

Dateien auf dem ESP8266 speichern mit LittleFS



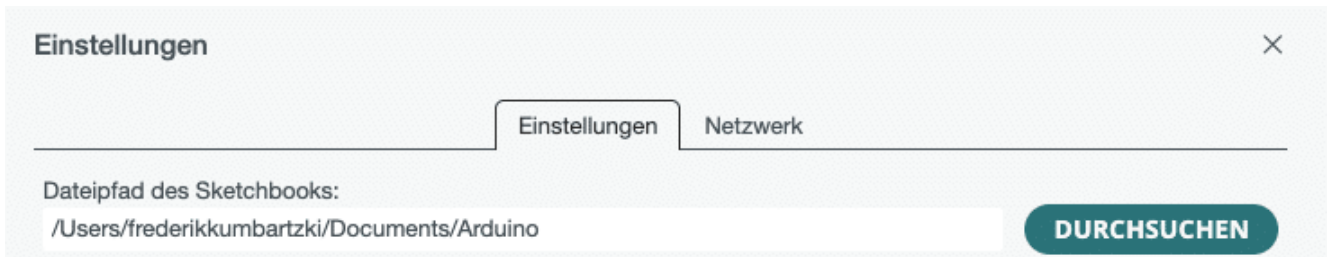
Für einige Projekte ist es praktisch, Dateien auf einem ESP8266 zu speichern – z.B. weil du Sensordaten sichern möchtest, oder auch die HTML-, CSS- und JavaScript-Dateien eines Webserverns sauber ablegen möchtest. Hier kommt das Tool **LittleFS** ins Spiel, mit dem du das problemlos über die Arduino IDE erledigen kannst.

das Tool herunterladen und installieren

Ein Hinweis vorab: Aktuell (Jan. 2024) funktioniert LittleFS nicht mit der Arduino IDE ab Version 2.0 – für dieses Tutorial habe ich deshalb die Version 1.8.19 verwendet. Diese kannst du nach wie vor [hier im Bereich Legacy IDE herunterladen](#).

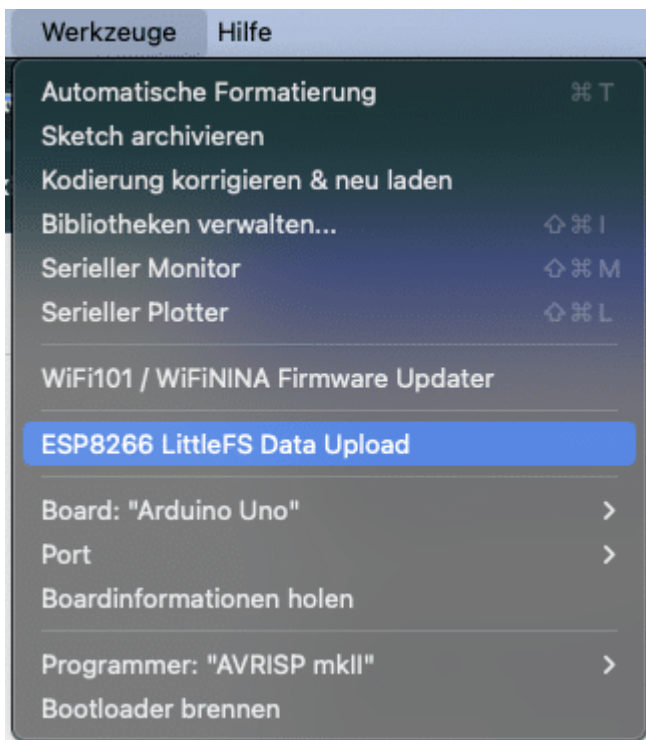
Du findest die aktuelle Version von LittleFS auf [GitHub](#). Aktuell ist das die Version [ESP8266LittleFS-2.6.0.zip](#), die du mit einem Klick auf den Dateinamen herunterladen kannst.

Steuere nun den Ordner deines Sketchbooks an. Das ist der Ort, in dem all deine mit der Arduino IDE erstellten Sketches wiederum in eigenen Ordnern gespeichert werden. Falls du dir nicht sicher bist, findest du den Pfad zum Ordner in den Einstellungen der Arduino IDE:



Erstelle in deinem Sketchbook-Ordner einen neuen Ordner mit dem Namen **tools** und entpacke dort die LittleFS ZIP-Datei, die du gerade heruntergeladen hast. In deinem Ordner **tools** sollte nun der entpackte Ordner **ESP8266LittleFS** zu finden sein.

Wenn du fertig bist, starte die Arduino IDE neu. Öffne nun das Menü **Werkzeuge** – dort siehst du nun den Eintrag **ESP8266 LittleFS Data Upload**.



