

Rapport de Projet : Conception, Mise en Œuvre et Validation des Capteurs de Polluants pour le Projet "Smart Territory"

Nicola Peuteau

Sommaire

1. Introduction.....	2
2. Contexte et Enjeux.....	2
3. Cahier des Charges.....	3
4. Analyse et Conception	5
Choix des Capteurs.....	5
Architecture Technique	6
5. Implémentation	6
Matériel Utilisé.....	6
Logiciel Utilisé.....	6
Gestion des Erreurs.....	10
Fréquence des Mesures	11
6. Tests et Validation	11
Méthode de Test	11
Conditions de Test.....	11
Résultats des Tests	11
Intégration des Fiches de Test	11
7. Intégration et Déploiement	16
Communication avec la Plateforme.....	16
Alimentation	16
Proposition de Déploiement	16
8. Défis Rencontrés et Solutions Apportées.....	16
9. Conclusion.....	17
Résumé.....	17
10. Annexes.....	17

1. Introduction

Dans le cadre du projet ambitieux intitulé "Smart Territory", qui a été initié en partenariat avec l'ESAIP, l'objectif principal est de concevoir et de déployer une infrastructure IoT complète et performante visant à surveiller en temps réel la pollution de l'air et de l'eau dans sept communes du département de Maine-et-Loire. En tant qu'étudiant n°1, ma responsabilité spécifique au sein de cette initiative repose sur la conception, la mise en œuvre et la validation d'un système de capteurs dédiés à la mesure des polluants atmosphériques. Ces capteurs, soigneusement sélectionnés et intégrés, ont pour mission de quantifier avec une précision optimale les concentrations de dioxyde de soufre (SO_2), de dioxyde d'azote (NO_2), d'ozone (O_3), ainsi que des particules fines en suspension, notamment les $\text{PM}_{2.5}$ et les PM_{10} . Ce système, une fois pleinement opérationnel, permettra de collecter des données fiables et exploitables, lesquelles seront mises à la disposition des citoyens et des autorités locales afin de répondre aux exigences strictes de la loi LAURE sur la qualité de l'air, tout en contribuant à une meilleure compréhension des enjeux environnementaux dans la région.

2. Contexte et Enjeux

La pollution atmosphérique représente aujourd'hui une problématique environnementale et sanitaire de premier ordre, dont les impacts délétères sur la santé publique ne cessent de croître, notamment à travers l'aggravation des maladies respiratoires telles que l'asthme ou la bronchite chronique, ainsi que des pathologies cardiovasculaires graves. Les polluants ciblés dans ce projet, à savoir le SO_2 , le NO_2 , l' O_3 , les $\text{PM}_{2.5}$ et les PM_{10} , sont particulièrement surveillés en raison de leurs effets néfastes sur la santé humaine et l'écosystème. À titre d'exemple, l'Organisation Mondiale de la Santé (OMS) recommande des seuils stricts pour ces polluants, comme une moyenne annuelle de $5 \mu\text{g}/\text{m}^3$ pour les $\text{PM}_{2.5}$ et de $10 \mu\text{g}/\text{m}^3$ pour le NO_2 , afin de limiter les risques sanitaires. Mon travail s'inscrit donc dans une démarche plus large visant à mettre en place une surveillance continue et en temps réel de la qualité de l'air, grâce à une infrastructure IoT robuste et innovante, contribuant ainsi à la réduction des impacts environnementaux et à l'amélioration de la qualité de vie des habitants des communes concernées.

