



Vhodno izhodne naprave

Laboratorijska vaja 3 - VP 2
TinkerCad-Arduino osnove

VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- Spoznavanje TinkerCad-a II.
- Breadboard
- TinkerCad + Arduino
- Domača naloga

VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- Spoznavanje TinkerCad-a II.

- Breadboard

- TinkerCad + Arduino

- Domača naloga

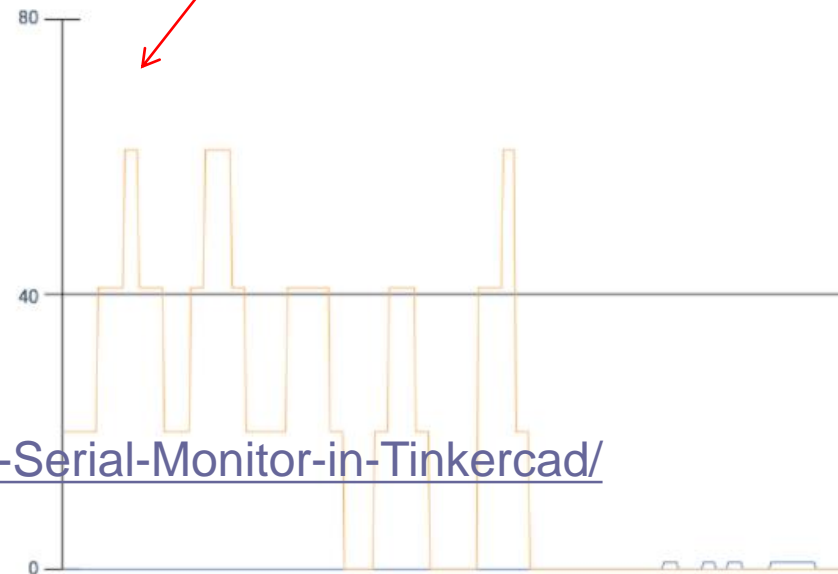
TinkerCad – Serijski izpis in vizualizacija

```
void setup()
{
  pinMode(2, INPUT);
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  // read the input pin
  buttonState = digitalRead(2);
  // read the input on analog pin 0:
  sensorValue = analogRead(A0);
  // print values to the serial monitor
  Serial.print(buttonState);
  Serial.print(", ");
  Serial.println(sensorValue);
  delay(10); // Delay a little bit t
}
```

lzpis

Vizualizacija

 Serial Monitor

<https://www.instructables.com/Arduino-Serial-Monitor-in-Tinkercad/>

TinkerCad – razhroščevanje (debugging)



The screenshot shows the TinkerCad IDE interface. At the top right, there are two buttons: 'Code' (blue) and 'Stop Simulation' (green). Below these, there are three icons: a download arrow, a file icon, and a bug icon. The main area is a code editor with a light gray background. The code is as follows:

```
1
2 int val;
3 int T;
4 void setup() {
5
6   pinMode(A0, INPUT);
7   Serial.begin(9600); // Begin the Serial at 9600 Baud
8 }
9
10 void loop() {
11   val = analogRead(A0);
12   //Serial.println(val);
13   T = map(val, 20, 358, 0, 165);
14   Serial.write(T);
15   delay(1000);
16   // Serial.println(val);
17 }
18
```

Line 14 is highlighted in blue, indicating a breakpoint. A red arrow points from a yellow box labeled 'Prekinitev (Breakpoint)' to line 14. Another red arrow points from a yellow box labeled 'Izpis vrednosti' to line 12. A small black box with the number '153' is visible next to line 14.

<https://www.instructables.com/Arduino-Serial-Monitor-in-Tinkercad/>

VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- Spoznavanje TinkerCad-a II.

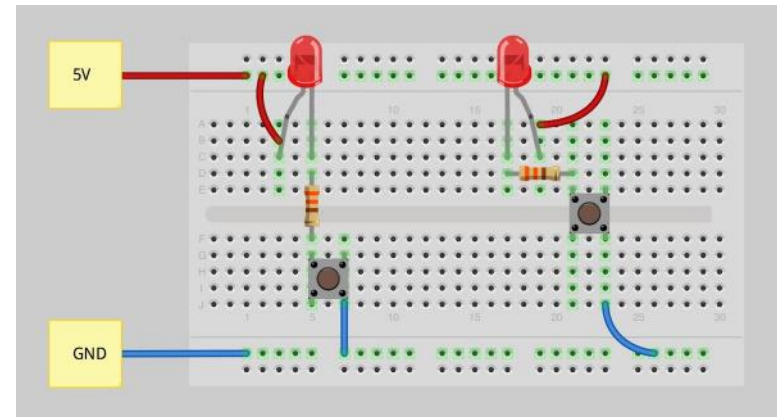
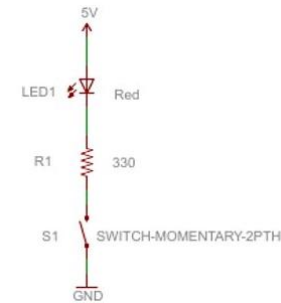
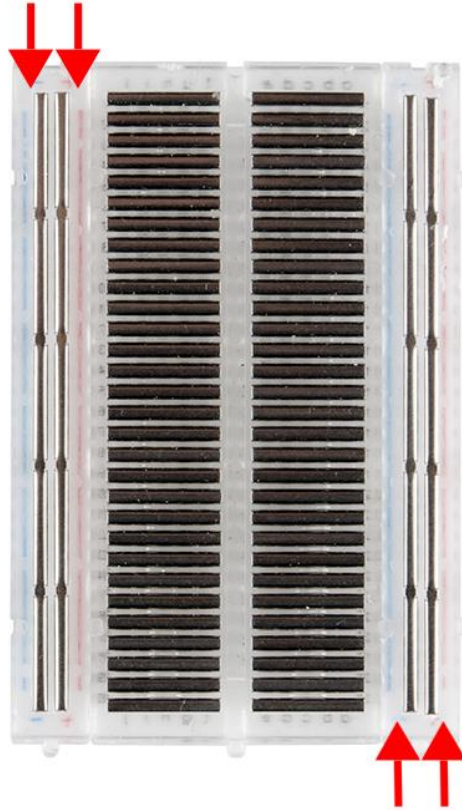
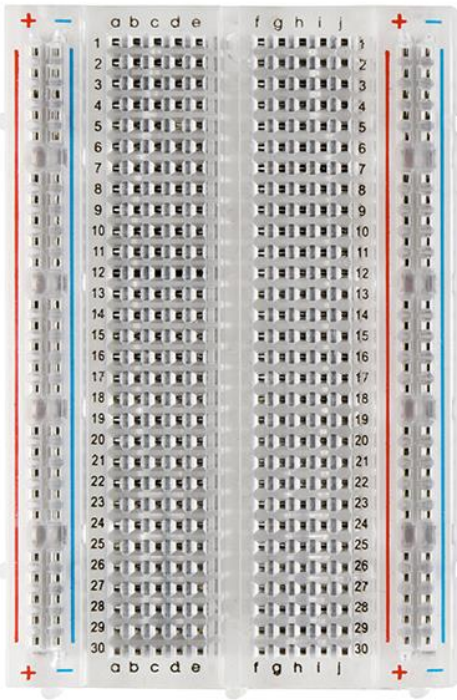
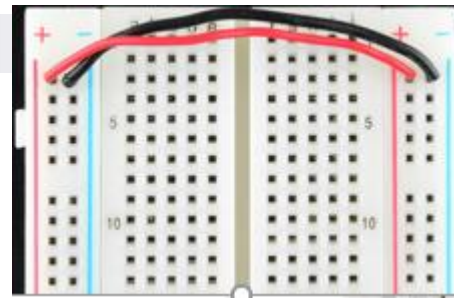
- Breadboard

