

---

# SPIFFS

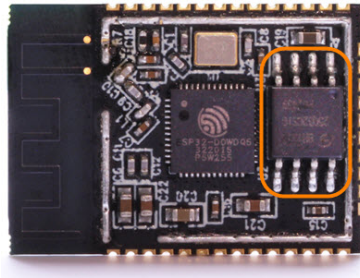
---

- <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/spiffs.html>
- <https://tttapa.github.io/ESP8266/Chap11%20-%20SPIFFS.html>
- <https://randomnerdtutorials.com/install-esp32-file-system-uploader-arduino-ide/>
- <https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/>
- <https://tommydesrochers.com/la-meilleure-methode-pour-heberger-une-interface-web-sur-esp32-version-avancee-esp32-ep4/>
- <https://www.onetransistor.eu/2019/12/upload-files-esp8266-esp32-spiffs.html>

# SPI flash memory

L'ESP32 nécessite une mémoire flash externe pour stocker les programmes, les données, les paramètres de configuration ... Von Neumann !

- La mémoire externe est connectée à la puce via le bus SPI (Serial Peripheral Interface) et la capacité supportée peut atteindre 16Mb.

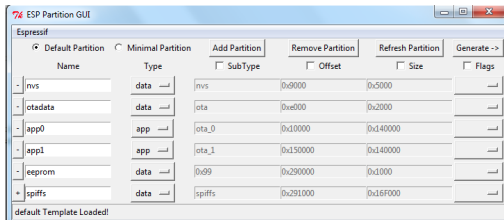


Le module Espressif officiel (ESP-WROOM-32) comprend une mémoire flash de 4Mb fabriquée par GigaDevice (GD25Q32).

- Toutes les cartes n'ont pas la même taille de flash !

# SPIFFS : SPI Flash Filesystem

Cette mémoire flash peut être "partitionnée" et utilisée pour ... **créer un système de fichiers (i.e Filesystem) : SPIFFS**



- ✓ <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/partition-tables.html>
- ✓ <https://github.com/espressif/arduino-esp32/tree/master/tools/partitions>

Selon la carte de développement que vous choisissez, vous pouvez avoir la possibilité de sélectionner la taille SPIFFS ou aucune option du tout.

- La taille SPIFFS est limitée par la taille globale du flash et la taille du programme (esquisse).

L'utilisation d'une partition SPIFFS permet donc de pouvoir y stocker des fichiers :

- des pages Web,
- des images,
- des fichiers de configuration ou CSS,
- ...

**MAIS** il y a des limitations.

SPIFFS est un "système de fichiers" **de base**

- ~~Qui ne prend pas en charge les répertoires, il stocke juste une liste "plate" de fichiers.~~ (ce n'est plus vrai !)
- Mais contrairement aux systèmes de fichiers traditionnels, le caractère barre oblique est autorisé dans les noms de fichiers.
- Il y a une limite de 32 caractères au total pour les noms de fichiers (en fait, seuls 31 sont utilisables).

# SPIFFS : Un exemple simple I

```
1  /*
2  * Fichier esp-spiffs_ls.ino
3  * a) Faire un ls
4  * b) Ouvrir un fichier (si il y est ?)
5  * Si il n'y est pas ... le mettre avec l'IDE Arduino !
6  */
7  #include "SPIFFS.h"
8
9  void setup() {
10     Serial.begin(9600);
11
12     if (!SPIFFS.begin(true)) { // "true" enable format if mount fails (only for ESP32)
13         Serial.println("An Error has occurred while mounting SPIFFS");
14         return;
15     }
16
17     // Getting information from SPIFFS
18     Serial.println("SPIFFS Information:");
19     Serial.print("Total bytes:"); Serial.println(SPIFFS.totalBytes());
20     Serial.print("Used bytes:"); Serial.println(SPIFFS.usedBytes());
21
22     // Make an "ls" from SPIFFS
23     File root = SPIFFS.open("/");
24     File file = root.openNextFile();
25     while(file){
26         Serial.print("FILE:");
27         Serial.println(file.name());
28         file = root.openNextFile();
29     }
30
31     // Find a file and read it
32     if (SPIFFS.exists("/readme.txt")) {
33         Serial.println("Found sample file.");
34         File f = SPIFFS.open("/readme.txt", "r");
```

# SPIFFS : Un exemple simple II

```
35     if (!f) {  
36         Serial.println("Failed to open file for reading");  
37         return;  
38     }  
39     Serial.println("Opened file for reading. File contents:\n");  
40     while (f.available()) {  
41         Serial.write(f.read());  
42     }  
43     f.close();  
44     Serial.println("\n\nEnd of file contents.");  
45 }  
46 else Serial.println("Sample file not found.");  
47 }  
48  
49 void loop() {}
```

# Plugins IDE Arduino

L'esptool et les autres outils (dont Arduino IDE) basés sur celui-ci ont la **possibilité de télécharger des fichiers** vers SPIFFS.

