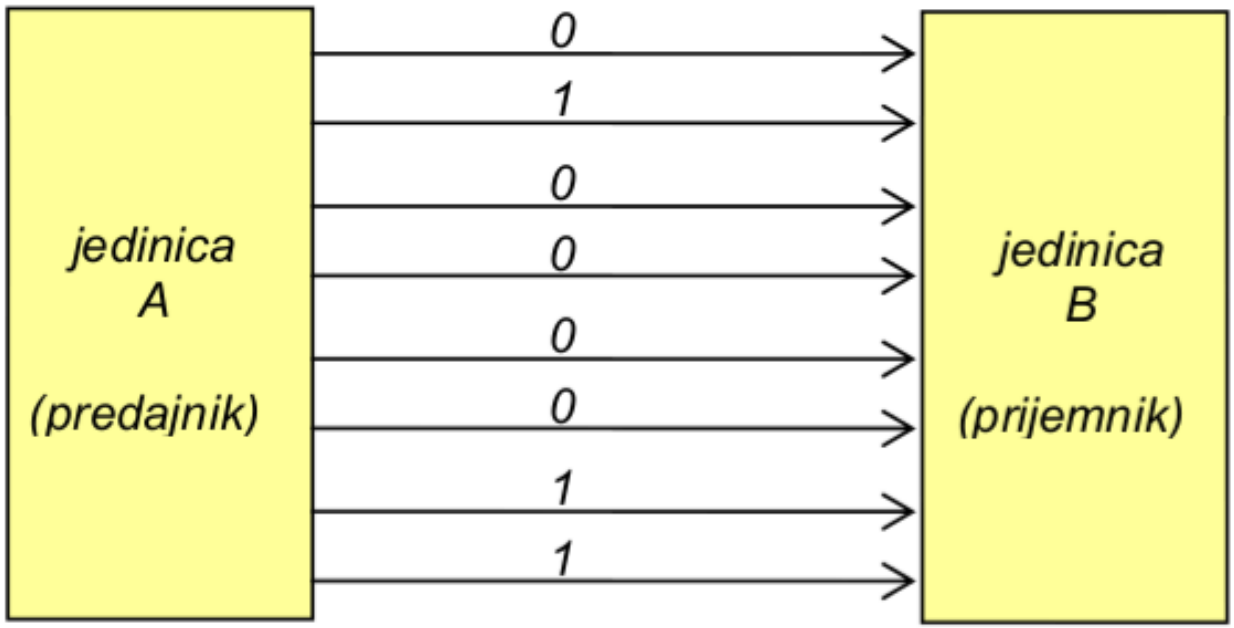
Serijski komunikacioni protokoli

dr Predrag Teodorovic

Fakultet Tehničkih Nauka, Novi Sad

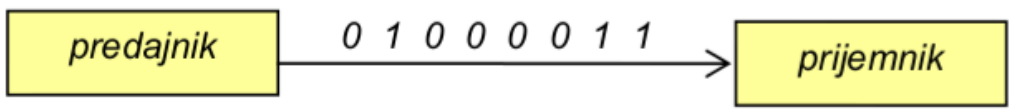
May 24, 2018

Paralelan prenos podataka?

