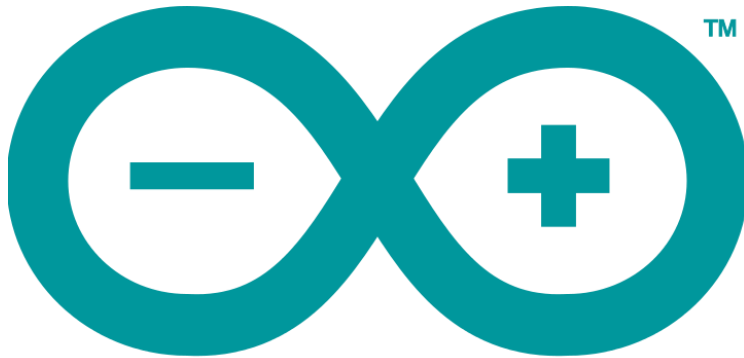


Merni sistemi u računarstvu, <http://automatika.etf.bg.ac.rs/sr/13e053msr>



# Arduino programiranje

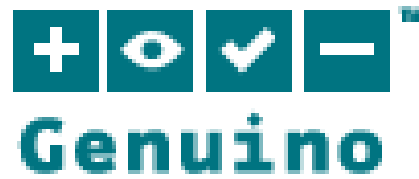
Dr Nadica Miljković, vanredni profesor kabinet 68, [nadica.miljkovic@etf.bg.ac.rs](mailto:nadica.miljkovic@etf.bg.ac.rs)

Prezentacija za ovu vežbu je delimično pokrivena knjigom Alan G. Smith, Introduction to Arduino: A piece of cake!, [Online], <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.



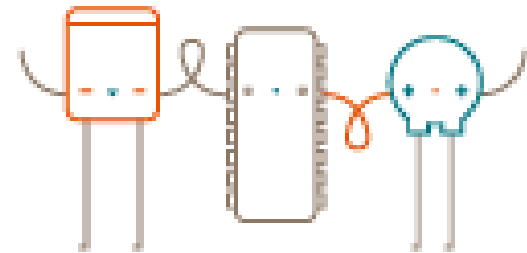
# Arduino softver

- Preporučena literatura za osnove programiranja u Arduino programskom okruženju je knjiga u kojoj se koristi hardver UNO R3:
  - Alan G. Smith, Introduction to Arduino: A piece of cake!, [Online], <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.
  - I drugi materijali na internetu i na sajtu predmeta
- Arduino softver je napisan u Java, C i C++. Operativni sistemi na kojima je moguće instalirati Arduino softver i programirati u njemu su: Windows, macOS i Linux, <https://en.wikipedia.org/wiki/Arduino>.
  - Arduino je veoma sličan C++ programskom jeziku.
- NAPOMENA: Nisu sve funkcije dostupne za UNO R3 hardver. Neke funkcije iako postoje u softverskom okruženju su specifično nemenjene za hardver koji se koristi. Potrebno je pogledati uputstvo za odgovarajuću funkciju.

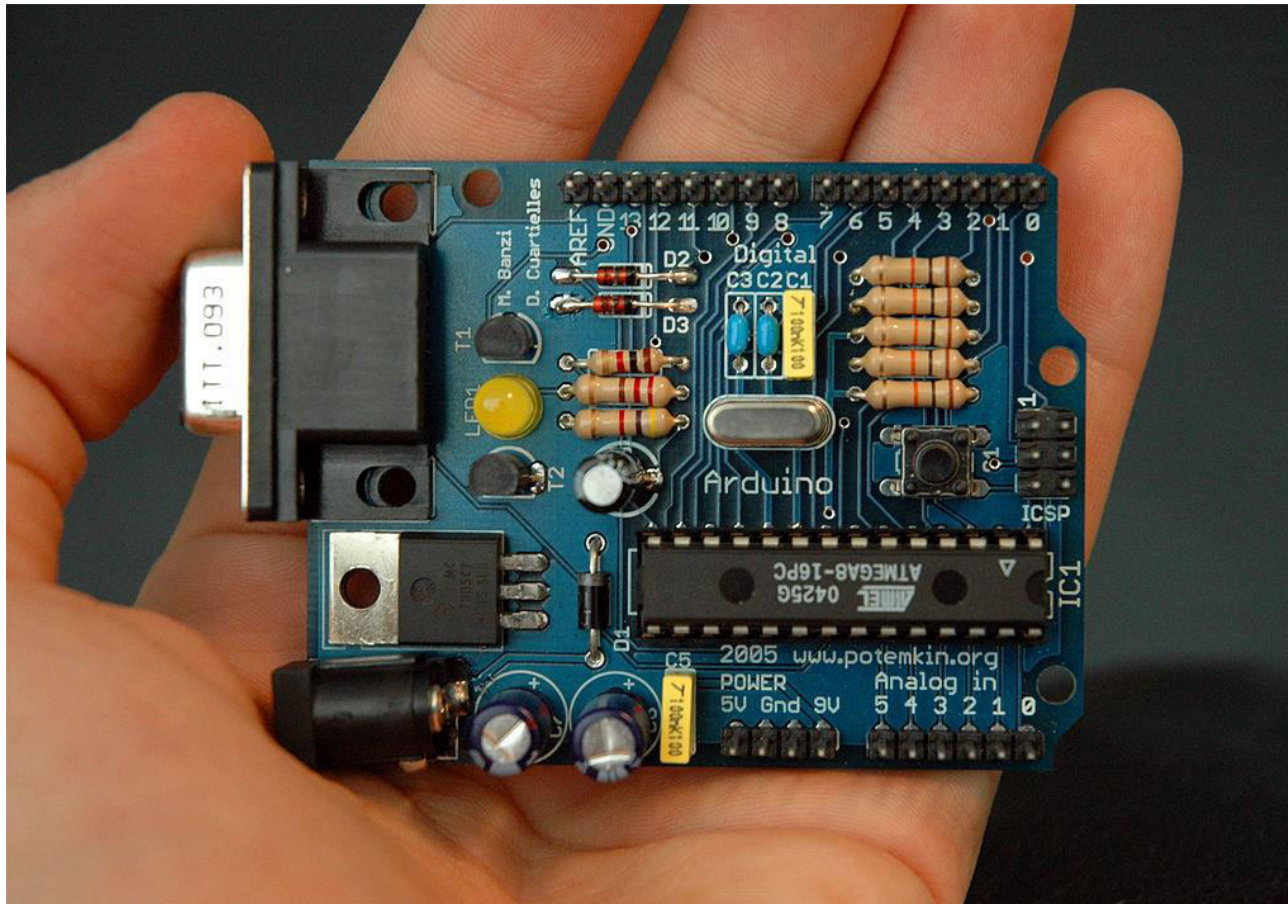


AN OPEN PROJECT WRITTEN, DEBUGGED,  
AND SUPPORTED BY ARDUINO.CC AND  
THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS  
OF [ARDUINO.CC](https://arduino.cc) on [arduino.cc/credits](https://arduino.cc/credits)