- TinkerCad + Arduino

- Domača naloga

VIN projekt : TinkerCad

Breadboard vezave



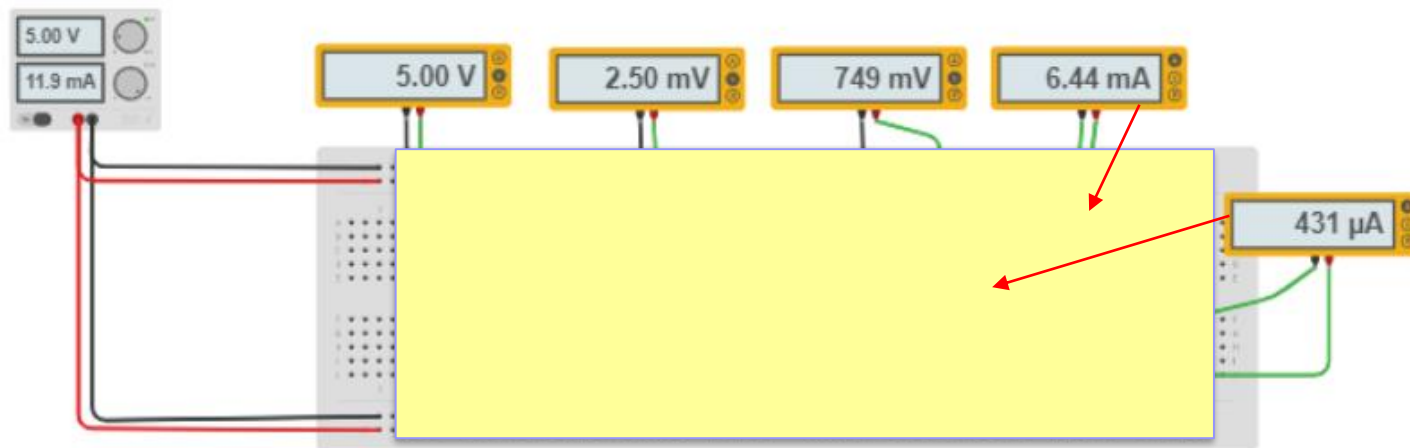
Viri

- <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/>
- <https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard>

VIN projekt : TinkerCad

Breadboard vezave – primeri vezav

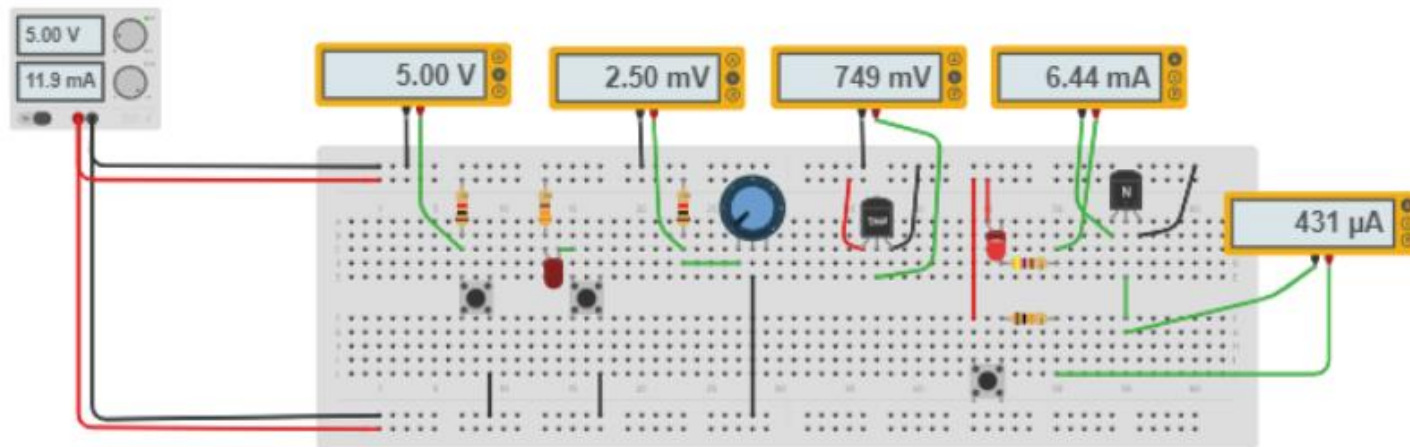
Breadboard vezave



VIN projekt : TinkerCad

Breadboard vezave – Primer rešitve

Breadboard vezave



VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- Spoznavanje TinkerCad-a II.

- Breadboard

- TinkerCad + Arduino

- Domača naloga

ARDUINO HW INTRO



<https://academy.programmingelectronics.com/arduino-hardware-basics/>

Arduino – dokumentacija

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

Language Reference

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

FUNCTIONS

For controlling the Arduino board and performing computations.

Digital I/O

`digitalRead()`
`digitalWrite()`
`pinMode()`

Analog I/O

`analogRead()`
`analogReference()`
`analogWrite()`

Math

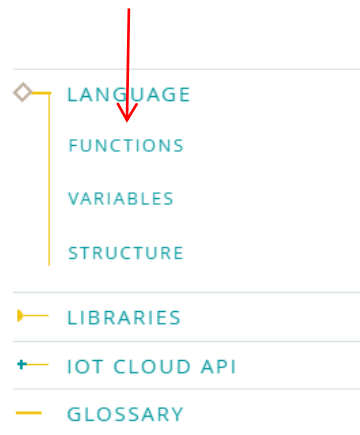
`abs()`
`constrain()`
`map()`
`max()`
`min()`
`pow()`
`sq()`
`sqrt()`

