



# Vhodno izhodne naprave

Laboratorijska vaja 5 - VP 4  
STM32-CubeIDE projekt, breadboard  
vezave, komunikacije

# VIN projekt - VP4: STM32-CubeIDE projekt, breadboard vezave, komunikacije

- Osvežitev

- Breadboard vezava

- STM32 CubeIDE + Breadboard

- LED, tipka, potenciometer, uporovna tipala
- PWM brenčoč z melodijami

- Primeri komunikacijskih projektov

- STM32+LIS3DSH
- Arduino

# VIN Projekt – Osnovna platforma

## STM32F407 ST Discovery

### STM Discovery F4 (Cortex M4)

- STM32F407VGT6 microcontroller featuring 32-bit Arm® Cortex®-M4 with FPU core, 1-Mbyte Flash memory and 192-Kbyte RAM in an LQFP100 package

### •USB OTG FS

### •ST MEMS 3-axis accelerometer

### •ST-MEMS audio sensor omni-directional digital microphone

### •Audio DAC with integrated class D speaker driver

### •User and reset push-buttons

### •Eight LEDs:

- LD1 (red/green) for USB communication
- LD2 (red) for 3.3 V power on
- Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)

### •Board connectors:

- USB with Micro-AB
- Stereo headphone output jack
- 2.54 mm pitch extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing

### •External application power supply: 3 V and 5 V

# STM32

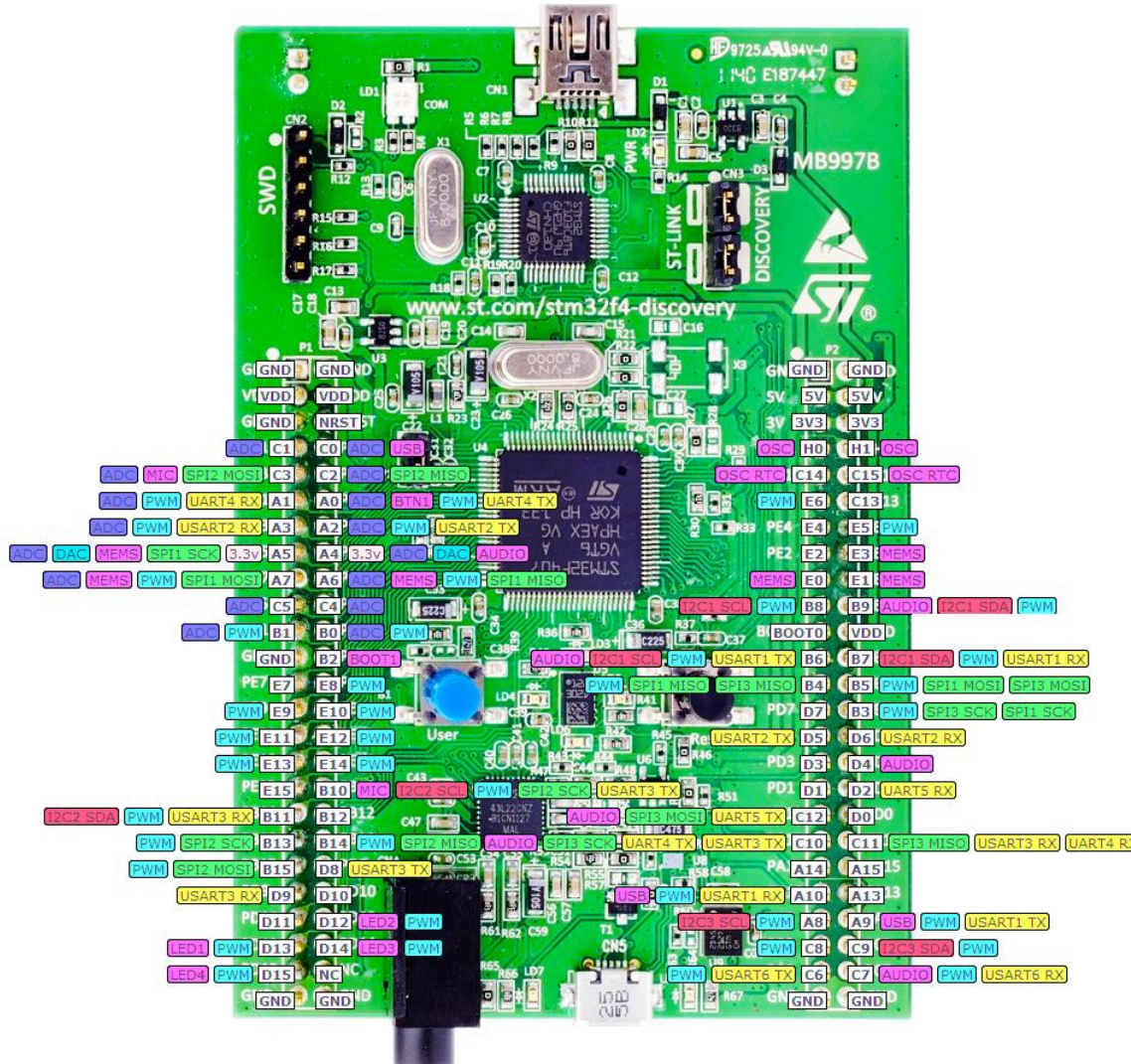


# STM32F4DISCOVERY

# 3.3V !!!

## P1

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50



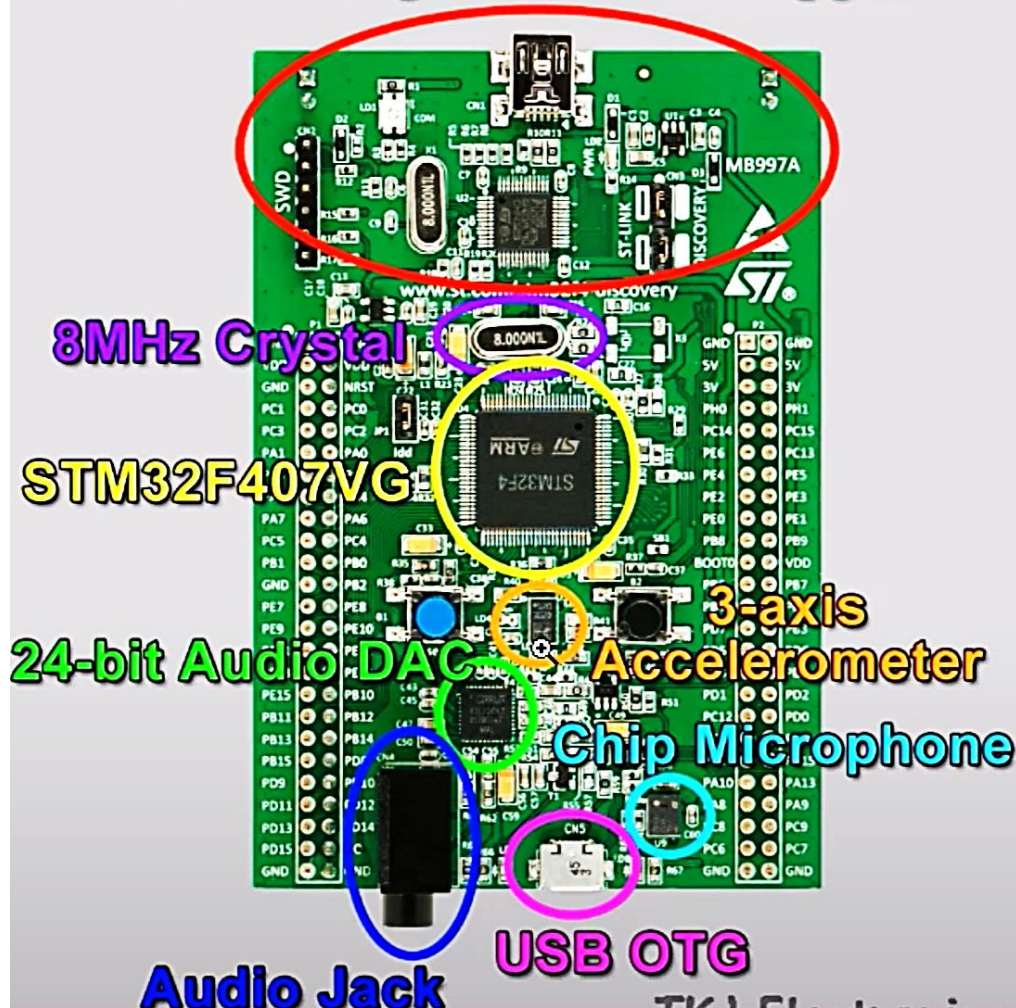
## P2

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50

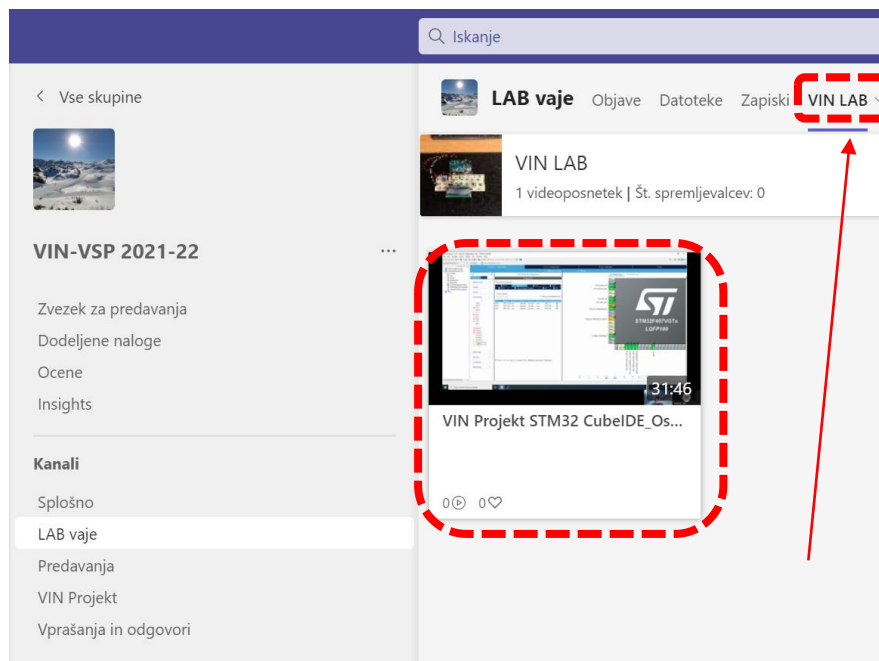


# STM32F4DISCOVERY USB Programmer/Debugger

3.3V !!!



