

Završni rad

Serijski port mikrokontrolera atmega328

Predmet: Primena mikrokontrolera

Mentor:

dr Zoran Milivojević

Student:

Filip Stojanović

REr 56/17

April 2021.

Završni rad

Serijski port mikrokontrolera atmega328

Predmet: Primena mikrokontrolera

Mentor:

dr Zoran Milivojević

Student:

Filip Stojanović

REr 56/17

Članovi komisije:

1. Nataša Nešić
2. Danijela Aleksić
3. Zoran Milivojević

Zahvalnica

Zahvaljujem se svim profesorima na akademiji tehničko vaspitačkih strukovnih studija na odseku Niš, a naročito mom mentoru, profesoru dr Zoranu Milivojeviću na vođenju kvalitetne nastave koja me je podstakla da želim više od sebe i naučim šta sam mislio da neću moći.

Sadržaj

1. Uvod	1
2. Istorija mikrokontrolera	2
2.1 Rani mikroprocesori	2
2.2 Rani mikrokontroleri	3
2.3 Istorija Arduino čipova	5
3. Paralelna komunikacija	7
3.1 Istorija paralelne komunikacije	8
3.2 Clock skew	9
4. Serijska komunikacija	11
4.1 Istorija serijske komunikacije	13
4.2 USART	14
5. Serijska komunikacija arduino uno uređaja	16
5.1 ATmega328p registri	17
5.2 Arduino metode	18
5.3 Format	21
5.4 Softverski i hardverski serial	21
5.5 Preglednost podataka za ljude ili mašine	22
6. Poređenje paralelne i serijske komunikacije	23
6.1 Korišćena oprema	24
6.2 Podešavanja i merenja paralelne komunikacije	24
6.3 Podešavanja i merenja serijske komunikacije	24
6.4 Poredjenje rezultata merenja	24
7. Serijska komunikacija između dva Arduino Uno uređaja	26
7.1 Korišćena oprema	26
7.2 Podešavanje prijemnika	26
7.3 Podešavanje predajnika	29
7.4 Tok komunikacije	29
8. Zaključak	30

Literatura	31
Sažetak / Abstract rada	33
Biografija	34

1. Uvod

Smatra se da je prvi elektronski računar bio ENAIC (Electronic Numerical Integrator and Computer) koji je proizveden u SAD-u 1965 godine. Od tada su računari doživeli veliki tehnološki napredak, prošli pet generacija i danas su za razliku od svojih predaka veoma kompaktni uređaji. U ovom radu istražićemo mikrokontroler atmega328, specifično mogućnosti serijskog interfejsa za dvosmernu (full duplex) komunikaciju, koristeći mikrokontrolerski sistem Arduino Uno R3 baziran na atmega328p.

Mikrokontroler je mali kompjuter koji sedi na integrisanom kolu i njegova namena je upravljanje uređajima i procesima u ugrađenim sistemima (Engl. embedded system). Mikrokontroler sadrži jedan ili više CPU (Central processing unit), memoriju i programabilne ulazno/izlazne periferije. Glavna razlika između modernih mikroprocesora i mikrokontrolera je da su prvi optimizovani za brzinu i performanse kod računarskih programa dok su drugi optimizovani u pravcu integracije većeg broja kola. Mikrokontroler može sadržati i oscilatore, timere, brojače, serijski port, analogno digitalni konvertor i druge dodatke za koje je nekada bio potreban niz posebnih integralnih kola.

Mikrokonktroleri su slični ali manje sofisticiraniji nego SoC (System on a chip) kao što su Raspberry Pi računari. SoC može sadržati mikrokontroler kao komponentu ali integriše ga i sa naprednim perifernim uređajima poput GPU (Graphics processing unit), wifi modulima, eksterenim diskovima... Sa nekim modernim mikrokontrolerima teško je tačno definisati šta je SoC a šta je mikrokontroler jer oba mogu sadržati GPU i wifi module.

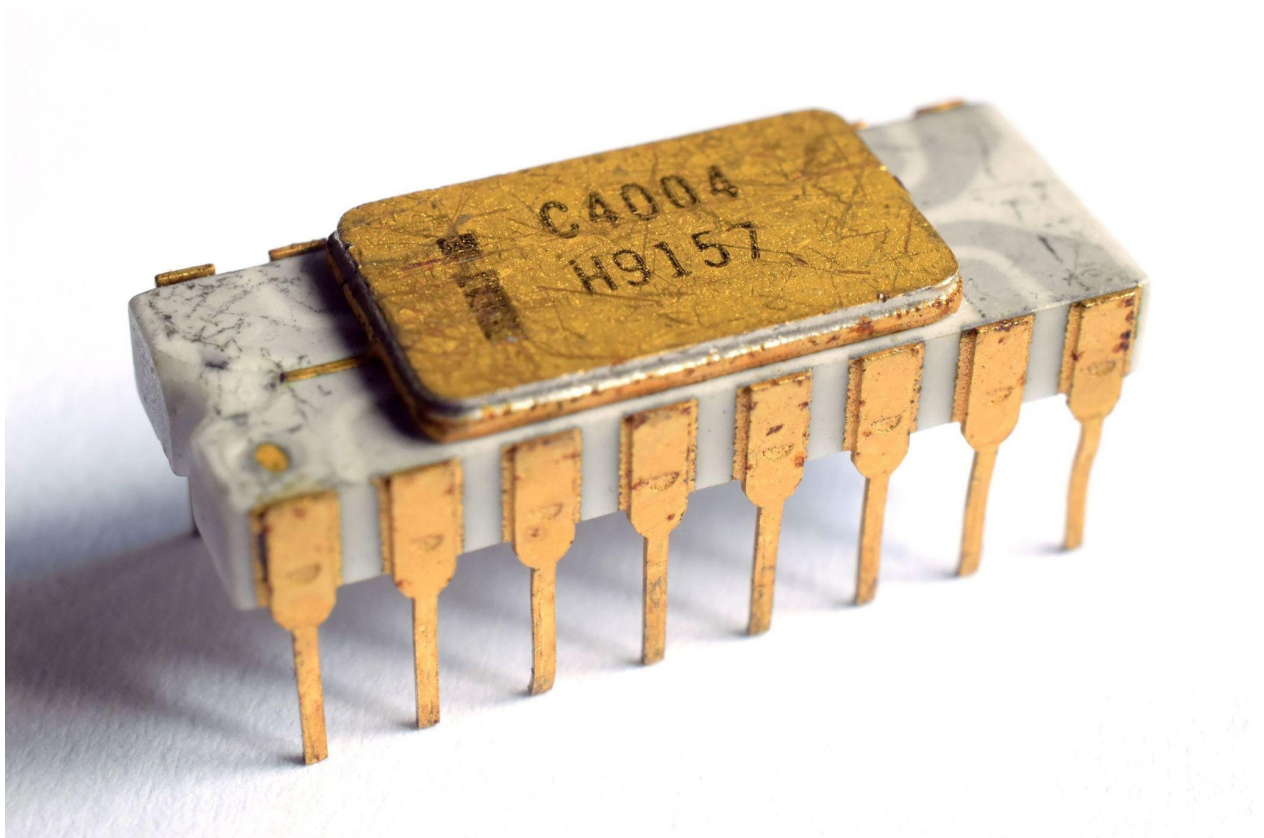
USART (Universal synchronous and asynchronous receiver-transmitter) je protokol za serijsku komunikaciju koji se koristi za slanje i primanje podataka po određenoj stopi bodova (Engl. Baud rate). Bod predstavlja broj prenetih signala kroz neku sredinu u sekundi - svaki signal može nositi jedan ili više bitova informacije. Tako da, ako imamo signal od 250 bodova a svaki signal ima po 4 bita informacija, onda nam je brzina prenosa 1000 bitova u sekundi (bit/s).

Arduino je kompanija koja proizvodi softver, razvojni sistem Arduino IDE otvorenog tipa kao i jednopločne mikroračunarske sisteme otvorenog hardverskog dizajna sa ciljem da omogući brzu proizvodnju embedded uređaja. Ime je dobila po kafiću u Italiji u kome su se osnivači nalazili. Arduino IDE razvojni okruženje primarno služi za programiranje Arduino i kompatibilnih Genuino ploča. Međutim, moguće je razvijati softver i za druge mikrokontrolere ali to obično zahteva zaseban programator.

2. Istorija mikrokontrolera

Začeci mikroprocesora kao i mikrokontrolera mogu se pronaći u MOSFETU (metal oxide silicon transistor), kompaktnom tipu operacionog pojačivača izumljenom od strane Bell Labs i javno prikazanom 1959 godine. Nakon početka mikroprocesora, inovacija u računarstvu dešavala se tolikim tempom da je donet zakon nazvan po zapažanjima Gordon Moore-a (Murov zakon) koji glasi da se broj tranzistora u integralnom kolu udvostručuje približno svakih 18-24 meseca. Ovaj zakon je važio 55 godina ali mu se bliži kraj početkom doba kvantnih kompjutera.

