

Cahier des charges — Projet Ventilation (VMC) + App de Monitoring

Version : 1.0

Date : 08/12/2025

Auteurs : Vincent Canaguy, Yahya Er-Rahmaouy, Tayeb Nazim Djeradi, Aladji Kassoum BINI

1. Résumé du projet

Notre projet consiste à réaliser un dispositif IoT permettant d'ajuster automatiquement la ventilation (VMC) selon les conditions météo et de qualité de l'air (pluie, vent, ensoleillement, pollution, humidité si disponible) et offrant une application de monitoring et de contrôle manuel pour l'utilisateur.

2. Contexte et objectifs

- **Contexte** : maison connectée, qualité de l'air intérieur, efficacité énergétique
- **Objectifs principaux** :
 - Assurer une bonne qualité de l'air intérieur en adaptant le débit de ventilation aux conditions (pollution extérieure, température, humidité, vent).
 - Réduire l'humidité et évacuer l'air vicié pour prévenir moisissures et inconfort.
 - Fournir un contrôle manuel et des notifications via une application.

3. Périmètre

- **Matériel** :
 - moteur
 - carte ESP32
- **Logiciel**: - Firmware embarqué pour l'actionneur (gestion ouverture 0–100%) - Application mobile pour monitoring et contrôle

4. Acteurs

- **Utilisateur final** : propriétaire / locataire
- **Système (embarqué)** : MCU + actionneurs
- **Application mobile** : monitoring et contrôle
- **Services tiers** : APIs météo externes

5. Exigences fonctionnelles

- **01 Récupération des conditions**
 - *Description* : Le système doit pouvoir récupérer la qualité de l'air (pollution), l'humidité (si disponible), la température, la luminosité et le vent via API.
 - *Priorité* : Haute
 - *Critères d'acceptation* : Données météo reçues toutes les 10 minutes; logs des valeurs stockés localement et/ou sur cloud.
- **02 Règles d'ajustement automatique de l'ouverture**
 - *Description* : Le système ajuste automatiquement l'ouverture du clapet (et donc le débit) selon des règles configurables (ex : ouvrir davantage si pollution/CO /humidité > seuil, réduire sinon).
 - *Priorité* : Haute
 - *Critères d'acceptation* : Scénarios de test montrant que la règle est appliquée correctement; possibilité de forcer manuellement.

- **03 Contrôle manuel depuis l'application**
 - *Description* : L'utilisateur peut ouvrir/fermer le clapet de ventilation et régler le pourcentage d'ouverture (paliers ou valeurs) depuis l'application.
 - *Priorité* : Haute
 - *Critères d'acceptation* : Commande envoyée et état mis à jour sous 3 s; remontée d'erreur si indisponible.
- **04 Mode automatique / manuel**
 - *Description* : Basculer entre mode automatique (règles air/météo) et mode manuel.
 - *Priorité* : Moyenne
 - *Critères d'acceptation* : L'état du mode est visible et persisté.
- **05 Notifications**
 - *Description* : Recevoir notifications en cas d'événement critique (ex : surcharge/surchauffe de l'actionneur, blocage clapet, conditions extrêmes) et non critiques (ex : changement de mode, ajustement d'ouverture).
 - *Priorité* : Moyenne
 - *Critères d'acceptation* : Notification reçue sur app et log enregistrés.
- **06 Historique & Dashboard**
 - *Description* : Visualiser l'historique des états (ouvert/fermé), niveaux d'ouverture (%), mesures météo/air et interventions manuelles.
 - *Priorité* : Basse
 - *Critères d'acceptation* : Graphiques consultables.
- **07 Sécurité & protection de l'actionneur**
 - *Description* : Arrêt immédiat en cas de surintensité/surchauffe de l'actionneur; alerte en cas d'obstruction/contre-pression détectable sur le clapet.
 - *Priorité* : Critique
 - *Critères d'acceptation* : Déclenchement de l'arrêt de sécurité sous 0.5 s lors d'une surintensité.

6. UI / UX

- Écrans essentiels : écran principal (ouverture ventilation: ouvert/fermé, % d'ouverture), écran historique, écran réglages (seuils air/météo), écran notifications.
- Flux : connexion -> page d'accueil -> contrôle rapide -> paramètres -> logs.

7. Cas d'utilisation

- En tant qu'utilisateur, je veux que l'ouverture du clapet augmente automatiquement quand la qualité de l'air est mauvaise (pollution/CO /humidité) ou que les conditions l'exigent.
- En tant qu'utilisateur, je veux pouvoir ouvrir/fermer le clapet et régler l'ouverture manuellement depuis l'app.

8. Tests & Critères d'acceptation

- Tests unitaires : firmware logique, backend endpoints.
- Tests d'intégration : envoi de télémétrie + réaction automatique.
- Tests d'acceptation utilisateur : scénarios (pluie, vent fort, obstruction) avec résultats attendus.