

The background is a dark green field filled with a complex network of lighter green lines that resemble a printed circuit board (PCB) layout. These lines are interconnected by small circles, some of which are yellow. On the left side, there is a vertical column of yellow dots. In the bottom right corner, there is a small cluster of yellow dots arranged in a grid-like pattern. The overall aesthetic is technical and digital.

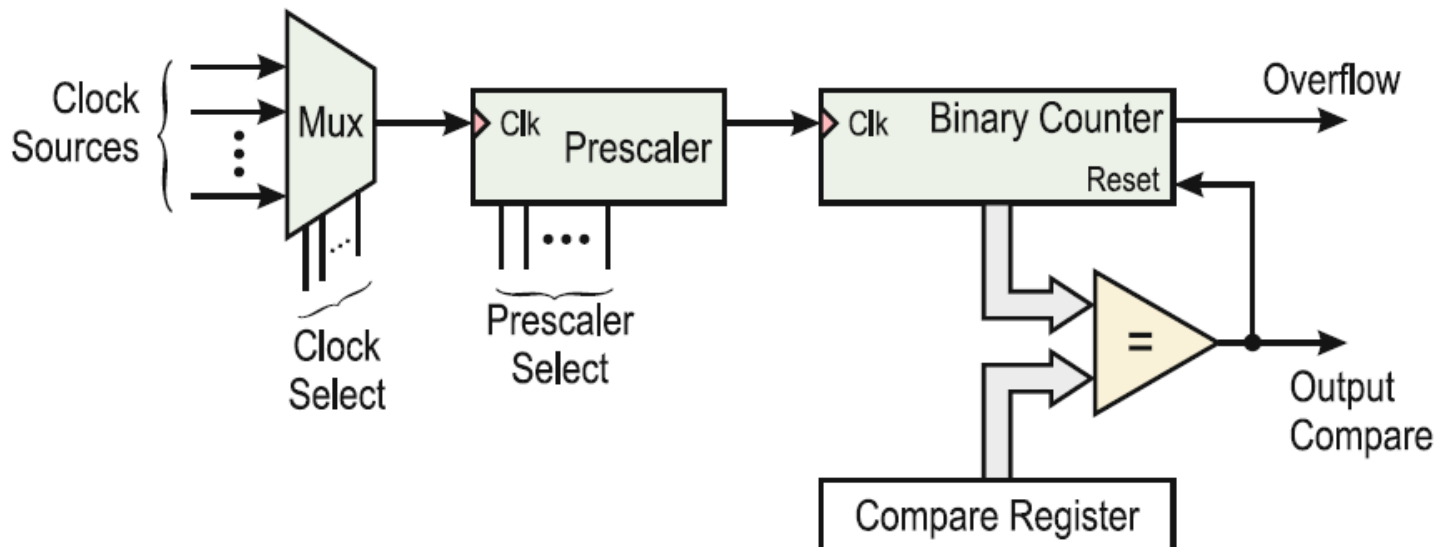
Tajmeri i brojači

Tajmeri - osnovne karakteristike

- Mikrokontroleri obično imaju jedan ili **više tajmerskih modula** kao periferni jedinice koje su integrisane na istom čipu sa procesorom. Tajmeri su obično visoko konfigurabilni, kako bi ih korisnik lako mogao prilagoditi svojoj konkretnoj potrebi.
- Tipično se koriste u jednoj od dve osnovne konfiguracije:
 - **Merenje proteklog vremena** (tajmer, eng. *timer*)
 - **Brojanje impulsa** koji se dovode na određeni pin mikrokontrolera (brojač, engl. *counter*)
- Tajmeri po pravilu u određenim uslovima mogu da generišu **zahteve za prekidom** (engl. *interrupt request*).
- Neki od tajmera imaju mogućnost čuvanja podatka o tačnom vremenskom trenutku pojave odgovarajućeg događaja u sistemu i koriste se za rad u sistemima sa realnim vremenom.
- Koristeći tajmere i pravilno ih konfigurirajući, projektant embeded sistema može u značajnoj meri rasteriti centralni procesor dugačke liste zadataka koji uključuju merenje i manipulaciju vremenom.
- Neke od najčešćih primena tajmera i brojača su:
 - **Interval tajmeri**
 - **Brojači događaja**
 - **Satovi realnog vremena** (eng. *Real Time Clock - RTC*)
 - **Generatori za impulsno-širinsku modulaciju** (eng. *Pulse Width Modulation*)
 - **Baud rate generatori**
 - **Watchdog tajmeri**

