

# Rapport de stage

En vue de l'obtention du grade de Master

Développement et déploiement de capteurs connectés pour le suivi de consommation, le suivi de temps machine ainsi que de l'analyse vibratoire dans le milieu industriel.

**Enseignant référent** : M. Clency Perrine,  
Université de Poitiers

**Tuteur d'alternance** : M. Alexis Bernazeau,  
Kanope, 5 rue des Tours 59000 Lille

**Année universitaire** 2022-2023  
**Période d'alternance** : 03/2023 - 09/2023

**Réalisé par** : Grégoire William

**Destinataire** : Université de Poitiers,  
UFR sciences fondamentales et appliquées,  
Master Objets Connectés

## SOMMAIRE :

<b>1. Remerciements</b>	<b>3</b>
<b>2. Introduction générale</b>	<b>4</b>
<b>3. Présentation de l'entreprise</b>	<b>5</b>
3.1. Profil de l'entreprise	5
3.2. Exemples de projets réalisés par Kanope	6
<b>4. Projets de l'alternance</b>	<b>7</b>
4.1. Sujets abordés	7
4.2. Projet principal, 5G mMTC et vélos connectés	8
4.2.1. Contexte	8
4.2.2. Présentation	10
4.2.3. Apport personnel au projet	10
4.3. Projet RTM, firmware pour semelles connectées	19
4.3.1. Contexte	19
4.3.2. Présentation	19
4.3.3. Apport personnel au projet	19
4.4. Suivi d'un déploiement de solution de suivi de temps machine	26
4.4.1. Contexte	26
4.4.2. Présentation	26
4.4.3. Apport personnel au projet	27
4.5. Remise en marche et gestion de la flotte de 70 capteurs "MEL 5G"	29
4.5.1. Contexte	29
4.5.2. Présentation	29
4.5.3. Apport personnel au projet	30
<b>5. Gestion des projets</b>	<b>32</b>
5.1. Gestion macroscopique des projets	32
5.1.1. Gestion du temps et diagrammes de Gantt	32
5.1.2. Organisation et place dans l'équipe	34
5.2. Gestion du projet principal	35
5.2.1. Livrables attendus par le projet	35
5.2.2. Analyse des risques et des coûts	35
5.3. Éléments de gestion des autres projets	37
5.3.1. Méthode de gestion pour les autres projets	37
5.3.2. Livrables et documentations réalisées pour les autres projets	37
5.4. Problèmes rencontrés et solutions apportées	38
5.4.1. Typologie des difficultés rencontrées	38
5.4.2. État d'esprit pour la résolution des points durs.	38
<b>6. Conclusions</b>	<b>39</b>
6.1. Réalisations techniques	39
6.1.1. Complétion et avancement des projets	39
6.1.2. Usabilité des résultats.	39
6.2. Progrès personnels et montée en compétences	40
6.2.1. Utilisation des notions universitaires et du monde professionnel	40
6.2.2. Perspectives pour la suite	40

## 1. Remerciements

Après cette enrichissante année en alternance, je tiens à remercier les différentes personnes m'ayant accompagnées tout du long de cette période.

Ainsi, du côté de Kanope :

- Je tiens tout d'abord à féliciter Benoît Poulain, ingénieur en électronique, Responsable des Développements Techniques, pour la qualité de son accompagnement et de sa pédagogie au quotidien sur toutes les problématiques techniques ou méthodologiques dans le cadre du rôle de "Tuteur technique" qu'il a pris.
- Je souhaite tout autant remercier Alexis Bernazeau, Directeur des Opérations et mon tuteur durant cette année, pour ses conseils, durant ce semestre comme durant ceux qui ont précédés.
- Je veux également remercier Sacha Guy, Directeur Général, pour ses apports en méthodologie et rigueur.
- Il me semble aussi nécessaire de remercier Vincent Fenet, électronicien et développeur pour son aide sur les aspects plus hauts niveaux dans le firmware et sur la gestion des dépôts GIT.
- Enfin, je ne peux rédiger ces remerciements sans saluer et remercier le reste et l'ensemble de l'équipe Kanope pour son soutien, ses conseils, son dynamisme et pour l'environnement cohésif dans lequel j'ai eu la chance de pouvoir évoluer.

Et pour l'université de Poitiers :

- Je me dois avant tout de remercier sincèrement M. Clency Perrine, pour son suivi et son accompagnement, réalisant ainsi une fois de plus à la perfection son rôle de tuteur pédagogique.
- Je tiens par la même occasion à remercier l'ensemble de l'équipe pédagogique pour son investissement constant dans la formation.

## 2. Introduction générale

Lors de ma seconde année de Master en Objets Connectés, j'ai travaillé dans le cadre d'un contrat d'alternance avec l'entreprise Kanope sur une solution de suivi de consommation énergétique et de suivi de temps de fonctionnement machine.

Je participe également à la conception d'un système de remontée de données d'entraînement pour l'équipe de France de cyclisme sur piste, dans le cadre d'un projet de déploiement de réseau 5G lauréat du plan France Relance.

Cette alternance se déroule sous la tutelle de M. Clency Perrine du côté de l'université de Poitiers et sous celle de M. Alexis Bernazeau pour Kanope. Le présent rapport porte sur la période de stage en entreprise ayant eu lieu entre le 27 mars et le 8 septembre 2023.

Ce document permet de présenter l'entreprise et mes missions, ainsi que les différents aspects de gestion de projet et perspectives qui en découlent.

## 3. Présentation de l'entreprise

### 3.1. Profil de l'entreprise

Kanope (raison sociale Sparkling Tech) est une société lilloise fondée en 2016 par Sparkling Partners, un startup studio qui a créé de nombreuses entreprises innovantes dans des secteurs variés (santé, retail, agroalimentaire, tech...).

Kanope est un studio IoT avec pour objectif de devenir sur toute sa chaîne de valeur, un expert de la donnée (création, collecte, exploitation, optimisation), ceci combiné à la capacité d'accompagner et conseiller ses partenaires sur le pilotage de processus reposant sur l'utilisation des données produites.

Il s'appuie pour cela sur trois piliers, indépendants et complémentaires :

- Ses capacités en analyse et conseil permettent de déterminer une architecture IoT adaptée et d'utiliser les données générées pour accompagner le pilotage et l'optimisation des processus clients, le tout à partir de l'identification de leurs besoins métier.
- Son bureau d'études électronique qui configure des infrastructures IoT et développe des projets sur mesure pour des clients (start-ups, PME, grands groupes).
- Son équipe de développement qui, au-delà de la réalisation de projets clients, a mis en place une plateforme universelle de gestion d'objets connectés, servant d'outil de base pour la visualisation et l'utilisation des données.

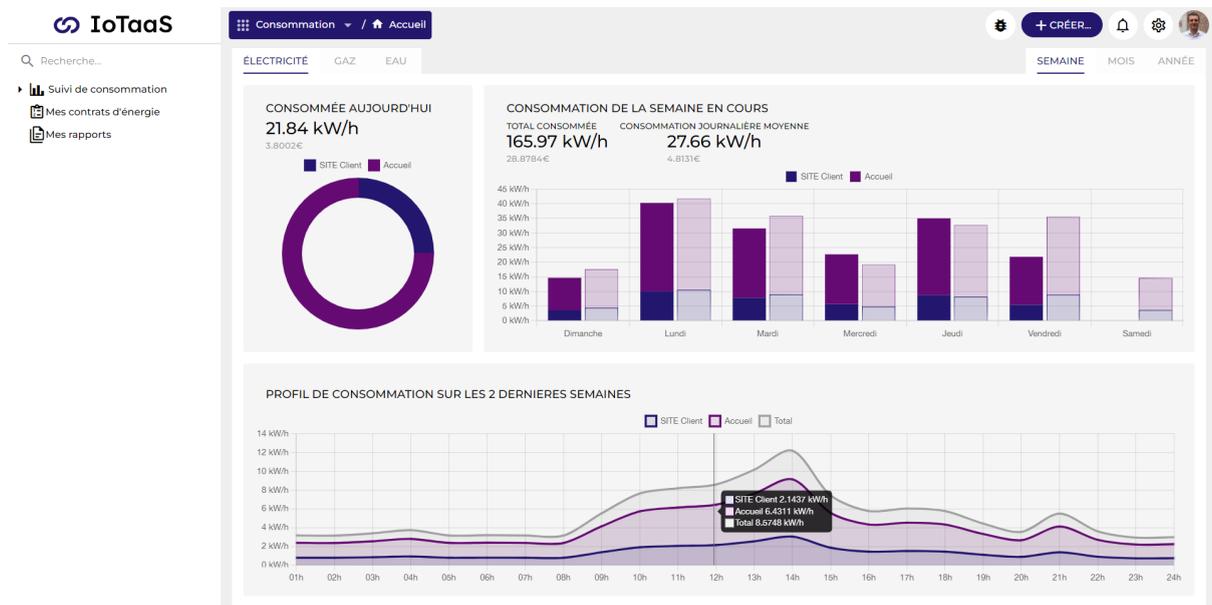
La finalité est naturellement de traiter un nombre importants de données, matérialisant ainsi l'objectif d'optimisation de processus industriels complexes impliquant de multiples paramètres. Ceci est rendu possible par l'utilisation d'outils comme principalement du machine learning.

Aujourd'hui, Kanope regroupe 10 personnes : 1 directeur général, 1 directeur des opérations, 1 business developer, 3 développeurs informatique et 4 électroniciens, dont moi-même.

## 3.2. Exemples de projets réalisés par Kanope

Afin d'illustrer l'activité de l'entreprise, voici quelques exemples des projets marquants (en cours ou terminés) :

- **Javelot** : création d'un dispositif de remontées de température pour les silos à grains. Projet utilisant carte électronique, boîtier, remontée de données et visualisation. (L'entreprise a bénéficié d'une importante levée de fonds en 2022 (>10 Mio€))
- **Enedis**, projet Belette : système de mesure et de validation des données de consommations électriques, projet nécessitant la maîtrise de l'électronique et de l'aspect "puissance" électrique. (Prix de l'innovation Enedis 2022)
- **IoTaaS** (IoT as a Service) : plateforme web universelle de visualisation de données et de gestion d'objets connectés.
- **5G mMTC** : en partenariat avec un consortium d'entreprises, participation à l'un des use-case du projet, à savoir la création d'un système de remontées de données de vélos de compétition sur un réseau 5G local.



Exemple - Page de suivi de consommation d'électricité sur IoTaaS

## 4. Projets de l'alternance

### 4.1. Sujets abordés

Durant mon alternance, j'ai eu l'occasion d'aborder plusieurs projets, avec différents niveaux d'implication. Kanope réalise des projets dans trois catégories : les projets électroniques, les projets logiciels et les projets de conseils pour le pilotage de processus par la donnée, qui regroupent électronique et logiciel. Mes projets font partie de la catégorie électronique, avec pour l'un d'entre eux une implication dans la dimension conseil :

- Les projets électroniques, avec :
  - Le projet 5G mMTC, vélos connectés pour la FFC.
  - Le projet "RTM", semelles connectées.
  - Le projet "Mel 5G", déploiement d'une flotte de capteurs.
  
- Le projet électronique, faisant également partie d'un projet de conseil :
  - Suivi post installation d'une solution pour le suivi de temps machine.

