Fiche technique variateur

Test et mise en oeuvre rapide des liaisons Modbus filaire et Ethernet TCP Modbus entre un automate M340 et les variateurs de vitesse ATV600-ATV900

Gamme : ATV600 – ATV900 – M340 – Modbus RS485 – Modbus TCP

Introduction

Le but de ce dossier est de montrer comment mettre en œuvre ou tester les liaisons Modbus filaire, Ethernet TCP Modbus et Ethernet IP entre un automate M340 et les variateurs de vitesse Altivar Processe de référence ATV6x0 et ATV9x0, avec une **programmation minimum**.

Sommaire

1- Comment piloter un variateur Altivar avec un automate 2

1-1 Les variables variateur Altivar process ATV6x0 et ATV9x0 2

1-2 Canal Consigne et canal commande 2

1-3 Les profils 3

Non séparés 3

Séparés 3

Profil IO 4

1- 4 Le scanner d'IO 4

2- Architectures 6

3- Configuration du variateur 6

3-1 Configuration de l'ATV600-ATV900 en Ethernet TCP/modbus (processeur, NOE, NOC) .. 6

3-3 Configuration du variateur en Modbus filaire RS485 6

3-4 Configuration du variateur en profil IO 6

3-5 Ajoût de variable dans les scanners d'IO 7

4- Configuration de l'automate 8

4-1 Liaison Ethernet à partir d'un port natif du processeur 8

4-2 Liaison Ethernet à partir du coupleur NOE 13

4-3 Liaison Ethernet à partir du coupleur BMXMOC401 16

4-4 Liaison Modbus filaire ou RS485 à partir du port processeur 22

5- Annexes – Détail des requêtes de lecture / écriture 25

5-1 Requête lecture / écriture sans utilisation des scanners d'IO 26

5-2 Requête de lecture/écriture avec utilisation du scanner d'IO issu d'une page WEB 27

5-3 Requête de lecture/écriture avec utilisation du scanner d'IO issue de la mémoire interne du variateur 28

1- Comment piloter un variateur Altivar avec un automate

Les variateurs de vitesse ATV12, ATV320, ATV600 et ATV900 peuvent être pilotés soit par **leur bornier LI1, LI2,. ., AI1,AI2**,.. , soit par **le HMI intégré,** soit à **distance** par des automates **M221, M340 ou M580** via les bus ou réseaux de communication **TCP Modbus**, CanOpen , Modbus série.

Les automates verront alors les variateurs comme une table de paramètres, chaque paramètre ayant une adresse et une signification précise.

Chaque type de variateur dispose de tables de paramètres conséquentes, mais, de fait, 4 variables sont suffisantes pour entraîner le moteur :

- **CMD** : variable à écrire, permet d'activer le variateur de Rdy ou NsT vers RUN,
- **LFR** : variable à écrire, lorsque le variateur est en RUN, permet de lui fournir la consigne vitesse,
- **ETA** : permet de connaître l'état du variateur : Prêt ? , sous tension ? , en défaut ?, etc. : variable à lire exclusivement ,
- **RFR** : permet de lire la fréquence réelle du variateur.

1-1 Les variables variateur Altivar process ATV6x0 et ATV9x0

Lees Les variateurs ATV600 et ATV900 comportent 1210 variables possibles et les plus utilisées sont :

	A	B	C	D	E	F
1	Code	Name	Logic address	CANopen index	DeviceNet path	Link
2	CMD	Command register	16#2135 = 8501	16#2037/2	16#8B/01/66 = 139/01/102	CMD
3	CMI	Extended control word	16#2138 = 8504	16#2037/5	16#8B/01/69 = 139/01/105	CMI
4	RPR	Time counter reset	16#0C30 = 3120	16#2001/15	16#70/01/79 = 112/01/121	RPR
5	LFRD	Speed setpoint	16#219A = 8602	16#2038/3	16#8C/01/03 = 140/01/03	-
6	LFR	Reference frequency	16#2136 = 8502	16#2037/3	16#8B/01/67 = 139/01/103	-
7	PISP	PIDSet Point	16#2137 = 8503	16#2037/4	16#8B/01/68 = 139/01/104	-
8	AIV1	AIV1 Image input	16#14A1 = 5281	16#2016/52	16#7B/01/52 = 123/01/82	-
9	MFR	Multiplying coefficient	16#2E37 = 11831	16#2058/20	16#9C/01/20 = 156/01/32	-
10	ETA	CIA402 State Register	16#0C81 = 3201	16#2002/2	16#71/01/02 = 113/01/02	ETA
11	HMIS	Drive state	16#0CA8 = 3240	16#2002/29	16#71/01/29 = 113/01/41	HMIS
12	ETI	Internal State Register	16#0C86 = 3206	16#2002/7	16#71/01/07 = 113/01/07	ETI
13	CRC	Channel for reference frequency	16#20F9 = 8441	16#2036/2A	16#8B/01/2A = 139/01/42	CCC
14	CCC	Active command channel	16#20FA = 8442	16#2036/2B	16#8B/01/2B = 139/01/43	CCC
15	CNFS	Active configuration	16#1F54 = 8020	16#2032/15	16#89/01/15 = 137/01/21	CNFS
16	STUN	Tune selection	16#2591 = 9617	16#2042/12	16#91/01/12 = 145/01/18	STUN
17	SMOT	Status of motor tune in term of saliency	16#25AD = 9645	16#2042/2E	16#91/01/2E = 145/01/46	SMOT
18	RFRD	Output velocity	16#219C = 8604	16#2038/5	16#8C/01/05 = 140/01/05	-
19	RFR	Motor Frequency	16#0C82 = 3202	16#2002/3	16#71/01/03 = 113/01/03	-
20	OTR	Motor torque	16#0C85 = 3205	16#2002/6	16#71/01/06 = 113/01/06	-
21	LCR	Motor current	16#0C84 = 3204	16#2002/5	16#71/01/05 = 113/01/05	-

1-2 Canal Consigne et canal commande

Un variateur Schneider Electric est piloté via deux canaux : commande et consigne :

- son canal commande : c'est l'ordre de marche : une information binaire délivrée par exemple au bornier par l'entrée DI1 mais qui peut l'être aussi par les mots CMD et ETA échangé entre un automate et le variateur à partir d'une connexion Ethernet Modbus TCP, Ethernet IP ou Modbus filaire.
  En ce cas,l'automate pilote le variateur en écrivant son mot de commande CMD et en vérifiant si l'ordre qu'il a envoyé a bien été interprété par le variateur par la lecture du mot ETA.
- son canal consigne : c'est la consigne de vitesse : une information numérique délivrée par exemple par un entrée analogique au bornier AI1 ou par Ethernet TCP via le mot LFRD.

1-3 Les profils

Il est possible de configurer ces canaux consigne et commande via trois profils : **Séparés**, **Non séparés** et **Profil IO.**

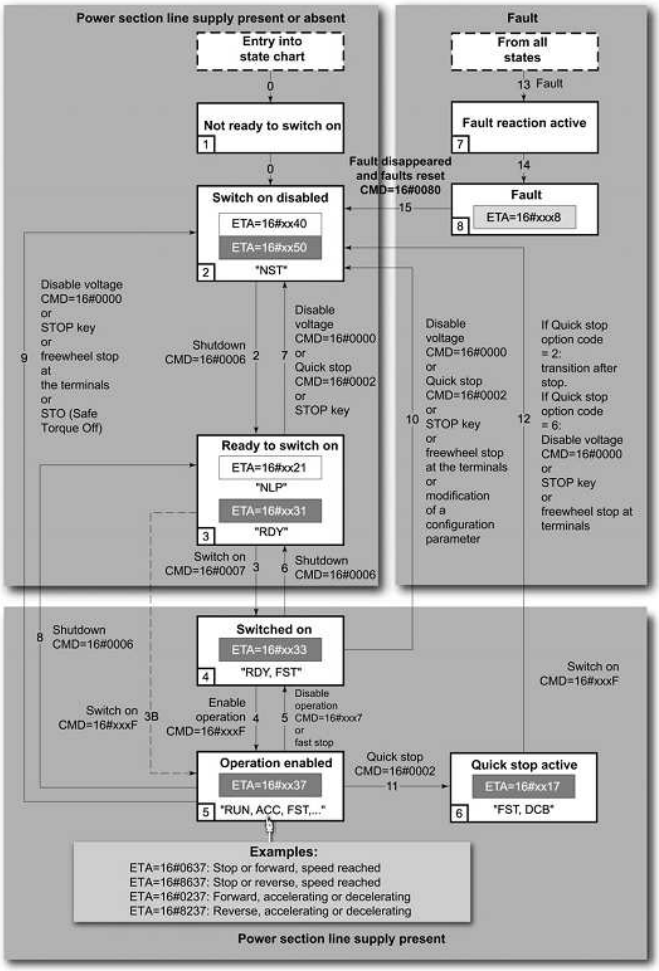
Non séparés

Déterminer le canal consigne revient à déterminer de facto le canal commande.

Séparés

Il est nécessaire de spécifier le canal consigne et le canal commande, exemple faire démarrer le variateur avec l'entrée bornier DI1 et lui fournir une consigne de vitesse via un automate par sa connexion intégrée Ethernet TCP.

Pour ces deux profils, lorsque le canal commande du variateur est choisi comme étant un bus (CanOpen, Modbus RS485) ou un réseau (Ethernet), l'automate et le variateur vont respecter un protocole d'échange normalisé (CIA402) aussi appelé DriveCom et qui se traduit par une séquence bien déterminée entre les mots de Commande CMD et mot d'état ETA.



Séquence à respecter :

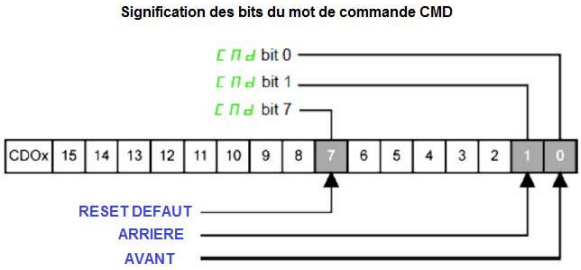
- Etat initial du variateur : ETA = **16#0250** ou **16#02D0** ➔ le variateur affiche NST,
- Ecrire **CMD** = **16#0006,** attendre que **ETA** passe à **16#x231** ou à **16#x2B1** ➔ le variateur affiche RDY,
- Ecrire **CMD** = **16#0007**, attendre que **ETA** passe à **16#x233** ou **16#2B3,**
- Ecrire **CMD** = **16#000F,** attendre que **ETA** passe à **16#x637** ou **16#2B7** ➔ le variateur affiche RUN,
- Dès que ETA = **16#x637,** écrire LFRD = valeur de fréquence souhaitée, vérifier que LFR ait cette valeur, puis ETA passe à **16#x6B7,** le variateur affiche la fréquence, le moteur fonctionne.

Pour arrêter le moteur, écrire **CMD** = **16#0000,**
Pour acquitter un défaut, écrire **CMD** = **16#0080.**

Profil IO

Il est caractérisé par :

- la spécification des deux canaux séparés comme pour le profil Séparés,
- un échange très simplifié entre le variateur et l'automate, Le CMD n'a plus la même signification que lorsque l'on est en mode Séparés ou Non séparés, il suffit, dans ce mode, de positionner le CMD à 1 et d'écrire la consigne.



Etat initial du variateur : ETA = **16#x233,**
Ecrire **CMD = 16#0001** dans le variateur, attendre que **ETA** passe à **16#x637,**
Dès que **ETA = 16#x637**, écrire **LFRD** = fréquence souhaitée dans le variateur,

1- 4 Le scanner d'IO

Les automates Schneider disposent de requêtes qui permettent de lire ou écrire des tableaux de variables consécutives.

Les variables à lire ou écrire dans le variateur ne sont pas toutes consécutives, auquel cas il sera nécessaire d'utiliser autant de fois ces requêtes de lecture/écriture qu'il y a de variables non consécutives.

C'est pour cette raison qu'il existe dans les variateurs un scanner d'IO, en fait deux tables de x mots (en écriture et en lecture) dans lesquelles il est possible de définir des adresses de variables non consécutives. L'automate lira alors en une seule requête de lecture/écriture jusqu'à 2x variables.

Deux scanners d'IO co-existent pour le variateur ATV320 :

- **Scanner d'IO inclus directement dans la mémoire du variateur**, il est adressable dans la continuité des autres variables généralement : 12741 pour la première adresse d'écriture et 12761 pour la première adresse de lecture, la dimension est variable suivant les variateurs ainsi le scanner
  Il est utilisé lors de tous type de communication : modbus filaire, CanOpen, Ethernet , etc..

  Ce scanner est modifiable directement à partir de l'afficheur intégré du variateur.

Code	Name	Logic address
NC1	Communication scanner, value of write word 1	16#31D9 = 12761
NC2	Communication scanner, value of write word 2	16#31DA = 12762
NC3	Communication scanner, value of write word 3	16#31DB = 12763
NC4	Communication scanner, value of write word 4	16#31DC = 12764
NC5	Communication scanner, value of write word 5	16#31DD = 12765
NC6	Communication scanner, value of write word 6	16#31DE = 12766
NC7	Communication scanner, value of write word 7	16#31DF = 12767
NC8	Communication scanner, value of write word 8	16#31E0 = 12768
NM1	Communication scanner, value of read word 1	16#31C5 = 12741
NM2	Communication scanner, value of read word 2	16#31C6 = 12742
NM3	Communication scanner, value of read word 3	16#31C7 = 12743
NM4	Communication scanner, value of read word 4	16#31C8 = 12744
NM5	Communication scanner, value of read word 5	16#31C9 = 12745
NM6	Communication scanner, value of read word 6	16#31CA = 12746
NM7	Communication scanner, value of read word 7	16#31CB = 12747
NM8	Communication scanner, value of read word 8	16#31CC = 12748

CMD
Consigne
LSP : Low Speed

ETAT
Vitesse
LCR : Retour courant

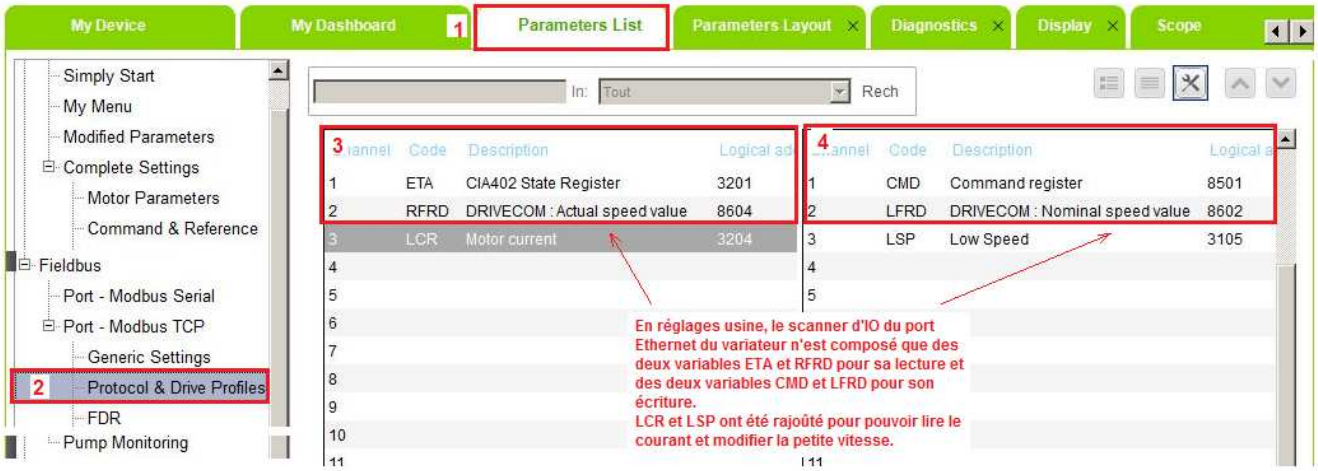
Scanner d'IO du variateur ATV320
8 mots en lecture
8 mots en écriture

Réglages usine

Variables ajoûtées en exemple

- **Scanner d'IO extérieur au variateur**, il est typique des connexions Ethernet, il est inclus soit dans la page WEB d'une carte Ethernet extérieure comme La carte VW3 pour le variateur ATV320, soit dans celle du variateur lui-même dans le cas où le variateur dispose d'une connexion Ethernet native, ainsi les variateurs ATV600 et ATV900. La première adresse d'écriture est zéro ainsi que la première adresse de lecture.