Random Numbers

`random()`
`randomSeed()`

Bits and Bytes

`bit()`
`bitClear()`
`bitRead()`
`bitSet()`

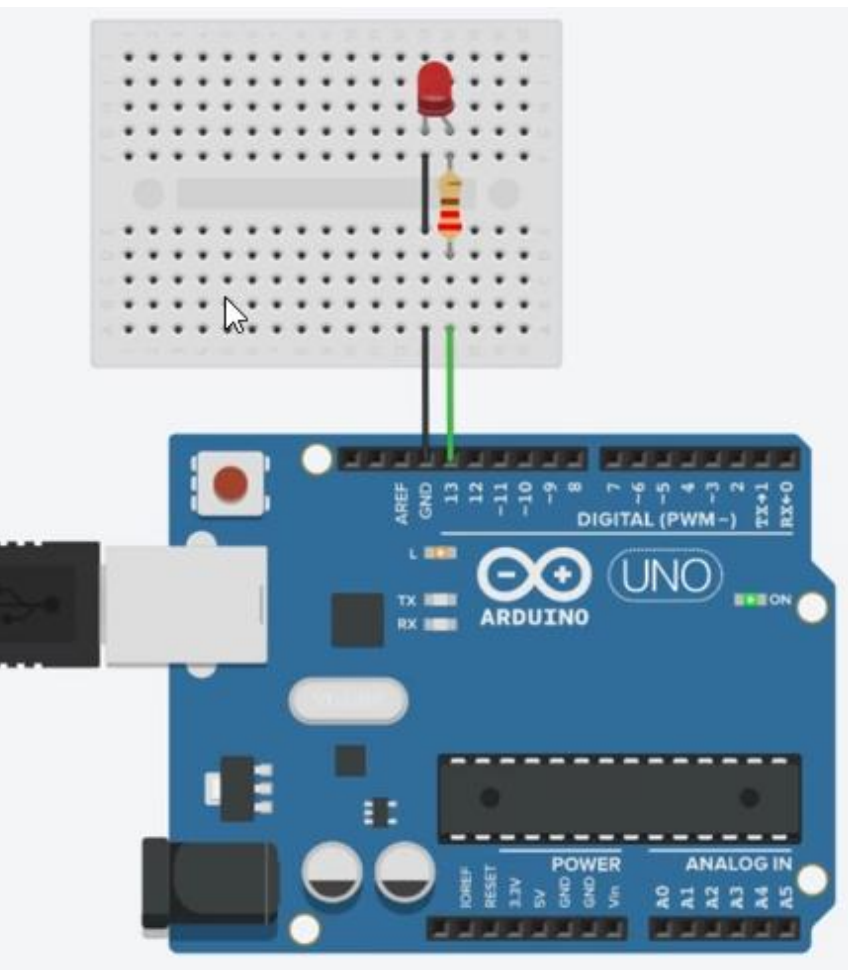


The Arduino Reference text is licensed under a [Creative Commons Attribution-Share Alike 3.0 License](#).

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

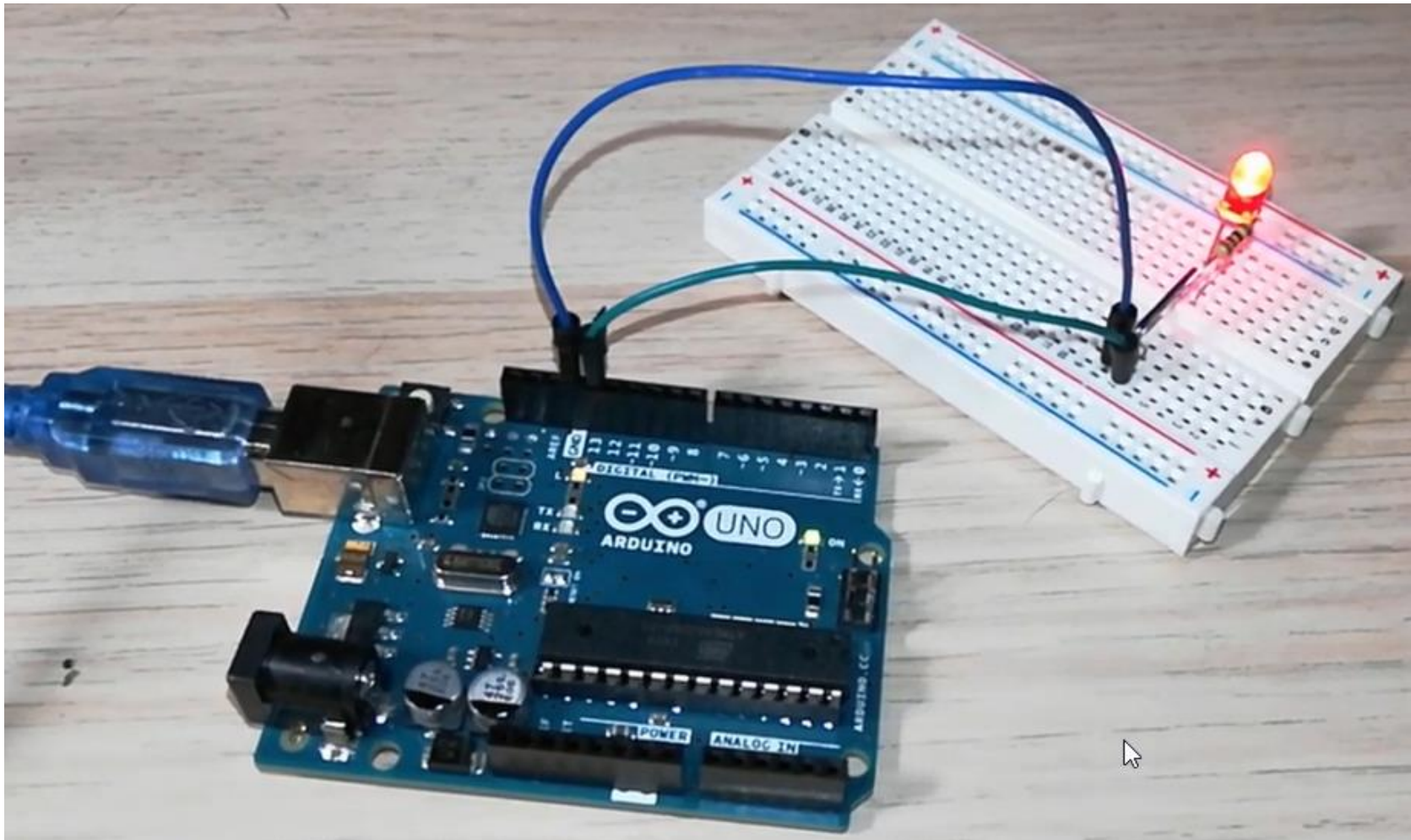
Doubts on how to use Github? Learn everything you need to know in [this tutorial](#).

Preprosto vezje z LED diodo in program

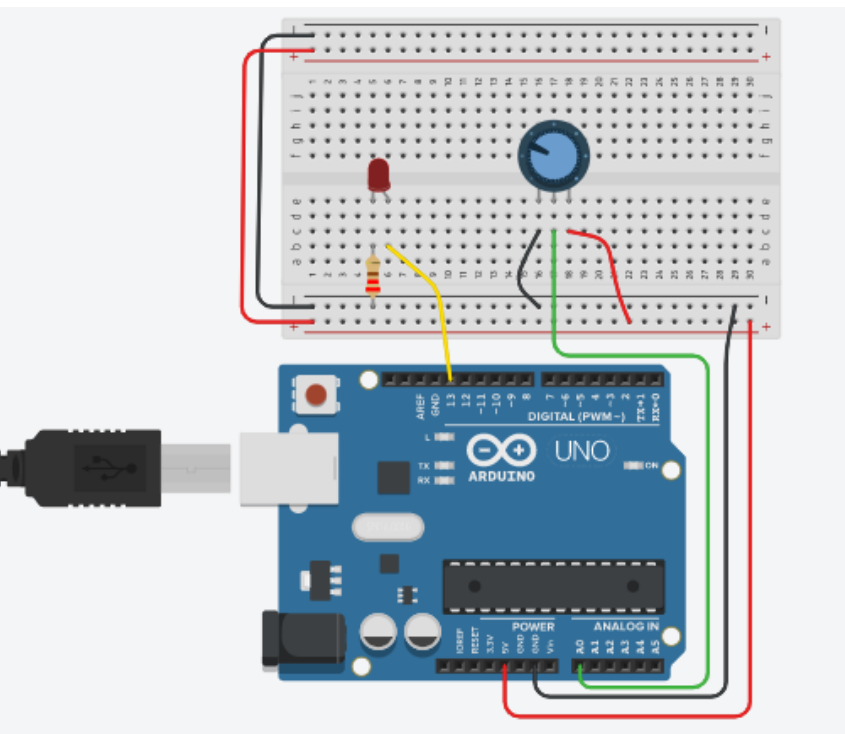


```
hello_world
1 /*
2  This program blinks pin 13 of the Arduino (the
3  built-in LED)
4  */
5
6 void setup()
7 {
8   pinMode(13, OUTPUT);
9 }
10
11 void loop()
12 {
13   // turn the LED on (HIGH is the voltage level)
14   digitalWrite(13, HIGH);
15   delay(1000); // Wait for 1000 millisecond(s)
16   // turn the LED off by making the voltage LOW
17   digitalWrite(13, LOW);
18   delay(1000); // Wait for 1000 millisecond(s)
19 }
```

