

TEHNIČKA ŠKOLA ŽUPANJA

LABORATORIJSKE VJEŽBE

**AUTOMATSKO VOĐENJE PROCESA**

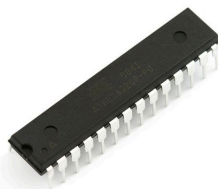
LABORATORIJSKA VJEŽBA

- UPOZNAVANJE S ARDUINOM –

**ŽUPANJA 2017**

## Uvod u mikrokontrolere i Arduino

Već smo nekoliko puta spomenuli mikrokontrolere, a zapravo nismo pojasnili što su to mikrokontroleri. Postoje razne definicije, ali najjednostavnije rečeno mikrokontroler je računalo u malom. Mikrokontroleri su mala računala smještena na jedan integrirani sklop. Na sljedećoj slici prikazan je mikrokontroler s kojim ćemo mi raditi u ovoj knjizi – njegov naziv je ATMEGA328 a proizvodi ga tvrtka ATMEL.



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

Kako bi cijeli taj 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 mikokotrolerima je Arduino, pa ćemo tako i mi, kroz ove laboratorijske vježbe savladati mikrokontrolere upravo pomoću Arduina.

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

Cijela Arduino platforma napravljena je i objavljena kao otvoreni sustav – to znači da su sve sheme i izvorni kôd programa kojega ćemo koristiti za programiranje Arduina besplatno dostupni svima za preuzimanje i modificiranje sa službene stranice Arduino platforme – [www.arduino.cc](http://www.arduino.cc). Mi ćemo u ovoj knjizi koristiti jednu od Arduino pločica pod nazivom – Arduino UNO. Pločica je prikazana na sljedećoj slici.



## Instalacija Arduino IDE programa i upoznavanje

Kako bi započeli s radom, moramo preuzeti i instalirati Arduino IDE – program za pisanje programa za Arduino mikrokontrolere. Program možete besplatno preuzeti na web-adresi [www.arduino.cc](http://www.arduino.cc).

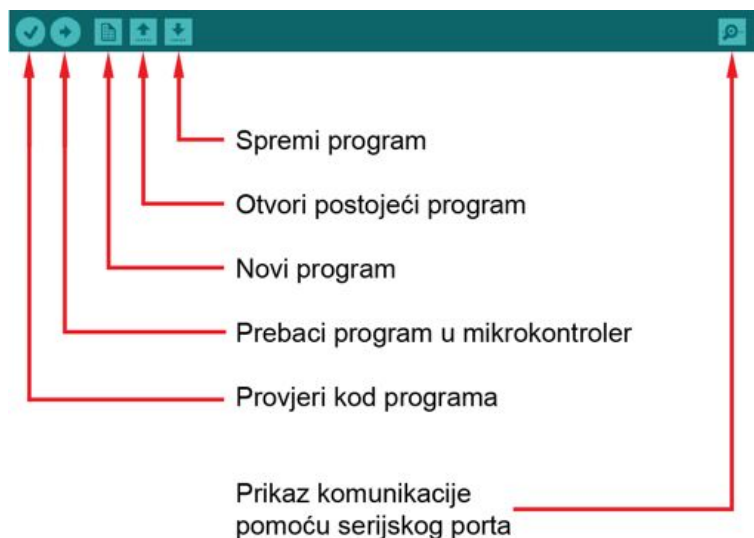
Nakon preuzimanja programa pokrenite instalaciju. Nakon nekoliko pritisaka na tipku *next* uspješno ste instalirali Arduino IDE. Pri instalaciji svakako omogućite učitavanje svih *drivera* koje Vam instalacija nudi, kako biste nesmetano mogli raditi sa Arduino sklopovljem koje ćete priključivati na računalo.

Kada ste instalirali program pokrenite ga i pogledajte osnovni izgled i funkcije koje su dostupne u programu. Prikaz programa možemo podijeliti na tri osnovna dijela kako je prikazano na sljedećoj slici.

