الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Mention Électronique
Spécialité Électronique des Systèmes embarqués

présenté par

MESBAH MOHAMED BACHIR

&

CHELHA CHEMS EDDINE

COMMANNDE A DISTANCE D'UN ROBOT MOBILE VIA L'IoT

Proposé par : Kabir Yacine

Année Universitaire 2017-2018

REMERCIMENET

On remercie ALLAH, le tout puissant, qui nous a donné la force, la volonté et surtout le courage pour accomplir ce modeste mémoire

Au terme de ce travail, on tient à remercier particulièrementMr KABIR YACINE, notre encadreur, notre enseignant de l'université SAAD DAHLEB BLIDA pour son encadrement, Sa disponibilité, ses critiques et ses remarques pertinentes .Il nous a conseillé et guidé du début du projet à sa fin. Vous nous avez toujours réservé un chaleureux accueil, malgré vos obligations et les contraintes professionnelles. Vos talents ainsi que vos compétences et votre sens du devoir nous ont marqué à jamais. Vos encouragements inlassables, votre amabilité, votre gentillesse et votre patience méritent toute notre attention. Veuillez trouver ici l'expression de notre estime et notre considération.

On voudrait aussi adresser nos sincères remerciements à tous les enseignants Du département d'électronique, nous remercions également, les membres du jury de nous avoir fait l'honneur de juger ce mémoire.

Et finalement on tient à remercier chaleureusement et respectivement tous ceuxqu'ont contribués de près ou de loin à la réalisation de ce modeste projet de fin d'étude.

ملخص

ينطويهذاالمشروععلىإجراءبحوثمستفيضةبشأنإنترنتالأشياء. وكذلكعلىجوانبهاالمختلفة.
NodeMcu ESP8266 واختيارب روتوكولالبياناتالمعتمدةلهذاالمشروع،استنادااإلىأحدثميكروكنترولر
وريسبريبي،ومنصاتهالمرتبطة،تمتخصيصالجزءالثانيوالثالثلبيئةالعملوالموادالمستخدمةلتصميموتنفيذالروبوتالمحمولللضمانال
.thnigs سيطرةالكاملةوالسماحللكاميراالمدمجةلإعادةالإرسالفيالوقتالحقيقيعلىأساسالإنترنتمن
كلماتسيطرعليهتطبيقاتالهاتفالمحمول

Résumé

Ce projet consiste à faire des recherche approfondit sur les IoT. Ainsi que sur leurs différents aspects. et le choix du Protocol de donnée adopté pour ce projet, basé sur les derniers microcontrôleurs que sont le NodeMcu ESP8266 et le RASPBERRY PI ZERO, et ses différentes plates-formes associés ,la deuxième et la troisième partie a été consacré à l'environnement du travail et le matériels utilisé pour la conception et la réalisation du robot mobile afin d'assurer son contrôle total et permettre à la camera intégré de retransmettre en temps réel se basant sur l'internet of thnigs . Le tout contrôlé par une application mobile.

Mots clés : NodeMcu ESP8266, L'Internet Of Things (IOT), RASPBERRY PI ZERO

Abstract:

This Project involves doing extensive research on IoTs. As well as on their different aspects. and the choice of data protocol adopted for this project, based on the latest microcontrollers NodeMcu ESP8266 and RASPBERRY PI ZERO, and its associated platforms, the second and the third part was dedicated to the environment of the work and materials used for the design and implementation of the mobile robot to ensure full control and allow the integrated camera to retransmit in real time based on the Internet of thnigs. All controlled by a mobile application.

Keywords : NodeMcu ESP8266, The Internet Of Things (IOT), RASPBERRY PI ZERO

Listes des acronymes et abréviations

FPGA : Fild Programmable Gate Aray

ASIC : Application Specific Inregrated Circuits

IOT : Internet Of Things

RFID : Radio Frequency Identification

TCP : Transmission Control Protocol

IP :internet Protocol

WNS :wireless sensor network

M2M : machine to machine

BigData :data-processing application software

MIT : *Massachusetts Institute of Technology*

MQTT : Message Queuing Telemetry Transport

Ios :iphone operating system

SSL : Secure Sockets Layer

TLS : Transport Layer Security

QoS : Quality of Service

JMS : Java message service

J2EE : Java 2 Platform Entreprises Edition

STOMP : Simple Text Orientated messaging Protocol

AMQP : Advanced Message Queuing Protocol

ARM : Reduced Instruction Set Compute

REST : Representational State Transfer

LGPL : Lesser General Public License

API :application programming interface

HTTP: The Hypertext Transfer Protocol

CPU :central processing unit

ADSL : Asymmetric digital subscriber line

GSM : Global System for Mobile communications

UMTS : Universal Mobile Télecommunications system

IdO : Internet des objets

SoC : System On Chip

UART :universal asynchronous receiver-transmitte

PWM : Pulse Width Modulation

I2C : Inter-Integrated Circuit

I2S : Inter-IC Sound

ARM : Reduced Instruction Set Compute

Mini HDMI : miniHigh Definition Multimedia Interface

USB : Universal Serial Bus

CSI : Camera Serial Interface

SSH : SecureShell

Introduction général :

Un système embarqué est un système électronique et informatique autonome qui est utilisé pour réaliser des taches prédéfinie en temps réel, possédant une taille limité et ayant une consommation énergétique restreinte.

Les systèmes embarqués sont désormais utilisé dans divers application tels que les transports (avion, automobile, ferroviaire), dans les appareils électriques et électronique (appareil photo, électroménagers, téléphones portables), dans la distribution d'énergie, et dans la robotique.

Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numérique programmable FPGA (Fild Programmable Gate Aray), des circuits dédiées à des applications spécifique ASIC (Application Specific Inregrated Circuits), des microcontrôleurs ou des modules analogique sont de plus en plus utilisé.

Au fil du temps, on a intégré l'ordinateur dans différents objets de notre vie quotidienne, de plus avec l'internet les machines liées aux WEB sont conçus pour exécuter, puis penser et aujourd'hui elles apprennent à percevoir, sentir et réagir, c'est dans ce contexte que nous orientons notre projet qui est lié aux objets connecté à l'internet connues comme IOT (Internet Of Things).

En raison de ses applications l'IOT a retenu beaucoup notre intention, dans notre projet nous voulons apporter notre contribution à cet égard en proposant une solution aux problèmes rencontrée lorsque les divers objets dans le monde de l'IOT s'interconnectent et collaborent.

Chapitre 1 : Internet of things

1 .1 introduction :

L'Internet Of Things (ou IOT) se traduit à l'heure actuelle par l'accroissement du nombre d'objets connectés, c'est-à-dire d'appareils possédant une identité propre et des capacités de calcul et de communication de plus en plus sophistiquées : téléphones, montres, appareils ménagers, etc. Ces objets embarquent un nombre grandissant de capteurs et d'actionneurs leur permettant de mesurer l'environnement et d'agir sur celui-ci.

Les appareils électronique et les ordinateurs et internet sont devenue une nécessité dans notre vie quotidienne.

Au fil des temps, on a intègres l'ordinateur dans différent objets de notre vie, de plus, avec le web, ses objets peuvent se connecter et communiquer entre eux ,grâceà cette émergence du web les machines liées à l'internet sont conçues pour exécuter puis pensée , et aujourd'hui, elles apprennentà percevoir, sentir et réagir c'est dans ce contexte que nous orientons notre projet de fin d'étude sur le concept de liées les objet à internet connu de nos jours comme <internet of things> (IOT).

L'IOT a retenue beaucoup d'attention autant dans le milieu universitaire et de l'industrie, dans notre Projet de fin d'étude nous allons l'utiliser pour contrôler un robot mobile.

1.2 Problématique :

L'homme a de plus en plus besoin d'avoir recours aux machines, aux robots et aux Object capable de fournir des services avec peu d'effort, voire sans aucune intervention humaine, et pour faciliter ça, en va essayer de résoudre :

- Comment Contrôler Un robot mobile Avec L'IOT.
- Recevoir La vidéo en temps réel pour contrôler le robot.

1.3 Objective :

L'objectif sera de construire un petit robot mobile entièrement commandé et contrôlé par le NodeMcu ESP8266 que ce soit les moteurs ou les servomoteurs via une plate-forme de l'IOT, tout en intégrant une petite caméra qui transmettra le flux vidéo en temps réel sur un smartphone ou un ordinateur.

1.4 Fonctionnement de L'IOT

L'internet of things permet l'interconnexion des différents Object intelligent via l'internet, ainsi pour son fonctionnement plusieurs systèmes technologiques sont nécessaires.

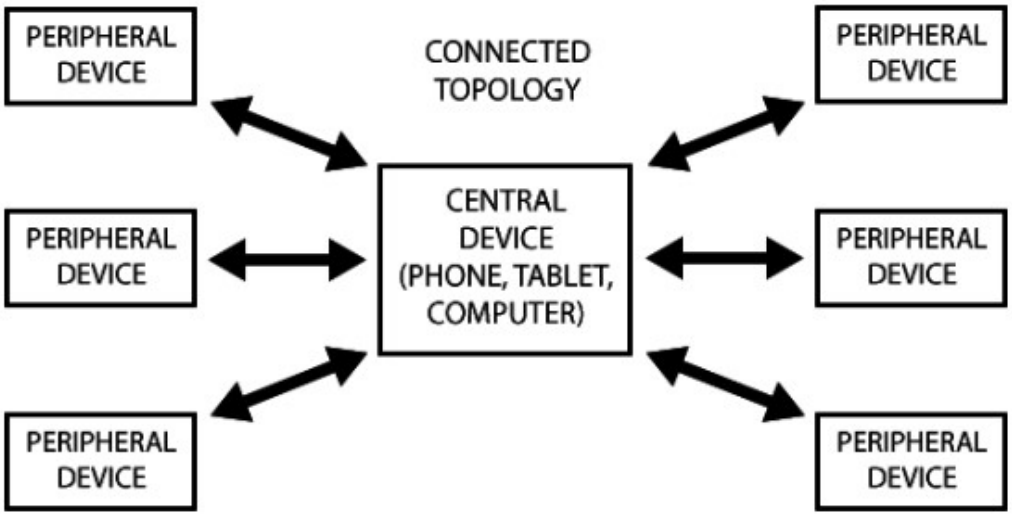


Figure ***Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.****..1.1 fonctionnement de L'IOT*

L'IOT désigne divers solution technique (RFID, TCP/IP, technologie mobile, etc. …), qui permettent d'identifier des objets, de capter, stocker, traiter et transférer des données dans les environnements physique, mais aussi entre des contextes physiques et des univers virtuels (1).

En effet bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de L'IOT nous mettent l'accent sur quelques 'un qui sont :

- WNS (wireless sensor network) : c'est un ensemble de nœuds qui communiquent sans fil et qui sont organiser en un réseau coopératif chaque nœud possédé une capacité de traitement et peut contenir différent type de mémoire un émetteur-récepteur RF, et une source d'alimentation, comme il peut aussi tenir compte divers capteurs et des actionneurs. [2].
- RFID (Radio Frequency Identification) : le terme RFID englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. C'est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio [3]. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance [4].
- M2M (machine to machine) : c'est l'association des technologies de l'information et de la communication avec des objets intelligents dans le but de donnée à ces dernier les moyen d'interagir sans intervention humaine avec le système s'information d'information d'une organisation ou d'une entreprise [5].

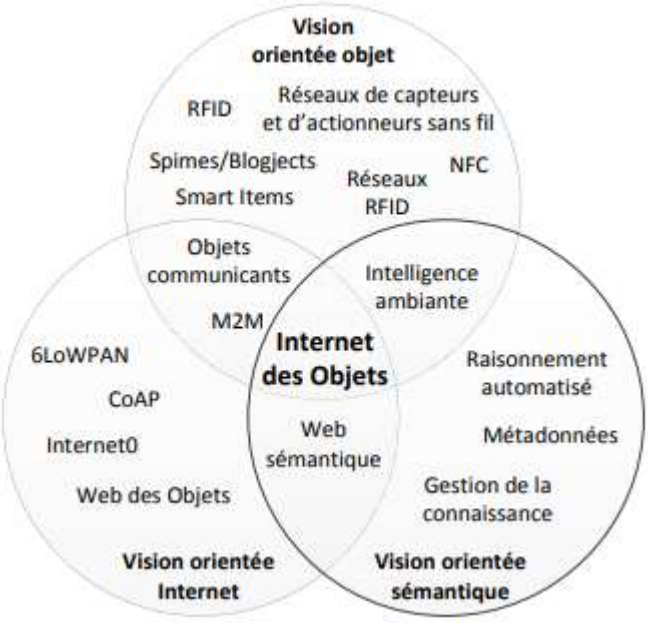


Figure 1.2différents technologie utilisé dans IOT

1.5 Domaine d’utilisation IOT :

Plusieurs domaines d’application sont touchés par l'IoT. Dans leur article, Gubbi et al. [6] ont classé les applications en quatre domaines :

- Le domaine personnel.
- Le domaine du transport.
- L’environnement.
- L’infrastructure et les services publics

Comme le schéma ci-dessous le montre :

*Figure **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**.3: Schéma de Domaine d'utilisation de L'IOT*

1.6 Robot mobile :

L’objective principale du robot mobile consiste à réaliser un mouvement en reliant un point source a un point destination, la liberté de mouvement lui confère une autonomie qui lui permet de trouver d’autre utilisation la nature de sa tâche, ainsi que des performances recherché.

*Figure **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**.4: exemple robot mobile*

1.7 Conclusion :

Dans ce présent chapitre qui a été consacré aux principales approches utilisées dans le domaine de l'internet of things (IOT), une présentation général de l'IOT et son fonctionnement, fixé quelques problématiques pour le control d'un robot mobile à partir de l'IOT, et les quelques objectifs à atteindre dans ce projet de fin d'étude

Chapitre 2 : Etat de l'art

2.1 Introduction :

L'IOT peut être abordée sous un angle de données comme sous un angle infrastructure. Dans le premier cas, la problématique concerne le traitement et la transformation d'une masse imposante de données en informations pertinentes (notion de BigData). En effet, les nœuds de ce réseau capturent des informations de l'environnement qui peuvent aller des caractéristiques classiques comme la température et la luminosité à la fréquentation d'un site ou l'intensité du champ magnétique. Regrouper les sources pour corréler les informations peut donner le jour à des applications novatrices. De même, dans ce contexte de crowdsourcing, s'assurer que les données observées sont les plus appropriées est un autre défi. Le développement d'un middleware devient l'enjeu principal puisqu'il rend transparentes les différences matérielles dans la mise en relation avec l'environnement. La plateforme Vital IoT

Approcher l'IoT avec une vision axée sur l'infrastructure emmène autant à développer des protocoles permettant l'interaction sécurisée et à distance avec des équipements, que la

mise en place de structures permettant l'étude et le test de ces solutions. Nous nous intéresserons plutôt aux aspects communication de la problématique de l'Internet des Objets. La suite de cette section présentera le Protocol de données adopté et un panorama des plateformes actuellement disponibles, ce qui nous permettra de positionner clairement notre contribution. Nous proposons également un tour d'horizon des plateformes disponibles

2.2 Historique de L'IOT :

L'histoire des objets connectés débute en 1999 lorsque **Kevin Ashton**, pionnier de la technologie **RFID** (Radio Frequency IDentification − Technologie d'identification automatique), invente l'expression **"Internet of things" .** Cette même année, le concept naît aux États-Unis et particulièrement au **MIT** (*Massachusetts Institute of Technology*). Ce laboratoire est dédiéà la création d'objets connectés à l'aide de l'identification par radiofréquence et les réseaux de capteurs sans fil.

L'Internet des objets est apparu dans le cadre d'une tendance lourde, issue de la mécanisation et la standardisation, appliquée à l'automatisation du traitement du document et de l'information sur support matériel puis numérique (dont au service de la production et recherche documentaire). Apparu aux États-Unis, il s'est rapidement diffusé avec la mondialisation, aboutissant à connecter des machines à des serveurs capables de les superviser (ces machines étant notamment des ordinateurs mis en réseau dans ce que certains ont nommé l'« Internet des machines »). Peu à peu des objets ont été modifiés (avec des puces RFID par exemple) ou conçus pour « parler le protocole IP », devenant des « objets connectés », reliés à des serveurs centralisés ou capables de communiquer entre eux ou avec des réseaux de serveurs et divers acteurs, d'une manière de moins en moins centralisée

En 2003, **Rafi Haladjian**, inventeur du premier opérateur Internet en France (**Francenet**), créé la lampe DAL. Une lampe d'ambiance équipée de 9 LEDs, proposant différentes couleurs et commercialisée à 790 euros. Deux ans plus tard, l'entreprise du créateur lance le Nabaztag, un lapin connecté en Wi-Fi qui lit les mails à haute voix, émets des signaux visuels et diffuse de la musique

C’est néanmoins en 2007 que le phénomène des IOT a pris de l’ampleur, avec la démocratisation des **Smartphones** et la sortie du premier **iPhone** par **Apple**. La dématérialisation est en marche. (**10)**

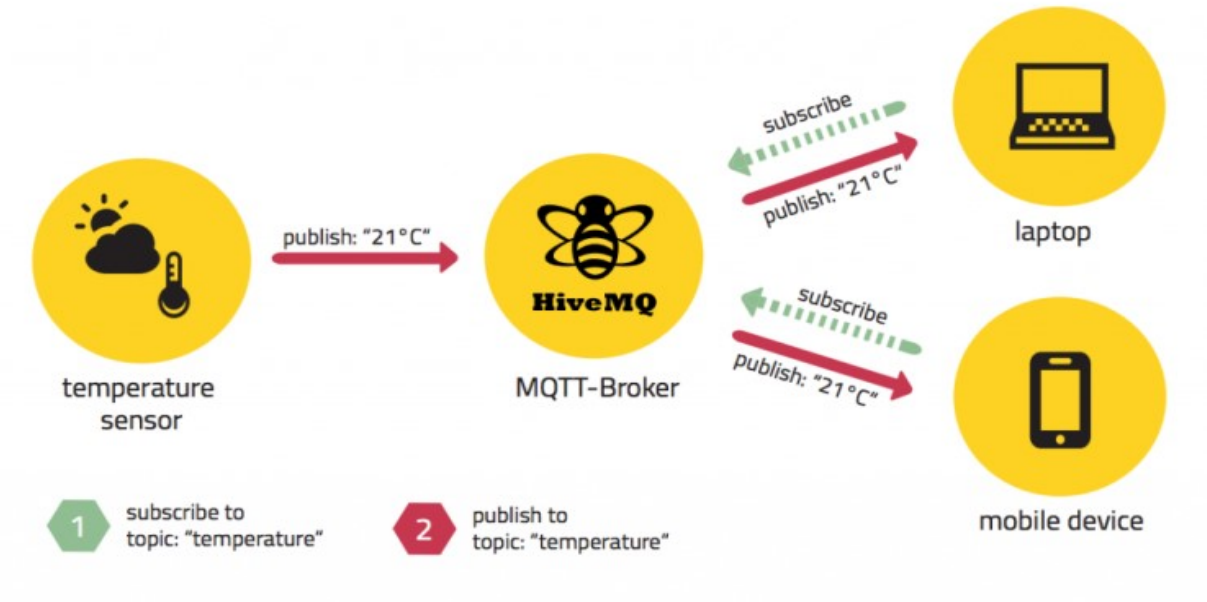
2.3 Les plates-formes IOT :

Nous assistons depuis quelques années à l’éclosion de plusieurs plateformes de développement, dans notre travaille on est passé Par de nombreux plateforme qui adopte le Protocol de données MQTT, est on a testé plusieurs plates-formes pour commander notre robot mobile est recevoir des informations de celui-là, envoyer la donnée est la recevoir

2. 3.1 MQTT :

MQTT(Message Queuing Telemetry Transport) est un protocole de messagerie publish-subscribe basé sur le protocole TCP/IP.

De très nombreuses bibliothèques sont disponibles pour programmer des clients MQTT, pour la plupart des langages (C, C++, Java, JavaScript, PHP, Python, etc) et sur la plupart des plates-formes (GNU/Linux, Windows, iOS, Android, Arduino…).



2.3.1.1Histoire du MQTT :

MQTT a été inventé en 1999 Il a été initialement développé par Andy Stanford-Clark (IBM) et Arlen Nipper (EuroTech), qui cherchaient à avoir un protocole de messagerie léger pour faire

communiquer des machines dans un environnement ou les déconnexions sont fréquentes, puis offert à la communauté Open Source.

2.3.1.2 Fonctionnements du MQTT :

Une session MQTT est divisée en quatre étapes : connexion, authentification, communication et terminaison.

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.

Les ports standards sont les suivants : 1883 pour la communication non chiffrée et 8883 pour la communication chiffrée utilisant SSL/TLS. 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.

MQTT est considéré comme un protocole léger parce que les messages ont tous une faible empreinte logicielle. Chaque message se compose d'un en-tête fixe (2 octets), d'un en-tête variable facultatif, d'une charge utile de message limitée à 256 Mo et d'un niveau de qualité de service. Les trois niveaux de qualité de service déterminent la façon dont le protocole MQTT gère le contenu. Bien que les niveaux plus élevés soient plus fiables, ils sont également plus gourmands en termes de latence et de bande passante. Les clients abonnés peuvent spécifier le niveau de QoS maximal qu'ils souhaitent recevoir.

Le niveau de qualité de service le plus simple est le service non confirmé. Ce niveau utilise une séquence de paquets PUBLISH : l'éditeur envoie un message une seule fois au broker et ce dernier transmet ce message une seule fois aux abonnés. Aucun mécanisme ne garantit la réception du message et le broker ne l'enregistre pas non plus.

