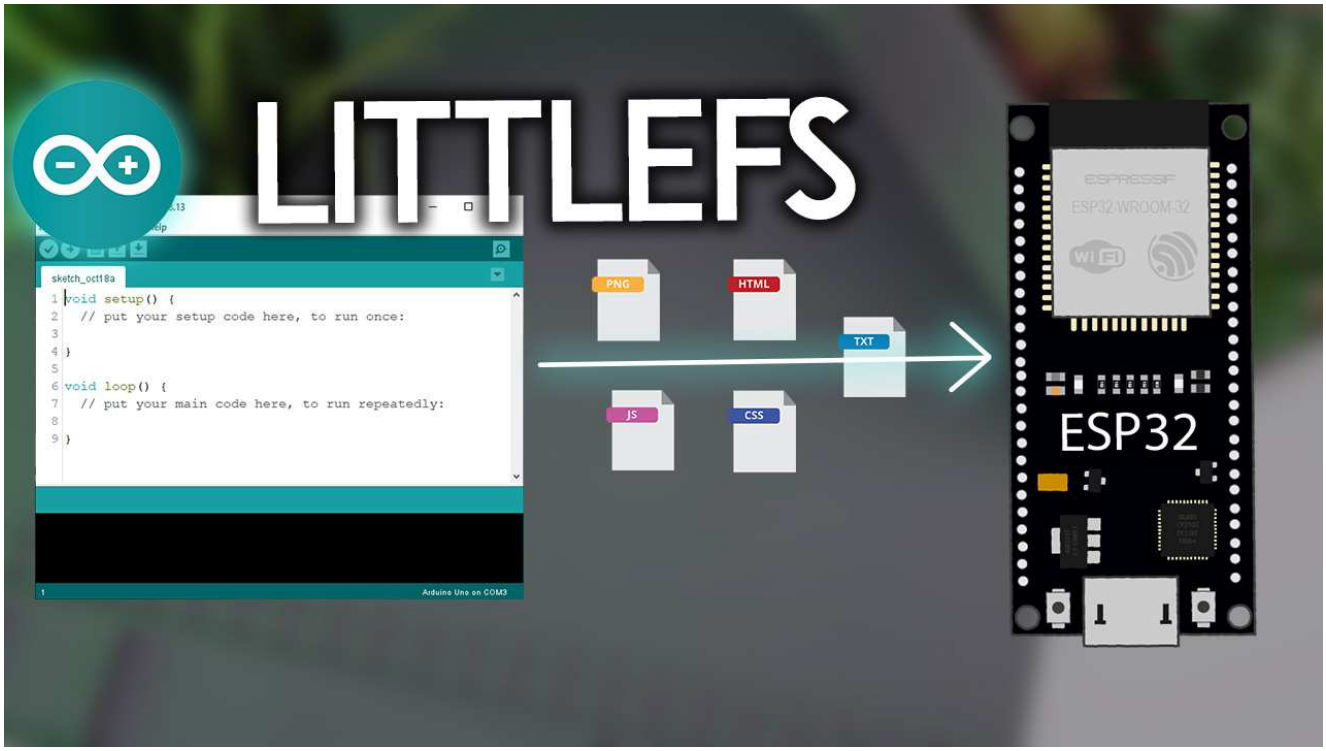
ESP32: Upload Files to LittleFS using Arduino IDE

In this guide, you'll learn how to upload files to the ESP32 Filesystem (LittleFS) by using a plugin for Arduino IDE (1.8.X). LittleFS is a lightweight filesystem created for microcontrollers that lets you access the flash memory like you would do in a standard file system on your computer, but simpler and more limited. The plugin we'll install lets you use three different filesystems: LittleFS, SPIFFS, or FatFS.



> At the moment, this method is **not** compatible with Arduino 2.0. So, you should be using Arduino IDE version 1.8.X.

If you want to use LittleFS with the ESP8266, read: Install ESP8266 NodeMCU LittleFS Filesystem Uploader in Arduino IDE.

Table of Contents

Introducing LittleFS

LittleFS is a lightweight filesystem created for microcontrollers that lets you access the flash memory like you would 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 a filesystem with the ESP32 boards is especially 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 Arduino ESP32 filesystem uploader

Currently, there is a plugin for the Arduino IDE (version 1.8.X) that allows you to pack and upload files to the SPIFFS, LittleFS, or FatFS filesystem image in the ESP32 filesystem.

Note: in most of our projects we use SPIFFS for the ESP32 filesystem. It's still compatible with the ESP32, and you can use SPIFFS without any issues. However, currently, many libraries are moving to LittleFS.

The plugin we'll install is both compatible with SPIFFS and LittleFS. So, it's an advantage over the older plugin and you can still use SPIFFS.

