

Rapport d'avancement

PROJET DE FIN D'ETUDES

ESEO – Filière Électronique des Objets Connectés

Livrable

Nom du projet :	VENTILATION INTELLIGENTE
Etudiants :	AGOSSA KEWEN EDZANG MORO OLIVIER MADONA GEORGES TCHAPTCHE WOLGANE SERE OKANA RACHID

[PFE]_AGOSSA_Kewen_EDZANGMORO_Olivier_MADONA_Georges_TCHAPTCH
ET_Wolgane_SERE_Rachid.pdf

Type de projet : Conception et réalisation

1 Manuel d'utilisation

Système de ventilation intelligente

◆ Introduction

Le Système de Ventilation Intelligent (SVI) est une fenêtre automatisée permettant de renouveler l'air intérieur sans intervention humaine.

Il mesure en continu la qualité de l'air grâce à un capteur dédié, et déclenche automatiquement l'ouverture ou la fermeture de la fenêtre lorsque les seuils définis sont atteints.

Ce manuel explique le fonctionnement général, les composants utilisés, les procédures de mise en service, les méthodes d'utilisation, ainsi que les règles de sécurité et les solutions de dépannage rapide.

◆ a. Principe de fonctionnement

Le système mesure en continu la qualité de l'air intérieur grâce au capteur BME680 qui analyse la température, l'humidité et les composés organiques volatils.

Lorsque les valeurs dépassent les seuils définis (air vicié, humidité trop élevée, pollution interne), la station de commande envoie automatiquement une instruction au vérin électrique installé sur la fenêtre. Celui-ci ouvre la fenêtre pour ventiler la pièce.

Dès que la qualité de l'air revient à un niveau normal, la fenêtre se referme automatiquement.

L'utilisateur peut également piloter l'ouverture ou la fermeture via l'application Android en mode manuel.

◆ **b. Composants principaux**

Élément	Fonction	Emplacement
BME680	Mesure température, humidité, qualité d'air Détection des gaz (CO ₂ , fumées, COV)	Intérieur
Microcontrôleur nRF52832 (x2)	Un module pour l'acquisition des capteurs (station intérieure). Un module pour le contrôle du vérin (station fenêtre).	Module principal
Vérin électrique 750 N – 100 mm – 12 V	Ouvre/ferme la fenêtre	Fixé au châssis de la fenêtre
Ecran TFT	Visualisation locale des données et de l'état du système.	Près de la fenêtre
Bouton poussoir d'urgence	Ouverture/fermeture manuelle en cas de problème	Sur la carte principale
Application Android	Contrôle manuel et configuration.	Smartphone utilisateur
Batterie 12V	Alimentation principale du système + alerte sonore en cas de niveau faible	

◆ **c. Mise en service**

1. Vérifier l'alimentation

- a. S'assurer que la batterie 12 V est chargée.
- b. Observer le buzzer/LED : aucun signal = batterie correcte.

2. Mettre sous tension la station capteur

- a. Le BME680 démarre en moins de 2 secondes.

3. Mettre sous tension la station fenêtre

- a. Le module NRF se synchronise automatiquement avec la station capteur (BLE).

4. Initialisation automatique

- a. L'écran TFT affiche :
 - i. Température, humidité, COV
 - ii. État fenêtre : OUVERT / FERMÉ
 - iii. Niveau batterie

5. Lancer l'application Android

- a. Activer le Bluetooth du smartphone.

- b. Ouvrir l'application.
- c. Sélectionner "SVI – Window Control".

Le système est prêt à fonctionner.

◆ **d. Utilisation normale**

Le système est conçu pour fonctionner **en autonomie complète**, sans intervention :

- Lorsque la qualité de l'air **se dégrade** → **Ouverture automatique**.
- Lorsque les valeurs reviennent **sous les seuils** → **Fermeture automatique**.
- L'utilisateur peut consulter l'état et les mesures via l'écran TFT ou l'application.

Aucune action manuelle n'est requise en fonctionnement normal.

◆ **e. Affichage local et voyants**

Indicateur	Couleur / État
LED verte	Fenêtre ouverte
LED rouge	Fenêtre fermée
Buzzer	Batterie faible ou défaut majeur
Ecran TFT	Température (°C), Humidité (%), Taux de COV (équivalent CO ₂ approximé), État fenêtre : OUVERT / FERMÉ, Batterie : OK / FAIBLE

◆ **f. Commande manuelle depuis l'application**

Depuis l'application Android :

- Bouton **Ouvrir** → active le vérin pour ouvrir la fenêtre.
- Bouton **Fermer** → ordonne la fermeture complète.
- Température
- Humidité
- Niveau de COV

Si un obstacle est détecté ou si les conditions météo extérieures sont défavorables (pluie, vent), la commande manuelle est temporairement ignorée pour protéger le système.

◆ **g.** Sécurité et protection

Le SVI intègre plusieurs sécurités :

- **Arrêt automatique du vérin** en cas d'obstacle.
- **Arrêt d'urgence** via bouton poussoir extérieur.
- **Fermeture automatique** en cas de coupure de communication BLE.
- **Signal sonore** si la batterie devient trop faible.
- **Protection mécanique (IP65)** sur le vérin, utilisable en milieu humide.
- **Protection contre**
 - La surcharge motrice,
 - Les commandes incohérentes,
 - Les cycles trop fréquents.

