

Rapport d'Étude : Système de Surveillance d'Hangar d'Élevage Avicole basé sur Raspberry Pi et Réseau de Capteurs Sans Fil

1. Introduction

Ce rapport détaille la conception et la mise en œuvre d'un système de surveillance environnemental pour un hangar d'élevage de poulets de chair, situé à Djelfa, Algérie. L'objectif est d'assurer un contrôle précis des conditions ambiantes (température, humidité, qualité de l'air, ventilation) afin d'optimiser le bien-être animal, la productivité et de prévenir les risques sanitaires liés aux variations climatiques extrêmes de la région (étés chauds, hivers froids). Le système s'articule autour d'un réseau de capteurs sans fil distribués dans le hangar, communiquant avec un Raspberry Pi centralisé pour la collecte, l'analyse et la visualisation des données.

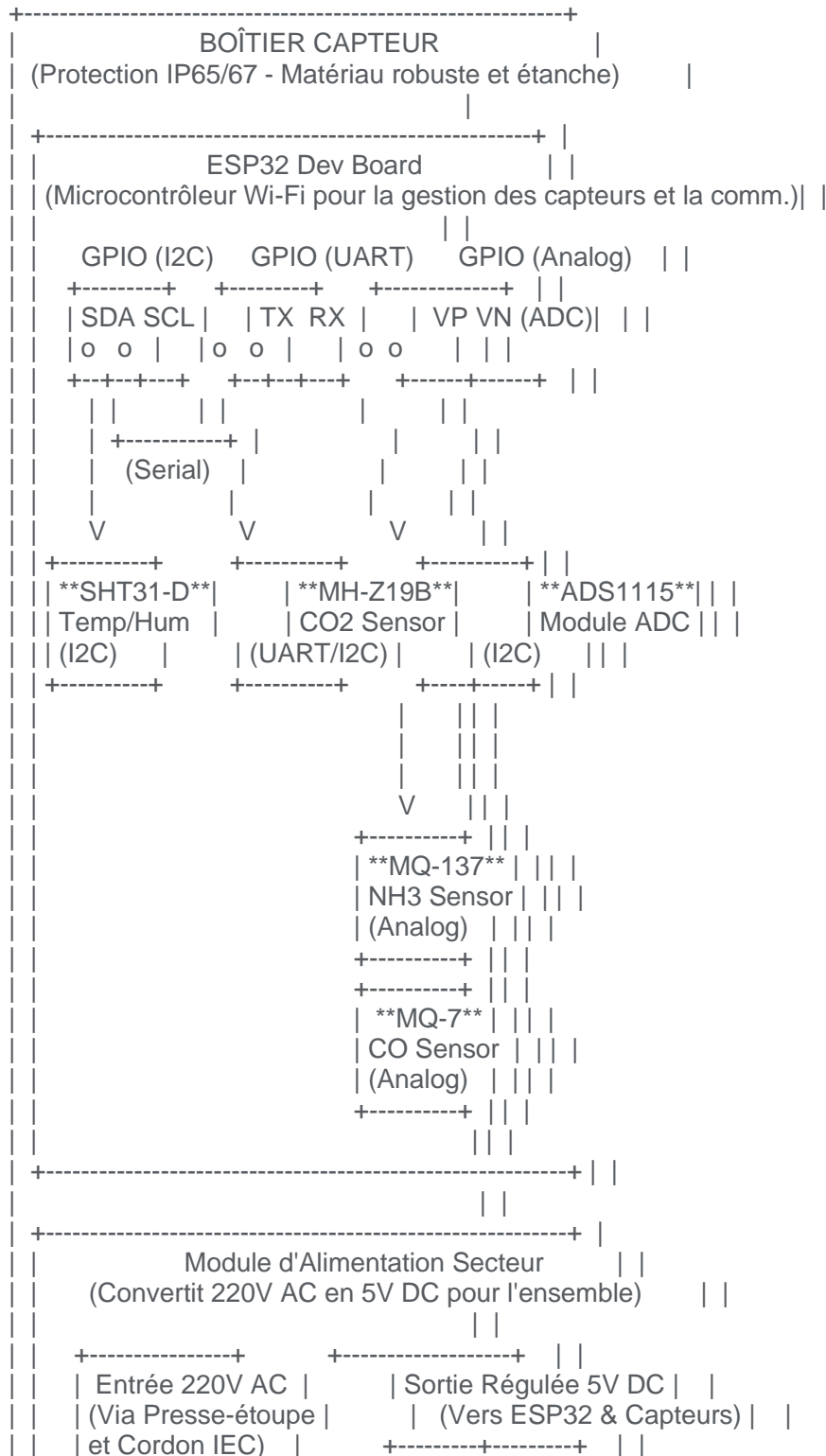
2. Rappel des Paramètres à Surveiller et Leur Importance