# Delo na STM32F4 razvojnem sistemu



## Lastni viri :

[https://github.com/LAPSyLAB/STM32F4\\_Discovery\\_VIN\\_Projects](https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects)

[https://github.com/LAPSyLAB/STM32F4\\_Docs\\_and\\_Examples](https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples)

<https://github.com/LAPSyLAB/ORLab-STM32>

# Delo na STM32F4 razvojnem sistemu

## Priključitev :

- **Mini USB** prikllop na **krajši stranici**, svetila rdeči **LED** diodi

## STM32 CubeIDE

- <https://www.st.com/en/development-tools/stm32cubeide.html>

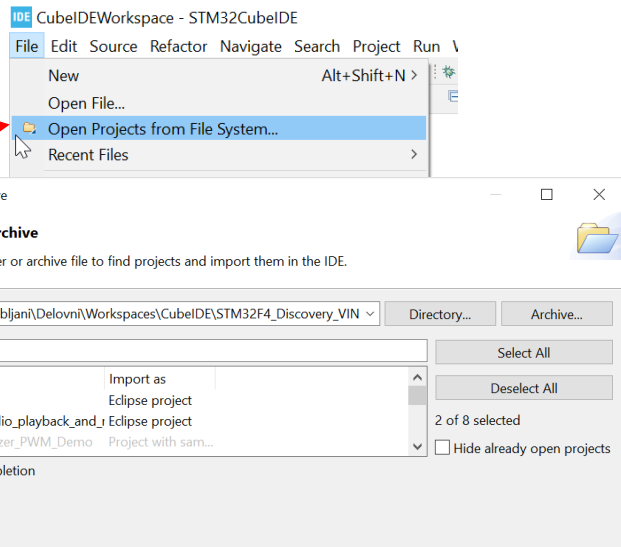


```
107 /* USER CODE BEGIN 2 */
108
109 HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
110 HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_2);
111 HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_3);
112 HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_4);
113
114 /* USER CODE END 2 */
115
116 /* Infinite loop */
117 /* USER CODE BEGIN WHILE */
118 while (1)
119 {
120     htim4.Instance->CCR1 = duty;
121     htim4.Instance->CCR2 = 100-duty;
122     htim4.Instance->CCR3 = duty;
123     htim4.Instance->CCR4 = 100-duty;
124
125     /* USER CODE END WHILE */
126
127     /* USER CODE BEGIN 3 */
128     snprintf (SendBuffer, BUFSIZE, "USB:0-1 secs. Duty=%d%%\r\n", duty);
129     CDC_Transmit_FS(SendBuffer, strlen(SendBuffer));
130
131     duty = (duty + 1) ;
132     if (duty > 100 )
133         duty = 0;
134
135     HAL_Delay(100);
136 }
137 /* USER CODE END 3 */
138
139 /**
140  * @brief System Clock Configuration
141  * @retval None
142  */
143 void SystemClock_Config(void)
144 {
145     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
146     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
147
148     /** Configure the main internal regulator output voltage
149     */
150     if (HAL_PWREx_EnableVDDIO2() != HAL_OK)
151     {
152         Error_Handler();
153     }
154     /** Initializes the CPU, I/O and memory clock domains
155     */
156     HAL_RCC_OscConfig(&RCC_OscInitStruct);
157     HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2);
158
159     /** Configure the main internal regulator output voltage
160     */
161     if (HAL_PWREx_EnableVDDIO2() != HAL_OK)
162     {
163         Error_Handler();
164     }
165 }
```

# Delo na STM32F4 razvojnem sistemu

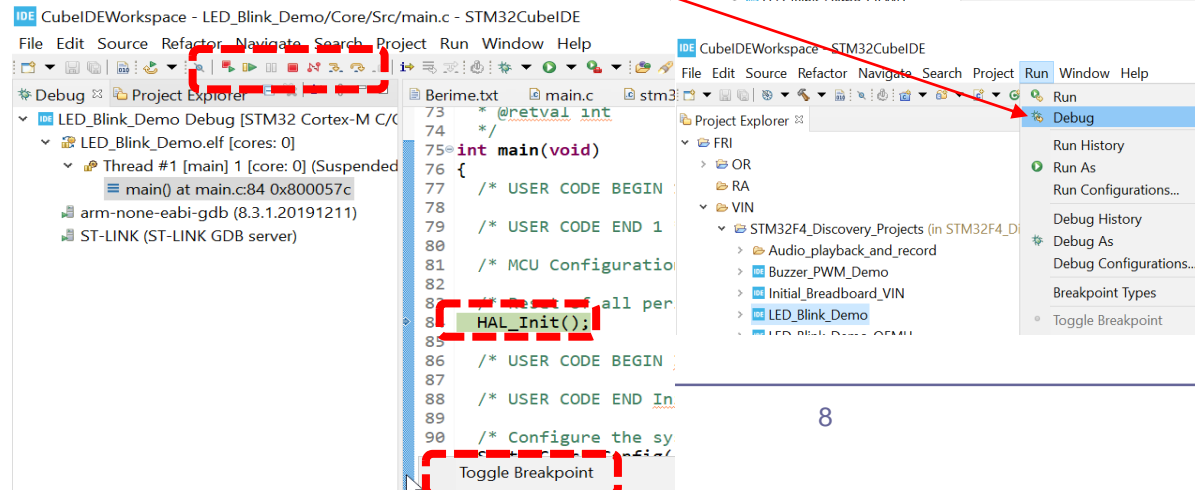
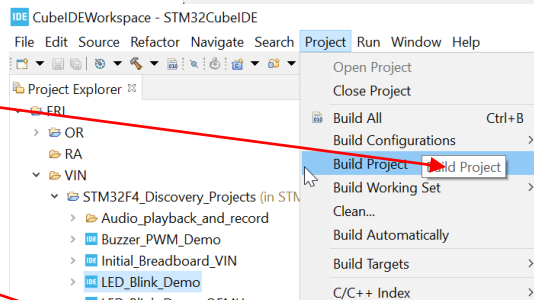
## Vzpostavitev začetnega projekta :

- Uvoz obstoječega
  - Open projects from File System
  - Select project(s)
- **Nov projekt Cube MX ->**  
(v nadaljevanju)



## Prevajanje, zagon :

- Project -> Build Project
- Run -> Debug
- Step (Into, Over), Breakpoints



- Navodila :
- CubeIDE asm projekt
    - 1) Edit > Copy.
    - 2) Edit > Paste.
    - 3) Delete the Debug launch file.
    - 4) Project > Clean.
    - 5) Project > Build Project.
    - 6) Debug As Stm32 Application.
    - 7) And debug the application
    - 8) Add breakpoint on first instruction if necessary
  - CubeIDE projekt z CubeMX
    - 1) Edit > Copy.
    - 2) Edit > Paste.
    - 3) Rename the ioc files.
    - 4) Delete the Debug launch file.
    - 5) Project > Clean.
    - 6) Generate the CubeMX.
    - 7) Project > Build Project.
    - 8) Debug As Stm32 Application.
    - 9) And debug the application.

Skopiram, preimenujem, generiram ioc, clean in build



# VIN projekt - VP4: STM32-CubeIDE projekt, breadboard vezave, komunikacije

- Osvežitev

- Breadboard vezava

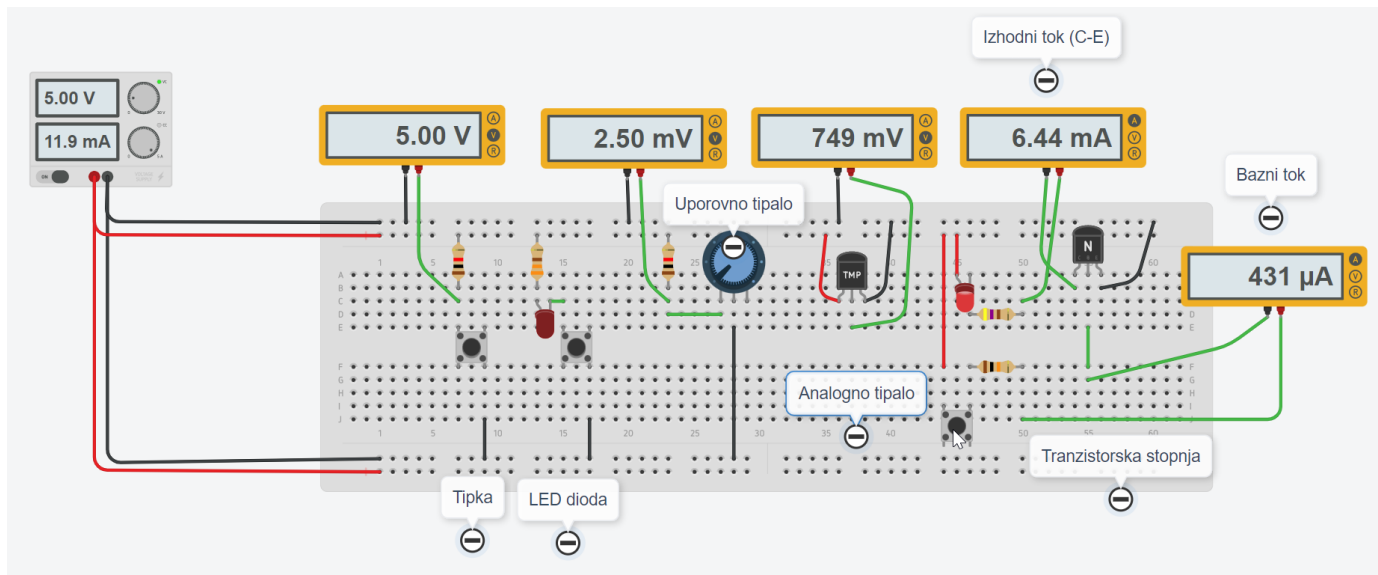
- STM32 CubeIDE + Breadboard

- LED, tipka, potenciometer, uporovna tipala
- PWM brenčoč z melodijami

- Primeri komunikacijskih projektov

- STM32+LIS3DSH
- Arduino

Izhodišče : breadboard vezava TinkerCad



Priključitev na STM32 : 1x analogni, 1x digitalni vhod, 1x digitalni izhod, 4x vgrajene LED diode

