



الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieure et de la Recherche  
Scientifique

جامعة وهران 2 محمد بن أحمد  
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي  
Institut de Maintenance et de Sécurité Industrielle  
Département de Maintenance en Instrumentation

## MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Génie Industriel

### Thème

# Développement d'Un Système Connecté a Base de Protocol MQTT (Message Queue Telemetry Transport)

Présenté et soutenu publiquement par :

Hamdad Abdelkader

et

Chaoui Djaafer

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Lalaoui Mohamed El Amine	MAA	IMSI-Univ. D'Oran2	Président
Chennoufi Mohammed	MCA	IMSI-Univ. D'Oran2	Encadreur
Rouan Serik Mehdi	MCB	IMSI-Univ. D'Oran2	Examinateur

Année 2021/2022

# *Remerciement*

*Je remercie vont premièrement Dieu le tout puissant pour la volonté, la santé, et la patience, qu'il nous a données durant toutes ces années d'étude.*

*Nous tiens à remercier grandement nos Encadreur*

*Chennoufi Mohammed*

*pour sa grande disponibilité et ses précieux conseils.*

*Nous remerciant également tous les enseignants du département maintenance en instrumentation d'université d'oran 02*

*plus spécialement les membres de jury de notre travail.*

*Enfin, nous adressons nos plus sincères remerciements à tous*

*nos amis et collègues qui nous ont toujours*

*soutenue et encouragée au cours de*

*la simulation de ce mémoire.*

*Merci à tous.*

# Sommaire

## Chapitre I: l'Internet des objets

- I.1. Introduction
- I.2. Que ce que l'Internet des objets
- I.3. Historique sur l'Internet des objets
- I.4. L'IOT aujourd'hui
- I.5. Le futur de l'IOT
- I.6. Internet des objets et son importance
- I.7. Internet des objets comme un système
  - I.7.1. L'Internet des objets comme un système des systèmes
  - I.7.2. Réseaux et Internet des Objets
  - I.7.3. L'Internet des Objets et les données
  - I.7.4. L'Internet des Objets et les nouveaux services
- I.8. Que ce qu'un objet connecté
  - I.8.1. Les avantages d'un objet connectés
  - I.8.2. Les inconvénients d'un objet connecté
- I.9. L'architecture de l'Internet des objets :
- I.10. Les applications de l'Internet des objets
- I.11. L'importance des services
  - I.11.1. 80 % de la valeur ajoutée des TIC
  - I.11.2. Des plateformes de données par communautés
  - I.11.3. La nécessité de l'identification des objets
- I.12. Les problèmes de l'Internet des objets
- I.13. Conclusion

## Chapitre II: Protocol MQTT

- II.1. Introduction
- II.2. Mode de fonctionnement
  - II.2.1 Topologies MQTT
  - II.2.2 Fonctionnement
    - II.2.2.1 Connexion et Déconnexion
    - II.2.2.2 Abonnements et Publications
  - II.2.3 Topic et motifs de filtrage
  - II.2.4 Qualité de service
  - II.2.5 Dernière volonté
  - II.2.6 Sécurité
- II.3 Description informelle du protocole
  - II.3.1 Représentations de données
  - II.3.2 Structure d'un paquet de contrôle MQTT
    - II.3.2.1 Entête fixe
    - II.3.2.2 CONNECT

- II.3.2.3 CONNACK
- II.3.2.4 PUBACK, PUBREC, PUBREL, PUBCOMP
- II.3.2.5 SUBSCRIBE
- II.3.2.6 SUBACK
- II.3.2.7 UNSUBSCRIBE
- II.3.2.8 UNSUBACK
- II.3.2.9 PINGREQ, PINGRESP
- II.3.2.10 DISCONNECT
- II.3.3 La synchronisation des requêtes envoyées au broker
- II.3.4 L'identifiant de client MQTT
- II.3.5 Callbacks
  - II.3.5.3 MessageArrived
  - II.3.5.2 DeliveryComplete
  - II.3.5.1 ConnectionLost

## **Chapitre III: Matériaux et composants**

- III Introduction
  - III.1 La carte de Arduino
    - III.1.1 Le type de la carte
    - III.1.2 Composantes de la carte Arduino UNO
  - III.2 Servo-moteur
    - III. 2.1 Les types de servo-moteur
    - III.2.2 Composition d'un servo-moteur
    - III.2.3 Principe de fonctionnement d'un servo-moteur
  - III.3 Capteur Infrarouge
    - III.3.1 Définition
    - III.3.2 Principe de fonctionnement
    - III.3.3 Caractéristiques Techniques
  - III.4 Conclusion

## **Chapitre IV: Matériaux et composants**

- IV. Introduction
  - IV.1 Simulation des different circuits
    - IV.1.1 Simulateur Isis Proteus
    - IV.1.2 Interface utilisateur
    - IV.1.3 Langage Arduino
      - IV.1.3.1 Définition
      - IV.1.3.2 Interface d'utilisateur
  - IV.2 Schéma de parking intelligent
  - IV.3 Réalisation du parking intelligent
  - IV.4 Exécuter le projet
  - Conclusion générale

## Liste des figures

- Figure 1 : Internet des objets
- Figure 2 : L'utilisation de l'Internet des objets
- Figure 3 : L'Internet des objets aujourd'hui
- Figure 4 : Réseaux et Internet des Objets
- Figure 5 : L'architecture de l'Internet des objets
- Figure 6 : Domaines d'application de l'IOT
- Figure 7 Google Trends pour MQTT (18/02/2017)
- Figure 8 Exemples de topologies MQTT
- Figure 9 Souscription au topic
- Figure 10 Niveaux de Qualité de Service MQTT
- Figure 11 Format d'un paquet CONNECT
- Figure 12 Exemple d'utilisation du drapeau clean session
- Figure 13 Exemple d'utilisation du drapeau Keep alive
- Figure 14 Format d'un paquet
- Figure 15 Format d'un paquet PUBLISH
- Figure 16 Format d'un paquet PUB [ACK/REC/REL/COMP]
- Figure 17 Format d'un paquet SUBSCRIBE
- Figure 18 Format d'un paquet SUBACK
- Figure 19 : Format d'un paquet UNSUBSCRIBE
- Figure 20 Format d'un paquet UNSUBACK
- Figure 21 Format d'un paquet PING [REQ/RESP]
- Figure 22 Format d'un paquet DISCONNECT
- Figure 23 Composants et caractéristiques L'Arduino UNO
- Figure 24 servomoteur 9 grammes
- Figure 25 Vue interne d'un servomoteur
- Figure 26 simulation d'un pont en H réaliser par Proteus à base des transistors
- Figure 27 Capteur infrarouge
- Figure 28 les ondes infrarouges
- Figure 29 Simulation Isis Proteus
- Figure 30 Interface utilisation
- Figure 31 Langage Arduino
- Figure 32 Interface d'utilisateur
- Figure 33 Fritzing
- Figure 34 Interface d'utilisateur
- Figure 35 schéma électrique du parking intelligent

Figure 36 montage reliant les différent composantes du parking intelligent

Figure 37 image de réalisation du parking intelligent

Figure 38 image du parking a l'état de repos

Figure 39 image de l'état de barrière o/f (pas de voiture)

Figure 40 image de ouverture de barrière pour entrer la voiture

Figure 41 image de l'état de barrière o/f (voiture entrer)

Figure 42 image de ouverture de barrière pour sortie la voiture

Figure 43 image de l'état de barrière o/f (voiture sortie)

Figure 44 image de ouverture de barrière pour sortie/entrer les voitures

Figure 45 image de l'état de barrière o/f (voiture entrer/sortie)

## **Liste des tableaux**

Tableau 1 Entête fixe du protocole MQTT

Tableau 2 Type des paquets MQTT

Tableau 3 Description du drapeau dans l'entête fixe du protocole MQTT

Tableau 4 Les valeurs du code de retour d'un paquet CONNACK

Tableau 5 Code de retour de paquet SUBACK

Tableau 6 Comportement de synchronisation des méthodes qui donnent lieu à des requêtes au broker

# Chapitre I

## Internet Des Objets

## **I.1. Introduction :**

L'Internet des objets (IOT) est une révolution technologique en informatique et en communication qui a mobilisé le domaine de l'industrie ces dernières années. Dans ce premier chapitre, nous allons survoler la notion d'Internet des objets. Comme premier titre, nous allons parler de la définition de cette nouvelle technologie qui a pu marquer son utilisation presque dans tous les domaines tel que l'industrie. Comme deuxième titre nous parlerons de son histoire depuis qu'elle était inventée jusqu'à nos jours. En troisième position, nous définissons l'IOT comme un système en relation avec le réseau, les données et les nouveaux services. Nous allons parler ensuite des objets connectés, en citons leurs avantages et inconvénients. Puis nous allons voir l'architecture des IOT avant de passer, finalement, aux différents domaines d'applications et les problèmes de l'IOT.

## **I.2. Que ce que l'Internet des objets :**

L'Internet des objets est un terme très vaste et riche il nous fait imaginer un monde entier qui est relié et peut se communiquer grâce à l'échange d'information entre ses objets. Leurs utilisateurs en utilisant Internet mais aussi d'autres réseaux de communication bien moins connus mais tous efficaces. C'est le monde de l'Internet des Objets. en 2020, l'Institut Gartner prévoit plus 50 milliards d'objets connectés sur le marché. Autant vous dire que nous sommes en train d'assister à une véritable révolution numérique qui va radicalement changer nos styles de vie [1].

Pour illustrer, prenons un exemple dans le domaine de l'habitat intelligent, aussi connu sous le nom de Smart Home. Imaginez que votre réfrigérateur devienne intelligent. Un réfrigérateur capable de vous dire en temps réel le type de denrées qu'il y a à l'intérieur et capable de passer commande pour vous quand vous avez besoin de vous réapprovisionner. Le réfrigérateur est un exemple. Mais le nombre d'objets qu'il est aujourd'hui possible de connecter est illimité. Le potentiel qu'offre l'Internet des Objets pour chacun d'entre nous.





Figure 1 : Internet des objets

### I.3. Historique sur l'Internet des objets

Le terme « Internet of Things » (en Français Internet des Objets) est né en 1999 au centre MIT (Massachusetts Institute of Technology), grâce à Kevin Ashton, un chercheur britannique, pionnier dans son domaine (IDO). Son équipe lança la promotion d'une connectivité ouverte de tous les objets en utilisant les étiquettes RFID (Radio Frequency Identification). Grâce à l'apparition du nouveau protocole IPv6, des secteurs comme l'aéronautique s'emparent rapidement du concept de l'Internet des objets, et participent aux recherches. Ce concept de l'Internet des Objets commence à connaître une popularité en 2007. On a envisagé alors de mettre en place un Internet des Objets Global, Ubiquitaire [2].



Figure 2 : L'utilisation de l'Internet des objets.

## I.4. L'IOT aujourd'hui

En 2003, la population mondiale s'élevait à environ 6,3 milliards d'individus et 500 millions d'appareils étaient connectés à Internet.<sup>3</sup> Le résultat de la division du nombre d'appareils par la population mondiale (0,08) montre qu'il y avait moins d'un appareil connecté par personne. Selon la définition de Cisco IBSG, l'IOT n'existait pas encore en 2003 car le nombre d'objets connectés était relativement faible. En outre, les appareils les plus répandus actuellement, et notamment les Smartphones, faisaient tout juste leur apparition sur le marché. En raison de l'explosion des Smartphones et des tablettes, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 (1,84 pour être exact) pour la première fois de l'histoire[3]

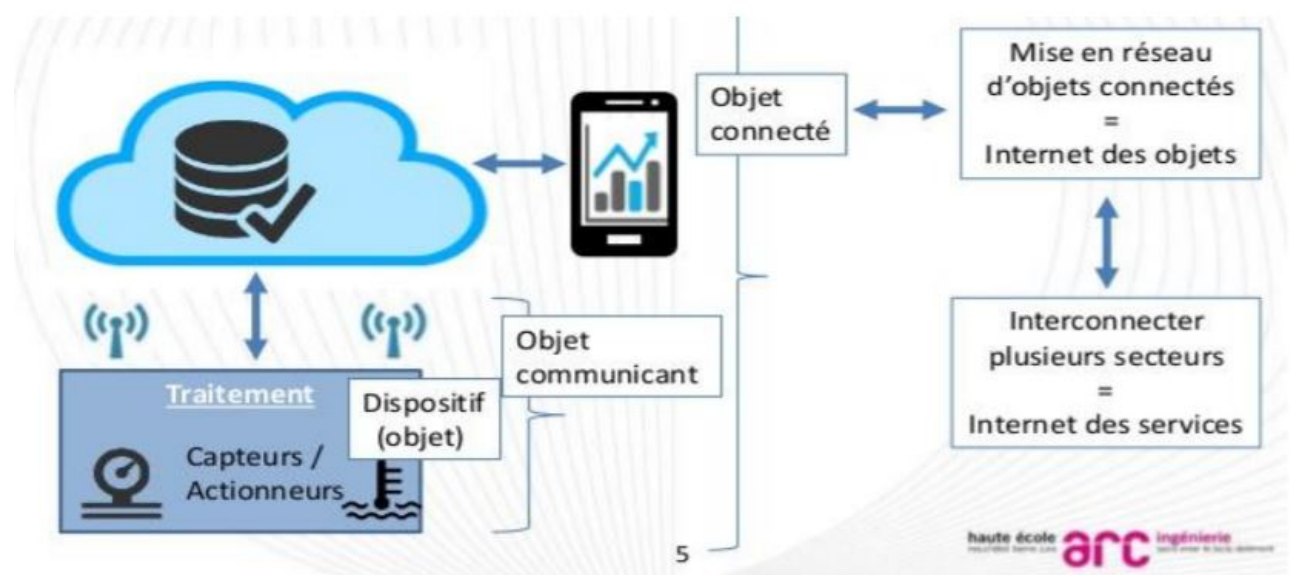


Figure 3 : L'Internet des objets aujourd'hui.

Aujourd'hui, derrière l'économie de la donnée, ce sont les objets connectés et l'industrie qui deviennent les principaux générateurs de contenu à analyser. Il est le fondateur de Objecte, une société de recherche, de formation et de conseil autour de l'Internet des objets. Il est également chargé de cours à l'ISAE-SUPAERO de Toulouse, au Collège de Bois de Boulogne, et formateur professionnel pour Data Science Institute. Il anime de nombreuses conférences chaque année, en Europe, au Canada et en Amérique du Sud. Il a écrit plusieurs livres sur le Big Data, le CRM, les places de marché, et son prochain ouvrage soulignera le potentiel combiné des objets, des données et des algorithmes. L'Internet des Objets, ou M2M à l'échelle industrielle, est en pleine expansion depuis quelques années : nous passerons de 15

milliards d'objets connectés dans le monde en 2012 à 80 milliards d'ici 2020 selon les prévisions. Entrez dans la révolution IOT du 21ème siècle basé sur des objets connectés, source de données tout au long de leur cycle de vie, qui se focalise sur la connectivité permanente et l'exploitation systématique de ces données grâce au Big-data. Ces données, relatives à l'état et l'environnement d'exploitation des objets en question, élargissent les possibilités de services du fabricant dans de nombreux domaines tels que l'observance, l'optimisation des flux, la maintenance prédictive et l'automatisation (mise à jour). Ces objets connectés trouvent une utilité dans tous les domaines. Les objets connectés produisent de grandes quantités de données dont le stockage et le traitement entrent dans le cadre de ce que l'on appelle les Big data. En logistique, il peut s'agir de capteurs qui servent à la traçabilité des biens pour la gestion des stocks et les acheminements. Dans le domaine de l'environnement, il est question de capteurs surveillant la qualité de l'air, la température, le niveau sonore, l'état d'un bâtiment, etc. En domotique, l'IOT recouvre tous les appareils électroménagers communicants, les capteurs (thermostat, détecteurs de fumée, de présence...), les compteurs intelligents et systèmes de sécurité connectés des appareils de type box domotique. Le phénomène IOT est également très visible dans le domaine de la santé et du bien-être avec le développement des montres connectées, des bracelets connectés et d'autres capteurs surveillant des constantes vitales. Selon diverses projections (cf. Cisco et le cabinet Gartner), le nombre d'objets connectés devrait largement augmenter au fil des ans.

## **I.5. Le futur de l'IOT**

Des chercheurs de l'EPFL (Ecole Polytechnique Fédérale de Lausanne) en Suisse ont annoncé en juillet 2016 avoir réussi à mettre au point un condensateur en graphène modulable qui permet l'utilisation des hautes fréquences pour connecter les objets c'est-à-dire que la transmission, des données et des informations circulant d'un appareil à l'autre dans un écosystème fermé comme l'ensemble des objets connectés d'une maison, sera plus rapide. Les données circulant mieux et plus rapidement vont permettre une utilisation plus simple, plus intuitive et plus efficace des appareils. D'autant que leur autonomie de fonctionnement sera largement plus importante[4]. Ce sont des chercheurs spécialisés dans la nano électricité qui ont eu élaboré un dispositif en graphène modulable. Ce dernier étant très petit et donc peu encombrant (sa taille= quelques centaines de micromètres), est miniaturisé. Ainsi, en plus d'améliorer la fréquence des transmissions et d'allonger la durée d'utilisation des batteries, il permettra aussi de fabriquer des appareils connectés encore plus compacts

[4]. Le graphène semble être aussi le matériau idéal dans le développement de la wear able technologie, des vêtements intelligents, et de la biotechnologie. Le graphène provoque donc un véritable défi pour l'avenir des communications sans fil et de l'IOT.

## **I.6. Internet des objets et son importance**

De nos jours les objets se transforment en objets intelligents capables de détecter, d'interpréter et de réagir à l'environnement grâce à la combinaison des IOT et les technologies émergentes. La liaison et la gestion de plusieurs objets est devenue facile grâce à l'Internet des objets qui est capable de relier un monde de réseau d'appareils intelligents, ou tout est interconnectés et a une entité numérique. De tous les jours les objets se transforment en objets intelligents capables de détecter, d'interpréter et de réagir à l'environnement grâce à la combinaison de l'Internet et technologies émergentes.

- Pour la technologie à disparaître de la conscience de l'utilisateur, l'Internet des objets exige une compréhension commune de la situation de ses utilisateurs et de leurs appareils.
- Des architectures logicielles et réseaux de communication omniprésents pour traiter et transmettre les informations contextuelles pour où il est pertinent.
- Les outils d'analyse dans l'Internet des objets qui visent un comportement autonome et intelligent.

Avec ces trois raisons fondamentales en place, la connectivité intelligente et sensible au contexte de calcul peut être accomplie. Depuis la production industrielle est l'un des plus grands facteurs économiques du monde l'un des principaux objectifs de ces initiatives est d'amener les paradigmes de la IOT aux usines leur permettant de faire face aux défis posés par populaires mégatendances. Les mégatendances avant tout pertinentes pour les usines sont la mondialisation, progresser l'évolution technologique, la dynamisation des cycles de vie des produits, la main-d'oeuvre vieillissante et la pénurie de ressources. Effets centraux sont l'accélération des cycles d'innovation et de la demande croissante des clients pour masse individualisée produit avec les plus hautes exigences de qualité. Dans le cadre de la production industrielle des projets et applications IOT se développent dans la fabrication, la chaîne d'approvisionnement, la surveillance et l'entretien. Une question importante dans tous les projets est sur la valeur, la prestation telle application peut apporter à l'utilisateur, au propriétaire ou à la société. L'Internet des objets regroupe tous les objets physiques communicants dotés d'une identité numérique unique. Selon l'Union

internationale des télécommunications, l'Internet des objets(IOT)est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution »[5]. D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel.

