

# Osnovi mikroprocesorskih i mikrokontrolerskih sistema - Serijska komunikacija

prof. dr Ivan Mezei

13. maj 2022.





# Glava 10

## Serijska komunikacija

U dosadašnjim razmatranjima upoznali smo se sa paralelnim prenosom podataka kod koga se između dve jedinice istovremeno prenosi dva ili više bita. Međutim, serijski komunikacioni kanali su danas bez sumnje najzastupljeniji vid komunikacije koja se koristi unutar digitalnih sistema. Različiti oblici serijskih komunikacionih formata i protokola koriste se u aplikacijama počevši od kratkih komunikacionih linkova koji povezuju module unutar istog ili odvojenih integrisanih kola, pa sve do komunikacionih linkova koji obezbeđuju komunikaciju sa satelitima.

Praktično skoro svi oblici komunikacije koji se danas koriste u potrošačkoj elektronici baziraju na serijskoj komunikaciji. Neki od najpoznatijih serijskih komunikacionih protokola korišćenih danas su:

- RS-232,
- I2C, TWI, 1-Wire, SPI,
- USB, FireWire,
- Bluetooth,
- Ethernet, WiFi,
- PCIe, SATA i dr.

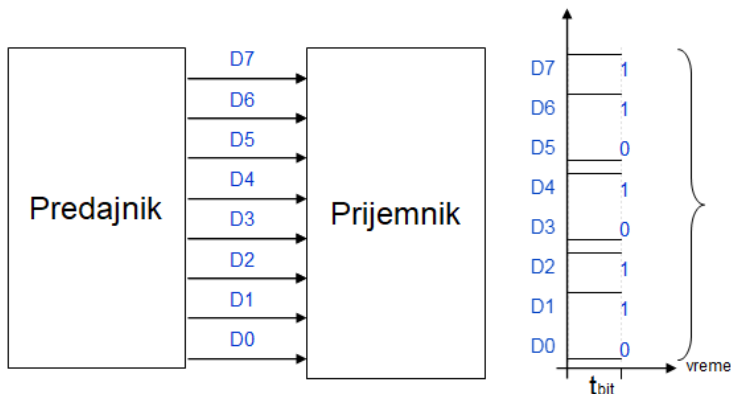
Mikrokontroler ATmega328P poseduje sledeće serijske interfejsse. To su: USART (sinhroni i asinhroni), SPI (eng. *Serial Peripheral Interface*) i TWI<sup>1</sup> (eng. *Two*

---

<sup>1</sup>Ovaj protokol je poznatiji kao I2C.

*Wire Interface*). U okviru ovog predmeta ćemo detaljno obraditi UART i RS-232 komunikaciju, a SPI i TWI se detaljno obrađuje u okviru predmeta Računarska elektronika u VIII semestru.

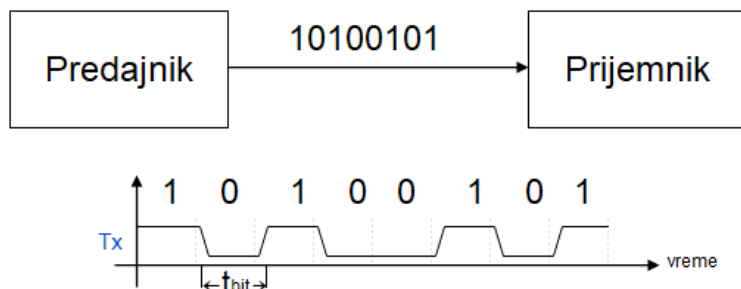
Paralelni komunikacioni kanal, prilikom prenosa  $n$ -bitnog podatka, koristi  $n$  simultanih signalnih linija, jednu za svaki bit podatka koji se prenosi. Dakle u jedinici vremena  $t_{bit}$  se prenosi 8 bita odjednom. Primer za 8-bitnu paralelnu komunikaciju je dat na slici 10.1.



Slika 10.1: 8-bitna paralelna komunikacija

Sa druge strane, serijski komunikacioni kanal, prilikom prenosa  $n$ -bitnog podatka, koristi samo JEDNU signalnu liniju, pri čemu se  $n$ -bita podataka prenosi sekvencijalno, jedan za drugim. Svaki bit koji se prenosi serijski zahteva unapred definisanu količinu vremena,  $t_{bit}$ , tako da je za prenos  $n$ -bitnog podatka potrebno ukupno  $n * t_{bit}$  sekundi. Ovakav prenos je prikazan na slici 10.2.

Kako je u jednom trenutku moguće preneti samo jedan bit, očigledan nedostatak serijskog prenosa je u tome što je sporiji u odnosu na paralelni prenos. Intuitivno, paralelni prenos sa  $n$  provodnika brži je  $n$  puta u odnosu na serijski prenos, naravno pod pretpostavkom da su svi ostali parametri prenosa isti. Ovo je uvek tačno, međutim kako rastojanje na kojem se vrši komunikacija i brzina kojom se prenose podaci rastu, talasni efekti na komunikacionim vodovima, preslušavanje i uticaj šuma imaju sve veći negativan uticaj u slučaju paralelnih komunikacionih kanala, ograničavajući njihovu praktičnu primenu. Ovo je jedan od najvažnijih razloga za masovno korišćenje serijskih komunikacionih kanala u savremenim elektronskim uređajima i sistemima. Ukoliko je ipak potrebna paralelna komunikacija,



Slika 10.2: Serijski prenos 8-bita

a da se uticaj šuma i drugih efekata smanji tada se koristi metodologija diferencijalnog povezivanja što je dosta skuplje rešenje i od običnog paralelnog, a pogotovo od serijskog povezivanja.

