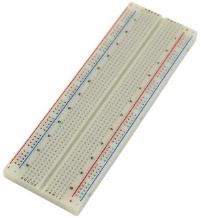
**1.DIO -** upoznavanje s opremom i primjeri

**Aktivnosti za učenike**

*Upoznavanje s opremom*

Pokazati učenicima uređaje i opremu s kojom ćemo raditi.

* Arduino MKR1000
* Arduino starter kit
* Eksperimentalne pločice
* Set senzora
* Slikovni rezultat za arduino potentiometerPrijenosni Wi-fi router



**Aktivnosti za učenike**

*Preuzimanje i instalacija Arduino razvojnog okruženja*

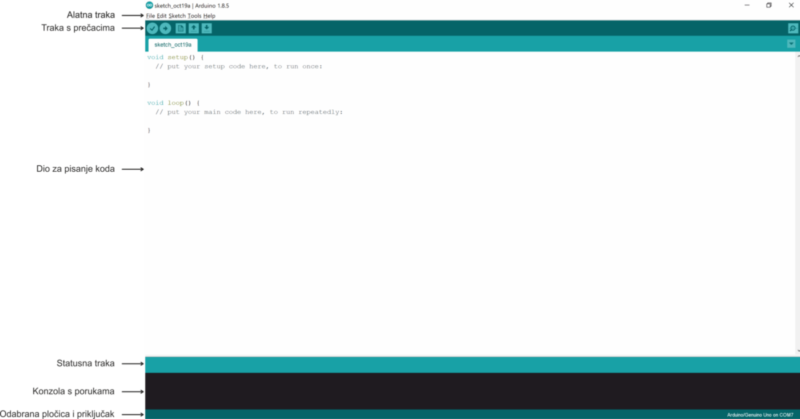
Prije nego započnemo s radom, potrebno je na računalo instalirati softver za programiranje naših Arduino pločica. Softver je otvorenog koda i potpuno je besplatan, a može se preuzeti na web stranici Arduino.cc, odnosno na sljedećem linku: https://www.arduino.cc/en/Main/Software

Nakon preuzimanja pokrenite jednostavnu instalaciju i nakon nekoliko klikova na gumb Next softver je spreman za prvo pokretanje.

**Aktivnosti za učenike**

*Pokretanje programa i podešavanje početnih postavki*

Pokrenite program putem Start izbornika ili prečaca na radnoj površini. Nakon pokretanja možemo zapaziti da je programsko okruženje podijeljeno u nekoliko dijelova kako je prikazano na sljedećoj slici.

[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/Dijelovi-IDE-su%C4%8Delja-e1516004699499.png)

Alatna traka nam osim standardnih File, Edit i Help izbornika nudi i dva specifična – Sketch i Tools. Sketch izbornik služi za podešavanje opcija korištenih knjižnica (library), i trebati će nam nešto kasnije, a Tools izbornik, između ostalog, služi za odabir korištene Arduino pločice.

U traci s prečacima možete pronaći opcije za provjeru napisanog koda programa, prebacivanje programa na mikrokontroler i neke druge opcije o kojima ćemo više govoriti kasnije.

Centralni cijeli dio služi za pisanje programskog koda.

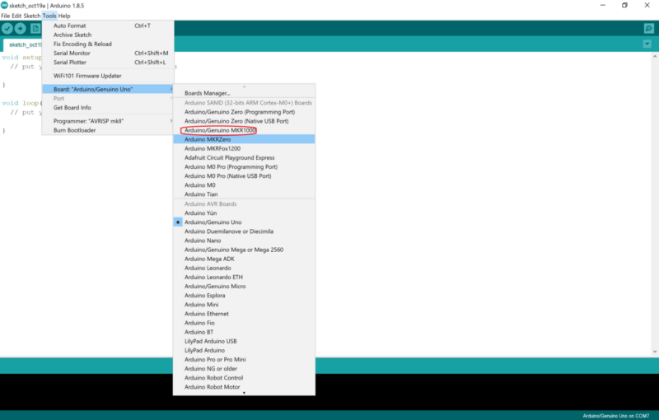
Zelena traka je statusna traka u kojoj će se ispisivati poruke o statusu prebacivanja programa u mikrokontroler.

Donji crni dio je konzola u kojoj će se ispisivati detaljne poruke vezane uz previšenje programa na strojni jezik i statusa prebacivanja programa u mikrokontroler, te druge tehničke informacije.

U donjem desnom uglu možete vidjeti koja je odabrana pločica za rad te na kojem je priključku računala spojena.

