

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉBADJI MOKHTAR-ANNABA  
BADJI MOKHTAR- ANNABAUNIVERSITY



جامعة باجي مختار - عنابة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Electronique

Spécialité : Electronique des systèmes Embarqués

## Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**Systeme de contrôle et d'acquisition de données à travers les réseaux mobiles GSM/4G**

Présenté par : LAKNAOUI MOHAMED RACIM

Encadrant : ZERMI NARIMA

Grade MCA

UniversitéUBM-ANNABA

### Jury de Soutenance :

BOULEBTATECHE brahim	MCB	UBM-ANNABA	Président
ZERMI narima	MCA	UBM-ANNABA	Encadrant
BOUTERAA Nadia	MCA	ANNABA-UBM	Examineur

Année Universitaire : 2021/2022

الحمد لله رب العالمين والصلاة والسلام على خاتم  
الأنبياء والمرسلين

## Remerciement et dédicaces

Je remercie particulièrement mon encadreur madame **Zermi Narima** professeur au département d'électronique à l'université Badji Mokhtar de Annaba pour ses précieux conseils, ses qualités humaines et surtout pour la confiance qu'elle m'a témoignée.

Je dédie ce travail à mes chers parents et à mes grands-parents pour toute l'aide et le soutien qu'ils m'ont apportés.

Je tiens également à remercier tous mes professeurs de l'université Badji-Mokhtar Annaba qui ont contribué à ma formation depuis le tout début.

---

## Liste des abréviations

<b>IoT</b>	Internet of things (Internet des objets)
<b>4G</b>	Forth generation of mobile technology
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communication
<b>SMS</b>	Short Message Service
<b>NB-IoT</b>	Narrow band Internet of Things
<b>LTE-M</b>	Long Term Evolution for Machines
<b>IP</b>	Internet Protocol
<b>TCP</b>	Transmission Control Protocol
<b>HTTP</b>	HyperText Transfer Protocol
<b>HTTPS</b>	HyperText Transfer Protocol
<b>API</b>	Application programming Interface
<b>FTPS</b>	File Transfert Protocol Secure
<b>SSL</b>	Secure Socket Layer
<b>MQTT</b>	Message Queuing TelemetryTransport
<b>LTE</b>	Long Term evolution
<b>UART</b>	Universal Asynchronous Receiver/Transmitter
<b>IDE</b>	integrated development environment
<b>LPWAN</b>	Low Power Wide Area Network
<b>SoC</b>	System on Chip

## Liste des figures

### Chapitre I

Figure I.1 : L'internet des objets	15
Figure I.2 : Application IoT	17

### Chapitre II

Figure II.1 : Console de communication avec le modem sur terminal putty	25
Figure II.2 : Les principaux types de réseaux cellulaires support pour la connectivité IoT	33
Figure II.3 : Normes 3GPP par réseau. Source : 3GPP	35
Figure II.4 : Création d'un canal ThingSpeak	45
Figure II.5 : Génération de la clé API ThingSpeak	45
Figure II.6: Script de mise à jour d'un champ ThingSpeak	46
Figure II.7 : Description des canaux ThingSpeak	46
Figure II.8 : Configuration d'un canal ThingSpeak	47
Figure II.9 : Configuration des champs ThingSpeak	48
Figure II.10 : Visualisation de données sur ThingSpeak	48

### Chapitre III

Figure III.1: Puce ESP32	50
Figure III.2 ESP32 WiFi/Bluetooth Low Energy Module	50
Figure III.3 : Carte de développement ESP32	51
Figure III.4: BLOCK DIAGRAM system on chip ESP32	53
Figure III.5: Quelques variantes de carte de développement ESP32	53
Figure III.6: Kit de développement BK-SIMCOM7600G-H	55
Figure III.7: Block Diagramme de chip SIMCOM7600G	57
Figure III.8: Assignement des broches puce SIM7600 et SIM7600G-H	58
Figure III.9: Brochage du module	59
Figure III.10: DHT22 breakout board	59
Figure III.11: câblage internet du module	60
Figure III.12: Module GY-NEO6MV2	60
Figure III.13: Principe d'une plateforme IoT sur système embarqué	62
Figure III.14: Schéma électrique du système	63

## SOMMAIRE

Remerciements et dédicaces	2
Liste des abréviations	3
Liste des figures	4
Sommaire	5
Résumé	8
<b>Introduction Générale</b>	<b>11</b>
<b>CHAPITRE I :</b>	<b>15</b>
I.1 Concepts fondamentaux de l'Internet des objets	15
I.1.1 qu'est-ce que L'IoT(Internet des Objets) ?	15
I.1.2 brève histoire de l'IoT	15
I.1.3 comment fonctionne l'IoT ?	16
1-capteurs	
2-connectivité	
3-traitement des données	
4-interface utilisateur	
5-recevoir des notifications automatiques	
6-surveillance proactive des informations	
7-contrôler le système à distance	
I.1.4 Les applications de l'IoT	17
1-« wearables »	17
2-smarthome	18
3-systèmes de santé	18
4-villes intelligentes (smartcities)	18
5-agriculture	18
6-automatisation industrielle	18
I.1.5 Internet des objets :défis et solutions	19
I.1.6 Solutions de sécurité IoT	20
1-sécuriser le réseau IoT	20
2-authentifier les appareils IoT	20
3-utiliser le cryptage des données	20
4-utiliser les méthodes de sécurité IoT PKI	20
5-utiliser l'analyse de la sécurité IoT	21
6-tester le matériel IoT	21
I.1.7 Les contraintes à lever pour l'IoT	21
1-flexibilité	21
2-complexité	21
3-conformité	21
I.1.8 Les bonnes pratiques IoT	21
I.2 Les systèmes embarqués pour l'Internet des Objets	21
I.2.1 Qu'est ce qu'un système embarqué ?	22
I.2.2 Qu'est ce qu'un système embarqué dans l'IoT ?	22
I.2.3 L'importance des systèmes embarqués dans L'IoT	22
I.2.4 Comment les entreprises adoptent l'IoT	23
I.3 Logiciels embarqués IoT : défis	23
I.3.1 Sécurité	23
I.3.2 mémoire limitée	23
I.3.3 mise à jour	24

**CHAPITRE II**

II.1 commandes AT pour modem cellulaire	24
II.1.1 Une introduction aux commandes AT	24
II.1.2 Comment envoyer des commandes AT	25
II.1.3 Comment formater les commandes AT	25
II.1.4 Champs de la ligne de commandes AT	26
II.1.5 Réponse à la commande AT	27
II.1.6 Types de commandes AT et tests	27
II.1.7 Comment répondre aux requêtes des modems	28
II.1.8 Quelques commandes AT clés	28
II.2 L'IoT cellulaire	32
II.2.1 Définition	32
II.2.2 Facteurs de croissance pour l'IoT cellulaire	35
1-couverture accrue	35
2-facilité et faible coût de déploiement	35
3-gestion à distance	36
4-maturité des protocoles de communication cellulaire pour les Applications IoT(exemple :NB-IoT)	36
5-commutation réseau et itinérance	36
6-faible latence et haute fiabilité	36
7-sécurité dès la conception	36
8-faible consommation d'énergie	36
II.2.3 IoT cellulaire : cas d'usages les plus courants	37
1-surveillance des objets à distance en lecture seule	37
2-automatisation des processus basés sur l'IoT	37
3-surveillance et contrôle des objets à distance (lecture/écriture)	38
4-gestion du parc véhicules (track&trace)	38
5-localisation (GPS)	38
6-IoT pour l'optimisation des performances des actifs/usines	38
7-contrôle et gestion de la qualité basés sur l'IoT	38
8-surveillance état des marchandises en transit basée sur l'IoT	39
9-maintenance prédictive	39
10-suivi et traçabilité sur site	39
II.3 Technologies réseaux pour les objets connectés	39
II.3.1 Qu'est ce qu'un réseau IoT ?	39
II.3.2 Réseaux à courte ou longue portée ?	40
II.3.3 exemples de réseaux longue portée	40
II.3.4 exemples de réseaux courte portée	42
II.4 Plates-formes IoT :cas du service THINGSPEAK	43
II.4.1 Présentation de la plate-forme THINGSPEAK	43
II.4.2 Fonctionnalités de THINGSPEAK	43
II.4.3 Principes de base de THINGSPEAK	43

**CHAPITRE III**

III.1 Conception et réalisation d'un système embarqué pour une solution IoT Mobile	49
III.1.1 Etat de l'art	49
III.1.2 présentation du projet	49
III.1.3 SoC (System on Chip) ESP32	49

---

III.1.3.1	présentation	49
III.1.3.2	caractéristiques du SoC ESP32	51
III.1.4	choix de la plate-forme ESP32	54
III.1.4.1	caractéristiques de la carte	54
III.1.4.2	avantages	54
III.1.4.3	outils de développement (SDK) pour les ESP32	54
III.2	Module communication et positionnement BK-SIM7600G-H58	55
III.2.1	Caractéristiques de la puce SIMCOM7600G-H	55
III.2.1.1	caractéristiques générales	55
III.2.1.2	DATA	56
III.2.1.3	autres caractéristiques	56
III.2.1.4	interfaces	56
III.2.1.5	bandes de fréquences supportées	57
III.2.1.6	le module BK-SimCom7600G-H	58
III.2.1.7	brochage du module BK-SimCom7600G-H	58
III.3	Capteur de température DHT22 (am2302)	59
III.3.1	Présentation du module	59
III.3.2	Principales caractéristiques	59
III.4	Module GPS NEO-6M	60
III.4.1	Présentation du module	60
III.4.2	Caractéristiques principales	61
III.5	Réalisation du projet	62
III.5.1	Partie hardware	62
III.5.2	Schéma électrique du système	62
III.5.3	Connexion des modules à l'ESP32	63
III.5.4	software de commande du système	64
III.5.5	Librairies utilisées pour la gestion des modules	64
III.5.6	Software : code ESP32 du système	65
<b>Conclusion générales</b>		<b>67</b>
<b>Références bibliographiques</b>		<b>68</b>

## Résumé

Les Systèmes embarqués ont évolué depuis leur avènement, en vase clos (voitures, appareils électroménager...etc.) et leur ouverture sur le monde extérieur (INTERNET, INTRANET) a dû attendre l'apparition des réseaux et protocoles de télécommunication de troisième et quatrième génération pour connaître la forte et soudaine expansion que nous vivons actuellement.

Le présent travail est une application pratique des concepts novateurs de systèmes embarqués ouverts sur l'univers de l'Internet et des objets connectés, il consiste en **la conception, la réalisation, et la programmation d'une plateforme pour la géolocalisation et la surveillance de processus en situation de mobilité.**

L'originalité de ce modeste travail tient en quelques concepts clés suivants :

- Totale mobilité du système grâce à l'utilisation d'un modem –routeur (SIM7600G-H CAT-M) LTE optimisé pour le monde de l'IoT (**supportant en outre les protocoles IoT LTE-M et NB-IoT**) et dont le débit théorique peut atteindre 150Mbs en download et 50Mbs en upload, et le CellHandOver approchant bien supérieur à 100KM/H.
- Sécurité accrue des communications grâce aux protocoles sécurisés HTTPS, FTPS, SSL ...etc.
- Autonomie en énergie de longue durée grâce à la faible consommation électrique en plus du mode « Power Saving Mode » largement implémenté dans les dispositifs utilisés.
- Faible encombrement grâce à des modules de tailles optimisés pour l'univers de L'IoT.
- Plage de température de service (-40°C~+85°C) largement suffisante pour les besoins de pareils projets.

## ملخص

تطورت الأنظمة المدمجة منذ ظهورها في الفراغ (السيارات، والأجهزة المنزلية، وما إلى ذلك) واضطر انفتاحها على العالم الخارجي (الإنترنت) إلى انتظار ظهور شبكات وبروتوكولات الاتصالات من الجيل الثالث والرابع لمعرفة القوة. والتوسع المفاجئ الذي نشهده حاليًا.

هذا العمل هو تطبيق عملي للمفاهيم المبتكرة للأنظمة المدمجة المفتوحة لعالم الإنترنت والأشياء المتصلة، ويتكون من تصميم وإنتاج وبرمجة منصة لتحديد الموقع الجغرافي ومراقبة عملية المحمول. أصالة هذا العمل المتواضع تكمن في بعض المفاهيم الأساسية:

- إمكانية تنقل كاملة للنظام بفضل استخدام جهاز توجيه مودم (LTE SIM7600G-H CAT-M) مُحسّن لعالم إنترنت الأشياء (يدعم أيضًا بروتوكولات IoT LTE-M و NB-IoT) والتي يمكن أن تصل سرعتها النظرية إلى 150 ميجا بت في الثانية تنزيل و 50 ميجا بايت في التحميل ، وتقترب خلية HandOver من 100 كيلومتر / الساعة.
- زيادة أمان الاتصالات بفضل البروتوكولات الآمنة HTTPS و FTSP و SSL وما إلى ذلك.
- استقلالية طاقة طويلة الأمد بفضل انخفاض استهلاك الطاقة بالإضافة إلى "power saving mode" المطبق على نطاق واسع في الأجهزة المستخدمة.
- مساحة صغيرة بفضل وحدات الحجم المحسّنة لعالم إنترنت الأشياء.
- نطاق درجة حرارة الخدمة °C (40 ~ -85 +) كافٍ إلى حد كبير لاحتياجات مثل هذه المشاريع.

## ABSTRACT

Embedded systems have evolved since their advent in a vacuum (cars, household appliances, etc.) and their opening to the outside world (INTERNET, INTRANET) had to wait for the appearance of third and fourth generation telecommunication networks and protocols to know the strong and sudden expansion that we are currently experiencing.

This work is a practical application of the innovative concepts of embedded systems open to the world of the Internet and connected objects, it consists of the design, production, and programming of a platform for the geolocation and monitoring of mobile process.

The originality of this modest work lies in a few keys' concepts:

- Total system mobility thanks to the use of an LTE modem-router (SIM7600G-H CAT-M) optimized for the IoT world (also supporting the IoT LTE-M and NB-IoT protocols) and whose the theoretical speed can reach 150Mbps in download and 50Mbps in upload, and the Cell Handover approaching well above 100KM/H.
- Increased security of communications thanks to secure protocols HTTPS, FTPS, SSL, etc.
- Long-lasting energy autonomy thanks to low power consumption in addition to the "Power Saving Mode" widely implemented in the devices used.
- small footprint thanks to size modules optimized for the IoT universe.
- Service temperature range (-40°C~+85°C) largely sufficient for the needs of such projects.

## Introduction Générale

Le nombre croissant d'appareils et d'applications IoT exige, pour une bonne prise en charge, des normes de communication assez strictes combinant un faible coût de déploiement et d'exploitation, une faible consommation d'énergie, une longue portée et de faibles débits de données. En outre les stations de base doivent présenter une forte montée en charge (scalabilité) et pouvoir ainsi supporter plusieurs milliers d'appareils. Comme des technologies telles que Bluetooth, ZigBee, WiFi et cellulaire conventionnelles ne répondent pas toujours à ces exigences, des solutions non **cellulaires** de réseau étendu à faible puissance (LPWAN : Low Power WAN) comme **LoRa** et **SigFox** ainsi que les **normes cellulaires** comme **NB-IoT** et **LTE-M** ont vues le jour. Voyons comment ces technologies IoT s'intègrent dans le paysage et comment évaluer leurs impacts sur le marché des objets connectés.

