

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 02	Arduino – Primjena UART serijske komunikacije
Ishodi vježbe:	Ostvariti serijsku komunikaciju između dvije Arduino UNO pločice. Prenositi informacije i upravljati komponentama spojenim na susjedni Arduino.

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:

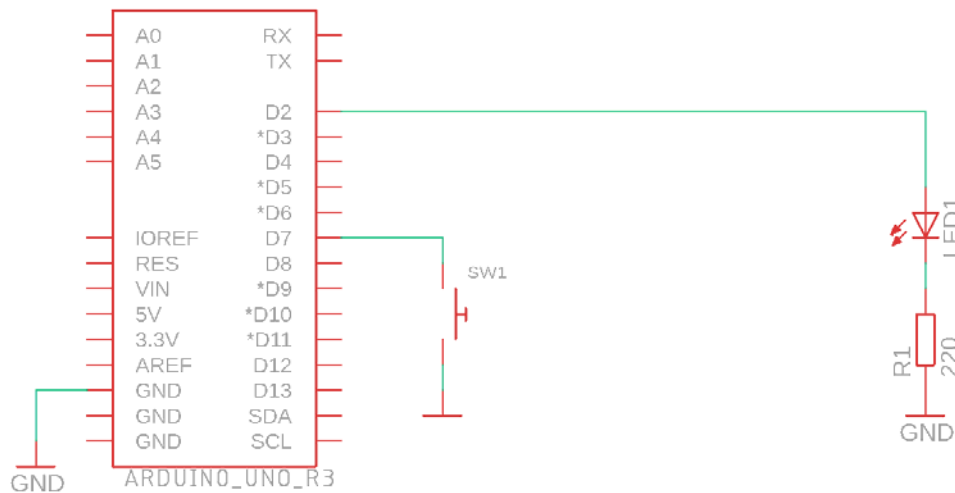
1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u nastavku, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije, 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

Zadatak 1. Spoji komponente prema priloženoj shemi. Pritiskom na tipkalo SW1, potrebno je uključivati i isključivati LED1 na pinu 2. Koristiti biblioteku JC_Button-master koja omogućuje bolju kontrolu tipkala. Više informacija na: https://github.com/JChristensen/JC_Button. Pritiskom na tipku potrebno je uključiti LED1. Ponovnim pritiskom LED1 se isključuje. Držanje tipke pritisnutom ne mijenja stanje LED1.

Električna shema



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

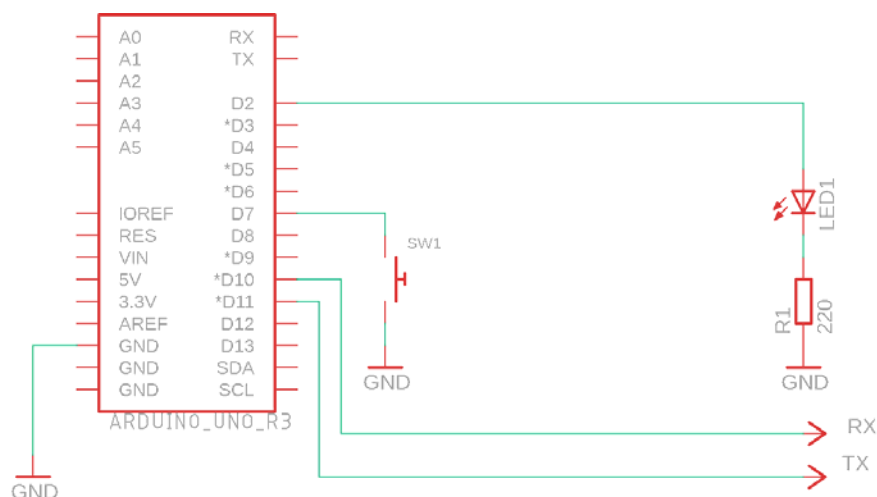
    if (myButton.wasPressed()) // if the button was pressed,
                             // change the LED state
    {
        ledState = !ledState;
        digitalWrite(pin_LED, ledState);
    }
}

```

NAPOMENA: Zadatke u nastavku izvode dvije grupe zajedno. Zadaci se kontinuirano proširuju i dorađuju i ne treba odspajati prethodno spojene komponente. Sve funkcionalnosti ostvarene u prethodnim zadacima trebaju ostati.

Zadatak 2. Potrebno je povezati dva Arduina serijskom UART vezom. Za komunikaciju sa susjednim Arduinoom treba koristiti softverski serijski port koristeći pinove 10 (RX) i 11 (TX). Izlaz TX (*transmit*) s jednog Arduina povezati s ulazom RX (*receive*) drugog Arduina i obratno. Hardverski serijski port na pinovima 0 i 1 ostaviti slobodnim za programiranje Arduina i za ispis na Serial monitor. Modificirati program na oba Arduina tako da pritiskom na tipku uključuješ i isključuješ LED1 na svom i susjednom Arduinou.

Električna shema



Kòd zadatka