2.1 Rani mikroprocesori



Microprocessor Intel 4004 (CC Thomas Nguyen)

Prvi komercijalno dostupan (4 bit) mikroprocesor, Intel 4004 proizveden je u Sjedinjenim Američkim Državama 1971. godine. Prvobitno je bio korišćen u digitronima koje je proizvodila Japanska kompanija BUSICOM. Intel je 1972. godine proizveo prvi 8 bit mikroprocesor, intel 8008. Godinu dana kasnije, Toshiba je proizvela 12 bit mikroprocesor, TLCS-12. 1974. nastao je Intel 8080 koji je bio sastavljen od 6000 tranzistora. Po nekim izvorima 1971. godine napravljen je prvi mikrokontroler, TMS1802NC, za čiju su kreaciju zaslužni inženjeri Gary Boone i

Michael Cochran iz Texas Instruments kompanije što je dovelo u pitanje šta je mikroprocesor a šta mikrokontroler. U cilju bolje klasifikacije, danas koristimo termin SoC kao oznaku za sofisticiranije čipove.

2.2 Rani mikrokontroleri

Jedan od prvih mikrokontrolera je Motorola 6801, dostupan 1974. godine i razvijen na bazi mikroprocesora Motorola 6800 koristeći MOS. Nakon njega ubrzo je predstavljena i TMS 1000 serija mikrokontrolera. NEC μ COM-16 je prvi 16 bit mikroprocesor, napravljen 1974. godine.

Tipičan rani mikrokontrolerski (mikroračunarski) sistem zahtevao je veliki broj dodatnih kola za rad, kao što su A/D pretvarači (Engl. A/D converters), brojači, oscilatori i drugo.



Microcontroller Intel P8051 (CC Konstantin Lanzet)

Vremenom je došlo do integrisanja više komponenti u jedno kolo i tako su nastali moderniji mikrokontroleri. Jedan od prvih mikrokontrolera ovakvog tipa, koji je zbog svojih softversko-hardverskih karakteristika postao industrijski standard je Intel 8051, od koga je nastala MCS-51 serija koja je bila napravljena osamdesetih godina ali zbog njene popularnosti napravljene su modernije varijante. MCS-51 serija koristi NMOS, CISC (complex instruction set computer) instrukcioni set i nasledila je MCS-48 seriju koja je koristila CMOS i kojoj je pripadao Intel 8048, prvi Intelov mikrokontroler napravljen 1976. godine. Intel 8086 je 16 bit mikroprocesor dostupan 8. juna 1978, od ovog modela nastala je x86 arhitektura koja je

intelova najuspešnija linija procesora. Varijanta ovog mikroprocesora, Intel 8088 (dostupan 1.og. 1979.) korišćena je u dizajnu prvog IBM kompjutera. 1979. godine napravljen je prvi 32 bitni mikroprocesor, Motorola 68000. 32 bitna arhitektura prevaziđena je tek 1991. kada je napravljen 64 bitni mikroprocesor, MIPS R4000. 64 bit arhitektura ostaće dominantna još neko vreme jer nije bilo potrebe za 128 bitnim mikroprocesorima sem u eksperimentalne svrhe.

Na prvim mikrokontrolerima ROM (Read only memory) memorija bila je podešena od strane proizvođača. Nakon pronalaska PROM i EPROM memorija postalo je moguće da se korisnički programiraju mikrokontroleri, PROM (one time programmable ROM) memorija je mogla da se programira samo jedanput dok EPROM (erasable programmable ROM) može više puta.

Poznati tip memorije, UV-EPROM (ultraviolet EPROM) mogao je da se briše uparivanjem ultraljubičaste svetlosti ka transparentnom kvartznom prozoru u poklopcu paketa, međutim ovo je moglo da traje i do 20 minuta tako da nije bilo idealno. 1993 godine predstavljena je EEPROM (electrically erasable ROM) memorija sa kojom je memorija mikrokontrolera (počevši sa mikročipom PIC16C84) mogla biti električno obrisana brzo i jeftino. Iste godine Atmel je predstavio i flash memoriju.

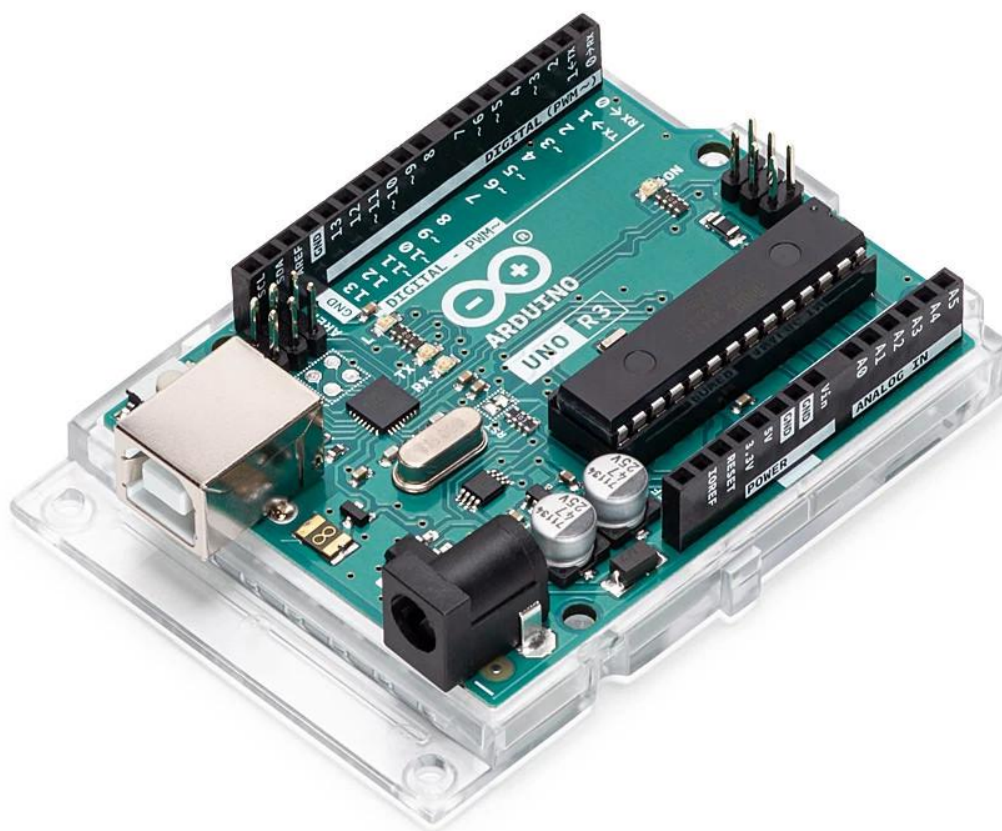


ATMega 382p

AVR familiju mikrokontrolera od 1996 godine razvijao je Atmel (deo Microchip Technology od 2016. godine). To je serija mikrokontrolera sa RISC (reduced instruction set) instrukcionim setom i modifikovanom harvard arhitekturom.

2.3 Istorija Arduino čipova

Atmega328p pripada Atmel porodici mikrokontrolera sa modifikovanom harvard arhitekturom. Postoji par varijanti ovog mikrokontrolera: originalni ATmega328, picoPower ATmega328P koji koristi manje energije i moderniji ATmega328PB koji ima više komponenti i funkcionalosti na pinovima.



Arduino UNO Rev 3

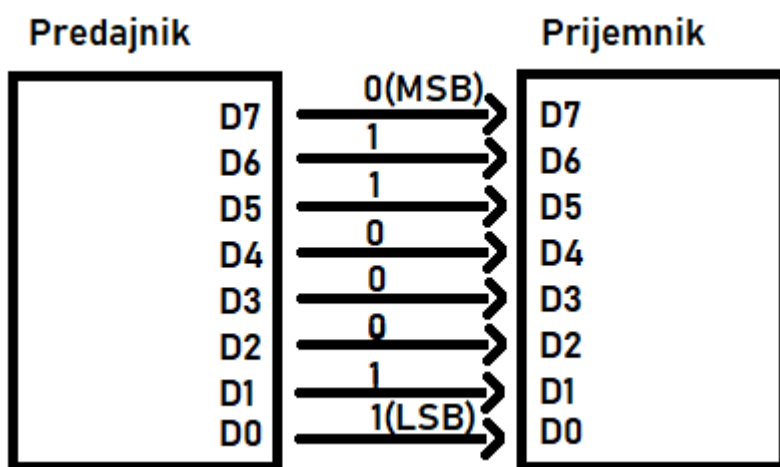
Arduino je započet 2005-te godine u Italiji kao projekat grupe studenata koji su hteli da naprave proste i jeftine alate sa kojima bi ljudi i bez inženjerskog iskustva mogli da prave digitalne projekte. Trenutno ima preko 17 zvaničnih ploča i većina je koristila kombinaciju flash i EEPROM memorije a prva ploča koju je Arduino proizveo bila je “Serial Arduino”, dostupna 30tog marta 2005. Još neke značajne ploče bile su LilyPad Arduino (dostupna 17tog oktobra 2007.) i Arduino Duemilanove (dostupna 19tog oktobra 2008). Prva revizija popularne Arduino

uno ploče dostupna je od 24tog septembra 2010. i bazirana je na atmega328p mikokontroleru. Dodatne funkcionalnosti dostupne su za sve a pogotovo za ovu ploču korišćenjem takozvanih štitova (Engl. shield) i dodatnih biblioteka. Prva Arduino ploča bazirana na ARM arhitekturi je Arduino Due, dostupna od 22og oktobra 2012. godine. Od oktobra 2017. godine Arduino je u partnerstvu sa ARM. Od 2016. godine moguće je vizuelno programirati Arduino ploče upotrebom XOD programskog jezika.