Capture d'écran de SoMove



En retour réglage usine, les scanners d'IO, qu'ils soient internes ou intégrés au port Ethernet du variateur sont prédéterminés avec deux variables en lecture (ETA et RFRD) et de deux variables en écriture (CMD et LFRD), il est possible d'ajoûter des variables ainsi dans l'exemple LCR (Courant moteur) et LSP (vitesse minimum).

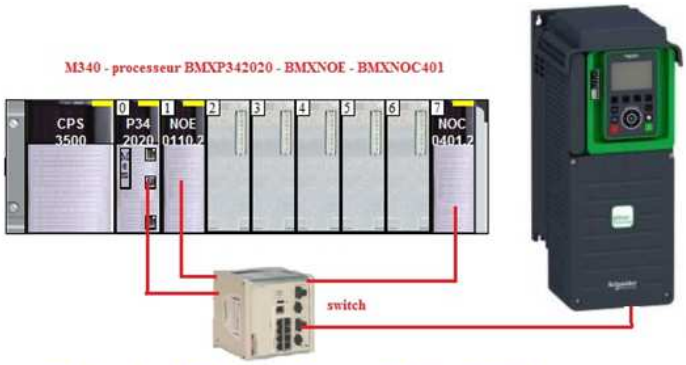
Scanner de lecture
Adresse 0 ou 12761 : ETA : Etat du variateur (valeur en retour réglage usine),
Adresse 1 ou 12762 : RFRD : Vitesse (valeur en retour réglage usine),
Adresse 2 ou 12763 : LCR : Courant (valeur à définir).

Scanner d'écriture
Adresse 0 ou 12741 : CMD :Commande du variateur (valeur en retour réglage usine),
Adresse 1 ou 12742 : LFRD : Consigne vitesse (valeur en retour réglage usine),
Adresse 2 ou 12743 : LSP : Petite vitesse (valeur à définir).

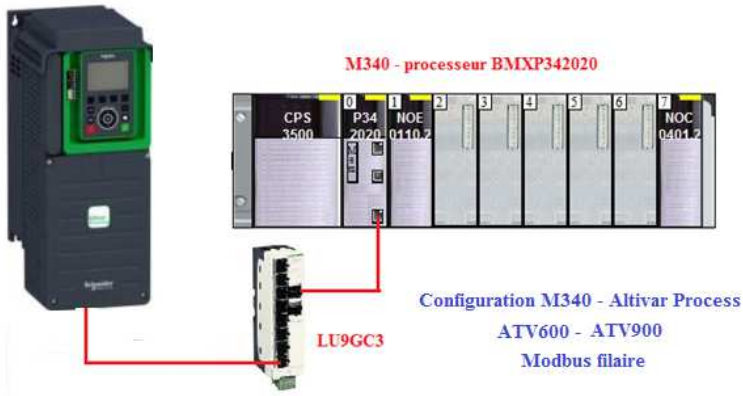
N'importe quel système de commande (automate programmable) peut accéder à l'un ou l'autre. La différenciation est réalisée par le paramètre Id :

- Id = 0 ➔ Scanner d'IO du variateur,
- Id=255 ➔ Scanner d'IO de la page WEB du coupleur Ethernet qu'il soit natif ou résident dans une carte externe.

2- Architectures



Configuration M340 - Altivar Process ATV600 - ATV900
Ethernet TCP Modbus



Configuration M340 - Altivar Process
ATV600 - ATV900
Modbus filaire

3- Configuration du variateur

3-1 Configuration de l'ATV600-ATV900 en Ethernet TCP/modbus (processeur, NOE, NOC)

Attention : les coupleurs Ethernet intégrés ne disposent que du protocole Ethernet TCP Modbus et non pas Ethernet IP

Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Config.Réf.Fréq 1= Ethernet Embarqué,**
➔**Mode Contrôle = Non Séparé,**

Menu 6 Communication ➔ 6.1 Paramètres Communi.➔ Configuration Ethernet
➔Mode IP Ethern. Emb.= **Fixe**
➔Adresse IP = adresse souhaitée, dans cet exemple **84.0.2.6**
➔**Passerelle=x.x.x.x, (0.0.0.0 si pas besoin de passerelle),**
➔**Masque = 255.255.255.0.**

3-3 Configuration du variateur en Modbus filaire RS485

Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Config.Réf.Fréq 1= Fréq.Réf.Modbus,**
➔**Mode Contrôle = Non Séparé,**

Menu 6 Communication ➔ 6.1 Paramètres Communi.➔ Modbus SL ➔ Bus Terrain Modbus
➔Adresse Modbus =1
➔Vitesse Modbus = 19200 bits/s
➔**Format Modbus = $8^{E}1$**

3-4 Configuration du variateur en profil IO

Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔ Mode contrôle = Profil E/S

Modbus TCP
Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Config.Réf.Fréq 1= Ethernet Embarqué,**
Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Canal Commande 1= Ethernet**

Modbus filaire
Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Config.Réf.Fréq 1= Fréq.Réf.Modbus,**
Réglages complets ➔ 5.4 ➔ Commandes et référen. ➔**Canal Commande 1= MODBUS**

3-5 Ajoût de variable dans les scanners d'IO

Le propre des scanners d'IO est de pouvoir ajouter des variables en lecture et écriture afin de compléter les deux jeux de variables prévus par Schneider en retour réglage usine :

- CMD et Consigne pour écriture,
- ETAT et Vitesse pour lecture.

Ajoût dans le scanner d'IO du variateur

Pour pouvoir lire l'état thermique variateur, il est nécessaire d'écrire son adresse (9630) dans le troisième mot du scanner d'entrée interne au variateur, les deux premiers mots étant habituellement dévolus aux paramètres ETAT et Vitesse.

Menu 6 Communication ➔ 6.1 Paramètres Communi.➔ Modbus SL ➔ Bus Terrain Modbus ➔ ScannerComEntrée ➔ **ScanComEntAdr.3 = 9630**.

Pour écrire LSP (petite vitesse) dans la variateur, il est nécessaire d'écrire son adresse (3105) dans le troisième mot du scanner de sortie interne au variateur, les deux premiers mots étant habituellement dévolus aux paramètres CMD et Consigne.

Menu 6 Communication ➔ 6.1 Paramètres Communi.➔ Modbus SL ➔ Bus Terrain Modbus ➔ ScannerComSore ➔ **ScanComSortAdr.3 = 3105**.

Code	Name	Logic address	
NC1	Communication scanner, value of write word 1	16#31D9 = 12761	CMD
NC2	Communication scanner, value of write word 2	16#31DA = 12762	Consigne
NC3	Communication scanner, value of write word 3	16#31DB = 12763	LSP : Low Speed
NC4	Communication scanner, value of write word 4	16#31DC = 12764	
NC5	Communication scanner, value of write word 5	16#31DD = 12765	
NC6	Communication scanner, value of write word 6	16#31DE = 12766	
NC7	Communication scanner, value of write word 7	16#31DF = 12767	
NC8	Communication scanner, value of write word 8	16#31E0 = 12768	
NM1	Communication scanner, value of read word 1	16#31C5 = 12741	ETAT
NM2	Communication scanner, value of read word 2	16#31C6 = 12742	Vitesse
NM3	Communication scanner, value of read word 3	16#31C7 = 12743	LCR : Retour courant
NM4	Communication scanner, value of read word 4	16#31C8 = 12744	
NM5	Communication scanner, value of read word 5	16#31C9 = 12745	
NM6	Communication scanner, value of read word 6	16#31CA = 12746	
NM7	Communication scanner, value of read word 7	16#31CB = 12747	
NM8	Communication scanner, value of read word 8	16#31CC = 12748	

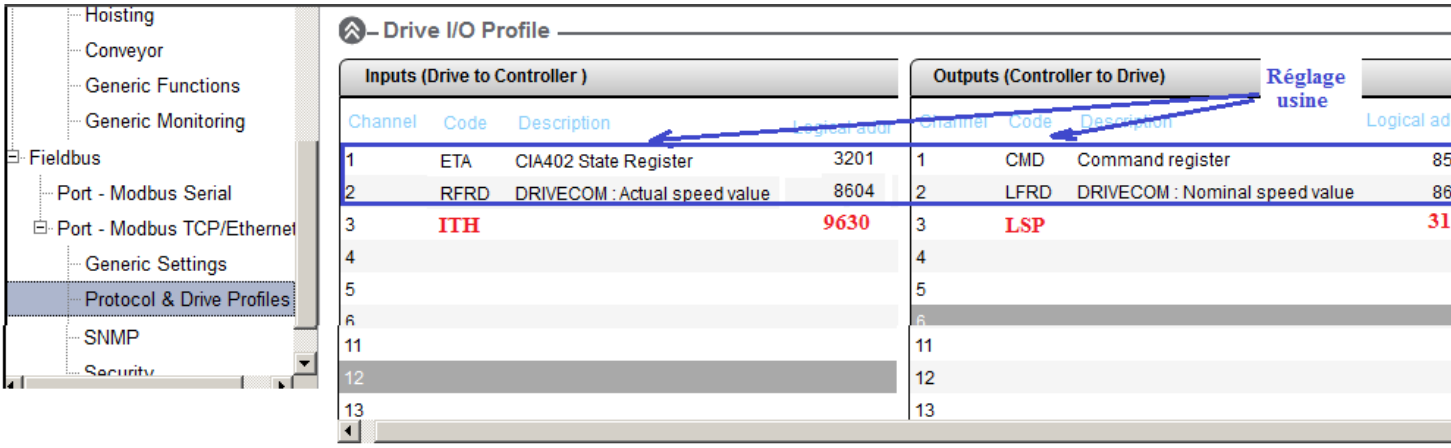
Scanner d'IO du variateur ATV320
8 mots en lecture
8 mots en écriture

Réglages usine

Variables ajoûtées en exemple

Ajoût dans le scanner d'IO lié au port Ethernet TCP modbus natif intégré aux variateurs ATV600 et ATV900

Ce scanner d'IO n'est accessible qu'à partir de SoMove

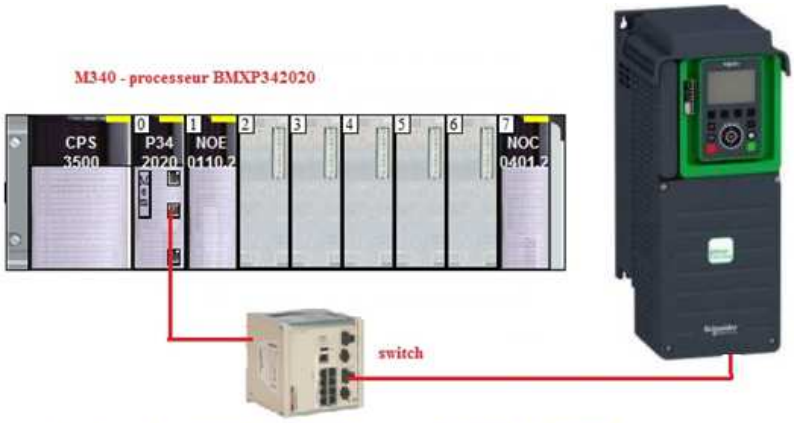


4- Configuration de l'automate

Suivant la nature des coupleurs utilisés, il y a trois façons de gérer une communication Ethernet sur l'automate M340 :

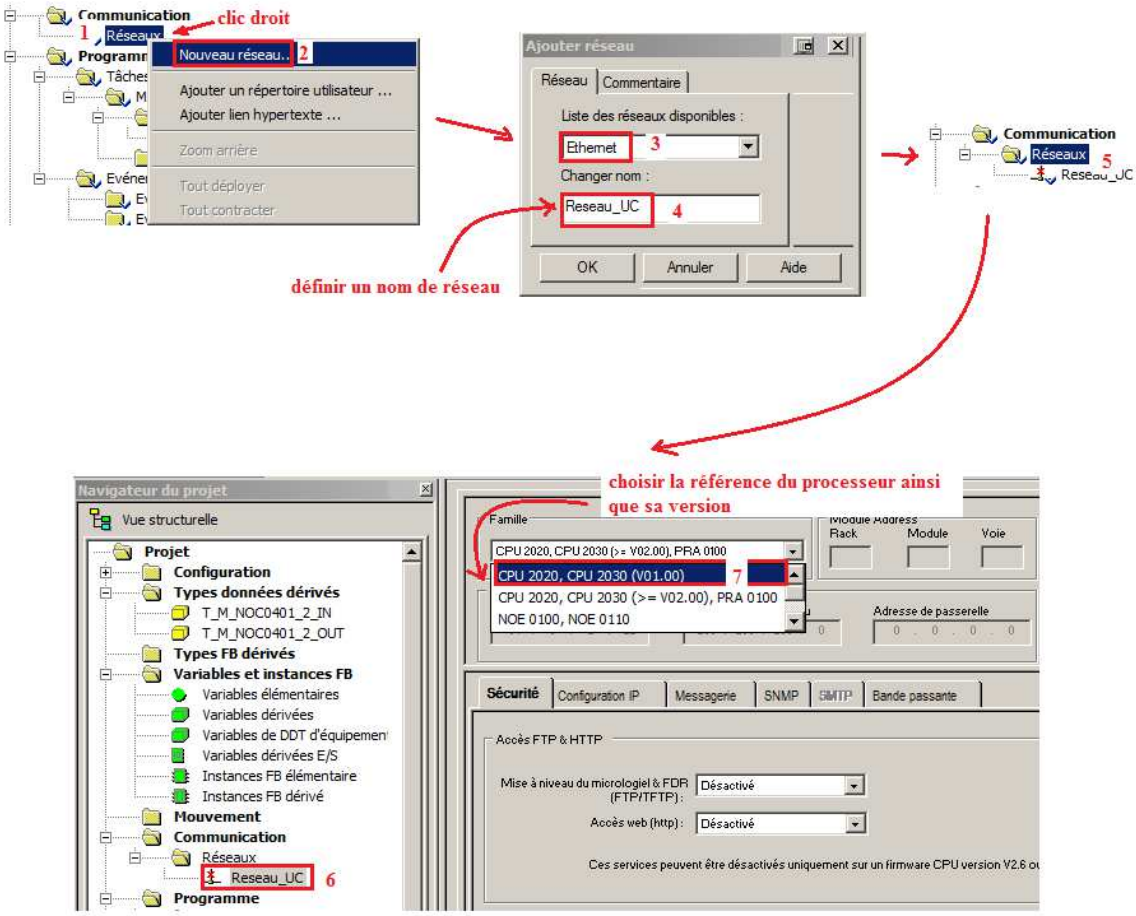
- à partir d'un port Ethernet natif sur certains processeurs M340, exemple le processeur P342020, utilisation des DFBs ReadVar et WriteVar,
- à partir d'anciens coupleurs type NOE, ils disposent d'un outil ''IO Scannnig'' intégré à Unity,
- à partir de coupleur de type NOC, ceux –ci exigent une gestion par Unity de fichiers DTMs.

