
ELEKTRONIKA - zbirka vaj

Ta zbirka vaj je namenjena študentom Pedagoške fakultete,
Univerze v Ljubljani.

dr. David Rihtaršič

2020-02-18 13:11

Kazalo

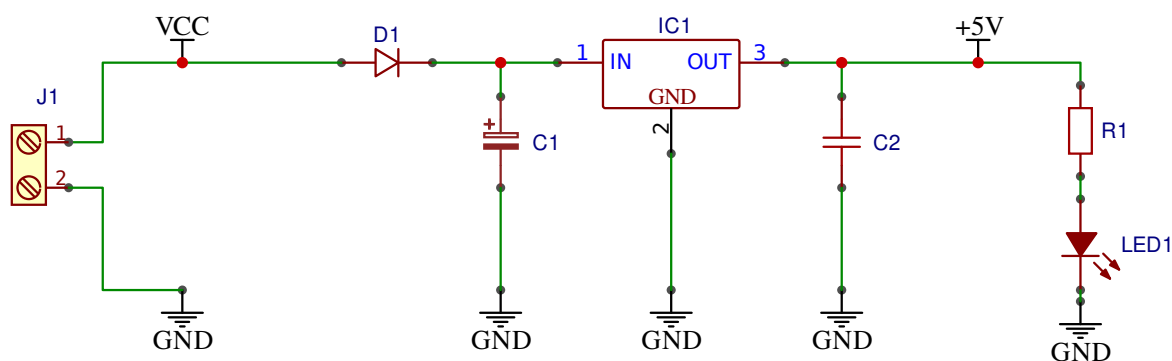
1	Napajanje	3
1.1	Napetostni potencial	4
1.2	Napetost	5
2	Krmilnik Arduino Nano	7
2.1	Testni program "BLINK.ino"	7
2.2	Napajanje krmilnika Arduino Nano	8
2.3	Model semaforja	10
2.4	Analiza vezja	11
3	Uporaba tipke	13
3.1	Priključitev tipke na digitalni vhod	14
3.2	Priključitev tipke z uporom proti napajanju	15
3.3	Uporaba uporov vezanih proti napajanju v mikrokrmilniku	15
4	Uporaba potenciometra	17
5	Dodatki in pogosta vprašanja	21
5.1	Barvno kodiranje upornosti	21

1 Napajanje

Vsako električno vezje potrebuje napajanje. Pogosto vezja napajamo z baterijami (prenosne električne naprave) ali pa z uporabo omrežnega napajanja. Za današnje vezje bomo uporabili omrežno napajanje. S pomočjo 9 V napajalnika in 5 mm napajalnega priključka (J1) dobimo napajalno napetost 9 V. Oba napetostna potenciala (+9 V in 0 V) napajalne napetosti priključimo na testno ploščico in od tam zgradimo vezje, ki bo zagotavljalo 5 V napajanje za naše nadaljnja vezja.

VAJA: SESTAVITE VEZJE NA PROTOTIPNI PLOŠČICI

Izgradite vezje po shemi 1.1.



Slika 1.1: Shema napajalnega vezja.

V pomoč bomo uporabili program **Fritzing**, s katerim lahko učitelj zelo nazorno pokaže kako moramo povezati elemente med seboj. S pomočjo Fritzing programa dobimo naslednjo skico *realnega* vezja.



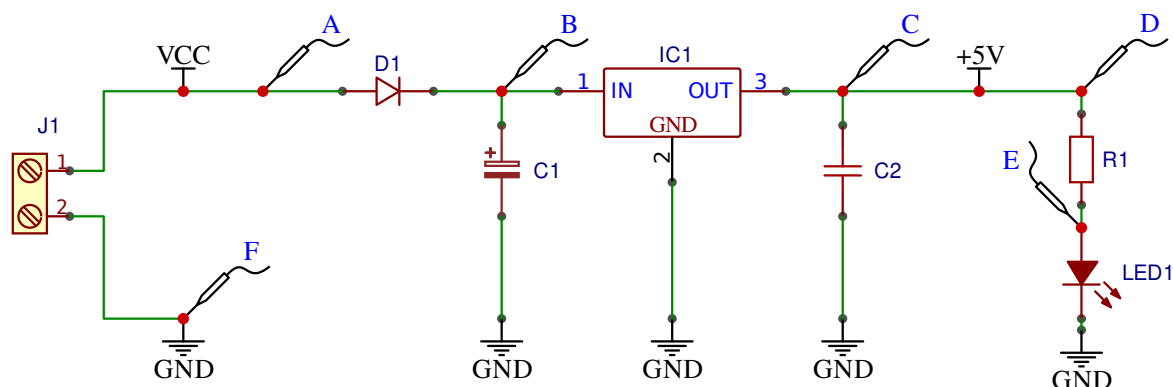
Slika 1.2: Skica realnega vezja.

1.1 Napetostni potencial

VAJA: IZMERITE NAPETOSTNE POTENCIALE

Izmeri napetostne potenciale, ki so vrisani v naslednji shemi 1.3.

Točka v vezju	Nap. potencial [V]
A	
B	
C	
D	
E	
F	



Slika 1.3: Shema električnega vezja 5 V napaljalne napetosti.