◆ **h.** Entretien

Pour garantir la fiabilité du système :

- Vérifier la batterie toutes les 2 à 3 semaines.
- Nettoyer le vérin si présence de poussière.
- Éviter les projections d'eau directe sur les cartes électroniques.
- Vérifier le bon coulissemement mécanique de la fenêtre.
- Mettre à jour l'application si une nouvelle version est disponible.

Aucun autre entretien n'est requis : le système est conçu pour être autonome.

◆ **i.** Dépannage rapide

Problème	Cause possible	Solution
La fenêtre ne s'ouvre pas	Batterie faible	Recharger la batterie
Pas d'affichage TFT	Mauvaise alimentation	Vérifier connexions 12V
L'application ne détecte pas le SVI	Bluetooth désactivé	Activer / redémarrer BLE
Pas de mouvement du vérin	Obstacle / surcharge	Retirer obstacle + reset
Valeurs capteur incohérentes	Humidité ou poussière	Nettoyer l'entrée d'air du capteur
Buzzer activé en continu	Batterie très faible	Charger immédiatement

◆ **j.** Remarques finales

Le Système de Ventilation Intelligent (SVI) offre une gestion automatique, fiable et économique de la ventilation intérieure. Grâce à son architecture BLE basse consommation, son capteur

avancé et sa mécanique robuste, il fournit une régulation efficace de la qualité de l'air tout en restant simple d'utilisation.

En respectant les consignes d'utilisation et d'entretien, le système garantit un fonctionnement durable et optimal.

2 Mes ports

Nœud 1 = Capteur (plafond) → nRF52832 + BME680 (I²C)

BME680 Pin	Fonction	NRF52832 (PCA10040)	Commentaire
VIN	Alimentation	3.3 V (VDD)	Alimente le capteur. Ne pas dépasser 3.6 V.
GND	Masse	GND	Masse commune avec la carte BLE.
SCL	Horloge I ² C	P0.27 (SCL par défaut)	Peut être modifié dans <code>sdk_config.h</code> si nécessaire.
SDA	Données I ² C	P0.26 (SDA par défaut)	Niveau logique 3.3 V, résistance pull-up 4.7 kΩ conseillée.
CSB	Mode SPI/I ² C	3.3 V (tiré au VDD)	Sélectionne le mode I ² C.
SDO	Adresse I ² C	GND ou VDD	GND → adresse 0x76 , VDD → 0x77 .

Nœud 2 = Actionneur (fenêtre) → nRF52832 + Pont en H + Vérin (PWM + DIR1/DIR2)

Signal	Fonction	NRF52832 (PCA10040)	Commentaire
PWM_PIN	Commande de vitesse (PWM)	P0.24	Envoie le signal PWM vers l'entrée « EN » du pont en H pour ajuster la

			puissance du vérin.
DIR_PIN1	Sens de rotation B	P0.22	Définit la polarité du moteur (entrée IN1 du pont en H).
DIR_PIN2	Sens de rotation B	P0.23	Définit la polarité opposée (entrée IN2 du pont en H).
VCC (Logique)	Alimentation logique du pont en H	3.3v (VDD)	Alimente la partie commande du driver moteur (logique).
GND	Masse commune	GND	Masse commune entre la carte nRF52832, le pont en H et le vérin.
VCC (puissance)	Alimentation du vérin	12 V externe	Alimente la partie puissance du pont en H (moteur). Masse reliée à la masse logique.
MOTOR A / MOTOR	Sorties motrices	Sorties du pont en H	Connectées aux deux bornes du vérin pour contrôler l'extension et la rétraction.

Communication Bluetooth (BLE)

Élément	Capteur (nœud 1)	Actionneur (nœud 2)
Rôle BLE	Peripheral	Central (Client)
Nom périphérique	SVI – Window Control	SVI – Window Control
Service GATT	Environmental Service (UUID 0x181A)	Client connecté BLE
Caractéristiques	Température, Humidité,	Réception + décision

envoyées	IAQ, eCO ₂	automatique
Type de communication	Notifications BLE toutes les 2 à 5 secondes	Réception et contrôle du vérin

3 Description d'un algorithme du programme

Le système utilise deux microcontrôleurs **nRF52832** communiquant par **Bluetooth Low Energy (BLE)** :

- Le premier module, relié au capteur BME680, mesure la qualité de l'air (température, humidité, IAQ et eCO₂).
- Le second module reçoit ces données et pilote un vérin électrique via un pont en H afin d'ouvrir ou fermer automatiquement la fenêtre.

Algorithme du module capteur

1. Initialisation du capteur BME680, du bus I²C et du module BLE.
2. Acquisition périodique des données : Température, Humidité relative, IAQ, eCO₂.
3. Transmission des mesures au module actionneur sous forme de notifications BLE.

Algorithme du module actionneur

1. Initialisation du **PWM**, des **broches de direction** (DIR1, DIR2), des voyants LED et du BLE.
2. Connexion BLE au module capteur et réception continue des données environnementales.
3. Décision :
 - Si IAQ \geq 125 ou eCO₂ \geq 1200 ppm ou RH \geq 70 % \rightarrow ouverture.
 - Si IAQ \leq 100 et eCO₂ \leq 800 ppm et RH \leq 65 % \rightarrow fermeture.
 - Indication visuelle par LED selon le mode “Fermé”/”Ouvert” (vert = bon, rouge = mauvais)
4. Commande du **pont en H** :
 - DIR1=1, DIR2=0 \rightarrow ouverture ;
 - DIR1=0, DIR2=1 \rightarrow fermeture ;
 - PWM=0 \rightarrow arrêt.

