

Gestion des réseaux électriques MT

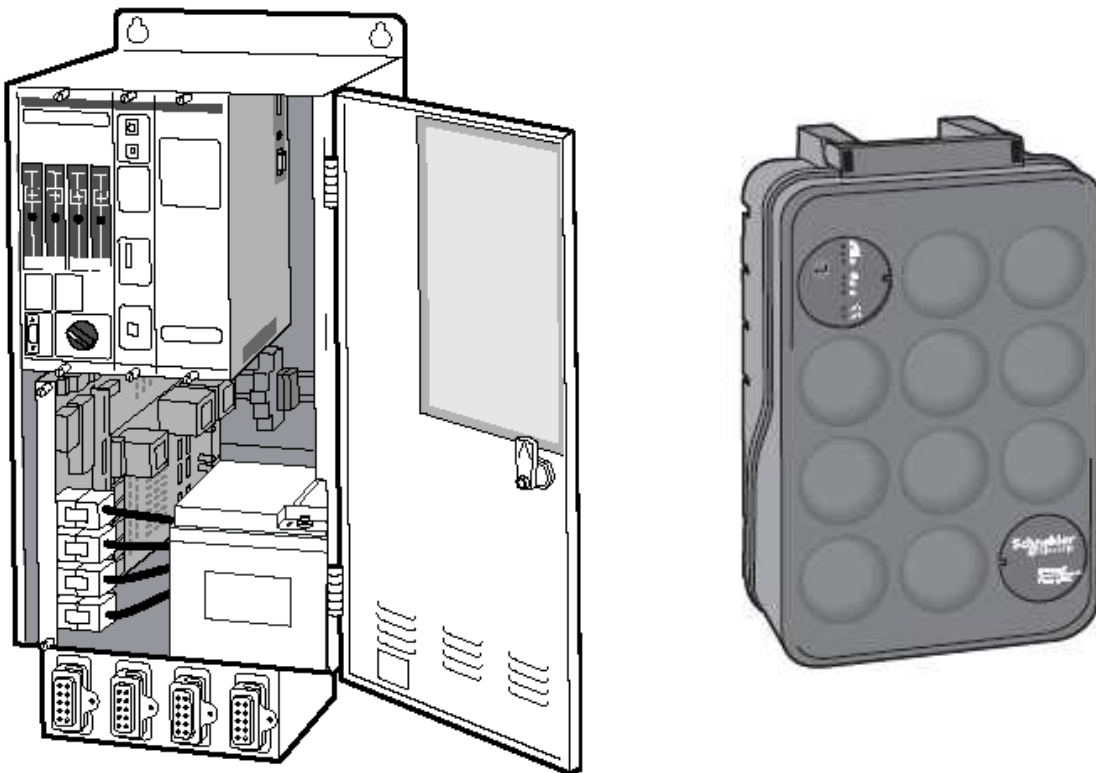
Gamme Easergy

Flair 200C et T200

Unités de téléconduite et de télésurveillance du poste MT

RTU Control et Formules

Additif au manuel utilisateur



1	AUTOMATE VIRTUEL	3
1.1	DESCRIPTION	3
1.2	INSTALLATION DU LOGICIEL RTU CONTROL	5
1.3	CHARGEMENT DE PROGRAMME PAR RTU CONTROL	6
1.3.1	Ouverture du projet	6
1.3.2	Connexion avec l'automate et chargement de programme	7
1.4	LES VARIABLES PARTAGEES	8
1.4.1	Partager une variable	8
1.4.2	Utilisation de tableaux	11
1.4.3	Divers	12
1.5	CONFIGURATION DE L'AUTOMATE	13
1.5.1	Configuration de l'automate	13
1.5.2	Chargement de programme par le serveur Web	14
1.5.3	Etat de marche de l'automate virtuel	15
1.5.4	Messages d'erreurs	15
1.5.5	Les variables RTU Control	16
1.5.6	Paramètres Programme	16
1.5.7	Configuration de la fonction "Peer to Peer" (communication inter équipements)	17
1.5.8	Traces automate	19
1.6	EXEMPLES	20
2	FORMULES DE CALCUL	21
2.1	PRESENTATION	21
2.2	SYNTAXES ET REGLES	22
2.2.1	Les opérandes	22
2.2.2	Les opérateurs	23
2.3	LES FONCTIONS	26
2.3.1	Les fonctions mathématiques	26
2.3.2	Les fonctions statistiques:	27
2.3.3	Les fonctions Logiques:	28
2.3.4	Les fonctions Horaires / Dates et heures:	30
2.3.5	Les autres fonctions:	31
2.4	CONFIGURATION	32
2.4.1	Création des variables	32
2.4.2	Saisie des formules	33

1 Automate virtuel

1.1 Description

Le T200 et le Flair 200C permettent de fournir en option des fonctions d'automatisme programmable IEC 61131-3 grâce à un automate virtuel embarqué.

Les applications typiques pouvant être créées avec l'automate programmable :

- Reconfiguration automatique de boucle MT (réseau auto-cicatrisant).
- Automatismes similaires à ceux du T200 (ATS, BTA) : ATS entre 2 T200 ou impliquant plus de 2 voies par exemple.
- Applications de délestage.
- Tout autre automatisme spécifique à un client.

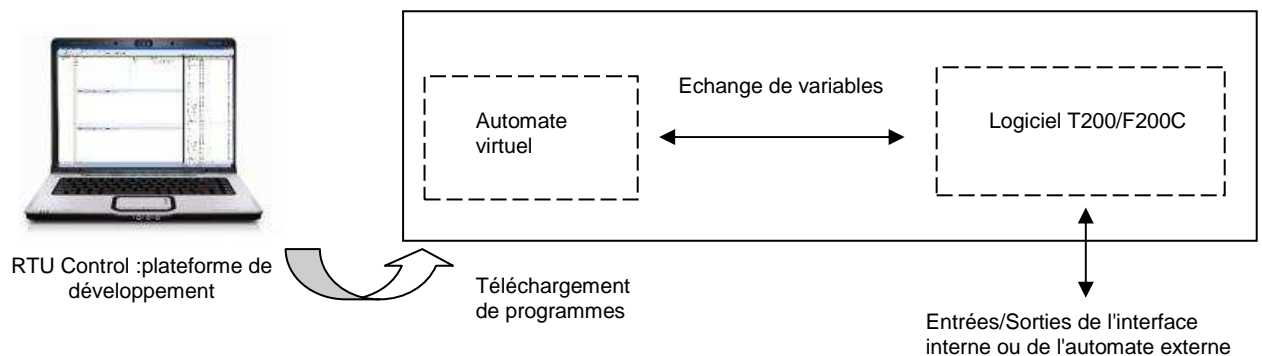
Attention : il n'est pas possible d'avoir simultanément dans le produit, l'option automate programmable IEC 61131-3 et les automatismes prédéfinis (ATS, Sectionnaliseur,...).

Une plateforme de développement permet de concevoir, simuler et tester sur cible des programmes écrits dans les langages automates suivants :

- SFC – Sequential Function Chart
- FBD – Function Block Diagram
- LD – Ladder Diagram
- ST – Structured Text
- IL – Instruction List

Cette plateforme de développement est le logiciel RTU Control, livré avec le produit.

Cette plateforme permet de générer un programme exécuté par l'automate virtuel. Cet automate virtuel, embarqué dans le produit, échange des données en lecture/écriture avec le logiciel de l'équipement, permettant ainsi d'exploiter les variables internes.



Les détails de la programmation IEC61131-3, ainsi que la maîtrise de la plateforme de développement ne seront pas détaillés dans ce document. La documentation associée est accessible depuis le logiciel RTU Control.

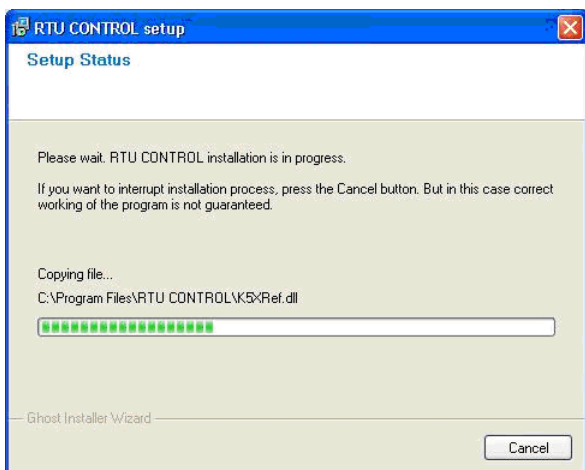
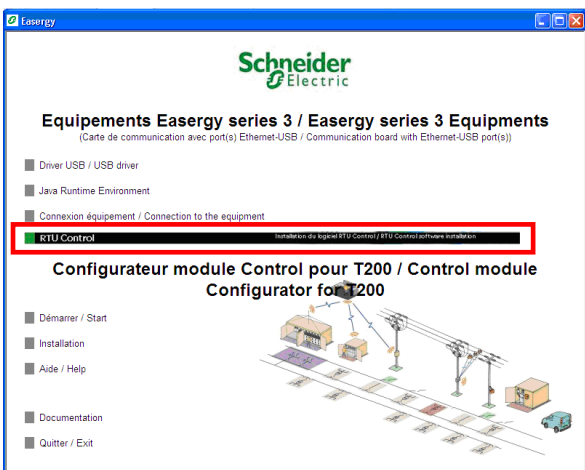
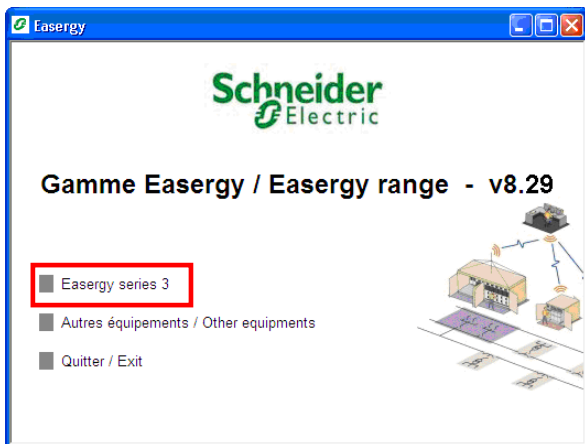
Néanmoins, nous allons expliquer dans ce document comment charger et utiliser un programme automate établi dans le serveur Web du T200/F200C ainsi qu'une prise en main sommaire de la plateforme et la gestion des paramètres et des variables associées.