1.2 Napetost

V vezju imamo imamo kar nekaj elektronskih elementov. Na shemi 1.1 so različno označeni, npr.:

- polprevodniška dioda - D1,
- elektrolitski kondenzator - C1,
- keramični kondenzator - C2,
- upor - R1,
- svetleča dioda - LED1

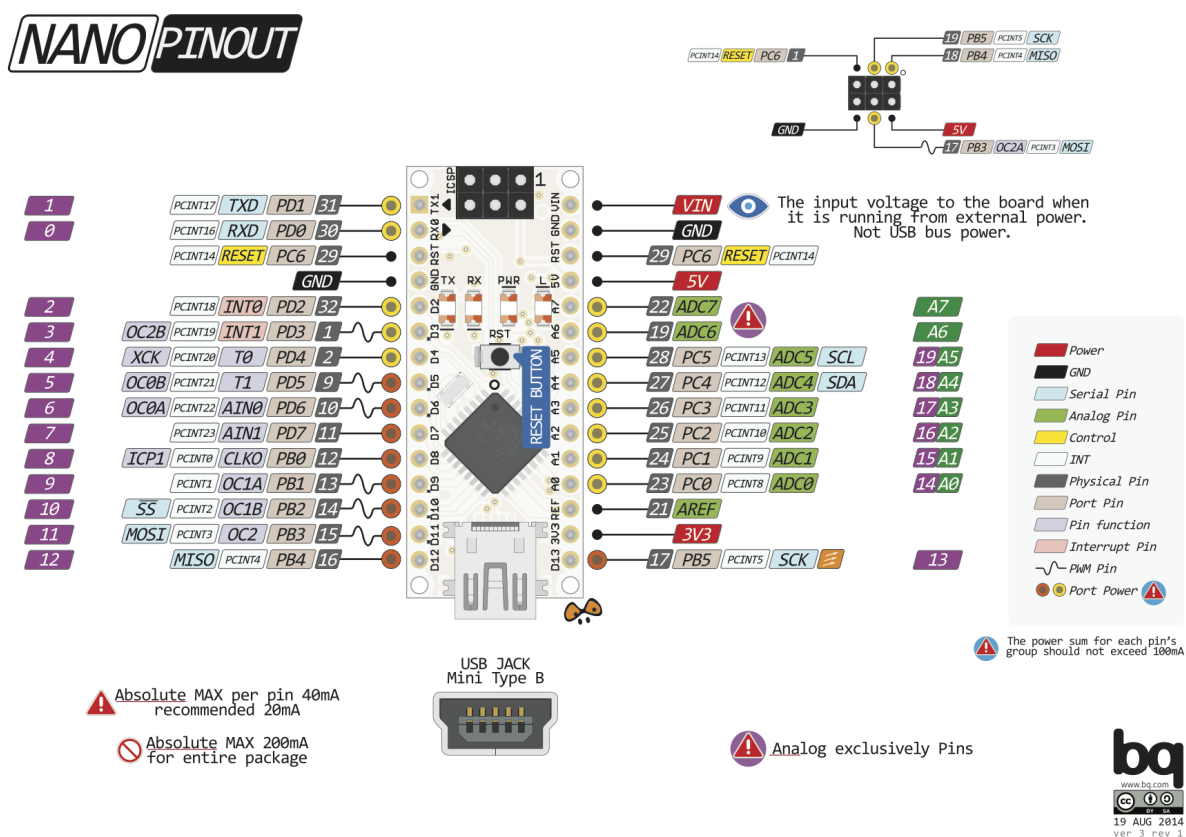
VAJA: IZRAČUNAJTE NAPETOSTI

Za vse našteje elemente najprej izračunajte kolikšna napetost je na njih, nato pa izračun preverite z inštrumentom.

Element	$U_{izr}[V]$	$U_{izm}[V]$
D1		
C1		
C2		
R1		
LED1		

2 Krmilnik Arduino Nano

Krmilnik Arduino Nano je relativno cenovno ugoden (cca. 3-5€) in ker je programirljiv, ga lahko uporabimo v najrazličnejših aplikacijah. Razporeditev njegovih priključkov pa lahko vidimo na naslednji sliki 2.1.



Slika 2.1: Razporeditev priključkov na krmilniku Arduino Nano.

2.1 Testni program “BLINK.ino”

Preden bomo krmilnik uporabili v našem vezju, ga bomo preizkusili. S programskim orodjem “Arduino IDE” bomo na krmilnik naložili program “blink.ino” in s tem preverili, da vse komponente na krmilniku

delujejo pravilno. To je priporočljivo narediti pred vsakim projektom.

VAJA: PREIZKUS KRMILNIKA ARDUINO NANO.

1. Krmilnik Arduino Nano povežite z računalnikom preko USB povezave,
2. zaženite program Arduino IDE in ga pravilno nastavite:
 - Tools -> Processor : Arduino Nano,
 - Tools -> Port : USB2
3. Odprite primer 01-BLINK.ino in
4. prenesite program na krmilnik.

