



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Monastir

Institut Supérieur d'Informatique et de Mathématiques de Monastir

Département de Technologie



N° d'ordre : 4

Mémoire de Projet de Fin d'Etudes

Présenté en vue de l'obtention du

**Diplôme National de Licence en Technologie de
l'Information et de la Communication**

Spécialité :

Réseaux et IOT

Par

Chaima ZBIDI

**Conception et implémentation d'un système
intelligent de détection de gaz et des flammes**

Soutenu le 24 juin 2022 devant le jury composé de :

M. Abdessalem BEN ABDELALI

Mme. Marwa CHOUCHENE

M. Sadok BAZINE

Président

Rapporteur

Encadrant Pédagogique

الْحَمْدُ لِلَّهِ الَّذِي بِنِعْمَتِهِ تَمَّ الصَّالِحَاتُ وَتَنْقَضِي الْحَاجَاتُ، وَتَأْتِي الْحَيْرَاتُ وَتَزُولُ الْعَقَبَاتُ،
الْحَمْدُ لِلَّهِ دَائِمًا وَأَبَدًا،

Résumé

Ce travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention de la licence en Technologies de l'information et de la communication.

L'objectif de ce travail est de réaliser et de mettre en place un système intelligent de détection des flammes et des fuites de gaz en utilisant l'intelligence artificielle.

Notre système est à base des cartes Raspberry pi, nodeMCU ESP32 et un ensemble des capteurs et des actionneurs. Il permet de contrôler la concentration de gaz naturel et la détection de flamme d'une manière intelligente avec une solution Web et mobile.

Mots clés : Mosquitto, MQTT, Machine Learning, Détection de gaz, Détection de flammes, notification, Raspberry Pi, nodeMCU ESP32, Classification d'images, Firebase.

Abstract

This work is part of the end-of-study project for obtaining the license in Information and Communication Technology.

The objective of this work is to realize and implement an intelligent system for detecting flames and gas leaks using artificial intelligence.

Our system is based on a Raspberry pi card, ESP32 nodeMCU and a set of sensors and actuators.

It allows you to control the concentration of natural gas and the flame prediction in an intelligent way with a web and mobile solution.

Keywords : Mosquitto, MQTT, Machine Learning, gas detection, flame detection, notification, Raspberry Pi, nodeMCU esp32, image classification, Firebase

Dédicaces

C'est à travers ces quelques notes que je dédie ce travail à :

Mon père 'Abdel Waheb', Ma mère 'Leila'

Aucune dédicace ne saurait exprimer la fierté d'être leur fille. Vous avez guidé mes premiers pas, Vous avez toujours su me soutenir et m'épauler, parfois à vos dépends, avec amour et tendresse.

Ma plus grande reconnaissance n'égalera jamais l'ampleur de vos sacrifices. Merci pour les principes et les valeurs que vous m'avez transmis. Merci maman, merci mon père pour hier, pour aujourd'hui et pour demain.

*J'espère que j'ai pu aujourd'hui vous rendre plus fier de moi en soutenant cette mémoire
Que ce travail soit le témoignage de mon profond et éternel amour.*

Mon frère Alaa et sa femme Abir

Pour les moments de bonheur partagés. Pour m'avoir encouragé dans les moments de doute et soutenu dans mes projets professionnels et personnels.

Mes sœurs Jaweher, Abir et Mariem

Sans leurs conseils et leur amour je serai perdue, aucune dédicace ne saurait exprimer assez profondément ce que je ressens envers vous.

Mes gendres Sedki et Khaled et ses enfants

Pour leur soutien et leur amour

A tous mes amis

Tasnim, Majdi, Aziz, Khalfaoui, Oumaima... pour la merveilleuse ambiance qui caractérise notre amitié.

Qu'ils soient heureux sur les plans personnels, professionnels et sociaux.

A tous ceux qui ont été là pour moi quand j'ai besoin de leur soutien mais surtout à tous ceux qui ont été, qui sont et qui seront inchallah présents pour moi-même quand je n'aurais pas besoin d'eux.



Z. Chaima

Remerciement

*J*e remercie avant tout Dieu qui m'a donné la force et la volonté d'accomplir ce modeste travail.

*E*n second lieu, aucun remerciement ne saurait suffire pour traduire ma reconnaissance et ma profond respect à mon encadrant Monsieur **Sadok BAZINE** pour ses précieux conseils, son soutien et son encouragement continu. Je tiens à lui exprimer ma grande admiration pour ses compétences scientifique et ses qualités humaine.

*M*es vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ce Projet en acceptant d'examiner ce travail et de l'enrichir par leurs propositions.

*J*insi, je tiens à remercier vivement tous les enseignants et employés du département de technologie à l'**ISIMM** pour tous leurs aides et services qu'ils m'ont octroyé tout le long de mon parcours universitaire.

*E*nfin, je tiens également à remercier toutes les personnes qui ont participées de près ou de loin à la réalisation de ce travail

Table des matières

Introduction générale.....	13
1. Chapitre 1 : Contexte général du projet.....	15
1.1 Introduction	15
1.2 Présentation du projet.....	15
1.2.1 Cadre général.....	15
1.2.2 Problématique.....	15
1.3 Etude de l'existant	16
1.3.1 Solutions existantes	16
1.3.2 Critique de l'existant	16
1.4 Solution proposée	16
1.5 Analyse et spécification des besoins	18
1.5.1 Etude préliminaire des solutions proposées.....	18
1.5.2 Spécification de la solution choisie :	21
1.6 Planification du projet	24
1.7 Conclusion.....	25
2. Chapitre 2 : Conception détaillée du système de protection	26
2.1 Introduction	26
2.2 Principe de fonctionnement.....	26
2.3 Environnement matériel du système	27
2.7 Etude conceptuelle	35
2.7.1 Diagramme de cas d'utilisation	35
2.7.2 Description comportementale du système.....	39
2.8 Conclusion.....	43
3. Chapitre 3 : Implémentation et tests de validation	44
3.1 Introduction	44
3.2 Architecture globale de solution proposée	44
3.2.1 Architecture plate de la solution proposée	45
3.2.2 Architecture par couche de la solution proposée.....	45
3.3 Environnement logiciel et outils de programmation	46
3.4 Tests des capteurs à utiliser.....	49
3.3.1 Mise en marche du capteur de gaz MQ9	49
3.3.2 Mise en marche du capteur de fumée MQ2.....	51
3.3.3 Mise en marche du capteur de flamme.....	53
3.3.4 Mise en marche du capteur DHT22.....	53
3.3.5 Mise en marche de l'afficheur LCD	54

3.5	Implémentation de l'unité d'émission/réception	54
3.5.1	Esp32 en tant que « publisher ».....	54
3.5.2	Configuration de la Raspberry PI en tant que « subscriber»	56
3.6	Unité de détection d'incendie par vision artificielle.....	59
3.5.1	Transfer Learning « apprentissage par transfert »	60
3.5.2	Classifications des images en utilisant « Tensorflow ».....	60
3.5.3	Intégration du modèle de reconnaissance de flamme dans Node-RED.....	61
3.7	Unité de protection contre les incendies.....	62
3.6.1	Système de ventilation.....	63
3.6.2	Dispositif de coupure de l'arrivée de gaz	64
3.8	Unité d'alerte	64
3.9	Développement de la plateforme Web	65
3.9.1	Mode citoyen.....	65
3.9.2	Mode Protection Civile.....	69
3.10	Développement de l'application Android	70
3.10.1	Partie graphique.....	71
3.11	Partie Traitement	75
3.11.1	Mode citoyen.....	75
3.10.2	Mode protection civile.....	77
3.12	Test de fonctionnement du système	78
3.11.1	les scénarios de détection de gaz et de fumée	78
3.11.2	les scénarios de détection de flamme	79
3.13	Réalisation de la carte d'interfaçage.....	79
3.15	Conclusion.....	81
	Conclusion générale et perspectives.....	82

Liste des figures

Figure 1.1: Schéma synoptique de la solution envisagée	17
Figure 1.2: Schéma synoptique de la première solution.....	18
Figure 1.3: Modulation radio.....	19
Figure 1.4 Topologie de LoRa	20
Figure 1.5 : Diagramme de pieuvre	21
Figure 1.6 : Diagramme d'analyse fonctionnelle descendante (SADT)	22
Figure 1.7:Diagramme niveau A-0 du système de détection de fuite du gaz et de flamme	22
Figure 1.8: Diagramme de Gantt de notre projet.....	25
Figure 2.1 Description fonctionnelle du système	26
Figure 2.2: Principaux modules de la partie matérielle du système	27
Figure 2.3: Carte Raspberry PI 3 modèle B	28
Figure 2.4: les composants de la carte ESP32	29
Figure 2.5:camera Raspberry pi v2	29
Figure 2.6: les différentes broches de relais	30
Figure 2.7 : Relais à 8canaux	30
Figure 2.8:Relais à 4 canaux	30
Figure 2.9:Relais à 2 canaux	30
Figure 2.10:capteur DHT22	31
Figure 2.11: Capteur MQ-9	31
Figure 2.12:Capteur de fumée MQ2.....	32
Figure 2.13 : capteur de flamme.....	33
Figure 2.14 : Afficheur LCD I2C 16x2	34
Figure 2.15 : Buzzer	34
Figure 2.16 : électrovanne	34
Figure 2.17: ventilateur 12V	35
Figure 2.18 : Diagramme de cas d'utilisation global	36
Figure 2.19 : Diagramme des cas d'utilisation raffiné pour le citoyen	37
Figure 2.20: Diagramme des cas d'utilisation Raffiné pour un agent de protection civil	38
Figure 2.21: Diagramme de séquence pour l'utilisateur citoyen « authentification »	40
Figure 2.22: Diagramme de séquence d'un agent de protection civil « authentification »	41
Figure 2.23: Diagramme de séquence « consulter les données des capteurs ».....	42
Figure 2.24: Diagramme de séquence « consulter les alerte et l'image de l'incendie ».....	43
Figure 3.1: Mode de fonctionnement global du système.....	44
Figure 3.2 : Architecture plate de la solution proposée.....	45
Figure 3.3: Architecture par couche de la solution.....	45
Figure 3.4 : Serveur MQTT.....	47
Figure 3.5: diagramme actuel de la plateforme TensorFlow. [5]	47
Figure 3.6:Capteur de gaz MQ9 avec la carte ESP32	49
Figure 3.7:Courbe d'étalonnage de capteur MQ9.....	50
Figure 3.8 : Envoi de valeur de gaz vers firebase.....	50
Figure 3.9: Capteur MQ2 avec la carte ESP32.....	51
Figure 3.10: Courbe d'étalonnage du capteur MQ2.....	52
Figure 3.11: L'envoie de valeur de fumée vers firebase	52
Figure 3.12: Branchement du capteur de flamme avec ESP32	53
Figure 3.13: Branchement de capteur DHT22 avec la carte ESP32.....	53
Figure 3.14 : le résultat sur le moniteur série	53
Figure 3.15: Branchement d'afficheur LCD avec la carte ESP32 via l'adaptateur I2C	54
Figure 3.16 : Bibliothèque MQTT	54
Figure 3.17: gestionnaire de bibliothèque Arduino IDE	55

