

CONCEPTION ET DEPLOIEMENT D'UNE STATION ENVIRONNEMENTALE SATELLITE ECOLOGGING : RAPPORT TECHNIQUE

Master 2 - Ingénierie des Systèmes Embarqués



Réalisé par :

- **AZALLAL Mohamed**
- **CHALAL Katia**
- **GOUICHICHE Hani**
- **MARIR Badr Eddine**

Encadré par :

- **Mr P. Bordenave**
- **Mr P. Chaumeil**

Table des matières :

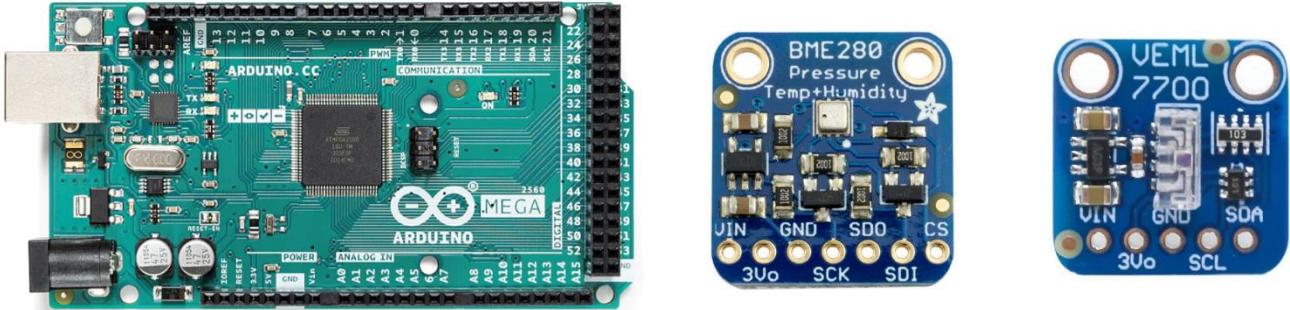
I. Architecture générale du système :	4
1.Brochage :.....	4
i. RTC DS3231 :	4
ii. MicroSD Module :	5
iii. BME280 et VEML7700 :	5
II. Prototype 1 : Conception du système avec le module KIM2 :.....	6
1.Connexion du module satellite KIM2 à l'Arduino :.....	6
2.Séquence d'initialisation du module KIM2 :.....	9
3.Encodage et transmission des données :.....	9
i. Encodage des données :.....	9
ii. Taille des messages et choix de la configuration :	10
iii. Transmission vers le satellite :	10
iv. Fonctionnement de l'envoi :	11
4.Partie serveur : réception et visualisation des données :	11
i. Utilisation de l'API CLS :	12
ii. Base de données et traitement :	13
iii. Interface web de visualisation (dashboard) :	13
iv. Utilisation de GitHub dans le projet :	14
v. Hébergement du serveur sur Render :	14
vi. Fonctionnement global du serveur :	15
vii. Rôle de la partie serveur dans le projet :	16
5.Fonctionnement global du prototype 1 avec KIM2:.....	16
6.Validation expérimentale du système :.....	17
7.Limites du prototype 1 avec KIM2:	19
i. Fréquence réelle de réception des données :	19
ii. Latence de transmission :	19
iii. Consommation énergétique :	19
iv. Limitation du volume de données :	19
v. Prototype en phase expérimentale :	20
8.Perspectives d'amélioration :.....	20
i. Ajout de nouveaux capteurs :	20
ii. Optimisation énergétique :	20

iii. Amélioration de la transmission satellite :	20
iv. Extension de l'interface web	20
v. Industrialisation du prototype.....	20
9. Conclusion :	20
III. Prototype 2 : Conception du système avec le module SMD Wing Arribada :	22
1.Connexion du module satellite SMD Wing à l'Arduino :	22
2.Difficultés rencontrées:.....	23
i. Problème majeur : absence de modem satellite :.....	23
ii. Autres difficultés :	23
3.Tests réalisés et résultats :	23
i. Tests fonctionnels validés :	23
ii. Tests non validés :	24
4.Synthèse :.....	24
5.Ce que nous aurions aimé tester :	24
6.Bilan :	24

I. Architecture générale du système :

Le système **ECOLOGGING** est composé de deux parties principales : la station de mesure sur le terrain et la partie serveur permettant la réception et l'exploitation des données transmises par satellite.

La station terrain est basée sur une carte **Arduino Mega**, qui assure la **lecture** des capteurs et la **gestion** de la **transmission**. Plusieurs capteurs sont connectés à l'Arduino afin de mesurer les paramètres environnementaux. Le capteur **BME280** permet de mesurer la température, l'humidité et la pression atmosphérique. Le capteur **VEML7700** est utilisé pour mesurer la luminosité ambiante.



Un module **RTC DS3231** est utilisé pour horodater précisément chaque mesure. Une carte **SD** permet d'enregistrer localement les données afin de conserver un historique et de sécuriser les mesures en cas de problème de transmission.

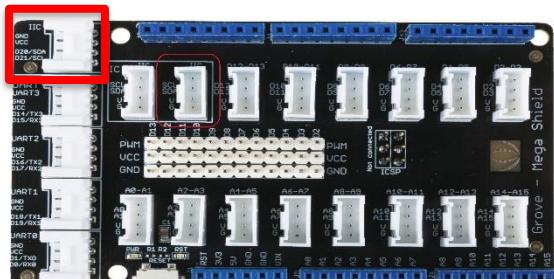


Les données sont ensuite encodées par l'Arduino et envoyées vers le module satellite. Ce module assure la transmission des données vers les satellites **Kinéis/Agros**, permettant l'envoi d'informations même en zone isolée sans réseau **GSM** ou **Wifi**.

1. Brochage :

i. RTC DS3231 :

Ce module, qui utilise le protocole de communication I2C, est relié dans l'ensemble des pins de I2C du Grove - Mega Shield comme montré dans la figure suivante :

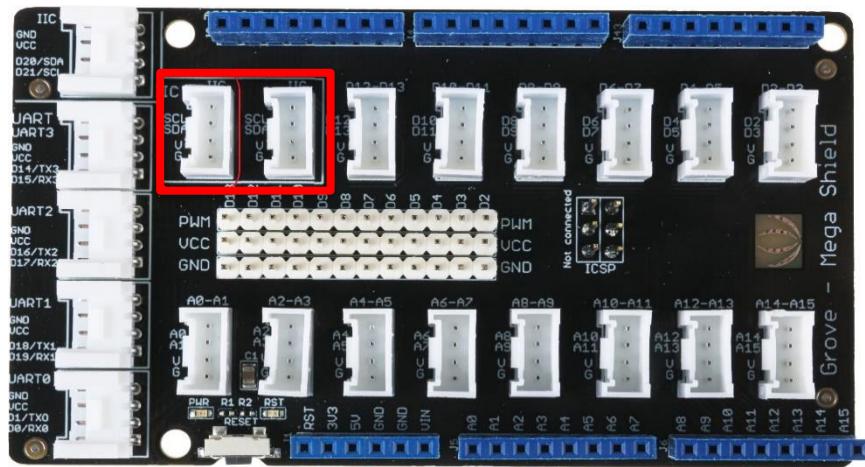


ii. MicroSD Module :

Pin du module	Pin de l'Arduino
MISO	52
SCK	54
SS	53
MOSI	51
GND	GND
+5V	5V

iii. BME280 et VEML7700 :

Les deux capteurs sont interconnectés à l'aide de quatre connecteurs WAGO 222. Chaque broche homologue (VCC, GND, SDA et SCL) des deux capteurs est mise en commun, puis raccordée aux broches I2C correspondantes du Grove - Mega Shield.



