

Nastavni predmet:	MIKROUPRAVLJAČI
Vježba: 03	Arduino – Očitavanje s analognog ulaza i upravljanje zvukom
Ishodi vježbe:	Učenik će moći očitati A/D vrijednost s analognog ulaza i pomoću nje upravljati mikroupravljačem, koristiti funkciju map() i translirati jedan opseg veličina u drugi, te upravljati zvučnim signalima pomoću Arduina.

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:

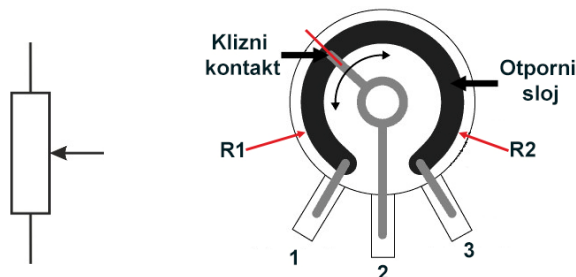
1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u nastavku, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije,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

POTENCIOMETAR

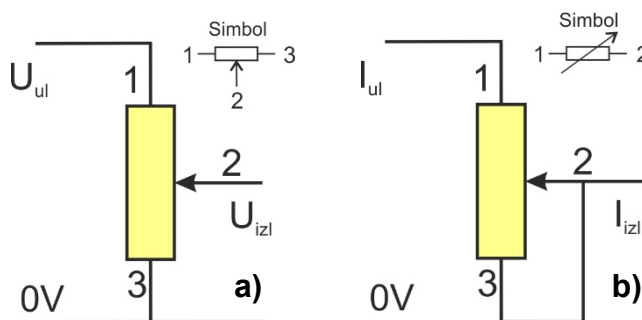
Potenciometar ili promjenjivi otpornik je elektronička komponenta kojoj se pomicanjem klizača ili okretanjem osovine mijenja otpor. Potenciometar ima 3 kontakta. Između dva vanjska kontakta otpor je uvijek isti, dok je srednji izvod spojen na osovinu koja se okreće i ovisno o položaju mijenja otpor. Okretanjem osovine u jednom smeru (udesno) R_1 se povećava, a R_2 smanjuje. Nazivna vrijednost potenciometra je R_p i vrijedi $R_p = R_1 + R_2$. Simbol i dijelovi potenciometra dani su na slici 1.



Slika 1. Simbol i dijelovi potenciometra

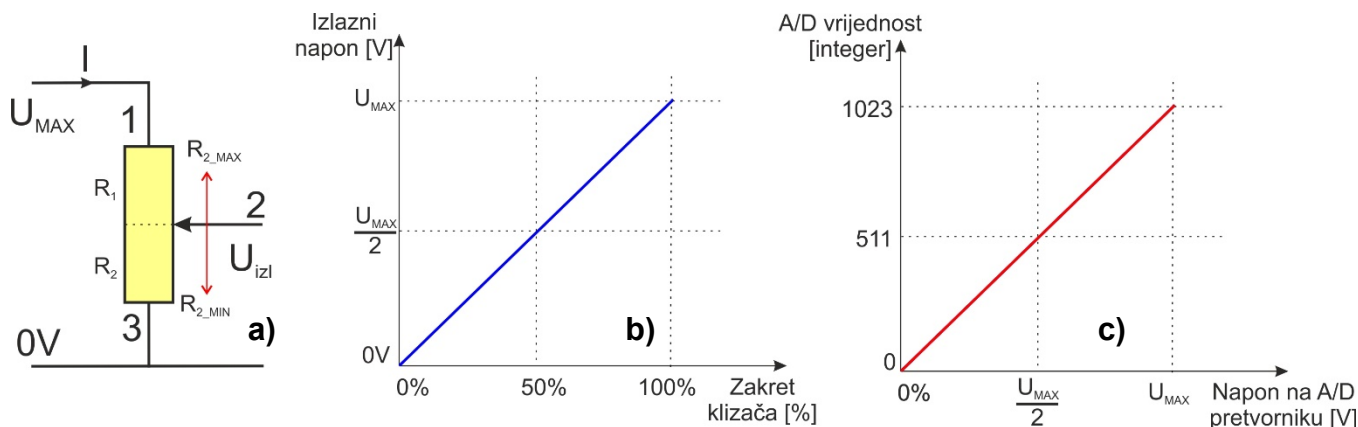
Ovisno o načinu spajanja, potenciometar može raditi na 2 načina: kao dijelilo napona – potenciometarski spoj i kao promjenjivi otpornik.

