الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département du Génie Électrique

Projet de fin d'études

Présenté pour l'obtention du Diplôme De Master en Systèmes des Télécommunications
Spécialité : Systèmes Des Télécommunications

THEME

Conception D'un Système D'acquisition Des Données En Temps Réel A Base De IOT En Utilisant Le Protocole MQTT

Proposé et dirigé par

Encadreur : Dr. M. BOUHEDDA

Co-encadreur : Mr.Y.ANIBA

Préparé et présenté par

MOULAI MOSTEFA sakina

LARBI chaimaa

Année Universitaire 2020-2021

إهداء

أولا لك الحمد ربي على كثير فضلك و جميل عطائك و جودك ، الحمد لله ربي و مهما حمدنا فلن نستوفي حمدك و الصلاة و السلام على من لا نبي بعده. أما بعد :

إلى درعي الذي به احتميت ، إلى الذي لم أسر يوما في طريق إلا و رأيته سبقني يمهده لي ، إلى فخري و اعتزازي حبيبي **أبي** ، اللهم **أبي** في كل اتجاهاتي ، اللهم **أبي** إلى الأبد .

إلى الينبوع الذي لا يمل العطاء ، إلى من حاكت سعادتي بخيوط منسوجة من قلبها ، إلى من بحنانها ارتويت ، إلى مدرستي الأولى في الحياة ، إلى ملاكي **أمي** حفظك الله يا جنة دنياي .

إلى وحيدتي ، يدي اليمنى ، ضلعي الثابت الذي لا يميل ، إلى أختي **أمال** .

إلى من بهم أكبر و عليهم أعتمد و أسند ظهري ، إخوتي نبض قلبي **محمد و عثمان** .

إلى أميرتي الصغيرة ، الحفيدة الأولى ، بهجة بيتنا **أروى** .

إلى سندي الثاني ، شريك حياتي عبد الرحمان .

إلى قدوتي خالتي **فاطمة الزهراء** نور عيني ، أرجو من الله أن يمد في عمرك يا غالية .

إلى من رحلت عن دنيتي و لم ترحل من قلبي خالتي **رقية** رحمك الله يا طيبة .

إلى حبيبة قلبي و نجمة حياتي ابنة خالتي **زكية** .

إلى التي ساعدتني و ساندتني في عملي هذا ، إلى الحنونة ابنة عمي **حنان** .

إلى صديقات العمر **سلمى** ، **صفية** ، **اسما** ،**أميرة** ، **شامية** ، و **سارة** .

إلى من قاسمتني هذا العمل ، شريكتي الحلوة ، رفيقة الدرب أختي **شيماء** .

إلى **أساتذتي** ، و كل من علمني حرفا فصرت له بعلمي عبدا .

إلى كل من كان عونا لي ، إلى كل من دعا لي و ساندني ، إلى كل زميلاتي و زملائي تخصص أنظمة اتصالات سلكية و لا سلكية دفعة -2021.

أهدي هذا العمل .

سكينة

المدية في : 2021-07-06

إهداء

في البداية الشكر و الحمد لله، جل في علاه فإليه ينسب الفضل كله في إكمال هذا العمل . أما بعد :

اهدي تخرجي إلى رجلي الأول ورجلي الأخير إلى من كلله الله بالهيبة والوقار من احمل اسمه بكل افتخار إلى من أمضى وقته في الكد والعمل ليرسم لنا طريق ممهدا نحو المستقبل أدعو الله أن يمد في عمرك لترى ثمارا قد حان قطافها **أبي الغالي**.

إلي الغالية التي أرى الأمل من عينيها ملاكي في الحياة ومعنى الحب من كان دعائها سر نجاحي وحنانها بلسم جراحي **أمي الغالية.**

والدي الكريمين "وقُل رّبّ ارحمهُما كما ربّياني صغيرًا"

إلى أخي **محمد** حمى ظهري وذخري وثروتي العظيمة يا حائطي الذي أتكئ عليه عند شدتي وملاذي بعد الله عند كُربتي أيها الأخ الكريم والعظيم لقد كنت لنا أباً وأخاً ومربّيا ولن أنسى تشجيعك لي ما حييت.

إلى من حبه يجري في شراييني أخي الصغير **سراج.**

إلى بحر الحُب والحنان والنبض الساكن في عروقي **أختي شهد**.

الى نجوم سمائي المتلألئة وسندي في الحياة وصديقات عمري:**رانيا، خوله، سعيدة.**

الى من سرنا سويا ونحن نشق طريق النجاح معا صديقتي وأختي **سكينة.**

إلي حبيبة قلبي وبهجة حياتي **أميرة.**

الى سند المستقبل وشمعة العطاء وأمل الغد المُشرق أساتذتي وزملائي إلى كُل قلب خفق ليّ حباً وخوفاً عليّ أهدي إليهم ثمرة جُهدي المتواضع هذا الكلام يطول والمختصر شكراً من القلب وأسال الله أن يفتح لي ولأصدقائي أبواب الخير والتوفيق.

شيماء

المدية في : 2021-07-06

Remerciement

A l'issue du cycle de notre thèse nous tenons à remercier dieu le tous puissant.

Nos remerciements les plus sincères vont à:

Monsieur Mr.M.BOUHEDDA et Monsieur Mr. Y.ANNIBA Pour leurs conseils précieux

et leurs suivis qu'il ma prodigué

durant tout notre travail

Nos remerciements à nos amis et collègues pour leur soutien et leurs encouragements continus tout au long du parcours de la thèse.

En fin toute personne qui a participé de prés

Ou de loin à l'accomplissement de ce mémoire

Soit sincèrement remerciée et les

enseignants qui ont participé

à nos travail soient

Sincèrement remerciés.

Résumé :

Le **IdO** (Internet des Objets) est plus vaste et plus varié puisqu'elle exploite des innovante technologies et des normes que l'internet classique.

Ce travail consiste à faire la conception d'un d' acquisition des données en temps réel à base de **IdO** en utilisant le protocole **MQTT** .

L'objectif de ce travail est l'implantation de protocole **MQTT** dans les applications **IdO** et prendre en compte les valeurs transmises par les différents dispositifs situés dans l'environnement (capteur d'humidité, capteur de température), et visualiser ces données à distance sur un smart phone ou **PC**.

Les mots clés : **IdO , MQTT , PC .**

ملخص :

IdO (إنترنت الأشياء) أكبر وأكثر تنوعًا لأنه يستخدم تقنيات ومعايير مبتكرة من الإنترنت التقليدي.

يتكون هذا العمل من تصميم اكتساب البيانات في الوقت الفعلي المستند إلى إنترنت الأشياء باستخدام بروتوكول **MQTT**.

الهدف من هذا العمل هو تنفيذ بروتوكول **MQTT** في تطبيقات إنترنت الأشياء ومراعاة القيم التي تنقلها الأجهزة المختلفة الموجودة في البيئة (مستشعر الرطوبة ، مستشعر درجة الحرارة) ، وعرض هذه البيانات عن بعد على هاتف ذكي. أو الكمبيوتر الشخصي.

الكلمات المفتاحية : **انترنت الأشياء ، بروتوكول MQTT ، كمبيوتر** .

Abstract :

The IoT (Internet of Things) is larger and more varied since it uses innovative technologies and standards than the traditional internet.

This work consists of designing an IoT-based real-time data acquisition using the MQTT protocol.

The objective of this work is the implementation of the MQTT protocol in IoT applications and take into account the values transmitted by the various devices located in the environment (humidity sensor, temperature sensor), and view these data remotely. on a smart phone or PC.

Keywords: **IoT**, **MQTT**, **PC**.

LISTE DES FIGURES

Chapitre I :Généralités sur l'Internet des Objets

Figure I.1: Une nouvelle dimension pourl'IdO……...…5

Figure I.2 : Différentes Domaines d'application d'IdO ……..8

Figure I.3 : Parrot Capteur Intelligent pour Plantes Intérieur/Extérieur Résine……...8

Figure I.4 : Capteur pour plantes PARROT FLOWER POWER VERT…….9

Figure I.5 : la montre sport GPS / Cardio Garmin ForeRunner 735XT…….……10

Figure I. 6 : Le casque Oculus Go……...10

Figure I.7 : Lentille connectée Samsung……11

Figure I.8 : Architecture interne d'un microcontrôleur…….…..14

Figure I.9 : Les composants d'une carte Arduino……....15

Figure I.10: Les ports GPIO de Raspberry Pi 3…….…….…18

Figure I.11 : Les composants standards d'un Raspberry…….……..…19

Figure I.12 : Le Raspberry Pi modèle B……...…...19

Figure I.13 : principe de fonctionnement d'un capteur……...20

Figure I.14 : Le capteur DHT22……..…22

Chapitre II :Le protocol MQTT

Figure II.1 :Google Trends pour MQTT……..…….….26

Figure II.2 : Présentation de fonction du protocole MQTT…….…...28

FigureII.3 : Exemple de communication machine-à-machine suivant le protocole MQTT…….…….29

Figure II.4 : Modèle publisher/subscriber…….…….…….……30

Figure II.5 : Exemple de topic MQTT…….……...34

Figure II.6 : Le protocole MQTT dans le modèle OSI……...……...….35

Figure II.7 : Connexion MQTT……...…...35

Figure II.8 : Exemples de topologies MQTT…….38

Figure II.9 : Exemple concret d'un flux MQTT……39

Figure II.10 : Aperçu sur Node Red……..……44

Figure II.11 : Installation d'un nouveau nœud à travers 'Manage palette'……45

Figure II.12: Installation de nouveau nœud dans la palette……...45

Figure II.13 : Importation des flows……..46

Figure II.14 : Exportation des flows…….…….47

Chapitre III :Implémentation

Figure III.1 : la fenetre de Node-Red……..……50

Figure III.2 : schéma globale……....51

Figure III.3 :la réalisation de notre système ……52

Figure III.4 : le diagramme de notre projet…….…….53

Figure III.5: Lire la dernière valeur de la température et l'humidité……...54

Figure III.6 : gauge de « température"…….55

Figure III.7 : gauge de" humidité"…….……...…56

Figure III.8 : le nœud DHT11……..…….…….....57

Figure III.9 : la fonction "température "……...…….....57

Figure III.10 : la fonction " humidité"…….…….....57

Figure III.11: Affichage de température et l'humidité en temps réel……..…….....58

Figure III 12 : Affichage de température et l'humidité "debug "…….……...58

Figure III.13: Le nœud MQTT…….…….……...59

Figure III. 14: Configuration du nœud MQTT out……..…….59

Figure III.15 : acquisition des données avec le protocole MQTT……..…..…….....60

Figure III.16 : le terminal……..……...…….61

LISTE DES ABREVIATIONS

HTTP Hyper text Transfert Protocol

IdO Internet des Objets

IoT Internet of Things

MQTT Message Queuing Telemetry Transport

FRED Front End pour Node-RED

IRIS Internet Routing inSpace

Wi-Fi Wireless Fidelity

TCP Transmission Control Protocol

RFID Radio Frequency Identification

CO2 Le dioxyde de carbone

M2M Machine to Machine

POT parrot pot

GPIO General Purpose Input/Output

RPI Raspberry PI

GPS Global Positioning System

PC Personal Computer

LCD Liquid Crystal Display

IP Internet Protocol

IEEE Institute Of Electrical and Electronics Enegineers

M2S Machine to Server

S2S Server to Server

RAM Random Access Memory

ROM Read-Only Memory (mémoire morte)

EEPROM Electrically Erasable Programmable Read-Only Memory

HDML Handheld Devices Markup Language

HDMI Hight Definition Multimedia Interface

USB Universal Serial Bus

RCA Radio Corporation of America

LED Light Emitting diode

CERP-IoT Cluster des projets européens de recherche sur l'internet des objets

IBM Inernational Business Machines

UTF-8 8-Bit Universal Character Set Transformation

OSI Open System Interconnection

SSL/TLS Transport Layer Security / Secure Sockets Layer

JS Java Scrip

SUB Subscribe

SQL Structured Query Language

LISTE DES TABLEAUX

Chapitre I :Généralités sur l'Internet des Objets

Tableau I.1 : Comparaison entre les modèles du RPI…….....17

Chapitre III :Implémentation

Tableau III.2 : Comparaison entre les protocoles MQTT et http……73

TABLE DES MATIERES

INTRODUCTION GENERALE……1
CHAPITRE I : GENERALITE SUR L'INTERNET DES OBJETS…..……..3
I.1 INTRODUCTION……..……...…….4
I.2 L'INTERNET DES OBJETS……..5
I.3 L'IMPORTANCED'IDO…….……5
I.4 COMPASANTS D'INTERNET DES OBJETS ……..……..6
I.5 DOMAINES D'APLLCATION D'IDO…….……6
I.6 EXEMPLE D'OBJET CONNECTE……...8
I.7 PLATES-FORMES D'INTERNET DES OBJETS……..…….11
I.8 DEFIS D'IDO……12
I.9 COMMUNICATION DANS IDO……...…….13
I.10 ETUDE DETAILLER DES COMPOSANTS D'UN OBJET CONNECTE…….....13
I.11 CARTES ARDUINO…….....……...15
I.12 CARTE RASPBERRY PI……..……..16
I.13 CAPTEURS …….....20
I.14CONCLUSION……..…..23
CHAPITRE II : LE PROTOCOLE MQTT 45
II.1 INTRODUCTION 25
II.2 DEFINITION…….……..…28
II.3 PROTOCOLE MQTT…….……30
II.4 MODE DE FONCTIONNEMENT…….....…..37
II.5 FONCTIONNEMENT…….....38
II.6 APPLICATION NODE-RED……..43
II.7 CONCLUSION……48
CHAPITRE III : IMPLEMENTATION……49
III.1 INTRODUCTION……..50
III.2 CAHIER DE CHARGE …….……50
III.3 PRINCIPE DE FONCTIONNEMENT DE SYSTEME PROPOSE……...…54
III.4 ENVOYER LES DONNEES A DISTANT…….....63
III.5 CONCLUSION……..…….75
CONCLUSION GENERALE……...……..76

INTRODUCTION GENERALE

Le monde d'aujourd'hui se caractéris par l'avènement massif de nouvelles technologies de plus en plus fascinantes et revolutionaries. Ce sont surtout les objets connectes. Dans notre quotidien nous vivons avec ces technologies sans même parfois le savoir ; nous faisons partie ainsi de cet environnement digital qui est en plein expansion.

Le terme Internet of Things (IoT) ou Internet des objets désigne l'extension d'Internet à des objets de notre quotidien. On parle des objets connectés que l'on connaît à travers les smartwatchs, les bracelets sportifs, la domotique ou encore dans le domaine de la e-santé.

L'IdO fait partie d'une évolution puisque cela permet d'avoir de plus en plus d'objets connectés qui interagissent entre eux et qui proposent des solutions au quotidien pour améliorer les conditions de vie. Mais il s'agit également d'une révolution, et ce dans tous les domaines d'application. D'ici 2021 on prévoit 40 à 50 milliards d'objets sur terre soit l'équivalent de cinq appareils par personne.

Un nombre impressionnant quand on sait que de nombreux pays n'ont pas accès à internet. Les consommateurs y recherchent avant tout l'aspect pratique, facilitateur de la vie de tous les jours, tout en laissant place à l'émerveillement.

Ainsi la gestion de données échangées par tous ces objets connectées est une priorité absolue pour pouvoir en tirer vraiment profit. De nombreuses contraintes s'imposent tel que la gestion de l'autonomie des objets, les collectes de données et leur traitement en temps réel ainsi que le débit de transmission. Il faut donc trouver un protocole qui répond au mieux à ces attentes. C'est l'idée principale de notre travail qui s'intitule « Acquisition des données à base de l'IdO avec le protocole MQTT ».

Pour bien aborder le thème de ce présent mémoire nous allons voir en **premier chapitre** quelques généralité sur l'internet des objets. Ensuite dans le **deuxième chapitre** nous étudierons le protocole MQTT en détaille. **Le troisième chapitre** est dédié totalement au protocole de communication MQTT (Message Queuing Telemetry

Transport) qui a un rôle principal tout au long de ce travail et nous avons mesurer la température et l'humidité avec le RPI et le capteur DHT11 .

Et enfin, nous terminerons ce mémoire avec une conclusion générale et des perspectives.

Chapitre I : Généralités sur l'Internet des Objet

Chapitre I : Généralités sur l'Internet des Objets

I.1 INTRODUCTION

L'internet a connu depuis son apparition, un grand essor et a touché pratiquement tous les domaines de notre vie de tous les jours : politique, économique, socioculturel,…etc. Elle est devenue la principale source d'information.

L'internet a également facilité la Communication entre les gens, en transformant le monde en une petite ville.

Le progrès scientifique et technologique a eu, spécialement dans le domaine de l'électronique, de la communication des objets intelligents et des systèmes de capteurs très développés, variés et de différentes tailles, des répercussions positives sur l'internet et l'apparition également d'un nouveau paradigme qui est « l'Internet des Objets », en anglais : Internet of Things (IoT).

L'IdO constitue la prochaine génération de la révolution numérique. Ce sont des milliards d'objets du quotidien qui seront connectés à l'horizon 2020. Les objets connectés ont la possibilité de simplifier la vie quotidienne et d'améliorer le bien-être des individus, chez eux, dans leur jardin privé, au bureau, en famille ou au sein de toute autre communauté[1]..

En outre, l'Internet des Objets a la possibilité d'intégration avec le réseau Internet, ce qui va permettre une connectivité pour tout le monde, tout le temps et partout et idéalement depuis n'importe quelle plate-forme [2].

Ce chapitre est consacré à l'étude des systèmes d'IoT. L'objectif de telle étude est de comprendre leurs caractéristiques et spécificités afin de pouvoir développer un système d'IdO assurer le transfert des données à base de protocole MQTT

Finalement , nous allons introduire et présenter la carte électronique embarquée que nous utiliserons dans l'élaboration de notre système d'acquisition des données climatiques: la carte Raspberry Pi.

I.2 L'INTERNET DES OBJET

I.2.1 définition de l'internet des objets

l'Internet des objets (IdO) est une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physique ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ».

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée (adresse IP, protocoles smtp, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wifi. [3] (figure I.1).

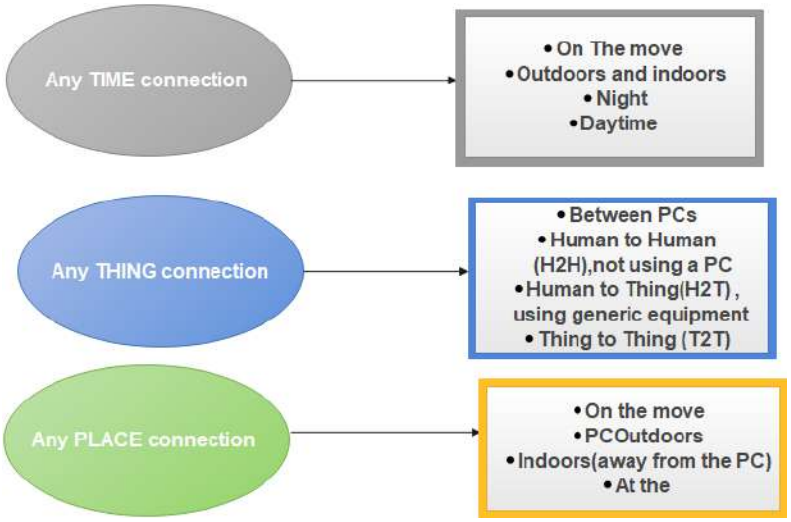


Figure I.1: Une nouvelle dimension pour l'IdO.

I.3 L'IMPORTANCE D'IdO

L'importance de l'IdO devient considérable, puisqu'il s'agit de la première véritable évolution de l'Internet.

Celle-ci donnera lieu à des applications révolutionnaires capables de transformer profondément notre mode de vie, et notre façon d'apprendre, de travailler et de nous divertir. L'IdO a

déjà doté l'Internet de capacités sensorielles (température, pression, vibration, luminosité, humidité, tension), ce qui nous permet d'anticiper plutôt que de simplement réagir.

En outre, l'Internet couvre maintenant des endroits jusqu'alors inaccessibles. Des patients ingèrent même des dispositifs connectés qui aident les médecins à diagnostiquer certaines pathologies et à en déterminer les causes , des capteurs extrêmement miniaturisés peuvent être placés sur des plantes, des animaux et des sites géologiques, et connectés à l'Internet , dans le même temps, l'Internet est en train de conquérir l'espace grâce au programme IRIS de Cisco [1].

I.4 COMPOSANTS D'INTERNET DES OBJETS

Le concept d'Internet des Objets exige la coordination des dispositifs suivants [4] :une étiquette physique identifie chaque objet / une étiquette virtuelle identifie chaque lieu .

un dispositif mobile (téléphone cellulaire, organiseur, ordinateur portable…) doté d'un logiciel additionnel, lit les étiquettes physiques ou localise les étiquettes virtuelles;
un réseau sans fil relie le dispositif portable à un serveur contenant l'information relative à l'objet étiqueté;
les informations sur les objets sont gérées dans des pages existantes du web;
un dispositif d'affichage (écran d'un téléphone mobile) permet de consulter les informations relatives à l'objet ou à un ensemble d'objets.

I.5 DOMAINES D'APPLICATION D'IDO

L'IdO couvrira un large éventail d'applications et touchera quasiment à tous les domaines que nous affrontons au quotidien, ceci permettra l'émergence d'espaces intelligents (Figure I.2). Parmi ces espaces intelligents, on peut citer :

▪ **Les villes intelligentes (Smart Cities)** [5] **:** l'IdO permettra une meilleure gestion des réseaux divers qui alimentent nos villes (eaux, électricité, gaz, etc.) en permettant un contrôle continu en temps réel et précis. Des capteurs peuvent être utilisés pour l'économie de l'eau et pour améliorer la gestion des parkings et du trafic urbain et diminuer les embouteillages et les émissions en CO2.