Sve ove dijelove upoznati ćete kroz rad sa sučeljem, pa ako niste sve zapamtili nemojte brinuti 🙂

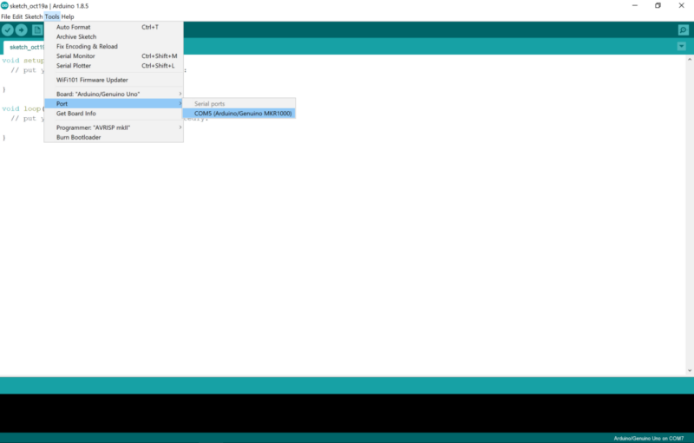
Kako mi rabimo Arduino MKR 1000, idemo to odmah podesiti. Kliknite na Tools, potom na Board i odaberite Arduino/Genuino MKR1000, kako je prikazano na sljedećoj slici.

[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/2017-10-20-08_35_46-Upute-skripta-Arduino-Word-e1516004455642.png)

Nakon odabira korištene Arduino pločice, istu možete putem USB kabla priključiti na računalo. Ukoliko pri priključenju bude potrebna instalacija upravljačkih programa i pojavi se skočni prozor svakako potvrdite instalaciju.

Nakon priključenja pločice na računalo na pločici mora zasvijetliti zelena svijetleća dioda kraj koje piše ON.

Kako bi potvrdili da je računalo uspješno prepoznalo priključnu Arduino pločicu, pogledati ćemo pojavljuje li se u listi priključnih uređaja. Kliknite na Tools, potom Port i pogledajte pojavljuje li se na listi COMx (Arduino/Genuino MKR1000), kao na slici u nastavku. Za kraj odaberite s liste stavku u kojoj piše Arduino/Genuino MKR1000.

[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/Odabir-COM-porta-e1516004470300.png)

Ove početne postavke dovoljno je napraviti jednom, na početku korištenja, ukoliko ne mijenjate pločicu ili priključni USB port na računalu.

Sada smo spremni za pisanje našeg prvog programa!

**Aktivnosti za učenike**

*Upoznajmo Arduino MKR1000*

MKR 1000 je jedan od mnogo razvojnih alata iz obitelji Arduino proizvoda. Pločica na sebi sadrži mikrokontroler XXX, modul za povezivanje na WiFi, sklopovlje za napajanje putem LiPo baterije i njezino punjenje putem USB kabla te drugo sklopovlje potrebno za rad mikrokontolera. Izgled pločice prikazan je na sljedećoj slici.

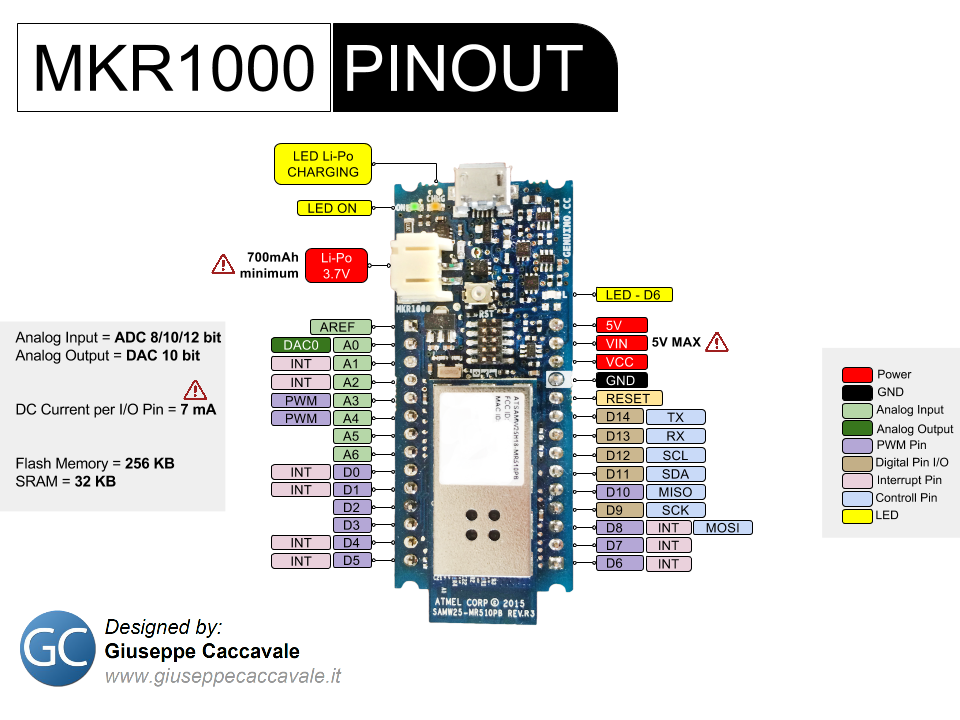
[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/mkr1000_Arduinocc.jpg)