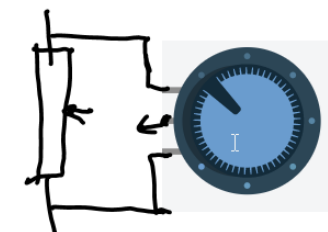
Preprosto vezje z LED diodo



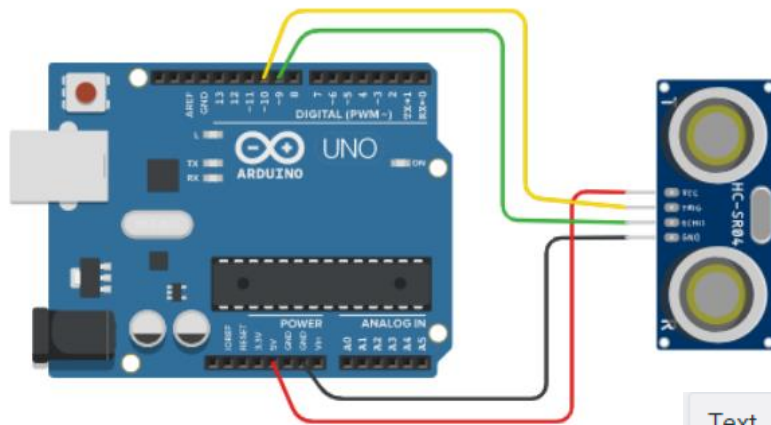
■ Vezje z uporovnim „tipalom“ (potenciometrom) - ADC



```
Text
1 int sensorValue = 0;
2
3 void setup()
4 {
5   pinMode(A0, INPUT);
6   pinMode(13, OUTPUT);
7 }
8
9 void loop()
10 {
11   // read the value from the sensor
12   sensorValue = analogRead(A0);
13   // turn the LED on
14   digitalWrite(13, HIGH);
15   // pause the program for <sensorValue> milliseconds
16   delay(sensorValue); // Wait for sensorValue millisecond(s)
17   // turn the LED off
18   digitalWrite(13, LOW);
19   // pause the program for <sensorValue> milliseconds
20   delay(sensorValue); // Wait for sensorValue millisecond(s)
21 }
```



■ Preprosto vezje z UZ tipalom – Časovnik (Timer)



```
Text
1 (Arduino Uno R3)
6 void setup() {
7   Serial.begin(9600); //Initialize Serial communication
8   pinMode(echo_Pin, INPUT); //Echo pin as Input
9   pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
10 }
11
12 void loop() {
13   digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
14   delay(1);
15   digitalWrite(trigger_Pin, HIGH);
16   delayMicroseconds(10); //Make Trigger pin High for 10 uS to st
17   digitalWrite(trigger_Pin, LOW);
18   duration = pulseIn(echo_Pin, HIGH); //Save the time it took ul
19   // distance = duration * 0.017; //((340*100)/10e6)/2
20   distance = duration / 58; //((340*100)/10e6)/2
21   /* Speed of the sound in Air = 340 m/s
22    * multiply it by 100 to get the data in cm
23    * divide by 1,000,000 as duration is measured in microseconds
24    * divide by 2 as ultrasound signal travels to object and comes b
25    */
26   Serial.print("Distance (cm) : ");
27   Serial.println(distance);|
```


Vezje z ultrazvočnim senzorjem SR04

```
Distance (cm) : 106
Distance (cm) : 103
Distance (cm) : 94
Distance (cm) : 88
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) :
```

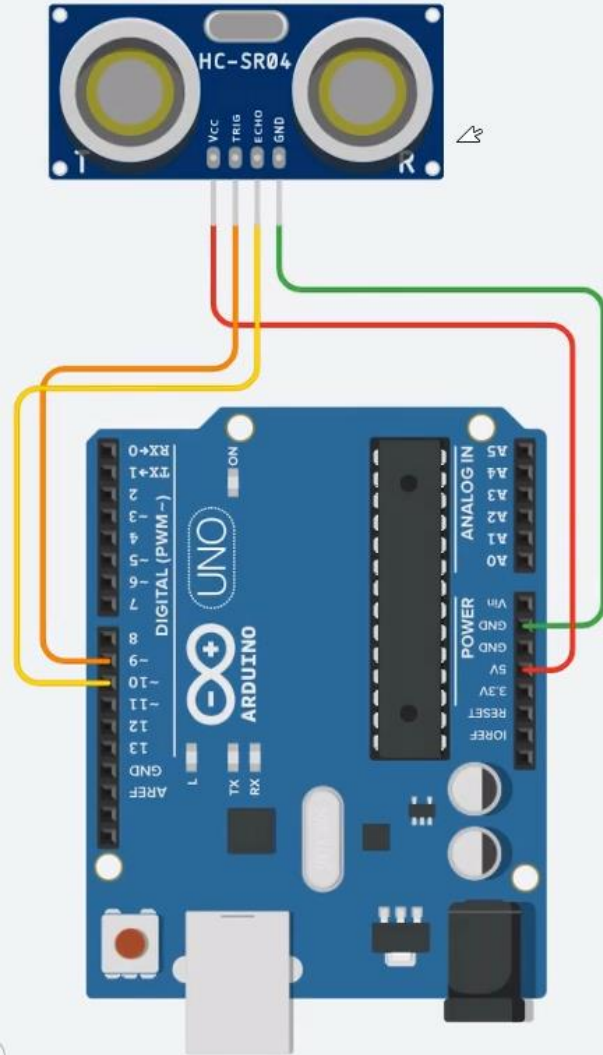


```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance
```

```
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}
```

```
void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);
```

```
  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //((340*100)/10e6)/2
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
   * divide by 1,000,000 as duration is measured in microseconds
   * divide by 2 as ultrasound signal travels to object and comes back
   */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```



Vezje z ultrazvočnim senzorjem SR04 - program

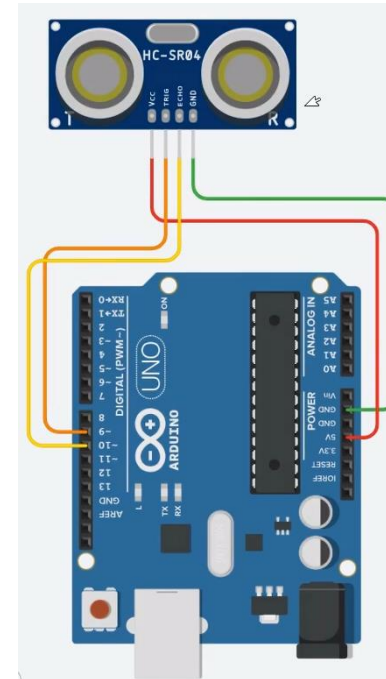
```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance
```