▪ **La santé (Smart Health)** [5] **:** dans le domaine de la santé, l'IdO permettra le déploiement de réseaux personnels pour le contrôle et le suivi des signes cliniques, notamment pour des personnes âgées, les objets connectés permettent de suivre la tension, le rythme cardiaque, la qualité de respiration ou encore la masse graisseuse. Ceci permettra ainsi de faciliter la télésurveillance des patients à domiciles, et apporter des solutions pour l'autonomie des personnes à mobilité réduite.

▪ **Le Transport** [6] **:** Des voitures connectées ou autonomes aux systèmes de transport/logistique intelligents, l'IdO peut sauver des vies, réduire le trafic et minimiser l'impact des véhicules sur l'environnement.

▪ **L'énergie** [6]] **:** 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 efficaces de l'énergie.

▪ **L'industrie** [5] **:** La technologie IdO permettra un suivi total des produits, de la chaîne de production, jusqu'à la chaîne logistique et de distribution en supervisant les conditions d'approvisionnement. Cette traçabilité de bout en bout permet aux usines d'améliorer l'efficacité de ses opérations, d'optimiser la production et d'améliorer la sécurité des employés

▪ **Le bien-être et le confort** [7] **:** La domotique ou la maison intelligente est un classique. Imaginez un instant que votre thermostat soit capable de se mettre en marche tout seul en fonction de l'emplacement de votre voiture vous permettant de vous réchauffer une fois rentré à la maison. Aussi, imaginez que votre réfrigérateur vous informe lorsque vous aurez besoin d'acheter du lait ou qu'il soit capable de créer une liste d'achats personnalisée en fonction de vos articles les plus achetés. Ou encore vous dire quand votre nourriture est sur le point de périmer.

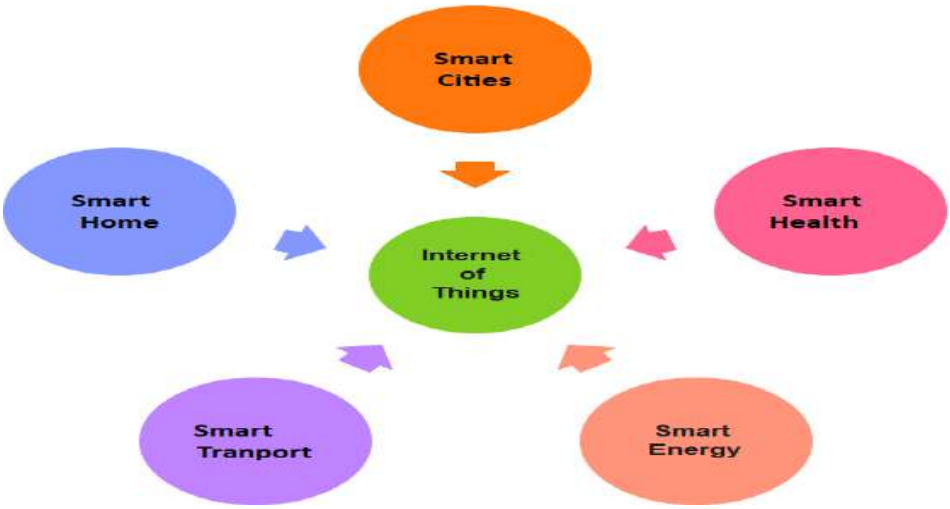


Figure I.2 : Différentes Domaines d'application d'IdO [4].

I.6 EXEMPLES D'OBJETS CONNECTES

I.6.1 parrot pot (pot de fleurs intelligent et connecté pour plantes)

Un pot de fleur connecté qui s'occupe des plantes ! Doté de divers capteurs et d'un réservoir d'eau, le POT pourra entretenir les même fleurs lors d'absences du domicile. L'application donnera des conseils d'entretien et des rappels divers. Un gadget idéal pour les personnes qui n'ont pas forcément la main verte [8] (figure I.3).



Figure I.3 : Parrot Capteur Intelligent pour Plantes Intérieur/Extérieur Résine [8] [9].

I.6.2 flower power

Parrot, célèbre pour ses drones, œuvre aussi dans le jardinage avec le Flower Power. Ce capteur, au look de lance-pierre, se plante dans la terre (en pot ou dans le sol) juste à côté de la plante à surveiller. Le Flower Power mesure quatre paramètres : la luminosité, la température, l'engrais et l'humidité. Lorsque votre téléphone est à proximité du capteur (et avec le Bluetooth activé), les données récoltées sont synchronisées, et vous êtes informé : elle manque un peu d'eau, d'engrais, la température est bonne, etc. [10]. (figure I.4) .



Figure I.4 : Capteur pour plantes PARROT FLOWER POWER VERT [10].

I.6.3 montre sport (garmin forerunner735xt)

La montre sport (Garmin Forerunner 735XT) est une montre destinée aux sportifs et notamment aux coureurs. Elle offre une autonomie allant jusqu'à 11 heures et dispose d'un GPS et cardio-fréquencemètre intégré, le tout piloté via un écran tactile. Le point fort de cette Forerunner est toutefois la possibilité d'organiser des entraînements comme pour des professionnels. Si elle parlera aux adaptés de running avec l'analyse de la foulée, les nageurs et les cyclistes pourront compter sur des modes permettant d'analyser la nage et le pédalage. On retrouvera également les suivies des VO2 Max et des aides à la récupération [11].(figure I.5).



Figure I.5 : la montre sport GPS / Cardio Garmin Forerunner 735XT [11].

I.6.4 casque oculusgo

pas besoin de PC ou de Smartphone pour faire fonctionner le nouveau casque de Facebook, au contraire des concurrents HTC Vive ou Samsung Gear VR par exemple. L'Oculus Go propose un champ de vision à 110 degrés et un écran LCD doté d'une définition 2560 x 1440 pixels. Il intègre des haut-parleurs qui offrent un son "spatialisé" et une prise jack pour le casque audio. Accompagné de son contrôleur, il sera vendu 199 dollars. **Sortie prévue :** "Au début de l'année 2018 pour les États-Unis" selon Facebook. Sûrement un peu plus tard dans l'Hexagone [12] (figure I.6).



Figure I. 6 : Le casque Oculus Go [12].

I.6.5 lentilles connectées samsung

C'est là un gros projet que Samsung s'est mis en tête de réaliser. Mais étant donné que le projet ne reste pour l'instant qu'un simple brevet que la marque a déposé dans le but de ne pas se faire voler son idée, il faudra encore laisser ces lentilles quelques temps dans le domaine de la science-fiction… En effet, nombreuses sont les marques ayant déposé un nombre invraisemblable de brevets sans jamais les avoir vu passer au stade de production ou même de développement.[14].(figure I.7).

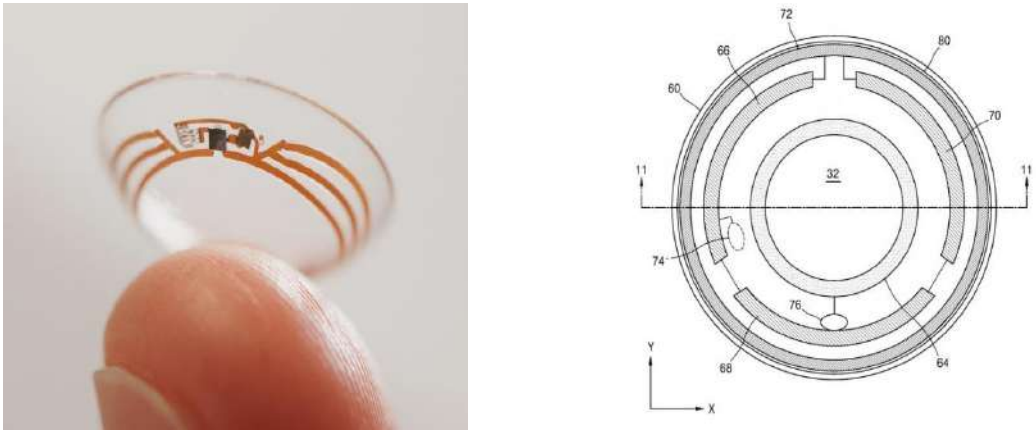


Figure I.7 : Lentille connectée Samsung [13].

I.7 PLATES-FORMES D'INTERNET DES OBJETS

Avec l'explosion de l'internet des objets, de nombreux projets IdO ont vu le jour et on assiste à l'émergence de nombreux développements et solutions IdO. L'internet des objets constitue un ensemble d'objets connectés qui utilisent le réseau comme support pour échanger des données. Ces objets sont divers et peuvent être des cartes (ex : Arduino), des logiciels ou des capteurs.

L'existence de ce vaste réseau nécessite une ou plusieurs plateformes qui servent de support. De plus, avec des cartes microprocesseurs comme Arduino ou Raspberry Pi devenus accessibles et bon marché, nombreux ont voulu dans le développement de

projets IdO. C'est dans cette lancée que de nombreuses entreprises ont créé des plateformes IdO pour les développeurs ou tout simplement des amateurs qui sont intéressés par des projets IdO [15].

Le marché des plateformes IdO est vaste comme un océan! Aujourd'hui, l'écosystème IdO détient plus de 300 entreprises de la plateforme IdO [16], Alors qu'il est devenu d'autant plus nécessaire de connaître correctement les noms des meilleures plates-formes IdO sur le marché. Ce qui est important pour nous, c'est de jeter une œil exhaustive sur chacun d'entre eux, de les explorer et de mettre les mains sur les points et les caractéristiques les plus importants afin de pouvoir choisir la plate-forme correspondant à notre projet et nos moyens matérielles modestes.

I.8 DEFIS D'IDO

Plusieurs obstacles pourraient toutefois ralentir la progression de l'IdO, notamment le -déploiement du protocole IPv6, l'alimentation des capteurs et la définition de normes [1].

▪ **Le déploiement du protocoleIPv6:** Nous avons atteint le nombre maximal d'adresses IPv4 en février 2010. Si cela n'a pas eu d'incidence visible pour le grand public, le développement de L'IdO pourrait s'en trouver ralenti, puisque chacun des milliards de nouveaux capteurs potentiels devra avoir sa propre adresse IP. En outre, le protocole IPv6 facilite la gestion des réseaux grâce à des fonctions de configuration automatiques, et propose des fonctions de sécurité améliorées.

▪ **L'alimentation des capteurs :** Pour que l'IdO puisse dévoiler tout son potentiel, les capteurs devront être autosuffisants. Imaginez qu'il faille changer les piles de milliards d'appareils déployés aux quatre coins de la planète et même dans l'espace. Cela serait évidemment impossible. Nous devons donc trouver un moyen de générer de l'électricité en puisant dans l'environnement, par exemple en utilisant les vibrations, la lumière et les courants d'air.

▪ **Les normes :** D'énormes progrès ont déjà été accomplis dans le domaine des normes, mais le chemin à parcourir est encore long, notamment dans les domaines de la sécurité,

de la confidentialité, de l'architecture et des communications. Tout comme l'IEEE, de nombreux organismes s'efforcent de relever ces défissent s'assurant que les paquets IPv6 peuvent être acheminés sur différents types de réseau.

I.9 COMMUNICATION DANS IDO

En IdO les objets peuvent communiquer avec les autres (M2M) « machine to machine » ou recueillir des données et envoyé à l'infrastructure de serveur (M2S) « machine to server ». Cette infrastructure de serveur peut partager les données avec autre serveur (S2S) « server to server », ou les objets pour analyser les données. Comme le énorme nombre d'objets augment´e par l'année, trouver un bon protocole pour le réseau est toujours un grand défi.

Dans cette section, on va avoir une vue ensemble sur quelques protocole pour la communication entre les objets et serveur. L'analisation sur leurs points forts et les points faibles pourrait nous donner une solution possible pour notre objet .

I.10 ETUDE DETAILLEE DES COMPOSANTS D'UN OBJET CONNECTE

I.10.1 le microcontrôleur

I.10.1.1 définition d'un microcontrôleur

Un microcontrôleur se présente sous la forme d'un circuit intégrés qui contient en interne, l'équivalent de la structure complète d'un micro-ordinateur. C'est-à-dire qu'il rassemble dans un même boitier un microprocesseur, plusieurs types de mémoires et des périphériques de communication (Entrées-Sorties) [17].

Les microcontrôleurs améliorent l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré. Nous parlons alors de « système sur une puce » en anglais : « System on Chip »[18].

I.10.1.2 domaine d'application

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, de multiples industries et applications, notamment dans les domaines de l'automatisation, de la fabrication, de la robotique, de l'automobile, de l'éclairage, de l'énergie intelligente, de l'automatisation industrielle, des communications et de l'internet des objets [18] (figure I.8).

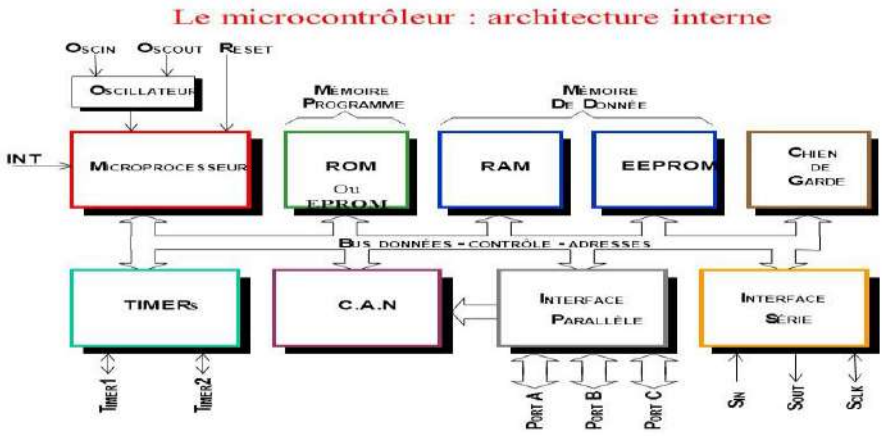


Figure I.8 : Architecture interne d'un microcontrôleur [19].

I.10.1.3 critère de choix d'un microcontrôleur

Les critères principaux que nous devons retenir pour choisir un microcontrôleur sont les suivants [17] :

- le ou les circuits de la famille doivent être facilement disponibles sur le marché.
- le prix des circuits doit être à la portée .
- la programmation de la mémoire morte interne doit être facile.
- il est important de considérer la vitesse maximale, la quantité de RAM ou de ROM.
- la consommation d'énergie et les contraintes et le soutien au développement.

I.10.1.4 différentes modèles de cartes à microcontrôleur

Cette catégorie de cartes renferme un microcontrôleur complet ; c'est à dire l'association en un seul circuit d'une unité centrale de microprocesseur, de mémoire morte, de mémoire vive, de mémoire EEPROM, d'une interface d'entrée/sortiesérie et de toute la logique nécessaire pour faire fonctionner tout ce la.

Il existe plusieurs types de cartes à microcontrôleur, nous citons à titre d'exemple juste les deux suivants : Arduino et RaspberryPi.

I.11 CARTES ARDUINO

Les cartes Arduino possèdent un microcontrôleur facilement programmable ainsi que de nombreuses entrées-sorties. Plusieurs cartes Arduino existent et qui se différencient par la puissance du microcontrôleur ou par la taille et la consommation de la carte.

Le choix du type de la carte Arduino s'effectue en fonction des besoins de votre projet. La carte Arduino UNO est la carte la plus couramment utilisée qui constitue un bon choix pour les débutants.

L'ensemble des cartes Arduino se programment en C++ à l'aide d'un logiciel de programmation gratuit et open-source fourni par Arduino [13] (figure I.9).

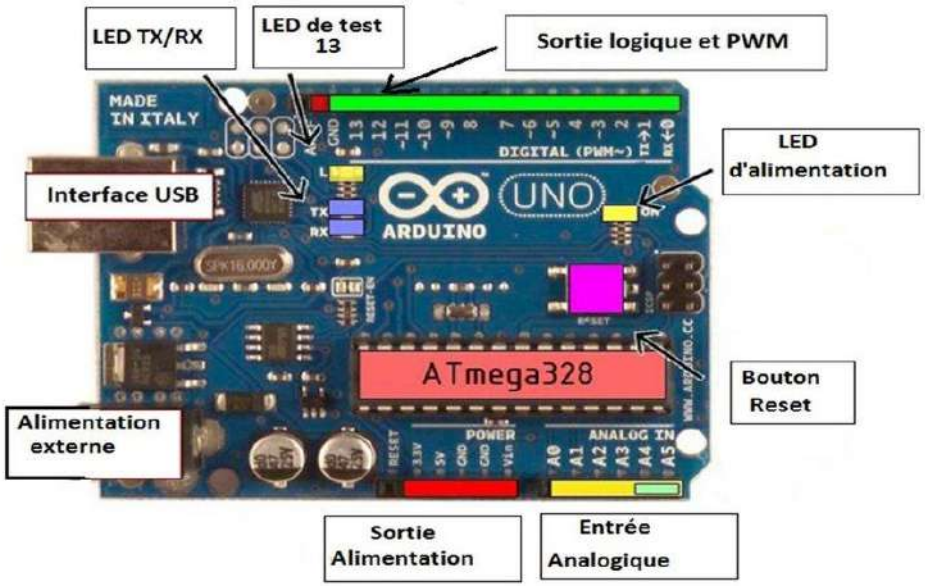


Figure I.9 : Les composants d'une carte Arduino.

I.12 CARTES RASPBERRY PI

Nous présenterons dans la suite, le Raspberry Pi que l'on a utilisé dans notre projet. Ce dernier est considéré comme le cerveau de notre système parce que joue le rôle d'un serveur Web affichant en temps réel plusieurs valeurs stockée dans une base de données.

I.12.1 présentation du Raspberry

Le Raspberry Pi est en effet un mini-ordinateur, c'est un petit ordinateur mono carte capable de faire fonctionner des systèmes d'exploitation comme Linux et même une version réduite de Windows 10. Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch, etc.).

Il dispose de ports USB, d'un connecteur Ethernet, d'un port HDMI et de sortie audio, tout comme l'ordinateur de bureau. Le Raspberry pi destiné à des applications d'informatique embarquée. Le cœur de Raspberry est un FPGA (Broadcom 2835) intégrant un processeur ARM11 cadencé à 700MHz et de nombreux périphériques [20].

I.12.2 spécifications matérielles et architectures

Il y a eu plusieurs modèles différents du Raspberry Pi depuis sa conception:

- **Modèle A**: Utilisez ce modèle pour un projet à faible coût qui nécessite un ordinateur complet sans capacités de mise en réseau et un support E/S décent
- **Modèle B:** Ce modèle peut être utilisé pour un projet où le prix n'est pas un problème et où le Pi le plus puissant est nécessaire. Ce modèle contient également des E/S faciles à utiliser, il convient donc aux premiers projets Pi.
- **Model Zéro**: Ce modèle est idéal pour un projet à très faible coût, dans un espace restreint, qui nécessite un ordinateur en parfait état de fonctionnement et qui bénéficierait d'une connectivité sans fil.

Tableau I.1 : Comparaison entre les modèles du RPI.

	Générations Zéro	**Générations 1**		**Générations 2**	**Générations 3**	**Générations 4**
Modèle	Zéro	A	B	2B	3A+	4B
RAM	512 Mo	256 Mo	512 Mo	1 Go	512 Mo	1Go / 2Go / 4Go / 8Go
Processor	1GHz ARM11	700 MHz ARM11	700 MHz ARM11	900 quadricoeur ARM Cortex-A7	1,4 quadricoeur ARM Cortex-A53	1,5 quadricoeur ARM Cortex-A72
Sorties Vidéo	Mini HDMI et Composite via soudures	HDMI et Composite via une Price RCA		HDMI et Composite (via un connecteur Jack		2 Micro HDMI et Composite via un connecteur Jack
Sorties Audio	Mini-HDMI	stéréo Jack 3,5 sortie son 5.1 sur la prise HDMI				
USB	1 Micro-USB 2.0	1 (2.0)	2 (2.0)	4 (2.0)	1 (2.0)	2 (2.0) 2 (3.0)
Poids	9 g	45 g	45 g	45 g	29 g	46 g
Réseau	non		10/100 Mbits/s Ethernet		Wifi 802.11 ac Bluetooth 4.2	Ethernet Wifi Bluetooth

I.12.3 les composants standards de raspberry

- **Port USB :** est un port série qui sert à connecter le raspberry aux autres périphériques ou pour connecter une souris et un clavier.
- **Emplacement pour carte SD :** le raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe. C'est ici que sont stockés le système d'exploitation et tes fichiers.
- **Port Ethernet :** c'est un port qui correspond au protocole international ETHERNET de réseau local à communication de paquets.
- **Prise audio ou Une prise jack** : c'est une connectique femelle stéréo de 3.5 mm de diamètre destiné à la connexion d'un casque, d'un pair de haut-parleurs amplifiés comme ceux que l'on utilise avec les PC, ou bien encore d'un amplificateur externe.
- **Port HDMI :** « Hight Definition Multimedia Interface » permet de relier le Raspberry Pi à un dispositif compatible (écran LCD ou un vidéoprojecteur),capable de surcroît de fournir de la vidéo au format full HD de 1080p, et véhicule également le son.
- **Port RCA :** « Radio Corporation of America », véhicule quant à lui le signal vidéo généré par le Raspberry Pi au format vidéo composite PAL ou NTSC et ne permet donc qu'un affichage en basse résolution, sur un vieux moniteur à tube cathodique de surcroit.
- **Connecteur d'alimentation Micro USB :** pour connecte à une alimentation électrique. Toujours faire ceci en dernier, après avoir connecté tous tes autres composants.
- **Ports GPIO : «** General Purpose Input/Output **»** sont des ports d'Entres/Sorties qui nous permettent de connecter des composants électroniques tels que des LEDs et des boutons au Raspberry Pi.(figure I.10)

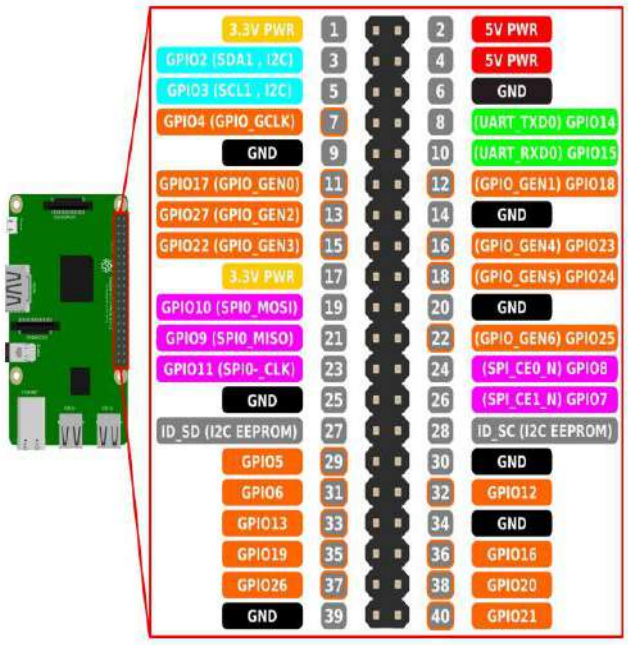


Figure I.10: Les ports GPIO de Raspberry Pi 3.