Mikrokontroler možemo zapravo razmatrati kao malo računalo. Ovo računalo ima drugačiju namjenu od tipičnog stolnog ili prijenosnog računala – njima možemo očitavati stanja različitih senzora te upravljati raznim uređajima putem programa koji pišemo u Arduino IDE razvojnom sučelju, a koji se onda izvršava na samom mikrokontroleru.

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 učitati u mikrokontroler.

Za priključivanje dodatnih elemenata na Arduino pločice služe nam izvodi mikrokontrolera kojima možemo pristupiti preko rubnih konektora (crni konektori na rubu Arduino pločice). Prikaz svih izvoda mikrokontrolera dostupan je na sljedećoj slici.

[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/Arduino-pinout.png)

Kao što vidite, svaki izvod ima svoju jedinstvenu oznaku. Ove ćemo oznake koristiti i u programima koje pišemo, kako bi mogli upravljati izvodima. Kod Arduino pločica izvodi se označavaju slovima i brojevima. Postoje dvije vrste izvoda – digitalni izvodi i analogni izvodi.

Analogni izvodi obilježeni su slovom A i brojem – A0, A1, A2 pa do A6. Ovi izvodi mogu mjeriti analogni napon od 0-3.3V.

Digitalni izvodi obilježeni su samo brojem – 0,1,2 pa do 14, ili slovom D i brojem. Oni mogu biti u samo dva stanja – 0 ili 1, odnosno uključeno ili isključeno.

Analogne izvode možemo koristiti i kao digitalne, ali digitalni izvodi ne mogu biti analogni.

Svi izvodi mogu raditi u dva režima rada – u ulaznom i izlaznom režimu.

Izlazni režim rada koristimo kada pomoću mikrokontrolera želimo upravljati nekim komponentama koje spajamo na taj izvod (LED diodama, motorima, relejima i sl.), a ulazni režim koristimo kada želimo očitati stanje pojedinog izvoda, odnosno kada na njega spajamo jednostavne senzore poput tipkala, krajnjih sklopki, termostata i slično.

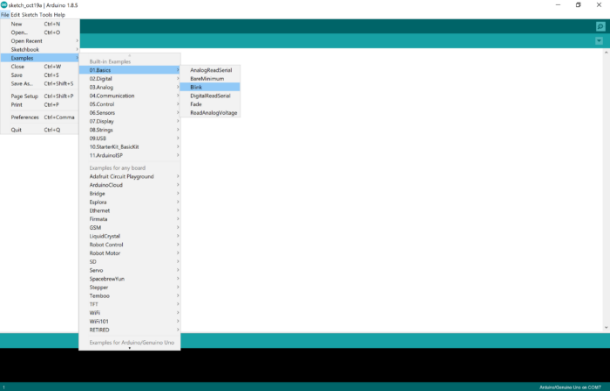
**Aktivnosti za učenike**

*Prvi program – blinkajuća svjetleća dioda*

Kako bi napisati naš prvi program bilo što jednostavnije i kako bi se upoznali s time od čega se programi za Arduino uopće sastoje prvi ćemo program zapravo „prepisati“. U samom programskom okruženju postoji niz gotovih primjera programa, koje možete pronaći kroz File izbornik.

Za početak ćemo otvoriti osnovni primjer, koji upravlja LED diodom koja je ugrađena na našu Arduino MKR1000 pločicu. Kako bi početak rada bio jednostavniji i kako bi mogli isprobati svoj Arduino a da na njega ne moramo spajati nikakvu dodatnu elektroniku, na samoj je pločici ugrađena jedna svjetleća dioda kojom možemo programski upravljati. Ova dioda je spojena na izvod broj 6 mikrokontrolera, a u sljedeći program će uključivati i isključivati diodu u jednakim intervalima – ona će „blinkati“.

Za otvaranje primjera kliknite na File, potom odaberite Examples, pa Basics i odaberite primjer Blink, kako je prikazano na sljedećoj slici.

[](https://izradi.croatianmakers.hr/wp-content/uploads/2018/01/2017-10-20-08_51_05-Filmovi-i-TV-e1516004038977.png)

U dijelu za pisanje koda pojavljuje se sljedeći program:

/\*

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.

http://www.arduino.cc/en/Tutorial/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

delay(1000); // wait for a second

}

Ovaj programski kod sastoji se od 3 osnovna dijela – komentari, setup dio koda i loop dio koda.

