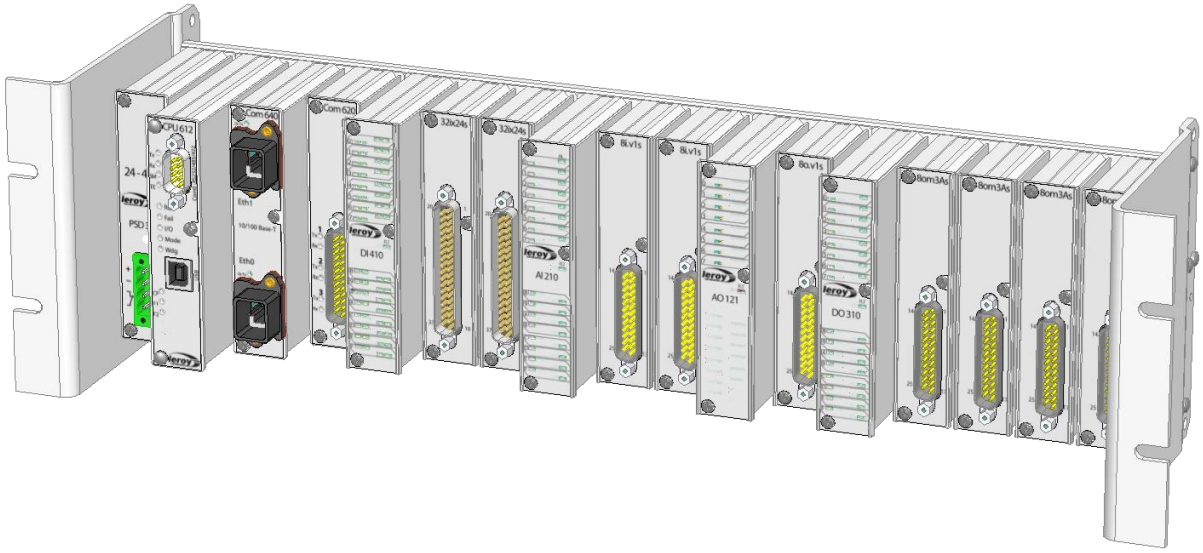
Gamme LT

Plate-forme d'automatisme modulaire



MANUEL DE PROGRAMMATION STRATON
Pour unités centrales CPU6xx

P DOC LT 006 F - V05

Page laissée blanche intentionnellement

Présentation

Les unités centrales LUC4xxx (Module CPU6xx) de la gamme LT sont programmables :

- Soit en langage C sous Linux, dans l'IDE Eclipse.
- soit en langages IEC 61131-3 avec l'atelier logiciel Straton de la société COPADATA.

Le manuel de mise en œuvre matérielle est disponible sur notre site Internet.

Dans la suite du document les mentions « LT » et « LT Straton » désignent une CPU6xx équipée d'un noyau Straton.

Prérequis

Le développement d'applications Straton nécessite des connaissances sur les langages de programmation définis dans la norme IEC61131-3.

La mise en œuvre matérielle nécessite des compétences en électricité et automatismes industriels.

Le matériel nécessaire est :

- Un PC de développement sous Windows, muni d'un port USB et d'une carte réseau Ethernet
- Un câble USB entre le PC de développement et l'automate
- Une liaison Ethernet entre le PC de développement et l'automate

Version

Cette documentation présente les fonctionnalités présentes dans la version 2.3 du kernel LT Straton.

Propriété

Contact

✉ Leroy Automation
250 rue Max Planck
31670 LABEGE
France

☎ +33 562 240 550

http://www.leroy-automation.com

Support technique:

☎ +33 562 240 546

mailto:support@leroy-autom.com

Page laissée blanche intentionnellement

Table des matières

Page laissée blanche intentionnellement

Chapitre

1

Présentation générale

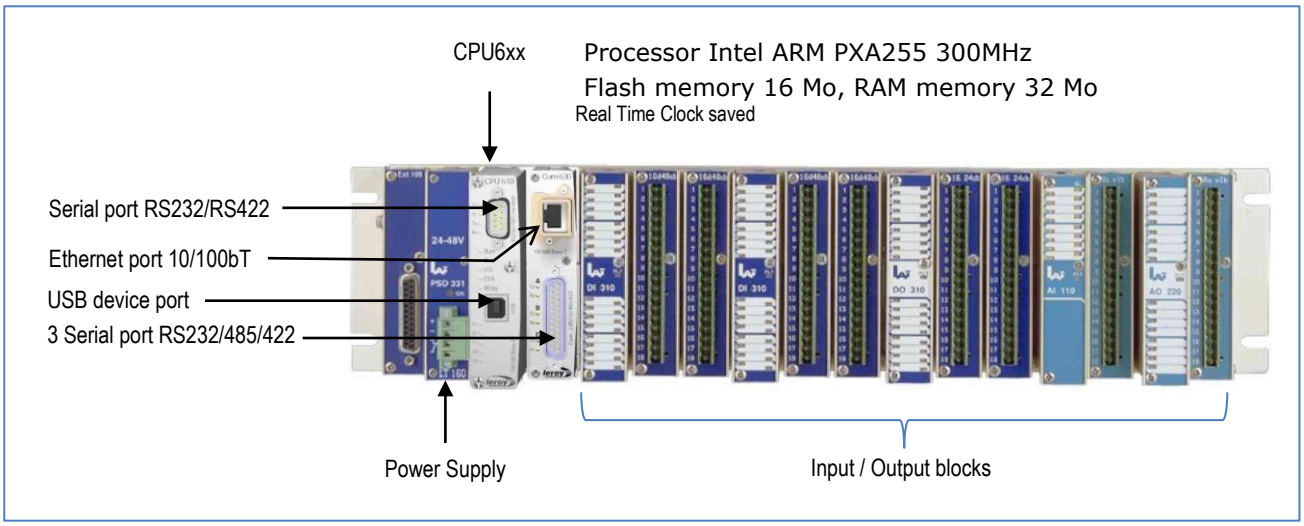
Présentation

Ce chapitre décrit :

- Les ressources matérielles de l'automate
- Les ressources logicielles de l'automate

Ressources matérielles

Pour connaître la liste des CPU disponibles, consulter le catalogue dans le manuel de mise matérielle. Les unités centrales programmbles avec Straton sont basées sur les modules CPU6xx.



Ressources logicielles

Les CPU6xx contiennent un noyau Linux 2.6.12 et un noyau Straton. A la mise sous tension, si un projet Straton est valide il sera démarré.

La version de l'atelier de programmation Straton à utiliser est V8.1 ou supérieure.

Page laissée blanche intentionnellement

Chapitre

2

Démarrage rapide

Présentation

Ce chapitre décrit l'ensemble des opérations nécessaires pour mettre en œuvre et tester un programme élémentaire en moins de 20 minutes.

Nous détaillons dans ce chapitre les étapes suivantes :

- Installation du driver USB
- Installation de l'atelier Straton
- Création d'un nouveau projet
- Déclaration d'une configuration matérielle
- Compilation, chargement et mise au point

Installation du driver USB

Ce driver permet à l'atelier de communiquer avec l'automate par le port USB. Ce driver est compatible avec les versions de Microsoft Windows XP,7,8,10 (32 et 64 bits).

Le fichier à exécuter est un fichier setup, livré sur le CD ROM dans le répertoire « DriverUSB-PC-LT200 » : fichier « LEROY USB DRIVER Setup.exe »

Procédure d'installation :

- IMPORTANT : le LT ne doit pas être connecté en USB au PC, pendant l'installation du driver
- Exécuter le fichier « LEROY USB DRIVER Setup.exe »
- Connecter le port USB de l'automate au PC : un « bip » doit prévenir de la reconnaissance par le PC de celui-ci.
- Vérifier le port com USB pour le LT : ouvrir le « panneau de configuration », puis aller dans « Système », onglet « Matériel », « gestionnaire de périphériques », « Ports (Com et LPT) » : noter le numéro x du com : « Leroy USB Device (COMx)»

Nota : le numéro du port com USB pour le LT est affecté automatiquement par Windows ; si nécessaire, le numéro du port est modifiable, via les paramètres du port.

Installation de Straton

Installer l'atelier Straton

Depuis le CD Rom Straton de Copalp, lancer l'installation de l'atelier sur votre PC sous Microsoft Windows 7 ou supérieur.

Activer la licence logicielle de l'atelier Straton via l'outil Straton « Licence Manager» ; il peut être démarré depuis le menu démarrer de Windows, lien *Straton/Licence*.

Ajouter la clé d'activation de votre licence qui a été livrée avec le CD Straton.

Paramétrer le niveau de privilège Windows du logiciel « Straton editor » (propriétés Windows) pour qu'il soit exécuté en tant qu'administrateur.

Intégrer le fichier de librairie LT de Leroy Automation à l'atelier Straton

Démarrer le logiciel Straton Editor depuis le menu démarrer de Windows.
Ouvrir le menu *Outils/Importer/Importer des librairies OEM*, et importer le fichier suivant présent sur le CD-ROM Leroy Automation "LT200_Straton_LNX_xx":

- " LT200_straton_libraries_Vxx.XL5"

Création d'une nouvelle liste de projets

Ouvrir le menu *Fichier/Nouvelle Liste…* : renseigner le nom de votre nouvelle liste de projets, et valider.

Création d'un nouveau projet

Ouvrir le menu *Fichier/Ajouter un nouveau projet…* : renseigner le nom de votre nouveau projet, et valider.

Configuration matérielle

Dans la barre des boutons, cliquer sur l'icône « Ouvrir E/S »: 
La fenêtre « Cartes d'E/S » apparaît.
Double cliquer sur l'emplacement « 0 » et sélectionner dans la liste la carte « CPU6xx ».
Fermer l'éditeur de gestion des cartes d'E/S.

Compilation

Dans la barre des boutons, cliquer sur l'icône « Compiler le projet de démarrage »: 
La compilation du projet est effectuée, et le message suivant doit apparaître dans la fenêtre « Générer »: « Aucune erreur détectée ».

Paramètres de communication console PC ↔ LT

Ouvrir le menu *Outils/Paramètres de Communication*, et entrer dans la zone « Paramètres de communication » les paramètres nécessaires. La communication peut être établie en USB ou Ethernet :

- communication USB : sélectionner le port USB ouvert sur votre PC : « com3 » par exemple
- communication Ethernet : saisir l'adresse IP du LT, suivie du port TCP 502:
  - « xxx. xxx. xxx.xxx:502 » avec xxx compris entre 0 et 255
  - Exemple : « 192.168.1.190:502 »

Valider les nouveaux paramètres : cliquer sur le bouton « OK ».

Chargement et mise au point

Cliquer sur le bouton « Exécution » dans la barre de menu principale : 
Différents cas peuvent se produire :

- le programme dans le LT est différent du programme de l'atelier : il apparaît une fenêtre "Mauvaise version!", cliquer sur le bouton « Arrêter et charger la nouvelle version ». 
- un autre programme est en fonctionnement dans le LT : arrêter le programme puis charger le  nouveau.

Après chargement, le message suivant doit apparaître dans la barre des menus: "Marche".
Pour revenir dans le mode édition du projet, cliquer une nouvelle fois sur le bouton "Exécution".
Vous venez de créer et charger votre premier programme Straton pour LT.

Chapitre

3

Configuration matérielle

Présentation

Ce chapitre décrit le paramétrage des cartes de l'automate.

Nous détaillons dans ce chapitre la configuration des cartes CPU et IO :

- Configuration matérielle
- Carte CPU 610 : Paramètres WdG et réseau Ethernet
- Cartes d'entrées / sorties TOR : DI310, DI410, DI312, DIO210, DO310, DI130, DIO130
- Cartes d'entrées / sorties analogiques : AI110, AI210, AO121, AIO320
- Status des cartes d'E/S

Configuration matérielle

Dans la barre des boutons, cliquer sur l'icône « Ouvrir E/S »: 

La fenêtre « Cartes d'E/S » apparaît.

16 cartes au maximum peuvent être définies.