```
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}
```

```
void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);
```

```
  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //((340*100)/10e6)/2
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
   * divide by 1,000,000 as duration is measured in microseconds
   * divide by 2 as ultrasound signal travels to object and comes back
   */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```

```
Distance (cm) : 106
Distance (cm) : 103
Distance (cm) : 94
Distance (cm) : 88
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) :
```



Vezje z ultrazvočnim senzorjem SR04 - program

```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance
```

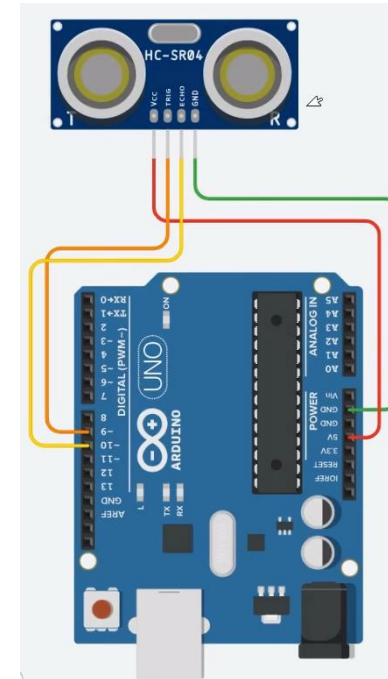
```
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}
```

```
void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);
```

```
  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //((340*100)/10e6)/2
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
   * divide by 1,000,000 as duration is measured in microseconds
   * divide by 2 as ultrasound signal travels to object and comes back
   */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```

Serial Monitor

```
Distance (cm) : 106
Distance (cm) : 103
Distance (cm) : 94
Distance (cm) : 88
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) : 84
```



Vezje z ultrazvočnim senzorjem SR04 - program

```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance

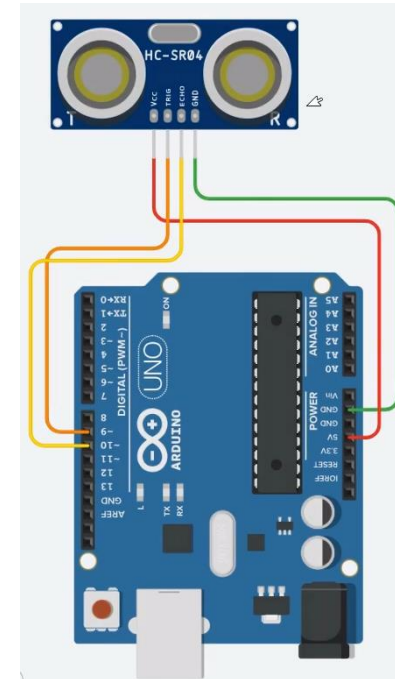
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}

void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);

  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //(((340*100)/10e6)/2
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
   * divide by 1,000,000 as duration is measured in microseconds
   * divide by 2 as ultrasound signal travels to object and comes back
   */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```

Serial Monitor

Distance (cm) :	106
Distance (cm) :	103
Distance (cm) :	94
Distance (cm) :	88
Distance (cm) :	84
Distance (cm) :	84
Distance (cm) :	84



Vezje z ultrazvočnim senzorjem SR04 - program

```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance

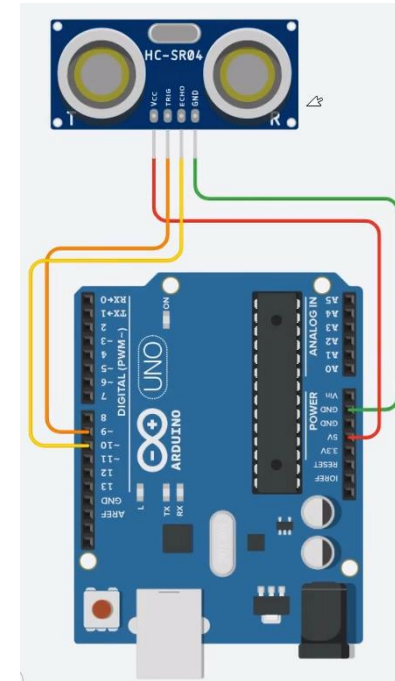
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}

void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);

  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //(((340*100)/10e6)/2 or duration/58
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
  * divide by 1,000,000 as duration is measured in microseconds
  * divide by 2 as ultrasound signal travels to object and comes back
  */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```

Serial Monitor

```
Distance (cm) : 106
Distance (cm) : 103
Distance (cm) : 94
Distance (cm) : 88
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) :
```



Distance (cm) = Measured Echo Time (in μ sec)/58
Distance (inch) = Measured Echo Time (in μ sec)/148

Vezje z ultrazvočnim senzorjem SR04 - program

```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance

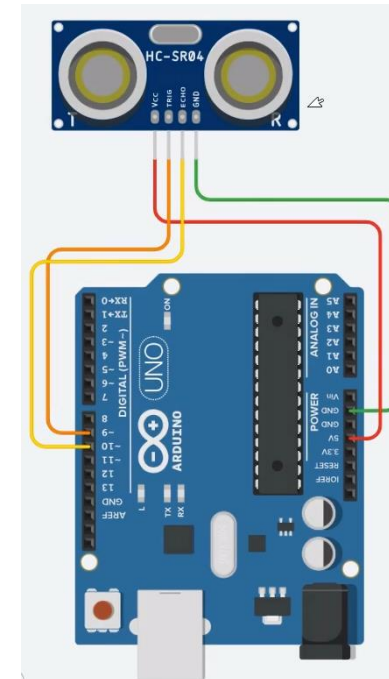
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}

void loop() {
  digitalWrite(trigger_Pin, LOW); //Make Trigger pin Low at start
  delay(1);
  digitalWrite(trigger_Pin, HIGH);
  delayMicroseconds(10); //Make Trigger pin High for 10 uS to start sending the pulse
  digitalWrite(trigger_Pin, LOW);

