

Predmet:	MIKROUPRAVLJAČI
Vježba: 02	Arduino – Digitalni ulazi i izlazi – tipkala i LED
Ishodi vježbe:	Proširiti poznavanje mogućnosti mikroupravljača, upravljati mikroupravljačem putem tipkala, razlikovati načine povezivanja tipkala na mikroupravljač, ispisivati stanje unutarnjih varijabli mikroupravljača na računalo

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:

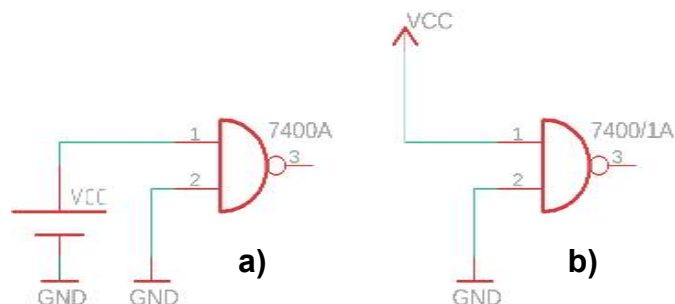
1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u nastavku, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije,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

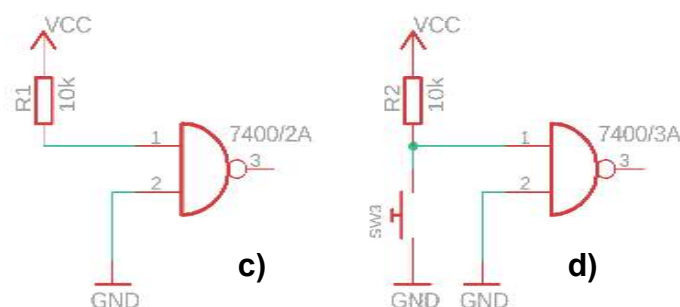
POVEZIVANJE MIKROUPRAVLJAČA S FIZIČKIM SVIJETOM

Jedan od osnovnih načina prenošenja informacije iz fizičkog svijeta u mikroupravljač je pomoću tipkala ili sklopke. Poznato je da na ulaz digitalnog sklopa (npr. TTL serija ili mikroupravljač ATmega328p u Arduino UNO) možemo dovesti logičku 1 spajanjem pina na +5V, a logičku 0 spajanjem na GND. Slika 1 a) i b). **NAPOMENA: Na pinu (nožici) na koji nije spojeno ništa, kaže se da je „u zraku“, nije dovedena logička 0!**



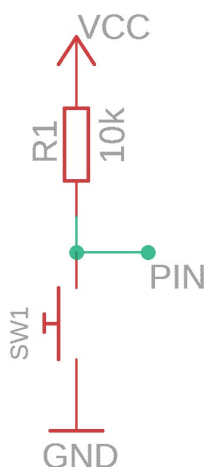
Slika 1a i 1b. Dovođenje logičke 1 i logičke 0 na nožice digitalnog sklopa

Umjesto spajanja +5V ili GND – po potrebi, jednostavnije je izvesti sklop pomoću tzv. **Pull-Up** otpornika i tipkala (*engl. pushbutton*). Slika 2 a) i b).

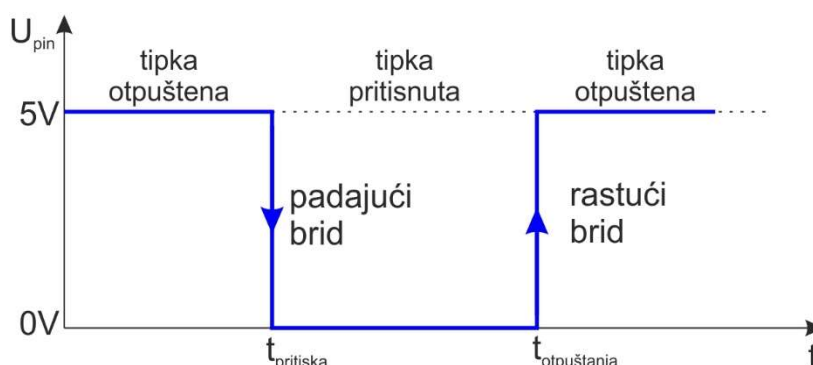


Slika 2a i 2b. Dovođenje logičke 1 i logičke 0 na nožice digitalnog sklopa pomoću tipkala