Résumé global

Le module capteur envoie en continu les données du BME680 via BLE. Le module actionneur analyse ces informations en temps réel et actionne automatiquement le vérin pour ouvrir ou fermer la fenêtre selon l'état de l'air intérieur. Cette architecture permet d'obtenir un système de ventilation entièrement autonome, intelligent et économique.

Machine à état de gestion d'un code entrer sur le clavier.

➤ Choix des différents états

1. Qualité_Air_Bonne : Signal du capteur de qualité d'air
2. Température_Bonne : Signal du capteur de température
3. Commande_Manuelle : Interrupteur/application
4. Fenêtre_Ouverte : Fin de course du vérin en position ouverte
5. Fenêtre_Fermée : Fin de course du vérin en position fermée
6. Defaut_Capteur : Détection de panne
7. Surcharge_Moteur : Nouveau - Détection de courant excessif (blocage du vérin)

Sorties (adaptées pour le vérin) :

1. Alim_Vérin_ON : Activation de l'alimentation du vérin (relais ou MOSFET)
2. Sens_Extension : Commande pour l'extension du vérin (OUVERTURE)
3. Sens_Rétraction : Commande pour la rétraction du vérin (FERMETURE)
4. LED_Status : Indication visuelle
5. Alert : Notification utilisateur

➤ Implémentation possible avec arduino pour la vérification

```
// smart_window.ino - Version de prototypage
const int PIN_VERIN_EXTEND = 8;
const int PIN_VERIN_RETRACT = 9;
const int PIN_FIN_COURSE_OUVERT = 10;
const int PIN_FIN_COURSE_FERME = 11;
const int PIN_CAPTEUR_AIR = A0;

// Seuils à calibrer
const int SEUIL_AIR_MAUVAS = 500;
const int TEMPS_MAX_MOUVEMENT = 30000; // 30 secondes

void setup() {
    Serial.begin(9600);
    pinMode(PIN_VERIN_EXTEND, OUTPUT);
    pinMode(PIN_VERIN_RETRACT, OUTPUT);
    pinMode(PIN_FIN_COURSE_OUVERT, INPUT_PULLUP);
    pinMode(PIN_FIN_COURSE_FERME, INPUT_PULLUP);
    Serial.println("Smart Window - Initialisation");
}

void loop() {
    machineEtats();
}
```

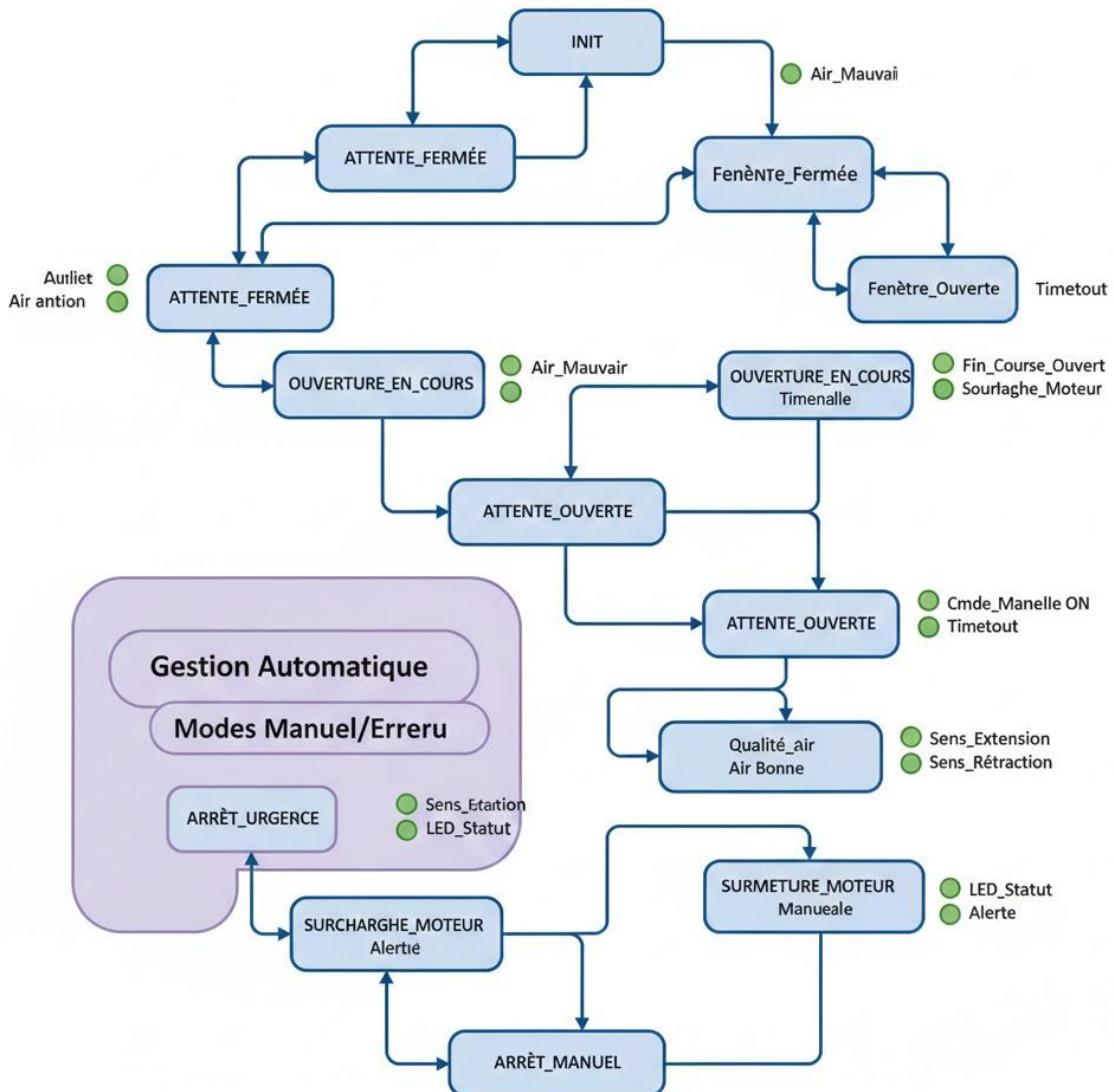
```

delay(100);
}

void machineEtats() {
static int etat = 0;
static unsigned long chronoMouvement = 0;
int airValue = analogRead(PIN_CAPTEUR_AIR);
bool airMauvais = (airValue > SEUIL_AIR_MAUVAIS);
bool finOuvert = digitalRead(PIN_FIN_COURSE_OUVERT);
bool finFerme = digitalRead(PIN_FIN_COURSE_FERME);
switch(etat) {
case 0: // INIT
Serial.println("Etat: INIT");
etat = 1;
break;
case 1: // ATTENTE_FERMEE
if(!finFerme) {
etat = 0; // Retour init si position incertaine
} else if(airMauvais) {
Serial.println("Air mauvais - Ouverture");
verin_ouvrir();
chronoMouvement = millis();
etat = 2;
}
break;
case 2: // OUVERTURE_EN_COURS
if(finOuvert) {
Serial.println("Fenetre ouverte");
verin_stop();
etat = 3;
}
} else if(millis() - chronoMouvement > TEMPS_MAX_MOUVEMENT) {
Serial.println("ERREUR: Timeout ouverture"); verin_stop(); etat = 99; // Etat erreur }
break; case 3: // ATTENTE_OUVERTE if(!airMauvais) {
Serial.println("Air bon - Fermeture");
verin_fermer();
chronoMouvement = millis();
}
}