- Le **LTE-M** développé par le « 3GPP » (organisme de normalisation en télécommunications) en tant que solutions LPWAN qui s'appuient sur la connectivité LTE standard tout en préservant les ressources existantes.
- **NB-IoT** est une autre initiative « 3GPP » pour répondre aux problèmes d'immunité face aux interférences dont souffrent les réseaux Sigfox et LoRa Alliance. Il est différent de LTE-M en fonctionnant en dehors de l'architecture LTE.

NB-IoT présente une faible consommation d'énergie et un coût réduit, grâce à des chipsets spécialement conçus pour ce protocole NB-IoT. Il est en outre optimisé pour les Smart Cities grâce à une meilleure pénétration des bâtiments que LTE-M.

NB-IoT est mieux adapté aux actifs statiques comme les compteurs intelligents, tandis que LTE-M présente ses avantages dans les applications itinérantes comme les véhicules ou les drones.

Le LTE-M présente également des avantages notables. Tout d'abord, il offre des débits de données plus élevés, ce qui est important pour les cas d'utilisation riches en données. Contrairement à NB-IoT, son front-end est relativement simple.

L'efficacité énergétique du LTE-M, et les problèmes liés aux licences peuvent limiter la diffusion à grande échelle du LTE-M.

### **LoRa**

L'Alliance LoRa est une association ouverte, à but non lucratif, créée pour favoriser un écosystème pour certaines technologies LPWAN.

LoRaWAN est communément définie comme une couche réseau standard ouverte régie par l'Alliance LoRa. Cependant, elle n'est pas vraiment ouverte puisque la puce sous-jacente permettant de mettre en œuvre une pile LoRaWAN complète n'est disponible qu'en licence. Fondamentalement, LoRa est la couche physique : la puce. LoRaWAN est la couche MAC : le logiciel qui est placé sur la puce pour permettre la mise en réseau.

Sa fonctionnalité est similaire à celle de SigFox. Elle est principalement destinée aux applications de liaison montante (données provenant de capteurs/d'appareils vers une passerelle) avec de nombreux points d'extrémité. Toutefois, au lieu d'utiliser la transmission à bande étroite, elle distribue les informations sur différents canaux de fréquence et débits de données à l'aide de paquets codés. Ces messages sont moins susceptibles d'entrer en collision et d'interférer les uns avec les autres, ce qui augmente la capacité de la passerelle.

### **SigFox**

SigFox utilise une technologie propriétaire, par exemple en utilisant un taux de modulation lent pour obtenir une portée plus étendue. En raison de ce choix de conception, SigFox est une excellente option pour les applications où le système ne doit envoyer que de petites rafales de données peu fréquentes.

Parmi les applications possibles, citons les capteurs de stationnement, les compteurs d'eau ou les poubelles intelligentes. Cependant, SigFox présente aussi quelques inconvénients. La transmission de données aux capteurs/appareils (capacité de liaison descendante) est très limitée et les interférences de signal peuvent poser problème.

Ce tour d'horizon des technologies et protocoles IoT **[10]** nous conforte dans l'idée que beaucoup reste à faire dans ce domaine et des opportunités verront le jour très prochainement pour exploiter au mieux ces nouvelles technologies, sans oublier que

---

l'avènement de la 5G sera un facteur disruptif pour tout l'écosystème technologique et l'impact sur les sociétés humaines est certains.

**Ce projet constitue une timide intrusion dans le domaine de L'IoT mobile, et nous aura permit d'entrevoir le potentiel formidable des technologies IoT** , il est articulé autour de trois chapitre :

**Chapitre I** : on introduit les concepts fondamentaux de l'Internet des objets, ainsi que le rôle et l'importance des systèmes embarqués dans l'univers de l'Internet des objets.

**Chapitre II** : on présente les technologies réseaux et les protocoles Internet utilisés dans le cadre de ce projet.

**Chapitre III** : cette partie est l'application pratique des concepts évoqués dans les chapitres précédents, et représente l'implémentation d'une solution au problème de la mobilité dans le cadre des systèmes embarqués appliqués à l'Internet des objets.

Enfin, nous terminerons par une conclusion générale et les perspectives qui s'ouvrent relativement aux avancées à moyen et court terme des technologies sous-jacentes.

## CHAPITRE I

### I-1-Concepts fondamentaux de l'Internet des objets [11]

#### I-1-1 Qu'est-ce que l'IoT (Internet des objets) ?

L'Internet des objets (IoT) est un réseau d'objets physiques qui sont intégrés à des logiciels, des composants électroniques, un réseau de communication et des capteurs qui permettent à ces objets de collecter et d'échanger des données. L'objectif de l'IoT est d'étendre la connectivité Internet à des objets de la vie courante (une montre, un lave-linge, mobilier urbain, compteur de gaz...etc.).



Figure I.1L'Internet des objets

#### I-1-2 brève histoire de l'IoT

- Le concept de l'Internet des objets naît en 1999, au MIT (*Massachusetts Institute of Technology*). Le MIT Auto-ID laboratoire est dédié à la création d'objets connectés à l'aide de l'identification par radiofréquence et les réseaux de capteurs sans fil.

- 1999- Le terme "Internet des objets" a été utilisé par Kevin Ashton lors de son travail chez P&G qui est devenu largement accepté.
- 2005-L'Union internationale des télécommunications (UIT) publie son premier rapport sur ce sujet.
- 2008- L'Internet des Objets est né.

### **I-1-3 Comment fonctionne l'IoT ?**

L'ensemble du processus IoT est basé sur quatre composants fondamentaux qui forment la plateforme IoT.

**1) Capteurs :** Les capteurs sont un élément clé qui aide à collecter des données en temps réel du milieu environnant. Toutes ces données brutes peuvent avoir différents niveaux de complexité. Il peut s'agir d'un simple capteur de température ou d'un flux vidéo crypté. Une fois les données collectées, l'objet doit être capable d'envoyer les données vers des infrastructures dédiées qui pourront stocker ces données

**2) Connectivité :** Toutes les données collectées sont envoyées vers une infrastructure Cloud dédiée. Les capteurs sont connectés au Cloud à l'aide de divers moyens de communication. Ces supports de communication incluent les réseaux mobiles ou satellitaires, Bluetooth, WI-FI, WAN, etc.

**3) Traitement des données :** une fois ces données brutes sont collectées et parviennent au Cloud elles sont structurées pour pouvoir être analysées via des algorithmes afin de restituer à l'utilisateur une information à valeur ajoutée. Ce processus peut simplement consister à vérifier la température sur des objets. Cependant, cela peut parfois aussi être plus complexe et nécessiter un traitement sophistiqué comme la reconnaissance faciale, l'identification d'objets...etc.

**4) Interface utilisateur :** les données structurées et mises sous forme adéquates sont les informations qui seront disponibles pour l'utilisateur final afin de permettre leur visualisation au travers de tableaux de bord dédiés. Il existe de nombreuses manières par lesquelles un utilisateur peut interagir avec une solution IoT :

### 5) Recevoir des notifications automatiques

Le cas le plus courant dans les applications IoT est que nous souhaitons recevoir des notifications ou des alertes pour les événements inhabituels. Par exemple, si la température d'un processus franchie un certain seuil.

### 6) Surveillance proactive des informations

Nous voudrions peut-être être en mesure de surveiller les informations de manière proactive. Par exemple, surveiller l'emplacement de véhicules même si rien d'inhabituel ne se produit. Nous pourrions utiliser une application mobile ou informatique pour surveiller les informations.

### 7) Contrôler le système à distance

L'interface utilisateur peut également permettre à l'utilisateur de contrôler le système IoT à distance. Par exemple, l'utilisateur peut régler les lumières ou éteindre le chauffage via une application mobile. Cela pourrait également être fait automatiquement par l'application elle-même selon les directives qui lui sont données.

## I-1-4 Les Applications de l'IoT[24]

On prévoit que le nombre d'équipements connectés atteindra 55 milliards d'ici à 2025 et tous les domaines d'activités (entreprises ou grand public) seront touchés par cet engouement pour les objets connectés. [1]

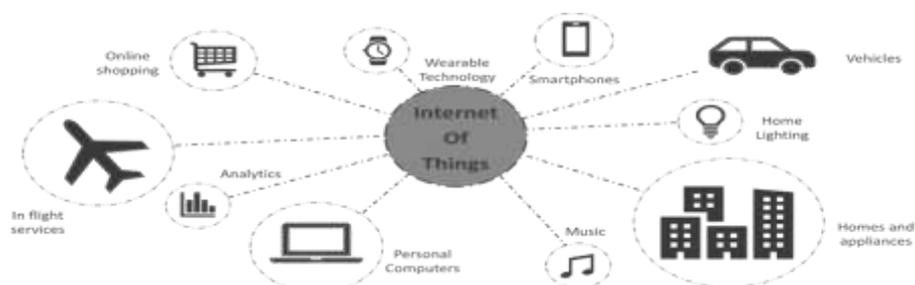


Figure I.2 : Applications IoT

l'IoT est déjà largement déployé, dans divers domaines, à savoir :

**1)Wearables :** (vêtement ou accessoire intégrant de l'informatique et de l'électronique) La technologie portable est une caractéristique des applications IoT et est probablement l'une des premières industries à avoir déployé l'IoT à son service. Fit Bits, moniteurs de fréquence cardiaque, montres intelligentes ...

**2)Smart Home :** l'IoT concerne tous les appareils électroménagers qui communiquent (compteurs intelligents, système de sécurité connectés, détecteurs de fumée, de présence etc.).

**3)Système de santé :** Les appareils de santé représentent l'un des secteurs à la croissance la plus rapide du marché de l'IoT. En fait, la valeur de ce secteur, parfois appelé Internet of MedicalThings (IoMT), devrait atteindre 176 milliards de dollars d'ici 2026.

Les appareils IoT offrent un certain nombre de nouvelles opportunités aux professionnels de la santé pour surveiller les patients, ainsi qu'aux patients pour se surveiller eux-mêmes.

- . Surveillance à distance des patients
- . Surveillance de la glycémie
- . Surveillance de la fréquence cardiaque
- . Surveillance de l'hygiène des mains
- . Surveillance de la dépression et de l'humeur
- . Surveillance de la maladie de Parkinson

**4)Villes intelligentes :** Les gouvernements et les ingénieurs peuvent utiliser l'IoT pour analyser les facteurs souvent complexes de l'urbanisme propres à chaque ville. L'utilisation d'applications IoT peut aider dans des domaines tels que la gestion de l'eau, le contrôle des déchets et les urgences, la gestion du trafic routier, la surveillance détérioration de la qualité de l'air...

**5)Agriculture :** Les statistiques donnent un chiffre de près de 10 milliards d'humains sur terre à l'horizon 2025. Pour nourrir une population aussi massive, il faut optimiser les récoltes grâce à la technologie pour obtenir de meilleurs rendements. Il existe de nombreuses possibilités dans ce domaine. L'un d'eux est la serre intelligente.

Une technique de culture sous serre améliore le rendement des cultures en *contrôlant les paramètres environnementaux*. Cependant, la manipulation manuelle entraîne une perte de production, une perte d'énergie et des coûts de main-d'œuvre, ce qui rend le processus moins efficace. Une serre avec des dispositifs intégrés facilite la surveillance et le contrôle de tout le processus. Les capteurs mesurent différents paramètres en fonction de besoins prédéfinis les envoient dans le Cloud, où les données sont traitées et des actions de contrôle sont appliquées.

**6)L'automatisation industrielle :** À l'aide de données en temps réel capturées par des capteurs et d'autres matériels embarqués, l'IoT utilise l'analyse approfondie, l'IA et l'apprentissage automatique pour permettre aux entreprises d'économiser du temps et de l'argent en identifiant les tendances et les domaines à améliorer.

Les systèmes IoT d'aujourd'hui sont également capables d'intervention proactive. Cela permet des ajustements minutieux du fonctionnement de la machine elle-même ainsi que des tâches de planification et d'ordonnancement, telles que la modification des horaires de production pour minimiser les temps de rotation.

Les appareils IoT permettent également aux usines d'adopter des stratégies de maintenance prédictive pour identifier à l'avance les défaillances des pièces, planifier la réparation au moment optimal, notifier l'approvisionnement en pièces de rechange et informer les techniciens de la réparation à venir. Tout cela se produit sans prise de décision humaine. Grâce à la technologie IoT, les entreprises ont une visibilité presque complète sur les processus de production. L'IoT peut permettre une vue complète de l'approvisionnement des matières premières et des composants (et donc de la traçabilité), jusqu'à la production, et l'expédition des produits finis.

## **I-1-5 Internet des objets (IoT) : défis et solutions**

Comme toute technologie émergente, l'Internet des objets (IoT) présente de nombreux avantages et risques. D'une part, les « pro » IoT et les fabricants d'appareils IoT font la promotion de l'Internet des objets comme un moyen d'améliorer et de faciliter notre vie quotidienne grâce à des milliards d'appareils IoT « intelligents » connectés à Internet. D'autre part, les professionnels de la sécurité informatique le considèrent comme trop risqué en raison des problèmes de confidentialité et de sécurité des données que soulèvent les appareils IoT.

## **I-1-6 solutions de sécurité IoT**

### **1 – Sécuriser le réseau IoT**

Protégez et sécurisez le réseau connectant les appareils IoT aux systèmes back-end sur Internet en mettant en œuvre des fonctionnalités de sécurité des terminaux traditionnelles telles que l'antivirus, l'anti-malware, les pare-feux et les systèmes de prévention et de détection des intrusions.

### **2 – Authentifier les appareils IoT**

Mettre en œuvre des mécanismes d'authentification robustes tels que l'authentification à deux facteurs, les certificats numériques et la biométrie.

### **3 – Utiliser le cryptage des données IoT**

Protéger la vie privée des utilisateurs et prévenir les violations de données IoT, en cryptant les données au repos et en transit entre les appareils IoT et les systèmes back-end en utilisant des algorithmes cryptographiques standard et des processus de gestion du cycle de vie des clés entièrement cryptés.

### **4 – Utiliser les méthodes de sécurité IoT PKI**

Pour garantir une connexion sécurisée entre un appareil et une application IoT, utilisez des méthodes de sécurité d'infrastructure à clé publique IoT telles que le certificat numérique X.509, la clé cryptographique et les capacités de cycle de vie, y

compris la génération, la distribution, la gestion et la révocation de clés publiques/privées.

### 5 – *Utilisez l'analyse de la sécurité IoT*

Utilisez les solutions d'analyse de sécurité IoT capables de détecter les attaques et les intrusions spécifiques à l'IoT, qui ne peuvent pas être identifiées par les solutions de sécurité réseau traditionnelles telles que les pare-feux.

### 6– *Testezle matériel IoT*

Mettez en place un cadre de test robuste pour garantir la sécurité du matériel IoT. Cela inclut des tests rigoureux de la portée, de la capacité et de la latence de l'appareil IoT. Les fabricants de puces des appareils IoT doivent également renforcer les processeurs pour plus de sécurité et moins de consommation d'énergie en conservant un prix accessible.

## **I-1-7 Les contraintes à lever pour l'IoT**

Voyons maintenant certains des inconvénients de l'IoT :

- **Flexibilité** : Il existe une énorme préoccupation concernant la flexibilité d'un système IoT. Il s'agit principalement de l'intégration avec un autre système car il existe de nombreux systèmes divers impliqués dans le processus.
- **Complexité** : La conception du système IoT est également assez compliquée. De plus, son déploiement et sa maintenance ne sont pas non plus très faciles.
- **Conformité** : l'IoT a son propre ensemble de règles et de réglementations. Cependant, en raison de sa complexité, la tâche de conformité est assez difficile.

## **I-1-8 Les bonnes pratiques IoT**

Les meilleures pratiques pour l'IoT :

- Concevoir des produits sûr et fiable.
- Utiliser des protocoles d'authentification et de sécurité robustes

- Désactiver les services non essentiels
- Assurez une connectivité sécurisée
- Des algorithmes économes en énergie doivent être conçus pour que le système soit actif plus longtemps.

## **I-2 Systèmes embarqués pour l'Internet des objets [17]**

### **I-2-1 Qu'est-ce qu'un système embarqué ?**

Un système embarqué est défini comme étant un système informatique et électronique autonome. Il est dédié à la réalisation d'une tâche spécifique. La technologie des systèmes embarqués couvre de nos jours tous les aspects de la vie humaine. Les systèmes embarqués sont pilotés par des microcontrôleurs ou des microprocesseurs à basse consommation d'énergie. En outre le système d'exploitation et le logiciel applicatif sont entièrement ou partiellement « noyés » dans le « hardware ».

### **I-2-2 Qu'est-ce qu'un système embarqué dans l'IoT ?**

Un système embarqué a été traditionnellement, conçu dans un but dédié avec une connectivité limitée avec d'autres appareils. L'objectif était de traiter en temps réel les informations du monde réel provenant des capteurs.

Alors que les systèmes embarqués permettent d'envoyer et souvent d'interpréter ces données localement, Internet transmet ces données vers et depuis des services en ligne (Cloud) ou des Datacenter.

Ainsi, lorsque on souhaite qu'un appareil embarqué communique avec un écosystème (autres systèmes embarqués, Cloud, Internet), des réseaux de communication comme le Bluetooth, Wifi, 4G/LTE, LoRa et autres entrent en jeu.

Par conséquent l'IoT est l'ouverture des systèmes embarqués sur le réseau Internet (ou tout réseau privé/public) pour l'envoi et la réception des données. L'ensemble de ce réseau de systèmes embarqués, de canaux de communication et de logiciels qui effectuent ces tâches de collecte, analyse, traitement et transmission de données s'appelle l'Internet des objets.

### **I-2-3 L'importances des systèmes embarqués dans l'IoT**

Les systèmes embarqués sont une composante essentielle dans l'émergence de l'écosystème IoT. Une citation répéter par des acteurs de poids dans l'univers IoT

résume ainsi la situation : « Tous les appareils IoT ont des systèmes embarqués, mais tous les systèmes embarqués ne sont pas IoT. » en effet la plupart des IoT n'existerait pas telle que nous la connaissons sans les systèmes embarqués qui aident à les faire à fonctionner.

Bien qu'Internet soit essentiel pour transmettre des données vers et depuis des appareils IoT vers des services en ligne (Cloud, Datacenter...), ce sont des systèmes intégrés qui permettront à ces données d'être interprétées et transmises si elles sont programmées pour ça.

### **I-2-4 Comment les entreprises adoptent l'IoT**

Alors que l'IoT est plus connue pour son impact sur les appareils personnels (montre connectée, pèse-personne connecté...etc.) d'énormes progrès sont réalisés dans l'univers industriel, au point que certains parlent de 4-ième révolution industrielle ou Industrie 4.0 et ont inventé IIoT (Industrial Internet of Things) en référence aux avantages que peut apporter une augmentation de la mise en réseau et de la connectivité, à l'échelle industrielle, en termes d'impact sur la fabrication mondiale.

### **I-3 Logiciels embarqués IoT : défis [26]**

Les défis à relever lors de développement de logiciels embarqués dans l'IoT sont de même nature lorsqu'on est confronté au développement de logiciels pour des systèmes embarqués qui ne sont pas connectés à Internet. Trois de ces défis sont particulièrement cruciaux :

#### **I-3-1 Sécurité**

La sécurité du système est probablement le défi le plus grand et le plus important rencontré lors du codage d'un produit IoT. Alors qu'une grande partie du défi vient du choix du matériels permettant aux appareils de se connecter, le logiciel est ne doit pas comporter des failles susceptibles de compromettre la sécurité et l'intégrité du système. Les appareils connectés sont plus vulnérables aux attaques réseaux d'où le risque de fuite de données... etc...mais le pire scénario c'est une atteinte à la sécurité physique des personnes.

### **I-3-2 Mémoire limitée**

Les ressources limitées en particuliers l'espace mémoire des systèmes embarqués (pour diverses raisons, y compris le coût), imposent des lignes de codes embarquées optimisées autant que possible.

### **I-3-3 Mises à jour**

L'avantage significatif d'un appareil IoT est sa connectivité, ce qui implique que le logiciel intégré dans un système IoT peut être mis à jour en direct. Les mises à jour logicielles peuvent corriger des bugs qui n'avaient pas été découverts auparavant lors des tests ou corriger les failles de sécurité trouvées dans les bibliothèques de logiciels utilisées pour coder le logiciel de l'appareil. Les mises à jour des appareils peuvent également ajouter de nouvelles fonctionnalités ou améliorer celles existantes pour les utilisateurs finaux.

## **Chapitre II**

### **II-1 Commandes AT pour modem Cellulaire [2]**

#### ***II-1-1 Une introduction aux commandes AT***

Pour travailler avec un modem cellulaire il faut utiliser ce qu'on appelle les "commandes AT" à un moment donné. Les commandes AT sont essentiellement des instructions de modem. Développées à l'origine par le fabricant de modems Hayes comme moyen de faire fonctionner leurs produits de téléphonie fixe commuté, les commandes AT — le « AT » signifie « come to Attention » — sont maintenant utilisées par tous les modems, de tous types. Les commandes AT sont principalement utilisées pour configurer un modem et établir sa connexion réseau. Ils peuvent également être utilisés pour obtenir des informations sur l'état du modem et de la connexion, ce qui peut être très utile pour déboguer des applications et confirmer qu'un modem fonctionne correctement : il s'est connecté au bon réseau, utilise la bonne technologie cellulaire, a activé l'itinérance , etc. Nous verrons plus loin quelques commandes utiles pour des tâches comme celles-ci.

## II-1-2 Comment envoyer des commandes AT

Les commandes AT sont envoyées au modem sous forme de texte brut via une connexion série (UART) comprenant deux fils, un pour la réception (RX) et un pour la transmission (TX), ou via USB. Sur le terrain, un appareil IoT compatible cellulaire gèrera son modem en lui envoyant des commandes AT, mais lors du développement et du débogage de l'application, il n'est pas rare d'exploiter la connexion USB du modem. Cela nous permet de lancer un terminal et d'interagir directement avec le modem en émettant nos propres commandes AT. On fait cela avec un logiciel de terminal série (PUTTY- HYPERTERMINAL....) et après configuration de la vitesse de la connexion, du port COM, et de certains autres paramètres.

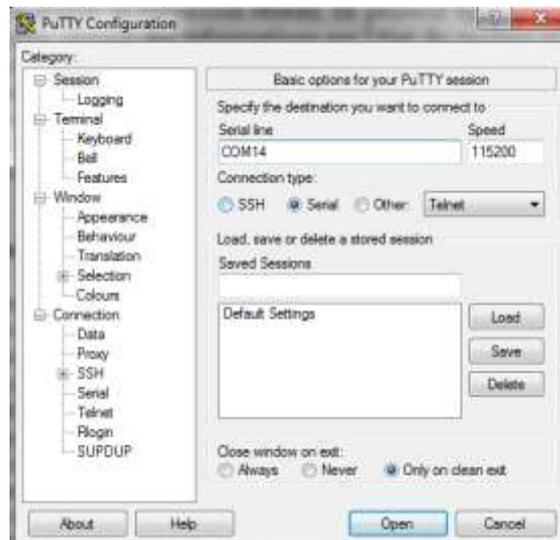


Figure II.1 Console de communication avec le modem sur terminal PUTTY

## II-1-3 Comment formater les commandes AT

Les commandes que nous envoyons doivent obéir à la syntaxe de base suivante :

```
AT<COMMAND><SUFFIX><DATA><CR>
```

C'est ce qu'on appelle une "ligne de commande". Comme on peut le voir, elle commence par **AT**, suivi d'une commande, d'un suffixe pour indiquer le mode de commande ou "type", et enfin de certaines données, bien que toutes les commandes n'aient pas besoin que ce champ soit inclus.

Cette structure est appelée 'ligne de commande' et se termine toujours par un simple retour chariot, <CR>. On peut inclure plusieurs commandes dans la même ligne, chacune séparée par un point-virgule, la longueur de la ligne est généralement, mais pas toujours, limitée à 80 caractères. Voici un exemple :

```
AT+CMEE=2;+CMGF=1;+CMGD=,4;+CNMP=38;+CMNB=1;+CGDCONT=1,"IP","supper"
```

Cette ligne définit le rapport d'erreur au niveau 2, définit le modem en mode texte, supprime tous les SMS stockés, sélectionne le mode LTE uniquement, sélectionne le mode Cat-M uniquement et définit l'APN.

## II-1-4 Champs de la ligne de commande AT

Le <COMMAND>champ indique ce que nous voulons que le modem fasse. Les anciennes commandes "de base" ont des noms à un seul caractère ; les ajouts ultérieurs au jeu de commandes, appelés commandes « étendues », sont précédés d'un signe+, tel que **AT+COPS** (rechercher les réseaux) et **AT+CGMI** (obtenir le nom du fabricant du modem). Certaines commandes sont précédées de & – par exemple, **AT&F**, qui réinitialise les paramètres du modem. Toutes les commandes cellulaires sont des commandes étendues. Selon la spécification de commande AT, les commandes doivent comprendre des caractères majuscules, mais de nombreux modems permettent également d'envoyer des commandes en minuscules. Nous nous en tiendrons à la norme.

Si on envoie une séquence de commandes AT comme indiqué ci-dessus, on doit s'assurer d'inclure le +ou &après chaque point-virgule.

Les valeurs des champs <DATA>et <SUFFIX>dépendent du type de commande, il y en a quatre :

Taper	Suffixe	Rôle
<b>Lire</b>	?	Obtenir un paramètre de configuration du modem
<b>Régler</b>	=	Définir un paramètre de configuration du modem
<b>Exécuter</b>		Déclencher une opération de modem
<b>Test</b>	=?	Vérifiez si un modem prend en charge la commande nommée

Taper	Suffixe	Rôle
		- mais notez que cela <b>AT+COPS= ?</b> fonctionne légèrement différemment.

Nous verrons plus bas comment ces types affectent l'utilisation des commandes AT, mais voyons d'abord comment le modem répond aux commandes.

## II-1-5 Réponse à la commande AT

Lorsqu'une commande AT est émise, selon la commande envoyée, le modem peut ou non renvoyer des données, c'est-à-dire faire ce qu'on appelle une « réponse d'information ». Dans tous les cas, l'opération de commande se termine toujours par le retour d'un "code de résultat" par le modem.

Par exemple, supposons que nous demandions le nom du fabricant du modem à l'aide de la commande suivante :

```
AT+CGMI
```

Cela générera la réponse suivante lorsqu'il sera appelé sur un modem SIMCOM :

```
SIMCOM
```

```
OK
```

## II-1-6 Types de commandes AT et tests

L'exemple ci-dessus peut être utilisé pour démontrer le test des commandes. On peut demander au modem s'il supporte la commande **+CGMI** :

```
AT+CGMI=?
```

Le mode test peut également être utilisé pour déterminer les paramètres pris en charge par une commande prise en charge. Par exemple, si nous envoyons la commande **+CBC** en mode test (il demande au modem quel est son niveau de batterie actuel), nous obtiendrons un résultat indiquant le nombre de paramètres et la plage de valeurs pour chacun :

```
AT+CBC=?
```

```
+CBC: (0-2),(1-100),(voltage)
```

```
OK
```

Les paramètres sont séparés par des virgules ; la plage de valeurs de chaque paramètre est présentée entre parenthèses sous la forme **<LOWEST VALUE>-<HIGHEST VALUE>**.

Envoyons la commande à un Simcom 7600G et nous verrons :

```
AT+CBC
```

```
+CBC: 0,66,3863
```

```
OK
```

## II-1-7 Comment répondre aux requêtes du modem

Parfois, le modem demandera à notre code d'application des données. Il le signalera en incluant le chevron > dans sa réponse à une commande que nous avons émise. Nous envoyons maintenant un retour chariot, **<CR>** ou "**\r**" suivi des données demandées. Enfin, nous envoyons la combinaison clavier **Ctrl-Z (CTRL26)** pour dire au modem d'utiliser les données que nous avons fournies, ou **ESC** pour annuler l'opération.

Un exemple de retour de réponses à un modem est l'envoi d'un message texte. Habituellement, on fait cela avec la commande **AT+CMG** et fournissons le numéro cible. Nous envoyons un message non enregistré en mémoire au numéro fictif « 12345678 »

```
AT+CMGS="12345678"
```

Le modem répondra avec > et notre code doit rechercher cela, puis envoyer une chaîne comprenant un retour chariot, le corps du SMS et soit **Ctrl-Z** (pour envoyer), soit **ESC** pour annuler.

## II-1-8 Quelques commandes AT clés

Le tableau suivant répertorie certaines des commandes AT les plus utiles dont nous pourrions avoir besoin lorsque nous travaillons avec un modem cellulaire.

- Commande  
**AT+CGSN**
- Exemple de réponse  
**1234567890**  
**OK**
- **Description**  
Renvoie le numéro de série du modem.

- Commande  
**AT+CPIN?**
- Exemple de réponse  
**+CPIN:READY**  
**OK**
- Description  
Confirmez que la carte SIM du modem est prête.

- Commande  
**AT+CIMI**
- Exemple de réponse  
**<imsi>**  
**OK**
- **Description**  
Obtenez l'identifiant international d'abonné mobile (IMSI) actuel de la carte SIM.

- Commande  
**AT+CCID**
- Exemple de réponse  
**<iccid>**  
**OK**
- Description  
Obtenez l'ICCID (ID de carte à circuit intégré) de la carte SIM.

- Commande  
**AT+CMEE=2**
- Exemple de réponse  
**OK**
- Description  
Rapport d'erreur étendu activé : au lieu de l'habituel **ERROR**, vous recevrez **+CME ERROR:**suivi d'un message d'erreur.

- Commande  
**AT+CSQ**
- Exemple de réponse  
**+CSQ: 4,7**  
**OK**
- Description  
Obtenez la puissance actuelle du signal au format : est une valeur dbMindexée.est un taux d'erreur sur les bits en pourcentage indexé. Un **99** dans l'un ou l'autre des champs indique que la valeur est inconnue.

- Commande  
**AT+CREG?**

- Exemple de réponse

**+CREG: 0,7**

**OK**

- Description

Obtenez l'état actuel de l'inscription au réseau au format : indique si les notifications d'état d'enregistrement réseau sont activées (1) ou non (0). Est une valeur indiquant l'état du réseau, par exemple, **0** indique que le modem n'est pas enregistré, **5** indique que le modem est enregistré et est en itinérance, et **7** indique que le modem est enregistré pour les SMS uniquement.

- Commande

**AT+COPS?**

- Exemple de réponse

**+COPS: 0,0,"OOREDOO DZ",0**

**OK**

- Description

Les champs de sortie sont : le mode de sélection de l'opérateur, le format du nom de l'opérateur, le nom de l'opérateur et le type de réseau.

- Commande

**AT+CGDCONT?**

- Exemple de réponse

**+CGDCONT: 1,"IP","internet","0.0.0.0",0,0,0,0**

**OK**

- Description

Obtenez l'APN (le troisième paramètre de réponse).

- Commande

**AT+CMGF=<MODE>**

- Exemple de réponse

**OK**

- Description

Définissez le format du message SMS. Généralement un choix entre un format binaire et du texte. Pour certains modems est un entier, pour d'autres c'est une chaîne, indiquant éventuellement le jeu de caractères à utiliser.

- Commande

**AT+CGDCONT=1,"IP","internet"**

- Exemple de réponse

**AT+CGDCONT=1,"IP","internet"**

**OK**

- Description

Définissez l'APN du modem. Il est possible de définir plusieurs APN, pour différents types d'IP (le deuxième paramètre), mais nous vous recommandons d'en définir un seul.

## II-2 L'IoT Cellulaire[22]

L'Internet des objets (IoT) connecte des produits de consommations courants (caméra, thermostat, Pèse personne, montre...etc.), et de nos jours un nombre toujours croissant de ces objets repose sur une connexion mobile.

### II-2-1 Définition

L'IoT cellulaire utilise des réseaux mobiles pour connecter des appareils physiques à Internet. Cette connexion permet l'envoi et la réception des données, intégrant ainsi ces dispositifs à l'Internet des objets.

L'IoT cellulaire s'appuie sur les technologies 2G 3G, 4G ,5G(en cours de déploiement) Low Power Area Networks(LPWAN) , LTE-M et NB-IoT pour transmettre et recevoir des données.

La popularité de l'IoT cellulaire tient essentiellement à ses avantages dont, on peut citer :

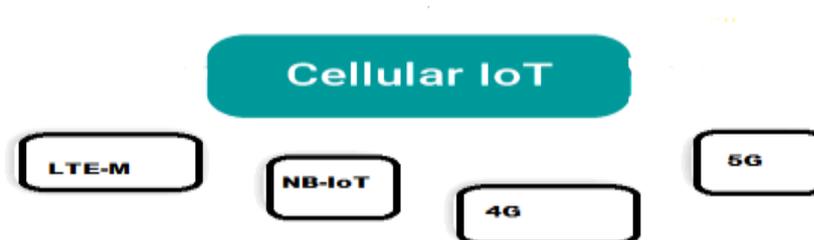
- mobilité totale

- excellent HandOver
- excellente couverture
- Connexion sécurisée grâce à des protocoles cryptographiques robuste de la SIM/eSIM
- Simplicité de la connectivité en Roaming
- Très haut débit dans les sens up/down

L'Internet des objets a connu une croissance rapide au cours de la dernière décennie. Environ 75 milliards **d'appareils IoT seront connectés à Internet en 2025.**

Jusqu'à récemment, la grande majorité des appareils IoT étaient connectés à Internet via LAN ou Wifi. Cela signifiait qu'ils étaient attachés à un emplacement spécifique ou au moins à un emplacement où la connectivité Wifi ou LAN était disponible. Aujourd'hui, les déploiements IoT industriels et d'entreprise rompent ce lien grâce à l'IoT cellulaire.

L'un des facteurs à l'origine de la popularité de la technologie IoT cellulaire est qu'elle permet de se greffer sur l'infrastructure GSM dense et fiable existante, ce qui signifie à la fois des performances et une économie améliorée. Pourtant, il est évident que les options cellulaires standard et bien établies, telles que 2G, 3G ou LTE[18], sont très gourmandes en énergie et ne sont donc pas adaptées pour les applications IoT habituelles, où seules de petites quantités de données sont envoyées par intermittence. , tels que les compteurs intelligents, les capteurs et actionneurs agricoles, les dispositifs de suivi des actifs des entreprises, les équipements de santé intelligents et bien d'autres.



**Figure II.2 -principaux types de réseaux cellulaires pour la connectivité IoT**

Les normes de la technologie des réseaux cellulaires sont maintenues par le 3rd Generation Partnership Project (3GPP) qui développe des protocoles pour chaque nouvelle

génération du réseau cellulaire. Pour le cas d'utilisation de l'IoT nous avons l'évolution suivante :

- **2G** – obsolète et mise hors service dans certains pays dans le monde, la 2G est encore largement utilisée dans les régions rurales et éloignées qui sont souvent des emplacements de choix pour les entreprises agricoles et industrielles. Les mêmes entreprises qui déploient l'IoT cellulaire.
- **3G** – La première génération d'IoT cellulaire. Les normes ont été initialement définies dans la version 3GPP1999 et mises à jour dans la version 2008 de la révision des normes 3GPP. En raison d'une bande passante limitée, les connexions 3G IoT ne pouvaient pas, par exemple, diffuser de la vidéo. Cependant, ils offraient une fiabilité suffisante pour les données de paiement dans les déploiements statiques et aidaient à jeter les bases des futures versions à haut débit.
- **4G LTE** - En 2008, 3GPP a annoncé la version , qui couvrait le Long Term Evolution , ou LTE. Avec cette version, l'accent a été mis sur la demande des utilisateurs pour des vitesses élevées et une latence réduite, ainsi que sur le besoin des fournisseurs de services de réduire les coûts. LTE est le véritable saut de la 3G vers la 4G.
- **LTE Advanced** – LTE-Advanced a été officialisé en 2011 avec la version 10 du 3GPP . Cette version a mis l'accent sur une capacité plus élevée et sur le désir de fournir des débits binaires plus élevés de manière rentable. LTE-Advanced a vu les vitesses passer à 3 Gbps en téléchargement et 1,5 Gbps en téléchargement.
- **LTE Cat.o** - Introduit lors de la version 12 du 3GPP en 2012. Le débit des liaisons montantes et descendantes a été réduit à 1 Mbps dans cette version. Le mode d'économie d'énergie a été introduit ici, ce qui a permis de prolonger la durée de vie de la batterie, ce qui le rend plus pratique pour l'IoT cellulaire.
- **LTE Cat M** - La prochaine génération de Cat.o a été annoncée avec la version 13 du 3GPP en 2016. Cette version a permis d'améliorer considérablement la durée de vie de la batterie des appareils, jusqu'à 10 ans dans certain cas. Il a permis une mobilité totale et un transfert dans le véhicule et a considérablement amélioré la connectivité à l'intérieur du bâtiment. Il a également vu des vitesses de liaison

montante et descendante de 375 kb/s en mode semi-duplex. Cela signifiait qu'il était idéal pour les applications IoT qui avaient des besoins de débit de données faibles à moyens.

- **NB-IoT** - Également défini dans le cadre de la version 13 du 3GPP en 2016 L'IoT à bande étroite (NB-IoT) a introduit des portées plus longues, une fiabilité améliorée et une durée de vie de la batterie améliorée, similaire à LTE Cat M. Il s'agit de l'un des types de connexions qui gagnent en popularité dans les déploiements IoT et qui est essentiel pour combler le fossé entre la 4G et la 5G.
- **5G** - Actuellement en cours de déploiement dans le monde, la 5G a été définie en deux étapes par 3GPP dans la version 15 en 2018 et la version 16 début 2020. La 5G poursuit le travail commencé par LTE Cat M et NB-IoT en améliorant les vitesses (jusqu'à 10 Gbps), couverture et durée de vie de la batterie considérablement augmentée.



**Figure II.3 -normes 3GPP par réseau. Source : 3GPP**

## II-2-2 Facteurs de croissance pour l'IoT cellulaire[12]

Il existe une demande croissante de connectivité entre les appareils utilisés dans diverses industries. Plusieurs facteurs stimulent la croissance de l'IoT cellulaire :

### 1. Couverture accrue

Contrairement au Wifi, les réseaux cellulaires 5G offrent une couverture beaucoup plus large et prennent en charge un nombre beaucoup plus importants d'appareils. La possibilité de déployer des réseaux 5G d'entreprise permet aux entreprises de connecter davantage d'appareils sur une zone plus étendue, avec une fiabilité et une vitesse supérieure à celles qu'elles auraient pu obtenir avec le Wifi.

## **2. Facilité et faible coût de déploiement**

Les appareils IoT non cellulaires doivent être configurés sur un réseau spécifique avant de pouvoir se connecter au monde (sélectionnez un réseau Wifi ou une connexion Bluetooth spécifique). Les appareils cellulaires, en revanche, n'ont besoin que d'une carte SIM/eSIM activée et ils sont opérationnels. Cela facilite une « scalabilité » vers le haut et vers le bas selon les besoins, sans investissement lourd en ressources.

## **3. Gestion à distance**

L'importante couverture fournie par le réseau cellulaire signifie que les objets peuvent être gérés à distance à partir d'à peu près n'importe où dans le monde. On peut surveiller à distance la connectivité, configurer les périphériques et s'assurer que les mises à jour logicielles et du firmware sont appliquées.

## **4. Maturité des protocoles de communication cellulaire pour les applications IoT (NB-IoT) [4]**

L'IoT à bande étroite (NB-IoT) développé par le 3GPP pour les appareils IoT cellulaires apporte une de nouveaux avantages pour les appareils IoT tels qu'une faible consommation d'énergie, une fiabilité accrue, et une portée plus importante.

## **5. Commutation de réseau/itinérance**

Tout comme nos téléphones portables passe d'un réseau à l'autre lorsque nous nous voyageons (c'est-à-dire en itinérance), les appareils IoT cellulaires peuvent se déplacer sans avoir à craindre de perdre le signal. Ceci est idéal pour les appareils comme les voitures qui nécessitent une connexion constante pour assurer la sécurité.

## **6. Faible latence et haute fiabilité**

De nombreuses applications critiques, telles que les communications des services d'urgence, reposent encore sur une technologie radio désuète, facile à perturber. Avec l'IoT cellulaire sur les réseaux 5G, les communications des services d'urgence peuvent être plus sûres et plus fiables, fournissant des données en temps réel aux centres de commandement.

## **7. Sécurité dès la conception**

Les réseaux 5G sont dotés de mesures de sécurité intégrées pour protéger les consommateurs. La connectivité comprend des protocoles tels que l'authentification sécurisée, la protection de la signalisation et le chiffrement des données. Ces

mesures permettent de garantir que les problèmes de sécurité qui affectent les appareils IoT grand public ne constituent pas un frein à leur utilisation industrielle.

### **8. Faible consommation d'énergie**

Certaines normes IoT cellulaires, telles que NB-IoT, sont conçues **pour** maximiser la durée de vie de la batterie et réduire la consommation d'énergie. Cela permet de déployer des appareils dans des zones reculées, où la durée de vie de la batterie atteint parfois 10 ans.

## **II-2-3 IoT cellulaire : cas d'usage les plus courants [14]**

Voici quelques cas d'usage d'utilisation de l'IoT les plus adoptés :

### **1. Surveillance des objets à distance en lecture seule**

C'est le cas d'utilisation IoT le plus simple est également le plus répandu. La surveillance des objets connectés à distance en lecture seule, on peut visualiser les données des capteurs mais on ne peut renvoyer aucune commande à l'objet surveillé). Ce cas d'utilisation est l'un des plus faciles et des moins chers à mettre en place en raison de sa simplicité.

### **2. Automatisation des processus basée sur l'IoT**

L'automatisation des processus basée sur l'IoT a été déployée par beaucoup d'entreprises. Ce type d'automatisation de processus décrit des processus opérationnels qui étaient soit entièrement manuels dans le passé, soit reposaient sur des configurations d'automatisation industrielle obsolètes, mais qui ont maintenant été mis à niveau avec du matériel et des logiciels de pointe.

### **3. Surveillance et contrôle des objets à distance (lecture/écriture)**

C'est l'extension du cas précédent où en plus de "lire" les données de l'objet, on peut également communiquer en retour et ainsi influencer le contrôle des objets (c'est-à-dire réécrire les données).

### **4. Gestion du parc de véhicules (track/trace)**

La gestion des véhicules dans une flotte est actuellement le cas d'utilisation numéro un de la chaîne d'approvisionnement IoT. Plus la flotte des moyens de transport à gérer est importante, plus la complexité est élevée. La gestion de flotte

transfrontalière peut être particulièrement lourde, c'est pourquoi beaucoup d'entreprises ont déployé une solution professionnelle de gestion de flotte de véhicules pour recueillir des informations en temps réel.

Aujourd'hui, la plupart des solutions de gestion de flotte reposent sur une connectivité étendue, telle que le cellulaire (2G, 3G, 4G). Plusieurs nouvelles technologies satellitaires IoT à faible coût ont récemment été lancées et promettent d'ajouter une autre option à ce cas d'utilisation afin que les flottes de véhicules soient toujours réellement connectées à un prix raisonnable.

### **5. Localisation (par exemple, GPS)**

Le suivi de l'emplacement (géolocalisation) d'un objet IoT est un des cas d'usage les plus fréquents adoptés par les acteurs du marché pour son utilité tant pour les fournisseurs que pour les clients.

### **6. IoT pour l'optimisation des performances des actifs/usines**

La gestion des performances des actifs (APM) est un terme utilisé dans la fabrication pour décrire les méthodes de capture et d'intégration des données, de visualisation et d'analyse pour améliorer la fiabilité et la disponibilité des actifs physiques de l'usine. Il intègre des outils de capture et d'intégration de données de pointe (comme les passerelles IoT) et des outils logiciels (comme les plates-formes IoT) pour analyser comment les actifs peuvent être exécutés et maintenus à des niveaux optimaux (par exemple, des paramètres de vitesse d'actifs optimisés, des matériaux optimisés réglés d'entrée ou intervalles de maintenance optimisés).

### **7. Contrôle et gestion de la qualité basés sur l'IoT**

Le contrôle et la gestion de la qualité basés sur l'IoT impliquent l'utilisation de la vision artificielle ou d'autres données de capteurs IoT pour détecter les problèmes de qualité en temps réel pendant les opérations.

### **8. Surveillance de l'état des marchandises en transit basée sur l'IoT**

La surveillance de l'état des marchandises est essentielle dans des secteurs tels que les produits pharmaceutiques ou l'alimentation et les boissons et peut même être considérée comme un moyen d'aider à lutter contre la pénurie alimentaire

mondiale. Les données du capteur de température sont l'une des valeurs de capteur les plus importantes pour ce cas d'utilisation ; il garantit la sécurité du produit final par une surveillance étroite tout au long de la chaîne d'approvisionnement, tant en transit qu'en stockage.

### **9. Maintenance prédictive**

Pour prévoir la durée de vie utile restante de ces outils de production et s'assurer de leur disponibilité dans les meilleures conditions fonctionnelles beaucoup d'entreprises ont investi dans des solutions à l'intersection de l'intelligence artificielle et de la maintenance.

### **10. Suivi et traçabilité sur site**

La disponibilité de capteurs et de trackers à faible coût a rendu facile et relativement bon marché le suivi des marchandises et des outils sur les chantiers de construction, les ports et même à l'intérieur des bâtiments (par exemple, les usines).

## **II-3 Technologies réseaux pour les objets connectés [25]**

### **II-3-1 Qu'est-ce qu'un réseau IoT ?**

Le réseau IoT fait référence aux technologies de communication utilisées par les dispositifs de l'Internet des objets (IoT) pour partager ou diffuser les données vers d'autres appareils ou interfaces disponibles à portée. Faire transiter les données d'un système IoT par un réseau suppose des choix à faire selon les caractéristiques de ces réseaux, en tenant compte de contraintes intrinsèques au système utilisé :

- Environnement où l'objet doit évoluer (Espace urbain, sous-terrain, rural, indoor ou outdoor).
- Distance de l'objet du point relais le plus proche (exemple : smartphone, antenne-relais, passerelle Wifi).
- La quantité d'informations, de données à communiquer par jour.
- Fréquence d'envoi des données (quelques envois, plusieurs centaines ou plus)
- Nécessité d'une prise en main à distance de l'objet

- Si l'objet doit être « joignable » périodiquement, voire à tout moment
- Délai maximum toléré pour transmettre une donnée du Système d'Information à l'objet
- Est-ce qu'il s'agit d'un objet mobile ou fixe
- Besoin de géolocaliser l'objet et si oui, quelle est la précision tolérée en mètres
- Besoin ou pas de tracer en temps réel l'objet
- Le type de source d'énergie qui alimente l'objet (secteur, batterie ou pile)
- L'objet doit-il avoir une source d'énergie limitée, combien de temps doit-elle durer ?
- Le coût de service.

### II-3-2 Les réseaux à courte ou longue portée ? [12]

Deux grandes catégories de réseaux existent sur le marché :

- **Les réseaux longue portée** : on distingue les réseaux basse consommation dits LWPAN comme Sigfox, LoRa, et des technologies cellulaires (GSM, 2G, 3G, 4G, 5G). Ces deux types de réseaux longue portée sont capables de faire transiter des data d'un appareil à l'autre sur de vastes distances. Ils sont utilisés par les entreprises qui veulent connecter des kilomètres d'infrastructures à Internet ou dans des projets de SmartCities par exemple.
- **Les réseaux à courte portée** comme le Wifi, le Z-Wave, le ZigBee, ou encore le Bluetooth Low Energy, permettent de transférer des données sur de faibles distances. Ils sont utilisés dans la domotique ou sur le marché des « wearables » grand public.

### II-3-3 Les réseaux longue portée

Parmi les réseaux de grande portée, on distingue : Sigfox, LoRa, les réseaux cellulaires GSM, le LTE-M, le NB-IoT ou encore la technologie Wize.

#### **Sigfox**

- **Sigfox** est un réseau propriétaire (protocole non ouvert...). En ville, il a une portée qui peut être supérieure à 10 kilomètres, celle-ci peut atteindre les 30 kilomètres en zones dégagées. Les appareils consomment relativement peu d'énergie pour envoyer leurs données sur ce réseau, .Sigfox, coûte en moyenne 2 euros par an et par objet, et est déployé (2021) dans 71 pays.

- En contrepartie de sa faible consommation d'énergie, Sigfox ne permet de transporter que de très faibles quantités de données, entre 10 et 100 bits par seconde maximum (bps). Monodirectionnel au départ, il permet désormais d'envoyer des informations à des objets connectés, même si le débit est encore faible pour réaliser des mises à jours rapidement.

### **LoRa**

- Le protocole très basse consommation LoRa est déployé par différentes entreprises, bidirectionnel et à faible coût (de 1 euro par an et par objet à 12 euros pour certains fournisseur), il permet de transmettre des données à des distances de 2 à 5 kilomètres en milieu urbain et jusqu'à 45 kilomètres en milieu rural. Ce protocole présente aussi l'avantage d'être développé en réseau privé pour qu'une entreprise puisse en assurer elle-même la gestion.
- Seuls 0,3 à 50kbps (kilobits par seconde) peuvent transiter sur ce réseau, qui est (tout comme Sigfox) idéal pour des capteurs émettant périodiquement une faible quantité de données de température, pression ou de géolocalisation par exemple.

### Les réseaux cellulaires GSM (2G, 3G, 4G, 5G)

- Les réseaux cellulaires, fournis par les opérateurs télécoms traditionnels, présentent l'avantage de transférer d'importantes quantités de données. La 5G annonce de grands bouleversements pour l'IoT, notamment une consommation moindre et un temps de latence très faible.
- Très gourmands en énergie ces réseaux sont à réserver à des appareils branchés au secteur. Le prix des abonnements (qui dépend de l'opérateur choisi) est nettement plus élevé que ceux de Sigfox et LoRaWAN. Par ailleurs, un espace est réservé sur l'objet connecté pour une carte SIM.

### **Le LTE-M [25]**

Conçue pour l'IoT, la technologie LTE-M Long Terme Evolution for Machines est une solution standardisée qui opère dans des bandes de fréquences sous licences allouées aux opérateurs mobiles pour la 4G. Elle bénéficie ainsi de ses fonctionnalités (sécurité avec authentification par Sim, connectivité en temps réel, itinérance à l'international), ce qui lui permet de faciliter les échanges de données enrichies, comme l'image ou la voix grâce à ses débits élevés, pouvant aller jusqu'à 10 Mb/s. Le LTE-M cible ainsi la télésurveillance ou les options combinant l'IoT et la voix.

**Le LTE-M est assez énergivore. Il a aussi pour caractéristique l'utilisation d'un modem réduit en performance par rapport aux modems LTE classiques.**

### **Le NB-IoT [25]**

Le NB-IoT (Narrow Band Internet of Things) est aussi une solution standardisée basé sur l'infrastructure de la 4G. Elle utilise une bande étroite de 180 KHz **et assure une pénétration à l'intérieur des bâtiments ou en sous-sol**, ce qui lui permet de répondre aux besoins de parcs importants d'appareils fixes nécessitant un faible volume de données, comme la télé-relève de compteurs d'eau ou de gaz. Il y aurait plus de 9 réseaux NB-IoT déployés dans le monde en 2020.

Le débit de transmission de données du NB-IoT se limite à 150 Kbits/s et le déploiement de ce standard dépend des opérateurs mobiles et de leurs investissements dans des stations de base dotées de cette technologie.

## **II-3-4 Les réseaux courte portée**

Dans la famille des réseaux courte portée, on retrouve le Wi-Fi, le Z-Wave, le ZigBee, le Bluetooth Low Energy, (BLE) et la RFID.

### **Le Wi-Fi**

- Le wifi permet de transférer un grand nombre de données rapidement, jusqu'à 600 mégabits par seconde (mbps). Ce protocole de communication peut être utilisé pour connecter des caméras de surveillance HD 24/24. Il est bidirectionnel, ce qui permet de mettre facilement à jour les appareils.
- La technologie WIFI est très consommatrice d'énergie et nécessite donc une alimentation électrique sur secteur.

### **Le Z-Wave**

- Le Z-Wave est un protocole de communication dédié à la domotique. Sans fil, il est facile à installer dans la maison. Il a une portée de base de 30 mètres. C'est un réseau maillé, c'est-à-dire que chaque appareil connecté au système est émetteur de données mais peut aussi relayer celles qui sont émises par ses voisins. Cela permet d'élargir sa portée.
- Son principal défaut et pas les moindres c'est qu'il est énergivore par rapport au BLE (Bluetooth Low energy) et les dispositifs qui fonctionnent sur batterie sont la plupart du

temps en veille pour ne diminuer la consommation électrique ce qui risque de créer des « zones blanches » empêchant de relayer les données de certains dispositifs du réseau.

### **Le ZigBee**

- Le ZigBee permet de faire circuler plus de données que le Z-Wave (jusqu'à 250 kbps, contre 100 maximum). Il est également moins cher et plus facile à implémenter pour les fabricants d'objets connectés que le Z-Wave ou le Bluetooth (dans sa version 4, dite Low Energy comme dans sa version 5, qui vient d'être standardisée).
- Mais ce réseau n'a que 10 mètres de portée en moyenne soit 20 de moins que le Z-Wave et 50 de moins que le Bluetooth Low Energy.

### **Le Bluetooth Low Energy**

- Le Bluetooth Low-Energy est très largement utilisé dans le monde. Quasiment tous les smartphones sont équipés de cette technologie, fréquemment utilisée pour faire communiquer les wearables. Il a une portée de 60 mètres en terrain dégagé et "consomme environ 20 fois moins d'énergie que le Wifi". La dernière version de cette technologie, le Bluetooth 5, est plus adaptée à l'IoT et dispose d'une portée deux fois supérieure à celle de la version antérieure.
- Ce réseau de courte portée permet de transporter nettement moins d'infos que le Wifi : 1 mbps seulement, même si le nouveau standard permet de transférer quatre fois plus de données.

### **La RFID**

- La RFID (Radio Frequency Identification) est une technologie d'identification par radiofréquence, sans aucune batterie. C'est l'énergie radio qui permet d'identifier des objets et de transmettre l'information par un code d'identification. La RFID s'utilise à courte distance - du contact directe à quelques centimètres (NFC) et jusqu'à 15 mètres en haute fréquence (UHF). Cette technologie est utilisée principalement dans l'identification, la gestion des stocks ou la traçabilité. Un lecteur doit nécessairement se situer dans le champ du TAG pour pouvoir en récupérer les données, ce qui représente une contrainte lors de l'installation. La RFID représente par ailleurs un certain coût puisqu'un investissement est fait dans l'achat de puces RFID, de balises pour la transmission des informations et dans l'intégration dans le système d'information.

## II-4 Plates-formes IoT : cas du service Thingspeak

### II-4-1 Présentation de la plateforme ThingSpeak[1]

Thingspeak est une plateforme de données ouvertes avec MATLAB Analytics, et un service WEB(API REST) pour l'Internet des objets. Cette plateforme permet de simplement collecter des flux de données de capteurs puis de les analyser et de déclencher des actions souhaitées tout ceci en temps réel sur le Cloud. Avec la possibilité d'exécuter du code MATLAB dans ThingSpeak, on peut effectuer une analyse et un traitement en ligne des données au fur et à mesure qu'elles arrivent.

En outre le service ThingSpeak donne la possibilité de créer des plugins et des applications pour collaborer avec les services Web, les réseaux sociaux et d'autres API.

ThingSpeak est souvent utilisé pour le **prototypage** et la **preuve de concept** des systèmes IoT qui nécessitent des analyses.

### II-4-2 Fonctionnalités de ThingSpeak[1]

- Collecter des données dans des canaux privés
- Partager des données avec des chaînes publiques
- API RESTful et MQTT
- Analyses et visualisations MATLAB
- Programmer des événements
- Lancer des alertes
- Intégrer des applications tierces ou personnalisées (Android, Twitter ...etc.)

### II-4-3 Principes de base de ThingSpeak[1]

-Création d'un compte ThingSpeak sur le site <https://thingspeak.com/> en tant que nouvel utilisateur.

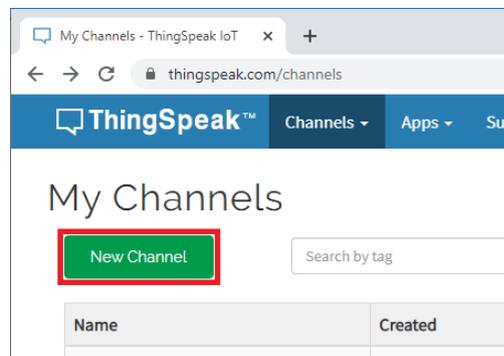
-L'élément central de ThingSpeak est un « canal ThingSpeak ». Un canal stocke les données que nous envoyons à ThingSpeak et comprend les éléments ci-dessous :

-8 champs pour stocker des données de tout type - Ceux-ci peuvent être utilisés pour stocker les données d'un capteur ou d'un appareil embarqué.

-3 champs de localisation - Peuvent être utilisés pour stocker la latitude, la longitude et l'altitude. Ceux-ci sont très utiles pour suivre un appareil en mouvement.

-1 champ d'état - Un court message pour décrire les données stockées dans le canal.

La création d'un nouveau canal se fait en cliquant sur le bouton comme indiqué dans l'image ci-dessous



**Figure II.4 Création d'un canal ThingSpeak [13]**

-Ouvrir le canal nouvellement créé et copier la clé privée API nécessaire pour écrire et publier les données sur la plateforme.

Il faut noter que le fait de cliquer sur "Générer une nouvelle clé d'API d'écriture" écrasera la clé précédente. On n'aura qu'une seule clé d'API d'écriture à tout moment. De plus, si notre chaîne est privée, les autres ne peuvent afficher le flux et les graphiques de la chaîne qu'en utilisant une clé API de lecture. Il faut générer et partager les clés API de lecture avec les personnes approuvées et autorisées à voir votre chaîne.

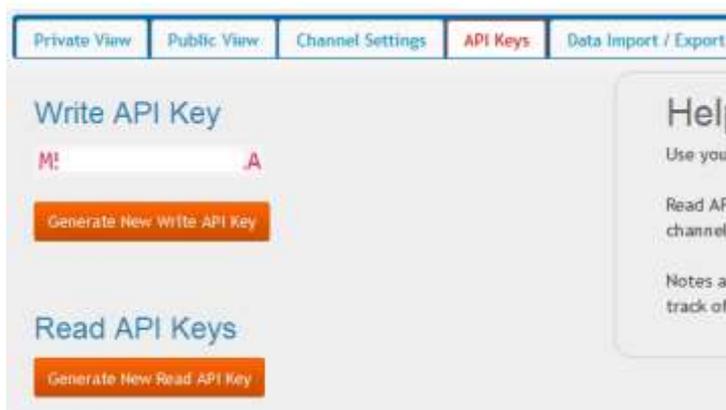


Figure II.5 Génération de la clé API ThingSpeak [13]

-Les données sont lues et publiées depuis et vers ce canal en fournissant la clé API et les noms des champs sous forme de requêtes POST ou GET du protocole HTTP/HTTPS :[1]

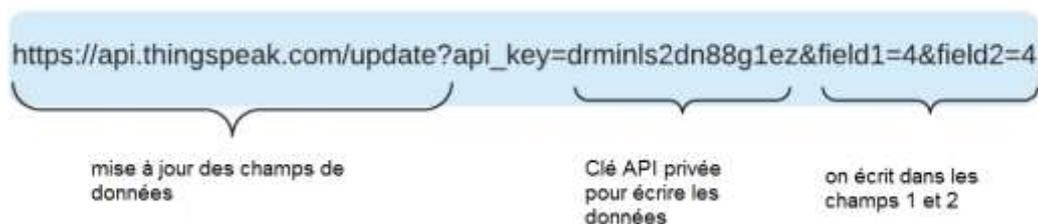


Figure II.6. Script de mise à jour d'un champ ThingSpeak [1]

Cette requête permet d'envoyer des valeurs dans les champs 1 et 2 d'un Channel.

