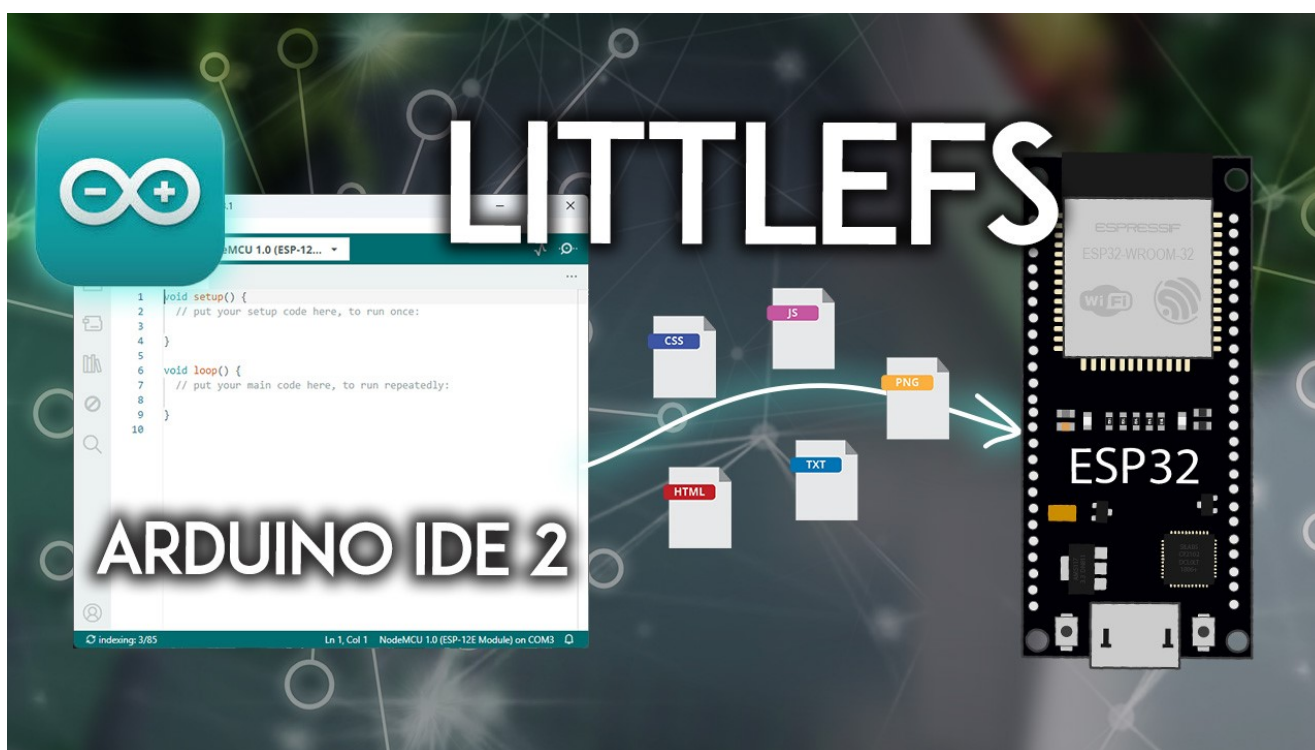


Arduino IDE 2: Install ESP32 LittleFS Uploader (Upload Files to the Filesystem)

In this guide, you'll learn how to upload files to the ESP32 LittleFS Filesystem using Arduino IDE 2 (2.2.1 or a higher version). We'll install an uploader plugin that will add a new menu to the Arduino IDE. This plugin is also compatible with the ESP8266 and Raspberry Pi Pico boards.



Using an ESP8266? Follow this tutorial instead: [Arduino IDE 2: Install ESP8266 NodeMCU LittleFS Uploader \(Upload Files to the Filesystem\)](#).

Table of Contents

- [Introducing LittleFS](#)
- [Installing LittleFS Filesystem Uploader Plugin](#)
 - [Windows Instructions](#)

- [Mac OS X Instructions](#)
- [Uploading Files to ESP32 using the Filesystem Uploader](#)
- [Testing the ESP32 LittleFS Uploader](#)

If you're still using **Arduino 1.8**, you can follow this tutorial instead: [ESP32: Upload Files to LittleFS using Arduino IDE \(legacy\)](#).

Introducing LittleFS

LittleFS is a lightweight filesystem created for microcontrollers that lets you access the flash memory as you do in a standard file system on your computer, but it's simpler and more limited. You can read, write, close, and delete files. Using LittleFS with the ESP32 boards is useful to:

- Create configuration files with settings;
- Save data permanently;
- Create files to save small amounts of data instead of using a microSD card;
- Save HTML, CSS, and JavaScript files to build a web server;
- [Save images, figures, and icons](#);
- And much more.

Installing the LittleFS Uploader Plugin on Arduino IDE 2

To upload files to the ESP32 on [Arduino IDE 2](#), we'll use [this LittleFS Uploader plugin](#) that is compatible with Arduino 2.2.1 or higher and can be used with the ESP32, [ESP8266](#), and [Raspberry Pi Pico boards](#).

Windows Instructions

Follow the next steps to install the filesystem uploader if you're using Windows ([click here for MacOS instructions](#)):

1) Go to the [releases page](#) and click the `.vsix` file to download.

