Aplicație pentru monitorizarea resurselor SO
Client Modbus TCP

Ioana Ichim 1309A
Gafencu Gabriel 1309A

Dispozitivele Modbus comunică utilizând o tehnică master-slave în care un singur dispozitiv(master) poate iniția tranzacții (numite interogări). Celelalte dispozitive (slaves) răspund furnizând datele solicitate masterului sau luând măsurile solicitate în interogare. Un slave este orice dispozitiv periferic care procesează informații și își trimite ieșirea către master folosind Modbus.

Master-ul stabilește o conexiune cu Slave-ul . Acesta așteaptă o conexiune primită de la Master. Odată stabilită o conexiune, serverul răspunde la cererile de la client până când clientul închide conexiunea.

Fig. 1 Schema logică de funcționare Master-Slave

Protocolul MODBUS definește o unitate de date de protocol (PDU) simplă.Maparea protocolului MODBUS poate introduce câteva câmpuri suplimentare pe Application Data Unit (ADU).

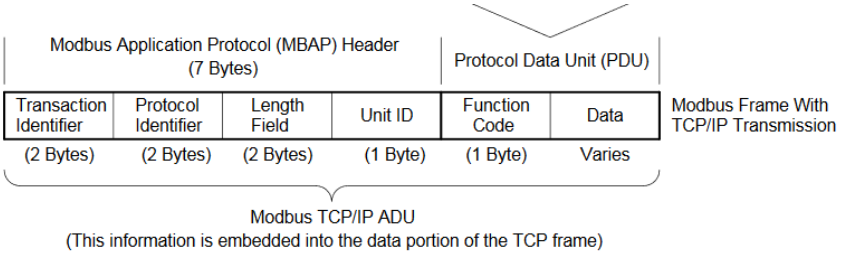

Fig. 2 Structura unui pachet Modbus TCP

Modbus TCP / IP Application Data Unit (ADU) ia forma unui antet format din MBAP(transaction identifier + protocol identifier + length field + unit identifier) și Protocol Data Unit(PDU) (cod funcție + date).

Antetul MBAP are o lungime de 7 octeți și include următoarele câmpuri:

• *Transaction Identifier* (2 octeți): Acest câmp de identificare este utilizat pentru asocierea tranzacțiilor atunci când mai multe mesaje sunt trimise de-a lungul aceleiași conexiuni TCP de către un client fără a aștepta un răspuns anterior .

• *Protocol Identifier* (2 octeți): acest câmp este întotdeauna 0 pentru serviciile Modbus și alte valori sunt rezervate pentru viitoarele extensii. Pentru Modbus acesta este 0.

• *Length Field* (2 octeți): acest câmp este un număr de octeți al câmpurilor rămase și include octetul identificator de unitate, octetul codului funcției și câmpurile de date.

• *Unit Identifier* (1 octet): Acest câmp este utilizat pentru a identifica un server la distanță situat pe o rețea non TCP / IP (pentru conectarea în serie). Într-o aplicație tipică de server Modbus TCP / IP, ID-ul unității este setat la 00 sau FF.

Modelul de date Modbus are o structură simplă care cu patru tipuri de date de bază:

Object type	Access	Size	Address Space
Coil	Read-write	1 bit	00001 - 09999
Discrete input	Read-only	1 bit	10001 - 19999
Input register	Read-only	16 bits	30001 - 39999
Holding register	Read-write	16 bits	40001 - 49999

Fig. 3 Structurile de date în Modbus

Programul va folosi biblioteca psutil pentru a obține informații legate de starea sistemului de operare, precum procentul de utilizare a procesorului, memoria RAM utilizată, procentul de memorie valabilă.

- În cadrul **Input Registers** va fi mapat in primii 2 regiștri procentul de utilizare al CPU (partea întreagă în primul, iar cea fracționară în al doilea).
- Memoria utilizată va fi mapată pe primii 2 regiștrii din **Holding Registers**(la fel parte întragă,parte fracționară), iar în regiștrii 11 si 12 se vor reține date despre spațiul valabil de pe disk .

Protocolul Modbus definește mai multe coduri de funcții pentru accesarea regiștrilor Modbus. Există patru blocuri de date diferite definite de Modbus, iar adresele sau numerele de registru din fiecare dintre acestea se suprapun. Prin urmare, pentru accesarea locului unde se găsește o bucată de date necesită atât adresa (sau numărul de registru), cât și codul funcției (sau tipul de registru).

Function Code	Register Type
1	Read Coil
2	Read Discrete Input
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Holding Register
15	Write Multiple Coils
16	Write Multiple Holding Registers

Fig. 4 Funcțiile de bază

Când un Slave Modbus preia un pachet, dar determină că există o eroare în cerere, acesta va returna un răspuns de cod de excepție în locul unui răspuns de date. Răspunsul la excepție constă din adresa slave sau numărul unității, o copie a codului funcției cu bitul cel mai semnificativ setat pe 1 (cu 80H mai mare) și un cod de excepție. Codurile de excepție pot fi:

1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.
3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.
4	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action
6	Slave Device Busy	The slave is engaged in processing a long-duration command. The master should try again later.