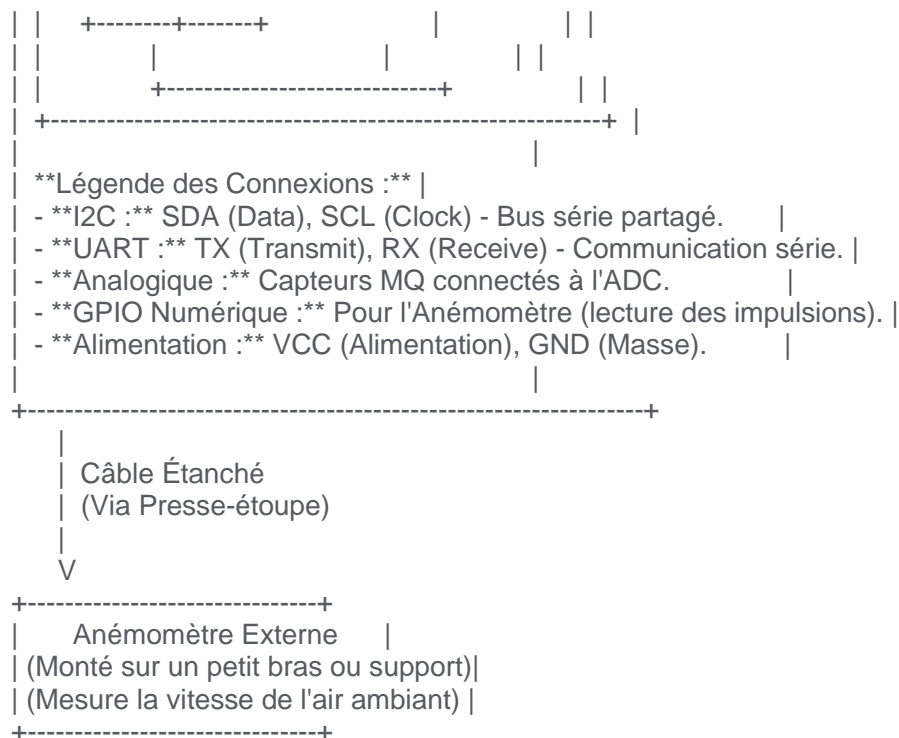
Pour garantir un environnement optimal aux poulets de chair, les paramètres suivants doivent être mesurés et surveillés en continu :

- **Température de l'air** : C'est le facteur le plus critique. Des températures trop élevées provoquent un stress thermique, une réduction de la consommation d'aliments et une mortalité accrue. Des températures trop basses entraînent un inconfort, une augmentation de la consommation d'aliments pour maintenir la chaleur corporelle et une baisse des performances.
- **Humidité relative de l'air** : En combinaison avec la température, elle influence fortement le ressenti thermique. Une humidité trop élevée favorise le développement de maladies respiratoires, l'humidité de la litière et la prolifération bactérienne. Une humidité trop basse peut causer de l'irritation des voies respiratoires.
- **Concentration en Ammoniac (NH_3)** : Produit par la décomposition des fientes, l'ammoniac est un gaz irritant pour le système respiratoire et les yeux des volailles, pouvant entraîner des lésions et des prédispositions aux infections. Des niveaux élevés sont un signe de litière humide ou de ventilation insuffisante.
- **Concentration en Dioxyde de Carbone (CO_2)** : Indicateur de la qualité de la ventilation et de la présence des animaux (respiration). Des niveaux élevés de CO_2 sont le signe d'une ventilation insuffisante, pouvant entraîner une mauvaise qualité de l'air, une léthargie et un stress chez les oiseaux.
- **Concentration en Monoxyde de Carbone (CO)** : Un gaz dangereux et inodore, principalement produit par des systèmes de chauffage à combustion mal réglés ou mal ventilés. Des niveaux même faibles sont toxiques et peuvent être fatals pour les animaux et le personnel. Sa surveillance est cruciale si des chauffages à flamme nue sont utilisés.
- **Vitesse de l'air (Ventilation)** : Essentielle pour l'évacuation de la chaleur, de l'humidité et des gaz nocifs. Une bonne vitesse de l'air est bénéfique en période chaude pour le rafraîchissement, mais doit être contrôlée en période froide pour éviter les courants d'air excessifs et les pertes de chaleur.