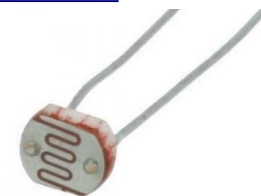
Testno vezje (primer) :

GPIO	Vrsta	Povezava
PA0	User tipka	
PA1	Analogni vhod	Rumena žička
PB4	Dig. Vhod	Zelena žička
PB5	Dig. Izhod - LED	Oranžna žička
PD12-PD15	Dig. Izhodi	vgr. LED diode

Uporovna tipala

LDR – Light Dependent Resistor PGM5337

FOTO UPOR PGM5537 100mW 16-50k $\Omega$  540nm



▶ Electronics Characteristics

Model	Vmax (VDC)	Pmax (mW)	Ambient Temp (°C)	Spectral Peak (nm)	Photo Resistance (10Lx) (k $\Omega$ )	Dark Resistance (M $\Omega$ )min	$\gamma$ min	ResponseTime (ms)	
								Rise	Decay
PGM5506	100	90	-30 ~ +70	540	2 ~ 6	0.15	0.6	30	40
PGM5516	100	90	-30 ~ +70	540	5 ~ 10	0.2	0.6	30	40
PGM5526	150	100	-30 ~ +70	540	8 ~ 20	1.0	0.6	20	30
PGM5537	150	100	-30 ~ +70	540	16 ~ 50	2.0	0.7	20	30



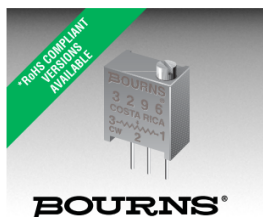
NTC – Termistor NTCC-2K2

UPOR NTC 2K2 5%

NTCC-2K2 SR PASSIVES

NTC thermistor; 2.2k $\Omega$ ; THT; 3900K; -55 ÷ 125° C; 500mW; Ø6.5mm

TrimPot – Trimer Potenciometer TSR-3296Z-104

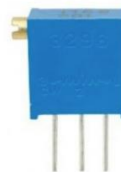


Features

- Multiturn / Cermet / Industrial / Sealed
- 5 terminal styles
- Tape and reel packaging available
- Chevron seal design
- Listed on the QPL for style RJ24 per MIL-R-22097 and RJ24 per High-Rel Mil-R-39035
- Mounting hardware available (H-117P)
- RoHS compliant\* version available
- For trimmer applications/processing guidelines, [click here](#)

3296 - 3/8 " Square Trimpot® Trimming Potentiometer

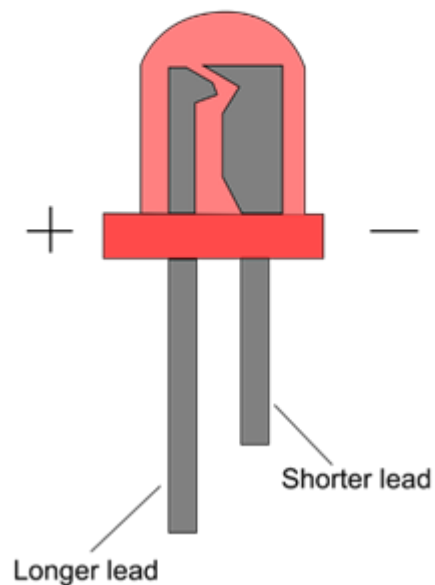
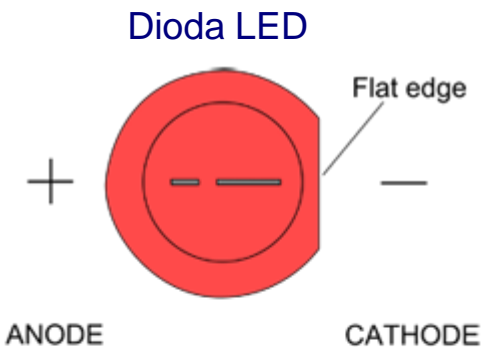
Proizvajalec	Suntan
Številka proizvajalca	TSR-3296Z-104



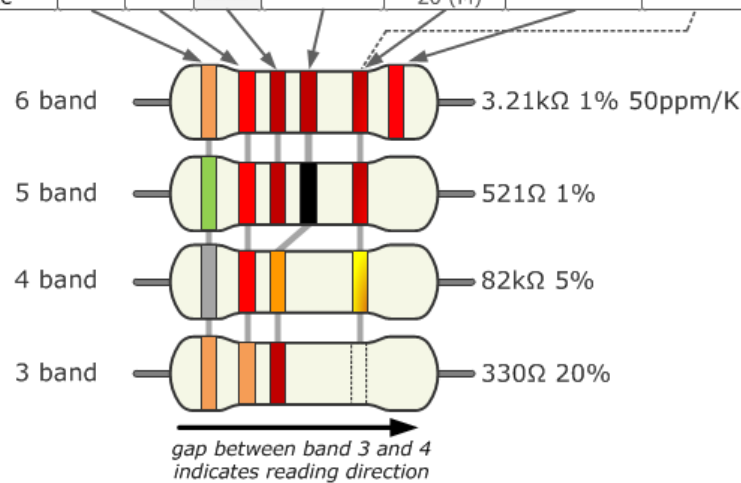
Standard Resistance Table

Resistance (Ohms)	Resistance Code
10	100
20	200
50	500
100	101
200	201
500	501
1,000	102
2,000	202
5,000	502
10,000	103
20,000	203
25,000	253
50,000	503
100,000	104
200,000	204

Elektronske komponente



	Color	Significant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold				x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		



Resistor Color Code Calculator and Chart (4-band, 5-band or 6-band)

Z naslova <<https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>>



# Multimeter EMOS MD-420



Preverjanje povezav

Merjenje upornosti

Merjenje el. Napetosti DC

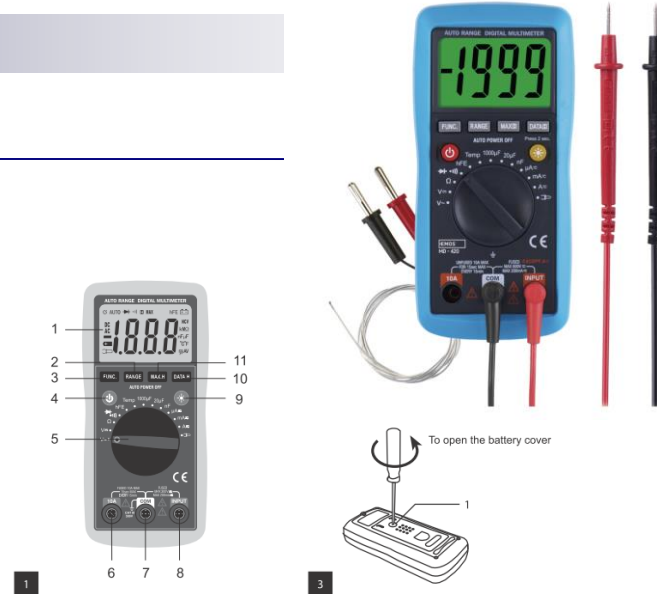
- vzporedna** vezava !!!
- visoka** upornost

Merjenje el. toka

- zaporedna** vezava !!!
- nizka** upornost

### Praktični nasveti :

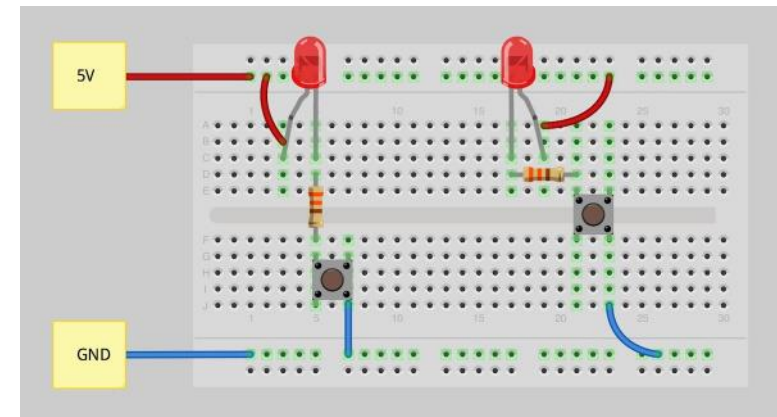
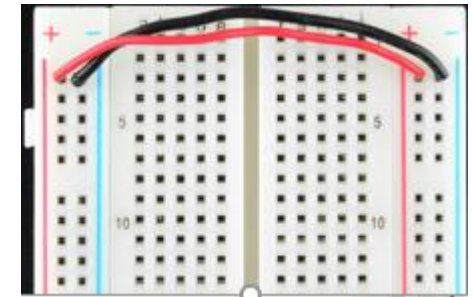
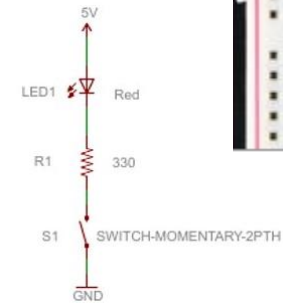
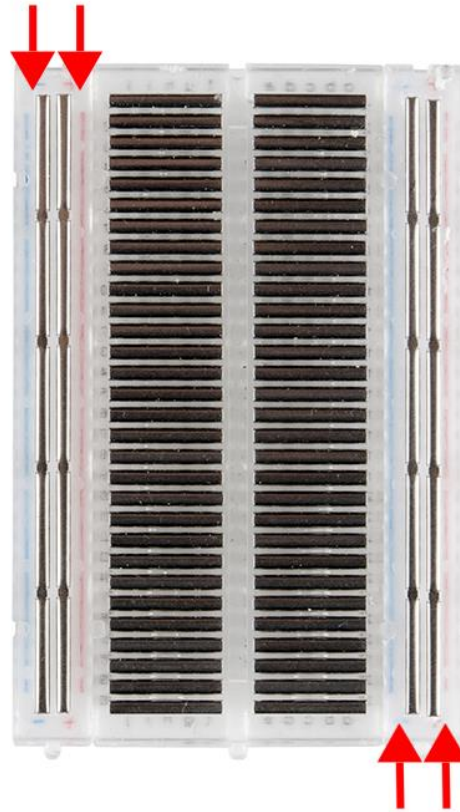
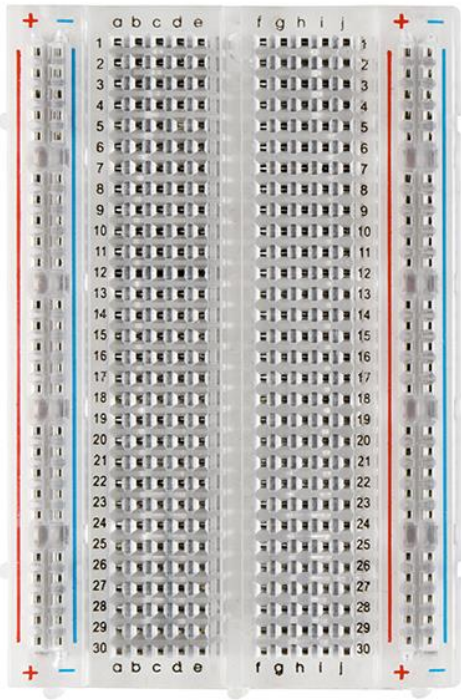
- Večinoma merimo napetost, upornost
- upornost samo izven tokokroga
- pazimo, da ne sklenemo kratkega stika z merilno sondo
- pazimo predvsem na majhne upornosti:
  - Med +V in GND
  - Na izhodih, vhodih mikrokrmilnikov



