Modbus komunikacioni protokol

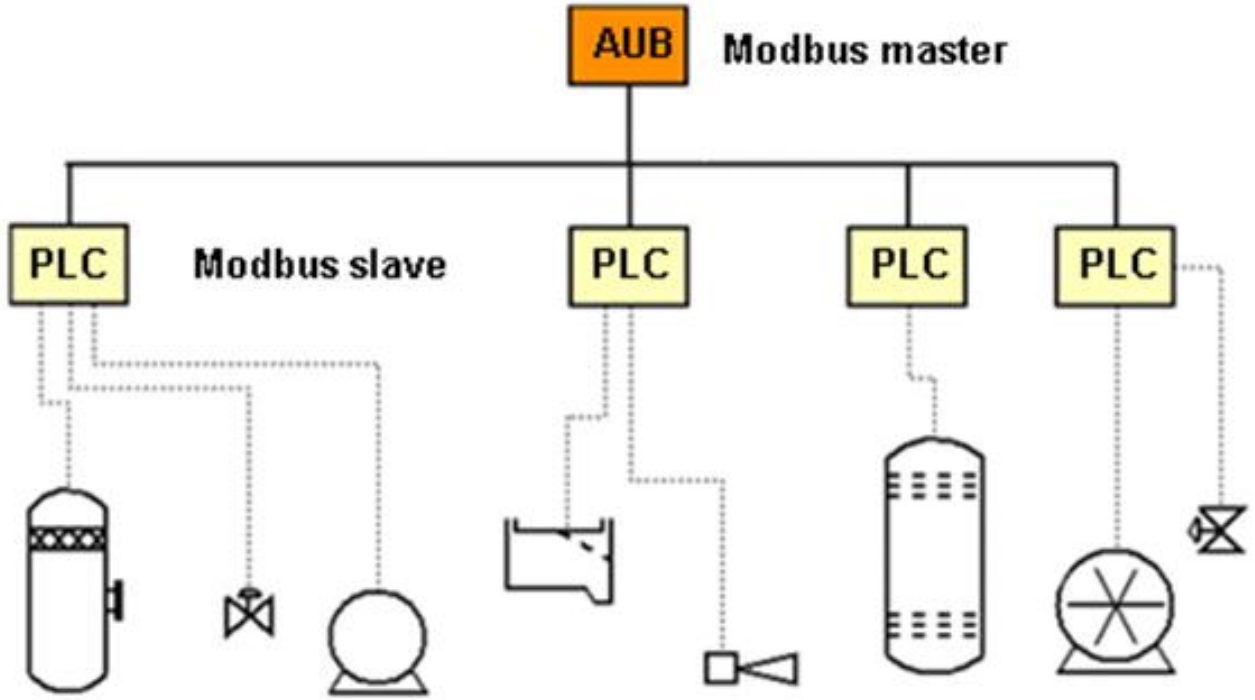
Osnove softvera sa kritičnim odzivom u elektroenergetskim sistemima

Modbus protokol

- Modbus je industrijski komunikacioni protokol
  - Nalazi se na aplikacivnom nivou komunikacionog stack-a (ISO OSI ; TCP/IP).
  - Klijent-server komunikacija između uređaja povezanih na različite vrste magistrala ili mreža.
- Primarno je telemetrijski protokol za komunikaciju između akviziciono upravljačkog bloka SCADA stanice (AUB) i procesnih kontrolera (PK)
  - AUB (eng. FEP – Front End Processor).
  - PK (eng. RTU – Remote Terminal Unit, PLC – Programmable Logic Controller).
- Modbus spada među najstarije i najrasprostranjenije od svih aktuelnih industrijskih protokola
  - popularnost duguje jasnom modelu podataka koji garantuje laku programsku implementaciju i visoku fleksibilnost u primeni.
  - predstavljen 1979. godine – *Modicon* (sada *Schneider Electric*).