Release 1.1.5 - ESP32, ESP8266, and Pico Support Latest

New in this release: ESP32 family board name fixes

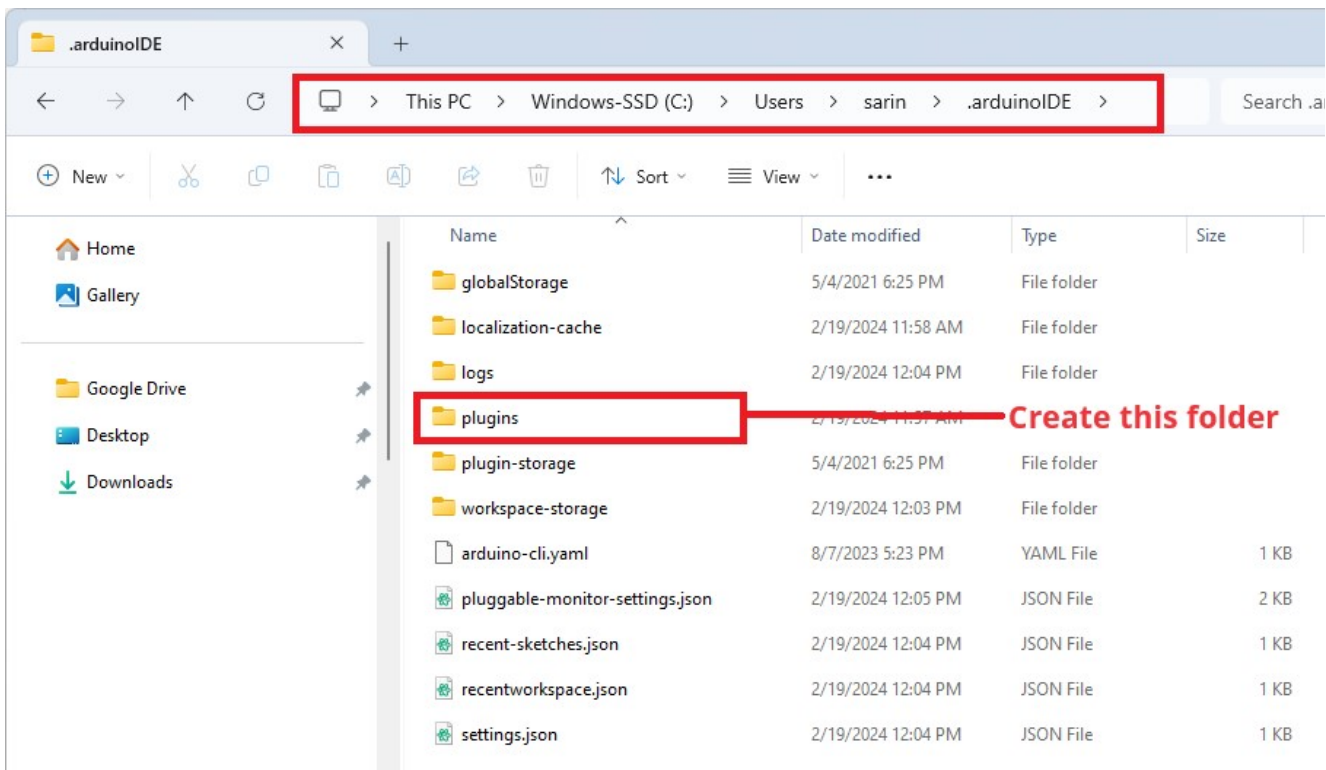
▼ Assets 3

 arduino-littlefs-upload-1.1.5.vsix	1.03 MB	last week
 Source code (zip)		last week
 Source code (tar.gz)		last week

 4 4 people reacted

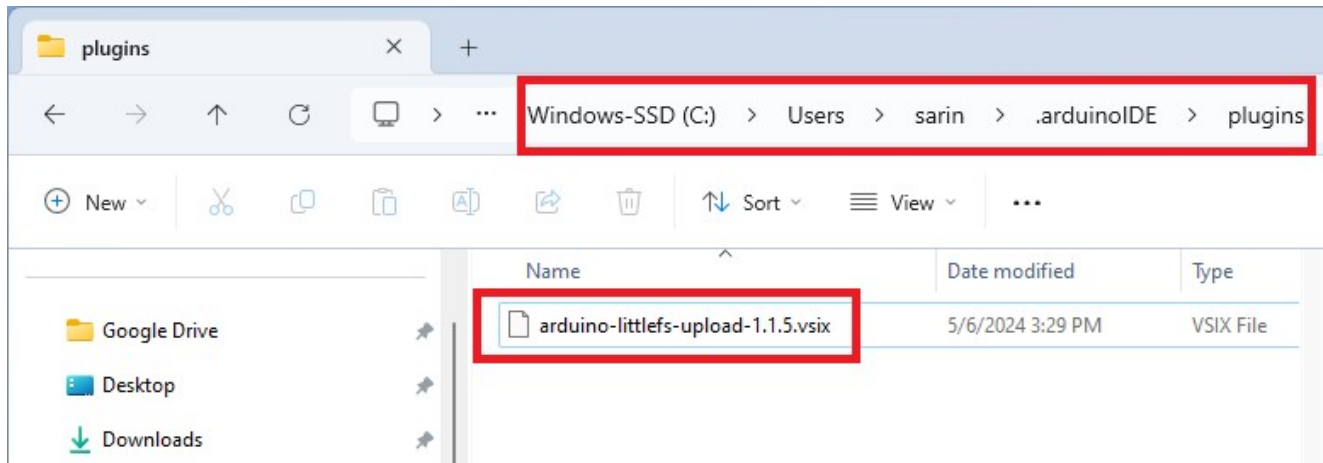
2) On your computer, go to the following path:

C:\Users\<username>\.arduinoIDE\ . Create a new folder called *plugins* if you haven't already.