1. Kad se koristi kao potenciometar, koriste se sva tri priključka (slika 2a). Ovaj spoj se ponaša kao naponsko djelilo koje dijeli napon tako da napon između točaka 2 i 3 direktno ovisi o kutu zakreta osovine.
2. Kad se koristi kao promjenjivi otpornik, koriste se samo 2 priključka – jedan krajnji priključak i klizni priključak (slika 2b). Ovakav spoj se ponaša kao promjenjivi otpornik i služi za podešavanje struje.



Slika 2. Primjena potenciometra kao djelila napona i kao promjenjivog otpornika

Za povezivanje na mikroupravljač i regulaciju različitih veličina, najčešće se primjenjuje potenciometarski spoj (slika 3a).



Slika 3. Ovisnost izlaznog napona potenciometra o kutu zakreta osovine i A/D vrijednosti u ovisnosti o naponu na ulazu A/D pretvornika

Struja kroz potencijometar se ne mijenja budući da je otpor između točaka 1 i 3 uvijek isti. Iz tog razloga, izlazni napon je napon na otporniku R_2 i on se linearno mijenja ovisno o položaju klizača (slika 3b). Kad se klizač pomiče prema gore ili zakreće ulijevo, otpor R_2 se povećava, a time se povećava i izlazni napon U_{izl} . Kad se klizač pomiče prema dolje ili zakreće udesno, R_2 se smanjuje, a time i izlazni napon.

$$U_{izl} = I * R_2 = \frac{U_{max}}{R_1 + R_2} * R_2 = \frac{U_{max}}{R_p} * R_2 = a * R_2$$

Ovaj izraz vrijedi u slučaju da je naponsko djelilo neopterećeno, ili da je izlaz 2 spojen na komponentu velikog ulaznog otpora pa vrlo malo opterećuje naponsko djelilo. Budući da je $U_{max}/(R_1+R_2)=R_p$ =konstantno, cijeli izraz u tom slučaju predstavlja pravac.

Postoji više vrsta potencijometara: okretni, klizni, trimer i višeokretajni (helikoidalni, multi-turn),...



Slika 4. Vrste potencijometara: okretni, klizni, trimer i višeokretajni

A/D PRETVORNIK

A/D pretvornik je elektronička komponenta koja pretvara analogni signal (napon) u niz digitalnih signala koji predstavljaju digitalni podatak (broj). Većina mikroupravljača ima ugrađen A/D pretvornik koji omogućava čitanje ne samo digitalnih 0 ili 1 (naredba `digitalRead(pinX)`), već može pretvoriti analogne napone, npr. vrijednosti između 0 i 5V u brojčanu vrijednost.

U ATmega328p se nalazi **10-bitni A/D pretvornik**, pa nakon A/D pretvorbe, funkcija `analogRead(pinX)` vraća vrijednost od 0 do 1023 (slika 3c). Pri tome je rezolucija $5V/1024 \approx 4,9 \text{ mV}$. Maksimalna vrijednost je 5V budući da je **Aref** spojen na Vcc. Ukoliko želimo drugu referentnu (tj. maksimalnu) vrijednost, potrebno je na pin **Aref** dovesti željeni referentni napon. A/D pretvorba je moguća na pinovima A0 do A5.

Uporaba potencijometra korisna je iz dva razloga:

1. pomoću potencijometra možemo simulirati neku analognu veličinu u obliku napona koju ćemo mjeriti u fizičkom svijetu, a koja nam trenutno nije dostupna, te
2. pomoću potencijometra možemo podešavati i druge parametre kojima upravlja mikroupravljač, npr. svjetlinom LED diode, brojem okretaja motora, kut zakreta motora, određivati kut ili poziciju i sl.

Proces pretvorbe fizičke veličine je slijedeći: kut zakreta osovine potencijometra određuje napon na klizaču pina. Taj se napon dovodi na A/D pretvornik i pretvara u broj 0-1023. Softverski određujemo funkciju koja odgovara zakretu osovine.

Kut zakreta potencijometra (α) → napon na potencijometru (V) → A/D pretvornik (broj 0 do 1023) → varijabla (integer) → if() funkcija za određivanje daljnjeg toka programa

FUNKCIJA MAP()

Funkcija MAP linearno translata neki broj iz jednog opsega vrijednosti npr. iz A/D pretvornika 0-1023 u novi opseg vrijednosti, npr. 0-255, ili 0 – 100%, 100Hz – 1000Hz, 200-1500 ms, ovisno o potrebi.

