Etude expérimentale des performances de communication LTE-M pour l'Internet des objets industriels

Hugo Raps, Nathan Cornelie, Thom Guillot, Eddy Bajic, Kais Mekki

Etude Expérimentale des Performances de Communication LTE-M pour l'Internet des Objets Industriels

Hugo Raps[1], Nathan Cornelie[1], Thom Guillot[1], Eddy Bajic[2], Kais Mekki[3]

[1]TELECOM Nancy, 193 Av. Paul Muller, 54600 Villers-lès-Nancy, France.
[2]Research Centre for Automatic Control of Nancy, Campus Sciences, BP 70239, 54506 Vandoeuvre-lès-Nancy, France.
[3]OKKO SAS, 107 rue Saint Jean, 57510 Remering-lès-Puttelange, France.

{hugo.raps, nathan.cornelie, thom.guillot}@telecomnancy.eu
eddy.bajic@univ-lorraine.fr, kais.mekki@okko-france.com

***Résumé*— Cet article présente une étude expérimentale des performances de communication du réseau LTE-M pour l'Internet des Objets Industriels (IIoT). Nous avons mené une analyse bibliographique approfondie sur les réseaux de communication utilisés par les dispositifs IIoT, en nous basant sur des critères tels que la taille de données transportables, le débit, la connectivité, la latence, le budget énergétique des transmissions et le coût. Ensuite, nous avons expérimenté la communication LTE-M pour un prototype de dispositif IoT contraint en énergie. Grâce au microcontrôleur et au modem, nous avons réussi à établir des requêtes TCP (envoi et réception) ainsi que des requêtes HTTP (POST et GET). Les résultats de notre étude montrent que la communication LTE-M offre des avantages significatifs en termes de débit, de latence et de performance énergétique par rapport aux autres réseaux LPWAN (Low Power Wide Area Networks) existants, ce qui la rend particulièrement adaptée aux besoins des applications IIoT. Ce travail expérimental ouvre de nombreuses perspectives pour la mise en place de solutions IoT efficaces et performantes dans les environnements industriels.**

Mots clés—Internet des Objets Industriels, Low Power Wide Area Network, LTE-M, Etude expérimentale.

I. INTRODUCTION

L'Internet des Objets Industriels est un domaine en constante évolution, avec une croissance rapide du nombre de dispositifs IoT intégrés dans des architectures de communication à plusieurs niveaux pour collecter et traiter des données industrielles en masse. Les réseaux de communication jouent un rôle crucial dans cette évolution, en offrant des solutions pour répondre aux besoins de différents scénarios industriels.

Les réseaux LPWAN (Sigfox, LoRaWAN, LTE-M, NB-IoT) sont des technologies de communication clés qui sont choisies en fonction de critères spécifiques tels que la taille des données transportables, le débit, la connectivité, la latence, le budget énergétique des transmissions et le coût. Actuellement en France, les réseaux Sigfox et LoRaWAN ont montré leur inefficacité en termes de connectivité/couverture en large déploiement notamment en zone rurale. A titre d'exemple, Objenious la filière IoT de Bouygues Telecom a arrêté son réseau LoRaWAN [1]. Tandis que le réseau LTE-M est actuellement en grand déploiement et commercialisation par rapport au réseau NB-IoT.

Dans ce contexte, notre étude expérimentale se concentre sur les performances de communication du réseau LTE-M. Pour cela, nous avons premièrement effectué une analyse bibliographique approfondie sur les réseaux LPWAN pour les communications IoT en nous appuyant sur la littérature scientifique disponible. Puis, nous avons mis en place une expérience pour tester et évaluer les performances de la communication LTE-M pour un terminal IoT prototype contraint en énergie.

Vu notre étude de la littérature existante, nous jugeons que ce travail représente une première contribution qui détaille la réalisation de ce type d'expérimentation. Il détaille l'enchainement des commandes AT à utiliser pour faire fonctionner un modem LTE-M. Cet enchainement n'est détaillé nulle part sur Internet même sur la documentation des fabricants de ces modems. D'où ce travail représente un tutorial et un document référent pour des futures projets et ouvre de perspectives pour la mise en place de solutions IoT efficaces et performantes dans les environnements industriels.

