BE MAKER

BE MAKER

ELETTRONICA, ROBOTICA E CODING PER RAGAZZI... E NON SOLO !

COURSE OF ELECTRONICS, ROBOTICS AND CODING FOR CHILDREN... and not only!

BASIC COURSE – Lesson 2

Index

Warnings

With regard to the safety aspects, since the projects are based on a very low voltage power supply supplied by the USB port of the PC or by support batteries or power supplies with a maximum of 9V output, there are no particular risks of an electrical nature. It is however necessary to specify that any short circuits caused during the exercise phase could produce damage to the PC, to the furnishings and in extreme cases even to burns, for this reason every time a circuit is assembled, or changes are made on it, it will be necessary to do so in the absence of power and at the end of the exercise it will be necessary to provide for the disconnection of the circuit by removing both the USB cable connecting to the PC and any batteries from the appropriate compartments or external power connectors. In addition, always for safety reasons, it is strongly recommended to carry out projects on insulating and heat-resistant carpets that can be purchased in any electronics store or even on specialized websites.

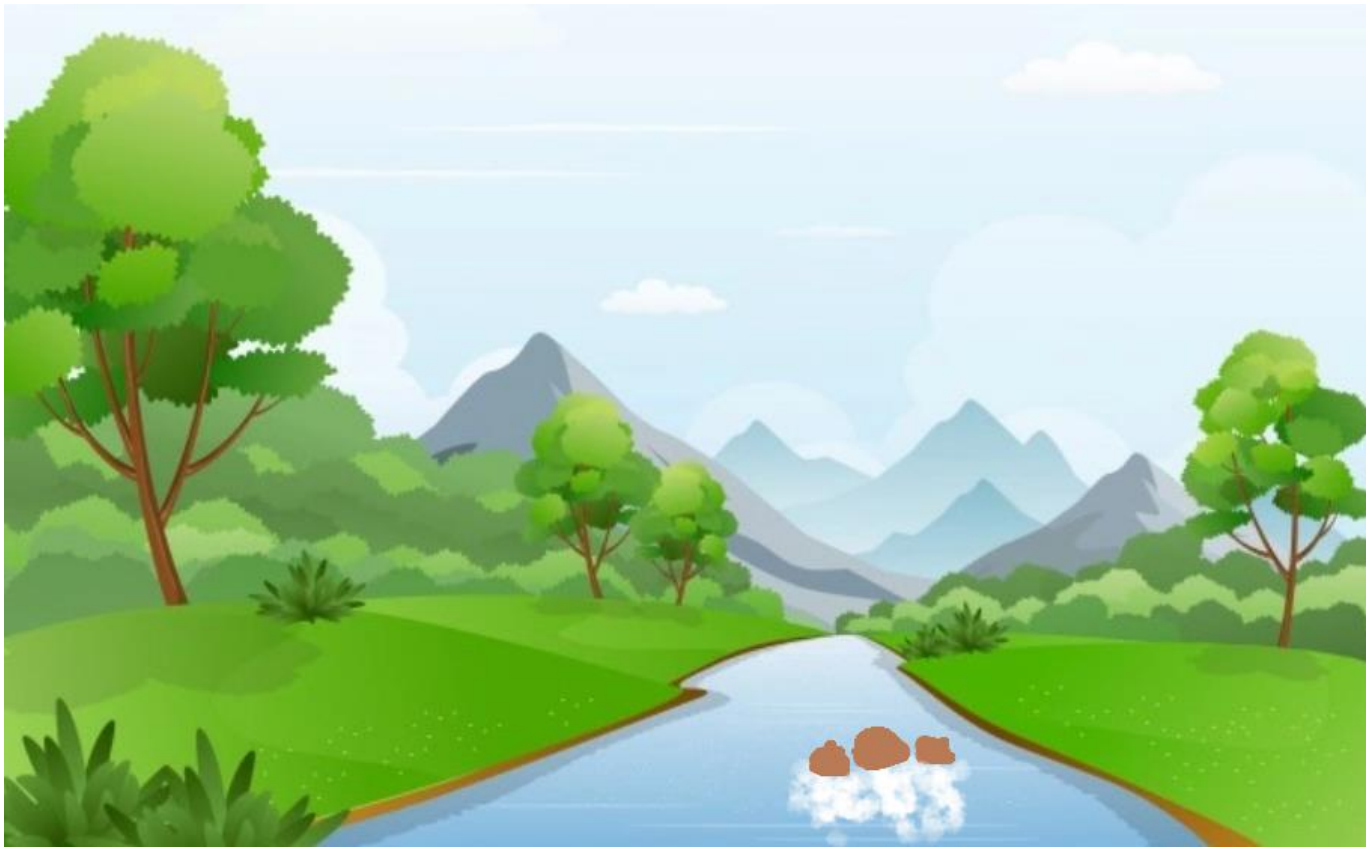
At the end of the exercises it is advisable to wash your hands, as the electronic components could have processing residues that could cause damage if ingested or if in contact with eyes, mouth, skin, etc. Although the individual projects have been tested and safe, those who decide to follow what is reported in this document, assume full responsibility for what could happen in the execution of the exercises provided for in the same. For younger children and / or the first experiences in the field of Electronics, it is advisable to perform the exercises with the help and in the presence of an adult.

Notes sul Copyright

All trademarks are the property of their respective owners; third-party trademarks, product names, trade names, corporate names and companies mentioned may be trademarks owned by their respective owners or registered trademarks of other companies and have been used for purely explanatory purposes and for the benefit of the owner, without any purpose of violation of the copyright rights in force. What is reported in this document is the property of Roberto Francavilla, Italian and European laws on copyright are applicable to it – any texts taken from other sources are also protected by the Copyright and property of the respective Owners. All the information and contents (texts, graphics and images, etc.) reported are, to the best of my knowledge, in the public domain. If, unintentionally, material subject to copyright or in violation of the law has been published, please notify info@bemaker.org by email *and I will promptly remove it.*

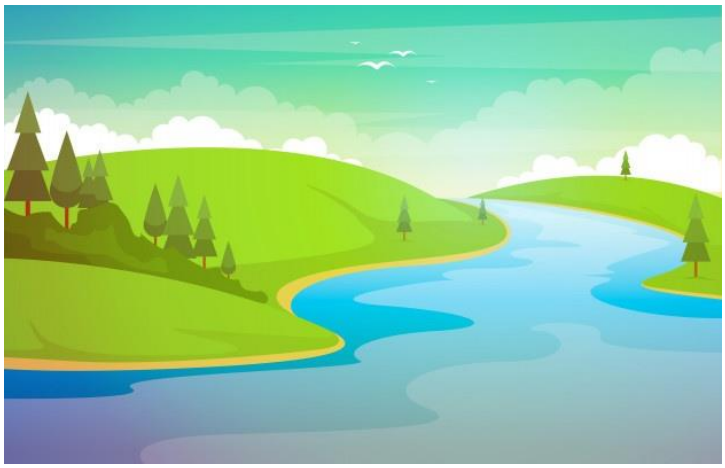
Roberto Francavilla

Insights on Electrical Resistance



We saw in the previous lesson, referring to a watercourse, that electrical resistance can be defined as the "obstacle" that water, or electric current, encounters during its natural path in the riverbed, or electrical cables.
Obviously it is easy to understand how this difficulty of the current of water (but also of the electric current) depends on several factors, namely the length of the path (length of the electrical cables), the width of the riverbed (section of the cables) and the type of land that makes up the riverbed (from the material used for electrical conductors, if copper, made of aluminum, or other material).

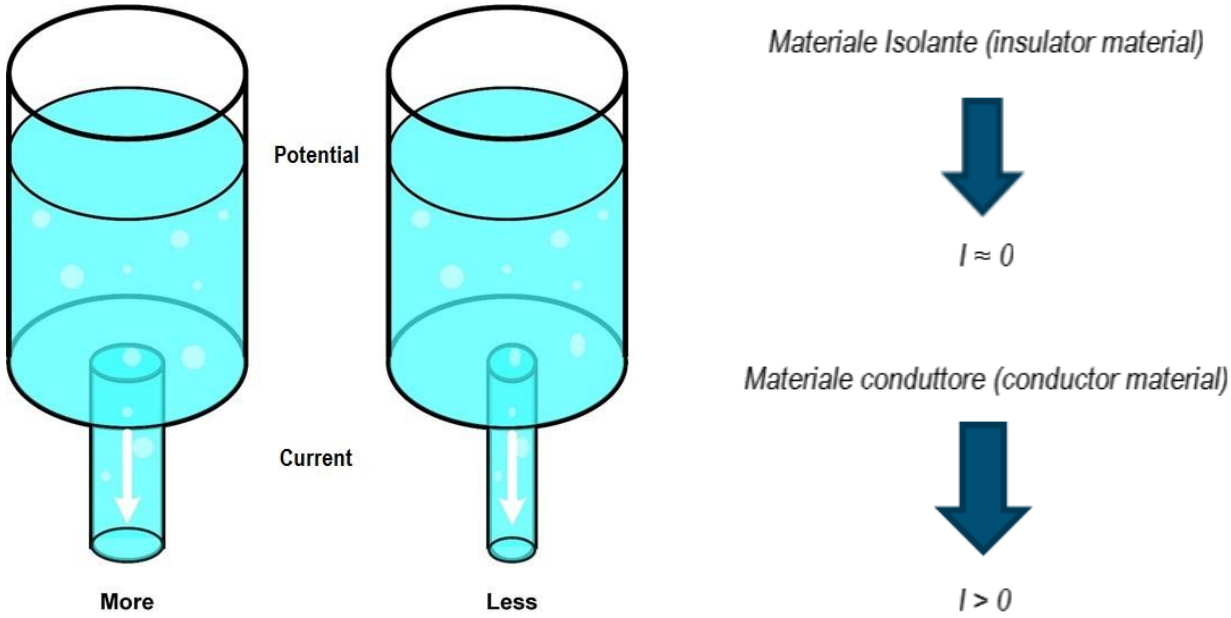
In fact, the longer the path of the river, the higher the resistance; the larger the section of the riverbed (of the electrical cables) and the more easily the water flows inside it, as a result of which the resistance is lower; if the riverbed consists of fine and compact sand or a well-polished surface, it will offer a lower resistance to water than a riverbed composed of irregular rocks, in



