

Εκπαίδευση STEAM

Arduino 3^ο μέρος

ΚΑΜΟΥΤΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Ακαδημαϊκός Υπότροφος, Υποψήφιος Διδάκτορας

Τμήμα Μηχανικών Παραγωγής και Διοίκησης, ΔΙΠΑΕ

Επικοινωνία: kamoutsis.k@gmail.com

Ιστοσελίδες για δημιουργία project

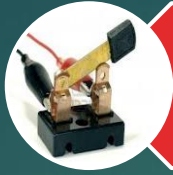
<https://www.arduino.cc/>

<http://blog.ardublock.com/>

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

<https://fritzing.org/>

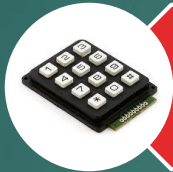
6 ΣΗΜΑΝΤΙΚΕΣ ΕΝΝΟΙΕΣ



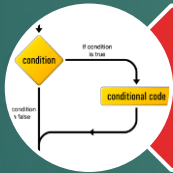
digitalWrite()



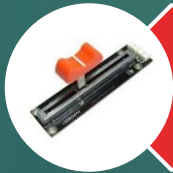
analogWrite()



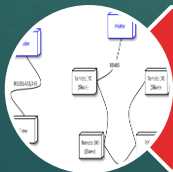
digitalRead()



if() statements / Boolean



analogRead()



Serial communication

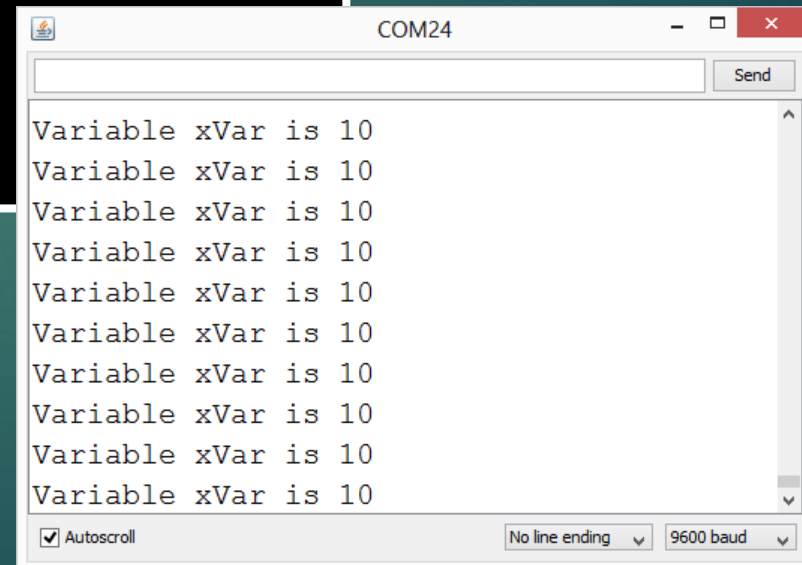
Analog Sensors

Παραδείγματα:

| Sensors | Μετρούμενο μέγεθος |
|---------------|--------------------|
| Mic | Ένταση ήχου |
| Photoresistor | Ένταση φωτός |
| Potentiometer | Θέση δρομέα |
| Temp Sensor | Θερμοκρασία |
| Flex Sensor | κάμψη |
| Accelerometer | επιτάχυνση |

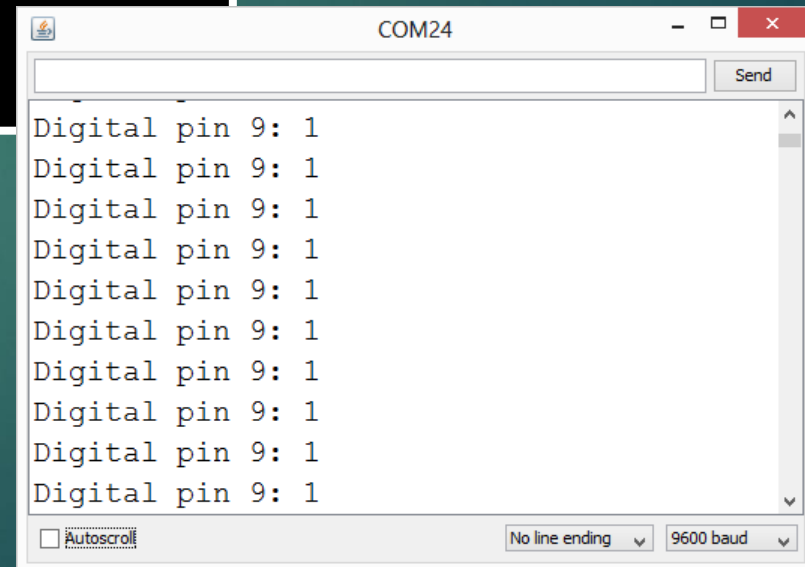
Serial Communication: Serial Debugging - Αποσφαλμάτωση

```
void loop()  
{  
    int xVar = 10;  
    Serial.print("Variable xVar is ");  
    Serial.println(xVar);  
}
```



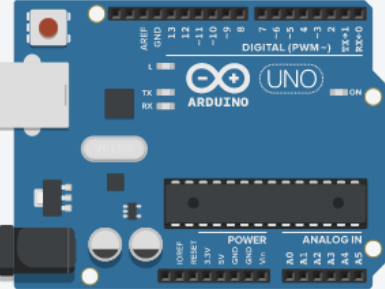
Serial Communication: Serial Troubleshooting - έλεγχος

```
void loop ( )  
{  
  Serial.print ("Digital pin 9: ");  
  Serial.println (digitalRead(9));  
}
```



Παράδειγμα 1: Serial Print

9



The image shows the Arduino IDE interface for an Arduino Uno R3. On the left is a photograph of the board. The main workspace is divided into three sections: a block palette, a block canvas, and a code editor.

Block Palette: Shows categories: Output (blue), Input (purple), Notation (grey), Control (orange), Math (green), and Variables (pink). Available blocks include: set built-in LED to HIGH, set pin 0 to HIGH, set pin 3 to 0, rotate servo on pin 0 to 0 degrees, and play speaker on pin 0 with tone 60.

Block Canvas: Contains a sequence of blocks: three 'print to serial monitor' blocks with text 'Hands on', 'Learning', and 'is Fun!!!' (each with 'without' and 'newline' options), followed by a 'wait 2 secs' block.

Code Editor: Shows the following C++ code for '1 (Arduino Uno R3)':

```
1 void setup()
2 {
3   Serial.begin(9600);
4 }
5
6
7 void loop()
8 {
9   Serial.print("Hands on ");
10  Serial.print("Learning ");
11  Serial.println("is Fun!!!");
12  delay(2000); // Wait for 2000 millisecond(s)
13 }
```

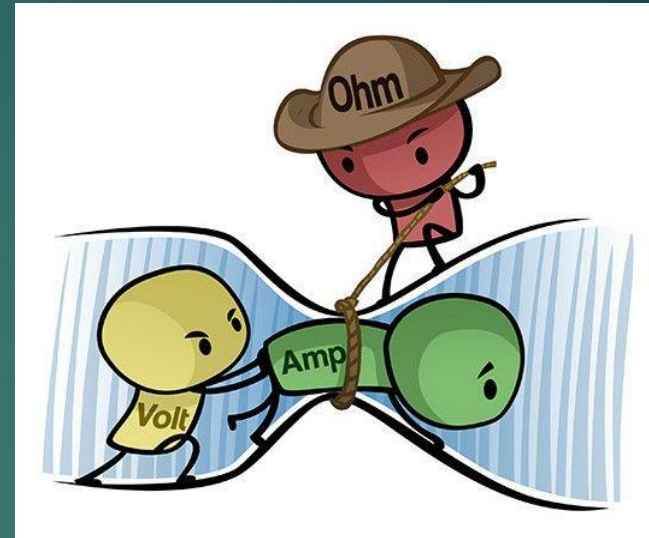
Serial Monitor: Displays the output of the program: 'Hands on Learning is Fun!!!' repeated six times.

At the bottom right, there are 'Send', 'Clear', and a waveform icon button.

Νόμος του Ohm

$V = I \times R$ $I = \frac{V}{R}$ $R = \frac{V}{I}$

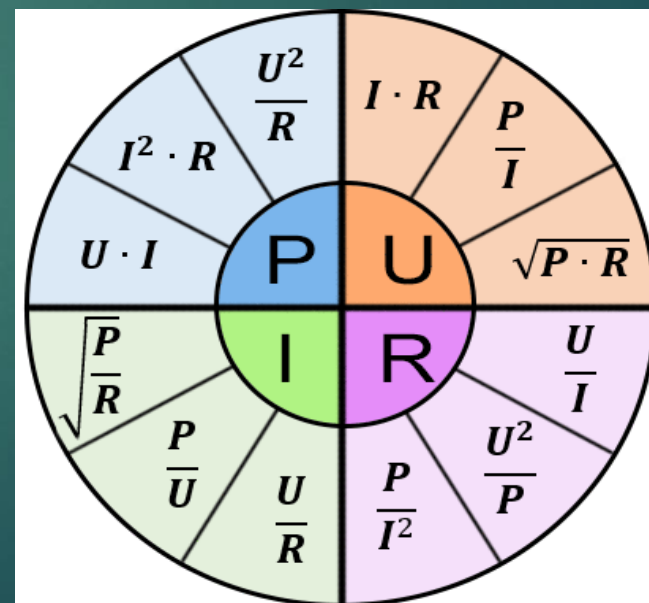
Νόμος του Ohm - Διαδραστικό περιβάλλον


$$R = \frac{\rho L}{A}$$

| | | |
|---------------------------|---------------|-----------------------|
| ρ resistivity | L length | A area |
| 0.50 Ωcm | 10.00 cm | 7.50 cm^2 |

resistance = 0.67 ohm

Αντίσταση σε καλώδιο - Διαδραστικό περιβάλλον



Χρωματικός κώδικας αντιστάσεων

ΚΩΔΙΚΑΣ ΧΡΩΜΑΤΩΝ ΑΝΤΙΣΤΑΣΕΩΝ

| | 1ο | 2ο | 3ο | 4ο |
|----------|------|----|-------------|------------|
| | ==== | 0 | x 1 | 10 % ΑΣΗΜΙ |
| ΚΑΦΕ | 1 | 1 | x 10 | 5 % ΧΡΥΣΟ |
| ΚΟΚΚΙΝΟ | 2 | 2 | x 100 | |
| ΠΟΡΤΟΚΑΛ | 3 | 3 | x 1.000 | |
| ΚΙΤΡΙΝΟ | 4 | 4 | x 10.000 | |
| ΠΡΑΣΙΝΟ | 5 | 5 | x 100.000 | |
| ΜΠΛΕ | 6 | 6 | x 1.000.000 | |
| ΜΩΒ | 7 | 7 | : 10 | |
| ΓΚΡΙ | 8 | 8 | ΧΡΥΣΟ | |
| ΑΣΠΡΟ | 9 | 9 | | |

1ο ΨΗΦΙΟ

2ο ΨΗΦΙΟ

3ο ΨΗΦΙΟ
(Πολλαπλασιαστής)

4ο ΨΗΦΙΟ
(Ανοχή)

4-Band code

47k Ω ±5%

| 1st Digit | 2nd Digit | 3rd Digit | Multiplier | Tolerance |
|-----------|-----------|-----------|-------------|------------|
| 0 | 0 | 0 | 1 | |
| 1 | 1 | 1 | 10 | 1% |
| 2 | 2 | 2 | 100 | 2% |
| 3 | 3 | 3 | 1000 | |
| 4 | 4 | 4 | 10000 | |
| 5 | 5 | 5 | 100000 | |
| 6 | 6 | 6 | 1000000 | |
| 7 | 7 | 7 | | |
| 8 | 8 | 8 | 0.1 Gold | 5% Gold |
| 9 | 9 | 9 | 0.01 Silver | 10% Silver |

5-Band code

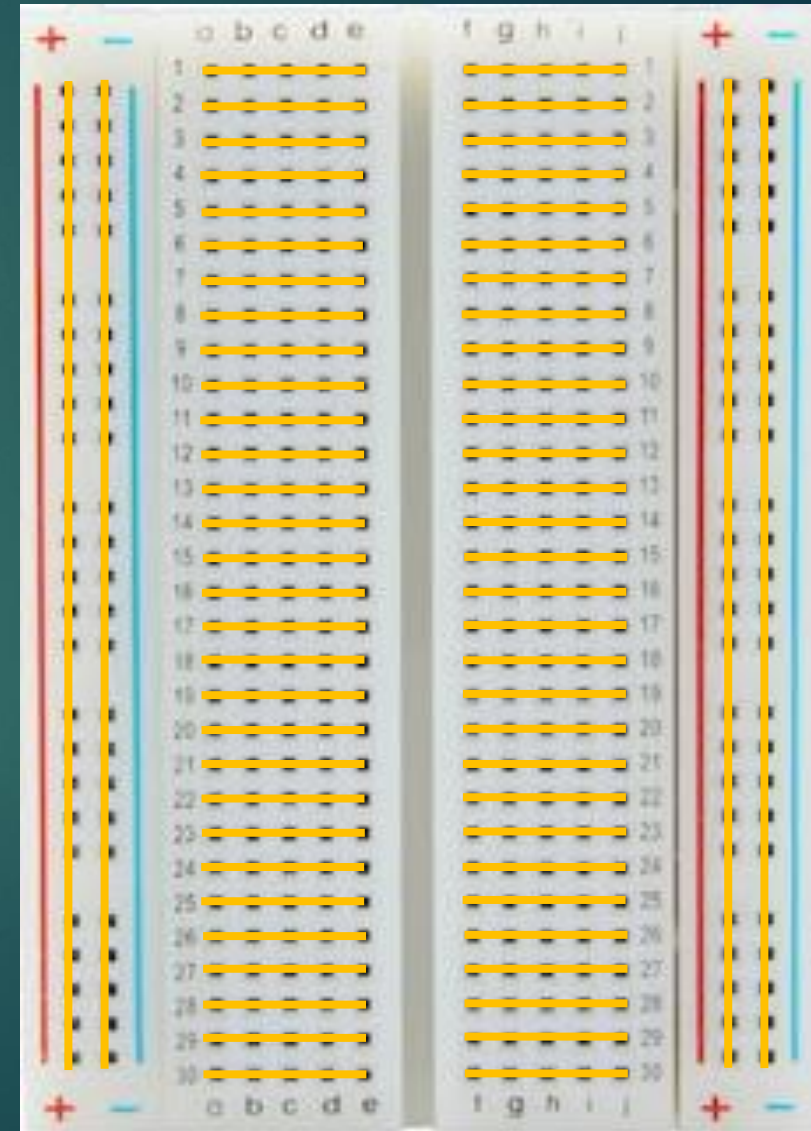
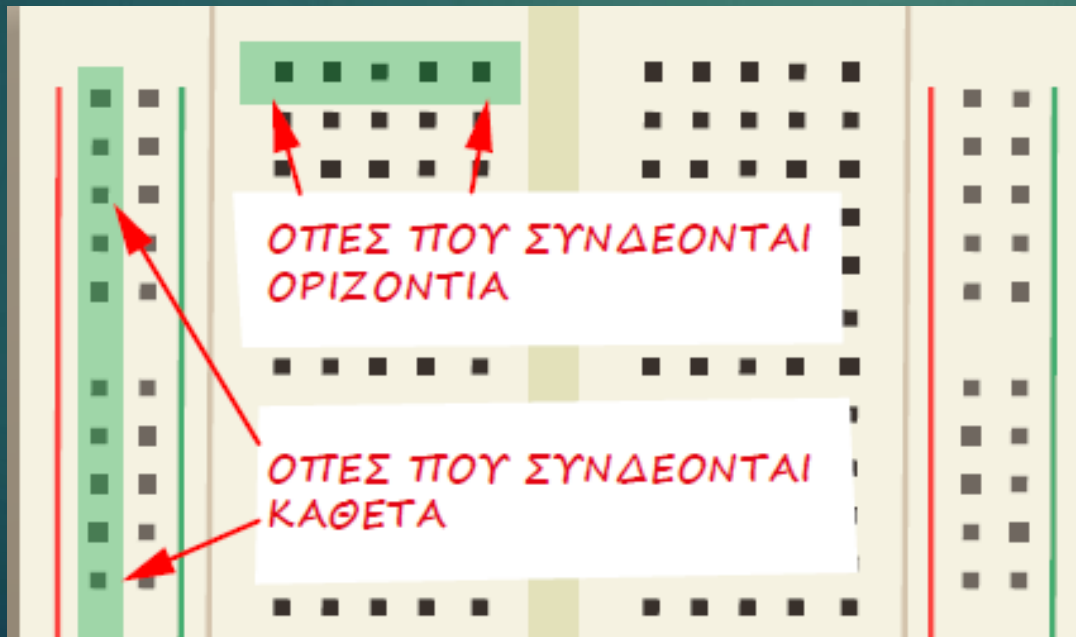
35k Ω ±1%

Πλακέτα δοκιμών (2)

13

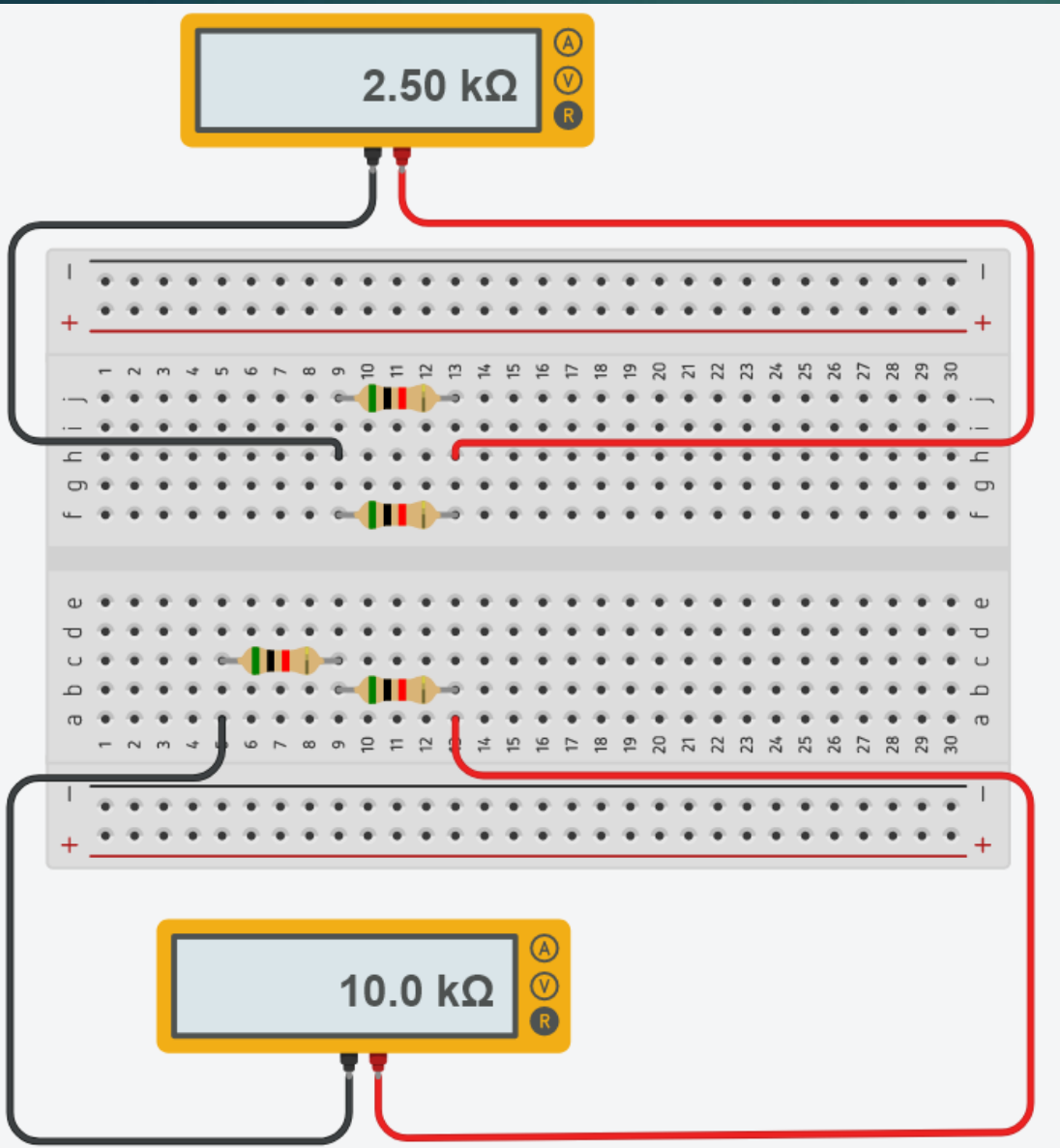
Πλακέτα δοκιμών (raster) – Solderless Breadboard

- Κάθε οριζόντια γραμμή 5 οπών είναι συνδεδεμένη (βραχυκυκλωμένη).
- Οι κατακόρυφες στήλες τροφοδοσίας είναι συνδεδεμένες (βραχυκυκλωμένες).



