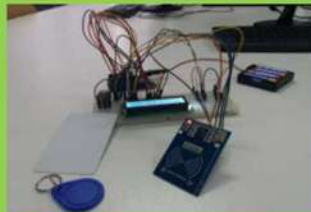


Didaktički set za ROBOTIKU



TUTORIAL

UVOD

Šta očekivati?

Dragi čitatelju, dobrodošao u svijet mikrokontrolera!

Ovaj tutorial namijenjen je svima koji žele naučiti ponešto o elektronici, mikrokontrolerima i programiranju. Tutorial će Vas kroz jednostavne primjere voditi korak po korak kroz to šta su mikrokontroleri, kako rade, kako ih možemo programirati te šta s njima možemo uraditi.

Tutorial možete koristiti na dva načina. Ako ste potpuni početnik preporučujem Vam da tutorial čitate od početka do kraja, poglavlje po poglavlje, rješavajući zadatak po zadatak i na taj način upoznate način rada mikrokontrolera, programiranje i načine spajanja određenih elektroničkih komponenti na mikrokontroler. Ukoliko ste već radili s mikrokontrolerima ili imate neko predznanje, tutorial možete koristiti kao zbirku zadataka za vježbu ili možete pročitati pojedino poglavlje kako biste se upoznali s načinom korištenja elektroničke komponente koja se u tom poglavlju pojavljuje.

Mikrokontroler – šta je to?

Već smo nekoliko puta spomenuli mikrokontrolere, a zapravo nismo pojasnili šta su to mikrokontroleri. Postoje razne definicije, a najjednostavnije rečeno mikrokontroler je računar u malom. Mikrokontroleri su mali računari smješteni na jedan integrisani sklop. Na sljedećoj slici prikazan je mikrokontroler s kojim ćemo raditi u ovom tutorialu – njegov naziv je ATMEGA 328 a proizvodi ga kompanija ATMEL.



Unutar tog integrisanog sklopa nalaze se razni sastavni dijelovi mikrokontrolera, kao što su processor, radna memorija, programska memorija ulazne i izlazne jedinice itd.

Za početak možete mikrokontroler zamisliti kao crnu kutiju s desetak izvoda. Izvodi mikrokontrolera služe kako bismo na njega mogli spajati neke elektroničke elemente i s njima upravljati pomoću mikrokontrolera.

Da bi mikrokontroler znao šta zapravo treba raditi i kako da upravlja s onime što smo na njega spojili za njega moramo napisati program a potom taj program moramo učitati u mikrokontroler.

Kako bi cijeli postupak korištenja mikrokontrolera – spajanja elektroničkih komponenti na njega, njegovog programiranja te korištenja bio što jednostavniji osmišljeni su razni alati koji nam olakšavaju njihovo korištenje. Jedan od najpoznatijih alata koji se u svijetu najviše koristi upravo za učenje i početak rada s mikrokontrolerima je Arduino, pa ćemo tako i mi, kroz ovu knjigu i njezine primjere savladati mikrokontrolere upravo pomoću Arduina.

Arduino

Arduino je platforma za učenje programiranja i korištenja mikrokontrolera nastao prije desetak godina. Njegovi kreatori kažu da je Arduino open – source platforma za kreiranje elektroničkih prototipova bazirana na sklopovima i programskom paketu koji je fleksibilan i jednostavan za korištenje. Arduino je namijenjen umjetnicima, dizajnerima, hobistima, elektroničarima, učenicima i svima koji su zainteresovani za kreiranje interaktivnih objekata ili okruženja.

Jednostavnije, Arduino platforma je skup elektroničkih i programskih dijelova koji se mogu jednostavno povezati u složenije cjeline s ciljem izrade zabavnih i poučnih elektroničkih sklopova.

Kao što vidite, Arduino je namijenjen svima, velikima i malima, elektroničarima, tehničarima ali i umjetnicima i kreativcima da lakše i brže ostvare svoje ideje.

U ovom tutorialu koristit ćemo jednu od Arduino pločica pod nazivom – Arduino UNO. Pločica je prikazana na sljedećoj slici.



Srce Arduino UNO pločice je upravo ATMEGA 328 mikrontroler. Sklopovi koji se nalaze oko njega služe kako bi se mikrokontroler mogao pokrenuti i kako bismo u njega mogli ubaciti program koji pišemo na računaru. Komunikacija između Arduino UNO pločice i računara odvija se preko USB priključka.

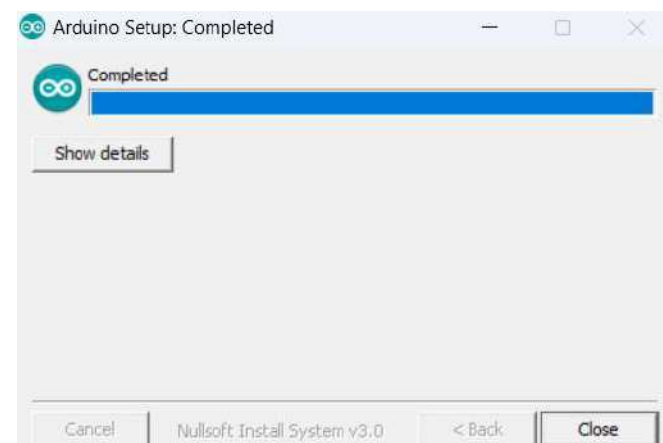
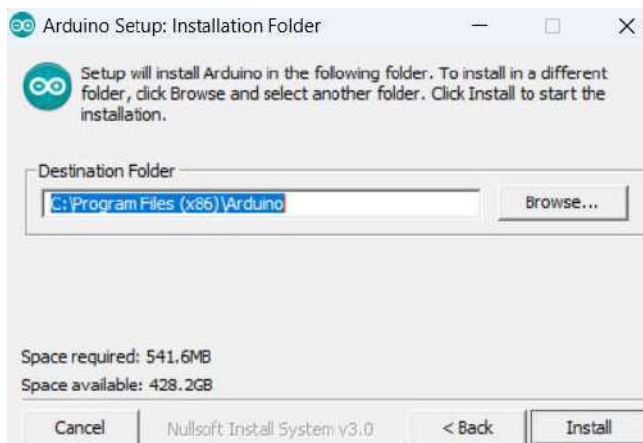
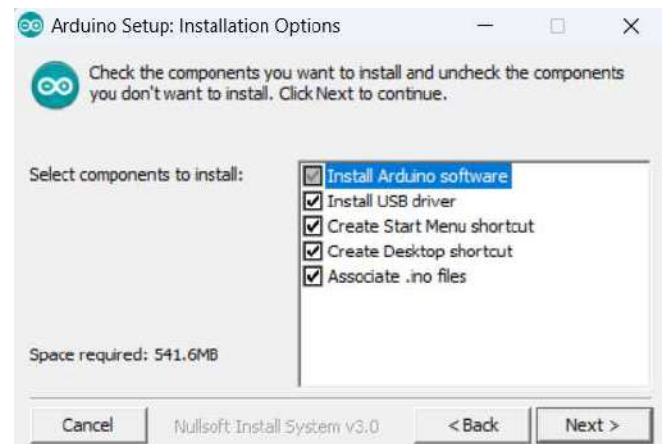
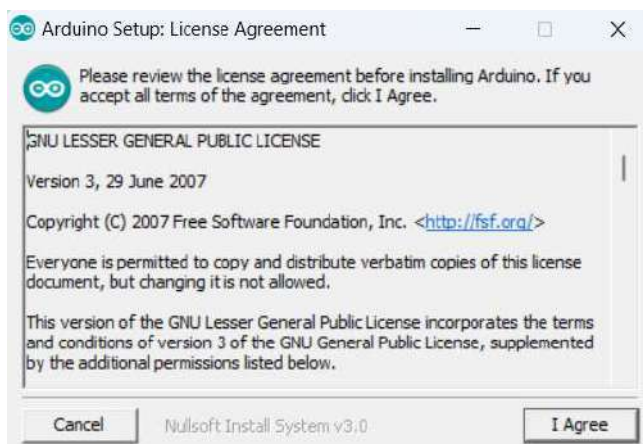
Korišteni programski alati

U knjizi se koristi nekoliko računarskih alata. Za programiranje mikrokontrolera koristit ćemo Arduino IDE programsko okruženje. Više o njemu u sljedećem poglavlju. Za prikaz spajanja Arduino UNO pločice i elektroničkih elemenata na eksperimentalnoj pločici, te shematske prikaze spajanja korišten je besplatni program pod nazivom Fritzing. Više informacija o tom programu možete pronaći na www.fritzing.org. Njega ne morate nužno preuzimati i instalirati, ali bi Vam mogao dobro doći u budućnosti za dokumentovanje Vaših projekata s mikrokontrolerima.

Preuzimanje, instalacija i podešavanje

Kako bi započeli s radom, moramo preuzeti i instalirati Arduino softver (IDE - Integrated development environment) koji se koristi za pisanje i učitavanje (upload) koda na Arduino mikrokontrolere. Arduino IDE možete besplatno preuzeti na sljedećoj adresi: www.arduino.cc.

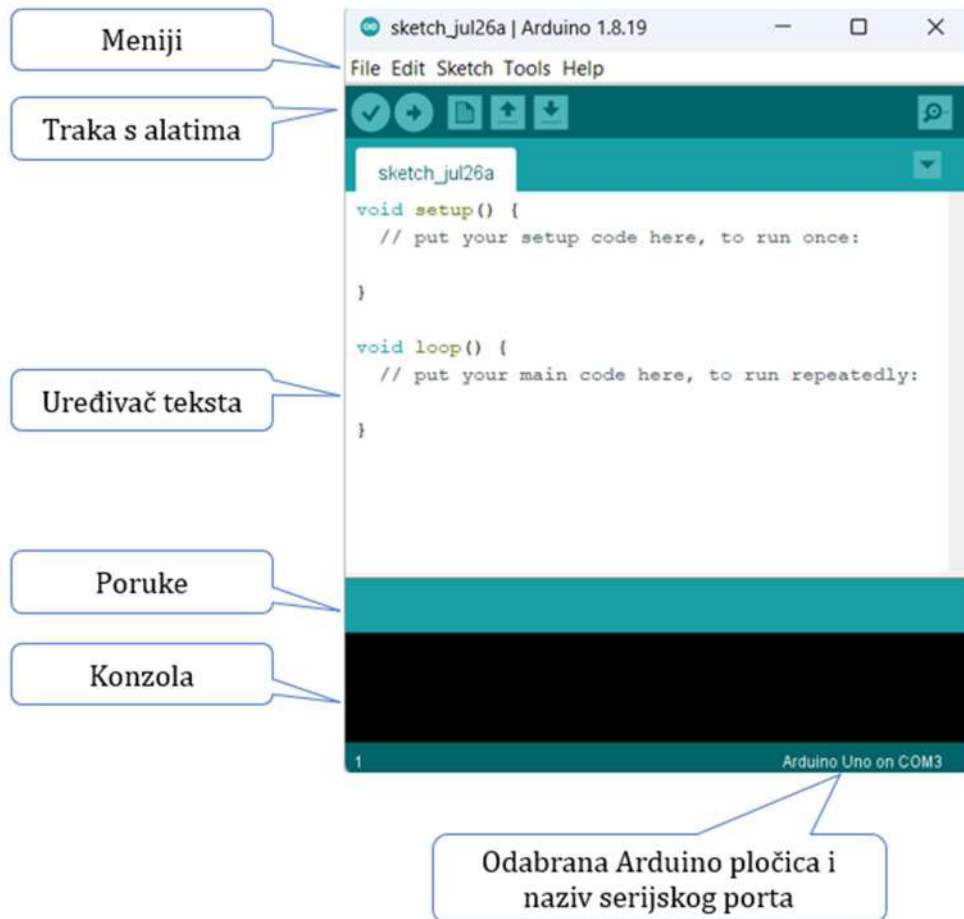
Nakon preuzimanja programa pokrenite instalaciju. Nakon nekoliko pritisaka na tipku *next* uspješno ste instalirati Arduino IDE. Postupak instalacije prikazan je korak po korak na sljedećim slikama.



Nakon završetka instalacije, na radnoj površini će se generirati prečica Arduino softvera. Pokrenite Arduino softver.



Interfejs Arduino softvera je sljedeći:



Programi napisani pomoću Arduino softvera (IDE) nazivaju se skice. Skice napisane u uređivaču teksta (Tekst editor) spremaju se s nastavkom datoteke .ino. Uređivač ima funkcije za izrezivanje/ljepljenje i traženje/zamjenu teksta.

Područje za poruke daje povratne informacije prilikom spremanja i učitavanja , a također se prikazuju i greške.

Konzola prikazuje tekstualni izlaz Arduino softvera (IDE), uključujući kompletne poruke o grešci i druge informacije.

U donjem desnom uglu prozora prikazana je priključena Arduino pločica i broj serijskog porta.

Dugmad na traci s alatima omogućavaju provjeru i učitavanje programa, kreiranje, otvaranje i čuvanje skice kao i otvaranje serijskog monitora.



Verify – Provjerava greške u kodu prilikom kompajliranja.



Upload – Učitava program na priključenu pločicu.



New – Kreira novu skicu.



Open – Otvara postojeće skice.



Save – Sprema skicu.

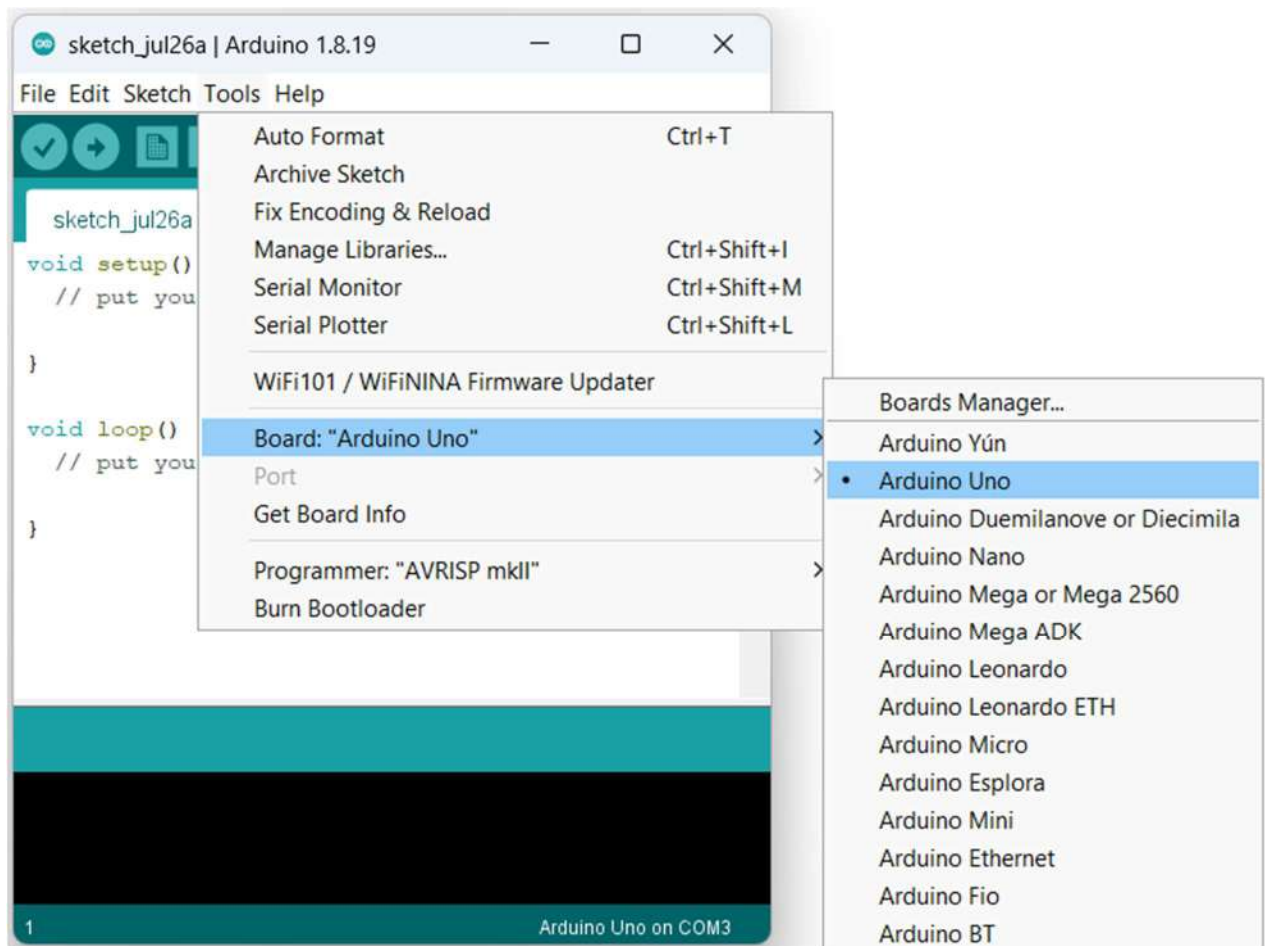


Serial Monitor – Otvara serijski monitor.

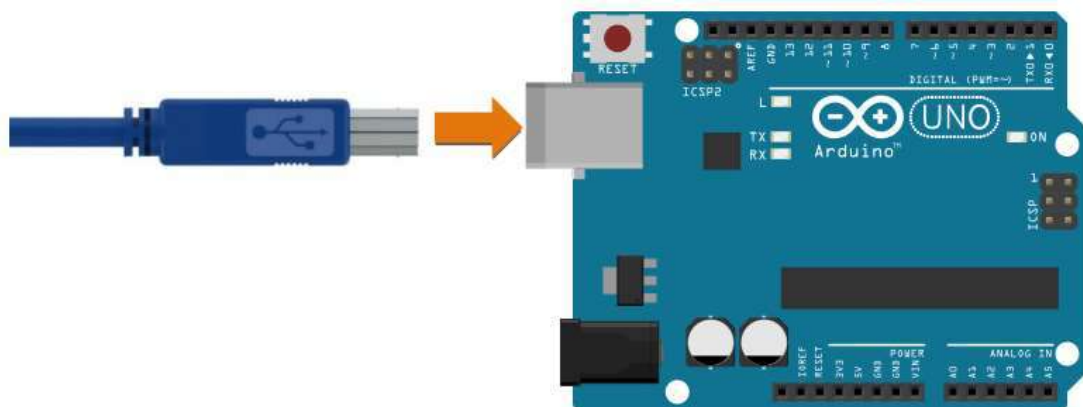
Dodatne komande nalaze se unutar pet menija: File, Edit, Sketch, Tools, Help. Meniji su kontekstno osjetljivi, što znači da su dostupne samo one stavke relevantne za posao koji se trenutno obavlja.

Prije početka rada potrebno je podesiti neke od parametara Arduino IDE programa. Moramo odabrati koju Arduino pločicu koristimo, na koji je port (priključak) računara spojena te na koji način želimo slati programe na mikrokontroler. Ova podešavanja dovoljno je napraviti jednom ako ne mijenjate pločicu ili priključak računara.