<https://www.emos-si.si/multimeter-md-420>

# VIN projekt : TinkerCad

## Breadboard vezave



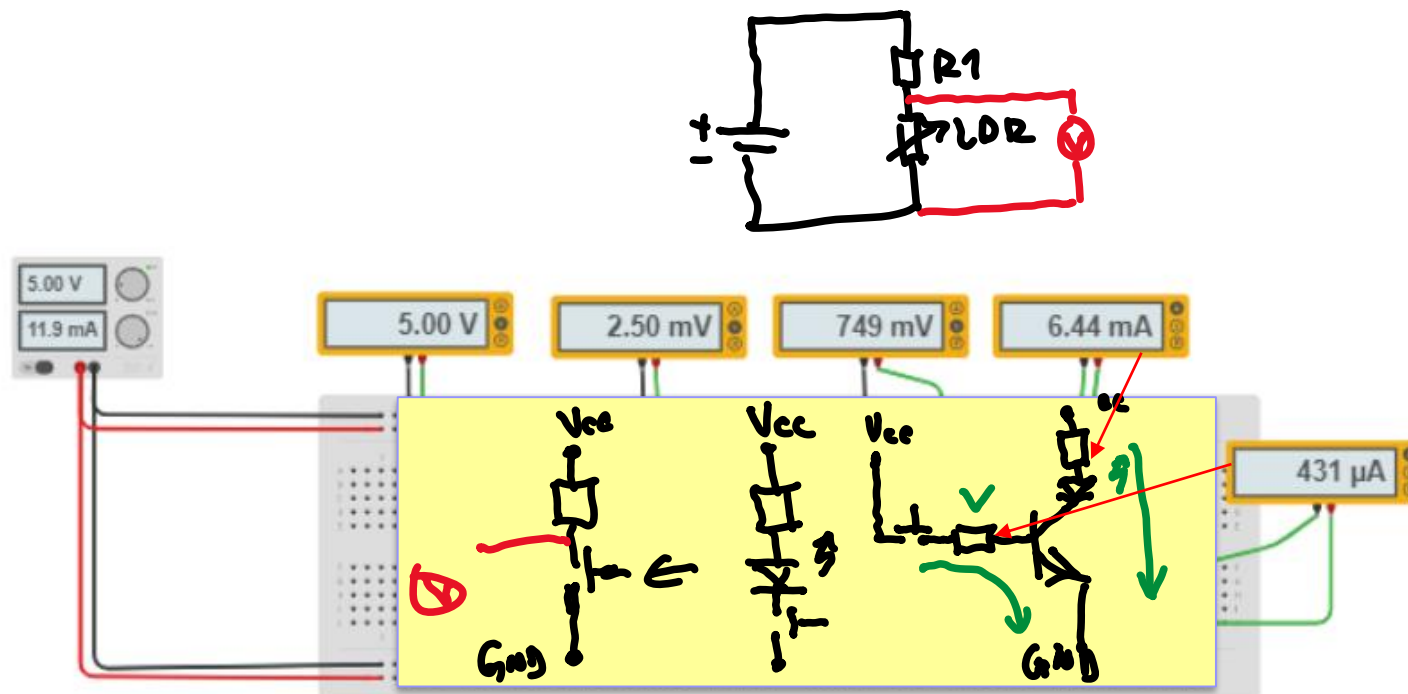
### Viri

- <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





# Breadboard vezava

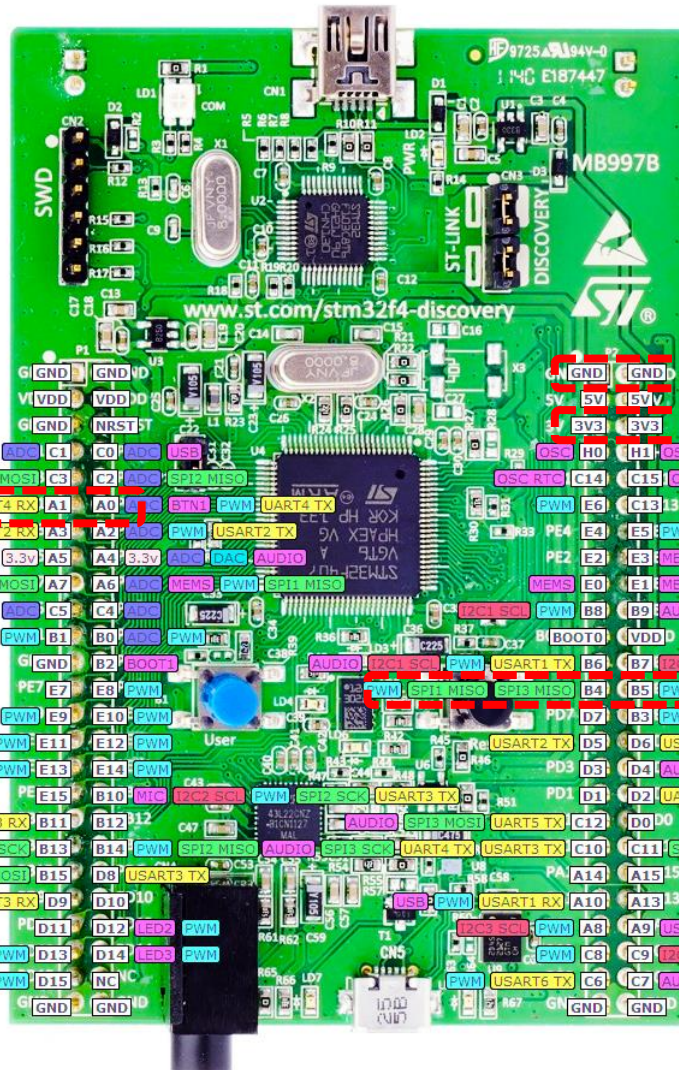
## Priključitev na STM32 : 1x analogni, 1x digitalni vhod, 1x digitalni izhod, 4x vgrajene LED diode

Testno vezje (primer) :

GPIO	Vrsta	Povezava
PA0	User tipka	
PA1	Analogni vhod	Rumena žička
PB4	Dig. Vhod	Zelena žička
PB5	Dig. Izhod - LED	Oranžna žička
PD12-PD15	Dig. Izhodi	vgr. LED diode

P1

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50



P2

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50

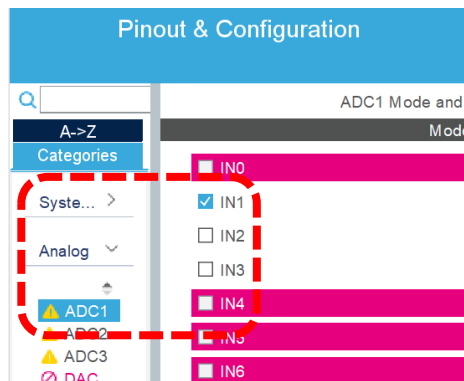


# VIN projekt - VP4: STM32-CubeIDE projekt, breadboard vezave, komunikacije

- Osvežitev
- Breadboard vezava
- STM32 CubeIDE + Breadboard
  - LED, tipka, potenciometer, uporovna tipala
  - PWM brenčoč z melodijami
- Primeri komunikacijskih projektov
  - STM32+LIS3DSH
  - Arduino

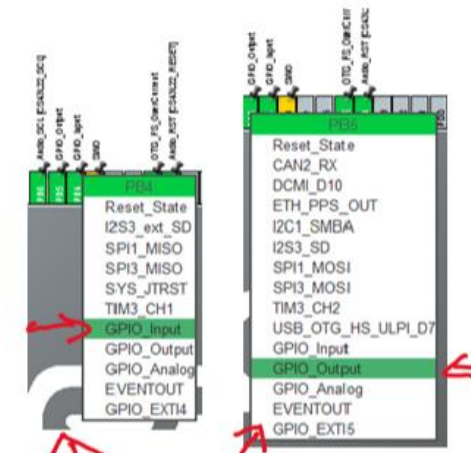
# Osnovni projekt CubeIDE

## Spremembe v projektu :



Testno vezje (primer) :

GPIO	Vrsta	Povezava
PA0	User tipka	
PA1	Analogni vhod	
PB4	Dig. Vhod	
PB5	Dig. Izhod - LED	
PD12-PD15	Dig. Izhodi	vgr. LED diode



## Program : za pošiljanje po USB Virtual COM Port

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
    AnalogValue = HAL_ADC_GetValue(&hadc1);

    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

    HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_5); //External LED on PB5
    KeyState = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4); //External Key on PB4

    snprintf (SendBuffer,BUFSIZE,"Hello [%d]: Key:%d Analog:%d\r\n",Counter++, 1-KeyState, AnalogValue);
    CDC_Transmit_FS(SendBuffer,strlen(SendBuffer));

