

BIBLIOTHEQUE OBJET PESAGE



SOMMAIRE

GENERALITES.....	3
LE COMPOSANT ENOD4-T.....	4
DEFINITION DES FONCTIONS DU DFB eNOD4-T:.....	5
BUS CAN	6
GENERALITE SUR LA COMMUNICATION PAR BUS CAN :	6
<i>Structure en bus Can direct :</i>	6
<i>Structure en bus Can derrière STB :</i>	6
<i>Installation de la bibliothèque Scaime pour le bus CAN :</i>	7
EXPLOITATION EN BUS CAN SOUS AUTOMATE M340:.....	8
<i>Déclaration et configuration du bus CAN sous Unity :</i>	8
<i>Configuration de module de pesage Scaime sous le bus CAN :</i>	11
EXPLOITATION DERRIERE UN ILOT STB:.....	14
<i>Configuration automate sous Unity :</i>	14
<i>Configuration d'ilot sous Advantys :</i>	15
DFB COMPLET	17
REPRESENTATION DU DFB SOUS UNITY :	17
ENTREES DU DFB	18
SORTIES DU DFB	18
ENTREES SORTIES DU DFB	19
VARIABLE PRIVE DU DFB :	22
SECTIONS PROGRAMME DU DFB :	22
CYCLE SPECIFIQUE AU DFB :	23
<i>Etalonnage :</i>	23
<i>Portée :</i>	23
<i>Echelon :</i>	23
DFB SIMPLIFIE.....	24
REPRESENTATION DU DFB SOUS UNITY :	24
ENTREES DU DFB	25
SORTIES DU DFB	25
ENTREES SORTIES DU DFB	26
VARIABLE PRIVE DU DFB :	29
SECTIONS PROGRAMME DU DFB :	29
PROGRAMME EXEMPLE	30
EXEMPLE D'INSTANCE POUR DFB COMPLET EN CAN AUTOMATE.....	31
EXEMPLE D'INSTANCE POUR DFB LIGHT EN CAN AUTOMATE	32
EXEMPLE D'INSTANCE POUR DFB COMPLET EN COM DERRIERE STB.....	33
EXEMPLE D'INSTANCE POUR DFB LIGHT EN COM DERRIERE STB	34
DFB DE COMMUNICATION POUR BUS CAN AUTOMATE M340	35
DFB DE COMMUNICATION POUR I/O SCANNING VERS ILOT STB.....	35
UTILISATION DU COMPOSANT SOUS IHM VIJEODESIGNER.....	36
CONSTRUCTION DES VUES.....	36
<i>Import dans la bibliothèque :</i>	36
<i>Import des objets de la bibliothèque :</i>	36
<i>Variable :</i>	36
<i>Description des vues :</i>	37

Généralités

La DLF pesage a été développé avec les logiciels suivants :

- Unity V6.0
- Vijéo Designer V6.0
- Advantys V5.5.0.1

Préambule :

- Intégration du fichier constructeur pour la configuration du bus CAN de l'eNod4t pour Unity (eNod4T.cpx) ou (eNod4-T_EDS_196730-J.eds)
- Import des objets pesage dans VijéoDesigner (PesageScaime.ztc)

Exemple applicatif :

- Pour Unity, le fichier est : DemoEnod_1p0.Stu ou .Sta
- Pour Vijéo Designer, le fichier est : DemoEnod_1p0.vdz

Configuration :

Unity doit être configuré avec des options de projet suivant :

- Utiliser les tableaux dynamiques
- Affectation multiple en cascade

Description :

- DFB Complet : Permet d'exploiter l'appareil eNod4t avec les fonctions d'étalonnage, de réglage de portée et d'échelon. Des informations système sont disponibles.
- DFB Light : Permet d'exploiter les mesures de l'appareil eNod4t (Pas de possibilité de configuration ou d'étalonnage de l'appareil).
- Pour toutes commandes évoluées de réglage de l'appareil, se reporter au logiciel du constructeur Scaime eNodView

Le composant eNod4-T

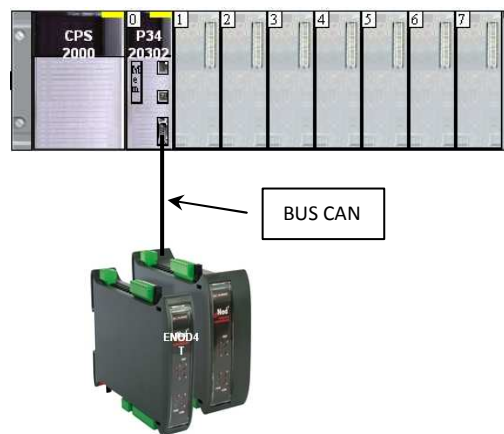
Ce composant permet d'exploiter un bloc transmetteur numérique de pesage de type SCAIME eNod4-T.

Ce bloc est déployé sous bus CANOpen. Ce bus doit être préalablement configuré pour exploiter le DFB proposé.

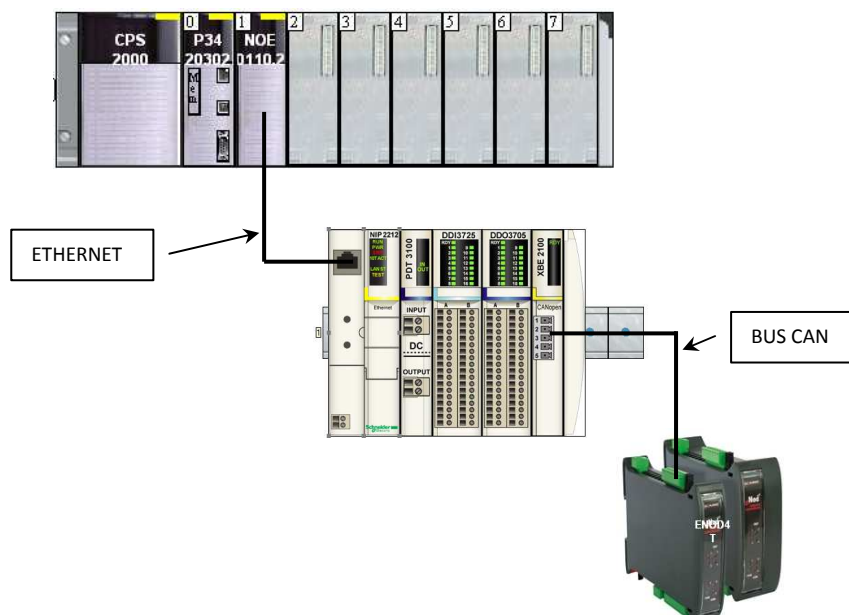
Une partie de la communication de ce DFB repose sur l'intégration de bloc Communication de la bibliothèque DPL.

Deux architectures sont proposées :

- Déploiement du transmetteur eNod4t sous un bus CAN directement géré par un automate type M340



- Déploiement du transmetteur eNod4t sous un bus CAN d'un module STB vu en I/O scanning sous Ethernet par un automate type M340



Définition des fonctions du DFB eNod4-T:

Fonction	Description
Contrôle	Permet la gestion d'un transmetteur de pesage exploité sur bus CAN derrière un automate M340 ou derrière un module STB connecté en Ethernet sur un automate M340.
Mesures	Permet d'afficher les mesures net et brute de l'appareil
Tarage	Permet de gérer des fonctions process de tarage et d'annulation de tare.
Programme/Operateur (Auto/Manu)	Commande en mode Programme (Automatique) ou en mode Opérateur (Manuel) à travers la supervision (Signal OWNER) Permet de saisir la mesure NET en mode manuel.
Etalonnage (*)	Permet de calibrer l'appareil avec une masse étalon.
Configuration (*)	Permet de définir la portée et l'échelon de travail de l'appareil
Calage du zéro	Permet de mettre à zéro les mesures de l'appareil
Simulation	Le transmetteur peut être simulé pour permettre de travailler sans sa présence. Permet de saisir le poids brut, la tare et de commander les fonctions de tarage, RAZ tare et RAZ bascule. Pas d'alarme de générée dans ce mode.
OK	Gestion d'un signal permettant d'indiquer si l'actionneur est dans une position utilisable pour un process.
Densité	Permet de gérer l'affichage NET suivant une conversion de densité. Par défaut égale à 1.0
Mise hors service	Désactive la gestion du bloc. Signal OOS
Système (*)	Permet d'afficher le versionning de l'appareil
Alarmes	Gestion de 4 seuils d'alarmes sur la mesure NET

(*) Disponible uniquement avec le DFB complet

Remarque : Les unités et échelle d'affichages des valeurs sont à prendre en considération avec le système d'affichage utilisé. L'exploitation de L'eNod est réalisée en point du convertisseur suivant la portée de la chaine de mesure sans conversion ni mise à l'échelle.

Toute mise à l'échelle ou décalage de virgule dans l'affichage doit être réalisé dans le scada.

Bus CAN

Généralité sur la communication par bus CAN :

La communication sous bus CAN est disponible suivant deux mécanismes:

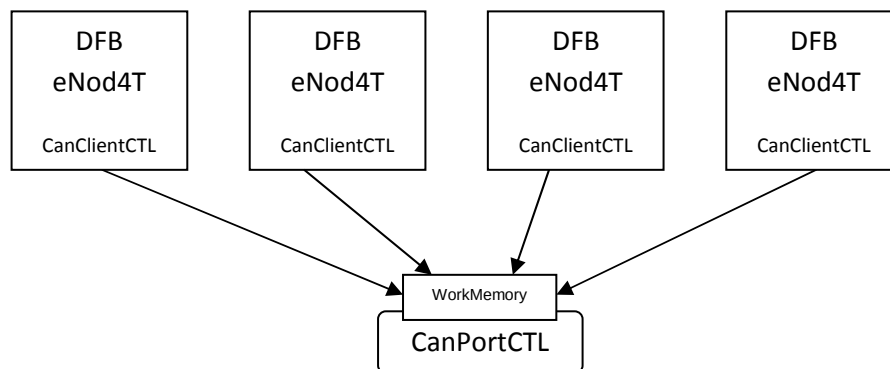
-Les échanges implicites PDO (informations configurées pour être naturellement échangées sur le bus CAN sans code automate). Ces informations sont rafraichies suivant le rythme de la tâche automate comme des entrées sorties. C'est le fichier .EDS du constructeur qui définit les variables mises à disposition pour ce type d'échange. Ce mode de communication est utilisé pour le DFB Light décrit ci-après.

-Les échanges explicites SDO (Trame d'informations construites en code automate). Ces informations sont déclenchées par l'exécution d'instructions d'émission/réception par le code l'automate. Toutes les variables d'un l'équipement Can sont accessibles avec ce type d'échange. Le DFB Complet utilise ces deux mécanismes.

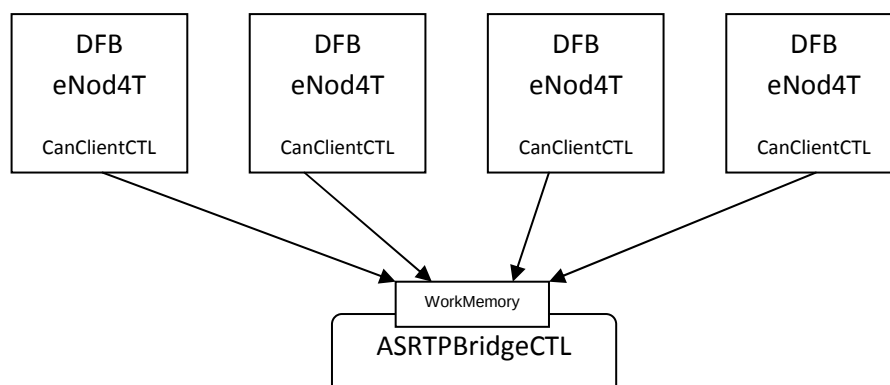
Que ce soit en bus Can direct ou derrière un STB, ces deux modes d'échanges sont exploités.

Le DFB Complet proposé repose sur l'exploitation des blocs de communication de la bibliothèque DPL. Pour ce faire, le DFB encapsule un bloc « CanClientCTL ». Il y a donc autant de bloc CanClientCTL qu'il y a de bloc eNod4T instancié. Ces blocs « CanClientCTL » doivent être couplés avec un seul bloc de communication vers l'extérieur. En bus Can direct, c'est le bloc « CanPortCTL », et en I/O scanning vers STB, c'est le bloc « ASRTPBridgeCTL ». Le lien entre ces blocs est une variable tableau de mots appelée 'WorkMemory'.

Structure en bus Can direct :



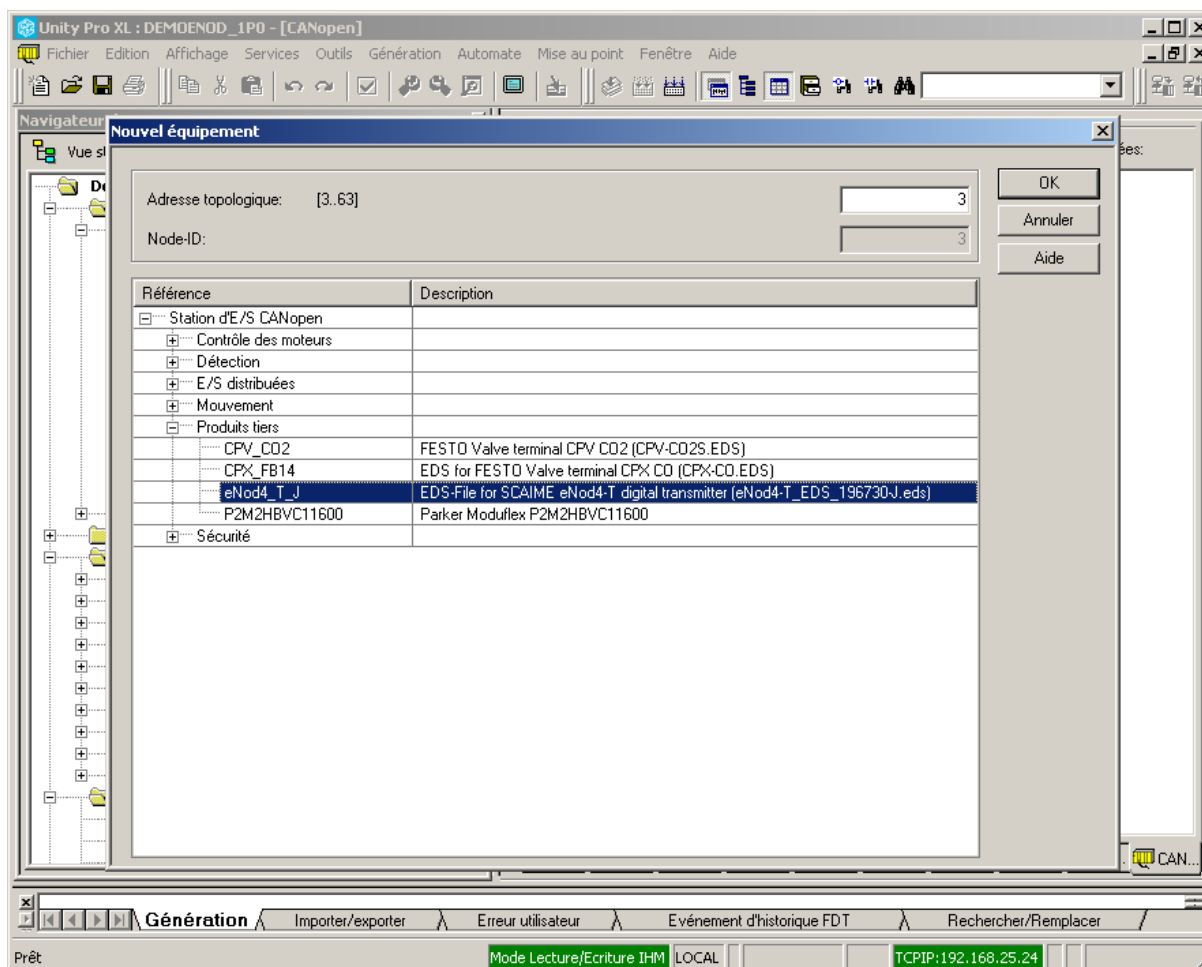
Structure en bus Can derrière STB :



Installation de la bibliothèque Scaime pour le bus CAN :

Avant de lancer Unity, ouvrir le logiciel <Hardware Catalog Manager> et sous Fichier/Importer, sélectionner <eNode4T.cpx>. Lancer ensuite la construction du catalogue. Unity est alors prêt à utiliser ce composant.