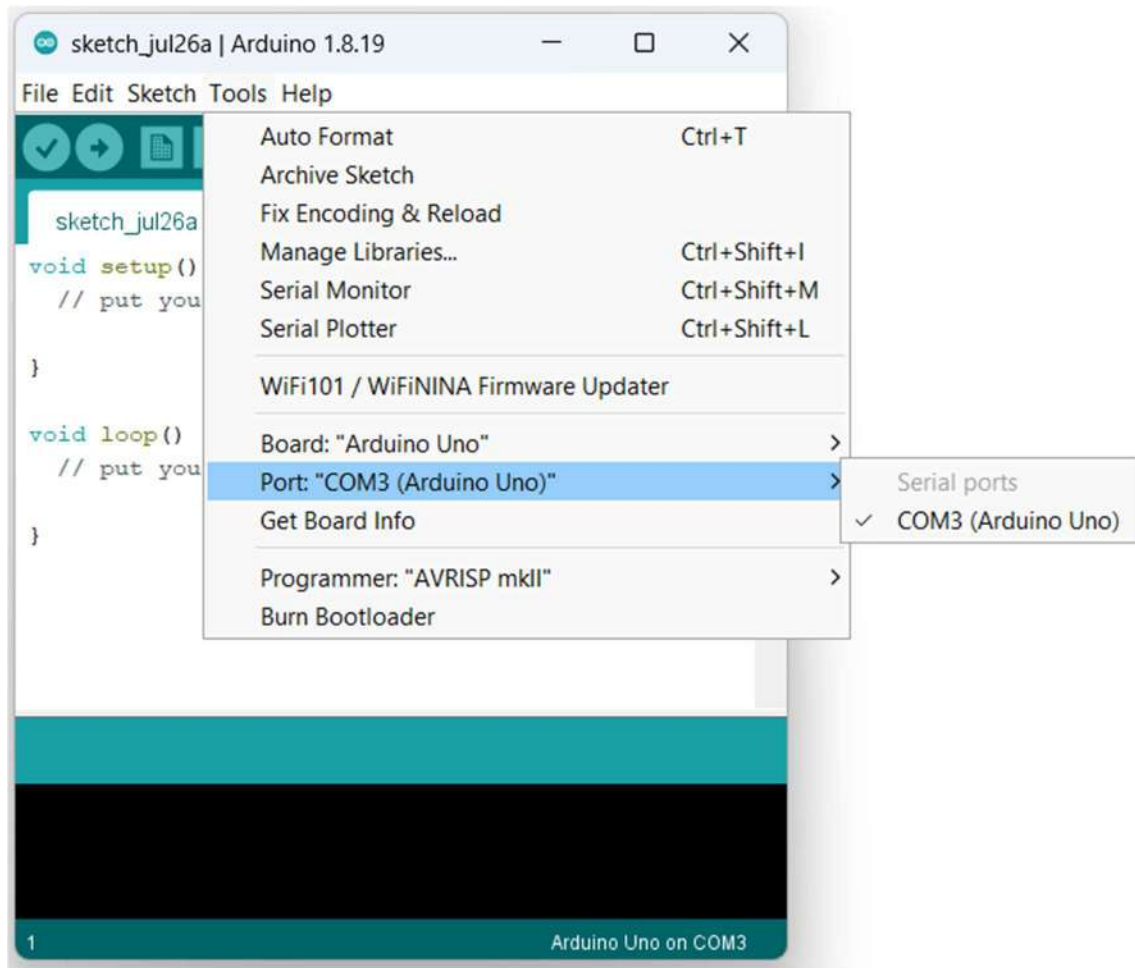
Za odabir pločice kliknite na **Tools** u meniju, potom na **Board** i odaberite **Arduino Uno**.



Povežite Arduino Uno na računar pomoću USB kabela.

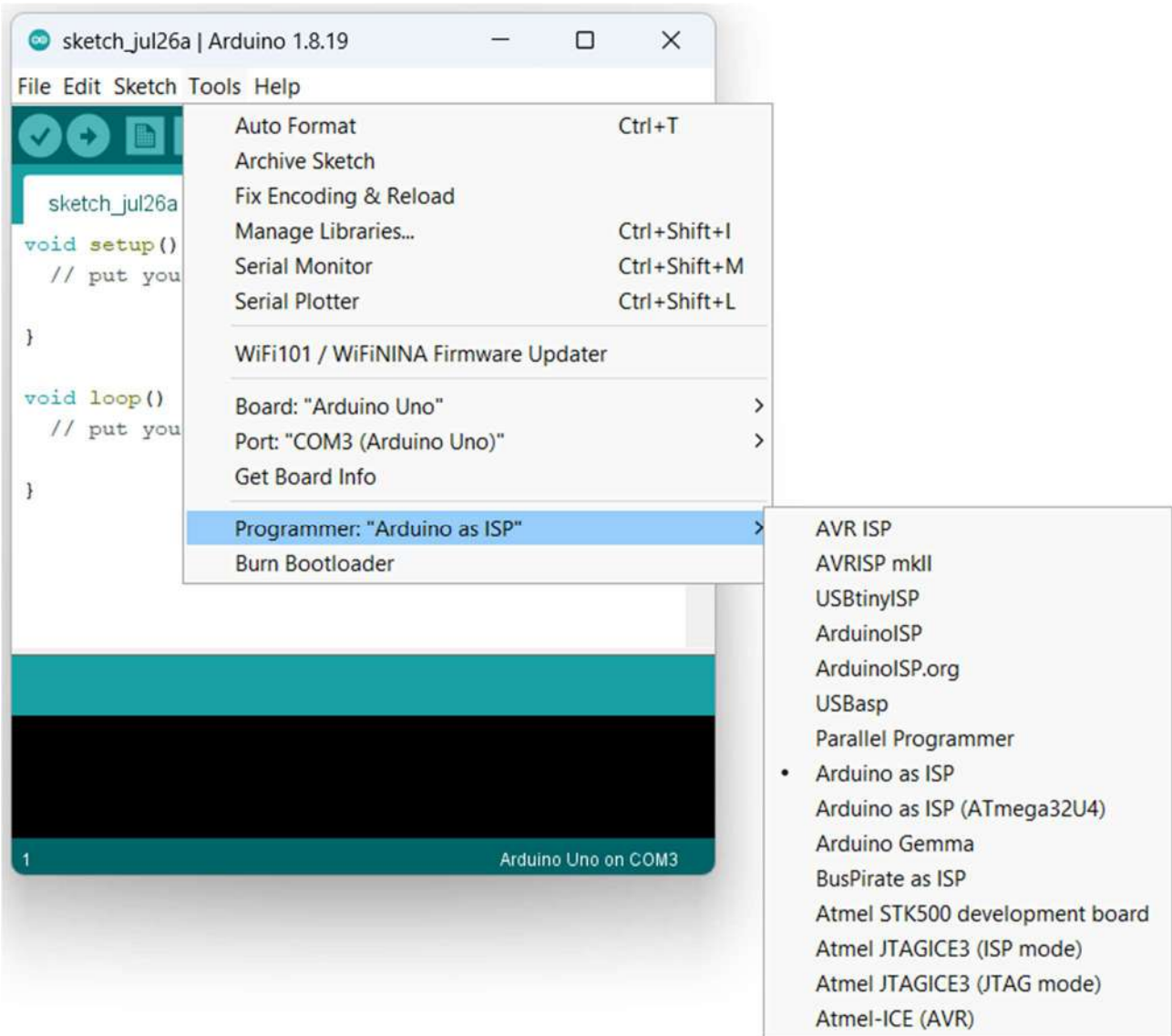


Za odabir porta računara kliknite na **Tools** u meniju, potom na **Port** te odaberite port na koji je spojena Vaša Arduino pločica.



Ukoliko je prikazano više serijskih priključaka (portova) i ne znate koji priključak odabrati pokrenite upravitelj uređaja (Device manager) i pronađite Arduino Uno pločicu. Kraj nje stoji oznaka s nazivom serijskog priključka Vaše Arduino pločice. Unutar Arduino IDE programa odaberite taj serijski priključak.

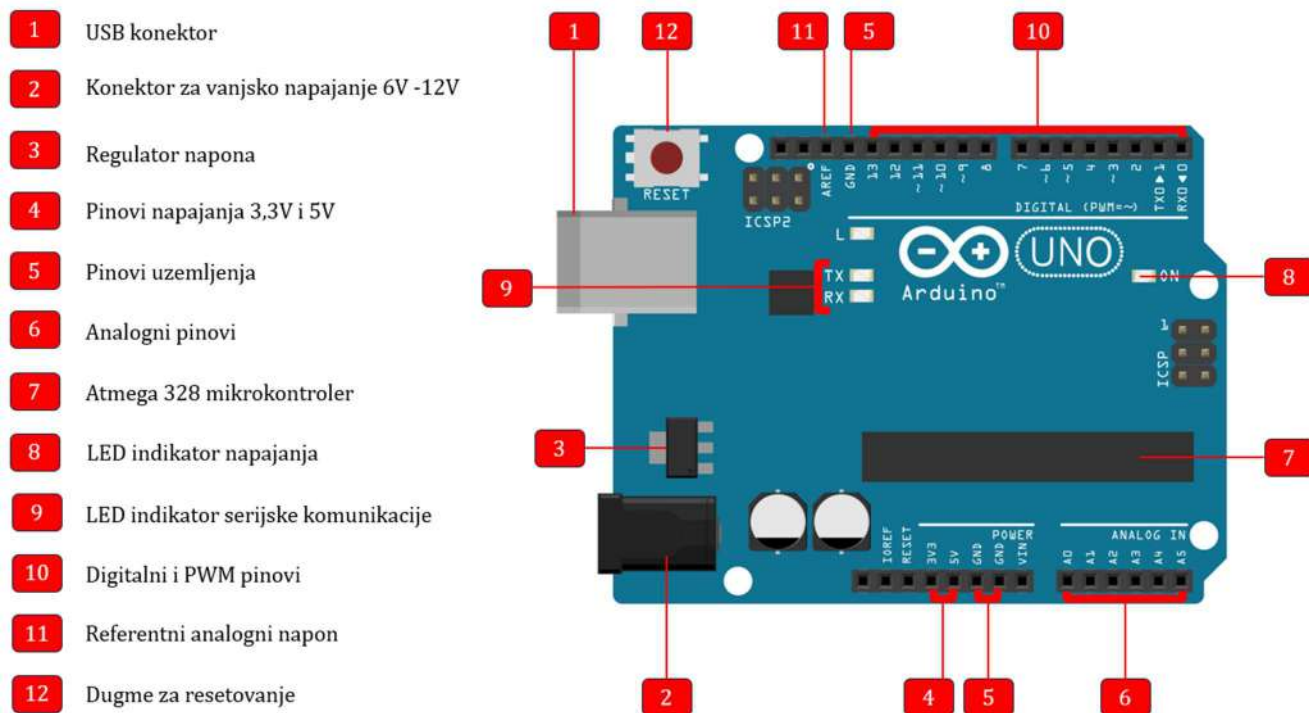
Na kraju je potrebno odabrati programator koji koristimo. Za taj odabir kliknite na **Tools** u meniju potom na **Programmer** te odaberite **Arduino as ISP**.



Sad kad smo sve podesili krenimo sa radom.

Osnovna znanja

Nakon što smo instalirali i upoznali Arduino IDE program vrijeme je da malo bolje upoznamo mikrokontrolere.

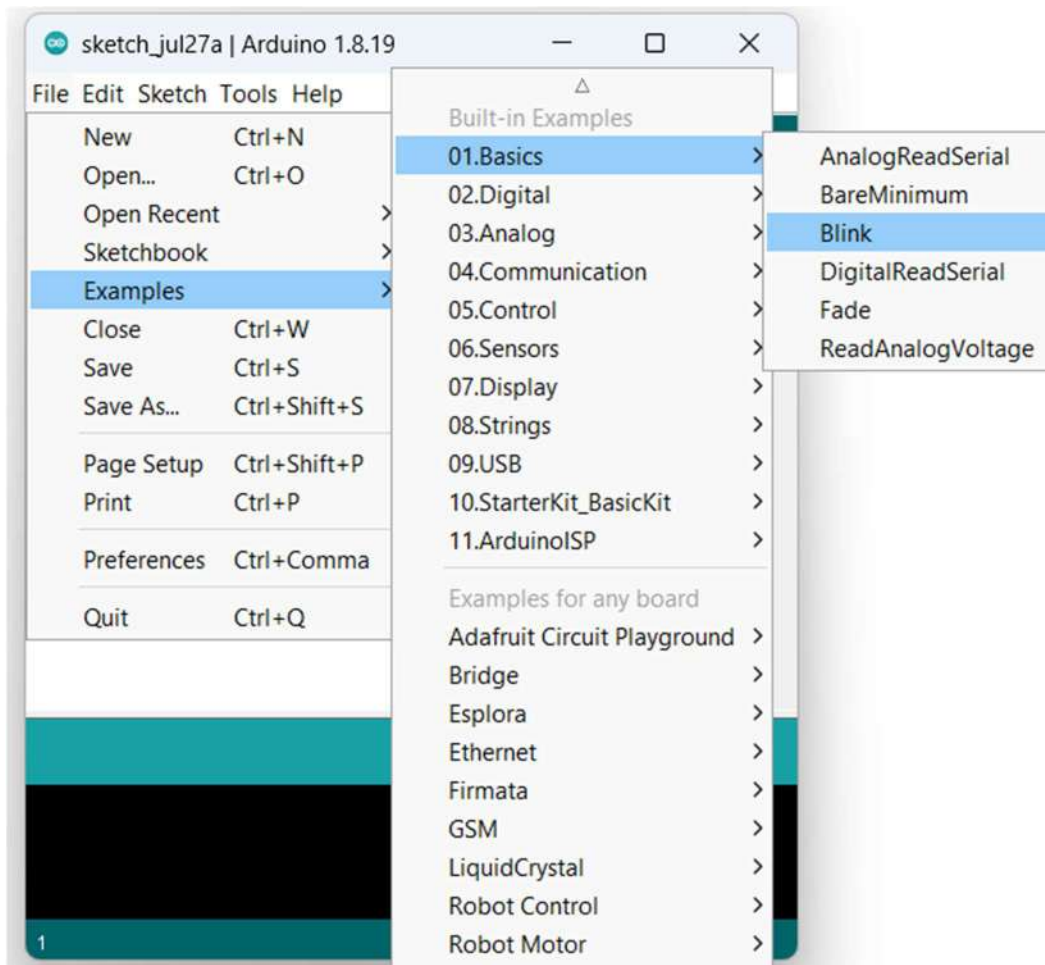


1. USB konektor služi za povezivanje Arduino pločice i računara kao i prijenos programa na Atmega 328 mikrokontroler.
2. Konektor za vanjsko napajanje 6V – 12V – služi za vanjsko napajanje Arduino pločice naponom od 6V do 12V.
3. Regulator napona – regulira jačinu napona ka Arduino pločici.
4. Pinovi napajanja 3,3V i 5V – služe za napajanje komponenti spojenih na pinove naponom od 3,3V ili 5V.
5. Pinovi uzemljenja (Ground – GND). Termin zemlja se u elektronici koristi za provodnik pod nultim potencijalom, on je referenca za sva naponska mjerenja sa potencijalom od 0 volti.
6. Analogni pinovi – služe da signal sa analognih senzora konvertuje u digitalni signal.
7. Atmega 328 mikrokontroler – mini računar. Kontrolira stanje pinova, prima i šalje signale, pohranjuje i izvršava program ...
8. LED indikator napajanja – svijetli kada je Arduino pločica spojena na napajanje.

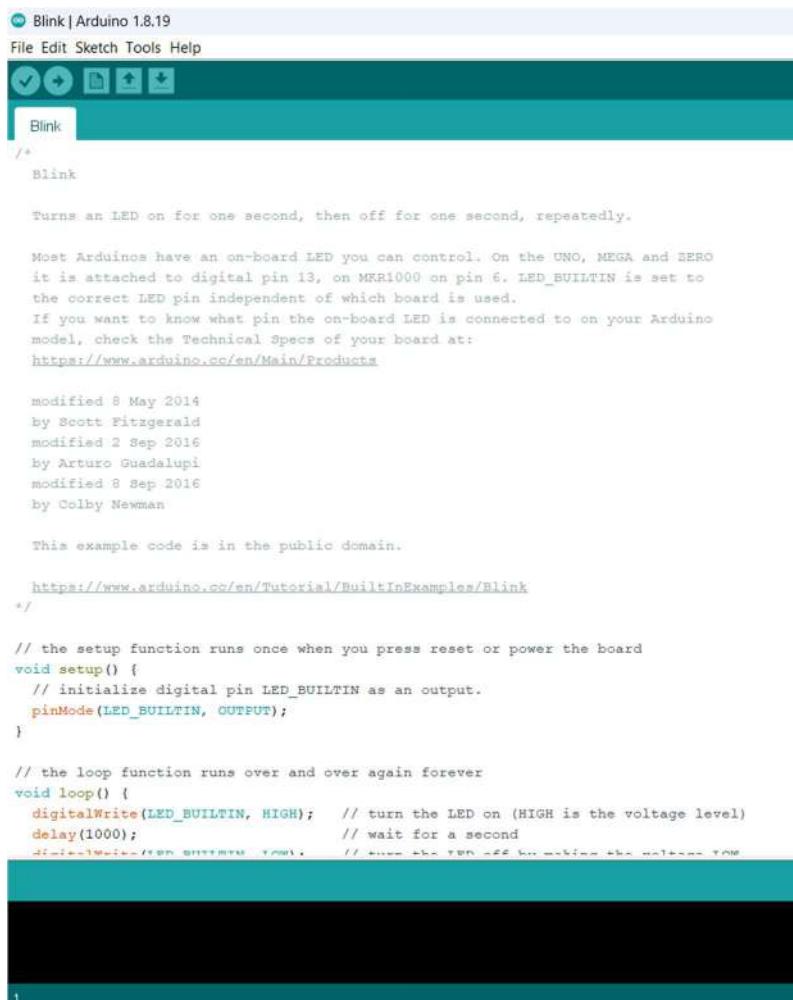
9. LED indikator serijske komunikacije – TX LED treperi kad mikrokontroler šalje podatke, a RX LED kad mikrokontroler prima podatke.
10. Digitalni i PWM pinovi. Digitalni pinovi sa oznakama od 0 – 13 mogu biti korišteni za digitalni Input ili Output.
Neki digitalni pinovi imaju oznaku ~ (tilda), što znači da pinovi sa tom oznakom podržavaju pulsno širinski modulirani izlaz. Pošto Atmega 328 nema DAC (Digital to Analog Converter) logiku, koristi se tehnika PWM (Pulse Width Modulation) za generiranje analognih signala putem modulacije kvadratnog talasa.
11. Referentni analogni napon – (Analog Reference – AREF). Služi kao referenca za usporedbu napona pri analognim mjerenjima. Podrazumijevana vrijednost je 5V za modele koji rade na naponu od 5V i 3,3V za uređaje koji nemaju napajanje od 5V. Dovođenje većeg napona na ovaj pin može oštetiti ili uništiti uređaj. Nije namijenjeno za početnike.
12. Dugme za resetovanje (Reset Button) – služi za ponovno pokretanje koda koji je učitao na Arduino pločicu.

Prvi program

Jedna od velikih prednosti Arduino platforme je što postoji velik broj već gotovih primjera koje možemo iskoristiti za učenje i kreiranje vlastitih uređaja. Kako bi na samom početku vidjeli radi li naša Arduino UNO pločica i kako bi naučili kako učitati program s računara u mikrokontroler iskoristit ćemo jednostavan primjer iz primjera u Arduino IDE programu. Klikom na meni **File** odabirom opcije **Examples** pojavljuju se kategorije dostupnih primjera. Pod kategorijom **Basic** nalazi se program **Blink**.



