Onboard Spacecraft Software Course
Laboratory 1: Introduction to Arduino.
Analogical and Digital Input/Output
2023/2024

1 Objectives

This laboratory introduces the Arduino platform and, in particular, it focuses on studying the analogical and digital input/output ports. The goals of this laboratory are the following:

1. To present the ARDUINO UNO microcontroller.
2. To design an electronic circuit and to develop the program, called sketch, which allows to control the state of a led in two ways: 1) digital way (part 1) and analogical (part 2).

2 Electronic components

To carry out this laboratory the student needs the following electronic components:

- Arduino UNO microcontroler.
- 1 breadboard.
- 2 LEDs (Light-Emitting Diode).
- 1 Light Dependent Resistor (LDR).
- 1 220 ohmios resistors.
- 1 10 K ohmios resistors.
- 1 pushbotton.
- Hook up wire.

To save time during the session of laboratory, we recommend to bring a set of wires previously cuted with different lengths: 2cm, 5cm, 10cm. Appendix A shows a brief description of these electronic components.

3 Arduino platform description

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Figure 1 shows the main components of this microcontroller.

Among its components, we highlight the next ones:

- 6 analogic input pins, which are named from A0 to A5. We can connect to them some device of analogic input, as for instance, a sensor. The microcontroller contains an onboard 6 channel analog-to-digital (A/D) converter of 10-bit resolution. The result of this A/D is a digital value ranging between 0 and 1023.

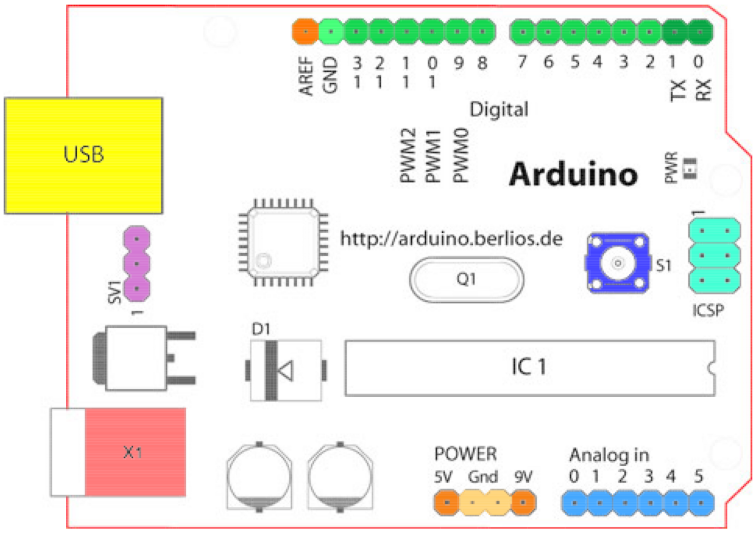


Figure 1: Arduino microcontroller

- 14 digital pins, which are named from 0 to 13. The pins on the Arduino can be configured as either inputs or outputs in each program.
- 6 analogic output pins, corresponding to the digital pins 3,5,6,9,10 and 11. These pins can be configured as analogic output in each program.
- An Atmega168 chip of 28 pins (see http://arduino.cc/es/Hacking/PinMapping168).
- An USB port, which provides the power to the microcontroller as well as provide the serial interface to communicate the PC with the Arduino platform (and viceverse).

3.1 Tinkercad development environment

Tinkercad Circuits is a free, online service from Autodesk that began in 2017 and is probably the most user-friendly Arduino simulator. You can easily design your own circuits, create a program in block or text format and then debug it. Figure 2 shows an example of the platform.

This platform allows the simulation of Arduino boards, breadboard circuits and IO interfaces and the interaction with the arduino code. The model and the code can be downloaded and shared with other makers.

To access the platform open on the browser the url `http://tinkercad.com`, and then log in with your Google account. There is a tutorial on how to use the platform at the url `hhttps://www.tinkercad.com/learn/circuits`. Each lesson includes a sample circuit and a step-by-step procedure to create and use it.

