

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 02	Arduino – UART serijska komunikacija
Cilj vježbe:	Savladati UART serijsku komunikaciju u Arduino UNO pločici. Upravljanje komponentama naredbama iz Serial monitora. Ostvariti serijsku komunikaciju između dvije Arduino UNO pločice.

Upute za izradu pripreme za vježbu:

Budući da je mikroupravljač računalo na čipu koji se programira kako bi upravljao priključenim vanjskim elektroničkim komponentama, priprema za vježbu se sastoji od dva dijela:

1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u nastavku, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije, objasniti njihovu namjenu i argumente. Ako ne možeš pronaći sve informacije u kodu priloženih zadataka, posluži se internetom npr. www.arduino.cc ...

Upute za izradu vježbe:

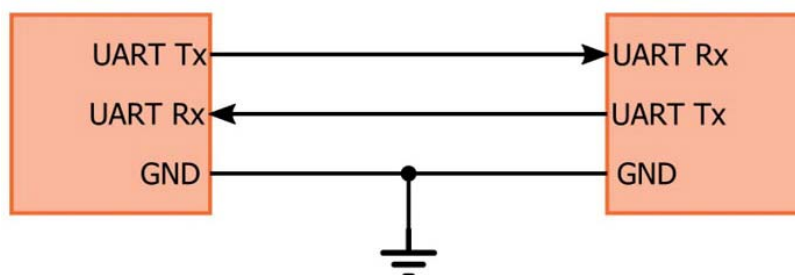
- Svaki zadatak treba prije prevođenja (eng. compile) pohraniti u napravljeni folder na Desktopu, tako da, u slučaju pogreške (HW, SW) imaš sačuvan kod.
- Na kraju LV, sve zadatke spremi na USB ili pošalji na svoj mail.
- Nazivi datoteka, zbog preglednosti, neka budu: LV01_ZAD01, LV01_ZAD02, itd.
- Vježbe se rade u paru, preporuka - jedan učenik spaja komponente, drugi piše programski kod, a na slijedećoj vježbi se uloge zamjenjuju.
- U zadacima koji zahtijevaju samostalno rješavanje, oba učenika sudjeluju u spajanju i programiranju.
- Za pojedini zadatak potrebno je u bilježnicu nacrtati električnu shemu s vidljivim oznakama korištenih pinova i vezu istih s oznakama u programskom kodu.
- Dobiveno rješenje treba komentirati, tj. dati zaključak što je novo u tom zadatku i kako je to riješeno, ukratko ispisati važniji dio koda (ne prepisivati cijeli kod) te navesti eventualne probleme i kako su isti riješeni.
- Ako su uz neki zadatak postoje pitanja, potrebno je u bilježnicu odgovoriti na ista.
- Ako u kodu postoji greška (negdje će biti namjerno stavljena) kod treba korigirati i objasniti!
- Budući da se na vježbama koriste stvarne komponente, postoji mogućnost da je neka neispravna (pregorena LED, oštećen kontakt tipkala, prekinut vodič...). Ukoliko se sklop ponaša drugačije od očekivanog, predvidjeti i tu mogućnost i pokušati zamijeniti komponentu drugom.
- Prilikom spajanja, za Vcc (+5V) koristi crveni vodič, a za GND (-) crni vodič. Za ostale signale koristiti ostale boje.
- Za zadatke koje nisi stigao odraditi na vježbi, treba kod kuće razmisliti kako bi ih riješio
- Po završetku izvođenja vježbe, na temelju odrađene pripreme te riješenih zadataka, očekuje se da učenik zna odgovoriti na pitanja na kraju ovih materijala.
- Pregledavanje priprema i provjeravanje znanja bit će na svakoj LV, uključujući i prethodne vježbe

UART serijska komunikacija

Serijska komunikacija omogućava jednostavan prijenos podataka s jednog uređaja na drugi. Podaci se prenose u vremenskom slijedu, jedan iza drugoga. U jednom trenu prijenosi se samo jedan bit (signal + GND).

Prednost je mali broj vodiča, jeftinija je, lakše se štiti od smetnji (npr. oklapanjem), pa se podaci mogu prenositi na veće udaljenosti. Ukoliko želimo ostvariti dvosmjernu vezu, potrebno je 3 vodiča (Rx, Tx i GND).

Nedostatak je što je sporija u odnosu na paralelnu komunikaciju [1].



Slika 1. Dvosmjerna serijska komunikacija

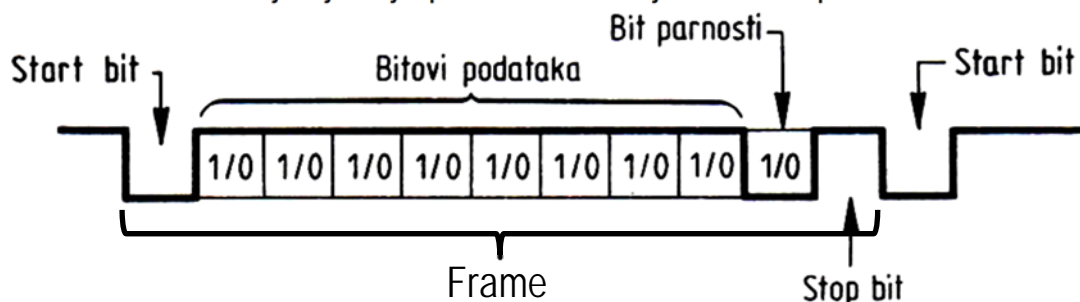
TX (*transmit*) je izlazni pin, a RX (*receive*) ulazni pin. Uređaji se spajaju tako da se TX jednog uređaja spaja na RX drugog uređaja i obratno.

