Programabilni uređaji i objektno orjentisano programiranje

Uvod

Programabilni uređaji i objektno orjentisano programiranje

- Predmetni nastavnik:
  - Prof. dr Vesna Popović-Bugarin – kabinet 322
    - Konsultacije četvrtkom od 13:00-15:00h
    - Za kraće konsultacije koristiti e-mail: pvesna@ucg.ac.me
- Saradnici
  - mr Miloš Brajović Računske vježbe + Lab.
  - mr Nikola Bulatović Računske vježbe + Lab

CILJEVI PREDMETA

- Upoznavanje i ovladavanje programabilnim uređajima;
- objektno-orjentisanim programiranjem;
- vizuelnim programskim alatima.

ISHODI UČENJA

Kroz polaganje ovog ispita student će da:

- Ovlada programabilnim platformama tipa Arduino (Raspberry), komunikacijom sa njima, programiranjem, upravljanjem.
- Ovlada objektno orjentisanim programiranjem, pojmom klase, klasinim interfejsom, metodima klase, prijateljskim funkcijama i klasama, zajedničkim podacima i funkcijama za klasu, nasljeđivanjem i obradom izuzetaka.

STRUKTURA KURSA (I dio)

I nedjelja	Uvod. Osnovne karakteristike i mogućnosti programabilnih platformi. Hardverski interfejsi.
II nedjelja	Struktura programa na platformama, jednostavne aplikacije. Rad sa portovima, komunikacija sa periferijerima i drugim uređajima.
III nedjelja	Dizajniranje jednostavnih kolaborativnih rješenja u sistemima sa programabilnim platformama. Povezivanje sistema sa bazama podataka, udaljenim čvorovima, koncept upravljanja.
IV nedjelja	Uvod u objektno orjentisano programiranje (OOP) Koncepti OOP. Razlike C i C++.
V nedjelja	**Odbrana i prezentacija mini projekata. (21. mart 2019.)**

STRUKTURA KURSA (II dio)

VI nedjelja	Klasa i klasin interfejs. Konstruktor, inspektori i mutatori.
VII nedjelja	Metodi članovi klase, statički djelovi interfejsa klase. Konstruktor kopije
VIII nedjelja	Prijateljske funkcije i klase. Pokazivači na članove klase.
IX nedjelja	Preklapanje operatora (osnovni binarni i unarni operatori).
X nedjelja	Preklapanje operatora (napredne opcije);
XI nedjelja	**II kolokvijum (09. maj 2019.)**
XII nedjelja	Nasljeđivanje.
XIII nedjelja	Šabloni

STRUKTURA KURSA (III dio)

XV nedjelja	Obrada izuzetaka *Popravni kolokvijum* ***(30. maj 2019.)***
Završni ispit (po posebnom rasporedu)	

Opterećenje studenata

- **Nedjeljno opterećenje studenata**:

  6 kredita x 40/30 = 8 sati

- Struktura:
  - 2 časa predavanja
  - 1 čas računskih vježbi
  - 2 časa laboratorijskih vježbi
  - 3 sata samostalnog rada
- Osnovna literatura:
  - D. Milićev: "Objektno orjentisano programiranje na programskom jeziku C++", Mikro knjiga 1998.
  - dodatni materijali u vidu prezentacija, koji će biti dostupni studentima preko sajta ETF-a.

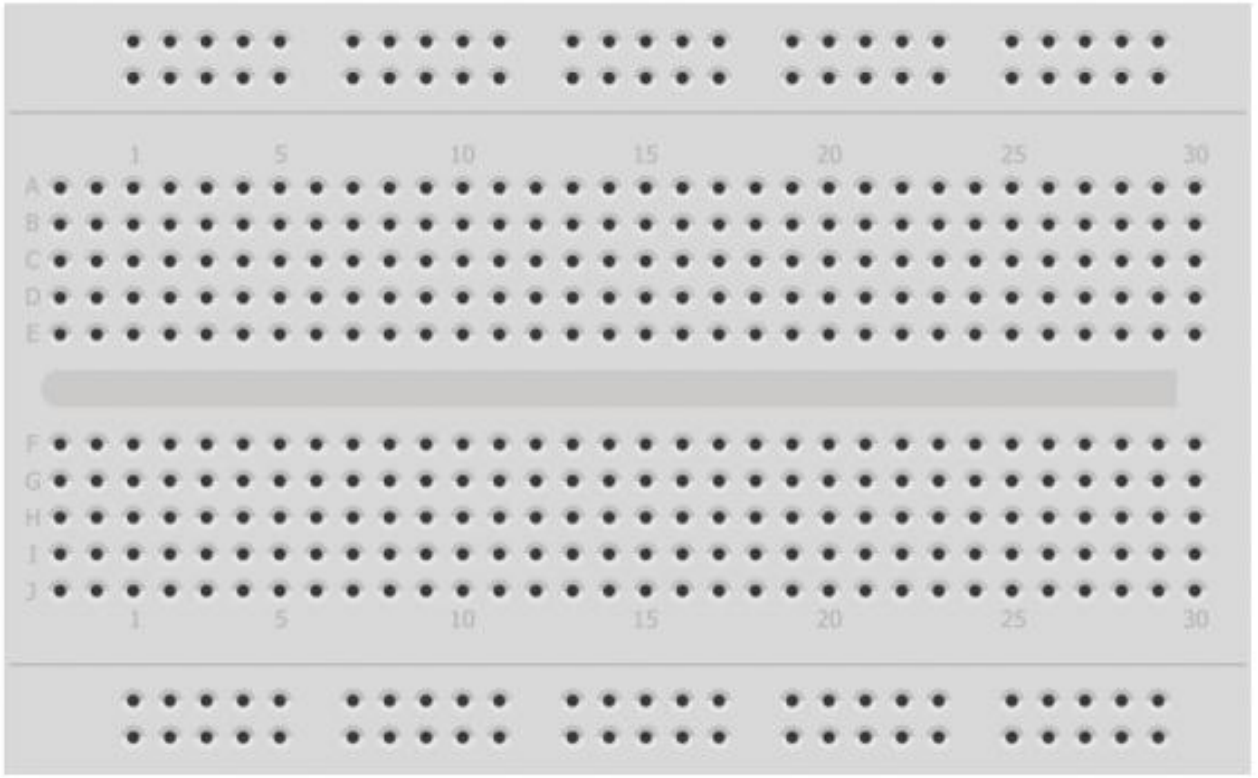
Provjera znanja

- **Načini provjeravanja znanja**:
  - Miniprojekat: 25 poena
  - Laboratorijske vježbe: 10 poena
  - Kolokvijum: 20 poena
  - Završni ispit: 45 poena – radi se u računarskoj sali
- Ispit je položen sa 50 i više poena u ukupnom zbiru.

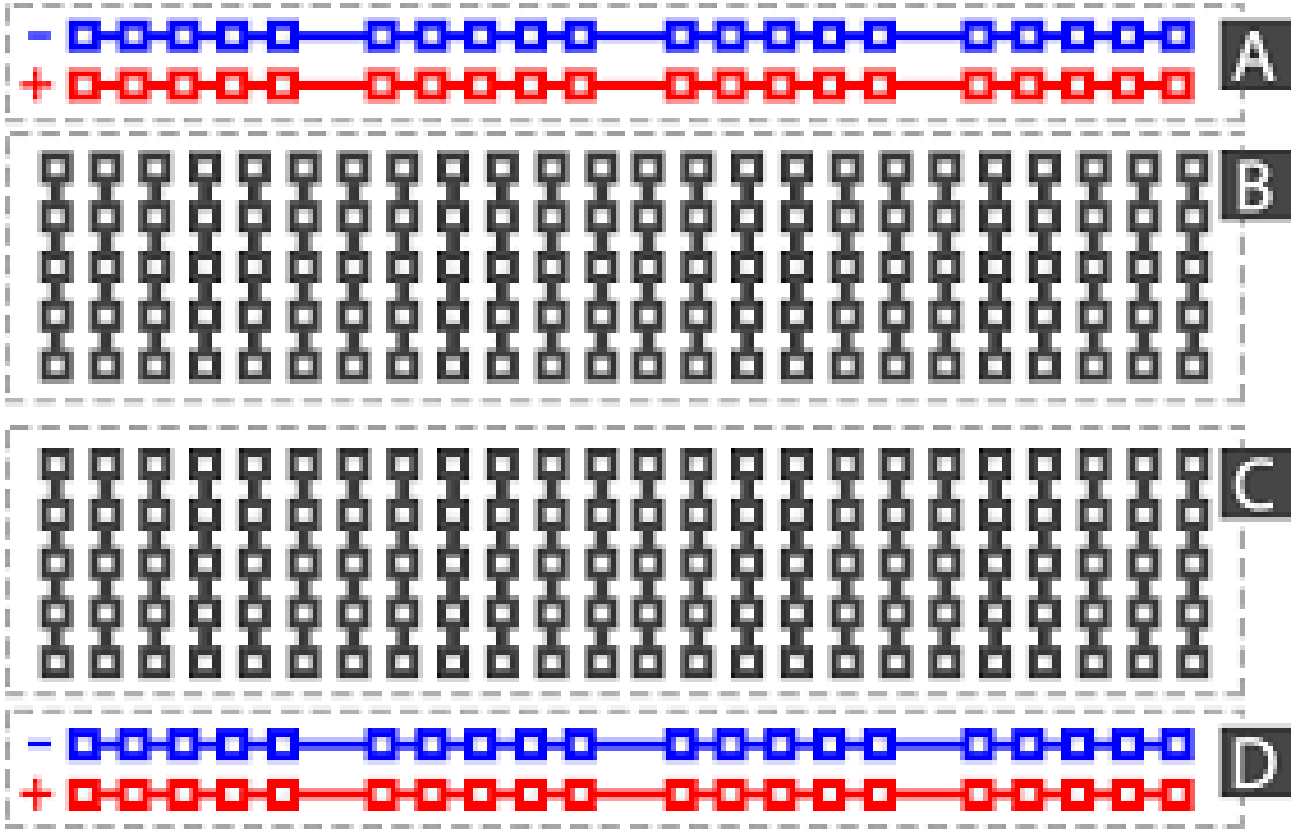
Programabilne platforme - potreba

- Ne tako davno, rad na hardveru je značio sastavljanje električnih kola od nule, korišćenjem stotine različitih komponenti poput otpornika, kondenzatora, kalemova, tranzistora ...
- Svako električno kolo je bilo povezano „žicama“ kako bi se osposobilo za određenu vrlo specifičnu namjenu.
- Promjene su zahtevale sječu „žica“ (provodnika), spojeva za lemljenje i još mnogo toga.
- Sa pojavom digitalnih tehnologija i mikroprocesora, ove funkcionalnosti, koje su nekada bile implementirane žicama, zamijenjene su programima.
- Softver je lakše modifikovati od hardvera. Sa nekoliko pritisaka na tipke, možete radikalno promeniti logiku uređaja i isprobati dve ili tri verzije u isto toliko vremena koliko bi vam trebalo da spajate par otpornika.

