Institut National des postes
et télécommunications Rabat
Filière Systèmes Embraqués
et Services Numériques

PROJET DE DÉVELOPPEMENT

CONCEPTION ET REALISATION D'UNE MAISON INTELLIGENTE

Présenté par:

- FAHYM Abd Elfattah
- LAAOUINI HAITAM

Enquadré par:

M.OUSSAMA ELISSATI
Mlle. ASSIA ELHADBI

Année de formation : 2021/2022

Dédicace

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que nos parents

méritent pour tous les sacrifices qu'ils n'ont cessé de nous donner.

Pour cela, nous dédions ce modeste travail à nos parents respectifs. Nous tenons à rappeler qu'aucun hommage ne pourrait être à la hauteur de l'amour

dont ils ne cessent de nous combler, que Dieu leur procure bonne santé et longue vie
Ce travail est le fruit de leurs sacrifices

FATMA et LAHOUAJ ✍

Remerciement

Nous remercions Dieu « ALLAH » tout puissant qui nous a Donné le courage pour réaliser le défi de ce projet, et la force de finaliser ce travail.

Nous voudrons dans un premier temps remercier Mme Assia Elhadbi, Mr.Issati, Mr. Alami ainsi les membres du jury de nous avoir fait l'honneur d'accepter d'évaluer notre projet de développement. Nous avons également été honorés de votre participation à notre jury de soutenance malgré la période critique que nous vivons.

Nous remercions notre coordinateur de filière monsieur Oussama EL ISSATI pour sa disponibilité. Nous le remercions de nous avoir encadrés, orienté, conseillé et surtout de nous avoir fait confiance. Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce projet.

Je ne saurais oublier le corps professoral de l'École Nationale des Postes et Télécommunications Rabat pour la formation d'excellence qui nous donne.

Merci!

Abstract

Nowadays, the Internet has been developed quickly, which the use is not limited only to the network management , but also extends to the management of objects ,which called the internet of things that is used in several domains and we can mention home automation, currently called smart home. As part of this research project, our goal is to integrate IoT technology in smart homes based on the four basic elements: Central organs. Sensors. Actuators. Control interface. Keywords: smart home, internet of things, home automation, sensor, actuator, control interface.

Résumé

Aujourd'hui plus que jamais, l'internet s'est développé rapidement , dont l'utilisation ne se limite pas de la gestion des réseaux seulement, mais s'étend aussi à la gestion des objets et c'est ce qu'on appelle l'internet des objets qui est utilisée dans plusieurs domaines dont on cite le domaine de la domotique, actuellement appelé maison intelligente .Dans le cadre de ce projet de recherche, notre objectif est d'intégré la technologie IoT dans les maisons intelligentes en se basant sur les quatre éléments de base : Organes centraux. Capteurs. Actionneurs. Interface de commande. Mots clé : maison intelligente, internet des objets, domotique, capteur, actionneur, interface de commande.

Contents

Table des figures

Introduction générale

Pareil à notre vie en général, nos maisons se trouvent assez dotées de technologie. Nos habitats et les habitats du futur répondent à une probable insatisfaction innée de l'homme qui croit augmenter sa dominance sur son environnement par la technologie. On voit donc que sa maison répond à lui et à ses besoins. Ainsi, la technologie sert à la fois ses besoins, ses habitudes et son envie de confort. Elle prend en compte des situations significatives dans sa vie quotidienne : quitter son domicile, se réveiller dans un habitat chauffé, créer une ambiance désirée, avoir le café prêt et les volets ouverts.

Piloter notre bien-être, contrôler nos appareils et nos accès de près ou à distance en quelques clics. Construire un intérieur rassurant pour nous et nos proches afin de transformer notre maison en habitat moderne, intelligent et sécurisé est devenu un besoin de plus en plus exigé.

Pour cela il fallait rassembler et intégrer l'ensemble des techniques de l'électronique, de l'informatique, d'automatisme, de physique du bâtiment et des télécommunications afin de centraliser le contrôle de nos différents systèmes et sous- systèmes (volets roulants, porte de garage, portail d'entrée, prises électriques, chauffage, etc.). C'est autour de ce rassemblement et cette intégration que notre étude s'articule.

I- Aperçu général sur les réseaux locaux et LPWAN pour IOT

Les réseaux locaux et les LPWAN (Low-Power Wide-Area Network) sont deux types de technologies de réseau qui sont souvent utilisés dans l'Internet des objets (IoT).
Les réseaux locaux, également connus sous le nom de LAN (Local Area Network), sont des réseaux de communication qui permettent à des périphériques tels que des ordinateurs, des imprimantes et des serveurs de partager des ressources et des données entre eux sur une zone géographique limitée. Les réseaux locaux peuvent être câblés ou sans fil, et ils sont souvent utilisés dans des environnements tels que les bureaux, les maisons et les établissements d'enseignement.
Les LPWAN, d'autre part, sont des réseaux de communication à faible consommation d'énergie qui peuvent fournir une connectivité à longue portée pour les dispositifs IoT. Les LPWAN sont souvent utilisés dans des applications où les dispositifs doivent fonctionner pendant de longues périodes avec une source d'énergie limitée, tels que des capteurs de température, des compteurs d'eau et d'électricité, et des systèmes de sécurité. Les LPWAN utilisent généralement des fréquences radio publiques pour la communication et offrent une couverture étendue allant de quelques kilomètres à plusieurs dizaines de kilomètres.
Les LPWAN ont des avantages en termes de coût, de consommation d'énergie et de couverture de réseau, ce qui en fait une solution populaire pour les applications IoT qui nécessitent une surveillance à distance et un contrôle des actifs. Les réseaux locaux sont plus adaptés pour les applications qui nécessitent une connectivité rapide et fiable sur des zones géographiques plus petites.
En résumé, les réseaux locaux et les LPWAN sont deux technologies de réseau distinctes avec des avantages et des applications spécifiques dans le domaine de l'IoT. Le choix entre les deux dépendra des exigences spécifiques de l'application et de la portée géographique requise

1) Bluetooth :

Bluetooth (standard IEEE 802.15.1) est un protocol de communication sans fil, pour les appareils électroniques fonctionnant dans la bande libre des 2,4 GHz et fondée sur l'étalement de spectre par saut de fréquence (FHSS – Frequency Hoping Spread Spectrum).

FIGURE 0.1: Bluetooth

2) Zigbee :

ZigBee est un protocole de haut niveau permettant la communication d'équipements personnels ou domestiques équipés de petits émetteurs radios à faible consommation ; il est basé sur la norme IEEE 802.15.4 pour les réseaux à dimension personnelle (Wireless Personal Area Networks : WPAN).

FIGURE 0.2: Zigbee

3) RFID :

Le sigle RFID signifie Radio Frequency Identification, comprenez radio- identification. Il désigne une méthode utilisée pour stocker et récupérer des données à distance en utilisant des balises métalliques, les « Tags RFID ». Ces balises, qui peuvent être collées ou incorporées dans des produits, réagissent aux ondes radio et transmettent des informations à distance. Cette technologie pourrait, à terme, remplacer les codes-barres.

FIGURE 0.3: RFID

4) Wifi :

Le terme Wifi est l'abréviation de Wireless Fidelity. Il désigne un système de connexion à internet sans fil. L'accès à Internet se fait grâce à la transmission d'ondes radioélectriques. Il permet également de connecter plusieurs équipements entre eux (une imprimante à un ordinateur par exemple).

Figure 0.4: Wifi

5) La 4ème génération (4G) :

En télécommunications, la 4G est la quatrième génération des standards pour la téléphonie mobile correspondant au LTE-Advanced (IMT-Advanced). Succédant à la 2G, la 3G et 3.5G (HSPA) ; elle permet des débits plus élevés jusqu'à 3 Gbps en LTE-Advanced et 300 Mbps en LTE Cat 5 et 6.

Figure 0.5: La 4ème génération (4G)

6) La 5ème génération (5G) :

Le terme 5G est déjà évoqué par les industriels de l'électronique dans les années 1980 ; cette technologie pourrait voir le jour vers 2020. Le développement de la 5G en Chine est principalement l'œuvre de China Mobile, Hawaii, et ZTE, en coopération avec Ericsson depuis 2015.

FIGURE 0.6: La 5ème génération (5G)

7) Conclusion :

En résumé, les réseaux locaux tels que Bluetooth, Zigbee, RFID et Wifi sont utilisés pour connecter des dispositifs IoT à courte portée et à faible consommation d'énergie dans des environnements personnels, domestiques, industriels et publics. Les réseaux LPWAN tels que la 4G et la 5G sont utilisés pour connecter des dispositifs IoT à longue portée et à haut débit dans des domaines tels que les villes intelligentes, l'automatisation industrielle et la télémédecine.

II- Généralité sur la maison intelligente

1) Historique de la domotique :

Brièvement, le mot domotique a été introduit dans le dictionnaire « le petit Larousse » en 1988. Ce mot a été construit à partir de « Domus », la demeure de maître en latin, associé au suffixe « tique », couramment employé pour évoquer le terme des technologies (automatique, électronique, électrique, informatique). On associe souvent le début des travaux domotiques aux années 1970, voire 1980, avec les problématiques énergétiques dues aux crises pétrolières qui ont considérablement affecté le domaine de la construction et de l'exploitation du bâtiment. Depuis le milieu des années 1990, un autre segment, orienté sur la microinformatique et les loisirs numériques, se développe. Cette nouvelle apparition marque en particulier l'introduction de l'informatique dans l'habitat et l'apparition des supports numériques : les cédéroms, puis les DVD et internet. Ainsi aujourd'hui, la gestion de l'habitat, la sécurité, les réseaux de communication et les loisirs numériques esquissent le paradigme de domotique.