Hardverska struktura tajmera

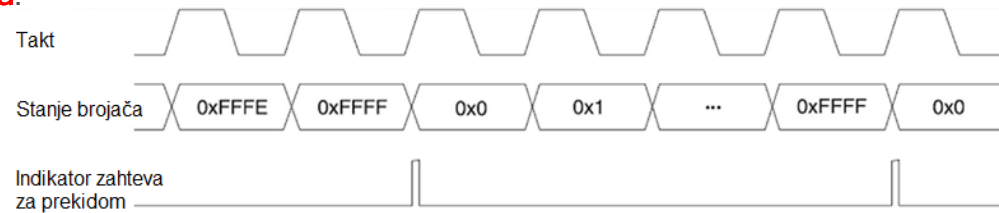
- Osnovne komponente u okviru strukture većine tajmera su:
 - **Selektor signala takta** (multiplekser) koji dozvoljava izbor jednog od više mogućih **signala takta**
 - **Preskaler** obezbeđuje mogućnost **deljenja učestanosti signala takta** (usporavanja signala takta) pre nego što on uđe u brojač
 - **N-bitni brojač** obezbeđuje osnovnu brojačku funkciju
 - **N-bitni komparatorski registar** omogućuje da se definiše maksimalna vrednost koju brojač može dostići
 - **N-bitni komparator** omogućava detekciju trenutka kada brojač dostiže vrednost koja je smeštena u komparatorskom registru. Po pravilu, dostizanjem ove vrednosti automatski se vrši **resetovanje brojača**, aktiviranje prekida ukoliko tajmer to podržava, što je najčešći slučaj i postavljanje odgovarajućeg indikatora.



Vremenski dijagrami tajmera - normalni i komparatorski mod

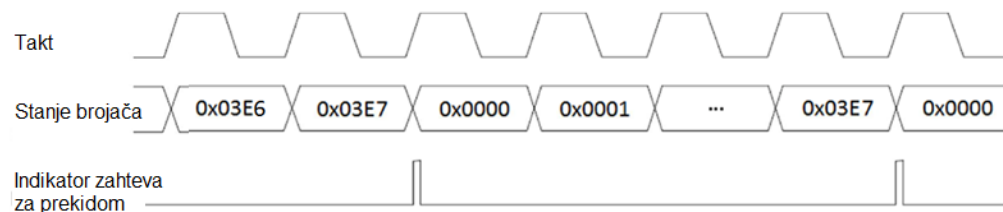
NORMALNI MOD

- Obzirom da je brojački registar realizovan kao **N-bitni registar**, brojač broji od vrednosti 0 do vrednosti 2^N-1 . Kada dostigne vrednost 2^N-1 , brojač generiše signal prekoračenja opsega (engl. **overflow**) i započinje brojanje ponovo od vrednosti 0. **Overflow** signal se obično može konfigurisati da izazove pojavu zahteva za prekidom.
- Na slici je prikazan rad jednog 16-bitnog tajmera, gde je takođe prikazano ponašanje signala zahteva za prekidom koji se aktivira kada se u brojaču desi prekoračenje opsega (prelazak sa vrednosti 65535 na vrednost 0). Ovaj događaj se naziva još i **preticanjem tajmera**.



KOMPARATORSKI MOD (Tajmer sa skraćenim modulom brojanja)

- U većini aplikacija javlja se potreba za limitiranjem maksimalne vrednosti koju brojač može dostići. U takvim slučajevima potrebno je postojanje komparatorskog registra i komparatora kao sastavnih delova tajmera.
- Korišćenjem ova dva modula može se generisati signal koji će biti **aktivan kada brojač dostigne vrednost koja je specificirana u komparatorskom registru**. Ovaj signal može zatim generisati odgovarajući zahtev za prekidom i resetovati tajmer, kao što je prikazano na slici, u slučaju da je vrednost komparatorskog registra jednaka 999 (0x03E7).

