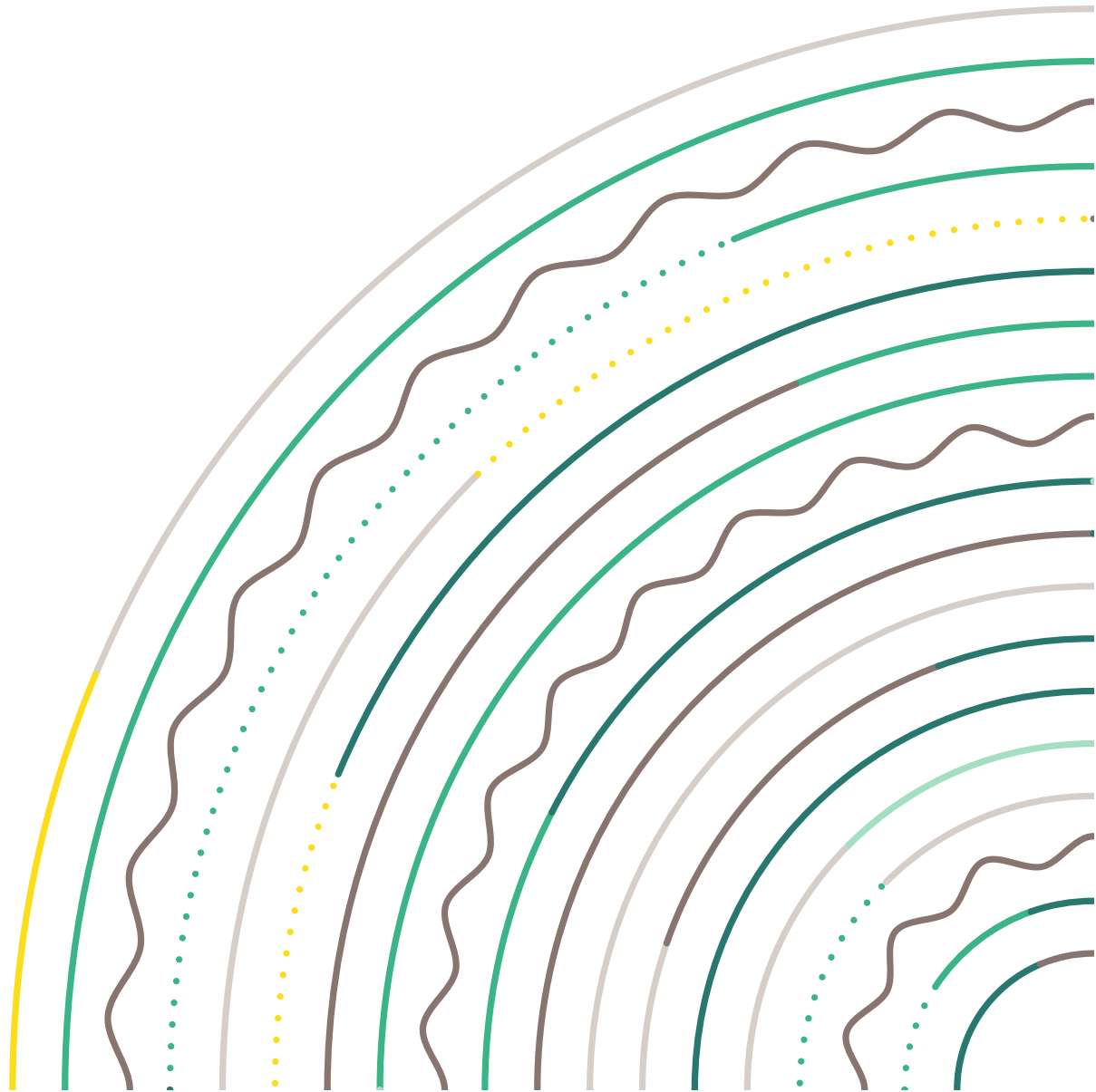


Lifedomus

Modbus

06/06/2018

Version 1.5



Le protocole Modbus TCP, mode « Maître »