2) La maison intelligente :

La maison intelligente est une maison avec des fonctions qui simplifient le quotidien de ses habitants, pour générer de l'énergie et assurer certaines fonctions avec un certain degré de confort de toiture et de sécurité. Elle est en constante évolution et s'ouvrant sur le monde. C'est un mot récent de la langue française et il est en réalité la somme des mots « doums » qui signifient domicile en latin et du suffixe « tique » rattaché au mot technique. La maison

FIGURE 0.7: La maison intelligente

intelligente utilise plusieurs critères clés : la sécurité, le confort de vie, les économies d'énergies et la santé et la communication.

3) La sécurité :

Le système domotique offre une protection complète des biens et des personnes grâce à l'utilisation de capteurs pour détecter les intrusions, surveiller les accès et détecter les anomalies techniques telles que les fuites d'eau ou les incendies. Les occupants peuvent recevoir des alertes en temps réel par e-mail ou sur leur téléphone portable en cas d'incident potentiel, ce qui leur permet de prendre des mesures pour protéger leur propriété et leur sécurité.

4) Le confort :

En utilisant un smartphone,la maison intelligente est capable de savoir quand vous rentrez à la maison et donc d'ouvrir le portail avant même que vous n'arriviez. Les volets peuvent s'ouvrir et se fermer au rythme du soleil, et peuvent même aller jusqu'à s'adapter à la saison et la température pour laisser entrer la lumière et la chaleur du soleil l'hiver, ou au contraire conserver le frais l'été en fermant les volets des fenêtres exposées au soleil.

5) La santé :

La maison intelligente trouve aujourd'hui de nouvelles applications dans le domaine de la santé. Afin d'améliorer l'autonomie et l'indépendance des personnes fragiles, handicapées ou âgées le souci de leurs mises en garde à distance chez eux peut être maintenant possible.

6) L'économie d'énergie :

En gérant les volets selon la saison, ainsi que le chauffage, le système domotique vous permet d'économiser de l'énergie, et donc de l'argent, même si au départ on ne recherchait que le confort en plus. La consommation d'énergie peut être suivie très finement, qu'il s'agisse de votre consommation d'électricité, d'eau, ou même de gaz.

7) La communication :

La communication dans la maison intelligente est Le mariage de l'informatique, des télécom et l'électronique. Au royaume des normes domotique, il est difficile de se retrouver. On trouve des types différents de la communication dans la smart house comme montre le figure ci dessous .

FIGURE 0.8: Les différents protocoles de communication sans fil

8) Conclusion :

La maison intelligente est une maison équipée de fonctions automatisées qui simplifient le quotidien de ses habitants en termes de sécurité, de confort, d'économies d'énergie, de santé et de communication. Elle utilise des capteurs pour détecter les intrusions, surveiller les accès et prévenir les incidents tels que les fuites d'eau ou les incendies. En termes de confort, la maison intelligente peut s'adapter à la saison et à la température pour laisser entrer la lumière et la chaleur du soleil en hiver ou conserver la fraîcheur en été. Elle peut également être utilisée pour améliorer l'autonomie des personnes fragiles, handicapées ou

âgées. En gérant les volets et le chauffage, la maison intelligente permet également des économies d'énergie. Enfin, la communication est également intégrée à la maison intelligente grâce à l'informatique, les télécommunications et l'électronique.

III- Matérial et outillage :

1) Arduino Mega :

L'Arduino Mega est une carte de microcontrôleur basée sur le microcontrôleur ATmega2560. C'est l'une des plus grandes cartes Arduino disponibles, avec 54 broches d'entrée/sortie numériques (dont 15 peuvent être utilisées pour la PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur à cristal de 16 MHz, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. La carte est conçue pour fournir une plateforme plus grande et plus puissante pour des projets plus avancés, et convient particulièrement aux projets nécessitant plus de broches d'entrée/sortie ou plus de puissance de traitement que les autres cartes Arduino peuvent fournir. Elle est compatible avec la plupart des shields Arduino, qui sont des cartes d'extension permettant d'ajouter des fonctionnalités supplémentaires à la carte.
L'Arduino Mega peut être programmé à l'aide du logiciel Arduino, disponible gratuitement sur le site web d'Arduino. Le logiciel comprend un langage de programmation simple basé sur C++ et un ensemble de bibliothèques qui facilitent la communication avec les différentes entrées et sorties de la carte. Avec l'Arduino Mega, vous pouvez créer une large gamme de projets, depuis des clignoteurs LED simples jusqu'à des systèmes robotiques complexes.

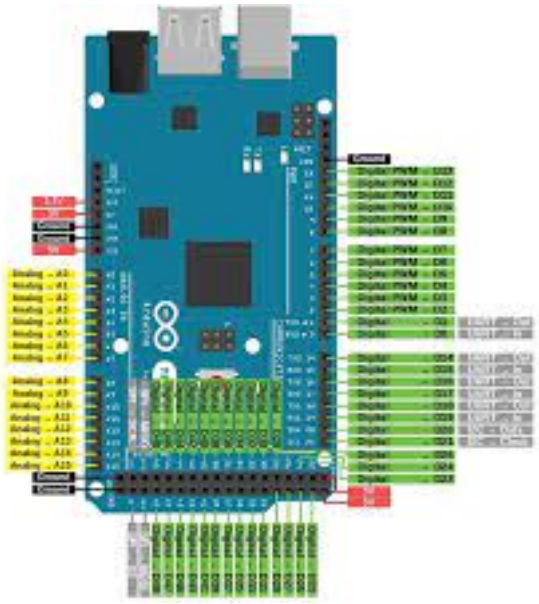


FIGURE 0.9: Arduino Mega

2) Bluetooth HC-06 :

Le HC-06 est un module Bluetooth qui permet la communication sans fil entre des appareils électroniques. Il s'agit d'un module couramment utilisé pour la communication sans fil entre des microcontrôleurs et d'autres appareils.
Le HC-06 est un esclave, ce qui signifie qu'il ne peut communiquer qu'avec un appareil maître, tel qu'un smartphone ou un ordinateur, qui initie la communication. Le module fonctionne sur la norme Bluetooth 2.0 et prend en charge une plage de communication allant jusqu'à 10 mètres.
Le HC-06 peut être facilement intégré à un projet, car il ne nécessite que quatre connexions à un microcontrôleur ou à un autre appareil : l'alimentation, la terre, la transmission (TX) et la réception (RX). Une fois connecté, le module peut être configuré à l'aide de commandes AT, qui permettent à l'utilisateur de définir différents paramètres tels que le nom de l'appareil, le débit en bauds et le code d'appariement.

Dans l'ensemble, le HC-06 est une solution polyvalente et économique pour la communication sans fil entre les appareils.

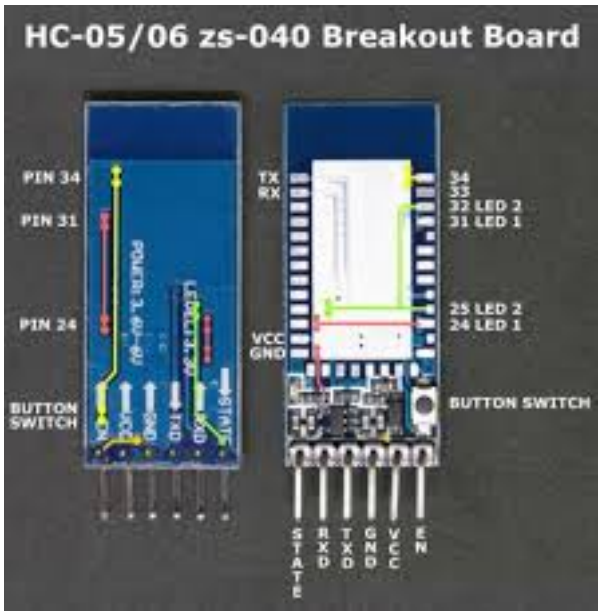


FIGURE 0.10: Bluetooth HC-06

3) ESP8266 :

L'ESP8266 est une micro-puce Wi-Fi à faible coût avec une pile TCP/IP complète et des capacités de microcontrôleur produite par l'entreprise chinoise Espressif Systems. Il a été lancé pour la première fois en 2014 et a rapidement gagné en popularité dans les communautés de fabrication et d'IoT en raison de sa petite taille, de son faible coût et de sa connectivité Wi-Fi intégrée.
L'ESP8266 peut être programmé en utilisant l'IDE Arduino, MicroPython ou d'autres langages de programmation, et est largement utilisé dans une variété de projets IoT tels que l'automatisation domestique, l'éclairage intelligent, les stations météorologiques et la détection à distance. Il dispose d'une gamme de broches GPIO qui peuvent être utilisées pour interfacer avec des capteurs et des actionneurs, et prend en charge une variété de protocoles de communication tels que SPI, I2C et UART.
Il existe plusieurs variantes de l'ESP8266, notamment l'ESP-01, l'ESP-12 et l'ESP-32, chacune avec des broches de connexion, des capacités et des fonctionnalités différentes. L'ESP-32 est une version plus puissante de l'ESP8266, avec des processeurs à double cœur, une connectivité Bluetooth et la prise en charge d'applications plus volumineuses.

