

TP N°4
COMMUNICATION MODBUS
(SIEMENS TIA PORTAL V13)

I. OBJECTIF

Sur l'environnement SIEMENS TIA PORTAL V13 réaliser :

- Communication MODBUS TCP
 - MB_CLIENT (Modbus TCP)
 - MB_SERVER (Modbus TCP)
- Communication MODBUS RTU
 - MB_COMM_LOAD
 - MB_MASTER
 - MB_SLAVE

II. MATERIEL

1 Micro-ordinateur avec conditions requises minimales :
- <i>Processeur => Intel® Celeron® Dual Core 2,2 GHz (Ivy/Sandy Bridge)</i>
- <i>RAM => 2 GB</i>
- <i>Espace disque disponible => 20 GB</i>
- <i>Systèmes d'exploitation => Windows 7 (64 bits)</i>
- <i>Résolution de l'écran => 1024 x 768</i>
2 Automates programmables S7-1200_ CPU 1215C DC/DC/DC
2 câbles réseau Ethernet avec connecteurs rj45 croisés
2 modules de communication CM 1241 (RS485) ou 2 COMMUNICATION BOARD CB 1241, RS485
1 câble RS-485 pour Modbus RTU

III. MODALITÉ DE FONCTIONNEMENT

Une CPU fonctionnant en tant que maître Modbus RTU (ou client Modbus TCP) peut lire/écrire à la fois des données et des états d'E/S dans un esclave Modbus RTU éloigné (ou un serveur Modbus TCP). Les données éloignées peuvent être lues et traitées dans le programme utilisateur.

Une CPU fonctionnant en tant qu'esclave Modbus RTU (ou serveur Modbus TCP) permet à un appareil superviseur de lire/écrire à la fois des données et des états d'E/S dans une CPU

éloignée. L'appareil superviseur peut écrire dans la mémoire CPU éloignée de nouvelles valeurs qui peuvent être traitées dans le programme utilisateur.

1. Communication MODBUS TCP

Modbus TCP utilise des liaisons client-serveur comme voie de communication Modbus. Plusieurs liaisons client-serveur peuvent exister en plus de la liaison entre STEP 7 et la CPU. Les liaisons mixtes client et serveur sont prises en charge jusqu'au nombre maximum de liaisons autorisées par le modèle de CPU. Chaque liaison MB_SERVER doit utiliser un DB d'instance et un numéro de port IP uniques. Une seule liaison est acceptée par port IP. Chaque instruction MB_SERVER (avec son DB d'instance et son port IP uniques) doit être exécutée individuellement pour chaque liaison.

1.1. MB_CLIENT (Modbus TCP)

Établir une liaison TCP client-serveur, envoyer un message de commande, recevoir la réponse et gérer la déconnexion du serveur

On commence par créer un sous-réseau Profinet PN/IE_1 entre 2 CPU 1215C (fig.1)

Le PLC1 sera configuré comme CLIENT tandis que le PLC2 sera configuré comme SERVER.

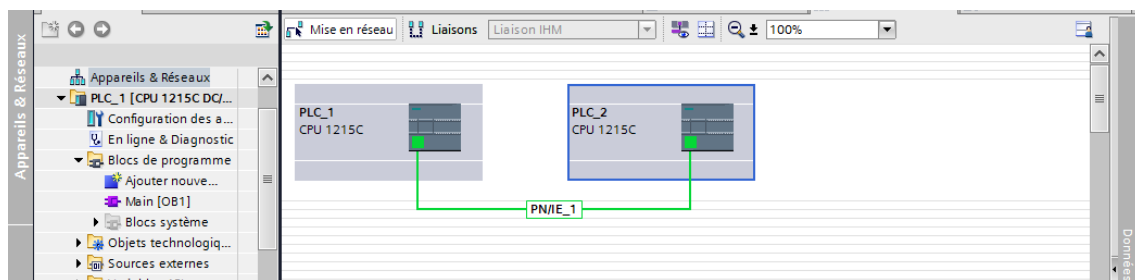


Figure 1_création d'un sous réseau PN/IE_1

On insère, ensuite, un bloc de données DB MB_CLIENT (fig.2)