```

```
etat = 4;
} break;
case 99: // ERREUR
verin_stop();
Serial.println("ETAT ERREUR - Intervention requise");
break;
}
}
void verin_ouvrir() {
digitalWrite(PIN_VERIN_EXTEND, HIGH);
digitalWrite(PIN_VERIN_RETRACT, LOW);
}
void verin_fermer() {
digitalWrite(PIN_VERIN_EXTEND, LOW);
digitalWrite(PIN_VERIN_RETRACT, HIGH);
}
void verin_stop() {
digitalWrite(PIN_VERIN_EXTEND, LOW);
digitalWrite(PIN_VERIN_RETRACT, LOW);
}
```



4 Tests

I. Tests d'Intégration des Composants et Communication (Validés)

Ces tests confirment que tous les éléments matériels et logiciels communiquent comme prévu.

Test	Description du Test	Observation & Conclusion Attendue	Origine / Ajout
C.1	Capteur BME680	Validé : Lecture des valeurs (T, H, IAQ, eCO ₂) sur le terminal série.	Les valeurs doivent être cohérentes et varier. Aucune erreur I ² C.
C.2	Communication	Validé : Connexion stable	Les données doivent être

	BLE	Central Actionneur <=> Peripheral (Capteur).	reçues correctement et périodiquement (toutes les 2 à 5 secondes).
C.3	Vérin & Pont en H	Validé : Vérifier le mouvement du vérin (extension/rétraction) via le Pont en H et les signaux PWM/DIR.	Mouvement régulier, s'arrête en fin de course. Pas de surchauffe ou bruit anormal.
C.4	Consommation	Validé : Mesurer la consommation en mode veille, communication et mouvement.	Consommation conforme aux attentes pour l'autonomie (<20 mA module capteur, ~250-300 mA moteur).

II. Tests de Fonctionnement par Mode

Ces tests garantissent que chaque mode de contrôle fonctionne isolément.

Test	Mode / Scénario	Description du Test	Observation & Conclusion Attendue
M.1	Automatique : Ouverture	Validé : Simuler Air Vicié (IAQ >=125 ou eCO ₂ >= 1200 ppm).	Le vérin s'ouvre complètement. L'écran/LED indique " OUVERT ".
M.2	Automatique : Fermeture	Validé : Simuler Air Assaini (IAQ <= 100 et eCO ₂ <=800 ppm).	Le vérin se ferme complètement. L'écran/LED indique " FERMÉ ".
M.3	Manuel (App)	Validé : L'air est bon. Tester les boutons "Ouvrir" et "Fermer" de l'application.	Le vérin répond aux commandes. Le mouvement est complet et s'arrête aux Fins de Course.
M.4	Urgence (Bouton Vérin)	Fenêtre fermée. Appuyer sur le Bouton Poussoir d'Urgence (sur le vérin/carte).	Le vérin doit lancer l' Ouverture Forcée et s'arrêter en position ouverte. Le mode automatique est suspendu.