## **I.7. Internet des objets comme un système**

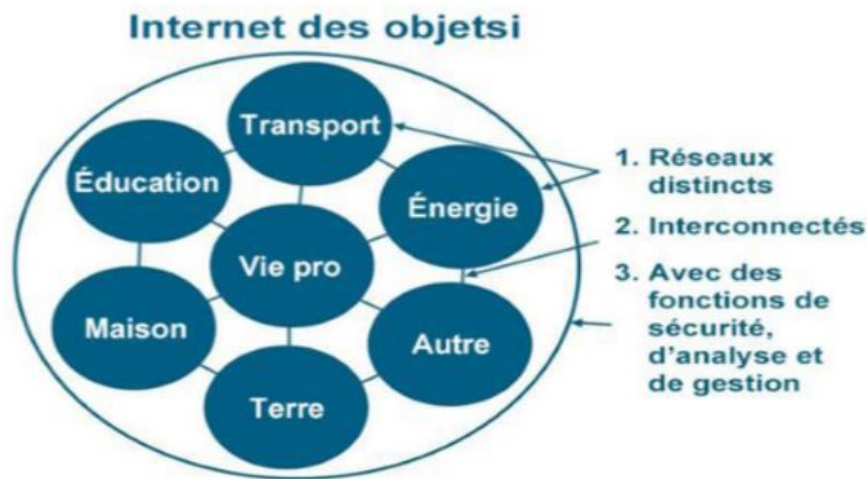
L'Internet des objets est en relation avec le réseau, les données et les nouveaux services :

### **I.7.1. L'Internet des objets comme un système des systèmes**

L'IOT désigne plutôt diverses solutions techniques (RFID, TCP/IP, technologies mobiles, etc.) qui permettent d'identifier des objets, de capter, stocker, traiter, et transférer des données dans les environnements physiques mais aussi entre des contextes physiques et des univers virtuels. L'enjeu majeur n'est pas tant d'inventer de nouvelles technologies que de perfectionner celles qui existent déjà, de les connecter, et de les intégrer. Nous listons ci-après les principales classes de solutions nécessaires au fonctionnement de l'IOT. Plutôt que d'en décrire tous les éléments, nous considérerons trois composantes considérées comme critiques : les solutions RFID, les solutions logicielles middleware et le réseau EPC global

### **I.7.2. Réseaux et Internet des Objets**

L'IOT se compose d'un ensemble hétérogène de réseaux qui permettent la communication de ces objets. Parmi les plus connus, les réseaux cellulaires des opérateurs télécoms qui permettent aux objets équipés d'une carte SIM M2M de remonter et envoyer les données. En plein émergence, les réseaux LPWA, avec notamment LoRa et Sigfox. Réseaux bas débit longue portée, ce sont des protocoles entièrement dédiés aux communications entre objets. Mais d'autres technologies sont aussi utilisées comme le Narrow Band ou le LTE-m[6].



Source : Cisco IBSG, avril 2011

Figure 4 : Réseaux et Internet des Objets

### **I.7.3. L'Internet des Objets et les données**

La petite pépite de l'Internet des Objets, c'est la donnée. Et surtout la capacité à capter une donnée brute, de température, de vibration, d'humidité... pour la transformer en information intelligente et exploitable. Ces milliards d'objets connectés vont créer un volume exponentiel de données qu'il faudra stocker, analyser, sécuriser et restituer pour des usages divers. Un enjeu de taille donc.

### **I.7.4. L'Internet des Objets et les nouveaux services**

La valeur ajoutée apportée par l'Internet des Objets est dans les nouveaux usages que cela va amener. Dans le secteur de l'industrie par exemple on peut désormais surveiller les machines à distance, faire de la maintenance prédictive des équipements, ou améliorer la traçabilité des produits. Chaque jour, les objets connectés vont générer des milliards d'informations qui permettront aux entreprises de créer de nouveaux services. L'Internet des Objets va littéralement transformer notre société. Cette affirmation pourra vous sembler utopiste pourtant il suffit de voir à quel point les choses ont évolué depuis l'arrivée de l'iPhone, il y a dix ans tout pile. Il ne tient qu'à nous de tirer parti du potentiel de l'Internet des Objets et d'en inventer les usages de demain.

## **I.8. Que ce qu'un objet connecté**

L'IOT repose avant tout sur les objets connectés. Un objet connecté a la capacité de capter une donnée et de l'envoyer, via le réseau Internet ou d'autres technologies, pour que celle-ci soit analysée et visualisée sur des tableaux de bord dédiés. Les objets connectés interagissent avec leur environnement par le biais de capteurs : température, vitesse, humidité, vibration... Dans l'Internet des Objets, un objet peut aussi bien être un véhicule, qu'une machine industrielle ou encore une place de parking.

### **I.8.1. Les avantages d'un objet connectés :**

Un objet connecté est la plupart du temps simplement composé

- D'un capteur (podomètre, thermomètre, capteur de fumée...)
- D'une petite mémoire permettant de stocker des préférences
- D'un processeur permettant de comparer des valeurs (atteinte d'un objectif de pas, atteinte de la consigne de température)
- D'une antenne de communication (Bluetooth, wifi, Sigfox ou autre) lui permettant ainsi d'envoyer les informations du capteur pour visualisation et de recevoir les préférences de l'utilisateur
- Parfois d'un mécanisme permettant d'effectuer une action simple (vibreur, sirène, déclencheur de chaudière...)

De ce fait, un objet connecté n'étant pas autonome, il ne nécessite pas d'embarquer de composants permettant une visualisation autonome, de stocker de nombreuses informations (historique) et de les traiter. Les avantages de cela sont multiples :

- Taille réduite de l'équipement nécessitant moins de composants
- Coût réduit à la production de l'équipement
- Cycles de vente raccourcis puisqu'il est possible de proposer rapidement à la vente le produit et de faire évoluer l'application avec de nouvelles fonctions ultérieurement (tout en les promettant dès la mise sur le marché).

## **I.8.2. Les inconvénients d'un objet connecté :**

- Les données du capteur transitent sur Internet et sont stockées sur les serveurs du fabricant (le tout de façon plus ou moins sécurisé).
- Certaines fonctions initialement promises ne sont jamais mises en place pour diverses raisons stratégiques et il est parfois nécessaire de réinvestir dans le modèle de 2e génération pour en bénéficier.
- En cas d'arrêt du service, l'objet devient inutile : le service peut être abandonné si non rentable, en cas de liquidation ou si changement de positionnement ou envie de générer nouvel investissement de la part du fabricant.

Enfin, un objet connecté se cantonne généralement à une seule fonction alors que dans de nombreuses occasions il est nécessaire d'agir sur plusieurs points pour obtenir un bénéfice réel et malheureusement dans de nombreux cas, les divers objets connectés ne permettent pas un fonctionnement en commun ce qui améliorerait grandement le confort, la sécurité et les économies d'énergie. Par exemple, en cas d'incendie, un système domotique allume les lumières et lève les volets afin de faciliter l'évacuation alors qu'un détecteur d'incendie connecté se contenterait d'envoyer une notification sur Smartphone.

Il existe certains services comme IFTTT où certaines box domotiques qui lient entre eux des objets connectés mais le système est alors entièrement dépendant sur la connexion Internet et les serveurs du fabricant.

Les objets connectés sont une grande avancée : ils permettent de faire connaître la domotique au grand public, de la démocratiser et même à certains de mettre en place des fonctions qui améliorent le quotidien. Le modèle des objets connectés est inspiré de l'évolution des systèmes informatiques des entreprises ces dernières années principalement hébergés par des services Cloud. Cependant, alors que les entreprises peuvent se permettre de vérifier et modifier leurs contrats garantissant ainsi une continuité de service, il n'en est pas de même du particulier qui est tributaire de CGV définis unilatéralement par le fabricant en termes juridiques peu accessibles et qui évoluent au gré de ses positionnements stratégiques[7].



## I.9. L'architecture de l'Internet des objets :

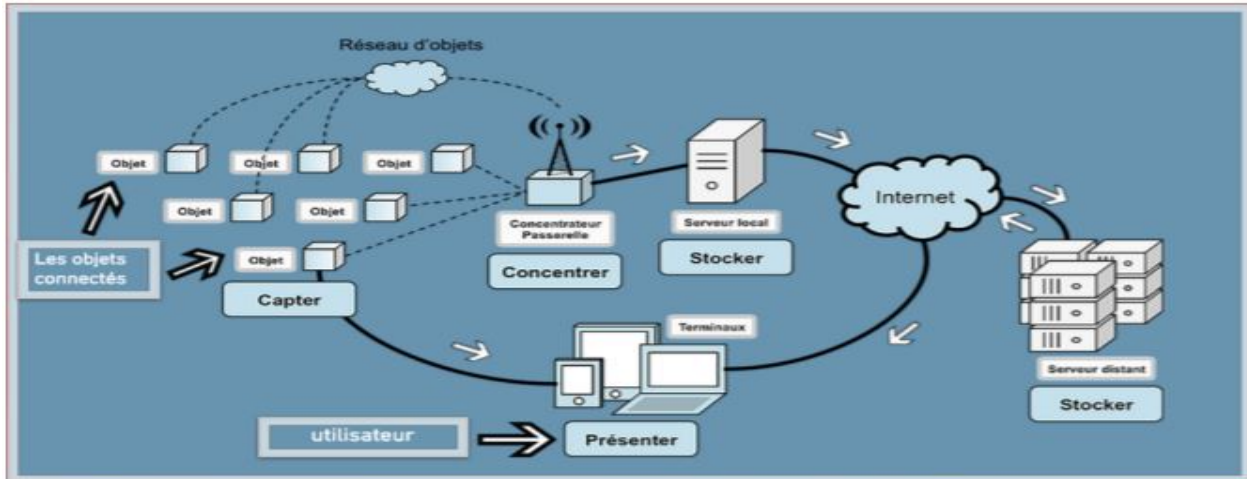


Figure 5 : L'architecture de l'Internet des objets

Précisons le rôle des différents processus présentés sur ce schéma :

- Capter désigne l'action de transformer une grandeur physique analogique en un signal numérique.
- Concentrer permet d'interfacer un réseau spécialisé d'objet à un réseau IP standard ou des dispositifs grand public.
- Stocker qualifie le fait d'agréger des données brutes, produites en temps réel, méta taguées, arrivant de façon non prédictible.
- Enfin, présenter indique la capacité de restituer les informations de façon compréhensible par l'Homme, tout en lui offrant un moyen d'agir et/ou d'interagir.

Deux autres processus n'apparaissent pas sur le schéma, car ils sont à la fois transverses et omniprésents :

- Le traitement des données est un processus qui peut intervenir à tous les niveaux de la chaîne, depuis la capture de l'information jusqu'à sa restitution. Une stratégie pertinente, et commune quand on parle d'Internet des objets, consiste à stocker l'information dans sa forme intégrale. On collecte de manière exhaustive, « big data », sans préjuger des traitements qu'on fera subir aux données. Cette stratégie est possible aujourd'hui grâce à des architectures distribuées type NoSQL, capables d'emmagasiner de grandes quantités d'information tout en offrant la possibilité de réaliser des traitements complexes en leur sein (Map/Reduce par exemple).

- La transmission des données est un processus qui intervient à tous les niveaux de la Chaîne. Deux réseaux, supports des transmissions, cohabitent généralement :
- Réseau local de concentration. On utilise alors des technologies comme ANT, Zig Bee, Z-Waves, NFC ou Bluetooth LE.
- Réseau WAN, permettant d'interconnecter les réseaux spécialisés et de les interfacer avec des fermes de serveur. On utilise alors Wifi, les réseaux cellulaires (GSM, UMTS, LTE) ou encore les connexions physiques standard (Ethernet, fibre optique). Ces réseaux sont généralement connectés à Internet[8].



Figure 6 : Domaines d'application de l'IOT.

## **I.10. Les applications de l'Internet des objets :**

Intel et son écosystème de partenaires aident les entreprises à utiliser l'Internet des objets (IOT) pour résoudre des défis de longue date spécifiques à chaque secteur d'activité.

Son portefeuille de solutions ouvertes et évolutives permet de développer rapidement des objets connectés, de collecter des données et d'obtenir des informations afin de réduire les coûts, d'améliorer la productivité et d'augmenter les recettes. Il existe plusieurs domaines d'application qui seront touchés par l'Internet émergent des choses. Les applications peuvent être classifiés en fonction du type de la disponibilité du réseau, la couverture, l'échelle, hétérogénéité, la répétabilité, la participation des utilisateurs et de l'impact[9].

IOT peut être utilisée dans plusieurs domaines :

### **✓ Le transport :**

Des voitures connectées ou autonomes aux systèmes de transport/logistique intelligents, l'IoT peut sauver des vies, réduire le trafic et minimiser l'impact des véhicules sur l'environnement.

### **✓ La sante :**

Des appareils médicaux wear ables aux tablettes de premiers secours en passant par les équipements chirurgicaux sophistiqués, l'IOT transforme les soins.

### **✓ Industrie :**

La technologie IOT permet aux usines d'améliorer l'efficacité de ses opérations, d'optimiser la production et d'améliorer la sécurité des employés.

### **✓ Énergie :**

L'Internet des objets permet aux innombrables appareils qui composent le réseau électrique de partager des informations en temps réel pour une distribution et une gestion plus efficace de l'énergie.

### **✓ Automobile :**

Reliée à l'IOT, la voiture transforme les données en informations exploitables, à la fois dans le véhicule et dans le monde qui l'entoure.

### ✓ **Maisons intelligentes :**

De la reconnaissance de votre voix à l'identification de la personne à la porte d'entrée, la technologie IOT donne vie au rêve d'une maison intelligente protégée.

### ✓ **Bâtiments intelligents :**

L'IoT résout les problèmes de l'augmentation des coûts énergétiques, de développement durable et de conformité au code en connectant, gérant et sécurisant les appareils qui collectent les données.

### ✓ **Vente au détail :**

L'IOT offre aux détaillants des possibilités illimitées d'améliorer l'efficacité des chaînes d'approvisionnement, de développer de nouveaux services et de proposer de nouvelles expériences aux clients.

### ✓ **Les applications dans la vie de tous les jours :**

L'Internet des objets c'est l'Internet que l'on porte, qui dépasse le seuil de votre domicile pour vous proposer des services quelques soit le lieu où vous vous trouvez. On va illustrer cette omniprésence en expliquant leurs implications, en prenant séparément l'espace professionnelle, l'espace privée et l'espace publique.

### ✓ **L'espace professionnel :**

Dans l'espace professionnel, on les retrouve dans les chaînes de logistiques ou de distribution avec notamment des puces dans les produits pour en connaître la traçabilité, et optimiser des activités comme la maintenance et la consommation d'énergie. C'est également le cas des balises communicantes en magasin (Beacon) qui guident un client dans ses achats. Pour ce qui est de l'usine, on parle "d'usine du futur" pour la France, c'est à dire une quatrième révolution industrielle qui se consacre aux objets connectés.

### ✓ **L'espace privé :**

Pour ce qui est de l'espace privé, "la diffusion des objets dans l'espace privé" comme "plus incertaine". Pour lui une majorité de personnes n'en voit pas l'intérêt et les délaisse rapidement car jugée désuet. Pour lui la clé du virage passe par des produits plus simples et plus attractif. On peut néanmoins voir qu'avec une population française qui possède à 60% Smartphone, la française savent gérer

majoritairement la technologie, et que la difficulté d'accès des objets connectés n'est peut-être pas une explication valable.

### ✓ **L'espace public :**

Dans l'espace public, on vise plutôt les villes connectées. Si des exemples existent au point de vue international, comme Songdo en Corée du Sud ou I-City en Malaisie, l'auteur oublie de parler du boulevard connecté de Nice. Présenté comme le premier boulevard connecté au monde, le boulevard Victor Hugo est équipé de nombreux capteurs afin de vous proposer un maximum de services. Cela va des capteurs de stationnements qui vous informent en temps réel des places de parking disponible, ou encore les éclairages intelligents qui visent à réduire l'empreinte énergétique de la ville. C'est également le cas pour la gestion des déchets. Le but est de faire de la ville un environnement plus sain et plus simple [10].

## **I.11. L'importance des services**

### **I.11.1. 80 % de la valeur ajoutée des TIC**

Le constat est simple. L'objet connecté en utilisant la connexion numérique propose plus de service que l'objet classique. Pour l'auteur, cela fait des fournisseurs de services les grands gagnants du tournant de l'Internet des objets, en rajoutant actuellement 80% de la valeur ajoutée générée dans l'OCDE par TIC sont produites par les services[10].

### **I.11.2. Des plateformes de données par communautés**

Ces services passent également par l'importance d'une communauté des périphériques connectés. Ces communautés seront chargées de gérer les échanges de données et de réunir les acteurs d'une communauté d'objet connecté. C'est d'ailleurs déjà le cas, avec certaines plateformes qui sont prévus, comme l'Androïde Wear ou encore Riot, développé par les français et les allemands. L'un des dangers tient d'ailleurs pour l'auteur à ces plateformes, car étant donné que des géants du numérique comme Google ou Apple ont développé la leur, tout acteurs qui n'aura pas opté sur cette plateforme se verra marginaliser[10].

### **I.11.3. La nécessité de l'identification des objets**

Dans un univers de Big Data, où l'on étudie les besoins des utilisateurs par leurs traces numériques, on peut également aller encore plus loin dans la connaissance du client, et connaître ses habitudes pour lui proposer les solutions les plus à même de le satisfaire. C'est d'ailleurs l'un des enjeux de la bataille des standards de nommage des objets, entre le système GS1 qui propose une technologie pour remplacer les codes-barres par des étiquettes RFID et Google qui lui a lancé un projet nommé « Physical Web » afin d'associer une adresse URL à un objet[10].

### **I.12. Les problèmes de l'Internet des objets**

Comme toutes les autres technologies l'Internet des objets a des difficultés d'application et de réalisation. Les problèmes majeurs sont listés ci-dessous :

- ✓ Personne ne portera 50 appareils. Les appareils sont de plus en plus spécifiques : ils sont conçus pour faire une chose et la faire vraiment bien, que ce soit mesurer le sommeil, sa vitesse d'alimentation ou la santé de son genou. Au nom de la simplicité, les entreprises cherchent trop à résoudre un seul problème, un seul cas d'utilisation. Pas sûr que cela dessine un marché.
- ✓ Plus d'inférence moins de détection. Avons-nous vraiment besoin d'un bracelet qui détecte le niveau d'UV auquel nous sommes exposés ? L'inférence, qui prend en charge le contexte est plus nécessaire que la détection permanente. Dans l'Internet des objets, les objets capables de comprendre l'information par inférence plutôt que par une détection permanente ont plus de chance de survivre que les autres.
- ✓ Le problème du découpage des données. Dans le monde de l'Internet des objets, trop souvent, les interfaces et la façon dont sont produites les données découragent l'échange. Le manque d'interopérabilité et la guerre des formats est la règle. Nombreux sont ceux qui veulent créer le système d'exploitation unique de l'IOT, mais en travaillant seulement depuis leurs propres appareils, regrettait le PDG de Samsung.
- ✓ Le contexte est la clef. Si les données ne changent pas notre comportement, alors il ne sert à rien de les recueillir. V1bes ou les solutions sociométriques mesurent le niveau de stress, mais s'ils ne captent pas le contexte qui génère ce stress, alors à quoi le rattacher, comment l'interpréter, le comprendre ?

L'Internet des Objets, ou M2M à l'échelle industrielle, est en pleine expansion depuis quelques années : nous passerons de 15 milliards d'objets connectés dans le monde en 2012 à 80 milliards d'ici 2020 selon les prévisions.