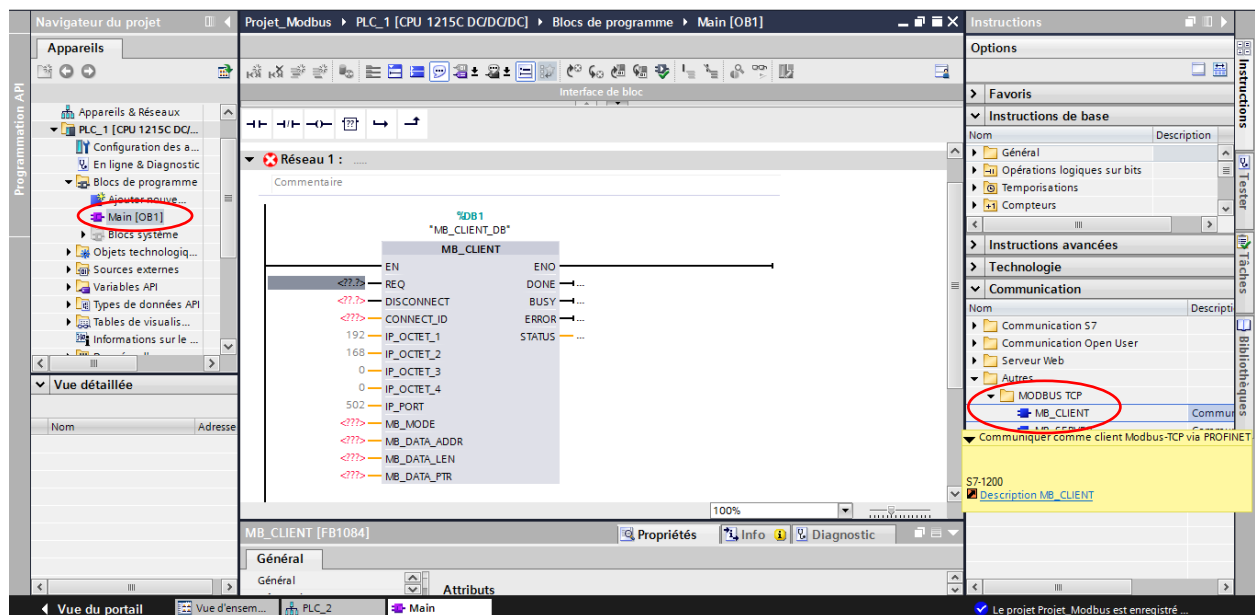


Figure 2_DB MB_CLIENT

Paramètres

Paramètre	Déclaration	Type de données	Description
REQ	Input	BOOL	<p>Requête de communication avec le serveur Modbus TCP</p> <p>Le paramètre REQ est commandé par niveau. Cela signifie que tant que l'entrée est à 1 (REQ=true), l'instruction envoie des requêtes de communication.</p> <ul style="list-style-type: none"> • La requête de communication verrouille l'accès au DB d'instance pour les autres clients. • Les modifications aux paramètres d'entrée ne s'appliquent qu'à partir du moment où il y a une réponse du serveur ou un message d'erreur a été émis. <p>Si le paramètre REQ est mis à nouveau à "1" pendant une requête Modbus en cours, aucune autre transmission ne sera exécutée directement après.</p>
DISCONNECT	Input	BOOL	<p>Le paramètre vous permet de commander l'établissement et la coupure de la liaison au serveur Modbus :</p> <ul style="list-style-type: none"> • 0: Établissement de la communication à l'adresse IP et au numéro de port indiqués. • 1: Suspendre la connexion de communication. Aucune autre fonction n'est exécutée durant la coupure de la liaison. Après avoir réalisé la coupure de la liaison avec succès, le paramètre STATUS affiche la valeur 7003. <p>Si le paramètre REQ est mis à "1" lors de l'établissement de la liaison, la requête est émise immédiatement.</p>
CONNECT_ID	Input	UINT	ID univoque pour l'identification de la connexion. A chaque instance des instructions "MB_CLIENT" et " MB_SERVER " doit être assignée une ID de liaison univoque.
IP_OCTET_1	Input	USINT	1. Octet de l'adresse IP* du serveur Modbus TCP.
IP_OCTET_2	Input	USINT	2. Octet de l'adresse IP* du serveur Modbus TCP.