Univerzalni sinkroni/asinkroni prijemnik/predajnik (USART, engl. A universal synchronous/asynchronous receiver/transmitter) je vrsta komunikacijskog uređaja koji omogućava računalu sinkronu i asinkronu komunikaciju uređajima spojenih putem serijske veze. USART radi na način da prima paralelne podatke od procesora i pretvara ih u serijski niz podataka pogodnim za slanje na serijski port (uređaj). Slično, prihvaća serijske podatke sa serijskog uređaja i pretvara ih u paralelne, pogodne za procesor. [2]

UART komunikacija uključuje samo asinkronu komunikaciju što znači da se podaci šalju u paketima potpuno nezavisno jedan od drugog. Svaki je podatak samostalna cjelina, tzv. **Frame** – s vlastitim početkom (i završetkom) i nepoznatim trenutkom slanja paketa.

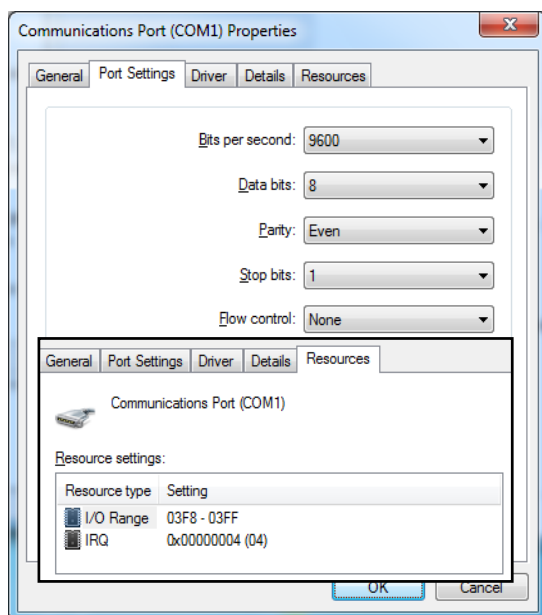
Prijemnik ne zna kad će početi prijenos, nego na Rx pinu 'osluškuje' kad će se pojaviti START bit koji označava početak komunikacije. START bit ima vrijednost '0' budući da kad nema prijenosa, linija je na visokom logičkom nivou čime se prati da je vod aktivan (spojen). Iza Start bita, slijede bitovi podatka i eventualno bit pariteta. Kao posljednji bit šalje se STOP bit čija je vrijednost '1'.

Pomoću ovih bitova uređaj koji šalje podatke određuje trenutak početka i završetka poruke (**frame**).



Slika 2. Izgled jednog paketa serijske UART komunikacije

Da bi se komunikacija mogla ostvariti, oba uređaja – prijemnik i predajnik moraju imati podešene iste parametre:



Slika 3. Podešavanje parametara UART komunikacije u Windows OS

Bits per second: 9600, 14400, 19200, 38400, 57600, 115200,...

