Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroničke sustave i obradbu informacija

Mara Živčić
0130018531

Sustavi za praćenje i vođenje procesa
Seminarski rad

MODBUS

Zagreb, 27. svibnja 2007.

UVOD

Modbus je komunikacijski protokol izvorno zamišljen za upotrebu s programabilnim logičkim kontrolerima (PLC), no zbog svoje jednostavnosti i lake dostupnosti danas je to praktički industrijski standard primijenjiv u raznim elektroničkim uređajima.

Modbus protokol zasnovan je na serijskoj komunikaciji između *master* jedinice i jedne ili više (do 247) *slave* jedinica spojenih u istu mrežu, izravno ili pomoću modema. Svaka *slave* jedinica ima svoju adresu i samo jedinica kojoj je naredba poslana reagira na naredbu. Izuzetak od ovog pravila su broadcast naredbe, koje se odnose na sve jedinice i na koje nije potrebno odgovarati.

Standardni *Modbus* uređaji upotrebljavaju RS-232C kompatibilnu vezu koja definira pinove konektora, kabel, razine signala, brzinu prijenosa i provjeru pariteta, no *Modbus* protokol je moguće primijenjivati i na drugačijim mrežama, primjerice na Ethernetu. Sam *Modbus* protokol određuje strukturu poruke koju uređaji mogu prepoznati bez obzira na tip mreže te način na koji će pojedini uređaj prepoznati svoju adresu, pročitati njemu namijenjenu poruku i na nju primjereno reagirati. Pojedina poruka sastoji se od adresnog dijela, funkcijskog koda, podatkovnog dijela i dijela koji se odnosi na provjeru ispravnosti poruke.

VARIJANTE PRIJENOSA PODATAKA

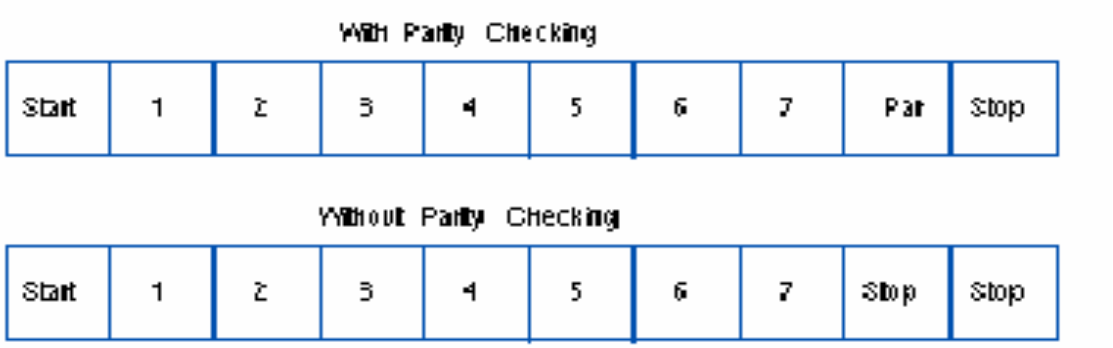
Modbus protokol definira dvije varijante prijenosa podataka (ASCII i RTU) koje opisuju način pakiranja podataka u poruku i njihovog dekodiranja. Željenu varijantu prijenosa potrebno je odabrati pri inicijalizaciji mreže i ona mora biti ista za sve uređaje spojene na jednu mrežu.

ASCII

Kod ASCII prijenosa podataka svaki bajt se šalje kao dva ASCII znaka, od kojih svaki predstavlja jednu heksadecimalnu znamenku (0..9, A..F). Pri tome se svaki ASCII znak pakira u riječ na sljedeći način:

- 1 start bit
- 7 bitova podataka poredanih po rastućoj bitnosti (LSB)
- 1 paritetni bit (ako je pri inicijalizaciji odabrana opcija provjere pariteta)
- 1 stop bit (ako je pri inicijalizaciji odabrana opcija provjere pariteta), odnosno 2 stop bita (ako pri inicijalizaciji nije odabrana opcija provjere pariteta).