Le deuxième niveau de service est le service confirmé. Ce niveau utilise une séquence de paquets PUBLISH/PUBACK entre l'éditeur et son broker, ainsi qu'entre le broker et les abonnés.

Un paquet de confirmation vérifie que le contenu a été reçu et un mécanisme de renvoi du contenu d'origine est déclenché si l'accusé de réception n'est pas reçu en temps voulu. Cela signifie que l'abonné peut recevoir plusieurs copies du même message

Le troisième niveau de QoS est le service garanti. Ce niveau délivre le message avec deux paires de paquets. La première est appelée PUBLISH/PUBREC et la seconde, PUBREL/PUBCOMP. Les deux paires s'assurent que quel que soit le nombre de tentatives, le message ne sera délivré qu'une seule fois.

Au cours de la phase de communication, un client peut effectuer les opérations suivantes : publication, abonnement, désabonnement ou Ping.

2.3.1.3 Plateforme utilisant MQTT :

Il existe de nombreux brokers MQTT disponibles, ils varient dans leurs fonctionnalités et certains d'entre eux mettent en œuvre des fonctionnalités additionnelles.

Les principaux brokers open-sources sont :

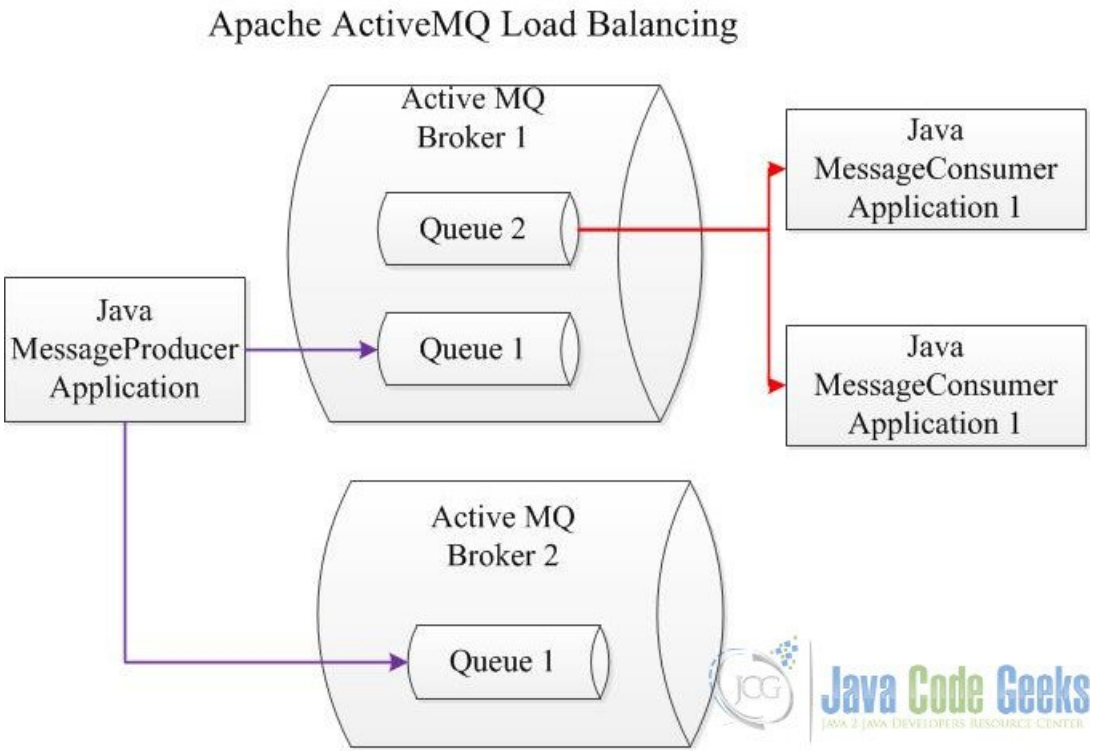
2.3.1.2.1 Apache active MQ :

Apache ActiveMQ ™ est le serveur de messagerie et d'intégration d’open source le plus populaire et le plus puissant.

Apache ActiveMQ est rapide, prend en charge de nombreux clients et protocoles, est livré avec des modèles d'intégration d'entreprise faciles à utiliser et de nombreuses fonctionnalités avancées tout en prenant en charge JMS 1.1 (Java message service) et J2EE 1.4 [6].

Le projet ActiveMQ a été créé à l'origine par ses fondateurs de LogicBlaze en 2004, en tant que courtier de messages open source, hébergé par CodeHaus. Le code et la marque ActiveMQ ont été donnés à la Fondation Apache Software en 2007, où les fondateurs ont continué à développer la base de code avec la communauté étendue d'Apache.

ActiveMQ utilise plusieurs modes de haute disponibilité, notamment des mécanismes de verrouillage pour les lignes de fichiers et les bases de données, le partage de stockage de persistance via un système de fichiers partagé ou une réplication réelle à l'aide d'Apache ZooKeeper. Un robuste mécanisme de mise à l'échelle horizontale appelé réseau de courtiers [7] est également pris en charge dès sa sortie de la boîte. Dans l'entreprise, ActiveMQ est connu pour sa configuration flexible et sa prise en charge d'un nombre relativement important de protocoles de transport, notamment OpenWire, STOMP, MQTT, AMQP, REST et WebSockets. [8]



2.3.1.2.2 : OW2 JORAM :

JORAM est une implantation open-source (LGPL) d'un intergiciel à messages (MOM), il implante l'API Java Message Service(JMS) 1.1. Et 2.0 est certifié conforme JMS par Oracle. JORAM est disponible gratuitement au téléchargement depuis mai 2000.

JORAM offre de nombreuses fonctionnalités avancées comme le clustering de destinations ou la haute-disponibilité, il est aussi accessible au travers d'une API C/C++ et d'un client léger J2ME. Il implante les protocoles MQTT (Message Queuing Telemetry Transport) et AMQP (Advanced Message Queuing Protocol).

JORAM est l'implantation JMS de référence du serveur d'application Java EE JOnAS (certifié Java EE 5). Il est aussi intégré au bus d'entreprise orienté services PEtALS et à l'infrastructure SCA FraSCAti. [6].

2.3.1.2.3 : Mosquito :

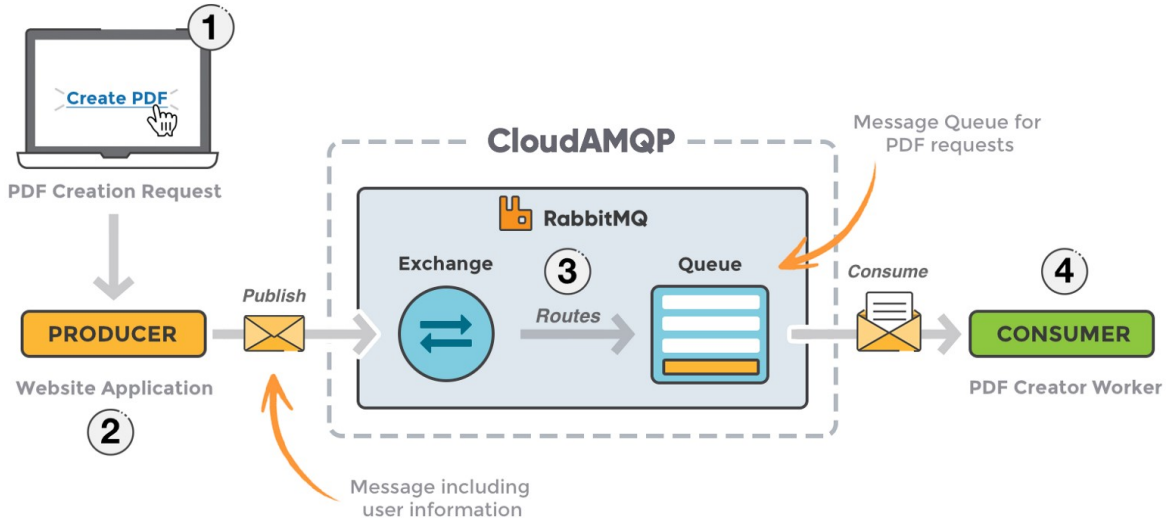
Eclipse Mosquitto est un courtier de messages open source (sous licence EPL / EDL) qui implémente les versions 3.1 et 3.1.1 du protocole MQTT. Mosquitto est léger et peut être utilisé sur tous les appareils, des ordinatrices mono cartes basses consommation aux serveurs complets.
Le protocole MQTT fournit une méthode légère d'exécution de la messagerie à l'aide d'un modèle de publication / abonnement. Cela le rend approprié pour la messagerie de l'Internet of things , par exemple avec des capteurs de faible puissance ou des appareils mobiles tels que des téléphones, des ordinateurs embarqués ou des microcontrôleurs.[7].
Le projet Mosquitto fournit également une bibliothèque C pour la mise en œuvre des clients MQTT, ainsi que les très populaires clients MQTT mosquitto_pub et mosquitto_sub.

Mosquitto fait partie de la Fondation Eclipse

2.3.1.2.4 RabbitMQ :

RabbitMQ est le message broker open source le plus largement déployé.

Avec plus de 35 000 déploiements de RabbitMQ dans les petites entreprises et les grandes entreprises, RabbitMQ est open source message broker le plus populaire.
RabbitMQ est léger et facile à déployer sur site et dans le cloud. Il prend en charge plusieurs protocoles de messagerie, il utilise le protocole AMQP (Advanced Message Queuing Protocol). RabbitMQ peut être déployé dans des configurations distribuées et fédérées pour répondre à des exigences de haute disponibilité et de haute disponibilité[8].



2.3.1.2.5 ThinkSpeak :

ThingSpeak est une API et une application open source pour l'internet of things« IOT », permettant de stocker et de collecter les données des objets connectés en passant par le protocole HTTP via Internet ou un réseau local.

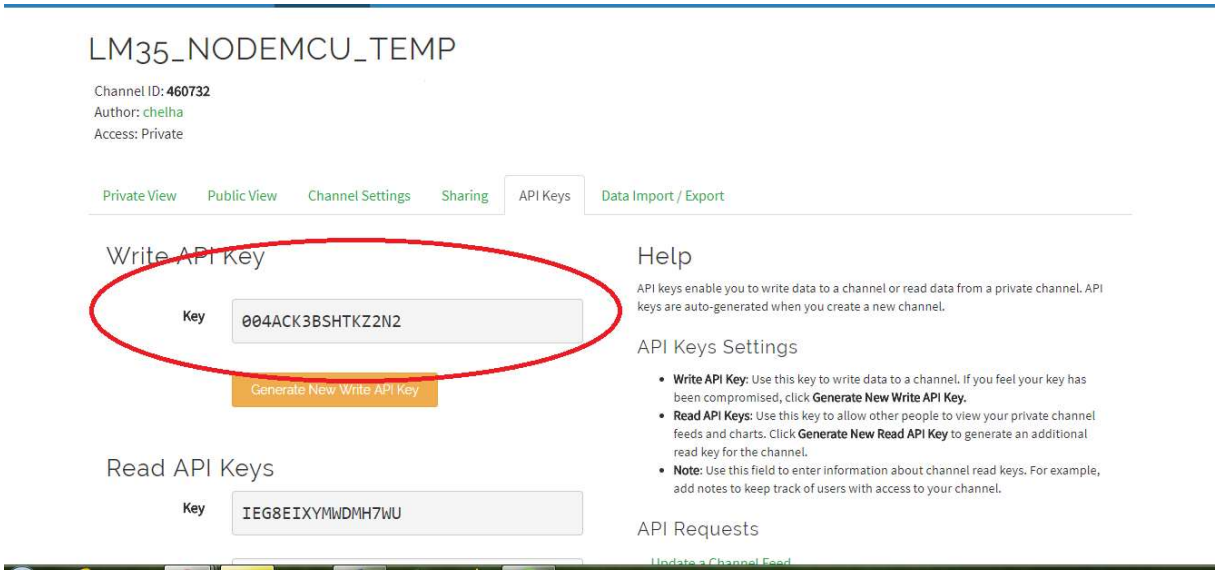
Avec thinkspeak on peut faire plusieur chose :

- API ouverte
- Collecte de données en temps réel
- Donnés de géolocalisation
- Traitement des données
- Visualisations de données
- Messages d'état des circuits
- Plugins

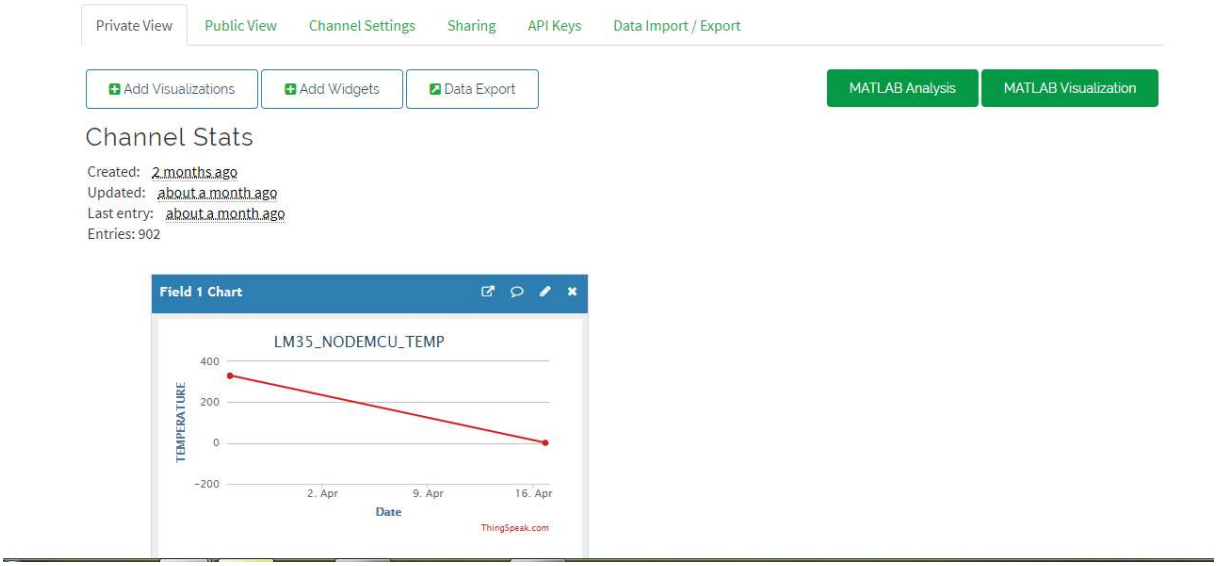
Et les données de capteur peuvent être envoyées à thinkspeak à partir d'ARDUINO, RASPBERRY PI, IoBridge

2.3.1.5.1Fonctionnement de Thingspeak :

Pour avoir L'API keys pour le faire entrer dans le programme ARDUINO pour communiquer entre les capteurs et ThingSpeak est ça à travers une carte NodeMcu et l'internet



Apres cela il faut varier le capteur de température qui les analyses et collectes leurs donnée



2.1 Node-Red :

Node-RED est un outil de programmation pour le câblage de périphériques matériels, d'API et de services en ligne de manière nouvelle et intéressante.

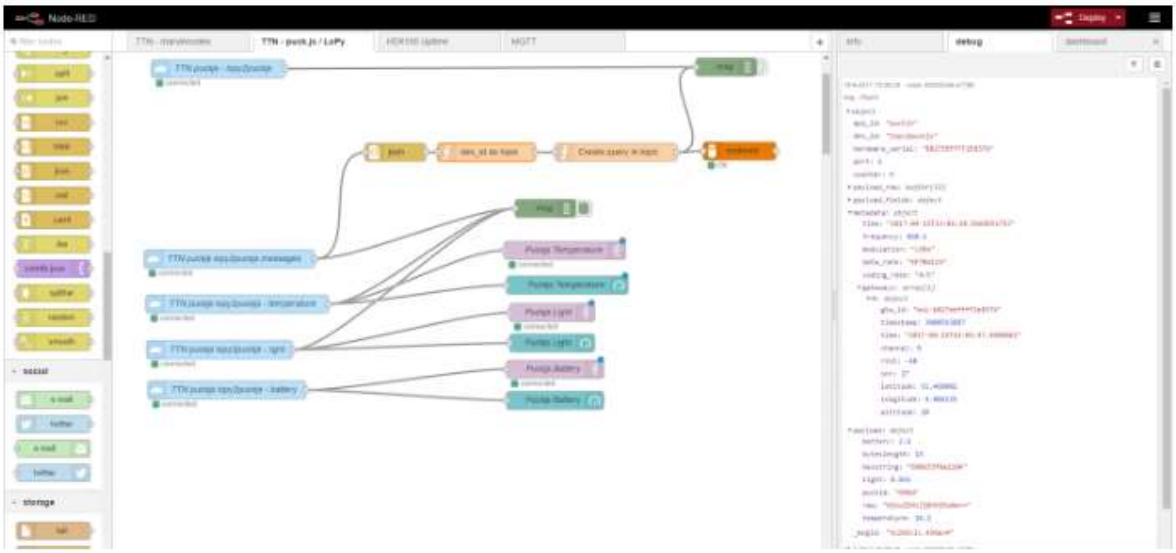
Il fournit un éditeur basé sur un navigateur qui facilite le câblage des flux à l'aide de la vaste gamme de nœuds de la palette qui peut être déployée en un seul clic

Node-RED fournit un éditeur de flux basé sur un navigateur qui facilite le câblage des flux à l'aide de la vaste gamme de nœuds de la palette. Les flux peuvent ensuite être déployés dans l'environnement d'exécution en un seul clic.

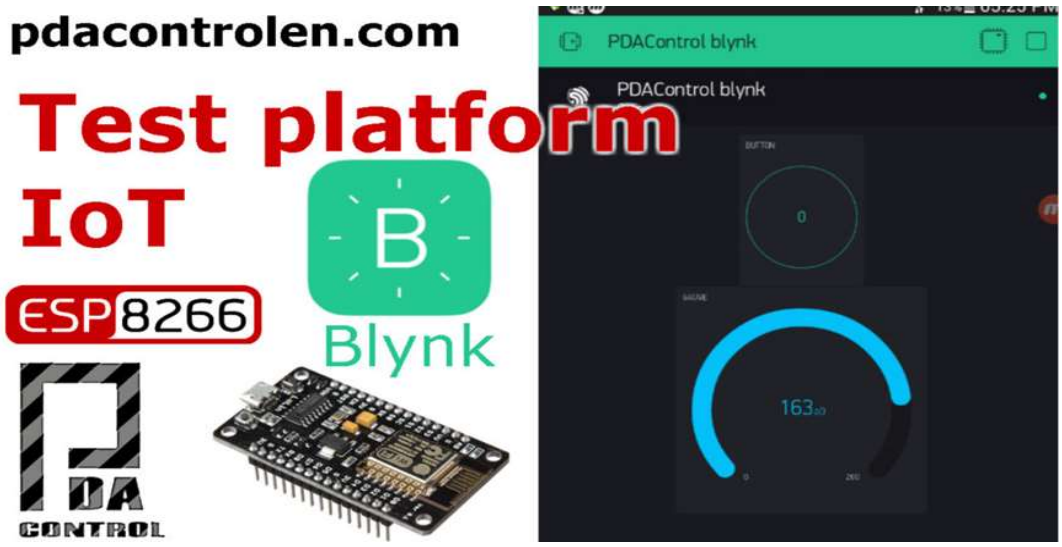
Les fonctions JavaScript peuvent être créées dans l'éditeur à l'aide d'un éditeur de texte enrichi.

Une bibliothèque intégrée vous permet d'enregistrer des fonctions, des modèles ou des flux utiles pour la réutilisation. L'environnement d'exécution léger est construit sur Node.js, en tirant pleinement parti de son modèle non bloquant, piloté par les événements. Cela le rend idéal pour fonctionner à la périphérie du réseau sur du matériel à faible coût tel que le Raspberry Pi ainsi que dans le Cloud.[]

Avec plus de 225 000 modules dans le référentiel de paquets de Node, il est facile d'étendre la gamme de nœuds de la palette pour ajouter de nouvelles fonctionnalités.[9]



2.3 .1.2 BLYNK :



Blynk a été conçu pour l'Internet des objets. Il peut contrôler le matériel à distance, il peut afficher les données des capteurs, il peut stocker des données, les visualiser et faire beaucoup d'autres choses intéressantes.

Il y a trois composants principaux dans la plate-forme:

Blynk App - vous permet de créer des interfaces incroyables pour vos projets en utilisant divers widgets que nous fournissons.

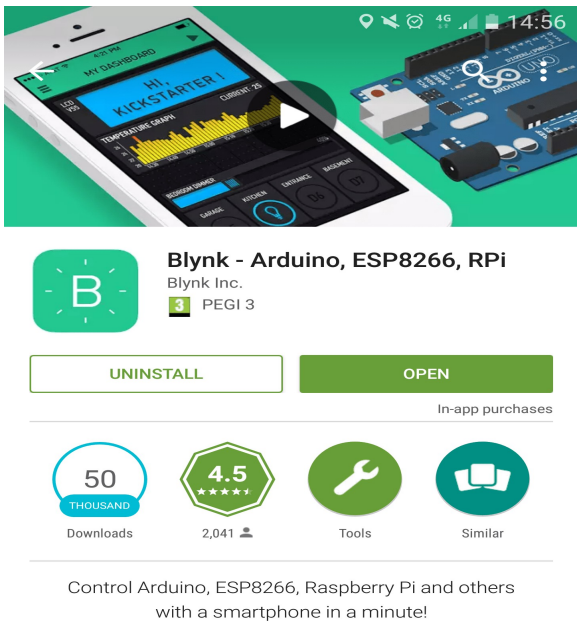
Blynk Server - responsable de toutes les communications entre le smartphone et le matériel. Vous pouvez utiliser notre Cloud Blynk ou exécuter votre serveur Blynk privé localement. Il est open-source, pourrait facilement gérer des milliers d'appareils et peut même être lancé sur un RASPERRY Pi.