Data bits: 4, 5, 6, 7, 8

Parity: Even, Odd, None, Mark, Space

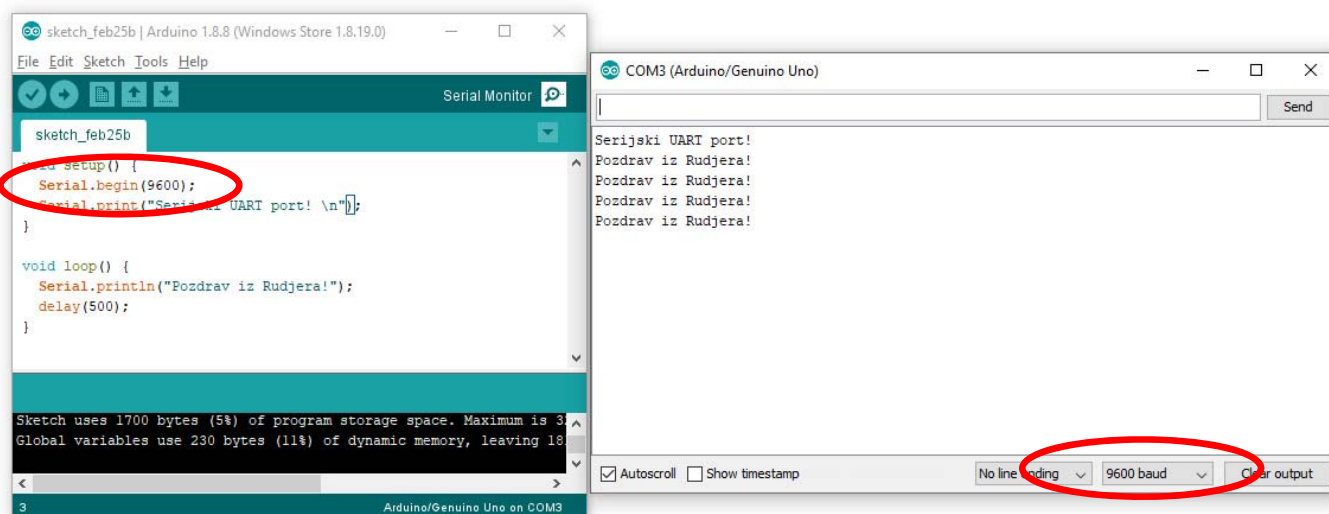
Stop bits: 1, 1.5, 2

Flow control: None, Xon/Xof, Hardware

Za komunikaciju s PC računalom potreban je program za emulaciju Terminala (Terminal Emulator Program) koji govori UART jezikom. Uz Arduino IDE dolazi ugrađeni Serial monitor. Osim njega, moguće je koristiti i druge terminal emulator: TeraTerm, Putty,...

Inicijalizacija serijske komunikacije u Arduinou ostvaruje se naredbom **Serial.begin(9600)** i ona postavlja brzinu prijenosa na 9600 bps. Ako se ne navedu dodatni argumenti, to odgovara **Serial.begin(9600, SERIAL8N1)** što znači 8 bita, No parity, 1 stop bit.

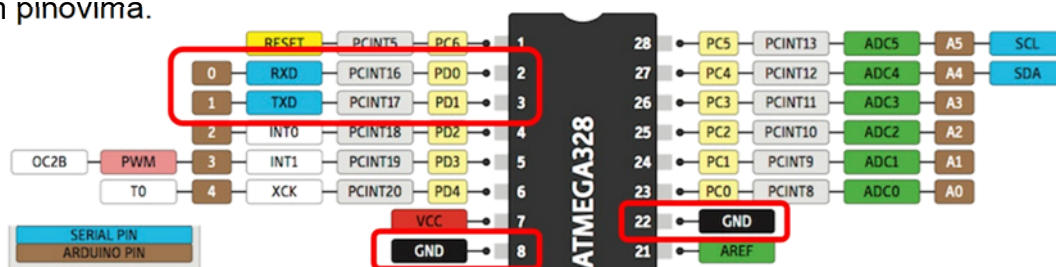
Da bi se ostvarila komunikacija parametri u programu i u Serial monitoru moraju međusobno odgovarati. **Isto vrijedi i za povezivanje dva Arduinoa!**



Slika 4. Usklađivanje parametara komunikacije u Arduino sketch-u i terminalu Serial monitor

Arduino UNO ima samo jedan hardverski USART port koji koristi pinove 0 (Rx) i 1 (Tx). Oni se koriste programiranje Arduinoa, ali i za ispis informacija na Serial monitor. [3]

Za potrebe vježbe koristit ćemo softverski emulator serijskog porta, ostvaren pomoću ugrađene biblioteke **SoftwareSerial.h**. Tako možemo emulirati UART port na bilo kojim preostalim pinovima.



ASCII American Standard Code for Information Interchange [4]

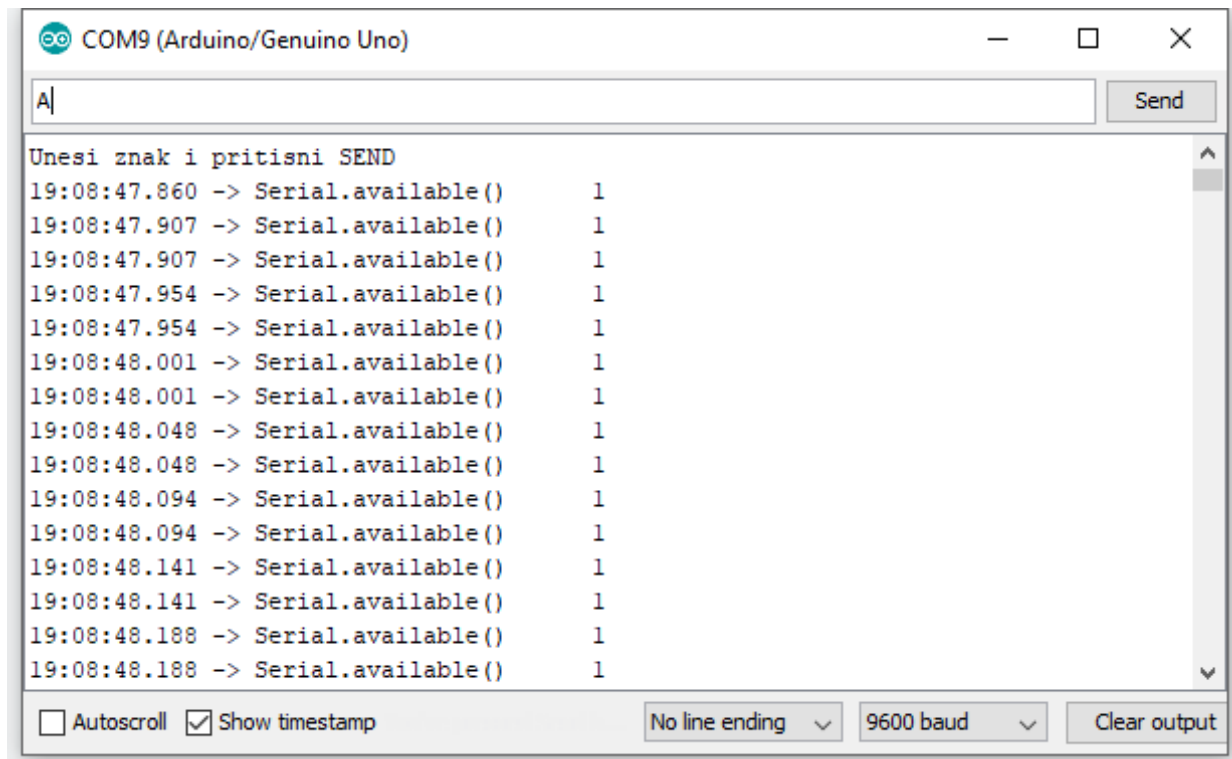
Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

