



Vhodno izhodne naprave

Laboratorijska vaja 2 - VP 2
TinkerCad-Arduino osnove

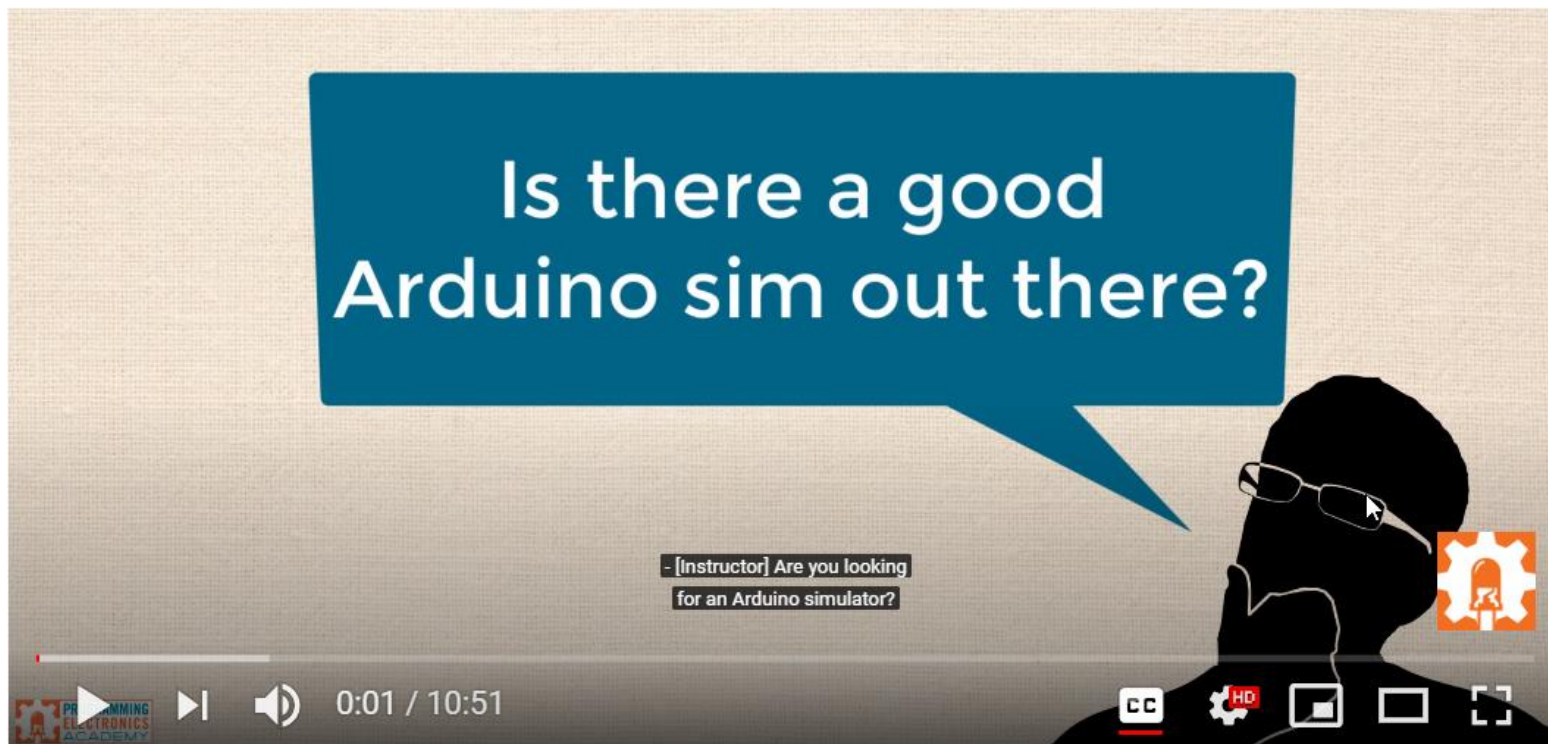
VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- TinkerCad - ponovitev
- Breadboard – osnovne vezave
- TinkerCad + Arduino
- Domača naloga (DN2-1, DN2-2)

VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- TinkerCad - ponovitev
- Breadboard – osnovne vezave
- TinkerCad + Arduino
- Domača naloga

TinkerCad: the Arduino Simulator you've been looking for!

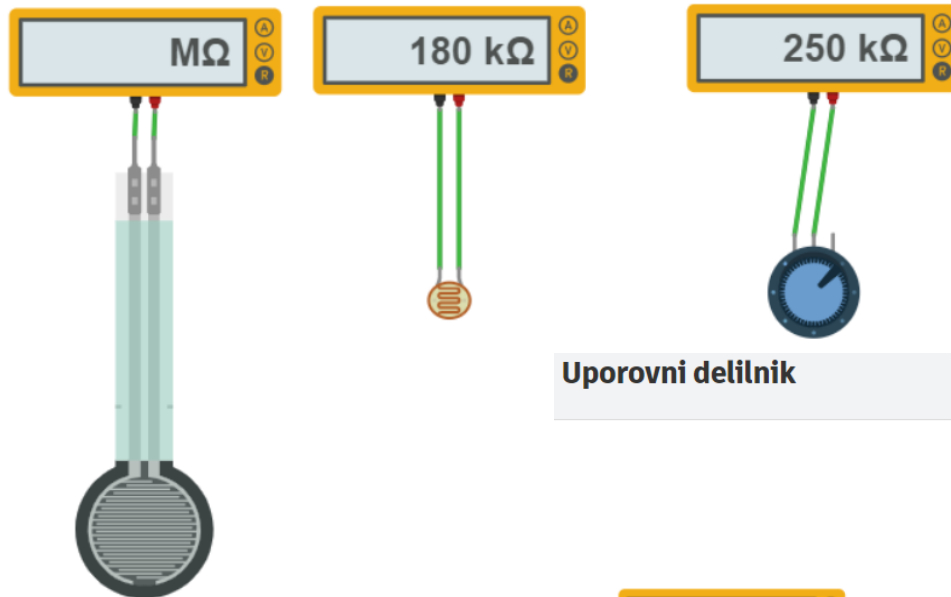


Z naslova <<https://www.youtube.com/watch?v=6uz1sCA9joc>>

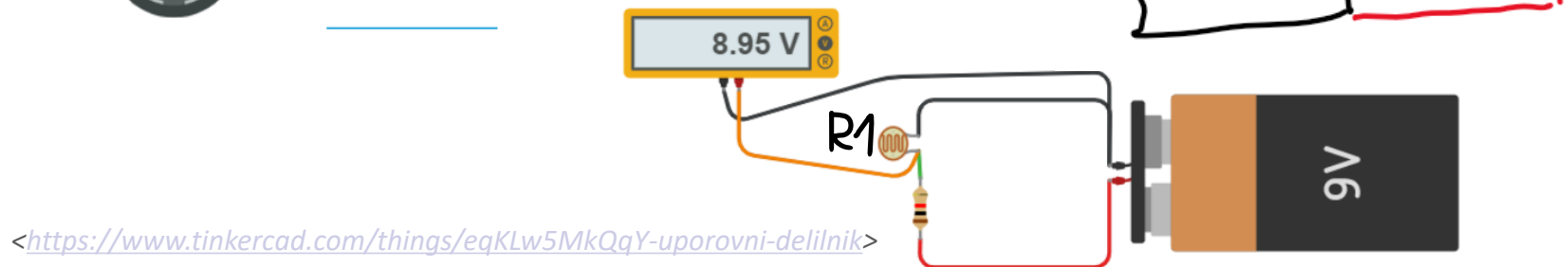
TinkerCad: Uporovna tipala in delilnik napetosti