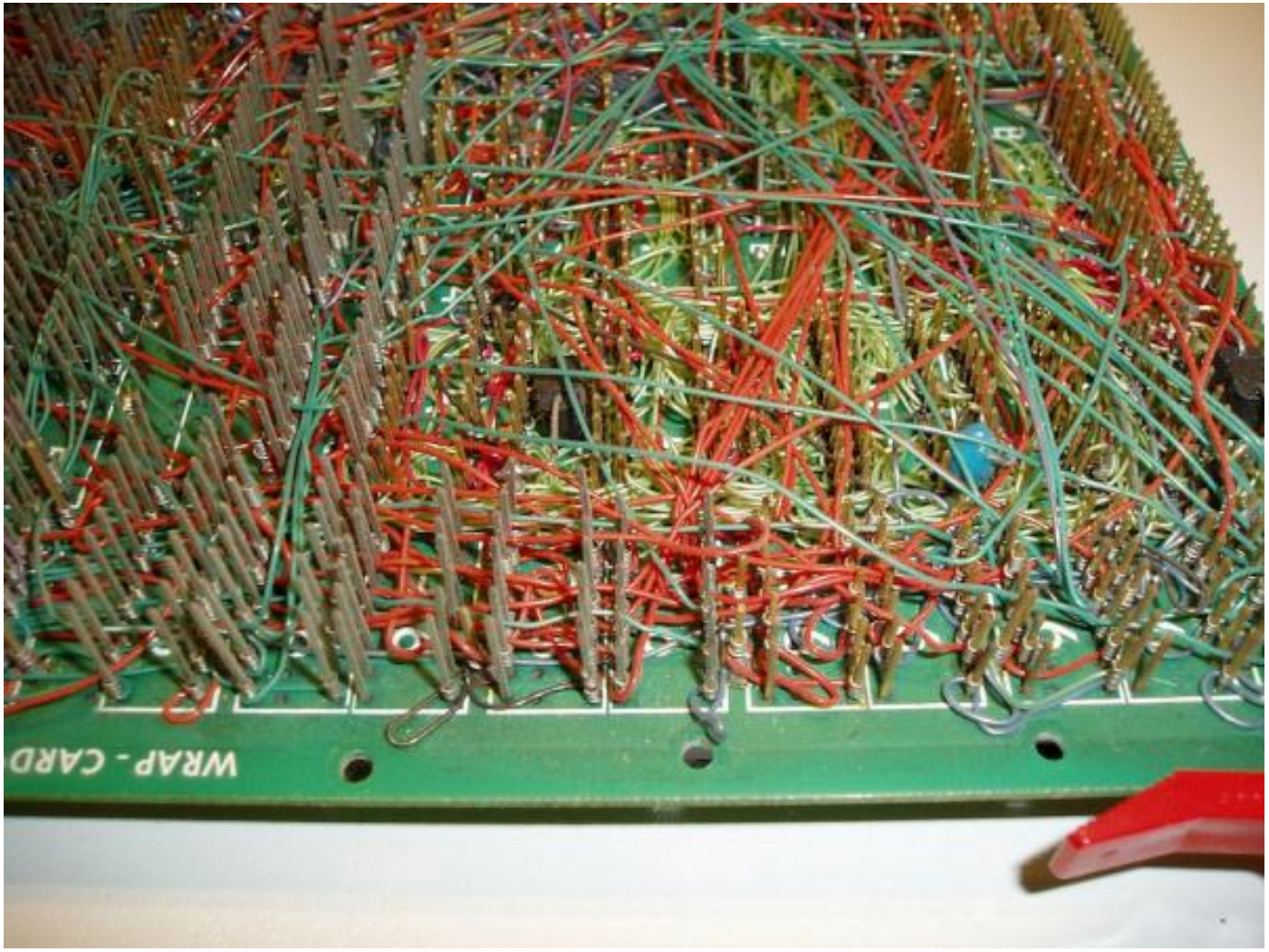
Breadboard – štampana ploča



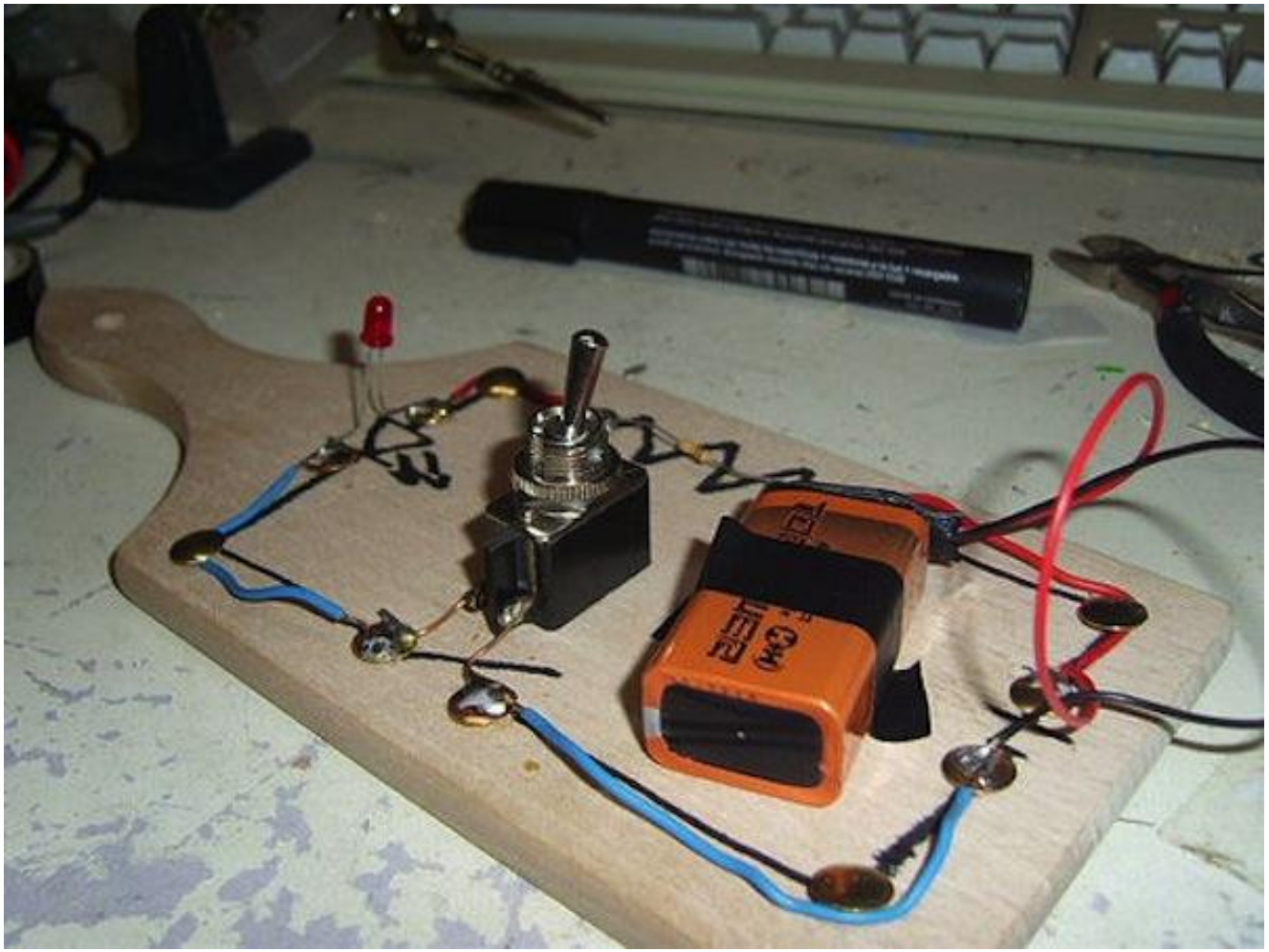
Breadboard



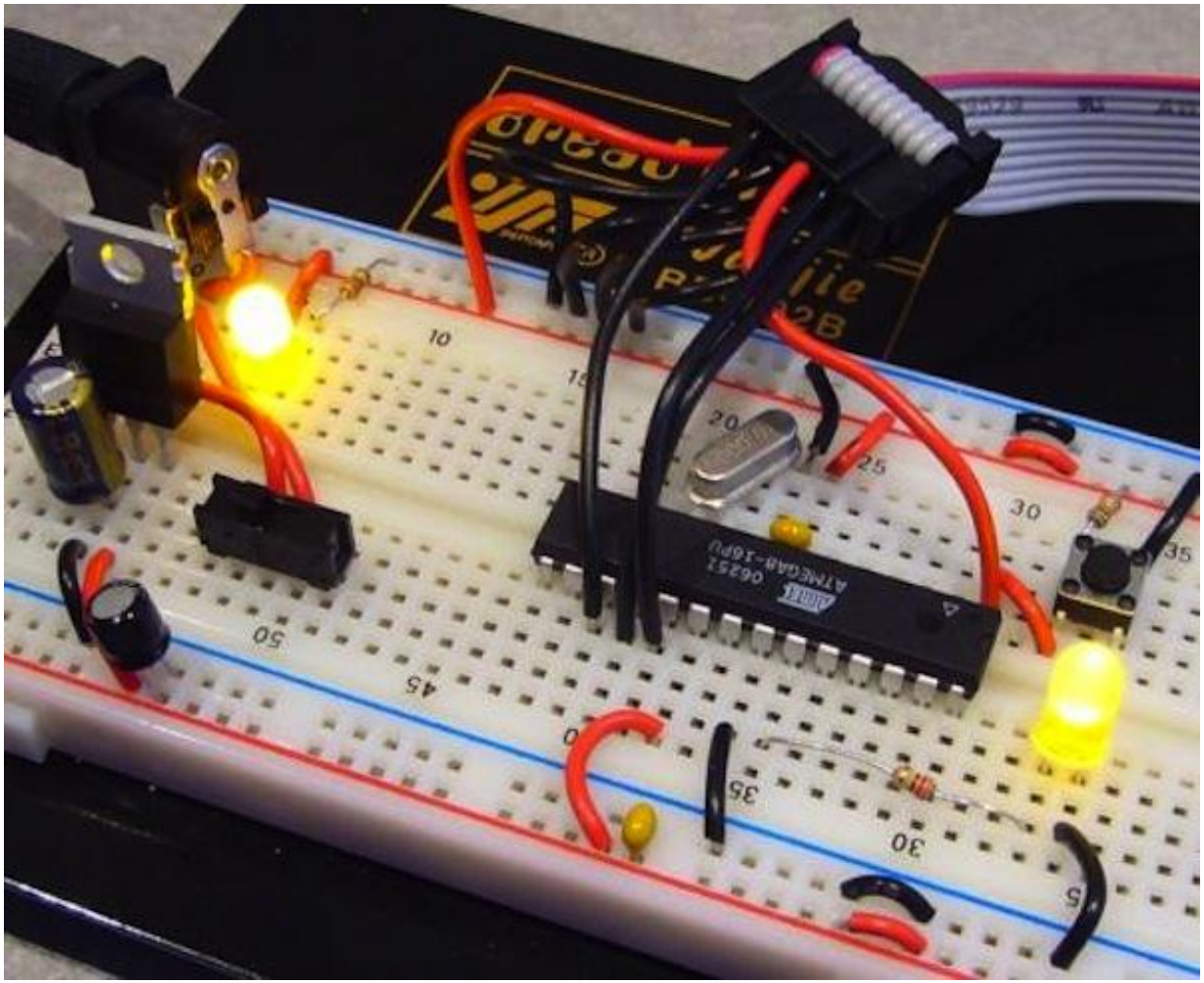
Breadboards 1960-tih



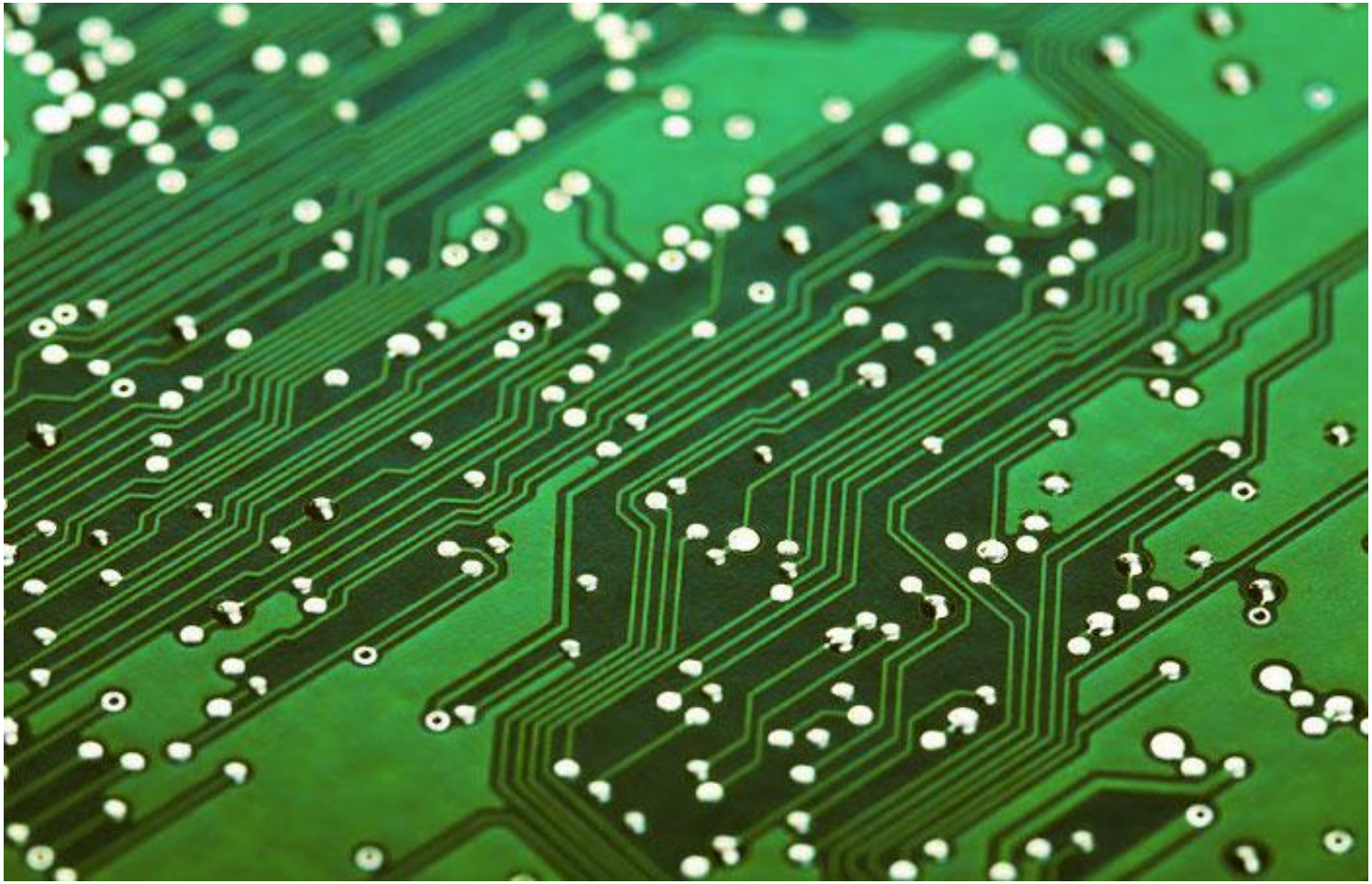
Improvizovani breadboard



Alat za pravljenje prototipova



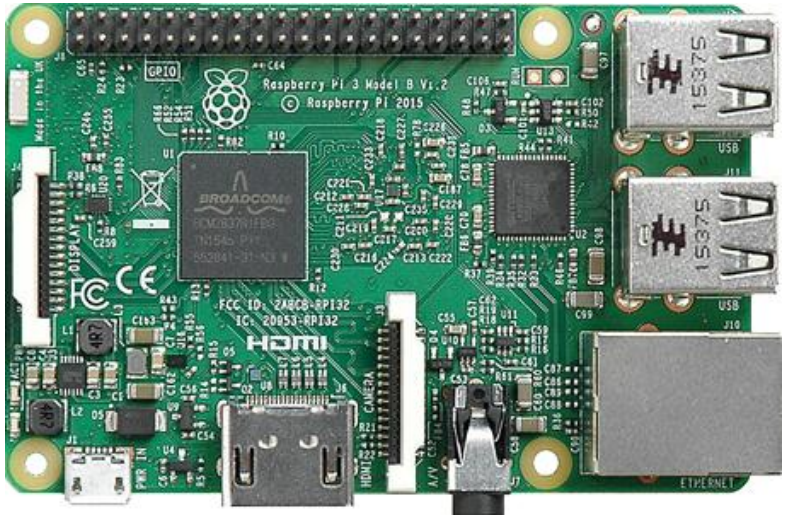
Integrisana Kola



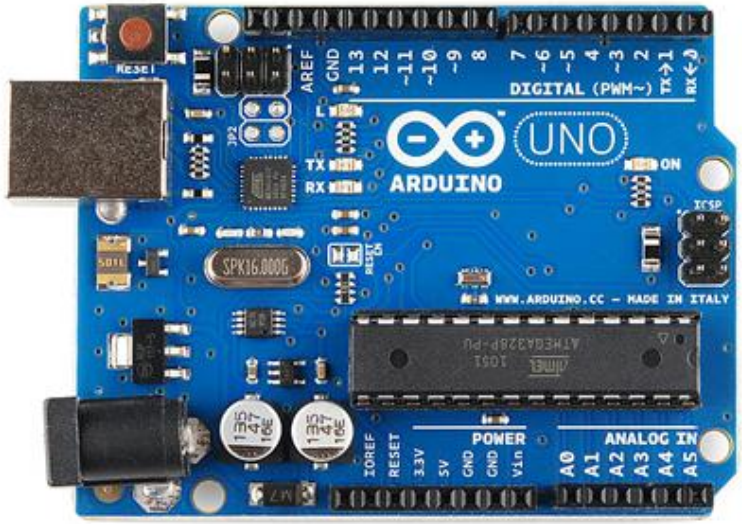
Šta znači akronim "DIY"?

- Do It Yourself - umjesto da angažujete profesionalca da uradi određeni zadatak ili da kupite gotov proizvod –uređaj odaberite da sami riješite problem ili da kreirate željeni proizvod bez direktne pomoći eksperta.
- Najpopularnije platforme za DIY su **Arduino** and **Raspberry Pi**.

Raspberry Pi



Arduino



Raspberry Pi

- Raspberri Pi je samostalni mikro-računar. Ima ugrađenu RAM, CPU, USB i Ethernet portove, standardne opcije izlaza, prikaza i još mnogo toga.
- Pokreće operativni sistem, najčešće LINUX i može se koristiti za kreiranje softverskih aplikacija sa terminalnog nivoa, do programskih jezika visokog nivoa kao što su Pithon i Scratch.
- Ogroman je broja aplikacija koje se mogu napraviti Raspberri Pi platformom bez povezivanja bilo čega osim napajanja, tastature / miša i ekrana.
- Savršen je za učenje programiranja na različitim jezicima, kao i za interakciju sa tradicionalnim kompjuterima.
- Ima mnogo više snage od bilo kog Arduina, ali sve što se na njemu radi je na softverskom nivou.
- Mnogo je komplikovaniji softver za povezivanje i upravljanje senzorima i spoljašnjim kolima.

ARDUINO

- Platforma dizajnirana za početnike!