asciichars.com

Prošireni ASCII set

Extended ASCII characters											
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ô
130	82h	é	162	A2h	ó	194	C2h	Ł	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	ł	227	E3h	Õ
132	84h	ä	164	A4h	ñ	196	C4h	Ł	228	E4h	ö
133	85h	à	165	A5h	Ñ	197	C5h	ł	229	E5h	Ö
134	86h	å	166	A6h	ª	198	C6h	Ł	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	ł	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	Ł	232	E8h	ß
137	89h	ë	169	A9h	®	201	C9h	ł	233	E9h	Ú
138	8Ah	è	170	AAh	™	202	CAh	Ł	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	ł	235	EBh	Ü
140	8Ch	ì	172	ACH	¼	204	CCh	Ł	236	ECh	ý
141	8Dh	ï	173	ADh	í	205	CDh	ł	237	EDh	Ý
142	8Eh	Ä	174	A Eh	«	206	CEh	Ł	238	EEh	ˆ
143	8Fh	Å	175	AFh	»	207	CFh	ł	239	EFh	˜
144	90h	É	176	B0h	⋮	208	D0h	Ł	240	F0h	±
145	91h	æ	177	B1h	⋮	209	D1h	ł	241	F1h	±
146	92h	Æ	178	B2h	⋮	210	D2h	Ł	242	F2h	¾
147	93h	ø	179	B3h	⋮	211	D3h	ł	243	F3h	¾
148	94h	ö	180	B4h	⋮	212	D4h	Ł	244	F4h	¶
149	95h	ó	181	B5h	⋮	213	D5h	ł	245	F5h	§
150	96h	ù	182	B6h	⋮	214	D6h	Ł	246	F6h	÷
151	97h	û	183	B7h	⋮	215	D7h	ł	247	F7h	÷
152	98h	ÿ	184	B8h	⋮	216	D8h	Ł	248	F8h	÷
153	99h	Ö	185	B9h	⋮	217	D9h	ł	249	F9h	÷
154	9Ah	Ü	186	BAh	⋮	218	DAh	Ł	250	FAh	÷
155	9Bh	ø	187	BBh	⋮	219	DBh	ł	251	FBh	÷
156	9Ch	£	188	BCh	⋮	220	DCh	Ł	252	FCh	÷
157	9Dh	Ø	189	BDh	⋮	221	DDh	ł	253	FDh	÷
158	9Eh	×	190	BEh	⋮	222	DEh	Ł	254	FEh	÷
159	9Fh	f	191	BFh	⋮	223	DFh	ł	255	FFh	÷

Zadatak 1. Učitaj program na Arduino. Pomoću tipke Send ili Enter na tipkovnici, šalji znakove serijskom vezom prema Arduino. U Serial monitoru pratiti stanje napunjenosti spremnika koju vraća funkcija `Serial.available()`. Provjeri koliki je spremnik, tj. koliko znakova može prihvatiti spremnik! Što se događa kad se spremnik napuni?



Kòd zadatka

```
void setup() {
    Serial.begin(9600);
    Serial.println("Unesi jedan ili više znakova u polje iznad i pritisni
SEND");
}

void loop() {
    // čekam podatke
    if (Serial.available() < 1) return; // ako je serijski spremnik
                                        // prazan vrati se na početak
                                        // petlje loop(). Nakon naredbe
                                        // return, ostali kod se preskače

    Serial.print("Serial.available()");
    Serial.print("\t");
    Serial.print(Serial.available()); // ispiši broj bajtova
                                        // pohranjenih u spremnik
                                        // (engl. buffer); max 63

    Serial.print("\n");
}
```

Zadatak 2. Proširi prethodni zadatak dodavanjem funkcije `Serial.parseInt()` prema priloženom kodu. Preko Serial monitora, šalji različite znakove - slova, brojeve, njihovu kombinaciju, te kombinaciju s razmakom. U Serial monitoru prati koje vrijednosti vraća funkcija `Serial.parseInt()`.

