

Arduino

UVOD U ARDUINO SVIJET

Što je Arduino?

Arduino je platforma za učenje programiranja i korištenje mikrokontrolera. Nastala je 2005. godine.

Arduino platforma je skup elektroničkih i programskih dijelova koji se mogu jednostavno povezati 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 kodovi programa besplatni i dostupni svima za preuzimanje i modificiranje sa službene stranice Arduino platforme – www.arduino.cc.

Mikrokontroler – što je to?

Najjednostavnije rečeno mikrokontroler je računalo u malom.

Mikrokontroler je malo računalo smješteno u jednom integriranom sklopu.

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

Mikrokontroler ATMEGA 328P u DIP i SMD kućištu

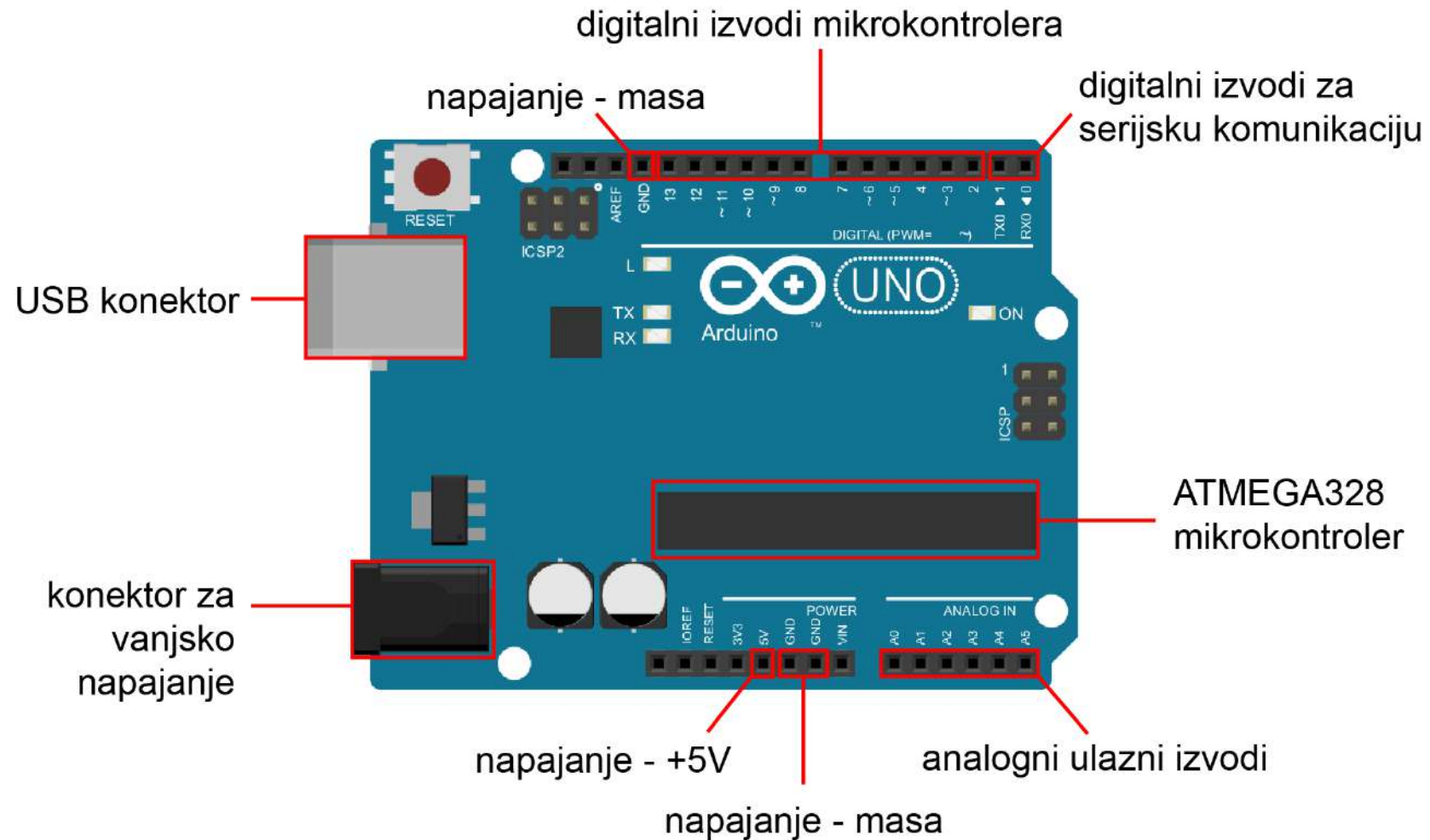


Arduino UNO pločka



PUBLIC

Arduino UNO pločica



Arduino IDE

Za programiranje mikrokontrolera koristi se Arduino IDE programsko sučelje.

Program se može besplatno preuzeti na www.arduino.cc.

Aktualna verzija programa je 1.8.10.

Nakon preuzimanja programa potrebno je pokrenuti instalaciju ... prihvatiti GNU Lesser General Public License i sa par klika na *Next*, *Install* i *Close* Arduino programsko sučelje uspješno je instalirano.

Na radnoj površini pojavi se prečac za pokretanje Arduino programa.

Pokretanjem prečaca otvara se Arduino programsko sučelje, prikazano na sljedećem slajdu, zajedno sa opisom.