Table des matières

1	Pré-requis.....	3
2	Connecteur Modbus	3
2.1	Ajout d'un connecteur Modbus TCP	3
2.2	Configuration d'un connecteur Modbus TCP.....	3
3	Équipement Modbus	4
3.1	Propriétés d'un équipement.....	4
3.1.1	Configuration du champ « lecture » d'une propriété	4
3.1.2	Configuration du champ « écriture » d'une propriété	4
3.2	Variables d'un équipement.....	5
3.2.1	Gestion des variables	5
3.2.2	Configuration d'une variable	6
3.3	Commandes d'un équipement.....	8
4	Création de l'interface « Design Studio ».....	9
5	Configuration des « Datablocs ».....	10
5.1	Accès à la configuration des datablocs	10
5.2	Création d'un databloc	10

1 Pré-requis

Afin de suivre correctement cette documentation, quelques pré-requis sont nécessaires :

- ✓ Ajout, configuration et utilisation d'un connecteur dans le « Config Studio »
- ✓ Création d'un équipement générique Lifedomus
- ✓ Le protocole Modbus
- ✓ Des bases sur les opérations logiques binaires pour comprendre et utiliser le « masque »

2 Connecteur Modbus

La communication entre Lifedomus et un bus Modbus nécessite l'utilisation d'une passerelle Modbus TCP/IP.

Comme tous les autres protocoles, la communication se fait par l'intermédiaire d'un connecteur Lifedomus. Actuellement, Lifedomus se comporte en tant que « maître » sur le bus. Un mode « esclave » sera implémenté ultérieurement.