3. Cahier des Charges

Ma mission principale, désignée sous l'intitulé "Fp1", consiste à concevoir un système capable de mesurer en continu les concentrations des polluants atmosphériques mentionnés ci-dessus, avec un niveau de précision et de fiabilité conforme aux standards scientifiques et réglementaires. Parmi les contraintes techniques et organisationnelles imposées, on retrouve notamment l'obligation de connecter les capteurs à un microcontrôleur ESP32 via le protocole de communication I2C, ce qui garantit une intégration fluide et une gestion efficace des données. Le projet devait être réalisé dans un délai total de 150 heures, ce qui a nécessité une planification rigoureuse des différentes étapes, de la conception initiale jusqu'aux tests finaux. Par ailleurs, aucune enveloppe budgétaire spécifique n'a été allouée, ce qui m'a conduit à privilégier des solutions techniques abordables sans compromettre la qualité des mesures. Les critères d'évaluation de la réussite de cette mission reposent principalement sur la capacité du système à fournir des mesures précises par rapport aux seuils recommandés par l'OMS, tout en offrant une flexibilité de niveau F1, c'est-à-dire une adaptabilité négociable en fonction des besoins futurs ou des contraintes imprévues.

Norme ATMO France et Calcul de l'Indice de Qualité de l'Air (IQA)

Présentation de l'Organisme

ATMO France est l'organisme national qui fédère les Associations Agréées de Surveillance de la Qualité de l'Air (AASQA) sur l'ensemble du territoire français. Sous la supervision du ministère de la Transition écologique, cette structure a pour mission de coordonner la surveillance réglementaire de la qualité de l'air, de définir les méthodologies d'évaluation et de garantir l'harmonisation des données environnementales recueillies à l'échelle nationale. Grâce à cette coordination, les données sur la pollution atmosphérique sont cohérentes, fiables et comparables, quel que soit le lieu de mesure.

Polluants Pris en Compte

L'indice de qualité de l'air (IQA) tel que défini par ATMO France repose sur la mesure des polluants atmosphériques les plus significatifs pour la santé humaine et l'environnement. Il prend en compte six principaux polluants :

- Le dioxyde de soufre (SO_2)
- Le dioxyde d'azote (NO_2)
- L'ozone troposphérique (O_3)

- Les particules en suspension PM10
- Les particules fines PM2.5

Chaque polluant est associé à des seuils réglementaires définis par la législation française et européenne. À titre d'exemple, la valeur limite annuelle du NO₂ est de 40 µg/m³, tandis que la moyenne journalière maximale recommandée pour les PM10 est de 50 µg/m³, à ne pas dépasser plus de 35 jours par an.

		Bon	Moyen	Dégradé	Mauvais	Très mauvais	Extrêmement mauvais
Moyenne journalière	PM2.5	0-10	10-20	20-25	25-50	50-75	> 75
Moyenne journalière	PM10	0-20	20-40	40-50	50-100	100-150	> 150
Max Horaire Journalier	NO2	0-40	40-90	90-120	120-230	230-340	> 340
Max Horaire Journalier	O3	0-50	50-100	100-130	130-240	240-380	> 380
Max Horaire Journalier	SO2	0-100	100-200	200-350	350-500	500-750	> 750

(image de <https://www.airparif.fr/indice-atmo>)

Méthode de Calcul de l'IQA

L'indice de qualité de l'air est calculé quotidiennement selon la méthodologie officielle d'ATMO France. Pour chaque polluant mesuré, un sous-indice est calculé en fonction de la concentration enregistrée et des seuils de qualité définis. Les règles de calcul sont les suivantes :

- Pour les PM10 et PM2.5 : moyenne journalière des concentrations (en µg/m³).
- Pour le NO₂, le SO₂ et l'O₃ : concentration horaire maximale de la journée.

L'indice global de la qualité de l'air correspond au **sous-indice le plus élevé** parmi les polluants mesurés. Il est exprimé sur une échelle de 1 (très bon) à 10 (très mauvais), ce qui permet une lecture simplifiée mais rigoureuse de la pollution atmosphérique à un instant donné.

Importance d'une Référence Nationale

Se baser sur les méthodes et les seuils définis par ATMO France permet de garantir l'uniformité et la validité scientifique des résultats obtenus. Cela renforce la légitimité des données produites dans le cadre du projet, et assure leur compatibilité avec les dispositifs de surveillance officiels. De plus, cette conformité favorise l'interprétation correcte des résultats par les décideurs publics et les citoyens, dans une logique de transparence et de santé publique.

