

Aplicatie pentru monotorizarea resurselor SO ( Client Modbus TCP )

**Documentatie**

**Echipa :**

Barbu Bogdan-Cosmin ( *Grupa 1310A* )

Petrisor Iosif-Marcelin ( *Grupa 1310A* )

**Link Github :**

https://github.com/BogdanCosminBarbu/RC\_PR

**Ce este Modbus ?**

Protocolul Modbus a fost realizat in 1979 de Modicon, pentru automatizare industriala si pentru controlere programabile. De atunci a devenit o metoda standard pentru transfer discret/analog al informatiilor si datelor I/O dintre controlul industrial si componentele de monotirizare.

Dispozitivele Modbus folosesc o comunicare master-slave ( client – server ) in care doar un singur dispozitiv poate initia tranzactiile ( query ). Celelalte dispozitive ( slaves/servers ) raspund prin oferirea de data catre master, sau prin luarea unei actiuni cerute in query.

**Ce este Modbus TCP/IP ?**

Este un protocol Modbus RTU cu o interfata TCP ce functioneaza prin Ethernet.

TCP/IP se refera la Transmission Control Protocol and Internet Protocol care ofera mediul de transmisie pentru ModBus.

ModBus TCP/IP foloseste TCP/IP si Ethernet pentru a transfera data de la mesajul ModBus-ului intre dispozitive compatibile, date de tip binar intre calculatoare.

Acesta utilizeaza 4 tipuri de date : coils, discrete inputs, input registers si holding registers, fiecare cu spatiu de adrese diferite.

Coils: 0xxxx , Discrete inputs: 1xxxx

Input registers: 3xxxx, Holding registers: 4xxxx

**Ce resurse monotorizam ?**

Vom monotoriza coils, registri,procent memorie ,procent cpu , daca este conectat sau nu la internet .

**Functiile de baza ale ModBus-ului :**

*FUNCTIA 01 - Read Coil Status*

Aceasta comanda va citi starea ON / OFF a iesirilor discrete sau a bobinelor (adrese de referinta 0x) în slave / server.Transmisia prin difuzare(broadcast) nu este acceptata.

Interogarea de citire a starii bobinei(Read Coil Status query)

specifica bobina de pornire (canalul de iesire) si cantitatea de bobine de citit.

*FUNCTIA 03 - Read Holding Registers*

Citeste continutul binar al registrelor de pastrare (4x adrese de referinta) în dispozitivul slave. Transmisia prin difuzare nu este acceptata.

Interogarea de citire a registrelor de stocare specifica registrul de pornire si cantitatea de registre care trebuie citite.

*FUNCTIA 04 - Read Input Registers*

Aceasta comanda va citi continutul binar al registrelor de intrare (adrese de referinta 3x) în dispozitivul slave. Transmisia prin difuzare nu este acceptata.

Interogarea registrelor de intrare specifica registrul de pornire si cantitatea de registre care trebuie citite.

*FUNCTIA 05 - FORCE SINGLE COIL*

Forteaza un singur coil/output ON sau OFF (adresa de referinta 0x ) . Cu transmisie broadcast(adresa 0) forteaza acelasi coil in toate retelele slave.

Query-ul de force single coil specifica adresa de referinta a coil-ului unde sa fie fortat si starea la care il fortam.

State-ul de on/off se specifica printr-o constanta.

The force single coil ar ca mesaj de raspuns o copie a query-ului dar returnat dupa executia comenzii

Nu se primeste raspuns de la query-ul de tip master( serial modbus )

*FUNCTIA 06 - PRESET SINGLE REGISTER*

Functia preseteaza un registru cu o valoare anume. Transmisia de tip broadcast este suportata de comanda si va preseta acelasi registru in toate retelele slave.

Query-ul de preset single register specifica adresa de referinta a registrului sa fie presetata si valoarea acestuia.

Raspunsul functiei este o copie a query-ului returnat dupa ce se seteaza elementele acestuia.

*FUNCTIA 15 - FORCE MULTIPLE COILS*

Forteaza simultan o serie de coils sa fie ON sau OFF. Transmisia de tip broadcast este suportata si va forta acelasi block de coils in toate retelele slave.

The force multiple coils query specifica adresa coil-ul de inceput sa fie fortat, nr de coils, si data pe care o fortam sa fie scrisa in ordine. Starile de ON/OFF sunt specificate de datele din fieldurile de query.

Un bit de 1 logic in acest field solicita coil-ul sa fie ON, in timp ce un bit de 0 logic corespunde o cerere de coil OFF. Un bit nefolosit ar trebui sa fie setat 0.

*FUNCTIA 16 - PRESET MULTIPLE REGISTERS*

Preseteaza un block de registri cu o valoare anume. Transmisia broadcast este suportata si va preseta acelasi bloc de registri in toate retelele slave.

Query-ul de preset multiple registers speficia adresa de start a registrului de start, numarul de registri, si datele care trebuie scrie in registri .

**Codurile de expceptie la erori :**

1) *Illegal Function*

Codul functiei primit în interogare nu este permis sau invalid.