1.2 Installation du logiciel RTU Control

Le logiciel RTU Control est inclus sur le CD Easergy fourni avec chaque équipement :

Pour installer le logiciel depuis le CD Easergy, suivre les instructions suivantes :

- Insérer le CD Easergy dans le lecteur du PC.
- Le logiciel du CD doit démarrer automatiquement et une fenêtre "Gamme Easergy – V8-xx" apparaît alors à l'écran.
- Cliquer ensuite sur le lien "Easergy series 3".
- Dans la nouvelle fenêtre qui apparaît, cliquer sur le lien "RTU Control" pour installer le logiciel.
- Suivre ensuite les différentes étapes d'installation, à priori sans modifier les options proposées.
- Attendre la fin d'installation, puis quitter le programme d'installation.



Une fois le logiciel RTU Control installé, il convient maintenant de charger un programme établi dans l'automate.

Deux méthodes pour charger un programme automate dans l'équipement T200/F200C :

- **Par la plateforme RTU Control**, à condition que le logiciel RTU Control soit installé sur le PC connecté au T200/F200C et que l'équipement soit sous-tension avec la fonction automate démarrée.
- **Par le Serveur Web embarqué** du T200/F200C, à condition de disposer du programme automate sous forme de fichier avec extension *.cod.

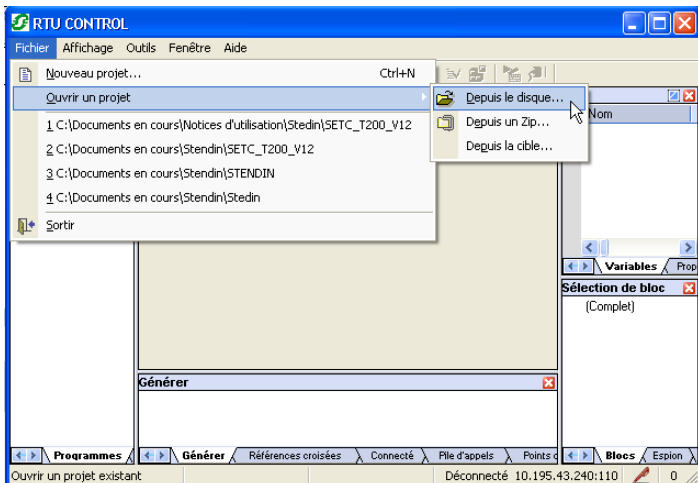
1.3 Chargement de programme par RTU Control

Ce chapitre montre la première méthode de chargement de programme depuis le logiciel automate "RTU control". Cela suppose que le logiciel RTU Control soit installé sur le PC qui est connecté au serveur Web du T200/F200C, et que le projet correspondant soit disponible.

1.3.1 Ouverture du projet

L'ouverture du projet s'effectue depuis l'interface du programme RTU Control :

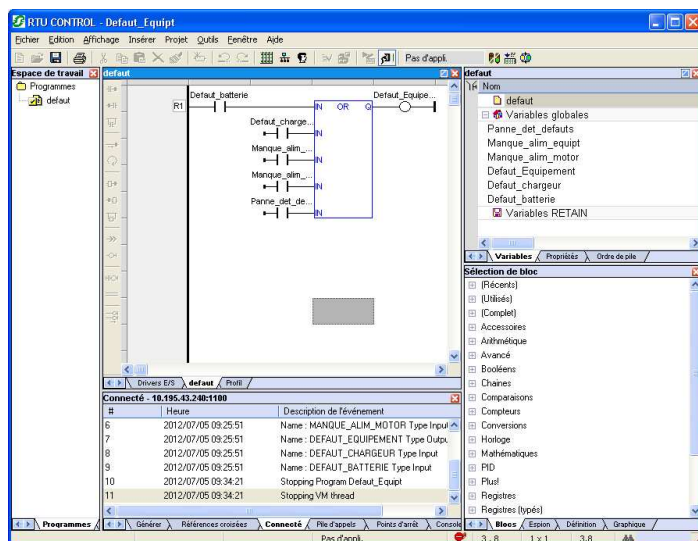
Menu Fichier → Ouvrir un projet → Depuis le disque



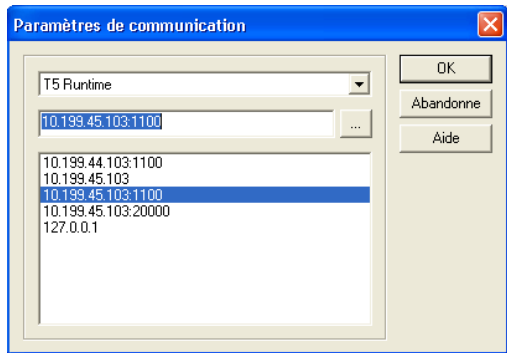
Sélection du programme à ouvrir dans RTU Control

Choisir ensuite le projet à charger.

Le projet s'ouvre alors dans la plateforme de développement.



Programme ouvert dans la plateforme



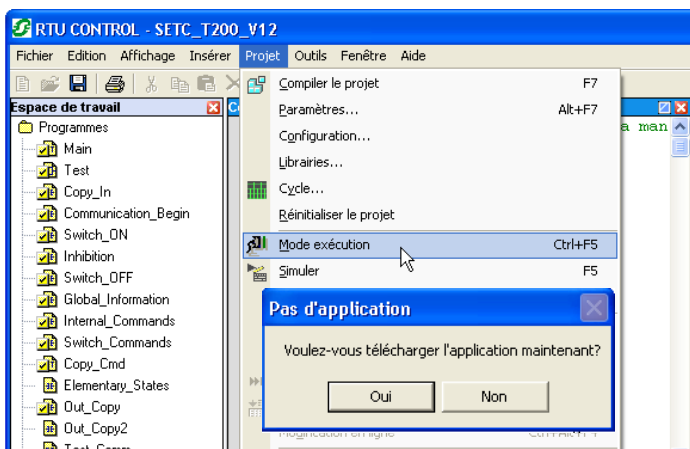
1.3.2 Connexion avec l'automate et chargement de programme

Une fois le projet correctement ouvert, il faut configurer la connexion avec l'automate virtuel.

Dans la plateforme de développement, ouvrir le lien suivant :

Menu Outils → Paramètres de communication :

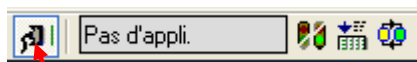
- Sélectionner T5 Runtime
- Entrer l'adresse IP de votre T200/F200C suivi du port d'écoute de l'automate virtuel (configuré précédemment sur le RTU ; 1100 par défaut). Les deux champs sont séparés par le caractère ':'
- La fenêtre ci-joint montre la configuration pour un T200/F200C dont l'adresse IP est 10.199.45.103 et l'automate virtuel configuré pour écouter sur le port 1100
- Cliquer sur OK



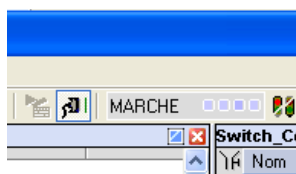
Pour connecter la plateforme de développement au T200/F200C, et télécharger l'application dans la cible, cliquez sur :

Menu Projet / Mode exécution / Télécharger l'application

Page de téléchargement d'application



Bouton de téléchargement d'application



Statut "Marche" du programme dans RTU Control

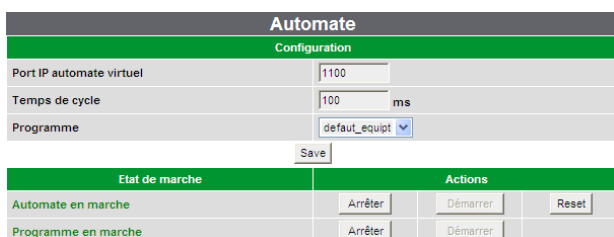
Pour arrêter l'application, cliquer sur le même lien.

Nota : cette commande est accessible également par un bouton dans la barre des tâches.

Le programme est désormais prêt à fonctionner dans l'équipement. Celui-ci apparaît alors avec le statut "Marche" :

- Dans le logiciel RTU Control.
- Dans la page "Automate" du serveur Web du T200/F200C.

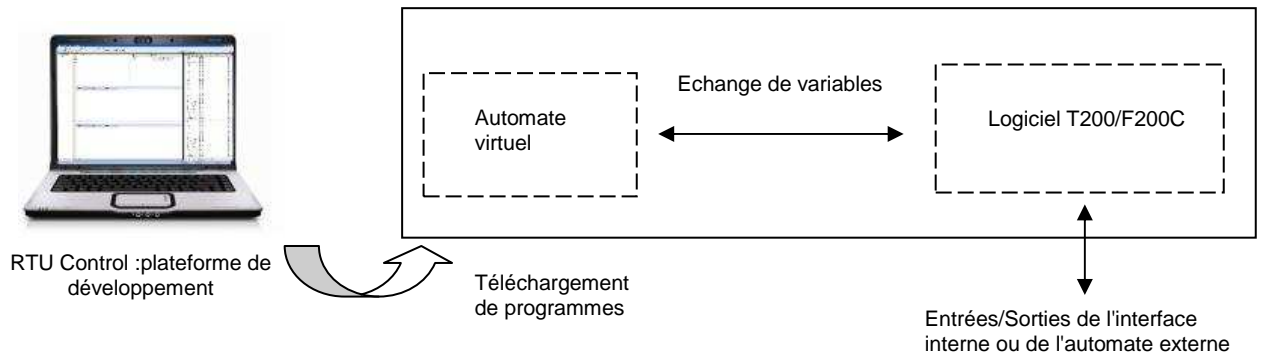
Nota : le détail concernant cette page "Automate" sera expliqué ultérieurement dans ce document.



Statut "Marche" du programme dans la page "Automate" du serveur Web

1.4 Les variables partagées

L'automate virtuel inclus des variables internes et peut également partager des variables avec le T200/F200C. Une variable partagée sur la plateforme sera automatiquement liée à une variable T200/F200C au démarrage de l'application.



Les variables pouvant être partagées sont des variables de type **Globales**. Celles-ci peuvent être utilisées n'importe où dans le programme.