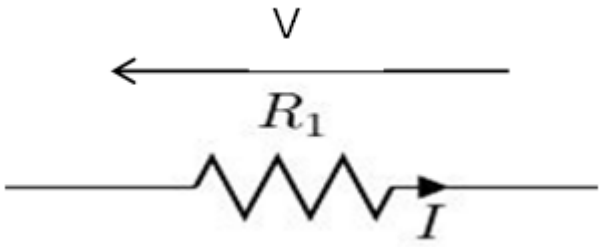
the same way there are materials that have properties such as to be crossed by the electric current better than others.

For example, gold is an excellent electrical conductor, as well as silver, copper, aluminum... (all metals are good conductors), while there are materials that hinder the passage of electric current to the point of being called "insulators" such as plastic, rubber,...

Another way to visualize the electrical resistance in order to fully understand its physical principle is the one represented in the figure below:



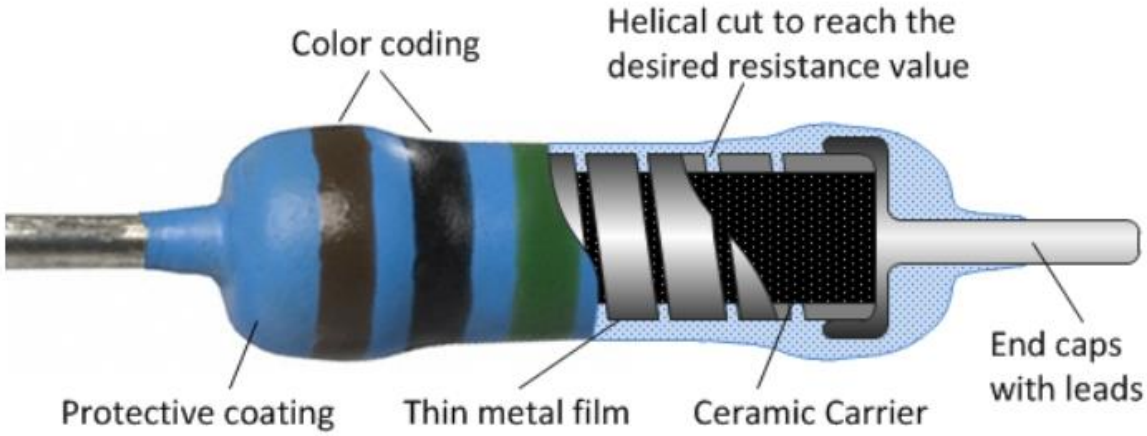
The amount of water contained in the two tanks is initially the same and the water level represents the electric potential, that is, the force with which the water is pushed out of the pipe. The narrower the outlet pipe, the more difficult the water will have to get out.
Taking this speech to the extreme, if we were to have such a small tube (imagine a hair wide) it is understandable that the water will have a lot of difficulty getting out. The pipe that represents the electrical conductor, if it were a hair, will obviously almost totally prevent the exit of water, in which case the material constituting the conductor is called ***insulator*** and therefore the current that runs through this conductor is practically zero.



The electrical resistance, the symbol of which is a zigzag line, is indicated by the capital letter "R", then you put a subscript "1", "2",... especially when there are several electrical resistances of different value, the electric current that runs through it is indicated with the "I", while the electrical voltage at the ends of the electrical resistance is indicated with a "V" (remember the waterfall seen in Lesson 1 .... the water falls from the highest point "+" to the lowest point "-") .
Note that the voltage V conventionally indicates an arrow opposite to the direction of the current and the tip of the voltage arrow corresponds to the "+". The current, always conventionally, goes from the positive "+" pole to the negative "-" pole.
On the market, electrical resistances are easily available components and are found of different types also depending on the currents they must be able to withstand. In reality, the seller must be indicated the value of the resistance you want and the electrical power (the current is not indicated). Electrical power is another important electrical quantity in the field of electrical engineering and in the case of continuously powered circuits, as in our case, it is indicated with P and is measured in Watts (electrical symbol "W"). The Electric Power is calculated by making the product between the value of the voltage applied to the resistance and that of the current that passes through it, but it is also equal to the product of the electrical resistance and the current value that crosses it raised to the square, that is:

$$P = V \; x \; I = R \; x \; I^2$$

For our projects we will generally refer to electrical resistance values ranging from a few hundred Ohms up to a few tens of miles of Ohms and we will use the resistance to adequately distribute the electrical voltage generated by Arduino, or the 5 V, in such a way as to make our components work better in the various electronic circuits that we will develop. Also for the Ohms, as for other units of measurement of physical quantities, there are multiples and submultiples. However, we will always be in the field of multiples, that is, we will often use the "kOhm" (we read chiloOhm) [it is a multiple of the Ohm, that is, 1 kOhm corresponds to 1000 Ohm]. So when I talk about tens of miles of ohms, I'm talking about tens of kOhms. The values of current circulating in the circuits that we normally realize are very low, of the order of milli-Ampere (mA). From this you understand very well that the electrical powers at stake are very low of the order of 0.25 W or at most 0.5 W (on the market, electrical resistances capable of withstanding powers of this type are indicated with 1/4 watt or with 1/2 of watt).
The figure below shows how the resistance we use in our circuits is made.



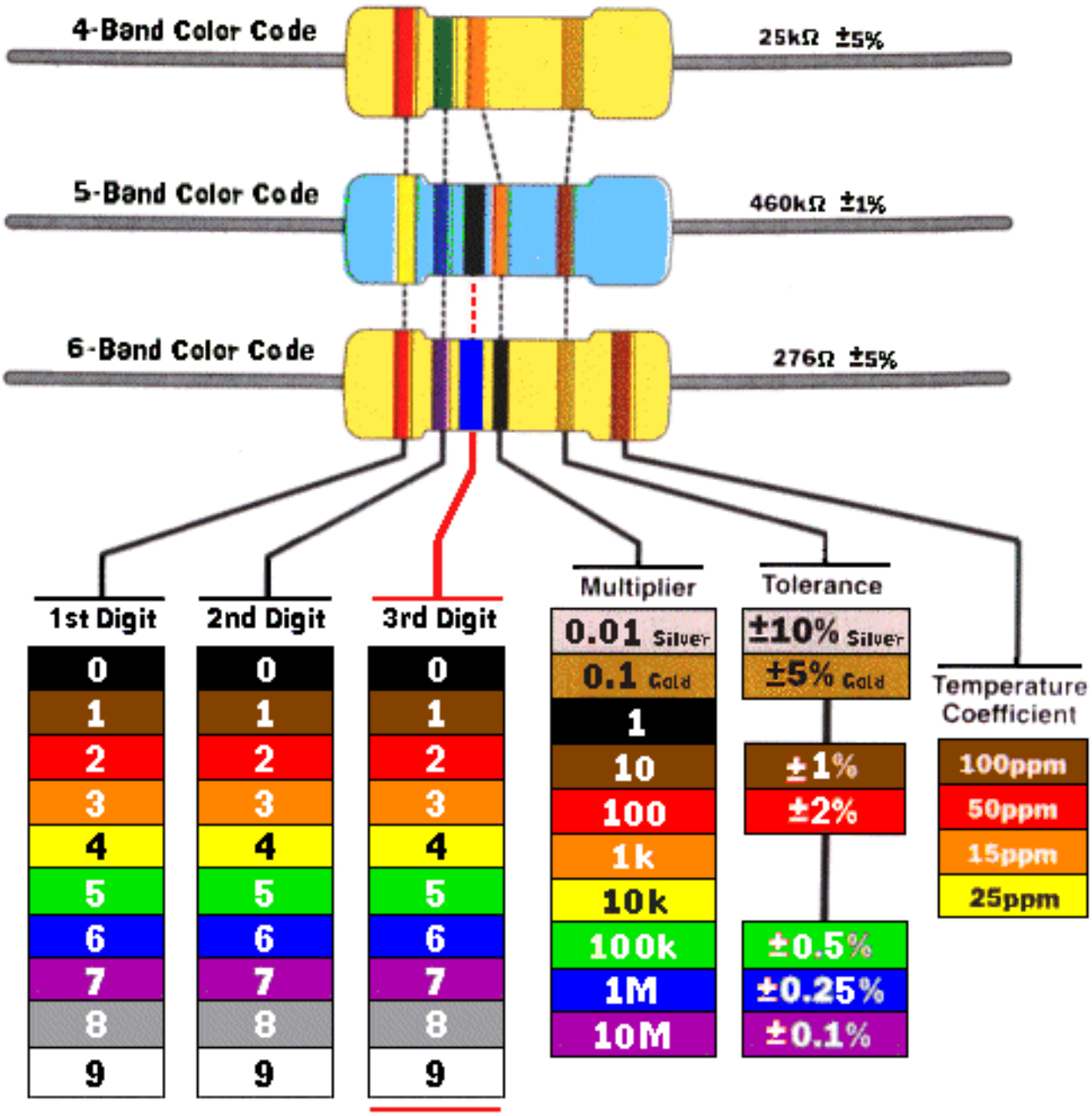
There is a ceramic support on which a wire of a particular conductive material is wound that has its own electrical resistance, the number of revolutions represents its length and therefore the total resistance. The conductor wire ends at the ends on two platelets with two metal terminals that are the output connectors. Everything is then covered with other ceramic material. The ceramic material is insulating and therefore in addition to mechanically protecting the resistance, it prevents the creation of short circuits.