Pour plus de lisibilité, les projets, leurs contextes, les réalisations et résultats associés seront présentés ensemble, selon un ordre suivant la part de temps qu'ils ont occupés durant cette partie de l'alternance .

La partie Gestion de projet reviendra ensuite sur ceux qui ont eu le plus d'impacts sur ces derniers mois.

## 4.2. Projet principal, 5G mMTC et vélos connectés

### 4.2.1. Contexte

Le projet des vélos connectés fait partie du plan France Relance sur la “Souveraineté dans les réseaux de télécommunications afin d’accélérer les applications de la 5G.”

Plusieurs sujets avancent en parallèle, l’un concernant le suivi du réseau électrique avec EDF, et le second, pour lequel Kanope a été retenu en tant qu’intégrateur IoT, sur le développement d’outils de performance pour la FFC.

La Fédération Française de Cyclisme (FFC) a proposé un projet visant à améliorer le processus de remontée des données lors des phases d’entraînement en cyclisme sur piste, ce qui permettrait de faire entre autres des calculs de traînées et de permettre l’étude et l’amélioration de la position des athlètes. Ce projet rejoint des idées de développement de stade connecté, d’où l’utilisation du côté mMTC (massive Machine Type Communication) de la 5G qui permet de faire transiter une très grande quantité de données.

Ce projet est réalisé avec de nombreux partenaires, comme Sequans pour les puces 5G ou Amarisoft pour les cœurs de réseau 5G. Il y a également des partenaires publics avec l’INRIA qui étudie les technologies 5G. Tous les partenaires sont regroupés dans le tableau, visible sur la page suivante, pour illustrer la diversité d’acteurs participant à ce projet.

Le but de ce “plan de relance” est de permettre à différents acteurs industriels français d’acquérir des connaissances sur l’utilisation et l’innovation autour de cette nouvelle génération pour les télécommunications. Si nous développons une solution pour un cas d’application donné, nous aurons aussi une approche globale afin de tester largement les différentes technologies impliquées et de voir comment cela peut être réutilisé sur d’autres projets propres à Kanope.

Membres du consortium selon leurs fonctions

<b>Responsables des Use cases</b>		
<b>Fournisseurs de technologies 4G/5G</b>		
<b>Cybersécurité et virtualisation</b>		
<b>Intégrateurs IoT</b>		
<b>Académiques, recherche</b>		 
<b>Organismes responsables et organisateurs du plan France relance sur la 5G.</b> 		
	 <p>Contrat Stratégique de filière Infrastructures numériques</p>	

### 4.2.2. Présentation

L'objectif est de développer un "concentrateur" récupérant les informations des capteurs d'un vélo (puissance, vitesse, cadence...) connectés en sans fil par ANT+ ou BLE. Ces données sont ensuite renvoyées sur un réseau local en 5G.

À cela, on ajoute une "station bord de piste" pouvant récupérer les conditions atmosphériques et servir de compte-tour. Il y a également tout un côté remontée de données, sécurité et accès réseau pour connecter la solution et la rendre utilisable à distance pour la FFC.



*Vélodrome de St Quentin en Yvelines, base d'entraînement pour la FFC et sélectionné pour les Jeux olympiques de 2024.*

*Notre système serait déployé depuis le centre de la piste, où sont postées les équipes.*

### 4.2.3. Apport personnel au projet

Dans la continuité du premier semestre, je me suis occupé sur cette période de plusieurs parties du firmware, sur lequel je n'avance pas seul étant donné l'échelle et les échéances du projet. J'ai également généré les premières données en 5G pour valider le bon fonctionnement de nos solutions.

Au cours de ces derniers mois j'ai donc travaillé sur du firmware et différents tests, m'ayant occupé d'Avril à fin Juin avec :

- Le développement d'un parseur de messages AT
- La création d'un prototype fonctionnel pour la station bord de piste
- Des tests sur les performances du MQTT et du COAP en nb-IoT

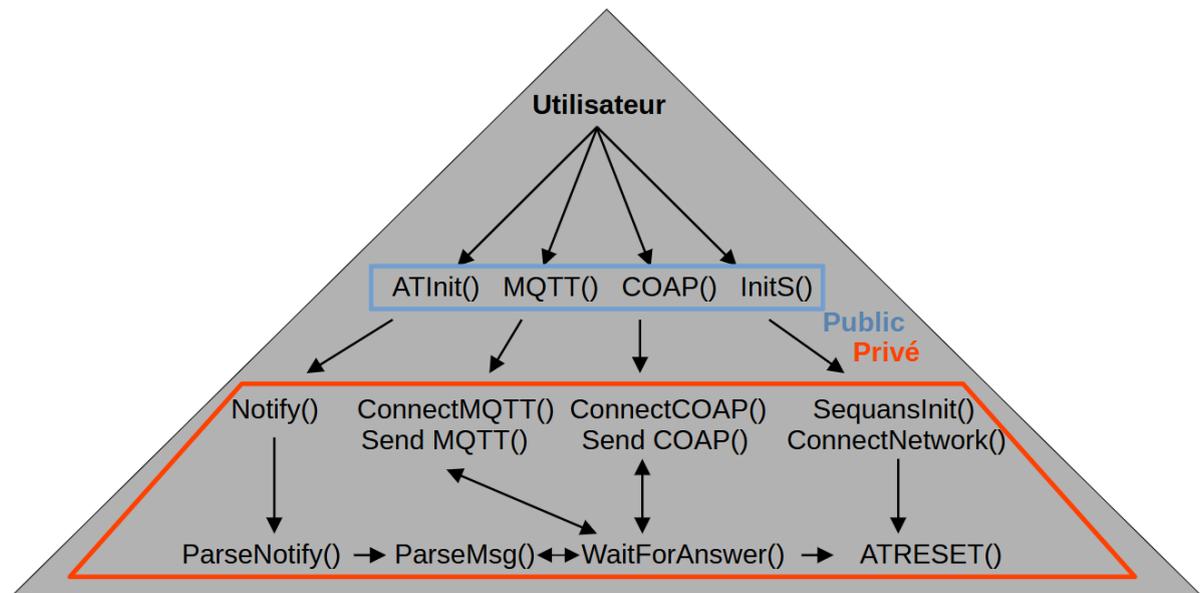
## Développement d'un parseur de messages AT pour la board Sequans.

Un des points centraux de ce projet est la communication en 5G. Pour ce faire nous utilisons un chip GM02S de Sequans, qui se commande par le biais de commandes AT. La solution finale dispose d'un nRF52 en tant que MCU hôte, qu'il faudra donc munir d'une librairie permettant de facilement utiliser les différentes commandes AT lors de l'intégration finale.

Mon premier objectif est donc de développer cette librairie, les commandes AT utilisées par Sequans leur étant propres, il n'y a pas de code préexistant réalisant cette fonctionnalité, même si des exemples partiels de cas similaires peuvent être trouvés.

Pour ce qui est des séquences de commandes AT à utiliser, je peux m'appuyer sur les tests effectués en LTE-M lors du premier semestre et au début de celui-ci, j'ai alors une liste de commandes déterminées à implémenter.

Pour éviter différentes erreurs, les fonctions suivent une "hiérarchie" telle que résumée sur le schéma ci dessous :

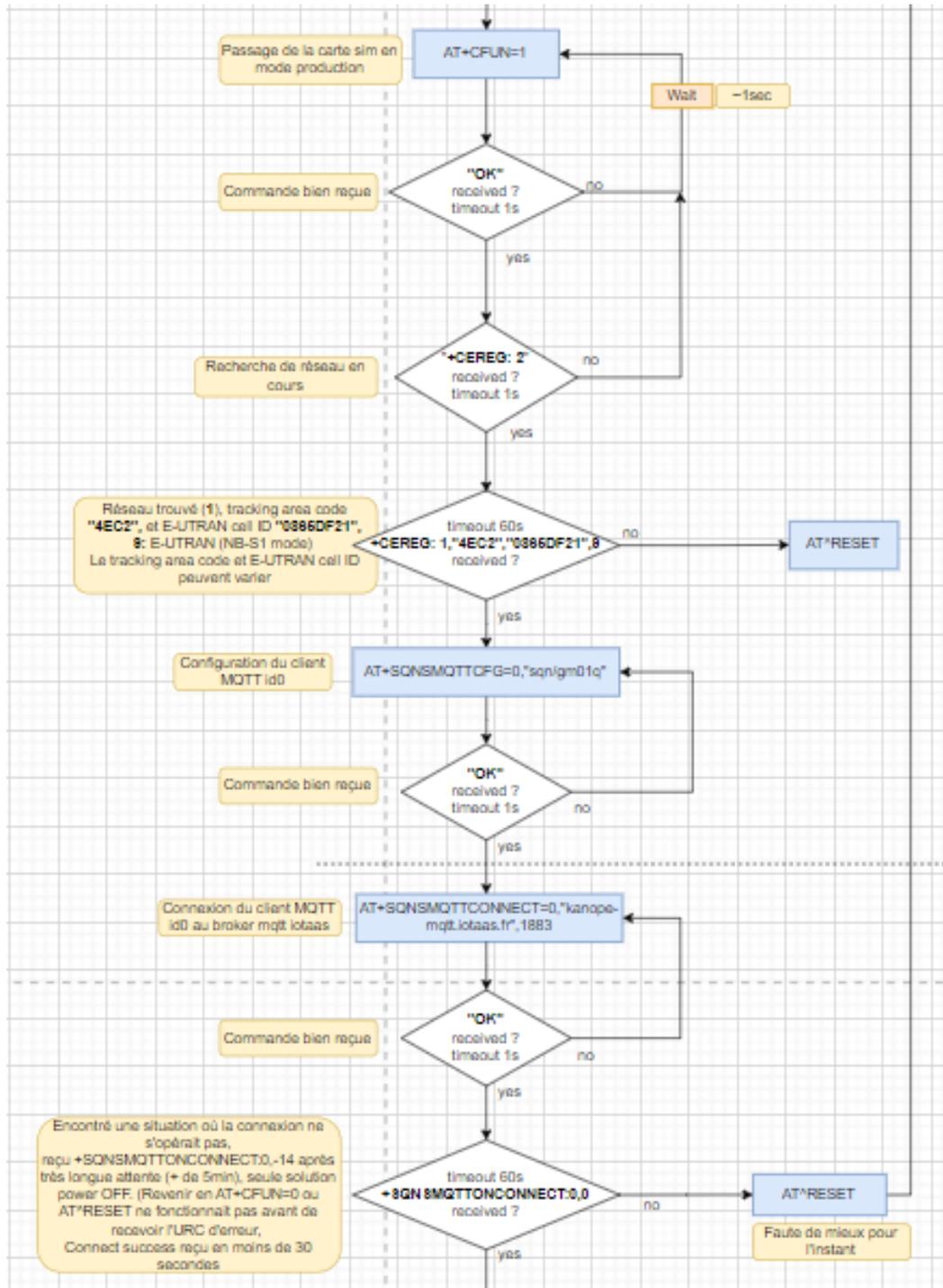


*Architecture reliant les fonctions de la librairie des commandes AT*

J'avais eu lors du développement des problèmes de fonctions qui s'appelaient en boucle dans certains cas de bords (échecs successifs de connexions réseaux par exemple) ce qui finissait par faire crasher le NRF52 car il avait trop de fonctions en cours.

Cette architecture unidirectionnelle (les fonctions du haut appellent celles du bas) permet de faciliter la lisibilité, le débogage et renforce la stabilité du code.