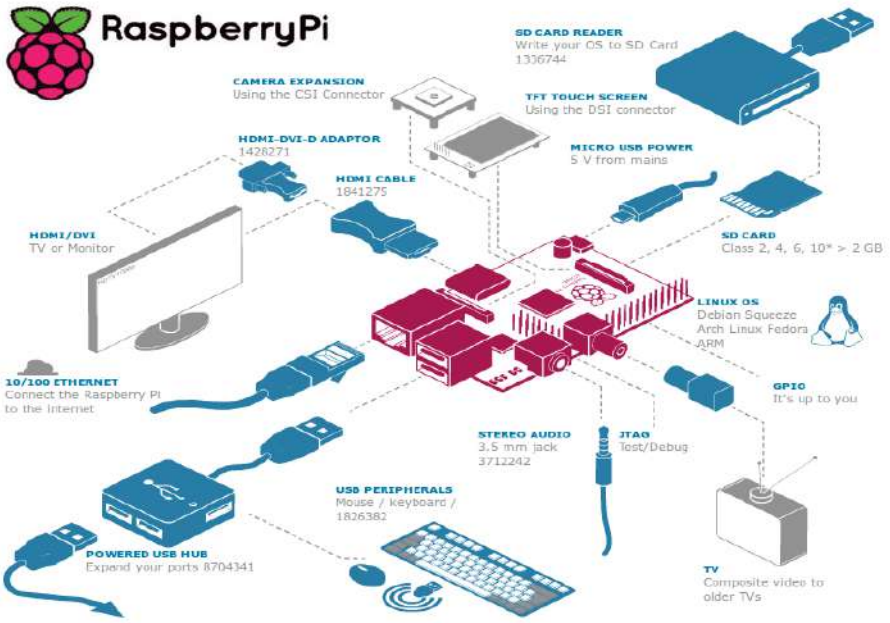


Figure I.11 : Les composants standards d'un Raspberry .

Dans notre projet , nous utiliserons le Raspberry Pi modèle B qui est représente dans la figure ci-dessous (figure I.12) .

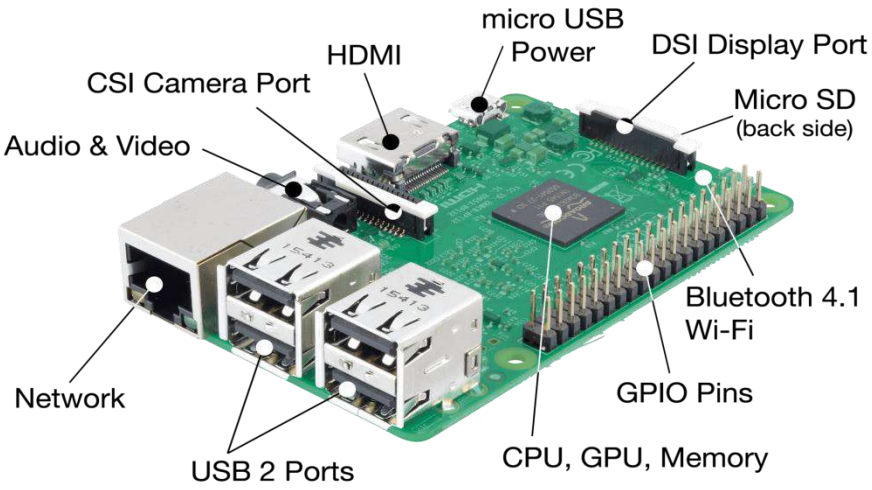


Figure I.12 : Le Raspberry Pi modèle B.

I.13 CAPTEURS

I.13.1 définition d'un capteur

Un capteur est un dispositif qui permet de convertir une grandeur physique observée (température, luminosité, humidité, débit, présence d'objet,…) en une grandeur normée et utilisable, généralement un signal électrique (courant, tension, niveaux logiques, valeur moyenne, fréquence, amplitude, nombre binaire,…), qui peut être interprétée par un dispositif de contrôle commande [21] (figure I.13)

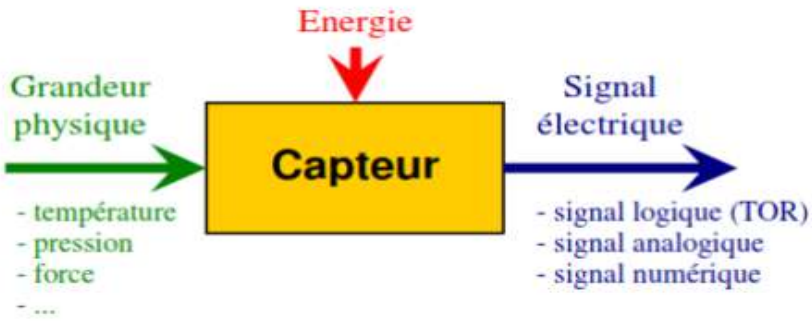


Figure I.13 : principe de fonctionnement d'un capteur [21].

Les capteurs jouent un rôle indispensable dans le domaine d'IdO, ils peuvent être classés selon deux critères à savoir [21] :

- **Grandeur mesurée :** nous distinguons par exemple un capteur de température, de position, d'humidité, de vitesse, de force, de pression,..etc.

- **Le signal délivré :** nous trouvons par exemple des capteurs logiques (appelés aussi capteurs TOR), capteurs analogiques et numériques
- **Capteurs logiques :** La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs exemple: capteurs de fin de course.

La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique

- **Capteurs analogiques :** mesurée par le capteur.
- **Capteur numérique :** La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type : code numérique binaire, train d'impulsions, etc.

I.13.2 caractéristiques des capteurs

Les capteurs sont distingués par les caractéristiques suivantes [22] :

- **Étendue de mesure (Pleine Echelle) :** L'étendue de mesure est la différence entre la limite supérieure et la limite inférieure de la grandeur mesurable par un capteur. Lorsque le capteur fournit une valeur de la grandeur entre 0 et le maximum, ce maximum est appelé « Pleine Echelle »
- **Résolution :** plus petite variation de grandeur mesurable par le capteur
- **Rapidité :** temps de réaction du capteur. La rapidité est liée à la bande passante.
- **Précision :** Aptitude du capteur à donner une mesure proche de la valeur vraie
- **Sensibilité :** représente la variation du signal de sortie par rapport à la variation du signal d'entrée.
- **Linéarité :** représente l'écart de sensibilité sur l'étendue de mesure
- **Finesse :** Elle permet d'estimer l'influence que peut avoir le capteur et de son support ou de ses liaisons sur la grandeur à mesurer. Par exemple, dans le cas d'un capteur de température, une capacité calorifique importante réduit sa finesse.

▪ **Fréquence de résonance :** Un capteur possède une réponse qui peut dépendre de la fréquence de la grandeur mesurée. Lorsqu'il existe une fréquence à laquelle la réponse est particulièrement élevée, celle-ci est appelée fréquence de résonance.

I.13.3 capteur de température et d'humidité :

Le DHT22 est un capteur numérique de base, à faible coût permettant de mesurer de manière efficace la température et l'humidité de l'air ambiant grâce à sa combinaison deux en un d'un capteur d'humidité capacitif et d'une thermistance (figure I.14).

De plus il se connecte facilement à votre microcontrôleur préféré grâce à un signal numérique sur un un seul fil [23]

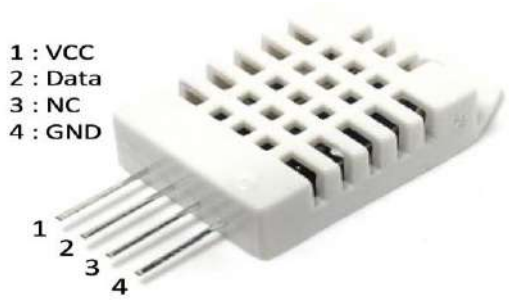


Figure I.14 : Le capteur DHT22.

I.13.4Caractéristiques techniques:

Tension d'alimentation: 3 à5V

Plage de température: -40 à+80

Humidité de 0 à100% RH

Dimension: 15.1*25*7.7mm

I.14 CONCLUSION

Dans ce chapitre, nous avons présenté les différents concepts, composants et les principales caractéristiques liées à l'internet des objets. Nous avons aussi éclairé la majorité des points essentiels qui vise la réalisation de notre projet.

En outre, nous avons montré que cette nouvelle technologie a contribué de manière significative et efficace au développement de la vie humaine, car elle est en fait partie intégrante dans divers domaines. aussi nous avons pris les différents outils hardware (Raspberry et Arduino).

Dans ce projet nous choisissons MQTT grâce – aux performance et on faire une étude détail dans le prochaine chapitre .

Chapitre II :

Le Protocole MQTT

Chapitre II : Le protocol MQTT

II.1 INTRODUCTION

MQTT est un protocole de transport de messagerie de publication / abonnement Client Server. Il est léger, simple et conçu de façon à être facile à mettre en œuvre. Ces caractéristiques le rendent idéal pour une utilisation dans de nombreuses situations, y compris des environnements contraints tels que pour la communication dans les contextes Machine to Machine (M2M) et Internet of Things (IoT) où une petite empreinte de code est requise et / ou la bande passante du réseau est à la prime. Créé en 1999 par le Dr Andy Stanford-Clark d'IBM (International Business Machines Corporation), et Arlen Nipper d'Arcom (maintenant Eurotech) MQTT est le protocole qui joue un rôle important dans l'Internet des objets. D'où l'intérêt de bien le connaître.

En août 2010, IBM a publié le protocole MQTT V3.1 spécification, qui a été utilisé dans le développement de futures normes. Eclipse, et plus précisément le projet Eclipse Paho, a été le premier à déplacer MQTT sur son chemin de normalisation. En octobre 2014, MQTT est devenu une norme OASIS ,lorsque leur cas d'utilisation était de créer un protocole avec les objectifs suivants : [33]

- Simple à mettre en œuvre.
- Fournir une qualité de service de livraison de données.
- 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**. : [34].

Ces objectifs sont toujours au cœur de MQTT tandis que l'attention s'est déplacé des systèmes intégrés propriétaires aux cas d'usage de l'Internet des objets. Il y a souvent une confusion quant à la signification appropriée de l'abréviation MQTT. La réponse courte est que MQTT n'a officiellement plus d'acronyme, c'est juste MQTT. La réponse longue histoire raconte que l'ancien acronyme signifiait "Transport de télémétrie MQ" (Message Queuing Telemetry Transport), alors que MQ fait référence à "MQ Series", un

produit développé par IBM qui prend notamment en charge MQTT. Le protocole a été nommé après quand il a été inventé 1999.

Il devient un standard ISO (ISO/IEC 20922) en 2016 [35].

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(FigureII.1)

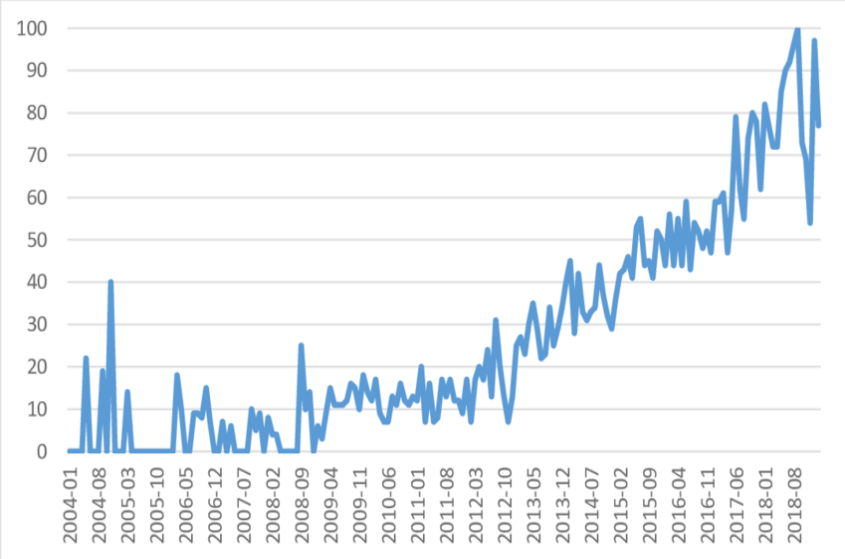


Figure II.1 :Google Trends pour MQTT.

MQTT, une norme de base sur l'Internet des Objets (IdO) développée par le consortium OASIS, a été approuvée pour publication par l'Organisation internationale de normalisation (ISO) et la Commission électrotechnique internationale (IEC) en janvier 2016. La version 3.1.1 du MQTT a été votée par l'entremise du Comité technique mixte sur les technologies de l'information (JTC1) de l'ISO et de la IEC et a reçu la désignation «ISO / IEC 20922 ».

Les qualités de MQTT font de lui un excellent candidat pour les communications au sein de l'Internet des Objets [36], mais il a également été utilisé avec succès par Facebook pour être au cœur de leur système de messagerie à cause de sa performance [37], comme il est utilisé par Amazon web services dans ses passerelles pour permettre la diffusion des données à plusieurs abonnés sur un sujet donné [38].

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 DEFINITIONS

II.2.1 Message Queuing Telemetry Transport

MQTT (Message Queuing Telemetry Transport) [39] est un autre exemple de protocole applicatif de messagerie sur le web, dont son efficacité est de plus en plus approuvée dans de célèbres applications comme la messagerie sur le réseau social Facebook. Avec un mécanisme de communication suffisamment simple pour mieux répondre aux fortes contraintes des réseaux de capteurs connectés à Internet.

Un protocole Publish/Subscribe asynchrone qui fonctionne avec TCP. C'est un protocole M2M (Machine to Machine) largement utilisé dans l'IOT. Le protocole MQTT est un protocole à message extrêmement léger. C'est pour cette raison qu'il est adopté dans l'écosystème IOT. Presque toutes les plates formes IOT prennent en charge le protocole MQTT pour envoyer et recevoir des données à partir des objets intelligents.

Trois niveaux de qualité de service sont prévus pour la transmission des messages :

- Fire and Forget : le message est envoyé une fois et aucun acquittement n'est exigé.
- Delivered at least once : le message est envoyé au moins une fois et un acquittement est exigé.
- Delivered exactly once : un mécanisme en quatre temps assure la délivrance du message une seule et unique fois.

La figure II.2. suivante présente la fonction du protocole MQTT.

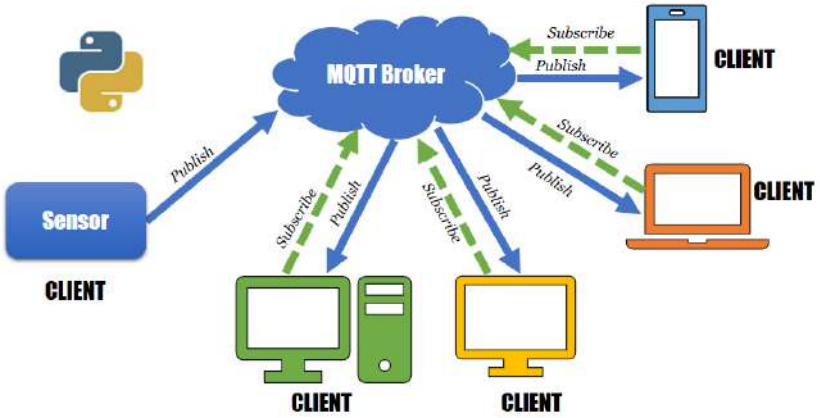


Figure II.2 : Présentation de fonction du protocole MQTT [40].

Du point de vue architectural, les nœuds capteurs sont des publieurs qui se connectent tous à une entité centrale dite broker. Chaque message est publié au niveau du broker dans une rubrique qui convient au type de la donnée contenue dans le message. Les abonnés peuvent souscrire à plusieurs rubriques. A chaque fois qu'un nouveau message est publié dans l'une des rubriques d'intérêts, il sera immédiatement diffusé aux abonnés intéressés. Dans la figure ci-dessous, on illustre un exemple de l'architecture de communication M2M dans MQTT. Le modèle est composé de deux abonnés qui élaborent des connexions TCP avec le broker. L'abonné numéro 1 adhère à la rubrique température ensuite, l'abonné 2 publie un nouveau message dans cette rubrique. Le broker se met, immédiatement, à transmettre une copie du message au client 1.

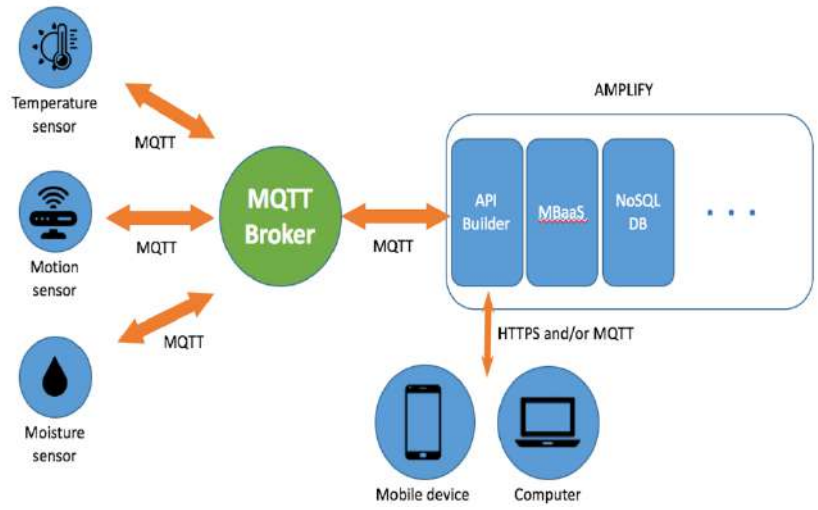


Figure II.3 : Exemple de communication machine-à-machine suivant le protocole MQTT.

L'efficacité du protocole MQTT a été approuvée pour certaines applications comme :

Les applications médicales et industrielles. Il s'est avéré même qu'il représente une bonne solution pour les applications mobiles dans des environnements contraints car il présente une faible empreinte mémoire, une faible consommation d'énergie avec une meilleure distribution de l'information aux destinataires.

Il y a aussi d'autre modèle de protocole tel que le modèle XMPP, le modèle JMS, le modèle AMQP, etc.

II.3 PROTOCOLE MQTT

II.3.1 présentation

MQTT permet concrètement aux appareils d'envoyer des informations sur un sujet donné à un serveur qui fonctionne comme un courtier de messages appelé broker. Le broker pousse ces informations vers les clients qui se sont précédemment abonnés.

Pour l'utilisateur, un sujet ressemble à un chemin hiérarchique. [41] Les clients peuvent s'abonner à un niveau spécifique de la hiérarchie d'un sujet ou à plusieurs niveaux s'ils utilisent un caractère générique.

II.3.1.1 modèle de publisher / subscriber

Le modèle publish / subscribe est une alternative au modèle client-serveur traditionnel, où un client communique directement avec un terminal.
Avec ce pattern, chaque application n'a pas besoin de connaître les autres applications avec lesquelles elle communique. Les messages ne sont pas envoyés d'un émetteur à un récepteur, mais sont échangés sur des files d'attente (les « topics ») :

Une application réceptrice peut s'abonner (« subscribe ») à un topic pour recevoir les messages associés, sans forcément connaître les applications qui vont émettre dessus. De la même manière, une application émettrice va envoyer (« publish ») un message dans un topic sans connaître les applications qui se sont abonnées. Les abonnés, s'il y en a, sont alors notifiés et reçoivent le message.

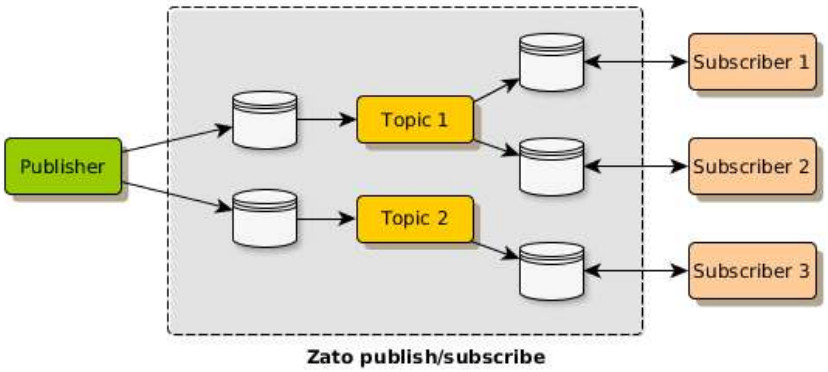


Figure II.4 : Modèle publisher/subscriber.

