LABORATOIRE – carte microSD

Questionnement

- Comment enregistrer des données sur une mémoire amovible d'un dispositif?
- Quel est le format de données à considérer?
- Comment programme un système à microcontrôleur pour réaliser l'enregistrement de données sur une carte microSD?

Matériel requis :

- Plaquette protoTPhys
- Module de carte micro SD
- Carte micro SD
- Fils de prototypage
- Outils

Théorie et références :

- https://randomnerdtutorials.com/esp32-microsd-card-arduino/
- https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/
- Notes de cours : 244-470-AL_SD-format_H25.pdf

Mise en situation :

On vous demande de joindre à votre appareil la capacité d'enregistrer des données de mesure d'un capteur ainsi que l'état d'un dispositif. Le choix le plus simple pourrait être celui de la carte microSD, abordable, facile à intégrer, versatile et de grande capacité.

Le format et l'organisation des données à écrire sur la carte microSD demeurent plutôt flexibles, étant réalisé de façon logicielle.

Pour ce laboratoire, vous utiliserez le format CSV : Comma-Separated Values. Les données de ce format sont écrites par enregistrement. Un enregistrement est une suite de données, séparées par une virgule. Il y a 1 enregistrement pas ligne. Un ligne se terminant par un code «CR/LF » (Carriage Return, LineFeed, soit via l'instruction « println() »).

Ce format requiert toutefois d'écrire 1 seule fois, en entête de fichier, la désignation de chaque champ de donnée.

Manipulation 1 – initiation

Préparation

Suivre la procédure donnée par RNT. L'important est le branchement du module, comme suit :

MicroSD card module	ESP32
~~3V3~~ V_{IN}	~~3.3V~~ V_{IN}
CS	GPIO 5
MOSI	GPIO 23
CLK	GPIO 18
MISO	GPIO 19
GND	GND

Programmation

Utiliser le code exemple fourni par RNT. Puisque nous travaillons avec le ESP32, il n'y a pas de librairie à installer dans votre environnement Arduino car les librairies nécessaires sont fournies implicitement par le package ESP32 (soit arduino-esp32 : https://github.com/espressif/arduino-esp32/tree/master/libraries/SD).

Réalisation

Faire la démonstration du code exemple, qui passe en revue diverses fonctionnalités de traitement avec la carte microSD :

- List a directory;
- Create a directory;
- Remove a directory;
- Read a file content;
- Write content to a file;
- Append content to file;
- Rename a file;
- Delete a file;
- Initialize microSD card;
- Get microSD card type;
- Get microSD card size;

Prenez un moment pour comprendre les divers essais que réalise cet exemple. Il est fort probable que vous ayez à réutiliser des bouts de ce code dans un futur laboratoire.

Manipulation 2 – Écrire un document format CSV

Pour cette seconde manipulation, il sera requis de réutiliser quelques routines précédentes afin de créer une simple application permettant d'écrire des données sur la carte microSD, prélevées à intervalle régulière.

Votre code doit :

Initialiser et valider la carte microSD correctement

Initialiser le fichier de données :

- Si le fichier n'existe pas, il faut le créer ET inscrire l'entête (mode Write)
- Si le fichier existe déjà, il n'y a rien à faire de plus

Initialiser un compteur d'enregistrements

À chaque fois qu'un enregistrement est réalisé sur la carte SD, cette variable est incrémentée.

À intervalle régulière (réglé par une minuterie) :

- Collecter les données
- Forger un enregistrement en concaténant les champs, séparés par une virgule
  Ex : numéro d'enregistrement, millis()/1000, donnée 1, donnée 2 …
- Écrire l'enregistrement au fichier (mode Append)

Exemple/suggestion de données à collecter:

- Lecture d'un potentiomètre,
- Lecture d'un capteur : DHT11,
- Lecture d'un nombre aléatoire (https://docs.arduino.cc/language-reference/en/functions/random-numbers/random/)

Après quelques cycles de fonctionnement, retirez la carte microSD du module et utiliser le lecteur/adaptateur approprié à insérer à l'ordinateur. Ouvrir le fichier dans un éditeur de texte (NotePad++, blocNote, à votre convenance).

Voici un exemple de ce que vous pourriez obtenir :

```
numero,timestamp,systemID,temperature,humidite,random
1,10000,sysID_007,21.4,45.50,10
2,20000,sysID_007,20.8,45.56,22
3,30000,sysID_007,22.1,45.57,15
[…]
```