Sivi tekst na početku su komentari. Oni služe kako bi autor programa pojasnio korisniku što ovaj program radi i/ili dao komentare o samom programu. Ovaj dio nije nužan za ispravno funkcioniranje mikrokontrolera, ali olakšava korištenje programa krajnjem korisniku. U Arduino programskom okruženju postoje dvije vrste komentara – jednolinijski i višelinijski komentari.

Višelinijski se komentari, kako sam naziv kaže, protežu kroz više linija, a kako bi programsko okruženje znalo raspoznati komentare od naredbi komentari su odvojeni posebnim znakovima.

Višelinijski komentar započinje sa znakovima  „/\*“ (bez navodnika), nakon čega slijedi komentar, a na kraju komentara mora biti „\*/“ (bez navodnika). Sve što se nalazi između ovog para znakova smatra se komentarom. Evo jednog primjera:

Višelinijski komentar započinje sa znakovima  „/\*“ (bez navodnika), nakon čega slijedi komentar, a na kraju komentara mora biti „\*/“ (bez navodnika). Sve što se nalazi između ovog para znakova smatra se komentarom. Evo jednog primjera:

/\*

Primjer višelinijskog komentara.

Ovaj komentar proteže se kroz nekoliko linija.

\*/

Jednolinijski komentar sadržan je u samo jednoj liniji, a na početku linije koja sadrži komentar moraju biti znakovi „//“ (bez navodnika). Sve što se nalazi u redu iza ovog para znakova smatra se komentarom. Evo jednog primjera:

// ovo je jednolinijski komentar

Dio koda koji definira što će naš mikrokontroler raditi nalazi se u setup i loop dijelu koda.

Setup dio koda izvršava se samo jednom, odmah nakon uključivanja mikrokontrolera. Ovaj dio koda služi za inicijalizaciju mikrokontrolera, a ovdje ćemo najčešće pisati naredbe koje definiraju što je na mikrokontroler spojeni i na koji način. Pogledajmo setup dio ovog primjera:

// 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);

}

U prvoj liniji imamo jednolinijski komentar, koji nam govori da će se ovaj dio koda izvršiti kada se stisne tipka reset na Arduino pločici, odnosno kada se pločica priključi na napajanje.

Setup dio započinje riječima void setup(), a sve naredbe koje se nalaze u njemu moraju biti upisane između vitičastih zagrada koje označavaju početak i kraj tog dijela programa. Sve naredbe koje upišemo u setup dio programa izvrđavaju se samo jednom, na početku rada mikrokontrolera, odnosno kada ga prikljuimo na napajanje ili neposredno nakon *restarta*.

Pojednostavljeno, možemo reći da u setup dijelu programa podešavamo početne postavke za rad mikrokontrolera i definiramo način rada s komponentama koje su spojene na mikrokontroler.

U ovom primjeru jedina naredba koja se nalazi u setup dijelu je pinMode naredba. Ova naredba služi za definiranje je li pojedini izvod mikrokontrolera ulaznog ili izlaznog tipa. Obzirom da mi želimo upravljati svjetlećom diodom na jednom izvodu koristiti ćemo izlazni režim rada izvoda.

U pinMode naredbu moramo upisati dva argumeta. Argumente naredbe upisujemo u zagradu nakon naziva naredbe. U prvom argumentu definiramo za koji izvod želimo postaviti režim rada, a drugi u koji režim rada želimo postaviti izvod. Svjetleća dioda na Arduino MKR1000 pločici spojena je na izvod broj 6, ali on ima i dodatni naziv na ovoj pločici – možemo ga obilježavati s LED\_BUILTIN. Izvod ćemo postaviti u izlazni režim rada jer želimo da mikrokontroler upravlja radom svjetleće diode. Znači, koristimo naredbu pinMode(LED\_BUILTIN, OUTPUT), kao što piše u programu. Isto bi postigli i da smo umjesto LED\_BUILTIN pisali 6, odnosno da je naredba pinMode(6, OUTPUT).

Odmah nakon izvršenja setup dijela izvršava se loop dio programa. Sve naredbe koje upišemo u loop dio ponavljaju se sve dok ne *restartamo* mikrokontroler.

Kao i kod setup dijela programa i loop dio započinje ključnom rječju loop, a a sve naredbe koje se nalaze u njemu moraju biti upisane između vitičastih zagrada koje označavaju početak i kraj tog dijela programa. Unutar loop dijela nailazimo na dvije nove naredbe – digitalWrite i delay.