Le modèle pub / sub est le découplage du publisher (ou celui qui publie le message sur un topic) et le récepteur, qui peut être différencié en plusieurs dimensions :

- Découplage de l'espace: le publisher et le subscriber n'ont pas besoin de se connaître(par adresse IP et port par exemple)
- Découplage de temps: le publisher et le subscriber n'ont pas besoin de s'exécuter en même temps.
- Découplage de synchronisation: les opérations sur les deux composants ne sont pas arrêtées pendant la publication ou la réception.

Donc, ce qui est intéressant, c'est comment le broker filtre tous les messages, de sorte que chaque subscriber reçoit uniquement les messages qui l'intéressent. Le filtrage par sujet est basé sur un sujet ou « topic », qui fait partie de chaque message. Le client récepteur s'abonne sur les sujets auxquels il est intéressé à travers le broker et à partir de là il reçoit tous les messages publiés sur le sujet donné. Les « Topics » sont en général des chaînes de caractères avec une structure hiérarchique que nous détaillerons un peu plus tard.

II.3.1.2 les spécifications

La spécification identifie un certain nombre de fonctions distinctives du protocole :

- Il s'agit d'un protocole de type publication/abonnement.
- Outre la distribution de type "un à plusieurs" des messages, la fonction de publication/abonnement découple les applications. Ces deux fonctions sont importantes dans les applications qui ont beaucoup de clients.
- Il ne dépend aucunement du contenu des messages.
- Il fonctionne sur TCP/IP, qui fournit la connectivité réseau de base.
- Il propose trois qualités de service pour la distribution des messages que nous verrons plus tard.
- Il dispose de la fonction « Dernières volontés et testament » qui notifie les abonnés de la déconnexion anormale d'un client du serveur MQTT.

II.3.1.3 les avantages du protocole

Il est idéal pour répondre aux besoins suivants :

- Protocole ouvert, simple, léger et facile à mettre en œuvre.
- Idéal pour l'utilisation sur les réseaux sans fils.
- Particulièrement adapté pour utiliser une très faible bande passante.
- Faible consommateur en énergie.
- Très rapide, il permet un temps de réponse supérieur aux autres standards du web actuel.
- Permet une forte fiabilité si nécessaire.
- Nécessite peu de ressources processeurs et de mémoires.

II.3.2 les composants du protocole

II.3.2.1 client

En parlant d'un client, cela signifie presque toujours un client MQTT. Cela inclut le publisher ou les subscribers, les deux étiqueter un client MQTT qui ne font que publier ou s'abonner (En général, un client MQTT peut être à la fois publisher et subscriber en même temps). Un client MQTT est un périphérique à partir d'un microcontrôleur jusqu'à un serveur à part entière, qui a une bibliothèque MQTT en cours d'exécution et se connecte à un broker MQTT sur tout type de réseau. Il peut s'agird'un périphérique réellement limité et restreint en ressources, connecté à un réseau sans fil et doté d'une bibliothèque au minimum ou d'un ordinateur typique exécutant un client MQTT graphique à des fins de test, essentiellement tout périphérique possédant une pile TCP / IP et parle MQTT au-dessus. La mise en œuvre client du protocole MQTT est très simple et vraiment réduit à l'essence.

C'est un aspect, pourquoi MQTT est idéalement adapté pour les petits appareils.

II.3.2.2broker

Il s'agit du composant central du middleware, dont le rôle est de router les messages entre les applications et les clients. La contrepartie d'un client MQTT est le broker MQTT. En fonction de la mise en œuvre concrète, un broker peut gérer jusqu'à des milliers de clients simultanément connectés MQTT. Le broker est principalement responsable de recevoir tous les messages, de les filtrer, de décider qui s'y intéresse et d'envoyer le message à tous les clients subscribers.

Il contient également la session de tous les clients persistants, y compris les abonnements et les messages manqués.

Une autre responsabilité du broker est l'authentification et l'autorisation des clients. Et la plupart du temps, un broker est également extensible, ce qui permet d'intégrer facilement l'authentification, l'autorisation et l'intégration personnalisées dans les systèmes back-end. Particulièrement l'intégration est un aspect important, parce que souvent le broker est le composant, qui est directement exposé sur l'Internet et gère beaucoup de clients et passe ensuite des messages le long de l'analyse en aval et des systèmes de traitement.

Dans l'ensemble, le broker est le hub central et que chaque message doit passer. Par conséquent, il est important, qu'il est hautement évolutif, intégrable dans les systèmes back-end, facile à surveiller et bien sûr résistant aux défaillances.

II.3.2.2.1 les différents brokers MQTT

- Activemq qui permet d'ajouter MQTT à un serveur Web Apache (Développé par la fondation Apache).
- Jorammq pour l'intégration de MQTT en Java.
- Mosquitto, le broker open-source le plus utilisé dans les projets DIY soutenu par la fondation eclipse.org.
- Rabbitmq, un projet open source disponible également avec un support commercial.
- EMQTT, un projet développé en Erlang/OTP disponible pour Windows, Mac Os X et Linux.

Conçu pour recevoir de très nombreuses connexions (jusqu'à 1 million par serveur). Il est possible de créer un cluster (réseau de serveur) pour accroitre les nombres de connexions simultanées.

ActiveMQ et JoramMQ sont des brokers assez spécifiques. RabbitMQ est plus orienté entreprise avec son offre commerciale.

II.3.2.3 topic

Un sujet est une chaîne UTF-8, utilisée par le broker pour filtrer les messages de chaque client connecté. Un sujet se compose d'un ou plusieurs niveaux thématiques. Chaque niveau de sujet est séparé par une barre oblique (séparateur de niveau de sujet).



Figure II.5 : Exemple de topic MQTT.

Par rapport à une file d'attente de messages, un sujet est très léger. Il n'est pas nécessaire pour un client de créer le sujet souhaité avant de le publier ou de s'y abonner, car un broker accepte chaque sujet valide sans initialisation préalable.

II.3.2.4 connexion MQTT

Le protocole MQTT est basé sur TCP / IP et le client et le broker doivent disposer d'une pile TCP /IP (figure II.6).

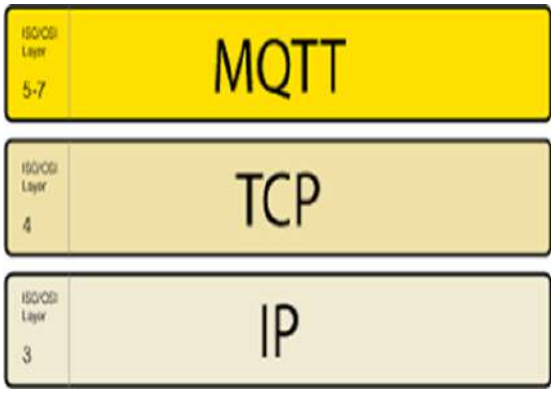


Figure II.6 : Le protocole MQTT dans le modèle OSI.

La connexion MQTT elle-même est toujours entre un client et le broker, aucun client n'est connecté directement à un autre client. La connexion est initiée par un client envoyant un message connecte au broker. La réponse du broker avec un CONNACK et un code d'état. Une fois la connexion établie, le broker le gardera ouvert tant que le client n'enverra pas une commande de déconnexion ou qu'il perdra la connexion.

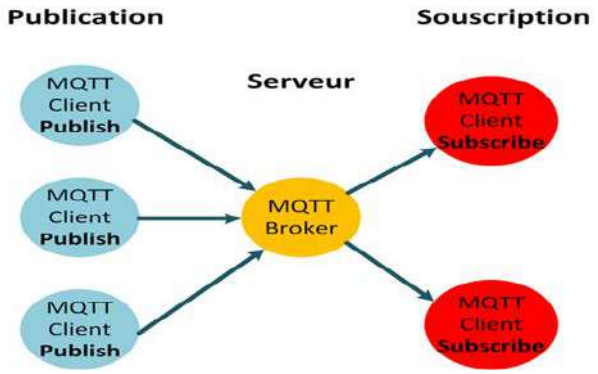


Figure II.7 : Connexion MQTT.

II.3.3 utilisations de MQTT

A l'origine, le MQTT fut développé pour les liens de données à haute latence et faible bande passante utilisés **dans l'industrie du gaz et du pétrole**. Beaucoup d'autres applications fonctionnent avec ce système de transfert de données.

De nombreux objets connectés destinés aux particuliers, ainsi que des applications comme Facebook Messenger, reposent sur le MQTT. De même **Amazon IoT se base**

sur ce protocole. En général, ce protocole est le plus adapté pour les systèmes de contrôle utilisés par les entreprises industrielles. Son taux d'adoption devrait continuer à augmenter dans les années à venir.

- La principale caractéristique de MQTT est sa légèreté, le protocole ne requiert que des ressources minimales et peut donc être utilisé sur de petits microcontrôleurs. "L'idée de MQTT est de faire dialoguer des équipements qui ne disposent pas de ressources propres pour assurer une connexion permanente", détaille Fabien Pereira Vaz, de Presser AG.
- Le protocole Mqtt est utilisé par les objets connectés pour communiquer. Il s'agit d'un protocole très léger de messagerie. Il permet donc à un objet connecté d'envoyer et/ou de recevoir des messages.
- Le Broker peut être installé n'importe où dans le monde et souvent bénéficiera d'une adresse IP fixe et ou un nom de domaine.
- Dans le cas de la démonstration avec Node-Red le Broker utilisé se trouve chez OVH en France.
- Le but de cette explication raccourcie de MQTT était de vous exposer son principe, qui sera utilisé.
- Dans la suite de cette présentation avec une application graphique qui s'appel « NODE-RED ».

II.3.4 les principales forces de MQTT

- Il est agnostique quant aux informations qu'il permet de faire transiter, c'est uniquement un protocole de transport pour les objets connectés. La taille maximale du payload d'un message est de 256Mo.
- Sa légèreté : n'augmente que légèrement la consommation de bande passante.
- Il permet de contrôler facilement la fiabilité de transmission des informations.
- Il constitue une abstraction pour la gestion du réseau : pour des connexions instables, la gestion des déconnexions/reconnexions est simplifiée.
- Il permet à de nombreux clients de recevoir ou de diffuser une information.
- Le chiffrement via TLS/SSL.
- La possibilité de gérer quels clients ont le droit d'accéder à une information ou de la publier.

II.3.5 principes de fonctionnement

- Les topics dédiés à chaque capteur de la maison.
- Pour le salon: Maison/Capteurs/Salon
- Pour la cuisine: Maison/Capteurs/cuisine
- Pour avoir la température de tous les capteurs de la maison il suffit de s'abonner au topic:Maison/Capteurs/#

II.4 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) [42]. 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 paterne 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.4.1 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 évidement possible, avec en particulier les systèmes à base de bridge dans lesquels plusieurs brokers sont utilisés, chacun étant client des autres brokers, comme sur la Figure II.8.

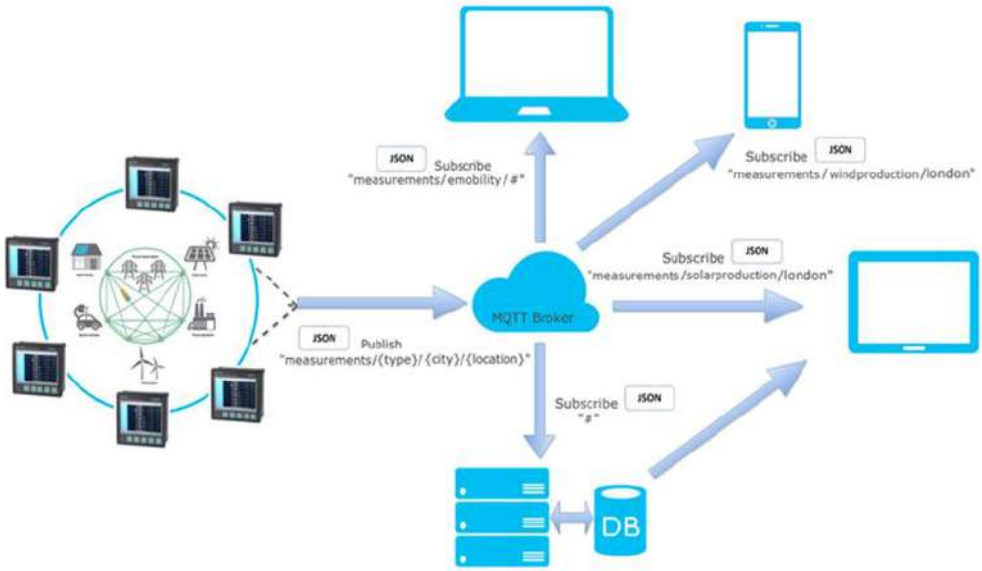


Figure II.8 : Exemples de topologies MQTT [43].

II.5 FONCTIONNEMENT

MQTT est un service de publication/abonnement TCP/IP simple et extrêmement léger. Il fonctionne sur le principe client/serveur. [44]

Le serveur, nommé broker, va collecter des informations que les publishers (les Objet communicants) vont lui transmettre. Certaines informations collectées par le broker seront renvoyées à certains publishers ayant préalablement fait la demande au broker.

Le principe d'échange est très proche de celui de Twitter. Les messages sont envoyés par les publishers sur un canal appelé topic. Ces messages peuvent être lus par les subscribers (abonnés).

Les topics (ou canaux d'informations) peuvent avoir une hiérarchie qui permet de sélectionner finement les informations que l'on désire.
Les messages envoyés par les objets communications peuvent être de toutes sortes mais ne peuvent excéder une taille de 256 Mo.

- **Exemple**

Le topic'/home/salon/température' communiquera les températures du salon si un objet connectés' y abonne (la sonde de température présente dans le salon publiera régulièrement la température relevée sur ce topic).
Si un Publisher s'abonne au topic'/home/salon/#' il recevra toutes les données du salon (on peut imaginer : luminosité, humidité, température…).
S'il s'abonne au topic'/home/#', il collectera toutes les données des sondes de la maison.

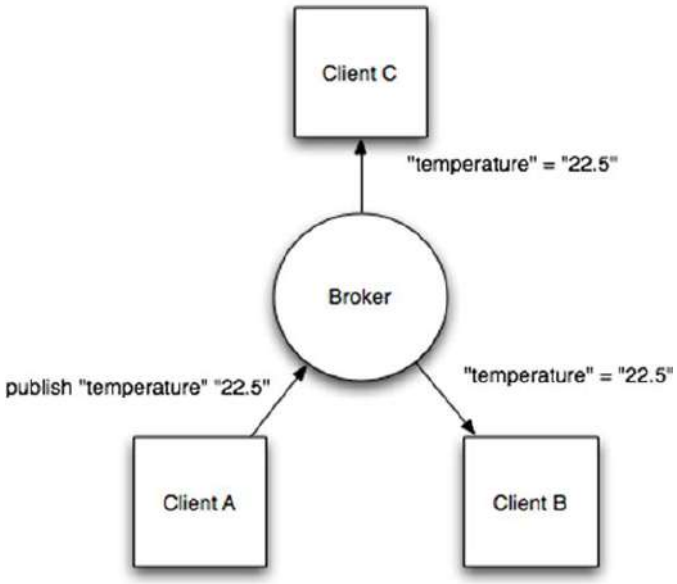


Figure II.9 : Exemple concret d'un flux MQTT.

Le Client A publié sur le topic température une valeur de 22,5 °. Le broker propage le message à tous les clients ayant préalablement souscrit au topic Température.
Une session MQTT est divisée en quatre étapes : connexion, authentification, communication et terminaison.

II.5.1 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 s'enregistrer auprès du broker, ce qui se fait avec la commande CONNECT.

Un client commence par créer une connexion TCP/IP vers le broker en utilisant soit un port standard, soit un port personnalisé défini par les opérateurs du broker. Lors de la connexion, le serveur peut continuer une ancienne session s'il reconnaît une identité client précédemment utilisée.

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, [45] 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.5.2 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 :

II.5.2.1 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 basse.

La bonne réception de cette commande est confirmée par le broker par un **SUBACK** portant le même identifiant de paquet.

II.5.2.2 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.

II.5.2.3 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.5.3 authentifications

Les ports standards sont les suivants : 1883 pour la communication non chiffrée et 8883 pour la communication cryptée utilisant SSL/TLS (Transport Layer Security / Secure Sockets Layer). Pendant l'établissement de liaison (ou handshake) SSL/TLS initial, le client valide le certificat du serveur afin d'authentifier le serveur. Lors de cet échange, le client peut également fournir au broker un certificat client que le broker pourra ensuite utiliser pour authentifier le client. Bien que cela ne soit pas spécifié dans la norme MQTT, les brokers prennent habituellement en charge l'authentification des clients avec leurs certificats SSL/TLS.

Le protocole MQTT étant avant tout destiné aux appareils disposant de ressources limitées, SSL/TLS n'est pas toujours disponible et dans certains cas, il n'est pas souhaité. Le client authentifie alors en envoyant un nom d'utilisateur et un mot de passe en clair au serveur lors de la séquence de paquets CONNECT/CONNACK.

II.5.4 communications

Au cours de la phase de communication, un client peut effectuer les opérations suivantes :

- Publication, abonnement, désabonnement ou Ping.
- Les abonnements à des sujets sont réalisés au moyen d'une paire de paquets SUBSCRIBE/SUBACK. Le désabonnement est réalisé de manière similaire à l'aide d'une paire de paquets UNSUBSCRIBE/UNSUBACK.
- Les chaînes décrivant un sujet forment une arborescence en utilisant la barre oblique (/) comme caractère de séparation. Un client peut s'abonner à des branches entières de l'arborescence d'un sujet(ou se désabonner) à l'aide des deux caractères génériques suivants : le signe plus (+), qui correspond à un seul niveau, et le dièse (#) pour plusieurs niveaux. Un caractère spécial, le dollar ($), permet d'exclure un sujet de tout abonnement utilisant un caractère générique à la racine. En règle générale, le dollar sert à transporter des messages système ou serveur spécifiques.

II.5.5 terminaisons

Lorsqu'un éditeur ou un abonné souhaite mettre fin à une session MQTT, il envoie un message DISCONNECT au broker, puis met fin à la connexion. On parle d'arrêt progressif, car il permet au client de se reconnecter facilement en fournissant son identité client et de reprendre la session là où il l'avait laissée.
En cas de déconnexion brutale ne laissant pas le temps à l'éditeur d'envoyer un message DISCONNECT, le broker peut envoyer aux abonnés un message de l'éditeur que le broker avait précédemment mis en cache. Le message, appelé dernières volontés ettestament, indique aux abonnés les mesures à prendre en cas d'arrêt inopiné de l'éditeur.

II.6 APPLICATION NODE -RED

II.6.1 présentations de Node - Red

Node-RED a été créé par IBM en 2013 afin de faciliter la création d'objets connectés. C'est un langage graphique (sous-jacent en JavaScript) utilise une approche de programmation visuelle qui permet de créer des liens entre les flux de donnée s'entrant et sortant. Il est gratuit et nécessite : Node JS (JavaScript du coté serveur) [21].

L'accès à l'interface de programmation se fait à partir d'un navigateur internet.

Il est préférable d'utiliser le navigateur d'un ordinateur connecté sur le même réseau que la Rosebery.

Voici un petit aperçu de l'environnement de développement proposé :

- Sur la gauche de l'écran se trouvent toutes les fonctions disponibles.
- Au centre se trouve la zone de programmation.
- Le panneau de droite regroupe un onglet info regroupant des informations Contextuelles (exemple correspondant à la fonction sélectionnée) et un ongle 'debug' qui renvoie les messages d'erreurs ainsi que les messages de la fonction 'debug' qui se trouve dans les outils output.
- A chaque fois que vous voulez déployer votre programme, appuyez sur le bouton 'Deploy' rouge qui se trouve dans le coin supérieur droit.

Le menu situé à côté du bouton 'Deploy' contient quelques outil supplémentaires:

- View : Permet d'afficher/masquer la grille qui peut aussi être magnétique (snap to grid)
- Import : Permet d'importer des nœuds. C'est avec cette option que vous

pouvez ajouter à votre flow des nœuds récupérés sur internet par exemple.

- Export : Vous pouvez exporter vers la presse papier (une chaine de texte au format json) ou dans un fichier dont vous devrez donner un nom.

- Flows : Vous pourrez ajouter un nouveau flow (Add), renommer (Rename),supprimer (delete) le flow en cours.

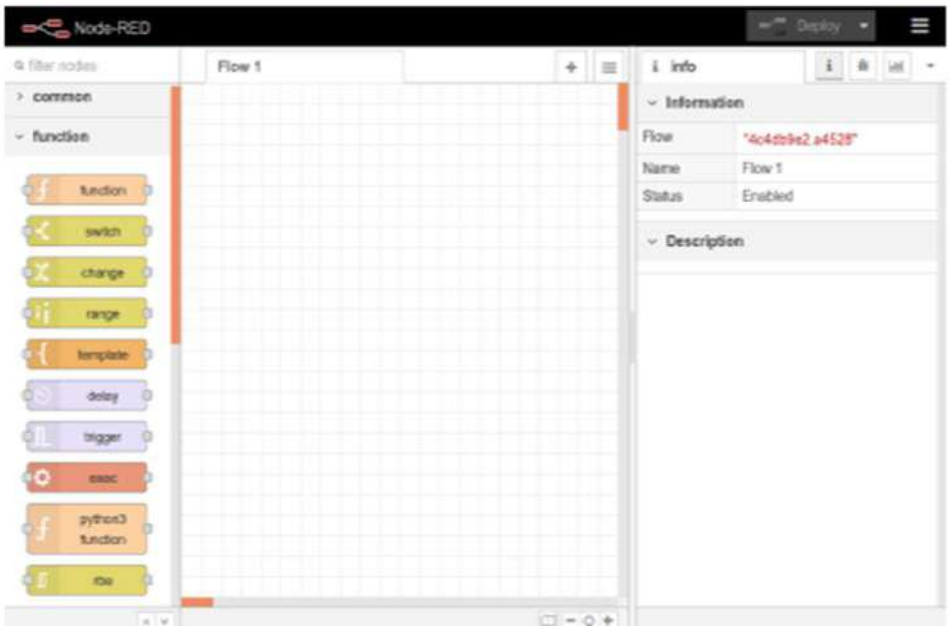


Figure II.10 : Aperçu sur Node Red.