Figure 3.18: Bibliothèque Wifi	55
Figure 3.19 : Formatage de la carte microSD.....	56
Figure 3.20 : Raspberry PI Imager	56
Figure 3.21 : Rendu de commande Mosquitto -v	58
Figure 3.22 : Diagramme de flot des données.....	59
Figure 3.23 : Configuration de node MQTT	59
Figure 3.24 : Branchement de Camera avec la carte Raspberry Pi	59
Figure 3.25 : Type de projet dans l'outil "Teachable machine"	60
Figure 3.26 : Exemple de reconnaissance d'une flamme.....	61
Figure 3.27 : Modéle Tensorflow dans Node-Red	62
Figure 3.28 : Conversion d'image à base64 et l'envoi vers le Cloud	62
Figure 3.29 : Format d'image dans le Cloud Firebase	62
Figure 3.30 : Branchement de ventilateur avec la carte Raspberry Pi.....	63
Figure 3.31: L'ouverture et la fermeture de ventilateur	63
Figure 3.32: système de coupure de gaz.....	64
Figure 3.33: Notification d'alerte de l'application mobile	65
Figure 3.34 : SMS d'alerte	65
Figure 3.35 : Interface principale	66
Figure 3.36: Interface d'authentification.....	67
Figure 3.37: Interface d'inscription	67
Figure 3.38: Tableau de bord de consultation pour un citoyen	68
Figure 3.39: L'interface d'authentification pour un agent de protection civile.....	69
Figure 3.40 : Tableau de bord d'un agent de protection civile	70
Figure 3.41 : Interface Principale d'application mobile.....	71
Figure 3.42: Interface d'authentification.....	72
Figure 3.43: Interface d'inscription	73
Figure 3.44 : Interface de consultation	74
Figure 3.45: Interface d'authentification pour un agent de protection civile.....	74
Figure 3.46 :Interface de consultation d'incendie.....	75
Figure 3.47: Authentication Firebase.....	76
Figure 3.48: Realtime Database Firebase.....	76
Figure 3.49: Notification d'alerte en cas de détection de gaz	77
Figure 3.50: Notification d'alerte en cas de détection de flamme	78
Figure 3.51 : Branchement des capteurs avec le nodeMCU.....	78
Figure 3.52 : Schéma de la carte sous EasyEDA	79
Figure 3.53 :Routage de carte électronique.....	80
Figure 3.54 : La vue 3D de la carte électronique.....	80
Figure 3.55: Implémentation du système	81
Figure 3.56 : Prototype finale du système de détection de gaz et de flamme.....	81
Figure 3.57 : DHT22 partie mesure de température.....	83
Figure 3.58 :DHT22 partie mesure d'humiditéFigure 3.59 : DHT22 partie mesure de température....	83
Figure 3.60: description des différents composants du Raspberry pi 3 modèle B	83
Figure 3.61 : les entrées et les sorties numérique de carte raspberry	83

Listes des tableaux

Tableau 1.1 :Les solutions existantes	16
Tableau 1.2 : Les avantages et les inconvénients du protocole de communication Wifi	19
Tableau 2.1: Comparaison entre DHT22 et DHT11.....	31
Tableau 2.2 : Les broches du capteur MQ-9	32
Tableau 2.3 : Les acteurs interagissent avec le système.....	36
Tableau 3.1 : Branchement de capteur MQ-9 avec ESP32	49
Tableau 3.2 : Branchement de capteur MQ-2 avec ESP32	51
Tableau 3.3: Installation et mise en marche d'un broker Mosquitto	57
Tableau 3.4 : Les etapes d'activation l'accès à distance.....	58
Tableau 3.5 : Brochage de ventilateur avec la carte Raspberry pi	63
Tableau 3.6: Tableau des boutons de l'interface principale.....	66

Liste des abréviations

GPL : Gaz de Pétrole Liquéfié

MQTT : Message Queuing Telemetry Transport

NOSQL : Not Only SQL.

HTTP : HyperText Transport Protocole

UML : Unified Modeling Language

PPM : Partie Par Million

HTML : HyperText Markup Langage

LORA : Long Range

T/H : Température/Humidité

JS : JavaScripte

IOT : Internet Of Things

IA : Intelligence Artificielle

Introduction générale

Le web et l'internet sont toujours en évolution, en partant du web d'information jusqu'à arriver au web des objets connectés, via plusieurs objets connectés grâce au développement technologique et la miniaturisation, qui permet un double aspect : être communicants et connecté en même temps afin de satisfaire les besoins de l'usager. Ces dernières années, l'IoT est développé dans des nombreux domaines de la vie, notamment les maisons, les villes, les voitures, l'agriculture, les bureaux et les lieux de travail intelligents. Les objets physiques du quotidien tels que les lumières, les serrures et les machines industrielles peuvent désormais faire partie de l'écosystème IoT.

La Computer Vision ou Vision par Ordinateur est une technique d'intelligence artificielle qui consiste à analyser puis interpréter une image ou une vidéo. En d'autres termes, ils cherchent à fournir à un ordinateur la capacité de voir.

Le méthane est le gaz le plus communément connu sous le nom de gaz naturel. La limite inférieure d'explosivité est de 5%. Cela signifie qu'un mélange avec juste assez de gaz naturel pour s'enflammer est composé de : 5% de méthane et 95% d'air, c'est-à-dire 5% de méthane, 19,95% d'oxygène et 75,05% d'azote.

Les systèmes de contrôle à distance font de plus en plus partie de notre quotidien et ont totalement améliorés notre environnement. Il est possible de contrôler et suivre à distance, depuis notre Smartphone ou PC, la température intérieur de notre maison et de prévenir, par exemple, les risques de fuite de gaz ou d'incendie... Cette révolution technologique rend possible la conception d'un système performant qui offre le confort, la sécurité, le gain de temps et le bien-être.

C'est dans ce contexte qu'est née l'idée de ce projet afin d'assurer la protection de nos habitats et de nos familles contre les fuites de gaz et les incendies tout en supervisant à distance aussi d'autre grandeurs comme la température l'humidité et en signalant toutes anomalies aux propriétaires et aux agents de la protection civile.

Pour répondre à ces objectifs ce présent manuscrit est scindé en 3 chapitres :

- Le premier chapitre intitulé « contexte générale de projet » est consacré en premier lieu à la présentation du cadre général du projet. Ensuite, nous allons détailler l'objectif du

projet en analysant les problématiques trouvés, les solutions existantes et on finit par la présentation des solutions possibles pour résoudre ces problèmes.

- Le deuxième chapitre intitulé « Analyse et spécifications des besoins » : contient une description détaillée du modèle conceptuel de notre système. En particulier, les besoins matériels du système sont présentés.
- Le troisième chapitre nommé « Implémentation et tests de validation » : est consacré à la présentation de notre système réel. Nous commençons ce chapitre par la présentation des outils logiciels que nous allons utiliser pour la réalisation pratique de notre projet. Par la suite, nous citons les différentes étapes par lesquelles nous avons passé au cours de la réalisation pratique de l'application en ce qui concerne la configuration et la préparation de nos matériels ainsi que l'installation des différents logiciels et bibliothèques. Une fois les tests sont validés nous réalisons le circuit imprimé de la carte électronique en question. Ensuite nous présenterons les différentes interfaces de l'application Web et mobile et finalement nous présenterons la phase de validation qui sera consacrée à la présentation des résultats du fonctionnement du système.

Ce manuscrit sera clôturé par une conclusion générale qui résume les points traités tout le long de ce travail et qui ouvre des perspectives qui peuvent être intéressantes.

Chapitre 1 : Contexte général du projet

1.1 Introduction

Dans ce premier chapitre, on va attaquer l'ensemble des problèmes qui nous ont poussés à penser à une solution, nous allons définir tout d'abord la problématique, puis nous allons analyser et critiquer les solutions existantes sur le marché, puis nous définissons notre solution. Et pour finir, nous mentionnons notre plan de travail durant notre stage.

1.2 Présentation du projet

1.2.1 Cadre général

Le présent travail s'inscrit dans le cadre d'un projet de fin d'études pour l'obtention du diplôme de licence en technologie de l'information et de communication à l'Institut Supérieur de l'Informatique et de Mathématique de Monastir (ISIMM) durant une période allant du 01 février 2022 au 30 juin. La mission qui nous a été confiée au sein de l'institut est de mettre en place un système intelligent de détection des flammes et des fuites de gaz.

1.2.2 Problématique

« Tunisie – Explosion et fuite de gaz dans l'usine du groupe chimique de Gabes »
« Tunisie: Décès de deux femmes après une fuite de gaz à leur domicile à Monastir »
« Tunisie : Une explosion au gaz a eu lieu, tôt ce matin, dans une maison du quartier d'Ibn Sina à Tunis, ce qui a entraîné la mort d'un homme sur place et d'un bébé par une projection de débris alors qu'il se trouvait dans sa poussette dans la rue. L'accident a également fait deux autres victimes grièvement blessés, dont un jeune homme et une femme **03-11-2021** »

Ces articles sont des échantillons de plusieurs cas que ce soit à domicile ou dans un milieu industriel dans la Tunisie et dans le monde entier. Les alarmes sont les solutions les plus répondues en Tunisie elles sont à base de détection de fuite est une alarme. L'absence de la prise en considération de la probabilité, le suivi en temps réel et la notification automatique de la protection civile en cas d'urgence sont des problèmes majeurs de ces solutions. En plus l'absence de la détection de flammes et le manque d'autonomie en cas de la détection de gaz, représentent les inconvénients majeurs de ces solutions.