    /* USER CODE END WHILE */

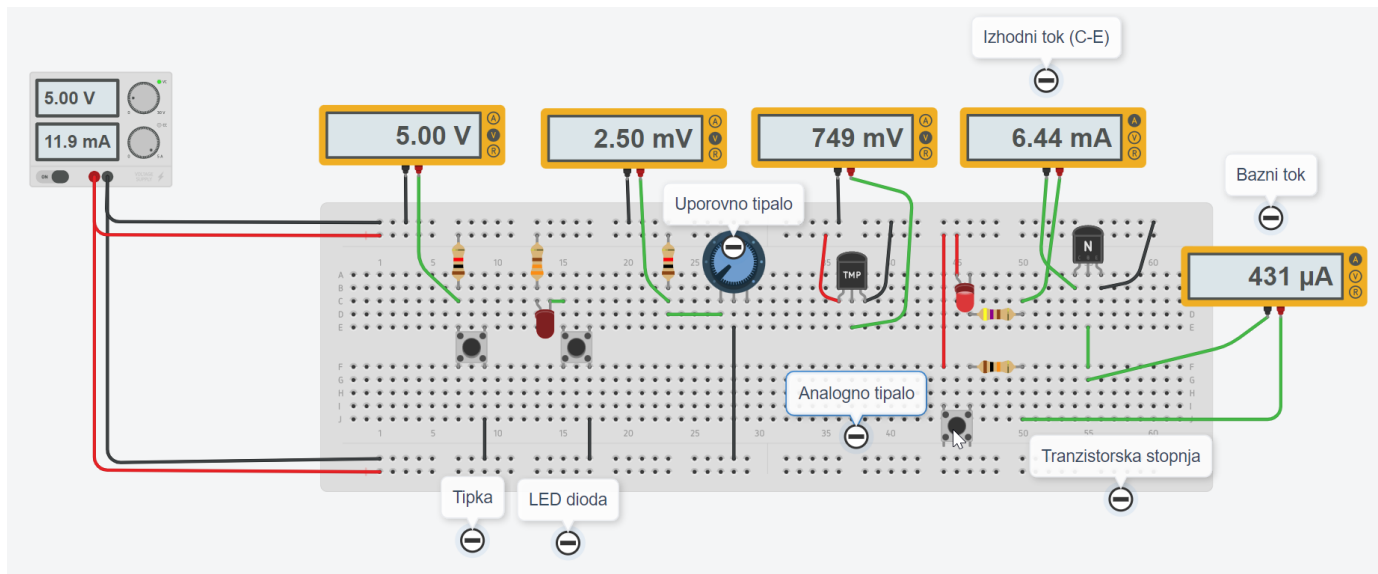
    /* USER CODE BEGIN 3 */
        HAL_Delay(1000);
    }
    /* USER CODE END 3 */
```

# VIN projekt - VP4: STM32-CubeIDE projekt, breadboard vezave, komunikacije

- Osvežitev
- Breadboard vezava
- STM32 CubeIDE + Breadboard
  - LED, tipka, potenciometer, uporovna tipala
  - PWM brenčoč z melodijami
- Primeri komunikacijskih projektov
  - STM32+LIS3DSH
  - Arduino



Izhodišče : breadboard vezava



Priključitev na STM32 : 1x PWM izhod, 1x GND, 4x vgrajene LED diode

Testno vezje (primer) :

GPIO	Vrsta	Povezava
PA15	Brenčac	+
GND	Brenčac	-
PD12-PD15	Dig. Izhodi	vgr. LED diode

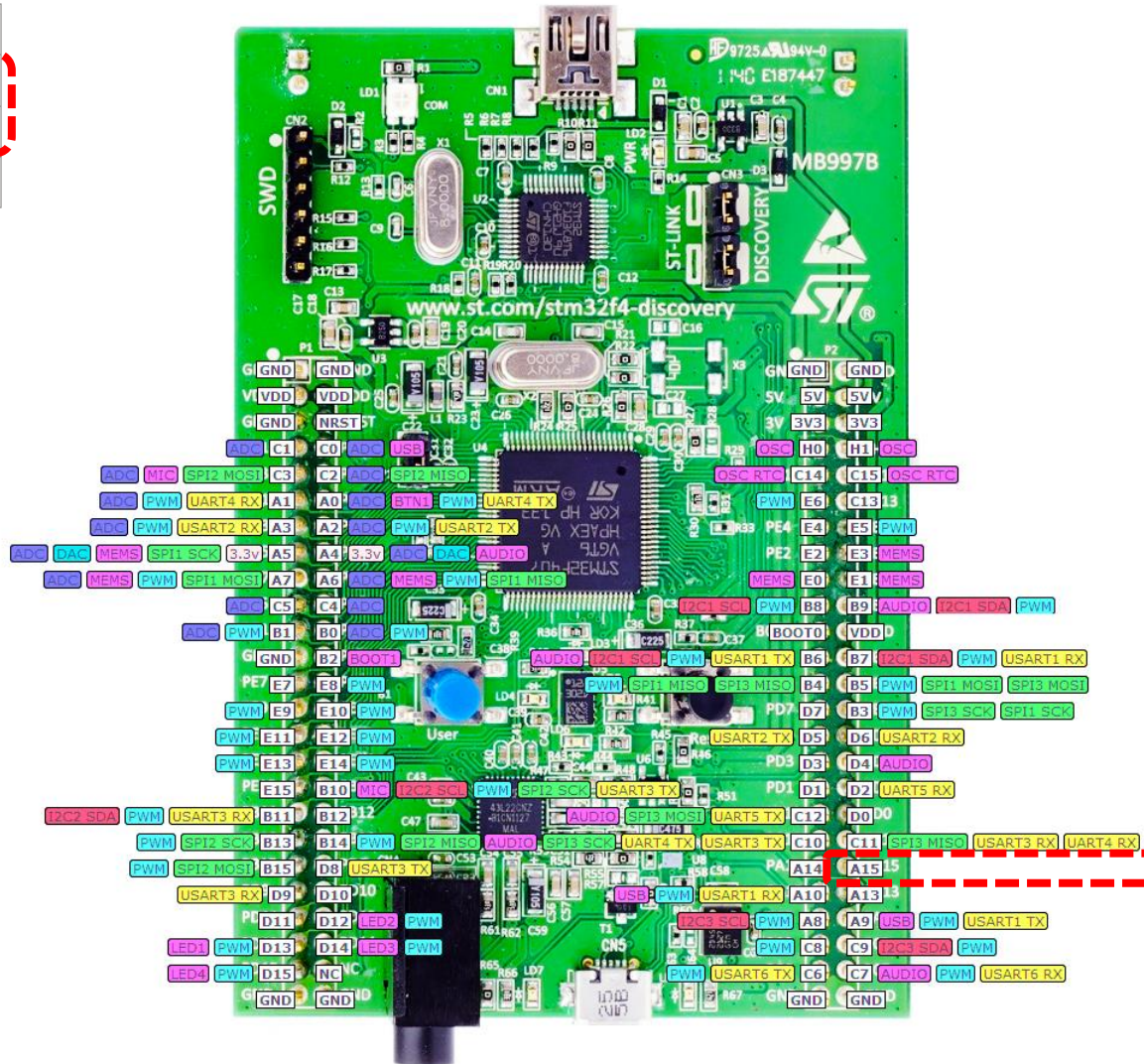
## Priključitev na STM32 : 1x PWM izhod, 1x GND, 4x vgrajene LED diode

Testno vezje (primer) :

GPIO	Vrsta	Povezava
PA15	Brenčac	+
GND	Brenčac	-
PD12-PD15	Dig. Izhodi	vgr. LED diode

### P1

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50

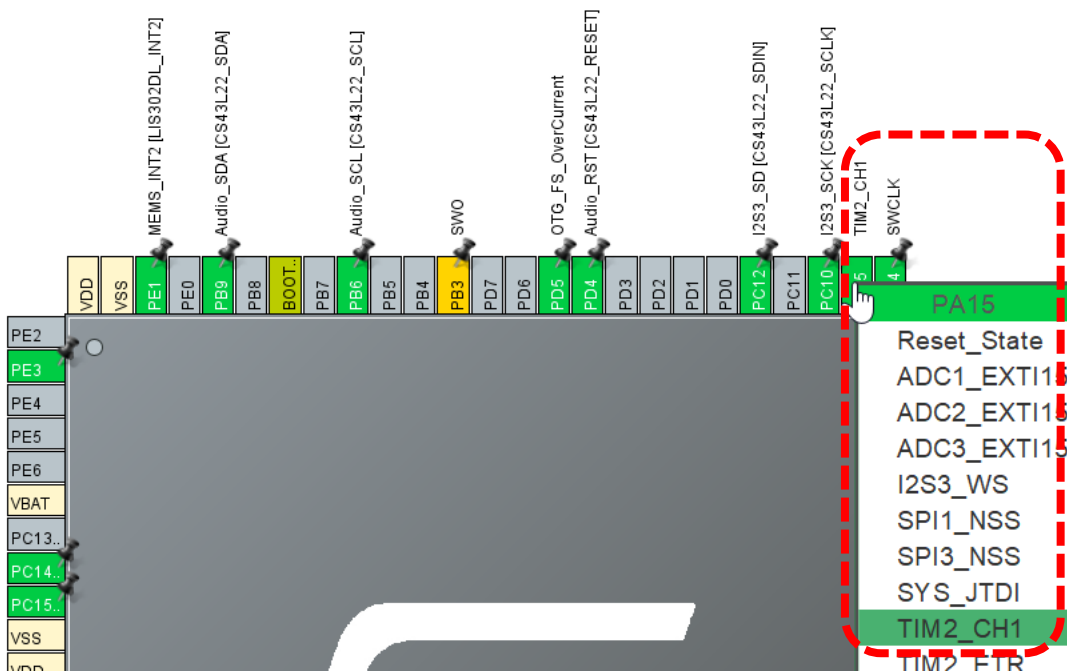


### P2

- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 11 12
- 13 14
- 15 16
- 17 18
- 19 20
- 21 22
- 23 24
- 25 26
- 27 28
- 29 30
- 31 32
- 33 34
- 35 36
- 37 38
- 39 40
- 41 42
- 43 44
- 45 46
- 47 48
- 49 50

## Spremembe v projektu :

The screenshot shows the 'Pinout & Configuration' window in STM32CubeIDE. The 'TIM2 Mode and Configuration' section is active. In the 'Mode' section, 'Channel1' is set to 'PWM Generation CH1'. In the 'Configuration' section, 'Counter Settings' are expanded, showing 'Prescaler (PSC - 16 bits...)' set to '83'. The left sidebar shows 'TIM2' selected under 'Timers'.



## Program : za pošiljanje po USB Virtual COM Port

```
/* Inicializacija*/
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);

while (1) {

melodyCount = sizeof(melodySizes)/ sizeof(uint32_t);

...