Otvorimo ga i pogledajmo.

The image shows a screenshot of the Arduino IDE interface. At the top, the window title is "Blink | Arduino 1.8.19". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for opening, saving, and other file operations. The main area of the IDE displays the code for the "Blink" sketch. The code is enclosed in a multi-line comment block that explains the purpose of the sketch: "Turns an LED on for one second, then off for one second, repeatedly." It also provides information about the on-board LED pin (pin 13 for UNO, MEGA, and ZERO) and a link to the Arduino website for more details. The code itself consists of a setup function that initializes the LED pin as an output, and a loop function that turns the LED on for one second, waits for one second, and then turns it off. The code is as follows:

```
/*  
  Blink  
  
  Turns an LED on for one second, then off for one second, repeatedly.  
  
  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO  
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to  
  the correct LED pin independent of which board is used.  
  If you want to know what pin the on-board LED is connected to on your Arduino  
  model, check the Technical Specs of your board at:  
  https://www.arduino.cc/en/Main/Products  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
  modified 2 Sep 2016  
  by Arturo Guadalupi  
  modified 8 Sep 2016  
  by Colby Newman  
  
  This example code is in the public domain.  
  
  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
}
```

Cilj ovog programa je uključivati i isključivati svjetleću (LED) diodu koja se nalazi na samoj Arduino UNO pločici. Naime, kako bi mi kao korisnici odmah nakon instalacije mogli isprobati njezinu funkcionalnost na samu je pločicu, na izvod (pin) broj 13 spojena jedna svjetleća diode. Kako biste učitali napisani program u mikrokontroler možete iskoristiti drugu ikonu u traci s alatima – **Upload** ikonu. Učitajmo program u mikrokontroler i pogledajmo šta se događa.

Svjetleća diode se naizmjenično uključuje i isključuje – **blinka**.

Proučimo program da vidimo kako je to postignuto.

```
/*  
Blink  
  
Turns on an LED for one second, then off for one second, repeatedly.  
  
Most Arduinos have an on-board LED you can control. On the UNO and  
Leonardo, it is attached to digital pin 13. If you're unsure what  
pin the on-board LED is connected to on your Arduino model, check  
the documentation at http://arduino.cc  
  
This example code is in the public domain.  
  
Modified 8 May 2014  
By Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage level)  
  delay(1000);                // wait for a second  
  digitalWrite(13, LOW);      // turn the LED off by making the voltage LOW  
  delay(1000);                // wait for a second  
}
```

Svaki od Arduino programa (engleski naziv sketch) sastoji se od tri osnovna dijela. U prvom dijelu programa govorimo mikrokontroleru na koji smo pin šta spojili odnosno definiramo varijable, u drugom dijelu programa govorimo na koji način koristimo pojedine pinove te pišemo onaj dio koda koji se izvodi samo jednom na početku, pri pokretanju mikrokontrolera, a u trećem dijelu programa pišemo šta zapravo mikrokontroler treba raditi.

```
/*
```

```
Blink
```

```
Turns on an LED for one second, then off for one second, repeatedly.
```

```
Most Arduinos have an on-board LED you can control. On the UNO and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://arduino.cc
```

```
This example code is in the public domain.
```

```
Modified 8 May 2014
```

```
By Scott Fitzgerald
```

```
*/
```

U programu je korisno opisati šta se kojim od dijelova programa želi postići, šta su argumenti, objasniti deklaraciju varijabli i sl. Takvi se pomoćni opisi nazivaju komentari.

Komentar je tekst koji započinje s dvostrukom kosom crtom //, a završava krajem reda. Može biti napisan u istom redu s naredbom ili u zasebnom redu. Prilikom prevođenja programa u mašinski jezik komentari se preskaču.

Uz navedeni oblik komentara programski jezik C++ podržava i komentare unutar para znakova /* */. Takvi komentari započinju /* (kosa crta i zvjezdica) , a završavaju */ (zvjezdica i kosa crta). Kraj reda ne znači podrazumijevani završetak komentara, pa se ovakvi komentari mogu protezati na nekoliko redova programa, a da se znak za komentar ne mora pojavljivati u svakom redu.

Unutar drugog dijela – kojeg nazivamo setup dijelom rekli smo mikrokontroleru da je pin broj 13, na kojem je spojena svjetleća diode izlaznog tipa. To znači da mikrokontroler upravlja uključivanjem i isključivanjem svjetleće diode.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
```


Kao što vidimo za postavljanje načina rad pojedinog pina koristimo naredbu **pinMode(nazivpina, načinrada)**. Ova naredba ima dva argumenta (argumenti su podaci koje upisujemo u zagrade kod naredbi) – **nazivpina** definira na koji pin se odnosi naredba a **načinrada** definira jeli taj pin ulaznog ili izlaznog tipa. Kako mi želimo upravljati svjetlećom diodom onda postavljamo pin u izlazni način rada koristeći ključnu riječ **OUTPUT**.

U trećem dijelu programa – takozvanom **loop** dijelu koji se stalno ponavlja dok mikrokontroler radi napisali smo dio koda koji uključuje diodu, nakon toga čeka jednu sekundu, pa isključuje diodu te ponovo čeka jednu sekundu.

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);           // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(13, LOW);          // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

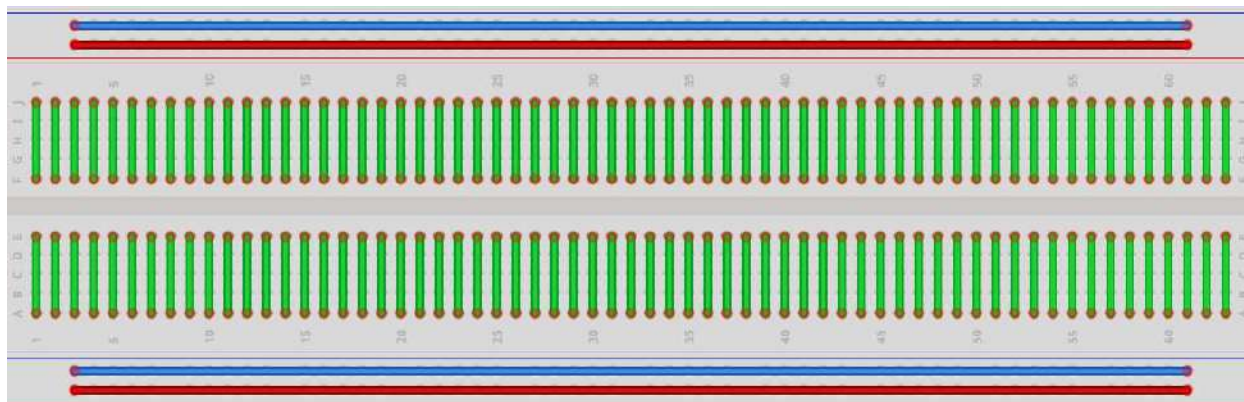
Za uključivanje i isključivanje svjetleće diode koristimo naredbu **digitalWrite**. Ova naredba također ima dva argumenta **digitalWrite(nazivpina, stanje)**. Argument **nazivpina** definira na koji pin se odnosi naredba a **stanje** definira postavljamo li pin u visoko ili nisko stanje. Kada se izvod nalazi u stanju visoko svjetleća dioda je uključena, a kada je stanje nisko svjetleća je dioda isključena.

Naredba **delay(vrijeme)** koristi se za zaustavljanje rada odnosno čekanje određenog vremenskog perioda. Argument **vrijeme** govori nam koliko dugo treba zaustaviti rad i izražava se u milisekundama. Naredba **delay(1000)** će zaustaviti rad na 1000 milisekundi tj. jednu sekundu.

Budući da se dio programa koji uključuje i isključuje svjetleću diodu nalazi unutar **loop** dijela programa koji se stalno ponavlja dobijamo efekt **blinkanja** svjetleće diode.

Eksperimentalna pločica (breadboard)

Eksperimentalna pločica služi za spajanje elektronskih komponenti i modula, na taj način omogućava jednostavnu izradu prototipova uređaja, njihovo testiranje i razvijanje. Pogodna je za brzo sklapanje uređaja koji se sastoje od manjeg broja komponenti. Izrađuje se u različitim dimenzijama i oblicima, a na bazi istog principa.



Eksperimentalna pločica sastoji se od plastičnog kućišta na čijoj gornjoj strani se nalazi mnoštvo rupica namijenjenih za umetanje nožica različitih komponenti, kao i provodnika za njihovo povezivanje (tzv. jumper). Razmaci između dvije susjedne rupice su standardizovani i iznose 2,5 mm. Razmak je usklađen s razmakom između nožica kod integralnih kola i elektronskih komponenti.

Rupice su u unutrašnjosti pločice međusobno povezane trakama od materijala koji je provodnik, prema određenom pravilu. S obje strane pločice imamo područje koje se sastoji od dva niza rupica međusobno spojenih na način da formiraju uzdužne staze koje koristimo za napajanje. Često su uz svaku od ovih staza nacrtane dvije linije, crvena i plava, zajedno sa simbolima + (plus) i – (minus). Na većini pločica rupice unutar jedne staze nisu spojene po cijeloj dužini, nego su prekinute na sredini što je obično prikazano prekidom plave i crvene linije.

U sredini pločice nalazi se rascjep koji pločicu dijeli na dva dijela. Njegova širina je standardizovana i omogućava umetanje integralnih kola. Rupice sa svake strane rascjepa su međusobno povezane po vertikali čineći grupe od po 5 povezanih rupica.

Kada se utaknu pinovi/nožice elektronskih komponenti ili kratkospojnici (jumper) u rupice iz iste grupe ostvaruje se električni kontakt među njima. Često su na pločici redovi označeni brojevima (1-30), a kolone slovima (a-j).

Kratkospojnici (Jumper Wires)

Kratkospojnici (Jumper Wires) su kratke izolirane žice koje služe za povezivanje komponenti na eksperimentalnoj pločici, za međusobno povezivanje komponenti ili njihovo povezivanje sa Arduino pločicama putem pinova. Međusobno se, osim po boji i dužini, razlikuju i po konektorima na kraju.

Postoje sljedeće kombinacije konektora na kratkospojniku:

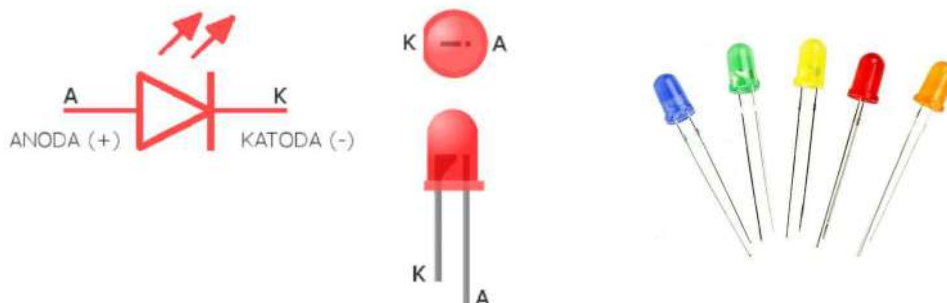
- MUŠKO-MUŠKI (eng. MALE TO MALE),
- MUŠKO-ŽENSKI (eng. MALE TO FEMALE) i
- ŽENSKO-ŽENSKI (eng. FEMALE TO FEMALE).








LED (eng. Light Emitting Diode)

LED (eng. Light Emitting Diode) u prijevodu “dioda koja emitira svjetlost” ili češće u našem jeziku “svjetleća dioda” je poluprovodnički elektronički element koji električni signal pretvara u optički (svjetlost). LE dioda propušta struju samo u jednom smjeru (od anode prema katodi) pri čemu svjetli, dok u drugom pruža otpor.

LE dioda ima dvije elektrode, anodu i katodu. Anoda je pozitivna elektroda dok je katoda negativna. Ako se LE dioda pažljivo pogleda može se vidjeti da je jedna nožica duža od druge. Duža nožica je uvijek anoda dok je kraća katoda. Osim toga, obod stakla na jednom dijelu nije okrugao, nego je ravan što označava s koje strane se nalazi katoda.



Boja LED	Napon (U)	Struja (I)
	1,8 V	20 mA
	2,1 V	20 mA
	2,2 V	25 mA
	4,5 V	30 mA
	3,5 V	35 mA

Zavisno od materijala od kojeg su izrađene i talasne dužine, diode emitiraju različitu svjetlost. LE diode prema boji priključujemo na predviđeni napon. Svaka dioda ima svoj tzv. napon praga koji mora biti osiguran kako bi dioda ispravno radila. Kada je napon niži od predviđenog LE dioda neće svjetliti, a kada je viši od predviđenog dioda će pregorjeti.

Zaštita LE diode se postiže preko Omovog zakona, dodavanjem otpornika u strujni krug kako kroz diodu ne bi protekla prevelika struja. Katoda se spaja na uzemljenje Arduino pločice (0 V ili GND), a anoda preko otpornika na digitalni pin kojim se upravlja diodom.

Otpornik (eng. Resistor)

Otpornici su pasivne elektroničke komponente koje pružaju otpor proticanju električne struje, stvarajući pri tom pad napona između priključaka. Drugim riječima, oni apsorbiraju dio električne energije koju pretvaraju u toplotu. Možemo ih pronaći u svim električnim mrežama i elektroničkim sklopovima gdje se koriste za ostvarivanje željenog odnosa struje i napona, ograničenje struje, smanjenje napona i sl.

Najvažnija karakteristika otpornika je električni otpor. Osnovna mjerna jedinica za otpor je 1 Ω (om). U praksi se koriste i manje jedinice, miliom (1 m Ω =10⁻³ Ω), kao i veće jedinice, kiloom (1 k Ω =10³ Ω) i megaom (1 M Ω =10⁶ Ω).

Osnovni simbol otpornika, prema IEC standardu, je prazan pravougaonik koji ima jedan ulaz i jedan izlaz. Otpornik prema ANSI standardu ima cik-cak linije koje isto imaju jedan ulaz i jedan izlaz.

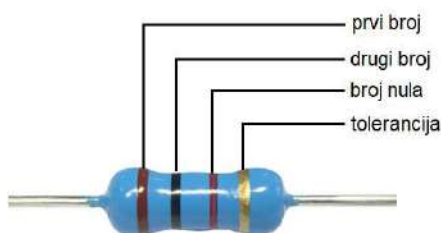


Fiksni otpornici su najčešće primjenjivani otpornici zbog jednostavnosti očitavanja vrijednosti otpora i tolerancije. Fiksni otpornici u strujnom krugu imaju stalnu vrijednost.



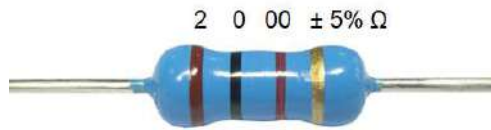
Vrijednost otpornika se predstavlja brojem ili prstenovima određene boje po obodu valjkastog kućišta. Svaka boja ima drugu vrijednost što je prikazano u tabeli.

Pri očitavanju vrijednosti počinjemo od prvog prstena, koji je suprotan od prstena tolerancije (sive ili zlatne boje). Prvi prsten predstavlja prvi broj, drugi prsten drugi broj, treći prsten broj nula i posljednji prsten toleranciju.



Koristeći tabelu lahko možemo očitati vrijednost otpornika prikazanog na slici.

Crna	0	0		
Smeđa	1	1	0	1 %
Crvena	2	2	00	2 %
Narandžasta	3	3	000	
Žuta	4	4	0 000	
Zelena	5	5	00 000	
Plava	6	6	000 000	
Ljubičasta	7	7	0 000 000	
Siva	8	8	00 000 000	
Bijela	9	9	000 000 000	
Zlatna			0,1	5 %
Srebrna			0,01	10 %
Bez boje				20 %

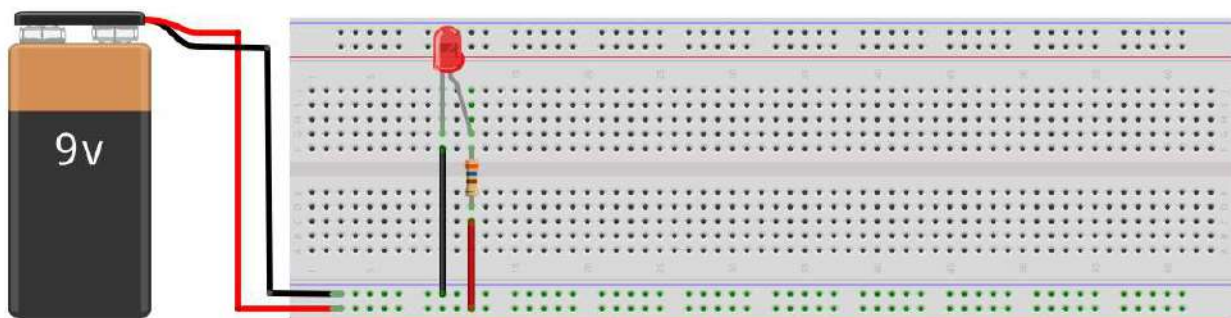


Crvena boja na prvom prstenu predstavlja prvi broj (2), crna boja na drugom prstenu predstavlja drugi broj (0), crvena boja na trećem prstenu predstavlja broj nula (00) i zlatna boja na četvrtom prstenu predstavlja toleranciju vrijednosti otpornika ($\pm 5\%$).

Očitana vrijednost otpornika je $2000\ \Omega$ ili $2\ \text{k}\Omega$, uz toleranciju vrijednosti od $\pm 5\%$.

Spajanje LE diode i izračunavanje vrijednosti otpornika

LE diodu spajamo u strujni krug tako što katodu LE diode spajamo na (-) minus pol baterije, a anodu preko otpornika na (+) pol baterije. Svaka dioda ima svoj tzv. napon praga koji mora biti osiguran kako bi dioda ispravno radila. Kada je napon niži od predviđenog LE dioda neće svjetliti, a kada je viši od predviđenog dioda će pregorjeti. Dakle, kako je napon od $9\ \text{V}$ prevelik za ispravan rad diode u strujni krug je potrebno spojiti otpornik odgovarajuće vrijednosti kako bi dioda ispravno radila.



Izračunavanje potrebnog otpora se vrši putem Omovog zakona, koji glasi $I=U/R$. Iz ove formule dobijamo da $R=U/I$.

Vrijednost otpornika (R) se izračunava tako što se od priključnog napona baterije (U_b) oduzme pad napona diode (U_d), a dobijena razlika podijeli se sa strujom diode I .

U ovom slučaju koristit ćemo formulu gdje je: $R = \frac{U_b - U_d}{I}$

- R-otpor,
- U_b -napon baterije, za koji znamo da je 9 V,
- U_d -napon diode za koji iz tabele znamo da je 1,8 V i
- I-jačina struje za koju iz tabele znamo da je 20 mA.

$$R = \frac{9 \text{ V} - 1,8 \text{ V}}{0,02 \text{ A}} \quad R = 360 \Omega$$

Lekcija 1

Treptajuća LE dioda

Zadatak

Spojimo jednu LE diodu na pin broj 5 Arduina i napišimo program koji će upravljati LE diodom tako da ona trepti, neka bude uključena 1000 milisekundi, pa isključena 1000 milisekundi i neka se to ponavlja.

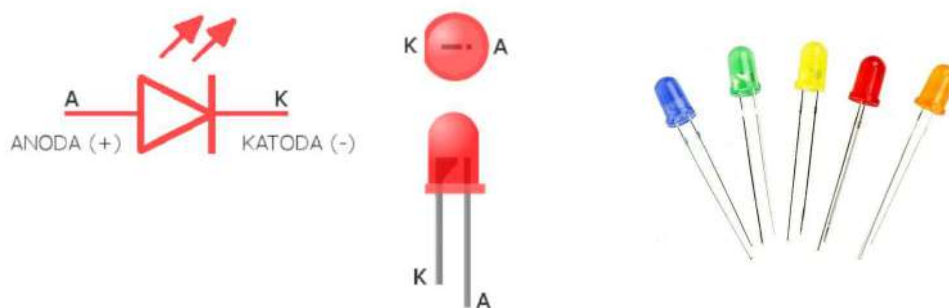
Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*LE dioda,
- 1*Otpornik 220 Ω ,
- 1*Eksperimentalna pločica,
- 2*Kratkospojničke žice (muško-muške).

Šta je LE dioda?

LE dioda je poluprovodnički elektronički element koji električni signal pretvara u optički (svjetlost). LE dioda propušta struju samo u jednom smjeru (od anode prema katodi) pri čemu svjetli, dok u drugom pruža otpor.

LE dioda ima dvije elektrode, anodu i katodu. Anoda je pozitivna elektroda dok je katoda negativna. Ako se LE dioda pažljivo pogleda može se vidjeti da je jedna nožica duža od druge. Duža nožica je uvijek anoda dok je kraća katoda. Osim toga, obod stakla na jednom dijelu nije okrugao, nego je ravan što označava s koje strane se nalazi katoda.



Zavisno od materijala od kojeg su izrađene i talasne dužine, diode emitiraju različitu svjetlost. LE diode prema boji priključujemo na predviđeni napon. Svaka dioda ima svoj tzv. napon praga koji mora biti osiguran kako bi dioda ispravno radila. Kada je napon niži od predviđenog LE dioda neće svjetliti, a kada je viši od predviđenog dioda će pregorjeti.

Zaštita LE diode se postiže preko Omovog zakona, dodavanjem otpornika u strujni krug kako kroz diodu ne bi protekla prevelika struja. Katoda se spaja na uzemljenje Arduino pločice (0 V ili GND), a anoda preko otpornika na digitalni pin kojim se upravlja diodom.

Šta je otpornik?

Otpornik je pasivna elektronička komponenta koja pruža otpor proticanju električne struje, stvarajući pri tom pad napona između priključaka. Drugim riječima, otpornik apsorbira dio električne energije koju pretvara u toplotu. U ovoj lekciji koristimo otpornik sa stalnom vrijednosti. Vrijednost otpornika se predstavlja prstenovima određene boje po obodu valjkastog kućišta.



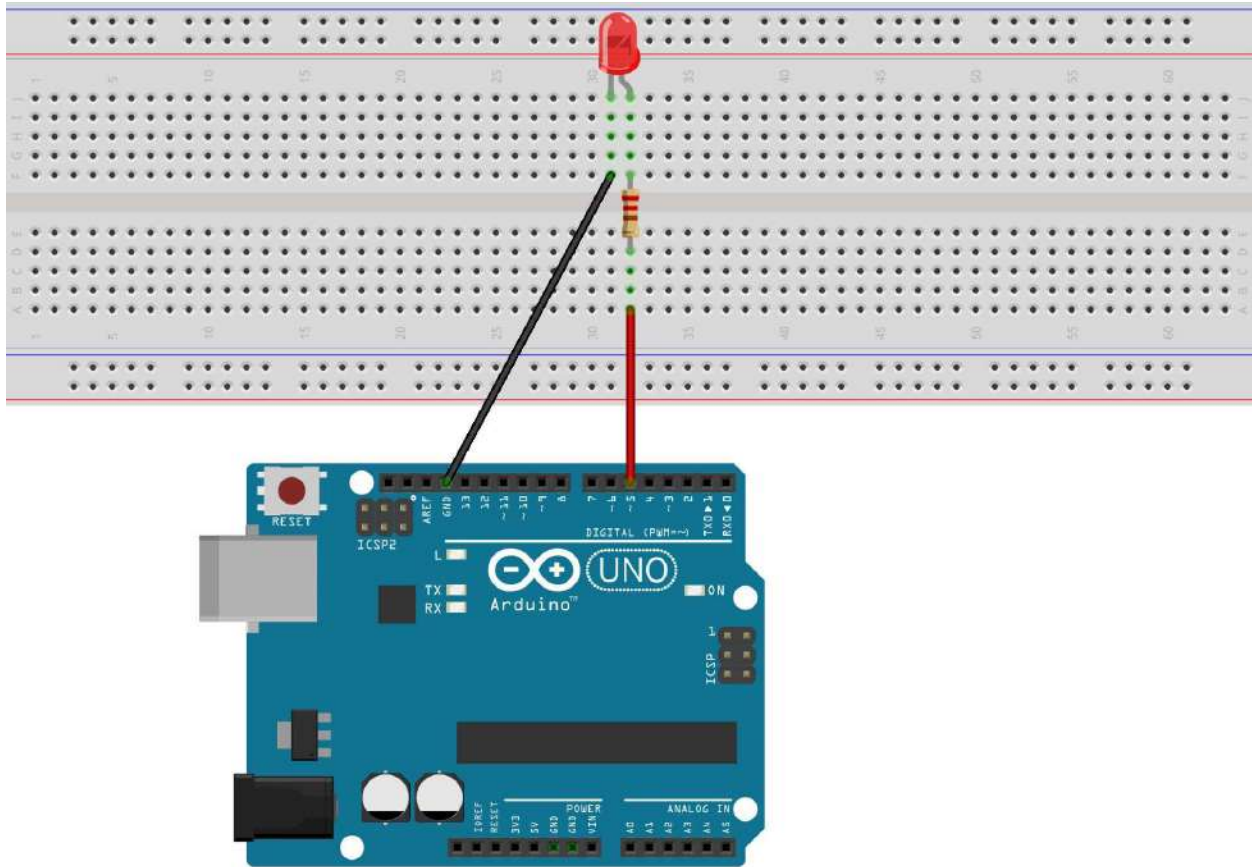
Najvažnija karakteristika otpornika je električni otpor (R). Osnovna mjerna jedinica za otpor je 1 Ω (om). Budući da je struja potrebna da se uključi LE dioda je 5~20 mA, a izlani napon Arduinoa 5 V, otpor dobijamo po formuli:

$$R=U/I = 5V/(5 \sim 20 \text{ mA}) = 250 \Omega \sim 1 \text{ k}\Omega$$

Budući da je LE dioda sam otpornik, ovdje koristimo otpor od 220 oma.

Postupak

1. Izgradite sklop



fritzing

2. Program

