

# Prižiganje lučk z Arduinom



# Kaj bomo danes počele?

Spoznale:

- Arduino Nano
  - Testno ploščico (Breadboard)
  - LED lučke, upornike, žičke, foto senzorje
  - Osnove programiranja v C++ (spremenljivke, funkcija, for zanka)
- 
- Koda: <https://github.com/22nds/lfu-arduino-basics>
  - Arduino IDE: <https://www.arduino.cc/en/Main/Software>

# Sestavni deli

- 2 x LED
  - 1 x RGB LED
  - 3 x 220 Ohm upornik
  - 1 x 1k Ohm upornik
  - 2 x žičke
  - 1 x senzor svetlobe
  - Testna ploščica (Breadboard)
- 

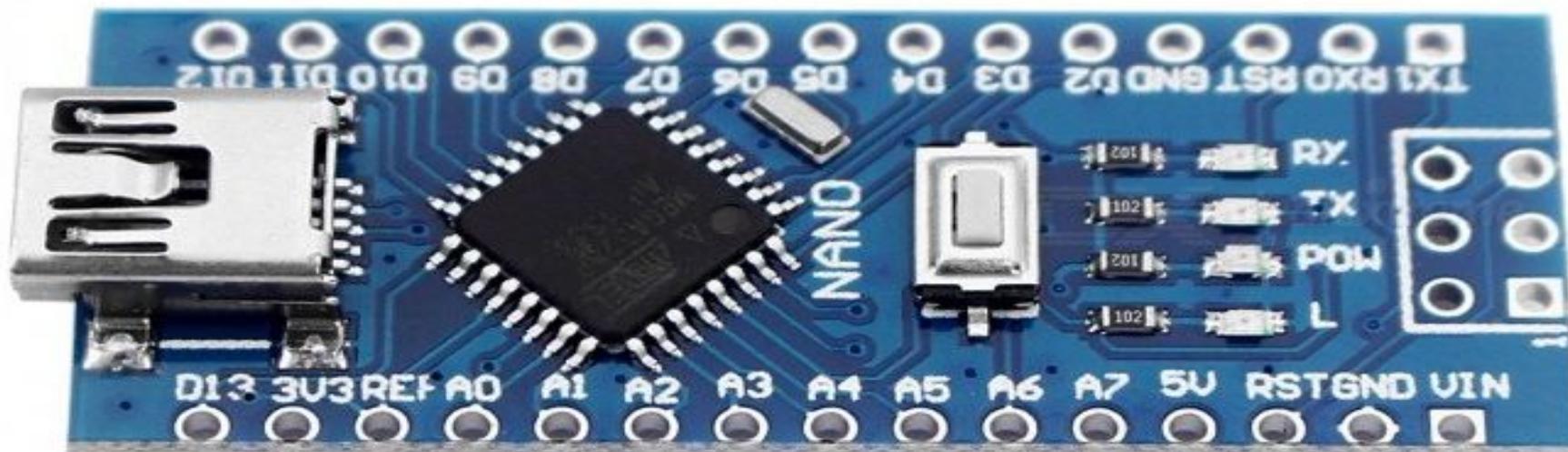
- USB kabel
- Računalnik
- Programska oprema (Arduino, Processing)

# Arduino Nano

Arduino Uno ->

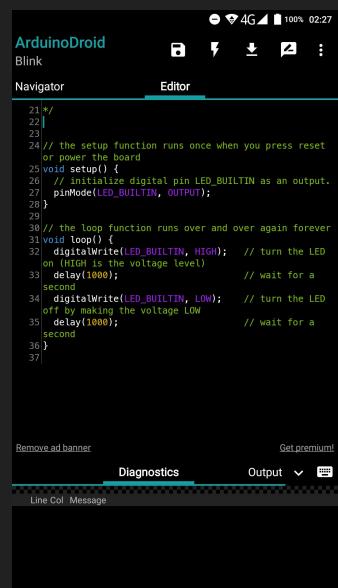
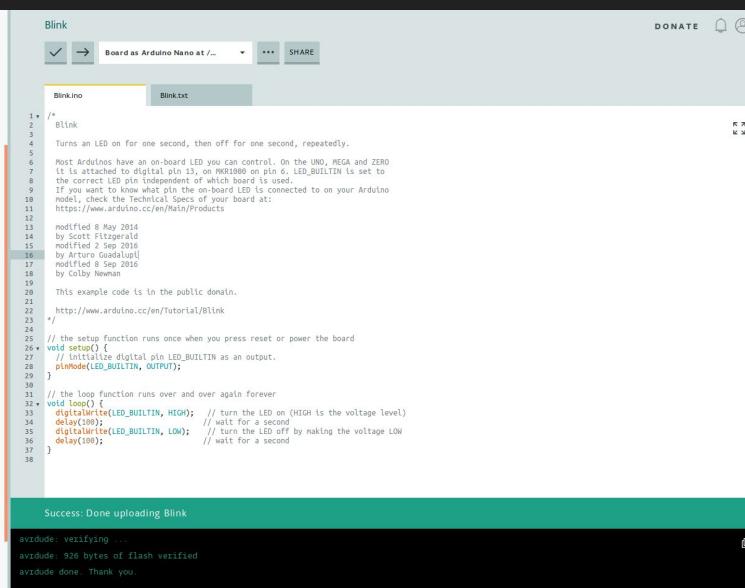
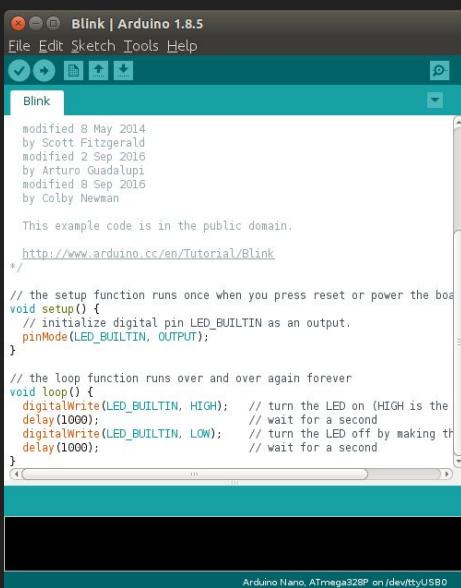


- Uradna stran: <https://store.arduino.cc/arduino-nano>

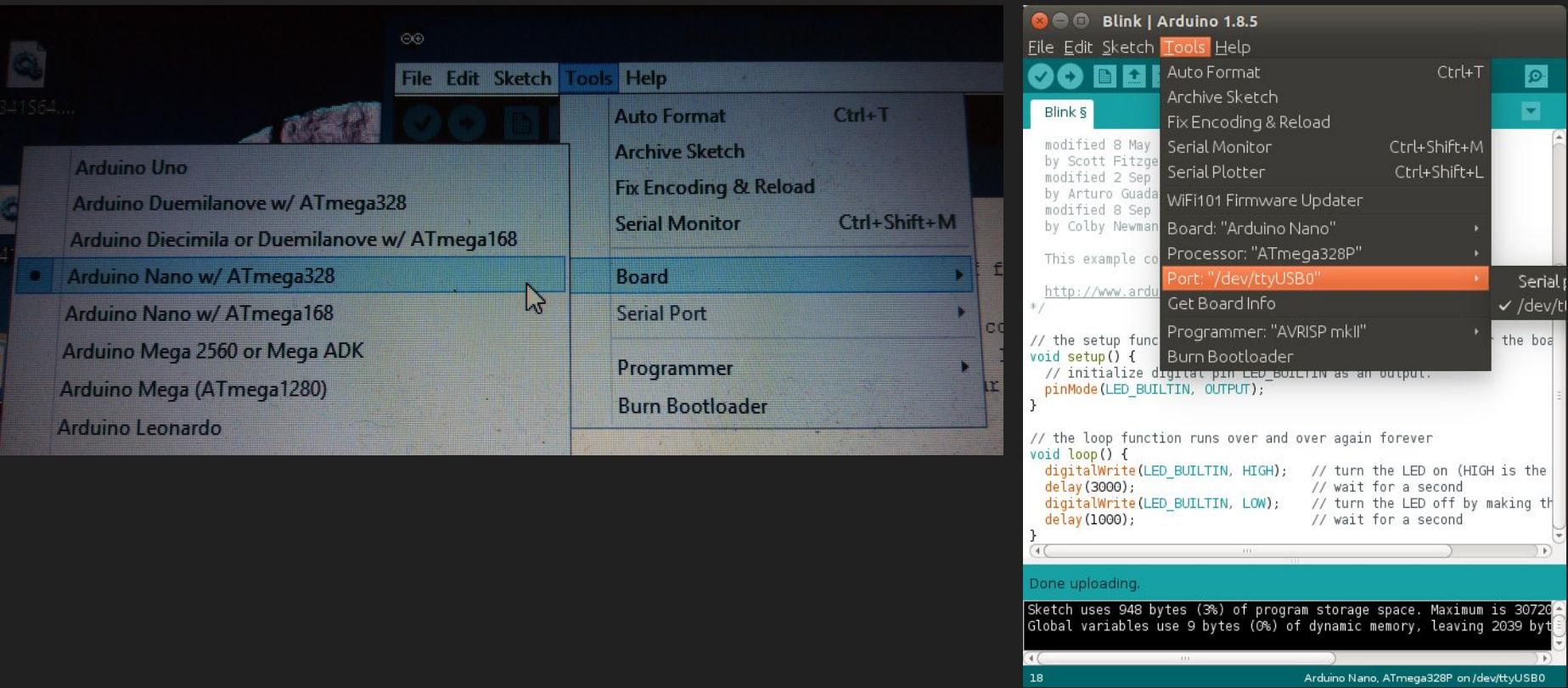


# Programska oprema za Arduino

- Arduino IDE <https://www.arduino.cc/en/Main/Software> ali
- Arduino Editor (online) <https://create.arduino.cc/editor/> ali
- ArduinoDroid (Android): <https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinodroid2>

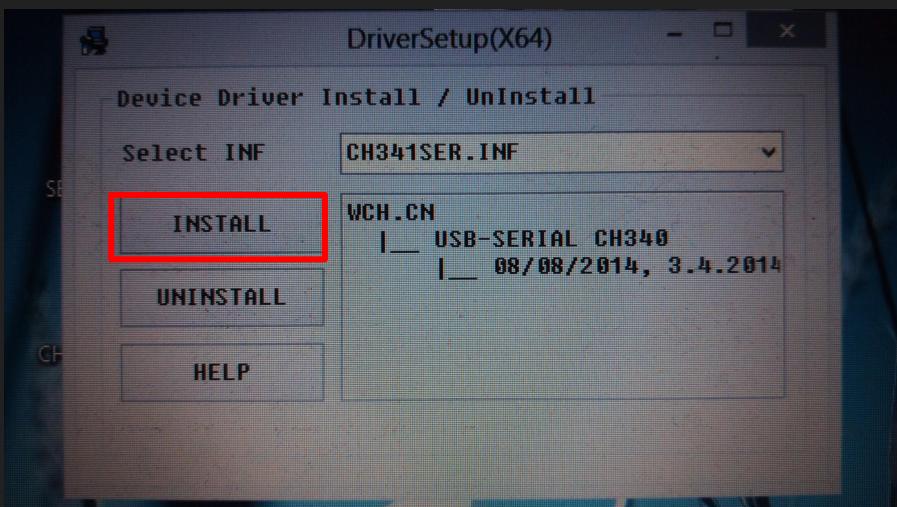


# Nastavitve za Arduino v Arduino IDE



# Namestitev gonilnikov za Windows

- [I:\driver\CH34x\Install\\_Windows\\_v3.4.zip](I:\driver\CH34x\Install_Windows_v3.4.zip) ali
- <http://sparks.gogo.co.nz/ch340.html> ali
- <http://www.arduined.eu/ch340-windows-8-driver-download/>



