

How to write with ESP32 partition table specified

How to apply arbitrary partition table in writing to ESP32 mounted on TWELITE SPOT

We will show you how to apply an arbitrary partition table when writing sketches and files to the ESP32 installed in TWELITE SPOT.

i This article introduces applied content (how to specify the partition table of the flash area).

If you are using the partition table settings that come with the ESP32 Arduino Core (e.g. Default 4MB with spiffs), you can ignore this article.

! We do not take any responsibility for any damage caused by the settings in this article.

Creating a definition file

The partition table definition is written in a csv file.

The example below specifies that 8MB of the 16MB flash space is to be used by the file system.

```
# TWELITE SPOT 16MB with 8MB LittleFS
# Name, Type, SubType, Offset, Size, Flags
nvs, data, nvs, 0x9000, 0x5000,
otadata, data, ota, 0xE000, 0x2000,
app0, app, ota_0, 0x10000, 0x7F0000,
spiffs, data, spiffs, 0x800000, 0x800000,
```

- `TWELITE SPOT 16MB with 8MB LittleFS` This is the name displayed in the Arduino IDE.
- `nvs` Area used by the system. Do not change.
- `otadata` This is the area used when using OTA. Do not change.
- `app0` This is the area where the firmware is written.
- `spiffs` The space used by the LittleFS file system.

`Offset` The units of and columns in the csv file `Size` are bytes and are hexadecimal.

したがって、上記の例では、ファームウェアとファイルシステムが使えるサイズは下記のように計算できます。

- `app0` のサイズ： `0x7F0000` = 8323072 より、 7.875MB
- `spiffs` のサイズ： `0x800000` = 8388608 より、 8MB

定義ファイルの登録

[Arduino15フォルダ](#)を開き、下記のパスに csv ファイルを追加します。


```
Arduino15/packages/esp32/hardware/esp32/x.x.x/tools/partitions
```

x.x.x は Arduino core for the ESP32 のバージョン

パーティションテーブルの適用

Arduino IDE のツールバーから ツール -> Partition Scheme を開き、追加したパーティションテーブルを選びます。

選択したパーティションテーブルが次回以降のファームウェアの書き込みやファイルシステムの書き込みに反映されます。

 スケッチファイルと同じ場所に `partitions.csv` という名前で設定ファイルを置くと、そちらが優先されます。ただし、Arduino IDE 上の表記は変わらないため、紛らわしい場合があります。