Uporovna tipala

Z naslova <<https://www.tinkercad.com/things/gRnhGlsvr0z-uporovna-tipala>>



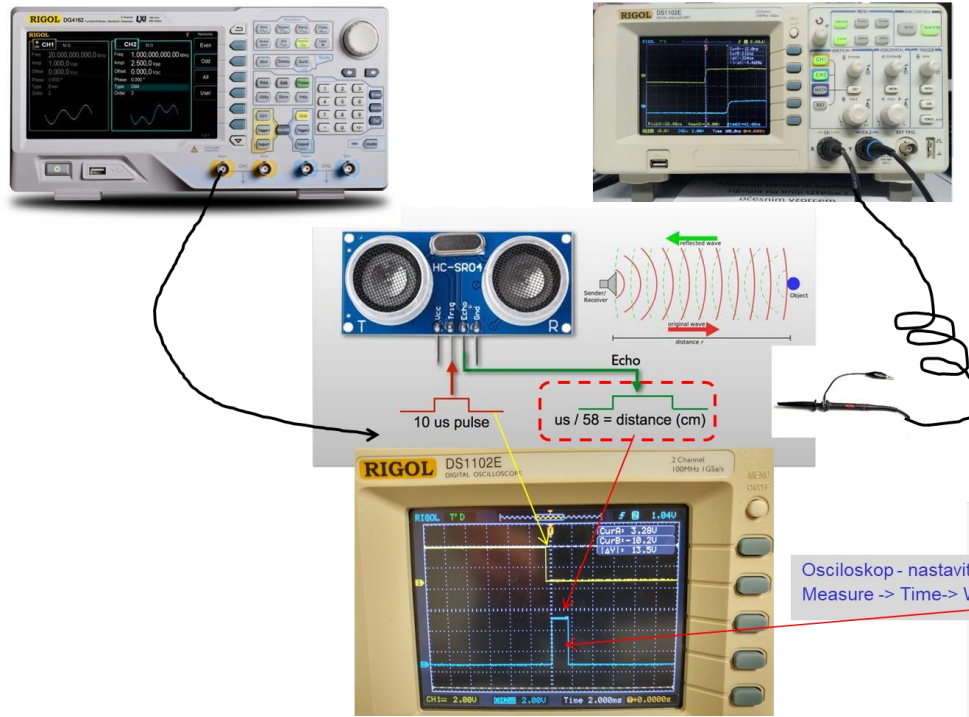
Uporovni delilnik



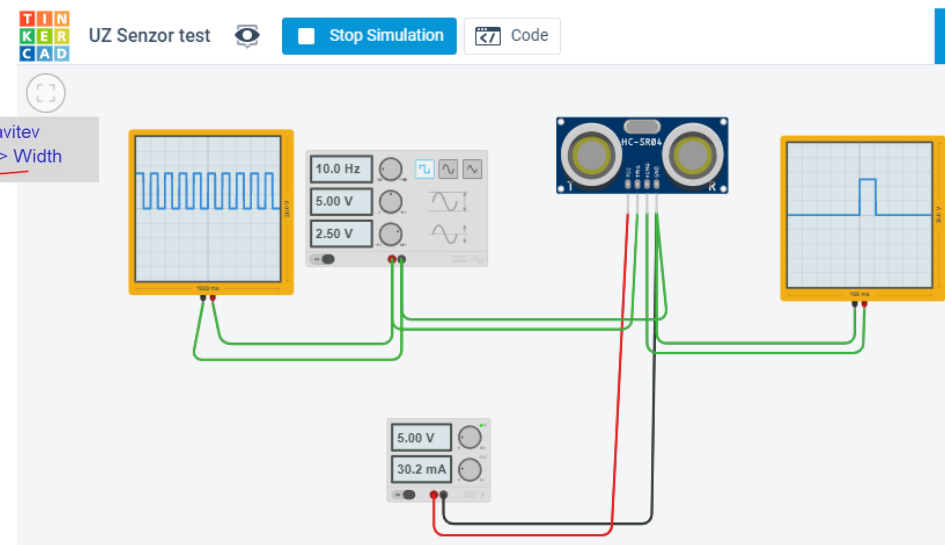
<<https://www.tinkercad.com/things/eqKLw5MkQqY-uporovni-delilnik>>

UZ senzor in HC-SR04

LAB Preizkus



Simulacija - TinkerCad



VIN projekt - VP2: TinkerCad, Breadboard, Arduino

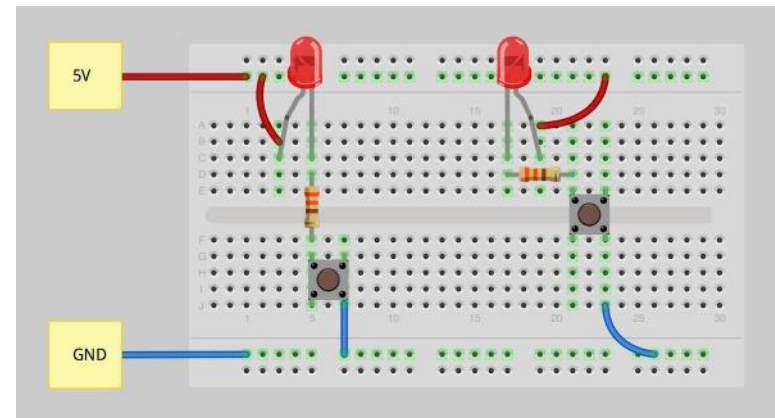
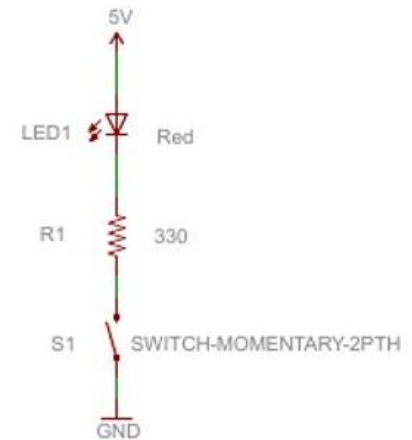
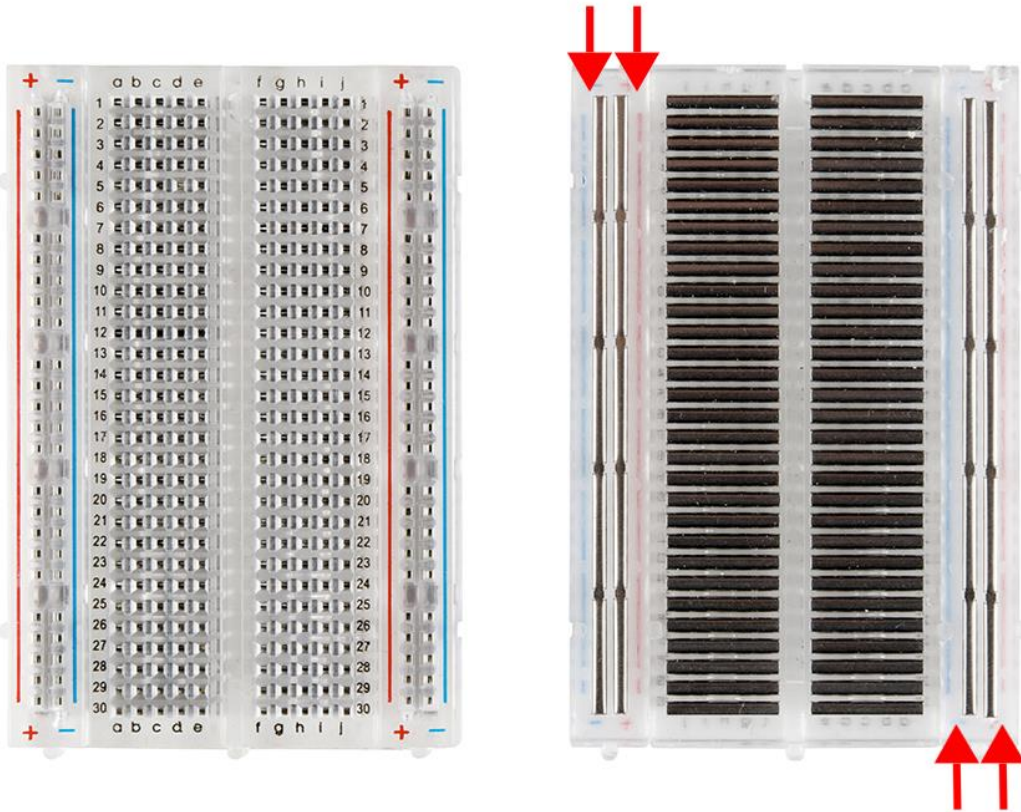
- TinkerCad - ponovitev