II. ETAT DE L'ART

A. Internet des Objets

L'Internet des Objets (en anglais : Internet of Things IoT) décrit le réseau de terminaux physiques, les "objets", qui embarquent des capteurs, des actionneurs, des softwares et d'autres technologies en vue de se connecter aux serveurs (Cloud) sur Internet et d'échanger des données avec eux. Ces terminaux peuvent aussi bien être de simples appareils domestiques que des outils industriels d'une grande complexité [2]. Avec plus de 7 milliards de terminaux IoT connectés aujourd'hui, les experts s'attendent à ce que ce nombre passe à 10 milliards d'ici 2025 et 22 milliards d'ici 2030.

C'est grâce à différentes technologies que l'IoT a pu voir le jour. Des technologies telles que des progrès en connectivité, avec la prolifération des protocoles réseaux qui a facilité la connexion des capteurs au Cloud et à d'autre "objets" pour un gain d'efficacité des transferts des données, les technologies des capteurs à coût réduit et faible consommation qui quant à eux on permit de rendre ce domaine plus accessible et les avancées effectuées dans le domaine du Machine Learning et des analyses, et avec l'accès à de vastes quantités de données diversifiées stockées dans le Cloud qui permettent aux entreprises d'obtenir les informations plus rapidement.

B. Internet des Objets Industriel

L'Internet des Objets Industriel (en anglais : Industrial IoT - IIoT) désigne l'application de la technologie IoT dans un cadre industriel, en particulier en ce qui concerne l'instrumentation et le contrôle des capteurs et

des terminaux qui font appel à des technologies Cloud. Pour faire transiter toutes ces données entre les objets, il a fallu mettre en places des réseaux adaptés aux objets mais aussi à la quantité de donnée à envoyer et à l'environnement industriel dans lequel les objets évoluent. L'Internet des Objets Industriels utilise généralement une architecture de communication en réseau à deux ou trois niveaux Device-Edge-Cloud pour collecter et traiter des données industrielles massives [3]. Les réseaux LPWAN sont de plus en plus utilisés pour les applications IoT industrielles en raison de leur capacité à fournir une connectivité à faible coût et à faible consommation d'énergie. Les caractéristiques de ces réseaux sont donc très importantes pour le choix du réseau approprié pour une application IoT industriel.

C. Les réseaux de l'IoT

Avec l'avancée rapide des technologies de communication sans fil, le choix du réseau approprié pour les dispositifs IoT peut sembler compliqué. Cependant, pour un architecte de système IoT, il est essentiel de prendre en compte les spécifications et les contraintes liées à l'application pour choisir le réseau le plus adapté. Le choix du réseau pour un dispositif IoT doit être basé sur une évaluation minutieuse de ces spécifications et contraintes. Pour choisir un réseau, on se base sur quatre critères majeurs :

- La portée
- Le débit de transmission de données
- La consommation en énergie
- Le coût de déploiement

La portée est déterminée par la distance maximale à laquelle les données peuvent être transmises entre les dispositifs et les points d'accès, tandis que le débit de transmission de données est la vitesse à laquelle les données peuvent être transmises. La consommation d'énergie est cruciale pour les dispositifs IoT alimentés par batterie, car elle détermine l'autonomie des appareils. Aussi, le coût de déploiement est un facteur important à prendre en compte dans la décision d'utiliser un réseau spécifique.

D. LPWAN

LPWAN (Low Power Wide Area Networks) caractérise comme son nom l'indique l'ensemble des réseaux ayant une longue portée et une basse consommation en énergie. En réduisant le débit des données, les données peuvent être envoyées sur de plus grandes distance tout en conservant cette faible consommation.
Quand on parle de grande distance pour ce type de réseaux on est sur un ordre de grandeurs de quelques kilomètres et on retrouve une autonomie de dispositif qui peut aller jusqu'à plusieurs années et la bande passant est d'environ plusieurs centaines de kbits/sec.
Les diverses implémentations de LPWAN diffèrent en termes de débit et de bandes de fréquences utilisées. Néanmoins, on peut classifier ces implémentations suivant deux catégories : les LPWAN cellulaires et les LPWAN non-cellulaires.