Kada tipka nije pritisnuta, preko otpornika na PIN dovodimo 5V (slika 3). Kad pritisnemo tipku, na pin dovodimo 0V jer je nožica spojena preko tipkala na GND. Otpornik ujedno ograničava struju koja teče dok je tipkalo pritisnuto. U trenutku **pritiska na tipkalo**, napon na nožici pada s 5V na 0V i pojavljuje se **padajući brid**. Kod **otpuštanja tipkala** napon na nožici se mijenja s 0V na 5V i javlja se **rastući brid**. Budući kad je tipkalo pritisnuto, na izlazu spoja imamo 0V, za ovu logiku kažemo da je aktivna u 0 (slika 4).



Slika 3. Spoj s Pull-Up otpornikom



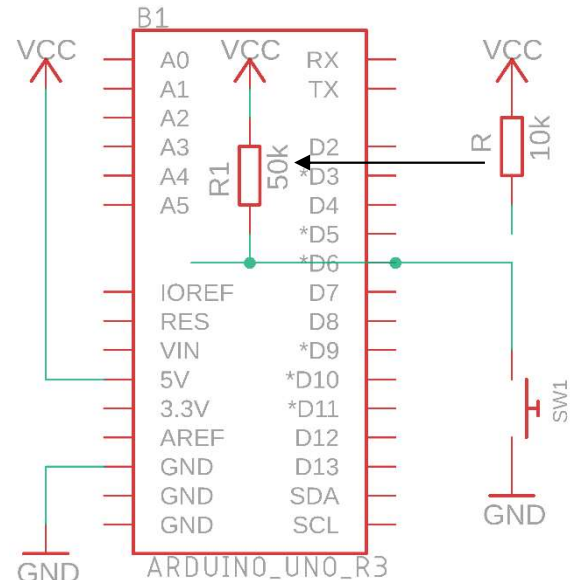
Slika 4. Napon na nožici (pinu) mikroupravljača uz korištenje Pull-Up otpornika i tipkala

Kako bi se pojednostavnilo priključivanje sklopki i tipkala, svaki pin unutar mikroupravljača ATmega328p ima **interni Pull-Up otpornik**, koji se **može softverski uključiti ili isključiti**.

Jedan od načina je pomoću naredbi:

1. **pinMode(pinX, INPUT)** definira pinX kao ulazni, bez Pull-Up otpornika
2. **pinMode(pinX, INPUT_PULLUP)** definira pinX kao ulazni s uključenim internim Pull-Up otpornikom.

Pull-Up otpornik je potreban za priključivanje tipkala i sklopki. Za senzore koji na svojim izlazima daju i 0V i 5V, nije potrebno uključivati Pull-Up otpornik.