Entrez dans la révolution IOT du 21ème siècle basé sur des objets connectés, source de données tout au long de leur cycle de vie, qui se focalise sur la connectivité permanente et l'exploitation systématique de ces données grâce au Big Data. Ces données, relatives à l'état et l'environnement d'exploitation des objets en question, élargissent les possibilités de services du fabricant dans de nombreux domaines tels que l'observance, l'optimisation des flux, la maintenance prédictive et l'automatisation[11].

# Chapitre II

# Protocol MQTT



## II.1. Introduction

MQTT (Message Queue Telemetry Transport) est un protocole M2M (Machine-to-Machine) open source [12]. C'est un protocole de messagerie basé sur le publish/subscribe à la fois extrêmement simple et léger idéal pour l'Internet des objets. Il a été inventé par Andy StanfordClark (IBM) et Arlen Nipper (Arcom, maintenant Cirrus Link) en 1999, lorsque leur cas d'utilisation était de créer un protocole avec les objectifs suivants : [13]

- 1- Simple à mettre en œuvre.
- 2- Fournir une qualité de service de livraison de données.
- 3- Léger et faible consommation de la bande passante.

Par la suite, la version 3.1 a été libérée de droits en 2010, et le protocole est standardisé par l'OASIS en 2014 dans sa version 3.1.1. : [14]. Il devient un standard ISO (ISO/IEC 20922) en 2016[15].

Aujourd'hui, avec l'essor de l'IOT, MQTT suscite un intérêt grandissant, dont on peut se rendre compte en observant l'évolution du nombre de recherches pour MQTT sur Google Trends

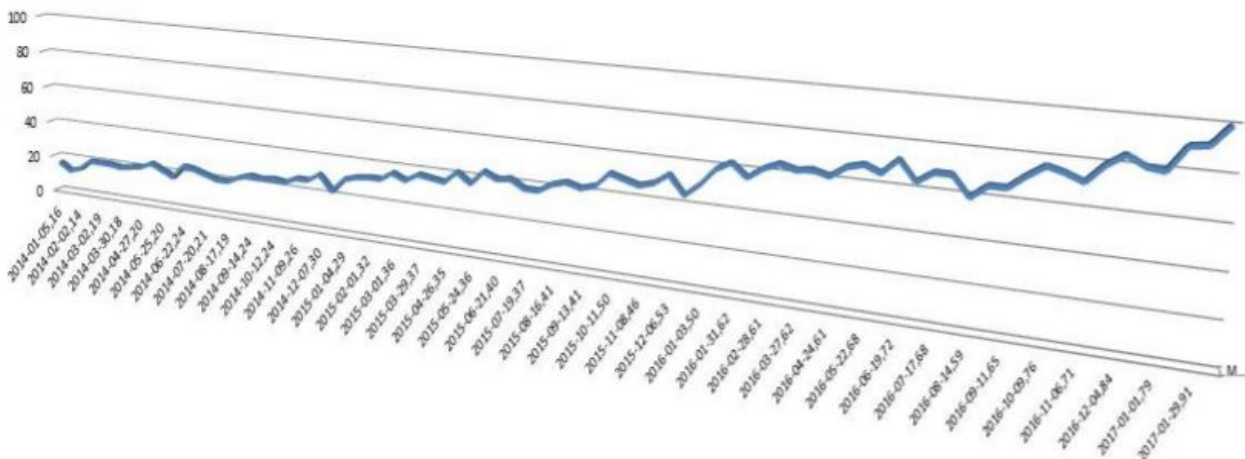


Figure 7 Google Trends pour MQTT (18/02/2017)

Les qualités de MQTT font de lui un excellent candidat pour les communications au sein de l'Internet des Objets [16], mais il a également été utilisé avec succès par Facebook pour être au cœur de leur système de messagerie à cause de sa performance[17], comme il est utilisé par amazon web services dans ses passerelles pour permettre la diffusion des données à plusieurs abonnés sur un sujet donné [18]. La norme MQTT décrit le comportement attendu des brokers et clients MQTT, ainsi que les détails du format binaire employé pour les échanges.

La suite de ce chapitre synthétisera l'essentiel de cette norme afin de mieux l'exploité ce protocole dans notre projet. Nous expliquerons en détail le fonctionnement du protocole MQTT. La partie suivante sera consacrée au format binaire défini par la norme pour les paquets MQTT.

## II.2. Mode de fonctionnement

MQTT est un protocole de publication et de souscription de message. Les clients ne communiquent pas directement entre eux, ils publient des messages sur un broker (appelé aussi courtier), les messages sont composés d'un contenu et d'un sujet (topic) [19]. Le broker garde en mémoire le dernier message pour chaque sujet. Les clients qui sont intéressés par les messages d'un sujet peuvent les récupérer en se connectant au broker. L'avantage de cette solution est qu'elle permet à plusieurs clients de communiquer même s'ils ne sont jamais connectés en même temps au broker.

Ce patternne consiste à organiser les messages par sujets (topics) et à gérer leur distribution selon le principe d'abonnement. Ainsi, toute application peut publier ses messages sur les topics de son choix, tandis que les applications intéressées par les messages d'un topic donné peuvent s'abonner pour recevoir tous les nouveaux messages publiés sur ce topic.

### II.2.1 Topologies MQTT

Le standard ne définit pas de topologie particulière d'utilisation, la plus simple étant celle avec un seul broker et un certain nombre de clients. D'autres organisations restent évidemment possible, avec en particulier les systèmes à base de bridge dans lesquels plusieurs brokers sont utilisés, chacun étant client des autres brokers

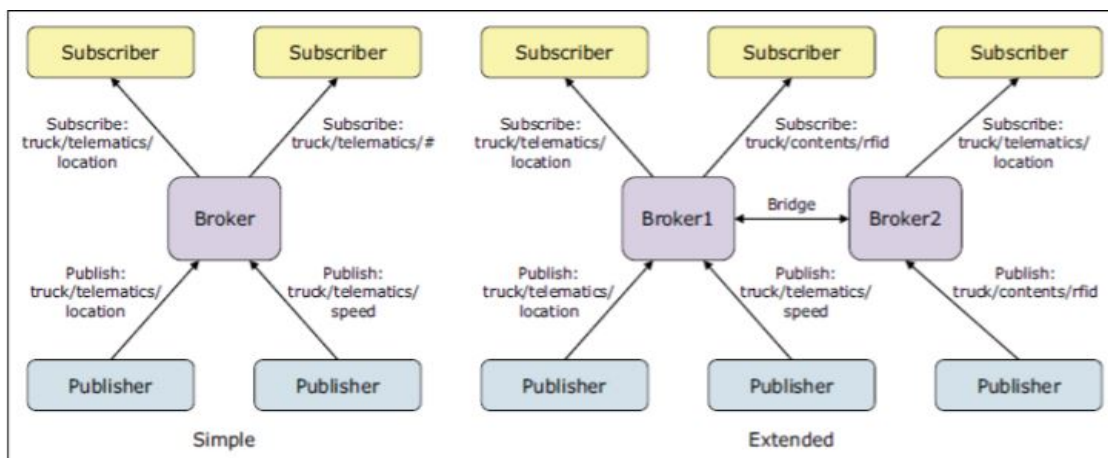


Figure 8 Exemples de topologies MQTT [20]

## II.2.2 Fonctionnement

### II.2.2.1 Connexion et Déconnexion

MQTT se sert de connexions persistantes entre les clients et le broker, et exploite pour cela les protocoles de réseau garantissant un bon niveau de fiabilité comme TCP. Avant de pouvoir envoyer des commandes de contrôle, un client doit au préalable "enregistrer auprès du broker, ce qui se fait avec la commande **CONNECT**.

Divers **paramètres** de connexion peuvent alors être échangés, comme les identifiants du client ou encore le mode de persistance souhaité. Le broker doit confirmer au client que son **inscription** a bien été prise en compte, soit indiquer qu'une erreur est survenue en renvoyant un **CONNACK** accompagné d'un code de retour.

Il existe une commande **PINGREQ** permettant de faire savoir au broker que le client est toujours actif, le broker répondra de son côté avec un **PINGRESP** pour indiquer au client que la connexion est toujours active.

Lorsque le client veut se déconnecter, il envoie au préalable une commande **DISCONNECT** au broker pour lui faire part de son intention. Dans le cas contraire, le broker considérera la déconnexion comme anormale et agira en conséquence, [21] en envoyant le message de volonté **WILL** au nom du client à tous les abonnés. Un message que le client peut spécifier et conservera l'indicateur dans la charge utile de message **CONNECT**.

### II.2.2.2 Abonnements et Publications

Chaque message publié est nécessairement associé à un sujet, ce qui permet sa distribution aux abonnés. Les sujets peuvent être organisés en hiérarchie arborescente, ainsi les abonnements peuvent porter sur des motifs de filtrage que nous détaillerons plus loin.

La gestion des abonnements est très simple et consiste en trois commandes essentielles :

## 1- SUBSCRIBE

Permet à un client de s'abonner à un topic , une fois abonné il recevra par la suite toutes les publications concernant ce sujet. Un abonnement définit également un niveau de qualité de service, dont il sera question plus bas. La bonne réception de cette commande est confirmée par le broker par un **SUBACK** portant le même identifiant de paquet.

## 2- UNSUBSCRIBE

Donne la possibilité d'annuler un abonnement, et ainsi ne plus recevoir les publications ultérieures. La bonne réception de cette commande est confirmée par le broker par un **UNSUBACK** portant le même identifiant de paquet.

## 3- PUBLISH

Initié par un client, permet de publier un message qui sera transmis par le broker aux abonnés éventuels. La même commande sera envoyée par le broker aux abonnés pour délivrer le message.

Si la qualité de service requise est supérieure à zéro, des messages seront échangés pour confirmer la prise en charge de la publication

### II.2.3 Topic et motifs de filtrage

Un sujet est une chaîne UTF-8, qui est utilisé par le broker pour filtrer les messages pour chaque client connecté. Un sujet est constitué d'un ou plusieurs niveaux de sujet. Chaque niveau de sujet est séparé par une barre oblique. A noter que chaque sujet doit avoir au moins 1 caractères pour être valide et il peut également contenir des espaces. Aussi un sujet est sensible à la casse, ce qui rend **myhome / température** et **MyHome / Température** deux sujets individuels. En outre, la barre oblique seule est un sujet valable. Un client peut souscrire à plus de sujets à la fois comme illustré dans la figure qui suit:

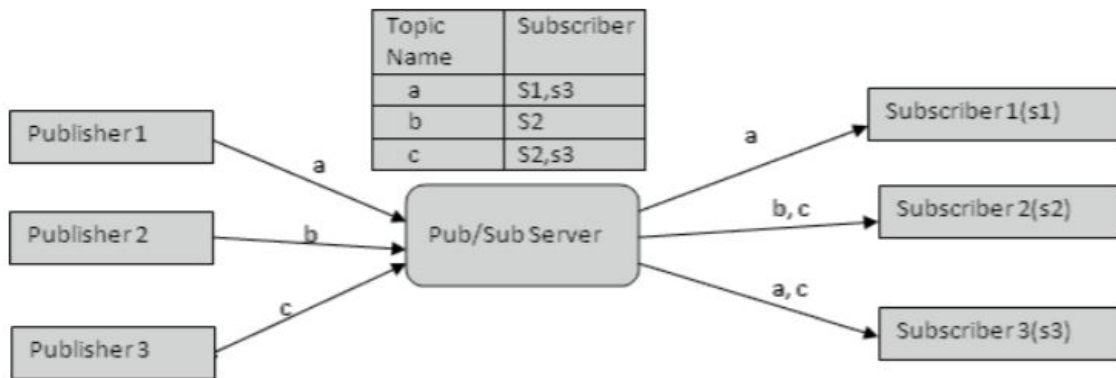


Figure 9 Souscription au topic

Afin de proposer un système de filtrage efficace sur les sujets, il est possible de définir une arborescence à l'aide du séparateur /.

Deux jokers sont réservés pour représenter un et plusieurs niveaux d'arborescence:

- + représente un niveau d'arborescence, ainsi T1/T2/T3 peut être mis en correspondance avec divers filtres tels que T1/T2/+, T1/+/T3 ou T1/+/+.
- # représente autant de niveaux que possible, et ne peut être utilisé qu'à la fin d'un motif de filtrage ; ainsi T1/# permettra de filtrer tous les sujets dont le premier niveau est T1, et # permet de recevoir tous les sujets publiés par le broker (A l'exception des sujets spéciaux commençant par\$).

## II.2.4 Qualité de service [22]

Trois niveaux de qualité de service (QoS) sont définis pour la publication des messages:

**QoS 0:** Livraison une fois maximum.

Les messages sont envoyés en fonction des capacités maximum du réseau TCP/IP sousjacent. Aucune réponse n'est attendue. Aucune sémantique de relance n'est définie dans le protocole. Par conséquent, le message ne parvient pas du tout ou une seule fois au broker de destination. Le niveau QoS 0 est également connu comme fire and forget.

**QoS 1: Livraison au moins une fois.**

L'arrivée d'un message QoS 1 au broker est reconnue. En cas d'échec identifiable de la liaison ou de l'unité d'envoi, ou bien après une certaine période de temps sans réception du message d'accusé de réception, l'expéditeur envoie à nouveau une copie du message. Par conséquent, le message est sûr d'arriver, mais il peut le faire plusieurs fois.

**QoS 2: Livraison exactement une fois.**

Pour le niveau QoS 2, d'autres flux de protocoles sont utilisés au-delà de QoS 1 pour que des messages ne soient pas envoyés en double à l'application de réception. Il s'agit du niveau de service le plus élevé qui sert lorsque des messages en double ne sont pas appropriés. Il existe évidemment des conséquences en matière de trafic réseau, mais cet impact est souvent acceptable sachant l'importance du contenu du message.

La qualité de service peut être sélectionnée message par message, ce qui permet la publication des messages d'importance mineure avec le niveau QoS 0 et la distribution des messages plus importants avec QoS 2.

Ces niveaux sont mis en œuvre par des échanges supplémentaires entre l'expéditeur et le récepteur, et plus la qualité demandée est élevée, plus il faudra d'échanges pour valider une publication comme illustré dans la figure. Pour tous les niveaux supérieurs à zéro, un identifiant est associé au message pour permettre son suivi, MQTT prévoyant la possibilité d'avoir au plus 65535 messages en attente (l'identifiant de message tenant sur 16 bits)

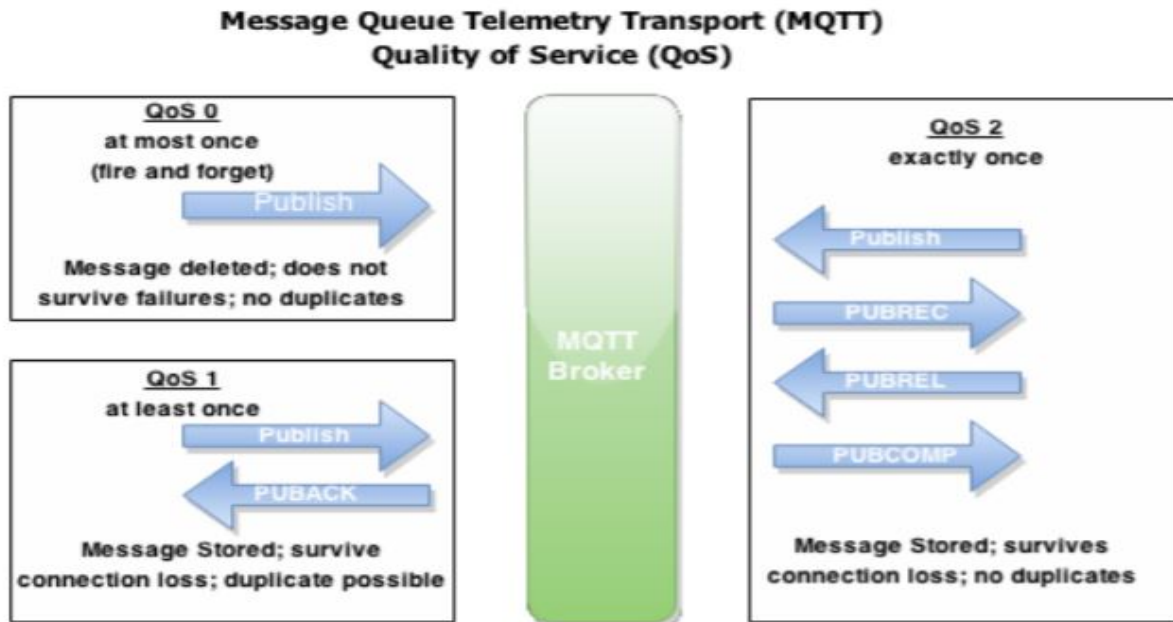


Figure 10 Niveaux de Qualité de Service MQTT [23]

## II.2.5 Dernière volonté et testament (Last Will & Testament)[24]

Permet de notifier une déconnexion anormale, ils ont défini lors de la connexion.

- Dernière volonté (LastWill): Dans le protocole MQTT parce qu'il n'y a pas de file d'attente le message n'est pas mis en cache. Pour éviter ce problème, les utilisateurs peuvent définir une (retenue dans le drapeau) lors de la publication. Ce drapeau invite le broker pour stocker le message et à l'offrir aux abonnés futurs dès leur inscription comme dernière publication connue pour un sujet donné.
- Testament: Est une caractéristique importante de MQTT. Un client indique au broker au cours de la configuration de la connexion que: «En cas de ma mort, s'il vous plaît envoyer le message suivant pour ce sujet à mes abonnés». Si le broker détecte alors que ce client est plus en vie, il le fait comme il a été dit et publie le message pour les abonnés pour le sujet désiré.

## II.2.6 Sécurité [25]

Trois concepts sont fondamentaux pour la sécurité MQTT : identification, authentification et autorisation.

- **L'identification:** consiste à nommer le broker et le client auquel on donne des droits d'accès.
- **L'authentification:** cherche à prouver mutuellement l'identité du client et du broker.
- **L'autorisation:** consiste à gérer les droits du client. Le broker MQTT s'identifie auprès du client avec son adresse IP et son certificat numérique.

Le client MQTT utilise le protocole SSL pour authentifier le certificat envoyé par le broker. Un client authentifie un broker à l'aide du protocole SSL.

Un broker MQTT authentifie un client à l'aide du protocole SSL, d'un mot de passe, ou des deux.

L'autorisation ne fait pas partie du protocole MQTT. Elle est fournie par les brokers MQTT. Ce qui est autorisé ou non dépend de ce que fait le broker.

## II.3 Description informelle du protocole

Dans cette section, nous définissons le format binaire des paquets spécifié par la norme MQTT.

### II.3.1 Représentations de données

#### II.3.1.1 Bits

Les bits dans un octet sont étiquetés de 7 à 0. Le bit numéro 7 est le bit le plus important, le bit le moins significatif est affecté au bit 0.

#### II.3.1.2 Valeurs de données entières

Les valeurs de données entières sont 16 bits: l'octet d'ordre supérieur précède l'octet d'ordre inférieur. Cela signifie qu'un mot de 16 bits est présenté sur le réseau comme octet le plus significatif (MSB), suivi de l'octet le moins important (LSB).



### II.3.1.3 chaînes codées enUTF-8

Les chaînes de caractères doivent toutes être encodées en utf8, et préfixées par leur longueur sur deux octets, ces chaînes sont donc limitées à une longueur de  $2^{16} - 1 = 65535$  octets.

### II.3.2 Structure d'un paquet de contrôle MQTT

Le protocole MQTT fonctionne en échangeant une série de paquets de contrôle MQTT d'une manière définie. Cette section décrit le format de ces paquets. Un paquet de contrôle MQTT comprend jusqu'à trois parties, toujours dans l'ordre suivant: En-tête fixe, En-tête variable, Charge utile,

#### II.3.2.1 Entête fixe

Tous les packets MQTT commencent par un format d'entête fixe, contenant le code de commande MQTT (sur 4 bits), un drapeau de 4 bits et la remaining length.