Les fonctions InitS, COAP et MQTT utilisent des séquences de commandes AT qui sont décrites dans la documentation Sequans, on retrouve ici un extrait du logigramme suivi pour se connecter au réseau avant de rejoindre un serveur MQTT :



Logigramme des commandes AT pour la connexion au réseau et à notre serveur MQTT de test

Ce même logigramme continue pour l'envoi de messages MQTT, la connexion et l'envoi de messages à un serveur COAP (plus sur ces deux protocoles dans une partie ultérieure)

Au final, la librairie AT propose les fonctionnalités suivantes :

- Connexion au réseau par utilisation de la carte sim
  - Réseaux NB-IoT (5G) comme LTE-M (tests)
- Connexion à un broker MQTT
  - Envoi de données sur des topics prédéfinis
  - Validation de l'acknowledge de réception
  - Mécanisme de reconnexion en cas de perte du réseau ou broker.
- Connexion à un serveur COAP
  - Envoi de requêtes de POST de datas
  - Parsing et validation de la réponse serveur
  - Mécanisme de reconnexion en cas de perte du réseau ou de déconnexion serveur.

Tous ces éléments sont, comme abordés plus tôt, utilisés par les codes externes via 4 fonctions élémentaires.

Pour fonctionner avec mon environnement de développement Mbed et celui de mon collègue Martin D. qui utilise Zephyr pour le concentrateur, nous avons développé la librairie de manière à ce qu'elle soit portable entre les deux plateformes.

Sa structure en est donc affectée et elle se découpe en quatre parties, avec deux classes applicatives et deux classes systèmes :

- Une classe bas niveau, reprenant la gestion des ports Uarts, des buffers et des interruptions.
- Une classe haut niveau qui gère les messages reçus et à envoyer, indépendamment de la plateforme où elle a été déployée.

Les deux classes systèmes permettent de mutualiser les noms de fonctions selon les différentes plateformes, pour Mbed comme pour Zephyr on aura :

- Une classe d'utilitaires pour définir un timer et une méthode de print
- Une classe buffer pour gérer les différences sur la réception, le stockage et l'envoi de données transitants par Uart.

Cette structure nous permet d'avoir un code lisible séparant le code applicatif, système et plateforme-dépendant.

## Création de la station bord de piste et premières remontées de données en 5G (NB-IoT).

Pour pouvoir tester les communications 5G et le parseur présenté précédemment, j'ai travaillé sur la mise en place d'un prototype filaire pour la station bord de piste qui comprend des capteurs météo et un capteur laser compte tour.

J'ai d'abord eu à adapter les bibliothèques existantes à l'environnement Mbed avec lequel je travaille pour cette partie du projet. J'ai ensuite utilisé l'évent queue de l'OS Mbed pour gérer les différentes tâches de la station bord de piste.

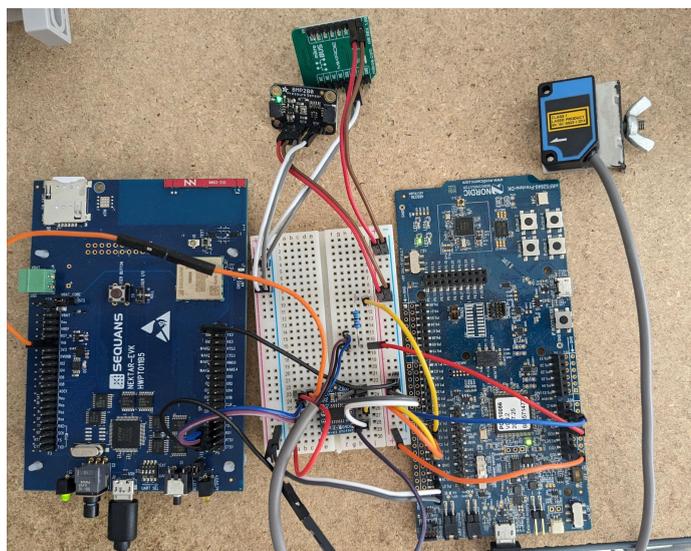
Côté hardware, j'ai eu à relier les différents boards de démo utilisés, à savoir :

- Un NRF52840 DevKit pour le MCU principal
- Une carte Nektar Monarch 2 Sequans pour la 5G
- Un capteur BMP280 pour la mesure de pression
- Un capteur HTU31D pour la mesure de température et humidité
- Un capteur laser 24V NPN WL100L E1131 pour le comptage des tours.

Selon les périphériques, différents types de communication étaient utilisés :

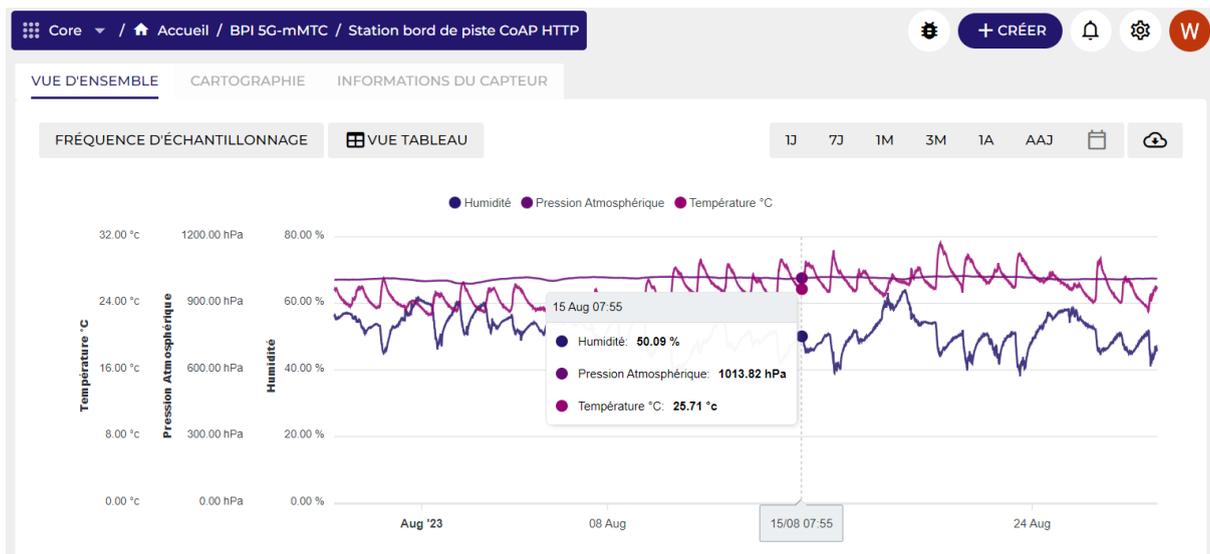
- Un bus i2C pour les capteurs météo
- Un port GPIO avec Pull UP pour le compte tour
- Une série de ports Uarts pour la carte Sequans. On notera d'ailleurs quelques problèmes rencontrés suite à une utilisation peu conventionnelle des pins, là où d'habitude les pins RX et TX de l'hôte se relient aux pins marquées respectivement TX et RX (on croise donc les connexions) la board sequans fait ce croisement en interne et on veillera donc à brancher le TX sur le TX et le RX sur le RX.

Ci dessous une image du prototype filaire de la station bord de piste assemblé :



*Prototype pour la Station bord de piste*

Après ce développement, nous avons mis en place la visualisation des données météo produites sur notre plateforme IoTaaS. Nous avons eu deux versions, les données remontées en MQTT, puis en COAP. Nous voyons ci-dessous les données de température, pression et humidité envoyées depuis le début des tests continus du COAP fin Juillet.



*Fenêtre IoTaaS pour les données météo de la Station Bord de Piste en COAP*

De l'autre côté, nous avons le capteur compte tour qui ne tourne pas continuellement et n'est donc pour l'instant pas connecté sur IoTaaS. On peut visualiser ses données sur le PC de logging qui tourne en parallèle des tests pour avoir un historique de ce qu'il s'est passé sur le système quelle que soit l'heure.

Côté Station bord de piste, on formate les données du compte tour telles que :

```
snprintf(toursmessage,100,"Nb tour %d, duree tour %d ms
\n",nbtours,tourduration);
```

Le compte tour est relié à une interruption qui générera la demande d'un envoi de message au plus tôt après avoir sécurisé le "numéro" et la durée du tour dans le contexte de l'interruption. On protège ainsi les données sensibles d'un décalage si une tâche est déjà en cours.

Si on regarde un extrait des logs lors d'un passage coupant le rayon laser du capteur, on a :

<pre>[2023-08-28 17:37:45.252] 325 s : Nombre de tour : 1, duree du dernier tour : 23288 ms [2023-08-28 17:37:45.252] msg size : 32 [2023-08-28 17:37:45.252] wried : AT+SQNCOAPSEND=0,0,2,32 [2023-08-28 17:37:45.269] [2023-08-28 17:37:45.269] We want '&gt;' [2023-08-28 17:37:45.269] payload : Nb tour 1, duree tour 23288 ms [2023-08-28 17:37:45.276] [2023-08-28 17:37:45.276] wried : Nb tour 1, duree tour 23288 ms [2023-08-28 17:37:45.276] [2023-08-28 17:37:45.276] On attend OK [2023-08-28 17:37:45.385] *****On a recu: &gt; ***** [2023-08-28 17:37:45.385] out [2023-08-28 17:37:45.385] msg parsed [2023-08-28 17:37:45.433] *****On a recu: OK ***** [2023-08-28 17:37:45.433] out OK [2023-08-28 17:37:45.448] On attend +SQNCOAPRING: 0, [2023-08-28 17:37:46.231] *****On a recu: +SQNCOAPRING: 0,8531,1,2,205,39 ***** [2023-08-28 17:37:46.246] out +SQNCOAPRING: 0,8531,1,2,205,39 [2023-08-28 17:37:46.246] send success [2023-08-28 17:37:46.246] status Send compte tour : 1</pre>	<p>← Interruption, signal pour un premier tour fini.</p> <p>← Commande AT de préparation de l'envoi.</p> <p>← Attente du caractère "&gt;"</p> <p>← Saisie de la Payload</p> <p>← Message écrit sur le port Uart</p> <p>← Attentes des différentes réponses ("&gt;","OK" ...)</p> <p>← Notification "COAPRING", le serveur COAP nous répond.</p> <p>← On valide le bon envoi du message du compte tour.</p>
---	--

Le mécanisme d'envoi pour les données météo est le même. Les logs se complexifient lors d'échecs car ceux-ci génèrent des tentatives répétées puis un reset de la board Sequans. Un cas d'erreur est présenté lors de la section suivante sur le coap pour illustrer les problèmes rencontrés et servir de base pour l'explication des développements encore en cours.

## Tests pour utiliser le protocole COAP au lieu du MQTT

Lors des tests en MQTT, nous nous sommes aperçus que son aspect TCP ralentissait grandement les échanges de données en NB-IoT au point que cela poserait problème dans la suite du projet, j'ai donc mis en place un plan de test pour le COAP qui est lui basé sur l'UDP et qui requiert donc moins d'allers retours pour la validation d'un même message. L'idée est de tester le débit en fonction de la taille des messages envoyés, et le taux de succès d'envoi des messages.

J'ai pour ce faire pu réutiliser le parseur de commandes AT et l'implémenter dans un code faisant les différents tests voulus, avec un suivi de ceux-ci sur un port Uart relié à un terminal sur ordinateur en utilisant un outil dédié comme Tera Term.