II.6.2 installations de Node - Red

Sur un Raspberry Pi qui tourne sur Raspbian, Node Red est déjà installé. Pour le lancer, il suffit de cliquer sur le menu Programmation puis Node Red.

Node Red peut également être lancé ou stopper à partir d'un terminal :

- Pour le démarrage du Node Red, on utilise : $ Node-Red-Start.

- Pour arrêter Node Red : $ Node-Red-stop.

II.6.3 installations de nouveaux Node

La simplification de la programmation en Node-RED se fait grâce à des blocs de code prédéfinis, appelés « nodes » pour effectuer des tâches.

Les nœuds connectés, généralement une combinaison de nœuds d'entrée, de nœuds de traitement et de nœuds de sortie, lorsqu'ils sont câblés ensemble, constituent un « flow ».

Pour installer de nouveaux Node dans la palette, utiliser le menu de droit et choisir manage palette (Figure II.10). Puis aller dans l'onglet Install et lancer une recherche **Exemple :** Dashboard, Les nœuds apparaissent alors dans la palette.

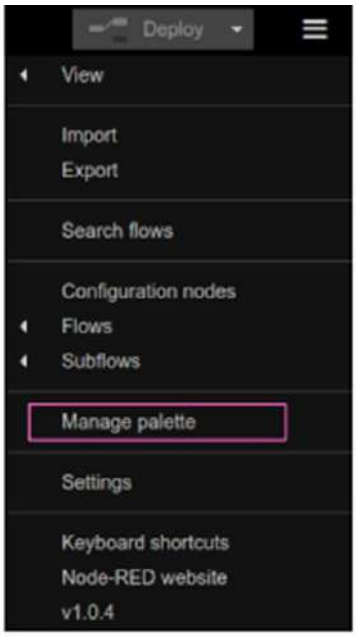


Figure II.11 : Installation d'un nouveau nœud à travers 'Manage palette'.

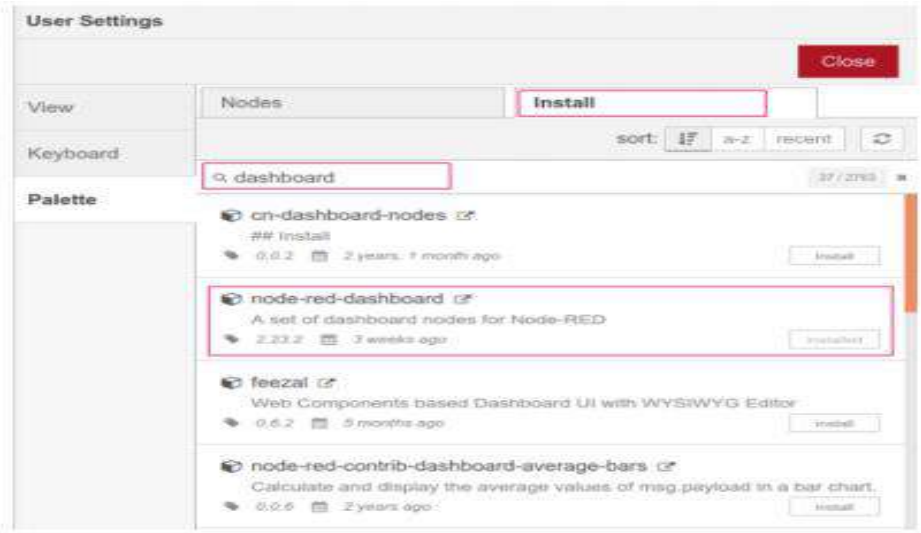


Figure II.12: Installation de nouveau nœud dans la palette.

II.6.4 créations d'un dashboard (tableau de bord IHM)

La palette de Nœuds d'origine est déjà bien fournie mais il en manque au moins une essentielle qui va permettre de réaliser l'interface web. Elle se nomme Dashboard.

Cliquez sur Manage Palette puis recherchez la palette de nœuds Dashboard.

On accède au site web généré par l'URL de sélection du groupe, cliquez sur+group, puis créer un groupe (ici : "first flow") et faite de même pour le Tab (ici :"home"). Cliquez ensuite sur le bouton Donne, puis déployer le Flow.

II.6.5 exportation et importation des flows

Node Red facilite la sauvegarde, le partage et le déplacement des flow entre les ordinateurs grâce àla fonction d'exportation et d'importation.

II.6.5.1 importation des flows

Ouvrir le flow dans un éditeur de texte, puis copier son contenu, ensuite allez dans Menu, cliquez sur import et collez les fichiers dans la fenêtre(Figure II.13).

Importer le flow dans un flow existant ou dans un nouveau flow, Cliquer sur le bouton Import pour terminer l'importation.



Figure II.13 : Importation des flows.

II.6.7 exportations des flows

On peut exporter un nœud sélectionné, un flow ou tous les flows.

Sélectionner chaque nœud du flux que nous souhaitons exporter dans l'espace de travail, Cliquer ensuite sur Menu, ensuite cliquez sur Export.

Il y a trois onglets à l'écran avec l'onglet des nœuds sélectionnés. Nous pouvons le modifier pour exporter le flow actuel ou tous les flows de l'espace de travail.

Cliquer sur Export to clip board, puis ouvrer un éditeur de fichiers et coller le contenu dans le fichier, Donner un nom au fichier et enregistrer-le, Ou bien cliquer sur Download pour exporter un fichier sous format JSON (Figure II.14).

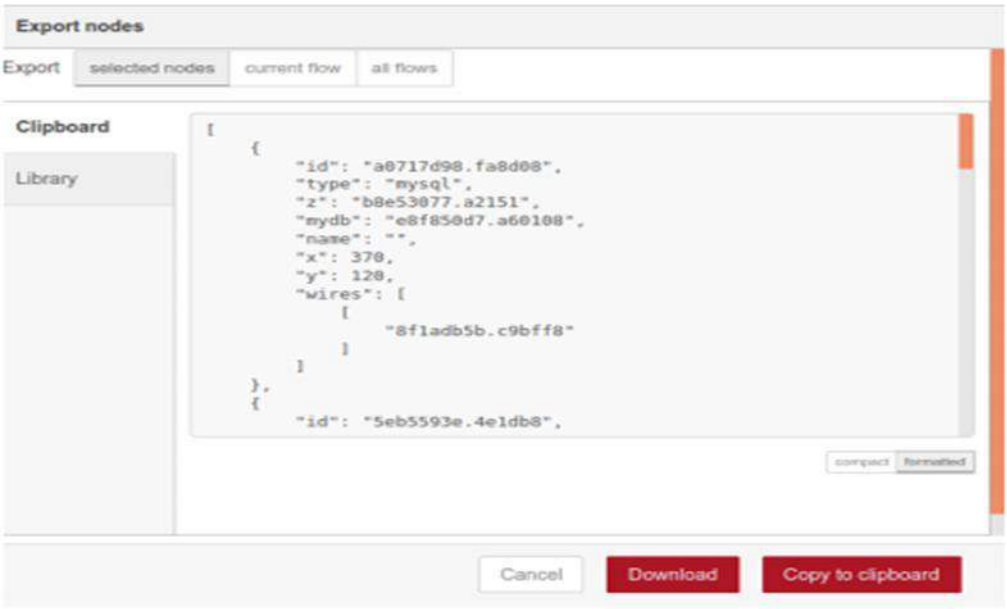


Figure II.14 : Exportation des flows.

II.7 CONCLUSION

Nous avons pu voir dans cette chapitre le protocole MQTT qui s'appuie surtout sur le modèle publish / subsribe ou encore publication/souscription. Ce protocole de communication joue un grand rôle dans un réseau IdO grâce à ses différents avantages dont le principal est le fait d'être léger ainsi que sa faible consommation.

Le protocole s'améliore en permanence et prend désormais en charge Web Sockets, un autre protocole qui permet une communication bidirectionnelle en temps réel entre les clients et les brokers.

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'IdO . C'est la raison pour laquelle nous avons choisi ce protocole pour l'implantation et la réalisation d'une application de contrôle de température et humidité dans IdO

Chapitre III :

Implémentation

Chapitre III : Implémentation

III.1 INTRODUCTION

Dans ce chapitre nous allons parler de l'implémentation de tous les composants du système et des langages de solutions que nous avons utilisé.

III.2 CHAIER DE CHARGE

Node-Red : Node-Red est un outil puissant pour construire des applications de l'Internet des Objets en mettant l'accent sur la simplification de la programmation qui se fait grâce à des blocs de code prédéfinis, appelés nodes pour effectuer des tâches. Il utilise une approche de programmation visuelle qui permet aux développeurs de connecter les blocs de code ensemble. (figure III.1).

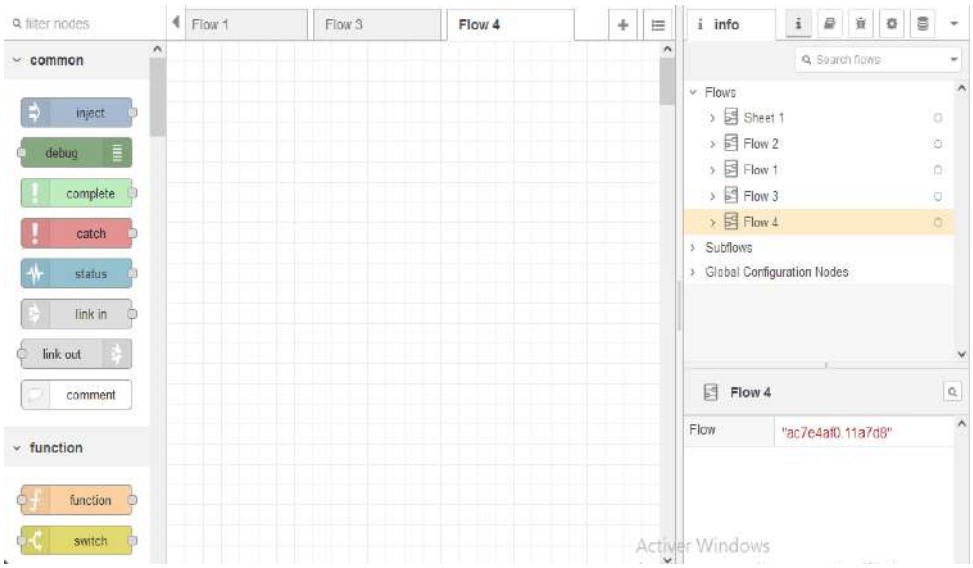


Figure III.1 : la fenetre de Node-Red.

Dans notre projet, nous avons créé une solution pour envoyer des données à distance via le protocole **MQTT** qui contient un Broker (mosquitto), et la forme générale de notre projet est représentée dans la figure suivante(figure III.2) .

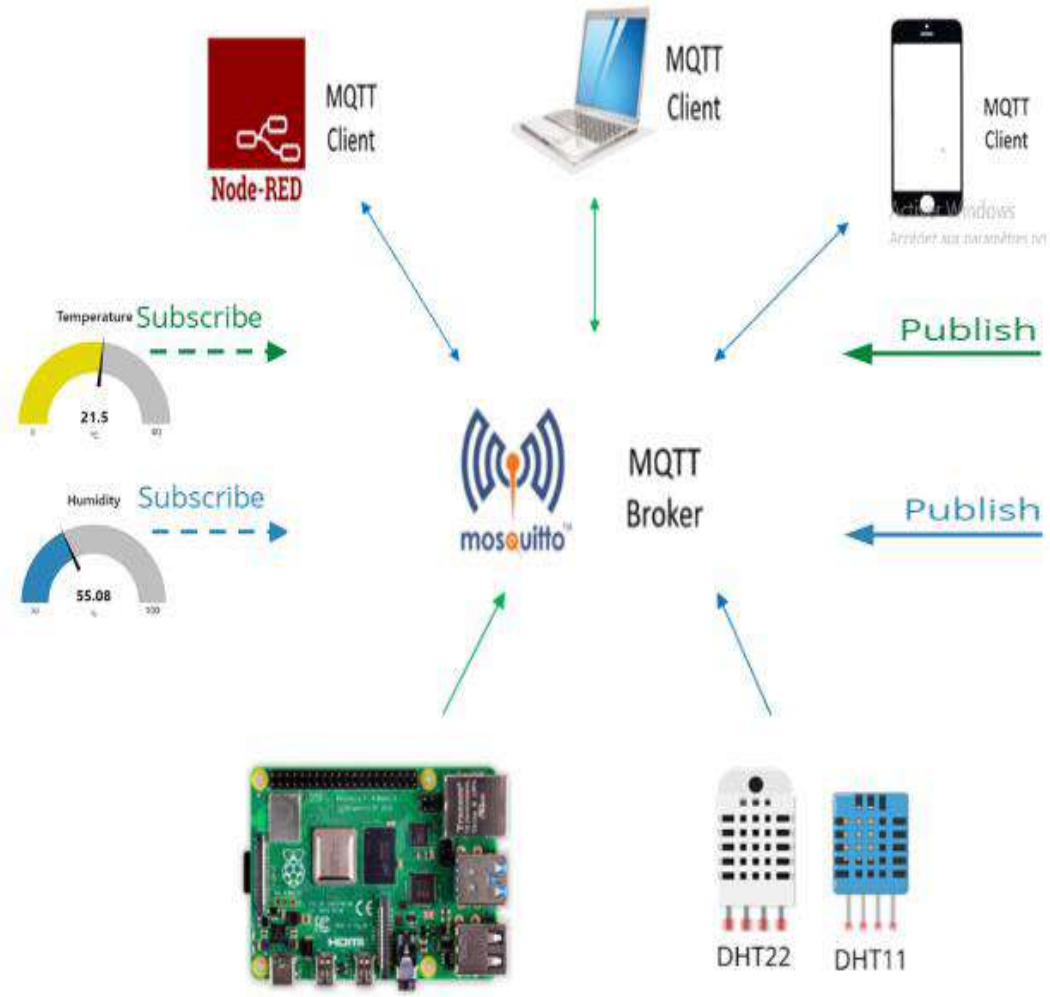


Figure III.2 : schéma globale.

Notre système est constitué de :

- Module de capture des informations :
  - Capteur d'humidité de l'air et de température : **DHT11**
- Module de communication :
  - La communication **MQTT** : entre le nœud principal et le RPI

La figure III.3 représente la réalisation de notre système

Et la figure III.4 représente le diagramme de notre projet

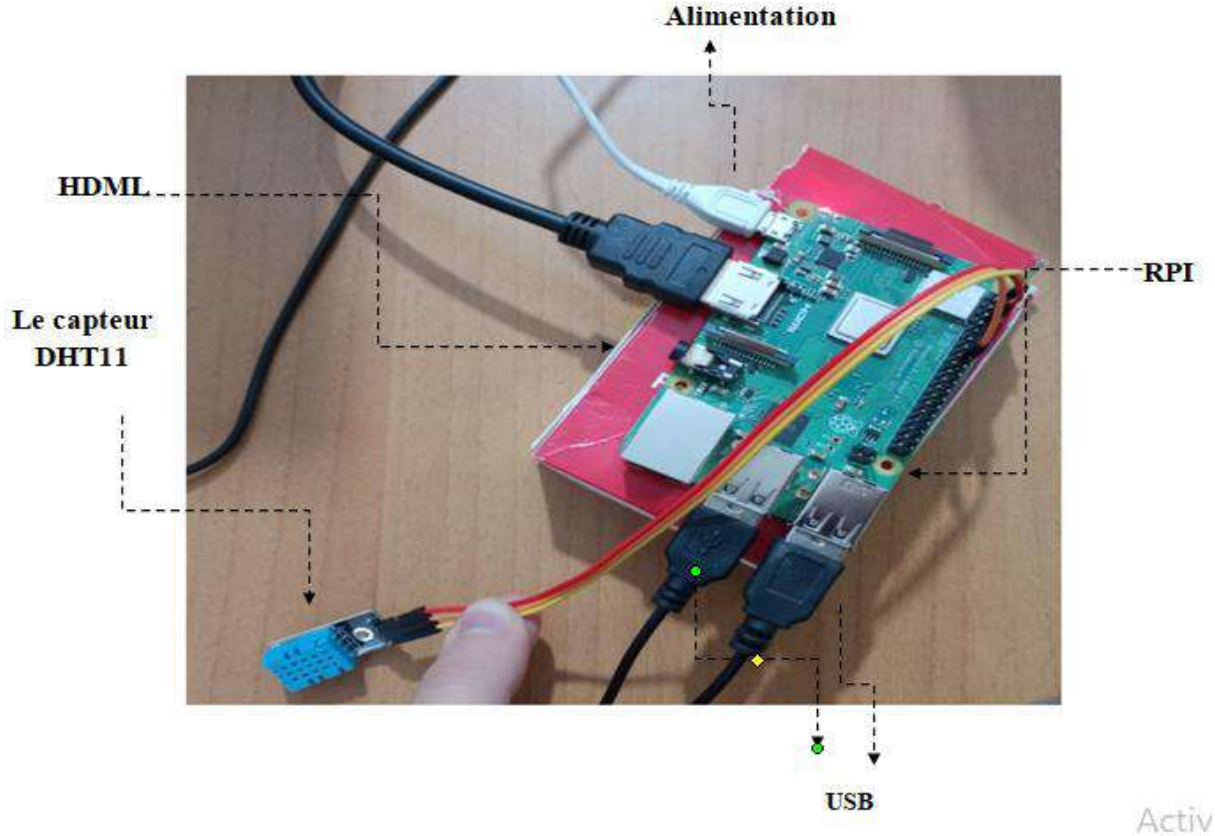


Figure III.3 : la réalisation de notre système.

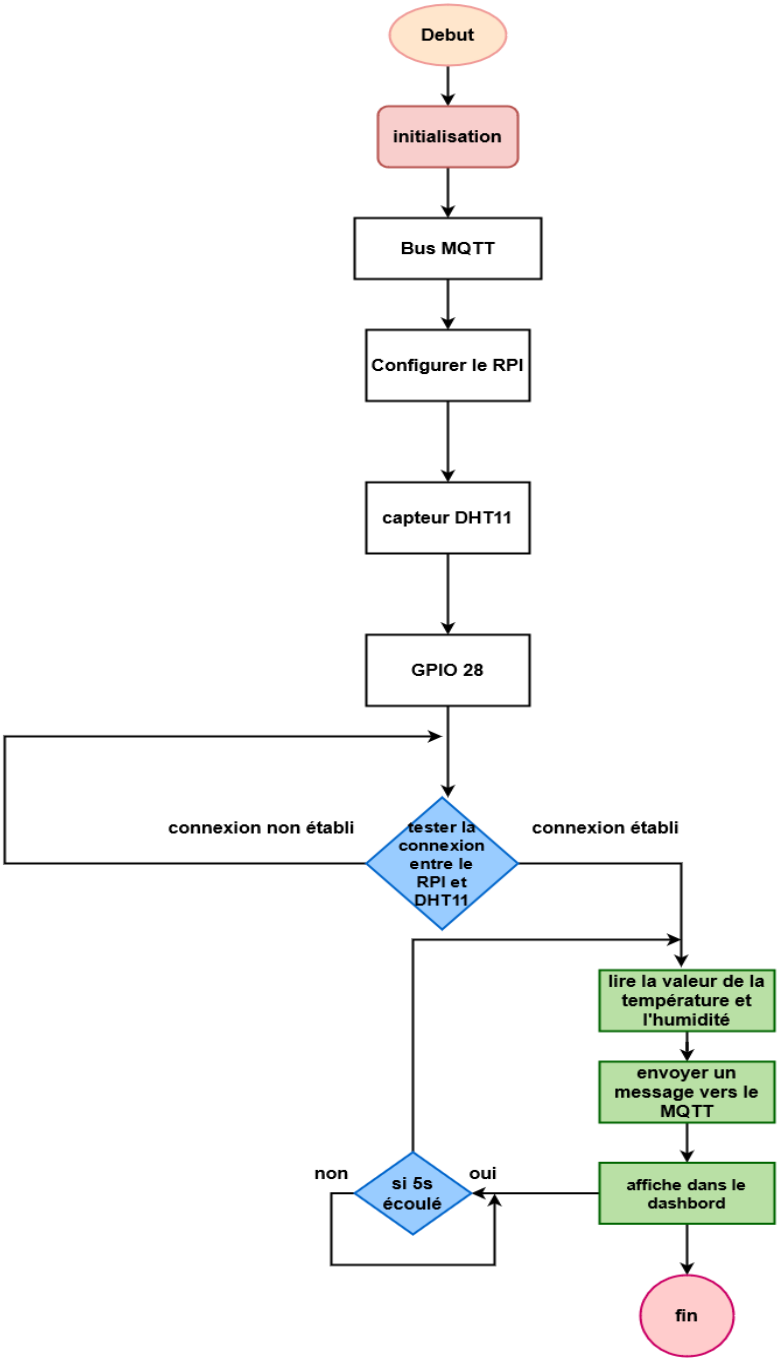


Figure III.4 : le diagramme de notre projet.