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2	Remaining Length							

Tableau 1 Entête fixe du protocole MQTT.

### II.3.2.1.1 Control Packet type:

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Client request to connect to Server
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server/Server to Client	Publish message
PUBACK	4	Client to Server/Server to Client	Publish acknowledgment
PUBREC	5	Client to Server/Server to Client	Publish received (assured delivery part 1)
PUBREL	6	Client to Server/Server to Client	Publish release (assured delivery part 2)
PUBCOMP	7	Client to Server/Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	8	Client to Server	Client subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgment
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server	Client is disconnecting
Reserved	15	Forbidden	Reserved

Tableau 2 Type des paquets MQTT

### II.3.2.1.2 Remaining length:

Une particularité du champ **remaining length** est qu'il est de longueur variable, et peut utiliser de 1 à 4 octets en fonction de la taille encodée c'est-à-dire : La taille maximale d'un paquet MQTT est donc de 256 Mo.

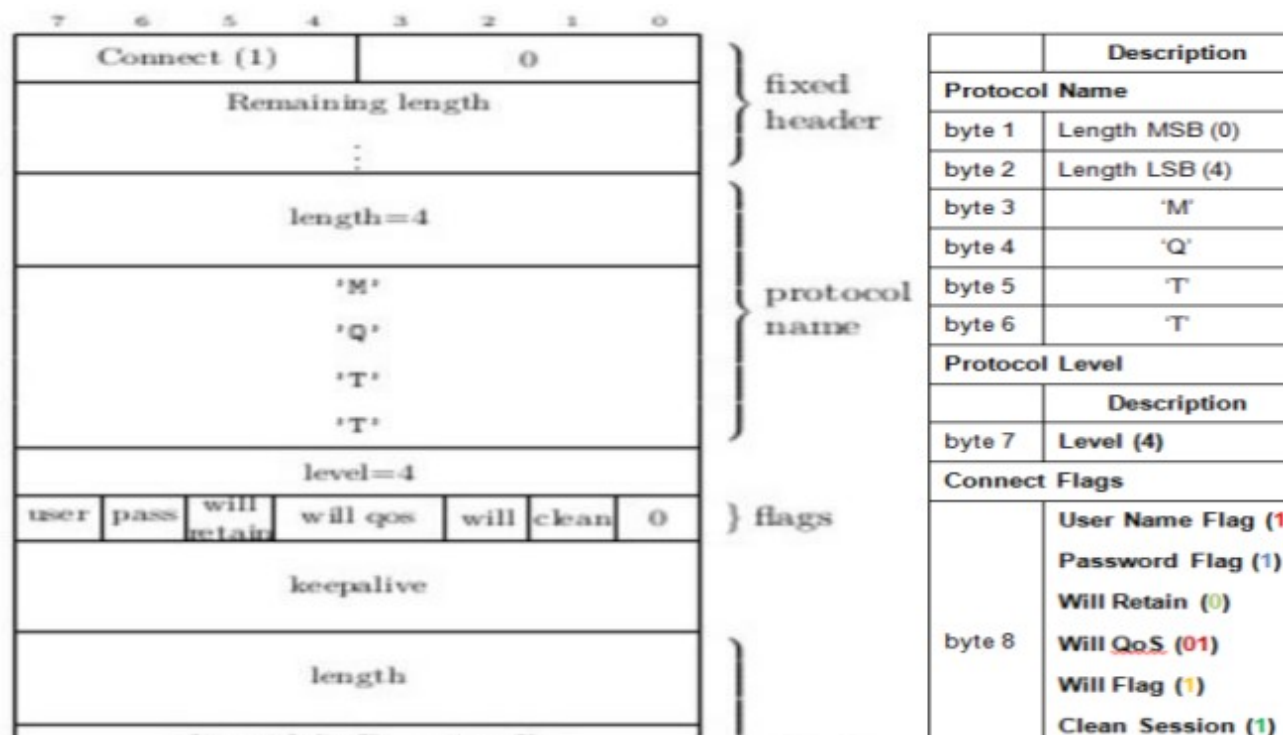
### II.3.2.1.3 Flag:

Les bits restants [3-0] de l'octet 1 de l'en-tête fixe contiennent des drapeaux spécifiques à chaque type de paquet de contrôle MQTT. Lorsqu'un bit de drapeau est marqué comme «Réservé», il est réservé pour une utilisation future.

Control Packet	Fixed header flags	Bit 3	Bit 2
CONNECT	Reserved	0	0
CONNACK	Reserved	0	0
PUBLISH	Used in MQTT 3.1.1	DUP <sup>1</sup>	QoS <sup>2</sup>
PUBACK	Reserved	0	0
PUBREC	Reserved	0	0
PUBREL	Reserved	0	0
PUBCOMP	Reserved	0	0
SUBSCRIBE	Reserved	0	0
SUBACK	Reserved	0	0
UNSUBSCRIBE	Reserved	0	0
UNSUBACK	Reserved	0	0

Tableau 3 Description du drapeau dans l'entête fixe du protocole MQTT

### II.3.2.2 CONNECT



l'entête de CONNECT dans la version 3.1.1 se présente sous la forme suivante :

Figure 11 Format d'un paquet CONNECT

Tout d'abord le nom du protocole (préfixé par sa taille), puis le niveau du protocole, des drapeaux indiquant la présence ou non de paramètres optionnels ainsi que la persistance de la session (clean indique que la session est non persistante, et constitue en même temps une directive pour effacer une éventuelle session existante), et enfin la valeur du keepalive

### II.3.2.2.1 Level

La valeur non signée de 8 bits qui représente **le niveau de révision du protocole** utilisé par le Client. La valeur du champ Level de protocole pour la version 3.1.1 du protocole est **4 (0x04)**.broker DOIT répondre au paquet CONNECT avec un code de retour CONNACK 0x01 (niveau de protocole inacceptable), puis déconnecter le client si le niveau de protocole n'est pas pris en charge par le broker

#### II.3.2.2.2 Connect Flags

L'octet Connect Flags contient un certain nombre de paramètres spécifiant le comportement de la connexion MQTT. Il indique également la présence ou l'absence de champs dans la charge utile.

- Le broker doit valider que le drapeau Reserved dans le paquet de Control CONNECT est mis à zéro et déconnecter le client s'il n'est pas zéro.

#### II.3.2.2.2.1 Nom de l'utilisateur Position: bit 7 de l'octet Connect Flags.

- Si le drapeau de nom d'utilisateur est défini sur 0, un nom d'utilisateur NE DOIT PAS être présent dans la charge utile.

- Si le drapeau de nom d'utilisateur est défini sur 1, un nom d'utilisateur DOIT être présent dans la charge utile

#### II.3.2.2.2.2 Indicateur de mot de passe Position: bit 6 de l'octet Connect Flags.

- Si le drapeau du mot de passe est défini sur 0, un mot de passe NE DOIT PAS être présent dans la charge utile.

- Si le drapeau de mot de passe est défini sur 1, un mot de passe DOIT être présent dans la charge utile. - Si le drapeau de nom d'utilisateur est défini sur 0, l'indicateur de mot de passe DOIT être réglé sur 0.

**II.3.2.2.2.3 Will Retain Position:** bit 5 de l'octet Connect Flags. Ce bit spécifie si le message de volonté doit être conservé lorsqu'il est publié.

- Si le drapeau Will est défini sur 0, alors le drapeau Will Retain doit être réglé sur 0.
- Si le drapeau de volonté WILL est défini sur 1:
  - Si Will Retain est défini sur 0, le broker DOIT publier le message Will comme un message non retenu.
  - Si Will Retain est défini sur 1, le broker DOIT publier le message Will comme message conservé.

#### **II.3.2.2.2.4 Will QoS**

**Position:** bits 4 et 3 de l'octet Connect Flags

Ces deux bits spécifient le niveau QoS à utiliser lors de la publication du message Will. Si le drapeau de Will est défini sur 0, alors la QoS de will DOIT être réglée sur 0 (0x00).

Si le drapeau Will est défini sur 1, la valeur de Will QoS peut être 0 (0x00), 1 (0x01) ou 2 (0x02). Il NE DOIT PAS être 3.

#### **II.3.2.2.2.5 Will (volonté)**

**Position:** bit 2 de l'octet Connect Flags. Si le drapeau Will est défini sur 1, cela indique que, si la demande de connexion est acceptée, un message de volonté DOIT être stocké sur le broker et associé à la connexion réseau. Le message de volonté DOIT être publié lorsque la connexion réseau est fermée par la suite, sauf si le message de volonté a été supprimé par le broker lors de la réception d'un paquet DISCONNECT

Les situations dans lesquelles le message de volonté est publié comprennent, sans s'y limiter:

- Une erreur d'E / S ou une panne de réseau détectée par le broker.
- Le Client ne parvient pas à communiquer dans le temps de KeepAlive.
- Le client ferme la connexion réseau sans d'abord envoyer un paquet DISCONNECT.
- Le broker ferme la connexion réseau en raison d'une erreur de protocole.

- Si le drapeau de volonté est défini sur 1, les champs Will QoS et Keep Retain dans les Flags Connect seront utilisés par le broker et les champs Will Topic et Will Message DOIVENT être présents dans la charge utile.

Le message de volonté DOIT être supprimé de l'état de session stocké dans le broker une fois qu'il a été publié ou le broker a reçu un paquet DISCONNECT du client

- Si le drapeau de volonté est défini sur 0, les champs Will QoS et Keep Retain dans les drapeaux de connexion DOIVENT être mis à zéro et les champs Will Topic et Will Message NE DOIVENT PAS être présents dans la charge utile. - Si le drapeau de volonté est défini sur 0, un message de volonté NE DOIT PAS être publié lorsque cette connexion réseau se termine.

- Le broker DEVRAIT publier Will Messages rapidement. Dans le cas d'un arrêt ou d'une panne du broker, le broker PEUT différer la publication des messages de volonté jusqu'à un redémarrage ultérieur. Si cela se produit, il peut y avoir un délai entre le moment où le broker a éprouvé une panne et un message de volonté en cours de publication.

#### **II.3.2.2.2.6 clean Session**

**Position:** bit 1 de l'octet Connect Flags.

Ce bit spécifie la gestion de l'état Session. Le drapeau clean session indique au broker, que le client souhaite établir une session persistante ou non. Le client et le broker peuvent stocker l'état de session pour permettre une messagerie fiable de

continuer à travers une séquence de connexions réseau. Ce bit est utilisé pour contrôler la durée de vie de l'état Session.

- Si Clean Session est défini sur 0, le broker DOIT reprendre les communications avec le client en fonction de l'état de la session en cours (identifié par l'identifiant du client). S'il n'y a pas de session associée à l'identifiant client, le broker DOIT créer une nouvelle session. Le client et le broker DOIVENT enregistrer la session après que le client et le broker soient déconnectés. Après la déconnexion d'une session qui avait CleanSession réglé sur 0, le broker DOIT stocker plus de messages QoS 1 et QoS 2 qui correspondent aux souscriptions que le client avait au moment de la déconnexion dans le cadre de l'état Session. Il PEUT également stocker des messages QoS 0 répondant aux mêmes critères.

- Si Clean Session est défini sur 1, le Client et le Broker DOIVENT renoncer à toute Session précédente et en créer une nouvelle. Cette session dure aussi longtemps que la connexion réseau. Les données d'état associées à cette session NE DOIVENT PAS être réutilisées lors d'une session ultérieure

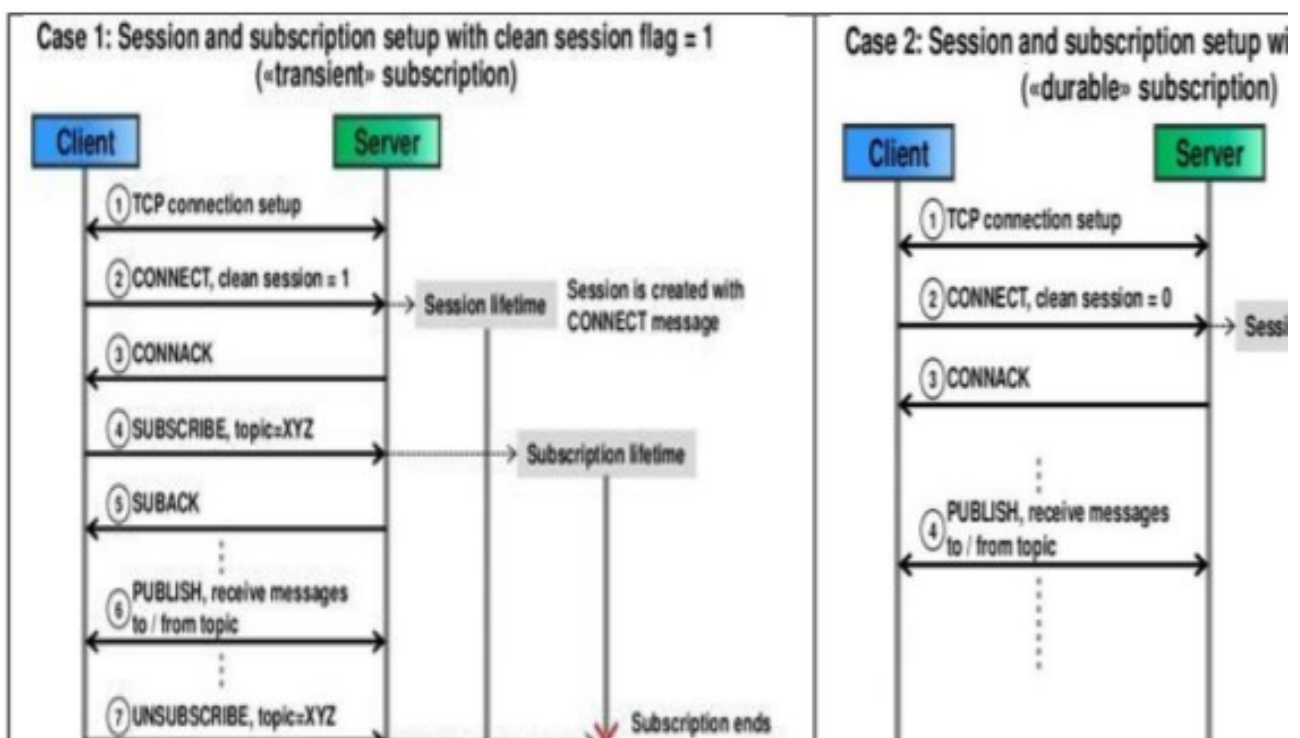


Figure 12 Exemple d'utilisation du drapeau clean session [26]

### II.3.2.2.3 Keep Alive

Keep Alive est un intervalle de temps mesuré en secondes. Exprimé en tant que mot de 16 bits, c'est l'intervalle de temps maximum qui peut s'écouler entre le point auquel le client finit de transmettre un paquet de contrôle et le point qu'il commence à envoyer le prochain. Il revient au client de s'assurer que l'intervalle entre les paquets de contrôle envoyés ne dépasse pas la valeur Keep Alive. En l'absence d'envoi d'autres paquets de contrôle, le client DOIT envoyer un paquet PINGREQ.

Le client peut envoyer PINGREQ à tout moment, indépendamment de la valeur Keep Alive, et utiliser le PINGRESP pour déterminer si le réseau et le broker fonctionnent.

Si la valeur Keep Alive est non nulle et que le broker ne reçoit pas un paquet de contrôle du client en une fois et demi la période de temps de Keep Alive, il DOIT ANNULER la connexion réseau au client comme si le réseau avait échoué.

Si un client ne reçoit pas un paquet PINGRESP dans un délai raisonnable après avoir envoyé un PINGREQ, il DOIT fermer la connexion réseau au broker.

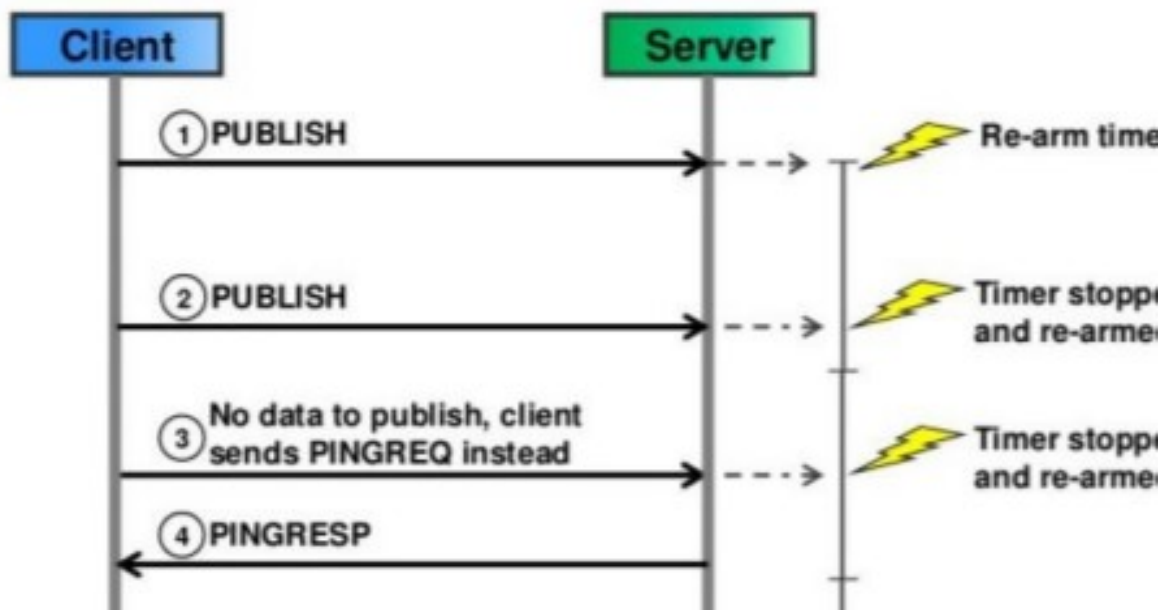


Figure 13 Exemple d'utilisation du drapeau Keep alive[26]



### II.3.2.2.4 Payload (Charge utile) :

La charge utile du paquet CONNECT contient un ou plusieurs champs préfixés en longueur, dont la présence est déterminée par les indicateurs dans l'entête de la variable. Ces champs, le cas échéant, DOIVENT apparaître dans l'ordre: Identifiant du client, Nom d'utilisateur, Mot de passe, Sujet (Will topic), WillMessage, Exemple: de paquet MQTT- CONNECT



### II.3.2.3 CONNACK

L'entête du CONNACK est constitué d'un bit (SP) pour indiquer si le broker possède déjà une session pour ce client, dans le cas où le client demande à restaurer une session existante, et d'un code de retour sur un octet (seuls les codes de 0 à 5 étant actuellement définis comme illustré dans le tableau 4).