## 10.1 Brzina serijske komunikacije

Brzina prenosa podataka preko serijskog kanala određena je količinom vremena potrebnog za prenos jednog bita informacije ( $t_{bit}$ ). U praksi se koriste dve metrike za iskazivanje brzine prenosa podataka preko serijskih kanala:

- Bitska brzina (eng. *bit rate*) i
- Simbolska brzina (eng. *baud rate*).

**Bitska brzina** (eng. *bits per second* - bps) pokazuje koliko je bita moguće preneti tokom jedne sekunde preko serijskog kanala. Ukoliko je poznato bitsko vreme,  $t_{bit}$ , bitska brzina računa se pomoću sledećeg izraza. Na primer, serijski kanal sa bitskim vremenom  $t_{bit} = 10 \text{ ns}$  ima bitsku brzinu od 100 Mbit-a u sekundi (100 Mbps, 100 Mbit/s).

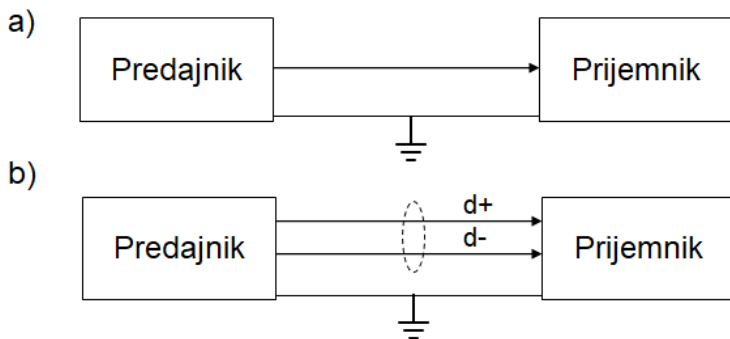
**Simbolska brzina** (eng. *baud rate*), odnosi se na broj karaktera (simbola) koji se mogu preneti preko serijskog kanala tokom jedne sekunde. Ukoliko je svaki simbol predstavljen pomoću jednog bita, simbolska i bitska brzina su jednake.

Ukoliko se koriste složenije modulacione tehnike, moguće je kodirati veći broj bita za svaku promenu signala na serijskom kanalu. Na primer, ukoliko se koristi

fazna modulacija (eng. *Phase-Shift Keying* - PSK), moguće je kodirati dva ili više bita u svaku promenu faze signala koji ide preko serijskog komunikacionog linka. U ovom slučaju bitska brzina će biti veća od simbolske brzine.

## 10.2 Komunikacioni kanali

Komunikacioni kanali mogu biti realizovani na razne načine. Ožičeni kanali mogu da koriste jednu liniju ili diferencijalne veze (žice) za prenos podataka. U slučaju jedne linije, kanal se sastoji iz jednog fizičkog voda (žice) i referentne linije (masa, eng. *ground*). U slučaju  $n$ -bitnog paralelnog kanala bilo bi potrebno ukupno  $n+1$  žica za ostvarivanje veze: po jedna žica za svaki bit plus referentna linija mase. U slučaju  $n$ -bitnog serijskog kanala dovoljne bi bile samo dve žice: jedna signalna i jedna referentna (videti a) sliku 10.3).



Slika 10.3: a) Prenos preko jedne linije b) Diferencijalna veza

U slučaju diferencijalne veze, informacija koja se prenosi kodirana je kao naponska razlika između dve fizičke žice. Na slici b) 10.3 to su žice  $d+$  i  $d-$ . Ova vrsta veze je otpornija na uticaj smetnji, jer se uticaj aditivnih smetnji međusobno poništava prilikom prenosa podataka preko diferencijalne veze. Paralelni diferencijalni  $n$ -bitni kanal zahtevao bi  $2n+1$  žica za ostvarivanje veze: po dve žice za svaki bit, plus jedna žica za prenos referentnog napona (mase). U slučaju serijskog diferencijalnog kanala bile bi potrebne samo 3 žice: dve za prenos informacija i jedna za prenos referentnog napona.

Na osnovu prethodne analize može se naslutiti glavna prednost serijskih kanala u odnosu na paralelne, a to je njihova cena. Mnogo je jeftinije i jednostavnije imati

samo jedan serijski link za prenos podataka nego n paralelnih, sinhronizovanih linkova u slučaju paralelnog prenosa. U slučaju diferencijalnog paralelnog prenosa situacija je takva da je to još skuplje.

Nasuprot žičanih komunikacionih kanala koji koriste metalne provodnike (obično od bakra ili aluminijuma), **bežični komunikacioni kanali** koriste bežične načine komunikacije za prenos podataka bazirane na različitim fizičkim fenomenima:

- Optičke, u slučaju infracrvenih ili laserskih primopredajnika
- Akustičke, u slučaju podvodnih komunikacija
- Elektromagnetne, u slučaju radio komunikacija kao što su WiFi, Bluetooth i sl.