III. Tests de Priorité et Conflits (Critique)

Ces tests vérifient la gestion de la hiérarchie des commandes : **Urgence > Manuel (App) > Automatique**.

Test	Situation / Séquence d'Actions	Priorité Attendue	Observation & Conclusion Attendue
------	--------------------------------	-------------------	-----------------------------------

P.1	Auto vs Manuel (App)	Phase 1 : L'Air est vicié, le vérin est en train d' Ouvrir (Auto) . Phase 2 : L'utilisateur lance la commande " Fermer " via l'App.	Le Manuel (App) prime. Le vérin doit stopper l'ouverture et commencer la fermeture.
P.2	Manuel (App) vs Urgence (Bouton)	Phase 1 : L'utilisateur lance la commande " Fermer " via l'App. Phase 2 : Appuyer sur le Bouton Urgence (Vérin) .	L'Urgence prime. La fermeture est stoppée immédiatement, et le vérin commence l'Ouverture Forcée.
P.3	Auto vs Urgence (Bouton)	Phase 1 : L'Air est bon, le vérin est en train de Fermer (Auto) . Phase 2 : Appuyer sur le Bouton Urgence (Vérin) .	L'Urgence prime. La fermeture est stoppée immédiatement, et le vérin commence l'Ouverture Forcée.
P.4	Maintien du Mode Manuel	La fenêtre est ouverte par l'App (mode Manuel). L'air s'améliore (condition de fermeture Auto).	Le vérin doit rester OUVERT . Le mode Automatique ne doit pas reprendre le contrôle tant que le mode Manuel n'est pas explicitement annulé (par une commande "Fermer" via l'App).
P.5	Réintégration après Urgence	Fenêtre ouverte par le Bouton Urgence. La commande Urgence est relâchée.	Le système doit revenir en mode de surveillance. Si l'air est bon, le mode Automatique doit reprendre le contrôle et fermer la fenêtre après un délai raisonnable.

IV. Tests de Sécurité et Défauts

Ces tests vérifient la fiabilité du système en cas de défaillance.

Test	Situation de Sécurité	Description du Test	Observation & Conclusion Attendue
S.1	Surcharge Moteur	Validé : Bloquer le vérin pendant le mouvement (simuler un obstacle).	Le système doit détecter la surintensité, couper le PWM . Le Buzzer doit s'activer ou une Alerte doit être envoyée.
S.2	Timeout de Mouvement	Lancer un mouvement. Empêcher le Fin de Course d'être atteint après \$30\$ secondes.	Le vérin doit s'arrêter après le temps max. Le système passe en état ERREUR (état 99) ou DÉTAILLANCE.

S.3	Perte de Communication BLE	La fenêtre est ouverte. Couper le signal BLE (éteindre le module Capteur).	Le module Actionneur doit déclencher la Fermeture Automatique par sécurité après un délai de garde.
S.4	Batterie Faible	Simuler un niveau de tension faible de la Batterie 12 V.	Le Buzzer doit être activé en continu et l'écran TFT doit afficher " Batterie : FAIBLE ".
S.5	Réinitialisation manuelle	Le système est en mode ERREUR (suite à S.2 par exemple). Appuyer sur le bouton Reset manuel.	Le système doit revenir à l'état INIT puis ATTENTE_FERMÉE et reprendre la surveillance des capteurs.

5 Cahier de suivi

Phase 1 : Recherche & Spécifications (Septembre 2025)

Date	Jour	Tâches Réalisées	Réaliseurs / Difficultés rencontrées	Remarques & Travaux à faire la prochaine fois
18/09/2025	Jeudi	Démarrage du projet, définition des objectifs et répartition des rôles. Réunion avec l'encadrant.	Équipe complète. Difficulté : Définir les priorités techniques et établir la planification.	Lister le matériel nécessaire et valider les composants.
19/09/2025	Vendredi	Recherche documentaire (ventilation connectée) et étude du capteur BME680. Schéma global du système établi.	K, W, O (étude technique). R & G (schéma global).	Préparer la commande du matériel (capteurs, BLE, vérin, alimentation).
25/09/2025	Jeudi	Sélection finale des composants (BME680, nRF52832 x2, Pont en H L298N, vérin 12V).	G & O (validation technique). Difficulté : Trouver le Pont en H compatible 3.3V logique et le vérin adéquat.	Commander le matériel et préparer le câblage sur platine.