1.3 Etude de l'existant

1.3.1 Solutions existantes

Nous allons présenter quelques systèmes de détection de gaz comme le montre le tableau 1.1 :

Titre	Image	Description
Détecteur de gaz – Autonomes-sans fils		C'est un détecteur de gaz qui mesure la concentration de gaz GPL, naturel ou de ville, ce principe est simple si le niveau de gaz devient trop élevée des alarmes visuelle et sonore se déclenchent immédiatement.
Détecteur de gaz LH-89F		ce détecteur de gaz peut détecter les fuites de gaz à temps et envoyer un signal d'alarme aux systèmes.
Détecteur de fuite de gaz « gaz-88 » Autonome		Lors de la détection de gaz naturel ou GPL un message vocal en coordination avec l'alarme sera déclenché

Tableau 1.1 :Les solutions existantes

1.3.2 Critique de l'existant

Après une étude des solutions existantes, nous avons constaté les points faibles suivants:

- Absence de suivi en temps réel de la part de l'utilisateur.
- Absence de notification, de l'utilisateur et de la protection civile en cas d'urgence.
- Absence de détection de l'incendie.
- Manque de sécurité qui permet de protéger le lieu

1.4 Solution proposée

Après avoir présenté les solutions existantes ainsi que les limitations de ces dernières, on a pensé à réaliser un projet pour trouver une solution optimale qui peut détecter le gaz naturel ainsi que la surveillance en temps réel de la part de l'utilisateur et la protection civile avec la

détection des flammes en cas de présence de ces dernières. Notre solution doit satisfaire les exigences suivantes :

- Portabilité.
- Alerte instantanée en cas d'urgence.
- Détection automatique de la fuite et l'incendie à l'aide de l'intelligence artificielle.

❖ Solution proposée

Parmi les solutions de gestion de contrôle de fuite du gaz et l'incendie nous citons essentiellement les solutions web et Android.

Des interfaces web sont à la disposition de l'utilisateur et des agents de la protection civile, afin d'assurer la détection et la signalisation de la présence de flammes et des fuites de gaz. Parmi les fonctionnalités offertes. Il y a la consultation automatique des valeurs du gaz température et d'humidité, ces dernières sont stockées dans une base de données. Nous pouvons doter le système d'autres fonctionnalités comme la consultation des images ainsi que la localisation géographique du lieu de l'incendie de la part de la protection civile.

Un schéma synoptique de la solution proposée est donné par la figure 1.1

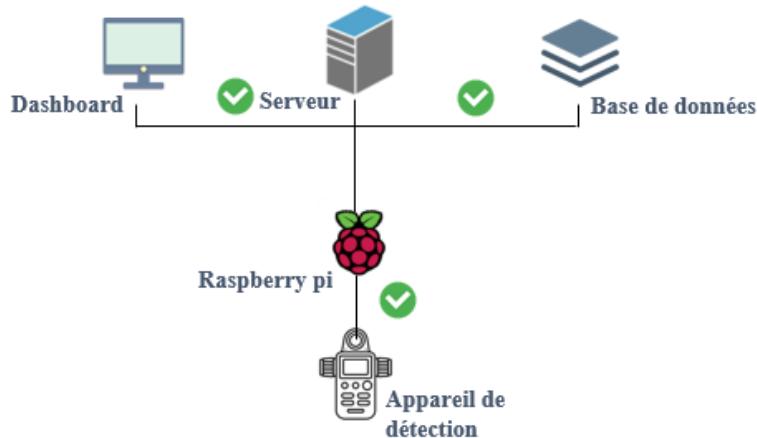


Figure 1.1: Schéma synoptique de la solution envisagée

Outre que la solution Web nous proposons aussi une application Android sur Smart Phone, ce qui va garantir une meilleure sécurité grâce à un système de notification plus réactif fourni par l'application mobile

1.5 Analyse et spécification des besoins

1.5.1 Etude préliminaire des solutions proposées

Avant de se plonger dans la conception et la réalisation d'un système de détection de fuite de gaz et de flamme avec une fiabilité et une facilité d'intégration dans les différents domaines, l'étude et la spécification des besoins représentent deux phases cruciales de ce projet.

Dans ce qui suit nous allons proposer des solutions permettant de satisfaire les besoins de notre projet, tout en respectant les exigences de sécurité et un bon rapport qualité-prix. Notre projet nécessite une partie centrale capable de détecter et de traiter les différentes données « gaz, Température, Humidité et images » les envoyer à l'utilisateur grâce au serveur par le biais d'un protocole de communication. Dans ce qui suit nous allons présenter différentes solutions selon le protocole de communication utilisé.

1.5.1.1 Solution à base du protocole WIFI

Cette solution se base sur le protocole de communication Wi-Fi afin d'assurer la communication entre les différentes parties de système. Il s'agit d'un protocole de communication sans fil régi par les normes du groupe IEEE 802.11 (ISO/IEC 8802-11) [2], il permet à plusieurs appareils informatiques de se connecter via des ondes radio dans un réseau informatique afin de transférer des données entre eux. La figure 1.2 présente un schéma synoptique de cette solution.

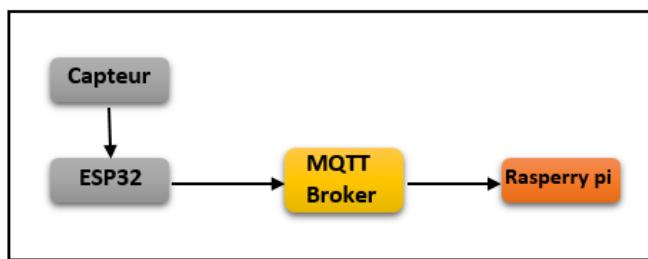


Figure 1.2: Schéma synoptique de la première solution

Le principe est simple notre capteur sera branché avec un microcontrôleur « ESP32 WROOM32 », ce dernier publie les données via le protocole MQTT sur un topic, la Raspberry PI « subscrire » écoute sur le même topic, pour assurer le traitement des données et les envoyer au cloud. Une application Android et une plateforme web est développée pour récupérer et afficher les données reçues en temps réel.

Cette solution avec le protocole de communication Wi-Fi peut être adaptée dans tous les secteurs, que ce soit dans les foyers, les restaurants, les parcs ou les industries. Le Wifi a certes de nombreux avantages, mais la technologie présente aussi des inconvénients: les

communications et les échanges de données ne sont pas acquittées donc c'est un manque de fiabilité.

Dans le tableau 2.1 nous citons quelques avantages et inconvénients de ce protocole.

Avantages	Inconvénients
Supprimé les contraintes de câblage	Débit théorique de 11Mbits/s parfois insuffisant
Maniabilité et simplicité	Partage la bande passante disponible
Débit du 11 Mbits/s théorique soit 6 Mbits/s réel	Distance de connexion de 30 à 100mètres parfois insuffisantes
Favorise la mobilité	Manque de sécurité donc obstacle à son développement
Prix du wifi devenu adorable	

Tableau 1.2 : Les avantages et les inconvénients du protocole de communication Wifi

1.5.1.2 Solution à base du protocole LORA

Afin de remédier aux limitations du protocole WIFI nous pouvons développer une autre solution en se basant sur la technologie LORA.

LoRa est une technologie de communication de données sans fil qui utilise une technique de modulation radio qui peut être générée par les puces de l'émetteur-récepteur 'Semtech' LoRa figure 1.3.

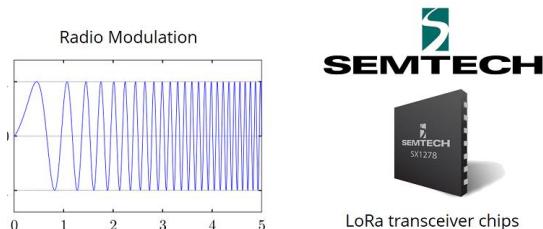


Figure 1.3: Modulation radio

Cette technique de modulation permet la communication à longue portée de petites quantités de données (ce qui signifie une faible bande passante), une immunité élevée aux interférences, tout en minimisant la consommation d'énergie. Ainsi, il permet la communication à longue distance avec des exigences de faible puissance.

❖ Les Fréquences de LoRa

LoRa utilise des fréquences libres qui sont disponibles dans le monde entier. Parmi les fréquences les plus utilisées :

- 868 MHz pour l'Europe.

- 915 MHz pour l'Amérique du Nord.
- 433 MHz pour l'Asie.

Parce que ces bandes sont sans licence, n'importe qui peut les utiliser librement sans payer ou avoir à obtenir une licence.

❖ Les Applications de LoRa

LoRa à longue portée et à faible puissance, le rend parfait pour les capteurs à batterie et les applications à faible puissance dans :

- L'internet des objets (IoT).
- La maison intelligente.
- La communication de machine à machine (M2M).

Ainsi, LoRa est un bon choix pour les nœuds de capteur fonctionnant sur une cellule à bobine ou alimentée par l'énergie solaire, qui transmettent une petite quantité de données. Mais il faut garder à l'esprit que LoRa ne convient pas aux projets qui :

- ✓ Nécessitent une transmission à haut débit de données.
- ✓ Nécessitent des transmissions très fréquentes.

❖ Topologie de LoRa (figure 1.4)

Le protocole LORA peut être utilisé pour des communications point à point ou pour construire un réseau LoRa (en utilisant Lora WAN par exemple)

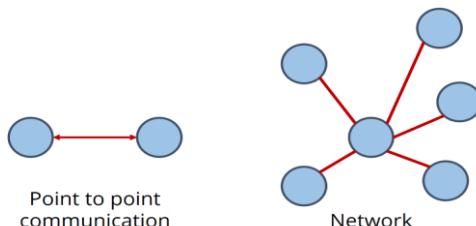


Figure 1.4 Topologie de LoRa

❖ Solution proposée

La même solution, exposée précédemment, peut être élaborée en ayant retourné cette fois-ci au protocole LORA. Ainsi les capteurs mesurent et stockent les données avant de les transmettre sur le réseau Lora, ils envoient leurs trames à une (ou plusieurs) Gateway(s) qui fait office de relais entre un grand nombre de capteurs dans son rayon de couverture et les serveurs de stockage et de traitement de données.

- ➔ Cette solution permet alors de résoudre le problème de manque de fiabilité ainsi que la couverture des grandes superficies.

1.5.2 Spécification de la solution choisie :

Pour des raisons de manque des matériels d'une part et de constraint de temps d'autre part, nous choisissons la première solution, sachant que nous pouvons toutefois faire la migration à la deuxième solution si nous disposons des composants nécessaires (les modules LORA).