3. Schéma du Nœud Capteur (Boîtier) avec Intégration Externe et Alimentation Secteur 220V

Le cœur du système repose sur des **nœuds capteurs autonomes**, alimentés par le secteur 220V et communiquant sans fil. Chaque nœud est un boîtier étanche contenant l'électronique et les capteurs internes, tandis que l'anémomètre est monté à l'extérieur pour des mesures non perturbées.





4. Détail de Chaque Composant Utilisé dans le Schéma

Chaque nœud capteur est composé des éléments suivants :

- **ESP32 Dev Board (Microcontrôleur) :**
 - **Rôle :** Cerveau du nœud capteur. Il lit les données des différents capteurs, les traite, gère la connexion Wi-Fi et envoie les informations au Raspberry Pi central via le protocole MQTT.
 - **Avantages :** Intègre le Wi-Fi, possède des capacités de traitement suffisantes, est économe en énergie (même sans mode veille profonde dans cette configuration), et est très supporté par une vaste communauté de développeurs (Arduino IDE, MicroPython).
- **SHT31-D (Capteur de Température et d'Humidité) :**
 - **Rôle :** Mesure la température ambiante et l'humidité relative.
 - **Technologie :** Capteur CMOS avec sortie numérique (I2C), offrant une excellente précision et fiabilité par rapport à des alternatives moins chères.
- **MH-Z19B (Capteur de Dioxyde de Carbone - CO2) :**
 - **Rôle :** Mesure la concentration de CO2 dans l'air.
 - **Technologie :** Capteur infrarouge non dispersif (NDIR), la méthode la plus fiable et précise pour mesurer le CO2. Il communique via le port série (UART) ou I2C.
 - **Note :** Le calibrage automatique doit être désactivé dans le code si le niveau de CO2 ne revient jamais à la valeur extérieure (400 ppm) pour éviter des dérives de mesure sur le long terme. Un calibrage manuel périodique en air frais est recommandé.
- **ADS1115 (Convertisseur Analogique-Numérique - ADC) :**
 - **Rôle :** Permet à l'ESP32 de lire les signaux analogiques des capteurs MQ-137 et MQ-7. L'ESP32 a un ADC interne, mais l'ADS1115 offre une

meilleure résolution (16 bits) et précision, ce qui est crucial pour les capteurs de gaz.

- **Technologie** : Communique avec l'ESP32 via le bus I2C.
 - **MQ-137 (Capteur d'Ammoniac - NH₃) :**
 - **Rôle** : Détecte la présence et la concentration approximative d'ammoniac.
 - **Technologie** : Capteur à semi-conducteur (MOS). Sa sortie est analogique.
 - **Limitations** : Moins précis et moins spécifique que les capteurs électrochimiques industriels. Il peut réagir à d'autres gaz, ce qui nécessite un étalonnage soigné et une interprétation des données comme une "tendance" plutôt qu'une mesure exacte. Sa lecture est également très sensible à la température et à l'humidité.
 - **MQ-7 (Capteur de Monoxyde de Carbone - CO) :**
 - **Rôle** : Détecte la présence de monoxyde de carbone.
 - **Technologie** : Capteur à semi-conducteur (MOS) avec sortie analogique.
 - **Limitations** : Similaires au MQ-137 en termes de précision et spécificité. Nécessite un cycle de chauffage particulier pour des lectures optimales (précisions à intégrer dans le code).
 - **Anémomètre à Impulsions (ex: à godets ou à hélice) :**
 - **Rôle** : Mesure la vitesse de l'air.
 - **Technologie** : Génère des impulsions électriques dont la fréquence est proportionnelle à la vitesse du vent. Connecté à une broche GPIO numérique de l'ESP32 pour un comptage via interruption.
 - **Spécificité** : Doit être un modèle adapté à l'environnement intérieur d'un hangar (faible vitesse, résistant à la poussière).
 - **Module d'Alimentation Secteur 220V AC vers 5V DC :**
 - **Rôle** : Convertit le courant alternatif 220V du réseau en courant continu 5V DC stable pour alimenter l'ensemble du nœud (ESP32 et tous les capteurs).
 - **Sécurité** : Composant critique nécessitant une isolation électrique irréprochable et des protections (fusible, surtension).
-

5. Disposition des Capteurs dans le Boîtier

La conception du boîtier doit garantir l'exactitude des mesures et la durabilité du système face à l'environnement exigeant de l'élevage (poussière, humidité, fluctuations de température).