Naziv programa / sketch-a i verzija korištenog Arduino sučelja →

Standardni izbornik →

Izbornik s prečacima →

Ikone u izborniku prečaca:

Provjeri kod programa

Učitaj program u mikrokontroler

Novi program

Otvori postojeći program

Spremi program

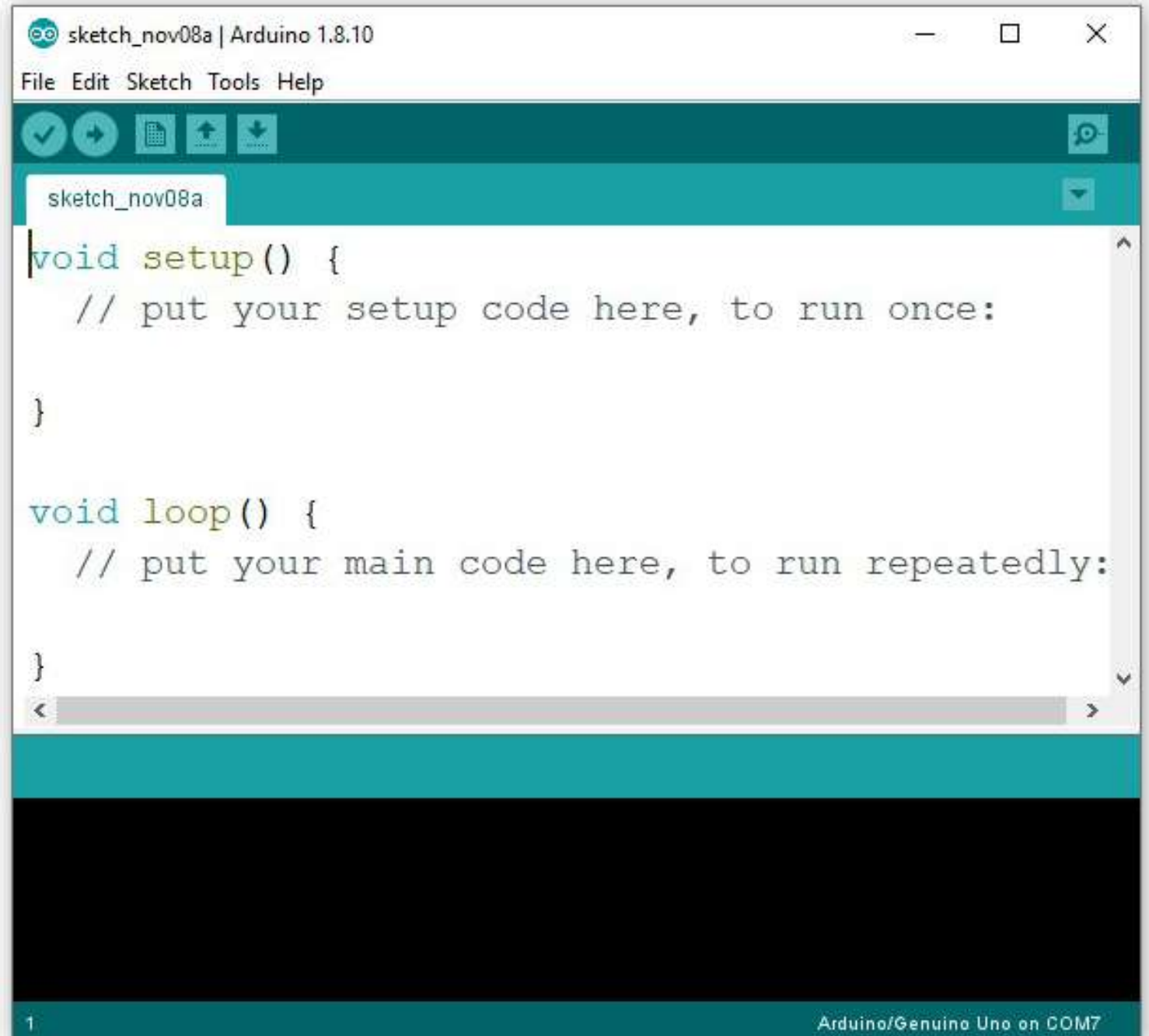
Prikaz komunikacije pomoću serijskog porta