III.3 PRINCIPE DE FONCTIONNEMENT DE SYSTEME PROPOSE

Le système que nous proposons est une solution pour envoyer et recevoir un ensemble des informations par exemple un message par un moyen de communication (PC , Smart Phone ..) à distance.

Dans notre projet nous avons détailler comment mesurer la température et l'humidité par le RPI et le protocole MQTT on utilise le capteur DHT11 , et l'envoyer depuis le lieu de mesure à plusieurs PC distants , après on ajouter le nœud FRED et on fait un programme qui est envoyer un donnée par des deux réseaux différents , et à la fin on fait la comparaison entre eux : MQTT et HTTP .

Notre travail est devisé en quatre parties :

- **Partie 1 :** Commençons par réaliser un flow très simple, ce flow lit la dernière valeur de la températures et l'humidité et affiché sous forme d'une jauge dans une fenêtre qui s'appelle le " dashboard " , on utilise dans cette partie le **RPI** et le capteur **DHT11** .

On a réaliser le programme suivant sur notre Node-Red comma la figure III.5 est présenté:

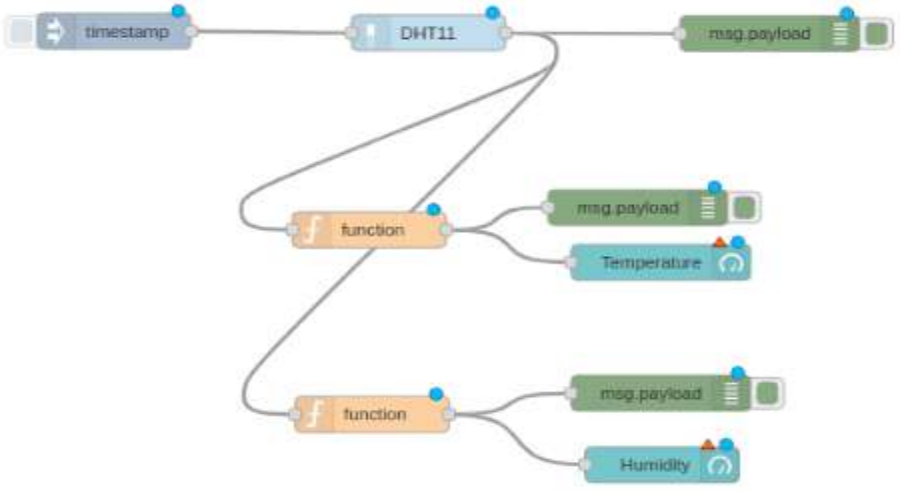


Figure III.5: Lire la dernière valeur de la température et l'humidité.

Ce flow composé de :

- **un nœud Gauge :** pour ajouter un widget de type jauge à l'interface utilisateur.

Dans les figures ci-dessous avons expliqué comment nous avons fait la configuration de notre nœud " gauge " (figure III.6) (figure III.7).

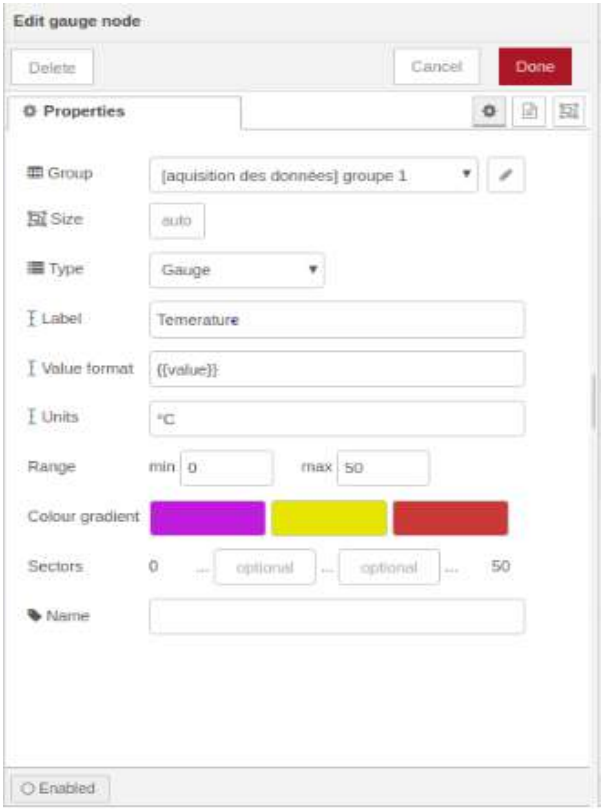


Figure III.6 : gauge de " température ".



Figure III.7 : gauge de" humidity ".

- **un nœud debug :** qui affiche le msg.payload
- **un nœud DHT11 :** pour mesure la température et l'humidité (figure III.8).

dans notre cas nous avons choisi le code PIN 16 de broche GPIO .

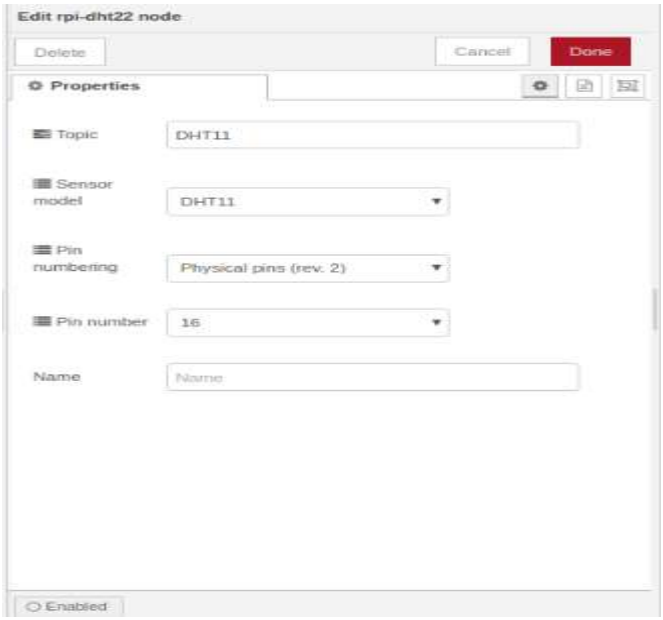


Figure III.8 : le nœud DHT11.

▪ **un nœud function :** Le nœud Fonction permet de faire passer un message par une fonction JavaScript.

Dans les figures représentées ci-dessous , nous montrant comment nous avons écrit la fonction de «la température et l'humidité ». (figure III.9) (figure III.10)

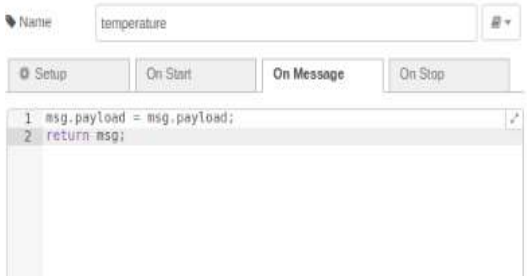


Figure III.9 : la fonction "température ".



Figure III.10 : la fonction " humidity ".

▪ **un nœud Inject :** Le nœud d'injection nous permet d'injecter des messages dans un flux, soit en cliquant sur le bouton du nœud, soit en fixant un intervalle de temps entre les injections. En déployant (en cliquant sur le bouton deploy) on obtient bien la dernière valeur de la température et l'humidité (figure III.11) :

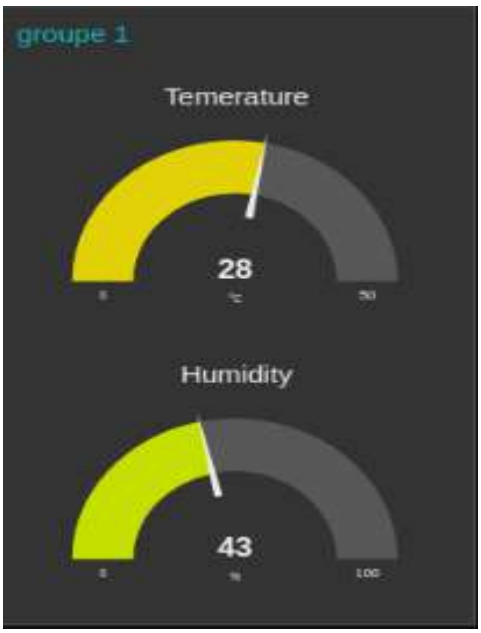


Figure III.11: Affichage de température et l'humidité en temps réel.



Figure III 12: Affichage de température et l'humidité "debug ".

▪ **Partie 2 :** dans cette partie on a utilise le protocole MQTT pour envoyer les données à distance , et pour cette raison on utilise le nœud d'envoi des données MQTT comme représente la figure III.13 .

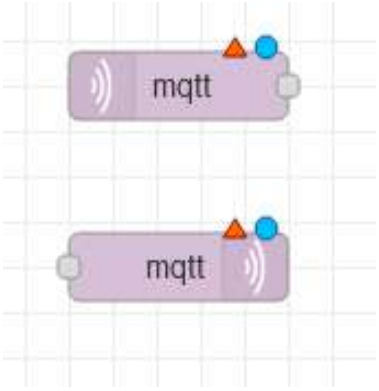


Figure III.13: Le nœud MQTT.

Pour envoyer via le protocole MQTT , on clique sur le nœud pour configurer les paramètres de protocole MQTT (topic , adresse du broker, etc.)

Nous avons choisi le topic "maison/humidity " et le topic " maison/temperature " avec une même adresse de broker "10.0.62.127 :1883 " comme le montre la figure III.14 :

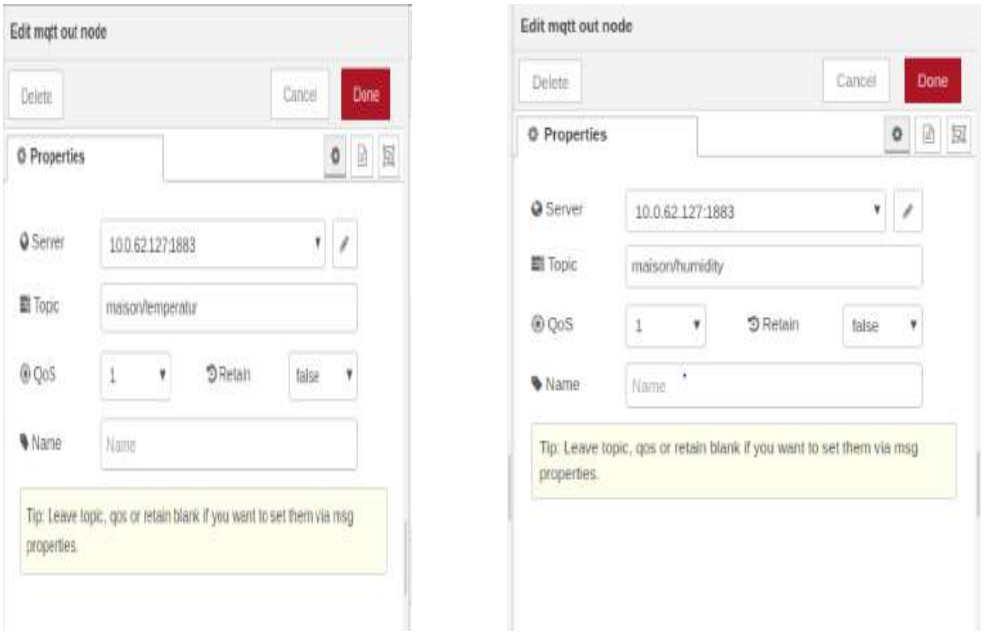


Figure III. 14: Configuration du nœud MQTT out.

D'Après avoir ajouté le protocole MQTT à la première partie on obtient le programme suivant : (figure III.15)

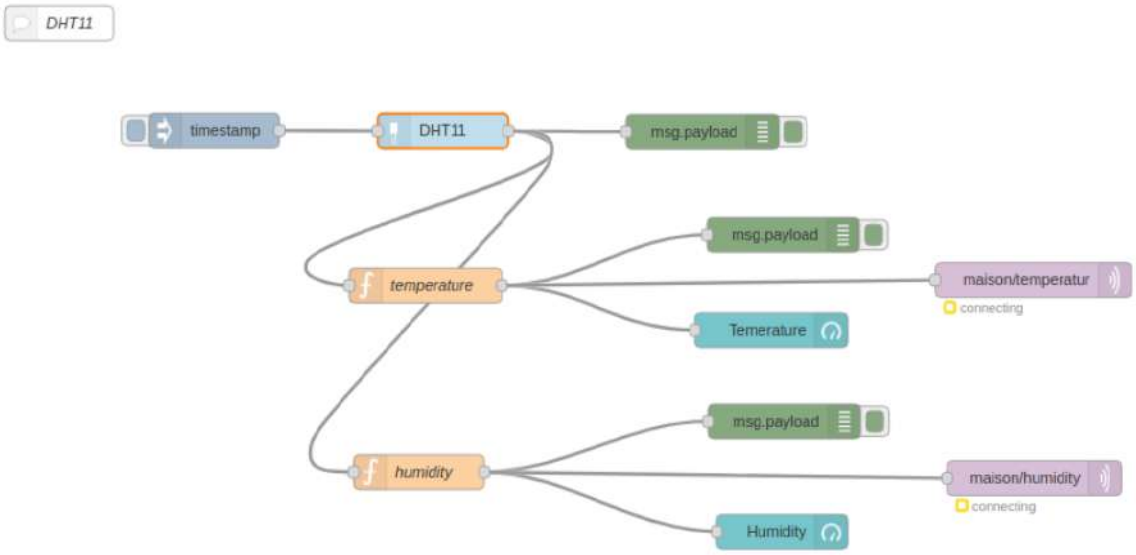


Figure III.15 : acquisition des données avec le protocole MQTT.

On suivi les étapes suivants pour envoyer et recevoir les données (la température et l'humidité) à distants :

- Nous ouvrons un terminal dans le RPI et on a écrit le mot «mosquitto » et on fait enter , après on écrit le mot «ifconfig » et on fait entrer pour obtenir l'adresse de Broker (figure III.15).

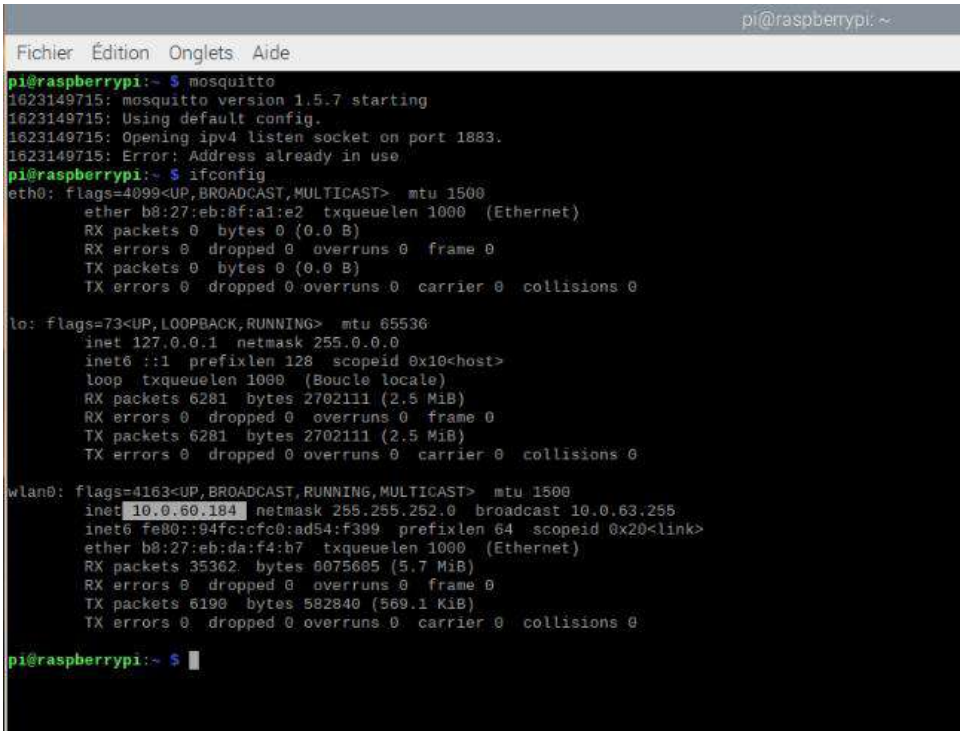
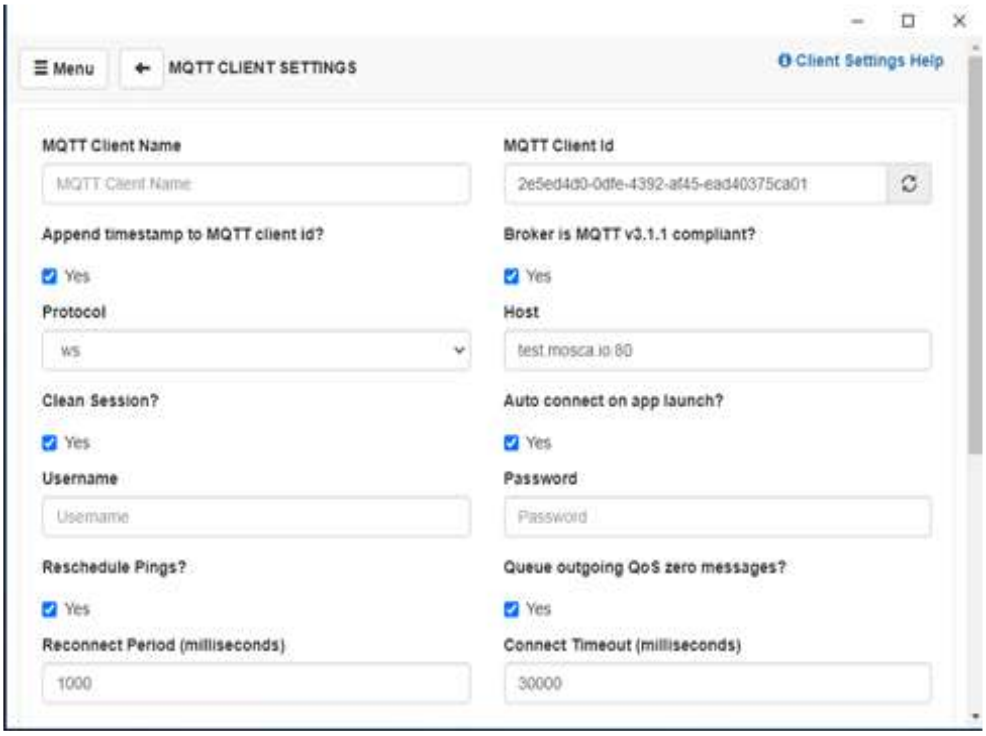
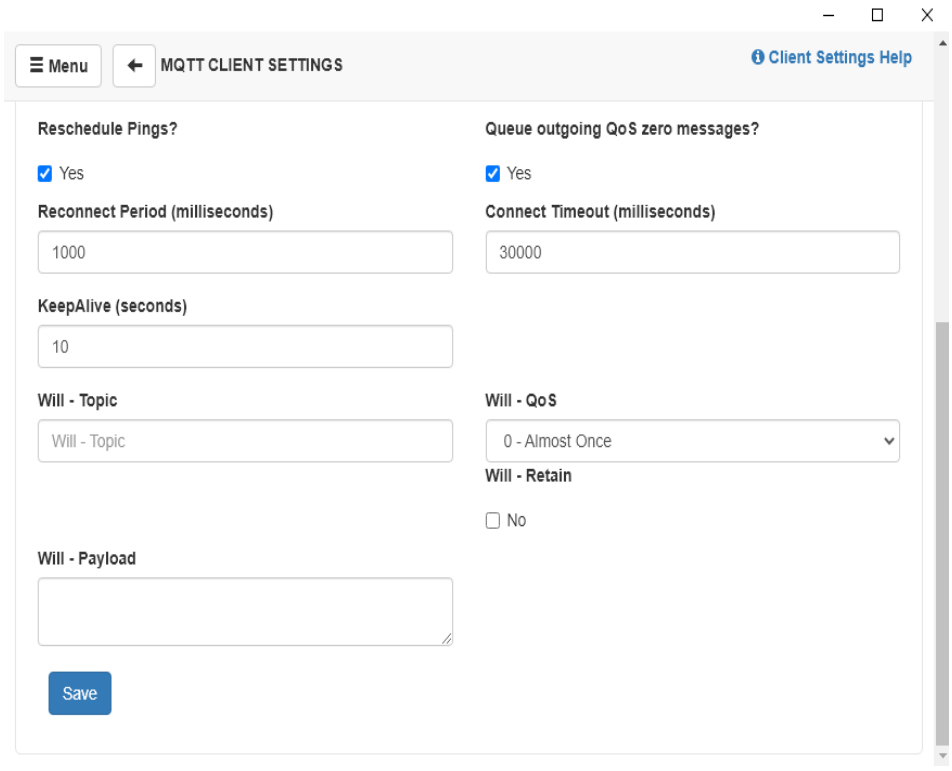


Figure III.16 : le terminal.

- On fait l'installation de mqttbox
- La configuration de MQTT :
  - Host c'est le serveur et on a écrit l'adresse de mosquitto : Broker
  - On choisi le protocole : MQTT/TCP.
  - Appuyer sur Save .