- **Boîtier Général :**
 - **Matériau** : Plastique robuste (ABS, Polycarbonate) avec un indice de protection **IP65 ou IP67** pour une étanchéité totale à la poussière et aux projections d'eau (nettoyage).
 - **Montage** : Système de fixation robuste pour suspendre ou monter le boîtier à la hauteur des animaux (environ 30-50 cm du sol).
- **Capteurs de Température/Humidité (SHT31-D) et Gaz (CO₂, NH₃, CO) :**
 - **Emplacement** : Ces capteurs doivent être placés sur la **face inférieure ou latérale du boîtier**.
 - **Ouvertures** : Des ouvertures sont nécessaires pour permettre l'accès de l'air aux capteurs. Ces ouvertures doivent être couvertes par une

membrane microporeuse hydrophobe (type Gore-Tex®). Cette membrane est essentielle : elle permet la circulation de l'air tout en bloquant efficacement la poussière, l'eau, les insectes et les grosses particules.

- **Isolation Thermique** : Assurez-vous que le capteur de température/humidité est éloigné de tout composant électronique interne qui génère de la chaleur (ESP32, module d'alimentation) pour éviter les lectures faussées. Les capteurs MQ chauffent : prévoyez un espace suffisant pour leur dissipation thermique et pour qu'ils n'impactent pas les autres capteurs.
 - **Anémomètre** :
 - **Emplacement** : Il doit être monté à l'**extérieur du boîtier principal**, idéalement sur un petit bras de déport ou sur le dessus du boîtier.
 - **Raison** : Cette disposition permet à l'anémomètre de mesurer la vitesse de l'air ambiant sans être perturbé par les turbulences créées par le boîtier lui-même.
 - **Connexion** : Le câble de l'anémomètre entrera dans le boîtier via un **presse-étoupe étanche** pour se connecter à l'ESP32.
 - **Composants Électroniques Internes (ESP32, ADS1115, Module d'Alimentation)** :
 - **Emplacement** : Fixés solidement à l'intérieur du boîtier, sur un support ou un PCB dédié.
 - **Protection** : Un "conformal coating" (revêtement protecteur) peut être appliqué sur les cartes électroniques pour une protection accrue contre l'humidité et les particules fines, même à l'intérieur d'un boîtier étanche.
-

6. Intégration avec Le "Cerveau Central", Raspberry Pi en Mode Sans Fil

Le Raspberry Pi centralise la collecte, le traitement et la visualisation des données provenant de tous les nœuds capteurs sans fil.

- **Rôle du Raspberry Pi** :
 1. **Réception des données** : Il agit comme un récepteur principal pour toutes les données envoyées par les nœuds capteurs.
 2. **Stockage des données** : Il enregistre les données dans une base de données.
 3. **Analyse et calculs** : Il effectue les calculs nécessaires (ex: Indice de Température-Humidité - ITH) et applique les règles de seuils critiques.
 4. **Visualisation** : Il héberge l'interface utilisateur pour afficher les données en temps réel et historiques.
 5. **Alertes** : Il déclenche des alertes en cas de dépassement des seuils.
- **Communication Sans Fil** :
 1. Les nœuds capteurs (ESP32) et le Raspberry Pi se connecteront au même **réseau Wi-Fi du hangar**.
 2. Le protocole de communication utilisé est **MQTT (Message Queuing Telemetry Transport)**. C'est un protocole léger, efficace et idéal pour l'IoT. Chaque nœud capteur "publie" ses données sur des "sujets" MQTT spécifiques (ex: hangar/zone1/temperature, hangar/zone2/ammoniac). Le Raspberry Pi "s'abonne" à ces sujets pour recevoir toutes les données.

7. Logiciels et Protocoles Nécessaires pour le Bon Fonctionnement du Système

La pile logicielle sur le Raspberry Pi est cruciale pour le traitement complet des données :

- **Sur les Nœuds Capteurs (ESP32) :**
 - **Firmware (code embarqué) :** Écrit en **Arduino (C++)** ou **MicroPython**. Ce code gère la lecture des capteurs, la communication Wi-Fi et la publication des messages MQTT.
 - **Bibliothèques spécifiques :** Pour chaque capteur (ex: SHT31, MH-Z19B, ADS1115) et pour la communication MQTT (ex: PubSubClient pour Arduino ou umqtt.simple pour MicroPython).
- **Sur le Raspberry Pi Central :**
 - **Système d'exploitation :** **Raspberry Pi OS (anciennement Raspbian).**
 - **Serveur MQTT (Broker) :** **Mosquitto.** C'est le "cœur" du système MQTT. Il reçoit les messages de tous les nœuds et les distribue aux "clients" abonnés (comme le script Python et Grafana).
 - **Base de Données :** **InfluxDB.** Une base de données optimisée pour les séries temporelles, idéale pour stocker de grandes quantités de données de capteurs avec des horodatages.
 - **Outil de Visualisation :** **Grafana.** Interface utilisateur web puissante pour créer des tableaux de bord interactifs affichant les données en temps réel et historiques, avec des graphiques personnalisables.
 - **Langage de Script :** **Python.**
 - **Script de collecte et traitement :** Un script Python s'abonnera aux sujets MQTT, lira les données, effectuera les calculs (ITH, vérification des seuils), et écrira les données dans InfluxDB.
 - **Script d'alerte :** Intégré au script précédent ou séparé, il enverra des notifications (e-mail, SMS via un service tiers, Telegram, etc.) en cas de dépassement des seuils critiques.
 - **Protocoles :**
 - **Wi-Fi (IEEE 802.11) :** Pour la connectivité sans fil.
 - **MQTT :** Protocole de communication entre les nœuds et le Raspberry Pi.
 - **HTTP/HTTPS :** Pour accéder à l'interface web de Grafana depuis un navigateur.