Dans le cadre de notre application, et pour l'exploitation du transmetteur de pesage en bus CAN directement sous l'automate, la configuration du bus CAN est réalisée en sélectionnant l'équipement eNod4_T_J dans la bibliothèque d'esclave CAN sous <Produits tiers>.



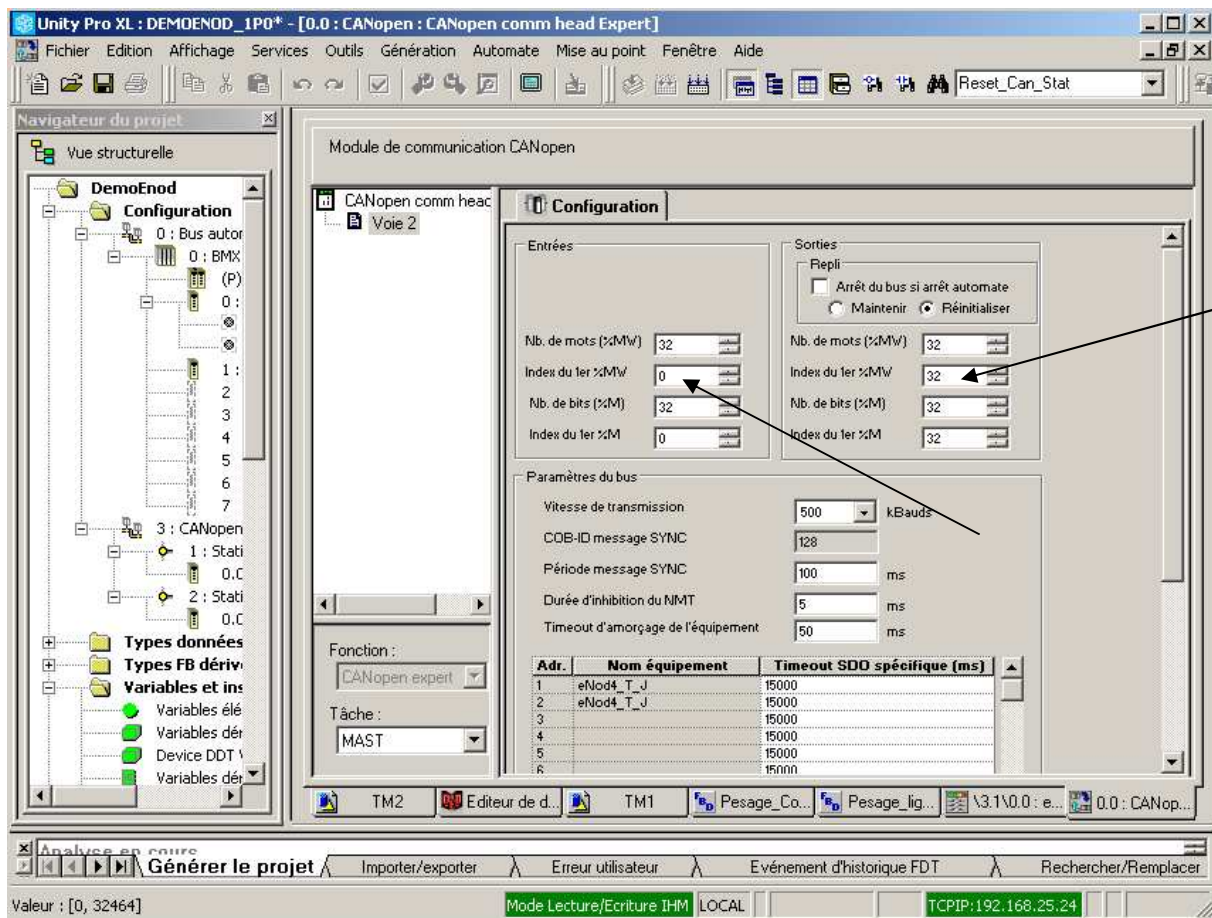
Exploitation en bus CAN sous automate M340:

Déclaration et configuration du bus CAN sous Unity :

Utiliser un automate avec bus Can intégré, et déployer sa configuration.

Préparation du mapping sous CANOpen :

- Définir les adresses de base et la taille du mapping mémoire du bus CAN. Ouvrir l'onglet CANOpen sous le processeur de la configuration du bus automate puis Voie 2 Configuration

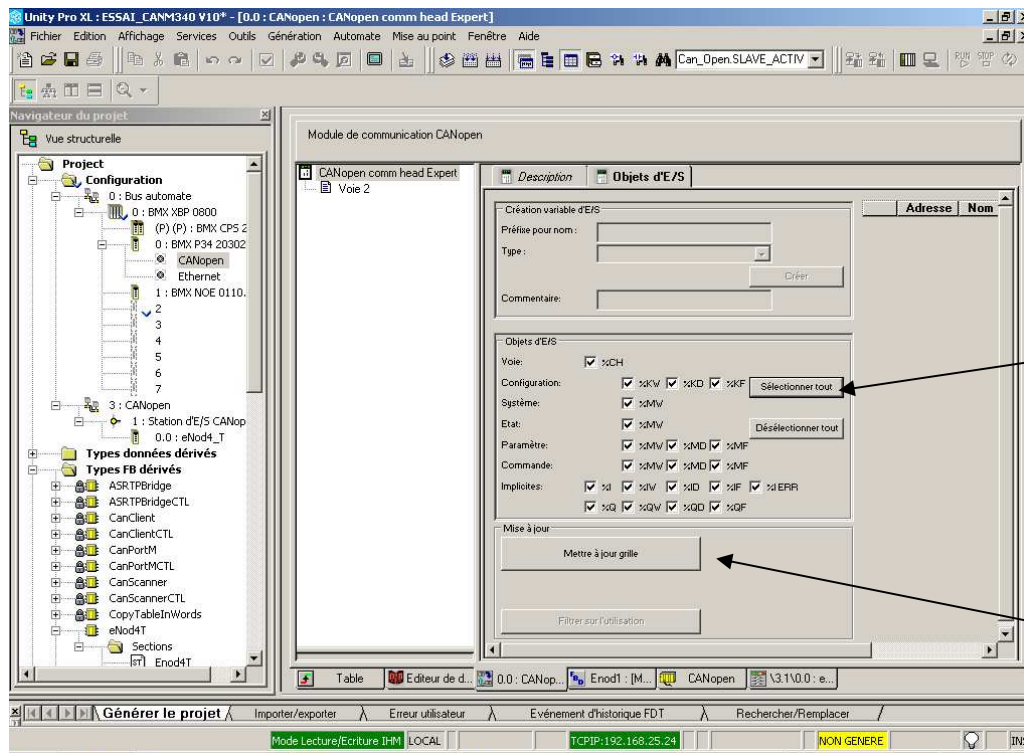


- Définissez vos adresses de début de chaque zone d'entrées et de sorties. (Ex %MW0 pour la zone d'entrées et %MW32 pour la zone de sorties)
- Définissez la longueur de chaque zone (Ex 32 mots)
- Idem pour les bits si vous les utilisez.

Les valeurs configurées dans les paramètres du bus peuvent restées pas défaut.

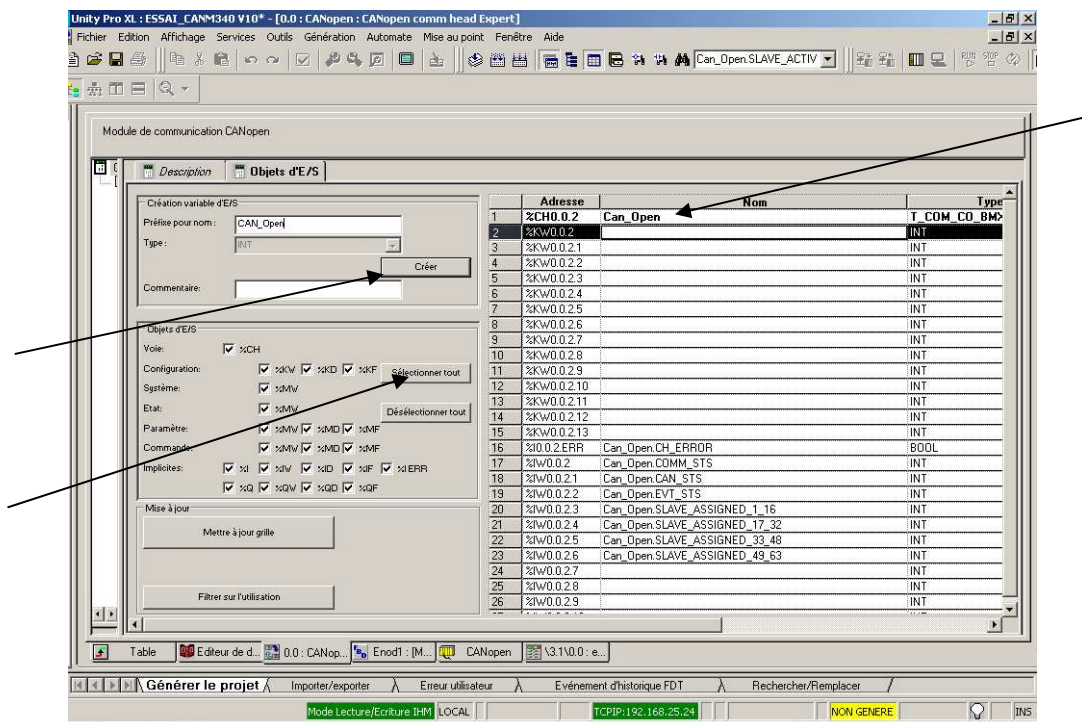
Une structure de variable image de l'ensemble du bus CAN peut être générée :

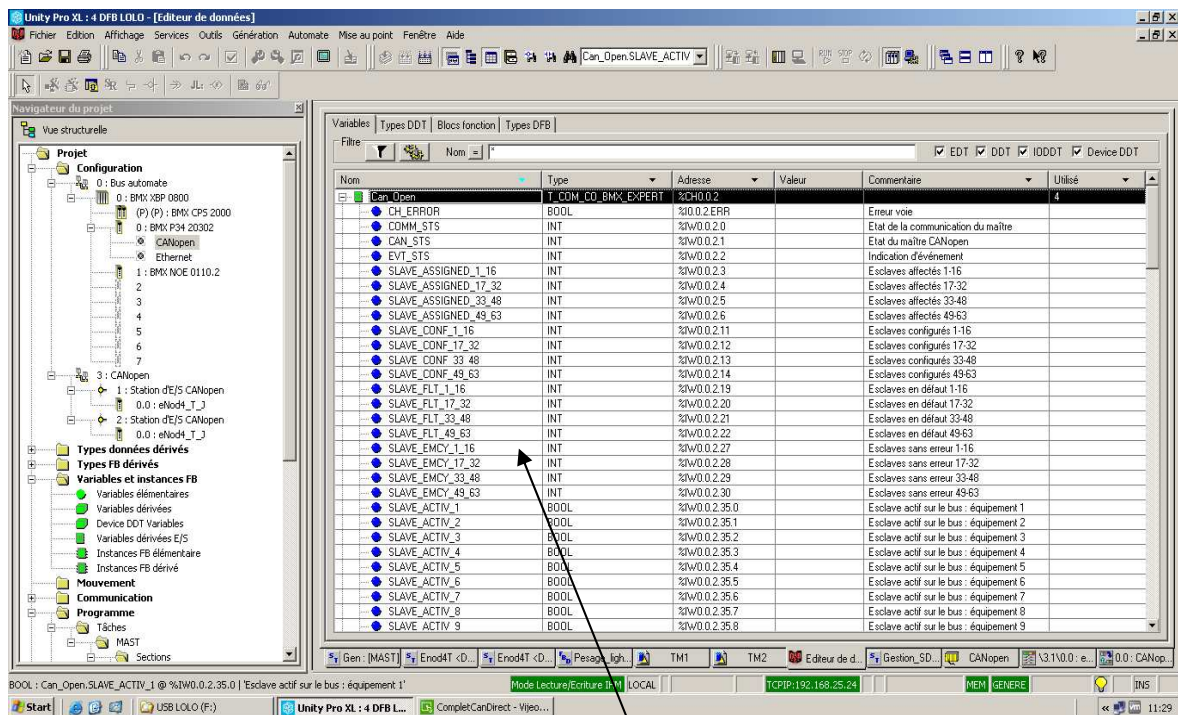
- Ouvrir l'onglet CANopen en déployant le processeur BMX de la configuration du bus automate
- Puis <CANopen comm head Expert> onglet <Objets d'E/S>



é

- Sélectionner tous les objets
- Mettre à jour la grille
- Sélectionner le premier objet dans la liste
- Choisir un nom de préfixe (Ex Can_Open) et créer la structure de variable