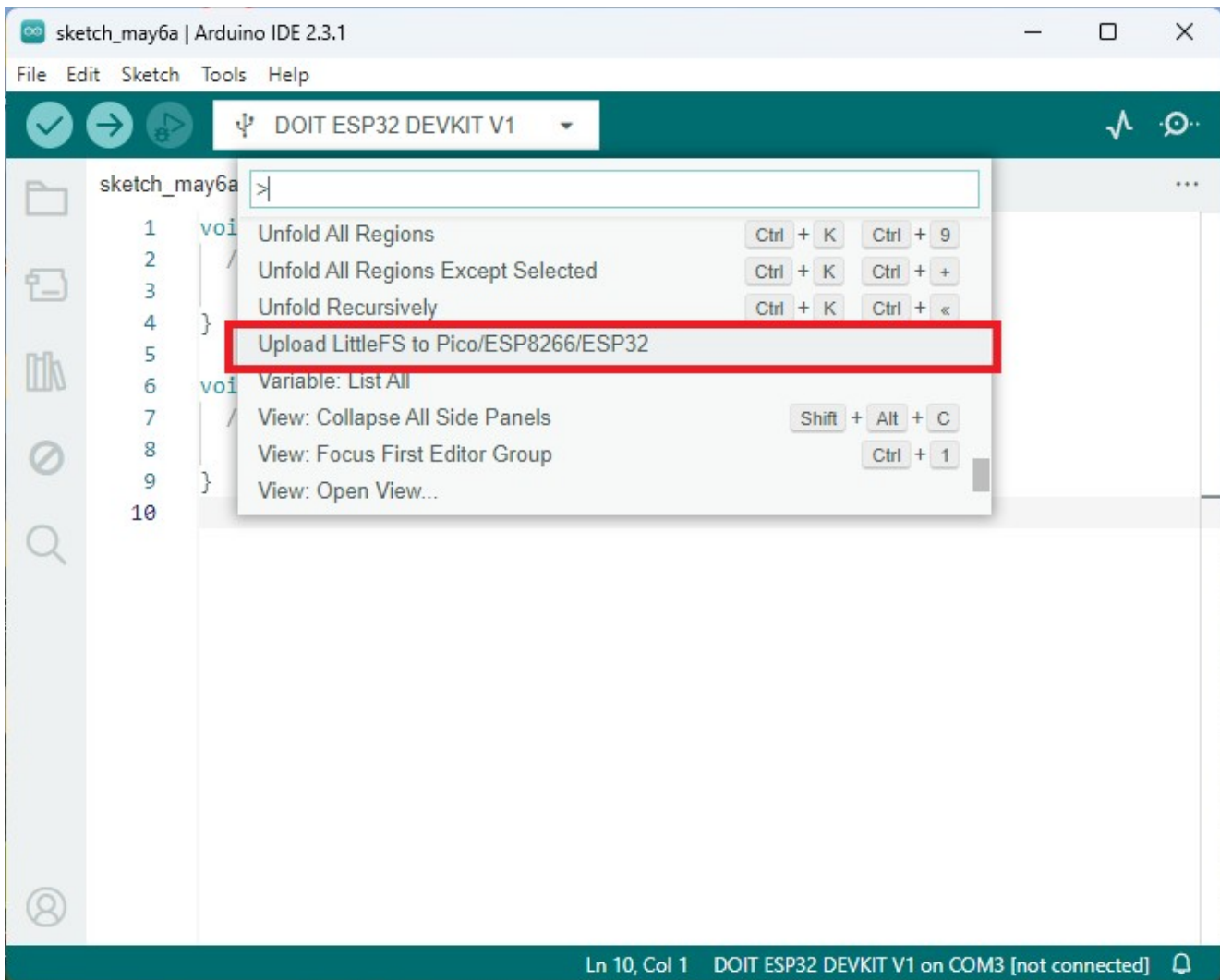


3) Move the .vsix file you downloaded previously to the *plugins* folder (remove any

other previous versions of the same plugin if that's the case).



4) Restart or open the Arduino IDE 2. To check if the plugin was successfully installed, press **[Ctrl] + [Shift] + [P]** to open the command palette. An instruction called **'Upload Little FS to Pico/ESP8266/ESP32'** should be there (just scroll down or search for the name of the instruction).



That means the plugin was installed successfully. Proceed to [this section](#) to test the filesystem uploader plugin.

Mac OS X Instructions

Follow the next steps to install the filesystem uploader if you're using Mac OS X:

- 1) Go to the [releases page](#) and click the `.vsix` file to download.

Release 1.1.5 - ESP32, ESP8266, and Pico Support Latest

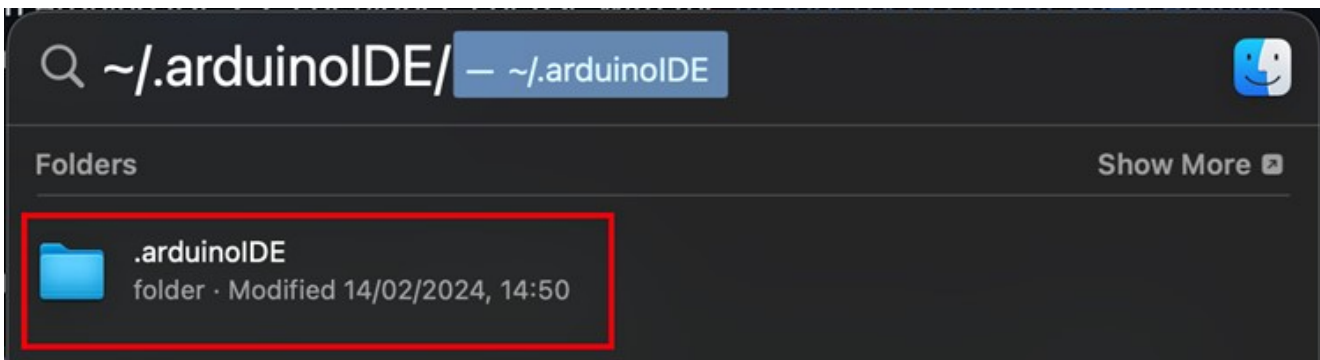
New in this release: ESP32 family board name fixes