Bibliothèques Blynk - pour toutes les plates-formes matérielles populaires - permettent la communication avec le serveur et traitent toutes les commandes entrantes et sortantes.

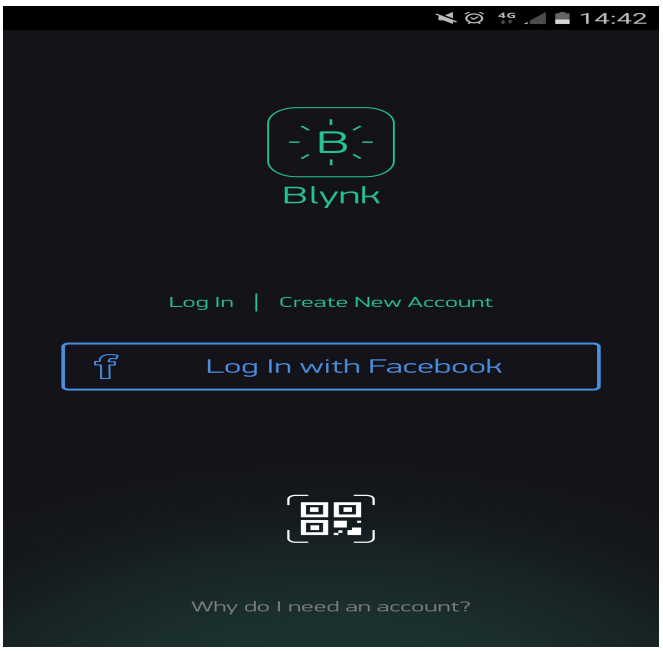
Maintenant, imaginez: chaque fois que vous appuyez sur un bouton dans l'application Blynk, le message se déplace pour espacer le Cloud Blynk, où il trouve par magie son chemin vers votre matériel. Il fonctionne de la même manière dans la direction opposée et tout se passe dans un angle d'œil

La mise en place de l’application :

Tous d’abord télécharger l’application Blynk sur le store d’un smartphone



Démarrer l’application et créer un compte



2 .4 les moyens de communication :

2.4.1 LORA :

Le protocole LoRaWAN est un protocole de communication pour l'internet des objets qui utilise une technique de modulation par étalement de spectre de type Chirp spread spectrum propriétaire appelée LoRa. Ce protocole se veut simple, peu coûteux à implémenter et économe en énergie plutôt que permettant des débits élevés. La cible de LoRaWAN est clairement les communications longues portées à bas coût et basse consommation plutôt que les communications à débit élevé qui sont plus consommatrices en ressource CPU et en énergie. En effet, les défis concernant l'interconnexion des objets résident dans leur coût, leur autonomie ainsi que leur nombre d'un point de vue réseau. Ce faible coût est obtenu par l'utilisation d'une architecture en étoile (plus simple qu'une architecture maillée), une technique de modulation plus simple à implémenter que celle des réseaux cellulaires classiques ce qui réduit le coût des composants électroniques qui lui sont dédiés ainsi que l'utilisation de bandes de fréquences libres (ne nécessitant pas de payer pour leurs utilisations)

2.4.1.1 architectures :

Un réseau LoRaWAN est constitué d'équipements sans-fils basse consommation qui communiquent avec des serveurs applicatifs au travers de passerelles. La technique de modulation utilisée entre les équipements et les passerelles est LoRa. La communication entre les passerelles et les serveurs est établie via le protocole IP au moyen d'un réseau de collecte Ethernet ou 3G.

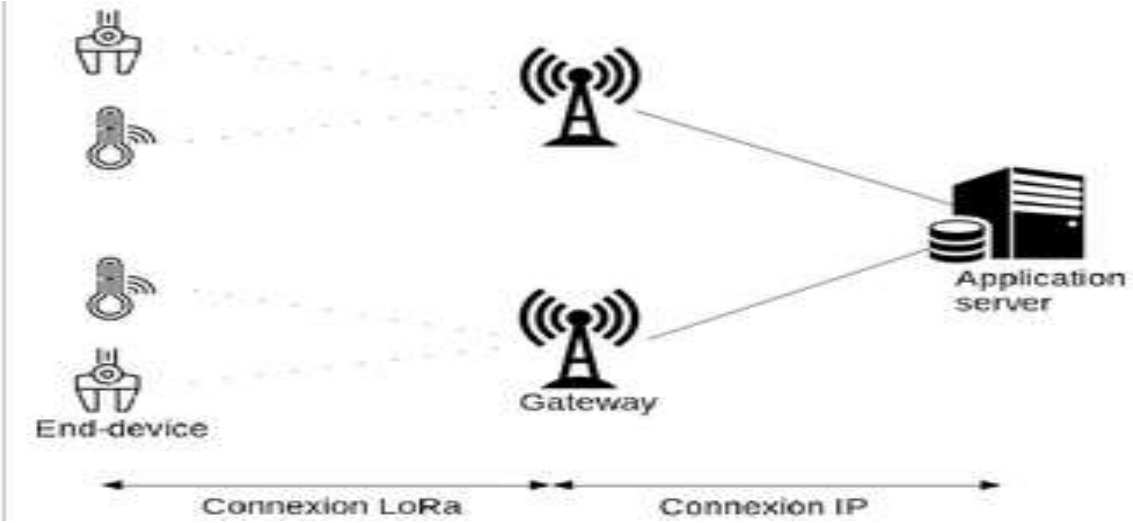
La topologie réseau LoRaWAN est dite en étoile d'étoiles (star-of-stars) car un serveur applicatif est connecté à une multitude de passerelles elles-mêmes connectées à une multitude d'équipements terminaux.

Au sens réseau, les équipements ne sont pas connectés aux passerelles, elles leur servent uniquement de relais pour joindre le serveur gérant leur application. Les paquets envoyés par les équipements sont retransmis par les passerelles après y avoir uniquement ajouté des informations concernant la qualité du signal reçu.

Si la couverture radio le permet, plusieurs passerelles peuvent retransmettre le même message d'un équipement, il est alors dupliqué dans le réseau de collecte, c'est le serveur hébergeant l'application qui assure le dédoublement des paquets

. Cette particularité permet notamment la localisation des équipements via la comparaison des différents temps d'arrivée pour un même paquet dupliqué

*Figure **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**.5 connexion de lora*



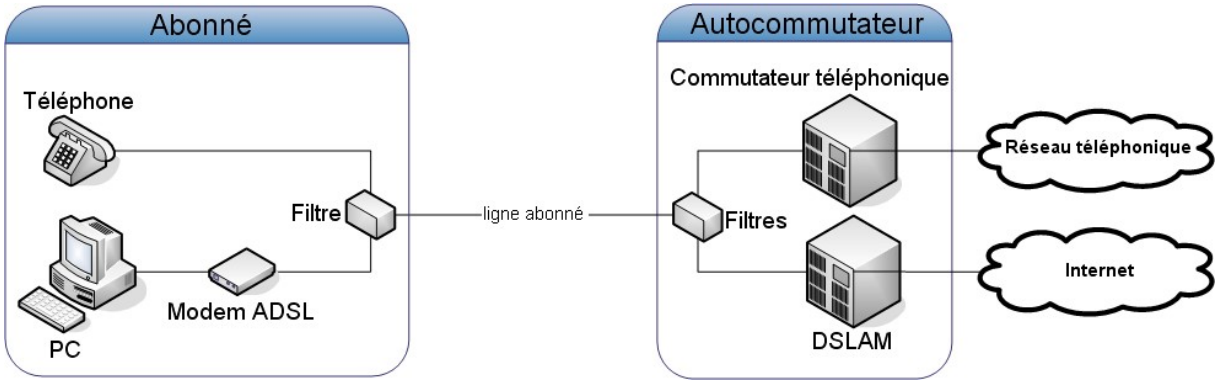
2.4.2 ADSL :

L'ADSL (Asymmetric Digital Subscriber Line) est une technique de communication numérique (couche physique) de la famille xDSL. Elle permet d'utiliser une ligne téléphonique, une ligne spécialisée, pour transmettre et recevoir des données numériques de manière indépendante du service téléphonique conventionnel. À ce titre, cette méthode de communication diffère de celle utilisée lors de l'exploitation de modems dits « analogiques », dont les signaux sont échangés dans le cadre d'une communication téléphonique .La technologie ADSL est massivement mise en œuvre par les fournisseurs d'accès à Internet pour le support des accès dits « haut-débit ».( )

2.4.2.1 Principe de fonctionnement :

La ligne téléphonique qui relie le domicile d'un abonné à l'autocommutateur public qui dessert son quartier (le « central téléphonique ») est constituée d'une paire de fils de cuivre, en général continue entre ces deux points (la boucle locale). Les signaux utilisés pour la téléphonie classique (sonnerie, numérotation multifréquences, voix) occupent une bande de fréquences qui s'étend entre 25 et 3 400 Hzenviron. Le principe de l'ADSL consiste à exploiter une autre bande de fréquence, située au-dessus de celle utilisée pour la téléphonie, pour échanger des données numériques en parallèle avec une éventuelle conversation téléphonique. Grâce à

cette séparation dans le domaine fréquentiel, les signaux ADSL qui transportent les données et les signaux téléphoniques qui transportent la voix circulent donc simultanément sur la même ligne d'abonné sans interférer les uns avec les autres.



2 .4.3 3G :

La 3 G est une technologie de connexion sans fils destinée au préalable à remplacer le système de communication des GSM et autres téléphones mobiles 2G. En plus de communication téléphonique classique, il permet de transférer des appels vidéos et des données numériques (connexion Internet). Basé sur la norme UMTS (pour Universal Mobile Télecommunications system), il permet un débit jusque 2 Mb/s.

2.5 Les géants du web :

Les grandes firmes du web visent non seulement à vendre des objets connectés mais aussi des services liées à ceux-ci, attirées par les prévisions de revenus qui pourraient excéder les 300 milliards de dollars grâce aux services en 2020 (15)

Ces services dérivent directement de l'énorme quantité de données produites par les objets connectés qui demandent espace de stockage, vitesse de traitement et souvent bande passante pour le streaming de données audio ou vidéo. Pour certains, la solution idéale à ces problèmes est le Cloud computing (16)

2 .5 .1 INTEL :

En avril 2016, Intel, qui est historiquement un fabricant de microprocesseurs pour ordinateurs, annonce le licenciement de 12 000 employés (11 % de ses effectifs) afin de se focaliser sur l'IdO, les FPGA et les processeurs pour centres de données qui ensemble ont généré 40 % de ses revenus pour l'année 2015(17)

Les rapports aux investisseurs de l'entreprise montrent l'importance économique croissante que prend l'IOT : alors qu'en 2013 la dénomination « Internet of Things » n'apparaissait même pas dans ses documents(18), en 2014, la division IOT annonce un revenu de 2,1 milliards de dollars, en augmentation de 19 % par rapport à 2013(19)

En 2015, la tendance continue avec une augmentation de 7 % par rapport au 2014 avec 2,3 milliards de dollars(20)

2.5.2 Samsung :

Le géant coréen Samsung Electronics entre à tous les niveaux dans le marché de l'IOT en 2014 avec l'acquisition de SmartThings, une startup américaine développant des objets connectés pour la maison. Le prix d'achat n'est pas public, mais Samsung annonce à ses

investisseurs que cette acquisition a apportés une augmentation du revenu de 2 469 millions de wons (21)

2.4.3 IBM :

En 2015, IBM annonce l'investissement de 3 milliards de dollars pour établir une nouvelle unité IOT à Munich. Le but est de proposer :

- une plateforme pour analyser les nécessités des entreprises en matière d'IdO ;
- une plateforme pour aider les programmeurs à exploiter les informations produites par l'IdO ;
- créer un écosystème avec les fabricants de matériels et les fournisseurs de services(22)

En juillet 2016, Informix a reçu de Cisco le prix de la « *Best IOT Database* »(23)

2.6 Conclusion

Dans ce chapitre de l'état de l'art nous nous sommes intéressées aux principales approches utiliser dans le domaine de l'internet of things(IOT), nous avons commencé par définir le Protocol de données MQTT et son fonctionnement, puis nous avons découverts les différents plates-formes utilisant le Protocol MQTT et leurs différents services afin de choisir la meilleurs plate-forme qui soit la plus adapté et la plus flexible a notre projet

Pour finir nous avons exposé les différents moyens de communication qui permettra de communiquer avec les différents objets

Chapitre3 : environnement de travail

3 .1 introduction :

L'internet of things (IOT) et la technologie web sont les outils majeur de la nouvelle ère de transformation, à travers IOT on recherche le potentiel des nouvelle technologie et leur impact dans l'amélioration du monde réel, en mêmes temps, pour la gestion des robots on trouve plusieurs alternatives proposer comme: Raspberry, ESP , Des Carte dédié .mais le choix diffère de la carte diffère d'un robot a l'autre, par exemple si en prend un robot de commande par vision il faut choisir la carte raspberry PI, car il a besoin d'une carte robuste (précision , un bon asservissement ,etc.), mais lorsqu'on cherche un commander un petit robot il est préférable d'utiliser une carte basique , un microcontrôleur car il est plein de ressource.

3.2 Hardware utilisé :

Dans notre projet en a une partie pour la commande des moteurs et une partie pour collecter l'image et sa diffusion on a divisé la partie hardware en 2 parties :

3.2.1 NodeMCU :

NodeMCU est une plate-forme open source IOT. Il contient le firmware qui fonctionne sur le ESP8266 Wi-Fi SoC de Espressif Systèmes et le matériel () qui est basé sur le module ESP-12. Le terme "NodeMCU" se réfère par défaut au firmware plutôt qu'aux kits de développement. Le firmware utilise le Lua qui est un langage de script. Il est basé sur le projet eLua[7] et construit sur l'Espressif Non-OS SDK pour ESP8266. Il utilise de nombreux projets open source comme lua-cjson et spiffs. ()

NodeMCU a été créé peu de temps après l'apparition commerciale de l'ESP8266. Le 30 décembre 2013, la société « Espressif Systems » a commencé la production du modèle : ESP8266(38). L'ESP8266 est un SoC Wi-Fi gratuit intégré à un Tensilica Xtensa LX106 de base, largement utilisé dans les applications IoT (voir les projets associés ci-dessous). NodeMCU a commencé le 13 octobre 2014, lorsque Hong a publié le premier fichier de nodemcu-firmware sur GitHub(38). Deux mois plus tard, le projet a été étendu pour inclure une plate-forme ouverte (open-hardware) lorsque le développeur *Huang R* a publié le fichier au format gerbé à base du composant ESP8266, nommé devkit v0.9(39). Dans le mois

courant, *Tuan PM* a mis à disposition de téléchargement la bibliothèque client MQTT de Contiki vers la plate-forme SoC ESP8266(40) et l'a inséré au projet NodeMCU. Dès lors NodeMCU a été en mesure de supporter le protocol MQTT IOT, à l'aide de Lua pour accéder au broker MQTT. Une autre mise à jour importante a été faite le 30 janvier 2015, Durant l'été 2015, les créateurs ont abandonné ce projet de firmware et un groupe indépendant de contributeurs a pris le relais. À l'été 2016, le NodeMCU incluait plus de 40 modules différents. En raison de contraintes de ressources, les utilisateurs doivent sélectionner les modules pertinents pour leur projet et construire un firmware adapté à leurs besoins.

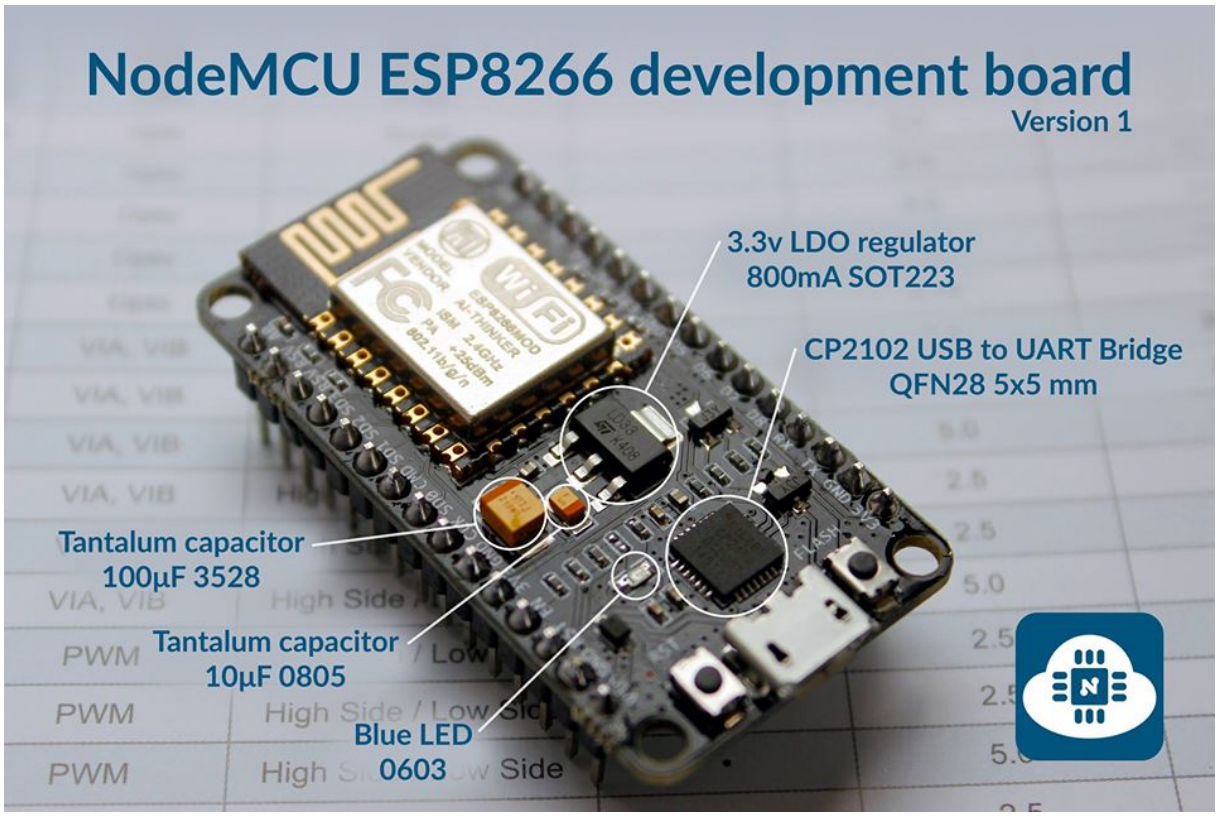


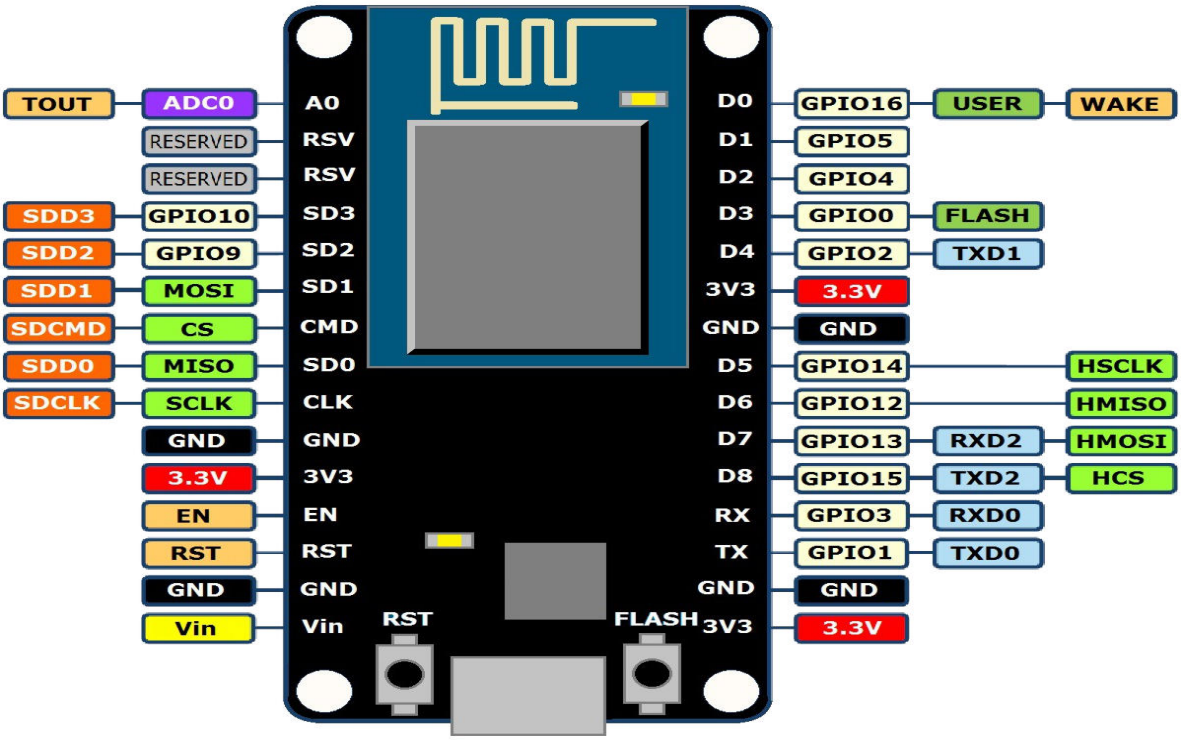
Figure ***Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.****.6*

3.2.1.1 spécifications :

On trouve sur cette Mini carte Le module Esp-wroom-02 Qui est un module Wifi mcu 32bits de faible puissance, basé sur la puce esp8266, les Piles réseaux TCP/IP, L'adc 10 bits et les interfaces HSPI/UART/PWM/I2C/I2S sont toute intègres dans ce module.