- ARDUINO je najomiljenija platforma mikrokontrolera na svetu sa razlogom.
- Kombinuje mikrokontrolere i direktnu interakciju sa hardverom sa intuitivnim Open Source okruženjem uz pomoć kojeg se može dizajnirati gotovo sve.
- Kontrolisanjem mikrokontrolera se dobija željena funkcionalnost – ni manje ni više.
- Ne postoji softver koji treba da prođe i možete biti sigurni da će uraditi ono što ste mu rekli, bez brige o drugom sloju apstrakcije softvera.
- Platforma izbora za povezivanje senzore i drugog hardvera kako bi se stupilo u direktnu interakciju sa spoljnim svetom.

ARDUINO VS RASPBERRY PI

ARDUINO	RASPBERRY PI
Arduino je mikrokontoler – čip koji se može koristiti uz malo pomoćnih sklopova.	Ima više različitih čipova na ploči, uključujući mikroprocesor, kako bi se kreirao funkcionalni kompjuter.
Sensori i ostali hardver se mogu kontrolisati direktno kroz I/O pinove.	Da bi se kontrolisali I/O pinovi za senzore, motore i slično, mora se na njemu pisati kod koji bi kontrolisao softver nižeg nivoa.
Ne pokreće operativni sistem već prolazi kroz kod koji mu se prosleđuje.	Ponaša se kao samostalni računar i može se kodirati u njemu.
Dizajniran je za hardver nižeg nivoa i direktno programiranje	Dizajniran je za interakciju softvera i hardvera višeg nivoa.

Šta je mikrokontroler?

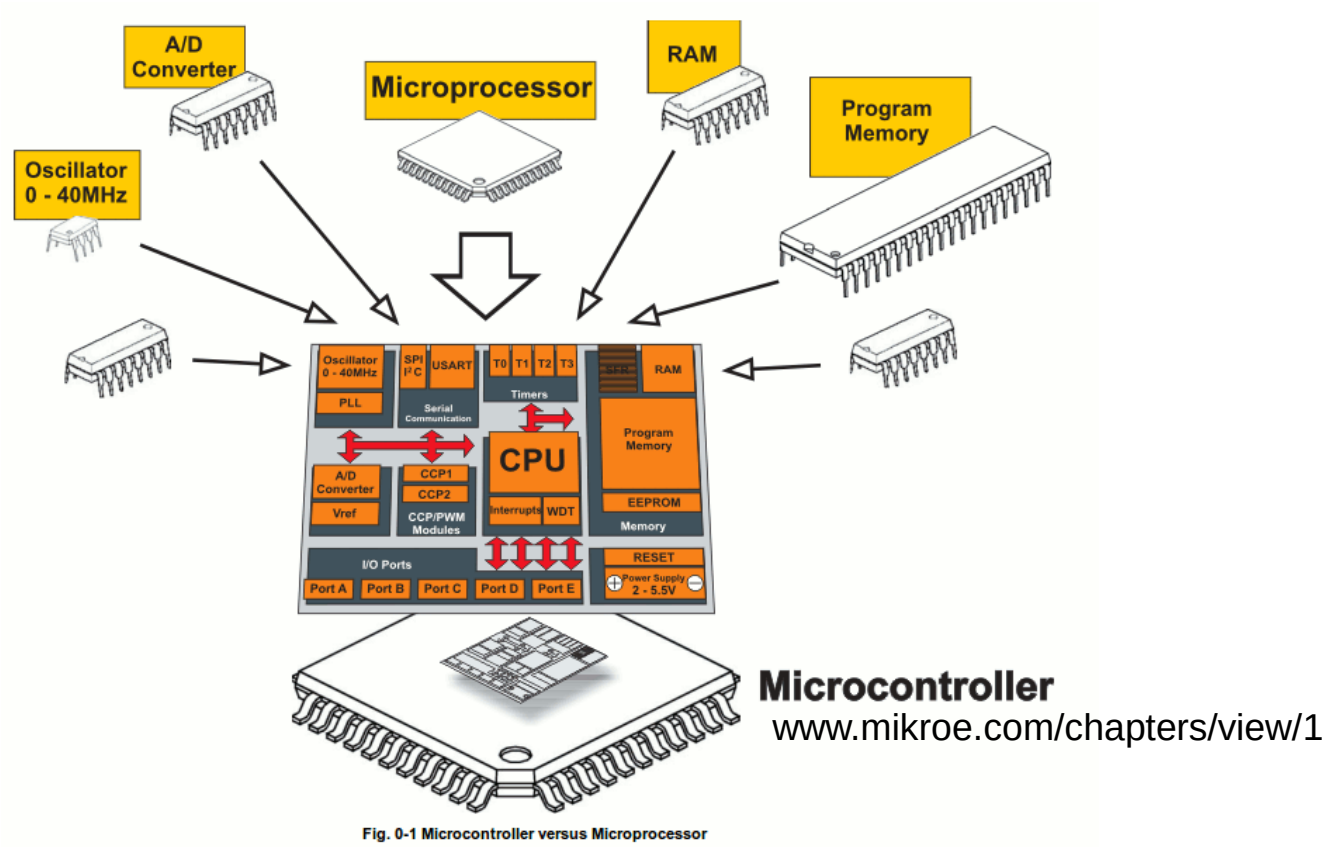
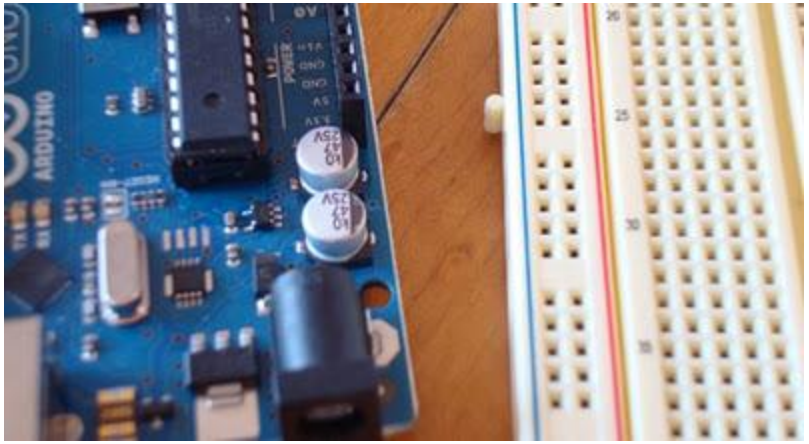


Fig. 0-1 Microcontroller versus Microprocessor

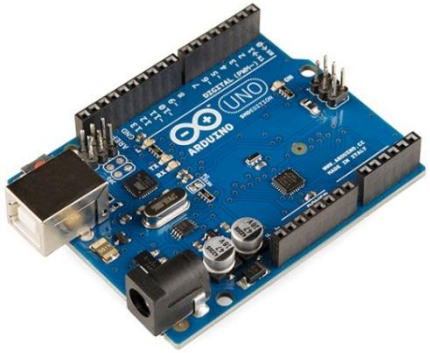
- Mali računar na jednom čipu
  - Sadrži procesor, memoriju i I/O
- Tipično je "ugrađen" unutar nekog uređaja koji kontroliše.
- Male je veličine i niske cijene.