IP_OCTET_3	Input	USINT	3. Octet de l'adresse IP* du serveur Modbus TCP.
IP_OCTET_4	Input	USINT	4. Octet de l'adresse IP* du serveur Modbus TCP.
IP_PORT	Input	UINT	Numéro de port IP du serveur vers lequel le client établit la liaison et avec lequel il communique au moyen du protocole TCP/IP (valeur par défaut : 502).
MB_MODE	Input	USINT	Sélection du mode de la requête (lire, écrire ou diagnostic).
MB_DATA_ADDR	Input	UDINT	Adresse de début des données auxquelles l'instruction "MB_CLIENT" a accès.
DATA_LEN	Input	UINT	Longueur des données : Nombre de bits ou de mots pour l'accès aux données (voir "Paramètre MB_MODE et MB_DATA_ADDR" - longueur des données).
MB_DATA_PTR	InOut	VARIANT	<p>Pointeur indiquant le registre de données Modbus : le registre est un tampon pour les données reçues du serveur Modbus ou à envoyer à ce serveur. Le pointeur doit faire référence à un bloc de données global avec accès standard.</p> <p>Le nombre de bits adressés doit être divisible par 8.</p>
DONE	Out	BOOL	Le bit au paramètre de sortie DONE est mis à "1" dès que la dernière tâche a été exécutée sans erreur.
BUSY	Out	BOOL	<ul style="list-style-type: none"> • 0: Pas de tâche de "MB_CLIENT" en cours 1: Tâche de "MB_CLIENT" en cours
ERROR	Out	BOOL	<ul style="list-style-type: none"> • 0: Aucune erreur 1: Une erreur s'est produite. L'origine de l'erreur est affichée via le paramètre STATUS.
STATUS	Out	WORD	Code d'erreur de l'instruction

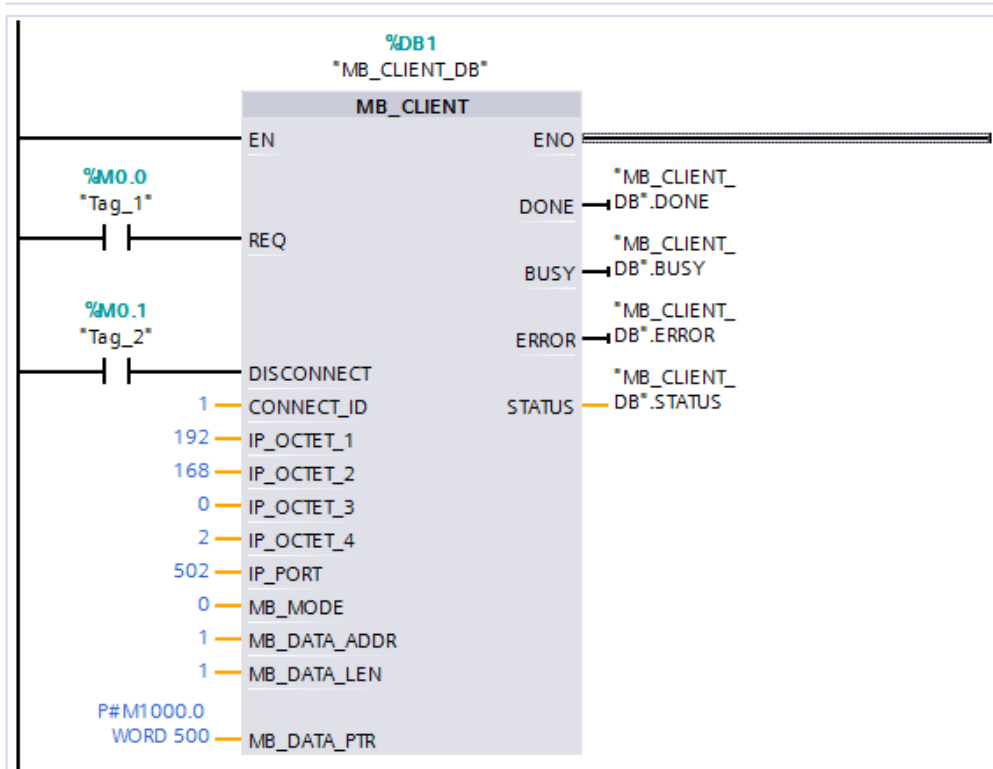


Figure 3_ Paramétrage du DB MB_CLIENT

1.2. MB_SERVER (Modbus TCP)

Se connecter à un client Modbus TCP sur demande, recevoir un message Modbus et envoyer la réponse.