Sintaksa funkcije: `int new_val = map(value, fromLow, fromHigh, toLow, toHigh)`

Gdje su parametri:

value: broj koji se mapira, tj. ulazna varijabla

fromLow: donja granica sadašnjeg opsega vrijednosti

fromHigh: gornja granica sadašnjeg opsega vrijednosti

toLow: donja granica novog opsega vrijednosti

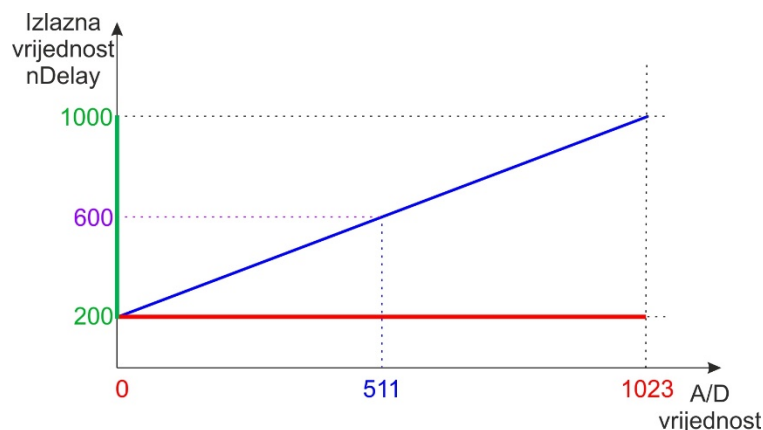
toHigh: gornja granica novog opsega vrijednosti

new_val: nova vrijednost mapiranog broja u novom opsegu

Na slici 7 prikazan je graf funkcije „`int nDelay = map(AD_vrijednost_s_potenciometra, 0, 1023, 200, 1000)`“, gdje opseg od 0 do 1023 preslikavamo u opseg 200 do 1000.

Pri tome, za ulaznu A/D vrijednost 511, što je polovica sadašnjeg opsega, funkcija `map()` će vratiti broj 600, što odgovara polovici novog opsega.

Za ulaznu vrijednost 0, funkcija `map` vraća vrijednost 200, dok za maksimalnu ulaznu vrijednost 1023, funkcija `map()` vraća 1000.



Slika 7. Grafički prikaz funkcije `map()`

U kombinaciji s potenciometrom, na ovaj način možemo podešavati različite veličine: svjetlinu LED diode (0-255), postotak neke izlazne veličine u ovisnosti o zakretu potenciometra (0%-100%), frekvenciju zvuka (100-1000 Hz), podešavati vrijeme za funkciju `delay(200-1000 ms)` i sl.

Primjeri:

- `int nBright = map(AD_vrijednost_s_potenciometra, 0, 1023, 0, 255);`
- `int nPercent = map(AD_vrijednost_s_potenciometra, 0, 1023, 0, 100);`
- `int nFrequency = map(AD_vrijednost_s_potenciometra, 0, 1023, 100, 1000);`
- `int nDelay = map(AD_vrijednost_s_potenciometra, 0, 1023, 200, 1000);`