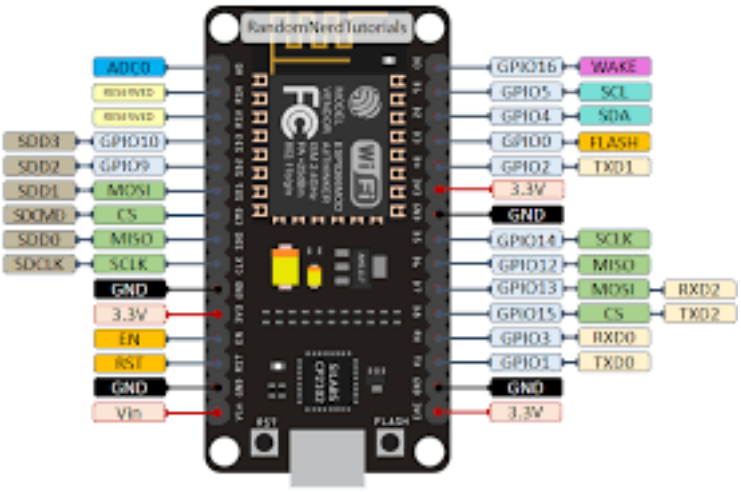


FIGURE 0.11: ESP8266

4) Raspberry Pi 4 :

Le Raspberry Pi 4 est la quatrième génération d'ordinateur monocarte Raspberry Pi. Il a été lancé en juin 2019 et succède au Raspberry Pi 3B+. Parmi les principales caractéristiques du Raspberry Pi 4, on peut citer :
Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) SoC 64 bits à 1,5 GHz
2 Go, 4 Go ou 8 Go de SDRAM LPDDR4-3200 (selon le modèle)
Wi-Fi IEEE 802.11ac 2,4 GHz et 5,0 GHz, Bluetooth 5.0, BLE
Ethernet Gigabit
2 ports USB 3.0 ; 2 ports USB 2.0.
2 ports micro-HDMI (jusqu'à 4kp60 pris en charge)
Port d'affichage MIPI DSI 2 voies, port de caméra MIPI CSI 2 voies, port audio stéréo à 4 pôles et port vidéo composite
Graphiques OpenGL ES 3.0
Décodeur H.265 (4kp60), décodeur H.264 (1080p60), encodeur H.264 (1080p30)

Le Raspberry Pi 4 peut être utilisé pour une large gamme d'applications, notamment en tant qu'ordinateur de bureau, centre multimédia, console de jeu ou pour divers projets DIY. Il fonctionne avec une variété de systèmes d'exploitation, notamment Raspbian, Ubuntu et autres.

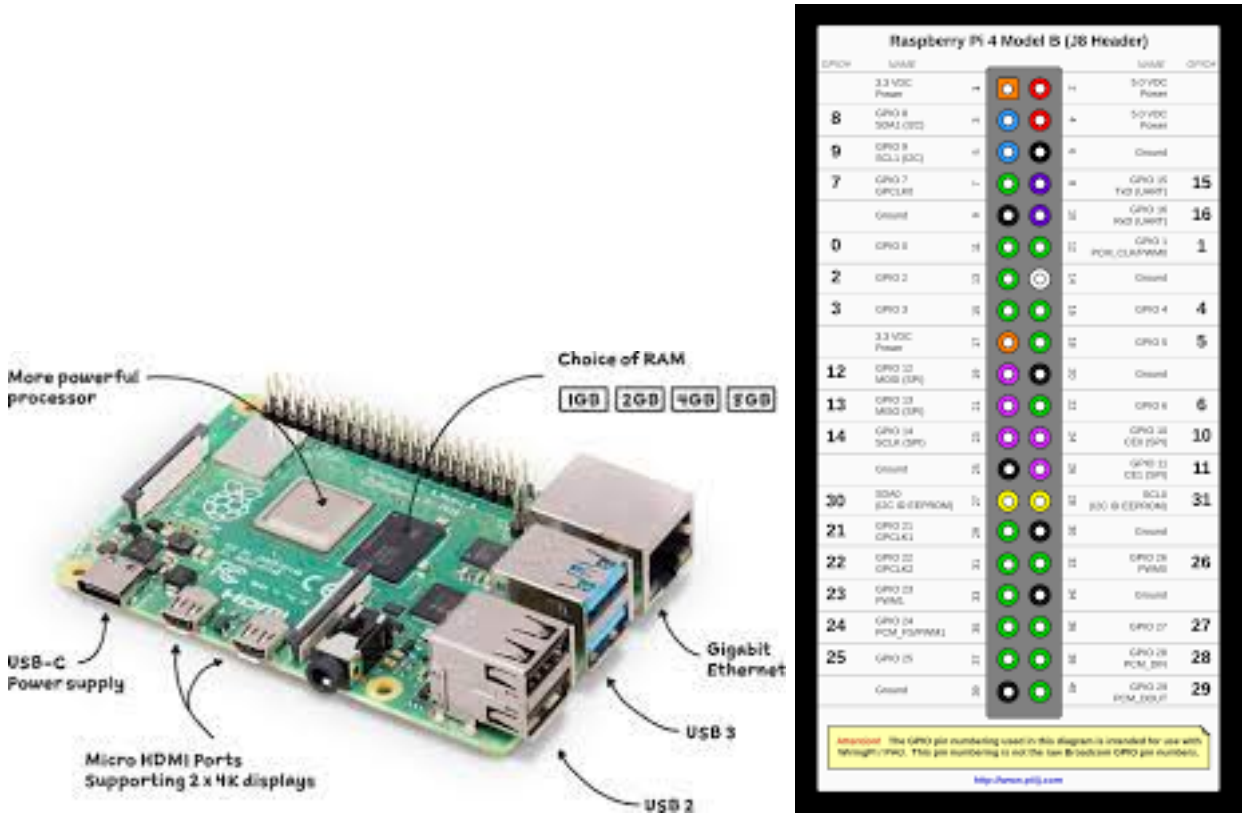


FIGURE 0.12: Raspberry Pi 4

5) Mosquitto et Node Red :

Mosquitto et Node-RED sont deux outils de l'écosystème de l'Internet des objets (IoT). Mosquitto est un broker MQTT (Message Queuing Telemetry Transport) open-source, qui est utilisé pour transférer des données entre des dispositifs IoT. Il peut être utilisé pour collecter et distribuer des données de capteurs, de l'état des dispositifs et d'autres informations pertinentes pour la prise de décision. Mosquitto est souvent utilisé avec Node-RED.
Node-RED est un outil open-source de programmation visuelle pour l'IoT, qui permet de créer des flux de traitement de données pour des dispositifs IoT. Les flux peuvent être créés simplement en faisant glisser et déposer des blocs sur une interface graphique. Node-RED permet de connecter différents dispositifs IoT entre eux, de traiter les données, de les stocker et de les visualiser. Il peut également être utilisé pour envoyer des commandes à des dispositifs, pour automatiser des tâches et pour créer des applications IoT complètes. En utilisant Mosquitto avec Node-RED, il est possible de collecter des données à partir de dispositifs IoT et de les transférer à Node-RED pour être traitées. Node-RED peut également envoyer des commandes à des dispositifs via Mosquitto. Cette combinaison d'outils est très populaire pour la création de solutions IoT car elle est facile à utiliser et peut être mise en œuvre rapidement.

FIGURE 0.13: Mosquitto avec Node-RED

6) Mit App Inventor :

MIT App Inventor est une plateforme en ligne qui permet aux utilisateurs de créer des applications mobiles pour les appareils Android sans avoir besoin d'apprendre des langages de programmation complexes. Il s'agit d'une interface visuelle de type drag-and-drop qui simplifie le processus de développement d'applications et permet aux utilisateurs de prototyper et de tester rapidement leurs idées. App Inventor utilise un langage de programmation basé sur des blocs, ce qui signifie que les utilisateurs peuvent créer des applications fonctionnelles en connectant visuellement des blocs, plutôt que d'écrire du code traditionnel.
La plateforme a été développée par Google avant d'être transférée au Massachusetts Institute of Technology (MIT) en 2012, où elle a été maintenue et mise à jour par le laboratoire d'informatique et d'intelligence artificielle (CSAIL) du MIT. MIT App Inventor est un logiciel gratuit et open-source, et sa communauté est active dans la création et le partage de ressources pour aider les autres à apprendre et à créer leurs propres applications.
Dans l'ensemble, MIT App Inventor est un excellent outil pour les débutants qui veulent se lancer dans le développement d'applications mobiles sans avoir besoin de connaissances en programmation approfondies.

FIGURE 0.14: Mit App Inventor

7) Conclusion :

Les composants essentiels pour la construction et la mise en œuvre d'une maison intelligente :
L'Arduino Mega est une carte microcontrôleur qui peut être programmée pour contrôler les différents capteurs et actionneurs de votre maison intelligente. L'ESP8266 est un module Wi-Fi qui permet de connecter votre système domotique à Internet et de le contrôler à distance. La Raspberry Pi est un ordinateur monocarte qui peut être utilisé pour héberger un serveur Mosquitto, qui est un serveur de messagerie MQTT utilisé pour connecter les différents composants de votre système domotique. Le module Bluetooth HC-06 peut être utilisé pour connecter des appareils Bluetooth tels que des haut-parleurs ou des écouteurs. Les capteurs sont des dispositifs qui peuvent détecter des changements dans l'environnement, tels que la température, l'humidité et la lumière, et envoyer des données à votre système domotique. Node-RED est un outil de programmation graphique qui permet de créer des flux de données pour votre maison intelligente. Enfin, le MIT App Inventor est un environnement de développement d'applications qui permet de créer des applications pour contrôler votre maison intelligente à partir de votre smartphone. En combinant ces composants et outils, vous pouvez créer un système domotique intelligent pour votre maison.