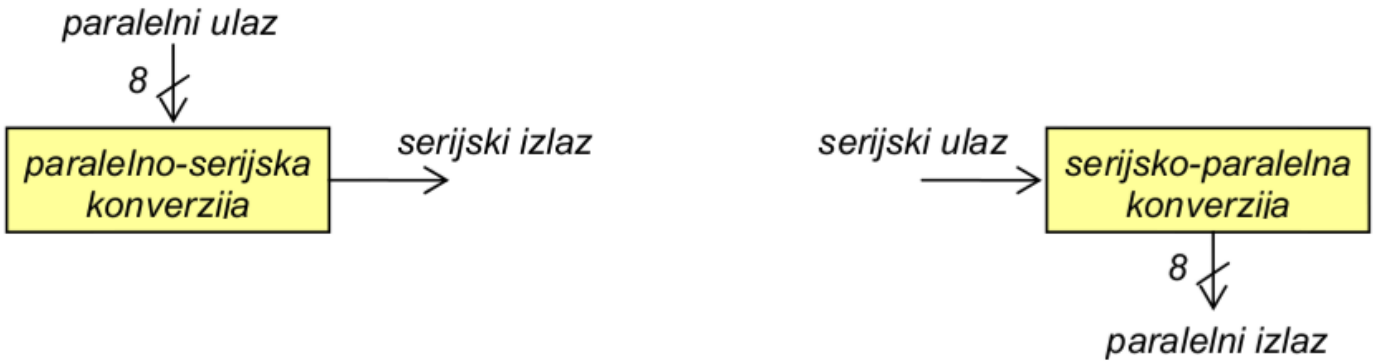


Kad je ovo zgodno a kad nije?

- Pogodan za velike brzine prenosa i relativno kratka rastojanja izmedu jedinica
- Na većim rastojanjima cena provodnika i ovo nije praktično
- Zbog toga je alternativa da se koristi samo jedan provodnik za prenos podataka



Šta je ovde problem?

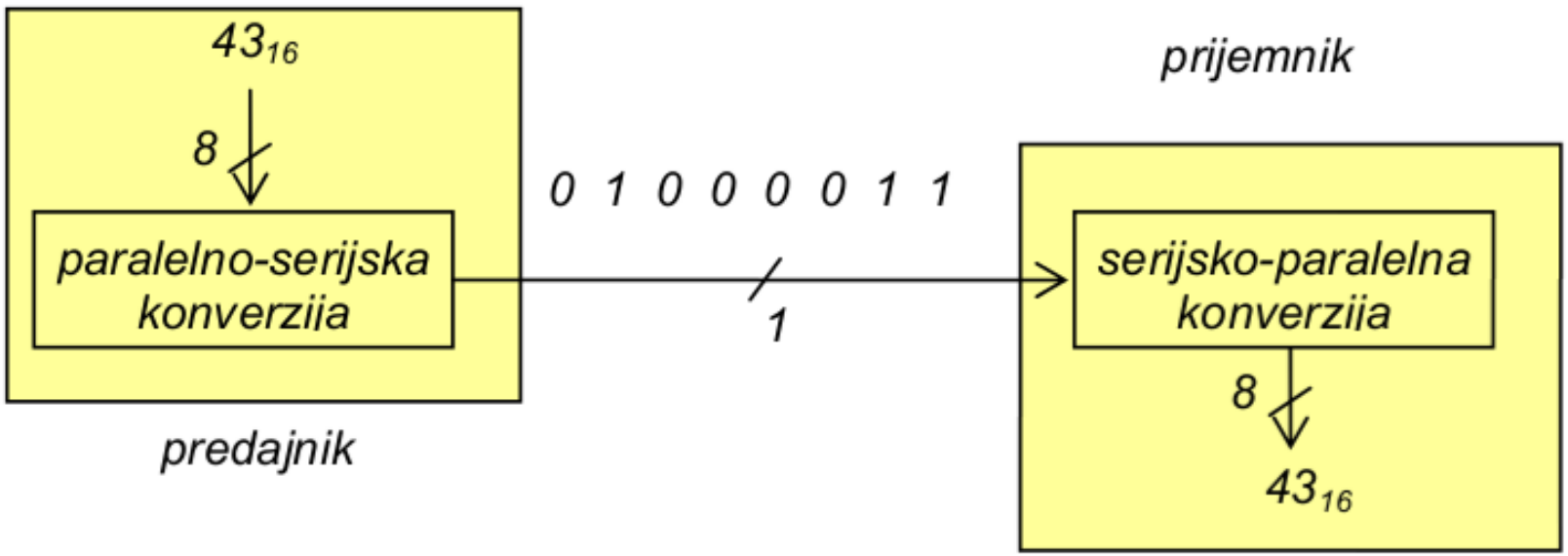


- Očigledno je sporiji od paralelnog prenosta (***n*** puta za ***n*** provodnika)
- Zahteva paralelno-serijsku, odnosno serijsko-paralelnu konverziju

Da li znamo kako bismo to uradili?!?