Ces tests ont mené à un tableau de résultats de performance et débit nous permettant de confirmer l'idée de passer par le COAP.

FUNCTIONS	TEST PROCEDURE	STATUS	RESULTS
<b>Test de Débit et de Fiabilité du Coap</b>	<a href="https://git.kanope.app/bpi/Test_NB-IoT">https://git.kanope.app/bpi/Test_NB-IoT</a>		
Tester le débit Max de la connexion Coap	Envoyer successivement plusieurs paquets de taille max au serveur Coap.(5 fois) - Chronométrer le délai pris par envoi à partir des timestamps des commandes AT dans TerraTerm. - Valider que chaque envoi s'effectue sans erreur	Done	Débit max obtenu : 20*1024char : 1159,89 o/s 40*512 char : 784,83 o/s 80*256 char : 392,92 o/s  Points de mesures : (20*1024char) Bureau : 799,14 o/s Sous sol : 1246,92 o/s Etage (fenêtre) : 1297,70 o/s (historique complet en feuille 2)  Sans erreur (y/n) : y
Tester la fiabilité de la connexion au serveur Coap	Par connexions / déconnexions successives : - Calculer le temps de connexion au serveur (timestamps des commandes AT) - Déterminer le taux de succès de connexions Par connexions successives : - Calculer le délai avant déconnexion du serveur	Done	Taux de fiabilité de connexion : ToR  Délai 1ere connexion : 1,2 s Délai de reconnexion : 209,3ms  Délai de déconnexion : NC (s)

*Extrait des résultats du test des performances du COAP*

Une fois ces tests faits, nous avons mis en place le test continu du COAP, dont vous avez vu les données en page 15. À la suite de ce mois de tests, je me suis aperçu que le device était très régulièrement rejeté du réseau. La raison la plus probable, est qu'un envoi de données toutes les 5 minutes est trop peu pour maintenir la connexion.

Cela soulève deux problèmes :

- Il faut s'assurer que les déconnexions sont liées à cela et donc programmer un envoi de données plus fréquent.
- Il faut mettre en place un mécanisme de renvoi des messages après reconnexion pour ne pas perdre les messages compromis.

Page suivante, on retrouve un log entre un envoi réussi et le suivant échouant, j'ai retiré certaines lignes que nous avons déjà vu et qui n'ont pas d'intérêt dans ce contexte pour gagner en lisibilité :

```
[2023-08-09 12:36:50.986] COAP config success
[2023-08-09 12:36:51.002] writed : AT+SQNCOAPSEND=0,0,2,49
[2023-08-09 12:36:51.002] We want '>'
[2023-08-09 12:36:51.002] payload : Temp 26.689857, hum 46.523232,
pres 1017.005493
. . .
[2023-08-09 12:36:51.017] On attend OK
[2023-08-09 12:36:51.173] *****On a recu: OK *****
[2023-08-09 12:36:51.173] out OK
[2023-08-09 12:36:51.189] On attend +SQNCOAPRING: 0,
[2023-08-09 12:36:51.940] *****On a recu: +SQNCOAPRING:
0,8530,1,2,205,56 *****
[2023-08-09 12:36:51.940] out +SQNCOAPRING: 0,8530,1,2,205,56
[2023-08-09 12:36:51.940] send success
[2023-08-09 12:36:51.956] Initialized : 1
. . .
[2023-08-09 12:40:36.706] 1204760 s      Temperature : 26.720070
C, humidite : 45.488670, pression : 1017.003723 hPa
[2023-08-09 12:40:36.722] writed : AT+SQNCOAPSEND=0,0,2,49
[2023-08-09 12:40:36.722] We want '>'
[2023-08-09 12:40:36.722] payload : Temp 26.720070, hum 45.488670,
pres 1017.003723
. . .
[2023-08-09 12:40:36.737] On attend OK
[2023-08-09 12:40:36.837] *****On a recu: +SQNCOAPCLOSED:
0,"NETWORK" *****
```

← On vient de réussir à se connecter au COAP

← On envoie un premier jeu de données météo.

← On reçoit bien le "OK"

← Le serveur acknowledge l'envoi d'un message.

← premier envoi réussi, connexion COAP correctement initialisée.

← 5 minutes plus tard, on reçoit une nouvelle mesure météo.

← Mais lorsque l'on attend le "OK" on reçoit un "COAPCLOSED:0,Network", notre connexion (Id 0) au serveur COAP à été fermée par le réseau.

On voit bien à travers ce log le problème tel que rencontré, une mise à jour du code pour ajouter un historique de données non envoyées et faire un test avec plus de débit utilisé est en cours de développement et devrait être fini peu de temps avant la remise de ce rapport.

## 4.3. Projet RTM, firmware pour semelles connectées

### 4.3.1. Contexte

En tant que Startup Studio, Sparkling Partners a lancé un projet de semelles connectées pour suivre la température des pieds de personnes diabétiques et permettre aux équipes médicales de mieux les accompagner dans le traitement de leur maladie.

Dans ce cadre, Kanope occupe une place centrale, en s'occupant du design d'une carte électronique, du firmware embarqué et des premières applications de démonstration. D'autres entités du groupe Sparkling s'occupent de la gestion du projet ou du développement d'une application mobile permettant de s'interfacer avec les données récupérées.

### 4.3.2. Présentation

Une carte électronique est embarquée dans la semelle et communique en BLE. Il y a possibilité de faire des sessions d'enregistrement, où les mesures sont acquises selon une période donnée puis stockées dans un fichier, et il y a la possibilité d'activer un mode "Live" qui émet les données de température en temps réel suivant un intervalle donné également.

Ces messages sont récupérés par une application sur téléphone portable qui pourra traiter les données et les présenter à l'utilisateur final (médecin ou patient).

### 4.3.3. Apport personnel au projet

J'ai été mobilisé sur ce projet pour développer une version de démonstration pour les communications BLE. En créant un parseur de commandes AT dans le même état d'esprit que le projet 5GmMTC, on veut parfaitement simuler le comportement qu'aura le device auprès de l'application sur smartphone. L'idée est de ne pas avoir à utiliser un prototype complet pour le développement de l'application mais seulement un NRF52 qui ne sera relié à aucun capteur et donc à aucune donnée. Arriver à un démonstrateur livrable à l'entreprise responsable du développement de l'application mobile nécessite plusieurs étapes :

- Développer le parseur de commandes AT et les fonctions liées selon les spécifications.
- Intégrer des données factices et un comportement répétable pour faciliter les tests.
- Tester le programme de manière automatisée et corriger les problèmes relevés (bugs).

## **Développement d'un parseur de messages AT pour les communications BLE.**

Lorsque je rejoins le projet, des spécifications précises rédigées par Benoît existent et permettent de savoir exactement quelles seront les fonctionnalités à implémenter.

La différence avec le parseur AT précédent est que je ne dois pas me limiter à l'envoi de messages liés à des fonctions prédéfinies par l'utilisateur mais que je dois reconnaître une multitude de messages entrant et les traiter selon leurs spécificités.

Le programme suit donc une architecture relativement simple :

- Un callback de réception de message BLE est appelé lorsqu'un nouveau message arrive. Ce callback appelle le parseur de messages.
- Le parseur trie le message reçu, isole les variables présentes dans le message si nécessaire et appelle la fonction AT dédiée.
- La fonction AT dédiée renvoie un message d'erreur particulier pour chaque cas empêchant la réalisation de la commande (suivi d'une machine d'états). Si rien n'empêche sa réalisation elle appelle la fonction propre demandée
- La fonction demandée agit sur le système et sur la machine d'état, réalisant alors la commande envoyée puis renvoi une confirmation de succès.
- La fonction AT dédiée peut alors utiliser le return précédent pour émettre un message de confirmation ou les données demandées en BLE.

La première étape était donc de créer le parseur et de créer les fonctions correspondantes pour chaque commandes AT recevable possible (13 commandes)

Il faut ensuite isoler les paramètres envoyés dans les commandes AT (durée, fréquences, noms de fichiers...)

Je peux alors compléter les différentes fonctions pour cette version avec des données factices ou pour les versions suivantes des acquisitions réelles. Les fonctions "propres" seules seront à modifier pour obtenir les données "réelles" pour la température mais aussi pour l'espace mémoire disponible, la batterie restante...

À la fin de cette phase de développement, nous avons un ensemble de fonctions qui permettent d'avancer sur le reste du projet en ayant la partie "longue" qu'est le parseur déjà faite.

Celui ci regroupe les fonctions présentées ci dessous :

On réutilise le callback BLE du code d'exemple de départ sous une forme modifiée pour gérer nos messages parfois plus longs.

```
static void bt_receive_cb(struct bt_conn *conn, const uint8_t *const data, uint16_t len)
```

Nous avons par la suite les fonctions principales suivantes qui seront appelées par leurs commandes AT respectives :

```
enum States Get_State()
enum Responses Start_Rec(int periode, char* Nom_Session)
enum Responses Stop_Rec()
enum Responses Get_Files_Names(int* nb_files, char** NomFichier)
enum Responses Get_File(char* NomSession, char** file)
void File_Transfert (int continueTransfer)
enum Responses Erase_file(char* NomSession)
enum Responses Format_Memory()
int Get_Bat_Voltage()
enum Responses Get_Available_Memory(int* AvailableMemoryValue)
bool Set_adv_interval(int interval)
enum Responses Start_Live(int periode)
enum Responses Stop_Live()
```

Chaque fonction ci dessus à également une fonction d'appel AT qui validera en amont la possibilité d'exécuter la fonction par rapport à la machine d'état : (renvoi d'erreur si demande de démarrage d'enregistrement alors qu'un live est en cours par exemple)

```
void AT_Get_State()
void AT_Start_Rec(int periode, char* NomSession)
. . .
```

On utilise ce code pour ensuite créer un démonstrateur complet et "fonctionnel".

## Création d'un démonstrateur utilisant des données factices.

Je peux désormais implémenter les fonctions plus "intelligentes" du code de démonstration. Plusieurs fonctionnalités sont modifiées pour pouvoir simplement récupérer un maximum de comportements de manière fiable. On a ainsi :

- Un jeu de températures allant de 20,0 à 22,0°C par pas de 0,1° et encodés en *least significant byte first* comme spécifié et suivant une formule donnée permettant de travailler avec des entiers sur 16 bits chacun.
- Les fichiers d'enregistrements sont remplacés par une liste pouvant contenir 7 noms de fichiers, on stock uniquement le nombre d'enregistrements fait pour chaque fichier.
- La mémoire disponible dépend du nombre de fichiers créés, indépendamment de leur longueur (répétabilité et facilité de test de la fonctionnalité "mémoire basse")
- A chaque demande de récupération d'un fichier, on regardera le "nombre d'enregistrements" associés à ce nom et on enverra les données générées le bon nombre de fois.
- Lors du passage en mode live, on renverra toujours la même trame de données

Est également implémenté de manière plus définitive :

- Le changement de l'interval d'advertising BLE (temps entre 2 messages d'affichage aux clients BLE)

A la fin de cette étape, chaque commande AT envoyée retourne le comportement attendu, comme sur le produit final, ce qui va permettre de commencer le développement de l'application mobile.

Avant cela, il reste à valider le code produit pour s'assurer de son respect absolu des spécifications pour éviter d'avoir à envoyer d'innombrables patches lors des tests pour l'application.