1. LPWAN non cellulaires

SigFox : La startup française SigFox fut la première à rendre la technologie LPWAN populaire. Au début des années 2010, l'entreprise développe le réseau SigFox qui repose sur une technologie brevetée de bande ultra-étroite UNB (Ultra Narrow Band) et utilise des bandes de fréquences sans licence ISM (Industriel, Scientifique et Médical), à savoir 868 MHz en Europe, 915 MHz en Amérique du Nord et 433 MHz en Asie.
Les antennes SigFox ont une portée qui varie entre 3 à 10 kilomètres en zones denses et qui peut aller jusqu'à 50 kilomètres sur des zones avec peu d'obstacles [2]. Toutefois, il est à noter que le réseau SigFox couvre la France dans sa globalité avec 2000 antennes déployées tout au long du territoire. Du point de vue international, le réseau couvre 71 autres pays en Europe et dans le monde, dont 21 qui bénéficient d'une couverture globale.

LoRaWAN : LoRaWAN assure une transmission de données bidirectionnelle à très faible débit et à très faible consommation d'énergie entre les objets et les passerelles (en anglais : Gateway). Elle utilise une modulation à étalement de spectre (en anglais : Chirp Spread Spectrum CSS) appelée LoRa. Comme SigFox, cette modulation s'effectue principalement sur les bandes de fréquences sans licence ISM. L'utilisation de la modulation CSS pour l'internet des objets a été brevetée par Cycléo, une entreprise française ayant été rachetée plus tard par Semtech en 2012 [3]. Cette modulation permet en moyenne une distance entre une passerelle et un objet jusqu'à 5 km en zone urbaine et 15 km en zone rurale.

2. LPWAN cellulaires

NB-IoT : Les antennes NB-IoT ont une portée pouvant aller jusqu'à un kilomètre en zones urbaines et 10 kilomètres en zones rurales. Sur ces portées, le NB-IoT assure une transmission bidirectionnelle avec un débit de données pouvant aller de 20 jusqu'à 250 Kbits/s. Ce bas débit a permis une réduction significative de la consommation énergétique des terminaux IoT, les rendant plus adaptés à des cas d'usage de monitoring distant sur batterie. Ce bas débit a aussi permis une réduction de la complexité et par conséquent une réduction des coûts de déploiement.

LTE-M : La portée des signaux LTE-M peut aller jusqu'à 400 mètres en zones urbaines et à peu près 8 kilomètres en zones rurales. De la même façon que pour le NB-IoT, la réduction du débit de transmission des données a permis de réduire la consommation énergétique des terminaux déployés et a ouvert ainsi le champ d'applications de l'IoT à des cas d'usage qui étaient réservés auparavant uniquement pour les LPWAN non cellulaires.
A la différence du NB-IoT, le LTE-M permet une transmission de données à des débits plus faibles de l'ordre de la dizaine de bits/s tout comme il permet la transmission de données à des débits pouvant avoisiner les 1 Mbits/s. De ce fait, le LTE-M permet de couvrir un champ d'applications plus large que le NB-IoT [4].

III. EXPERIMENTATION DU RESEAU LTE-M

A. Matériel

Pour réaliser l'implémentation et faire les différents tests, nous avons utilisé une carte de développement IoT conçu par l'entreprise OKKO [6] partenaire de notre projet. Cette carte intègre un microcontrôleur ESP32-S2 compact et performant, spécialement conçu pour les applications de l'IoT. Son choix s'est avéré judicieux en raison de sa taille réduite, de ses fonctionnalités avancées et de sa faible consommation d'énergie. Il intègre également un modem HL7800 de la marque Sierra Electronics, un module de communication sans fil qui utilise les réseaux LTE-M et NB-IoT pour offrir une connectivité à bas débit et une faible consommation d'énergie, ce qui en fait un choix idéal pour les applications de l'IoT [5]. Ce modem offre aussi un module intégré GPS pour des applications IoT de géolocalisation. Cette carte offre plusieurs connecteurs I2C pour brancher des capteurs ambiants. Nous avons utilisé une carte SIM IoT du multi-opérateur Thingsmobile [8]. Figures 1, 2 et 3 illustrent ce matériel utilisé.