3. Paralelna komunikacija

Data link predstavlja način povezivanja dve lokacije zarad upostavljanja digitalne komunikacije tj primo-predaje informacija. Format komunikacije određuje protokol koji je u sloju veze OSI referentnog modela. Postoje bar tri osnovna tipa komunikacije koji se mogu koristiti:

- Simplex, komunikacija samo u jednom smeru.
- Polu dupleks, komunikacija u dva smera ali ne i istovremena.
- Potpuni dupleks, istovremena komunikacija u dva smera.

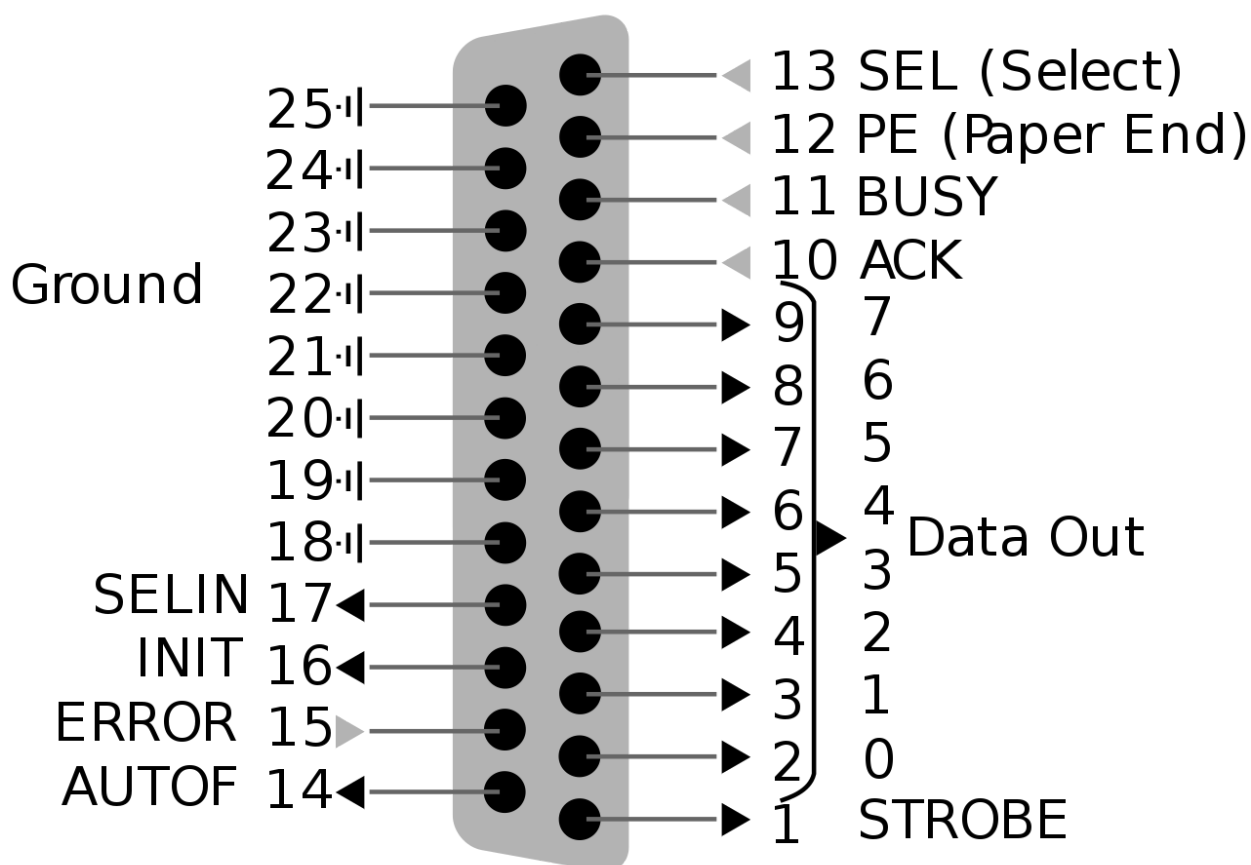


Paralelni interfejs, big endian

Paralelna komunikacija je metoda prenosa podataka gde se više bitova transimituju istovremeno u toku jednog taktnog signala putem više linija prenosa (za razliku od transmitovanja bita po bita sekvencijalno kao u morsovom kodu i BPSK modulaciji). Ovo se dešava na fizičkom sloju OSI referentnog modela, dobro poznate tehnike modulacije kao što su PSM, PAM i MIMO (Engl. Multiple-input multiple-output) šalju po nekoliko bitova u paraleli (svaka grupa bitova zove se simbol). Ove tehnike mogu se primeniti i za slanje celog bajta od jednom (Na primer 256-QAM). Zbog potrebnih dodatnih linija za prenos, paralelna komunikacija je obično half duplex. Pored linija za prenos podataka može biti dodatnih linija na primer za taktni signal. Ova metoda prenosa primenjuje se kada se podaci trebaju poslati na kraćoj razdaljini (obično do 6 metara). Primeri uređaja koji koriste paralelnu komunikaciju su CNC mašine, integrisana kola, RAM, periferni busevi u računaru kao ISA, rane verzije SCSI-a, PATA (Parallel ATA), PCI (Peripheral Component Interconnect), USB (Universal Serial Bus) 3.2 uređaji, IEEE-1284 štampači i drugi periferni uređaji.

Postoje dosta paralelnih portova, IEEE 1284 je bio de facto standard 1970tih do 2000tih godina, Od verzije 3.2 USB je vratio paralelnu komunikaciju u širokoj upotrebi. USB C je paralelni port i većina uređaja će se prebaciti na njegovo korišćenje, ali termin “paralelni port” i dalje najviše asocira na printer port tj Centronics port ili paralelni printer port (na IBM PC računarima) Ovaj port je bio standardizovan devedesetih godina gde je 1996. prva verzija USB-a bila objavljena i nije koristila paralelnu komunikaciju do USB 3.2 u 2017.

3.1 Istorija paralelne komunikacije



Pin out DB-25 paralelnog konektora (CC AndrewBuck)

IBM personalni računar napravljen je 1981. i imao je varijantu Centronics porta (koji je razvijen sedamdesetih). Ovaj interfejs je standardizovao paralelnu komunikaciju sa DB25F konektorom - ovo je 25-pin port koji ima 17 linija podataka za signale i 8 za uzemljenje. Sedamnaest linija za signale podeljeno je na:

- 4 linije za inicijaciju komunikacije i izlaznu kontrolu
- 5 linija za notifikaciju o greškama i izlaznu kontrolu
- 8 linija za transmisiju podataka

Ovaj port je bio dominantni IBM port do 1987. kada je IBM napravio bidirekcionni interfejs IBM PS/2. Paralelni interfejs je bio adaptiran za upotrebu na više vrsta perifernih uređaja pored štampača. Jedna od prvih upotreba za paralelni port bili su dongle uređaji, optički dražvovi poput CD čitača, čitača disketa, skenera, eksternih modema, gejmpadova i džojstika. Neki od ranih portabilnih mp3 plejera zahtevali su korišćenje paralelnog porta za prebacivanje pesama na uređaj. Drugi uređaji poput programera EPROM memorije i hardverskih kontrolera povezivali su se preko paralelnog porta i za povezivanje drugih uređaja postojali su adapteri za povezivanje - na primer SCSI uređaja preko paralelne komunikacije.

Na sistemima baziranim na DOS-u logički paralelni portovi detektovani na BIOS-u bili su dostupni operativnom sistemu pod inkrementalnim imenima poput LPT1, LPT2, LPT3,) Ova imena su skraćenice za termine poput line print terminal, local print terminal i line printer). LPT kao i par i drugih imena kao PRN, CON, AUX su zabranjeni nazivi datoteka i fascikla u DOS operativnom sistemu i sistemima baziranim na njemu, čak i u Windows 11 operativnom sistemu ne može se napraviti datoteka ili fascikla sa ovim imenima. Paralelna komunikacija je od uvek bila korišćena u integrisanim kolima, perifernim busevima i u memoriji. Direct control feature system/360, iz standard System/360 (1964 godina) varijante modela IBM računara i Laboratory Instrumentation bus IEEE-488 imali su 8-bitni port, IBM model 44, process-control varijanta imala je 32-bitni port. Busevi u računarima su prvobitno koristili paralelnu komunikaciju ali tokom vremena noviji računari prešli su na korišćenje serijske komunikacije za buseve.

I ako je zbog visoke upotrebe USB uređaja paralelna komunikacija bila većinski zamenjena serijskom, u skorijem vremenu je ponovo došlo je do povećanja upotrebe paralelne - nakon verzije 3.2 USB je postao paralelni port i paralelni data linkovi su videli uvećanu upotrebu u RF radio komunikaciji.