Σύνδεση αντιστάσεων

14



Πρόγραμμα προσομοίωσης
www.tinkercad.com

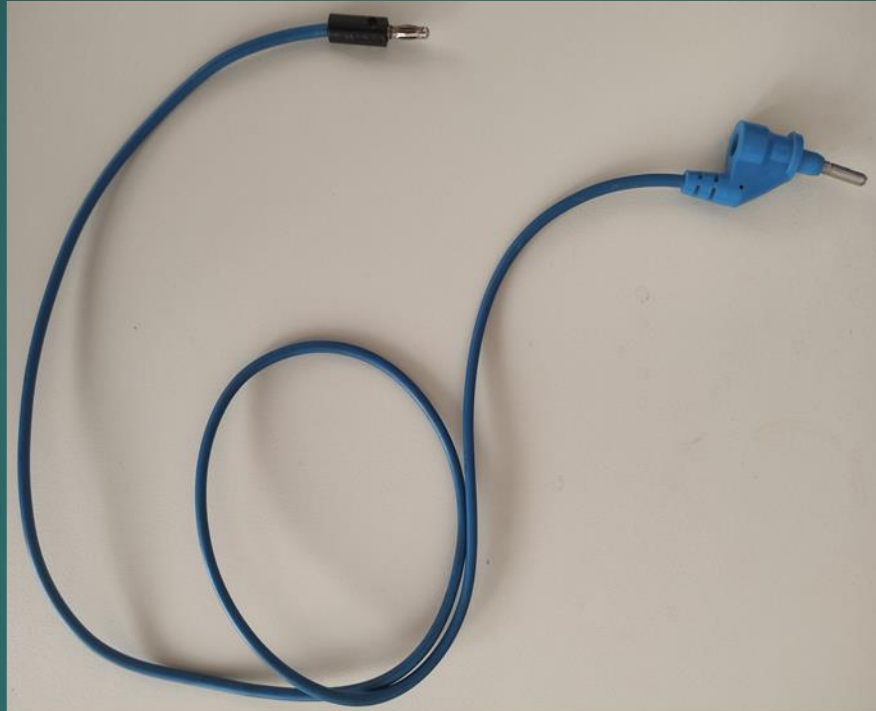
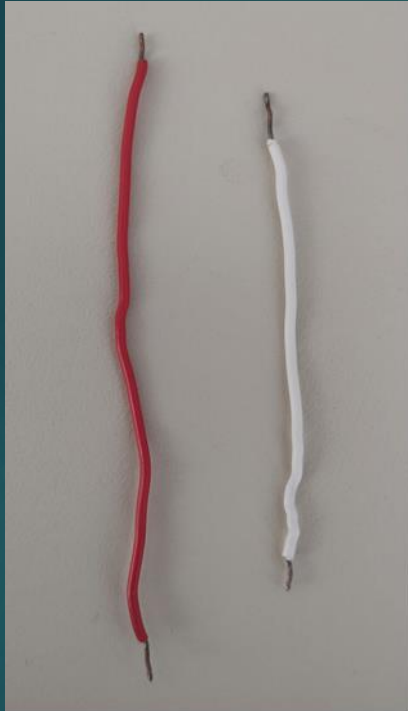
Σύνδεση αντιστάσεων παράλληλα

$$\frac{1}{R_{O\Lambda}} = \frac{1}{R_1} + \frac{1}{R_2} \Rightarrow R_{O\Lambda} = \frac{R_1 * R_2}{R_1 + R_2} = \frac{5 * 5}{5 + 5} = \frac{25}{10} = 2.5k\Omega$$

Σύνδεση αντιστάσεων σε σειρά

$$R_{O\Lambda} = R_1 + R_2 = 5 + 5 = 10k\Omega$$

Καλώδια



Μπανάνα



Κροκοδειλάκι



Ποτενσιόμετρο (1)

17

Το **ποτενσιόμετρο** είναι μία μεταβλητή αντίσταση. Πρόκειται για μία πίστα που αποτελείται από κάποιον αντιστάτη και πάνω σε αυτή την πίστα τρέχει ένας δρομέας. Μεταβάλλεται η αντίσταση στο ένα άκρο και τον δρομέα.

Η αντίσταση της πίστας παραμένει σταθερή (π.χ. $10\text{k}\Omega$) και είναι αυτή που χαρακτηρίζει την τιμή του ποτενσιόμετρου. Ο **δρομέας** μεταβάλλει την αντίσταση από το 0 έως τη μέγιστη τιμή.

Υπάρχουν διάφορα είδη ποτενσιόμετρων ανάλογα με την μετατόπισή τους. Εμείς θα χρησιμοποιήσουμε τα **περιστρεφόμενα** και της **ευθύγραμμης μετατόπισης**.

Με το ποτενσιόμετρο μπορούμε να μετρήσουμε **απόσταση** (μετατόπιση) ή **γωνία** (εκτροπή).

Η σχέση που συνδέει το ερέθισμα και την απόκριση είναι **γραμμική**.

Συνήθως την κλίση της ευθείας την υπολογίζουμε από τη μέγιστη απόκλιση (x_p ή θ_p) και τη μέγιστη αντίσταση (αντίσταση πίστας R_p).

Πάντα στα 2 άκρα είναι η **αντίσταση της πίστας** και το μεσαίο άκρο ο δρομέας.

Φροντίζουμε να βρούμε το **κοινό** (0) το οποίο είναι ένα από τα 2 άκρα (αντίσταση μεταξύ του άκρου και του δρομέα μηδενική). Όταν είμαστε στη θέση μηδέν (βαθμονόμηση) πρέπει και η αντίσταση του να είναι μηδενική.

Συνήθως δεν μας ενδιαφέρει να μετράμε αντίσταση αλλά τάση. Οπότε συνήθως στα ποτενσιόμετρα εφαρμόζουμε τάση (όπως και στο διαιρέτη τάσης).

Προσοχή! Στο ποτενσιόμετρο η τάση εφαρμόζεται **πάντα** στα 2 **άκρα** (όχι στο δρομέα).

Ποτενσιόμετρο (2)

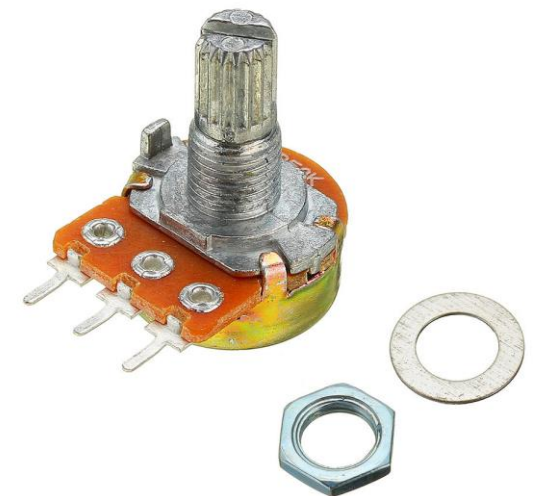
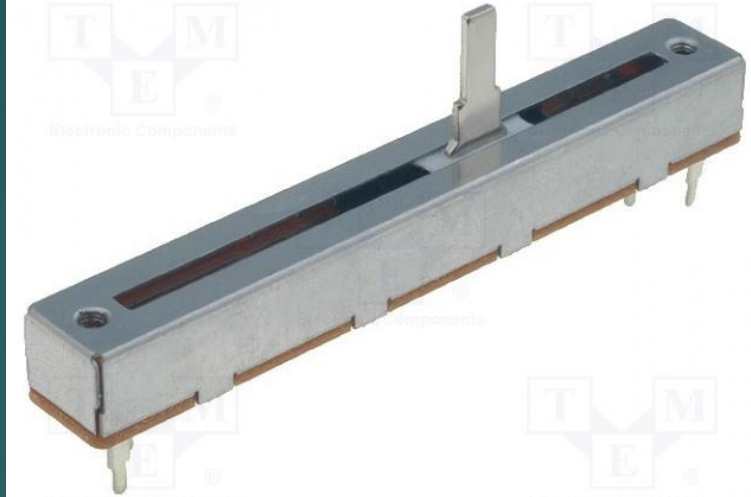
Περιστρεφόμενο πολλαπλών
στροφών (> 360)



Trimmer



Ευθύγραμμης μετατόπισης



Ποτενσιόμετρο (3)

19

(α) Άνθρακα (Carbon)

Σε αυτά η πίστα αποτελείται από συμπαγή άνθρακα ή επίστρωση άνθρακα πάνω σε μονωτικό υλικό.

(β) Σύρματος (wire wound)

Σε αυτά η πίστα αποτελείται από σύρμα χρωμιονικελίνης τυλιγμένο πάνω σε μονωτικό υλικό.

(γ) Φιλμ μετάλλου (Cermet thin film)

Σε αυτά η πίστα είναι ένα παχύ στρώμα μεταλλικού φιλμ κολλημένο πάνω σε κεραμικό υλικό.

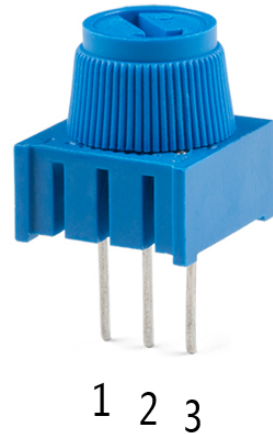
Ποτενσιόμετρο (4)

PANEL MOUNT POT 10KΩ



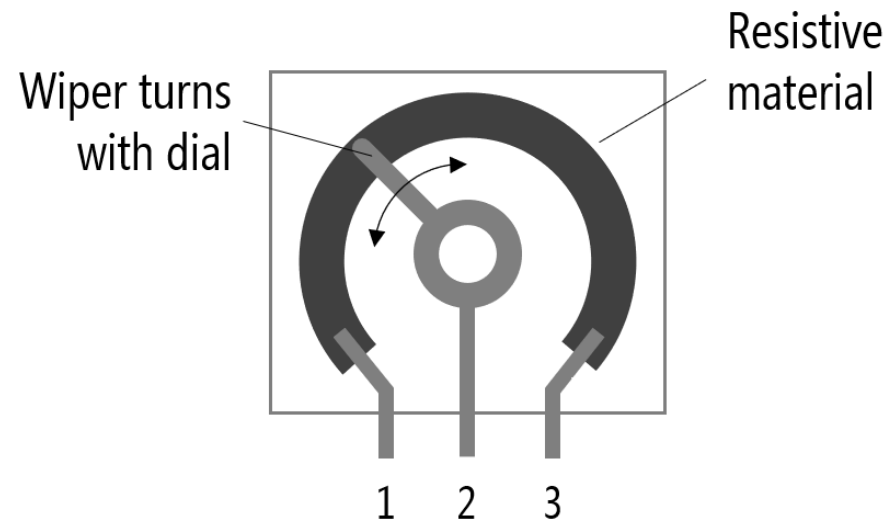
Wiper contact

TRIM POT 10KΩ



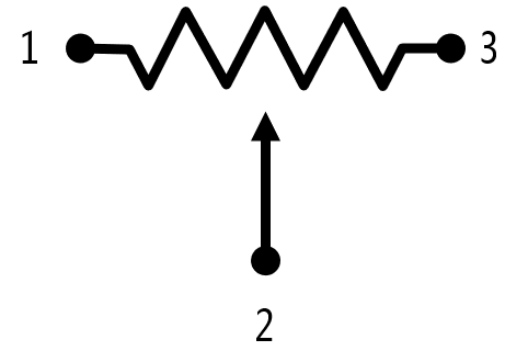
Wiper contact

FUNCTIONAL DIAGRAM



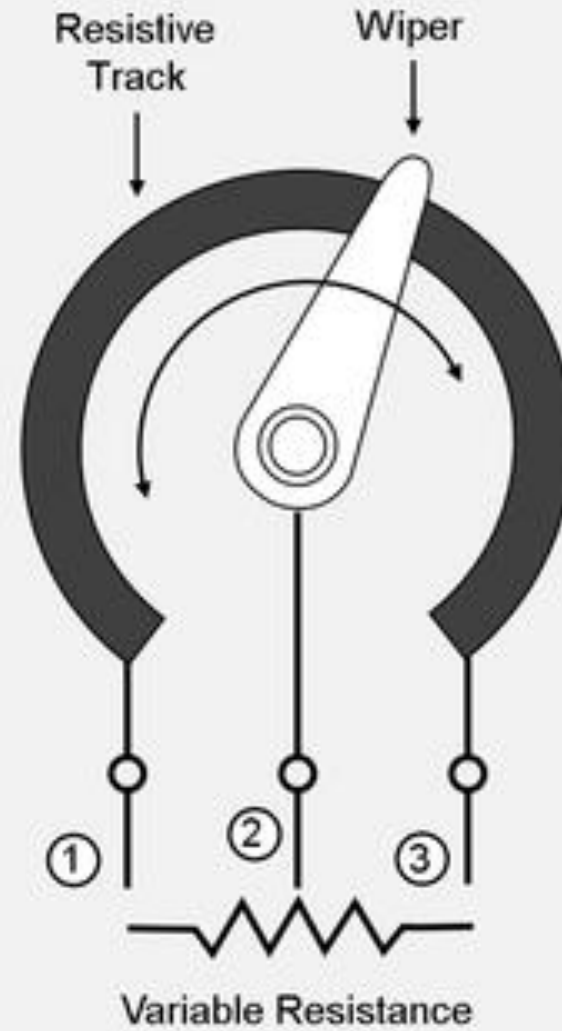
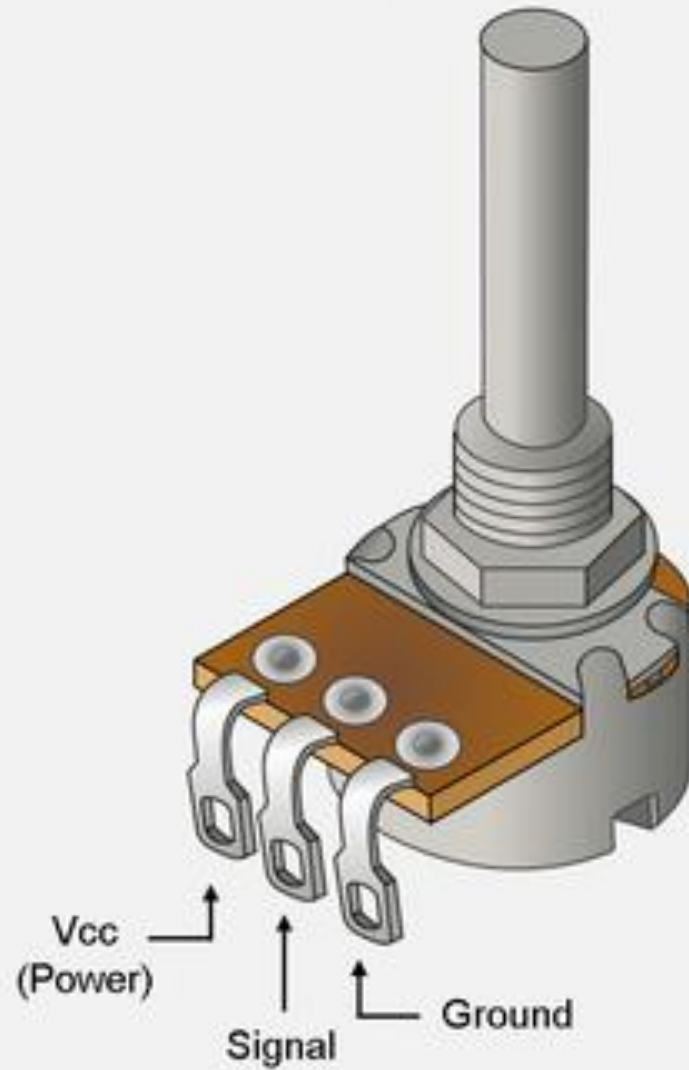
Wiper contact