B. Analyse du problème

L'objectif principal de notre projet était de développer un système capable de transmettre efficacement les données collectées par différents capteurs à un serveur distant. Pour y parvenir, nous avons utilisé le modem LTE-M pour établir une connexion Internet et envoyer les données via des requêtes TCP et HTTP. Le fonctionnement de la carte devait assurer une très faible consommation d'énergie. Nous avons donc implémenté une stratégie d'enchainement d'actions suivie d'une mise en veille prolongée, permettant d'économiser l'énergie lors des périodes d'inactivité. Cette approche a permis de prolonger considérablement l'autonomie du système, tout en garantissant une transmission fiable et efficace des données collectées.

Le microcontrôleur ESP32-S2 utilisé pour ce projet est une version personnalisée qui présente quelques différences avec les modèles standards, notamment en termes de nombre et de limitation des pins programmables. Cette particularité a nécessité une adaptation de nos codes pour tenir compte de cette contrainte.

Figure 1. La carte OKKO: microcontrôleur ESP32-S2 + Modem LTE-M HL7800 + Connecteurs I2C.

Figure 2. Montage pour téléversé le code Arduino à la carte OKKO.

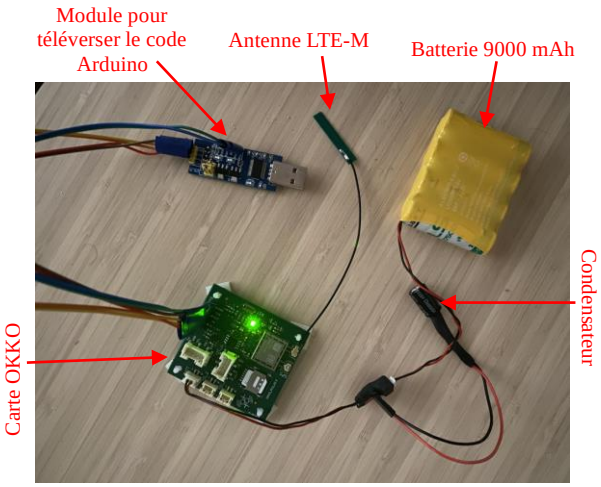

Figure 3. Montage complet pour fonctionnement sur batterie.

En outre, nous avons constaté que la communication entre le modem LTE-M et l'ESP32-S2 n'est pas totalement fiable. En effet, le terminal de retour des commandes peut parfois présenter des inversions ou des suppressions de caractères incontrôlables, qui peuvent mener à l'échec d'une commande AT. Nous avons dû faire face à ce problème lors de l'implémentation de certaines fonctionnalités, comme la mise en place d'un système de surveillance et de reprise de connexion en cas de perte de signal. Malgré ces contraintes, nous avons réussi à surmonter ces difficultés grâce à une analyse minutieuse des retours du terminal et à une adaptation de nos codes pour en tenir compte. Nous avons ainsi travaillé pour atteindre deux objectifs clés : la transmission de données via des requêtes TCP, puis l'envoi de requêtes HTTP vers un serveur Cloud externe.

C. Etude de la documentation du fabricant

Pour comprendre le fonctionnement du microcontrôleur ESP32-S2, nous avons étudié sa documentation et effectué des tests en modifiant les valeurs des broches de la carte électronique. Nous avons ainsi réussi à contrôler le clignotement du LED de la carte en fonction de nos besoins. Cette étape a été cruciale pour la suite du projet, car elle nous a permis de créer une méthode de retour d'informations visuelle fiable pour les futurs tests de communication avec le modem LTE-M. Nous avons également exploré les différentes bibliothèques disponibles pour communiquer avec ce modem de type HL7800 [5], en testant plusieurs d'entre elles pour déterminer la plus adaptée à nos besoins.

Après ceci, nous avons donc consulté pleinement la documentation de ce modem. Le microcontrôleur ESP32-S2 peut communiquer directement avec le modem HL7800 par le biais de commandes AT répertoriées dans celle-ci. On peut y trouver des explications sur les commandes possibles et même des exemples de chaines d'instruction pour effectuer certaines tâches. Notre but étant les requêtes TCP et les requêtes HTTP, nous nous sommes concentrés sur ces parties dans la documentation.
Chacune de ces commandes est utilisable de différente manière, la plupart des fonctions possèdent trois versions :

- Un mode lecture (ajout d'un ? au nom de la commande) qui renvoi l'état actuel de certains paramètres en rapport avec la commande.
- Un mode écriture (ajout d'un = au nom de la commande) qui permet la modification de ces paramètres et l'exécution de la commande.
- Un mode de test qui renvoie une liste des valeurs possibles pour chaque paramètre.