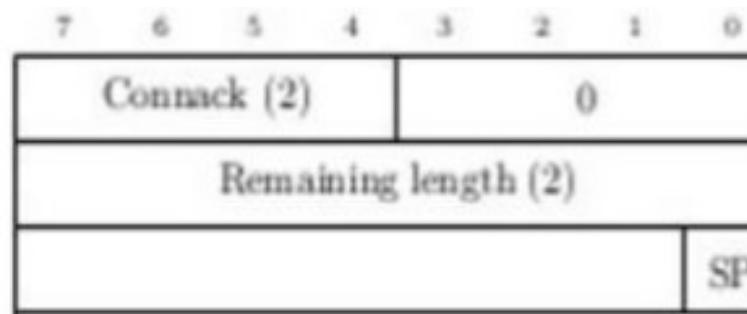


Figure 14 Format d'un paquet CONNACK

### II.3.2.3.1: Les valeurs du code de retour

Valeur	Réponse du code de retour	Description
0	0x00 Connexion acceptée	Connexion acceptée
1	0x01 Connexion rejetée, Version protocole inacceptable	Le broker ne prend pas en compte le niveau du protocole MQTT
2	0x02 Connexion refusée, identifiant rejeté	L'identifiant du client n'est pas autorisé
3	0x03 Connexion refusée, broker indisponible	La connexion réseau au broker n'est pas possible
4	0x04 Connexion refusée, mauvais nom d'utilisateur ou mot de passe	Les données dans le paquet sont incorrectes

Tableau 4 Les valeurs du code de retour d'un paquet CONNACK

### II.3.2.3.2 PUBLISH

Un paquet PUBLISH est comme illustrer sur la figure suivante:

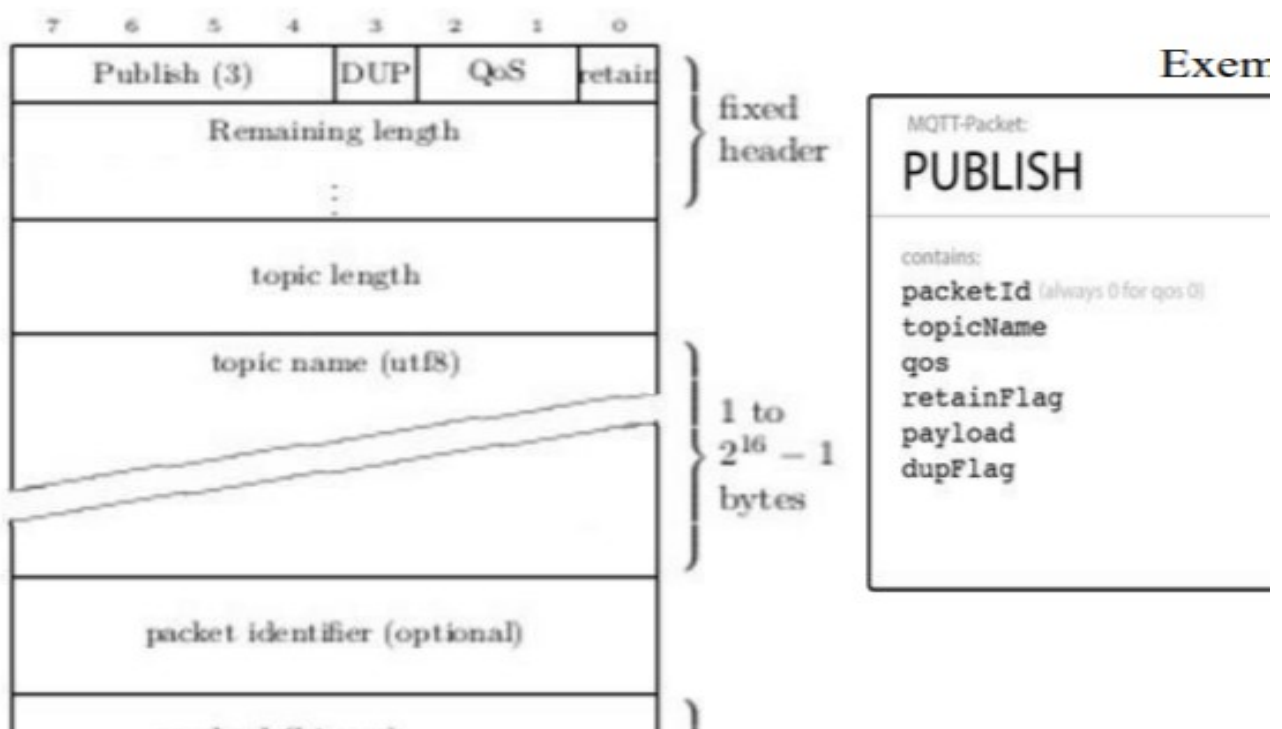


Figure 15 Format d'un paquet PUBLISH

Un paquet PUBLISH contient les valeurs:

### **II.3.2.3.3 Fixedheader**

#### **a) DUP**

- Si l'indicateur DUP est réglé sur 0, cela indique que c'est la première fois que le Client ou le Broker a tenté d'envoyer ce paquet MQTT PUBLISH.

- Si l'indicateur DUP est défini sur 1, cela indique que cela pourrait être la remise à neuf d'une tentative antérieure d'envoi du paquet.

#### **b) QoS:** Ce champ indique le niveau d'assurance pour la livraison d'un message d'application.

- Si un broker ou un client reçoit un paquet PUBLISH qui a deux bits QoS réglés sur 1, il DOIT fermer la connexion réseau.

#### **c) RETENIR**

- Si l'indicateur RETAIN est réglé sur 1, dans un paquet PUBLISH envoyé par un client à un broker, le broker DOIT stocker le message d'application et sa QoS, afin qu'il puisse être envoyé aux futurs abonnés dont les abonnements correspondent à son nom de sujet.

- Si le broker reçoit un message QoS 0 avec l'indicateur RETAIN réglé sur 1, il DOIT rejeter tout message précédemment retenu pour ce sujet. Il DOIT stocker le nouveau message QoS 0 comme nouveau message retenu pour ce sujet.

#### **d) Champ de longueur restante**

Il s'agit de la longueur de l'en-tête variable plus la longueur de la charge utile.

### **II.3.2.3.4.En-tête variable**

L'en-tête variable contient les champs suivants dans l'ordre: Nom du sujet, Identificateur de paquets.

### a) Nom du sujet (topic name)

Le nom du sujet identifie le canal d'information auquel les données de la charge utile sont publiées. Le nom du sujet DOIT être présent comme premier champ dans l'en-tête variable de paquet PUBLISH. Il DOIT être une chaîne codée UTF-8 telle que définie dans la section 2.3.

### b) Identificateur de paquet

Le champ Identificateur de paquets n'est présent que dans les paquets PUBLISH où le niveau QoS est 1 ou 2.

#### II.3.2.4 PUBACK, PUBREC, PUBREL, PUBCOMP

Tous ces paquets ont la même structure, seules les valeurs du code de commande et du champ Reserved peuvent varier.(voir tableau 1).

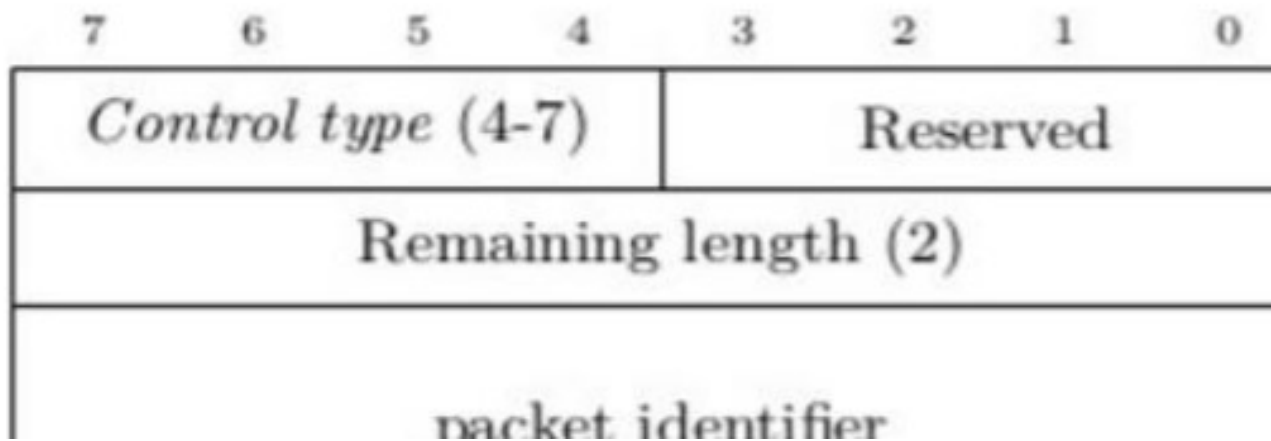


Figure 16 Format d'un paquet PUB [ACK/REC/REL/COMP]

### II.3.2.5 SUBSCRIBE

Un paquet SUBSCRIBE envoyé par un client comprend un identifiant pour la réponse, puis une liste de souscriptions d'au moins un élément. La structure d'une souscription est composée d'un filtre (longueur et chaîne de caractères) et de la qualité de service demandée

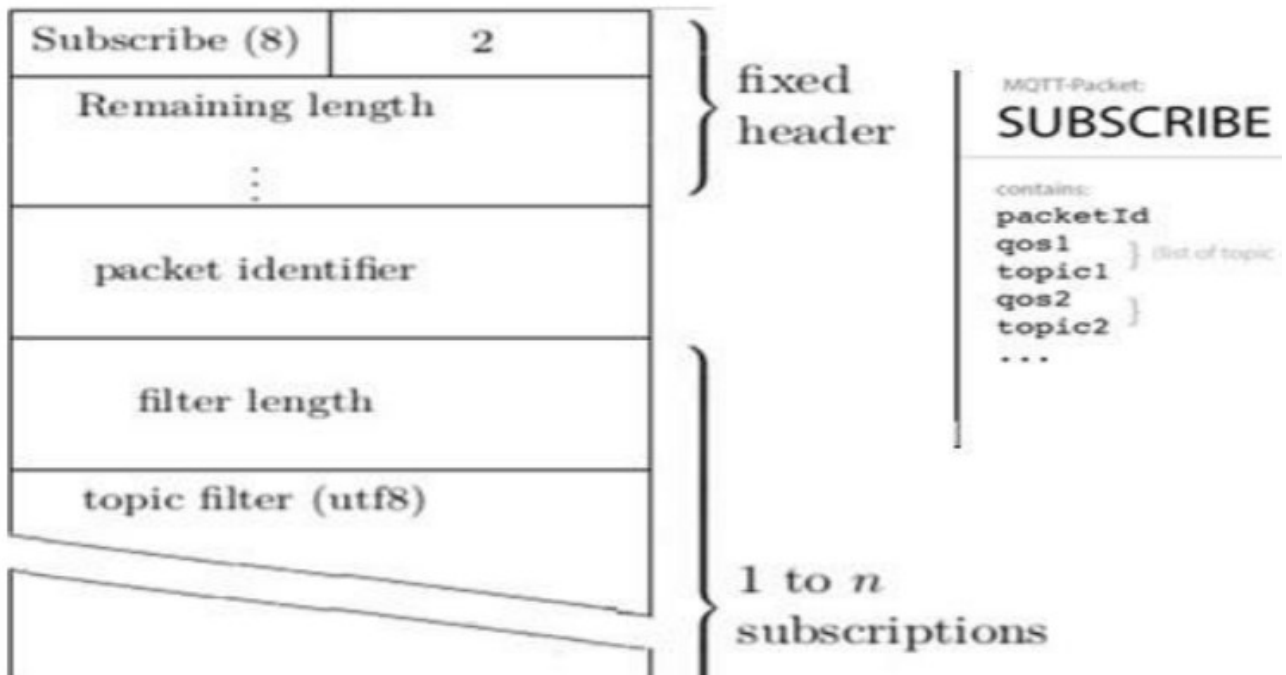


Figure 17 Format d'un paquet SUBSCRIBE

Les bits 3,2,1 et 0 de l'en-tête fixe du paquet de contrôle SUBSCRIBE sont réservés et DOIVENT être réglés respectivement à 0,0,1 et 0. Le broker DOIT traiter toute autre valeur comme mal formée et fermer la connexion réseau

**Exemple de paquet SUBSCRIBE :**

	Description	7	6	5	4							
<b>Topic Filter</b>												
byte 1	Length MSB (0)	0	0	0	0							
byte 2	Length LSB (3)	0	0	0	0							
byte 3	'a' (0x61)	0	1	1	0	<table border="1"> <tr> <td>Topic Name</td> <td>"/a</td> </tr> <tr> <td>Requested QoS</td> <td>0</td> </tr> <tr> <td>Topic Name</td> <td>"/a</td> </tr> </table>	Topic Name	"/a	Requested QoS	0	Topic Name	"/a
Topic Name	"/a											
Requested QoS	0											
Topic Name	"/a											
byte 4	'/' (0x2F)	0	0	1	0							
byte 5	'b' (0x62)	0	1	1	0							
<b>Requested QoS</b>												
byte 6	Requested QoS(1)	0	0	0	0							
<b>Topic Filter</b>												
byte 7	Length MSB (0)	0	0	0	0							
byte 8	Length LSB (3)	0	0	0	0							
byte 9	'c' (0x63)	0	1	1	0							

**II.3.2.6 SUBACK**

Un paquet SUBACK est envoyé par le broker au client pour confirmer la réception et le traitement d'un paquet SUBSCRIBE. Dans un paquet SUBACK, le broker répond avec le même identifiant de paquet, puis retourne les codes de retour dans le même ordre que les souscriptions du paquet SUBSCRIBE correspondant.

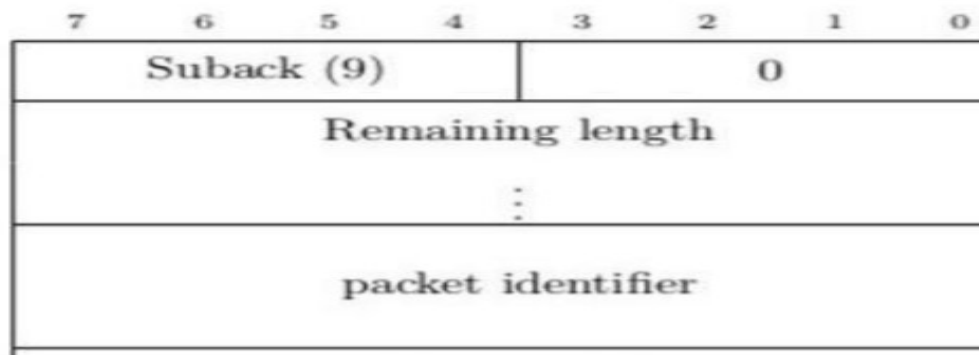


Figure 18 Format d'un paquet SUBACK

Un paquet SUBACK contient une liste de codes de retour, qui spécifient le niveau de QoS maximum qui a été accordé dans chaque abonnement qui a été demandé par le SUBSCRIBE. Ces codes de retour correspondent à la qualité de service garantie par le broker (qui peut être inférieure à celle initialement demandée), ou le code d'erreur 0x80 en cas de problème.

Description de code de retour		7	6	5
0x00 Success - Maximum QoS 0	0	0	0	0
0x01 Success - Maximum QoS 1	1	0	0	0
0x02 Success - Maximum QoS 2	2	0	0	0

Tableau 5 Code de retour de paquet SUBACK

### II.3.2.7 UNSUBSCRIBE

Un paquet UNSUBSCRIBE est envoyé par le client au broker, pour se désabonner des sujets, il ressemble à SUBSCRIBE, à ceci près que le client ne spécifié pas de qualité de service.

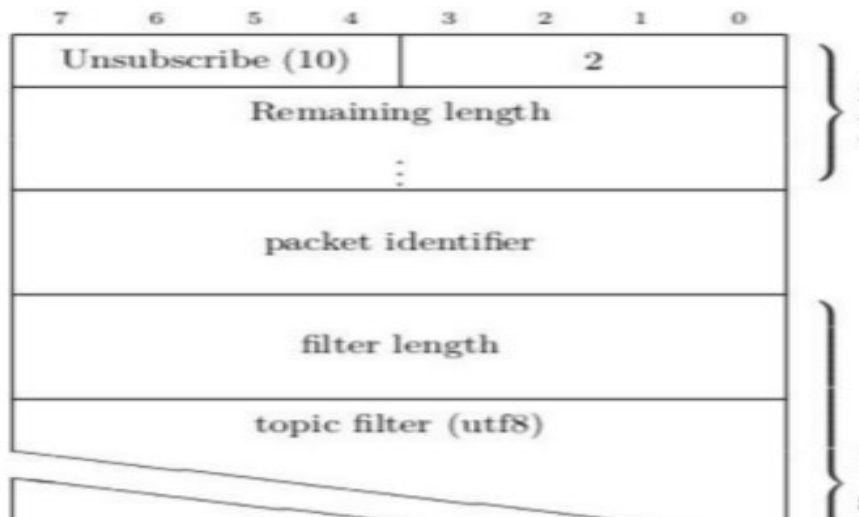


Figure 19 : Format d'un paquet UNSUBSCRIBE

### II.3.2.8 UNSUBACK

UNSUBACK sert de confirmation pour UNSUBSCRIBE, et contient simplement l'identifiant de paquet de la requête correspondante

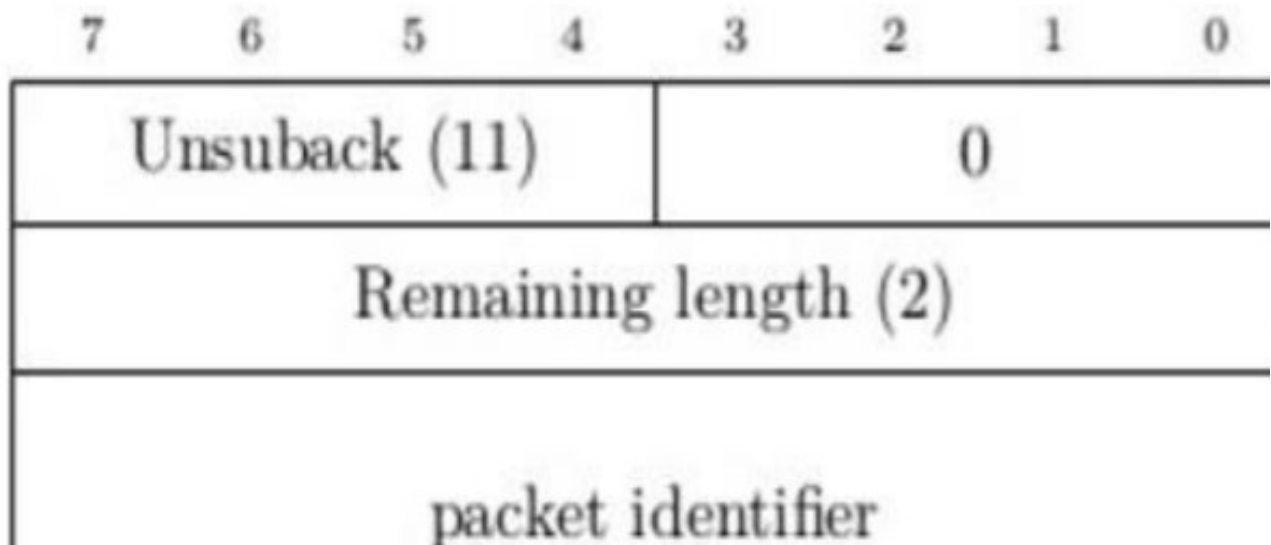


Figure 20 Format d'un paquet UNSUBACK



### II.3.2.9 PINGREQ, PINGRESP

Ces deux paquets sont très simples, et sont constitués uniquement de l'entête fixe du protocole, tenant ainsi sur deux octets.

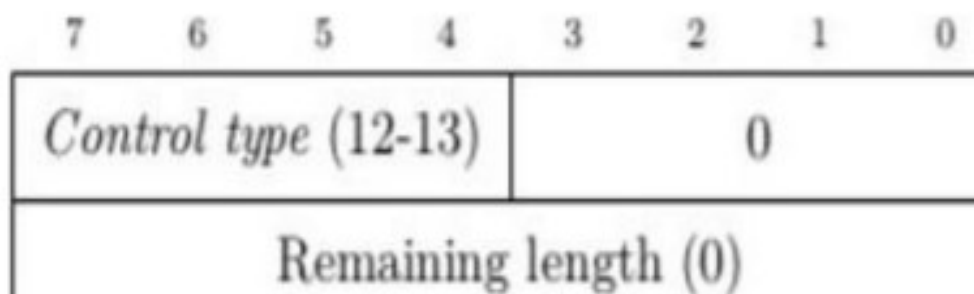


Figure 21 Format d'un paquet PING [REQ/RESP]

\*- Le paquet **PINGREQ** est envoyé d'un client au broker. Il peut être utilisé pour:  
1- Indiquez au broker que le client est en vie en l'absence de tout autre paquet de contrôle envoyé par le client au broker.