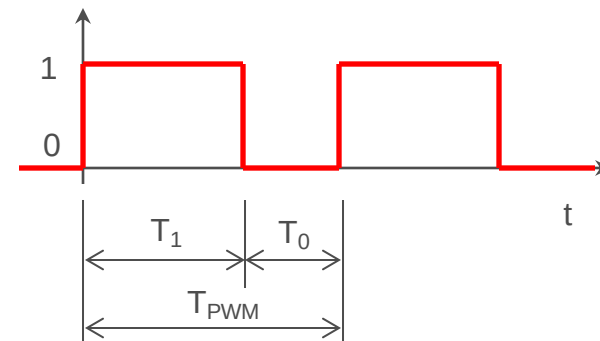


Tajmeri i impulsno - širinska modulacija (PWM)

- Impulsno - širinska modulacija (engl. **PWM = Pulse Width Modulation**) je često korišćena tehnika digitalnog upravljanja uređajima kao što su prekidački naponski regulatori, LED diode, DC motori, pojačavači klase D i sl.
- Osnovna ideja: generisati periodični signal gde se u toku jedne periode stanje menja sa 1 na 0 u tačno određenim vremenskim razmacima. Prosečna (usrednjena) vrednost napona zavisi odnosa trajanja logičke jedinice i logičke nule:
- Za PWM signal definiše se **faktor ispune D** (engl. Duty) na sledeći način:

$$D = \frac{T_1}{T_{PWM}} = \frac{T_1}{T_1 + T_0}$$

$$D \in [0,1], \text{ odnosno } [0\%, 100\%]$$

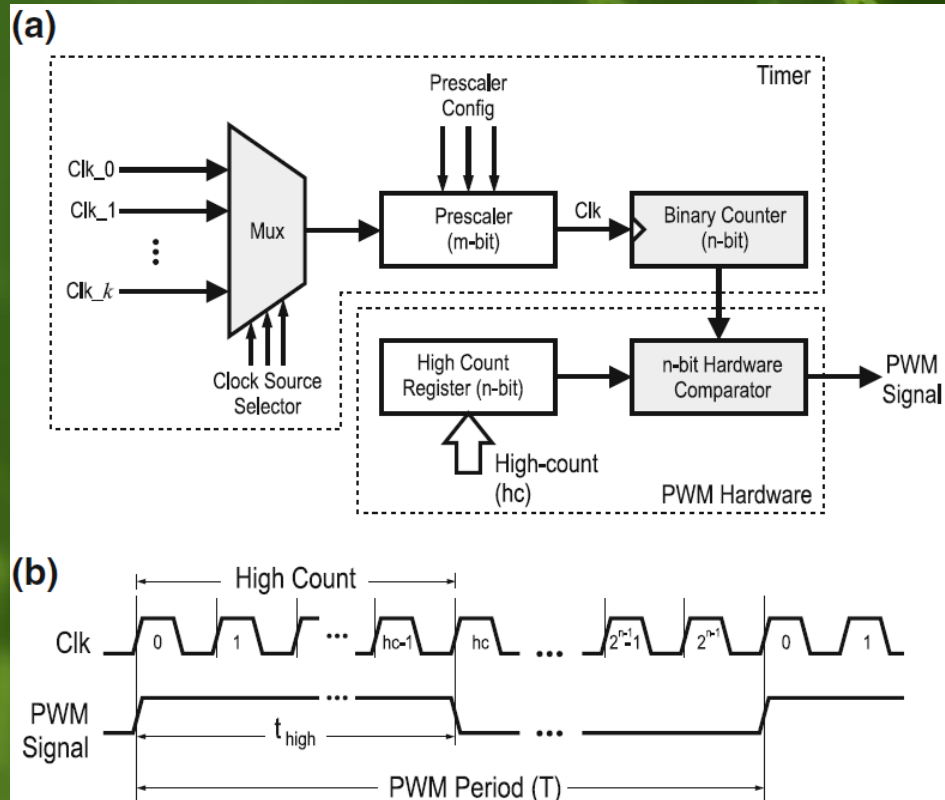


- PWM signal je moguće generisati **hardverski** korišćenjem **tajmerskog modula**. Tajmer se pušta da broji od nule do maksimalne vrednosti ($2^n - 1$ za n-bitni tajmer). U **komparatorski registar** se upisuje **vrednost m** između 0 i $2^n - 1$ koja određuje faktor ispune: dok je vrednost brojača manja od m, stanje izlaznog pina je 1, nakon čega se stanje menja na logičku 0. U ovom slučaju faktor ispune iznosi:

$$D = \frac{m}{2^n}$$

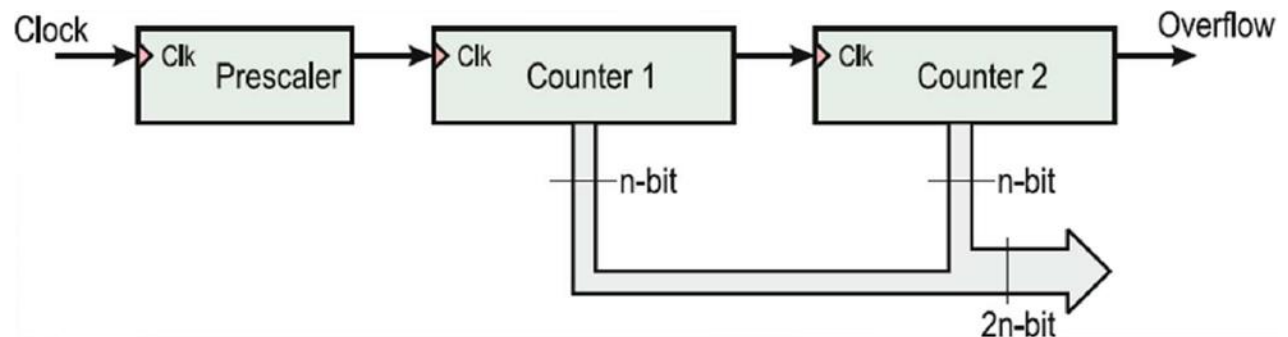
PWM generator

- Struktura tipičnog PWM modula prikazana je na slici (a)
- PWM modul sastoji se iz N -bitnog tajmera (sa selektorom klok signala i preskalerom) čiji se izlaz upoređuje sa sadržajem HC registra (*High Count*)
- Dok je vrednost tajmera manja od HC vrednosti, vrednost PWM signala jednaka je jedinici, u protivnom jednaka je nuli
- Vremenski dijagram PWM signala prikazan je na slici (b)
- **Broj bitova (N) unutar brojačkog registra tajmera određuje rezoluciju PWM signala**, dok vrednost HC registra određuje širinu impulsa (faktor ispune, t_{high}/T) PWM signala u datoj periodi
- Učestanost PWM signala može se kontrolisati pomoću preskalera i selektora klok signala



Proširenje opsega brojanja tajmera

- U pojedinim aplikacijama može se pojaviti situacija da je **maksimalni broj događaja** koji je potrebno detektovati veći od opsega brojanja brojačkog registra tajmera koji je na raspolaganju. Navedeni problem može se rešiti na tri načina:
 1. Proširenjem brojačkog opsega tajmera korišćenjem **softverske varijable** - vrednost varijable se uvećava za jedan svaki put kada tajmer dostigne svoj maksimum.
 2. Proširenjem brojačkog opsega tajmera **kaskadnim vezivanjem** više tajmerskih modula.
 3. Korišćenjem preskalera: Kada je prescaler konfigurisan da deli ulazni takt sa faktorom P, tajmer efektivno broji pojavu svakog P-tog događaja.

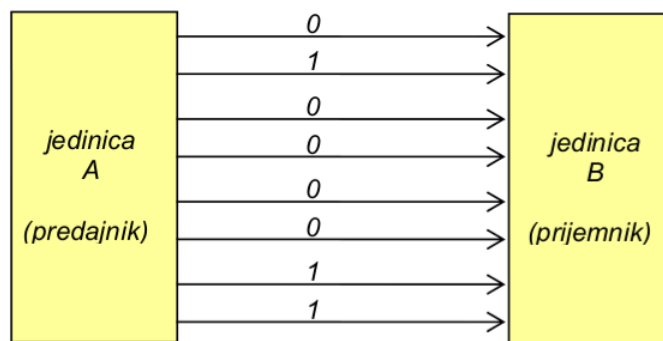


The background is a dark green field filled with a complex network of lighter green lines that resemble a printed circuit board (PCB) layout. These lines are interconnected by small circles, some of which are highlighted in a bright yellow color. The overall effect is a sense of digital connectivity and electronic architecture.