Objašnjenje funkcije MAP = jednadžba pravca kroz dvije točke

```
long map(long x, long in_min, long in_max, long out_min, long out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

Više informacija: <https://www.arduino.cc/reference/en/language/functions/math/map/>

UPRAVLJANJE ZVUKOM

Piezo zujalica je izlazni uređaj kojem je uloga davanje **jednostavnih zvučnih signala**. Služe kako bi mikroupravljač (općenito neki uređaj) **informirao korisnika o određenoj akciji**, primjerice da je **registrirao neki događaj** ili informirao o prekoračenju određene kritične vrijednosti u obliku **alarma**.

Radi na principu obrnutog **piezoelektričnog efekta**, na način da ako na piezoelektrični kristal dovedemo promjenjivi napon, doći će do mehaničkih promjena dimenzija kristala koji će stvarati zvuk u skladu s frekvencijom dovedenog napona.

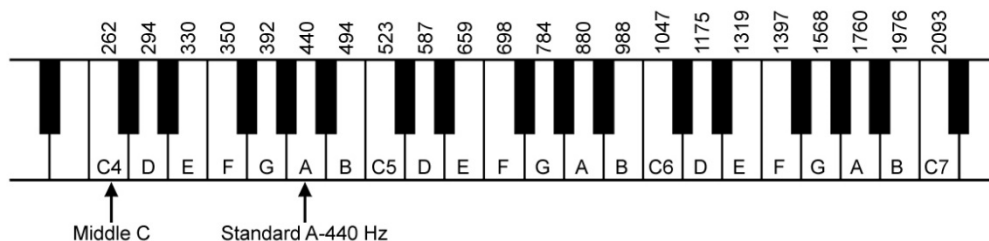
Budući da se obično na zujalicu spaja izlazni pin Arduinoa, gdje se pojavljuje **digitalni signal**, dakle 0V ili 5V (3,3V), **ne treba očekivati spektakularni zvuk**.

Za slanje zvuka na piezo zujalicu, koristi se funkcija `tone()`, čija je sintaksa `tone(pin, frekvencija, trajanje tona)`. Druga mogućnost je funkcija `tone(pin, frekvencija)`. U tom slučaju, da bi isključili zvuk, treba pozvati funkciju `noTone(pin)`.

Funkcija `tone()` koristi jedan od **ugrađenih tajmera** unutar mikroupravljača, tako da **može raditi nezavisno od funkcije delay()**. Moguće je pokrenuti funkciju `tone()` i nakon toga raditi druge operacije, dok ton svira u pozadini.

```
tone(piezo, 175, 1000);  
delay(3000);
```

Umjesto da se vremena zbroje, 1000 ms zvuka + 3000 ms kašnjenje, obje funkcije počinju približno u isto vrijeme (funkcija `delay()` se izvršava odmah iza `tone()`), tako da će rezultat biti umjesto 4000 ms, dobit ćemo 1000 ms zvuka i 2000 ms pauze, tj. ukupno trajanje 3000 ms.

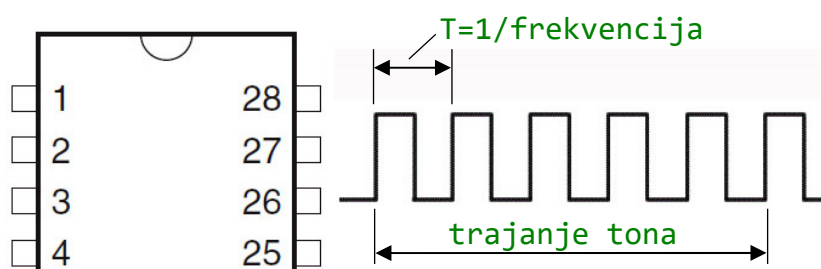


Slika 5. Frekvencije pojedinih tonova

Ograničenja funkcije tone:

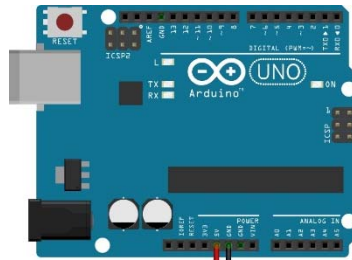
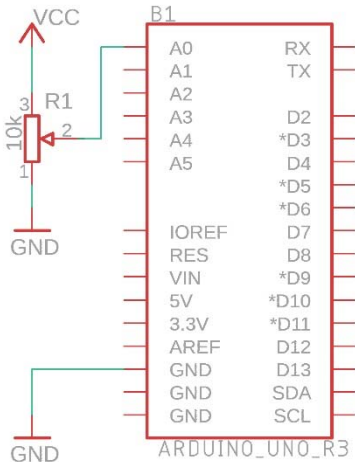
1. Nije moguće koristiti funkciju `tone()` istovremeno s funkcijom `analogWrite()` na pinovima 3 ili 11, jer koriste isti tajmer unutar mikroupravljača. Ako se koristi, rezultat neće biti ispravan i niti jedna funkcija neće raditi ispravno.
2. Nije moguće stvarati ton frekvencija niže od 31 Hz.
3. Funkciju `tone()` nije moguće koristiti istovremeno na dva pina. Ako imamo dvije piezo zujalice spojene na dva različita pina, ne mogu istovremeno proizvesti dva tona. Prvo može svirati jedan, a onda drugi. Prije nego je moguće proizvesti ton na drugoj zujalici, potrebno je funkcijom `noTone()` isključiti prvu zujalicu.

Funkcija `tone`: `tone(pin, frekvencija, trajanje tona)`

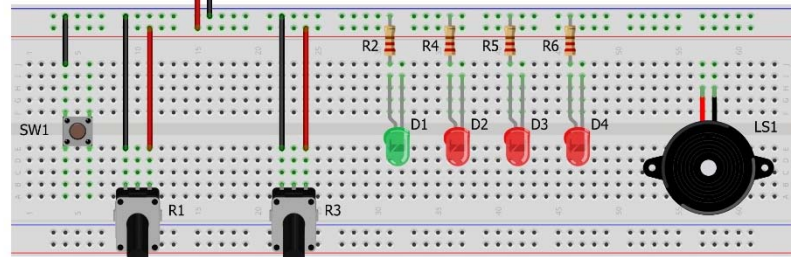


Zadatak 1. Spoji potencijometar R1 prema shemi i na Serial monitoru prikazuj očitane A/D vrijednosti.

Električna shema:



- prijedlog za raspored komponenta do kraja vježbe
- komponente dodavati postupno
- (dopuniti vodičima koji nedostaju):



fritzing

Kod zadatka

