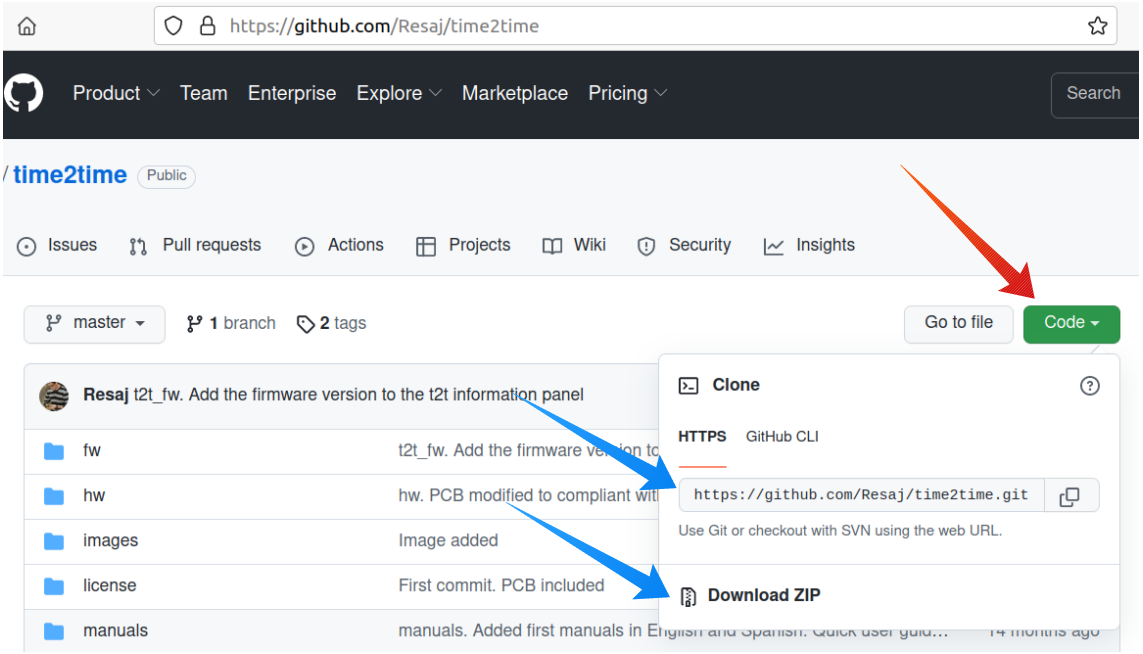
HOW TO UPLOAD THE FIRMWARE

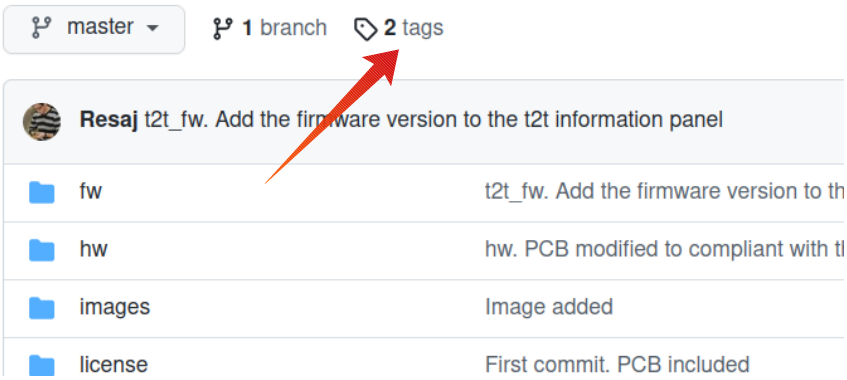
Repository download

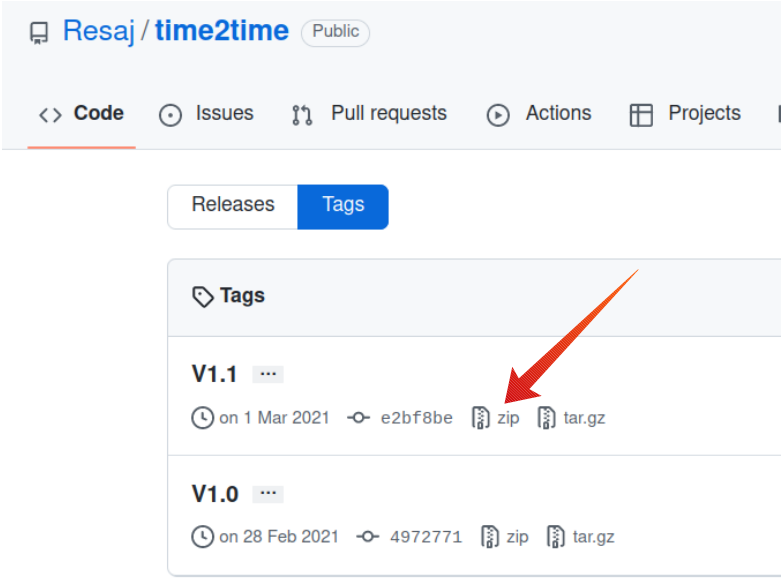
The first step is downloading the firmware of #time2time. There are two options for that:

1. Download the last version of the repository: access to the [Github website of #time2time] and download the whole repository clicking on the "Code" button. The repository can be cloned as a local one in your PC with Git, or can be downloaded as a whole ZIP:



2. Or you can download as a ZIP file any of the last versions listed in the tags section:





Once the repository has been downloaded, unzip the file in the case that you have downloaded it as a ZIP file.

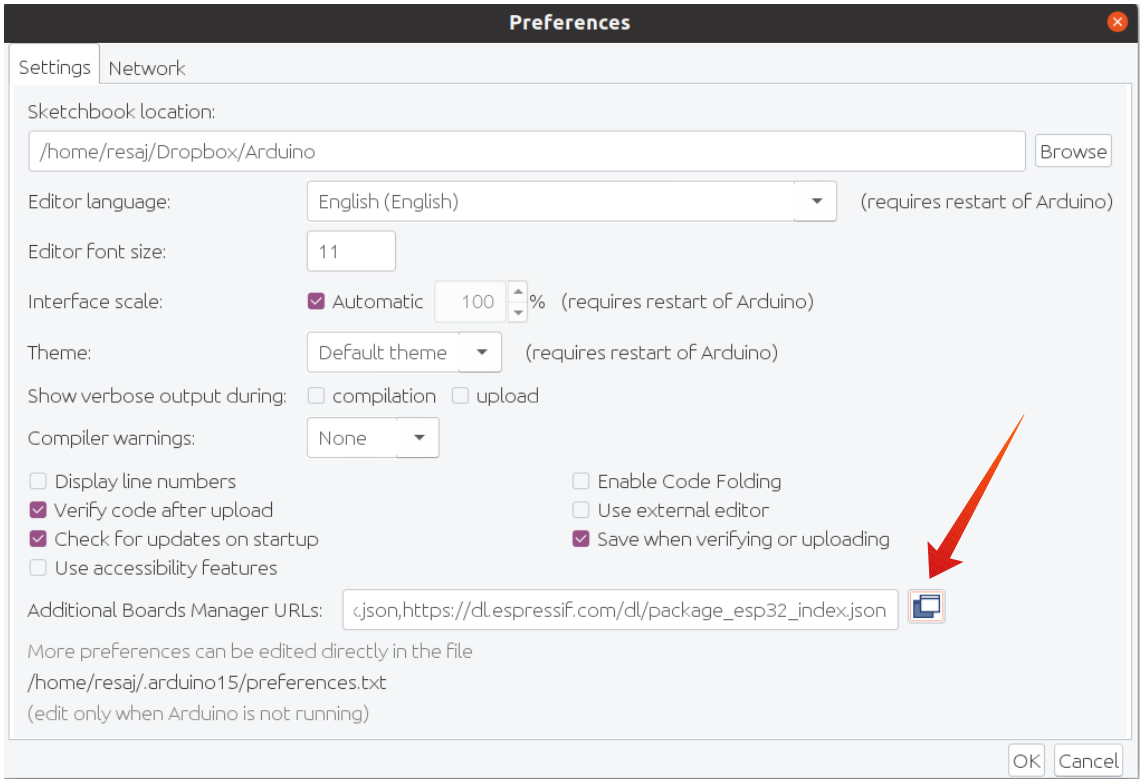
Arduino IDE installation

Download and install the Arduino IDE for your operating system:

ESP32 support installation

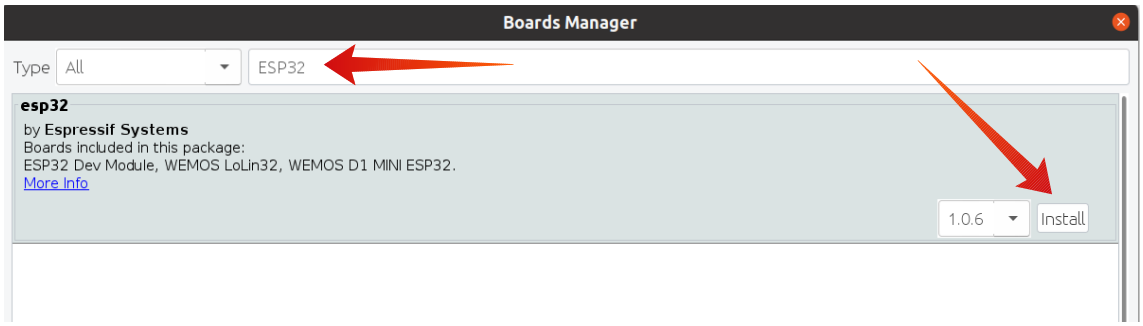
Open the Arduino IDE and enter the menu File/Preferences. Add the next line on the field “Additional Boards Manager URLs”:

https://dl.espressif.com/dl/package_esp32_index.json



Click on “Ok” to save the changes and exit the preferences menu.

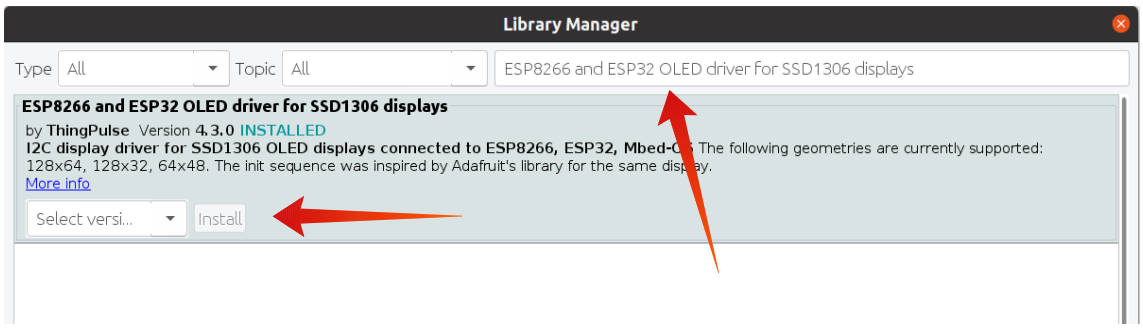
Then, it’s necessary to install the ESP32 libraries. For this, enter the menu Tools/Board/Boards Manager. Write “ESP32” on the filter and install the option that appears:



Now you’ll see that the option “ESP32 Arduino” appears inside the menu Tools/Board. Unfold its submenu and select the option “ESP32 Dev module”.

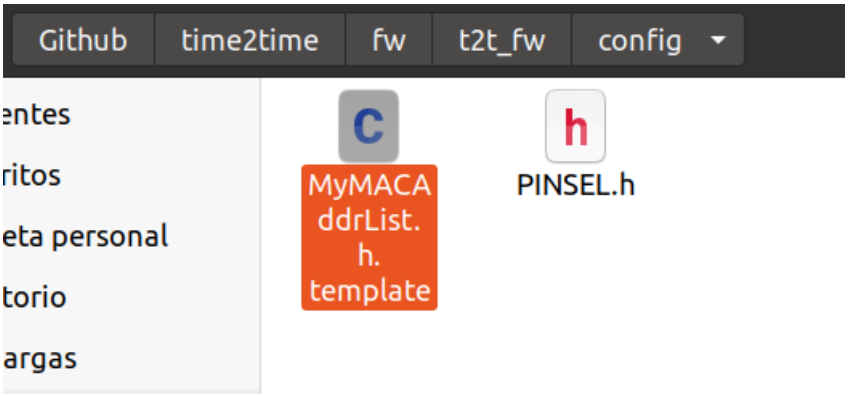
Program libraries installation

Access the menu Sketch/Include Library/Manage Libraries. Search and install the library “ESP8266 and ESP32 OLED driver for SSD1306 displays”:



MAC addresses file configuration

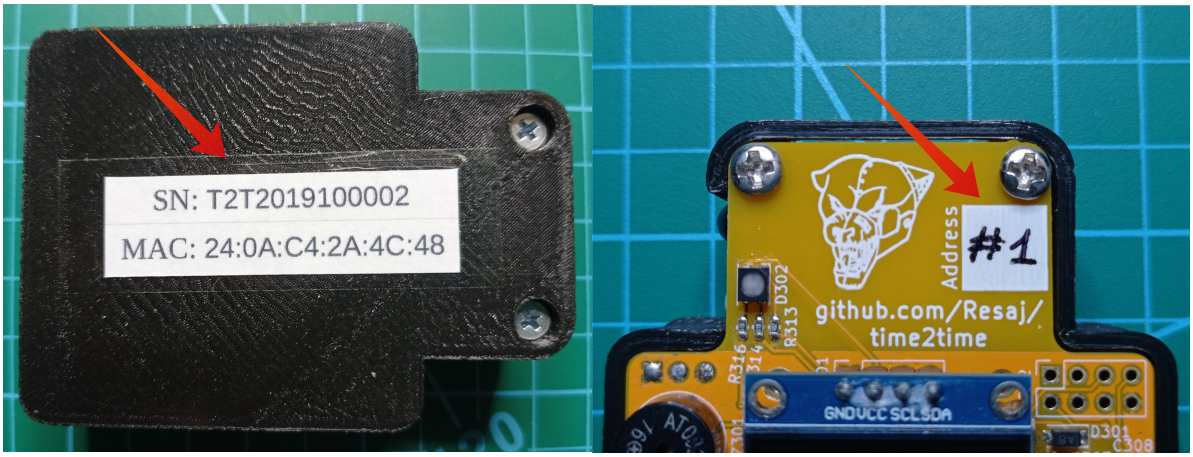
To link several #time2time between them, you’ll have to include their MAC addresses into a list inside the program. For this, go to the project folder that you downloaded in the first step and rename the file fw/t2t_fw/config/MyMACAddrList.h.template. Its new name will be the same but removing the “.template” extension. Thus, the file will become a header file with “.h” extension.



Open the file that you’ve just renamed and complete the next list with the MAC addresses of your #time2time nodes:

```
uint8_t MyMACAddrList[][6] = {
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* MAC of the node with address #0 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #1 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #2 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #3 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #4 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #5 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},  /* #6 */
  {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}   /* #7 */
};
```

You can find the node address and the MAC address in the front and back parts of your #time2time, respectively:



Make sure you associate each MAC address with the same line number than the node address. For example, in my case I have four #time2time, and the one that has the node address #1 owns the MAC 24:0A:C4:2A:4C:48. You'll see this data match with the second line of my table, since it starts to count by zero:

```
uint8_t MyMACAddrList[][6] = {
  {0x24, 0x0A, 0xC4, 0x2B, 0x44, 0x2C},  /* MAC of the node with address #0 */
  {0x24, 0x0A, 0xC4, 0x2A, 0x4C, 0x48},  /* #1 */
  {0x24, 0x0A, 0xC4, 0x2B, 0x44, 0xCC},  /* #2 */
  {0x8C, 0xAA, 0xB5, 0xBF, 0x89, 0x74}   /* #3 */
};
```

You can remove the rest of the lines in the table if you don't use them, like I did with mine. Save the file and close it.

This file is the same for all your #time2time. Thus, they can always connect between them if they can find themselves in the table.

Firmware upload

You are ready to upload the firmware. Now you only have to connect your nodes #time2time one by one to a USB port, pick the upload port in the menu Tools/Port and upload the firmware by clicking on the option "Upload":