



Brevet de Technicien Supérieur CIEL

Session 2025
Lycée Saint Aubin la Salle

Smart Territories

Nathan Hebert

Dossier technique du projet - partie individuelle

1 TABLE DES MATIÈRES

Table des matières

1 Table des matières.....	1
1 - Situation dans le projet.....	3
1.1 - Synoptique de la réalisation.....	3
1.2 - Description de la partie personnelle.....	4
2 - Réalisation de la fonction et/ou cas d'utilisation « Localiser les capteurs et les zones de la ville ».....	4
Diagrammes.....	4
2.1 - Conception détaillée.....	5
Description technique.....	5
1. Initialisation et configuration des capteurs et du module SIM7080G.....	6
Éléments clés :.....	6
a) Déclaration et initialisation des capteurs :.....	6
b) Configuration réseau et MQTT :.....	6
2. Lecture des données environnementales.....	7
Mesures effectuées :.....	7
3. Publication des données via MQTT.....	7
a) Connexion au broker :.....	7
b) Formatage et publication des messages :.....	7
c) Déconnexion :.....	8
2.2 - Tests unitaire.....	9
Tests réalisés.....	9
2.2.1 - Test unitaire du module matériel/logiciel.....	9
Description du Plan de Test – Capteurs météorologiques et module SIM7080G.....	10
Objectif du Test.....	10
Type et Conditions de Test.....	10
Ressources Matérielles Nécessaires.....	10
Conditions de Réalisation du Test.....	11
Critères d'entrée :.....	11
Critères de sortie :.....	11
2.3 - Schéma Structurel.....	11
2.3.1 - PCB.....	14
2.3.2 - Problèmes rencontrés.....	16
3 - Bilan de la réalisation personnelle.....	17

1 - Situation dans le projet

1.1 - Synoptique de la réalisation

Le projet Smart Territories vise à mettre en place un réseau de collecte de données environnementales à l'échelle d'une ville.

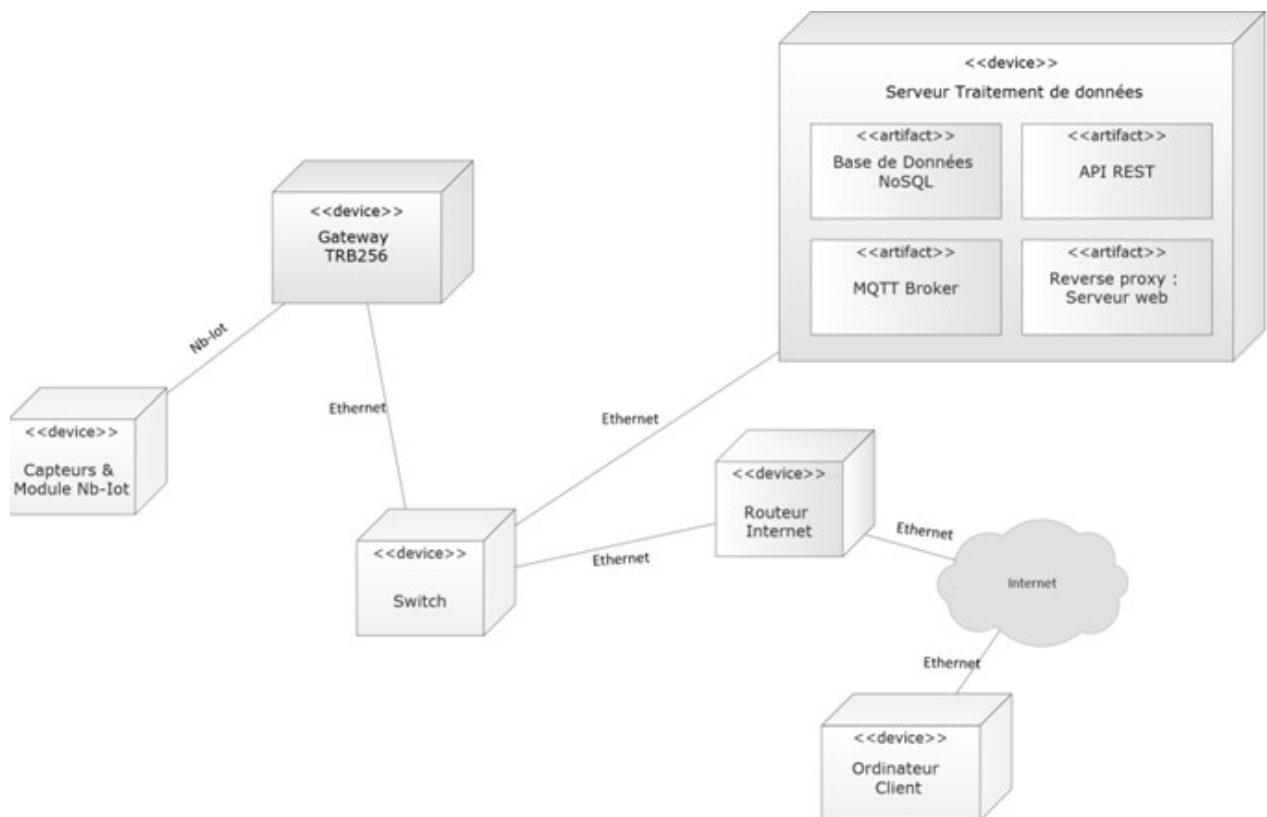
Pour cela, des capteurs météorologiques sont déployés afin de mesurer en continu la température, l'humidité relative, et la pression atmosphérique.