Comme représentées les figures ci-dessous :


Menu
MQTT CLIENT SETTINGS
Client Settings Help
MQTT Client Name
MQTT Client Name
MQTT Client Id
2e5ed4d0-0dfe-4392-af45-ead40375ca01
Append timestamp to MQTT client id?
Yes
Broker is MQTT v3.1.1 compliant?
Yes
Protocol
ws
Host
test.mosca.io 80
Clean Session?
Yes
Auto connect on app launch?
Yes
Username
Username
Password
Password
Reschedule Pings?
Yes
Queue outgoing QoS zero messages?
Yes
Reconnect Period (milliseconds)
1000
Connect Timeout (milliseconds)
30000


Menu
MQTT CLIENT SETTINGS
Client Settings Help
Reschedule Pings?
Yes
Queue outgoing QoS zero messages?
Yes
Reconnect Period (milliseconds)
1000
Connect Timeout (milliseconds)
30000
KeepAlive (seconds)
10
Will - Topic
Will - Topic
Will - QoS
0 - Almost Once
Will - Retain
No
Will - Payload
Save

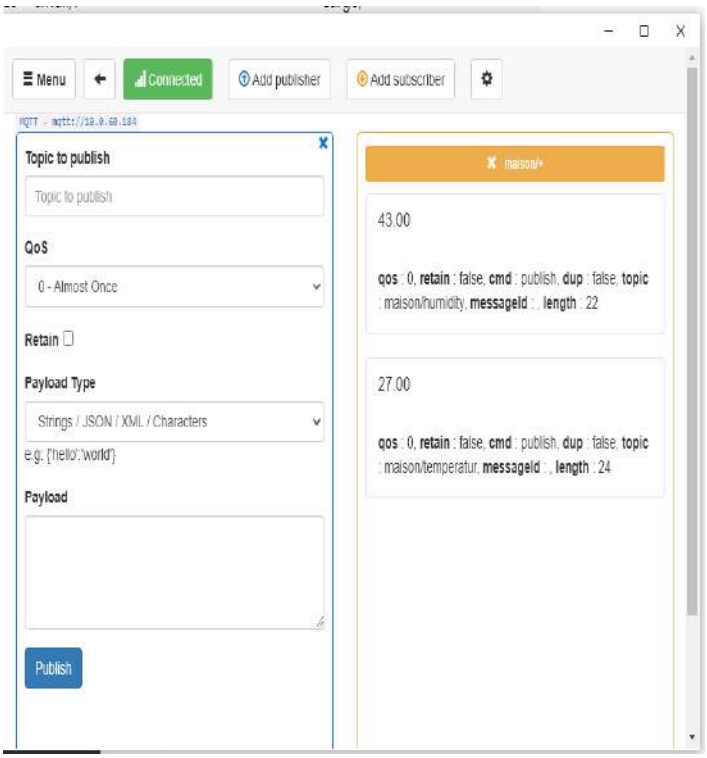
III.4 ENVOYER LES DONNEES A DISTANT :

III.4.1 De RPI via le PC :

La figure ci-dessous présentée comment envoyer la valeur de la température et l'humidité de **RPI** et recevoir même valeur dans un autre **PC** avec un même broker .et avant faire cela , le RPI et le PC doivent être connecté au même réseau , et ils doivent être les mêmes topics .

- La valeur de la température est : **27.00 °**
- Le valeur de l'humidité est : **43.00 %**

Les figures ci-dessous montre le résultat :

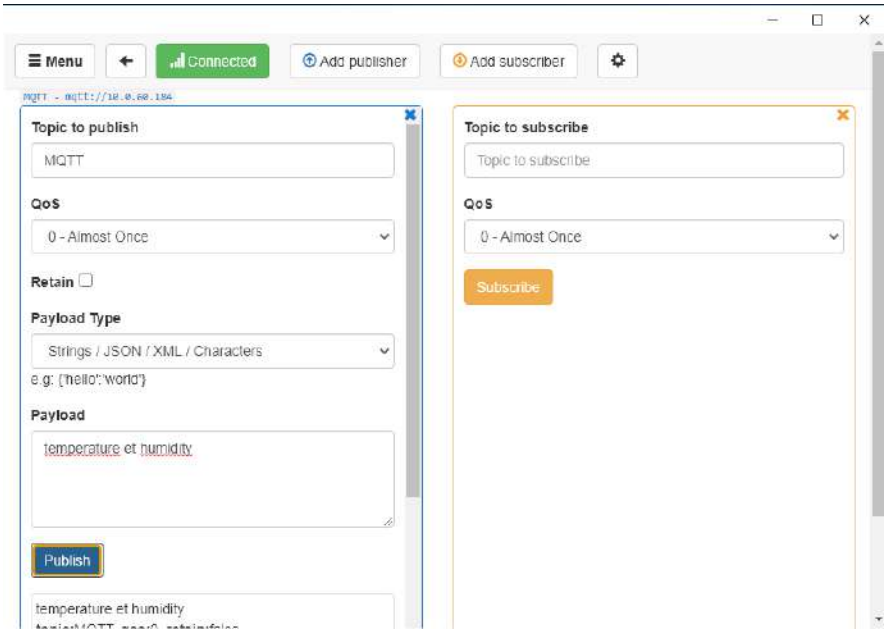




III.4.2 De PC via le RPI :

Nous avons envoyé le message " la température et l'humidité " de **PC** vers le **RPI** dans un même réseau , on utilise ici le nœud de **MQTT** et le nœud de débogage et on a écrit un programme dans notre Node-Red , et nous avons appuyer sur le mot " subscribe " pour faire l'abonnement entre le **PC** et le **RPI** et après on a appuyer sur le mot " publish" pour envoyer le message et on a reçu le même message dans notre **RPI** .

Comme représentées les figures suivantes :





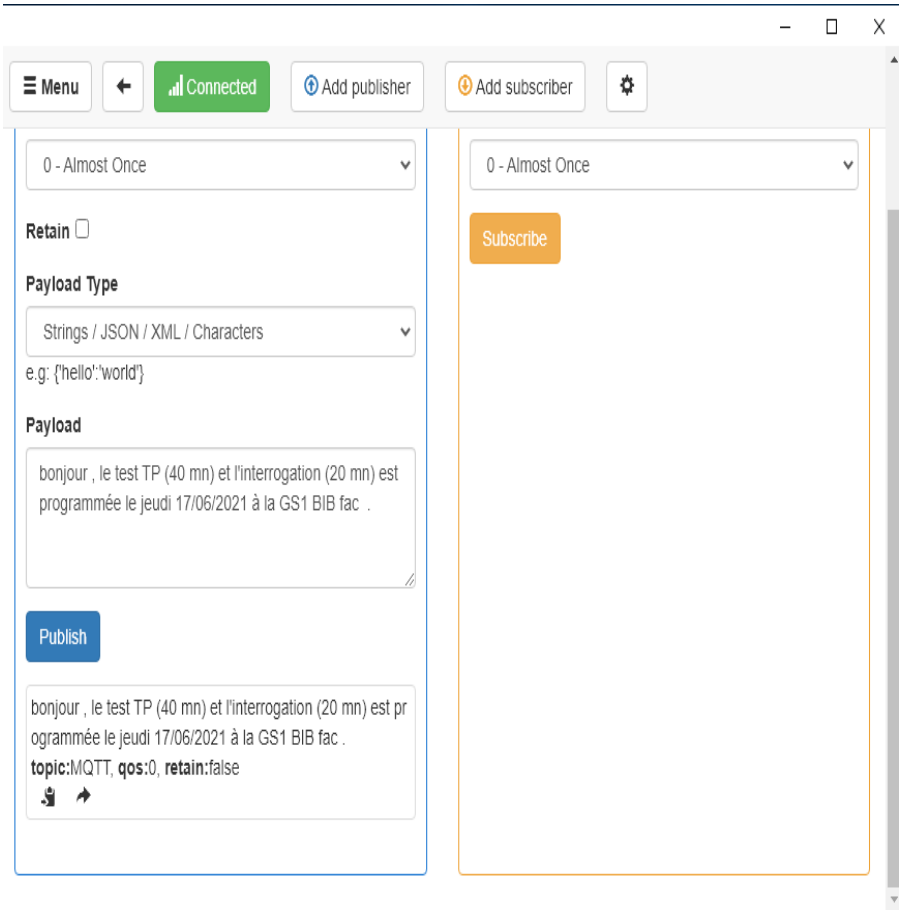


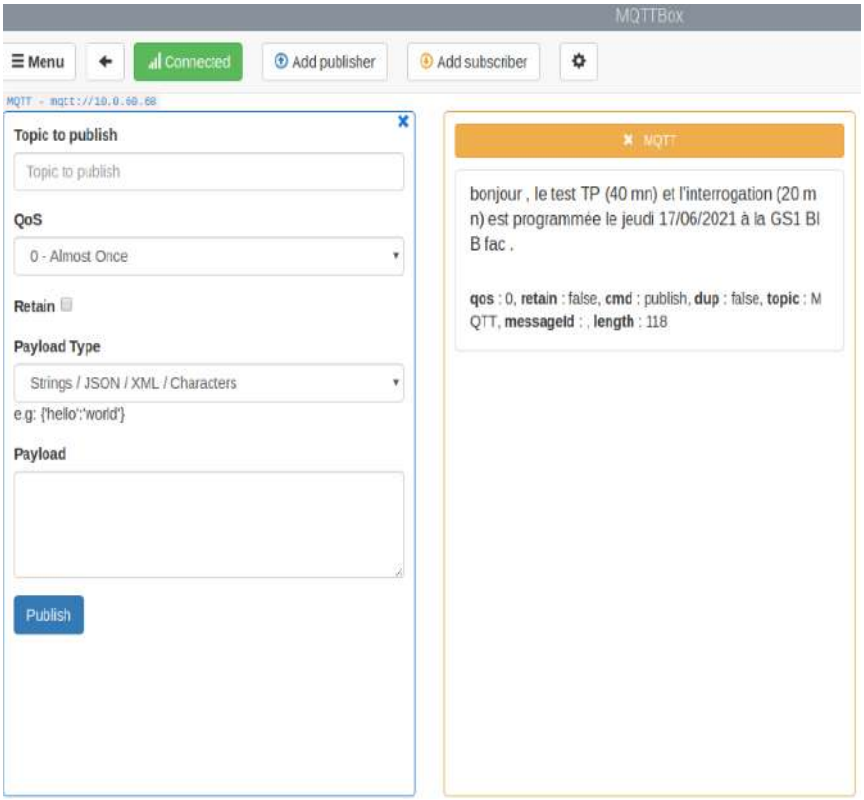
III.4.3 De PC via PC :

le message **" bonjour, Le test de TP (40 mn) et l'interrogation (20mn) est programmée le jeudi 4/03/2021 à la GS1 BIB fac "** a été envoyer de PC 1 vers le PC 2 d'après les étapes suivants :

- les deux pc doivent être connectés au même réseaux .
- On a appuyer sur le mot "subscribe "dans le PC 2 pour faire l'abonnement de PC 1.
- On va écrire le message , après on a appuyer sur le mot " publish " , le message doit être reçu dans le deuxième PC .

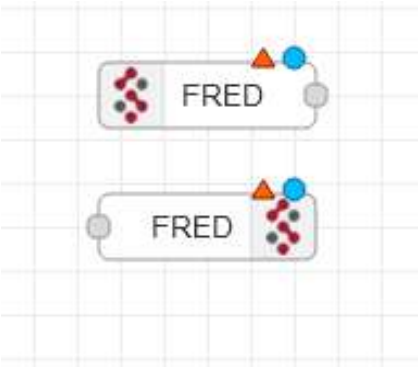
Comme le présent les figures suivants :

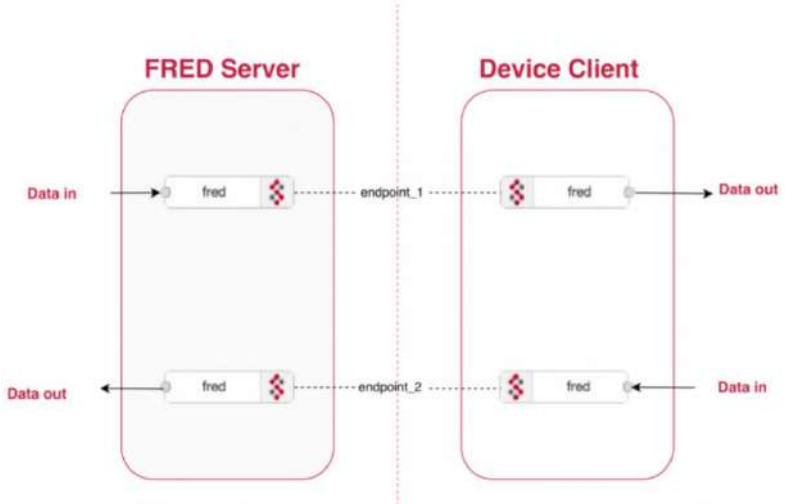




- **Partie 3** : dans cette partie Nous avons créé un nouveau nœud FRED pour faciliter cette connexion une fois configuré et exécuté, nous pouvons câbler un flux sur un appareil qui se termine par un nœud FRED , et les messages sont ensuite envoyer automatiquement à un nœud d'entrée FRED .

  - **Front End pour Node-RED** (FRED) gère les instances de Node-RED pour plusieurs utilisateurs dans le Cloud . Nous gérons et optimisons une instance Node RED afin de nous soucier de la réalisation de notre projet, et non de la configuration et de la maintenance d'une instance Node-RED.



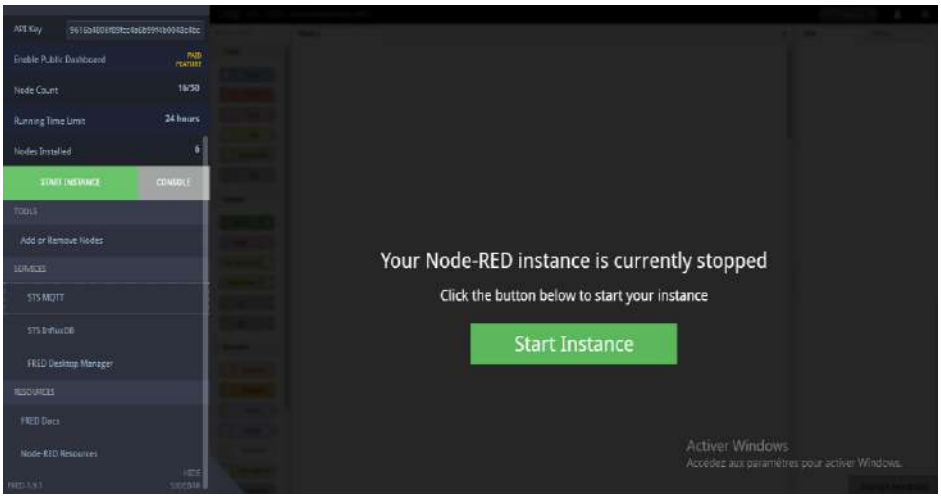


Il s'agit d'une illustration de haut niveau de l'utilisation du nœud FRED. Nous avons à la fois le nœud d'entrée fred et le nœud de sortie fred côté serveur et côté client de l'appareil.

- **Créer un compte FRED :**

Pour commencer notre tutoriel on va créer un compte gratuit sur https://fred.sensetecnic.com/ .

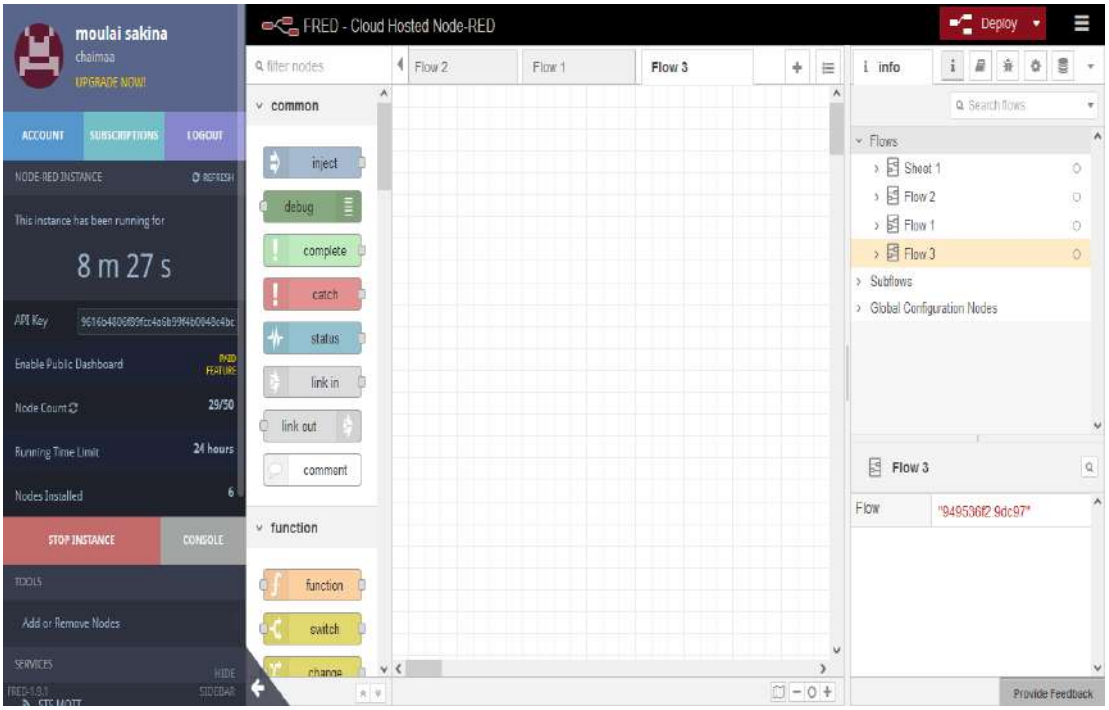
Après l'inscription, nous nous assurons d'activer notre compte via notre e-mail. Nous pourrons nous connecter jusqu'à ce que nous ayons vérifié notre compte.



- Dans ce tutoriel , nous expliquerons comment installer le nœud FRED et comment le configurer pour parler à notre flux Cloud Node-RED exécuter sur FRED .
- Une fois la configuration terminée , nous redémarrons Node-REd sur notre appareil et on a appuyer sur **Start Instance** .



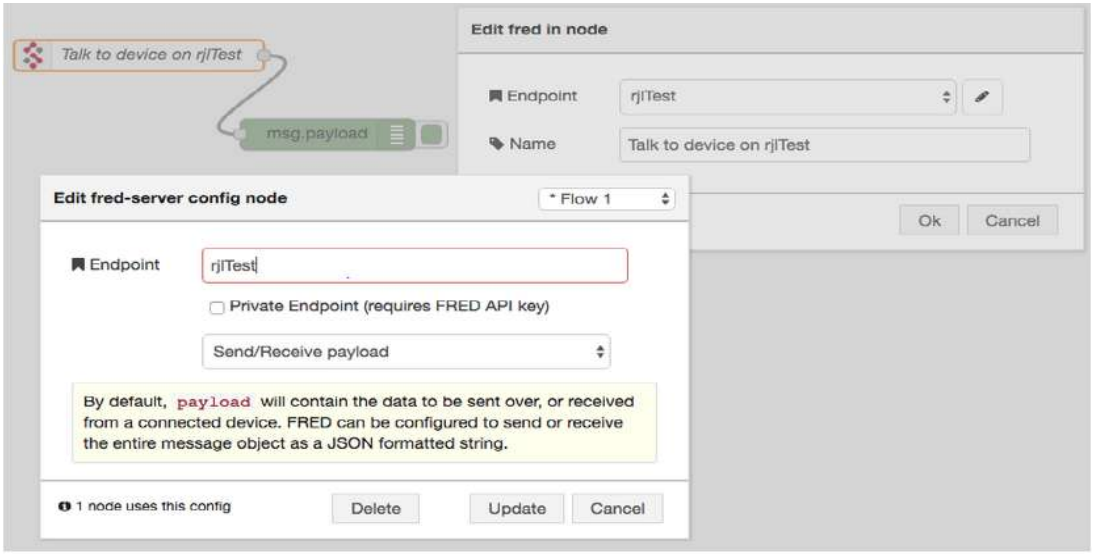
Une interface apparait :

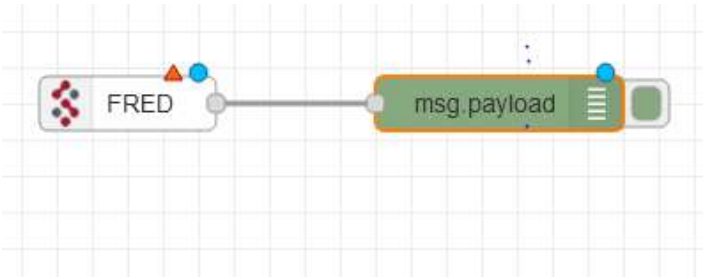


- Une fois que nous avons connecté au service FRED nous faisons glisser le nœud d'entrée de Fred sur le canevas et attachons un nœud de débogage. Ensuite, pour configurer le nœud d'entrée fred, nous double-cliquons et un écran de configuration apparaîtra avec deux champs de données, EndPoint et Name. Le point de terminaison est configuré en sélectionnant l'icône de modification, puis en fournissant un nom textuel pour le point

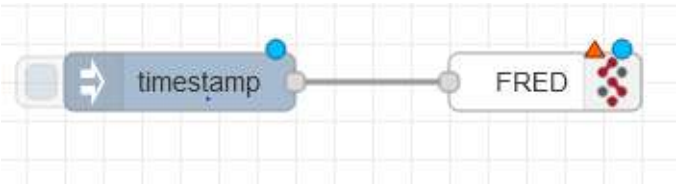
de terminaison de connexion. Dans l'exemple, nous utilisons "**rjlTest**" - vous pouvez utiliser n'importe quel nom.

