

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 03	Arduino – UART serijska komunikacija – primjena
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](http://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

**Zadatak 1.** Slijedeći zadatak izvode dvije grupe zajedno. Povezati dva Arduina serijskom UART vezom. Modificirati program na oba Arduina tako da pritiskom na tipku uključuješ i isključuješ LED na svom i susjednom Arduinu.

```
#define PIN_HIGH 1
#define PIN_LOW 0

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

#include <SoftwareSerial.h>
SoftwareSerial Serial_A(10, 11); // RX | TX

const byte pin_SW1 = 7;
const byte pin_LED = 2;

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

void setup()
{
    Serial.begin(9600);           // inicijalizacija HARDVERSKE serijske
                                // komunikacije za ispis na Serial monitor

    Serial_A.begin(9600);        // inicijalizacija SOFTVERSKE serijske
                                // komunikacije za slanje informacije na
                                // drugi Arduino preko pinova 10 i 11

    myButton.begin();            // inicijalizacija objekta myButton
    pinMode(pin_LED, OUTPUT);
}

void loop()
{
    static bool ledState;        // varijabla koja sadrži stanje LED
    myButton.read();            // čitanje stanja tipkala

    // kod zadužen za slanje podataka na SOFTVERSKI serijski port

    if (myButton.wasPressed()) // ako je tipkalo bilo jednom pritisnuto
    {
        ledState = !ledState;
        digitalWrite(pin_LED, ledState);

        Serial_A.print(ledState); // Slanje vrijednosti na SOFTVERSKI
                                // serijski port 10(RX) 11(TX)
    }

    int pin_value = 0;
```

```
// kod zadužen za prihvatanje podataka sa SOFTVERSKOG serijskog porta

if (Serial_A.available() < 1) return; // čekam podatke sa SOFTVERSKOG
// serijskog porta

pin_value = Serial_A.parseInt();
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;

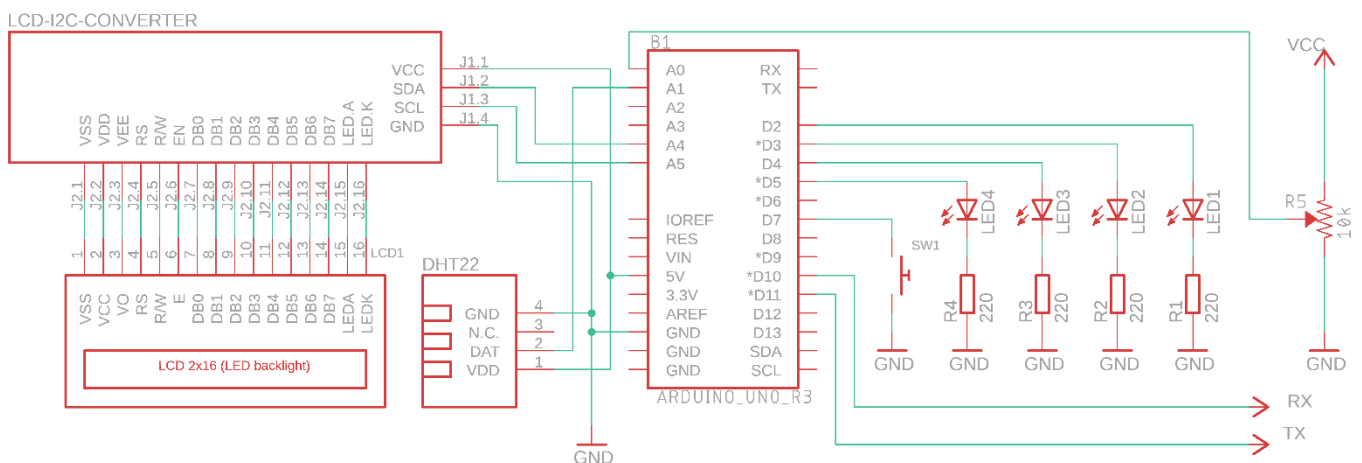
digitalWrite(pin_LED, pin_value);
}
```

**Zadatak 2.** Zadatak također izvode dvije grupe zajedno i obje grupe trebaju imati istu funkcionalnost. Funkcionalnost iz prethodnog zadatka treba ostati.

Obje grupe proširuju spoj dodavanjem po jednog potencijometra i jedne LED diode na jedan od analognih (PWM) izlaza.

- A/D vrijednost s vlastitog potencijometra 0 – 1023 treba ispisati u prvom retku vlastitog LCD zaslona
- A/D vrijednost vlastitog potencijometra slati serijskom vezom na susjedni Arduino. Slati A/D vrijednost serijskom vezom samo kad dođe do promjene A/D vrijednosti.
- Primljenu A/D vrijednost sa susjednog Arduina treba ispisivati u drugom retku LCD-a.
- Vlastitom A/D vrijednošću potencijometra treba podešavati svjetlinu LED diode na susjednom Arduinu. Isto tako omogućiti regulaciju svjetline LED diode primljenom A/D vrijednosti serijskom vezom od susjednog Arduina.

**Zajednička električna shema za obje grupe i za zadatak 12 i zadatak 13.**



**Zadatak 3.** Zadatak također izvode dvije grupe zajedno i obje grupe trebaju imati istu funkcionalnost. Funkcionalnost iz prethodnog zadatka treba ostati.

Proširiti spoj dodavanjem temperaturnog senzora DHT22 koji ima temperaturnu rezoluciju 0,1°C i rezoluciju mjerenja vlage od 0,1%.

- Informacije o temperaturi i vlazi slati na susjedni Arduino koji će temperaturu ispisati u gornjem desnom kutu LCD zaslona, a relativnu vlažnost u donjem desnom kutu.
- Kreirati znakove za stupanj celzijusa i %.

**Zadatak 4.** Zadatak također izvode dvije grupe zajedno i obje grupe trebaju imati istu funkcionalnost. Funkcionalnost iz prethodnog zadatka treba ostati.

Modificirati program koristeći funkciju `Serial.write()` umjesto `Serial.print()`. Sve funkcionalnosti zadataka 12 i 13 trebaju ostati jednake.

Prilagoditi očitane vrijednosti temperature i vlage za slanje serijskom UART vezom prema susjednom Arduinu. Slati A/D vrijednost serijskom vezom samo kad dođe do promjene A/D vrijednosti ili promjene temperature ili relativne vlažnosti.

- a) A/D vrijednost iz potencijometra treba razdvojiti na 2 broja koja stanu u 8 bita (2 bajta), budući da se naredbom `Serial.write()` mogu prenositi samo brojevi 0-255. Na susjednom Arduinu potrebno je koristiti obrnuti algoritam.
- b) Za slanje temperature i relativne vlažnosti, budući da se radi o **float** vrijednostima s dvije decimale, potrebno ih je prvo pretvoriti u cjelobrojnu vrijednost `int`. Na susjednom Arduinu potrebno je koristiti obrnuti algoritam.