Dio za pisanje i uređivanje koda →

Prikaz statusa programa →

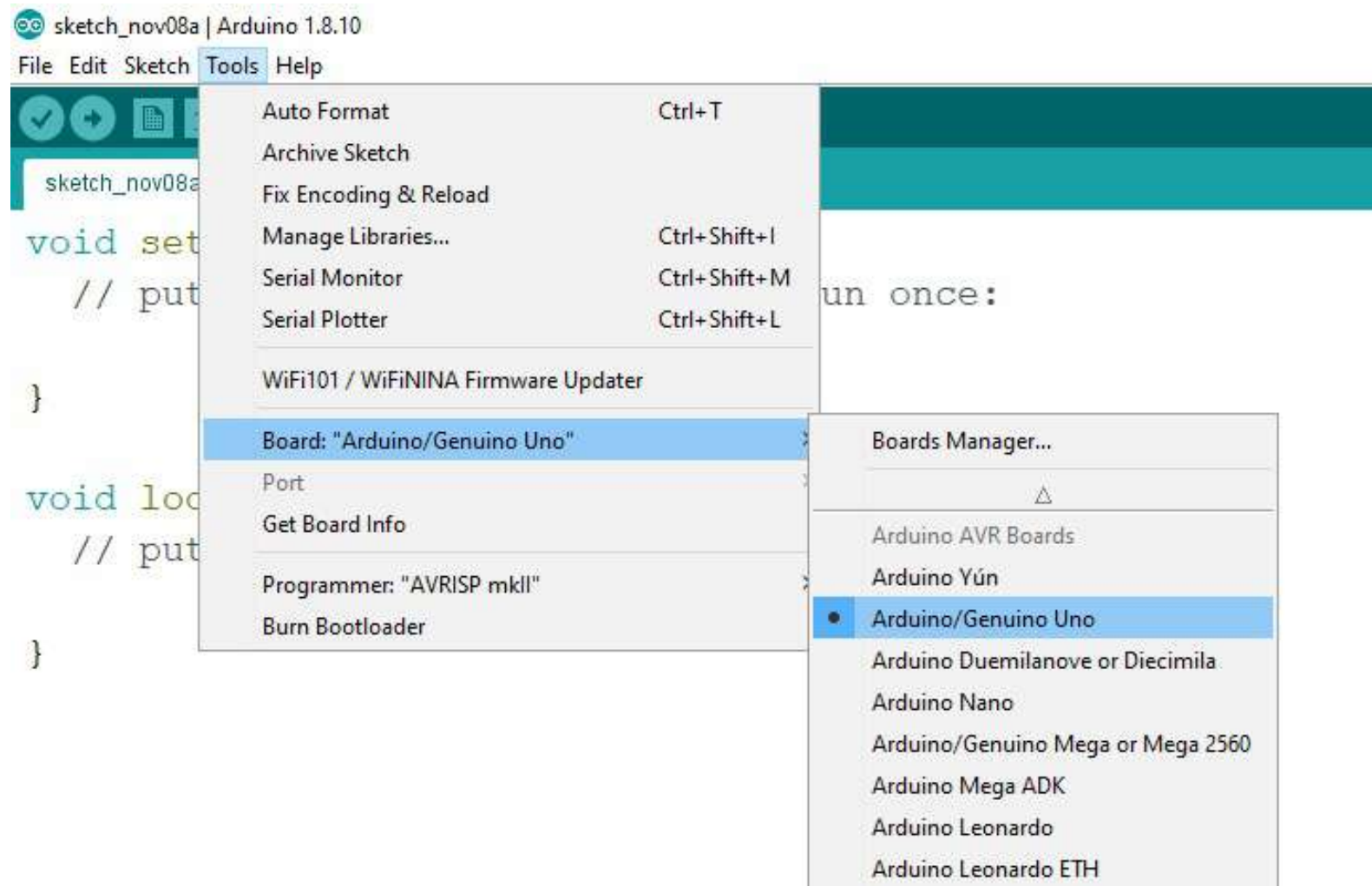
Konzola →

Odabrana Arduino pločica i naziv serijskog porta →

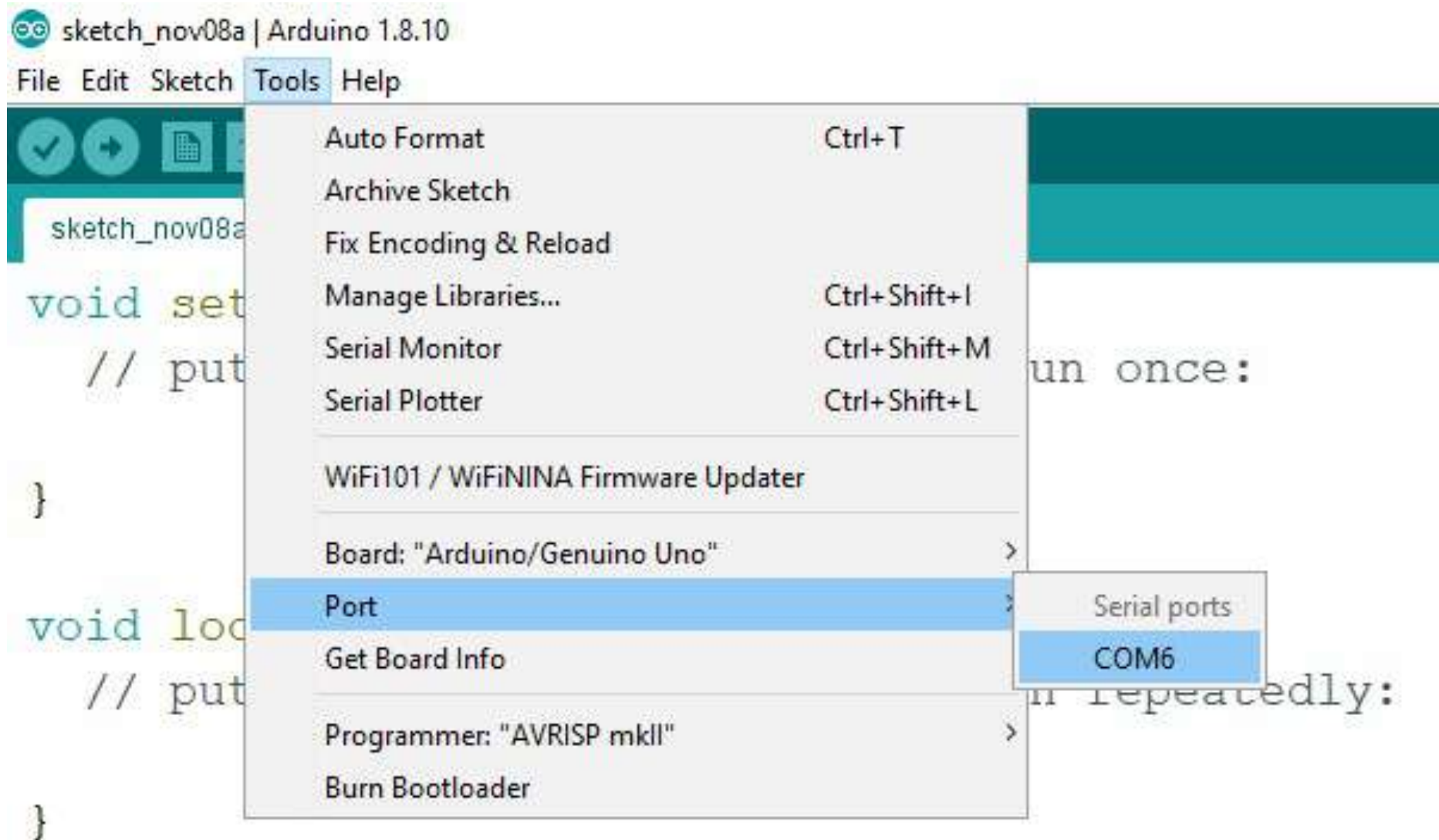


Nakon spajanja računala i Arduino pločice USB kablom, prije početka rada potrebno je podesiti neke parametre Arduino IDE sučelja.

Odabrati Arduino pločicu ...