Program je napisan v programskem jeziku C++, ki uporablja nekaj funkcij za lažje rokovanje s krmilnikom.

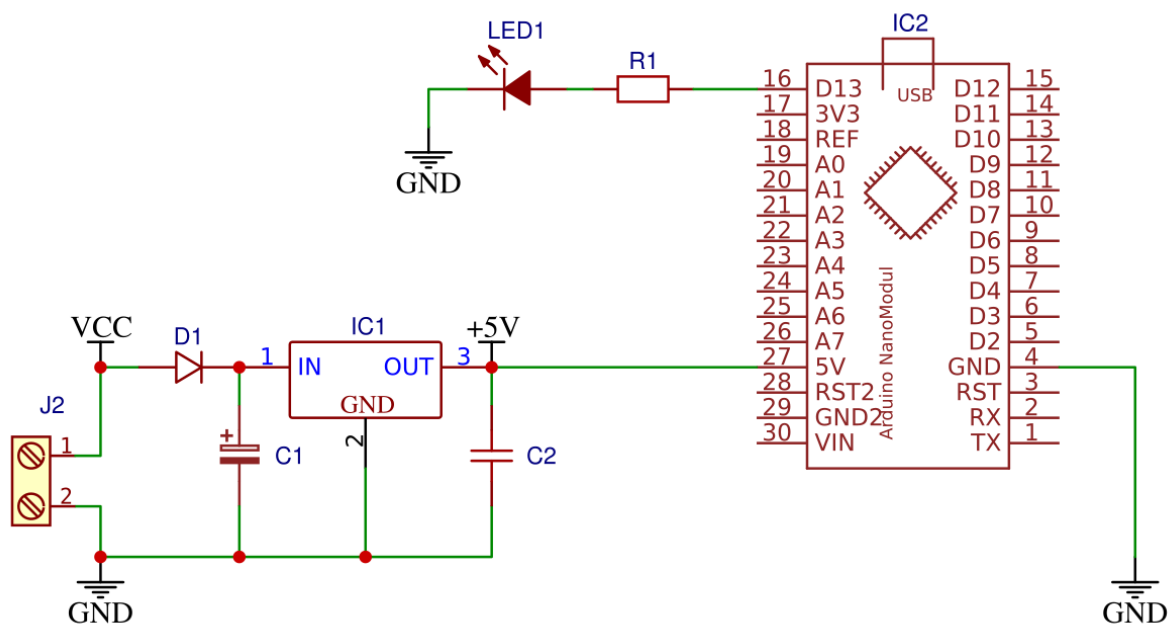
```
1  void setup() {  
2      // initialize digital pin LED_BUILTIN as an output.  
3      pinMode(LED_BUILTIN, OUTPUT);  
4  }  
5  
6  // the loop function runs over and over again forever  
7  void loop() {  
8      digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the  
9          voltage level)  
10     delay(1000);                        // wait for a second  
11     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the  
12         voltage LOW  
13     delay(1000);                        // wait for a second  
14 }
```

2.2 Napajanje krmilnika Arduino Nano

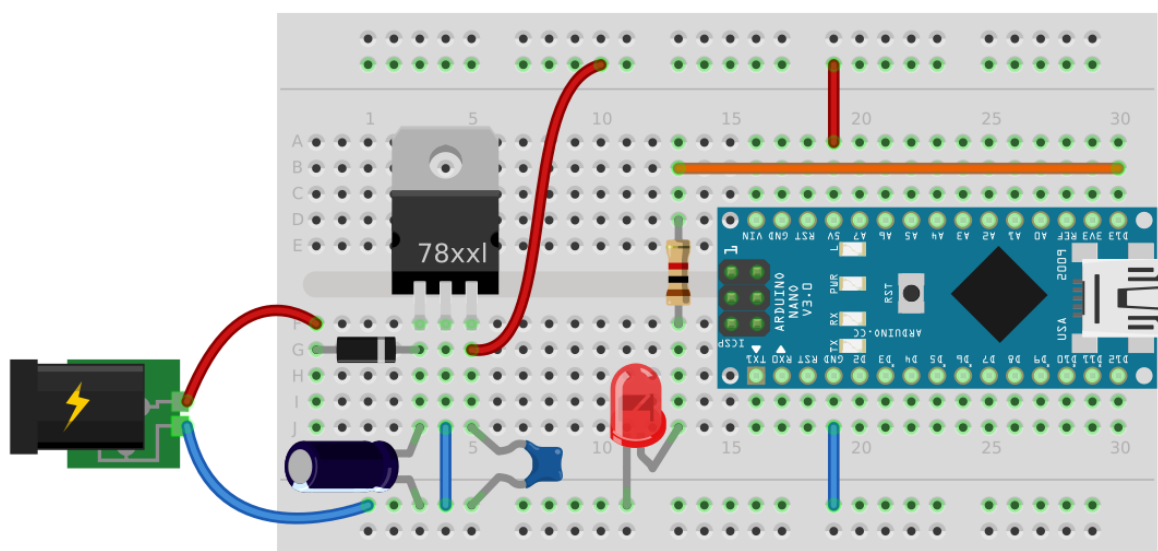
Krmilnik Arduino Nano lahko vstavimo tudi v prototipno ploščico in ga napajamo z zunanjim napajanjem.