Šta je breadboard

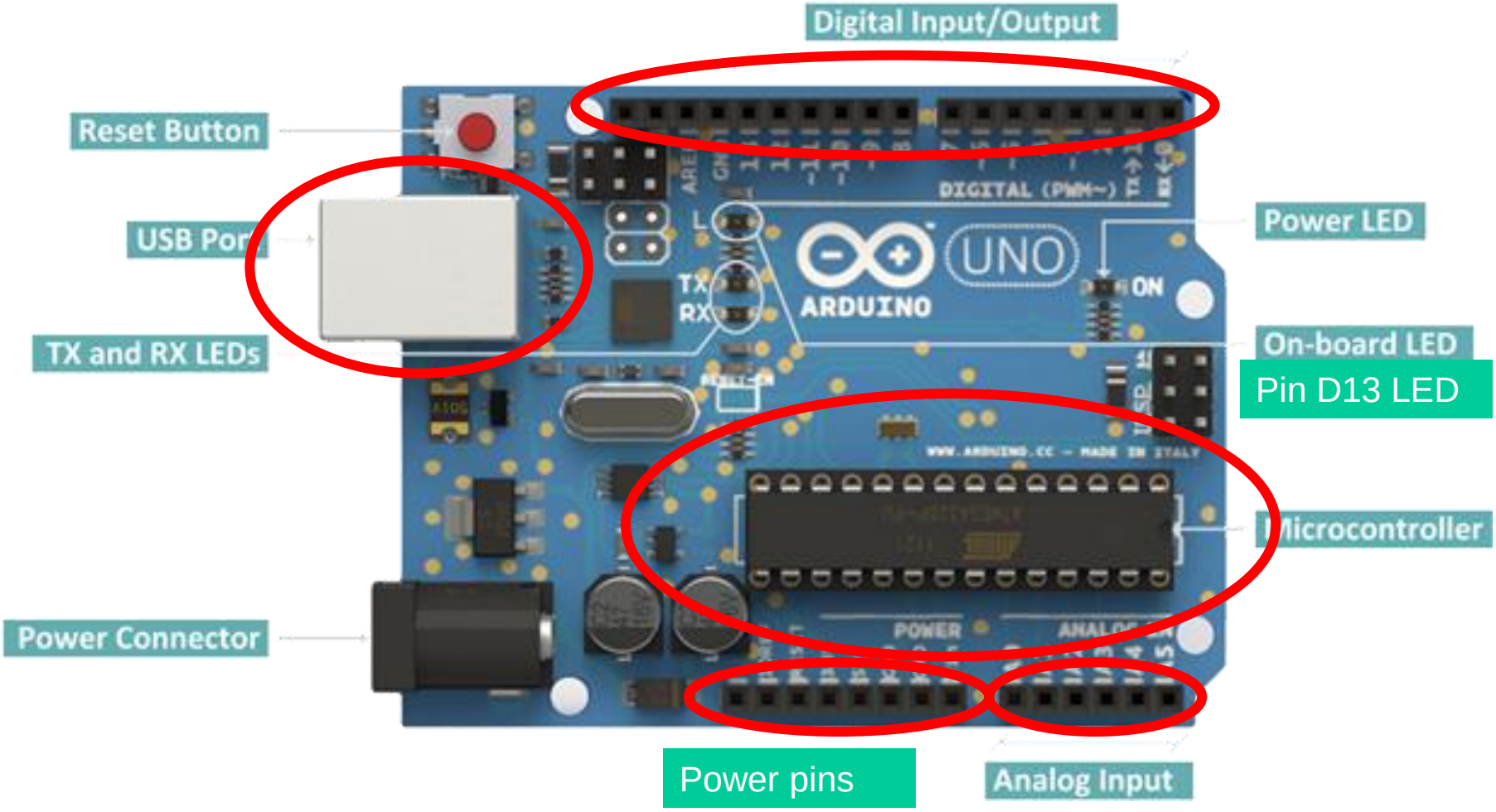


- Štampana pločica dizajnirana da olakša rad sa određenim mikrokontrolerom.

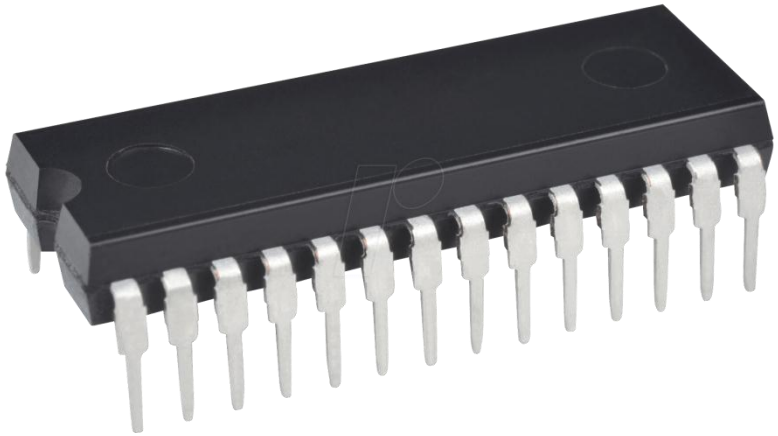
Arduino UNO

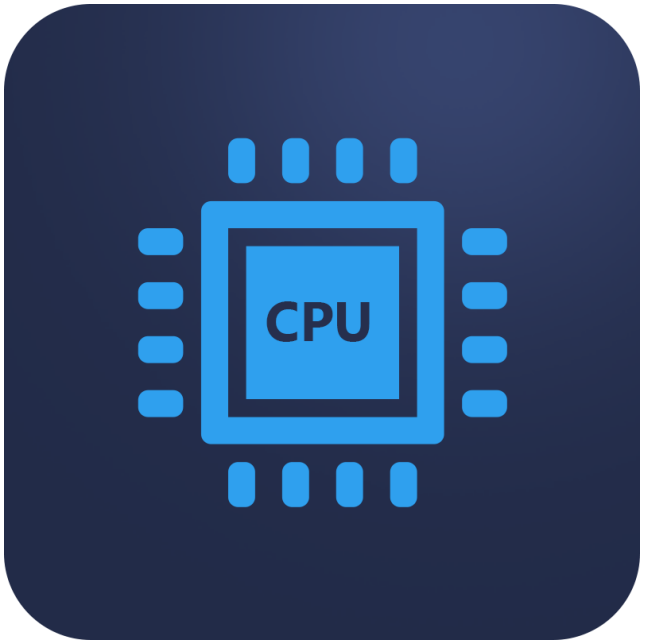


Arduino Uno


Digital Input/Output
Reset Button
USB Port
TX and RX LEDs
Power LED
On-board LED
Pin D13 LED
Microcontroller
Power Connector
Power pins
Analog Input
DIGITAL (PWM~)
UNO
ARDUINO
TX
RX
ON
POWER

Mikrokontroler
Mozak Arduina



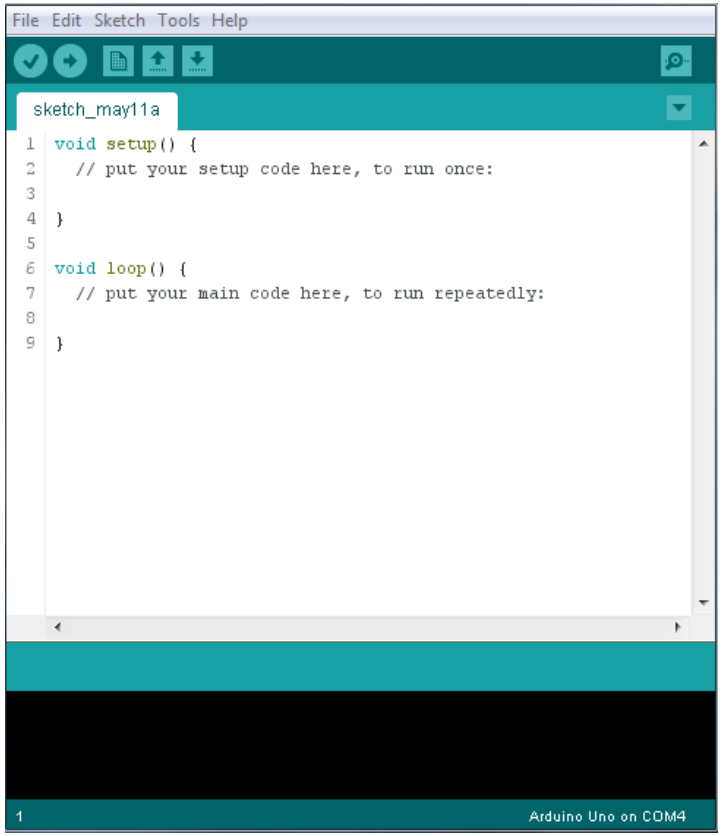


Prednosti

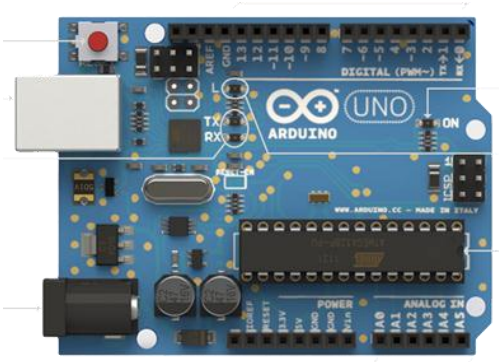
- Niska cijena
- Cross-platform - Arduino Softver (IDE) se može pokrenuti na Windows, Macintosh OSX i Linux operativnim sistemima.
- Jednostavno, jasno okruženje za programiranje - Arduino Softvare (IDE) je jednostavan za upotrebu za početnike, ali dovoljno fleksibilan za napredne korisnike.
- Open source i jednostavan za nadogradnju softvera i hardvera. Jezik se može proširiti preko C ++ biblioteka, a ljudi koji žele da razumiju tehničke detalje mogu napraviti skok od Arduina do AVR C programskog jezika na kojem se zasniva. Slično tome, može se po želji dodati AVR-C kod direktno u Arduino program.

Šta ćemo koristiti?