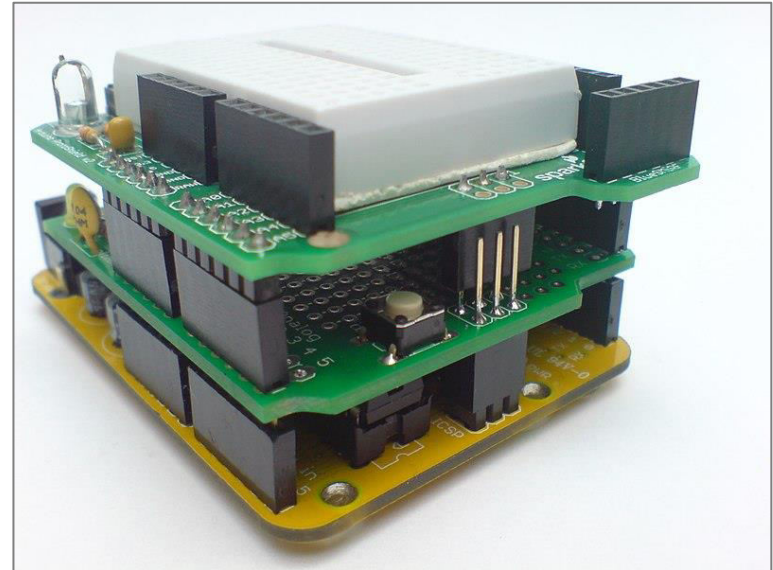
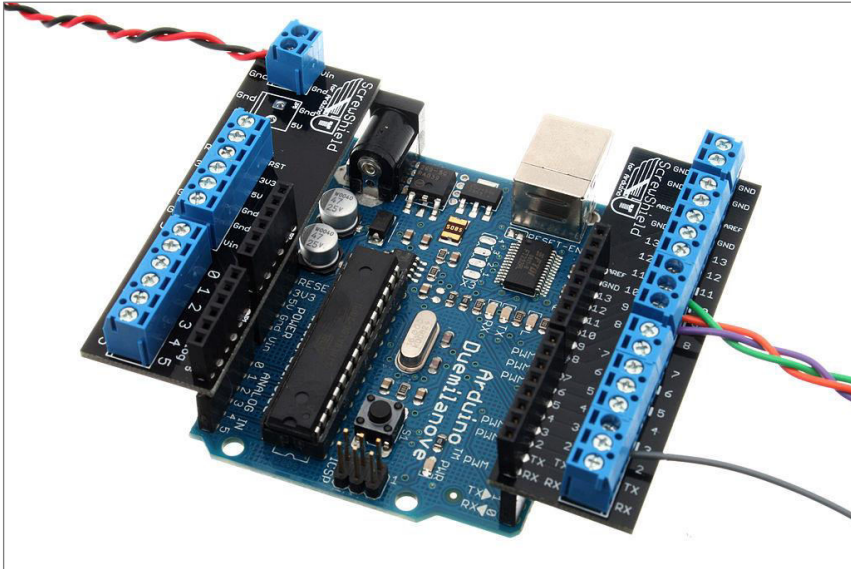


# Par zanimljivosti ...



By Nicholas Zambetti - <http://www.arduino.cc/>, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=9182627>