3.2 Clock skew

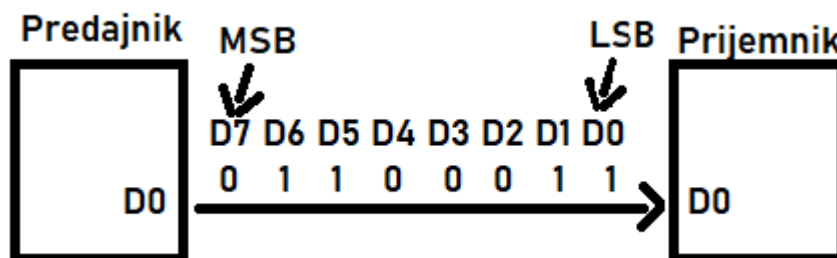
Clock Skew je pojava u paralelnoj komunikaciji koja ograničava brzinu prenosa transmisije na brzinu prenosa najsporije linije prenosa. Kada signali od istog izvora stignu do drugih komponenti u različito vreme, razlika između najbržeg i najsporijeg vremena se zove clock skew. Clock skew može napraviti da taktni signal signal prvo stigne na predajniku a zakasni na prijemniku (negativan skew) ili da prvo stigne na prijemniku a zakasni na predajniku (pozitivan skew). Nulti skew je kada signal stigne na vreme na primo-predajniku. Clock skew se dešava jer za razliku od idealnog scenarija svaka žica tj linija prenosa može imati (nenamerno) blago drugačije karakteristike tako da neki bitovi mogu stići pre drugih što može imati negativan uticaj na integritet poruke. Bit parnosti može pomoći u smanjenju ovog efekta ali i dalje se smatra da

paralelna komunikacija nije pouzdana pri prenosu podataka na većim distancama jer sa većom distancom raste mogućnost korumpiranih poruka. Pod uslovom da su R_i i R_j dva susedna registra a T_{Ci} i T_{Cj} vremena koja su potrebna signalu tih registra da stigne od izvora do destinacije clock skew se može definisati kao:

$$T_{\text{skew } i, j} = T_{Ci} - T_{Cj}.$$

Clock skew može negativno uticati na komunikaciju. Može se desiti da se na drugom registru zamene inicijalno prebačeni podaci i time se korumpiraju podaci koji se već bili tu i uništi integritet poruke. Može se desiti da clock skew pozitivno utiče na kolo ako se njime smanji period satnog signala i time smanji vreme koje je potrebno da kolo funkcioniše korektno. Za postizanje pozitivnog clock skew-a razvijeni su algoritmi za optimizovanje sa podešivim varijablama sa kojima se može postići takozvani optimalni skew.

4. Serijska komunikacija



Serijski interfejs, big endian

Serijska komunikacija je metoda prenosa podataka gde se jedan po jedan bit sekvencijalno transmituju preko jedne linije prenosa. Brzina prenosa preko te jedne linije može biti veoma visoka i može se lako koristiti i na većim distancama kao i jeftino implementirati full duplex komunikacija. Uz podatke lako se mogu poslati i bitovi za proveru podataka ili bit pariteta. Primeri uređaja koji koriste serijsku komunikaciju su kompjuterska tastatura i miš, PS/2, nove verzije SCSI-a i USB (do verzije 3.2 su USB portovi isključivo koristili serijsku komunikaciju). Može se naći više senzora, kamera i LCD monitora koji šalju i primaju podatke serijski. Serijska transmisija podataka je rešila mnoge probleme sa cenom i dostupnošću pinova.

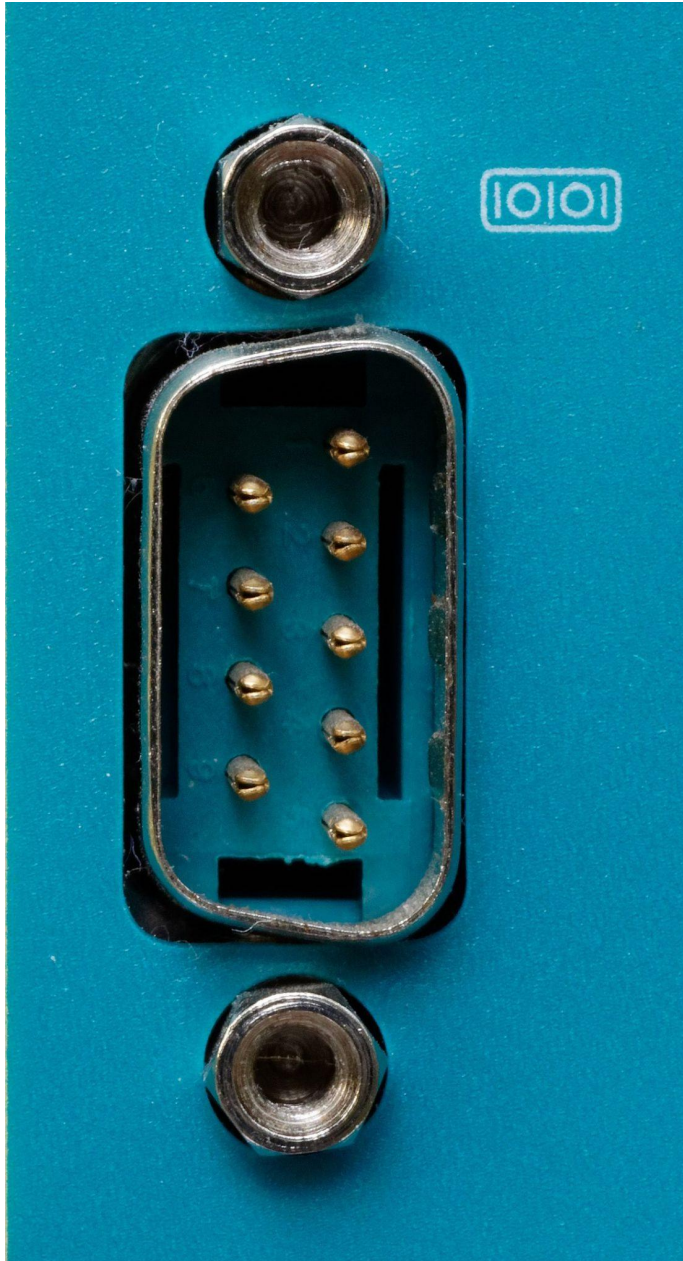
Postoje dve vrste serijskog prenosa: asinhroni kod koga se dodaje extra bit svakom bajtu podataka kako bi primalac bio obavešten o primanju novih podataka. 0 se koristi kao start bit a 1 kao stop bit. Tipična start-stop transmisija je ASCII (nasledio RS-232) koja je bila korišćena u pisaćim mašinama. U sinhronoj komunikaciji ne dodaje se extra bit svakom bajtu podataka već se podaci šalju u kontinualnom toku konstantnom brzinom budući da sinhrona komunikacija zahteva da satovi signala šalju podatke istom sinhronizovanom brzinom i primaocu i pošiljaocu. Može se desiti da ima potrebe da se resinhronizuju jer vremenom imaju tendenciju ka desinhronizaciji. Podaci se u serijskoj komunikaciji prebacuju u serijama i svaka sadrži više bajtova. Neki protokoli koji koriste sinhronu serijsku komunikaciju su SCSI, SPI i I2C. Sa druge strane asinhronu koriste RS-422, RS-485 i RS-232 (čije kablove sačinjavaju 25 žica ali samo dve se koriste za dvosmernu transmisiju podataka, druge se koriste za kontrolne signale).

Serijski prenos sastoji se od:

- Bitova sa podacima
- Sinhronizacionih Bitova

- Bitova pariteta

Standardne brzine prenosa u bodovima su: 300, 1200, 2400, 4800, 9600... 250000... 1000000, pa i 2000000 i ako Arduino uno podržava ovu brzinu samo u režimu dvostruke brzine.



D-subminiature konektor (CC Jud McCranie)

Ima mnogo različitih serijskih konektora. D-subminiature konektor ima 9 pinova u obliku slova D koji prebacuju podatke serijski. Drugi interfejsi poput Ethernet, FireWire, i USB do verzije 3.2 koriste serijsku komunikaciju ali termin serijski port se najčešće asocira sa D-9 konektorom ili hardware saglasan sa RS-232 standardom ili nekim drugim poput RS-485 ili RS-422. USB je zamenio većinu drugih konektora ali moguće je koristiti USB-to-serial konvertore kako bi se omogućila konekcija sa RS-232 i drugih klasičnim serijskim uređajima.

Arduino za konvertovanje podataka iz serijskih pinova RX i TX u USB format koristi poseban mikročip atmega16u2 ali dosta modernih uređaja koriste USART integrisano kolo kako bi implementirali serijski port.

4.1 Istorija serijske komunikacije

Početkom osamdesetih godina, kada je IBM predstavio prvi IBM računar, IBM i druge kompanije su ubrzo napravile R232 dodatne ploče kako bi omogućile konekcije računara ka spoljnim uređajima. Opadajuća cena i bolje performanse integrisanih kola dovele su do veće upotrebe serijske komunikacije nego paralelene. Jedna velika prednost serijske komunikacije bila je korišćenje manje žica i pinova za komunikaciju što je inicijalno omogućilo i pravljenje kompaktnijih uređaja. Primeri zamene paralelne komunikacije za serijsku su zamenila zamena IEEE 1284 štampačkih portova za USB, zamena parallel ATA sa serijskom ATA, zamena PCI za PCI Express. Uređaji za prebacivanje audiovizuelnih (AV) podataka kao što su digitalne kamere i profesionalni skeneri koji su ranije zahtevali posebnih konektora kao SCSI HBA su većinski standardizovani upotrebom serijske komunikacije u potrošačkim uređajima.