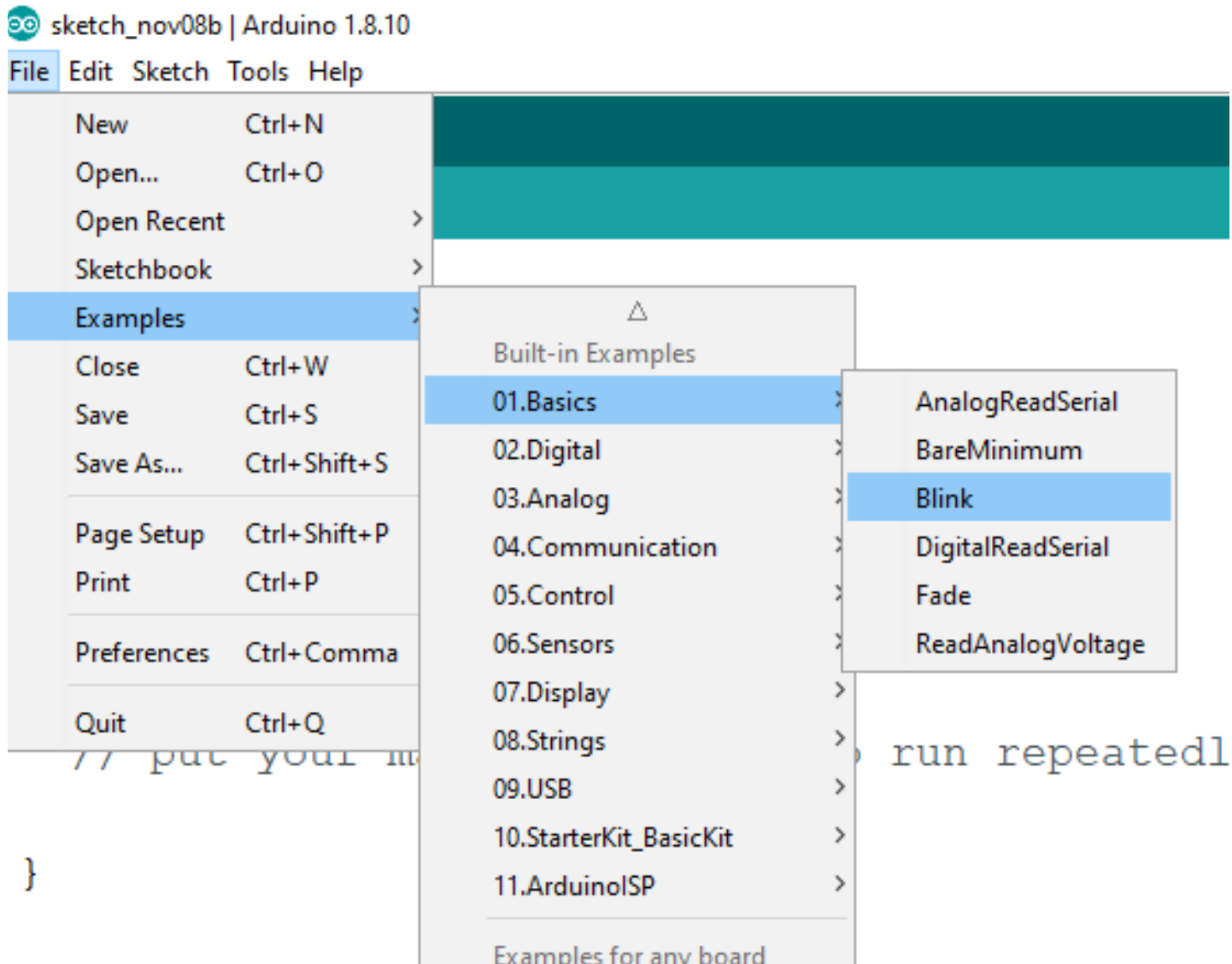


... i odabrati port na koji je priključena Arduino pločica.



Pokretanje najjednostavnijeg programa

Blinkanje LED diode



Nakon spajanja računala i Arduino pločice, na Arduino pločici svijetle dvije SMD LED diode, zelena i žuta. Zelena označava da je pločica priključena na napon. Program Blink, omogućuje provjeru rada/provjeru komunikacije između Arduino programskog sučelja i Arduino pločice.

→ Žuta LED dioda blinka, tj. Naizmjenice se pali i gasi svake sekunde.



Blink

/*

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>

*/

Početak programa je veliki komentar.

Iz komentara saznajemo što program radi, tko je autor te specifičnosti raznih Arduino pločica glede aktiviranja ugrađene LED diode, koja se može iskoristiti za brzo i jednostavno testiranje Arduino pločice i komunikacije s programskim sučeljem. Navedeno je da je to program otvorenog koda i gdje se može naći taj kod.