4. Analyse et Conception

Choix des Capteurs

Pour répondre aux exigences du projet, j'ai opté pour une gamme de capteurs développés par DFRobot, reconnus pour leur fiabilité, leur précision et leur compatibilité avec le protocole I2C, qui constitue un élément clé de l'architecture technique. Les capteurs sélectionnés sont les suivants :

SEN0470 (SO₂) : Ce capteur offre une plage de mesure allant de 0 à 20 ppm avec une résolution fine de 0,1 ppm, ce qui le rend idéal pour détecter les variations subtiles des concentrations de dioxyde de soufre dans l'atmosphère.

SEN0471 (NO₂) : Également doté d'une plage de mesure de 0 à 20 ppm et d'une résolution de 0,1 ppm, ce capteur est parfaitement adapté à la surveillance du dioxyde d'azote, un polluant particulièrement présent dans les zones urbaines ou industrielles.

SEN0472 (O₃) : Avec une plage de mesure de 0 à 10 ppm et une résolution de 0,01 ppm, ce capteur permet une détection précise de l'ozone, un gaz dont les concentrations peuvent fluctuer rapidement en fonction des conditions météorologiques.

SEN0460 (PM2.5/PM10) : Ce capteur, capable de mesurer les particules fines dans une plage de 0 à 999 $\mu\text{g}/\text{m}^3$ avec une précision de $\pm 10 \mu\text{g}/\text{m}^3$, est essentiel pour évaluer la présence de PM2.5 et PM10, deux indicateurs critiques de la qualité de l'air.

Les capteurs utilisés dans le cadre du projet "Smart Territory" ont été spécifiquement choisis pour être compatibles avec la norme ATMO France. Ils permettent de mesurer en continu les concentrations des principaux polluants retenus pour le calcul de l'IQA.

Ces capteurs électrochimiques et optiques fournissent des mesures numériques précises qui peuvent être directement exploitées pour calculer les sous-indices conformément aux

prescriptions d'ATMO France. Les données collectées par l'ESP32 sont converties en concentrations (en $\mu\text{g}/\text{m}^3$ ou ppm) puis comparées aux seuils réglementaires pour générer l'indice final.

Architecture Technique

L'architecture du système repose sur l'intégration des capteurs à un microcontrôleur ESP32, choisi pour sa capacité à gérer plusieurs bus I2C simultanément et pour sa connectivité Wi-Fi intégrée, qui facilitera ultérieurement la transmission des données. Chaque capteur est connecté au bus I2C avec une adresse spécifique, garantissant une communication sans interférence :

SO₂ : Adresse I2C 0x68

NO₂ : Adresse I2C 0x69

O₃ : Adresse I2C 0x74

PM2.5/PM10 : Adresse I2C 0x19

Cette configuration permet une collecte centralisée et synchronisée des données, tout en offrant une base solide pour une éventuelle extension du système.

5. Implémentation

Matériel Utilisé

Le système a été construit autour des éléments matériels suivants :

Capteurs : SEN0470 (SO₂), SEN0471 (NO₂), SEN0472 (O₃), SEN0460 (PM2.5/PM10).

Microcontrôleur : ESP32, un composant polyvalent et performant, largement utilisé dans les projets IoT pour sa robustesse et ses capacités avancées.

Logiciel Utilisé

Le développement logiciel a été réalisé à l'aide de l'IDE Arduino, un environnement bien adapté à la programmation de microcontrôleurs comme l'ESP32. Le code mis en place initialise les capteurs, configure leur mode de fonctionnement, lit les données à une fréquence d'une mesure par seconde, et les affiche sur le moniteur série pour un suivi en temps réel. Pour ce faire, j'ai utilisé les bibliothèques officielles de DFRobot, notamment