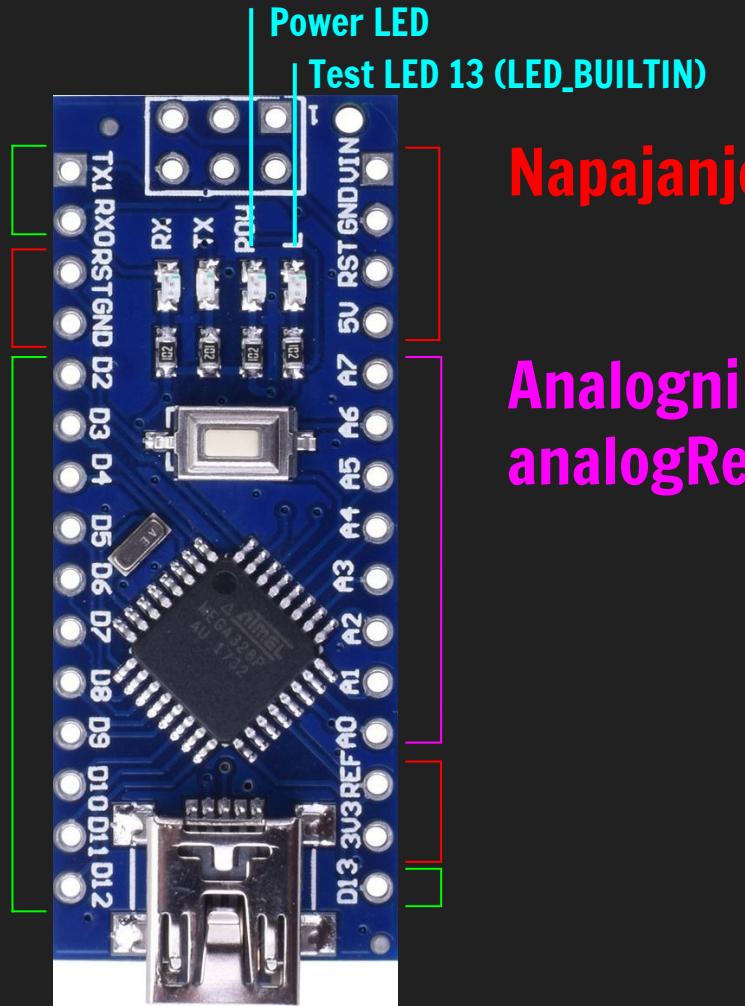
# Gonilniki za Mac

<https://kig.re/2014/12/31/how-to-use-arduino-nano-mini-pro-with-CH340G-on-mac-osx-yosemite.html>

Na Linuxu so gonilniki že nameščeni!



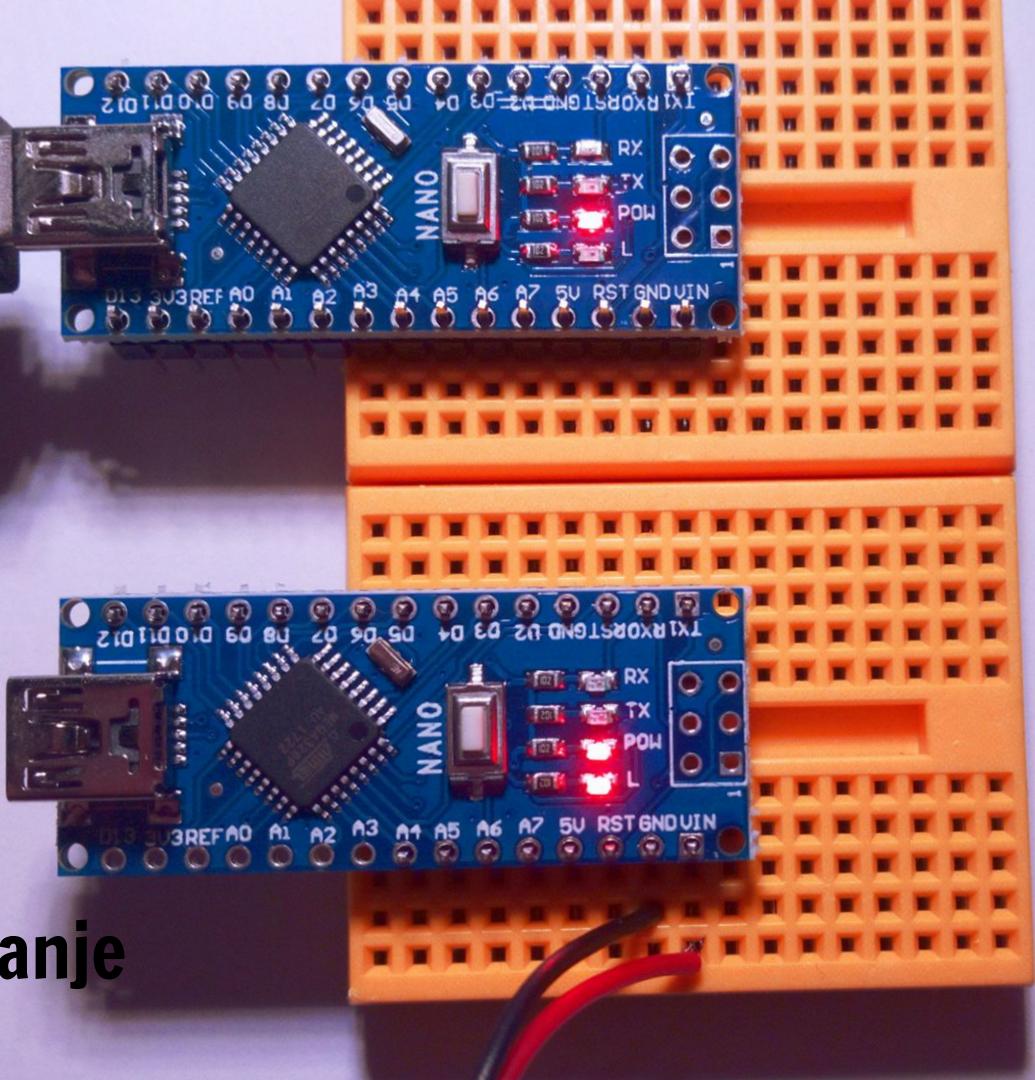
**Digitalni pini**  
pinMode()  
DigitalRead()  
DigitalWrite()  
**AnalogWrite()** - PWM  
3, 5, 6, 9, 10, 11



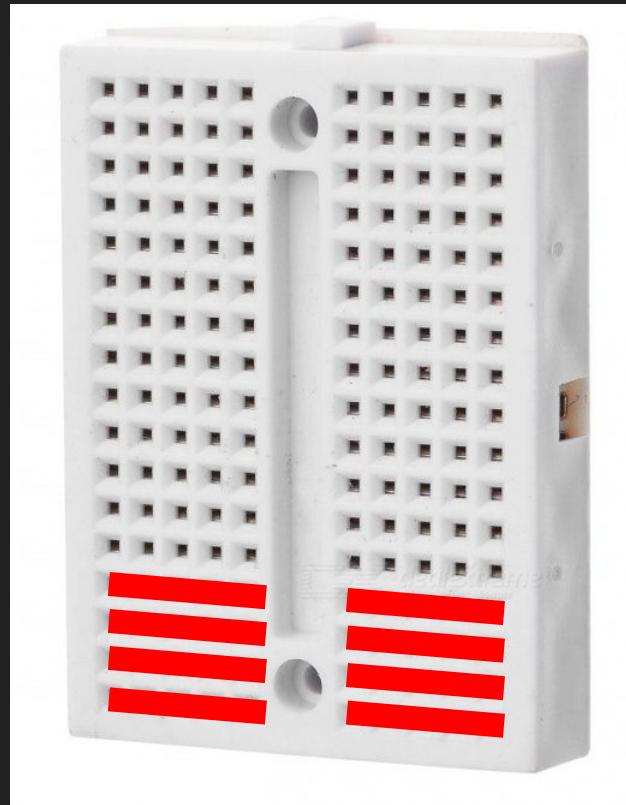
Napajanje

Analogni pini  
analogRead()

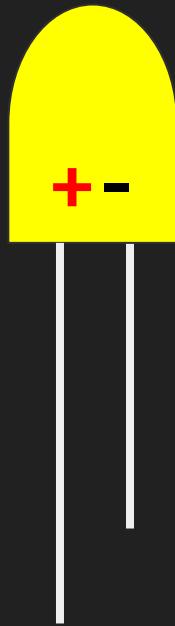
# Napajanje



# Testna ploščica



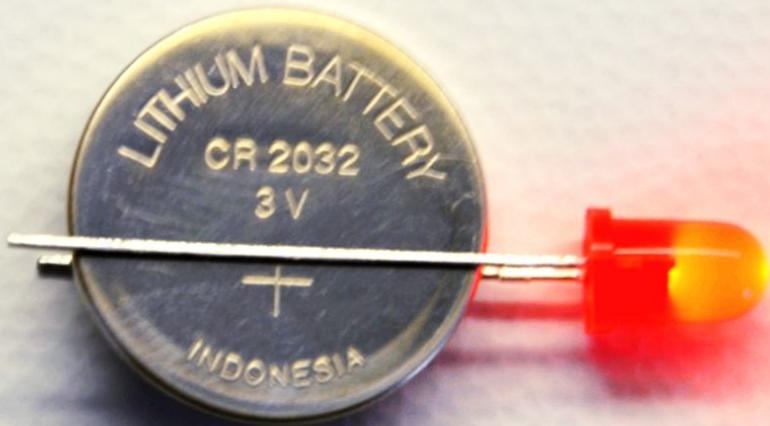
# LED lučke



LED



RGB LED



# Uporniki - moč in računanje upora

Upornik se upira električnemu toku in mu preprečuje, da bi nemoteno tekel skozenj.

Napetost (V) = Tok(I) \* Upor(R)

Upor: 220 Ohmov

Napetost: 5 Voltov

=====

Tok: 23 mA (mili amperov)

