

Responsable pédagogique	AF	AM	PB	PM	<div>PNET</div> <div>TP Client Modbus over IP</div>
Période	Sem1	Sem2	Sem3	Sem 4	
Volume horaire	Cours		TP		
			8		

Indicateur temporel (hors rédaction du compte-rendu) :

questions	1h	2-8h
1		
2 - 4		

Documents à rendre : **Compte-rendu** selon modèle TP, contenant en plus les sources (en Annexe), les captures d'écran **pertinentes (requête / réponse pour un paramètre modifié par exemple)**. Des résultats obtenus sous forme de photos (films...) et / ou screen validant le fonctionnement du client et serveur pourront aussi être joints.

BUT : Etudier et mettre en œuvre des échanges client / serveur modbus sur IP. Etudier, modifier les différentes zones d'une cartographie mémoire (map) Modbus.

Note : Vous travaillerez en deux temps :

Votre client (que vous allez coder) devra communiquer d'abord avec :

- un serveur **émulant** un système (kart, bateau, hélico...) en adresse ip 127.0.0.1 ! (plus simple dans un premier temps)
- puis quand tout semblera communiquer, vous communiquerez avec un système serveur réel lié (soit au kart, bateau, simulateur hélico en salle 103, voir le vrai en bas...)

Partie 1 : Prise en main avec les binaires fournis (client et serveur)

1. Test avec le serveur en localhost

Après avoir décompressé l'archive *slave_test.zip* (disponible sur rimatara dans PNET/TP4) Tapez la commande : **sudo sh ./slave_test.sh <nom_fichier.csv>** pour lancer le serveur.

Cf figure 1bis. Il est important de lancer le script, et non le serveur directement pour que les liens avec les bibliothèques dynamiques de Qt s'opèrent correctement.

<nom_fichier.csv> est bien sûr à remplacer par kart_banc_2019.csv pour « simuler » le kart ;
arduino_helico.csv pour « simuler » l'hélicoptère ou S420-5.csv pour le bateau 420.

Puis Tapez la commande : `sudo ./cliModbus_prof 127.0.0.1` pour lancer le clientle serveur
nommé `slave_test(esclave)` doit vous répondre `pm@patrickMAC:~/Documents/test_banc$`
`./cliModbus_prof 127.0.0.1`

hostAddr : 100007f

Test en localhost

adresse serveur 127.0.0.1

N socket cliente 3

Tapez votre choix operation : 1 lecture , 2 ecriture , e pour sortir

1

parametre : BT1 tension batterie1 ; BT2 tension batterie2; BT3 tension batterie3; BT4 tension batterie4;

C1 : courant charge batterie1 ; C2 : courant charge batterie2 ; C3 : courant charge batterie3 ; C4 : courant
charge batterie4 ;

P pente ; M masse ; V vitesse; C couple; L:local / distant

P

client de pid 12428

13 carac recus du serveur

octet 3 3f octet 4 33 octet 1 33 octet 2 33

Affiche 4 octets du code
du float en IEEE754

pente lue : 0.700000

Tapez votre choix operation : 1 lecture , 2 ecriture , e pour sortir

2

parametre : BT1 tension batterie1 ; P pente ; M masse ; V vitesse ; C couple ;L:local / distant

P

0.5

val en float 0.500000 , en int 3f000000

client de pid 12428

12 carac recus du serveur

Tapez votre choix operation : 1 lecture , 2 ecriture , e pour sortir

1

parametre : BT1 tension batterie1 ; BT2 tension batterie2; BT3 tension batterie3; BT4 tension batterie4;

C1 : courant charge batterie1 ; C2 : courant charge batterie2 ; C3 : courant charge batterie3 ; C4 : courant
charge batterie4 ;

P pente ; M masse ; V vitesse; C couple; L:local / distant

P

client de pid 12428

13 carac recus du serveur

octet 3 3f octet 4 0 octet 1 0 octet 2 0

pente lue : 0.500000



Figure 1 : menu utilisateur pour le kart

```
sudo sh ./slave_test.sh kart_banc_2019.csv
[sudo] Mot de passe :
config: loading kart_banc_2019.csv...
tcp/ip: Listening to port 502...
tcp/ip: Client 5 connected
tcp/ip: Request from client 5
modbus: FC 3, Address 0016, Number 2
```

Figure 1bis : le côté serveur

2. Développement du client

Méthode et mise en œuvre : Partir du source fourni cliModbus_suj.c

COMPILER avec l'aide du Makefile fourni. make all pour refabriquer le binaire nommé cliModbus ; make clean pour effacer.