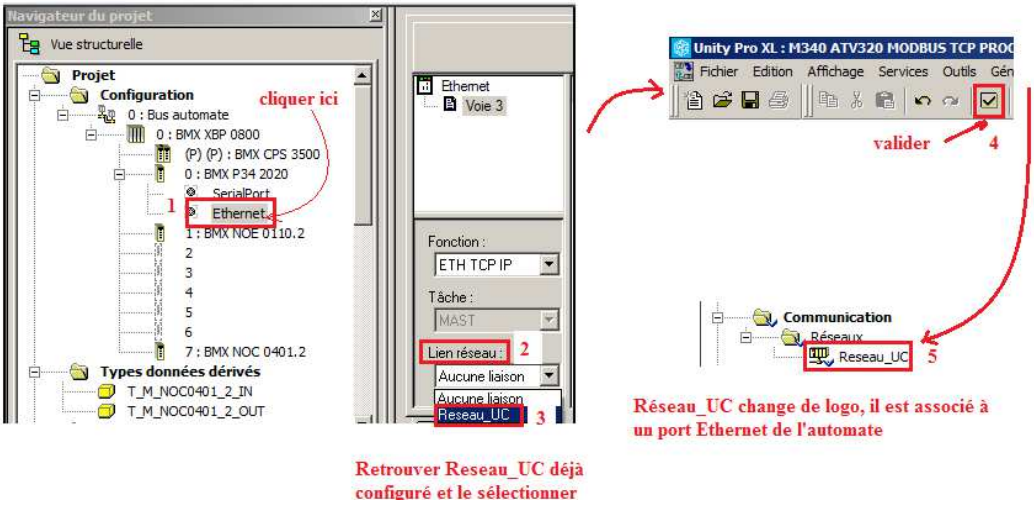
4-1 Liaison Ethernet à partir d'un port natif du processeur



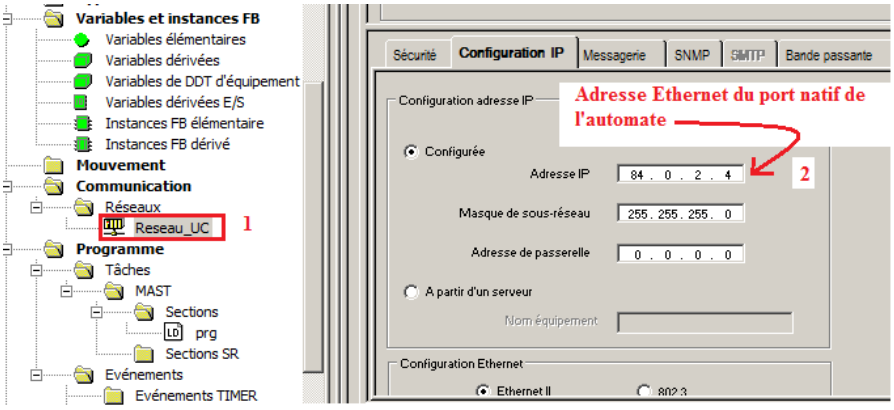
Configuration M340 - Altivar Process ATV600 - ATV900
Ethernet TCP Modbus

Configuration du port Ethernet natif du processeur



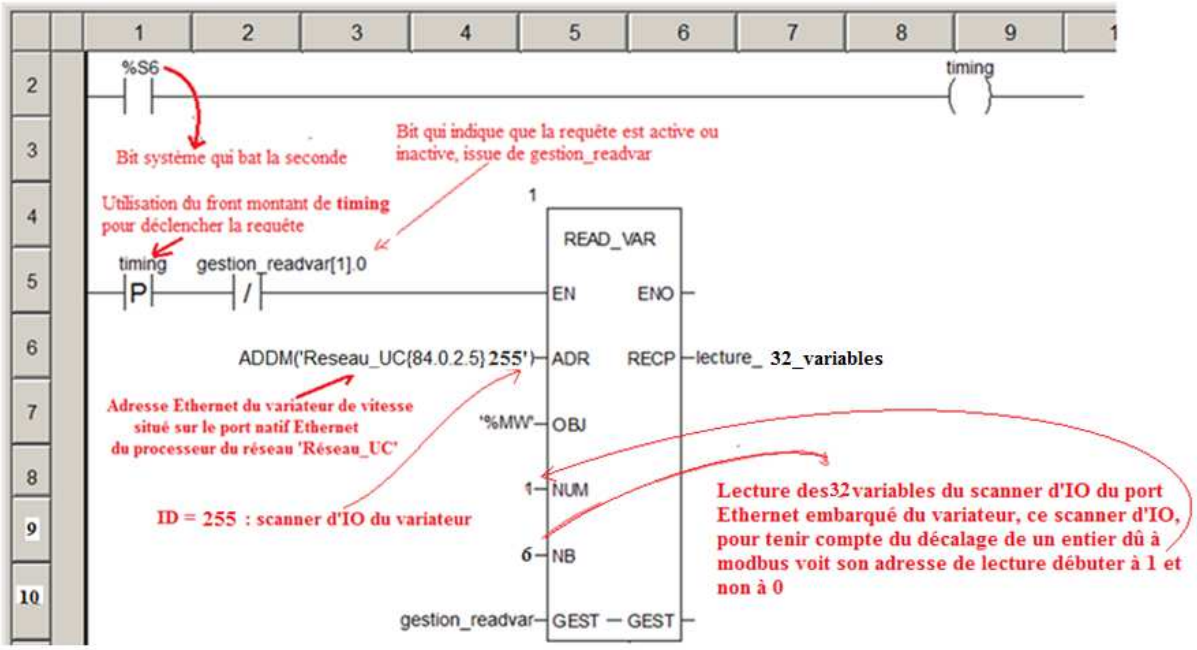


Déclaration de l'adresse Ethernet du variateur ATV6x0 ou ATV9x0

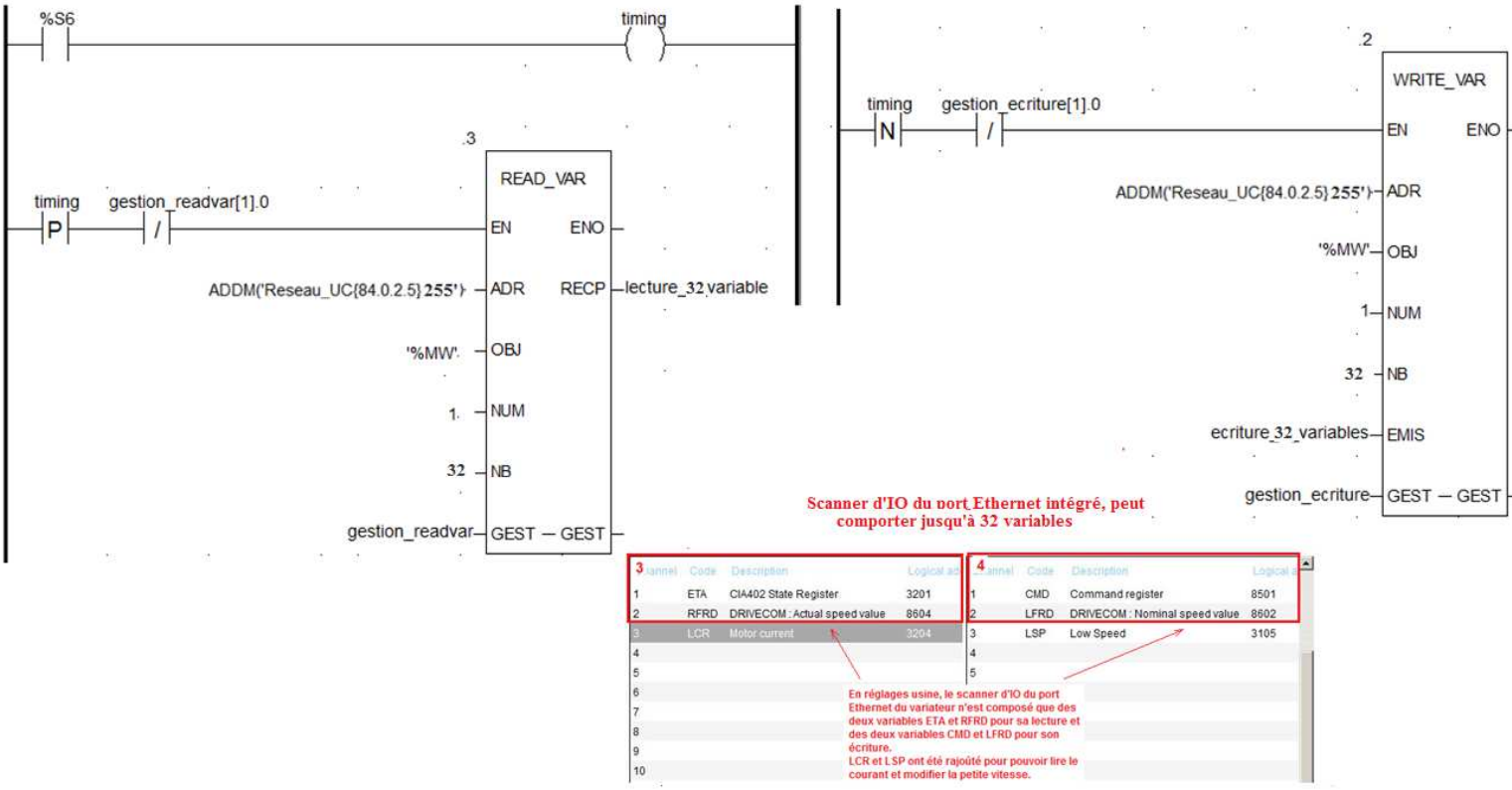


Il est nécessaire d'utiliser et gérer les requêtes READ_VAR et WRITE_VAR pour pouvoir lire et écrire dans la variateur.

4-1-1Communication à partir du scanner d'IO intégré dans le variateur



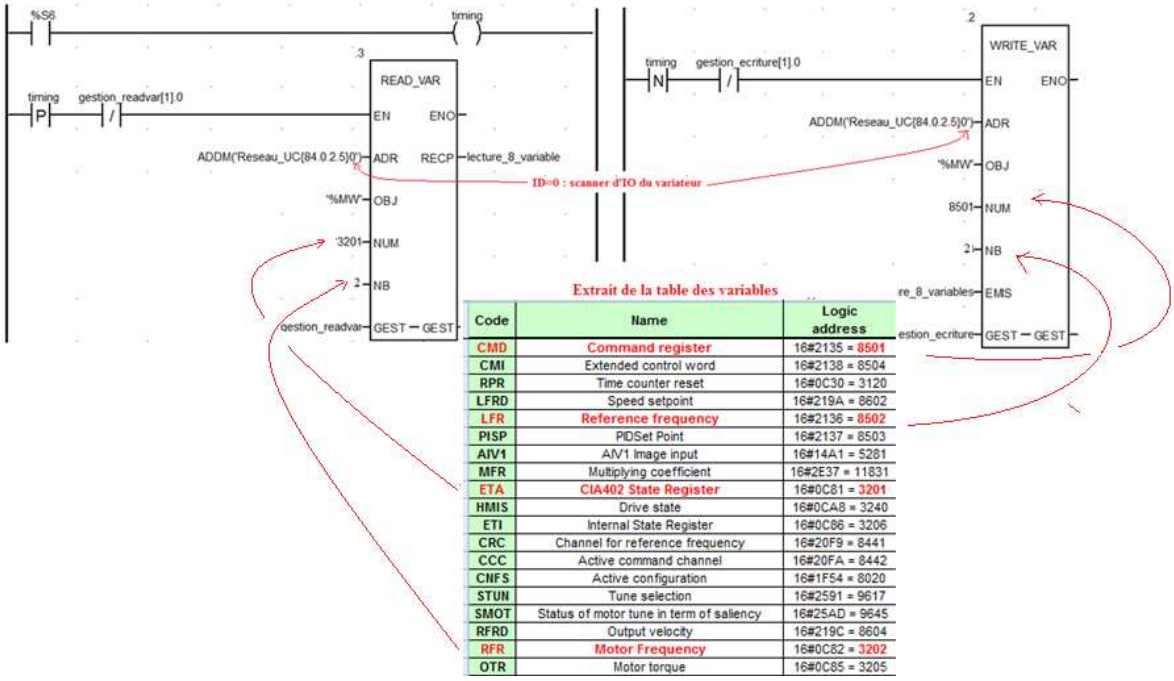
La programmation de test complète sera alors :



3			
1	ETA	CIA402 State Register	3201
2	RFRD	DRIVECOM : Actual speed value	8604
3	LCR	Motor current	3204
4			
5			
6			
7			
8			
9			
10			

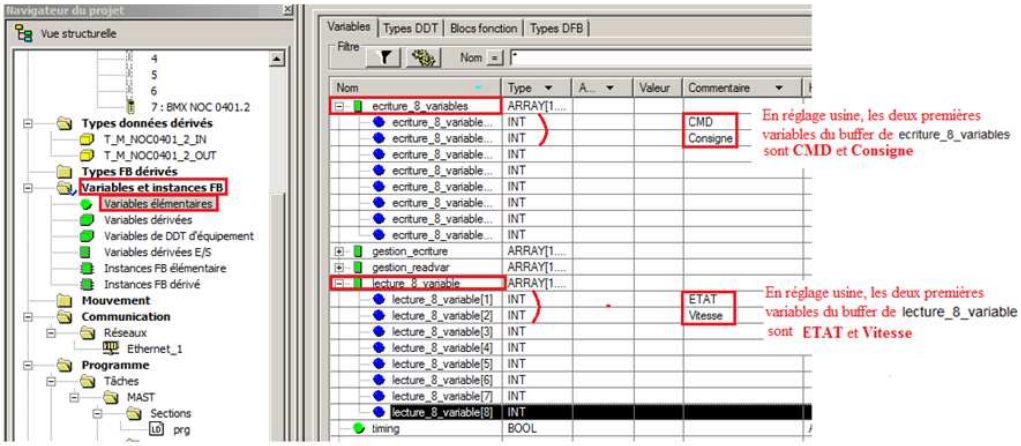
4			
1	CMD	Command register	8501
2	LFRD	DRIVECOM : Nominal speed value	8602
3	LSP	Low Speed	3105
4			
5			

4-1-2 Communication sans scanner d'I0 directement à partir du tableau des variables automates

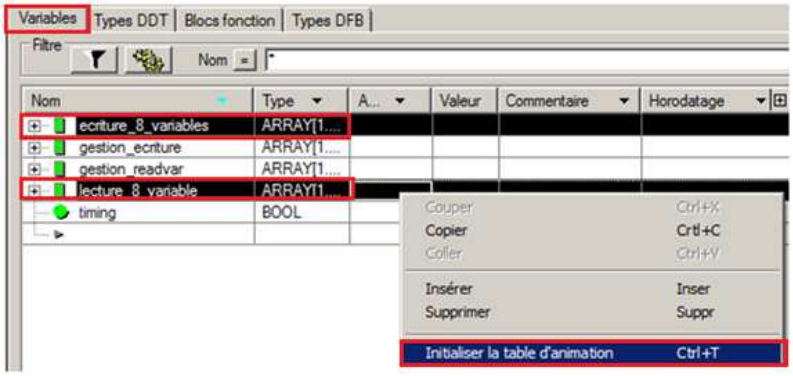


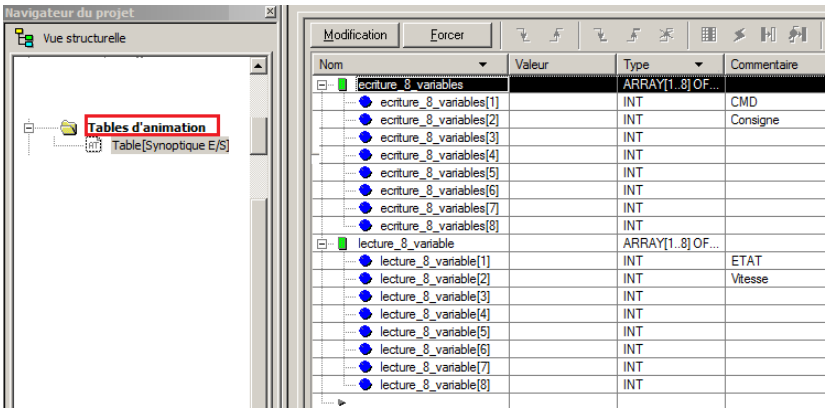
Code	Name	Logic address
CMD	Command register	16#2135 = 8501
CMI	Extended control word	16#2138 = 8504
RPR	Time counter reset	16#0C30 = 3120
LFRD	Speed setpoint	16#219A = 8602
LFR	Reference frequency	16#2136 = 8502
PISP	PIDSet Point	16#2137 = 8503
AIV1	AIV1 Image input	16#14A1 = 5281
MFR	Multiplying coefficient	16#2E37 = 11831
ETA	CIA402 State Register	16#0C81 = 3201
HMIS	Drive state	16#0CA8 = 3240
ETI	Internal State Register	16#0C86 = 3206
CRC	Channel for reference frequency	16#20F9 = 8441
CCC	Active command channel	16#20FA = 8442
CNFS	Active configuration	16#1F54 = 8020
STUN	Tune selection	16#2591 = 9617
SMOT	Status of motor tune in term of saliency	16#25AD = 9645
RFRD	Output velocity	16#219C = 8604
RFR	Motor Frequency	16#0C82 = 3202
OTR	Motor torque	16#0C85 = 3205

Il y a création de 4 variables dont deux sont les résultats des lectures/écriture :