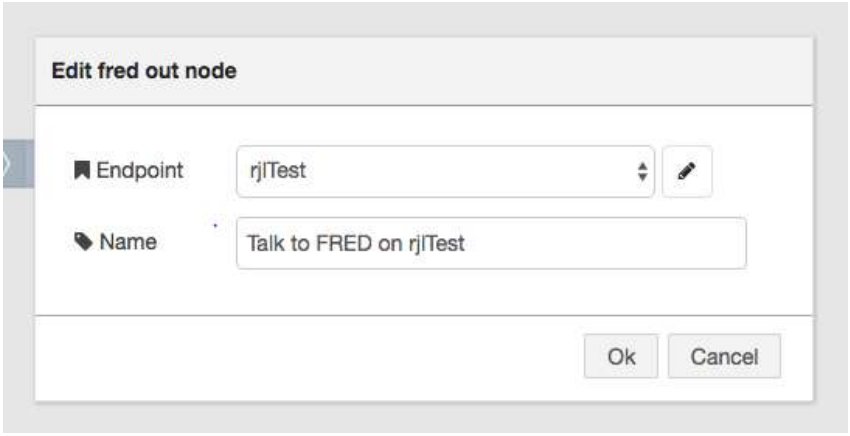
- Une fois le point de terminaison créé, nous le définissons, puis nommons le nœud, nous utilisons "**Talk to device on rjlTest**".



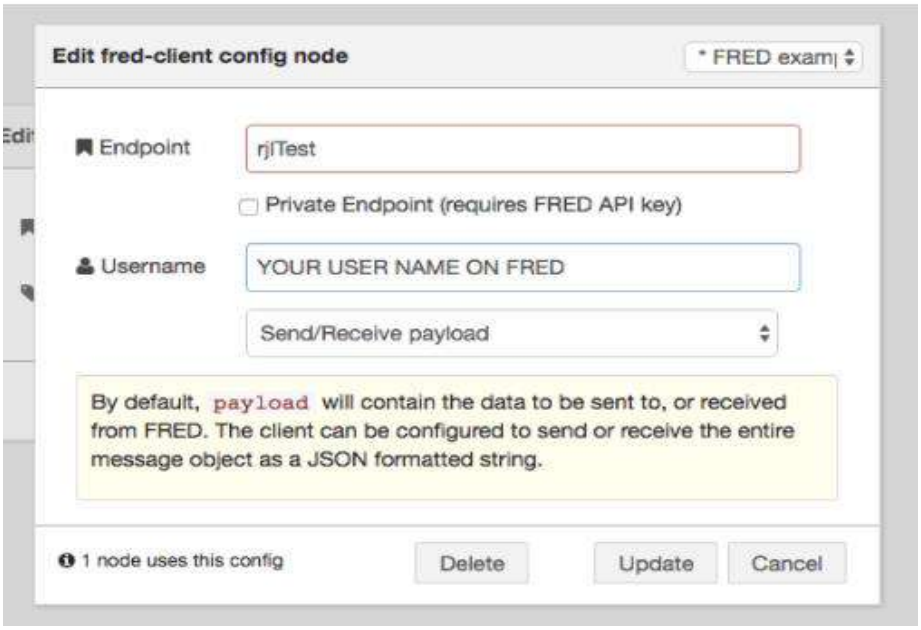


- Pour configurer un nœud FRED sur notre appareil, commençons par le nœud de sortie fred. Tout d'abord, nous connectons le nœud d'injection (illustré ci-dessus) au nœud de sortie unique. Ensuite, nous double-cliquons sur le nœud de sortie fred pour pouvoir le configurer.



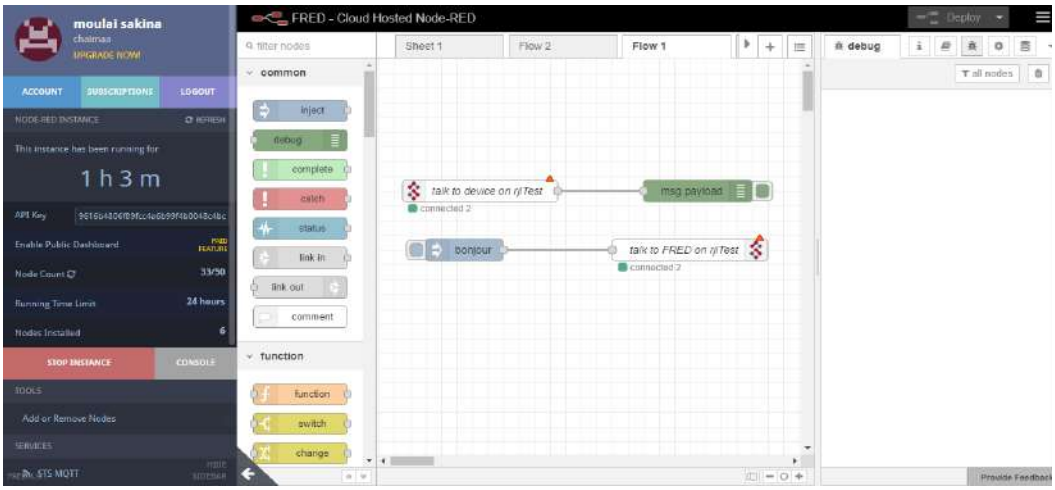


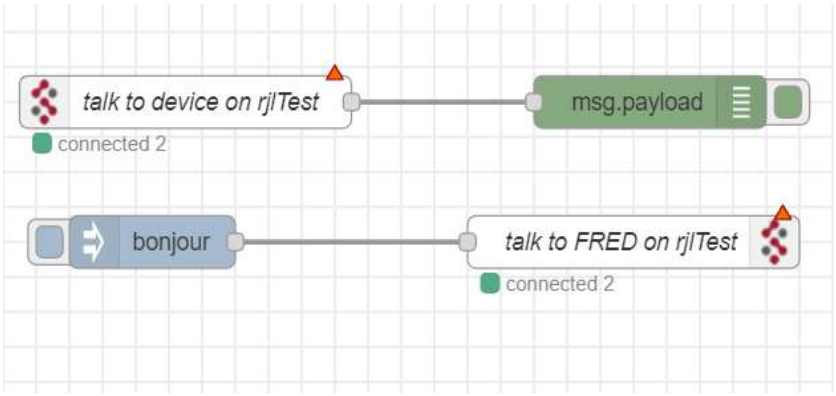
- Nous donnons un nom au nœud - nous avons utilisé" **Talk to FRED onrjlTest** "



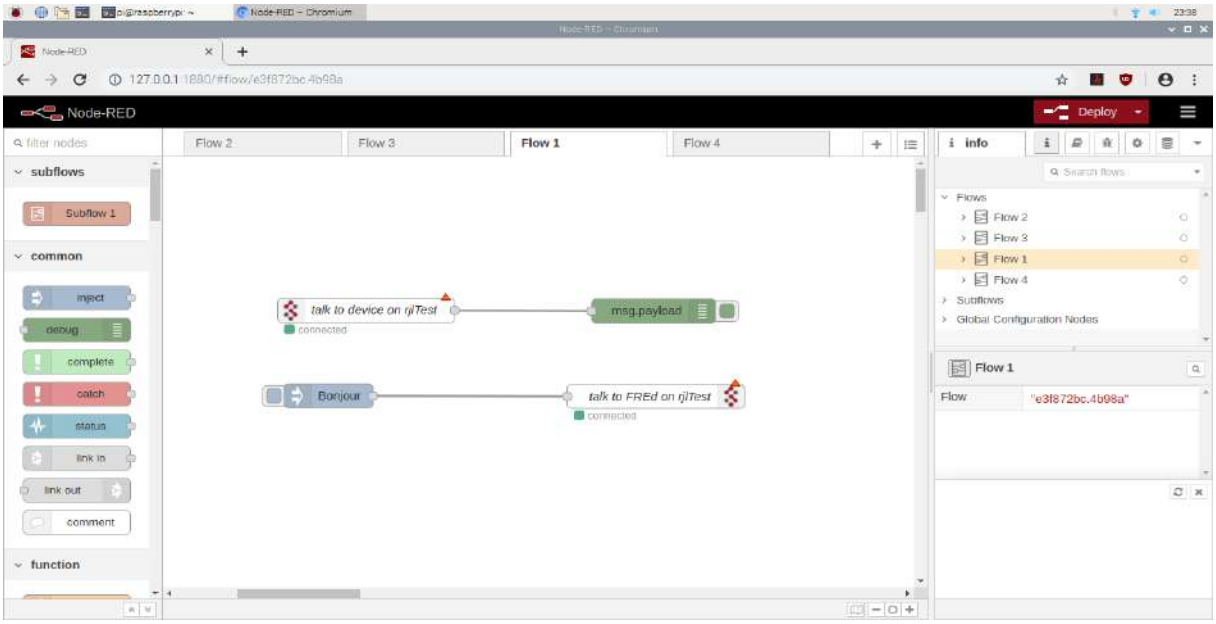
- Nous remplaçons « VOTRE NOM D'UTILISATEUR SUR FRED » par le nom d'utilisateur (nom de connexion) sur le système FRED afin que le nœud sache à qui il envoie les données. Nous n'avons pas eu besoin de le faire côté serveur, car nous sommes déjà connectés à FRED donc il connaît déjà notre nom. Cependant, dans une instance de périphérique autonome Node-RED, le nœud doit connaître notre nom d'utilisateur FRED.
- En fait ce programme pour envoyer et recevoir les messages entre deux PC avec deux réseaux différents :

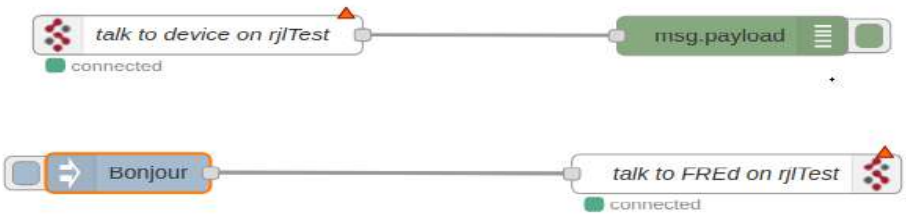
PC 1 : avec un réseau wifi



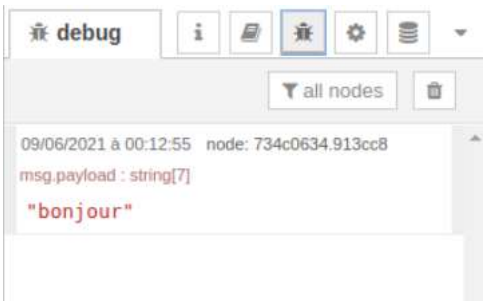


PC 2 : avec un réseau 4G

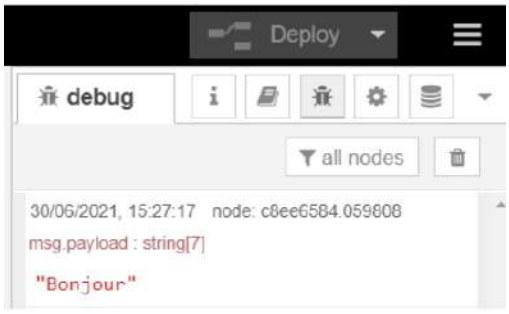




Les figures suivantes représente le résultat de le message " Bonjour " qui a été envoyer depuis le premier PC et reçu dans le deuxième PC avec des deux réseaux différents :



PC 1 : wifi



PC 2 : 4G

▪ **Partie 4 :** En approchant du monde IdO, on peut se demander combien de protocoles nous pouvons utiliser pour communiquer des appareils IdO.
Il existe de nombreux protocoles tels que AMQP, CoAP, MQTT et l'omniprésence HTTP.

En essayant de rester dans le monde pratique, nous pouvons rapidement comparer deux d'entre eux : MQTT et HTTP.

Le protocole MQTT est conçu comme un protocole de connectivité machine à machine (M2M) et IdO.

▪ MQTT : Message Queuing Telemetry Transport.

Ce protocole est donc léger car il est souvent pris en charge par certains des plus petits appareils de mesure et de surveillance, et il peut transmettre des informations sur des

chemins atteignant parfois des réseaux intermittents. architecture de protocole MQTT simple (basée sur un courtier).

▪HTTP : Hyper Text Transfer Protocol.

la plupart d'entre nous le savent, la connexion HTTP entre le client (utilisateur Web) et le serveur Web.

C'est le protocole couramment utilisé pour Internet, il peut être le plus utilisé pour les appareils IoT qui doivent publier beaucoup de données. Il utilise l'en-tête IP normal pour le routage des paquets et les données ne sont pas cryptées avant la transmission

▪ Comparaison entre les protocoles MQTT et http

Tableau III.2 : Comparaison entre MQTT et HTTP.

Features	MQTT	HTTP
Formulaire complet	Message Queue Telemetry Transport	Hyper Text Transfer Protocol
Architecture	Il a une architecture de publication/abonnement. Ici, les appareils peuvent publier n'importe quel sujet et peuvent également s'abonner à n'importe quel sujet pour toutes les mises à jour.	Il a des moyens requête/réponse en architecture Client/serveur.
Protocole de couche supérieure	Il fonctionne sur TCP.	Il fonctionne sur TCP et UDP.
taille du message	Petite.	Grande
Format des messages	Binaire avec en-tête 2 octets	format ASCII
Diffusion des données	1 à 0/1/N	un à un seulement, plus de POST
Sécurité des données	Oui, il utilise SSL/TLS pour la sécurité	NON, donc HTTPS est utilisé pour assurer la sécurité des données
Complexité	Simple	Client plus complexe (parseur ASCII)
Chiffrement	Il crypte la charge utile, c'est-à-dire qu'il est indépendant de la charge utile	les données ne sont pas cryptées avant la transmission
Quand utiliser	si votre projet est de laisser le réfrigérateur communiquer avec le thermomètre pour adapter la pompe du moteur, vous pouvez utiliser le MQTT facilement	si vous avez besoin de collecter des données volumineuses du monde entier, vous pouvez penser à utiliser HTTP

- Le protocole MQTT est facile à utiliser. Si votre projet est de laisser le réfrigérateur communiquer avec le thermomètre pour adapter la pompe du moteur, vous pouvez utiliser le MQTT facilement, si vous avez besoin de collecter des Big data c'est-à-dire une grande quantité de données du monde entier, alors vous devriez penser à utiliser le Protocole HTTP.

- HTTP est digne et extensible. Mais MQTT est plus approprié lorsqu'il est appelé développement

III.5 CONCLUSION

Nous avons présenté dans ce chapitre en détails la réalisation de tout les composants de notre système , et nous avons vu leur implémentation étape par étape .

Par la suit on décrit comment mettre en place le broker mosquitto et à la fin on a utiliser un autre méthode pour envoyer et recevoir les données à distants par le FRED.

CONCLUSION GENERALE

Ce travail nous a permis de faire un conception d'un système d'acquisition des données en temps réel à base de l'IdO en utilisant le protocole MQTT, En outre, la mise en œuvre d'un système d'IdO pour le contrôle a distant du température et humidité nous a permis d'apprendre et d'exploiter plusieurs plateformes .

Nous a permis de se familiariser avec les nouvelles technologies que ce soit du hardware tel que le Raspberry pi qui est en plein essor actuellement pour pouvoir piloter des différents capteurs ou encore des objets et un micro-ordinateur très pratique pour implémenter et programmer les systèmes embarqués comme dans notre cas. Et en ce qui concerne la partie software, on s'est focalisé sur l'administration du système Node.js.

Nous avons aussi vu pourquoi avons-nous choisis le protocole MQTT pour l'envoie et la réception des messages entre les utilisateurs. En effet la rapidité et la fluidité étaient remarquable en ajoutant aussi la précision de destination. Il est important a rappeler aussi que nous avons vu que l'implémentation du protocole était très simple et rapide qui ce qui donne un autre point au profit du protocole.

En conclusion nous pourront dire que le protocole MQTT est un protocole simple a utiliser et a implémenter et reste l'une des meilleurs solutions a proposer dans le domaine de l'IdO.

REFERENCES BIBLIOGRAPHIQUES :

[1] DAVE, Evans. LInternet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?. Avril 2011, 12 p. (Cisco Internet Business Solutions Group (IBSG))

[2] E. Dobin et all. Analyse et perspectives de l'Internet des Objets. Juillet 2011, 132 p. (Les Cahiers de la RFID - Collection CITC)

[3] «Internet des objets : FUTURA TECH ». [En ligne]. Disponible sur :

https://www futura-sciences.com/tech/definitions/internet-internet-objets-15158/ .

[Consult éle :05/06/2020]

[4] COULON Alain. L'Internet des Objets Un gisement à exploiter. La Lettre d'ADELI n °78 – Hiver 2010. (PDF, 5 p [page 26 à30])

[5] CHALLAL, Yacine. Sécurité de l'Internet des Objets : vers une approche cognitive et Bibliographie 119 syst émique. Thèse de doctorat : Technologies de l'Information et des Syst èmes. France : Juin 2012, 78 p.

[6] Intel. Applications pour l'internet des objets dans l'industrie. [en ligne]. Disponible sur : https://www.intel.fr/content/www/fr/fr/internet-of-things/industry-solutions.html (Consult éle 02/02/2019)

[7] AUTOMATION&PLC KNOWLEDGE CENTER. Les domaines d'application de l'IoT et du machine to machine. [en ligne]. Disponible sur : Internet of Things Global Standards Initiative (itu.int)

[8] Objeko.com. Disponible sur : < https://www.objeko.com/selection-meilleurobjetconnecte-2016-1777/ > (consult éle 02/02/2019

[9] CHALLAL, Yacine. Sécurité de l'"Internet des Objets : vers une approche cognitive et Bibliographie 119 systémique. Thèse de doctorat : Technologies de l'"Information et des Syst èmes. France : Juin 2012, 78 p.

[10] Objeko.com. [en ligne]. Disponible sur : <https://www.amazon.fr/gp/product/B01LR8PG3O?ie=UTF8&linkCode=as2&camp=1634&creative=6738&tag=objeko-21&creativeASIN=B01LR8PG3O >(Consulté le 09/02/2019)

[11] DARTY. Disponible sur : < https://www.darty.com/darty-et-vous/high-tech/objets connectes/les-objets-connectes-au-jardin-vous-la-main-verte > (Consulté le 09/02/2019)

[12] CNET. Disponible sur :< http://www.cnetfrance.fr/special/objets-connectes 4000237247.hm > (Consulté le 09/02/2019

[13] Techno-Science. Microcontrôleur. [en ligne]. Disponible sur :< http://www.technoscience.net/?onglet=glossaire&definition=6737 > (Consulté le 11/02/2019)

[14] LeClubGQ. Les 10 objets High-Tech les plus attendus en 2018. [en ligne]. Disponible sur : <http://www.gqmagazine.fr/lifestyle/high-tech/diaporama/les-10-objets-high-tech-les-plus-attendus-en-2018/48067#un-deux-ou-trois-iphone-x-1 >. (Consulté le 20/03/2019)

[15] Thomas et Mathieu DUBAËLE. Disponible sur : : < https://www.apprendre-enligne.net/crypto/passecret/resistances.pdf > [PDF]. (Consulté le 21/02/2019)

[16] NDIAYE , Yacine. IoT Cloud Platforms . [en ligne]. Disponible sur : https://air.imag.fr/index.php/VT2016_IoT_cloud_platforms#Comparaison_de_quelques _plate formes IoT (Consulté le 23/02/2019)

[17] TechMoka. [en ligne]. Disponible sur : < http://techmoka.com/lentillesconnectees-samsung/ >. (Consulté le 20/03/2019)

[18] IoT Agenda. [en ligne]. Disponible sur : <http://internetofthingsagenda.techtarget.com/definition/microcontroller > (Consulté le 10/02/2019)

[19] ADJIBA, Brahim, CHALGHOUM, Abdelmonaim. Commande des équipements électriques par microcontrôleurs "Simulations et Réalisations". Mémoire de Master

académique, Commande Électrique. D'El-Oued: Université Echahid Hamma Lakhdar d'El-Oued, 2015, 51p

[20] «Get started with Raspberry Pi: Site Officiel Raspberry Pi ».[En ligne]. Disponible sur :https://projects.raspberrypi.org/en/pathways/getting-started-with-raspberry-pi .[Consulté le :23/06/2020]

[21] GO TRONIC. Robotique et composants électroniques. [en ligne]. Disponible sur :

< https://www.gotronic.fr/cat-capteurs-1114.htm >. (Consulté le 17/02/2019)

[22] SANOGO, Yacouba.. MELITI Nedjema. Conception et fabrication de capteurs et de leur techniques d'nterrogation por des applications dans Les domaines de la santé et de l'environnement. Thèse de doctorat. Physique. France : laboratoire de photonique quantique et moléculaire, 2013, 180 p

[23] https://www.robot-maker.com/shop/capteurs/31-dht22-31.html

[24] Carles Anton-Haro and Mischa Dohler,Machine-to-machine (M2M) Communications Architecture, Performance and Applications, Elsevier,2015, p 91.

[25] http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt

[26] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS

[27] https://www.oasis-open.org/news/pr/oasis-mqtt-internet-of-things-standard-now-

approved-by- iso-iec-jtc1.

[28] Protocol Analysis - oneM2M, page 21, URL: http://www.onem2m.org/images/files/deliverables/TR-0009-Protocol_Analysis-V0_7_0_1.pdf

[29] Protocol Analysis - oneM2M, page 21, URL: http://www.onem2m.org/images/files/deliverables/TR-0009-Protocol_Analysis-V0_7_0_1.pdf

[30] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. (consulté en Aout 2015).

[31] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things : A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, Article, 17(4) :2347–2376, 2015.

[32] La Rédaction, « Internet des Objets : bien comprendre MQTT », Le MagIT, juillet 2016.

[33] mqtt-topics-best-practices, URL: http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt- topics-best-practices.

[34]Sébastien Bonnet « MQTT : Un protocole dédié pour l'IoT », http://www.digitaldimension.solutions/blog/avis-d-experts/2015/02/mqtt-un-protocole-dedie-pour-liot/ 27 février 2015.

[35] « Creating your first flow : Site Officiel Node-Red ». [En ligne]. Disponible sur : http://nodered.org/docs/ . [Consulté le :03/07/2020]