- Breadboard – osnovne vezave

- TinkerCad + Arduino

- Domača naloga

VIN projekt : TinkerCad Breadboard vezave

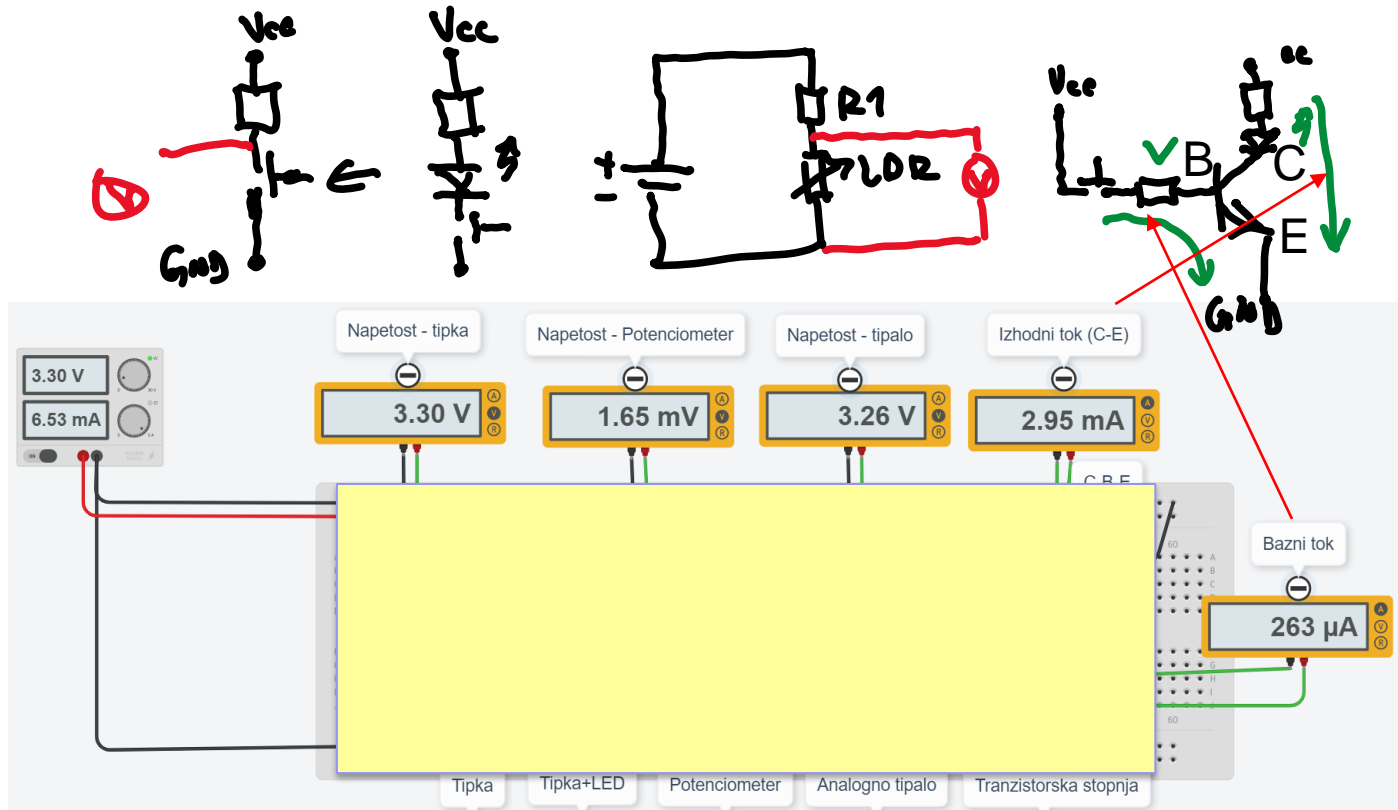


Viri

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

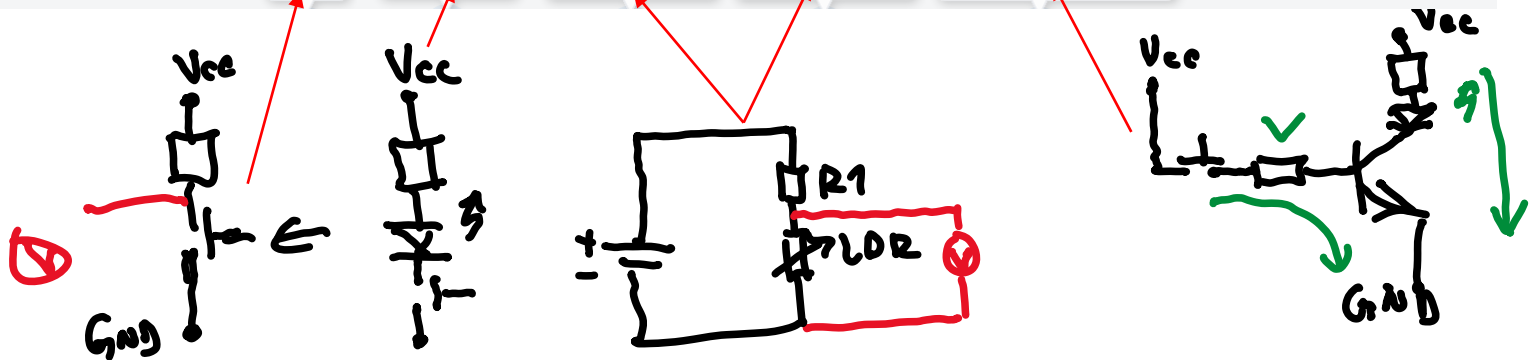
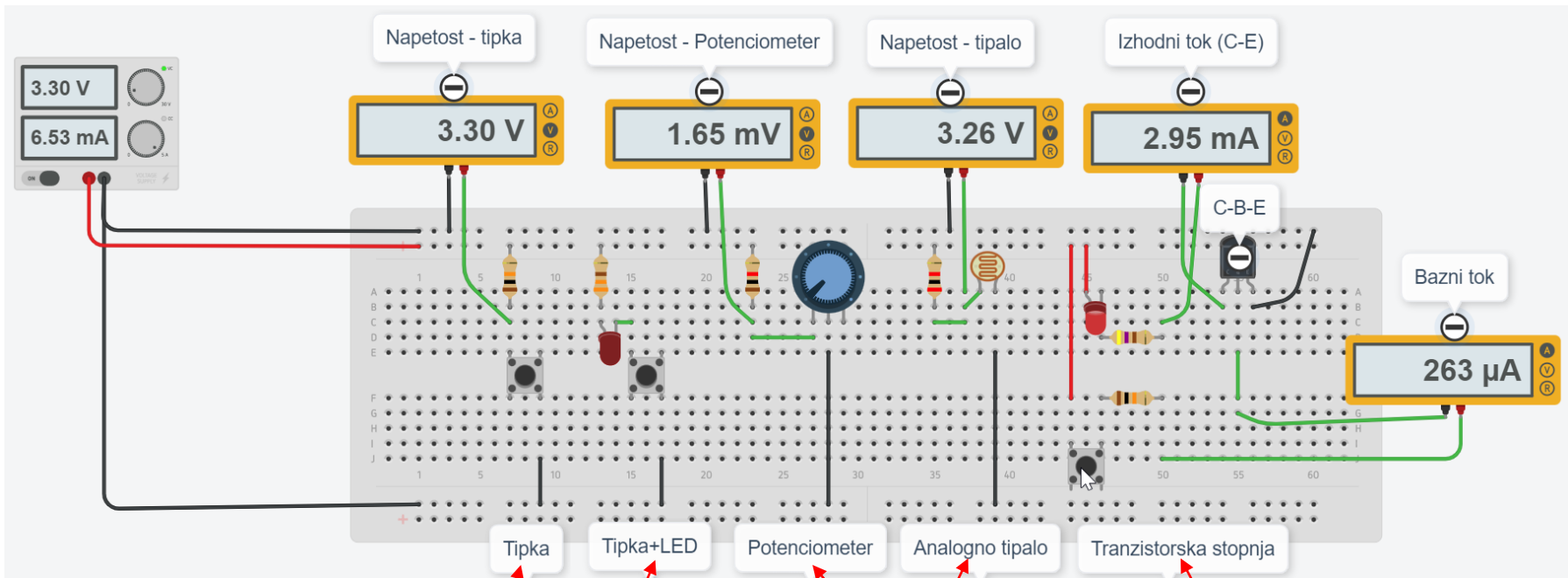
VIN projekt : TinkerCad

Breadboard – osnovne vezave



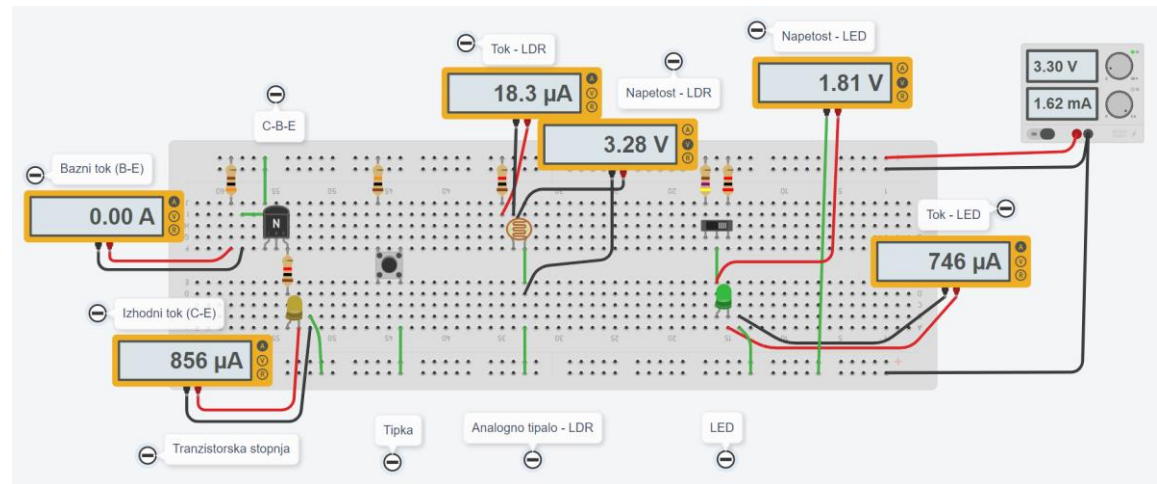
VIN projekt : TinkerCad

Breadboard vezave – Primer rešitve

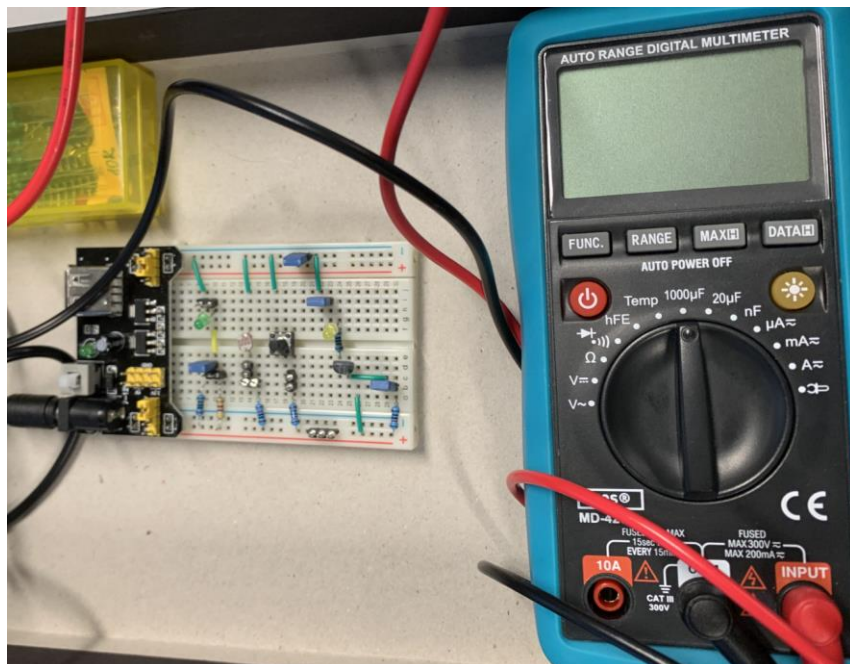


VIN projekt : Breadboard vezave + Unimer

V različnih točkah vezja lahko merimo el. napetost, tok.



<https://www.tinkercad.com/things/1UQpxVO5DSY-vin-lab-breadboard-demo>

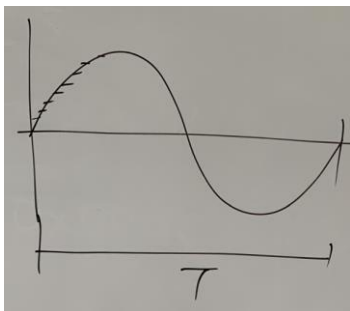


VIN projekt : Breadboard vezave + STM32H7

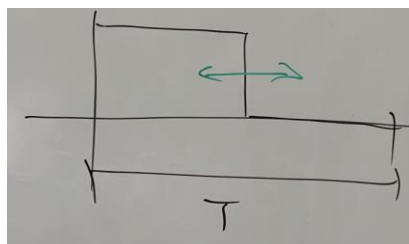
Demo

- Priključitev breadboard-a na STM32H7
- PWM melodije

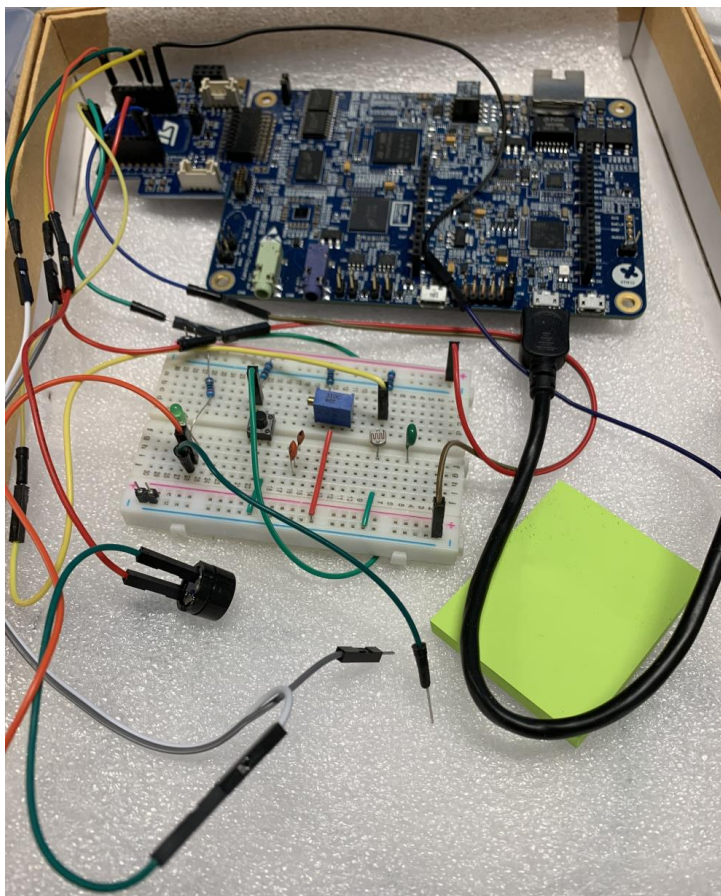
Sinusni signal (ton)