VAJA: VEZAVA KRMILNIKA ARDUINO NANO NA PROTOTIPNI PLOŠČICI.

Vstavite krmilnik Arduino Nano v prototipno ploščico in ga povežite kot prikazuje naslednja shema. Priključite tudi upor in LED na priključek 13.



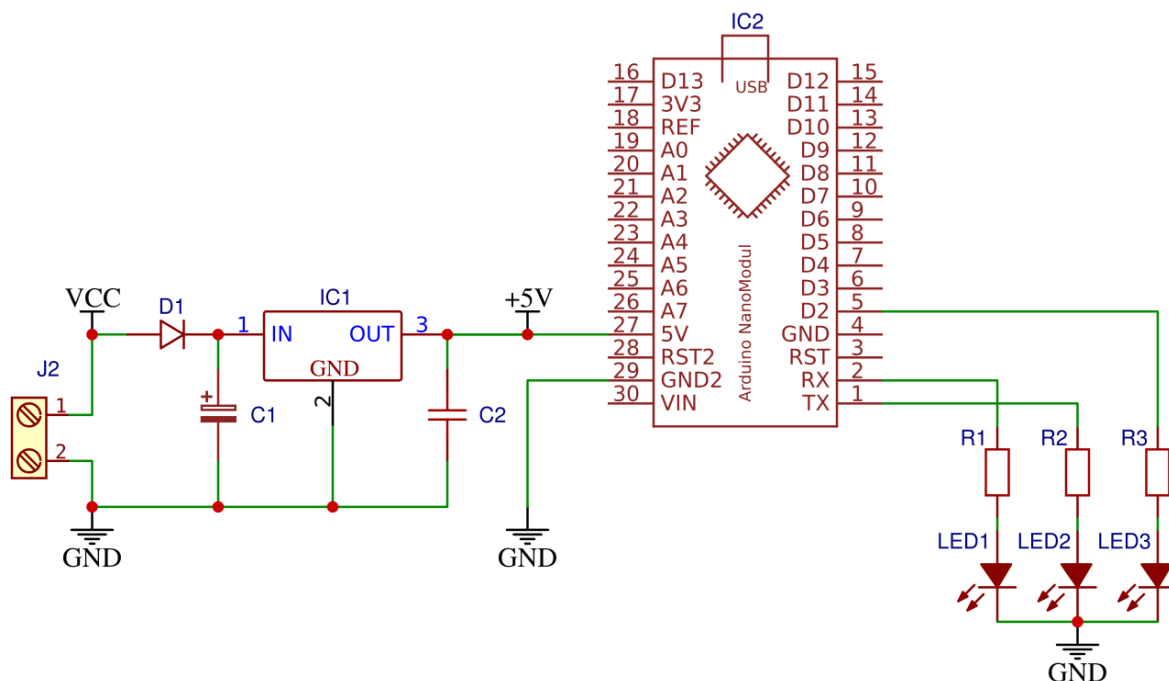
Slika 2.2: Priključitev napajanja in dodatne LED na izhodni priključek.



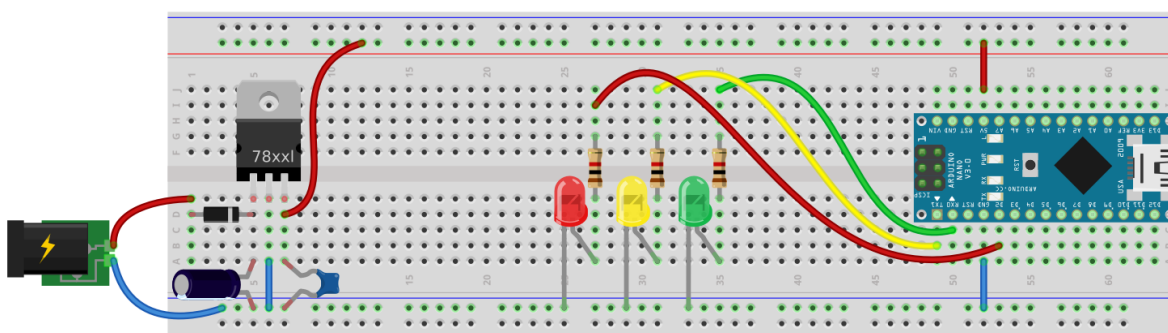
Slika 2.3: Shema vezave krmilnika Arduino Nano na prototipni ploščici.

2.3 Model semaforja

Vezje bomo preoblikovali tako, da bo delovalo kot semafor na cestnem križišču. Uporabili bomo tri LED svetila različnih barv in preoblikovali program.



Slika 2.4: Shema vezave treh LED na krmilnik Arduino Nano.



Slika 2.5: Ter shema vezave na prototipni ploščici.

VAJA: MODEL SEMAFORJA.

Preoblikujte vezje po shemi 2.4 in uporabite naslednji program ter ga ustrezno preoblikujte. Program, ki zagotavlja podobno delovanje, kot pri cestnem semaforju dokumentirajte in komentirajte uporabljene programske stavke (t.j. programske ukaze).

Vsak programski stavek morate zaključiti s podpičjem ;.

Preskustite naslednji program in ga ustrezno preoblikujte.

```
1  void setup() {  
2      pinMode(0, OUTPUT);  
3      pinMode(1, OUTPUT);  
4      pinMode(2, OUTPUT);  
5  }  
6  
7  void loop() {  
8      digitalWrite(0, HIGH);  
9      digitalWrite(1, HIGH);  
10     digitalWrite(2, HIGH);  
11     delay(1000);  
12     digitalWrite(0, LOW);  
13     digitalWrite(1, LOW);  
14     digitalWrite(2, LOW);  
15     delay(1000);  
16 }
```

2.4 Analiza vezja

Elektronski elementi so omejeni z njihovo največjo dopustno električno moč. Če to električno moč prekoračimo, jih bomo najverjetneje uničili.

Naprimera: Največja dopustna moč, ki se še lahko troši na uporih, ki jih uporabljate (premer upora = 2.4 mm) je 0,25 W.

Električno moč lahko izračunamo po enačbi:

$$P = UI$$

Pri nekaterih drugih elementih (kot na primer pri LED) pa so omejitveni pogoji postavljeni že s samim tokom.

Na primer za običajne 5mm LED je najpogostejše največji tok, ki lahko teče skozi 20 mA.

Tok skozi element lahko izračunamo po Ohmovem zakonu:

$$I_R = \frac{U_R}{R}$$

Če ne vemo kolikšno upornost ima element (tako kot je to v primeru LED), si največkrat pomagamo z izračunom toka skozi drug zaporedno vezan element. Kajti v tem primeru je tok isti.

VAJA: IZRAČUNAJTE ELEKTRIČNI TOK

Izračunajte kolikšen električni tok teče skozi elemente R1, R2, R3, LED1, LED2 in LED3 ter preverite kakšne so električne omejitve tega elementa. Izračunajte tudi električno moč, ki se troši na tem elementu.

Element	U [V]	I [mA]	P[W]
R1			
R2			
R3			
LED1			
LED2			
LED3			

3 Uporaba tipke

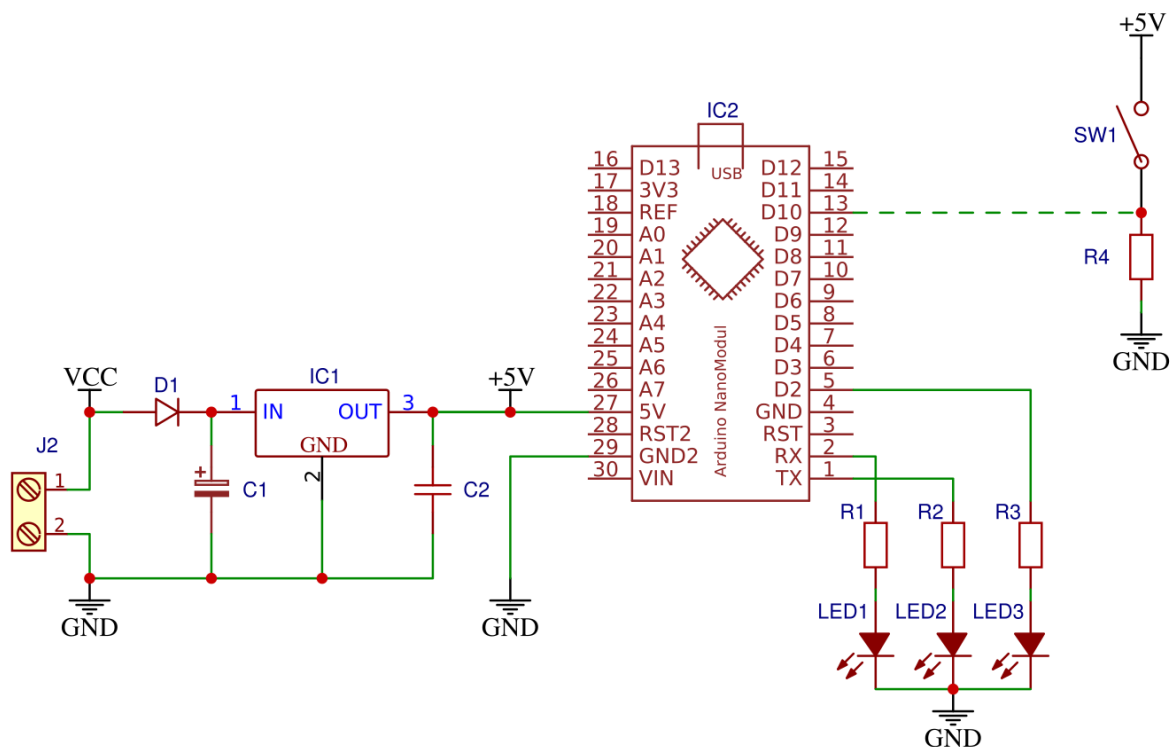
Tipka je element, ki je podoben stikalu. Prav tako kot pri stikalu, ob pritisku povežemo dva priključka med sabo, razlika pa je v tem, da se pri tipki ta povezava razklene ob prenehanju pritiska nanjo.