Dans ma partie du projet, je suis responsable :

- De la lecture des mesures environnementales (température, humidité, pression) issues des capteurs SHT35 et SEN0517 via bus I2C.
- De la transmission fiable de ces données en utilisant une liaison NB-IoT (via le module SIM7080G).
- De l'envoi des données formatées au sein d'un broker MQTT local pour stockage et exploitation ultérieure.

Ce transfert utilise une architecture basse consommation (NB-IoT) adaptée aux transmissions de longue durée et à faible volume de données.

Chaque capteur est également associé à une zone géographique de la ville pour permettre une cartographie précise des mesures environnementales.



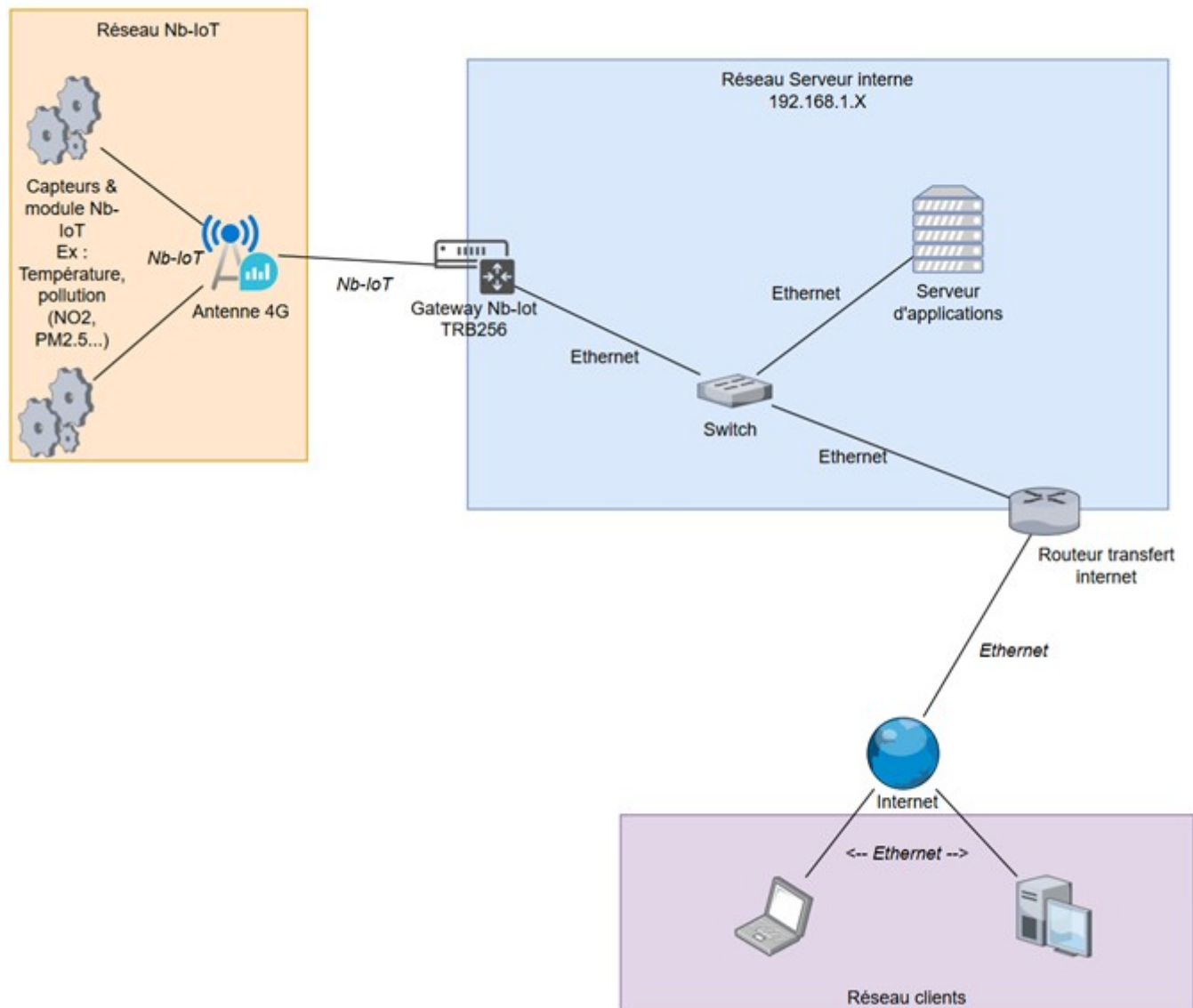
1.2 - Description de la partie personnelle