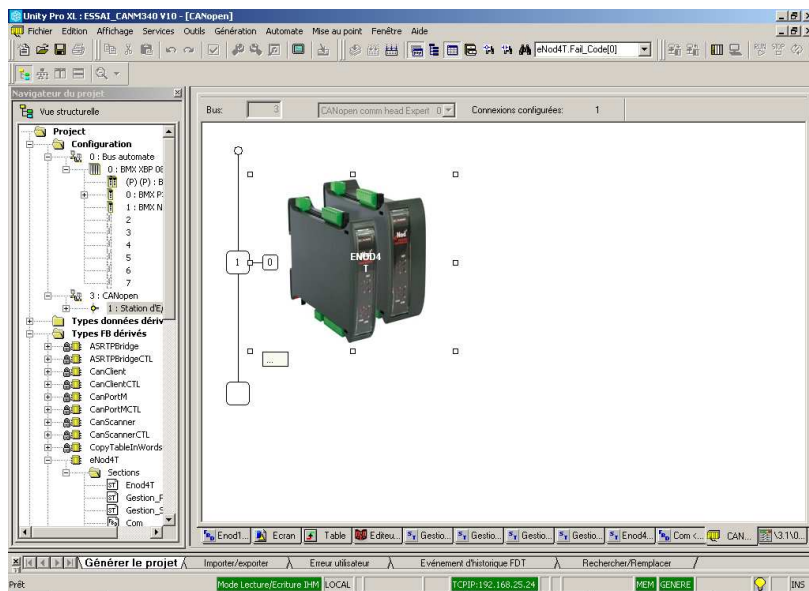




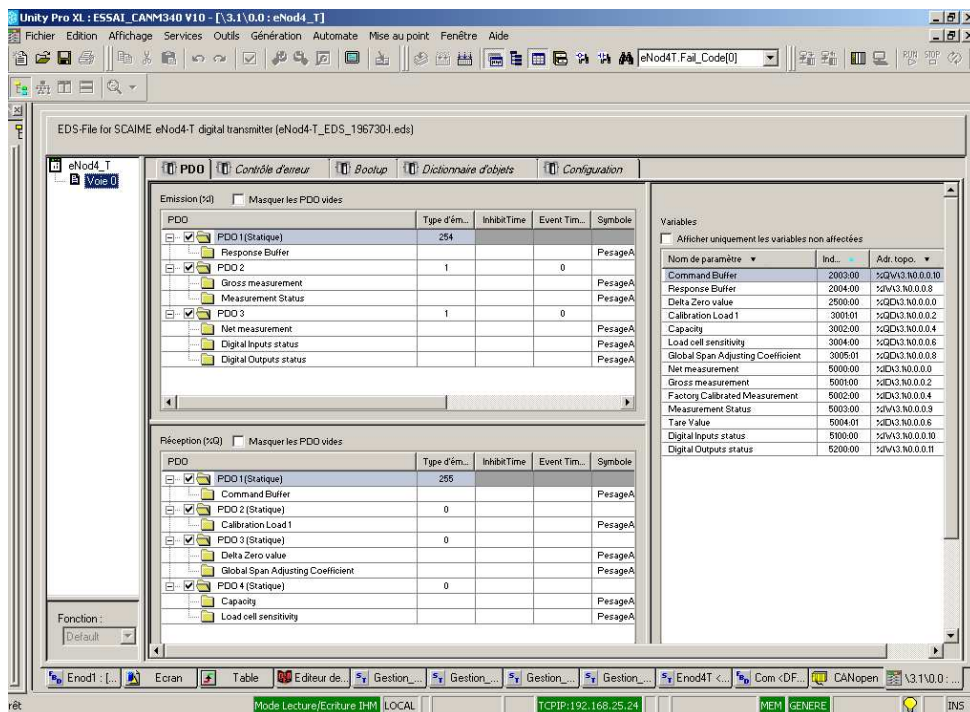
- La structure de variables Can_Open de type T_COM_CO_BMX_EXPERT a été créée en IODDT
- La variable Active_Slave_i (i=n° du nœud CAN) issue de cette structure sera utilisée en entrée de DFB décrit ci-après.

Configuration de module de pesage Scaime sous le bus CAN :

Sous Unity, en lançant la configuration du bus Can, vous pouvez ajouter vos modules de pesage comme suit :



En double cliquant dans l'esclave ENOD4T puis Voie 0, la vue d'affectation des PDO apparaît.

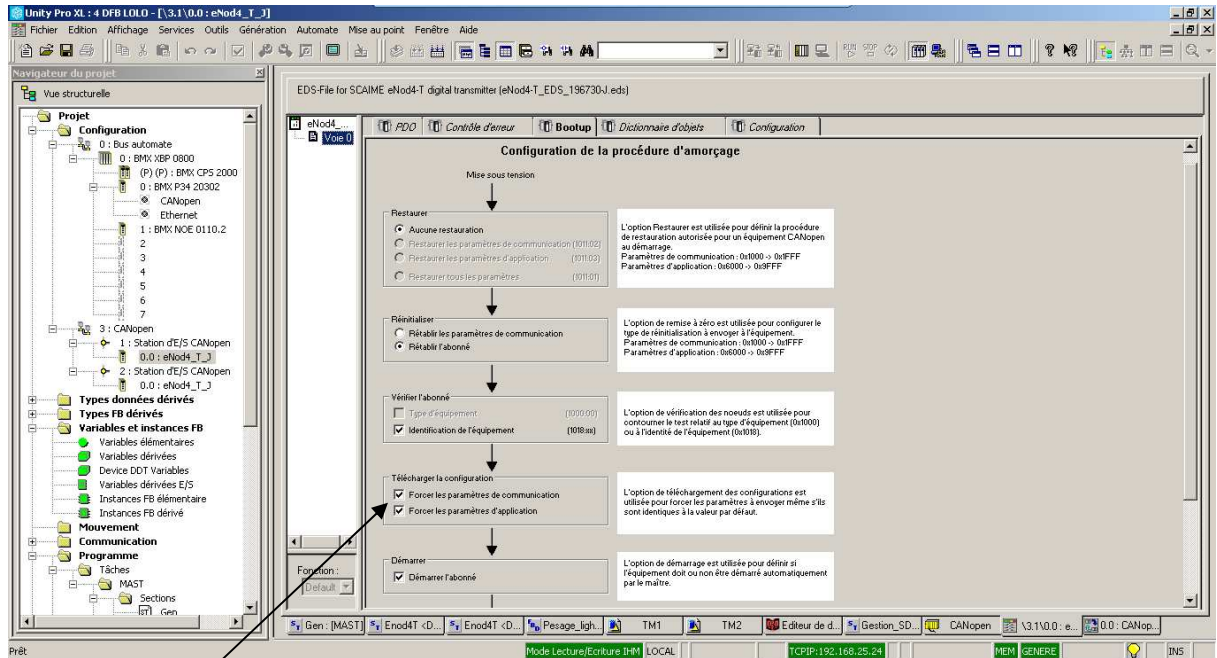


Les PDO sont classés en Réception et en Emission. C'est le fichier .EDS du constructeur qui régit le choix des variables accessibles par PDO (partie droite de la vue).

Nota : L'affectation des PDO Statique n'est pas modifiable.

La configuration de démarrage de chaque esclave enod sur le bus CAN doit être définie comme suit :

- Ouvrir l'onglet eNOD4_T sous 3 : CANopen de la configuration du projet
- Sélectionner l'onglet Voie 0 et <Bootup>



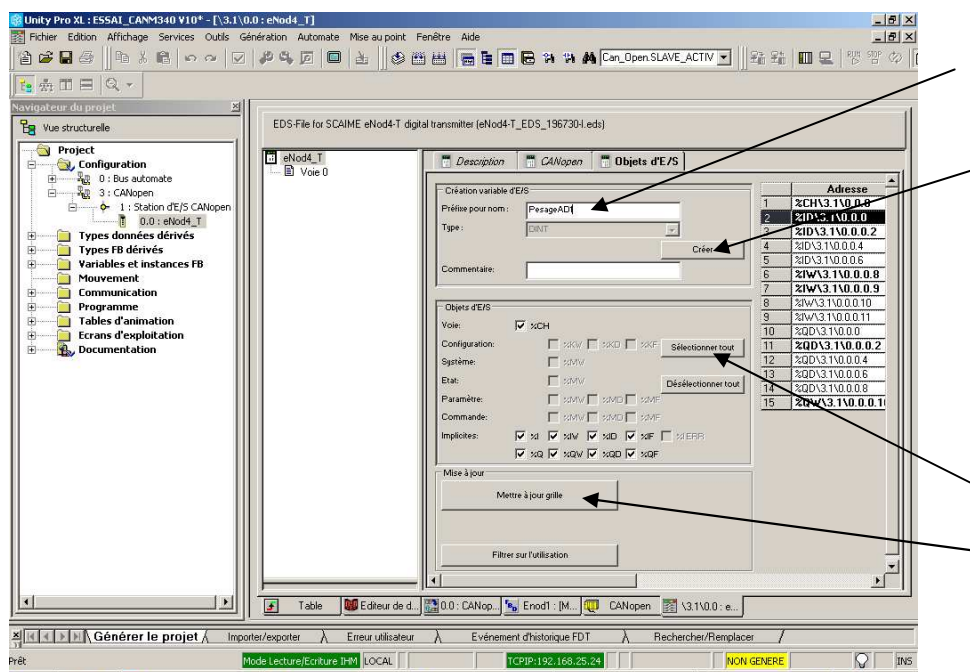
ATTENTION, Par défaut le téléchargement de la configuration dans l'esclave Can n'est pas sélectionné !!

Les temps configurés dans l'écran de contrôle d'erreur peuvent restés par défaut (Heartbeat)

L'ensemble des variables PDO sont adressées sous le bus CAN et peuvent formées une structure en IODDT.

Cette structure de variables PDO peut être générée pour chaque esclave enod configuré:

- Ouvrir l'onglet eNOD4_T sous 3 : CANopen de la configuration du projet
- Puis <Objets d'E/S>



- Sélectionner tous les objets
- Mettre à jour la grille
- Sélectionner le premier objet dans la liste
- Choisir un nom de préfixe (Ex PesageAD1) et créer la structure de variable

	Adresse	Nom	Type	Commentaire	%MW...
1	%CH\3.1\0.0.0	PesageAd1	T_eNod4_T		
2	%ID\3.1\0.0.0	PesageAd1.Net_measurement	DINT	Net_measurement	%MW0
3	%ID\3.1\0.0.0.2	PesageAd1.Gross_measurement	DINT	Gross_measurement	%MW2
4	%ID\3.1\0.0.0.4	PesageAd1.Factory_Calibrated_Measurement	DINT	Factory_Calibrated_Measurement	%MW4
5	%ID\3.1\0.0.0.6	PesageAd1.Tare_Value	DINT	Tare_Value	%MW6
6	%IW\3.1\0.0.0.8	PesageAd1.Response_Buffer	INT	Response_Buffer	%MW8
7	%IW\3.1\0.0.0.9	PesageAd1.Measurement_Status	INT	Measurement_Status	%MW9
8	%IW\3.1\0.0.0.10	PesageAd1.Digital_Inputs_status	INT	Digital_Inputs_status	%MW10
9	%IW\3.1\0.0.0.11	PesageAd1.Digital_Outputs_status	INT	Digital_Outputs_status	%MW11
10	%QD\3.1\0.0.0	PesageAd1.Delta_Zero_value	DINT	Delta_Zero_value	%MW1000
11	%QD\3.1\0.0.0.2	PesageAd1.Calibration_Load_1	DINT	Calibration_Load_1	%MW1002
12	%QD\3.1\0.0.0.4	PesageAd1.Capacity	DINT	Capacity	%MW1004
13	%QD\3.1\0.0.0.6	PesageAd1.Load_cell_sensitivity	DINT	Load_cell_sensitivity	%MW1006
14	%QD\3.1\0.0.0.8	PesageAd1.Global_Span_Adjusting_Coefficient	DINT	Global_Span_Adjusting_Coefficient	%MW1008
15	%QW\3.1\0.0.0.10	PesageAd1.Command_Buffer	INT	Command_Buffer	%MW1010

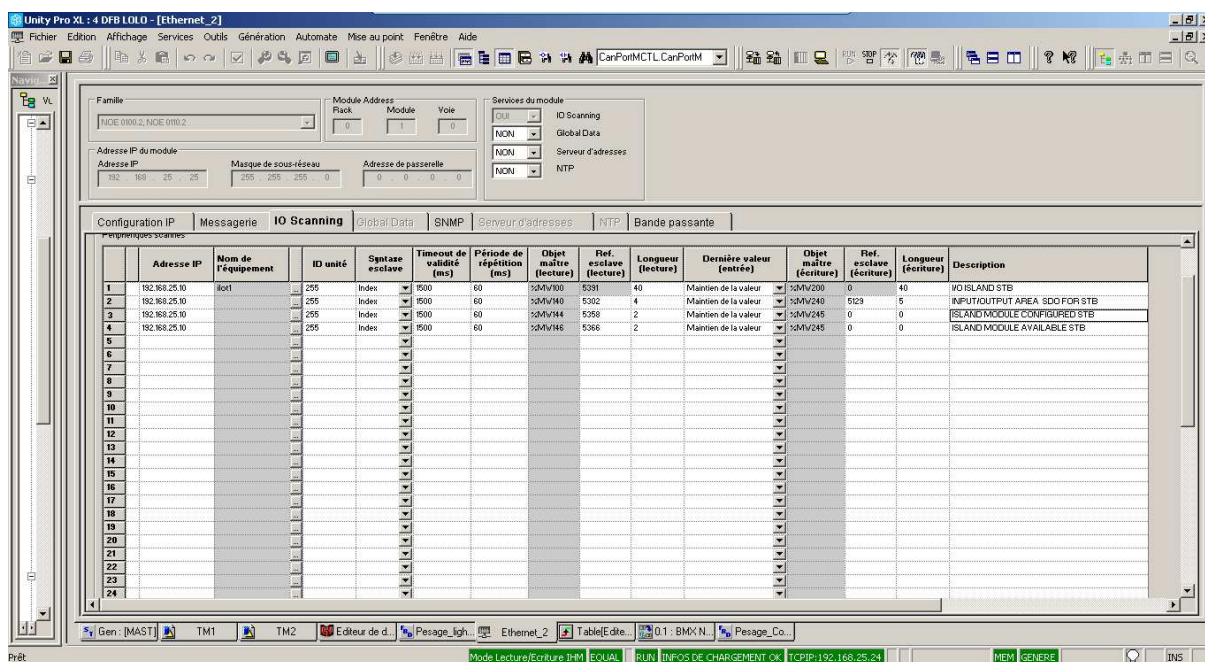
- Une structure PesageAD1 de type T_eNod4_T a été créée en IODDT et se trouve localisée aux adresses définies par le bus CAN

Exploitation derrière un îlot STB:

Configuration automate sous Unity :

Pour automate M340, utilisation du scanning d'I/O à travers une carte BMX NOE 0110.2 pour adresser l'îlot STB.

