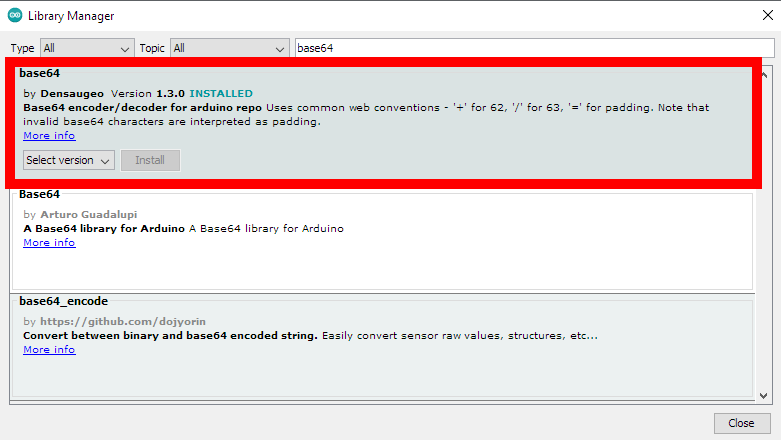
AlgoIoT: instructions

v20231017-1

# Prerequisites

* ESP32 development board. Any [ESP32-DevKitC](https://www.espressif.com/en/products/devkits/esp32-devkitc) - compatible board will do; we used a UM TinyPICO <https://www.tinypico.com/>
* USB-Serial drivers if required by your development board / Operating System
* Working WiFi 2.4 GHz wireless network
* Arduino IDE v1.8.19 or latest v2.x, from <https://www.arduino.cc/en/software> (we used IDE v1.8.19, Windows version). A basic knowledge of the Arduino environment and IDE is required
* Arduino-ESP32 2.0.x core package (follow instructions at <https://github.com/espressif/arduino-esp32>). We used v2.0.14
* ArduinoJSON library (follow instructions at <https://arduinojson.org/v6/doc/installation/>)
* Arduino Crypto library (follow instructions at <https://www.arduino.cc/reference/en/libraries/crypto/>)
* Base64 library by Densaugeo: from Arduino IDE, select Tools->Manage Libraries, search for “base64”, select “base64 by Densaugeo”, click Install:



# Instructions

Download project from GitHub, either cloning the git repository

git clone <https://github.com/GT50-SRL/algoiot.git>

or downloading and uncompressing the ZIP archive from

<https://github.com/GT50-SRL/algoiot> (**Code**->**Download ZIP**, then unzip “algoiot-main.zip”)

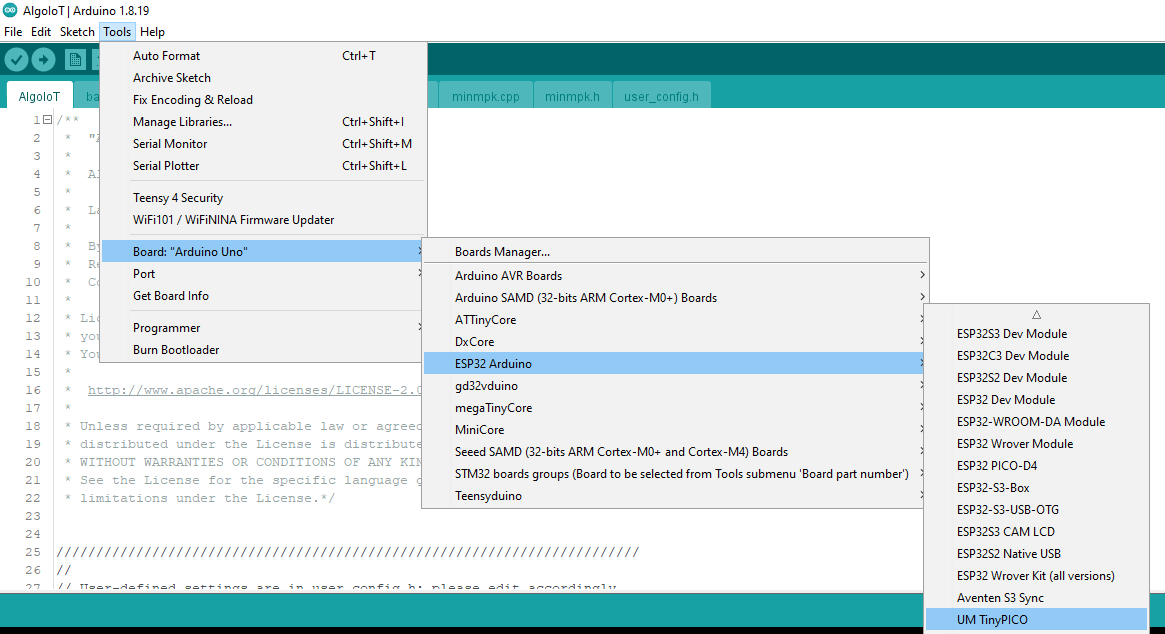
Open “**AlgoIoT.ino**” from the Arduino IDE

Edit “**user\_config.h**” with your WiFi SSID and password

Check that “**SERIAL\_DEBUGMODE**” is **not** commented (otherwise you would have no output on the Serial Monitor)

Select your development board from **Tools->Board->ESP32 Arduino->[your board]**

Since we used “UM TinyPICO”, this is our selection (as an example):

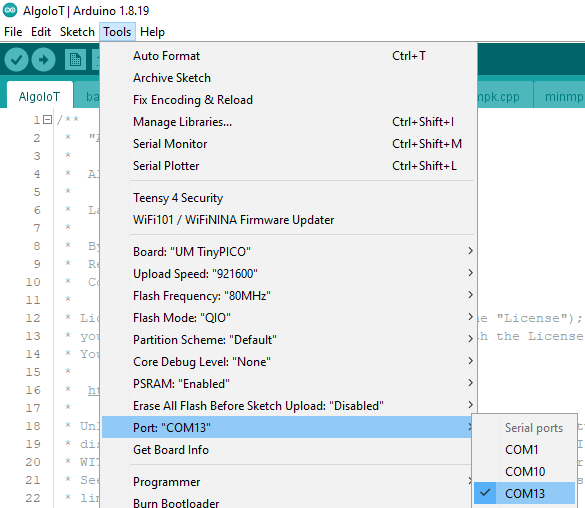


Now you will have to set your board options, under “**Tools**”.

In particular, the “**Partition Scheme**” should allow **at least 1 MB for the App**. This is usually the default option.

You should connect your board now, if not already connected. Please set **Tools->Port** to the COM port corresponding to your ESP32 board.

In our case, it is COM13:



You are now ready to build and upload the code (Right Arrow button on Arduino IDE toolbar). Please note that building and uploading takes some time (a few minutes).

Instructions for uploading vary among ESP32 development boards: you may have to press a Boot/En/Reset button the first time you upload a sketch; please refer to your board’s instructions.

Please avoid opening Serial Monitor the first time you upload a sketch: this may affect the ability for the uploader to properly drive the COM port. Wait for upload to finish, then open Serial Monitor to check board messages and confirm everything works.



For **troubleshooting** the build/upload phase, you may want to enable Verbose mode in Arduino IDE:

**File->Preferences->Show verbose output during: compilation / upload**

If everything works, you should a similar output on the IDE Serial Monitor:

