

# ESP32 のパーティションテーブルを指定した書き込み方法

TWELITE SPOT に搭載された ESP32 に対する書き込みにおいて、任意のパーティションテーブルを適用する方法

TWELITE SPOT に搭載された ESP32 ヘスケッチやファイルを書き込む際に、任意のパーティションテーブルを適用する方法をご案内します。

**i** 本稿では、応用的な内容（フラッシュ領域のパーティションテーブルを指定する方法）を紹介しております。

ESP32 Arduino Core に最初から含まれているパーティションテーブルの設定（例：Default 4MB with spiffs）を使用される場合は、本稿を無視していただいて構いません。

**!** 本稿の設定を行ったことによるいかなる損害についても、弊社は一切の責任を負いません。

## 定義ファイルの作成

パーティションテーブルの定義は、csv ファイルに記述します。

下記の例では、16MB のフラッシュ領域のうち、8MB をファイルシステムで使うように指定しています。

```
# 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 Arduino IDE に表示される名前です。
- nvs システムで使用する領域です。変更しません。
- otadata OTA を使う際に使用する領域です。変更しません。
- app0 ファームウェアを書き込む領域です。
- spiffs LittleFS ファイルシステムで使用する領域です。

csv ファイル中の Offset および Size 列の単位はバイトで、16進数です。

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。