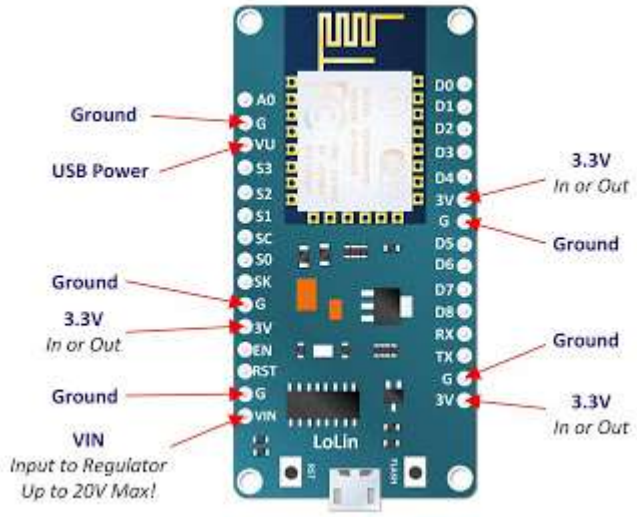
- **Adressage :** 32 bits

- **Fréquence d'horloge** : 80 MHZ
- **Mémoire flash :** 512ko
- **Mémoire RAM :** 96ko
- **Nombre E/S :** 11 (D0…….D8 + RX/TX) est toute les sortie sont PWM
- **Nombre d'entrée analogique :** 1 (A0) 10bits, 1v Max
- **UART port série :** 1
- **I2C :** 1
- **I2S :**1



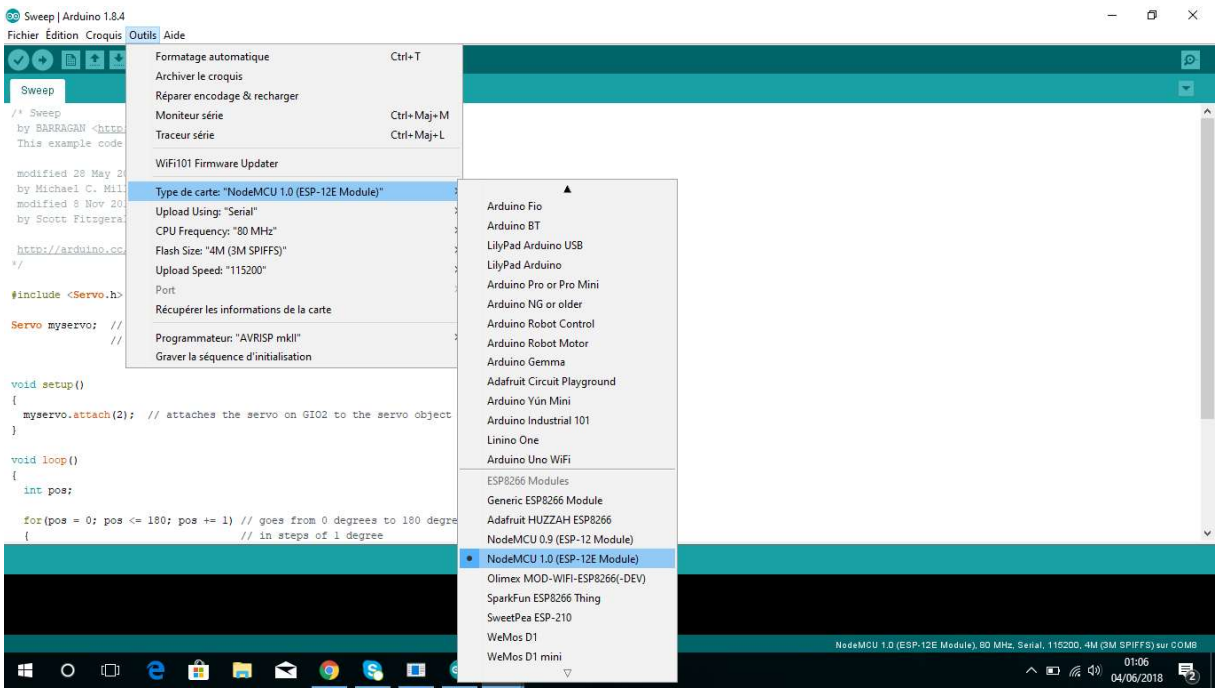
3.2.1.1 Alimentation :

Le schéma ci-dessous, montre les pins sur lesquels on peut raccorder une alimentation électrique ainsi que la tension maximum pour chacun.



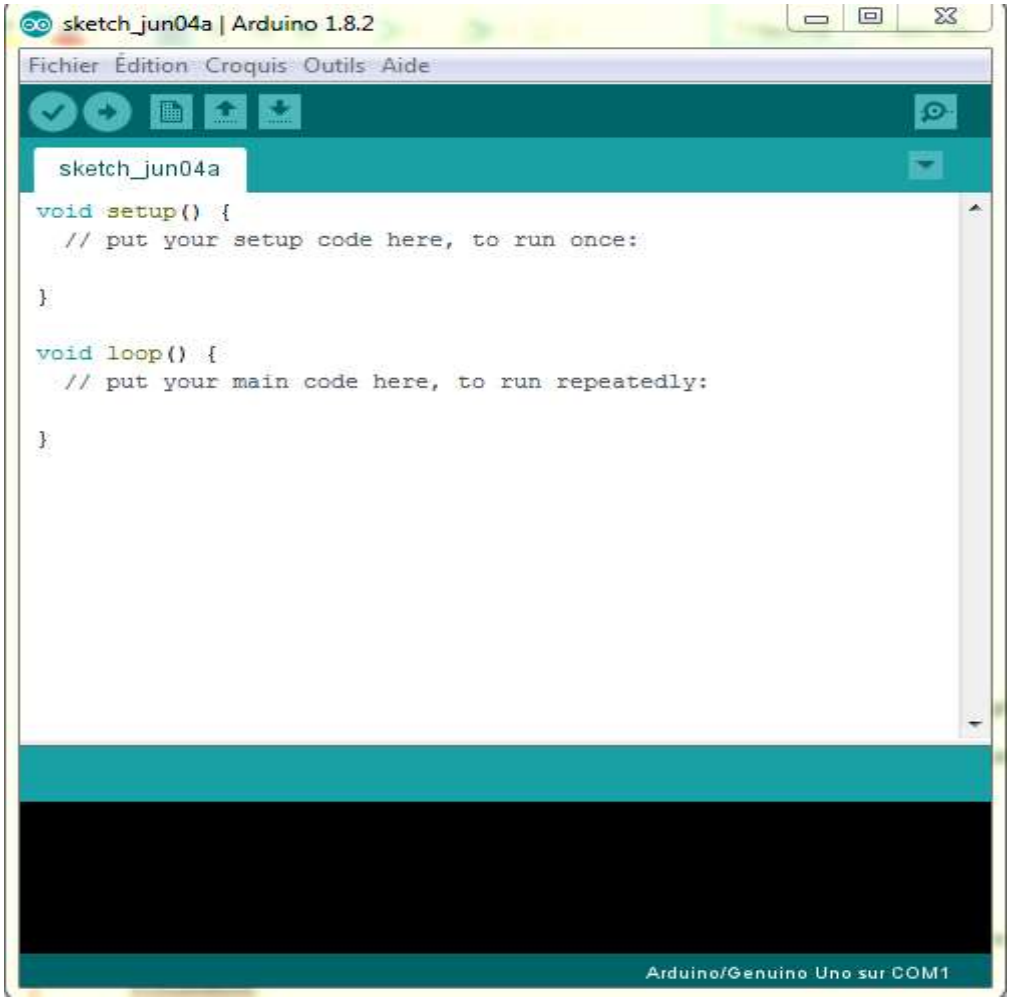
3.2.1.4 Utilisation de la carte Sur Arduino :

La programmation et l'implémentation de la carte NodeMcu sur La plateforme Arduino reste basée sur le même principe car le microcontrôleur utilisés dans cette carte est programmée par Loader .et Pour sélectionné cette carte voir la figure ci-dessous



Ce Logiciel est téléchargeable sur le site officielle Arduino.cc, vous trouvé la version installable.

L’interface sert comme éditeur de programme Voir la figure



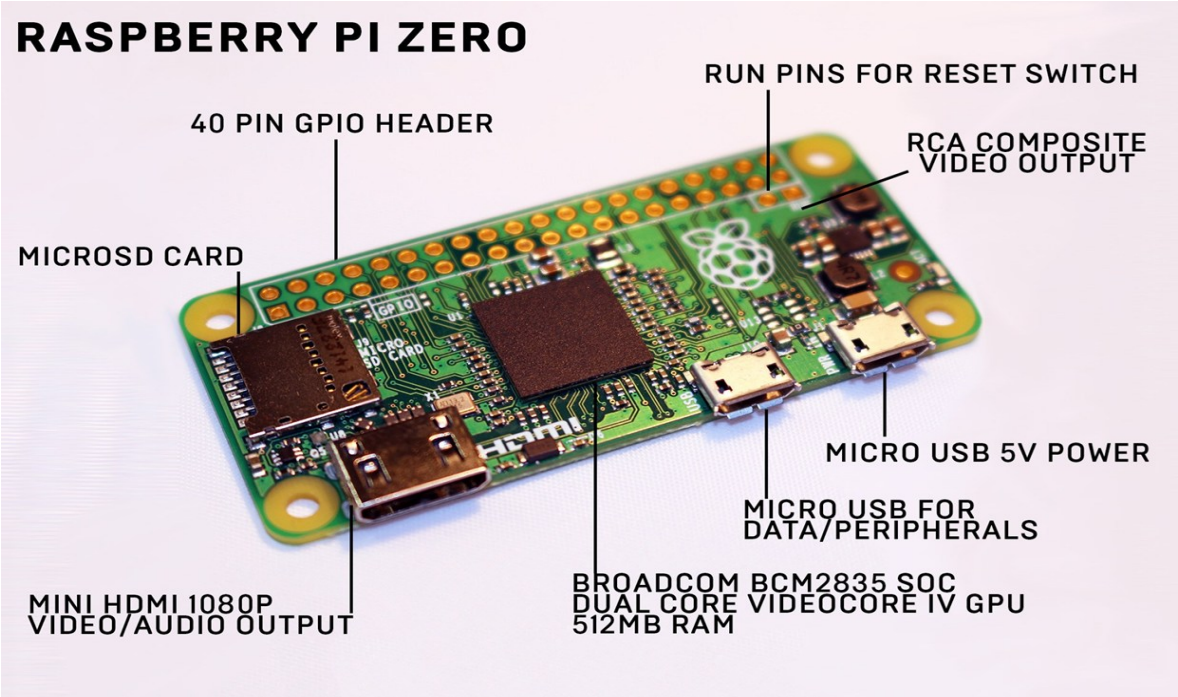
Le langage de programmation utilisé es le C, C++, compilé avec avr-g++ () pour générer le fichier binaire, est-il est liées à la bibliothèque de développement Arduino, permettant d’utiliser la carte est ses entrée sortie

Apres avoir écrit le programme reste qu’a le compilé voir figure

3.2.2RASPBERRY :

Le RASPBERRY PI est un nano-ordinateur mono carte à processeur ARM conçu par le créateur David Braben, dans le cadre de sa fondation RASPBERRY PI().

Cette mini carte est destiner a félicité l’apprentissage de la programmation informatique il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux-Debian et des logiciels compatibles



Il y a plusieurs modèle de RASPBERRYmais dans notre projet en a pas besoin d'un RASPBERRYpuissant, on a besoin juste pour envoyer l'image, donc le meilleur choix a été le RASPBERRY Pi ZERO avec un Processeur ARM11 Broadcom BCM2835, Single-core 1GHz avec 2 micro-port USB un pour l'alimentation et l'autre pour USB et 1 Port Mini HDMI est 1 port CSI est un lecteur de carte micro SD

3.3 Software utilisé :

3 .3.1 software sur le RASPBERRY :

3.3.1.1 le noyau linux :

Le noyau Linux est un noyau de système d'exploitation de type UNIX. Il est utilisé dans plusieurs systèmes d'exploitation dont notamment GNU/Linux (couramment appelé « Linux ») et Android. Le noyau Linux est un logiciel libre développé essentiellement en langage C

Le noyau est le cœur du système, c'est lui qui s'occupe de fournir aux logiciels une interface de programmation pour utiliser le matériel. Le noyau Linux a été créé en 1991 par Linus Torvalds pour les compatibles PC. Initialement conçu pour l'architecture de processeur x86, il a ensuite été porté sur de nombreuses autres, dont m68k, PowerPC, ARM, SPARC et MIPS. Il s'utilise dans une très large gamme de matériel, des systèmes embarqués aux superordinateurs, en passant par les téléphones mobiles et ordinateurs personnels

Ses caractéristiques principales sont d'être multitâche et multi-utilisateur. Il respecte les normes POSIXce qui en fait un digne héritier des systèmes UNIX

Comme tous les programmes informatiques, le noyau Linux est écrit sous forme de code source, et doit être transformé en binaire exécutable pour être compris par le microprocesseur.

Dans la mesure où le code source du noyau Linux contient une très grande quantité de fonctionnalités, l'utilisateur peut choisir de n'intégrer que celles qui lui sont utiles ou les mieux adaptées (de nombreuses fonctionnalités sont concurrentes) : c'est l'étape de configuration du noyau.

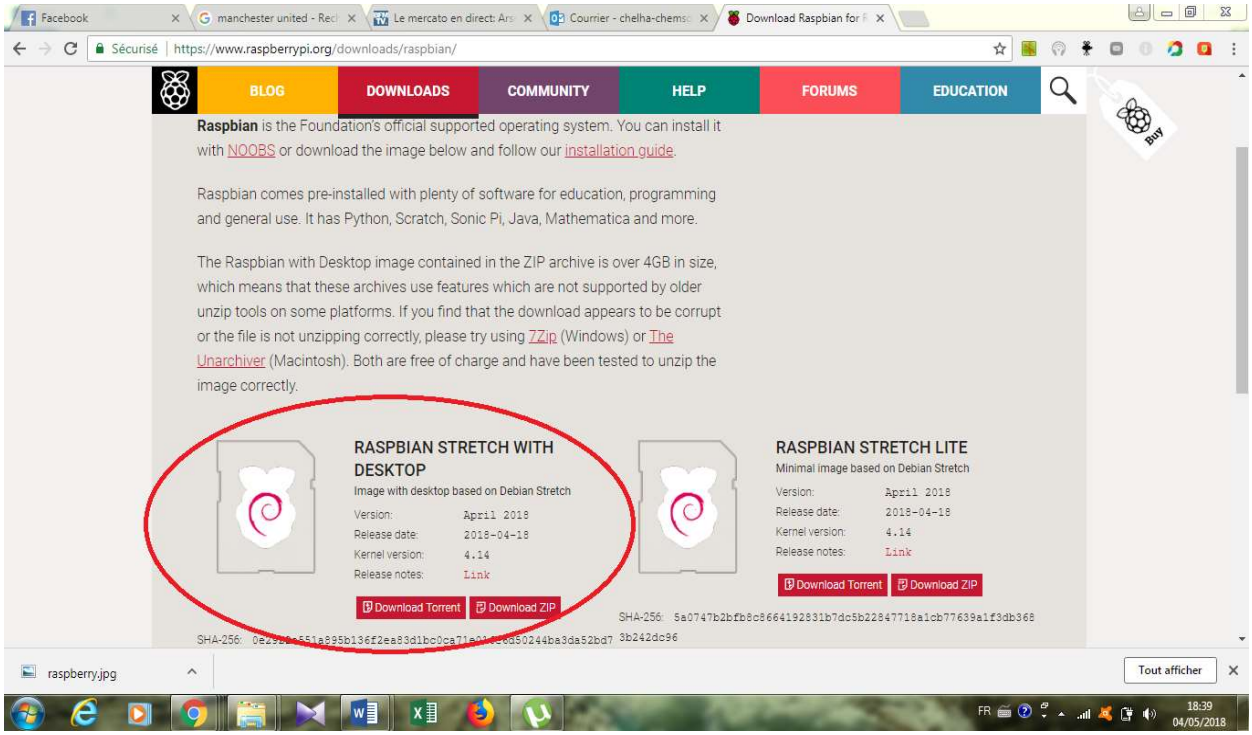
La grande majorité des distributions GNU/Linux installent un noyau compilé préalablement qui répond aux besoins des postes de travail et serveurs. Il est donc rare qu'un utilisateur de Linux ait à compiler un noyau. La compilation permet d'adapter le noyau à des besoins spécifiques comme le support de matériels peu répandus, l'activation de fonctionnalités expérimentales ou l'adaptation à des plateformes particulières comme des systèmes embarqués.

3.3.1.2 implémentations sur la carte :

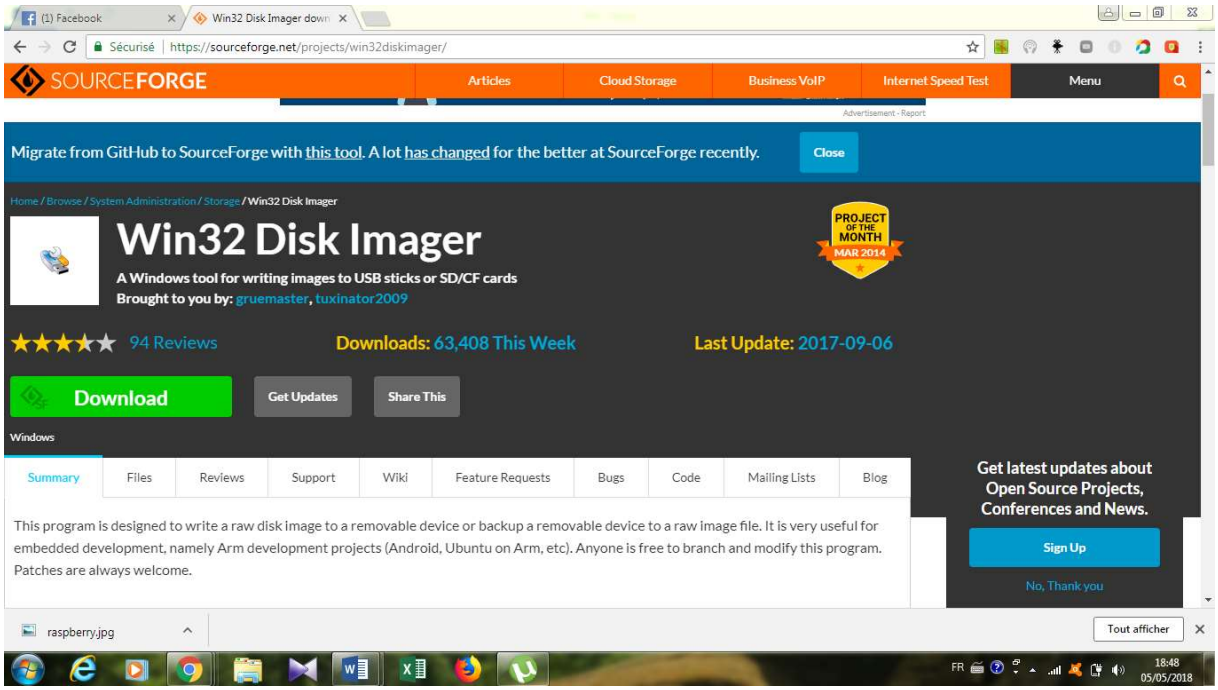
Il existe quatre-vingt-dix distributions qui tournent autour du noyau Linux, mais trois grande entre elle sont les plus populaires et les plus essentielles on trouve (Debian, Slackware et Redhate). Le nombre des héritiers de ses trois grandes familles est incomptable

Mais le problème qui se pose c'est que pour booter un systèmeembarqué avec l'une des distributions il faut chercher l'image dédié au processeur si elle existe

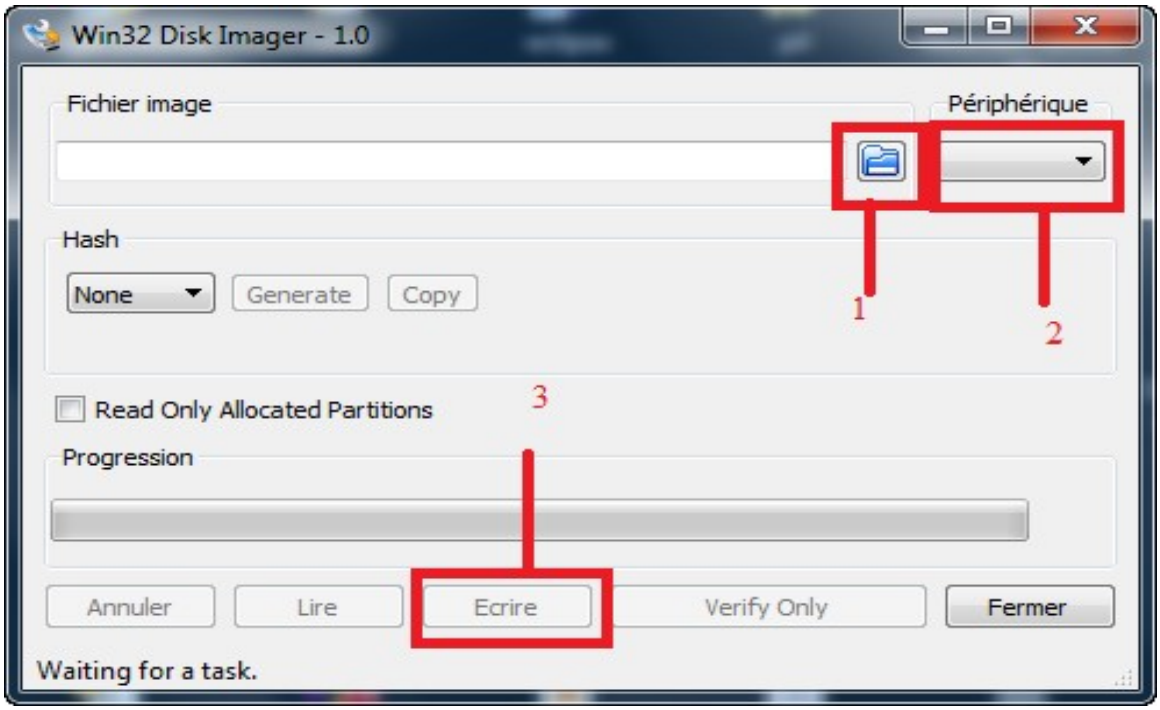
Nous avons choisi pour ce projet la distribution RASBIAN qui est disponible sur le site officiel https://www.raspberrypi.org/downloads/raspbiancomme le montre la figure



Ont trouvedeux version, vous télécharger la version qui encercle en rouge, après avoir fini et en sachant que le système d'exploitation d'un Raspberry doit être booter sur une carte mémoire, il faut télécharger le logiciel qui s'occupera de le télé-verser en mode bootable, pour cela il faut telecharger Win32DiskImager disponible sur le site https://sourceforge.net/projects/win32diskimager/



Après avoir installé le logicielle, il faut placer la carte mémoire du type micro SD SDHC ou SDXC dans un lecteur de carte mémoire a fin de la brancher sur un ordinateur, après lancer le programme Win32DiskImager comme illustrer dans la figure 3.12



Le bouton numéro un, permet de choisir l'emplacement de votre système d'exploitation que vous voulez la télé-verser, lecombo box numéro deux pour choisir quel disque-dur vous aller latélé-verser (carte SD), il resteà écrire par le bouton trois.