ARR_period = (int)(1000000/NoteFreq); //Already prescaled to 1 MHz
setPWM(htim2, TIM_CHANNEL_1, ARR_period, ARR_period/2);

Delaymsecs = noteDurations[melodyIndex][noteIndex] * melodySlowfactor[melodyIndex];

snprintf (SendBuffer, BUFSIZE, "Melody[%d], Note # %d F=%d Hz Duration: %d ms | ARR=%d
CCR1=%d\r\n", melodyIndex, noteIndex, melody[melodyIndex][noteIndex], Delaymsecs, htim2.Instance->ARR, htim2.Instance->CCR1);
CDC_Transmit_FS(SendBuffer, strlen(SendBuffer));

HAL_Delay(Delaymsecs);
}

HAL_Delay(2000);

}
```

melody.h

Melodije in notni zapis : [https://github.com/bogusz2233/Buzzer\\_music\\_stm32/blob/master/Inc/melody.h](https://github.com/bogusz2233/Buzzer_music_stm32/blob/master/Inc/melody.h)

<https://circuitdigest.com/microcontroller-projects/playing-melodies-on-piezo-buzzer-using-arduino-tone-function>

```
void setPWM(TIM_HandleTypeDef timer, uint32_t channel, uint16_t period, uint16_t pulse)
{
    HAL_TIM_PWM_Stop(&timer, channel); // stop generation of pwm
    TIM_OC_InitTypeDef sConfigOC;
    timer.Init.Period = period; // set the period duration
    HAL_TIM_PWM_Init(&timer); // reinitialise with new period value
    sConfigOC.OCMode = TIM_OC_MODE_PWM1;
    sConfigOC.Pulse = pulse; // set the pulse duration
    sConfigOC.OCpolarity = TIM_OC_POLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OC_FAST_DISABLE;
    HAL_TIM_PWM_ConfigChannel(&timer, &sConfigOC, channel);
    HAL_TIM_PWM_Start(&timer, channel); // start pwm generation
}
```

# VIN projekt - VP4: STM32-CubeIDE projekt, breadboard vezave, komunikacije

- Osvežitev
- Breadboard vezava
- STM32 CubeIDE + Breadboard
  - LED, tipka, potenciometer, uporovna tipala
  - PWM brenčač z melodijami

## ■ Primeri komunikacijskih projektov

- STM32+LIS3DSH

- Arduino



# Serijske komunikacije – I2C

## ■ How to use I2C in Arduino: Communication between two Arduino Boards

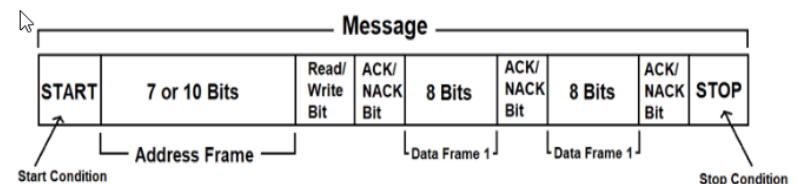
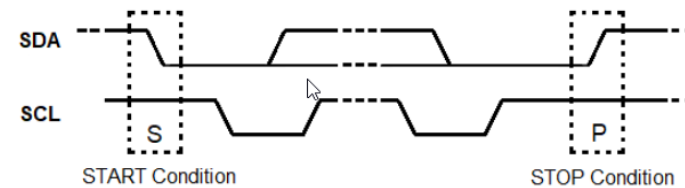
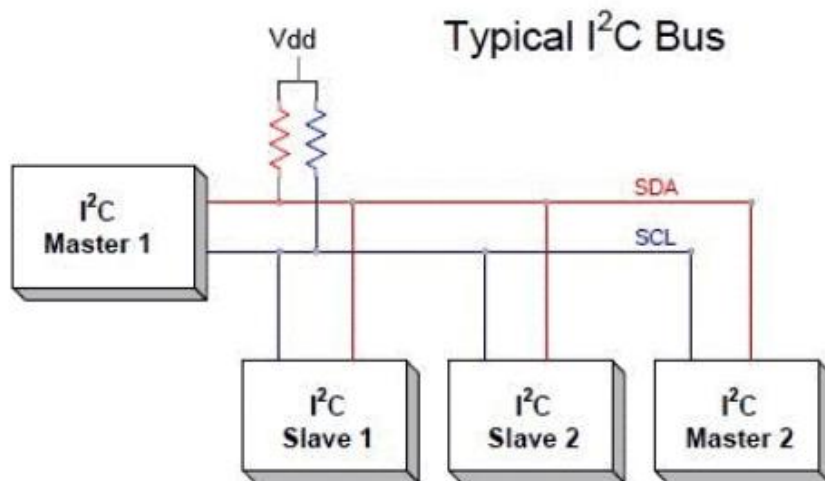
- Z naslova <<https://circuitdigest.com/microcontroller-projects/arduino-i2c-tutorial-communication-between-two-arduino>>

## ■ Communication Between Two Arduinos (I2C)

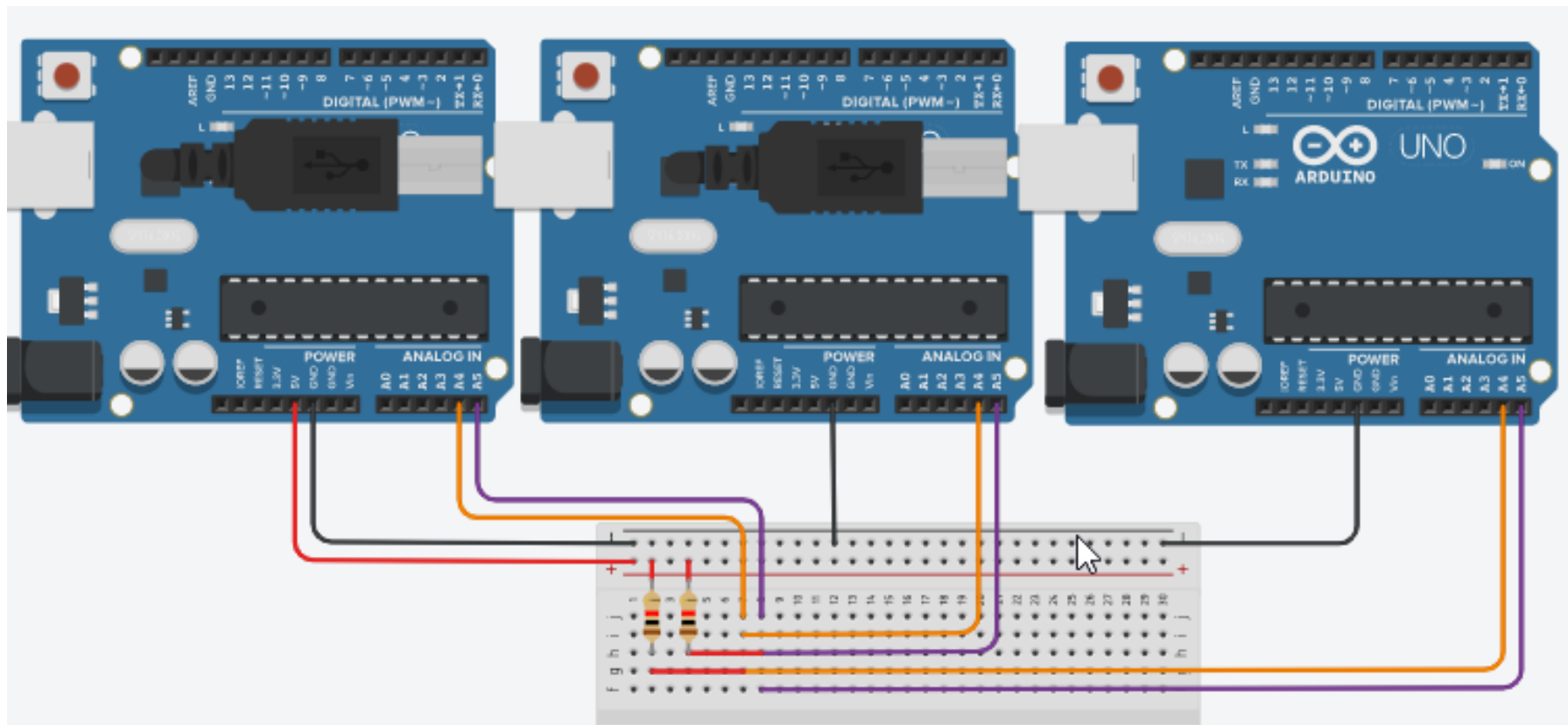
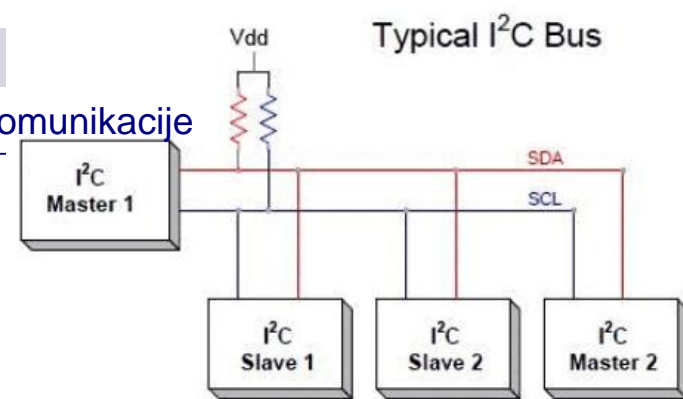
- Z naslova <<https://www.instructables.com/id/Communication-Between-Two-Arduinos-I2C/>>

## ■ Master Slave I2C Connection

- Z naslova <[https://www.hackster.io/PIYUSH\\_K\\_SINGH/master-slave-i2c-connection-f1aa53](https://www.hackster.io/PIYUSH_K_SINGH/master-slave-i2c-connection-f1aa53)>



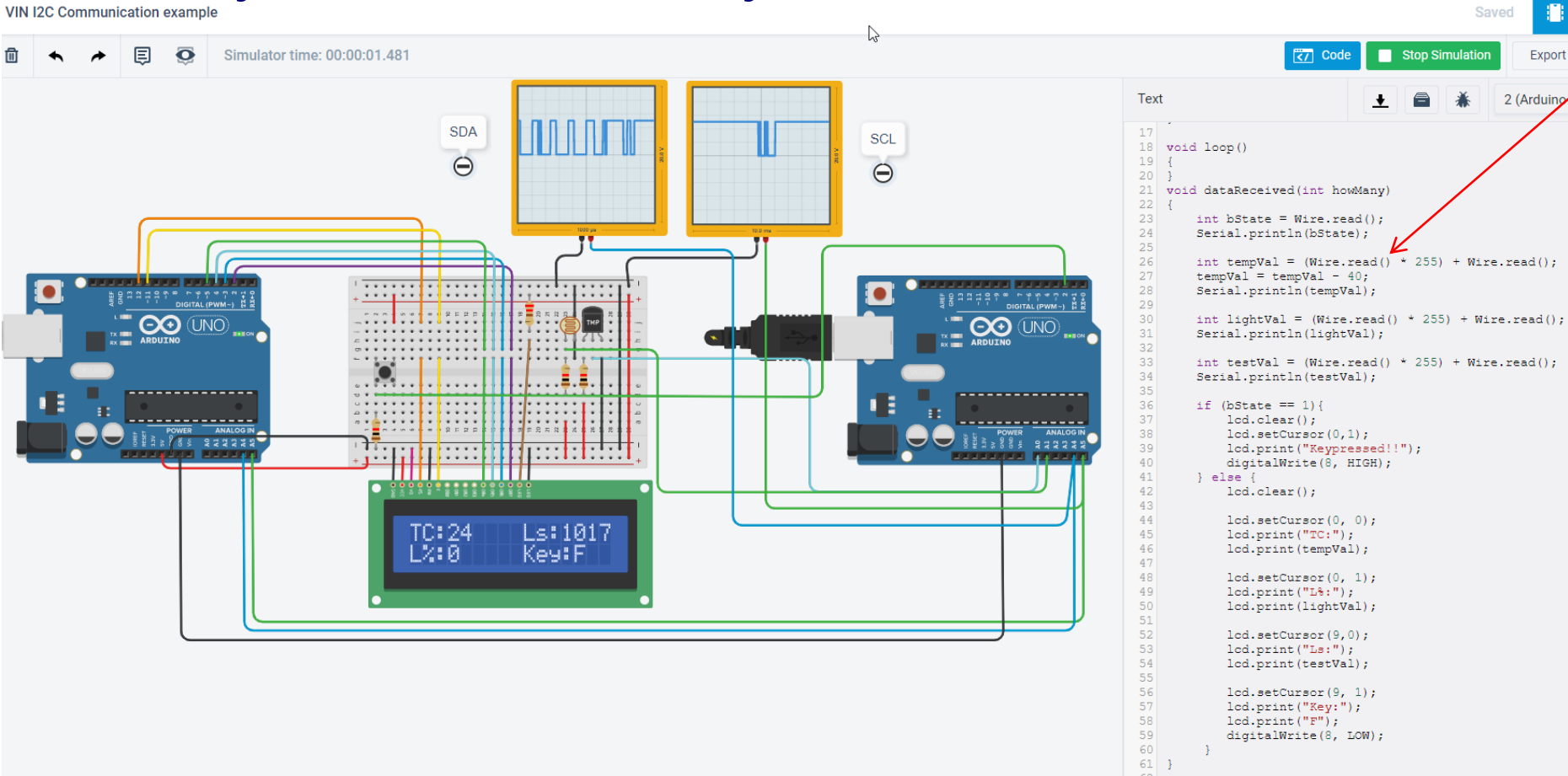
# Serijske komunikacije – I2C



# Serijske komunikacije – I2C - Primer

VIN I2C Communication example

Simulator time: 00:00:01.481



SDA

SCL