• • •

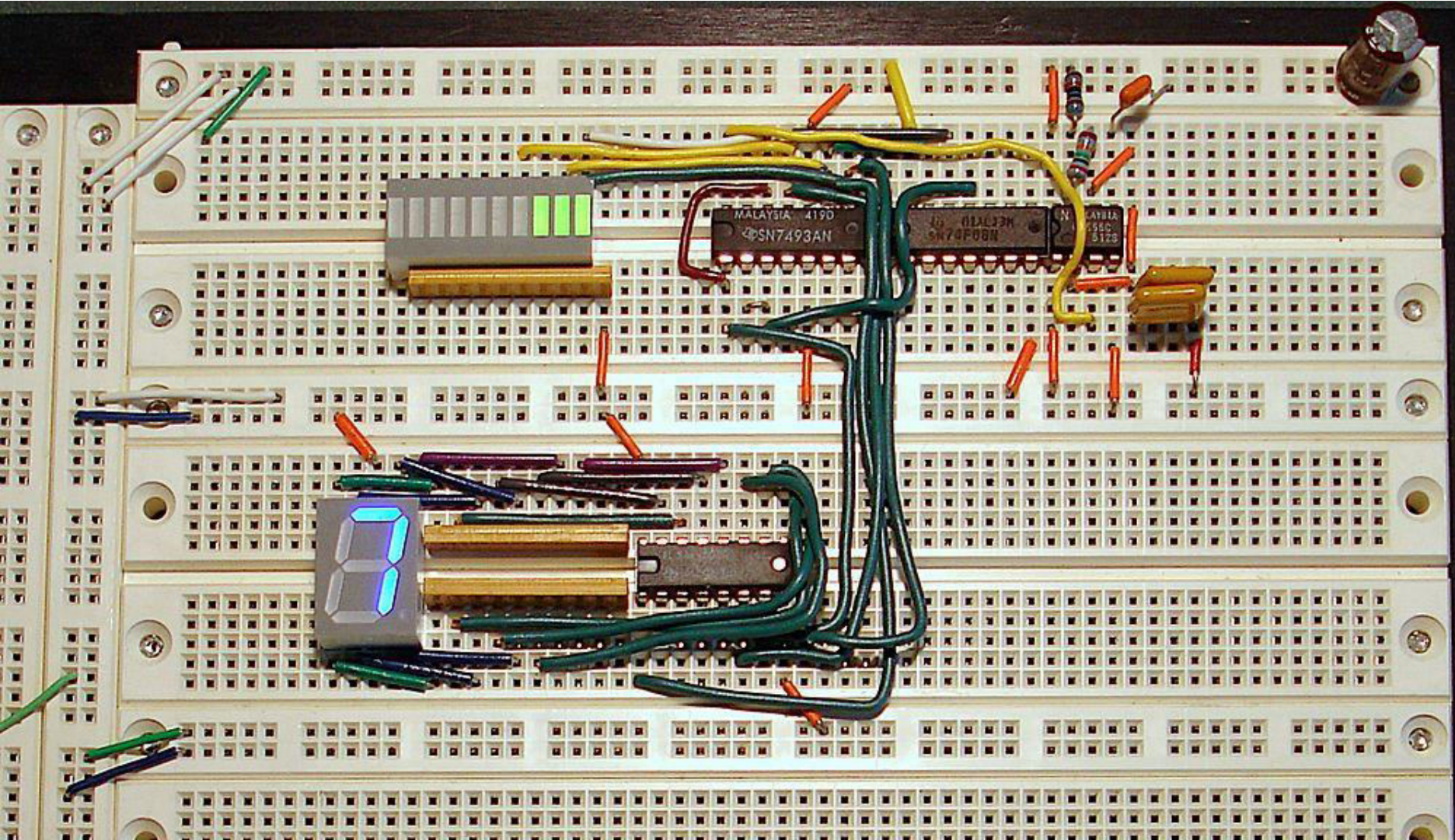


Slika levo:

By oomlout - Flickr: Wingshield on Arduino - ARSH-05-WI, CC BY-SA 2.0,  
<https://commons.wikimedia.org/w/index.php?curid=15911319>.

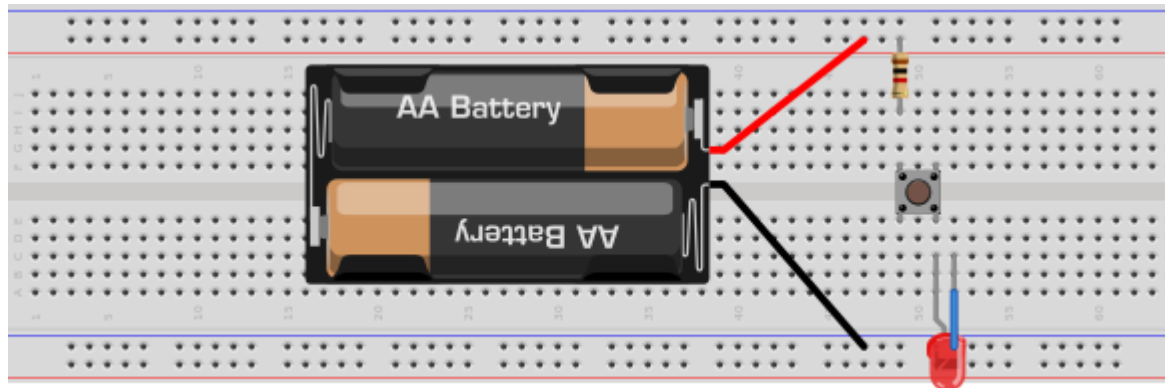
Slika desno:

By Marlon J. Manrique - Flickr: Arduino Protoboard Shield, CC BY-SA 2.0,  
<https://commons.wikimedia.org/w/index.php?curid=15916962>.

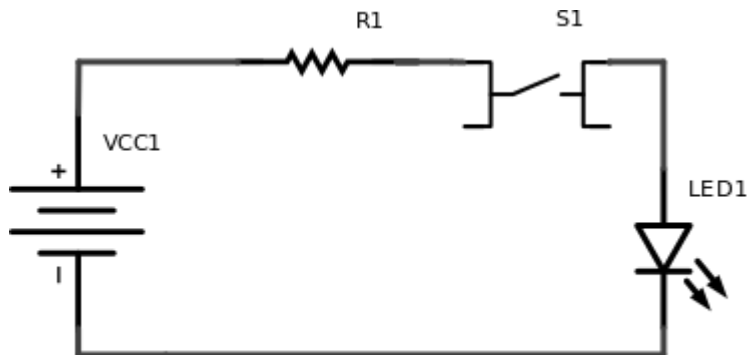


By en:User:Fulladder - en:Image:Breadboard counter.jpg, Copyrighted free use, <https://commons.wikimedia.org/w/index.php?curid=1776425>.

# Fritzing, <http://fritzing.org/home/>



Made with  Fritzing.org



Made with  Fritzing.org

Slika gore: By LA2 - Own work, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=22714341>.

Slika dole: By LA2 - Own work, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=22714342>.





# Ugrađen primer koda

## Blink

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  
  delay(1000); // wait for a second  
}
```

# Promenljive. Kako ih definisati?

```
int ledPin = 8;  
const int cledPin = 8;
```

- Ako su globalne:
  - Pre *setup()* funkcije. Mogu i pre *loop()* funkcije, ako se ne pozivaju u *setup()* funkciji.
  - Na slici je dato za celobrojne vrednosti (broj digitalnog pina).
  - Uobičajeno bi konstante trebalo da počinju slovom “c” kako je i prikazano na slici.
- Ako su lokalne:
  - Unutar funkcije (*setup()*, *loop()*) ili neke druge koju korisnik definiše)

# Kontrolne strukture (tok programa)

## Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

- Tipovi kontrolnih struktura su prikazani u tabeli.
- Detalje možete pogledati na: <https://www.arduino.cc/en/Reference/HomePage>, pristupljeno 11.10.2023.

# *if* struktura

```
int ledPin = 8;
void setup() {
  if (ledPin < 0) {
    ledPin = 13;
  }
  pinMode(ledPin, OUTPUT);
}
```

- Prikazana je na slici.
- Početak i kraj se označavaju vitičastim zagradama “{}”.

# *if else* struktura