Jedan od prvih serijskih kompjuterskih komunikacionih uređaja bio je UART (universal asynchronous receiver-transmitter). Gordon Bell iz DEC-a dizajnirao je prvu verziju ovog uređaja kome je inicijalna upotreba bila povezivanje pisaćih mašina na interfejs komande linije kao i uređaja u ranom internetu. USART je kao deo integrisanog kola za serijski port standardizovao serijske portove, pre toga čak se povezivalo na izlazni pin mikroprocesora za izvršenje serijske komunikacije koji je često funkcionisao po zatvorenom nestandardnom principu. Glavni napredak koji je UART doneo bila je automatska obrada signala iz analognih u digitalne preciznim uzorkovanjem (Engl. sampling). UART na početku nije bio uređaj koji je stao na samo jednom čipu ali vremenom je DEC napravio kompresovanu verziju. Western Digital je 1971. godine unapredio ovo u prvi javno dostupan UART model, WD1402A, još jedan popularan čip bio je SCN2651 iz signetics 2650 familije.

Zavisno od proizvođača istorijski su korišćeni različiti termini za opis uređaja koji su imali UART sposobnosti. Intel je nazvao I8251 “programabilni komunikacioni interfejs, MOS Technology 6551 je bio poznat kao “adapter za interfejs asinhronne komunikacije”. Termin “serijski komunikacioni interfejs” (Engl. SCI) iskoristila je Motorola 1975. godine za njihov start-stop uređaj sa asinhrono serijskim interfejsom koji je isto bio poznat kao UART. Zilog je napravio više komada “kontrolera za serijsku komunikaciju” (Engl. SCC).

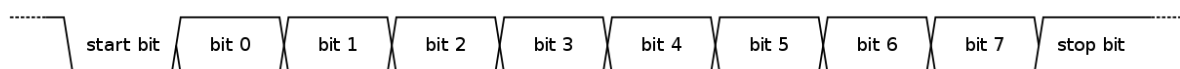
Osamdesetih godina napravljen je National Semiconductor 8250, popularan model UART-a koji je dosta puta kloniran. Jedna njegova upotreba bila je dodavanje asinhronne komunikacije preko adapterske kartice u originalnim IBM računarima. Devedesetih godina noviji UARTovi razvijeni su sa baferima integrisanim u čipovima koji su omogućili podršku većih brzina transmisije bez gubitka podataka.

UART je podržavao samo asinhronu transmisiju podataka; sličan uređaj koji je nastao kasnije, USART (universal synchronous and asynchronous receiver-transmitter) podržava i sinhronu i asinhronu transmisiju podataka. Serijske sposobnosti USART-a su originalno namenjene za protokole poput IBMovog STR (synchronous transmit-receive), BSC (binary synchronous communications) (BSC), SDLC (synchronous data link control), kao i ISO-standard HDLC (high-level data link control). U ranijim vremenima najbrži asinhroni modem mogao je raditi pri brzini od 300 bita u sekundi koristeći FSK (frequency-shift keying) modulaciju dok su sinhroni modemi radili brzinama do 9600 bit/s koristeći PSK (phase-shift keying). Ovi protokoli su bili napravljeni tako da naprave najbolji protok sa tadašnjim analognim modemima. Sinhrona transmisija mogla je da koristi malo preko 80% protoka asinhronne transmisije jer su start i stop bitovi bili nepotrebni.

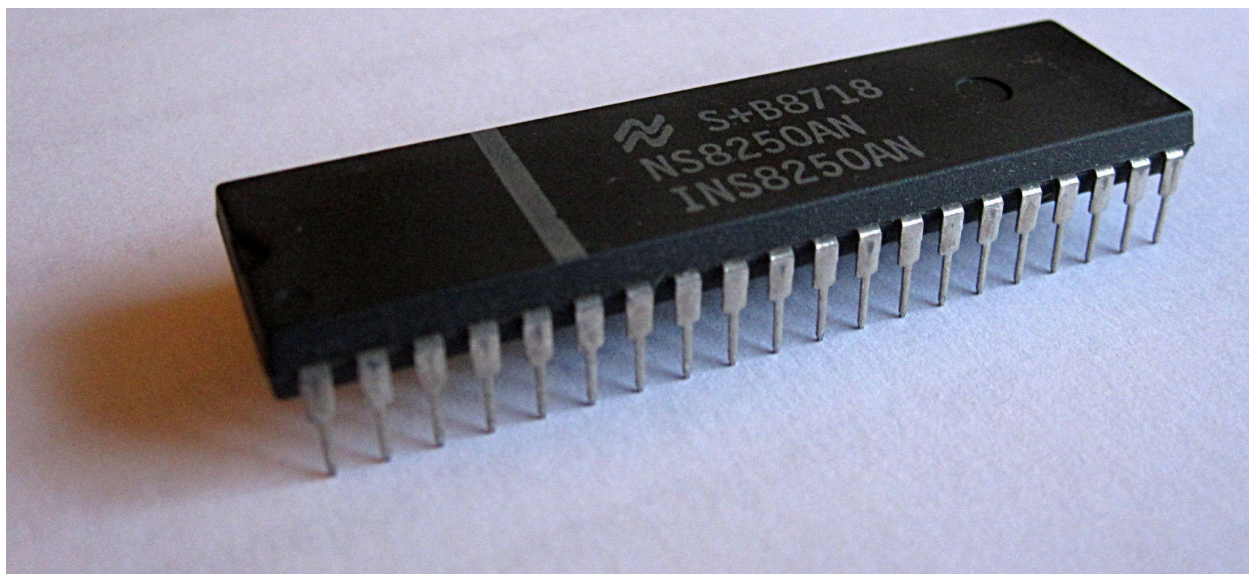
Početkom 2000ih godina većina računara kompatibilnih sa IBM-ovim računarom su zamenili RS-232 COM portove sa USB portovima koji mogu slati podatke mnogo brže. Zarad kompatibilnosti sa starijim uređajima napravljeni su eksterni uređaji gde čip konvertuje sa USB na UART za korisnike kojima je RS-232 i dalje bio potreban. Cypress Semiconductor i FTDI su jedni od najvećih komercijalnih distributera ovih čipova. Iako RS-232 portovi više nisu ugrađeni na modernim računarima, većina mikroprocesora ima UART stepen hardverski ugrađen u njima sa kojim se može komunicirati RS-232 ili RS-485 protokolima preko eksternih uređaja.

4.2 USART

USART (universal synchronous and asynchronous receiver-transmitter) je vrsta uređaja za komunikaciju preko serijskog porta koji je zasebno integrisano kolo ili deo nekog postojećeg koje se može programski podesiti da radi u sinhronom ili asinhronom režimu (ista funkcionalnost kao i UART). Takođe se mogu programski podesiti format i brzina prenosa.



USART konvertuje karaktere iz i u asinhroni serijski format i poštuje specifikacije serijskog protokola. Koriste se kao serijski port i mogu povezati USB sa serijskim portom. U IBM računarima serijski portovi su implementirani sa jednim ili više UART stepenima.



National Semiconductor 8250 UART (CC Nixdorf)

Rani model UARTA iz osamdesetih godina bio je National Semiconductor 8250 koji je korišćen u adapterskoj kartici za asinhronu komunikaciju u originalnim IBM računarima. Primeri uređaja koji koriste USART su automobili, smart i sim kartice, Intel 8251A i drugi mikrokontrolerski čipovi. USART se koristi u raznim protokolima kao što su RS-232 koji je 12 voltni sistem i RS-485 koji je 5 voltni sistem. Bitovi se šalju jedan po jedan, od najmanje značajnog bita do najznačajnijeg. U sinhronom radnom režimu bitovi podataka su uokvireni start i stop bitovima tako da se osigurava vremenski precizna komunikacija - ukoliko procesor ne prosledi karaktere USART-u do početka transmisije nekog okvira, transmisija tog okvira se otkazuje i dolazi do underrun greške. Kroz sinhronu liniju prenosa uvek prolaze neki bitovi čak i ako nisu uokvireni bitovi podataka jer će USART u zavisnosti od uređaja i protokola slati karaktere ili bitove kao indikaciju da je modem aktivan.

Slični protokoli za sinhronu serijsku telekomunikaciju postoje poput veoma rasprostranjenog IEEE 802.2 (Ethernet) protokola, link level u OSI referentnom modelu. USART stepeni se i dalje u nekim mikrokontrolerima integrišu i koriste u ruterima koji se mogu povezati na eksternim CDU/DSU uređajima i najčešće koriste Cisco HDLC implemetanciju zatvorenog tipa ili IETF standard PPP (Engl. point to point) uređaje sa protokolima koji uokviraju podatke slično kao HDLC kao što je definisano RF 1662 standardom.