8. Solutions pour Résoudre les Problèmes du Signal Wi-Fi

Un hangar de 50x10m peut présenter des défis pour le signal Wi-Fi en raison de sa taille, de la présence de structures métalliques et de la possible densité d'animaux.

- **Emplacement Stratégique du Point d'Accès Wi-Fi :**
 - Placez votre routeur Wi-Fi (ou point d'accès) au **centre du hangar** et en hauteur si possible, pour maximiser la couverture et minimiser les obstacles.
 - Évitez les murs épais ou les grandes structures métalliques qui peuvent bloquer le signal.

- **Répéteurs Wi-Fi (Range Extenders) ou Système Mesh :**
 - Si un seul point d'accès ne suffit pas, envisagez d'ajouter des **répéteurs Wi-Fi** aux extrémités du hangar pour étendre la portée du signal.
 - Pour une solution plus robuste et performante, un **système Wi-Fi Mesh** peut créer un réseau unifié avec plusieurs points d'accès qui coopèrent pour fournir une couverture homogène.
 - **Antennes Externes :**
 - Certains routeurs ou points d'accès permettent d'utiliser des **antennes externes à gain élevé**, qui peuvent améliorer significativement la portée et la qualité du signal par rapport aux antennes internes.
 - **Analyse du Spectre Wi-Fi :**
 - Utilisez des applications d'analyse Wi-Fi sur smartphone ou PC (ex: WiFi Analyzer) pour identifier les canaux Wi-Fi encombrés et choisir un canal moins utilisé pour votre réseau, réduisant ainsi les interférences.
 - **Qualité du Matériel Wi-Fi :**
 - Investissez dans des routeurs et des répéteurs de bonne qualité, conçus pour les grands espaces ou les environnements industriels si le budget le permet.
 - **Sécurité et Isolation du Réseau :**
 - Créez un **réseau Wi-Fi dédié** à votre système de surveillance, séparé de votre réseau personnel ou professionnel. Cela améliore la sécurité et évite que d'autres appareils ne surchargent le réseau.
-

9. Le Nombre et l'Emplacement Idéal de Chaque Nœud dans l'Hangar

Pour un hangar de 50x10m, une couverture adéquate nécessite une planification soigneuse.

- **Nombre de Nœuds :**
 - Un minimum de **trois (3) à cinq (5) nœuds** est recommandé pour une bonne représentativité.
 - **3 nœuds** : Un au centre, un à chaque extrémité. ($50\text{m} / 2 = 25\text{m}$ pour le central, donc 25m de chaque côté).
 - **5 nœuds** : Un à chaque extrémité, et trois répartis uniformément entre elles. ($50\text{m} / 4 = 12.5\text{m}$ entre chaque nœud).
 - **Justification** : Cette répartition permet de capter les variations longitudinales de température, d'humidité et de gaz, qui peuvent être influencées par la ventilation, les ouvertures, ou les zones de chauffage/refroidissement.
- **Emplacement Idéal de Chaque Nœud :**
 - **Hauteur** : Les nœuds doivent être placés à la **hauteur des animaux**, généralement entre 30 et 50 cm du sol, car c'est là que les conditions environnementales les affectent directement.
 - **Distance des Obstacles** : Évitez de placer les nœuds directement sous ou très près des ventilateurs, des chauffages, ou des abreuvoirs/mangeoires, car cela pourrait fausser les lectures locales. Laissez un certain dégagement.