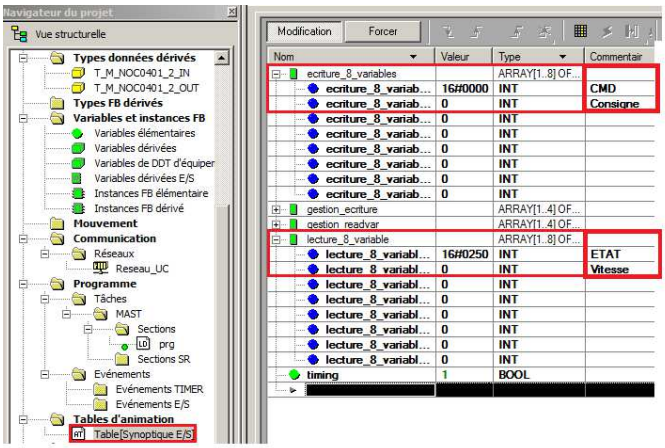
Il est ensuite nécessaire de bâtir la table d'animation :

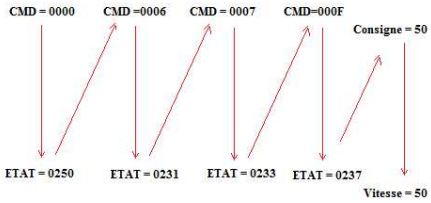




4-1-3 Test du programme lorsque le variateur est configuré en mode séparé ou non séparé

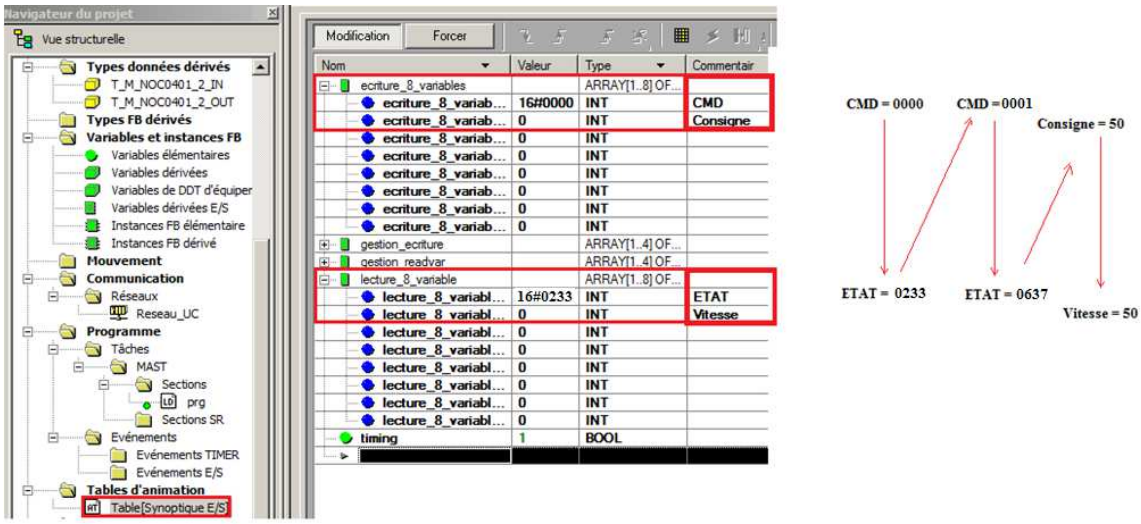
Après avoir établi la connexion, transféré le programme et mis en ''RUN', il s'agira de tester le ''DriveCom'' du variateur en affectant les valeurs en hexa de 6,7 et F et en vérifiant si les réponses du variateur sont correctes en hexa : 255, 231, 233 et 237.



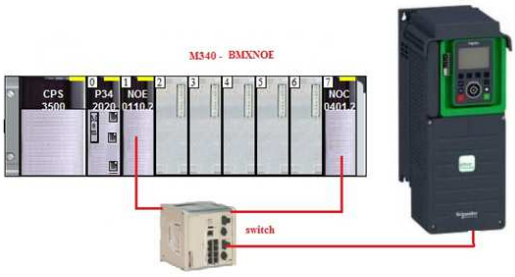


4-1-4 Test du programme lorsque le variateur est configuré en profil IO

Il suffira de positionner CMD à 1 pour démarrer.



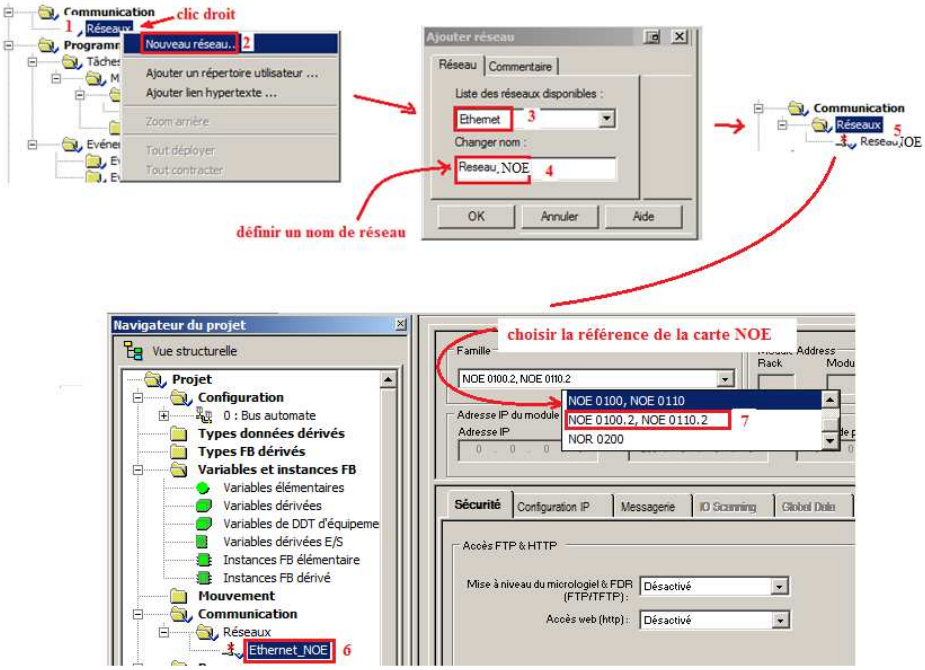
4-2 Liaison Ethernet à partir du coupleur NOE

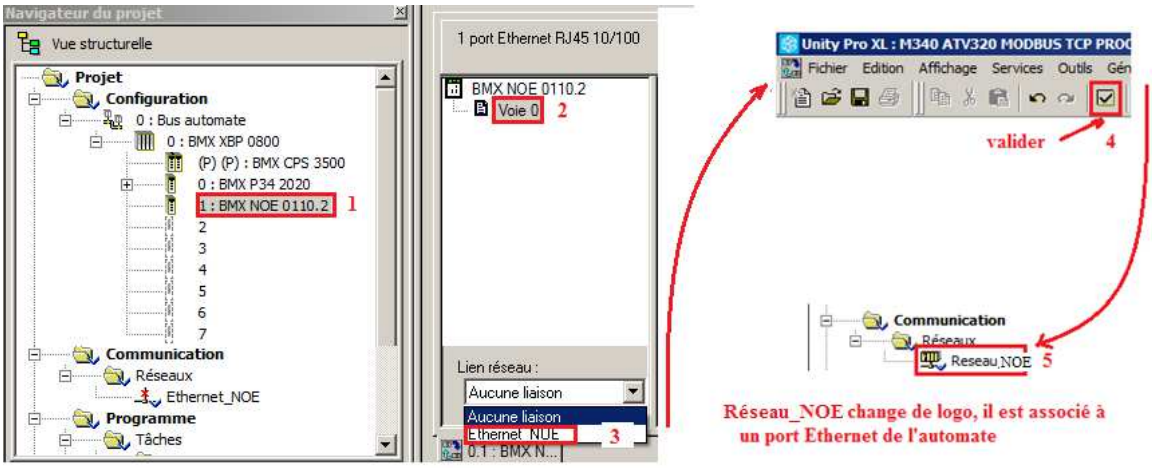


Configuration M340 - Altivar Process ATV600 - ATV900
Ethernet TCP Modbus

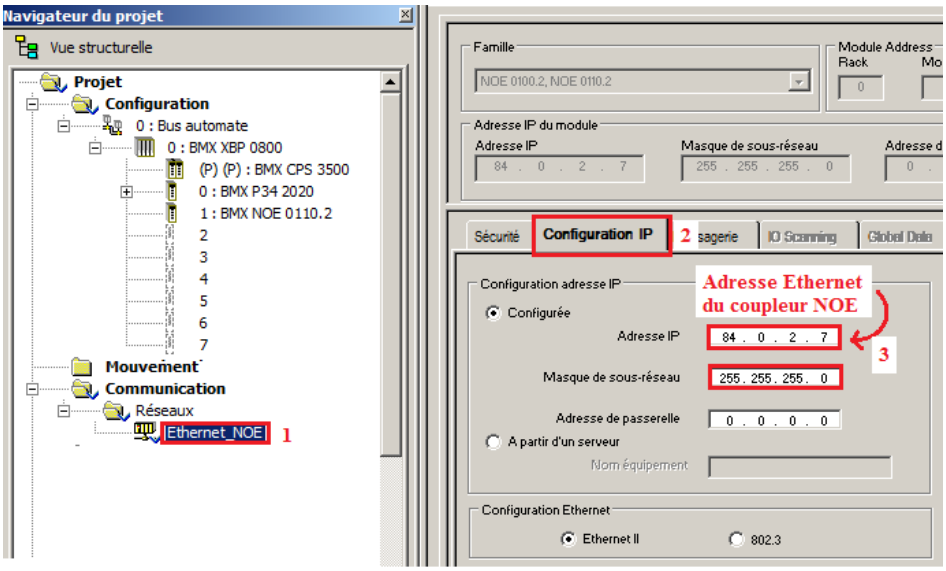
Le coupleur NOE utilise l'outil I/O Scanning natif sur Unity, il évite toute programmation pour seulement établir une table de dialogue en quelques clics.

Configuration du port Ethernet de la carte NOE

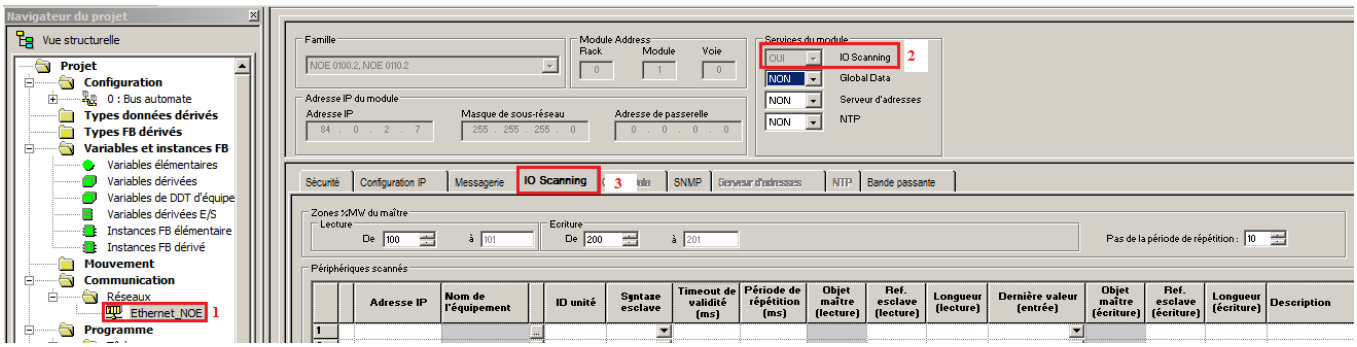




Configuration de l'adresse Ethernet du coupleur NOE (84.0.2.7)



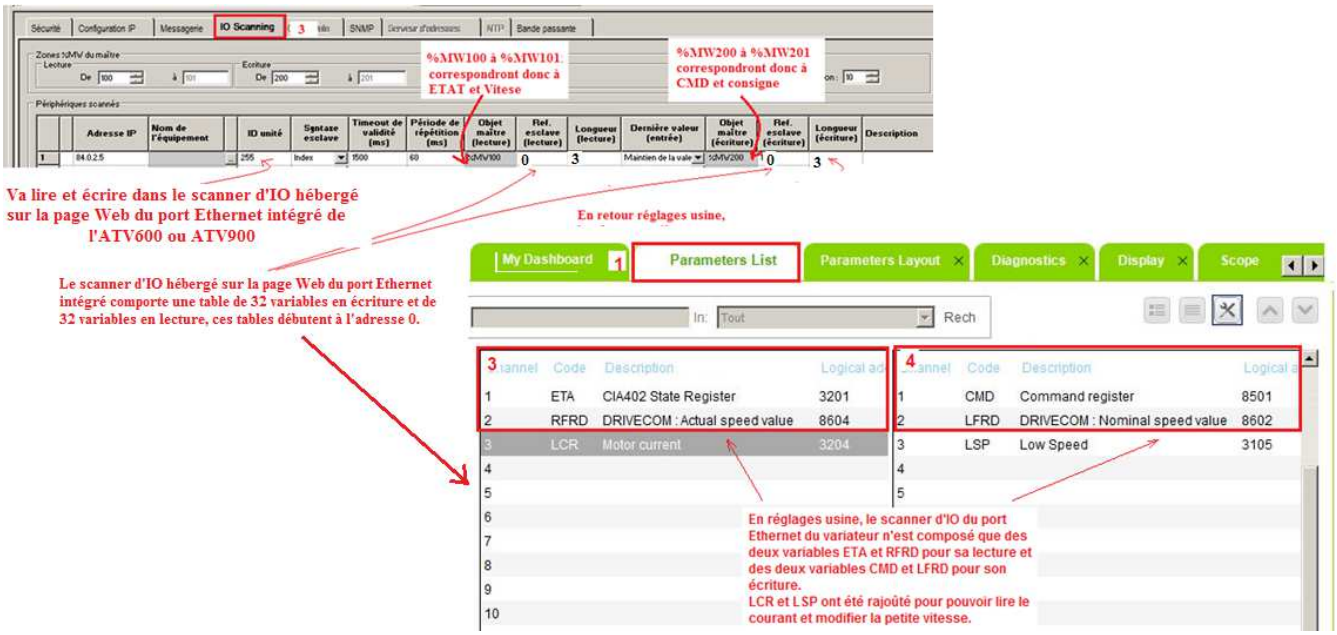
Utilisation de l'outil I/O Scanning natif Sur UNITY pour la configuration de la requête



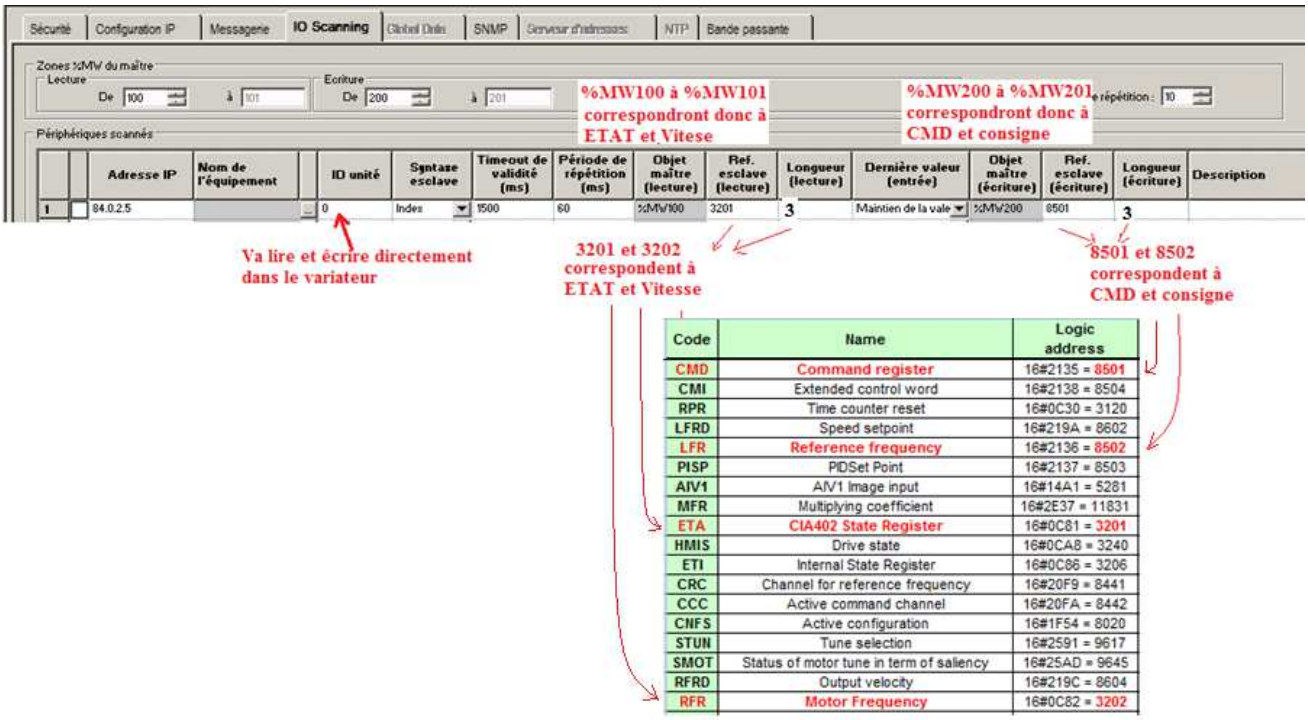
Le coupleur NOE permet de lire et écrire dans le variateur ATV6x0 ou ATV9x0 de deux façons différentes :

- par utilisation du scanner d'IO hébergé sur la page Web du port Ethernet intégré,
- hors scanner d'IO, en adressant directement les variables du variateur.