CF sur rimatara dans PNET/TP4 : TP_MODBUS_IP.zip

2.1. Renseignement des paramètres de la structure socket

Compléter l'initialisation correcte de la structure socket de type *sockaddr_in* ; notamment les champs *famille de socket*, *port réseau*, et *adresse IP* respectivement liés aux champs *sin_family*, *sin_port* et *sin_addr*.

Remarque : Le port serveur distant étant lié au service réseau « modbus sur ip », il est référencé selon le port 502.

L'adresse ip du serveur devra être saisie sur la ligne de commande.

Enfin rappelez vous que le client et le serveur seront sur des machines potentiellement distinctes.

2.2. Création de la socket cliente

Coder l'initialisation d'une variable nommée *sock_CLIENT* ; descripteur de la socket cliente. Comme vu en cours, vous utiliserez la primitive système *socket()*. Cf *man 2 socket* ...

2.3. Connexion à la socket serveur

Vous devez ici appeler la primitive système connect(). Cf man 2 connect()

Pensez à tester le retour de l'appel système qui au moyen de perror(char*) ; peut vous permettre de savoir pourquoi le serveur n'est pas joignable !

Compilez et testez. Normalement, vous devriez être capable de vous connecter au serveur.

Pour la suite Q3, vous allez utiliser les fonctions suivantes fournies en binaire .o / interface .h (modbus.o / modbus.h dont un extrait est donné ci-dessous) :

```
/* ***** */
/* envoi requete de lecture modbus sur serveur */
/* parametre : entree unsigned short adresse du registre modbus */
/* retour -1.0 : erreur ; valeur lue (float) : si OK */
/* ***** */
float lecture(unsigned short key);

/* ***** */
/* envoi requete de ecriture modbus sur serveur */
/* parametre1 : entree unsigned short adresse du registre modbus */
/* parametre2 : entree float valeur a ecrire dans registre */
/* retour -1 : erreur ; ou 0 : si OK */
/* ***** */
int ecriture(unsigned short key, float val) ;
```

Point IMPORTANT : Les adresses des registres Modbus sont à prendre dans le fichier .csv mis en Annexe.

*Exemple : 40001 donne **adresse 0** à mettre en paramètre de lecture(unsigned short **key**);*

Faire toujours adresse souhaitée (lue dans .csv – 40001) donne offset à mettre dans key.

Exemple : Pente en 40023 donne adresse 22.

3. Enrichir le menu utilisateur

Il devra se rapprocher (dans le cas du kart), si possible de celui montré en exemple (CF figure 1) pour lire / écrire différents paramètres.

Dans le cas des autres parties opératives vous pourrez adapter selon vos besoins.

Dans la cas du kart, certains paramètres (**tension, courant**) doivent être **divisés par 1000** pour avoir une valeur cohérente. (lié à un calcul interne dans automate -10000 / +10000)



Remarque p/r à la variable Local/Distant : Il est conseillé de **mettre 1** dans cette variable AVANT de pouvoir modifier d'autres paramètres Masse pilote par exemple.

4. Tests avec les parties opératives réelles

Remarque : vous pourrez tester votre client avec le serveur réel (armoire automate près du bateau, Kart , arduino esclave en salle 103 ou le vrai serveur en bas près de l'allouette) , en remplaçant simplement l'adresse ip *localhost* par l'adresse réelle ; au lancement de votre client .