U bilo kom od prethodnih slučajeva bežičnih komunikacija nema potrebe za zajedničkom referencom, ali bi u slučaju paralelnog prenosa i dalje bilo potrebno n bežičnih linkova, dok bi u slučaju serijskog prenosa bio dovoljan samo jedan ovakav link.

Svi serijski komunikacioni kanali zahtevaju postojanje stabilnog izvora takta koji predstavlja referentni sinhronizacioni signal na osnovu kojega se mogu odrediti brzine predaje i prijema podataka i izvršiti unutrašnja sinhronizacija prijemnika i predajnika. U zavisnosti od načina na koji se obezbeđuje signal takta, kanal može biti:

- asinhroni ili
- sinhroni.

Asinhroni kanali koriste nezavisne generatore takta na prijemnoj i predajnoj strani. Sinhroni kanali distribuiraju jedan, zajednički takt, zajedno sa podacima koji se prenose. Ovaj zajednički takt generiše se na jednom mestu, najčešće na predajnoj strani. Kod sinhronih kanala obično je potrebna jedna dodatna linija za prenos signala takta.

## 10.3 Komunikacioni paketi

I asinhroni i sinhroni serijski komunikacioni protokoli dele poruke koje prenose preko kanala na osnovne komponente koji se nazivaju paketi podataka ili datagrami. Svaki paket sastoji se iz tri dela:

- zaglavlja (eng. *header*),



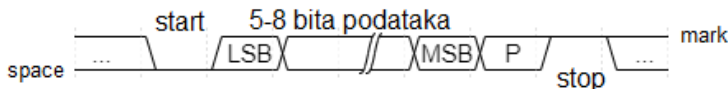
- tela (eng. *body*) i
- repa (eng. *footer*).

Zaglavlje i rep započinju, odnosno završavaju, svaki paket i sadrže sinhronizaciona informaciona polja koja su dodata od strane komunikacionog protokola kako bi se obezbedio pouzdan prenos podataka kroz komunikacioni kanal. Zaglavlje i rep razlikuju se po dužini i formatu u slučaju asinhronih i sinhronih kanala kao u zavisnosti od komunikacionog protokola koji se koristi.

Zaglavlje paketa sadrži polje koje označava početak paketa, opciona polja koja sadrže adresne informacije i opciona polja koja sadrže informaciju o dužini i tipu paketa. Rep paketa sadrži polje koje označava kraj paketa i opciono polje koje sadrži informacije na osnovu kojih se može proveriti ispravnost primljenog paketa.

Za razliku od zaglavlja i repa koji služe da obezbede prenos i sinhronizaciju podataka, podaci (eng. *payload data*) se prenose u okviru tela paketa.

Kao primer ćemo ilustrovati asinhronu serijsku paket podataka koji imaju vrlo jednostavnu strukturu. Njihovo zaglavlje sadrži samo jedan bit, koji se zove start bit. Vrednost start bita jednaka je logičkoj nuli (tzv. 'space' simbol) i označava početak paketa. Telo paketa sadrži karakter, kodovan pomoću 5 do 8 bita organizovanih od bita najmanje značajnosti (LSB) do bita najveće značajnosti (MSB). Rep paketa sadrži opcioni bit parnosti (P), koji omogućava detekciju 1-bitnih grešaka koje su se javile prilikom prenosa, kao i jedan ili više stop bita koji označavaju kraj paketa. Vrednost stop bita jednaka je logičkoj jedinici (tzv. 'mark' simbol) i označava kraj paketa. Ovo je ilustrovano na slici 10.4.



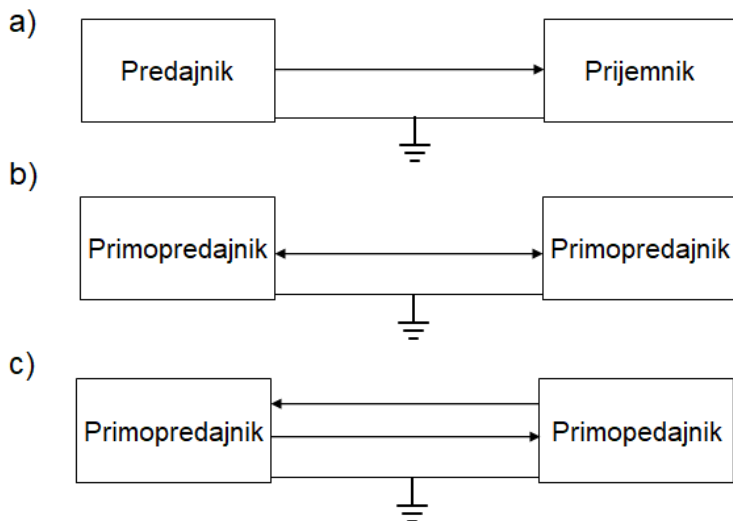
Slika 10.4: Struktura asinhronog serijskog paketa

## 10.4 Vrste serijskih komunikacionih kanala

Na osnovu mogućnosti prenosa podataka između predajnika i prijemnika, serijski komunikacioni kanali dele se u sledeće grupe:

- Unidirekcioni (eng. *simplex*)
- Polu-bidirekcioni (eng. *half-duplex*) i

- Bidirekcionni (eng. *full-duplex*).



