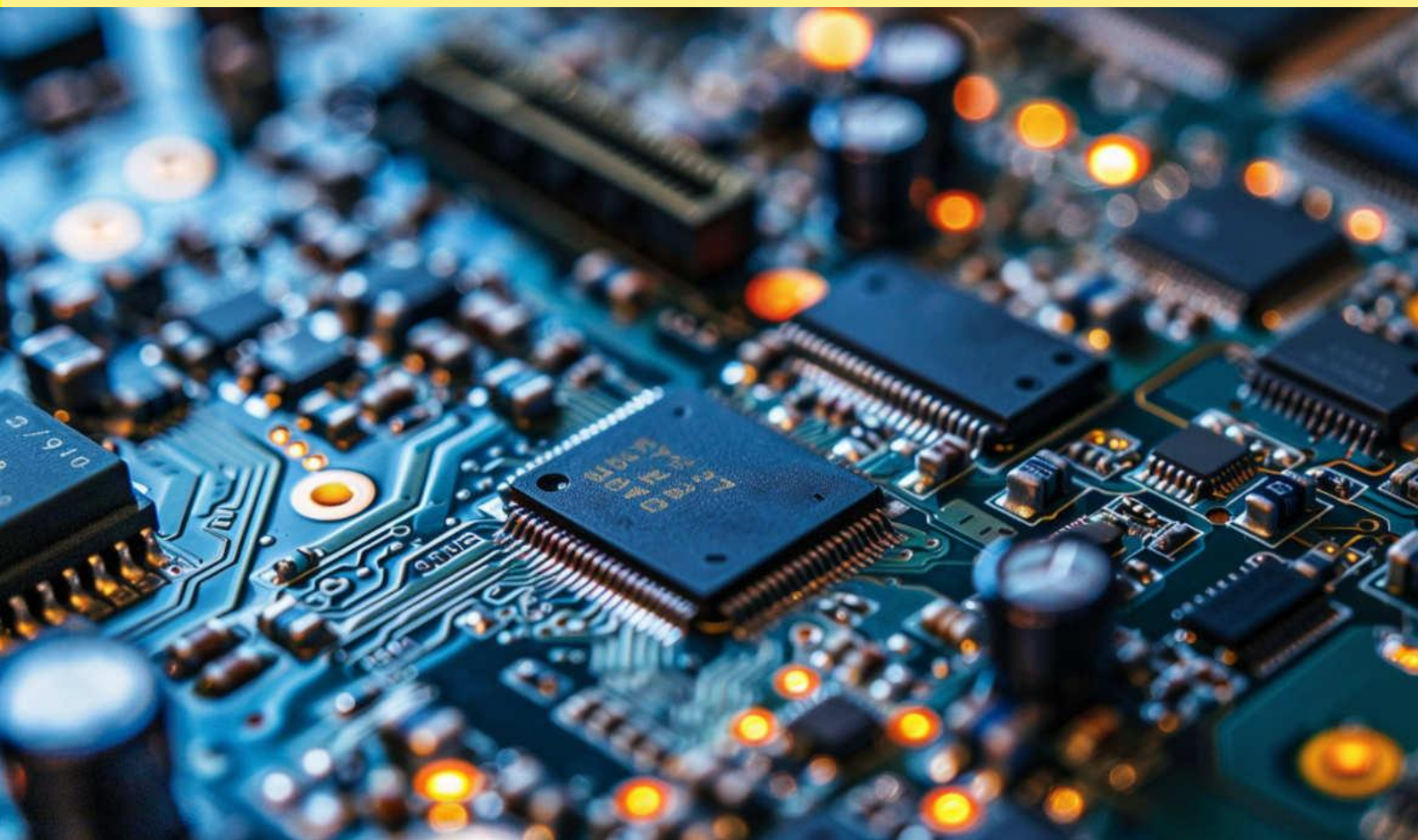


CPI A2 – PROJET SYSTEMES EMBARQUÉS

PROSIT 7 – ON ENCHAINE LES FICHIERS



Contexte :

Des difficultés sont apparues lors de la phase finale, qui consiste à sauvegarder les données sur une carte SD de 2 Go. Le format FAT32 choisi pour le système de fichiers crée des limitations en termes de gestion des fichiers et d'espace de stockage. L'équipe explore donc des alternatives, dont un système de fichiers plus léger, mais cela nécessite des adaptations complexes pour l'Arduino.