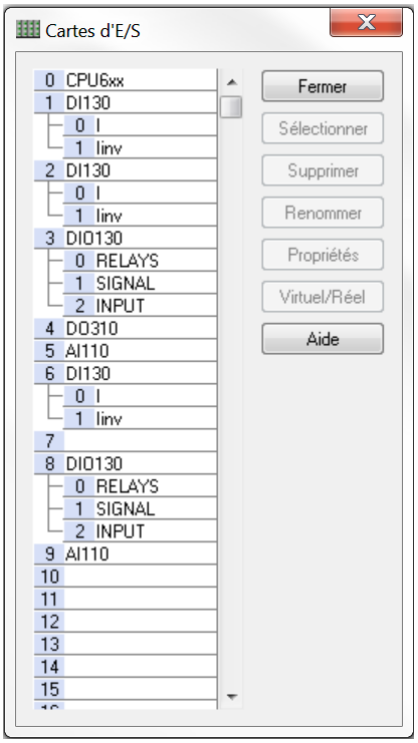
Chaque carte est repérée par un « Index d'équipement ».

Index d'équipement d'E/S n° 0 : réservé à la CPU.

Index d'équipement d'E/S n° 1 à 15 : réservés aux cartes d'entrée / sortie. L'index d'équipement des cartes correspond à la position physique de celles-ci sur le rack principal et les racks d'extensions (2 au maximum).

Exemple : configuration matérielle sur 2 racks, et configuration logicielle correspondante :

EXT101	PSD331	CPU610	COM630	DI130			DI130			DIO130			DO310		
		0		1			2			3			4		
EXT100	AI110			DI130			Libre			DIO130			AI110		
	5			6			7			8			9		



Carte CPU6xx

Paramètres carte

- **Key** (lecture seule) : code de la carte board. Valeur = 1
- **WDGLevel** : gestion du watchdog si le temps de cycle Straton dépasse la valeur WDGTimer
  - 0 (default): pas de watchdog.
  - 1 : LED Fail allumée et cartes IO en défaut
  - 2 : LED Fail allumée et programme Straton en mode « pas à pas »
  - 3 : reboot du LT

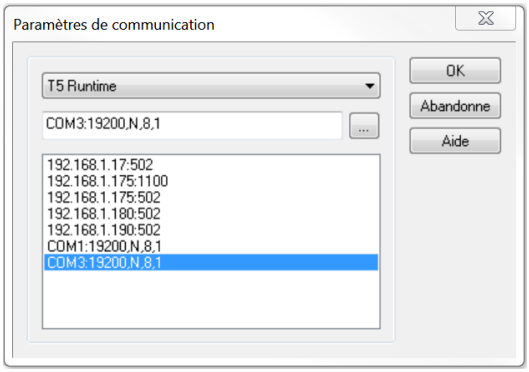
Une fois le Wdg déclenché, il est nécessaire de redémarrer le LT (coupure et remise sous tension).

- **WDGTimer** : valeur du watchdog en ms ; valeur minimale : 100ms.
  - 0 (default): pas de watchdog.
  - >0 : temps maximum autorisé pour le cycle

Paramètres Ethernet

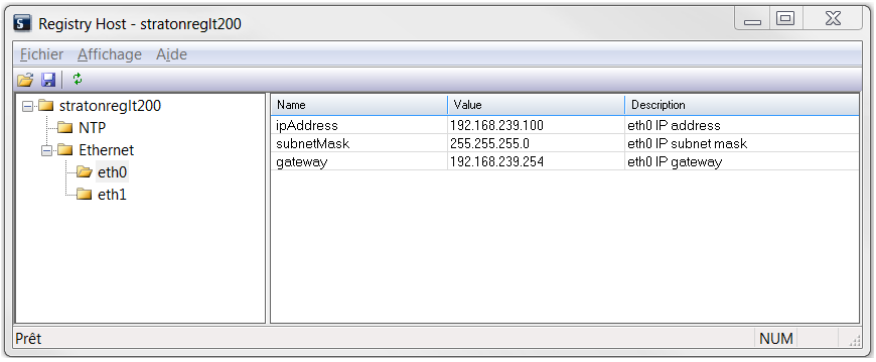
Les paramètres Ethernet LT doivent être réglés avec le logiciel « **T5 Registry Host** »:

- Ouvrir le menu *Outils/Paramètres Cible/Editer*
- Sélectionner le port de communication pour établir la communication avec le LT:
- Exemple : « COM3 :19200,N,8,1 »



- Valider par appui sur le bouton « OK » : le logiciel Registry Host va lire dans le LT les réglages des paramètres présents.

Des groupes de paramètres vont apparaître :

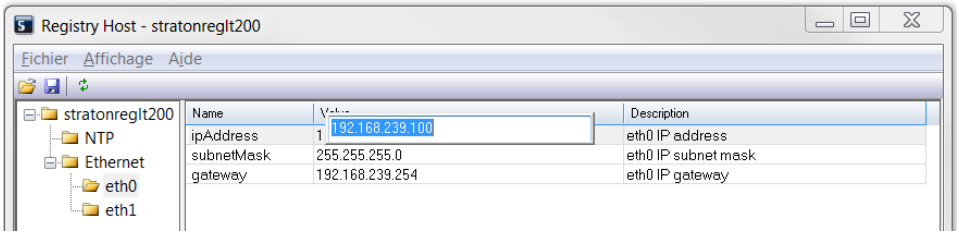


Le groupe **NTP** permet de mettre à l'heure le LT par le protocole Network Time Protocol.

Paramètres NTP : Par défaut les champs sont vides, et la fonction NTP n'est alors pas active.

- ipServer : adresse IP du serveur NTP sur un réseau TCP/IP qui fournira la date et l'heure UTC.
- pollingFrequency : fréquence d'interrogation du serveur NTP.
- timeZone : fuseau horaire d'installation du LT.

Groupe **Ethernet**



Paramètres **eth0** : paramètres Ethernet pour le premier port Ethernet

- **ipAddress** : adresse IP sur un réseau TCP/IP.

Par défaut ce champ est vide. Dans ce cas le LT va ignorer les autres paramètres et utiliser un serveur d'adresse BOOTP ou DHCP pour obtenir automatiquement une adresse IP disponible sur le réseau.

Format : xxx.xxx.xxx.xxx avec xxx [0..255]

- **subNetMask** : masque d'adresse permettant de mettre en évidence la division de l'adresse IP en adresse de réseau et sous-réseau et adresse de l'équipement sur ce sous-réseau. Ce masque de 32 bits comporte des 1 pour les parties de l'adresse de réseau et sous-réseau et des 0 pour les parties de l'adresse de l'équipement.

Format : xxx.xxx.xxx.xxx avec xxx [0..255]

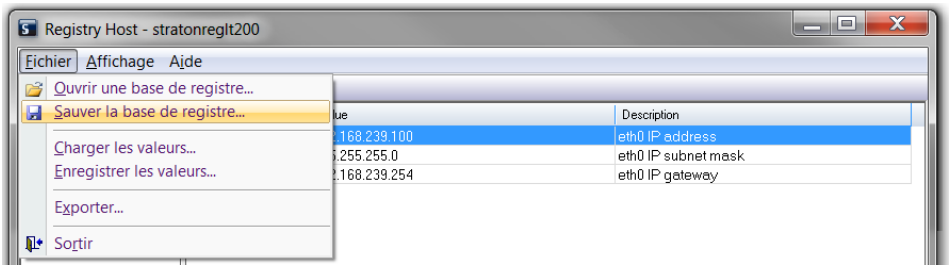
- **gateway** : adresse IP de la passerelle se trouvant sur le réseau. Si le LT veut communiquer en dehors du réseau auquel il appartient, il doit s'adresser à cette passerelle. Par défaut, ce champ est vide.et identifie le LT lui-même (pas de passerelle).

Format : xxx.xxx.xxx.xxx avec xxx [0..255]

Paramètres **eth1** : paramètres Ethernet pour le second port Ethernet .

Les valeurs des paramètres peuvent être modifiées en double cliquant sur le champ « Value » du paramètre désiré : exemple de modification de l'adresse IP.

Après modification des valeurs des paramètres, sauvegarder les nouvelles valeurs via le menu *Fichier/Sauver la base de registre*, ou le bouton « Sauver la base de registre… » de la barre des boutons :



Les nouveaux paramètres seront pris en compte au redémarrage du LT.

LEDs F1 & F2

Deux fonctions permettent de commander chacune des LEDs F1 et F2 de la CPU.

Fonction	**LEDF1**
Action	Commande de la LED appelée F1
Paramètres	(Bool) : FALSE=OFF ; TRUE=ON
Valeur retournée	(SINT) : fonction status [0..FFh] Status = 0 : succès. Status <> 0 : erreur.
Exemple	*Status := LEDF1(TRUE);* (*LED F1 allumée*)

Fonction	**LEDF2**
Action	Commande de la LED appelée F2
Paramètres	(Bool) : FALSE=OFF ; TRUE=ON
Valeur retournée	(SINT) : fonction status [0..FFh] Status = 0 : succès. Status <> 0 : erreur.
Exemple	*Status := LEDF2(TRUE);* (*LED F2 allumée *)

Carte DI310 : 32 entrées TOR

Elle est composée de 32 entrées TOR.

32 LEDs sur la face avant de la carte DI310 représentent l'état des 32 entrées.

Notation : %IXi.j avec :

- L'index "i" (1≤i≤15) donne la position du bloc DI310
- L'index "j" (0≤j≤31) numéro de l'entrée, pour I1 à I32

Carte DI410 : 64 entrées TOR

Elle est composée de 64 entrées TOR.

64 LEDs sur la face avant de la carte DI410 représentent l'état des 64 entrées.

Notation : %IXi.j avec :

- L'index "i" (1≤i≤15) donne la position du bloc DI410
- L'index "j" (0≤j≤63) numéro de l'entrée, pour I1 à I64

Carte DI312 : 32 entrées de sécurité

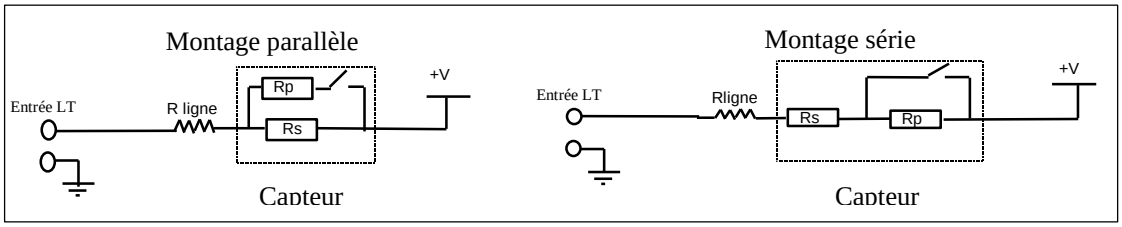
La **carte DI312** comporte un dispositif paramétrable de comparaison pour **contrôler la filerie** des capteurs en leurs connectant un réseau de 2 résistances.

Les réseaux de résistances sont de 2 types :

- **montage série** : 2 résistances en série
- **montage parallèle** : 2 résistances en parallèle.

La résistance série est toujours présente. Dans le montage parallèle, le capteur est en série avec Rp qu'il supprime par ouverture. Dans le montage série, le capteur est en parallèle avec Rp qu'il supprime par fermeture.

Câblage des entrées de sécurité:



La résistance équivalente du réseau de résistances lorsque le capteur est **normalement ouvert (RCNO)** et lorsque le capteur est **normalement fermé (RCNF**) doit être calculé. Les valeurs des résistances sont **en Ohms**.

Montage parallèle	**Montage série**
RCNF = Rs//Rp + Rligne	RCNF = Rs + Rligne
RCNO = Rs + Rligne	RCNO = Rs+Rp + Rligne

ATTENTION : Le **paramétrage est unique** pour les valeurs de résistance d'une **carte DI312** donc le même pour toutes les voies d'un même module DI312.

Contraintes sur les résistances :

- Les résistances doivent avoir une tolérance de 1% max.
- 0.7 kΩ < Rcno < 22 kΩ
- Rcno induit le courant dans le dispositif de mesure : I(mA) = 22(V) / (1+Rcno(kΩ)) sachant que I doit être compris entre 1 mA et 9.96 mA . Si I calculé est supérieur à 9.96 mA, le saturer à 9.96 mA.

  (2.95(V) / I (mA)) < Rcnf (kΩ) < Rcno (kΩ) - Rligne (kΩ) - 1.95 (V) / I (mA)
- Rligne < 0.2 kΩ : c'est la résistance du conducteur reliant le capteur à la carte DI312 : elle est calculée suivant la formule : ρ * L / S , avec :
  - ρ : résistivité du cuivre : ρ = 1.72 10^{-8} Ω.m pour T° = 20 °C,
  - L : longueur (m) du conducteur aller retour du bornier carte DI312 au capteur ,
  - S : section du conducteur en m².