```
/*  
Treptajuća LE dioda  
STEM Lab  
*/  
int ledPin=5; // Definiranje digitalnog pina 5 kao pina  
              // za upravljanje LE diodom  
  
void setup()  
{  
  pinMode(ledPin,OUTPUT); // Postavite digitalni način rada s pina 5, IZLAZ:  
                          // izlazni način  
}  
  
void loop()  
{  
  digitalWrite(ledPin,HIGH); // ledPin (pin 5) postavite u stanje UKLJUČI  
  delay(1000);              // Postavite vrijeme čekanja, 1000 milisekundi  
                          // (1 sekunda)  
  
  digitalWrite(ledPin,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI  
  delay(1000);              // Postavite vrijeme čekanja, 1000 milisekundi  
                          // (1 sekunda)  
}
```

3. Prenesite program na Arduino UNO pločicu.

Lekcija 2

Toplo-hladno

Zadatak

Spojimo plavu LE diodu na pin broj 5 Arduina a crvenu LE diodu na pin broj 10 Arduina. Napišimo program koji će upravljati LE diodama tako da se crvena i plava LE dioda naizmjenično uključuju svakih 1000 milisekundi.

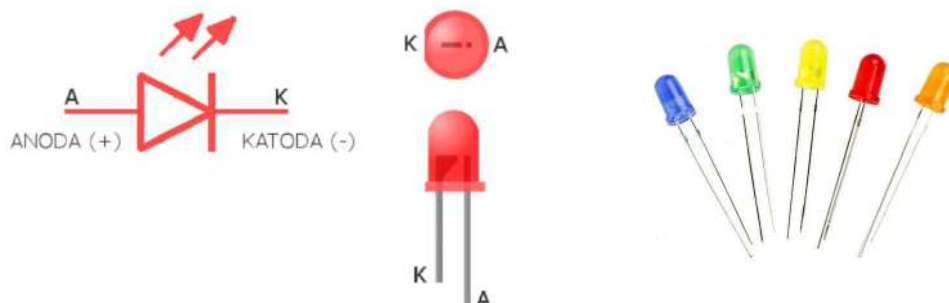
Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*LE dioda (crvena)
- 1*LE dioda (plava)
- 2*Otpornik 220 Ω ,
- 1*Eksperimentalna pločica,
- 5*Kratkospojničke žice (muško-muške).

Šta je LE dioda?

LE dioda je poluprovodnički elektronički element koji električni signal pretvara u optički (svjetlost). LE dioda propušta struju samo u jednom smjeru (od anode prema katodi) pri čemu svjetli, dok u drugom pruža otpor.

LE dioda ima dvije elektrode, anodu i katodu. Anoda je pozitivna elektroda dok je katoda negativna. Ako se LE dioda pažljivo pogleda može se vidjeti da je jedna nožica duža od druge. Duža nožica je uvijek anoda dok je kraća katoda. Osim toga, obod stakla na jednom dijelu nije okrugao, nego je ravan što označava s koje strane se nalazi katoda.



Zavisno od materijala od kojeg su izrađene i talasne dužine, diode emitiraju različitu svjetlost. LE diode prema boji priključujemo na predviđeni napon. Svaka dioda ima svoj tzv. napon praga koji mora biti osiguran kako bi dioda ispravno radila.

Kada je napon niži od predviđenog LE dioda neće svjetliti, a kada je viši od predviđenog dioda će pregorjeti.

Zaštita LE diode se postiže preko Omovog zakona, dodavanjem otpornika u strujni krug kako kroz diodu ne bi protekla prevelika struja. Katoda se spaja na uzemljenje Arduino pločice (0 V ili GND), a anoda preko otpornika na digitalni pin kojim se upravlja diodom.

Šta je otpornik?

Otpornik je pasivna elektronička komponenta koja pruža otpor proticanju električne struje, stvarajući pri tom pad napona između priključaka. Drugim riječima, otpornik apsorbira dio električne energije koju pretvara u toplotu. U ovoj lekciji koristimo otpornik sa stalnom vrijednosti. Vrijednost otpornika se predstavlja prstenovima određene boje po obodu valjkastog kućišta.



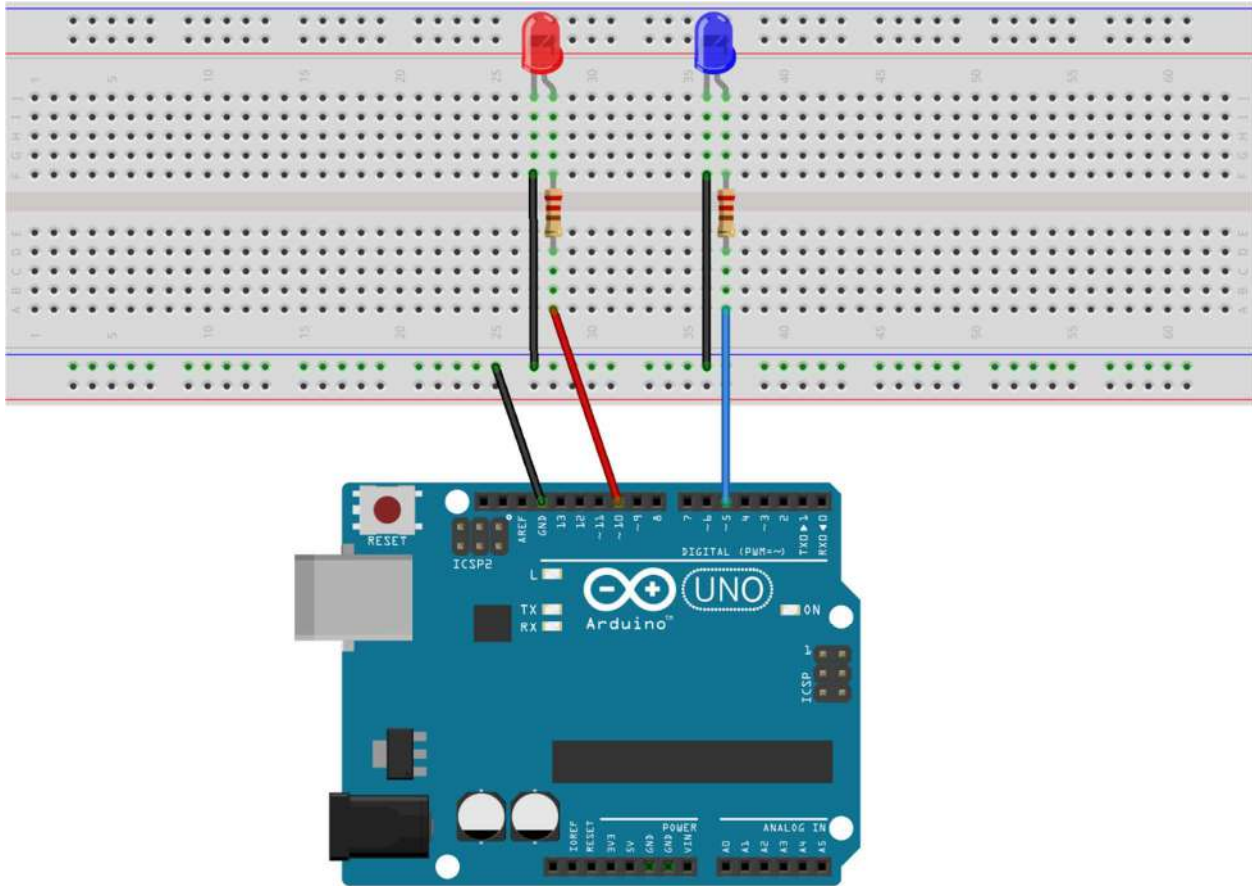
Najvažnija karakteristika otpornika je električni otpor (R). Osnovna mjerna jedinica za otpor je 1 Ω (om). Budući da je struja potrebna da se uključi LE dioda je 5~20 mA, a izlani napon Arduinoa 5 V, otpor dobijamo po formuli:

$$R=U/I = 5V/(5 \sim 20 \text{ mA}) = 250 \Omega \sim 1 \text{ k}\Omega$$

Budući da je LE dioda sam otpornik, ovdje koristimo otpor od 220 oma.

Postupak

4. Izgradite sklop



fritzing

5. Program

```
/*  
Toplo-hladno  
STEM Lab  
*/  
int ledPlava=5; // Definiranje digitalnog pina 5 kao pina za  
                // upravljanje plavom LE diodom  
int ledCrvena=10; // Definiranje digitalnog pina 10 kao pina za  
                 // upravljanje crvenom LE diodom  
void setup ()  
{  
  pinMode(ledPlava, OUTPUT); // Postavite digitalni način rada s pina 5, IZLAZ:  
                             // izlazni način  
  pinMode(ledCrvena, OUTPUT); // Postavite digitalni način rada s pina 10, IZLAZ:  
                              // izlazni način  
}  
void loop()  
{  
  digitalWrite(ledPlava, HIGH); // ledPlava (pin 5) postavite u stanje  
                                // UKLJUČI-uključite plavu LE diodu  
  digitalWrite(ledCrvena, LOW); // ledCrvena (pin 10) postavite u stanje  
                                // ISKLJUČI-isključite crvenu LE diodu  
  delay(1000); // Postavite vrijeme čekanja, 1000 milisekundi  
              // (1 sekunda)  
  