Naredba digitalWrite služi za uključivanje i isključivanje pojedinog izvoda. Ova naredba ima dva argumenta – izvod i stanje izvoda. Stanje izvoda može biti LOW ili HIGH. Stanje LOW isključuje izvod, odnosno na izvodu će  biti napon od 0V, a stanje HIGH uključuje izvod, odnosno na njemu će biti napon od 3.3V. Naredba digitalWrite(LED\_BUILTIN, HIGH) će uključiti svjetleću diodu na MKR1000 pločici, a digitalWrite(LED\_BUILTIN, LOW) će ju isključiti.

Naredba delay služi za zaustavljanje izvršavanja programa na određano vrijeme. Ova naredba ima samo jedan argument koji definira trajanje zaustavljanja u milisekundama. Npr. delay(1000) će zaustaviti izvođenje programa na 1000 milisekundi odnosno na jednu sekundu.

Ukoliko promjenimo iznos argumenta delay naredbe možemo ubrzati ili usporiti blinkanje. Pokušajte promjeniti na 500 milisekundi, kako slijedi:

// 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(500); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

delay(500); // wait for a second

}

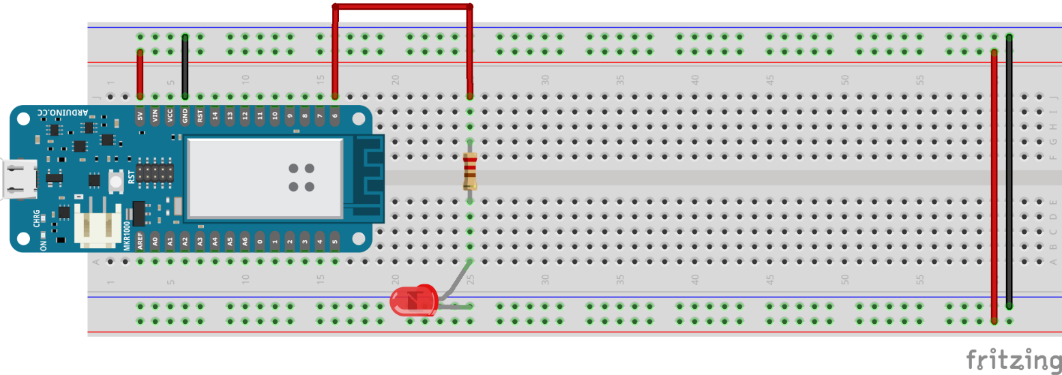
Kako bi se program počeo izvoditi na mikrokontroleru potrebno je prebaciti program na njega. To možete učiniti koristeći drugu ikonu iz trake s prečacima (okrugla ikona sa strelicom u desno) ili odabirom Uplad opcije iz Sketch izbornika.

**Aktivnosti za učenike**

*Spajanje jedne svjetleće diode*

Spojite jednu svjetleću diodu na izvod broj 6 Arduina i napravite program koji će upravljati svjetlećom diodom tako da ona blinka – neka bude uključena 300 milisekudni, pa isključena 300 milisekundi i neka se to ponavlja.

**Prikaz spajanja**

[](https://izradi.croatianmakers.hr/wp-content/uploads/2020/01/SpajanjeJedneDiode_bb.png)

**Arduino kod**

int LedCrvena = 6;

void setup() {

 pinMode(LedCrvena, OUTPUT);      //postavi izvod LedCrvena (6) kao izlazni

}

void loop() {

 digitalWrite(LedCrvena, HIGH);   //postavi izvod LedCrvena u st. visoko - uključi diodu

 delay(300);                     //čekaj 300 milisekundi

 digitalWrite(LedCrvena, LOW);   //postavi izvod LedCrvena u st. nisko - isključi diodu

 delay(300);                     //čekaj 300 milisekundi

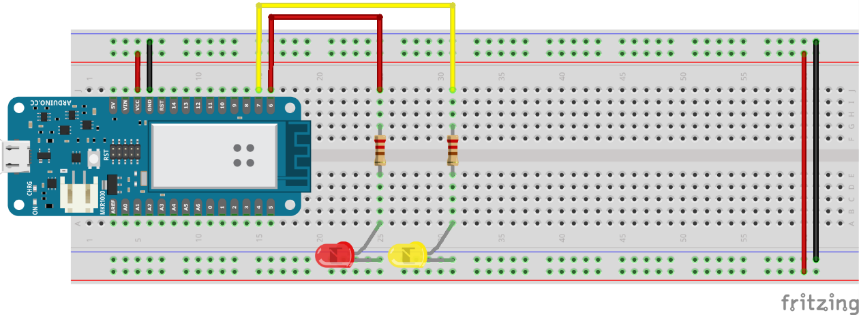
}

**Aktivnosti za učenike**

*Naizmjenično blinkanje dviju dioda*

Na Arduino spojite dvije svjetleće diode – jednu crvenu i jednu žutu, koristeći izvode 6 i 7. Napravite program da se diode naizmjenično uključuju. Na početku neka bude crvena uključena, a žuta isključena na pola sekunde te nakon toga neka bude crvena isključena, a žuta uključena na pola sekunde i neka se to ponavlja.