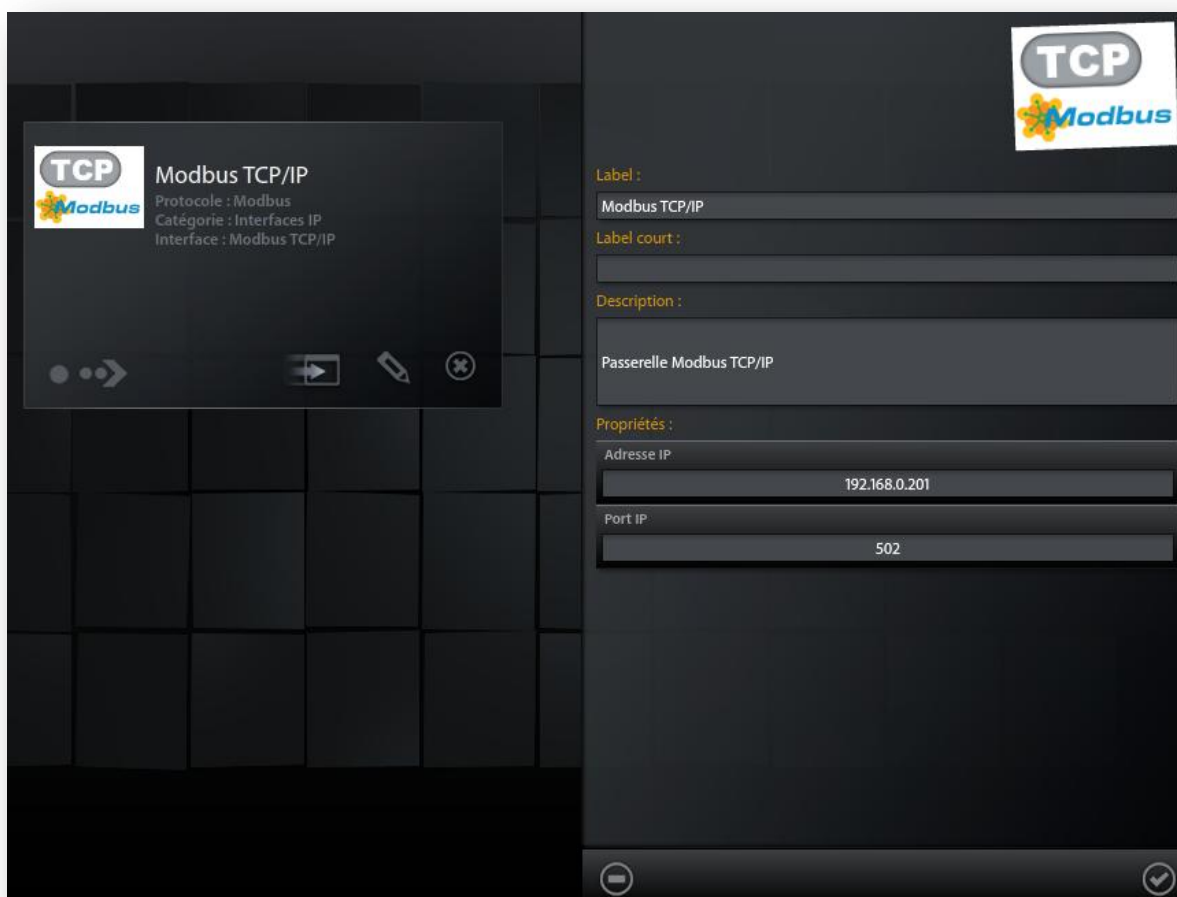
2.1 Ajout d'un connecteur Modbus TCP

L'ajout d'un connecteur « Modbus TCP/IP » se fait de la même façon que pour tout autre connecteur.

2.2 Configuration d'un connecteur Modbus TCP

Les éléments de configuration d'un connecteur Modbus TCP sont les suivants :


- Adresse IP : L'adresse IP de la passerelle Modbus TCP
- Port IP : Le port de la passerelle Modbus TCP (par défaut : 502)

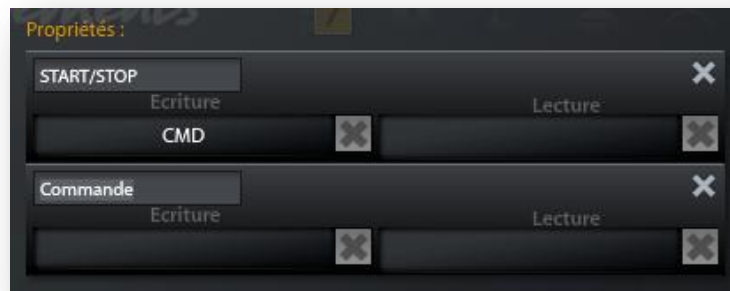


3 Equipement Modbus

Un connecteur Modbus fonctionne exclusivement avec des équipements Lifedomus de type « **Universel** ». Il faut donc ajouter un « **Equipement (Générique)** » et le lier au connecteur Modbus précédemment créé.

3.1 Propriétés d'un équipement

Pour ajouter une propriété à l'équipement nouvellement créé, utilisez l'icône  en bas à droite. Une propriété va permettre la lecture et l'écriture de données dans un ou plusieurs registres de l'esclave Modbus.



3.1.1 Configuration du champ « lecture » d'une propriété

Lors d'un clic sur le champ « lecture » d'une propriété, une nouvelle fenêtre s'affiche et vous donne accès aux différentes variables créées. Il vous est possible de créer de nouvelles variables.

Sélectionner une **variable** et valider la fenêtre permet de lier le champ de lecture de la propriété à cette variable. Un retour d'état correspondant à la valeur de cette variable devient alors accessible dans Lifedomus.

Le fonctionnement et les détails de configuration d'une variable sont décrits dans la partie [Variables d'un équipement](#) de ce document

3.1.2 Configuration du champ « écriture » d'une propriété

Lors d'un clic sur le champ « écriture » d'une propriété, une nouvelle fenêtre s'affiche et vous donne accès aux différentes variables et commandes créées.