IV- Conception et Réalisation du maison intelligente :

1) Première solution utilisant le Bluetooth

a- Circuits électriques : Schéma, câblage et branchement

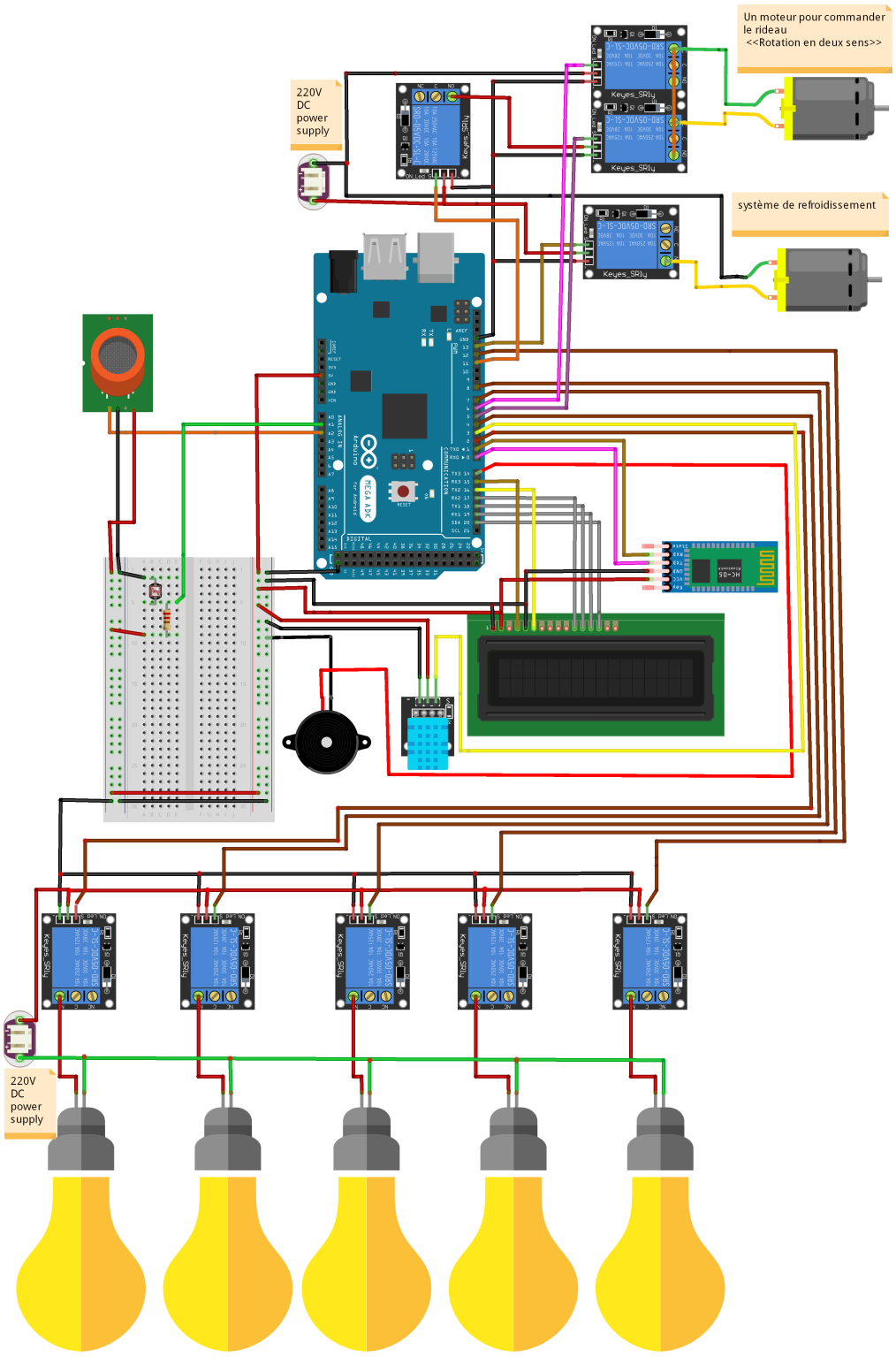


FIGURE 0.15: Circuits électriques

b- Conception de l'interface utilisateur avec Mit App Inventor :

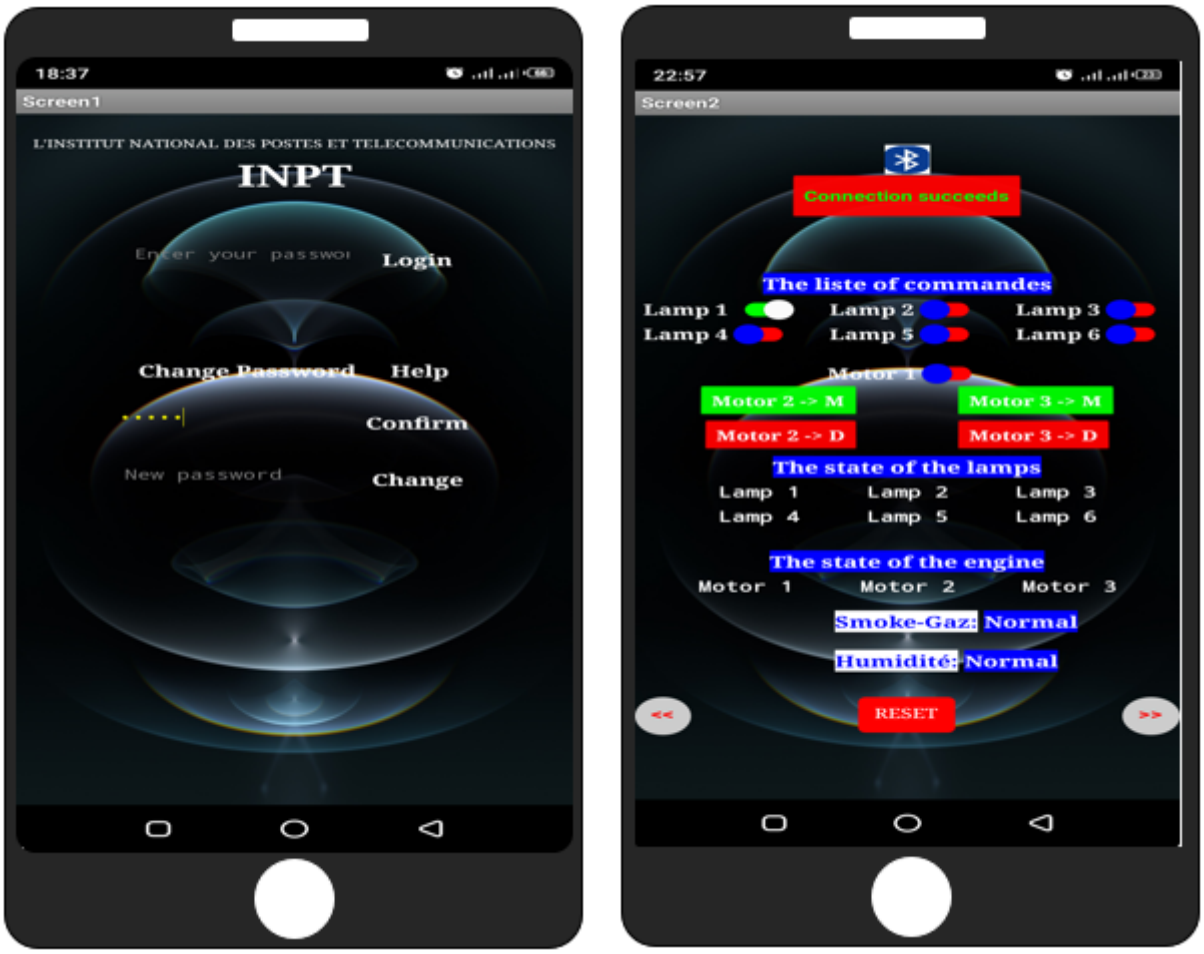


FIGURE 0.16: l'interface utilisateur avec Mit App Inventor

c- Réalisation finale du prototype :

Notre prototype de maison connectée avec la technologie Bluetooth se compose d'une variété d'éléments soigneusement intégrés pour créer un système complet et fonctionnel. Nous avons utilisé une carte Arduino Mega comme base, et avons ajouté des fonctionnalités telles que des LED pour simuler le comportement des lampes, des moteurs pour simuler le système de refroidissement et le contrôle du rideau de la fenêtre, ainsi que des capteurs de température, d'humidité et de gaz pour surveiller l'environnement. Nous avons également inclus un afficheur LCD et une LDR pour réguler l'éclairage en fonction de la lumière ambiante (jour/nuit), ainsi qu'une application Android créée avec l'outil MIT App Inventor pour permettre un contrôle à distance facile et pratique du système.

