# ذخیره اطلاعات در حافظه FLASH ماژول ESP8266 و کار با LITTLEFS

تاریخ انتشار۱۸ تیر, ۱۴۰۱ توسط  سید حسین سلطانی

سلام خدمت همه شما مایکروالکامی ها. در مطالب قبلی از سری مطالب مربوط به ماژول WiFi مدل ESP8266 به راه اندازی و اجرای حافظه EEPROM در ماژول وایفای ESP8266 پرداخته شد. همانطور که بیان شد با هر بار قطع و وصل دیتا ذخیره شده از بین خواهد رفت. لذا برای حل این مشکل از فایل سیستم جهت ذخیره دیتا برروی حافظه فلش (FLASH IC) موجود استفاده می‌کنیم. در این مطلب نحوه استفاده از فایل سیستم LittleFS و ذخیره دیتا برروی حافظه

فلش ESP8266 بدون از بین رفتن بعد از هر ریست شدن یا خاموش و روشن شدن پرداخته خواهد شد. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتویند سایر مطالب من رو از این لینک مطالعه و بررسی کنید.

---

## سیستم فایل LittleFS و SPIFFS در ESP8266

در تراشه ها یا بهتر بگویم SoC های وایفای سری ESP8266 یک فایل سیستم روی همان IC حافظه فلش (Flash IC) که مربوط به کد های اصلی برنامه است، می‌تواند پیاده سازی و استفاده شود. فایل سیستم با پارتیشن بندی حافظه Flash موجود به بخش های کد، آپدیت OTA، سیستم فایل و پیاده سازی حافظه EEPROM و تنظیمات مربوط به بخش WiFi ایجاد می‌شود. در ESP8266 دو نوع فایل سیستم با عناوین SPIFFS و LittleFS برای استفاده وجود دارد.

امروزه بیان می‌شود که ممکن است در نسل های بعدی از ESP8266 دیگر SPIFFS پشتیبانی نشده و طبعاً بهتر از است از نوع دیگر یعنی LittleFS بهره برد. همچنین این فایل سیستم سرعت عملکردی بیشتری نیز دارد. لذا از همین رو در این مطلب به ذکر همین مورد پرداخته خواهد شد. در جدول زیر این دو فایل سیستم مقایسه شده است. با استفاده از LittleFS قادر خواهیم بود تا برروی حافظه Flash موجود برروی ماژول وایفای ESP8266 اطلاعات مورد نیاز خود را ذخیره و برخلاف مطلب قبلی، با قطع و وصل شدن ماژول یا ریست آن اطلاعات پاک نشود.

### مقایسه LittleFS و SPIFFS

| ویژگی | SPIFFS | LittleFS |
|---|---|---|
| **پشتیبانی از دایرکتوری** | خیر | بله |
| **overhead (سر بار) هر فایل** | حداقل 256byte | حداقل 4K |
| **سرعت** | کند | سریع تر |
| **File Timestamp** | ندارد | دارد |

---

## معرفی LittleFS

LittleFS با تمرکز بر سرعت عملکرد بالا تر و پشتیبانی از دایرکتوری باعث بهتر شدن استفاده از فایل سیستم شده

است. همچنین مزایای LittleFS نسبت به SPIFFS به مراتب بهتر است. با استفاده از LittleFS می‌توان فایل ها را در حافظه فلش بدون استفاده از حافظه های خارجی و اضافی دیگر انجام دهیم. با استفاده از این قابلیت کاربر قادر خواهد بود فایل ها و اطلاعات را از/به حافظه فلش میکروکنترلر بخواند یا بنویسد.

اگر قبلا در پروژه های خود از SPIFFS استفاده کرده اید جای نگرانی برای مهاجرت به LittleFS نیست. کافی است کتابخانه LittleFS.h را جایگزین SPIFFS.h در برنامه خود کنید. همچنین سایر متد ها و دستورات SPIFFS را با LittleFS جایگزین کنید. مثلا بجای SPIFFS.begin() آن را بصورت LittleFS.begin() تغییر دهید.

## پتشتیبانی از دایرکتوری در LittleFS

مشابه یک فایل سیستم لینوکس، در ESP8266 از طریق LittleFS می‌توان از دایرکتوری های واقعی پشتیبانی کرد. می‌توان نام فایل را تا 31 کاراکتر و به همین میزان دایرکتوری و زیر مجموعه دایرکتوری متناسب با مقدار حافظه فلش ESP8266 ایجاد نماییم. چنانچه در حافظه دایرکتوری و مسیری ایجاد نشده باشد، بصورت پیشفرض فایل ها در root (معنی تحت اللفظی=ریشه) "/" ذخیره خواهند شد.