```
if (pinLed >= 13) {  
    delay(200);  
}  
else {  
    delay(800);  
}
```

- Prikazana je na slici.
- Početak i kraj se označavaju vitičastim zagradama “{}”.
- Automatsko popunjavanje koda: kada se ukuca početak strukture tj. “{” i pritisne taster ENTER za prelazak u novi red, onda se automatski generiše i kraj strukture tj. “}”.

# switch struktura

```
switch (range) {  
  case 0:    // your hand is on the sensor  
    Serial.println("dark");  
    break;  
  case 1:    // your hand is close to the sensor  
    Serial.println("dim");  
    break;  
  case 2:    // your hand is a few inches from the sensor  
    Serial.println("medium");  
    break;  
  case 3:    // your hand is nowhere near the sensor  
    Serial.println("bright");  
    break;  
}
```

- Primer korišćenja ove strukture je prikazan na slici.
- Prikazan je deo ugrađenog koda iz *switchCase.ino*.

# *while* petlja

```
// while the button is pressed, take calibration readings:  
while (digitalRead(buttonPin) == HIGH) {  
    calibrate();  
}
```

- Primer *while* petlje je prikazan na slici.
- Pokazan je ugrađen primer *WhileStatementConditional.ino*.
- Kakva se petlja dobija ako je uslov *while(1)*?
- Kakva se petlja dobija ako je uslov *while(303)*?



# *for* petlja

```
const int kPinLed = 13;

void setup()
{
  pinMode(kPinLed, OUTPUT);
}

void loop()
{
  for(int i = 0; i < 4; i++){
    digitalWrite(kPinLed, HIGH);
    delay(200);
    digitalWrite(kPinLed, LOW);
    delay(200);
  }
  delay(1000); // 1 second
}
```

- Primer *for* petlje je prikazan na slici.
- Šta je rezultat koda sa slike?
- Kod je pruzet iz knjige: Alan G. Smith, Introduction to Arduino: A piece of cake!, [Online], <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.

# Šta je rezultat koda sa slike?

```
void loop()
{
    delayTime = delayTime - 100;
    if(delayTime <= 0){ // If the delay time is zero or ↵
        ↵ less, reset it.
        delayTime = 1000;
    }
    digitalWrite(kPinLed, HIGH);
    delay(delayTime);
    digitalWrite(kPinLed, LOW);
    delay(delayTime);
}
```

- Kod je pruzet iz knjige: Alan G. Smith, Introduction to Arduino: A piece of cake!, [Online], <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.
- Na slici nije prikazana *setup()* funkcija, bez definisanje globalnih promenljivih tj. prikazan je samo deo koda.
- Moguće je da ovakav zadatak i sl. bude dat na ispitu sa ciljem da studenti/kinje opišu funkciju koda ili da predvide izlaz programa!



# Aritmetički operatori i poređenja

## Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- \* (multiplication)
- / (division)
- % (modulo)

## Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

- Operatori su prikazani na slikama.
- Više o operatorima na sajtu: <https://www.arduino.cc/reference/en/>, pristupljeno 11.10.2023.
- Celobrojno deljenje se realizuje sa znakom "/" ako su imenilac i brojilac celobrojne vrednosti.

# Logički i složeni (eng. *compound*) operatori

## Boolean Operators

- `&&` (and)
- `||` (or)
- `!` (not)

## Bitwise Operators

- `&` (bitwise and)
- `|` (bitwise or)
- `^` (bitwise xor)
- `~` (bitwise not)
- `<<` (bitshift left)
- `>>` (bitshift right)

## Compound Operators

- `++` (increment)
- `--` (decrement)
- `+=` (compound addition)
- `-=` (compound subtraction)
- `*=` (compound multiplication)
- `/=` (compound division)
- `%=` (compound modulo)
- `&=` (compound bitwise and)
- `|=` (compound bitwise or)

- Operatori su prikazani na slikama.
- Više o operatorima na sajtu: <https://www.arduino.cc/reference/en/>, pristupljeno 11.10.2023.

# Šta je rezultat koda sa slike?

```
int ledState = LOW;

void loop()
{
    ledState = !ledState;    // toggle value of ledState
    digitalWrite(kPinLed, ledState);
    delay(1000);
}
```

- Kod je pruzet iz knjige: Alan G. Smith, Introduction to Arduino: A piece of cake!, [Online]. <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.
- LOW i HIGH su podrazumevane logičke promenljive koje imaju vrednosti FALSE i TRUE.



# Tipovi podataka

## Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- short
- float
- double
- string - char array
- String - object
- array

- Tipovi podataka su prikazani u tabeli.
- Detalji se mogu pogledati na: <https://www.arduino.cc/>, pristupljeno 11.10.2023.
- Konverzija se postiže primenom odgovarajućih funkcija.

## Conversion

- char()
- byte()
- int()
- word()
- long()
- float()





# Promena rezolucije

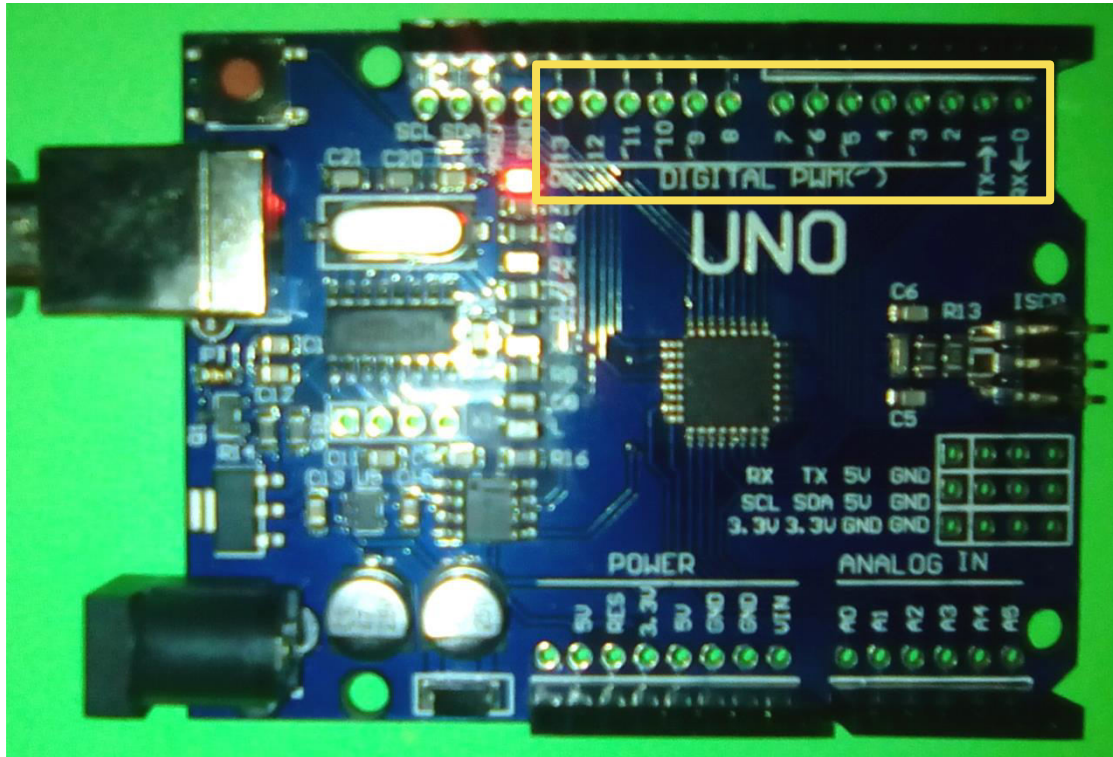
```
// change the resolution to 16 bits and read A0
analogReadResolution(16);
Serial.print(", 16-bit : ");
Serial.print(analogRead(A0));