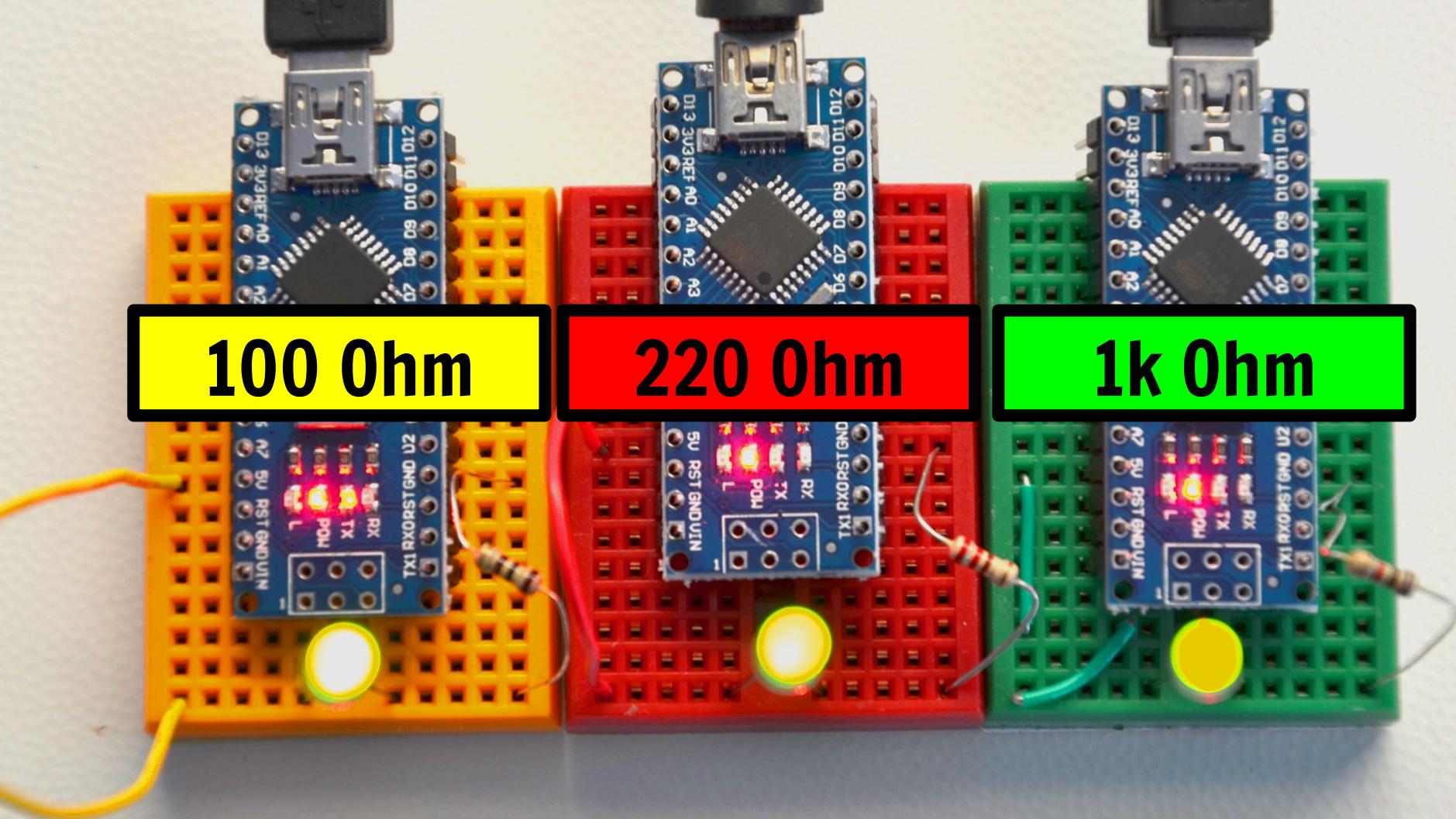
$$5V = 23 \text{ mA} * 220 \text{ Oh}$$

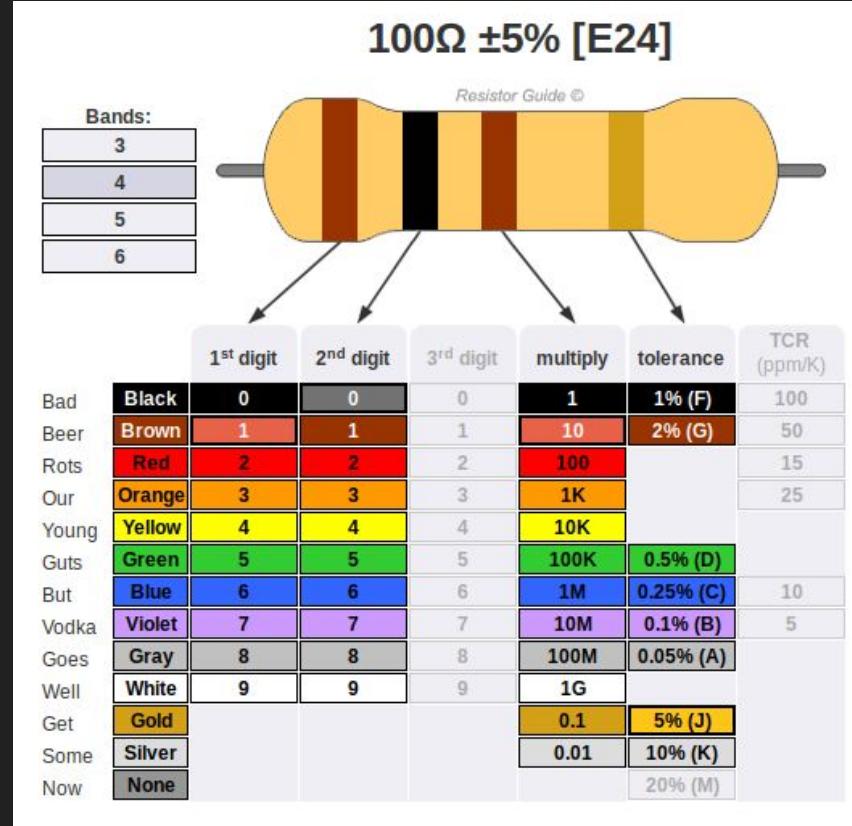
$$5V = 0.0227A * 220 \text{ Oh}$$

**100 Ohm**

**220 Ohm**

**1k Ohm**





<http://www.resistorguide.com/resistor-color-code-calculator/>

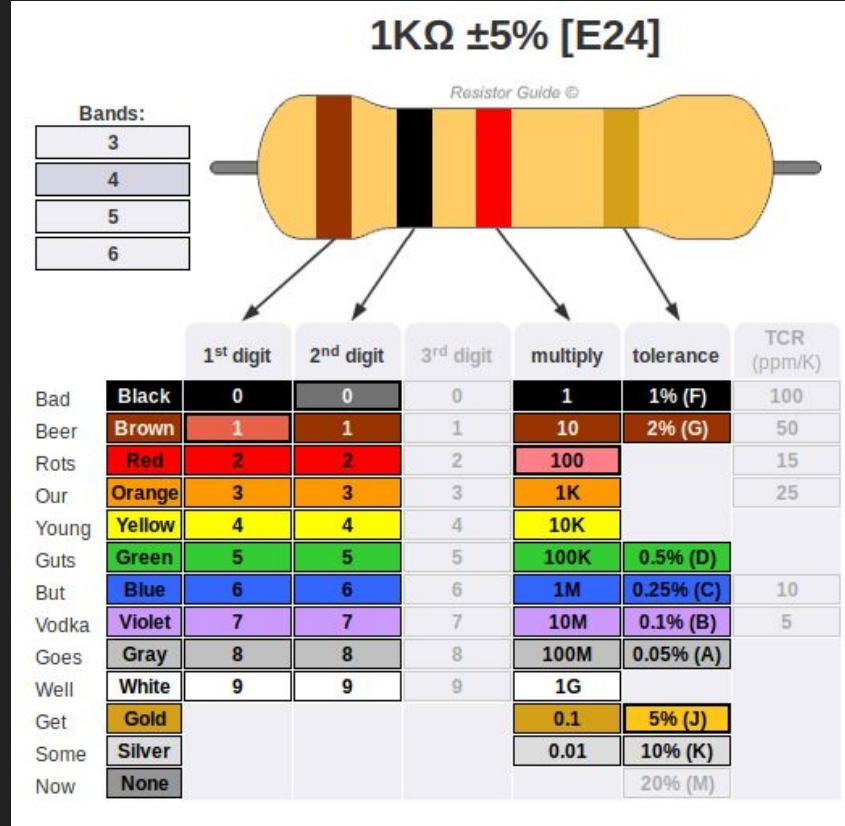
**$220\Omega \pm 5\% [E24]$**

Resistor Guide ©

The diagram illustrates the resistor color code. A resistor is shown with its four color bands. Arrows point from each band to a corresponding column in a table below. The first three bands represent the resistance value (220 ohms), and the fourth band represents the tolerance (+5%).

	1 <sup>st</sup> digit	2 <sup>nd</sup> digit	3 <sup>rd</sup> digit	multiply	tolerance	TCR (ppm/K)	
Bad	Black	0	0	0	1	1% (F)	100
Beer	Brown	1	1	1	10	2% (G)	50
Rots	Red	2	2	2	100		15
Our	Orange	3	3	3	1K		25
Young	Yellow	4	4	4	10K		
Guts	Green	5	5	5	100K	0.5% (D)	
But	Blue	6	6	6	1M	0.25% (C)	10
Vodka	Violet	7	7	7	10M	0.1% (B)	5
Goes	Gray	8	8	8	100M	0.05% (A)	
Well	White	9	9	9	1G		
Get	Gold				0.1	5% (J)	
Some	Silver				0.01	10% (K)	
Now	None					20% (M)	

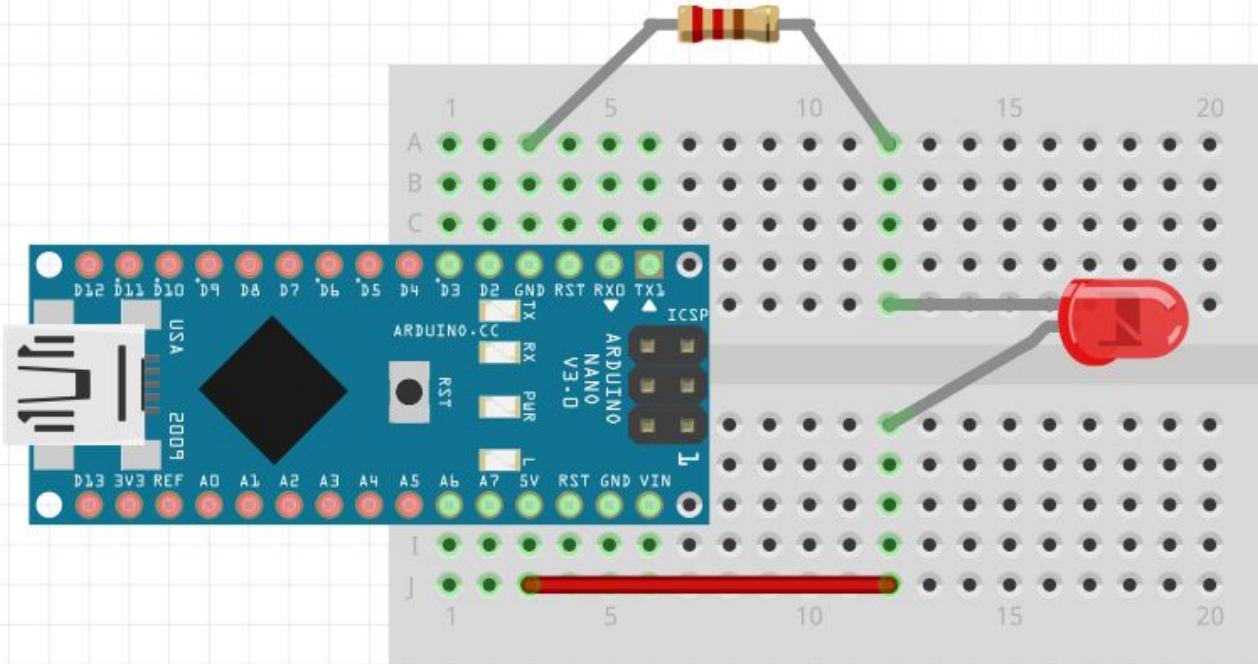
<http://www.resistorguide.com/resistor-color-code-calculator/>



<http://www.resistorguide.com/resistor-color-code-calculator/>

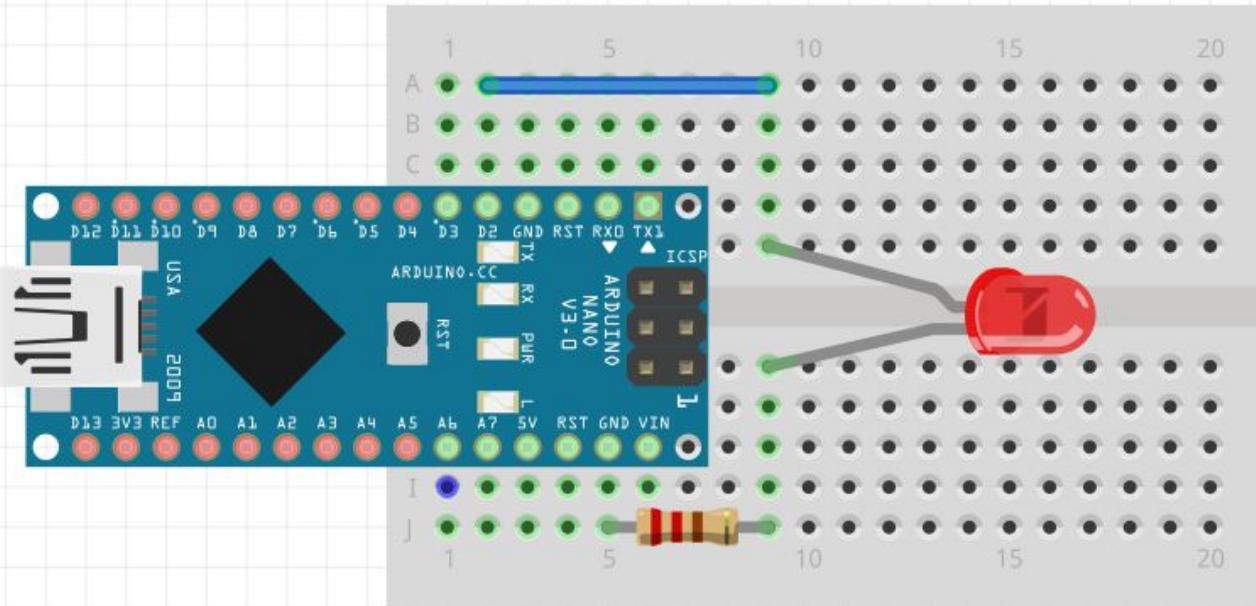
[ 00 ]

# Lučka brez programa



[01]