## کاربرد

کاربرد LittleFS در ماژول ESP8266 و برد های آن، می‌تواند موارد مختلفی باشد. در زیر چند مورد از آن بیان شده است. شایان ذکر است شما می‌توانید افزونه این فایل سیستم را در محیط Arduino IDE براحتی نصب و اضافه نموده و کار خود را انجام دهید. اما در این قسمت صرفا از کتابخانه آن استفاده خواهد شد.

- ذخیره برخی تنظیمات مد نظر بصورت دائمی و بدون نیاز به SD کارت ها
- ذخیره فایل و کد های راه انداز لوکال وب سرور
- در برخی موارد پیاده سازی یک دیتا بیس ساده یا lookup table جهت استفاده از داده های مورد نظر

---

## کتابخانه LittleFS

جهت استفاده از این فایل سیستم باید کتابخانه مربوطه را از گیت هاب یا طریق این لینک دانلود کنید. پس از نصب فایل zip از مسیر زیر جهت نصب و افزدون کتابخانه به Arduino IDE اقدام نمایید.

انتخاب فایل دانلود شده از محل ذخیره فایل دانلودی <-- *Sketch --> Include Library --> Add .ZIP Library*

---

## قطعات مورد نیاز

- ماژول ESP8266
- برد بورد
- مبدل سریال
- سیم برد بوردی
- کلید فشاری
- مبدل کاهنده ولتاژ

---

## نوشتن دیتا توسط LittleFS در ESP8266

جهت استفاده از فایل سیستم LittleFS ابتدا باید کتابخانه مورد را در ابتدای برنامه معرفی کنیم. پیش از شروع کار خواندن یا نوشتن، باید از آماده بودن فایل سیستم از طریق LittleFS.begin() مطلع شویم. همچنین برای نوشتن یک دیتا برروی حافظه Flash موجود در ESP8266 با دستور LittleFS.open() دایرکتوری مورد نظر را باز یا آدرس مورد نظر را داده و در نهایت با انتخاب یک نام با پسوند ".txt " برای فایل، با دستور print محتوای متنی مورد نظر خود را درون آن قرار خواهیم داد. برای آگاهی از این روند، به تکه کد زیر دقت فرمایید.

```
#include <LittleFS.h>

String command;
String text = "دیتا مورد نظر خود را برای نوشتن در حافظه در این قسمت وارد کنید.";

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

```
 اطلاع از آماده بودن فایل سیستم//
  if(!LittleFS.begin())
  {
    Serial.println("An ERROR has occured in LittleFS!");
  }
  else Serial.println("LittleFS is begin...");
}

void loop()
{
 دریافت فرامین از کاربر از طریق سریال مانیتور جهت نوشتن/خواندن/حذف کردن//
در این تکه کد صرفا وضعیت نوشتن بررسی شده است//  .
  if(Serial.available())
  {
    command = Serial.readString();
    Serial.println(command);
    if(command == "write data")
    {
   ابتدا یک فایل txt در دایرکتوری ایجاد میکنیم.//
      File file = LittleFS.open("/FileTest.txt", "w");
   نوشتن در فایل ایجاد شده و بررسی کامل شدن فرایند write کردن آن//
      if (file.print(text))
      {
        Serial.println("File Written Successfully.");
      }
      else
      {
        Serial.println("File Written Failed.");
      }
      delay(1);
      file.close();
    }
```

```
  }
}
```

## خواندن دیتا از طریق LittleFS در ESP8266

برای خواندن دیتا، ابتدا باید مسیر و نام فایل در دایرکتوری را داده و بعد از آن بررسی کنیم آیا این فایل وجود دارد یا خیر. در صورت وجود داشتن آن توسط دستور readString() نسبت به خواندن محتوای مربوطه بصورت رشته (چون محتوایی که پیشتر نوشته شد بصورت رشته بوده است) و ذخیره آن نیز در یک String اقدام نماییم. جهت آگاهی از روند کلی خواندن به تکه کد زیر دقت نمایید.