Le lien entre les variables de l'automate virtuel et les variables de T200/F200C peut se faire selon différents niveaux :

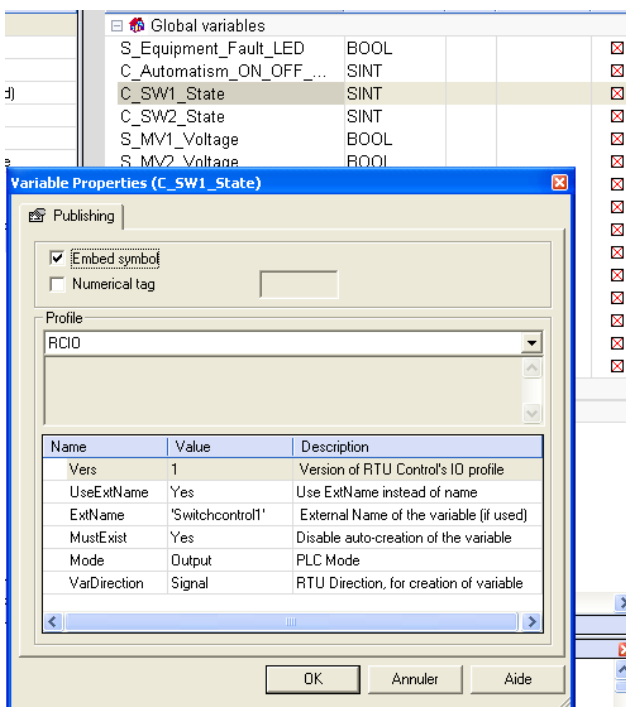
- Stricte identité du nom de la variable ou utilisation d'un nom différent.
- Création automatique de variable dans le T200/F200C, ou utilisation de variables existantes.
- En cas de création, choix du mode d'échange au niveau du T200/F200C.

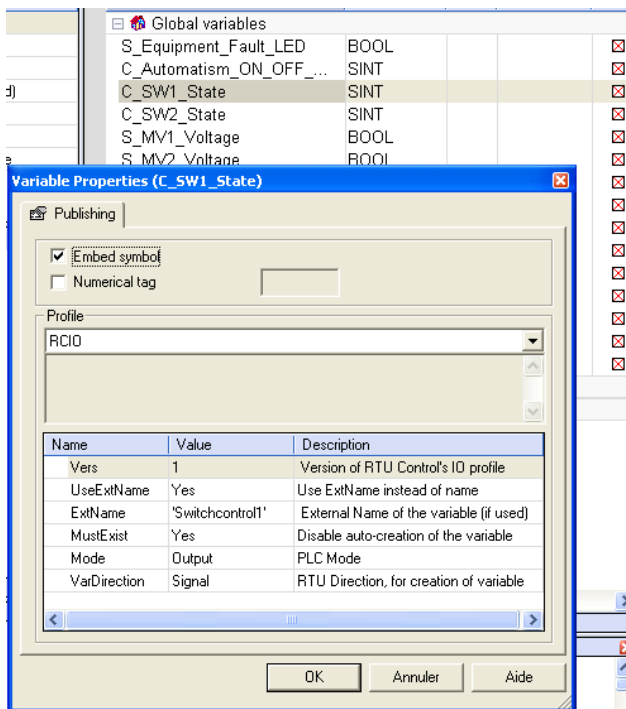
1.4.1 Partager une variable

Depuis la plateforme de développement, le programme étant arrêté, faire un clic droit sur la variable que vous souhaitez partager et cliquer sur "Propriétés".

Afin de partager la variable sélectionnée avec le T200/F200C :

- Cocher la case "Embarquer les symboles"
- Choisir le profil RCIO dans la liste des profils
- Compléter les paramètres suivant le type de lien souhaité (voir ci-après) :





UseExtName : choix d'utiliser le nom de la variable déclaré dans l'atelier de programmation lorsque celui-ci est différent de celui défini dans le T200/F200C ("Yes"), ou d'imposer d'utiliser le même nom que celui de la variable du T200C/F200C ("No" par défaut).

ExtName : nom de la variable T200/F200C associée à la variable de l'atelier. Ce paramètre permet de faire la relation entre la variable du T200/F200C et celle correspondante de l'atelier, si elles ont des noms différents (utilisé si "UseExtName" = "Yes"). La longueur du champ est limitée à 15 caractères

MustExist : choix d'autoriser la création de variable dans le T200/F200C ("No", par défaut) ou d'imposer que la variable existe.

Au démarrage du programme, le T200/F200C récupère toutes les variables partagées par l'automate. Si la variable associée existe, le lien est effectué. Si elle n'existe pas, on peut autoriser le T200/F200C à la créer ou lui interdire. Ce paramètre permet un plus grand contrôle si on veut utiliser des variables existantes

Mode : [Input/Output] Mode d'échange de données de l'automate.

Le mode est utilisé pour déterminer si, du point de vue de l'automate, une variable est en lecture seule ou en lecture/écriture. Les variables définies en Input seront en lecture seule dans l'application automate. Les variables définies comme Output seront en lecture/écriture.

Pour les deux modes, la valeur courante sera lue au début de chaque cycle automate. Seuls les outputs seront mis à jour à la fin de chaque cycle.

Notez que les entrées physiques (comme les TSS) sont automatiquement en lecture seule.

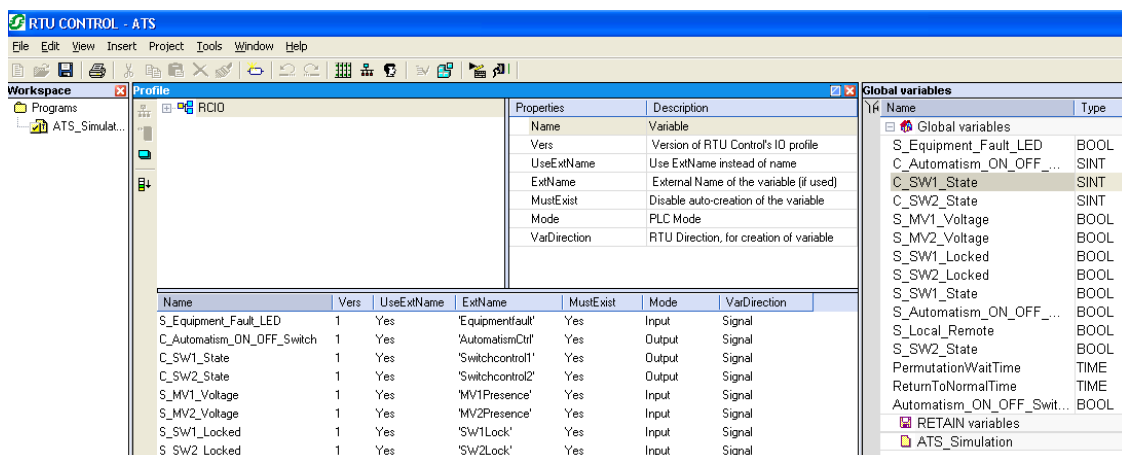
VarDirection: [Signal/Command] Type de variable, du point de vue T200/F200C. Utilisé seulement dans le cas d'une création automatique de variable par le T200/F200C (MustExist = no et si la variable correspondante n'existe pas).

Une variable définie comme "Signal" sera créée comme une entrée (DI ou AI). Une variable définie comme "Command" sera créée comme une sortie (DO ou AO). Les sorties sont modifiables par l'utilisateur via l'interface ou le protocole (exemple : paramètre, consigne, option).

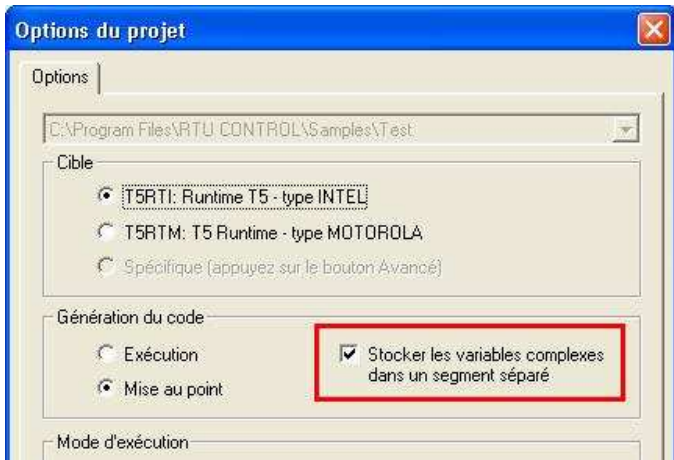
Attention : la notion d'Input/Output au niveau du T200/F200C est différente de celle au niveau de l'automate virtuel, où un paramètre est une des "input" d'un bloc fonction par exemple.

Le programme devra ensuite être recompilé et renvoyé sur la cible afin que la modification soit prise en compte.

La gestion des paramètres des variables partagées peut se faire via l'explorateur de profils (icône "Ouvrir Profils") :



L'explorateur permet d'avoir une vue d'ensemble des variables associées au profil RCIO, et de les modifier. Il permet aussi d'associer facilement une variable au profil en utilisant la méthode "drag and drop" (cliquer sur une variable, puis en maintenant le click la déplacer vers le profil).



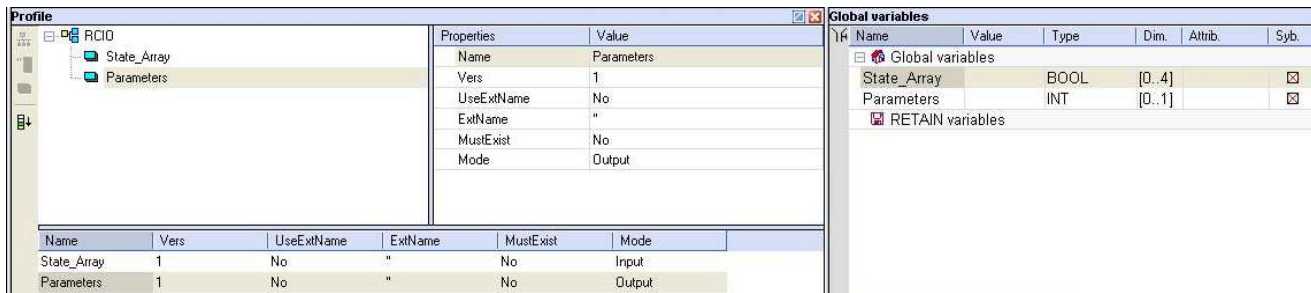
1.4.2 Utilisation de tableaux

Des tableaux complexes de variables ou de structures peuvent être utilisés en interne du programme. Dans le cas de variables partagées avec le T200/F200C, l'utilisation des tableaux est limitée aux tableaux de type simple à 1 dimension.

Dans ce cas il est nécessaire de cocher l'option "Stocker les variables complexes dans un segment séparé" dans les options du projet (Projet\Paramètres...)