Sur la page suivante, on retrouve un échange de commandes AT entre un nrf52 et un terminal BLE sur portable. On teste à cette occasion les différentes fonctions liées au lancement d'un enregistrement, auquel on ne laisse le temps de prendre qu'une seule mesure.

On remarque que la donnée renvoyée semble "corrompue" (@^@^V]>]f...) Ce n'est en fait pas le cas, nos données sont compressées de manière à ce que quelles que soit leurs valeurs, elles puissent tenir sur 58 octets, les données sont donc travaillées au niveau du bit et de l'octet, ce qui rend leur lecture sous forme ascii inappropriée.

On teste le maximum de commandes avec la version de démonstration, on aura donc un faux système d'utilisation de la mémoire et une tension de batterie choisie arbitrairement, comme on peut le voir avec les dernières commandes envoyées.

```
17:55 [status icons]
Terminal [back] [trash] [menu]
17:54:44.795 Connecting to Test RTM device ...
17:54:47.198 Connected
17:54:51.492 AT
17:54:51.556 OK
17:54:52.461 AT+START_REC=10,TestDeDemonstration
17:54:52.545 OK
17:54:53.795 AT+STOP_REC
17:54:53.850 OK
17:54:56.762 AT+GET_FILES_NAMES
17:54:56.821 OK
17:54:56.910 NB files : 1
17:54:57.135 TestDeDemonstration
17:54:58.679 AT+GET_FILE=TestDeDemonstration
17:54:58.756 OK
17:54:58.890 NB blocks : 1
17:54:59.070 >[hex dump]
17:55:00.699 ^B
17:55:05.349 AT+GET_AVAILABLE_MEMORY
17:55:05.416 OK
17:55:05.506 Free space = 86%
17:55:07.366 AT+ERASE_FILE=TestDeDemonstration
17:55:07.441 OK
17:55:10.449 AT+GET_AVAILABLE_MEMORY
17:55:10.501 OK
17:55:10.590 Free space = 100%
17:55:14.651 AT+GET_BAT_VOLTAGE
17:55:14.730 OK
17:55:14.820 Vbat = 1498 mV
17:55:22.771 AT+FORMAT_MEMORY
17:55:22.831 OK
```

*Échanges entre le portable et le prototype de semelle connectée.*

Lors de cet échange, on a :

- Réalisé un enregistrement (AT+START et AT+STOP)
- Récupéré le nom des fichiers enregistrés en mémoire (AT+GET\_FILES\_NAMES)
- Récupéré le fichier en mémoire (AT+GET\_FILE)
- Récupéré la mémoire disponible, écrasé le fichier présent puis formaté la mémoire (AT+GET\_AVAILABLE\_MEMORY, AT+ERASE\_FILE et AT+FORMAT\_MEMORY)
- Récupéré la tension de la batterie.

## Utilisation d'un banc de tests pour le debugging et la livraison à nos partenaires.

Pour les tests, je détermine le plan de tests à suivre avec mon collègue Vincent F. qui "découvre" le sujet et qui programmera le banc de test en python (la personne programmant le code à tester et le code de tests n'étant préférablement pas la même).

Une fois d'accord sur l'ensemble des tests à réaliser, nous commençons à mettre en place le banc de test et je corrige les quelques problèmes lorsqu'il me les signale.

Nous trouvons toutefois un problème majeur. J'avais jusque là utilisé un smartphone Kanope pour mes tests et lors de l'automatisation, Vincent à pris un ESP32 en tant que client Bluetooth. Or, l'ESP s'avère moins coopératif que le smartphone et nous avons eu des problèmes avec la taille maximale des messages que nous pouvons envoyer, le paramètre changé dans mon code ne suffisant pas selon les devices et leur version bluetooth.

En changeant la méthode d'envoi et de réception des messages, nous arrivons à contourner ce problème et le mettre de côté. Il y aura également d'autres bugs plus mineurs, le principal étant lié à la taille du buffer utilisé pour les noms de fichiers.

Ces tests et le fonctionnement de la démonstration ont ensuite pu être présentés au responsable du projet chez Sparkling, avant qu'une carte NRF52 préprogrammée puisse être envoyée aux développeurs de l'application mobile.

*Setup pour le banc de test avec l'ESP32 et le NRF52 partageant la pin Reset*



On observe ci-dessous un extrait du code pour le test 11, qui est celui sur lequel j'ai le plus travaillé pour couvrir des effets de bords ou certains bugs auxquels je m'attendais.

Je teste premièrement l'envoi de noms trop longs (plus de 30 caractères selon les spécifications) et valide leur bonne gestion.

Je valide ensuite que toutes les commandes ne pouvant s'exécuter pendant un enregistrement renvoient bien le bon code d'erreur.

```
def test11(e): #Errors (Extrait)

    e.check(b'AT', b'OK')
    e.check(b'AT+START_REC=10,TESTAVECNAMETROPLONG3232MEMETROP', b'ERROR')
    e.check(b'AT+START_REC=10,TESTAVECNAMETROPLONG3131MEMETRO', b'ERROR')
    e.check(b'AT+START_REC=10000,TESTAVECNAMETROPLONG3030MEMETR', b'OK')
    e.check(b'AT+STOP_REC', b'OK')

    e.check(b'AT+START_REC=10,TEST', b'OK')
    e.check(b'AT+GET_FILE=TEST', b'ERROR, REC IN PROGRESS', testid='F4')
    e.check(b'AT+FORMAT_MEMORY', b'ERROR, REC IN PROGRESS', testid='H4')
    e.check(b'AT+ERASE_FILE=TEST', b'ERROR, REC IN PROGRESS', testid='G4')
    e.check(b'AT+GET_AVAILABLE_MEMORY', b'ERROR, REC IN PROGRESS', testid='J4')
    e.check(b'AT+START_LIVE=32', b'ERROR, REC IN PROGRESS', testid='M2')
    e.check(b'AT+STOP_REC', b'OK')
    return 'PASS'
```

Ici on retrouve le log produit lors de l'exécution du code de test, avec lequel on valide le bon passage du test 11.

```
Test #11

OK: b'AT' -> b'OK'
OK: b'AT+START_REC=10,TESTAVECNAMETROPLONG3232MEMETROP' -> b'ERROR'
OK: b'AT+START_REC=10,TESTAVECNAMETROPLONG3131MEMETRO' -> b'ERROR'
OK: b'AT+START_REC=10000,TESTAVECNAMETROPLONG3030MEMETR' -> b'OK'
OK: b'AT+STOP_REC' -> b'OK'
OK: b'AT+START_REC=10,TEST' -> b'OK'
OK: b'AT+GET_FILE=TEST' -> b'ERROR, REC IN PROGRESS'
OK: b'AT+FORMAT_MEMORY' -> b'ERROR, REC IN PROGRESS'
OK: b'AT+ERASE_FILE=TEST' -> b'ERROR, REC IN PROGRESS'
OK: b'AT+GET_AVAILABLE_MEMORY' -> b'ERROR, REC IN PROGRESS'
OK: b'AT+START_LIVE=32' -> b'ERROR, REC IN PROGRESS'
OK: b'AT+STOP_REC' -> b'OK'
PASS
```

Au départ, les tests avec des noms trop longs ne passaient pas dans deux scénarios : si le nom faisait 31 caractères, un test dans le code passait pour  $\leq 31$  au lieu de  $< 31$ . Et si la commande AT atteignait 40 caractères au total (somme des plus longs paramètres possibles autorisés). Ces deux bugs ont pu être corrigés en modifiant les variables définissant le maximum autorisé pour ces deux aspects.

## 4.4. Suivi d'un déploiement de solution de suivi de temps machine

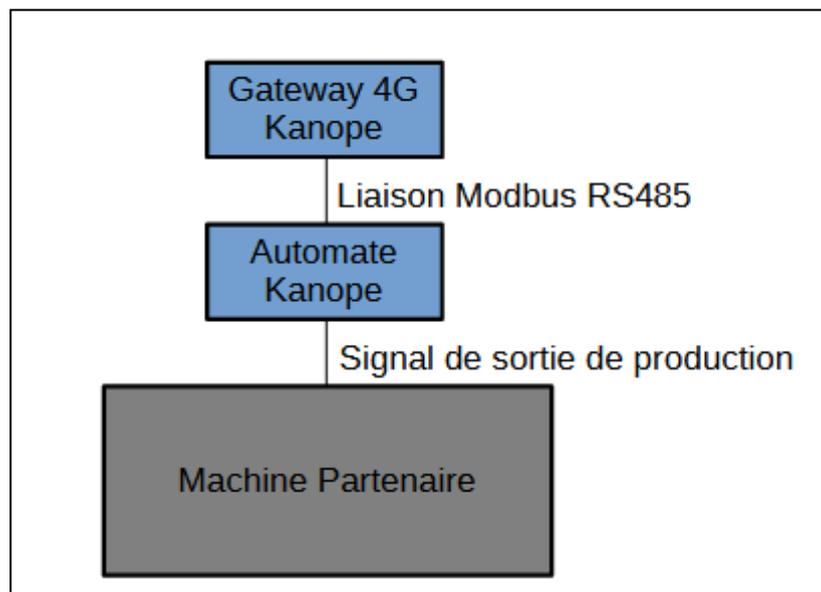
### 4.4.1. Contexte

Lors du premier semestre, Kanope a conclu un accord avec un partenaire industriel pour déployer et tester la solution IoTaaS, avec les modules de suivi de temps machine et de suivi de consommation énergétique en conditions réelles sur un site de production. Ce projet nous permet donc de tester de nombreux systèmes développés, dont une solution de suivi de temps machine et une solution de suivi énergétique sur lesquelles j'ai travaillé durant le premier semestre et mon stage de M1. Nous avons pu déployer nos capteurs début 2023 et commencer à utiliser les remontées de données pour fiabiliser nos solutions.

### 4.4.2. Présentation

Nous avons 4 machines équipées d'un système de suivi de cadence de production reliées à des gateways 4G et deux compteurs énergétiques connectés en LoRaWAN. Ce projet est la première occasion pour nous de déployer ces dispositifs et nous cherchons sur cette seconde partie de mon alternance à valider les données reçues et corriger les éventuelles dérives que nous trouverons.

Ci dessous l'architecture utilisée pour le côté suivi de temps machine, qui est l'aspect sur lequel j'ai eu à revenir lors de ce semestre :



*Architecture générale pour la solution de suivi de temps machine*

### 4.4.3. Apport personnel au projet

Notre partenaire nous a indiqué que les données que nous remontions étaient incorrectes et l'ensemble de l'équipe projet s'est donc penchée sur la question et nous avons remis l'ensemble de la chaîne d'acquisition en question pour isoler et corriger le problème. J'ai pu de mon côté travailler sur une robustification de l'algorithme embarqué dans les automates de suivi du temps machine. Sur demande de Benoît, j'ai rapidement pu mettre en place et suivre l'efficacité d'un module optocoupleur HW complémentaire pour correctement isoler le signal de sortie de production.

#### Patch de correction en vue de l'amélioration de la fiabilité du système.

Quand bien même il était dur de voir un lien entre les variations parfois erratiques des données remontées et les configurations/algorithmes embarqués, j'ai repris le code pour le robustifier en ajoutant d'un côté de nombreuses sécurisation entre autres sur les overflows de variables, mais aussi en ajoutant un topic MQTT nous permettant d'obtenir en un int32 la valeur du compteur de production et le timestamp associé pour faciliter le débogage et comprendre ce qu'il se passait.