DFRobot_MultiGasSensor pour les capteurs de gaz et DFRobot_AirQualitySensor pour le capteur de particules. Voici une version complète et détaillée du code implémenté :

```
#include "DFRobot_MultiGasSensor.h"
```

```
#include "DFRobot_AirQualitySensor.h"
```

```
// Définition des adresses I2C spécifiques à chaque capteur pour éviter tout conflit
```

```
#define I2C_ADDRESS_SO2 0x68
```

```
#define I2C_ADDRESS_NO2 0x69
```

```
#define I2C_ADDRESS_O3 0x74
```

```
#define I2C_ADDRESS_PM 0x19
```

```
// Création des instances des capteurs avec leurs adresses respectives
```

```
DFRobot_GAS_I2C gasSO2(&Wire, I2C_ADDRESS_SO2);
```

```
DFRobot_GAS_I2C gasNO2(&Wire, I2C_ADDRESS_NO2);
```

```
DFRobot_GAS_I2C gasO3(&Wire, I2C_ADDRESS_O3);
```

```
DFRobot_AirQualitySensor pmSensor(&Wire, I2C_ADDRESS_PM);
```

```
void setup() {
```

```
    // Initialisation de la communication série à une vitesse de 115200 bauds
```

```
    Serial.begin(115200);
```

```
    delay(1000); // Pause initiale pour stabiliser le système
```

```
    // Initialisation et vérification de la connexion du capteur SO2
```

```
    while (!gasSO2.begin()) {
```

```
        Serial.println("Erreur : Le capteur de SO2 n'a pas été détecté correctement !");
```

```
    delay(1000);  
}  
  
gasSO2.changeAcquireMode(gasSO2.PASSIVITY); // Mode passif pour des mesures  
contrôlées  
  
gasSO2.setTempCompensation(gasSO2.ON); // Compensation de température activée  
  
// Initialisation et vérification de la connexion du capteur NO2  
while (!gasNO2.begin()) {  
    Serial.println("Erreur : Le capteur de NO2 n'a pas été détecté correctement !");  
    delay(1000);  
}  
  
gasNO2.changeAcquireMode(gasNO2.PASSIVITY);  
gasNO2.setTempCompensation(gasNO2.ON);  
  
// Initialisation et vérification de la connexion du capteur O3  
while (!gasO3.begin()) {  
    Serial.println("Erreur : Le capteur d'O3 n'a pas été détecté correctement !");  
    delay(1000);  
}  
  
gasO3.changeAcquireMode(gasO3.PASSIVITY);  
gasO3.setTempCompensation(gasO3.ON);  
  
// Initialisation et vérification de la connexion du capteur PM2.5/PM10  
while (!pmSensor.begin()) {  
    Serial.println("Erreur : Le capteur de particules PM n'a pas été détecté correctement !");
```



```
    delay(1000);  
}  
  
Serial.println("Succès : Tous les capteurs ont été détectés et initialisés avec succès !");  
}
```

```
void loop() {  
  
    // Lecture et affichage des données du capteur SO2  
  
    Serial.print("[SO2] Type de gaz détecté : ");  
    Serial.print(gasSO2.queryGasType());  
    Serial.print(" | Concentration mesurée : ");  
    Serial.print(gasSO2.readGasConcentrationPPM());  
    Serial.println(" ppm");  
  
    // Lecture et affichage des données du capteur NO2  
  
    Serial.print("[NO2] Type de gaz détecté : ");  
    Serial.print(gasNO2.queryGasType());  
    Serial.print(" | Concentration mesurée : ");  
    Serial.print(gasNO2.readGasConcentrationPPM());  
    Serial.println(" ppm");  
  
    // Lecture et affichage des données du capteur O3  
  
    Serial.print("[O3] Type de gaz détecté : ");  
    Serial.print(gasO3.queryGasType());  
    Serial.print(" | Concentration mesurée : ");  
    Serial.print(gasO3.readGasConcentrationPPM());
```

```
Serial.println(" ppm");

// Lecture et affichage des concentrations de PM2.5 et PM10

uint16_t PM2_5 =
pmSensor.gainParticleConcentration_ugm3(PARTICLE_PM2_5_ATMOSPHERE);

uint16_t PM10 =
pmSensor.gainParticleConcentration_ugm3(PARTICLE_PM10_ATMOSPHERE);

Serial.print("[PM2.5] Concentration mesurée : ");
Serial.print(PM2_5);
Serial.println(" µg/m³");

Serial.print("[PM10] Concentration mesurée : ");
Serial.print(PM10);
Serial.println(" µg/m³");

// Séparateur visuel pour une meilleure lisibilité des données
Serial.println("-----");

delay(1000); // Délai d'une seconde entre chaque cycle de mesure
}
```