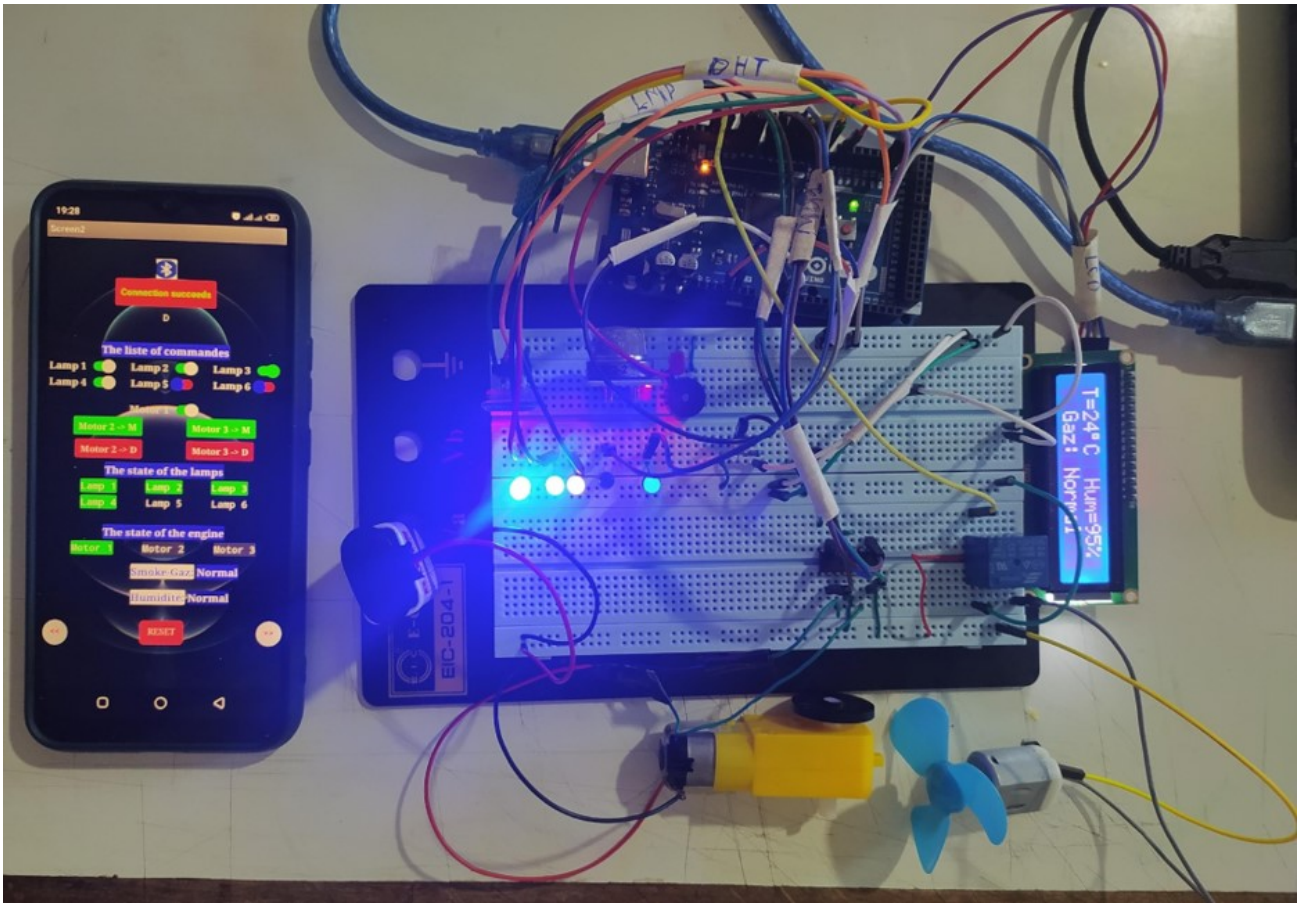


FIGURE 0.17: Réalisation finale du prototype

2) Deuxième solution utilisant le WiFi

a- Circuits électriques : Schéma, câblage et branchement

Notre prototype de maison connectée, équipé de la technologie WiFi, est composé d'une carte Arduino Mega et d'une carte ESP8266 pour le système. L'interface utilisateur est réalisée grâce à Node-RED, qui est relié à un broker Mosquitto installé sur un Raspberry Pi 4. Le prototype comprend également des LED pour simuler le comportement des lampes, un moteur pour simuler le système de refroidissement, un autre moteur pour simuler le contrôle du rideau de la fenêtre, ainsi que des capteurs de température, d'humidité et de détection de gaz. Nous avons également inclus un afficheur LCD et une LDR pour commander une lampe en fonction de la présence de lumière (jour/nuit).

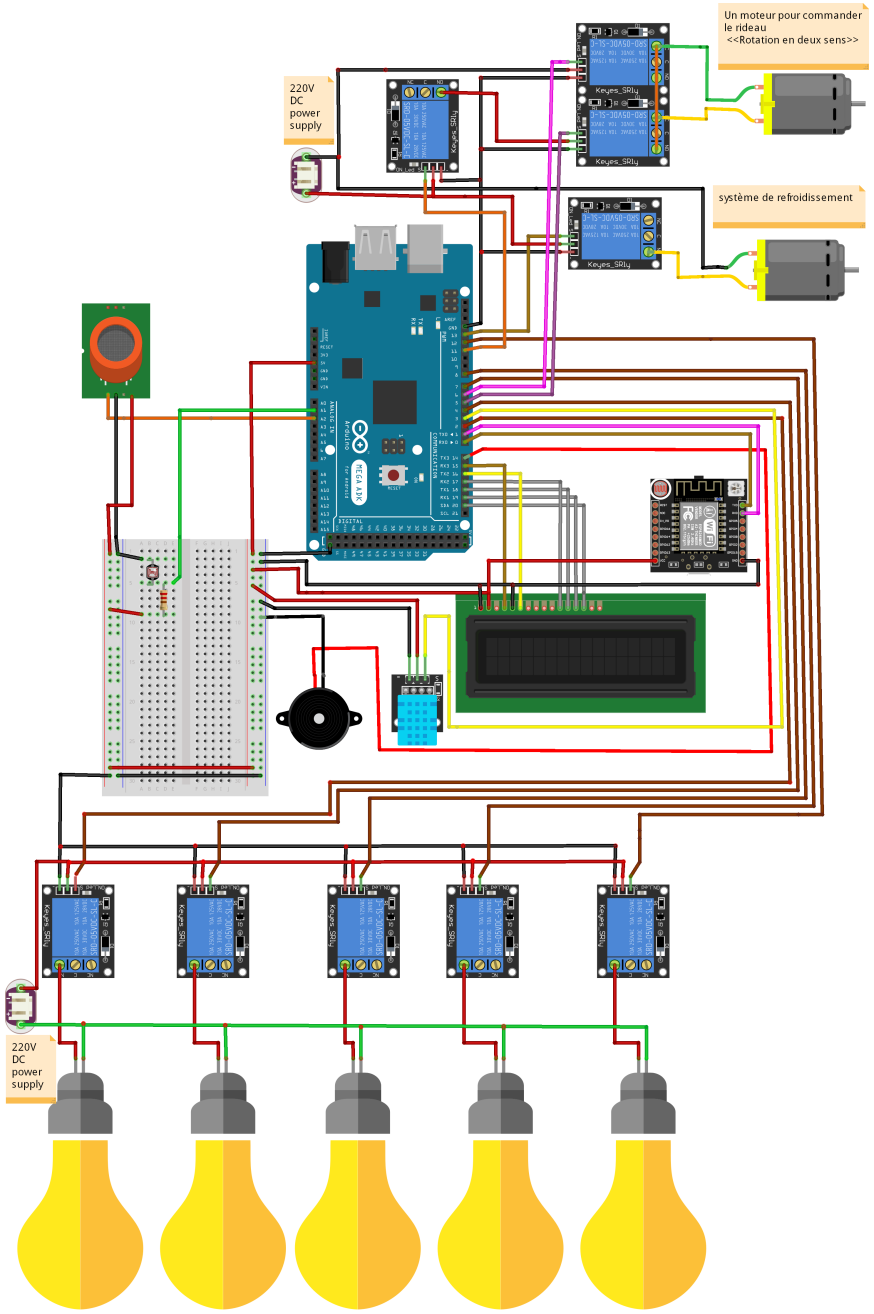
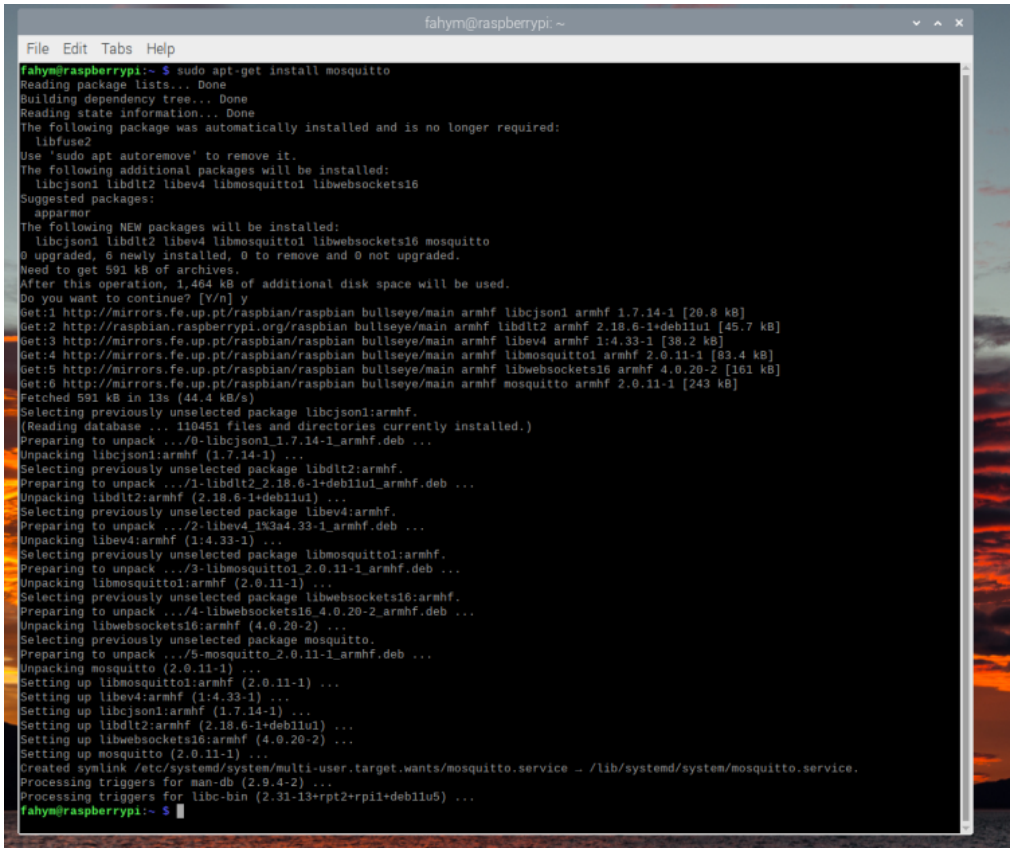


FIGURE 0.18: Circuits électriques

b- Installation et démarrage du broker Mosquitto sur Raspberry Pi 4 :

sudo apt-get install -y mosquitto : installera le broker Mosquitto sur votre Raspberry Pi. Cette commande installera également toutes les dépendances nécessaires.
systemctl status mosquitto : vérifiera le statut du service Mosquitto sur votre Raspberry Pi. Cette commande vous montrera si le service Mosquitto est en cours d'exécution ou non, et s'il y a des erreurs à corriger.