# Utripanje lučke



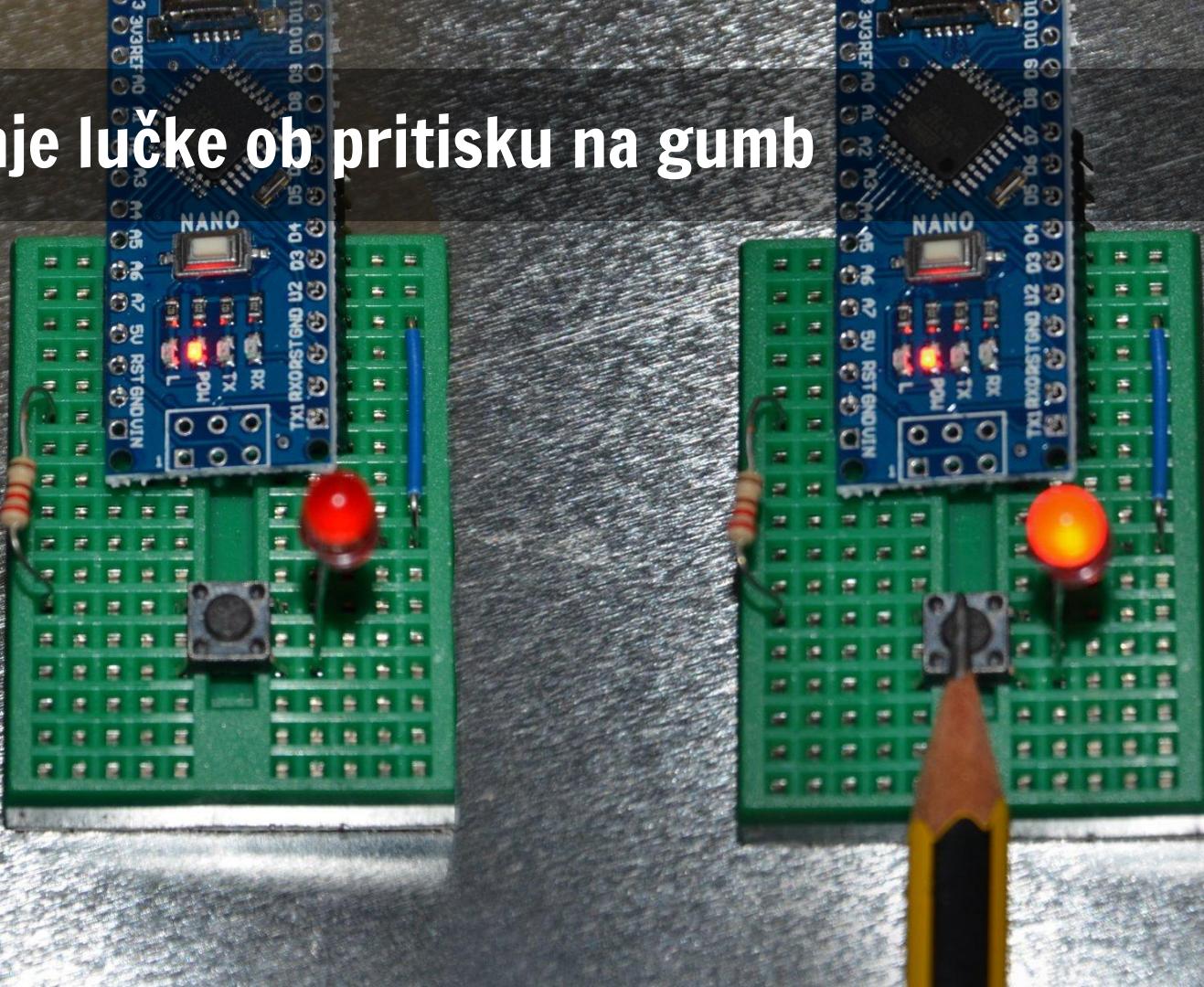
```
int LED = 2;

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

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

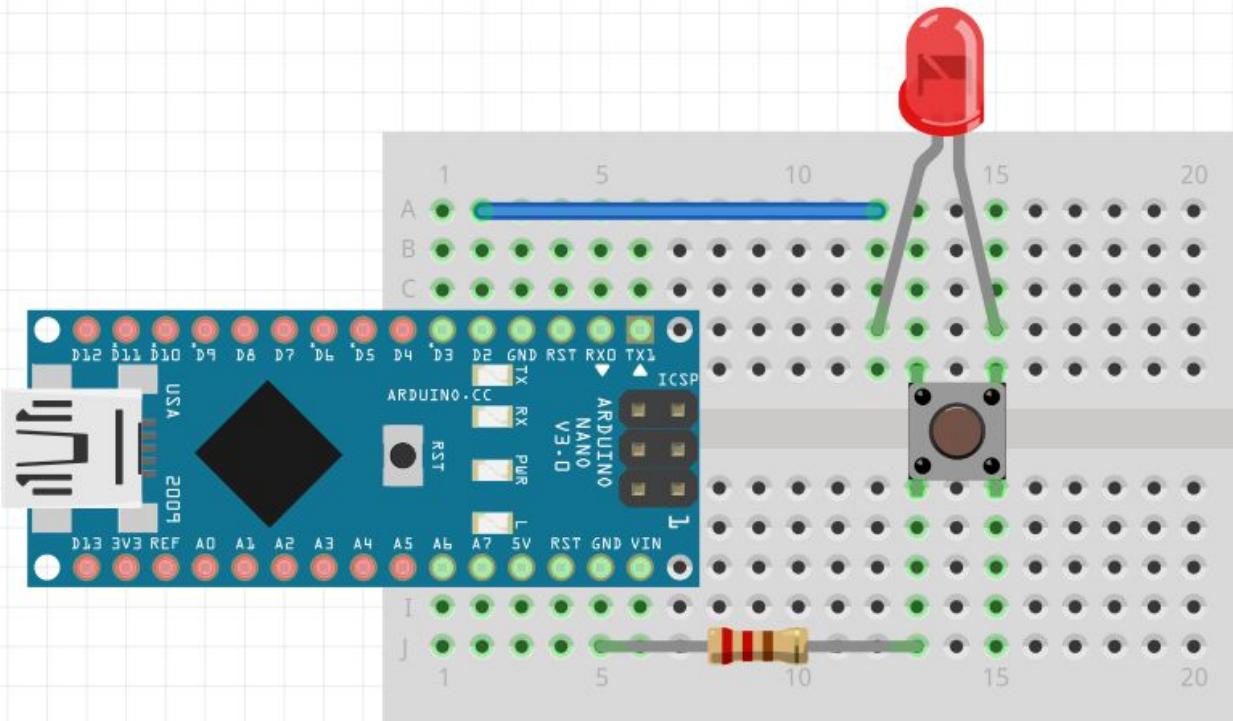
# Utripanje lučke ob pritisku na gumb

[02]



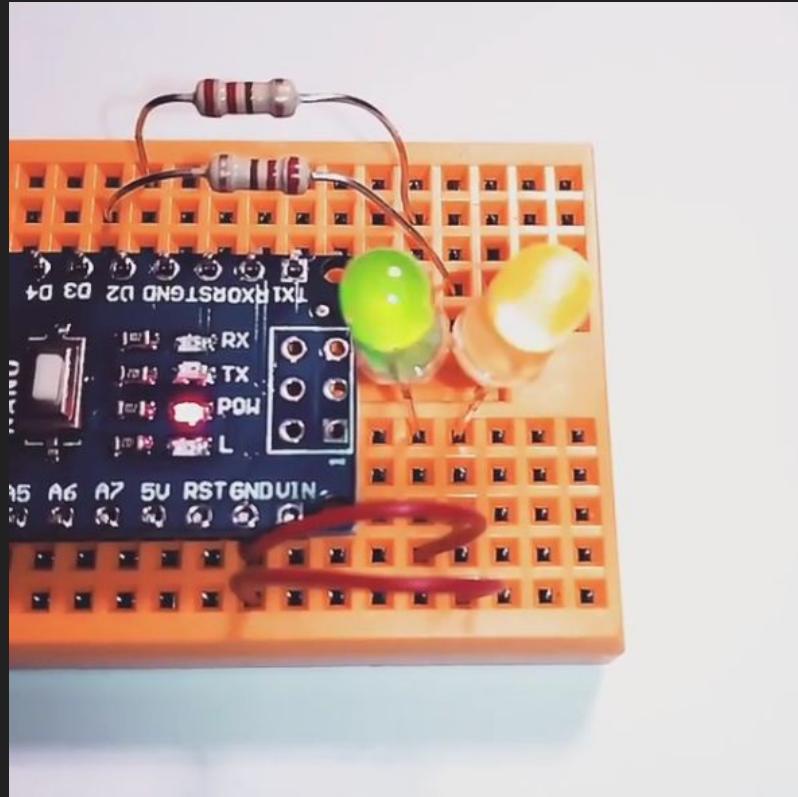
[02]

# Utripanje lučke ob pritisku na gumb

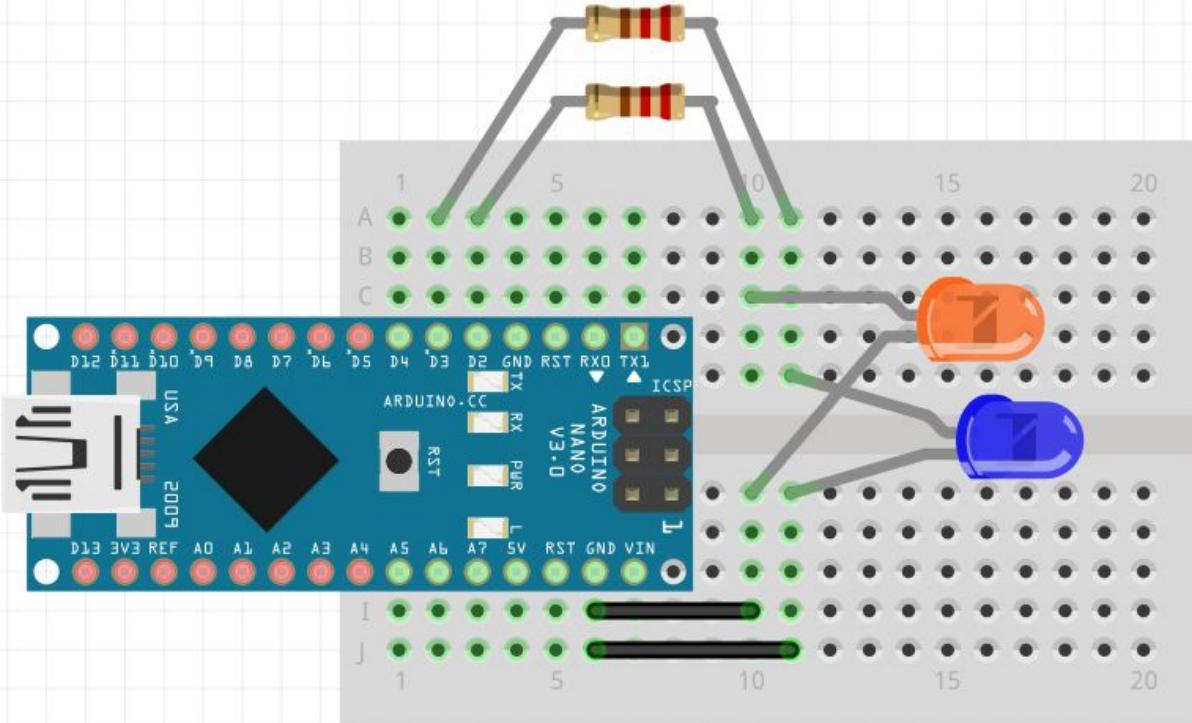


[03]

# Izmenično utripanje LED lučk



# Izmenično utripanje



```
int led_one = 2;
int led_two = 3;