Les requêtes fonctionnent en https ou en http sur le port 80.

**ThingSpeak™** Channels ▾ Apps Blog Support ▾ Account ▾ Sign Out

## New Channel

**Name**

**Description**

**Field 1**

**Field 2**

**Field 3**

**Field 4**

## Help

### ThingSpeak Channel

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

### Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.

**Figure II.7 Description des canaux ThingSpeak[13]**

-Selon les besoins on peut rendre la chaine privée ou public en cochant les cases appropriées. [1]

The screenshot shows the 'Using the Channel' configuration page on ThingSpeak. The page includes a navigation bar with 'Channels', 'Apps', 'Blog', and 'Support' links, and 'Account' and 'Sign Out' options. The main content area is split into two columns. The left column contains a form with the following fields: 'Make Public' (checkbox), 'URL' (text input), 'Elevation' (text input), 'Show Location' (checkbox), 'Latitude' (text input with '0.0'), 'Longitude' (text input with '0.0'), 'Show Video' (checkbox), 'Video ID' (text input), and 'Show Status' (checkbox). A green 'Save Channel' button is located at the bottom of the form. The right column has a heading 'Using the Channel' followed by explanatory text and a 'Learn More' link.

**Figure II.8 Configuration d'un canal ThingSpeak [13]**

- Après avoir enregistré la chaîne, l'écran des détails de la chaîne s'affichera (comme indiqué ci-dessous). Notons qu'il y a 5 onglets :

1. Vue privée
2. Vue publique
3. Paramètres de la chaîne
4. Clés API
5. Importation/exportation de données.

Notons l'ID de chaîne (ChannelID) dont on aura besoin pour publier et lire des données sur ThingSpeak. [1]

The screenshot shows the configuration page for a ThingSpeak channel. At the top, there is a navigation bar with 'ThingSpeak™', 'Channels -', 'Apps', 'Blog', and 'Support -'. Below this, the channel name 'Temperature container' is displayed. The channel details are as follows:

Channel ID:	153702	Thermomètre
Author:	USER	
Access:	Public	

Below the details, there are tabs for 'Private View', 'Public View', 'Channel Settings', 'API Keys', and 'Data Import / Export'. At the bottom, there are two buttons: 'Add Visualizations' and 'Data Export'.

Figure II.9 -Configuration des champs thingSpeak [13]

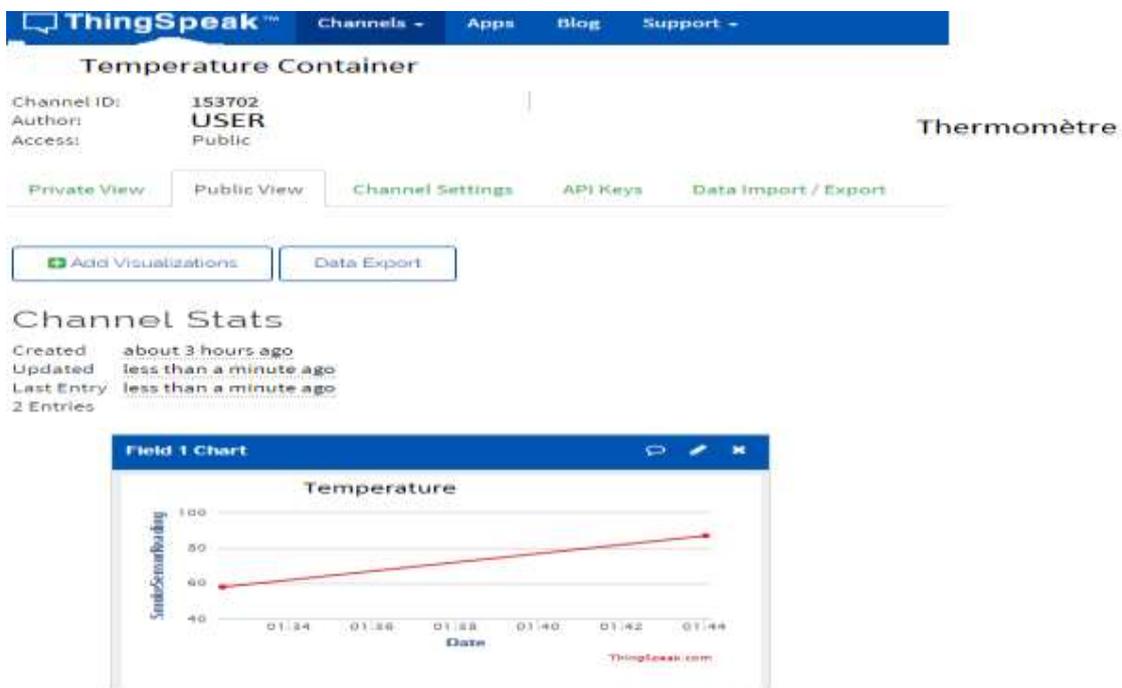


Figure II.10 -Visualisation de données sur ThingSpeak[13]

# Chapitre III

## III-1 Conception et réalisation d'un système embarqué pour une solution IoT mobile

### III-1-1 Etat de l'art

La problématique des systèmes embarqués appliqués à l'univers de l'internet des objets tient principalement à l'exigence d'une connectivité fixe ou de faible portée (wifi, Bluetooth ...etc.), ce qui limite drastiquement le vaste éventail des objets connectés nécessitant une mobilité totale.

La solution envisagée dans le cadre de ce projet consiste à mettre à profit la connectivité cellulaire à travers les réseaux innovants 4G-LTE (et bientôt la 5G...) pour intégrer toutes les solutions envisageables des objets connectés à forte mobilité dans le champ à très forte croissance qu'est l'Internet des objets.

### III-1-2 Présentation du projet

Nous exposons dans ce qui suit les détails techniques d'un système embarqué tournant sur une plateforme ESP32, muni d'un module GPS NEO-6m, d'un capteur DHT22 et d'un module SIM7600G-H pour la connectivité au réseau Internet.

L'objectif d'un tel système est d'assurer la surveillance des paramètres de fonctionnement de n'importe quel objet en situation de mobilité, sa géolocalisation, et feed-back éventuel sur le processus quand des exceptions

Surviennent dans le fonctionnement du système.

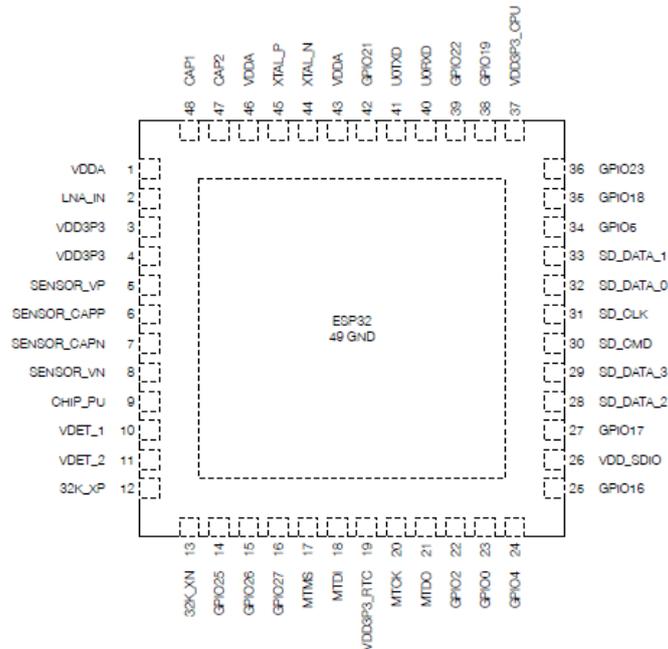
### III-1-3 SoC (System on Chip) ESP32 [7]

#### III-1-3-1 présentation

ESP32 est le nom commun d'un ensemble de plates-formes basées sur un SoC (system on Chip) développé par Espressif Systems et dont les fabricants d'appareils et les développeurs ont développé tout un écosystème technologique autour, à travers

leurs propres produits. On utilisera dès lors invariablement le nom « ESP32 » pour désigner :

- **La puce ESP32** : fabriquée par EspressifSystems , livrée non blindé et ne peut être fixée à un module ou à une carte sans soudure.



**Figure III.1 -Puce ESP32 [20]**

- **Modules ESP32** : Ce sont des modules « montables » en surface qui contiennent la puce. L'avantage d'acquérir un module est que celui-ci peut être facilement monté sur un MCU dans le processus de production. La puce est généralement également blindée.



**Figure III.2 ESP32 WIFI /Bluetooth Low Energy Module [20]**

- **Cartes de développement ESP32** : il s'agit de cartes de développement IoT MCU sur lesquelles les modules contenant la puce ESP32 sont préinstallés. Ils sont utilisés par les amateurs, les fabricants d'appareils et les développeurs pour tester et prototyper des appareils IoT avant phase de production de masse. Il existe une grande variété de marques et de modèles de cartes de développement ESP32, produites par différents fabricants.

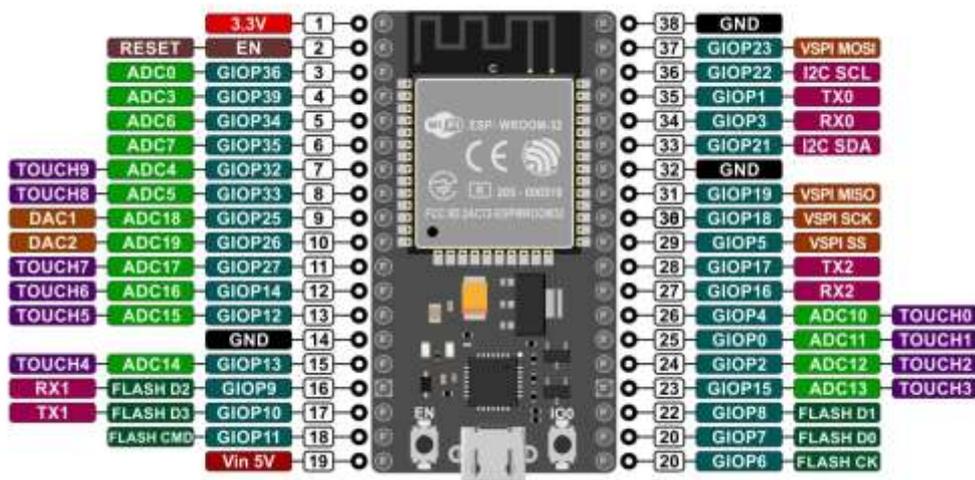
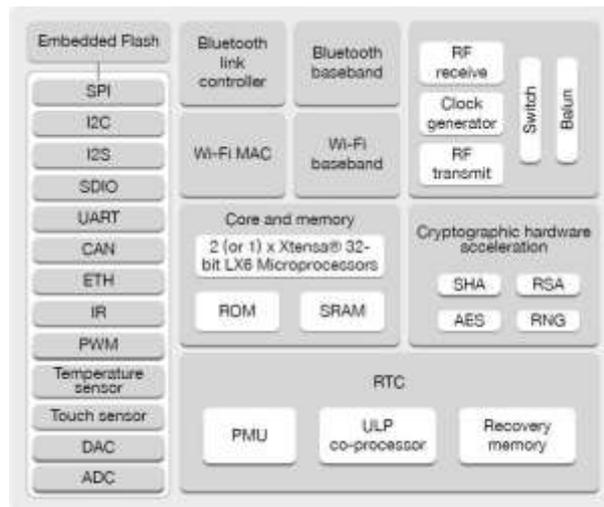


Figure III.3 Carte de développement ESP32 [19]

### III-1-3-2 Caractéristiques du SoC ESP32 :

- **Processeurs**
  - Microprocesseur Xtensa simple/double cœur LX6 32 bits, 160 MHz à 240 MHz, 600 DMIPS
  - Coprocesseur ultrabasse consommation
  - ROM de 448 Ko
  - SRAM de 520 Ko
- **Connectivité sans fil**
  - Wi-Fi : 802.11 b/g/n/e/i
  - BLUETOOTH® : v4.2 BR/EDR et BLE
  - **Sécurité**  
Norme de sécurité IEEE 802.11 WPA, WPA/WPA2 et WAPI
  - Démarrage sécurisé

- Cryptage Flash
- OTP 1024 bits, jusqu'à 768 bits pour les clients
- Accélération matérielle cryptographique : AES, SHA-2, RSA, cryptographie à courbe elliptique (ECC), générateur de nombres aléatoires (RNG)
- **Gestion de l'alimentation**
  - Régulateur interne à faible chute de tension
  - Domaine de puissance individuel pour RTC
  - Courant de veille prolongée de 5µA
  - Réveil à partir d'une interruption GPIO, d'une minuterie, de mesures ADC, d'une interruption de capteur tactile capacitif
- **Interfaces périphériques**
  - CAN SAR 12 bits jusqu'à 18 canaux
  - 2 × CNA 8 bits
  - 10 × capteurs tactiles (GPIO à détection capacitive)
  - Capteur de température
  - 4× SPI
  - Interfaces 2× I<sup>2</sup>S
  - Interfaces 2 × I<sup>2</sup>C
  - 3 × UART
  - Contrôleur hôte (SD/SDIO/CE-ATA/MMC/eMMC)
  - Contrôleur esclave (SDIO/SPI)
  - Interface MAC Ethernet avec prise en charge DMA et IEEE 1588 dédiées
  - CAN (Controller Area Network) 2.0
  - Télécommande infrarouge (TX/RX, jusqu'à 8 canaux)
  - Moteur PWM
  - LED PWM jusqu'à 16 canaux
  - Capteur à effet Hall
  - Préamplificateur analogique à très faible puissance



**FIGURE III.4 BLOCK DIAGRAM System on Chip ESP32 [20]**

- Voici quelques types de carte de développement ESP32 disponibles sur le marché avec des spécifications assez diverses pour s'adapter aux besoins de chaque utilisation :



**Figure III.5 Quelques variantes de carte de développement ESP32**

Sur la base de son rapport qualité-prix, de sa petite taille et de sa consommation d'énergie relativement faible, la plateforme ESP32 est bien adaptée à un certain nombre d'applications IoT différentes. Ces systèmes sont le plus souvent conçus pour les appareils mobiles, les technologies portables et les applications IoT. De plus, avec Mongoose OS introduisant un kit de démarrage ESP32 IoT, l'ESP32 a acquis une réputation de puce ou de module ultime pour les amateurs et les développeurs IoT.

L'ESP32 a connu depuis sa sortie une évolution fulgurante de ces capacités et de ses ressources que de grands groupes industriels l'ont intégré dans leurs produits et à titre d'exemple PANASONIC Industry l'utilise sur sa gamme PAN9520 Series en tant que kit d'évaluation pour le prototypage rapide.

### III-1-4 Choix de la plate-forme ESP32

Pour notre projet nous avons opté pour une carte de développement ESP32 basée sur la puce ESP-WROOM-32.