SECURITE : (Dans tous les cas demander confirmation aux enseignants avant exécution des programmes) ; il y a potentiellement risque aux personnes .

Cas de figure :

- 1) Kart (en bas dans l'atelier) : `sudo ./cliModbus 192.168.20.205.`

Remarque p/r à la variable Local/Distant : Il est conseillé de mettre 1 dans cette variable AVANT de pouvoir modifier d'autres paramètres Masse pilote par exemple.

Si vous faites des écritures dans certains registres « sensibles » , par exemple dans le cas du kart pour les variables `on_off_run` et `on_off_enable`, attention à remettre dans l'état initial ces registres. Typiquement, un 1 dans ces 2 variables va démarrer le moteur du banc. Il est impératif de remettre 0 ensuite pour arrêter.

- 2) « helico » en salle 103 : `sudo ./cliModbus 192.168.20.201` (IP à confirmer auprès de M.Menu)
- 3) bateau : (en bas dans l'atelier) : `sudo ./cliModbus 192.168.20. ??` .(IP à confirmer auprès de M.Facchin)



Annexe 1 : Fichier kart_banc_2019.csv

(contenant la description de la cartographie)

HOST;192.168.20.205

#HOST;127.0.0.1

PORT;502

INFO;Kart Banc Test 2018 et Surveillance Batteries !

Mise à jour PM 21/3/18

holding registers

commandes du Enable et Run + Mode local ou distant

40001;2; FFFF ;local_dist; prise en main local = 0 - distant = 1; Float ;0.0

40003;2; FFFF ;on_off_run ; start stop; Float ;0.0

40005;2; FFFF ;on_off_enable ;Enable Variateur Parvex; Float ;0.0

#40011 et 40013 sont interdits car utilisés dans Asservissement Couple : Gain , Lag

mesures des parametres dynamiques

40015;2; FFFF ;vit_simu;vitesse moteur (tr/mn); Float ;0.0

40017;2; FFFF ;couple_simu;couple moteur Nm; Float ;0.0

parametres simulateur conduite + essai banc

40019;2; FFFF ;masse_pilote;masse du pilote (kg);Float ;80.0

temps du parcours

40021;2; FFFF ;temps_parcours;temps absolu du parcours ; Float ;0.0

40023;2;FFFF ;cons_pente;consigne de pente piste (%) ;Float ;0.0

mesures courants de "decharges"

40025;2;FFFF ;courant_batt1;courant reel dans batterie 1 ? ;Float ;0.0

40027;2;FFFF ;puissance_totale_batt;puissance totale 4 batteries en Watts ;Float ;0.0

40029;2;FFFF ;charge_elect_totale;charge Q (en coulomb:C) des 4 batteries ;Float ;0.0

mesures tensions batteries en charge et/ou utilisation !!

40033;2;FFFF ;tension_batt1;tension reelle dans batterie 1 ;Float ;0.0

40035;2;FFFF ;tension_batt2;tension reelle dans batterie 2 ;Float ;0.0

40037;2;FFFF ;tension_batt3;tension reelle dans batterie 3 ;Float ;0.0

40039;2;FFFF ;tension_batt4;tension reelle dans batterie 4 ;Float ;0.0

mesures courants de charges !!

40041;2;FFFF ;courant_charge_batt1;courant reel dans batterie 1 ;Float ;0.0

40043;2;FFFF ;courant_charge_batt2;courant reel dans batterie 2 ? ;Float ;0.0

40045;2;FFFF ;courant_charge_batt3;courant reel dans batterie 3 ? ;Float ;0.0

40047;2;FFFF ;courant_charge_batt4;courant reel dans batterie 4 ? ;Float ;0.0



Annexe 2 : Fichier S420-5.csv

(contenant la description de la cartographie du S420)