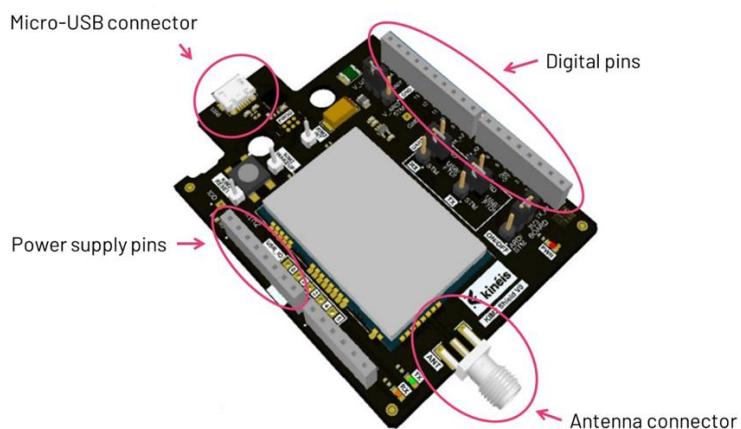
II. Prototype 1 : Conception du système avec le module KIM2 :

1. Connexion du module satellite KIM2 à l'Arduino :

Le module satellite **KIM2** est utilisé pour transmettre les données collectées vers les satellites **Kinéis/CLS**. Il est relié à l'**Arduino Mega** via une communication série (**UART**).

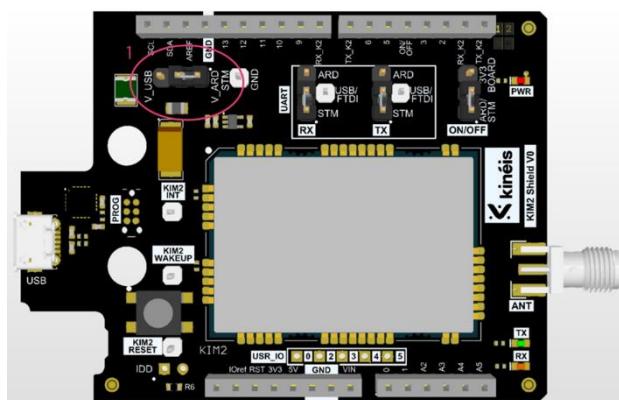
Avant de relier physiquement l'Arduino au module **KIM2**, il est nécessaire de configurer correctement les broches et les cavaliers (jumpers) présents sur la carte **KIM2**. Cette étape est essentielle pour assurer une communication correcte entre l'Arduino et le module satellite.

Le module KIM2 nécessite tout d'abord la connexion d'une **antenne** compatible **50 ohms** sur le connecteur SMA. **Cette antenne doit impérativement être connectée avant toute transmission afin d'éviter d'endommager le module lors de l'émission.**

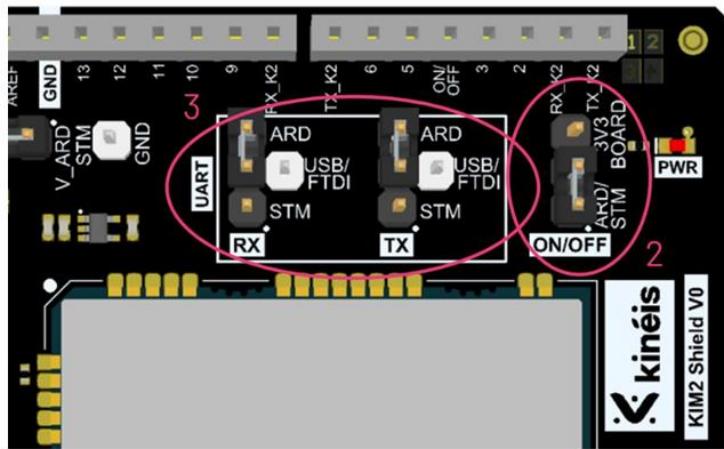


Ensuite, l'**alimentation** du module doit être configurée. Le module KIM2 nécessite une alimentation adaptée pour fonctionner correctement. Lors des tests réalisés dans ce projet, l'alimentation du module a été fournie par le port USB de l'ordinateur lors des phases de développement et de programmation.

Le cavalier d'alimentation du module doit être positionné en fonction de la source utilisée afin de sélectionner le mode d'alimentation approprié. Une masse commune entre l'Arduino et le module KIM2 est indispensable pour garantir une communication stable et éviter les perturbations lors de la transmission des données.



La communication entre l'Arduino et le module **KIM2** se fait en **UART** (communication série). Pour cela, les cavaliers de configuration de la communication doivent être positionnés sur le mode Arduino. Les broches RX et TX du module doivent être configurées pour communiquer avec l'Arduino via l'interface série. Le débit de communication utilisé est de **9600 bauds**.



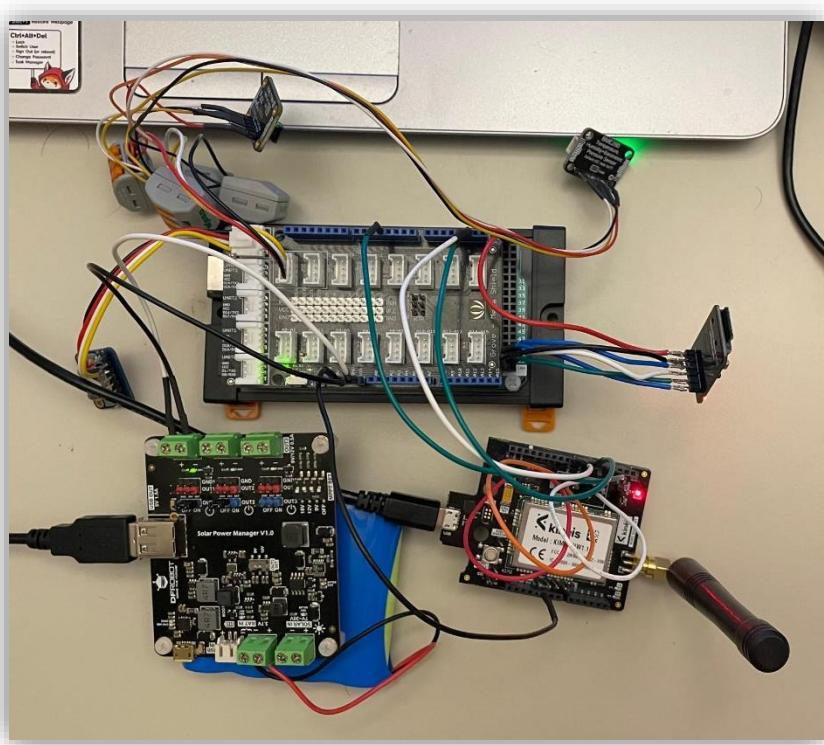
Une fois les cavaliers configurés correctement (alimentation et UART), le module peut être relié à l'Arduino Mega. Les connexions principales sont alors :