There are a few advantages of using LittleFS over SPIFFS:

- LittleFS is optimized for low resource usage and it employs a **wear-leveling**

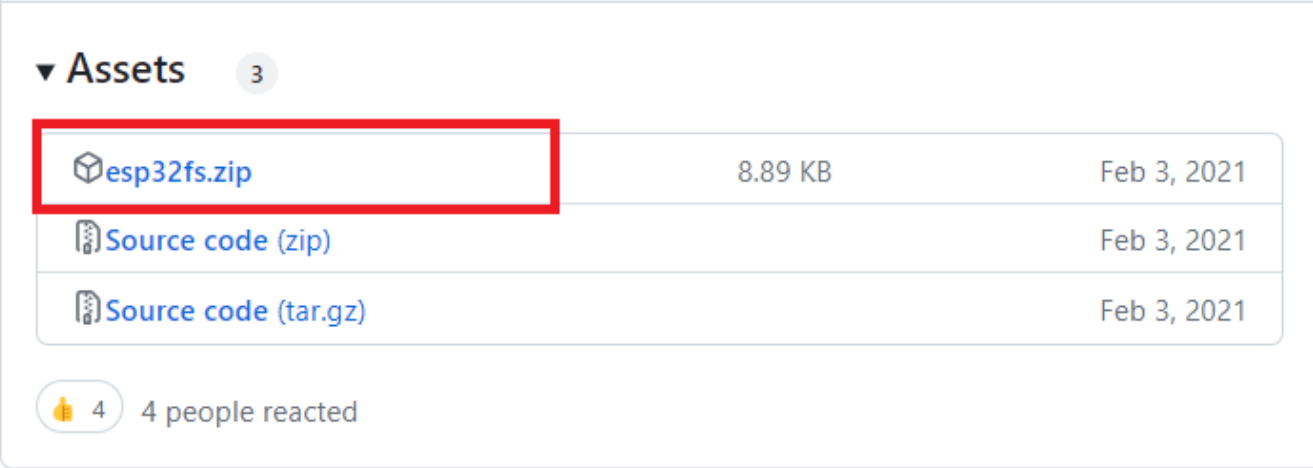
algorithm that evenly distributes writes across the flash memory, prolonging its lifespan.

- LittleFS provides faster mount times and file access by utilizing a **directory indexing structure**.
- LittleFS minimizes the risk of data corruption during power loss or system failures.
- LittleFS is under active development.

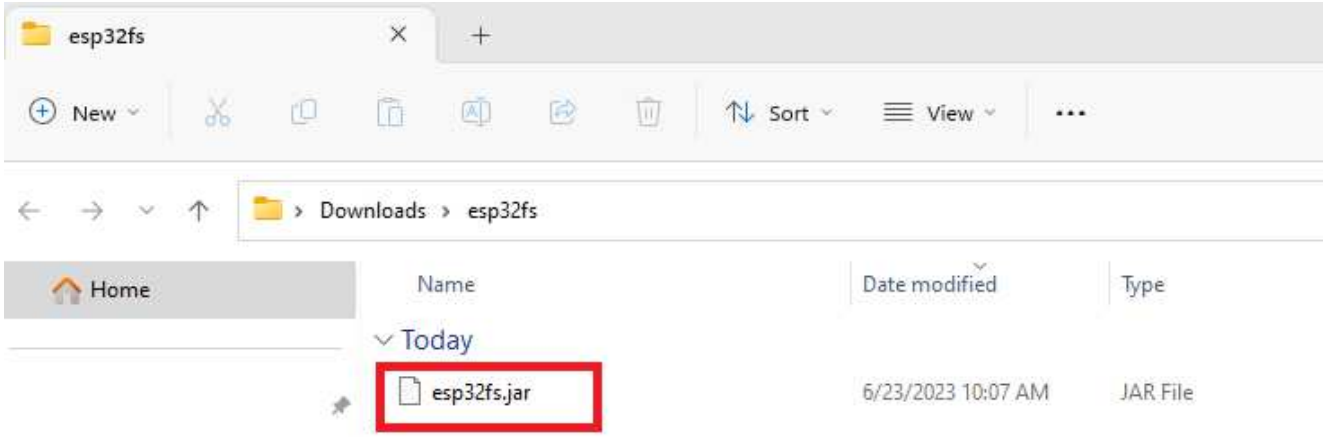
Windows Instructions

Follow the next steps to install the filesystem uploader if you're using Windows:

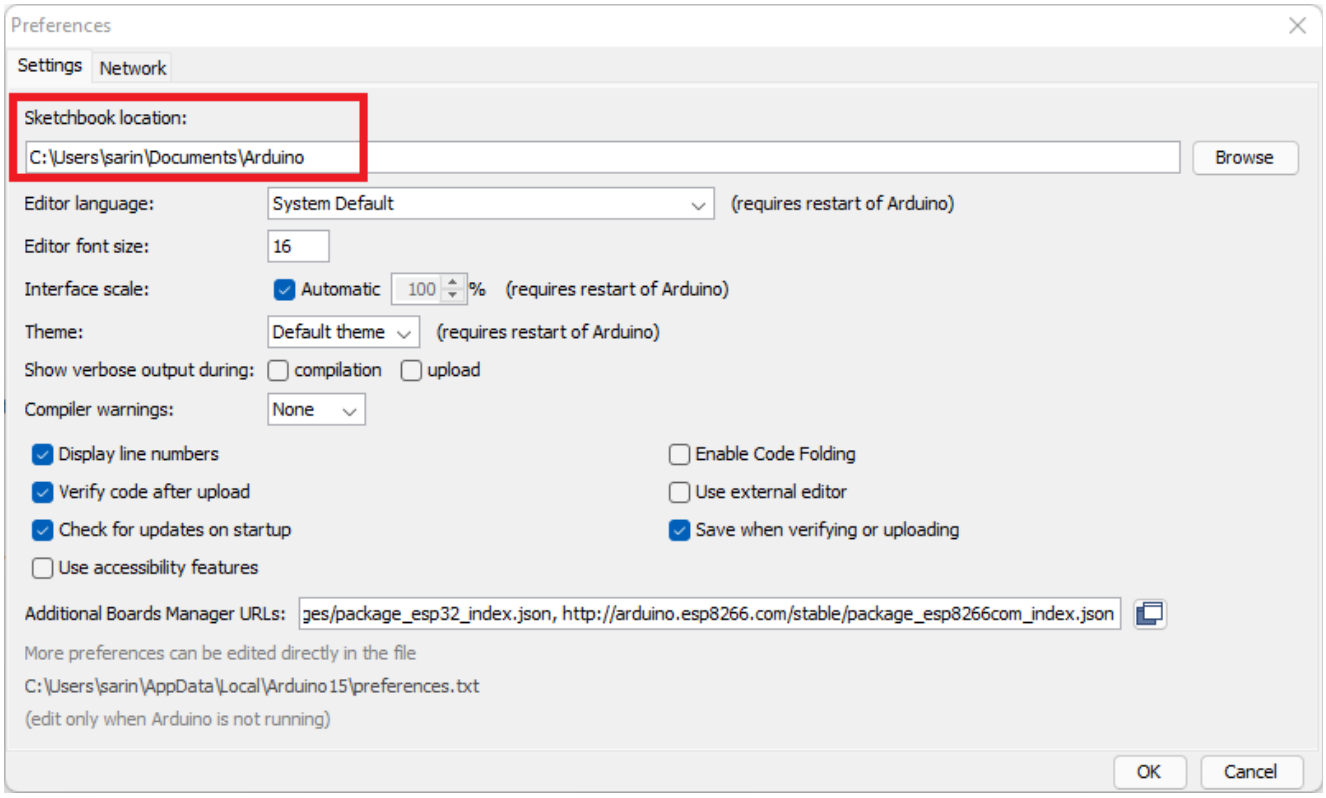
1) Go to the releases page and click the latest **esp32fs.zip** file to download.



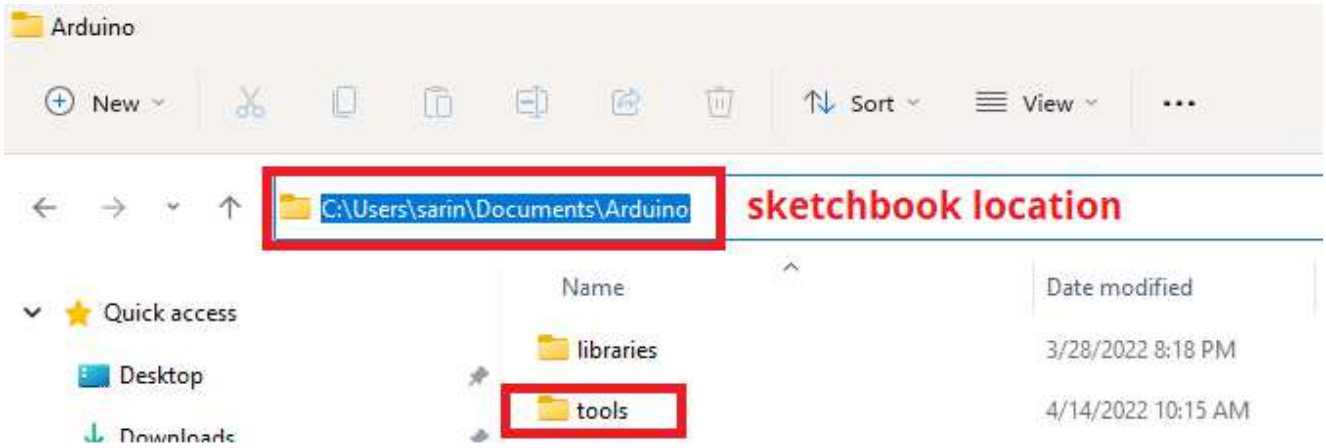
2) Unzip the downloaded file. You should have a folder called **esp32fs** with a file called **esp32fs.jar** inside.