Paramètrage de la carte:

Il est réalisé par la fonction suivante, à appeler dans un programme lors de l'initialisation.

Fonction	**DI312CONFIG**
Action	Paramétrage de la carte DI312
Paramètres	➢ BoardOrder (USINT) : numéro d'ordre de la carte à paramètrer (1 à 15) ➢ Mask (UDINT) : masque de 32 bits du contrôle de filerie des 32 entrées. Contrôle de filerie actif sur l'entrée n si le bit de rang n est à l'état 0. ➢ RCNO (UDINT) : valeur unique pour toutes les entrées. ➢ RCNF (UDINT) : valeur unique pour toutes les entrées. ➢ Rligne (UDINT) : valeur unique pour toutes les entrées.
Valeur retournée	(SINT) : fonction status [0..FFh] Status = 0 : succès. Status <> 0 : erreur.
Exemple	*StatusDI312_3 := DI312CONFIG (3, 16#FFFFFF90, 2000, 1000, 100);* (* carte emplacement 3, contrôle sur les entrées n°0, 1, 2, 3, 5, 6, montage résistances en série de 1kΩ*)

La carte DI312 est composée de 2 sous cartes :

- INPUT: 32 entrées TOR
- FAULT: 32 alarmes

32 LEDs vertes sur la face avant de la carte DI312 représentent l'état des 32 entrées, 32 LEDs rouges représentent les 32 alarmes des entrées.

Etat Entrée LED verte	**Alarme (LED rouge)**	**Description**
0 (éteinte)	0 (éteinte)	capteur normalement ouvert
1 (allumée)	0 (éteinte)	capteur normalement fermé
0 (éteinte)	1 (allumée)	entrée non connectée ou court-circuit au 0V
1 (allumée)	1 (allumée)	court circuit au +V

Notation : L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.

- Entrées : %IXi.0.j : index "j" (0≤j≤31) numéro de l'entrée.
- Alarmes : %IXi.1.j : index "j" (0≤j≤31) numéro de l'alarme.

Carte DO310 : 32 sorties TOR

Elle est composée d'une seule carte : 32 sorties TOR sont disponibles.
32 LEDs sur la face avant de la carte DO310 représentent l'état des 32 sorties.

Notation : %QXi.j avec :

- L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.
- L'index "j" (0≤j≤31) numéro de la sortie, pour O1 à O32

Carte DIO210 : 16 entrées et 8 sorties TOR

Elle est composée de 2 sous cartes :

INPUT: 16 entrées TOR

OUTPUT: 8 sorties TOR

16 LEDs vertes sur la face avant de la carte DIO210 représentent l'état des 16 entrées, 8 LEDs rouges représentent l'état des 8 sorties.

Notation : L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.

- Entrées : %IXi.0.j : index "j" (0≤j≤15) numéro de l'entrée.
- Sorties : %QXi.1.j : index "j" (0≤j≤7) numéro de la sortie.

Carte DI130 : 16 entrées TOR redondées

Elle est composée de 2 sous cartes :

- I: 16 entrées booléennes (signaux)
- Iinv: 16 entrées booléennes complémentées (signaux complémentés)

Chaque entrée est lue à travers deux circuits électroniques indépendants : l'un fournit la valeur de l'entrée directe, l'autre fournit la valeur de l'entrée complémentée.
Ce système de double lecture permet de vérifier par comparaison l'intégrité de chacun des deux circuits, et donc en cas de discordance dans la mesure de détecter une erreur interne sur l'un des deux circuits.
Exemple : FaultI1 est pour l'entrée I1 et sera à VRAI s'il y a un problème sur une des deux mesures.
FaultI1 := NOT (I1 XOR Iinv1) ;

Notation :

- L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.
- Entrées directes : %IXi.0.j : index "j" (0≤j≤15) numéro de l'entrée pour I1 à I16.
- Entrées inverses : %IXi.1.j : index "j" (0≤j≤15) numéro de l'entrée pour Iinv1 à Iinv16

Carte DIO130 : entrées sorties de sécurité

La carte est divisée en 4 groupes de contrôle/commande. Un groupe contrôle un appareillage électrique. Chaque groupe comprend :

- 2 sorties à relais. Chaque relais x=[1..8] est commandé par 2 bits Rx et Rxs
- 2 entrées logiques:
  - Une entrée qui indique la position fermée de l'appareillage (LED rouge).
  - Une entrée qui indique la position ouverte de l'appareillage (LED verte)
- 1 LED de signalisation par groupe (Sa, Sb, Sc, Sd)

Les entrées/sorties sont rassemblées en 3 sous cartes:

- RELAYS : 16 sorties relais (remote controls) : R1 à R8, R1s à R8s
- SIGNAL : 4 sorties de signalisation (LEDs)
- INPUT : 8 entrées TOR (remote signaling)

Notation :

- L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.
- Carte des commandes des sorties relais : **%QXi.0.j** avec j selon le tableau suivant :

J	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Relais	R2	R4	R6	R8	R2s	R4s	R6s	R8s	R1	R3	R5	R7	R1s	R3s	R5s	R7s

- Carte des commandes des LED de signalisation : **%QXi.1.j** avec j selon le tableau suivant :

J	0	1	2	3
LED	Sa	Sb	Sc	Sd

- Carte des entrées : **%IXi.2.j** avec j selon le tableau suivant :

J	0	1	2	3	4	5	6	7
Entrée	I2	I4	I6	I8	I1	I3	I5	I7

Carte AI110 : 8 entrées analogiques

Elle est composée d'une seule carte : 8 entrées analogiques sont disponibles.
8 LEDs vertes sur la face avant de la carte AI110 représentent les 8 entrées : elles sont allumées dès la gestion de la carte.

Notation : %IWi.j avec :

- L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.
- L'index "j" (0≤j≤7) numéro de l'entrée.

Conversion signaux analogiques / numériques :

Table de conversion courant/tension <=> nombre de points	
Entrées courant	±21,1mA => ±32767 points.
Entrées tension	±10,25V => ±32767 points.

Carte AI210 : 16 entrées analogiques

Elle est composée d'une seule carte : 16 entrées analogiques sont disponibles.
16 LEDs vertes sur la face avant de la carte AI210 représentent les 16 entrées : elles sont allumées dès la gestion de la carte.

Notation : %IWi.j avec :

- L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.
- L'index "j" (0≤j≤15) numéro de l'entrée.

Conversion signaux analogiques / numériques :

Table de conversion courant/tension <=> nombre de points	
Entrées courant	±21,1mA ➔ ±32767 points.
Entrées tension	±10,25V ➔ ±32767 points.

Carte AO121 : 8 sorties analogiques

Elle est composée d'une seule carte : 8 sorties analogiques sont disponibles.
8 LEDs orange sur la face avant de la carte AO121 représentent les 8 sorties : elles sont allumées dès la gestion de la carte.

Notation : **%QWi.j** avec :

- L'index "i" = [1 … 15] = la position logique du bloc sur l'embase.

- L'index "j" = [0..7]= le numéro logique de la sortie dans le bloc.

Conversion signaux analogiques / numériques :

Table de conversion courant/tension <=> nombre de points	
Sorties courant	0 / 32767 points => 4 / 20mA
Sorties tension	±32767 points => ±10V

Carte AIO320 : 8 entrées et 4 sorties analogiques

Elle est composée de 2 sous cartes :

- INPUT: 8 entrées analogiques
- OUTPUT: 4 sorties analogiques

8 LEDs vertes sur la face avant de la carte AIO320 représentent les 8 entrées, 4 LEDs oranges représentent les 4 sorties : elles sont allumées dès la gestion de la carte.

Notation : L'index "i" = [1 … 15] donne la position logique du bloc sur l'embase.

- Entrées : **%IWi.0.j** : index "j" (0≤j≤7) numéro de l'entrée.
- Sorties : **%QWi.1.j** : index "j" (0≤j≤3) numéro de la sortie.

Conversion signaux analogiques / numériques :

Table de conversion courant/tension <=> nombre de points	
Entrées courant non isolées	±20mA => ±32767 points.
Entrées tension non isolées	±10V => ±32767 points.
Sorties courant	0 / 32767 points => 4 / 20mA
Sorties tension	±32767 points => ±10V
Entrées sondes PT100	-50°C=>-500 points +350°C=>+3500 points

Status des cartes d'Entrées/Sorties

La fonction ***IOStatus(Position Logique du bloc)*** lit le status des cartes d'entrées/sorties :

Fonction	**IOStatus**
Parametres	Position logique du bloc = [0..15] : 0=CPU, 1=1er bloc I/O, ... , 15 =15ème bloc I/O
Valeur retournée	(INT) = [0..FFFFh] Status = 0: bloc inaccessible Status = -1: Mauvaise Position Logique ou Bloc non existant dans la librairie
Exemple	*Status1 := IOStatus(1);* (* litt le status du bloc en position 1 *)

Composition du mot de Status dans le cas d'un bloc Unité centrale

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signification	RESERVED										PRM	Wdg output	init	IO fault	Wdg LED	Cycle

BIT	Commentaire
0	Mis à 1 à la fin de chaque cycle
1	À 1 si le LED WDG est allumée
2	À 0 si il y a un défaut I/O. à 1 si OK
3	Mis à 1 à la fin des des initialisations et à la fin de chaque cycle
4	Mis à 0 si le WDG est active. À 1 si OK
5	Mis à 1 si l'état PRM est active
6-15	Réservés

Composition du mot de status dans le cas d'un bloc d'entrées/sorties:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signification	Position				Fault	Wdg	Alim	VCC	Code de la carte							

- **Position** (Bits 12 à 15) du bloc sur l'embase = [1 …15].
- **Fault** (Bit 11) est une alarme spécifique à la carte. Voir le tableau ci-dessous.
- **Wdg** (Bit 10) est l'état du Chien de Garde de la carte.
  - à 1 au repos,
  - à 0 si le Chien de Garde déclenché.
- **Alim** (Bit 9) est l'état de l'alimentation externe de puissance sur les connecteurs. Voir tableau ci-dessous.
- **Vcc** (Bit 8) est l'état de l'alimentation logique de la Carte.
  - 1 si OK
  - 0 si défaut.
- **Le code de la carte (Bits 0 à 7)** permet de distinguer les différentes cartes existantes **:**

Carte	Status Bit 11	Status Bit 9	Code de la carte
DI310	1	Al Ext	03h ou 43h (masquer le bit 6)
DI312	NS	Al Ext(*)	14h
DI410	1	Al Ext	06h
DI130	Fault	Al Ext	59h
DIO130	Fault	Al Ext	58h
DO310	Surcharge	Al Ext	05h ou 45h (masquer le bit 6)
DIO210	Monostable	Vrel	16h
AI110	0	0	80h
AI210	0	0	81h
AO121	0	0	88h
AIO320	Monostable	0	83h

Avec :

- NS : non significatif
- Al Ext : à 1 si l'alimentation extérieure est présente sur les borniers et comprise entre Valim ±20%.
- (*) Al ext de la DI312 : Alimentation extérieure entre 24V ± 10%
- Surcharge : à 0 si surcharge sur une voie de sortie TOR.
- Monostable : à 1 si carte correctement rafraîchie
  - ATTENTION : le fonctionnement est inverse sur le module DIO130
- VRel : à 1 si les relais sont correctement alimentés.

Page laissée blanche intentionnellement

Chapitre

4

Fonctions de la CPU

Présentation

Ce chapitre décrit les fonctions propres à la CPU :

- Gestion des paramètres IP
- Gestion de l'heure
- Stockage de données
- Pages Web

Gestion des paramètres IP

Les CPU6xx disposent de 1 ou 2 ports Ethernet. Leur configuration peut être réalisée

- via le Registry Host (cf Chap3),
- via la fonction IPCONFIG à appeler une seule fois lors de l'initialisation du programme :

Fonction	**IPCONFIG**
Action	Configure un port Ethernet
Paramètres	• (STRING) Nom du port Ethernet : 'eth0' ou 'eth1' • (STRING) Adresse IP du port Ethernet. • (STRING) Masque de sous réseau du port Ethernet. • (STRING) Passerelle du port Ethernet.
Valeur retournée	(INT) : status 1 : commande exécutée. Autre valeur : erreur d'exécution de la commande.
Exemple	*ConfigEth0:=IPCONFIG('eth0','192.168.1.200','255.255.255.0','192.168.1.1');*

Gestion de l'heure

La CPU6xx dispose d'une horloge logicielle secourue. Cette horloge fournit la date, l'heure et le jour de la semaine. Ces données peuvent être lues via les fonctions inclues dans Straton : DTCurDate, DTCurTime, DTCurDateTime, DTDay, DTMonth, DTYear, DTSec, DTMin, DTHour, DTMs.