Slika 10.5: Vrste serijskih komunikacionih kanala

**Unidirekcionni serijski kanal** prenosi podatke samo u jednom smeru koristeći odgovarajući komunikacioni link. Na jednom kraju nalazi se predajnik, dok se na drugom kraju nalazi prijemnik (slika 10.5 a)). Ovakvi kanali nemaju način za potvrdu i verifikaciju ispravnog prijema podataka od strane prijemnika. Primeri ovih kanala su radio i TV predajni sistemi, ili izlazne periferije poput LED ili displeja.

**Polubidirekcionni serijski kanali** koriste jedan komunikacioni link, ali se preko njega podaci mogu prenositi u oba smera pri čemu se u svakom trenutku podaci mogu prenositi samo u jednom smeru (slika 10.5 b)). Kod ovih sistema na oba kraja nalaze se serijski primopredajnici (eng. *serial transceiver*), koji imaju mogućnost slanja i prijema podataka. Svaki put kada je potrebno promeniti smer prenosa, primopredajnici na oba kraja menjaju modove rada. Ova promena zahteva postojanje odgovarajućih pravila kako bi se izbegla situacija da na oba kraja primopredajnici pokušavaju da pošalju podatke. Ovaj skup pravila koji određuje ponašanje primopredajnika na oba kraja kanala i način na koji oni organizuju i interpretiraju podatke koji putuju kanalom naziva se **komunikacioni protokol**.

**Bidirekcionni serijski kanali** koriste dva posebna komunikaciona linka, jedan za prenos a drugi za prijem podataka, omogućavajući istovremeni prenos podataka u oba smera (slika 10.5 c)). Primopredajni uređaji na krajevima kanala su složeniji od prethodnih i imaju mogućnost istovremenog prijema i predaje podataka, ne zahtevajući nikakvu promenu u režimu rada, na taj način omogućavajući neprekidni bidirekcionni prenos podataka. Većina savremenih serijskih komunikacionih kanala koji se koriste u embeded sistemima koriste bidirekzione serijske kanale.

Komunikacioni kanali prikazani na slici 10.5 predstavljaju takozvane *point-to-point* topologije, kod kojih kroz jedan kanal komuniciraju tačno dva uređaja. Postoje i složenije topologije koje dozvoljavaju postojanje većeg broja uređaja koji međusobno komuniciraju koristeći jedan, zajednički kanal. Ovakve topologije poznate su pod imenom *multi-point* ili *multi-drop* topologije (npr. RS-422, RS-485, ožičena I magistrala i dr.).

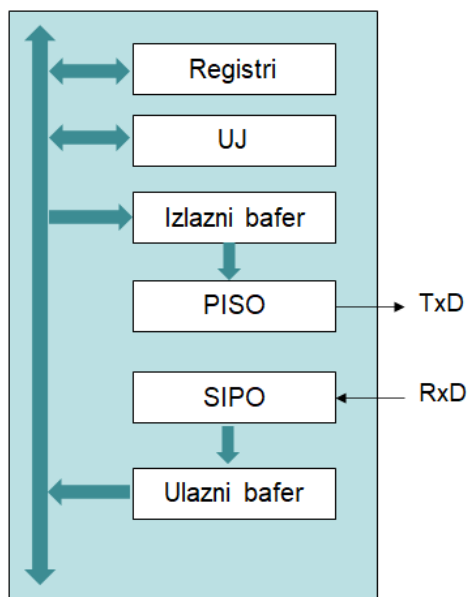
## 10.5 Modul za asinhronu serijsku komunikaciju UART

Modul koji obezbeđuje asinhronu serijsku komunikaciju poznat je pod nazivom UART (eng. *Universal Asynchronous Receiver and Transmitter*). UART se nekada ranije proizvodio kao zasebno integrisano kolo, a danas je to uobičajen blok unutar velike većine mikrokontrolera.

UART kombinuje funkcionalnost predajnika i prijemnika unutar jednog modula sa ciljem obezbeđivanja bidirekcionnog serijskog interfejsa. Da bi se obezbedila komunikacija između dva uređaja putem asinhronog serijskog komunikacionog kanala oba uređaja moraju programirati njihove UART module tako da: koriste istu bitsku brzinu, koriste isti broj bita prilikom prenosa karaktera, ako se koristi detekcija grešaka prilikom prenosa da se odredi koji tip detekcije će biti korišćen, da se zna koliki broj stop bita će biti korišćen za označavanje kraja tekućeg paketa i dr. Konfigurisanje različitih parametara UART modula vrši se pomoću kontrolnih registara.

Kod serijskog prenosa podataka na predajnoj strani mora da se obavi paralelno-serijska (eng. *Parallel In Serial Out* - PISO), a na prijemnoj strani serijsko-paralelna (eng. *Serial In Parallel Out* - SIPO) konverzija. O principima ovih konverzija je bilo reči u poglavlju o povezivanju periferija. Obe vrste konverzija zasnovane su na primeni pomeračkog registra sa serijskim i paralelnim ulazima i izlazima. Unutrašnje komponente UART-a nalaze se oko dva ovakva pomeračka registra. Pojednostavljena struktura UART modula je prikazana na slici 10.6.