Kòd zadatka

```
void setup() {
    Serial.begin(9600);
    Serial.println("Unesi znak u polje iznad i pritisni SEND");
}

void loop() {
    // čekam podatke
    if (Serial.available() < 1) return; // ako je serijski spremnik
                                        // prazan vrati se na početak
                                        // petlje loop(). Nakon naredbe
                                        // return, ostali kod se preskače

    Serial.print("Serial.available()");
    Serial.print("\t");
    Serial.print(Serial.available()); // ispiši broj bajtova
                                        // pohranjenih u spremnik
    Serial.print("\t");               // (engl. buffer); max 63

    Serial.print("Serial.parseInt()");
    Serial.print("\t");
    int x = Serial.parseInt(); // vraća integer, čita samo brojeve, za
                              // ostale znakove, vraća rezultat 0
                              // VAŽNO: nakon što funkcija pročita znak
                              // briše taj znak iz serijskog spremnika i
                              // oslobađa jedno mjesto (bajt) u spremniku!!

    Serial.print(x);
    Serial.print("\n");
}
```

Zadatak 3. Modificiraj prethodni zadatak dodavanjem funkcije `Serial.read()` prema priloženom kodu. Preko Serial monitora, šalji različite znakove - slova, brojeve, njihovu kombinaciju, te kombinaciju s razmakom (space). Na Serial monitoru prati koje vrijednosti vraća funkcija `Serial.read()`.

Kòd zadatka

```
void setup() {
    Serial.begin(9600);
    Serial.println("Unesi znak u polje iznad i pritisni SEND");
    // Serial.flush();
}

void loop() {
    // čekam podatke
    if (Serial.available() < 1) return; // ako je serijski spremnik
                                        // prazan vrati se na početak
```

```

// petlje loop(). Nakon naredbe
// return, ostali kod se preskače

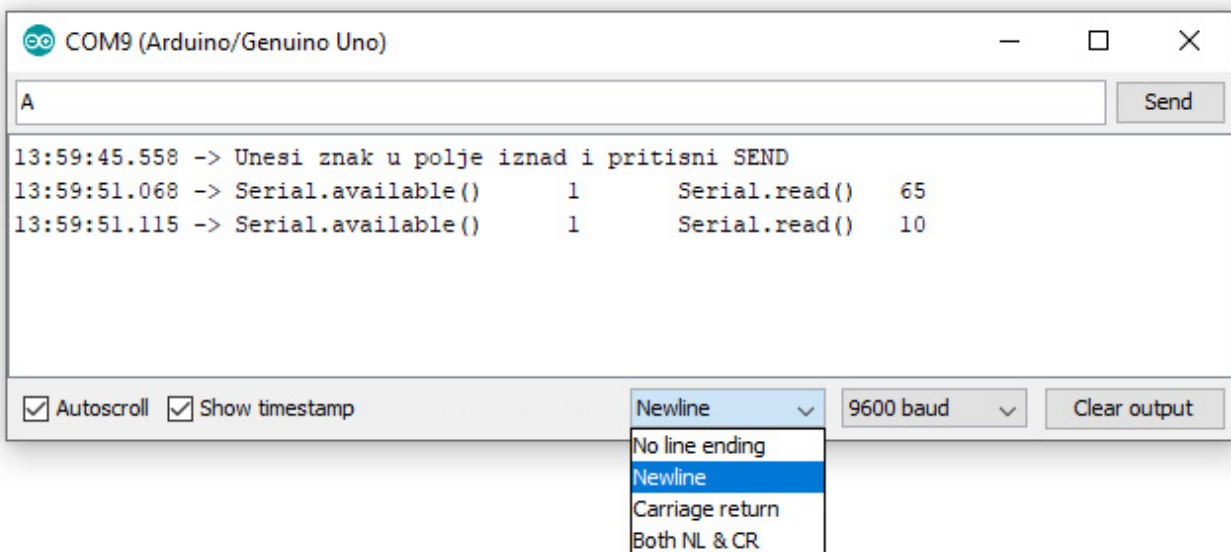
Serial.print("Serial.available()");
Serial.print("\t");
Serial.print(Serial.available());    // ispiši broj bajtova
                                     // pohranjenih u spremnik
Serial.print("\t");                  // (engl. buffer); max 63

Serial.print("Serial.read()");
Serial.print("\t");
int x = Serial.read();               // vraća ASCII kòd bilo kojeg primljenog
                                     // znaka - vrijedi i za brojeve
                                     // VAŽNO: nakon što funkcija pročita znak
                                     // briše taj znak iz serijskog spremnika i
                                     // oslobađa jedno mjesto (bajt) u spremniku!!

Serial.print(x);
Serial.print("\n");
}

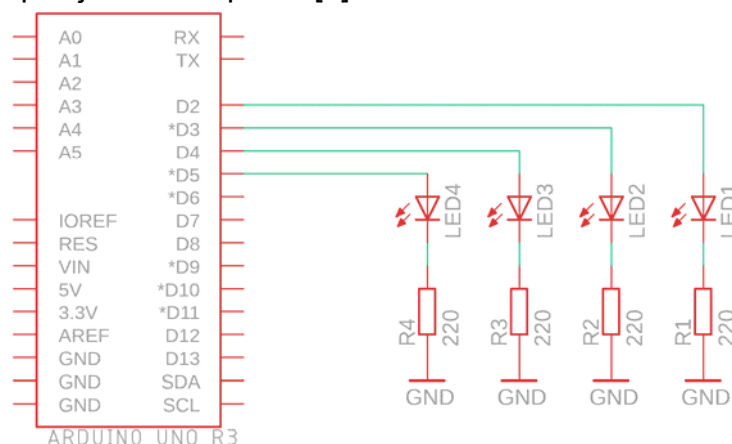
```

Zadatak 4. Ponovi testove iz prethodnog zadatka koristeći Serial monitor i prati što vraća funkcija `Serial.read()` u ovisnosti o vrijednosti podešenoj u padajućem izborniku: „No line ending“, „Newline“, „Carriage return“, „Both NL & CR“. Usporedi primljene ASCII vrijednost s danom ASCII tablicom.