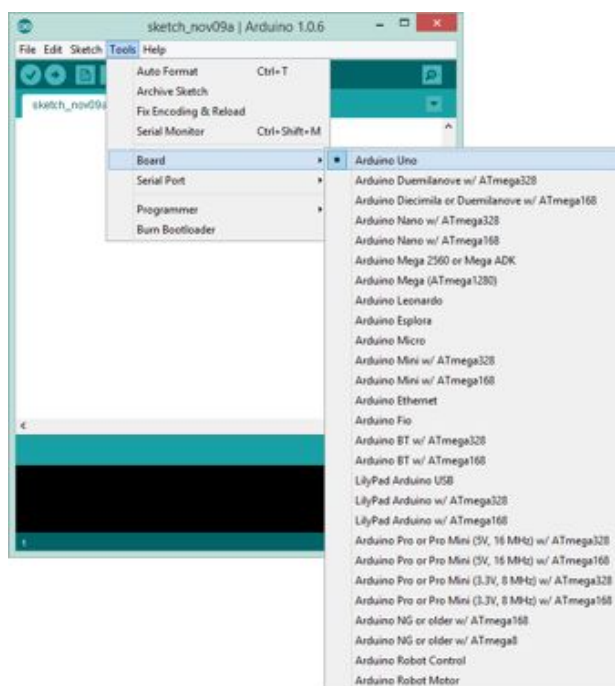


Gornji dio programa čini alatna traka sa standardnim izbornikom te dodatnim gumbima za brzi dolazak do često korištenih opcija. Srednji dio programa rezerviran je za pisanje kôda, dok je u donjem dijelu smještena konzola. Konzola služi za prikaz statusa programa te za poruke uspjehnosti prijenosa napisanog programa u mikrokontroler ili za poruke o detektiranim greškama u napisanom programu.

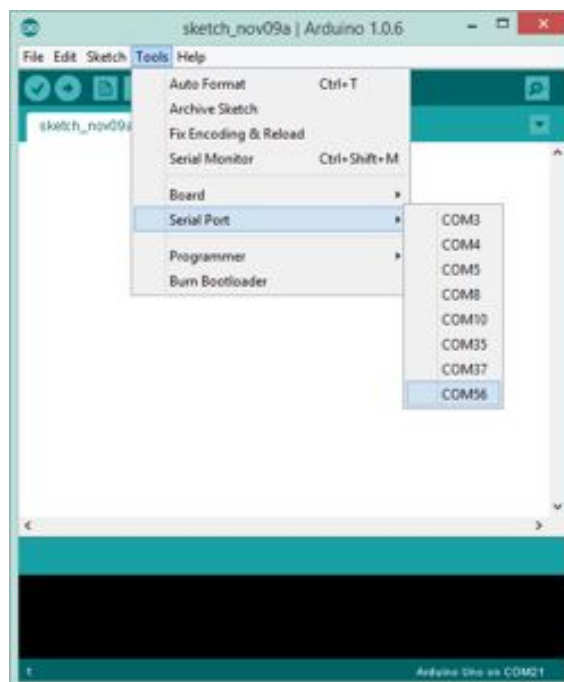
Na sljedećoj slici prikazan je izbornik s prečacima.



Prije početka rada bitno je podesiti neke od parametara Arduino IDE programa. Moramo odabrati koju Arduino pločicu koristimo, na koji je priključak računala ona spojena te na koji način želimo slati programe na mikrokontroler. Ova je podešenja dovoljno napraviti jednom ako ne mijenjate pločicu ili korišteni priključak računala. Za odabir pločice kliknite na *Tools* u gornjem izborniku, potom na *Boards* i odaberite *Arduino UNO*.



Za odabir priključka računala kliknite na *Tools* u gornjem izborniku, potom na *Serial port* te odaberite serijski priključak na kojem je spojena vaša pločica.



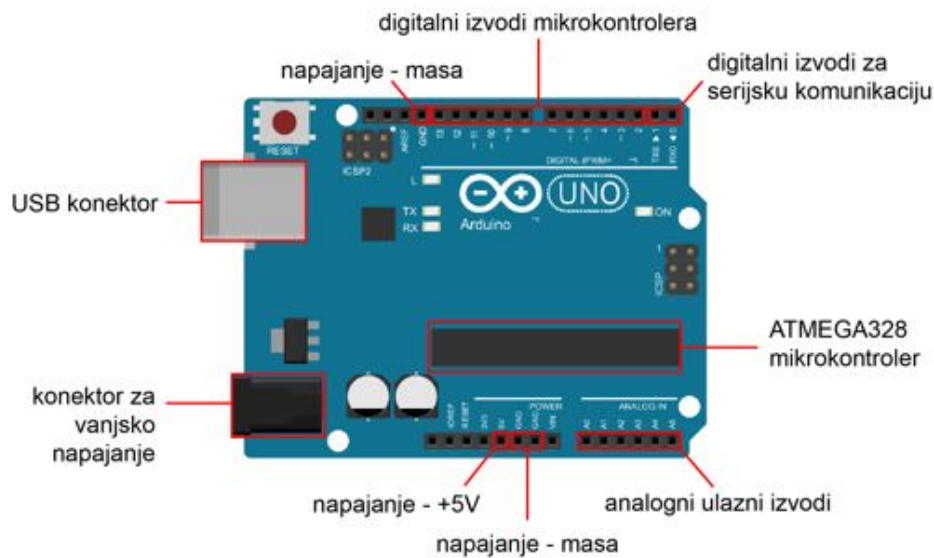
U novijoj verziji Arduino IDE programa, kada priključite Arduino pločicu uz broj serijskog porta (COM porta) na kojem je detektirana pločica pisati će i naziv pločice koja je detektirana. Na taj način možete raspoznati na koji je port pločica spojena.