1.5.2.1 Analyse fonctionnelle du système proposé

Avant d'entamer la conception de notre système, commençons par une analyse et une décomposition des fonctionnalités à implémenter par le système, comme détaillé par les diagrammes suivants :

- ❖ Diagramme de pieuvre: le diagramme de pieuvre permet de représenter les fonctions de service d'un produit figure 1.5

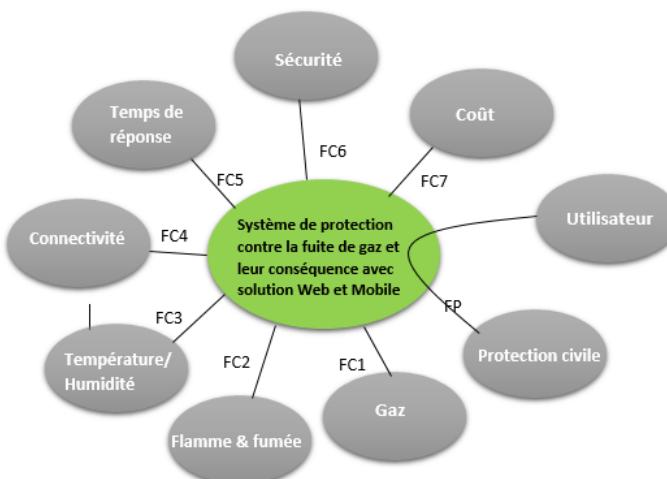


Figure 1.5 : Diagramme de pieuvre

FP: Protection, surveillance et notification des usagers et de la protection civile en cas de suspicion de danger

FC1 : Mesurer la valeur de gaz en temps réel d'une façon quotidienne

FC2 : détecter la présence de flamme et de fumée

FC3 : Mesurer les valeurs de température et d'humidité d'une façon quotidienne

FC4 : Connecter le système à un réseau Wifi

FC5 : répondre en temps réel

- ❖ Diagramme d'analyse fonctionnelle descendante (SADT) (figure 1.6): La méthode SADT est une méthode graphique qui part du général pour aller vers les détails

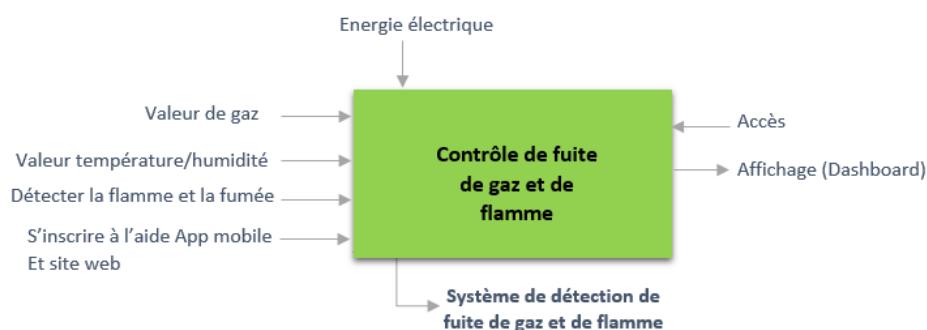


Figure 1.6 : Diagramme d'analyse fonctionnelle descendante (SADT)

- ❖ Diagramme niveau A-0 du système: (figure 1.7) la décomposition au premier niveau de diagramme de SADT. Cette description fait apparaître les différents éléments constituant la solution à développer.

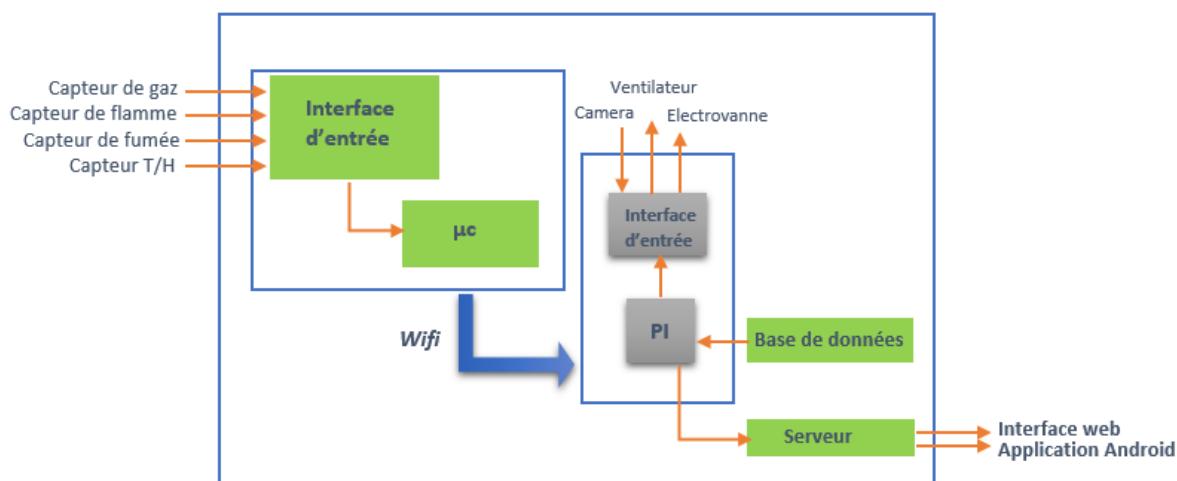


Figure 1.7: Diagramme niveau A-0 du système de détection de fuite du gaz et de flamme

1.5.2.2 Spécifications des besoins :

❖ Besoin fonctionnels

La phase d'analyse des besoins traduit une idée en vision de produit fini. Elle consiste à comprendre l'application à développer et présenter les tâches à accomplir. On va présenter les fonctionnalités de notre application.

Notre application à deux utilisateurs :

➤ Citoyen qui a le droit de :

- Consulter les données du système: cette fonctionnalité permet à un Citoyen de consulter les données qui concernent la dernière valeur de concentration de gaz dans sa « maison, usine, restaurant, etc. » ainsi que les valeurs de la T/H en temps réel.
- Recevoir des alertes en cas de détection de gaz sur l'appareil mobile de l'utilisateur et sur la téléphone pour éviter l'hors ligne de citoyen.

➤ Agent de protection civil qui a le droit de :

Consulter les alertes de détection de l'incendie en temps réel sur l'appareil mobile ainsi la localisation et le numéro de citoyen

❖ Besoins non fonctionnels

Les contraintes non fonctionnelles sont des besoins non négociables qui sont fixés par le client et doivent être respectés au cours de la réalisation de la solution.

Dans notre système à élaborer, nous devons se conformer aux besoins non fonctionnels suivants

➤ Les contraintes de sécurité :

- Fournir une page d'authentification par un mail et un mot de passe.
- Utiliser un système de gestion et sauvegarde de données (SGBD).

➤ Les contraintes ergonomiques :

- Assurer un accès rapide aux données et à l'information.
- Accomplir une interface simple et impressionnante.
- Afficher les informations d'une manière simple et compréhensible.

➤ Les contraintes techniques :

- Assurer un temps de réponse rapide.
- Garantir l'homogénéité de toutes les interfaces de la solution.

1.6 Planification du projet

Nous présentons dans cette partie le déroulement de notre projet dans l'institut supérieur de l'informatique et de mathématique « ISIMM », durant les 4 mois, à partir du 1^{er} février 2022 jusqu'à la fin du mois de Juin. Nous avons fait un plan qui s'organise en 6 phases :

↳ Recherche et Analyse :

- Identifier le problème à résoudre
- Développer un énoncé projet (résumer les recherches, les besoins, les caractéristiques et les critères de la solution proposée).

↳ Réalisation de la partie Hardware de notre système

- Implantation sur une carte nodeMCU ESP32 des capteurs (T/H, flamme, gaz, fumée et un afficheur LCD)
- Implantation sur une carte Raspberry PI une caméra et deux relais avec un ventilateur et une electrovanne.

↳ Réalisation de la première partie software (interfaçage web)

- Construire la forme Login/Sign up pour l'utilisateur.
- Construire l'interface « Dashboard » où nous allons mettre les valeurs de « gaz, T/H et l'historique des valeurs ».
- Communication avec la base de données à la carte pour l'échange des données.
- Construire la forme Login/Sign up pour l'agent de protection civile.
- Construire l'interface « Dashboard » où nous allons afficher l'image du foyer de flamme et la géolocalisation de l'incendie.

↳ Réalisation de la deuxième partie software (Application Android)

- Construire deux modes un pour l'utilisateur et l'autre pour la protection civile.
- Envoi de notification en cas de détection de fuite de gaz, ou d'un dépassement de valeurs usuelles.
- Envoi d'un SMS à l'utilisateur pour éviter le cas de hors ligne.

↳ Tests et validation des fonctionnalités implémentées.

↳ La rédaction du rapport va commencer à partir du mois d'avril.

Dans la figure 1.8 ci-dessous, nous montrons le diagramme de Gantt que nous avons réalisé pour modéliser la planification de notre projet suivant ces différentes phases

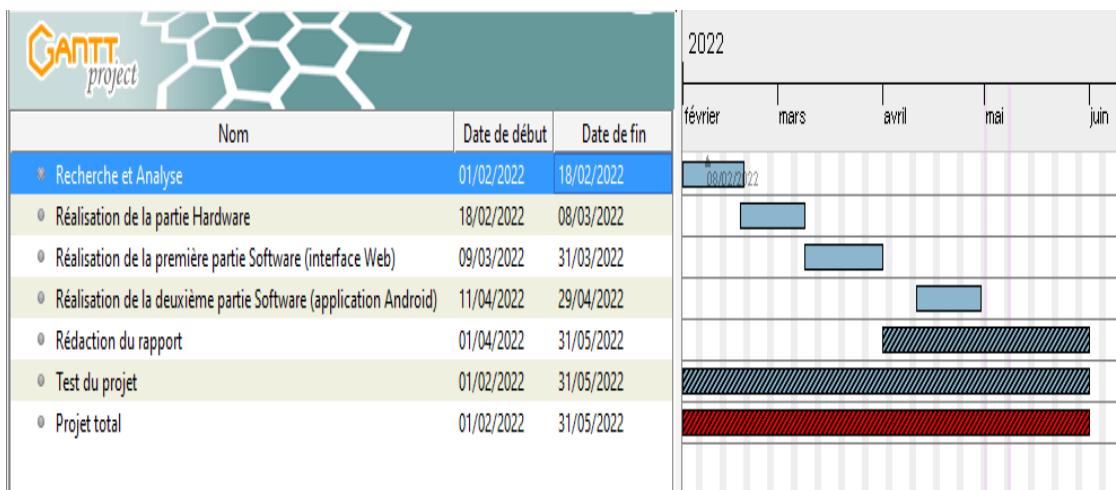


Figure 1.8: Diagramme de Gantt de notre projet.