PLC2 du sous-réseau PN/IE_1 (fig.1) sera configuré comme SERVER

On insérer un bloc de données DB MB_SERVER (fig.4) et on fait la configuration de la figure 4 :

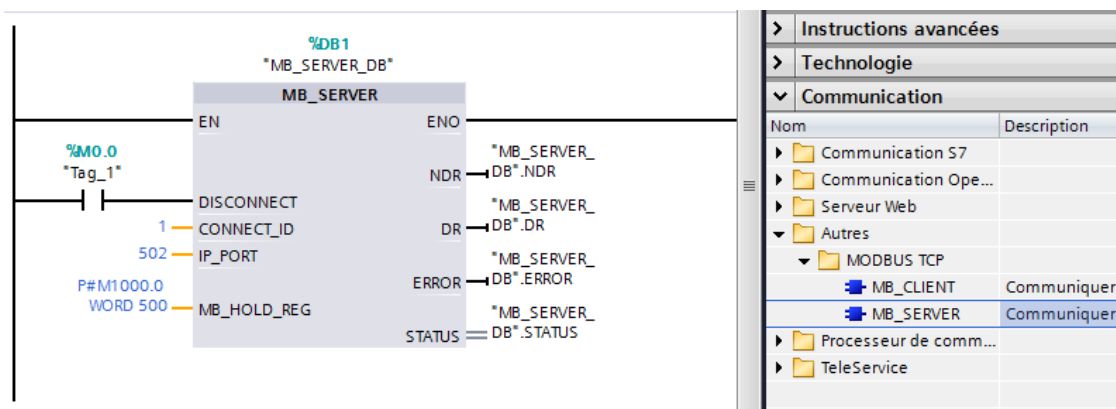


Figure 4_ Paramétrage du DB MB_SERVER

2. Communication MODBUS RTU

Modbus RTU (Remote Terminal Unit : unité terminale distante) est un protocole de communication de réseau standard qui utilise la liaison électrique RS232 ou RS485 pour le transfert de données série entre dispositifs réseau Modbus. Vous pouvez ajouter des ports de réseau point à point (PtP) à une CPU avec un CM RS232 ou RS485 ou un CB RS485.

Modbus RTU utilise un réseau **maître/esclave** où toutes les communications sont déclenchées par un maître unique et où les esclaves peuvent uniquement répondre à une demande du maître. Le maître envoie une demande à une adresse d'esclave et seule cette adresse d'esclave répond à la commande.

2.1. MB_COMM_LOAD

Une exécution de MB_COMM_LOAD sert à configurer les paramètres de port point à point, tels que la vitesse de transmission, la parité et le contrôle de flux. Une fois qu'un port CPU est configuré pour le protocole Modbus RTU, il peut être utilisé uniquement par les instructions MB_MASTER ou MB_SLAVE.

"MB_COMM_LOAD" doit être appelé une fois afin de configurer le port pour le protocole Modbus-RTU. Une fois la configuration achevée, le port peut être utilisé par les instructions "[MB_MASTER](#)" et "[MB_SLAVE](#)".

Paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L	Exécution de l'instruction dans le cas d'un front montant.
PORT	Input	PORT	I, Q, M, D, L ou constante	Identification du port de communication : Une fois que vous avez inséré le module de communication dans la configuration de l'appareil, l'identification du port apparaît dans la liste déroulante sur la connexion de la boîte PORT. Cette constante peut aussi être référencée dans le tableau des variables dans l'onglet "Constantes".
BAUD	Input	UDINT	I, Q, M, D, L ou constante	Sélection de la vitesse de transmission : 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 Toutes les autres valeurs sont invalides.
PARITY	Input	UINT	I, Q, M, D, L ou constante	Sélection de la parité : <ul style="list-style-type: none"> • 0 - Aucune • 1 - Impaire 2 - Paire
MB_DB	Input	MB_BASE	D	Renvoi au bloc de données d'instance de l'instruction " MB_MASTER " ou " MB_SLAVE ". Lorsque vous avez inséré " MB_SLAVE " ou " MB_MASTER " dans votre programme, l'identification du DB est disponible dans la liste déroulante sur la connexion de la boîte MB_DB.
DONE	Output	BOOL	I, Q, M, D, L	Exécution de l'instruction terminée sans erreur.
ERROR	Output	BOOL	I, Q, M, D, L	Erreur : <ul style="list-style-type: none"> • 0 - Pas d'erreur détectée 1 – Indique qu'une erreur a été détectée. Un code d'erreur est renvoyé dans le paramètre STATUS.