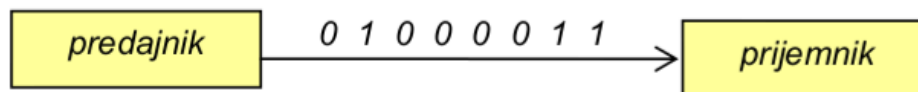
Serijska komunikacija

Paralelni i serijski prenos podataka

- Osnovni principi prenosa podataka između različitih uređaja u digitalnom sistemu:
 - Paralelni prenos** - istovremeno se prenosi n bita podataka, posredstvom n različitih linija za prenos
 - Serijski prenos** - koristi se samo jedna linija za prenos, pri čemu se informacija prenosi tokom n uzastopnih vremenskih intervala
- Paralelni prenos je generalno brži od serijskog pošto se simultano prenosi veći broj bita, ali zahteva veći broj provodnih linija i nije pogodan za upotrebu na većim rastojanjima pošto veći broj provodnika znatno poskupljuje realizaciju.



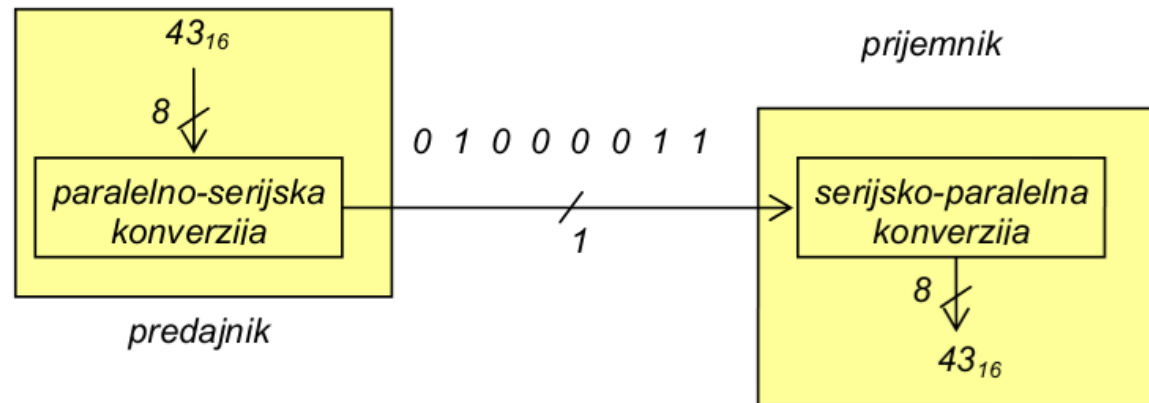
Paralelni prenos podataka



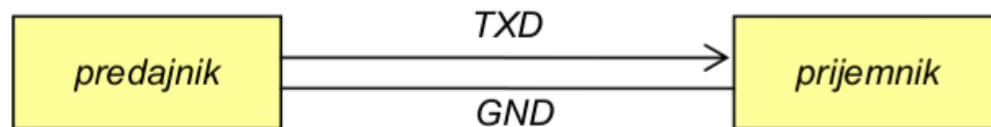
Serijski prenos podataka

Principi serijskog prenosa

- Prilikom serijskog prenosa podataka, na predajnoj strani mora da se obavi **paralelno-serijska**, a na prijemnoj strani **serijsko-paralelna konverzija**.
- Paralelno-serijska konverzija vrši se pomoću **pomeračkog registra sa mogućnošću paralelnog upisa**, a **serijsko-paralelna konverzija** pomoću **pomeračkog registra sa paralelnim izlazima**.



- Po pitanju smera komunikacije, serijski prenos može biti:
 - **Simplex** - Komunikacija je jednosmerna, od predajnika ka prijemniku. Za ostvarivanje veze potrebna je jedna linija za prenos podataka i referentna linija (masa).
 - **Half duplex** - Komunikacija se obavlja u oba smera, posredstvom iste linije. Pri tome se tokom vremena uloge predajnika i prijemnika raspodeljuju među uređajima koji komuniciraju u skladu sa korišćenim komunikacionim protokolom.
 - **Full duplex** - Puna dvosmerna komunikacija, gde je moguće da oba uređaja istovremeno vrše prijem i slanje podataka. U ovom slučaju su pored mase potrebne još dve linije za prenos, po jedna za oba smera.