PWM signal (več tonov)



Na brenčacu s pomočjo PWM signala lahko tvorimo melodije



VIN projekt - VP2: TinkerCad, Breadboard, Arduino

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

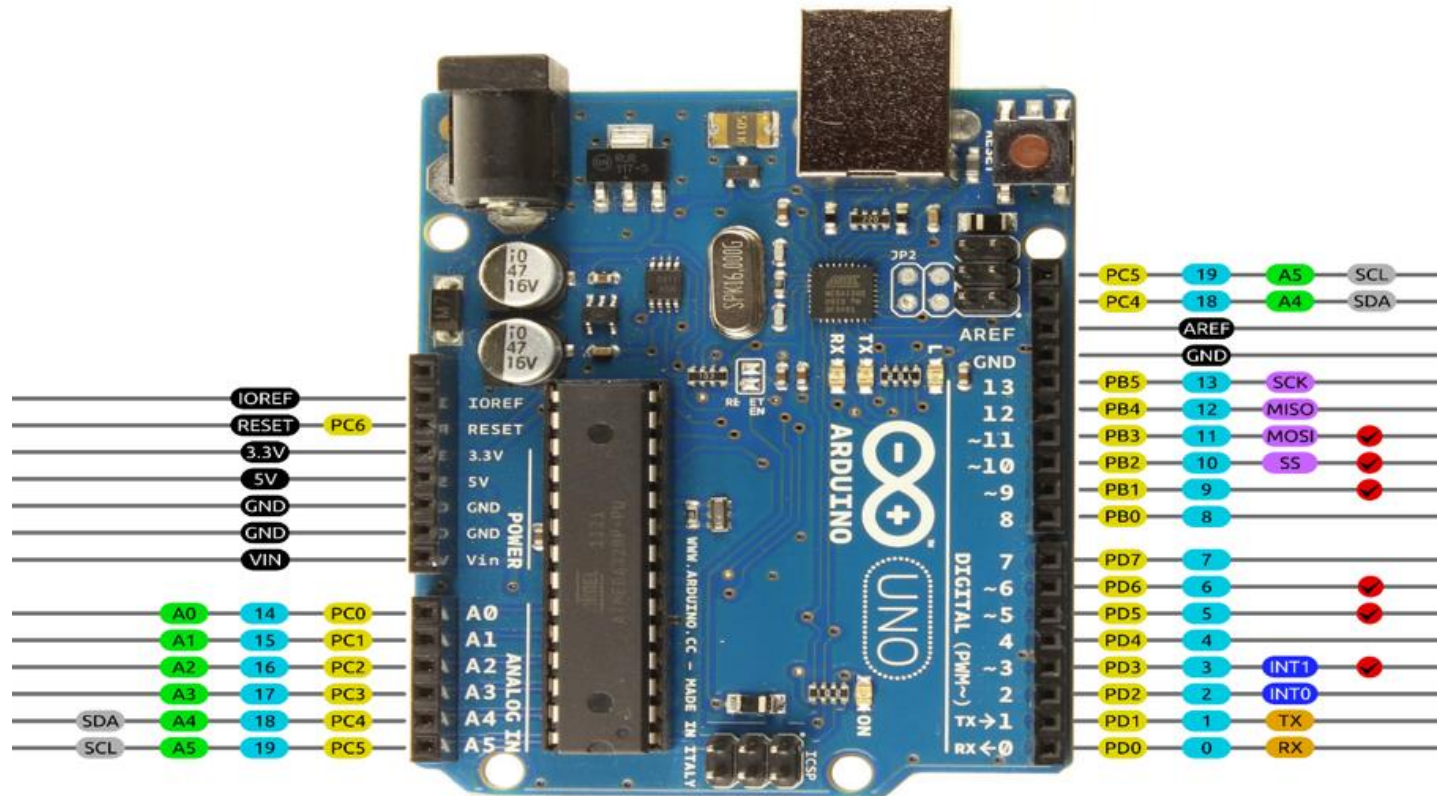
ARDUINO HW INTRO



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

■ Arduino :

5V !!!



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()`

LANGUAGE

FUNCTIONS

VARIABLES

STRUCTURE

LIBRARIES

IOT CLOUD API

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](#).

TinkerCad – razhroščevanje (debugging)



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

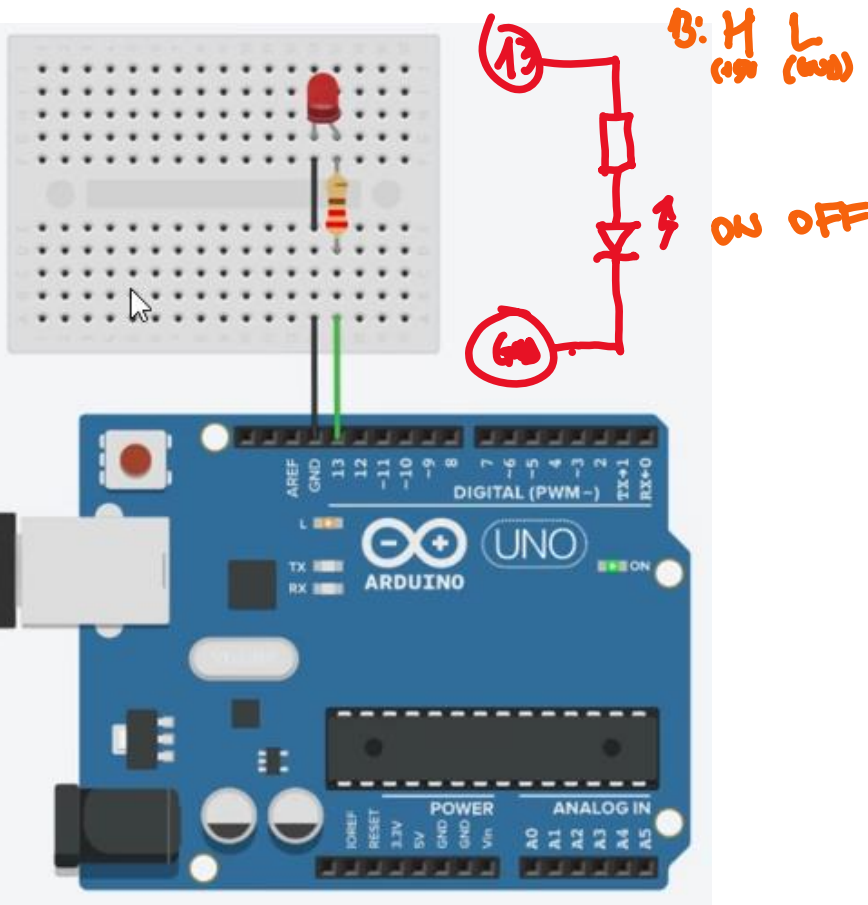
```
1
2 int val;
3 int T;
4 void setup() {
5
6     pinMode(A0, INPUT);
7     Serial.begin(9600); // Begin the Serial at 9600 Baud
8 }
9
10 void loop() {
11     val = analogRead(A0);
12     //Serial.println(val);
13     T = map(val, 20, 358, 0, 165);
14     Serial.write(T);
15     delay(1000);
16    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

Annotations in the image:

- A yellow box labeled "Prekinitev (Breakpoint)" with a red arrow pointing to line 14.
- A yellow box labeled "Izpis vrednosti" (Print values) with a red arrow pointing to the `Serial.write(T);` statement on line 14.
- A black box containing the number "153" is positioned over the `Serial.write(T);` statement, indicating the value of `T` at that point.

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

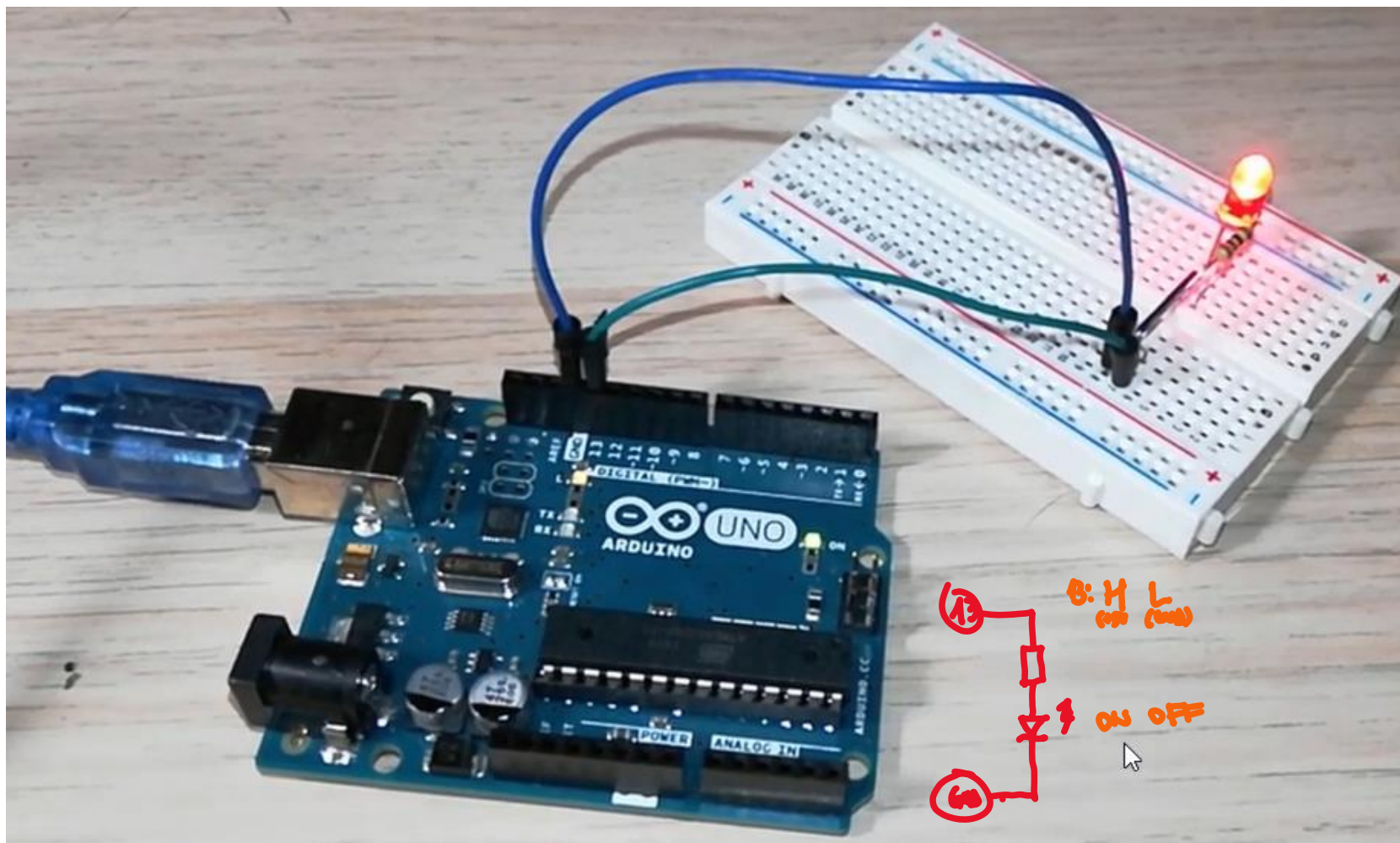
Preprosto vezje z LED diodo in program