Après avoir branché la carte SD sur le RASPBERRY vous le démarré, Et ont aura directement l'interface comme la montre la figue figure du raspbeery

3.3.1.3 installations open cv :

L'installation d'open cv dépend de deux facteurs essentiels :

- le système d'exploitation sur lequel on travail.
- le langage de développement qu'on souhaite utiliser.

  Etape1 : on commence par faire une mise à jour du raspberry upgrade et update

  Sudo APT-get update

  Sudo APT-get upgrade

  Etape 2 : on supprime les bibliothèques qui pourraient entrer en conflit avec les nouveaux paquets ceux-ci peuvent ou non déjà exister sur le RASPBERRY :

  sudo apt-get remove libavcodec-extra-56 libavformat56 libavresample2 libavutil54

Etape 3 : télécharger et installer les packages en entrant la commande suivante dans le terminal :

wget https://github.com/ccrisan/motioneye/wiki/precompiled/ffmpeg_3.1.1-1_armhf.deb
sudo dpkg -i ffmpeg_3.1.1-1_armhf.deb

Etape 4 : installer les packages,on a besoin d'eux car le logiciel motion s'appuie sur eux

sudo apt-get install curl libssl-dev libcurl4-openssl-dev libjpeg-dev libx264-142 libavcodec56 libavformat56 libmysqlclient18 libswscale3 libpq5

Etape5 : avec ces paquets installés, nous pouvons maintenant récupérer la dernière version du logiciel de mouvement et l'installer, pour le faire on exécute les commandes suivantes :

wgethttps://github.com/MotionProject/motion/releases/download/release4.0.1/pi_jessie_motion_4.0.1-1_armhf.deb
sudo dpkg -i pi_jessie_motion_4.0.1-1_armhf.deb

Etape6 : Installer les librairiesd'on a besoin :

sudo apt-get install libmariadbclient18 libpq5 libavcodec57 libavformat57 libavutil55 libswscale4

3.3.1.4 Configuration motion :

Etape 1 : télécharger motion est l'installer

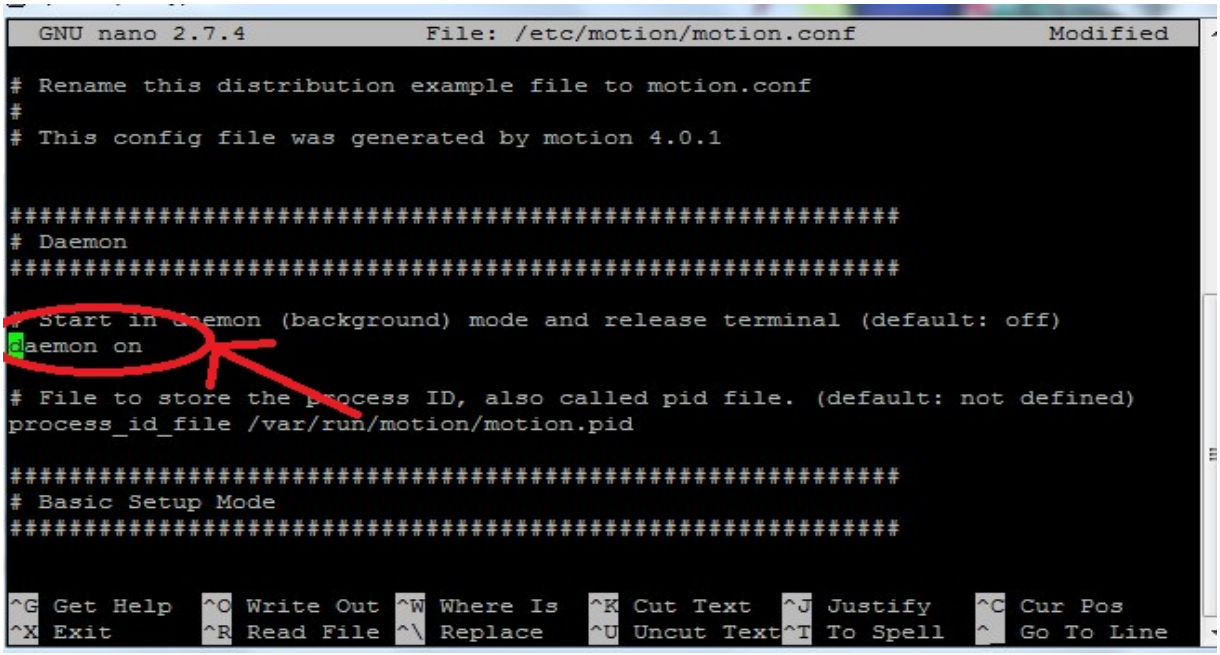
sudo wget https://github.com/Motion-Project/motion/releases/download/release-4.0.1/pi_stretch_motion_4.0.1-1_armhf.deb

sudo dpkg -i pi_stretch_motion_4.0.1-1_armhf.deb

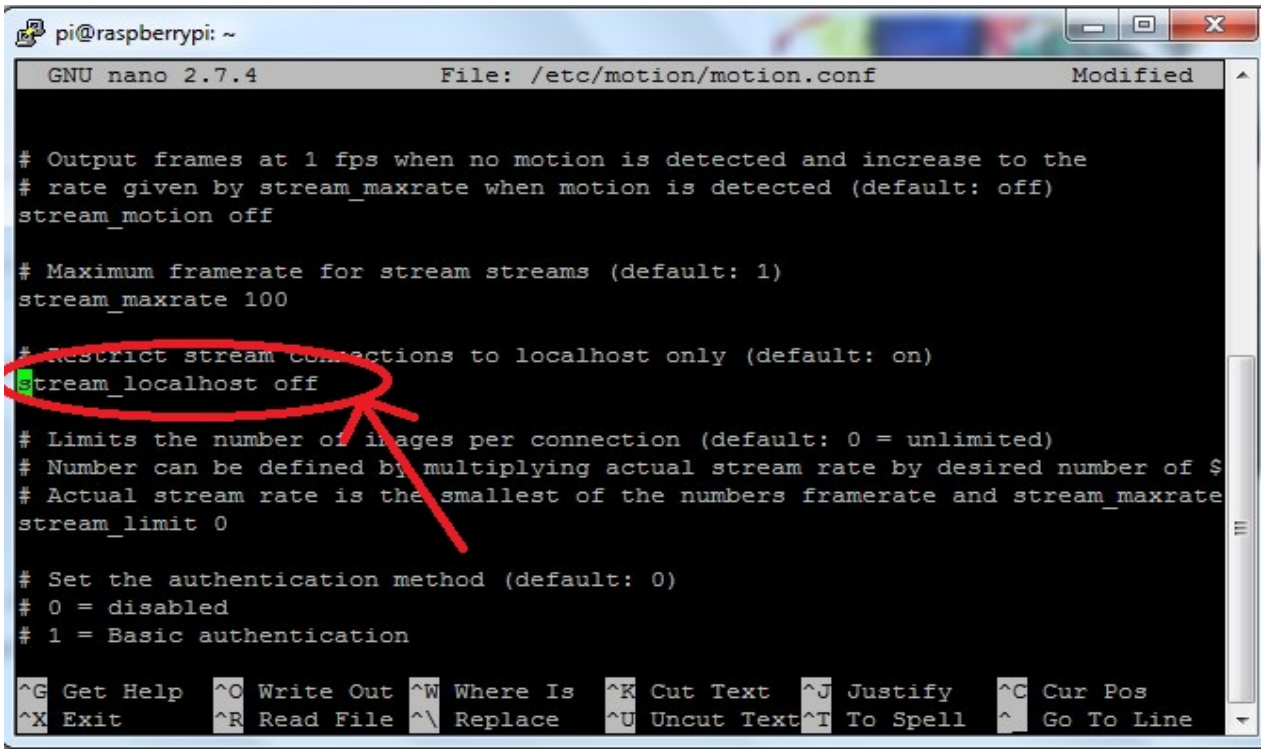
Etape 2 : on abesoin de faire quelque modification sur le au fichier de configuration (motion) pour accéder au fichier on fait la commande :

Sudo nano /etc/motion/motion.conf

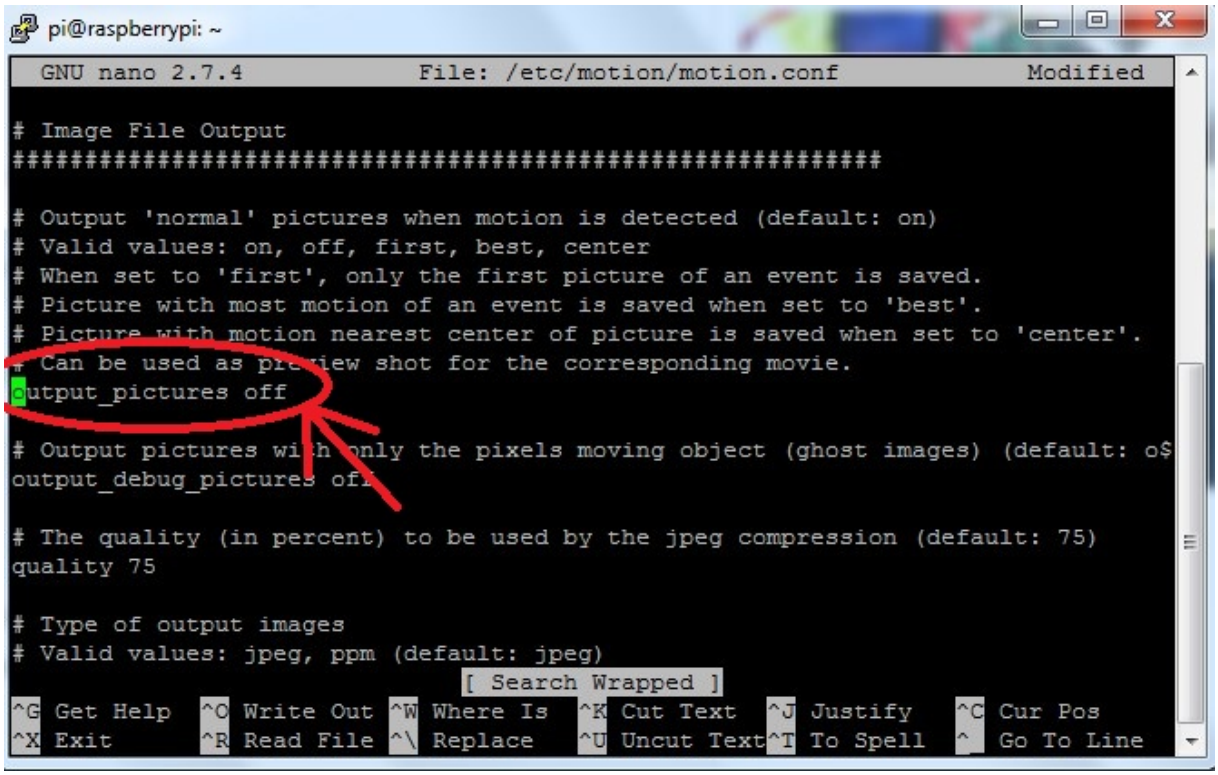
activer le damond comme il est montres dans la figure

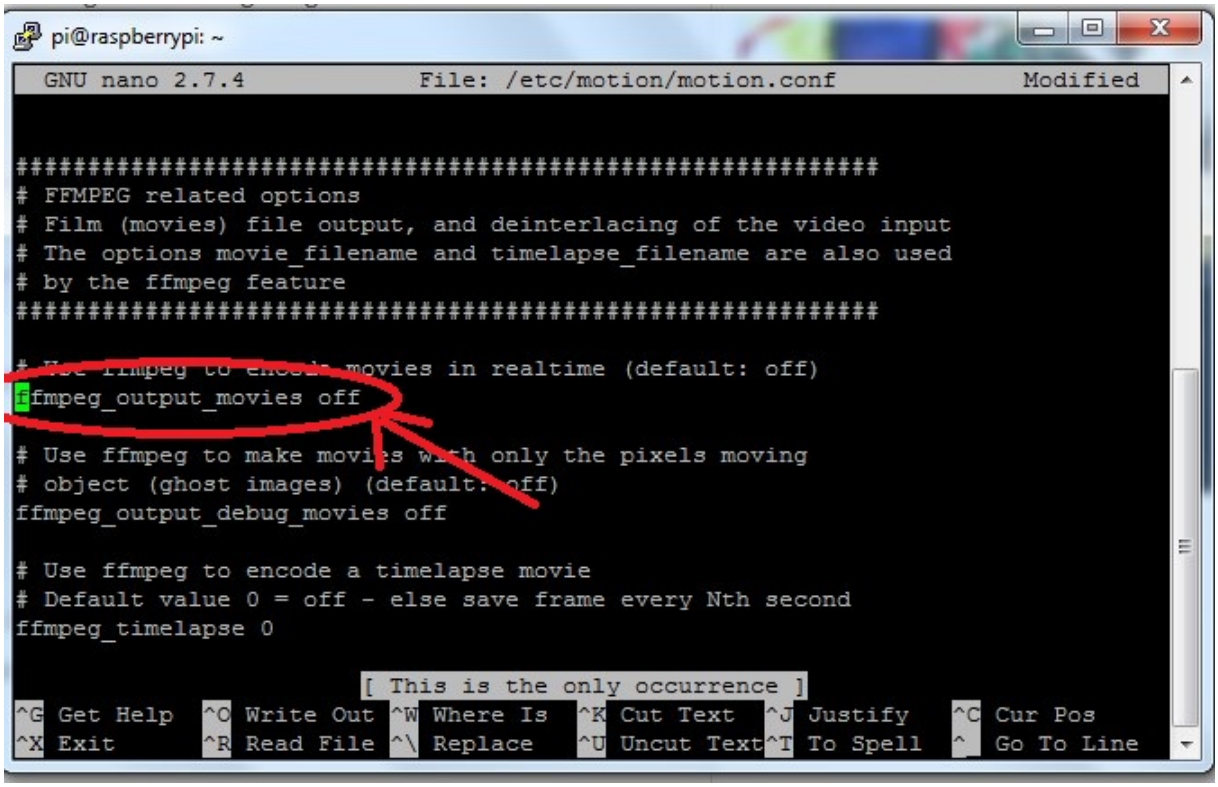


Apres désactiver le stream_localhost comme il est montres dans la figure

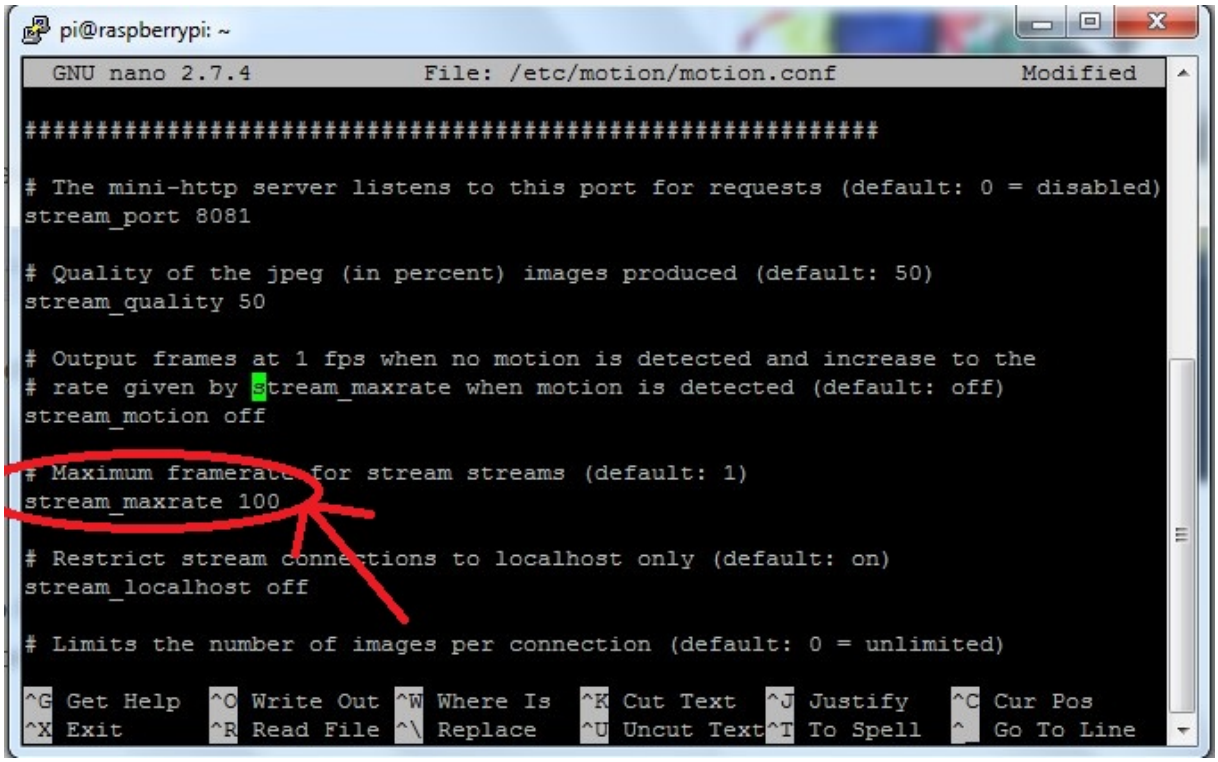


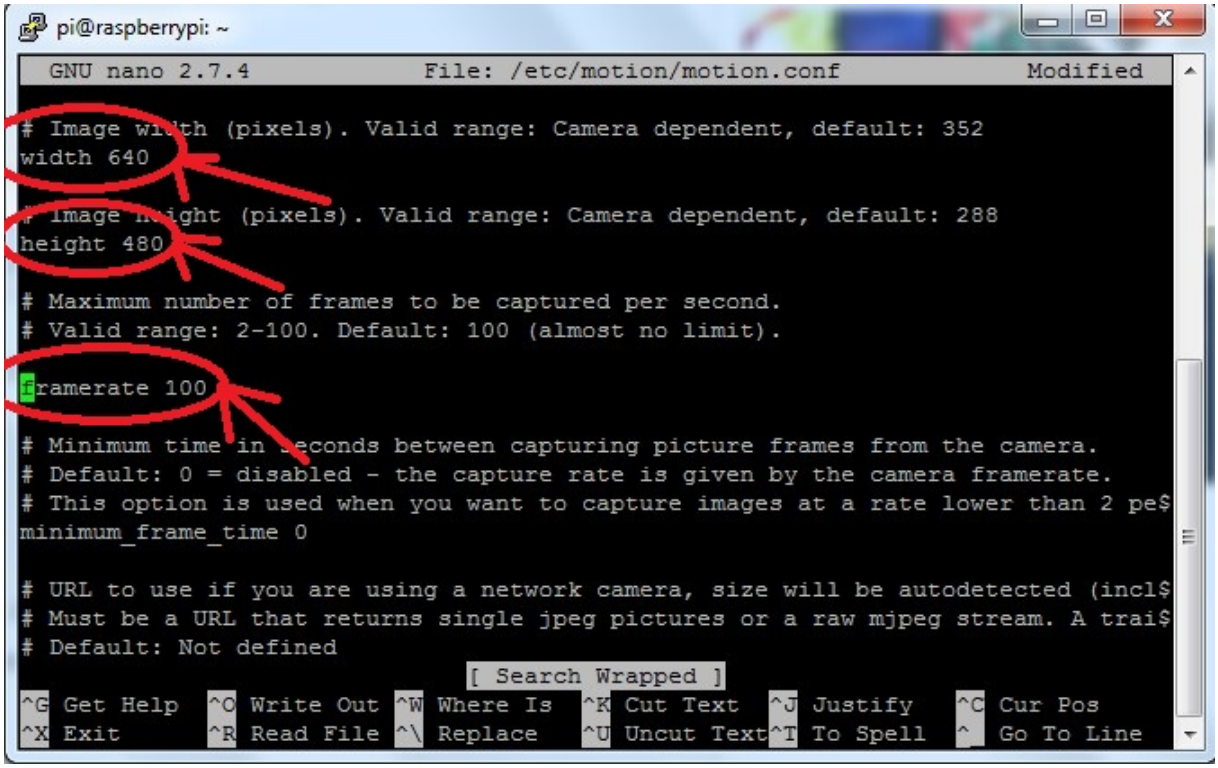
Apres désactiver output_pictures pour ne pas avoir un blocage lors de la retransmission d'image, comme il est montres dans les deux figure suivante