## Osnovna znanja

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

Kao što smo već rekli, mikrokontroler za početak možemo zamisliti kao crnu kutiju koja ima određen broj izvoda kojima je moguće upravljati pomoću programa kojeg korisnik napiše na računalu i koji se onda izvodi na samom mikrokontroleru.

Izvodi mikrokontrolera su na Arduino UNO pločici spojeni na crne konektore na samom rubu pločice. Svaki izvod ima svoje jedinstveno ime koje je napisano kraj njega. Tako na konektoru s gornje strane imamo izvode 0,1,2,3 itd. do 13. Sa donje strane Arduino UNO pločice dodatno nalazimo izvode koji se nazivaju A0, A1, A2 itd. do A5. Dodatno u donjem dijelu pločice postoje i VCC (+5V) i GND izvodi. Osim ovih izvoda na pločici su označeni i drugi izvodi ali nam oni za sada nisu važni.



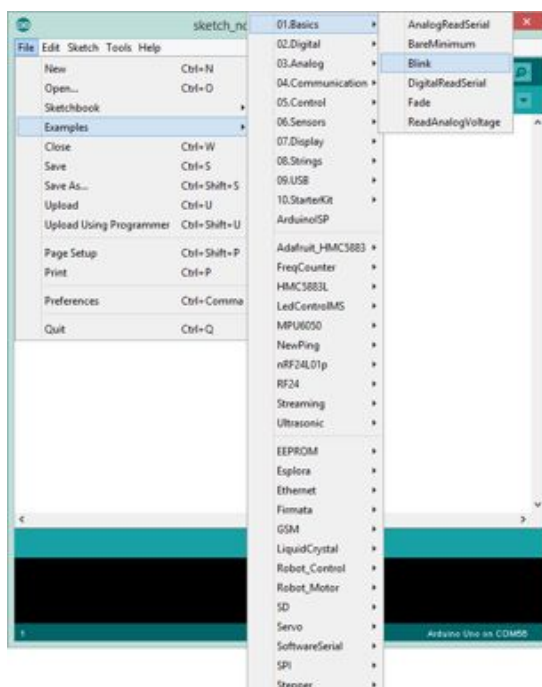
2-12.png

Izvodi mikrokontrolera služe za spajanje mikrokontrolera s vanjskim elektroničkim elementima. Mogu se koristiti za upravljanje raznim uređajima (svjetlećim diodama, motorima itd.) ili za očitavanje stanja s različitih senzora (tipkala, sklopke, senzori temperature itd.). Tako kažemo da izvodi mogu raditi na dva načina rada – izlazni način rada i ulazni način rada.

Također, izvode možemo podijeliti na digitalne i analogne. Digitalni izvodi mogu biti u samo jednom od dva moguća stanja – nisko logičko stanje i visoko logičko stanje. Kod niskog stanja na izvodu je napon od 0V, a kod visokog logičkog stanja na izvodu je napon od 5V. Kod analognih izvoda moguće je očitavati ili regulirati napon bilo koje vrijednosti između 0V i 5V.

## Prvi program

Jedna od velikih prednosti Arduino platforme je što postoji vrlo velik broj već gotovih primjera koje možemo iskoristiti za učenje i stvaranje vlastitih uređaja. Kako bi na samom početku mogli vidjeti radi li naša Arduino UNO pločica i kako bi naučili kako prebaciti program s računala u mikrokontroler iskoristit ćemo jednostavan program iz primjera u Arduino IDE programu. Klikom na izbornik *File* i odabirom kategorije *Examples* pojavljuju se kategorije dostupnih primjera. Pod kategorijom *Basics* nalazi se program *Blink*.