1.7 Conclusion

Dans ce chapitre, nous avons mis le sujet dans son cadre général, après nous avons identifié la problématique. Dans une seconde étape nous avons étudié les solutions existantes et nous avons proposé notre solution et son architecture globale et le fonctionnement du système avec solution Android et web. Toutes ces solutions seront développées grâce aux outils offerts par la technologie IOT, ils seront détaillés dans les chapitres suivants ainsi dans la prochaine partie nous allons aborder la spécification des besoins de notre nouveau système de protec

Chapitre 2 : Conception détaillée du système de protection

2.1 Introduction

Quand l'IoT est accompagné de capteurs et d'actionneurs, la technologie entre dans la catégorie des systèmes cyber-physique, qui englobent également des technologies telles que les réseaux de distribution d'électricité intelligents (Smart Grid), la domotique, le transport intelligent et les villes intelligentes. Chaque objet connecté est identifiable de façon unique grâce à son système informatique embarqué, mais il est également capable d'interagir au sein de l'infrastructure Internet existante. [1]

Ainsi, nous nous sommes basés sur les outils et les plateformes offertes par l'IOT pour concevoir et mettre au point la solution proposée.

2.2 Principe de fonctionnement

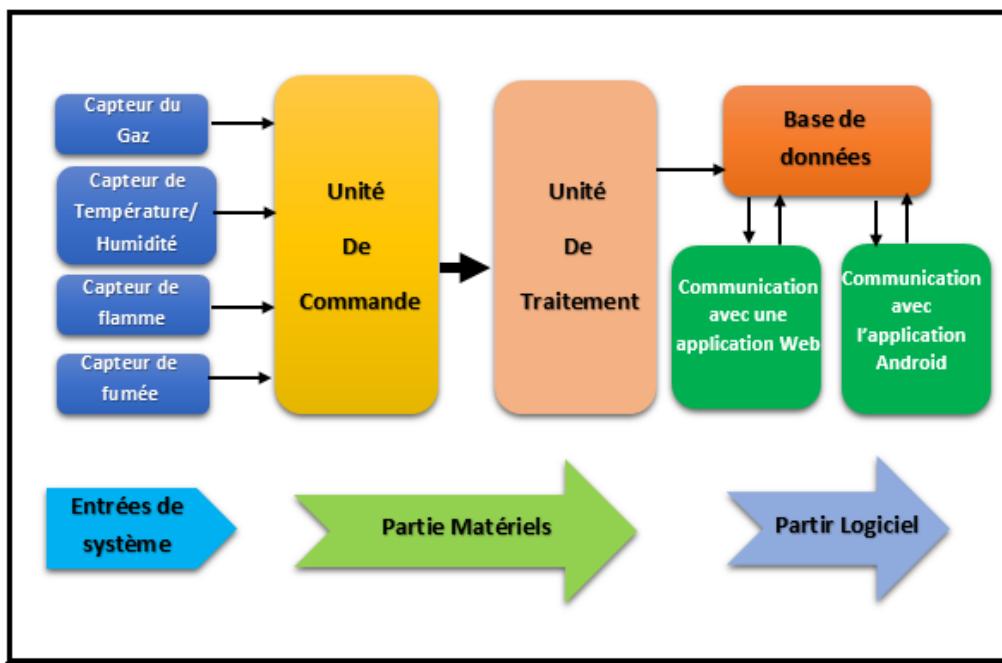


Figure 2.1 Description fonctionnelle du système

Dans la figure 2.1, nous indiquons les principaux blocs qui composent notre système à concevoir et à réaliser. Principalement, en plus des périphériques d'entrées du système, notre système peut être divisé en deux grandes parties : la partie matérielle et la partie logicielle.

Ces différentes parties interagissent entre eux afin de détecter le taux de gaz « GPL ». En effet, les capteurs branchés à un NodeMCU « Esp32 WROOM32 » publie les données de la concentration de gaz et de la température/Humidité via le protocole « MQTT » sur un topic, le raspberry pi « Subsicerber » écoute sur le même topic où le traitement des données ainsi que l'adaptation de ces derniers pour les envoyer à la base des données cloud « Firebase » via « HTTP ». D'autre part, une Application Android et un site Web sont développés pour récupérer et afficher les données en temps réel.

La caméra est activée si une condition est validée sur la concentration de LPG ou la présence de fumée pour capturer des images et les envoyer vers un modèle de classification d'image « Tensorflow ». Le modèle computer vision va décider la présence d'un incendie ou non. Si oui le système envoi l'image capturée de l'incendie vers le cloud puis l'application Android et l'interface Web récupéré

En cas d'incendie ou de fuite de gaz, une notification est envoyée aux usagers et aux agents de protection civile.

2.3 Environnement matériel du système

La partie matérielle du système de détection de fuite du gaz et de flamme proposée est composée essentiellement des modules sont résumés dans la figure 2.2:

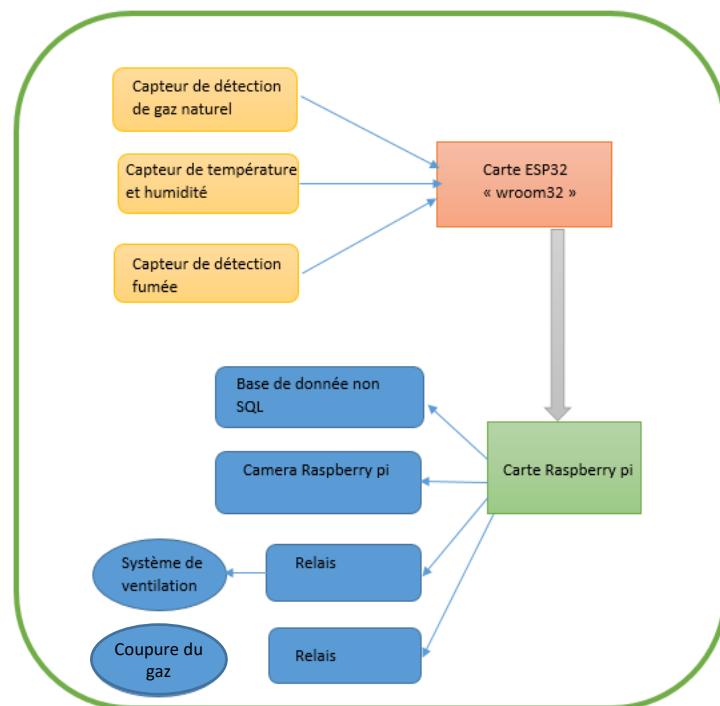


Figure 2.2: Principaux modules de la partie matérielle du système

La description de ces différents modules est donnée ci-après

➤ **Carte Raspberry Pi**

La Raspberry Pi figure 2.3 est ce que l'on appelle une nano-ordinateur mono carte qui peut être branchée à n'importe quel périphérique (souris, clavier...) et disposant d'une connectivité Wi-Fi et Bluetooth capable de lire les vidéos à haute définition. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian, Debian ...) et d'autres logiciels compatibles.

Pour notre solution nous avons choisi la carte Raspberry pi 3 B. La version 3 B est basée sur un processeur ARM Cortex-A53 64 bits quatre cœurs à 1.2 GHZ et possède 1 GB de mémoire RAM. (Voir annexe B)



Figure 2.3: Carte Raspberry PI 3 modèle B

➤ **Carte ESP32 « WROOM32 »**

Les cartes ESP32 sont des cartes très polyvalentes orientées IOT. Elles intègrent la gestion du wifi, du Bluetooth, C'est un microcontrôleur fabriqué par la multinationale chinoise expressif system, l'ESP32 est le successeur de l'ESP8266 avec beaucoup d'améliorations faites au niveau du hardware et du software.[3]

Cette carte est caractérisée par les propriétés suivantes :

- CPU : tensilia Dual Core 32bit 240MHZ LX6 processeur intégré 520KB SRAM.
- Wi-Fi : 802.11 b/g/n/e/i.
- Bluetooth : V4.2 BR/EDR and BLE
- Mémoire : 4MByte Flash

- Température range : 40 to 125 Deg c°.

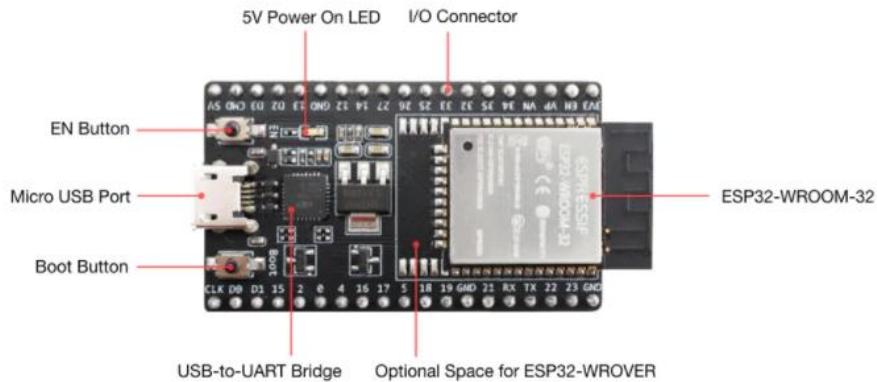


Figure 2.4: les composants de la carte ESP32

L'esp32 possède 30 PIN comme le montre la figure 2.4.

➤ Caméra Raspberry v2

La Raspberry Pi Noir Camera Module V2 figure 2.5 est une caméra de 8 Mégapixels avec capteur Sony IMX219, elle est dotée de la technologie OmniBSI pour Raspberry Pi.



Figure 2.5:camera Raspberry pi v2

Principales caractéristiques :

- Module caméra 8 Mégapixels pour Raspberry Pi
- Technologie OmniBSI
- Résolution photo : 3280 x 2464 pixels
- Résolution vidéo : 1080p30, 720p60, 640x480p90
- Compatible avec toutes les cartes Raspberry (Pi Model A/A+/B/B+, Pi 2 Model B, Pi 3 Model B/B+ ...)

➤ Relais

Le relais est un interrupteur pilotable qui permet d'isoler la partie commande de la partie puissance. C'est un composant très utilisé lorsqu'on veut faire le lien entre électronique et

électricité. Il permet d'ouvrir ou fermer un contacteur sur un circuit de puissance en fonction d'un signal entre 0 et 5V.

La sortie du relais dispose de trois bornes :

- la borne NC (pour "normalement fermé") qui laisse passer le courant au repos.
- la borne NO (pour "normalement ouvert") dont le contact par défaut est ouverte ou déconnecté.



Figure 2.6: les différentes broches de relais

On trouve différents modèles des relais à « 2 canaux, 4 canaux, 8 canaux, etc.. »Comme se montre les figures suivantes



Figure 2.9: Relais à 2 canaux



Figure 2.8: Relais à 4 canaux



Figure 2.7 : Relais à 8 canaux

➤ Capteur DHT22

Le DHT11/DHT22 est un capteur qui mesure l'humidité relative et la température. Il fournit une sortie numérique calibrée avec un protocole à 1 fil. Les deux capteurs sont peu coûteux. Ils sont assez semblables les uns aux autres avec certaines différences de spécifications.