STATUS	Output	WORD	I, Q, M, D, L	Code d'erreur de la configuration des ports
--------	--------	------	------------------	---

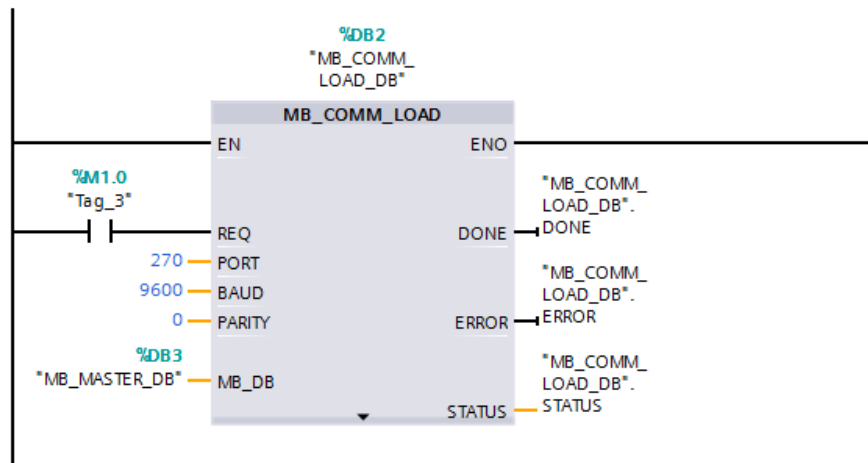


Figure 5_Paramétrage du DB MB_COMM_LOAD

2.2. MB_MASTER

L'instruction "MB_MASTER" permet à votre programme de communiquer en tant que maître Modbus via le port d'un module point à point (CM) ou d'une carte de communication (CB). Vous pouvez accéder aux données d'un ou plusieurs appareils esclaves Modbus.

Pour que l'instruction "MB_MASTER" puisse communiquer avec un port, il faut auparavant exécuter "[MB_COMM_LOAD](#)".

Un DB d'instance est créé lorsque vous insérez l'instruction "MB_MASTER" dans votre programme. Vous indiquez ce DB d'instance dans le paramètre d'entrée MB_DB de l'instruction "[MB_COMM_LOAD](#)".

Paramètre

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L	Requête : <ul style="list-style-type: none"> 0 - Aucune requête 1 - Requête pour envoyer des données à un (des) esclave(s) Modbus
MB_ADDR	Input	UINT	I, Q, M, D, L ou constante	Adresse de station Modbus RTU : <ul style="list-style-type: none"> Plage d'adresses standard : 0 à 247 Plage d'adresses étendue : 0 à 65535 La valeur "0" est réservée à la multidiffusion (broadcast) d'un message adressé à tous les esclaves Modbus. Seuls les codes de fonction Modbus 05, 06, 15 et 16 sont supportés pour une multidiffusion.
MODE	Input	USINT	I, Q, M, D, L ou constante	Sélection du mode : Indique le type de requête : Lecture, écriture ou diagnostic Vous trouverez d'autres détails dans le tableau des fonctions Modbus.
DATA_ADDR	Input	UDINT	I, Q, M, D, L ou constante	Adresse de début dans l'esclave : indique l'adresse de début des données auxquelles on souhaite accéder dans l'esclave Modbus. Vous trouverez les adresses valides dans le tableau des fonctions Modbus.