```
int pin_pot = A0;
int AD_value = 0;

void setup() {
    pinMode(pin_pot, INPUT);
    Serial.begin(9600);
}

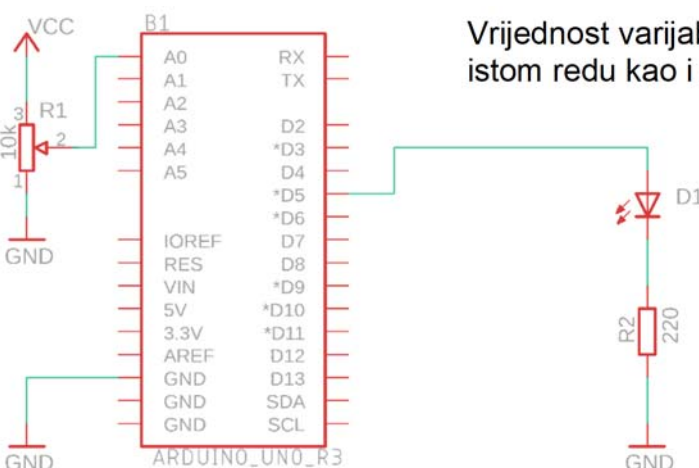
void loop() {
    AD_value = analogRead(pin_pot); //analognu vrijednost potencijometra
                                   // zapisujemo u varijablu AD_value