Slika 1 prikazuje redoslijed bitova u jednoj riječi kodiranoj za ASCII varijantu prijenosa podataka sa, odnosno bez provjere pariteta:



Slika 1. Redoslijed bitova riječi (ASCII) sa i bez pariteta

Za provjeru ispravnosti poruke kod ASCII prijenosa upotrebljava se LRC metoda (longitudinal redundancy check), a znakovi se mogu slati s do 1 sekundom razmaka bez da uređaj to detektira kao grešku. U svrhu uokvirivanja poruke kao prvi znak šalje se dvotočka (:) (heksadecimalno 3A), dok se kao zadnji znak šalje CRLF (carriage return-line feed – ASCII 0D, heksadecimalno 0A). Uređaji na mreži kontinuirano nadziru mrežnu sabirnicu i čekaju da se pojavi znak za početak poruke. Nakon pojave dvotočke, svi uređaji pročitaju adresni dio poruke da bi provjerili odnosi li se poslana poruka na njih. Struktura poruke prikazana je na slici 2.

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS CRLF

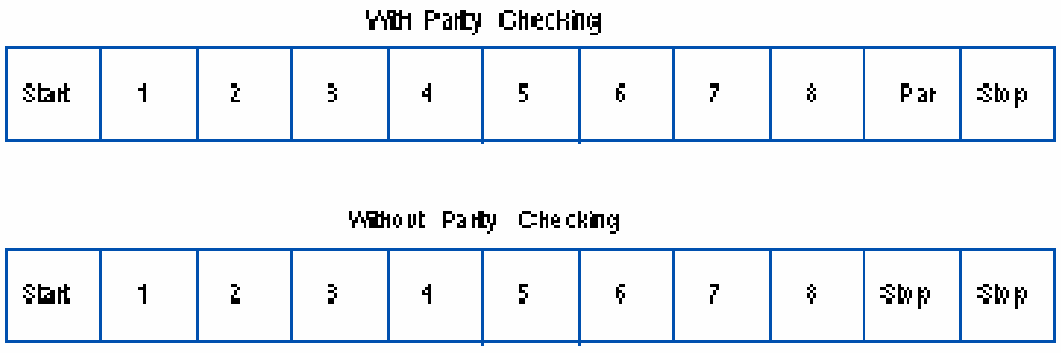
Slika 2. Struktura poruke (ASCII)

RTU

RTU (remote terminal unit) prijenos podataka izravno prenosi heksadecimalne znamenke bez njihove konverzije u ASCII kod. Pri tome jedan bajt sadrži dvije heksadecimalne znamenke koje se šalju kao jedan znak u jednoj riječi, što za rezultat ima veću korisnost pri istoj brzini prijenosa. Pakiranje podataka u riječ provodi se na sljedeći način:

- 1 start bit
- 8 bitova podataka poredanih po rastućoj bitnosti (LSB)
- 1 paritetni bit (ako je pri inicijalizaciji odabrana opcija provjere pariteta)
- 1 stop bit (ako je pri inicijalizaciji odabrana opcija provjere pariteta), odnosno 2 stop bita (ako pri inicijalizaciji nije odabrana opcija provjere pariteta).

Na slici 3 prikazan je redoslijed bitova jedne riječi kodirane za RTU prijenos podataka sa, odnosno bez provjere pariteta:



Slika 3. Redoslijed bitova riječi (RTU) sa i bez pariteta

Za provjeru ispravnosti poruke upotrebljava se CRC metoda (cyclic redundancy check), a znakovi se moraju slati kontinuirano. Početak i kraj poruke označavaju se pauzama u komunikaciji u trajanju od barem 3.5 znakovnih intervala (obično se uzima višekratnik znakovnog intervala pri brzini prijenosa definiranoj na mreži, na slici dolje označeno kao T1-T2-T3-T4). Ako se u toku prijenosa poruke pojavi pauza dulja od 1.5 znakovnih intervala, primatelj će izbrisati nedovršenu poruku i pretpostaviti da sljedeći bajt predstavlja adresni dio nove poruke. Osim toga, ako nova poruka počne za manje od 3.5 znakovnih intervala nakon završetka prethodne, primatelj je neće prepoznati kao novu poruku, već kao nastavak prethodne, što će uzrokovati grešku. I u ovoj varijanti uređaji konstantno nadziru mrežnu sabirnicu, a kad se poruka pojavi, svi dekodiraju adresni dio da bi ustanovili odnosi li se ista na njih. Struktura tipične RTU poruke prikazana je na slici 4.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	n x 8 BITS	16 BITS	T1-T2-T3-T4

Slika 4. Struktura poruke (RTU)

STRUKTURA PORUKE

Svaka poruka sastoji se od istih dijelova: okvira, adresnog dijela, funkcijskog koda, podatkovnog dijela i dijela za provjeru ispravnosti poruke.