Mots inconnus :

Fat32 /16 système de gestion de fichier

Problématique :

Comment adapter la gestion des fichiers pour envoyer le plus rapidement possible les fichiers dans la mémoire ?

Plan d'action :

1. Espace disque et unités d'allocation 90 Min]
2. Systèmes de fichiers : généralités
3. Structure d'un système de fichiers
4. Les listes chaînées (+ WS)

Rappels sur la complexité 60 Min]

Calculer la complexité d'un programme

- a. En espace
- b. En temps
- c. Autre ...
5. Étudier FAT et [30Min]
6. Proposer l'organisation la plus pertinente [30 Min]

Réalisation :

Premièrement on déclare la liste chaînée

```
1  #include <SPI.h>
2  #include <SD.h>
3
4  const int chipSelect = 4; // Pin pour la carte SD
5
6  // Structure pour un élément de la liste chaînée
7  struct Node {
8      int data;
9      Node* next;
10 };
11
12 // Déclaration de la tête de la liste
13 Node* head = NULL;
14 File dataFile;
15
16 void setup() {
17     Serial.begin(9600);
18 }
```

Après cela on initialise la carte sd puis on crée le fichier où les données seront stockées

```
16 void setup() {
17     Serial.begin(9600);
18
19     // Initialisation de la carte SD
20     if (!SD.begin(chipSelect)) {
21         Serial.println("Erreur : carte SD non détectée.");
22         while (1); // Arrêt si la carte SD n'est pas présente
23     }
24     Serial.println("Carte SD initialisée.");
25
26     // Création du fichier
27     dataFile = SD.open("LISTE.TXT", FILE_WRITE);
28
29     if (!dataFile) {
30         Serial.println("Erreur : impossible de créer le fichier.");
31         while (1);
32     }
33
34     // Création et initialisation de la liste chaînée avec des valeurs forcées
35     int valeurs[5] = {10, 20, 30, 40, 50}; // Valeurs forcées
36     createLinkedList(valeurs, 5);
37
38     // Parcourir la liste et écrire les éléments dans le fichier SD
39     storeListInSD();
40
41     // Fermeture du fichier
42     dataFile.close();
43     Serial.println("Liste enregistrée avec succès sur la carte SD.");
```

On crée la liste chaînée avec les valeurs forcées, et on met en place les allocations de la mémoire qu'on va utiliser

```
49
50 // Fonction pour créer une liste chaînée avec des valeurs spécifiques
51 void createLinkedList(int* valeurs, int n) {
52     Node* temp;
53     for (int i = 0; i < n; i++) {
54         // Allocation dynamique de mémoire pour un nouveau nœud
55         Node* newNode = new Node;
56         newNode->data = valeurs[i]; // Initialisation avec les valeurs données
57         newNode->next = NULL;
58
59         if (head == NULL) {
60             head = newNode; // Le premier nœud devient la tête de la liste
61         } else {
62             temp = head;
63             // Parcourir la liste jusqu'à la fin pour ajouter le nœud
64             while (temp->next != NULL) {
65                 temp = temp->next;
66             }
67             temp->next = newNode;
68         }
69     }
70 }
```

Et finalement, on crée la fonction qui va nous permettre de stocker les données dans la carte sd

```
71
72 // Fonction pour stocker la liste chaînée sur la carte SD
73 void storeListInSD() {
74     Node* temp = head;
75
76     while (temp != NULL) {
77         // Écriture de chaque élément de la liste dans le fichier
78         dataFile.print("Element: ");
79         dataFile.println(temp->data);
80
81         // Affichage des éléments dans le moniteur série
82         Serial.print("Element: ");
83         Serial.println(temp->data);
84
85         temp = temp->next;
86     }
87
88     // Forcer l'écriture dans le fichier
89     dataFile.flush();
90 }
91 |
```