▼ Assets 3

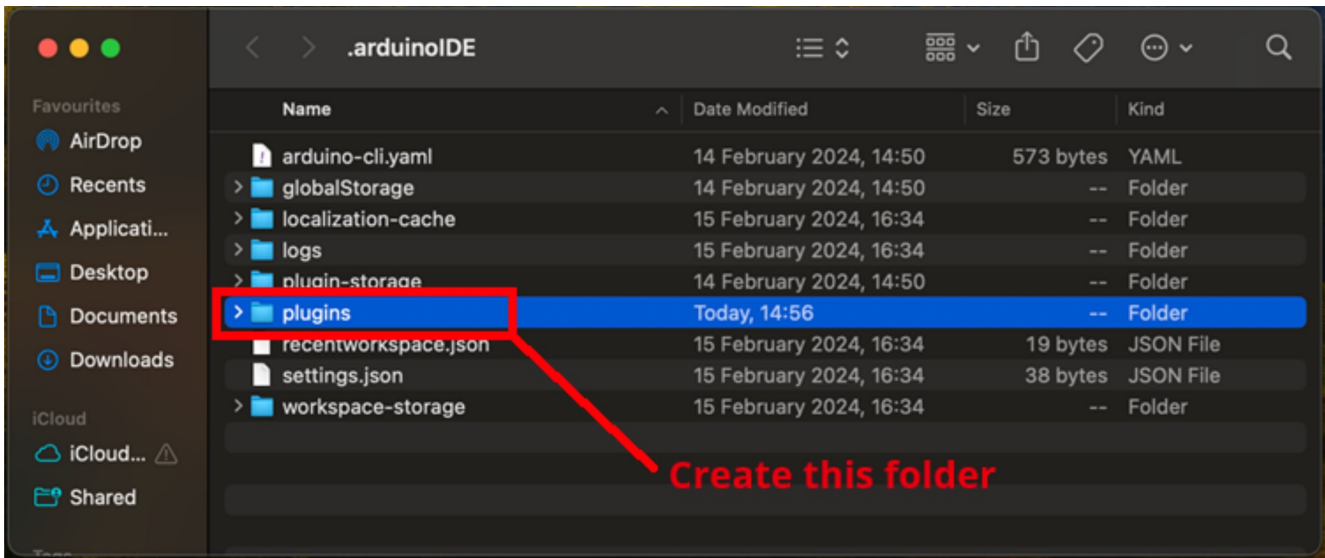
 arduino-littlefs-upload-1.1.5.vsix	1.03 MB	last week
 Source code (zip)		last week
 Source code (tar.gz)		last week

 4 4 people reacted

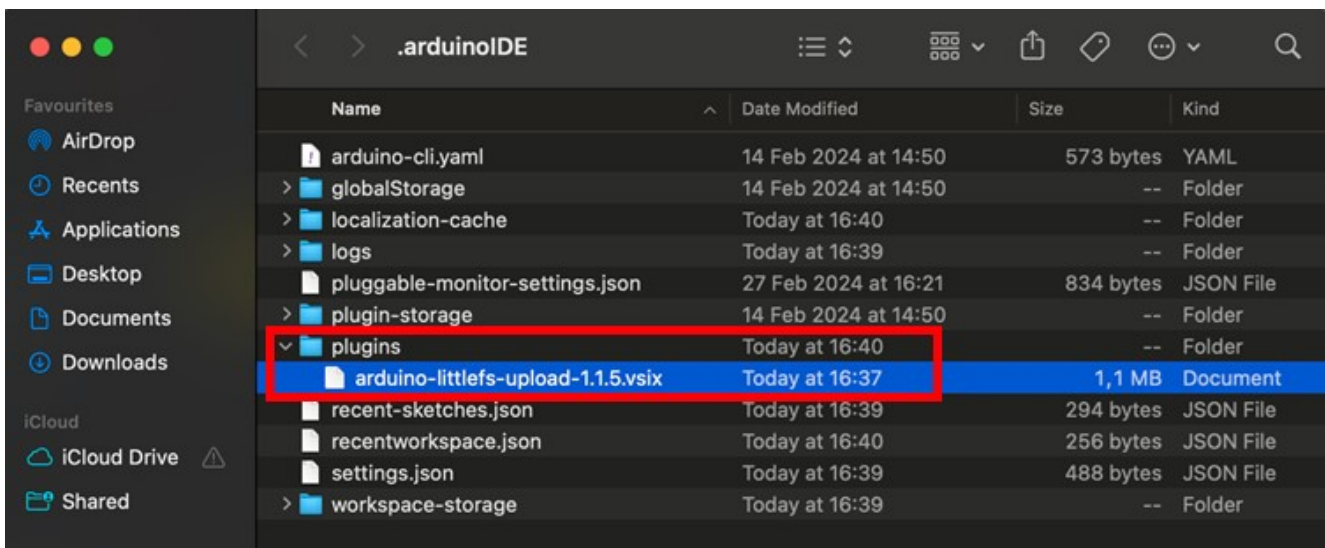
2) In Finder, type `~/ .arduinoIDE/` and open that directory.