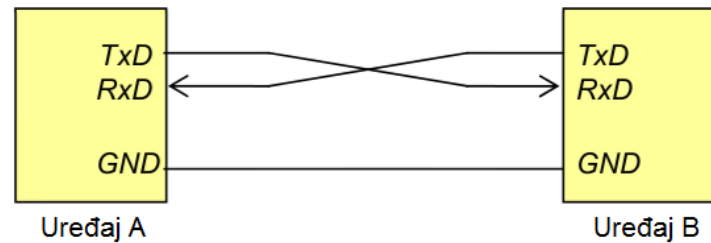
Simplex komunikacija

Sinhronizacija i detekcija grešaka prilikom serijskog prenosa

- Sinhronizacija rada između predajnika i prijemnika predstavlja jedan od osnovnih preduslova za ispravan serijski prenos podataka. Po pitanju sinhronizacije, serijska komunikacija može biti:
 - **Sinhrona** - Pored linije za prenos podataka, predajnik upravlja još i linijom za takt. Prenos se vrši sinhrono sa signalom takta.
 - **Asinhrona** - Ne postoji takt signal, nego predajnik prilikom slanja svake reči prvo šalje **početni (start)** a na kraju reči završni **(stop) bit**. Pri tome se podrazumeva da su predajnik i prijemnik podešeni na istu brzinu prenosa. Brzina prenosa izražava se brojem bita koji se šalje u jedinici vremena (u sekundi) - engl. **Baud Rate**.
- Sinhronizacija između predajnika i prijemnika može se vršiti i putem razmene upravljačkih signala, odnosno "rukovanja" (engl. **handshaking**). U ovom slučaju, potrebne su **dodatne linije** za prenos upravljačkih signala.
- U okviru serijske komunikacije, mogu biti implementirani različiti mehanizmi za detekciju (i eventualnu korekciju) grešaka. Najčeće korišćene metode detekcije grešaka su:
 - **Repetitivni kodovi** (engl. Repetition Codes) - Svaka reč koja se šalje ponavlja se unapred određen broj puta.
 - **Bit pariteta** (engl. Parity Bit) - Na kraju svake reči dodaje se po jedan bit koji odražava paritet poslate reči.
 - **Kontrolna suma** (engl. Checksum) - Na kraju bloka podataka sastavljenog od određenog broja reči, šalje se i njihova suma.
 - **CRC** (engl. Cyclic redundancy check) - Kontrolni kod koji se dodaje na kraj bloka podataka, zasnovan na ostatku pri deljenju polinoma formiranih na osnovu sadržaja bloka.

RS-232 komunikacioni protokol - osnovni pojmovi

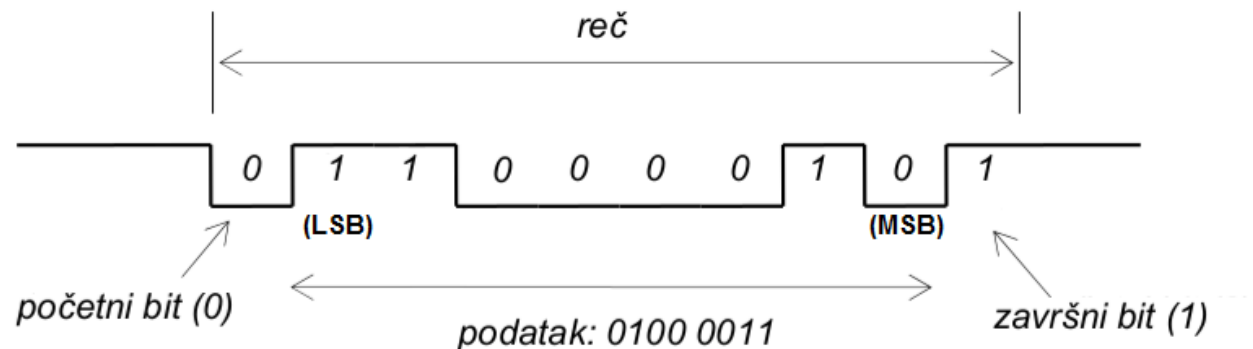
- Rad serijskog porta kod PC računara i kod većine savremenih mikrokontrolera zasnovan je na standardizovanom **RS-232 protokolu**.
- Hardverska jedinica unutar mikrokontrolera koja je zadužena za ovakav tip komunikacije obično se naziva **UART (engl. *Universal Asynchronous Receiver-Transmitter*)**.
- Po RS-232 standardu, komunikacija se obavlja posredstvom linije za slanje podataka (T_xD), linije za prijem (R_xD), a postoje i dodatni signali za sinhronizaciju (engl. *handshaking*). UART jedinice obično ne koriste sinhronizacione signale, nego samo linije koje su apsolutno neophodne za ostvarivanje veze: T_xD , R_xD i GND (masa).