La mise à l'heure de l'horloge peut être réalisée par différents moyens :

- par le protocole NTP, automatiquement.
  Le paramètrage NTP peut être réalisé :
  - par paramétrage via le RegistryHost (cf Chap3),
  - par programme via deux fonctions « NTPSTOP » et « NTPSTART »
- par le protocole DNP3 (cf Chap8), automatiquement.
- par les fonctions « DATE_WRITE » et « TIME_WRITE », par programmation.

Les quatre fonctions spécifiques sont décrites ci-dessous.

Fonction	**NTPSTOP**
Action	Arrête le client NTP.
Paramètres	-
Valeur retournée	(INT) : status 1 : commande exécutée. Autre valeur : erreur d'exécution de la commande.
Exemple	*Stop_ntp:= NTPSTOP () ;*

Fonction	**NTPSTART**
Action	Démarre le clien NTP
Paramètres	• (STRING) Adresse IP du serveur de temps. • (UDINT) Fréquence de synchronisation avec le serveur (en seconde). • (UINT) Code du fuseau horaire (cf Annexe1 « Codes des fuseaux horaires »).
Valeur retournée	(INT) : status 1 : commande exécutée. Autre valeur : erreur d'exécution de la commande.
Exemple	*Start_ntp:= NTPSTART ('192.168.239.250',60, 105) ;*

Fonction	**DATE_WRITE**
Action	Mise à jour de la date
Paramètres	(STRING) Date au format JJ/MM/AAAA
Valeur retournée	(BOOL) : status TRUE : Date mise à jour. FALSE : Echec de la mise à jour de la date.
Exemple	*Status_date:= DATE_WRITE('20/10/2014') ;*

Fonction	**TIME_WRITE**
Action	Mise à jour de l'heure
Paramètres	(STRING) Heure au format hh:mm:ss ou hh:mm:ss.ms
Valeur retournée	(BOOL) : status TRUE : Heure mise à jour. FALSE : Echec de la mise à jour de l'heure.
Exemple	*Status_date:= TIME_WRITE('12 :34 :56.789') ;*

Stockage de données

La CPU6xx dispose d'une mémoire Flash. La taille disponible pour l'utilisateur est de 1 Mo. L'utilisateur peut y stocker des fichiers contenant soit des mots de 16bits, soit des chaines de caractères.

L'utilisateur doit gérer l'espace de la mémoire. Il doit utiliser les fonctions Straton de gestion de fichier : F_ROPEN, F_WOPEN, F_AOPEN, F_CLOSE, F_EOF, …. ; se reporter à l'aide en ligne de Straton pour les détails de mise en œuvre.

L'utilisation de ces fonctions doit être prudente et mesurée car le temps de cycle de l'application peut en être affecté.

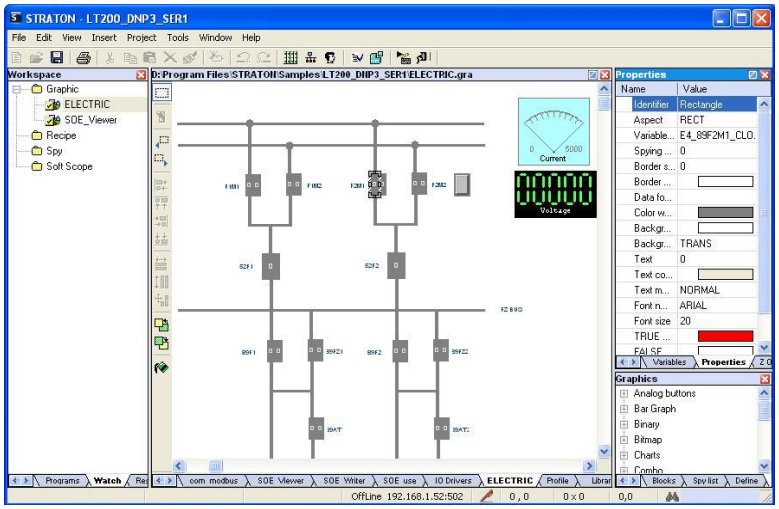
Lecture des fichiers clients : connexion ftp

The CPU6xx contient un serveur FTP. La lecture via un client FTP des fichiers créés par le programme applicatif est donc possible.
Les paramètres de connexion doivent utiliser le type « anonymous ».

Pages Web

Des pages web peuvent être stockées dans la mémoire Flash.

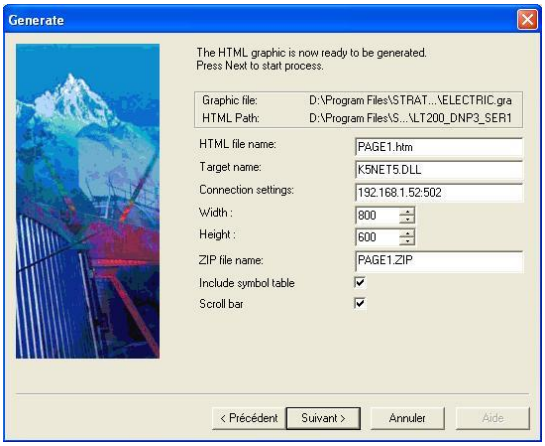
Dans votre projet Straton, pour créer un nouveau document graphique, cliquer sur « Ajouter un nouvel élément » « Graphique ».



L'atelier Straton contient un grand nombre d'objets et fonctionnalités relatif aux pages graphiques : voir l'aide en ligne de l'atelier Straton pour plus de détail.

Une fois votre page graphique développée, elle doit être transformée en page Web puis exportée dans le LT. Les étapes à respecter pour cela sont les suivantes :

- Sélectionner votre page dans Straton.
- cliquer sur le menu *Outils/Générer une page HTML*
- cliquer sur le bouton « Suivant » et remplir les différents champs :
  - Nom du fichier HTML
  - Nom de la cible : DLL utilisé par l'ActiveX Straton: « K5NET5.DLL »
  - Paramètres de connexion : adresse IP du LT suivie par « :502 » (port TCP)
  - Largeur : de la page Web
  - Hauteur : de la page Web
  - Nom du fichier ZIP : les caractères doivent être en majuscule.



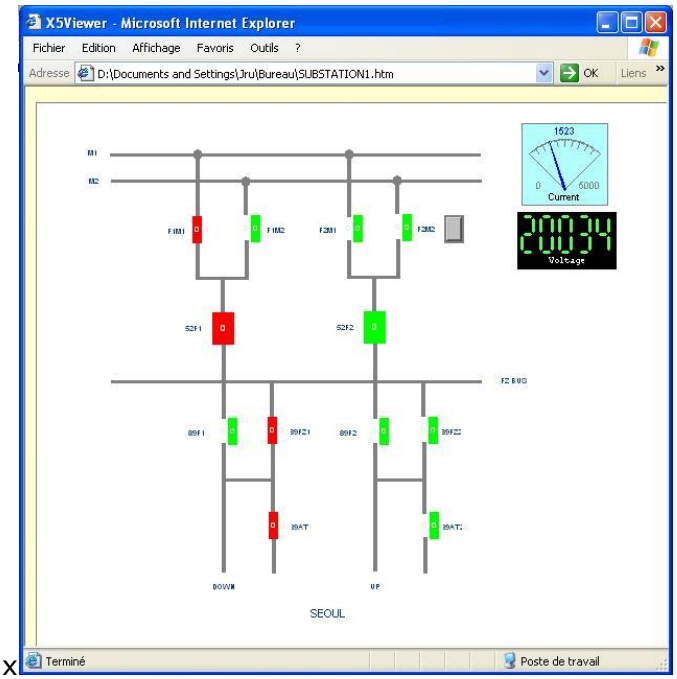
- Cliquer sur le bouton "Suivant" : cette action va automatiquement :
  - Générer le ficher ZIP et le transférer dans le LT.
  - Générer la page HTML locale.

Pour voir la page web dans l'automate depuis votre navigateur internet, les éléments suivants doivent être installés sur votre ordinateur :

- Internet Explorer ou un tout autre navigateur gérant les activeX
- Straton ActiveX (installé automatiquement avec l'atelier Straton)
- Le fichier HTML créé lors de l'opération précédente : par exemple « Page1.htm »

Ouverture de la page Web : double cliquer sur le fichier HTML.

Votre navigateur internet va afficher la page HTML et établir automatiquement la connexion avec le LT, pour rafraîchir toutes les variables de votre page HTML.



Chapitre

5

Communication Série et Ethernet sans protocole

Présentation

Ce chapitre décrit la communication sur port série simple et sur Ethernet sans protocole. Le LT utilise les fonctions intégrées dans l'atelier Straton. Leur mise en œuvre détaillée est décrite dans l'aide en ligne de l'atelier Straton.

Nous détaillons dans ce chapitre les étapes suivantes :

- Principe de la communication série
- fonctions d'initialisation et de fermeture
- fonctions de lecture et écriture d'octets
- fonctions de lecture et écriture de caractères

Principe de communication

Le LT peut gérer simultanément sur chacune de ses 4 liaisons séries une communication série sans protocole avec d'autres équipements : l'utilisateur du LT Straton a la possibilité d'implémenter son propre protocole d'émission/réception d'octets ou caractères, par l'utilisation de fonctions.

Les fonctions fournies permettent d'installer et de gérer des files d'attente de type FIFO : une en émission et une en réception.

Une liaison série est gérée indifféremment en RS232, RS485 ou en RS422.

Après une initialisation de la liaison série, l'utilisateur peut lire ou écrire des octets dans les files d'émission et de réception. Le LT Straton se charge, sous interruption, d'émettre ou de recevoir les octets sur la ligne.

Communication sur liaison série

Le LT Straton utilise le bloc fonctionnel intégré à l'atelier « SERIO », qui permet de :

- Configurer et ouvrir le port de communication choisi
  Le paramètre d'entrée « CONF » de ce bloc doit avoir la forme suivante :
  exemple : ' tts/1:9600,N,8,1,0' :
  - **tts/**1 : numéro du port série utilisé : 0 pour com0, 1 pour com1, 2 pour com2, 3 pour com3
  - 19200 : vitesse de communication : vitesses possibles : 2400, 4800, 9600, 19200, 38400.
  - N : parité : N (aucune) ou E (pair) or O (impair)
  - 8 : nombre de bits de données (7 ou 8)
  - 1 : nombre de bits de stop (1 ou 2)
  - 1 : mode : 0 pour RS232 ou 1 pour RS485
- Envoyer des caractères sur ce port
- Vérifier si des caractères ont été reçus,
- Recevoir des caractères sur ce port

Communication sur TCP-IP et UDP

Le LT Straton dispose des fonctions Straton suivantes pour gérer des « sockets » TCP-IP ou « UDP », afin de réaliser des applications clientes ou serveur sur Ethernet :

- tcpListen : crée une socket serveur.
- tcpAccept : accepte une connexion cliente.
- tcpConnect : crée une socket cliente et la connecte à un serveur.
- tcpIsConnected : teste si une socket client est connectée.
- tcpClose : ferme une socket.
- tcpSend : envoi de caractères.
- tcpReceive : réception de caractères.
- tcpIsValid : teste si une socket est valide.

L'utilisation de ces fonctions nécessite la maîtrise des principes de communication des réseaux TCP/IP : une mauvaise utilisation de ces fonctions peut dégrader le fonctionnement du LT.

Chapitre

6

Communication Modbus

Présentation

Ce chapitre décrit le protocole Modbus et sa mise en œuvre simplifiée. Le LT utilise le protocole Modbus intégré dans l'atelier Straton. Pour les fonctions avancées se reporter à l'aide en ligne de l'atelier Straton.

Nous détaillons dans ce chapitre les étapes suivantes :

- Protocole Modbus
- Protocole Modbus Esclave : RTU et TCP
- Protocole Modbus Maître : RTU et TCP

Dans Straton, depuis l'espace de travail, ouvrir le configurateur de bus de terrain.

Protocole Modbus

Modbus est un protocole de communication permettant d'échanger des données entre plusieurs équipements ; c'est un protocole de type maître / esclave, avec pour support physique soit une liaison série (RS232, RS485, RS422), soit le support Ethernet (10/100Mb).