Da bi inicirao prenos, CPU upisuje podatak (u slučaju serijske komunikacije



Slika 10.6: Struktura UART modula

se obično kaže karakter) koji je potrebno preneti u izlazni bafer. Svaki karakter, jedan po jedan, se zatim okružuje sa odgovarajućim poljima zaglavlja i repa, što je funkcija koju realizuje upravljački automat i upisuje u PISO registar. U ovom trenutku upravljački automat takođe računa i vrednost bita parnosti (ukoliko se on koristi za detekciju grešaka). Sadržaj PISO registra se zatim šalje preko serijskog linka brzinom koju određuje baud-rate generator predajnika. Kada se svi karakteri iz izlaznog bafera pošalju, UART podiže TxReady indikator, označavajući da je spreman za prijem novih karaktera koje je potrebno poslati preko serijskog linka. Upis u izlazni bafer automatski briše TxReady indikator. Jednostavni UART moduli imaju izlazni bafer u koji se može smestiti samo jedan karakter, dok je kod složenijih UART-a ovaj bafer veći i dostiže veličine i od nekoliko kilobajta.

Prilikom prijema, validni start bit resetuje SIPO registar i započinje upis bita koji se primaju preko serijskog linka u SIPO registar. Kada su svi biti smešteni u SIPO registar, upravljački automat uklanja start i stop bite, izvodi proveru parnosti (ukoliko je neophodna) i uklanja bit parnosti, postavlja primljeni karakter u ulazni bafer i podiže RxReady indikator. Kada detektuje da je RxReady indikator

aktiviran, CPU čita primljeni karakter iz ulaznog bafera. Čitanje ulaznog bafera obično briše RxReady indikator, mada je nekada to potrebno uraditi u programu za obradu prijema. U slučaju da su detektovane greške parnosti ili je paket nepravilno formiran, UART signalizira novonastalu situaciju podizanjem odgovarajućih indikatora greške unutar statusnog registra.

Ukoliko se prilikom prijema ulazni bafer puni brže nego što CPU iz njega čita karaktere, može doći do gubitka nekih od karaktera koji su bili poslati. Ovakva situacija poznata je pod nazivom '*overrun error*'. Da bi se izbegla ova situacija u praksi se koriste ulazni baferi koji imaju mogućnost prihvatanja većeg broja karaktera, na taj način oslobađajući CPU od potrebe za brzom reakcijom nakon indikacije prijema podataka PISO i SIPO registri rade na učestanosti predajnog i prijemnog takta koji određuju i brzine predaje i prijema podataka. Svi indikatori koji označavaju trenutni status predajnika, prijemnika i eventualno postojanje grešaka dostupni su preko odgovarajućih statusnih registara.

Statusni registar (ili više njih) omogućava utvrđivanje statusa u kome se nalazi kanal i UART modul. On obično sadrže indikatore grešaka (npr. *buffer overrun*, *framing error*, *parity error*) i status predajnika i prijemnika (*TxReady*, *RxReady*).

Glavne linije interfejsa UART-a ka serijskom kanalu se obično označavaju sa TxD (eng. *Transmitted Data*) i RxD (eng. *Received Data*), koje zajedno sa linijom referentnog napona (GND) prenose dolazne i odlazne serijske podatke ka i od UART modula. TxD predstavlja serijski izlazni port koji je direktno povezan na izlaz PISO registra. RxD predstavlja serijski ulazni port koji je direktno povezan na ulaz SIPO registra. Kod većine mikrokontrolera TxD i RxD signali su multipleksirani sa GPIO signalima ili nekim drugim signalima tako da je potrebno izvršiti odgovarajuća konfigurisanja mikrokontrolera ukoliko se žele koristiti podešavanjem odgovarajućih registara. Znatno detaljniji opis UART modula i drugih aspekata serijske komunikacije je moguće pročitati u poglavlju 'Principi serijske komunikacije' knjige 'Uvod u embeded sisteme' (na engleskom) [16].

## 10.6 RS-232 standard

Pored TxD i RxD signala, neki UART moduli mogu uključiti i dodatne signale unutar interfejsa serijskog kanala pomoću kojih je moguće obezbediti hardversku kontrolu toka podataka. To su DSR (eng. *Data Set Ready*), ulazni signal UART modula koji predstavlja indikaciju da je spoljašnji komunikacioni modul spreman za rad, DTR (eng. *Data Terminal Ready*), izlazni signal UART modula koji predstavlja indikaciju da je UART modul spreman za rad, RTS (eng. *Ready to Send*), izlazni signal UART modula koji obaveštava spoljašnji uređaj da želi da izvrši

slanje podataka, CTS (eng. *Clear to Send*), ulazni signal UART modula koji predstavlja indicaciju da je spoljašnji komunikacioni modul spreman da primi podatke. Ovi dodatni signali, zajedno sa RI (eng. *Ring Indicator*) i DCD (eng. *Data Carrier Detected*) čine skup signala koji definišu dobro poznati RS-232 komunikacioni interfejs [39]. Međutim većina savremenih mikrokontrolera ne podržava direktno sve ove dodatne signale, a u većini slučajeva nisu ni potrebni.