Ecran de configuration I/O Scanning de la carte NOE :



Adresse IP	Nom de l'équipement	ID unité	Syntaxe esclave	Timeout de validité (ms)	Période de répétition (ms)	Objet maître (lecture)	Ref. esclave (lecture)	Longueur (lecture)	Dernière valeur (entrée)	Objet maître (écriture)	Ref. esclave (écriture)	Longueur (écriture)	Description
192.168.25.10	ilot1	255	Index	1500	60	%MW100	5391	40	Maintien de la valeur	%MW200	0	40	I/O ISLAND STB
192.168.25.10		255	Index	1500	60	%MW140	5302	4	Maintien de la valeur	%MW240	5129	5	INPUT/OUTPUT AREA SDO FOR STB
192.168.25.10		255	Index	1500	60	%MW144	5358	2	Maintien de la valeur	%MW245	0	0	ISLAND MODULE CONFIGURED STB
192.168.25.10		255	Index	1500	60	%MW146	5366	2	Maintien de la valeur	%MW245	0	0	ISLAND MODULE AVAILABLE STB

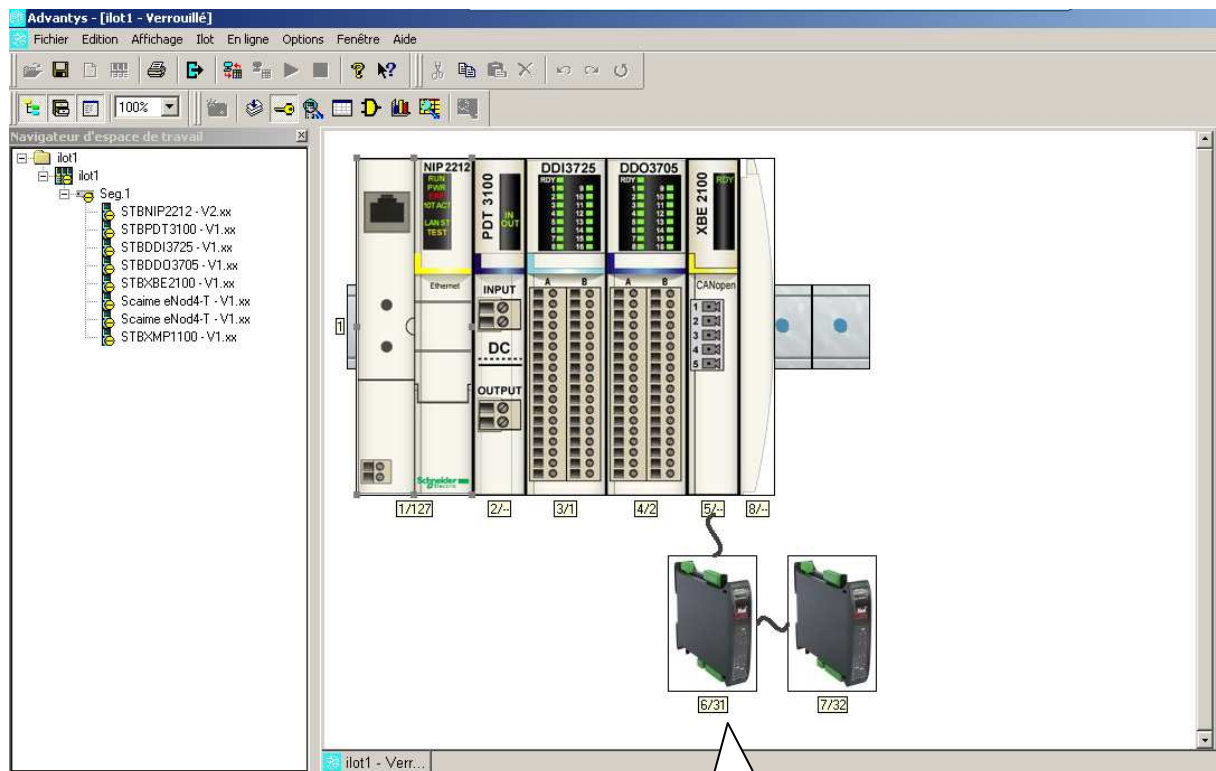
La 1^{ère} ligne correspond au mapping d'entrées sorties de l'îlot STB contenant ses cartes. Cette ligne fait référence au nom d'équipement <ilot1> créé par le logiciel Advantys.

La 2^{ème} ligne contient 4 mots en entrées et 5 mots en sorties pour adresser l'extension du bus Can sous l'îlot STB pour les échanges SDO. Ces adresses de variable serviront aux InputArea et OutputArea du bloc ASRTPBridge permettant la communication avec le bus de l'îlot STB. (Non utilisé avec le DFB Light).

La 3^{ème} ligne permet de récupérer deux mots correspondants aux 32 esclaves pouvant être configuré dans un îlot. (1=configuré)

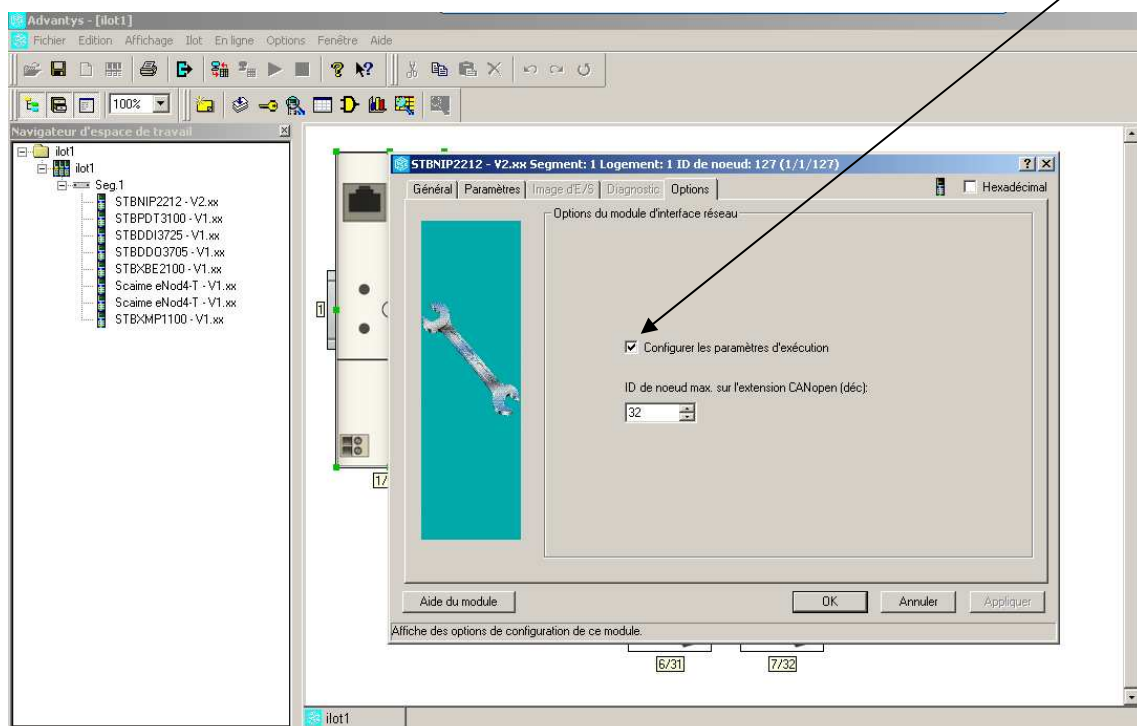
La 4^{ème} ligne permet de récupérer deux mots correspondants aux 32 esclaves présents dans l'îlot STB. (1=non présent) Ces bits de mots pourront servir dans la construction du signal « Active_Slave » du DFB

Configuration d'îlot sous Advantys :

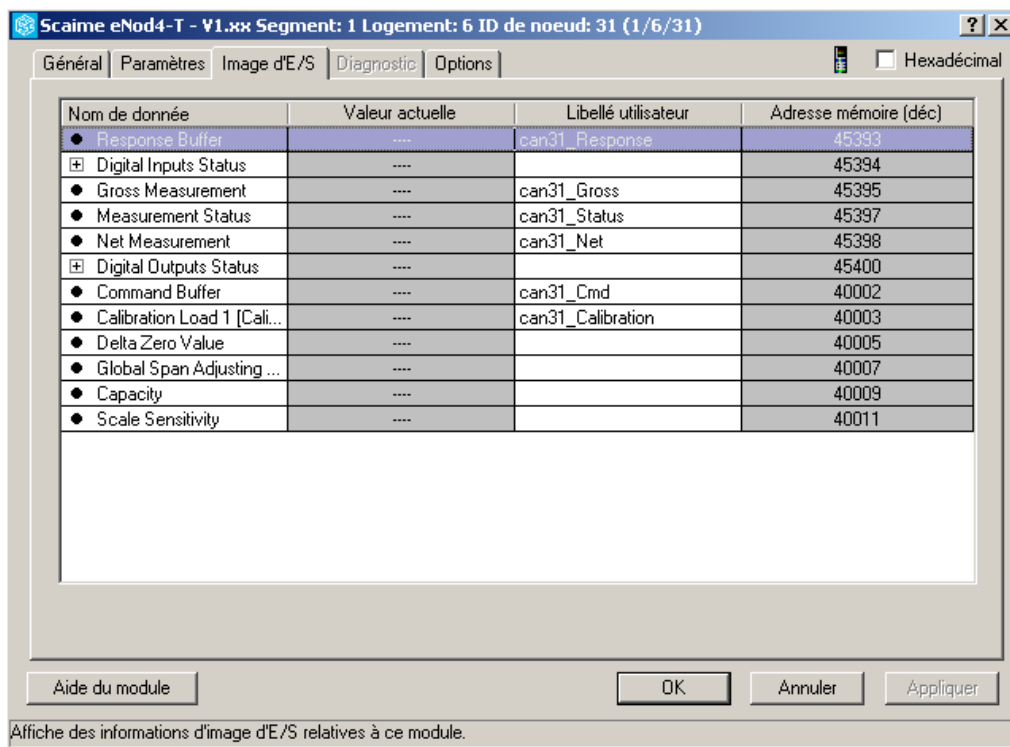


Exemple de deux eNod
configurés en adresse 31 & 32

Pour définir la communication avec le bus Can de l'îlot STB, il faut ouvrir la NIP2212 et sélectionner la boîte à coché suivante. Cela permet de communiquer en I/O Scanning avec l'automate (ligne 2 de la configuration NOE)



Ouvrir un module de pesage (exemple Ad 31) et dans l'onglet <Image d'E/S> saisir les noms des variables qui seront utilisées dans le projet Unity : (Réponse, Brute, Net, Status, Commande, Valeur d'étalonnage) avec un nom d'entête lié à l'adresse (exemple Can31_)



Faire de même avec tous les eNods configurés avec un nom d'entête différent.

Enregistrer l'ilot sous Advantys, charger le projet dans l'ilot physique et fermer Advantys.

En fermant Advantys, Unity va demander de mettre à jour sa base de données avec les variables créées sous Advantys.

InputArea	ARRAY[0..3] OF INT	%mw140	
OutputArea	ARRAY[0..4] OF INT	%mw240	
IL0T1can31_Response	INT	%mw101	ilot1_R1_S6_Scaime eNod4-T_Response Buffer
IL0T1can31_Gross	INT	%mw103	ilot1_R1_S6_Scaime eNod4-T_Gross Measurement
IL0T1can31_Gross_WD2	INT	%mw104	ilot1_R1_S6_Scaime eNod4-T_Gross Measurement
IL0T1can31_Status	INT	%mw105	ilot1_R1_S6_Scaime eNod4-T_Measurement Status
IL0T1can31_Net	INT	%mw106	ilot1_R1_S6_Scaime eNod4-T_Net Measurement
IL0T1can31_Net_WD2	INT	%mw107	ilot1_R1_S6_Scaime eNod4-T_Net Measurement
IL0T1can32_Response	INT	%mw109	ilot1_R1_S7_Scaime eNod4-T_Response Buffer
IL0T1can32_Gross	INT	%mw111	ilot1_R1_S7_Scaime eNod4-T_Gross Measurement
IL0T1can32_Gross_WD2	INT	%mw112	ilot1_R1_S7_Scaime eNod4-T_Gross Measurement
IL0T1can32_Status	INT	%mw113	ilot1_R1_S7_Scaime eNod4-T_Measurement Status
IL0T1can32_Net	INT	%mw114	ilot1_R1_S7_Scaime eNod4-T_Net Measurement
IL0T1can32_Net_WD2	INT	%mw115	ilot1_R1_S7_Scaime eNod4-T_Net Measurement
IL0T1can31_Cmd	INT	%mw201	ilot1_R1_S6_Scaime eNod4-T_Command Buffer
IL0T1can31_Calibration	INT	%mw202	ilot1_R1_S6_Scaime eNod4-T_Calibration Load 1 [Calibration Loads]
IL0T1can31_Calibration_WD2	INT	%mw203	ilot1_R1_S6_Scaime eNod4-T_Calibration Load 1 [Calibration Loads]
IL0T1can32_Cmd	INT	%mw212	ilot1_R1_S7_Scaime eNod4-T_Command Buffer
IL0T1can32_Calibration	INT	%mw213	ilot1_R1_S7_Scaime eNod4-T_Calibration Load 1 [Calibration Loads]
IL0T1can32_Calibration_WD2	INT	%mw214	ilot1_R1_S7_Scaime eNod4-T_Calibration Load 1 [Calibration Loads]

Répondre OUI.

Les variables suivantes sont importées dans Unity avec les adresses définies par la configuration d'I/O Scanning.

DFB COMPLET

Représentation du DFB sous UNITY :

eNod4T			
BOOL	- ModeCom	ERR	- BOOL
BOOL	- Active_Slave	Net_Measurement	- DINT
INT	- Address_Can	Gross_Measurement	- DINT
BOOL	- Tare_Cmd	Tare_Value	- DINT
BOOL	- Cancel_Tare_Cmd	Stability	- BOOL
BOOL	- Zero_Cmd	Tare_In_Progress	- BOOL
INT	- PDO_Response_Buffer		
INT	- PDO_Measurement_Status	Over_Measurement	- BOOL
DINT	- PDO_Net_Measurement	I	
DINT	- PDO_Gross_Measurement	Fail_Code	- ARRAY [0..1] of INT
INT	- STB_PDO_Net_Measurement	PDO_Command_Buffer	- INT
INT	- STB_PDO_Net_Measurement_WD2	PDO_Calibration_Load	- DINT
INT	- STB_PDO_Gross_Measurement	STB_PDO_Calibration_Load	- INT
INT	- STB_PDO_Gross_Measurement_WD2	STB_PDO_Calibration_Load_WD2	- INT
eNod4t_ST_DDT	- eNod4t_ST -----	eNod4t_ST	- eNod4t_ST_DDT
ANY_ARRAY_INT	- WorkMemory-----	WorkMemory	- ANY_ARRAY_INT
		OOS	- BOOL

Entrées du DFB

Nom	Type	Description	CAN	STB
ModeCom	BOOL	0=utilisation en bus CAN direct 1=utilisation en bus CAN derrière un module STB vu en I/O Scanning par l'automate	0	1
Active_Slave	BOOL	Esclave CAN présent sur le bus	x	x
Can_Address	INT	Adresse du nœud CAN de l'esclave	x	x
Tare_Cmd	BOOL	Sur front montant de ce signal, permet la commande de tarage. Acquisition de la valeur Brute comme valeur de tare. La valeur Net est remise à zéro	x	x
Cancel_Tare_Cmd	BOOL	Sur front montant de ce signal, permet la commande d'annulation de la tare.	x	x
Zero_Cmd	BOOL	Sur front montant de ce signal, permet la commande du réglage du zéro des mesures de l'appareil. ATTENTION cette commande est suivie d'une sauvegarde en EEPROM.	x	x
PDO_Response_Measurement	INT	Registre de réponse des commandes PDO	x	x
PDO_Measurement_Status	INT	Registre de statuts de l'eNod en PDO	x	x
PDO_Net_Measurement	DINT	Registre PDO de la mesure NET	x	
PDO_Gross_Measurement	DINT	Registre PDO de la mesure BRUTE	x	
STB_PDO_Net_Measurement	INT	Mesure net PDO en com vers STB poids faible		x
STB_PDO_Net_Measurement_WD2	INT	Mesure net PDO en com vers STB poids fort		x
STB_PDO_Gross_Measurement	INT	Mesure brute PDO en com vers STB poids faible		x
STB_PDO_Gross_Measurement_WD2	INT	Mesure brute PDO en com vers STB poids fort		x

Sorties du DFB

Nom	Type	Description	CAN	STB
ERR	BOOL	Erreur en cours	x	x
Net_Measurement	DINT	Mesure NET affectée de la densité	x	x
Gross_Measurement	DINT	Mesure BRUTE	x	x
Tare_Value	DINT	Valeur de la tare calculée en cours	x	x
Stability	BOOL	1= bascule stable	x	x
Tare_In_Progress	BOOL	1= tare en cours (Résultat d'une commande de tarage)	x	x
Over_Measurement	BOOL	1= bascule en dehors de la plage de la portée définie	x	x
Fail_Code	ARRAY [0..1] of INT	Code d'erreur du bloc de communication. Voir Documentation DPL des blocs 'CanClientCTL' Concerné uniquement la communication SDO.	x	x
PDO_Command_Buffer	INT	Registre de commande PDO	x	x
PDO_Calibration_Load	DINT	Valeur du poids d'étalonnage PDO pour la séquence de calibration.	x	
STB_PDO_Calibration_Load	INT	Valeur du poids d'étalonnage PDO pour la séquence de calibration en mode de com STB poids faible.		x
STB_PDO_Calibration_Load_WD2	INT	Valeur du poids d'étalonnage PDO pour la séquence de calibration en mode de com STB poids fort.		x
OOS	BOOL	Indique la mise hors service du transmetteur depuis la supervision (voir Enod4t_ST.CFGW)	x	x

Entrées Sorties du DFB

La première variable est une structure de 25 variables qui permet la mise en relation avec la supervision.