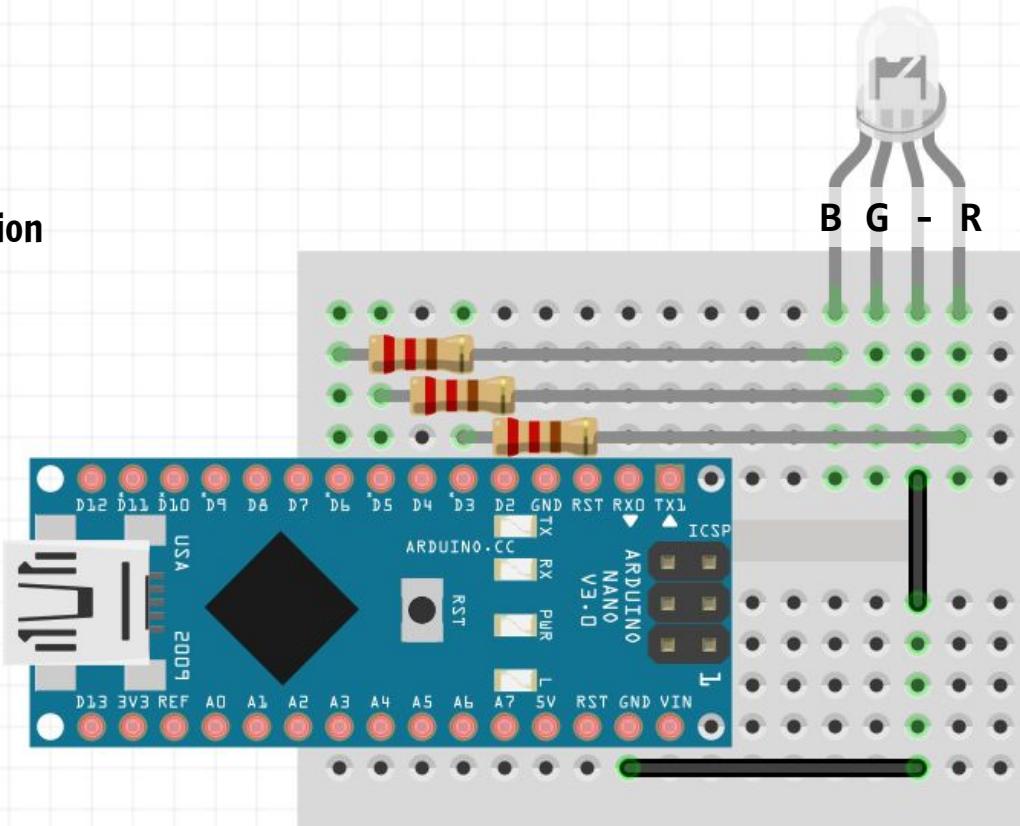
void setup() {
    // set up LED as OUTPUT
    pinMode(led_one, OUTPUT);
    pinMode(led_two, OUTPUT);
}

void loop() {
    digitalWrite(led_one, HIGH);
    digitalWrite(led_two, LOW);
    delay(500); // wait 0.5 second
    digitalWrite(led_one, LOW);
    digitalWrite(led_two, HIGH);
    delay(500); // wait 0.5 second
}
```

[04]  
[05]

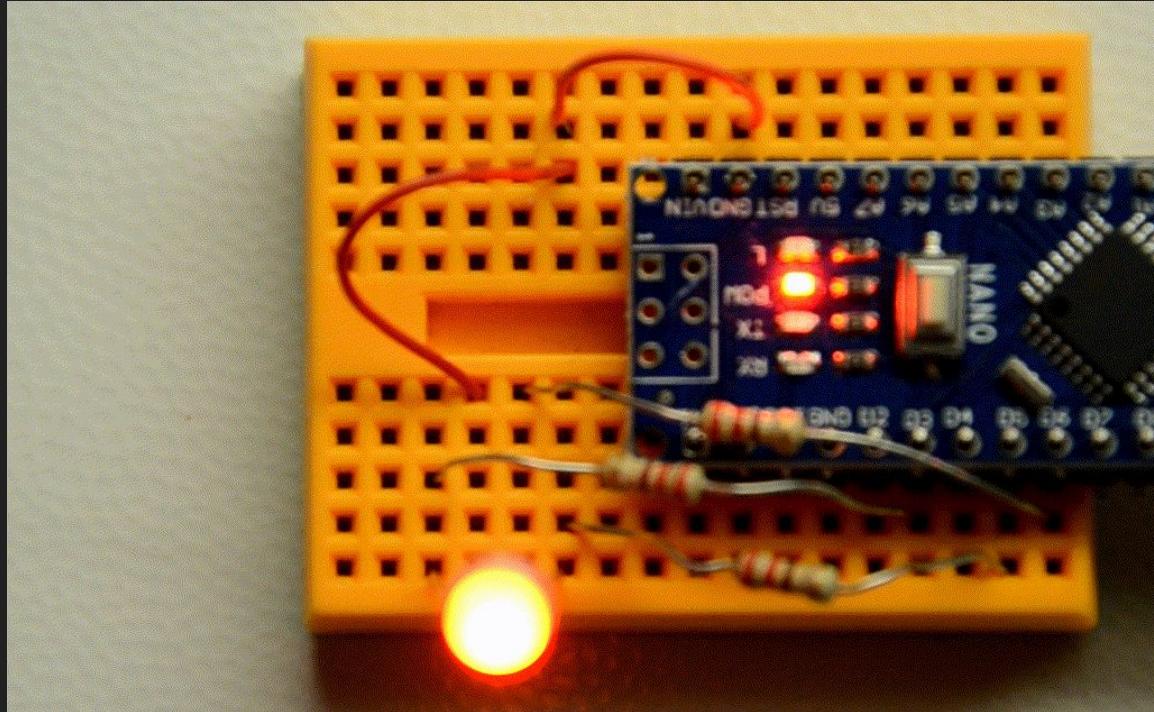
# RGB lučka

PWM pins \*Pulse Width Modulation



[ 04 ]

# RGB lučka menja barve



```
int redPin = 3;
int greenPin = 6;
int bluePin = 5;

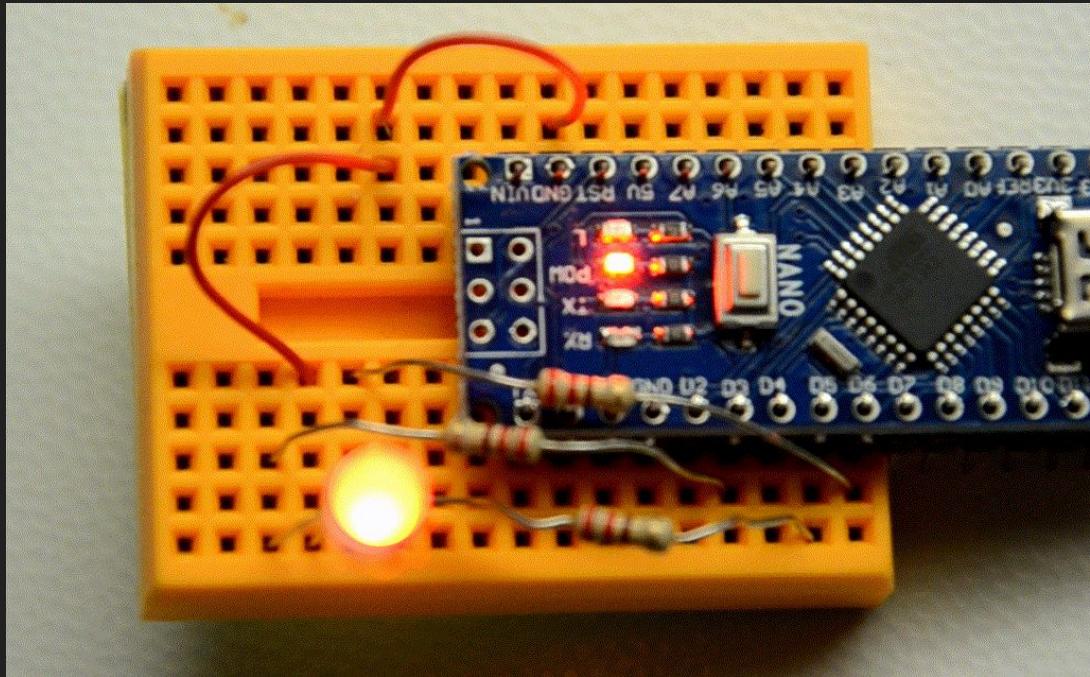
void setup() {
    pinMode(redPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
    pinMode(bluePin, OUTPUT);
}

void loop()
{
    setColor(255, 0, 0); // red
    delay(3000);
    setColor(0, 255, 0); // green
    delay(2000);
    setColor(0, 0, 255); // blue
    delay(1000);
}

void setColor(int red, int green, int blue)
{
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
```

[05]

# RGB lučka pulzajoče barve



```
int redPin = 3;
int greenPin = 5;
int bluePin = 6;
int i;

void setup() {
    // set up OUTPUTS
    pinMode(redPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
    pinMode(bluePin, OUTPUT);
}

void loop()
{
    for (i=0; i<=255; i++) {
        analogWrite(redPin, i);
        analogWrite(greenPin, 0);
        analogWrite(bluePin, 0);
        delay(5);
    }
}
```

# Senzorji in serial port



The image shows the Arduino IDE interface with a sketch named "06\_photo\_resistor". The code reads an analog value from pin A7 and prints it to the serial port:

```
int sensorPin = A7;  
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int sensorValue = analogRead(sensorPin);  
  Serial.println(sensorValue);  
}
```

A breadboard in the foreground has a red LED connected to digital pin 13. A photoresistor is connected between digital pin 2 and ground. A 10k pull-down resistor is also connected between pin 2 and ground.

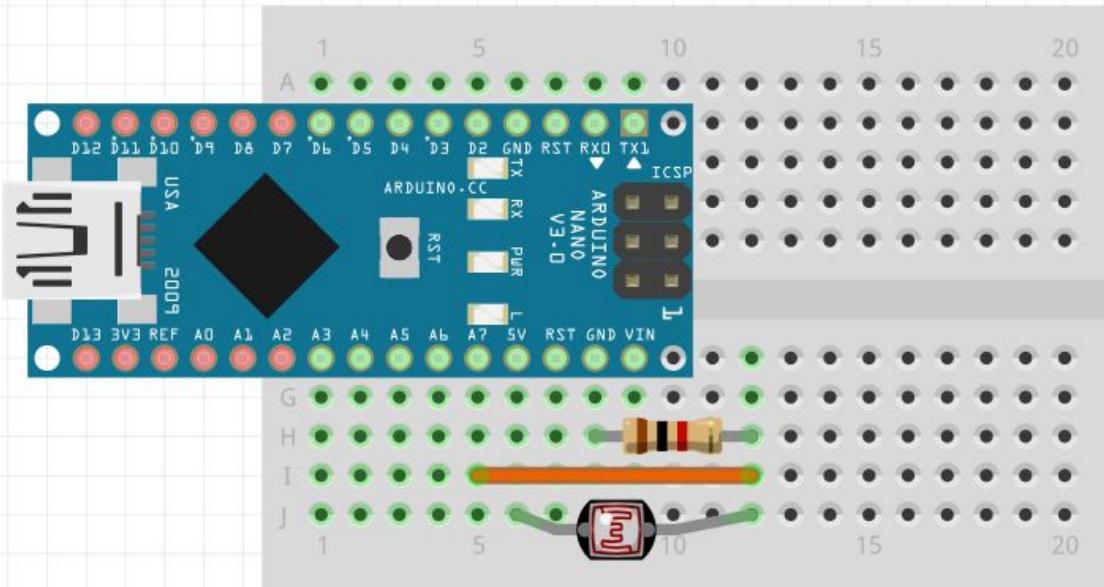


The serial monitor window shows a continuous stream of data, with the "Autoscroll" checkbox checked. The data consists of the value read from the photoresistor, which fluctuates between 882 and 902. The baud rate is set to 9600.

882  
883  
889  
899  
892  
894  
896  
898  
899  
899  
900  
901  
902  
902  
902

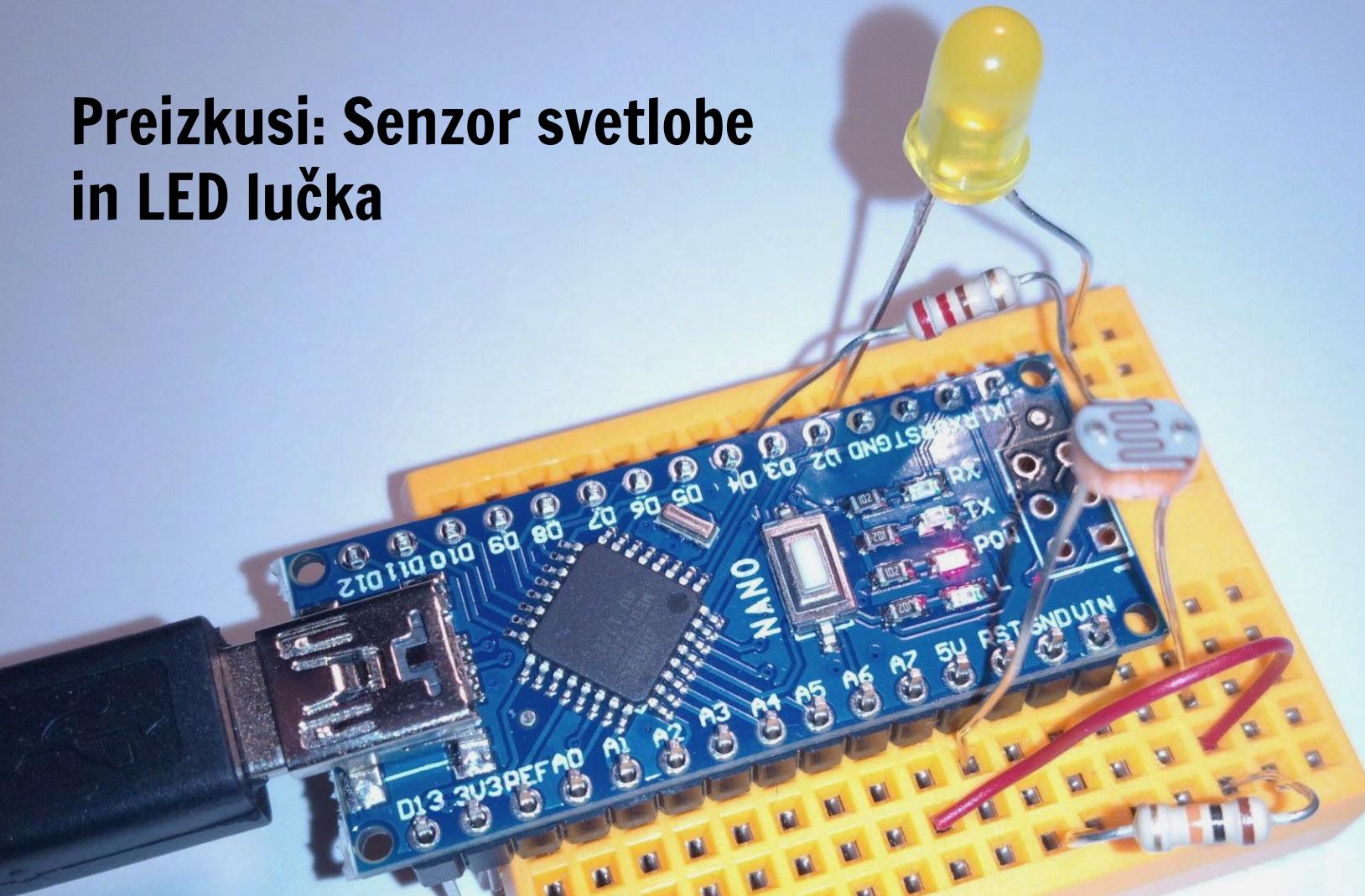
[ 06 ]

# Foto-upornik & Serial port

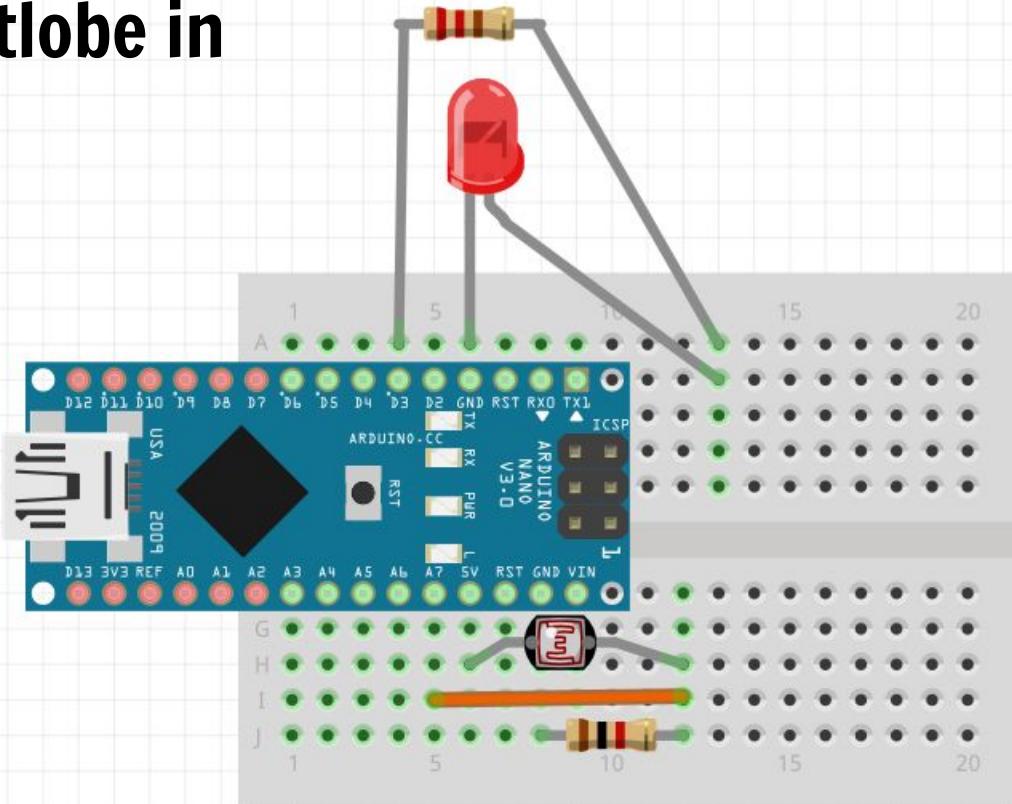


[07]

# Preizkusi: Senzor svetlobe in LED lučka



# Senzor svetlobe in LED lučka



```
int sensorPin = A7;  
int led = 3;  
int input;  
int output;  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    input = analogRead(sensorPin);  
    output = input / 4;  
    delay(1000);  
    analogWrite(led, output);  
    Serial.print( input );  
    Serial.print( " - " );  
    Serial.println( output );  
}
```

# Processing Demo

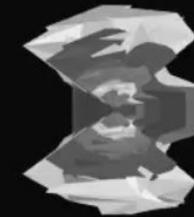
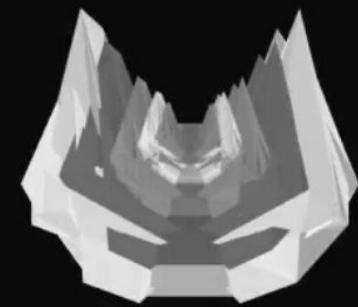
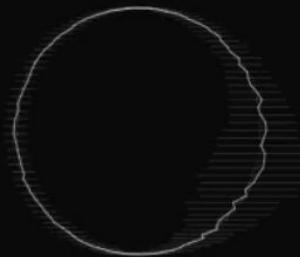
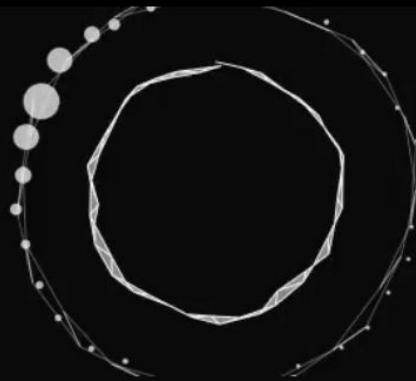
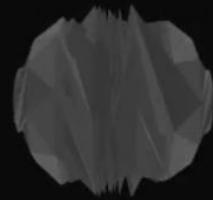
# Processing

[Cover](#)[Download](#)[Donate](#)[Exhibition](#)[Reference](#)[Libraries](#)[Tools](#)[Environment](#)[Tutorials](#)[Examples](#)[Books](#)

3.3.6 (4 September 2017)

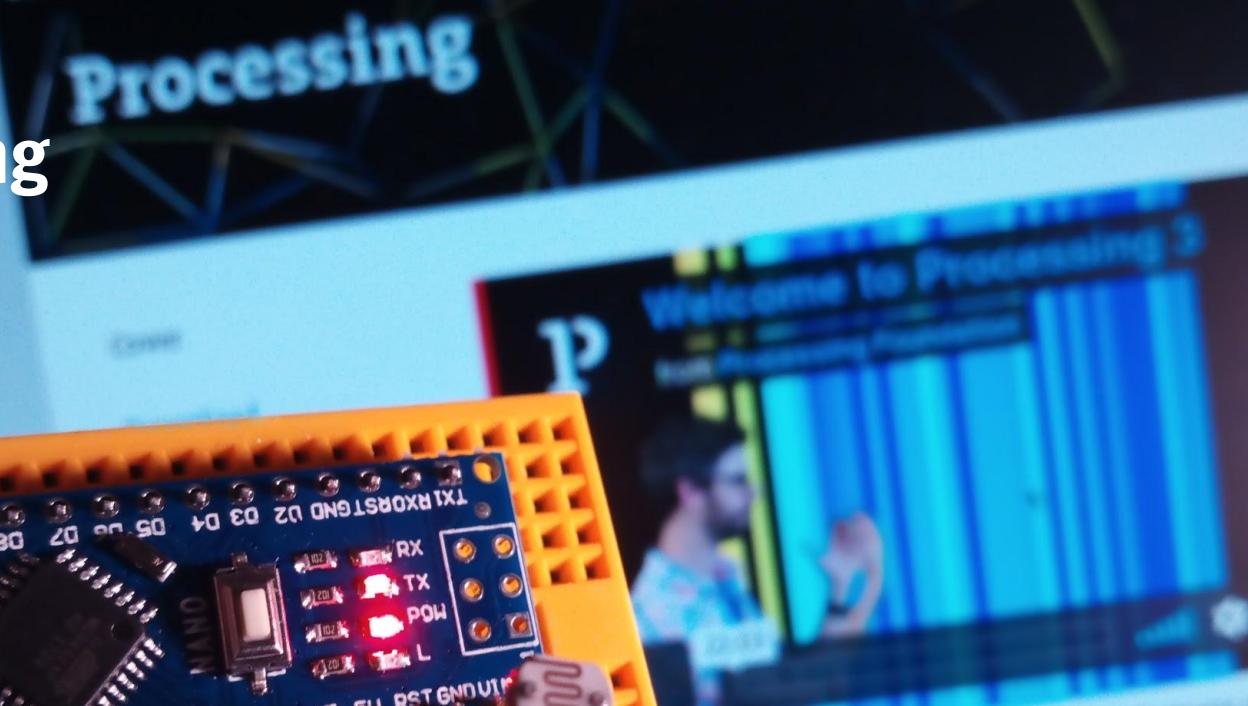
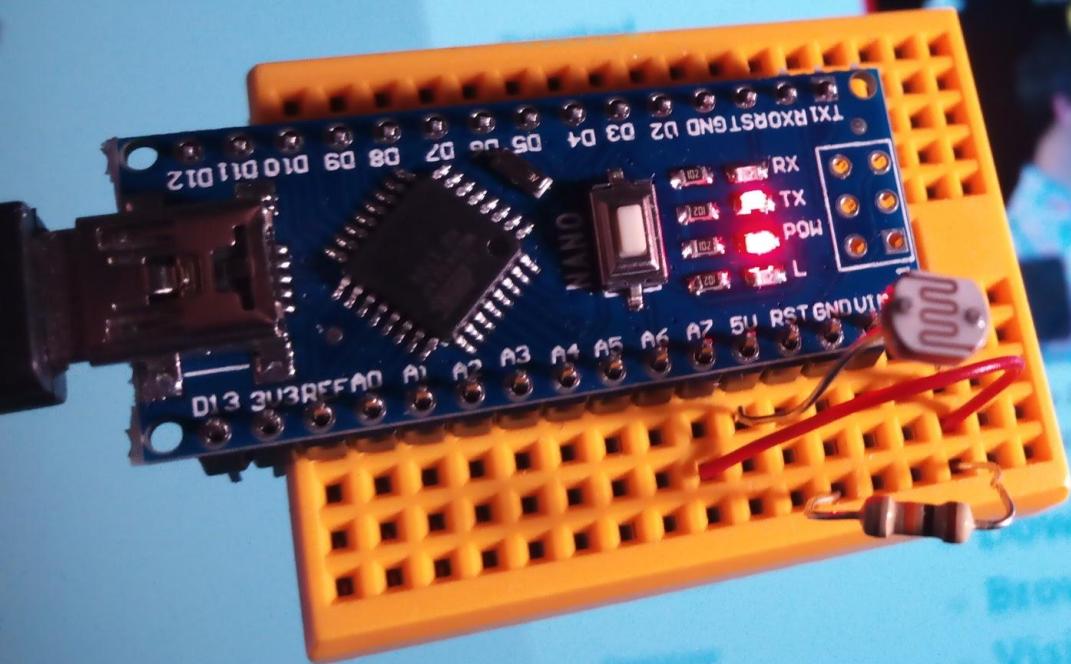
[Windows 64-bit](#)[Windows 32-bit](#)[Linux 64-bit](#)[Linux 32-bit](#)[Linux ARMv6hf](#)[Mac OS X](#)[» Github](#)Read about the [changes in 3.0](#). The [list of revisions](#) covers the differences

**<https://processing.org/download/>**



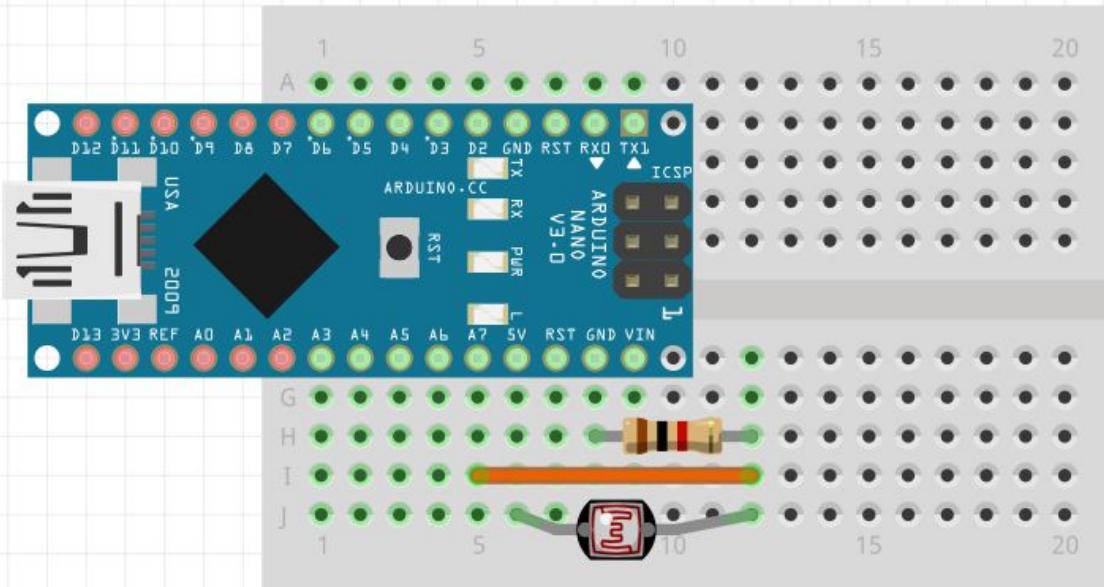
[08]

# Processing



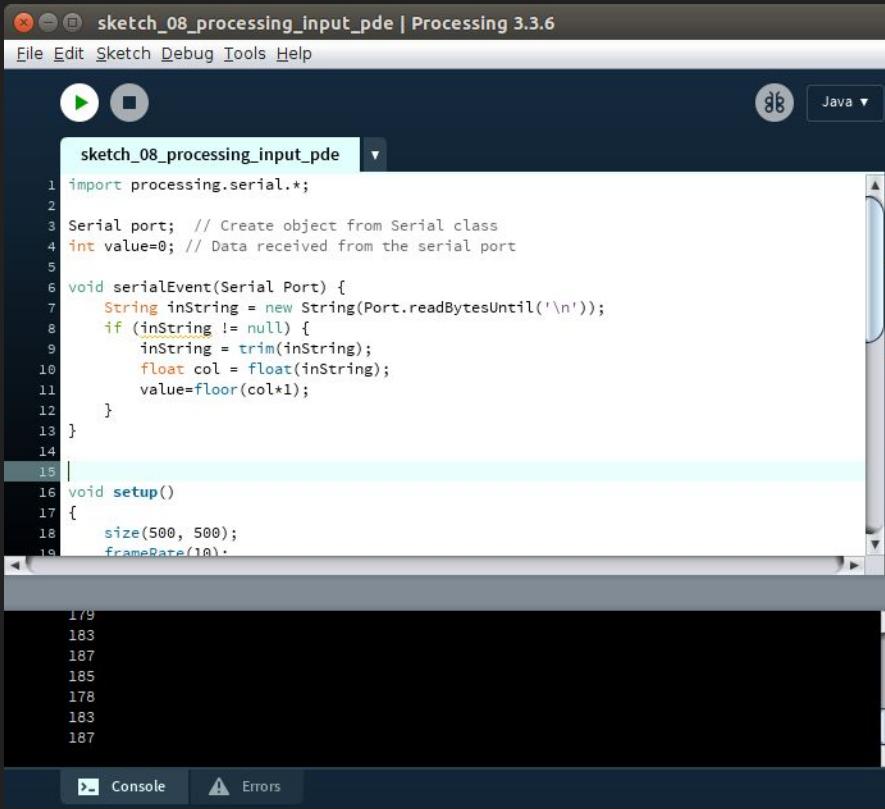
[08]

# Foto-upornik & Serial port