DATA_LEN	Input	UINT	I, Q, M, D, L ou constante	Longueur des données : indique le nombre de bits ou de mots auxquels cette requête doit accéder. Vous trouverez les longueurs valides dans le tableau des fonctions Modbus.
DATA_PTR	Input	VARIANT	M, D	Pointeur sur l'adresse du DB ou du memento de la CPU pour les données à écrire ou à lire. Dans le cas d'un DB, celui-ci doit être créé avec le type d'accès "Standard - compatible avec S7-300/400".
DONE	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> 0 : La transaction n'est pas achevée 1 : Transaction terminée sans erreur
BUSY	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> 0 : Pas de transaction de "MB_MASTER" en cours 1 : Transaction de "MB_MASTER" en cours
ERROR	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> 0 : Aucune erreur 1 : Erreur ; le code d'erreur est affiché au paramètre STATUS
STATUS	Output	WORD	I, Q, M, D, L	Condition d'exécution

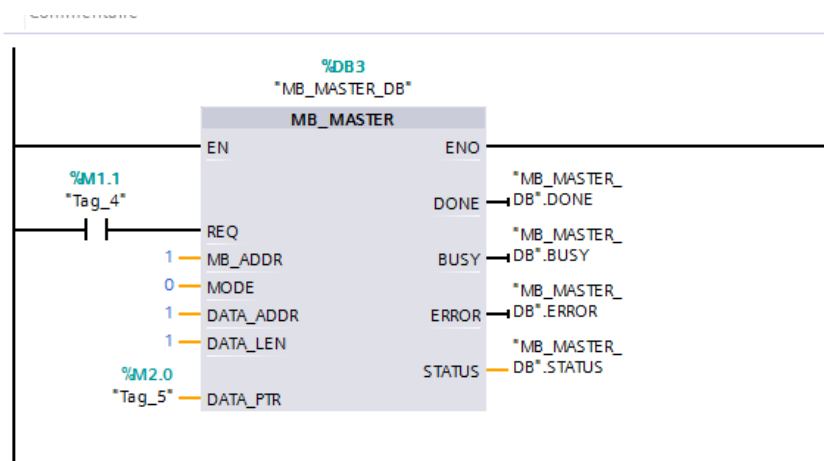


Figure 6_ Paramétrage du DB MB_MASTER

2.3. MB_SLAVE

L'instruction "MB_SLAVE" permet à votre programme de communiquer en tant qu'esclave Modbus via le port d'un module point à point (PtP) ou d'une carte de communication (CB). Un maître Modbus RTU peut envoyer une requête et votre programme répond avec l'exécution de "MB_SLAVE".

Paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
MB_ADDR	Input	USINT	I, Q, M, D, L ou constante	Adresse de station de l'esclave Modbus (plage d'adresses : 0 à 255)
MB_HOLD_REG	Input	VARIANT	D	Pointeur sur le DB de registre de maintien Modbus Le DB doit être créé avec le type d'accès "Standard - compatible avec S7-300/400".
NDR	Output	BOOL	I, Q, M, D, L	De nouvelles données sont prêtes : <ul style="list-style-type: none"> 0 : Pas de nouvelles données

				1: Indique que de nouvelles données ont été écrites par le maître Modbus
DR	Output	BOOL	I, Q, M, D, L	Lecture de données : <ul style="list-style-type: none"> 0: Pas de données lues 1: Indique que des données ont été lues par le maître Modbus
ERROR	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> 0: Pas d'erreur détectée 1: Erreur ; le code d'erreur correspondant est indiqué au paramètre STATUS
STATUS	Output	WORD	I, Q, M, D, L	Code d'erreur

The image shows a software configuration window for a Modbus Slave database. The main area displays a block diagram for the 'MB_SLAVE' block. Inputs include 'EN', 'MB_ADDR' (configured with 'P#M1000.0 WORD 500'), and 'MB_HOLD_REG'. Outputs include 'ENO', 'NDR' (pointing to 'DB.NDR'), 'DR' (pointing to 'DB.DR'), 'ERROR' (pointing to 'DB.ERROR'), and 'STATUS' (pointing to 'DB.STATUS'). The right-hand pane shows a library of communication modules, with 'MODBUS V2.2' selected and 'MB_SLAVE V2.1' highlighted.

Figure 7_ Paramétrage du DB MB_SLAVE