Zadatak 5. Spoji 4 LED diode prema shemi. Koristeći Serial monitor omogućiti uključivanje ili isključivanje LED diode priključene na pin 2. [5]

Električna shema



Kòd zadatka

```
#define PIN_HIGH 1
#define PIN_LOW 0

int pin_num = 2;

void setup() {
    Serial.begin(9600);
    Serial.println("Unesi vrijednost: 0 = LED OFF; 1 = LED ON\n");
    pinMode(pin_num, OUTPUT);
}

void loop() {

    int pin_value = 0;

    if (Serial.available() < 1) return;    // čekam podatke

    pin_value = Serial.parseInt();    // čita vrijednost u spremniku i
                                     // pohranjuje u varijablu
                                     // nakon što pročita, briše tu
                                     // vrijednost iz spremnika

    Serial.print("pin_value\t");
    Serial.print(pin_value);
    Serial.print("\n");

    if (pin_value == PIN_LOW) pin_value = LOW;
    else if (pin_value == PIN_HIGH) pin_value = HIGH;
    else return;    // za bilo koju drugu primljenu vrijednost
                   // vrati se petlji loop()

    digitalWrite(pin_num, pin_value);
}
```

Zadatak 6. Koristeći Serial monitor omogućiti uključivanje ili isključivanje svake LED diode zasebno spojene na pinove 2, 3, 4 i 5. Naredbu iz Serial monitora slati u obliku „2 1“, gdje prva vrijednost predstavlja broj pina, a druga vrijednost (0 = LOW, 1 = HIGH). [5]

Kòd zadatka

```
#define PIN_HIGH 1
#define PIN_LOW 0

// int pin_num = 2;

void setup() {
    Serial.begin(9600);
    // radi bržeg izvođenja zadatka, možeš preskočiti slijedeće tri naredbe
    Serial.print("Unesi vrijednost: 'pin_num pin_value'; pin_num(2-5); ");
    Serial.println("pin_value(0 = LED OFF; 1 = LED ON)");
    Serial.println("Na primjer: '2 1' uključuje LED na pinu broj 2");
    // pinMode(pin_num, OUTPUT);
}
```



```
void loop() {  
  
    int pin_num = 0;  
    int pin_value = 0;  
  
    if (Serial.available() < 1) return;    // čekam podatke  
  
    pin_num = Serial.parseInt();           // čita 1. vrijednost iz spremnika  
                                           // i briše ju  
    pin_value = Serial.parseInt();         // čita 2. vrijednost iz spremnika  
                                           // i briše ju  
  
    Serial.print("pin_num\t");  
    Serial.print(pin_num);  
  
    Serial.print("\tpin_value\t");  
    Serial.print(pin_value);  
    Serial.print("\n");  
  
    if (!(pin_num >= 2 & pin_num <= 5)) return; // provjera pina  
  
    if (pin_value == PIN_LOW) pin_value = LOW;  
    else if (pin_value == PIN_HIGH) pin_value = HIGH;  
    else return;                          // za bilo koju drugu primljenu vrijednost  
                                           // vrati se petlji loop()  
  
    pinMode(pin_num, OUTPUT);  
    digitalWrite(pin_num, pin_value);  
}
```

Zadatak 7. Koristeći Serial monitor, omogućiti promjenu svjetline Led diode na pinovima 3 i 5. Umjesto vrijednosti 0=LOW i 1=HIGH, u kodu prihvaćati vrijednosti 0-255. Naredbu iz Serial monitora slati u obliku „3 255“, „3 127“, „3 1“, ili slično. [5]

Kòd zadatka

```
#define PIN_HIGH 1  
#define PIN_LOW 0  
  
void setup() {  
    Serial.begin(9600);  
    // radi bržeg izvođenja zadatka, možeš preskočiti slijedeće 4 naredbe  
    Serial.print("Unesi vrijednost: 'pin_num pin_value'; pin_num(2-5); ");  
    Serial.println("pin_value(0-255) 0 = LED OFF; 255 = LED MAX");  
    Serial.print("Na primjer: '3 127' uključuje srednju svjetlinu ");  
    Serial.println("LED na pinu 3");  
}  
  
void loop() {  
  
    int pin_num = 0;  
    int pin_value = 0;
```

```

    if (Serial.available() < 1) return;    // čekam podatke

    pin_num = Serial.parseInt();           // čita 1. vrijednost iz spremnika
                                           // i briše ju
    pin_value = Serial.parseInt();         // čita 2. vrijednost iz spremnika
                                           // i briše ju

    Serial.print("pin_num\t");
    Serial.print(pin_num);

    Serial.print("\tpin_value\t");
    Serial.print(pin_value);
    Serial.print("\n");

    if (!(pin_num >= 2 & pin_num <= 5)) return; // provjera primljenog pina

    pin_value = constrain(pin_value, 0, 255); // ograničava primljenu
                                           // vrijednost u zadani
                                           // interval 0 do 255

    pinMode(pin_num, OUTPUT);
    analogWrite(pin_num, pin_value);
}