Gestion des Erreurs

Au cours du développement, plusieurs problèmes techniques ont été identifiés et résolus de manière systématique. Par exemple, une initialisation incorrecte des capteurs a été corrigée en ajoutant des boucles de vérification dans la fonction setup(). De plus, des valeurs anormalement élevées ont été observées lorsque l'alimentation était réglée à 5V ; ce problème a été résolu en stabilisant l'alimentation à 3,3V, conformément aux spécifications des capteurs. Enfin, des conflits sur le bus I2C ont été éliminés en reconnectant les capteurs un par un et en vérifiant leurs adresses respectives.

Fréquence des Mesures

La fréquence des mesures a été fixée à une fois par seconde, un paramètre ajustable via la fonction delay(1000) dans le code. Cette cadence permet un suivi en temps réel tout en limitant la surcharge du microcontrôleur.

6. Tests et Validation

Méthode de Test

La validation du système a été réalisée en deux étapes principales. Tout d'abord, chaque capteur a été testé individuellement à l'aide d'un oscilloscope pour analyser les trames I2C et confirmer leur conformité aux spécifications techniques. Ensuite, l'ensemble du système a été intégré et comparé à un capteur de qualité de l'air (IQA) commercial afin d'évaluer la précision globale des mesures.

Conditions de Test

Les tests ont été effectués dans une salle de classe, dans des conditions environnementales stables (température ambiante d'environ 20°C et pression atmosphérique standard), afin de garantir la reproductibilité des résultats.

Résultats des Tests

Les mesures obtenues se sont révélées cohérentes avec les attentes : les concentrations de SO₂ et NO₂ étaient inférieures à 0,1 ppm, tandis que les PM_{2.5} variaient entre 10 et 15 µg/m³, des valeurs alignées avec celles d'un capteur commercial testé dans les mêmes conditions. Ces résultats confirment la fiabilité et la précision du système conçu.

Intégration des Fiches de Test

Pour documenter de manière exhaustive le processus de validation, les fiches de test ont été intégrées ci-dessous. Ces fiches détaillent les procédures, les résultats attendus, les critères de réussite et les résultats obtenus pour chaque étape de test, offrant ainsi une traçabilité complète des vérifications effectuées.

Fiche de Test PT-CAPTEUR-01

Identification du Test : PT-CAPTEUR-01

Référence du Module Testé : ESP32-WROOM avec capteurs IQA

Objectif du Test : Vérifier la précision et la stabilité des mesures des capteurs de polluants atmosphériques dans des conditions contrôlées.

Moyens Mis en Œuvre :

Logiciel : Code déployé sur l'ESP32 via l'IDE Arduino.

Matériel : ESP32-WROOM connecté aux capteurs de polluants (SEN0460, SEN0470, SEN0471, SEN0472).

Outils de Développement : IDE Arduino pour la programmation et un oscilloscope pour l'analyse des trames I2C.

Prérequis/Préconditions :

Le microcontrôleur doit être programmé pour lire les données des capteurs via le protocole I2C.

Les capteurs doivent être correctement connectés au bus I2C avec des adresses distinctes.

Les tests doivent être réalisés dans un environnement stable (température ambiante de 20°C, pression atmosphérique standard).

Une alimentation stable de 3,3V doit être fournie au système.

