

0001	PROGRAM COMM
0002	
0003	VAR CONSTANT
0004	C_COM_PORT: BYTE := 2;
0005	C_COM_BAUDRATE: COM_BAUDRATE := BAUD_9600;
0006	C_COM_PARITY: COM_PARITY := PARITY_NO;
0007	C_COM_STOPBITS: COM_STOPBITS := STOPBITS_1;
0008	C_COM_BYTESIZE: COM_BYTESIZE := BS_8;
0009	C_COM_FLOW_CONTROL: COM_FLOW_CONTROL := FLOW_CONTROL_TERMINAL_DEFAULT ;
0010	NB_MAX_REQUETES : BYTE := 2;
0011	END_VAR
0012	
0013	VAR
0014	
0015	MODBUS_MASTER1: MODBUS_MASTER_RTU;
0016	
0017	Start: BOOL:=TRUE;       (* Emission d'une trame sur un front montant de cette variable*)
0018	
0019	Send : ARRAY[1..3] OF typModbusQuery;  (* Buffer d'émission: tableau contenant les requêtes à exécuter *)
0020	Receive: typModbusResponse;  (* Buffer de réception*)
0021	
0022	NB_requete: BYTE := 0;
0023	requete: BYTE := 1;
0024	
0025	ErrorComSlave AT %QW256: ARRAY[1..3] OF enumMB_ERROR;
0026	
0027	init: BOOL := TRUE;
0028	
0029	tab1 AT %MW400: ARRAY[0..124] OF WORD;               (*Tableau recup ADR 1*)
0030	
0031	END_VAR
0001	(*Appel de la fonction INIT_REQUETES, appelée lors du premier tour de cycle, pour une initialisation des requêtes à exécuter*)
0002	IF init THEN
0003	COMM.INIT_REQUETES();
0004	init :=FALSE;
0005	requete:=1;
0006	END_IF
0007	
0008	
0009	
0010	(*Appel de l'instance MODBUS_MASTER*)
0011	MODBUS_MASTER1( bCOM_PORT:= C_COM_PORT,                               (*Port de communication*)
0012	cbCOM_BAUDRATE:= C_COM_BAUDRATE,                   (*Vitesse de communication*)
0013	cpCOM_PARITY:= C_COM_PARITY,                       (*Parité*)
0014	csCOM_STOPBITS:= C_COM_STOPBITS,                   (*Nombre de bits de stop*)
0015	cbsCOM_BYTESIZE:= C_COM_BYTESIZE,                   (*Nombre de bits de données*)
0016	cfCOM_FLOW_CONTROL:= C_COM_FLOW_CONTROL,           (*Contrôle de flux*)
0017	TimeOut:= T#1s,                                      (* Temps max avant erreur de TimeOut *)
0018	StartFunction:= Start,
0019	Query:= Send[requete],                               (* Buffer d'émission des données *)
0020	Response:= Receive );                               (* Buffer de réception des données *)
0021	
0022	
0023	IF NOT Start THEN (* Le Start est remis à FALSE par le bloc fonctionnel lorsqu'une réponse est reçue ou si il y a un timeout *)
0024	
0025	ErrorComSlave[requete] := MODBUS_MASTER1.MB_Error;               (*Recopie du statut d'erreur de l'esclave de
0026	
0027	IF ErrorComSlave[requete] = MB_NO_ERROR THEN                       (*S'il n'y a pas d'erreur sur l'escave*)
0028	
0029	(* Gestion des requêtes de lecture*)
0030	CASE requete OF
0031	1 : tab1:= Receive.Data;                                   (* Receive: buffer de réception du bloc fonc
0032	
0033	END_CASE
0034	
0035	END_IF

ans un tableau\*)

ctionnel.\*)

0036		
0037	requete:=requete+1;	
0038	IF requete>NB_requete THEN	(* Si toutes les requêtes ont été exécutées,
0039	requete:=1;	
0040	END_IF	
0041		
0042	Start := TRUE;	
0043	END_IF	

, on reprend au départ pour une exécution en boucle des 2 requêtes\*)



ERROR: syntaxerror  
OFFENDING COMMAND: --nostringval--

STACK:

/Title  
( )  
/Subject  
(D:20120621160447+02'00' )  
/ModDate  
( )  
/Keywords  
(PDFCreator Version 0.9.5)  
/Creator  
(D:20120621160447+02'00' )  
/CreationDate  
(robert.cyril)  
/Author  
-mark-