  digitalWrite(ledPlava, LOW); // ledPlava (pin 5) postavite u stanje  
                                // ISKLJUČI-isključite plavu LE diodu  
  digitalWrite(ledCrvena, HIGH); // ledCrvena (pin 10) postavite u stanje  
                                 // UKLJUČI-uključite crvenu LE diodu  
  delay(1000); // Postavite vrijeme čekanja, 1000 milisekundi  
              // (1 sekunda)  
}
```

6. Prenesite program na Arduino UNO pločicu.

Lekcija 3

Semafor

Zadatak

Izradimo model semafora za vozila. Na Arduino pločicu spojimo crvenu LE diodu na pin broj 7, žutu LE diodu na pin broj 6 i zelenu LE diodu na pin broj 5.

Napišimo program da semafor radi na sljedeći način: Na početku je uključena crvena LE dioda u trajanju od 5 sekundi, zatim su uključene zajedno crvena i žuta LE dioda 2 sekunde, pa zelena LE dioda 5 sekundi.

Na kraju je uključena žuta LE dioda 2 sekunde. Ciklus se stalno ponavlja.



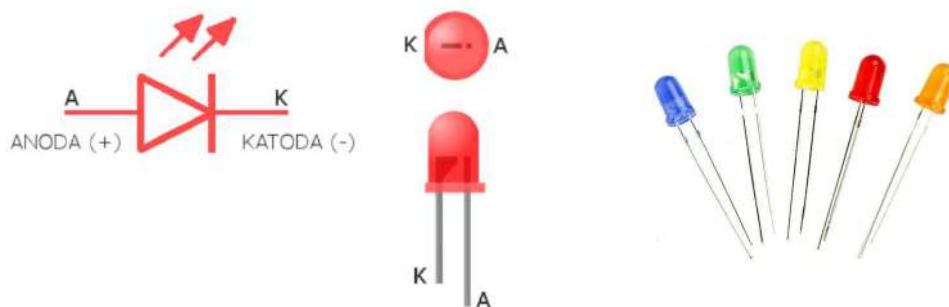
Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*LE dioda (crvena)
- 1*LE dioda (žuta)
- 1*LE diode (zelena)
- 3*Otpornik 220 Ω ,
- 1*Eksperimentalna pločica,
- 7*Kratkospojničke žice (muško-muške).

Šta je LE dioda?

LE dioda je poluprovodnički elektronički element koji električni signal pretvara u optički (svjetlost). LE dioda propušta struju samo u jednom smjeru (od anode prema katodi) pri čemu svjetli, dok u drugom pruža otpor.

LE dioda ima dvije elektrode, anodu i katodu. Anoda je pozitivna elektroda dok je katoda negativna. Ako se LE dioda pažljivo pogleda može se vidjeti da je jedna nožica duža od druge. Duža nožica je uvijek anoda dok je kraća katoda. Osim toga, obod stakla na jednom dijelu nije okrugao, nego je ravan što označava s koje strane se nalazi katoda.



Zavisno od materijala od kojeg su izrađene i talasne dužine, diode emitiraju različitu svjetlost. LE diode prema boji priključujemo na predviđeni napon. Svaka dioda ima svoj tzv. napon praga koji mora biti osiguran kako bi dioda ispravno radila. Kada je napon niži od predviđenog LE dioda neće svjetliti, a kada je viši od predviđenog dioda će pregorjeti.

Zaštita LE diode se postiže preko Omovog zakona, dodavanjem otpornika u strujni krug kako kroz diodu ne bi protekla prevelika struja. Katoda se spaja na uzemljenje Arduino pločice (0 V ili GND), a anoda preko otpornika na digitalni pin kojim se upravlja diodom.

Šta je otpornik?

Otpornik je pasivna elektronička komponenta koja pruža otpor proticanju električne struje, stvarajući pri tom pad napona između priključaka. Drugim riječima, otpornik apsorbira dio električne energije koju pretvara u toplotu. U ovoj lekciji koristimo otpornik sa stalnom vrijednosti. Vrijednost otpornika se predstavlja prstenovima određene boje po obodu valjkastog kućišta.



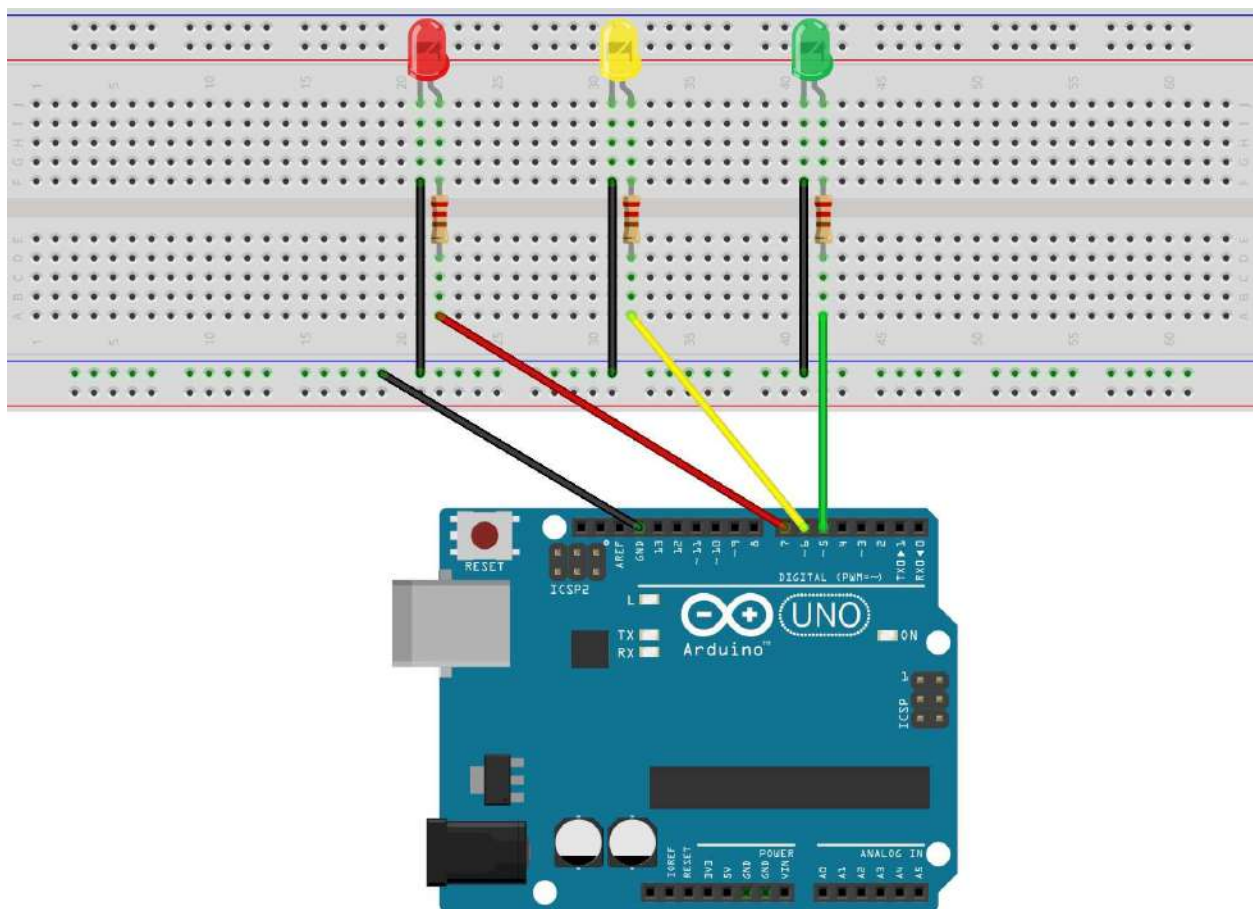
Najvažnija karakteristika otpornika je električni otpor (R). Osnovna mjerna jedinica za otpor je 1 Ω (om). Budući da je struja potrebna da se uključi LE dioda je 5~20 mA, a izlani napon Arduina 5 V, otpor dobijamo po formuli:

$$R=U/I = 5V/(5 \sim 20 \text{ mA}) = 250 \Omega \sim 1 \text{ k}\Omega$$

Budući da je LE dioda sam otpornik, ovdje koristimo otpor od 220 oma.

Postupak

7. Izgradite sklop



fritzing

8. Program

```
/*  
Semafor  
STEM Lab  
*/  
int ledPin_Red=7; // Definiranje digitalnog pina 7 kao pina za  
 // upravljanje crvenom LE diodom  
int ledPin_Yellow=6; // Definiranje digitalnog pina 6 kao pina za  
 // upravljanje žutom LE diodom  
int ledPin_Green=5; // Definiranje digitalnog pina 5 kao pina za  
 // upravljanje zelenom LE diodom  
  
void setup()  
{  
  pinMode(ledPin_Red,OUTPUT); // Postavite digitalni način rada s pina 7, IZLAZ:  
 // izlazni način  
  pinMode(ledPin_Yellow,OUTPUT); // Postavite digitalni način rada s pina 6, IZLAZ:  
 // izlazni način  
  pinMode(ledPin_Green,OUTPUT); // Postavite digitalni način rada s pina 5, IZLAZ:  
 // izlazni način  
}  
void loop() // Stalno se izvršava  
{  
  digitalWrite(ledPin_Red,HIGH); // ledPin (pin 7) postavite u stanje UKLJUČI  
 // (uključite crvenu LE diodu)  
  digitalWrite(ledPin_Yellow,LOW); // ledPin (pin 6) postavite u stanje ISKLJUČI  
 // (isključite žutu LE diodu)  
  digitalWrite(ledPin_Green,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI  
 // (isključite zelenu LE diodu)  
  delay(5000); // Postavite vrijeme čekanja, 5000 milisekundi  
 // (5 sekundi)
```

```

digitalWrite(ledPin_Red,HIGH); // ledPin (pin 7) postavite u stanje UKLJUČI
                                (uključite crvenu LE diodu)
digitalWrite(ledPin_Yellow,HIGH); // ledPin (pin 6) postavite u stanje UKLJUČI
                                    (uključite žutu LE diodu)
digitalWrite(ledPin_Green,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI
                                    (isključite zelenu LE diodu)
delay(2000); // Postavite vrijeme čekanja, 2000 milisekundi
              (2 sekunde)

digitalWrite(ledPin_Red,LOW); // ledPin (pin 7) postavite u stanje ISKLJUČI
                                (isključite crvenu LE diodu)
digitalWrite(ledPin_Yellow,LOW); // ledPin (pin 6) postavite u stanje ISKLJUČI
                                    (isključite žutu LE diodu)
digitalWrite(ledPin_Green,HIGH); // ledPin (pin 5) postavite u stanje UKLJUČI
                                    (uključite zelenu LE diodu)
delay(5000); // Postavite vrijeme čekanja, 5000 milisekundi
              (5 sekundi)

digitalWrite(ledPin_Red,LOW); // ledPin (pin 7) postavite u stanje ISKLJUČI
                                (isključite crvenu LE diodu)
digitalWrite(ledPin_Yellow,HIGH); // ledPin (pin 6) postavite u stanje UKLJUČI
                                    (uključite žutu LE diodu)
digitalWrite(ledPin_Green,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI
                                    (isključite zelenu LE diodu)
delay(2000); // Postavite vrijeme čekanja, 2000 milisekundi
              (2 sekunde)
}

```

9. Prenesite program na Arduino UNO pločicu.

Lekcija 4

Stalna vrijednost

Zadatak

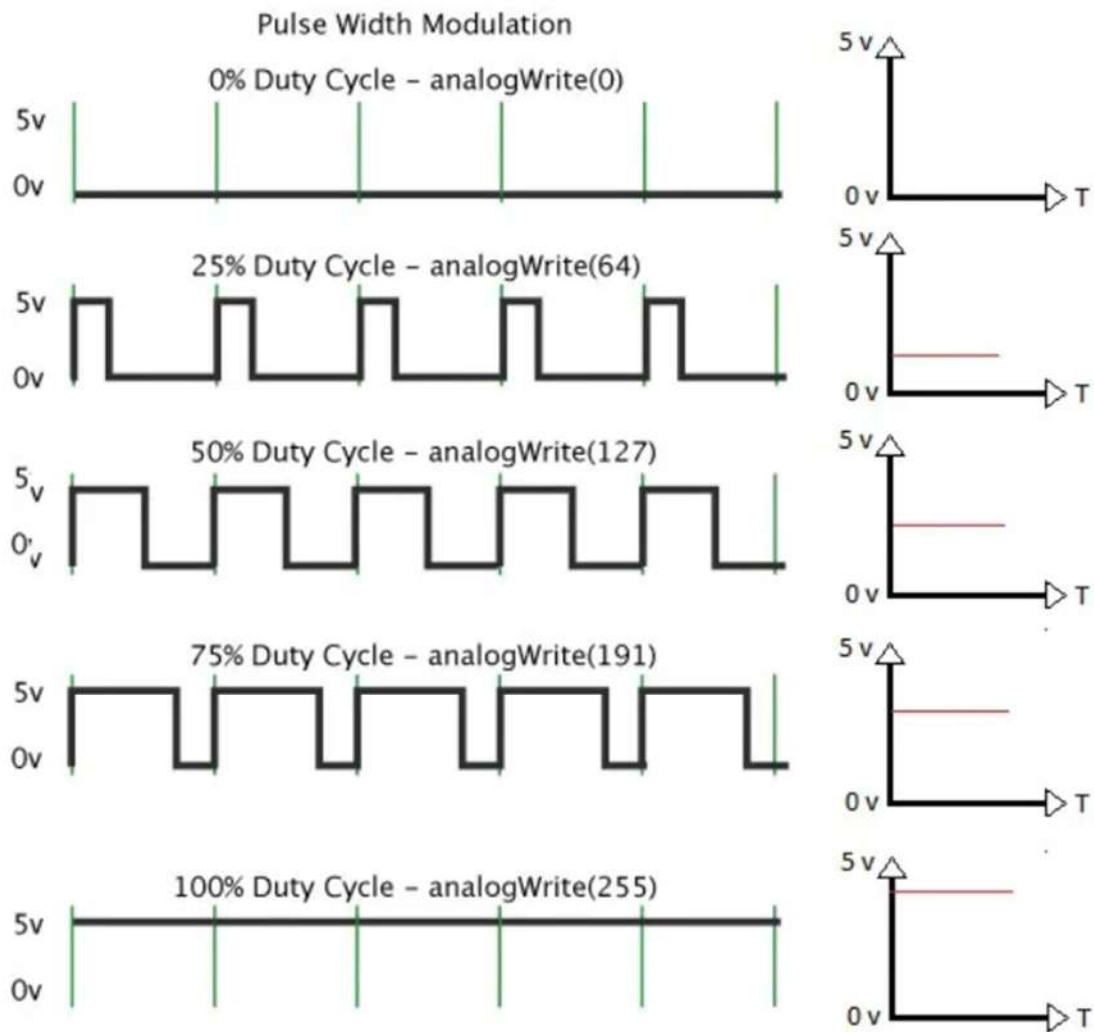
Na pinove Arduino pločice s mogućnošću pulsno širinske modulacije (PWM) spojimo dvije LE diode i napišimo program koji će prvu LE diodu uključiti da svijetli sa 20 % jačine svjetlosti a drugu sa 80 % jačine svjetlosti.

Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 2*LE dioda,
- 2*Otpornik 220 Ω ,
- 1*Eksperimentalna pločica,
- 5* Kratkospojničke žice (muško-muške).

Šta je PWM?

PWM (Pulse Width Modulation) ili pulsno širinska modulacija je tehnika dobijanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. PWM signal se smatra digitalnim signalom jer je u svakom trenutku nivo signala ili na nuli ili na maksimumu. Na Arduino PWM pinovima ovom tehnikom, odnosno upisivanjem različitih vrijednosti između 0 i 255, možemo dobiti vrijednosti izlaznih napona između 0 i 5V, ne samo granične vrijednosti. Na nekim mikrokontrolerima PWM je dostupan samo na odabranim pinovima. Označavaju se znakom tilde (~).

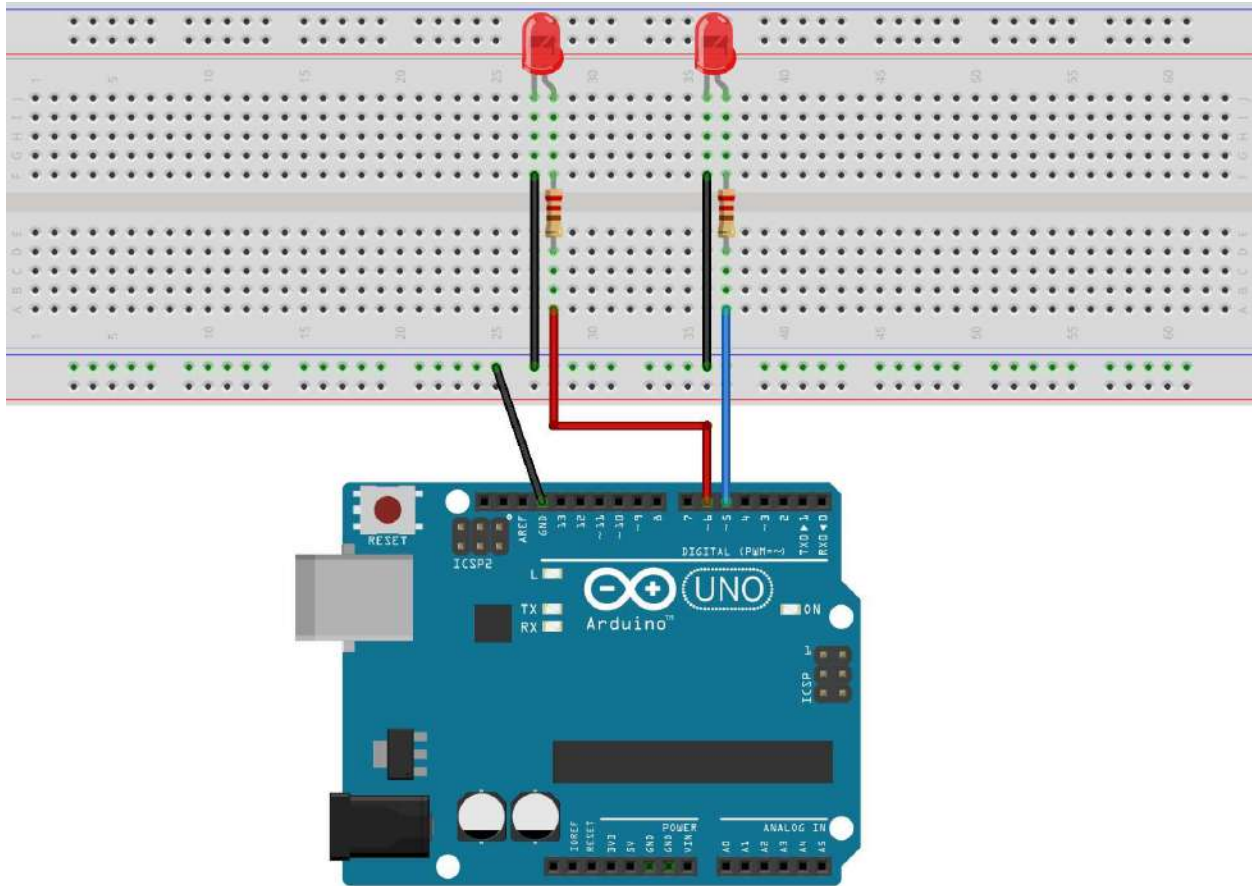


U grafičkom prikazu, zelene linije predstavljaju redovno vremensko razdoblje. Ovo trajanje ili razdoblje je obrnuto od PWM frekvencije. Drugim riječima, s Arduino PWM frekvencijom na oko 500 Hz, zelene linije bi mjerile svake 2 milisekunde.

Poziv na **analogWrite()** je na skali od 0 do 255, tako da **analogWrite(255)** zahtijeva 100% radni ciklus (uvijek uključen), a **analogWrite(127)** je 50% radnog ciklusa (na pola radnog vremena).

Postupak

10. Izgradite sklop



fritzing

11. Program