#### III-1-4-1 Caractéristiques de la carte:

- Microcontrôleur: Espressif ESP32 (monté sur puce ESP-WROOM-32)
- Microprocesseur: Tensilica Xtensa Dual-Core 32-bit LX6
- Fréquence d'horloge: 80 à 240 MHz (réglable)
- Mémoire Flash: 4MB
- ROM: 448 kB
- SRAM: 520 kB
- Bluetooth 4.0 (BLE/Bluetooth Smart) and Bluetooth Classic (BT)
- 802.11b/g/n HT40 Wi-Fi transceiver
- Alimentation: 5 à 9V
- Entrées/sorties digitales: 48 GPIO pins
- Courant par broche E/S: 40 mA
- Courant par broche 3.3V: 50 mA
- Pilote: CP2102
- Dimensions 55 x 28 x 13 mm

#### III-1-4-2 Avantages

- WiFi+ Bluetooth
- Microprocesseur double-cœur
- Programmable en C++ et en Python
- Prix très abordable
- Consommation d'énergie bien adaptée au besoin du projet

#### III-1-4-3 outils de développement (SDK) utilisés pour les ESP32.

On trouve une vaste panoplie de kits de développement logiciel (SDK). Ceux-ci permettent aux développeurs de programmer la puce directement sans utiliser de MCU séparé.

Espressif fournit un SDK officiel à utiliser avec l'ESP32, l'ESP32-2 ou l'ESP8266. Il s'agit du SDK basé sur FreeRTOS.

Outre le SDK Espressif, il existe de nombreux SDK commerciaux et open source sur le marché, notamment :[8]

- ESP Arduino Core - Micrologiciel basé sur C++
- ESP-SDK-Tools - SDK intégré ouvert pour ESP8266
- Espruino - SDK et micrologiciel Javascript
- Micropython - Python pour les appareils embarqués
- Mongoose OS - OS open source C ou Javascript
- NodeMCU - Micrologiciel open source basée sur Lua, similaire à Node.js
- Zerynth – Framework Python pour l'IoT

### III-2 Module de communication et de positionnement BK-SIM7600G-H



Figure III.6 Kit de développement BK-SIMCOM7600G-H [21]

Basé sur la puce SIMCOM, SIMCOM7600G-H, cette carte de développement BK-SIMCOM7600G-H de AND TECHNOLOGIES CO LTD est un module de communication et de positionnement GNSS 4G/3G/2G, qui prend en charge LTE CAT4.

#### III-2-1 Caractéristiques de la puce SIMCOM7600G-H

##### III-2-1-1 Caractéristiques générales

- SIM7600G-H est un module de communication multi-band: LTE-FDD/LTE-TDD/HSPA+/UMTS/EDGE/GPRS/GSM au format LCC.
- Contrôle Via des commandes AT de base et étendues.
- Plage de tension d'alimentation: 3.4v~4.2v, Typiquement : 3.8V
- Dimension: 30 x 30 x 2.9mm
- Température de fonctionnement :-40°C à +85°C
- Poids 5.7 g

### III-2-1-2 DATA

- LTE CAT4 : Uplink up to 50Mbps, Downlink up to 150Mbps
- HSDPA/HSUPA - Uplink up to 2.2Mbps,- Downlink up to 2.8Mbps
- HSPA+ Uplink up to 5.76Mbps , Downlink up to 42.0Mbps
- UMTS Uplink/Downlink up to 384.0Kbps
- EDGE - Uplink/Downlink up to 236.8Kbps
- GPRS- Uplink/Downlink up to 85.6Kbps

### III-2-1-3Autrescaractéristiques

- USB2.0 with High speed up to 480Mbps
- UART
- (U)SIM card (1.8V /3.0V)
- Digital Audio through PCM
- SGMII\*
- MMC/SD
- SDIO
- ADC
- I2C
- GPIO
- Antenna: Support Primary/Rx-diversity /GNSS

### III-2-1-4 Interfaces

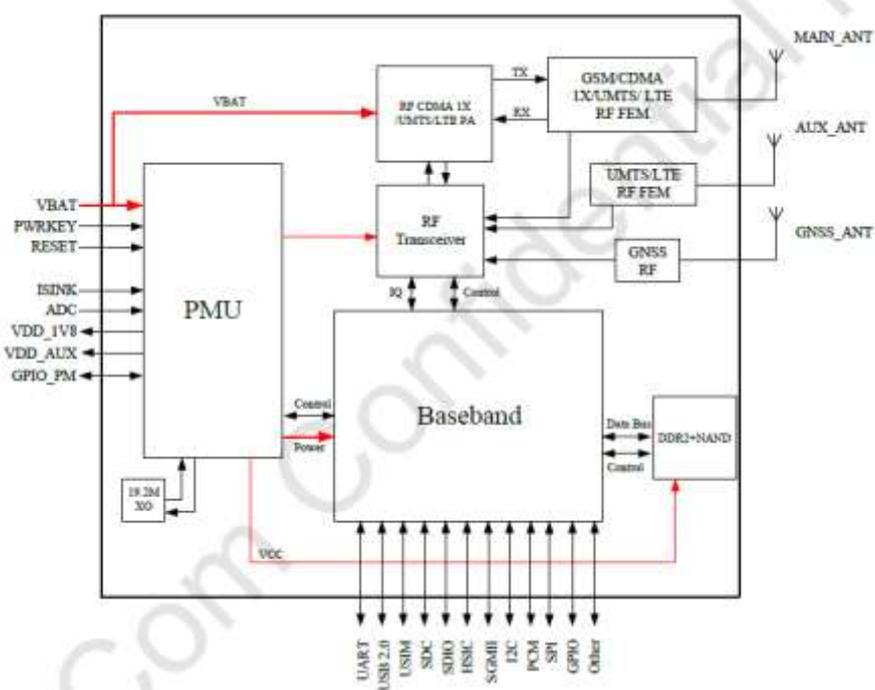
- USB Driver for Microsoft Windows 2000/XP/Vista/Win7/Win8/Win10
- USB Driver for Linux /Android
- RIL supporting for Android 2.3/4.0/5.0/6.0/7.0
- MBIM to Win8
- Firmware update via USB

- TCP/IP/IPV4/IPV6/Multi-PDP/FTP/FTPS /HTTP/HTTPS/DNS
- SSL3.0/TLS1.0/TLS1.2
- DTMF (Sending and Receiving)
- Audio Playing
- USB Audio and VoLTE
- FOTA

**III-2-1-5 Bandes de fréquence supportée**

LTE-FDD	B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66
LTE-TDD	B34/B38/B39/B40/B41
UMTS/HSDPA/HSPA+	B1/B2/B4/B5/B6/B8/B19
GSM/GPRS/EDGE	850/900/1800/1900MHz

**NB: La version SIM7600G où le G pour “GLOBAL” signifie que la puce inclue toutes les bandes de fréquences utilisées par les opérateurs télécoms à travers le monde.**



**Figure III.7 Block Diagramme de chip SIMCOM7600G [3]**

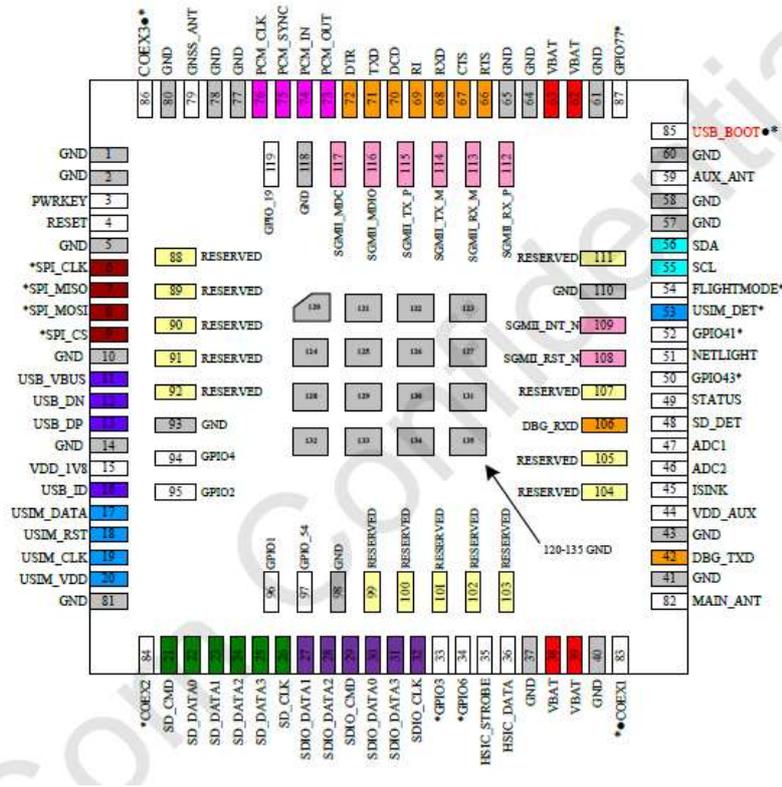
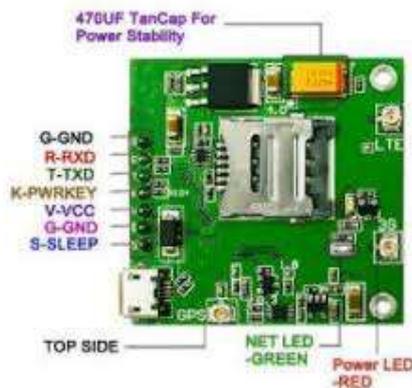


Figure III.8 -Assignment des broches puce SIM7600G et SIM7600G-H [3]

### III-2-1-6 Le module KB-Simcom7600G-H

Le module KB-SIMCOM7600G-H est un circuit imprimé sur lequel sont montés des circuits annexes pour faciliter le montage. Ainsi on trouve sur le module un circuit d'adaptation de niveau logique (logical shifter) à 3.3V tension de service de l'UART [5] SIMCOM7600G, un régulateur de tension 5V à 3.8V typique au SIMCOM7600G. Une interface USB pour faciliter la connexion directe au PC.

### III-2-1-7 Brochage du module BK-SIM7600G-H



Pin name	Pin number	Mark on PCB	Description	Comment
SLEEP	0	S	control module into sleep mode	Connect with DTR via Q102
GND	1,6	G	GND	
VCC	2	V	The power supply	5V-10V
PWRKEY	3	K	Power on/off the module	Shorted with GND via R104 by default
TXD	4	T	Transmit Data	TTL Series data Output
RXD	5	R	Receive Data	TTL Series data Input

Figure III.9 Brochage du module : [21]

### III-3 Capteurs de température DHT22[15]

#### III-3-1 présentations du module



Figure III.10 HT22 breakout Board

Le circuit DHT22(type breakout board) utilise un capteur d'humidité capacitif et une thermistance pour mesurer la température et l'humidité de l'air ambiant. **Les données sont émises via un signal numérique.**

Le capteur a une tension d'entrée variable (3,3 à 6V) ce qui permet de le connecter à un grand nombre de plates-formes MCU du marché. Sa bonne stabilité à long terme et sa faible consommation électrique le rendent intéressant pour des applications IoT.

#### III-3-2 principales caractéristiques

- Puce : DHT22/AM2302
- Compatible Raspberry, Arduino (3.3 à 6V)
- Plage de mesure : -40 à 80 degrés Celsius

- Humidité : 0 à 100 %
- Particularités très bonne stabilité à long terme
- Dimensions 45 x 15 x 10 mm
- Connexions 3 broches : VCC, sortie numérique, GND
- Protocole de communication 1 fil
- Précision de résolution 0,1
- Précision de mesure Température :  $\pm 0,5$  °C.
- Plage d'humidité de l'air 0 - 100 %
- Consommation électrique 1 - 1,5 mA
- Consommation électrique en veille 40 - 50 $\mu$ A
- Taux d'échantillonnage 2 secondes

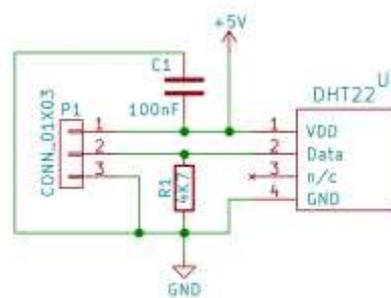


Figure III.11 Câblage interne du module

### III-4 Module GPS NEO-6m

#### III-4-1 présentation du module



Figure III.12- Module GY-NEO6MV2

Le NEO-6M est un module de géolocalisation basé sur la puce GPS NEO-6M-0-001de U-blox[9]. Il est conçu pour fournir la plupart des fonctionnalités GPS NMEA utiles - position, date, heure, altitude, vitesse et cap – avec des dimensions très appréciables.

Ce module compact peut suivre près de 22 satellites sur près de 50 canaux.

Le module vérifie simplement sa position sur terre grâce aux informations de triangulation satellites et fournit des données de sortie qui sont entre autres la longitude et la latitude de sa position. Il fait partie d'une famille de récepteurs GPS autonomes de haute performance 6. Les options d'alimentation et de mémoire rendent les modules NEO-6 idéaux pour les appareils mobiles fonctionnant sur batterie avec des contraintes de coût et d'espace très souples.

Il consomme 45 mA de courant et possède la fonction « Power Saving Mode », ce qui réduit la consommation de courant à 11 mA. Le circuit imprimé comporte une LED qui clignote lorsque le verrouillage se fait avec différents satellites, opération qui peut prendre quelques secondes selon l'endroit où l'on se trouve.