5. Serijska komunikacija arduino uno uređaja

Povezivanjem dodatnih uređaja na Arduino ploče omogućeno je korišćenje dodatnih funkcionalnosti iz tih uređaja, na primer serijski se mogu povezati mikrokontroleri ESP8266, ESP32 ili GSM modul i dodati IoT (Internet of Things) mogućnosti.

USART stepen ATmega328p mikrokontrolera omogućava mikrokontroleru da primi podatke preko serijske komunikacije dokle god 64 bajtni serijski buffer nije pun. USART stepens ATmega328p mikrontrolera obezbeđuje:

Potpuni dupleks (nezavistan rad predajnika i prijemnika)

Parnosti

Tri izvora prekida

Većina kompleksnih integrisanih kola (Engl. IC) za sinhronizaciju između delova kola koristi taktni signal. To je specifičan tip signala koji osciluje između visokog i niskog stanja i koristi se kao metronom radi koordinisanja rada strujnih kola. Jedini izuzeci koji ne koriste taktni signal su asinhrona kola poput asinhronih mikroprocesora.

Asinhroni režim serijske transmisije koristi start i stop bitove da bi obeležio početak i kraj transmije podataka; ova metoda transmisije podataka se koristi kada se podaci šalju sa prekidima umesto konzistentno.

Sinhroni režim serijske transmisije zahteva da se podaci šalju i čitaju istom brzinom inače dolazi do korumpiranja poruka. To radi tako što se sinhronizuju brzine prenosa na tački slanja i prenosa koristeći clock signal. On može biti poseban signal ili uključen u podatke u zavisnosti od formata prenosa. Konzistentan tok podataka se šalje preko linije prenosa, brzina prenosa podataka je efikasnija jer nema bitova za start i stop.

Brzine dostupne za prenos u Arduino serial monitoru idu od 300 bodova do čak 2000000 bodova; međutim, zbog loše optimizacije serijskog koda u Arduino bibliotekama teško je dobiti brzinu prenosa podataka preko 500 KB/s čak i u kratkim intervalima ukoliko ne kreiramo sopstvene funkcije ali tada gubimo mogućnost korišćenja Arduino metoda vezanih za Serijski port.

Komunikacija sa spoljnim svetom je preko pinova PD0 (RX) i PD1 (TX)

ATmega16U2 čip u Arduino Uno sistem omogućava povezivanje računara i serijskog porta, tačnije pinove 0 i 1 Atmega328p mikrokontrolera preko USB-a. Ranije verzije Arduino Uno i Arduinio Mega 2560 koristile su ATmega8U2 čip.

Preko serijskog porta se novi sketchevi prepisuju na arduino uno. Kada se atmega 328p čip na arduinu resetuje, bootloader proveri da li na serijskoj liniji ima novi program koji čeka da se instalira - ako ima novog programa na serijskoj liniji bootloader ga prepíše preko postojećeg, u slučaju da nema, sistem se upali sa prethodnim programom i serijski port radi uobičajeno.

5.1 ATmega328p registri

Moguće je u Arduino IDE pristupiti registrima atmega328p mikrokontrolera. Neki od osnovnih i bitnih registara su:

- **UBRR0**

Generalno serijski uređaji će tolerisati grešku u stopi bodova do 5%, kako oba kraja mogu imati grešku u suprotnom smeru poželjno je da ni jedan nema grešku veću od 2.5%

Tabela 5.1 UBRRn Settings for Commonly osc = 16.0000MHz Used Oscillator Frequencies

Stopa bodova	U2X0 = 0		U2X0 = 1	
	UBRR0	Greška	UBRR0	Greška
2400	416			
	207			
	103			
	68			
	51			
	34			
	25			
	16			
	12			
	8			
	3			
	3			
	1			
	0			
Maksimalna	1Mbps		2Mbps	

Dok je teorijski moguće ostvariti brzine i do 2M bodova u kratkim intervalima, u praksi je sa podrazumevanom Arduino serial bibliotekom teško postići brzine veće od 0.5M bodova ali je moguće ukoliko se napišu sopstvene funkcije.

```
#define USART_BAUDRATE 9600
```

```
#define UBRR_VALUE ((F_CPU / (USART_BAUDRATE * 16UL)) - 1)
```

```
UBRR0 = UBRR_VALUE;
```

- **UCSR0C**
- **UCSR0B**
- **DDRx : Data Direction Register for port 'x'**

The DDRx register is responsible for initializing the pins for use either as inputs or outputs.

A 1 bit signifies a pin initialized as output. A 0 bit signifies a pin initialized as input.

DDRD = 0b01000100 initializes pins 2, 6 as output, pin 0,1,3,4,5,7 as input.

- **PORTx : Defines state of Output Pins on port 'x'**

The PORTx register determines whether the output state of a pin is HIGH or LOW.

A 1 bit signifies HIGH and 0 signifies LOW.

DDRx takes precedence over PORTx. A pin previously defined as input won't be affected by PORTx. The pin must be initialized as an output by DDRx first to manipulate it through PORTx.

DDRD = 0b01000100

PORTD = 0b01000000 //declares pin 2 LOW and pin 6 HIGH.

- **PINx : Reads state of Input Pins on port 'x'**

The PINx register simply reads the value from the pins. This value, as obvious, is digital.

Although Port C pins have ADCs attached, they can still be used as digital GPIOs

- **EIMSK**
- **EICRA**

5.2 Arduino metode

Navešćemo neke od osnovnih i bitnih metoda pri korišćenju serijske komunikacije Arduino Uno sistema:

- **Serial.begin(stopova, config(neobavezno))**

Pokreće serijsku komunikaciju, argument podešava brzinu prenosa bodova, podrazumevano se prvo proba režim duple brzine (U2X0 = 1). Nakon upotrebe ove metode digitalni pinovi 0 (RX) i 1 (TX) se ne mogu koristiti kao IO pinovi.

Primer upotrebe:

→ `Serial.begin(9600);` // Pokreće serijsku komunikaciju sa stopom bodova 9600 i podrazumevanim formatom SERIAL_8N1

- **Serial.end()**

Zaustavlja serijsku komunikaciju i omogućava upotrebu digitalnih pinova 0 (RX) i 1 (TX) kao IO pinove.

- **Serial.print(podaci, brojevni sistem(neobavezno))**

Šalje brojeve, karaktere i stringove preko serijskog porta enkodirane u ASCII formatu tako da se mogu lako pročitati, mogu se proslediti funkcije koje vraćaju stringove poput F (podaci). Ako se prosledi celobrojni broj može se proslediti brojevni sistem kao argument (BIN - 2, OCT - 8, HEX - 16, DEC - 10 (podrazumevano). Ukoliko se prosledi floating point broj može se proslediti broj decimalnih mesta (podrazumevana vrednost je 2). Vraća broj bajtova koji su poslali.

Primeri upotrebe i izlazne vrednosti:

→ `Serial.print("10");` // Izlaz: 10

→ `Serial.print(10);` // Izlaz: 10

→ `Serial.print(10, HEX);` // Izlaz: A

→ `Serial.print(1.0123, 0);` // Izlaz: 1

→ `Serial.print(1.0123, 1);` // Izlaz: 1.0

→ `Serial.print(1.0);` // Izlaz: 1.00

- **Serial.println(podaci, brojevni sistem(neobavezno))**

Radi isto što i print metoda i na kraju pošalje karaktere `\r\n` koji dodaju novi red.

Primer upotrebe i izlazne vrednosti:

→ `Serial.println("Zdravo");` // Izlaz: Zdravo\r\n (\r\n se vidi kao novi red)

- **Serial.write(bajt)**

Šalje brojeve, karaktere, stringove i nizove preko serijskog porta i ne enkodira ih, pravi razliku između "10" i 10. Ukoliko se prosledi pokazivač na niz treba se proslediti kao drugi argument koliko vrednosti da pročita iz niza. Vraća broj bajtova koji su poslali.

Primeri upotrebe i povratne vrednosti:

→ `size_t num = Serial.write(65);` // Šalje bajt 1000001 tj slovo A na serijski port, num je 1

→ `size_t pom = Serial.write("hello");` // Šalje niz bajtova na serijski port, pom je 5

→ `uint8_t p[3] = {65, 66, 67}; Serial.write(p, 3);` // Šalje ABC na serijski port

- **Serial.available()**

Vraća broj bajtova koji su primljeni preko serijskog porta i čekaju da se pročitaju sa serijskog bafera.

Primer upotrebe i povratne vrednosti:

`int bafer = Serial.available();` // Vraća 64 kada je prazan

- **Serial.read()**

Čita jedan bajt iz serijskog bafera.

Primer upotrebe i povratne vrednosti:

`uint8_t received_data = Serial.read();` // Popuje 1 bajt iz serijskog bafera

- **Serial.peek()**

Vraća sledeći bajt (karater) iz serijskog bafera bez toga da ga izbaci iz bafera.

- **Serial.availableForWrite()**

Vraća broj bajtova (karaktera) koji se mogu upisati u transmisioni serijski bafer bez da se blokira slanje. Serijska komunikacija u arduino sistemima je asinhrona, što znači da ukoliko ima prostora u transmisionom serijskom baferu `Serial.print` i `Serial.write` metode će upisati karaktere tamo i vratiti povratne vrednosti pre nego što se bilo koji karakter

pošalje. Ako je transmisioni serijski bafer pun onda će ove funkcije blokirati izvršenje drugih operacija dok ne bude bilo dovoljno prostora u baferu.