**Prikaz spajanja**

[](https://izradi.croatianmakers.hr/wp-content/uploads/2020/01/Naizmjenicno-blinkanje-dviju-dioda_bb-1.png)

**Arduino kod**

int LedCrvena = 6;

int LedZuta = 7;

void setup() {

 pinMode(LedCrvena, OUTPUT);     //postavi izvod LedCrvena (6) kao izlazni

 pinMode(LedZuta, OUTPUT);       //postavi izvod LedZuta (7) kao izlazni

}

void loop() {

 digitalWrite(LedCrvena, HIGH);  //postavi izvod LedCrvena u st. visoko - uključi diodu

 digitalWrite(LedZuta, LOW);     //postavi izvod LedZuta u st. nisko - isključi diodu

 delay(500);                     //čekaj 500 milisekundi

 digitalWrite(LedCrvena, LOW);   //postavi izvod LedCrvena u st. nisko - isključi diodu

 digitalWrite(LedZuta, HIGH);    //postavi izvod LedZuta u st. visoko - uključi diodu

 delay(500);                     //čekaj 500 milisekundi

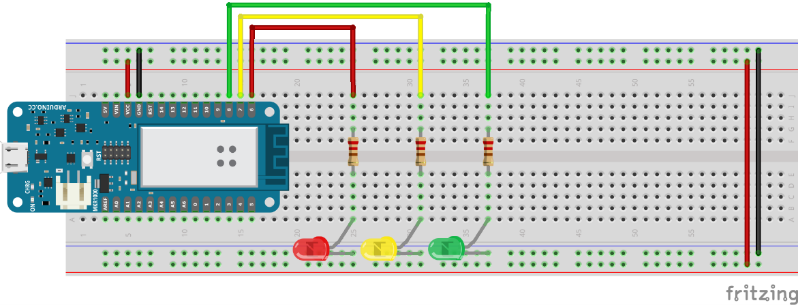
}

**Aktivnosti za učenike**

*Naizmjenično blinkanje dviju dioda*

Napravite model semafora za automobile. Na Arduino spojite tri svjetleće diode – crvenu, žutu i zelenu, na izvode 6, 7 i 8. Semafor mora raditi kako inače rade semafori u prometu. Na početku neka bude uključeno crveno svijetlo na tri sekunde. Potom neka svijetle crveno i žuto zajedno jednu sekundu. Nakon toga neka svijetli zeleno tri sekunde. Naposlijetku, neka svijetli samo žuto na jednu sekundu i neka se ciklus ponavlja ispočetka.

**Prikaz spajanja**

[](https://izradi.croatianmakers.hr/wp-content/uploads/2020/01/Model-semafora_bb-2.png)

**Arduino kod**

int LedCrvena = 6;

int LedZuta = 7;

int LedZelena = 8;

