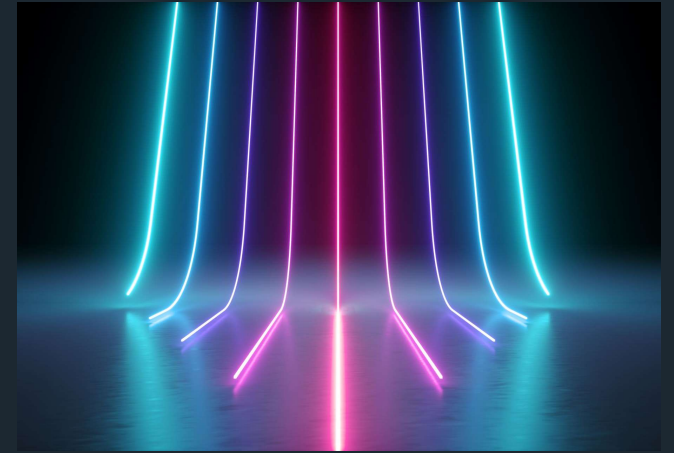


Activity 1: turn on and turn off a LED

ARDUINO

Begginning with
Arduino UNO



Work environment

It is important work on a spacious table, clear and with good light.

Programming environment

Programme Arduino is a **IDE** (Integrated Development Environment).

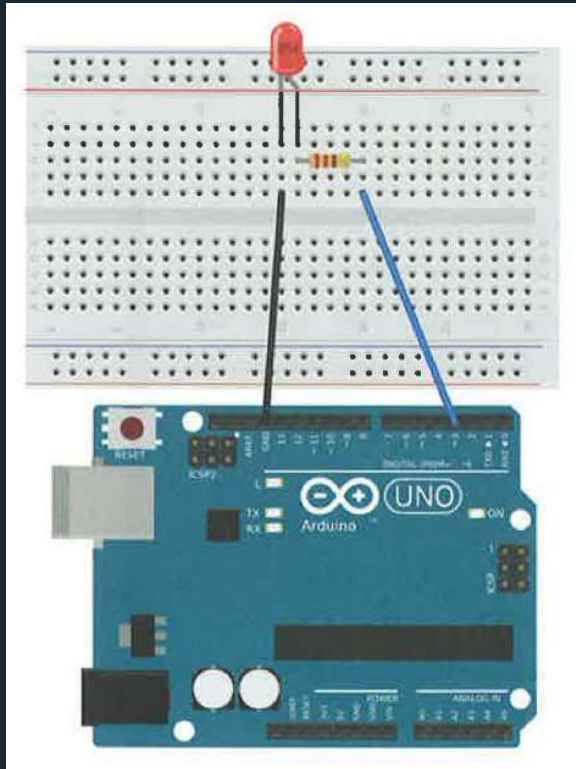
It allows us to **write** and **edit** programs, just as **compile** them (to translate them to a language which the PC understand) and **load** then into the Arduino board.

Arduino IDE is a free download program. We can find it in the Arduino project web page (www.Arduino.cc)

You can download and install the Arduino IDE based on your PC operative system.

Activity 1: turn on and turn off a LED

This is the circuit that we are going to assemble and the list of materials that we will need for it.

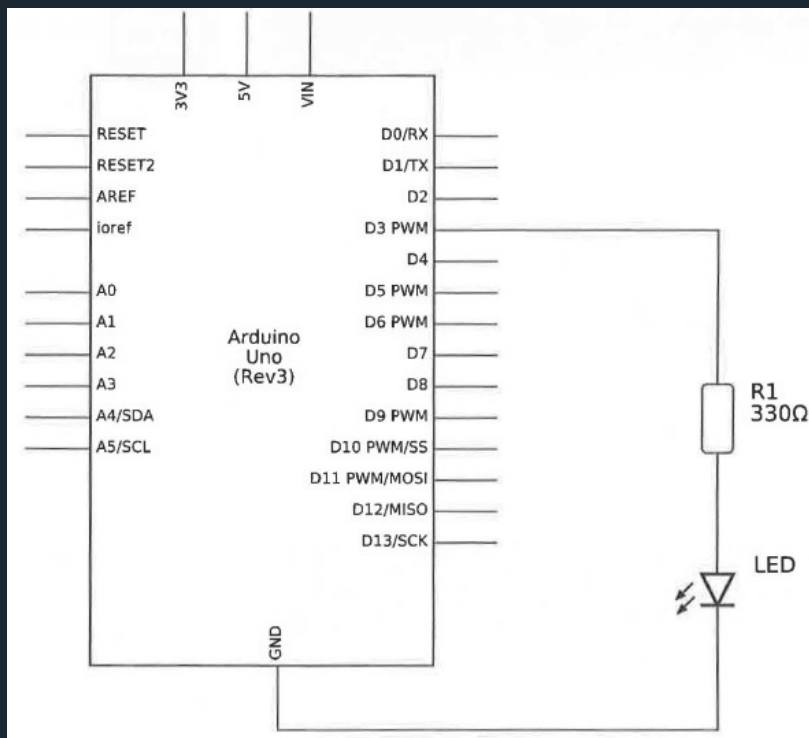


Materials:

- A USB type A-B cable
- An Arduino UNO board
- One 400-pin connexions board
- One LED red 5mm
- A 330 Ohm resistor (orange, orange and brown)
- Two M – M wires

Activity 1: turn on and turn off a LED

WARNING: remember to take the precaution of working with the Arduino board UNO disconnected from the computer or any power supply.

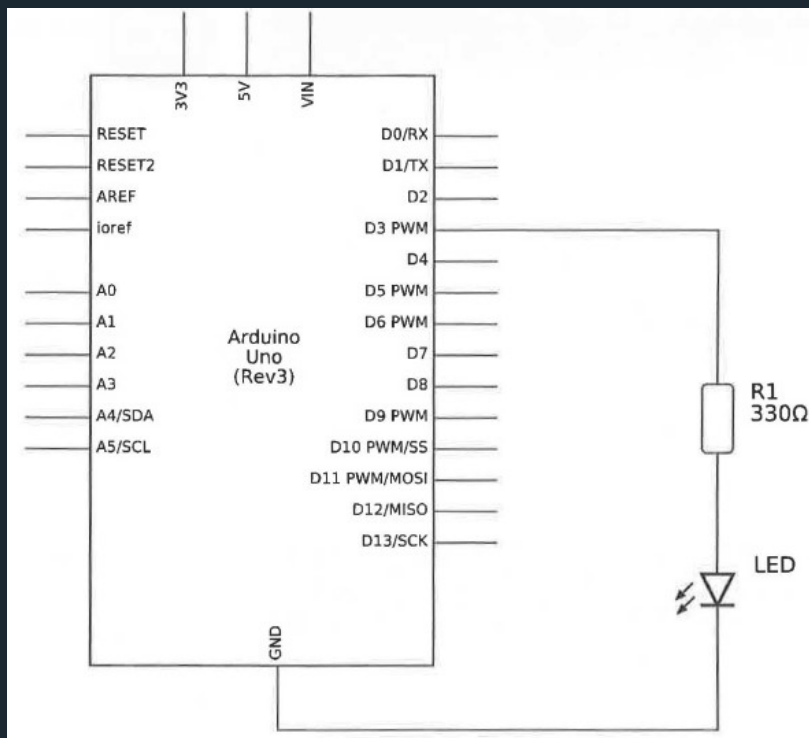


GINESJGARCIA

This is the electrical circuit that represents the assembly that we have to make.

Activity 1: turn on and turn off a LED

WARNING: remember to take the precaution of working with the Arduino board UNO disconnected from the computer or any power supply.



This is the electrical circuit that represents the assembly that we have to make.

As the **LED has polarity**, the only precaution we must take is to place the positive terminal of the LED (the longest) on the side that joins with the resistor. If not, the LED will not light up.

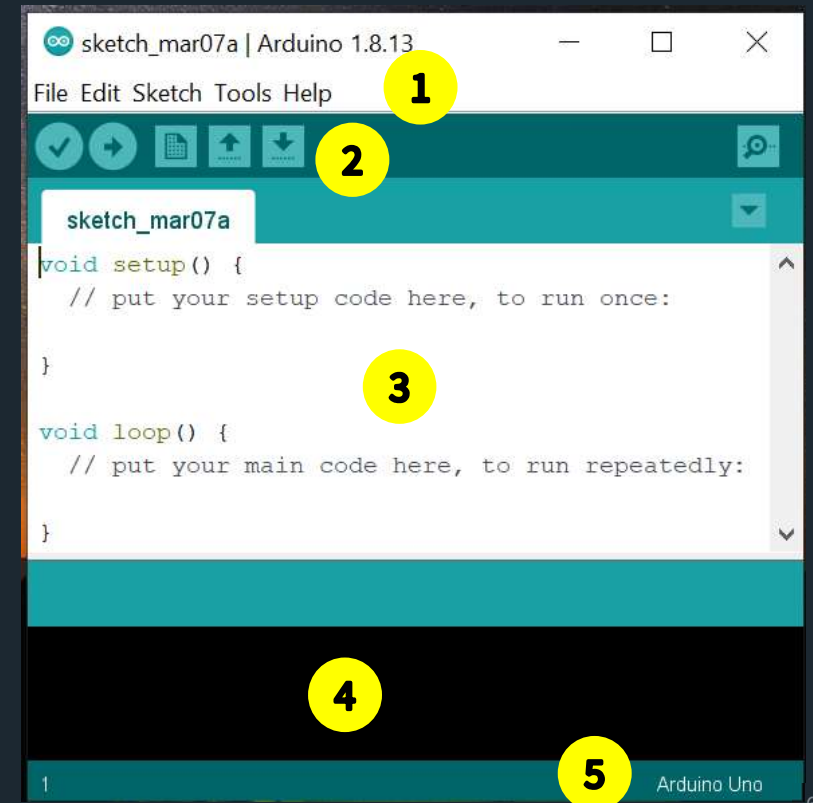
Activity 1: turn on and turn off a LED

Once the assembly is finished, we will connect the Arduino board with the computer using a USB cable, and open IDE Arduino.

- 1 Dropdown menu
- 2 Quick menu of most common functions
- 3 Program text editor area
- 4 Message window
- 5 Status bar

We will describe its different options and its mode of operation as we go needing throughout the different activities and exercises.

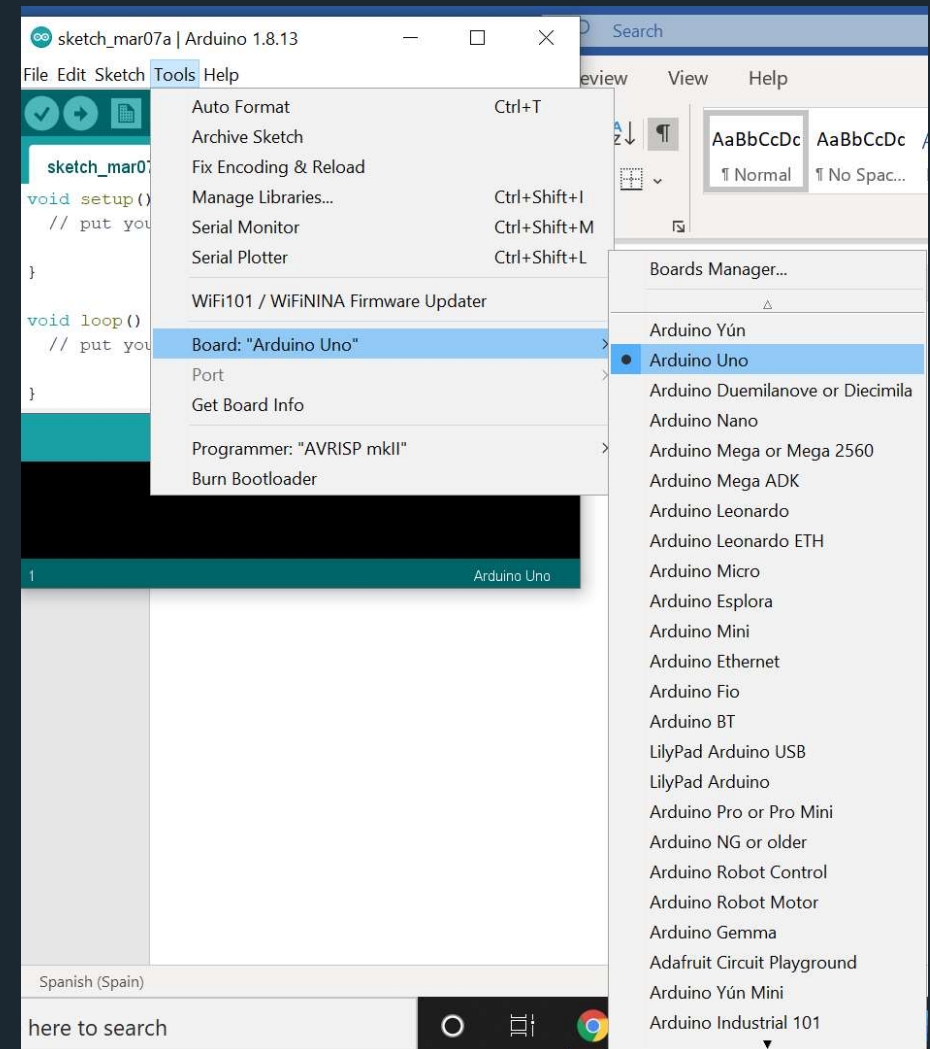
GINESJGARCIA



Activity 1: turn on and turn off a LED

So, and before doing anything else, we are going to configure the Arduino IDE so that it can work correctly with our board.

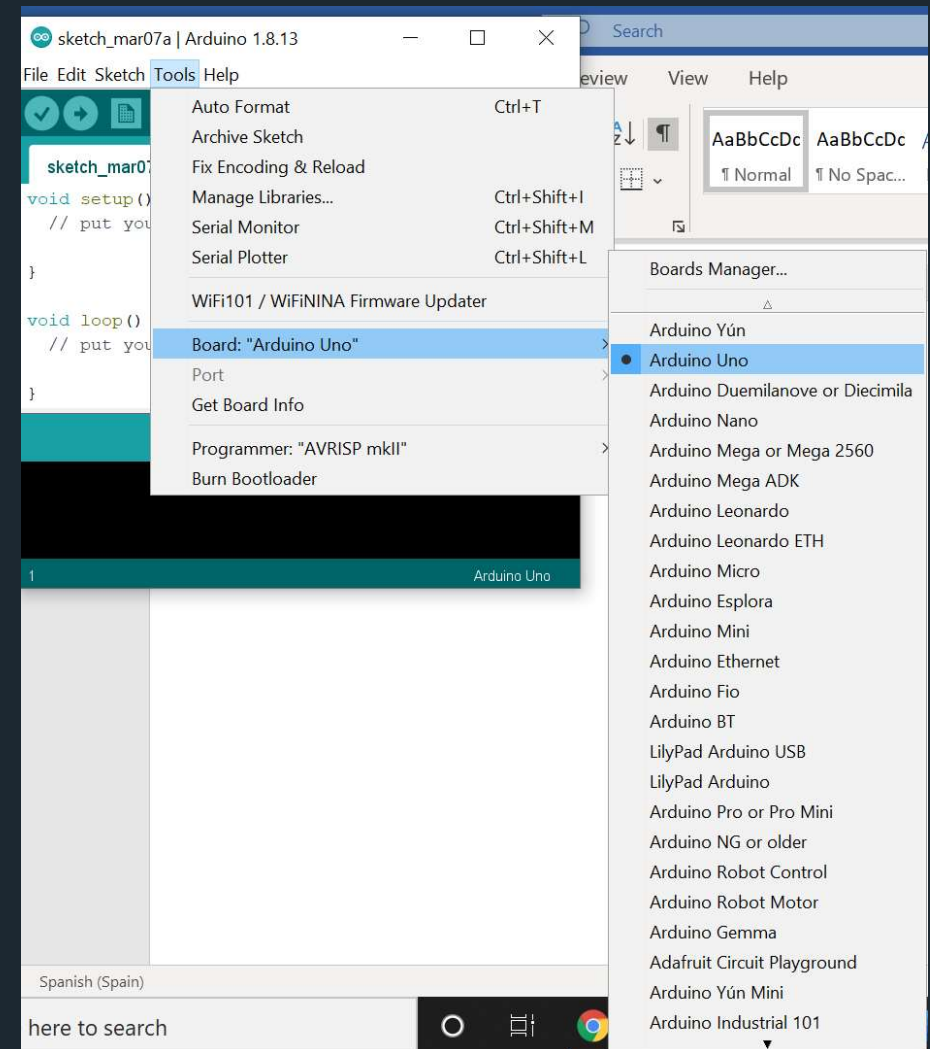
DM Tools >> Board: "Arduino Uno" >> Arduino Uno



Activity 1: turn on and turn off a LED DRIVERS

In Windows 7, 8 and 8.1, the normal thing is that the **driver** necessary for the operation of the card is **installed automatically** the first time we connect the Arduino UNO board to our computer. So, once connected, we only have to wait a few moments to allow time for this process to finish (the computer indicates this with an acoustic signal). **In case** the driver was not installed automatically, **we can install it manually** with the Windows device manager. The drivers for the Arduino board can be found in the corresponding folder of the Arduino IDE installation, for example: e: \ Program Files (x86) \ Arduino \ drivers

GINESJGARCIA



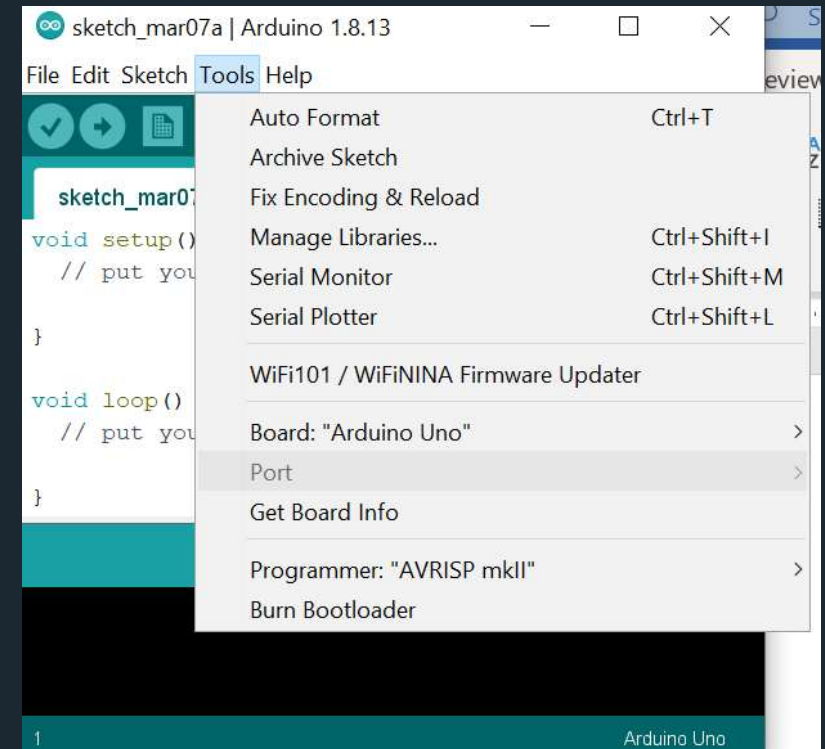
Activity 1: turn on and turn off a LED

By itself, the Arduino IDE is not able to automatically determine the virtual serial **port associated with our Arduino UNO board**. For this reason, it is **necessary to make this selection manually** by clicking on:

DM Tools >> Port >> COM3

In Windows it is normal to select the "COM" with the **highest number**.

If in doubt, you can disconnect the Arduino UNO board and see which port disappears from the list. That will be the COM associated with our card!



Activity 1: turn on and turn off a LED

Writing our first program

With the Arduino IDE we can write programs in text form in **C or C++**.

Yes, I know that many people say that programming like this is very complicated.

The reason that programming in C with Arduino is simple lies in the **enormous simplification work that the Arduino team has done**, thanks to which we can program complex tasks in a simple and efficient way. In terms of programming, it could be said that by using the Arduino IDE we are at a "higher abstraction level", or in other words, that **we will use previously programmed functions** that are part of something that we could call "the Arduino language".

Activity 1: turn on and turn off a LED

Writing our first program

To make our first program we are going to copy the text shown below, in the text editor area of the Arduino IDE. Surely someone may think that it would be much faster to have a download link instead of going through the painful task of transcribing this text. That is true, but this is not written with the idea of doing things the way as quickly as possible, but to teach how to do them. So let's get to work, which is just a few lines of code.

Just one more detail before we start: in the C programming language, upper case and lower case (capital letter and lower case letter) are different things, so pay special attention to writing your text exactly as it is displayed.

Activity 1: turn on and turn off a LED

Writing our first program

```
// Beginning with Arduino UNO
//Activity 1: turn on and turn off a LED
//
// Turns an LED on and off at 2 second intervals
// 1 second on and 1 second off
//
// www.complubbt.com

// We associate LED to the digital pin where we have it
connected
#define LED 3

void setup ()
{
// We configure the digital pin of the LED as output
pinMode (LED, OUTPUT);
}
```

Activity 1: turn on and turn off a LED

Writing our first program

```
Void loop ()  
{  
  // We turn on the LED activating the digital output  
  digitalWrite (LED, HIGH);  
  //Esperamos 1 segundo  
  delay (1000);  
  // We turn off the LED by deactivating the digital output  
  digitalWrite (LED, LOW);  
  // We wait 1 second  
  delay (1000);  
}  
// We wait 1 second  
delay ( 1000);  
}
```

Activity 1: turn on and turn off a LED

Writing our first program

As you have already seen, the text editor of the **Arduino IDE** has a great **help**: the **coloured syntax**. This means that as soon as it recognizes something that is written correctly, it assigns it a certain colour.

Once written, the program resides on our computer. In order for the electronic circuit we have assembled to work, we need to translate (**compile**) **and upload** the compilation result to our Arduino UNO board.

Activity 1: turn on and turn off a LED

Writing our first program

Before doing this, we will analyse the program by looking at some of its elements so that its operation can be understood a little better.

Activity 1: turn on and turn off a LED

Writing our first program

Commentaries:

All the lines that start with the double dividing bar `"//"` are **comments or clarifications made by the programmer about the program**. Its objective is to **better explain the operation of the different parts** of it and in reality they are not part of the program code, since they will not translate into anything that our Arduino UNO board may need.

In other words, we could have saved ourselves writing these lines (sorry, but it's already a bit late). Fortunately, the comments are very useful and important and it is better to get used to always adding them to the programs.

Activity 1: turn on and turn off a LED

Writing our first program

Functions:

Every C program is made up of one or more functions. In the case of using the Arduino IDE, we have to clarify that every program is made up of a minimum of two functions called **setup** and **loop**.

Each function has its name (setup or loop in this example) and two braces that delimit the content of each function. That is, the code of each function is delimited by the symbols "{" and "}".

Activity 1: turn on and turn off a LED

Writing our first program

Functions:

The **setup function** is the **first one that our program will execute** and it will do it **only once** at the beginning of it.

This is why the setup function is often called *the setup or initialization function*.

Activity 1: turn on and turn off a LED

Writing our first program

Functions:

The **loop function** will **execute right after the setup function** and it will do so **repeatedly as long as the power to the Arduino board is connected**.

For this reason we can say that within **the loop function is our main program**, that is, the task or set of tasks that our Arduino UNO board will have to perform.

Activity 1: turn on and turn off a LED

Writing our first program

Definitions:

In this first program we can find a line that says: `#define LED 3`

This is not part of the program itself

Only serves to establish an association between the symbol (or word) "LED" and the number 3.

From the moment we put this definition in our program every time we put LED, the program will know that we are actually putting the number 3 (which is the terminal of the Arduino board in which we have connected the LED).

Activity 1: turn on and turn off a LED

Writing our first program

Definitions:

If at any given moment we decide to change the LED connection to another terminal, for example to 7, we would only have to modify the `#define` by:

`#define LED 7`

With this simple modification, everything will work correctly without having to make any additional changes to the program.

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

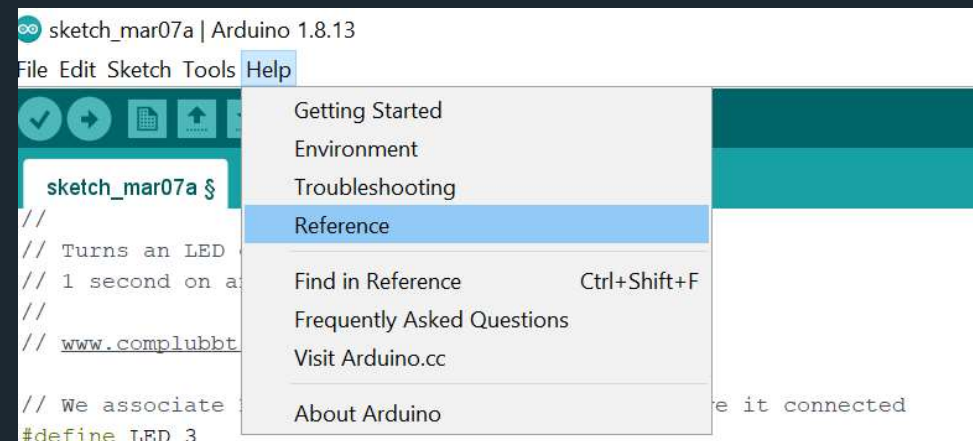
There are a number of functions that are not part of the C programming language. These added functions allow to simplify certain tasks related to the Arduino UNO board itself. The number of these functions is quite large.

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

We can see them by selecting
Help >> Reference in the text menu
of the Arduino IDE



[file:///C:/Program%20Files%20\(x86\)/Arduino/reference/www.arduino.cc/en/Reference/HomePage.html](file:///C:/Program%20Files%20(x86)/Arduino/reference/www.arduino.cc/en/Reference/HomePage.html)

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

In this first program we have used three of these functions.

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

pinMode (*pin*, *mode*)

digitalWrite (*pin*, *estate*)

delay (*time*)

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

digitalWrite (pin, *estate*)

Allows you to **change the modus of a pin** on an Arduino board that we have **previously defined as output**. We can activate it (**HIGH**) with which the pin will be set to 5V, or deactivate (**LOW**), with which said output will be 0V.

In our activity we use this function to **turn** the LED **on** or **off**.

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

delay (time)

Time function that stops the operation of the program for a time specified in ms (1s =1000 ms).

In our activity, it allows the LED to remain on or off for the desired time.

Activity 1: turn on and turn off a LED

Writing our first program

Arduino primitive functions:

pinMode(pin, mode)

It allows defining the behaviour of a pin on the Arduino board **as an input (INPUT)** or **as an output (OUTPUT)**.

In our activity we have defined the pin where the LED is connected as an output.

Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

Once we have written our program in the Arduino IDE editor, it is convenient **to verify** that we have not made any syntactic errors, that is, that everything is written correctly.

This check does **not imply that a program works correctly**. It only indicates that there are **no errors in the way of writing** what we have put.

Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

To do this check we have to press the check button (the first one from the left) of the quick functions menu:



Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

If everything is written correctly, a message similar to this one will appear in the message window:

Done compiling.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

Indicates that the compilation process has been successful and has generated code.

Done compiling.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

6

The generated code has a size of 924 bytes out of a maximum of 32,256 bytes that our program could have on an Arduino UNO board (Internal FLASH memory of the Arduino UNO board, which is 32K (32,768 bytes))

Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

On the other hand, if we have made some kind of syntactic error, the message will be similar to the following:

In this case, the error that appears indicates that we have misspelled the `pinMode` function. (We have put the "m" in lowercase). To solve this problem we just have to correct the text and press the Verify button again.

GINESJGARCIA

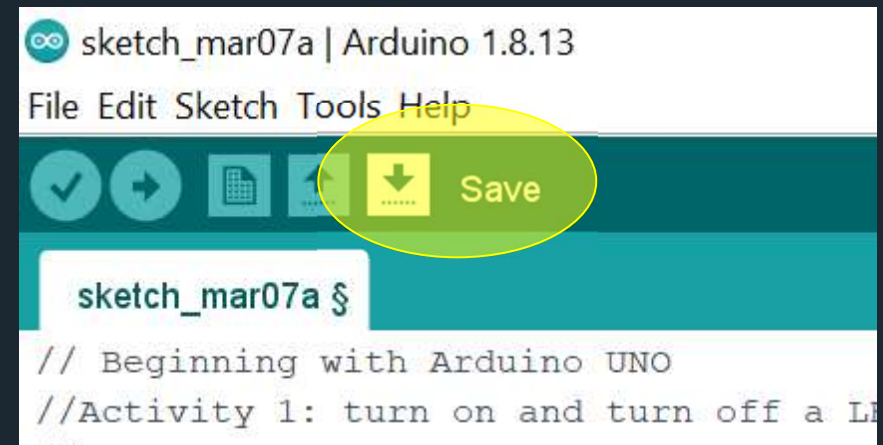
```
1 // We configure the digital pin of the LED as output
2 pinMode(LED, OUTPUT);
3 }
4
5 void loop ()
6 {
7   // We turn on the LED activating the digital output
8   digitalWrite (LED, HIGH);
9   //Esperamos 1 segundo
10  delay (1000);
11  // We turn off the LED by deactivating the digital output
12  digitalWrite (LED, LOW);
13  // We wait 1 second
14  delay (1000);
15
16  'pinmode' was not declared in this scope
17  pinMode
18  exit status 1
19  'pinmode' was not declared in this scope
```

Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

Once we have the program correctly written we are going **to save** it before doing any more operations. To do this, click on the fifth button of the quick functions menu:



Activity 1: turn on and turn off a LED

Writing our first program

Check the syntax of the program:

As this is **the first time that we have saved** this program, a **dialog box** will appear where we will be asked for the **name of the file and the place** where we want to save it.

It is recommended to **create a folder** that we can call "**Arduino_Activities**" within the working directory (Documents in Windows), and save this program as **Activity_01**.

Arduino will save the program inside a folder that will have the same name as the one we have chosen for the program. A good way to organize your information!

Activity 1: turn on and turn off a LED

Writing our first program

Loading and running the program:

And we come to **the most exciting part** of our first activity. Where we will see if everything we have done is correct and the result that this provides us.

This last step that we are going to carry out has, for its part, a certain internal complexity, but for us it will be as simple as **connecting the Arduino UNO board to the computer and pressing a button.**

Activity 1: turn on and turn off a LED

Writing our first program

Loading and running the program:

Specifically, the second button, from the left, of the quick functions menu:

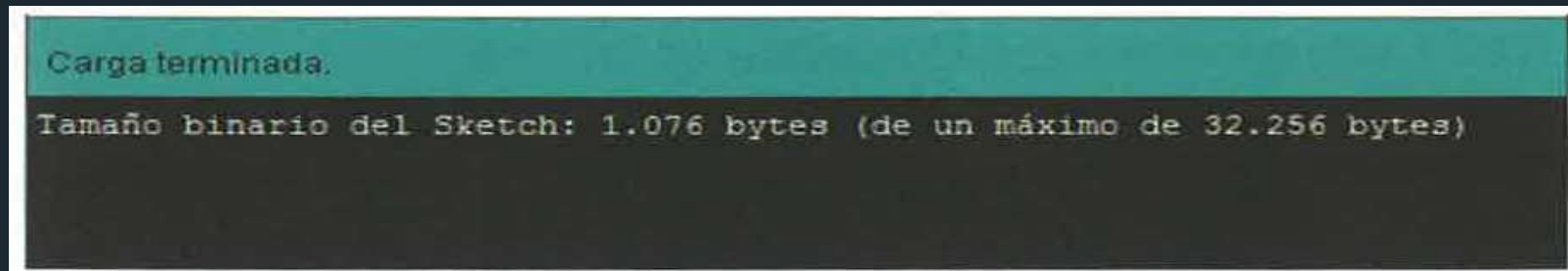


Activity 1: turn on and turn off a LED

Writing our first program

Loading and running the program:

When pressed, **the program** that we have written in the editor **will be verified (again), compiled and loaded** onto our Arduino UNO board. At the end of this process the message window will show us something like this:



Activity 1: turn on and turn off a LED

Writing our first program

Loading and running the program:

Immediately and automatically, the execution of the program will begin. If everything has gone well, the LED will turn on and off in 2 second intervals: 1 second on and the next off.

Activity 1: turn on and turn off a LED

Writing our first program

Proposed exercises:

Below, you will find some proposed exercises that will help you better understand the operation of the circuit and the program. Don't stop doing them. They are a good way to practice before continuing!

Activity 1: turn on and turn off a LED

Writing our first program

Proposed exercises:

Exercise 1.1- Modifying the blink frequency

The blink frequency or number of times the LED lights up per second can be changed by adjusting the times of the delay function. In the current activity the LED turns on once every 2 seconds, this implies a frequency of 0,5 Hz (Hz = times per second). Now we suggest that you modify the program so that the LED flashes at a frequency of 1Hz.

Activity 1: turn on and turn off a LED

Writing our first program

Proposed exercises:

Exercise 1.2 - Emitting flashes

We can make the LED flash, making the on time shorter than the off time. Modify the program so that the switch-on time is half the time the LED is off.