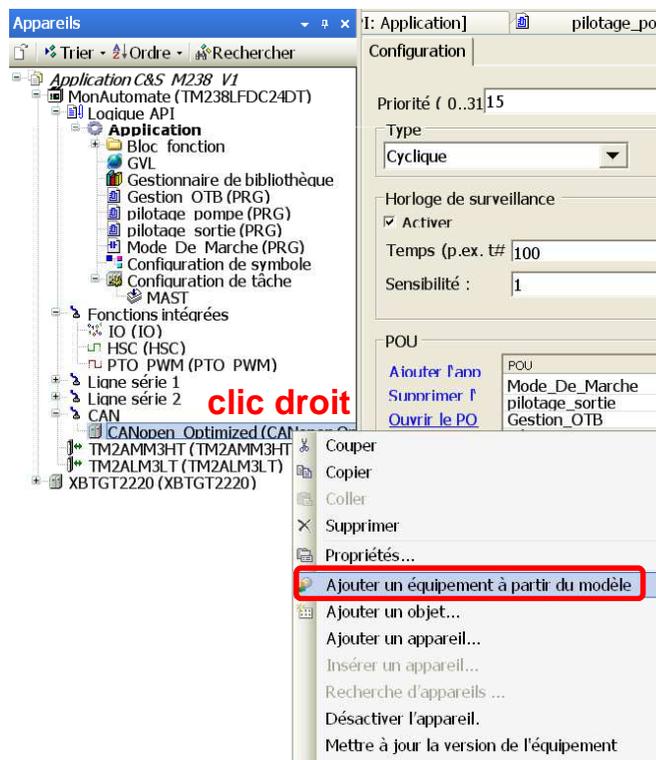
- Etape 2 : Créer un nouveau POU

- Créer un nouveau POU vide en langage CFC appelé pilotage_pompe et l'appeler dans la tâche MAST :



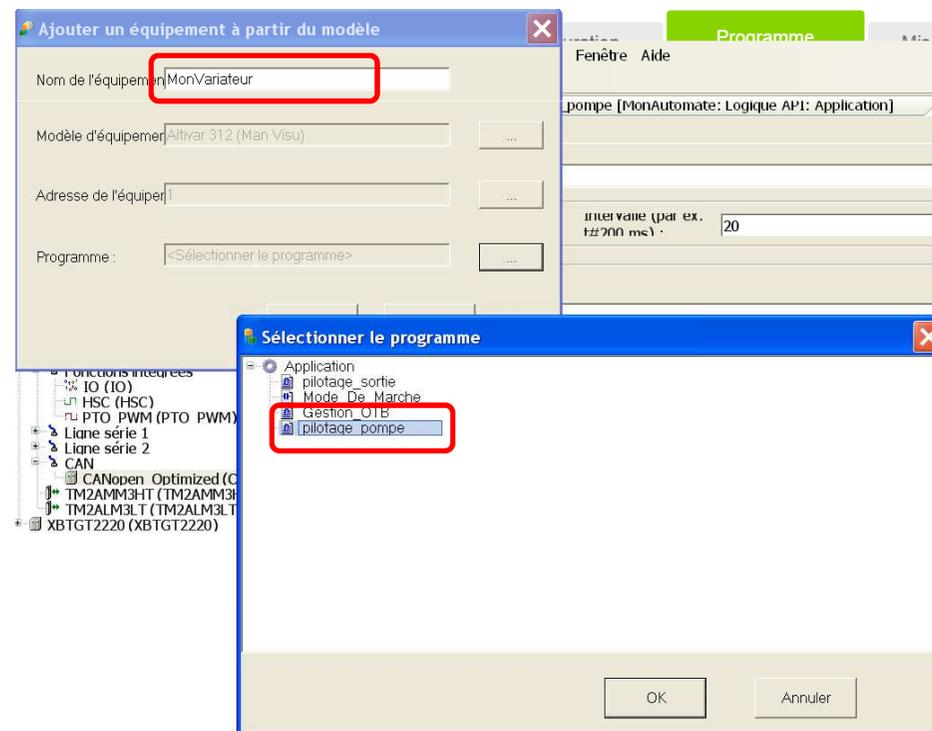
Pour aller plus loin

- Etape 3 : Créer un esclave CANopen et le bloc intégré de démarrage du variateur
 - Créer un nouvel esclave CANopen appelé « MonVariateur » à partir d'un modèle. Choisir un ATV312 à l'adresse 1 :