```

Zadatak 8. Doradi program, tako da ovisno o primljenoj naredbi, možeš LED na pinovima 3 i 5 uključiti naredbom digitalWrite() ili analogWrite(). Umjesto naredbe, koristiti broj 10 za digitalWrite(), a broj 11 za analogWrite(). [5]

Kòd zadatka

```

#define CMD_DIGITALWRITE 10
#define CMD_ANALOGWRITE 11

#define PIN_HIGH 1
#define PIN_LOW 0

void setup() {
    Serial.begin(9600);
    // radi bržeg izvođenja zadatka, možeš preskočiti slijedeće 4 naredbe
    Serial.println("Unesi vrijednost: 'ard_command pin_num pin_value'");
    Serial.println("ard_command(10 ili 11) 10=digitalWrite, 11=analogWrite");
    Serial.println("pin_num(2-5);");
    Serial.println("pin_value(0-255) 1 = LED OFF; 255 = LED MAX");
    Serial.print("Npr: '11 3 63' uključuje manju svjetlinu naredbom analogWrite");
    Serial.println("LED na pinu 3");
    Serial.println("Npr: '10 2 1' uključuje LED na pinu 2 naredbom digitalWrite");
}

void loop() {

    int ard_command = 0;
    int pin_num = 0;
    int pin_value = 0;

    if (Serial.available() < 1) return;    // čekam podatke

```

```

ard_command = Serial.parseInt(); // čita 1. vrijednost (naredba)
pin_num = Serial.parseInt();    // čita 2. vrijednost (PIN)
pin_value = Serial.parseInt();  // čita 3. vrijednost (vrijednost)

Serial.print("ard_command\t");
Serial.print(ard_command);

Serial.print("\tpin_num\t");
Serial.print(pin_num);

Serial.print("\tpin_value\t");
Serial.print(pin_value);
Serial.print("\n");

if (!(ard_command == 10 | ard_command == 11)) return; // provjera
                                                    // naredbe
if (!(pin_num >= 2 & pin_num <= 5)) return; // provjera primljenog
                                                    // pina

if (ard_command == CMD_DIGITALWRITE) {
    Serial.println("CMD_DIGITALWRITE");
    if (pin_value == PIN_LOW) pin_value = LOW;
    else if (pin_value == PIN_HIGH) pin_value = HIGH;
    else return; // za bilo koju drugu primljenu vrijednost
                // vrati se petlji loop()
    pinMode(pin_num, OUTPUT);
    digitalWrite(pin_num, pin_value);
    return;
}

if (ard_command == CMD_ANALOGWRITE) {
    Serial.println("CMD_ANALOGWRITE");
    pin_value = constrain(pin_value, 0, 255); // ograničava primljenu
                                              // vrijednost u zadani
                                              // interval 0 do 255

    pinMode(pin_num, OUTPUT);
    analogWrite(pin_num, pin_value);
    return;
}
}

```

Zadatak 9. Doradi program tako da uz odgovarajuću naredbu, preko Serial monitora, na Serial monitor ispisuješ niz znakova, npr. svoje Ime i Prezime. Umjesto naredbe za tekst koristiti broj 12. Završetak teksta treba označiti znakom „#“. Radi kompatibilnosti s prethodnim zadacima, naredbu slati u obliku „12 0 0 Ime Prezime#“. [5]

```

#define END_CMD_CHAR '#'
#define DIV_CMD_CHAR '|'

#define CMD_DIGITALWRITE 10