Dans ce cas, un tableau de N variables sera associé à N variables dans le T200/F200C. Le nom de ces variables sera NomTableau_@x avec x de 1 à N.

Exemple : Création de tableaux "State_Array" de 5 booléens (BOOL), et "Parameters" de 2 entiers (INT)



Les variables ainsi créées dans le T200/F200C sont State_Array_@1 à State_Array_@5 et Parameters_@1 à Parameters_@2.

STATE ARRAY @1		DI18	-
STATE ARRAY @2		DI19	-
STATE ARRAY @3		DI20	-
STATE ARRAY @4		DI21	-
STATE ARRAY @5		DI22	-
PARAMETERS @1		AO13	-

1.4.3 Divers

Conseils concernant les noms des variables :

- N'utilisez que des majuscules.
- Ne donnez pas le même nom à deux variables différentes.
- N'utilisez pas de nombre entier comme nom de variable (par exemple "5").

Type de variables :

Le T200/F200C supporte les types suivants : BOOL, SINT, INT, BYTE, WORD, REAL

Attention : les types doivent être compatibles.

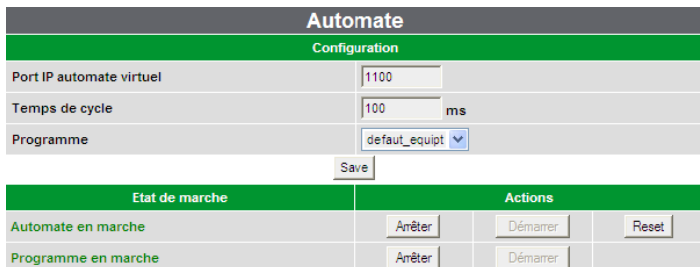
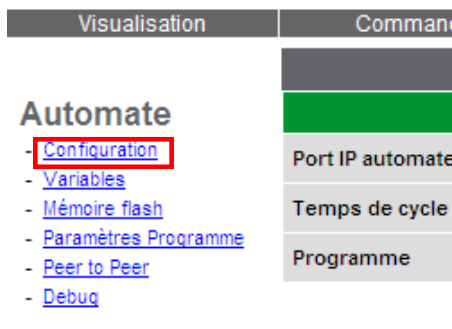
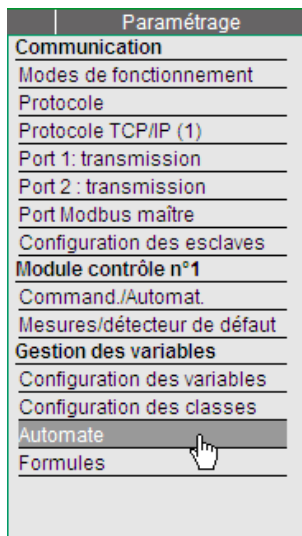
BOOL <--> TSS, TSL, DI ou DO

SINT,INT, DINT, WORD, DWORD, REAL <--> TM, AI ou AO.

DINT <--> Compteurs

Important : les TSD/TCD ne sont pas des variables binaires mais des variables numériques (BYTE...), avec trois valeurs possibles.

Valeur TSD/TCD	Etat
0	indéfini
1	Off
2	On



Page de configuration de l'Automate

1.5 Configuration de l'automate

Dans la même partie "Automate" du serveur Web, d'autres pages sont également liées à l'automate, notamment celle concernant la configuration des paramètres de fonctionnement.

1.5.1 Configuration de l'automate

Cliquer sur le lien "Configuration" pour accéder à la page de configuration des paramètres de l'automate.

Cette page contient des éléments de configuration, des informations sur l'état de marche de l'automate virtuel et du programme associé, et éventuellement des messages d'erreurs.

1.5.1.1 Port IP Automate virtuel :

port IP du T200/F200C par lequel la connexion à l'atelier de programmation sera réalisée.
Valeur par défaut : 1100

1.5.1.2 Temps de cycle :

Cette valeur en millisecondes est la période à laquelle sera exécuté un cycle de l'automate virtuel. Cette valeur en ms devra être comprise entre 100 et 10000ms (10 secondes).

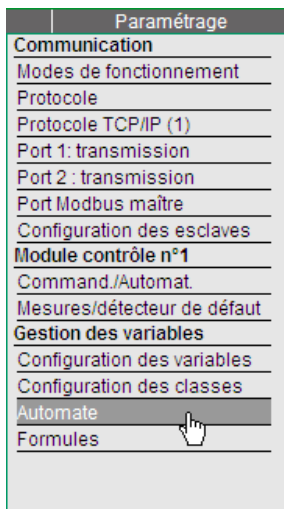
1.5.1.3 Programme :

Nom du programme chargé et exécuté par l'automate virtuel, parmi tous les programmes précédemment chargés en mémoire flash.

Le T200/F200C permet de conserver plusieurs programmes en mémoire, mais un seul peut être exécuté.

(Voir le chapitre "Chargement de programme par le serveur Web" pour plus d'informations.)

Nota : l'option de démarrage de l'automate n'est pas configurable. Ce démarrage est toujours effectué "à froid". En conséquence, toutes les variables internes de l'automate seront réinitialisées à leurs valeurs initiales au moment du démarrage de l'automate. Les valeur des variables « retain » ne sont pas utilisées.



1.5.2 Chargement de programme par le serveur Web

Ce chapitre explique la deuxième méthode de chargement de programme automate, depuis l'interface Web du T200/F200C.

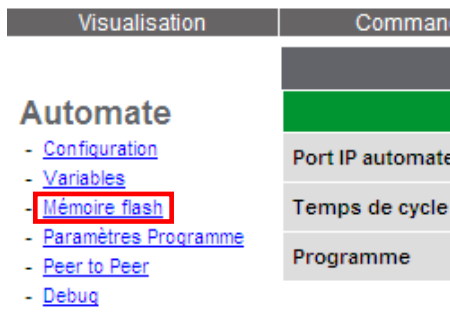
Pour pouvoir effectuer ce chargement, il convient tout d'abord de disposer du fichier *.cod, relatif au programme à charger.

Pour accéder à la page Automate du serveur Web embarqué, cliquer sur le lien correspondant ("*Automate*").



Page d'activation du code RTU Control

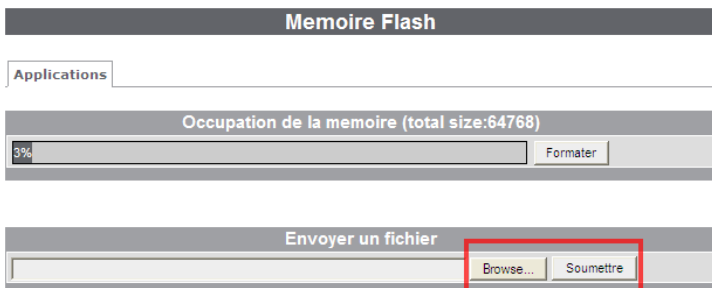
Si le logiciel RTU Control n'a jamais été activé dans l'équipement, entrer le code d'activation (Ce code est fourni par Schneider Telecontrol).



Cliquer ensuite sur le lien "*Mémoire flash*" pour accéder à la page de gestion des fichiers et de la mémoire de l'automate.

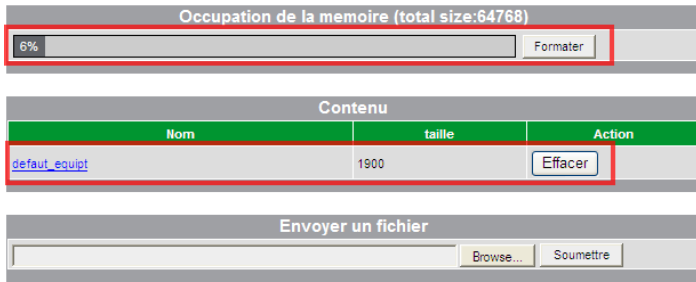
Cette page gère les programmes de l'application à charger ou déjà chargés dans la mémoire flash.

Cliquer sur le bouton "*Browse*" pour sélectionner le fichier du programme automate (extension « *.cod ») à télécharger.



Page "Mémoire Flash"

Cliquer ensuite sur le bouton "*Soumettre*" pour prendre en compte le programme dans l'automate virtuel.



Page "Mémoire Flash"

Le programme est prêt à être utilisé.

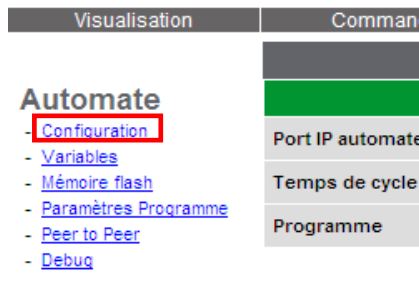
Le nom du programme s'affiche alors dans la partie "Contenu" avec l'indication de la taille du fichier.

La possibilité est donnée d'effacer ce programme dans l'automate par le bouton "Effacer".

Il est possible de télécharger ce fichier sous forme d'extension *.cod, sur le disque dur du PC, en cliquant sur le nom du programme.

La taille mémoire occupée par le programme est affichée également en pourcentage.

Il est conseillé de formater la mémoire avant la première utilisation, grâce au bouton "Formater".



1.5.3 Etat de marche de l'automate virtuel

1.5.3.1 Etat de l'automate virtuel

Dans la page "Automate/configuration", la première ligne de la section "Etat de marche" indique si l'automate virtuel est en fonctionnement ou non.

L'automate virtuel peut être arrêté ou démarré grâce aux boutons correspondants.

Ce paramètre est sauvegardé (en cas de redémarrage du T200/F200C, le module sera dans le dernier état demandé).

Nota : pour communiquer avec l'atelier de programmation, le module doit être en marche.

1.5.3.2 Etat du programme

Si l'automate virtuel est en marche, une seconde ligne apparaît, indiquant l'état de marche du programme de l'automate virtuel. Il est possible de stopper ou de démarrer le programme depuis l'interface.

Nota : suivant le nombre de variables dans le programme, le temps de démarrage de celui-ci peut-être plus ou moins long.

1.5.4 Messages d'erreurs

Si le démarrage de la machine virtuelle a échoué, un (ou des) message(s) d'erreur(s) peuvent apparaître.