```
/*  
Stalna vrijednost  
STEM Lab  
*/  
int ledPin_Led1=5;           // Definiranje digitalnog pina 5 kao pina za  
                             // upravljanje LE diodom 1  
int ledPin_Led2=6;         // Definiranje digitalnog pina 6 kao pina za  
                             // upravljanje LE diodom 2  
  
void setup()  
{  
  pinMode(ledPin_Led1,OUTPUT); // Postavite digitalni način rada s pina 5, IZLAZ:  
                               // izlazni način  
  pinMode(ledPin_Led2,OUTPUT); // Postavite digitalni način rada s pina 6, IZLAZ:  
                               // izlazni način  
  
  digitalWrite(ledPin_Led1,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI  
                                  // (isključite LE diodu 1)  
  digitalWrite(ledPin_Led2,LOW); // ledPin (pin 6) postavite u stanje ISKLJUČI  
                                  // (isključite LE diodu 2)  
}  
void loop()                  // Stalno se izvršava  
{  
  
  analogWrite(ledPin_Led1,51); // ledPin (pin 5) postavite u stanje UKLJUČI  
                               // (uključite LE diodu 1 na 20% jačine svjetlosti)  
  analogWrite(ledPin_Led2,204); // ledPin (pin 6) postavite u stanje UKLJUČI  
                               // (uključite LE diodu 2 na 80% jačine svjetlosti)  
}
```

12. Prenesite program na Arduino UNO pločicu.

Lekcija 5

Pojačavanje

Zadatak

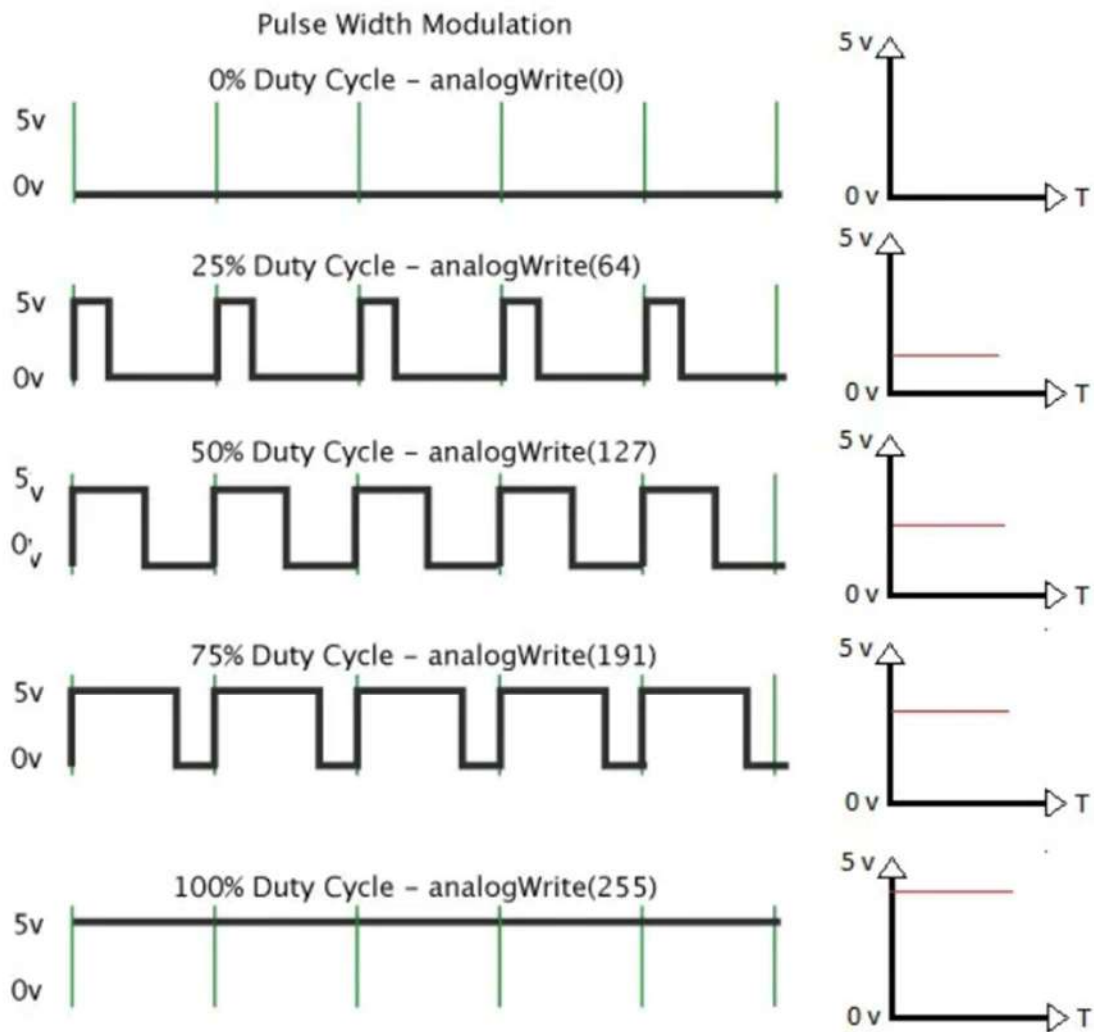
Na pin broj 5 Arduina s mogućnošću pulsno širinske modulacije (PWM) spojimo jednu LE diodu i napišimo program koji će mijenjati jačinu svjetlosti LE diode od 0% do 100% u jednakim koracima. Kada LE diode postigne maksimalan intenzitet svjetlosti neka ciklus krene ispočetka.

Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*LE dioda,
- 1*Otpornik 220 Ω ,
- 1*Eksperimentalna pločica,
- 2* Kratkospojničke žice (muško-muške).

Šta je PWM?

PWM (Pulse Width Modulation) ili pulsno širinska modulacija je tehnika dobijanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. PWM signal se smatra digitalnim signalom jer je u svakom trenutku nivo signala ili na nuli ili na maksimumu. Na Arduino PWM pinovima ovom tehnikom, odnosno upisivanjem različitih vrijednosti između 0 i 255, možemo dobiti vrijednosti izlaznih napona između 0 i 5V, ne samo granične vrijednosti. Na nekim mikrokontrolerima PWM je dostupan samo na odabranim pinovima. Označavaju se znakom tilde (~).



U grafičkom prikazu, zelene linije predstavljaju redovno vremensko razdoblje. Ovo trajanje ili razdoblje je obrnuto od PWM frekvencije. Drugim riječima, s Arduino PWM frekvencijom na oko 500 Hz, zelene linije bi mjerile svake 2 milisekunde.

Poziv na **analogWrite()** je na skali od 0 do 255, tako da **analogWrite(255)** zahtijeva 100% radni ciklus (uvijek uključen), a **analogWrite(127)** je 50% radnog ciklusa (na pola radnog vremena).

14. Program

```
/******  
Pojačavanje  
STEM Lab  
*****/  
int ledPin=5; // Definiranje digitalnog pina 5 kao pina za  
              // upravljanje LE diodom  
int strength; // Definiranje varijable strength - jačina  
              // svjetlosti LE diode  
  
void setup()  
{  
  pinMode(ledPin,OUTPUT); // Postavite digitalni način rada s pina 5, IZLAZ:  
                          // izlazni način  
  
  digitalWrite(ledPin,LOW); // ledPin (pin 5) postavite u stanje ISKLJUČI  
                             // (isključite LE diodu)  
}  
void loop() // Stalno se izvršava  
{  
  // Izvršavaj varijablu jačina od 0 do 255 u koracima od 1  
  
  for (strength=0; strength<256; strength++)  
  {  
    analogWrite(ledPin,strength); // LE dioda mijenja jačinu svjetlosti  
    delay(50); // Postavite vrijeme čekanja, 50 milisekundi  
  }  
}
```

15. Prenesite program na Arduino UNO pločicu.

Lekcija 6

Tipkalo i LE dioda

Zadatak

Na pinove broj 6 i 7 Arduino pločice spojimo jedno tipkalo i jednu LE diodu i napišimo program koji će upravljati LE diodom na način da kada je tipkalo pritisnuto uključi LE diodu a kada je tipkalo otpušteno isključi LE diodu.

Komponente

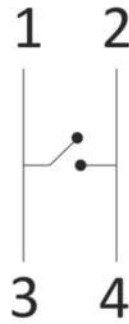
- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*LE dioda,
- 1*Otpornik 220 Ω ,
- 1*Tipkalo
- 1*Eksperimentalna pločica,
- 5* Kratkospojničke žice (muško-muške).

Šta je tipkalo?

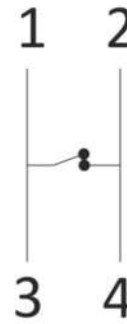
Tipkalo (eng. Pushbutton) je jedna od najčešćih komponenti za unos podataka. Tipkalo ima dva stanja: otpušteno (PULL UP) i pritisnuto (PULL DOWN).



Kao što se može vidjeti na slici ispod krajevi 1 i 3 su uvijek povezani, kao i krajevi 2 i 4.



NORMAL

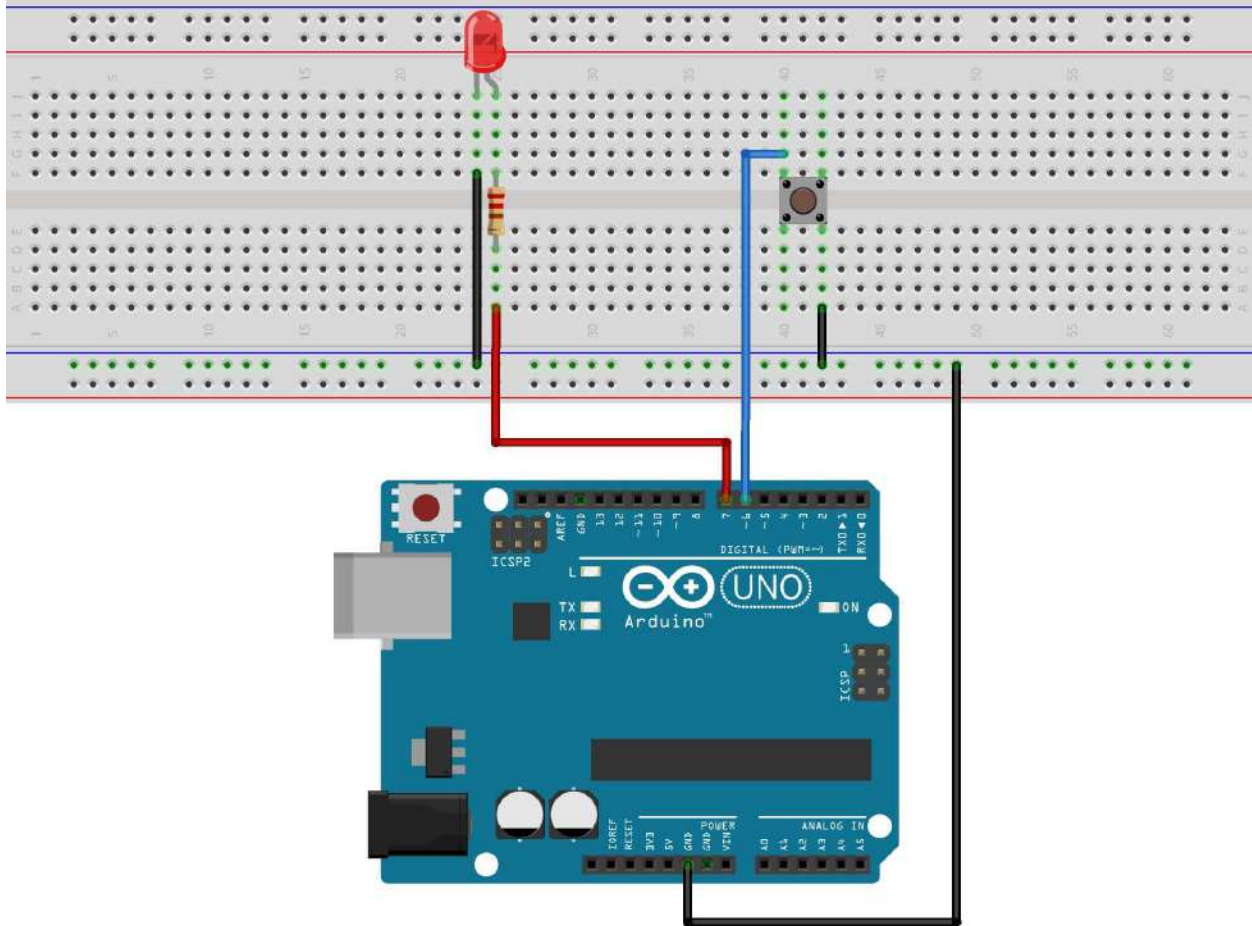


BUTTON
PRESSED

U slučaju kada je tipkalo pritisnuto ostvaruje se veze između dva kraja. Na slici ispod je prikazana šema i kod, gdje je korišten interni otpornik naredbom `INPUT_PULLUP`. Također, postavljen je i jedan uslov `if`. Obzirom da je korištena naredba `digitalRead` moguća su dva digitalna stanja, 0 ili 1. Prema šemi i korištenju `INPUT_PULLUP` (internog otpornika), kao rezultat dobit ćemo stanje 1 kada tipkalo nije pritisnuto i stanje 0 kada je tipkalo pritisnuto.

Postupak

16. Izgradite sklop



fritzing

17. Program

```
/*  
Tipkalo i LE dioda  
STEM Lab  
*/  
int ledPin=7; // Definiranje digitalnog pina 7 kao pina za  
// upravljanje LE diodom  
int buttonPin=6; // Definiranje digitalnog pina 6 kao pina za  
// upravljanje tipkalom  
int stateButton; // Deklaracija varijable stanje tipkala  
void setup()  
{  
  
pinMode(ledPin, OUTPUT); // Postavite digitalni način rada s pina 7, IZLAZ:  
// izlazni način  
pinMode(buttonPin, INPUT_PULLUP); // Postavite digitalni način rada s pina 6, ULAZ:  
// ulazni način  
digitalWrite(ledPin, LOW); // ledPin (pin 7) postavite u stanje ISKLJUČI  
}  
void loop()  
{  
stateButton=digitalRead(buttonPin); // Očitavanje stanja tipkala  
  
if(stateButton==HIGH) { // Ako je tipkalo pritisnuto  
digitalWrite(ledPin, LOW); // Uključi LE diodu  
}  
else { // Ako je tipkalo otpušteno  
digitalWrite(ledPin, HIGH); // Isključi LE diodu  
}  
}
```

18. Prenesite program na Arduino UNO pločicu.

Lekcija 7

Buzzer (zujalica)

Zadatak

Na pin broj 6 Arduino pločice spojimo pasivnu zujalicu (buzzer) i napišimo program koji će generisati zvuk frekvencije 440 Hz (ton A4) u trajanju od 1 sekunde.

Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*Pasivna zujalica (buzzer),
- 1*Eksperimentalna pločica,
- 2* Kratkospojničke žice (muško-muške).

Šta je buzzer?

Piezzo buzzer, također poznak kao zujalica je komponenta koja se koristi za generisanje zvuka. Moguće je podešavati frekvenciju kako bi proizveli ton određene frekvencije. Postoje aktivni i pasivni Piezzo buzzeri.



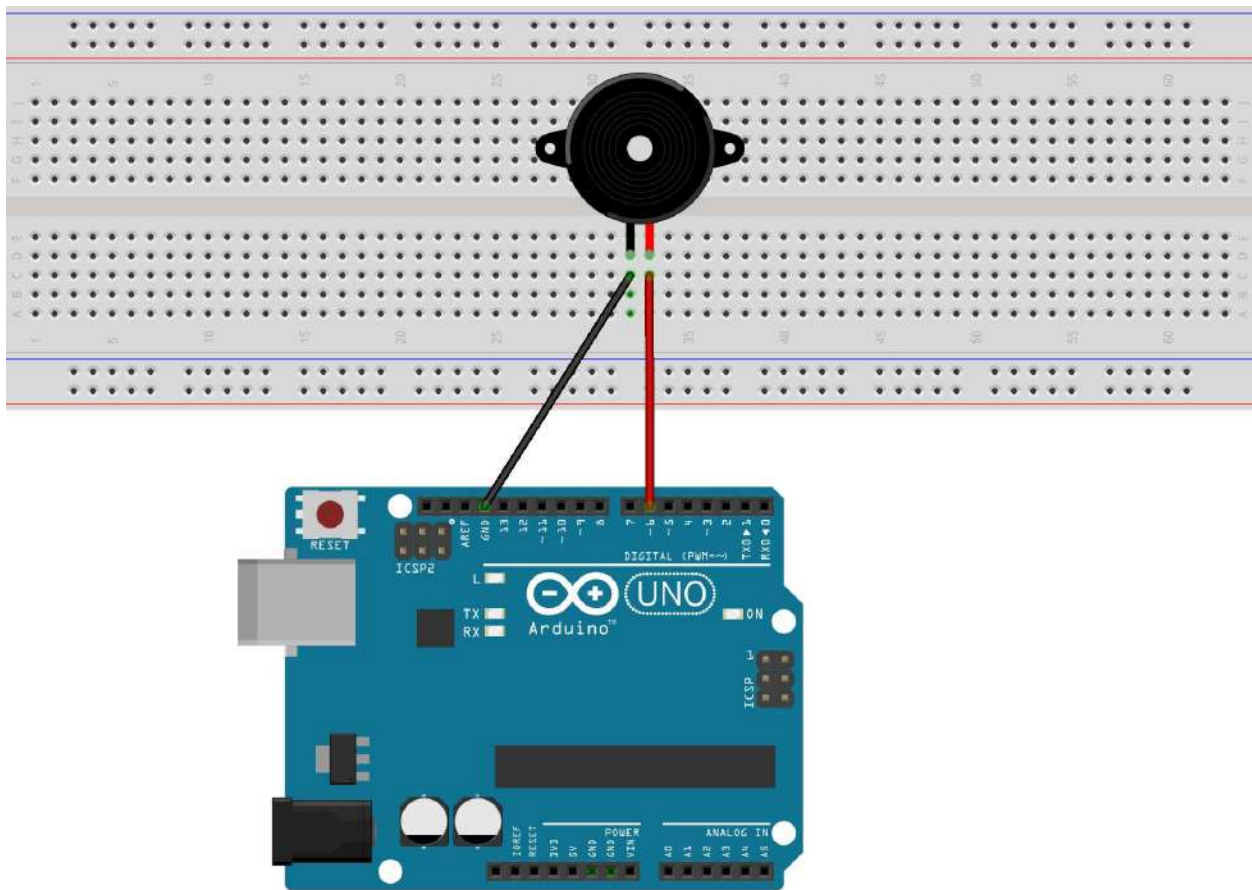
Aktivni imaju ugrađen oscillator koji generiše konstantnu frekvenciju te se aktiviraju postavljanjem stanja HIGH na pinove, dok pasivni nemaju ugrađen osciloskop i kontrolišu se dovođenjem izmjeničnog signala na pinove. Razlikuju se po tome što pasivni imaju oznaku HX na kućištu, dok aktivni nemaju.

U vježbi koristimo pasivni buzzer koji ima dvije nožice. Pozitivna nožica sa oznakom + na kućištu spaja se na digitalni pin Arduino pločice, a negativna nožica na pin uzemljenja (GND).

Zvuk se generiše pomoću funkcije **tone()** koja mora primiti parameter rednog broja pina i frekvencije zvuka kojeg treba proizvesti, odnosno note. Možemo dodati treći parameter koji govori koliko milisekundi će se izvoditi određena nota. Funkcija **noTone()** prima samo oznaku broja pina te prekida izvođenje zvuka na zadanom pinu.

Postupak

19. Izgradite sklop



fritzing

20. Program