- **Accessibilité** : Positionnez les nœuds de manière à ce qu'ils soient accessibles pour la maintenance (nettoyage, vérification) sans perturber excessivement les animaux.
 - **Représentativité** : L'objectif est de mesurer les conditions moyennes d'une "zone" du hangar. La répartition longitudinale est la plus critique.
-

10. Avez-vous oublié des choses pour avoir un rapport détaillé?

Votre liste était déjà très complète ! Cependant, voici quelques points supplémentaires qui pourraient être ajoutés pour rendre le rapport encore plus exhaustif et pertinent pour la réalisation pratique du projet :

- **Sécurité Électrique et Normes** : Insister sur l'importance de suivre les normes électriques algériennes et internationales (ex: CEI) pour la sécurité des installations 220V, l'utilisation de différentiels, de fusibles et de boîtiers conformes.
- **Calibration et Maintenance des Capteurs** : Souligner la nécessité d'un plan de calibration régulière des capteurs (surtout CO₂, NH₃, CO) pour garantir la précision des mesures sur le long terme. Mentionner la maintenance préventive (nettoyage des capteurs, vérification des connexions).
- **Logiciel de Traitement des Données** : Préciser les algorithmes simples (ex: ITH) et les seuils d'alerte spécifiques pour les poulets de chair en fonction de leur âge, qui devront être implémentés dans le script Python du Raspberry Pi.
- **Planification de la Capacité** : Estimer la capacité de stockage nécessaire dans la base de données (InfluxDB) en fonction de la fréquence d'échantillonnage et du nombre de nœuds.
- **Interface Utilisateur et Alertes** : Détailler les options d'interface utilisateur (tableau de bord Grafana via web) et les modalités d'alerte (email, SMS, notifications push) qui seront mises en place.
- **Budget Prévisionnel** : Bien que non demandé, une estimation sommaire des coûts par nœud et pour le Raspberry Pi central, ainsi que les logiciels, pourrait être un ajout utile dans un rapport final de projet.
- **Phase de Test et de Déploiement** : Décrire les étapes de test initiales (en laboratoire, puis sur site), et le plan de déploiement progressif des nœuds.

Ce rapport devrait vous fournir une base solide pour la documentation et la mise en œuvre de votre système de surveillance. N'hésitez pas si vous avez d'autres questions !

Cahier de Test Détaillé - Système de Surveillance d'Hangar Avicole

Objectif : Valider le fonctionnement de chaque composant du système de surveillance environnemental (nœuds capteurs et Raspberry Pi central) avant le déploiement en environnement réel. Ce cahier de test vise à s'assurer de la précision des mesures, de la robustesse de la communication et de la fiabilité de l'alerte.

Environnement de Test :

- Laboratoire ou espace de travail propre et contrôlé.
 - Alimentation électrique stable (220V).
 - Accès à un PC/ordinateur portable pour la configuration initiale et la surveillance.
 - Matériel de mesure de référence (thermomètre/hygromètre calibré, détecteur de gaz portable pour les MQ si disponible).
-

Phase 1 : Tests Unitaires des Nœuds Capteurs (Par Nœud)

Chaque nœud capteur (ESP32 + capteurs) doit être testé individuellement avant d'être intégré au réseau.

1.1. Test de l'Alimentation et de l'ESP32

- **Description :** Vérifier que le module d'alimentation secteur fonctionne correctement et que l'ESP32 démarre.
- **Étapes :**
 1. Connecter le module d'alimentation secteur 220V.
 2. Mesurer la tension de sortie du module (doit être 5V DC).
 3. Connecter l'ESP32 au module d'alimentation.
 4. Vérifier que le témoin lumineux de l'ESP32 s'allume.
 5. Téléverser un programme simple (ex: "Blink") sur l'ESP32 pour confirmer son bon fonctionnement.
- **Critère de Réussite :** Tension de sortie stable de 5V DC, ESP32 s'allume et exécute le programme "Blink".

1.2. Tests des Capteurs de Température et d'Humidité (SHT31-D)

- **Description :** Vérifier la lecture correcte des valeurs de température et d'humidité.
- **Étapes :**
 1. Connecter le SHT31-D à l'ESP32 (broches I2C).
 2. Téléverser un code ESP32 qui lit et affiche les valeurs de température et d'humidité sur le moniteur série.
 3. Comparer les lectures du SHT31-D avec un thermomètre/hygromètre de référence.
 4. Simuler des variations (souffler de l'air chaud/froid, respirer près du capteur pour l'humidité).
- **Critère de Réussite :** Lectures stables et cohérentes (à $\pm 0.5^{\circ}\text{C}$ et $\pm 3\%\text{HR}$) avec l'appareil de référence.