Modbus protokol

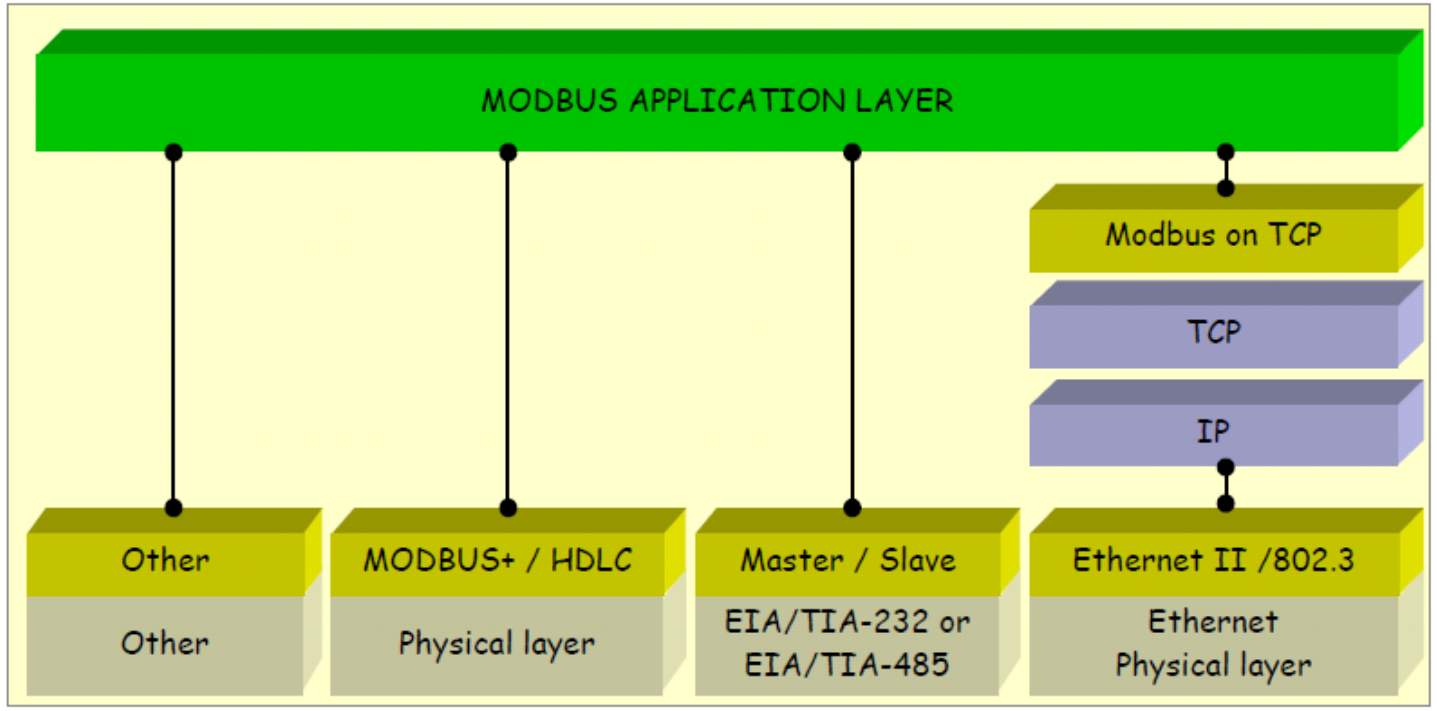
- Veza jedan master (AUB) na više slave uređaja (RTU/PLC)



Modbus sistem sa jednim master uređajem i više slave uređaja

Modbus protokol

- Prilagođen je za rad u lokalnoj mreži na TCP/IP komunikacionom stack-u.
  - Originalno razvijen za komunikaciju preko asinhrone serijske magistrale.