    Serial.print("A/D vrijednost: "); // ispis stringa na Serial monitor
    Serial.print("\t");               // Ispis razmaka (Tab)
    Serial.print(AD_value);           // ispis varijable na Serial monitor
    Serial.println("\n");             // Prijelaz u novi red
}
```

Zadatak 2. Proširi spoj iz prethodnog zadatka dodavanjem LED1 prema shemi. Modificiraj program tako da pomoću potencijometra reguliraš svjetlinu LED1.

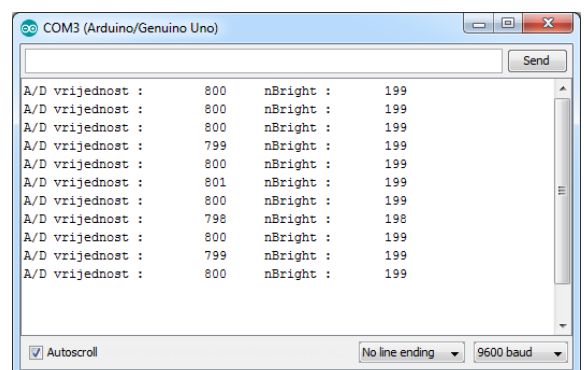
Upute: u funkciju loop(), potrebno je dodati novu varijablu **nBright** i pomoću funkcije „int nBright = map(AD_value, 0, 1023, 0, 255);“ translirati A/D vrijednost iz A/D pretvornika čiji je opseg 0-1023 u novi opseg 0-255.

Električna shema:



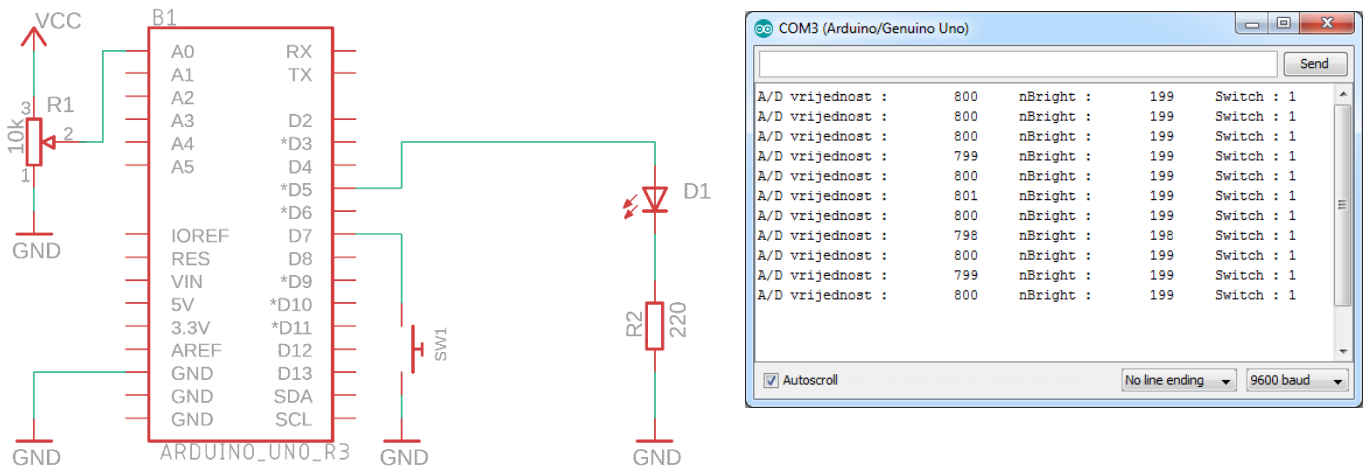
Varijablu nBright koristiti za regulaciju svjetline LED pomoću poznate funkcije iz LV01: „analogWrite(pin_LED, nBright)“.

Vrijednost varijable nBright također ispisuj na Serial monitor, u istom redu kao i A/D vrijednost (razmaknute pomoću „\t“).



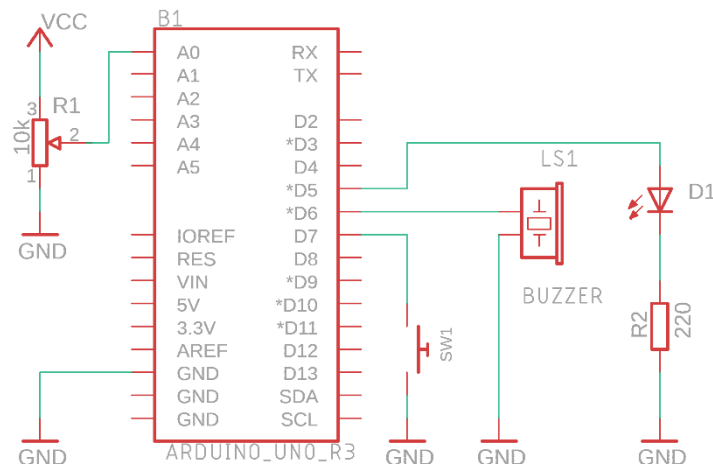
Zadatak 3. Proširi spoj iz prethodnog zadatka tako da dodaš tipkalo prema priloženoj shemi. Modificiraj program tako da LED svijetli samo kad je pritisnuto tipkalo. Svjetlina LED se i dalje podešava pomoću potencijometra. Stanje tipkala također ispisuj na Serial monitor u istom retku.

Električna shema:



Zadatak 4. Proširi spoj iz prethodnog zadatka tako da dodaš zujalicu (*buzzer*) prema priloženoj shemi. Doradi program tako da se tipkalom osim LED uključuje i zujalica. Frekvencija emitiranog tona treba biti 175 Hz. Svjetlina LED se i dalje podešava pomoću potencijometra.

Električna shema:



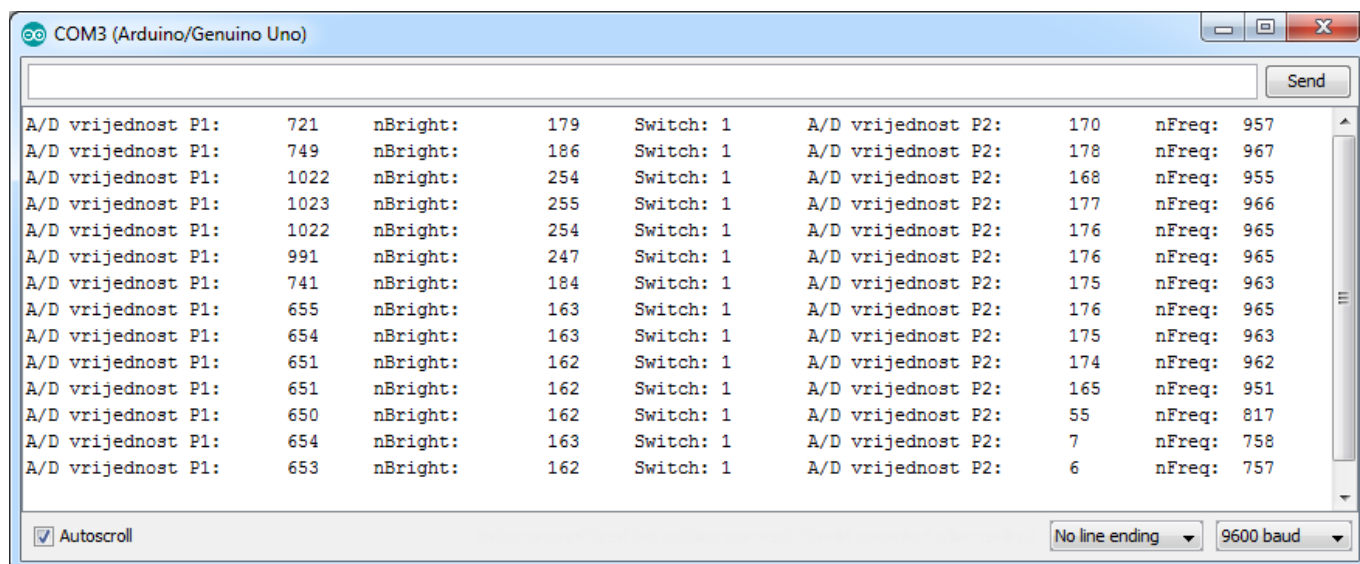
Djelomični kod zadatka