2- Demandez au Broker de répondre pour confirmer qu'il est en vie.

3- Exercer le réseau pour indiquer que la connexion réseau est active.

\*- Un paquet **PINGRESP** est envoyé par le broker au client en réponse à un paquet **PINGREQ**. Cela indique que le broker est en vie. Ces paquets sont utilisés dans le traitement **Keep Alive**, voir la section 3-2-2-3

### 3.2.10 DISCONNECT

Le paquet **DISCONNECT** est le paquet de contrôle final envoyé du client au broker. Cela indique que le client a l'intention de se déconnecter, et comporte uniquement l'entête fixe.

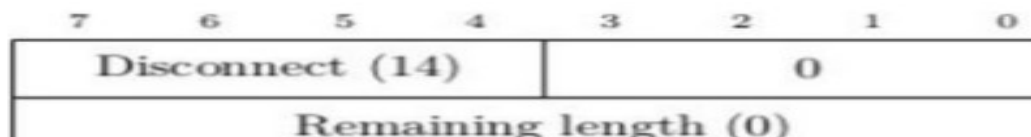


Figure 22 Format d'un paquet DISCONNECT

Sur réception de **DISCONNECT** le Broker **DOIT** rejeter tout message de volonté (Wil message) associé à la connexion actuelle sans le publier

### II.3.3 La synchronisation des requêtes envoyées au broker

Le tableau 6 décrit les méthodes du client MQTT qui envoient une demande au broker. À moins que le client d'application fixe un délai d'attente indéfini, le client attend indéfiniment jamais pour le broker. Si le client se bloque, il est soit un problème de programmation d'application, ou un défaut dans le client MQTT.

<b>Méthode</b>	<b>Synchronisation</b>	<b>inter</b>
Connect	Temps d'attente pour une connexion à établir avec le broker	La 30
Disconnect	Temps d'attente pour le client MQTT pour terminer les travaux qu'il doit faire, et pour la session TCP / IP pour déconnecter	défi lanc
Subscribe	Attend la fin de la méthode abonner ou désabonner	
UnSubscribe		

Tableau 6 Comportement de synchronisation des méthodes qui donnent lieu à des requêtes au broker

## II.3.4 L'identifiant de client MQTT

Le protocole MQTT définit un «identifiant client» (ID client) qui identifie de manière unique un client dans un réseau. En termes simples, lors de la connexion à un broker, un client doit spécifier une chaîne unique qui n'est pas utilisée actuellement et ne sera pas utilisée par un autre client qui se connectera au broker MQTT. Il existe plusieurs façons de choisir un identifiant client. Voici quelques exemples :

- Un capteur installé dans un emplacement particulier peut utiliser le code d'emplacement comme ID du client.

- Un appareil mobile possédant une capacité de réseau peut choisir l'adresse MAC ou un identifiant d'appareil unique comme ID de client mais MQTT restreint la longueur d'ID de client à 23 caractères. Par conséquent, il y aura une situation où l'identification du client doit être raccourcie.

En raccourcissant l'identifiant du client, il faut s'assurer que l'ID client n'est pas identique à toute autre ID client utilisée dans le réseau. Afin de garder l'identifiant court et unique, il faut présenter un mécanisme de génération d'identifiant fiable. Par exemple, créer un identifiant client à partir du périphérique MAC 48 bits adresse. Si la taille de la transmission n'est pas un problème critique, et utiliser les 17 octets restants pour rendre l'adresse plus facile à administrer, comme un texte lisible par l'utilisateur dans l'identifiant.

Maintenant, essayons de comprendre les implications de deux clients obtenant le même identifiant client.

Le broker MQTT surveille les messages en attente d'être envoyés à un client en fonction de l'identifiant du client. Ainsi, si un client utilise QoS 1 ou QoS 2 et abonné à n'importe quel sujet et déconnecté du broker, le broker sauvera les messages arrivés pour le client pendant qu'il était déconnecté. Une fois que le client se reconnecte, le broker enverra ces messages au client. Si un autre appareil MQTT utilise le même ID de client et se connecte au broker, le broker enverra les messages qu'il avait sauvegardés.

Un autre scénario lié aux identifiants clients est la connexion en double. Disons qu'un périphérique particulier à l'aide de l'ID client DeviceA est connecté au courtier. Si un autre client vient avec le même ID de client DeviceA, le broker peut décider s'il doit permettre au nouveau client de se connecter et de déconnecter le client existant ou de garder l'ancienne connexion en vie et interdire le nouveau client. Il s'agit d'une fonctionnalité facultative d'un broker MQTT.

## II.3.5 Callbacks

Le modèle de programmation client MQTT utilise des threads en profondeur. Les threads découplent une application client MQTT, autant qu'ils le peuvent, des retards dans la transmission de messages vers et depuis le broker. Les publications, les jetons de livraison et les événements perdus de connexion sont livrés aux méthodes d'une classe de rappel qui implémente `MqttCallback`. L'interface `MqttCallback` comporte trois méthodes de rappel

### II.3.5.1 ConnectionLost

o `Connection Lost` est appelé lorsqu'une erreur de communication entraîne la sortie de la connexion. Il est également appelé si le broker abandonne la connexion à la suite d'une erreur sur le broker une fois la connexion établie. Les erreurs de broker sont enregistrées dans le journal des erreurs du gestionnaire de files d'attente. Le broker supprime la connexion au client et le client appelle `MqttCallback.connectionLost`.

o Les seules erreurs distantes jetées comme exceptions sur le même thread que l'application client sont des exceptions de `MqttClient.connect`. Les erreurs détectées par le broker après que la connexion est établie sont un rapport à la `MqttCallback.connectionLost` méthode de rappel comme `Throwables`.

o Les erreurs typiques du broker qui résultent en `connectionLost` sont des erreurs d'autorisation. Par exemple, le broker de télémétrie tente de publier sur un sujet pour le compte d'un client qui n'est pas autorisé à publier sur le sujet. Tout ce qui se traduit par une `MQCC_FAIL` Le code d'état retourné au broker de télémétrie peut entraîner la chute de la connexion.

### II.3.5.2 DeliveryComplete

✓ `DeliveryComplete` est appelé par le client MQTT pour passer un jeton de livraison à la demande client, En utilisant le jeton de livraison, le rappel peut accéder au message publié avec la méthode **`token.getMessage`**.

✓ Lorsque le retour d'appel de l'application renvoie le contrôle au client MQTT après avoir été appelé par la méthode **`deliveryComplete`**, la livraison est terminée. Jusqu'à ce que la livraison soit terminée, les messages avec QoS 1 ou 2 Sont conservés par la classe de persistance.

- ✓ L'appel à `DeliveryComplete` est un point de synchronisation entre l'application et la classe de persistance. La méthode `DeliveryComplete` n'est jamais appelée deux fois pour le même message.
- ✓ Lorsque le retour d'appel de l'application revient de `DeliveryComplete` au client MQTT, le client appelle **`MqttClientPersistence.remove`** pour les messages avec QoS1 ou 2. **`MqttClientPersistence.remove`** supprime la copie stockée localement du message publié.
- ✓ Du point de vue du traitement des transactions, l'appel à `deliveryComplete` est une transaction monophasée qui engage la livraison. Si le traitement échoue pendant le rappel, lors du redémarrage du client, **`MqttClientPersistence.remove`** est appelé à nouveau pour supprimer la copie locale du message publié. Le rappel n'est pas encore appelé. Si vous utilisez le rappel pour stocker un journal des messages livrés, vous ne pouvez pas synchroniser le journal avec le client MQTT. Si vous souhaitez stocker un journal de manière fiable, mettez à jour le journal dans la classe **`MqttClientPersistence`**.
- ✓ Le jeton de livraison et le message sont référencés par le thread d'application principal et le client MQTT. Le client MQTT désélectionne l'objet **`MqttMessage`** lorsque la livraison est terminée et l'objet token de livraison lorsque le client se déconnecte. L'objet `MqttMessage` peut être récupéré lorsque la livraison est terminée si l'application cliente la détermine. Le jeton de livraison peut être récupéré après la déconnexion de la session.
- ✓ Vous pouvez obtenir `MqttDeliveryToken` et `MqttMessage` attributs après un message a été publié. Si vous tentez de définir des attributs `MqttMessage` après la publication du message, le résultat est indéfini.
- ✓ Le client MQTT continue de traiter les remerciements de la livraison si le client se reconnecte à la session précédente avec le même `ClientIdentifier`. L'application client MQTT doit configurer **`MqttClient.CleanSession`** pour faux Pour la session précédente, et la configurer faux Dans la nouvelle session. Le client MQTT crée de nouveaux jetons de livraison et des objets de message dans la nouvelle session pour les livraisons en attente. Il récupère les objets à l'aide de la classe `MqttClientPersistence`. Le rappel d'application est appelé dans la nouvelle session pour toutes les livraisons initiées dans la session précédente et complétées dans cette session.
- ✓ Le rappel de l'application est appelé après la connexion du client de l'application, lorsqu'une livraison en attente est terminée. Avant que le client de l'application ne se connecte, il peut récupérer les livraisons en attente en utilisant la méthode `MqttClient.get Pending Delivery Tokens`

### II.3.5.3 MessageArrived

- ✓ MessageArrived s'appelle lorsqu'une publication arrive pour le client correspondant à un sujet d'abonnement. Le sujet est le sujet de publication, pas le filtre d'abonnement. Les deux peuvent être différents si le filtre contient des caractères génériques.
- ✓ Si le sujet correspond à plusieurs abonnements créés par le client, le client reçoit plusieurs copies de la publication.
- ✓ Si un client publie sur un sujet auquel il s'inscrit également, il reçoit une copie de sa propre publication.
- ✓ Si un message est envoyé avec une QoS de 1 ou 2, Le message est mémorisé par la **classe MqttClientPersistence** avant que le client MQTT n'appelle message. **MessageArrived** se comporte comme deliveryComplete: il est appelé uniquement une fois pour une publication, et la copie locale de la publication est supprimée par **MqttClientPersistence.remove** lorsque **messageArrived** retourne au client MQTT

## II.4 Application de protocole

Après avoir expliqué ce protocole et présenté toutes caractéristiques et l'utilisation dans divers domaines , nous allons donc l'employer dans notre exemple de parking intelligent

# Chapitre III

## Matériaux et composants

## **Introduction**

Le choix des matériaux est une tâche fondamentale et très complexe, et pour réussir à bien l'aborder, on a posé un groupe de critères qu'on devait les satisfaire pour bien réussir notre choix.

Les critères imposés sont les suivants :

- **Au niveau de matériau :**

- 1) caractéristiques mécaniques
- 2) esthétiques, thermiques
- 3) électriques
- 4) économiques
- 5) environnementales
- 6) physiques.

- **Au niveau du procédé :**

- 1) Volume
- 2) Masse
- 3) Géométrie
- 4) taille de la série
- 5) caractéristiques économiques et environnementales.

### **III.1 La carte de Arduino**

Une carte Arduino est une petite carte électronique équipée d'un microcontrôleur. Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable. Elle peut être programmée pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le contrôle des appareils domestiques, éclairage, chauffage, le pilotage d'un robot, de l'informatique embarquée, ect

#### **III.1.1 Le type de la carte**

➤ **La carte Arduino UNO :**

C'est la carte idéale pour découvrir l'environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c'est la carte la plus utilisée, il est très facile de se référer aux tutoriels très nombreux sur le net et ainsi de ne pas rester seul dans son exploration. Sa simplicité devient par contre un handicap lorsqu'il s'agit de multiplier les périphériques, de manipuler



des algorithmes lourds ou d'interagie avec les OS / Android pour lesquels d'autres cartes Arduino sont plus adaptées.

### III.1.2 Composantes de la carte Arduino UNO :

Une carte Arduino UNO se compose essentiellement de :

- Prise jack : permet de brancher une alimentation (pile, batterie, adaptateur Secteur, + au centre Vin 7 à12 V
- Microcontrôleur : stocke le programme et l'exécute.
- Entrées et analogique : Permet de brancher des capteurs et des détecteur analogiques.
- Connexion USB (Universal Serial Bus) : Permet d'alimenter la carte en Énergie électrique (5V). Permet de téléverser le programme dans la carte.
- Entrées et sorties numériques (Digital) : Permet de brancher des actionneurs. Permet de brancher des détecteurs.

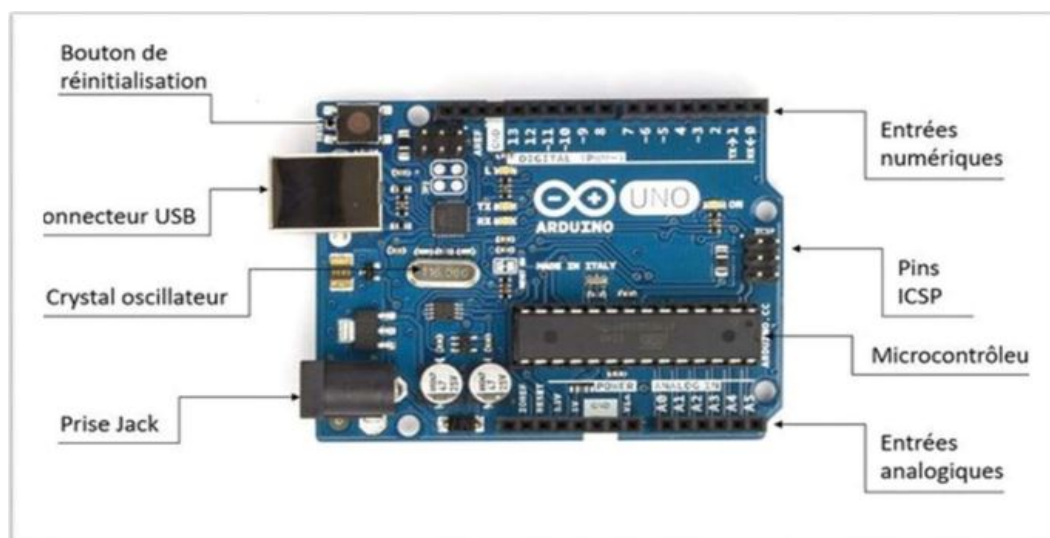


Figure 23 Composants et caractéristiques L'Arduino UNO

### III.2 Servo-moteur

Un servo-moteur est un type de moteur électrique. C'est un dispositif typiquement utilisé en modélisme par exemple, contrôler la direction d'une voiture télécommandée. Sur un servo-moteur, l'angle de l'axe reste fixé dans une position et peu varier entre 0 et 180° en fonction du signal envoyé

### III. 2.1 Les types de servo-moteur

Il existe divers types de servomoteur, de taille, poids et couple (force) différents. la photographie ci-dessous présente un servomoteur très classique en modélisme : le futaba S3003. Un peu plus bas l'article, on utilisera un autre servomoteur, communément appelé "servomoteur 9 grammes", par souci de consommation électrique



Figure 24 servomoteur 9 grammes

### III.2.2 Composition d'un servo-moteur

Le servomoteur est composé de plusieurs éléments visibles et invisible :

- Un moteur à courant continu
- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu
- Les fils, qui sont au nombre de trois
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège

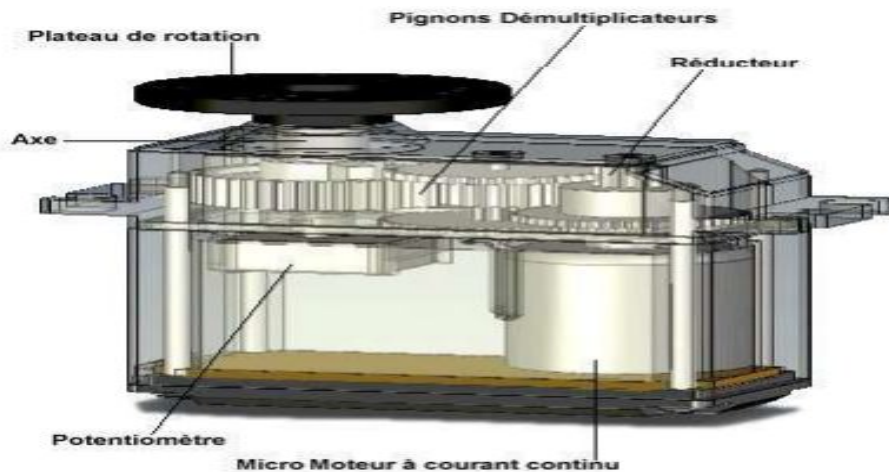


Figure 25 Vue interne d'un servomoteur

### III.2.3 Principe de fonctionnement d'un servo-moteur

La plupart des servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

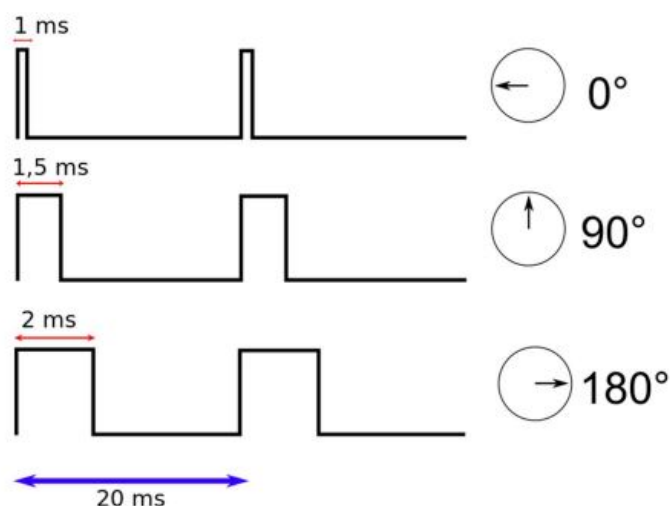


Figure 26 simulation d'un pont en H réalisé par Proteus à base des transistors

## III.3 Capteur Infrarouge

### III.3.1 Définition

Le capteur d'évitement d'obstacles à capteur infrarouge IR est une solution peu coûteuse pour la détection d'évitement pour la robotique, la voiture intelligente et d'autres utilisations électroniques. Dans cette leçon, nous allons vous montrer comment fonctionne le capteur d'évitement d'obstacles et comment l'utiliser avec la carte ARDUINO UNO.



Figure 27 Capteur infrarouge

Le capteur infrarouge ou le capteur IR a deux parties principales. Émetteur IR et récepteur IR. La responsabilité de l'émetteur IR ou de l'émetteur infrarouge est la transmission des ondes infrarouges tandis que le travail du récepteur IR est de recevoir ces ondes infrarouges. Le récepteur IR envoie constamment des données numériques sous la forme de 0 (LOW) ou 1 (HIGH) à la broche V out du capteur.

### III.3.2 Principe de fonctionnement

L'émetteur infrarouge envoie un signal infrarouge qui, en cas de surface réfléchissante (par exemple de couleur blanche), rebondit dans certaines directions, y compris celle du récepteur infrarouge qui capte le signal détectant l'objet

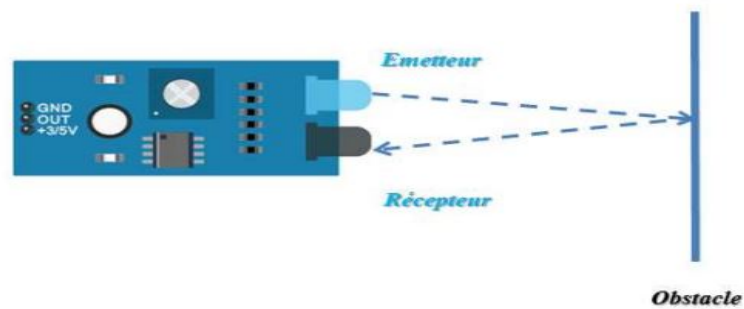


Figure 28 les ondes infrarouges

Lorsque la surface est absorbante (par exemple de couleur noire), le signal IR n'est pas réfléchi et l'objet ne peut pas être détecté par le capteur. Ce résultat se produirait même si l'objet est absent.