OKVIR

Okvir, koji služi za označavanje početka i kraja poruke, razlikuje se u ovisnosti o varijanti prijenosa. Kako je već gore navedeno, kod ASCII prijenosa on se sastoji od dvotočke na početku i CRLF znaka na kraju poruke, dok se kod RTU prijenosa kao okvir upotrebljavaju pauze određenog trajanja u prijenosu podataka.

ADRESNI DIO

Adresni dio sastoji se od dva ASCII znaka, odnosno 8 bitova. *Slave* jedinice imaju adrese u rasponu od 1 do 247, dok adresa nula označava broadcast naredbu. *Master* u adresni dio poruke postavlja adresu na koju se poruka upućuje, a *slave* jedinica u odgovoru u adresni dio postavlja svoju adresu.

FUNKCIJSKI DIO

Funkcijski dio sastoji se od dva ASCII znaka, odnosno 8 bitova. On sadrži funkcijski kod poruke, u rasponu od 0 do 255, koji *slave* jedinici prenosi naredbu koju treba izvršiti. *Slave* jedinica u odgovoru vraća funkcijski kod nepromijenjen ako je poruka primljena, dok u slučaju nemogućnosti izvršenja naredbe zbog greške u sadržaju poruke *slave* jedinica vraća funkcijski kod kojemu je najviši bit postavljen u 1. Primjeri naredbi koje *master* može poslati *slave* jedinici su provjera statusa ulaznih pinova, čitanje sadržaja registara,

dijagnostička provjera *slave* jedinice, pisanje u registre, promjena stanja izlaznih pinova, te učitavanje, snimanje ili provjera programa samog kontrolera.

PODATKOVNI DIO

Podatkovni dio sastavlja se od parova heksadecimalnih znamenaka (0x00 do 0xFF), pri čemu one mogu, ovisno o varijanti prijenosa, predstavljati par ASCII znakova ili jedan RTU znak. *Master* u ovaj dio poruke upisuje adrese registara ili vanjskih pinova kojima treba pristupiti, broj traženih podataka te, ako *master* šalje podatke koje treba nekuda upisati, broj bajtova podataka i potom same podatke. Za neke naredbe *slave* jedinica ne treba dodatne podatke, već joj je dovoljan funkcijski kod, pa neke poruke ne sadrže podatkovni dio. *Slave* jedinica u podatkovnom dijelu odgovora šalje tražene podatke, odnosno kod greške ako iz nekog razloga nije u mogućnosti izvršiti poslanu naredbu.

DIO ZA PROVJERU ISPRAVNOSTI PORUKE

Provjera ispravnosti poruke je obavezna i provodi se neovisno o odabiru paritetne provjere znakova. Ovisno o varijanti prijenosa, ispravnost poruke provjerava se pomoću LRC (za ASCII) odnosno CRC (za RTU) proračuna. Proračun provodi *master* jedinica pri slanju poruke i upisuje rezultat na kraj poruke (prije znaka koji označava kraj). *Slave* jedinica tijekom primitka poruke ponovno proračunava LRC odnosno CRC i uspoređuje rezultat s onim koji je *master* poslao. Ako se rezultati razlikuju, znači da je došlo do greške tijekom prijenosa podataka.

LRC se primijenjuje na poruku bez dvotočke i CRLF znaka, a sastoji se od 2 ASCII znaka (1 bajt). LRC se proračunava zbrajanjem

bajtova poruke uz odbacivanje prijenosa, nakon čega se nad rezultatom provede operacija dvojnog komplementiranja.