void setup() {

 pinMode(LedCrvena, OUTPUT);     //postavi izvod LedCrvena (6) kao izlazni

 pinMode(LedZuta, OUTPUT);       //postavi izvod LedZuta (7) kao izlazni

 pinMode(LedZelena, OUTPUT);     //postavi izvod LedZelena (8) kao izlazni

}

void loop() {

 //svijetli crveno 3 sekunde

 digitalWrite(LedCrvena, HIGH);  //uključi crveno svijetlo

 digitalWrite(LedZuta, LOW);     //isključi žuto svijetlo

 digitalWrite(LedZelena, LOW);   //isključi zeleno svijetlo

 delay(3000);                    //čekaj 3 sekunde

 //svijetle crveno i žuto 1 sekundu

 digitalWrite(LedCrvena, HIGH);  //uključi crveno svijetlo

 digitalWrite(LedZuta, HIGH);    //uključi žuto svijetlo

 digitalWrite(LedZelena, LOW);   //isključi zeleno svijetlo

 delay(1000);                    //čekaj 1 sekundu

 //svijetli zeleno 3 sekunde

 digitalWrite(LedCrvena, LOW);   //isključi crveno svijetlo

 digitalWrite(LedZuta, LOW);     //isključi žuto svijetlo

 digitalWrite(LedZelena, HIGH);  //uključi zeleno svijetlo

 delay(3000);                    //čekaj 3 sekunde

 //svijetli žuto 1 sekundu

 digitalWrite(LedCrvena, LOW);   //isključi crveno svijetlo

 digitalWrite(LedZuta, LOW);     //uključi žuto svijetlo

 digitalWrite(LedZelena, HIGH);  //isključi zeleno svijetlo

 delay(1000);                    //čekaj 1 sekundu

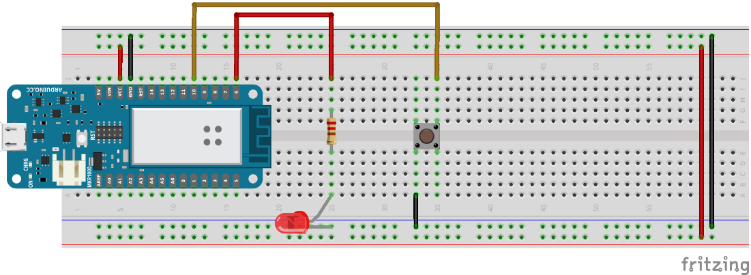
}

**Aktivnosti za učenike**

*Upravljanje jednom svjetlećom diodom*

Na Arduino spojite jedno tipkalo i jednu svjetleću diodu. Napišite program kojim ćete upravljati svjetlećom diodom na sljedeći način – kada je tipkalo pritisnuto neka dioda bude uključena, a kada je tipkalo otpušteno neka dioda bude isključena.

**Prikaz spajanja**

[](https://izradi.croatianmakers.hr/wp-content/uploads/2020/01/Upravljanje-jednom-svjetlecom-diodom_bb-1.png)

**Arduino kod**

int LedCrvena = 6;

int Tipkalo = 10;

int StanjeTipkala;

void setup() {

 pinMode(LedCrvena, OUTPUT);     //postavi izvod LedCrvena (6) kao izlazni

 pinMode(Tipkalo, INPUT\_PULLUP); //postavi izvod Tipkalo (10) kao ulazni, pull-up

}

void loop() {

 StanjeTipkala = digitalRead(Tipkalo); //očitaj stanje izvoda i pohrani u StanjeTipkala

 if (StanjeTipkala == LOW) {         //ako je tipkalo pritisnuto

   digitalWrite(LedCrvena, HIGH);    //uključi svjetleću diodu

 } else {                            //inače

   digitalWrite(LedCrvena, LOW);     //isključi svjetleću diodu

 }

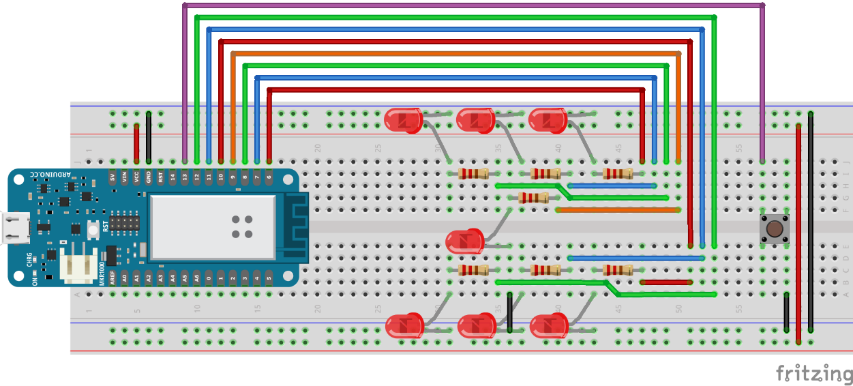
}

**Aktivnosti za učenike**

*Elektronička kockica*

Na mikrokontroler spojite sedam svjetlećih dioda i jedno tipkalo. Svjetleće diode spojite tako da izgledaju kao točkice na kockici za igre na sreću. Napišite program koji će na svjetlećim diodama prikazivati brojeve od 1 do 6 kako su oni inače prikazani na kockici. Promjena i nasumičnost brojeva postiže se na način da dok korisnik drži tipkalo program odbrojava od 1 do 6 velikom brzinom i prikazuje to na svjetlećim diodama. Kada korisnik otpusti tipkalo brojanje se zaustavlja i posljednji broj ostaje prikazan na svjetlećim diodama. Sljedeća slika prikazuje kombinacije mogućih dobivenih brojeva.

**Prikaz spajanja**

[](https://izradi.croatianmakers.hr/wp-content/uploads/2020/01/Elektronicka-kockica_bb-1.png)