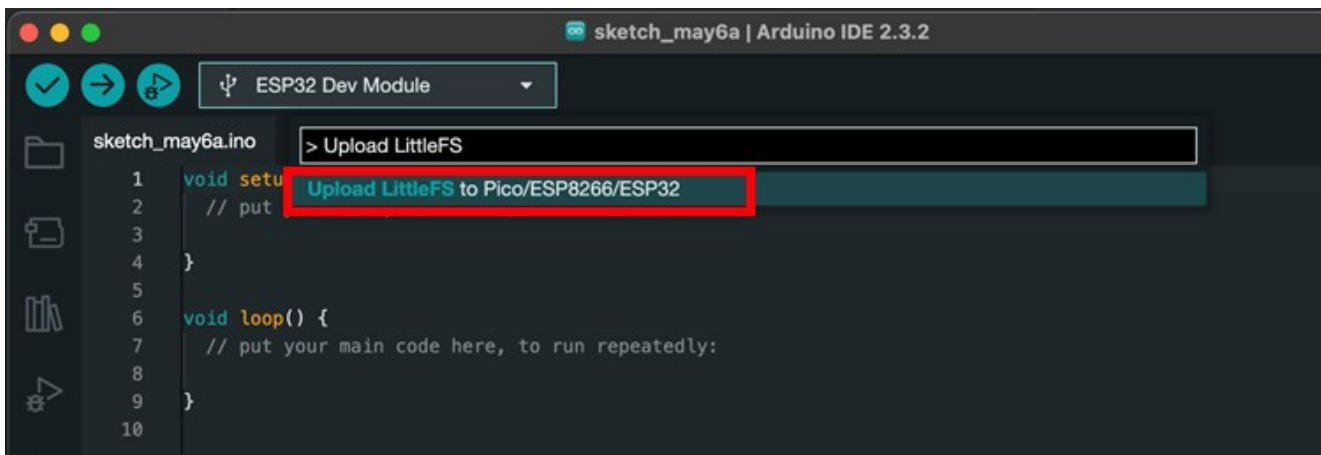
3) Create a new folder called *plugins*.



4) Move the `.vsix` file to the `plugins` folder (remove any other previous versions of the same plugin if that's the case).



5) Restart or open the Arduino IDE 2. To check if the plugin was successfully installed, press `[⌘] + [Shift] + [P]` to open the command palette. An instruction called **'Upload LittleFS to Pico/ESP8266/ESP32'** should be there (just scroll down or search for the name of the instruction).

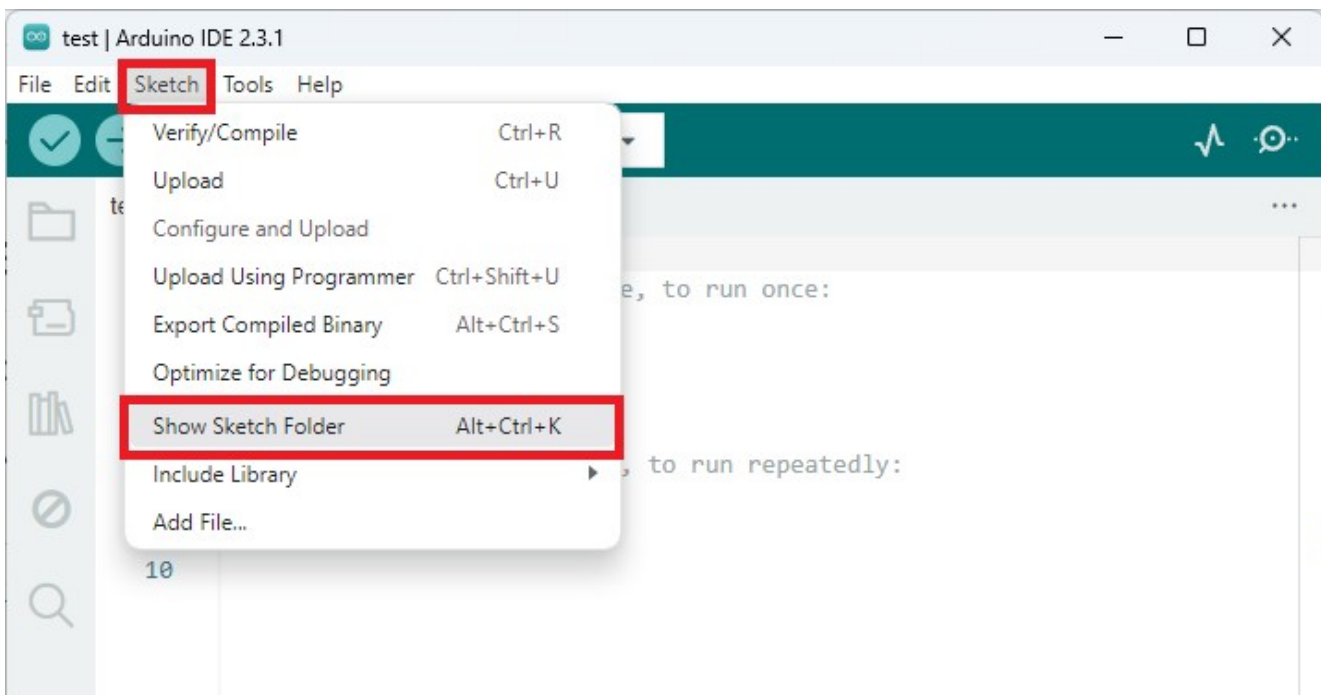


Uploading Files to ESP32 using the Filesystem Uploader in Arduino IDE 2

To upload files to the ESP32 LittleFS filesystem, follow the next instructions.

1) Create an Arduino sketch and save it. For demonstration purposes, you can save an empty sketch.