CRC se primijenjuje na cijelu poruku, ali se u proračun uzima u obzir samo osam podatkovnih bitova svake riječi (bez start i stop te paritetnih bitova). CRC se sastoji od dva bajta koji se postavljaju na kraj poruke, pri čemu se prvo upisuje niži, a potom viši bajt. Proračun CRC provodi se na sljedeći način:

1. 16-bitni registar napuni se jedinicama,
2. nad 8-bitnim znakom i sadržajem registra provede se operacija ekskluzivno ili (XOR),
3. rezultat se pomakne za jedan bit u smjeru najnižeg bita (LSB),
4. u najviši bit (MSB) upiše se nula,
5. ako je LSB = 1, provede se operacija XOR nad sadržajem registra i nekom prethodno definiranom vrijednosti

Koraci 2 do 5 ponavljaju se osam puta za svaki bajt podataka, pri čemu se operacija ekskluzivno ili za svaki novi bajt provodi sa trenutnim sadržajem registra, a krajnji rezultat, kad se obradi cijela poruka, upisuje se u poruku kao CRC.

Potrebno je još navesti da u slučaju pojave greške tijekom prijenosa podataka adresirana *slave* jedinica neće reagirati na poslanu poruku. U tu svrhu *master* ima konfigurirano vrijeme (timeout) koje treba čekati na odgovor *slave* jedinice (dovoljno dugo da *slave* jedinica stigne reagirati); nakon isteka tog vremena, *master* jedinica će prekinuti komunikaciju. Isto će se dogoditi i ako *master* adresira nepostojeću *slave* jedinicu.

KOMUNIKACIJA

Modbus se zasniva na serijskoj komunikaciji između *master* i *slave* jedinica. Primjer takve komunikacije (upita i odgovora) prikazan je na slici 5.

Query

Field Name	Example (hex)	ASCII Characters	RTU 8-Bit Field
Header		: (colon)	None
Slave Address	06	0 6	0000 0110
Function	03	0 3	0000 0011
Starting Address Hi	00	0 0	0000 0000
Starting Address Lo	6B	6 B	0110 1011
No. of Registers Hi	00	0 0	0000 0000
No. of Registers Lo	03	0 3	0000 0011
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
Total Bytes		17	8

Response

Field Name	Example (hex)	ASCII Characters	RTU 8-Bit Field
Header		: (colon)	None
Slave Address	06	0 6	0000 0110
Function	03	0 3	0000 0011
Byte Count	06	0 6	0000 0110
Data Hi	02	0 2	0000 0010
Data Lo	2B	2 B	0010 1011
Data Hi	00	0 0	0000 0000
Data Lo	00	0 0	0000 0000
Data Hi	00	0 0	0000 0000
Data Lo	63	6 3	0110 0011
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
Total Bytes		23	11

Slika 5. Primjer komunikacije *master* - *slave*

Prva tablica prikazuje poruku koju šalje *master*, u ovom slučaju zahtjev *slave* jedinici broj 6 da pročita sadržaje tri registra (40108 –

40110). U prvom stupcu navedeni su nazivi pojedinih dijelova poruke, redom:

- početak
- adresa *slave* jedinice
- funkcijski kod
- gornji bajt početne adrese
- donji bajt početne adrese
- gornji bajt broja registara koje treba pročitati
- donji bajt broja registara koje treba pročitati
- provjera ispravnosti
- kraj
- ukupni broj bajtova u poruci.

Drugi stupac sadrži heksadecimalni prikaz sadržaja pojedinih dijelova poruke, treći stupac sadrži ASCII varijantu naredbi, a u četvrtom se nalazi RTU varijanta.

U drugoj tablici prikazan je odgovor prozvane *slave* jedinice. Prvi stupac ponovno sadrži nazive dijelova poruke:

- početak
- adresa *slave* jedinice
- funkcijski kod
- broj bajtova
- gornji bajt podatka
- donji bajt podatka
- gornji bajt podatka
- donji bajt podatka
- gornji bajt podatka

- donji bajt podatka
- provjera ispravnosti
- kraj
- ukupni broj bajtova.

Ostali stupci imaju isto značenje kao i u prvoj tablici.

ZAKLJUČAK

Modbus je protokol za serijsku komunikaciju industrijskih elektroničkih uređaja. Zbog relativno niskog stupnja korisnosti (veliki broj kontrolnih bitova u odnosu na broj podatkovnih bitova) brzina prijenosa podataka nije visoka, no njegova jednostavnost i dostupnost učinili su ga vrlo popularnim i raširenim. *Modbus* protokol danas je praktički industrijski standard.

LITERATURA

1. http://en.wikipedia.org/
2. http://www.modicon.com/