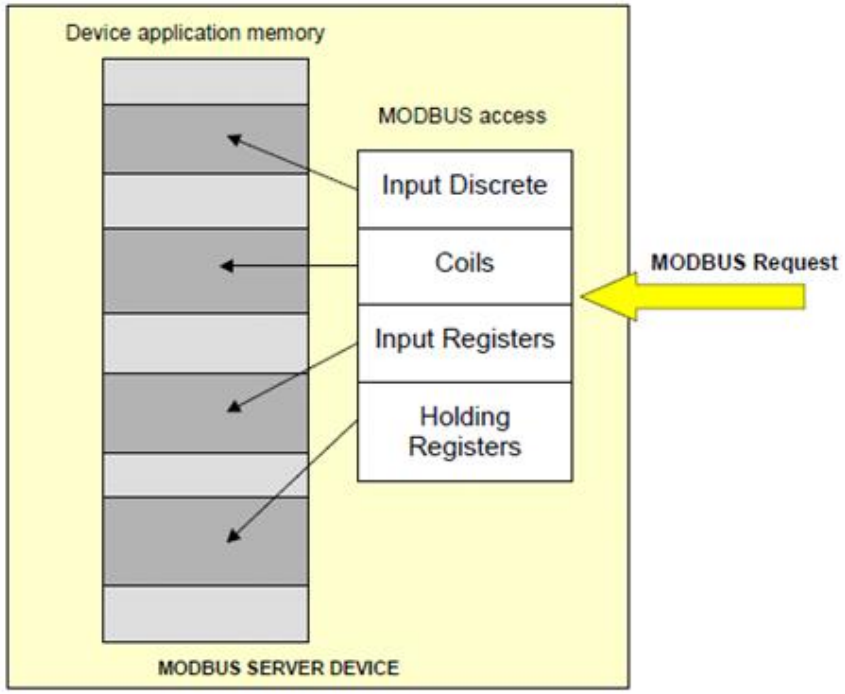
Pozicija Modbus-a u OSI referentnom modelu

Model podataka komunikacionog uređaja

- Model podataka predstavlja spoljnu sliku uređaja (komunikacionog entiteta) koji ga implementira
  - obuhvata sve spolja dostupne atribute i funkcije datog uređaja
  - suštinski definiše njegov logički automat vidljiv spoljnim klijentima
- Zato je model podataka u samom jezgru svakog protokola, jer određuje tipove podataka i aplikativne funkcije koje protokol podržava.
- Najbrži način za upoznavanje sa suštinom nekog protokola je analiza modela podataka iza njega.

Modbus model podataka

- Modbus model podataka je u uskoj vezi sa internom strukturom RTU/PLC uređaja.
  - obuhvata sve spolja dostupne atribute i funkcije datog uređaja
  - suštinski definiše njegov logički automat vidljiv spoljnim klijentima.
- Modbus logički model (*Modbus Register Map)*



- Četiri grupe podataka u RTU/PLC adresnom prostoru
  - dužine 1 ili 16 bita
- Reprezentuju najvažnije podatke
  - procesne ulaze i izlaze
- Dve osnovne operacije
  - čitanje (*read*) za sve tipove
  - upis (*write*) za izlaze