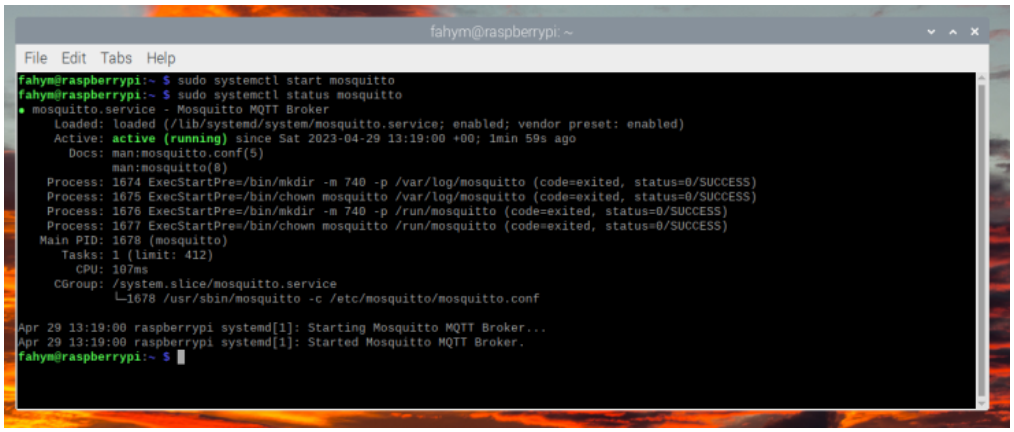


FIGURE 0.19: Installation et démarrage du broker Mosquitto sur Raspberry Pi 4

c- Conception de l'interface utilisateur avec Node-RED :

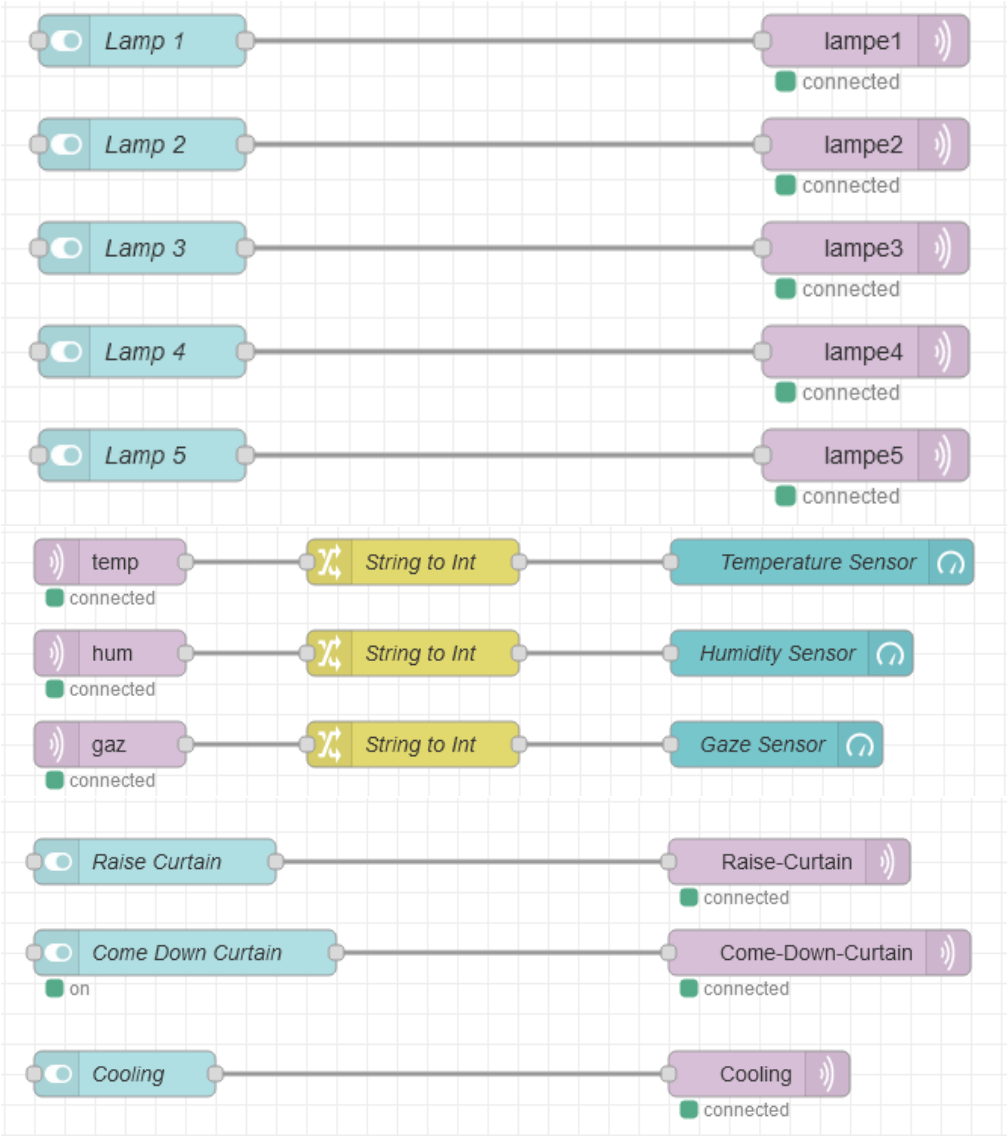


FIGURE 0.20: L'interface utilisateur "Back end"

d- Réalisation finale du prototype :

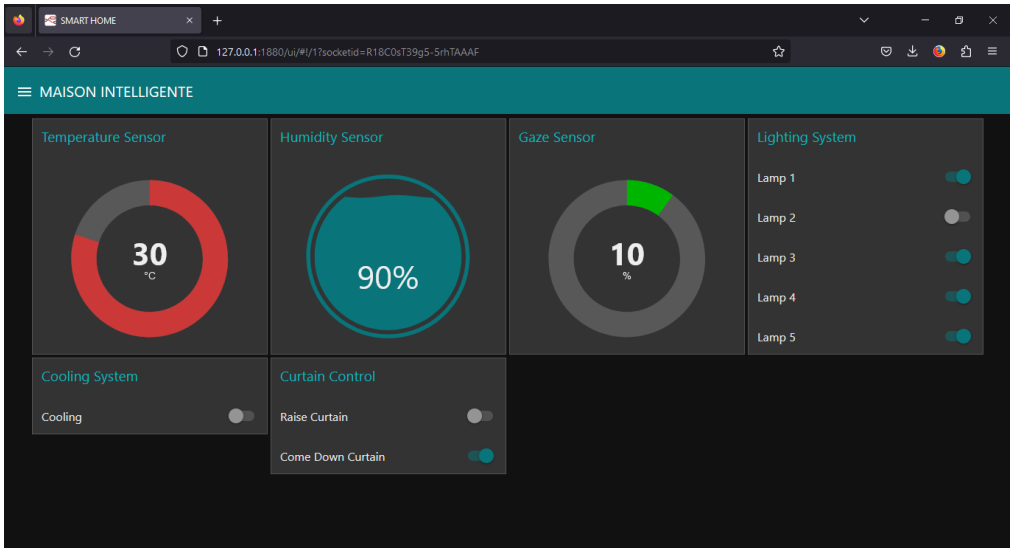


FIGURE 0.21: L'interface utilisateur "Front end"