- Na jednoj strani paralelni ulaz i serijski izlaz
- Na drugoj strani serijski ulaz i paralelni izlaz

Koncept serijskog prenosa

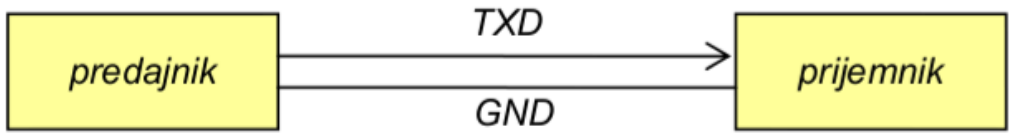


Šta određuje tip veze i način prenosa?

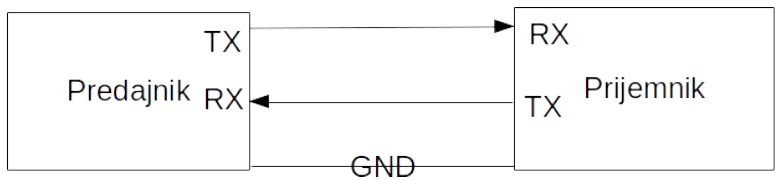
- Da li je serijska veza jednosmerna ili dvosmerna i ako je veza dvosmerna, da li može da se istovremeno obavlja predaja i prijem?
- Kako se vrši sinhronizacija rada predajnika i prijemnika?
- Da li postoji i kako se obavlja kontrola ispravnosti prenosa podataka?
- Da li se podaci prenose reč po reč ili u blokovima?

Smer veze

- Jednosmerna (simpleks)



- Polu-dvosmerna (polu-dupleks)
- Dvosmerna (dupleks)



Sinhronizacija

- Sinhronizacija rada između predajnika i prijemnika je važna jer je osnova za ispravan serijski prenos podataka
- Sinhronizacija pre svega podrazumeva da predajnik i prijemnik rade na istoj brzini prenosa, odnosno da je brzina prenosa izražena u broju bita u sekundi ista na predajnoj i prijemnoj strani
- Sinhronizacija rada može da se vrši i razmenom upravljačkih signala, odnosno *rukovanjem* (engleski *handshaking*) izmedu predajnika i prijemnika
- Naravno, rukovanje zahteva dodatne provodnike za prenos upravljačkih signala izmedu predajnika i prijemnika

Kontrola ispravnosti

- Može se vršiti na više načina
- Najjednostavnije dodavanjem bita parnosti na predajnoj strani
- Na prijemnoj strani, prijemnik na osnovu poruke i primljenog bita parnosti može (eventualno) da prepozna da je došlo do greške u slanju, ali ne i da ispravi grešku

Slanje podataka

- Podaci mogu da se prenose reč po reč ili u blokovima u kojima se nalazi više reči
- Prenos reč po reč obično se naziva asinhroni prenos, kod koga se za svaku reč prvo šalje početni (start) a na kraju reči završni (stop) bit
- Kod sinhronog prenosa obično se prenosi blok podataka u kome postoji početna i završna reč, a pored serijskih podataka prenosi se i sinhronizacioni signal od predajnika do prijemnika

