TP : Système d'Alerte Connecté avec Raspberry Pi

Prérequis et Remédiation

Acquis de la séquence 2 (requis pour ce TP)

- **Installation et configuration Raspberry Pi** (séance 2.2)
- **Premiers pas Linux** (séance 2.3)
  - Commandes shell de base (navigation, fichiers, droits)
  - Utilisation des éditeurs (nano, geany)
  - Exécution de programmes Python
- **Services réseau sur RPi** (séance 2.4)
  - Installation et configuration Flask

Acquis des séquences précédentes

- **Séquence 1 - Bases de données**
  - Concepts de client-serveur
  - Manipulation de données structurées
- **Programmation Python** (BTS 1ère année)
  - Variables, boucles, conditions, fonctions

Savoirs Liés

- Adressage IP et configuration réseau
- Architecture des systèmes embarqués (ARM vs x86)
- Audit et surveillance des systèmes

Compétences visées

Code	Compétence	Niveau visé
C11	Maintenir un réseau informatique	2

Activités et Tâches

Activité	Code Tâche	Intitulé de la tâche	Mise en œuvre dans le TP
R3 - Exploitation et maintien en condition opérationnelle	R3-T3	Supervision de l'état du réseau dans son périmètre	Analyse des dysfonctionnements et codes d'erreur

Activité	Code Tâche	Intitulé de la tâche	Mise en œuvre dans le TP
R5 - Maintenance des réseaux informatiques	R5-T4	Réalisation de diagnostics et d'interventions de maintenance curative	Surveillance du fonctionnement du système et logs

Objectifs

- Envoyer une alerte simulée via HTTP POST.
- Observer l'interaction entre système embarqué et serveur réseau.
- Déployer un serveur Flask recevant et affichant des alertes.

Matériel nécessaire :

- Raspberry Pi (un par binôme, OS à jour)
- 1 bouton (ou même clavier → simulation via `input()`)
- Accès réseau (filaire ou Wi-Fi)
- Python + dépendances :

```
pip3 install requests flask
```

Contexte :

> "Une petite entreprise souhaite équiper ses bureaux d'un **système de bouton d'urgence connecté**. En cas d'incident (incendie, intrusion...), l'utilisateur appuie sur un bouton physique et **envoie automatiquement un message d'alerte à un serveur distant (poste de sécurité)**."

Le système doit être **léger**, **réactif**, et pouvoir être **déployé facilement** sur d'autres Raspberry Pi.

Séance 1 (2h) – Déploiement et première alerte

Objectifs

- Prendre en main le projet
- Configurer le Raspberry Pi et le projet Python
- Déclencher une première alerte via le bouton simulé

Étape 1 – Préparation du projet

1. Créer un dossier de projet :

```
mkdir systeme_alerte
cd systeme_alerte
```

2. Créer et activer un environnement virtuel :

```
python3 -m venv venv
source venv/bin/activate
```

3. Installer les dépendances :

```
pip install flask requests
```

4. Créer les fichiers suivants (vides pour l'instant) :
    - `MockGPIO.py` (simulation du bouton).
    - `config.py` (informations de base).
    - `systeme_alerte.py` (programme du Raspberry Pi).
    - `serveur_alertes.py` (programme serveur Flask).

Étape 2 – Configuration initiale

Compléter `config.py` avec :

- Identifiant (`APPAREIL_ID`)
- Localisation et couleur associée
- Type d'alerte
- ...

> Vérifier : cohérence entre identifiant, couleur et localisation.

Étape 3 – Serveur d'alertes

1. Lancer le serveur :

```
python serveur_alertes.py
```

2. Ouvrir dans un navigateur :

```
http://localhost:5000/alertes
```

3. Tester avec `curl` (optionnel) :

```
curl -X POST http://localhost:5000/alertes \
  -H "Content-Type: application/json" \
```

```
  -d '{"message": "Test manuel", "source": "poste1"}'
```

Étape 4 – Déclenchement d'une alerte

```
python systeme_alerte.py
```

→ Appuyer sur **Entrée** pour simuler le bouton.

Question : que se passe-t-il si le serveur est arrêté ?

Évaluation du TP : Critères de maîtrise (grille pour l'enseignant)

A. Critères d'évaluation pratique

Critère	Atteint	En cours	Non atteint
Environnement virtuel et dépendances correctement installés	☐	☐	☐
Structure des dossiers et fichiers respectée	☐	☐	☐
Variables de configuration dans `config.py` complètes et cohérentes	☐	☐	☐
Serveur Flask démarré sans erreur	☐	☐	☐
Navigation sur `/alertes` affiche la liste des alertes reçues	☐	☐	☐
Envoi d'alerte réussi et confirmé par réponse HTTP 200	☐	☐	☐
Simulation bouton via `input()` déclenche bien l'envoi d'alerte	☐	☐	☐