Pored komunikacionog interfejsa RS-232 standard definiše i naponske nivoe i mehanička specifikacija konektora za povezivanje uređaja.

### 10.6.1 Naponski nivoi

Svi signali koji čine interfejs serijskog kanala moraju da koriste logičke naponske nivoe koji se koriste unutar UART modula. Na primer, ukoliko UART modul radi na naponu napajanja od 5 V, on će biti u stanju da radi i toleriše TTL kompatibilne signale. Ukoliko je UART deo nekog mikrokontrolera koji radi na 3.3 V, on će biti u stanju da prihvata signale iz opsega 0-3.3 V. Korišćenje logičkih naponskih nivoa za primopredaju serijskih podataka u mnogome će smanjiti maksimalno rastojanje na koje ovi podaci mogu da se šalju na desetine cm, pre nego što oni postanu previše zaprljani šumom, pogotovo u slučaju velikih brzina prenosa. Da bi se obezbedila mogućnost prenosa serijskih podataka na veće udaljenosti moraju se koristiti drugačiji naponski nivoi i formati signala.

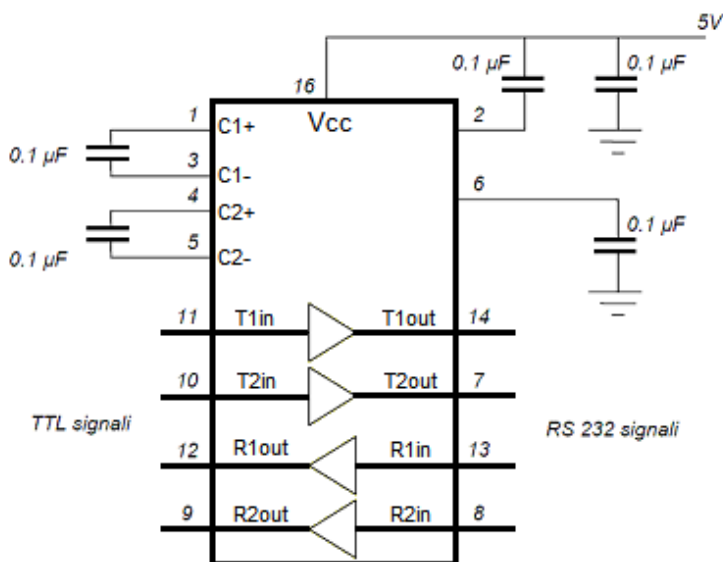
Fizički standardi za serijsku komunikaciju obično koriste veliki dinamički opseg naponskih nivoa i/ili odgovarajuće signalizacione mehanizme u cilju povećanja otpornosti na uticaje šuma prilikom prenosa podataka kroz kanal. Na primer, u slučaju RS-232 kanala, naponski nivoi koji se koriste za prenos 'nule' i 'jedinice' mogu biti -12 V i +12 V. Standardom je definisan opseg ovih napona. Za 'nulu' opseg iznosi od +3 do +25 V, a za 'jedinicu' iznosi od -3 do -25 V. Ovi naponski nivoi nisu direktno kompatibilni sa logičkim nivoima signala koje očekuje ili generiše UART modul. Zbog toga je u većini ovakvih sistema potrebno korišćenje naponskih translatora kako bi se ovo povezivanje ostvarilo na način koji će obezbediti pouzdan rad i neće dovesti do oštećenja komunikacione opreme.

Električne karakteristike signala u mikroračunarskom sistemu drugačije su od električnih signala definisanih standardom RS232. Signali mikroračunara i, na primer signali mikrokontrolera x51, kompatibilni su sa TTL standardom, pa je između mikroračunara i uređaja sa signalima koji su u električnom pogledu kompatibilni sa RS-232 standardom potrebno staviti konvertor za električno prilagođenje signala ova dva standarda.

Kao primer konvertora napona uzećemo kolo MAXIM MAX232E, čiji blok dijagram je prikazan na slici 10.7. Ovo kolo ima dva konvertora T1 i T2 koji

digitalne signale ulazne signale T1IN i T2IN kompatibilne sa TTL standardom, prevode u digitalne izlazne signale T1OUT i T2OUT koji su kompatibilni sa RS 232 signalima. Druga dva konvertora R1 i R2, digitalne ulazne signale R1IN i R2IN kompatibilne sa RS232 standardom, prevode u digitalne izlazne signale R1OUT i R2OUT koji su kompatibilni sa TTL signalima.

Zanimljivo je da kolo MAX232E ima samo jedno napajanje, +5 V. Interne naponske pumpe za čiji rad su potrebni kondenzatori prikazani na slici, napon od + 5 V konvertuju u napone +10 V i -10 V koji su potrebni za izlazne signale koji su kompatibilni sa RS-232 standardom. Na slici 10.7 prikazani su i brojevi spoljnih priključaka kola MAX232E i nazivi spoljnih signala.

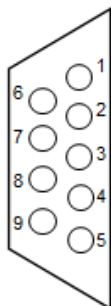


Slika 10.7: Konvertor napona MAX232E

### 10.6.2 Mehanička specifikacija

Originalni standard RS-232 predvideo je korišćenje konektora sa 25 nožica (naziva se DB25), koji podržava dva dupleks serijska kanala. Međutim, pošto je u praksi obično korišćen samo jedan kanal, vremenom je konektor od 25 nožica zamenjen konektorom sa 9 nožica (koji se naziva DB9). Raspored signala po nožicama konektora DB9 i njegov izgled prikazani su na slici 10.8.