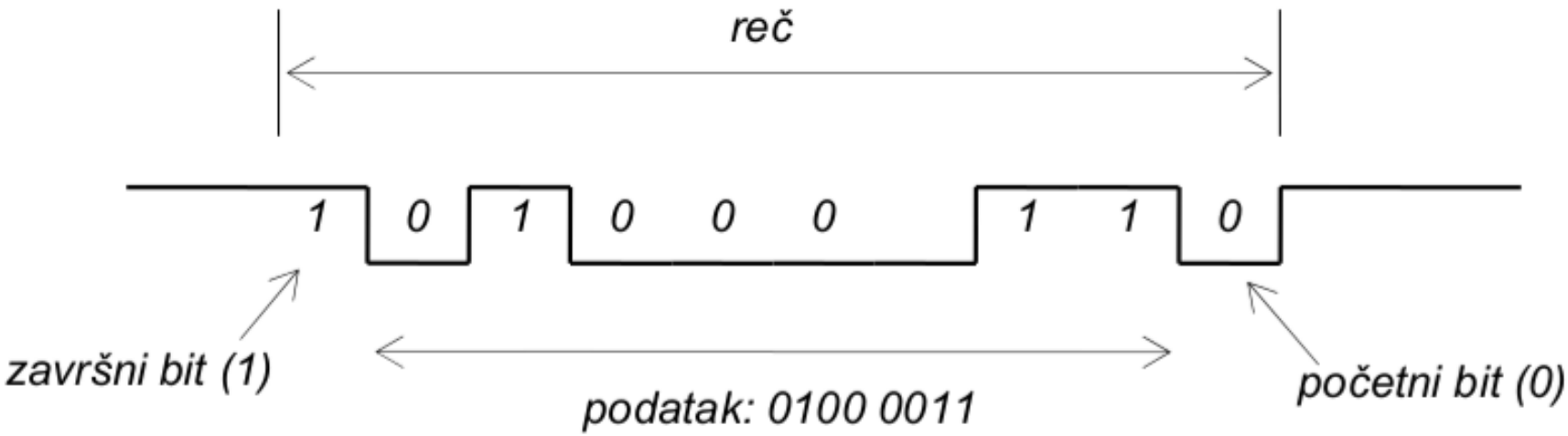
Asinhrona serijska komunikacija

- Podrazumeva komunikaciju izmedu dva UART kontrolera (engleski *Universal Asynchronous Receiver Transmitter*), uređaja koji podržavaju prijem i predaju serijskih podataka u asinhronom načinu rada
- RX linija se koristi za prijem podataka, dok se podaci šalju putem TX linije
- Kod predaje, UART kontroler prihvata podatke u paralelnom obliku od mikroprocesora, pretvara ih u serijski oblik, ubacuje dodatne bite i u serijskom obliku šalje poruku na izlaznu liniju
- Sličan postupak je kod prijema podataka - podatak se iz serijskog prevodi u paralelni oblik i u tom obliku predaje mikroprocesoru
- Kod asinhronog prenosa podaci se prenose reč po reč (engleski *character*)

Kako se obavlja komunikacija?

- Kada je linija neaktivna, na njoj je visok naponski nivo, tj pre početka serijskog prenosa i izmedu prenosa dve uzastopne reči, provodnik za serijski prenos drži se u stanju logičke 1
- Opadajuća ivica na komunikacionoj liniji je signal prijemnoj strani da predajna strana počinje sa slanjem reči
- Reč koja se šalje sastoji se od:
  - početnog bita (engleski start) koji je logička 0
  - 7 ili 8 bita podataka (ovo je zapravo informacija koja se prenosi)
  - Opcionog bita za proveru parnosti
  - 1 do 2 završna (stop) bita koji su uvek logičke “1”

Kako to izgleda?



