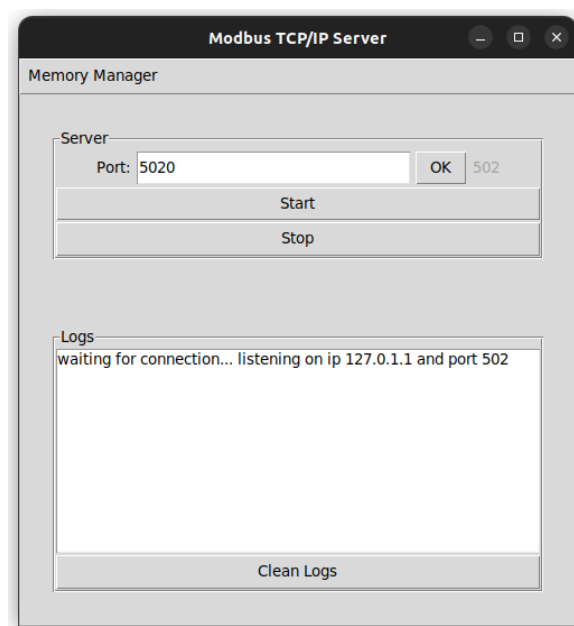


Aide pour le TP MODBUS TCP/IP

Lancement du serveur ModBus TCP/IP

- Télécharger depuis le gitlab « https://gitlab.u-angers.fr/cours/industrial_network_student/ », Dossier « TP », le dossier « MODBUS ».
- Dans le dossier « TCPIP_MODBUS_SERVER_PYTHON_PEP8 », exécuter le fichier main.py.

Description de l'interface



Le menu « Memory Manager », permet d'accéder en lecture et écriture à la mémoire du serveur, cette interface est détaillée après.

La zone « Server » permet de démarrer le server (Start), de le stopper (Stop) et de changer le port d'écoute. Par défaut le port est 502 pour Modbus TCP/IP mais sur certaines configurations il faut les droits administrateurs pour accéder à ce port. Si c'est le cas vous pouvez le modifier pour mettre 5020 par exemple, ce qui ne devrait pas poser de problème.

La zone « Logs » permet d'afficher l'état du serveur, l'IP et le port sur lequel il écoute quand il est lancé, les clients qui se connectent mais aussi les trames qu'il reçoit et qu'il envoie (ce qui peut être très pratique pour déboguer le code du client).

Memory Manager

The screenshot shows a window titled "Memory Manager" with standard window controls (minimize, maximize, close). It contains two main sections:

- Memory State (bits)**:
 - Memory range: B0 to B499
 - Read**: A text input field labeled "B" with the value "0" displayed below it. A "Read" button is at the bottom.
 - Write**: Two text input fields labeled "B" and ", value (0 or 1)". A "Write" button is at the bottom.
- Memory State (words)**:
 - Memory range: W0 to W299
 - Read**: A text input field labeled "W". Below it, "Value (10):" is 0 and "Value (16):" is 0x00. A "Read" button is at the bottom.
 - Write**: Two text input fields labeled "W" and ", value (16)". A "Write" button is at the bottom.

Cette interface permet d'accéder en lecture et écriture à la mémoire du serveur. La mémoire est divisée en deux parties indépendantes : les bits mémoires et les mots mémoires.