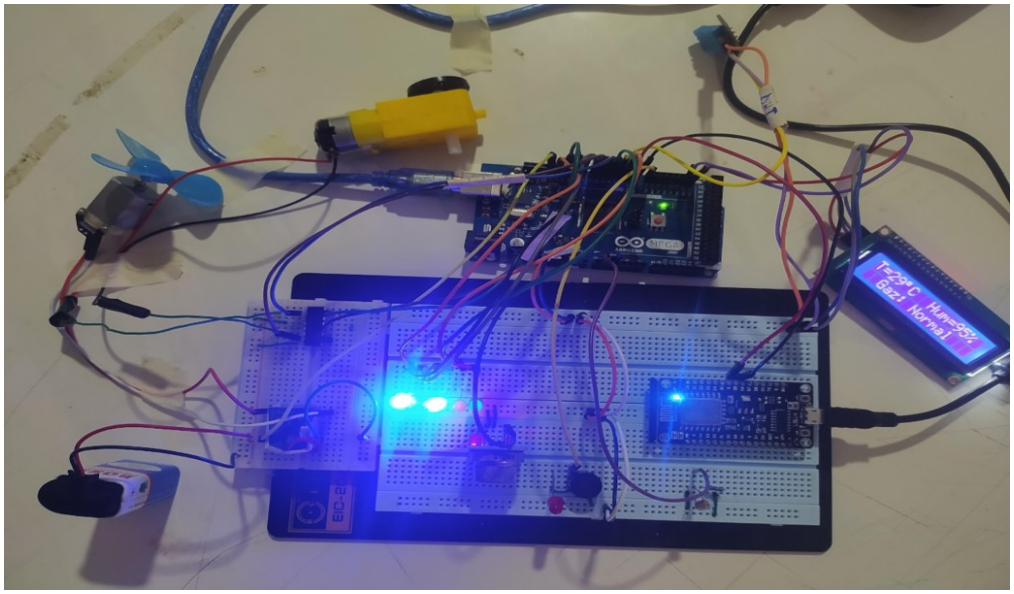


FIGURE 0.22: Réalisation finale du prototype

Conclusion générale

Avec le développement continu des technologies de communication, des ordinateurs, des logiciels et des systèmes intelligents, les maisons connectées sont désormais une réalité concrète plutôt qu'une simple utopie. Cette évolution a considérablement amélioré le confort des gens dans leur maison et a permis l'offre de plusieurs nouveaux services tels que la sécurité et la protection des personnes, la surveillance accrue et l'amélioration du confort. Cette problématique a fait l'objet de nombreux travaux de recherche et notre projet de fin d'études intitulé "Contrôle et suivi d'une maison intelligente via Internet" nous a permis de jauger notre capacité à travailler en groupe, de mettre en valeur nos connaissances préalables et d'en acquérir de nouvelles. De plus, étant un sujet très récent et en constante évolution, cela nous permettra de poursuivre notre apprentissage à long terme.

Notre mémoire illustre le fonctionnement d'un système domotique basé sur Arduino utilisant deux technologies différentes (Bluetooth et WiFi) afin de surveiller et de contrôler les appareils domestiques. Nous avons développé une application Android en utilisant la plateforme Mit App Inventor et un site web à l'aide de l'outil Node-red qui est connecté à

un broker Mosquitto installé sur une Raspberry Pi 4. Malgré la complexité du sujet, nous avons pu atteindre les trois objectifs principaux de ce projet : la commande via Bluetooth et WiFi, le contrôle de l'état des capteurs en temps réel et l'interface utilisateur.

Ce projet nous a permis de découvrir un nouveau domaine passionnant et innovant appelé domotique, qui nous a apporté énormément de connaissances. Nous pouvons dire que la période de réalisation de ce projet était une période éducative qui nous a permis d'explorer de nombreux domaines tels que l'internet des objets et le développement d'applications Android. Cependant, nous avons également rencontré plusieurs difficultés en raison de la nouveauté et de la complexité du sujet, ainsi que du respect des délais de réalisation.

ANNEXES

Programme arduino avec la communication bluetooth hc-06

```
#include<SoftwareSerial.h>
#include<DHT.h>
#include <Wire.h>
//#include<LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
//SoftwareSerial mySerial (1,0);
//-------------------------------------Les variabeles
int const LmpPin1=4,LmpPin2=12,LmpPin3=7,LmpPin4=8 ,LmpPin5=2,LmpPin6;
int const MotPin1=13, Mot1Pin1=5,Mot1Pin2=6, Mot2Pin1=9, Mot2Pin2=10 ;
int const BuzPin=14 ,GazPin= A2;
bool HautM1= true , BasM1= false ,HautM2= true , BasM2= false ;
DHT dht(3,DHT11);
//LiquidCrystal lcd(15,16,17,18,19,20);
LiquidCrystal_I2C lcd(0x27, 20, 4);
int Temperature,TempDht12,HumidDht12;
char c ;
String ContMOT3;
bool ContMOT2 ,ContMOT1,ContLMP;
bool Alume = false ;
bool EtatNormal =true ,ContLigne =true ;
bool EtatNormal1 =true ,ContLigne1 =true ;
//-------------------------------------setup
void setup() {
    lcd.init();
    Serial.begin(9600);
    dht.begin();
    lcd.begin(20,4);
    int i;
    for(i=5;i<=14;i++){
        pinMode(i,OUTPUT);
        digitalWrite(i,LOW);
    }
    lcd.backlight();
    lcd.setCursor(5, 0);
    lcd.print("SESNUM");
    lcd.setCursor(3,1);
    lcd.print("SMART HOME");
    delay(1000);
    lcd.clear();
}
//--------------------loop
void loop() {
    RegTemperature();
    AutoEclerage();
    DetectionFumee();
    ContRideaux();
    ContrGarage();
    if(Serial.available()){
        char c = Serial.read();
        switch(c){
            //----Controle des lampes par bluetoothe
            case 'A':
            ContLMP = true;
            Serial.print("A");
            break ;
            case 'a':
```

```
                ContLMP = false ;
                Serial.print("a");
                break;
                case 'B':
                digitalWrite(LmpPin2 , HIGH);
                Serial.print("B");
                break ;
                case 'b':
                digitalWrite(LmpPin2 , LOW);
                Serial.print("b");
                break ;
                case 'C':
                digitalWrite(LmpPin3 , HIGH);
                Serial.print("C");
                break ;
                case 'c':
                digitalWrite(LmpPin3 , LOW);
                Serial.print("c");
                break ;
                case 'D':
                digitalWrite(LmpPin1 , HIGH);
                Serial.print("D");
                break ;
                case 'd':
                digitalWrite(LmpPin1 , LOW);
                Serial.print("d");
                break ;
                case 'E':
                digitalWrite(LmpPin5 , HIGH);
                Serial.print("E");
                break ;
                case 'e':
                digitalWrite(LmpPin5 , LOW);
                Serial.print("e");
                break ;
                case 'F':
                digitalWrite(LmpPin6 , HIGH);
                Serial.print("F");
                break ;
                case 'f':
                digitalWrite(LmpPin6 , LOW);
                Serial.print("f");
                break ;
                //---Controle de moteur de refroidissement par bluetooth
                case 'G':
                ContMOT1 = true ;
                Serial.print("G");
                break ;
                case 'g':
                ContMOT1 = false ;
                Serial.print("g");
                break ;
                //------Controle du Rideaux par bluetooth
                case 'H':
                ContMOT2 = true ;
                break ;
                case 'h':
                ContMOT2 = false ;
                break ;
                //------------Control du Garage par bluetooth
                case 'I':
                ContMOT3 = "MONT" ;
                break ;
                case 'i':
```

```
                ContMOT3 = "DESC" ;
                break ;
                //-------------RESET par bluetooth
                case 'J':
                int j ;
                for(j=0 ;j<=15;j++){
                    digitalWrite(j,LOW);
                    Serial.print("J");
                }
                Serial.print("J");
                delay(30000);
                break ;
            }
        }
        delay(100);
        lcd.clear();
    }

    void RegTemperature(){
        /*float Valeur = analogRead(A0);
        float Vout = Valeur *5.0/1023*1000;
        int Temp = Vout/10;*/
        int Temp = dht.readTemperature();
        lcd.backlight();
        lcd.setCursor(0, 0);
        lcd.print("T=");
        lcd.print(Temp);
        lcd.write(223);
        lcd.print("C");
        if((Temp<30 && ContMOT1 == true) ||(Temp>=30 && ContMOT1 == false) ||(Temp
   >=30 &&
        ContMOT1 == true))
        digitalWrite(MotPin1,HIGH);
        if(Temp<30 && ContMOT1 == false)
        digitalWrite(MotPin1,LOW);
    }
    //----------Eclerage par LDR
    void AutoEclerage(){
        float Eclr = analogRead(A1);
        //Serial.println(Eclr);
        if ((Eclr < 600 && ContLMP == true)||(Eclr >= 600 && ContLMP == false)||(Eclr
    >= 600 &&
        ContLMP == true) ){
            digitalWrite(LmpPin4,HIGH);
        }
        if( Eclr < 600 && ContLMP == false){
            digitalWrite(LmpPin4,LOW);
        }
    }

    void DetectionFumee(){
        lcd.setCursor(0,1);
        int Reff = 600 ;
        int analogSensor = analogRead(A3);
        lcd.setCursor(2, 1);
        lcd.print("Gaz:");
        //lcd.print("analogSensor");
        //lcd.println(analogSensor);
        if(analogSensor > Reff ){
            tone(BuzPin,100,200);
            EtatNormal = false ;
            lcd.print(" Danger");
        }
        else{
```

```
            noTone(BuzPin);
            EtatNormal = true ;
            lcd.print(" Normal");
        }
        if(EtatNormal == false && ContLigne == false ){
            Serial.print("k");
            ContLigne = true ;
        }
        if(EtatNormal == true && ContLigne == true){
            Serial.print("j");
            ContLigne = false ;
        }
        delay(250);
    }
    //----------------------Controle du Rideaux avec DHT12
    void ContRideaux(){
        //HautM1= true , BasM1= false ContMOT2 = "MONT" ContMOT2 = "DESC"
        TempDht12 = dht.readTemperature();
        HumidDht12 = dht.readHumidity();
        lcd.setCursor(8, 0);
        lcd.print("Hum=");
        lcd.print(HumidDht12);
        lcd.print("\%");

        if(((ContMOT2 == true && HumidDht12 < 80 )|| (ContMOT2 == true && HumidDht12
    >=80 ))and
        BasM1== true and HautM1== false){
            // monterCommande MotPin2Start=9, MotPin2Rotat=10
            //Mot1Pin1=5,Mot1Pin2=6, Mot2Pin1=9,Mot2Pin2=10
            digitalWrite(Mot1Pin1,HIGH);
            digitalWrite(Mot1Pin2,LOW);
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,HIGH);
            Serial.print("H");
            delay(3000);
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,LOW);
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,LOW);
            Serial.print("h");
            HautM1= true;
            BasM1= false ;
        }
        if(((ContMOT2 == false && HumidDht12 <80 )||(ContMOT2 == false &&
        HumidDht12 >=80 ) )and HautM1 == true and BasM1 == false){
            // DescandeCommande
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,HIGH);
            digitalWrite(Mot2Pin1,HIGH);
            digitalWrite(Mot2Pin2,LOW);
            Serial.print("H");
            delay(3000);
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,LOW);
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,LOW);
            Serial.print("h");
            HautM1 = false;
            BasM1 = true ;
        }
        if(HumidDht12 >= 90){
            EtatNormal1 = false ;
        }
        else{
```

```
            EtatNormal1 = true ;
        }
        if(EtatNormal1 == false and ContLigne1 == false ){
            Serial.print("l");
            ContLigne1 = true ;
        }
        if(EtatNormal1 == true and ContLigne1 == true){
            Serial.print("m");
            ContLigne1 = false ;
        }
        delay(250);
    }
    //------------------------------Controle du Garage
    void ContrGarage(){
        if(ContMOT3 == "MONT" and BasM2== true and HautM2== false){
            // monterCommande
            digitalWrite(Mot2Pin1,HIGH);
            digitalWrite(Mot2Pin2,LOW);
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,HIGH);
            Serial.print("I");
            delay(3000);
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,LOW);
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,LOW);
            Serial.print("i");
            HautM2= true;
            BasM2= false ;
        }
        if(ContMOT3 == "DESC" and HautM2 == true and BasM2 == false){

            // DescandeCommande
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,HIGH);
            digitalWrite(Mot1Pin1,HIGH);
            digitalWrite(Mot1Pin2,LOW);
            Serial.print("I");
            delay(3000);
            digitalWrite(Mot2Pin1,LOW);
            digitalWrite(Mot2Pin2,LOW);
            digitalWrite(Mot1Pin1,LOW);
            digitalWrite(Mot1Pin2,LOW);
            Serial.print("i");
            HautM2 = false;
            BasM2 = true ;
        }
    }
```