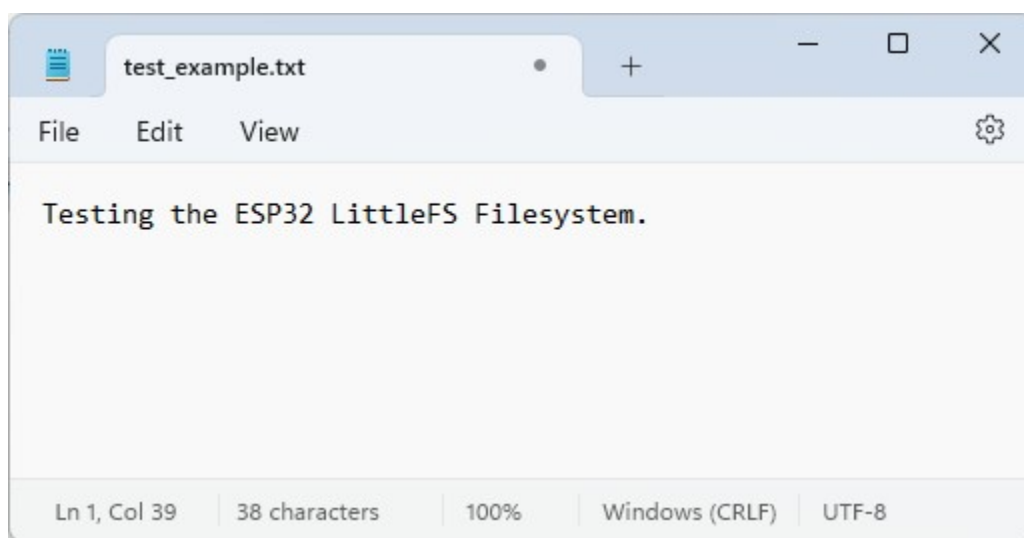
2) Then, open the sketch folder. You can go to **Sketch > Show Sketch Folder**. The folder where your sketch is saved should open.



3) Inside that folder, create a new folder called *data* .



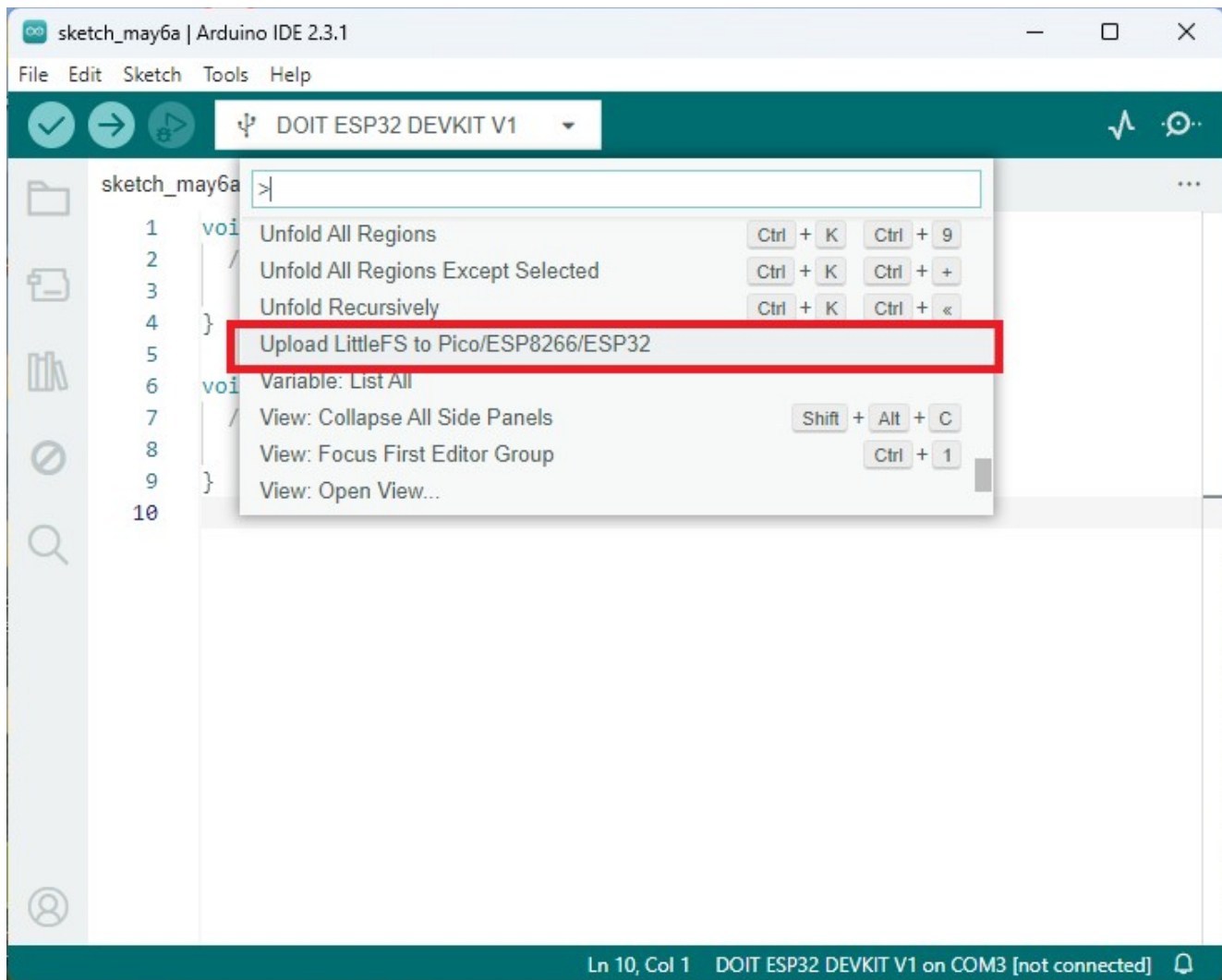
4) Inside the *data* folder is where you should put the files you want to upload to the ESP32 filesystem. As an example, create a *.txt* file with some text called *test_example.txt* (and save it inside the *data* folder).