VAJA: VEZAVA TIPKE V DELILNIK NAPETOSTI

Sestavite vezje s tipko, kot ga prikazuje 3.1 (brez črtkane povezave). Tipka naj bo vezana v delilnik napetosti in naj bo priključena proti napajanju.

V tabelo vpišite napetosti na vsakemu elementu posebej. Najprej poskušajte predvideti kolikšna je napetost na elementu, nato pa le-to preverite z inštrumentom.

Situacija	Element	$U_?$ [V]	U_{izm} [V]
Razkljenjeno	Tipka		
	Upor		
Slenjeno	Tipka		
	Upor		



Slika 3.1: Vezava tipke v delilnik napetosti.

3.1 Priključitev tipke na digitalni vhod

Nato srednji priključek delilnika napetosti povežite na digitalni vhod krmilnika Arduino nano na priključek D10, kot prikazuje 3.1 črtkana povezava.

VAJA: PRIKLJUČITEV TIPKE NA DIGITALNI VHOD

Priključite tipko po shemi 3.1 in preskusite spodnji program.

Nato popravite program tako, bo LED svetila, ko boste tipko pritisnili.


```
1 void setup() {  
2     pinMode(0, OUTPUT);  
3     pinMode(1, OUTPUT);  
4     pinMode(2, OUTPUT);  
5     pinMode(10, INPUT);  
6 }  
7  
8 void loop() {  
9     if (digitalRead(10) == LOW){  
10        digitalWrite(0, HIGH);  
11    }else{  
12        digitalWrite(0, LOW);  
13    }  
14 }
```

POMNI: DIGITALNI VHOD MIKROKRMILNIKA

Priključke mikrokrmilnika lahko uporabimo tudi za odčitavanje napetostnih potencialov v digitalni obliki (ločimo le dva napetostna nivoja). S funkcijo `pinMode(PIN, INPUT);` določimo, da priključek `PIN` lahko opravlja funkcijo digitalnega vhoda s katerim lahko odčitamo vrednost napetostnega potenciala. Funkcija `digitalRead(PIN);` vrne vrednost digitalnega vhoda. Če je na priključku `PIN` napetostni potencial večji od 2.0 V bo funkcija vrnila vrednost 1 (ali HIGH ali TRUE). Če pa je na priključku `PIN` napetostni potencial manjši od 0.8 V pa bo funkcija vrnila vrednost 0 (ali LOW ali FALSE).

3.2 Priključitev tipke z uporom proti napajanju

Zamenjajte elementa v delilniku napetosti tako, da bo upor vezan proti napajanju in tipka proti napetostnemu potencialu 0 V.

VAJA: UPOR VEZAN PROTI NAPA JANJU

Vezje spremenite, kot je opisano v nalogi in narišite shemo vezja.

Popravite program tako, da bo LED utripala, ko boste tipko držali. Utemeljite zakaj je sedaj napetostni potencial na vhodnem priključku krmilnika D10 enak 5 V, ko tipka ni pritisnjena (Utemeljitev podprite z Ohm-ovim zakonom in Kirchoff-ovima izrekoma).

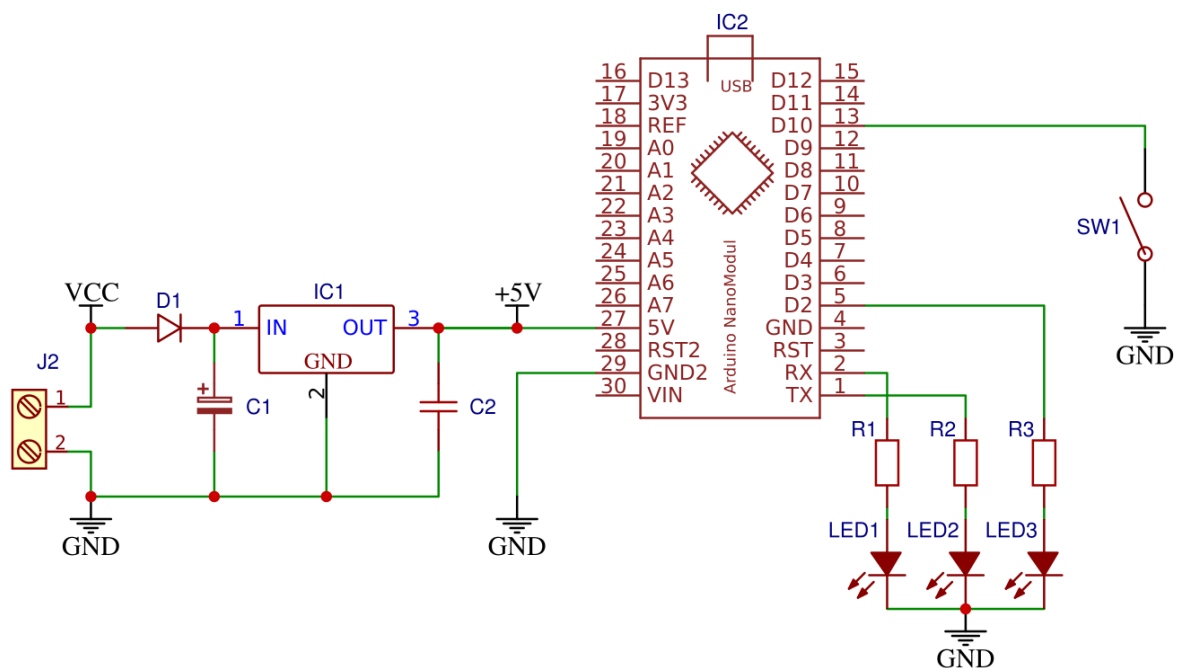
3.3 Uporaba uporov vezanih proti napajanju v mikrokrmilniku