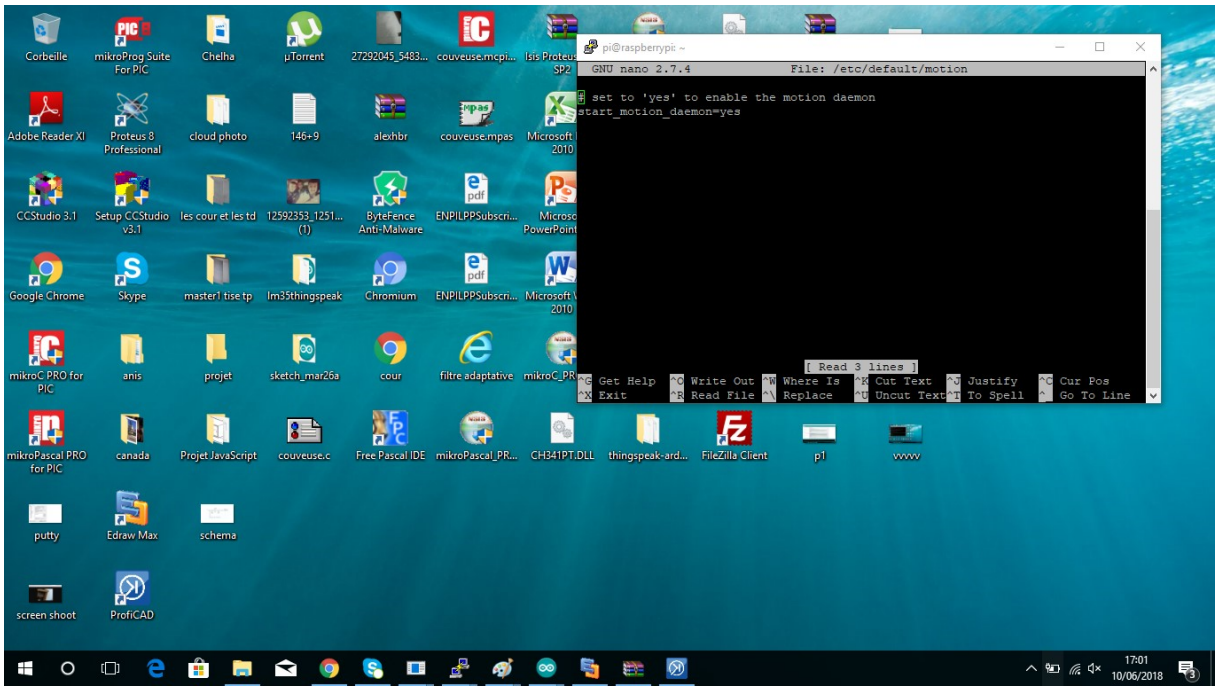
Apres ca configurer la bande passante et c'est ressource pour avoir un streaming en temps réel comme il est montre dans les figure suivante





Etape 3 : configurer le fichier daemond, nous devons éditer le fichier motion

Sudo nano /etc/default/motion



Pourfaire marcher la camera il faut entrer la commande suivante :

Sudo service motion start

3.3.1.5 activations SSH :

SSH (SecureShell) nous permetde commander le raspberry à travers un ordinateur, Avec le logiciel PUtty et pour cela il faut L'activer et suivre l'étape suivante :

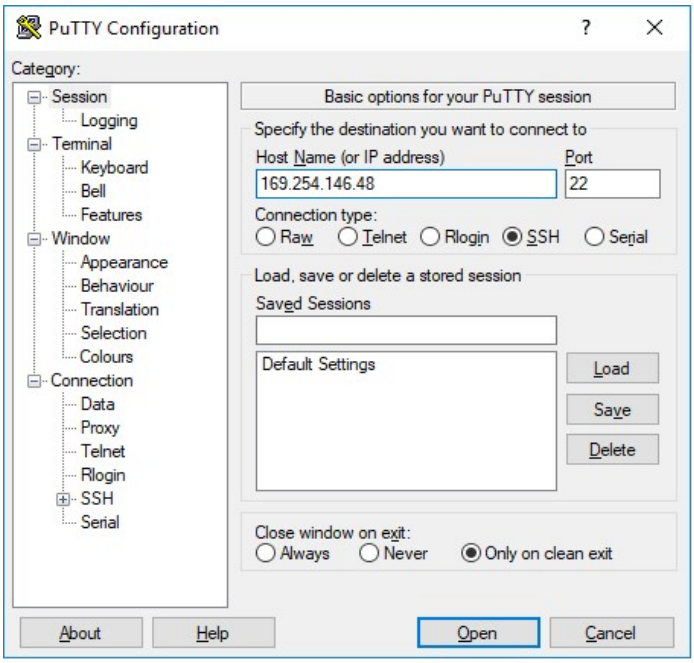
- entrer la commande sudo raspi-confi
- sélectionner Interfacing options
- aller à SSH
- choisir Yes pour l'activer
- Cliqué sur Finish

3.3.1.5.1 PuTTY :

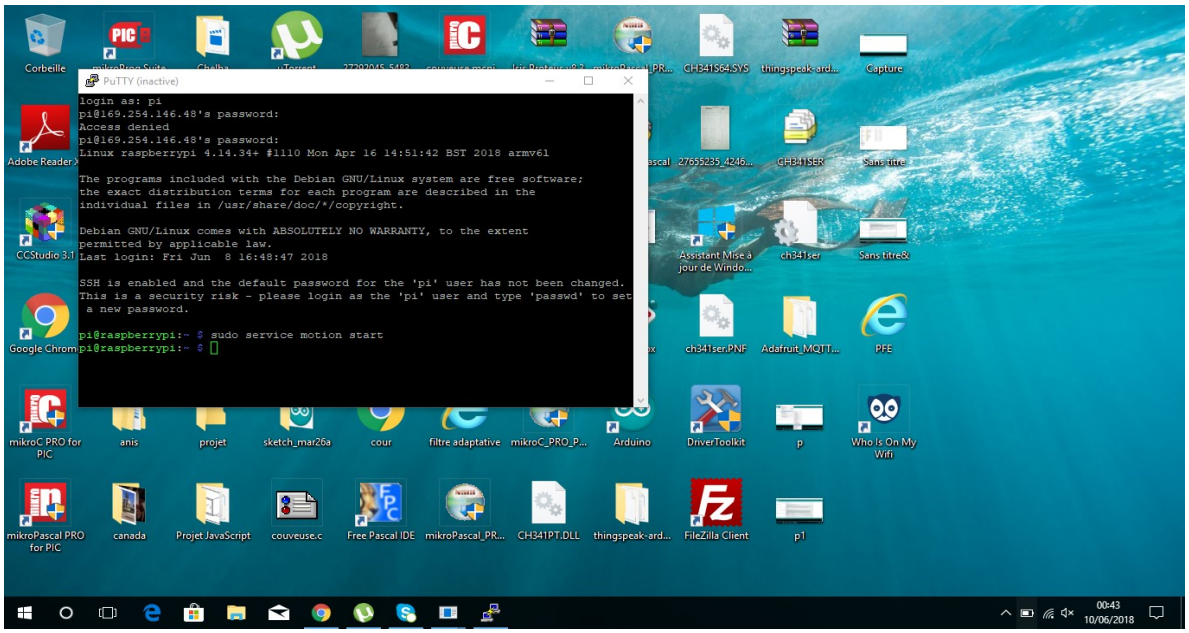
PuTTY est un programme de terminal polyvalent pour Windows. C'est le client SSH gratuit le plus populaire au monde. Il prend en charge les connexions SSH, telnet et raw socket avec une bonne émulation de terminal. Il prend en charge l'authentification par clé publique et l'authentification unique Kerberos. Il inclut également les implémentations SFTP et SCP en ligne de commande.

PuTTY nous permet d'accéder au terminal du RASPBERRY PI ZERO, et pour ça il faut accéder au terminal du RASPBERRY PI et faire entrer la commande **ifconfig**, on obtient une adresse IP du RASPBERRY PI concerné.

Puis faire entrer l'adresse IP obtenu du RASPBERRY PI dans l'interface du PuTTY comme il est montré dans la figure ci-dessous



Puis il nous est demander le login qui est PI, puis le mot de passe qui est RASPBERRY pour nous donner la main et commander le RASPERRY PI à partir d'un terminal d'un autre ordinateur et on obtient la commande comme la montre la figure ci-dessous



3.3.1.6 installation de la camera :

Sudo APT-get install cheese

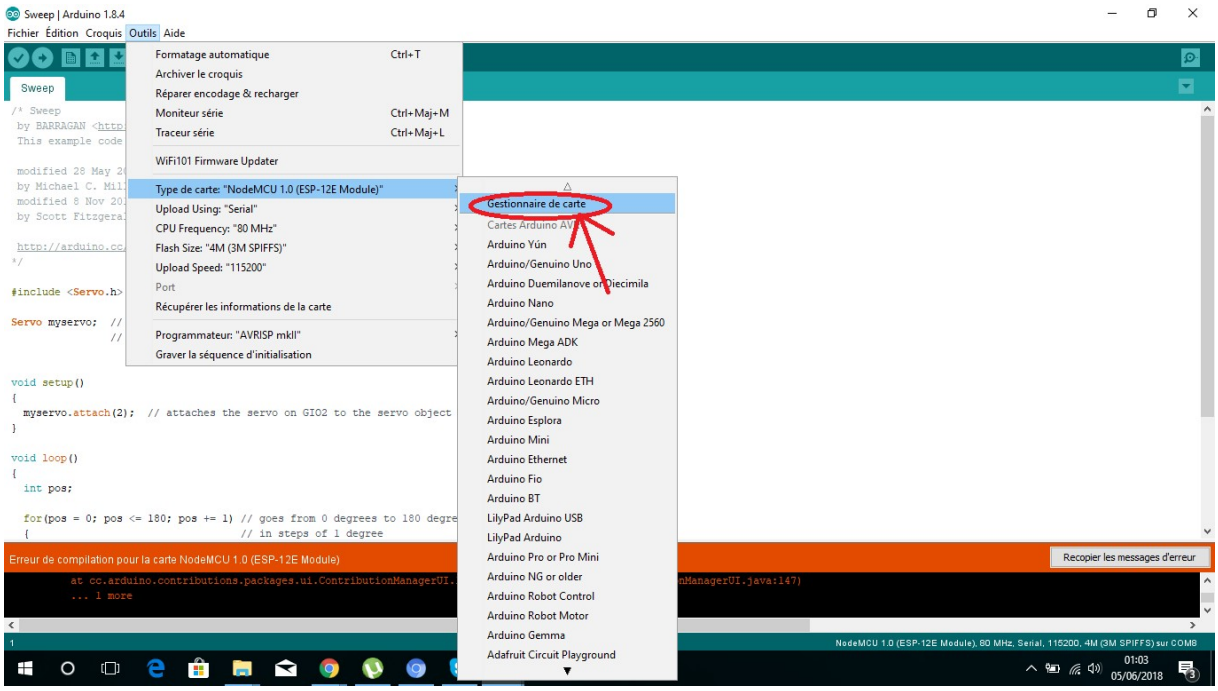
Et pour vérifier que la camera soit bien connecter il faut entrer le code

Ls getusb

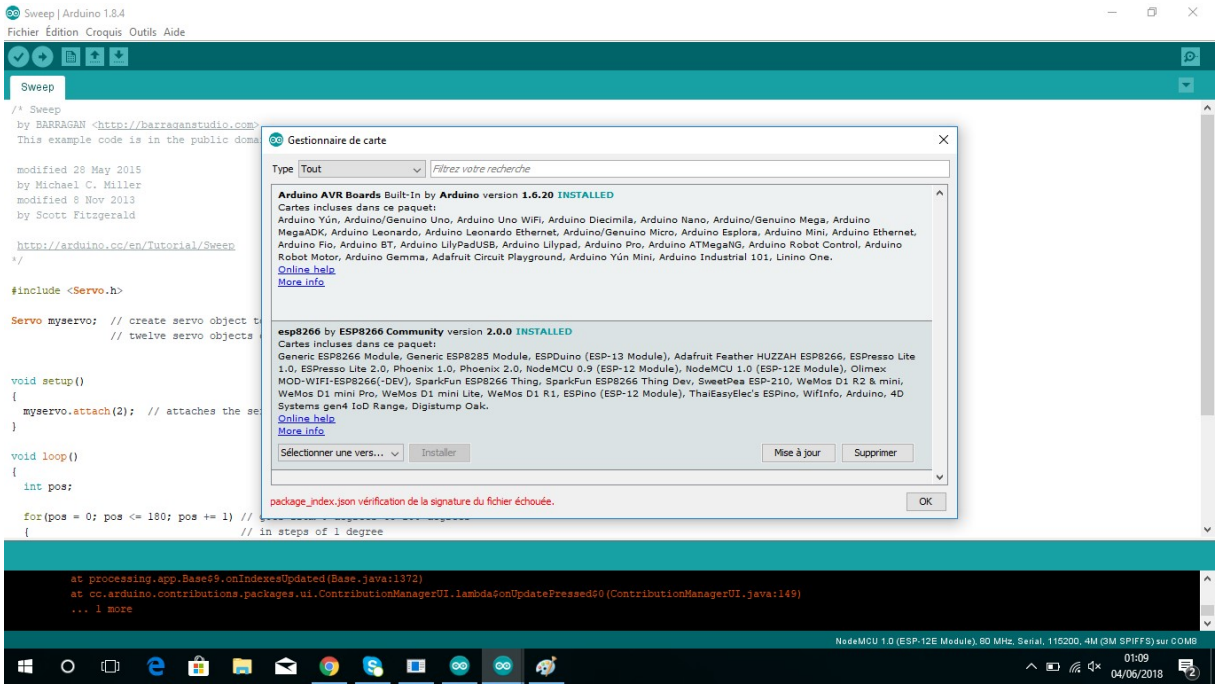
3.3.2 Software sur NodeMcu

3.3.2.1 Installation de L'esp8266 sur ARDUINO :

L'avantage de l'ESP étant tout intégré dans une puce elle a seulement besoin de l'ARDUINO pour êtreinstaller, tout d'abord il faut commencer par télécharger le packages sur http://arduino.esp8266.com/stable/package_esp8266com_index.json, après il faut l'installer sur la carte comme le montre les figures ci-dessous

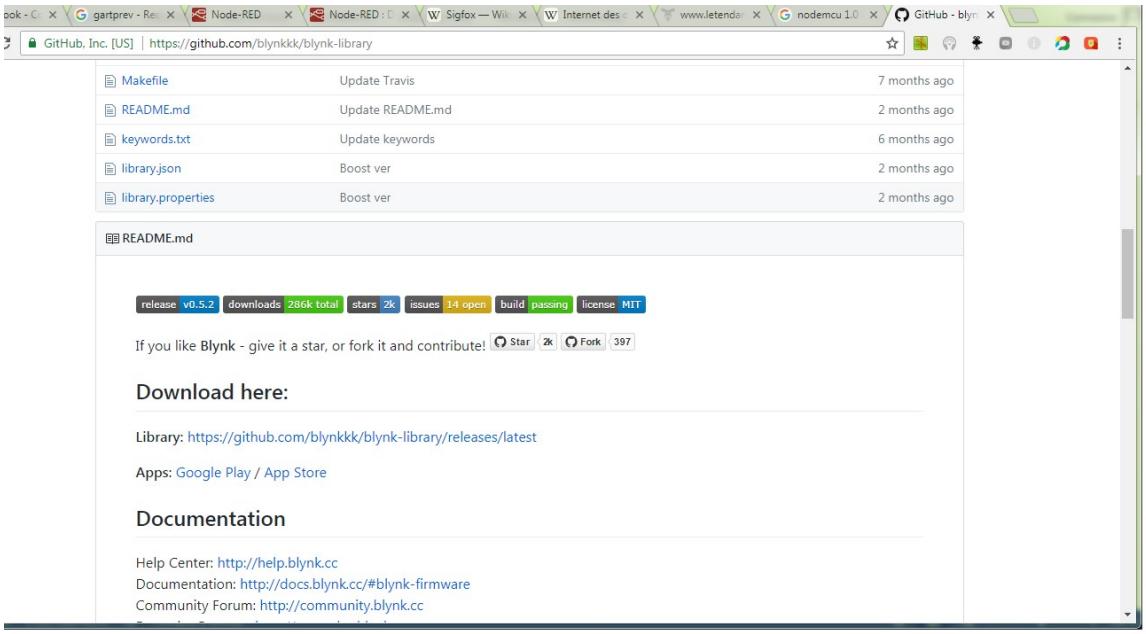


Et après faire la recherche sur L'esp8266 et l'installer comme il est sur la figure

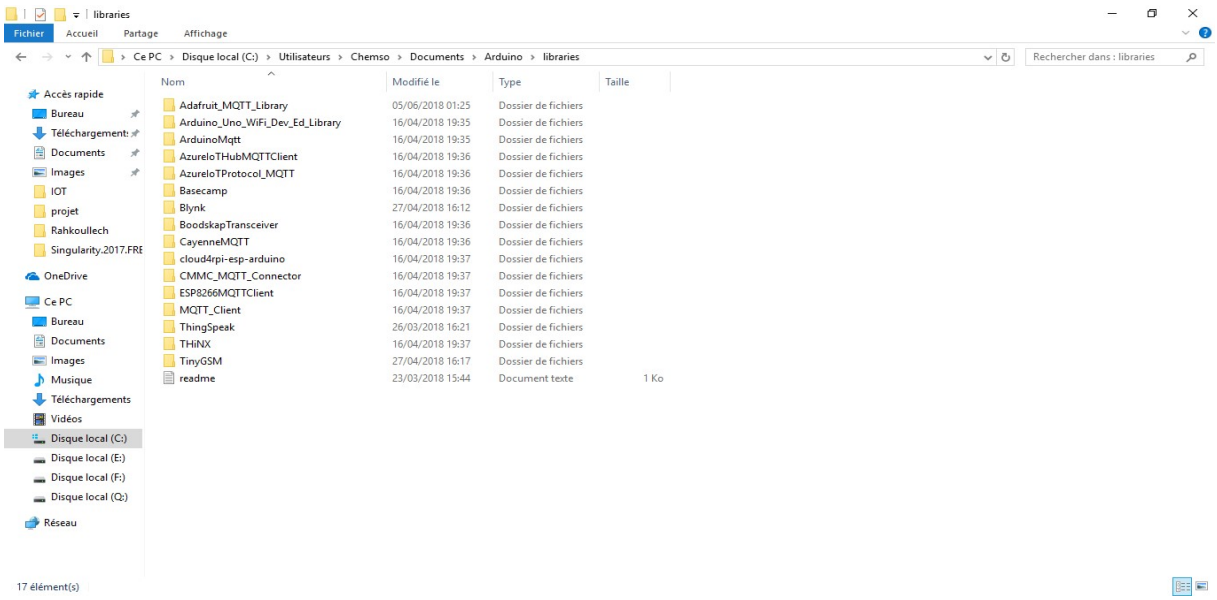


3.3.2.2 implémentations de la librairie Blynk sur arduino :