Mon travail personnel se concentre sur la lecture des capteurs météorologiques (SHT35 pour la température et l'humidité, SEN0517 pour la pression atmosphérique), la transmission de leurs données par NB-IoT, ainsi que sur l'association de chaque mesure à une zone de la ville préalablement définie.

Cela comprend :

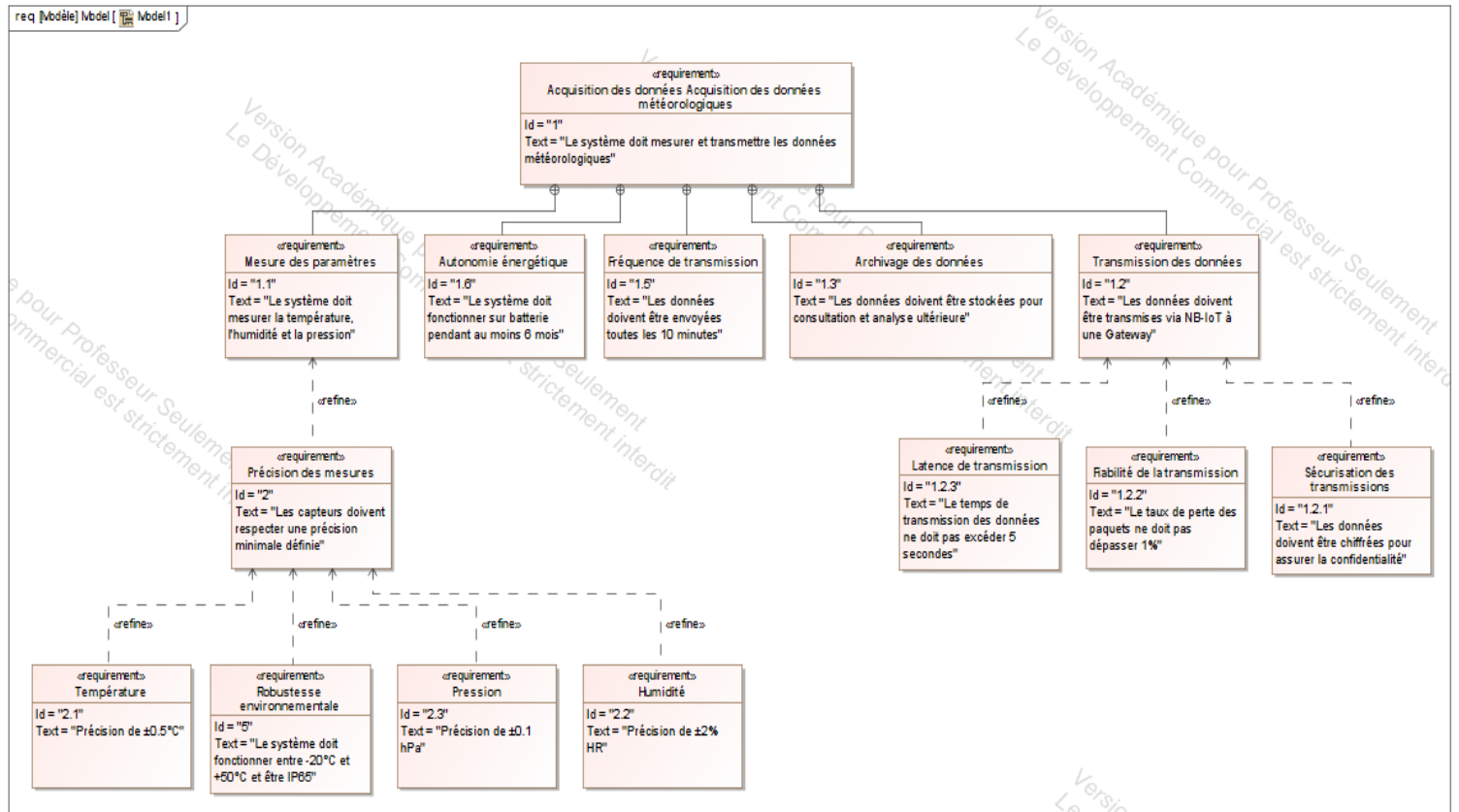
- La lecture des données capteurs.
- L'identification de la position du capteur (zone).
- L'envoi des mesures formatées vers un serveur MQTT local.
- La gestion de la fiabilité de la transmission sur un réseau NB-IoT.

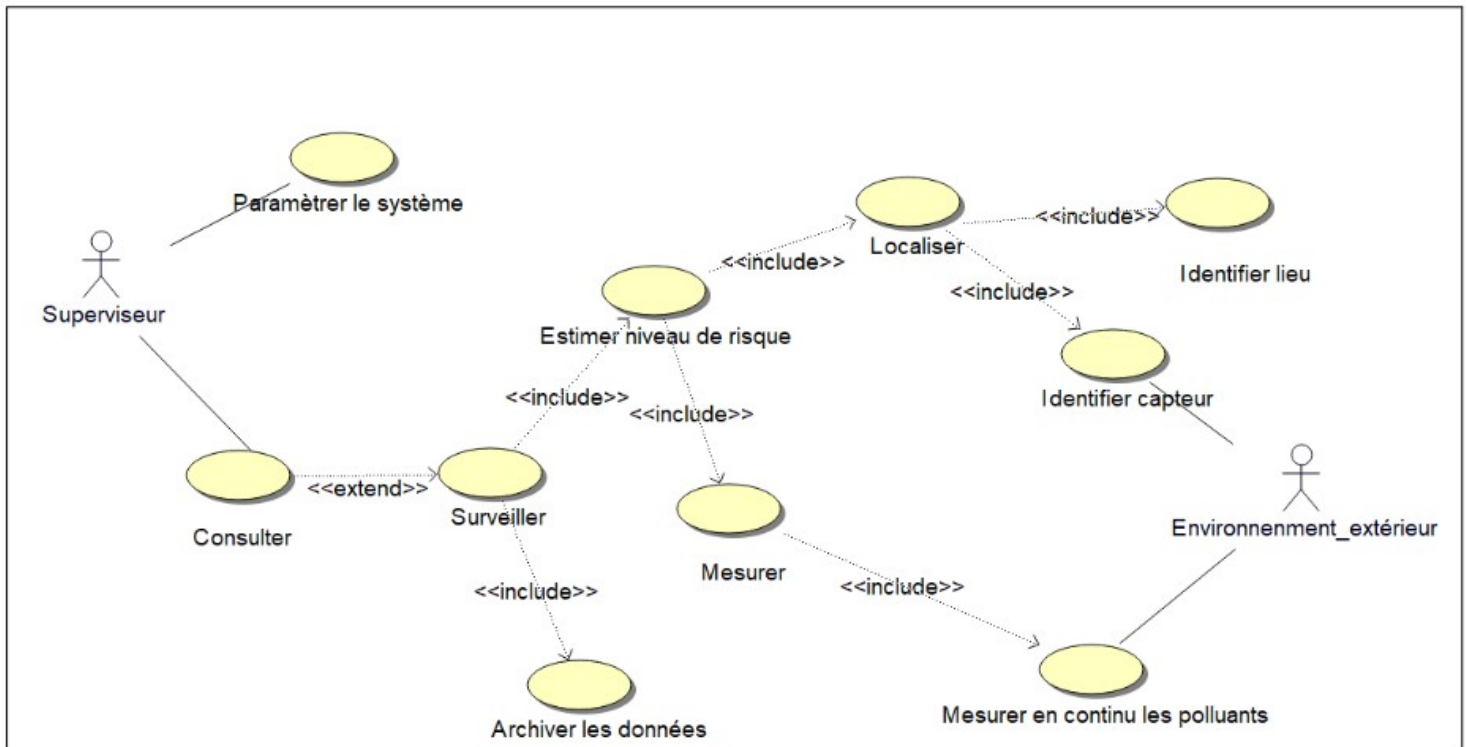
Schéma d'architecture projet
Smart Territories



2 - Réalisation de la fonction et/ou cas d'utilisation « Localiser les capteurs et les zones de la ville »

■ DIAGRAMMES





2.1 - Conception détaillée

Description technique

- Acquisition des données capteurs via bus I2C.
- Attribution d'une "zone" à chaque capteur en fonction de son identifiant.
- Formatage du message MQTT incluant les données météorologiques + la zone.
- Transmission via le module SIM7080G sur réseau NB-IoT.
- Réception sur un broker MQTT local pour analyse et stockage.