```

```
#define CMD_ANALOGWRITE 11
#define CMD_TEXT 12

#define PIN_HIGH 1
#define PIN_LOW 0

String inText;

void setup() {
    Serial.begin(9600);
    // radi bržeg izvođenja zadatka, možeš preskočiti slijedećih 8 naredbi
    Serial.println("Unesi vrijednost: 'ard_command pin_num pin_value'");
    Serial.println("ard_command(10, 11, 12) 10=digitalWrite, 11=analogWrite, 12=tekst");
    Serial.println("pin_num(2-5); ");
    Serial.println("pin_value(0-255) 1 = LED OFF; 255 = LED MAX");
    Serial.print("Npr: '11 3 127' uključuje srednju svjetlinu naredbom analogWrite");
    Serial.println("LED na pinu 3");
    Serial.println("Npr: '10 2 3' uključuje LED na pinu 2 naredbom digitalWrite");
    Serial.println("Npr: '12 0 0 Ime Prezime#' ispisuje Ime Prezime na Serial monitor");
}

void loop() {

    int ard_command = 0;
    int pin_num = 0;
    int pin_value = 0;

    if (Serial.available() < 1) return;    // čekam podatke

    ard_command = Serial.parseInt();    // čita 1. vrijednost (naredba)
    pin_num = Serial.parseInt();        // čita 2. vrijednost (PIN)
    pin_value = Serial.parseInt();      // čita 3. vrijednost (vrijednost)

    Serial.print("ard_command\t");
    Serial.print(ard_command);

    Serial.print("\tpin_num\t");
    Serial.print(pin_num);

    Serial.print("\tpin_value\t");
    Serial.print(pin_value);
    Serial.print("\n");

    // provjera naredbe
    if (!(ard_command == 10 | ard_command == 11 | ard_command == 12)) return;

    //if (!(pin_num >= 2 & pin_num <= 5)) return;    // provjera primljenog
                                                    // pina
    // preseliti unutar if-a za CMD_DIGITALWRITE i CMD_ANALOGWRITE
    // zbog različitih argumenata za CMD_TEXT

    if (ard_command == CMD_DIGITALWRITE) {
        Serial.println("CMD_DIGITALWRITE");
    }
}
```



```
    if (!(pin_num >= 2 & pin_num <= 5)) return; // provjera pina
    if (pin_value == PIN_LOW) pin_value = LOW;
    else if (pin_value == PIN_HIGH) pin_value = HIGH;
    else return; // za bilo koju drugu primljenu vrijednost
                  // vrati se petlji loop()
    pinMode(pin_num, OUTPUT);
    digitalWrite(pin_num, pin_value);

    return;
}

if (ard_command == CMD_ANALOGWRITE) {
    Serial.println("CMD_ANALOGWRITE");
    if (!(pin_num >= 2 & pin_num <= 5)) return; // provjera pina
    pin_value = constrain(pin_value, 0, 255);
    pinMode(pin_num, OUTPUT);
    analogWrite(pin_num, pin_value);
    return;
}

if (ard_command == CMD_TEXT) {
    Serial.println("CMD_TEXT");
    inText = ""; //varijabla u koju se iterativno pohranjuje znak
    while (Serial.available()) {
        char c = Serial.read(); //čita jedan bajt iz spremnika
        delay(5);
        if (c == END_CMD_CHAR) { // kad primim #, završavam daljnje
                                // primanje bajtova
                                // ispišem na LCD ili Serial
            Serial.print("Primio string: ");
            Serial.println(inText);
            break;
        }
        else {
            if (c != DIV_CMD_CHAR) {
                inText += c;
                delay(5);
            }
        }
    }
}

}
```

Zadatak 10. Proširi spoj dodavanjem tipkala na pin 7 Arduina. Pritiskom na tipkalo, potrebno je uključivati i isključivati LED na pinu 2. Koristiti odgovarajuću biblioteku JC_Button-master. Više informacija na: https://github.com/JChristensen/JC_Button. Pritiskom na tipku potrebno je uključiti LED. Ponovnim pritiskom LED se isključuje. Držanje pritisnute tipke ne mijenja stanje LED. [6]

Kòd zadatka

```
// Arduino Button Library
// https://github.com/JChristensen/JC_Button
// Copyright (C) 2018 by Jack Christensen and licensed under
// GNU GPL v3.0, https://www.gnu.org/licenses/gpl.html
//
// Example sketch to turn an LED on and off with a tactile button switch.
// Wire the switch from the Arduino pin to ground.

// Modificirano radi usklađenja s predavanjima URS TŠRB

#include <JC_Button.h>      // https://github.com/JChristensen/JC_Button

// pin assignments
const byte pin_SW1 = 7;    // connect a button switch from this pin to ground
const byte pin_LED = 2;    // the standard Arduino "pin 13" LED

// Button myButton(pin_SW1);    // kreiranje objekta myButton
Button myButton(pin_SW1, 50, TRUE, TRUE); // pin, debounce time, pullup
                                         // enabled, logic inverted

void setup()
{
    myButton.begin();        // initialize the button object
    pinMode(pin_LED, OUTPUT); // set the LED pin as an output
}

void loop()
{
    static bool ledState;    // a variable that keeps the
                             // current LED status
    myButton.read();         // read the button

    if (myButton.wasPressed()) // if the button was pressed,
                             // change the LED state
    {
        ledState = !ledState;
        digitalWrite(pin_LED, ledState);
    }
}
```

LITERATURA:

1. Andrea Bednjanec, Romana Bogut, Sklopovska oprema računala, Zagreb: Element , drugo izdanje
2. Techopedia, Universal Synchronous/Asynchronous Receiver/Transmitter (USART), <https://www.techopedia.com/definition/9850/universal-synchronous-asynchronous-receiver-transmitter-uart>, (pregledano 26. veljače 2019.)
3. Arduino CC, Serial communication, <https://www.arduino.cc/reference/en/language/functions/communication/serial/>, (pregledano 25. veljače 2019.)
4. All About ASCII, ASCII Table - Ascii character codes, <http://www.asciichars.com/>, (pregledano 25. veljače 2019.)
5. Instructables: ArduDroid: a Simple 2-Way Bluetooth-based Android Controller for Arduino, <http://www.instructables.com/id/Andruino-A-Simple-2-Way-Bluetooth-based-Android-C/> , (pregledano 25. veljače 2019.)
6. Arduino Button Library , Arduino library to debounce button switches, detect presses, releases, and long presses, Jack Christensen, https://github.com/JChristensen/JC_Button, (pregledano 25. veljače 2019.)