Nous avons remarqué que le modem LTE-M possède deux modes principaux et qu'en fonction de ce qui lui est envoyé, l'un ou l'autre sera activé :

- Le mode principal, dit mode commande, est celui dans lequel le modem doit être lorsqu'on souhaite lui transmettre une commande AT.
- Le mode donné, est celui dans lequel le modem se met lorsque par exemple il attend de recevoir des donnes à transmettre. Cela se remarque lors de la connexion à un serveur TCP où le mode est activé pour envoyer des données au serveur.

Une fois ces tâches réalisées, nous avons conclu qu'il était nécessaire d'avoir des routines, exécutables après l'initialisation du microcontrôleur et du modem.
La carte SIM du modem LTE-M doit être initialisée en premier lieu pour pouvoir établir une connexion avec le réseau cellulaire. Les commandes AT sont utilisées pour cette étape d'initialisation. Ces commandes permettent de configurer et de contrôler le modem via une interface de commande série. Lors de l'initialisation de la carte SIM, on peut également configurer les paramètres du réseau cellulaire tels que l'APN (Access Point Name), le nom d'utilisateur et le mot de passe, nécessaires pour se connecter au réseau cellulaire. Une fois la carte SIM initialisée, le modem peut ensuite être configuré pour établir une connexion TCP ou HTTP avec un serveur distant en utilisant les commandes AT correspondantes.
Afin de tester les capacités de notre système, nous avons mis en place et programmer en Python deux serveurs comme l'illustre la figure 4 : l'un capable de recevoir des requêtes HTTP et l'autre des requêtes TCP. Cette mise en place nous a permis de tester les deux modes de communication et de valider leur bon fonctionnement.
D'après la documentation du modem, l'envoi en HTTP s'effectue en une étape supplémentaire que l'envoi en TCP car il faut placer les champs nécessaires à la requête HTTP. Nous avons donc commencé par la requête TCP en utilisant les commandes AT suivants : AT-KTCPCFG, AT-KTCPSND, ATKTCPRCV et AT-KTCPCLOSE. Pour ce qui est des requêtes HTTP, nous avons utilisé les commandes AT suivants : AT-KHTTPCFG, AT-KHTTPGET, AT-KHTTPPOST, AT-KHTTPHEADER et AT-KHTTPCLOSE.

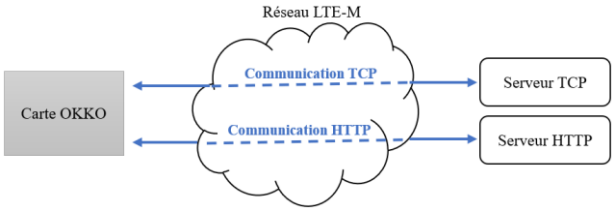

Figure 4. Communication entre la carte et les serveurs via le réseau LTE-M.

D. Implémentation

Pour l'implémentation de notre projet, nous avons utilisé l'éditeur de code Visual Studio Code avec l'extension Arduino PlatformIO pour compiler et envoyer notre code à notre microcontrôleur, ce qui nous a permis de travailler de manière efficace et de gagner du temps dans le processus de développement. Comme tout projet PlatformIO, on définit les spécifications de la carte et du Framework dans un fichier "platform.ini" visible sur la figure 5. On définit la vitesse de transmission des données du port série au moniteur, qui contribue à l'affichage des réponses du modem, à 115200 bauds. Le choix de cette vitesse dépend du microcontrôleur utilisé, dans notre cas un dérivé d'un ESP32-S2.

```
[env:sparkfun_esp32s2_thing_plus]
platform = espressif32
board = sparkfun_esp32s2_thing_plus
monitor_speed = 115200
framework = arduino
lib_deps =
    plerup/EspSoftwareSerial@^8.0.3
```
Figure 5. Extrait du fichier platform.ini.