1. INITIALISATION ET CONFIGURATION DES CAPTEURS ET DU MODULE SIM7080G

Cette première partie concerne la mise en place des composants matériels (capteurs et module de communication) et la préparation du système à la collecte et l'envoi de données.

ÉLÉMENTS CLÉS :

A) DÉCLARATION ET INITIALISATION DES CAPTEURS :

```
DFRobot_SHT3x sht3x(&Wire, 0x45);  
DFRobot_ICP10111 icp;  
DFRobot_GAS_I2C gasNO2(&Wire, I2C_ADDRESS_NO2);  
DFRobot_GAS_I2C gasO3(&Wire, I2C_ADDRESS_O3);
```

Ces lignes créent les objets pour chaque capteur :

- *sht3x* : mesure la température et l'humidité.
- *icp* : mesure la pression atmosphérique.
- *gasNO2* et *gasO3* : détectent les concentrations en dioxyde d'azote (NO₂) et ozone (O₃).

Chaque capteur est initialisé dans *set up ()* avec des vérifications de bon fonctionnement, par exemple :

```
if (sht3x.begin() == 0) capteurOK_SHT = true;
```

B) Initialisation de la communication avec le module SIM7080G :

```
HardwareSerial simSerial(1); // UART1 : TX=17, RX=16  
simSerial.begin(115200, SERIAL_8N1, 16, 17);
```

Le module SIM7080G est relié via UART1. On établit la communication série pour envoyer des commandes AT.

C) CONFIGURATION RÉSEAU ET MQTT :

Appel de la fonction :

```
configureSIM7080G_stepByStep();
```

Cette fonction envoie une série de commandes AT pour :

- Vérifier la carte SIM et l'état du module
- Choisir le bon mode LTE-M (AT+CMNB=1, AT+CNMP=38)
- Configurer le PDP (APN : "eapn1.net")
- Vérifier la qualité du signal et la connexion réseau
- Paramétrer les informations pour MQTT : serveur, port, clientID, topic...

Importance :

Cette étape est indispensable pour que l'ESP32 soit capable de communiquer sur le réseau LTE-M et d'envoyer les données capteurs à un serveur distant (ici Mosquitto via MQTT).

2. LECTURE DES DONNÉES ENVIRONNEMENTALES

Une fois les capteurs prêts, cette partie consiste à récupérer les mesures, à les afficher et à les préparer pour envoi.

MESURES EFFECTUÉES :

```
float temperature = capteurOK_SHT ? sht3x.getTemperatureC() : 0.0;  
float humidity = capteurOK_SHT ? sht3x.getHumidityRH() : 0.0;  
float pressure = capteurOK_ICP ? icp.getAirPressure() / 100.0 : 0.0;  
float no2_ppm = capteurOK_NO2 ? gasNO2.readGasConcentrationPPM() : 0.0;  
float o3_ppm = capteurOK_O3 ? gasO3.readGasConcentrationPPM() : 0.0;
```

Chaque variable contient la valeur lue si le capteur est disponible, ou 0.0 par défaut. On lit successivement :

- Température en °C
- Humidité en %
- Pression atmosphérique en hPa
- Concentration de NO₂ et O₃ en ppm

Les valeurs sont ensuite affichées dans la console série avec :

```
Serial.println("Température : " + String(temperature) + " °C");
```

Importance :

C'est le cœur de l'application capteur : cette section transforme les données physiques en valeurs numériques exploitables.

3. PUBLICATION DES DONNÉES VIA MQTT

Après acquisition, les données sont formatées en JSON et transmises à un broker MQTT distant.

A) CONNEXION AU BROKER :

```
sendATCommand("AT+SMCONN", 5000);
```

Cela établit une session MQTT avec le serveur `test.mosquitto.org`.

B) FORMATAGE ET PUBLICATION DES MESSAGES :

Chaque type de donnée est envoyé dans un message JSON avec la fonction :

```
publishMessage("{\"Capteur\":\"\" + String(ZONE_ID) + "\",\"TypeDeDonnee\":\"Temperature\",\"Date\":\"\" + dateTimeBuffer + "\",\"Valeur\":\"\"  
+ String(temperature, 1) + \"\"}");
```

La fonction `publishMessage()` construit d'abord la commande `AT+SMPUB`, puis envoie le message JSON sur le réseau.

Chaque message contient :

- Le nom de la zone ("SaintAubin")
- Le type de donnée (Température, Humidité, etc.)
- La date/heure obtenue via la commande `AT+CCLK?`
- La valeur mesurée

C) DÉCONNEXION :

```
sendATCommand("AT+SMDISC", 2000);
```

On termine proprement la session MQTT.

Importance :

Cette section fait le lien entre l'objet connecté et le cloud. Elle permet l'intégration des données dans une infrastructure IoT distante.