```
/******  
Buzzer (zujalica) - Generisanje zvuka  
STEM Lab  
*****/  
int buzzer=6; // Definiranje digitalnog pina 6 kao pina za  
              upravlanje zujalicom  
  
void setup(){  
  pinMode(buzzer,OUTPUT); // Postavite digitalni način rada s pina 6, IZLAZ:  
                          izlazni način  
  tone(6, 440, 1000); // Generišite zvuk frekvencije 440 Hz (nota A4)  
                      u trajanju od 1 sekunde  
  delay(1000); // Postavite vrijeme čekanja, 1000 milisekundi  
               (1 sekunda)  
  
  noTone(buzzer); // Isključite zvuk  
}  
  
void loop(){}
```

21. Prenesite program na Arduino UNO pločicu.

Budući da se funkcije **tone()** i **noTone()** nalaze u **setup()** dijelu programa zvuk će se čuti samo jednu sekundu. Kako bi se zvuk ponavljao svake sekunde funkcije **tone()** i **noTone()** je potrebno navesti unutar **loop()** dijela programa umjesto **setup()**.

Frekvencije za sve note je moguće pronaći u datoteci **pitches.h** na internetu koja se može inicijalizirati u kodu pozivom naredbe **#include<pitches.h>**.

Tabela ispod prikazuje tonove i njihove frekvencije.

0	1	2	3	4	5	6	7	8
C 16	C 33	C 65	C 131	C 262	C 523	C 1047	C 2093	C 4186
C# 17	C# 35	C# 69	C# 139	C# 278	C# 554	C# 1109	C# 2218	C# 4435
D 18	D 37	D 73	D 147	D 294	D 587	D 1175	D 2349	D 4699
D# 20	D# 39	D# 78	D# 156	D# 311	D# 622	D# 1245	D# 2489	D# 4978
E 21	E 41	E 82	E 165	E 330	E 659	E 1319	E 2637	E 5274
F 22	F 44	F 87	F 175	F 349	F 699	F 1397	F 2794	F 5588
F# 23	F# 46	F# 93	F# 185	F# 370	F# 740	F# 1475	F# 2960	F# 5920
G 25	G 49	G 98	G 196	G 392	G 784	G 1568	G 3136	G 6272
G# 26	G# 52	G# 104	G# 208	G# 415	G# 831	G# 1661	G# 3322	G# 6645
A 28	A 55	A 110	A 220	A 440	A 880	A 1760	A 3520	A 7040
A# 29	A# 58	A# 117	A# 233	A# 466	A# 932	A# 1865	A# 3729	A# 7459
B 31	B 62	B 124	B 247	B 494	B 988	B 1976	B 3951	B 7902

Lekcija 8

Servo motor

Zadatak

Na pin broj 6 Arduino pločice spojimo servo motor SG90 i napišimo program koji će svake sekunde mijenjati položaj osovine servo motora kako slijedi: 30°, 60°, 90°, 180°, zatim nazad na 90°, 0° i neka se to ponavlja.

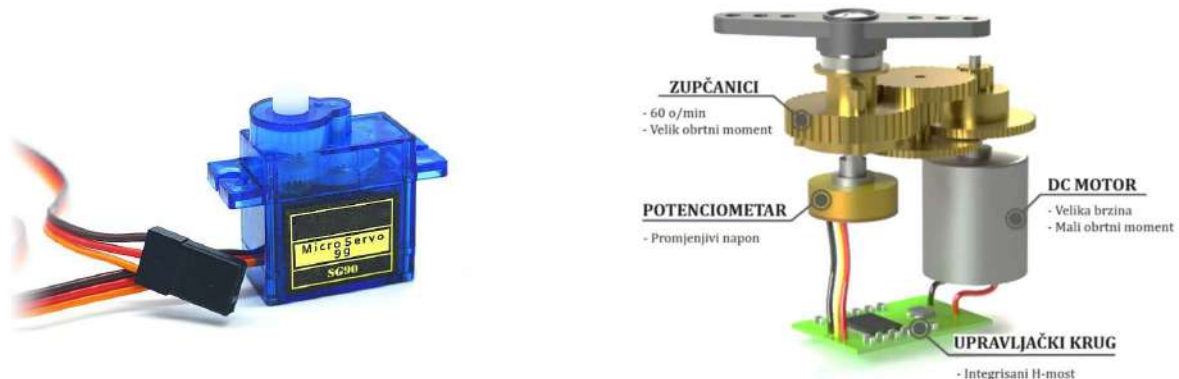
Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*Servo motor SG90,
- 3* Kratkospojničke žice (muško-muške).

Šta je servo motor?

Servo motor je sistem zatvorene petlje koji koristi povratnu informaciju o položaju za upravljanje svojim kretanjem i konačnim položajem. Postoji mnogo vrsta servo motora, a njihova glavna karakteristika je mogućnost precizne kontrole položaja njihove osovine.

U ovoj lekciji koristimo popularni plavi hobi servo motor SG90 koji ima obrtni moment od 1,5 kg/cm sa samo 9 grama. Često se koristi u RC automobilima, RC avionima, robotici i industriji.

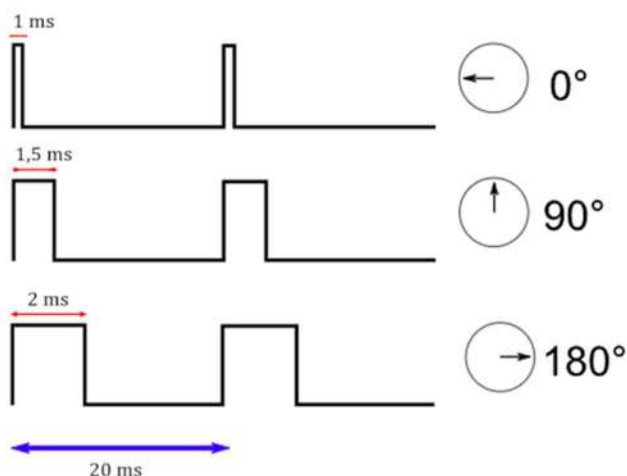


Postoje četiri glavne komponente unutar hobi servo motora: DC motor, zupčanici, potencijometar i upravljački krug.

DC motor ima veliku brzinu i mali obrtni moment, ali zupčanici smanjuju brzinu na oko 60 obrtaja u minuti i istovremeno povećavaju obrtni moment.

Potenciometar je pričvršćen na krajnji zupčanik ili izlaznu osovinu, tako da kako se osovina okreće, potenciometar se također okreće, proizvodeći napon koji je vezan za apsolutni ugao osovine.

U upravljačkom krugu ovaj se napon potenciometra uspoređuje s naponom koji dolazi iz signalne linije. Ako je potrebno, kontroler aktivira integrisani H-most koji omogućuje rotaciju motora u bilo kojem smjeru dok dva signala ne dostignu razliku od nule.



Ovim se aktuatorom može upravljati pomoću signala moduliranog širinom pulsa (PWM) od 50 Hz, koji proizvodi impuls svakih 20 milisekundi. Položaj aktuatora može se podesiti promjenom trajanja impulsa između 1 ms i 2ms i obično se mogu rotirati od 0° do 180° (imaju fizičko ograničenje kretanja). Položaj servo uređaja može se kontrolisati samo PWM signalom.

Servo motori troše mnogo energije, posebno kada su izloženi velikom momentu. Za rad servo motora SG90 optimalni napon je 5V. Budući da se pin od 5V Arduino pločice napaja direktno preko USB sabirnice, računar će ga ograničiti na oko 500 mA. Sa jednim ili dva spojena servo motora, USB napajanje bi trebalo držati opterećenje. Nakon što prekoračite dva servo motora, najbolje je koristiti zasebno napajanje. Obavezno spojite pin GND sa Arduino pločice na negativni priključak napajanja aktuatora.



Servo motor SG 90 ima tri izvoda: napajanje, uzemljenje i upravljački signal.

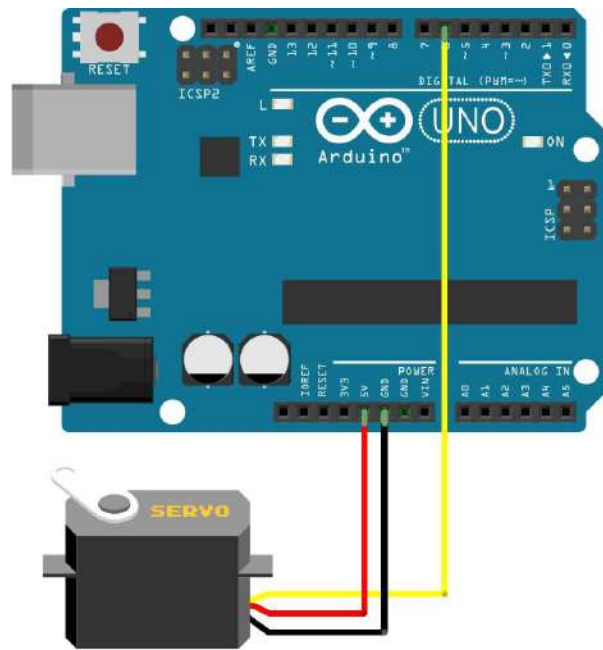
Izvod za napajanje je tipično crvena boja i spajamo ga na pin 5V na Arduino pločici.

Izvod za uzemljenje je obično crna ili smeđa boja i spajamo ga na pin GND na Arduino pločici.

Signalni izvod je žuta, narandžasta ili bijela boja i spajamo ga na digitalni pin (PWM) na Arduino pločici.

Postupak

22. Izgradite sklop



fritzing

23. Program

```

/*****
Servo motor
STEM Lab
*****/
#include <Servo.h> // Uključivanje biblioteke Servo
Servo myservo; // Kreiranje objekta Servo za kontrolu
                 servo motora

void setup(){
  myservo.attach(6); // Definiranje pina 6 kao pina za upravljanje
                    // servo motorom
  myservo.write(0); // Postavite polugu servo motora u početni
                   // položaj (0 stepeni)
}
void loop(){
  myservo.write(30); // Stalno se izvršava
                    // Pomjerite polugu servo motora u položaj
                    // (30 stepeni)

```

```

delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)
myservo.write(60);    // Pomjerite polugu servo motora u položaj
                       // (60 stepeni)
delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)
myservo.write(90);    // Pomjerite polugu servo motora u položaj
                       // (90 stepeni)
delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)
myservo.write(180);   // Pomjerite polugu servo motora u položaj
                       // (180 stepeni)
delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)

myservo.write(90);    // Pomjerite polugu servo motora u položaj
                       // (90 stepeni)
delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)
myservo.write(0);     // Pomjerite polugu servo motora u položaj
                       // (0 stepeni)
delay(1000);           // Postavite vrijeme čekanja, 1000 milisekundi
                       // (1 sekunda)

```

24. Prenesite program na Arduino UNO pločicu.

Lekcija 9

LCD

Zadatak

Na Arduino pločicu spojimo LCD 16x2 i potencijometar od 10 K Ω za upravljanje pozadinskim osvjetljenjem kao što je prikazano na slici ispod. Napišimo program koji će u prvom redu ispisati poruku "Arduino!" a u drugom redu poruku "STEM Lab Tesanj".

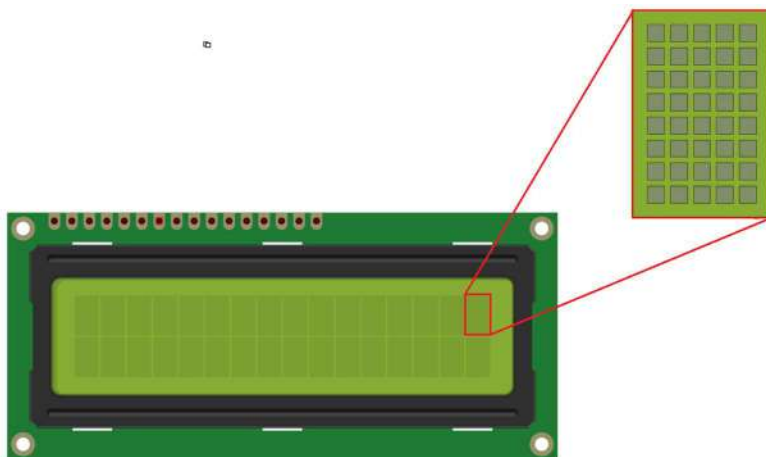
Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*Eksperimentalna pločica,
- 1*LCD 1602,
- 1*Potencijometar 10 K Ω ,
- 16*Kratkospojničke žice (muško-muške).

Šta je LCD?

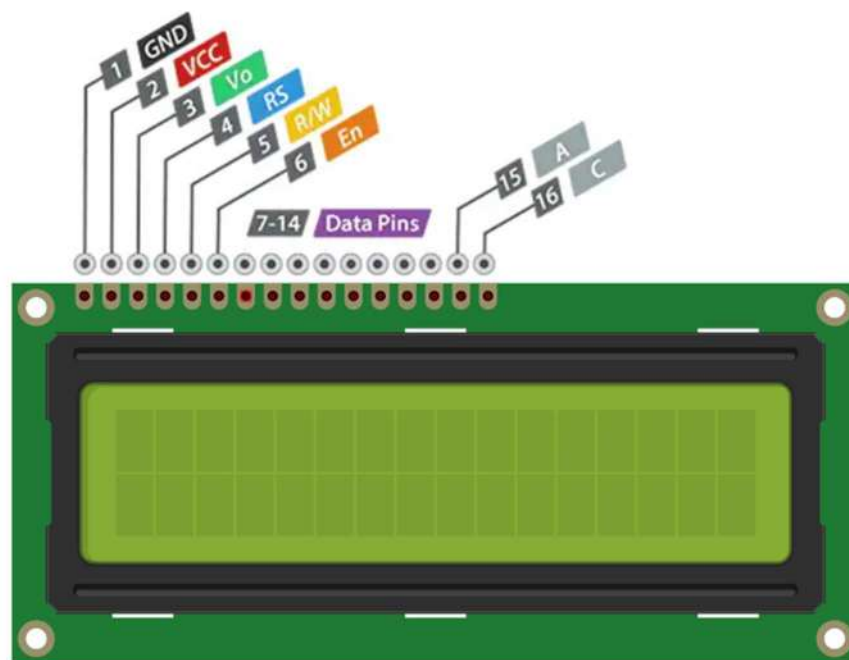
LCD (Liquid Crystal Display)-displej s tečnim kristalima je vrsta displeja koji koristi tečne kristale za prikaz znakova. Kada se aktiviraju električnom strujom, tečni kristali postaju neprozirni, blokirajući pozadinsko osvjetljenje koje se nalazi iza displeja. Kao rezultat toga, to će područje biti tamnije od ostatka. Aktiviranjem sloja tečnih kristala u određenim pikselima mogu se generisati znakovi.

LCD-i su idealni za prikaz znakova. LCD 16x2, na primjer, može prikazati 32 ASCII znaka u dva reda. Ako pažljivo pogledamo, vidjet ćemo sićušne pravougaonike za svaki znak na ekranu, kao i piksele koji čine znak. Svaki od pravougaonika je mreža od 5x8 piksela.



LCD displeji sa znakovima dostupni su u raznim veličina i bojama, uključujući 16x1, 16x4, 20x4, bijeli tekst na plavoj pozadini, crni tekst na zelenoj pozadini i mnoge druge. U ovom primjeru koristimo LCD 16x2, crni tekst na zelenoj pozadini.

Standardni karakterni LCD ima 16 pinova.



GND je kontakt za uzemljenje.

VCC je napajanje LCD-a i obično je spojen na 5V.

Vo (LCD Contrast) kontroliše contrast LCD-a. Koristeći jednostavnu mrežu razdjelnika napona i potencijometar, možemo izvršiti precizna podešavanja kontrasta.

RS (Register Select) pin se koristi za odvajanje naredbi (kao što je postavljanje pokazivača na određenu lokaciju, brisanje ekrana, itd) od podataka. RS pin je postavljen na LOW kada šalje naredbe na LCD i HIGH kada šalje podatke.

R/W (Read/Write) pin omogućava čitanje podataka ili pisanje podataka na LCD.

E (Enable) pin se koristi za uključivanje ekrana. Kada je ovaj pin postavljen na LOW, LCD ignoriše aktivnost na R/W, RS i linijama sabirnice podataka, kada je postavljen na HIGH, LCD obrađuje dolazne podatke.

D0-D7 (Data Bus) pinovi prenose 8-bitne podatke koje šaljem na ekran. Na primjer, da bismo vidjeli veliko slovo "A" na ekranu, ove pinove postavljamo na 0100 0001 (prema ASCII tabeli).

A-K (Anode & Cathode) koriste se za kontrolu pozadinskog osvjetljenja LCD-a.

Ovisno o proizvođaču, neki LCD-i uključuju otpornik za ograničavanje struje za pozadinsko osvjetljenje. Nalazi se na zadnjoj strani LCD-a, blizu pina 15. Ako vaš LCD ne sadrži ovaj otpornik ili ako niste sigurni sadrži li ga, dodajte jedan otpornik između pina 5V i pina 15. Trebalo bi biti sigurno da koristite otpornik vrijednosti 220Ω , iako ovako visoka vrijednost može prigušiti pozadinsko osvjetljenje. Za bolje podatke provjerite podatkovnu tablicu za maksimalnu struju pozadinskog osvjetljenja i odaberite odgovarajuću vrijednost otpornika.

Za fino podešavanje kontrasta ekrana i za najbolju vidljivost na LCD spojimo potenciometar.

Šta je potenciometar?

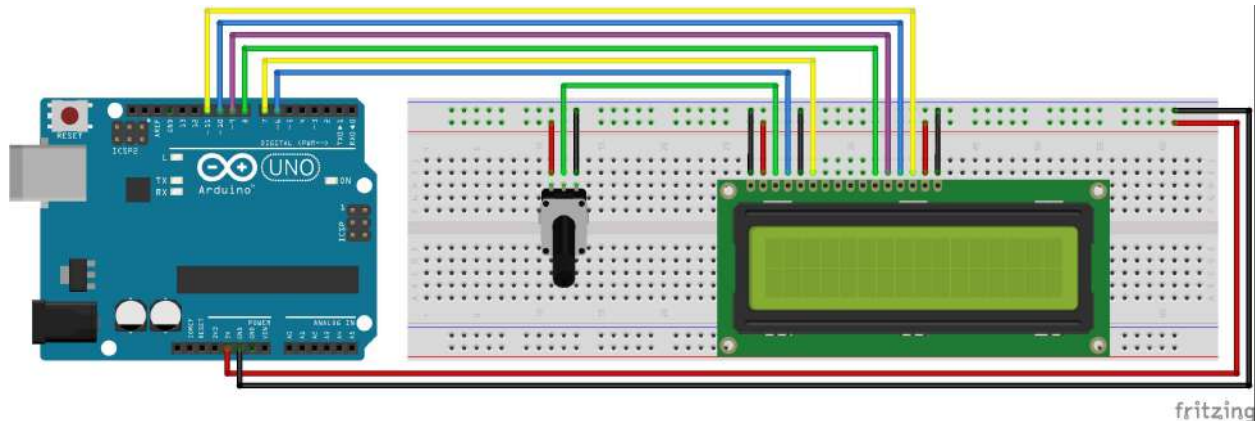
Potenciometar je promjenjivi otpornik koji funkcioniše kao razdjelnik napona. Koriste se u mnogim uređajima za regulaciju jačine zvuka, nivoa osvjetljenja itd.



Potenciometar ima tri terminala. Jedan krajnji terminal potenciometra spajamo na izvor napajanja (5V) a drugi terminal spajamo na uzemljenje (GND). Krajnji terminali (1) povezani su kliznom stazom-trakom (2) koja se izrađuje od otpornog materijala, najčešće grafita ili karbonskih materijala miješanih sa keramikom ili plastikom. Srednji terminal koji spajamo na pin Vo LCD-a je wiper-klizač (3) koji je i najvažniji dio potenciometra. Klizač klizi po kliznoj traci i njegova pozicija određuje napon na srednjem terminalu. Klizačom upravljamo okretanjem ručice. Ako je klizač bliže VCC (+) terminalu tada je napon na srednjem pinu veći, a ako je bliže GND (-) terminalu tada je napon manji.

Postupak

25. Izgradite sklop



26. Program