SCHEMATIC SYMBOL



Wiper contact

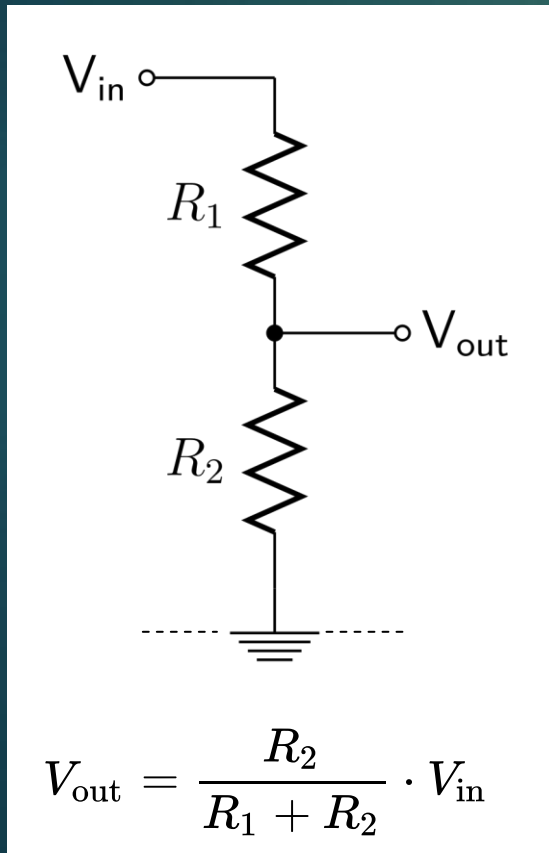
Ποτενσιόμετρο (5)



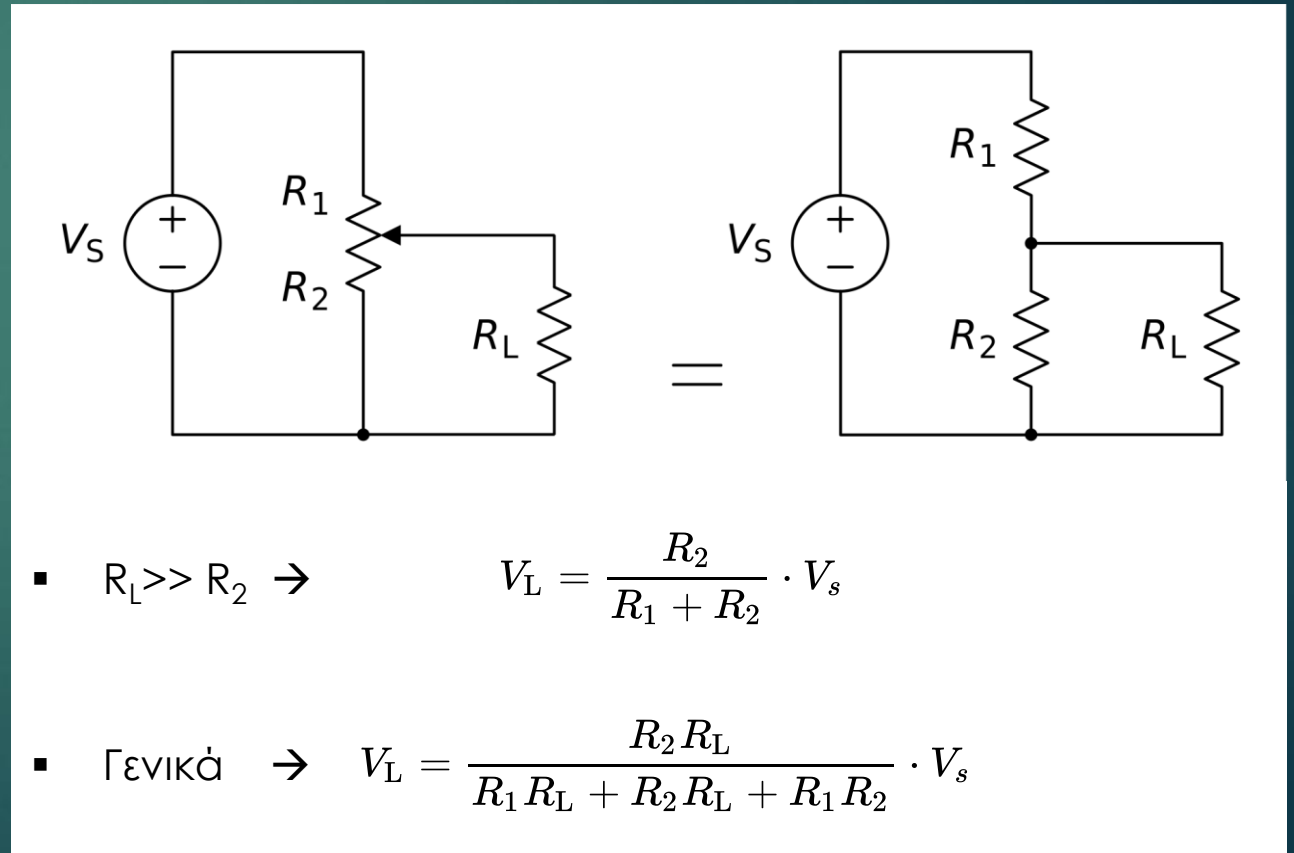
Ποτενσιόμετρο (6)

- Το ποτενσιόμετρο είναι ένας διαιρέτης τάσης.

Διαιρέτης τάσης
Υποβιβάζει την τάση

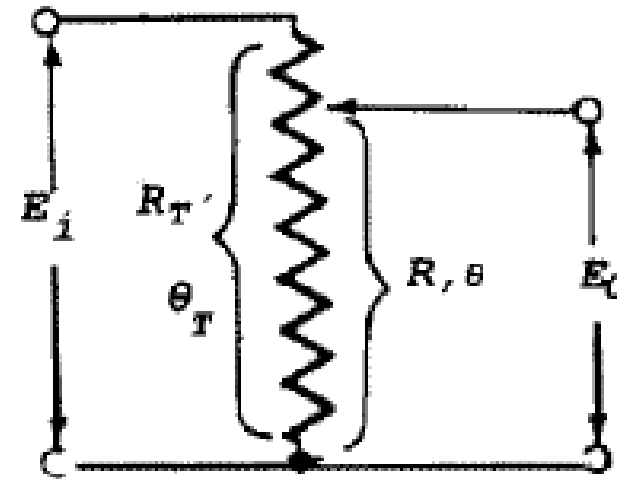
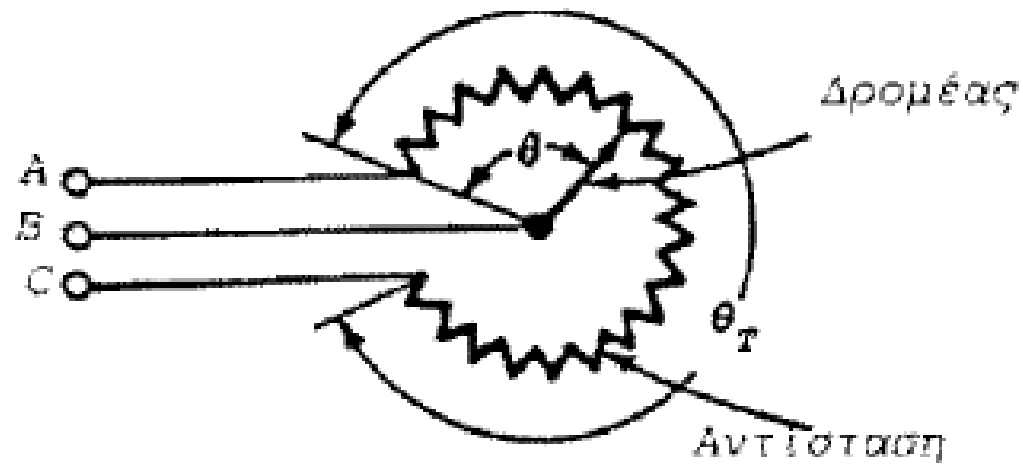


Ποτενσιόμετρο
Σχηματικό διάγραμμα



Ποτενσιόμετρο (7)

23



α)

β)

Σχήμα 2-3: Ποτενσιόμετρο: α) φυσική σχεδίαση και β) σχηματική σχεδίαση

$$\frac{R}{R_T} = \frac{\theta}{\theta_T}$$

$$\frac{E_o}{E_i} = \frac{R}{R_T} = \frac{\theta}{\theta_T}$$

$$\text{ή } E_o = \frac{\theta}{\theta_T} E_i$$

Πολύμετρο

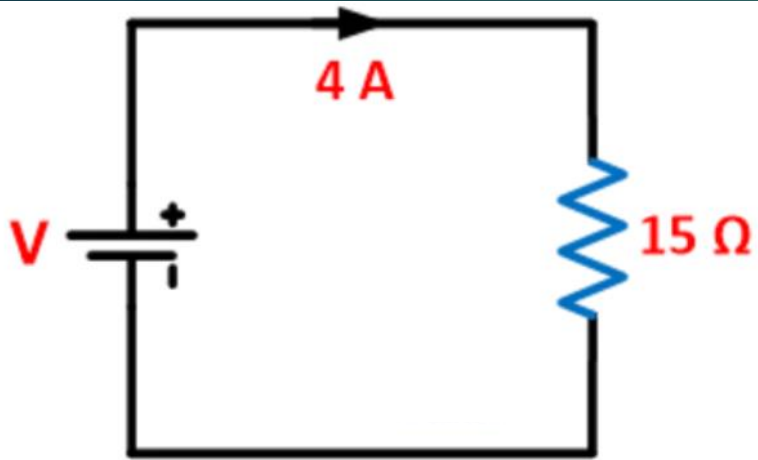
24



- Το **πολύμετρο** είναι ένα ηλεκτρολογικό/ηλεκτρονικό όργανο το οποίο μετράει την τάση, την ένταση και την αντίσταση σε ένα ηλεκτρικό κύκλωμα.
- Είναι μια φορητή συσκευή χρήσιμη για τη εύρεση ελαττωμάτων ή ένα όργανο που μπορεί να μετρήσει σε έναν πολύ υψηλό βαθμό ακριβείας.
- Μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει τα πρότυπα αντίστασης και τάσης ή να ρυθμίσει και να ελέγξει την απόδοση των πολλών χρήσεων.
- Έχει όλες τις δυνατότητες των παραπάνω οργάνων (ωμόμετρο, βολτόμετρο, αμπερόμετρο) και ανάλογα την μέτρηση που κάνουμε αλλάζουμε τον μεταγωγέα στο κατάλληλο όργανο ενώ ταυτόχρονα απομονώνουμε τα υπόλοιπα.

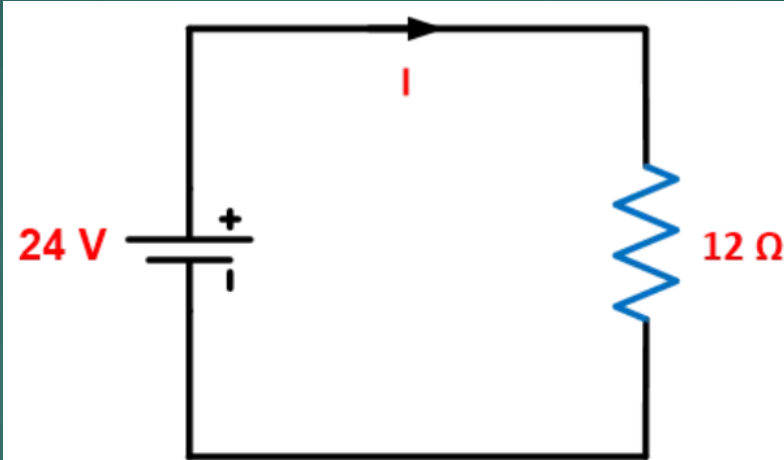
Νόμος του Ohm – Παραδείγματα (1)

25



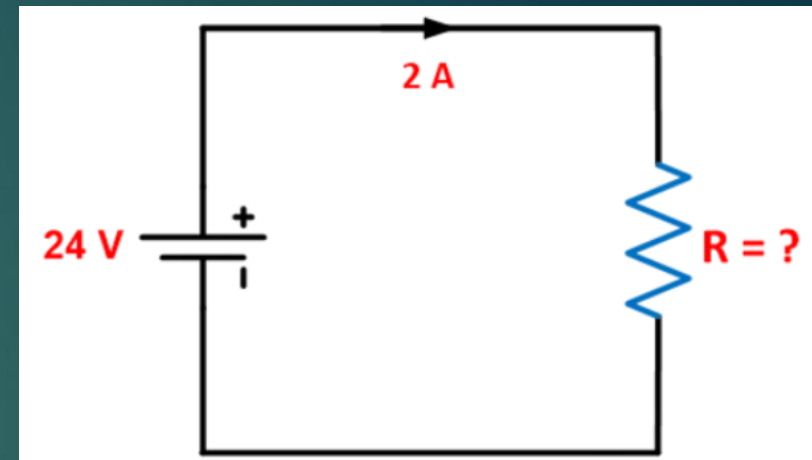
$I = 4 \text{ A}$ and $R = 15 \Omega$

$$\begin{aligned} V &= I * R \\ &= 4 * 15 \\ V &= 60 \text{ Volts} \end{aligned}$$



$V = 24 \text{ V}$ and $R = 12 \Omega$

$$\begin{aligned} I &= \frac{V}{R} \\ &= \frac{24}{12} \\ I &= 2 \text{ A (Ampere)} \end{aligned}$$

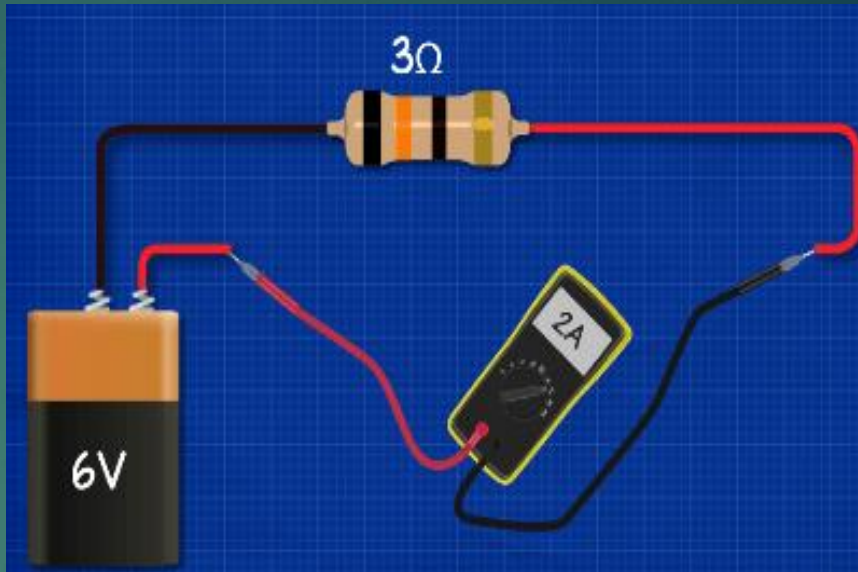
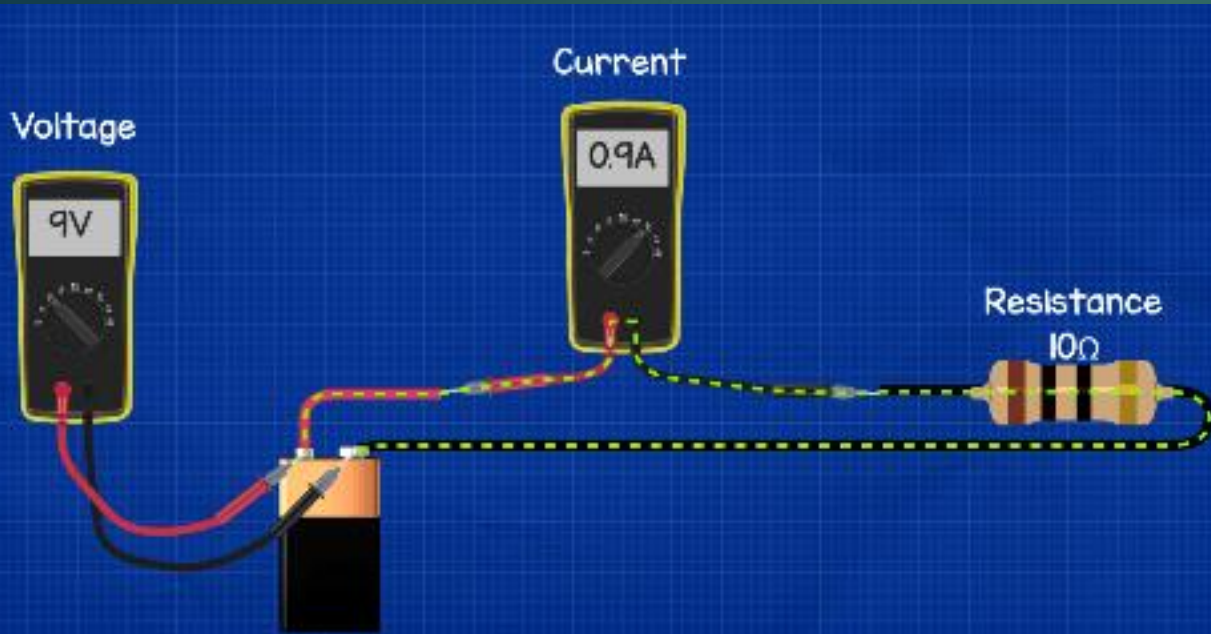


$V = 24 \text{ V}$ and $I = 2 \text{ A}$

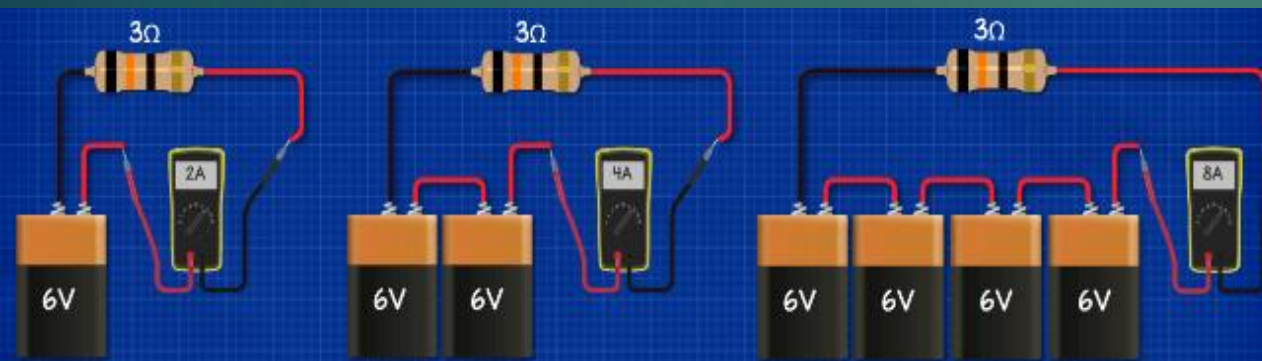
$$\begin{aligned} R &= \frac{V}{I} \\ &= \frac{24}{2} \\ R &= 12 \Omega \end{aligned}$$

Νόμος του Ohm – Παραδείγματα (2)

26



$$V = I \times R$$
$$V = 2A \times 3\Omega$$
$$V = 6$$



Relationship: current is therefore directly proportional to voltage.