Procédure de Test unitaire et Résultats :

Num Étape	Procédure du Test	Résultat Attendu	Critères de Réussite	Résultat Obtenu	Validation (O/N)
CAPT- 1	Connecter le SEN0460 (PM2.5 et PM10) à l'ESP32 via I2C.	Le capteur est détecté par l'ESP32 sans erreur.	Connexion réussie sans conflit I2C.	Capteur détecté avec adresse 0x19.	Oui
CAPT- 2	Téléverser le code Arduino (exemple fourni par DFRobot) pour le SEN0460.	Le code se téléverse sans erreur et le capteur commence à transmettre des données.	Compilation et exécution réussies.	Code téléversé et exécuté correctement.	Oui

Num Étape	Procédure du Test	Résultat Attendu	Critères de Réussite	Résultat Obtenu	Validation (O/N)
CAPT- 3	Ouvrir le moniteur série pour observer les valeurs de PM2.5 et PM10.	Le moniteur série affiche des concentrations cohérentes (ex. 10-15 $\mu\text{g}/\text{m}^3$ en air propre).	Lectures stables et plausibles.	PM2.5 : 12 $\mu\text{g}/\text{m}^3$, PM10 : 18 $\mu\text{g}/\text{m}^3$, stables.	Oui
CAPT- 4	Analyser les trames I2C avec un oscilloscope.	Les trames montrent l'adresse 0x19 et une transmission conforme à la documentation.	Trames I2C conformes, sans erreurs.	Trames correctes, pas d'interférences détectées.	Oui
CAPT- 5	Connecter le SEN0470 (SO_2) à l'ESP32 via I2C.	Le capteur est détecté par l'ESP32 sans erreur.	Connexion réussie sans conflit I2C.	Capteur détecté avec adresse 0x68.	Oui
CAPT- 6	Téléverser le sketch Arduino fourni par DFRobot pour le SEN0470.	Le code se téléverse sans erreur et le capteur commence à transmettre des données.	Compilation et exécution réussies.	Code téléversé et exécuté correctement.	Oui
CAPT- 7	Ouvrir le moniteur série pour observer les valeurs de SO_2 .	Le moniteur série affiche des concentrations dans la plage 0-20 ppm (ex. < 0,1 ppm en air propre).	Lectures stables et raisonnables.	SO_2 : 0,05 ppm, stable.	Oui

Num Étape	Procédure du Test	Résultat Attendu	Critères de Réussite	Résultat Obtenu	Validation (O/N)
CAPT- 8	Analyser les trames I2C avec un oscilloscope.	Les trames montrent l'adresse 0x68 et une transmission conforme à la documentation.	Trames I2C conformes, sans erreurs.	Trames correctes, pas d'interférences détectées.	Oui
CAPT- 9	Connecter le SEN0471 (NO ₂) à l'ESP32 via I2C.	Le capteur est détecté par l'ESP32 sans erreur.	Connexion réussie sans conflit I2C.	Capteur détecté avec adresse 0x69.	Oui
CAPT- 10	Téléverser le sketch Arduino fourni par DFRobot pour le SEN0471.	Le code se téléverse sans erreur et le capteur commence à transmettre des données.	Compilation et exécution réussies.	Code téléversé et exécuté correctement.	Oui
CAPT- 11	Ouvrir le moniteur série pour observer les valeurs de NO ₂ .	Le moniteur série affiche des concentrations dans la plage 0-20 ppm (ex. < 0,1 ppm en air propre).	Lectures stables et raisonnables.	NO ₂ : 0,08 ppm, stable.	Oui
CAPT- 12	Analyser les trames I2C avec un oscilloscope.	Les trames montrent l'adresse 0x69 et une transmission conforme à la documentation.	Trames I2C conformes, sans erreurs.	Trames correctes, pas d'interférences détectées.	Oui