Program se nastavlja na drugom slajdu

... nastavak programa

<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  
}
```



Dijelovi Arduino programa

Prvi dio – govori mikrokontroleru na koji izvod je što spojeno. U prvom dijelu se definiraju i potrebne (globalne) varijable.

→ *U programu Blink tog dijela nema ... jer je korišten digitalni izvod pin13 koji je ujedno i LED_BUILTIN izvod*

Drugi dio (setup dio) – govori mikrokontroleru na koji način se koriste pojedini izvodi (kao ulazni ili kao izlazni) te se u tom dijelu piše onaj dio koda koji se izvodi samo jednom na početku, pri pokretanju mikrokontrolera

Treći dio (loop dio) – u trećem dijelu piše se kod koji se izvodi u nedogled, tj. dok se ne prekine kružno izvođenje.

→ *Prekid izvođenja programa nastupa kad se isključi napajanje ili se u mikrokontroler pošalje novi program, neki novi program (koji radi nešto drugo) ili prazan program (koji ne radi ništa).*

Komentari u programu

Komentar može biti bilo gdje u programu.

Mikrokontroleru ne trebaju komentari u programu.

Komentari služe samo ljudima.

Autoru programa da se lakše snalazi dok piše program, a pogotovo kad nakon nekog vremena ide analizirati/modificirati neki stari program da se prisjeti što je tada radio i kojom logikom.

Novim korisnicima da se lakše snađu kod analize postojećeg programa i da lakše napišu neke izmjene, dopune, ... → open-source programi

- U više redova - započinje sa */** u prvom, a završava sa **/* nakon zadnjeg
Sve između */** i **/* je komentar ... *u svjetlo sivoj boji*
- U jednom redu sve iza dviju kosih crta *//* smatra se komentarom
... *u tamno sivoj boji*

void setup () {}

U drugom dijelu, u **void setup** () dijelu, između vitičastih zagrada,