Uporaba vezave uporov proti napajanju je zelo pogosta. Zato le to mikrokrmilniki že vsebujejo v samem integriranem vezju na vseh digitalnih vhidih.

VAJA: UPORABA UPOROV VEZANIH PROTI NAPAJANJU V MIKROKRMILNIKU

Odstranite upor iz vezja kakor veleva shema 3.2 in

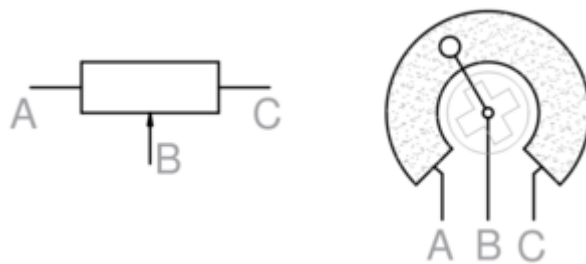
programsko vključite upor vezan proti napajanju na digitalnem vhodu D10.



Slika 3.2: Vezava tipke z notranjim uporom vezanim proti napajanju.

4 Uporaba potenciometra

Potenciometri so upori s tremi priključki, ko je prikazano na 4.1. Upornost potenciometra je fiksna in jo merimo med priključkoma A in C. Tretji priključek pa je nastavljiv in drsi po uporovni plati od ene skrajne lege do druge skrajne lege potenciometra.



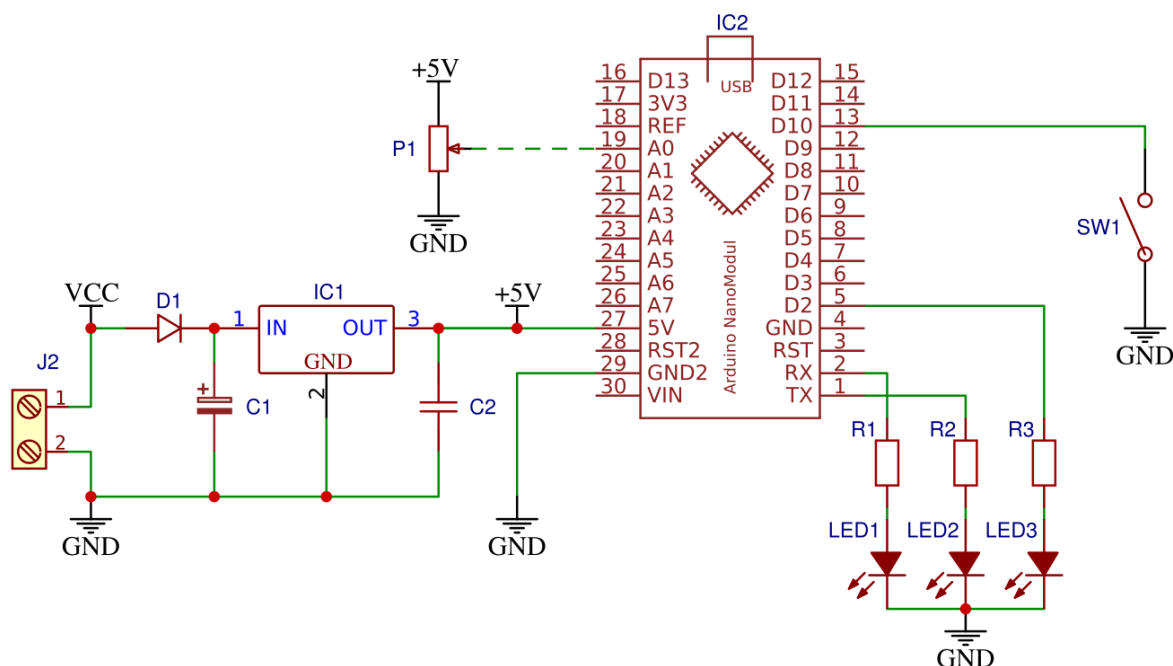
Slika 4.1: Simbol in shema potenciometra.

Delovanje potenciometra si lahko razložimo na dva načina:

1. Če na priključka A in C priključimo neko napetost, se bo napetostni potencial enakomerno zmanjševal vzdolž uporovne plasti potenciometra. Tako je napetostni potencial na priključku B odvisen od njegove lege.
2. Lahko si zamislimo, da priključek B razdeli potenciometer na dva upora: (1) R_{AB} in (2) R_{BC} . Tako se bo tudi napetost, ki jo bomo priključili na potenciometer razdelila v razmerju teh dveh upornosti.