La librairie EspSoftwareSerial est incluse dans PlatformIO et permet les différentes instructions pour la sortie sur le moniteur. Il était nécessaire de l'importer lorsqu'on travaille sur l'environnement Visual Studio Code.
Pour l'initialisation de la carte SIM, il est nécessaire de procéder à plusieurs commandes AT. Tout d'abord la commande AT+CREG= permet de démarrer la recherche d'opérateur et la connexion à celui qui sera choisi. On effectue ensuite deux vérifications :

- AT+COPS? permet de s'assurer que le modem a choisi le bon opérateur.
- AT+CREG? nous indique l'état actuel de la recherche d'opérateur et s'assurer qu'il s'est bien affecté au bon opérateur.

On lit ensuite la qualité du débit (en dB) avec AT+CSQ pour s'assurer du bon fonctionnement de l'antenne. Cet enchainement de commandes est illustré par la figure 6.

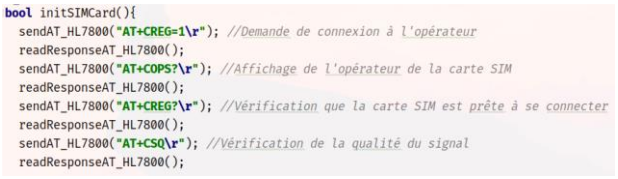

Figure 6. Fonction initSIM - partie 1.

Comme le présente la figure 7, la commande AT+KCNXCFG= indique au modem les paramètres à utiliser pour la connexion avec le réseau de l'opérateur.

Figure 7. Fonction initSIM - partie 2.

1. Envoi d'un message TCP

Comme le présente la figure 8, la commande AT+KTCPCFG= permet de spécifier au modem l'adresse IP et le port du serveur TCP auquel se connecter.

Figure 8. Fonction initSIM - partie 3.

Comme le présente la figure 9, la commande AT+KTCPCNX= permet au modem LTE-M de lancer la connexion TCP avec le serveur.

Figure 9. Extrait de la fonction connectTCP.

La commande AT+KTCPSND= passe le modem en mode données et transmet tout ce qu'il reçoit jusqu'à avoir atteint la taille maximale passée en paramètre ou bien le retour au mode commande. Finalement, on procède à la fermeture de la connexion TCP. Ces procédures sont illustrées par les figures 10 et 11, respectivement.
Cet enchainement de commandes AT permet d'exécuter entièrement la connexion au serveur TCP.

Figure 10. Extrait de la fonction sendMessageTCP.

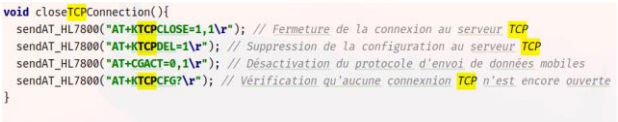

Figure 11. Extrait de la fonction closeTCPConnection.

2. Echange entre la carte et le serveur en HTTP GET et HTTP POST

Comme illustré par la figure 12, la commande AT+KHTTPCFG= permet de spécifier au modem l'adresse IP et le port du serveur HTTP auquel se connecter.

Figure 12. Extrait de la fonction connectHTTP.

Ensuite sur la figure 13, la commande AT+KHTTPGET= permet d'effectuer une requête GET à l'adresse spécifiée du serveur HTTP auquel le modem est connecté. Cela permet de récupérer les données du serveur.

Figure 13. Extrait de la fonction sendGETHTTP.

La commande AT+KHTTPHEADER= permet de spécifier les entêtes de la requête HTTP POST à envoyer. Nous spécifions par exemple le type de données que nous envoyons et sa taille (dans notre cas le type de données est un JSON). Ensuite, nous utilisons AT+KHTTPPOST= pour envoyer le message HTTP POST à l'adresse du serveur spécifiée. Par conséquent, le modem passe en mode données. Nous lui donnons donc les données à transmettre au serveur. Le "+++" permet ensuite au modem de retourner au mode commande comme illustré par la figure 14. Finalement, on procède à la fermeture de la connexion HTTP (voir figure 15).
Cet enchainement de commandes AT permet d'exécuter entièrement la connexion HTTP GET et POST au serveur.

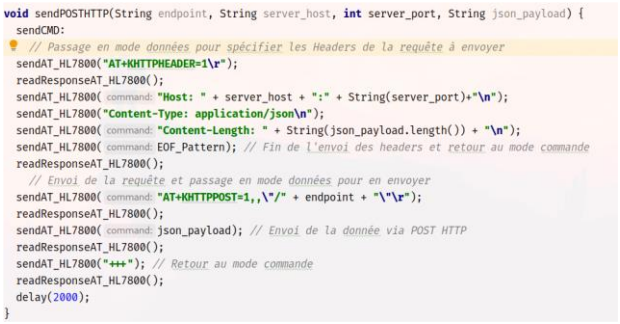

Figure 14. Extrait de la fonction sendPOSTHTTP.