```
int pin_piezo = 6;
int pin_SW1 = 7;

void setup() {
  pinMode(pin_piezo, OUTPUT);
  pinMode(pin_SW1, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(pin_SW1) == LOW) { // tipkalo pritisnuto
    tone(pin_piezo, 175, 100);      // tone(pin,frekvencija,trajanje)
  }
}
```

Zadatak 5. Proširi spoj iz prethodnog zadatka dodavanjem još potencijometra R3 na analogni ulaz A1 i dodatnom diodom LED2. Doradi program iz prethodnog zadatka tako da novim potencijometrom R2, mijenjaš frekvenciju zujalice u rasponu od 750 Hz do 2000 Hz. Kad je podešena frekvencija veća od 1500 Hz, potrebno je uključiti LED2. Tipkalom sada uključuješ samo zujalicu. LED1 više ne upravljaš tipkalom - svijetli cijelo vrijeme. Svjetlinu LED1 i dalje podešavaš potencijometrom na ulazu A0. Sve veličine prikazivati na Serial monitoru kao na slici.



Zadatak 6. Potrebno je uočiti koliko ima pinova i koji sve pinovi imaju mogućnost analognog izlaza! **Analogni pinovi imaju oznaku „~“ uz broj pina.**

Na sve pinove s analognim izlazom osim pinova 3 i 11 spoji LED diode. Napiši program pomoću kojeg ćeš ostvariti trčeće svjetlo. Pomoću potencijometra priključenog na pin A0 reguliraj brzinu 'trčanja' svjetla u intervalu 100 ms do 1000 ms. Pomoću potencijometra priključenog na A1 treba regulirati svjetlinu svih LED dioda koje koristiš, od potpuno ugašene LED diode do maksimalnog osvjetljenja. Pritiskom na tipkalo, trčeće svjetlo ide u suprotnom smjeru. Buzzer daje kratki zvučni signal prilikom svakog uključivanja prve LED diode.

Zadatak 7. Modificiraj program tako da potencijometrom na ulazu A0 simuliraš analogni ulaz u intervalu od 0V do 5V. Koristeći LED diode iz prethodnog zadatka kreirati VU metar. VU metar je instrument koji uključuje određeni broj LED dioda u ovisnosti u ulaznom naponu.

- Za ulazni napon od 0,0 – 1,0V ne svijetli niti jedna LED
- Za ulazni napon od 1,0 – 2,0V svijetli LED1
- Za ulazni napon od 2,0 – 3,0V svijetli LED1 i LED2
- Za ulazni napon od 3,0 – 4,0V svijetle LED1, LED2 i LED3
- Za ulazni napon od 4,0 – 5,0V svijetle LED1 – LED4

Potencijometrom na analognom ulazu A1 i dalje i dalje treba regulirati svjetlinu svih LED dioda koje koristiš, od potpuno ugašene LED diode do maksimalnog osvjetljenja.

Pritiskom na tipkalo uključuješ sve LED, neovisno o položaju potencijometra.

A/D vrijednost, stanje tipkala i izlazne logičke nivoe napone pojedinih dioda prikazati pomoću Serial monitora.

Zadaci za vježbu: razmisli kako bi riješio slijedeće zadatke.

Zadatak 8. Koristeći spoj iz prethodnog zadatka, ali koristeći samo jedan potencijometar, doradi program tako da potencijometrom simuliraš temperaturu u granicama 0°C do 60°C.

Temperaturu ispisuj na Serial monitoru svake sekunde. Prilikom mjerenja, uređaj treba svake sekunde oglašiti zvučni signal trajanja 20 ms frekvencije 1 kHz.

Kad temperatura prijeđe granicu iznad 40°C, uključuje se alarm i crvena LED dioda. U tom slučaju uređaj treba oglasiti zvučni alarm trajanja 500 ms, frekvencije 3 kHz. Na Serial monitor ispisati poruku „Opasnost!“ Zvučni alarm isključiti pritiskom na tipku SW1. Na Serial monitor se i dalje ispisuje poruka „Opasnost!“ sve dok temperatura ne padne u dozvoljeni interval.

Zadatak 9. Spoji piezo zujalicu i dva tipkala na Arduino te napiši program kojim će se pritiskom na prvo tipkalo odsvirati glazbena ljestvica. Pronađi na internetu tablicu svih tonova i njihove frekvencije ili koristi navedene frekvencije u uvodnom dijelu pripreme (slika 5). Pritiskom na drugo tipkalo, Arduino treba odsvirati note tvoje omiljene pjesme. Note također pronadi na internetu. Omogući da se pomoću potenciometra podešava brzina izvođenja melodije.

Izvadak iz koda zadatka