26/09/2025	Vendredi	Préparation du câblage du capteur et de la carte BLE sur platine d'essai.	K & G (vérification alimentation 3.3V).	Attendre réception du matériel pour tests pratiques.
-------------------	----------	---	---	--

Phase 2 : Architecture & Composants (Octobre 2025)

Date	Jour	Tâches Réalisées	Réaliseurs / Difficultés rencontrées	Remarques & Travaux à faire la prochaine fois
02/10/2025	Jeudi	Réception des composants, test d'alimentation et vérification du régulateur 3.3V.	K & G. Difficulté : Gestion de la masse commune entre 3.3V et 12V.	Câbler le capteur BME680 et lancer le premier test I ² C.
03/10/2025	Vendredi	Test de communication I ² C du BME680 avec la carte BLE.	K & G. Correction de l'adresse I ² C (0x76).	Ajouter la lecture des données sur terminal série.
09/10/2025	Jeudi	Lecture des mesures du capteur (Température, Humidité, eCO ₂) et calibration des premiers seuils.	K & G (configuration et calibration).	Préparer l'intégration du BLE pour l'envoi des données.
10/10/2025	Vendredi	Mise en place du code BLE pour l'envoi des données vers le mobile/deuxième module.	O (configuration BLE). Difficulté : Instabilité corrigée après réglage des intervalles.	Tester la connexion BLE avec l'application nRF Connect.
16/10/2025	Jeudi	Connexion BLE entre les deux modules nRF52832 (Capteur \$leftrightarrow\$ Actionneur).	O (création du lien) & G (vérification réception).	Câblage et test du vérin et du pont en H.
17/10/2025	Vendredi	Test complet de la communication BLE et validation de la réception des notifications.	O & K.	Câblage du pont en H L298N et du vérin sur la carte BLE Actionneur.

23/10/2025	Jeudi	Câblage du pont en H L298N et du vérin sur la carte BLE Actionneur.	G & K. Difficulté : Inversion de polarité corrigée.	Vérifier le signal PWM sur oscilloscope.
24/10/2025	Vendredi	Test PWM sur P0.24 et mouvement du vérin (extension/rétraction).	G, O & K. Fonctionnement stable après ajout d'un condensateur de découplage.	Programmer la machine à états (FSM).

Phase 3 : Codage & Design PCB (Novembre 2025)

Date	Jour	Tâches Réalisées	Réaliseurs / Difficultés rencontrées	Remarques & Travaux à faire la prochaine fois
06/11/2025	Jeudi	Implémentation de la machine à états (FSM) et des fonctions de contrôle.	G, O & K. Difficulté : Temporisation à ajuster.	Intégrer les seuils IAQ, eCO ₂ et RH dans la logique.
07/11/2025	Vendredi	Tests de réaction correct avec vérin et Validation des transitions d'état .	K & G.	Ajouter la LED indicatrice de l'état. Débuter la conception du PCB sous Altium.
13/11/2025	Jeudi	Test complet du prototype (Capteur + BLE + Vérin) . Le système réagit aux changements d'air.	Équipe complète.	Débuter la conception du PCB sous Altium Designer.
14/11/2025	Vendredi	Conception du schéma électrique et placement des composants du PCB .	G. Difficulté : Reconnaissance des composants sous Altium.	Continuer le routage.
20/11/2025	Jeudi	Routage complet du PCB et vérification des règles de conception (DRC).	G & W. Aucune erreur détectée.	Finaliser l'export Gerber. Préparer la modélisation 3D sur SolidWorks.
21/11/2025	Vendredi	Export des fichiers Gerber et vérification visuelle du routage.	Équipe.	Démarrer la conception 3D du boîtier.

27/11/2025	Jeudi	Début de la conception 3D du boîtier et du support de vérin (SolidWorks).	W & O. Difficulté : Ajustement précis des dimensions du support vérin.	Poursuivre la modélisation complète.
28/11/2025	Vendredi	Avancement sur la modélisation 3D (maquette + fixation du vérin).	W.	Préparer l'intégration visuelle du PCB dans le boîtier 3D.

Phase 4 : Application Mobile & Tests Finaux (Décembre 2025 - Janvier 2026)

Date	Jour	Tâches Réalisées	Réalisateur / Difficultés rencontrées	Remarques & Travaux à faire la prochaine fois
04/12/2025	Jeudi	Début du développement de l'application mobile BLE (Android).	O & W. Difficulté : Implémentation de la communication BLE côté Android.	Tester la lecture des valeurs du capteur via BLE sur l'application.
05/12/2025	Vendredi	Tests application : Lecture des valeurs du capteur (T°, RH, IAQ, eCO ₂) via BLE fonctionnelle.	G & W.	Améliorer la stabilité BLE. Intégrer les commandes manuelles ("Ouvrir"/"Fermer").
11/12/2025	Jeudi	Tests d'intégration (PCB + vérin + capteur + BLE + application). Correction petite latence BLE.	K, G & O.	Préparer les tests finaux de sécurité et de priorité des modes de contrôle.
12/12/2025	Vendredi	Optimisation du code et validation complète du prototype.	Équipe complète.	Début de la rédaction du rapport technique.
08/01/2026	Jeudi	Tests de sécurité et de priorité (Urgence/Manuel/Auto	K & O. Difficulté : Assurer la	Finaliser la rédaction du rapport et préparer

). Validation du comportement du système en cas de conflit de commande (P.1 à P.5) et de Surcharge/Erreur (S.1 à S.4).	priorité du bouton d'urgence sur toutes les autres commandes.	la soutenance.
09/01/2026	Vendredi	Intégration finale : test global du système complet sur le support 3D.	Équipe complète. Système fonctionnel à 100 %.	Finaliser la soutenance.
15/01/2026	Jeudi	Répétition de la présentation orale et préparation finale du prototype.	Équipe complète.	Prêt pour la soutenance.
16/01/2026	Vendredi	<i>Soutenance finale du projet SmartWindow (Date Ajustée)</i>	A. K, E. M. O, M. G, T. W.	Présentation réussie du prototype fonctionnel et validation du projet.