2) *Illegal Data Address*

Adresa de date primita în interogare nu este o adresa permisa pentru slave sau este nevalida.

3) *Illegal Data Value*

O valoare continuta în câmpul de date de interogare nu este o valoare admisibila pentru slave sau este nevalida.

4) *Slave/Server Device Failure*

Serverul a esuat in timpul executiei. O eroare a fost intalnita in timp de slave/serverul a incercat sa execute task-ul cerut.

5) *Acknowledge*

Slave-ul/Server-ul a acceptat cererea si o proceseara, dar este necesara o durata lunga de procesare.

Acest raspuns este returnat ca sa previna o eroare de timeout in master.

6) *Slave/Server Decive Busy*

Dispozitivul slave este intr-o procesare de lunga durata a unei comenzi.

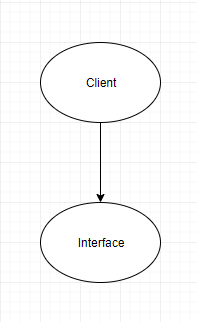
Dispozitivul master ar trebui sa retransmita mesajul mai tarziu cand dispozitivul slave este liber.

**Implementare:**

Pentru implementarea aplicatiei ne-am folosit de limbajul Python si de cateva librarii ale acestuia (tkinter, struct, socket, etc.)

Am construit o clasa de baza care ne descrie clientul si functionalitatile acestuia

si clasa de interfata care ce cuprinde User Interface-ul cu butoane care apeleaza functiile Modbus-ului.



Img. 1

**Clasa Client:**

Contine constructorul care creeaza un client cu adresa ip, id-ul si port-ul specificate global cat si functiile acestuia.

Functiile sunt implementate pe baza unei structuri care trimite toate infromatiile necesare continute intr-un pachet de baza al unui modbus:

*Transaction identifier (2 Bytes):* acest camp de identificare este folosit pentru tranzactii de legare cand mai multe mesaje sunt trimise pe aceeasi conexiune TCP de catre client fara a astepta pentru un raspuns.

*Protocol identifier (2bytes):* acest camp este mereu 0 pentru serviciile modbus.

*Length (2 bytes):* acest camp este numarul de bytes de la fieldurile ramase si include si unit identifier-ul, byte-ul codului functiei si campurile de date.

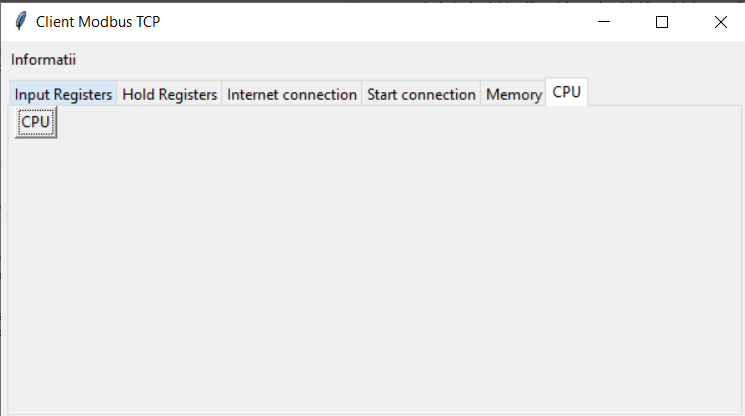
*Unit identifier (1 byte):* acest camp este folosit pentru identificarea unui remote de la server care e localizat pe o retea care nu e TCP/IP. De obicei aceasta este 00 ori FF, deseori ignorata.

Se regaseste si mecanismult de autentificare baza pe coils, ce se regasesc la adresele 1 2 3 4. Astfel se va face o verificare pentru a se putea realiza conexiunea.

Pe langa functiile de baza sunt si functiile de verificarea conexiunii la internet, de detectare a procentului de memorie folosita si procentul de cpu.

**Clasa Interfata:**

A fost efectuata cu ajutorul tkinter, iar cu aceasta am creat un notebook de unde putem alege ce resursa dorim sa vizualizam.



Img. 2

Pentru fiecare *frame* am implementat un buton care realizeaza o comanda de afisare a resursei specificate.

Comanda se realizeaza cu ajutorul unor functii lambda pentru a putea prelucra rezultatul functiilor din Client.

**Referinte:**

[1] [*https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/m218prg/m218prg/M218\_Ethernet\_Configuration/M218\_Ethernet\_Configuration-4.htm*](https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/m218prg/m218prg/M218_Ethernet_Configuration/M218_Ethernet_Configuration-4.htm)

[2] [*https://www.rtautomation.com/technologies/modbus-tcpip/*](https://www.rtautomation.com/technologies/modbus-tcpip/)

[3] [*https://www.prosoft-technology.com/kb/assets/intro\_modbustcp.pdf*](https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf)

[4] [*https://www.simplymodbus.ca/*](https://www.simplymodbus.ca/)

[5] [*https://docs.python.org/3/library/socket.html*](https://docs.python.org/3/library/socket.html)