```
int Pin = 9;
int brojTonova = 10;
int tonovi[] = { 262, 277, 294, 311, 330, 350, 370, 392, 415, 440 };
               // mid C  C#  D   D#  E   F   F#  G   G#  A

void setup() {
    pinMode(Pin, OUTPUT);
}
void loop() {
    for (int i = 0; i < brojTonova; i++) {
        tone(9, tonovi[i]);
        delay(500);
    }
    noTone(9);
}
```

Pitanja za provjeru znanja:

1. Što je potencijometar?
2. U kojim spojevima se može koristiti potencijometar?
3. Objasni potencijometarski spoj kad potencijometar koristimo kao naponsko dijelilo.
4. Zašto je izlazni napon linearan i koji uvjet treba biti zadovoljen za to? Skiciraj graf ovisnosti napona na klizaču u odnosu na kut zakreta potencijometra.
5. Nabroji nekoliko vrsta potencijometara!
6. Što je A/D pretvornik?
7. Objasni naredbu `analogRead()` i njene argumente. Koju vrijednost vraća?
8. Objasni razliku između naredbi `digitalRead()` i `analogRead()`!
9. Što određuje rezoluciju A/D pretvorbe?
10. Kolika je rezolucija 10 bitnog A/D pretvornika ako mu je referentni napon 5V?
11. Na koja dva načina možemo koristiti potencijometar s mikroupravljačem?
12. Koji su međukoraci od zakreta potencijometra do informacije u varijabli?
13. Čemu služi funkcija `map()` i koji su njeni argumenti?
14. Objasni funkciju `map()` preko grafičkog prikaza!
15. Koje veličine najčešće podešavamo pomoću potencijometra i funkcije `map()`?
16. Što je piezo zujalica i na kojem principu radi?
17. Ukratko objasni piezoelektrični efekt na kojem radi zujalica
18. Objasni funkciju `tone()` i moguće argumente!
19. Objasni funkciju `noTone(pin)` i kada se koristi!
20. Može li funkcija `tone()` raditi nezavisno od funkcije `delay()` i zašto?

LITERATURA:

1. Electronics Tutorials, Potentiometers, <https://www.electronics-tutorials.ws/resistor/potentiometer.html>, (pregledano 11. listopada 2019.)
2. Arduino web stranica, <https://www.arduino.cc/reference/en/language/functions/math/map/> (pregledano 11. listopada 2019.)