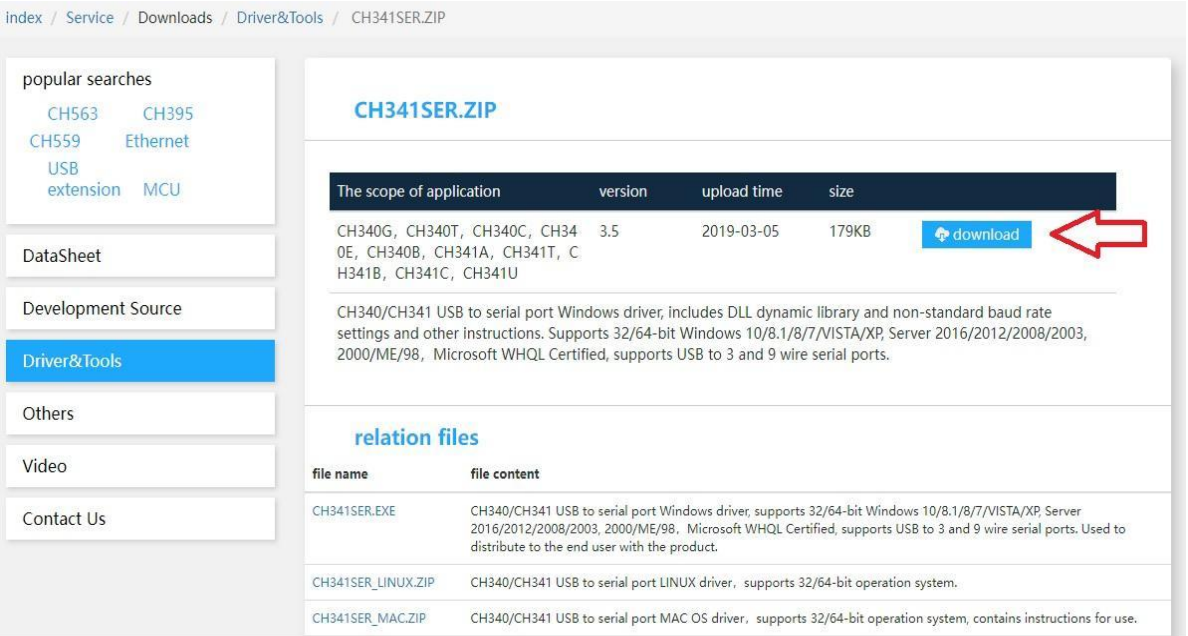
LEDBOY

Instructions for programming and using the hardware.

Below you can find a tutorial on how to use all the functions of the device as well as hardware information.

1. Install drivers for the serial programmer and be able to upload code.

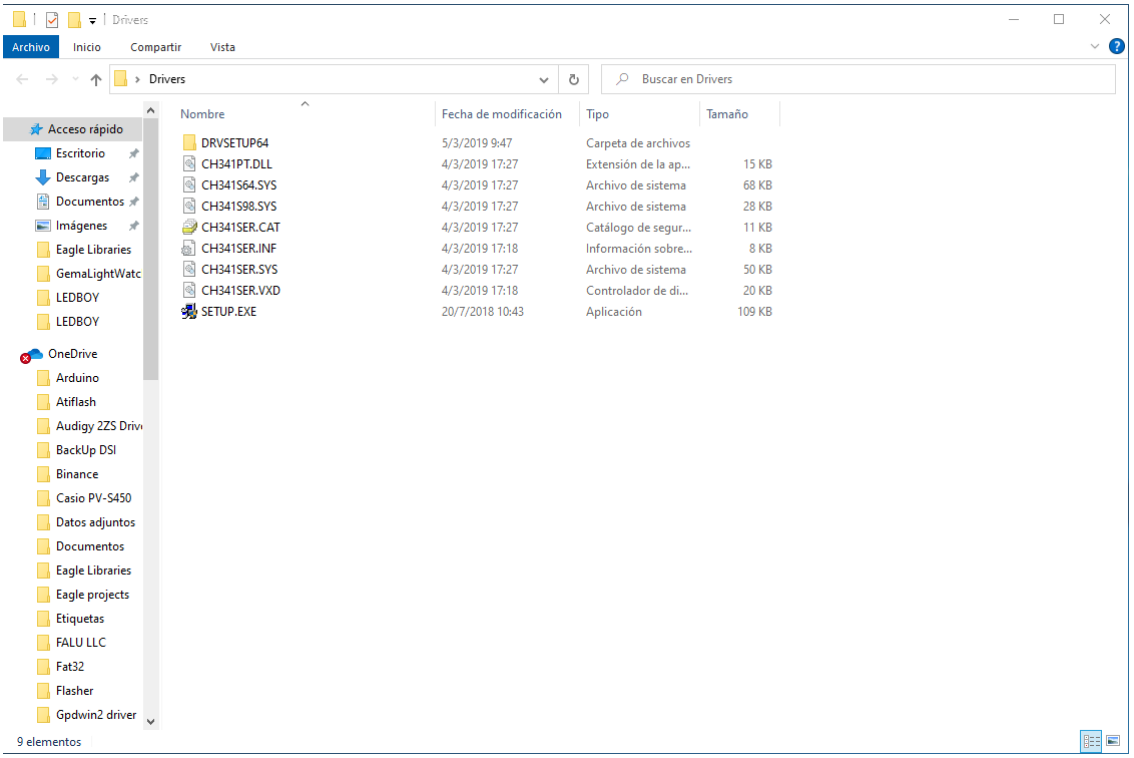
-Download the driver for the CH340C serial adapter



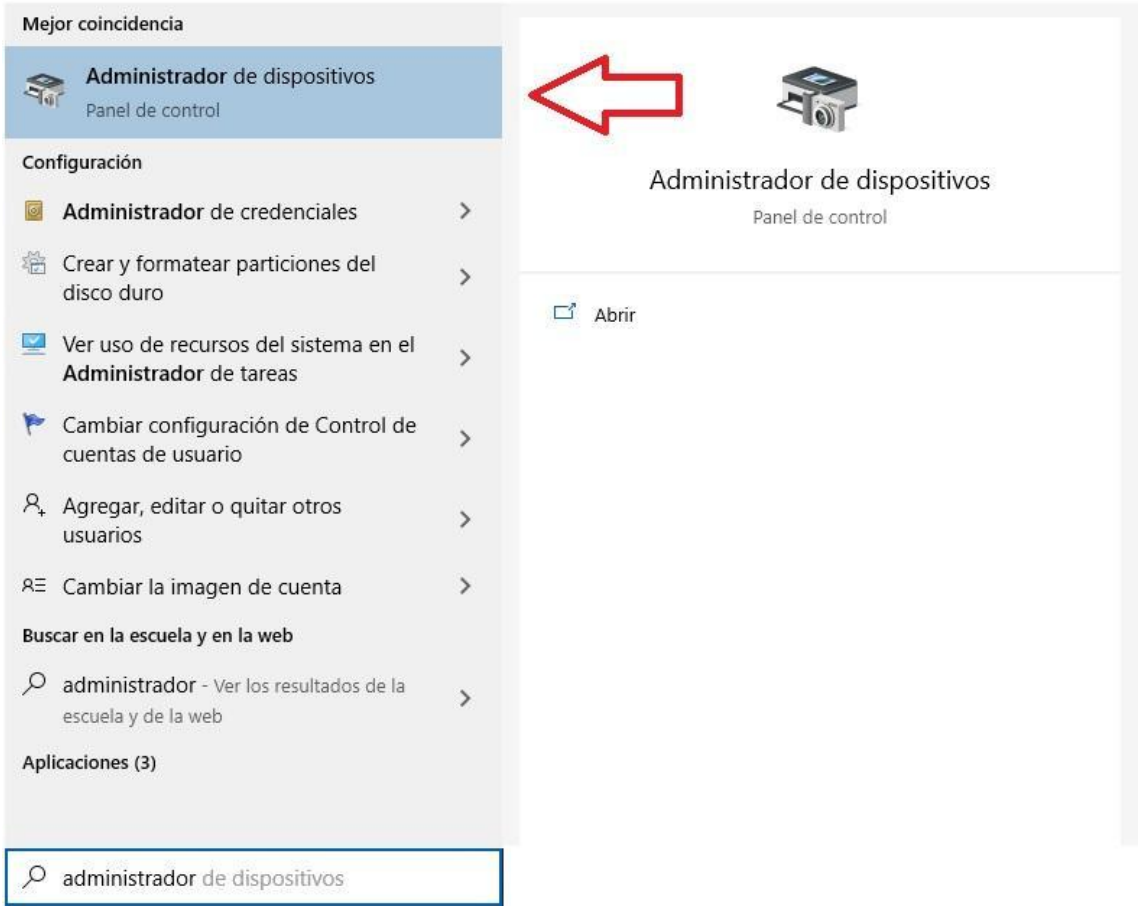
http://www.wch-ic.com/downloads/CH341SER_ZIP.html

These are compatible with: Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98.
Then the installation will be shown in Windows 10, but it is applicable to all Windows systems that the driver is compatible.