  duration = pulseIn(echo_Pin, HIGH); //Save the time it took ultrasonic wave to come back
  distance = duration * 0.017; //((340*100)/10e6)/2
  /* Speed of the sound in Air = 340 m/s, multiply it by 100 to get the data in cm
   * divide by 1,000,000 as duration is measured in microseconds
   * divide by 2 as ultrasound signal travels to object and comes back
   */
  Serial.print("Distance (cm) : ");
  Serial.println(distance);
  delay(100);
}
```

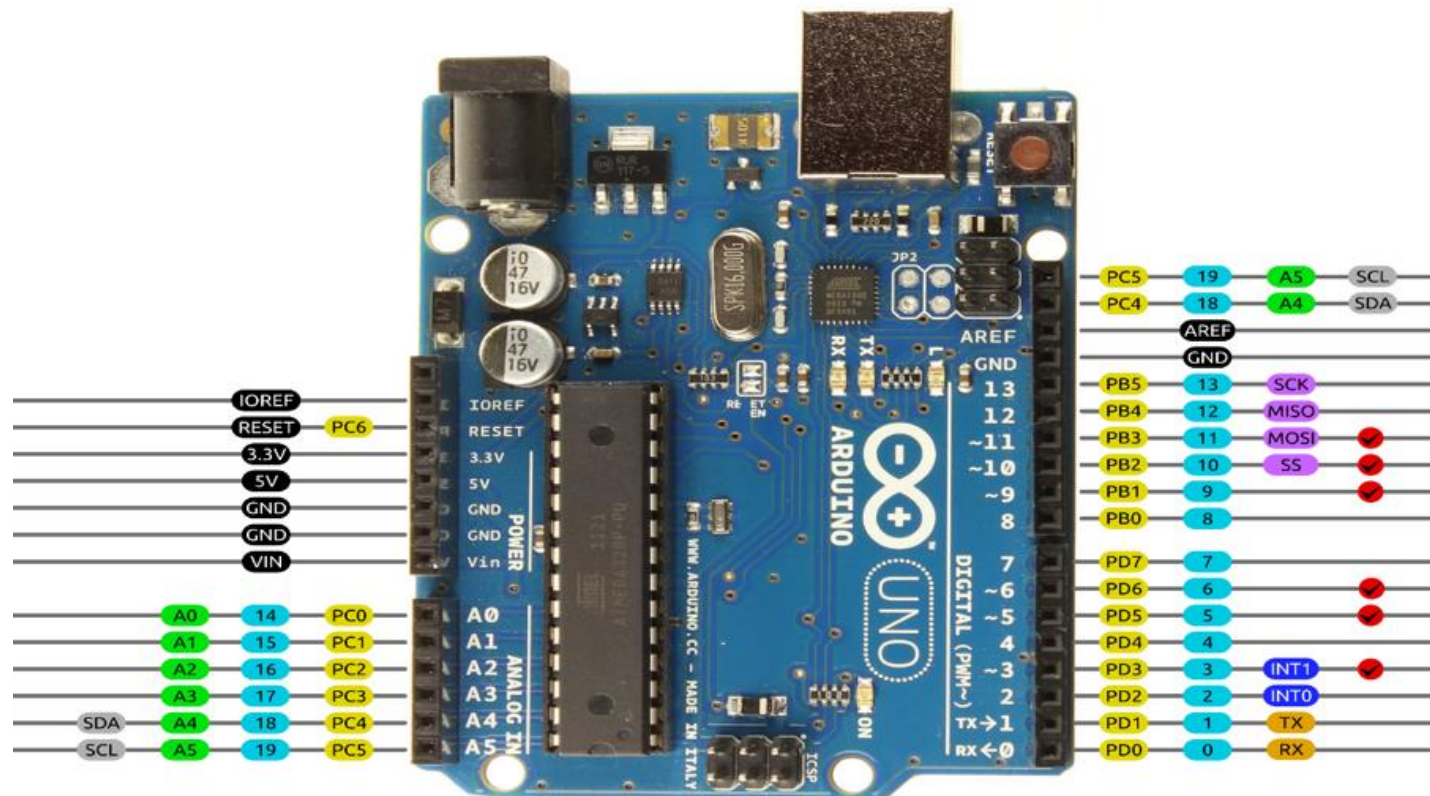
Serial Monitor