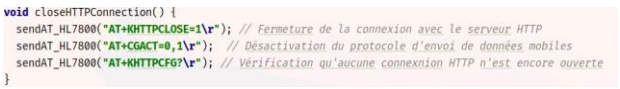

Figure 15. Extrait de la fonction closeHTTPConnection.

E. Etude de la consommation énergétique

Une fois le programme téléversé à la carte OKKO, le microcontrôleur commence l'initialisation du modem. Ensuite, ce dernier commence l'exécution de la communication TCP et HTTP décrites dans la partie

précédente. Une fois la transmission est finie, la carte se met en mode d'endormissement prolongé (en anglais : Deep Sleep) jusqu'à la prochaine demande de transmission. Comme discuter précédemment dans cet article, ce mode permet d'économiser l'énergie de la batterie durant les périodes d'inactivité de la carte.

La consommation énergétique de ce comportement a été mesuré par l'outil OTII [7].

Le tableau 1 présente la consommation des différentes phases d'exécution de la carte. Comme l'illustre ce tableau, la consommation de la carte OKKO en mode TCP est moins énergivore que le mode HTTP car ce dernier a plus de charge protocolaire (entête et établissement de connexion) pour établir la connexion entre le modem et le serveur. D'où, nous recommandant l'utilisation du protocole TCP pour envoyer les données de la carte vers le serveur via le réseau LTE-M.

En outre, le tableau illustre la consommation de la carte en mode endormissement. Cette consommation est égale à 32 micro-ampères ce qui représente une assurance garantit d'une longue autonomie de la batterie. Les calculs et les tests faites à l'entreprise OKKO, annonce une durée d'autonomie de 5 ans pour une carte qui envoie un seul message TCP chaque 24 heures (avec une batterie de capacité 9000 mAh).

Tableau 1. Consommation énergétique de la carte OKKO.

	Initialisation du modem		Communication		Endormissement
	Durée moyenne	Consommation moyenne	Durée moyenne	Consommation moyenne	Consommation moyenne
TCP	15 secondes	88 mA	22 secondes	96.2 mA	32 uA
HTTP	27 secondes	91 mA	106 secondes	128 mA	

IV. Conclusion

En conclusion, ce projet a permis de mener une étude des performances du réseau LTE-M pour l'Internet des Objets Industriels, en se basant sur une analyse bibliographique approfondie et une mise en œuvre pratique sur un terminal IoT prototype contraint en énergie de l'entreprise OKKO.

Grâce au microcontrôleur ESP32-S2 et au modem LTE-M, nous avons réussi à établir des communications TCP ainsi que des communications HTTP (POST et GET), ce qui contribue à la mise en place de solutions IoT efficaces et performantes dans les environnements industriels. Les résultats obtenus ont montré que le réseau LTE-M est parfaitement adapté aux applications nécessitant une transmission à faible débit sur une large zone de déploiement. Ces résultats ouvrent la voie à de nombreuses perspectives pour la mise en place de systèmes IoT autonomes dans l'industrie.

References

[1] Arrêt du réseau LoRaWAN de Bouygues Telecom: https://objenious.com/blog/technologie/arret-du-reseau-lorawan-de-bouygues-telecom/

[2] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment", ICT Express Journal, Vol. 5, No. 1, March 2019, pp. 1-7.

[3] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT", IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom'2018), 19-23 March 2018, Athens, Greece.

[4] A. Koohang, C.S. Sargent, J.H. Nord, J. Paliszkiewicz, "Internet of Things (IoT): From awareness to continued use", International Journal of Information Management, Vol. 62, February 2022.

[5] AirPrime HL78xx, AT Commands Interface Guide: https://source.sierrawireless.com/resources/airprime/software/hl78xx_at_commands_interface_guide.

[6] OKKO SAS, 107 rue Saint Jean, 57510 Remering-lès-Puttelange, France.

[7] Otii by QOITECH: www.qoitech.com

[8] Thingsmobile: www.thingsmobile.com