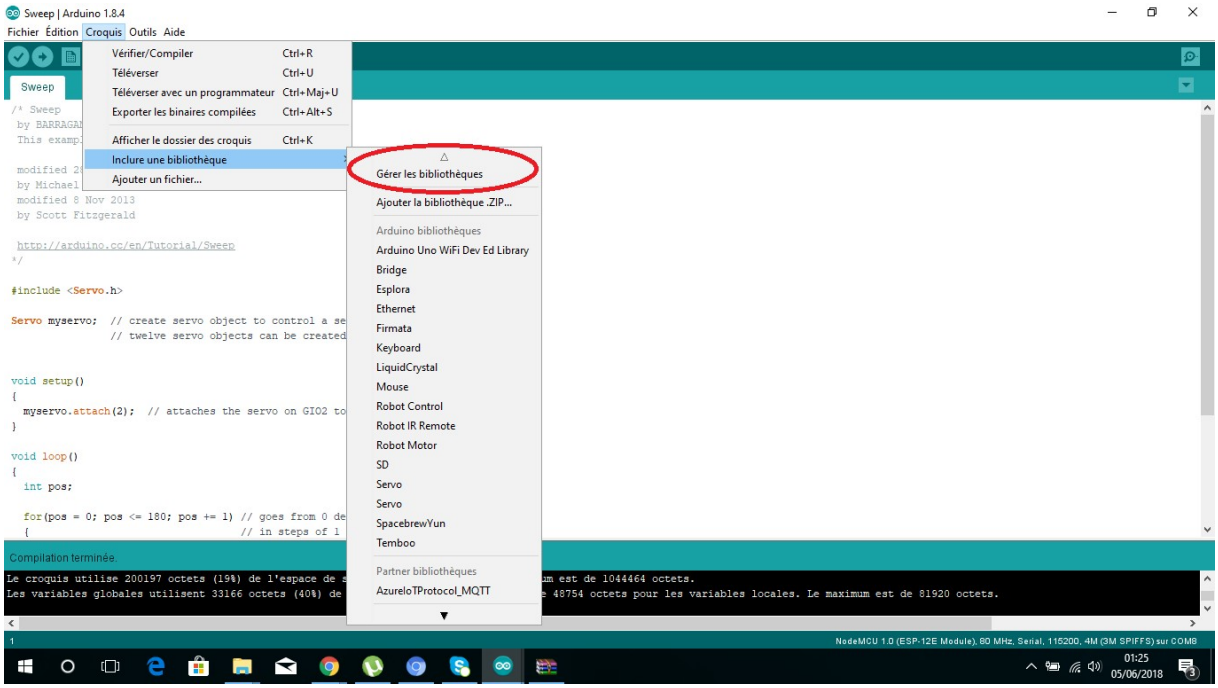
Pour implémenter la librairie Blynk il faut tout d'abordtélécharger son packages sur **github.com/blynkkk/blynk-library**comme le montre la figure



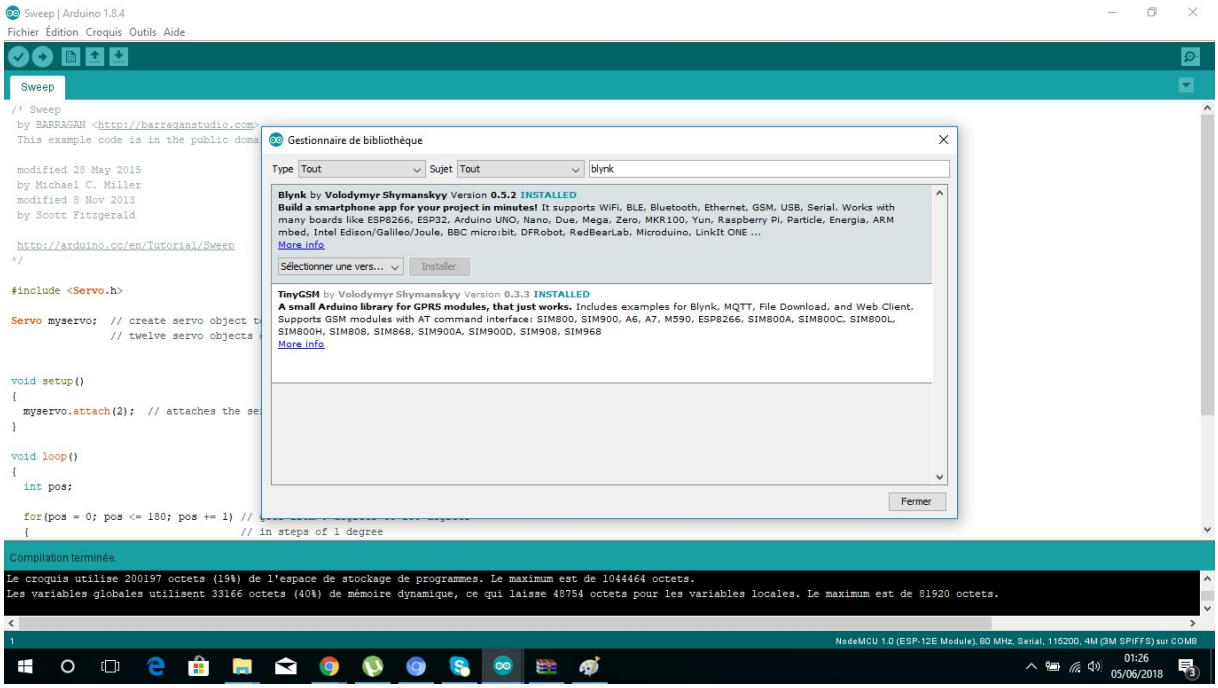
Apres le fichier télécharger, copierce fichier sur les librairies d'ARDUINO



Apres il faut aller sur arduino sur gérer les bibliothèques



Ensuite vous rechercher Blynk est vous l’installer le packages



CONCLUSION

Une parfaite combinaison entre le software et le hardware représente la meilleur manœuvre d'une conception électronique, dans ce chapitre nous décrivons le hardware utilisé pour le projet et aussi le software adopté, ainsi que leur installation et leurs configuration étape par étape pour un bon fonctionnement

Chapitre 4 : conception et réalisation

4.1 Introduction :

Dans ce chapitre nous allons apprendre à utiliser et programmer nos microcontrôleurs le NodeMcu ESP8266 et le RASPBERRY PI ZERO.

Dans un premier temps nous allons voir comment concevoir notre robot mobile à commande à distance, puis dans un second temps, nous allons voir une simple présentation concernant les composants du robot mobile et sa réalisation électronique.

Et enfin nous présenterons les différentes procédures, programmes, et l'explication pour le contrôle de notre robot mobile via l'application que nous avons choisi.

4.2 Conception :

Pour concevoir notre robot à commande à distance il doit rependre au cahier des charges suivantes :

Le système proposé doit être capable de se déplacer d'une position vers une autre positiontout en étant commandé par un smartphone via une application mobile qui sera intégré d'une caméra embarqué sur le robot commandé aussi par l'application , et qui retransmettra le flux de vidéo en temps réel sur le smartphone ou un ordinateur.

4.2.1 Les composants utilisés :

4.2.1.1 Composant hardware :

4.2.1.1.1 Camera web SKYMAX :

La camera web SKYMAX est une petite caméra vidéo qui alimente ou diffuse son image en temps réel vers ou via un ordinateur vers un réseau informatique, le flux vidéo capture peut être sauvegarde, visualise ou envoyé a d'autre réseau via des systèmes tels que l'internet, la camera web est généralement connectée par un câble USB, ou un câble similaire ou elles sont intégré dans un matérielles informatiques tels que les ordinateurs portables

4.2.1.1.2Servomoteur SG90

Le SG90 est un servomoteur capable de tourner environ 180 °. En outre, il est très petit et léger avec une puissance de sortie élevée, et fonctionne comme les types standard mais plus petit. Vous pouvez utiliser n'importe quel code servo, matériel ou bibliothèque pour contrôler ces servos. Il est livré avec 3 cornes (bras) et du matériel.

Ces caractéristiques :

-Une de tension de fonctionnement: 4,8 V (~ 5V)

-Une vitesse de fonctionnement: 0,1 s / 60 degrés

-Un couple de blocage: 1,8 kgf • cm

-Une largeur de bande morte: 10 µs

-Une plage de température: 0 ºC - 55 ºC

4.2.1.1.3 : Moteur FT DC 002 :

Ce kit de châssis à deux roues en métal avec moteurs à courant continu vous fournit un cadre de châssis en métal de haute qualité avec deux moteurs à courant continu, des roues et roulette, ainsi qu'une plaque métallique supérieure avec le matériel de montage.

On disposer de 2 plaques rondes en aluminium anodisé de couleur rouge, de 2 moteurs DC, de 2 roues équipées de pneus en silicone, d'une roue libre (ball caster) avec bille en plastique pour réaliser l'équilibre et de tous les éléments pour le monter.

On peut facilement crée un robot mobile de contrôle à distance on ajoutant nos microcontrôleurs qui sont le NodeMcu ESP8266 et le Rasberry PI ZERO.

Ces caractéristiques :

-Un matériel Aluminium Anodise

-Une tension du moteur: 3-6 VDC

-Un courant du moteur: 200-400 mA

-Un courant d'arrêt: 1.5A

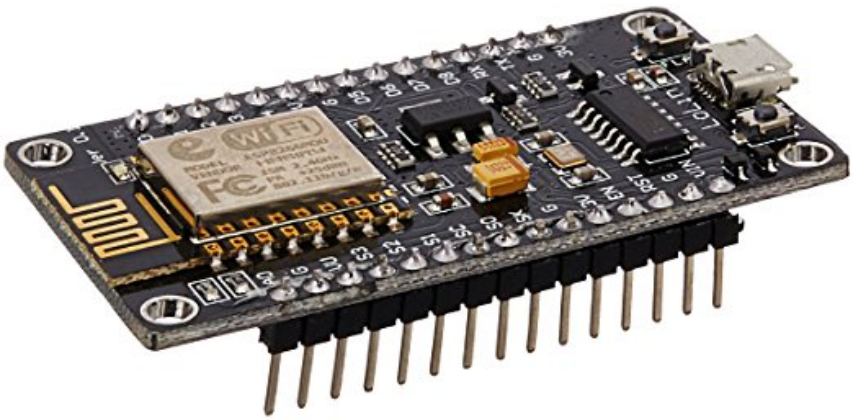
-Une longueur: 100 mm / 3,9 po

-Une largeur (avec roues): 109 mm / 4.3 po

-Une hauteur (roues et 2 plaques): 66 mm / 2.6 po

-Sa couleur: Rouge

4.2.1.1.4NodeMcu ESP8266

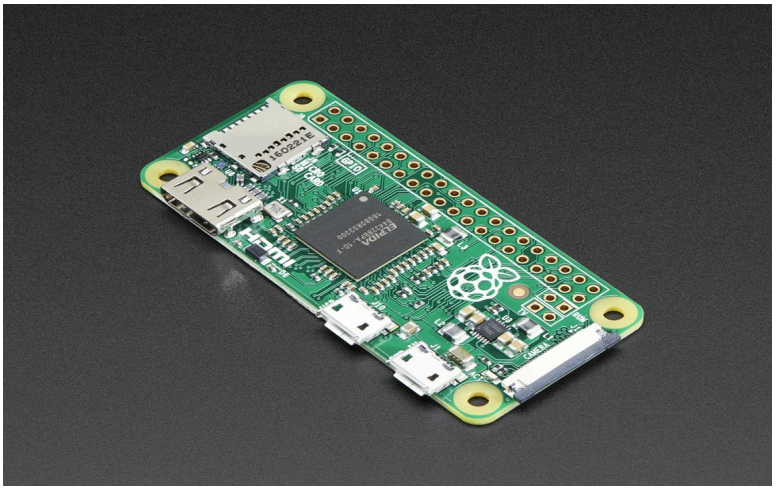


NodeMcu Est une carte de développement qui fonctionne sur l'ESP8266 avec le SDK Espressif non-OS et le matériel basé sur le module ESP-12. L'appareil dispose de 4 Mo de mémoire flash, 80 MHz d'horloge système, environ 50 Ko de RAM utilisable et un émetteur-récepteur Wifi sur puce.

L'ESP8266 est une puce hautement intégrée conçue pour les besoins d'un nouveau monde connecté. Il offre une solution de réseau Wi-Fi complète et autonome, ce qui lui permet

d'héberger l'application ou de décharger toutes les fonctions réseau Wi-Fi d'un autre processeur d'application.

4.2.1.1.5**Rasberry PI ZERO**



C'est le plus fin, le plus réduit Raspberry Pi à ce jour. C'est un peu comme le petit cousin du Pi, avec juste une fente pour carte micro SD, un port mini HDMI, deux ports micro USB (un pour l'alimentation, un pour USB) et 512 Mo de RAM. Il a une puce de processeur 1 GHz unipolaire, similaire aux Pi A + et B +.

Caractéristiques de Pi Zero :

BCM 2835 SOC @ 1 GHz

1 GHz

512 Mo de RAM

micro-SD

mini-HDMI

Micro-B USB pour les données

Micro-B USB pour le pouvoir

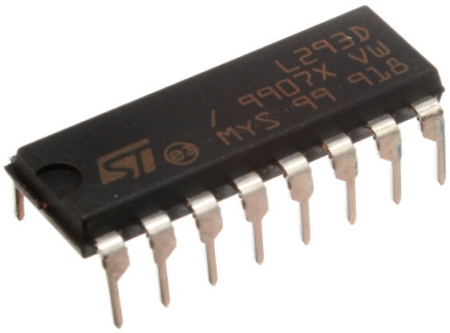
Connecteur de caméra CSI (nécessite un câble adaptateur)

Connecteur GPIO 40 broches non peuplé

Compatible avec les addons HAT existants

Dimensions: 65mm x 30mm x 5mm

4.**2**.**1**.**1**.**6La carte de puissance L293D :**



Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur.

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux.

Il est également possible de constituer deux pont-h afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

Il est important de noter que le L293D peut délivrer au maximum 600mA, veuillez donc choisir vos moteurs en conséquence.

Caractéristiques du L293D :

Nombre de pont-H: 2

Courant Max Régime continu: 600mA (x2)

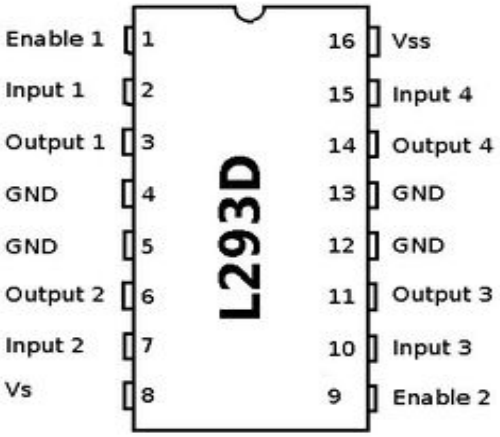
Courant de pointeMax < 2ms: 1200mA

VS Max Alim moteur: 36v

VSS Max Alim logique: 7v

Nombre de Broches: 16 DIP

Perte de tension: 1.3v

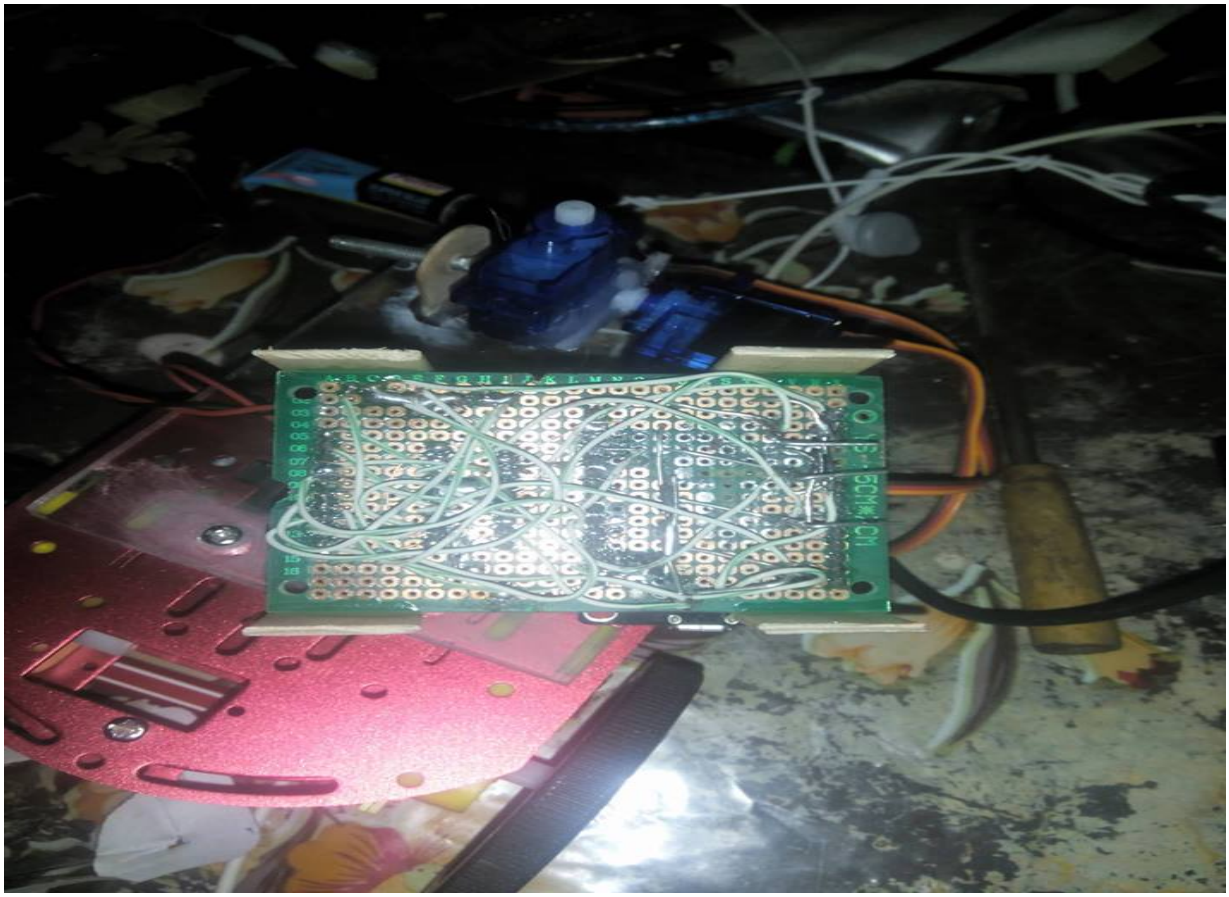


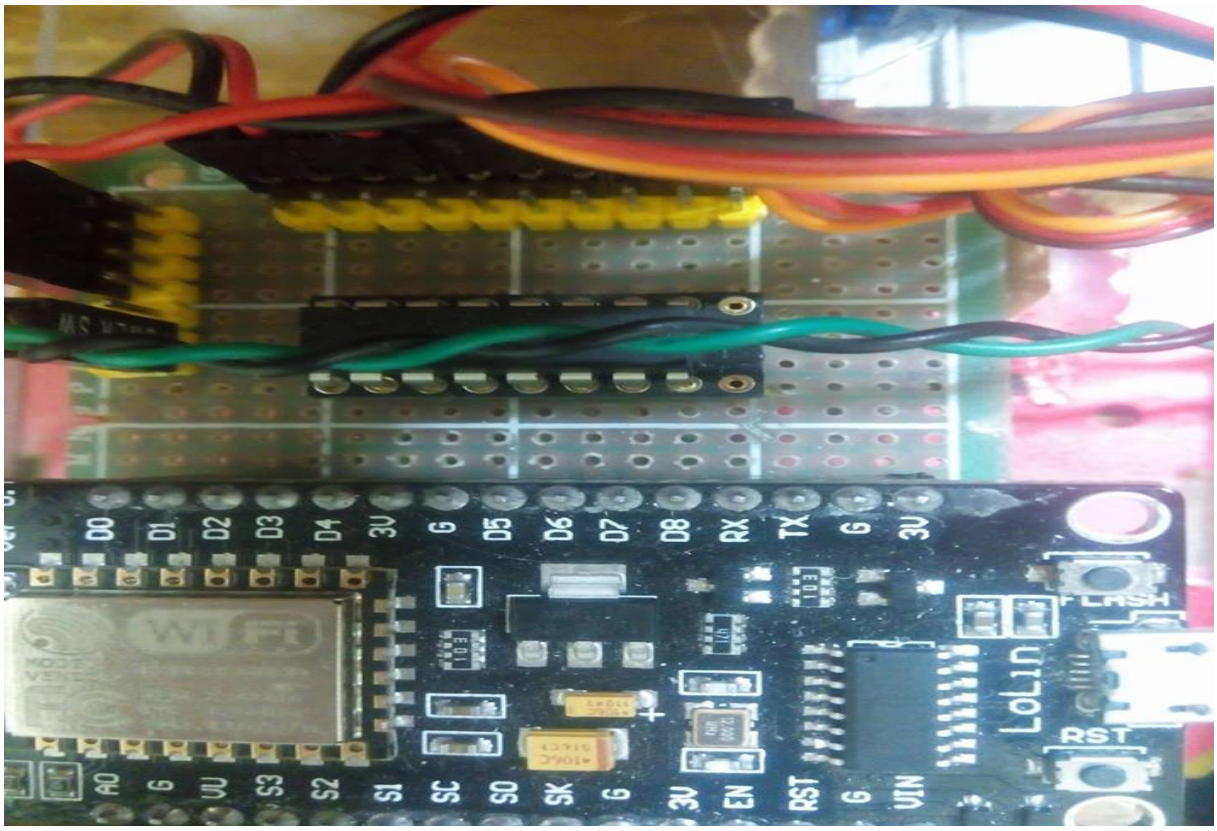
Branchements du L293D

4.3 Réalisation :

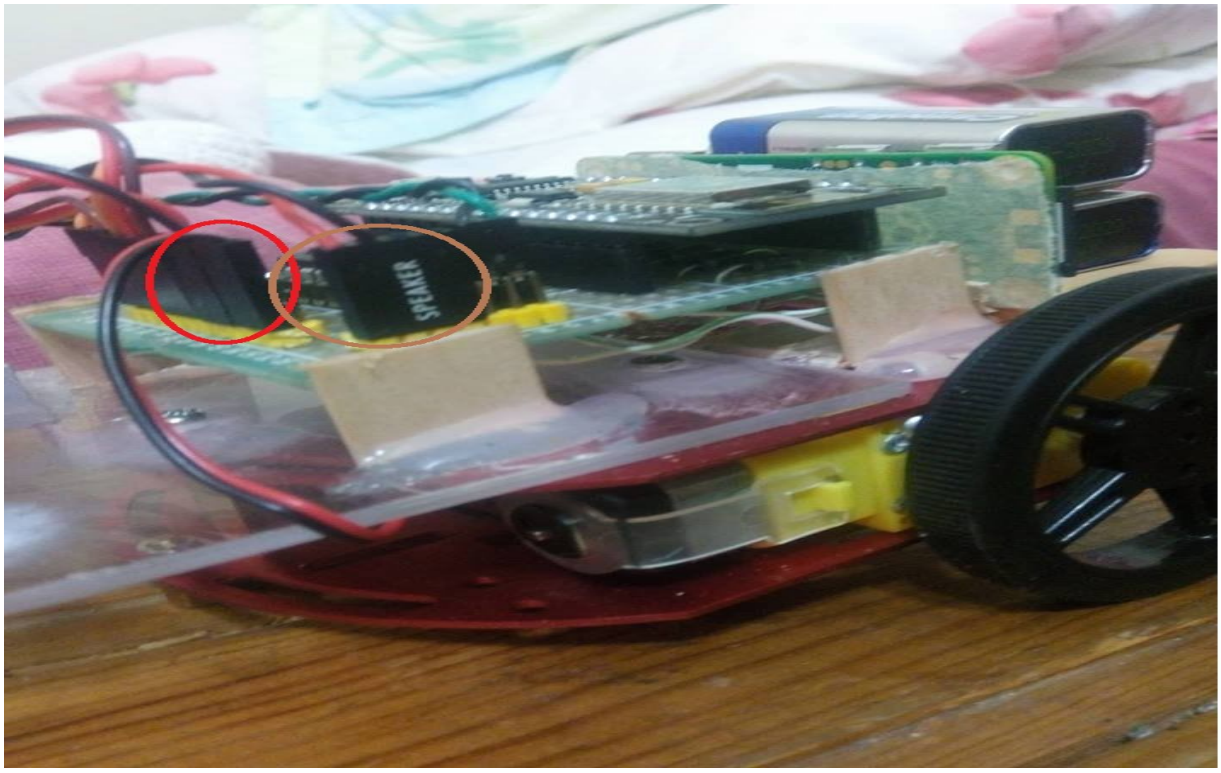
4 .3.1 Montage du robot :