```
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 }
```

<https://www.tinkercad.com/things/llkRDbp1u2C-vin-arduino-led-dioda>

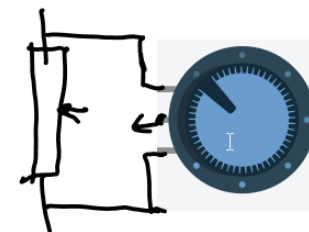
Preprosto vezje z LED diodo



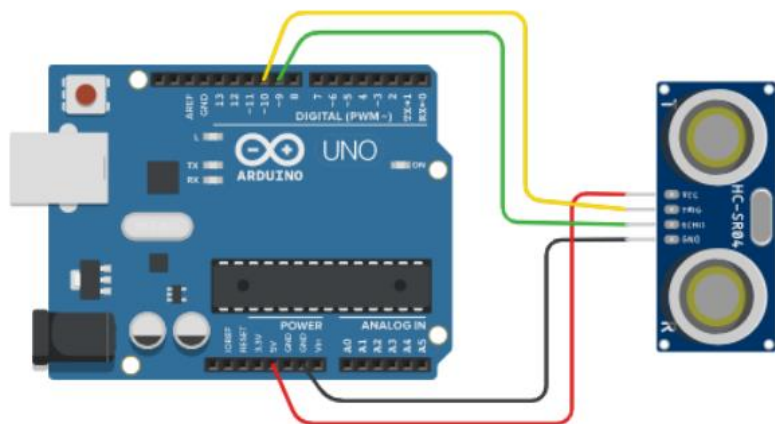
■ Vezje z uporovnim „tipalom“ (potenciometrom) - ADC

```
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 }
```

[VIN Arduino + LED dioda \(tinkercad.com\)](https://www.tinkercad.com)



■ Preprosto vezje z UZ tipalom – Časovnik (Timer)



```
Text [Download] [Save] [Run] 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);
28 }
```

<https://www.tinkercad.com/things/kAlkT1BfjB0-vin-arduino-sr04-uz-ultrazvocni-senzor>

Vežje z ultrazvočnim senzorjem SR04



```
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
```

```
}
```

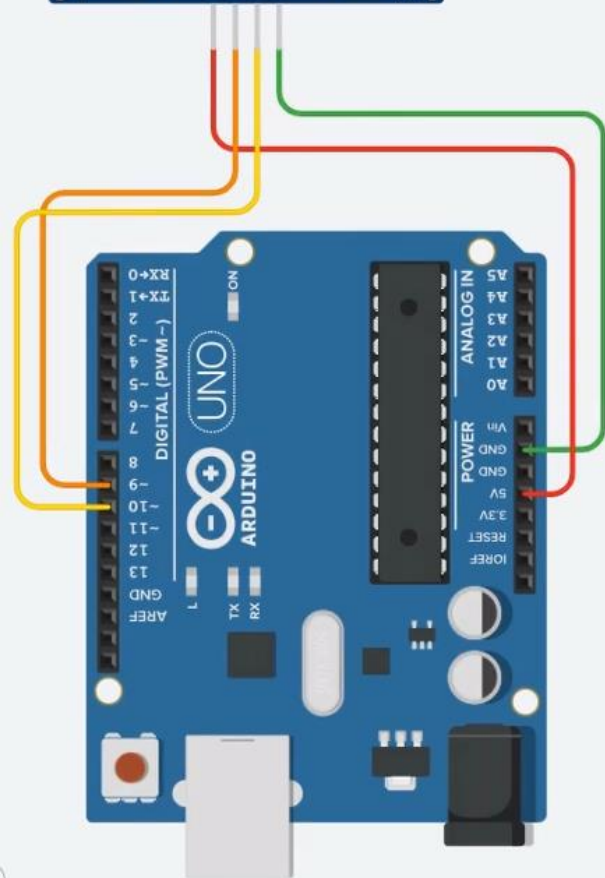
```
void loop() {
```

```
  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) :
```



Vežje z ultrazvočnim senzorjem SR04 - program

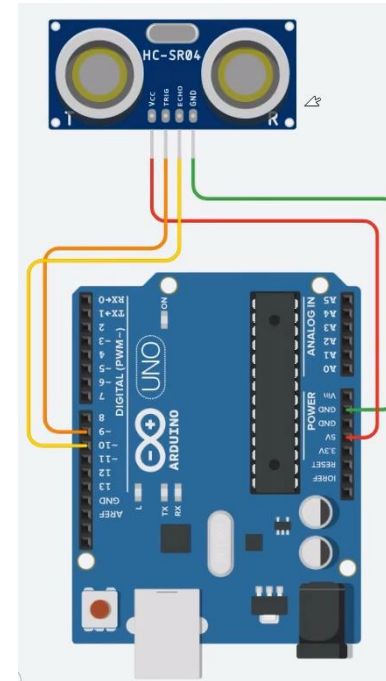
```
const byte trigger_Pin = 9; //Initialize I/O pins
const byte echo_Pin = 10;
unsigned long duration; //Since PulseIn return an unsigned Long
unsigned int distance; //To save the distance
```

```
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) :
```



<https://www.tinkercad.com/things/kAlkT1BfjB0-vin-arduino-sr04-uz-ultrazvocni-senzor>

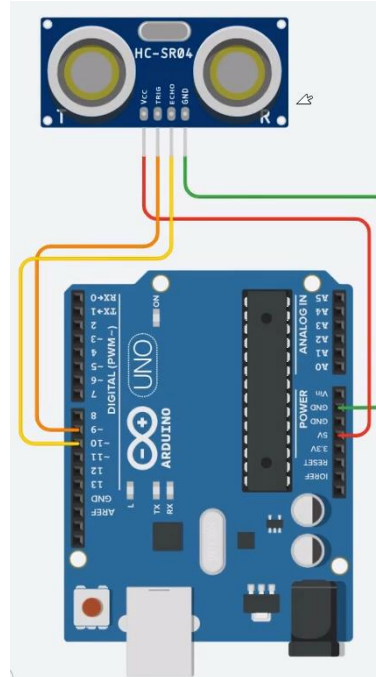
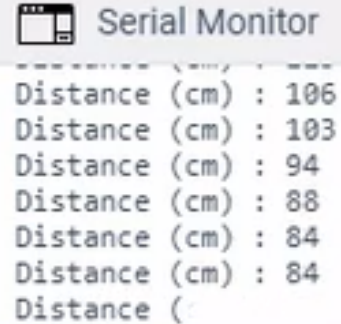
Vezje z ultrazvočnim senzorjem SR04 - program

```
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);
}
```



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
```

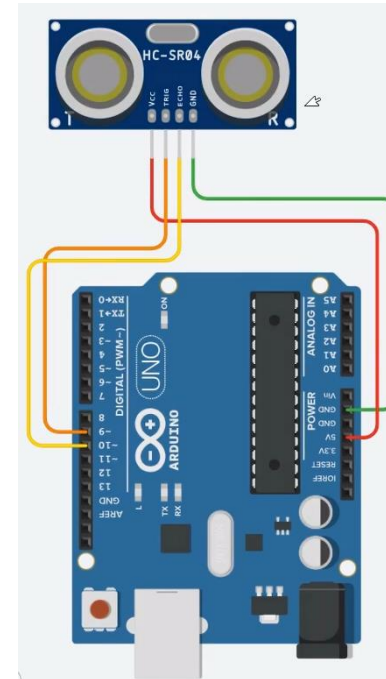
```
void setup() {
  Serial.begin(9600); //Initialize Serial communication
  pinMode(echo_Pin, INPUT); //Echo pin as Input
  pinMode(trigger_Pin, OUTPUT); //Trigger pin as Output
}
```

```
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);
}
```