How to read the value of a resistance

The resistance value is indicated on the outer casing by means of colored circles or colored bands.

Let's say right away that there are several methods used to indicate the value of the resistance, among the most widespread, of which standards have been created, are the indication of the value with 4 circles in total and the one with 5 colored circles in total, the 6-circle one is less widespread, but still used above all for laboratory components.

The table below shows the correlation between circle colors and resistance values. I would like to immediately and in any case make you observe that one of the circles is always more spaced than the others (in the case of a 6-circle resistor, the most spaced circles are 2).
The most distant circle indicates the ***tolerance*** *, for the reading of the colors we start from the first circle on the side opposite to the spaced one.*



Resistors with 4 colored circles

The first circle indicates the value of the ***first digit****, the second circle indicates the value of the* ***second digit*** *and the third circle indicates the* ***multiplier****, and finally the fourth circle indicates the* ***tolerance****.*

Let's take an example to understand how to apply the criterion of enhancement of resistance.
Suppose we have a resistance like the one in the figure below:

The first thing we must identify is the most spaced circle that is "gold" in color, this circle represents the tolerance on the resistance value of the component, in this case the tolerance, that is, the uncertainty on the value, is +/- 5%.
Then we go to see the color of the first circle on the opposite side to the most spaced, the color is "brown", that is, the first digit has a value of "1", the second circle is "black", so the second digit has a value of "0" and the third circle is "red", that is a multiplicative factor "x 100".
At this point we know the value of resistance, in fact:

R = 1 , 0, x 100, +/- 5% i.e. 10 x 100 +/- 5% and then R = 1,000 Ohms +/- 5%
Our resistance is one thousand Ohms, that is, one kilo Ohm with a tolerance of 5%.
Since we Makers like mathematics we are also going to calculate, in terms of values, what it means to have a tolerance of 5%:

We calculate 5% of 1,000, that is, 5 x 1,000 and then divide by 100, then we get 50, so:

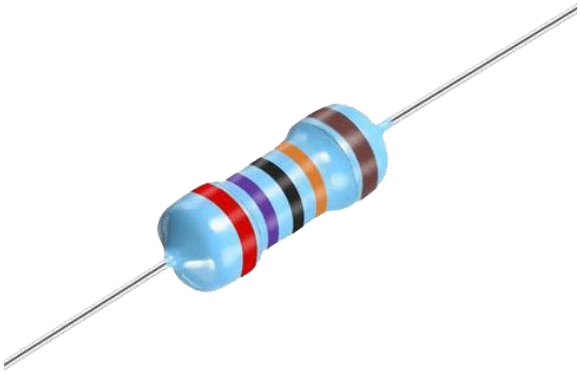
R = 1,000 +/- 5% Ohm means that the value of R is definitely a value between (1,000-50) 950 and (1,000+50) 1,050 Ohms.

Resistors with 5 colored circles

The first circle indicates the value of the ***first digit****, the second circle indicates the value of the* ***second digit****, the third circle indicates the value of the* ***third digit*** *and the fourth circle indicates the* ***multiplier****, and finally the fourth circle indicates the* ***tolerance****.*

Also in this case we make an example to understand well how to apply the criterion of enhancement of resistance.

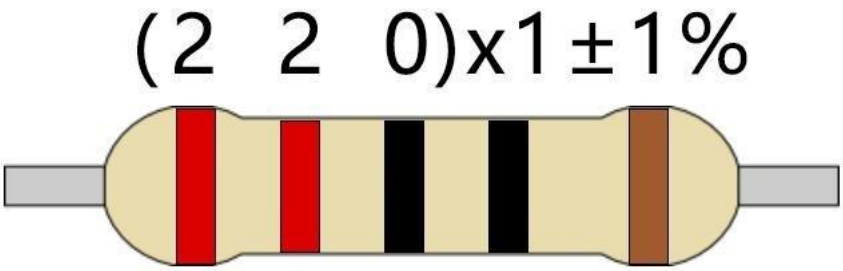
Suppose we have a resistance like the one in the figure below:



Also in this case the first thing we must identify is the most spaced circle that is "brown" in color, this circle represents the tolerance on the resistance value of the component, in this case the tolerance, that is, the uncertainty on the value, is +/- 1%.
After that we go to see the color of the first circle on the side opposite to the most spaced, the color is "red", that is, the first digit has a value of "2", the second circle is "purple", so the second digit has a value of "7", the third circle is "black", so the third digit is "0" and finally the fourth circle is "orange" , that is, a multiplicative factor " 1 K " i.e. "x 1000".
At this point we know the value of resistance, in fact:

R = "2","7","0", x 1,000 +/- 1%, i.e. R = 270,000 Ohms or even R=270 kOhm (kilo Ohms), with a tolerance of 1%, that is: 1 x 270,000 : 100 = 2,700 Ohms. So the true value of the resistance is a number between 267,300 Ohm and 272,700 Ohm.

Since we are not realizing Space Shuttle, for our projects having tolerances of 1%, 5%, but also 10% does not involve making significant errors, then from now on we will consider the tolerance only a qualifying element of the component (that is, we will say that a component with a narrower tolerance is better than one with a wider tolerance), but we will not take this into account in our calculations.
One last example:



The first circle is red, so 2, the second circle is still red and then another 2, the third circle is black, then the third digit is 0 and finally the fourth circle is black, so the multiplier is for 1. So the R = 220 x 1 i.e. 220 Ohms.

Curiosity: What is a Physical Quantity and uncertainty in measurements

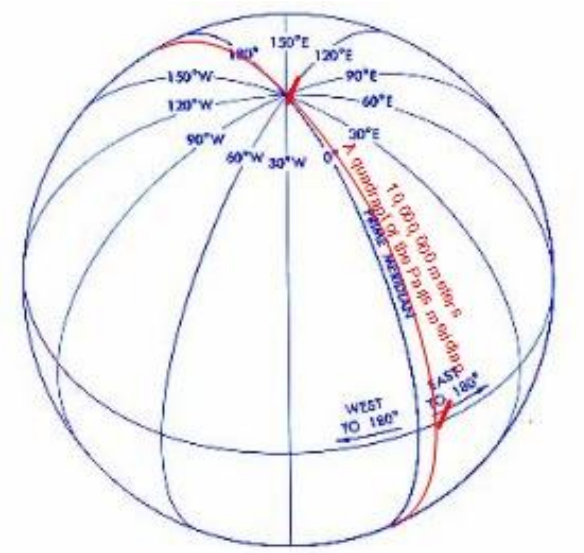
A physical quantity is a property of a body (or phenomenon) that can be measured. To specify the physical quantity you need a number (which indicates its value) and a unit of measurement. The unit of measurement is a reference quantity of a quantity, to which the value of 1 is assigned.

Let's take an example to better understand what I wrote:

One of the fundamental quantities is the **length** to which the **meter [m]** has been assigned as a unit of measurement.

But what is the meter?