Arduino razvojno okruženje

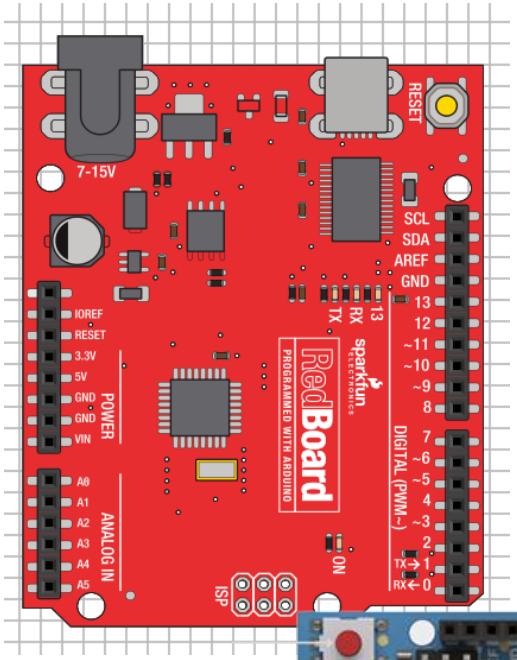


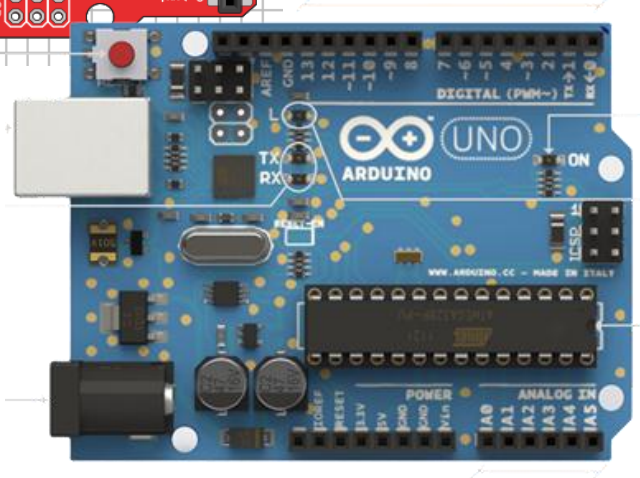
Arduino Uno



Senzori

Povezuje se sa računarom preko USB porta



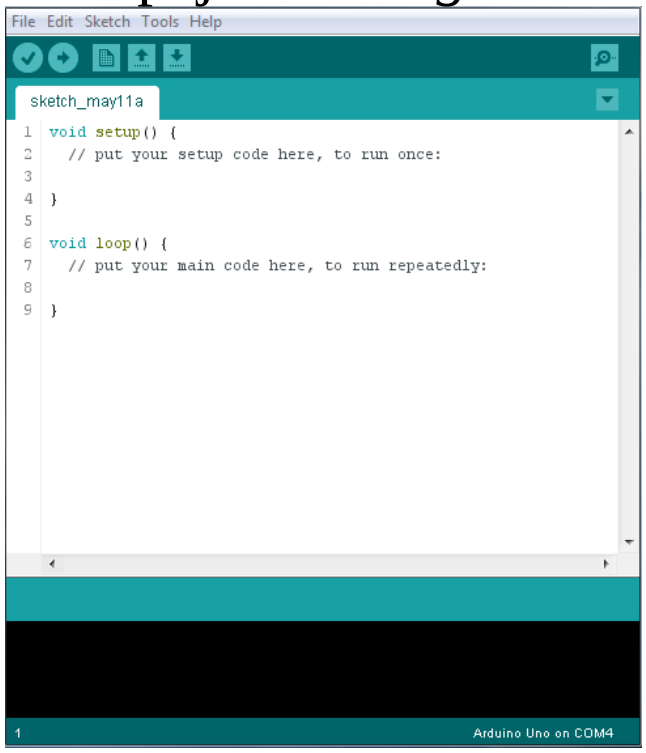


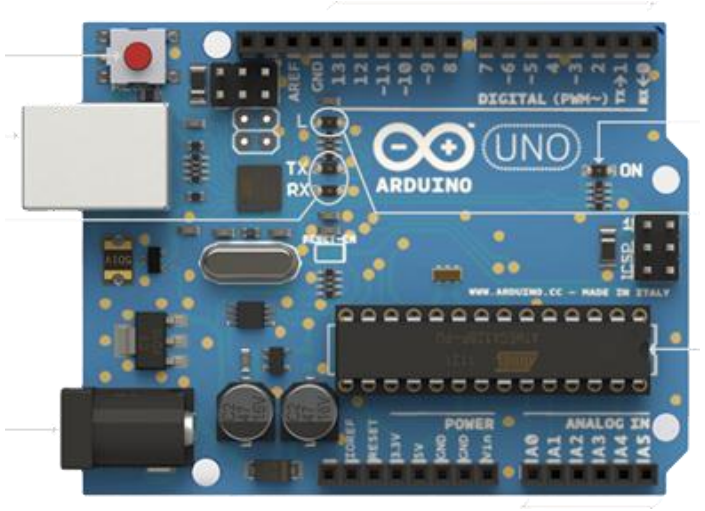
Dioda napajanja ostaje aktivna

Kontrolna dioda kratko zasvijetli

ARDUINO RAZVOJNO OKRUŽENJE

- Poseban program koji omogućava pisanje koda – skica (engl. sketches) za Arduino ploču na jednostavnom jeziku modeliranom po uzoru na Processing jezik (vvv.processing.org).
- Pritiskom na samo jedno dugme skica se šalje na ploču.
- Kod se prevodi na C programski jezik i prosleđuje open source kompajleru *avr-gcc*.





Počnimo sa radom

- http://arduino.cc/en/Guide/HomePage
- www.arduino.cc/en/Guide/Windows

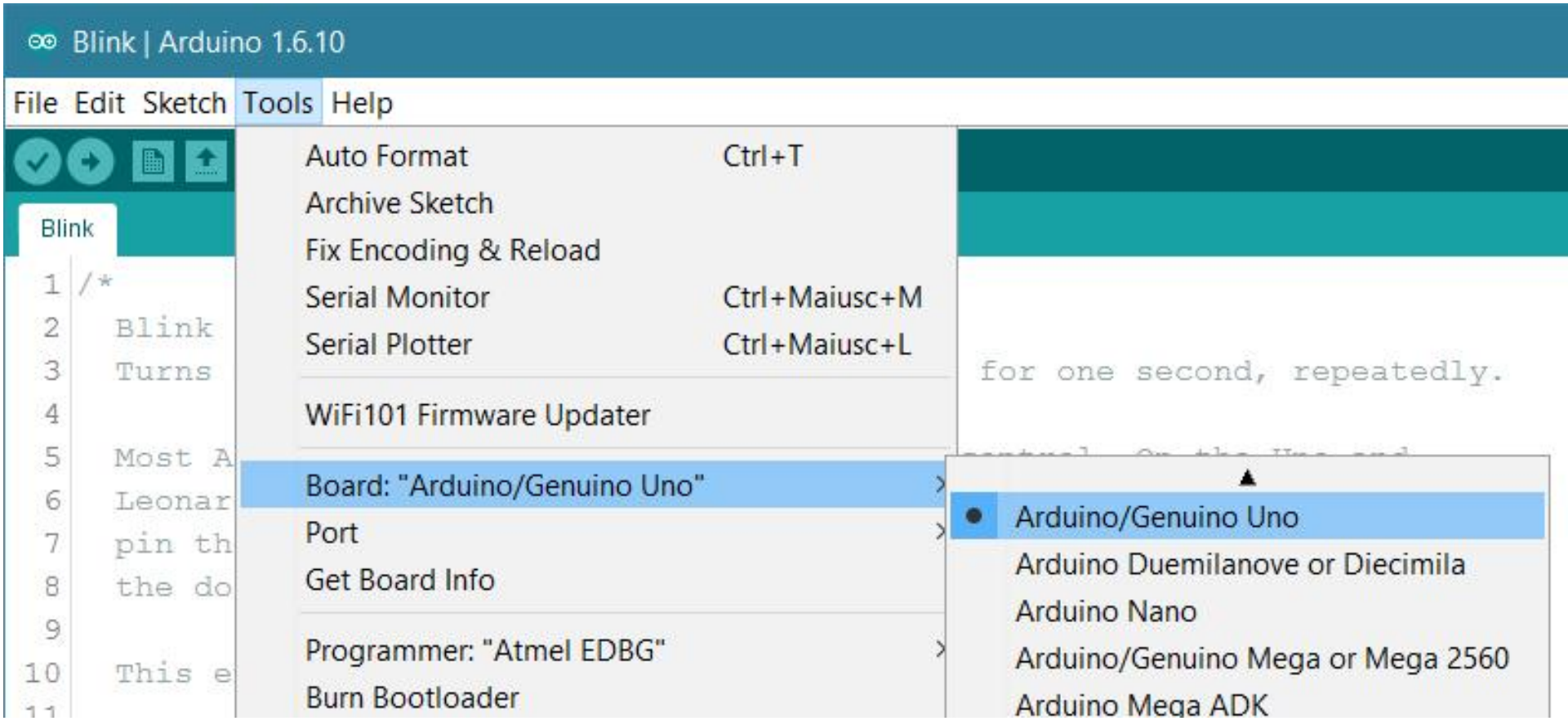
1. Preuzeti i instalirati Arduino razvojno okruženje (IDE);

https://www.arduino.cc/en/Main/Software

2. Povezati ploču sa računarom preko USB kabla;
3. Ukoliko je potrebno, instalirati drajvere; , ArduinoUNO.inf iz Drivers foldera Arduino Software download-a, za slučaj preuzimanja zip fajla;
4. Pokrenuti Arduino IDE;
5. Odabrati odgovarajuću ploču;
6. Odabrati odgovarajući serijski port;
7. Otvoriti primjer sa diodom koja blinka
   file/sketchbook/examples/digital/blink
8. Upload-ovati program

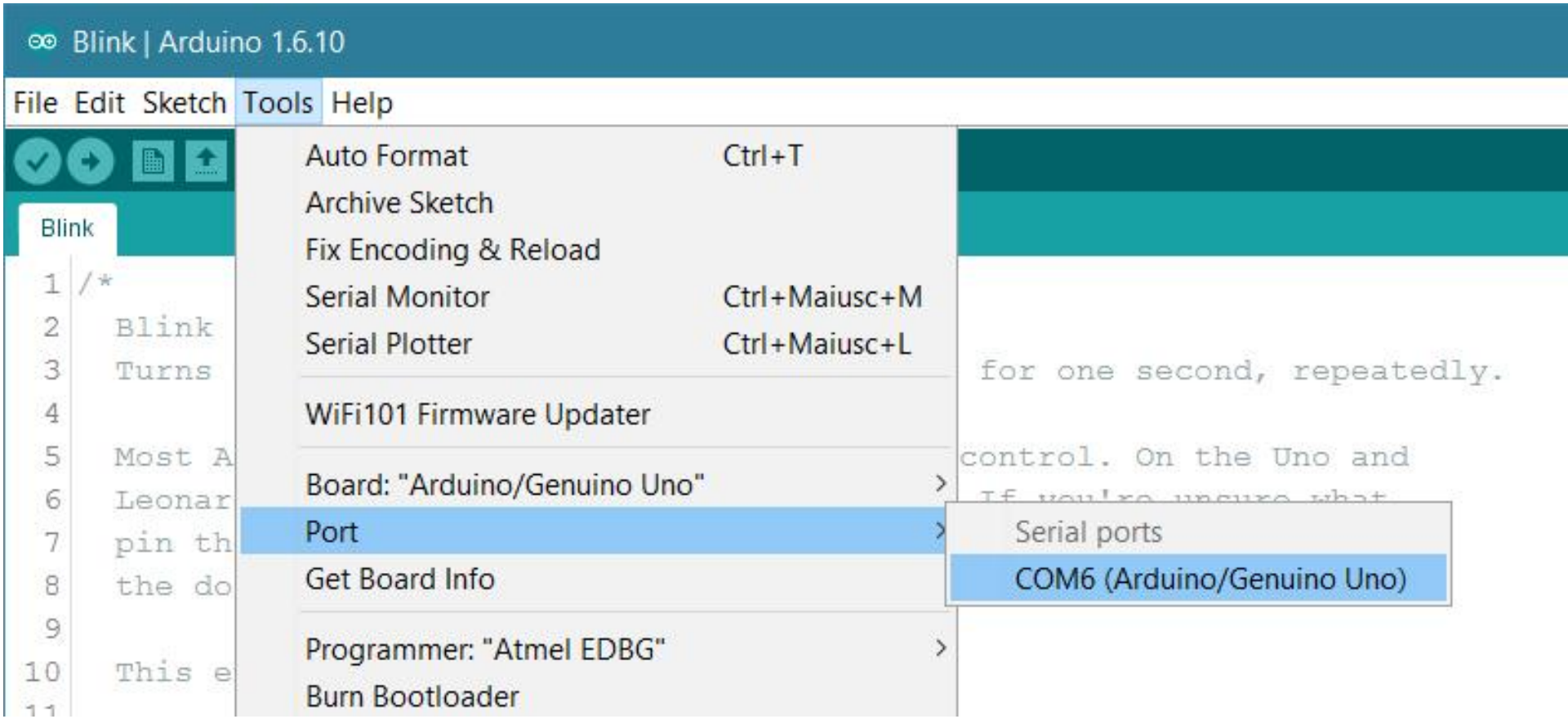
Odabrati tip ploče (board)