### **III-4-2 caractéristiques principales.**

- Plage d'alimentation : 3 V à 5 V
- Modèle : GY-GPS6MV2
- Antenne en céramique
- EEPROM pour sauvegarder les données de configuration
- Batterie de sauvegarde
- Indicateur de signal LED
- Taille de l'antenne : 25 x 25 mm
- Taille du module : 25 x 35 mm
- Débit en bauds par défaut : 9600 bps
- Récepteur GPS autonome
- Technologie anti-brouillage
- Interface UART aux broches de sortie (peut utiliser SPI, I2C et USB en soudant des broches au noyau de la puce)
- Moins de 1 seconde de temps pour le premier correctif pour les démarrages à chaud et assistés

- Type de récepteur : 50 Canaux - Fréquence GPS L1 - SBAS (WAAS, EGNOS, MSAS, GAGAN)
- Time-To-First-fix : pour le démarrage à froid 23s, pour le démarrage à chaud <1s
- Fréquence maximale de mise à jour de la navigation : 5 Hz
- Sensibilité : -160dBm
- Limites de fonctionnement : Gravité : 4g, Altitude : 50000m, Vitesse : 500m/s
- Plage de température de fonctionnement : -40 °C À +85°C

## III-5 Réalisation du projet

### III-5-1 Partie Hardware

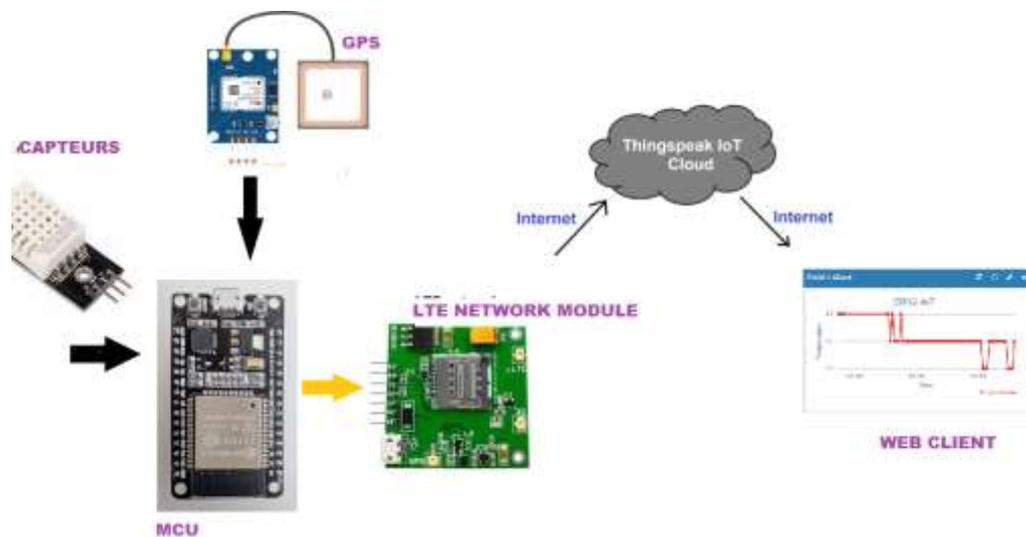
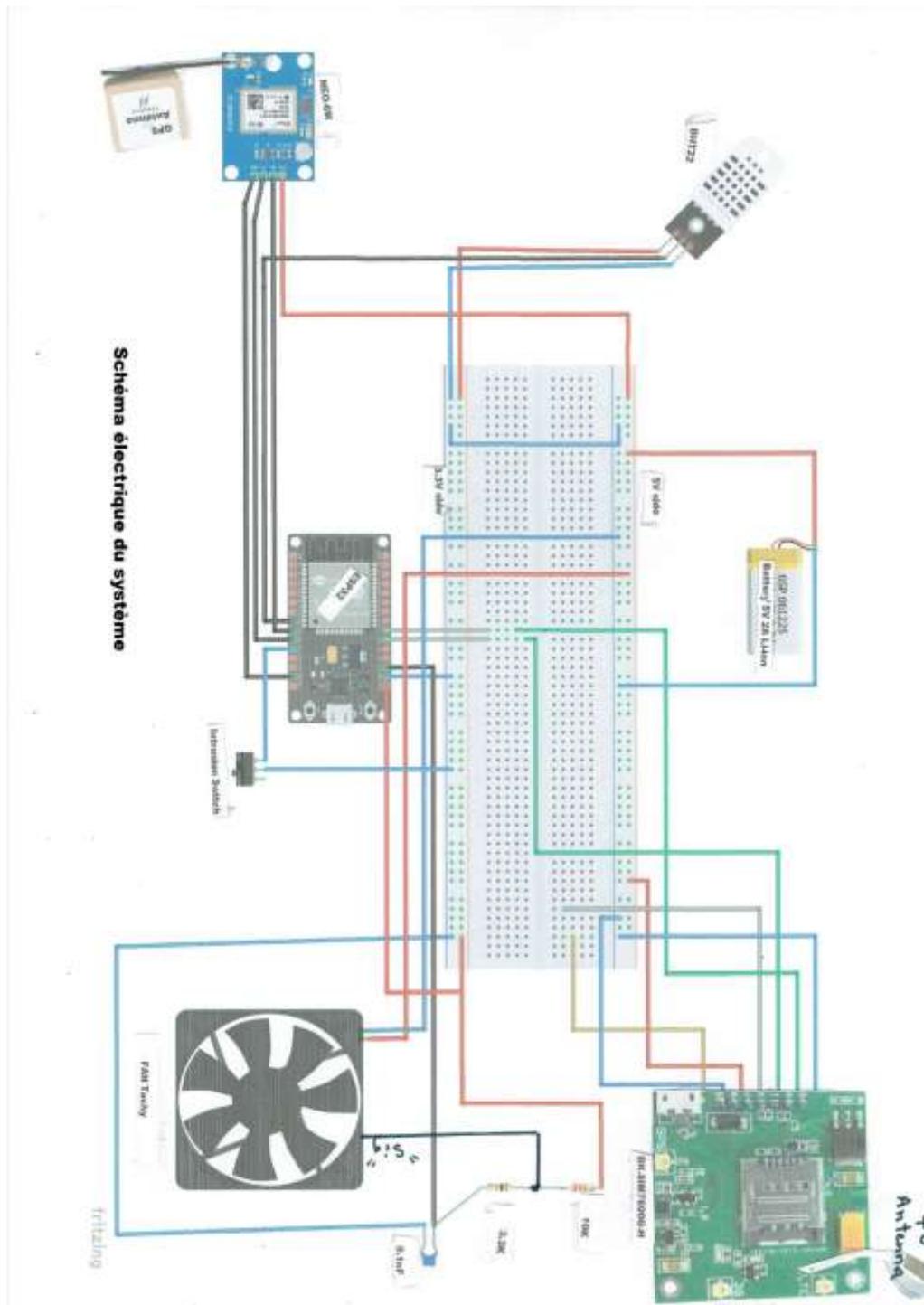


Figure III.52- Principe d'une plateforme IoT sur système embarqué

### III-5-2 Schéma électrique du système



**Figure III.14 - Schéma électrique du système**

### III-5-3 Connexions des modules à l'ESP32

Le système est destiné à géolocaliser et surveiller un objet en utilisant module à base de capteurs DHT22, d'un module GPS NEO-6M et d'un module 4G-LTE pour la connectivité Internet le tout piloté par un MCU de type ESP32.

Le DHT22 délivrant directement un signal numérique échantillonné par une puce interne au module, nous utilisons La broche GPIO25 de l'ESP32 pour lire la température ambiante.

Le module GPS NEO-6M est connecté au broches GPIO26 et GPIO27 de l'ESP32 remapper sur la **Serial1** pour la circonstance à 9600 bauds

**NB : Le circuit ESP32 possède 3 canaux UART (0,1,2) qui sont assignés à des pins fixes. Il est néanmoins possible de redéfinir les ports 'Serial' sur d'autres pins...**

Le module KB-SIM7600G-H est connecté à l'ESP32 sur la **Serial2** par défaut mapper sur les ports GPIO16 et GPIO17 à 115200 bauds.

Un ventilateur type 3 fils (muni d'un tachymètre grâce à un capteur à effet hall et un petit aimant mobile) assurera le refroidissement du système et dont la vitesse de rotation sera surveillée à distance pour s'assurer des conditions de fonctionnement optimales de l'ensemble des éléments. La broche « tacho » du ventilateur délivre un signal numérique directement exploitable sur la broche GPIO15 de l'ESP32., ce dernier comportant des entrées/sorties de 3.3V une adaptation du niveau de tension de 5V à 3.3V est réalisée à travers un classique circuit RC.

### III-5-4 Software de commande du système

La compilation des sketch 'esp32 se fait aussi avec l'EDI Arduino puisqu'il suffit d'ajouter les drivers nécessaires dans le « gestionnaire de cartes » de l'IDE arduino.

### III-5-5 Bibliothèques utilisées pour la gestion des modules

La plupart des « shields » compatible Arduino, ESP8266 ou ESP32...etc., ont des bibliothèques écrites par des volontaires (dans le cadre de l'initiative GitHub par exemple...) et mises à jours en fonction des bug's relevés ou de l'évolution du matériel, chaque bibliothèque est documentée pour permettre une utilisation simple pour les concepteurs de projets.

A ce titre nous utiliserons :

-La librairie « DHTLib » pour le breakoutboard DHT22.

-La librairie « TinyGPSPlus » pour le module GPS NEO-6M.

-Le module BK-SIM7600G-H sera lui, directement sollicité à travers ses commandes AT constructeur (très riches du reste...),ce qui nous a permis de comprendre en profondeur les mécanismes sous-jacents des protocoles HTTP/HTTPS [6] dans le cadre de l'utilisation IoT à travers les réseaux 4G-LTE.

De même, et bien que la plateforme « ThingSpeak » ait mis à disposition des développeurs une riche interface API pour faciliter la connexion au Cloud,nous utiliserons, dans le cadre de ce projet directement des requêtes POST/GET du protocole HTTPS à travers les commandes AT et les entrées/sorties « Serial » ESP32 pour publier/lire les données .

### III-5-6SoftWare : code ESP32 du système

Le code suit la configuration classique de l'IDE arduino :

- Déclaration des librairies utilisées, ici « DHT » et « TinyGPSPlus »[16]
- Déclaration et initialisation des différentes variables et constantes utilisées, on y trouve notamment la clé « **ApiKey** »[1] pour publier et lire sur l'interface web « ThingSpeak ».
- Instanciation des classes utilisées
- Appel de la fonction « **setup()** » :cette fonction est appelée au démarrage du programme. Elle est utilisée pour initialiser les variables, le sens des broches, les librairies utilisées, elle est exécutée une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte **ESP32**.Dans notre cas on inclut dans cette fonction :
  1. La déclaration et la configuration des ports « **serial** ».
  2. Activation du module KB-7600G-H est connexion au réseau GSM/LTE.
- Appel de la routine principale du programme **loop()**:
  1. Vérifier présence de données GPS correctement encodées
  2. Relève de la température ambiante du système.

3. Vérification du bon fonctionnement de la ventilation par contrôle de la vitesse de rotation des pales.
4. Mise en forme des scripts à envoyer sur ThingSpeak.
5. Initier une session HTTPS typique : ouverture connexion sécurisée par le client, publication des données collectées (Température, coordonnées GPS, Tachymètre) sur ThingSpeak, fermeture de la session, temporisation de 15S (imposée par ThingSpeak pour les comptes gratuits...) .
6. Vérifier si présence de SMS d'urgence (exemple connexion LTE perdue...) et réagir en conséquence toujours par SMS : Reset, mise en veille du système.
7. Vérification du système anti-intrusion par lecture d'un **Switch** sur le port GPIO14 et envoi de SMS de signalisation
8. Retour au programme principal.

## Conclusion Générale et perspectives

La couverture 4G LTE est ce qui se fait le mieux de nos jours en matière de réseaux de communications cellulaire (en attendant le déploiement massif de la 5G déjà commercialisée dans beaucoup de pays développés...), mais la 4G-LTE n'est pas la meilleure option pour de nombreux appareils IoT.

**Le projet réalisé ici dépend entièrement de cette connectivité LTE-réseaux d'opérateurs, proposée actuellement par tous les opérateurs télécoms mobiles en Algérie avec des débits optimisés pour le streaming et le téléchargement de fichiers volumineux.**

Au vu de ce constat, il est évident que **de tels réseaux sont surdimensionnés et le service proposé à des prix relativement élevés pour un usage limité au monde de l'IoT**, en effet la lecture d'un capteur de température sur un système embarqué mobile ou la relève de l'index d'un compteur de gaz ne consomme effectivement que quelques centaines d'octets par jour et par capteur.

Sous d'autres cieux et conscients de ce dilemme, des acteurs privés ont sauté sur l'occasion pour s'accaparer ce marché de niche et proposé des solutions de connectivité **non cellulaire** conçues spécifiquement pour l'univers IoT et à des tarifs assez attractifs, qui seront revus à la baisse quand la concurrence jouera pleinement entre ces opérateurs et les nouveaux venus sur le marché.

Parmi les plus connues de ces plates-formes **non cellulaires** de réseaux IoT, citant SigFox et LoRa deux opérateurs privés. La réponse des opérateurs cellulaires classiques ne s'est pas faite attendre puisque deux normes de communications IoT-Ready fonctionnant sur des infrastructures bien établis ont vues le jour : le LTE-M(LTE for machines) et le NB-IoT (Narrow Band Iot) et constituent une forte concurrence aux réseaux privés cités plus haut.

Pour conclure nous pouvons dire que dans l'état actuel des technologies réseaux disponibles dans notre pays ,aucune n'est spécifiquement destinée au monde de l'IoT

nous avons de fait conçu et réaliser notre projet en « l'adaptant » à un réseau dédié à l'origine à la voix et la data Internet, en attendant l'avènement en Algérie de réseaux privés et public « taillé » sur mesure pour le monde de l'IoT.

### Références Bibliographiques

- [1] <https://thingspeak.com>
- [2] SIM7500\_SIM7600 Series\_AT Command Manual\_V1.01(SIMcom)
- [3] <https://www.simcom.com/product/SIM7600X.html>
- [4] <https://www.matooma.com/fr/s-informer/actualites-iot-m2m/ltem-avantages-specificites-techniques>.
- [5] SIM7X00\_Series\_UART\_Application\_Note\_V1.00
- [6] SIM7500\_SIM7600\_Series\_HTTP(S)\_Application Note\_V3.00(SIMcom)
- [7] <https://en.wikipedia.org/wiki/ESP32>
- [8] [https://www.tutorialspoint.com/esp32\\_for\\_iot/installing\\_the\\_esp32\\_board\\_in\\_arduino\\_ide](https://www.tutorialspoint.com/esp32_for_iot/installing_the_esp32_board_in_arduino_ide).
- [9] [https://content.u-blox.com/sites/default/files/products/documents/NEO-6MSheet\\_GPS.pdf](https://content.u-blox.com/sites/default/files/products/documents/NEO-6MSheet_GPS.pdf)
- [10] <https://www.oracle.com/dz/internet-of-things/what-is-iot/>
- [11] [https://fr.wikipedia.org/wiki/Internet\\_des\\_objets](https://fr.wikipedia.org/wiki/Internet_des_objets)
- [12] <https://www.journaldunet.com/ebusiness/internet-mobile/1511189-l-industrie-iot-converge-vers-le-cellulaire-qui-s-affirme-en-tant-que-connectivite-du-futur/>
- [13] <https://www.mathworks.com/help/thingspeak/createchannel.html>
- [14] <https://www.objetconnecte.com/iot-cellulaire-dossier/>
- [15] <https://github.com/Erriez/ErriezDHT22>
- [16] <https://github.com/mikalhart/TinyGPSPlus/blob/master/src/TinyGPSPlus.h>
- [17] <https://ubidots.com/blog/iot-embedded-systems/>
- [18] [https://fr.wikipedia.org/wiki/LTE\\_\(réseaux\\_mobiles\)](https://fr.wikipedia.org/wiki/LTE_(réseaux_mobiles))
- [19] <http://esp32.net/>
- [20] <https://www.espressif.com/en/products/modules>
- [21] AND technologies,co,ltd
- [22] <https://actualiteinformatique.fr/internet-of-things-iot/iot-cellulaire>
- [23] <https://www.silicon.fr/iot-plus-de-25-milliards-de-connexions-cellulaires-dobjets-en-2025-170911.html?print=pdf>
- [24] <https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/>
- [25] <https://iotindustriel.com/technologies-solutions-iiot/technologie-nb-iot-vs-lte-m/>
- [26] [https://www.openscience.fr/IMG/pdf/iste\\_ido18v2n1\\_1.pdf](https://www.openscience.fr/IMG/pdf/iste_ido18v2n1_1.pdf)