- Paralelno sa prijemom poruke preko R_xD linije, moguće je istovremeno slati poruku preko nezavisne komunikacione linije T_xD . To znači da UART ustvaruje dvosmernu (**full-duplex**) komunikaciju

RS-232 komunikacioni protokol - format podatka

- Serijski port koji koristi RS-232 protokol slanje podataka vrši **asinhrono**, reč po reč (engl. *character*).
- Kada je linija neaktivna, na njoj je **visok naponski nivo**, tj. pre početka serijskog prenosa i između prenosa dve uzastopne reči, linija za serijski prenos drži se u stanju logičke 1. Opadajuća ivica na komunikacionoj liniji signalizira prijemnoj strani da predajna strana počinje sa slanjem reči.
- Reč koja se šalje sadrži:
 1. **Početni (start) bit**, koji je uvek logička 0
 2. **5, 6, 7 ili 8 bita podatka** koji se prenosi. Prenos se vrši od **najnižeg (LSB)** ka najvišem (MSB) bitu.
 3. **Opcioni bit za proveru pariteta**. Ovo predstavlja najjednostavniji način kontrole ispravnosti primljene reči. Bit pariteta se postavlja na "1" ukoliko je broj jedinica u okviru podatka (stavka 2) paran, ili neparan, što se može konfigurisati. Na prijemnoj strani se tada proverava koliko ima jedinica među bitima primljenog podatka i proverava se da li se to slaže sa bitom pariteta koji je primljen. Na ovaj način moguće je detektovati nastanak greške na jednom bitu, ali na nije moguće detektovati koji bit je pogrešno primljen, niti prepoznati grešku u slučaju pogrešnog prijema parnog broja bita.
 4. **1 do 2 završna (stop) bita**, koji su uvek logičke "1"