Ce protocole est décrit dans plusieurs documents téléchargeables : http://www.modbus.org/

Le LT peut gérer simultanément les fonctionnalités suivantes :

- sur chacune de ses 4 liaisons séries : maître ou esclave modbus RTU
- sur les liaisons Ethernet : maître et esclave modbus/TCP

Une table de données pourra être associée à chaque esclave.

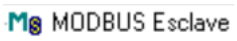
Les données sont de type bit et mot (16 bits).

Les codes modbus gérés par le LT sont :

- 1 : lecture de bits
- 2 : lecture de bits d'entrée
- 3 : lecture de mots
- 4 : lecture de mots d'entrée
- 5 : écriture d'un bit
- 6 : écriture d'un mot
- 15 : écriture de bits
- 16 : écriture de mots

Protocole Modbus esclave

Pour initialiser le service Modbus, aller dans le configurateur de bus de terrain Straton et suivre les étapes suivantes :

Ajouter une configuration : 

Cliquer sur le bouton « Insérer configuration », puis choisir dans la liste des protocoles « Modbus esclave ».

Insérer un port :

Cliquer sur le bouton « Insérer un maître / un port », puis choisir le numéro d'esclave modbus.

Pour ajouter un port série modbus esclave RTU, initialiser comme dans l'exemple ci-après, dans un programme par exemple en langage ST :

```
MySlave1 (TRUE, 'tts/1:9600,N,8,1,0', 1);
Q_myslave1 := MySlave1.Q;
```

Avec : « MySlave1 » une instance du bloc fonction « MBSlaveRTU ».

Les paramètres du bloc MBSlaveRTU sont :

- IN : (BOOL) Activation de la commande : le port est ouvert quand ce paramètre est à VRAI.
- PORT : (STRING) – chaîne de caractère ; exemple : ' tts/1:9600,N,8,1,0' :
  - tts/1 : numéro du port série utilisé : 0 pour com0, 1 pour com1, 2 pour com2, 3 pour com3
  - 19200 : vitesse de communication : vitesses possibles : 2400, 4800, 9600, 19200, 38400.
  - N : parité : N (aucune) ou E (pair) or O (impair)
  - 8 : nombre de bits de données (7 ou 8)
  - 1 : nombre de bits de stop (1 ou 2)
  - 1 : mode : type de port RS232 ou RS485 : 0 pour RS232 ou 1 pour RS485
- SLV : (DINT) numéro de l'esclave modbus.

Un autre bloc fonction « MBSlaveRTUex » est disponible : il permet en plus du paramètrage précédent de spécifier l'ID du serveur (paramètre supplémentaire) dans la configuration MODBUS esclave, et donc d'avoir plusieurs instances de blocs actives en même temps sur des ports différents.

Insérer un bloc de données : 

Quand le Serveur est sélectionné, utiliser le lien « Insérer Esclave / Bloc de données » pour insérer un bloc de données. Les données suivantes sont disponibles :

- Données lues par le maître :
  - Bits d'entrée : bits lus par le maître (fonction 2).
  - Mots d'entrée : mots lus par le maître (fonction 4).
- Données lues ou forcées par le maître :
  - Bits de sortie : bits lus ou forcés par le maître (fonction 1, 5 ou 15).
  - Mots de sortie : mots lus ou forcés par le maître (fonction 3, 6 ou 16).

Chaque bloc de données est identifié par une adresse de base modbus et un nombre de données (bits ou mots).

Insérer une variable : 

Quand un bloc de données serveur est sélectionné, utiliser le menu *Insérer/Insérer une variable* pour ajouter une variable à votre bloc. Chaque variable est identifiée par un symbole valide dans votre projet et un offset par rapport à l'adresse de base du bloc.

Pour échanger des variables booléennes à travers des mots modbus, un masque hexadécimal est disponible pour définir quel bit du mot la variable doit être rattachée.

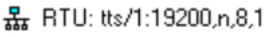
Les variables de chaque bloc peuvent être triées selon leur offset par la commande du menu *Edition / Trier les symboles*.

Protocole Modbus maître

Le service Modbus doit avant tout avoir été initialisé. Dans le configurateur de bus de terrain Straton, suivre les étapes suivantes :

Ajouter une configuration : 

Cliquer sur le bouton « Insérer configuration », puis choisir dans la liste des protocoles « Modbus maître ».

Insérer un port : 

Cliquer sur le bouton « Insérer un maître / un port », puis choisir le type de port, Ethernet ou série, puis les paramètres correspondants à celui-ci :

- Port « MODBUS sur Ethernet » : renseigner les champs prévus :
  - l'adresse IP de l'esclave modbus/TCP.
  - Le numéro du port TCP de l'esclave modbus/TCP: 502
  - Le protocole : TCP – Open MODBUS (les protocoles UDP – Open MODBUS et UDP – Modbus RTU ne sont pas supportés par le LT).

- Port « Série : MODBUS-RTU » : renseigner le champ « Port de com. » comme spécifié ci-dessous :
  exemple : ' tts/1:19200,N,8,1,1' :
  - tts/1 : numéro du port série utilisé : 0 pour com0, 1 pour com1, 2 pour com2, 3 pour com3
  - 19200 : vitesse de communication : vitesses possibles : 2400, 4800, 9600, 19200, 38400.
  - N : parité : N (aucune) ou E (pair) or O (impair)
  - 8 : nombre de bits de données (7 ou 8)
  - 1 : nombre de bits de stop (1 ou 2)
  - 1 : mode : type de port RS232 ou RS485 : 0 pour RS232 ou 1 pour RS485

Insérer un bloc de données : 

Quand le port est sélectionné, utiliser le lien « Insérer Esclave / Bloc de données » pour insérer un nouveau bloc de données. Les paramètres de la requête maître modbus sont :

- « Esclave/unité » : numéro d'esclave modbus
- « Requête MODBUS » : numéro de fonction modbus (1,2,3,4,5,6,15, ou 16)
- « Bloc de données » : adresse de base dans l'esclave et nombre d'items lus ou écris dans l'esclave
- « Activation » : type de requête, périodique, sur demande ou sur changement d'état.
- Timeout et nombre d'essais de la requête sur timeout.

Insérer une variable :

Quand un bloc de données est sélectionné, utiliser le menu *Insérer/Insérer une variable* pour ajouter une variable à votre bloc. Chaque variable est identifiée par un symbole valide dans votre projet et un offset par rapport à l'adresse de base du bloc.

Pour échanger des variables booléennes à travers des mots modbus, un masque hexadécimal est disponible pour définir quel bit du mot la variable doit être rattachée.

Les variables de chaque bloc peuvent être triées selon leur offset par la commande du menu *Edition / Trier les symboles*.

Page laissée blanche intentionnellement

Chapitre

7

Protocole T5 événementiel : Binding

Protocole T5 événementiel

Le protocole T5 événementiel sur TCP-IP permet d'échanger des variables entre plusieurs LT. Le protocole étant purement événementiel, il assure des temps d'échange très courts et une faible charge du réseau. Ce protocole peut être utilisé sur des réseaux Ethernet redondants. Pour les fonctions avancées se reporter à l'aide en ligne de l'atelier Straton.

Mécanisme

Le protocole T5 est basé sur le modèle de "publication-souscription". Chaque LT peut produire (publier) des variables sur le réseau, et consommer (souscription) des variables produites par les autres LT. Chaque variable produite est identifiée par un numéro (identificateur). Cet identificateur est utilisé pour associer les variables source et destination dans les LT respectifs. La même variable produite par un LT peut être consommée par plusieurs LT.

Echanges

La valeur de la variable n'est émise sur le réseau que lorsqu'elle change. Pour chaque variable produite, définir des hystérésis positif et négatif afin de régler la charge du réseau en fonction des besoins de votre application. Chaque nouvelle valeur émise sur le réseau est datée. Pour chaque variable consommée, la datation de la dernière valeur reçue, ainsi qu'un flag de qualité sont disponibles. Un statut global de la connexion pour chaque producteur relié est aussi disponible.

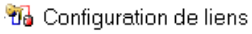
Limitations

Le nombre maximum de variables produites par un LT est limité à 256 variables.

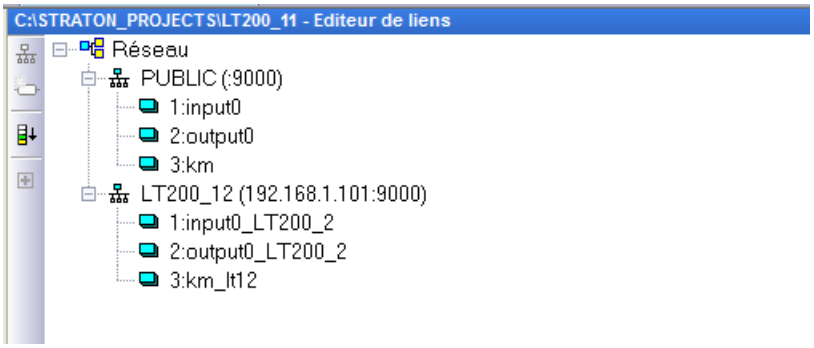
Seuls les variables booléennes, entières et TIME peuvent être échangées. Les variables STRING et les instances de blocs fonctionnels ne sont pas supportés par le protocole.

Mise en oeuvre

L'atelier offre un outil global pour spécifier les liaisons entre plusieurs projets. Pour bâtir une application distribuée :

- Créer les projets (pour chaque LT) avec l'atelier.
- Dans chaque projet LT, depuis le menu de la fenêtre principale de l'atelier, exécuter la commande *Outils/Editeur de liens* ou cliquer sur le raccourci « Configuration de liens ». 

Définition des configurations et échanges



Cliquer sur le bouton "Insérer un maître / un port"  de l'éditeur de liens pour définir un nouvel équipement dont les variables seront consommées par le LT.

Le groupe « PUBLIC » correspond au groupe de variables qui seront produites par le LT.

Cliquer sur le bouton "Insérer une variable"  pour ajouter une variable dans le projet existant.

Chaque LT est identifié comme un "projet" dans l'éditeur, et par son nom, une adresse IP et un numéro de port Ethernet. Le numéro utilisé pour la publication de variables est 9000.

Réseaux Ethernet redondants

Dans le cas de réseaux Ethernet redondants, spécifier deux adresses IP au lieu d'une seule pour chaque noeud concerné. Pour ceci, saisir deux adresses IP séparées par un signe de division "/".

L'information "Etat de la connexion" reste disponible dans le cas de liens redondants. Dans ce cas, la valeur de la variable connectée à l'état reflète l'état des deux connexions.

L'état est stocké sous la forme d'un entier. L'état de la première connexion est dans les bits 0 à 3, et l'état de la seconde dans les bits 4 à 7. Pour chaque connexion, la valeur "0" signifie OK.

Chapitre

8

Protocole DNP3.0 esclave

Présentation

Ce chapitre décrit la communication DNP3.0 sur port série simple et sur Ethernet sur le LT. Le LT utilise les fonctions intégrées dans l'atelier Straton. Leur mise en œuvre détaillée est décrite dans l'aide en ligne de l'atelier Straton.

Nous détaillons dans ce chapitre les étapes suivantes :

Principe de la communication
Configuration DNP3 esclave
Câblage des variables DNP3

Le document DNP3 Profil et la table d'implémentation sont en Appendice.

Principe de communication

Le Protocole de Réseau Distribué (DNP 3.0) est un standard de communication dans l'industrie entre des systèmes de supervision et des RTU ou IED (Intelligent Electronic Devices). Le protocole DNP est un protocole entre un "Maître" et un "Esclave". Les maîtres sont les systèmes de type supervision, les esclaves sont des RTU ou systèmes IED. Chaque nœud DNP sur le réseau DNP3 a une adresse : cette adresse permet aux maîtres de demander les données spécifiques de chacun des esclaves.

Le protocole DNP est utilisé pour l'automatisation des sous stations électriques, gestion des réseaux de gaz ou gestion des réseaux d'eau.