Fig. 5 Codurile de excepție de bază

Pentru *mecanismul de autentificare* biții din coils de la adresele 7, 77,777, 7777 vor fi trecuți pe 1. Serverul va verifica ca prima cerere să respecte

această convenție. Funcția de acces folosită pentru scrierea în coils va fi funcția 5.

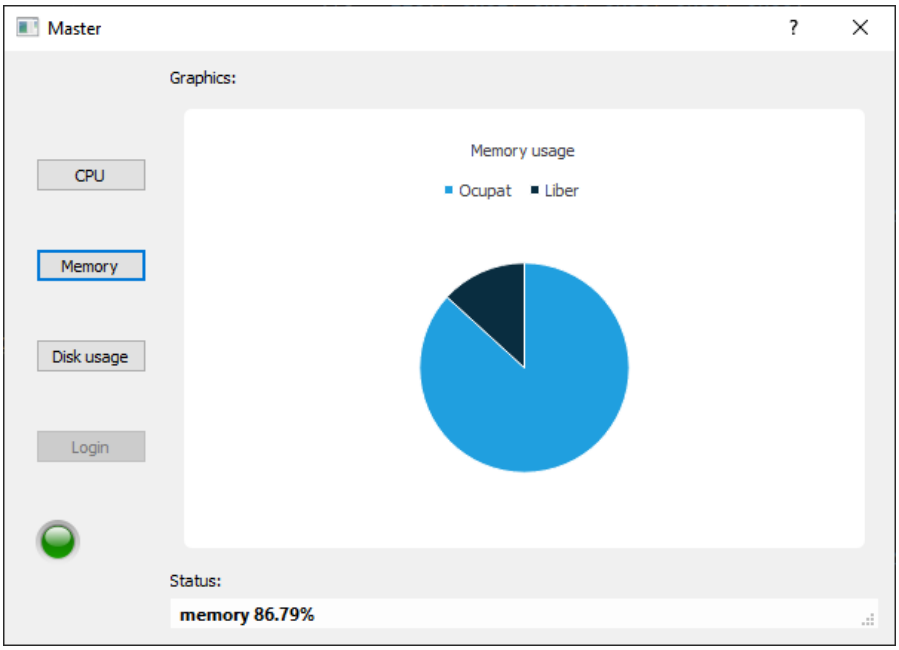

Fig. 6 Interfața Master

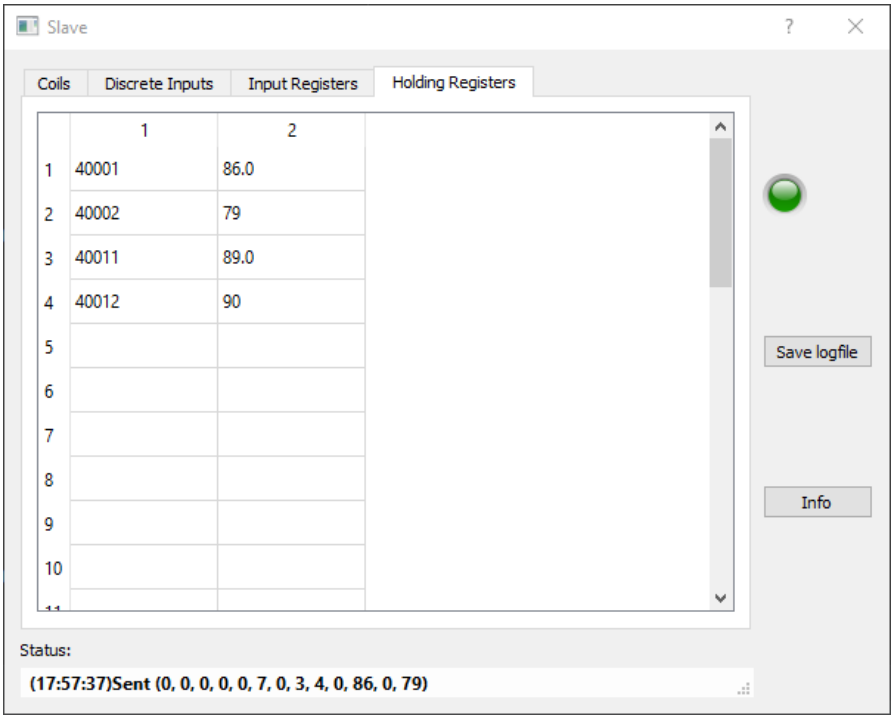

Fig. 7 Interfața Slave

Referințe

[1]. BusWorks® 900EN Series 10/100M Industrial Ethernet I/O Modules w/Modbus Technical Reference – Modbus TCP/IP,

[2]. https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf

[3]. https://www.automation.com/en-us/articles/2012-1/introduction-to-modbus