- za postavljanje načina rada pojedinog izvoda korištena je naredba:

pinMode (naziv izvoda, način rada)

naziv izvoda: *LED_BUILTIN*

način rada: OUTPUT

Napomena: iza svake naredbe dolazi točka zarez.

pinMode (*LED_BUILTIN*, OUTPUT);

void loop () {}

U trećem dijelu, u **void loop** () dijelu, između vitičastih zagrada,

- za uključivanje i isključivanje LED diode koristi se naredba:

digitalWrite (naziv izvoda, stanje);

naziv izvoda: *LED_BUILTIN*

stanje: HIGH – *LED dioda svijetli*

LOW - *LED dioda je ugašena*

- koliki period vremena će trajati pojedino stanje definirano je naredbom

delay (vrijeme);

Argument **vrijeme** izražava se u milisekundama.

Naredba **delay** (vrijeme) koristi se za zaustavljanje rada, odnosno, čekanje određenog vremenskog perioda.

SERIJSKA KOMUNIKACIJA ARDUINA S RAČUNALOM

Serijsku komunikaciju koristimo kad želimo da mikrokontroler ispiše neku poruku ili podatak na ekranu računala.

Ako želimo ostvariti tu komunikaciju u programu „žrtvujem” dva digitalna izvoda. To su nulti i prvi digitalni izvod. Označeni su brojem 0 i 1 te dodatno sa RX i TX.

Korištene naredbe:

Serial.begin (9600); // inicijalizira serijsku komunikaciju

Serial.println („Tekst unutar navodnika se ispisuje”);

// Nakon ispisa kursor se postavlja u novi red

// Naredba **Serial.print** („Tekst”); ispisala bi riječ Tekst i kursor bi ostao u istom redu. Slijedeća naredba **Serial.print()** bi se nadovezala na prethodno ispisanu riječ.

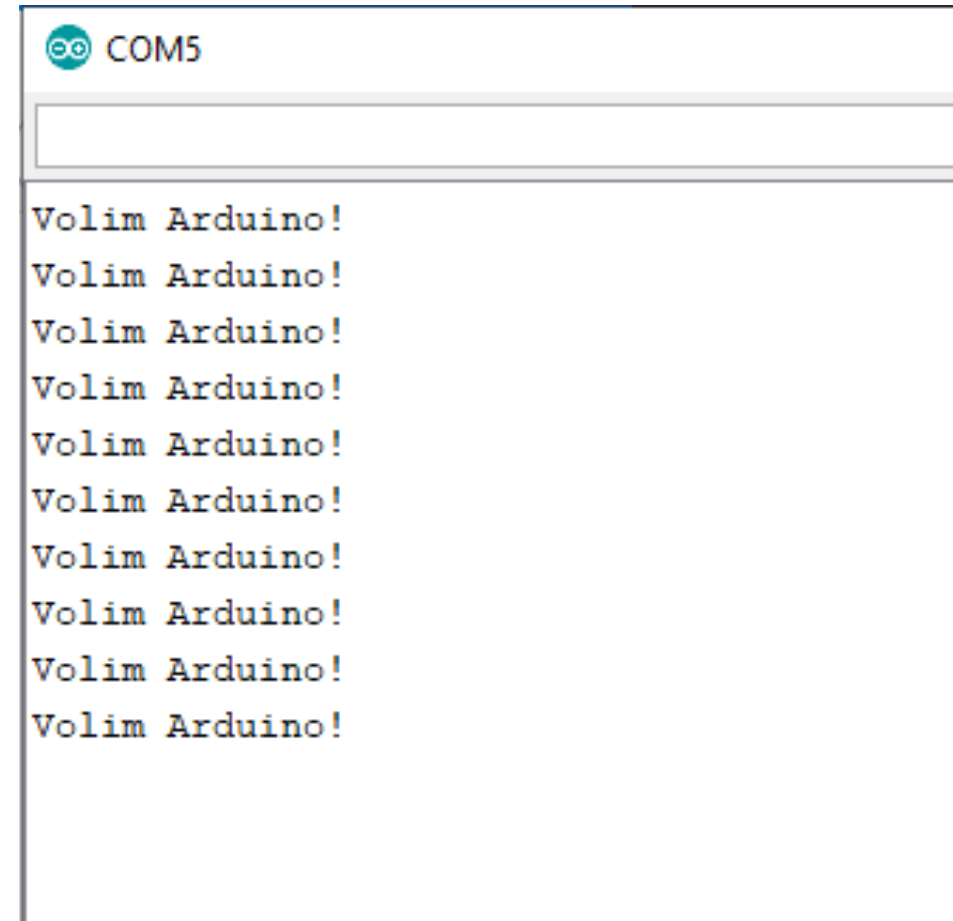
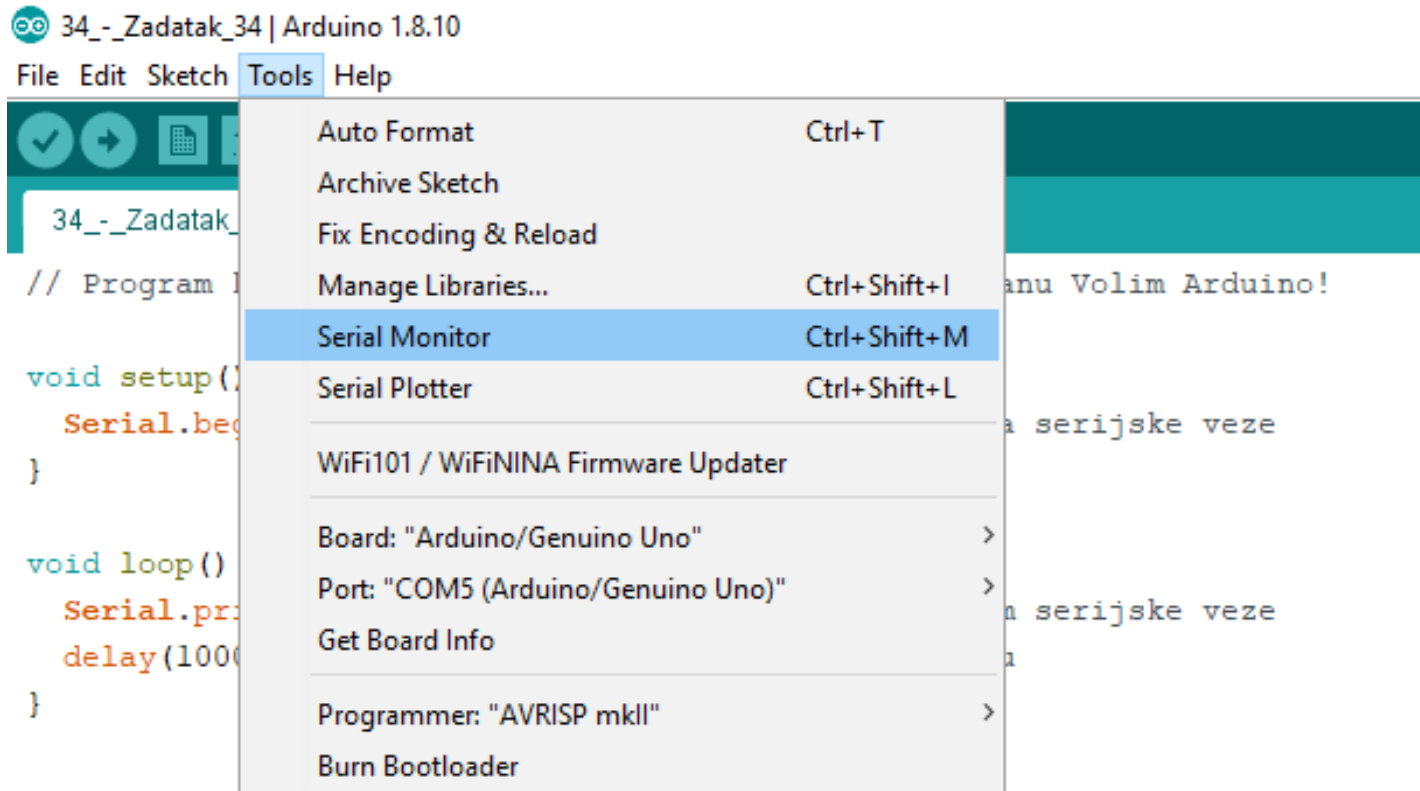
Zadatak: Napišite program koji će svake sekunde putem serijske veze poslati poruku "Volim Arduino!" na računalo. Pomoću Serial Monitor alata pogledajte dobivene podatke.

Rješenje:

```
void setup() { // naredbe unutar setup-a se izvode jednom
    Serial.begin (9600); // inicijalizacija serijske veze
                        // brzina komunikacije je 9600 bauda
}
void loop() { // naredbe se unutar loop-a izvode beskonačno puta
              // svake sekunde se ispisuje poruka u novom redu
    Serial.println ("Volim Arduino!"); // pošalji putem serijske veze
    delay(1000); // čekaj 1000 milisekundi / jednu sekundu
}
```

Aktiviranje prikaza na ekranu:

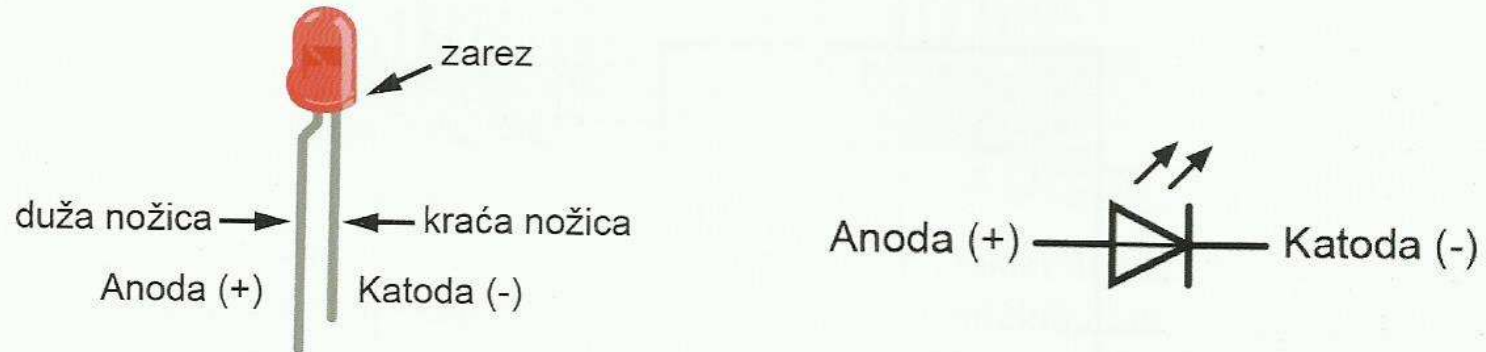
1. Klikom na simbol povećala (Prikaz komunikacije pomoću serijskog porta) u Izborniku s prečacima
2. Aktiviranjem Serial Monitor-a u padajućem izborniku Tools



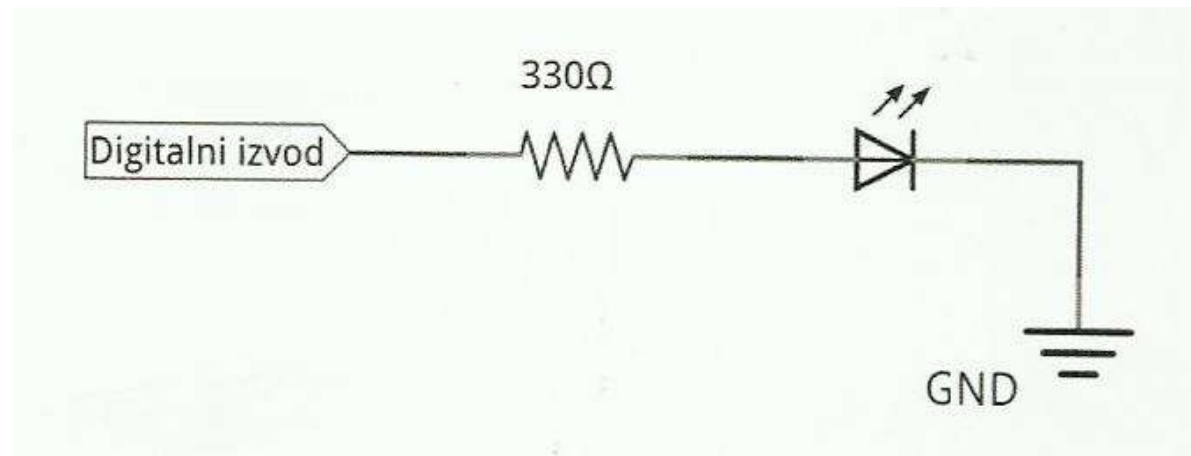
Napomena: Korištenje Serial Monitor-a jako je korisno kod pisanja složenijih programa. Omogućava praćenje stanja pojedinih varijabli, može se pratiti tijekom odvijanja programa, utvrditi da li program izvršava određene naredbe u programu, da li ulazi u petlju i slično ...

Spajanje prve elektroničke komponente na mikrokontroler

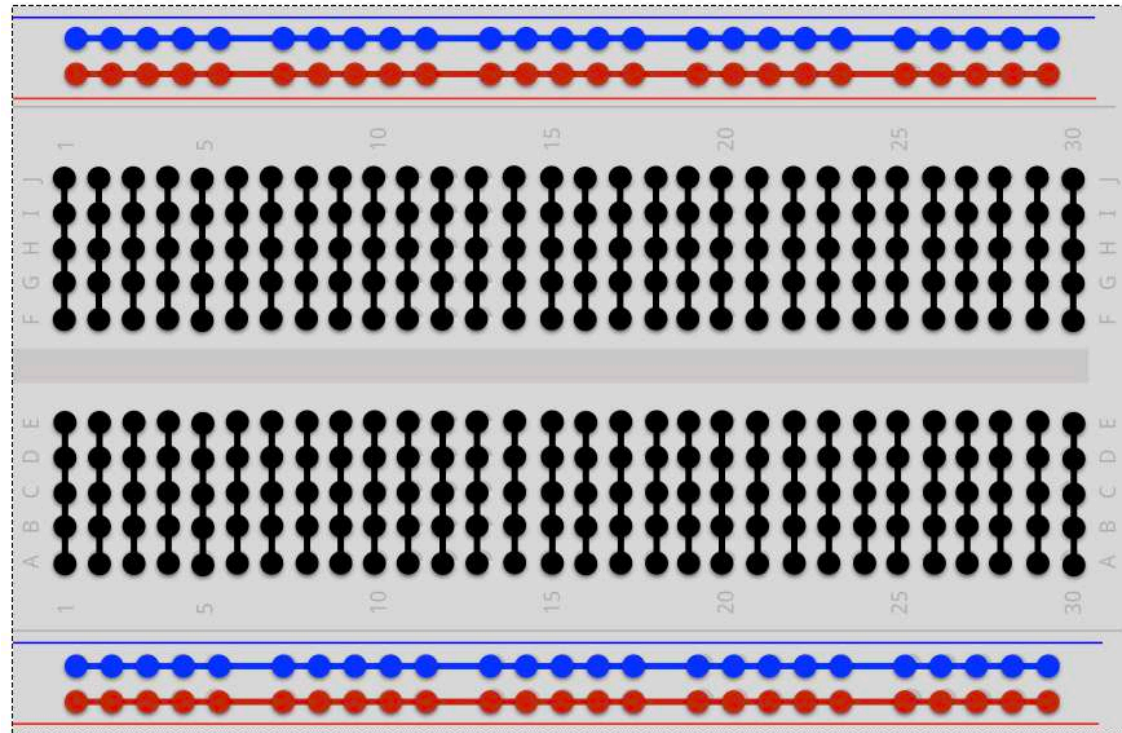
Svjetleća dioda – izgled i simbol



Shema spajanja led diode i otpornika na mikrokontroler

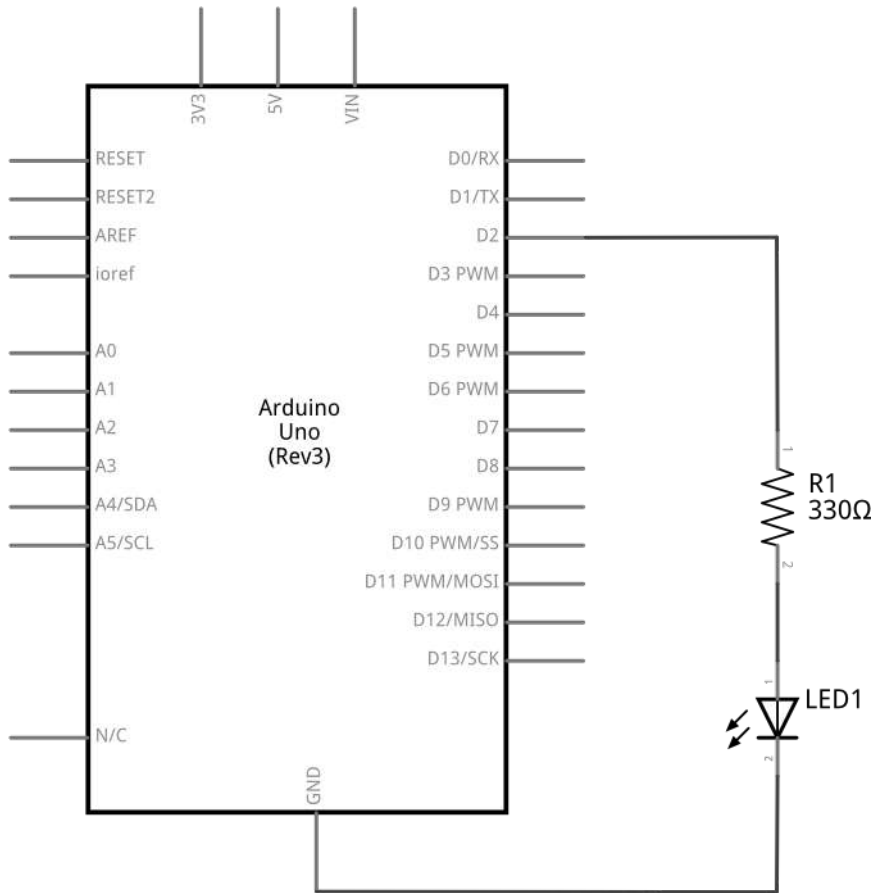


Izgled i veze na maloj eksperimentalnoj pločici