Pour ESP32, utilisez le plugin ESP32FS :

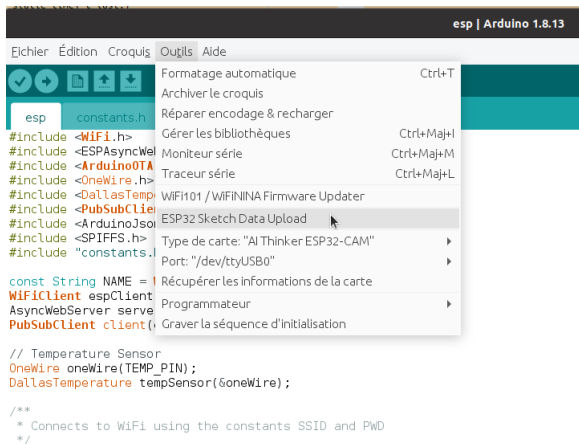
<https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/download/1.0/ESP32FS-1.0.zip>

- Téléchargez l'archive => contient un dossier "ESP32FS".
- Placez ce dossier dans l'emplacement de votre carnet de croquis (sketchbook), sous-dossier "tool" (Créez ce sous-dossier s'il n'existe pas).

Le dossier du carnet de croquis est par défaut dans Documents de l'utilisateur, Arduino pour les utilisateurs Windows.

Sous Linux :

```
$ ls ~/Arduino/tools/ESP32FS/tool/  
esp32fs.jar
```



Ce nouveau menu permet de placer en SPIFFs les fichiers qui se trouve dans le répertoire data qui "accompagne" le sketch :

```
$ cat ../TPs/sketchbook/esp-spiffs_ls/data/readme.txt
Bonjour !
```



```
esp-spiffs_ls $
#include "SPIFFS.h"

void setup() {
  Serial.begin(9600);

  if (!SPIFFS.begin(true)) { // "true" enable format if mount fails (only for ESP32)
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
  }

  // Getting information from SPIFFS
  Serial.println("SPIFFS Information:");
  Serial.print("Total bytes: "); Serial.println(SPIFFS.totalBytes());
  Serial.print("Used bytes: "); Serial.println(SPIFFS.usedBytes());

  // Make an "ls" from SPIFFS
  File root = SPIFFS.open("/");
  File file = root.openNextFile();
  while(file){
    Serial.print("FILE: ");
    Serial.println(file.name());
    file = root.openNextFile();
  }

  // Find a file and read it
  if (SPIFFS.exists("/readme.txt")) {
    Serial.println("Found sample file.");
    File f = SPIFFS.open("/readme.txt", "r");
    if (!f) {
      Serial.println("Failed to open file for reading");
      return;
    }
    Serial.println("Opened file for reading. File contents:\n");

    while (f.available()) {
      Serial.write(f.read());
    }
    f.close();
    Serial.println("\n\nEnd of file contents.");
  }
  else Serial.println("Sample file not found.");
}

void loop() {}
```

```
10:54:58.171 -> .....{tes: 896321
10:55:22.586 -> Used bytes: 709075
10:55:22.619 -> FILE: /test.txt
10:55:22.686 -> FILE: /readme.txt
10:55:22.686 -> Found sample file.
10:55:22.719 -> Opened file for reading. File contents:
10:55:22.752 ->
10:55:22.752 -> Bonjour !
10:55:22.785 ->
10:55:22.785 ->
10:55:22.785 -> End of file contents.
```

☒ Défilement automatique ☒ Afficher l'horodatage

# SPIFFS : Un exemple plus complet

[https://raw.githubusercontent.com/espressif/arduino-esp32/master/libraries/SPIFFS/examples/SPIFFS\\_Test/SPIFFS\\_Test.ino](https://raw.githubusercontent.com/espressif/arduino-esp32/master/libraries/SPIFFS/examples/SPIFFS_Test/SPIFFS_Test.ino)

Dans cet exemple, vous trouverez certainement votre bonheur pour manipuler ce SPIFFS !

```
1  /*
2  * Fichier esp-spiffs_test.ino
3  *
4  *
5  */
6  #include "FS.h"
7  #include "SPIFFS.h"
8
9  /* You only need to format SPIFFS the first time you run a
10     test or else use the SPIFFS plugin to create a partition
11     https://github.com/me-no-dev/arduino-esp32fs-plugin */
12
13 #define FORMAT_SPIFFS_IF_FAILED true
14
15
16 void listDir(fs::FS &fs, const char * dirname, uint8_t levels){
17     Serial.printf("Listing directory: %s\r\n", dirname);
18
19     File root = fs.open(dirname);
20     if(!root){
21         Serial.println("- failed to open directory");
22         return;
23     }
24     if(!root.isDirectory()){
25         Serial.println("- not a directory");
26         return;
```