5) Make sure you have the right board (**Tools > Board**) and COM port selected (**Tools > Port**).

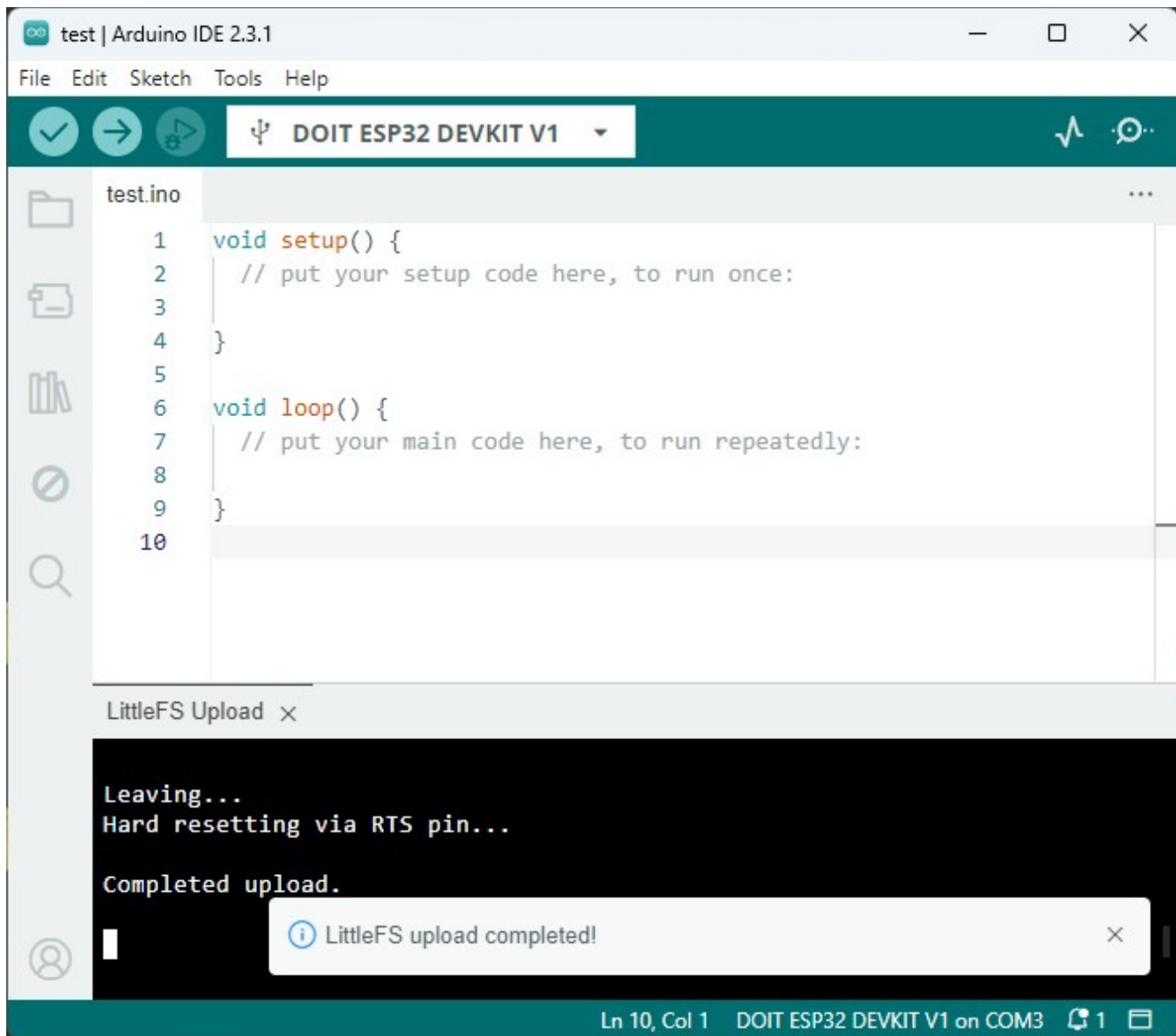
6) Depending on the ESP32 board selected, you may need to select the desired flash size (some boards don't have that option, don't worry). In the Arduino IDE, in **Tools > Flash size**, select the desired flash size (this will depend on the size of your files).

7) Then, upload the files to the ESP32 board. Press **[Ctrl] + [Shift] + [P]** on Windows or **[⌘] + [Shift] + [P]** on MacOS to open the command palette. Search for the **Upload LittleFS to Pico/ESP8266/ESP32** command and click on it.



Important: ensure the Serial Monitor is closed. Otherwise, the upload will fail.

After a few seconds, you should get the message “**Completed upload.** “. The files were successfully uploaded to the ESP32 filesystem.



Troubleshooting

If you get the following error message [“ERROR: No port specified, check IDE menus”](#), restart the Arduino IDE, and try again.