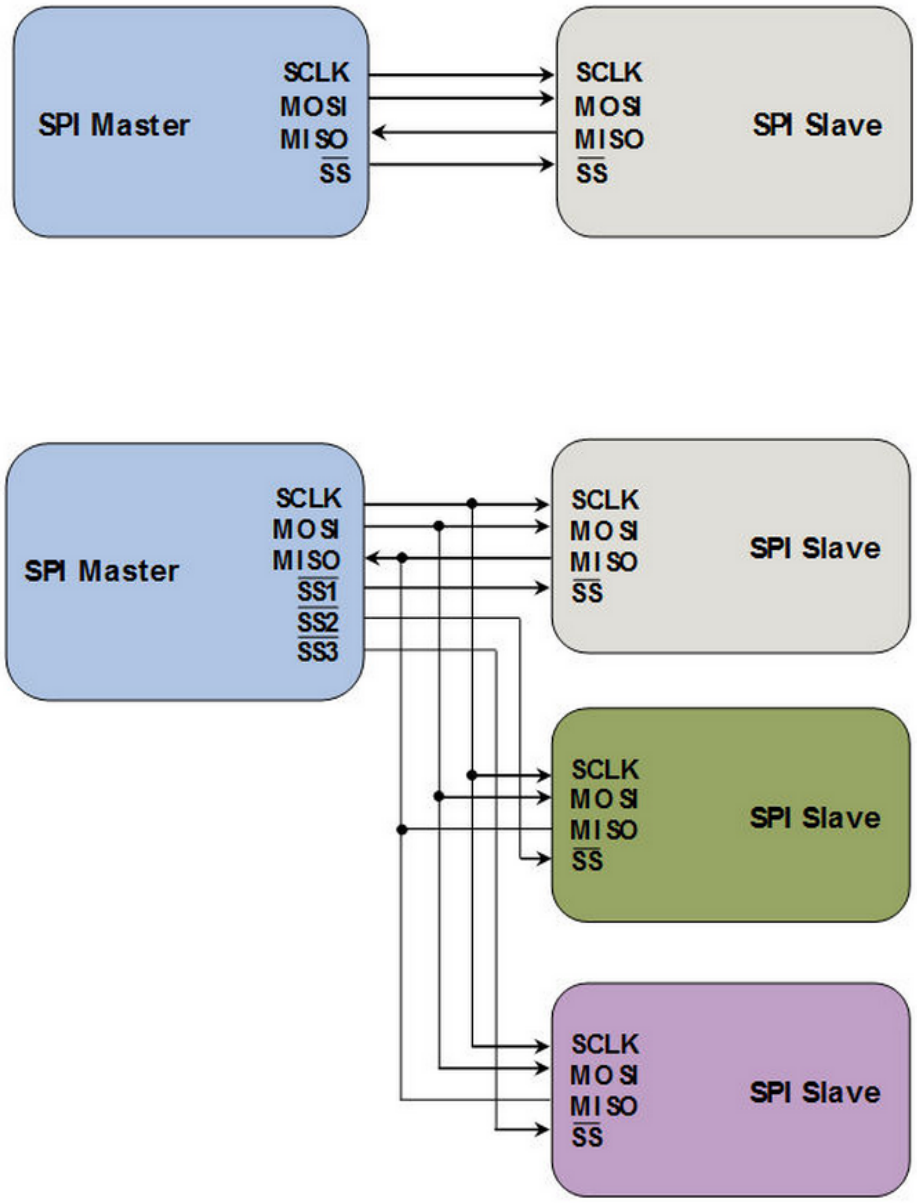
Šta je potrebno da bi komunikacija bila uspešna?

- Konfigurisati prijemnik i predajnik na identičan način:
  - Brzine serijske komunikacije (eng. *baud rate*) izražen u broju bita u sekundi. Tipične brzine asinhrone serijske komunikacije su 9600b/s, 19200b/s, 38400b/s, 57600b/s i 115200b/s
  - Broj bita podataka 7 ili 8
  - Da li se koristi bit parnosti i ako se koristi da li će on biti setovan za paran ili neparan broj jedinica u okviru bita podataka
  - Broj stop bita koji se koristi (1 ili 2)
- Kao što je već naglašeno ranije, u paraleli sa prijemom poruke, moguće je istovremeno slati poruku korišćenjem nezavisne komunikacione linije TX