1.3. Tests du Capteur de Dioxyde de Carbone (CO₂) - MH-Z19B

- **Description** : Vérifier la lecture du CO₂ et la configuration de son mode.
- **Étapes** :
 1. Connecter le MH-Z19B à l'ESP32 (broches UART ou I2C).
 2. Téléverser le code ESP32 pour lire et afficher les valeurs de CO₂ (en ppm) sur le moniteur série.
 3. **Désactiver l'ABC (Automatic Baseline Correction)** via le code ou les commandes spécifiques au capteur.
 4. Comparer une lecture en air frais (extérieur) avec la valeur attendue (~400 ppm).
 5. Respirer directement sur le capteur pour observer une augmentation rapide des ppm.
- **Critère de Réussite** : Lectures de CO₂ cohérentes, ABC désactivé, réaction rapide aux variations.

1.4. Tests des Capteurs de Gaz (NH₃ - MQ-137, CO - MQ-7) et ADC (ADS1115)

- **Description** : Vérifier la connexion de l'ADC et la lecture analogique des capteurs MQ.
- **Étapes** :
 1. Connecter l'ADS1115 à l'ESP32 (broches I2C).
 2. Connecter les capteurs MQ-137 et MQ-7 aux entrées analogiques de l'ADS1115 (avec les résistances de charge appropriées).
 3. Téléverser le code ESP32 pour lire les valeurs brutes de l'ADS1115 pour chaque capteur MQ sur le moniteur série.
 4. **Pour le MQ-7 (CO)** : Vérifier que le cycle de chauffage est correctement mis en œuvre dans le code.
 5. Exposer les capteurs MQ à une source connue (ex: solution d'ammoniac diluée pour le MQ-137, gaz briquet pour le MQ-7 *avec prudence et bonne ventilation*) et observer la réaction.
 6. **Note** : Sans équipement de calibration professionnel, ces tests servent à confirmer la *réactivité* et la *connectivité*, non la précision absolue.
- **Critère de Réussite** : Lectures brutes de l'ADC cohérentes, réaction visible des capteurs MQ aux gaz, cycle de chauffage du MQ-7 fonctionnel.

1.5. Test de l'Anémomètre

- **Description** : Vérifier la capacité de l'ESP32 à compter les impulsions et calculer la vitesse de l'air.
- **Étapes** :
 1. Connecter l'anémomètre à la broche GPIO numérique choisie de l'ESP32.
 2. Téléverser le code ESP32 qui compte les impulsions de l'anémomètre et les convertit en vitesse de l'air (en m/s ou km/h) en utilisant la formule de calibration du fabricant.
 3. Faire tourner manuellement l'anémomètre (ou le souffler) et observer les valeurs de vitesse affichées sur le moniteur série.
- **Critère de Réussite** : Comptage des impulsions précis, conversion en vitesse de l'air logique.

1.6. Test de Communication Wi-Fi et MQTT (Nœud vers Broker)

- **Description** : Valider que chaque nœud peut se connecter au réseau Wi-Fi du Raspberry Pi (mode AP) et envoyer des données via MQTT au broker.

- **Prérequis** : Le Raspberry Pi central doit déjà être configuré en mode AP et son broker Mosquitto doit fonctionner. éditer le fichier `/etc/hostapd/hostapd.conf` pour récupérer le **SSID** et le **mot de passe** du réseau Wi-Fi du Pi.
 - **Étapes** :
 1. Dans le code ESP32 de chaque nœud, configurer les identifiants Wi-Fi du Raspberry Pi et l'adresse IP du broker MQTT (l'IP statique du Pi).
 2. Téléverser le code sur l'ESP32.
 3. Sur le Raspberry Pi, utiliser un client MQTT (ex: `mosquitto_sub` en ligne de commande) pour s'abonner aux sujets des données que le nœud envoie (ex: `hangar/zoneX/#`).
 4. Observer les données reçues sur le Raspberry Pi.
 - **Critère de Réussite** : Le nœud se connecte au Wi-Fi du Pi, publie des messages MQTT, et le Raspberry Pi reçoit correctement ces messages.
-

Phase 2 : Tests d'Intégration et de Performance du Système Complet

Cette phase teste l'ensemble du système (tous les nœuds + Raspberry Pi) et son comportement.

2.1. Test de Connexion de Tous les Nœuds et Flux de Données

- **Description** : Assurer que tous les nœuds se connectent simultanément et envoient des données de manière stable au Raspberry Pi.
- **Étapes** :
 1. Mettre sous tension tous les nœuds capteurs.
 2. Sur le Raspberry Pi, vérifier les logs du broker Mosquitto et du script Python de collecte pour confirmer que les données de chaque nœud sont reçues.
 3. Laisser le système tourner pendant plusieurs heures pour détecter d'éventuelles déconnexions ou pertes de données.
- **Critère de Réussite** : Tous les nœuds se connectent et envoient des données de manière continue. Aucune déconnexion inattendue ou perte de données sur une période de 4 à 8 heures.

2.2. Test de la Base de Données (InfluxDB) et Visualisation (Grafana)

- **Description** : Vérifier que les données sont correctement stockées et affichées dans Grafana.
- **Étapes** :
 1. Accéder à l'interface Grafana via un navigateur (sur le Pi directement ou depuis un smartphone/PC).
 2. Vérifier que toutes les sources de données (les capteurs de chaque nœud) sont visibles et que les graphiques se mettent à jour en temps réel.
 3. Consulter l'historique des données sur différentes périodes.
- **Critère de Réussite** : Toutes les données de tous les capteurs sont stockées dans InfluxDB et affichées correctement dans Grafana.

2.3. Test des Règles d'Alerte

- **Description** : Valider que les alertes sont déclenchées correctement lorsque les seuils sont dépassés et que les notifications sont envoyées.
- **Étapes** :
 1. Modifier temporairement les seuils d'alerte dans Grafana (ou dans le script Python) à des valeurs facilement atteignables.

2. Simuler des conditions qui déclenchent les alertes (ex: souffler de l'air chaud sur un capteur de température, utiliser un gaz pour les capteurs MQ).
 3. Vérifier que les notifications (e-mail, Telegram, etc.) sont reçues.
- **Critère de Réussite** : Les alertes se déclenchent aux seuils configurés et les notifications sont reçues par les canaux choisis.
- 2.4. *Test de Robustesse du Réseau Wi-Fi*
- **Description** : Évaluer la stabilité de la connexion Wi-Fi dans des conditions plus exigeantes.
 - **Étapes** :
 1. Placer les nœuds à différentes distances et derrière des obstacles (si possible dans le labo).
 2. Introduire des interférences Wi-Fi (ex: utiliser d'autres appareils Wi-Fi à proximité).
 3. Surveiller la qualité de la connexion et les pertes de paquets via les logs du Raspberry Pi.
 - **Critère de Réussite** : La connexion Wi-Fi reste stable avec un minimum de pertes de paquets, même dans des conditions légèrement défavorables.
- 2.5. *Test de Stabilité et Fiabilité Globale*
- **Description** : Laisser le système fonctionner sans intervention pendant une période prolongée pour détecter des problèmes intermittents ou des fuites de mémoire.
 - **Étapes** :
 1. Lancer le système complet et le laisser tourner pendant au moins 24 à 48 heures.
 2. Surveiller régulièrement les tableaux de bord Grafana.
 3. Vérifier les logs du Raspberry Pi et des nœuds pour détecter des erreurs ou des redémarrages inattendus.
 - **Critère de Réussite** : Le système fonctionne sans interruption majeure ni erreur critique pendant la période de test prolongée.
-

Conclusion des Tests et Recommandations

Une fois tous les tests effectués, ce rapport doit conclure sur l'état de préparation du système pour le déploiement.

- **Bilan des Tests** : Résumer les succès et les échecs, ainsi que les problèmes identifiés.
 - **Actions Correctives** : Lister les actions qui ont été entreprises pour corriger les problèmes.
 - **Recommandations pour le Déploiement** :
 - Conseils spécifiques pour le placement final des nœuds dans le hangar (hauteur, distance des équipements).
 - Nécessité de protections physiques supplémentaires pour les boîtiers des nœuds si l'environnement est particulièrement agressif (eau, poussière).
 - Plan de calibration périodique des capteurs (surtout gaz).
 - Plan de maintenance préventive (nettoyage des capteurs, vérification des connexions).
-

Est-ce que j'ai oublié des choses pour avoir un rapport détaillé ?

Votre demande était très complète ! Ce rapport couvre l'essentiel. Pour être encore plus exhaustif, vous pourriez ajouter :

- **Historique des versions** : Une section pour suivre les versions du code des nœuds et du Raspberry Pi.
- **Journal des tests** : Un tableau pour chaque test, avec la date, l'exécutant, le résultat (pass/fail), les observations et les actions correctives.
- **Consommation électrique (optionnel)** : Si la consommation est une préoccupation, des mesures de consommation pour chaque nœud et le Raspberry Pi.
- **Tests de charge/scalabilité (optionnel)** : Si vous prévoyez d'avoir un très grand nombre de nœuds, des tests pour voir comment le Raspberry Pi gère la charge (si beaucoup de données arrivent simultanément).

Avec ce cahier de test, vous avez une méthodologie claire pour valider votre système avant de le mettre en service dans le hangar. Bonne chance !