J'ai ensuite pu déployer cette mise à jour sur les dispositifs installés chez notre partenaire après vérifications en local que tout fonctionnait bien.

J'ai par la suite suivi les données remontées par le même système en y ajoutant un optocoupleur sur la version que nous avons dans notre lab pour isoler le signal en entrée et déterminer le taux d'erreur restant (0,05%).

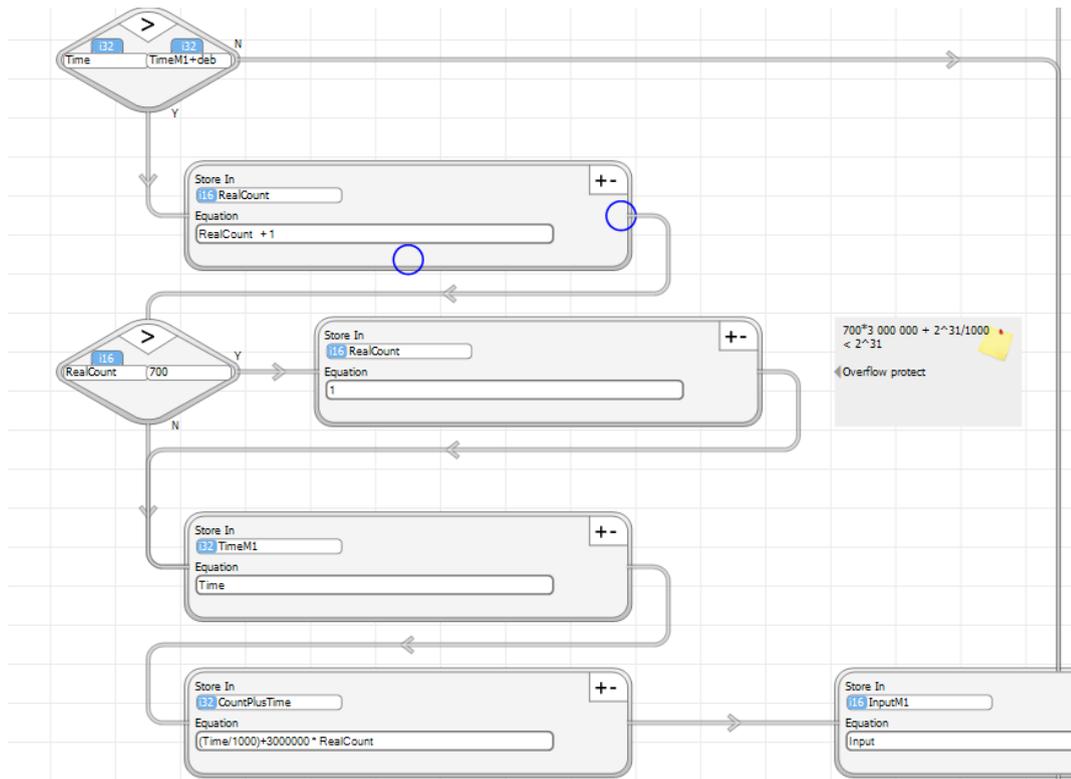
Après coup il s'avérera que le problème venait bien de l'isolation du signal d'entrée et que Benoît avait eu la bonne intuition en voulant tester l'ajout d'un optocoupleur. La mise à jour à quand même permis de solidifier le code et de faciliter le suivi du fonctionnement en cas d'autres problèmes dans le futur.

Sur la page suivante, on peut voir le coeur de la mise à jour produite pour l'algorithme avec l'ajout d'un debouncing customisé du signal et d'un champ "*CountPlusTime*" qui conserve les informations utiles au suivi de la production si les messages nous arrivent désynchronisés et sans corrélation avec leurs heures de départ.

On trouve juste ci-dessous les données reçues en MQTT pour une machine, si on fait les calculs, on trouve bien  $113 \times 10^6$  ( $339 \times 10^6$ ) + le timestamp 012 231s (3h, 23min, 51 secondes après le démarrage de notre système)

```
today at 15:01:22 Mottez/Time_Mottez_2/Count 113
today at 15:01:22 Mottez/Time_Mottez_2/CountAndTime 339012231
today at 15:02:09 Mottez/Time_Mottez_2/Count 114
today at 15:02:09 Mottez/Time_Mottez_2/CountAndTime 342012279
```

On retrouve ici la gestion du debounce pour être sûr de ne pas considérer deux fois la même impulsion de production machine. On voit également le calcul de compression du count+Temps qui est limité par un uint32, ce qui nous oblige à tourner par cycles de 700 produits pour pouvoir logger les deux informations simultanément.



Extrait du code pour les automates de suivi de temps machine.

Ici, la configuration mise à jour de la gateway, avec les nouveaux topics permettant un meilleur suivi de l'activité et une détection pour des défaillances réseau par exemple.

Publishing and Subscribing to MQTT Topics for Modbus

MQTT_Topic_to_Pub/Sub	Payload_Bit	Payload_Ana	Pub ↑ / Sub ↓	SID	Type_Reg.	Address	Poll	DBd	EN	
Time_6/Count	@[0,1]	@	PUE	1	I16	0	1	1	<input checked="" type="checkbox"/>	Delete
Time_6/Blink	@[0,1]	@	PUE	1	I16	11	5	5	<input checked="" type="checkbox"/>	Delete
Time_6/CountAlive	@[0,1]	@	PUE	1	I32	15	1	1	<input checked="" type="checkbox"/>	Delete
Time_6/ISAlive	@[0,1]	@	↓SI	1	BIT	0	1	1	<input checked="" type="checkbox"/>	Delete
Time_6/TimeAndCount	@[0,1]	@	PUE	1	I32	18	1	1	<input checked="" type="checkbox"/>	Delete
Time_6/Count2	@[0,1]	@	PUE	1	I16	17	1	1	<input checked="" type="checkbox"/>	Delete
Time_6/TimeAndCount2	@[0,1]	@	PUE	1	I32	20	1	1	<input checked="" type="checkbox"/>	Delete

Ajouts à la configuration d'une gateway de suivi de temps machine

## 4.5. Remise en marche et gestion de la flotte de 70 capteurs “MEL 5G”

### 4.5.1. Contexte

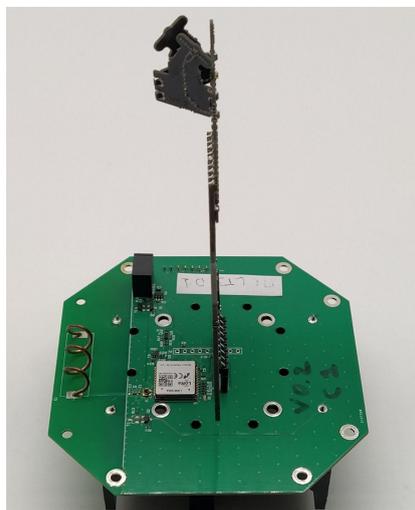
Dans le cadre d'un partenariat avec la Métropole Européenne de Lille (MEL) et sous l'impulsion du CNRS et de l'Université de Lille encadrant une thèse sur ce sujet, Kanope a développé un capteur permettant de mesurer la puissance des ondes électromagnétiques et ainsi de suivre l'impact du déploiement de la 5G. Ce capteur utilise une antenne conçue spécifiquement pour lui par un partenaire de Kanope, Wavecontrol. Suite à la phase de conception, 70 capteurs ont été déployés sur des candélabres de deux quartiers de Lille pour permettre de cartographier l'évolution dans le temps des ondes électromagnétiques présentes.

En plus des ondes radios, le capteur mesure les variables d'environnement classiques telles que la température et l'humidité. Les données acquises sont ensuite envoyées sur le réseau LoRaWANn. Les données reçues sont renvoyées vers un serveur de la MEL ainsi que vers notre outil de suivi de flotte de capteurs, loTaaS.

### 4.5.2. Présentation

Après quelques mois sur le terrain, il a été observé que certains capteurs indiquaient un très haut taux d'humidité et que les valeurs de puissances remontées commençaient à être incohérentes par moment pour certains capteurs.

Nous avons donc à ce moment étudié ce qu'il se passait et après validation et tests préliminaires de Wavecontrol validant la non destructivité de l'opération, nous avons testé de tropicaliser quelques cartes pour éliminer leur sensibilité à l'humidité. Après avoir pu valider l'efficacité de cette modification, il a été décidé d'un commun accord entre les membres du projet de récupérer l'ensemble des capteurs déployés. Cela permettant alors de les tropicaliser, valider leur bon retour en fonctionnement nominal et les redéployer dans Lille suivant le même plan que celui précédant l'intervention.



*Capteur d'ondes 5G  
développé par Kanope*

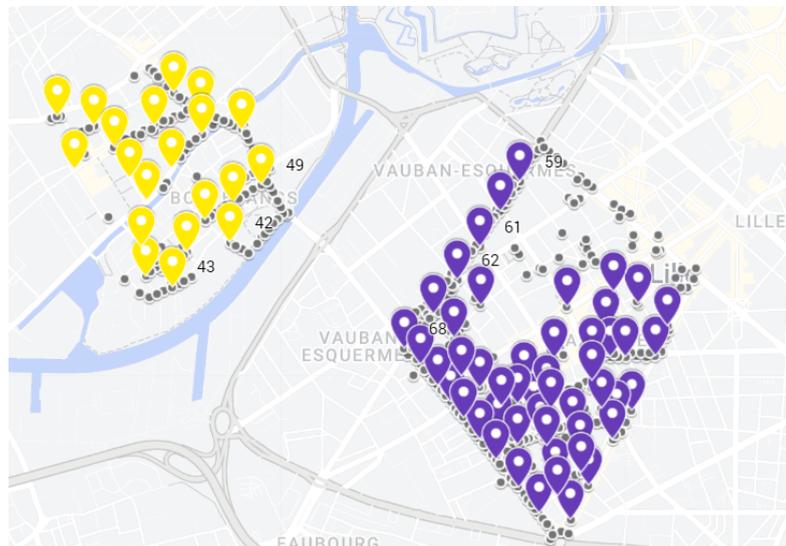
### 4.5.3. Apport personnel au projet

#### Préparation à la récupération et à la tropicalisation des capteurs.

Pour optimiser les ressources déployées à résoudre ces problèmes, il fallait correctement analyser et anticiper la situation en amont pour faire en sorte que tout se passe au mieux. Cette partie se découpe donc en deux étapes : la préparation de la récupération des capteurs par un intermédiaire et la préparation de la phase de tropicalisation.

Pour la préparation de la récupération des capteurs, il était question à un moment de le faire en interne pour éviter de mobiliser un intervenant extérieur. J'ai donc récupéré des informations terrains complémentaires à leur lieux d'installation, comme leur hauteur et les conditions environnantes (accessibilité...). Avec ces données nous avons pu regrouper les informations nécessaires pour pouvoir prendre les bonnes décisions et répartir correctement les rôles et responsabilités quant à la récupération des boîtiers. J'ai par la suite également remanié une carte reprenant tous les emplacements de capteurs.

En complément à cette partie, j'ai également préparé la phase de tropicalisation en mettant en place les outils et un espace de travail permettant de suivre le protocole développé par Benoît pour modifier les 70 capteurs en l'espace de deux jours.