```
#include <LittleFS.h>

String command, text;

void setup()
{
  Serial.begin(115200);
 //اطلاع از آماده بودن فایل سیستم
  if(!LittleFS.begin())
  {
    Serial.println("An ERROR has occurred in LittleFS!");
  }
  else Serial.println("LittleFS is begin...");
}

void loop()
{
 //دریافت فرامین از کاربر از طریق سریال مانیتور جهت نوشتن/خواندن/حذف کردن
//در این تکه کد صرفا وضعیت خواندن بررسی شده است .
```

```
  if(Serial.available())
  {
    command = Serial.readString();
    Serial.println(command);
    if(command == "read data")
    {
 باز کردن فایل مورد نظر ما در دایرکتوری //
      File file = LittleFS.open("/FileTest.txt", "r");
 بررسی کردن وجود فایل مورد نظر //
      if(!file) // معرفی شدهشرط مربوط به عدم وجود فایل
      {
        Serial.println("File does not Exists");
      }
      else  // شرط مربوط به وجود فایل معرفی شده
      {
        Serial.println("File Exists");
   خواندن از روی فایل و نمایش آن//
        while(file.available())
        {
    چون دیتا بصورت رشته ذخیره شده است، خواندن آن هم باید بصورت رشته باشد//
          Serial.println(file.readString());
        }
        file.close(); //بستن فایل
      }
    }
  }
}
```

## حذف کردن یک فایل با استفاده از LittleFS در ESP8266

برای حذف کردن یک فایل در دایرکتوری مورد نظر، از دستور remove() استفاده می‌کنیم. برای حذف فایل، تکه کد زیر را در محل مورد نظر برنامه خود قرار دهید. البته می‌توان مانند مرحله خواندن ابتدا چک کردن که آیا فایل مورد نظر وجود دارد یا خیر و چنانچه وجود داشت آن را حذف و در غیر اینصورت یک پیغام خطا نمایش داد.

```
LittleFS.remove("/FileTest.txt");
```

---

## مثال نوشتن و خواندن دیتا روی حافظه Flash در ESP8266

با آپلود برنامه زیر برروی ماژول ESP8266 و باز کردن سریال مانیتور و ارسال فرامین مربوط به نوشتن (write data) یا خواندن (read data) باید همان دیتا ذخیره شده را دریافت کنیم. چنانچه بعد از آپلود کد، فرمان خواندن را ارسال کردیم طبق برنامه، خطای " عدم وجود فایل یا File does not Exisits" را دریافت خواهیم کرد. قطعا هنگام اولین راه اندازی فایلی وجود ندارد یا ایجاد نکرده ایم که آن را بخوانیم.

```
#include <LittleFS.h>

String command;
String text = "Hello World";

void setup()
{
  Serial.begin(115200);
 //اطلاع از آماده بودن فایل سیستم
  if(!LittleFS.begin())
  {
    Serial.println("An ERROR has occurred in LittleFS!");
  }
  else Serial.println("LittleFS is begin...");
```

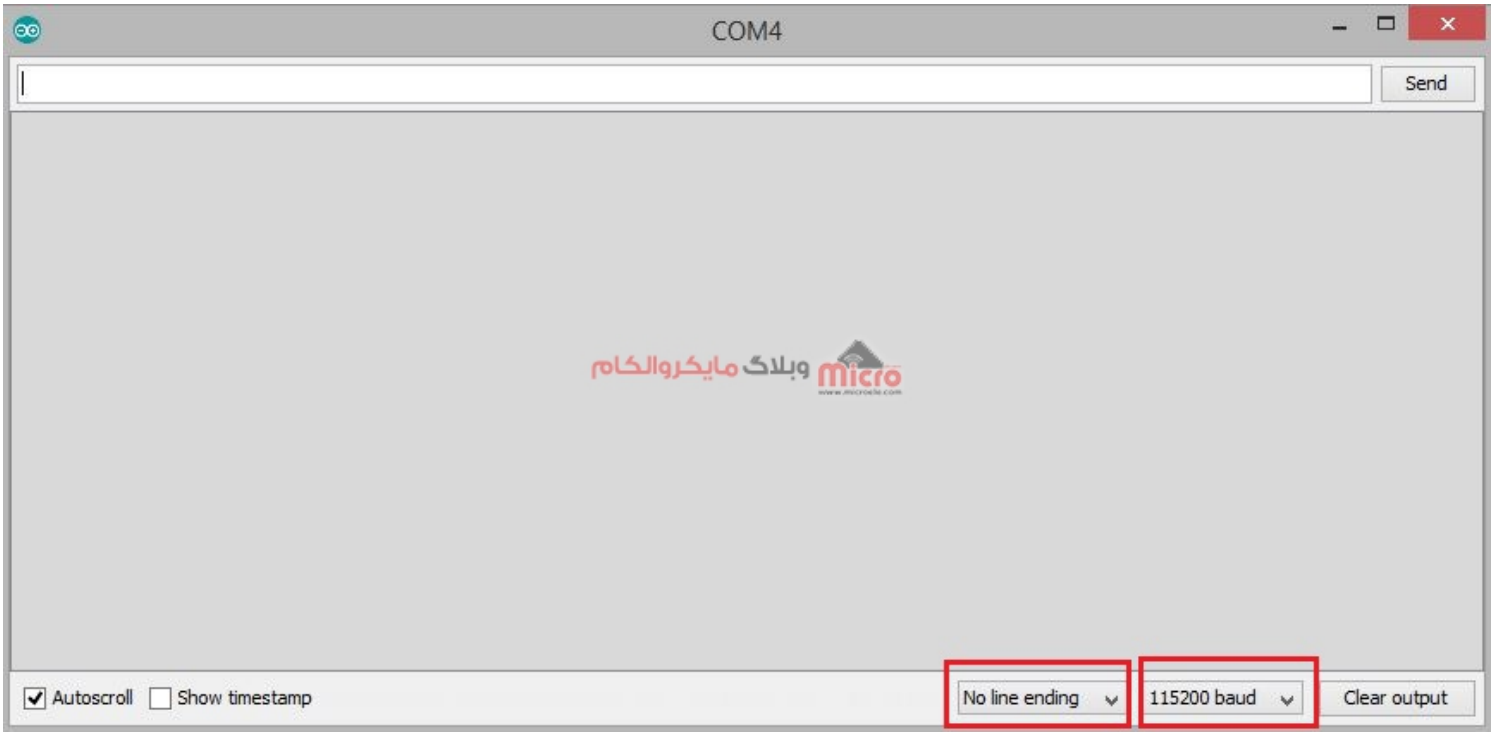
```
}