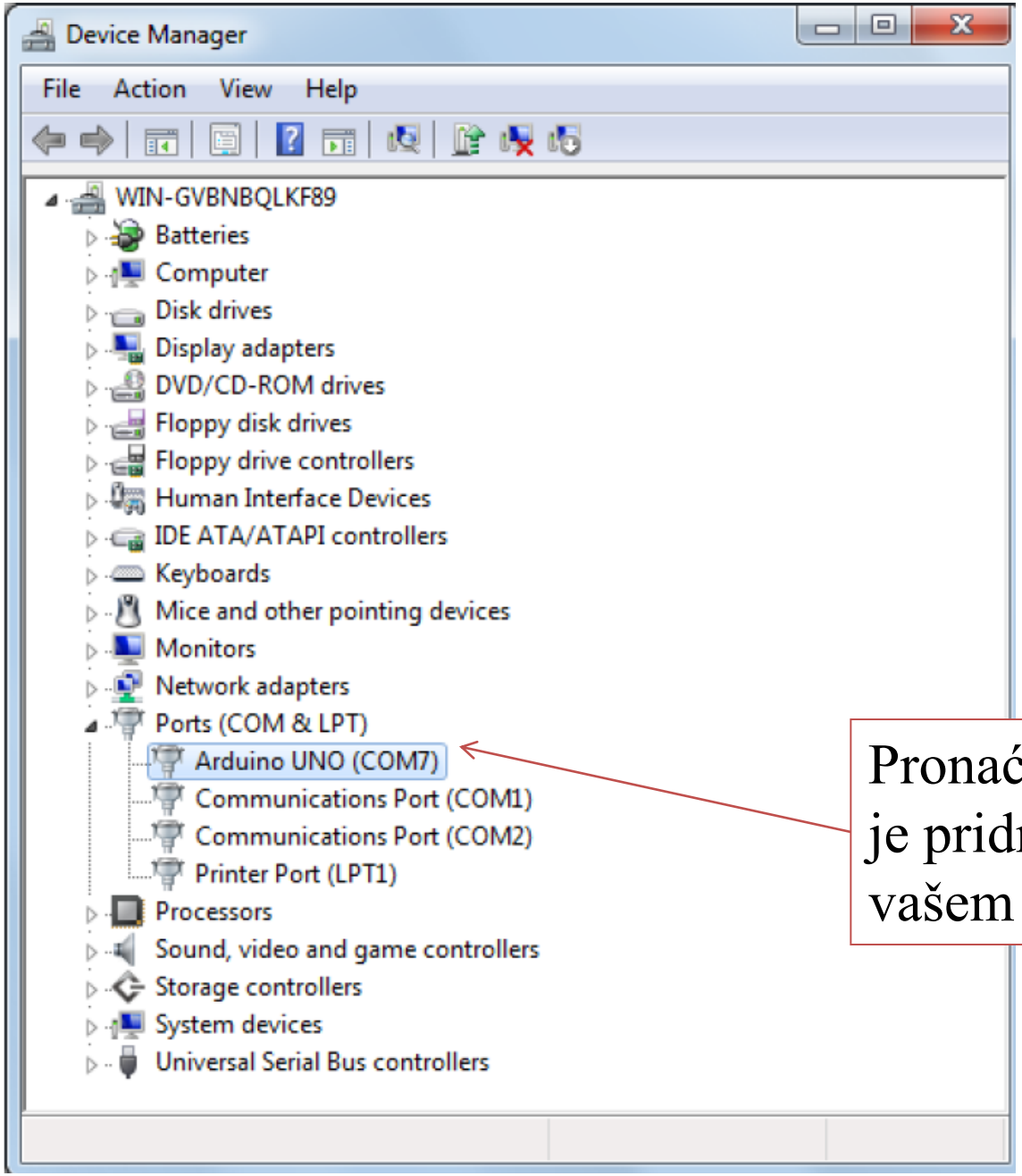
Tools - Board - Arduino/Genuino Uno



Odabrati komunikacioni port

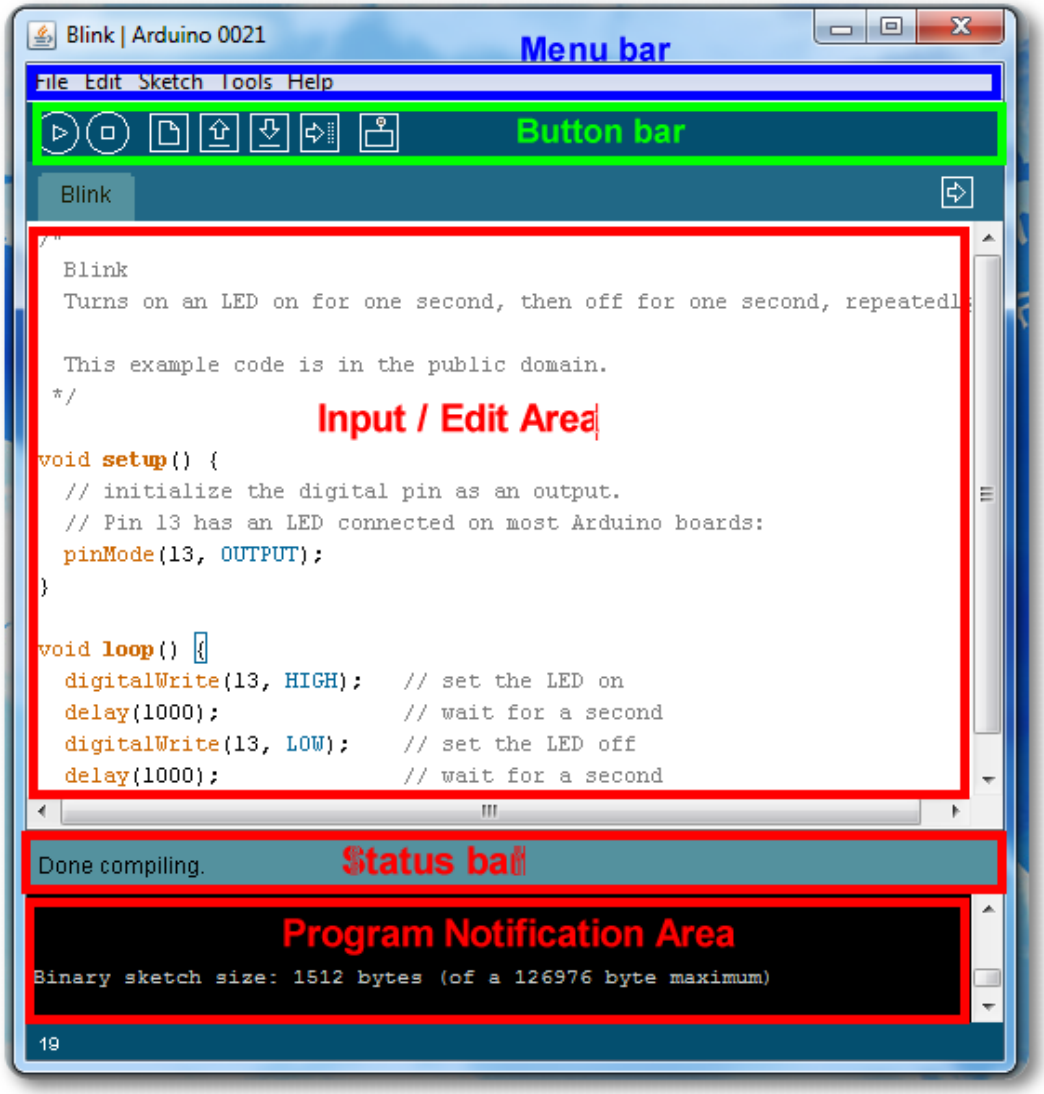
Tools - Port - COMx (x je broj porta)





Pronaći koji port je pridružen vašem Arduinu

Arduino razvojno okruženje

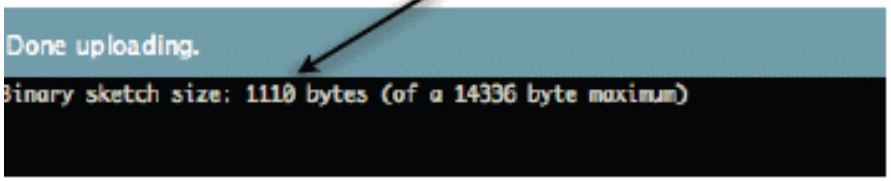


See: http://arduino.cc/en/Guide/Environment for more information

Statusne poruke

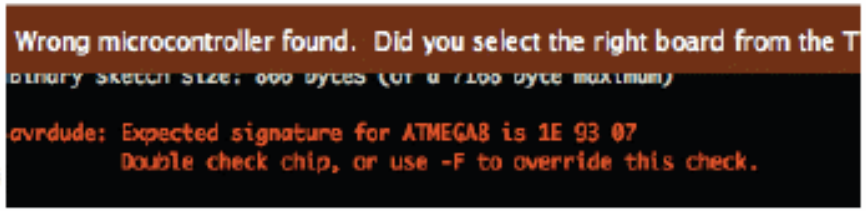
Veličina zavisi od kompleksnosti programa

Uspješan prenos podataka



Selektovan je pogrešan serijski port

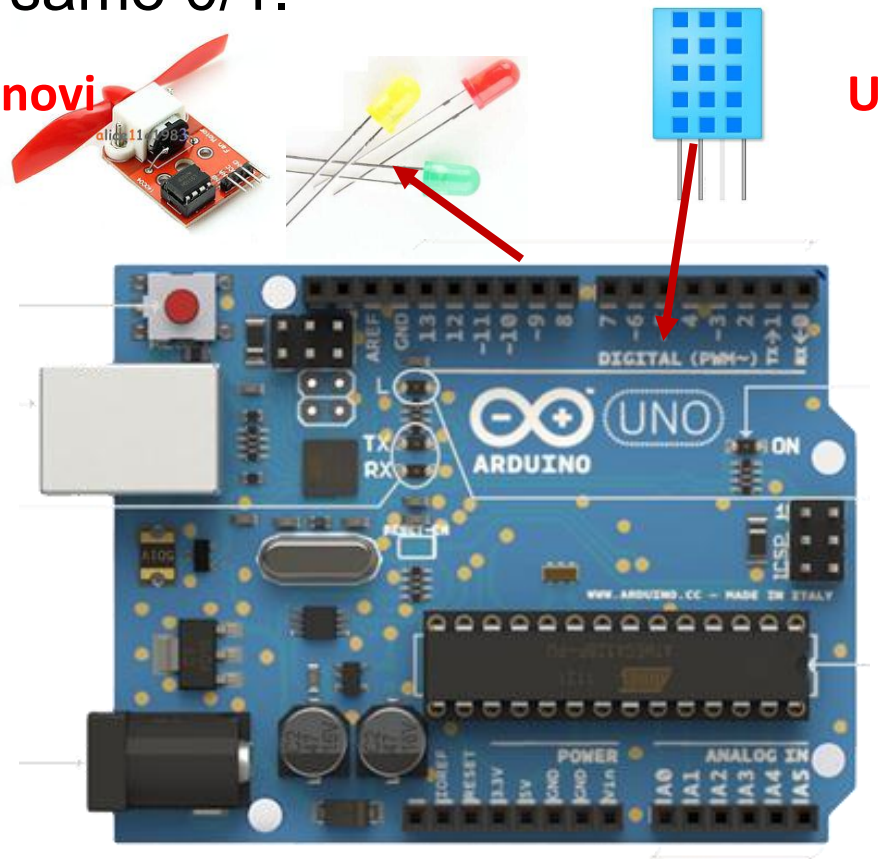
Selektovana je pogrešna ploča



Hardver – Digitalni U/I pinovi (D0-D13)

- Mogu biti ulazni ili izlazni pinovi.
- Mora se definisati softverski u IDE. U suprotnom dolazi do zabune.
- Očekuju samo 0/1.