Otvorimo ga i pogledajmo.



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

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

Proučimo program da vidimo kako je to postignuto.

```
/*
  Blink
  Turns on 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 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 izvod što spojili odnosno definiramo varijable, u drugom dijelu programa govorimo na koji način koristimo pojedine izvode te pišemo onaj dio kôda koji se izvodi samo jednom na početku, pri pokretanju mikrokontrolera, a u trećem dijelu programa pišemo što zapravo mikrokontroler mora raditi.

U ovome primjeru u prvom dijelu, koji seže do *setup* dijela programa, nemamo ništa osim komentara. Komentari su u Arduino IDE programu prikazani sivom bojom i služe samo za prijenos informacija korisnicima, kao način da autor pojasni korisnicima koji koriste ovaj program što program radi. Komentari se često koriste kako bi se pojasnile funkcije i pojedine naredbe u programu.



```

/*
  Blink
  Turns on 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 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
*/

```

Kao što vidite, komentari moraju započeti znakovima `/*` i završiti znakovima `*/` kako bi Arduino IDE program znao da je to komentar. Kada komentari stanu u jednu liniju moguće je komentar započeti sa `//` kao što vidite u sljedećem programskom odlomku. Tada se sav tekst od znakova `//` do kraja linije smatra komentarom.

Unutar drugog dijela – kojeg nazivamo *setup* dijelom rekli smo mikrokontroleru da je izvod broj 13, na kojem je spojena svjetleća dioda 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 rada pojedinog izvoda koristimo naredbu `pinMode(nazivizvoda, načinrada)`. Ova naredba ima dva argumenta (argumenti su podatci koje upisujemo u zagrade kod naredbi) – *nazivizvoda* definira na koji se izvod odnosi naredba a *načinrada* definira je li taj izvod ulaznog ili izlaznog tipa. Kako mi želimo upravljati svjetlećom diodom onda postavljamo izvod 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 kôda koji uključuje diodu, nakon toga čeka jednu sekundu, pa isključuje diodu te ponovno 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
}

```

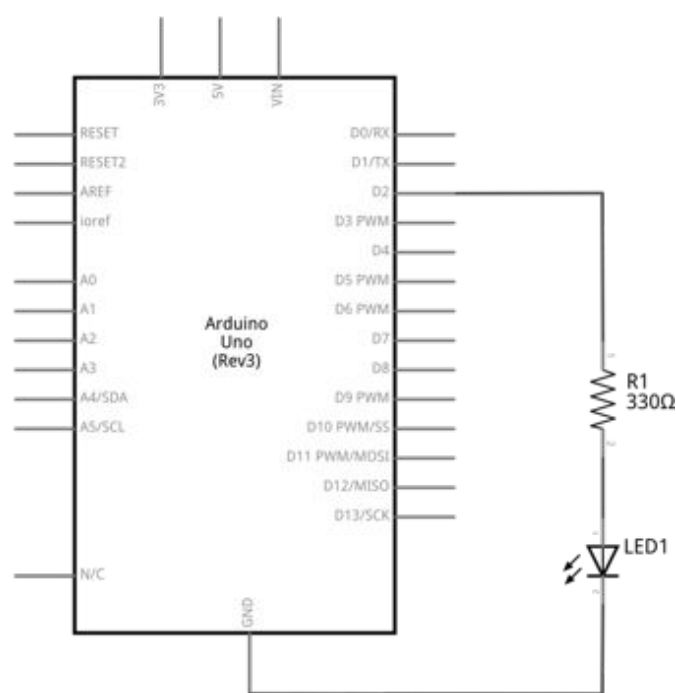
Za uključivanje i isključivanje svjetleće diode koristimo naredbu `digitalWrite`. Ova naredba također ima dva argumenta `DIGITALWRITE(nazivizvoda, stanje)`. Argument *nazivizvoda* definira na koji se izvod odnosi naredba a *stanje* definira postavljamo li izvod u visoko ili nisko stanje. Kada se izvod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