# SPIFFS : Un exemple plus complet II

```
27     }
28
29     File file = root.openNextFile();
30     while( file ){
31         if( file.isDirectory() ){
32             Serial.print("DIR:");
33             Serial.println( file.name() );
34             if( levels ){
35                 listDir( fs, file.name(), levels -1 );
36             }
37         } else {
38             Serial.print("FILE:");
39             Serial.print( file.name() );
40             Serial.print("\tSIZE:");
41             Serial.println( file.size() );
42         }
43         file = root.openNextFile();
44     }
45 }
46
47 void readFile( FS &fs, const char * path ){
48     Serial.printf("Reading file : %s\r\n", path);
49
50     File file = fs.open(path);
51     if( !file || file.isDirectory() ){
52         Serial.println("-failed to open file for reading");
53         return;
54     }
55
56     Serial.println("-read from file:");
57     while( file.available() ){
58         Serial.write( file.read() );
59     }
60     file.close();
```

# SPIFFS : Un exemple plus complet III

```
61 }
62
63 void writeFile(fs::FS &fs, const char * path, const char * message){
64     Serial.printf("Writing file: %s\r\n", path);
65
66     File file = fs.open(path, FILE_WRITE);
67     if(!file){
68         Serial.println("-failed to open file for writing");
69         return;
70     }
71     if(file.print(message)){
72         Serial.println("-file written");
73     } else {
74         Serial.println("-write failed");
75     }
76     file.close();
77 }
78
79 void appendFile(fs::FS &fs, const char * path, const char * message){
80     Serial.printf("Appending file: %s\r\n", path);
81
82     File file = fs.open(path, FILE_APPEND);
83     if(!file){
84         Serial.println("-failed to open file for appending");
85         return;
86     }
87     if(file.print(message)){
88         Serial.println("-message appended");
89     } else {
90         Serial.println("-append failed");
91     }
92     file.close();
93 }
94
```

# SPIFFS : Un exemple plus complet IV

```
95 void renameFile(fs::FS &fs, const char * path1, const char * path2){
96     Serial.printf("Renaming file %s to %s\r\n", path1, path2);
97     if (fs.rename(path1, path2)) {
98         Serial.println("_file renamed");
99     } else {
100         Serial.println("_rename failed");
101     }
102 }
103
104 void deleteFile(fs::FS &fs, const char * path){
105     Serial.printf("Deleting file: %s\r\n", path);
106     if (fs.remove(path)){
107         Serial.println("_file deleted");
108     } else {
109         Serial.println("_delete failed");
110     }
111 }
112
113 void testFileIO(fs::FS &fs, const char * path){
114     Serial.printf("Testing file I/O with %s\r\n", path);
115
116     static uint8_t buf[512];
117     size_t len = 0;
118     File file = fs.open(path, FILE_WRITE);
119     if(!file){
120         Serial.println("_failed to open file for writing");
121         return;
122     }
123
124     size_t i;
125     Serial.print("_writing ");
126     uint32_t start = millis();
127     for(i=0; i<2048; i++){
128         if ((i & 0x001F) == 0x001F){
```

# SPIFFS : Un exemple plus complet V

```
129         Serial.print(".");
130     }
131     file.write(buf, 512);
132 }
133 Serial.println("");
134 uint32_t end = millis() - start;
135 Serial.printf("_%u bytes written in %u ms\r\n", 2048 * 512, end);
136 file.close();
137
138 file = fs.open(path);
139 start = millis();
140 end = start;
141 i = 0;
142 if (file && !file.isDirectory()){
143     len = file.size();
144     size_t flen = len;
145     start = millis();
146     Serial.print("_reading ");
147     while(len){
148         size_t toRead = len;
149         if (toRead > 512){
150             toRead = 512;
151         }
152         file.read(buf, toRead);
153         if ((i++ & 0x001F) == 0x001F){
154             Serial.print(".");
155         }
156         len -= toRead;
157     }
158     Serial.println("");
159     end = millis() - start;
160     Serial.printf("_%u bytes read in %u ms\r\n", flen, end);
161     file.close();
162 } else {
```

# SPIFFS : Un exemple plus complet VI

```
163         Serial.println("-_failed_to_open_file_for_reading");
164     }
165 }
166
167 void setup(){
168     Serial.begin(9600);
169     if(!SPIFFS.begin(FORMAT_SPIFFS_IF_FAILED)){
170         Serial.println("SPIFFS_Mount_Failed");
171         return;
172     }
173
174     listDir(SPIFFS, "/", 0);
175     writeFile(SPIFFS, "/hello.txt", "Hello_");
176     appendFile(SPIFFS, "/hello.txt", "World!\r\n");
177     readFile(SPIFFS, "/hello.txt");
178     renameFile(SPIFFS, "/hello.txt", "/foo.txt");
179     readFile(SPIFFS, "/foo.txt");
180     deleteFile(SPIFFS, "/foo.txt");
181     testFileIO(SPIFFS, "/test.txt");
182     deleteFile(SPIFFS, "/test.txt");
183     Serial.println("Test_complete");
184 }
185
186 void loop(){
187
188 }
```