SPI komunikacioni protokol

- Naziv SPI je skraćenica od *Serial Peripheral Interface* i namenjen je komunikaciji izmedu integrisanih kola, kao i za relativno sporu komunikaciju sa periferijskim jedinicama koje se nalaze na istoj štampanoj ploči
- SPI protokol je osmislila i patentirala Motorola
- Inicijalno SPI je osmišljen kako bi povezao mikrokontroler sa periferijama koristeći 4 žice i prilično je jednostavan
- SPI komunikacija podrazumeva da uvek u sistemu postoji jedinstveni ***master***, dok je moguće postojanje većeg broja ***slave*** uređaja

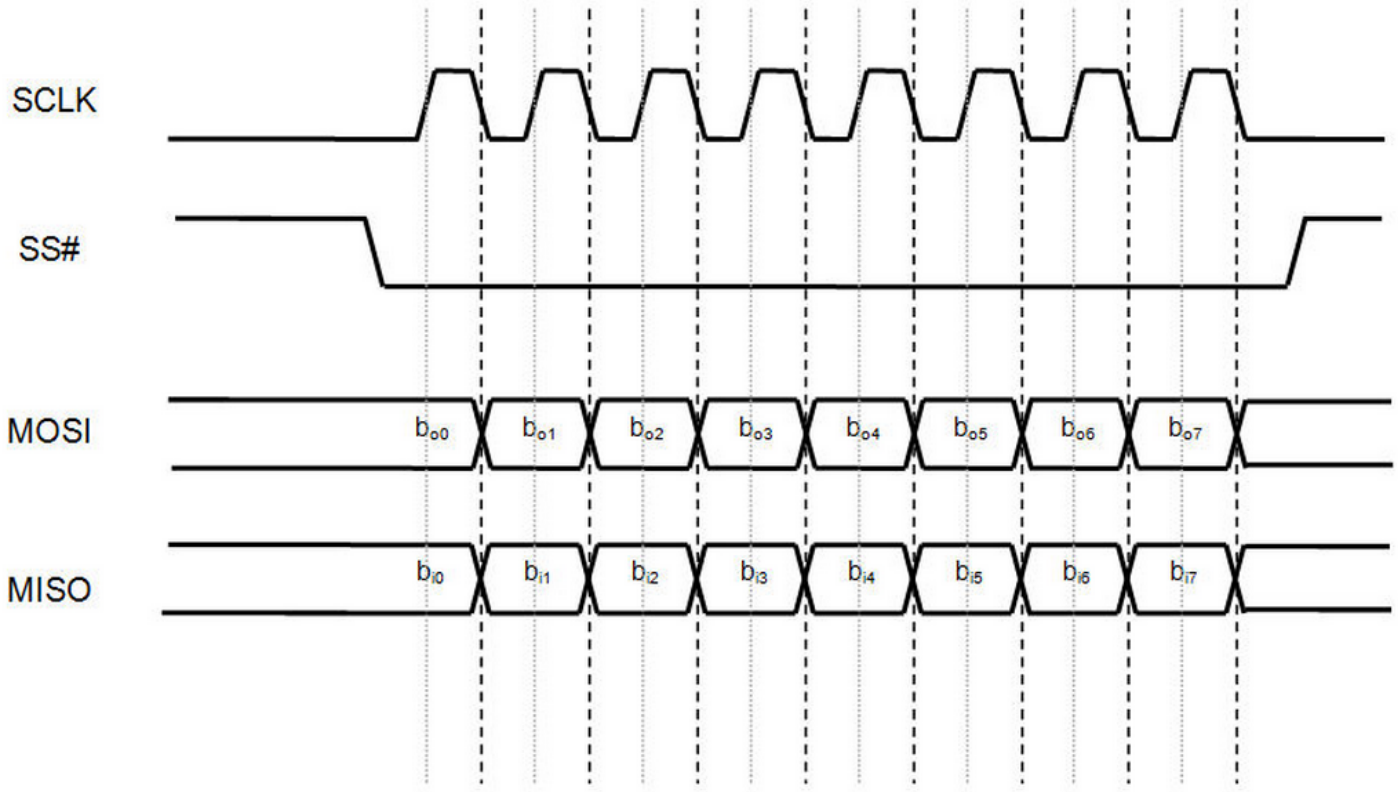
Povezivanje 1-1 i 1-više



SPI signali

1. SCLK signal je taktni signal koji uvek generiše master i vodi se ka svim slave uređajima. Svi SPI signali su sinhronizovani u odnosu na ovaj signal
2. SSn signal predstavlja odabir slave-a (engl. *Slave Select*) služi da odabere jedan od povezanih slave uređaja u datom trenutku, sa kojim će master da komunicira
3. MOSI (Master Out Slave In) je linija podataka od mastera ka svim slave uređajima
4. MISO (Master In Slave Out) je linija podataka od slave uređaja ka master uređaju

Signali na SPI magistrali



Karekteristike SPI komunikacionog interfejsa

1. SPI protokol ne definiše maksimalnu brzinu komunikacije (za razliku od asinhrone serijske komunikacije, kod SPI je sve sinhronizovano sa SCLK signalom)