Programme arduino avec la communication WiFi ESP8266

```
        #include <ESP8266WiFi.h>
        #include <PubSubClient.h>

        //Relays for switching appliances
        #define Relay1            D6
        #define Relay2            D2
        #define Relay3            D1
        #define Relay4            D5
        #define Relay5            D3
        #define Relay6            D4
```

```
#define Relay7             D7
#define MSG_BUFFER_SIZE  (50)

#define sub1 "lampe1"
#define sub2 "lampe2"
#define sub3 "lampe3"
#define sub4 "lampe4"

#define sub5 "lampe5"
#define sub6 "Raise-Curtain"
#define sub7 "Come-Down-Curtain"
#define sub8 "Cooling"



const char* ssid = "HTL-WS";
const char* password = "1234567890";
const char* mqtt_server = "XXX.XXX.XXX.XXX";  // Local IP address of Raspberry Pi

const char* username = "MQTT_USERNAME";
const char* pass = "MQTT_PASSWORD";


char str_hum_data[10];
char str_temp_data[10];
char str_gaz_data[10];
char msg[MSG_BUFFER_SIZE];

int value = 0;
unsigned long lastMsg = 0;

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
    pinMode(Relay1, OUTPUT);
    pinMode(Relay2, OUTPUT);
    pinMode(Relay3, OUTPUT);
    pinMode(Relay4, OUTPUT);

    Serial.begin(115200);
    WiFi_setup();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

void loop() {

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    unsigned long now = millis();
    if (now - lastMsg > 2000) {
        float hum_data = random(-50,50);
        Serial.println(hum_data);
        /* 4 is mininum width, 2 is precision; float value is copied onto str_sensor*/
        dtostrf(hum_data, 4, 2, str_hum_data);
        float temp_data = random(0,100); // or dht.readTemperature(true) for Fahrenheit
        dtostrf(temp_data, 4, 2, str_temp_data);
```

```
                lastMsg = now;
                Serial.print("Publish message: ");
                Serial.print("Temperature - "); Serial.println(str_temp_data);
                client.publish("temp", str_temp_data);
                Serial.print("Humidity - "); Serial.println(str_hum_data);
                client.publish("hum", str_hum_data);
                Serial.print("Gaze - "); Serial.println(str_gaz_data);
                client.publish("gaz", str_gaz_data);
            }
        }


        //_____________________________________________WiFi Conection_______________________________
        void WiFi_setup() {
            delay(10);
            Serial.print("Connecting to ");
            Serial.println(ssid);

            WiFi.mode(WIFI_STA);
            WiFi.begin(ssid, password);

            while (WiFi.status() != WL_CONNECTED) {
                delay(500);
                Serial.print(".");
            }
            randomSeed(micros());
            Serial.println("");
            Serial.println("WiFi connected");
            Serial.println("IP address: ");
            Serial.println(WiFi.localIP());
        }

        void callback(char* topic, byte* payload, unsigned int length) {
            Serial.print("Message arrived [");
            Serial.print(topic);
            Serial.print("] ");

            if (strstr(topic, sub1)){
                for (int i = 0; i < length; i++) {
                    Serial.print((char)payload[i]);
                }
                Serial.println();
                // Switch on the LED if an 1 was received as first character
                if ((char)payload[0] == '1') {
                    digitalWrite(Relay1, HIGH);   // Turn the LED on (Note that LOW is the voltage level
                    // but actually the LED is on; this is because
                    // it is active low on the ESP-01)
                } else {
                    digitalWrite(Relay1, LOW);  // Turn the LED off by making the voltage HIGH
                }
            }

            else if ( strstr(topic, sub2)){
                for (int i = 0; i < length; i++) {
                    Serial.print((char)payload[i]);
                }
                Serial.println();
                // Switch on the LED if an 1 was received as first character
                if ((char)payload[0] == '1') {
                    digitalWrite(Relay2, HIGH);   // Turn the LED on (Note that LOW is the voltage level
```

```
                    // but actually the LED is on; this is because
                    // it is active low on the ESP-01)
                } else {
                    digitalWrite(Relay2, LOW);  // Turn the LED off by making the voltage HIGH
                }
            }
            else if ( strstr(topic, sub3)){
                for (int i = 0; i < length; i++) {
                    Serial.print((char)payload[i]);
                }
                Serial.println();
                // Switch on the LED if an 1 was received as first character
                if ((char)payload[0] == '1') {
                    digitalWrite(Relay3, HIGH);   // Turn the LED on (Note that LOW is the voltage level
                    // but actually the LED is on; this is because
                    // it is active low on the ESP-01)
                } else {
                    digitalWrite(Relay3, LOW);  // Turn the LED off by making the voltage HIGH
                }
            }
            else if ( strstr(topic, sub4)){
                for (int i = 0; i < length; i++) {
                    Serial.print((char)payload[i]);
                }
                Serial.println();
                // Switch on the LED if an 1 was received as first character
                if ((char)payload[0] == '1') {
                    digitalWrite(Relay4, HIGH);   // Turn the LED on (Note that LOW is the voltage level
                    // but actually the LED is on; this is because
                    // it is active low on the ESP-01)
                } else {
                    digitalWrite(Relay4, LOW);  // Turn the LED off by making the voltage HIGH
                }
            }

            else{
                Serial.println("unsubscribed topic");
            }
        }

        void reconnect() {
            // Loop until we're reconnected
            while (!client.connected()) {
                Serial.print("Attempting MQTT connection...");
                // Create a random client ID
                String clientId = "ESP8266Client-";
                clientId += String(random(0xffff), HEX);
                // Attempt to connect
                if (client.connect(clientId.c_str(), username, pass) ) {
                    Serial.println("connected");
                    // Once connected, publish an announcement...
                    // ... and resubscribe
                    client.subscribe(sub1);
                    client.subscribe(sub2);
                    client.subscribe(sub3);
                    client.subscribe(sub4);
                } else {
                    Serial.print("failed, rc=");
                    Serial.print(client.state());
```

```
191                     Serial.println(" try again in 5 seconds");
192                     // Wait 5 seconds before retrying
193                     delay(5000);
194                 }
195             }
196         }
197
198
199
```

Bibliography

Livre : "Internet of Things for Architects : Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security" de Perry Lea. Article : "Comparison of LPWAN Technologies for IoT Applications" de Muhammad Ali Nawaz, Muhammad Adnan Khan et HyungSeok Kim.
Livre : "Smart Home Automation with Linux and Raspberry Pi" de Steven Goodwin. Article : "The Role of IoT in Smart Homes and Buildings" de Ahmed Banafa.
"Building Wireless Sensor Networks : with ZigBee, XBee, Arduino, and Processing" de Robert Faludi "Internet of Things with ESP8266" de Marco Schwartz "Raspberry Pi Cookbook : Software and Hardware Problems and Solutions" de Simon Monk "Learning Node-RED : Creating Lightweight IoT Applications" de James Hardie "App Inventor 2 : de David Wolber, Hal Abelson, Ellen Spertus, Liz Looney
Site web officiel d'Arduino : https ://www.arduino.cc/
Site web officiel de Raspberry Pi : https ://www.raspberrypi.org/
Site web officiel d'ESP8266 : https ://www.espressif.com/en/products/socs/esp8266
Documentation officielle de Mosquitto : https ://mosquitto.org/documentation/ Site web officiel de Node-RED : https ://nodered.org/ Site web officiel de MIT App Inventor : https ://appinventor.mit.edu/