Module KIM2	Arduino Mega	Fonction
TX_K2 (pin7)	RX1 (pin 19)	Réception des données envoyées par le module satellite
RX_K2 (pin 8)	TX1 (pin 18)	Envoi des commandes AT vers le module satellite
GND	GND	Masse commune nécessaire pour la communication
ON/OFF	Pin digitale (exemple pin 4)	Permet d'allumer ou d'éteindre le module.

Cette configuration préalable garantit une communication stable entre l'Arduino et le module satellite et permet d'assurer une transmission fiable des données vers les satellites.

Après l'alimentation du module, un temps d'initialisation de quelques secondes est nécessaire pour permettre au module satellite de démarrer correctement.

L'Arduino communique avec le module KIM2 en utilisant des commandes **AT**. Ces commandes permettent de vérifier la connexion avec le module, configurer la radio satellite et d'envoyer les données vers le satellite



2. Séquence d'initialisation du module KIM2 :

Lors du démarrage du système, l'Arduino initialise le module satellite en envoyant plusieurs commandes AT :

Étape	Commande envoyée	Rôle
Test communication	AT+PING=?	Vérifie que le module répond
Configuration radio	AT+RCONF=...	Configure le réseau satellite Kinéis
Vérification config	AT+RCONF=?	Vérifie la configuration radio
Activation MAC	AT+KMAC=1	Active la communication satellite
Envoi données	AT+TX=payload	Envoie les données au satellite

3. Encodage et transmission des données :

Le système mesure en continu les paramètres environnementaux à l'aide des capteurs connectés à l'Arduino Mega : température, humidité, pression atmosphérique et luminosité.

Les mesures sont réalisées toutes les **30 secondes**. Chaque valeur est d'abord enregistrée localement sur la carte SD afin de conserver un historique complet des données, même en cas d'absence de transmission satellite. Ces mesures brutes sont stockées dans un fichier CSV avec la date et l'heure.

Afin de garantir la transmission des données via satellite et leur réception sur la plateforme CLS, tout en limitant la consommation énergétique et le coût des transmissions, les données ne sont pas envoyées immédiatement après chaque mesure.

Le système calcule une moyenne des mesures puis effectue un envoi toutes les **10 minutes**. Avec cette période, on garantit un envoi et une réception des données **au minimum deux fois par heure** sur la plateforme CLS.

Pendant cette période de 10 minutes, toutes les valeurs mesurées sont cumulées puis une moyenne est calculée pour chaque paramètre (température moyenne, humidité moyenne, pression moyenne, luminosité moyenne). Cette moyenne est ensuite envoyée via le module satellite KIM2. Elle est également enregistrée sur la carte SD afin de conserver une trace locale des données transmises et d'assurer une sauvegarde en cas de problème de transmission satellite.

i. Encodage des données :

Avant l'envoi vers le satellite, les valeurs doivent être converties en un format compatible avec le module KIM2. Les données sont donc encodées en **format hexadécimal**.

Chaque valeur est transformée en entier afin d'éviter l'envoi de nombres décimaux :

- Température $\times 100$
- Humidité $\times 100$
- Pression $\times 10$
- Luminosité en valeur entière

Les valeurs sont ensuite converties en hexadécimal et regroupées dans une seule trame de données.

ii. Taille des messages et choix de la configuration :

Le module satellite impose une taille précise pour les messages envoyés. Selon la configuration radio utilisée, plusieurs tailles de trames sont possibles :

- En mode **LDK** : taille fixe de **16 bytes**.
- En mode **LDA2** : tailles possibles de **4, 8, 12, 16, 20 ou 24 bytes**.

Dans ce projet, la configuration **LDA2** a été choisie.

Ce choix permet une plus grande flexibilité pour l'évolution du système. En effet, la station pourra intégrer à l'avenir d'autres capteurs environnementaux (qualité de l'air, niveau d'eau, etc.). Le mode **LDA2** permet donc d'augmenter la taille des messages si nécessaire **sans modifier toute l'architecture du système**.

Actuellement, une trame de **16 bytes** est utilisée. Cette taille est suffisante pour transmettre la température, l'humidité, la pression et luminosité, Tout en conservant une structure claire et optimisée.

Le choix de 16 bytes permet également de réduire la consommation énergétique, limiter le temps de transmission satellite, d'assurer une transmission fiable et de garder de l'espace pour ajouter de futurs capteurs si nécessaire.

La configuration radio utilisée pour le module KIM2 est définie par la commande AT suivante :

AT+RCONF=3d678af16b5a572078f3dbc95a1104e7

Cette commande configure le module en mode LDA2 et prépare la communication avec le réseau satellite **Kinéis/CLS**.

iii. Transmission vers le satellite :

Une fois la trame construite, l'Arduino l'envoie au module KIM2 via la liaison série UART en utilisant des **commandes AT**. Ces commandes permettent d'initialiser le module, de configurer la radio et d'envoyer les données vers les satellites.

Le module satellite est d'abord initialisé et configuré au démarrage grâce à plusieurs commandes AT. Ces commandes assurent la vérification de la communication, la configuration radio et la préparation du module à la transmission satellite.