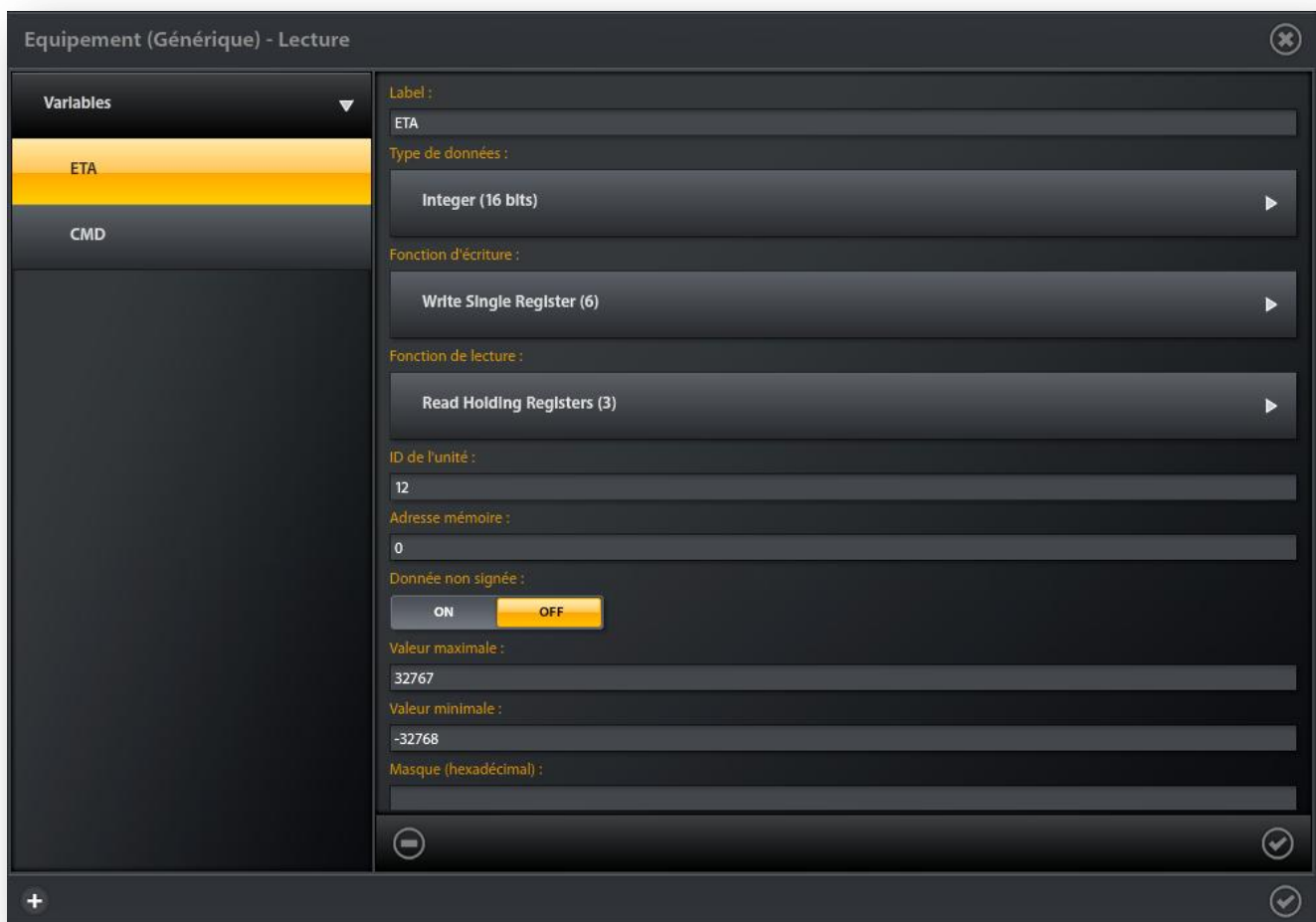
Sélectionner une **variable** et valider la fenêtre permet de lier le champ d'écriture de la propriété à cette variable. Une action permettant l'écriture dans la variable devient alors accessible dans Lifedomus.

Sélectionner une **commande** et valider la fenêtre permet de lier le champ d'écriture de la propriété à cette commande. Une action permettant l'exécution de cette commande devient alors accessible dans Lifedomus.

Le fonctionnement et les détails de configuration d'une commande sont décrits dans la partie [Commandes d'un équipement](#) de ce document


3.2 Variables d'un équipement


Une variable Modbus est une donnée dans Lifedomus représentant une zone mémoire (un ou plusieurs registres) d'un esclave du bus.



L'écran de gestion des variables ci-dessus permet d'ajouter, configurer et supprimer des variables Modbus. Ces variables créées sont spécifiques à l'équipement actuellement en cours de modification.

3.2.1 Gestion des variables

L'ajout d'une variable s'effectue en sélectionnant la liste « Variables » et en cliquant sur  (en bas à gauche).

Le bouton  permet l'enregistrement des modifications apportées à la variable et son association avec la propriété (en lecture ou en écriture) précédemment sélectionnée.

