

Série d'exercices 1 : Initiation de base du microcontrôleur

Exercice 1 : Manipulation de bits et macros

Objectif : Maîtriser le **set**, **clear**, **toggle** et **test de bits**.

Écrire les macros suivantes en C pour un μ C AVR32/PIC32 :

- **SET_BIT**(reg, bit)
- **CLR_BIT**(reg, bit)
- **TOG_BIT**(reg, bit)
- **CHK_BIT**(reg, bit)

Application : Écrire un programme qui allume une LED si le bit 3 du PORTB est à 1, l'éteint sinon, puis alterne l'état du bit 2 toutes les 500 ms.

Exercice 2 : Architecture mémoire et accès aux registres

Objectif : Comprendre Flash, RAM, EEPROM, registres généraux.

1. Dessiner l'architecture mémoire d'un AVR (ATmega328P) en identifiant chaque section.
2. Écrire un programme qui :
 - Stocke une constante **MAGIC_NUMBER = 0xAB** dans la Flash (section Boot)
 - Copie cette valeur en RAM
 - Sauvegarde une variable utilisateur en EEPROM
 - Lit l'EEPROM au démarrage et affiche la dernière valeur via la liaison série

Exercice 3 : Structure d'un programme Arduino embarqué

Objectif : Maîtriser l'ossature **setup()** / **loop()** et les sections.

Créer un programme avec :

- Une section déclarative contenant un **enum** des états système (**IDLE**, **ACTIVE**, **ERROR**)
- Une initialisation des broches
- Une boucle principale qui alterne entre les états
- Une fonction **afficherEtat()** utilisant **switch-case**

Ajouter un **typedef struct** pour regrouper les paramètres d'un capteur (type, valeur, unité).

Exercice 4 : Gestion d'un menu déroulant avec 3 boutons (Up, Down, OK)

Objectif : Appliquer **if-else**, **switch-case**, et la logique de menu.

Réaliser un menu à 4 entrées :

- Allumer LED1
- Allumer LED2
- Clignotement rapide
- Clignotement lent

- **Up / Down** : navigation
- **OK** : exécution de l'action
- Gérer les rebonds logiciels des boutons

Exercice 5 : Gestion des sorties numériques – **delay()** vs **millis()**

Objectif : Comparer le bloquant (**delay**) et non-bloquant (**millis**).

1. Version 1 : Faire clignoter 3 LEDs avec des périodes différentes (200ms, 500ms, 1s) en utilisant **delay()**.
2. Version 2 : Refaire avec **millis()** sans bloquer l'exécution.
3. Ajouter un bouton qui change la vitesse du clignotement en temps réel.

Exercice 6 : Affichage sur TM1637 et LCD 2x16

Objectif : Piloter des afficheurs externes.

- Afficher un compteur de 0 à 9999 sur le TM1637 (incrément toutes les secondes)
- Sur le LCD 2x16 :
 - Ligne 1 : "Compteur : XXXX"
 - Ligne 2 : "Temps (s) : YY"
- Utiliser un bouton pour remettre le compteur à zéro
- Gérer le rafraîchissement sans **delay()**