3) Find your Sketchbook location. In your Arduino IDE, go to **File** > **Preferences** and check your Sketchbook location. In my case, it's in the following path: `C:\Users\sarin\Documents\Arduino` .



4) Go to the sketchbook location, and create a **tools** folder if you don't have it already (make sure that the Arduino IDE application is closed).

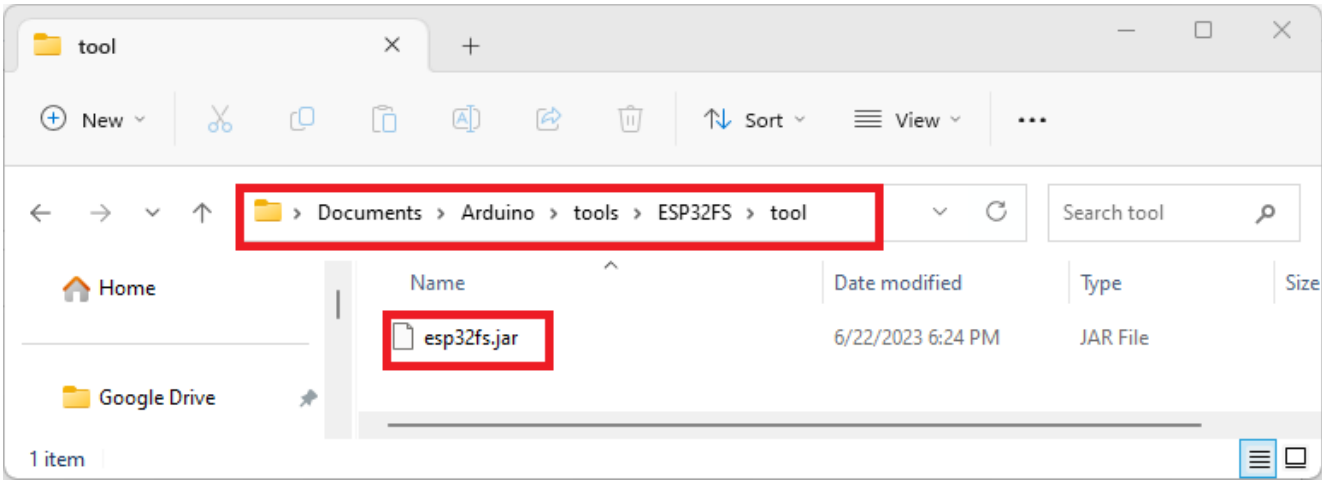


5) Inside the ***tools*** folder, create another folder called ***ESP32FS*** if you haven't already.

6) Inside the **ESP32FS** folder, create a folder called ***tool***.

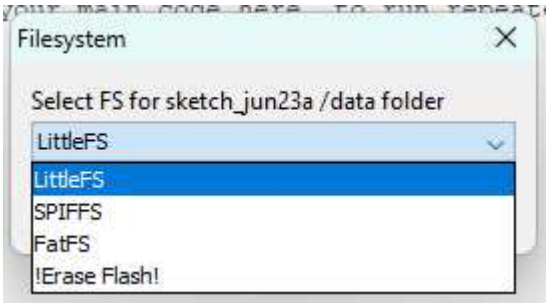
7) Copy the ***esp32fs.jar*** file to the ***tool*** folder (if you already have an ***esp32fs.jar*** file from a previous plugin, delete it and replace it with the new one). So, the directory structure will look like this:

```
<home_dir>/Arduino/tools/ESP32FS/tool/esp32fs.jar
```



8) Now, you can open Arduino IDE.

To check if the plugin was successfully installed, open your Arduino IDE and select your ESP32 board. In the **Tools** menu, check that you have the option "**ESP32 Sketch Data Upload**". Click on that option. A window will pop up for you to choose the filesystem you want to use.



As you can see, you have the option to choose from LittleFS, SPIFFS, or FatFS and you can even have the option to erase flash if needed.

Congratulations! You've successfully installed the Filesystem uploader plugin for the ESP32 on the Arduino IDE.