// change the resolution to 8 bits and read A0
analogReadResolution(8);
Serial.print(", 8-bit : ");
Serial.println(analogRead(A0));
```

- Nije moguća na mikrokontrolerskim pločama koje se koriste na laboratorijskim vežbama (UNO R3 i sličnim).
- Međutim, moguća je na Arduino Due, Arduino Zero i drugim mikrokontrolerskim pločicama.
- Koristi se funkcija *analogReadResolution()*.
  - Više o ovoj funkciji na <https://www.arduino.cc/en/Reference/AnalogReadResolution>, pristupljeno 11.10.2023. (slika je takođe preuzeta sa ovog sajta).
- U principu, kod neće javiti grešku ako se povećava rezolucija iznad hardverskih granica, ali će ti bitovi biti prazni.
  - Ovakav kod ima smisla pisati kada se planira povećanje hardverskog kapaciteta.



# Pulse Width Modulation



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

- Impulsno širinska modulacija!
- Umesto *digitalWrite()* koristi se *analogWrite()* funkcija. Funkcija *digitalWrite()* je korišćena kod *Blink.ino* primera.
- Na pločici piše pored digitalnih pinova PWM (pogledati sliku).
- To znači mogućnost da se upravlja sa *duty cycle* tj. faktorom ispunjenosti impulsa.

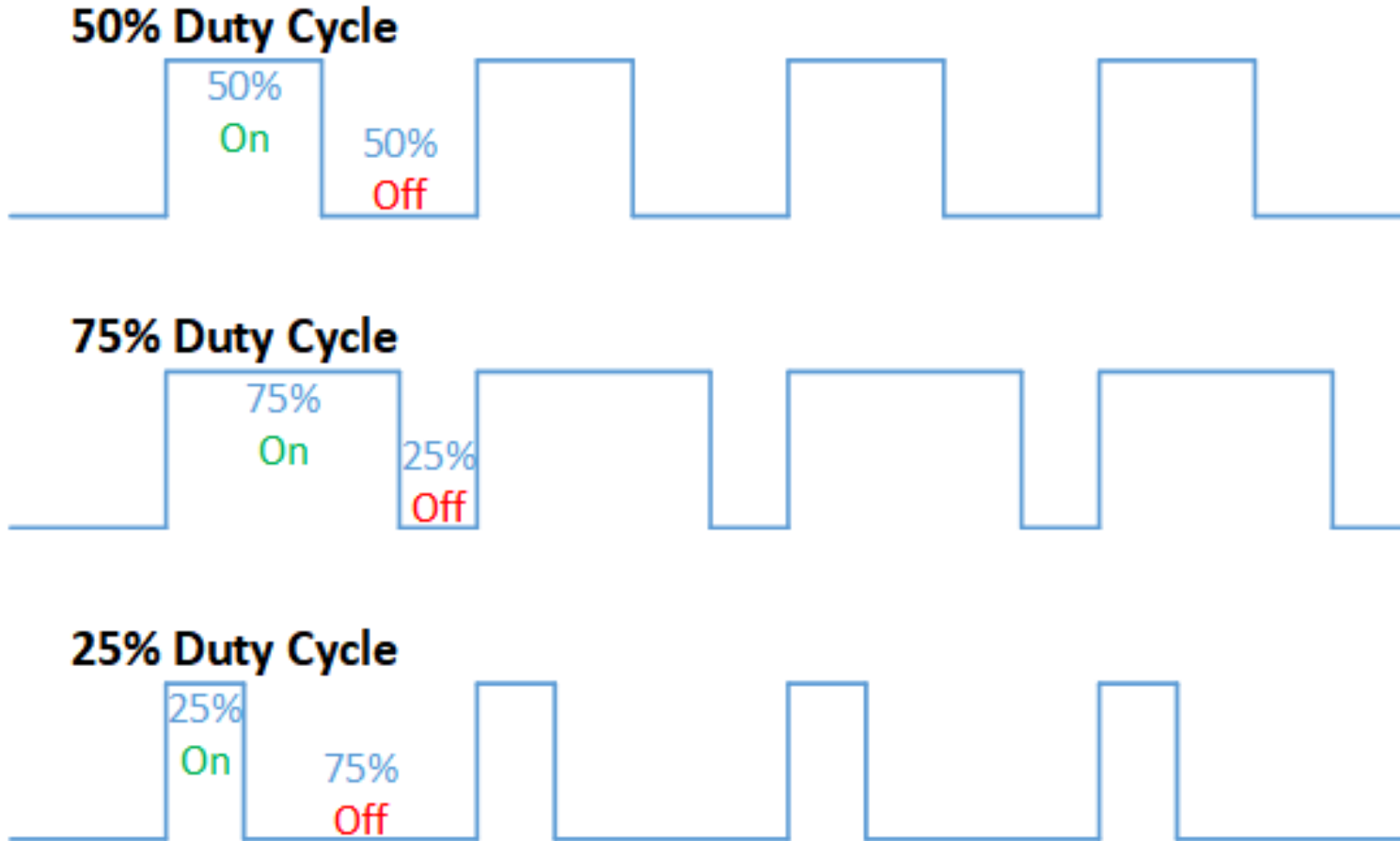
# PWM definicija

- “Pulse Width Modulation (PWM) is a fancy term for describing a type of digital signal. Pulse width modulation is used in a variety of applications including sophisticated control circuitry” (izvor: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>, pristupljeno 11.10.2023).
- “Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique used to encode a message into a pulsing signal” (izvor: [https://en.wikipedia.org/wiki/Pulse-width\\_modulation](https://en.wikipedia.org/wiki/Pulse-width_modulation)).
- “Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off” (izvor: <https://docs.arduino.cc/learn/microcontrollers/analog-output>, pristupljeno 11.10.2023).

# PWM definicija

- “Pulse Width Modulation (PWM) is a fancy term for describing a type of digital signal. Pulse width modulation is used in a variety of applications including sophisticated control circuitry” (izvor: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>, pristupljeno 11.10.2023).