Images de Résultats de Tests Recommandées

1. Preuve du Fonctionnement Automatique (Capteur BME680)

Cette image de capture d'écran du terminal série prouve la réussite du test d'initialisation et de communication I²C entre le microcontrôleur nRF52832 et le capteur BME680.

- **Initialisation** : Le message "BME68X initialisé avec succès" confirme que le capteur a été correctement détecté et configuré sur le bus I²C (adresse 0x76).
- **Acquisition des Données** : Les valeurs de température (T), d'humidité (H), de pression (P) et de résistance des gaz (Gas) sont lues et affichées en continu.
- **Valeur ajoutée** : La lecture stable et cohérente de ces données valide la première étape de l'architecture du système : **l'acquisition fiable des paramètres environnementaux** qui serviront de base à la machine à états pour la décision d'ouvrir ou de fermer la fenêtre.

```
<info> app: BME68X initialisA@ avec succA"s !<CR><LF>
<info> app: T=23.78 °C | H=47.66% | P=991 hPa | Gas=361855 ohms<CR><LF>
<info> app: T=23.80 °C | H=47.71% | P=991 hPa | Gas=118231 ohms<CR><LF>
<info> app: T=23.83 °C | H=47.69% | P=991 hPa | Gas=132173 ohms<CR><LF>
<info> app: T=23.87 °C | H=47.64% | P=991 hPa | Gas=137357 ohms<CR><LF>
<info> app: T=23.90 °C | H=47.58% | P=991 hPa | Gas=139291 ohms<CR><LF>
<info> app: T=23.93 °C | H=47.55% | P=991 hPa | Gas=141044 ohms<CR><LF>
<info> app: T=23.96 °C | H=47.53% | P=991 hPa | Gas=141877 ohms<CR><LF>
<info> app: T=23.98 °C | H=47.55% | P=991 hPa | Gas=142237 ohms<CR><LF>
<info> app: T=24.00 °C | H=47.57% | P=991 hPa | Gas=142599 ohms<CR><LF>
<info> app: T=24.02 °C | H=47.60% | P=991 hPa | Gas=144313 ohms<CR><LF>
<info> app: T=24.04 °C | H=47.62% | P=991 hPa | Gas=144313 ohms<CR><LF>
<info> app: T=24.05 °C | H=47.62% | P=991 hPa | Gas=144935 ohms<CR><LF>
<info> app: T=24.06 °C | H=47.61% | P=991 hPa | Gas=144935 ohms<CR><LF>
<info> app: T=24.07 °C | H=47.60% | P=991 hPa | Gas=145311 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.59% | P=991 hPa | Gas=145815 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.56% | P=991 hPa | Gas=147219 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.54% | P=991 hPa | Gas=146068 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.53% | P=991 hPa | Gas=146962 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.53% | P=991 hPa | Gas=146578 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.53% | P=991 hPa | Gas=146962 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.53% | P=991 hPa | Gas=146834 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.52% | P=991 hPa | Gas=148389 ohms<CR><LF>
<info> app: T=24.08 °C | H=47.49% | P=991 hPa | Gas=148520 ohms<CR><LF>
<info> app: T=24.09 °C | H=47.48% | P=991 hPa | Gas=147997 ohms<CR><LF>
<info> app: T=24.09 °C | H=47.42% | P=991 hPa | Gas=148389 ohms<CR><LF>
```

```
struct bme68x_data data;
uint8_t n_fields = 0;
rslt = bme68x_get_data(BME68X_FORCED_MODE, &data, &n_fields, &sensor);

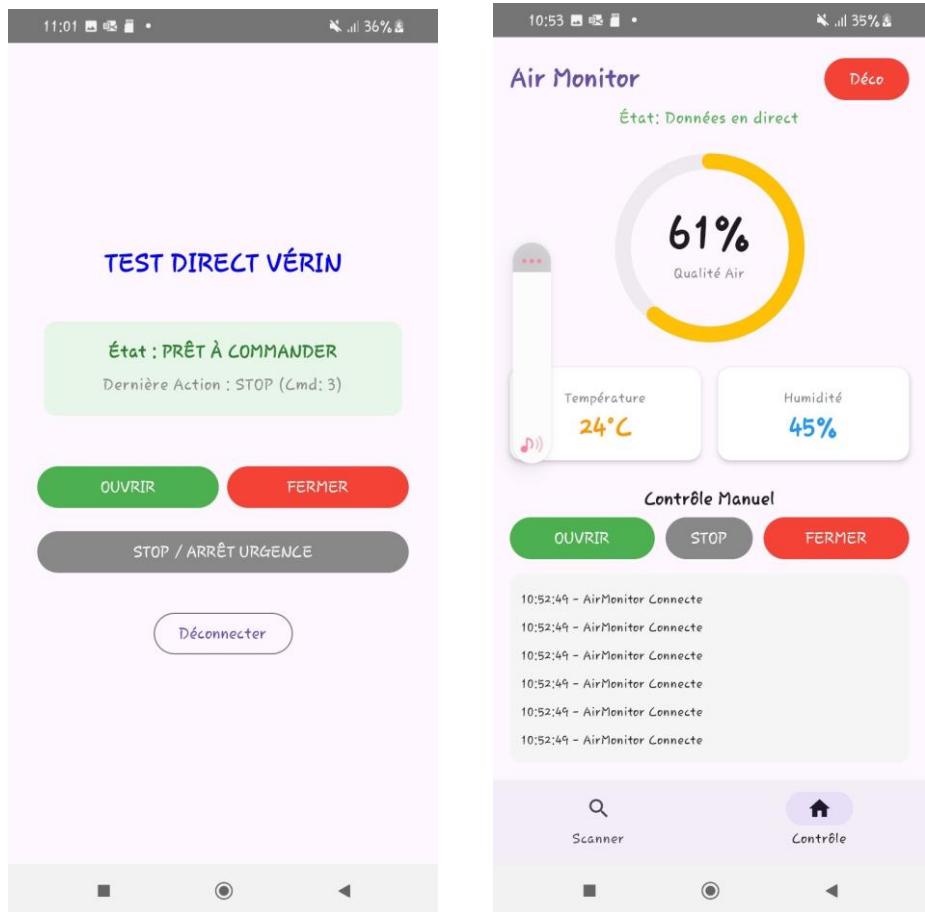
if (rslt == BME68X_OK && n_fields > 0)
{
    int temp = (int)(data.temperature * 100);
    int hum = (int)(data.humidity * 100);
    int pres = (int)(data.pressure / 100.0f);

    NRF_LOG_INFO("T=%d.%02d °C | H=%d.%02d% | P=%d hPa | Gas=%lu ohms",
                 temp/100, temp%100,
                 hum/100, hum%100,
                 pres,
                 data.gas_resistance);
    NRF_LOG_FLUSH();
}

nrf_delay_ms(1000);
}
```