2. SPI ne omogućava nikakvu potvrdu prijema podatka, kao ni kontrolu toka komunikacije
3. SPI master čak nije svestan ni da li je slave uređaj uopšte prisutan na SPI magistrali

I^2C komunikacioni protokol

- I2C je skraćenica od Inter Integrated Circuit (ili IIC)
- Razvijen je od strane kompanije Philips još davne 1972. godine
- Osnovna namena ovog protokola jeste bila sprega mikroprocesora sa integrisanim kolima u okviru TV uređaja uz postojanje samo dve žice
- Maksimalna brzina komunikacije u početku je bila 100kbps (*kilo bits per second*), nakon promene specifikacije 1995. godine, podignuta je na 400kbps, a od 1998. maksimalna brzina je 3.4Mbps (*mega bits per second*)
- I2C omogućava postojanje više master uređaja na I2C magistrali

Dve signalne linije

1. SDA (Serial DAta) za podatke
2. SCL (Serial CLock) za takt

Ne postoji potreba za dodatnim signalom koji bi omogućio CS funkcionalnost kao u slučaju SPI komunikacije

Kako izgleda protokol?

- Teorijski, proizvoljan broj master uređaja i proizvoljan broj slave uređaja mogu da se nadu na I2C magistrali, jer protokol definiše:
  - 7-bitne adrese slave uređaja pri čemu svaki uređaj povezan na I2C magistralu mora imati svoju jedinstvenu adresu
  - podatke podeljene u bajtove (8 bita informacije)
  - nekolicinu kontrolnih bita koji odreduju početak i kraj komunikacije, kao i smer toka podataka i potvrdu prijema