```

#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 myButton1(pin_SW1); // kreiranje objekta myButton
Button myButton1(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

    myButton1.begin(); // inicijalizacija objekta myButton
    pinMode(pin_LED, OUTPUT);
}

void loop()
{
    int pin_value = 0; // varijabla koja sadrži stanje LED

    static bool ledState; // varijabla koja sadrži stanje LED
    myButton1.read(); // čitanje stanja tipkala

    // kod zadužen za slanje podataka na SOFTVERSKI serijski port

    if (myButton1.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)
    }

    // 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");
}
```

```

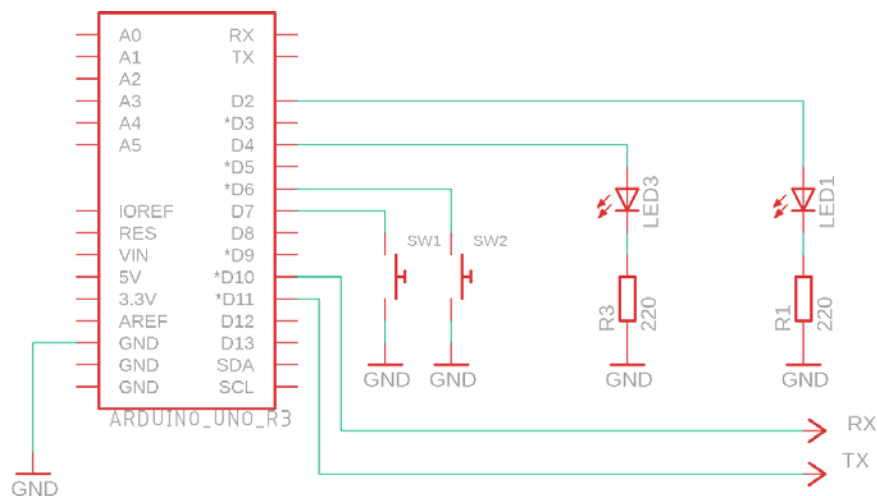
if (pin_value == PIN_LOW) pin_value = LOW;
else if (pin_value == PIN_HIGH) pin_value = HIGH;
else return;