Etat de marche	Actions		
Automate arrêté	Arrêter	Démarrer	Reset

Automate arrêté

Etat de marche	Actions		
Automate en marche	Arrêter	Démarrer	Reset
Programme arrêté	Arrêter	Démarrer	

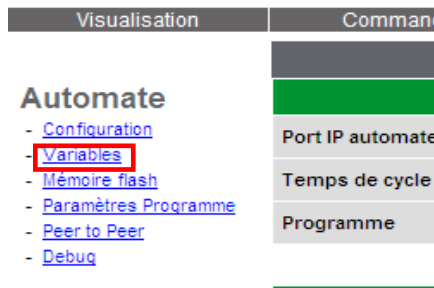
Automate en marche, programme arrêté

Etat de marche	Actions		
Automate en marche	Arrêter	Démarrer	Reset
Programme en marche	Arrêter	Démarrer	

Automate et programme en marche

Erreurs
Erreur au chargement du programme
Erreur lors de la sauvegarde du programme
Erreur lors de la sauvegarde du changement de code

Différents messages d'erreurs



1.5.5 Les variables RTU Control

Le lien "Variables" de la page "Automate" permet de visualiser et de gérer les variables internes d'échange entre l'automate RTU Control et le T200/F200C.

Les variables de cette page peuvent être :

- Des variables **existantes** du T200/F200C définies pour être partagées avec RTU Control, car celles-ci sont utilisées dans le programme automate.
- Des variables **virtuelles créées** spécifiquement pour le programme RTU Control.

L'état ou la valeur de certaines variables peut être changée depuis cette page, par les boutons "Change value".

Variables automate				
Nom	Type	Adresse Interne	Valeur	Action
Manque_alm_motor	Boolean	TSS19	FALSE	
Manque_alm_equip	Boolean	TSS20	FALSE	
Defaut_chargeur	Boolean	TSS21	FALSE	
Defaut_batterie	Boolean	TSS22	FALSE	
Panne_det_defaults	Boolean	TSS47	FALSE	
Defaut_Equipement	Boolean	DI2	FALSE	Change Value

Page "Variables automate"

Configuration des signalisations

Paramètres généraux

Nom de la variable: Default_Equipement Type :

Adresse logique: DI2 Classe: Defaults internes T200

Adresse interne: 0,0 Adresse:

Définition état Actif/Inactif: Actif (1): Oui Inactif (0):

Configuration Communication

Topic: **Virtuel** Type: Bit simple

Configuration des enregistrements

Sur changement d'état: Sur actif: Sur inactif:

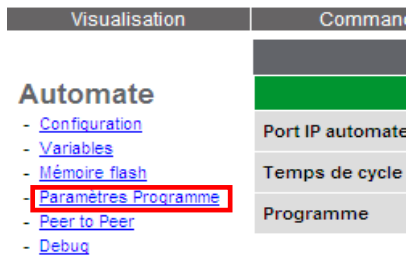
Page de configuration d'une variable virtuelle

1.5.6 Paramètres Programme

Une liste de paramètres utilisables par le programme automate peut être créée et modifiée dans cette page.

Les valeurs de ces paramètres sont sauvegardés avec la configuration Easergy sans avoir à changer le programme automate.

Ces paramètres peuvent être numériques, inclus dans une plage de valeurs, ou faire partie d'une liste qui associe un texte à une valeur numérique.



Paramètres Programme PLC

Nom	Valeur	Adresse logique de l'Ao associée
SHG_TOVloss_s	120	AO1 - SHG_TOVloss
SHG_TOCommand_s	17	AO2 - SHG_TOCommand
SHG_TOAutoComms_s	900	AO3 - SHG_TOAutoComms
SHG_Comm_NodeType	2 - 2: Middle Node	AO4 - SHG_Comm_NodeType
SHG_Comm_Path	1 - Path B	AO5 - SHG_Comm_Path
SHG_Master_Node	0 - No	AO6 - SHG_Master_Node
SHG_CheckVoltage	1 - Yes	AO7 - SHG_CheckVoltage
Corresp_Switch_A	1	AO8 - Corresp_Switch_A
Corresp_Switch_B	2	AO9 - Corresp_Switch_B
Corresp_Switch_C	3	AO10 - Corresp_Switch_C
SHGClosingPriority	1 - B,A,C	AO11 - SHGClosingPriority
SHG_StartDelay	0	AO12 - SHG_StartDelay
FeederLimit	630	AO13 - FeederLimit
Plc Parameter 14		AO1 - SHG_TOVloss

Enregistrer

Exemple de paramètres Programme

Pour transmettre les valeurs des paramètres à l'application automate, il est nécessaire de créer des commandes analogiques (Analog Output) de type virtuel. Chaque paramètre est lié à un AO, qui doit être défini comme une variable partagée dans le programme automate.

La valeur du paramètre est écrite sur la variable:

- quand l'équipement démarre (démarrage de la machine virtuelle)
- quand de nouvelles valeurs sont entrées dans cette page web
- après un changement de configuration du MODBUS maître et/ou des variables virtuelles.

Paramètres Programme PLC

Paramètres généraux

Nom du paramètre: SHG_TOAutoComms s
 Valeur: 900
 AO associée: A03 - SHG_TOAutoComms

Type: Numérique Liste
 Valeur Min: 0 Valeur Max: 65535

Paramètres Programme PLC

Paramètres généraux

Nom du paramètre: SHG_Comm_NodeType
 Valeur: 2 - 2: Middle Node
 AO associée: A04 - SHG_Comm_NodeType

Type: Numérique Liste
 Valeur Min: 0 Valeur Max: 3

Configuration de la liste

Nom	Valeur
0: Not configured	0
1: End Node	1
2: Middle Node	2
3: Tee Node	3

Ajouter un nouvel élément...

Jusqu'à 20 paramètres peuvent être définis. Pour chacun d'entre eux, on peut configurer:

- le nom
- la valeur active
- la variable analogique associée
- le type « Numérique » ou « Liste »
- la plage de valeurs autorisée (Min et Max)
- pour un paramètre de type 'Liste', les éléments de la liste peuvent être ajoutés et édités

1.5.7 Configuration de la fonction "Peer to Peer" (communication inter équipements)

Cette fonctionnalité permet de mettre en place une communication transversale entre deux (ou plus) équipements T200 ou Flair 200C, indépendamment de la communication « vers le haut » vers un superviseur. Les équipements doivent être installés sur un même réseau TCP/IP (Ethernet ou GPRS/UMTS).

Cette page montre la configuration des tables de transfert qui sont utilisés pour envoyer des informations entre deux équipements voisins.

Principe du Peer to Peer:

La fonction Peer to Peer permet de copier les valeurs d'un ensemble de variables d'un équipement vers un autre en utilisant le protocole MODBUS TCP et la fonction « write multiple registers ». Ce fonctionnement est similaire au concept « dataset » existant dans d'autres protocoles.

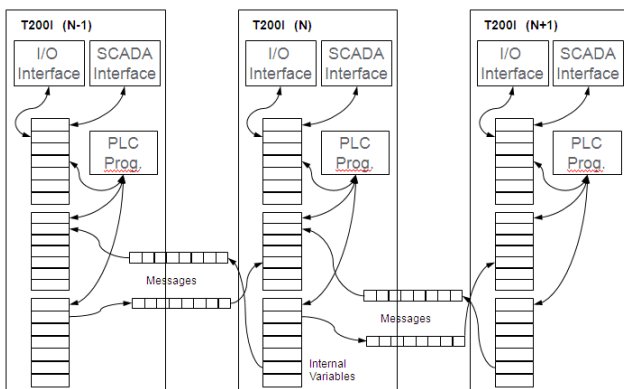
Chaque dataset est défini comme une table d'échange entre les deux équipements. Pour celui qui envoie les données, la table est définie en type « Write », et pour celui qui les reçoit, elle est définie en type « Read ». Chaque table de transfert définit un intervalle d'adresses MODBUS (Début + longueur). Une variable fait partie de ce dataset si son adresse Peer-to-Peer fait partie de cet intervalle.

Pour les tables « Write », les variables peuvent être de n'importe quel type. Pour les tables « Read », les variables doivent être de type DO ou AO.

Le programme automate peut déclencher l'envoi des tables de type « Write », et est prévenu lorsque les tables « Read » sont mises à jour.

Le processus utilise des variables de contrôle pour indiquer le succès ou l'échec de chaque transfert. Ces variables peuvent être utilisées par le programme automate pour répéter le transfert si besoin.

Visualisation	Commande
<p>Automate</p> <ul style="list-style-type: none"> - Configuration - Variables - Mémoire flash - Paramètres Programme <li style="border: 2px solid red; padding: 2px;">- Peer to Peer - Debug 	<p style="text-align: center; background-color: #cccccc;">Port IP automate</p> <p style="text-align: center; background-color: #cccccc;">Temps de cycle</p> <p style="text-align: center; background-color: #cccccc;">Programme</p>



Configuration des Tables					
Table de transfert 1					
Nom	A_In	Adresse Locale	310	Taille	6
Type de Transfert	02: Read	Numéro esclave	0	Adresse Distant	0
		Index SendDataActive	-	Index ReceiveDataValid	DO27
		Index SendDataInProgress	-		
		Index SendDataResult	-		
Sauvegarder					
Table de transfert 2					
Nom	A_Out	Adresse Locale	410	Taille	6
Type de Transfert	01: Write	Numéro esclave	21	Adresse Distant	320
		Index SendDataActive	DO52	Index ReceiveDataValid	-
		Index SendDataInProgress	DO53		
		Index SendDataResult	AO22		
Sauvegarder					

Configuration:

6 tables peuvent être configurées.

Pour chaque table :

- Nom
- Type de Transfert: Inactive, Write ou Read
- Adresse locale: adresse MODBUS locale qui correspond à la première variable de la table
- Taille: longueur de la table, en registre MODBUS (mots de 16 bits). Maximum 6 mots. Les variables digitales sont associée à un bit, les variables analogiques utilisent un mot de 16 bits. La longueur doit être le même pour une table « Write » que pour la table « Read » correspondante.

Pour une table de type "Write":

- Numéro esclave: correspond au numéro d'esclave dans la page "configuration des esclaves" (où son adresse IP est configurée)
- Adresse distante: adresse MODBUS dans l'esclave à laquelle la table sera écrite
- Index SendDataActive: index d'une DO (variable de contrôle). Il s'agit de l'index de la DO qui sera activée par le programme automate pour déclencher l'écriture du contenu de la table vers l'équipement distant. Cette DO sera automatiquement remise à zéro par le firmware lorsque le transfert sera en cours.
- Index SendDataInProgress index d'une DO (variable de contrôle). Il s'agit de l'index de la DO qui indique que le processus de transfert est en cours. Cette variable est automatiquement remise à zero lorsque le transfert est terminé ou a échoué. Utile pour la mise en service.
- Index SendDataResult: index d'une AO (variable de contrôle) qui montre l'état du transfert en cours.