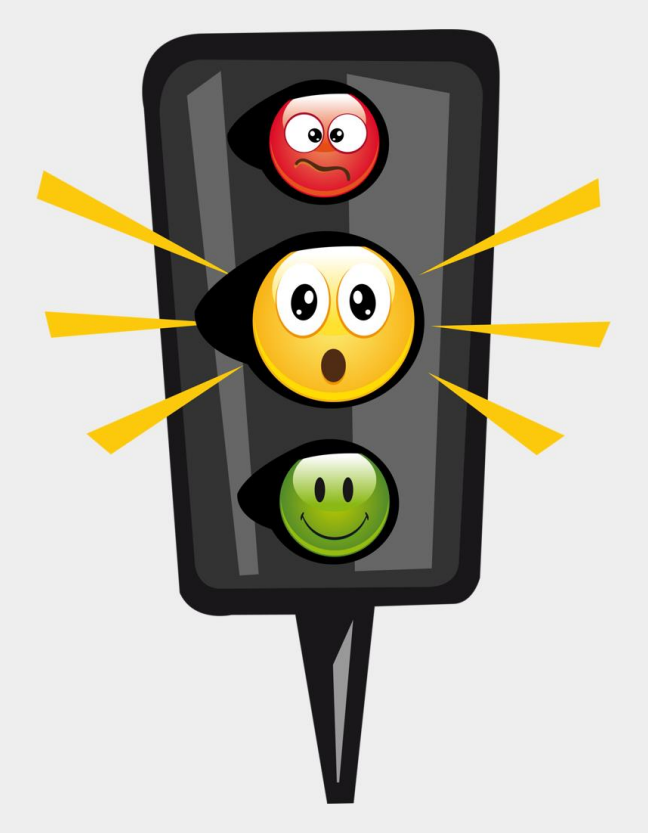
In France in Sèvres is preserved a rod that has a length of 1 in 10 million the length of the arc of the Earth's meridian that starts from the North Pole and passing through Paris, stops at the Equator. This rod, so determined, represents the length of 1 m. Obviously then for comparison the "meters" used when we want to take measurements of an object were reproduced.



In fact, to measure a length, we do nothing but see how many times the meter is contained in the length to be measured.

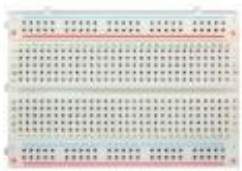
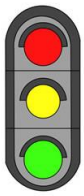
Obviously when making measurements of quantities, there are also errors, which can be instrumental errors due to the imperfect calibration of the instruments, errors in reading the instrument (the so-called parallax error)... we are obviously talking about small errors that are related to the measurement of magnitude. These errors then introduce a small indeterminacy in the measure and therefore you have the so-called **tolerance** in the measure.

Project 5 – The Traffic Light with Arduino

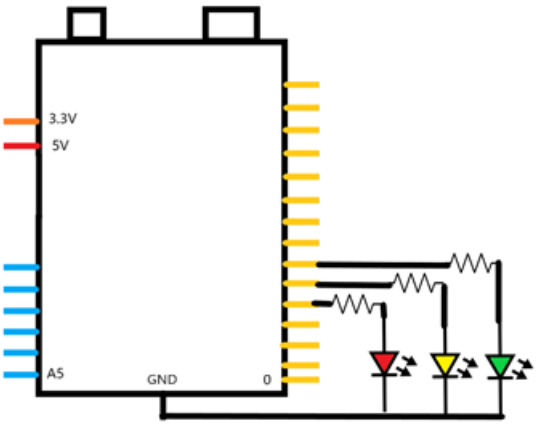


We continue with the use of Arduino digital PINs and create a simple project that will make us have so much fun. We make a traffic light with Arduino.

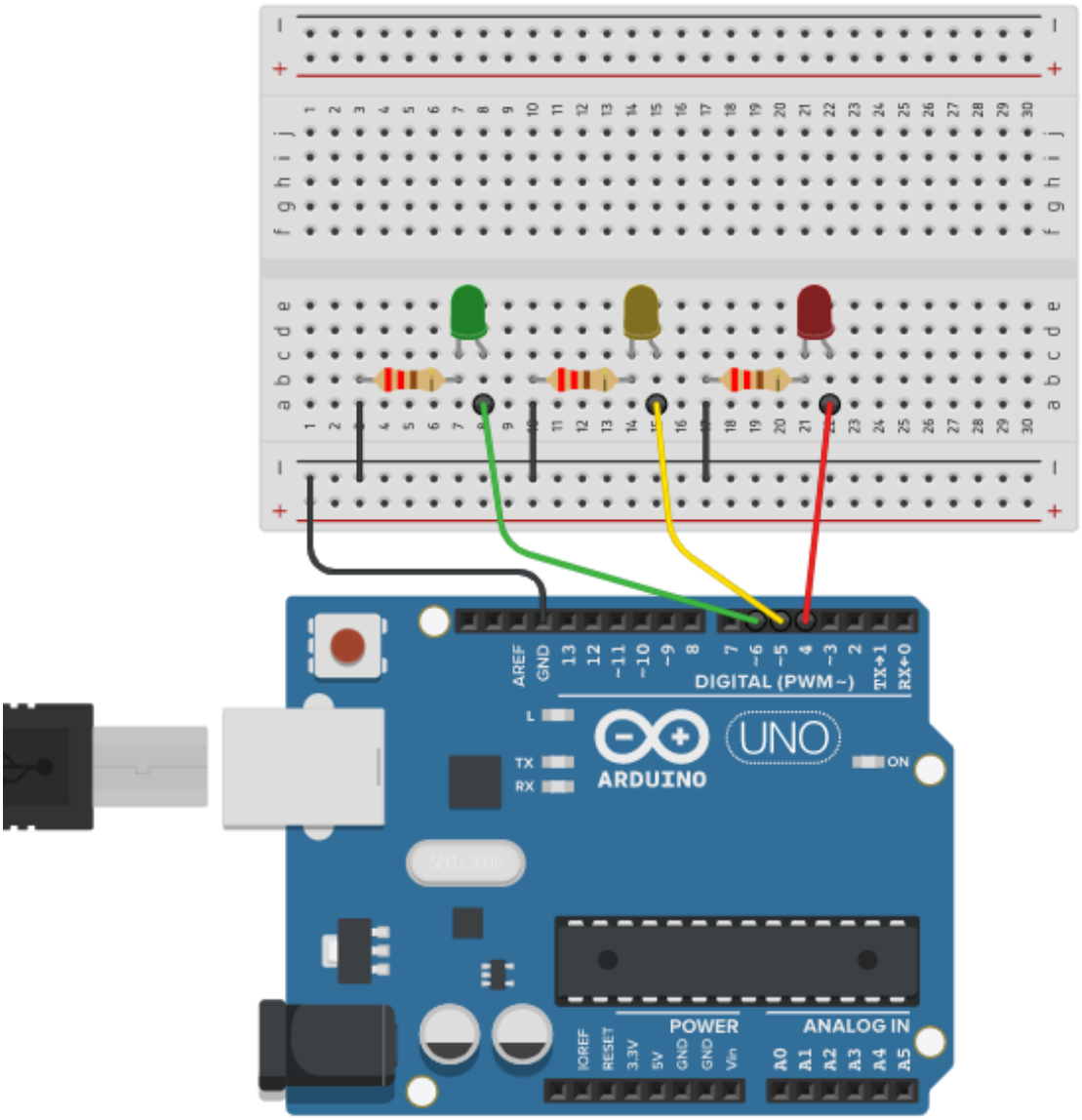
For this project we need:

			
Arduino Uno R3 or compatible	*Breadboard*	*Dupont cables male - male*	*USB connection cable*
			
3 x Resistance 220 Ohm	*3 x LED (Red, Yellow and Green)*	*Satbed Cardboard*	

The wiring diagram is:

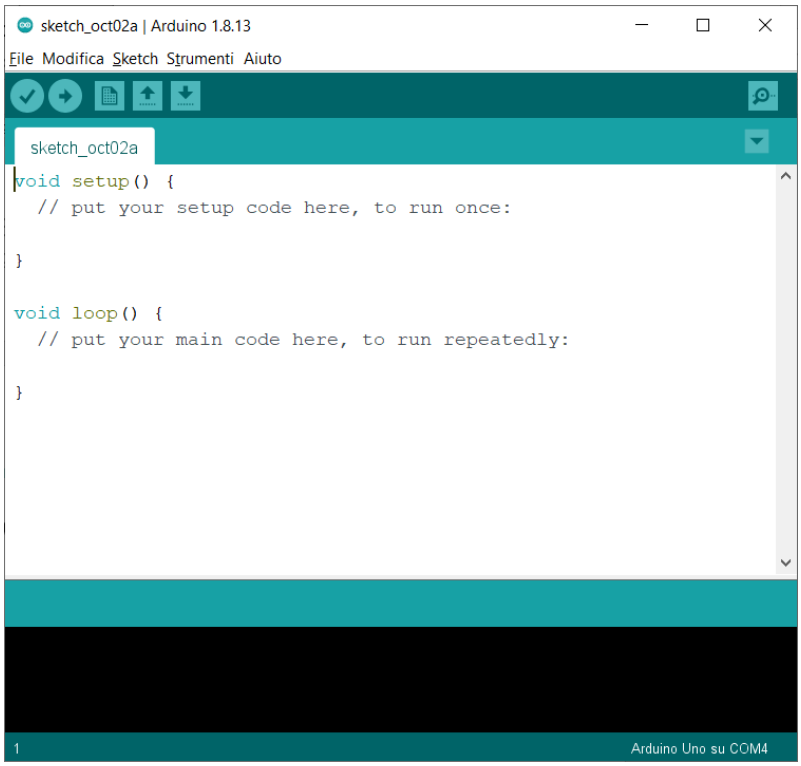


For the assembly follow the diagram below:



After the links we move on to write the sketch.
Connect Arduino to the PC via the USB cable and launch the Arduino IDE application by clicking twice on the relevant icon.