Le bouton  permet la suppression de la variable sélectionnée.

3.2.2 Configuration d'une variable

La configuration d'une variable s'effectue en sélectionnant celle-ci dans la liste des variables. Les éléments de configuration permettent de définir précisément la façon dont seront lues et interprétées les zones mémoires.

- **Label** : Le label permet simplement d'attribuer un titre à la variable.
- **Type de données** : Le type de données que vous souhaitez manipuler dans Lifedomus : booléen, nombre entier, nombre à virgule (IEEE 754) ou valeur hexadécimale. Les valeurs de 16 bits ou moins sont lues ou écrites sur un seul registre, tandis que les valeurs d'un type de 32 bits seront lues ou écrites sur deux registres consécutifs.
- La **fonction de lecture** et la **fonction d'écriture** sont les fonctions Modbus qui seront utilisées pour lire et écrire les zones mémoire correspondantes à cette variable sur le bus. Ces fonctions sont généralement précisées dans la documentation du constructeur.
- **ID de l'unité** ou « UnitID » (1 à 255) : Ce champ est spécifique au protocole Modbus TCP et permet d'identifier un esclave sur le bus.
- **Adresse mémoire** (1 à 65535): L'adresse mémoire est l'adresse du registre que vous souhaitez lire ou écrire. Lorsque qu'un type de données de 32 bits est sélectionné, les registres lus ou écrits sont le registre indiqué dans le champ « adresse mémoire » et le registre suivant.
Exemple : un Integer (32bits) lu ou écrit sur l'adresse mémoire 2202 sera en fait lue ou écrite sur les adresses 2202 et 2203.
- **Donnée non signée** : L'activation de ce champ permet de considérer la donnée comme une donnée non signée, c'est-à-dire une valeur positive. Changer cette valeur met à jour les champs « Valeur minimale » et « Valeur maximale ».
- **Valeur minimale et maximale** : Ces valeurs sont définies par défaut en fonction du « **type de données** » choisi et de l'activation ou désactivation des « **données signées** ». Exemples :
 - Un Booléen a pour valeur 0 (faux) ou 1 (vrai)
 - Un Entier (16 bits) signé a une valeur comprise entre -32768 et 32767
 - Un Entier (16 bits) non-signé a une valeur comprise entre 0 et 65535

Les valeurs minimale et maximale définies par défaut sont les limites des valeurs lues ou écrites dans Lifedomus. S'il n'est pas possible de dépasser ces limites, il est tout à fait possible de les restreindre.

- **Masque (hexadécimal)** : Ce champ permet d'attribuer un masque au(x) registre(s). Il doit être défini sous forme hexadécimale. Exemple : FFF7, 0xFFF7 ou 16#FFF7
 - **En cas de lecture** : Afin de déterminer la valeur finale disponible dans Lifedomus, un « ET » logique est effectué entre le masque et la valeur lue sur le bus.
 - **En cas d'écriture** : Seuls les bits présents à « 1 » dans le masque seront changés dans le(s) registre(s) cible(s).

L'utilisation du masque ajoute une grande puissance dans la manipulation des données.

Des exemples pratiques sont disponibles dans la partie [Exemples et particularités du masque](#) de ce document.

- **Inverser les octets** : Il est généralement précisé dans la documentation du constructeur si ce paramètre doit être utilisé. Il permet d'inverser l'octet de poids fort et l'octet de poids faible lors de la lecture et l'écriture du registre. Un registre compte 16 bits, soit 2 octets. Si un registre a pour valeur 0x7F00, inverser les octets permet de lire : 0x007F.
- **Inverser les mots** : Un mot compte 16 bits (soit la taille d'un registre). L'inversion des mots est utile lors de la lecture et l'écriture de plusieurs registres, lorsque le type de données choisi est Integer (32 bits) ou Float (32 bits).
Exemple : Les registres n°2202 et n°2203 sont utilisés pour lire ou écrire un entier de 32 bits. Sans inversion des mots, le registre n°2202 contient les bits de poids fort, ainsi que le bit de signe (si la valeur est signée) tandis que le registre n°2203 contient les bits de poids faible. Avec inversion des mots, le registre n°2202 contient alors les bits de poids faible et le registre n°2203 les bits de poids fort et le bit de signe.
- **Coefficient** : Ce champ est un coefficient multiplicateur appliqué à la valeur lue sur le bus (divisée lors de l'écriture).
Lecture : « Valeur Lifedomus » = « Valeur bus » x « Coefficient »
Ecriture : « Valeur bus » = « Valeur Lifedomus » / « Coefficient »
- **L'unité** : Ce champ facultatif permet de sélectionner l'unité de la valeur dans Lifedomus.