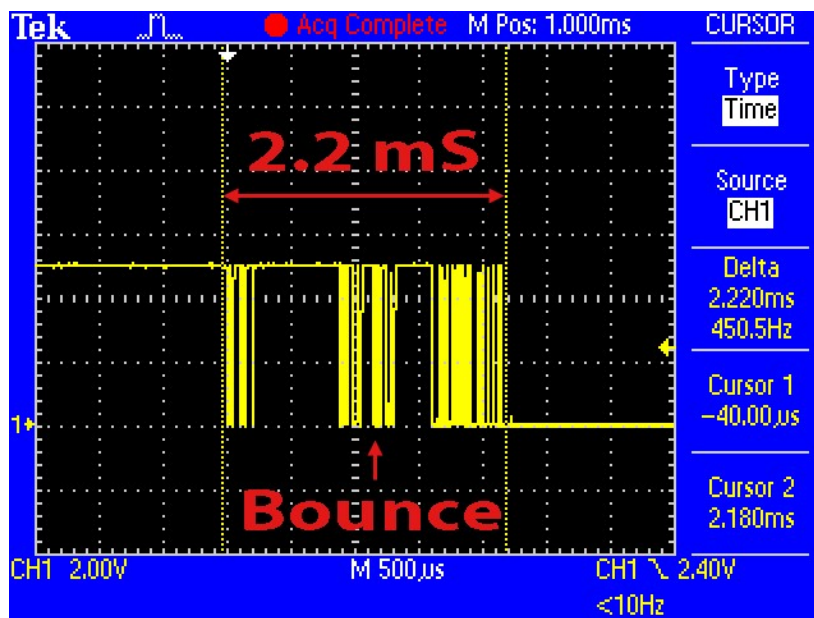


Slika 5. Priključenje tipkala s unutarnjim Pull-Up otpornikom

TITRANJE SKLOPKE (engl. *Switch Bouncing*)

Valni oblik na slici 4. je idealiziran i nije prikazano titranje napona na tipkalu (engl. *Bouncing*) koje se javlja u stvarnom svijetu. Ono se odvija puno brže od ljudske percepcije pa nije odmah uočljivo, a može stvarati probleme u elektroničkim projektima.

Većina tipkala su mehanička i imaju opruge. Prije nego se dva kontakta tipkala pozicioniraju na konačnu poziciju, oni se mehanički dodirnu i razdvoje nekoliko puta stvarajući time privremeni kontakt pa se dobije efekt kao da smo sklopku stisnuli i otpustili nekoliko puta.

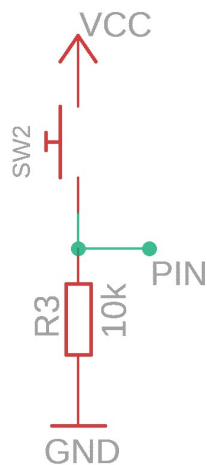
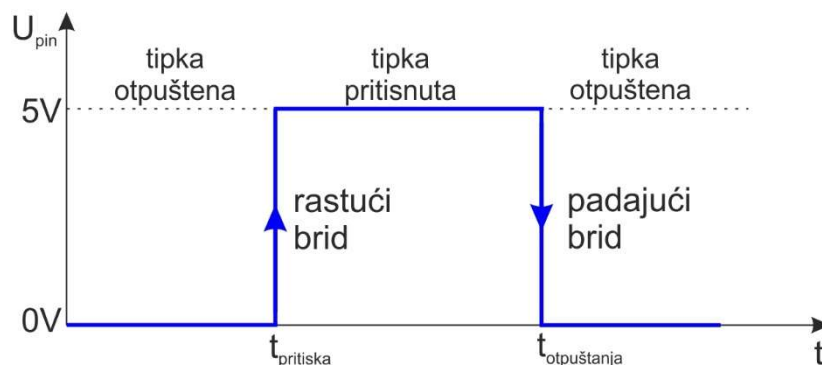


Slika 6. Valni oblik napona na tipkalu (*Switch Bouncing*) [1]

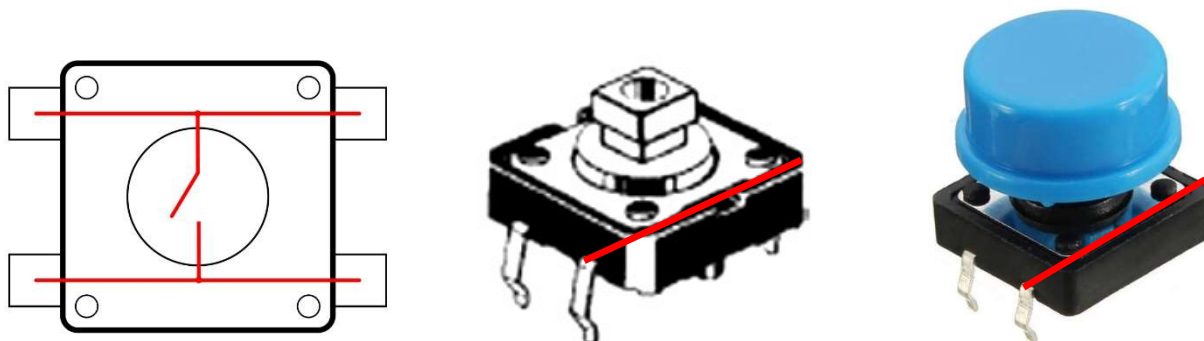
Kao što se vidi iz slike, trajanje titranja najčešće je kraće od 5 ms. Stoga se prilikom dizajna ugradbenih računalnih sustava treba napraviti tzv. DEBOUNCING koji se može izvesti hardverski ili softverski.