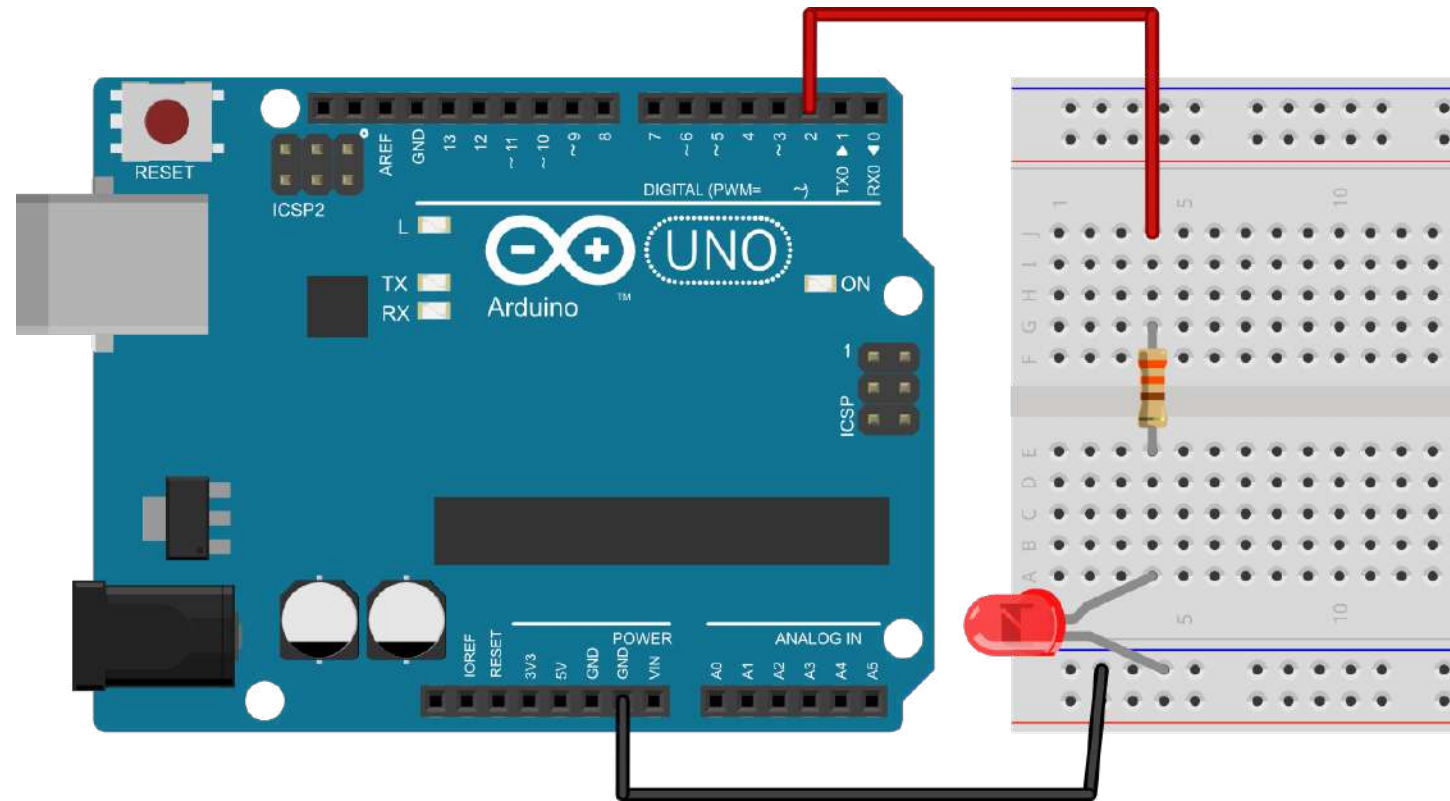


Napomena: Kod nekih velikih eksperimentalnih pločica plava i crvena linija mogu biti prepolovljene te ih je potrebno povezati mostom.

Zadatak 1. Na mikrokontroler spojite jednu svjetleću diodu. Napišite program koji će diodu uključivati i isključivati u intervalima od pola sekunde kako biste postigli efekt blinkanja.



Električna shema



Shema spajanja

Zadatak1: Rješenje

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

```
void setup() {  
    pinMode(led, OUTPUT); //postavi izvod 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  
}
```

Nadopuna prvog zadatka: Dodajte naredbe koje će na ekranu kontinuirano ispisivati trenutno stanje LED diode.

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

```
void setup() {  
    pinMode(led, OUTPUT); // postavi izvod led kao izlazni  
    digitalWrite(led, LOW); // isključi LED diodu - početno stanje  
    Serial.begin(9600); // inicijalizacija serijske veze  
}
```

```
void loop() {  
    digitalWrite(led, HIGH); // uključi LED diodu  
    delay(500); // čekaj 500 ms - pola sekunde  
    Serial.println("LED dioda svjetli.");  
    digitalWrite(led, LOW); // isključi LED diodu  
    delay(500); // čekaj 500 ms - pola sekunde  
    Serial.println("LED dioda ugašena.");  
}
```

Kraj uvodnog dijela.