Le DHT22 est presque similaire au DHT11 mais le premier mesure la température et l'humidité avec une plus grande précision.



Figure 2.10: capteur DHT22

Le tableau 2.2 présente la comparaison des deux capteurs DHT.

	DHT22	DHT11
Température	-40 to 80 °C +/- 0.5°C	0 to 50 °C +/- 2 °C
Humidité	0 to 100% +/-2%	20 to 90% +/-5%
Résolution	Humidité: 0.1%Temperature: 0.1°C	Humidité: 1%Temperature: 1°C
Tension de fonctionnement	3- 6 V DC (alimenté directement par ESP32/ESP8266)	3-5.5 V DC (alimenté directement par ESP32/ESP8266)
Temps d'échantillonnage	2 seconds	1 second
Type de données de sortie	Float	Int
brochage	4-pin (même que DHT11)	4-pin (même que DHT22)
Prix	35.00	9.00

Tableau 2.1: Comparaison entre DHT22 et DHT11

Le capteur est constitué de deux parties, une pour la mesure de température et l'autre pour la mesure d'humidité (voir l'Annexe A).

➤ Capteur MQ9

La plupart des fuites proviennent du gaz naturel (GPL), alors on a décidé de sélectionner le capteur de gaz MQ9 (figure 2.11), à travers la variation de sa résistance interne le capteur est capable de détecter plusieurs gaz combustible comme le méthane (CH4), le monoxyde de carbone (CO) et le gaz naturel.

➔ Ce capteur doit être étalonné conformément aux indications de « l'annexe C ».



Figure 2.11: Capteur MQ-9

✓ **Caractéristiques et spécifications du capteur MQ9**

- Bonne sensibilité au CO/gaz combustible
- Haute sensibilité au méthane, au propane et au CO
- Longue durée de vie, stable et à faible coût

Ce capteur admet 4 pins dont 2 pour l'alimentation (VCC, GND) Tension d'alimentation +5V, et 2 pins pour deux sorties (analogique et numérique), concentration de détection maximale 1000/10000 ppm, concentration minimal de détection 200 ppm, résistance de chauffage 0.5-340mW et résistance chauffante $33 \Omega \pm 5$.

✓ **Brochage du capteur MQ-9 (tableau 2.3)**

Numéro	Nom	Description
1	VCC	Le vcc pour alimenter le module, nécessite 5V
2	GND	Pour connecter le module à la masse commune du système
3	A0	Broche de sortie analogique. Tension analogique basée sur la concentration du gaz
4	D0	Broche de sortie numérique, la valeur de seuil peut être définie à l'aide du potentiomètre

Tableau 2.2 : Les broches du capteur MQ-9

➤ **Capteur MQ2**

Le capteur MQ2 (figure 2.12) est dédié à la détection de fumée et d'autre gaz, on a décidé de sélectionner ce capteur grâce à son temps de réponse court, les mesures peuvent être prises rapidement. La sensibilité peut également être ajustée par un potentiomètre.

Le principe de fonctionnement et l'étalonnage de ce capteur se trouvent dans « l'Annexe C »



Figure 2.12: Capteur de fumée MQ2

✓ **Caractéristiques**

- La tension de fonctionnement est de + 5 V
- Peut mesurer ou détecter le GPL, l'alcool, le propane, l'hydrogène, le CO et même le méthane
- Tension de sortie analogique : 0V à 5V

- Tension de sortie numérique : 0 V ou 5 V (logique TTL)
 - Durée de préchauffage 20 secondes
 - Peut être utilisé comme capteur numérique ou analogique
 - Il peut faire varier la sensibilité de la broche numérique à l'aide du potentiomètre
 - Large portée de détection
 - Réponse rapide et haute sensibilité
 - Stable et longue durée de vie Circuit d'entraînement simple
- ✓ **Brochage du capteur MQ-2**

Les broches du capteur MQ2 sont identiques à celles du capteur MQ9.

➤ **Capteur de flamme IR LM393**

Nous avons utilisé le détecteur de flamme qui fait partie essentiellement de l'installation de détection d'incendie. Le module (illustré dans la figure 2.13) permet la détection de flamme à l'aide d'un récepteur infrarouge qui va capter les émissions lumineuses de sources de chaleur.

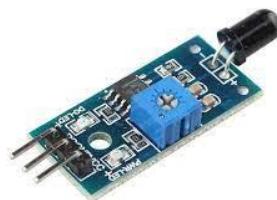


Figure 2.13 : capteur de flamme

Les caractéristiques de ce capteur sont les suivantes.

- La photosensibilité est élevée
- Le temps de réponse est rapide
- La sensibilité est réglable
- L'angle de détection est de 60°,
- Il est sensible à la plage de flamme.
- La tension de fonctionnement de ce capteur est de 3,3 V à 5 V
- La taille du PCB est de 3 cm X 1,6 cm

➤ **Afficheur LCD I2C 16x2**

Afficheur LCD 2 x 16 (figure 2.14) caractères rétro-éclairé se raccordant via le bus I2C sur un microcontrôleur, nous utilisons l'afficheur LCD pour l'affichage des valeurs de température d'humidité et de la valeur de gaz.

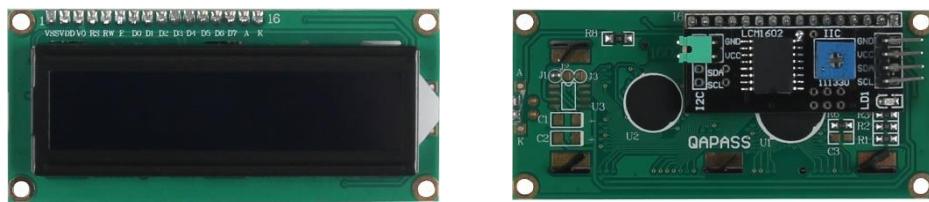


Figure 2.14 : Afficheur LCD I2C 16x2

✓ Caractéristique

- Alimentation: 5 Vcc
- Interface I2C (adresse 0x27)
- Caractères blancs sur fond bleu
- Contraste ajustable via potentiomètre
- Dimensions: 80 x 38 x 18 mm

➤ Buzzer

Un Buzzer (figure 2.15) est une sorte de haut-parleur mais de faible puissance qui va émettre un son en fonction de la fréquence et amplitude de vibration



Figure 2.15 : Buzzer

Il s'agit d'un actionneur utilisé lorsqu'il y a une fuite de gaz.

➤ Electrovanne

Une électrovanne (figure 2.16) est un dispositif qui ouvre ou ferme un circuit en envoyant un courant électrique déclenchant un champ électromagnétique créé par la bobine installée sur l'électrovanne.



Figure 2.16 : électrovanne

➤ Ventilateur

Un ventilateur (figure 2.17) est conçu pour éliminer l'humidité, les fumées et faire baisser la concentration de gaz en cas de fuite.



Figure 2.17: ventilateur 12V

2.7 Etude conceptuelle

Après avoir détaillé l'idée du projet, nous explicitons dans cette partie du chapitre la conception détaillée. Nous présentons la conception détaillée à travers les diagrammes de cas d'utilisation et de séquences qui suivent.

❖ Outils de modélisation UML

UML est un outil de conception simple avec lequel les utilisateurs peuvent rapidement configurer des diagrammes UML. Il est généralement utilisé en conception orientée objet et en développement logiciel.

L'UML utilise des éléments et les associe de différentes manières pour former des diagrammes qui représentent les aspects statiques ou structurels d'un système, ainsi que des diagrammes comportementaux qui capturent les aspects dynamiques d'un système.

2.7.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les relations entre les acteurs et les fonctionnalités du système. Les cas d'utilisation présentent une vue externe de la façon d'utiliser un système, que ce soit l'application, un sous-système, une fonction ou un composant.

➤ Identification des cas d'utilisations

Chacun des acteurs interagissant avec le système possède divers cas d'utilisations dont on distingue dans le tableau 2.5 :

Cas d'utilisation pour le citoyen	Cas d'utilisation pour l'agent de protection civil
S'inscrire	S'inscrire
S'authentifier	S'authentifier
Consulter les valeurs de gaz, température et humidité	Consulter l'alerte en cas de détection de flamme
Recevoir des alertes en cas de fuite de gaz	Consulter la localisation et l'image de l'incendie

Tableau 2.3 : Les acteurs interagissent avec le système

➤ Diagramme des cas d'utilisation global :

La figure 2.18 représente le diagramme de cas d'utilisation global

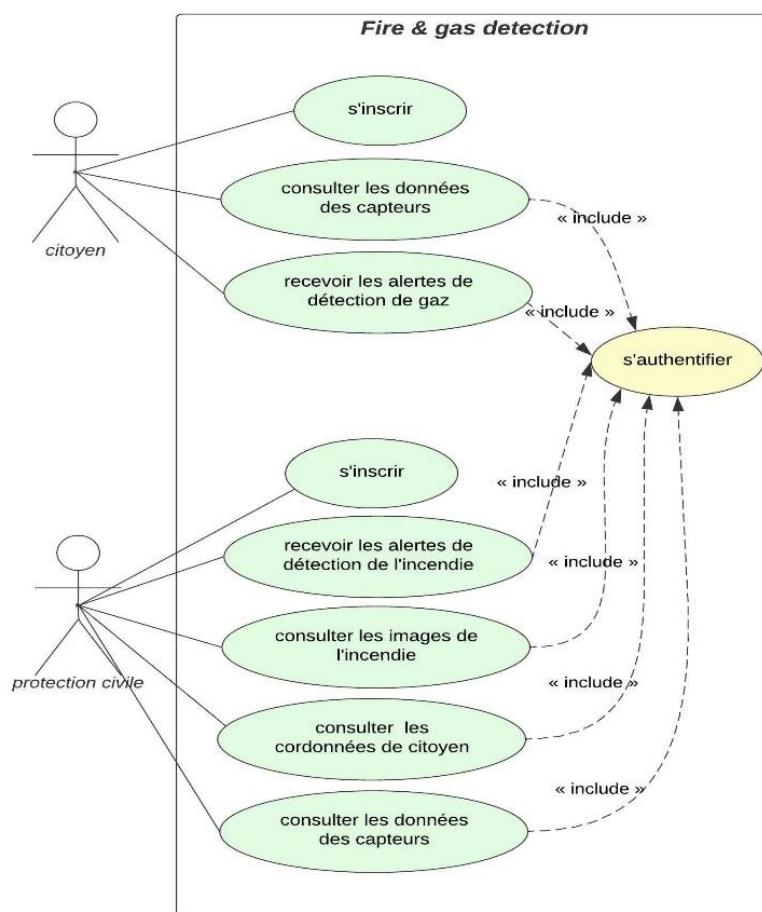


Figure 2.18 : Diagramme de cas d'utilisation global

➤ diagramme des cas d'utilisation Raffiné pour le citoyen:

La figure 2.19 ci-dessous présente le diagramme de cas d'utilisation Raffiné pour un citoyen :

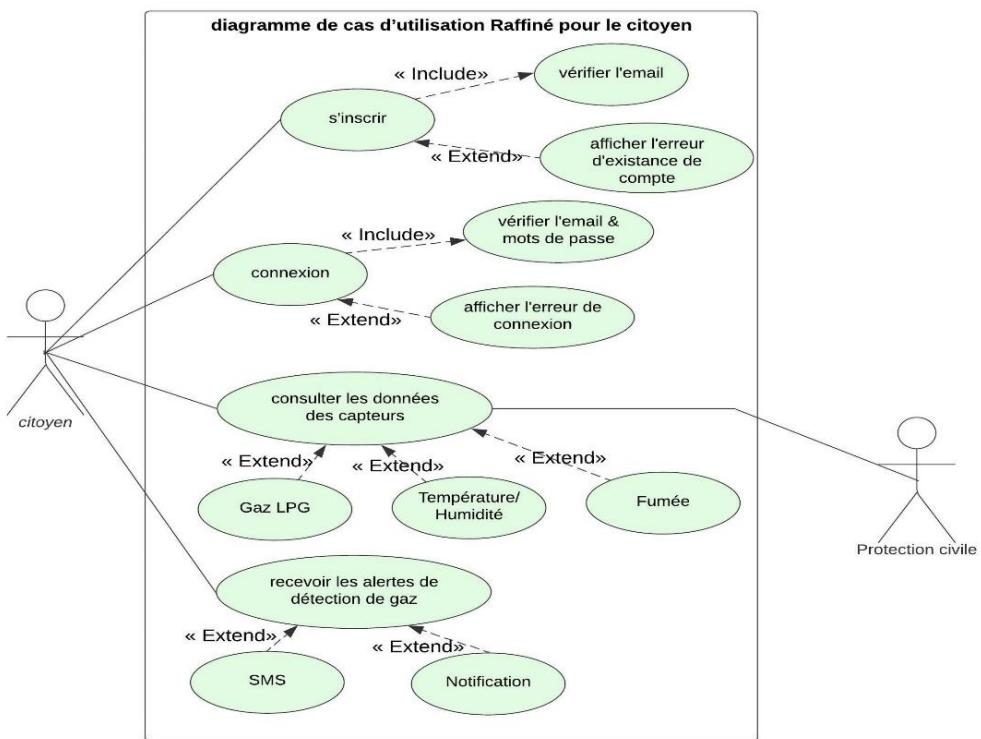


Figure 2.19 : Diagramme des cas d'utilisation raffiné pour le citoyen

- Description des cas d'utilisation « s'inscrire » :**

Ce cas permet à l'utilisateur de créer un compte que ce soit dans l'application mobile ou la plateforme Web et vérifier si le citoyen a déjà un compte ou pas.

- Description des cas d'utilisation « connexion » :**

Ce cas permet à l'utilisateur de s'authentifier que ce soit dans l'application mobile ou la plateforme web et vérifier l'existence de compte

- Description des cas d'utilisation « Consulter les données des capteurs » :**

Après l'authentification l'utilisateur peut consulter les valeurs de gaz naturel, la température, la valeur d'humidité et la valeur de fumée. Dans ce cas, l'agent de protection civile joue le rôle d'un acteur secondaire où peut également consulter les valeurs des détecteurs.

- Description des cas d'utilisation « Recevoir d'alerte » :**

Lors de la détection de gaz ou de fumées l'utilisateur reçoit un message d'alerte pour dire qu'il y a une fuite de gaz, le message est reçu sous forme d'une notification sur l'application Android pour éviter les problèmes de jouabilité de l'utilisateur on a prévu d'envoyer le message d'alerte sous forme d'un SMS

➤ Diagramme de cas d'utilisation Raffiné pour un agent de protection civil :

La figure 2.20 représente le diagramme de cas d'utilisation Raffiné pour l'agent de protection civil.

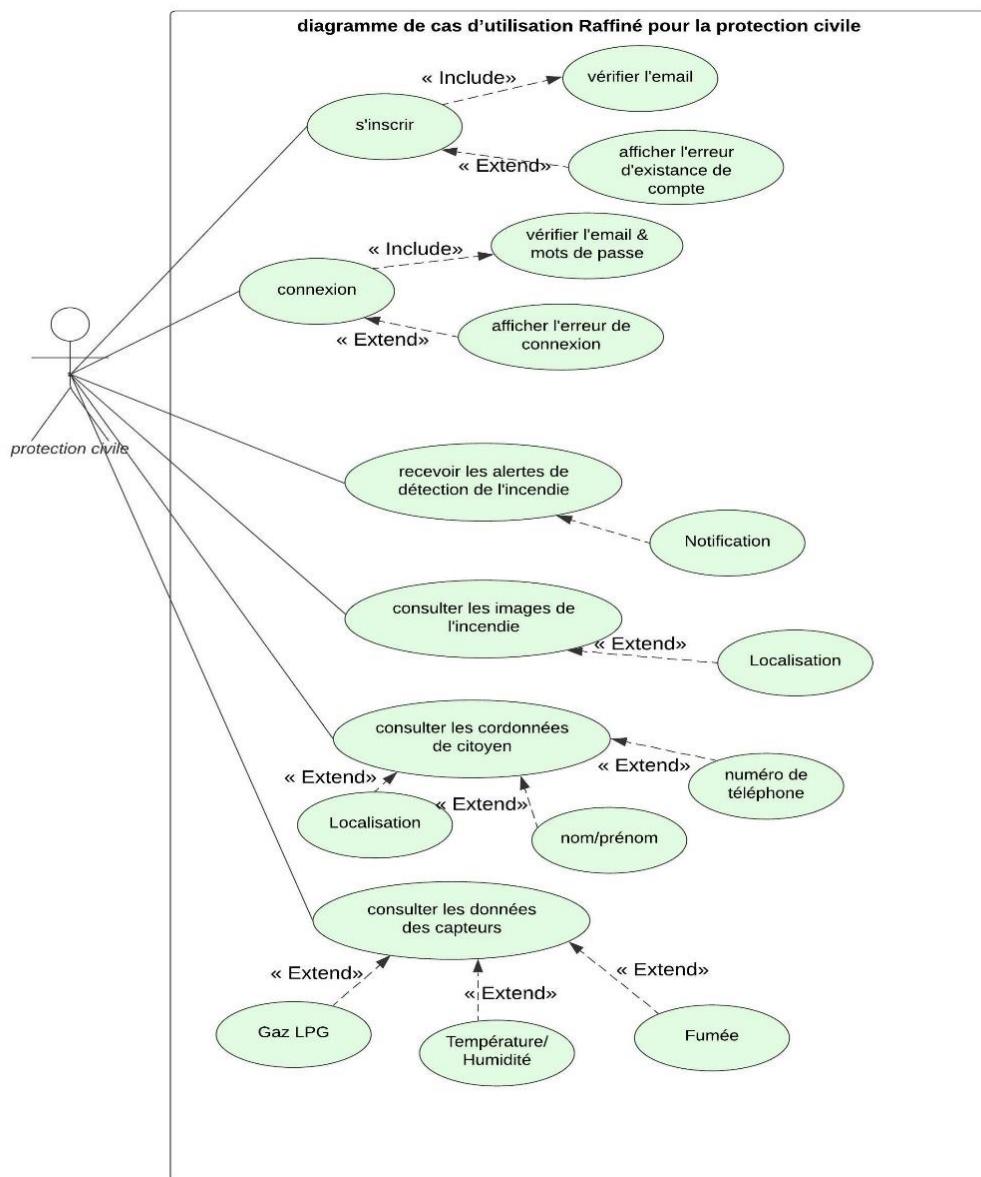


Figure 2.20: Diagramme des cas d'utilisation Raffiné pour un agent de protection civil

• **Description des cas d'utilisation « S'inscrire » & « connexion » :**

Ce cas permet l'agent de protection civil de créer un compte dans l'application mobile et la plateforme Web par le nom /prénom et le numéro de série.

• **Description des cas d'utilisation « Recevoir les alertes de détection de gaz » :**

Lors de la détection de gaz ou de fumée une caméra sera ouverte pour détecter la présence de flamme, en cas de présence une notification d'alerte sera envoyée à la protection civile.

- **Description des cas d'utilisation « consulter les images de l'incendie» :**

En cas de présence de flamme, une image du feu sera envoyée à la protection civile afin d'analyser le volume de flamme.

- **Description des cas d'utilisation « consulter les coordonnées de citoyen» :**

Dans cette partie l'agent de protection civil consulte-les coordonnées de l'utilisateur, la localisation et le numéro de téléphone.

- **Description des cas d'utilisation « consulter les données des capteurs » :**

Pour ce cas, l'agent de protection civile a la droite de consulter les valeurs de gaz LPG, T/H, fumée pour juger la source de l'incendie.

2.7.2 Description comportementale du système

Le diagramme de séquence décrit la façon dont le système effectue les opérations spécifiées dans le diagramme de cas d'utilisation. En fait, on l'utilise pour dessiner la ligne de vie d'un objet UML et montrer une série chronologique d'appels de méthode entre les objets UML.