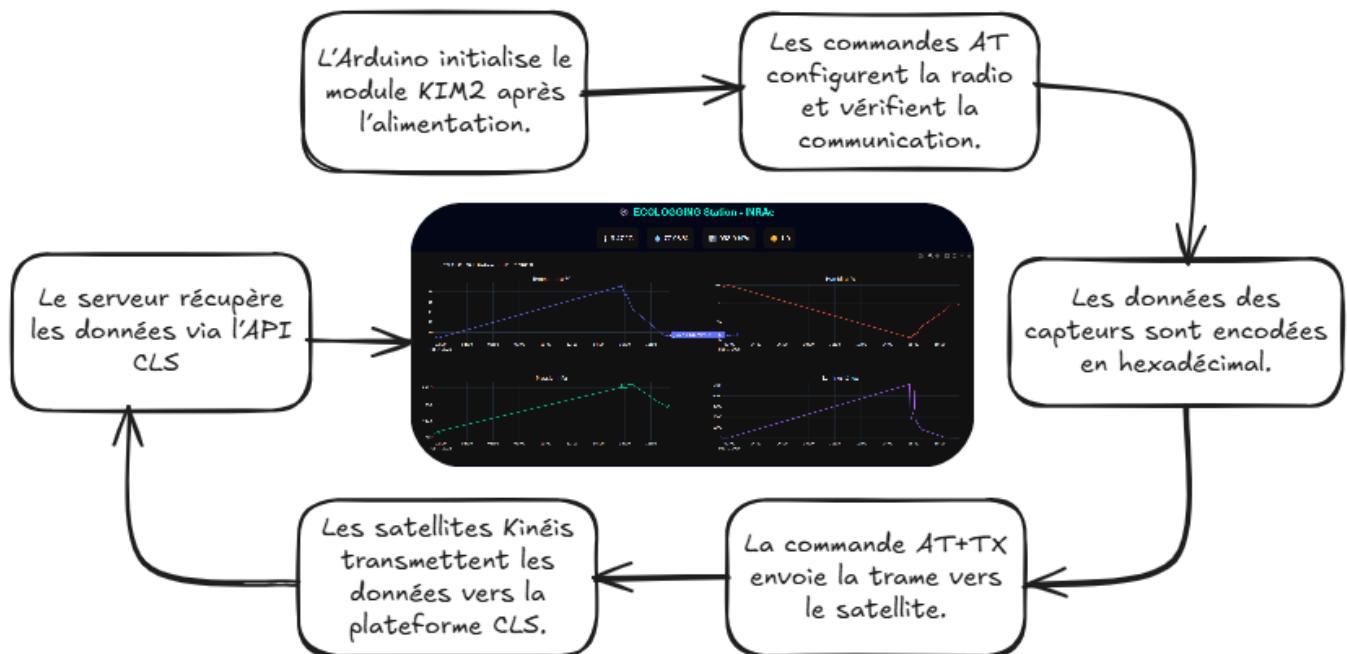
Lorsque la moyenne des données est prête, la trame hexadécimale est envoyée au module avec la commande d'émission. Le module KIM2 transmet ensuite les données vers les satellites Kinéis, qui relaient les informations vers la plateforme CLS où elles sont récupérées par le serveur et affichées sur le **dashboard web**.

Commandes AT utilisées pour la communication avec le module KIM2 :

Commande AT	Description	Rôle dans le système
AT+PING=?	Vérifie la communication avec le module KIM2	Permet de s'assurer que le module est bien alimenté et répond aux commandes
AT+RCONF=3d678af16b5a572078f3dbc95a1104e7	Configuration radio du module (mode LDA2, paramètres réseau Kinéis)	Configure la communication satellite et le protocole de transmission

AT+RCONF=?	Lecture de la configuration radio actuelle	Permet de vérifier que la configuration a bien été appliquée
AT+KMAC=1	Activation de l'adresse MAC du module	Autorise le module à émettre sur le réseau satellite
AT+TX=payload,00	Envoi des données vers le satellite	Transmet la trame hexadécimale contenant les mesures environnementales

iv. Fonctionnement de l'envoi :



Ce protocole de communication permet d'assurer une transmission fiable des données environnementales depuis une zone isolée vers une plateforme web accessible à distance.

4. Partie serveur : réception et visualisation des données :

Une fois transmises par satellite, les données sont reçues sur la plateforme **CLS (Collecte Localisation Satellites)**. CLS est une entreprise spécialisée dans la transmission de données par satellite et l'Internet des objets (IoT). Elle permet de recevoir, stocker et mettre à disposition les données envoyées par les objets connectés utilisant la communication satellite.



Dans le cadre du projet ECOLOGGING, CLS joue le rôle **d'intermédiaire** entre la station terrain et le serveur de visualisation. Les données envoyées par le module KIM2 sont d'abord transmises aux satellites Kinéis, puis redirigées vers les serveurs CLS. Ces données deviennent ensuite accessibles via l'interface web CLS ou via une API sécurisée.

Cependant, sur la plateforme CLS, les données reçues sont affichées **sous forme hexadécimale (HEX)** correspondant directement à la trame envoyée par le module satellite. Ce format brut est adapté à la transmission satellite mais reste difficile à interpréter pour un utilisateur final, car les valeurs des capteurs (température, humidité, pression, luminosité) ne sont pas directement lisibles.

Pour cette raison, nous avons décidé de développer une **interface web dédiée** permettant :

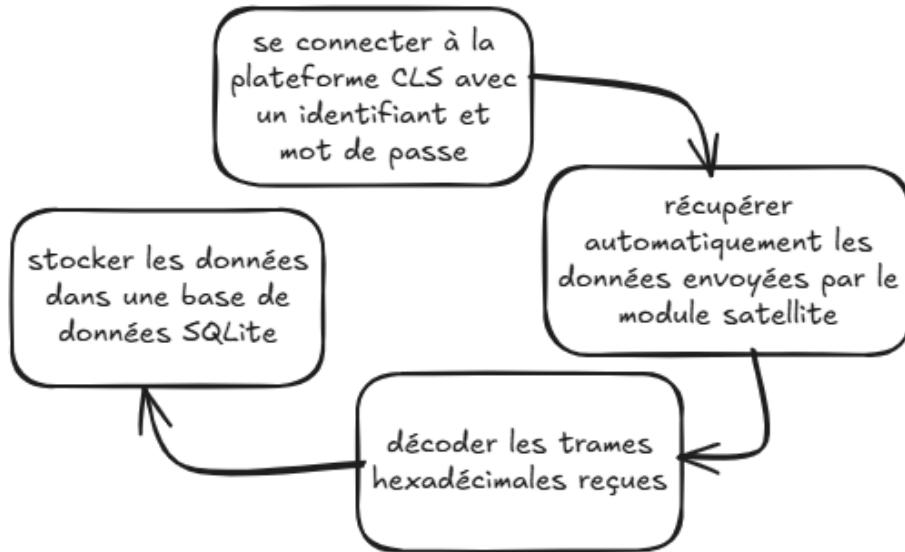
- de décoder automatiquement les trames hexadécimales reçues depuis CLS
- d'afficher les mesures avec les bonnes unités (°C, %, hPa, lux)
- de visualiser les données sous forme de graphiques
- de suivre l'évolution des mesures
- de faciliter l'exploitation des données par l'utilisateur

Cette interface constitue ainsi un tableau de bord clair et exploitable, contrairement à l'affichage brut en hexadécimal proposé par défaut sur la plateforme CLS.

i. Utilisation de l'API CLS :

Pour automatiser la récupération des données, nous utilisons l'API CLS. Une API (Application Programming Interface) est une interface permettant à un programme informatique de communiquer avec un serveur distant. Dans ce projet, l'API CLS permet de récupérer automatiquement les données satellite sans passer manuellement par l'interface web CLS.