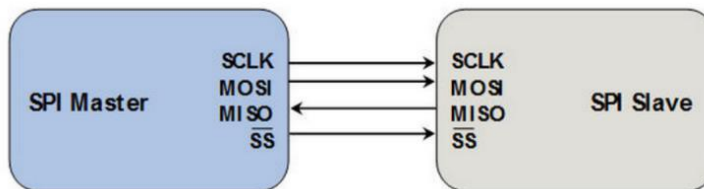


Parametri RS-232 komunikacije

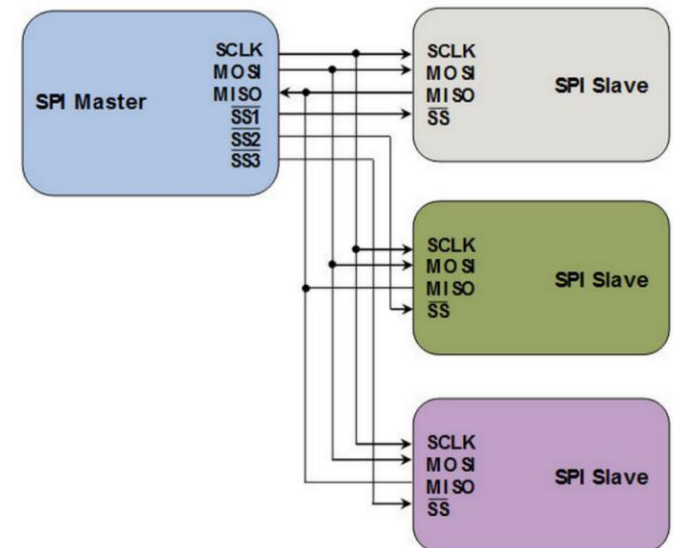
- Da bi prijemnik i predajnik uspešno komunicirali, potrebno je **konfigurisati** i prijemnik i predajnik na **identičan** način. Ova konfiguracija minimalno podrazumeva definisanje:
 1. **Brzine serijske komunikacije** (engl. *baud rate*), izražene **brojem bita u sekundi**. Tipične brzine RS-232 komunikacije su 9600b/s, 19200b/s, 38400b/s, 57600b/s i 115200b/s
 2. **Broja bita podataka** (5, 6, 7 ili 8)
 3. Da li se koristi **bit pariteta** i ako se koristi, da li će on biti setovan za paran ili neparan broj jedinica u okviru bita podataka
 4. **Broj stop bita** koji se koristi (1 ili 2)
- Nakon što su prijemnik i predajnik konfigurisani, prijemnik poruke prolazi kroz sledeća stanja čekajući na poruku:
 1. Čeka **početak poruke** pojavom start bita, tj. opadajuće ivice na prijemnoj R_xD liniji.
 2. Nakon detekcije početka poruke, prima **jedan po jedan bit** poruke tako što očitava napon na R_xD liniji svakih $kT + T/2$, $k=0,1,...,8$, $T=1/(\text{baud rate})$. Da bi ovo bilo moguće, neophodno je da **prijemnik i predajnik** imaju usklađenu brzinu komunikacije, jer je slanje asinhrono (prijemnik nema nikakvu indikaciju kada je na liniji dostupan "novi" bit).
 3. Nakon prijema cele poruke, proverava da li je **bit pariteta** u skladu sa **bitovima podataka** (u slučaju da je konfigurisano korišćenje bita pariteta). Ukoliko jeste, podatak je prihvaćen, u suprotnom primljeni podatak se odbacuje i neophodno ga je ponovo primiti. .

SPI (serijski periferni interfejs)

- SPI (engl. **Serial Peripheral Interface**) je protokol namenjen komunikaciji između integriranih kola, kao i za komunikaciju sa perifernim jedinicama koje se nalaze na istoj štampanoj ploči.
- Arhitektura SPI magistrale je takva da postoji jedan glavni (engl. **master**) uređaj i jedan ili više sporednih (engl. **slave**) uređaja.
- SPI interfejs koristi 4 linije, sa sledećim funkcionalnostima:
 1. **SCLK** (engl. *Serial CLock*) je taktni signal koji uvek generiše master i vodi se ka svim slave uređajima. Prenos podataka putem SPI magistrale sinhronizovan je u odnosu na ovaj signal.
 2. **SS_n** signal ima ulogu odabira slave uređaja (engl. **Slave Select**). Aktivnim stanjem SS signala odabira se jedan od povezanih slave uređaja sa kojim će master ostvariti komunikaciju.
 3. **MOSI** (engl. *Master Out - Slave In*) je linija za prenos podataka od mastera ka svim slave uređajima.
 4. **MISO** (engl. *Master In - Slave Out*) je linija za prenos podataka od slave uređaja ka master uređaju.



Jedan master - jedan slave uređaj



- **Master** inicira svaku komunikaciju sa ostalim uređajima na SPI magistrali. Kada SPI master želi da šalje podatak slave-u, ili da prihvati neki podatak od slave-a, on će ga adresirati tako što odgovarajuću **SS** liniju obori na nizak naponski nivo i aktivira taktni signal.
- Master šalje podatke putem MOSI linije, dok ih prima preko MISO linije.
- Postoje četiri moda (MODE 0, 1, 2 i 3) koji definišu na kojoj ivici SCLK signala se **menja stanje** na **komunikacionim linijama**, na kojoj ivici se očitavaju trenutne vrednosti signala na komunikacionim linijama i u kom stanju se nalazi SCLK linija kada je u neaktivnom stanju (nizak ili visok naponski nivo).
- Primer na slici pokazuje mod u kojem se **ažuriranje** bita informacije na MISO i MOSI liniji vrši na **silaznu** ivici SCLK signala (tj. i master i slave tada postavljaju nove bite), dok se na **rastuću** ivici **učitavaju vrednosti** na komunikacionim linijama.