<https://www.tinkercad.com/things/kAlkT1BfjB0-vin-arduino-sr04-uz-ultrazvocni-senzor>

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



Vežje 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
```

```
*/
```

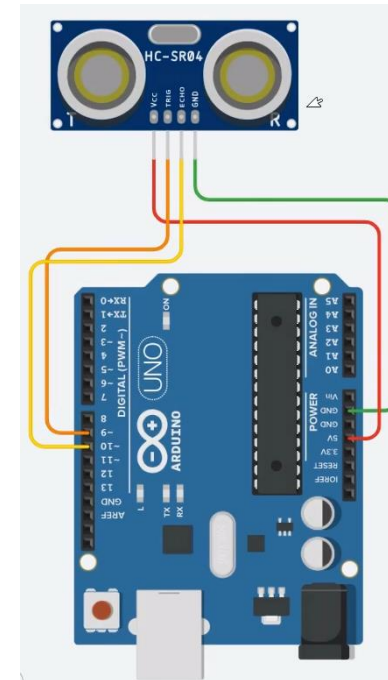
```
Serial.print("Distance (cm) : ");
```

```
Serial.println(distance);
```

```
delay(100);
```

```
}
```

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



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

<https://www.tinkercad.com/things/kAlkT1BfjB0-vin-arduino-sr04-uz-ultrazvocni-senzor>

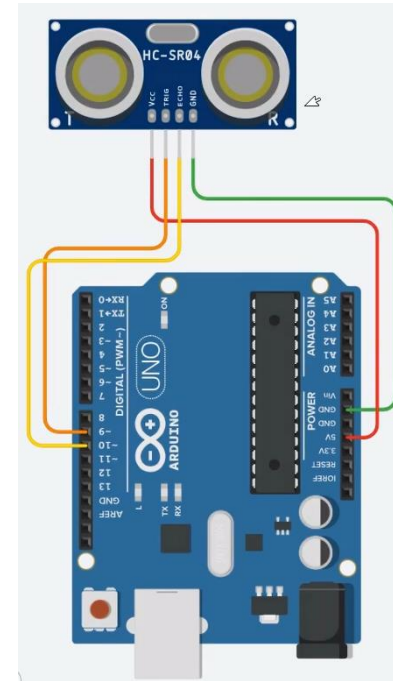
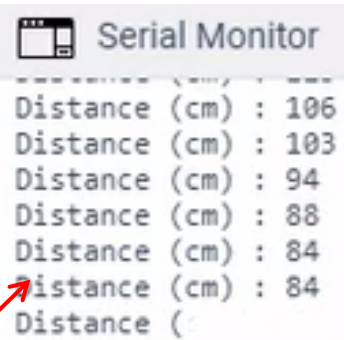
Vezje z ultrazvočnim senzorjem SR04 - program