Num Étape	Procédure du Test	Résultat Attendu	Critères de Réussite	Résultat Obtenu	Validation (O/N)
CAPT- 13	Connecter le SEN0472 (O ₃) à l'ESP32 via I2C.	Le capteur est détection par l'ESP32 sans erreur.	Connexion réussie sans conflit I2C.	Capteur détecté avec adresse 0x74.	Oui
CAPT- 14	Téléverser le sketch Arduino fourni par DFRobot pour le SEN0472.	Le code se téléverse sans erreur et le capteur commence à transmettre des données.	Compilation et exécution réussies.	Code téléversé et exécuté correctement.	Oui
CAPT- 15	Ouvrir le moniteur série pour observer les valeurs de O ₃ .	Le moniteur série affiche des concentrations dans la plage 0-10 ppm (ex. < 0,1 ppm en air propre).	Lectures stables et raisonnables.	O ₃ : 0,03 ppm, stable.	Oui
CAPT- 16	Analyser les trames I2C avec un oscilloscope.	Les trames montrent l'adresse 0x74 et une transmission conforme à la documentation.	Trames I2C conformes, sans erreurs.	Trames correctes, pas d'interférences détectées.	Oui

- **Résultats** : [x] OK
- **Conclusion du Test** : Les tests ont validé avec succès la précision et la stabilité des mesures des capteurs de polluants. Toutes les étapes ont été réalisées sans anomalies majeures, et les données obtenues sont cohérentes avec les attentes dans un environnement contrôlé.

7. Intégration et Déploiement

Communication avec la Plateforme

À terme, les données collectées par les capteurs seront transmises via le protocole NB-IoT grâce au module SIM7080G, qui permettra une connexion fiable vers un broker MQTT. En phase de test, les données sont simplement affichées sur le moniteur série, mais l'infrastructure est prête pour une intégration complète avec la plateforme IoT.

Alimentation

Les tests initiaux ont été réalisés avec une alimentation USB connectée à l'ESP32. Pour un déploiement sur le terrain, une solution basée sur une batterie rechargeable, potentiellement associée à des panneaux solaires, est envisagée afin de garantir une autonomie suffisante dans des environnements extérieurs.

Proposition de Déploiement

Pour maximiser l'efficacité du système, je propose de positionner les capteurs à proximité des axes routiers principaux et des zones industrielles des communes concernées, où les niveaux de pollution sont susceptibles d'être les plus élevés. Cette stratégie permettra de capturer des données représentatives des scénarios les plus critiques.

8. Défis Rencontrés et Solutions Apportées

Au cours du projet, plusieurs défis techniques ont émergé, nécessitant des ajustements précis :

Problème : Des valeurs de mesure anormalement élevées ont été observées lors des premiers tests, rendant les données peu fiables.

Solution : Après analyse, il s'est avéré que l'alimentation à 5V était inadaptée ; le passage à une alimentation stabilisée à 3,3V a résolu ce problème, alignant les mesures sur les attentes.

Problème : Des conflits sur le bus I2C ont entraîné des erreurs de communication entre les capteurs et l'ESP32.

Solution : Une reconnexion séquentielle des capteurs, associée à une vérification minutieuse des adresses I2C, a permis d'éliminer ces interférences et de rétablir une communication fluide.

9. Conclusion

Résumé

Ce projet m'a permis de concevoir, développer et valider avec succès un système fonctionnel capable de mesurer en continu les concentrations de SO_2 , NO_2 , O_3 , $PM_{2.5}$ et PM_{10} avec une précision conforme aux exigences scientifiques et réglementaires. Grâce à une architecture bien pensée, un code et des tests le système répond aux objectifs initiaux du projet "Smart Territory".

10. Annexes

Schémas Techniques : Diagrammes de connexion I2C entre les capteurs et l'ESP32.

Code Source : Fourni dans la section "Logiciel Développé" ci-dessus.

Références :

- Documentation des capteurs DFRobot : [SEN0465-SEN0476](#)
- Documentation du capteur $PM_{2.5}/PM_{10}$: [SEN0460](#)

Fiches de Test : Intégrées dans la section "Tests et Validation" pour une traçabilité complète.