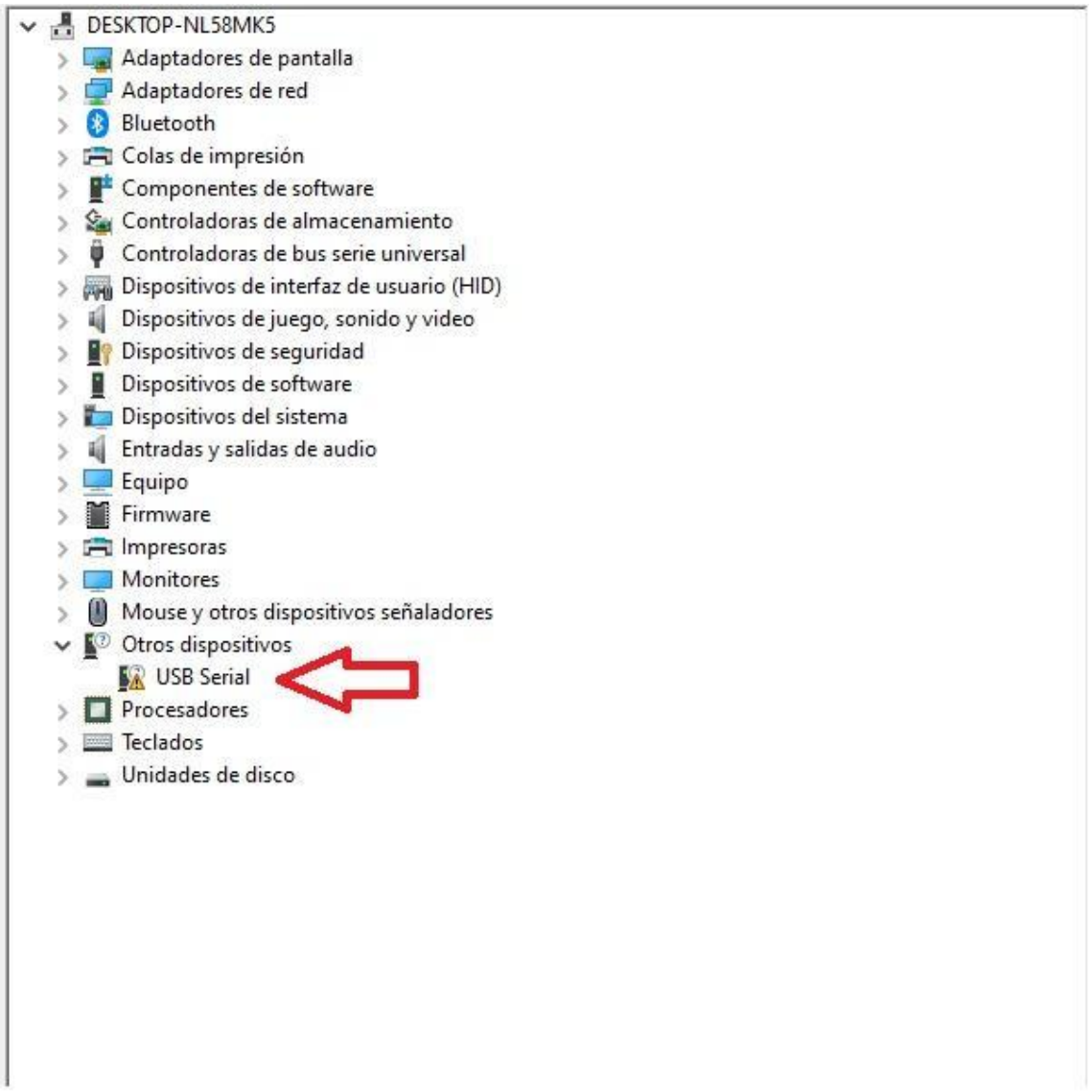
After downloading the file .zip extract all its contents to a new folder on our desktop that we can call "Drivers".
When finish, there is a folder containing all drivers.



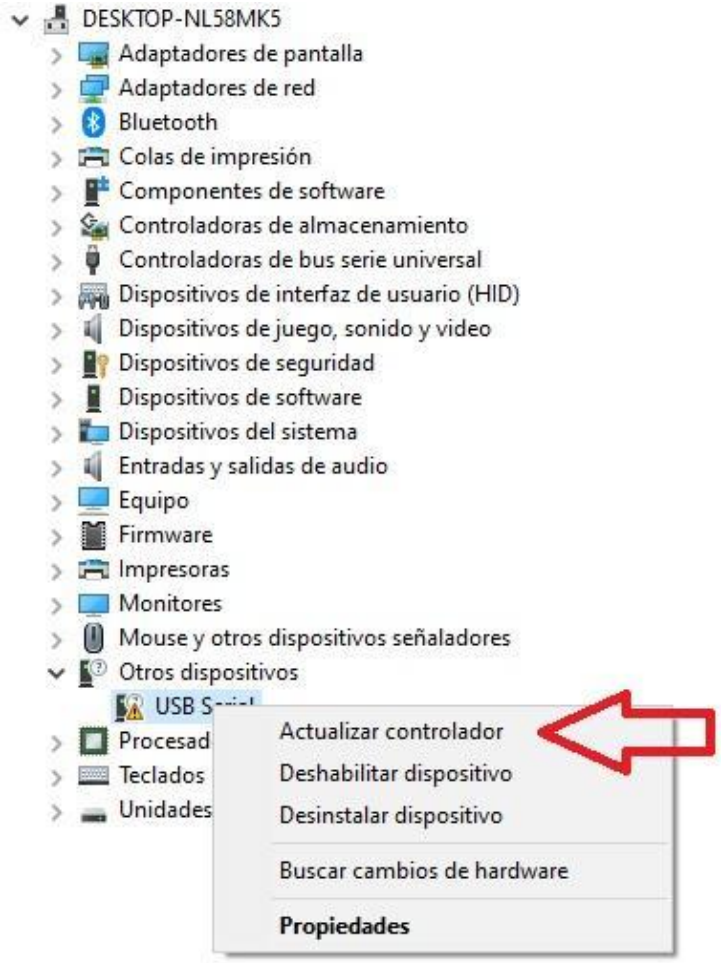
Then do a search in Windows for: DeviceManager.



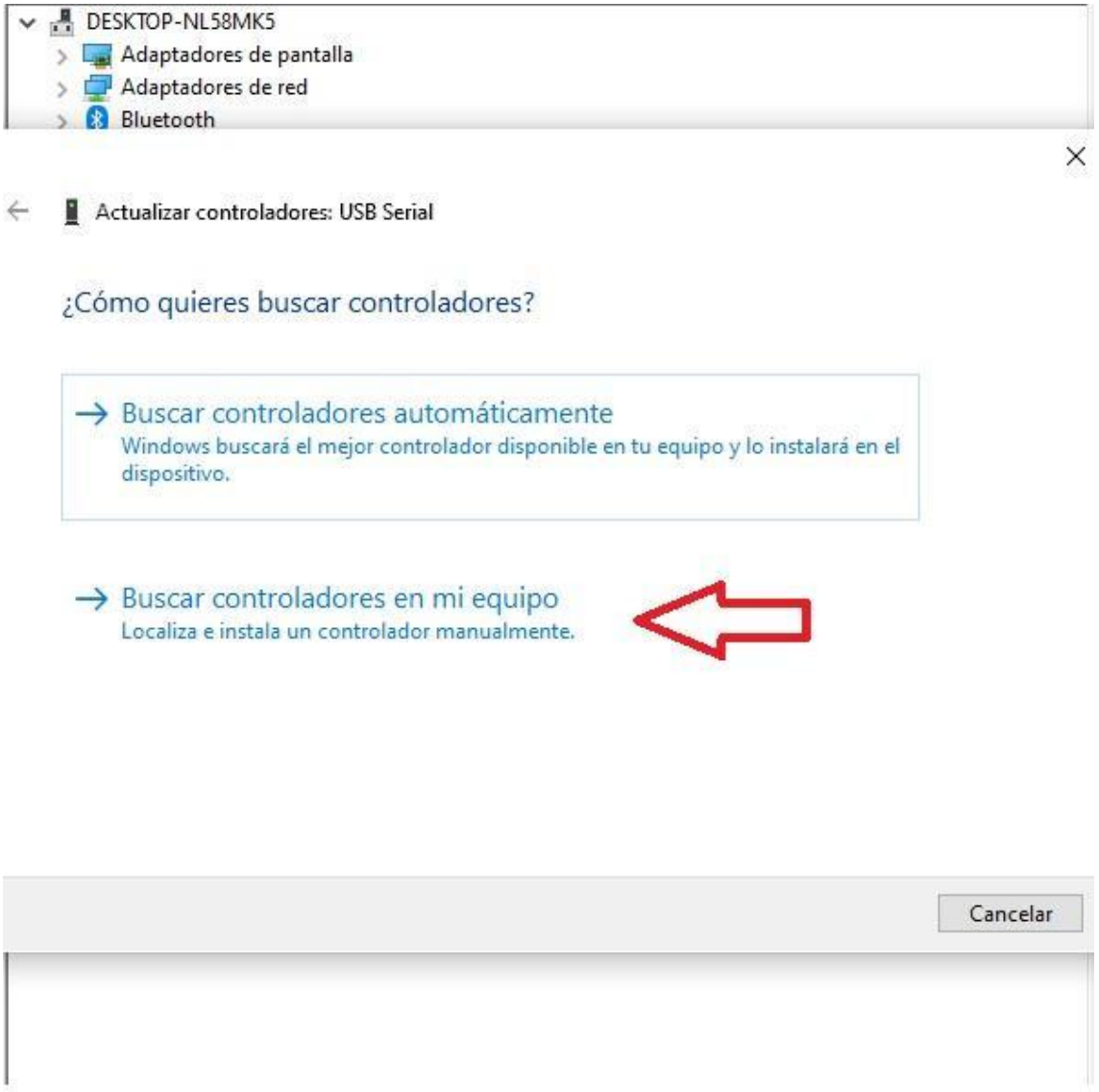
Next, connect our LEDBOY to a USB port and search the device manager under "Other Devices"
“USB Serial”



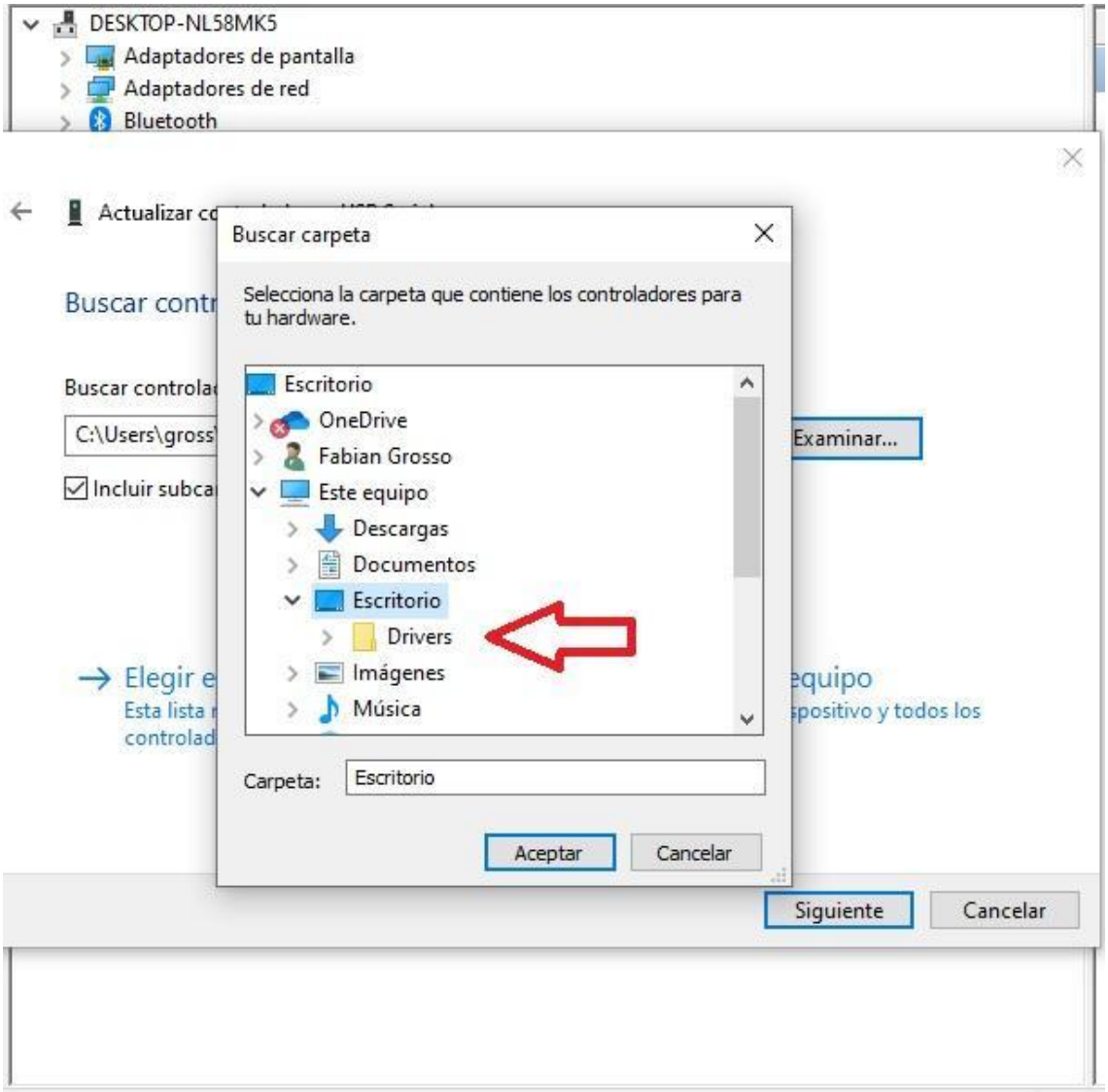
We right-click on "USB Serial" and the following window will appear, click "Update driver"



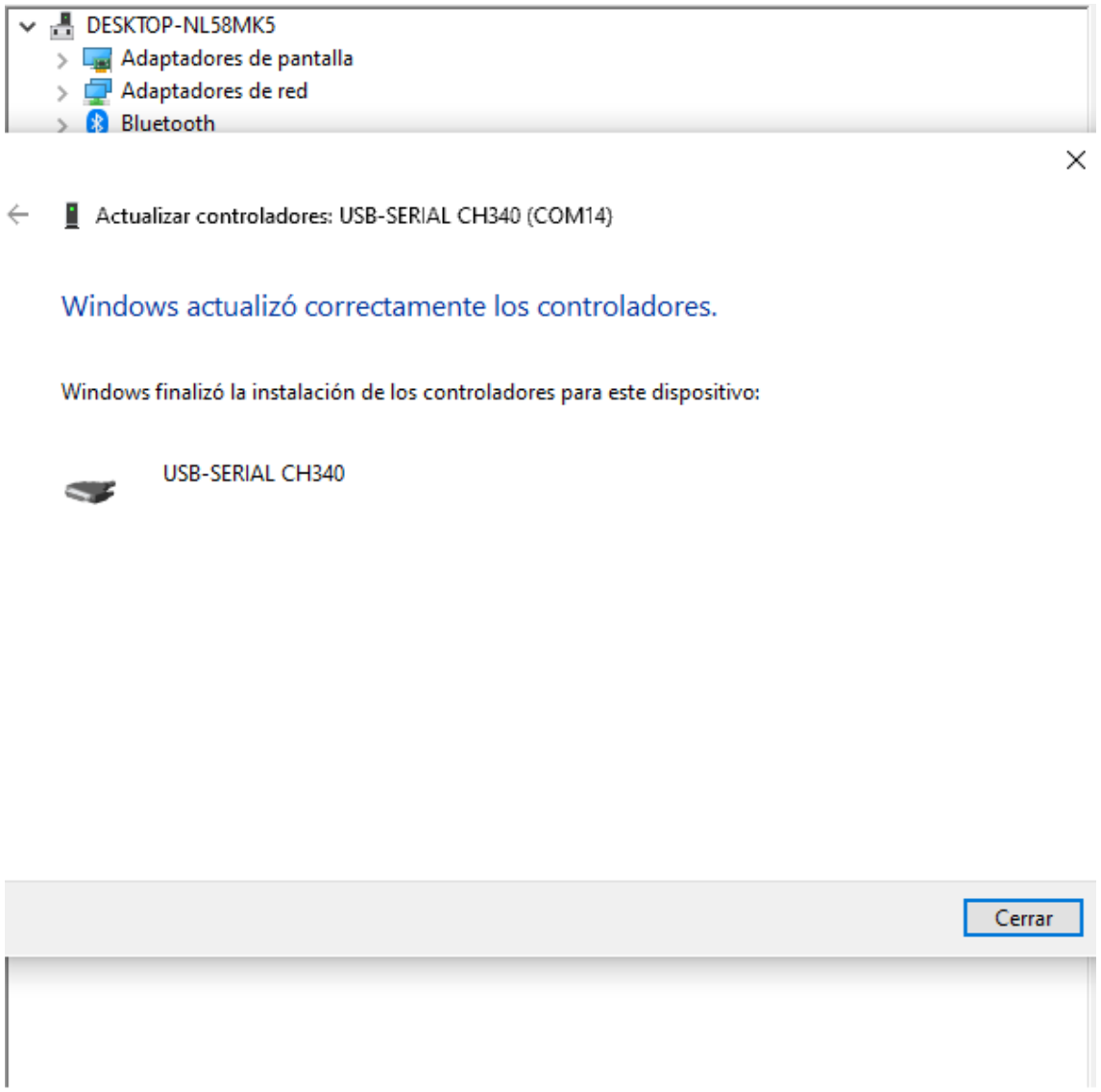
Then choose the option to search for drivers on the computer



Chose the "Browse" option and looked for the "Drivers" folder created earlier with the driver files.



Accept and then give the Next option, if everything goes well, this message will be shown, and the drivers will already be installed.



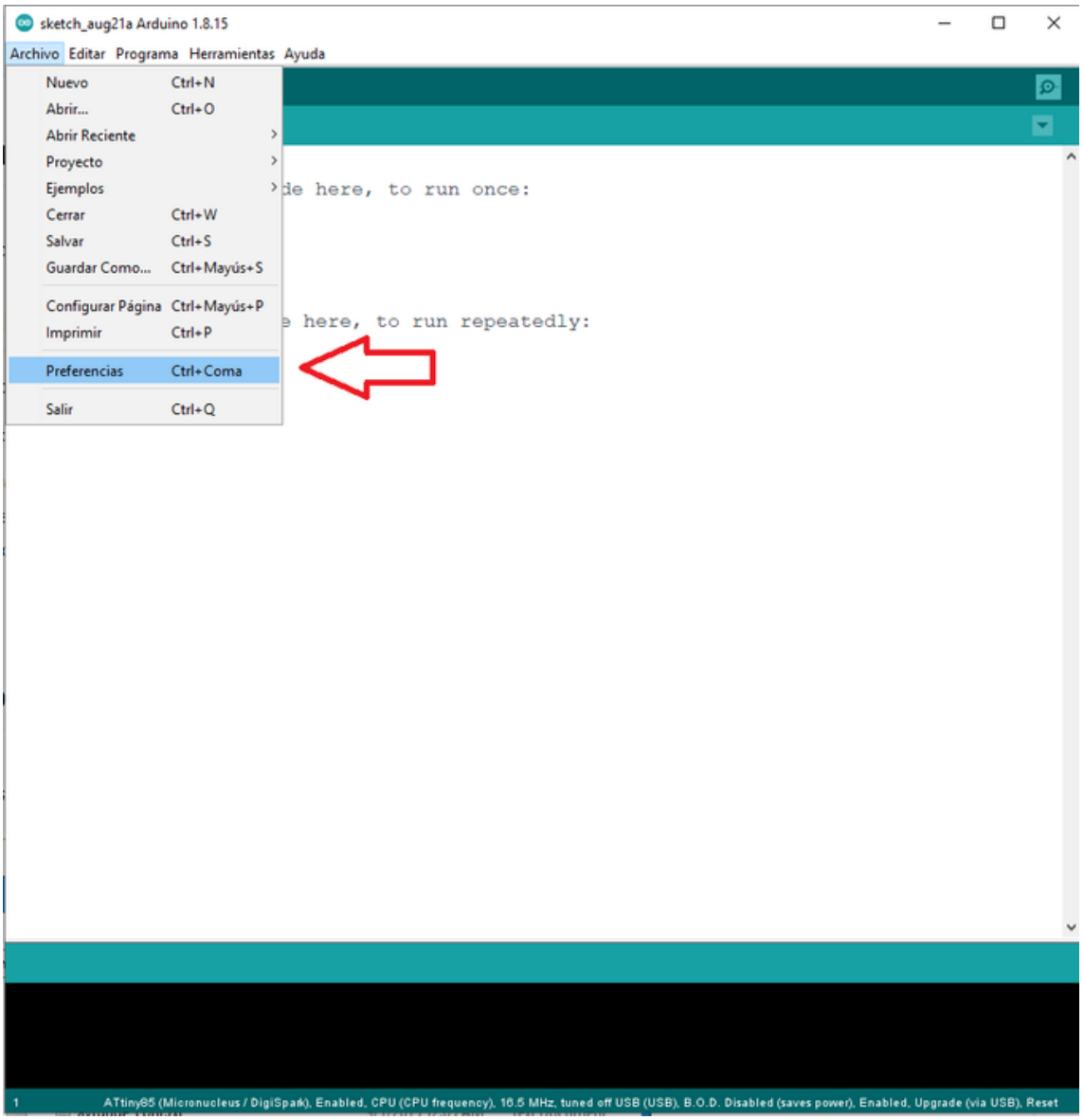
Now close all windows and continue installing the IDE to program

.

2. Preparing to program the console.

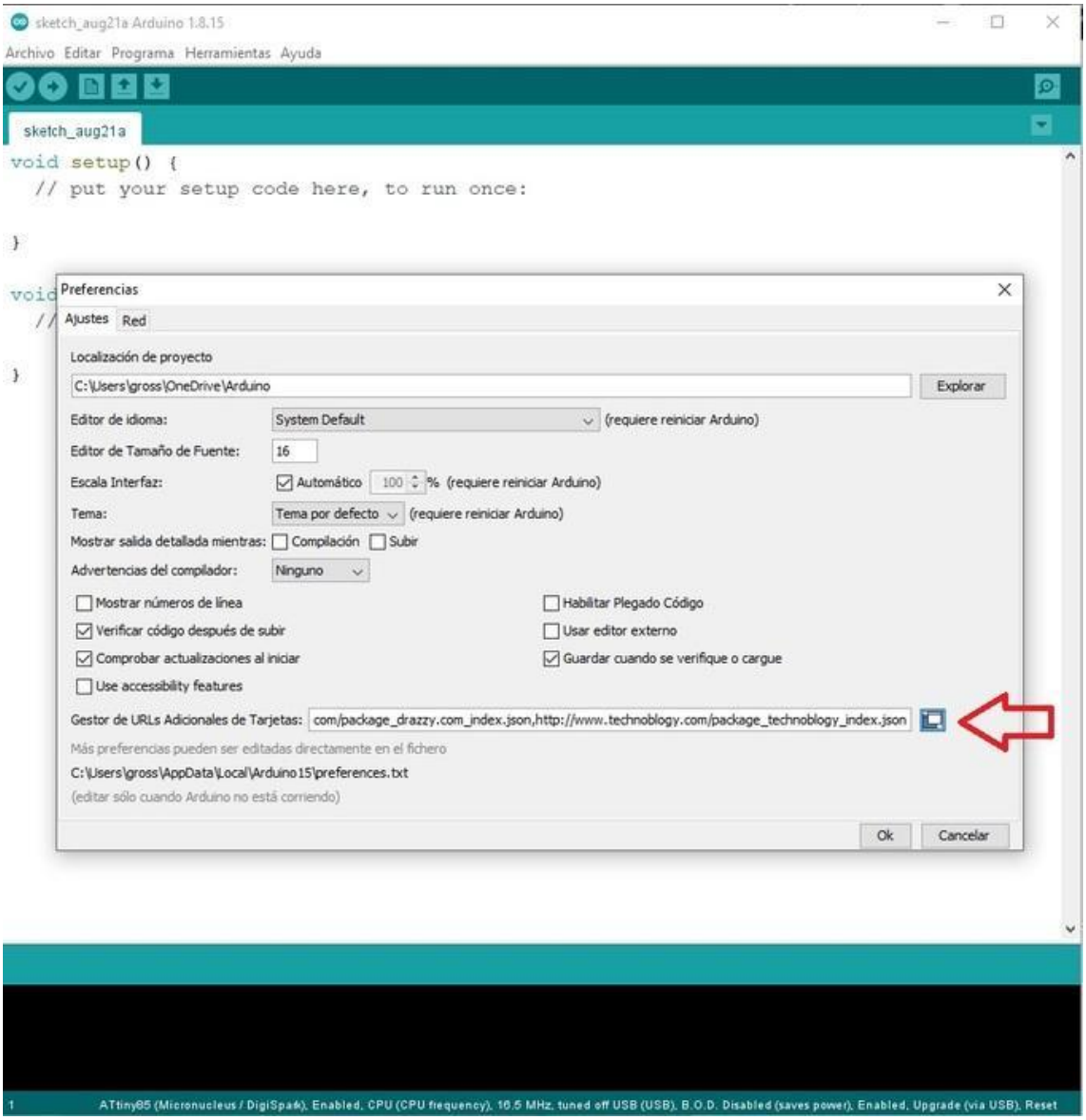
LEDBOY uses the Arduino IDE to program itself
-Download the Arduino IDE: https://www.arduino.cc/en/software
Download the desired version (minimum recommended version 1.8.13).
Next, we will need to make an initial configuration.

-Manager of tarjetas:
Open the downloaded program and a window will open with a blank project.
Proceed to select the preferences option in the files tab.

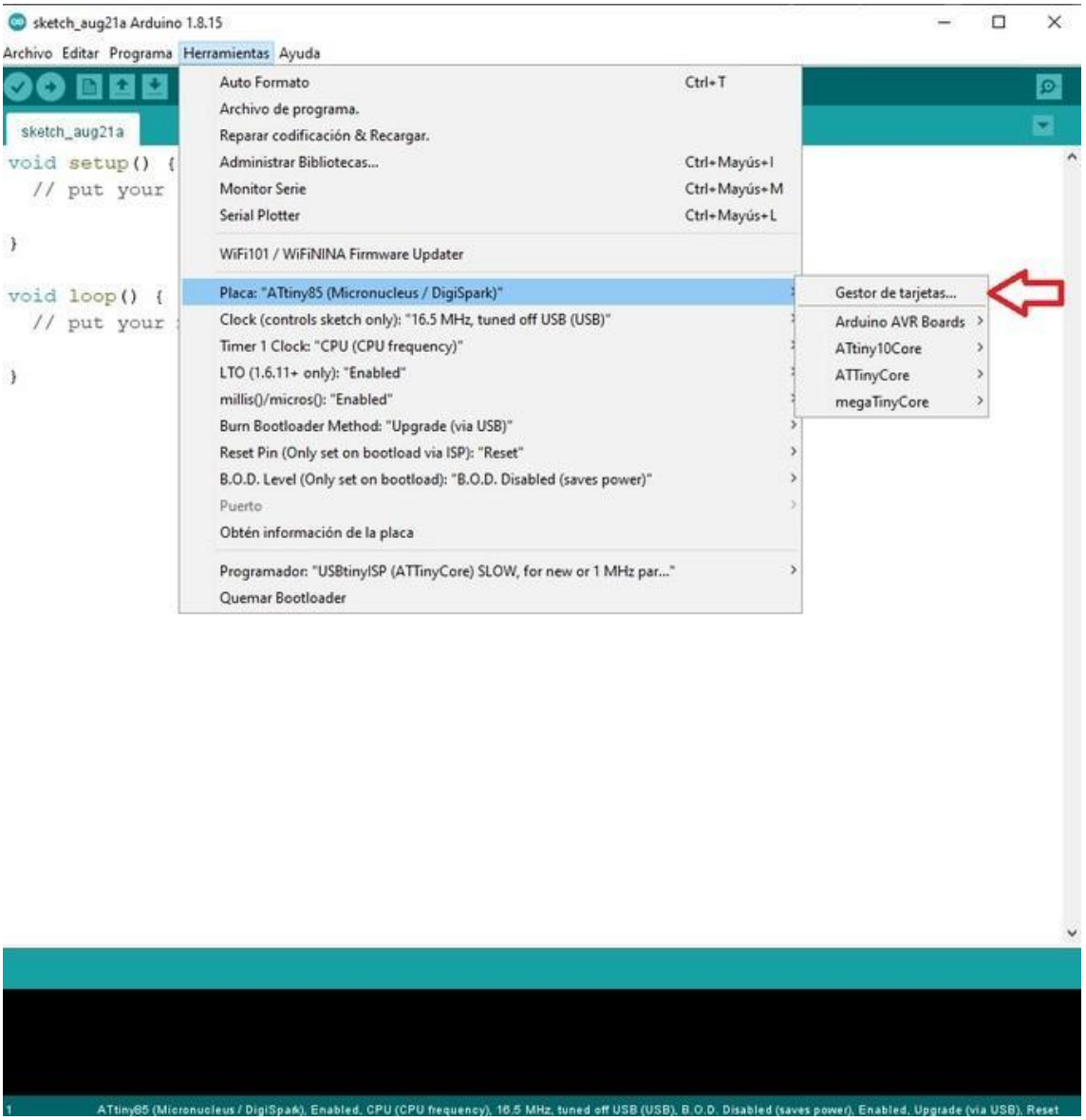


Once the window is opened, enter the following URL:

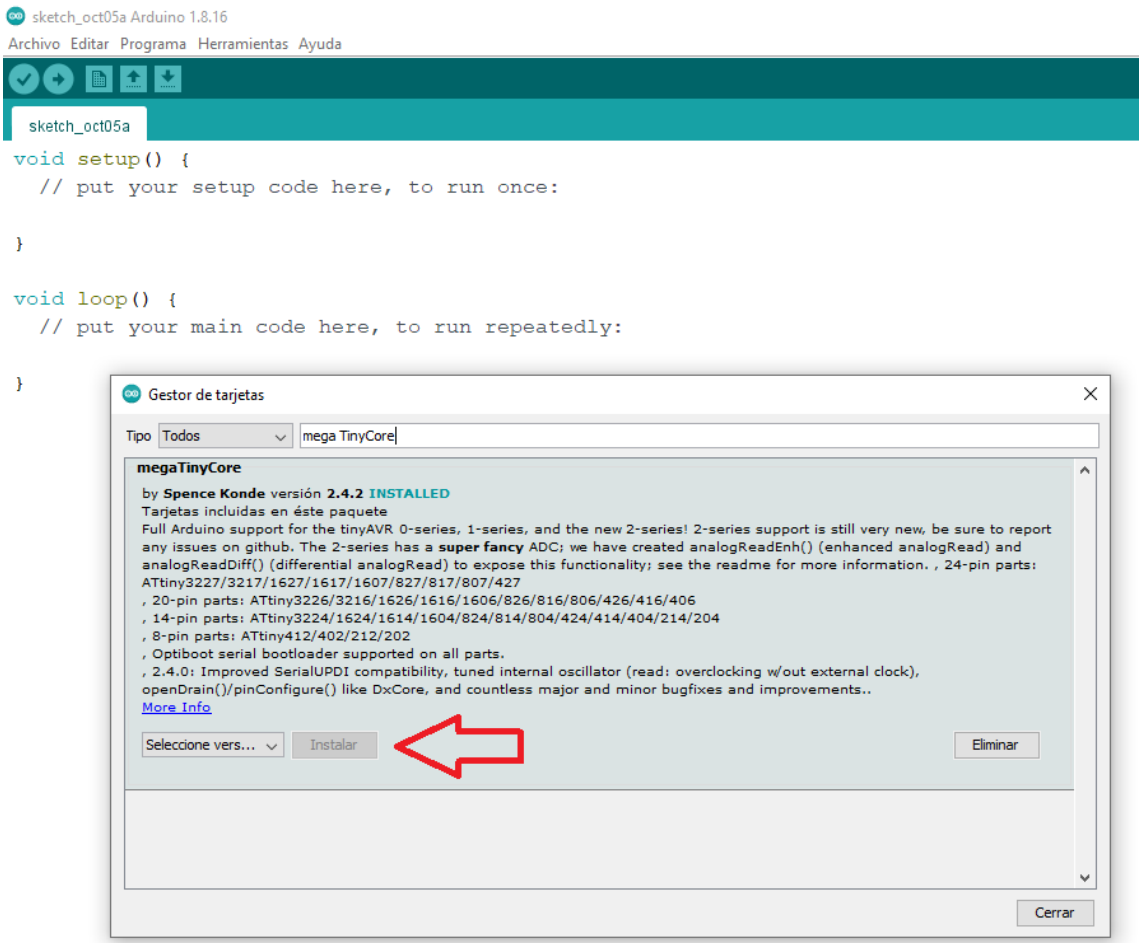
http://drazzy.com/package_drazzy.com_index.json



If there is a URL already entered, it is not necessary to delete it, we simply add the new one below. We give OK to all windows to save the changes. Then we must go to Tools-Board-Card Manager.



When the card manager opens we will need to look for **mega TinyCore** and install it to have support for the Attiny 1614.



Once this is done, support to program the microcontroller integrated in LEDBOY is installed.

The good thing about the Arduino IDE is that there are many libraries available to execute functions or integrate peripherals in a very simple way, you can install as many as you need since these are installed on the PC but when integrating them into the project you must be careful since these consume resources of the microcontroller, and we can easily run out of space.

3.Create a program, compile it and upload it to the microcontroller
Also applicable for downloaded games or programs.

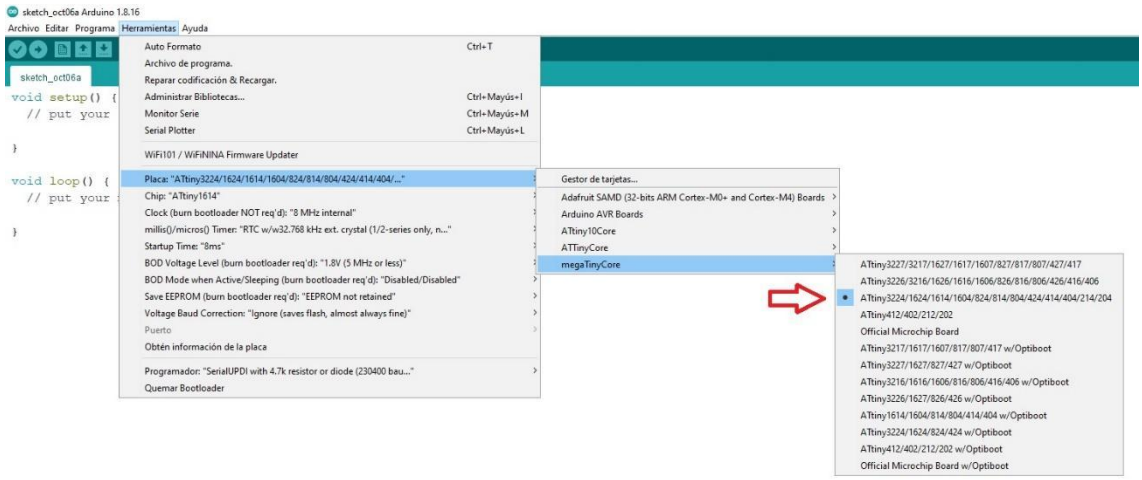
To create an Arduino program : first of all, integrate the libraries needed for our program, this provides us with functions to facilitate the use of hardware, for that you can copy the following text at the beginning of the sketch without deleting anything.

```
#include <tinyNeoPixel.h>
#include <avr/sleep.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>

// pins
#define NUMLEDS 100
#define RIGHT PIN_PA1
#define LEFT PIN_PA2
#define SLEEPMENU PIN_PA3
#define MOSFET PIN_PA4
#define ROTARYA PIN_PA5
#define ROTARYB PIN_PA6
#define EXT PIN_PA7
#define NEOPIN PIN_PB0
#define BUZZER PIN_PB1

tinyNeoPixel strip = tinyNeoPixel(NUMLEDS, NEOPIN, NEO_GRB);

void setup() {
pinMode(SLEEPMENU, INPUT);
pinMode(LEFT, INPUT);
pinMode(RIGHT, INPUT);
pinMode(ROTARYA , INPUT);
pinMode(ROTARYB , INPUT);
pinMode(EXT, INPUT_PULLUP);
pinMode(BUZZER, OUTPUT);
pinMode(MOSFET, OUTPUT);

digitalWrite(MOSFET, LOW);// P CHANNEL mosfet low to activate
strip.begin();
strip.setBrightness(50); // set all pixels brightness
```

```
strip.setPixelColor(0, 50, 0 , 0); //set individual pixels color in dec. RGB
strip.show();
}
```

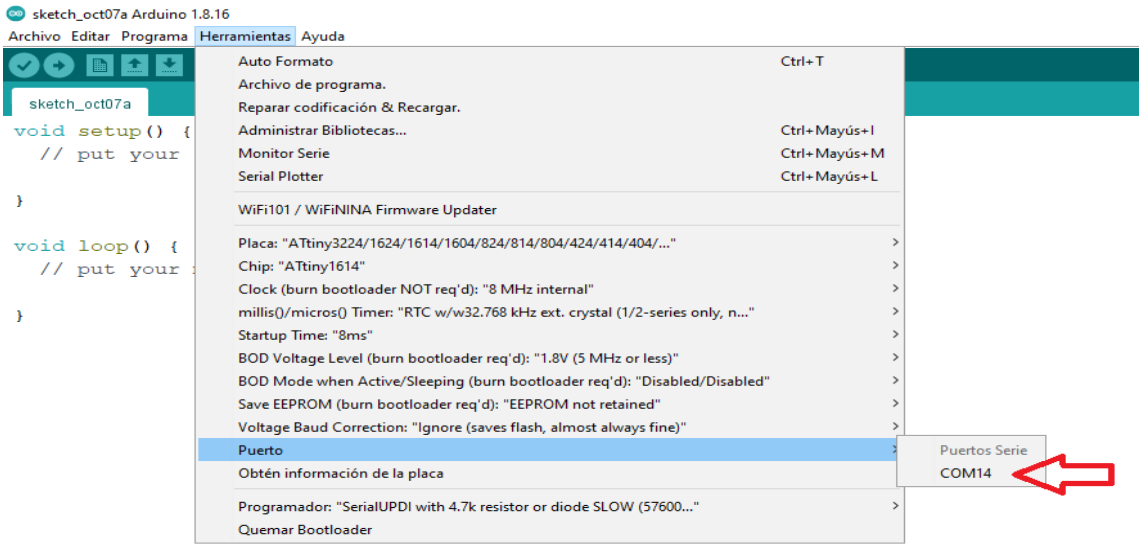
Now you are ready to write your programs and make use of all the functions to program your LEDBOY.
We recommend looking at our sample programs/tutorials on our channel for more examples.

Then select these parameters in the Arduino IDE before uploading our program.



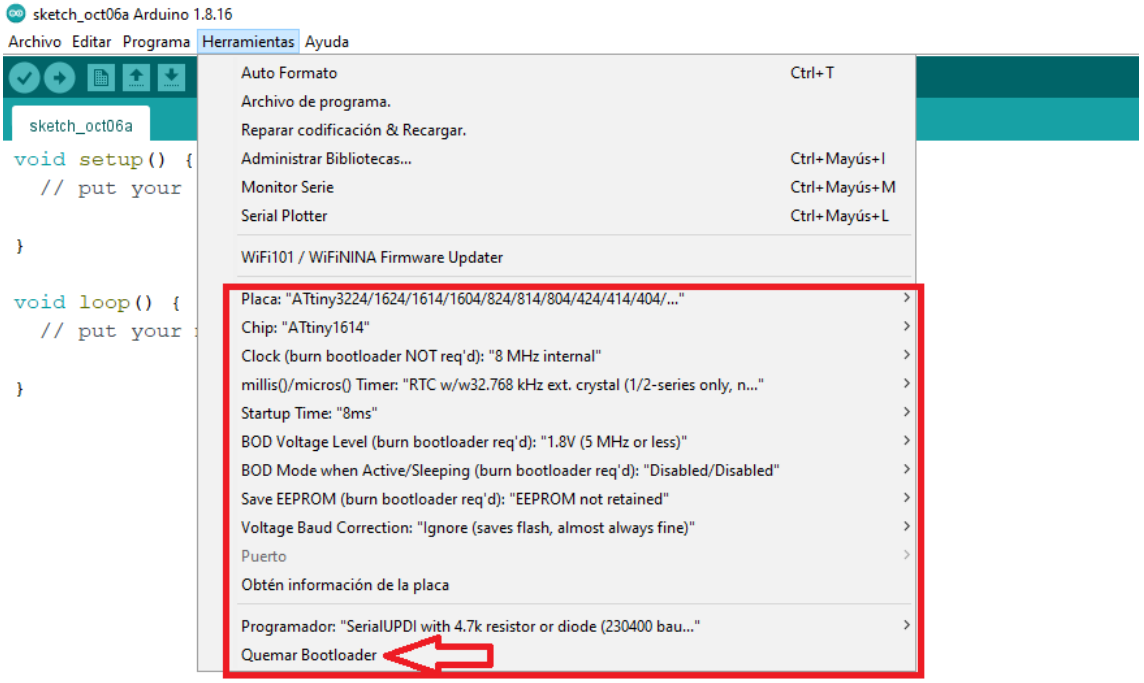
In tools, board, Mega Tiny core we must choose the microcontroller Attiny 1614.

Before flashing LEDBOY choose the correct COM port, if we have a single COM port that is the correct one.
If there is more than one and we do not know which one corresponds to our console.
Simply unplug the USB cable from our LEDBOY we leave the options, we go back in and see that port is no longerthere, the COM port that is not when disconnecting the console is the one that corresponds to LEDBOY.
Always have it selected.



Mainly need to be careful NOT to select any option that has"Optiboot"unless we are sure of what we do, in this option it does notgive the possibility to change the behavior ofl PINUPDI that our LEDBOY uses to upload programs,if we choose the wrong option WE WILL NOT BE ABLE TO RE-UPLOAD MORE PROGRAMS.

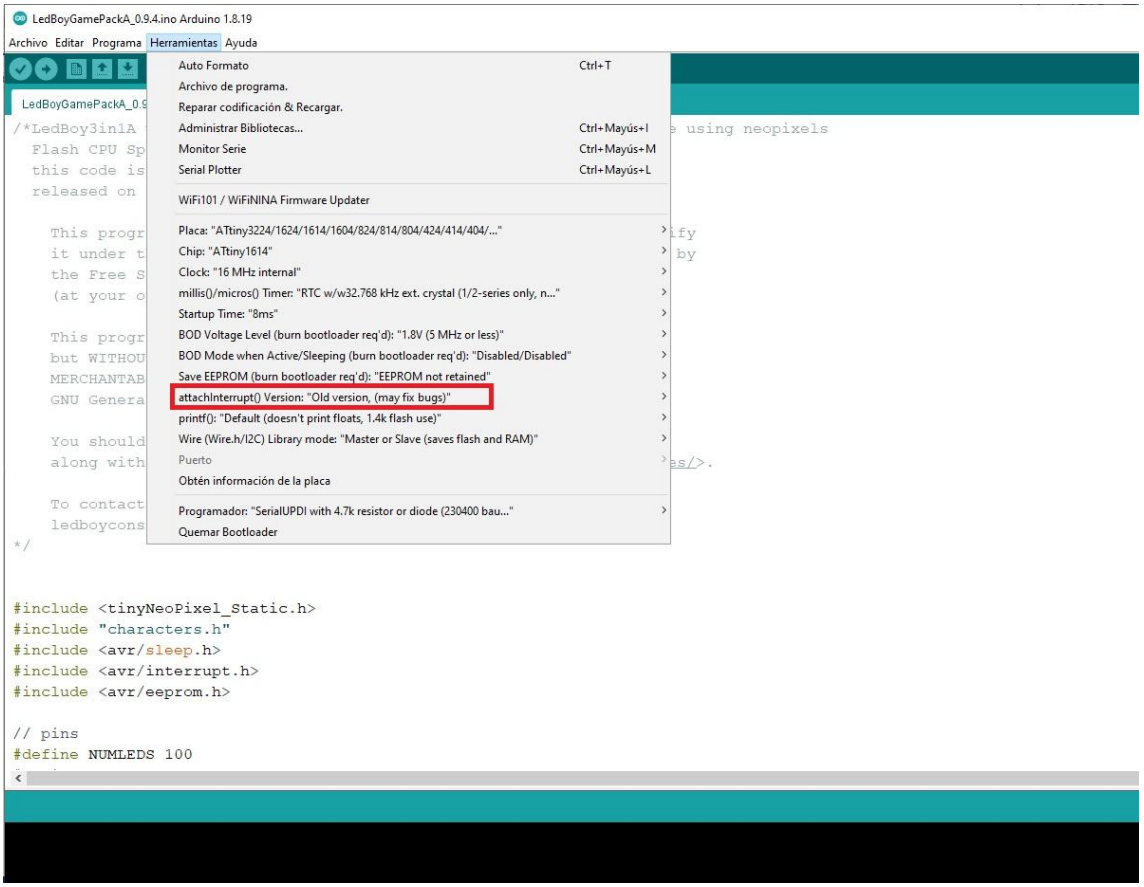
It can only be reversed with a special 12vprogrammer.

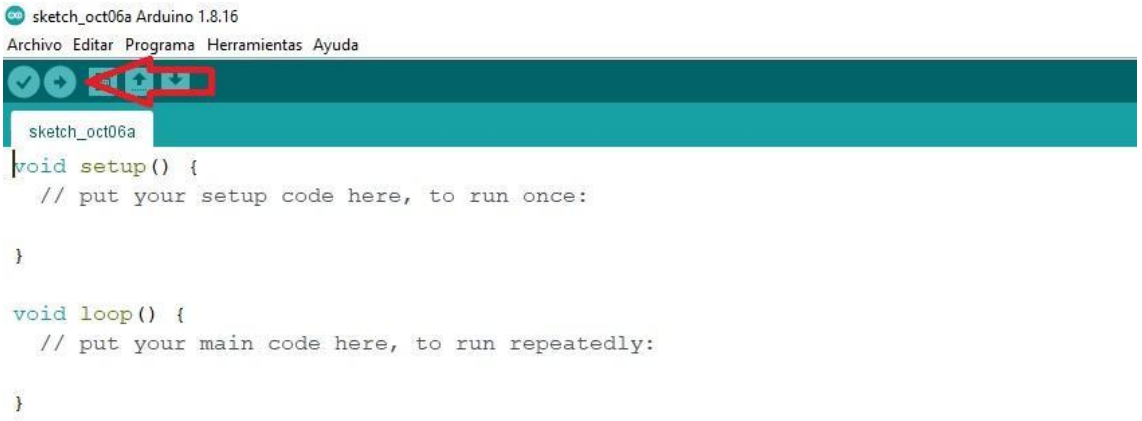


Before uploading our program please check that the options shown in the image match those we have selected in our program.