Les valeurs possibles sont:

0: opération réussie

1: fonction MODBUS invalide

2: adresse MODBUS invalide (adresse distante)

3: données invalides

15: erreur de communication – problème durant la création de la connexion TCP

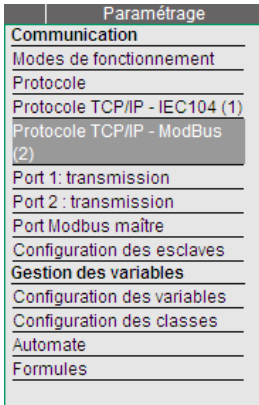
16: erreur de communication – timeout durant les échanges

20: opération en cours

Les codes 1, 2 et 3 indiquent que le message a été correctement transmis, mais que l'équipement distant l'a refusé. Cela est habituellement lié à un problème de configuration

Pour une table de type "Read":

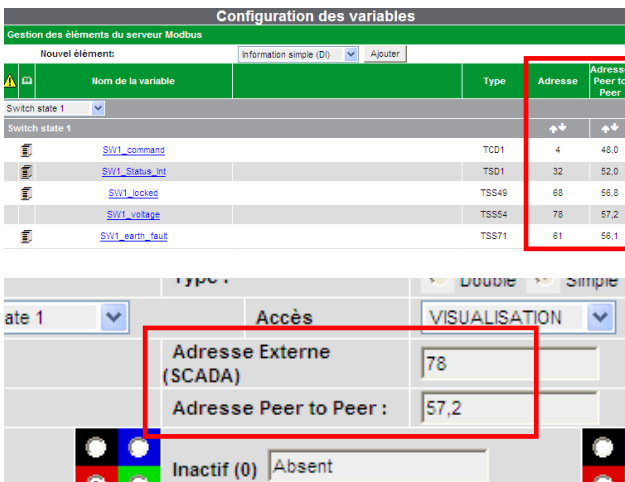
- Index ReceiveDataValid: index d'une DO (variable de contrôle). Cette variable sera activée lorsque de nouvelles données auront été écrites dans la table par un équipement distant. Elle devrait être remise à zéro par le programme automate.



Protocoles et Adresses des variables

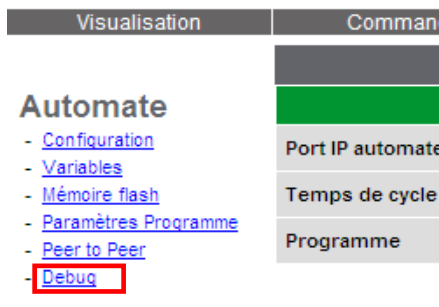
Comme la fonctionnalité Peer-to-peer utilise le protocole MODBUS TCP, le MODBUS maître et le MODBUS TCP esclave sont toujours actif sur l'équipement, en plus du protocole esclave principal (IEC101/104 ou DNP3/DNP3 IP).

Chacun de ces protocoles a sa propre page de configuration.



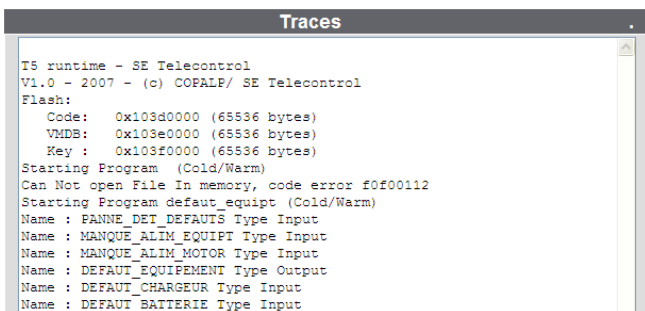
Pour chaque variable, l'adresse externe est utilisée par le protocole principal et l'adresse Peer to Peer est utilisée par le protocole MODBUS TCP pour la fonction Peer to Peer.

Si le protocole principal est MODBUS/MODBUS TCP, les mêmes adresses (adresses externes) sont utilisées pour la communication scada et la communication Peer-to-Peer



1.5.8 Traces automate

Le lien "Debug" de la page "Automate" permet de vérifier que le programme RTU Control Program fonctionne correctement et que les variables ont été chargées sans erreurs dans le T200/F200C.



En condition normale de fonctionnement, il n'est pas nécessaire de vérifier l'exécution d'un programme. C'est uniquement dans le cas d'un comportement anormal de l'automate que le contrôle de bon fonctionnement peut-être utile.

Exemple de trace automate

1.6 Exemples

Des projets de démonstration sont donnés à titre d'exemple. Ceux-ci sont intégrés à RTU Control et se trouvent dans le répertoire "RTU CONTROL*Samples\Easergy*".

Ces projets permettent par exemple de montrer le lien entre RTU control et les variables du T200, ainsi que la gestion des TSD/TCD. Ils incluent les configurations T200 ou F200C correspondantes dans un répertoire dédié.

Ces projets sont dédiés à la démonstration et ne sont pas supportés par le support technique.

Schneider Electric n'assume aucune responsabilité des conséquences éventuelles découlant de l'utilisation de ces programmes sur un site réel.

2 Formules de calcul

2.1 Présentation

Contrairement à l'automate programmable qui est fourni en option, les formules de calcul sont fournies de base avec le T200. Elles permettent des opérations arithmétiques et combinatoires entre les variables.

Elles offrent ainsi la possibilité :

- d'effectuer des calculs mathématiques à partir de mesures physiques (consommations, conversions, etc...)
- de créer des signalisations personnalisées (combinaisons de variables, défauts regroupés, etc...)
- de définir des automatismes de contrôle ou de gestion (permutations, etc...).

Une formule est associée à une variable. Sa valeur est mise à jour à chaque exécution de la formule. Cette exécution peut se faire suivant un temps de cycle défini. Le temps de cycle minimum est 100 ms.

2.2 Syntaxes et règles

Les formules utilisent une syntaxe naturelle et intuitive qui mélange références aux valeurs de mesures (variables, historiques, ...), expressions littérales (constantes, opérateurs...) et appels de fonctions (min, max, delta, ...).

L'utilisateur n'a pas à se soucier des formats de données, l'interpréteur prend en charge les conversions, il est ainsi possible de combiner les opérations logiques et arithmétiques: $(B > 3) * C$, $(B > 3)$ est une expression logique (vrai (1) ou faux (0)).

Les formules n'ont de sens que si à l'issue de leur exécution la valeur obtenue est affectée à une variable (ou à une cellule de la base de données). Il faut bien comprendre qu'**une formule n'est pas un langage de script**, il n'existe pas de notion de boucles (for, while) ni de sauts d'instructions (goto, if..then). Néanmoins il existe un opérateur conditionnel qui permet d'affecter telle ou telle valeur d'après une expression logique.

Les formules tiennent compte de la casse (minuscule ou majuscule) des opérandes et des fonctions. La longueur maximale d'une formule est de 200 caractères.

Une formule peut contenir jusqu'à 50 éléments (opérandes, opérateurs...).

Une formule peut être scindée en plusieurs "sous-formules" séparées par un **point virgule (la virgule est réservée pour séparer les paramètres d'une fonction)**.

Chaque "sous-formule" est exécutée successivement dans leur ordre d'écriture au cours du même cycle de traitement.

2.2.1 Les opérandes

2.2.1.1 Constantes numériques (nombres entiers, nombres à virgule):

Les nombres peuvent être positifs et négatifs, entiers ou à virgule et/ou exposant : 123, -45.1, 12.5E3, etc..

Pour saisir un nombre en hexadécimal, sa valeur doit être préfixée par le symbole \$ (ou la lettre H), par exemple \$10 (ou H10).

Pour les valeurs "à virgule", le séparateur décimal est le point.

Le type de la valeur (entier, mot, IEEE) est automatiquement déterminé par l'analyse de la formule.

2.2.1.2 Chaînes de caractères:

Une chaîne de caractères doit être saisie entre simples cotes ', par exemple '*ma chaîne*'. Les chaînes de caractères sont utilisées pour référencer certains paramètres de fonction ou bien des noms de tables ou de colonnes de la base de données.

2.2.1.3 Variables:

En saisissant le nom d'une **variable** (sans cotes), l'opérande sera interprété comme la **valeur de la variable** au moment de l'exécution. Le type de la valeur est celui de la variable.

Attention, le nom des variables utilisées dans les formules **ne doit pas contenir d'espace**.

2.2.2 Les opérateurs

2.2.2.1 Arithmétiques:

+	Addition
-	Soustraction
*	Multiplication
/	Division (la division par zéro provoque une erreur à l'exécution)
%	Modulo : Reste de la division entière (nombres entiers)
**	Puissance : $A^{**}B = A^{\text{exposant}} B = A^B$

2.2.2.2 Logiques:

&&	ET logique	$A \ \&\& \ B = \text{vrai}$ si $A = \text{vrai}$ ET $B = \text{vrai}$, faux sinon
##	OU logique	$A \ \#\# \ B = \text{vrai}$ si $A = \text{vrai}$ OU $B = \text{vrai}$, faux sinon
^^	OU EXCLUSIF logique	$A \ \wedge\wedge \ B = \text{vrai}$ si ($A = \text{vrai}$ ET $B = \text{faux}$) OU ($A = \text{faux}$ ET $B = \text{vrai}$)
!	NON logique	$!A = \text{vrai}$ si $A = \text{faux}$

Remarque : le symbole « | » est obtenu en pressant simultanément les touches [AltGr] et [6] du clavier.

2.2.2.3 Binaires:

&	ET bit à bit	$\$AA \ \& \ \$55 = \$00$
	OU bit à bit	$\$AA \ \ \$55 = \$FF$
^	OU EXCLUSIF bit à bit	$\$AA \ ^ \ \$A5 = \$0F$
~	INVERSE bit à bit (Complément à 1)	$\sim A$, inverse tous les bits de A , ceux à 1 passe à 0 et inversement
<<	Décalage à gauche	$A \ \ll \ B$, décale les bits de A de B bits vers la gauche
>>	Décalage à droite	$A \ \gg \ B$, décale les bits de A de B bits vers la droite

2.2.2.4 Comparaisons:

=	Egal
==	
!=	Différent
<>	
>	Supérieur
<	Inférieur
>=	Supérieur ou égal
<=	Inférieur ou égal

2.2.2.5 L'opérateur d'affectation

L'opérateur := permet d'affecter une valeur à une variable:

Exemple : *mavar := 2.*

L'utilisateur n'a pas à se soucier du format de donnée de destination, les conversions sont automatiques.