Annexe – compteur de ligne

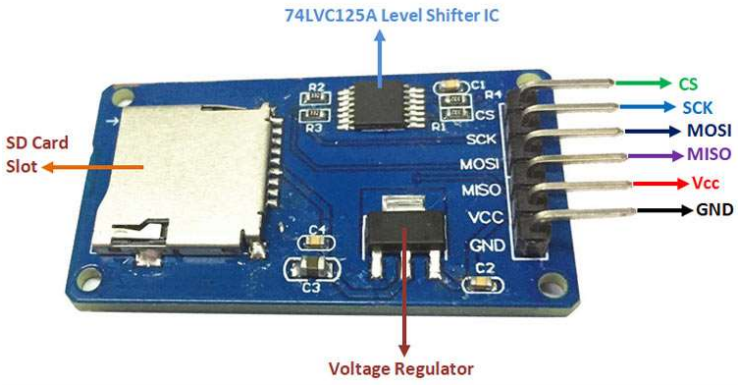
Il peut être pratique d'avoir une fonction permettant de compter le nombre lignes dans un fichier de données CSV. Ainsi il est possible de connaître le nombre de ligne de données. Il est aussi possible de poursuivre la séquence de numérotation des lignes. Voici une petite fonction permettant d'implémenter cette fonctionnalité à vos programmes :

```
// Compteur de ligne dans le fichier de donnees
uint16_t lireNbligne(File datFile) {

  uint16_t linecount = 0;
  char inputChar;

  if (datFile) {
    while (datFile.available()) {
      inputChar = datFile.read();
      if(inputChar == '\n') linecount++;
    }
  }
  return linecount;
}
```

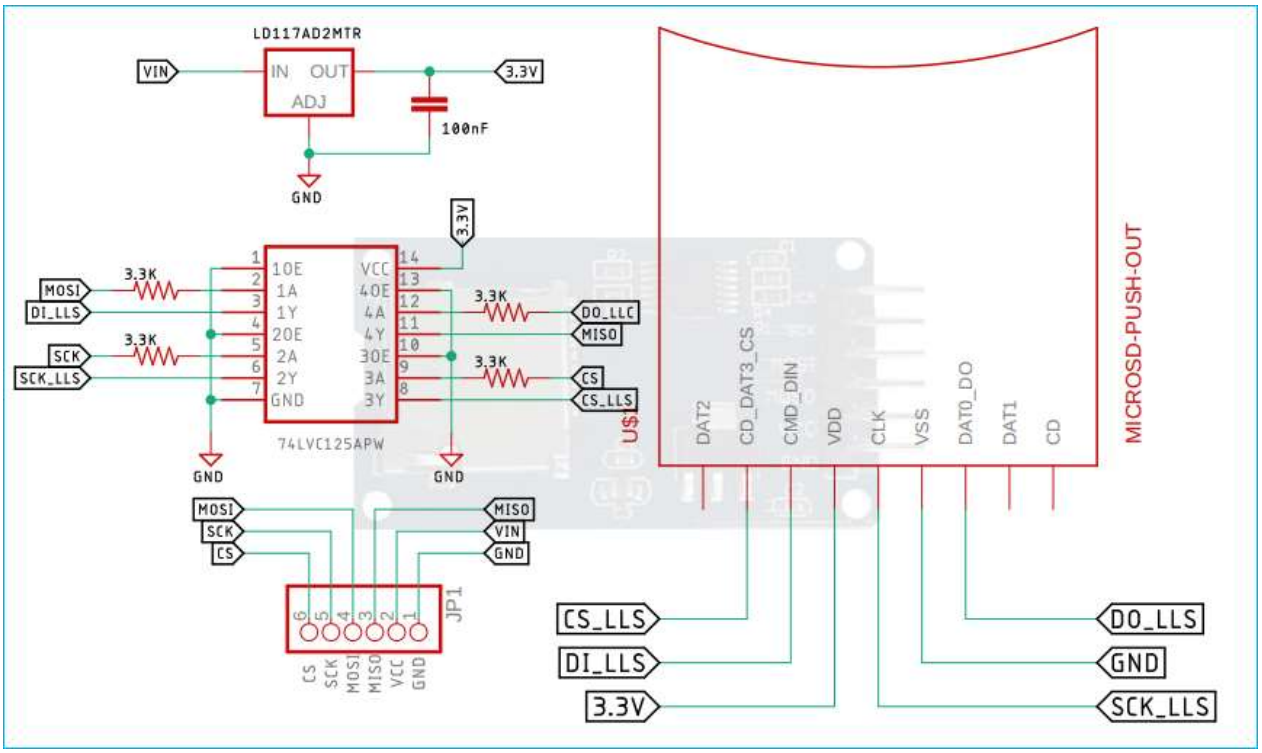
Ce programme requiert en entrée un fichier déjà ouvert par l'appel `SD.open(__);`

Annexe – divers modules

Attention à ce que vous trouverez sur les sites de vente de pièces. Les modules SD ne se valent pas tous. Certains on un design minimal, connectant directement à la carte SD. Cette dernière ne tolère pas une alimentation supérieure à 3.3V. Plusieurs (comme celles du laboratoire) intègrent un régulateur : une tension d'alimentation d'entrée pouvant aller entre 5V à 16V et qui sort à 3.3V.



Et son schématique :



Source : https://circuitdigest.com/microcontroller-projects/interfacing-micro-sd-card-module-with-arduino