*Carte des capteurs déployés dans Lille  
et adaptée pour la récupération des capteurs*

Note : *Tropicalisation* : Application d'une couche de protection acrylique (Acrylic Protection Layer) par pulvérisation qui forme un film de protection sur la surface où le produit a été appliqué, la protégeant alors des éléments extérieurs tel que l'humidité.

## Gestion de la remise en route de la flotte de capteurs

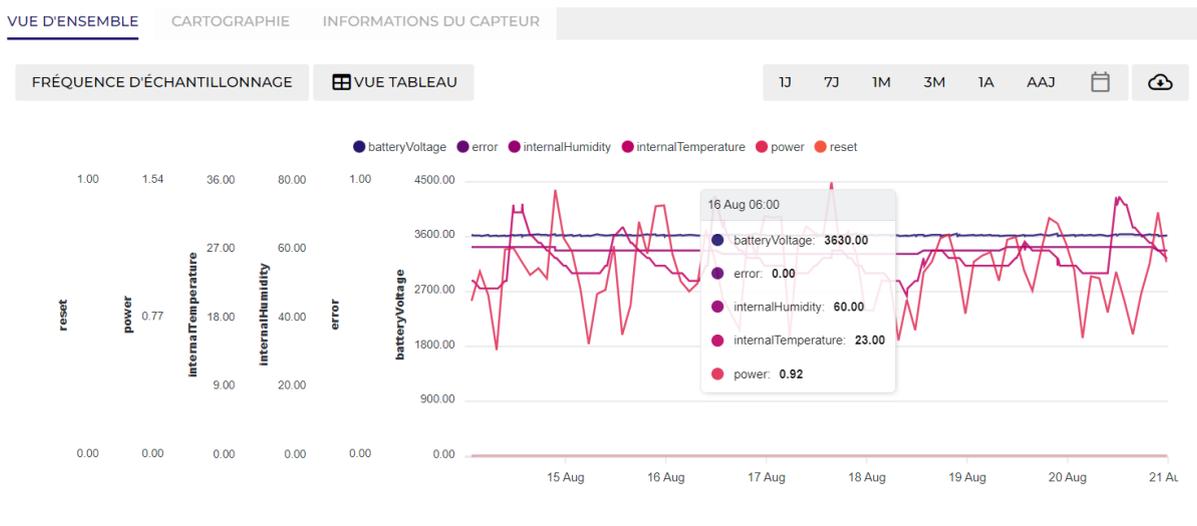
Une fois la tropicalisation effectuée par Benoît et celui-ci partant en congés à ce moment, il m'est revenu la responsabilité de veiller à la bonne reconnexion au réseau des capteurs puisque ceux-ci avaient été éteints (piles retirées) pour la tropicalisation.

Après que les capteurs aient été reconnus par le réseau, il fallait synchroniser leurs horloges à partir de l'uplink reçu et en utilisant un outil dédié à la création de downlink de resynchronisation. Une fois les capteurs mis à jour, il restait à leur envoyer un downlink de passage en mode production pour que ceux-ci collectent et émettent des données de puissances électromagnétiques.

Ces opérations sont simples en soi, mais valider les opérations, gérer les problèmes éventuels sur les 70 capteurs en simultanément s'est avéré être une tâche à part entière.

Parmi les problèmes rencontrés, j'ai observé que certains capteurs retournaient des valeurs aberrantes et après une rapide enquête sur ce qu'il pouvait se passer, la majorité des capteurs retournant des valeurs normales étaient tous regroupés au même endroit et après les avoir déplacés de quelques mètres à peine, ceux-ci ont repris des valeurs normales, échappant alors à une perturbation non identifiée les affectant.

J'ai ensuite pu regrouper les différents éléments pour les donner à Benoît à son retour quelques jours plus tard, lui permettant d'avoir une vision sur ce qu'il s'était passé pendant son absence.



*Page de suivi d'un capteur sur loTaaS, c'est avec cet outil et des pages telles que celle-ci que je pouvais suivre le fonctionnement des capteurs*

## 5. Gestion des projets

### 5.1. Gestion macroscopique des projets

#### 5.1.1. Gestion du temps et diagrammes de Gantt

Au début de cette période de stage, Kanope fonctionnait sur une base de sprints de deux semaines se concluant par une revue permettant de faire le point sur les activités de l'entreprise et l'ensemble des projets.

Après différentes observations sur les limites de cette organisation comme une fréquence des points non proportionnelle à l'importance des sujets ou le non blocage des plannings et priorités une fois le sprint commencé, celui ci perdant alors de son sens, il a été décidé de retirer le côté gestion des projets dans la réunion bi-mensuelle (anciennement la revue de sprint) et de sortir du principe de "sprints" pour gagner en dynamisme sur l'avancée des sujets.

Chaque projet à donc eu à partir de ce moment, une réunion dédiée de gestion permettant de mieux suivre les différents sujets. Pour les vélos connectés par exemple, comme le projet regroupe 4 personnes dont au moins deux à temps plein et que des jalons contraignants se rapprochent rapidement, il était nécessaire d'avoir un point projet hebdomadaire.

Sur le Gantt page suivante, nous pouvons observer l'utilisation du temps pour ce deuxième semestre, on notera que tels que présentés dans le côté technique des projets, la 5G mMTC occupe la plus grande partie des mois d'avril mai et juin. Puis les différents projets qui viennent occuper les deux derniers mois.

La section Avril Mai est plus ou moins ce qui avait pu être prévu au début du semestre. Pour la suite, l'impact du temps pris par les difficultés rencontrées, les nouveaux projets pour lesquels je ne savais pas nécessairement quand est ce que je serai mobilisé pour y participer, ont rendu l'idée de faire des pronostics sur le planning très hasardeuse. Les éventuels plannings prévisionnels se font au début des différentes phases projets avec une portée de deux à quatre semaines pour ma part (exception faite pour la 5G mMTC).

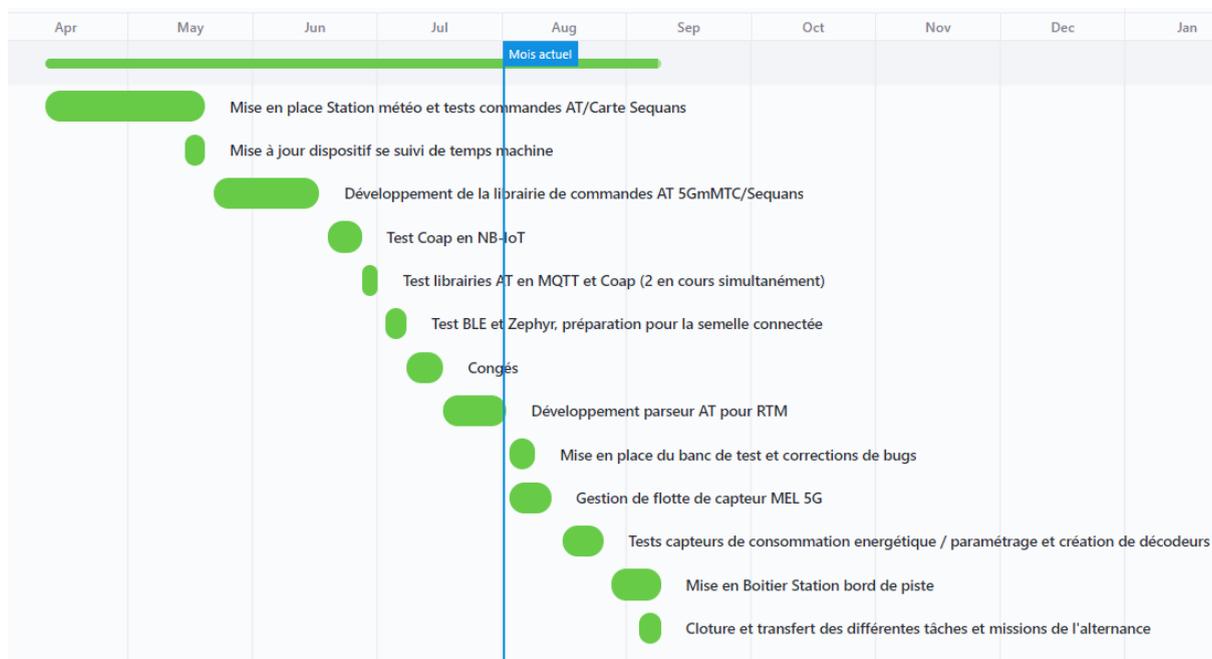
Ce gantt nous permet également de voir le temps pris pour chaque projet, on a donc ainsi :

- Projet 5G mMTC : 12 semaines dont :
  - Station météo, test des commandes AT et mise à jour board Sequans : 6 semaines
  - Parseur AT : 4 semaines
  - Tests Coap et corrections sur les parties précédentes : 2 semaines
  - Mise en boîtier de la station bord de piste et corrections finales des bugs éventuels : 2 semaines prévues

- Projet Suivi de temps machine : 1,5 semaines dont 4 jours de suivi post mise à jour et test hardware
- Projet RTM : 4 semaines
- Sujets à venir : 3 semaines (dont 2 déjà attribuée à la 5GmMTC)

Dans les temps donnés ci-dessus, certains projets se superposent et partagent du temps sur les mêmes périodes. Les retours sur un projet ou tâches de l'ordre de quelques demies journées (1 à 2 jours) ne sont pas représentées pour des raisons de lisibilité.

*Gantt du second semestre de l'alternance, prévisionnel à partir du 20/08*



### 5.1.2. Organisation et place dans l'équipe

La richesse de diversité entre les différents projets que j'ai pu aborder m'a amené à travailler avec toutes les parties de l'équipe Kanope :

- Alexis B, directeur des opérations et mon tuteur, pour m'accompagner sur les différents aspects de gestion de projet.
- Benoît P, qui prend un rôle de "tuteur technique" et qui m'accompagne sur le côté programmation et électronique.
- Romain B, développeur en charge de la gestion de la partie technique du projet 5G mMTC et des éléments software qui vont avec (infrastructure de réception des données)
- Floryan M, qui s'occupe de développer le module de suivi de temps machine sur IoTaaS avec qui je me coordonne pour les remontées de données et résolution de problèmes.
- Thomas G, premier développeur sur IoTaaS, avec qui j'échange pour toute problématique liée aux données de consommation d'énergie ou de besoin d'accès à des outils en ligne (enregistrement de données de test sur serveur...)
- Martin D, qui nous a récemment rejoint et qui s'occupe du firmware pour le concentrateur du projet 5GmMTC, avec qui je travaille conjointement pour l'inter implémentation des briques de codes entre la station bord de piste et le concentrateur (parseur AT)
- Vincent F, qui gère tous les aspects liés à l'utilisation de git et avec qui j'ai travaillé sur les tests pour la semelle connectée.

En principe, les missions que je réalise hors de mon projet principal me sont confiées en fonction du besoin de notre pôle électronique, par Alexis ou Benoît.

Quel que soit le projet sur lequel j'avance, je sais que je peux compter sur l'équipe Kanope pour me conseiller ou m'aider à faire en sorte que les choses aillent dans la bonne direction, de par la cohésion de l'équipe, sa complémentarité technique et la richesse de ses points de vue.

## 5.2. Gestion du projet principal

### 5.2.1. Livrables attendus par le projet

Avec le projet 5G mMTC, viennent plusieurs livrables, qu'ils soient pour les membres du consortium ou en interne à Kanope.