**Arduino kod**

int led1 = 6;                      //definiraj led1 = 6

int led2 = 7;                      //definiraj led2 = 7

int led3 = 8;                      //definiraj led3 = 8

int led4 = 9;                      //definiraj led4 = 9

int led5 = 10;                      //definiraj led5 = 10

int led6 = 11;                      //definiraj led6 = 11

int led7 = 12;                      //definiraj led7 = 12

int tipkalo = 13;                   //definiraj tipkalo = 13

int brojac = 0;                    //definiraj brojač = 0

void setup() {

 pinMode(led1, OUTPUT);           //postavi izvod led1 kao izlazni

 pinMode(led2, OUTPUT);           //postavi izvod led2 kao izlazni

 pinMode(led3, OUTPUT);           //postavi izvod led3 kao izlazni

 pinMode(led4, OUTPUT);           //postavi izvod led1 kao izlazni

 pinMode(led5, OUTPUT);           //postavi izvod led2 kao izlazni

 pinMode(led6, OUTPUT);           //postavi izvod led3 kao izlazni

 pinMode(led7, OUTPUT);           //postavi izvod led3 kao izlazni

 pinMode(tipkalo, INPUT\_PULLUP);  //postavi izvod tipkalo kao ulazni

}

void iskljucisveled() {

 digitalWrite(led1, LOW);         //isključi LED diodu 1

 digitalWrite(led2, LOW);         //isključi LED diodu 2

 digitalWrite(led3, LOW);         //isključi LED diodu 3

 digitalWrite(led4, LOW);         //isključi LED diodu 4

 digitalWrite(led5, LOW);         //isključi LED diodu 5

 digitalWrite(led6, LOW);         //isključi LED diodu 6

 digitalWrite(led7, LOW);         //isključi LED diodu 7

}

void loop() {

 if (digitalRead(tipkalo) == LOW) { //ukoliko je tipkalo pritisnuto

   brojac = brojac + 1;           //dodaj 1 na brojač

   if (brojac > 6) {              //ako je brojač veći od 6

     brojac = 1;                  //vrati brojač na 1

   }

 }

 if (brojac == 1) {               //ukoliko je brojač = 1

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led4, HIGH);      //prikaz broja 1

 } else if (brojac == 2) {        //ukoliko je brojač = 2

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led1, HIGH);      //prikaz broja 2

   digitalWrite(led7, HIGH);      //prikaz broja 2

 } else if (brojac == 3) {        //ukoliko je brojač = 3

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led3, HIGH);      //prikaz broja 3

   digitalWrite(led4, HIGH);      //prikaz broja 3

   digitalWrite(led5, HIGH);      //prikaz broja 3

 } else if (brojac == 4) {        //ukoliko je brojač = 4

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led1, HIGH);      //prikaz broja 4

   digitalWrite(led3, HIGH);      //prikaz broja 4

   digitalWrite(led5, HIGH);      //prikaz broja 4

   digitalWrite(led7, HIGH);      //prikaz broja 4

 } else if (brojac == 5) {        //ukoliko je brojač = 5

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led1, HIGH);      //prikaz broja 5

   digitalWrite(led3, HIGH);      //prikaz broja 5

   digitalWrite(led4, HIGH);      //prikaz broja 5

   digitalWrite(led5, HIGH);      //prikaz broja 5

   digitalWrite(led7, HIGH);      //prikaz broja 5

 } else if (brojac == 6) {        //ukoliko je brojač = 6

   iskljucisveled();              //isključi sve LED diode

   digitalWrite(led1, HIGH);      //prikaz broja 6

   digitalWrite(led2, HIGH);      //prikaz broja 6

   digitalWrite(led3, HIGH);      //prikaz broja 6

   digitalWrite(led5, HIGH);      //prikaz broja 6

   digitalWrite(led6, HIGH);      //prikaz broja 6

   digitalWrite(led7, HIGH);      //prikaz broja 6

 } else {                         //ukoliko je brojač = 0

   iskljucisveled();              //isključi sve LED diode

 }

}

**Aktivnosti za učenike**

*Senzori*

Senzor je uređaj koji mjeri fizikalnu veličinu (npr. temperaturu, vlažnost zraka, tlak, broj okretaja motora) i pretvara je u signal pogodan za daljnju obradu (najčešće u električni signal). Senzori se koriste u svakodnevnim predmetima, kao što su tipkala lifta (osjetilni senzor) ili lampa koja se pali na dodir. Postoje i mnoge druge aplikacije o kojima ljudi uopće ne razmišljaju. Neke od primjena uključuju automobile, letjelice, medicinu, proizvodnju i robotiku.

Senzore dijelimo prema:

* Mjerenoj veličini (mehanički, toplinski, magnetski itd.)
* Načinu pretvorbe (piezoelektrični, fotoelektrični i termoelektrični)
* Području primjene (medicina, kućanstvo, svemirska istraživanja itd.)
* Materijalu (vodič, poluvodič)
* Tehnologiji izrade (npr. nanotehnologija)
* Načinu rada (aktivni i pasivni)
* Složenosti (samostalne i nesamostalne jedinice)
* Vrsti izlaznog signala (digitalni i analogni)





**Aktivnosti za učenike**

*Blynk aplikacija*

<https://izradi.croatianmakers.hr/project/instalacija-blynk-aplikacije-i-prvi-iot-projekt/>

Izvor informacija i aktivnosti: <https://izradi.croatianmakers.hr/edukacija/>