- “Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique used to encode a message into a pulsing signal” (izvor: [https://en.wikipedia.org/wiki/Pulse-width\\_modulation](https://en.wikipedia.org/wiki/Pulse-width_modulation)).
- **“Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means.** Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off” (izvor: <https://docs.arduino.cc/learn/microcontrollers/analog-output>, pristupljeno 11.10.2023).

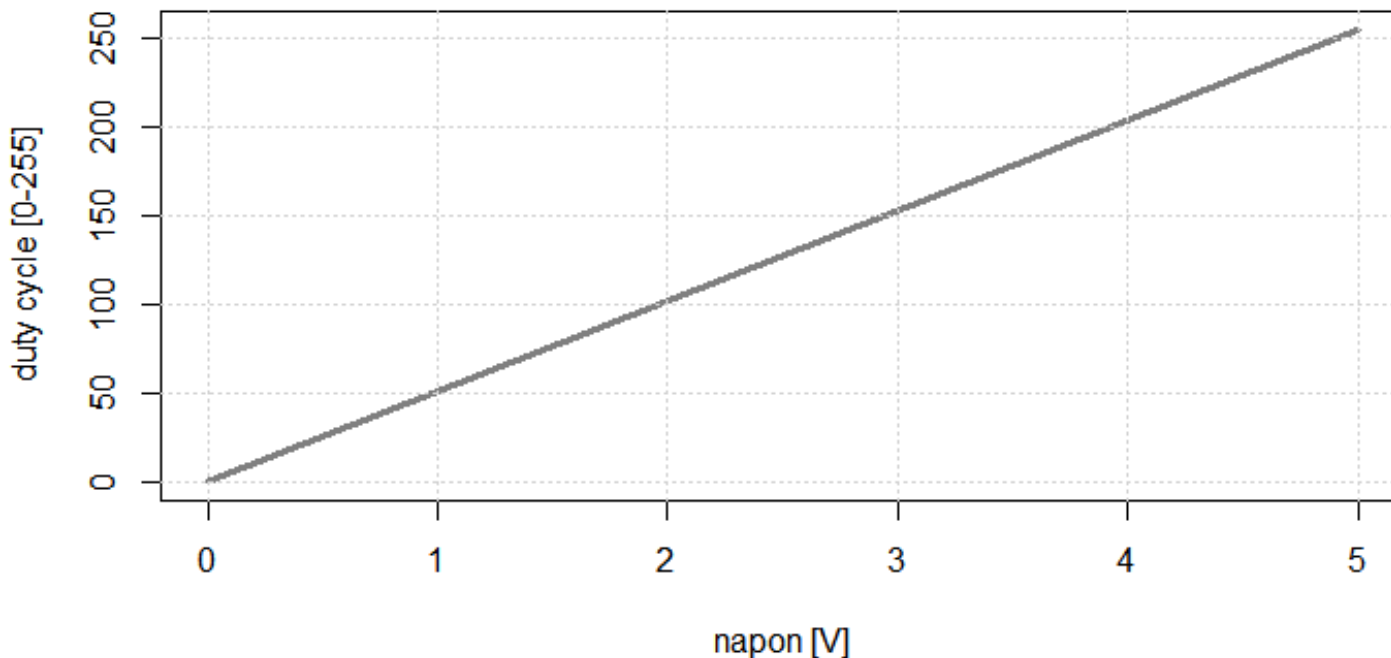
# PWM u praksi



Slika: By Thewrightstuff - Own work, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=72876123>.

# PWM vrednosti 0-255 na 0-5 V

Linearna zavisnost napona od faktora ispunjenosti impulsa



- Linearna skala je prikazana na slici.
- Skala je nacrtana u R-u (<https://www.r-project.org/about.html>, pristupljeno 11.10.2023).
- Opseg od 0 do 255 se odnosi na *duty cycle* tj. faktor ispunjenosti impulsa.



# *analogWrite()* primer

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

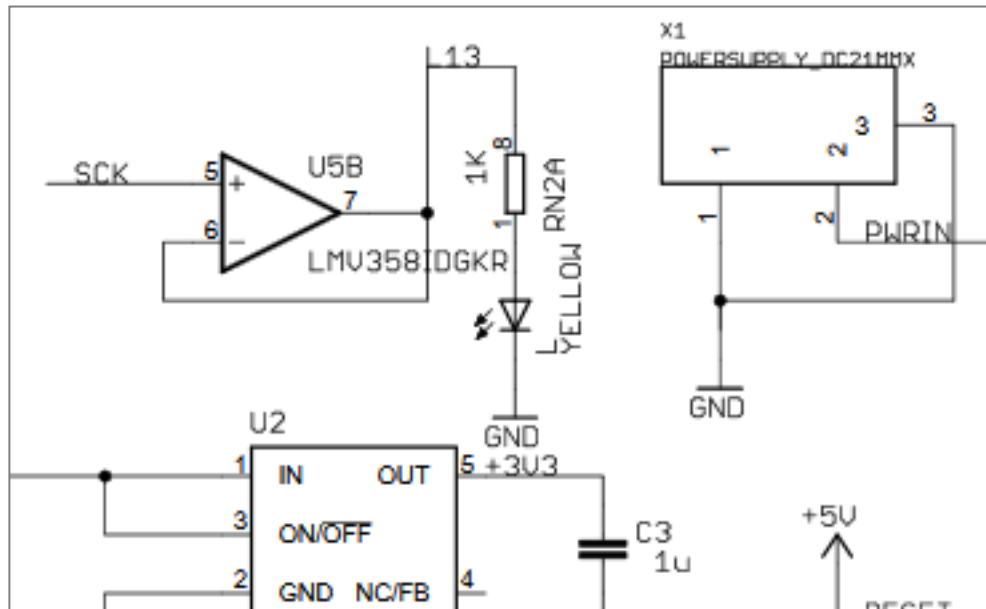
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

- Ugrađeni primer *Fade.ino* je prikazan na slici.
- Šta je funkcija ovog primera?
- Koristi se za kontrolu osvetljenosti (intenziteta svetlosti na diodi), ali i za kontrolu motora.



# Ugrađena dioda



- Na pinu 13 kod UNO R3 mikrokontrolerske pločice se nalazi ugrađena SMD (eng. *surface-mount technology*) dioda (oznaka "L" na pločici).
- Na slici je prikazana šema UNO pločice, gde je pokazano da je ugrađena dioda na pinu 13, [https://www.arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf) (pristupljeno 11.10.2023.) sa oznakom "L".
- Podrazumevano pin 13 je ON. Više do kraja semestra i na <https://docs.arduino.cc/learn/microcontrollers/digital-pins>, pristupljeno 11.10.2023.
- Da li postoji ugrađen otpornik koji služi da ograniči struju kroz ovu diodu?
- Da li je potrebno da se doda otpornik, ako se vezuje eksterna dioda na pin 13?

# Arduin od Ivreje



- Bio je kralj Italije od 1002. do 1014. godine.
- Jedan bar u Ivreji nosio je ime kralja Arduina u kome su se kreatori projekta okupljali.