Comme abordé précédemment, j'ai longuement travaillé sur tout le côté interfaçage entre MCU et 5G, ce qui nous a permis d'avoir nos premières données envoyées et de présenter une station bord de piste/ station météo fonctionnelle au COmité de PILotage (COFIL) en juin.

En interne, j'ai pu livrer des documentation sur l'utilisation et la mise à jour des cartes sequans que nous utilisons (non sans difficultés). J'ai également formalisé les résultats pour les tests du COAP.

### 5.2.2. Analyse des risques et des coûts

Déjà au précédent semestre, certains risques avaient été identifiés, comme celui des retards de livraison, auquel nous avons continué de faire face sur cette période :

	Impact			
Probabilité	X			
	X			
Retards de livraison				

Évolution du risque lié aux retards de livraison.

Comme anticipé, il y a eu des problèmes avec les commandes passant par l'INRIA qui en tant que membre du consortium et établissement public doit, pour certaines choses, passer par le marché public.

Nous avons donc eu à acheter certains capteurs en amont pour la station bord de piste essentiellement. Nous avons également pu bénéficier d'un prêt de matériel de la FFC sous la forme de capteurs pour vélos et d'un vélo comparable à ceux pour lesquels le produit est destiné.

Pour finir, ayant toujours des problèmes sur certaines commandes, indépendamment de leurs volontés, nos partenaires à l'INRIA nous ont prêté du matériel devenant critique pour nous permettre de continuer d'avancer sur le projet en attendant certaines livraisons.

	Impact			
Probabilité	Yellow	Red	Red	Red
	Yellow	Yellow	Red	Red
	Green	Yellow	Yellow	Red
	Green	Green	Yellow	Yellow
	Green	Green	Green	Yellow

Mise à jour de la board GM02S

### Évolution du risque lié à la mise à jour de la board Sequans

Nous savions que les boards sequans étaient produites et livrées avec une version LTE-M (4G) du firmware, il était donc nécessaire de faire la mise à jour vers une version supportant le NB-IoT (5G).

Lors de la première mise à jour, un problème a été rencontré suite à un mauvais backup de la carte, celle-ci avait donc perdu son IMEI qui est un identifiant lui permettant de correctement s'interfacer avec le réseau.

Après de nombreux échanges avec Sequans et comprenant que cela pouvait être amené à se reproduire lors de futures upgrades du firmware, la totalité du problème et de sa résolution ont été documentés pour facilement pouvoir faire les bonnes manipulations les prochaines fois.

	Impact			
Probabilité	Yellow	Red	Red	Red
	Yellow	Yellow	Red	Red
	Green	Yellow	Yellow	Red
	Green	Green	Yellow	Yellow
	Green	Green	Green	Yellow

Autres risques

### Évolution des autres risques pour les différents projets

Enfin, pour pallier les autres risques propres à la vie d'un projet, j'échange régulièrement avec l'équipe pour avoir leurs retours et pour résoudre plus facilement tout problème éventuel (point de blocage technique, logistique...) ou pour les anticiper. Le suivi du projet permet alors d'être dans la prévention plutôt que de se retrouver face à une "urgence".

Pour le côté coûts du projet, ceux-ci ont automatiquement augmenté avec l'arrivée de Martin D. qui est complètement dédié au développement du concentrateur mais son expérience dans le domaine permet de réduire les risques de retards ou de manque de maîtrise que j'aurais pu avoir sur certains aspects. Le financement de ce projet permettrait de rémunérer 3 ingénieurs à temps plein pendant 3 ans, ce qui correspond à la moyenne pour cette année de trois ingénieurs, entre le côté électronique, software et gestion de projet.

## 5.3. Éléments de gestion des autres projets

### 5.3.1. Méthode de gestion pour les autres projets

Ma participation aux autres projets étant souvent plus succincte, je n'y ai pas la même implication. Le point commun observé à ces différents sujets est l'importance de la passation d'informations pour permettre la bonne intégration de ma partie dans le projet global.

On veillera donc à bien récupérer les objectifs de la mission, ses contraintes et échéances. Cela pour permettre d'avoir une bonne visualisation sur le rétro planning et sur la priorité de la mission relativement aux autres projets et problématiques.

### 5.3.2. Livrables et documentations réalisés pour les autres projets

Il y a différents livrables qui ont pu être réalisés dans le cadre des différents autres projets :

- Projet RTM, semelles connectées :
  - Plan de test, sous forme d'un tableau en fonction des commandes AT existantes dans la spécification pour la version concernée.
  - Code de démonstration fonctionnel implémentant les fonctionnalités telles que décrites dans le cahier des charges du projet et adaptées au cadre de cette Version 0.1.0.
  
- Suite du projet de suivi de temps machine :
  - Analyse des résultats de tests pour l'utilisation d'un optocoupleur et non dérive des valeurs remontées avec le temps.
  - Fichier de configuration pour les gateways et nouveau programme pour les automates du système.
  - Note de version pour la mise à jour produite et déployée sur les automates et gateways du client.

## 5.4. Problèmes rencontrés et solutions apportées

### 5.4.1. Typologie des difficultés rencontrées

Au long de ce semestre, plusieurs difficultés ont pu être rencontrées, comme la mise à jour de la carte Sequans qui avait nécessité d'échanger avec leur support, ce qui avait été facilité par nos contacts chez eux de par le consortium du projet 5GmMTC.

J'ai également eu des difficultés avec la prise en main de Zephyr, que j'ai abordé, et lors de la station bord de piste, et lors du firmware pour la semelle connectée.

En dehors des aspects techniques, il y a aussi pu y avoir des problèmes liés à un passage d'information pas suffisamment complet et ou à un manque de remontées d'informations de ma part.

### 5.4.2. État d'esprit pour la résolution des points durs.

Qu'ils soient techniques ou non, les problèmes rencontrés dans la vie d'un projet se résolvent toujours par la communication avec le reste de l'équipe. Assurer une bonne remontée des informations permet d'anticiper les impacts sur la gestion du projet. Utiliser la richesse des différentes spécialités et connaissances des membres de l'équipe permet de voir les problèmes sous de nouveaux angles et de parvenir à résoudre ceux-ci.

## 6. Conclusions

### 6.1. Réalisations techniques

#### 6.1.1. Complétion et avancement des projets

Au cours de ces derniers mois, j'ai pu rendre un programme fonctionnel pour communiquer en 5G via COAP ou MQTT, mettre en place une station bord de piste collectant des données météo et de compte tour. Pour les autres projets, comme présentés précédemment, j'ai rendu un code de démonstration fonctionnel pour une semelle connectée ainsi qu'une mise à jour de robustification pour la solution Kanope de suivi de temps machine.

J'ai également pu faire avancer des problématiques sur le suivi de consommation énergétique et le déploiement de capteurs d'ondes 5G.

#### 6.1.2. Usabilité des résultats.

Pour les différents projets sur lesquels j'ai pu travailler, nous pouvons observer des résultats concrets et livrables :

- Du côté du projet des vélos connectés, la station bord de piste est utilisable en l'état pour la démonstration d'octobre auprès du consortium et de la BPI.
- La démonstration du fonctionnement de la semelle connectée en BLE a pu être livrée à notre partenaire.
- Les valeurs des données de suivi de temps machine ont pu être validées comme cohérentes avec les données "vraies" du client.

## 6.2. Progrès personnels et montée en compétences

### 6.2.1. Utilisation des notions universitaires et du monde professionnel

Avec cette année en alternance, j'ai pu appliquer, renforcer et enrichir les compétences et méthodes acquises ou évoquées durant mes études.

Parmi ces différentes notions, on retrouve :

- Le développement firmware embarqué, basé majoritairement sur des NRF52 avec Mbed ou Visual Studio code.
- La programmation par flowcharts d'automates.
- La gestion de projets, gestion du temps et des difficultés rencontrées.

Il y a également de nombreuses choses que j'ai pu découvrir et ou réellement approfondir avec Kanope :

- L'utilisation du RTOS Zephyr pour la gestion des threads et fonctions dans le projet RTM.
- La Conception Assistée par Ordinateur de cartes électroniques et l'assemblage de prototypes (Altium, placement de composant, soudure, utilisation d'un four CMS...).
- La configuration de capteurs ou gateways pour le milieu industriel.
- La gestion des priorités et la mobilisation des ressources sur les projets à fortes contraintes.

J'ai aussi par cette expérience pu renforcer le côté communication et travail d'équipe, essentiel à la vie des différents projets.

### 6.2.2. Perspectives pour la suite

Suite à mon expérience, extrêmement enrichissante chez Kanope, j'ai à ma disposition les outils et compétences nécessaires pour envisager des évolutions dans le secteur du développement de firmwares, pour objets connectés ou systèmes embarqués.

Je vais regarder pour continuer sur ma lancée dans l'IoT Industriel, qui est le domaine correspondant le plus à mes aspirations de ces dernières années, rejoignant ainsi l'automatisme industriel et les systèmes de productions, qui sont les sujets m'ayant le plus porté durant mes études.

## Résumé

Ce document retrace les réalisations faites lors de la phase de stage de l'alternance de M2 que j'ai effectuée chez Kanope (Sparkling Tech) à Lille.

Kanope est un studio IoT développant des solutions électroniques hardware et firmware ainsi que software pour le domaine de l'industrie, des collectivités territoriales, de l'agriculture et plus encore.

Lors de cette alternance et au fil de ce document sont évoqués divers projets, avec le développement d'une solution 5G pour la Fédération Française de Cyclisme, d'un firmware pour un prototype de semelle connectée ou encore la fin de mes travaux sur une solution de suivi de production industrielle. Avec ces projets et tous leurs aspects techniques sont aussi présentés leurs contextes, enjeux et contraintes.

Un poids important est également donné à l'aspect gestion de projets, avec le côté gestion du temps, des échéances et priorités, et de l'autre gestion des risques, du budget et des problèmes rencontrés au cours de ces très instructifs derniers mois.

Cette alternance n'aurait pu être possible sans eux, un grand merci à toute l'équipe Kanope et plus particulièrement Alexis Bernazeau et Benoît Poulain pour leur accompagnement. Merci également à M. Clency Perrine de l'université de Poitiers pour son suivi en tant qu'enseignant référent. Merci par la même occasion à l'ensemble de l'équipe pédagogique.

## Summary

This document traces the achievements made during the internship phase of the M2 work-study program that I did at Kanope (Sparkling Tech) in Lille.

Kanope is an IoT Studio, working on electronic, hardware and firmware as well as software solutions for several domains with among them industry, local authorities or the agricultural sector.

During this internship and through this document, various projects are presented with for example the development of a 5G solution for the French Cycling Federation (FFC), a firmware to test a connected sole or the completion of my long lasting work on an industrial production monitoring solution. Those projects and all their technical aspects, are also presented with their context, challenges and constraints.

An important weight is also given to the project management side with in one hand time management, deadlines and priorities and in the other, risk assessment, expenditures and encountered difficulties during these very instructive last months.

This work-study program wouldn't have been possible without them, a big thanks to the whole Kanope team and more especially Alexis Bernazeau and Benoît Poulain for their continuous support. Thanks also to M. Clency Perrine for his follow-up as referent teacher, and in the process, thanks to the complete pedagogic team of the IoT Master of the University of Poitiers.