Double the Voltage, double the current



To find current

$$I = V \div R$$

Current = Voltage ÷ Resistance

$$I = V \div R$$

$$I = 6V \div 3\Omega$$

$$I = 2A$$



Νόμος του Ohm – Παραδείγματα (3)

Relationship: Current is inversely proportional to resistance
Double the resistance, half the current

Find resistance
 $R = V \div I$
Resistance = Voltage \div Current
 $R = 12V \div 0.5A$
 $R = 24\Omega$

Note: this will measure the resistance of the entire circuit

Increase current by increasing the voltage

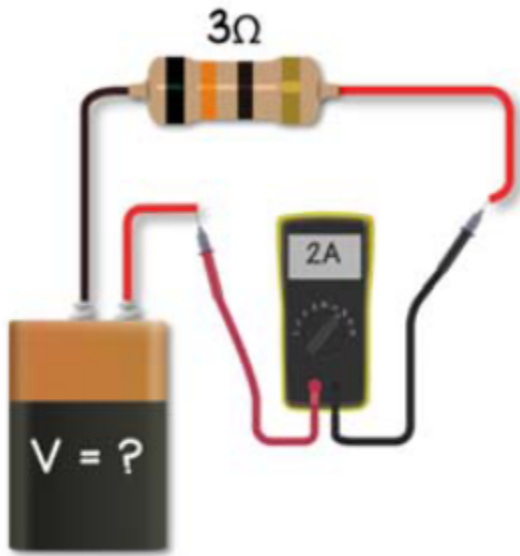
Low voltage, low current Higher voltage, higher current

Increase current by reducing the resistance
or
Reduce current by increasing resistance

Low voltage, low current Low voltage, high resistance, low current

Νόμος του Ohm – Παραδείγματα (4)

28



In this circuit we have a 3Ω resistor connected to a battery with an unknown voltage. The current in the circuit is 2A. Find the voltage.

Using ohm's triangle, we find the formula:
Voltage (V) = Current (I) x Resistance (R)

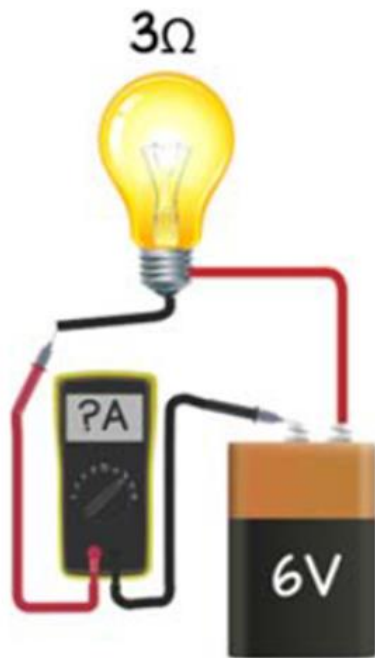
$$\text{Voltage} = 2\text{A} \times 3\Omega$$

$$\text{Voltage} = 6\text{V}$$

Answer is 6V

Νόμος του Ohm – Παραδείγματα (5)

29



In this circuit we have a 3Ω resistive lamp connected to a $6V$ battery. The current in the circuit is unknown. Find the current.

Using ohm's triangle, we find the formula:
Current (I) = Voltage (V) \div Resistance (R)

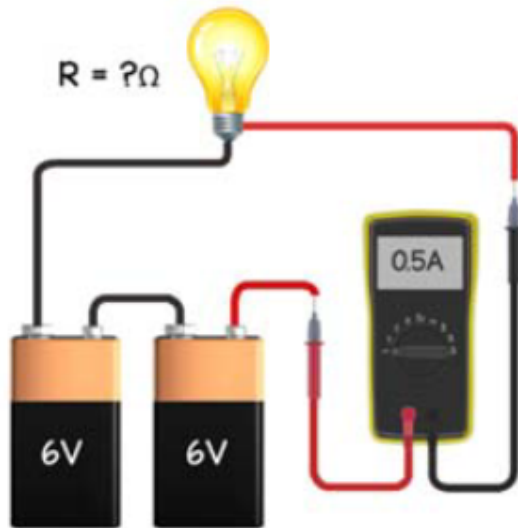
$$\text{Current} = 6V \div 3\Omega$$

$$\text{Current} = 2A$$

Answer is 2A

Νόμος του Ohm – Παραδείγματα (6)

30



In this circuit we have a resistive lamp connected to two 6V batteries, which are wired in series, providing 12V to the circuit. The current in the circuit is 0.5A. Find the resistance.

Using ohm's triangle, we find the formula:
Resistance (R) = Voltage (V) ÷ Current (I)

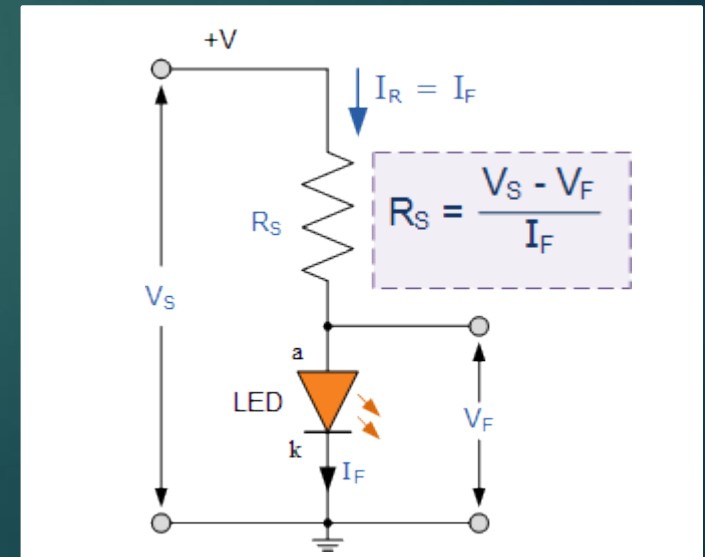
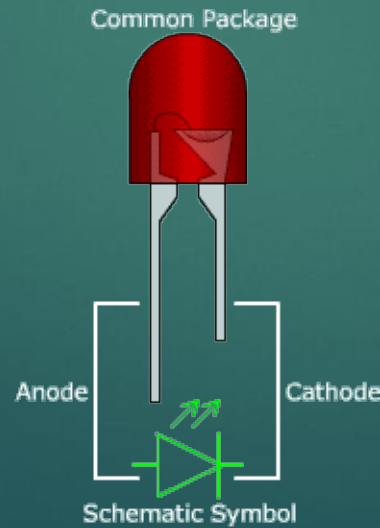
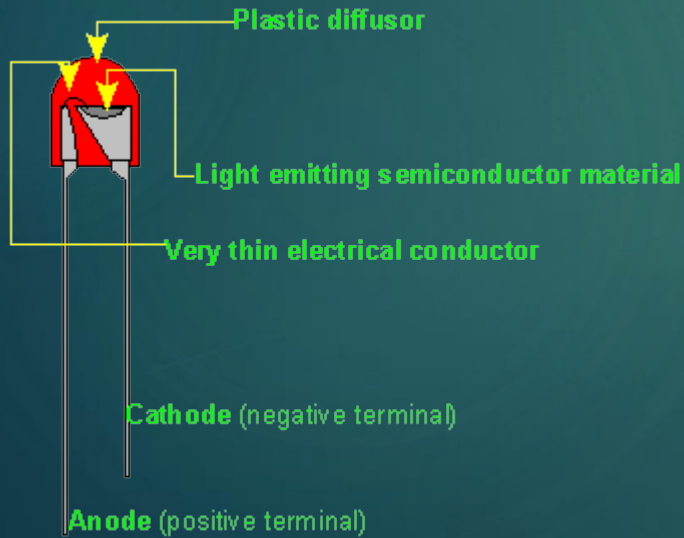
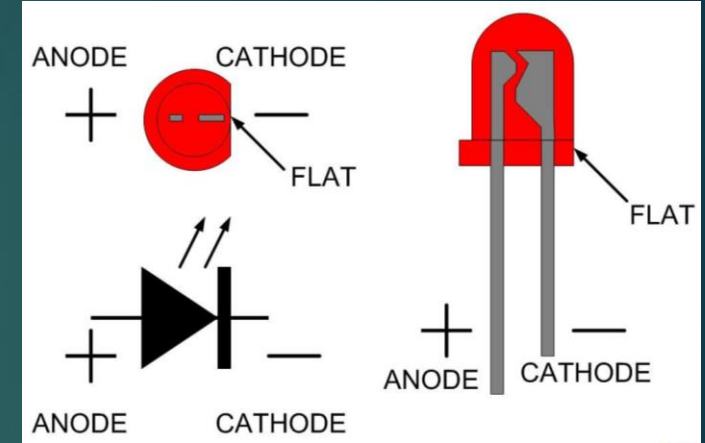
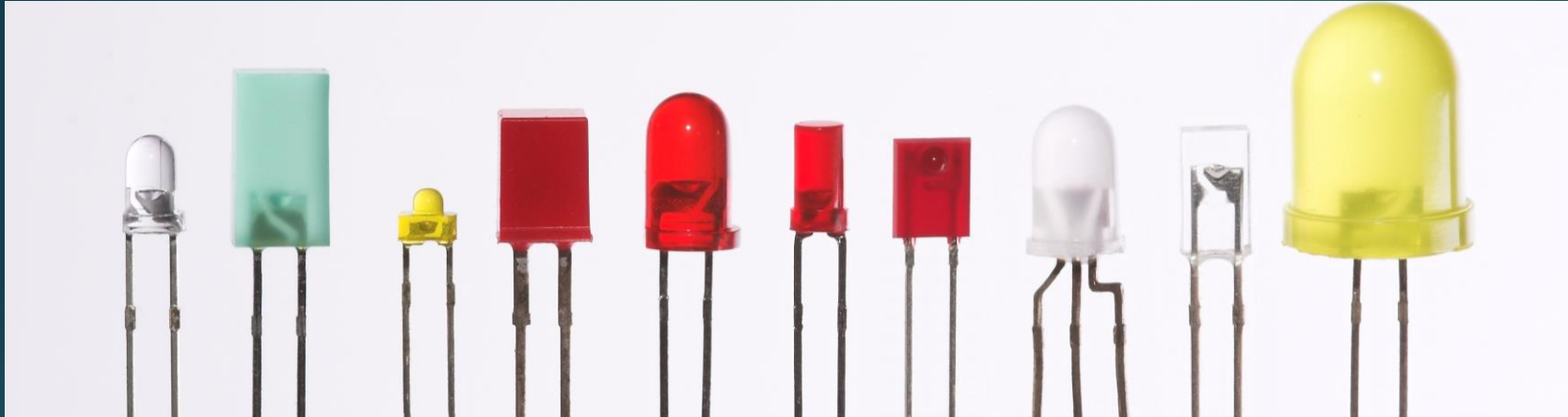
$$\text{Resistance} = 12\text{V} \div 0.5\text{A}$$

$$\text{Resistance} = 24\Omega$$

Answer is 24Ω

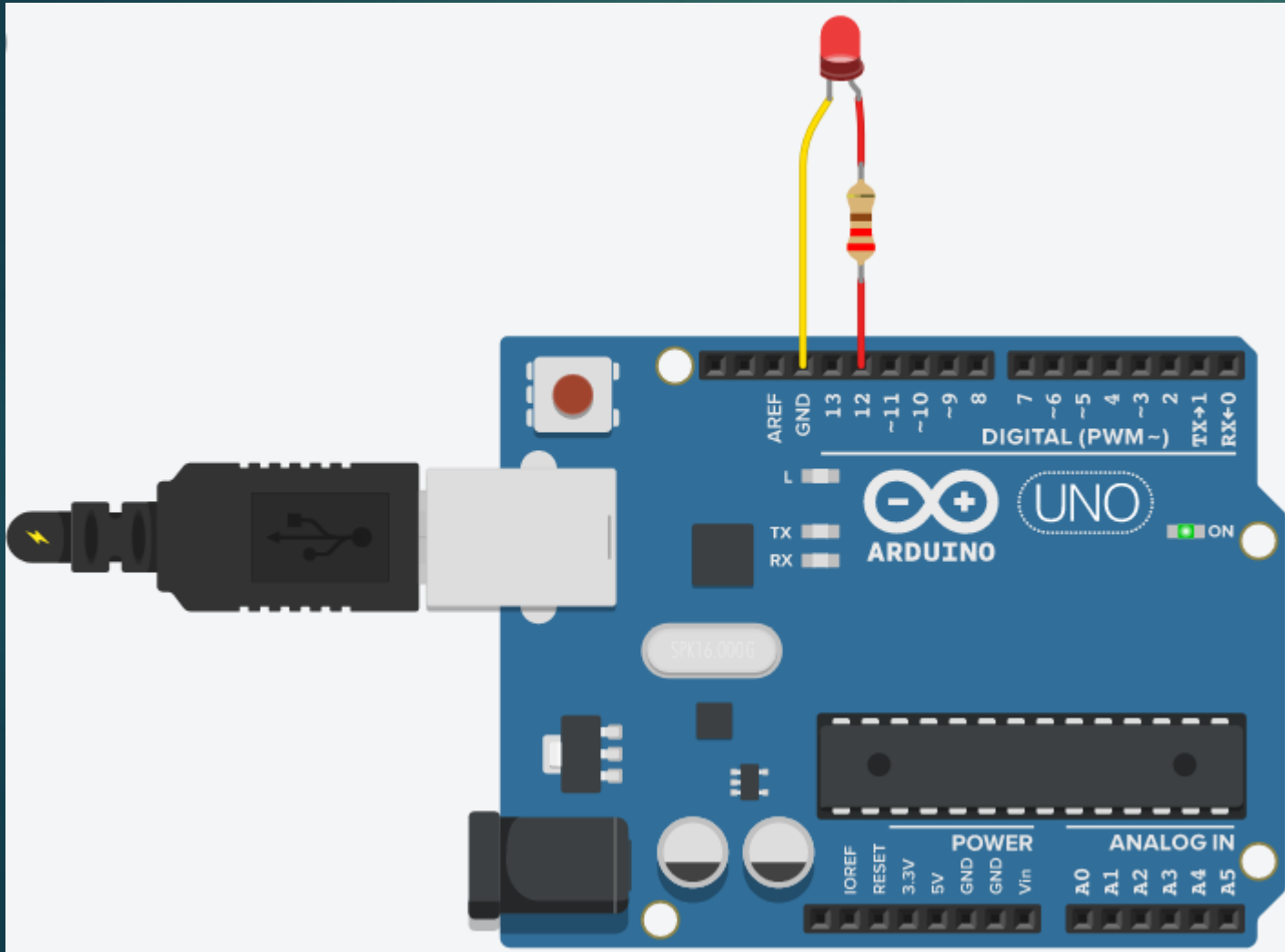
LED: Light Emitting Diode

31



Παράδειγμα 2: Turn On-Off a LED

32



```
1 int LED = 12; // Το Pin που συνδέουμε το LED
2
3 void setup() // Κώδικας που τρέχει μία φορά
4 {
5   pinMode(LED, OUTPUT); // Δήλωση του LED ως έξοδο
6 }
7
8 void loop() // Κώδικας που επαναλαμβάνεται
9 {
10  digitalWrite(LED, HIGH); // Ανάβει το LED
11  delay(1000); // ON για 1sec
12
13  digitalWrite(LED, LOW); // Σβήνει το LED
14  delay(1000); // OFF για 1sec
15 }
```

Παράδειγμα 2: Turn On-Off a LED

33

The image shows the Arduino IDE interface. On the left, the 'Blocks + Text' palette is visible, containing various blocks for controlling the Arduino. The main workspace displays a Scratch-style block-based program with the following sequence of blocks:

- set built-in LED to HIGH
- set pin 0 to HIGH
- set pin 3 to 0
- rotate servo on pin 0 to 0 degrees
- play speaker on pin 0 with tone 60
- turn off speaker on pin 0
- print to serial monitor hello world with
- set RGB LED in pins 3 3 3

In the center workspace, a specific sequence of blocks is shown:

- set pin 12 to HIGH
- wait 1 secs
- set pin 12 to LOW
- wait 1 secs

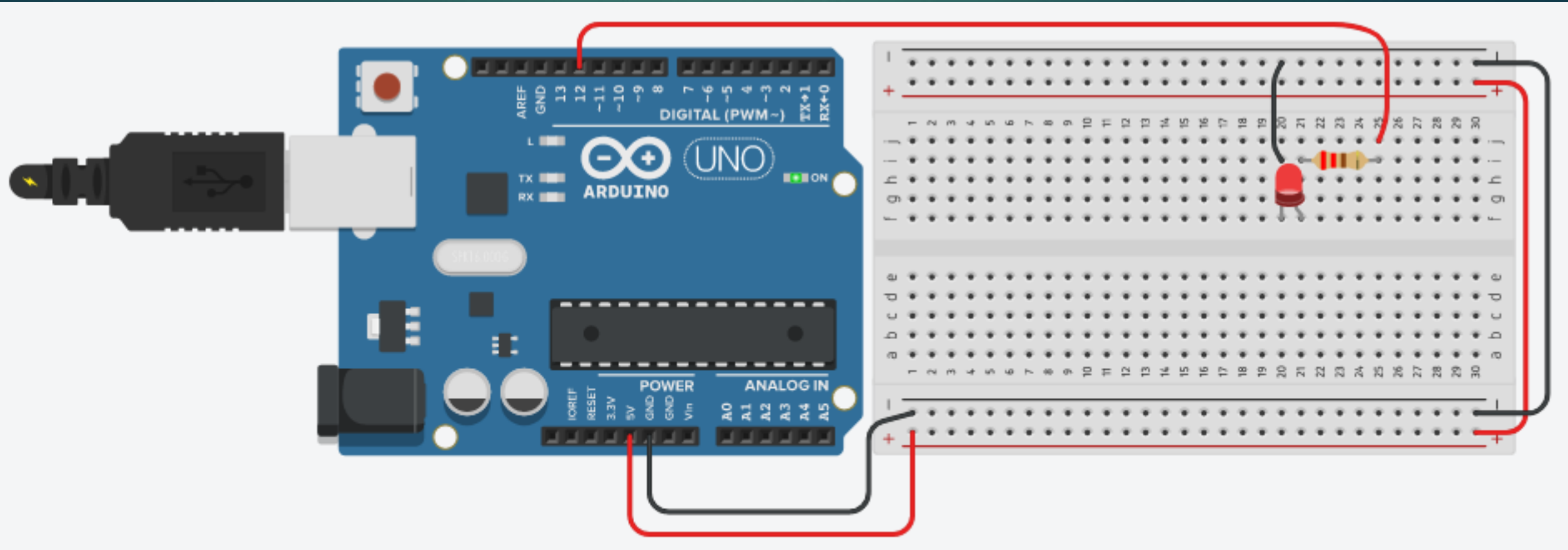
On the right, the C++ code for the program is displayed:

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

At the bottom left, the 'Serial Monitor' tab is visible.