DNP est géré par le groupe des utilisateurs DNP : toutes informations sur ce protocole est disponible sur : http://www.dnp.org/

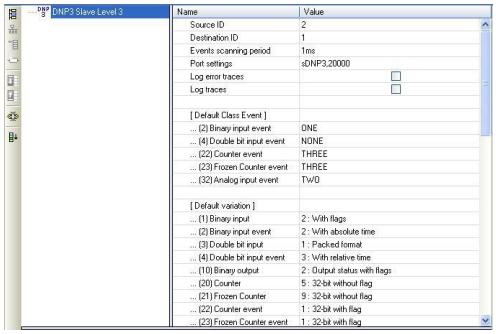
Le LT est un esclave DNP3.0 niveau 2. Il peut être un esclave sur liaison série ou/et sur Ethernet, en fonction des paramétrages logiciels et du câblage réalisé :

- DNP3.0 esclave série : un ou deux de ses ports série peut être utilisé ; le cas de deux ports série configurés est pour l'impémentation de la redondance de ligne.
- DNP3.0 esclave Ethernet : deux connections esclaves peuvent être paramétrées : les ports TCP doivent alors être différents : 20000 & 20001 par exemple.

configuration DNP3 esclave

L'atelier inclut un configurateur de réseau de terrain : cliquer sur le bouton 
et insérer alors une configuration "DNP3 Slave Level3".

L'outil de configuration permet de décrire alors les propriétés de l'esclave DNP3.



Propriétés principales

Nom	**Valeur**
Source ID	"Source identifier" du LT esclave
Destination ID	"Destination identifier" pour le maître (Superviseur)
Event Scanning period	Période (ms)
Port settings	Paramètre (caractères) du port de COM : exemples : Ethernet : « sDNP3,20000», avec 20000 numéro du port TCP Série : « tts/0:9600,n,8,1,0» pour un port série simple (1), « tts/0:9600,n,8,1,0;tts/1 » pour un port série double (2)
Log errors	Si coché, les messages d'erreurs sont envoyés à l'atelier.
Log warnings	Si coché, les messages de warnings sont envoyés à l'atelier

(1) détail du paramétrage du port série : « tts/0:9600,n,8,1,0 »

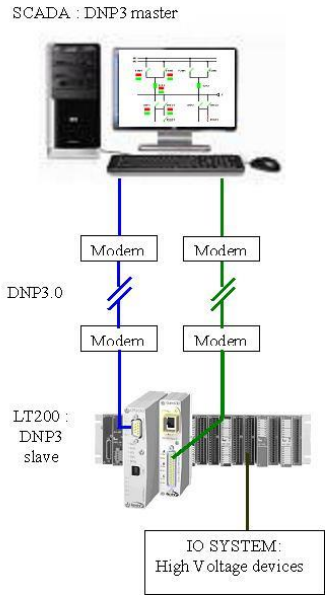
- tts/0 : numéro du port série utilisé : 0 pour com0, 1 pour com1, 2 pour com2, 3 pour com3
- 9600 : baudrate : available speeds : 2400, 4800, 9600, 19200, 38400.
- n : parité : N (aucune) ou E (even : pair) ou O (odd : impair)
- 8 : nombre de bits de donnée (7 ou 8)
- 1 : nombre de bits de stop (1 ou 2)
- 0 : mode : choix du type de port.
  - 0 pour RS232
  - 1 pour RS485

(2) détail du paramétrage du double port série : « tts/0:9600,n,8,1,0;tts/1»

- tts/1 : numéro de ce second port série utilisé : 0 pour com0, 1 pour com1, 2 pour com2, 3 pour com3

Le second port doit être différent du premier. Les autres paramètres pour le second port (vitesse, parité…) sont les mêmes que pour le premier port.

Cette propriété permet d'utiliser une ligne redondante : le maître peut basculer sur la seconde ligne si la première ligne tombe.



Paramètres avancés

Ils sont organisés en sections :

- Default class event
- Default variation
- Event Mode
- Unsollicited messages
- Device attributes
- Misc

Le symbole (*) dans le tableau suivant indique la valeur par défaut.

Default Class Event

Spécifie la classe par défaut qui sera utilisée pour les « unsolicited responses ». Si la classe n'est pas autorisée par le maître, les « unsolicited responses » ne sont pas envoyées pour la classe.

Nom	**Valeur**
(2) Binary input event	NONE (no effect), ONE (*), TWO, THREE
(4) Double bit input event	NONE (no effect) (*), ONE, TWO, THREE
(22) Counter event	NONE (no effect), ONE, TWO, THREE(*)
(23) Frozen Counter event	NONE (no effect), ONE, TWO, THREE(*)
(32) Analog input event	NONE (no effect), ONE, TWO(*), THREE

Default variation

Spécifie la variation qui sera utilisée pour les « unsolicited responses » et en réponse à une requête de lecture de variation 0.

Nom	**Valeur**
(1) Binary input	1(*), 2
(2) Binary input event	1, 2, 3(*)
(3) Double bit input	1(*), 2
(4) Double bit input event	1, 2, 3(*)
(10,12) Binary output	1, 2(*)
(20) Counter	1, 2, 5(*), 6
(21) Frozen Counter	1, 2, 5, 6, 9(*), 10
(22) Counter event	1(*), 2, 5, 6
(23) Frozen Counter event	1(*), 2, 5, 6
(30) Analog input	1, 2, 3(*), 4, 5
(32) Analog input event	1(*), 2, 3, 4, 5, 7
(40) Analog Output status	1(*), 2, 3

Event mode

Spécifie le « event mode » qui sera utilisé pour les « unsolicited responses »:

- ALL : Sequence of Events, retourne tous les événements
- MOST RECENT : seulemenent les événements les plus récents

Nom	**Valeur**
(2) Binary input event	ALL(*), MOST RECENT
(4) Double bit input event	ALL(*), MOST RECENT
(22) Counter event	ALL, MOST RECENT(*)
(23) Frozen Counter event	ALL(*), MOST RECENT
(32) Analog input event	ALL, MOST RECENT(*)

Unsollicited messages

Nom	**Valeur**	**Description**
Unsolicited allowed	Checkbox : Yes = checked NO = not checked	Determines whether unsolicited responses are allowed. If "Unsolicited Allowed" is set to FALSE no unsolicited responses will be generated and requests to enable or disable unsolicited responses will fail.
Unsolicited event mask	NONE(*), ONE, TWO, THREE, ALL	Specify the initial/new state of the unsolicited event mask. This mask is used to determine which event class(es) will generate unsolicited responses. According to the DNP specification, unsolicited responses should be disabled until an 'Enable Unsolicited Response' request is received from the master. Hence this value should generally be NONE, but some masters do not generate the 'Enable Unsolicited Response' message, in

		which case they must be enabled here.
Unsolicited retry number	3	Specify the maximum number of unsolicited retries before changing to the 'offline' retry period. This parameter allows you to specify up to 65535 retries.
Unsolicited retry delay	5s	Specifies the time to delay after an unsolicited confirm timeout before retrying the unsolicited response.
Unsolicited offline retry delay	30s	Specifies the time to delay after an unsolicited timeout before retrying the unsolicited response after Unsolicited retry number have been attempted.
Class 1 : Unsolicited events number	5	If unsolicited responses are enabled, Unsolicited events number specifies the maximum number of events in the corresponding class to be allowed before an unsolicited response will be generated.
Class 2 : Unsolicited events number	5	
Class 3 : Unsolicited events number	5	
Class 1 : Unsolicited events delay	5s	If unsolicited responses are enabled, Unsolicited events delay specifies the maximum amount of time after an event in the corresponding class is received before an unsolicited response will be generated.
Class 2 : Unsolicited events delay	5s	
Class 3 : Unsolicited events delay	5s	

Device attributes (object 0)

Tous les attributs suivants peuvent être modifiés. Double cliquer sur l'item "DNP3 Slave Level 3" pour ouvrir les attributs de l'équipement. Ensuite saisir la valeur appropriée pour chaque attribut.

(Variation) Nom
(211) Identifier of support for user-specific attributes
(212) Number of master-defined data set prototypes
(213) Number of outstation-defined data set prototypes
(214) Number of master-defined data sets
(215) Number of outstation-defined data sets
(216) Max number of binary outputs per request
(217) Local timing accuracy
(218) Duration of timing accuraccy
(219) Support for analog output events
(220) Max analog output index
(221) Number of analog outputs
(222) Support for binary output events
(223) Max binary output index
(224) Number of binary outputs
(225) Support for frozen counter events
(226) Support for frozen counters
(227) Support for counter events
(228) Max counter index
(229) Number of counter points
(230) Support for frozen analog inputs
(231) Support for analog input events
(232) Maximum analog input index
(233) Number of analog input points
(234) Support for double-bit binary input events
(235) Maximum double-bit binary input index
(236) Number of double-bit binary input points
(237) Support for binary input events

(238) Max binary input index
(239) Number of binary input points
(240) Max transmit fragment size
(241) Max receive fragment size
(242) Software version
(243) Hardware version
(245) User-assigned location name
(246) User assigned ID
(247) User assigned device name
(248) Serial number
(249) DNP subset and conformance
(250) Product name and model
(252) Manufacturer's name

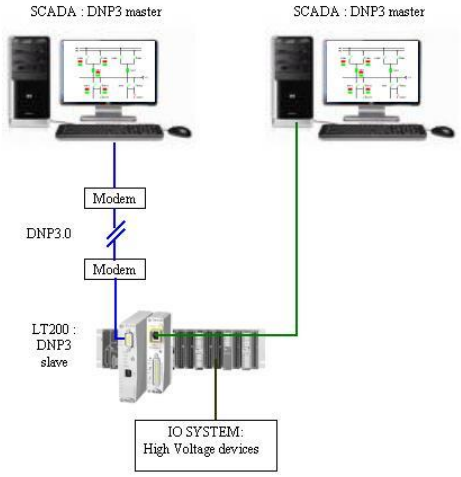
Misc

Nom	**Valeur**	**Description**
Self address enable (code (Oxfffc))		Specify whether or not to enable self address functionality on this slave device. Slave will respond to address 0xfffc as though it received a request for its configured address. It will respond with its own address so the master can automatically discover the slave address.
Clock valid period	30s	Specifies how long the local clock will remain valid after receiving a time synchronization.
Application confirm timeout	10s	Application confirm timeout specifies how long the slave DNP device will wait for an application layer confirmation from the master. This in combination with Unsolicited retry delay or Unsolicited offline retry delay will determine how frequently an unsolicited response will be resent.
Output select timeout	5s	SelectTimeout specifies the maximum amount of time that a select will remain valid before the corresponding operate is received. If an operate request is received after this period has elapsed since the previous select the select will not be valid and the operate request will fail.
Integrity poll response groups	1,3,10,20 ,21,30,40	Object groups included in response to read static data request.
Second port settings		Settings string of the second COM port for a multiport management: examples : Ethernet : « sDNP3,20000», with 20000 as TCP port number Serial : « tts/0:9600,n,8,1,0» for a single serial port (1), « tts/0:9600,n,8,1,0;tts/1 » for a double serial port (2)

Le paramètre "second port settings" permet d'ajouter un second port DNP3 au LT qui répondra alors à chaque maître DNP3 connecté par Ethernet ou un lien série :

Plusieur réseaux de configurations peuvent être réalisés : les DNP3 maîtres peuvent être :

- Des maîtres Ethernet
- Des maîtres série
- un maître Ethernet et un maître série.



Câblage des variables DNP3

L'atelier inclut aussi un outil intégré permettant de gérer les profils des variables : cliquer sur le bouton suivant  pour ouvrir l'outil et cliquer sur "DNP3S".

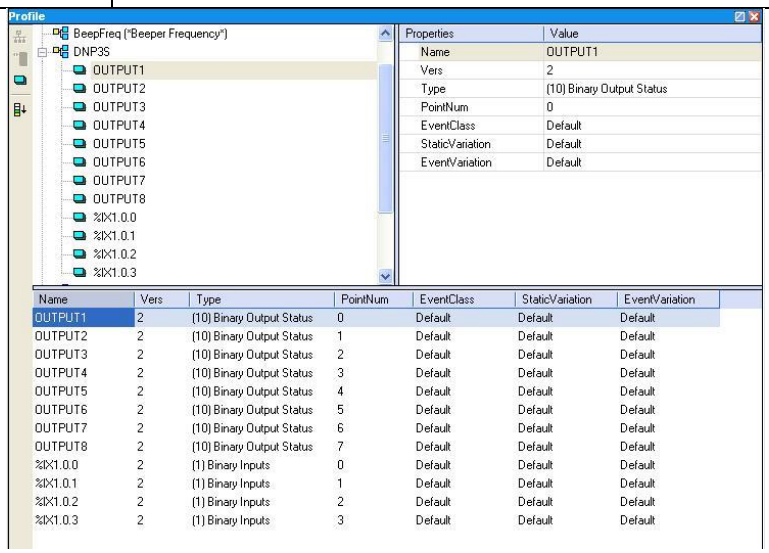
Pour mapper une variable sur le protocole esclave DNP3, sélectionner la variable dans le dictionnaire et faites simplement un glisser et déplacer dans la grille de profil.