```
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);
}
```



<https://www.tinkercad.com/things/kAlkT1BfjB0-vin-arduino-sr04-uz-ultrazvocni-senzor>

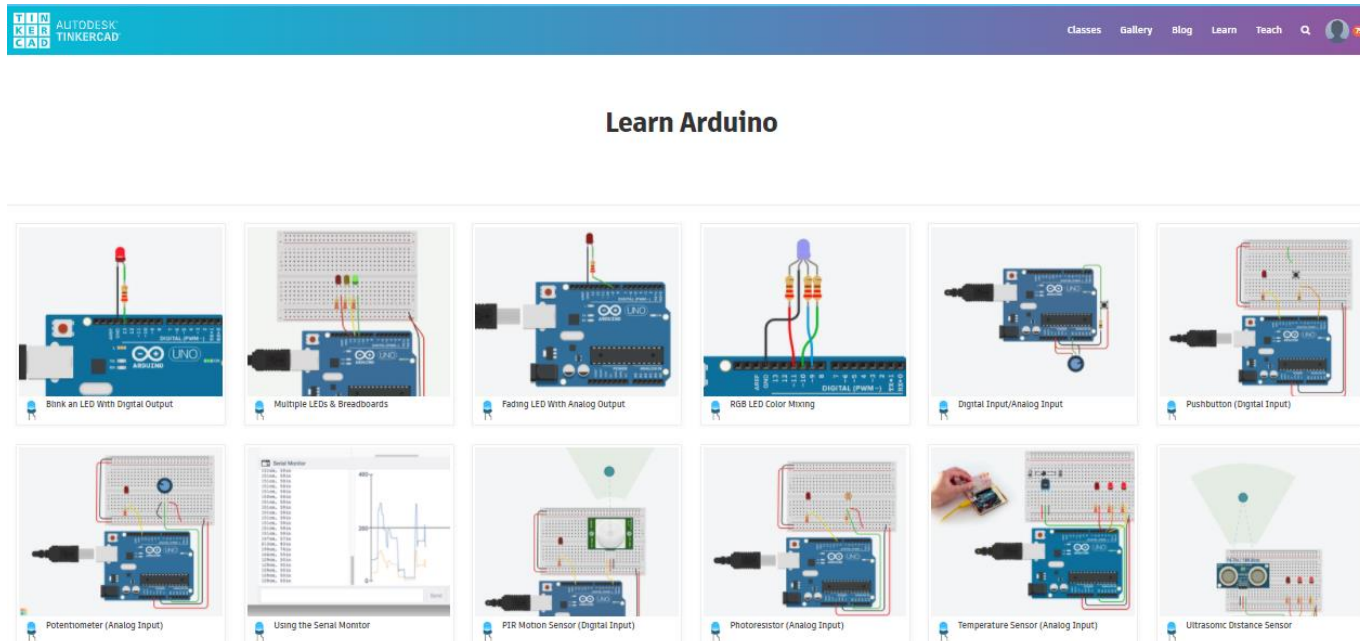
■ Viri – TinkerCad + Arduino :

□ 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

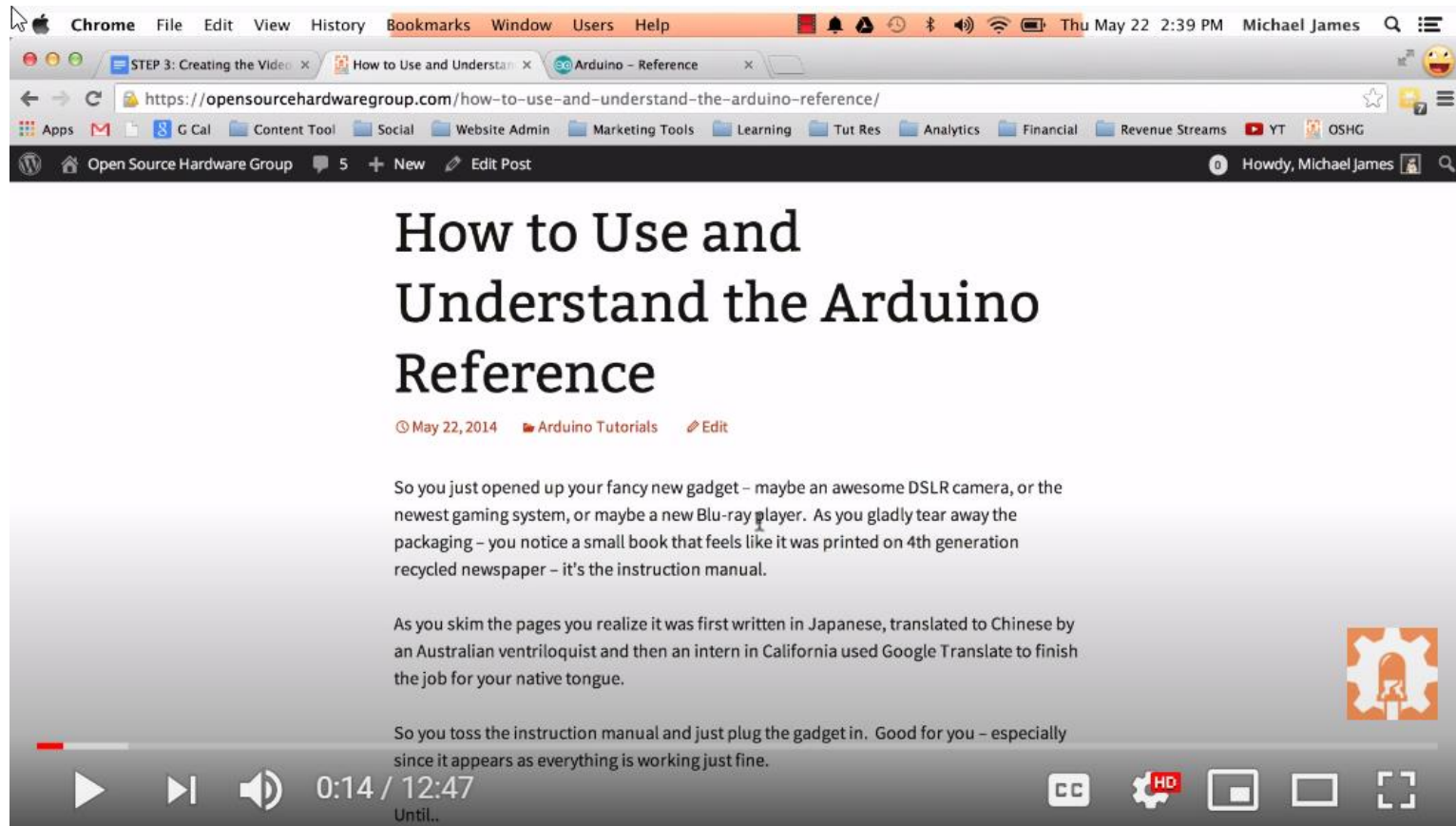
■ How to Use and Understand the Arduino Reference :: Open Source Hardware Group Arduino Tutorials

- 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 Chrome browser window displaying a video player. The video player is showing a webpage from 'opensourcehardwaregroup.com' with the title 'How to Use and Understand the Arduino Reference'. The video player controls at the bottom show a progress bar at 0:14 / 12:47, a play button, a volume icon, and a settings icon. The webpage content includes a title, a date (May 22, 2014), a category (Arduino Tutorials), and an edit link. The main text of the video reads: '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.'

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

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

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.

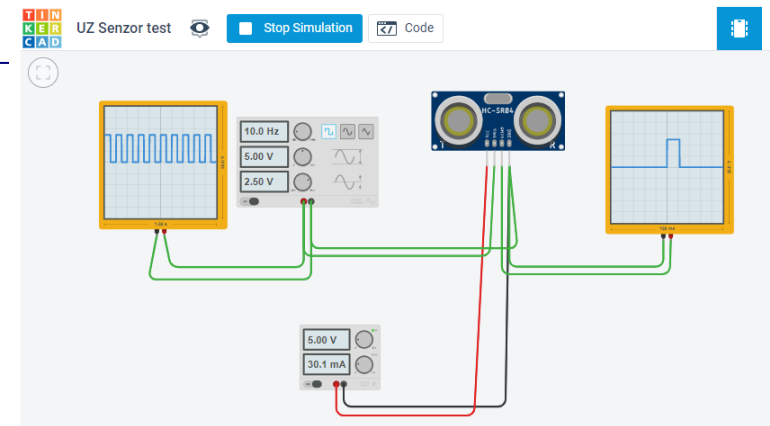
VIN projekt - VP2: TinkerCad, Breadboard, Arduino

- TinkerCad - ponovitev
 - Breadboard – osnovne vezave
 - TinkerCad + Arduino
- Domača naloga

VP1: TinkerCad in osnovne vezave

TinkerCad – DN2-VP1 :

- Spada v sklop poročila z LAB vaj (DN2)
- Naredite sebi zanimivo osnovno vezje(a),
 - še brez uporabe mikrokrmilnika (Arduino)
- Objavite v OneNote delovnem zvezku
 - Prostor za sodelovanje, razdelek DN2-VP1 TinkerCad_Osnova



OneNote za Windows 10

Osnovno Vstavlanje Risanje Ogled Pomoč Zvezek za predavanja

Calibri Light 20 K L P A

VIN-VSP 202324 zvezek

- Dobrodošli
- Preberi.me
- _Collaboration Space
- Uporaba prostora za sodel...
- DN1-VI naprave
- DN2-VP1 TinkerCad**
- DN2-VP2 TinkerCad+ARD
- VIN Projekt-Viri
- VIN Projekt-Ideje

Preberi.me

sreda, 16. marec 2022 18:09

Tukaj objavite svoje rešitve naloge:

- Naredite svojo stran z naslovom rešitve
- Par stavkov opisa, slika in povezava na [TinkerCad](#) vezje
- Rešitev shranite v svojem zvezku za vključitev v DN2 poročilo z laboratorijskih vaj

TinkerCad – DN2-VP1: Primer zapisa (objave)

Preprosto vezje za preizkus delovanja UZ tipala razdalje

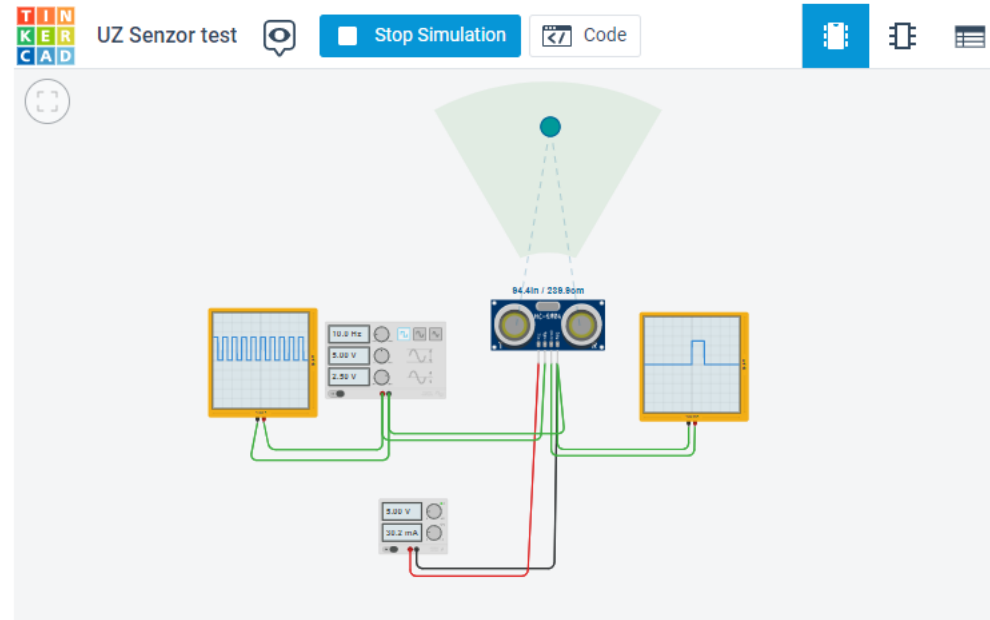
Sestavni deli:

- Napajalnik skrbi za napajanje tipala
- Generator pravokotnega signala proži Trigger vhod tipala, da sproži oddajo UZ signala
- Osciloskop prikaže izhod tipala, ki s širino pulza ponazarja oddaljenost objekta oziroma čas potovanja signala do objekta in nazaj

Opis delovanja:

Interaktivno se lahko z miško spreminja pozicija oziroma oddaljenost objekta, kar se odrazi na širini impulza na izhodu UZ tipala (osciloskop).

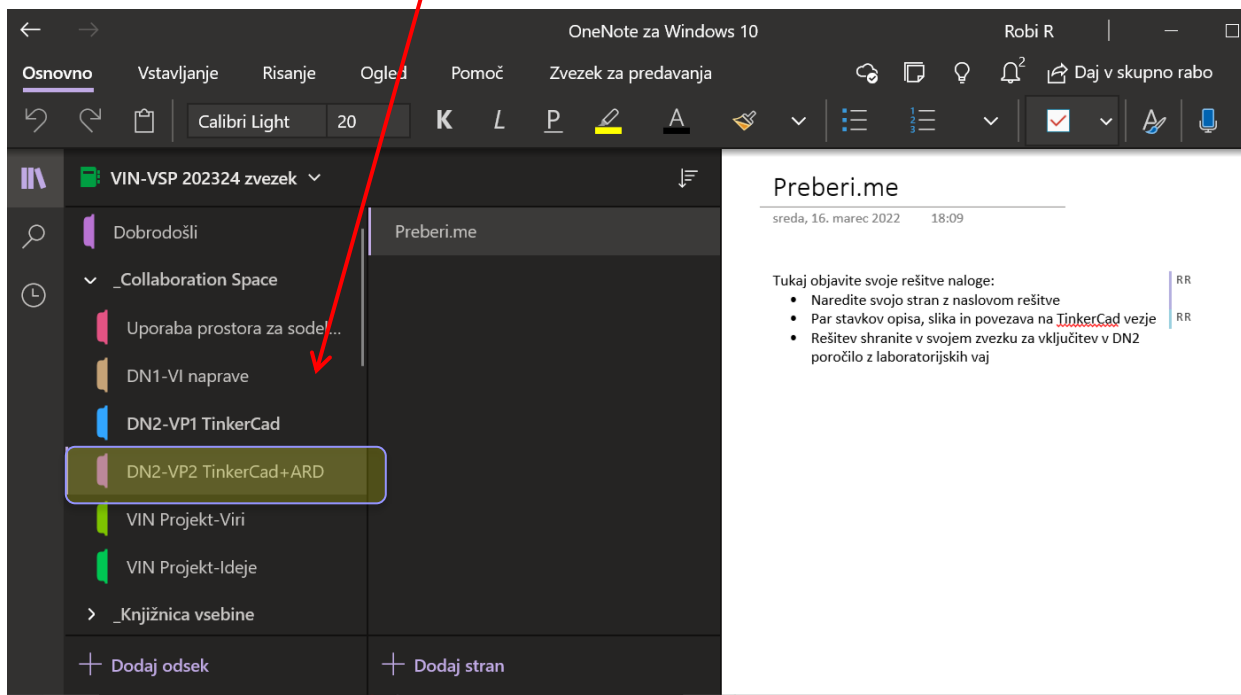
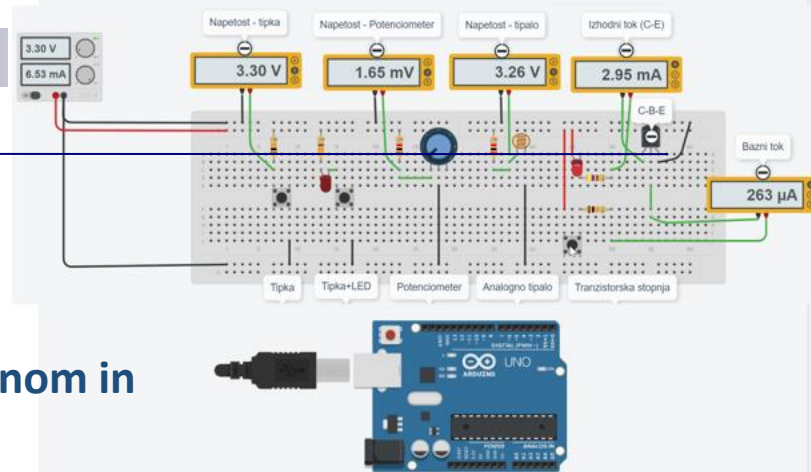
UZ Sensor test



<https://www.tinkercad.com/things/k6it1PauvwW>

TinkerCad – DN2-VP2:

- Spada v sklop poročila z LAB vaj
- Naredite sebi zanimivo osnovno vezje(a) z Arduino in ustrezno kodo
- Objavite v OneNote delovnem zvezku
 - Prostor za sodelovanje, razdelek DN2-2 TinkerCad+Arduino



TinkerCad – Domača naloga DN2-VP2:

The screenshot displays a TinkerCad workspace with a breadboard circuit and an Arduino Uno. The circuit is connected to a power source showing 3.30 V and 6.53 mA. Several measurement points are highlighted with callouts:

- Napetost - tipka: 3.30 V
- Napetost - Potenciometer: 1.65 mV
- Napetost - tipalo: 3.26 V
- Izhodni tok (C-E): 2.95 mA
- Bazni tok: 263 μ A

Labels for components on the breadboard include: Tipka, Tipka+LED, Potenciometer, Analogno tipalo, and Tranzistorska stopnja. The code editor on the right contains the following code:

```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
    
```

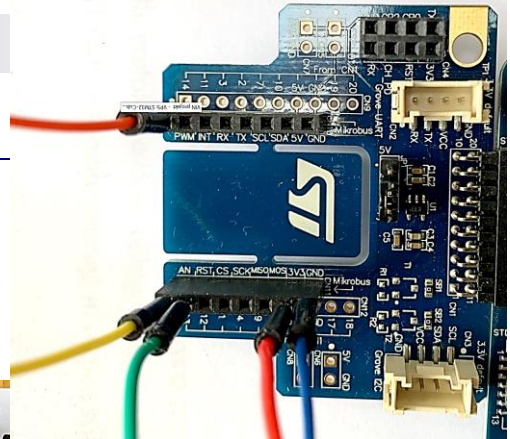
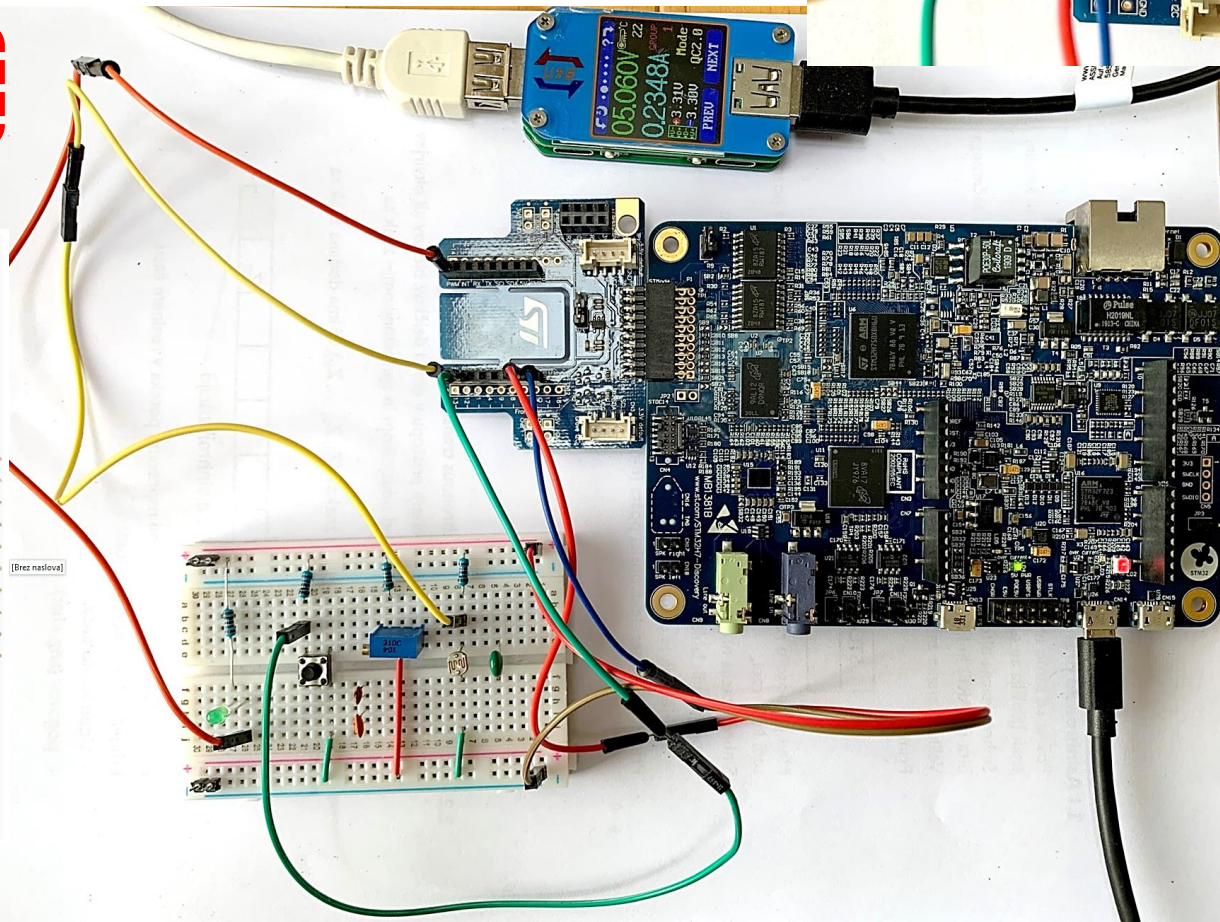
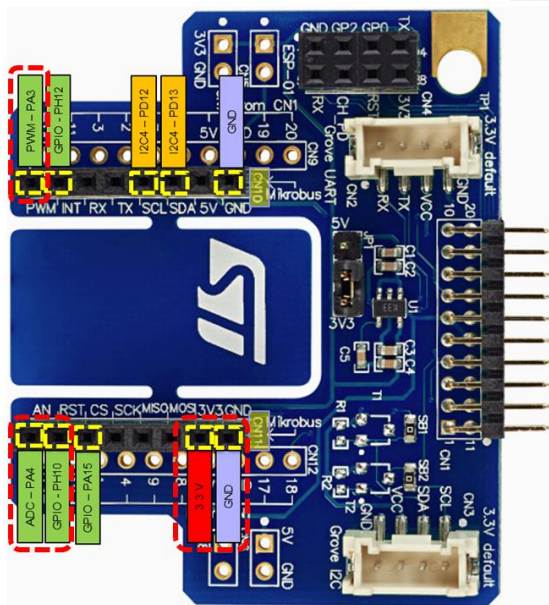
Napoved LAB vaje VP5: Breadboard + STM32H7

Priključitev na STM32 : 1x analogni,
1x digitalni vhod, 1x digitalni izhod

Testno vezje (primer) - STM32H7 :

GPIO	Vrsta	Povezava
PC13	User tipka	Modra tipka
PA4	Analogni vhod	Rumena žička
PH10	Dig. Vhod	Zelena žička
PA3	Dig. Izhod - LED	Oranžna žička
PJ2,PI13	Dig. Izhodi	vgr. LED diode

STM32H7



Napoved LAB vaje VP5: Breadboard + STM32F4

STM32F4

The image displays a circuit simulation and a physical breadboard setup. The simulation includes a digital multimeter with four readings: 3.30 V, 1.65 mV, 3.26 V, and 2.95 mA. It also shows a current source labeled 'Bazni tok' with a reading of 263 µA. A code editor window displays C++ code for an Arduino Uno.

```
Code Start Simulation Exp  
Text 1 (Ardui  
1 void setup()  
2 {  
3   pinMode(13, OUTPUT);  
4 }  
5  
6 void loop()  
7 {  
8   digitalWrite(13, HIGH);  
9   delay(1000); // Wait for 1000 millisecond(s)  
10  digitalWrite(13, LOW);  
11  delay(1000); // Wait for 1000 millisecond(s)  
12 }
```

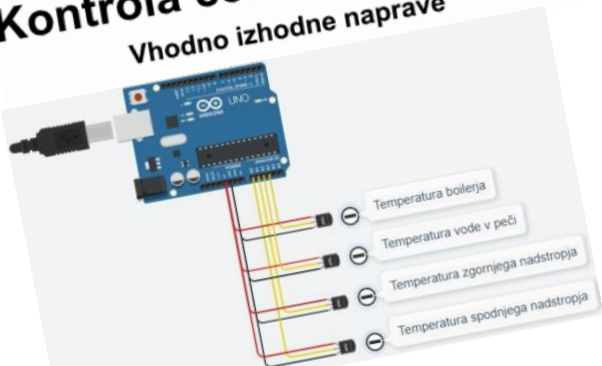
The physical setup shows an Arduino Uno connected to a breadboard with various components like resistors, a potentiometer, and an LED. Labels in the simulation include: Napetost - tipka, Napetost - Potenciometer, Napetost - tipalo, Izhodni tok (C-E), Bazni tok, Tipka, Tipka+LED, Potenciometer, Analogno tipalo, and Tranzistorska stopnja.

VIN Projekt Arduino – Primeri iz I. 19/20

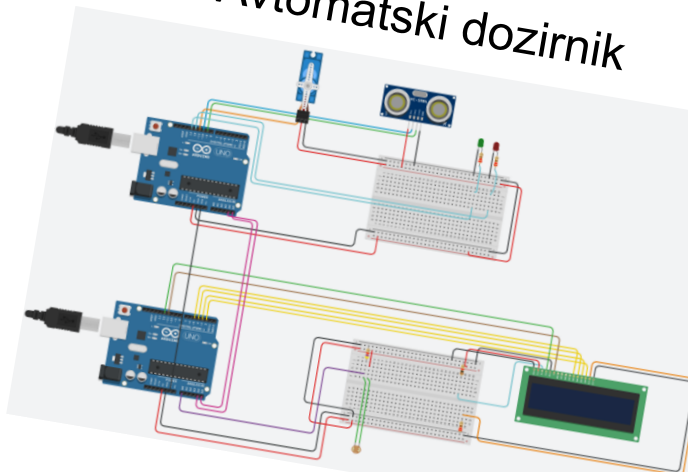
TinkerCad

Arduino projekt,
Kontrola centralne kurjave

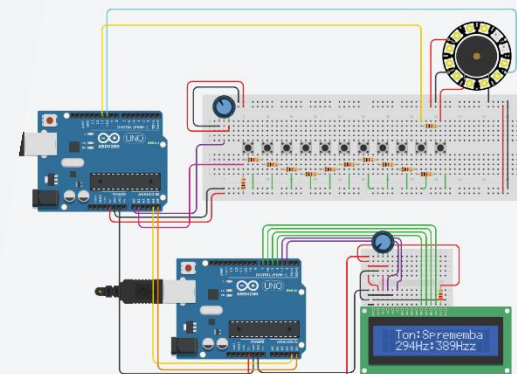
Vhodno izhodne naprave



Avtomatski dozirnik



Mini Piano



Sledenje



HW izvedba

SENZOR ZA ZAPORNICO

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