3.2 Arduino IDE

The IDE (Integrated Development Environment) of Arduino is a special software program that allows to write programs, called sketches, for the Arduino microcontroller. The IDE is an open-

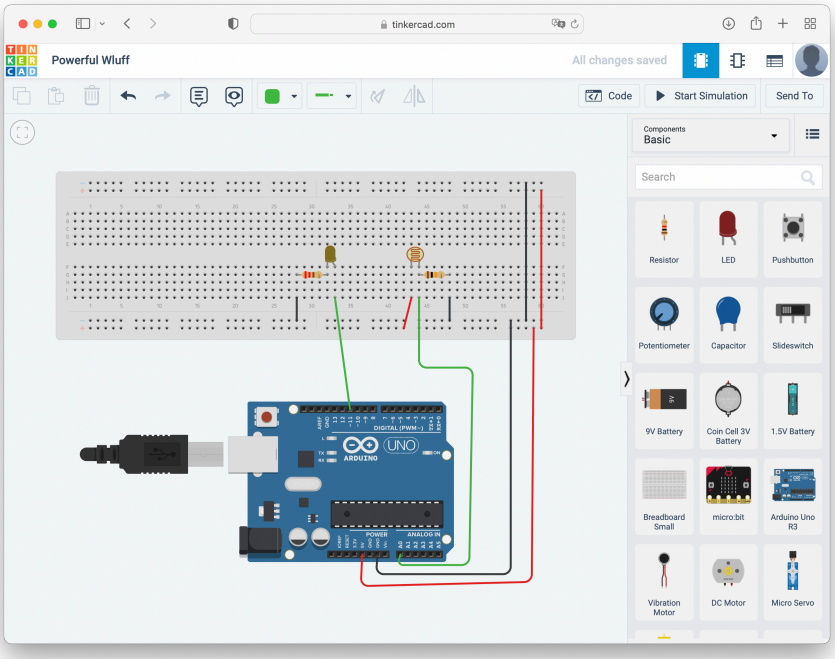


Figure 2: Tinkercad development platform

source software that can be download from http://arduino.cc/en/Main/Software For the Linux, Windows and Mac platforms. Figure 3 shows the Arduino IDE platform.

Regarding the Ubuntu 22.04 operating system the best way to install this tool is as follows:

```
$ sudo apt install arduino
```

Then, it is necessary to include the user account into the *dialout* group as follows:

```
$ sudo usermod -a -G dialout <username>
```

Finally, to start the environment, execute the next command:

```
$ arduino
```

Figure 4 shows the toolbox from the Arduino IDE. The two first buttons are intended to: 1) compile and verify the sketch and 2) upload the executable code into the microcontroller.

When the user press the *Upload* button, the code corresponding to the sketch is automatically translated to the C programming language. This source code (written in C) is passed to the avr-gcc compiler (the compiler for Atmel platforms), which finally translates the code to the specific language of the microcontroller. In this way, Arduino simplifies the programming and hides the hardware low-level details.

The development of an sketch consists of the following steps:

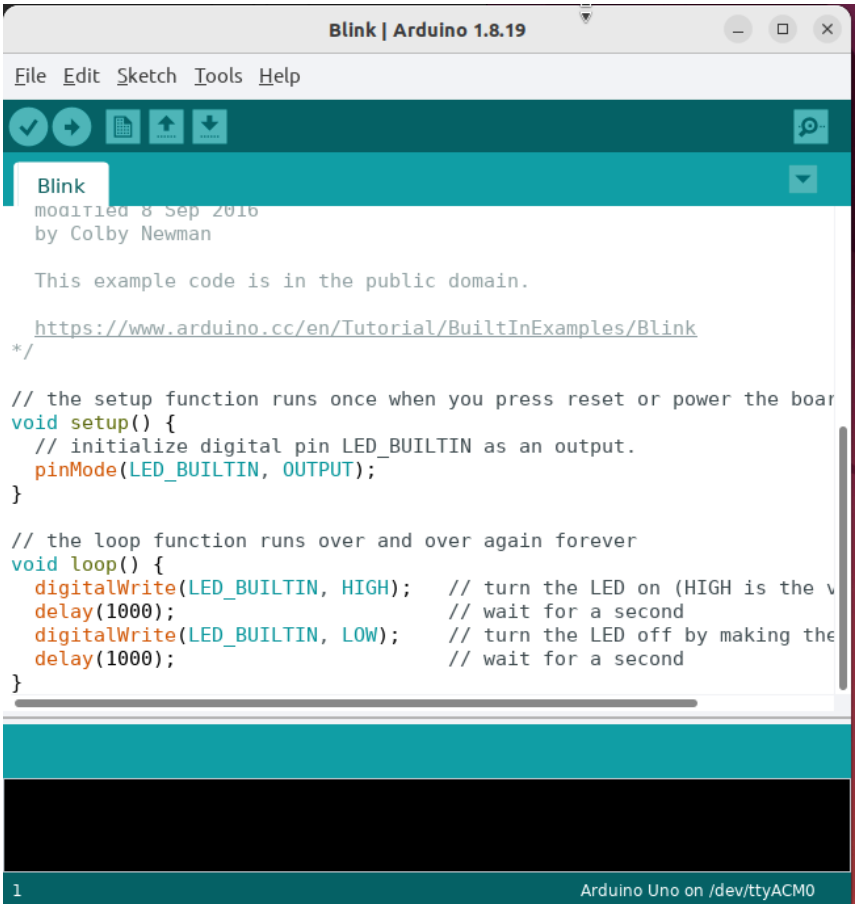


Figure 3: Arduino IDE

- To design the scheme that connects the electronic components. To do this, we recommend to use The Tinkercad platform.
- To interconnect the Arduino platform and the electronic components, as designed in the previous phase.
- To write a program (sketch) by using the Arduino IDE, which allows to control the designed system.
- To connect the Arduino platform to the PC by using the USB port.
- To compile the sketch.
- If no errors, to upload the sketch into the Arduino microcontroller. You must wait a few seconds until the platform restart and to start executing your program.
- Arduino executes the program in an infinite loop. You should check that the system behaves accordingly to your sketch.

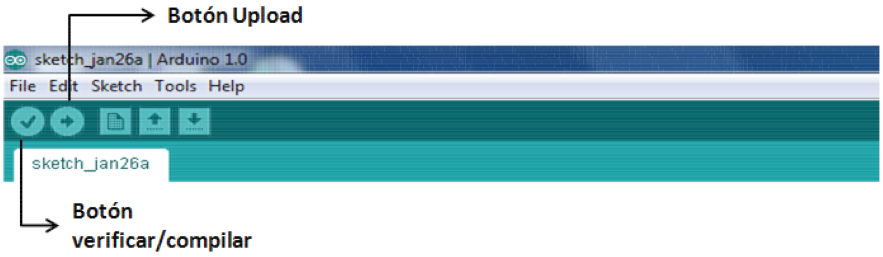
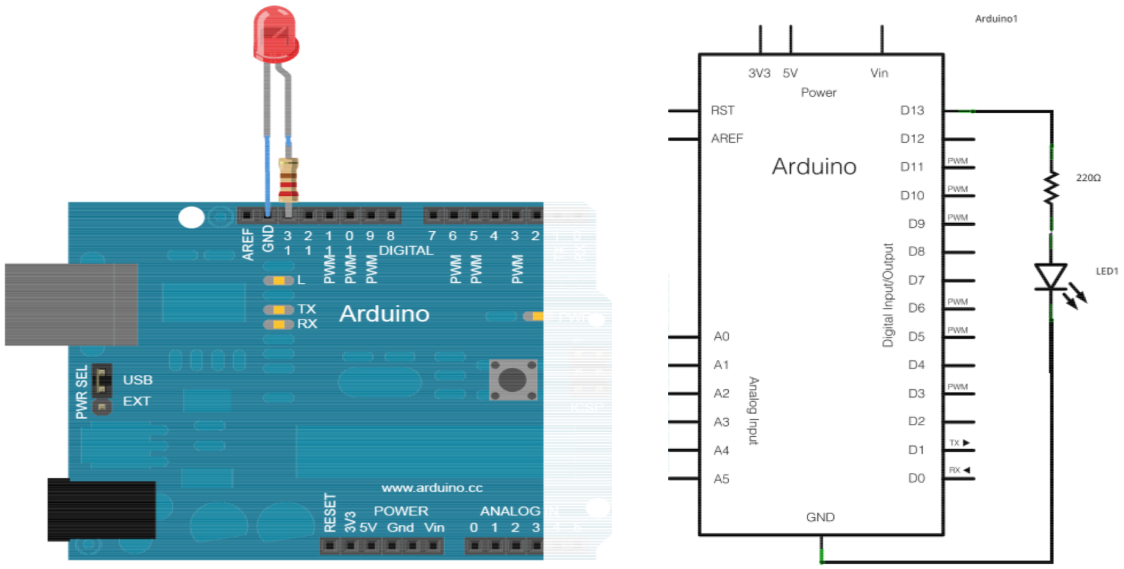


Figure 4: Arduino IDE button bar

3.3 A simple application: Blink

We describe the actions that the group of students should take in each laboratory by showing a simple application: Blink.

- To design a simple electronic circuit that connects a LED to the 13 digital output by mean of a resistor (the other pin should be connected to 5V) and the other leg of the LED to GND, as shown in Figure 5. On the right, you can see the connection scheme.



(a) Physical connection

(b) Electronic schema

Figure 5: Connection between the breadboard and Arduino for the Blink application

- To start the Arduino IDE and go to the menu: File > Examples > 1.Basics.
- To select *Blink* application and check that you see a new window with the source code (sketch) of this application.
- To compile and verify the code (first button of the toolbox).

- To upload the code into the platform by pressing the *Upload* button.
- To observe the behavior of the system: What does Blink do?

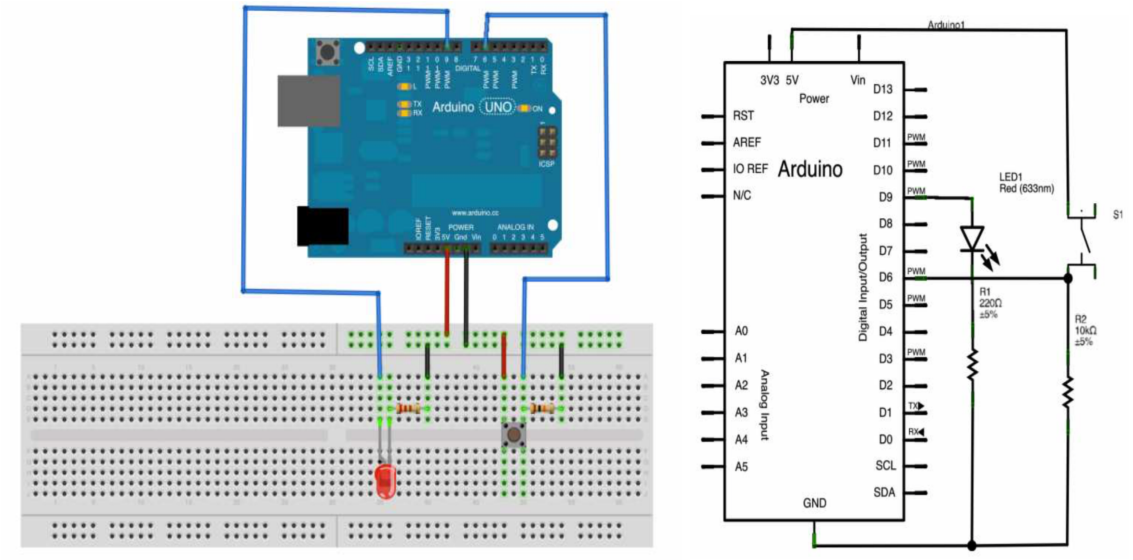
4 Description of the laboratory

The laboratory consists of designing a simple control system that allows to switch on and off a LED by taking into account some basic instructions. A LED is a semiconductor light source, that in the Arduino platform may be controlled in a digital way (0: off, 1:on) or analogic way (a range of values between 0 and max). The laboratory is divided in two parts:

1. Digital control of a LED by mean of a pushbutton. You have to design a electronic circuit that connects to Arduino the following two components: 1) a pushbutton, that acts as the digital input; 2) a LED that turns on and off when the user press the button (digital output, 0 is off and 1 is on). When the LED is in the state LOW (off) and the user presses the button the LED turns on; otherwise, if the LED is in the state HIGH (on) and the user presses the button, the LED turns off.

   You should follow the next steps:

   - To design the interconnection scheme to control the state of the LED by mean of the button. To do this, from the Blink scheme, you should connect the button in the next way: one of the "legs" has to be connect to 5V and the other, to any of the digital inputs (e.g. the 6 pin). In order to close the circuit, you should connect one "leg" of the resistor to the used digital input and the other one, to ground (GND). Figure 6 shows the physical connection between Arduino and the breadboard (on the left) and the connection scheme (on the right).
   - To start the Arduino IDE and modify slightly the Blink example in order to program a new sketch which performs the next operations:
     (a) To configure the 6 digital pin that is connected to the button as input by using the $pinMode$ function.
     (b) To read the value of the digital input by mean of the $digitalRead$ function: when the user presses the button, there exist a flow of current between the two legs of the button, and then, this function returns the value 1. Otherwise, there is not flow of current and the function returns 0.
     (c) To use the value returned by $digitalRead$ to write such a value into the digital output pin connected to the LED, by using the $digitalWrite$ function.
     (d) To observe that the LED turns on and off depending on if the button is pressed or not. To modify this behaviour and to enable that the LED just turns on and off when the user has pressed the button (e.g. the LED keeps the same state until the user presses the button again), you should save the previous state of the button (pressed or not pressed) in a variable and compare its value with the current reading from the button. The LED only changes it status if the button old state is OFF and the new ON. The next code makes this action:



(a) Physical connection

(b) Electronic schema

Figure 6: Physical connection (on the left) and electronic schema (on the right) for laboratory 1 (part 1))

```
state = 0;
value_old = 0;
while (1) {
value = digitalRead(BUTTON);
if ((value == 1) && (value_old == 0)){
state = 1 - state ;
}
digitalWrite(LED,state);
value_old=value;
}
```

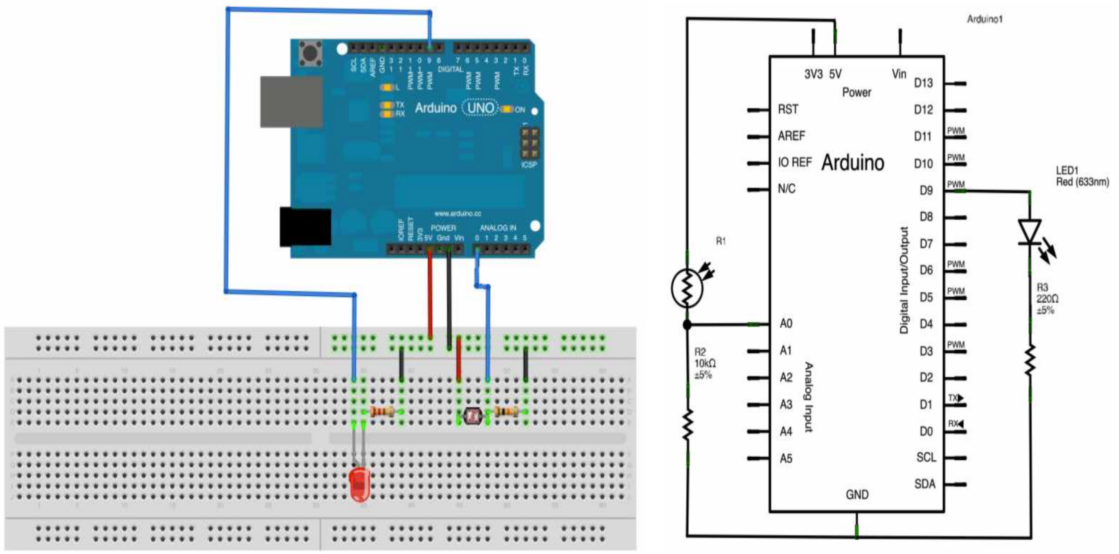
- To compile and verify el sketch.
- To connect the Arduino platform to PC by the USB port.
- To upload the code from the PC to the Arduino platform.
- To check your program is correctly executing.

2. Analogic control of a LED by mean of a photocell. You have to modify the previous electronic circuit and connect to Arduino microcontroller a sensor of luminosity given by a light dependent resistor (LDR). A sensor is a device able to sample an environmental variable (e.g. temperature, humidity, luminosity), and to obtain an analogical value as a result. An analogical/digital converter is then aim to transform such a value into its corresponding digital value. From the measure of light given by the LDR, you should analogically write the

LED. In this way, the bright of the LED can be modified according to the reading of the sensor. Finally, your should print on the serial port the value written on the LED.

You should follow the next steps:

- To design the interconnection scheme of the circuit that controls the state of the LED by analogically writing the value provided from the LDR sensor. To do this, from the Blink scheme, you should connect the LDR as shown in Figure 7, which shows the connection between the breadboard and Arduino platform (on the left) and the interconnection scheme (on the right).



(a) Physical connection

(b) Electronic schema

Figure 7: Physical connection (on the left) and electronic schema (on the right) for laboratory 1 (part 2))

- To start the Arduino IDE and modify slightly the Blink example in order to program a new sketch which performs the next operations:
  (a) To read the value from the LDR connected to the analogic input pin by mean of the $analogRead$ function. You can check that the sensor is correctly reading the environment by printing the result of the function on the serial port.
  (b) To use the value return by the $analogRead$ function to write it on the output connected to the LED, by mean of the $analogWrite$ function.
  (c) To observe that the brightness of the LED varies depending on if the sensor detects more light (try to cover the sensor with your hand).
  (d) To compile and verify el sketch.
  (e) To connect the Arduino platform to PC by the USB port.
  (f) To upload the code from the PC to the Arduino platform.
  (g) To check your program is correctly executing.

4.1 Evaluation of this laboratory

The evalution of this laboratory will consist on a serie of videos recorded to demostrate the correct operation of the two circuits described on the former section; Therefore the student should:

- Record a video showing a few seconds of the operation of the circuit described in part 1 of this laboratory (Figure 6).
- Record a video showing a few seconds of the operation of the circuit described in part 2 of this laboratory (Figure 7).

It is recomended to use a low resolution or well-compressed video format to reduce the size of the recordings.

Both recordings should be delivered using the proper delivering tool set on Aula Global. The delivery should be done a week from the class related to this laboratory.

5 Bibliography

- Arduino microcontroller (http://arduino.cc).
- TinkerCad Web Page(http://www.tinkercad.com).
- Slides of the Real-Time Systems course.

Appendix

A Appendix A. Description of the electronic components

A1 LED

A LED is a semiconductor light source diode. Many devices uses LEDs with different colours as indicators. There are several colours but the more commons are red, green and yellow. The pins of the LED have different sizes. The longer one is the anode (that should be connected with the positive size of the source) and the shorter one is the cathode (that should be connected with the negative size of the source). Figure A1 shows the look-alike of some LEDs.

Figure A1: Leds of diferent colors

A2 Light Dependent Resistor

A photoresistor or light dependent resistor (LDR) is an electronic component whose resistance decrease with increasing incident light intensity (it can decrease until 50 ohms) and increase in the dark (until several megaohms). Figure A2 shows an LDR.

Figure A2: Ligth Dependent Resistor (LDR)

A3 Breadboard

A breadboard is a plastic board that contains an array of holes electronically connected in a certain way. The holes are placed with a separation of 2.54 mm (the same as the pin separation of most electronic components). the pins of the electronic components have to be placed on this holes. Usually the inner arrays of the breadboard have all the holes of the same column connected between them. Furthermore, the holes of each row placed outside are also connected between them. this outer rows are intended for connecting power and ground wires to the rest of the circuit.

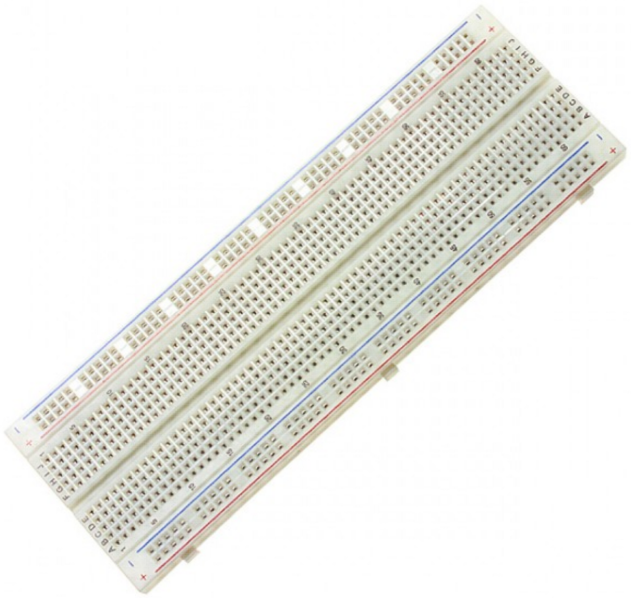


Figure A3: Breadboard

A4 Resistors

A resistor is electronic component used to incorporate certain amount of resistance of an specific part of a circuit. In any circuit, the current and the potential difference are related, using the Ohm?s law, by the resistance of the circuit as following:

$$R(resistance) = V(potentialdifference)/I(current)$$

The value of resistance is measured in ohms. Resistors have a colour code to indicate their resistance value. this code is composed of four coloured rings, where each colour (read from left to right, where the ring at the very right must be golden or silver) means the following:

1. The colour of the first ring represents the tens.
2. The colour of the second ring represents the units.
3. The colour of the second ring represents the measure unit that multiplies the value of the resistance.
4. The colour of the second ring represents the tolerance value.

Table A1 shows the colour code used. (example a resistor with brown, blue, orange and golden rings has a value of $16 * 10^3 \pm 5\%$ Ohms. Figure A4 shows several electric resistors.

Figure A4 shows several electric resistors.

Black	0	Blue	6
Brown	1	Purple	7
Red	2	Gray	8
Orange	3	White	9
Yellow	4	Silver	10 %
Green	5	Golden	5 %

Table A1: Resistors?s code of colours

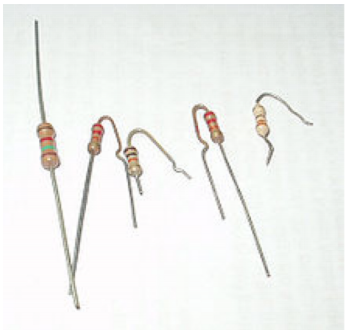


Figure A4: Electric resistors

A5 Push-Button

A push-button is a device composed of two metal pins and a plastic cover. When the button is released, the metal pins are not connected and there is no circulation of electricity. On the other side, when the button is pressed, the metal pins connect and there is circulation of electricity Figure A5 shows the aspect of a push-button.

Figure A5: Push-button