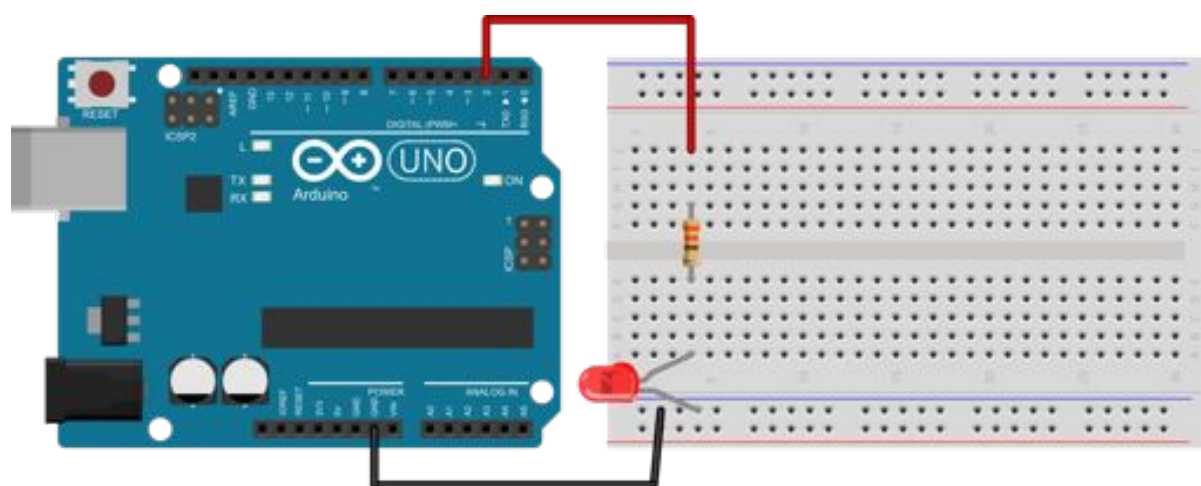
Budući da se dio kôda koji uključuje i isključuje svjetleću diodu nalazi unutar `loop` dijela programa koji se stalno ponavlja dobivamo efekt *blinkanja* svjetleće diode.

Zadatak: Na mikrokontroler spojite jednu svjetleću diodu. Napišite program koji će diodu uključivati i isključivati u intervalima od pola sekunde kako biste postigli efekt blinkanja.

Na sljedećim slikama prikazana je električna shema te način spajanja svjetleće diode na eksperimentalnoj pločici:



3-4.png



Iz gornje sheme i prikaza spajanja vidljivo je da smo odabrali izvod broj 2 za spajanje svjetleće diode. Ovo je naravno naš odabir, ništa nas ne obvezuje da koristimo upravo taj izvod – mogli smo iskoristiti bilo koji drugi digitalni izvod.

Kada smo uspješno spojili svjetleću diodu na mikrokontroler možemo započeti s programiranjem.

```
int led = 2; //definiraj led = 2

void setup() {
  pinMode(led, OUTPUT); //postavi led kao izlazni
  digitalWrite(led, LOW); //isključi LED diodu - početno stanje
}

void loop() {
  digitalWrite(led, HIGH); //uključi LED diodu
  delay(500); //čekaj 500 ms - pola sekunde
  digitalWrite(led, LOW); //isključi LED diodu
  delay(500); //čekaj 500 ms - pola sekunde
}
```

Gornji programski kôd je rješenje prvoga zadatka.

U prvom dijelu programa definirali smo da je led dioda spojena na izvod broj 2. Za to smo iskoristili varijablu koju smo nazvali *led*. Varijable su zapravo memorijske lokacije na koje možemo upisati neku vrijednost koja se može mijenjati tijekom izvođenja programa. Svaka varijabla ima svoj naziv i svoju vrijednost. U ovom slučaju naziv varijable je *led* a njezina vrijednost je 2. Korištenjem ove varijable olakšali smo si pisanje i čitanje kôda jer možemo koristiti njezin naziv umjesto broja izvoda pri radu sa izvodom.

Nakon toga, u *setup* dijelu pomoću naredbe *pinMode* postavili smo izvod koji nazivamo *led* – odnosno izvod broj 2 kao izlazni jer pomoću njega želimo upravljati svjetlećom diodom. Odmah nakon toga iskoristili smo naredbu *digitalWrite* kako bismo isključili svjetleću diodu. U ovome slučaju ovo nije nužno ali je dobra praksa na kraju *setup* dijela postaviti sva početna stanja svih izvoda s kojima nećimo upravljamo, kako bi točno znali u kojem će nam stanju biti ti izvodi.

Unutar *loop* dijela programa uključujemo i isključujemo svjetleću diodu svakih 500 milisekundi odnosno pola sekunde.

Prebacite program u mikrokontroleru i provjerite radi li zaista svjetleća dioda onako kako ste zamislili.