```
Distance (cm) : 106
Distance (cm) : 103
Distance (cm) : 94
Distance (cm) : 88
Distance (cm) : 84
Distance (cm) : 84
Distance (cm) : 84
```



■ Arduino :

5V !!!



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

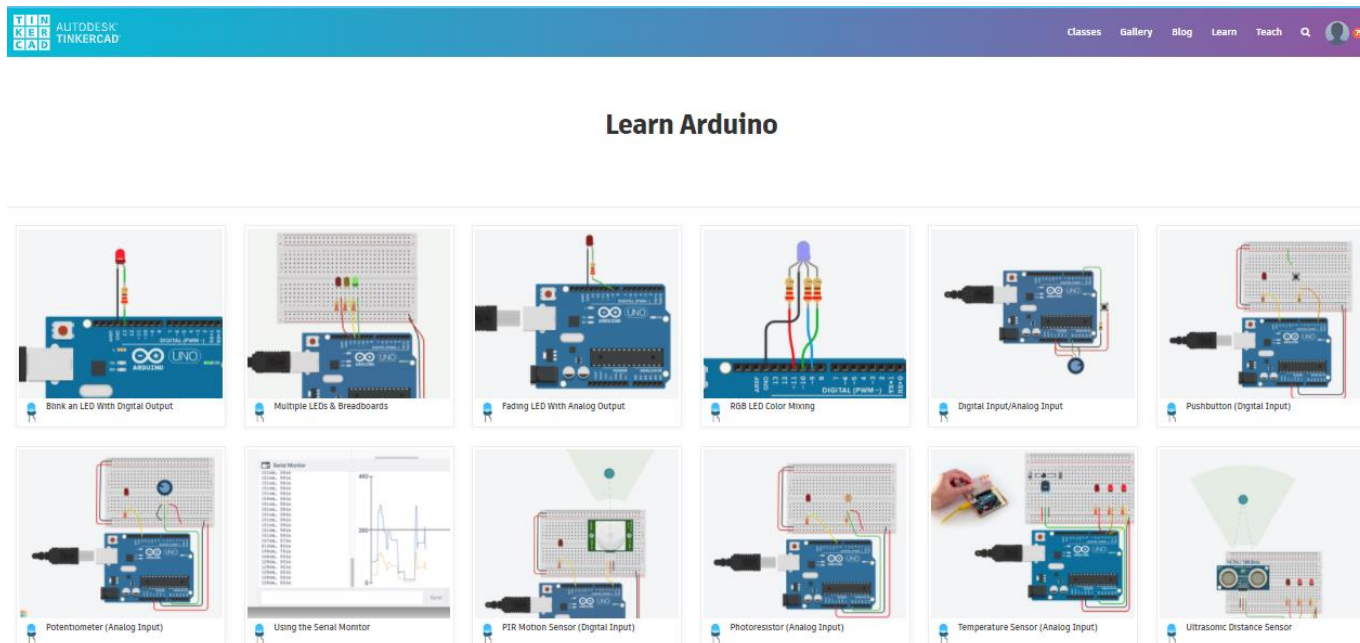
■ Viri – Arduino :

□ GitHub :

- https://github.com/LAPSyLAB/Arduino_projects

□ TinkerCad – Learn – Arduino

- <https://www.tinkercad.com/learn/circuits/projects>



Dodatni viri (po potrebi) :

■ GET STARTED PROGRAMMING ARDUINO TODAY!

- Learn the 2 most important Arduino programming functions
- Get familiar with Arduino coding
- Understand your Arduino hardware
- Learn the Arduino software setup
- 12 engaging video lessons
- Z naslova <<https://www.programmingelectronics.com/arduino-crash-course/>>

■ Coursera: Interfacing with the Arduino

- University of California, Irvine
- Z naslova <<https://www.coursera.org/lecture/interface-with-arduino/module-1-introduction-video-VgZmt>>

Poglobljena gradiva

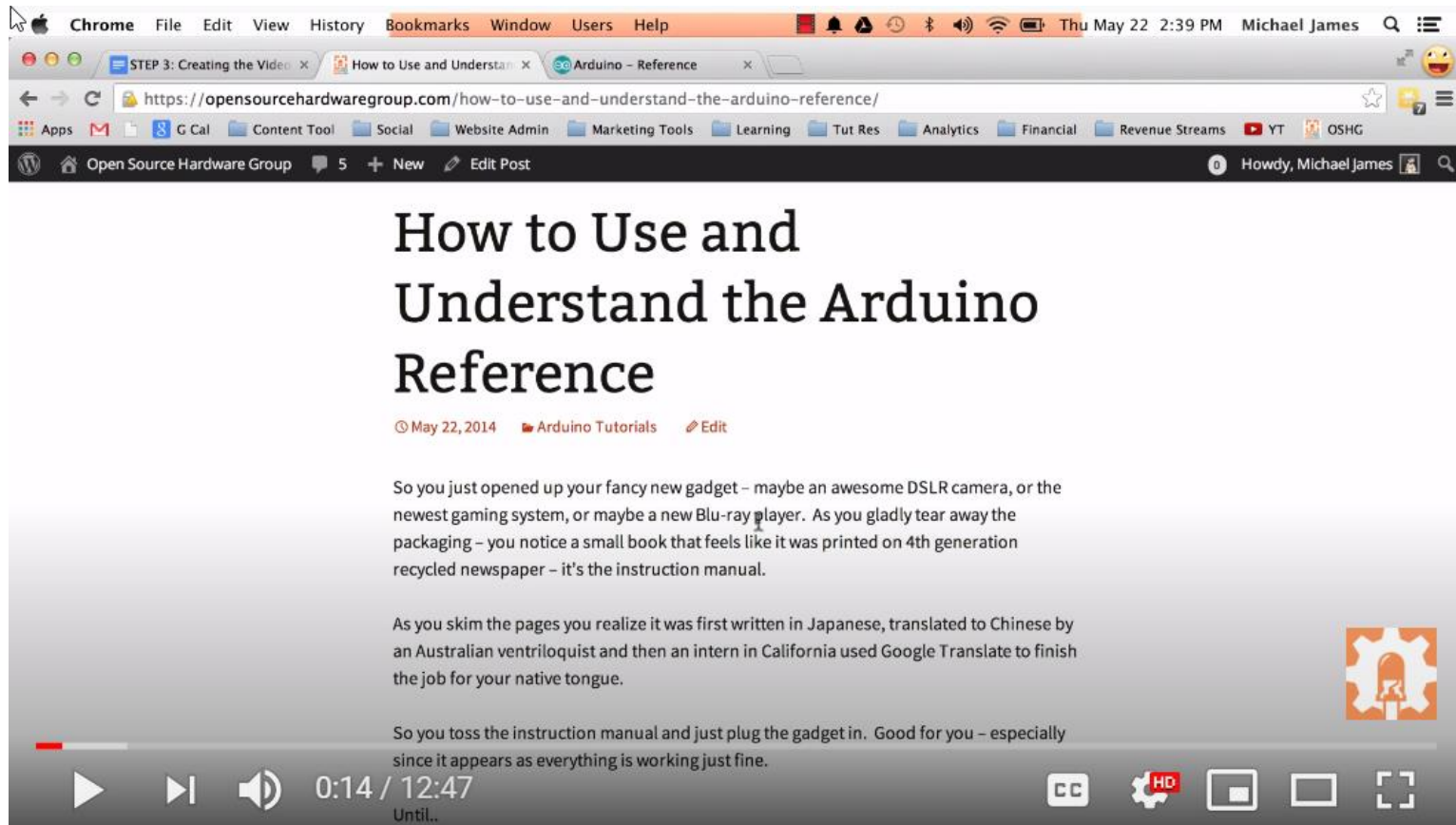
■ How to Use and Understand the Arduino Reference :: Open Source Hardware Group Arduino Tutorials

- Z naslova <https://www.youtube.com/watch?time_continue=739&v=f3h4pV_B2Dg&feature=emb_logo>

■ LED Blinking code with and without using Arduino library

- Z naslova <<http://electronicswithyou.com/arduino/led-blinking-code-with-and-without-using-arduino-library/>>

How to Use and Understand the Arduino Reference (neobvezno)



The screenshot shows a video player interface. The video content is a webpage from 'opensourcehardwaregroup.com' with the title 'How to Use and Understand the Arduino Reference'. The page is dated May 22, 2014, and is categorized under 'Arduino Tutorials'. The text on the page describes the process of understanding a new gadget, using the Arduino reference as an example. It mentions that the reference was first written in Japanese, translated to Chinese by an Australian ventriloquist, and then translated to English by an intern in California using Google Translate. The video player controls at the bottom show a progress bar at 0:14 / 12:47, with a play button, a volume icon, and a full screen button.