Modbus registarska mapa

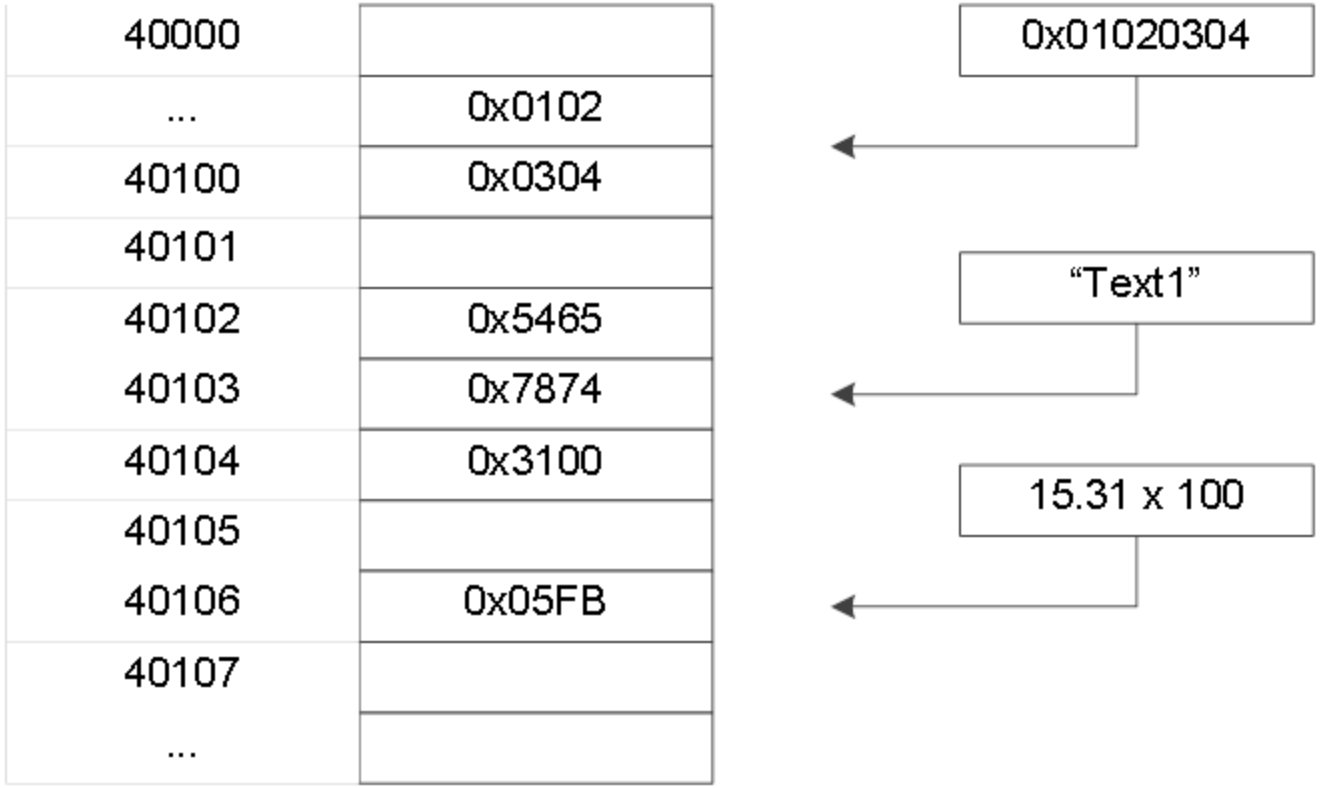
	Adresa		Oznaka	Dužina	Pristup	Opis
	00001	10000	Discrete Outputs (Coils)	1 bit	Read/Write	Digitalni izlazi
	10001	20000	Discrete Inputs	1 bit	Read	Digitalni ulazi
	30001	40000	Input Registers	16 bit	Read	Analogni ulazi i Brojači
	40001	50000	Holding Registers	16 bit	Read/Write	Analogni izlazi

Format Modbus podataka

- Binaran (sirov), neoznačen (unsigned), podrazumevano u *big endian rasporedu*.
- Prenos samo sirovih podataka je verovatno i najveća mana modbus modela podataka.
- U praksi često treba preneti neki 32-bitni integer, float ili string.
- Smeštanje 32-bitnih podataka, celobrojnih ili u pokretnom zarezu, može se uraditi korišćenjem dve susedne lokacije. Podatak se deli na dva dela od po 16 bita i smešta se na dve uzastopne adrese - u dva registra.
- Identični postupak se radi i kod smeštanja tekstualne (string) promenljive, jedino se koristi veći broj registara.

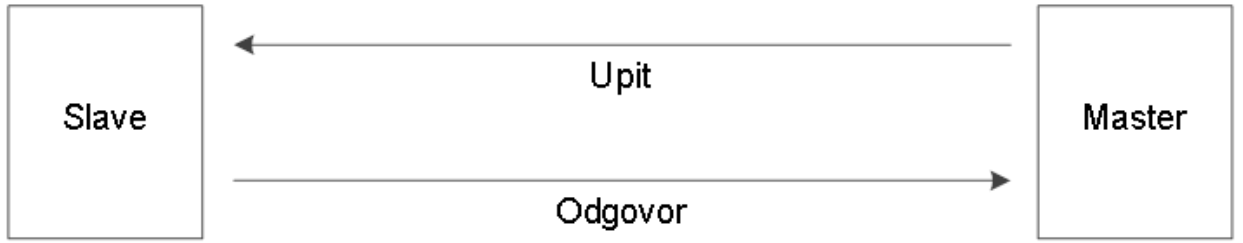
Primer smeštanja podataka dužih od 16 bita

- Svođenje broja u pokretnom zarezu na broj u fiksnom (*inferred decimal point*)
  - 16 bita dovoljno za smeštanje značajnog dela informacije