Exemples et particularités du masque

Le masque permet de ne prendre en compte que certains bits du registre dans la lecture et l'écriture sur le bus. Il peut être utilisé pour des données de type Entier (16 ou 32 bits), Hexadécimal ou Booléen.

Les valeurs des exemples suivants sont présentées sous forme hexadécimale et binaire afin de faciliter la compréhension de l'utilisation du masque lors de la lecture et de l'écriture sur le bus.

3.2.2.1 Utilisation du masque avec le type « entier » ou « hexadécimal » de Lifedomus

Exemple :

- Le registre #3303 contient la valeur suivante : **0x12FF** (0001 0010 1111 1111).
- Le masque est : **0x007F** (0000 0000 0111 1111).
- Une lecture de la variable donnera pour valeur de retour d'état « **masque** » ET « **valeur sur le bus** » : **0x007F** (0000 0000 0111 1111) ET **0x12FF** (0001 0010 1111 1111) = **0x007F** (0000 0000 0111 1111)
- L'écriture de la valeur **0x0000** (0000 0000 0000 0000) sur la variable produira l'écriture de la valeur : **0x1280** (0001 0010 1000 0000) sur le bus :
- Une nouvelle lecture de la variable donnera pour valeur de retour d'état : **0x007F** (0000 0000 0111 1111) ET **0x1280** (0001 0010 1000 0000) = **0x0000** (0000 0000 0000 0000)
- L'écriture de la valeur **0x0F22** (0000 1111 0010 0010) sur la variable produira l'écriture de : **0x12A2** (0001 0010 1010 0010) sur le bus.

3.2.2.2 Utilisation du masque avec le type « booléen » de Lifedomus

L'utilisation du masque avec une valeur de type booléen peut être fréquente en Modbus. En effet, plusieurs valeurs booléennes, « VRAI/FAUX », « Activé/Désactivé », ... sont souvent enregistrées dans un seul registre. Chaque bit représente alors une valeur booléenne (« VRAI » si le ou les bits en question sont à « 1 », « FAUX » sinon).

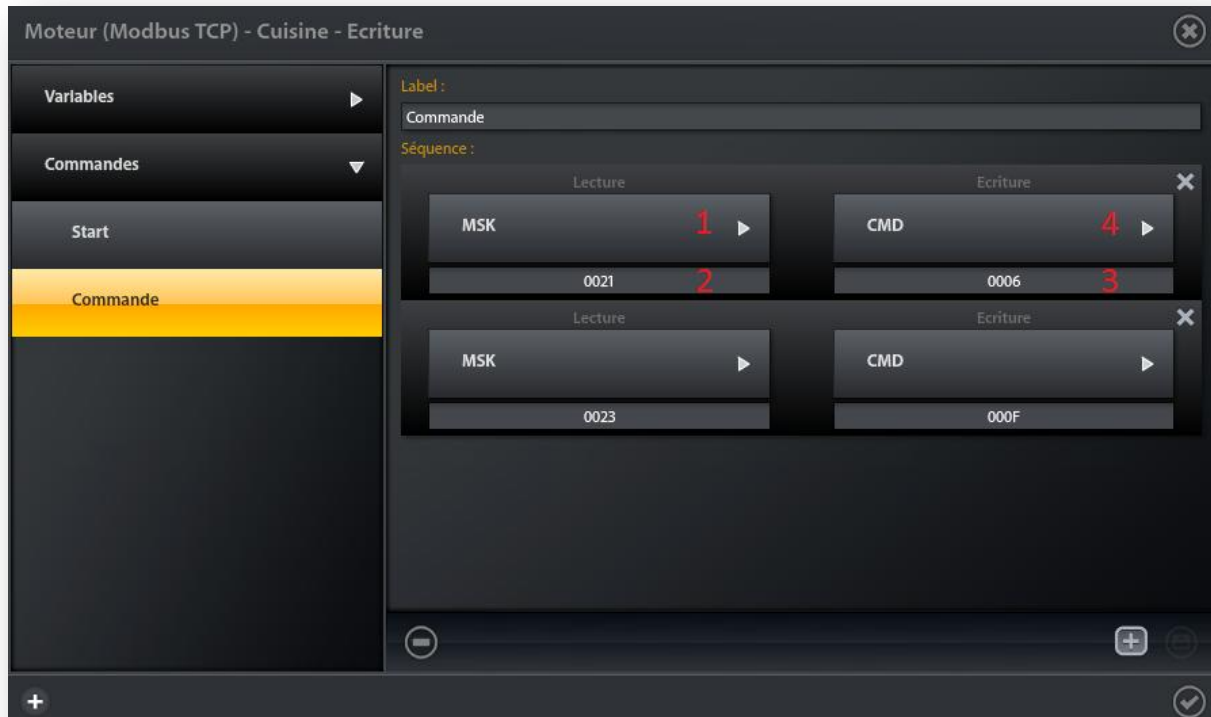
Exemple :