La deuxième variable est un tableau permettant d'échanger les informations entre les blocs de communication DPL CanClient et CanPort ou ASRTPBridge. Ce tableau est à dimensionner suivant la valeur de la sortie WantedArraySize du bloc de communication utilisé.

Structure	Type	Description
eNod4t_ST	eNod4t_ST_DDT	Structure de données
WorkMemory	ANY_ARRAY_INT	Tableau de mots de longueur variable suivant le nombre de client Can connecté au bloc CanPortCTL

Structure de variables eNod4t_ST_DDT

Nom	Type	Description	
.STW	WORD	Mot dont les bits représentent le statut de l'actionneur pour la supervision	
.STW1	WORD	Mot dont les bits représentent le statut de l'actionneur pour la supervision	
.CFGW	WORD	Mot dont les bits permettent la commande de l'actionneur par la supervision	
.GROSS	DINT	Valeur de la mesure BRUTE	
.NET	DINT	Valeur de la mesure NET	
.TARE	DINT	Valeur de la tare calculée en cours	
.GROSS_SP	DINT	Consigne brute en mode simulation	
.NET_SP	DINT	Consigne NET en mode manuel	
.TARE_SP	DINT	Consigne de tare en mode simulation	
.Capacity	DINT	Lecture de la capacité dans l'Enod4	
.ScaleInterval	INT	Lecture de l'échelon dans l'Enod4	
.CalibrationLoad	DINT	Consigne de poids de référence en mode étalonnage	
.CapacityLoad	DINT	Consigne de portée de l'appareil	
.ScaleIntervalLoad	INT	Consigne d'échelon de l'appareil	
.StepCalibration	INT	Numéro d'étape du séquentiel d'étalonnage MESSAGE=0 Vide MESSAGE=1 Mise en mode étalonnage MESSAGE=2 Validation charge nulle MESSAGE=3 Acquisition du zéro MESSAGE=4 Validation valeur étalon MESSAGE=5 Validation acquisition charge étalon MESSAGE=6 Validation fin détalonnage MESSAGE=7 Mémorisation réglage MESSAGE=8 Fin d'étalonnage	
.SoftVersion	REAL	Version logicielle	
.HardVersion	REAL	Version Matérielle	
.SerialNumber	DINT	Numéro de série	
.CanAddress	INT	Adresse de l'appareil sur le bus CAN	
.Density	REAL	L/E Correction à apporter sur la mesure Net pour affichage en volume. Par défaut à 1.0	
.Message	INT	Code de message à afficher MESSAGE=0 Vide MESSAGE=1 OK MESSAGE=2 En Cours MESSAGE=3 Esclave Non OK MESSAGE=4 Mode simulation MESSAGE=5 Défaut MESSAGE=6 déclenchement en time out MESSAGE=7 Défaut bus CAN	
.LL_Level	DINT	L/E Valeur du seuil très bas pour générer l'alarme LL	
.L_Level	DINT	L/E Valeur du seuil bas pour générer l'alarme L	
.H_Level	DINT	L/E Valeur du seuil haut pour générer l'alarme H	
.HH_Level	DINT	L/E Valeur du seuil très haut pour générer l'alarme HH	

Description des bits du mot de statut .STW

Bit	Nom	Description
0	TARE IN PROGRESS	Recopie du signal Tare In Progress
1	STABILITY	Recopie du signal Stability
2		Reserve
3	OVER	Recopie du signal Over measurement
4	IN1	Recopie du signal IN1
5	IN2	Recopie du signal IN2
6	OUT1	Recopie du signal OUT1
7	OUT2	Recopie du signal OUT2
8	OUT3	Recopie du signal OUT3
9	OUT4	Recopie du signal OUT4
10	SLAVE ACTIVATE	Recopie du signal Slave activate
11	MODE_COM	Recopie du signal de type d'utilisation du bus CAN
12	COMPLETE	1=DFB Complet 0=DFB Simple
13		Reserve
14	ERR	Défaut en cours
15	OK	Transmetteur opérationnel, indique qu'il n'est pas hors service, pas en défaut de capacité, pas en manuel, avec communication en activité, pas en défaut de compte rendu de Com

Description des bits du mot de statut .STW1

Bit	Nom	Description
0	LL_Level	Alarme Très basse sur seuil de mesure NET
1	L_Level	Alarme basse sur seuil de mesure NET
2	H_Level	Alarme haute sur seuil de mesure NET
3	HH_Level	Alarme Très haute sur seuil de mesure NET
4		Reserve
5		Reserve
6		Reserve
7		Reserve
8		Reserve
9		Reserve
10		Reserve
11		Reserve
12		Reserve
13		Reserve
14		Reserve
15		Reserve

Description des bits du mot de commande .CFGW

Bit	Nom	Description
0	OWNER	Lecture écriture par supervision pour positionner le transmetteur en mode acquisition (Automatique) (OWNER=0) ou en mode Opérateur (Manuel) (OWNER=1)
1		Reserve
2	SIMMD	Lecture écriture par supervision pour positionner l'actionneur en mode simulation. Ce mode déconnecte les entrées physique du bloc et les traite en interne par recopie de la sortie. Pas d'alarme de généré dans ce mode.
3	TARE_CMD	Ecriture par supervision pour commander le tarage de l'appareil. Acquisition de la valeur Brute comme valeur de tare. La valeur Net est remise à zéro. Signal remis à zéro par programme après traitement.
4	CANCEL_TARE_CMD	Ecriture par supervision pour commander l'annulation de la tare de l'appareil. Signal remis à zéro par programme après traitement.
5	ZERO_CMD	Ecriture par supervision pour commander la remise à zero des mesures de l'appareil. (si la mesure est stable et si la mesure est comprise dans une plage de 10% de la portée). Cette commande est suivie d'une sauvegarde EEPROM. Signal remis à zéro par programme après traitement.
6	CALIBRATION_CMD	Lecture écriture par supervision. Commande de la séquence d'étalonnage de l'appareil.
7	CHANGE_CAPACITY_CMD	Ecriture par supervision. Permet de prendre en compte la valeur saisie dans le champ 'CapacityLoad' pour changer la portée de traitement de l'appareil. Signal remis à zéro par programme après traitement.
8	CHANGE_SCALE_INTERVAL_CMD	Ecriture par supervision. Permet de prendre en compte la valeur saisie dans le champ 'ScaleIntervalLoad' pour changer l'échelon de mesure de l'appareil. Signal remis à zéro par programme après traitement.
9		Reserve
10	VAL_STEP_CALIBRATION	Signal de validation des phases de calibration de l'appareil. Signal remis à zéro par programme après traitement.
11	CANCEL_CALIBRATION_CMD	Signal d'annulation de la dernière commande dans le cycle de calibration en cours. Signal remis à zéro par programme après traitement.
12		Reserve
13		Reserve
14	ACK	Acquittement erreur
15	OOS	Out Of Service

Variable privé du DFB :

Nom	Type	Description
PDO_Request	INT	Mot de préparation contenant le code de commande a émettre en PDO vers l'appareil
SDO_Index	INT	Mot contenant la valeur de l'index a émettre en SDO vers l'appareil
SDO_SubIndex	INT	Mot contenant la valeur du sous index a émettre en SDO vers l'appareil
SDO_RW	BOOL	0=lecture SDO 1=écriture SDO
SDO_Start	EBOOL	Déclenchement de la Com SDO
Cmd_Tare	EBOOL	Pour front de déclenchement de la commande de tarage
Cmd_Cancel_Tare	EBOOL	Pour front de déclenchement de la commande de d'annulation de la tare
Cmd_Zero	EBOOL	Pour front de déclenchement de la commande de mise à zéro de la mesure
Seq_Com_SDO	INT	Mot de séquentiel de gestion d'émission SDO
Length	INT	Longueur des données à émettre en SDO
Cpt	INT	Compteur de gestion réponse SDO
Cpt_Fail	INT	Compteur pour time out SDO
Wait	INT	Pour compteur scrutation automate en attente réponse
Save_Cmd	BOOL	Commande de sauvegarde en EEPROM
Seconde	EBOOL	Pour déclenchement à la seconde
Top	EBOOL	Top de déclenchement lecture SDO
Word1	WORD	Mot de travail pour conversion
Word2	WORD	Mot de travail pour conversion
CanClientCTL	DFB	DFB de la bibliothèque DPL pour la gestion de la communication d'un client CanOpen
WorkMemory	ARRAY [0..29] of INT	Variables liés aux transferts d'information entre les DFB CanClientCTL, CanPortMCTL et ASRTPBridgeCTL
WantedArraySize	INT	Mot de gestion bloc COM DPL pour estimation de la taille du WorkMemory
Data	ARRAY [0..10] of INT	Variable de retour du DFB CanClientCTL
Tempo	TON	Temporisation de time out PDO
TempoAFF	TON	Temporisation de RAZ message d'affichage

Sections programme du DFB :

Nom	Type	Description
Enod4T	ST	Gestion des modes de marche et traitement des mesures
Gestion_PDO	ST	Gestion des séquentiels de fonctionnement en communication PDO
Gestion_SDO	ST	Gestion des séquentiels de fonctionnement en communication SDO
Com_Client	FBD	Bloc Can Client CTL

Cycle spécifique au DFB :

Etalonnage :

Cycle d'étalonnage :

- Démarrage du cycle (Scada par la variable DDT Calibration_CMD)
- Validation bascule vide (Scada par la variable DDT Val_Calibration_Step)
- Acquisition par l'eNod du poids vide
- Saisie de la valeur du poids d'étalonnage dans la variable DDT CalibrationLoad et validation par la variable DDT Val_Calibration_Step
- Ecriture du poids étalon dans le système eNod
- Validation du poids d'étalonnage par la variable DDT Val_Calibration_Step
- Acquisition du poids étalon
- Validation fin du cycle par la variable DDT Val_Calibration_Step
- Sauvegarde des caractéristiques d'étalonnage dans l'eNod.
- Fin

Le BP annulation par la variable DDT Cancel_Calibration_CMD permet d'annuler la dernière commande effectuée dans l'eNod.

Portée :

Saisir la valeur maximum d'étendu de votre système de pesage dans la variable DDT CapacityLoad et valider par la variable CFGW.7 (Change_Capacity_CMD)

Echelon :

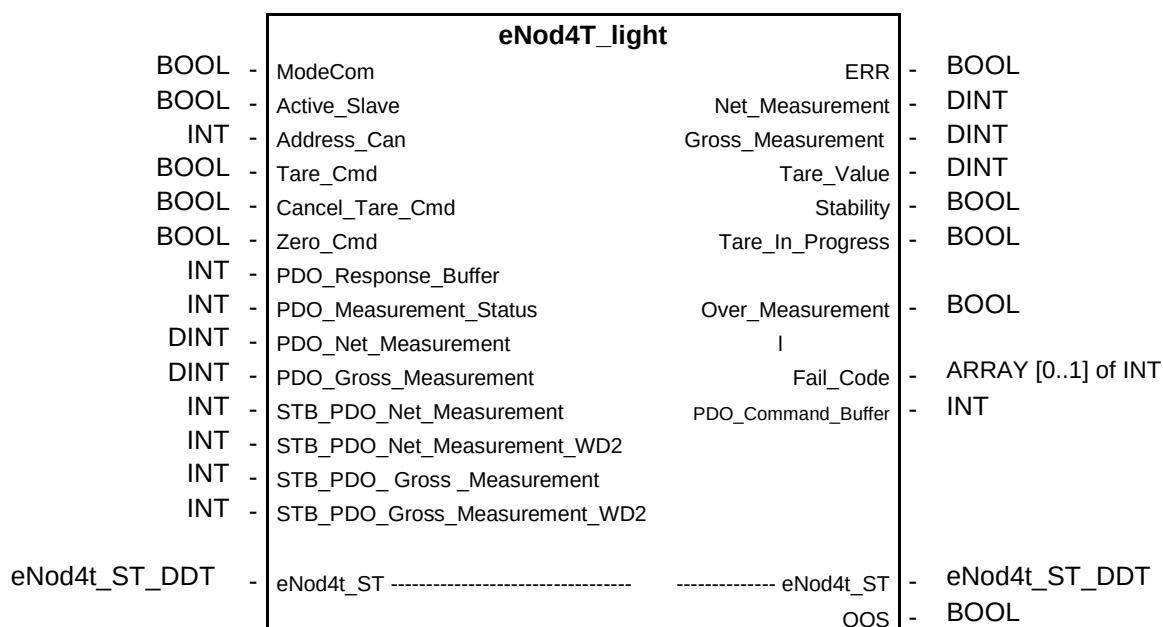
Saisir la valeur d'échelon désirée dans la variable DDT ScaleIntervalLoad et valider par la variable CFGW.8 (Change_Scale_Interval_CMD)

DFB SIMPLIFIE

Ce DFB est issu du DFB complet avec des fonctionnalités en moins. Il conserve la même structure de communication Scada que le DFB complet avec certaines variables qui ne sont plus utilisées.