Un script Python a été développé pour :



La connexion à l'API CLS s'effectue en deux étapes :

- Authentification avec un identifiant et un mot de passe afin d'obtenir un token sécurisé
- Utilisation de ce token pour récupérer les données envoyées par le module satellite

Les identifiants utilisés pour accéder à l'API sont les **mêmes que ceux de la plateforme web CLS**. Ainsi, un utilisateur disposant d'un compte CLS peut utiliser ces informations pour accéder automatiquement aux données via l'API.

Cependant, l'accès au service de **télémétrie n'est pas activé par défaut** pour tous les comptes. Pour obtenir l'autorisation d'accès aux données satellite et à l'API, il est nécessaire de faire une demande auprès du support CLS.

L'accès au service Telemetry doit être demandé en contactant: useroffice@groupcls.com, Une fois l'accès validé par CLS, l'utilisateur peut récupérer les données du device satellite via l'API sécurisée.

Ce système permet une récupération automatique, sécurisée et continue des données sans intervention manuelle sur la plateforme CLS.

ii. Base de données et traitement :

Les données récupérées depuis CLS sont enregistrées dans une base de données locale SQLite.



SQLite est une base de données légère intégrée directement au programme Python. Elle permet :

- de conserver un historique complet des mesures
- de stocker les données localement sur le serveur
- de permettre un affichage rapide des données
- d'éviter toute perte d'information

Chaque enregistrement contient : la date et heure, température, humidité, pression, luminosité

Les données sont décodées **automatiquement** depuis le format hexadécimal envoyé par le module satellite.

iii. Interface web de visualisation (dashboard) :

Un serveur web a été développé en **Python** à l'aide du framework **Flask**.



Flask est un framework web léger qui permet de créer rapidement une interface accessible depuis un navigateur internet. Cette interface constitue le tableau de bord principal du système ECOLOGGING.

Elle permet :

- d'afficher les données en temps réel
- de visualiser l'historique des mesures
- d'afficher des graphiques (température, humidité, pression, luminosité)
- de vérifier le bon fonctionnement de la station
- d'accéder aux données à distance

Des graphiques interactifs ont été développés avec la bibliothèque **Plotly**. Cette bibliothèque permet d'afficher des courbes dynamiques, de zoomer et d'analyser l'évolution des mesures dans le temps.

iv. Utilisation de GitHub dans le projet :

Le code du serveur a été stocké et versionné sur la plateforme GitHub.



GitHub est une plateforme de gestion de versions qui permet :

- de sauvegarder le code en ligne
- de suivre les modifications
- de collaborer entre membres du projet
- de déployer facilement l'application

Chaque modification du code est enregistrée dans le dépôt GitHub. Cela permet de sécuriser le projet et de conserver un historique complet du développement. GitHub est également utilisé pour déployer automatiquement l'application sur un serveur distant.

v. Hébergement du serveur sur Render :

Le serveur web a été déployé en ligne sur la plateforme Render.



Render est une plateforme cloud qui permet d'héberger des applications web accessibles depuis internet. Elle permet d'exécuter automatiquement le script Python et de rendre le dashboard accessible en ligne.

Configuration réalisée sur Render :

Les étapes de configuration ont été les suivantes :

1. Création d'un compte Render
2. Connexion du projet GitHub à Render
3. Déploiement automatique du code Python Flask
4. Configuration du port serveur (10000)
5. Lancement automatique du script en continu

Render exécute le programme Python en permanence. Un script interne interroge l'API CLS toutes les minutes afin de récupérer les nouvelles données satellite. Les données sont ensuite stockées dans la base SQLite puis affichées sur l'interface web.

vi. Fonctionnement global du serveur :

Le fonctionnement global de la partie serveur est le suivant :



vii. Rôle de la partie serveur dans le projet :

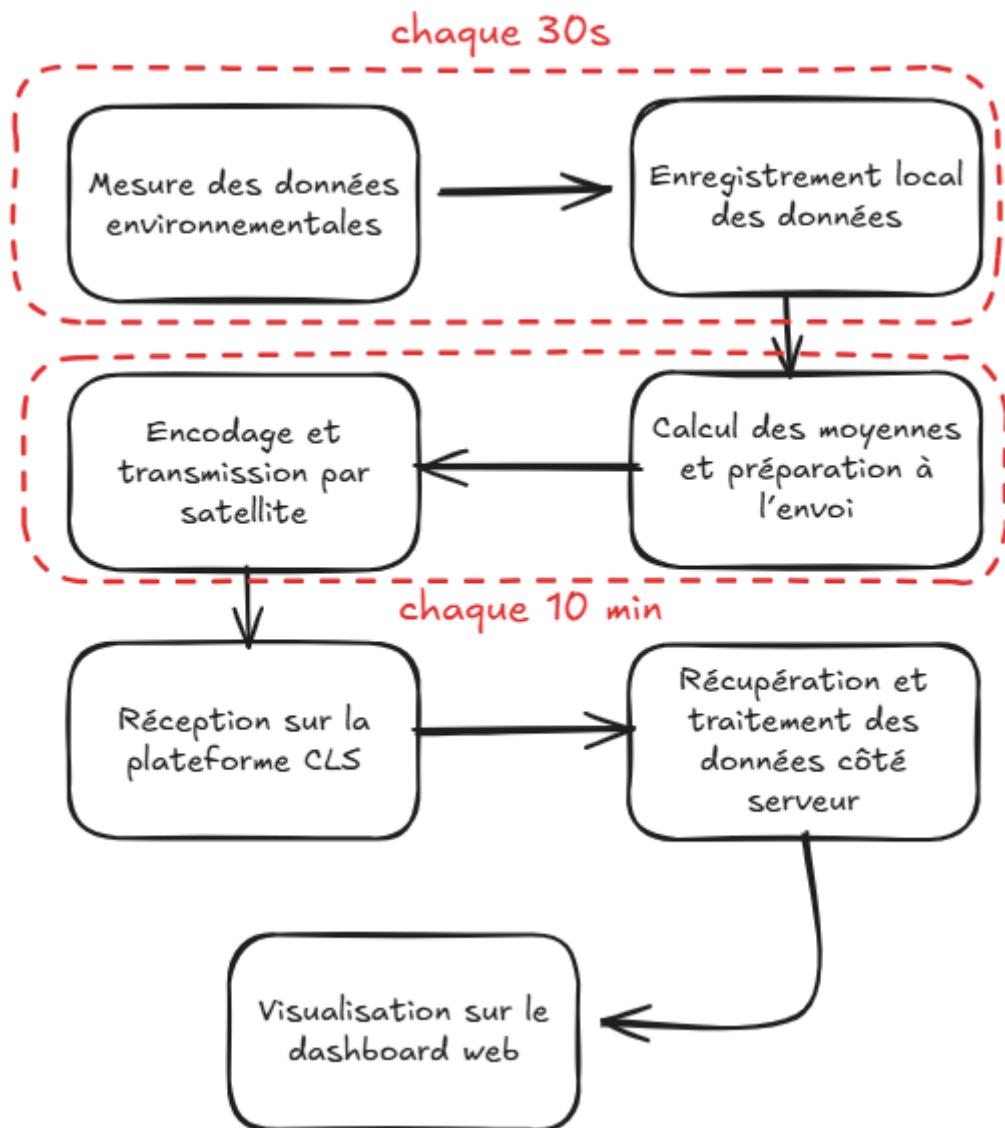
La partie serveur constitue le centre de supervision du système ECOLOGGING. Elle permet de vérifier le bon fonctionnement de la station, de consulter les données environnementales et d'analyser l'évolution des mesures.