Chrome File Edit View History Bookmarks Window Users Help Thu May 22 2:39 PM Michael James

STEP 3: Creating the Video x How to Use and Understand the Arduino Reference x

https://opensourcehardwaregroup.com/how-to-use-and-understand-the-arduino-reference/

Apps G Cal Content Tool Social Website Admin Marketing Tools Learning Tut Res Analytics Financial Revenue Streams YT OSHG

Open Source Hardware Group 5 + New Edit Post Howdy, Michael James

How to Use and Understand the Arduino Reference

May 22, 2014 Arduino Tutorials Edit

So you just opened up your fancy new gadget – maybe an awesome DSLR camera, or the newest gaming system, or maybe a new Blu-ray player. As you gladly tear away the packaging – you notice a small book that feels like it was printed on 4th generation recycled newspaper – it's the instruction manual.

As you skim the pages you realize it was first written in Japanese, translated to Chinese by an Australian ventriloquist and then an intern in California used Google Translate to finish the job for your native tongue.

So you toss the instruction manual and just plug the gadget in. Good for you – especially since it appears as everything is working just fine.

0:14 / 12:47 Until..

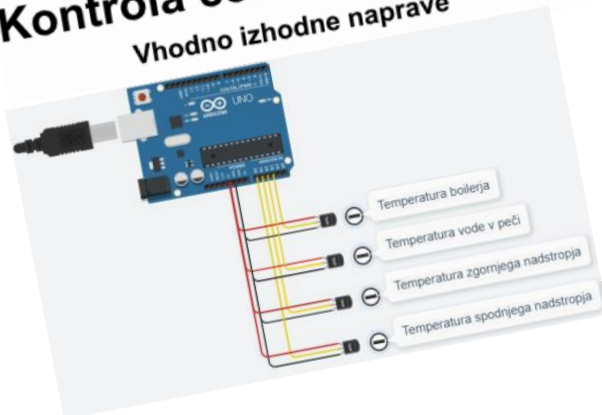
VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- Spoznavanje TinkerCad-a II.
- Breadboard
- TinkerCad + Arduino
- Domača naloga

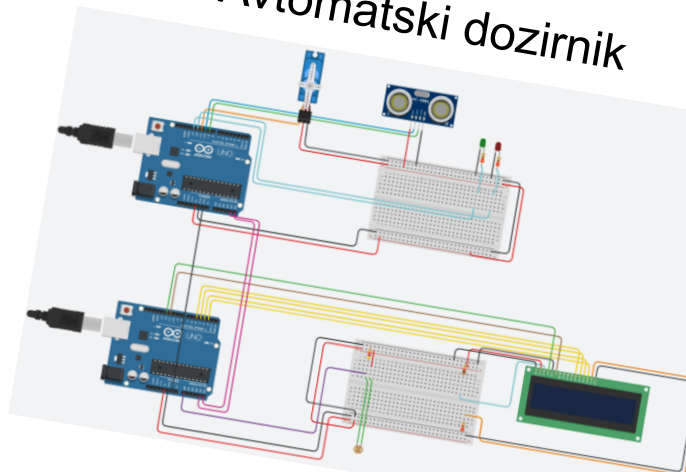
VIN Projekt Arduino – Primeri 19/20

TinkerCad

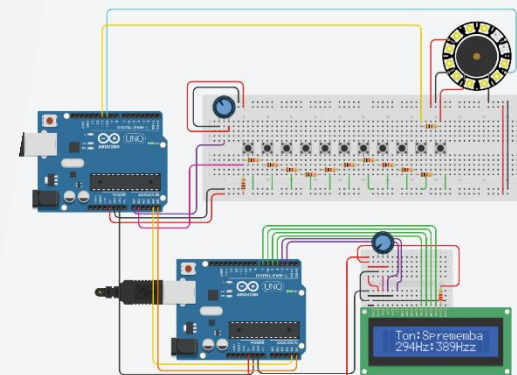
Arduino projekt,
Kontrola centralne kurjave
Vhodno izhodne naprave



Avtomatski dozirnik



Mini Piano



Sledenje

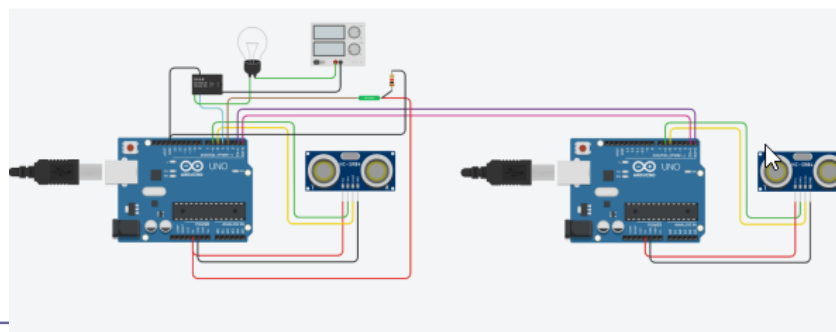


HW izvedba

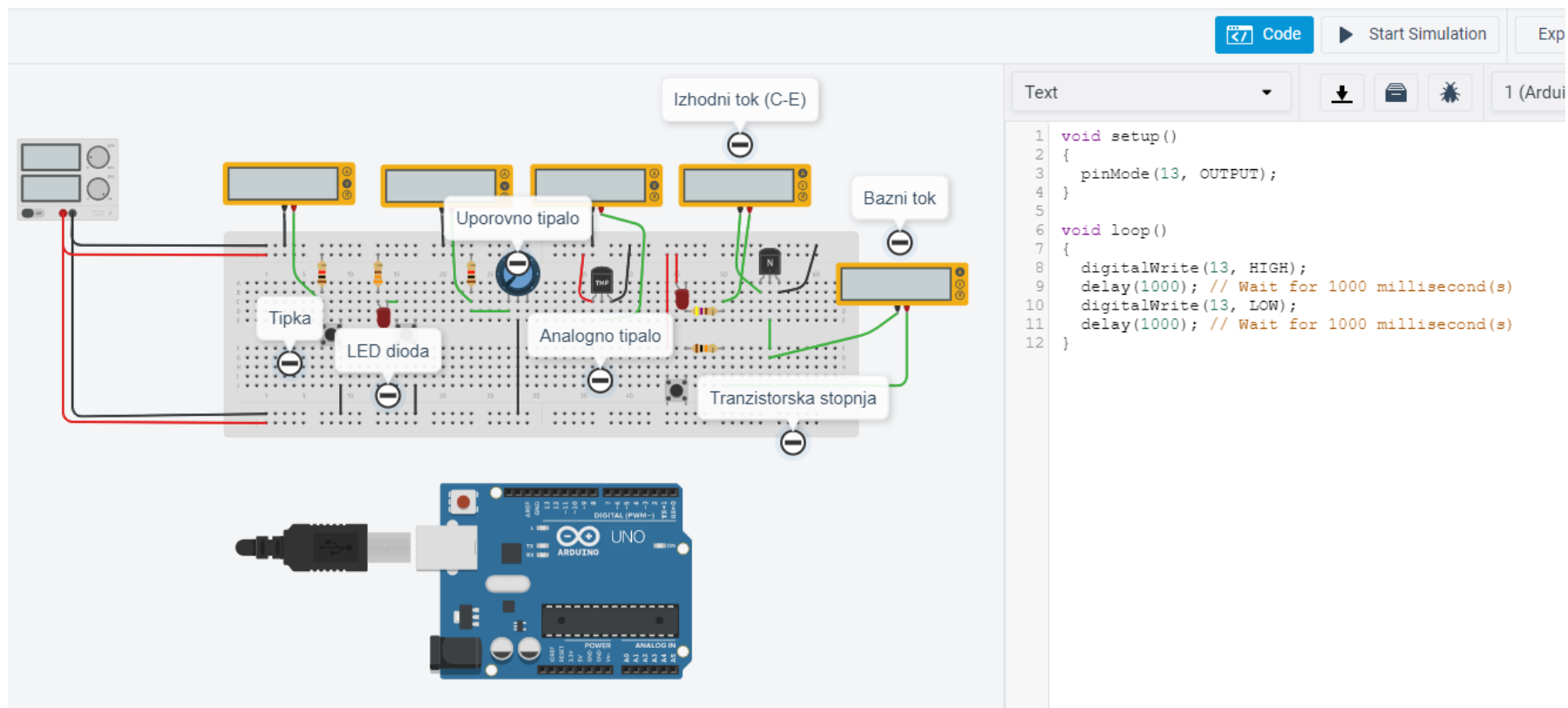
VIN - LV

SENZOR ZA ZAPORNICO

Varnostni sistem za preprečitev zaprtja parkirne zapornice v primeru, da je pod njo objekt.



TinkerCad – Domača naloga 2:



TinkerCad – Domača naloga 2:

- Spada v sklop poročila z LAB vaj
- Naredite sebi zanimivo osnovno vezje(a) z Arduino in ustrezno kodo
- Objavite v OneNote delovnem zvezku
 - Collaboration space, razdelek TinkerCad+Arduino