Ce DFB n'utilise pas de communication explicite SDO

Représentation du DFB sous UNITY :



Entrées du DFB

Nom	Type	Description	CAN	STB
ModeCom	BOOL	0=utilisation en bus CAN direct 1=utilisation en bus CAN derrière un module STB vu en I/O Scanning par l'automate	0	1
Active_Slave	BOOL	Esclave CAN présent sur le bus	x	x
Can_Address	INT	Adresse du nœud CAN de l'esclave	x	x
Tare_Cmd	BOOL	Sur front montant de ce signal, permet la commande de tarage. Acquisition de la valeur Brute comme valeur de tare. La valeur Net est remise à zéro	x	x
Cancel_Tare_Cmd	BOOL	Sur front montant de ce signal, permet la commande d'annulation de la tare.	x	x
Zero_Cmd	BOOL	Sur front montant de ce signal, permet la commande du réglage du zéro des mesures de l'appareil. ATTENTION cette commande est suivie d'une sauvegarde en EEPROM.	x	x
PDO_Response_Measurement	INT	Registre de réponse des commandes PDO	x	x
PDO_Measurement_Status	INT	Registre de statuts de l'eNod en PDO	x	x
PDO_Net_Measurement	DINT	Registre PDO de la mesure NET	x	
PDO_Gross_Measurement	DINT	Registre PDO de la mesure BRUTE	x	
STB_PDO_Net_Measurement	INT	Mesure net PDO en com vers STB poids faible		x
STB_PDO_Net_Measurement_WD2	INT	Mesure net PDO en com vers STB poids fort		x
STB_PDO_Gross_Measurement	INT	Mesure brute PDO en com vers STB poids faible		x
STB_PDO_Gross_Measurement_WD2	INT	Mesure brute PDO en com vers STB poids fort		x

Sorties du DFB

Nom	Type	Description	CAN	STB
ERR	BOOL	Erreur en cours	x	x
Net_Measurement	DINT	Mesure NET affectée de la densité	x	x
Gross_Measurement	DINT	Mesure BRUTE	x	x
Tare_Value	DINT	Valeur de la tare calculée en cours	x	x
Stability	BOOL	1= bascule stable	x	x
Tare_In_Progress	BOOL	1= tare en cours (Résultat d'une commande de tarage)	x	x
Over_Measurement	BOOL	1= bascule en dehors de la plage de la portée définie	x	x
Fail_Code	ARRAY [0..1] of INT	Code d'erreur du bloc de communication. Voir Documentation DPL des blocs 'CanClientCTL' Concerne uniquement la communication SDO.	x	x
PDO_Command_Buffer	INT	Registre de commande PDO	x	x
OOS	BOOL	Indique la mise hors service du transmetteur depuis la supervision (voir Enod4t_ST.CFGW)	x	x

Entrées Sorties du DFB

Cette variable est une s

Structure de 25 variables qui permet la mise en relation avec la supervision.

Structure	Type	Description
eNod4t_ST	eNod4t_ST_DDT	Structure de données

Structure de variables eNod4t_ST_DDT

Nom	Type	Description	
.STW	WORD	Mot dont les bits représentent le statut de l'actionneur pour la supervision	
.STW1	WORD	Mot dont les bits représentent le statut de l'actionneur pour la supervision	
.CFGW	WORD	Mot dont les bits permettent la commande de l'actionneur par la supervision	
.GROSS	DINT	Valeur de la mesure BRUTE	
.NET	DINT	Valeur de la mesure NET	
.TARE	DINT	Valeur de la tare calculée en cours	
.GROSS_SP	DINT	Consigne brute en mode simulation	
.NET_SP	DINT	Consigne NET en mode manuel	
.TARE_SP	DINT	Consigne de tare en mode simulation	
.Capacity	DINT	(*)	
.ScaleInterval	INT	(*)	
.CalibrationLoad	DINT	(*)	
.CapacityLoad	DINT	(*)	
.ScaleIntervalLoad	INT	(*)	
.StepCalibration	INT	(*)	
.SoftVersion	DINT	(*)	
.HardVersion	DINT	(*)	
.SerialNumber	DINT	(*)	
.CanAddress	INT	Adresse de l'appareil sur le bus CAN	
.Density	REAL	L/E Correction à apporter sur la mesure Net pour affichage en volume. Par défaut à 1.0	
.Message	INT	Code de message à afficher MESSAGE=0 Vide MESSAGE=1 OK MESSAGE=2 En Cours MESSAGE=3 Esclave Non OK MESSAGE=4 Mode simulation MESSAGE=5 Défaut MESSAGE=6 déclenchement en time out MESSAGE=7 Défaut bus CAN	
.LL_Level	DINT	L/E Valeur du seuil très bas pour générer l'alarme LL	
.L_Level	DINT	L/E Valeur du seuil bas pour générer l'alarme L	
.H_Level	DINT	L/E Valeur du seuil haut pour générer l'alarme H	
.HH_Level	DINT	L/E Valeur du seuil très haut pour générer l'alarme HH	

(*) Significatif uniquement avec le DFB complet

Description des bits du mot de statut .STW

Bit	Nom	Description
0	TARE_IN_PROGRESS	Recopie du signal Tare_In_Progress
1	STABILITY	Recopie du signal Stability
2		Reserve
3	OVER	Recopie du signal Over measurement
4	IN1	Recopie du signal IN1
5	IN2	Recopie du signal IN2
6	OUT1	Recopie du signal OUT1
7	OUT2	Recopie du signal OUT2
8	OUT3	Recopie du signal OUT3
9	OUT4	Recopie du signal OUT4
10	SLAVE_ACTIVATE	Recopie du signal Slave_activate
11	MODE_COM	Recopie du signal de type d'utilisation du bus CAN
12	COMPLETE	1=DFB Complet 0=DFB Simple
13		Reserve
14	ERR	Défaut en cours
15	OK	Transmetteur opérationnel, indique qu'il n'est pas hors service, pas en défaut de capacité, pas en manuel, avec communication en activité, pas en défaut de compte rendu de Com

Description des bits du mot de statut .STW1

Bit	Nom	Description
0	LL_Level	Alarme Très basse sur seuil de mesure NET
1	L_Level	Alarme basse sur seuil de mesure NET
2	H_Level	Alarme haute sur seuil de mesure NET
3	HH_Level	Alarme Très haute sur seuil de mesure NET
4		Reserve
5		Reserve
6		Reserve
7		Reserve
8		Reserve
9		Reserve
10		Reserve
11		Reserve
12		Reserve
13		Reserve
14		Reserve
15		Reserve

Description des bits du mot de commande .CFGW

Bit	Nom	Description
0	OWNER	Lecture écriture par supervision pour positionner le transmetteur en mode acquisition (Automatique) (OWNER=0) ou en mode Opérateur (Manuel) (OWNER=1)
1		Reserve
2	SIMMD	Lecture écriture par supervision pour positionner l'actionneur en mode simulation. Ce mode déconnecte les entrées physique du bloc et les traite en interne par recopie de la sortie. Pas d'alarme de généré dans ce mode.
3	TARE_CMD	Ecriture par supervision pour commander le tarage de l'appareil. Acquisition de la valeur Brute comme valeur de tare. La valeur Net est remise à zéro. Signal remis à zéro par programme après traitement.
4	CANCEL_TARE_CMD	Ecriture par supervision pour commander l'annulation de la tare de l'appareil. Signal remis à zéro par programme après traitement.
5	ZERO_CMD	Ecriture par supervision pour commander la remise à zéro des mesures de l'appareil. (si la mesure est stable et si la mesure est comprise dans une plage de 10% de la portée). Cette commande est suivie d'une sauvegarde EEPROM. Signal remis à zéro par programme après traitement.
6	CALIBRATION_CMD	(*)
7	CHANGE_CAPACITY_CMD	(*)
8	CHANGE_SCALE_INTERVAL_CMD	(*)
9		Reserve
10	VAL_STEP_CALIBRATION	(*)
11	CANCEL_CALIBRATION_CMD	(*)
12		Reserve
13		Reserve
14	ACK	Acquittement erreur
15	OOS	Out Of Service

(*) Significatif uniquement avec le DFB complet

Variable privé du DFB :

Nom	Type	Description
PDO_Request	INT	Mot de préparation contenant le code de commande a émettre en PDO vers l'appareil
Save_Cmd	BOOL	Commande de sauvegarde en EEPROM
Cmd_Tare	EBOOL	Pour front de déclenchement de la commande de tarage
Cmd_Cancel_Tare	EBOOL	Pour front de déclenchement de la commande de d'annulation de la tare
Cmd_Zero	EBOOL	Pour front de déclenchement de la commande de mise à zéro de la mesure
Tempo	TON	Temporisation de time out PDO
TempoAFF	TON	Temporisation de RAZ message d'affichage

Sections programme du DFB :

Nom	Type	Description
Enod4T	ST	Gestion des modes de marche et traitement des mesures
Gestion_PDO	ST	Gestion des séquentiels de fonctionnement en communication PDO

Programme exemple

Le programme de démo regroupe 4 configurations d'exemples. Il instancie deux DFB de pesage TM1 et TM2 dans chaque type de communication. Ces DFB sont adressés avec les mêmes structures de variables.

- Mode DFB complet en liaison Can automate
- Mode DFB Light en liaison Can automate
- Mode DFB complet en liaison derrière un ilot STB
- Mode DFB Light en liaison derrière un ilot STB

Un seul mode peut être actif à la fois. 4 bits permettent la sélectivité de scrutation des sections de programme. Ces bits sont armés dans la section 'Gen'.

- Le bit « Mode_Com_eNod_CAN » autorise les sections suivantes:
 - Pesage_Complet_CAN_TM1
 - Pesage_Complet_CAN_TM2
 - Pesage_Com_CAN

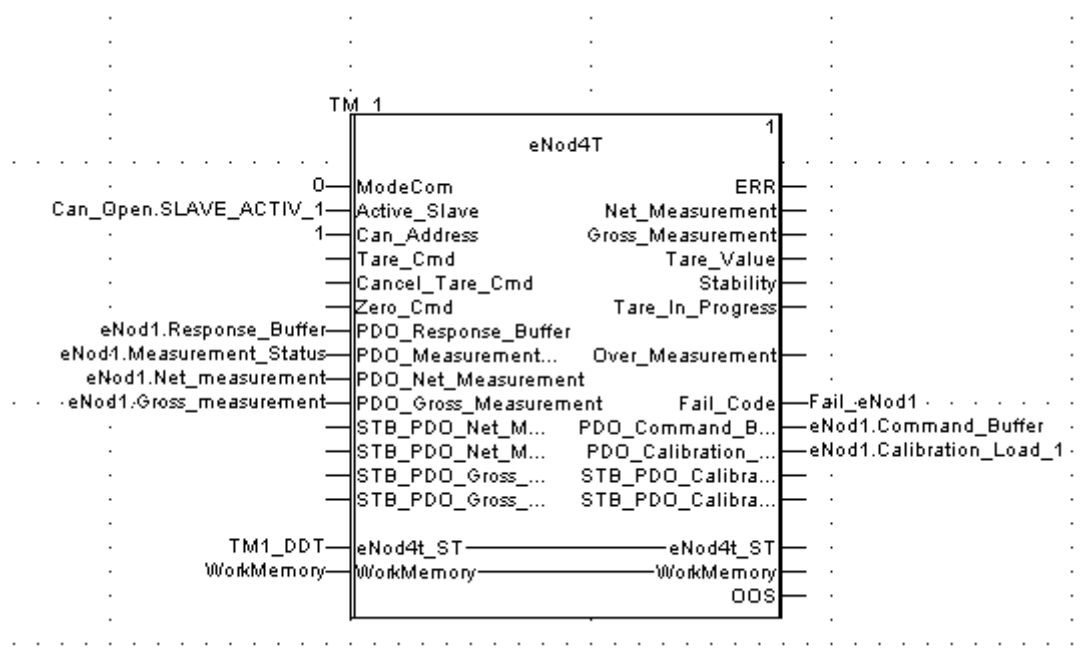
- Le bit « Mode_Com_eNod_CAN_L » autorise les sections suivantes:
 - Pesage_light_CAN_TM1
 - Pesage_light_CAN_TM2

- Le bit « Mode_Com_eNod_STB » autorise les sections suivantes:
 - Pesage_Complet_STB_TM1
 - Pesage_Complet_STB_TM2
 - Pesage_Com_STB

- Le bit « Mode_Com_eNod_STB_L » autorise les sections suivantes:
 - Pesage_light_STB_TM1
 - Pesage_light_STB_TM2

Deux écrans d'exploitation permettent de visualiser les deux pesages : TM1 TM2.

Exemple d'instance pour DFB Complet en Can automate



Toutes les variables de la structure eNod1 correspondent aux variables construites de façon générique dans l'onglet <Objets d'E/S> de la vue de configuration de l'esclave Can.

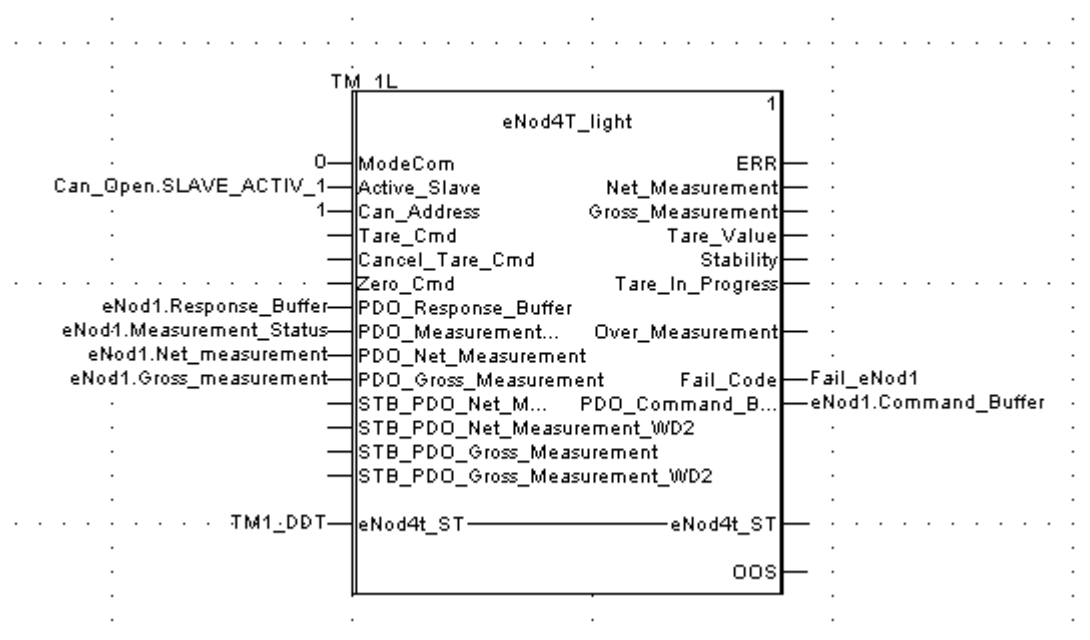
L'Active_Slave est renseigné par la variable issue de la structure montée par le bus CAN. (Voir le chapitre <configuration du bus CAN sous Unity>

Dans cette configuration, les pattes du DFB nommées STB_xx ne doivent pas être renseignées. Ne servent que pour l'utilisation du DFB en communication vers STB

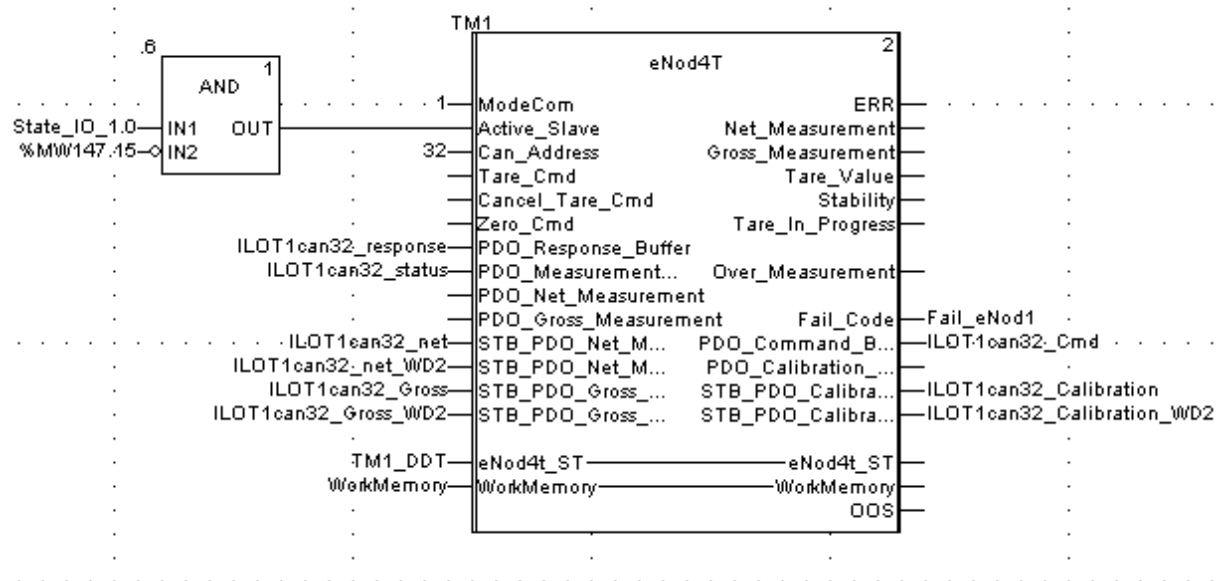
La variable ModeCom est positionnée à 0 pour le mode de communication en CAN direct.