Modbus transakcija i opšti format poruke

- Tipičan nebalansirani protokol (upit/odgovor)
  - AUB je klijent (*master*) koji šalje upit
  - RTU/PLC je server (*slave*) koji odgovara
- Adrese uređaja:
  - master nema adresu
  - slave ima dodeljenu adresu 1-247
  - adresa 0 je rezervisana za *broadcast* poruke (primenjivo na multidrop veze – 1 master, više slave uređaja, 1 komunikaciona veza)

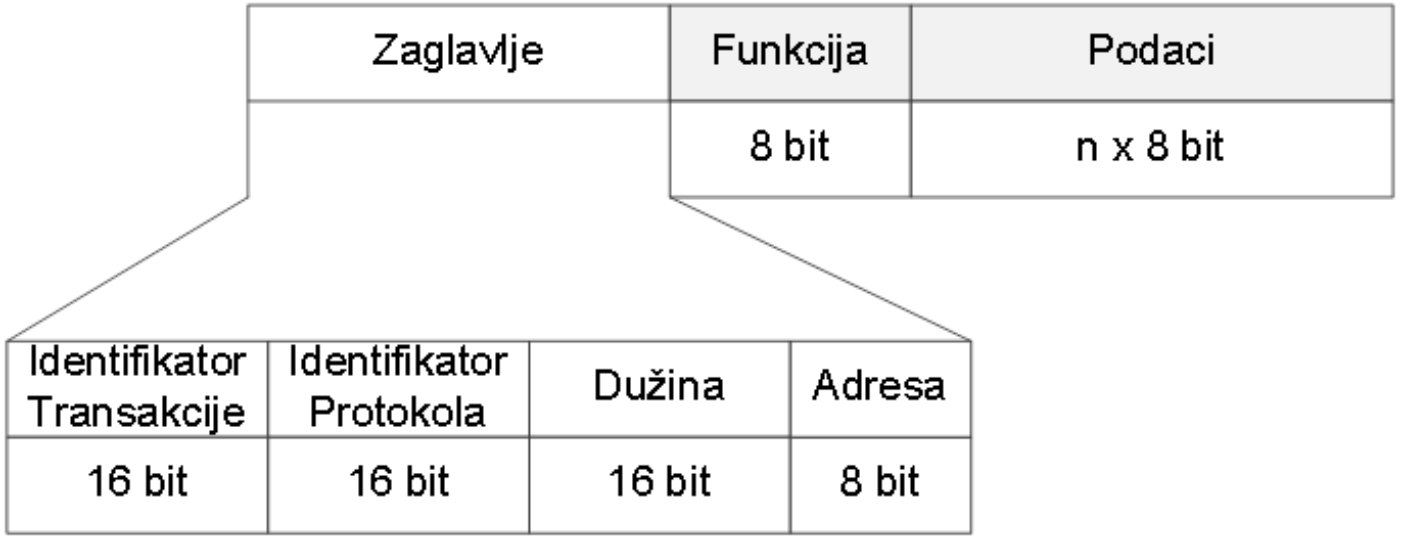


Format Modbus poruka

		Read Discrete Inputs	Read Coils	Write Single Coil	Read Input Register	Read Holding Registers	Write Single Register	Write Multiple Registers
Kod Funkcije		2	1	5	4	3	6	16
Upit	0	2	1	5	4	3	6	16
	1	Input Adr	Coil Adr	Coil Adr	InReg Adr	HoldReg Adr	HoldReg Adr	HoldReg Adr
	2							
	3	Num	Num	Value	Num	Num	Value	Num
	4							
	5							ByteCount
	6							Registers Values
	...							Num*2
Odgovor	0	2	1	5	4	3	6	16
	1	ByteCount	ByteCount	Coil Adr	ByteCount	ByteCount	HoldReg Adr	HoldReg Adr
	2							
	3	Inputs	Coils	Value	Input Registers	Holding Registers	Value	Num
	4							
	5	Num/8	Num/8		Num*2	Num*2		
	6							
	7							
	...							