DEBOUNCING nije predmet ove laboratorijske vježbe i bit će obrađen kasnije.

Druga varijanta priključenja tipkala je uporabom **Pull-Down** otpornika (slika 7). Kad je tipkalo pritisnuto, na središnjoj točki između tipkala i otpornika vlada napon 5V prema GND. Za ovu logiku kažemo da je aktivna u 1. Ova logika se ne koristi u Arduino UNO pločici, tj. u ATmega328p mikroupravljaču. Slika 8.

Slika 7. Spoj s *Pull-Down* otpornikomSlika 8. Napon na nožici (pinu) mikroupravljača uz korištenje *Pull-Down* otpornika i tipkala

Tipkalo koje koristimo na vježbama ima 4 kontakta. Dva kontakta su cijelo vrijeme spojena. Pritiskom na tipku međusobno su spojena sva 4 kontakta (slika 9).



Slika 9. Tipkalo koje se nalazi u kompletima koje se koriste na vježbama

Tipkala i sklopke možemo naći u puno različitih varijanti. Mogu imati jedan kontakt, kada se koriste kao tipkalo ili sklopka, a mogu imati i više kontakata kada se koriste kao preklopke (slika 10).



Slika 10. Različite varijante tipkala, sklopki i preklopki

Zadatak 1. Napiši program koji će provjeravati stanje na pinu 7 i ispisivati ga na **Serial monitor**. Na pin 7 treba dovoditi različite napone u skladu s točkama a) do h) u nastavku i promatrati na **Serial monitoru** kako mikroupravljač očitava pojedino stanje.

Kòd zadatka

```
int pin_SW1 = 7;          // pin 2 za tipkalo 1

int button1 = 0;          // varijabla koja će sadržavati stanje tipkala 1
                          // umjesto int moglo se koristiti i bool

void setup() {
    pinMode(pin_SW1, INPUT); // Definiramo PIN_SW1 kao ulazni
    Serial.begin(9600);      // inicijalizacija serijske komunikacije za
}                             // ispis na Serial monitor

void loop() {

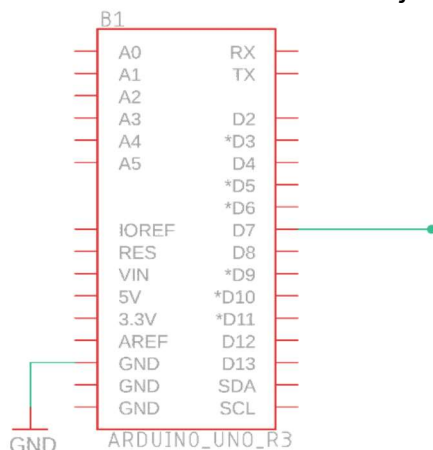
    button1 = digitalRead(pin_SW1); // ELEKTRIČNO STANJE NA PINU, TJ.
                                    // STANJE TIPKALA POHRANJUJEMO U
                                    // VARIJABLU button1 !!!

    Serial.print("SW1: ");          // ispis teksta „SW1:“ na Serial monitoru
    Serial.print("\t");             // ispis razmaka (Tab)
    Serial.print(button1);          // ispis varijable button1 na Serial monitoru
    Serial.print(„\n“);             // ili Serial.println(""); ispis u novi red

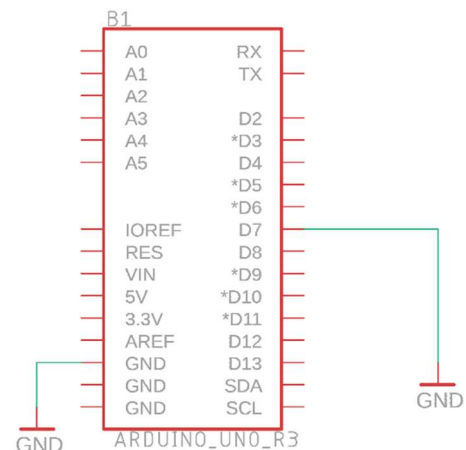
    delay(200);                    // pauza 200 ms
}
```

Za zadatke a) do c) potrebni su samo Arduino i jedan vodič. Za ostale zadatke koristiti i Breadboard. Isprobaj sve varijante od a) do h) i prati stanje pina na Serial monitoru!