Ce DFB est relié au DFB de communication par la variable WorkMemory

Exemple d'instance pour DFB Light en Can automate



Exemple d'instance pour DFB Complet en com derrière STB



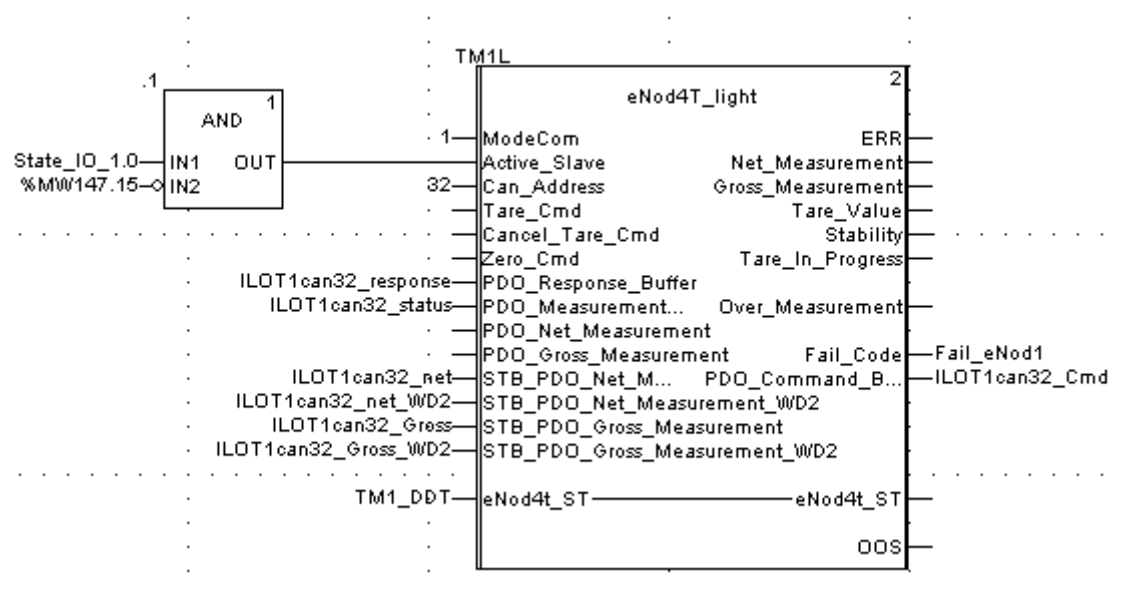
Les variables d'entrées sorties de ce DFB sont issues des variables récupérées par Advantys lors de la construction de l'ilot STB.

L'Active_Slave est renseigné par une combinaison de variables correspondant à l'activité de l'I/O scanning de la carte NOE et le retour de présence de l'esclave CAN derrière l'ilot STB.

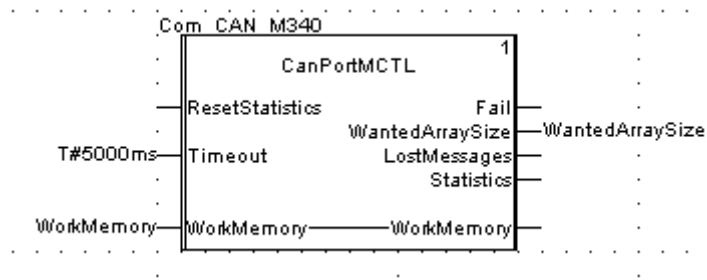
La variable ModeCom est positionnée à 1 pour le mode de communication à travers un ilot STB.

Ce DFB est relié au DFB de communication par la variable WorkMemory

Exemple d'instance pour DFB Light en com derrière STB



DFB de communication pour bus Can automate M340

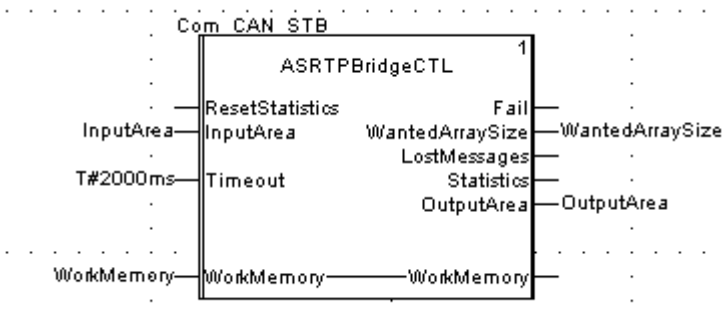


La variable tableau WorkMemory est le lien entre ce bloc et le DFB eNod4T utilisé en Com CAN

La variable WantedArraySize permet de dimensionner la longueur du tableau WorkMemory. Au 1^{er} scan automate cette variable contient la valeur qui doit être définie dans la longueur du tableau.

Pour utiliser les possibilités de diagnostic de communication de ce bloc, se référer à la documentation DPL

DFB de communication pour I/O scanning vers ilot STB



La variable tableau WorkMemory est le lien entre ce bloc et le DFB eNod4T utilisé en Com STB

La variable WantedArraySize permet de dimensionner la longueur du tableau WorkMemory. Au 1^{er} scan automate cette variable contient la valeur qui doit être définie dans la longueur du tableau.

La variable InputArea correspond aux 4 mots d'entrées définis dans le scanning d'I/O (voir ligne 2 § configuration carte NOE)

La variable OutputArea correspond aux 5 mots de sorties définis dans le scanning d'I/O (voir ligne 2 § configuration carte NOE)

Pour utiliser les possibilités de diagnostic de communication de ce bloc, se référer à la documentation DPL

Utilisation du composant sous IHM VijéoDesigner

Le principe de cette application est de récupérer la ou les structures Enod créés sous Unity pour les exploiter sous VijéoDesigner.

Pour créer votre application sous VijéoDesigner,

- il faut déclarer votre lien de communication par la création d'un pilote (ex ModbusTCP01) et d'un équipement (ex EquipementModbus01 [192.168.25.25]).
- Déclarer le fichier Unity contenant l'appliquatif automate sous <Variables\Variables de lien>, en sélectionnant le type de fichier en liaison (.STU)
- Sur sélection, le lien entre le fichier automate et VijéoDesigner s'effectue. Cela prend un peu de temps.
- Un écran s'affiche avec tout le contenu des variables de l'automate du projet Unity.
- Vous pouvez alors sélectionner les variables (structure) correspondant aux Enod à exploiter, cocher la boîte de dialogue <variable qui conservent le même nom> et cliquer sur Ajouter pour les inclure au dictionnaire du projet VijéoDesigner.

Nota : Le système VijéoDesigner doit être installé sous la même machine que Unity.

Construction des vues

Import dans la bibliothèque :

Dans la fenêtre Bibliothèque, cliquer sur Dossier, et sur la racine par clic droit importer le fichier <PesageScaime.ztc>

Import des objets de la bibliothèque :

Dans un écran de base de votre application, faire glisser de la bibliothèque l'afficheur de pesage.

Sous <fenêtre Popup> définir un nouveau groupe de fenêtre (ex Fenêtre1) de taille 300x400

Créer un nouvel écran et faire glisser de la bibliothèque (Pesage) la FacePlate 1

Recommencer l'opération avec les 3 autres faceplates dans 3 autres écrans

Sous fenêtre Popup définir un nouveau groupe de fenêtre (ex Fenêtre2) de taille 100x120.

Dans ce groupe de fenêtre popup, activer <le verrouillage de l'écran de base>

Créer trois nouveaux écrans et faire glisser de la bibliothèque la SousVue1, SousVue2 et SousVue3

Variable :

L'ensemble des vues d'affichages pour un Enod ont été importées dans le projet. Dans le dictionnaire de variable, la structure eNod4t_ST et Titre1 ont été importés.

Il faut maintenant affecter le vrai nom de variable à cette structure d'import. Pour cela, il faut détruire, si elle existe, la variable de nom final dans le dictionnaire. Ensuite il faut renommer la variable d'import eNod4t_ST par le nom de variable finale. Les vues de pesage sont affectées avec cette nouvelle variable. Pour affecter l'adressage de cette variable, il faut la détruire de nouveau, et la réimporter à partir de <Nouvelles variables à partir de l'équipement> en conservant le même nom.

Une autre variable a été importée par la bibliothèque ; Titre1. Cette variable sert à afficher un nom dans chacune des vues de l'objet. La renommer sous un autre nom et lui affecter une valeur initiale correspondant à votre titre de dénomination.

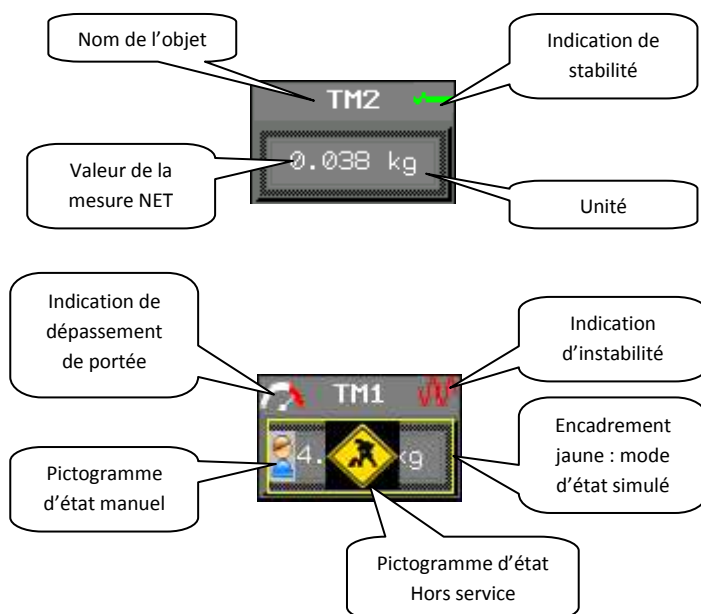
Pour le chainage des vues, il faut les modifier par rapport à votre appellation de fenêtre. (4 onglets supérieurs de chaque vue + appel de <portée>, <échelon> et <Réglage zéro> dans la vue de configuration)

Pour l'affichage de l'unité et du nombre de décimale, il faut rentrer dans chaque afficheur et les affecter.

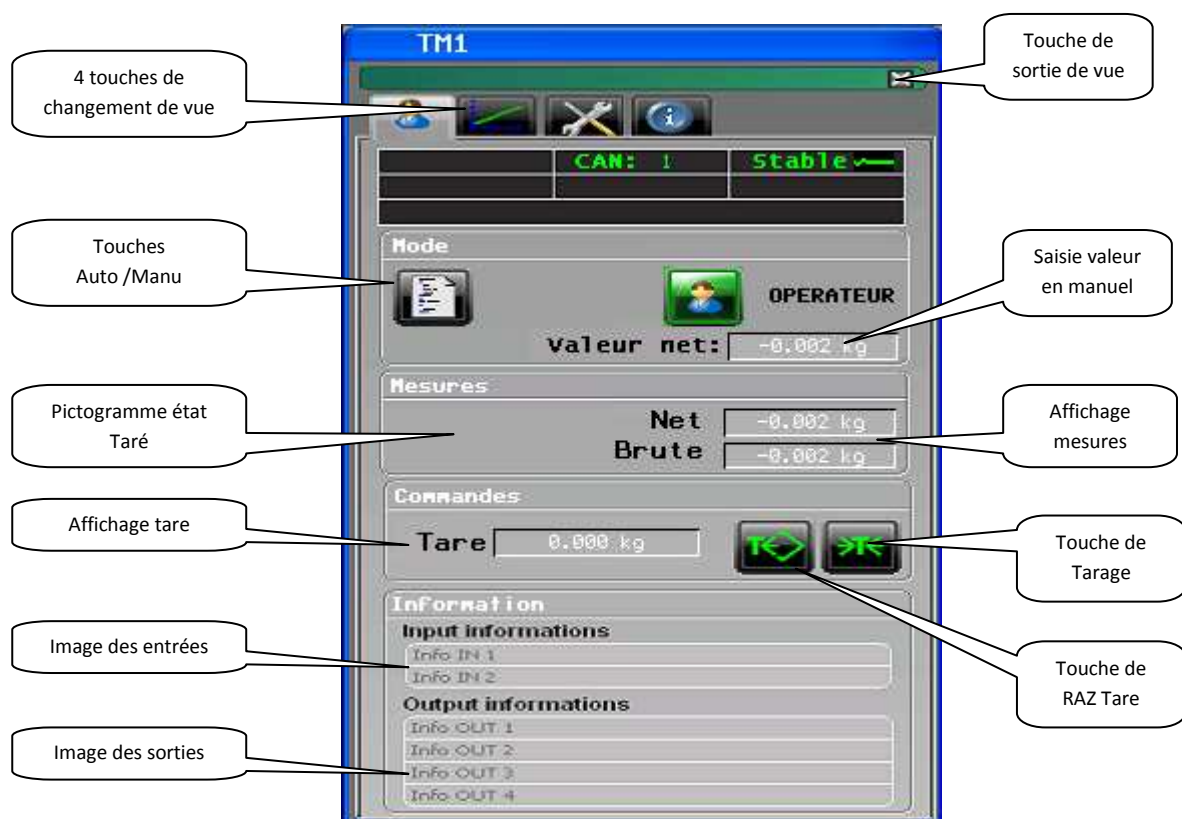
Pour créer un nouvel afficheur de pesage, procéder de la même façon dans des fenêtres différentes.