5. Fonctionnement global du prototype 1 avec KIM2:

Ce prototype ECOLOGGING repose sur une chaîne complète d'acquisition, de transmission et de visualisation des données environnementales par satellite.

Le fonctionnement global du système est le suivant :

1. ***Mesure des données environnementales*** : Les capteurs installés sur la station terrain mesurent en continu la température, l'humidité, la pression atmosphérique et la luminosité. Ces mesures sont réalisées toutes les 30 secondes par l'Arduino Mega.
2. ***Enregistrement local des données*** : Chaque mesure est enregistrée localement sur une carte SD avec la date et l'heure fournies par l'horloge temps réel (RTC). Cela permet de conserver un historique complet des données même en cas de perte de communication satellite.
3. ***Calcul des moyennes et préparation à l'envoi*** : Afin de réduire la consommation énergétique et le nombre de transmissions satellite, les mesures sont cumulées pendant 10 minutes. Une moyenne est ensuite calculée pour chaque paramètre (température, humidité, pression, luminosité). Cette moyenne représente les données envoyées au satellite.
4. ***Encodage et transmission par satellite*** : Les valeurs moyennes sont encodées en format hexadécimal sur 16 bytes. L'Arduino envoie ensuite cette trame au module satellite KIM2 via une communication série UART. Le module KIM2 transmet les données aux satellites Kinéis, qui les relaient vers la plateforme CLS.
5. ***Réception sur la plateforme CLS*** : Les données envoyées par la station sont reçues sur les serveurs CLS. Elles sont stockées et mises à disposition via une interface web et une API sécurisée.
6. ***Récupération et traitement des données côté serveur*** : Un script Python hébergé sur un serveur distant se connecte automatiquement à l'API CLS. Ce script récupère les données envoyées par la station, les décode et les enregistre dans une base de données SQLite.
7. ***Visualisation sur le dashboard web*** : Les données sont affichées sur une interface web développée avec Python Flask. Cette interface permet de visualiser les mesures en temps réel, l'historique des données et des graphiques d'évolution des paramètres environnementaux.



Ce prototype démontre la possibilité de concevoir une station environnementale connectée par satellite, capable de transmettre des données depuis des zones isolées. L'architecture mise en place permet une surveillance continue, une visualisation à distance et une exploitation fiable des données collectées.

6. Validation expérimentale du système :

Afin de valider le bon fonctionnement du prototype ECOLOGGING, plusieurs tests réels ont été réalisés avec la station connectée au module satellite KIM2 et à la plateforme CLS.

Les captures d'écran suivantes présentent les différentes étapes de fonctionnement du système complet :

- suivi des transmissions depuis le moniteur série Arduino
- réception des données brutes sur la plateforme CLS après transmission satellite
- affichage des trames hexadécimales envoyées par la station
- décodage et visualisation des mesures sur l'interface web développée
- affichage des graphiques environnementaux en temps réel

```

20:25:29.553 -> ===== MOYENNE 10 MIN =====
20:25:29.553 -> Temp: 9.47
20:25:29.553 -> Hum: 77.98
20:25:29.553 -> Pres: 998.08
20:25:29.553 -> Lux: 1.84
20:25:29.553 -> SD AVG OK
20:25:29.553 -> Payload HEX = 03B31E7626FC0000000010000000000000
20:25:29.553 -> ---- ENVOI SAT ----
20:25:29.553 -> AT+TX=03B31E7626FC0000000010000000000000,00
20:25:29.633 ->
20:25:29.633 -> --- MESURE 30s ---
20:25:29.633 -> Temp: 9.39
20:25:29.633 -> Hum: 78.33
20:25:29.633 -> Pres: 998.02
20:25:29.633 -> Lux: 1.84
20:25:29.633 -> SD RAW OK
20:25:31.612 -> +OK
20:25:32.319 -> +TX=0,03B31E7626FC000000001000000000000000
20:25:59.551 ->

```

Tram Envoyé

Tram Envoyé avec succès

Figure 3: Moniteur série Arduino

The screenshot shows a web-based environmental monitoring system. At the top, there's a header bar with various links like 'Bienvenue | ENT UB', 'CELCAT Calendar', 'Annuaires des entre...', 'Technopole Pays Ba...', 'Biome', 'Bourse', 'SEAI', 'Master', and a search bar. Below the header, the main interface has a dark blue header with the 'CLS Environmental Monitoring' logo and a date range '2026-02-06 (UTC) - 2026-02-07 (UTC)'. There are three tabs: 'Devices', 'Messages' (which is selected), and 'Positions'. A red box highlights the 'Messages' tab. Below the tabs, a message count '158 résultats' is shown. A red box highlights the first message in the list. The message details are as follows:

ID de l'appareil	Réf. du programme	Date du message (UTC)	Données brutes	Taille (bits)	GPS Da
278910	8629	2026-02-07 19:25:32	03b31e7626fc0000000010000000000000	128	
278910	8629	2026-02-07 19:15:33	03a91f0226fb0000000010000000000000	128	

Tram reçue avec succès

Figure 2: Plateforme CLS:



Figure 1: Dashboard web ECOLOGGING

Ces résultats confirment que :

- la station mesure correctement les paramètres environnementaux
- les données sont encodées puis transmises par satellite
- les messages sont bien reçus sur la plateforme CLS
- le serveur récupère automatiquement les données via l'API
- les mesures sont décodées et affichées correctement sur le dashboard

Le système ECOLOGGING est donc pleinement fonctionnel de bout en bout : de la mesure terrain jusqu'à la visualisation des données sur une interface web accessible à distance.