```
17
18 void loop()
19 {
20 }
21 void dataReceived(int howMany)
22 {
23   int bState = Wire.read();
24   Serial.println(bState);
25
26   int tempVal = (Wire.read() * 255) + Wire.read();
27   tempVal = tempVal - 40;
28   Serial.println(tempVal);
29
30   int lightVal = (Wire.read() * 255) + Wire.read();
31   Serial.println(lightVal);
32
33   int testVal = (Wire.read() * 255) + Wire.read();
34   Serial.println(testVal);
35
36   if (bState == 1){
37     lcd.clear();
38     lcd.setCursor(0,1);
39     lcd.print("KeyPressed!!");
40     digitalWrite(8, HIGH);
41   } else {
42     lcd.clear();
43
44     lcd.setCursor(0, 0);
45     lcd.print("TC:");
46     lcd.print(tempVal);
47
48     lcd.setCursor(0, 1);
49     lcd.print("Ls:");
50     lcd.print(lightVal);
51
52     lcd.setCursor(9, 0);
53     lcd.print("Ls:");
54     lcd.print(testVal);
55
56     lcd.setCursor(9, 1);
57     lcd.print("Key:");
58     lcd.print("F");
59     digitalWrite(8, LOW);
60   }
61 }
62
```

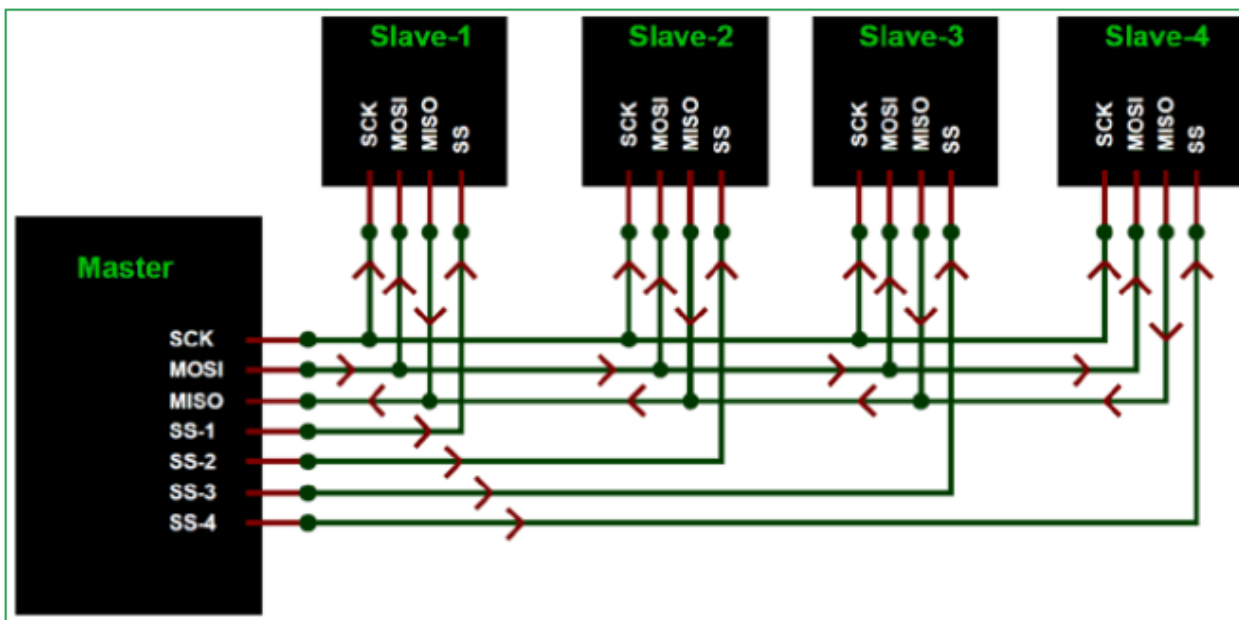
<https://www.tinkercad.com/things/14ImcXzZtzb-vin-i2c-communication-example>

# Serijske komunikacije – SPI

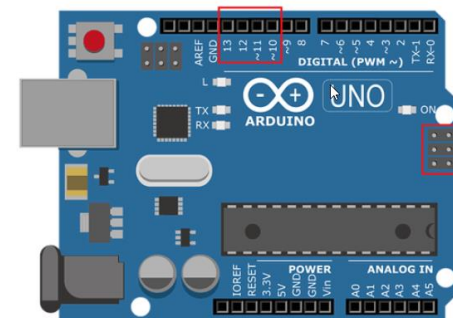
## ■ How to use SPI in Arduino: Communication between two Arduino Boards

- Z naslova <<https://circuitdigest.com/microcontroller-projects/arduino-spi-communication-tutorial>>