1 VAJA: MERJENJE NAPETOSTNEGA POTENCIALA NA POTENCIOMETRU.

2 Priključite potenciometer tako kot je prikazano na [fig:35-potenciometer.png]. Na srednji priključek potenciometra priključite V-meter. Preverite kako se napetostni potenciala spreminja v odvisnosti od položaja srednjega priključka potenciometra.



Slika 4.2: Priključitev potenciometra.

VAJA: ODČITAVANJE ANALOGNE NAPETOSTI S KRMILNIKOM

Povežite srednji priključek potenciometra na analogni vhod krmilnika (naprimer na A0) in preizkusite naslednji program. Program lahko najdete tudi v Arduino IDE programu:

File -> Examples -> 01. Basics -> AnalogReadSerial.

```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   int sensorValue = analogRead(A0);
7   Serial.println(sensorValue);
8   delay(10);
9 }

```

Kot ste verjetno opazili, se vam na ekranu v serijskem oknu izpisujejo številske vrednosti. Te vrednosti so v območju od 0..1023, saj je ADC v mikrokrmilniku 10-bitni in je največja možna binarna številka zapisana z 10-imi biti prav 1023. Lahko pa te vrednosti prikazujete tudi grafično, v ta namen morate uporabiti [Serial Plotter](#).

Krmilnik pa bi lahko na ta način (do neke mere) uporabljali tudi kot V-meter.

- 1 VAJA: PRETVORBA ADC VREDNOSTI V NAPETOST
- 2 Z ustrezno linearno funkcijo pretvorite ADC vrednosti v številске vrednosti napetosti. Enačbo funkcije tudi zapišite in priložite kodo programa.

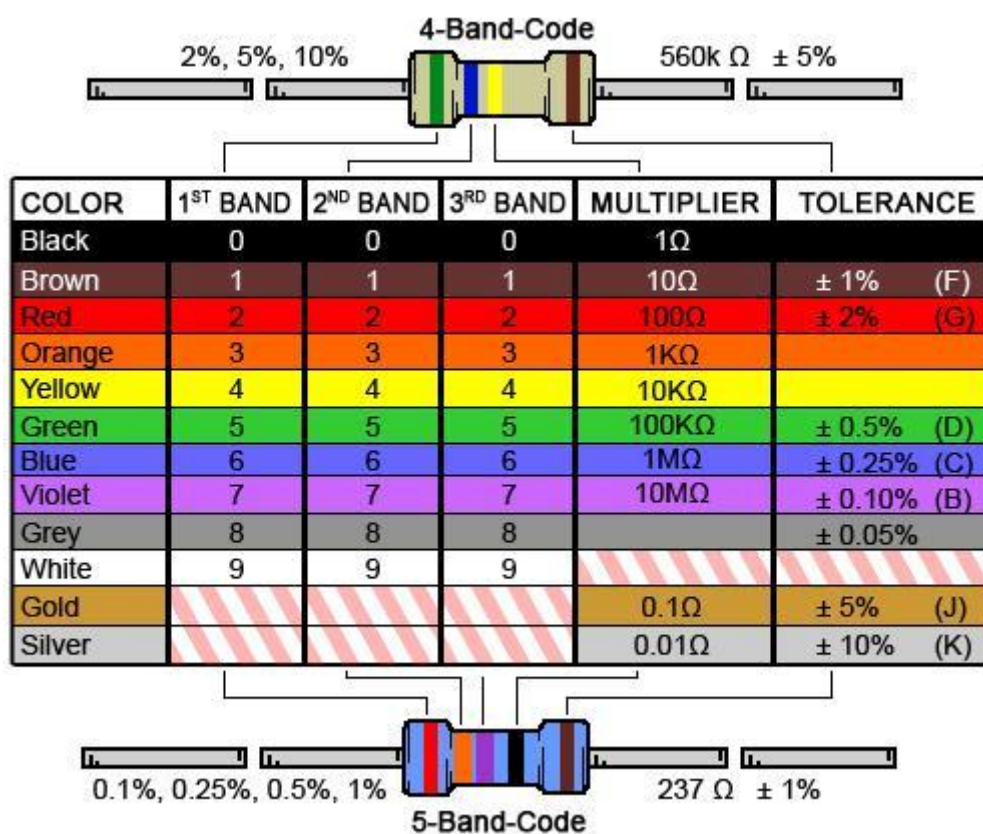
Prav tako enostavno pa lahko poskrbimo za grafični prikaz napetosti...

- 1 VAJA: VU-METER
- 2 Program preoblikujte tako, da ko boste s potenciometrom nastavili večjo napetost, naj se vključi vač LED. Podobno kot je to na VU-metru na glasbenih stolpih. Kodo programa tudi priložite.

5 Dodatki in pogosta vprašanja

5.1 Barvno kodiranje upornosti

Upornosti na uporih so bravno kodirane kot jih prikazuje naslednja slika 5.1



Slika 5.1: Bravno kodiranje vrednosti uporov.