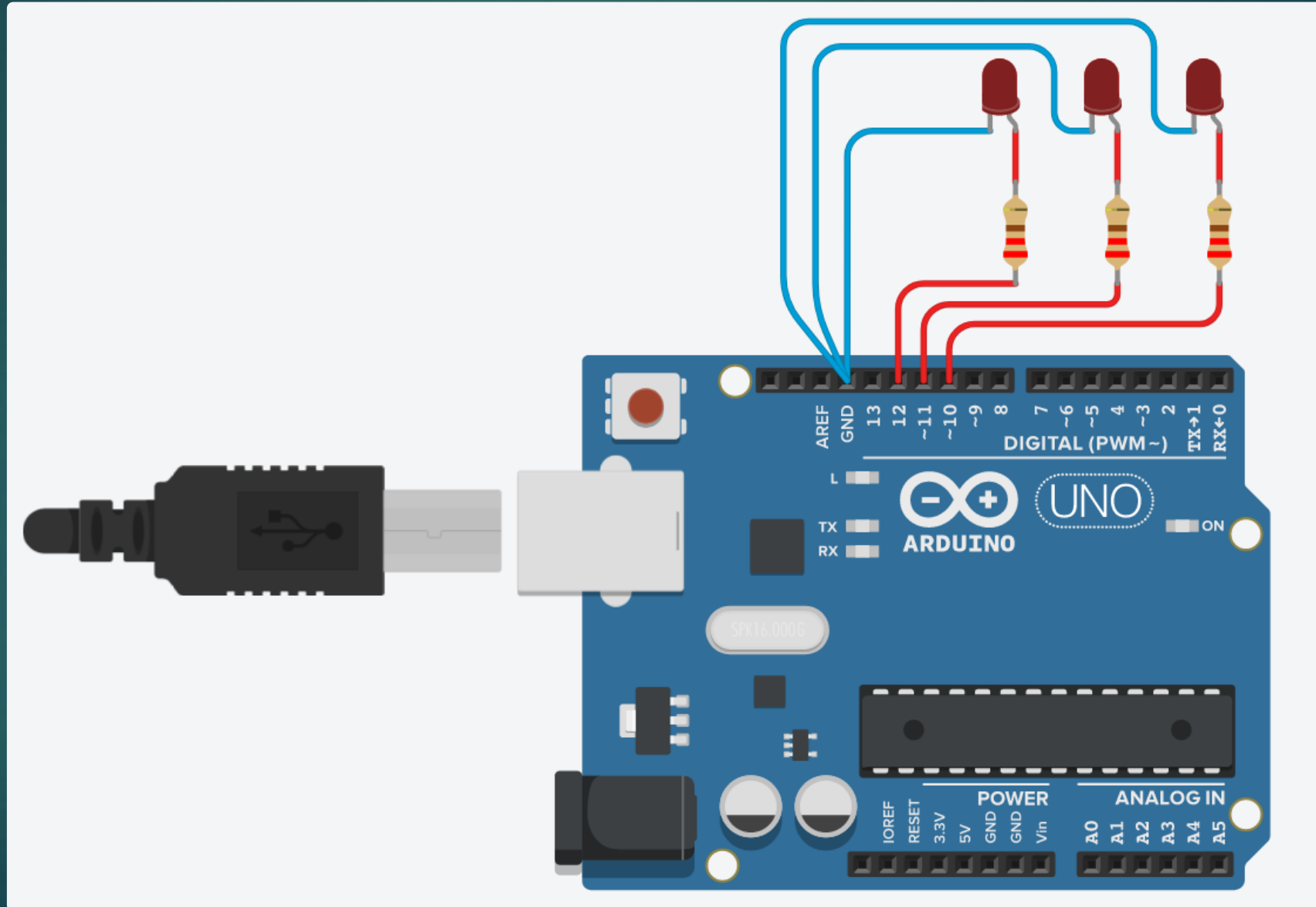
Παράδειγμα 2: Με χρήση Breadboard

34



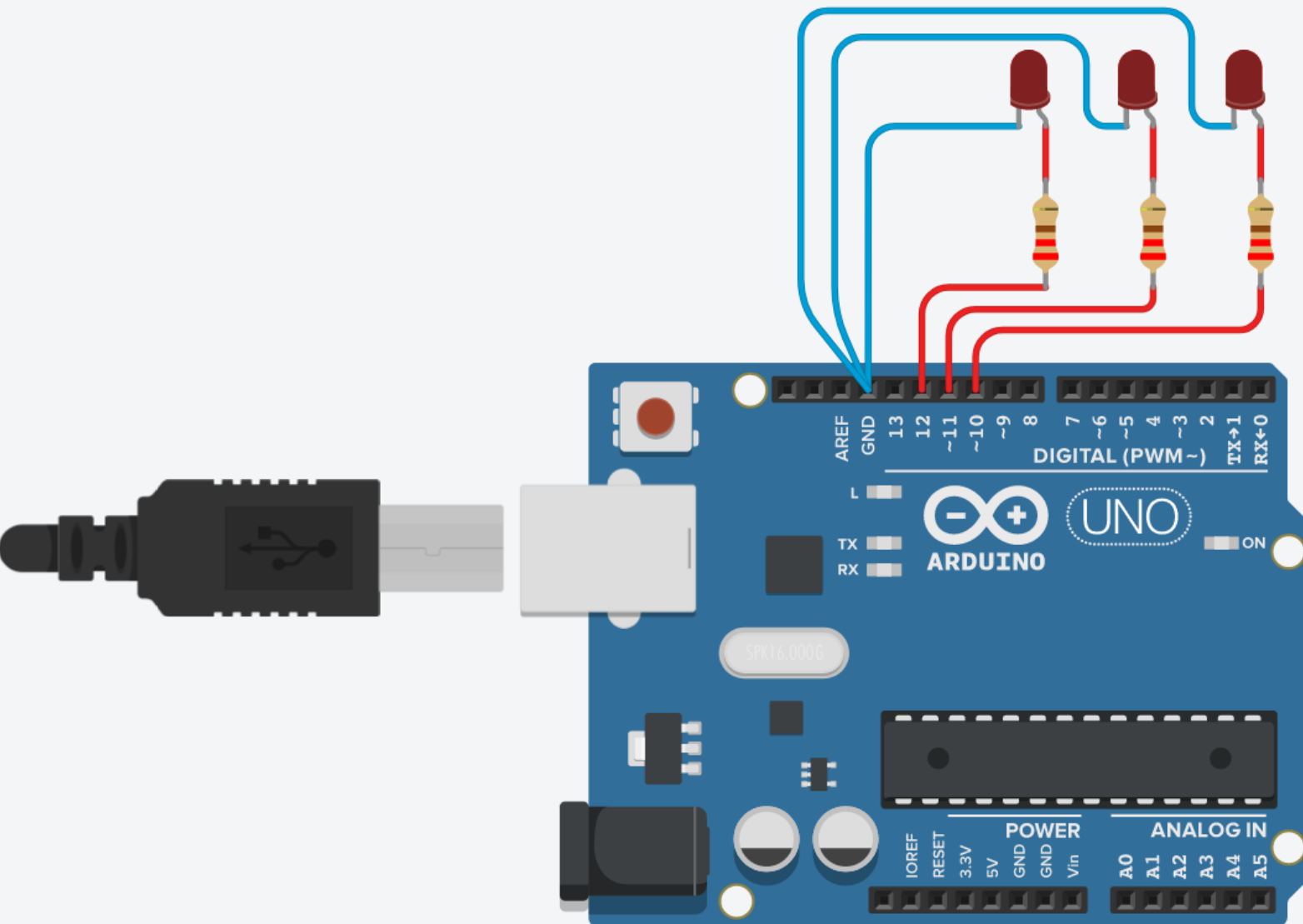
Παράδειγμα 3: For loop with 3 LEDs

35



Παράδειγμα 3: For loop with 3 LEDs

36

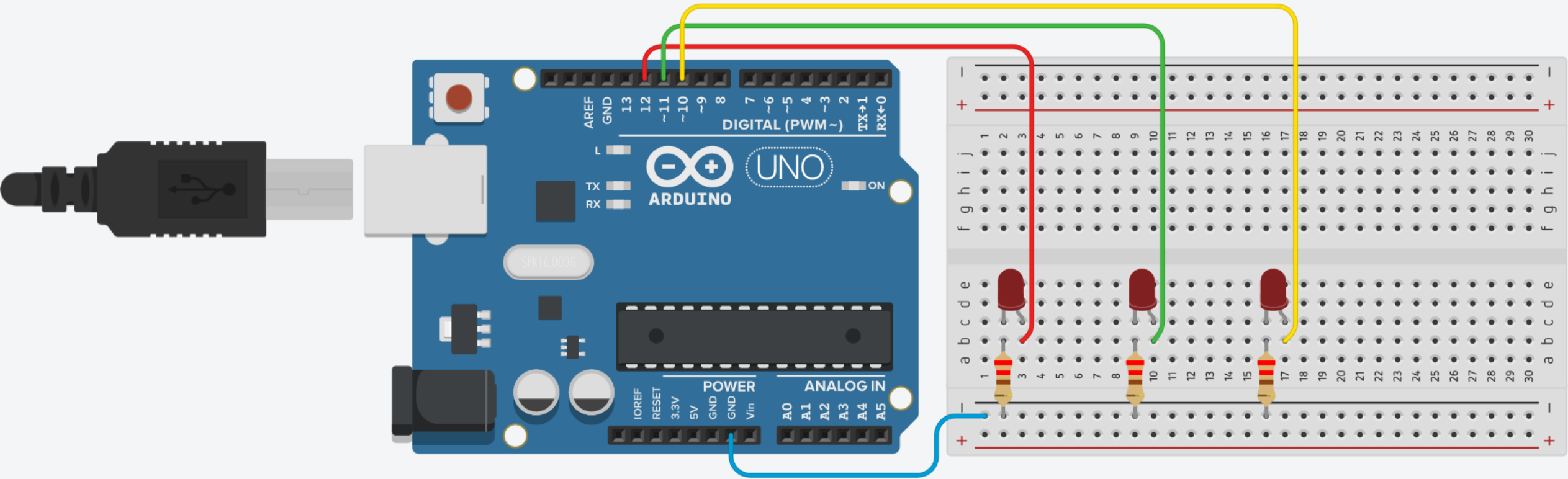


```
Text 1 (Arduino Uno R3)
1 int i;
2 void setup()
3 {
4   for (i=10;i<13;i++)
5   {
6     Serial.begin(9600);
7     pinMode(i,OUTPUT);
8   }
9 }
10
11 void loop()
12 {
13   for (i=10;i<13;i++)
14   {
15     digitalWrite(i,HIGH);
16     Serial.print("ON LED ");
17     Serial.print(i);
18     Serial.println();
19     delay(2000);
20     digitalWrite(i,LOW);
21   }
22 }
```

Serial Monitor

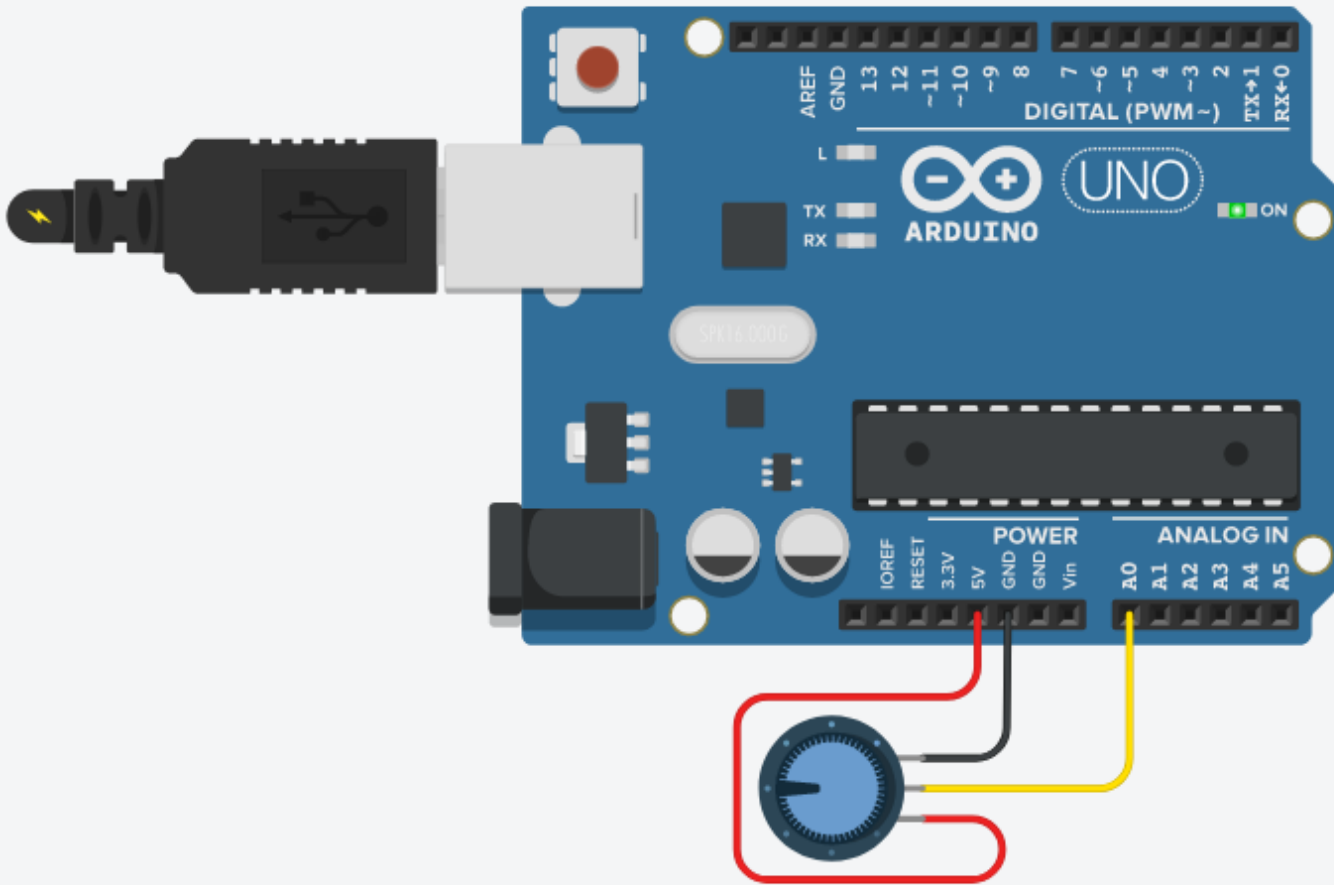
Παράδειγμα 3: Breadboard

37



Παράδειγμα 4α: Potentiometer

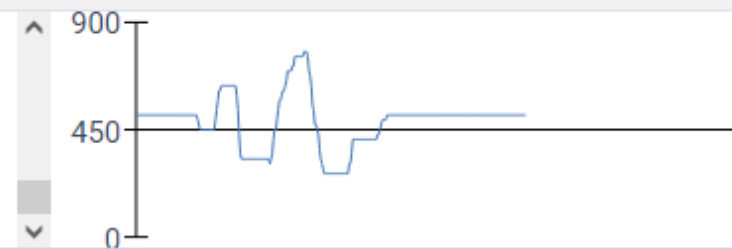
38



```
1 // the setup routine runs once when you press reset:
2 void setup() {
3   // initialize serial communication at 9600 bits per second:
4   Serial.begin(9600);
5 }
6
7 // the loop routine runs over and over again forever:
8 void loop() {
9   // read the input on analog pin 0:
10  int sensorValue = analogRead(A0);
11  // print out the value you read:
12  Serial.println(sensorValue);
13  delay(100); //delay in between reads for stability
14 }
```

Serial Monitor

511
511
511
511
511
511
511
511



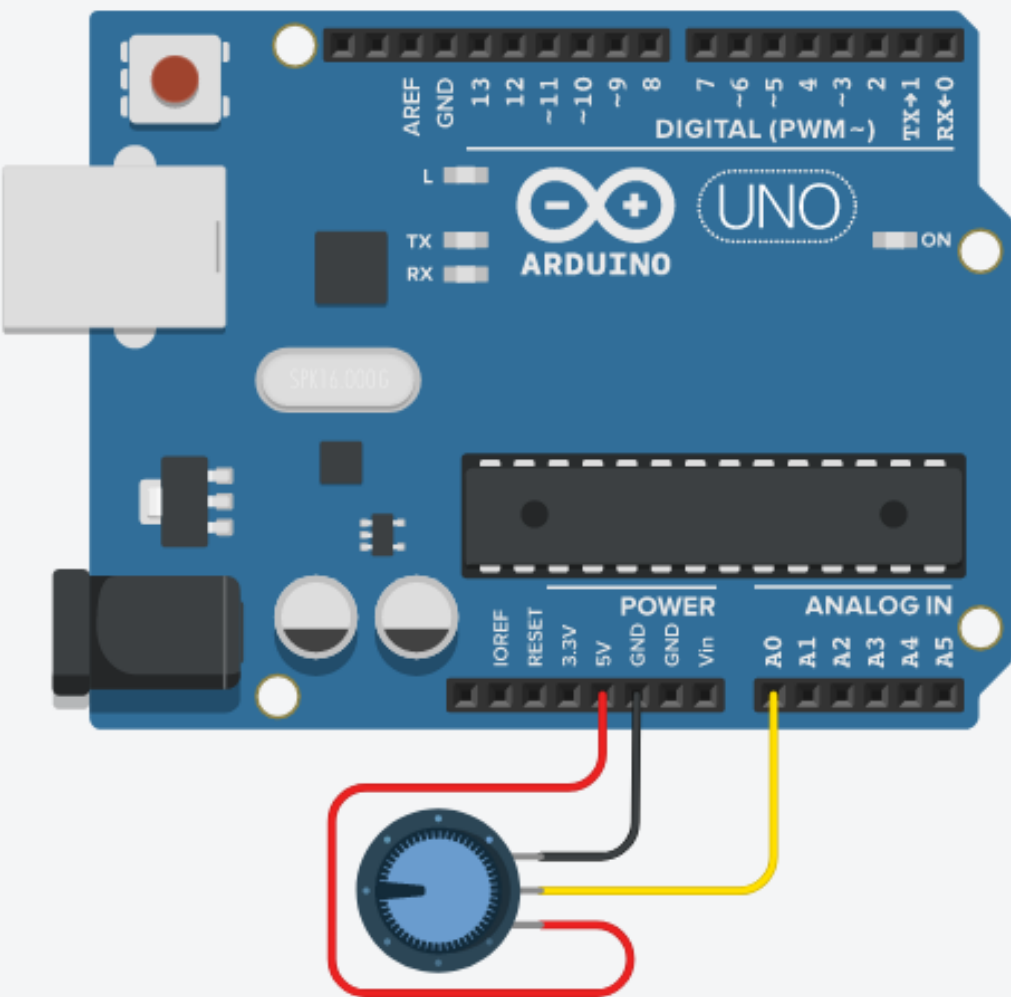
Send

Clear



Παράδειγμα 4b: Potentiometer

39



Text



1 (Arduino Uno R3)