Boucle loop() : Communication interactive

```
void loop() {  
  if (Serial.available()) { ... }  
  if (simSerial.available()) { ... }  
}
```

Elle permet à l'utilisateur d'envoyer manuellement des commandes AT au module SIM7080G depuis le moniteur série et de voir les réponses. Cela facilite le debug et les tests.

Conclusion :

Partie	Description détaillée	Rôle
Initialisation & config.	Déclaration, test des capteurs, configuration SIM7080G et MQTT	Préparer le matériel pour les mesures et la communication
Lecture des capteurs	Récupération des données environnementales	Obtenir les informations nécessaires à transmettre
Publication MQTT	Format JSON, envoi sur serveur via SIM7080G	Transmettre les mesures à un serveur IoT externe

2.2 - Tests unitaire

TESTS RÉALISÉS

- Validation de la lecture des données capteurs.
- Association correcte capteur <-> zone.
- Test de connexion NB-IoT stable.
- Test d'envoi et de réception MQTT.

2.2.1 - Test unitaire du module matériel/logiciel

Identification du Plan de Tests	PT-UNIT-001	Référence du module testé	Meteorological Sensors and SIM Module	
Objectif du test :	Vérifier le bon fonctionnement individuel des capteurs SHT35 (température et humidité), SEN0517 (pression), et du module SIM7080G (connexion <u>LTE-M</u> et communication <u>MQTT</u>).			
Type de test :	Fonctionnel			
Environnement du test	Salle de cours au lycée Saint Aubin La Salle. Conditions ambiantes : température ~20-25°C, humidité ~40-60%, pression ~1013 hPa.			
Ressources matérielles :	Capteur SHT35 (I2C, 3.3V). Capteur SEN0517 (I2C, 3.3V). Module SIM7080G Microcontrôleur ESP32. Alimentation 3.3V et 5V. Logiciels : Arduino IDE	Ressources humaines	Étudiant responsable du module météorologique.	
Conditions du test				
Critères d'Entrée		Critères de Sortie		
Capteurs et module <u>SIM</u> correctement connectés à l'ESP32. Bibliothèques installées et code source compilé. Instruments de référence disponibles (thermomètre, hygromètre, baromètre).		Tous les tests unitaires réussis avec mesures dans les tolérances et communications fonctionnelles. Rapport de test complété.		
Procédures du test				
ID Test	Description du Test	Planification	Responsable	Commentaire
T1	Mesure de la température (SHT35)	15/01/2025	Nathan	Vérifier la précision par rapport à un thermomètre de référence.
T2	Mesure de l'humidité relative (SHT35)	15/01/2025	Nathan	Comparer avec un hygromètre de référence.
T3	Mesure de la pression atmosphérique (SEN0517)	15/01/2025	Nathan	Comparer avec un baromètre de référence.
T4	Connexion <u>LTE-M</u> (SIM7080G)	15/01/2025	Nathan	Vérifier la connexion au réseau <u>LTE-M</u> .
T5	Communication <u>MQTT</u> (SIM7080G)	15/01/2025	Nathan	Vérifier l'envoi de données au broker.

DESCRIPTION DU PLAN DE TEST – CAPTEURS MÉTÉOROLOGIQUES ET MODULE SIM7080G

Ce plan de test, référencé PT-UNIT-001, vise à vérifier le bon fonctionnement individuel de chacun des composants essentiels du système météorologique embarqué, notamment :

- Le capteur SHT35, qui mesure la température et l'humidité relative.
- Le capteur SEN0517, utilisé pour la mesure de la pression atmosphérique.
- Le module SIM7080G, utilisé pour la communication LTE-M et l'envoi des données via MQTT.

OBJECTIF DU TEST

L'objectif est de s'assurer que chacun des capteurs fonctionne correctement lorsqu'ils sont connectés à un ESP32, et que le module SIM7080G est capable d'établir une connexion LTE-M et d'envoyer des données au serveur MQTT (broker). Ce test permet de valider que le système répond bien aux exigences de communication et de mesure définies dans le projet.

TYPE ET CONDITIONS DE TEST

- Type de test : Test fonctionnel.
- Environnement : Salle de cours au lycée Saint Aubin La Salle, dans des conditions ambiantes contrôlées (température entre 20 et 25 °C, humidité entre 40 et 60 %, pression atmosphérique autour de 1013 hPa).