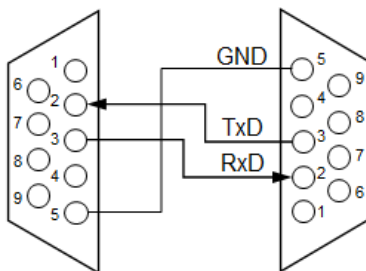
- 1: DCD
- 2: RxD
- 3: TxD
- 4: DTR
- 5: GND
- 6: DSR
- 7: RTS
- 8: CTS
- 9: RI



Slika 10.8: Konektor za povezivanje putem serijske komunikacije

### 10.6.3 Povezivanje dva računara

Ako su dva mikroračunara sa međusobnom serijskom vezom dovoljno blizu, moguće je prenositi digitalne signale u originalnom obliku, bez korišćenja modema za modulaciju serijskog signala. U ovom slučaju RS-232 signali dva mikroračunara mogu se neposredno spojiti provodnicima. Pošto u takvoj vezi nema modema, ona se obično naziva **null modem** veza, a kabel sa provodnicima naziva se null modem kabel (videti sliku 10.9).



Slika 10.9: Null modem kabel

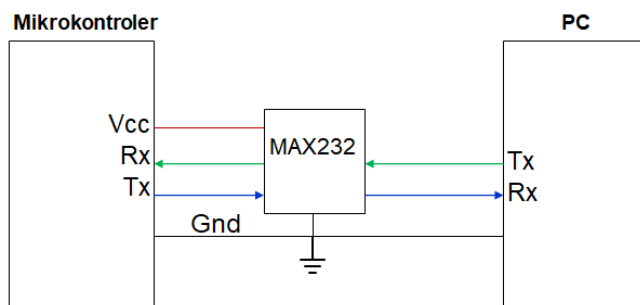
### 10.6.4 Povezivanje personalnog računara i mikrokontrolera

Da bismo povezali personalni računar (PC) i mikrokontroler potrebne su sledeće komponente (pogledati sliku 10.10):

- personalni računar ili laptop,

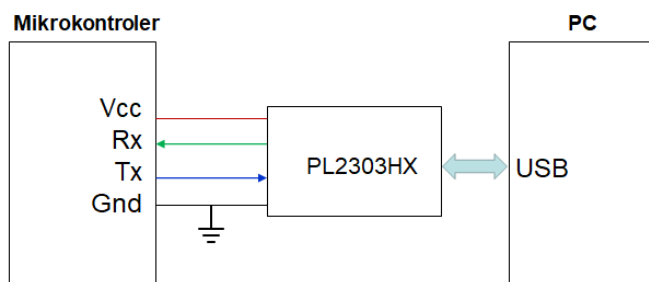


- serijski kabel,
- pretvarač naponskih nivoa (npr. MAX232 ili sl.) i
- mikrokontroler.



Slika 10.10: Povezivanje mikrokontrolera i računara

Današnji personalni računari i laptopovi najčešće nemaju konektor za serijski port na kućištu na raspolaganju, a imaju USB port. Tada se može koristiti konvertor USB-RS232 (npr. PL2303HX ili sl.) koji nakon instalacije formira virtuelni serijski port na personalnom računaru. Primer je dat na slici 10.11.



Slika 10.11: Povezivanje mikrokontrolera i računara preko USB porta

Temu povezivanja i umrežavanja u embeded sistemima je posvećen predmet 'Umreženi embeded sistemi' na master akademskim studijama.

## 10.7 Pitanja

1. Navesti koje se vrste serijskih interfejsa koriste kod ATmega328P.
2. Uporediti serijsku komunikaciju sa paralelnom?
3. Kako se određuje brzina serijske komunikacije?
4. Objasniti bitsku i simbolsku brzinu?
5. Kako se realizuju ožičeni komunikacioni kanali?
6. Zbog čega se koristi diferencijalni prenos?
7. Na bazi kojih fizičkih fenomena se realizuju bežični komunikacioni kanali?
8. Kako se dele komunikacioni kanali u zavisnosti od takta?
9. Objasniti razlike između sinhronih i asinhronih komunikacionih kanala.
10. Objasniti delove komunikacionih paketa.
11. Nacrtati strukturu asinhronih serijskih paketa i objasniti.
12. Koje vrste serijskih komunikacionih kanala postoje?
13. Nacrtati, objasniti i uporediti simpleks, half-dupleks i full-dupleks serijske komunikacione kanale.
14. Nacrtati pojednostavljenu strukturu UART modula i objasniti ga.
15. Čemu služe i zbog čega su potrebni konvertori naponskih nivoa u slučaju RS232? Navesti primer integrisanog kola koje to radi i dati njegovu šemu.
16. Šta je to null-modem veza?
17. Nacrtati i objasniti kako se mogu povezati personalni računar (ili laptop) i mikrokontroler?