4-2-1 Communication à partir du scanner d'IO hébergé par le serveur Web du port Ethernet intégré

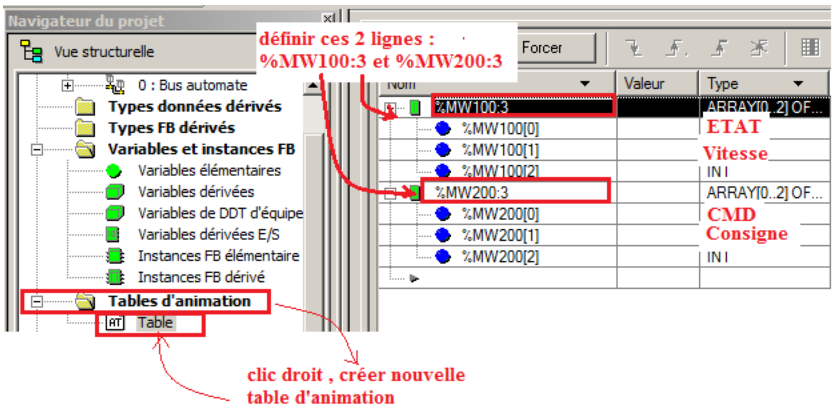


4-2-2 Communication sans scanner d'I0 directement à partir du tableau des variables automates



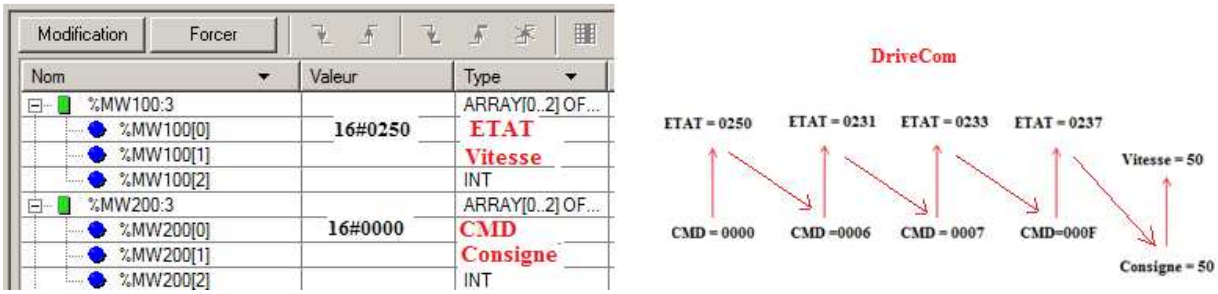
Code	Name	Logic address
CMD	Command register	16#2135 = 8501
CMI	Extended control word	16#2138 = 8504
RPR	Time counter reset	16#0C30 = 3120
LFRD	Speed setpoint	16#219A = 8602
LFR	Reference frequency	16#2136 = 8502
PISP	PIDSet Point	16#2137 = 8503
AIV1	AIV1 Image input	16#14A1 = 5281
MFR	Multiplying coefficient	16#2E37 = 11831
ETA	CIA402 State Register	16#0C81 = 3201
HMIS	Drive state	16#0CA8 = 3240
ETI	Internal State Register	16#0C86 = 3206
CRC	Channel for reference frequency	16#20F9 = 8441
CCC	Active command channel	16#20FA = 8442
CNFS	Active configuration	16#1F54 = 8020
STUN	Tune selection	16#2591 = 9617
SMOT	Status of motor tune in term of saliency	16#25AD = 9645
RFRD	Output velocity	16#219C = 8604
RFR	Motor Frequency	16#0C82 = 3202

Création de la table d'animation

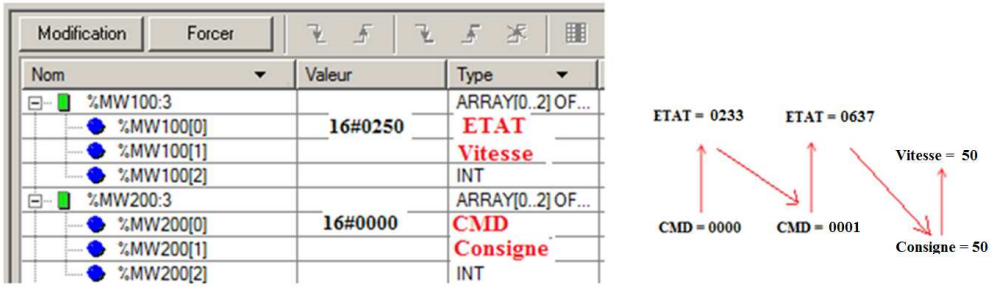


4-2-3 Test du programme lorsque le variateur est configuré en mode séparé ou non séparé

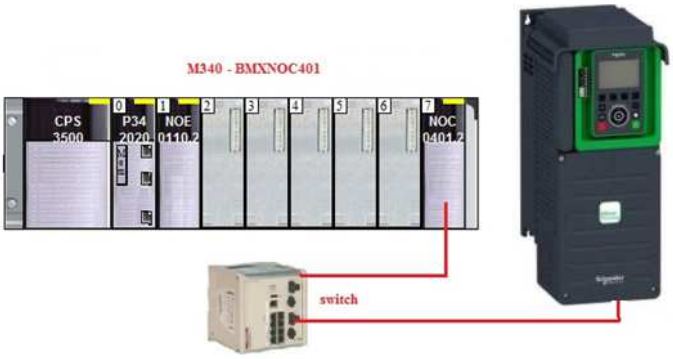
Après avoir établi la connexion, transféré le programme et mis en ''RUN', il s'agira de tester le ''DriveCom'' du variateur.



4-2-4 Test du programme lorsque le variateur est configuré en profil IO



4-3 Liaison Ethernet à partir du coupleur BMXMOC401



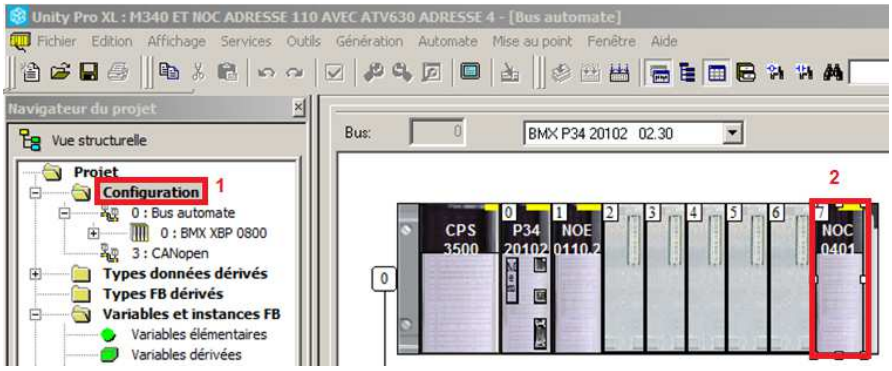
Configuration M340 - Altivar Process ATV600 - ATV900
Ethernet TCP Modbus

Le coupleur NOC401 est entièrement défini par un fichier DTM comme le sont les autres interlocuteurs comme l'ATV320, ATV600 et ATV900.
Ces fichiers DTMs sont rangés dans une bibliothèque où tous les logiciels bâtis sur le principe du FDT (container) peuvent aller les utiliser.

Dès que La NOC401 est définie dans la configuration automate via le ''navigateur de projet'', le DTM correspondant est chargé dans Unity et est accessible via le ''navigateur de DTMs''.

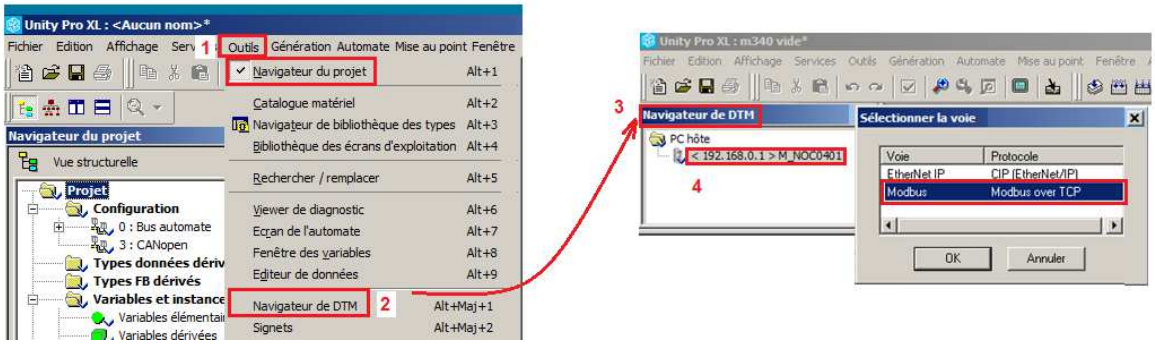
Le coupleur BMXNOC401 disposent de deux profils : ''Modbus over TCP'' et ''CIP''.

Navigateur de projet



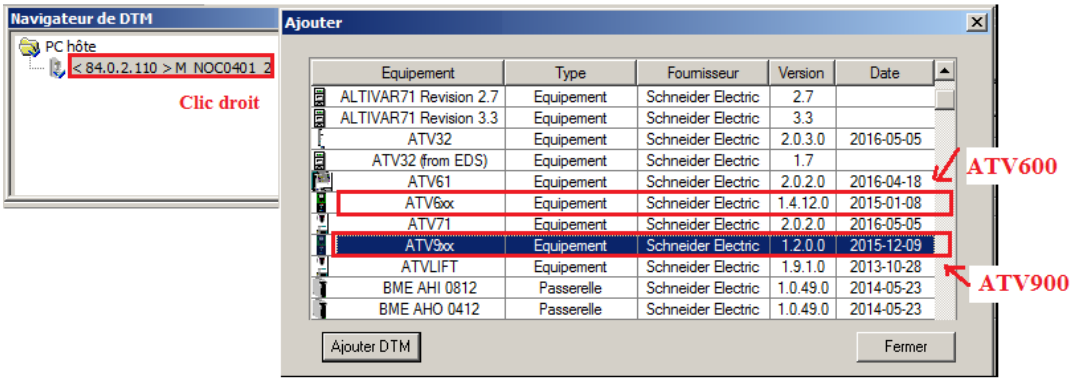
Navigateur de DTM

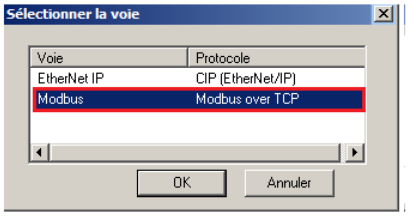
Unity attribue une adresse par défaut au coupleur NOC : 192.168.0.1, adresse qui sera modifiée par la suite.



Choisir dans la bibliothèque de DTMs incluse dans Unity V8, le DTM correspondant à ATV6x0 ou ATV9x0

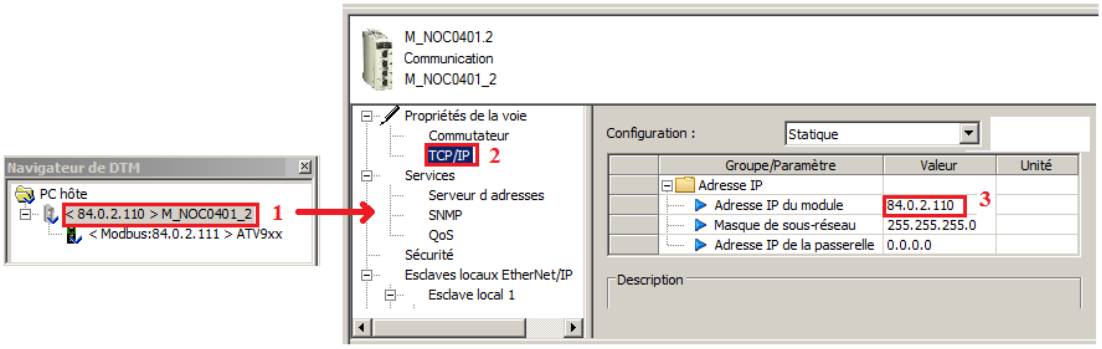
4-3-1 Utilisation du DTM "ATV9xx ou ATV6xx pour communication TCP Modbus"



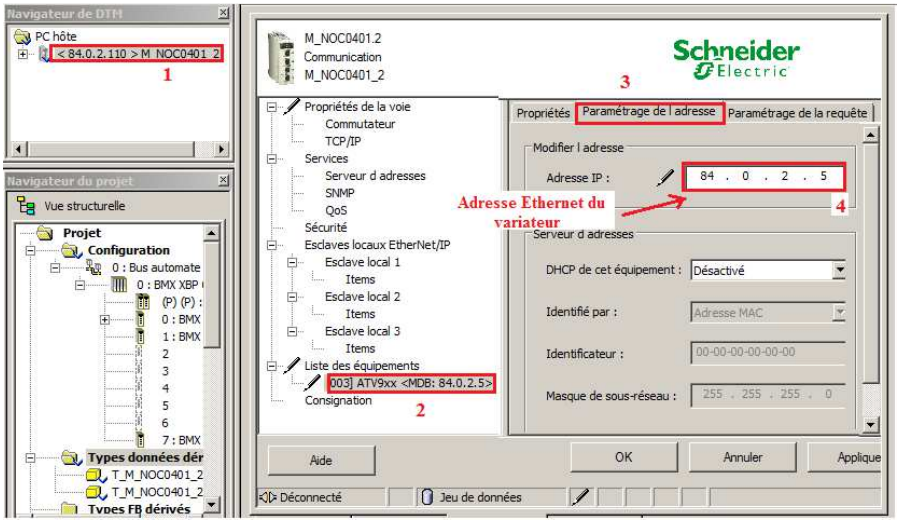


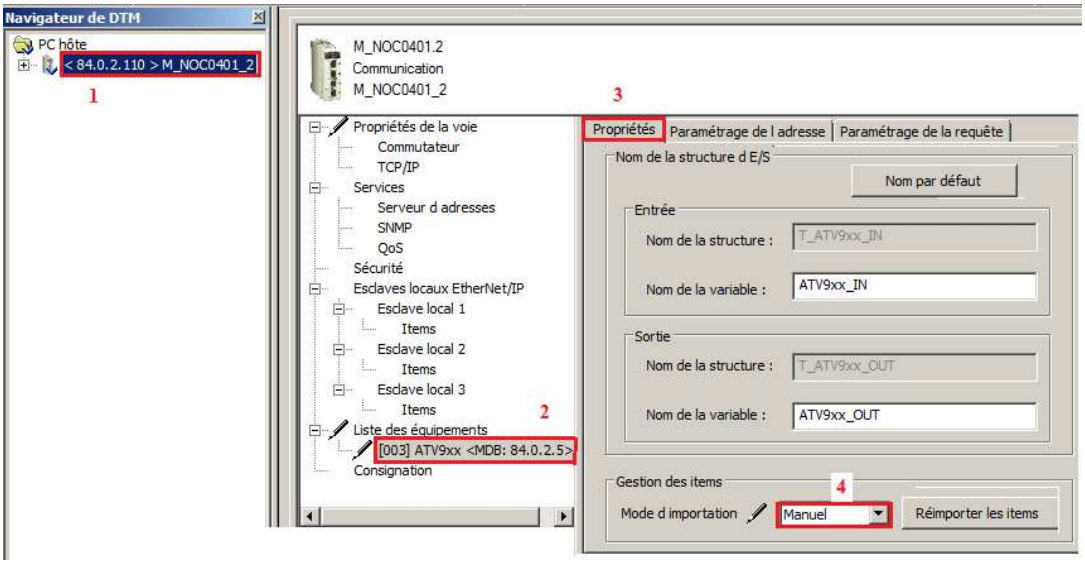
Lorsque les adresses proposées par défaut ne conviennent pas, il est nécessaire de définir :