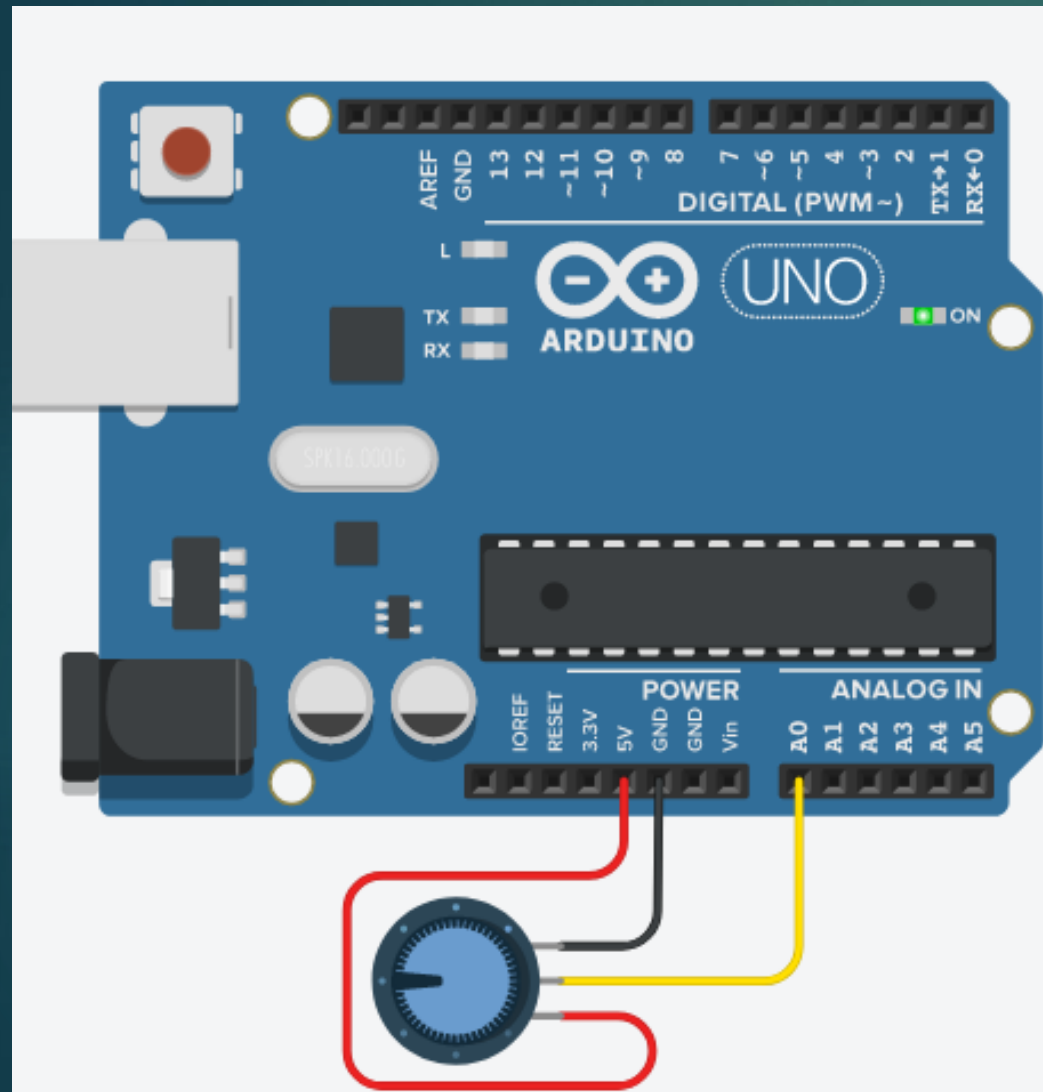
```
1  /*
2  ReadAnalogVoltage
3  Reads an analog input on pin 0, converts it to voltage,
4  and prints the result to the serial monitor.
5
6  OPEN THE SERIAL MONITOR TO VIEW THE OUTPUT >>
7  Attach the center pin of a potentiometer to pin A0,
8  and the outside pins to +5V and ground.
9  This example code is in the public domain.
10 */
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14   // initialize serial communication at 9600 bits per second:
15   Serial.begin(9600);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20   // read the input on analog pin 0:
21   int sensorValue = analogRead(A0);
22   // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V)
23   float voltage = sensorValue * (5.0 / 1023.0);
24   // print out the value you read:
25   Serial.println(voltage);
26 }
```



Serial Monitor

Παράδειγμα 4b: Potentiometer

40



```
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14   // initialize serial communication at 9600 bits per second:
15   Serial.begin(9600);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20   // read the input on analog pin 0:
21   int sensorValue = analogRead(A0);
22   // Convert the analog reading (which goes from 0 - 1023) to a voltage
23   float voltage = sensorValue * (5.0 / 1023.0);
24   // print out the value you read:
25   Serial.println(voltage);
26 }
```

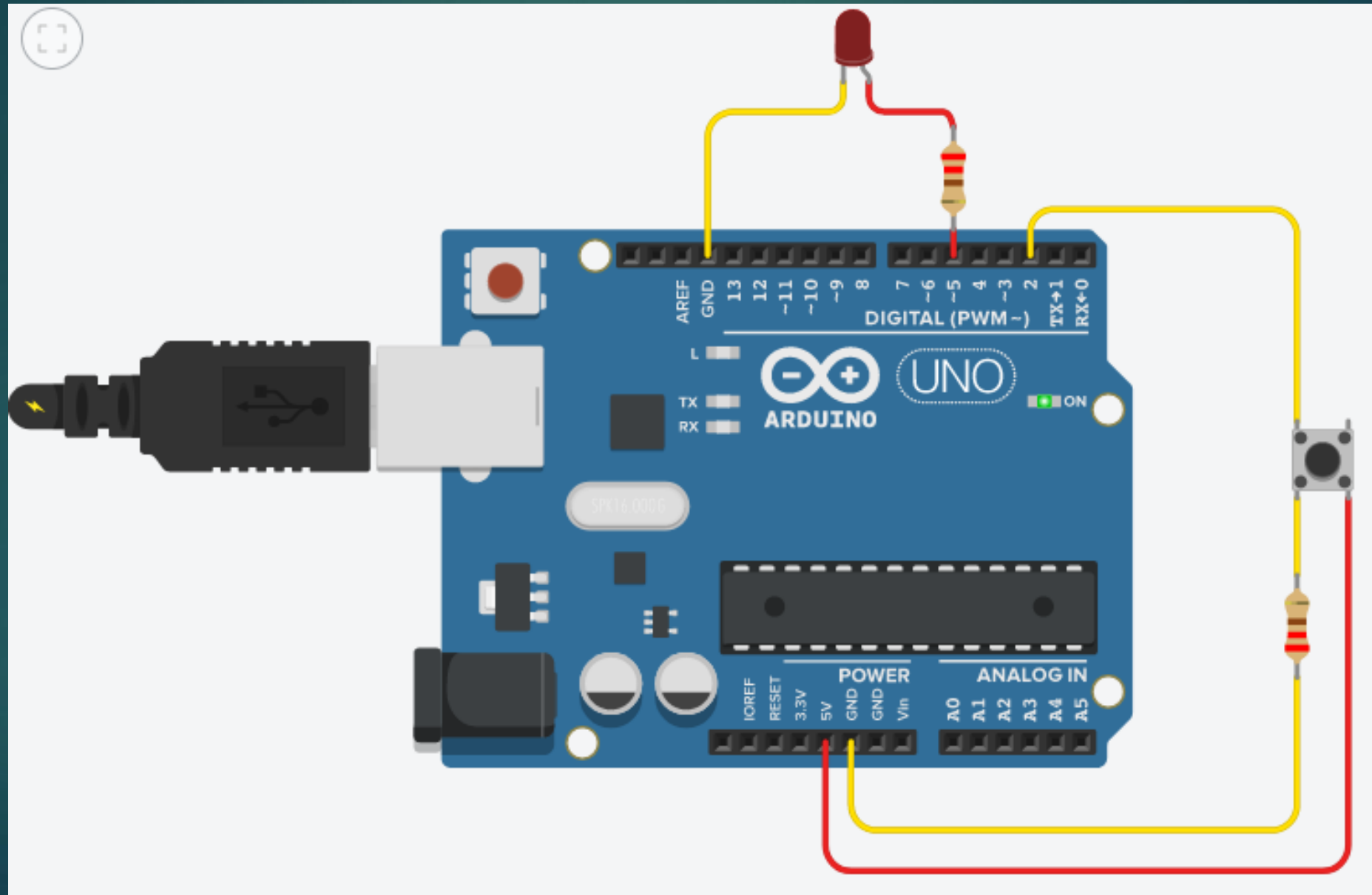
Serial Monitor

| | |
|------|---|
| 2.50 | 6 |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.50 | |
| 2.5 | 0 |

Send Clear

Παράδειγμα 5: Push-button

41



Παράδειγμα 5: Push-button

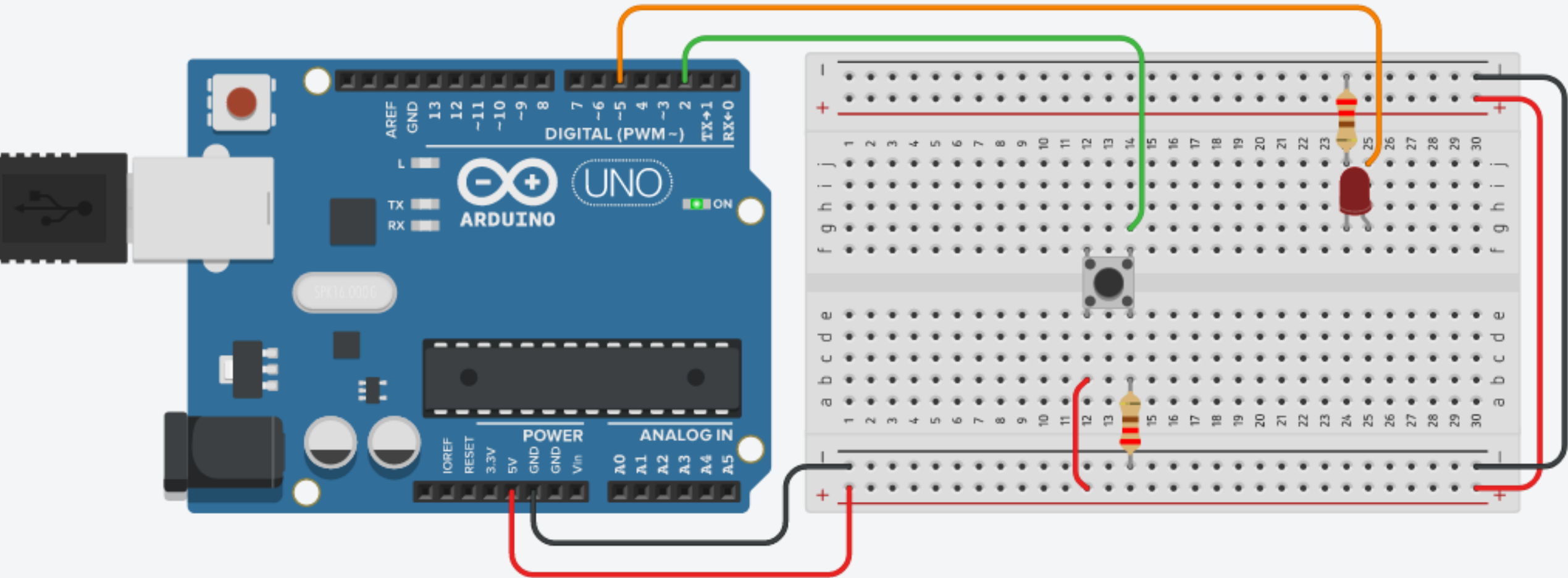
42

```
1 int buttonState = 0; // Initialize button state
2
3 void setup()
4 {
5   Serial.begin(9600);
6   pinMode(2, INPUT); // Set pin 2 as input
7   pinMode(5, OUTPUT); // Set pin 5 as output
8 }
9
10 void loop()
11 {
12   // read the state of the pushbutton value
13   int sensorValue = digitalRead(2);
14
15   // check if pushbutton is pressed.
16   if (sensorValue == HIGH) // if it is, the buttonState is HIGH
17   {
18     Serial.println(sensorValue);
19     digitalWrite(5, HIGH); // turn LED on
20   }
21   else // if it is not pressed, the buttonState is LOW
22   {
23     Serial.println(sensorValue);
24     digitalWrite(5, LOW); // turn LED off
25   }
26   delay(10); // Delay a little bit to improve simulation performance
27 }
```



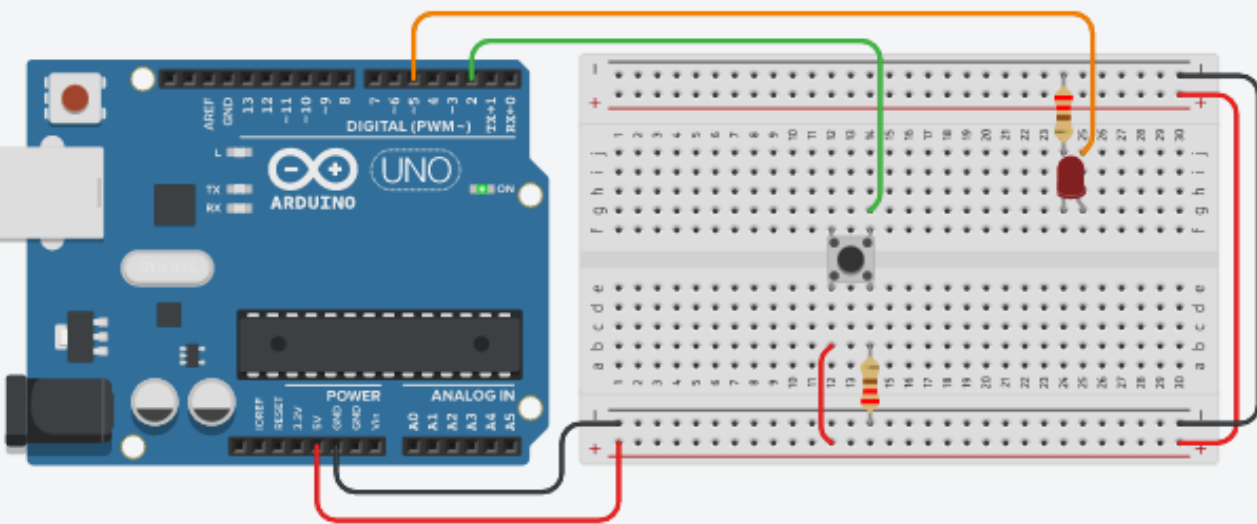
Παράδειγμα 5: Με Breadboard

43



Παράδειγμα 5: Με Breadboard

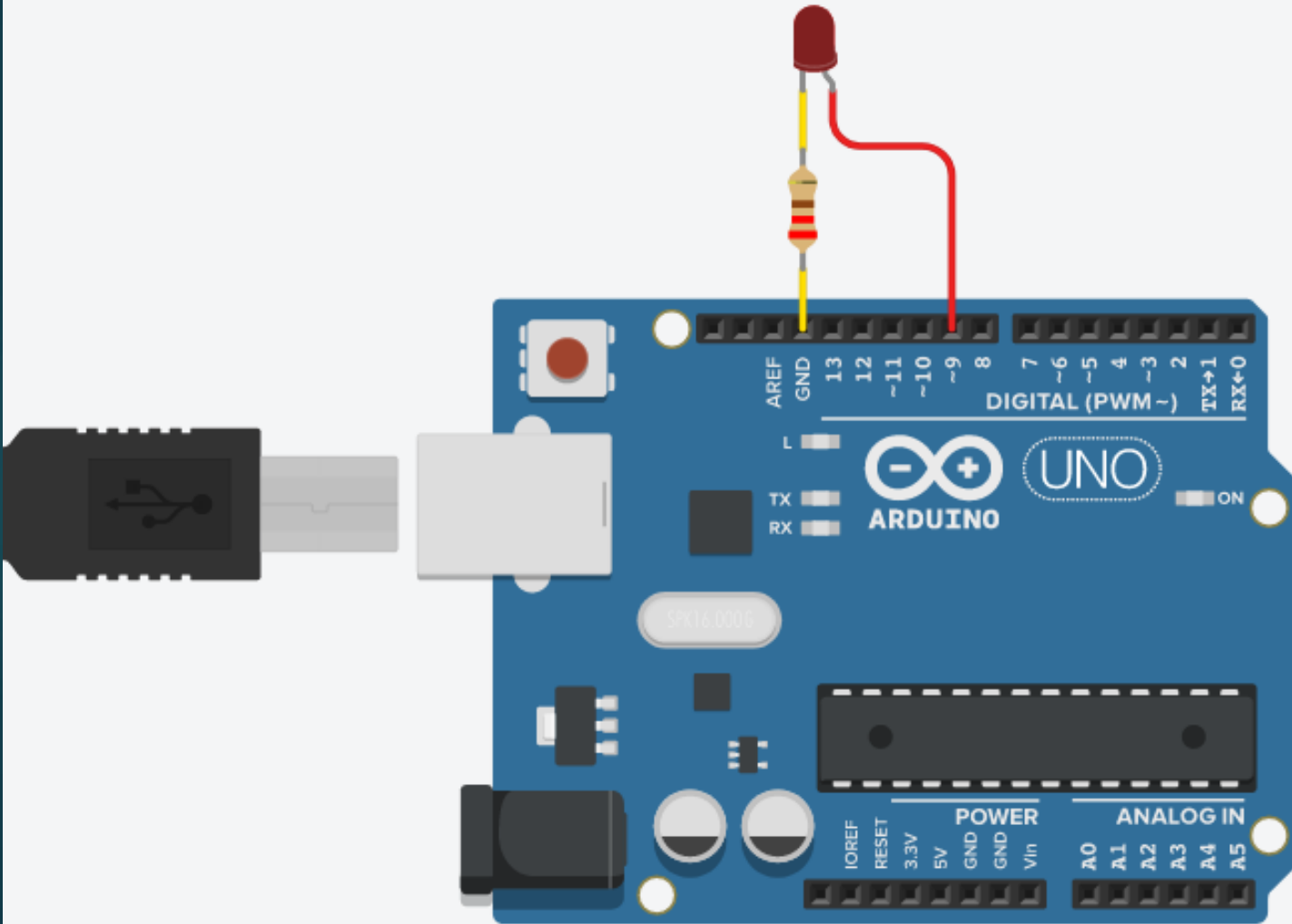
44



```
1 int buttonState = 0; // Initialize button state
2
3 void setup()
4 {
5     pinMode(2, INPUT); // Set pin 2 as input
6     pinMode(5, OUTPUT); // Set pin 5 as output
7 }
8
9 void loop()
10 {
11     // read the state of the pushbutton value
12     buttonState = digitalRead(2);
13
14     // check if pushbutton is pressed.
15     if (buttonState == HIGH) // if it is, the buttonState is HIGH
16     {
17         digitalWrite(5, HIGH); // turn LED on
18     }
19     else // if it is not pressed, the buttonState is LOW
20     {
21         digitalWrite(5, LOW); // turn LED off
22     }
23     delay(10); // Delay a little bit to improve simulation performance
24 }
```