### SPI Master with Multiple Slaves



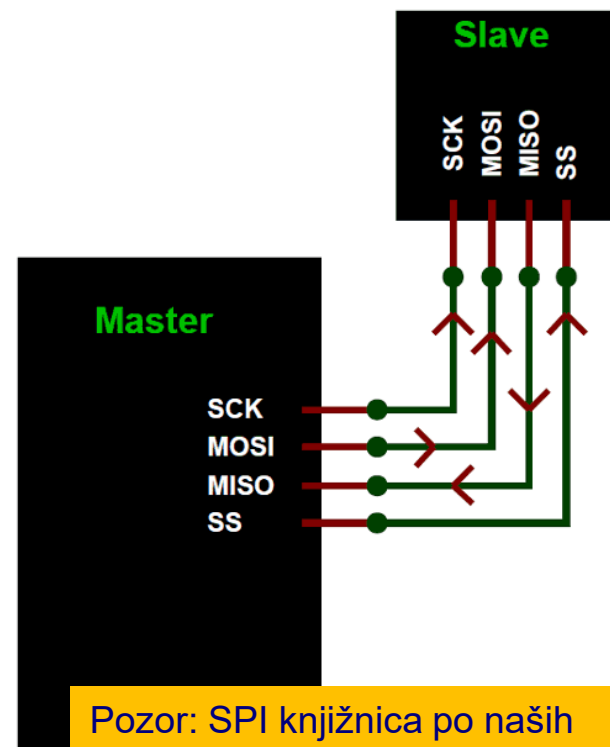
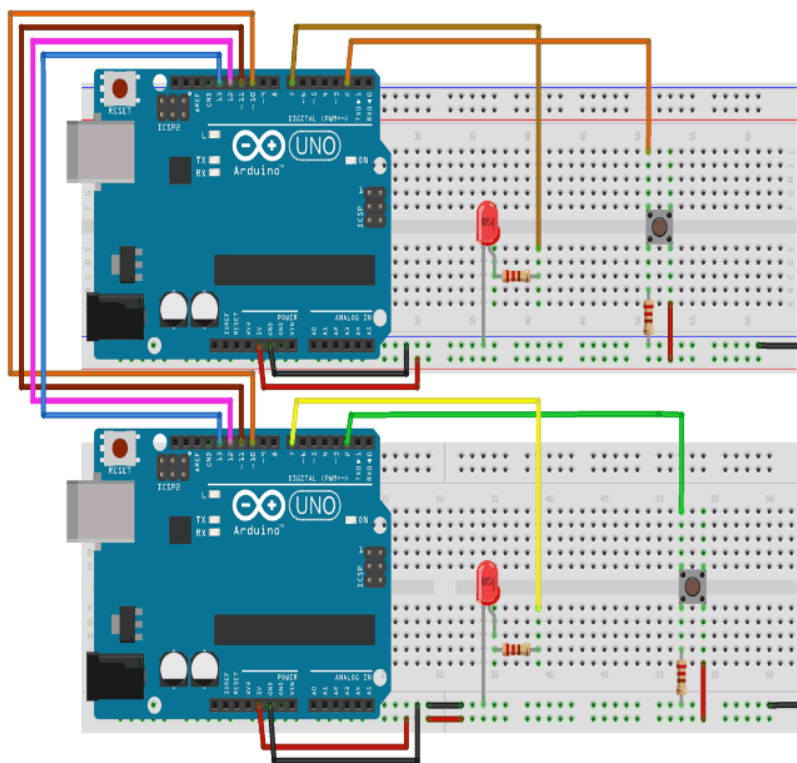
SPI Line	Pin in Arduino
MOSI	11 or ICSP-4
MISO	12 or ICSP-1
SCK	13 or ICSP-3
SS	10



# Serijske komunikacije – SPI

## ■ How to use SPI in Arduino: Communication between two Arduino Boards

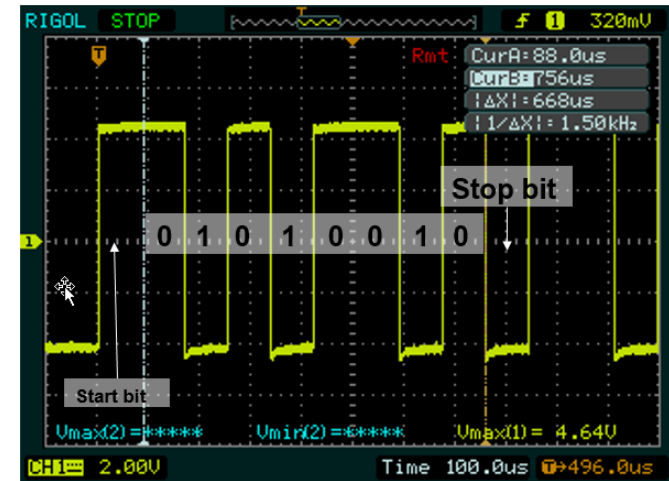
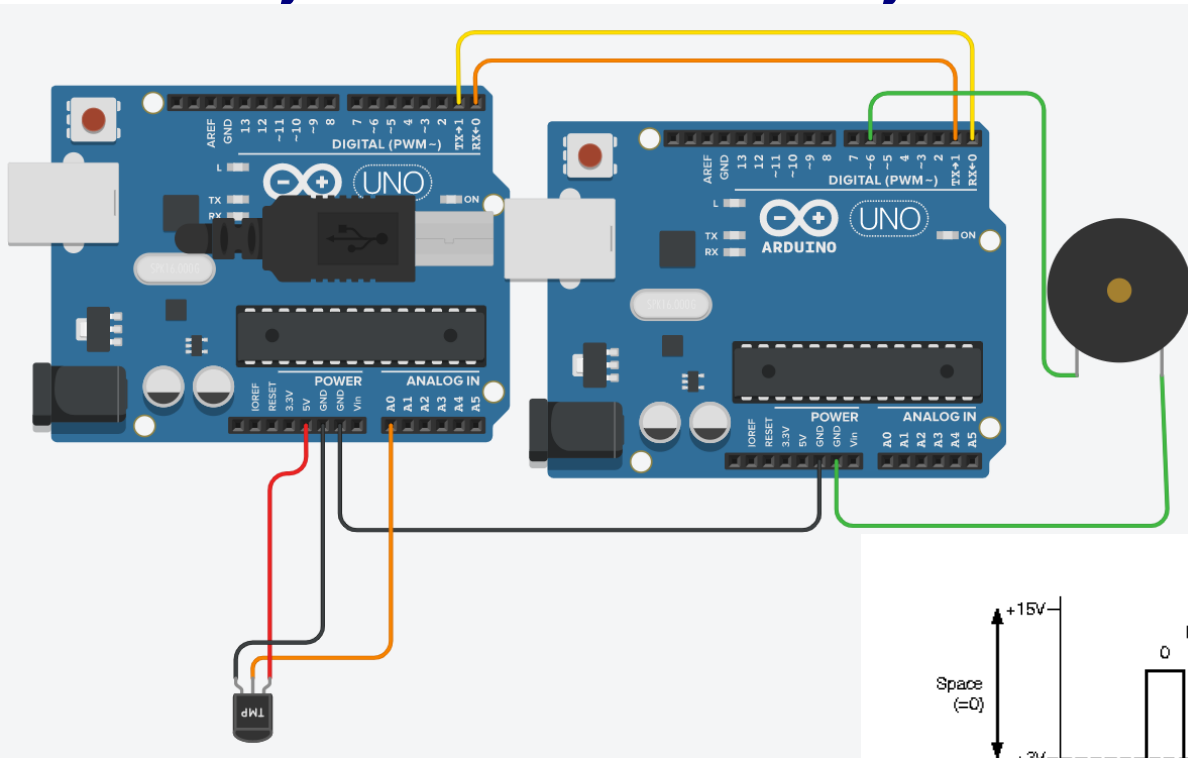
- Z naslova <<https://circuitdigest.com/microcontroller-projects/arduino-spi-communication-tutorial>>



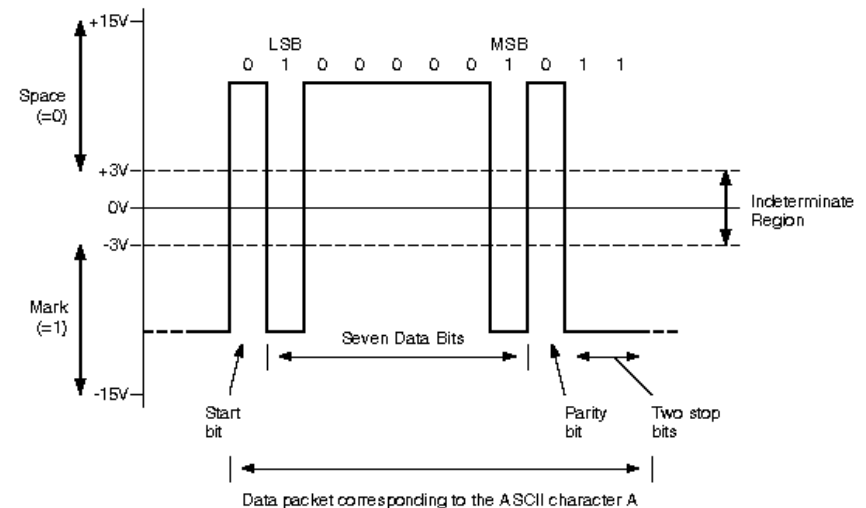
Pozor: SPI knjižnica po naših informacijah v TinkerCadu ne deluje, je pa mogoče protokol implementirati v lastni kodi.



# Serijske komunikacije – UART



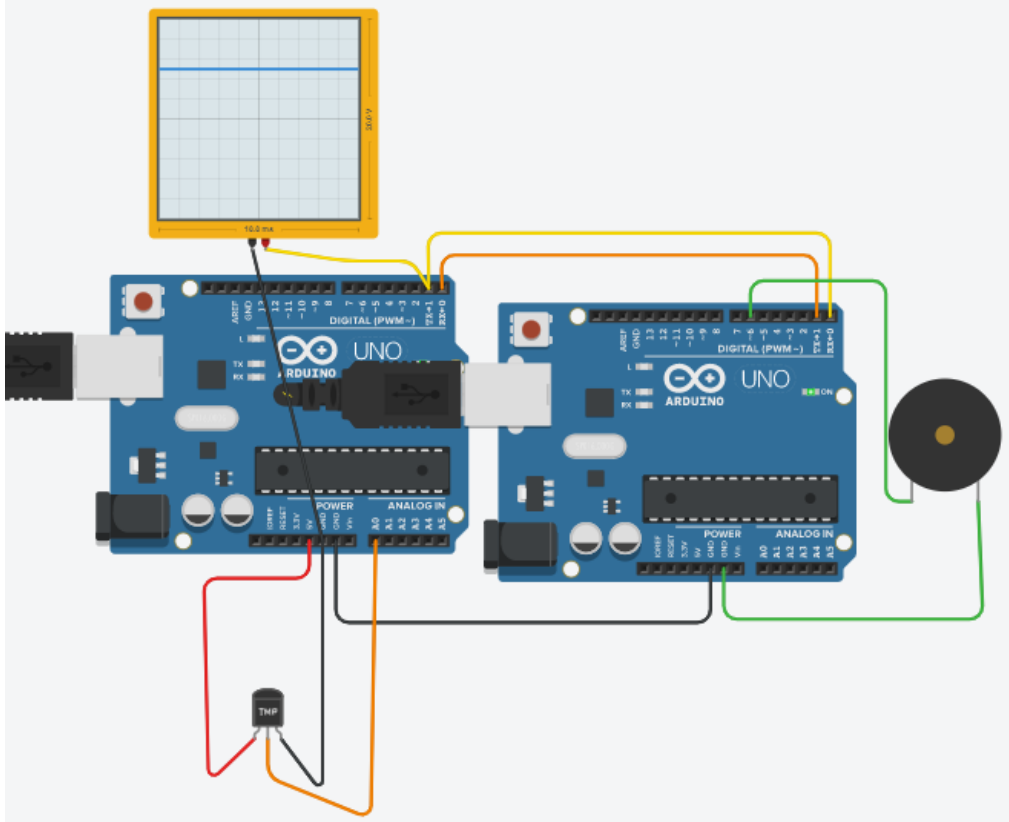
Pozor :  
Po tej povezavi poteka UART komunikacija na TTL napetostnih nivojih  
Ni namreč povezave na zunanji RS232 pretvornik !



# Serijske komunikacije – UART - Primer

01:12.023

Code



```
Text
1
2 int val;
3 int T;
4 void setup() {
5
6   pinMode(A0, INPUT);
7   Serial.begin(9600); // Begin the Serial at 9600
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
```

<https://www.tinkercad.com/things/8TvZHTxb8jv-vin-uart-communication-demo>