Modbus TCP Format

- TCP/IP varijanta RTU verzije Modbus protokola
  - TCP je *stream* protokol sa pouzdanim prenosom
  - Više klijenata (mastera) mogu imati istovremeni pristup jednom serveru, što može biti korisno (multidrop veza)



Modbus TCP zaglavlje

Fields	Length	Description	Client	Server
Transaction Identifier	2 bytes	Identification of a MODBUS request/response transaction	Initialized by the client	Copied by the server from the request to the response
Protocol Identifier	2 bytes	0 = MODBUS protocol	Initialized by the client	Copied by the server from the request to the response
Length	2 bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (response)
Unit identifier	1 byte	Identification of a remote slave connected on a serial line or on other buses	Initialized by the client	Copied by the server from the request to the response

Kod funkcije

Ime funkcije	Identifikator funkcije	Tip registra
Read Coils	0x01	Digitalni izlaz
Read Discrete Inputs	0x02	Digitalni ulaz
Read Holding Registers	0x03	Analogni izlaz
Read Input Registers	0x04	Analogni ulaz
Write Single Coil	0x05	Digitalni izlaz
Write Single Register	0x06	Analogni izlaz

Format Modbus poruka – Read

	Request			Response			Error		Comment
Function	**Function code 1 byte**	**Starting address 2 byte**	**Quantity to read 2 byte**	**Function code 1 byte**	**Byte count N 1 byte**	**Status / Value N byte**	**Function code 1 byte**	**Exception code 1 byte**	
Read coils	**0x01**	0x0000 to 0xFFFF	1 to 2000 (0x7D0)	**0x01**	**N**	Status of coils	Function code + 0x80	01 or 02 or 03 or 04	**N = Quantity of points / 8** *If remainder is different of 0 then* **N = N + 1**
Read discrete inputs	**0x02**	0x0000 to 0xFFFF	1 to 2000 (0x7D0)	**0x02**	**N**	Status of discrete inputs	Function code + 0x80	01 or 02 or 03 or 04	
Read holding registers	**0x03**	0x0000 to 0xFFFF	1 to 125 (0x7D)	**0x03**	**N**	Value of holding registers	Function code + 0x80	01 or 02 or 03 or 04	**N = Quantity of registers * 2**
Read Input registers	**0x04**	0x0000 to 0xFFFF	1 to 125 (0x7D)	**0x04**	**N**	Value of input registers	Function code + 0x80	01 or 02 or 03 or 04	

Format Modbus poruka – Write

	Request			Response			Error	
Function	**Function code** **1 byte**	**Output / Register address** **2 byte**	**Output / Register value** **2 byte**	**Function code** **1 byte**	**Output / Register address** **2 byte**	**Output / Register value** **2 byte**	**Function code** **1 byte**	**Exception code** **1 byte**
Write single coil	**0x05**	0x0000 to 0xFFFF	**0x0000** = OFF **0xFF00** = ON	**0x05**	0x0000 to 0xFFFF	**0x0000** = OFF **0xFF00** = ON	Function code + 0x80	01 or 02 or 03 or 04
Write single register	**0x06**	0x0000 to 0xFFFF	0x0000 to 0xFFFF	**0x06**	0x0000 to 0xFFFF	0x0000 to 0xFFFF	Function code + 0x80	01 or 02 or 03 or 04

Modbus ASCII/RTU Format

- Implementacija protokola Modbus nad serijskim asinhronim kanalom
- *Modbus ASCII* – tekstualan format
- *Modbus RTU* – binaran format sveden na minimalan obim podataka zbog sporog prenosa
- Provera ispravnog prenosa kontrolnom LRC/CRC sekvencom

Start	Adresa	Funkcija	Podaci	LRC	Kraj
:	2 char	2 char	n x 2 char	2 char	CR LF

Pauza	Adresa	Funkcija	Podaci	CRC
3.5-4 char	8 bit	8 bit	n x 8 bit	16 bit

Modbus UDP Format

- Modbus nema svoju standardnu UDP implementaciju
- U dScada-i je serijski kanal prosto zamenjen UDP kanalom.
- Prednost UDP prenosa je praktičnost, ali velika mana je nepouzdanost isporuke

Adresa	Funkcija	Podaci	CRC
8 bit	8 bit	n x 8 bit	16 bit

Literatura

1. Софтвер са критичним одзивом – Пројектовање SCADA система, Бранислав Атлагић, 2015.

2. MODBUS APPLICATION PROTOCOL SPECIFICATION - V1.1b3

http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

3. MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE - V1.0b

http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf