# ESP32_IOT_MODBUS_GATEWAY DOCUMENTATION

**STEP 1 -** Download the latest version of Arduino IDE (during documentation we use Arduino 1.8.19).
If you want to use internal flash download Arduino 1. xxx version (the ESP32 Filesystem Uploader plugin is not supported on Arduino 2.0. ) to download Arduino 1.8.19 use this link https://www.arduino.cc/en/software

**STEP 2 -** In your Arduino IDE, go to File > Preferences.

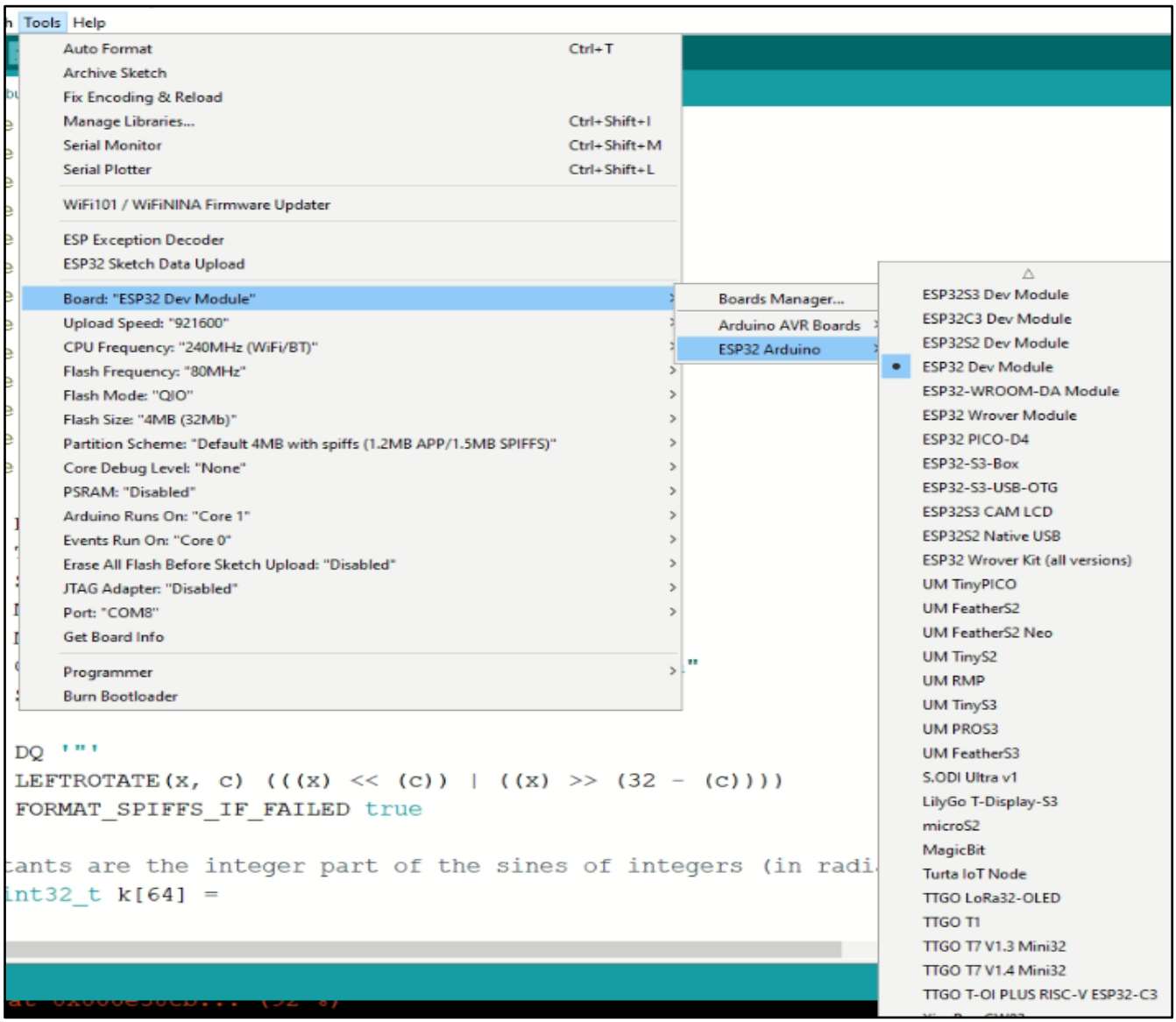
**STEP 3 -** Copy and paste the following line to the Additional Boards Manager URLs field.
(https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

**STEP 4 -** Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma, as follows:
http://arduino.esp8266.com/stable/package_esp8266com_index.json, https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

**STEP 5 -** Open the Boards Manager. You can go to Tools > Board > Boards Manager… or you can simply click the Boards Manager icon in the left-side corner.

**STEP 6 -** Search for ESP32 and press the install button for esp32 by Espressif Systems (during this documentation we use esp 2.0.8).

**STEP 7 -** Go to "tools->board->ESP32Arduino" select "ESP DEV MODULE"



**STEP 8 –** Set the correct port in “tools->port”.

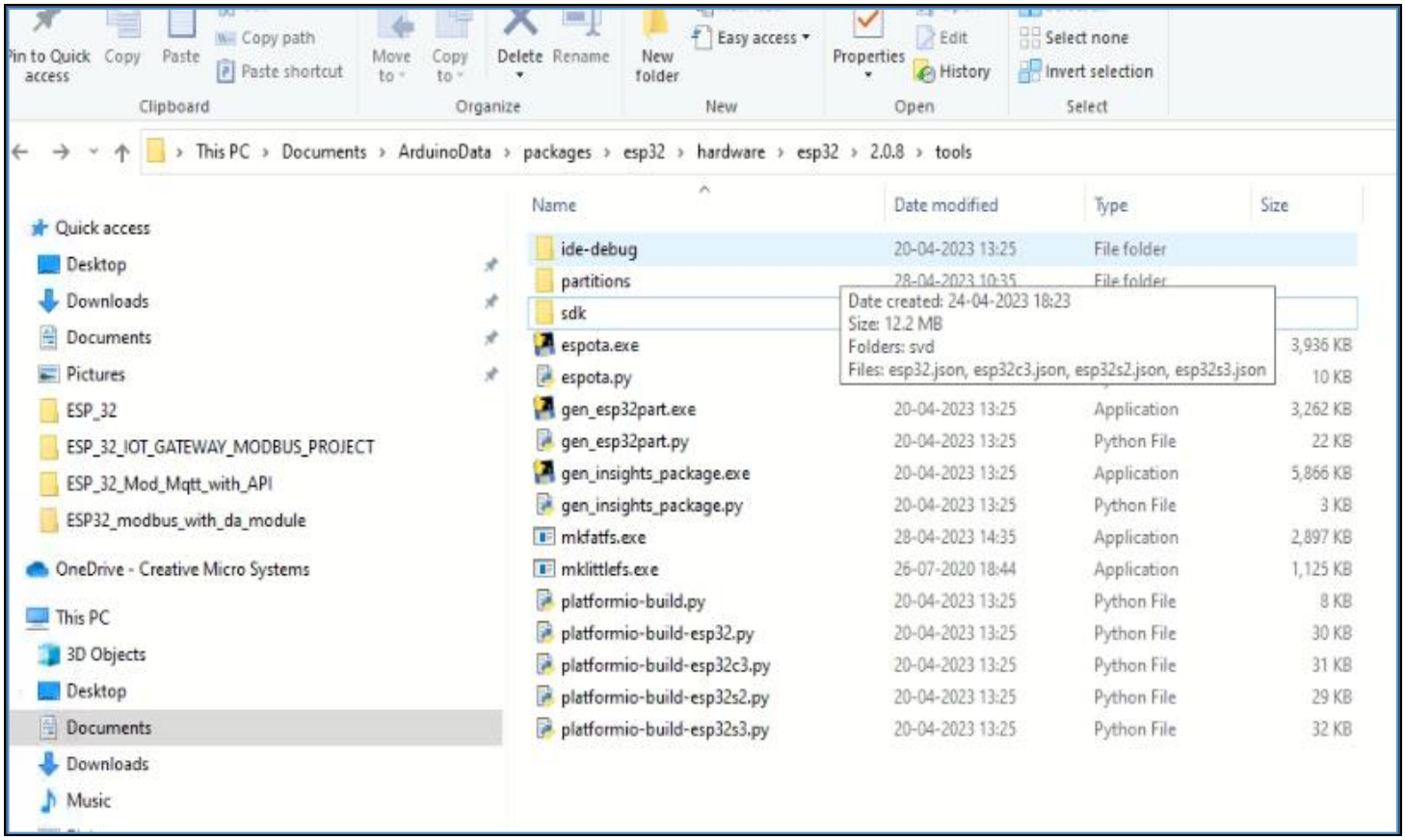
**STEP 9 -** To test the ESP32 add-on installation, we’ll upload a simple code that blinks the onboard LED (GPIO 2). Copy the following code to your Arduino IDE:

```
//**********************************
#include <Arduino.h>
#define LED 2
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED, HIGH);
  Serial.println("LED is on");
  delay(1000);j98
  digitalWrite(LED, LOW);
  Serial.println("LED is off");
  delay(1000);
}
//**********************************
```

**STEP 10 -** After running this program check whether the led in the esp32 is blinking for every 1 second. If it’s working fine then the Esp32 is mounted successfully to Arduino ide

**STEP 11 -** Open “file -> preference” in Arduino ide, there you can find the path under “more preference can be edited directly in In the file” where the ESP32 related files stored. There go to “Arduino data\packages\esp32\hardware\esp32\2.0.8\tools”

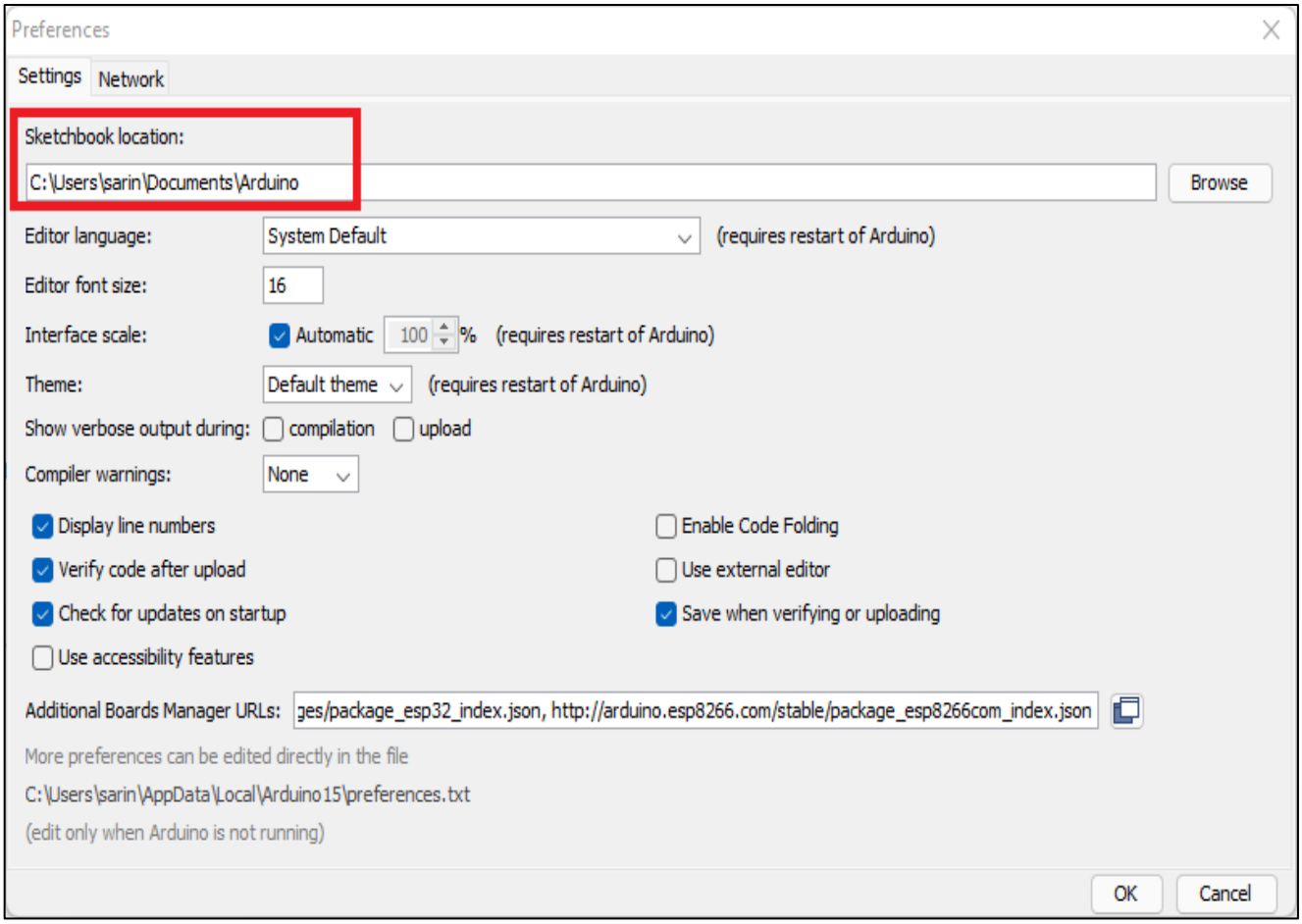
**STEP 12 –** check whether the “mkfatfs.exe” is there if it’s not there download and and paste it there
“mkfatfs.exe” – download link https://github.com/labplus-cn/mkfatfs/releases/tag/v1.0
**NOTE: While downloading, you will get “mkfatfs_v1.0.exe” RENAME that file to “mkfatfs.exe”.**



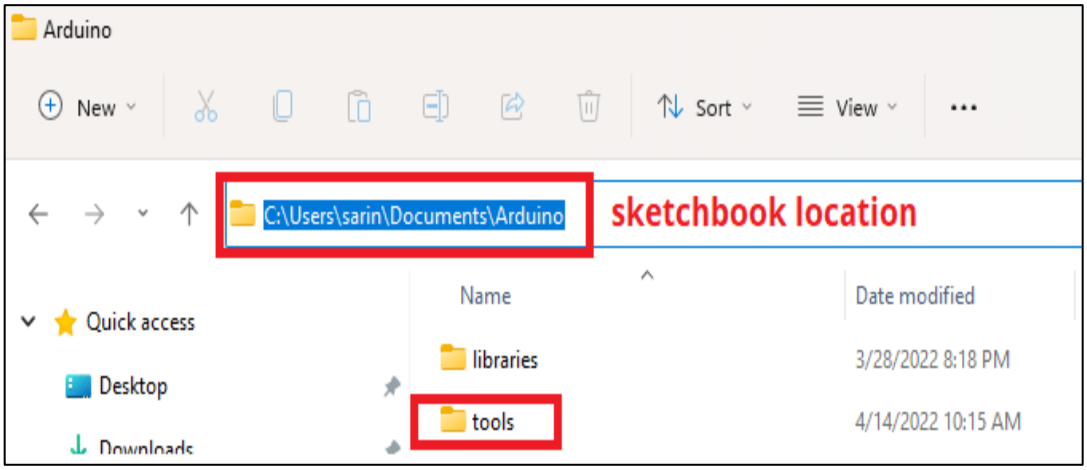
**STEP 12 –** There will be three file systems in esp32 we tested SPIFFS and FATFS.

**STEP 13 –** Download the "esp32fs.zip" file from the below link and extract and take the "esp32fs.jar" file from the zip file.
https://github.com/lorol/arduino-esp32fs-plugin/releases

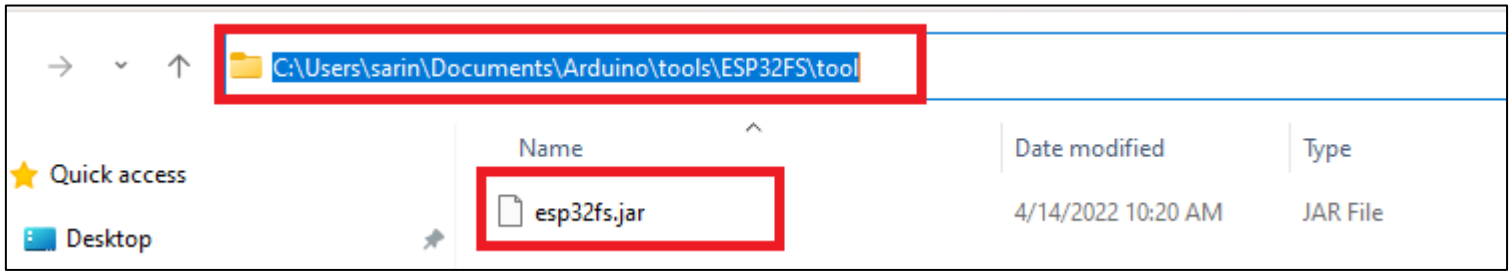
**STEP14 -** In your Arduino IDE, go to **File** > **Preferences** and check your Sketchbook location. In my case, it's in the following path "C:\Users\User1\Documents\Arduino".



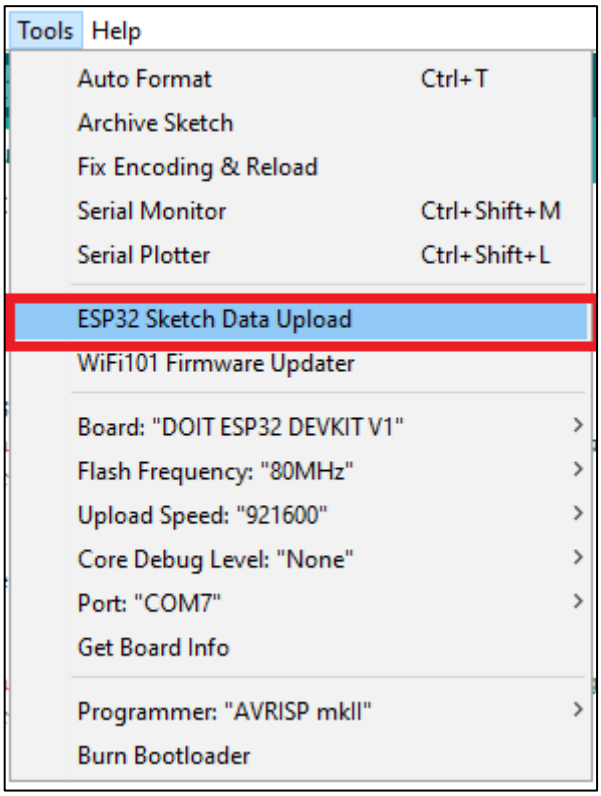

**STEP 15 -** Go to the sketchbook location, and create a **tools** folder.



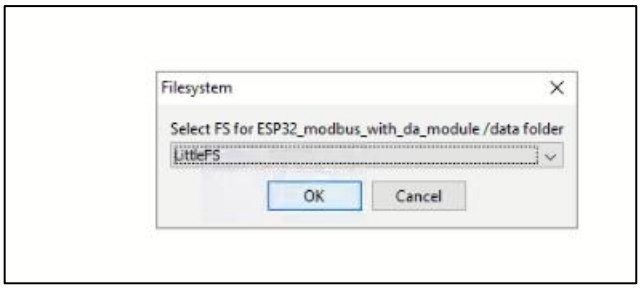
**STEP16 -** Unzip the downloaded *.zip* folder. Open it and copy the ***ESP32FS*** folder to the **tools** folder you created in the previous step. You should have a similar folder structure. <Sketchbook-location>/**tools/ESP32FS/tool/esp32fs.jar**




**STEP 17 -** Finally, restart your Arduino IDE. To check if the plugin was successfully installed, open your Arduino IDE. Select your ESP32 board, go to **Tools** and check that you have the option "**ESP32 Sketch Data Upload**".



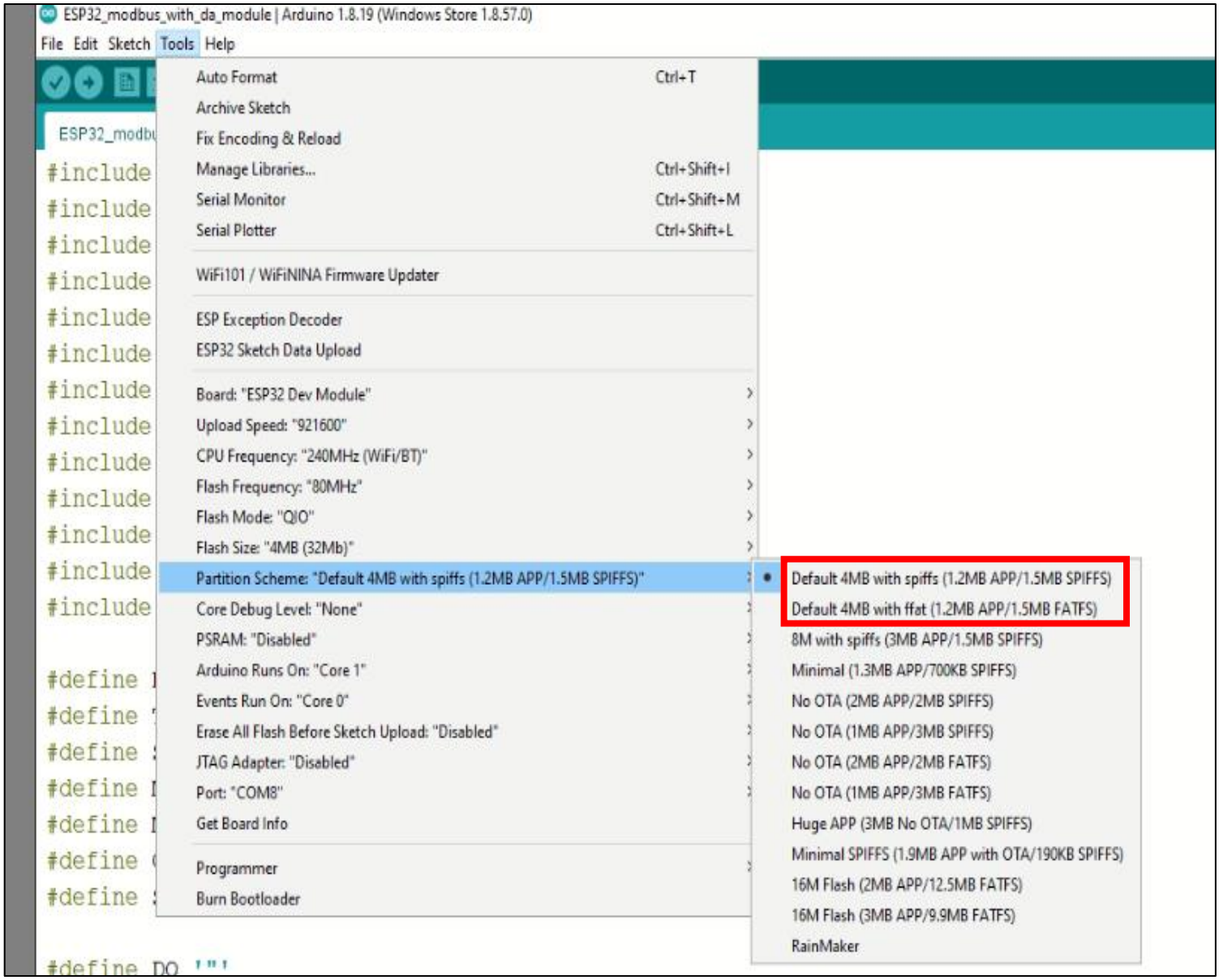

**STEP18 –** after clicking the "**ESP32 Sketch Data Upload**" it will show like the below image.




**STEP 19 –** select in which file system you going to flash the files.

**STEP 20 –** go to “tools->partition scheme” and select which file system you going to flash

**NOTE – default we use first two file systems in the partition scheme list as mentioned below.**




**NOTE : IF YOU ARE USING MQTT PROTOCOL MAKE SURE THAT THE COMMON NAME (CN) OF THE CERTIFICATE YOU ARE USING TO CONNECT WITH MQTT BROKER IS SAME AS THE BROKERS NAME**
**EG : common name( CN ) = cmscloud.in the mqtt broker name should be cmscloud.in**