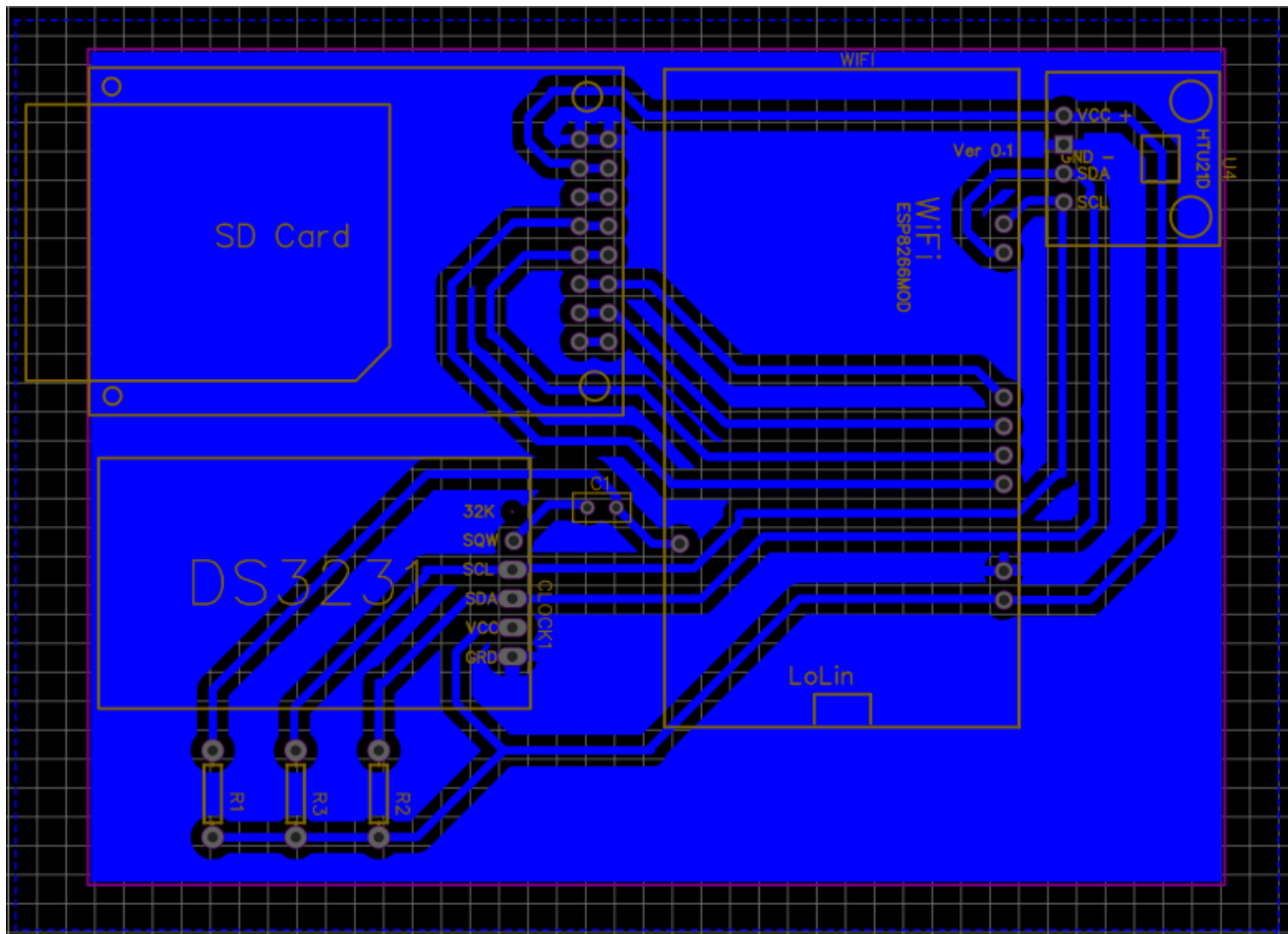


ASSEMBLY TUTORIAL



Sumário

1	PCB Board	2
2	Installing Arduino	2
3	Setting up the Environment	2
3.1	Add ESP8266 Board URL	2
3.2	Install ESP8266 Drivers	2
3.3	Install Required Libraries	3
4	Uploading the Sketch	4
4.1	Select the Right Devices	4
4.2	Upload the Code	6

1 PCB Board

In order to make things more practical, you can demand a PCB board using the gerber files of the project. Also, you can modify the original design at <https://oshwlab.com/gui.br57/low-cost-weather-station>.

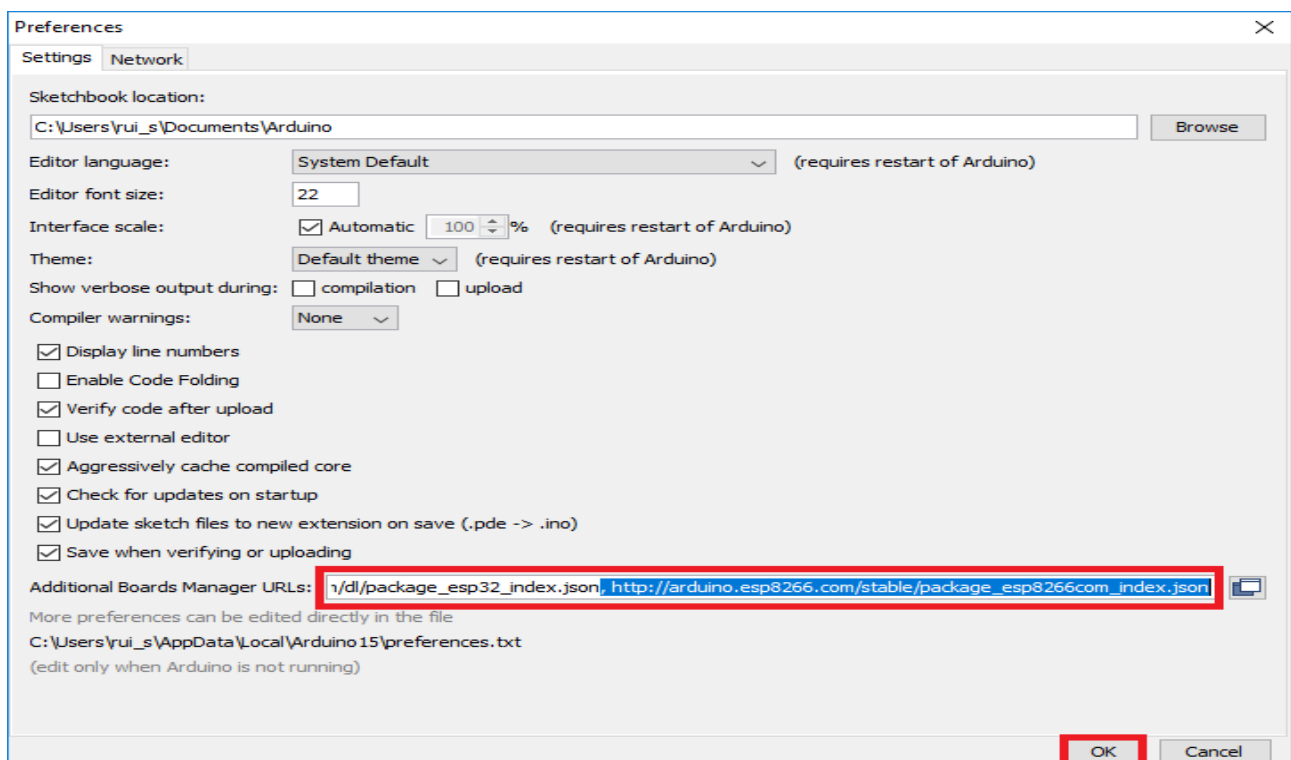
2 Installing Arduino

Arduino IDE provide a safe environment to work with development boards, which includes drivers, libraries, and etc. You can find the arduino installer at <https://www.arduino.cc/en/software>. If you are using Windows 10, just download it from windows store.

3 Setting up the Environment

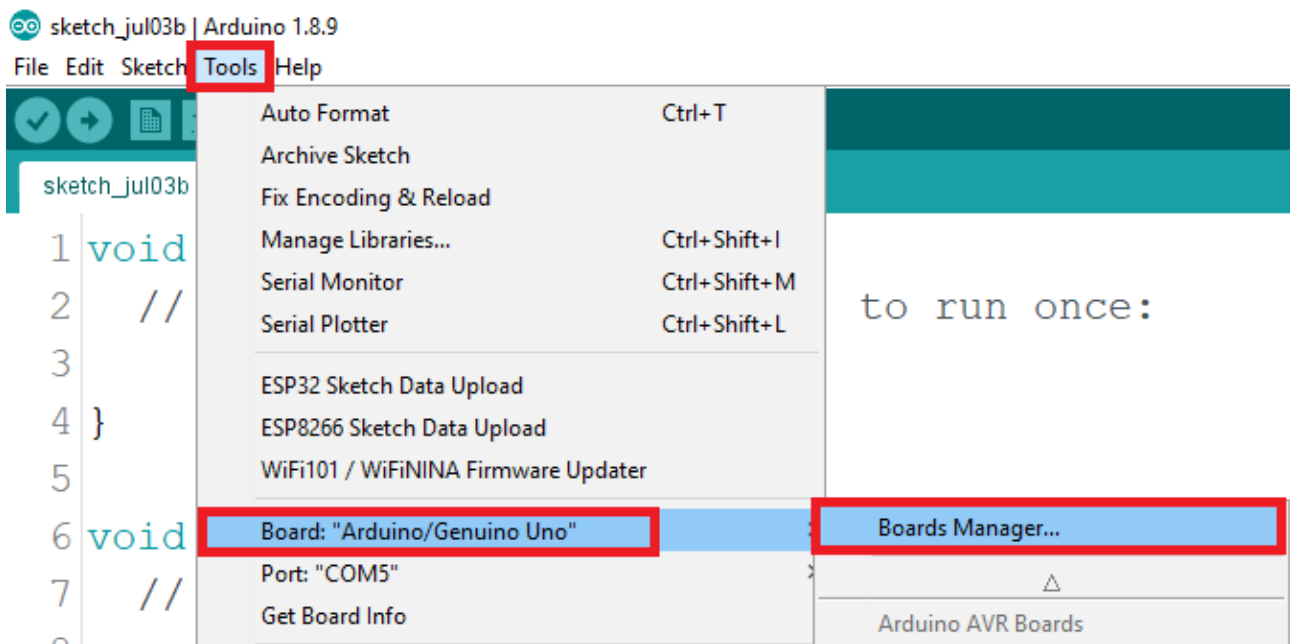
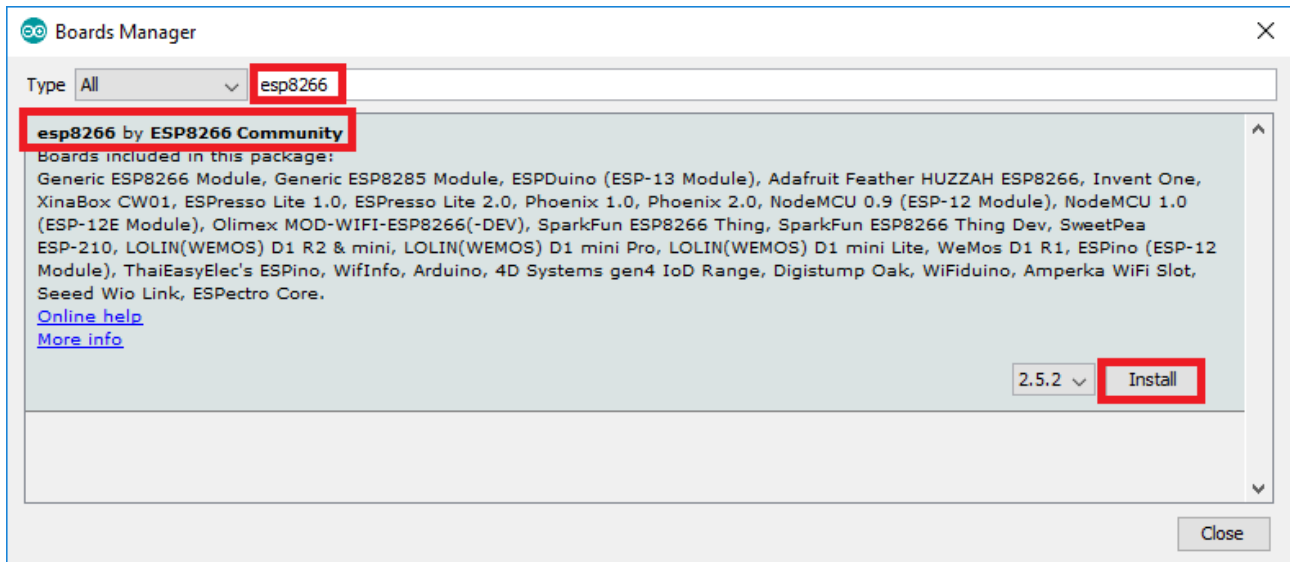
3.1 Add ESP8266 Board URL

In your Arduino IDE, go to File> Preferences and enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into the “Additional Boards Manager URLs” field as shown in the figure below. Then, click the “OK” button:



3.2 Install ESP8266 Drivers

Open the Boards Manager. Go to Tools > Board > Boards Manager.

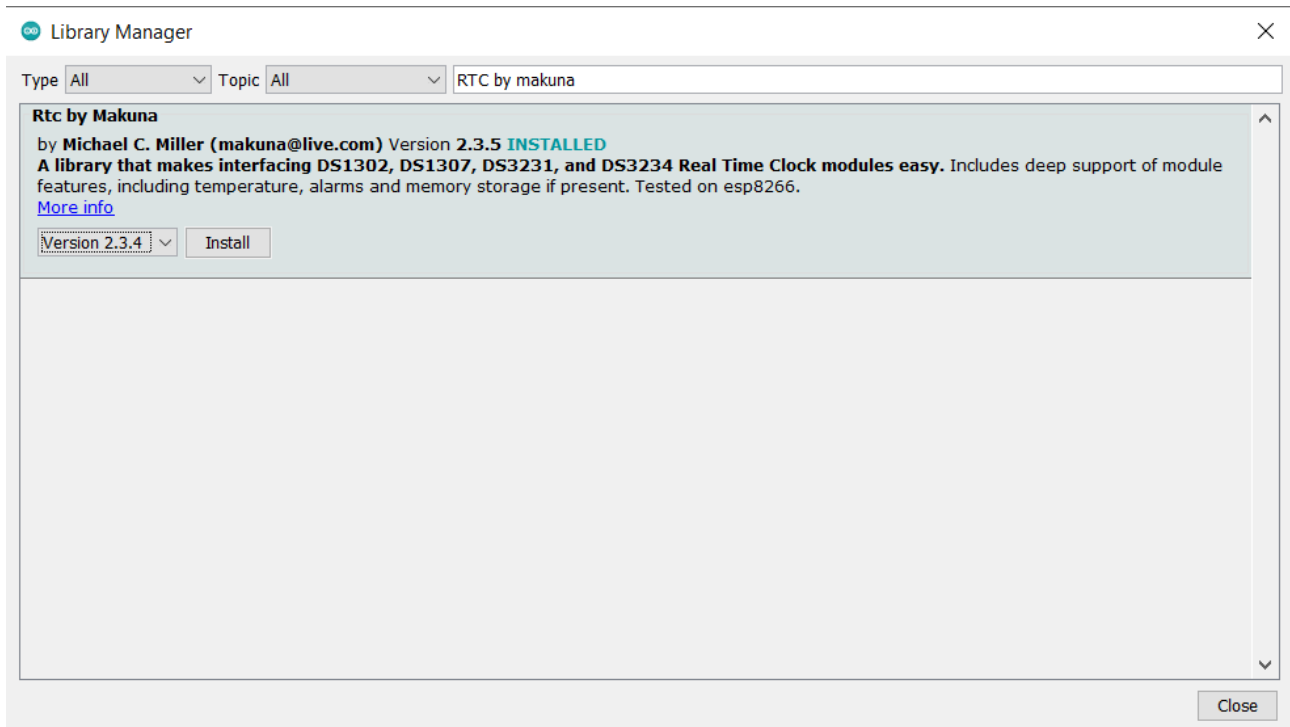


3.3 Install Required Libraries

In order to upload the code we've developed we have to install the list of the following libraries:

- RTC by Makuna
- Arduino Web Sockets by Gill Maimon
- Adafruit HTU21DF by Adafruit

Go to Tools > Manage Libraries. Type the name of the library and install it as the following example:



4 Uploading the Sketch

4.1 Select the Right Devices

Since we're using an ESP-12E NodeMCU Kit, uploading the sketch is very simple. Plug your board to your computer, make sure you have the right board selected, and select the correct USB port:



sketch_jul03b

```
1 void
2   //
3
4 }
5
6 void
7   //
8
9 }
```

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Manage Libraries...	Ctrl+Shift+I
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
WiFi101 / WiFinINA Firmware Updater	
Board: "Arduino/Genuino Uno"	
Port: "COM5"	
Get Board Info	
Programmer: "AVRISP mkII"	
Burn Bootloader	

to run once:

Boards Manager...

▲

ESP8266 Boards (2.5.2)

Generic ESP8266 Module

Generic ESP8285 Module

ESPduino (ESP-13 Module)

Adafruit Feather HUZZAH ESP8266

Invent One

XinaBox CW01

ESPresso Lite 1.0

ESPresso Lite 2.0

Phoenix 1.0

Phoenix 2.0

NodeMCU 0.9 (ESP-12 Module)

NodeMCU 1.0 (ESP-12E Module)

Olimex MOD-WIFI-ESP8266(-DEV)

SparkFun ESP8266 Thing

SparkFun ESP8266 Thing Dev

SweetPea ESP-210

LOLIN(WEMOS) D1 R2 & mini

LOLIN(WEMOS) D1 mini Pro

LOLIN(WEMOS) D1 mini Lite

WeMos D1 R1

ESPino (ESP-12 Module)

ThaiEasyElec's ESPino

WifiInfo

Arduino

4D Systems gen4 IoT Range

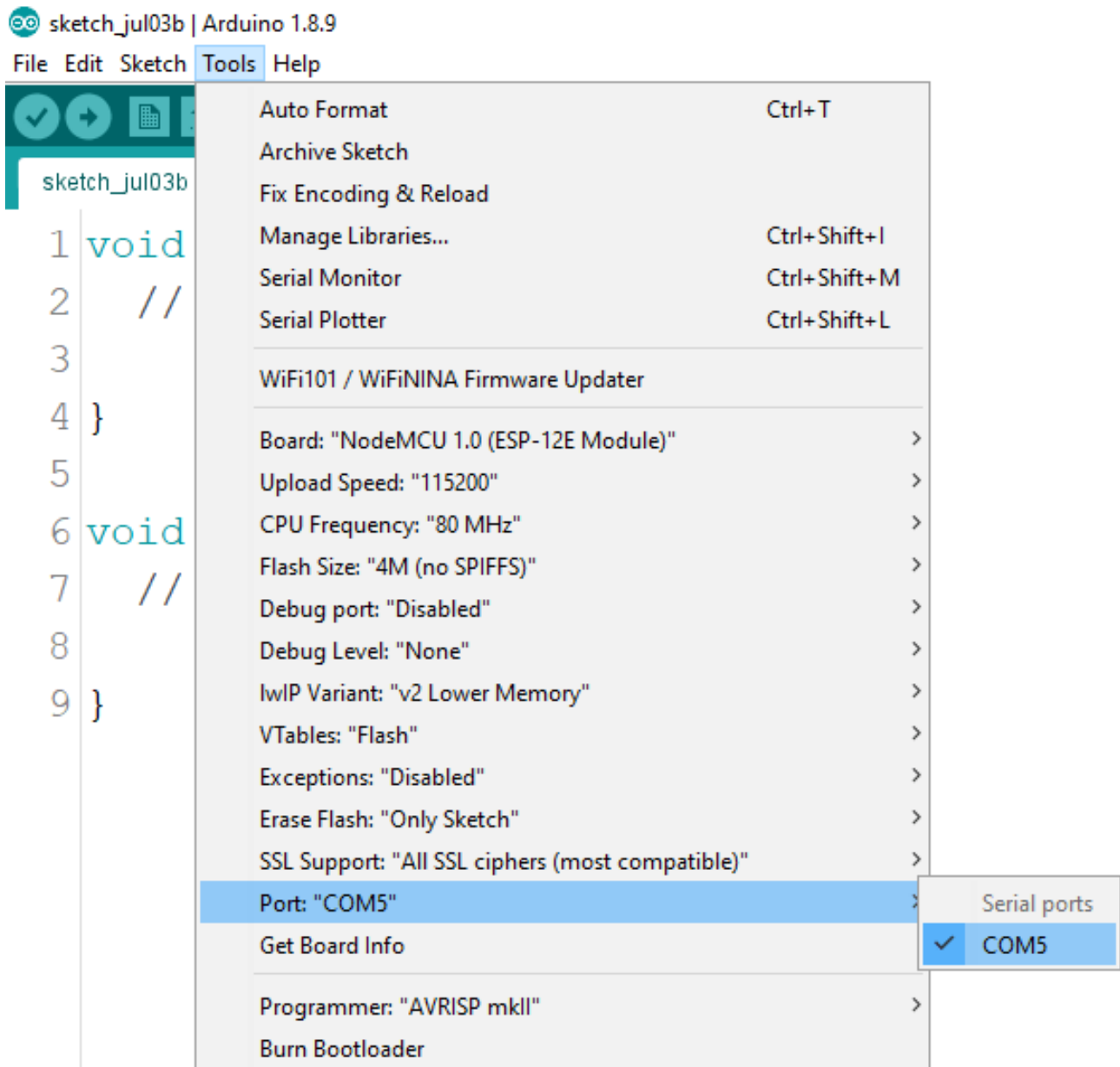
Digistump Oak

WiFiduino

Amperka WiFi Slot

Seed Wio Link

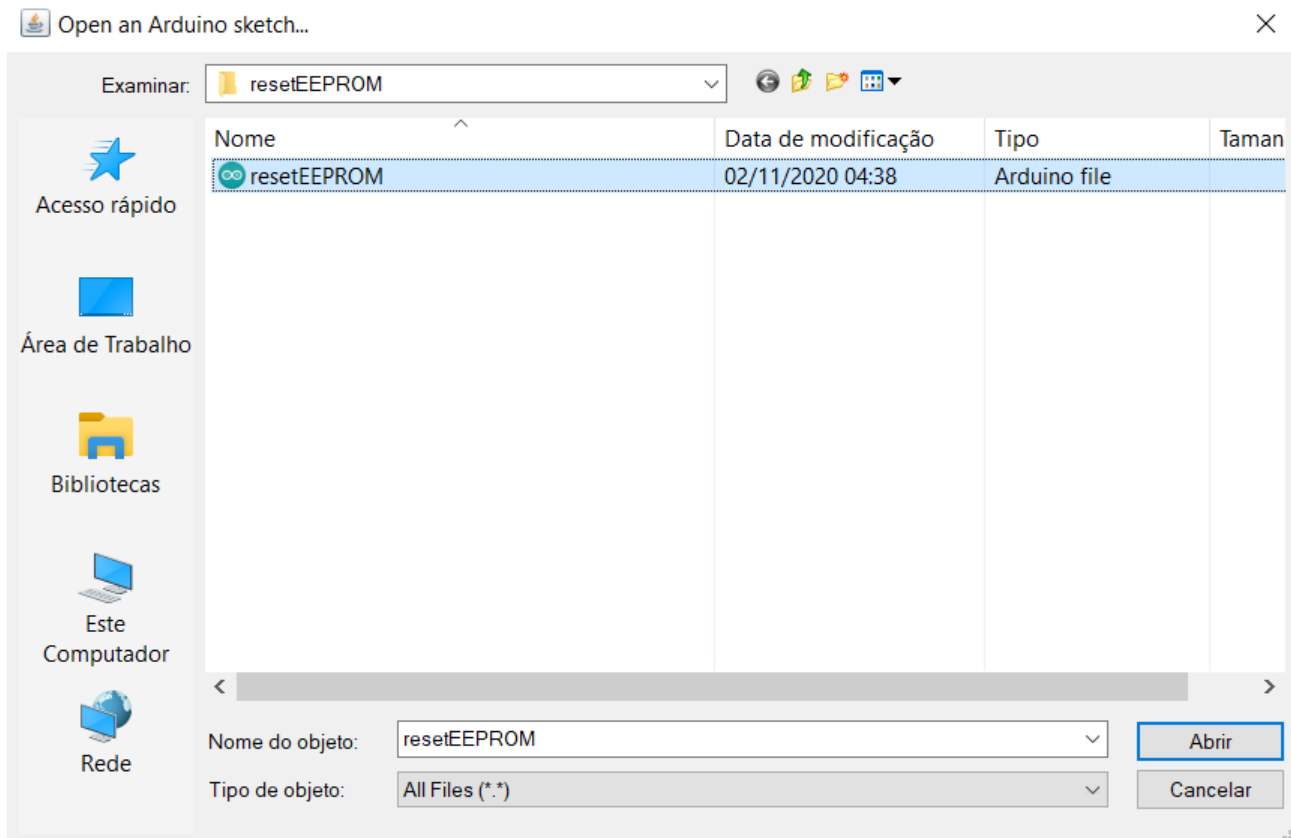
ESpectro Core



4.2 Upload the Code

The first code we must upload is the code to reset the first position of EEPROM. This is a simple code, and we use it just to let the program know if it's the first time the device has been turned on.

So go to File > Open... and open the resetEEPROM file.



Then you can click in the Upload Button to upload the code.

resetEEPROM | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help



Before uploading the last code, fulfill the file macros.hpp with the information about the exact date when the device will be turned on, station location, WiFi credentials, and server parameters. If you don't have a web socket server, don't worry, the station will work as a datalogger as long as you insert an SD card. If you don't have an SD card but you have a web socket server, don't worry too, your station will work just as a web socket client. If you have both, both of the features will work as well.

```
source macros.hpp

#ifndef _MACROS_
#define _MACROS_

/*
Modifie this document in order to give the exactly
location of the station and the date to initiate the measurements.
*/

#define HEADERSIZE 5

//Macros for date to initiate
#define year 2021
#define month 1
#define dayOfMonth 4
#define hour 9
#define minute 0
#define second 0

//Macros for Location
#define ID 6666
#define NAME "TESTES"
#define LAT -22.899874
#define LNG -47.093644
#define ELEVATION 695

//WIFI Credentials
const char* ssid = "xxxxx"; //Enter SSID
const char* password = "xxxxx"; //Enter Password

//Server Parameters
const char* server_ip = "xxxx";
const int port = 0000;

#endif
```

After that you can upload the other code, which implement the features we expect from our device. To do this, just repeat the last step opening the source file.

After uploading the last code, you have 5 seconds to unplug the board in order to not initiate a cycle of measuring. This means that after this time the device will run all routines and execute the first cycle of measurements and tasks.