### III.3.3 Caractéristiques Techniques

- Voltage de fonctionnement : DC 3,3 V-5 V
- Intensité du courant  $\geq 20\text{mA}$
- Température de fonctionnement :  $-10\text{ °C} \rightarrow +50\text{ °C}$
- Plage de distance : 2-40 cm
- Interface IO : interface 4 fils (GND / Vcc / OUT / EN)
- Signal de sortie : tension TTL
- Mode d'hébergement : régulation de résistance multi-cercle
- Angle effectif :  $35\text{ °}$
- Taille :  $41,7 * 16,7\text{ mm}$
- Poids : 5g

## Conclusion

Le choix des matériaux était la phase la plus délicate dans notre projet. En effet, il ne portait pas seulement sur un aspect purement technique répondant à des exigences fonctionnelles, mais aussi à des attentes relevant des préférences des utilisateurs dans le cadre d'un marché spécifique. En gros,

le choix des matériaux a été analysé sous l'angle de l'ingénierie des matériaux, du temps de réponse, sans oublier enfin le côté esthétique (le design industriel).

# Chapitre IV

## Simulation & Réalisation

## Introduction

Notre simulation et réalisation de parking intelligent était avant l'avant-dernière tâche que nous devons faire, les études étaient déjà terminées et les matériaux étaient déjà en place, il ne manquait plus qu'une structure pour accueillir ces appareils puis vérifier si

les mesures que nous avons faites concilieraient les côtés techniques avec le côté esthétique.

### IV.1 Simulation des différents circuits

#### IV.1.1 Simulateur Isis Proteus

Isis Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, elle est principalement connue pour éditer des schémas électriques. Par ailleurs le logiciel permet également de simuler ces schémas.

Les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

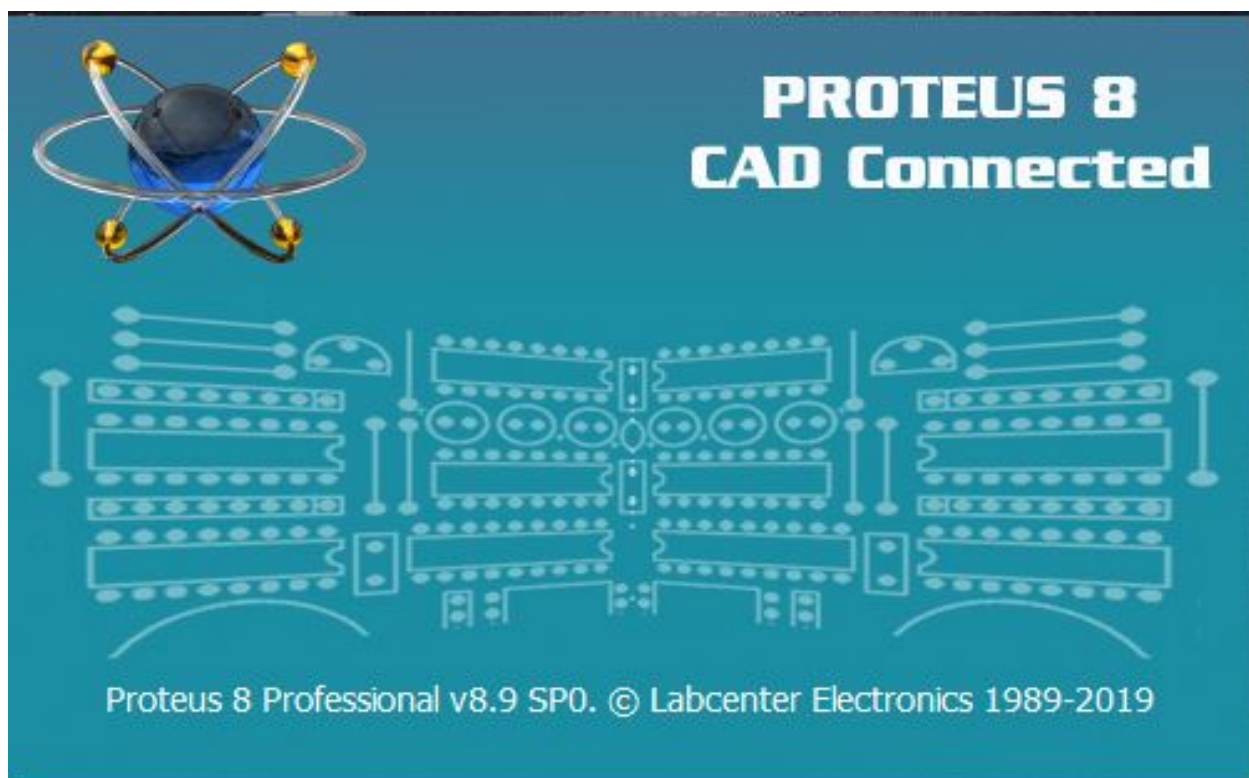


Figure 29 Simulation Isis Proteus



## IV.1.2 Interface utilisateur

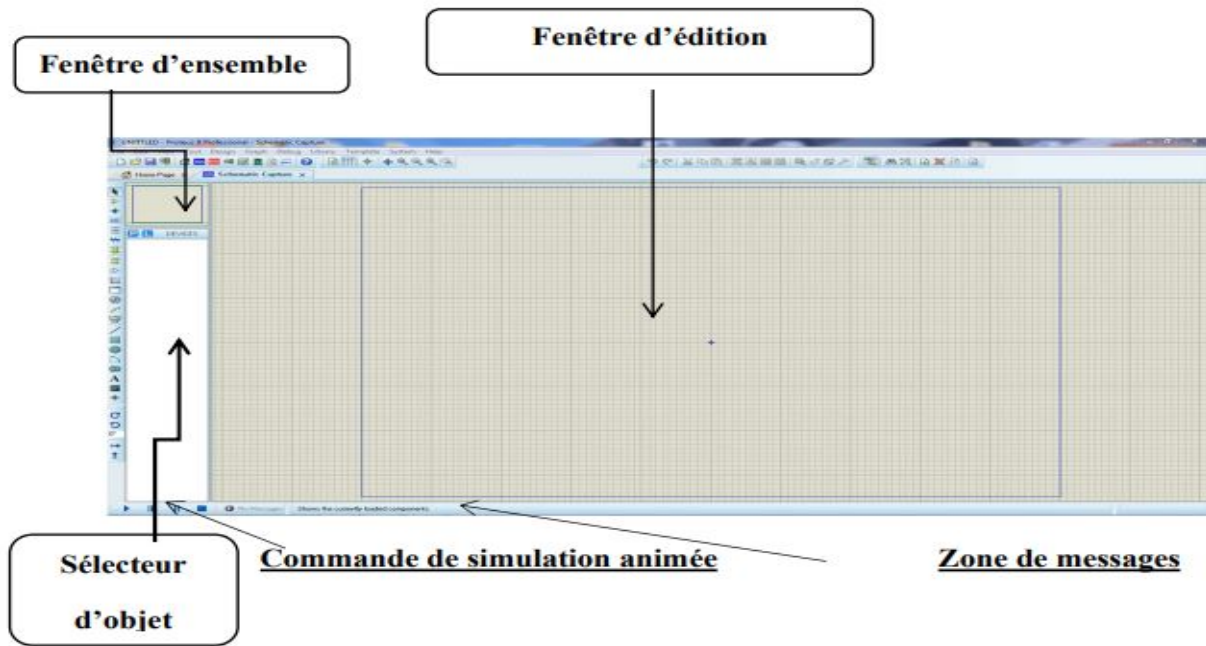


Figure 30 Interface utilisation

## IV.1.3 Langage Arduino

### IV.1.3.1 Définition

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++. Le langage impose une structure particulière typique de l'informatique embarquée. La programmation se fait de manière séquentielle c'est-à-dire une suite d'instruction élémentaires



Figure 31 Langage Arduino

### IV.1.3.2 Interface d'utilisateur

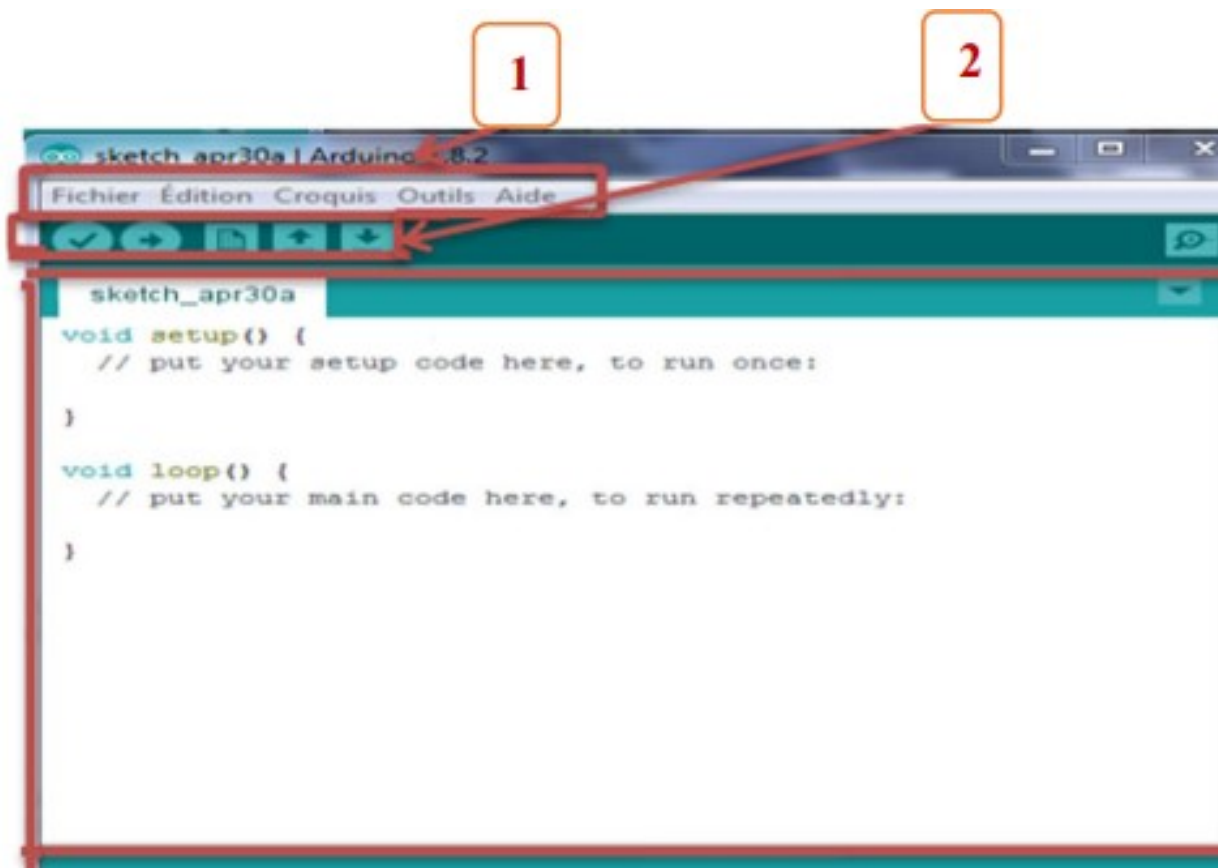


Figure 32 Interface d'utilisateur

**Le cadre numéro 1** : ce sont les options de configuration du logiciel

**Le cadre numéro 2** : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes

**Le cadre numéro 3** : ce bloc va contenir le programme que nous allons créer

**Le cadre numéro 4** : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.

**Un programme** (ou "**sketch**") Arduino est constitué de 2 fonctions distinctes :

La fonction de configuration "**void setup**" exécutée une seule fois au lancement du Programme. On appelle aussi cette fonction : fonction d'initialisation.

La fonction "**void loop**" qui est ensuite exécutée indéfiniment en boucle

## IV.1.4 Fritzing

### IV.1.4.1 Définition

Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon.



Figure 33 Fritzing

## IV.1.4.2 Interface d'utilisateur

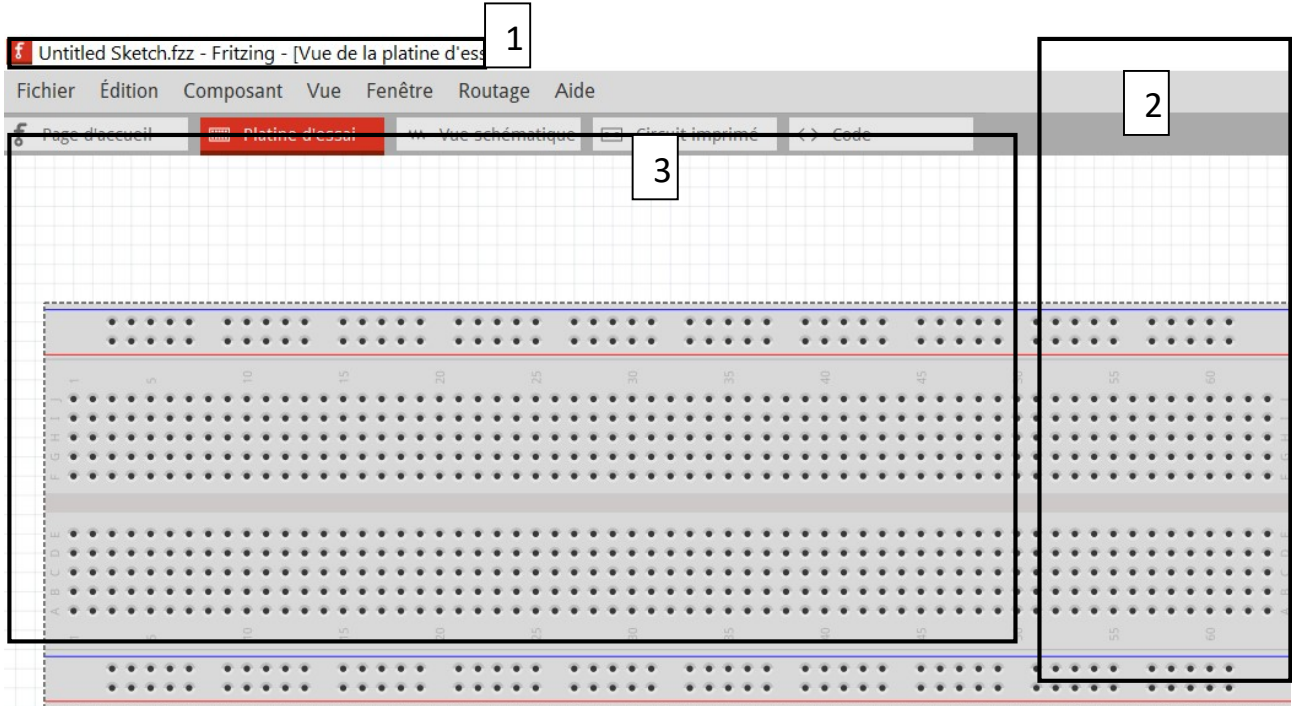


Figure 34 Interface Fritzing

**Le cadre numéro 1** : ce sont les options de configuration du logiciel

**Le cadre numéro 2** : ce bloc va contenir les composants

**Le cadre numéro 3** : le schéma que nous allons créer et type de voir

## IV.2 Schéma de parking intelligent

Voici le schéma de notre robot, on a abouti ce schéma en utilisant le logiciel Fritzing avec l'aide de Photoshop.

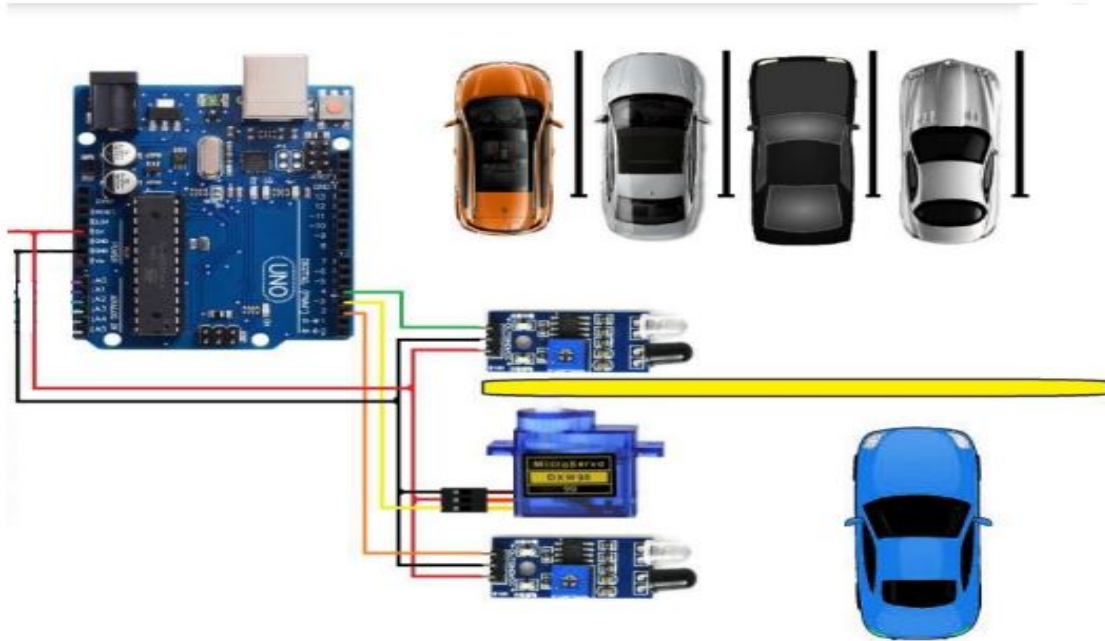


Figure 35 schéma électrique du parking intelligent

on les a assimilés tous ensemble et voilà ce qu'on a obtenu :

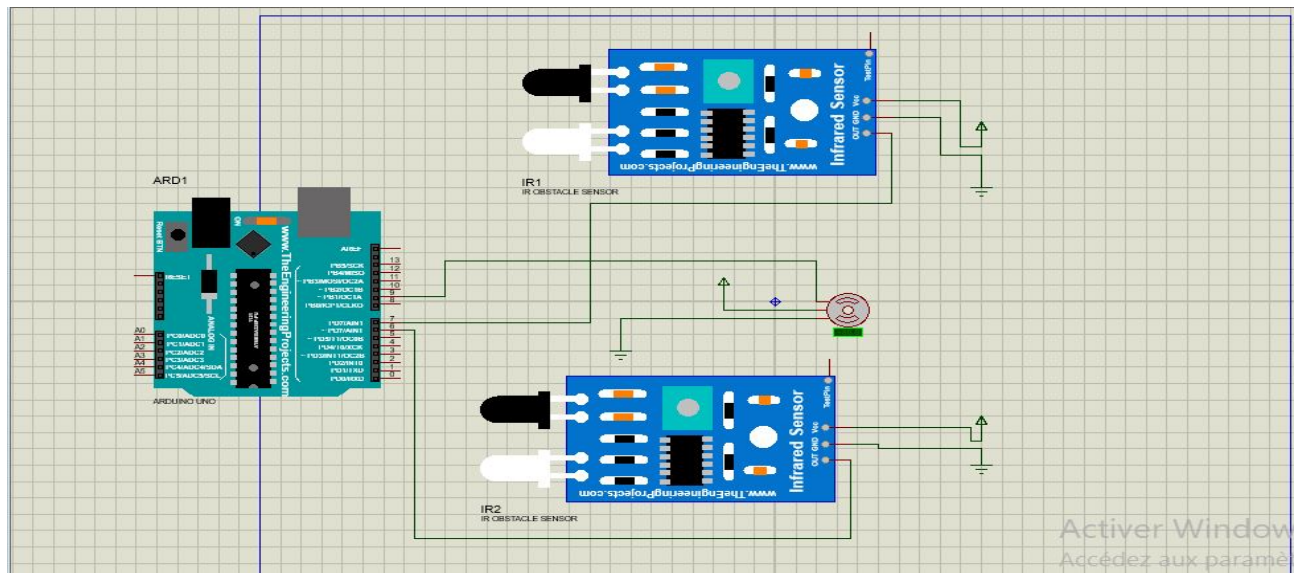


Figure 36 montage reliant les différents composants du parking intelligent

### IV.3 Réalisation du parking intelligent

La réalisation de notre projet qui donne les images suivant :