- Zato su svoju mikrokontrolersku pločicu nazvali Arduino.
- Ideja ovog projekta je bila da se omogući studentima i amaterima da jednostavno mogu da koriste senzore i aktuatore.
- Većina Arduino pločica se sastoji od 8-bitnih mikrokontrolera proizvođača Atmel (ATMega8, ATmega168, ATmega328, ATmega1280 ili ATmega2560), a 32-bitni Arduino Due je zasnovan na Atmel SAM3X8E mikrokontroleru.
- Više na <https://en.wikipedia.org/wiki/Arduino>.

Slika: CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17242>, crkva Sv. Arhangela Mihajla u Paviji u kojoj je Arduin proglašen za kralja.

# Arduino kompatibilnost

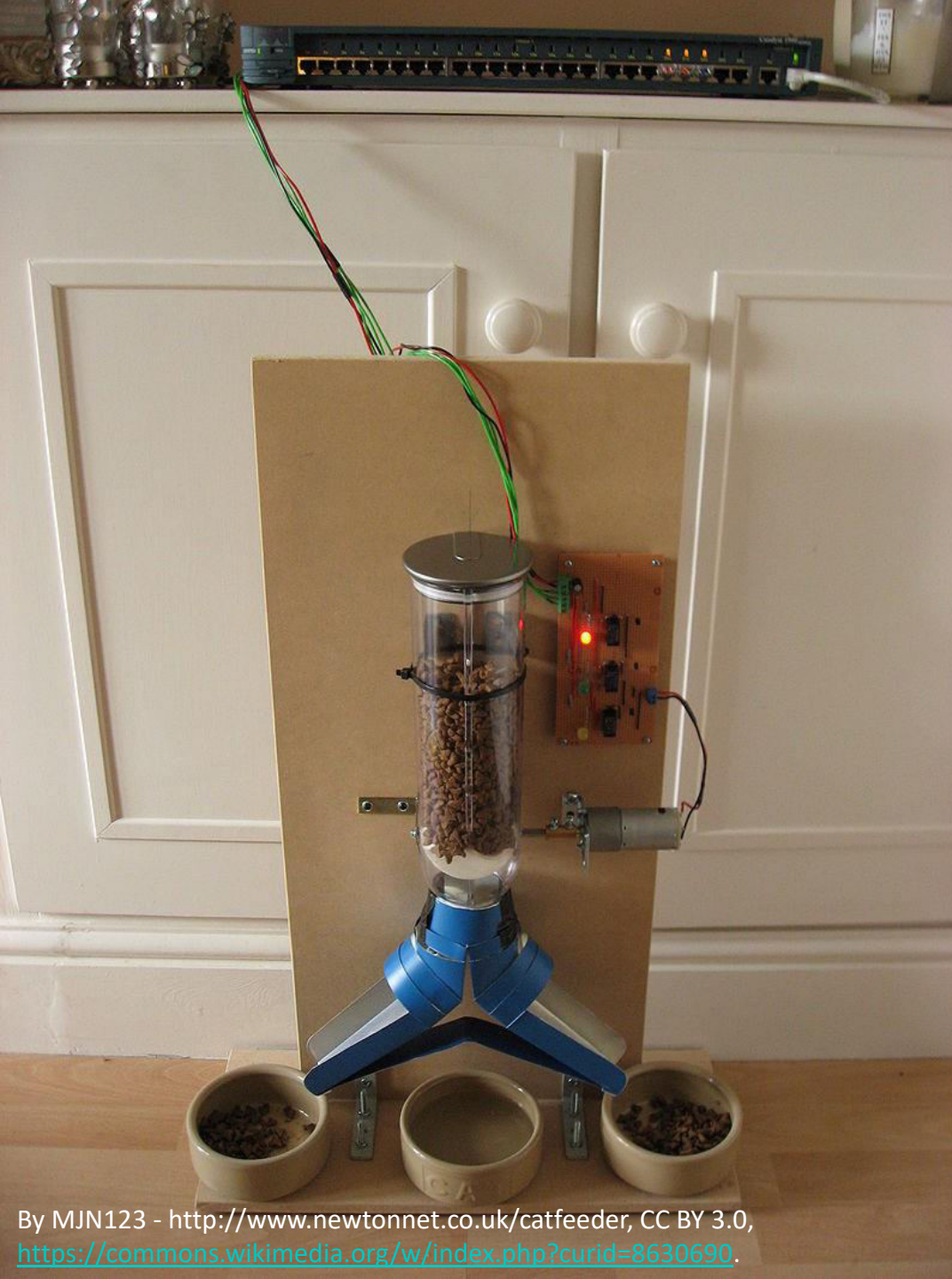
- Koliko je Arduino popularan svuda, a ne samo u nastavi i za hobiste, pokazuju i drugi proizvođači hardvera.
- Na primer, STM32 je familija 32-bitnih mikrokontrolera sa 32-bitnim ARM procesorom
  - ARM je skup instrukcija koju je razvija kompanija ARM Holding za procesore sa RISC arhitekturom. Poznato?
- STM32 je proizvod kompanije STMicroelectronics ([https://www.st.com/content/st\\_com/en.html](https://www.st.com/content/st_com/en.html), pristupljeno 11.10.2023.) koja je osnovana 1957. sa sedištem u Ženevi u Švajcarskoj.
- Njihovi proizvodi imaju Arduino kompatibilnost – u smislu rasporeda pinova i za korišćenje šildova.
- Pogledati Nucleo proizvode, <https://www.st.com/en/ecosystems/stm32-nucleo.html>, pristupljeno 11.10.2023.

# mBED okruženje

- Pored Arduino okruženja na kome se rade laboratorijske vežbe na MSR predmetu, mogu se koristiti i druga okruženja za programiranje mikrokontrolera.
- *Free and open – source* okruženje za programiranje mikrokontrolera je mBED, <https://os.mbed.com/>, pristupljeno 11.10.2023.
- Ovo je *online editor* i kompajler za programiranje mikrokontrolera i posebno namenjen IoT (eng. *Internet of Things*, internet stvari) sistemima
- Napisan je u C i C++
- Podržava preko 170 mikrokontrolera
- Jedne godine korišćen na Izazovu
- IoT, poznato?
  - Na sledećim slajdovima objašnjeno je šta je IoT, a slajdovi su preuzeti iz prezentacije "[IoT & električna merenja](#)" održane 26. marta 2019. tokom EESTech Challenge-a na ETF-u u Beogradu