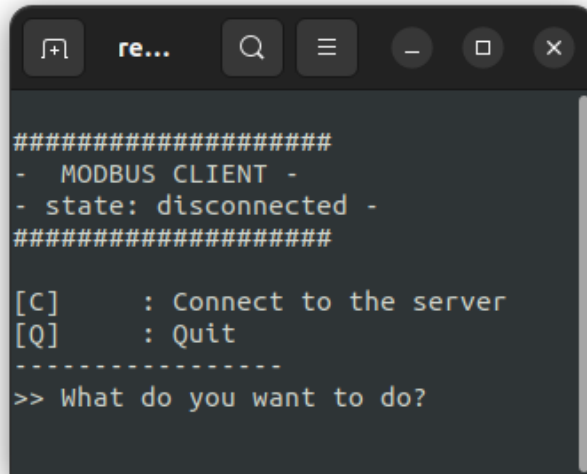
La zone « Memory State (bits) » permet d'accéder à la partie « bit » de la mémoire. Il est possible de lire la valeur d'un bit (« Read ») en mettant l'adresse en base 10. Il est aussi possible de modifier la valeur d'un bit (1 ou 0), toujours en mettant l'adresse en base 10, à l'aide du bouton « Write ».

La zone « Memory State (words) » permet d'accéder à la partie « mot » de la mémoire (qui est indépendante de la zone mémoire « bit » : écrire une valeur sur B0 ne modifie en rien la valeur de W0). Il est possible de lire la valeur d'un mot à l'aide du bouton « Read » en mettant l'adresse du mot en base 10. La valeur du mot est donnée elle en base 10 et en base 16. Il est aussi possible de modifier la valeur d'un mot (toujours en donnant l'adresse en base 10) à l'aide du bouton « Write ». Attention, pour l'écriture, la valeur du mot est à donner en base 16.

Lancement du client MODBUS TCP/IP

- Télécharger depuis le gitlab « https://gitlab.u-angers.fr/cours/industrial_network_student/ », Dossier « TP », le dossier « MODBUS ».
- Dans le dossier « CLIENT_MODBUS_PEP8 », exécuter le fichier main.py.

Vous devriez avoir la sortie suivante :



```
#####
- MODBUS CLIENT -
- state: disconnected -
#####

[C]      : Connect to the server
[Q]      : Quit
-----
>> What do you want to do?
```

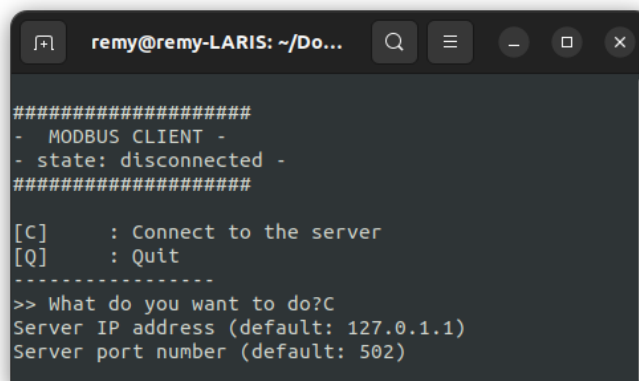
Se connecter au serveur

Il faut d'abord avoir lancé et démarré le serveur...

Pour se connecter il faut taper 'C', et valider en appuyant sur entrée. Ensuite est demandé l'adresse IP du serveur. Soit vous rentrer manuellement l'IP correspondante (cf les logs du serveur), soit l'IP par défaut est OK et vous pouvez directement appuyer sur entrée.

Remarque : pour modifier l'IP par défaut, vous pouvez la changer dans le fichier global_variables.py, c'est la variable MODBUS_SERVER_IP. Il est fortement recommandé de mettre l'IP de votre serveur pour ne pas avoir à la ressaisir à chaque test...