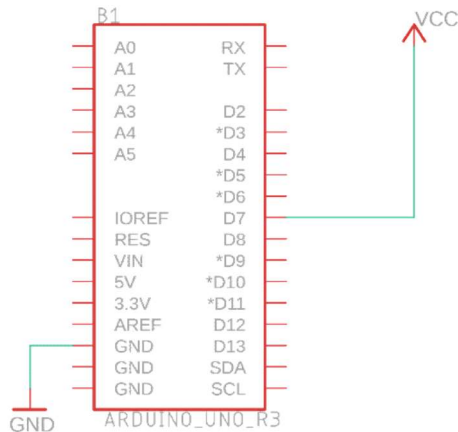
- a) Na pin 7 priključi vodič koji je na drugom kraju „u zraku“, tj. nije spojen nigdje. Prati je li očitavanje na Serial monitoru stabilno. Po potrebi dodirni Arduino ili vodič na različitim mjestima



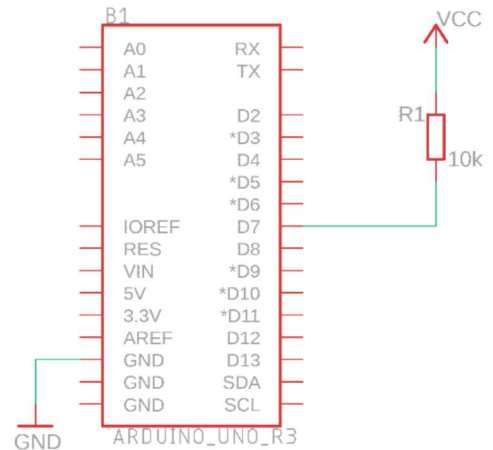
- b) Na pin 7 priključi vodič koji ćeš spojiti na GND



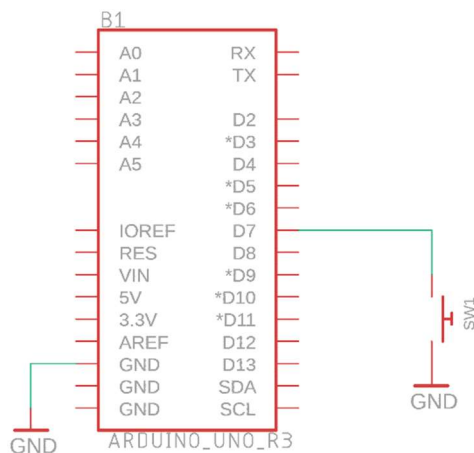
- c) Na pin 7 priključi vodič koji ćeš spojiti na +5V



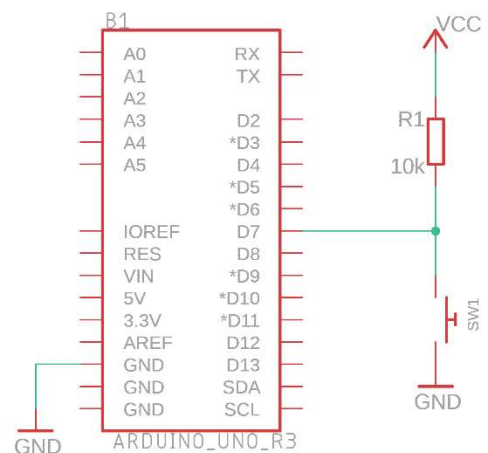
- d) Pin 7 priključi na +5V preko otpornika 10 kΩ (10k otpornik nije u kompletu, zatraži od profesora)



