Sauvegarde de données sur carte microSD avec ESP32

Houssem-eddine LAHMER

5 mai 2025

Plan du cours

Pourquoi enregistrer sur microSD ?

- Stockage local de grandes quantités de données (capteurs, logging)
- Indépendance réseau / cloud
- Facilité de récupération et de transfert des données

La carte ESP32

- Microcontrôleur dual-core 32 bits
- Interfaces SPI, SDMMC, I2C, UART...
- Alimentation 3.3 V
- Bibliothèque Arduino Core disponible

La carte microSD

- Formats : microSD, microSDHC, microSDXC
- Système de fichiers : FAT16 / FAT32
- Tension 3.3 V (attention au niveau logique)
- Carte adaptateur break-out pour prototypage

Schéma de câblage

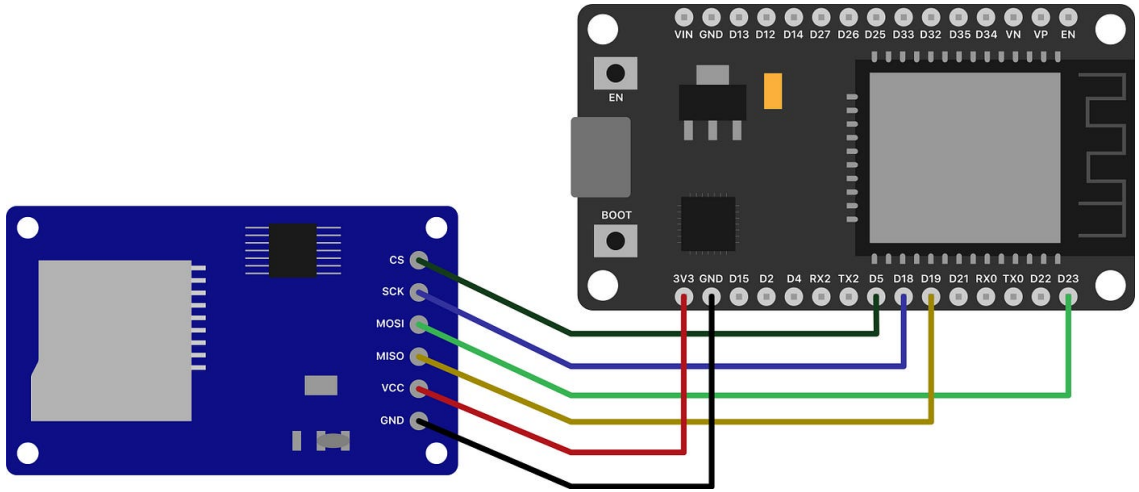

Figure – Connexion SPI ESP32 <–>microSD

Broches SPI par défaut :

- MOSI : GPIO 23
- MISO : GPIO 19
- SCK : GPIO 18
- CS : GPIO 5 (ou autre broche libre)

Bibliothèque Arduino SD

```
1 #include <SPI.h>
2 #include <SD.h>
```

- SD.begin(chipSelect) : initialisation de la carte
- SD.exists(path), SD.open(path, mode)

Paramétrage de l'ESP32

```
1  const int chipSelect = 5;
2
3  void setup() {
4    Serial.begin(115200);
5    if (!SD.begin(chipSelect)) {
6      Serial.println(" chec  initialisation SD");
7      return;
8    }
9    Serial.println("Carte SD pr te.");
10 }
```

Écriture de données

```
1  void loop() {
2    File dataFile = SD.open("datalog.txt", FILE_WRITE);
3    if (dataFile) {
4      dataFile.println("Nouvelle ligne de donn es");
5      dataFile.close();
6      Serial.println("Donn e   crite  .");
7    } else {
8      Serial.println("Erreur ouverture fichier");
9    }
10   delay(1000);
11 }
```

Lecture de données

```
1  void lireFichier() {
2    File dataFile = SD.open("datalog.txt");
3    if (dataFile) {
4      while (dataFile.available()) {
5        Serial.write(dataFile.read());
6      }
7      dataFile.close();
8    } else {
9      Serial.println("Impossible d'ouvrir le fichier");
10   }
11 }
```

Optimisations et fiabilité

- Éviter d'ouvrir/fermer trop fréquemment : tamponnez les écritures
- Vérifier l'espace disponible avant écriture
- Gérer les erreurs et contrôles CRC
- Utiliser un buffer circulaire en RAM pour logging intensif

Conclusion

- La carte microSD est une solution simple pour du stockage local
- L'ESP32 offre plusieurs interfaces pour la piloter
- Les bibliothèques Arduino simplifient grandement le code
- Adapter le code selon l'application et la fréquence d'écriture

Questions ?