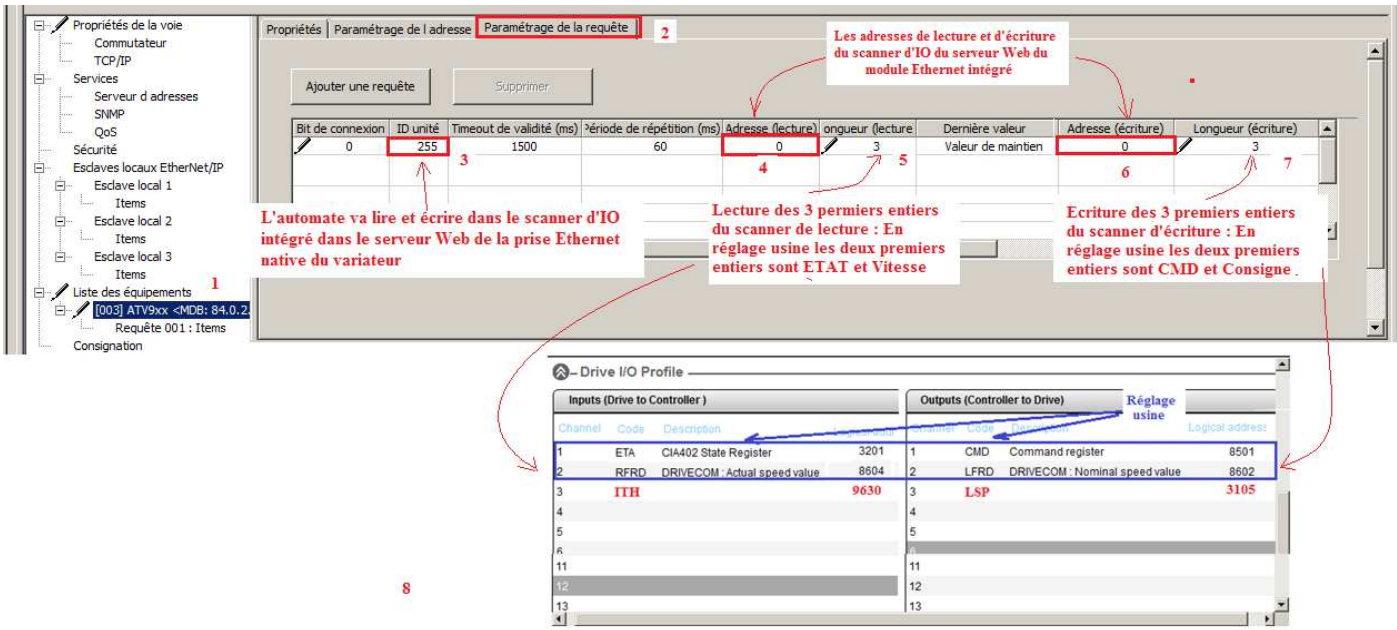
- les adresses IP réelles du coupleur NOC401 et de l'interlocuteur variateurATV6x0 ou ATV9x0
- la table d'entiers à lire ou écrire dans le variateur pour le faire fonctionner.



Ensuite, en cliquant sur <192.168.0.1> M_NOC401_2, il sera possible de définir l'adresse Ethernet réelle de l'Altivar, puis de définir la requête de lecture/écriture.



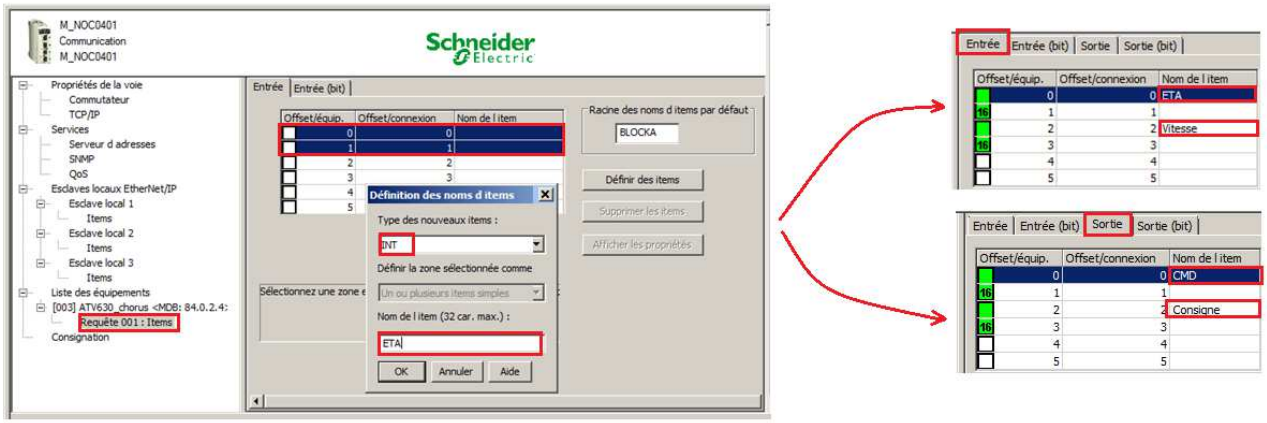




Le scanner d'IO du variateur sera accessible par le M340 comme suit :

Scanner IO lecture	**Scanner IO écriture,**
Adresse **0** du scanner de lecture :**ETA**	Adresse **0** du scanner d'écriture : **CMD,**
Adresse **1** du scanner de lecture :**RFRD**	Adresse **1** du scanner d'écriture : **LFRD.**

Les variables ETA, CMD, RFRD et LFRD sont des entiers (16 bits) hors ce qui est fourni par le DTM sont des octets, pour une meilleure compréhension, il est utile de fabriquer des entiers auxquels il est possible d'associer des variables.

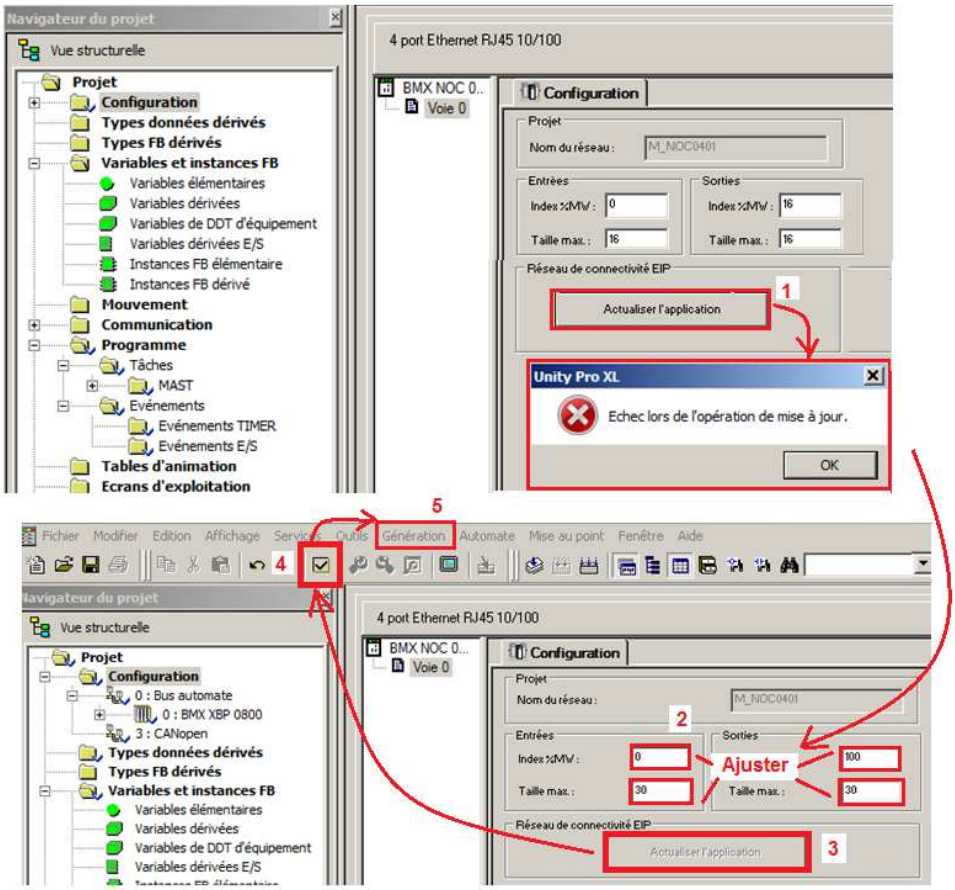


Le scanner d'IO est devenu :

Scanner IO lecture	**Scanner IO écriture,**
Adresse 0 du scanner de lecture :**ETA**	Adresse 0 du scanner d'écriture : **CMD,**
Adresse 1 du scanner de lecture :**Vitesse**	Adresse 1 du scanner d'écriture : **Consigne.**

Test de la liaison

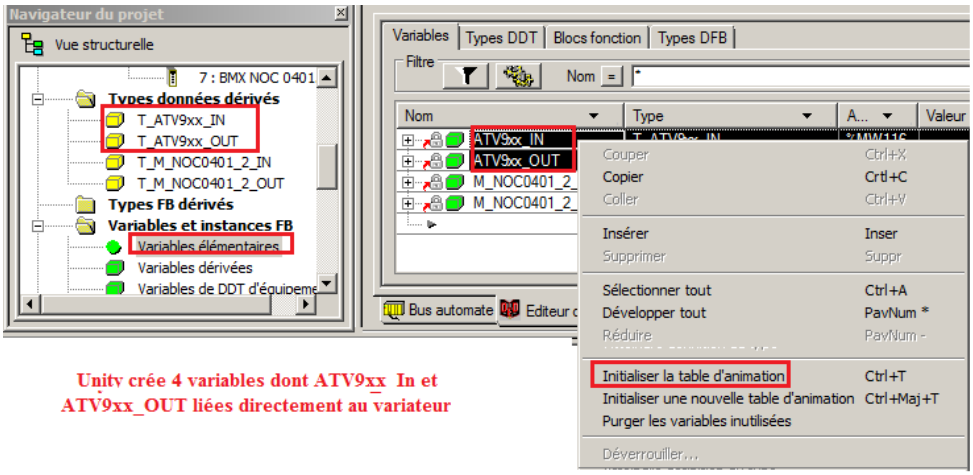
Revenir sous le navigateur de projet et ''actualiser l'application'' liée à la voie 0 de la NOC401, suivre la procédure détaillée dans la figure suivante.



Unity crée 4 variables visualisables sur le navigateur de projet, répertoire : Variables et instances FB / Variables élémentaires.

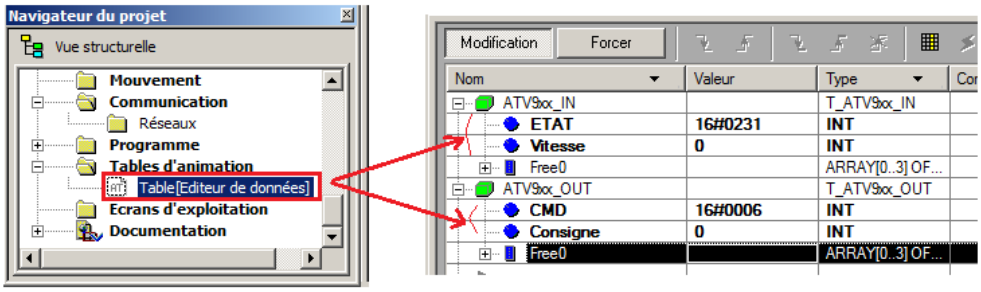
- deux pour la gestion de la NOC401,
- deux pour la mise en service du variateur ATV6x0 ou ATV9x0.

Pour tester les deux variables ATV6x0 ou ATV9x0, une table d'animation doit être créée. Sélectionner les deux variables, clic droit, commande ''Initialiser la table d'animation''.



Unity crée 4 variables dont ATV9xx_In et ATV9xx_OUT liées directement au variateur

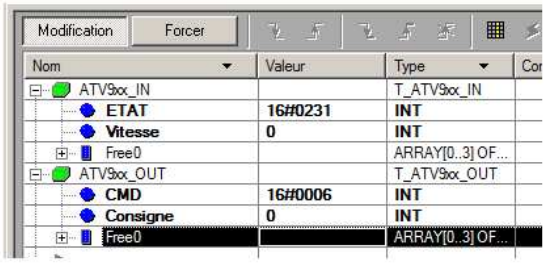
Il va être possible de tester les deux variables ATV9xx_IN et ATV9xx _OUT.

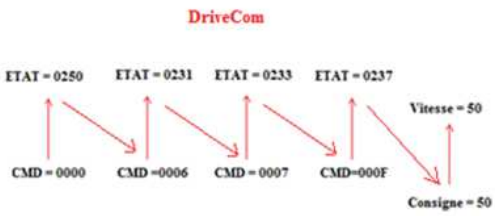


On retrouve les variables ETAT , Vitesse, CMD et Consigne déjà définies dans les Items

4-3-2 Test du programme lorsque le variateur est configuré en mode séparé ou non séparé

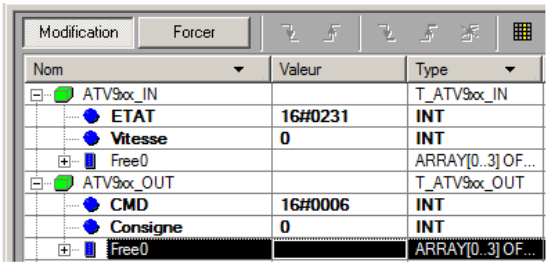
Ces variables doivent respecter le Protocole Drive COM 402 consultable en amont.

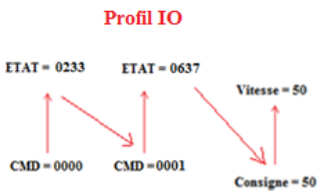




4-

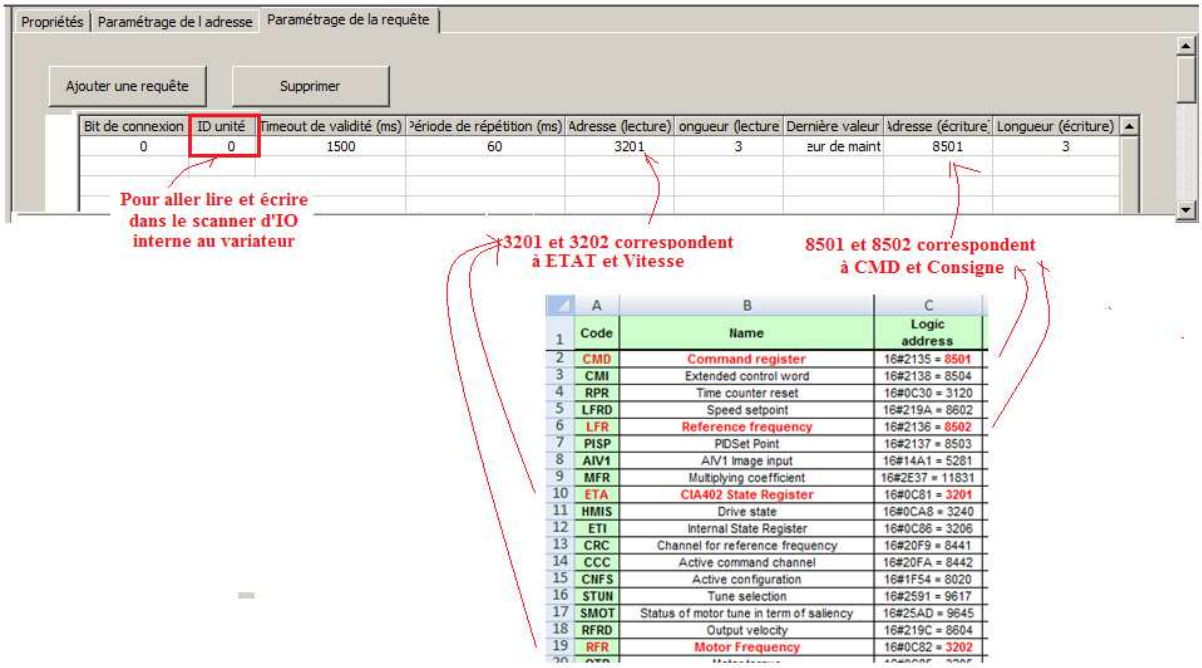
3-3 Test du programme lorsque le variateur est configuré en Profil IO





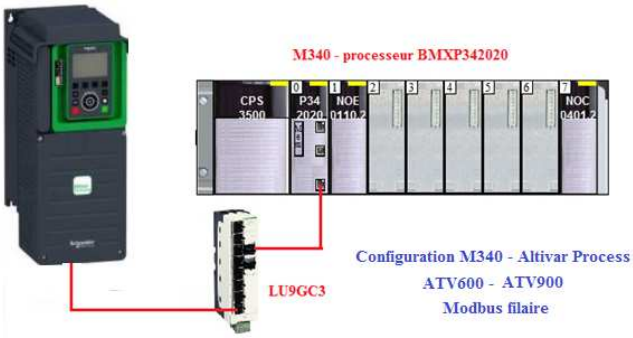
4-3-4 Utilisation de la NOC sans utilisation des scanners d'IO

Il est possible pour l'automate d'aller directement lire et écrire dans le variateur sans passer par les scanners d'IO.