Dans le cas d'une variable sur périphérique en entrée/sortie, une affectation provoquera l'envoi d'un ordre d'écriture vers le périphérique au moment de l'exécution de la formule.

2.2.2.6 L'opérateur conditionnel

L'opérateur $(expr) ? A : B$ renvoie A si $expr$ est vraie, B sinon.

$expr$ doit être une expression booléenne.

Par exemple :

$A := (B > C) ? D : E;$ affectera D à A si B est supérieur à C , sinon affectera E à A .

Il est bien sûr possible d'utiliser des parenthèses pour construire des expressions plus complexes.

De même dans une opération :

$A + ((B > C) ? D : E)$ vaut $A + D$ si $B > C$, elle vaut $A + E$ sinon.

L'opérateur conditionnel peut aussi être utilisé pour une affectation:

$((B > C) ? D : E) := A;$

dans ce cas, si $B > C$, D est affecté de la valeur A , sinon c'est E qui est affecté de la valeur A .

2.2.2.7 Ordre de priorité des opérateurs et utilisation des parenthèses

L'ordre décroissant de priorité dans l'exécution des opérations d'une expression est le suivant :

Fonction	Fcn ()	P0 = priorité max
Signe positif	+ (Var)	P1
Signe négatif	- (Var)	
Non logique	!	
Complément à 1 *	~	
Puissance	**	P2
Multiplication	*	P3
Division	/	
Reste *	%	
Addition	+	P4
Soustraction	-	
Décalage à gauche *	<<	
Décalage à droite *	>>	
Et bit à bit *	&	P5
Ou bit à bit *	/#	
Ou exclusif bit à bit	^	
Et logique	&&	P6
Ou logique	/##	
Ou exclusif logique	^^	
Test égal à	=	
Test inférieur ou égal	<=	
Test sup ou égal à	>=	
Test inférieur à	<	
Test supérieur à	>	
Test différent de	!=/<>	
Affectation	:=	

"*" : Opération sur des entiers uniquement (manipulation de bits)

2.3 Les fonctions

Les arguments des fonctions peuvent être indifféremment toutes sortes d'opérandes, mais aussi des expressions arithmétiques ou logiques ou encore des résultats de fonctions. Ainsi, il est possible d'imbriquer plusieurs appels de fonctions et d'opérations.

" , ...)" : les trois petits points désignent des fonctions à nombre variable d'arguments.

"[x]" : les crochets désignent un (ou plusieurs) argument(s) optionnel(s).

2.3.1 Les fonctions mathématiques

Sauf mention contraire, le format de donnée des valeurs renvoyées par ces fonctions est IEEE (flottant simple précision).

2.3.1.1 abs(x)

Retourne la valeur absolue de son argument x

Le type de donnée retourné par la fonction est le même que celui de x.

Exemple : `Abs (VAR1 * 3)`

2.3.1.2 sqrt(x)

Retourne la racine carrée de x.

Si x est inférieur à zéro, l'opération provoque une erreur à l'exécution.

2.3.1.3 log(x)

Retourne le logarithme népérien de x.

Si x est inférieur ou égal à 0, l'opération provoque une erreur à l'exécution.

2.3.1.4 log10(x)

Retourne le logarithme en base 10 de x.

Si x est inférieur ou égal à 0, l'opération provoque une erreur à l'exécution.

2.3.1.5 exp(x)

Retourne l'exponentiel de x. $\exp(\log(x)) = x$.

2.3.1.6 pow(x, y)

Retourne x à la puissance y. $\text{pow}(x, y) = x^{**} y = x^y$.

2.3.1.7 intg(x)

Retourne la partie entière de x. Par exemple, $\text{intg}(12.46) = 12$. La valeur retournée est de type entier (INT32).

2.3.1.8 rand(x)

Retourne un nombre entier aléatoire supérieur ou égal à 0 et inférieur à x.

La valeur retournée est de type double mot (DWORD).

2.3.2 Les fonctions statistiques:

2.3.2.1 mini(A, B, ...)

Retourne l'argument dont la valeur est la plus petite. Cette fonction doit comporter **au moins deux** arguments.

Notez que cette fonction ne retourne pas seulement la valeur de l'argument mais aussi sa référence. Ainsi il est possible de combiner avec une opération d'affectation:

mini(VAR1, VAR2, VAR3) := 4; avec VAR1, VAR2, VAR3 des références sur des variables, si VAR3 a la valeur minimum, cette variable sera ensuite affectée de la valeur 4.

2.3.2.2 maxi(A, B, ...)

Retourne l'argument dont la valeur est la plus grande. Cette fonction doit comporter **au moins deux** arguments.

Comme pour **mini**, cette fonction retourne la référence sur l'argument.

2.3.2.3 pmini(A, B, ...)

Retourne la position (à partir de 0) de l'argument dont la valeur est la plus petite. Cette fonction doit comporter **au moins deux** arguments.

2.3.2.4 pmaxi(A, B, ...)

Retourne la position (à partir de 0) de l'argument dont la valeur est la plus grande. Cette fonction doit comporter **au moins deux** arguments.

2.3.2.5 sum(A, B, ...)

Retourne la valeur de la somme des arguments. Cette fonction doit comporter **au moins deux** arguments.

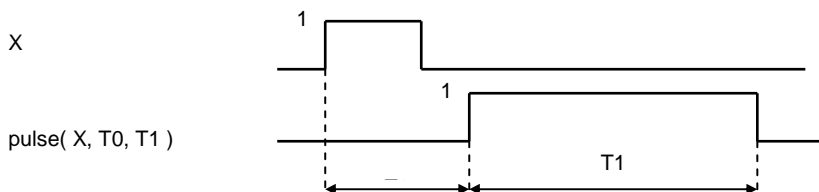
2.3.2.6 avg(A, B, ...)

Retourne la valeur de la moyenne des arguments. Cette fonction doit comporter **au moins deux** arguments.

2.3.3 Les fonctions Logiques:

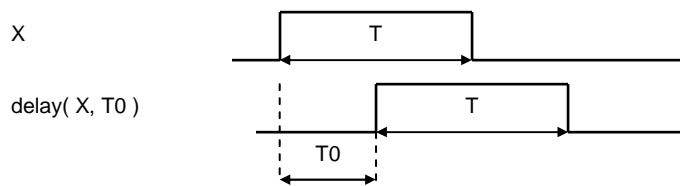
2.3.3.1 pulse(X, T0, T1)

X est une valeur booléenne. T0, T1 des nombres de secondes. Sur front montant de X (passage de 0 à 1), après T0 secondes, la fonction renvoie vrai (1) pendant T1 secondes. Le reste du temps elle renvoie faux (0).



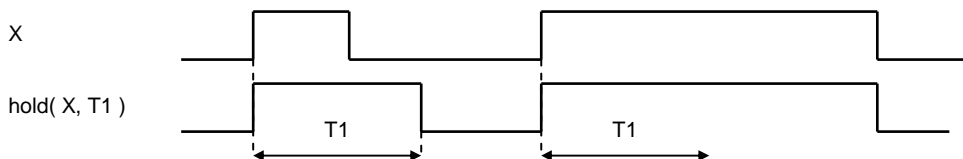
2.3.3.2 delay(X, T0)

X est une valeur booléenne. T0 un nombre de secondes. La fonction recopie la valeur de X décalée dans le temps de T0 secondes.



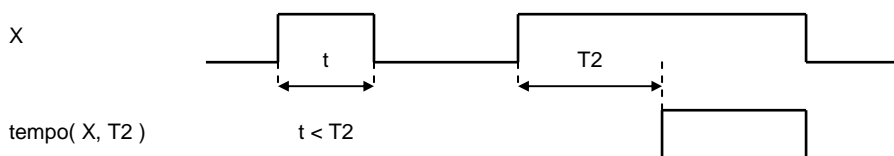
2.3.3.3 hold(X, T1)

X est une valeur booléenne. T1 un nombre de secondes. La fonction renvoie vrai pendant au moins T1 secondes sur front montant (passage de 0 à 1) de X. Si X se maintient à 1 (vrai) au-delà de T1 secondes, la fonction hold renvoie vrai pendant ce temps.



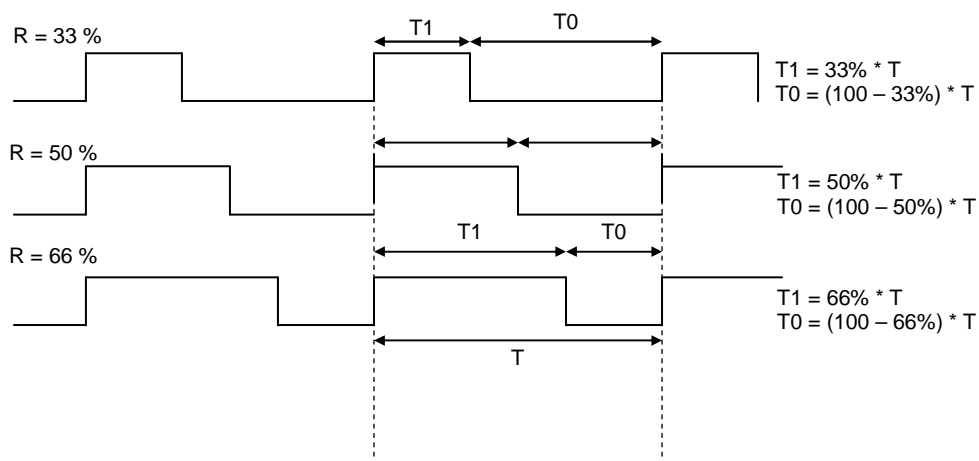
2.3.3.4 tempo(X, T2)

X est une valeur booléenne. T2 un nombre de secondes. La fonction renvoie vrai si X passe à 1 pendant au moins T2 secondes. Passé ce temps, la fonction renvoie vrai tant que X est à vrai.



2.3.3.5 pwm(T, R)

La fonction renvoie une valeur booléenne au rythme d'un signal de chrono proportionnel (PWM, Pulse Width Modulation) de période T et de rapport cyclique R en pourcentage. R doit être compris entre 0 et 100%.



