



Langages de Programmation et Logiciels par Composant du Système de Surveillance

Votre système se compose de deux parties principales : les **Nœuds Capteurs** (les "mini-stations") et le **Raspberry Pi Central** (le "cerveau").

Les Nœuds Capteurs (basés sur ESP32)

Chaque nœud est une unité autonome qui lit les capteurs et envoie les données.

► Langage de Programmation :

- **Arduino (C++)** : C'est le langage le plus couramment utilisé pour programmer les microcontrôleurs ESP32. L'IDE Arduino est simple d'utilisation, et il existe de très nombreuses bibliothèques pour les capteurs et la communication Wi-Fi/MQTT.

► Logiciels (sur votre ordinateur de développement) :

- **Arduino IDE** : Pour écrire, compiler et téléverser le code (firmware) sur l'ESP32. Vous devrez y ajouter le support des cartes ESP32 via le gestionnaire de cartes.
- **Bibliothèques Arduino spécifiques aux capteurs** : Vous installerez ces bibliothèques via le gestionnaire de bibliothèques de l'IDE Arduino. Par exemple :
 - `Adafruit_SHT31` pour le capteur de température/humidité.
 - `PubSubClient` ou `AsyncMqttClient` pour la communication MQTT.
 - `SoftwareSerial` si vous utilisez l'UART pour le MH-Z19B (bien que des broches UART matérielles soient préférables).
 - `Adafruit_ADS1X15` pour le module ADC.
 - Des bibliothèques communautaires pour le MH-Z19B et les capteurs MQ (vous devrez peut-être en trouver ou adapter du code).
- **Pilotes USB-Série** : Pour que votre ordinateur puisse communiquer avec l'ESP32 via le câble USB (souvent CP210x ou CH340, dépend du module ESP32).

Le Raspberry Pi Central

Le Raspberry Pi est le "quartier général" qui collecte, stocke, analyse et visualise toutes les données.

Langages de Programmation :

Python : C'est le langage de prédilection pour le Raspberry Pi dans ce type de projet. Il sera utilisé pour :

- Écrire le script qui s'abonne aux messages MQTT (pour recevoir les données des nœuds).
- Stocker les données dans la base de données InfluxDB.
- Implémenter la logique de calcul (ex: ITH) et de détection des seuils critiques.
- Configurer le Raspberry Pi en tant que point d'accès Wi-Fi (si nécessaire).

Shell Scripting (Bash) : Utile pour l'automatisation de tâches système, le démarrage des services, la gestion des mises à jour, et potentiellement pour la configuration du point d'accès Wi-Fi.

Le Raspberry Pi Central

Logiciels (à installer sur le Raspberry Pi) :

► **Système d'Exploitation :**

- **Raspberry Pi OS (Desktop)** : La version "Desktop" inclut une interface graphique et un navigateur, nécessaire si vous voulez visualiser Grafana directement sur le Pi avec un écran connecté.

► **Logiciels de Communication :**

- **Mosquitto (MQTT Broker)** : Le serveur qui reçoit et distribue les messages MQTT entre les nœuds et le script Python/Grafana.
- **hostapd** : Pour transformer le Raspberry Pi en point d'accès Wi-Fi.
- **dnsmasq** : Pour gérer l'attribution d'adresses IP et le DNS pour les appareils connectés au point d'accès Wi-Fi.

Le Raspberry Pi Central

► Base de Données :

- **InfluxDB** : Une base de données optimisée pour les données de séries temporelles, parfaites pour les capteurs.

► Outil de Visualisation et d'Alerte :

- **Grafana** : Pour créer des tableaux de bord interactifs et configurer les règles d'alerte.

Le Raspberry Pi Central

► Environnement Python :

- **Python 3** : Généralement préinstallé sur Raspberry Pi OS.
- **pip** : Le gestionnaire de paquets Python.
- **Bibliothèques Python** (à installer via `pip`) :
 - **paho-mqtt** : Pour que votre script Python puisse communiquer avec le broker MQTT.
 - **influxdb-client** : Pour que votre script Python puisse écrire des données dans InfluxDB.
 - Autres bibliothèques si vous utilisez des API pour les alertes (ex: `requests` pour les requêtes HTTP si vous utilisez une API Telegram ou SMS).

Le Raspberry Pi Central

► Outils Optionnels mais Utiles :

- **git** : Pour gérer les versions de votre code et récupérer des projets depuis GitHub.
 - **nano ou vim** : Éditeurs de texte en ligne de commande pour modifier les fichiers de configuration.
 - **htop ou glances** : Pour surveiller les ressources système (CPU, mémoire, etc.).
- **Un navigateur web** (ex: Chromium) : Pour visualiser Grafana directement.