Primer upotrebe i povratne vrednosti:

```
int num = Serial.availableForWrite(); // Vraća 63 kada je prazan
```

- **Serial.flush()**

Sačeka da slanje serijskih podataka završi tj. da se isprazni transmisioni serijski bafer.

5.3 Format

U asinhronom modu serijske komunikacije mora se odlučiti format transmisije. Dostupne opcije su:

- Potpuni ili polovični duplex
- Broj bitova po karakteru (obično 8 bita ali neki raniji transmiteri su koristili 5, 6 ili 7 bit ASCII)
- Endian redosled bitova
- Brzina bitova po sekundi prenosne linije (ekvivalentna stopi bodova kada svaki simbol predstavlja jedan bit). Neki sistemi mogu koristiti automatski detekciju brzine (stope bodove).
- Da li se koristi bit pariteta
- Parna ili neparna parnost, ukoliko se koristi bit pariteta
- Broj stop bitova (mora se poslati makar onoliko koliko prijemnik zahteva)
- Simboli oznake i razmaka (istorijski tok struje u ranoj telegrafiji, polaritet voltaže u EIA, RSA-232, polaritet promene frekvencije u FSK, itd)

U Arduino sistemu za prenos podrazumevano se koristi 1 start bit, 8 bitova podataka, ni jedan bit pariteta i 1 stop bit, popularna stopa bodova je 9600, drugi formati se mogu podesiti putem UBRR0 i UCSR0C registra ali najčešće je u upotrebi format 9600/8-N-1. Brzina od 9600 bodova u ovom formatu znači da će se poslati 10 bitova po transmisiji od kojih su 8 bitovi podataka, što znači da je brzina prenosa podataka zapravo 80% od brzine transmisije.

5.4 Softverski i hardverski serial

Arduino uno ima hardversku podršku za serijsku komunikaciju preko pinova 0 i 1. Softverski se može emulirati serijska komunikacija i preko ostalih digitalnih pinova pomoću dodatnih

biblioteka poput SoftwareSerial (bazirana od verzije 1.0 na NewSoftSerial biblioteci). Korišćenjem SoftwareSerial biblioteke moguće je imati više softverskih serijskih portova sa brzinama do 115200 bps, moguće je koristiti i invertovati signale za uređaje koji to zahtevaju.

SoftwareSerial ima određena ograničenja:

- Ne mogu se istovremeno slati i primiti podaci
- Ako se koristi više softverskih serijskih portova samo jedan port može u datom trenutku primiti podatke.
- Na određenim pločama na primer Arduino Micro i Mega 2560 nema podrške na svim pinovima za prekide po promeni logičkog stanja.

Da bi se koristila ova biblioteka mora se prvo dodati u projekat

```
#include <SoftwareSerial.h>
```

Alternativna biblioteka koja podržava istovremeno primanje i slanje podataka preko više softverskih serijskih portova je AltSoftSerial. Ova biblioteka prevazilazi neke fundamentalne probleme sa SoftwareSerial bibliotekom, ali ima drugih sopstvenih ograničenja..

5.5 Preglednost podataka za ljude ili mašine

U ovom radu poslati bitovi biće štelovani za lakše detektovanje i analiziranje na mašinama poput osciloskopa umesto standardnom izlazu terminala. Funkcija Serial.print() iskorišćena je na više mesta umesto Serial.println() gde bi bilo adekvatnije ili je Serial.println() izostavljena.

6. Poređenje paralelne i serijske komunikacije

I paralelna i serijska komunikacija imaju svoje prednosti i mane. Osnovna razlika između njih je broj električnih konduktivnih linija koji se na fizičkom sloju osi referentnog modela koriste za prenos bitova. U paralelnoj komunikaciji može se poslati više bitova istovremeno; obično se šalje 2 do 8 bitova (jedan bajt) po prenosu što znači da je transmisija podataka paralelnom komunikacijom brža nego transmisija istih podataka serijskom komunikacijom koja šalje jedan bit podatka po prenosu. Za uzvrat, serijska implementacija ima svoje dobre strane. Jedna velika prednost serijske komunikacije je smanjeni broj žica u kablovima i pinova potrebnih za prenos što značajno smanjuje veličinu i kompleksnost konektora i smanjuje cenu implementacije. Dizajneri uređaja poput pametnih telefona imaju veliku korist od konektora/portova koji su mali po veličini, izdržljivi a i dalje imaju visoke performanse.

Tabela 6.1 Poređenje paralelne i serijske komunikacije

Paralelna komunikacija	Serijska komunikacija
Skuplja implementacija jer koristi više linija prenosa	Jeftinija implementacija jer koristi jednu liniju prenosa
Veća brzina prenosa, prenosi 1 bajt po taktom signalu ali je podložna clock skewu	Manja brzina prenosa, prenosi 1 bajt po taktom signalu ali je konzistentnija
Mogućnost ometanja signala ako su žice međusobno preblizu jedna drugoj	Manje potencionalnih smetnji signalu
Primenljivija na kraćim distancama zbog većeg broja žica	Primenljiva za bilo koju distancu
Obično full duplex	Obično half duplex
Jednostavnija implementacija	Kompleksnija implementacija

Detaljna poređenja:

- Brzina: Na prvu pomisao, brzina prenosa paralelnog data linka jednaka je broju bitova (linija prenosa) koji se šalju u jednom momentu, pomnoženo sa stopom prenosa bitova (brojem bitova koji svaka linija šalje u jedinici vremena) tako da u teoriji dodavanjem još jedne linije prenosa može se duplirati stopa prenosa podataka. U praksi, clock skew ograničava stopu prenosa bitova na brzinu prenosa najsporije linije prenosa što znači da brzina prenosa nije tačno proporcionalna broju linija prenosa. Zbog ovoga brzina prenosa

jedne linije prenosa u serijskoj komunikaciji može biti veća nego brzina prenosa individualnih linija prenosa u paralelnoj komunikaciji. Serijska komunikacija je obično full duplex a paralelna je half duplex.

- Pouzdanost: Crosstalk, pojava gde električni signal jedne transmisije utiče na drugi pravi smetnje u linijama prenosa paralelne komunikacije. Paralelna komunikacija je isto podložna clock skew-u. Na manjim brzinama prenosa i distancama ovi efekti su zanemarljivi ali zbog njih serijska komunikacija je pouzdanija na većim brzinama prenosa i distancama.
- Veličina kablova: Na većim daljinama serijska komunikacija je bolji izbor u poređenju sa paralelnom, zbog toga uređaji koji koriste paralelnu komunikaciju imaju kablove koji su obično kraći od kablova uređaja koji koriste serijsku komunikaciju. Razlog za ovo su Crosstalk i Clock skew; što je veća distanca prenosa to će Crosstalk i Clock skew imati veći uticaj na prenete podatke. Zbog ovoga postoji gornja granica veličine kabla uređaja koji koriste paralelnu komunikaciju.
- Kompleksnost: Paralelni prenos podataka se lako hardverski implementira. Napraviti paralelni port je relativno prosto jer sve što je potrebno je električna sklopka koja će prekopirati podatke na magistralu podataka, tako da je paralelna komunikacija logični izbor u projektima sa manje resursa. Većina podataka prebačenim serijskom komunikacijom mora se prvo konvertovati UART uređajem nazad u format podataka paralelne komunikacije pre nego što se kabli može povezati direktno na magistralu podataka.

6.1 Korišćena oprema

Za svrhe ovog eksperimenta potrebno je

2 x Arduino Uno R3

10 x Konektora muško na muški

6.2 Podešavanja i merenja paralelne komunikacije

the same sourced [clock signal](#) arrives at different [components](#) at different time

6.3 Podešavanja i merenja serijske komunikacije

6.4 Poredjenje rezultata merenja

7. Serijska komunikacija između dva Arduino Uno uređaja

Polu dupleks

7.1 Korišćena oprema

Za svrhe ovog eksperimenta potrebno je

2 x Arduino Uno R3

Protoploča

7.2 Podešavanje prijemnika

Zarad lakšeg programiranja prijemnika koristićemo funkcije i metode dostupne uz arduino jezik. Za početak, da bi upostavili serijsku komunikaciju sa stopom bodova od 9600 koristićemo metodu `Serial.begin(9600)`; Iskoristićemo je u funkciji `setup()` koja se pokreće prilikom svakog učitavanja programa. Ova funkcija se pokreće sa memorijske lokacije 0 i kada se vrši softverski reset, tako da ćemo za simulaciju vraćanja u početno stanje u njoj odrediti i da se postave svih 20 (uključujući analog pinove) pinova dostupnih na Arduino uno kao input pinovi.

```
void setup() {  
    Serial.begin(9600);  
    for (uint8_t i = 0; i < 20; i++)  
        pinMode(i, INPUT);  
}
```

Kako će nam pinovi na početku programa biti podešeni u input režimu napisaćemo funkciju koja omogućava menjanje režima pin; mogući režimi su `input (0)`, `output (1)` i `input_pullup (2)` a parametri se primaju u serijski bafer preko RX pina.

```
void pinmode() {  
    uint8_t pin = Serial.read();  
    pinMode(pin, Serial.read());  
}
```