A blank window opens, or you need to open a new empty one:



Type the sketch below:

```
/*
Be Maker School - STEM - Project 5 - Traffic Light
*/
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{
  digitalWrite(4, HIGH); Bright red
  digitalWrite(5, LOW); Yellow off
  digitalWrite(6, LOW); Green off
  delay(6000); Waiting for 6 sec.
  digitalWrite(4, LOW); Red off
```

```
  digitalWrite(5, LOW); Yellow off
  digitalWrite(6, HIGH); Bright green
  delay(4000); Waiting for 4 sec.
  digitalWrite(4, LOW); Red off
  digitalWrite(5, HIGH); Bright yellow
  digitalWrite(6, HIGH); Bright green
  delay(2000); Waiting for 2 sec.
}
```

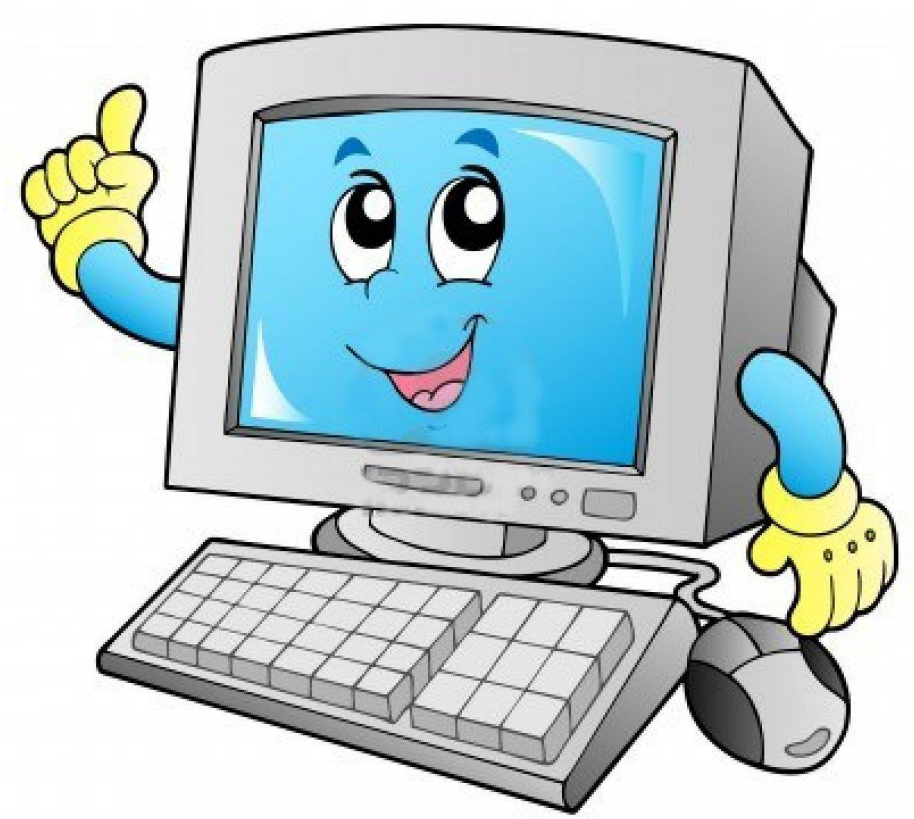
Once you have written the code launch the verification precompilation (check mark), it will ask you to save the sketch (you can change its name) and then click on the arrow for loading:

The result will be: the red LED lights up and stays on for 6 sec, after which the red one turns off and the green LED lights up. After another 4 sec. si also turns on the yellow LED and stays on for 2 sec. After that, turn off the yellow and green LEDs and start again with the red LED turned on.

If you place the cardboard with the traffic light, previously printed and cropped, it will certainly give you a nice effect.

*[For the Video-project click **here**]*

Introduction al Coding



It's time to enter the world of Coding and the first thing I want to introduce you to is the sketch. But first let's take a step back to understand what we mean by Coding.

The term Coding was born a few years ago and as the name suggests derives from "coding" or generating codes, which in turn, in computer terms, refers to Computer Programming (or more generally: of electronic devices) using a particular programming language.

So, in summary, the term coding refers to computer programming and therefore to the conception and development of software.

Said so... you will tell me, bravo you copied the definition from the book of the first year of high school computer science ....

.... ma really: what does it mean to Program a Computer or an electronic device????

So I try to explain better... and to explain it, I do not want to start from the birth of Computers, but from a concrete example.

Suppose we have an industrial process that we want to automate, for example the production of corks for wine bottles...

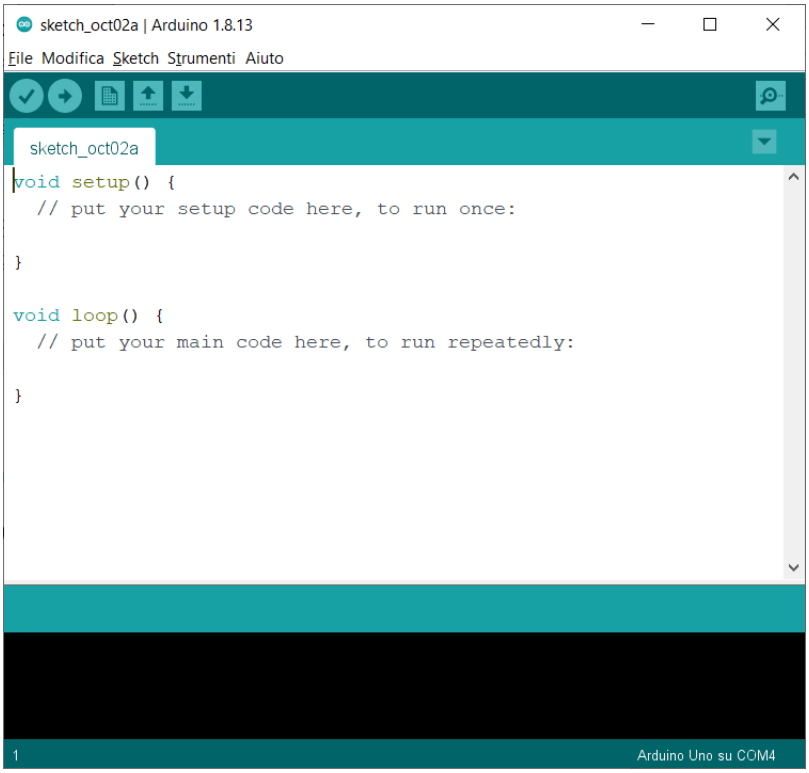
PROCESSO DA AUTOMATIZZARE	*The first thing we do is to identify in detail the entire production process: take the cork sheet, transfer it to the press, etc. ....* *so we define a very precise* ***ordered sequence*** *of actions...*
DEFINIZIONE ALGORITMI	*The various actions, although in a well-defined sequence, are linked together and the link is called an* ***algorithm****.* *For example: before operating the press check that the cork* sheet *is positioned correctly, if yes, then proceed to pressing, otherwise stop ... .* *The one described above is a logical algorithm.... But there are also mathematical* ones*...*
LINGUAGGIO DI PROGRAMMAZIONE	*Once the individual steps of the process and the algorithms that regulate them have been determined, then we move on to translate everything into the programming language and then we have the production of the* ***software*** *that installed on the control machines and PCs, carry out the desired automated process.*

Coding... means this!

Well! After this introduction, let's move on to our Arduino development platform and let's see the sketch...

The sketch

You are surely clear, at least from the latest projects performed, that Arduino, since it is nothing more than a minicomputer, needs to be programmed to run what we want, and this happens in its development environment called IDE.



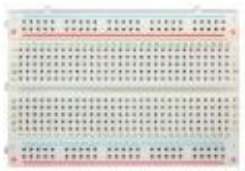
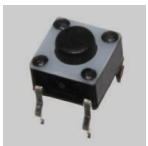
In fact, it is precisely in the sketch that we will write, in a particular language, step-by-step, what Arduino will have to do for us.

The programming language that we will use for sketches is basically ***C++***

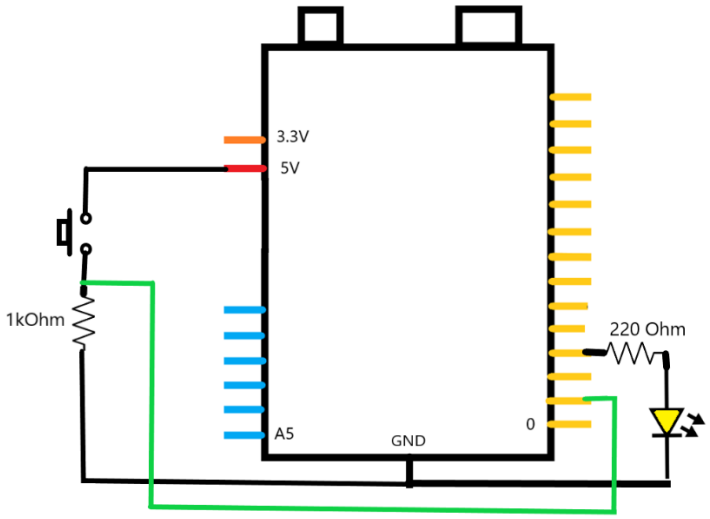
*The sketch consists of two parts, one starting with "**void setup() { ..... }** " and the other with "**void loop() { ..... }** " both parts enclose in two braces the instructions that Arduino will execute. And in particular, in the part enclosed by "void setup()..." as the name "setup" also says, the "variables" are defined, the Arduino PINs are "set", libraries are recalled, etc ... , however, you write all those instructions and / or all those operations that need to be performed at least once. Mentre in the part "void loop ()..." you enter all those instructions that Arduino will execute in an infinite loop (cycle).*