Description des vues :

Afficheur de synoptique :

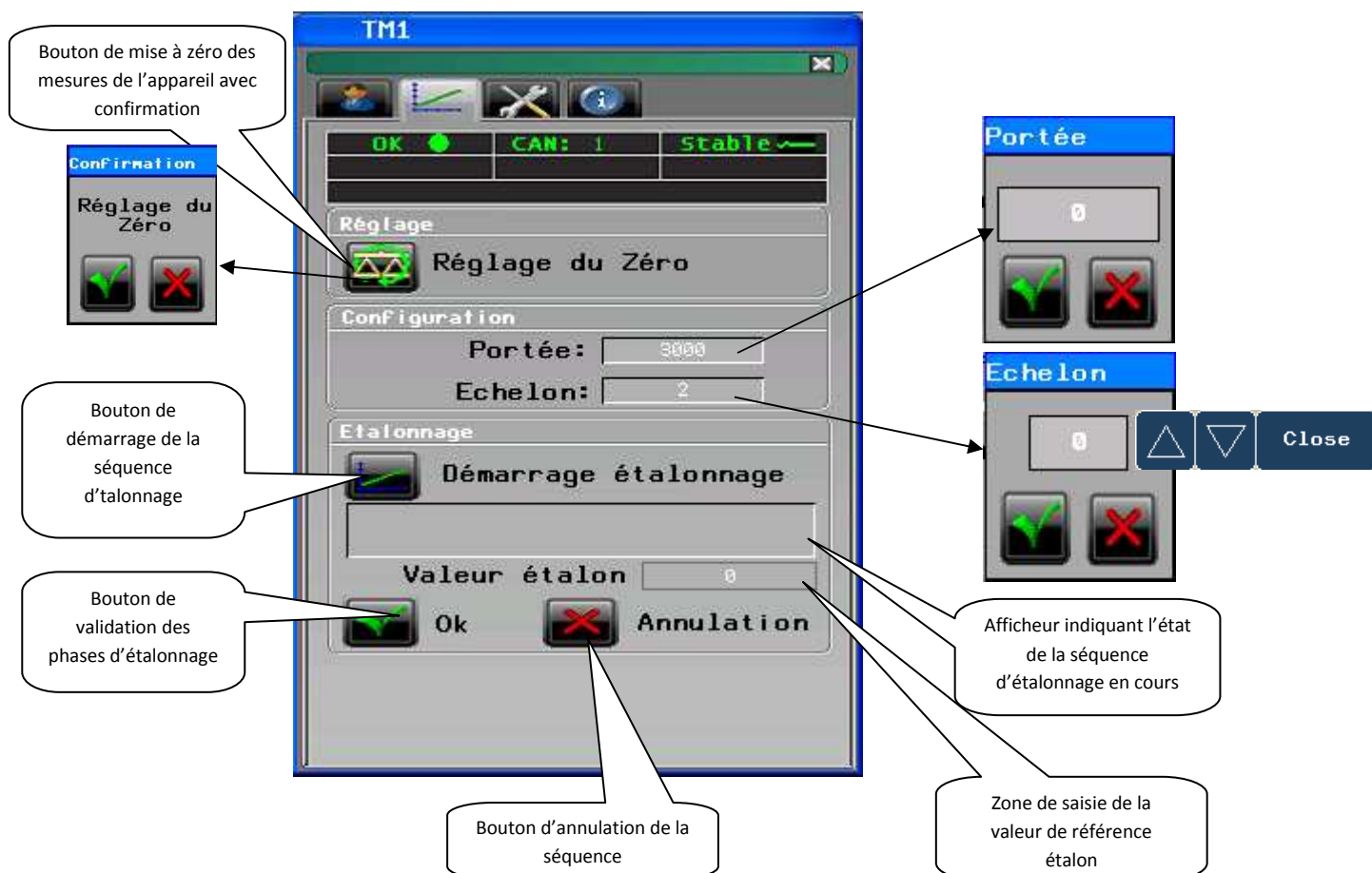


Face Plate 1, Conduite :



Face Plate 2, Réglage :

DFB COMPLET

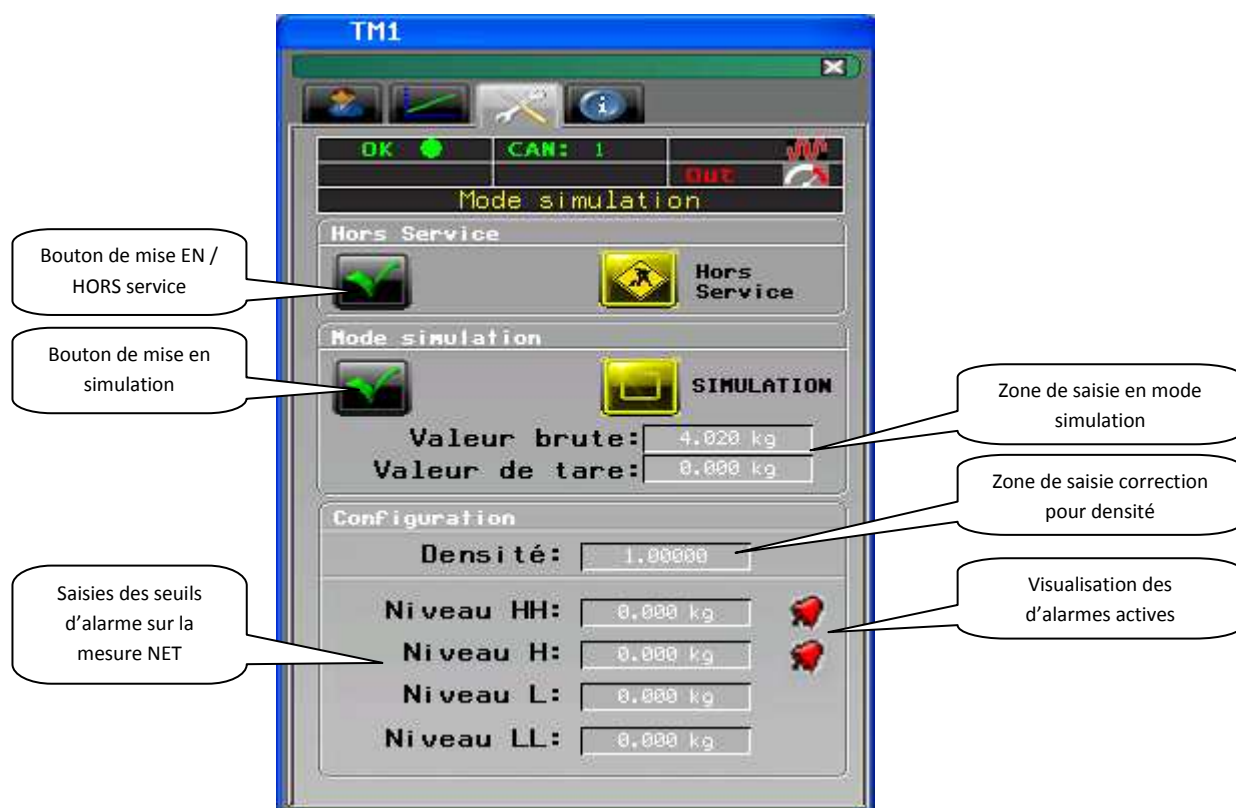


Nota : les valeurs de réglage de l'échelon sont définies par le constructeur (1, 2, 5, 10, 20, 50, 100) (réglage par ascenseur)

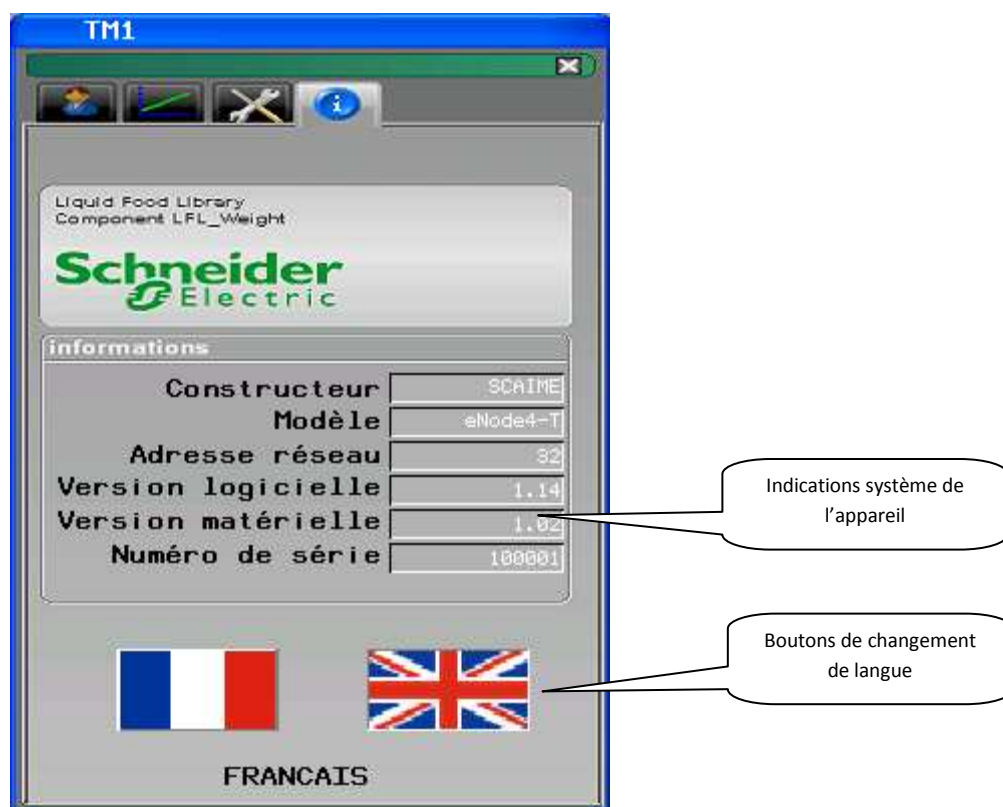
DFB SIMPLE



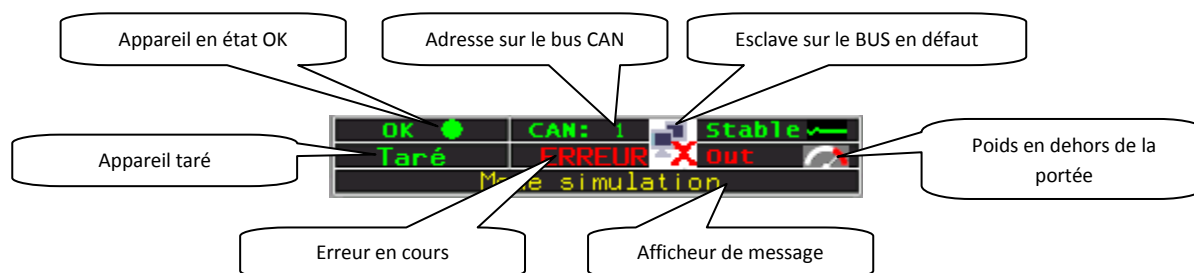
Face Plate 3, Fonctions:



Face Plate 4, système:

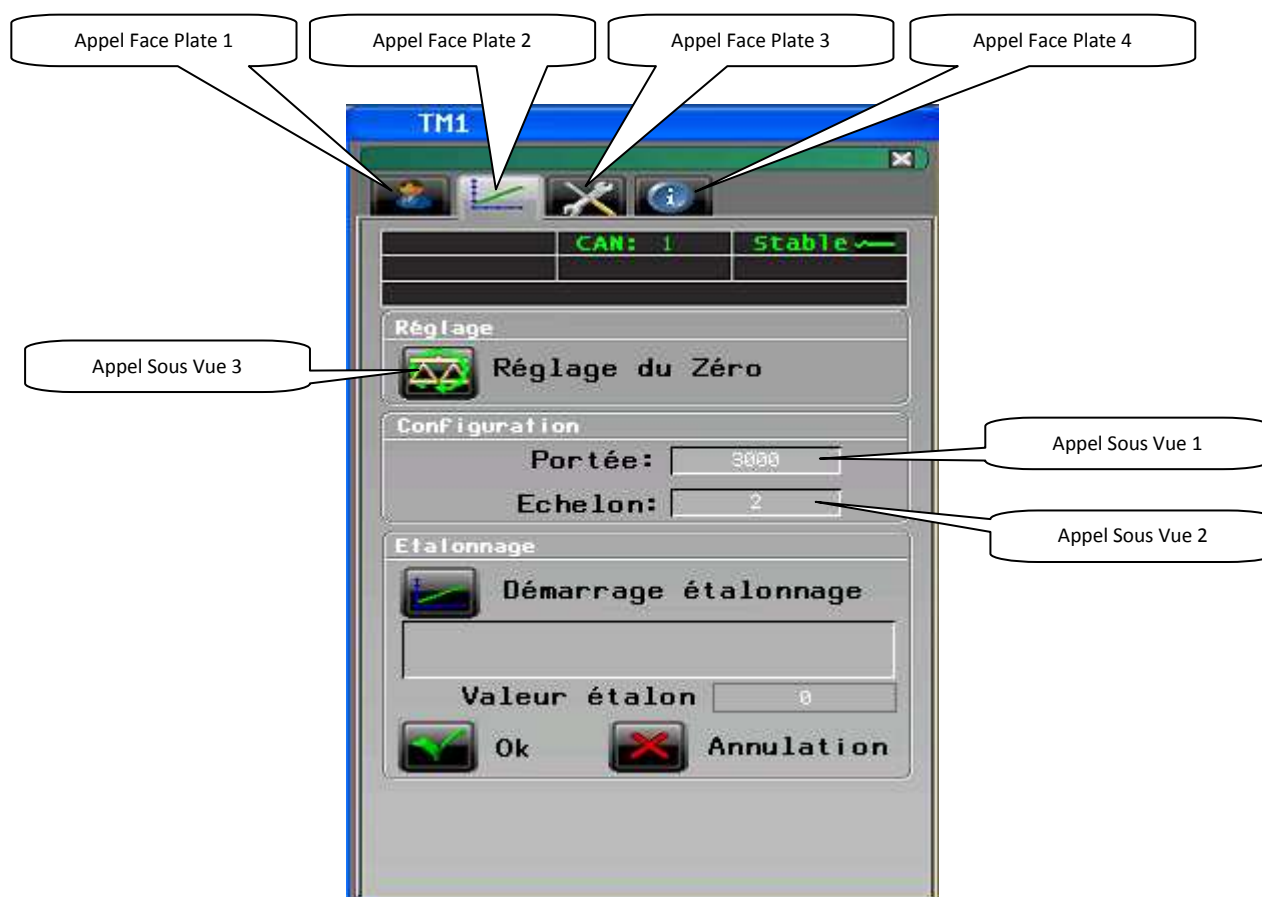
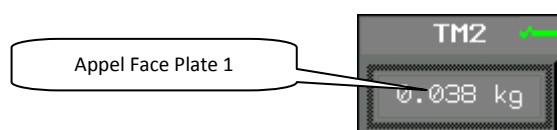


Fenêtre d'état:



Navigation dans les fenêtres:

Après avoir posé vos vues dans les différentes fenêtres, il faut réaffecter les chainages entre ces vues suivant les appels ci-dessous :



Pour tous vos appels de popup, définir la même position d'affichage pour que les vue se superposent :