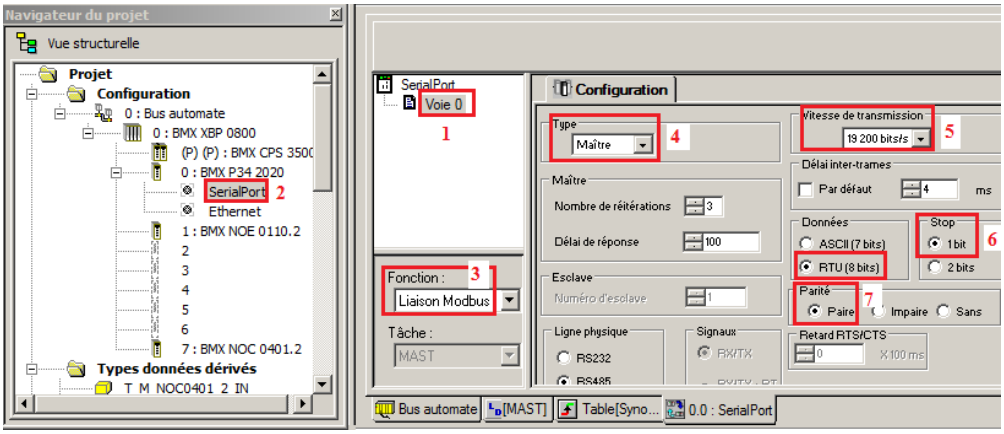


A part ce paramétrage de requête, la déclaration du variateur, la mise en œuvre et le test de cette configuration est rigoureusement identique au développement précédent (scanner d'IO du port Ethernet intégré)

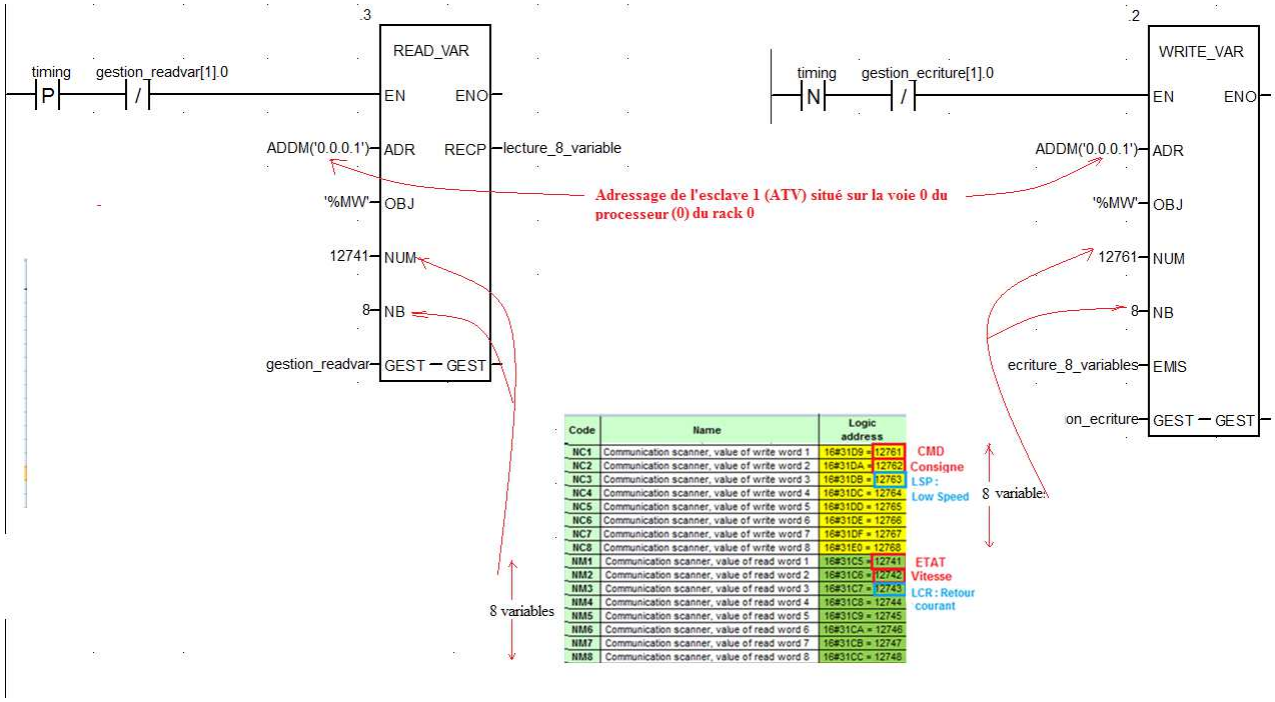
4-4 Liaison Modbus filaire ou RS485 à partir du port processeur



Configuration du coupleur

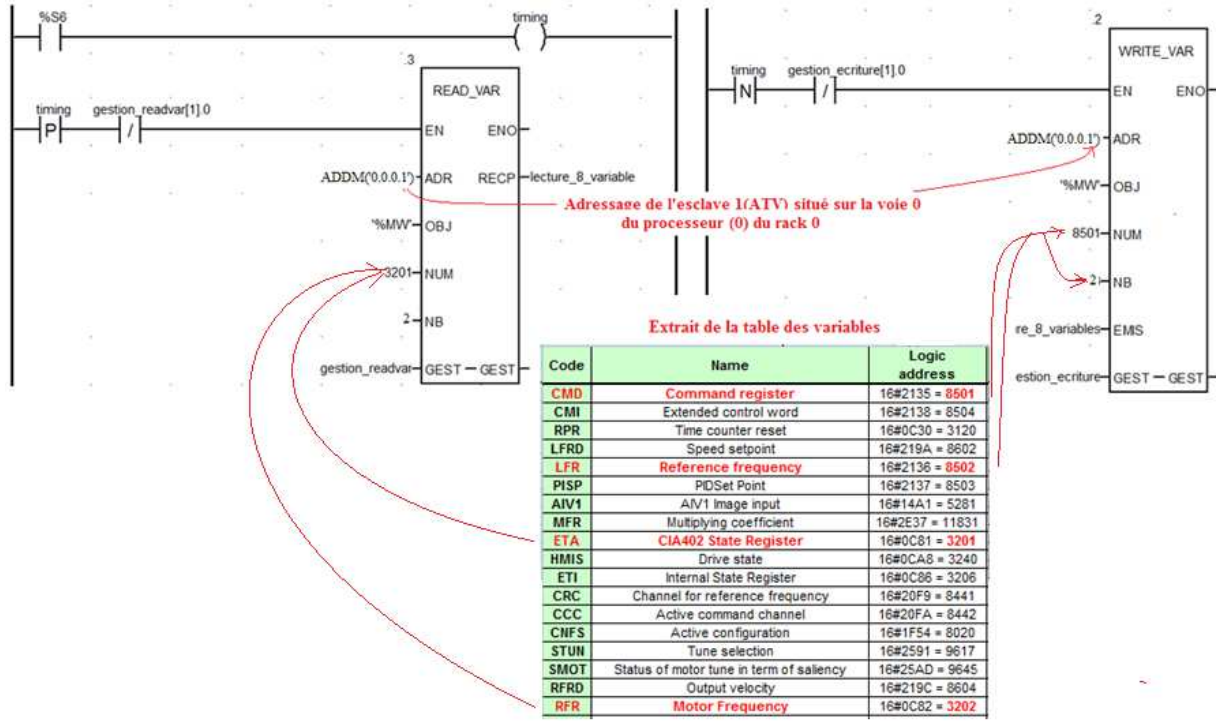


4-4-1 Communication à partir du scanner d'IO du variateur



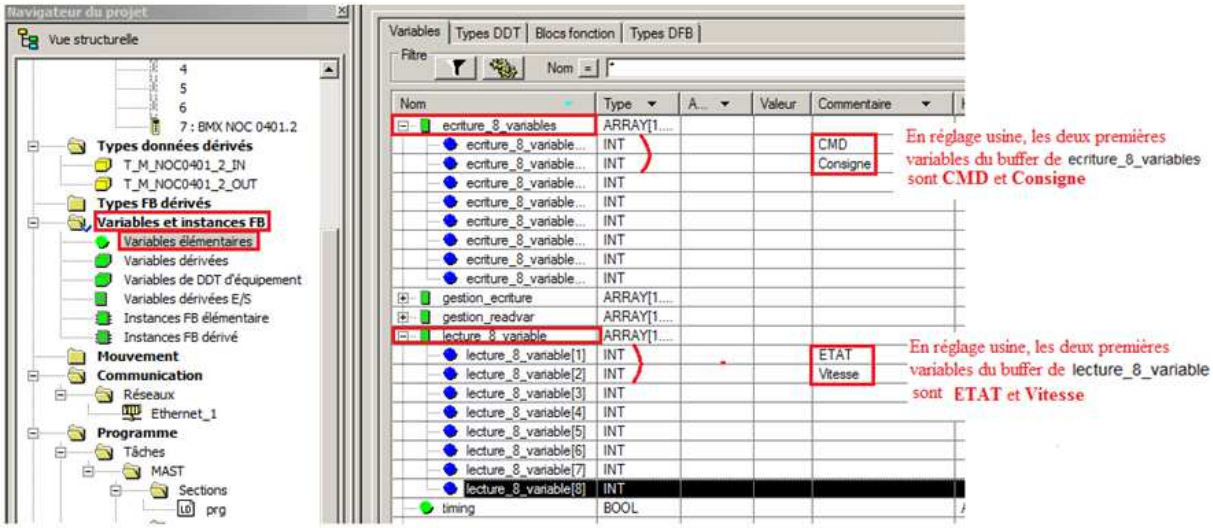
Code	Name	Logic address
NC1	Communication scanner, value of write word 1	16#31D9 = 12761
NC2	Communication scanner, value of write word 2	16#31DA = 12762
NC3	Communication scanner, value of write word 3	16#31DB = 12763
NC4	Communication scanner, value of write word 4	16#31DC = 12764
NC5	Communication scanner, value of write word 5	16#31DD = 12765
NC6	Communication scanner, value of write word 6	16#31DE = 12766
NC7	Communication scanner, value of write word 7	16#31DF = 12767
NC8	Communication scanner, value of write word 8	16#31E0 = 12768
NM1	Communication scanner, value of read word 1	16#31C5 = 12741
NM2	Communication scanner, value of read word 2	16#31C6 = 12742
NM3	Communication scanner, value of read word 3	16#31C7 = 12743
NM4	Communication scanner, value of read word 4	16#31C8 = 12744
NM5	Communication scanner, value of read word 5	16#31C9 = 12745
NM6	Communication scanner, value of read word 6	16#31CA = 12746
NM7	Communication scanner, value of read word 7	16#31CB = 12747
NM8	Communication scanner, value of read word 8	16#31CC = 12748

4-4-2 Communication sans scanner d'I0 directement à partir du tableau des variables automates

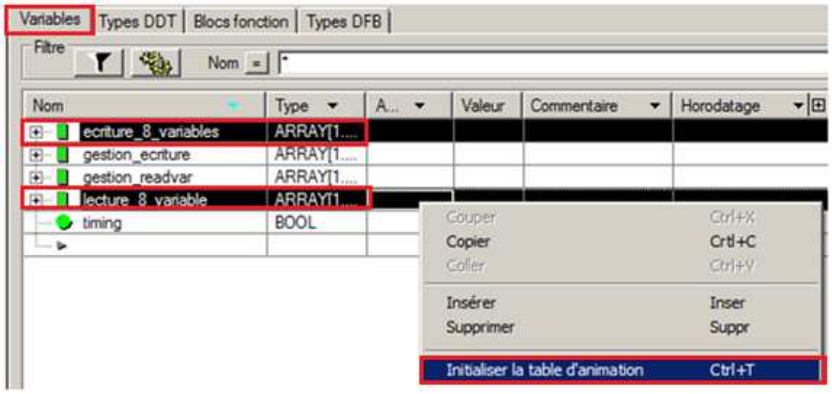


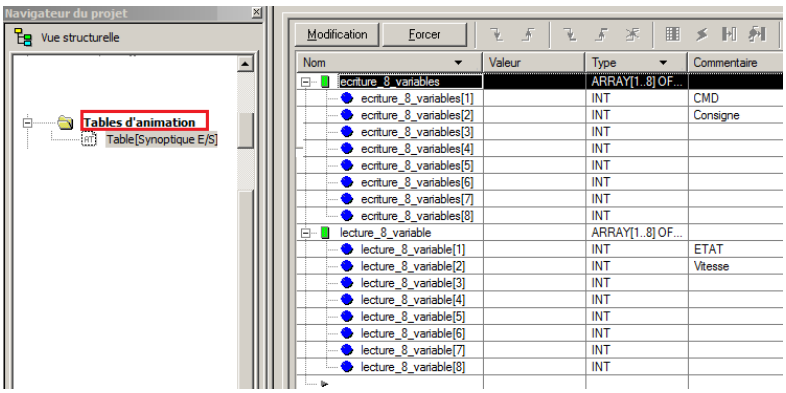
Code	Name	Logic address
CMD	Command register	16#2135 = 8501
CMI	Extended control word	16#2138 = 8504
RPR	Time counter reset	16#0C30 = 3120
LFRD	Speed setpoint	16#219A = 8602
LFR	Reference frequency	16#2136 = 8502
PISP	PIDSet Point	16#2137 = 8503
AIV1	AIV1 Image input	16#14A1 = 5281
MFR	Multiplying coefficient	16#2E37 = 11831
ETA	CIA402 State Register	16#0C81 = 3201
HMIS	Drive state	16#0CA8 = 3240
ETI	Internal State Register	16#0C86 = 3206
CRC	Channel for reference frequency	16#20F9 = 8441
CCC	Active command channel	16#20FA = 8442
CNFS	Active configuration	16#1F54 = 8020
STUN	Tune selection	16#2591 = 9617
SMOT	Status of motor tune in term of saliency	16#25AD = 9645
RFRD	Output velocity	16#219C = 8604
RFR	Motor Frequency	16#0C82 = 3202