- Le registre #3303 contient la valeur suivante : **0x12FF** (0001 0010 1111 1111).
- Le masque est : **0x0080** (0000 0000 1000 0000)
- **0x0080** (0000 0000 1000 0000) ET **0x12FF** (0001 0010 1111 1111) = **0x0080** (0000 0000 1000 0000)
Le résultat étant égal au masque, une lecture de la variable donnera pour valeur de retour d'état : **VRAI**
- L'écriture de la valeur **FAUX** écrira sur le bus : « **inverse du masque** » ET « **valeur du bus** » : **0x0F22** (1111 1111 01 1111 1111) ET **0x12FF** (0001 0010 1111 1111) = **0x127F** (0001 0010 0111 1111)
- **0x0080** (0000 0000 1000 0000) ET **0x127F** (0001 0010 0111 1111) = **0x0080** (0000 0000 0000 0000)
Le résultat étant différent du masque, une lecture de la variable donnera pour valeur : **FAUX**
- L'écriture de la valeur **VRAI** écrira sur le bus : « **masque** » ET « **valeur sur le bus** » : **0x0080** (0000 0000 1000 0000) ET **0x127F** (0001 0010 0111 1111) = **0x12FF** (0001 0010 1111 1111)

3.3 Commandes d'un équipement

L'écran de gestion des commandes permet d'ajouter, configurer et supprimer des commandes Modbus.

Tout comme les variables, ces commandes créées sont spécifiques à l'équipement actuellement en cours de modification.



Chaque élément de la séquence d'une commande est exécuté dans l'ordre saisi et fonctionne de la manière suivante : Si la variable de gauche (1) est égale à la valeur indiquée à gauche (2), alors la valeur indiquée à droite (3) est écrite dans la variable de droite (4). Il faut noter que si le champ de lecture (2) reste vide, la condition sera validée et l'écriture exécutée.

Exemple : Ces commandes sont utilisées dans certains cas comme celui du démarrage d'un moteur : Si le moteur est hors tension, on le met sous tension, s'il est arrêté on le démarre, puis on définit sa vitesse.

4 Création de l'interface « Design Studio »

Les équipements Modbus étant dans Lifedomus des équipements génériques, il n'y a pas de widget par défaut dans le Design Studio. Il est donc nécessaire d'utiliser le mode édition pour construire l'interface de supervision.

Voir les documentations concernant le « What I See » et le « What I Do » du « Design Studio »