RESSOURCES MATÉRIELLES NÉCESSAIRES

- Capteur SHT35 (alimenté en 3.3 V, communication I2C)
- Capteur SEN0517 (alimenté en 3.3 V, I2C)
- Module SIM7080G (UART, 3.3 V ou 5 V selon les besoins)
- Microcontrôleur ESP32
- Alimentation 3.3 V / 5 V
- Logiciel Arduino IDE
- Instruments de référence : thermomètre, hygromètre, baromètre
- Ressources humaines : étudiant responsable du sous-système météo

1. BLOC BATTERIE ET RÉGULATION DE TENSION

Ce bloc assure l'alimentation du circuit à partir d'une batterie Li-Ion/Li-Po :

- BT1 : connecteur batterie (3.7V typique).
- U3 : régulateur MCP1700 (3.3V) — régule la tension batterie vers 3.3V stable.
- Filtres en entrée et sortie : 10 μ F, 0.1 μ F, 10 pF, pour la stabilité.
- Sortie : 3.3V, alimente l'ESP32 et les capteurs.

2. BLOC MICROCONTRÔLEUR (ESP32)

C'est le cœur du système, il gère la lecture des capteurs, la communication avec le module SIM7080G, etc.

- Alimentation : 3.3V (VDD)
- Quartz externe 32 MHz avec deux condensateurs 10 pF
- Condensateurs de découplage : 10 μ F et 100 nF
- Bouton RESET connecté à EN
- IO0 utilisé pour flasher l'ESP32
- GPIOs connectés à divers périphériques, comme :
 - LED (GPIO10)
 - UART (ESP32_RX / TX)
 - I2C via borniers
- En appuyant sur le bouton, la broche PWRKEY est tirée à GND pour une courte durée, activant ainsi le module SIM7080G.
- C'est un allumage manuel du module cellulaire.

3. BLOC CONVERTISSEUR BOOST (5V)

Convertisseur MT3608 pour générer du 5V à partir de 3.7V batterie :

- Inductance L1 = 10 μ H
- Diode Schottky SS14
- Résistances 220k / 56k pour régler la sortie à 5V
- Filtres d'entrée/sortie : 10 μ F, 10 pF
- Sert à alimenter les modules 5V (ex : SIM7080G)

4. BLOC CH340C (USB VERS SÉRIE)

Permet la programmation de l'ESP32 via USB :

- Connecteur USB Micro-B (J1)
- CH340C (U1) convertit USB en UART
- Connexion TX/RX vers ESP32
- Q1/Q2 (BSS138) MOSFETs permettent le reset automatique (liés à EN et IO0)
- Condensateurs de filtrage : 100 nF

5. BLOC ACTIVATION DE LA KEY (SIM7080G)

- JP1 est un cavalier ou bouton connecté à PWRKEY
- Permet de tirer PWRKEY à GND pour allumer manuellement le module SIM7080G

6. BLOC INDICATEUR (LED)

- LED connectée à GPIO10
- Résistance de 1 kΩ
- Permet de visualiser l'état du système (debug, transmission, etc.)

7. BLOC BORNIERES CAPTEURS & SIM7080G

Regroupe les connecteurs pour capteurs I2C et le module SIM7080G :