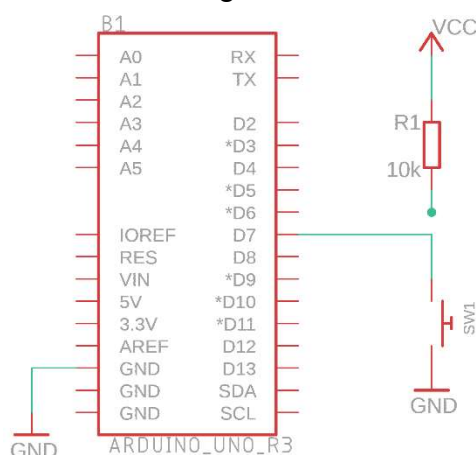
- e) Pin 7 priključi na GND preko tipkala



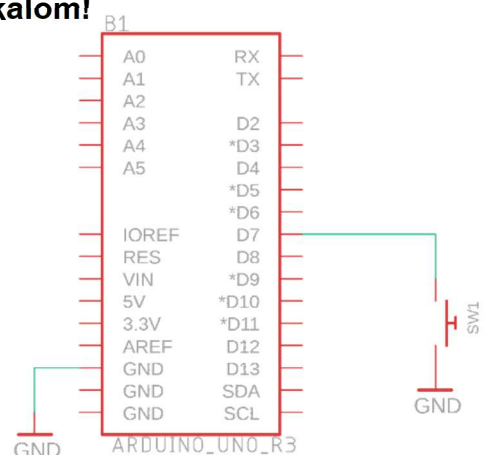
- f) Pin 7 priključi na +5V preko otpornika 10 kΩ i na GND preko tipkala kao što je prikazano na slici 1d i slici 3



- g) Odspoji 10 kΩ otpornik i pokušaj ponovo. Uoči da tipkalo osigurava logičku 0, ali ne osigurava +5V, tj. logičku 1. Uoči da bez Pull-Up otpornika nema logičke 1!



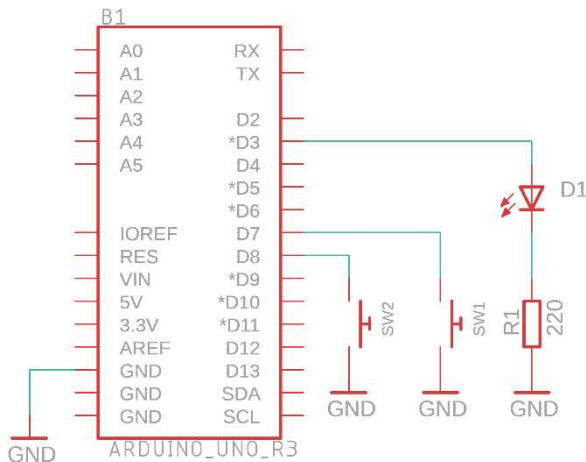
- h) **Važno! Uključi interni Pull-Up otpornik tako da naredbu `pinMode(pin_SW1, INPUT);` zamijeniš naredbom `pinMode(pin_SW1, INPUT_PULLUP);` Ovo je način rada koji treba koristiti ubuduće pri radu s tipkalom!**



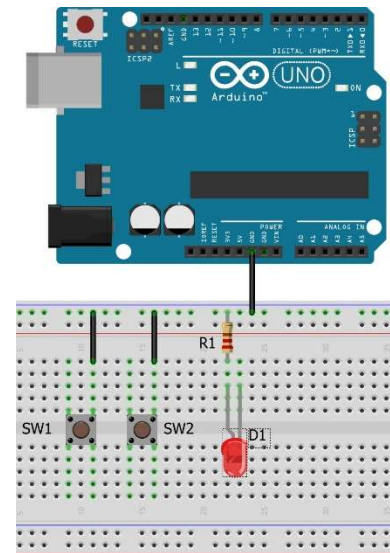
Zadatak 2. Složiti bravu koja će se otključati samo **kada su oba** ključa u bravi. Ključeve simuliraj tipkalicima, a otključana vrata LED diodom koja će svijetliti. Stanje tipkala treba pratiti na Serial monitoru! Tipkala i LED spoji na pinove prema priloženoj električnoj shemi. Zadatak riješi proširenjem koda iz prethodnog zadatka.

a) Objasni koja se logika koristi kod tipkala, a koja kod diode (pozitivna/negativna)?