Napisaćemo i funkciju za promenu logičkog stanja na pinovima. Ova funkcija će zavisno od bitova u serijskom baferu izvršiti analogWrite koji prima vrednosti između 0 i 255 ili digitalWrite koji prima vrednosti 0 i 1

```
void pinwrite() {  
    uint8_t pin = Serial.read();  
    uint8_t value = Serial.read();  
  
    if (Serial.read())  
        analogWrite(pin, value);  
    else  
        digitalWrite(pin, value);  
}
```

Sledeće što nam je potrebno je funkcija koja će pročitati logičko stanje na pinovima i poslati ga preko TX pina; zavisno od bitova u serijskom baferu ova funkcija će izvršiti ili analogRead i poslati karakter između 0 i 255 ili digitalRead i poslati karakter 0 ili 1

```
void pinread() {  
    uint8_t pin = Serial.read();  
  
    if (Serial.read())  
        Serial.print(analogRead(pin));  
    else  
        Serial.print(digitalRead(pin));  
}
```

Za kontrolu toka napravićemo funkciju koja pravi razmak u radu programa u milisekundama. Arduino delay funkcija potpuno pauzira izvršenje programa (sem prekida) što može biti loše.

```
void delayf() {  
    delay((Serial.read() << 8) + Serial.read());  
}
```


Napisaćemo i funkciju koja će praviti razmak u radu programa u mikrosekundama - ova funkcija koristi timer integrisan u atmega328p. Najveća vrednost sa kojom će ova funkcija proizvesti precizan razmak je 16383, tako da za veće razmake treba koristiti običnu delay funkciju

```
void delaymicroseconds() {  
    delayMicroseconds((Serial.read() << 8) + Serial.read());  
}
```

write funkciju ćemo koristiti za slanje bajta bez konvertovanja u karakter preko serijskog porta.

```
void writef() {  
    Serial.write(Serial.read());  
}
```

Napisaćemo i naprednu funkciju koja će pročitati logičko stanje na pinu i invertovati ga

```
void pinchange() {  
    uint8_t pin = Serial.read();  
    digitalWrite(pin, !digitalRead(pin));  
}
```

Sledeća funkcija će staviti visoko logičko stanje na pin i nakon određenog vremena staviti nisko logičko stanje

```
void pinclick() {  
    uint8_t pin = readw();  
    digitalWrite(pin, HIGH);  
    delayf();  
    digitalWrite(pin, LOW);  
}
```

Da bi iskoristili ove funkcije napravićemo niz funcs sa pokazivačima na njihove memorijske adrese, kao i memorijsku adresu reset funkcije (0)

```
void (*funcs[])() = {  
    pinmode,
```

```
pinwrite,  
pinread,  
delayf,  
delaymicroseconds,  
0,  
writef,  
pinchange,  
pinclick  
};
```

U funkciji koja se konstantno pokreće u Arduinu proveravaćemo da li su poslani neki bajtovi preko serijskog porta i ako jesu proslediti taj bajt prethodno napravljenom nizu. Bajt 0 biće pinmode u funcs nizu. Ova implementacija pretpostavlja da će se na prijemniku uvek slati korektni podaci za svaku funkciju.

```
void loop() {  
    if (Serial.available())  
        (*funcs[Serial.read()]());  
}
```

7.3 Podešavanje predajnika

Provera podataka

7.4 Tok komunikacije

8. Zaključak

Microcontrollers usually contain from several to dozens of general purpose input/output pins (GPIO). Mikrokontroleri su od sedamdesetih godina prošlog veka imali brz razvoj. Sve više i više ranije odvojenih kola je integrisano, programiranje je olakšano uvođenjem „flaš“ (flash) memorija (koje se mogu mnogo puta brisati i pisati), smanjena je potrošnja struje (važno za baterijske uređaje), a ponuda različitih kontrolera je neverovatno široka. Danas takođe postoje i 8, 16, 32 i 64 bitni modeli, kao i DSP (DSP) modeli podešeni za brze matematičke operacije sa proširenim setom instrukcija. On 21 June 2018, the "world's smallest computer" was announced by the [University of Michigan](#). The device is a "0.04mm³ 16nW wireless and batteryless sensor system with integrated [Cortex-M0+](#) processor and optical communication for cellular temperature measurement." It "measures just 0.3 mm to a side—dwarfed by a grain of rice. [...] In addition to the RAM and [photovoltaics](#), the new computing devices have processors and [wireless transmitters and receivers](#). Because they are too small to have conventional radio antennae, they receive and transmit data with visible light. A base station provides light for power and programming, and it receives the data."^[24] The device is 1/10th the size of IBM's previously claimed world-record-sized computer from months back in March 2018,^[25] which is "smaller than a grain of salt",^[26] has a million transistors, costs less than \$0.10 to manufacture, and, combined with [blockchain](#) technology, is intended for logistics and "crypto-anchors"—[digital fingerprint](#) applications.^[27] Micro-controllers have proved to be highly popular in [embedded systems](#) since their introduction in the 1970s. Manufacturers have often produced special versions of their micro-controllers in order to help the hardware and [software development](#) of the target system. Microcontrollers must provide [real-time](#) (predictable, though not necessarily fast) response to events in the embedded system they are controlling. A typical home in a developed country is likely to have only four general-purpose microprocessors but around three dozen microcontrollers. A typical mid-range automobile has about 30 microcontrollers. They can also be found in many electrical devices such as washing machines, microwave ovens, and telephones.

Literatura

- [1] ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET
- [2] Intel 8251A Programmable Communications Interface
- [3] Mikrokontroleri arhitektura 8051
- [4] <https://en.wikipedia.org/wiki/Microcontroller>
- [5] Programming Arduino: Getting Started with Sketches, Second Edition
Autor: [*Simon Monk*](#)
- [6] en.wikipedia.org/wiki/Microprocessor_chronology
- [7] David R. Smith, "Digital Transmission Systems", Kluwer International Publishers, 2003, [ISBN 1-4020-7587-1](#)
- [8] www.atmel.com
- [9] Colin A Programming for Microprocessors, 1979
- [10] Sergio Benedetto, Ezio Biglieri, "Principles of Digital Transmission: With Wireless Applications", Springer 2008, [ISBN 0-306-45753-9](#), [ISBN 978-0-306-45753-1](#)
- [11] www.arduino.cc/en/Main/Products
- [12] *Arduino For Dummies*; 2nd Ed; John Nussey; John Wiley & Sons; 400 pages; 2018; [ISBN 978-1119489542](#).
- [13] Intel, Embedded Microcontrollers 1996
- [14] www.brainboxes.com/files/pages/support/white-papers-and-presentations/Brainboxes%20Serial%20Products%20and%20Software%20Overview.pdf
- [15] www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf
- [16] Atmel, Microcontroller Data Book 1997
- [17] www.intel.com

- [18] commons.wikimedia.org/wiki/File:Pinout_of_ARDUINO_Board_and_ATMega328PU.svg
- [19] www.asciitable.com
- [20] Friedman, Eby G. (May 2001). "[Clock Distribution Networks in Synchronous Digital Integrated Circuits](#)" (PDF). *Proceedings of the IEEE*. **89** (5): 665–692. [CiteSeerX 10.1.1.7.7824](#). doi:[10.1109/5.929649](#). Archived from [the original](#) (PDF) on 2015-06-01. Retrieved 2017-01-09.

Sažetak / Abstract rada

Na kraju rada na posebnoj stranici potrebno je odštampati i naziv rada sa imenom kandidata sa sažetkom (*Abstract*) rada.

NAZIV RADA NA SRPSKOM

(MAX. DVA REDA)

Ime studenta / Mentor

Sažetak – Ove instrukcije su date kao smernice za pisanje diplomskih (završnih) radova u Visokoj tehničkoj školi u Nišu.

TITLE OF THE PAPER IN ENGLISH

(MAX. TWO LINES)

Name of student / Mentor

Abstract – *These instructions are for submitting a manuscripts for the final work in The College of Applied Technical Sciences in Nish and all students their manuscripts/works must be typewritten according to the text of these instructions.*

Biografija

Filip Stojanović rođen je 30.07.1998. godine u Nišu, Republika Srbija. Osnovnu i srednju školu završio je u Nišu. Nosilac je Vukove diplome za postignut uspeh u toku školovanja, kao i većeg broja diploma sa učešća na brojnim republičkim takmičenjima.

Visoku tehničku školu u Nišu, smer Savremene računarske tehnologije, upisao je školske 2017/18 godine. Za vreme studija učestvovao je u radu VTŠ Apps Tima Visoke tehničke škole i poseduje sertifikat o postignutim rezultatima.

Od 1.10.2014. do 1.12.2014. godine bila je zapošljena u preduzeću VTŠ Soft kao php programer.

Autor ili koautor je 5 radova objavljenih u zborniku radova Visoke tehničke škole u Nišu. Učestvovala je na domaćim i međunarodnim naučno-stručnim konferencijama sa radovima koji su publikovani u odgovarajućim zbornicima. Sa studentima Visoke tehničke škole u Nišu, 2014. godine je osvojila nagradu za najbolji rad na IEEEESTEC konferenciji studentskih projekata.