Παράδειγμα 6α: Fade Amount

45



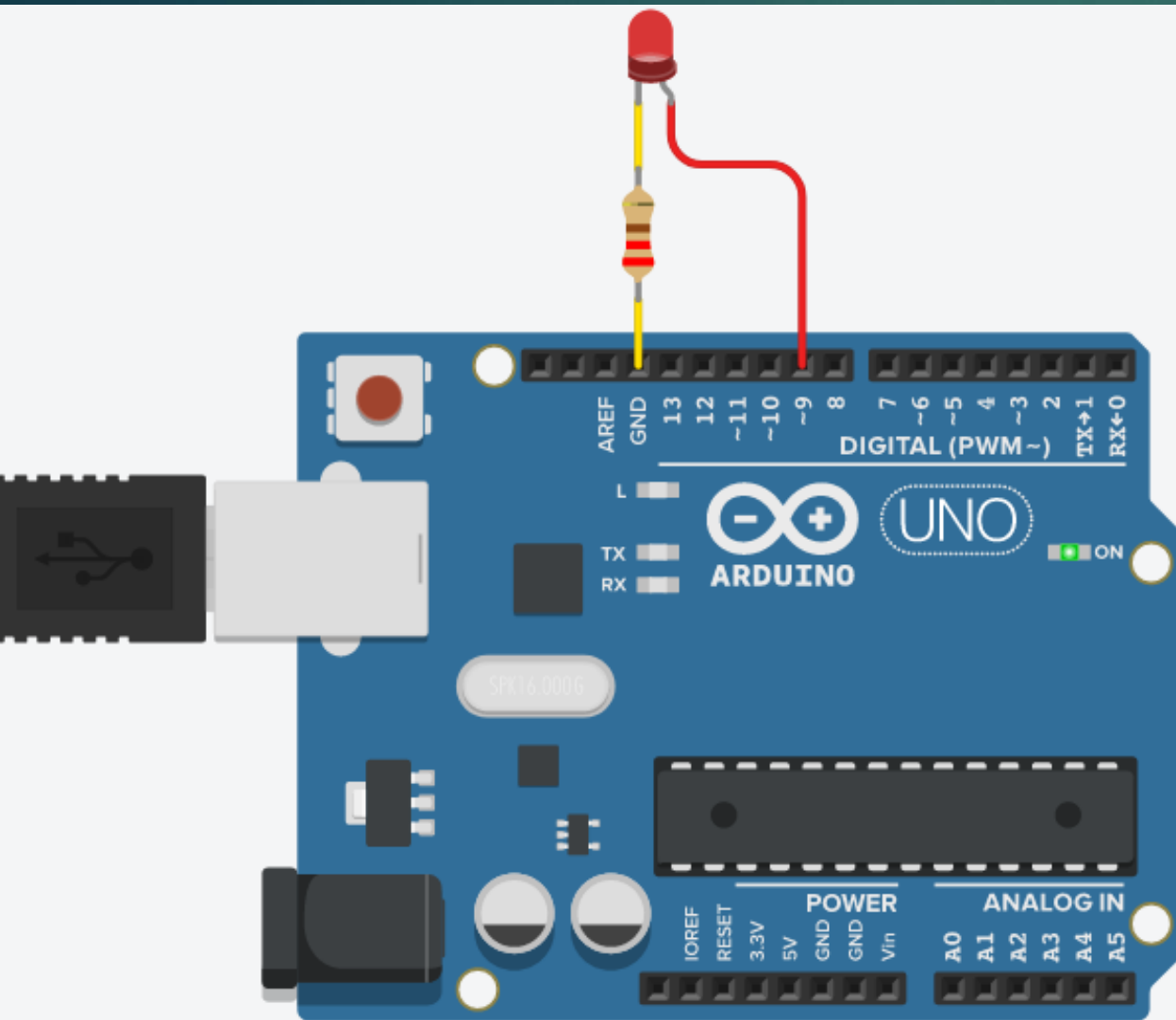
Text

1 (Arduino Uno R3)

```
1 int led = 9;           // the PWM pin the LED is attached to
2 int brightness = 0;   // how bright the LED is
3 int fadeAmount = 5;   // how many points to fade the LED by
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7   // declare pin 9 to be an output:
8   pinMode(led, OUTPUT);
9 }
10
11 // the loop routine runs over and over again forever:
12 void loop() {
13   // set the brightness of pin 9:
14   analogWrite(led, brightness);
15
16   // change the brightness for next time through the loop:
17   brightness = brightness + fadeAmount;
18
19   // reverse the direction of the fading at the ends of the fade:
20   if (brightness <= 0 || brightness >= 255) {
21     fadeAmount = -fadeAmount;
22   }
23   // wait for 60 milliseconds to see the dimming effect
24   delay(60);
25 }
26
```

Παράδειγμα 6b: Fade Amount

46

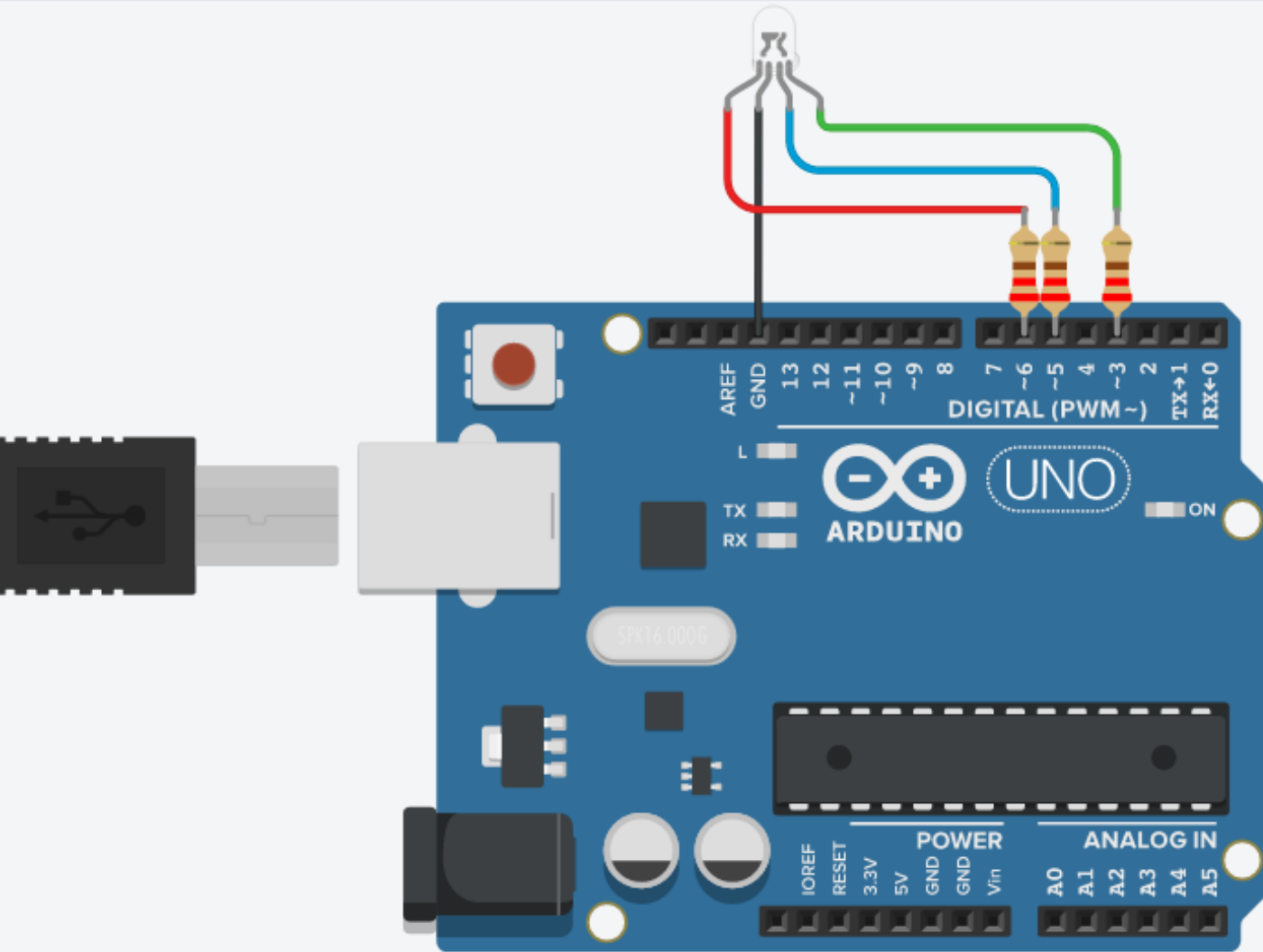


```
1  /*
2  This example shows how to fade an LED on pin 9
3  using the analogWrite() function.
4  The analogWrite() function uses PWM, so if you
5  want to change the pin you're using, be sure to
6  use another PWM capable pin. On most Arduino,
7  the PWM pins are identified with a "~" sign,
8  like ~3, ~5, ~6, ~9, ~10 and ~11.
9  */
10 int brightness = 0;
11 void setup()
12 {
13   pinMode(9, OUTPUT);
14 }
15 void loop()
16 {
17   for (brightness = 0; brightness <= 255; brightness += 5)
18   {
19     analogWrite(9, brightness);
20     delay(30); // Wait for 30 millisecond(s)
21   }
22   for (brightness = 255; brightness >= 0; brightness -= 5)
23   {
24     analogWrite(9, brightness);
25     delay(30); // Wait for 30 millisecond(s)
26   }
27 }
```

Serial Monitor

Παράδειγμα 7α: RGB Led

47



Text

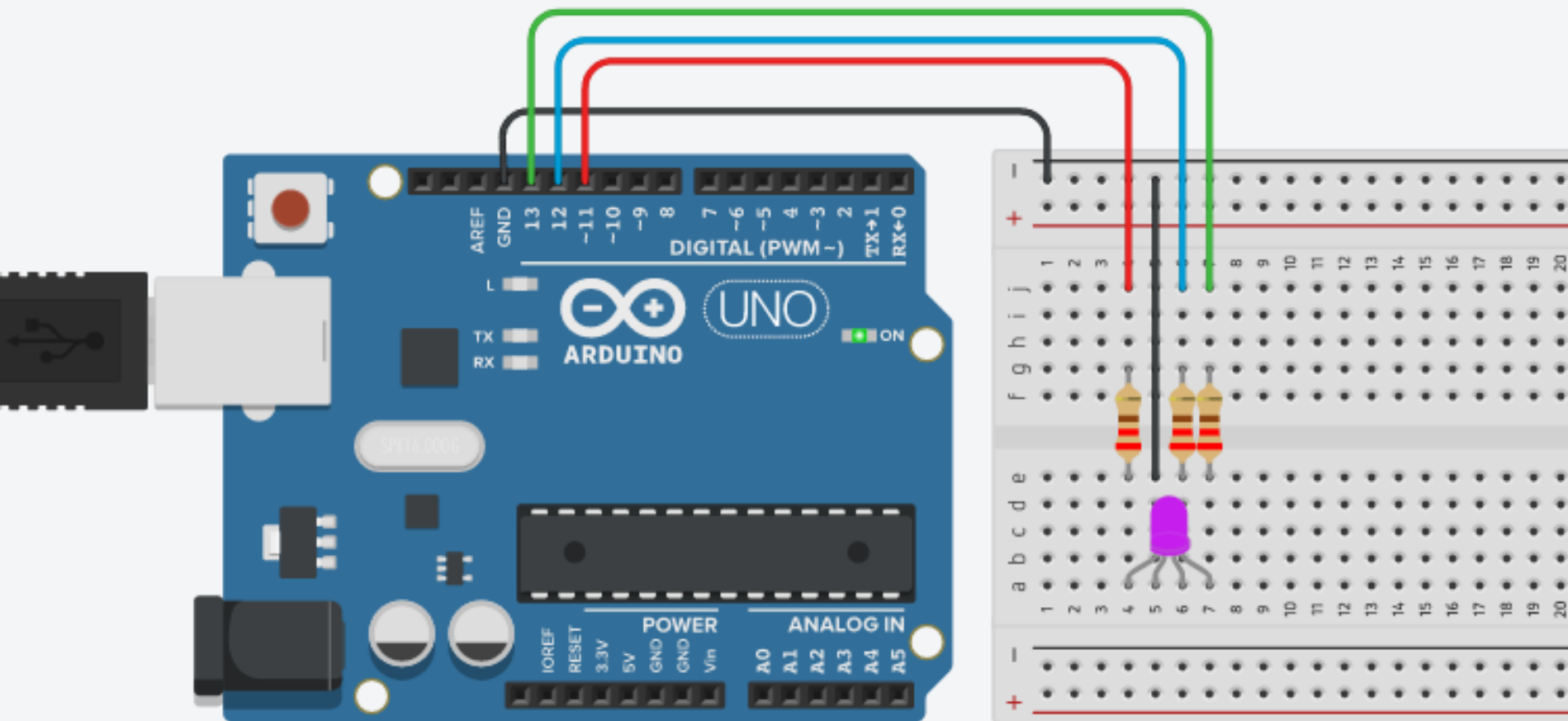


1 (Arduino Uno R3)

```
1 int led1 = 3;           // the pin that the LED1 is attached to
2 int led2 = 5;           // the pin that the LED2 is attached to
3 int led3 = 6;           // the pin that the LED3 is attached to
4 // the setup routine runs once when you press reset:
5 void setup() {
6   // declare pin 3,5,6 to be an output:
7   pinMode(led1, OUTPUT);
8   pinMode(led2, OUTPUT);
9   pinMode(led3, OUTPUT);
10 }
11
12 // the loop routine runs over and over again forever:
13 void loop() {
14   int i,j,k=0;
15
16   // analogWrite(led3,255);
17   // delay(2000);
18
19   for(i=0;i<10;i++)
20   {
21     analogWrite(led1, i*25);
22     for(j=0;j<10;j++)
23     {
24       analogWrite(led2, j*25);
25       for(k=0;k<10;k++)
26       {
27         analogWrite(led3, k*25);
28         delay(10);
29       }
30     }
31   }
32 }
```

Παράδειγμα 7b: Με breadboard

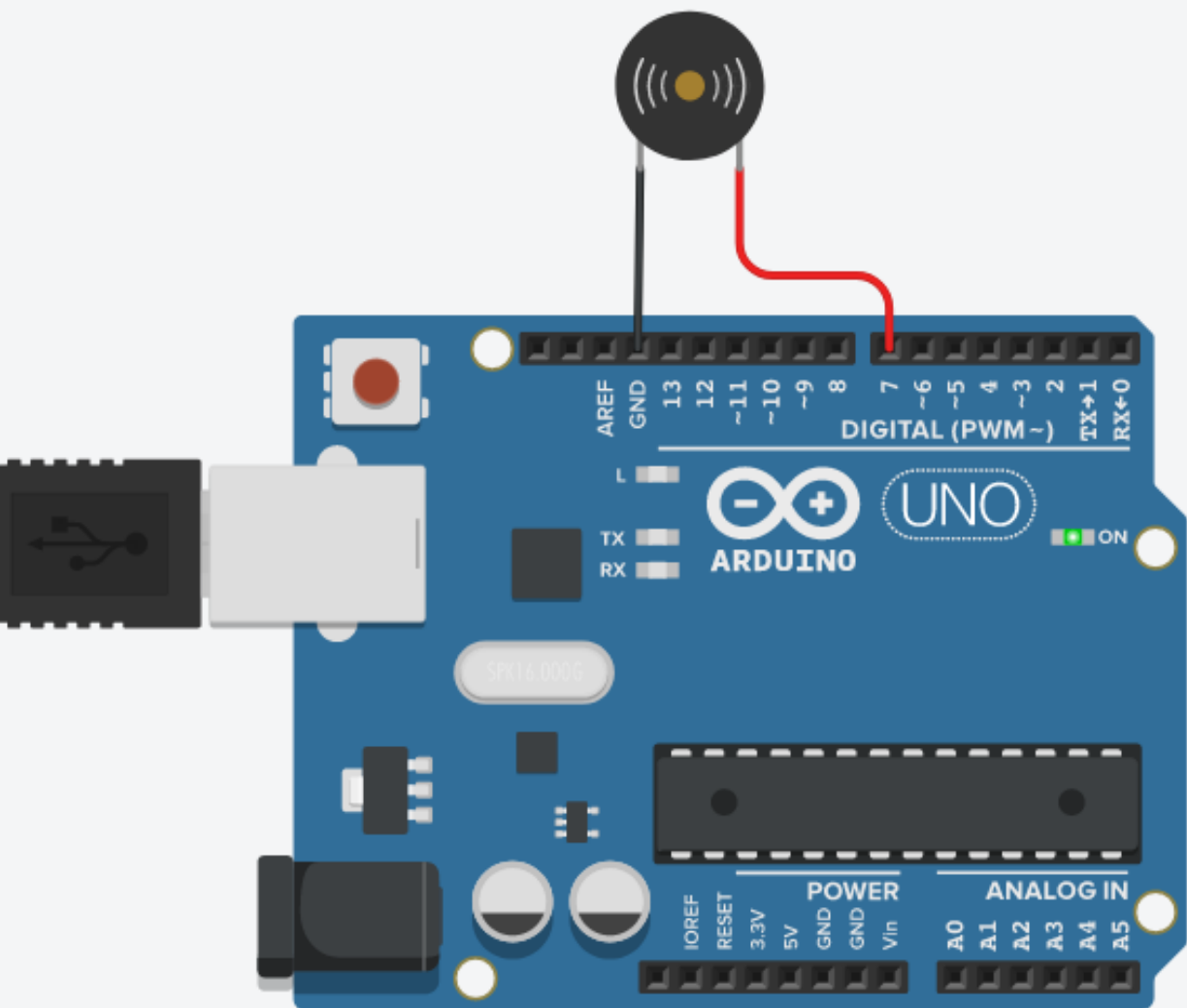
48



```
1 #define LEDR 11
2 #define LEDG 13
3 #define LEDB 12
4
5 void setup()
6 {
7   pinMode(LEDR, OUTPUT);
8   pinMode(LEDG, OUTPUT);
9   pinMode(LEDB, OUTPUT);
10 }
11
12 int r = 0;
13 int g = 0;
14 int b = 0;
15
16 void loop()
17 {
18   r = random(0, 255);
19   g = random(0, 255);
20   b = random(0, 255);
21   analogWrite(LEDR, r);
22   analogWrite(LEDG, g);
23   analogWrite(LEDB, b);
24   delay(1000);
25 }
```