5 Configuration des « Datablocs »

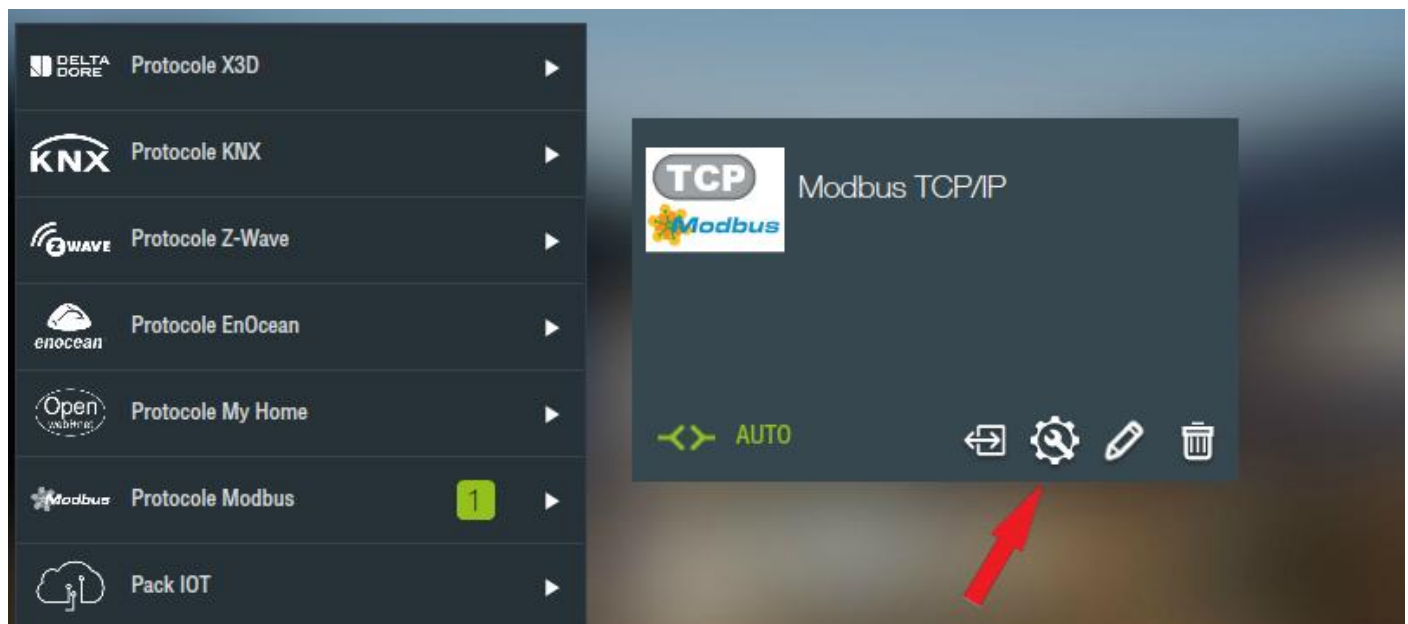
La configuration de datablocs permet d'optimiser de manière considérable la lecture de plusieurs registres consécutifs.

Ainsi quand (par exemple) les « Holding registers » n°0, 1, 2, 10 à 19, 30 à 49, 60 et 61 sont utilisés en lecture dans Lifedomus, de nombreuses requêtes de lectures sont effectuées sur le bus (35 dans cet exemple).

Il est possible de configurer un databloc sur des données de type « Holding registers », ayant pour adresse de départ « 0 » et de longueur « 62 ». De cette façon l'ensemble des registres utilisés seront récupérés en une unique requête de lecture sur le bus.

5.1 Accès à la configuration des datablocs

Un databloc est global à un connecteur Modbus. La création se fait via la vignette du connecteur.



5.2 Création d'un databloc

Les champs d'un databloc sont les suivants :

Nom du bloc	
Type d'objet	Coil / Discrete / Input Register / Holding Register.
Unit ID	Nombre compris entre 0 et 255.
Adresse de début	Adresse comprise entre 0 et 65535.
Nombre de blocs	Nombre de blocs contenu dans ce databloc. Pour les Coils et Discrete, le nombre max est de 104. Pour les Input et Holding Registers, le nombre max est de 125
Ecriture groupée ?	Non applicable aux objets en lecture seule (Discrete et Input Register).
Pooling (s)	Durée en secondes entre deux lectures de ce databloc.
Phasing (s)	Décalage de la lecture au démarrage du connecteur.
Activer ?	Permet de désactiver la lecture de ce databloc.

Attention : Les requêtes de lecture des datablocs sont prioritaires par rapport aux requêtes de lecture des registres non présents dans un databloc. Il est donc fortement recommandé de configurer des datablocs pour l'ensemble des lectures effectuées par Lifedomus.