Izlazni digitalni pinovi



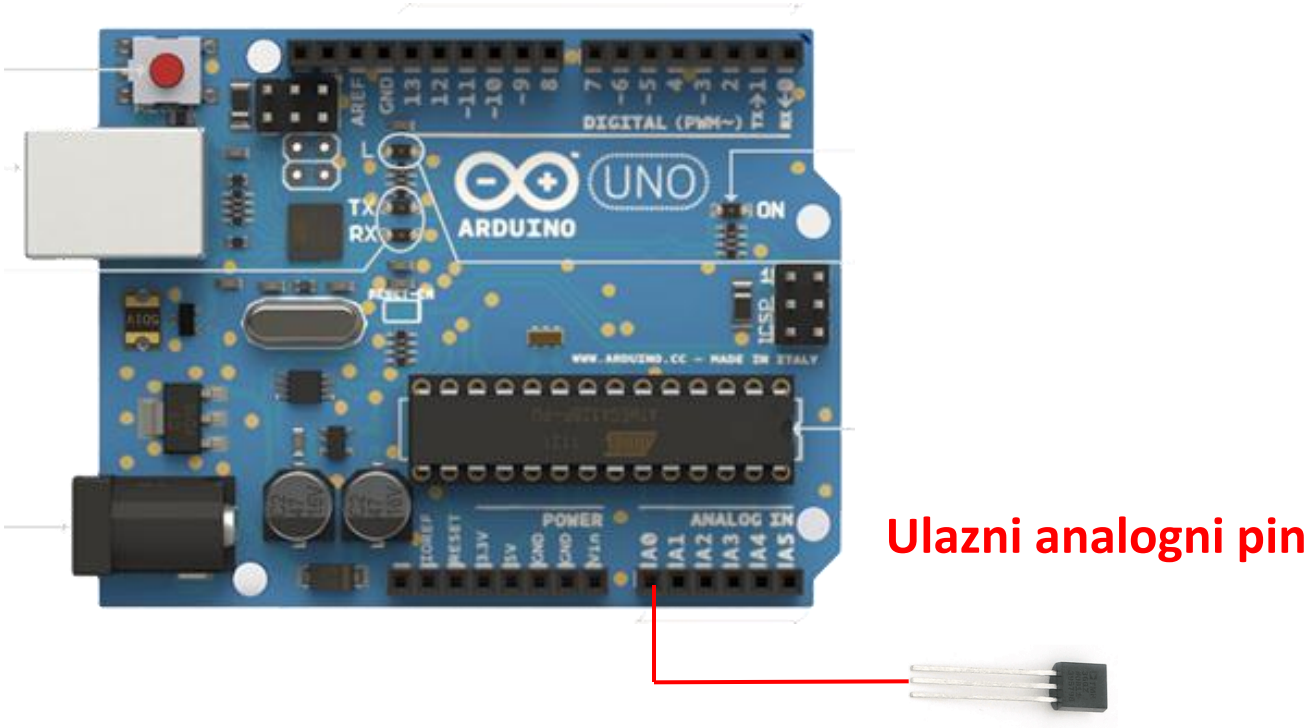
Ulazni digitalni pin

DIGITALNI PINOVI – DVA STANJA

DIGITALNA 1	DIGITALNA 0
HIGH LEVEL	LOW LEVEL
5 V	0 V

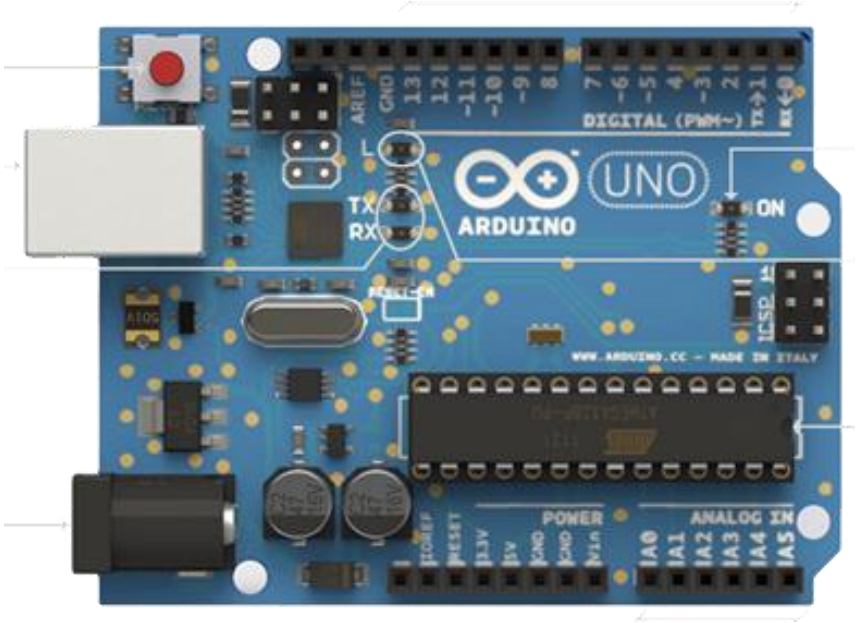
Hardver – Analogni U pinovi (A0-A5)

- Namjenski analogni ulazni pinovi.
- Uzimaju analogne vrijednosti (tj. napon očitavanja sa senzora) i pretvaraju ih u broj između 0 i 1023.



Programirani analogni I pinovi

- Pinovi D(3,5,6,9,10,11) – (obratiti pažnju na crticu uz njihov broj) se mogu reprogramirati u IDE tako da daju analogni izlaz.



Napajanje

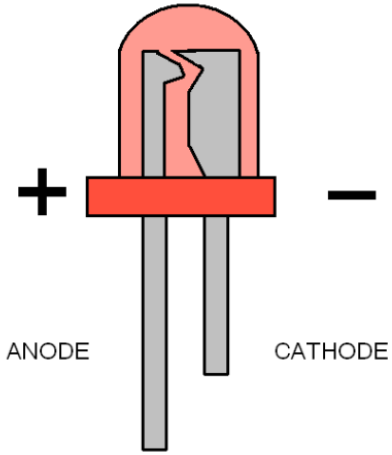
- Ploča se može napajati preko USB porta nakon povezivanja sa računarom ili AD adapterom od 9 V (2.1mm).
- Ako nema napajanja, napaja se preko USB kabla. Čim se priključi napajanje, automatski se napaja sa njega.

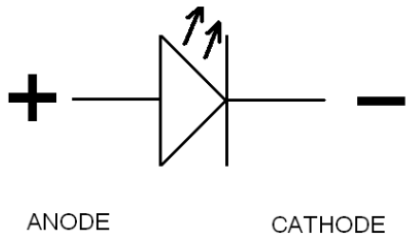
Interaktivni uređaji

- Sve aplikacije koje će se kreirati korišćenjem Arduino pločice će vršiti komunikaciju sa okruženjem po principu “interaktivnih uređaja”.
- Interaktivni uređaj je elektronski sklop koji je sposoban da:
  - Osjeti okruženje pomoću **senzora** (elektronskih komponenti koje konvertuju mjerenja iz okruženja u električne signale);
  - Obrađuje informacije koje dobija od senzora, a koje su implementirane kao softver i donosi odluku o akciji koju treba sprovesti, koju takođe definiše kroz softver.
  - Utiče na okruženje pomoću **aktuatora** (elektronskih komponenti koje pretvaraju električni signal u fizičko djelovanje).

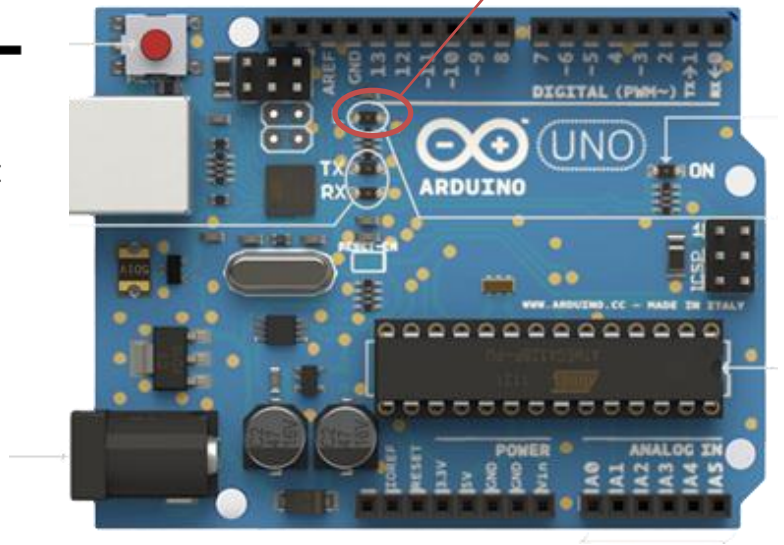
Light-Emitting Eiode (LED)

- Elektronska komponenta koja emituje svjetlost.
- Jedna LED dioda se nalazi na samoj Arduino ploču. Obilježena je velikim slovom „L“.
- Može se povezati i proizvoljna LED dioda. Kraća nožica je katoda, duža anoda.





Ugrađena LED dioda



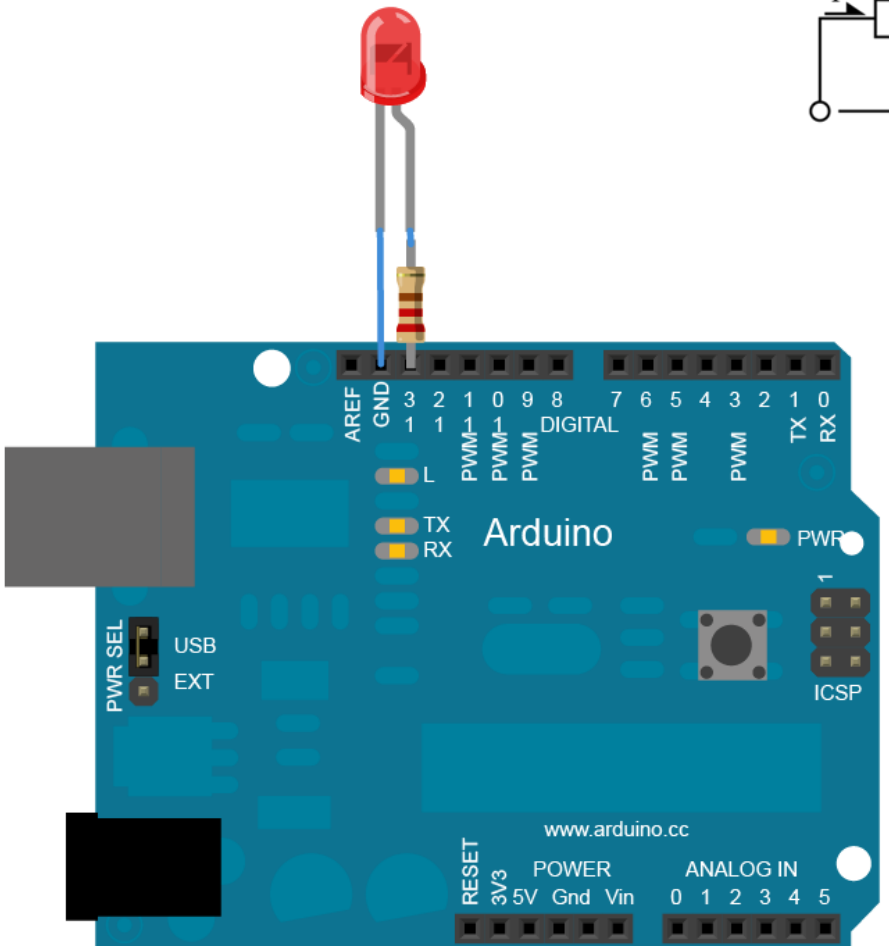
Glavne Funkcije

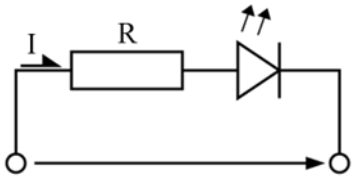
```
void setup()
{
  // pokreće se samo jedan put na početku programa
}

void loop()
{
  // kod koji se neprekidno izvršava
}
```

Arduino ne može da pokrene više programa istovremeno i programi se ne mogu sami prekinuti. Kada se uključi pločica i prenese program na nju, program se pokreće, kada se želi prekinuti izvršavanje programa, isključi se pločica.
Kada se kod jednom prenese na Arduino pločicu, ostaje na njoj i nakon njenog isključivanja ili resetovanja (kao na hard disku).