void loop()
{
 دریافت فرامین از کاربر از طریق سریال مانیتور جهت نوشتن/خواندن/حذف کردن//
 .  //در این تکه کد صرفا وضعیت نوشتن بررسی شده است
  if(Serial.available())
  {
    command = Serial.readString();
    Serial.println(command);
    if(command == "write data")
    {
   در دایرکتوری ایجاد میکنیم  txt  ابتدا یک فایل//
      File file = LittleFS.open("/FileTest.txt", "w");
   بررسی کامل شدن فرایند نوشتن//
      if (file.print(text))
      {
        Serial.println("File Written Successfully."); //پیغام موفقیت آمیز بودن
      }
      else
      {
        Serial.println("File Written Failed."); //پیغام خطا و عدم نوشتن
      }
      delay(1);
      file.close();
    }
    else if(command == "read data")
    {
   باز کردن فایل مورد نظر در دایرکتوی //
      File file = LittleFS.open("/FileTest.txt", "r");
   بررسی کردن وجود فایل مورد نظر //
      if(!file)
```

```
      {
        Serial.println("File does not Exists");
      }
      else
      {
        Serial.println("File Exists");
    خواندن از روی فایل و نمایش ان//
        while(file.available())
        {
     چون دیتا بصورت رشته ذخیره شده است، خواندن آن هم باید بصورت رشته باشد//
          Serial.println(file.readString());
        }
        file.close();
      }
    }
  }
}
```

## تنظیمات سریال مانیتور

در پنجره سریال مانیتور باودریت را برروی 115200 تنظیم کرده و گزینه No Line Ending را نیز مشابه تصویر زیر انتخاب نمایید.




تنظیمات سریال مانیتور برای ذخیره دیتا *LittleFS* در *ESP8266*

## نتیجه

خواندن و نوشتن دیتا در حافظه *Flash* با استفاده از *LittleFS* در *ESP8266*

## نکته مهم

به یاد داشته باشید محتوای نوشته شده در حافظه Flash توسط LittleFS پس از هر بار پروگرام مجدد پاک می‌شود. این حالت **فقط در پروگرام کردن مجدد رخ خواهد داد**. اما در اثر ریست شدن، قطع و وصل Vcc دیتا های ذخیره شده از بین نخواهد رفت.

---

# نتیجه گیری

در این مطالب به نحوه استفاده از فایل سیستم LittleFS جهت ذخیره دیتا برروی حافظه Flash ماژول وایفای ESP8266 پرداخته شد. با استفاده از این روش برخلاف روش پیاده سازی EEPROM در ESP8266، دیگر دیتا های نوشته شده بعد از ریست یا خاموش و روشن شدن از بین نخواهد رفت.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این آموزش اون رو

در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (microelecom@) رو هم منشن کنید.