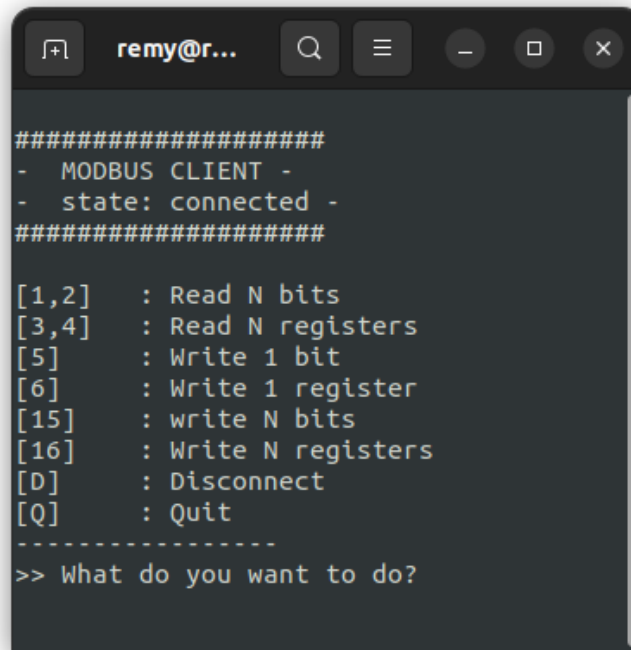
Ensuite vous devez rentrer le numéro de port, c'est le même principe que pour l'IP (la variable correspondante dans le fichier global_variables.py est MODBUS_SERVER_PORT).



```
#####
- MODBUS CLIENT -
- state: disconnected -
#####

[C]      : Connect to the server
[Q]      : Quit
-----
>> What do you want to do?C
Server IP address (default: 127.0.1.1)
Server port number (default: 502)
```

Une fois connecté vous devriez avoir le menu suivant :

A screenshot of a terminal window with a dark background. The window title bar shows 'remy@r...' and standard window controls. The terminal displays a menu for a Modbus client. It starts with a separator line of hashes, followed by '- MODBUS CLIENT -' and '- state: connected -', and another separator line. Below this is a list of options: '[1,2] : Read N bits', '[3,4] : Read N registers', '[5] : Write 1 bit', '[6] : Write 1 register', '[15] : write N bits', '[16] : Write N registers', '[D] : Disconnect', and '[Q] : Quit'. A dashed line separates the menu from the prompt '>> What do you want to do?'.

```
#####  
- MODBUS CLIENT -  
- state: connected -  
#####  
  
[1,2] : Read N bits  
[3,4] : Read N registers  
[5] : Write 1 bit  
[6] : Write 1 register  
[15] : write N bits  
[16] : Write N registers  
[D] : Disconnect  
[Q] : Quit  
-----  
>> What do you want to do?
```

Ce menu permet d'appeler les différentes fonctions ModBus pour la lecture et l'écriture de mots et de bits.

Travail à faire

Il faut programmer les fonctions ModBus présentes dans le fichier « functions.py ». Ce sont ces fonctions qui sont appelées lors d'un choix dans le menu ci-dessus.

Les codes à compléter sont marqués par un commentaire #TODO dans le fichier de départ. Au début de chaque fonction il y a un commentaire qui indique les entrées et sorties.

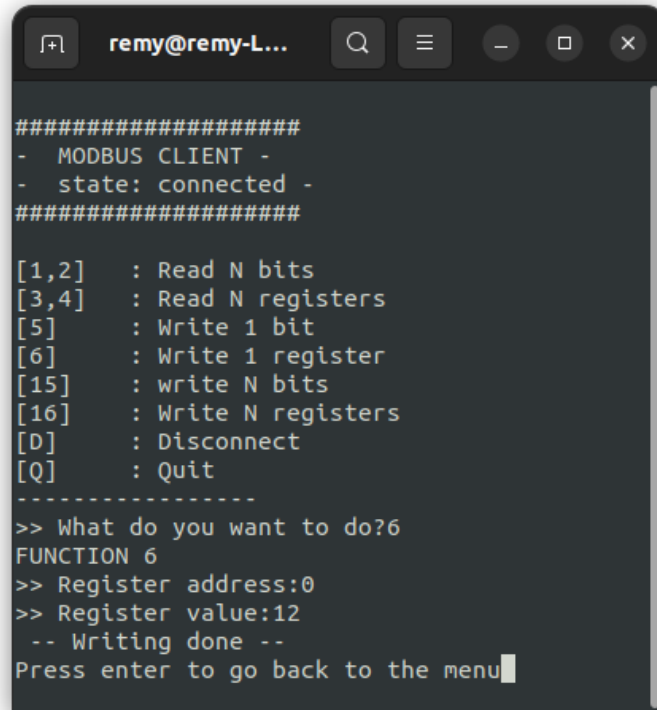
La syntaxe pour envoyer et recevoir une trame est indiquée dans les commentaires au début du fichier functions.py.