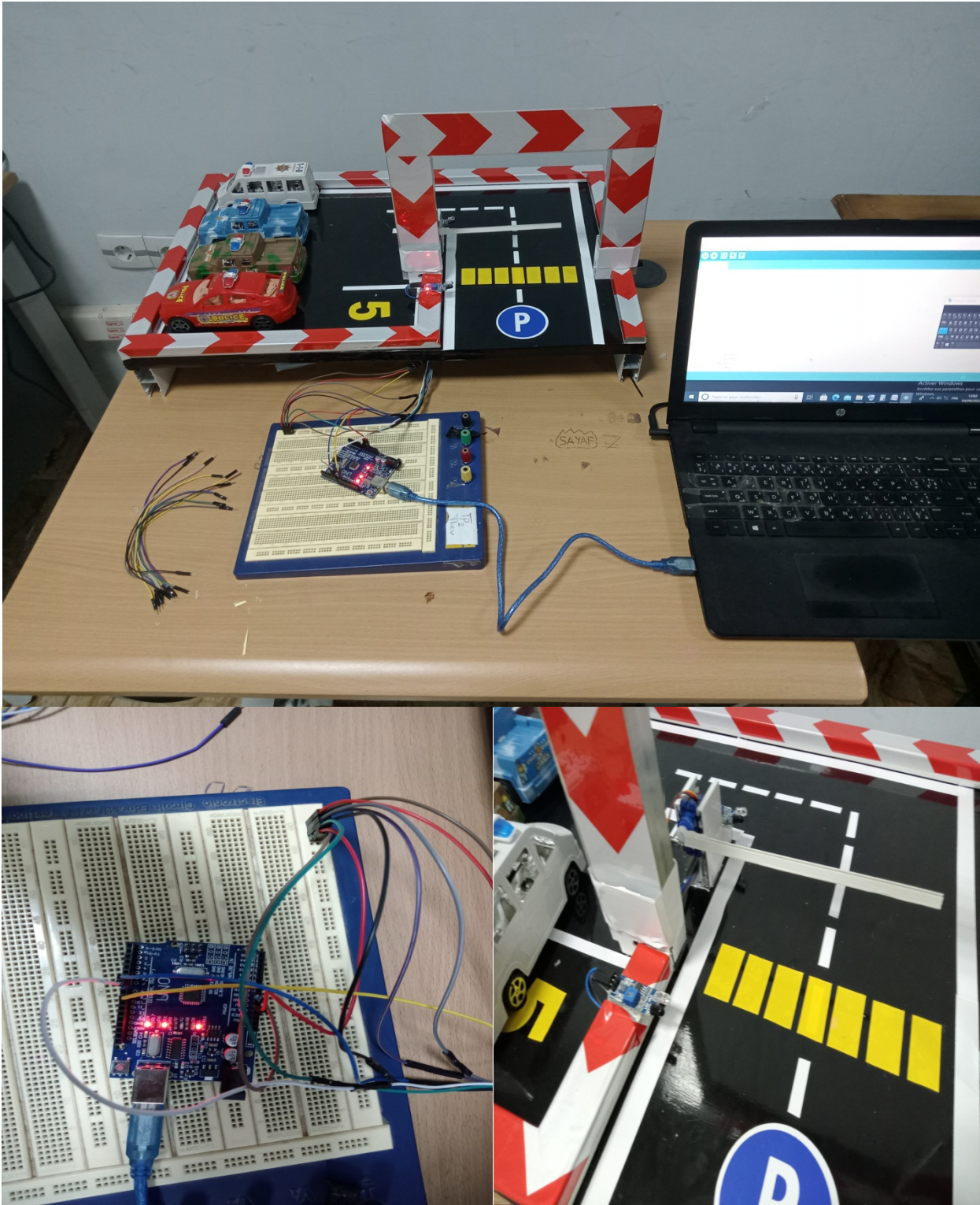


Figure 37 image de réalisation du parking intelligent

## IV.4 Exécuter le projet

### 1<sup>er</sup> cas :

Dans ce cas barrière est fermé :

- Position de barrière (0 deg)
- Capture n'indique pas les voiture , donner résultat 0 figure 39

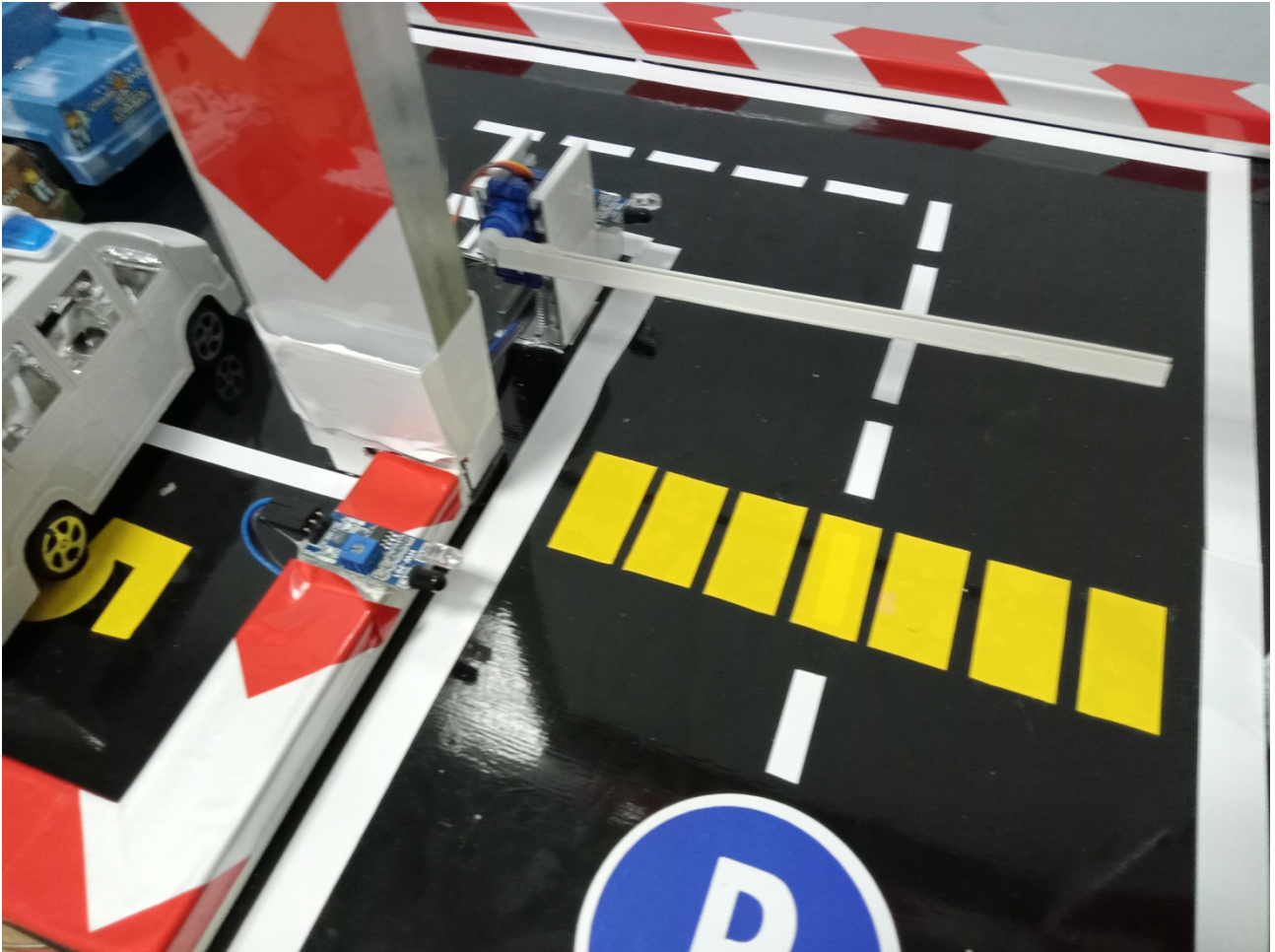


Figure 38 image du parking a l'état de repos

```
COM10
11:38:24.617 -> Entrer = 0
11:38:24.617 -> Sortie = 0
11:38:24.664 -> ----Fermie----
11:38:24.664 -> Entrer = 0
11:38:24.710 -> Sortie = 0
11:38:24.710 -> ----Fermie----
11:38:24.757 -> Entrer = 0
11:38:24.757 -> Sortie = 0
11:38:24.757 -> ----Fermie----
11:38:24.757 -> Entrer = 0
11:38:24.757 -> Sortie = 0
11:38:24.804 -> ----Fermie----
11:38:24.804 -> Entrer = 0
11:38:24.804 -> Sortie = 0
11:38:24.851 -> ----Fermie----
11:38:24.851 -> Entrer = 0
```

Défilement automatique  Afficher l'horodatage Nouvelle ligne 9600 baud Effacer la sortie

Figure39 image de l'état de barrière o/f (pas de voiture)

2<sup>eme</sup> cas :

**a/ Entrer de voiture**

Dans ce cas la voiture entrer :

- Position de barrière (90 deg)
- Capture indique la voiture , donner résultat 1 a l'ordre figure 41

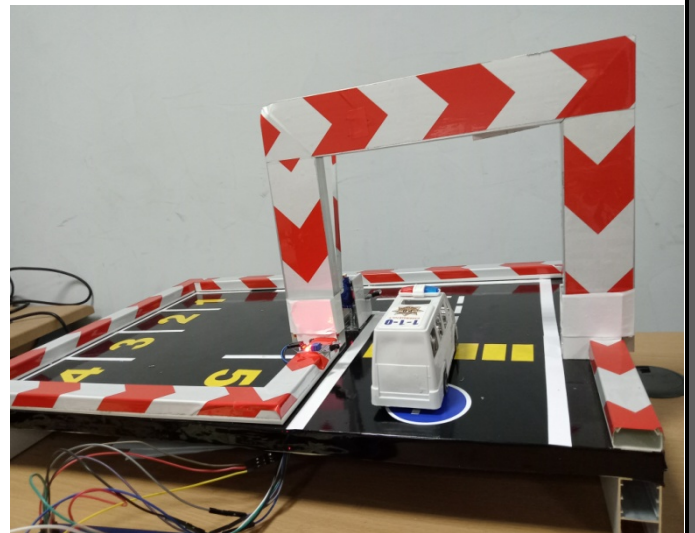


Figure 40 image de ouverture de barrière pour entrer la voiture



```
COM10
11:40:55.724 -> Entrer = 1
11:40:55.724 -> Sortie = 0
11:40:55.724 -> ----Ouvert----
11:40:56.708 -> ----Fermie----
11:40:56.708 -> Entrer = 1
11:40:56.708 -> Sortie = 0
11:40:56.755 -> ----Ouvert----
11:40:57.708 -> ----Fermie----
11:40:57.708 -> Entrer = 1
11:40:57.708 -> Sortie = 0
11:40:57.755 -> ----Ouvert----
11:40:58.724 -> ----Fermie----
11:40:58.724 -> Entrer = 1
11:40:58.724 -> Sortie = 0
11:40:58.724 -> ----Ouvert----
```

Défilement automatique  Afficher l'horodatage

Nouvelle ligne 9600 baud Effacer la sortie

Figure41 image de l'état de barrière o/f (voiture entrer)

### b/ Sortie de voiture

Dans ce cas la voiture sortie :

- Position de barrière (90 deg)
- Capture indique la voiture , donner résultat 1 a l'ordre figure 43



Figure 42 image de ouverture de barrière pour sortie la voiture

```
COM10
Envoyer
11:41:59.194 -> Sortie = 0
11:41:59.194 -> ----Fermie----
11:41:59.241 -> Entrer = 0
11:41:59.241 -> Sortie = 0
11:41:59.241 -> ----Fermie----
11:41:59.287 -> Entrer = 0
11:41:59.287 -> Sortie = 0
11:41:59.287 -> ----Fermie----
11:41:59.287 -> Entrer = 0
11:41:59.287 -> Sortie = 1
11:41:59.334 -> ----Ouvert----
11:42:00.319 -> ----Fermie----
11:42:00.319 -> Entrer = 0
11:42:00.319 -> Sortie = 1
11:42:00.319 -> ----Ouvert----
```

Défilement automatique  Afficher l'horodatage Nouvelle ligne 9600 baud Effacer la sortie

Figure 43 image de l'état de barrière o/f (voiture sortie)

### 3<sup>eme</sup> cas :

Dans ce cas les deux voitures dans la position d'entrer et sortie :

- Position de barrière (90 deg)
- Capture indique les voiture , donner resulta 1/1 figure 45



Figure 44 image de ouverture de barrière pour sortie/entrer les voitures

The screenshot shows a terminal window titled 'COM10'. The window contains a list of timestamped messages indicating the state of a barrier. The messages are as follows:

```
11:42:57.331 -> ----Fermie----  
11:42:57.331 -> Entrer = 1  
11:42:57.331 -> Sortie = 1  
11:42:57.377 -> ----Ouvert----  
11:42:58.297 -> ----Ouvert----  
11:42:59.329 -> ----Fermie----  
11:42:59.329 -> Entrer = 1  
11:42:59.329 -> Sortie = 1  
11:42:59.375 -> ----Ouvert----  
11:43:00.313 -> ----Ouvert----  
11:43:01.344 -> ----Fermie----  
11:43:01.344 -> Entrer = 1  
11:43:01.344 -> Sortie = 1  
11:43:01.344 -> ----Ouvert----  
11:43:02.297 -> ----Ouvert----
```

At the bottom of the window, there are several controls: a checked checkbox for 'Défilement automatique', a checked checkbox for 'Afficher l'horodatage', a dropdown menu for 'Nouvelle ligne', a dropdown menu for '9600 baud', and a button labeled 'Effacer la sortie'. An 'Envoyer' button is also present in the top right corner of the text area.

Figure 45 image de l'état de barrière o/f (voiture entrer/sortie)

## Conclusion générale

Nous avons vu que MQTT est un protocole tout à la fois minimaliste et très simple à mettre en œuvre (en particulier du côté client). Malgré tout, il reste suffisamment souple et extensible pour satisfaire aux besoins d'un bon nombre d'applications, et est donc particulièrement adapté au contexte de l'IoT. C'est la raison pour laquelle nous avons choisi ce protocole pour l'implantation et la réalisation d'une application de contrôle de parking intelligent dans IoT

# Bibliographie

## Références

- [1] E. L'hote, «Les 5 secteurs pour entreprendre en 2018,»  
[En ligne]. Available: <https://blog.compta-clementine.fr/>.
- [2] «Tout savoir sur l'Internet des objets,»  
[En ligne]. Available: <https://formati.online/internetobjets/>.
- [3] D. Evans, «internet-of-things,»  
[En ligne]. Available: [www.cisco.com](http://www.cisco.com).
- [4] Arnaud, «L'internet des objets à haute fréquence,»  
[En ligne]. Available: <https://www.objetconnecte.net/booster-iiot-1807/>.
- [5] «Internet des objets,» [En ligne]. Available:  
<https://www.futurasciences.com/tech/definitions/internet-internet-objets-15158/>.
- [6] «4- choses a savoir sur internet des objets,»  
[En ligne]. Available: <https://www.synox.io/>.
- [7] izihome, «Les objets connectés: avantages et inconvénients,»  
[En ligne]. Available: <http://izihome.fr/objets-connectes-avantages-inconvenients/>.
- [8] N. C. Guillaume Plouin, «Modèles d'architectures de l'Internet des Objets,»  
[En ligne]. Available: <https://blog.octo.com/>.
- [9] «L'IoT au travail aujourd'hui,»  
[En ligne]. Available: <https://www.intel.fr/>.
- [10] Renaud, «L'internet des objets et ses applications au quotidien,»  
[En ligne]. Available: <https://www.objetconnecte.com/>.
- [11] «Solution IoT,»  
[En ligne]. Available: <https://www.alpwise.com/>.
- [12] B. Berard M. Bidoit A. Finkel F. Laroussinie, A. Petit L. Petrucci Ph. Schnoebelen, with P. McKenzie, Systems and Software Verification- Model-Checking Techniques and Tools With 67 Figures, Springer 2001
- [13] Carles Anton-Haro and Mischa Dohler, Machine-to-machine (M2M) Communications Architecture, Performance and Applications, Elsevier, 2015, p 91.
- [14] <http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>
- [15] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS
- [16] <https://www.oasis-open.org/news/pr/oasis-mqtt-internet-of-things-standard-nowapproved-by-iso-iec-jtc1>.

- [17] Protocol Analysis - oneM2M, page 21,  
URL: [http://www.onem2m.org/images/files/deliverables/TR-0009-Protocol\\_Analysis-V0\\_7\\_0\\_1.pdf](http://www.onem2m.org/images/files/deliverables/TR-0009-Protocol_Analysis-V0_7_0_1.pdf)
- [18] Lucy Zhang, Building Facebook Messenger, août 2011,  
URL: <https://www.facebook.com/notes/facebook-engineering/building-facebookmessenger/10150259350998920/>
- [19] <https://aws.amazon.com/fr/iot-platform/how-it-works/>
- [20] mqtt-topics-best-practices,  
URL: <http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>.
- [21] Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry, IBM Redbooks, septembre 2012, p 26.
- [22] mqtt-essentials-part-9-last-will-and-testament,  
URL: <http://www.hivemq.com/blog/mqtt-essentials-part-9-last-will-and-testament>.
- [23] Eric J. Bruno, MQTT Programming In Depth, March 2016,  
URL: <http://programmingwithreason.com/article-mqtt-in-depth.html>.
- [24] Internet of Things MQTT Quality of Service Levels  
URL: <http://www.ossmentor.com/2015/04/internet-of-things-mqtt-quality-of.html>
- [25] Jan-Piet Mens, Using-the-MQTT-IoT-protocol-for-unusual-but-useful-purposes  
URL : <http://www.admin-magazine.com/Archive/2015/30/Using-the-MQTT-IoT-protocol-forunusual-but-useful-purposes>
- [26] Travaux de la 4e promotion (2013-2014) du Cycle « Sécurité des usages numériques », p 28

## Programme

```
#include <Servo.h>

Servo myservo;

#define Entre 6
#define Sorte 7
byte S,E,i = 0;
int D = 1000;

void setup() {
  myservo.attach(9);
  Serial.begin(9600);
  pinMode(Entre,INPUT);
  pinMode(Sorte,INPUT);
}
void loop(){
  E = digitalRead(Entre);
  S = digitalRead(Sorte);
  Serial.print("Entrer = ");
  Serial.println(1-E);
  Serial.print("Sortie = ");
  Serial.println(1-S);
  if(!E){
    myservo.write(90);
    Serial.println("----Ouvert----");
    delay(D);
  }
  if(!S){
    myservo.write(90);
    Serial.println("----Ouvert----");
    delay(D);
  }
  myservo.write(0);
  Serial.println("----Fermie----");
}
```