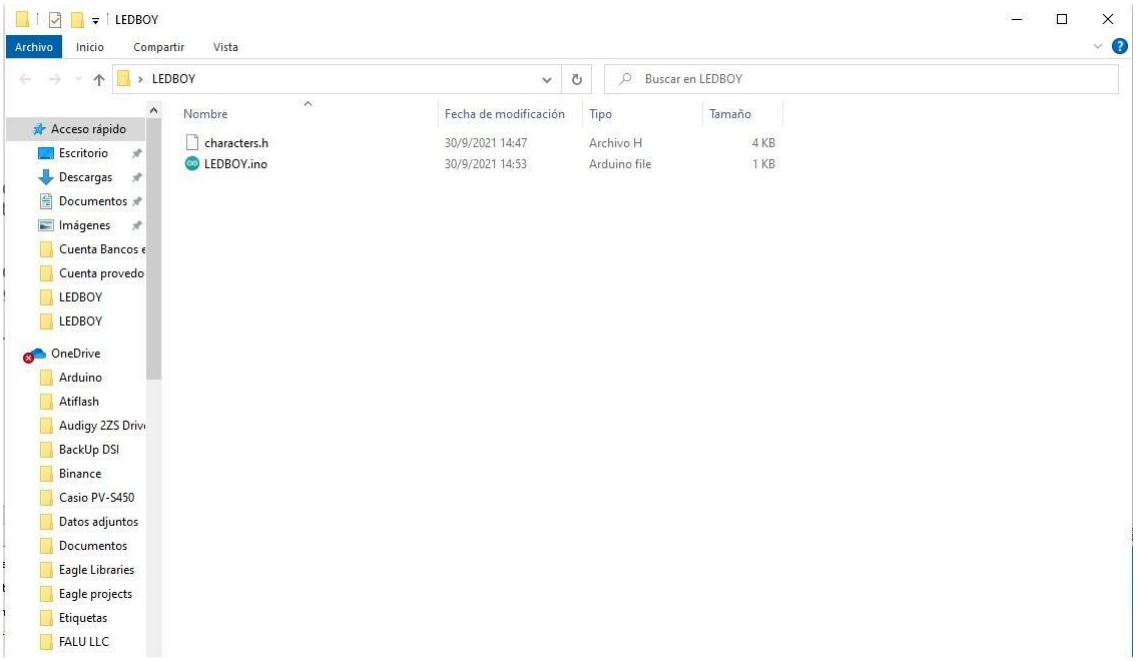
"burn bootloader" is not necessary to use if the console already has a program installed, this option is only to configure flags on the microcontroller, this is done only once.
It is ONLY necessary to do it again in case a program specifies it or we want to change some parameter.

Latest version of megaTiny core changed the way interrupts work, if you are having problems try to set interrupts option to the old version.





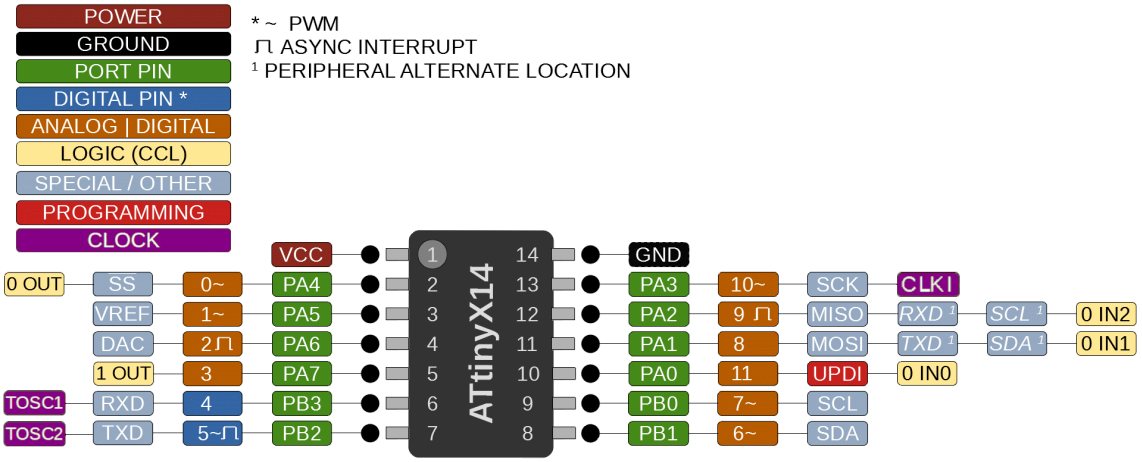
After we have everything selected, press the arrow which will compile and upload the program to our LEDBOY.
In the red boxed area, the progress will be shown and in the console we should already see the program working.
An average flashing only takes 10sec or less.

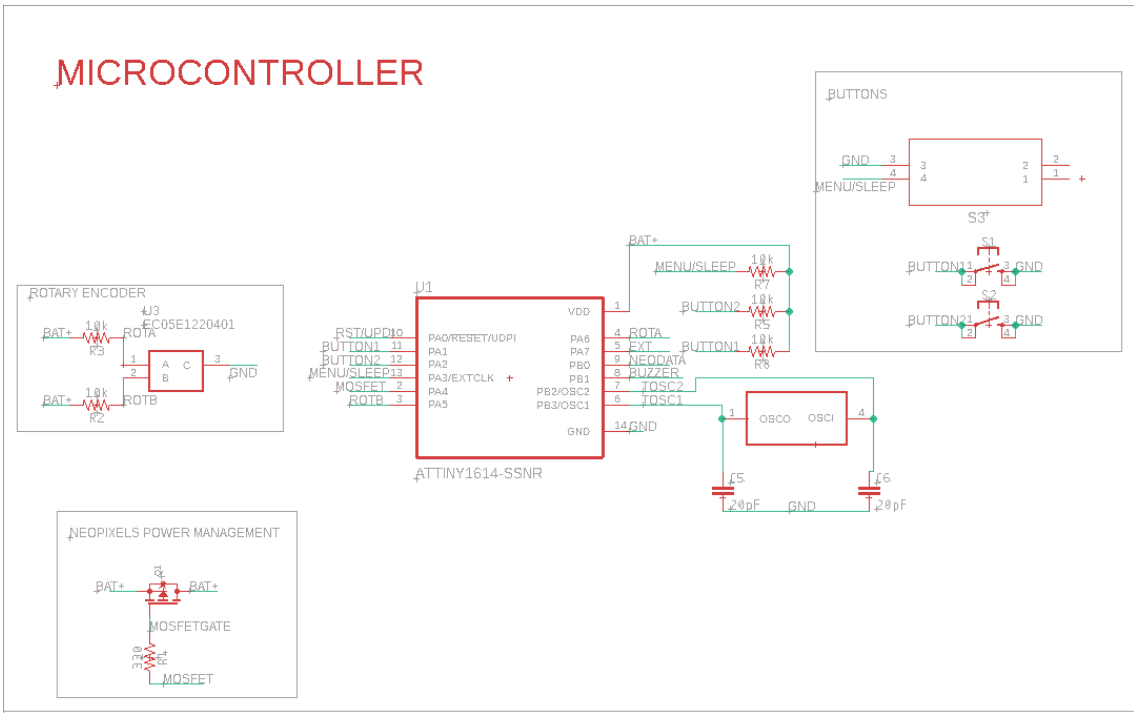


If we have downloaded a program or game, we will have a folder with the name of the program and within this an **.ino** file that is the one that we will have to double click and this will open in the Arduino environment, we can upload it with the method already described above.
It is important to keep all the files inside the folder and not to modify the names.

4.Technical specifications.

-Attiny 1614 microcontroller





-UPDI programming interface: CH340c USB to serial adapter
-Screen: 10 x 10 Nano (2.4mm x 2.7mm) RGB LED 2427 (24-bit color)
-Interface: 3 buttons,1 rotary encoder
-Battery: Lithium-ion 250mha.
-Charging: Max 1555 300mha
-Average consumption at 16 MHz: 30-130mha (depends on led brightness)
-Case: PLA+