Il y a création de 4 variables dont deux sont les résultats des lecture/écriture



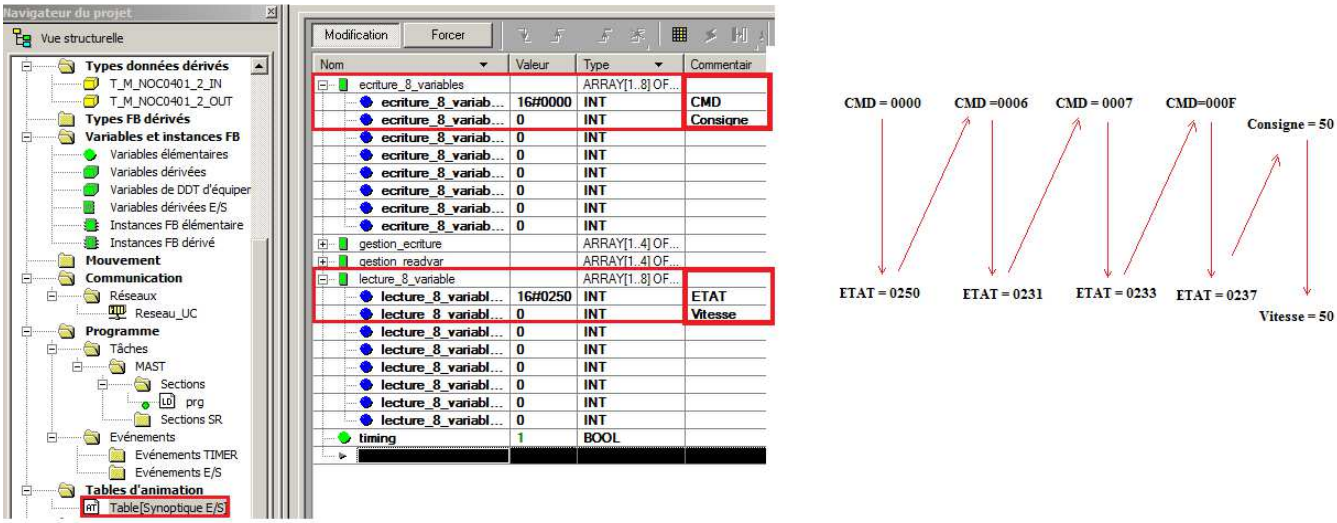
Il est ensuite nécessaire de bâtir la table d'animation



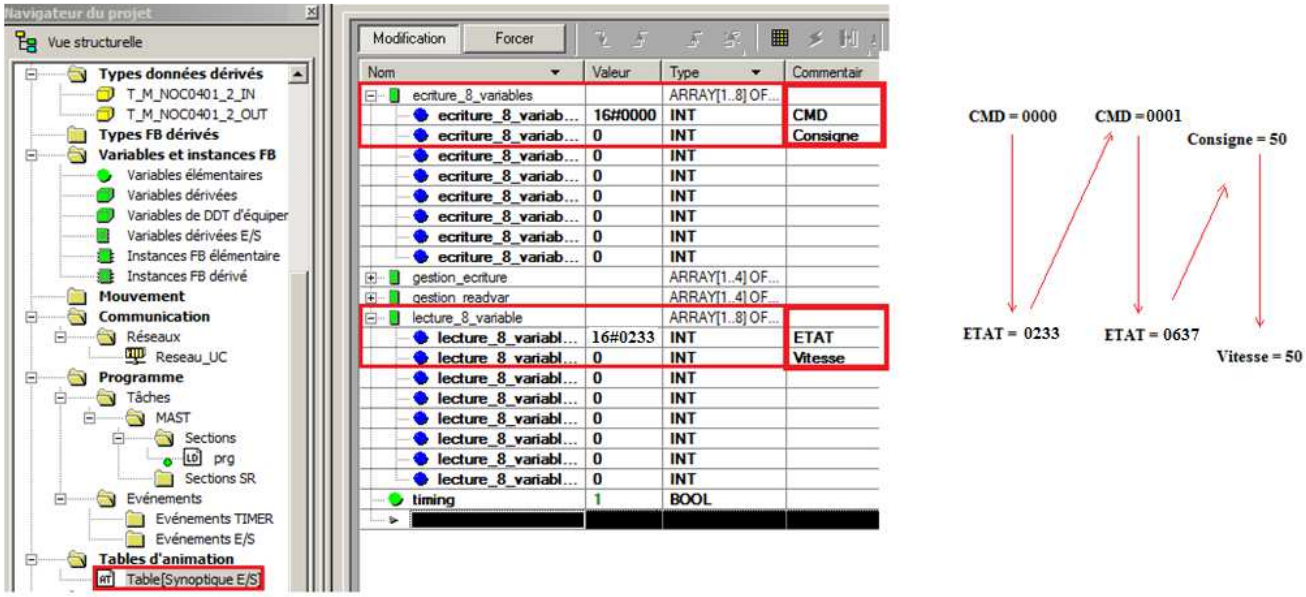


4-4-4 Test du programme lorsque le variateur est configuré en mode séparé ou non séparé

Après avoir établi la connexion, transféré le programme et mis en ''RUN', il s'agira de tester le ''DriveCom'' du variateur en affectant les valeurs en hexa de 6,7 et F et en vérifiant si les réponses du variateur sont correctes en hexa : 255, 231, 233 et 237



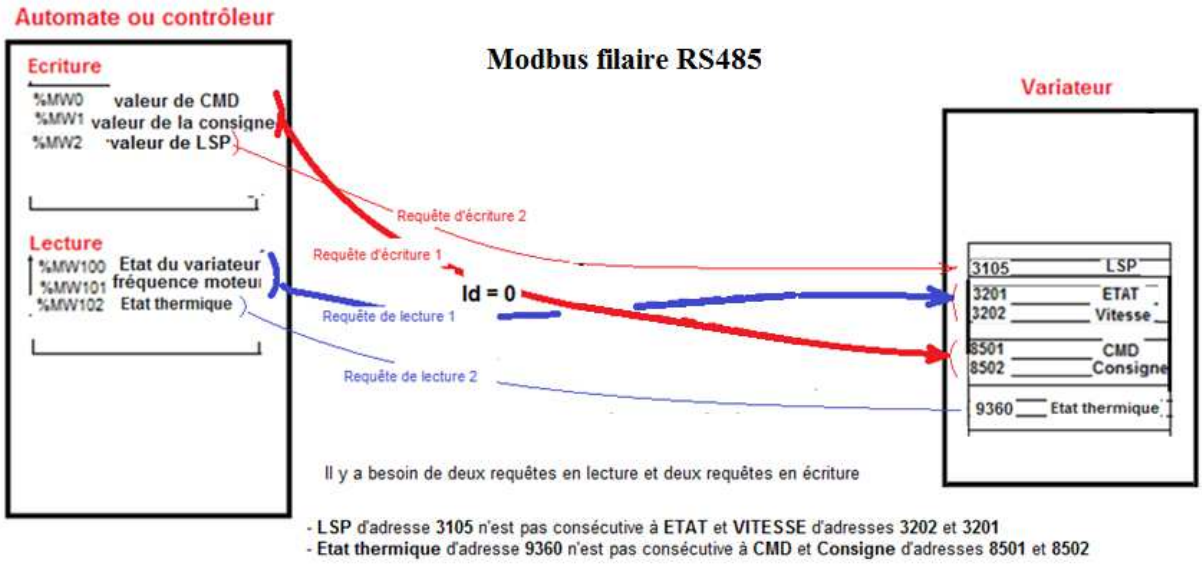
4-4-5 Test du programme lorsque le variateur est configuré en profil IO



5- Annexes – Détail des requêtes de lecture / écriture

Pour réaliser toute opération à distance via un réseau ou un bus, écrire des commandes et lire des états variateurs, Schneider a développé pour ses automates programmables (M340 , M580, M240, M221) des instructions ou des modules fonctionnels qui semblent de prime abord différents selon les langages et ateliers logiciels utilisés (Unity, SoMachine, SoMachinebasic) mais qui se révèlent n'être, en fin de compte, que des nuances différentes d'une requête de lecture/ écriture

5-1 Requête lecture / écriture sans utilisation des scanners d'IO



Il sera nécessaire d'utiliser deux requêtes de lecture/écriture, la première pour lire ETAT (3201) et RFRD (3202) et écrire CMD (8501) et Consigne (8502) parce que les adresses de ces variables se suivent et une deuxième requête pour lire ITH (9360) et écrire LSP(3105).

Détail de la requête

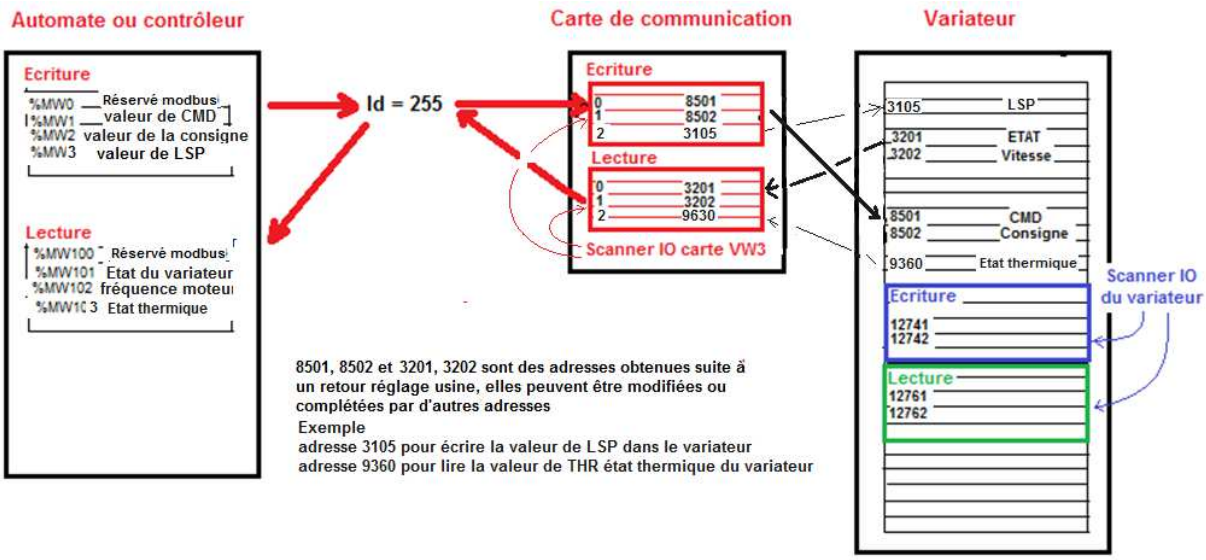
- Id =0: aiguillage , les requêtes de lecture/écriture sont routées directement aux adresses variateur
- Adresse de début de lecture de lecture et adresse du début d'écriture
  - Requête 1 : (pour lire ETAT et RFRD et écrire CMD et consigne)
    - adresse de lecture 3201 (ETAT)
    - adresse d'écriture 8501 (CMD)
    - nombre de variables en lecture et nombre de variable en écriture nécessaires à l'application, dans l'exemple ci-dessus
      - nombre de variables en lecture nécessaires à l'application de ETA et RFRD ➔ 2
      - nombre de variable en écriture nécessaires à l'application CMD et LFRD ➔ 2
    - les adresses des variables automate qui écriront dans la variateur sont %MW0, %MW1 soit une table de 2 paramètres débutant par %MW0
    - les adresses des variables automate qui liront dans la variateur sont %MW100, %MW101, soit une table de 2 paramètres débutant par %MW100
  - Requête 2 : (pour lire ITH et écrire LSP)
    - Adresse de lecture ITH 9360
    - Adresse d'écriture LSP 3105
    - Les nombres de variables en lecture et écriture nécessaires à l'application sont 1 et 1
    - Les adresses de variables qui écriront dans le variateur sont %MW10 soit une table de un paramètre
    - Les adresses de variable automate qui liront dans le variateur sont %MW110 soit une table de un paramètre

5-2 Requête de lecture/écriture avec utilisation du scanner d'IO issu d'une page WEB

Cette requête va lire ETAT, RFRD, ITH et écrire CMD, consigne et LSP à partir du scanner d'IO de la page Web associée au port ethernet
Attention ! les paramètres ETAT, RFRD et CMD,consigne sont positionnés en retour réglage usine, ITH et LSP doivent être positionné volontairement et directement en se connectant sur la page Web du port Ethernet ou par utilisation de SoMove

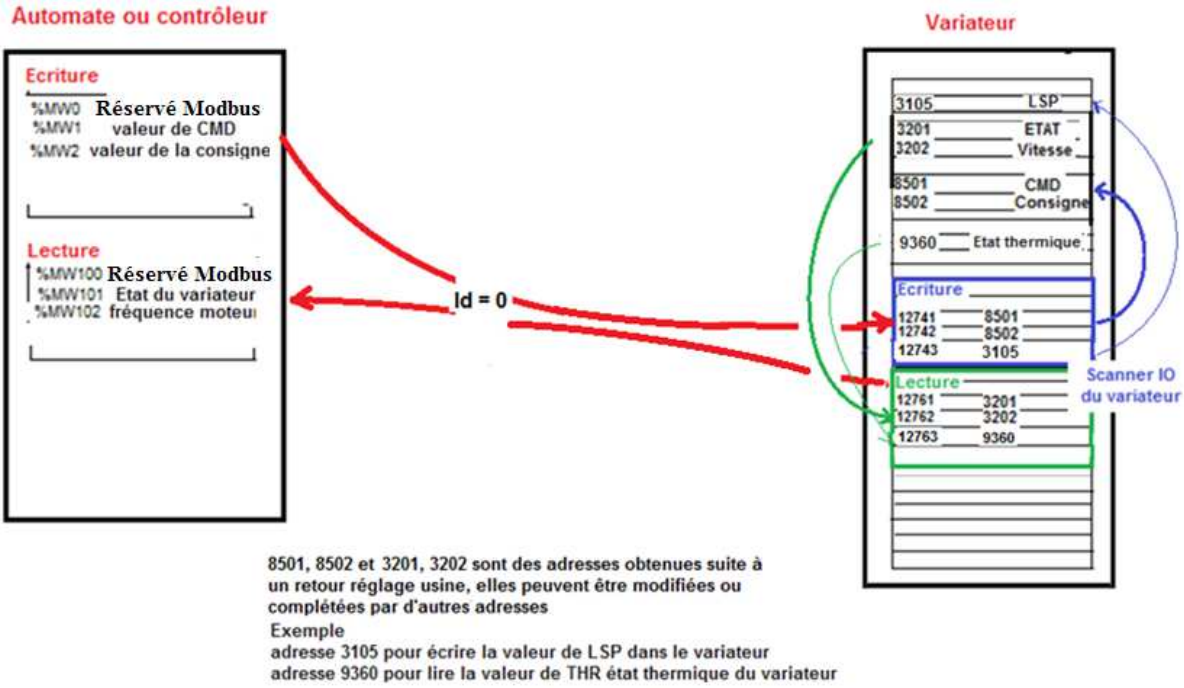
Détail de la requête

- Id =255: aiguillage , les requêtes de lecture/écriture sont routées directement aux adresses du scanner d'IO en lecture et écriture
- Adresse de début de lecture de lecture et adresse du début d'écriture
    - adresse de lecture 0 (ETAT)
    - adresse d'écriture 0 (CMD)
    - nombre de variables en lecture et nombre de variable en écriture nécessaires à l'application, dans l'exemple ci-dessus
        - nombre de variables en lecture nécessaires à l'application de ETA, RFRD et ITH➔ 3
        - nombre de variable en écriture nécessaires à l'application CMD et LFRD et LSP ➔ 3
    - les adresses des variables automate qui écriront dans la variateur sont %MW0, %MW1 et %MW2 soit une table de 2 paramètres débutant par %MW0
    - les adresses des variables automate qui liront dans la variateur sont %MW100, %MW101 et %MW102, soit une table de 3 paramètres débutant par %MW100
    -



5-3 Requête de lecture/écriture avec utilisation du scanner d'IO issue de la mémoire interne du variateur

Cette requête va lire ETAT, RFRD, ITH et écrire CMD, consigne et LSP.
Attention ! les paramètres ETAT, RFRD et CMD, consigne sont positionnés en retour réglage usine, ITH et LSP doivent être positionné volontairement et directement soit par l'afficheur

Détail de la requête

- Id =0: aiguillage , les requêtes de lecture/écriture sont routées directement aux adresses du scanner d'IO de en lecture et écriture
- Adresse de début de lecture de lecture et adresse du début d'écriture
  - adresse de lecture 12741 (ETAT)
  - adresse d'écriture 12761 (CMD)
  - nombre de variables en lecture et nombre de variable en écriture nécessaires à l'application, dans l'exemple ci-dessus
    - nombre de variables en lecture nécessaires à l'application de ETA, RFRD et ITH➔ 3
    - nombre de variable en écriture nécessaires à l'application CMD et LFRD et LSP ➔ 3
  - les adresses des variables automate qui écriront dans la variateur sont %MW0, %MW1 et %MW2 soit une table de 2 paramètres débutant par %MW0
  - les adresses des variables automate qui liront dans la variateur sont %MW100, %MW101 et %MW102, soit une table de 3 paramètres débutant par %MW100