

Série d'exercices 3 : Programmation structurée à haut niveau

Exercice 1 : Machine à états finis (FSM) – Feu tricolore

Objectif : Implémenter une FSM sans **delay()**.

Créer un feu de signalisation pour deux voies (Voie A et Voie B) avec les états :

- État 1 : A vert, B rouge (10s)
- État 2 : A orange, B rouge (3s)
- État 3 : A rouge, B vert (10s)
- État 4 : A rouge, B orange (3s)

Utiliser un **enum** pour les états et une fonction de transition appelée périodiquement (timer).

Exercice 2 : Système multi-tâches avec pointeurs sur fonctions

Objectif : Simuler un scheduler coopératif.

Définir un tableau de pointeurs sur fonctions pour 3 tâches :

- Tâche 1 : Lire un capteur
- Tâche 2 : Afficher sur LCD
- Tâche 3 : Gérer les boutons

Chaque tâche s'exécute à une fréquence différente (ex : 10 Hz, 2 Hz, 5 Hz). Implémenter un scheduler simple basé sur **millis()**.

Exercice 3 : Communication Bluetooth HC-05 avec MIT App Inventor

Objectif : Créer une application Android pour contrôler une carte Nano (Atmega328P).

Sur Arduino :

- Recevoir des commandes Bluetooth (ex : 'L' = LED ON, 'F' = LED OFF)

- Envoyer la température lue sur un capteur LM35

Sur MIT App Inventor :

- Interface avec 2 boutons et un afficheur
- Envoyer/recevoir via Bluetooth

Exercice 4 : Bus CAN – Réseau de deux μ C

Objectif : Échanger des données entre deux cartes via CAN.

Avec deux Arduino Uno + shields MCP2515 :

- Nœud A : envoie périodiquement une valeur de potentiomètre (ID = 0x100)
- Nœud B : reçoit et affiche la valeur sur LCD, et contrôle un servo selon la valeur
- Implémenter la détection de perte de communication (timeout)

Exercice 5 : Bus de terrain Modbus TCP/IP – S7-1200 avec Arduino UNO

Objectif : Faire communiquer une API Siemens S7-1200 et une Arduino.

- Arduino UNO + shield Ethernet comme esclave Modbus TCP
- Registres définis :
 - **HR0 (40001)** : commande de LED (0/1)
 - **HR1 (40002)** : lecture d'un capteur analogique
- Le S7-1200 lit/écrit les registres via TIA Portal
- Afficher l'état sur LCD de l'Arduino

Exercice 6 : Stockage sur carte SD via SPI

Objectif : Enregistrer des données sur carte SD et les visualiser.

- Lire un capteur (température ou luminosité) toutes les secondes

- Créer un fichier data.csv sur carte SD
- Format : timestamp (ms), valeur
- Ajouter un bouton pour arrêter/lancer l'enregistrement
- Lire le fichier et l'envoyer via UART sur demande