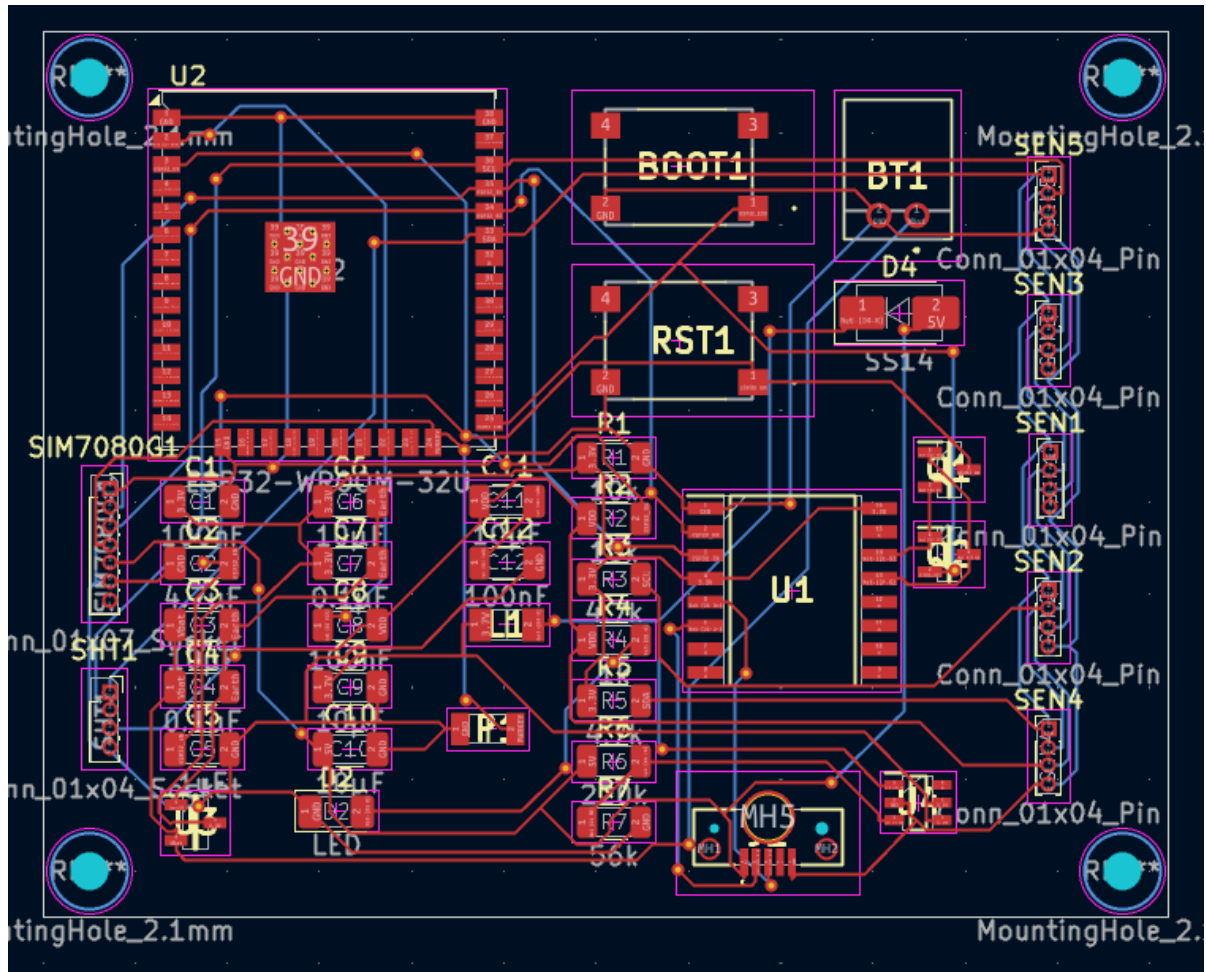
SIM7080G :

- Connexion UART :
 - ESP32_TX → RX du module
 - ESP32_RX ← TX du module
- PWRKEY pour allumage
- GND / 5V / I2C (optionnel)

CAPTEURS (I2C) :

- SEN1 à SEN5 : SDA, SCL, VDD, GND
- SHT1 : dédié au capteur SHT35 (température / humidité)

2.3.1 -PCB



2.3.2 - Problèmes rencontrés

*Connexion NB-IoT : Échec de connexion réseau dû à un signal faible. Solution : Vérification via AT+CSQ et activation mode NB-IoT.
Enseignement : Vérifier la couverture réseau en amont.*

Démarrage SIM7080G : Module non fonctionnel à cause d'une alimentation instable. Solution : Alimentation 5V et séquence correcte broche K. Enseignement : Garantir une alimentation stable.

*Configuration APN : Connexion données impossible par APN erroné. Solution : Configuration correcte via AT+CGDCONT.
Enseignement : Valider l'APN avec l'opérateur.*

*Communication UART : Réponses instables dues à un débit incorrect. Solution : Ajustement à 115200 et câbles optimisés.
Enseignement : Configurer précisément l'interface UART.*

*Envoi MQTT : Données non reçues à cause d'un format JSON incorrect. Solution : JSON standardisé et envoi espacé.
Enseignement : Assurer un formatage rigoureux des données.*

3 - Bilan de la réalisation personnelle

Contribution au Projet : J'ai conçu et mis en œuvre le système de mesure météorologique, intégrant les capteurs SHT35 (température, humidité) et SEN0517 (pression) à l'ESP32, ainsi que la transmission des données via le module SIM7080G vers un broker MQTT. J'ai développé le code Arduino, configuré les communications I2C et UART, et effectué les tests unitaires et d'intégration.

Compétences Développées : Ce projet m'a permis de maîtriser la programmation de microcontrôleurs, la gestion des protocoles NB-IoT et MQTT, et le débogage matériel/logiciel. J'ai renforcé mes compétences en analyse de problèmes techniques, notamment pour résoudre des erreurs de connexion réseau et de communication UART.

Enseignements Tirés : J'ai appris l'importance d'une planification rigoureuse, notamment pour vérifier les prérequis réseau et matériels dès le départ. La collaboration avec l'équipe et la consultation de documentations techniques ont été essentielles pour surmonter les obstacles.

Conclusion : Cette expérience a consolidé mes compétences techniques et ma capacité à gérer un projet complexe, tout en soulignant l'importance de la persévérance et de l'anticipation dans les projets IoT.