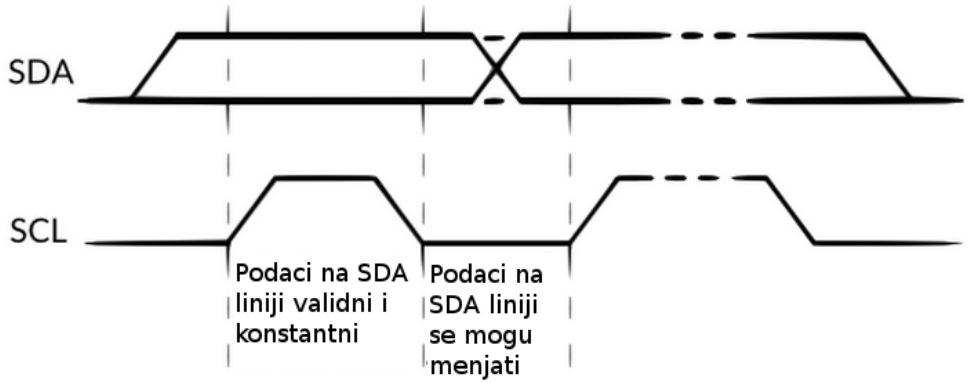
Tok razmene podataka

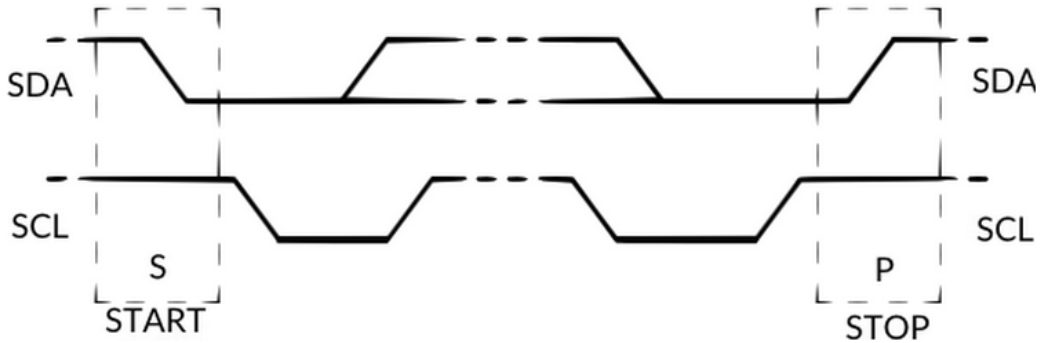
- Na početku komunikacije master uvek generiše START uslov
- Ovo daje signal svim uređajima koji se nalaze na I2C magistrali da krenu da osluškuju poruke
- Nakon START uslova, master generiše adresu
- Svi uređaji proveravaju adresu i porede sa svojom adresom
- Samo uređaj koji ima datu adresu nastavlja da komunicira sa master uređajem, dok ostali ignorišu ostatak poruke, sve dok ne stigne STOP uslov na magistrali

Tok razmene podataka.. nastavak

- Uređaj koji ima traženu adresu mora da generiše ACK (*acknowledge* signal) kako bi master znao da je uređaj sa traženom adresom pronađen
- Nakon prijema ACK signala master nastavlja sa slanjem ili čitanjem slave uređaja i kada završi sa tim generiše STOP signal
- U slučaju da master želi da šalje podatke slave uređaju nakon 7 bita adrese generiše se logička nula, dok se logička jedinica generiše u slučaju kada je potrebno čitati podatke poslate od strane slave-a
- Nakon svakog primljenog (poslatog) bajta master (slave) generiše ACK signal (logička nula na SDA liniji) nakon čega se nastavlja sa sledećim bajtom (slanje ili prijem). Prilikom prijema višebajtnog podatka od strane slave-a, master generiše NACK (logičko 1 na SDA liniji) nakon prijema poslednjeg bajta i time daje do znanju slave-u da je transfer završen

Koja su pravila? Šta je START, a šta STOP?





Format poruke na primeru slanja/prijema 2 bajta podataka..

START	Adr	R/$\overline{W}$	ACK	Data	ACK	Data	(N)ACK	STOP
1bit	7bit	1bit	1bit	8bit	1bit	8bit	1bit	1bit

Slanje 0x10 i 0x20 ka slave-u na adresi 0x7F

M	M	M	S	M	S	M	S	M
START	Adr	R/$\overline{W}$	ACK	Data	ACK	Data	ACK	STOP
1 bit	0x7F	0	0	0x10	0	0x20	0	1 bit

Prijem 0x30 i 0x40 od slave-a na adresi 0x1A

M	M	M	S	S	M	S	M	M
START	Adr	R/$\overline{W}$	ACK	Data	ACK	Data	NACK	STOP
1 bit	0x1A	1	0	0x30	0	0x40	1	1 bit