2.3.3.6 iswithin(X, min, max)

retourne une valeur booléenne qui vaut *vrai* si ($X \geq min$) **ET** ($x \leq max$), *faux* sinon.

2.3.4 Les fonctions Horaires / Dates et heures:

Le format TIME_T est un format de données qui permet de mémoriser un couple date/heure en nombre de secondes depuis une référence unique (le 1^{er} janvier 1970 à 00:00:00).

Les jours de la semaine sont associés aux valeurs suivantes :

1 = dimanche, 2 = lundi, 4 = mardi, 8 = mercredi, 16 = jeudi, 32 = vendredi, 64 = samedi.

2.3.4.1 dt([[T], X]) ou now([[T], X])

Sans argument, la fonction renvoie la date et l'heure courante en TIME_T.

Avec T en TIME_T et X, T sert de référence pour l'argument X

Avec X seul, la date et l'heure courante servent de référence pour l'argument X

si X = 1 ou 'Y', retourne l'année sur 4 chiffres

si X = 2 ou 'M', retourne le mois (de 1 à 12)

si X = 3 ou 'D', retourne le jour du mois (de 1 à 31)

si X = 4 ou 'H', retourne l'heure sur 24H (de 0 à 23)

si X = 5 ou 'm', retourne la minute (de 0 à 59)

si X = 6 ou 's', retourne la seconde (de 0 à 59)

si X = 7 ou 'd', retourne le jour dans la semaine (1, 2, 4, 8, 16, 32, 64)

si X = 8 ou 'p', retourne 0 si l'heure est avant midi (AM), 1 pour l'après midi (PM)

si X = 9 ou 'h', retourne l'heure sur 12H

si X = 10 ou 'y', retourne l'année sur 2 chiffres.

2.3.4.2 time(X [, F])

Avec X seul, si X est une chaîne de caractères, la fonction convertit en TIME_T la date et l'heure correspondant à X d'après le format de date et d'heure par défaut. Si X est un numérique la fonction convertit ce nombre en TIME_T.

Avec X et F, X doit être une chaîne de caractères, F est aussi une chaîne de caractères qui définit le format de lecture de la date et de l'heure. La fonction renvoie alors la date X en TIME_T d'après le format F.

F est de la forme '**d/m/y H:M:S**' avec:

d = le jour dans le mois (de 1 à 31)

m = le mois dans l'année (de 1 à 12)

y = l'année sur 2 chiffres

Y = l'année sur 4 chiffres

H = l'heure sur 24H

h = l'heure sur 12H

M = la minute (de 0 à 59)

S = la seconde (de 0 à 59).

2.3.4.3 isweek([X])

Si **X (optionnel) est absent**, la fonction renvoie la valeur booléenne *vrai* si la date courante est un jour ouvré (du lundi ou vendredi inclus).

Si **X est de type TIME_T**, la fonction renvoie *vrai* si la date X est un jour ouvré. **Si non** X désigne un numéro de jour dans la semaine, la fonction renvoie *vrai* si le jour X est ouvré.

2.3.4.4 deltats()

Retourne le temps en SECONDES entre deux exécutions de la formule.

Cette fonction est très puissante car elle permet de faire des intégrations, des dérivées, etc..

Elle est basée sur l'horloge temps réel du système.

2.3.4.5 deltatms()

Retourne le temps en MILLISECONDES entre deux exécutions de la formule.

Cette fonction est très puissante car elle permet de faire des intégrations, des dérivées, etc..

Elle est basée sur le quartz de la CPU. De ce fait, elle est relativement peu précise sur du long terme.

2.3.5 Les autres fonctions:

2.3.5.1 bounds(X, min, max)

Retourne la valeur de X bornée par *min* et *max*.

si X est inférieur à *min*, retourne *min*; si X est supérieur à *max*, retourne *max*; sinon retourne X.

2.3.5.2 delta(X)ou deltav(X)

Retourne la différence entre la valeur de X lors de la précédente exécution de la formule et la valeur courante de X. La période pour le calcul de la différence de X est donc la période d'exécution de la formule. Voir aussi deltats() et deltatms().

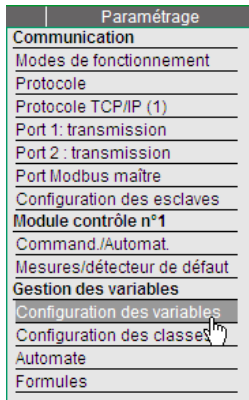
2.3.5.3 switch(X, A0, A1 [, A2, ...]) ou cond(X, A0, A1 [, A2, ...])

X doit être une valeur entière positive.

Cette fonction doit comporter **au moins trois** arguments.

Si X = 0, la fonction retourne l'argument A0, si X = 1, retourne A1, si X = 2, retourne A2, etc.

Si X < 0, la fonction retourne l'argument A0, si X est supérieur au nombre d'arguments, le dernier est retourné.



2.4 Configuration

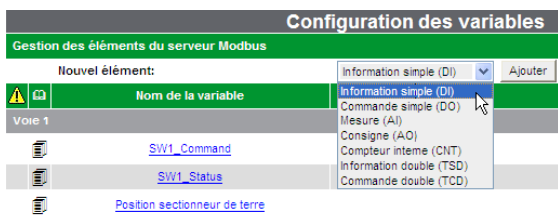
2.4.1 Création des variables

L'utilisateur doit déclarer des variables qui seront spécifiquement associées à la fonction "Formules de calcul". Elles ne seront mises à jour que par la fonction "Formules de calcul", mais seront gérées comme les autres variables au niveau du protocole et de la visualisation des états.

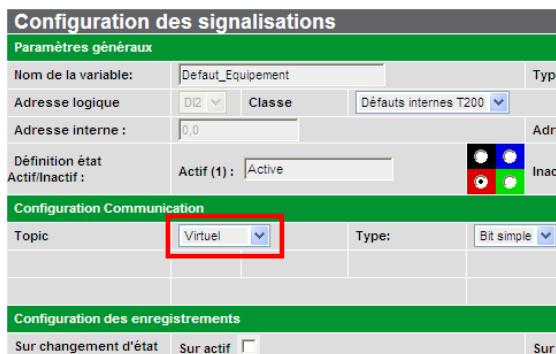
La création de ces variables s'effectue depuis la page de configuration des variables du serveur Web, de la même façon que les autres types de variables.

Les types de variables pouvant être créées spécifiquement pour les fonctions sont :

Type variable	Utilisation	Description
DI	Télésignalisation simple	Signalisation par bit simple
DO	Télécommande simple	Commande par bit simple
AI	Télémesures	Mesures analogiques effectuées par le T200/F200C (courant, tension...)
AO	Commande analogique	Ecriture analogique
CNT	Compteur	Compteur de changement d'état d'un bit
TCD	Télésignalisation double	Commande double sur interrupteur/disjoncteur
TSD	Télécommande double	Signalisation double d'interrupteur/disjoncteur

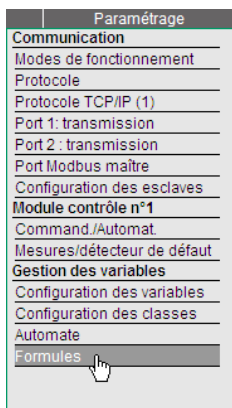


Page "Configuration des variables"



Le nom d'une variable associée à une formule de calcul ne doit pas contenir d'espace.

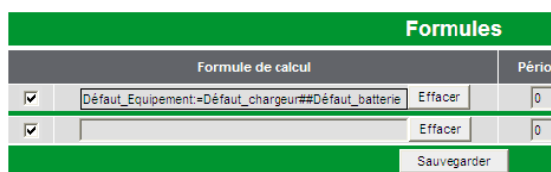
Une variable associée à une formule doit être configurée avec un type de type "Virtuel".



2.4.2 Saisie des formules

Une fois la variable créée, la formule associée doit être saisie dans la page "Formules".

Cette page peut inclure au maximum 100 formules. 20 formules à la fois peuvent être affichées.



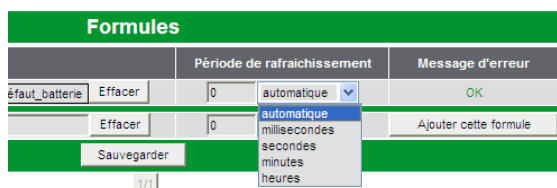
Exemple de formule saisie

Règles à respecter pour la saisie de la formule :

- Chaque formule doit commencer par "nom_de_variable :=".
- Aucun espace n'est accepté dans les noms des variables
- Le contenu de la formule avec les opérandes et les opérateurs est définie derrière le signe "...:="
- La casse doit être respectée pour les noms des variables et opérandes

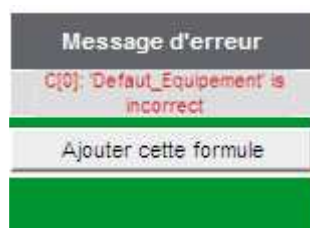
Chaque formule peut être activée/désactivée par une case à cocher.

Une formule est associée à une période de rafraichissement configurable. Une période définie à "Auto." correspond à une période la plus courte possible (100 ms)



Période de rafraichissement d'une formule

Un message d'erreur s'affiche si la syntaxe de la formule n'est pas correcte.



Exemple de message d'erreur sur saisie de formule incorrect

Le résultat de la formule peut être consulté dans la page "Visualisation".

Il correspond à l'état de la variable associée à la formule correspondant au résultat de celle-ci.

Défauts internes T200	
Manque_alim_motor.	Non
Manque_alim_équipt	Non
Défaut_chargeur	Non
Défaut_batterie	Oui
Panne_dét_Défauts	Non
Défaut_Equipement	Active

Résultat de la formule dans la page "Visualisation".

Schneider Electric Industries SAS

Schneider Electric Telecontrol
839 chemin des Batterses
Z.I. Ouest
01700 St Maurice de Beynost
Tel : +33 (0)4 78 55 13 13
Fax : +33 (0)4 78 55 50 00

<http://www.schneider-electric.com>
E-mail : telecontrol@schneider-electric.com

En raison de l'évolution des normes et du matériel, les caractéristiques indiquées par les textes et les images de ce document ne nous engageant qu'après confirmation par nos services.

As standards, specifications and designs change from time to time, please ask for confirmation of the information given in this publication.

Debido a la evolución de las normas y del material, las características y dimensiones indicadas en el texto y las imágenes nos comprometen solamente previa confirmación de nuestros servicios.