Električna shema:

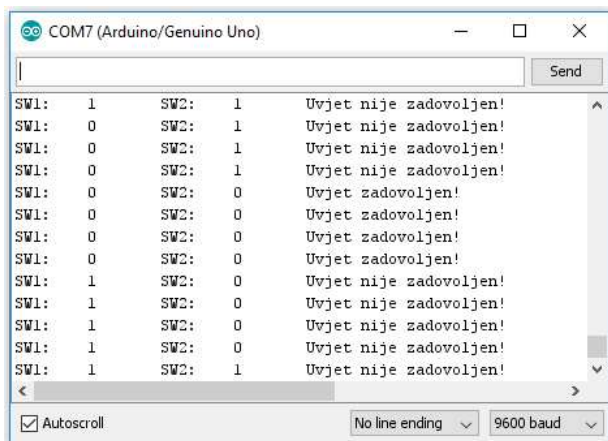


Prijedlog za raspored komponenta (dopuniti vodičima koji nedostaju):



Slika 11. Primjer električne brave s ključem

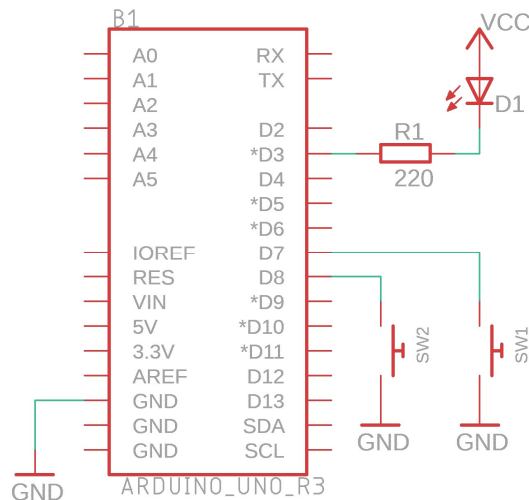
Ovako treba izgledati ispis na Serial monitoru!



Izvadak iz koda zadatka

```
void loop() { // proširiti loop() funkciju iz prethodnog zadatka...
...
    if (button1 == 0 && button2 == 0) { // ako su obje tipke pritisnute...
        digitalWrite(pin_LED, HIGH);
        Serial.print("\t");
        Serial.print("Uvjet zadovoljen!");
    }
    else {
        digitalWrite(PIN_led, LOW);
        Serial.print("\t");
        Serial.print("Uvjet nije zadovoljen!");
    }
    Serial.print("\n");
}
```

Zadatak 3. Modificiraj spoj iz prethodnog zadatka tako da diodu preko otpornika umjesto na GND spojiš na Vcc. Obrati pažnju na polarizaciju LED diode. Prilagodi kòd zadatka tako da i dalje LED dioda svijetli samo kad su pritisnute obje tipke. Kakva je sada logika kod LED diode?



Zadatak 4. Modificiraj spoj i kòd iz prethodnog zadatka tako da simuliraš infracrveni senzor i senzor temperature pomoću tipkala. U slučaju da **bilo koji** od senzora reagira (funkcija OR), treba se uključiti alarm. Alarm simuliraj LED diodom. LED ponovo spoji prema GND kao u zadatku 2.

Zadatak 5. Proširi spoj iz prethodnog zadatka s 4 dodatne diode i izvedi trčee svjetlo. Dioda se trebaju uključivati jedna iza druge, bez pauze. Za LED diode koristiti pinove 2 - 6. Duljina svijetljenja pojedine LED diode je 200 ms. U ovom zadatku, tipkala se privremeno ne koriste, ali ih ne treba odspajati.

Zadatak 6. Modificiraj program iz prethodnog zadatka deklariranjem globalne varijable na početku programa (`int nDelay = 200`) tako da se trajanje svijetljenja svih dioda može podešavati jednom varijablom. Umjesto naredbe **delay(200)** treba koristiti naredbu **delay(nDelay)**; Budući da se vrijeme trajanja svjetla nalazi u varijabli, moći će se primjerice pomoću tipkala ili potenciometra mijenjati brzina trčee svjetla.