# Bibliografija

- [1] Arthur W. Burks, Herman H. Goldstine, and John von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument", The Institute of Advanced Study, Princeton, USA, 1946-47.
- [2] J. Biggs, J. Myers, J. Kufel et al. A natively flexible 32-bit Arm microprocessor, *Nature* 595, pp. 532–536, 2021.
- [3] F. Arute, K. Arya, R. Babbush et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, pp. 505–510, 2019.
- [4] J. Hochstetter, R. Zhu, A. Loeffler et al. Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nat Commun* 12, 4008, 2021.
- [5] B. B. Brey, *Intel Microprocessors*, 8th Edition, Pearson, 2009.
- [6] S. Harris, D. Hariss, *Digital Design and Computer Architecture*, Arm edition, Morgan Kaufmann, 2015.
- [7] D. Patterson, J.L. Hennessy, *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*, The Morgan Kaufmann Series in Computer Architecture and Design, 2nd Edition, 2021.
- [8] M. M. Mano, C. R. Kime, T. Martin, *Logic and Computer Design Fundamentals*, Fifth edition, Pearson, 2015.
- [9] R. S. Sandige, M. L. Sandige, *Fundamentals of Digital and Computer Design with VHDL*, McGraw Hill, 2012.
- [10] L. Null, J. Lobur *Essentials of Computer Organization and Architecture*, 5th Edition, Jones and Barlett Publishers, 2018.

- [11] E.O. Hwang, Digital Logic and Microprocessor Design With VHDL with Interfacing, Cengage Learning, Second Edition, 2017.
- [12] I. Mezei "Evolution of an educational microprocessor", Computer Applications in Engineering Education, 28(5), Wiley, pp. 1265-1277, 2020.
- [13] W. Stallings, Computer organization and architecture, 9th Edition, Pearson, 2013.
- [14] M. Abd-El-Barr, H. El-Rewini, Fundamentals Of Computer Organization And Architecture, John Wiley and Sons, Inc., 2005.
- [15] C. Hamacher, Z. Vranesic, S. Zaky, N. Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw-Hill, 2012.
- [16] Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, Introduction to Embedded Systems, Springer, 2014.
- [17] D. Patterson, "Reduced Instruction Set Computers Then and Now" in Computer, vol. 50, no. 12, pp. 10-12, 2017. <https://doi.ieeecomputersociety.org/10.1109/MC.2017.4451206>
- [18] D. A. Patterson, J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Fourth Edition, Morgan Kaufmann, 2011.
- [19] J.L. Gustafson, Moore's Law. In: D. Padua D. (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [20] S.A. McKee, R.W. Wisniewski, Memory Wall. In: D. Padua (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [21] P. Bose, Power Wall. In: D. Padua (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [22] A. Tsakyridis, T. Alexoudi, A. Miliou, N. Pleros, and C. Vagionas, "10 Gb/s optical random access memory (RAM) cell," Opt. Lett. 44, pp. 1821-1824, 2019.
- [23] W.W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques", IEEE Wescon Proc., Aug. 1970.
- [24] C.G. Bell, A. Newell, Computer structures: Readings and Examples, McGraw-Hill, 1971.

- [25] S. Brown, Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, McGraw Hill, 2009.
- [26] D. L. Perry, VHDL: Programming by Example, Fourth edition, McGraw Hill, 2002.
- [27] L. Null, J. Lobur, Essentials of Computer Organization and Architecture, Jones and Bartlet Learning, 3rd Ed., 2010.
- [28] D. Patterson, J.L. Hennessy, Computer Organization and Design RISC-V Edition: The Hardware Software Interface, The Morgan Kaufmann Series in Computer Architecture and Design, 1st Edition, 2017.
- [29] S. L. Harris, D. Harris, Digital Design and Computer Architecture, RISC-V Edition, Morgan Kaufman, 1st edition, 2021.
- [30] D. A. Patterson, J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, ARM edition, The Morgan Kaufmann Series in Computer Architecture and Design, 2016.
- [31] D. Kushner, "The making of arduino", IEEE Spectrum 26, 2011.
- [32] M. Banzi, "How Arduino is open-sourcing imagination", TEDtalk, Scotland, 2012. <https://www.youtube.com/watch?v=UoBUXOOdLXY> (datum pristupa: 31.01.2022.)
- [33] Sparkfun, "Arduino shields v2", <https://learn.sparkfun.com/tutorials/arduino-shields-v2>, (datum pristupa: 01.02.2022.)
- [34] Arduino Uno Rev3 schematic, [https://www.arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf) (datum pristupa: 02.02.2022.)
- [35] Arduino software, <https://www.arduino.cc/en/Main/Software> (datum pristupa: 02.02.2022.)
- [36] 74HC595 datasheet, Philips, 1998, <https://www.arduino.cc/en/uploads/Tutorial/595datasheet.pdf> (datum pristupa: 03.03.2022.)
- [37] 74HC165 datasheet, Texas Instruments, 2015 <https://www.ti.com/lit/ds/symlink/sn74hc165.pdf> (datum pristupa: 03.03.2022.)
- [38] ATmega328 datasheet, Atmel, 2015, [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (datum pristupa: 15.03.2022.)

- [39] Tutorials 83, Fundamentals of RS-232 serial communications, Analog Devices, 2001. <https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html> (datum pristupa: 31.03.2022.)