

2025-01-27

跟PlatformIO爆了!!!

这两天想尝试把配置文件等等写到8266的flash里面。被狠狠的玩弄了。

```
"env" : "nodemcu2"  
"real_flash_size" : "1M"
```

复现

ArduinoIDE

配置中选择了 4M(1M) 的配置，错误在于编译上传报错，

这个报错真的救我一命。

PlatformIO

踏马的跟哑巴一样啥都没有就好像你nn的跟平常一样。**终端不语，只是一味的SUCCESS。**

但是还是很奇怪的是，我在PlatformIO下上传之后，8266也是不语，但是偶尔，偶尔会被看门狗重启。当flash的大小设置不正确的时候，8266就沉默了。

我懂了，一定是我给它画饼告诉它有4M的flash可以用，8266就很高兴，结果发现real_size只有1M，它看透了我，于是乎沉默了。

干啥了

当我跟PlatformIO折磨了一会之后呢，这个问题挡我路了，所以得解决一下。但是它一味的success。我也不知道到底哪里出问题了。。。但是我很聪明啊，折磨了大概一天多吧，我才意识到，上ArduinoIDE试试看呢？

在上ArduinoIDE之前，我先找kimi搞了个测试程序。

```
#include <Arduino.h>  
#include "LittleFS.h"  
  
void setup() {  
  Serial.begin(115200);  
  Serial.println();  
}
```

```

// 挂载文件系统
if (!LittleFS.begin()) {
    Serial.println("LittleFS 文件系统挂载失败!");
    return;
}
Serial.println("LittleFS 文件系统挂载成功!");
// 打开或创建文件并写入数据
File file = LittleFS.open("/test.txt", "w");
if (!file) {
    Serial.println("打开/创建文件失败!");
    return;
}
Serial.println("打开/创建文件成功!");
// 写入数据
char data[] = "Hello, World!\r\n";
file.write((uint8_t *)data, strlen(data));
file.close();
Serial.println("数据写入成功!");
// 读取文件内容
file = LittleFS.open("/test.txt", "r");
if (!file) {
    Serial.println("打开文件失败!");
    return;
}
Serial.println("打开文件成功!");
Serial.println("文件内容: ");
while (file.available()) {
    Serial.write(file.read());
}
file.close();
}

void loop() {
    // 空循环
}

```

然后呢，这个就是我做测试的前提了，在ArduinoIDE中，它给我报了一个什么几M不足的什么错误，具体的我忘了。（我去刚才回去尝试复现了一下，发现它居然也能写入成功，只是会卡在挂载文件系统的一步，那我解决这个问题岂不是纯纯的运气 🤖）然后呢，我才第一次意识到可能是flash的问题。

接下来我不知道在哪个小网站看到了有人说8266模块有1M有4M的。我本来没当回事来着，看多了到后面突然想到查查吧。**蓦然回首笑话之查查吧**。然后就又找了个查realsize的程序

```

#include <Arduino.h>

```

```
#include "LittleFS.h"

void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.print("Chip ID: ");
  Serial.println(ESP.getFlashChipId());
  Serial.print("Chip Real Size: ");
  Serial.println(ESP.getFlashChipRealSize());
  Serial.print("Chip Size: ");
  Serial.println(ESP.getFlashChipSize());
  Serial.print("Chip Speed: ");
  Serial.println(ESP.getFlashChipSpeed());
  Serial.print("Chip Mode: ");
  Serial.println(ESP.getFlashChipMode());
}

void loop() {
  // Do nothing
}
```

于是乎，我发现手上的这个ESP-12S的这个模块的realSize只有1M，其中艰辛不必多说 🤔

与此同时，在ArduinoIDE上取得了重大进展。我把设置里的那个管flash的选项换成了**Mapping defined by Hardware and Sketch**（我都会背了这个选项）。您猜怎么着，他成了。那我也不知道为什么他没有1M的选项，于是乎为了在PlatformIO上能用，我就想着说试试能不能在ArduinoIDE中配置一下呢，

但是配置里只有4M的，那我得改啊，我得改成1M的，于是我开始上网一顿查。这个期间，我才知道什么是ld文件 🤔（这里少说花了我两个小时）

然后我就一顿翻配置文件，我本来想自己写个ld嘛，且看了一会就放弃了，我怕搞出事来。

这个ld库库复杂，我哪里知道这个flash到第几位啊给我整懵逼了。

其中经过一系列波折，给我找到了 `nodemcu2.json` 这个配置文件。自从我发现是flash问题之后，我就发现了其实它编译的时候给了 `flash_size = 4M`；但也仅此而已了。

这个配置文件不简单啊，一打开就能看到一行

```
"ldscript": "eagle.flash.4m1m.ld"
```

这不坏事了么，我有4M么你就给我4M1M。。。

于是又是一顿找，找到了ldscript里面的文件，于是乎就翻到了1M512.ld这个救星文件。

于是又进入 `platformio.ini` 里面，狠狠加入一行

```
board_build.ldscript = eagle.flash.1m512.ld
```

在这之前我还照着各种找到的教程前面还加了

```
board_upload.flash_size = 1MB  
board_build.spiffs_size = 512K
```

不知到有没有影响了已经，反正现在正常了，就先不动了。

所以

虽然说PlatformIO给我干懵了，但是我还是会接着用PlatformIO的 🤖。