Zadatak. Prouči kako bi zadatak 5 riješio koristeći **for petlju** pomoću koje ćeš skratiti duljinu kòda u slučaju da treba upravljati s više LED dioda. Ovaj zadatak ne treba spajati ni programirati, već samo proučiti!

Kòd zadatka

```
int led[5]={6,5,4,3,2}; // u vitičaste zgrade pišemo popis
                          // pinova Arduina koje ćemo koristiti
int i;                  // za upravljanje LED diodama
                          // (od 6 do 2)

void setup() {
  pinMode(led[0],OUTPUT);
  pinMode(led[1],OUTPUT);
  pinMode(led[2],OUTPUT);
  pinMode(led[3],OUTPUT);
  pinMode(led[4],OUTPUT);
}
```



```
void loop() {  
    digitalWrite(led[0],HIGH);           //prvu LED diodu moramo upaliti  
                                         // izvan petlje  
    delay(25);  
    for(i=0; i<5;i++){                  //u for petlji gasimo trenutnu  
        digitalWrite(led[i],LOW);       //LED diodu, a palimo sljedeću i  
        digitalWrite(led[i+1],HIGH);    //tako od prve do zadnje  
        delay(25);  
    }  
    digitalWrite(led[4],LOW);           //zadnju LED diodu je nužno  
                                         //ugasiti izvan petlje  
}
```

Zadatak 7. Modificiraj spoj i kôd zadatka tako da:

- pritisakom na tipkalo SW1 LED diode trče slijeva nadesno
- pritisakom na tipkalo SW2 LED diode trče zdesna nalijevo
- kad su oba tipkala pritisnuta, svijetle sve diode
- kad nije pritisnuto niti jedno tipkalo svijetli samo središnja dioda

Zadatak 8. Modificiraj prethodni zadatak dodavanjem dodatna 2 tipkala kojima ćeš regulirati svjetlinu LED dioda na pinovima 3, 5 i 6. Jednim tipkalom treba povećavati svjetlinu diode „+“ u, a drugim smanjivati „-“. Kad dioda dođe do maksimalne svjetline, pritiskom na tipku „+“, svjetlina diode se više ne mijenja. Isto tako i kad se dioda ugasi, pritiskom na „-“ ostaje ugašena. Koju naredbu treba koristiti? Predvidjeti da se svjetlina mijenja u koracima ± 10 ili 20.

Pitanja za provjeru znanja:

1. Na koji način dovodimo logičku 1 ili logičku 0 na ulaz digitalnog sklopa ili mikroupravljača?
2. Koju logičku razinu ima nožica koja je „u zraku“?
3. Na koji način se spaja tipkalo koje omogućuje dovođenje logičke 0 ili 1 na digitalni sklop?
4. Objasni način rada spoja s Pull-Up otpornikom!
5. Objasni način rada spoja s Pull-Down otpornikom!
6. Skiciraj valni oblik napona prilikom pritiska i otpuštanja tipkala s Pull-Up otpornikom.
7. Koja je prednost današnjih mikroupravljača po pitanju priključenja tipkala?
8. Navedi jednu od mogućih naredbi pomoću kojih se može uključiti interni Pull-Up otpornik?
9. Što je to *Switch Bouncing*?
10. Kako su povezani kontakti na tipkalu koje koristimo na vježbama?
11. Koja je razlika između tipkala, sklopke i preklopke?
12. Pomoću koje naredbe očitavamo stanje na pinu? Objasni!
13. Kada je potrebno uključiti *Pull-Up* otpornik, a kada ne?
14. Objasni naredbu `Serial.begin(9600)`!
15. Objasni naredbu `Serial.print()`!
16. Objasni kako pomoću `Serial.print` kreirati pravilan stupac ili ispisati znak u novi redak?
17. Objasni naredbu `Serial.println()`!

LITERATURA:

1. Gammon Forum, Nick Gammon, Switches tutorial, <http://www.gammon.com.au/switches>, (pregledano 4. listopada 2020.)