```
int sensorPin = A7;  
int input;  
int output;  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    input = analogRead(sensorPin);  
    output = input / 4;  
    delay(10);  
    Serial.println( output );  
}
```

# Processing sketch



The screenshot shows the Processing 3.3.6 IDE interface. The title bar reads "sketch\_08\_processing\_input\_pde | Processing 3.3.6". The menu bar includes File, Edit, Sketch, Debug, Tools, and Help. Below the menu is a toolbar with a play button, a square button, and a Java dropdown set to "Java". The main area displays the sketch code. The code imports the Serial class and defines a variable value. It contains a serialEvent function to handle incoming data from a serial port and a setup function to initialize the window size and frame rate.

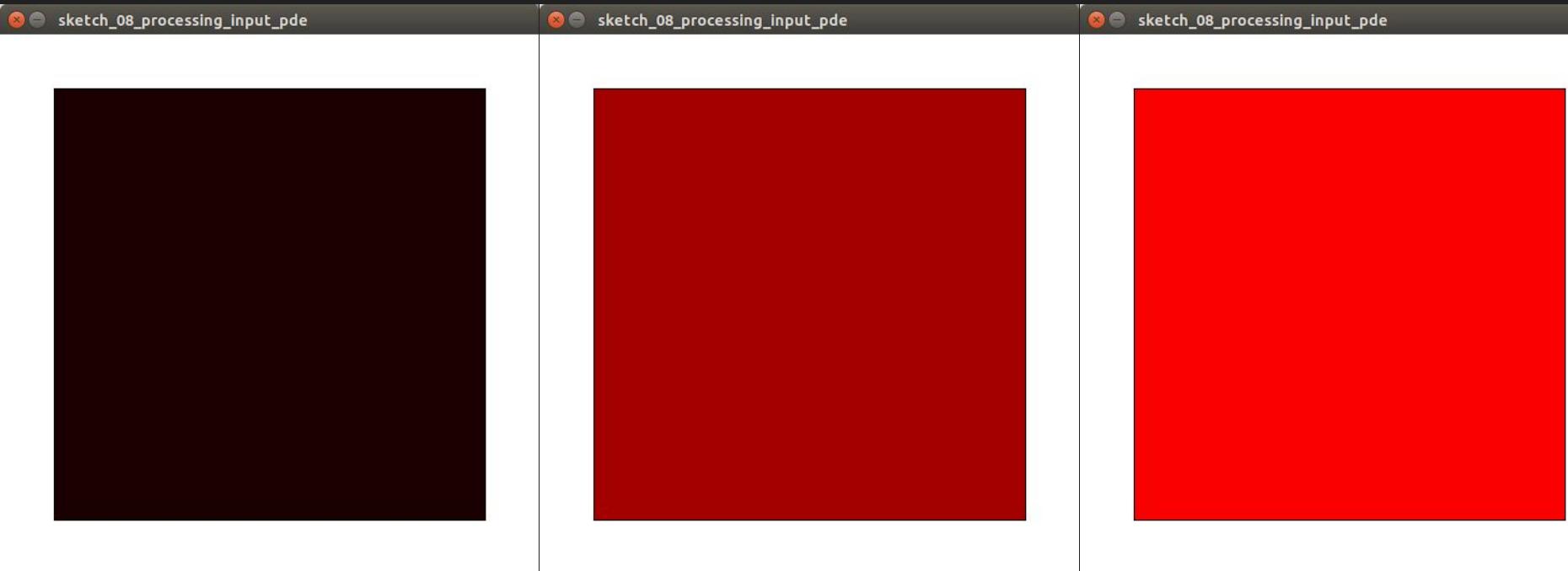
```
sketch_08_processing_input_pde
import processing.serial.*;
Serial port; // Create object from Serial class
int value=0; // Data received from the serial port
void serialEvent(Serial Port) {
    String inString = new String(Port.readBytesUntil('\n'));
    if (inString != null) {
        inString = trim(inString);
        float col = float(inString);
        value=floor(col*1);
    }
}
void setup()
{
    size(500, 500);
    frameRate(10);
}
```

The bottom console window shows the following output:

```
179
183
187
185
178
183
187
```

[ 08 ]

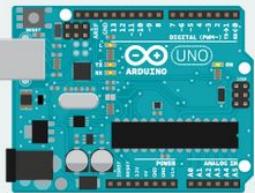
# Processing - Resultat



# Povezave



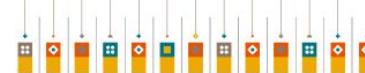
## WHAT IS ARDUINO?

[BUY AN ARDUINO](#)[LEARN ARDUINO](#)[DONATE](#)[BLOG](#)

THE IMPERIALIZER MAKES  
QUICK WORK OF METRIC  
CONVERSIONS



REDEFINING THE  
LEARNING EXPERIENCE  
ONE CLASSROOM  
AT A TIME

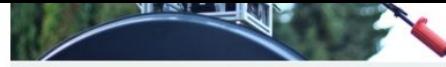
[BLOG](#)

ARDUINO MKR ZERO,  
THE POWER OF THE  
ZERO IN A SMALLER



# <https://www.arduino.cc/>

and access cool tutorials!





PROJECT HUB

MY DASHBOARD

NEW PROJECT

SEARCH PROJECTS



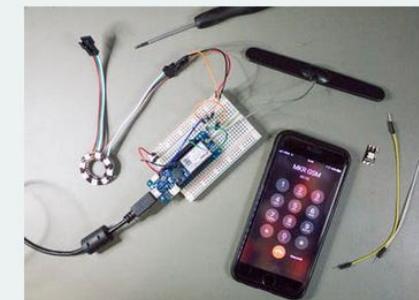
All products ▾

All categories ▾

Trending ▾

Any difficulty ▾

Any type ▾



<https://create.arduino.cc/projecthub>

77 VIEWS 0 COMMENTS 1 RESPECT

1,916 VIEWS 0 COMMENTS 4 RESPECTS

1,596 VIEWS 0 COMMENTS 7 RESPECTS

The screenshot shows the Arduino IDE's online editor interface. On the left, a sidebar lists categories like Sketchbook, Examples, Libraries, Monitor, Help, and Preferences. The Examples section is selected, showing examples categorized by type: 01.BASICS (6), 02.DIGITAL (9), 03.ANALOG (6), 04.COMMUNICATION (12), 05.CONTROL (6), and 06.SENSORS (4). The 'Blink' example under '01.BASICS' is currently selected. The main workspace displays the code for the 'Blink' sketch, which controls an LED on pin 13. The code includes comments explaining the setup and loop functions.

SEARCH EXAMPLES

SHOW EXAMPLES FOR ALL BOARDS

BUILT IN FROM LIBRARIES

01.BASICS (6)

- AnalogReadSerial
- BareMinimum
- Blink**
- DigitalReadSerial
- Fade
- ReadAnalogVoltage

02.DIGITAL (9)

03.ANALOG (6)

04.COMMUNICATION (12)

05.CONTROL (6)

06.SENSORS (4)

Blink

No Plugin Connection. Uploading is disabled until you reconnect.

HELP