Project 6 – The Stelliere

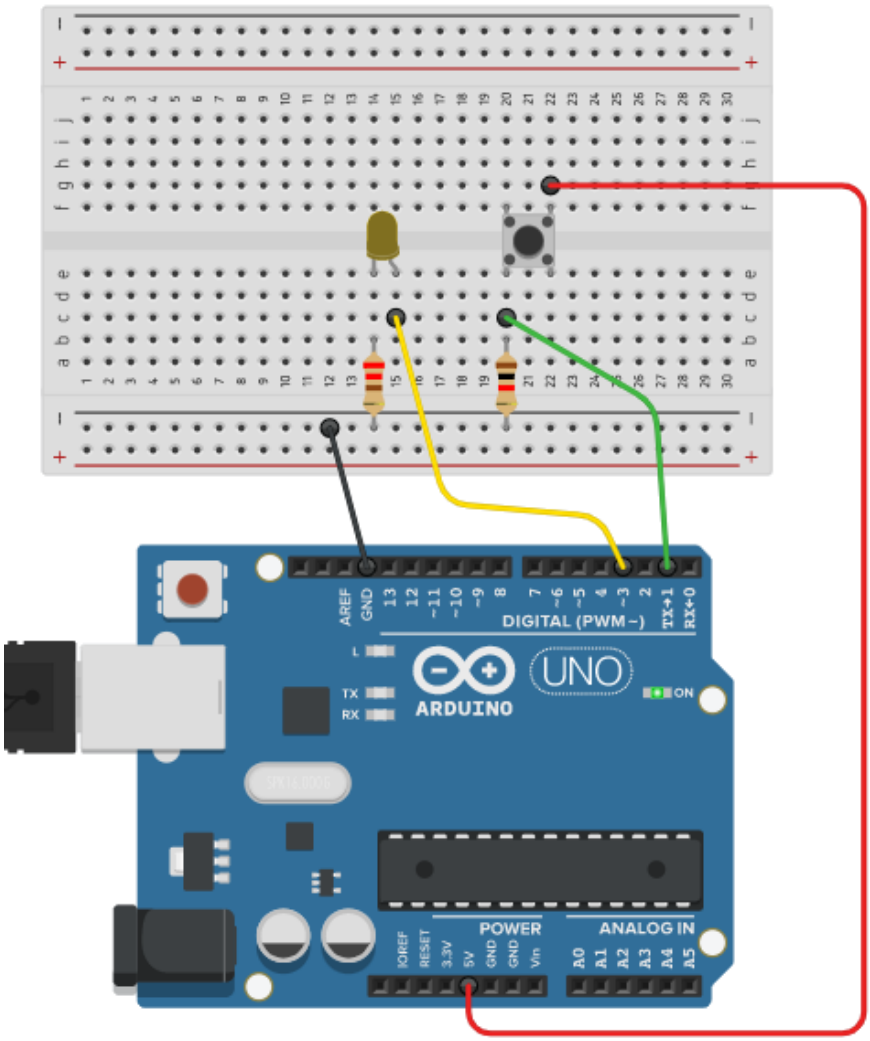
With this project we learn to activate an event on Arduino at the touch of a button. The event, in this particular case, is the lighting of an LED, but it could also be an entire starry sky!

			
Arduino Uno R3 or compatible	*Breadboard*	*Resistance 1kOhm + 220Ohm resistance*	*Dupont cables male - male*
			
USB connection cable	*Button*	*Yellow LED*	*Printed Cardboard*

The wiring diagram is as follows:



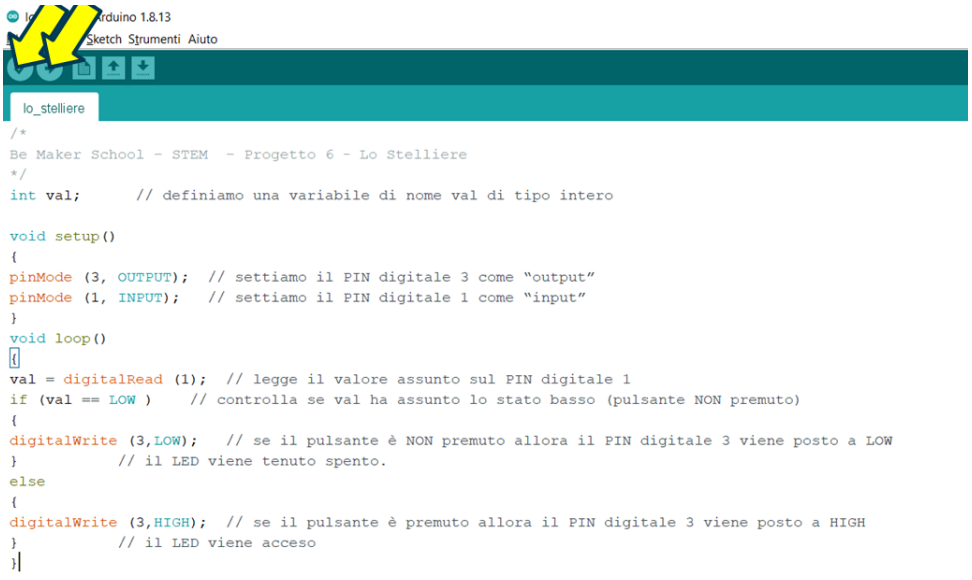
The mounting diagram is instead this:

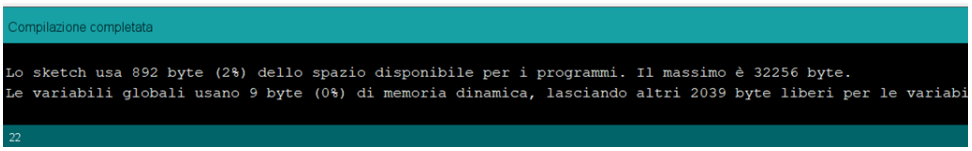


After the links, we move on to write the sketch. Connect Arduino to the PC via the USB cable and launch the Arduino IDE application and write the following code...

```
/*
Be Maker School - STEM - Project 6 – The Stelliere
*/
int val;//                  we define a variable named "val" of integer type

void setup()
{
pinMode (3, OUTPUT);//                  we set the digital PIN 3 as "output"
pinMode (1, INPUT);//           we set digital PIN 1 as "input"
}
void loop()
{
val = digitalRead (1);//           reads the value assumed on the digital PIN 1
if (val == LOW )//                 check if val has assumed the low state (button NOT pressed)
{
digitalWrite (3,LOW); //                  if the button is NOT pressed then the digital PIN 3 is placed at LOW
}                                  the LED is kept off.
else
{
digitalWrite (3,HIGH);//                  if the button is pressed then the digital PIN 3 is placed at HIGH
}                                  the LED is lit
}
```

Once you have written the code launch the verification precompilation (check mark), it will ask you to save the sketch (you can change its name) and then click on the arrow for loading.





Once the sketch is started, every time you press the button, Arduino controls the led to light. Place the printed cardboard in such a way that the yellow LED comes out of the star and you have become a Stellieri !

From this project onwards, in order to learn the programming language for the realization of the skatches, we also begin to do the analysis of the sketches of the projects that we will gradually realize.

[For the video-project click **here**]

Sketch Analysis: The Stelliere

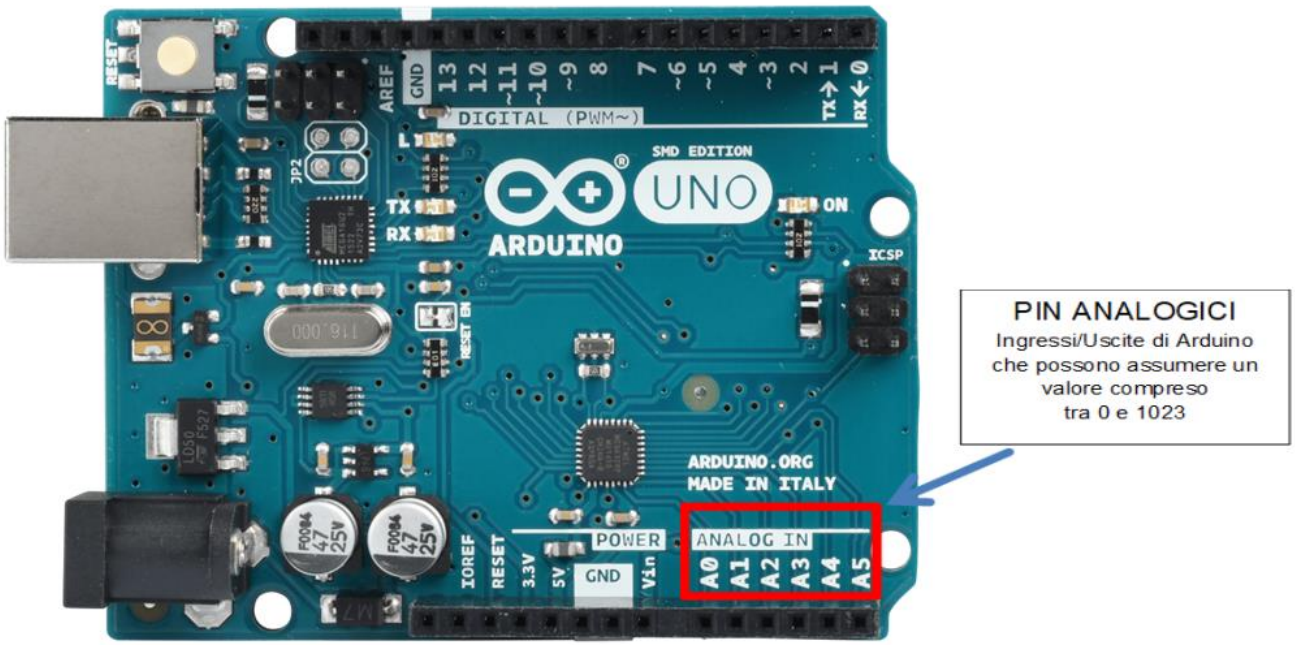
Analyzing the sketch, we can observe several interesting things, first of all the usefulness of inserting comments. Comments (which will be ignored by Arduino) can be entered in two different ways. The first is with the use of the divided "/" and the asterisk "* ", in particular they are put at the beginning and at the end of everything that Arduino must ignore as a comment:

/* [comment] */ . This method is used when the comment occupies multiple lines.

The second method is to put the double divided "", in this way everything that is to the right of the // and for the entire line, is ignored by Arduino and not executed because it is seen as a comment. This mode is used when the comment is on the same line as a statement to describe what the specific statement does.

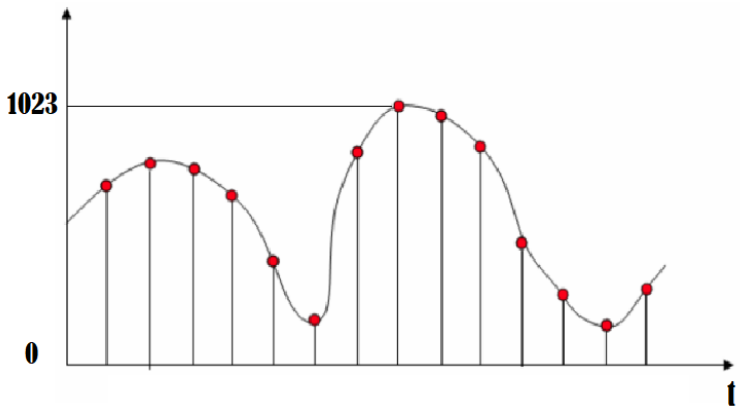
Another interesting thing that I want to highlight of the sketch is the definition of digital PINs in the void setup(), we said that digital PINs can take two values: HIGH and LOW (or even 0 and 1) and we said that Arduino digital PINs ranging from n. 0 to n. 13 can be INPUT and OUTPUT PINs. In this project the digital PIN 1 is INPUT, in fact when the button is pressed it brings voltage to the PIN, so it receives a signal that we read with the **digitalRead** instruction. Digital PIN 3, on the other hand, provides a LOW or HIGH signal with the **digitalWrite** instruction, so it is a PIN in OUTPUT. This functionality of digital PINs in the sketch, whether INPUT or OUTPUT must be declared in the void setup().

Arduino Analog PINs



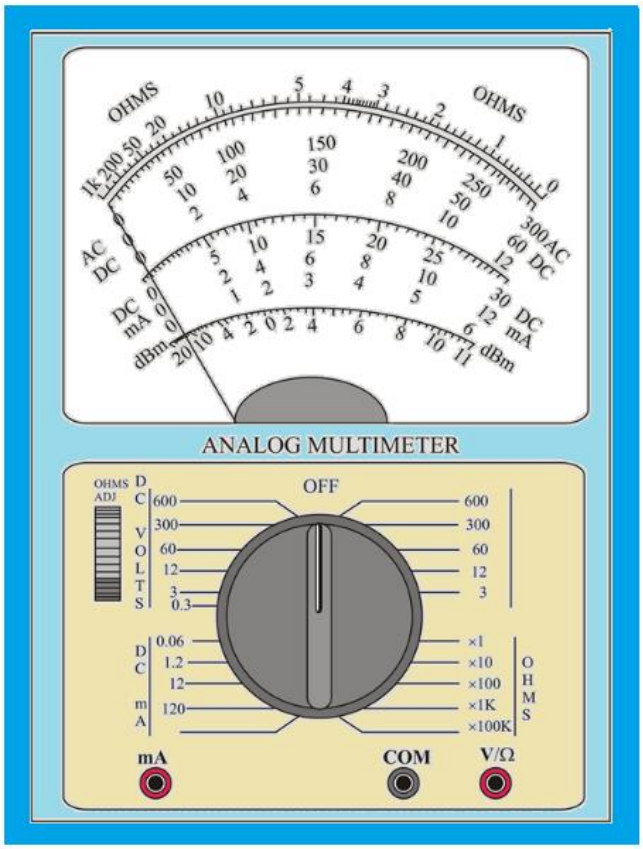
Arduino has 6 Analog PINs named A0, A1, A2,... A5. Analog PINs, such as Digital PINs, can be configured as inputs or outputs depending on their use.

It is necessary to make a clarification: an analog quantity can actually take on infinite values, since Arduino is essentially a minicomputer and is therefore characterized by digital quantities, it cannot reproduce the infinite values required by an analog quantity, but the magnitude is "discretized" and brought back to a value between 0 and 1023, that is, in a range of 1024 possible values. Since the quantities read or reproduced by the Analog PINs are actually electrical voltages and knowing that Arduino has a maximum voltage of 5V output from its PINs (both digital and analog), it is immediately clear that the value 0 is associated with 0 V and the value of 1023 with 5V.



At this point by applying a simple proportion, we can derive the value of the magnitude (if the value of the analog PIN is known), or the value of the analog PIN if what is known is the value of the magnitude. Let's see the application of this concept in a Project.

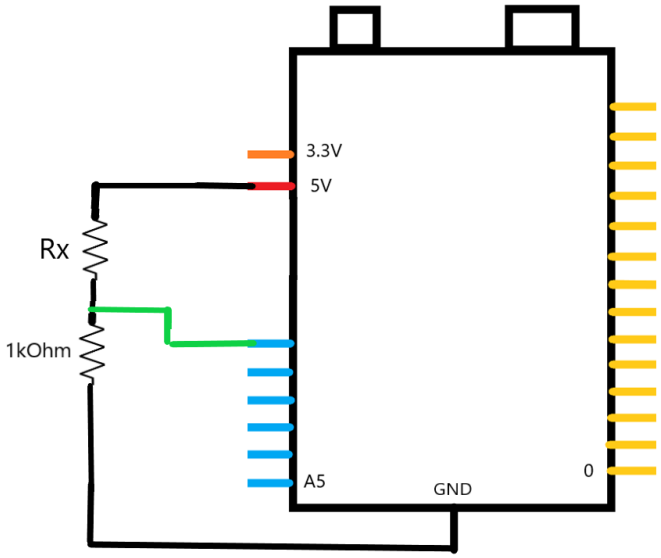
Project 7 – Ohmetro. Using Arduino to measure a resistance



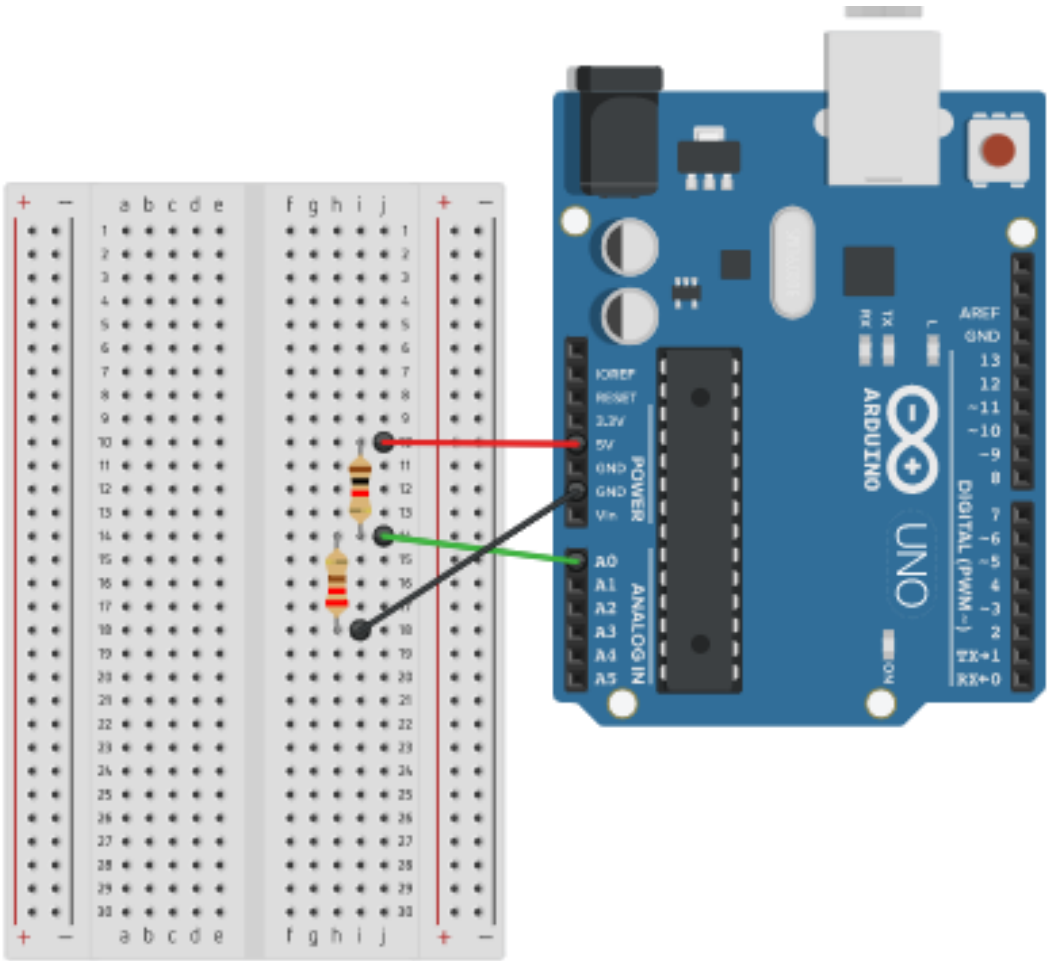
Suppose we have a resistor whose bands are not readable or have erased, the question is: how do we determine the resistance value of an unknown resistance?
This project shows the use of Arduino when you want to know the value of a resistance of which for any reason the colored bands are not reported above the component. We need:

			
Arduino Uno R3 or compatible	*Breadboard*	*Resistance 1kOhm*	*Dupont cables male - male*
			
USB connection cable	*Unknown resistance or alternatively... 220 Ohm resistance*		

The wiring diagram is as follows. We use a 220 Ohm resistor as Rx:



The mounting diagram is instead this:



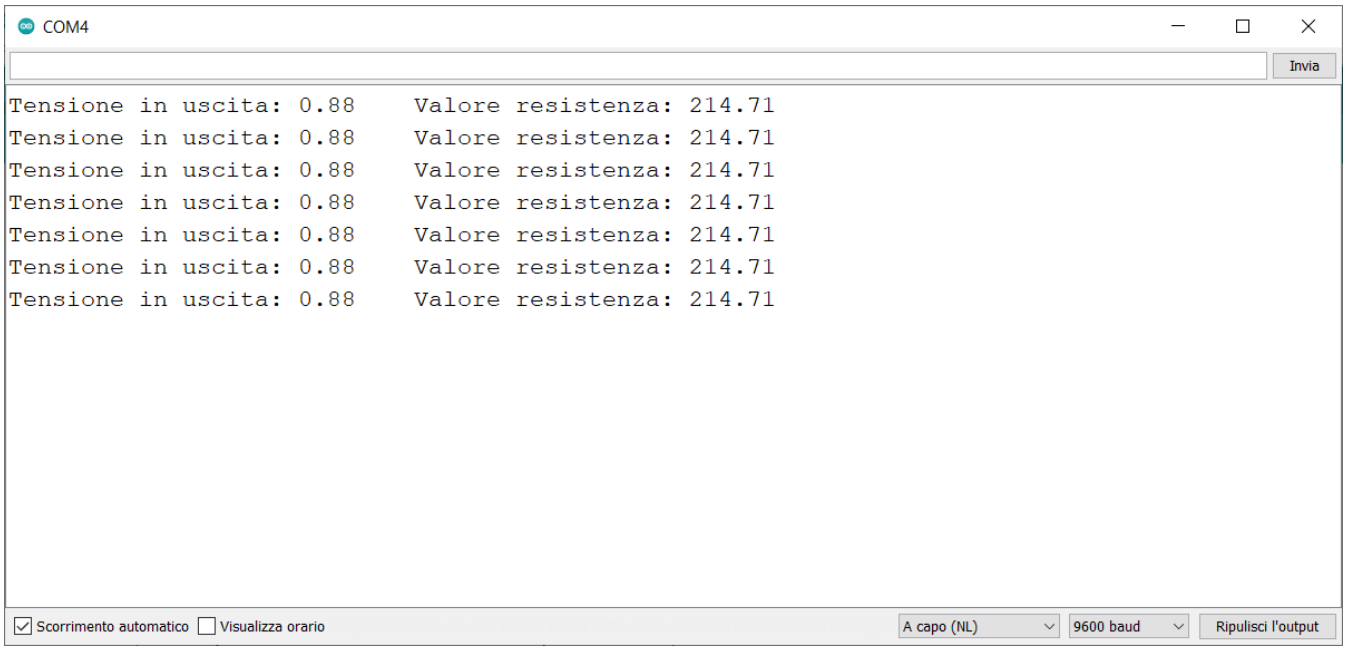
Note: A 220 Ohm resistance was used as the unknown resistance.

At this point we click twice on the Arduino icon on the Desktop and the IDE opens and we copy the following sketch, I recommend you follow each step indicated:

```
/*
Be Maker School - STEM - Project 7 – Ohmetro
*/

void setup() {
  Serial.begin(9600); serial port initialization
}

void loop() {
  int reading=analogRead(A0);          I read the value from pin A0
  float Vout=(read*5)/1024.0;        With the proportion I calculate the voltage
  current float=(5-Vout)/1000;         I apply Ohm law and calculate the current
  float ResIncognita=Vout/current;     calculation of unknown resistance

  Serial.print("Output voltage: "); print the results
  Serial.print(Vout);
  Serial.print(" Resistance Value: ");
  Serial.println(ResIncognita);
  delay(1000);                         the cycle starts again after 1 sec.
}
```

Once you have written the code launch the verification precompilation (check mark), it will ask you to save the sketch (you can change its name) and then click on the arrow to load and then on the lens to access the serial monitor.

Once the Serial Monitor is started, the result is this:



*If you remember tolerance.... so the measured value (214.71 Ohm) will always be slightly different from the theoretical one of 220 Ohms.... in fact, if you apply the formula of the percentages [(220-214,72)/220]*100 = 2,4, you then get an error of about 2.4%, that is, within the tolerance admitted which is the one declared by the manufacturer of the resistance (+/- 5%), otherwise you could ask for reimbursement ...*

Suggestion! : When you use Arduino to make measurements, always check the correct functionality of the components, otherwise you risk going crazy ... for example I happened to use a 1 kOhm resistor which was actually 10 kOhm! So I came up with values ten times smaller than expected. I solved the problem by verifying the actual value of the resistance with a tester and then I realized that a batch of resistors bought and sold for 1 kOhm, were actually 10 kOhm.

Sketch Analysis: Ohmetro with Arduino

Analyzing the sketch of the Ohmeter, we observe the following:
In the "void setup() { ..... }" enclosed by the two braces there is the following statement:
Serial.begin (9600);
this instruction sets the communication speed between Arduino and the PC at 9600 bits per second, that is the maximum communication speed on the COM port between PC and Arduino. This statement is inserted when you want to display the values of the variables on the serial monitor in the IDE. Variables are empty containers that are filled during the sketch (we will see more about this topic in a later lesson).
The instruction "analogRead(...)" is one of the most important instructions in the Arduino language, in fact through this instruction it is possible to read the values of the analog PINs present on Arduino. The correct syntax is as follows ("read" is an arbitrary name assigned to a variable):

```
int reading = analogRead(A0);
```

At the end of every instruction is the "; " , the semicolon indicates that the instruction is finished and Arduino will move on to the next one. After this instruction the sketch that runs on Arduino reads the value on the PIN and assigns it a number from 0 to 1023 (ie 1024 values), where "0" if the value of the voltage read at PIN A0 is 0 V and "1023", if the voltage read is 5 V (ie the maximum voltage that can be produced on the PIN by Arduino). For intermediate numbers at these two extremes, and therefore to have the intermediate values of voltage read on the PIN, it is necessary to make the proportions ..... Holy Mathematics!!!
In fact, for example, suppose that the value read on the PIN is "184", at this point we apply the proportion:

1024 : 5 = 184 : x

(it reads 1024 is at 5 Volts as 184 is at X Volts, where X is the unknown)
The solution formula of this proportion tells us that to derive the unknown value X, we must multiply the internal members of the proportion and divide by the external (known) member, whereby:

X = 5 * 184 /1024 = 0.898 V

So the voltage measured by Arduino on pin A0 is about 0.9 V. Note the voltage, applying Ohm's Law, seen in the previous lesson, we derive the value of the unknown resistance. I stop here with the explanation of the formulas used, because I do not want to bore you with further calculations that we will see in other projects.
At this point we see the last two instructions:

```
Serial.print("Output voltage: ");
Serial.println(Vout);
```

The Serial.print ("....") statement prints on the screen, on the Serial Monitor, the content between the " " (quotation marks) and prints on the same line. To access the Serial Monitor, as mentioned above, click on the lens. The Serial.println (...) statement prints and then goes to the wrapped line for the next print.
The Serial.println(Vout); statement prints the value contained in the Vout variable (and then wraps to the next line).

(print and crop)