Primjer 1 – LED dioda koja blinka





Primjer 1 – LED dioda koja blinka

```
//Komentar – skica je zapravo niz instrukcija mikrokontroleru
//const int LED_PIN 13
#define LED_PIN 13
void setup()
{
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN, HIGH);
  delay(1000);
  digitalWrite(LED_PIN, LOW);
  delay(1000);
}
```

Definisanje konstanti

Definisanje makroa

Konfigurisanje digitalnog pina kao izlaznog (output)

5V na LED pin

Kašnjenje od 1000ms

0V na LED pin

Kašnjenje od 1000ms

Predefinisane konstante – OUTPUT/INPUT, HIGH/LOW.

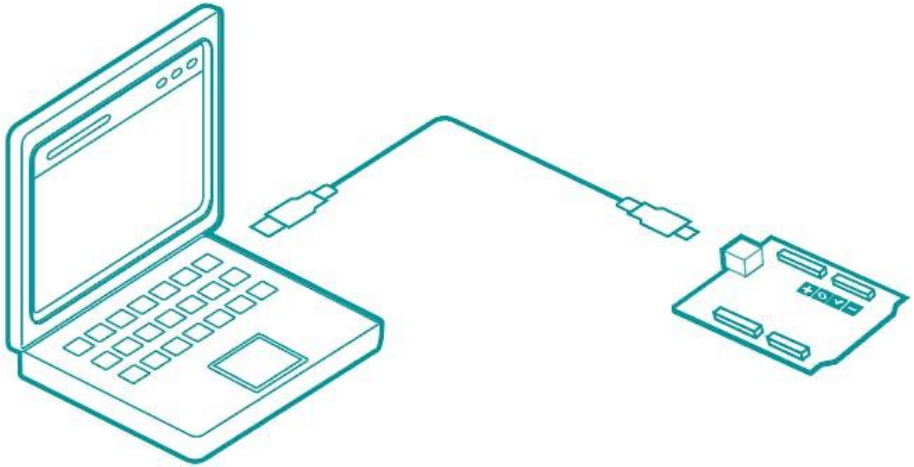
Kod - Definisanje digitalnih pinova makroima

```
#define NAME NUMBER
#define LED_PIN 7
#define TEMP_PIN 8
#define MOTOR_PIN 9
```

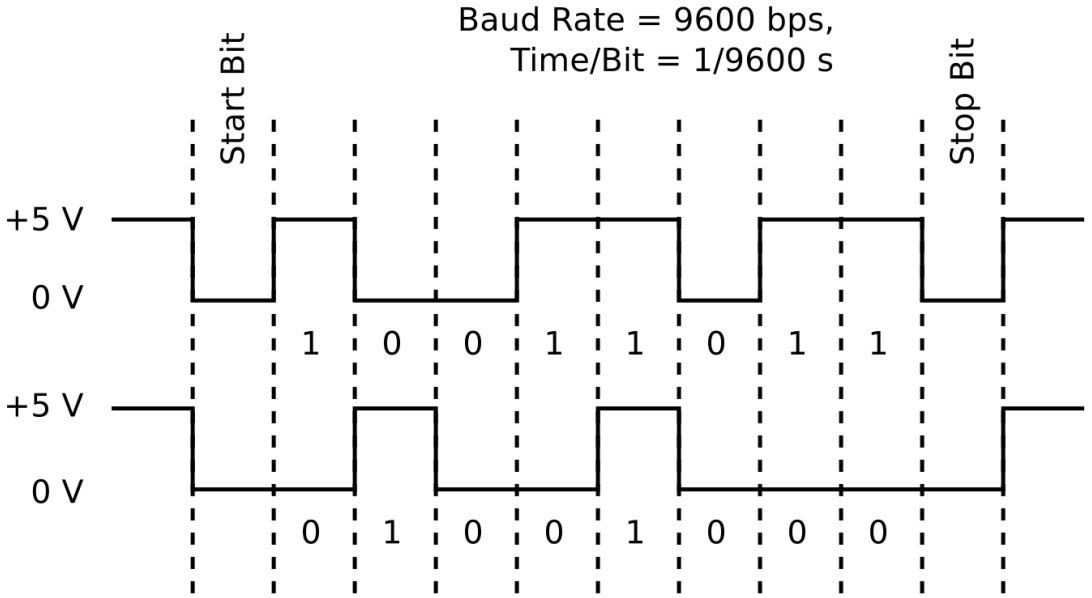
Kod – Konfigurisanje digitalnih pinova

- `pinMode(NAME,MODE)`
- Funkcija kojom se vrši konfiguracija pina NAME kao izlaznog pina (`pinMode(NAME,OUTPUT)`) ili kao ulaznog pina, (`pinMode(NAME,INPUT)`)

Serijska Komunikacija



Serijska komunikacija uspostavljena preko USB porta se može koristiti u programima za slanje podataka računaru i za preuzimanje komandi sa računara.

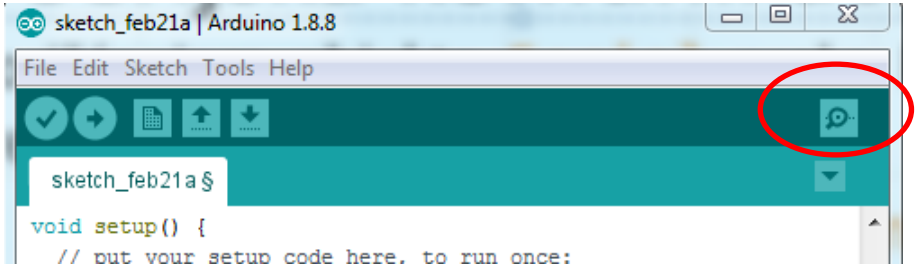


Serijska komunikacija

- Vrši se korišćenjem objekta `Serial` preko kojeg pristupamo svim potrebnim funkcijama

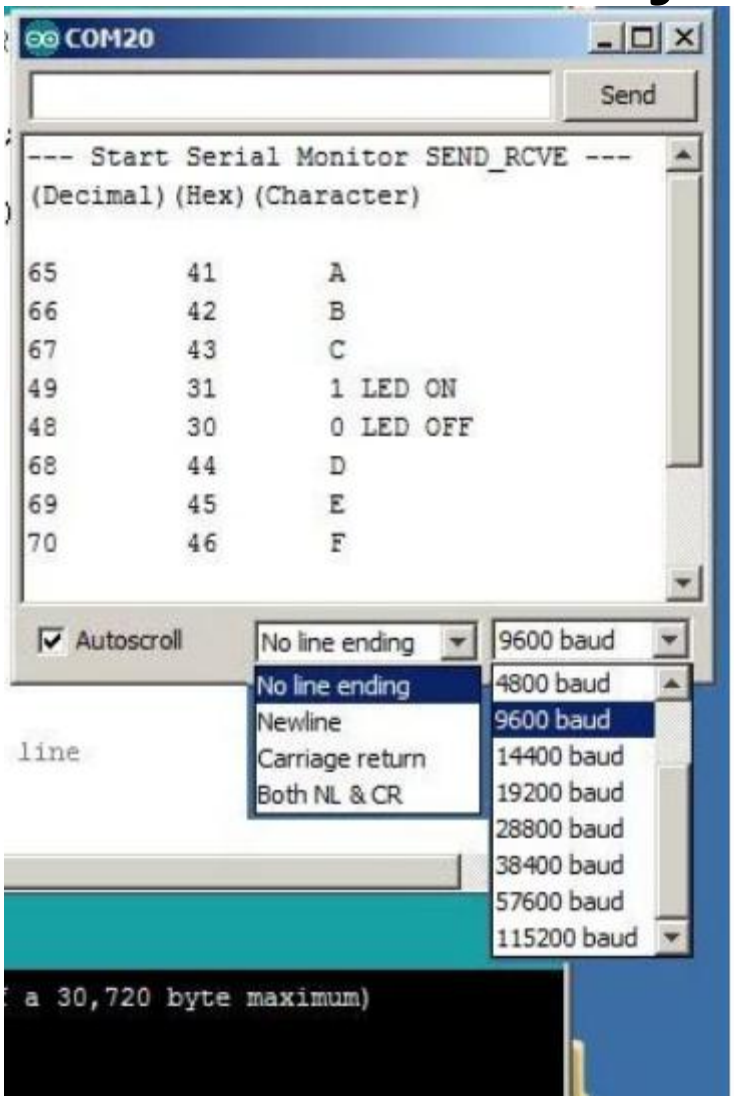
```
Serial.ImeFunkcije(argumenti)
```

- `Serial.begin(brzina)` – priprema Arduino za početak primanja i slanja serijskih podataka definišući brzinu njihovog prenosa. Najčešće se koristi brzina komunikacije od 9600 bps sa Arduino IDE serijskim monitorom.
- Serijski monitor je poseban prozor koji se aktivira klikom na krajnju desnu ikonicu Arduino IDE okruženja.



- Arduino pločica mora biti konektovana sa računarom preko USB kabla da bi se mogao aktivirati Serijski monitor.

Serijski monitor



https://www.instructables.com/id/HOW-TO-use-the-ARDUINO-SERIAL-MONITOR/

Gornje polje za editovanje služi za unošenje podataka. Unos se prihvata pritiskom na dugme *Send* ili *Enter* sa tastature.

Veći kvadrat je labela u kojoj se ispisuju podaci koje Arduino šalje.

Ne dnu su dvije padajuće liste:

- Lijevom se definiše „kraj linije“ koji će biti poslat na Arduino nakon što se klikne na dugme *Send*;
- Desnom se postavlja brzina prenosa podataka pri komunikaciji sa Arduino pločom. Mora odgovarati brzini podešenoj u programu za podešavanje parametara sa `Serial.begin(brzina)`.

Serijska komunikacija

- `Serial.print(data,Format)` – Šalje podatke serijskom portu. Format je opcion. Ako se ne specificira, podaci se štampaju kao obični tekst.
- Cijeli brojevi se štampaju koristeći ASCII karakter za svaku cifru.
- Realni brojevi se štampaju na sličan način, kao ASCII cifre, podrazumijevajući dvije decimale. Ukoliko se broj decimala želi promijeniti, isti se specificira *Format* argumentom.

  `Serial.print(2.3785,3)` štampa `"2.378"`
- Format može uzeti i vrijednosti (BIN,DEC,OCT,HEX) čime se zadati broj štampa uz pretvaranje u odgovarajući brojni sistem.
- `Serial.println(data,Format)` – funkcioniše na isti način uz prelazak u novi red nakon štampanja data podataka.

Serijski Monitor

```
void setup()
{
      Serial.begin(9600);
}

void loop()
{
      Serial.println(“Hello, World!”);
      delay(1000);
}
```

Serijska komunikacija

- `Serial.read()` – Preuzima jedan bajt serijskih podataka.
- `Serial.available()` – Vraća broj nepročitanih bajtova (karaktera) dostupnih u serijskom portu za čitanje preko `read()` funkcije. Ovo su podaci koji su već stigli i uskladišteni su u baferu serijskog prijema (koji sadrži 64 bajta).
- Ukoliko je pročitano sve što je dostupno, `Serial.available()` vraća 0 dok novi podaci ne stignu na serijski port.
- Podaci mogu pristizati preko serijskog porta brže nego što ih program može obraditi. Arduino čuva sve ulazne podatke u baferu serijskog prijema.
- `Serial.flush()` – Čeka da se završi prenos odlaznih serijskih podataka. (Prije Arduino 1.0, `Serial.flush()` je uklanjala sve buferovane dolazne serijske podatke).

Povežimo se sa monitorom

```
void loop(){
        if (Serial.available() > 0){
                int command = Serial.read();
                if (command == '1'){
                        digitalWrite(LED_PIN, HIGH);
                        Serial.println("LED on");
                }
                else if (command == '2'){
                        digitalWrite(LED_PIN, LOW);
                        Serial.println("LED off");
                }
                else{
                        Serial.print("Nepoznata komanda:");
                        Serial.println(command);
                }
        }
}
```

Čitanje komandi