Testing the ESP32 LittleFS Uploader

Now, let's check if the file was saved into the ESP32 filesystem. Upload the following code to your ESP32 board. This code will read the contents of the `.txt` file you saved previously on LittleFS.

```
#include "LittleFS.h"
```

```
void setup() {
  Serial.begin(115200);

  if(!LittleFS.begin()){
    Serial.println("An Error has occurred while mounting LittleFS");
    return;
  }

  File file = LittleFS.open("/test_example.txt", "r");
  if(!file){
    Serial.println("Failed to open file for reading");
    return;
  }

  Serial.println("File Content:");
  while(file.available()){
    Serial.write(file.read());
  }
  file.close();
}

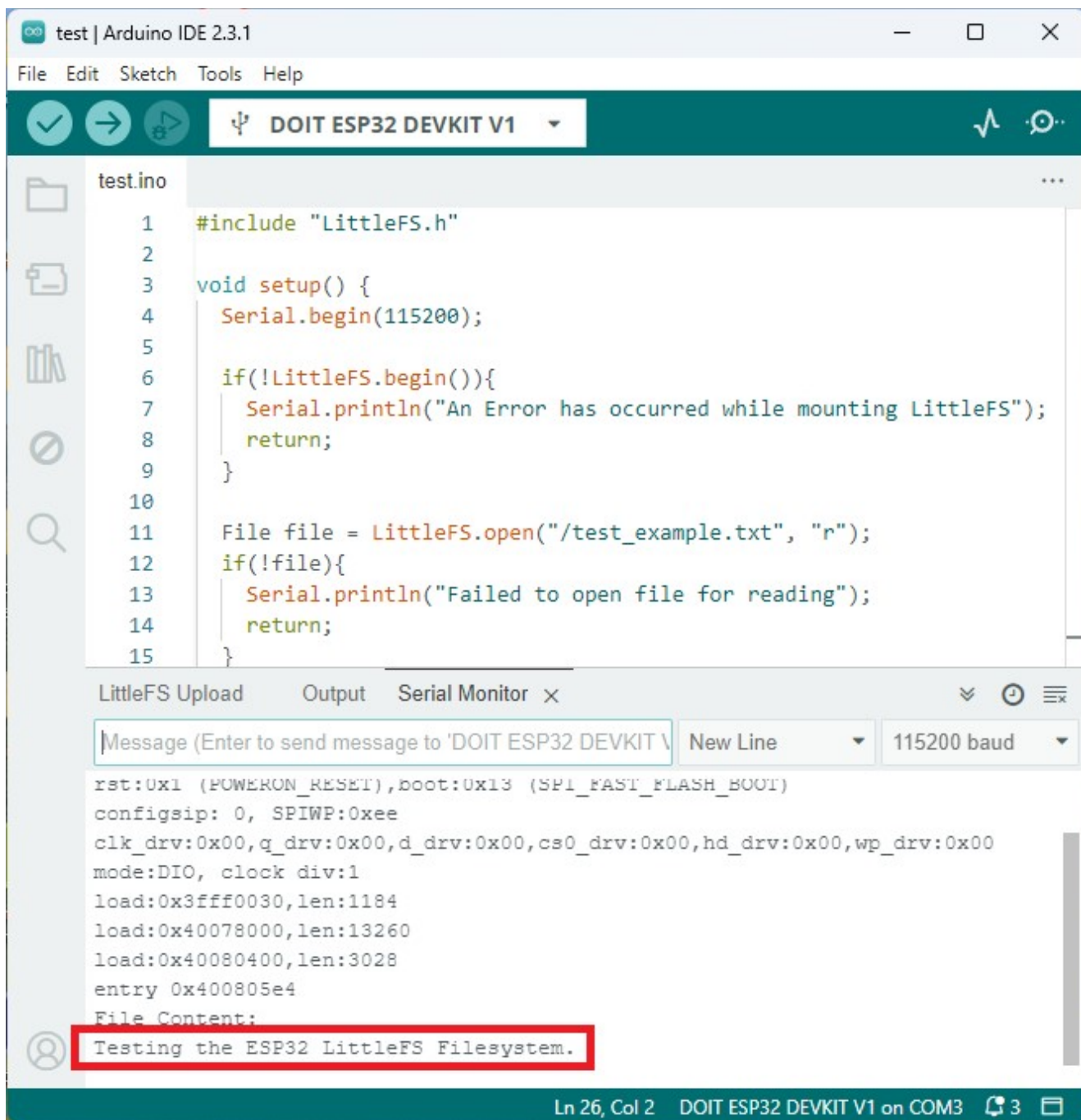
void loop() {
}
```

[View raw code](#)

After uploading, open the Serial Monitor at a baud rate of 115200.



Press the ESP32 on-board “RST” button. It should print the content of your *.txt* file in the Serial Monitor.



You've successfully uploaded files to the ESP32 filesystem using the plugin.

Wrapping Up

In this tutorial, we've shown you how to upload files to the ESP32 LittleFS filesystem on [Arduino IDE 2](#) using an uploader plugin.

We've shown you how to upload a `.txt` file, but you can upload other file formats like HTML, CSS, and Javascript files to build a web server, images, or small icons,

save configuration files, etc.

To learn more about the ESP32, check our resources:

- [Learn ESP32 with Arduino IDE \(eBook\)](#)
- [SMART HOME with Raspberry Pi, ESP32, ESP8266](#)
- [Build Web Servers with ESP32 and ESP8266](#)
- [Firebase Web App with ESP32 and ESP8266](#)
- [More ESP32 Tutorials and Guides](#)

Thanks for reading.



PCB Fabrication & Assembly

ONLY \$5 for 10 PCBs

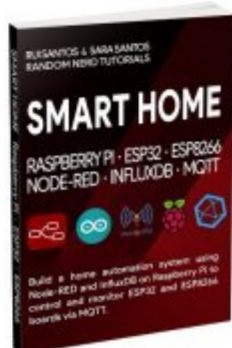
✓ 24-hour Build Time ✓ Quality Guaranteed

✓ Most Soldermask Colors:

Order now



www.pcbway.com



SMART HOME with Raspberry Pi, ESP32, ESP8266 [eBook]

Learn how to build a home automation system and we'll cover the following main subjects: Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT, and InfluxDB database [DOWNLOAD »](#)

Recommended Resources