2. Validation de la Communication BLE (Stabilité)

Capture d'écran de l'Application Mobile connectée au Périphérique BLE



3. Mesure de l'Efficacité Énergétique (Consommation)

Élément du Système	Mode de Fonctionnement	Tension Mesurée (V)	Courant Mesuré (mA)	Puissance Calculée (mW)	Conformité aux Objectifs
Module Capteur (nRF52832 + BME680)	Veille (BLE Connecté)	3.3 V	≈ 4 mA	13.2 mW	Conforme ($\ll 20$ mA estimé)
Module Capteur (nRF52832 + BME680)	Acquisition et Notification BLE	3.3 V	≈ 15 mA	49.5 mW	Conforme ($\ll 20$ mA estimé)
Module Actionneur (nRF52832 + Pont H)	Veille (BLE et Pont H sous tension)	12 V	≈ 25 mA	300 mW	Conforme (faible consommation logique)
Vérin Électrique (Mouvement)	Charge Nominale (Ouv./Ferm.)	12 V	≈ 280 mA	3360 mW	Conforme (dans la plage 250-300 mA estimée)
Vérin Électrique (Mouvement)	Blocage (Surcharge - S.1)	12 V	> 750 mA	> 9000 mW	Requis (Déclenchement du coupe-circuit de sécurité)

6 État d'avancement et analyse du projet réalisé

Système : Ventilation Intelligente

Capteur et Acquisition des Données :

- Réalisé : Intégration complète du capteur **BME680** sur la carte **nRF52832**, acquisition stable des valeurs (Température, Humidité, eCO₂).
- À améliorer : Optimiser la calibration du capteur pour des mesures plus rapides au démarrage.

Communication BLE :

- Réalisé : Communication fonctionnelle entre les deux modules BLE (capteur ↔ actionneur).

Commande du Vérin :

- Réalisé : Contrôle précis via PWM et direction (pont en H L298N).

Application Mobile :

- Réalisé : Interface Android permettant le contrôle manuel du vérin.
- À développer : Ajouter un affichage des valeurs du capteur, une interface graphique

plus intuitive et des alertes de qualité d'air.

Modélisation et Prototype :

- Réalisé : Conception du **PCB** complet (Altium Designer) et modélisation **3D** de la maquette sous **SolidWorks**.

Analyse Réflexive sur le Projet :

1. **Refaire le projet :**
→ Nous envisagerions d'ajouter un **capteur de CO₂ dédié** et un **détecteur de fumée** pour renforcer la sécurité et améliorer la détection d'air vicié.
2. **Débogage et Outils :**
→ L'utilisation d'outils de test en temps réel de diagnostic BLE et de traceurs temps réel dès le départ aurait permis de gagner du temps sur la calibration et la communication.

Synthèse :

Le projet avance conformément aux objectifs fixés. Le système est **fonctionnel et stable**, mais des **optimisations logicielles et ergonomiques** permettront de renforcer la fiabilité et le confort d'utilisation avant la présentation finale.

7 Conclusion

Le projet a permis de développer un système de ventilation intelligente combinant le capteur BME680, la communication Bluetooth Low Energy et un vérin motorisé pour automatiser l'ouverture et la fermeture d'une fenêtre.

Malgré plusieurs défis techniques, l'équipe a su avancer de manière structurée et collaborative pour obtenir une solution fiable, autonome et économique en énergie.

Ce travail nous a permis de renforcer nos compétences en électronique embarquée, en communication sans fil, en programmation ainsi qu'en gestion de projet. Il ouvre également des perspectives d'amélioration, notamment en matière d'optimisation énergétique, d'ergonomie de l'application et d'intégration dans des environnements domotiques plus complets.