digitalWrite(pin_LED, pin_value);
}

```

Zadatak 3. Doradi spoj dodavanjem drugog tipkala SW2 i diode LED3 prema shemi. Modificiraj program tako da pritiskom na tipkalo SW2 uključuješ i isključuješ LED3 na susjednom Arduino. Tipkalo SW1 i dalje uključuje i isključuje LED1 na vlastitom Arduino. Za rad s drugim tipkalom, potrebno je kreirati dodatni objekt: `Button myButton2(pin_SW2, 50, TRUE, TRUE);`.

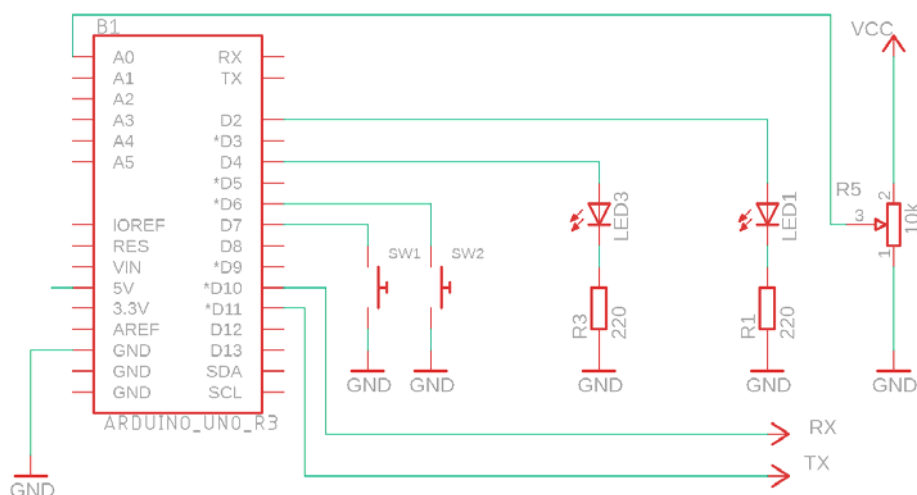
Električna shema



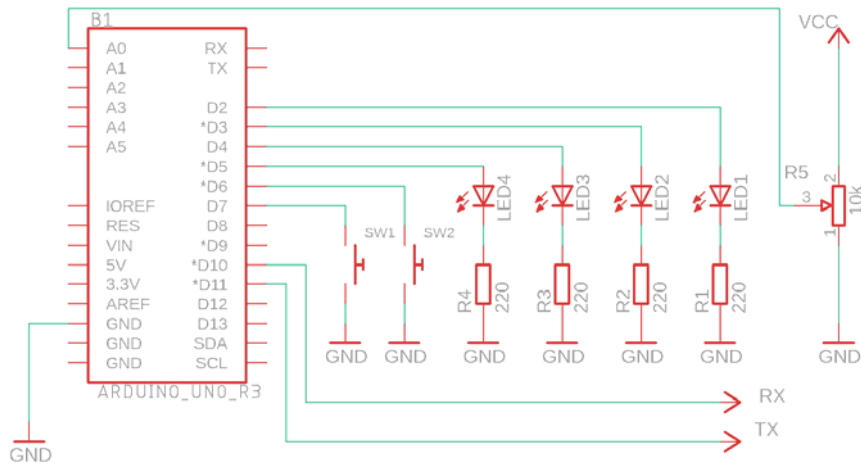
Zadatak 4. Proširi spoj dodavanjem potencijometra prema shemi. A/D vrijednost vlastitog potencijometra treba slati serijskom vezom na susjedni Arduino. A/D vrijednost treba slati samo kad dođe do promjene A/D vrijednosti. Svoju i primljenu A/D vrijednost prikazivati na Serial monitoru samo kod promjene vrijednosti jednog od potencijometara.

Napomena: za detekciju promjene vrijednosti treba dodati varijablu koja će sadržavati prethodno stanje A/D vrijednosti.

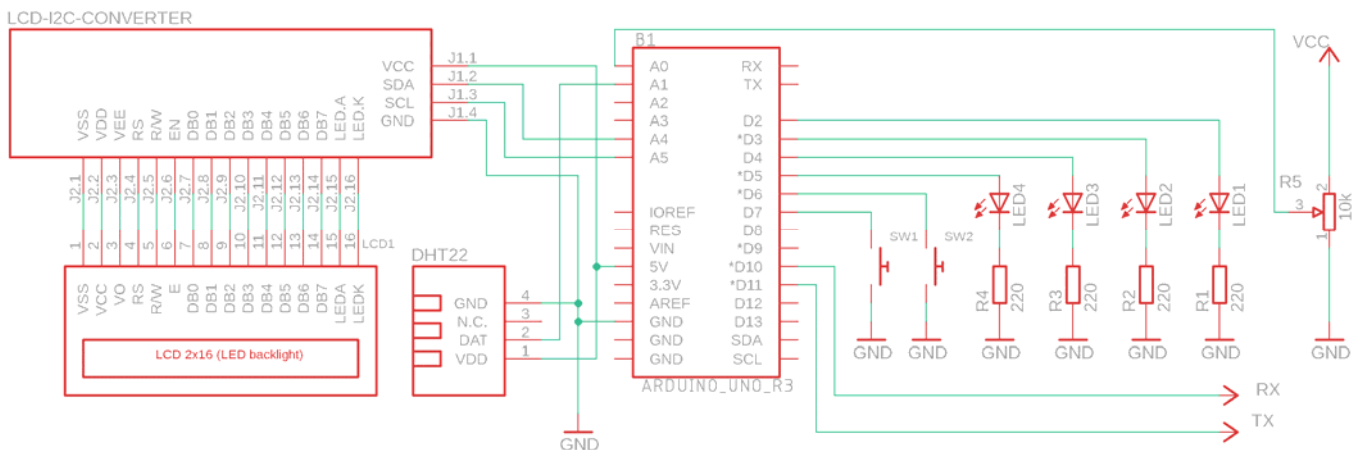
Električna shema



Zadatak 5. Proširi spoj dodavanjem dioda LED2 i LED4 na analogne (PWM) pinove prema električnoj shemi. Svjetlinu LED2 diode treba podešavati vlastitim potencijetrom, a svjetlinu LED4 regulirati A/D vrijednošću primljenu od susjednog potencijetra. Uočiti da su LED1 i LED2 diode upravljane vlastitim Arduinoom, a LED3 i LED4 upravljane susjednim Arduinoom.



Zadatak 6. Doradi spoj dodavanjem LCD-a prema priloženoj shemi. Modificiraj program tako da A/D vrijednost s vlastitog potencijetra ispisuješ u gornjem lijevom kutu LCD zaslona, a primljenu A/D vrijednost u donjem lijevom kutu LCD zaslona.



Zadatak 7. Proširi spoj dodavanjem temperaturnog senzora DHT22 prema shemi iz prethodnog zadatka. DHT22 ima temperaturnu rezoluciju 0,1°C i rezoluciju mjerenja vlage od 0,1%. Način uporabe DHT22 senzora potražiti u laboratorijskoj vježbi iz prethodne godine: „MU-LV12 - Arduino - Mjerenje temperature i vlage.pdf“.

Modificiraj program tako da:

- informacije o temperaturi i vlazi treba slati na susjedni Arduino serijskom vezom. Susjedni Arduino treba ispisati tu temperaturu u gornjem desnom kutu LCD zaslona, a relativnu vlažnost u donjem desnom kutu.
- kreiraš znakove za stupanj celzijusa i % i upotrijebiš ih za korektan ispis podataka u prethodnoj točki a)

Zadatak 8. 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 prethodnih zadataka 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.