Une autre façon de mapper une variable pour le protocole DNP3 esclave, est d'éditer ses propriétés en utilisant la commande Propriétés du menu contextuel dans l'éditeur de variables, puis sélectionner le profil DNP3S.

Ensuite, préciser les propriétés suivantes de la variable:

Pour chaque variable, les propriétés suivantes sont modifiables :

Propriétés	**Description**
Type	(1)Binary Input, (10)Binary Output Status, ...
PointNum	DNP3 point number
EventClass	NONE (no effect), ONE, TWO, THREE
StaticVariation	1 to 9
EventVariation	1 to 9



Chapitre

9

Diagnostic et dépannage

Leds des modules pilotes

Led de l'alimentation

L'alimentation comporte une led verte allumée en fixe si la tension est présente et correcte.

Leds de la CPU

Nom de la LED	Couleur	Signalisation	Signification
RUN	verte	clignote lentement (2s)	si l'application Straton est exécutée correctement.
		clignote rapidement (1/2s)	si l'application Straton est en STOP
		Clignotante en alternance avec la led PRM	phase de démarrage de la CPU
		allumée fixe	mode PRM ou pas à pas
FAIL	orange	éteinte	si fonctionnement correct.
		allumée fixe	si le programme lu dans la Flash est corrompu.
I/O	verte	allumée fixe	si fonctionnement correct.
		clignotante	si insertion carte incorrecte ou si un status carte d'E/S au moins est incorrect au cours de l'exécution du programme.
		clignotante en alternance avec la led Run	phase de démarrage de la CPU
PRM	verte	Allumée fixe	si le mode PRM est détecté au démarrage du LT. Ne s'éteint qu'au prochain reboot du LT sans PRM.
WDG	rouge	allumée fixe	par défaut.
		éteinte	dès que le kernel Straton est en RUN ; le watchdog hardware est rafraichi par le processeur.
CP	verte	Allumée fixe	si le coprocesseur de gestion des cartes d'entrée/sorties fonctionne correctement (programme chargé).
F1	verte		Pilotée par l'utilisateur
F2	verte		Pilotée par l'utilisateur

Leds des voies de communication : série et Ethernet

Nom de la LED	Couleur	Signification
Lk	orange	Liaison Ethernet : Allumée dès connection par cable Ethernet avec un autre équipement Ethernet
Rx	verte	Allumée dès réception d'octets sur la liaison série correspondante
Tx	verte	Allumée dès émission d'octets sur la liaison correspondante
SM	verte	Non utilisé
Te	verte	Non utilisé

Leds des cartes d'entrées / sorties

Les cartes d'entrée et sorties TOR et analogiques comportent des leds sur leur face avant, dont la signification est la suivante :

Type carte	Sérigraphie	Etat de la LED	Signification
Toutes	FLT : couleur rouge	allumée fixe	3 cas possibles: WDG général défaut alimentation interne de la carte pas de pilotage par la CPU
DI310 DI410 DI130 DIO210	1 LED verte par voie d'entrée TOR	allumée fixe	si l'entrée correspondante est à l'état TRUE
DIO130	1 LED verte pour la position ouverte (I2,I4,I6,I8) 1 LED rouge pour la position fermée (I1,I3,I5,I7)	allumée fixe	si l'entrée correspondante est à l'état TRUE
DO310 DIO130 DIO210	1 LED orange par voie de sortie TOR	allumée fixe	si la sortie correspondante est à l'état TRUE
AI110 AI210 AIO320	1 LED verte par voie d'entrée analogique	allumée fixe	si la CPU gère l'acquisition de la voie correctement
AO121 AIO320	1 LED verte par voie de sortie analogique	allumée fixe	si la CPU gère la commande de la voie correctement

Dépannage de la liaison console : passage du LT en mode paramétrage

Le mode paramétrage (PRM) est à utiliser pour rétablir le dialogue entre le PC et le LT Straton, dans le cas de perte de celui-ci suite à un problème avec l'applicatif Straton.

Dans le mode PRM seul le noyau Straton est exécuté mais pas l'application client.

Pour passer en PRM :

- Mettre le LT hors tension,
- Ponter les pins 7 et 8 du COM0
- Mettre le LT sous tension,
- Quelques secondes après le démarrage, la led **PRM** s'allume, le pont peut être retiré,

Straton démarre en mode sans échec et « écoute » l'atelier sur Ethernet et USB.

Annexe1 : Codes des fuseaux horaires utilisés par la fonction NTPSTART

Code	Fuseau horaire GMT	Zone géographique
002	GMT-10:00	Hawaii
003	GMT-09:00	Alaska
004	GMT-08:00	Pacific Time (US and Canada); Tijuana
010	GMT-07:00	Mountain Time (US and Canada)
013	GMT-07:00	Chihuahua, La Paz, Mazatlan
015	GMT-07:00	Arizona
020	GMT-06:00	Central Time (US and Canada
025	GMT-06:00	Saskatchewan
030	GMT-06:00	Guadalajara, Mexico City, Monterrey
033	GMT-06:00	Central America
035	GMT-05:00	Eastern Time (US and Canada)
040	GMT-05:00	Indiana (East)
045	GMT-05:00	Bogota, Lima, Quito
050	GMT-04:00	Atlantic Time (Canada)
055	GMT-04:00	Caracas, La Paz
056	GMT-04:00	Santiago
060	GMT-03:30	Newfoundland and Labrador
065	GMT-03:00	Brasilia
070	GMT-03:00	Buenos Aires, Georgetown
073	GMT-03:00	Greenland
080	GMT-01:00	Azores
083	GMT-01:00	Cape Verde Islands
085	GMT	Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
090	GMT	Casablanca, Monrovia
095	GMT+01:00	Belgrade, Bratislava, Budapest, Ljubljana, Prague
100	GMT+01:00	Sarajevo, Skopje, Warsaw, Zagreb
105	GMT+01:00	Brussels, Copenhagen, Madrid, Paris
110	GMT+01:00	Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
115	GMT+02:00	Bucharest
120	GMT+02:00	Cairo
125	GMT+02:00	Helsinki, Kiev, Riga, Sofia, Tallinn, Vilnius

Code	Fuseau horaire	Zone géographique
130	GMT+02:00	Athens, Istanbul, Minsk
135	GMT+02:00	Jerusalem
140	GMT+02:00	Harare, Pretoria
145	GMT+03:00	Moscow, St. Petersburg, Volgograd
150	GMT+03:00	Kuwait, Riyadh
155	GMT+03:00	Nairobi
158	GMT+03:00	Baghdad
160	GMT+03:30	Tehran
165	GMT+04:00	Abu Dhabi, Muscat
170	GMT+04:00	Baku, Tbilisi, Yerevan
175	GMT+04:30	Kabul
180	GMT+05:00	Ekaterinburg
185	GMT+05:00	Islamabad, Karachi, Tashkent
190	GMT+05:30	Chennai, Kolkata, Mumbai, New Delhi
193	GMT+05:45	Kathmandu
195	GMT+06:00	Astana, Dhaka
200	GMT+06:00	Sri Jayawardenepura
201	GMT+06:00	Almaty, Novosibirsk
203	GMT+06:30	Yangon Rangoon
205	GMT+07:00	Bangkok, Hanoi, Jakarta
207	GMT+07:00	Krasnoyarsk
210	GMT+08:00	Beijing, Chongqing, Hong Kong SAR, Urumqi
215	GMT+08:00	Kuala Lumpur, Singapore
220	GMT+08:00	Taipei
225	GMT+08:00	Perth
227	GMT+08:00	Irkutsk, Ulaanbaatar
227	GMT+08:00	Irkutsk, Ulaanbaatar
230	GMT+09:00	Seoul
235	GMT+09:00	Osaka, Sapporo, Tokyo
240	GMT+09:00	Yakutsk
245	GMT+09:30	Darwin

Annexe1 : Codes des fuseaux horaires utilisés par la fonction NTPSTART

Code	Fuseau horaire GMT	Zone géographique
250	GMT+09:30	Adelaide
255	GMT+10:00	Canberra, Melbourne, Sydney
260	GMT+10:00	Brisbane
265	GMT+10:00	Hobart
270	GMT+10:00	Vladivostok
275	GMT+10:00	Guam, Port Moresby
280	GMT+11:00	Magadan, Solomon Islands, New Caledonia
285	GMT+12:00	Fiji Islands, Kamchatka, Marshall Islands
290	GMT+12:00	Auckland, Wellington
300	GMT+13:00	Nuku'alofa

Code	Fuseau horaire	Zone géographique

Annexe 2 : DNP3.0 PROFIL & table d'implémentation

Ce document décrit les capacités de l'appareil, la valeur actuelle de chaque paramètre, ou les deux. Il est utilisé pour afficher les valeurs courantes, ou des paramètres Straton si ils sont configurables («NA» peut être indiqué pour les paramètres qui ne sont pas disponibles).
Les tests de conformité ont été réalisés avec le logiciel de Triangle Microworks Communication Protocol Test Harness Version 3.5.0.0 (DNP3 IED Certification Procedure Subset Level 2).

DEVICE PROFILE DOCUMENT

DNP V3.0 **DEVICE PROFILE DOCUMENT**	
Vendor Name: **Leroy Automation**	
Device Name: **LT200**	
Highest DNP Level Supported: For Requests: **Level 2** For Responses: **Level 2**	Device Function: ☐ Master ☒ **Slave**
Device manufacturer's software version string :	V1.3
Methods to set Configurable Parameters:	Software – Straton workbench
Connections Supported	☒ Serial ☒ IP Networking
Notable objects, functions, and/or qualifiers supported in addition to the Highest DNP Levels Supported (the complete list is described in the attached table): For static (non-change-event) object requests, request qualifier codes 07 and 08 (limited quantity), and 17 and 28 (index) are supported. Static object requests sent with qualifiers 07, or 08, will be responded with qualifiers 00 or 01. **Object 0 (device attributes) is supported: variations 211 to 252** **Object 3 (Double bit binary Input) is supported: variations 1, 2** **Object 4 (Double bit binary event) is supported: variations 1, 2, 3** **Object 23 (Frozen counter event) is supported: variations 1, 2, 5, 6** **16-bit, 32-bit and Floating Point Analog Change Events with Time may be requested.** **Floating Point Analog Output Status and Output Block Objects 40 and 41 are supported.** **Object 110 (Octet String Object) is supported.**	
Serial Connections	
Port name	com0, com1, com2 or com3
Serial Connection Parameters, Baud rate	Configurable (Straton 'Port settings')
IP Networking	
Type of End Point	TCP Listening
IP Address, Subnet Mask, Gateway IP Address	Configurable : Straton 'Registry Host'
TCP Listen Port Number	Configurable : Straton 'Port settings'
Multiple master connections(Outstations Only)	Supports 2 masters Method 2 (based on IP port number) used (Straton 'Port settings' & 'Second Port settings')
Time synchronization support	DNP3 Write Time (obj50, Var1)
Link Layer – Application Layer	
Maximum Data Link Frame Size (octets): Transmitted: **292** Received **292**	Maximum Application Fragment Size (octets): Transmitted: **2048** Received **2048**

DNP V3.0

DEVICE PROFILE DOCUMENT

Maximum Data Link Re-tries:	Maximum Application Layer Re-tries:
☐ None ☒ **Fixed : 3** ☐ Configurable from 0 to 65535	☒ **None** ☐ Configurable

Requires Data Link Layer Confirmation:

- ☒ **Never**
- ☐ Always
- ☐ Sometimes
- ☐ Configurable as: Never, Only for multi-frame messages, or Always

Requires Application Layer Confirmation:

- ☐ Never
- ☐ Always
- ☒ **When reporting Event Data (Slave devices only)**
- ☒ **When sending multi-fragment responses (Slave devices only)**
- ☐ Sometimes
- ☐ Configurable as: "Only when reporting event data", or "When reporting event data or multi-fragment messages."