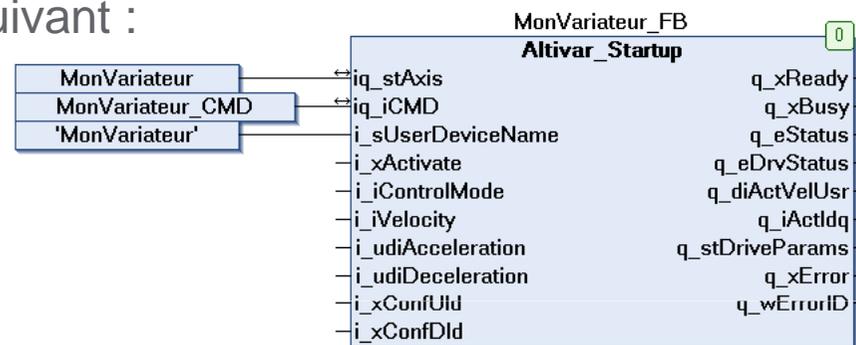
Pour aller plus loin

- Etape 3 : Créer un esclave CANopen et le bloc intégré de démarrage du variateur
 - Choisir le POU pilotage_pompe dans lequel sera créé automatiquement le bloc de pilotage du variateur :



Pour aller plus loin

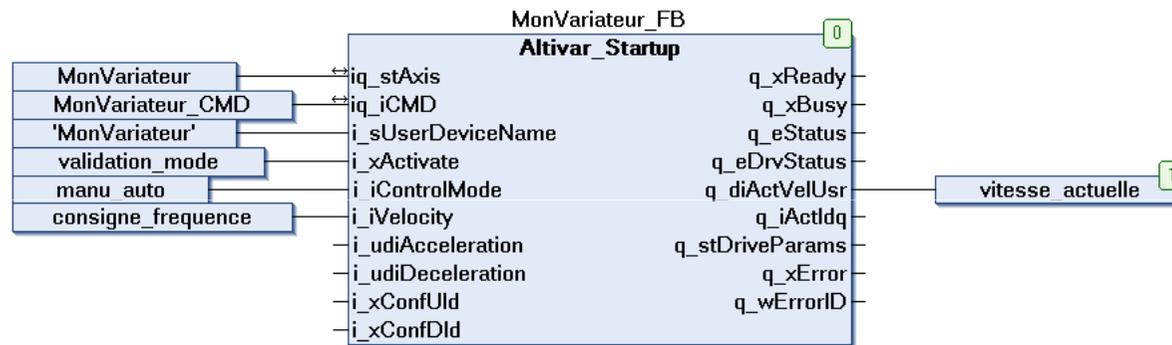
- Etape 4 : Tester la fenêtre de visualisation
 - On obtient ainsi le bloc suivant :



- Ainsi qu'une fenêtre de visualisation permettant le pilotage du variateur depuis SoMachine.
- Tester ce bloc en raccordant un ATV312 correctement paramétré sur le port CANopen

Pour aller plus loin

- Etape 5 : Création des variables de pilotage du bloc de démarrage du variateur
 - Créer les variables locales validation_mode, manu_auto, consigne_fréquence et vitesse_actuelle.

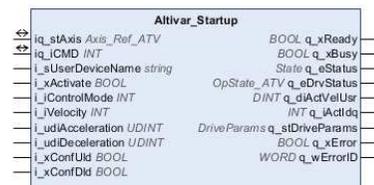


- Regarder l'aide en ligne du bloc Altivar_Startup puis essayer de piloter le bloc à partir des variables.

Function description

This function block supports you in commissioning a frequency inverter for the first time. The function block comprises two visualizations to facilitate usage of the function block.

Graphical representation



Pour aller plus loin

- Etape 5 : Pilotage du variateur

- Solution

- Mettre la variable manu_auto à 1 pour pilotage depuis le bloc.
 - Valider le pilotage auto en mettant la variable validation_mode à TRUE
 - Renseigner la variable consigne_frequence (ex : 30 Hz)
 - Piloter le variateur via la variable MonVariateur_CMD en procédant comme suit :
 - Enable : 1 le variateur passe en **Rdy**
 - MoveVel : 9 le variateur accélère jusqu'à sa consigne de fréquence. La variable vitesse actuelle indique la vitesse en tr/min du moteur
 - Stop : 4 le variateur décélère puis s'arrête
 - Disable : 2 le variateur passe en **NSt** il est dévalidé

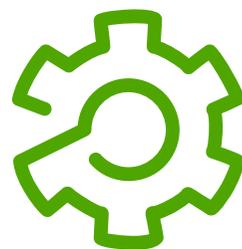


SoMachine

Découverte des valeurs du logiciel SoMachine

→ Conclusion

Machine  Struxure



SoMachine

Conclusion

- Les valeurs clés du logiciel SoMachine:
 - Aide à la conception
 - Logiciel unique et intuitif
 - Simplicité de programmation
 - Transparence des échanges

- Distribution du kit Clic'N Start contenant:
 - Les présentations de l'offre' et de l'atelier' présentés en séance
 - Les applications SoMachine utilisées en séance
 - Un exécutable permettant d'aider la prise en main du logiciel SoMachine à partir de vidéos.

- Le catalogue MachineStruXure est à votre disposition