Παράδειγμα 8α: Buzzer (Piezo)

49

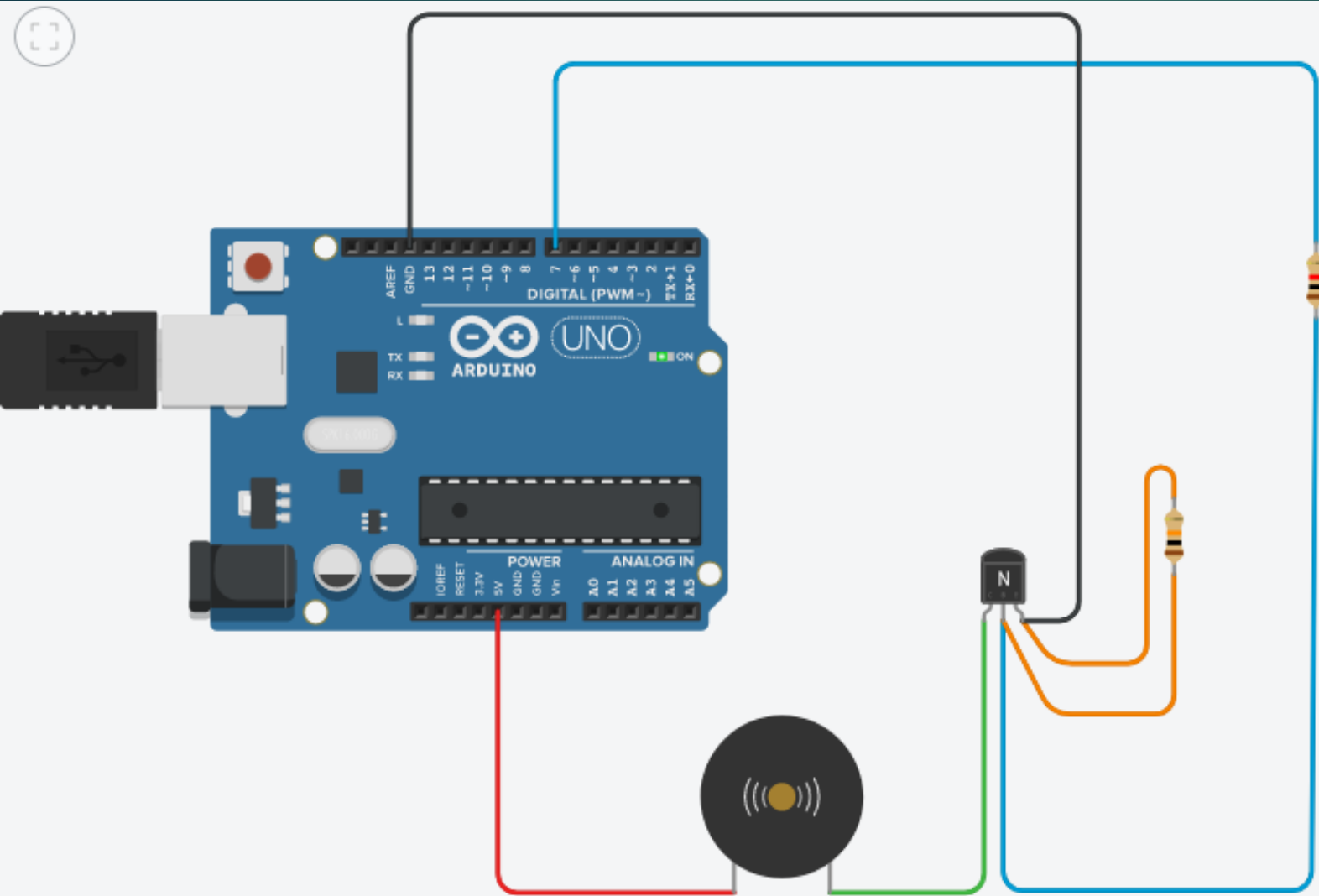


```
1 void setup()  
2 {  
3   pinMode(7, OUTPUT); // set pin 7 as output  
4 }  
5  
6 void loop()  
7 {  
8   digitalWrite(7,HIGH); // make the buzzer sound  
9   delay(1000); // wait for 1sec  
10  
11  digitalWrite(7,LOW); // set the buzzer off  
12  delay(1000); //wait for 1sec  
13 }
```

Serial Monitor

Παράδειγμα 8b: Buzzer (Piezo)

50

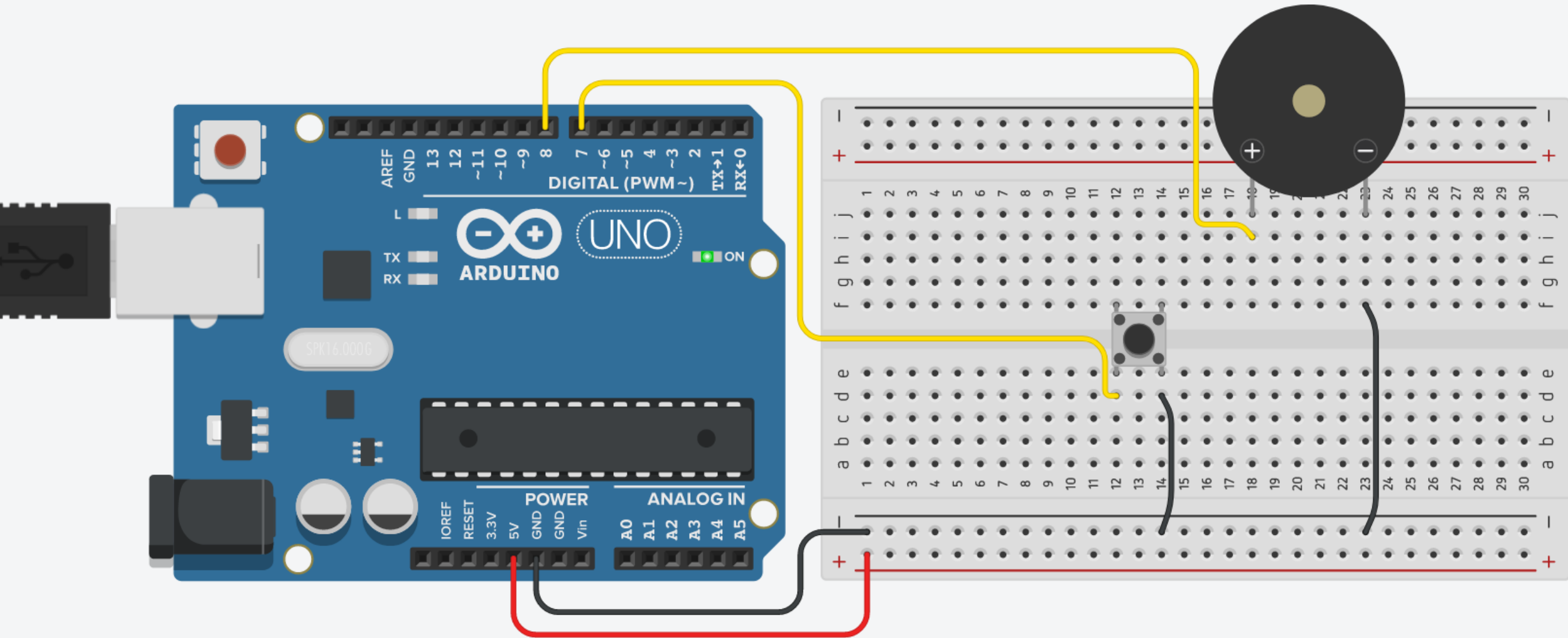


```
1 void setup()
2 {
3   pinMode(7, OUTPUT); // set pin 7 as output
4 }
5
6 void loop()
7 {
8   digitalWrite(7,HIGH); // make the buzzer sound
9   delay(1000); // wait for 1sec
10
11  digitalWrite(7,LOW); // set the buzzer off
12  delay(1000); //wait for 1sec
13 }
```

Serial Monitor

Παράδειγμα 8c: Buzzer with button

51



Παράδειγμα 8c: Buzzer with button

52

```
active_buzzer_with_button

int buzzerPin = 8;
int buttonPin = 7;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  int buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {
    digitalWrite(buzzerPin, HIGH);
  }

  if (buttonState == HIGH) {
    digitalWrite(buzzerPin, LOW);
  }
}
```

Χρησιμοποιήσαμε την ενσωματωμένη **pull-up** αντίσταση που διαθέτει η πλακέτα μέσω της εντολής INPUT_PULLUP.

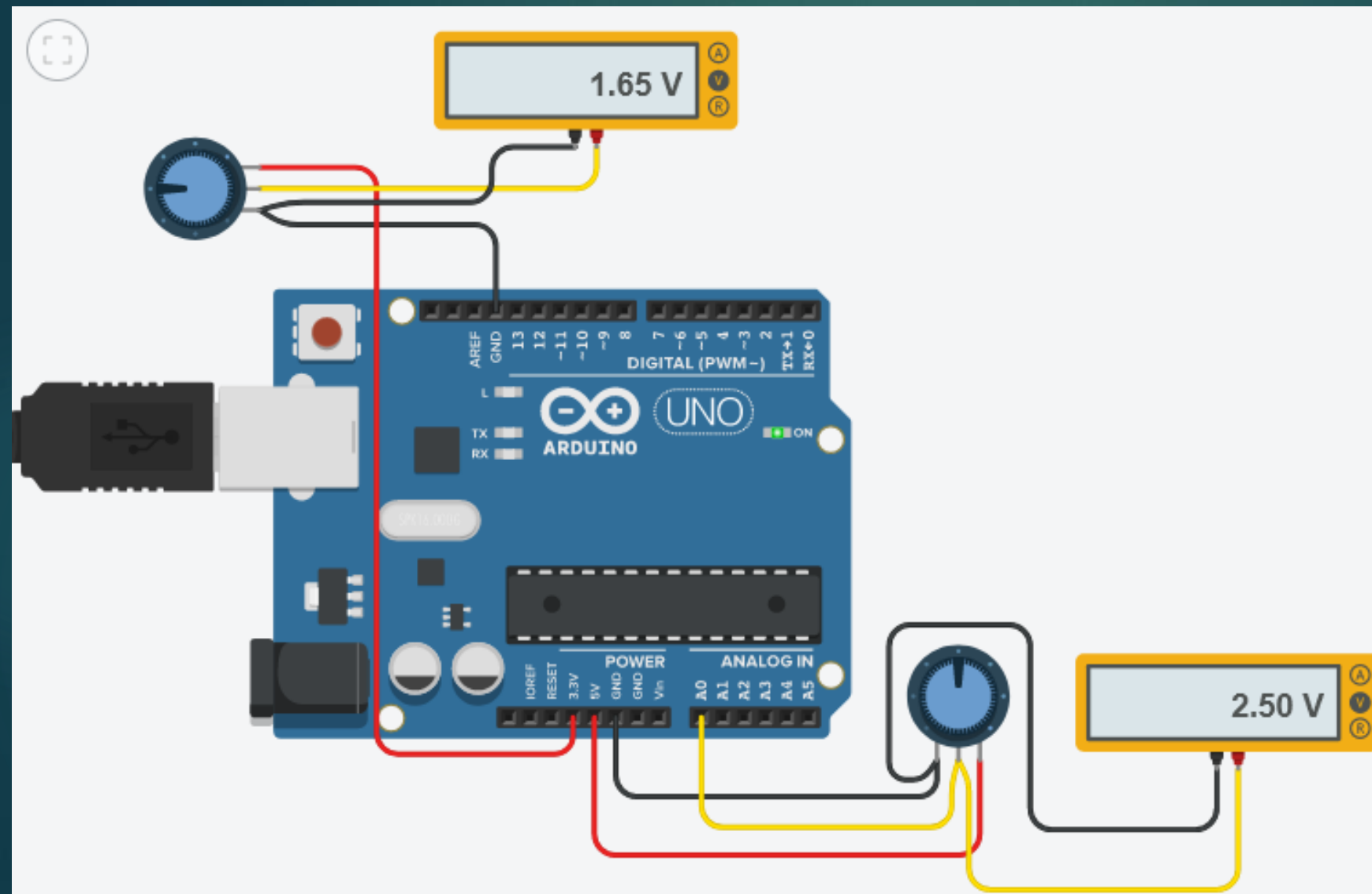
Επομένως η αντίσταση συνδέεται στα 5V.

Δηλαδή όταν δεν είναι πατημένο το button βρίσκεται στην κατάσταση HIGH λόγω της pull-up αντίστασης. Άρα εμείς πρέπει να ορίσουμε σε αυτή την περίπτωση το buzzer να είναι στην κατάσταση LOW (χωρίς ήχο) αφού δεν είναι πατημένο το button.

Αντίθετα όταν το button είναι πατημένο βρίσκεται στην κατάσταση LOW και το buzzer θέλουμε να ηχήσει (HIGH) αφού θα είναι πατημένο το button.

Παράδειγμα 9:

53



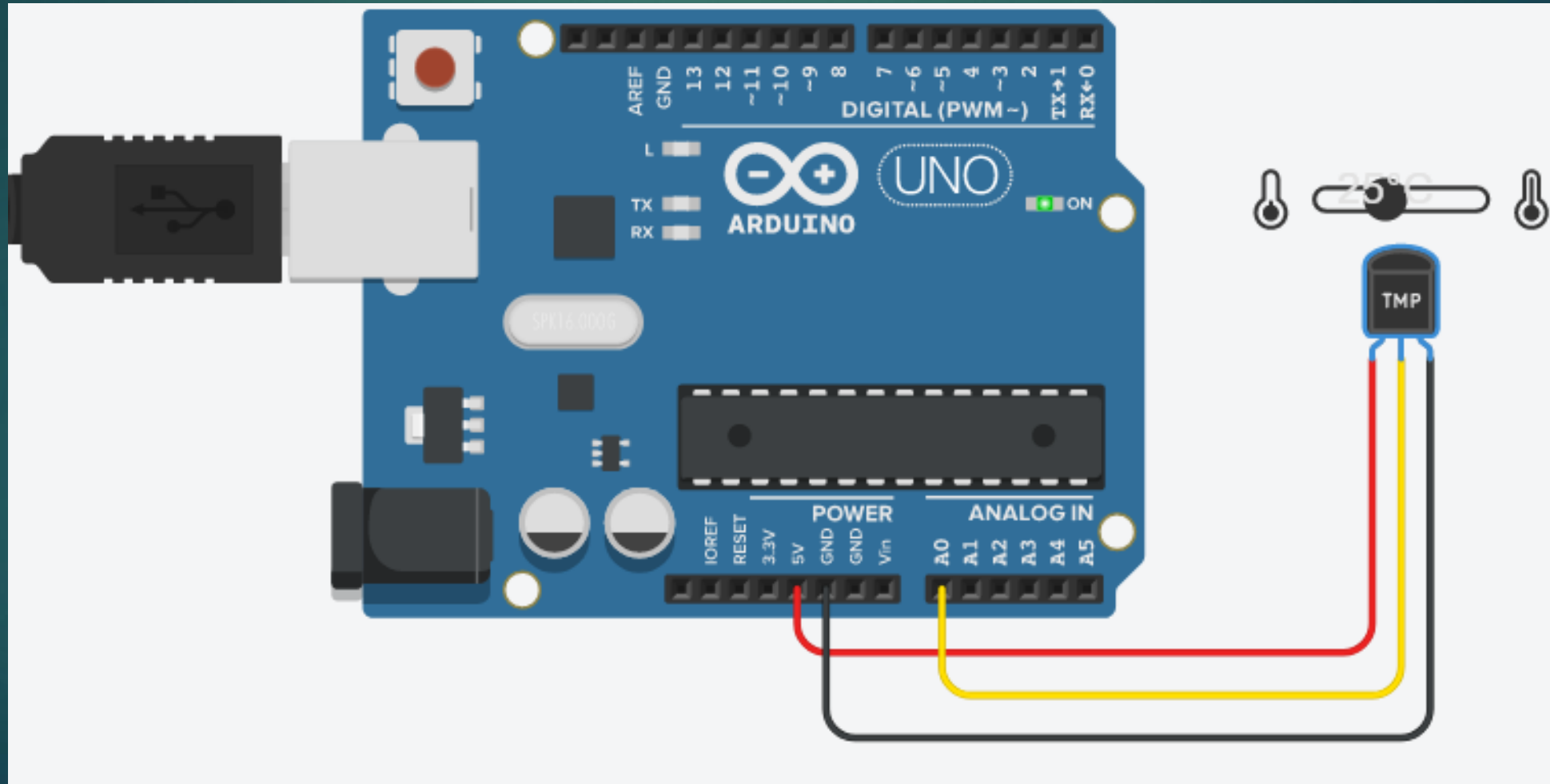
```
1 void setup()  
2 {  
3   Serial.begin(9600);  
4 }  
5  
6 void loop()  
7 {  
8   int sensorValue = analogRead(A0);  
9   Serial.println(sensorValue);  
10 }
```

Serial Monitor

511
511
511
511
511
511
511
511

Παράδειγμα 10: Αισθητήριο θερμοκρασίας

54



Παράδειγμα 10:

55

```
1  /*
2   Temperature Measurement
3   Reads an analog input on A0 (output of TMP36)
4  */
5  float powervoltage = 5; //Define power supply voltage
6
7  void setup()
8  {
9   //Initialize serial communication at 9600 bits per second:
10  Serial.begin(9600);
11  }
12
13  void loop()
14  {
15   float temperature;
16   //Read the input on analog pin 0 (A0):
17   float sensorValue = analogRead(A0);
18   //Print out the value you read:
19   Serial.println(sensorValue);
20   delay(1000); //Wait for 1sec
21   //Calculate the temperature (degC):
22   //0.5 is an offset that should be subtracted
23   //multiplied with 100 for conversion to degrees
24   temperature = (((sensorValue/1023)*powervoltage) - 0.5) * 100;
25   Serial.print("The room temperature degree is: ");
26   Serial.println(temperature);
27   delay(10); //delay in between reads for stability
28  }
```



Serial Monitor

```
The room temperature degree is: 44.82
123.00
The room temperature degree is: 10.12
127.00
The room temperature degree is: 12.07
145.00
The room temperature degree is: 20.87
153.00
The room temperature degree is: 24.78
153.00
The room temperature degree is: 24.78
```