```

/*****
LCD 1602
STEM Lab
*****/
#include<LiquidCrystal.h>           // Biblioteka
LiquidCrystal lcd (6, 7, 8, 9,10,11); // Inicijalizirajte biblioteku brojevima pinova

void setup(){
  lcd.begin(16,2);                // Postavite broj redova i kolona LCD-a
}

void loop(){
  lcd.setCursor (4,0);             // Postavite kursor u kolonu 4, red 0
  lcd.print ("Arduino!");         // Ispišite poruku na LCD-u
  lcd.setCursor (0,1);           // Postavite kursor u kolonu 0, red 1
  lcd.print ("STEM Lab Tesanj");  // Ispišite poruku na LCD-u
}

```

27. Prenesite program na Arduino UNO pločicu.

Lekcija 10

DHT11 i DHT22 senzor vlažnosti i temperature

Zadatak



Na Arduino pločicu spojimo DHT11 senzor vlažnosti i temperature, LCD 16x2 i potenciometar od 10 K Ω za upravljanje pozadinskim osvjetljenjem kao što je prikazano na slici ispod. Napišimo program koji će u prvom redu ispisati izmjerenu vrijednost vlažnosti izraženu u postocima a u drugom redu izmjerenu temperaturu izraženu u stepenima celzijusa.

Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*Eksperimentalna pločica,
- 1*DHT11 senzor vlažnosti i temperature,
- 1*LCD 1602,
- 1*Potenciometar 10 K Ω ,
- 19*Kratkospojničke žice (muško-muške).

DHT11 i DHT22 senzor vlažnosti i temperature

DHT11 i DHT22 su iste vrste senzora s nekim razlikama u izvedbi. Oba koriste kapacitivne senzore vlažnosti i termistore za mjerenje relativne vlažnosti i temperature okoline.

Senzor	DHT11	DHT22
Izgled		
Raspon temperature	0 – 50° C	-40 – 80° C
Tačnost temperature	$\pm 2^{\circ}$ C	$\pm 0,5^{\circ}$ C
Raspon vlažnosti	20 – 90%	0 – 100%
Tačnost vlažnosti	5%	2%
Rezolucija	Vlažnost: 1% Temperatura: 1° C	Vlažnost: 0,1% Temperatura: 0,1° C

DHT22 je skuplji, što znači da je sposobniji od DHT11. DHT22 može mjeriti temperature od -40° do +80° C s ±0,5° tačnosti, dok DHT11 može mjeriti od 0° do 50° C s ±2° tačnosti.

Također, senzor DHT22 ima bolji senzor vlažnosti koji može mjeriti relativnu vlažnost od 0 do 100% s 2% tačnosti, dok DHT11 može mjeriti od 20 do 80% s 5% tačnosti.

Za očitavanje vlažnosti, DHT11 ima otpornu komponentu koja ima dvije elektrode i supstrat koji zadržava vlagu između njih. Kada podloga koja apsorbira vodenu paru prisutnu u zraku, oslobađa ione koji povećavaju provodljivost između dvije elektrode.

Promjena otpora između dvije elektrode obrnuto je proporcionalna relativnoj vlažnosti. Kada se vlažnost u zraku poveća, otpor između elektroda se smanjuje, a kada se vlažnost smanji, otpor između elektroda se poveća.

DHT11 ima NTC termistor za mjerenje temperature. Termistor je toplotni otpornik čiji je otpor jako ovisi o temperature. Izraz "NTC" znači "negativan temperaturni koeficijent" što znači da njegova otpornost opada s povećanjem temperature.

Senzor dolazi s 14-pinskim, 8-bitnim mikrokontrolerom koji pretvara analogne podatke sa senzora vlažnosti i termistora u digitalne vrijednosti. Generiše digitalni signal koji sadrži vrijednosti relativne vlažnosti, temperature i bajt kontrolne sume.

Apsolutna vlažnost naspram relativne vlažnosti

DHT11 i DHT22 mogu mjeriti samo relativnu vlažnost. Apsolutna vlažnost je ukupna masa vodene pare prisutna u određenom volumenu ili masi zraka. Ne uzima u obzir temperature. S druge strane, relativna vlažnost je omjer količine vodene pare u zraku i količine vodene pare koju zrak potencijalno može sadržavati na određenoj temperaturi.

Formula za izračunavanje relativne vlažnosti je:

$$RH = \left(\frac{pw}{ps} \right) \times 100\%$$

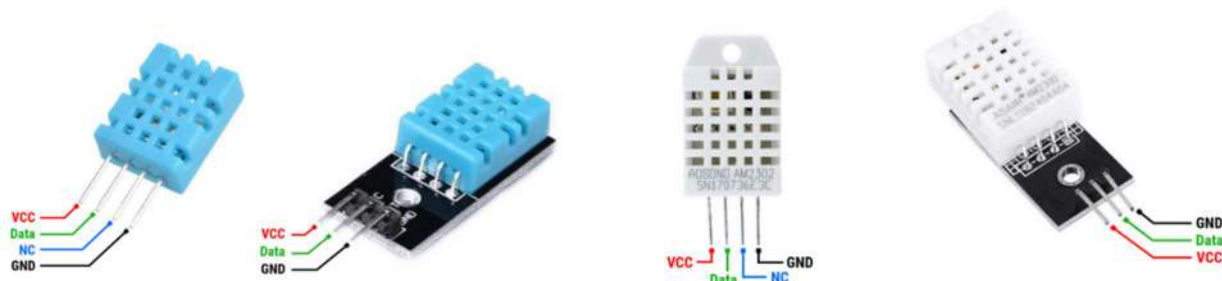
Gdje je:

RH-relativna vlažnost,
pw-gustoća vodene pare pri određenoj temperaturi,
ps-gustoća vodene pare pri zasićenju na toj temperature.

Relativna vlažnost se izražava u postocima, tako da 0% relativne vlažnosti znači da je zrak potpuno suh, a 100% relativne vlažnosti znači da dolazi do kondenzacije.

Pinovi DHT11 i DHT22

Na donjoj slici možete vidjeti da i DHT11 i DHT22 imaju četiri pina, VCC, Data, No Connection i GND pin. Također dolaze sa verzijom montiranom na PCB koja ima tri pina, VCC, Data i GND pin.



DHT11 pinovi

DHT22 pinovi

VCC pin osigurava napajanje senzora. Unatoč činjenici da se napon napajanja kreće od 3,3 V do 5,5 V, preporučuje se napajanje od 5 V. Uz napajanje od 5 V, senzor se može postaviti do 20 metara udaljenosti. Uz napon napajanja od 3,3 V, senzor se može postaviti do 1 metar udaljenosti, inače će pad napona uzrokovati greške u mjerenju.

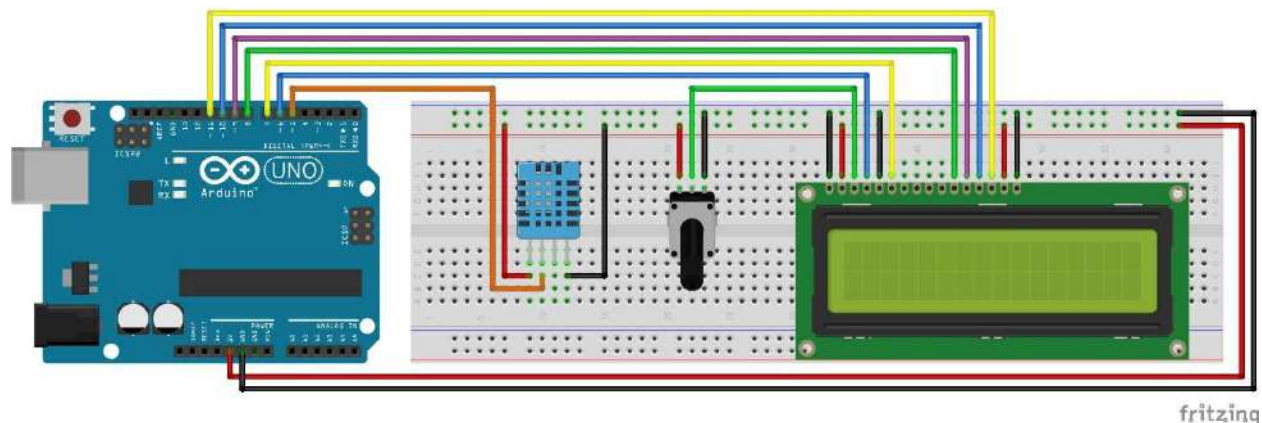
Data pin služi za komunikaciju između senzora i mikrokontrolera.

NC pin nije spojen.

GND je kontakt za uzemljenje.

Postupak

28. Izgradite sklop



29. Program

```
/******  
Senzor vlažnosti i temperature DHT11 i LCD 1602  
STEM Lab  
***** */  
  
#include <dht11.h> // Biblioteka DHT11  
#include <LiquidCrystal.h> // Biblioteka LCD 1602  
  
dht11 DHT11;  
#define DHT11PIN 5 // Inicijalizacija senzora DHT11 (pin 5)  
LiquidCrystal lcd(6, 7, 8, 9, 10,11);  
void setup()  
{  
  lcd.begin(16, 2); // Postavite broj kolona i redova na LCD-u  
  lcd.clear(); // Izbrišite LCD displej i postavite kursor u  
                // gornji lijevi ugao  
  delay(1000); // Kašnjenje 1000 ms  
}  
void loop()  
{  
  int chk = DHT11.read(DHT11PIN);  
  lcd.setCursor(0, 0); // Postavite kursor na kolonu 0, red 0  
  lcd.print("Vlaznost:"); // Ispišite poruku "Vlaznost:" na LCD-u  
  lcd.print((float)DHT11.humidity, 2); // Ispišite vrijednost izmjerene vlažnosti  
  lcd.print(" % "); // Ispišite jedinicu za izražavanje postotka  
                    // vlažnosti (% - postotak)  
  
  lcd.setCursor(0, 1); // Postavite kursor na kolonu 0, red 1  
  lcd.print("Temp: "); // Ispišite poruku "Temp:" na LCD-u  
  lcd.print((float)DHT11.temperature, 2); // Ispišite vrijednost izmjerene temperature  
  lcd.print(" C "); // Ispišite jedinicu za izražavanje temperature  
                    // (C - Celzijus)  
  delay(1000); // Kašnjenje 1000 ms  
}
```

30. Prenesite program na Arduino UNO pločicu.

Lekcija 11

Fotootpornik (LDR) i LE dioda

Zadatak

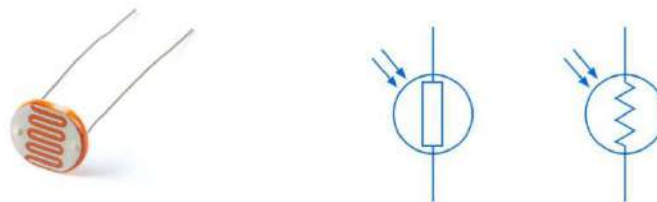
Na Arduino pločicu spojimo LDR na pin A1 i LE diodu na pin broj 8 kao što je prikazano na slici ispod. Napišimo program koji upravlja LE diodom tako da je ona uključena kada je očitana vrijednost na digitalnom pinu manja od vrijednosti analognog praga (500), inače LE dioda je isključena.

Komponente

- 1*Arduino UNO R3 mikrokontroler,
- 1*USB kabal,
- 1*Eksperimentalna pločica,
- 1*LDR (fotootpornik),
- 1*Otpornik 10 k Ω ,
- 1*LE dioda,
- 1*Otpornik 220 Ω ,
- 7*Kratkospojničke žice (muško-muške).

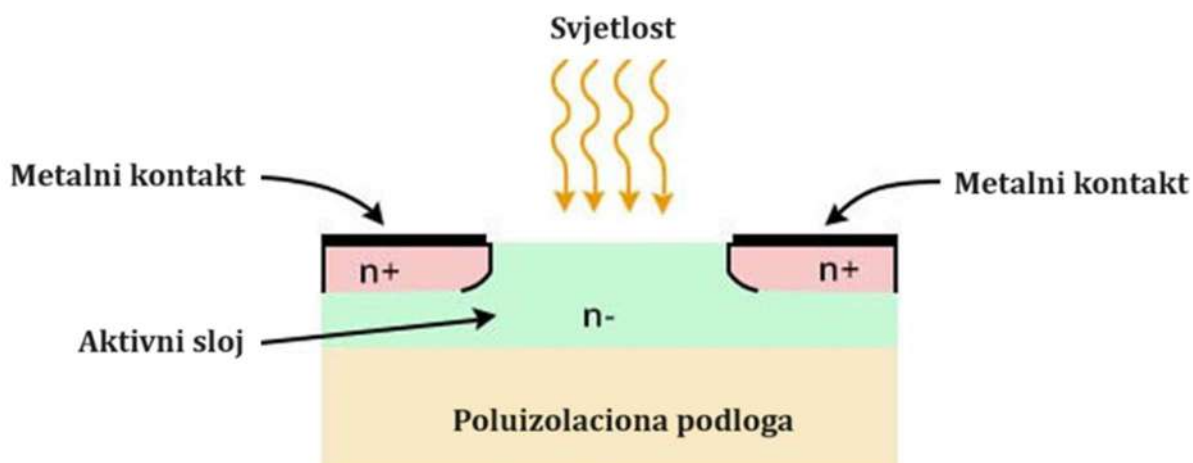
Šta je LDR?

LDR (Light Dependet Resistor) popularno poznat kao fotootpornik je fotoelektrični uređaj koji se koristi u dizajnu elektroničkih sklopova u svrhu otkrivanja prisutnosti ili intenziteta svjetlosti. LDR mijenja svoju otpornost u zavisnosti od količine svjetla koja pada ne njega. LDR se proizvodi korištenjem poluprovodničkih materijala kako bi im se dala svojstva osjetljivosti na svjetlost. Koriste se različiti materijali, a kadmijev sulfid (CdS) je polularan izbor za ove fotootpornike. Olovni sulfid (PbS) i iindijev antimonid (InSb) primjeri su drugih materijala koji se koriste.



Simbol koji se koristi za predstavljanje LDR u elektroničkim sklopovima temelji se na simbolu kruga otpornika, uključuje strelice koje prikazuju prisutnost svjetla.

Fotootpornici rade na principu fotoprovodljivosti. Prema principu fotoprovodljivosti određeni materijali, poznati kao fotoprovodnici, pokazuju promjene u svojoj električnoj provodljivosti kada su izloženi svjetlosti.



Kada svjetlost padne na LDR, fotoprovodljivi materijal apsorbira energiju koju svjetlost nosi. Kao rezultat toga, elektroni u valentnom pojasu fotoprovodljivog materijala postaju pobuđeni i prelaze u provodljivi pojas. Ovaj proces dovodi do povećanja provodljivosti proporcionalno intenzitetu upadne svjetlosti.

Da bi došlo do ovog fenomena, energija koju nosi upadna svjetlost mora biti veća od energije razmaka fotoprovodljivog materijala. Tek tada se elektroni u valentnom pojasu mogu pobuditi u provodljivi pojas.

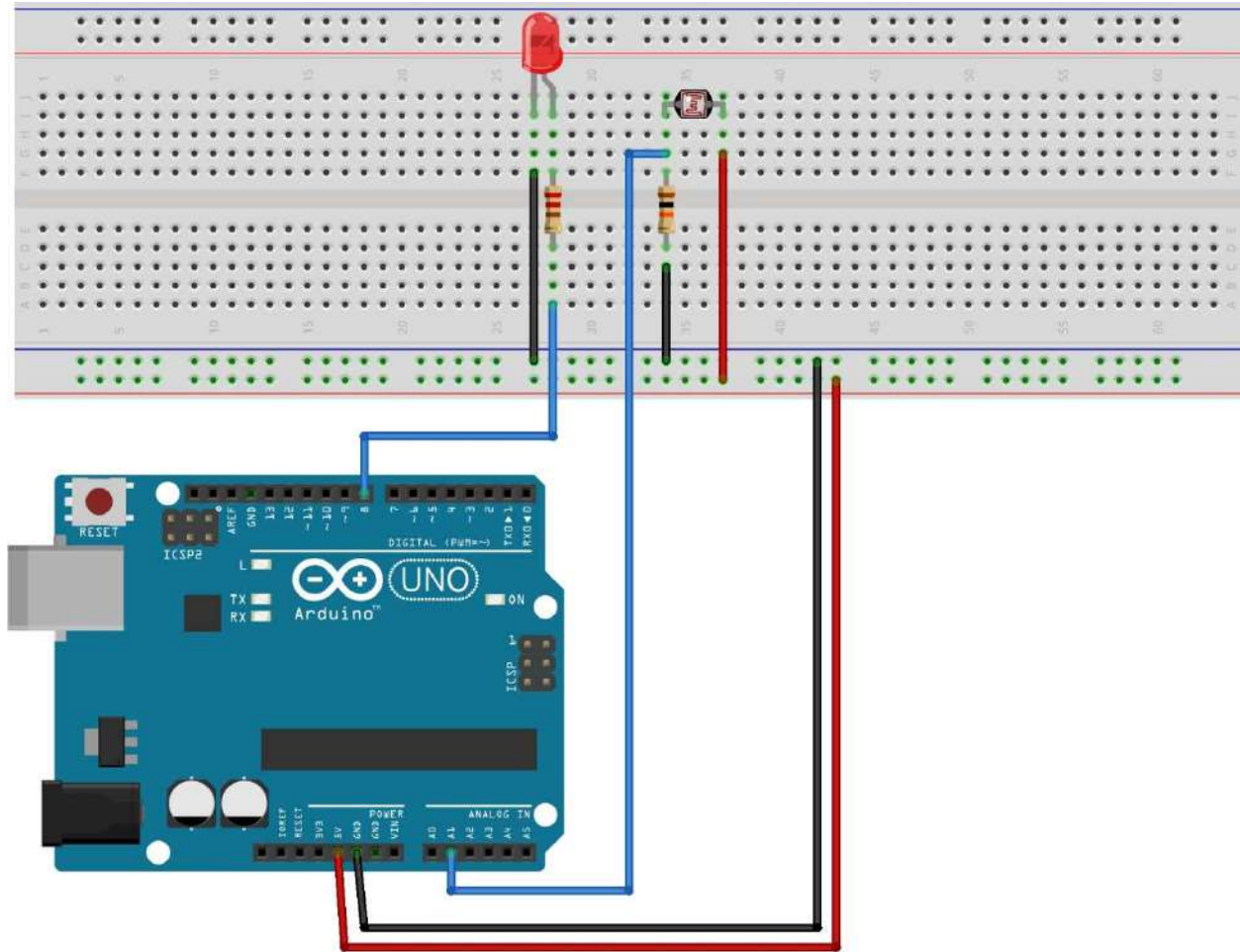
U svom neosvijetljenom stanju (tama), LDR pokazuje najveći otpor. Međutim, kako se intenzitet svjetlosti povećava, otpor LDR-a se smanjuje u skladu s tim.

LDR ima dva pina koji su simetrični, ne moramo ih razlikovati pri spajanju. Jedan pin LDR-a se spaja na analogni pin (A0 do A5), na što se u paralelnu vezu dodaje otpornik od 10 k Ω koji se dalje spaja na uzemljenje (GND). Drugi pin LDR-a se spaja na izvor napona (pin 5V).

Arduino Uno pinovi A0 do A5 mogu raditi kao analogni ulazi. Analogni ulazni pin pretvara napon (između 0V i VCC) u cjelobrojnu vrijednost (između 0 i 1023), koja se naziva ADC vrijednost ili analognu vrijednost. Povezivanjem pina LDR-a na pin analognog ulaza, možemo čitati analognu vrijednost sa pina koristeći **analogRead()** funkciju, a zatim relativno znati nivo svjetlosti.

Postupak

31. Izgradite sklop



fritzing

32. Program

```
/******  
Fotootpornik (LDR) i LE dioda  
STEM Lab  
***** */  
  
const int LIGHT_SENSOR_PIN=A1;      // Definiranje pina A1 kao pina za  
                                     upravljanje fotootpornikom (LDR-om)  
const int LED_PIN=8;                // Definiranje pina 8 kao pina za upravljanje  
                                     LE diodom  
const int ANALOG_THRESHOLD=500;     // Postavite analogni prag  
  
int analogValue;                    // Definiranje varijable analogna vrijednost  
  
void setup() {  
  pinMode(LED_PIN, OUTPUT);         // Postavite način rada s pina 8, IZLAZ:  
                                     izlazni način  
}  
  
void loop() {  
  analogValue = analogRead(LIGHT_SENSOR_PIN); // Očitavanje ulazne vrijednosti na  
                                               analognom pinu (LDR-u)  
  
  if(analogValue < ANALOG_THRESHOLD) // Ako je očitana vrijednost na  
                                       analognom pinu manja od  
                                       analognog praga (500)  
    digitalWrite(LED_PIN, HIGH);      // LE dioda je uključena  
  else                                  // inače  
    digitalWrite(LED_PIN, LOW);       // LE dioda je isključena  
}
```

33. Prenesite program na Arduino UNO pločicu.