# Internet stvari (IoT)

- Dostupnost interneta je omogućila direktno vezivanje objekata na mrežu tzv. *Internet of Things* (IoT).
- IoT, po definiciji, čini mrežu svakodnevnih objekata koji su umreženi sa računarima i mobilnim uređajima ([https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things), 23.03.2019).
- Popularne IoT primene su kućna automatizacija (eng. *home automation, domotics, smart home, smart house*, [https://en.wikipedia.org/wiki/Home\\_automation](https://en.wikipedia.org/wiki/Home_automation), 23.03.2019.) i sigurnosni nadzor.



# Stvari?

- Šta?
  - Kućna tehnika (frižider, šporet, mikser, bojler, televizor, ...)
  - Senzori (kamera, senzor pokreta, senzor temperature, senzor vlažnosti, ...)
  - ...
- Gde?
  - U kući
  - Na poslu
  - Na ulici
  - U automobilu
  - Na kućnim ljubimcima
  - ...



# Zašto?

- Automatizacija dosadnih poslova
- Komfor
- Igračka
- Zabava
- Nadzor
- Sigurnost
- Zdravlje
- Industrija
- ...



# Ali i biljke.



← → ↻ 🏠 <https://twitter.com/ABotanicar>

🏠 Почетна 🔔 Обавештења ✉ Поруке

Твитови 4 Праћења 1 Пратиоци 3

**AutoBotanicar**  
@ABotanicar Прати те  
📅 Придружио/ла се децембар 2018.  
Твитуј Порука  
👤 2 пратилаца које знаш

**Твитови** Твитови и одговори

 **AutoBotanicar** @ABotanicar · 4. јан  
Мрачно је  
💬 ↻ ❤️ 1 ✉

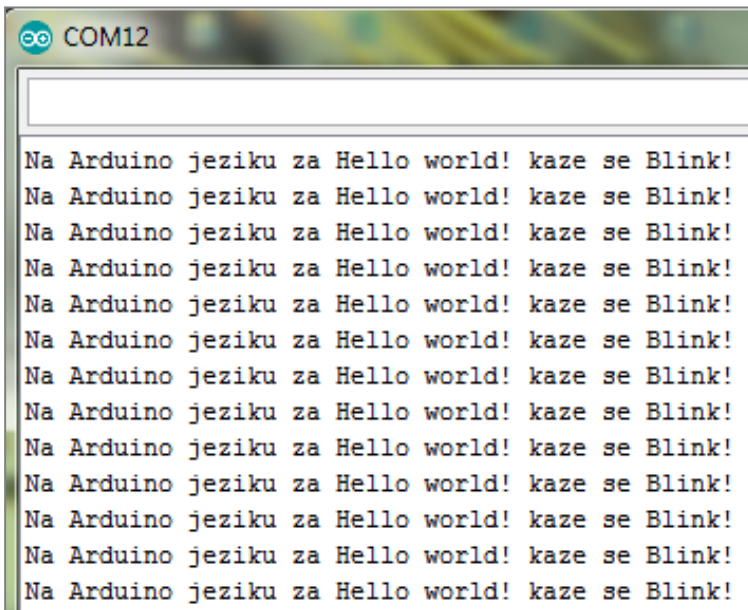
 **AutoBotanicar** @ABotanicar · 4. јан  
Sve je ok  
💬 ↻ ❤️ 1 ✉

Na slici je prikazano rešenje IZAZOVA u 2018. godini pod nazivom Auto Botaničar na 13E053MSR predmetu koje su radili Jovana Uskoković i Nikola Milenić.



# Primer

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Na Arduino jeziku za Hello world! kaze se Blink!");  
  delay(100);  
}
```



The screenshot shows a serial terminal window titled "COM12". The window contains a text area with the following output, repeated 13 times:

```
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!  
Na Arduino jeziku za Hello world! kaze se Blink!
```

- Na slici je prikazan primer ispisa poruke na serijskom portu.
- Dve osnovne funkcije za rad sa stringovima su *Serial.print()* i *Serial.println()*.
- Šta bi bilo da je iskorišćena funkcija *Serial.print()*?