```
1 /*
2 *  Blink
3 *
4 *  Turns an LED on for one second, then off for one second, repeatedly.
5 *
6 *  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7 *  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8 *  the correct LED pin independent of which board is used.
9 *  If you want to know what pin the on-board LED is connected to on your Arduino
10 *  model, check the Technical Specs of your board at:
11 *  https://www.arduino.cc/en/Main/Products
12 *
13 *  modified 8 May 2014
14 *  by Scott Fitzgerald
15 *  modified 2 Sep 2016
16 *  by Arturo Guadalupi
17 *  modified 8 Sep 2016
18 *  by Colby Newman
19 *
20 *  This example code is in the public domain.
21 *
22 *  http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
34   delay(1000);                      // wait for a second
35   digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36   delay(1000);                      // wait for a second
37 }
38 }
```

<https://create.arduino.cc/editor>



Simulator time: 00:00:15

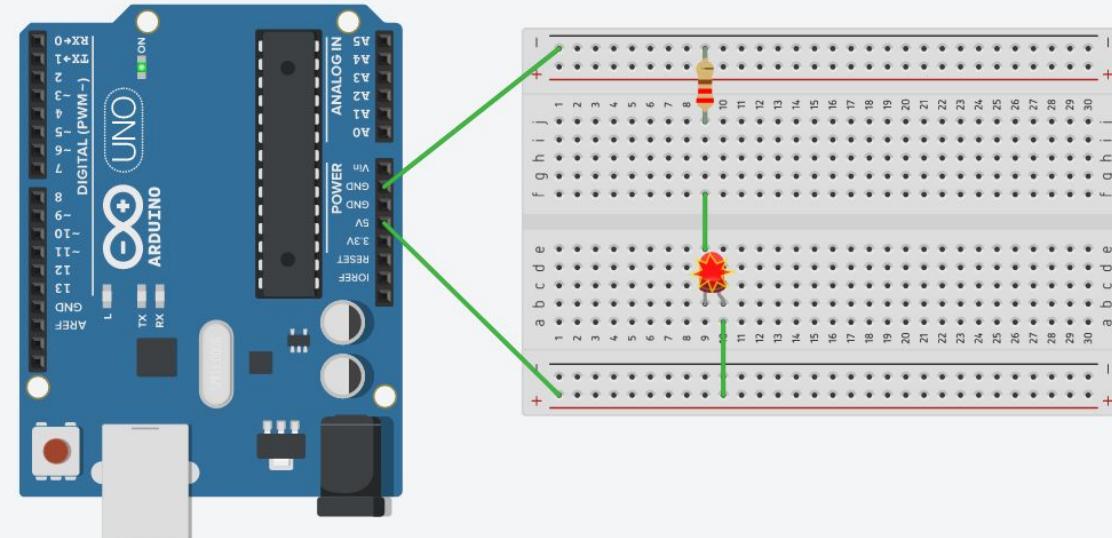
Code Editor

Components

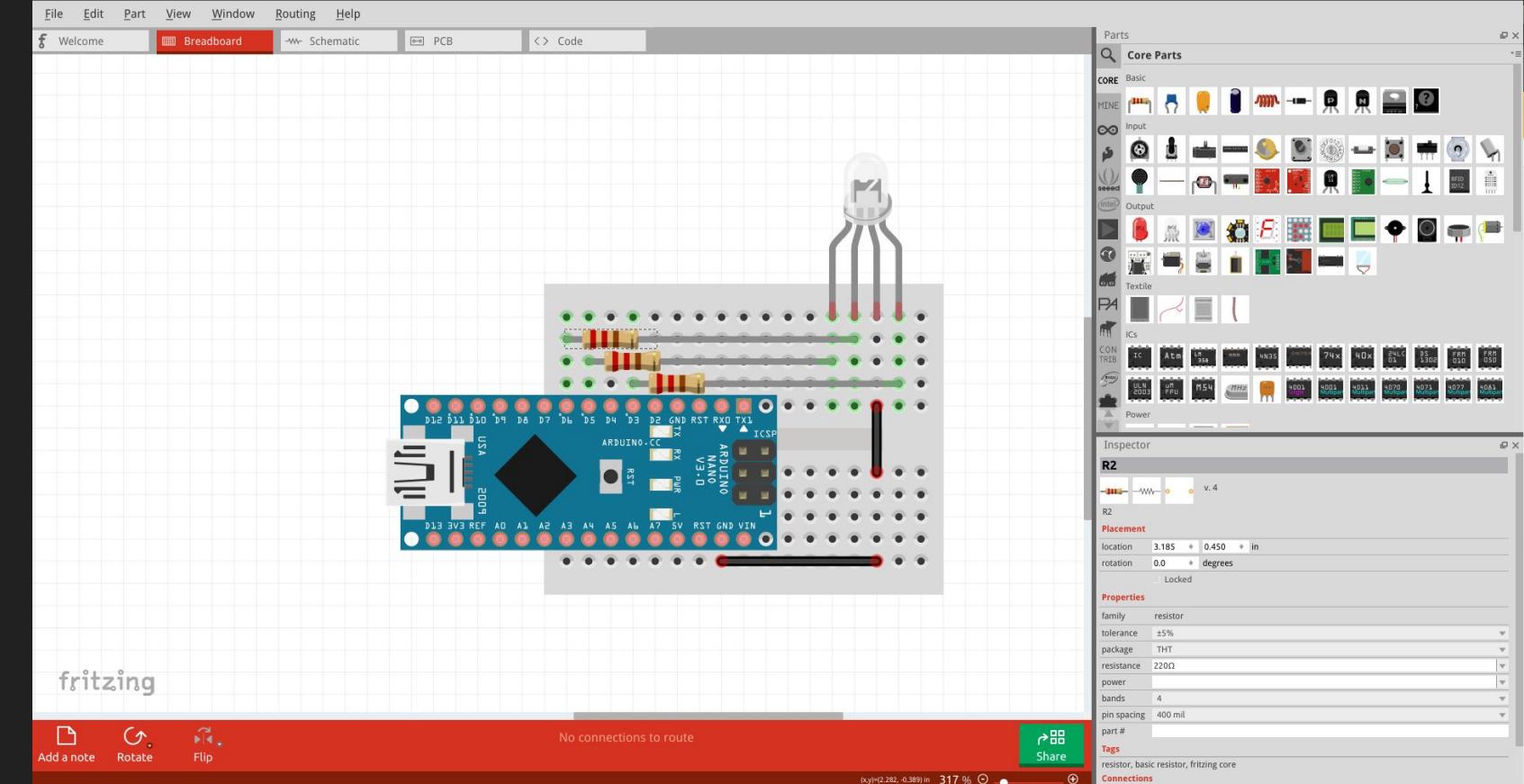
Stop Simulation

Export

Share



<https://www.tinkercad.com/>



<http://fritzing.org/download/>

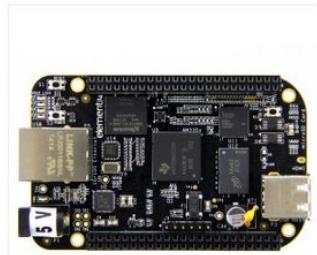
Išči po celotni trgovini...

[AKCIJA](#) [NOVO](#) [ARDUINO/GENUINO](#) [SBC](#) [SPARKFUN](#) [I/O](#) [MAKERBEAM](#) [TCT](#) [3D](#) [DRONI](#) [OSTALO](#)

SMAKSHOP : GENUINO/ARDUINO, LILYPAD, MAKERBEAM, 3D (TISKALNIKI, ABS, PLA), SPARKFUN



## AKCIJA



## NAROČI SE NA OBJAVE

\* potrebna polja  
email \*

Ime in priimek \*

NAROČI SE

Način dostave po pošti (Pošta Slovenije d.o.o.).

Strošek dostave je 3,55 EUR in brezplačna dostava pri naročilih nad 99,00 EUR.

Način dostave osebni prevzem na lokaciji podjetja.

Predhodno se je potrebno dogovoriti za termin prevzema.

Podpora kupcem(med 8:30-15:30, Pon - Pet)

elektronska pošta  
tel: 041 3344 86

# <https://smakshop.si>

149,99 € 129,99 €

299,99 € 249,99 €



All Categories ▾ Arduino



Cart(0)

Wish

ALL DEPARTMENTS ▾

New ▾ \$0.99

Top Sellers

Extreme Deals

Community ▾

MVP 24 hrs

Staff Picks

DX » "arduino"

## Category

### Hobbies & Toys(3)

- R/C Toys(3)
- Other Accessories(1)
- Repair Parts and Tools(1)
- R/C Tanks(1)

### Cell Phones&Accessories(1)

- Gadgets(1)
- NFC/Smart Control(1)

### Automobiles & Motorcycles(2)

- Gadgets & Auto Parts(2)
- Other Gadgets(2)

### Electrical & Tools(1191)

- Arduino & SCM Supplies(1155)
- Raspberry Pi(21)
- Other Accessories(104)
- Boards & Shields(294)
- Sensors(274)

## arduino(1198)



## Promotion Products



<http://www.dx.com/slarduino>

» Boards & Shields

» Boards & Shields

» Boards & Shields

» Displays

Computer & Office(1)



## LANGUAGE

FUNCTIONS

VARIABLES

STRUCTURE

## LIBRARIES

## GLOSSARY

The Arduino Reference text is licensed under a Creative Commons Attribution-Share Alike 3.0 License.

Find anything that can be improved? [Suggest corrections and new documentation via GitHub](#).

Doubts on how to use Github? Learn everything you need to know in this [tutorial](#).

# Language Reference

Arduino programming language can be divided in three main parts: structure, values (variables and constants), and functions.

## FUNCTIONS

For controlling the Arduino board and performing computations.

### Digital I/O

[`digitalRead\(\)`](#)[`digitalWrite\(\)`](#)[`pinMode\(\)`](#)

### Analog I/O

[`analogRead\(\)`](#)[`analogReference\(\)`](#)[`analogWrite\(\)`](#)

<https://www.arduino.cc/reference/en/>

Advanced I/O

[`noTone\(\)`](#)[`pulseIn\(\)`](#)

# Extra

# Namestitev gonilnikov za Windows - 2

- [http://www.wch.cn/download/CH341SER\\_ZIP.html](http://www.wch.cn/download/CH341SER_ZIP.html)

The screenshot shows the WCH website interface. In the top left, there's a logo with a blue arrow pointing upwards. The main navigation bar includes links for '关于我们' (About Us), '产品中心' (Product Center), 'BBS', '在线下载' (Online Download), '招贤纳士' (Recruitment), and '联系我们' (Contact Us). Below the navigation, there's a search bar with the placeholder '搜索 全部'. A large blue button labeled 'DOWNLOAD' with a downward arrow is highlighted with a red box.

**CH341SER.ZIP**

资料名称: CH341SER.ZIP  
资料类型: 驱动工具  
资料大小: 159KB  
资料版本: 3.4  
更新时间: 2018-09-27  
软件简介: CH341SER的USB串口WINDOWS驱动程序和DOLL驱动库，内含非标准波特率的设置等使用说明,支持32位Windows 10/8/7/VISTA/XP, SERVER 2016/2012/2008/2003, 2000/ME/98, 支持USB转串口的硬件串口。  
适用范围: CH340G, CH340C, CH340S, CH340T, CH340K, CH341A, CH341T, CH341H

相关资料:

- CH341SER.EXE
- CH340CH341SER的USB串口WINDOWS驱动程序的安装包,支...
- CH341SER\_LINUX.ZIP
- CH340CH341SER的USB串口Linux驱动程序,支持32位...
- CH341SER\_MAC\_ZIP
- CH340CH341SER的USB转串口 MacOS驱动程序的安装包,支持...
- CH341SER1.PDF
- CH341SER技术手册,USB总线转换芯片,接口方案,平台驱动齐全,附于...
- CH340PC1.PDF
- CH340的USB转串口的数据原理PCB,可用于OEM的RS232,...
- CH340PCB.ZIP
- CH340的USB转串口,USB转并行口,USB转串行的原理图和...
- CH340PCB.ZIP
- CH340的USB转串口,USB转并行口,USB转串行的原理图和...
- CH341SER\_ANDROID.ZIP
- CH340CH341SER的USB转串口安卓驱动程序,适用于andro...