Dateien auf den ESP8266 hochladen

Jetzt, wo alles eingerichtet ist, kann es losgehen. Erstelle einen neuen Sketch und öffne den dazugehörigen Ordner – z.B.

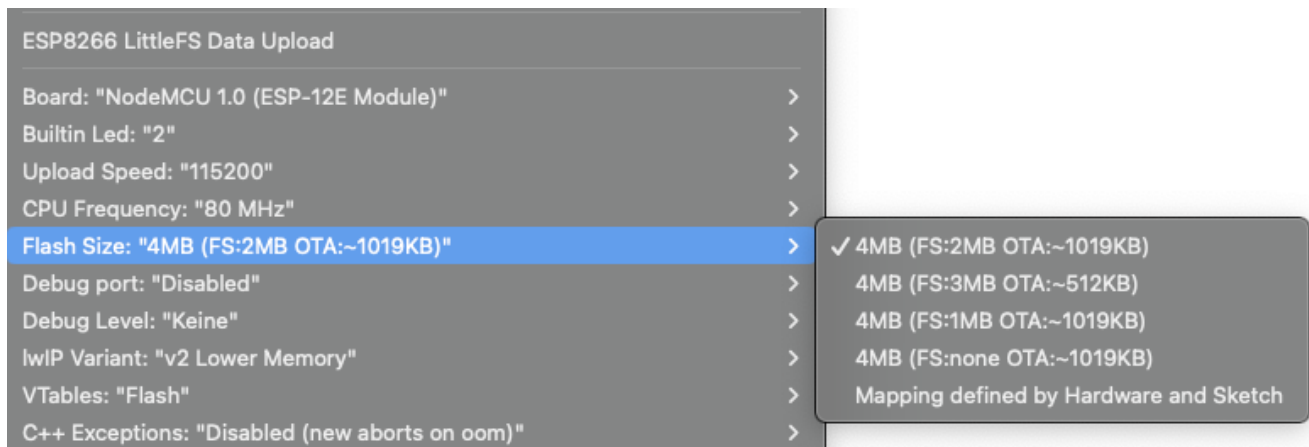
über den Menüpunkt **Sketch > Sketch-Ordner anzeigen**. Erstelle in diesem Ordner nun einen weiteren Ordner namens **data**.

Alle Dateien, die du später in diesem Ordner ablegst, werden mit dem gerade installierten Tool auf den ESP8266 geladen. Erstelle nun testweise eine einfache HTML-Datei mit folgendem Inhalt:

Wenn du das hier liest, funktioniert LittleFS. :)

Speichere diese Datei nun im Ordner **data** mit dem Namen **test.txt** ab. Später wirst du diese Datei in deinem Seriellen Monitor auslesen. Doch zunächst der Upload:

Wähle zunächst (falls noch nicht geschehen) im Menü **Werkzeuge** deinen ESP8266 aus. Anschließend kannst du im selben Menü unter **Flash size** die benötigte Größe für den Speicher einstellen. Für den folgenden Test musst du hier jedoch nichts umstellen.



Wähle anschließend den Menüpunkt **ESP8266 LittleFS Data Upload**. Nun wird die HTML-Datei im Ordner **data** übertragen – was nach wenigen Sekunden erfolgreich abgeschlossen sein sollte:

```
Compressed 2072576 bytes to 2393...
Writing at 0x00200000... (100 %)
Wrote 2072576 bytes (2393 compressed) at 0x00200000 in 0.2 seconds (effective 77387.7 kbit/s)
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Die TXT-Datei anzeigen

Ob der Upload wirklich funktioniert hat, lässt sich natürlich am besten überprüfen, indem du nachschaust, ob du den Inhalt von **test.txt** auslesen kannst.

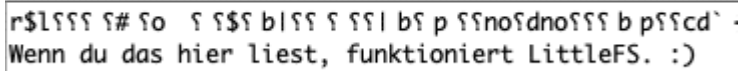
Füge hierfür den folgenden Code in deinen Sketch ein:

```
#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.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() {
}
```

Lade nun den Sketch auf deinen ESP8266 und öffne den Seriellen Monitor. Um die Datei aufzurufen, drücke kurz den Reset-Button **RST** – nun sollte der Inhalt von **test.txt** in Seriellen Monitor erscheinen:



```
r$1111 1# 1o 1 1$1 b111 1 111 b1 p 11no1dno111 b p11cd` .  
Wenn du das hier liest, funktioniert LittleFS. :)
```

In diesem Tutorial hast du die Grundlagen von LittleFS gelernt. Du weißt nun, wo du die Dateien im Sketch-Ordner ablegen musst und wie du diese auf deinen ESP8266 hochlädst.