7. Limites du prototype 1 avec KIM2:

Malgré les résultats positifs obtenus, le prototype présente certaines limites techniques liées principalement à la transmission satellite et aux conditions d'utilisation sur le terrain.

i. Fréquence réelle de réception des données :

Même si la station envoie une trame toutes les 10 minutes, la réception sur la plateforme CLS n'est pas systématique pour chaque envoi. En pratique, nous recevons en moyenne **3 à 4 messages par heure**. Cette différence s'explique par plusieurs facteurs :

- la couverture satellite (passage des satellites Kinéis au-dessus de la zone)
- les conditions météorologiques (pluie, humidité, obstacles)
- l'environnement d'installation (bâtiments, arbres, relief)
- les contraintes du réseau satellite IoT (temps de latence et priorités réseau)

Ainsi, la transmission satellite n'est pas continue comme une connexion GSM ou WiFi. Elle dépend du passage des satellites et des conditions de réception.

ii. Latence de transmission :

Un délai peut exister entre l'envoi de la trame par la station et sa réception sur la plateforme CLS.

Ce délai peut varier de quelques minutes à plusieurs dizaines de minutes selon la disponibilité des satellites.

iii. Consommation énergétique :

Même si l'envoi est optimisé toutes les 10 minutes, le module satellite reste un élément consommant de l'énergie.

Dans le cadre d'une station totalement autonome (solaire par exemple), l'optimisation de la consommation reste un point important à améliorer.

iv. Limitation du volume de données :

La taille des messages envoyés par satellite est limitée.

Le système utilise actuellement une trame de 16 bytes afin de rester compatible avec les contraintes du réseau et de limiter les coûts de transmission.

Cela limite la quantité d'informations pouvant être envoyées à chaque message.

v. Prototype en phase expérimentale :

Le système a été validé en conditions de test. Cependant, une utilisation longue durée sur le terrain nécessitera des tests supplémentaires de robustesse (autonomie énergétique, étanchéité, stabilité réseau).

8. Perspectives d'amélioration :

Plusieurs améliorations peuvent être envisagées afin d'optimiser et d'enrichir le système ECOLOGGING.

i. Ajout de nouveaux capteurs :

La configuration actuelle (LDA2) permet d'ajouter d'autres capteurs environnementaux sans modifier l'architecture globale. Par exemple, capteur de pluie, capteur de vent, capteur de qualité de l'air, capteur d'humidité du sol, etc...

ii. Optimisation énergétique :

Une version autonome alimentée par panneau solaire et batterie peut être développée pour un déploiement longue durée sur le terrain. L'utilisation de modes veille plus avancés permettrait de réduire la consommation globale du système.

iii. Amélioration de la transmission satellite :

L'optimisation du positionnement de l'antenne et des paramètres radio permettrait d'améliorer le taux de réception des messages.

Des tests sur différentes zones géographiques pourraient également être réalisés pour évaluer les performances du réseau.

iv. Extension de l'interface web

Le dashboard web peut être enrichi avec :

- export des données en CSV
- alertes automatiques en cas de valeurs anormales
- affichage multi-stations
- historique sur plusieurs mois
- interface mobile

v. Industrialisation du prototype

À long terme, le système pourrait être intégré dans un boîtier étanche et robuste pour un déploiement en conditions réelles.

Une carte électronique dédiée pourrait remplacer le montage prototype afin de réduire la taille et la consommation.

9. Conclusion :

Ce projet a permis de concevoir et de tester une station de mesure environnementale capable de transmettre des données par satellite. Le prototype développé permet de mesurer la température, l'humidité, la pression atmosphérique et la luminosité, puis d'envoyer ces informations vers les satellites Kinéis à l'aide du module KIM2. Les données sont ensuite reçues sur la plateforme CLS, récupérées automatiquement via une API et affichées sur une interface web développée en Python. Cette interface permet de visualiser les mesures en temps réel et de consulter l'historique sous forme de graphiques.

Les tests réalisés ont confirmé la faisabilité d'un système autonome de mesure et de transmission par satellite. Malgré certaines limites liées à la fréquence de réception et aux conditions environnementales, le prototype fonctionne correctement et répond aux objectifs du projet.

Ce travail constitue une base solide pour le développement futur d'une station environnementale connectée, autonome en énergie et capable d'être déployée sur le terrain pour des applications scientifiques ou agricoles.

- **Contacts techniques :**

Nom	Organisation	Contact
Maud PAGEOT	CLS	useroffice@groupcls.zohodesk.eu
Polino Susanna	CLS	spolino@groupcls.com

III. Prototype 2 : Conception du système avec le module SMD Wing Arribada :

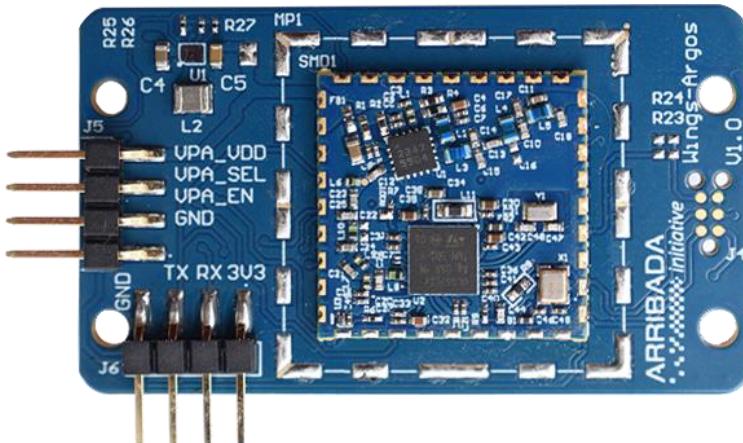
Le module WINGS-ARGOS V1.1 est une carte électronique développée pour permettre l'intégration d'une solution de communication satellite Argos dans un système embarqué. Il se présente sous forme de carte d'extension destinée à être connectée à un microcontrôleur hôte.

Sur le plan matériel, ce module ne constitue pas un modem satellite autonome. Il intègre principalement l'interface RF, les circuits d'adaptation et les lignes de contrôle nécessaires à l'émission vers le réseau Argos. Il nécessite l'association avec un modem satellite compatible pour assurer le traitement des trames et la gestion complète de la communication avec l'infrastructure satellite.

Le module expose plusieurs broches d'alimentation et de contrôle, notamment liées à l'amplificateur de puissance RF (VPA), ainsi que des interfaces de communication permettant son pilotage par un microcontrôleur externe. Son rôle dans l'architecture est d'assurer la liaison entre le système embarqué et la couche de transmission satellite, sous réserve de l'intégration du sous-système modem approprié.

Il est conçu pour communiquer avec un module Argos en UART.