Les fonctions lire et écrire n bits sont les plus compliquées. Il est recommandé de programmer les fonctions dans l'ordre suivant :

1. Écrire 1 mot, écrire 1 bit
2. Lire n mots, écrire n mots
3. Lire n bits, écrire n bits

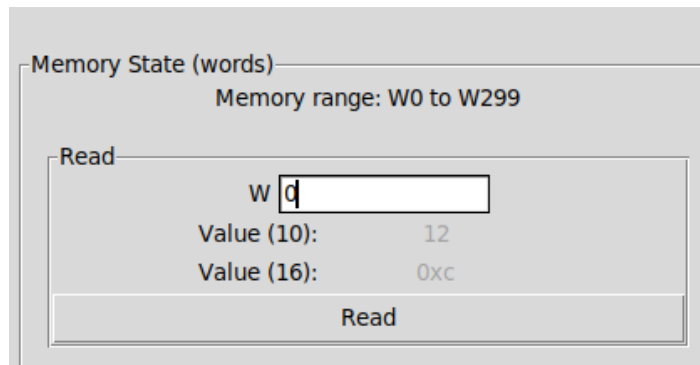
Tester avec le serveur

Il est fortement recommandé de tester les fonctions une par une... Pour cela vous pouvez utiliser le serveur et plus particulièrement l'interface de gestion de sa mémoire. Par exemple pour écrire 1 mot, on pourra essayer de mettre la valeur $(12)_{10}$ dans le mot W0 :

A terminal window titled 'remy@remy-L...' with a dark background and light text. It shows the MODBUS client interface. At the top, it says '##### - MODBUS CLIENT - - state: connected - #####'. Below this is a menu of functions: '[1,2] : Read N bits', '[3,4] : Read N registers', '[5] : Write 1 bit', '[6] : Write 1 register', '[15] : write N bits', '[16] : Write N registers', '[D] : Disconnect', and '[Q] : Quit'. A dashed line separates the menu from the prompt '>> What do you want to do?6'. The user has entered '6', and the prompt changes to 'FUNCTION 6'. Then, the user enters '0' for the register address, and the prompt changes to '>> Register value:12'. The user enters '12', and the prompt changes to '-- Writing done --'. Finally, the prompt changes to 'Press enter to go back to the menu'.

```
#####  
- MODBUS CLIENT -  
- state: connected -  
#####  
  
[1,2] : Read N bits  
[3,4] : Read N registers  
[5] : Write 1 bit  
[6] : Write 1 register  
[15] : write N bits  
[16] : Write N registers  
[D] : Disconnect  
[Q] : Quit  
-----  
>> What do you want to do?6  
FUNCTION 6  
>> Register address:0  
>> Register value:12  
-- Writing done --  
Press enter to go back to the menu
```

Et vérifier sur le serveur que la valeur a bien été écrite.

A window titled 'Memory State (words)' with a light gray background. It shows the memory range 'W0 to W299'. Below this is a 'Read' section with a text input field containing 'W0'. Below the input field are two labels: 'Value (10):' with the value '12' and 'Value (16):' with the value '0xc'. At the bottom of the 'Read' section is a 'Read' button.