HOST localhost
PORT 502

INFO Roll a Boat

holding registers

40001	2	FFFF	lect-roulis	position roulis (deg)	Float	0
40003	2	FFFF	lect-tangage	position tangage (deg)	Float	0
40005	2	FFFF	lect-ventr	position baume (deg) - consigne pour la bome physique	Float	0
40007	2	FFFF	lect-vitazimut	vitesse azimuth (deg/s)	Float	0
40009	2	FFFF	lect-posazimut	position azimuth (deg)	Float	0
40011	2	FFFF	cons-etat	prise en main a distance 0 = jeu 1 = manu	Float	0
40013	2	FFFF	cons-gouv	position gouvernail (+/-deg)	Float	0
40015	2	FFFF	cons-roulis	consigne de position roulis (deg)	Float	0
40017	2	FFFF	cons-tangage	consigne de position tangage (deg)	Float	0
40019	2	FFFF	cons-anglebome	consigne de position baume (deg) angle réel bome, resultant de l'ecart avec lect-ventr	Float	0
40021	2	FFFF	cons-vitazimut	consigne de vitesse azimuth (+/-deg/s)	Float	0
40023	2	FFFF	cons-acc	consigne de l'acceleration (-1 a 1)	Float	0
40025	2	FFFF	cons-forcevent	consigne de la force du vent en nœuds	Float	0
40027	2	FFFF	cons-anglevent	consigne de la direction du vent (absolue, renseignée la pom)	Float	0
40029	2	FFFF	cons-hautvague	consigne de hauteur de vague (m)	Float	0
40031	2	FFFF	cons-vitvague	consigne vitesse de vague (m/s)	Float	0
40033	2	FFFF	cons-intervague	consigne de distance inter vagues (m inter crêtes)	Float	0



Annexe 3 : Fichier arduino_helico.csv

(contenant la description de la cartographie de l'hélicoptère)

MODBUS.CSV

LLF FS configuration file for QamModbusMap

adresse reelle a decommenter si dialogue avec simulateur en salle 103

#HOST 192.168.1.201

HOST 127.0.0.1

PORT 502

INFO LLF FS Modbus Slave #1

Configuration Esclave

4xxxx - holding registers

"40001	1 FFFF	Sampling	période d'échantillonnage en ms	Uint	200"
"40002	1 FFFF	Slow	période clignotement lent en ms	Uint	500"
"40003	1 FFFF	Fast	période clignotement rapide en ms	Uint	250"
"40004	1 FFFF	Short	durée min. clic court en ms	Uint	300"
"40005	1 FFFF	Long	durée min. clic long en ms	Uint	3000"

Module 1 (ALT)

4xxxx - holding registers

"40010	1 FFFF	TOR-1	module 1 - E/S TOR	Hex	0"
"40010	1	30 SelD-1	module 1 - Sélecteur D	Uint	0"
"40010	1	300 LedA-1	module 1 - Led A rouge	Uint	0"
"40010	1 0C00	LedB-1	module 1 - Led B jaune	Uint	0"

3xxxx - input registers

"30010	1 FFFF	AdjC-1	module 1 - réglage C	Uint	0"
--------	--------	--------	----------------------	------	----

Module 5

4xxxx - holding registers

"40050	1 FFFF	TOR-5	module 5 - E/S TOR	Hex	0"
"40050	1	30 SelD-5	module 5 - Sélecteur D	Uint	0"
"40050	1	300 LedA-5	module 5 - Led A rouge	Uint	0"
"40050	1 0C00	LedB-5	module 5 - Led B bleue	Uint	0"

Module 9

4xxxx - holding registers

"40090	1 FFFF	TOR-9	module 9 - E/S TOR	Hex	0"
"40090	1	3 IntA-9	module 9 - Interrupteur A	Uint	0"
"40090	1 000C	IntB-9	module 9 - Interrupteur B	Uint	0"
"40090	1	30 IntC-9	module 9 - Interrupteur C	Uint	0"
"40090	1 00C0	InvD-9	module 9 - Inverseur D	Uint	0"
"40090	1	300 LedA-9	module 9 - Led A rouge	Uint	0"
"40090	1 0C00	LedB-9	module 9 - Led B jaune	Uint	0"
"40090	1	3000 LedC-9	module 9 - Led C verte	Uint	0"