-La première étape du montage de notre robot consiste à faire un mini système qui relie notre carte NodeMcu avec la carte de puissance L293D qui va amplifier la tension de commande , et de faire les branchements nécessaire pour faire marcher les moteurs et les servomoteurs de notre robot comme c'est montrer dans les figure suivantes :

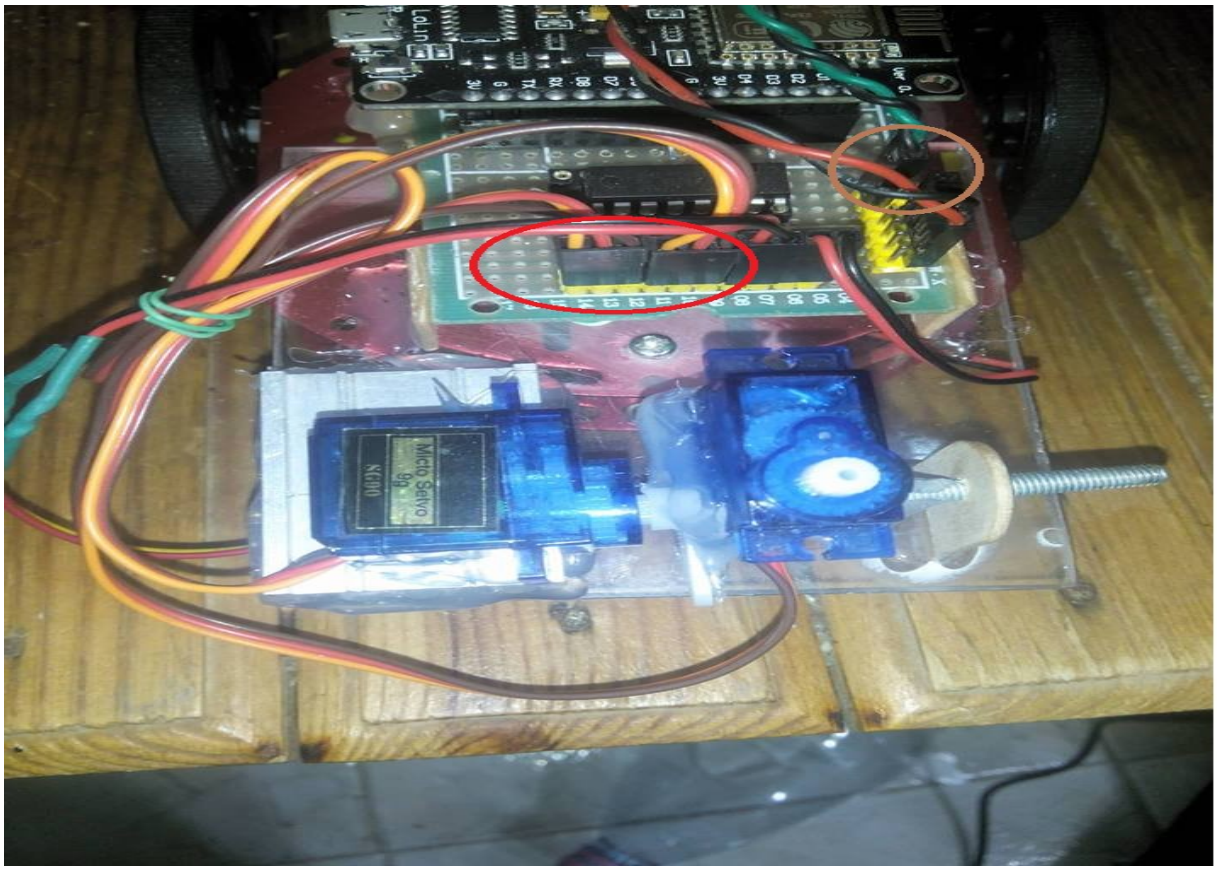




-Puis mettre le petit système sur lemoteur FT DC 002, il faut relier les deux moteurs pour leurs fonctionnements comme c'est montrer dans la figure encerclé par le rouge, et une source d'alimentation comme c'est montrer par le cercle marron :



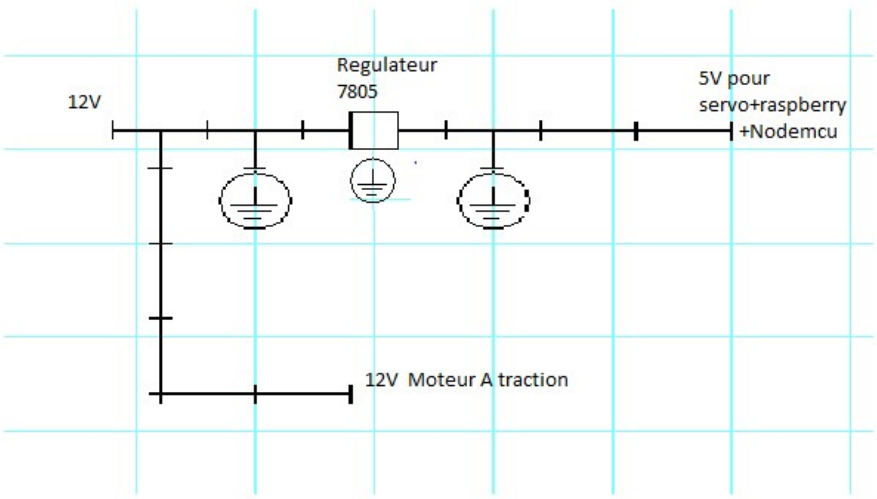
-Pour le contrôle de la camera WEB dans tous ces angles possible, sa réalisation consiste à faire un montage de deux servomoteurs l'un pour pivoter verticalement, et l'autre horizontalement avec leur alimentation sur le petit système voir la figure suivante :



-Enfin pour la réception du flux de l'image de la camera intégrer en temps réel, c'est de lié le RASPBERRY PI ZERO a la camera WEB du robot, et une carte WIFI N802 à partir d'un USB hub 3 0 voir la figure suivante :

4.3.2Le montage de l'Alimentation général du robot :

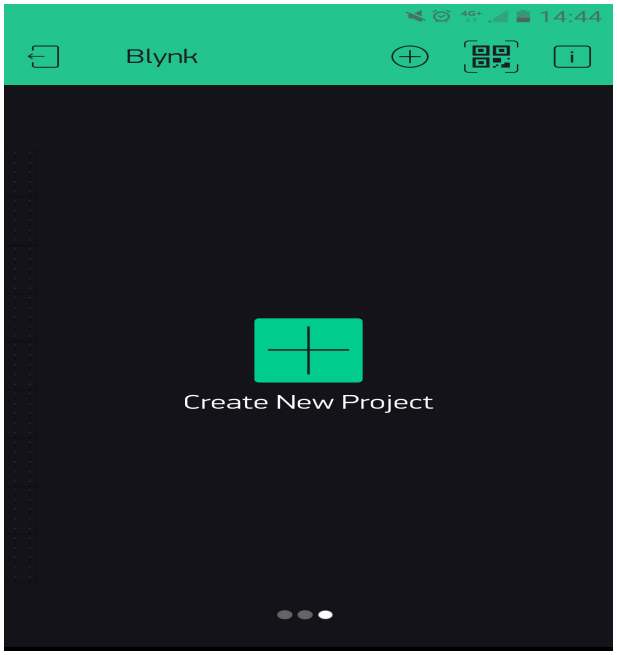
Le schéma ci-dessous, montre la tension optimale qui permettra la marche de tous les composants (RASPBERRY PI ZERO, NodeMcu, les moteurs à traction et les servomoteurs)



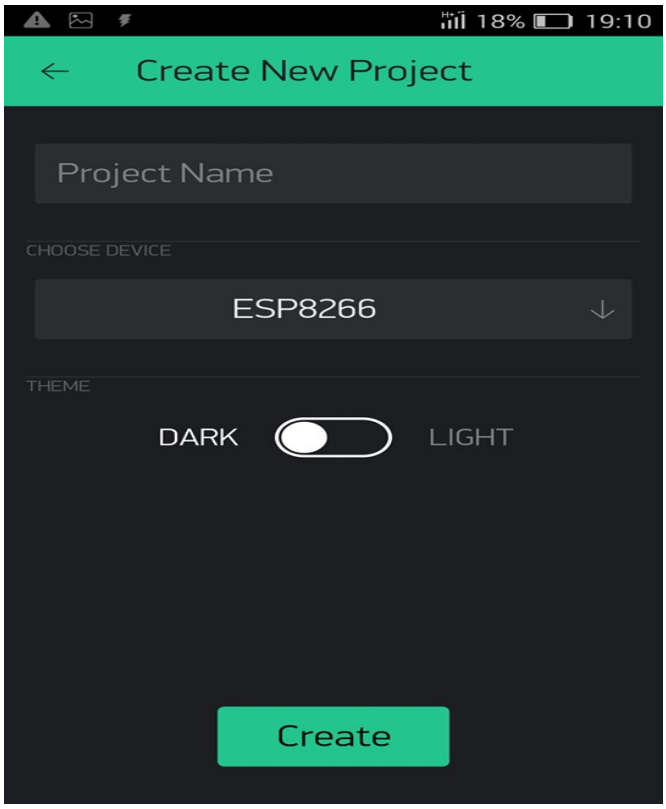
4.4L'organigramme du programme télé-verser sur l'arduino

4.5 Création et implémentation du projet sur blynk :

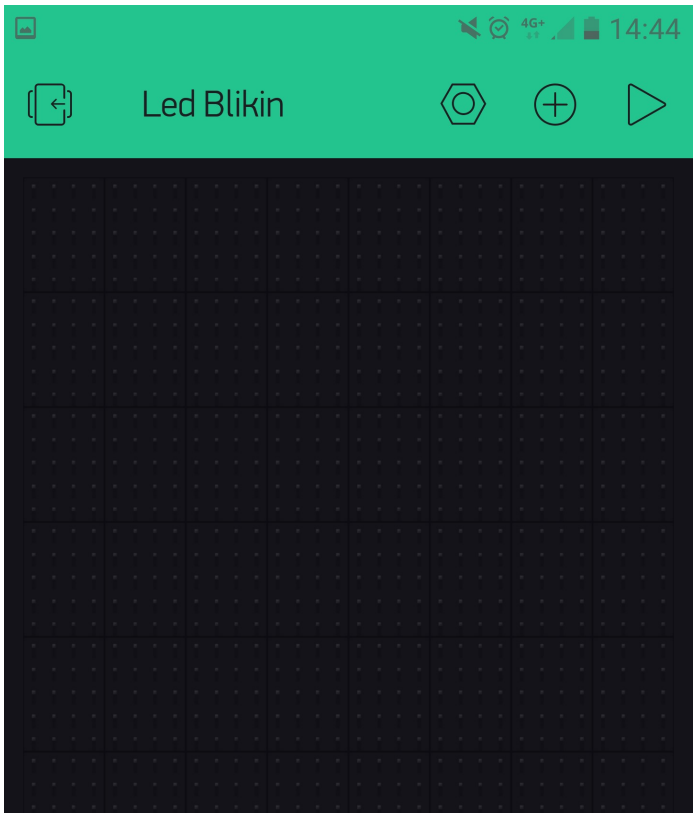
Créer un nouveau projet

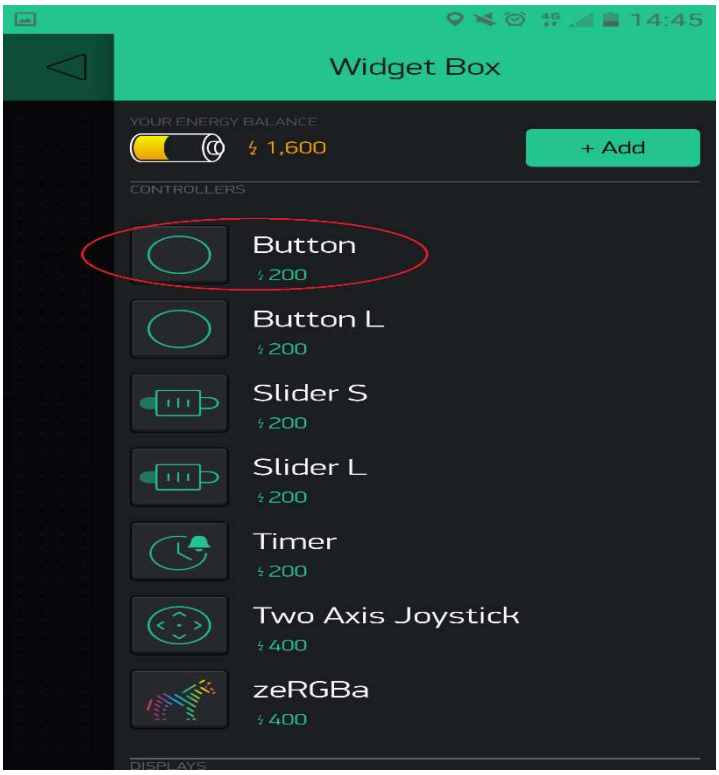


Choisir un nom de projet (ici: « comRobot »), le modèle de microcontrôleur utilisé (ici: NodeMcu ESP8266) et envoyez par e-mail le token (code API généré) généré, il sera utile pour la liaison entre l'NodeMcu et notre smartphone.

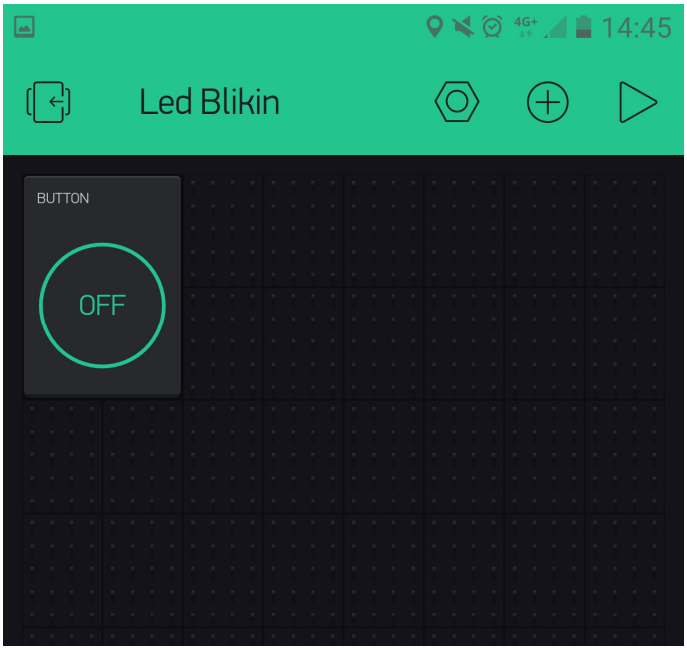


Il faut donc ajouter un bouton qui simulera l'état de sortie de la Pin en question. Pour cela, cliquez sur le petit « + » encerclé, cela nous donnera accès à une multitude de widgets. Nous choisirons dans notre cas le bouton du haut

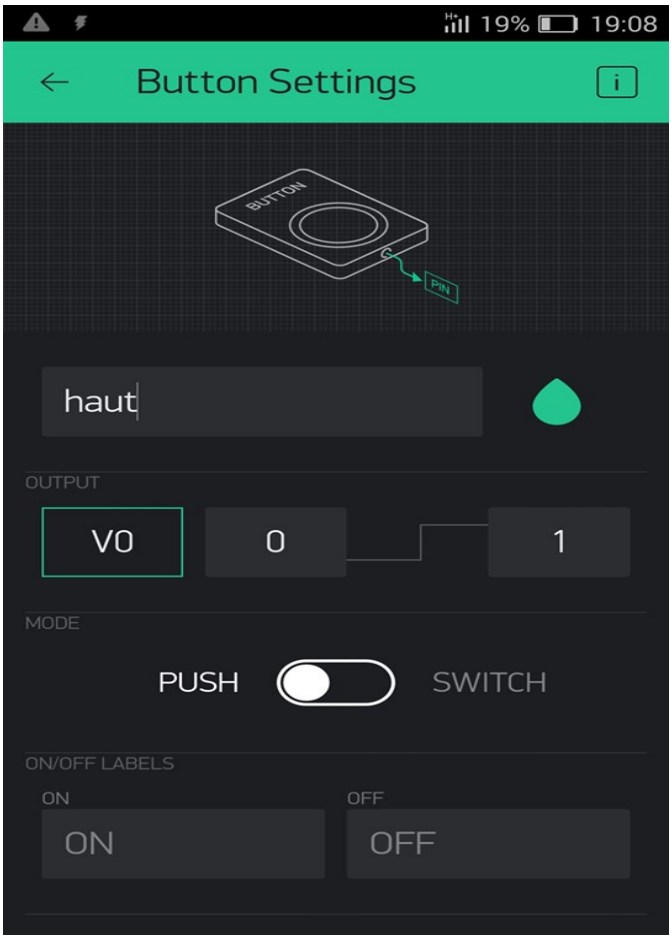




Le voilà maintenant placé nous allons le configurer : cliquez dessus.



Ensuite il nous faut choisir le pin sur laquelle l'application va interagir (ici:). Nous voulons que le bouton se comporte comme un switch ou un push et on peut également personnaliser le message affiché en fonction de l'état du bouton (ici: Light ON/Light OFF).



4.6resultats du raspberry

Conclusion

Dans ce dernier chapitre présente la conception et la réalisation du robot mobile a commande à distance ainsi que l'organigramme et l'implémentation de l'application Blynk pour le contrôle du robot afin d'assurer un bon fonctionnement, et permettre un contrôle efficace sur le robot mobile.

Conclusion général

Au bout de notre cursus en master système embarque, nous avons été chargés de réaliser un projet de fin d'étude. Notre travail c'est basé sur la conception et la réalisation d'un robot mobile commandé à distance. Ce projet était alors une occasion d'apprendre à travailler en binôme. Ceci nous amener à découvrir et d'enrichir nos connaissance dans le domaine de l'IoT, ce dernier est devenue un grand domaine de recherche et un grand marche de travail au monde.

L'objectif de ce travail a été de concevoir un robot mobile commandé à distance muni d'une caméra en se basant sur la technologie d'internet of things (IoT), le tout contrôler via une application mobile et d'en recevoir les images de la camera en temps réel. Pour cela nous avons utilisé la carte NodeMcu ESP8266 et le RASPBERRY PI ZERO qui sont encore nouveaux dans le domaine des systèmes embarque.

Durant ce projet, il nous a été confié la mission d'étudier et de mettre en place le robot mobile commandé à distance. Pour cela, notre travail a été décomposé en quatre étapes la première et la deuxième partie consistait à approfondir nos connaissances sur l'IoT sur ces différents aspects, et le choix du Protocol de données adopté pour ce projet et ses différentes plates-formes associés, la troisième partie a été consacré à l'environnement du travail et le matériels utilisé et enfin le dernier chapitre est pour la conception et la réalisation du robot mobile afin d'assurer son contrôle total et permettre à la camera intégré de retransmettre en temps réel.

Cela n'empêche pas d'avoir rencontrés pleins d'obstacle et de contraintes, notamment le manque de composants électroniques ce qui nous a gêné pendants la réalisation de notre robot mobile, ainsi que la carte NodeMcu ESP 8266 et le RASPBERRY PI ZERO sont nouvelles on a compté seulement que sur quelque documentations et la documentation sur internet, d'où nous avons rencontré quelque problèmes de programmation et configuration des différents modules électroniques car cela était notre premier expérience dans ce domaine.

Finalement, on espère par notre travail, apporter une validation pratique de ces techniques et donner une bonne cause pour mieux explorer de domaine d'internet des objets.