Timeouts while waiting for:

Data Link Confirm:	☐	None	☒	**Fixed at 3**	☐	Variable	☐	
Complete Appl. Fragment:	☒	**None**	☐	Fixed at ____	☐	Variable	☐	Configurable
Application Confirm:	☐	None	☐	Fixed at ____	☐	**Variable**	☒	**Straton (appl.Conf. timeout).**
Complete Appl. Response:	☒	**None**	☐	Fixed at ____	☐	Variable	☐	Configurable

Others: Transmission Delay : **fixed 0**
Select/Operate Arm Timeout : **configurable Straton (output select timeout)**
Need Time Interval : **configurable Straton (clock valid period)**
Application File Timeout : **configurable Straton (file transfer timeout)**
Unsolicited Notification Delay : **configurable Straton (unsolicited events delay)**
Unsolicited Response Retry Delay : **configurable Straton (unsolicited retry delay)**
Unsolicited Offline Interval : **configurable Straton (unsolicited offline retry delay)**
Binary Change Event Scan Period : **configurable Straton (event scanning period)**
Double Bit Change Event Scan Period : **configurable Straton (event scanning period)**
Analog Change Event Scan Period : **configurable Straton (event scanning period)**
Counter Change Event Scan Period : **configurable Straton (event scanning period)**
Frozen Counter Change Event Scan Period : **configurable Straton (event scanning period)**
String Change Event Scan Period : **Never**
Virtual Terminal Event Scan Period : **Never**

DNP V3.0

DEVICE PROFILE DOCUMENT

Sends/Executes Control Operations:

WRITE Binary Outputs	☒ **Never**	☐ Always	☐ Sometimes	☐ Configurable
SELECT/OPERATE	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
DIRECT OPERATE	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
DIRECT OPERATE – NO ACK	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
Count > 1	☒ **Never**	☐ Always	☐ Sometimes	☐ Configurable
Pulse On	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
Pulse Off	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
Latch On	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
Latch Off	☐ Never	☒ **Always**	☐ Sometimes	☐ Configurable
Queue	☒ **Never**	☐ Always	☐ Sometimes	☐ Configurable
Clear Queue	☒ **Never**	☐ Always	☐ Sometimes	☐ Configurable

Attach explanation if 'Sometimes' or 'Configurable' was checked for any operation.

Reports Binary Input Change Events when no specific variation requested:	Reports time-tagged Binary Input Change Events when no specific variation requested:
☐ Never ☐ Only time-tagged ☐ Only non-time-tagged ☒ **Configurable Straton (default variation)**	☐ Never ☐ Binary Input Change With Time ☐ Binary Input Change With Relative Time ☒ **Configurable Straton (default variation)**

Outstation Unsolicited Response Support

Sends Unsolicited Responses:	Sends Static Data in Unsolicited Responses:
☐ Never ☒ **Configurable Straton (unsolicited allowed)** ☐ Only certain objects ☐ Sometimes (attach explanation) ☒ **ENABLE/DISABLE UNSOLICITED Function codes supported**	☒ **Never** ☐ When Device Restarts ☐ When Status Flags Change No other options are permitted.
Default Counter Object/Variation:	Counters Roll Over at:
☐ No Counters Reported ☒ **Configurable Straton (default variation)** ☐ Default Object Default Variation: ☐ Point-by-point list attached	☐ No Counters Reported ☐ Configurable (attach explanation) ☐ 16 Bits ☒ **32 Bits** ☐ Other Value: _____ ☐ Point-by-point list attached

Sends Multi-Fragment Responses:

☒ **Yes**

☐ No

☐ Configurable

DNP définition : table d'implémentation

Le tableau suivant présente la mise en œuvre des groupes d'objets et de variations qui, codes de fonction et les codes qualifieurs que le LT supporte dans les demandes et les réponses.

La colonne « *Request* » identifie toutes les demandes qui peuvent être envoyées par un Maître, ou toutes les demandes qui doivent être analysées par le LT.

La colonne « *Response* » identifie toutes les réponses qui doivent être analysées par un Maître, ou toutes les réponses qui peuvent être envoyées par le LT.

Pour des objets statiques (non-change-event), les requêtes envoyées avec les codes qualifiers 00, 01, 06, 07, or 08, les réponses seront avec les qualifiers 00 or 01. Les requêtes envoyées avec les qualifiers 17 ou 28, les réponses seront avec les qualifiers 17 ou 28. Pour les objets de type « change-event », les qualifiers 17 ou 28 sont toujours utilisés dans la réponse.

DNP OBJECT GROUP & VARIATION			**REQUEST** Master may issue Outstation must parse		**RESPONSE** Master must parse Outstation may issue	
Group Num	**Var Num**	**Description**	**Func Codes (dec)**	**Qual Codes (hex)**	**Func Codes (dec)**	**Qual Codes (hex)**
0	1-253	Device Attribute Specific	1	00,01,06,07,08,17,27,28	129	00,01,17,27,28
			2	00,01		
0	254	Device Attribute – Non Specific All Attributes Request	1	00,01,06, 07,08, 17,27,28	129	00, 01 17,28
0	255	Device Attribute – List of Attribute Variations	1	00,01,06,07,08,17,27,28	129	00,01,17,28
1	0	Binary Input – Any Variation	1, 22	00,01,06		
1	1 (default see note1)	Binary Input – Packed format	1	00,01,06	129	00,01,17,27,28
1	2	Binary Input – With flags	1	00,01,06	129	00,01,17,27,28
2	0	Binary Input Event – Any Variation	1	06,07,08		
2	1	Binary Input Event – Without time	1	06,07,08	129 130	17, 28
2	2	Binary Input Event – With absolute time	1	06,07,08	129 130	17, 28
2	3 (default see note1)	Binary Input Event – With relative time	1	06,07,08	129 130	17, 28
3	0	Double bit Input – Any Variation	1, 22	00,01,06		
3	1 (default see note1)	Double bit Input – Packed format	1	00,01,06	129	00,01,17,27,28
3	2	Double bit Input – With flags	1	00,01,06	129	00,01,17,27,28
4	0	Double bit Input Event – Any Variation	1	06,07,08		
4	1	Double bit Input Event – Without time	1	06,07,08	129 130	17, 28
4	2	Double bit Input Event – With absolute time	1	06,07,08	129 130	17, 28
4	3 (default see note1)	Double bit Input Event – With relative time	1	06,07,08	129 130	17, 28
10	0	Binary Output – Any Variation	1,22	00,01,06,07,08,17,27,28		
10	1	Binary Output	1 (read)	00,01,06,07,08,17,27,28	129	00,01,17,28
			1 (write)	00,01		
10	2 (default see note1)	Binary Output – Output status with flags	1	00,01,06	129	00,01
12	1	Control Relay Output Block	3,4,5,6	17,28	129	Echo of request
12	2 (default see note1)	Pattern Control Block	3,4,5,6	00,01	129	Echo of request
20	0	Counter – Any Variation	1,22	00,01,06, 07,08,17,27,28		

DNP OBJECT GROUP & VARIATION			**REQUEST** Master may issue Outstation must parse		**RESPONSE** Master must parse Outstation may issue	
Group Num	**Var Num**	**Description**	**Func Codes (dec)**	**Qual Codes (hex)**	**Func Codes (dec)**	**Qual Codes (hex)**
			7,8,9,10	00,01,06,07,08		
20	1	Counter – 32 bit with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
20	2	Counter – 16 bit with flag	1	00,01,06,07	129	00,01,17,28
20	5 (default see note1)	Counter – 32 bit without flag	1	00,01,06	129	00,01,17,28
20	6	Counter – 16 bit without flag	1	00,01,06	129	00,01,17,28
21	0	Frozen Counter – Any Variation	1,22	00,01,06		
21	1	Frozen Counter – 32 bit with flag	1	00,01,06	129	00,01,17,28
21	2	Frozen Counter – 16 bit with flag	1	00,01,06	129	00,01,17,28
21	5	Frozen Counter – 32 bit with flag and time	1	00,01,06	129	00,01,17,28
21	6	Frozen Counter – 16 bit with flag and time	1	00,01,06	129	00,01,17,28
21	9 (default see note1)	Frozen Counter – 32 bit without flag	1	00,01,06	129	00,01,17,28
21	10	Frozen Counter – 16 bit without flag	1	00,01,06	129	00,01,17,28
22	0	Counter Event – Any Variation	1			
22	1 (default see note1)	Counter Event – 32 bit with flag	1	06,07,08	129,130	17, 28
22	2	Counter Event – 16 bit with flag	1	06,07,08	129,130	17, 28
22	5	Counter Event – 32 bit with flag and time	1	06,07,08	129,130	17, 28
22	6	Counter Event – 16 bit with flag and time	1	06,07,08	129,130	17, 28
23	0	Frozen Counter Event – Any Variation	1	06,07,08		
23	1 (default see note1)	Frozen Counter Event – 32 bit with flag	1	06,07,08	129,130	17, 28
23	2	Frozen Counter Event – 16 bit with flag	1	06,07,08	129,130	17, 28
23	5	Frozen Counter Event – 32 bit with flag and time	1	06,07,08	129,130	17, 28
23	6	Frozen Counter Event – 16 bit with flag and time	1	06,07,08	129,130	17, 28
30	0	Analog Input – Any Variation	1,22	00,01,06,07,08,17,27,28		
30	1	Analog Input – 32 bit with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
30	2	Analog Input – 16 bit with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
30	3 (default see note1)	Analog Input – 32 bit without flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
30	4	Analog Input – 16 bit without flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
30	5	Analog Input – Single prec flt pt with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
32	0	Analog Input Event – Any Variation	1	06,07,08		
32	1 (default see note1)	Analog Input Event – 32 bit Without time	1	06,07,08	129,130	17, 28
32	2	Analog Input Event – 16 bit Without time	1	06,07,08	129,130	17, 28
32	3	Analog Input Event – 32 bit With time	1	06,07,08	129,130	17, 28
32	4	Analog Input Event – 16 bit With time	1	06,07,08	129,130	17, 28
32	5	Analog Input Event – Single prec flt pt without time	1	06,07,08	129,130	17, 28
32	7	Analog Input Event – Single prec flt pt with time	1	06,07,08	129,130	17, 28
40	0	Analog Output Status – Any Variation	1	00,01,06,07,08,17,27,28		
40	1 (default see note1)	Analog Output Status – 32 bit with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
40	2	Analog Output Status – 16 bit with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
40	3	Analog Output Status – Single prec flt pt with flag	1	00,01,06,07,08,17,27,28	129	00,01,17,28
41	0	Analog output	22	00,01,06,07,08,17,27,28		
41	1	32 bit Analog output	3,4,5,6	17,27,28	129	Echo of request

DNP OBJECT GROUP & VARIATION			**REQUEST** Master may issue Outstation must parse		**RESPONSE** Master must parse Outstation may issue	
Group Num	**Var Num**	**Description**	**Func Codes (dec)**	**Qual Codes (hex)**	**Func Codes (dec)**	**Qual Codes (hex)**
41	2	16 bit Analog output	3,4,5,6	17,27,28	129	Echo of request
41	3	Short floating point Analog output	3,4,5,6	17,27,28	129	Echo of request
50	1	Time and Date – Absolute time	1	07	129	07
			2	07		
51	1	Time and Date CTO – Absolute time, synchronized			129,130	07
51	2	Time and Date CTO – Absolute time, unsynchronized			129,130	07
52	1	Time Delay – Coarse			129	07
52	2	Time Delay – Fine			129	07
60	1	Class Objects – Class 0 data	1	06		
60	2	Class Objects – Class 1 data	1	06,07,08		
			20,21,22	06		
60	3	Class Objects – Class 2 data	1	06,07,08		
			20,21,22	06		
60	4	Class Objects – Class 3 data	1	06,07,08		
			20,21,22	06		
80	1	Internal Indications – Packed format	2 (default see note3)	00 (index =4 or 7)		
110	String length	Octet String Object	1 (read), 22	00,01,06,07,08,17,27,28	129	00,01
			2 (write)	00,01,07,08,17,27,28		
No object (function code only)			13 (cold restart)			
No object (function code only)			23 (delay measure)			

Note 1: Par défaut la variation se réfère à la variation retournée lorsque la variation 0 est demandée et / ou dans la classe 0, 1, 2, ou 3 demandée. Les variations par défaut sont configurables ; les paramètres par défaut pour la configuration paramètrable sont indiqués dans la table ci-dessus.

Note 2: Pour les objets statiques (non-change-event), uniquement les codes 17 ou 28 sont retournés quand une requête est envoyée avec les codes 17 ou 28. Sinon, les requêtes d'objets statiques envoyées avec les codes 00, 01, 06, 07, ou 08, auront des réponses avec les codes 00 ou 01. (Pour les objets « change-event », les codes 17 ou 28 sont toujours retournés)

Note 3: L'écriture d'indication interne est supportée uniquement pour les index 4 ou 7 (Need Time IIN1-4 ou Restart IIN1-7).