Dans l'architecture de ce prototype, il était destiné à transmettre les données environnementales vers le satellite. L'absence d'un modem satellite a empêché la réalisation d'une transmission réelle.



1. Connexion du module satellite SMD Wing à l'Arduino :

Une interface UART a été configurée entre l'Arduino Mega et le module WINGS-ARGOS V1.1. Des trames série ont été émises depuis le microcontrôleur. Toutefois, en l'absence de modem satellite intégré et sans validation matérielle approfondie (analyse du signal ou confirmation firmware), il n'a pas été possible de confirmer le bon fonctionnement complet de la couche de communication série vers le module.

Pin du module	Pin de l'Arduino
TX	Pin 19 RX1
RX	Pin 18 TX1
VCC	OUT2 du Solar Power Manager V1.0
GND	GND du Solar Power Manager V1.0

La couche de communication satellite n'a pas pu être validée expérimentalement. En effet, le module WINGS-ARGOS V1.1 utilisé dans cette configuration ne constitue pas un modem satellite autonome et nécessite un composant modem externe pour assurer l'émission et la réception des trames RF. En l'absence de cet élément, aucune transmission satellite réelle n'a pu être réalisée, limitant la validation du système à la partie acquisition et interface locale.

Technologie de communication	Utilisation	Etat
I ² C	Capteurs ↔ Microcontrôleur	Fonctionnel
UART	Arduino ↔ SMD Wing	Fonctionnel
Satellite (Argos)	Transmission distante	Non fonctionnel

2. Difficultés rencontrées:

i. Problème majeur : absence de modem satellite :

La principale difficulté rencontrée au cours du projet concerne l'architecture matérielle du module utilisé. Le module WINGS-ARGOS V1.1 ne constitue pas un modem satellite autonome. Il s'agit d'une carte d'interface nécessitant l'intégration d'un modem satellite compatible pour assurer l'émission et la réception des données via le réseau Argos.

Cette caractéristique n'a pas été pleinement identifiée lors de la phase initiale d'intégration, ce qui a conduit à une impossibilité de valider la chaîne complète de transmission satellite.

En l'absence de modem intégré :

- aucune trame satellite ne peut être émise,
- aucune donnée ne peut être reçue depuis le réseau,
- les tentatives de transmission génèrent des erreurs système comme +ERROR=253 et +ERROR=1203 (Ces codes d'erreur indiquent que la commande de transmission n'a pas pu être exécutée par le système), sans qu'il soit possible de distinguer un défaut logiciel d'une limitation matérielle.

ii. Autres difficultés :

Outre l'absence de modem satellite, plusieurs difficultés secondaires ont été rencontrées au cours du projet. La documentation technique disponible pour le module WINGS-ARGOS V1.1 s'est révélée parfois incomplète ou peu explicite, notamment concernant l'architecture interne et les exigences matérielles exactes pour l'émission RF. Une ambiguïté initiale est apparue entre la notion de carte d'interface et celle de modem satellite autonome, ce qui a retardé l'identification de la véritable contrainte matérielle du système. Enfin, l'impossibilité d'établir une transmission effective a empêché tout test lié aux fenêtres de passage satellite, à la latence réseau ou au comportement du système en conditions réelles d'exploitation.

3. Tests réalisés et résultats :

i. Tests fonctionnels validés :

Test	Résultat
Acquisition capteurs	OK
Communication I ² C	Stable
Horodatage RTC	Fonctionnel
Génération payload	OK
Communication UART	OK

ii. Tests non validés :

Test	Statut	Cause
Transmission satellite	NON	Modem absent
Réception satellite	NON	Modem absent
Consommation RF réelle	NON	RF inactive
Passage satellite	NON	Aucun uplink

4. Synthèse :

Le système a été développé jusqu'au niveau de l'acquisition des données et de leur préparation en vue d'une transmission satellite. Les capteurs environnementaux ont été intégrés au microcontrôleur via le bus I²C, permettant la lecture et l'affichage des mesures sur le moniteur série. Les données collectées ont été structurées sous forme de payload conformément au format prévu pour une transmission ultérieure.

L'interface série UART entre le microcontrôleur et le module WINGS-ARGOS V1.1 a été mise en place. Toutefois, les tentatives d'envoi de commandes de transmission ont généré des messages d'erreur, sans permettre d'établir une communication effective avec une couche satellite. En l'absence de modem satellite intégré ou compatible, il n'a pas été possible de vérifier expérimentalement le comportement complet de la chaîne de transmission.

Ainsi, le projet a permis d'implémenter l'architecture logicielle et matérielle jusqu'à l'étape de préparation des données, mais la validation de la communication satellite n'a pas pu être réalisée.

5. Ce que nous aurions aimé tester :

Si un modem satellite compatible avait été disponible, plusieurs validations expérimentales auraient pu être réalisées afin d'évaluer les performances réelles du système en conditions d'exploitation. Il aurait notamment été possible de tester un uplink satellite effectif, c'est-à-dire l'émission réelle d'une trame vers le réseau et la confirmation de sa réception côté serveur. La récupération des données sur une plateforme distante aurait permis de vérifier l'intégrité des messages transmis ainsi que la conformité du format de payload.

Par ailleurs, une analyse quantitative aurait pu être menée afin de mesurer le taux de réussite des transmissions sur plusieurs cycles d'émission, en tenant compte des contraintes de visibilité satellite et des fenêtres de passage. Le temps de latence entre l'envoi de la trame et sa réception côté serveur aurait également constitué un indicateur important pour caractériser le comportement du système. Enfin, des mesures liées à la sensibilité RF, à la qualité du signal et à la stabilité de la liaison radio auraient permis d'évaluer la robustesse de l'architecture dans un contexte réel d'utilisation en environnement isolé.

6. Bilan :

Malgré l'impossibilité de valider expérimentalement la communication satellite, Ce prototype utilisant le SMD Wing a permis d'acquérir une compréhension approfondie de l'architecture complète d'une chaîne IoT par satellite, depuis l'acquisition des données jusqu'à la préparation des trames destinées à l'émission RF. La mise en œuvre matérielle et logicielle du système a permis d'intégrer les capteurs environnementaux, de structurer les données collectées et d'implémenter les mécanismes nécessaires à une transmission ultérieure.

Le prototype de ce projet a également mis en évidence les contraintes spécifiques aux systèmes de communication satellite, notamment la dépendance forte à l'architecture matérielle du modem et l'importance d'une analyse précise des composants avant intégration. Cette expérience a permis d'identifier clairement les limites actuelles du système et les conditions nécessaires à sa compléction.

Ainsi, bien que la transmission satellite n'ait pas pu être réalisée, l'architecture développée demeure cohérente et documentée. Le système pourrait être rendu pleinement opérationnel par l'intégration d'un modem satellite compatible, sans remise en cause majeure de la structure logicielle existante.