Memory State (words)

Memory range: W0 to W299

Read

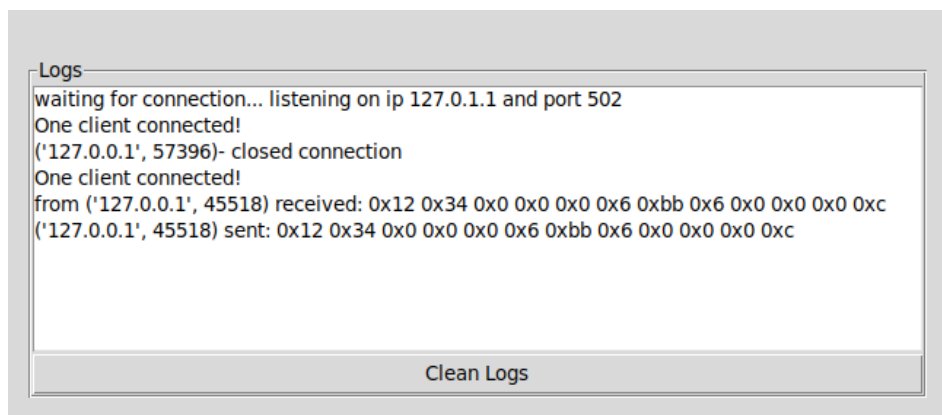
W0

Value (10): 12

Value (16): 0xc

Read

En cas de problème, n'oubliez pas de regarder les logs du serveur pour voir les trames envoyées.

A window titled 'Logs' with a light gray background. It shows a list of log messages. The messages are: 'waiting for connection... listening on ip 127.0.1.1 and port 502', 'One client connected!',>('127.0.0.1', 57396)- closed connection', 'One client connected!', 'from ('127.0.0.1', 45518) received: 0x12 0x34 0x0 0x0 0x0 0x6 0xbb 0x6 0x0 0x0 0x0 0xc', and>('127.0.0.1', 45518) sent: 0x12 0x34 0x0 0x0 0x0 0x6 0xbb 0x6 0x0 0x0 0x0 0xc'. At the bottom of the window is a 'Clean Logs' button.

Logs

waiting for connection... listening on ip 127.0.1.1 and port 502

One client connected!

('127.0.0.1', 57396)- closed connection

One client connected!

from ('127.0.0.1', 45518) received: 0x12 0x34 0x0 0x0 0x0 0x6 0xbb 0x6 0x0 0x0 0x0 0xc

('127.0.0.1', 45518) sent: 0x12 0x34 0x0 0x0 0x0 0x6 0xbb 0x6 0x0 0x0 0x0 0xc

Clean Logs

Tester avec les tests unitaires

Il est aussi possible d'utiliser les tests unitaires pour valider votre code. Il est cependant recommandé de les utiliser une fois que vous avez un comportement qui vous semble OK sur le serveur (il est plus facile de déboguer sur le serveur qu'avec les tests).

Pour utiliser les tests unitaires vous pouvez exécuter le script run_tests.py. L'objectif final est de passer tous les tests pour avoir :

```
[OK] Read N bits 1
. [OK] Read N bits 2
. [OK] Read N bits Error
. [OK] Read N words 1
. [OK] Read N words 2
. [OK] Read N words Error
. [OK] Write 1 bit Error
. [OK] Write 1 bit to False
. [OK] Write 1 bit to True
. [OK] Write 1 word 1
. [OK] Write 1 word 2
. [OK] Write 1 word Error
. [OK] Write N bits 1
. [OK] Write N bits 2
. [OK] Write N bits Error
. [OK] Write N words 1
. [OK] Write N words 2
. [OK] Write N words Error
. ***** NOTES *****
[1.0] Read N bits
[1.0] Read N words
[1.0] Write 1 bit
[1.0] Write 1 word
[1.0] Write N bits
[1.0] Write N words
-----
[6.0] Total
```

(Si vous n'avez pas de couleurs sur votre sortie écran vous pouvez ajouter le paquet colorama à python, mais rien d'obligatoire).

Remarque : la partie « Notes » est donné à titre indicatif, elle peut être amenée à changer.