



EPREUVE PRATIQUE – PROJET 01

Equipe projet : Binôme (02 étudiants/équipe)

1. Contexte industriel

Une usine dispose d'une installation automatisée composée de :

- machines industrielles (convoyeurs, moteurs, fours, presses…)
- capteurs (température, pression, vitesse, niveau…)
- actionneurs (moteurs, vannes, alarmes)
- un service de maintenance chargé de :
 - surveiller le fonctionnement
 - détecter les anomalies
 - enregistrer les événements
 - planifier les actions de maintenance

Vous devez développer un outil Python de supervision industrielle simulée, utilisable en ligne de commande, structuré comme un mini-SCADA / GMAO simplifié.

2. Objectifs pédagogiques du projet

Le projet doit permettre aux étudiants de démontrer leur capacité à :

- concevoir un programme Python structuré et modulaire
- manipuler correctement :
 - variables
 - conditions
 - boucles
 - listes, tuples, dictionnaires
- écrire des fonctions métier industrielles
- organiser le code en modules
- gérer les fichiers de données et de logs
- implémenter une logique d'automatisme et de diagnostic
- travailler efficacement en binôme
- produire un outil exploitable et documenté

3. Architecture du projet

industrial_supervisor/

```
|  
|   main.py          # menu principal  
|   machines.py      # gestion des machines  
|   sensors.py       # gestion des capteurs  
|   supervision.py   # logique de supervision  
|   diagnostics.py   # détection d'anomalies  
|   maintenance.py   # états & maintenance  
|   logger.py        # journalisation  
|   reports.py       # rapports industriels  
|  
|   data/  
|       machines.txt  
|       sensors.txt  
|       logs.txt  
|       maintenance.txt  
|  
└── README.md
```

4. Menus

Menu principal :

===== INDUSTRIAL MONITOR =====

- [1] : Gestion des machines
 - [2] : Gestion des capteurs
 - [3] : Supervision de l'installation
 - [4] : Diagnostic & alarmes
 - [5] : Maintenance & états
 - [6] : Journal des événements
 - [7] : Rapports industriels
 - [0] : Quitter
- =====

Votre choix :

Menu 1 : Gestion des machines

- Gestion des machines ---
- [1] : Ajouter une machine
- [2] : Supprimer une machine
- [3] : Modifier l'état d'une machine
- [4] : Afficher les machines
- [5] : Charger les machines
- [0] : Retour menu principal

Menu 2 : Gestion des capteurs

- Gestion des capteurs ---
- [1] : Ajouter un capteur
- [2] : Supprimer un capteur
- [3] : Afficher les capteurs
- [4] : Sauvegarder les capteurs
- [5] : Charger les capteurs
- [0] : Retour menu principal

Menu 3 : Supervision de l'installation	Menu 4 : Diagnostic & Alarmes
<p>--- <i>Supervision</i> ---</p> <p>[1] : Saisir les valeurs des capteurs [2] : Vérifier les seuils [3] : Simuler un cycle de production [4] : Afficher l'état global [0] : Retour menu principal</p>	<p>--- <i>Diagnostic & Alarmes</i> ---</p> <p>[1] : Lancer le diagnostic des machines [2] : Déetecter les anomalies des capteurs [3] : Déclencher les alarmes [4] : Mettre à jour l'état des machines [5] : Afficher les anomalies détectées [0] : Retour menu principal</p>
Menu 5 : Maintenance & états	Menu 6 : Journal des événements
<p>--- <i>Maintenance</i> ---</p> <p>[1] : Mettre une machine en maintenance [2] : Sortir une machine de maintenance [3] : Afficher l'état de maintenance [0] : Retour menu principal</p>	<p>--- <i>Journal</i> ---</p> <p>[1] : Afficher les événements [2] : Sauvegarder le journal [3] : Charger le journal [0] : Retour menu principal</p>
Menu 7 : Rapports industriels	
<p><i>Rapport attendu :</i></p> <p>===== RAPPORT INDUSTRIEL =====</p> <p><i>Machines totales : X</i></p> <p><i>En service : X</i></p> <p><i>En panne : X</i></p> <p><i>En maintenance : X</i></p> <p><i>Alertes détectées : X</i></p> <p><i>Dernier incident : XXX</i></p> <p>=====</p>	

5. Contraintes techniques

- Variables et types de base
- Conditions (*if / elif / else*) et Boucles (*for, while*)
- Utilisation de fonctions dans chaque module
- Utilisation de *try / except*
- Listes, tuples et dictionnaires
- Lecture et écriture de fichiers
- Menus interactifs
- Code clair, commenté, structuré