2.7.2.1 Authentification d'un utilisateur « citoyen »

La figure 2.21 représente diagramme qui permet de représenter des échanges entre les différents objets et acteurs du système pour le cas d'utilisation du citoyen « authentification »

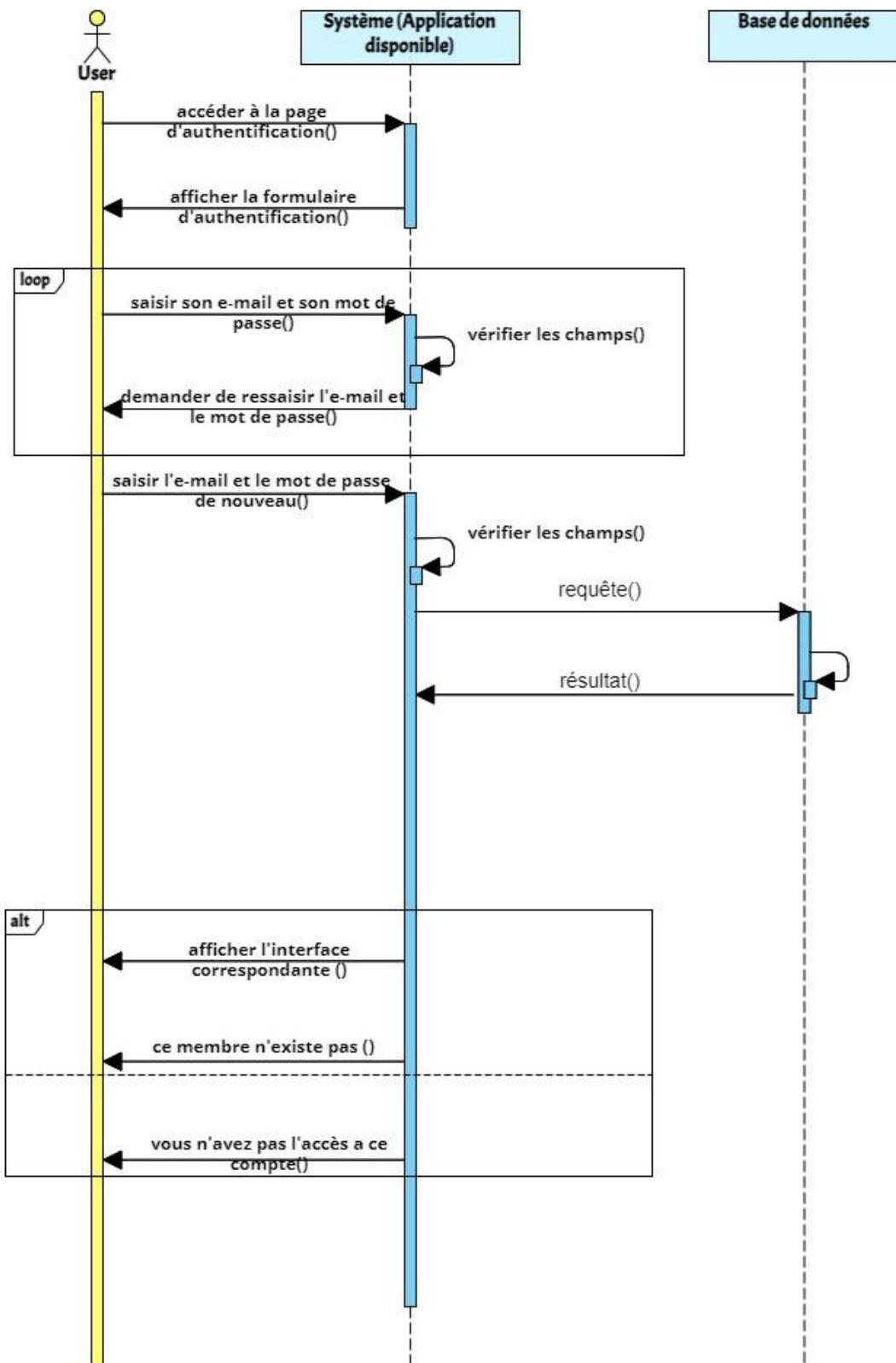


Figure 2.21: Diagramme de séquence pour l'utilisateur citoyen « authentification »

2.7.2.3 Authentification d'un agent de protection civile

Le diagramme suivant (figure 2.22) permet de représenter des échanges entre les différents objets et acteurs du système pour le cas d'utilisation d'un agent de protection civile « authentification »

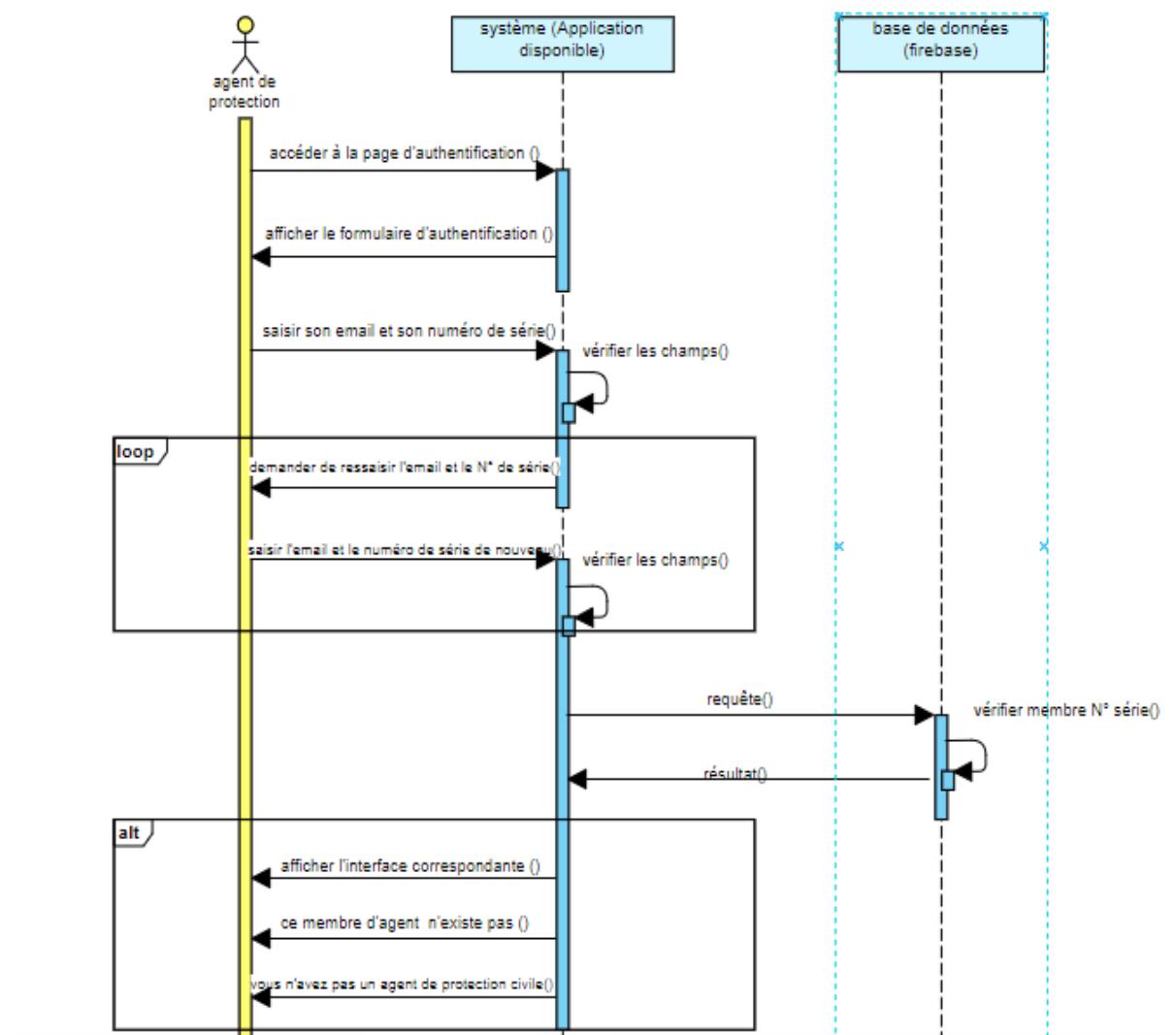


Figure 2.22: Diagramme de séquence d'un agent de protection civil « authentification »

2.7.2.4 Consulter les données des capteurs

Le diagramme suivant permet de représenter des échanges entre les différents objets et acteurs du système pour le cas d'utilisation « consulter les données des capteurs » par le citoyen après l'authentification auprès du système.

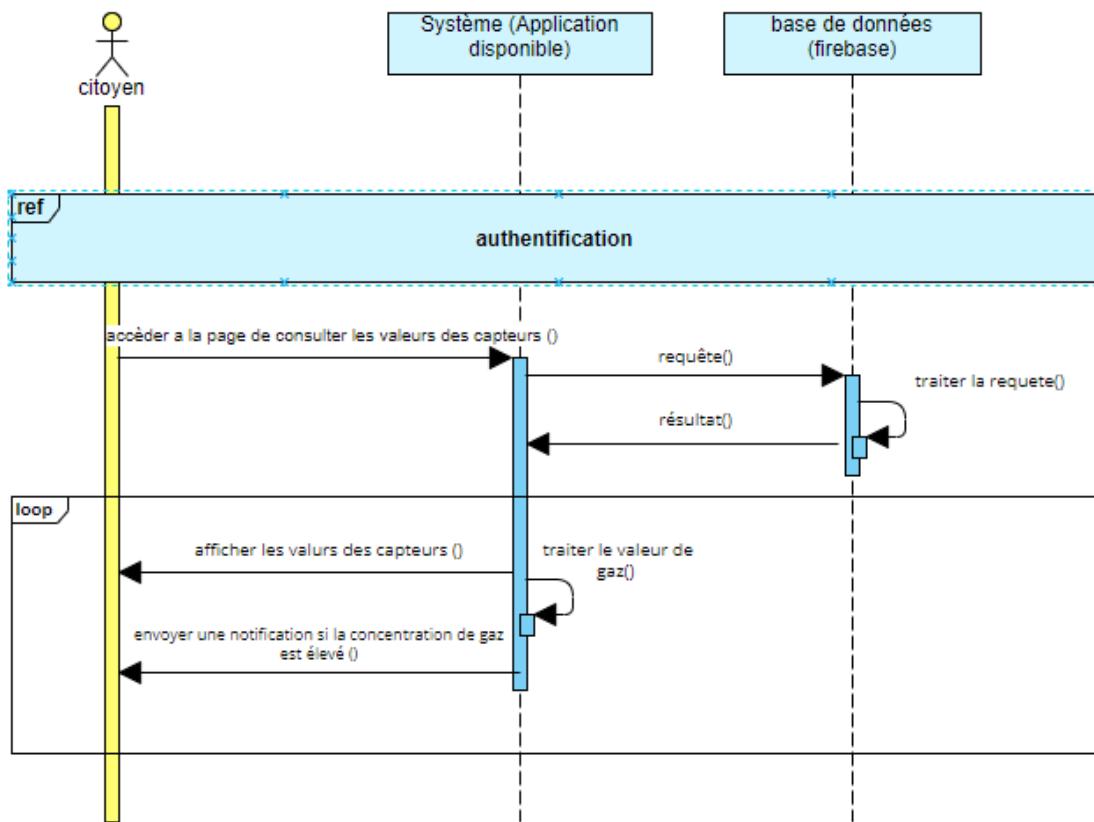


Figure 2.23: Diagramme de séquence « consulter les données des capteurs »

2.7.2.5 Consulter les alertes et l'image de l'incendie

Le diagramme suivant permet de représenter des échanges entre les différents objets et acteurs du système pour le cas d'utilisation « consulter les alerte de détection de feu » et « consulter l'image de l'incendie » par l'agent de protection civile après l'authentification auprès du système.

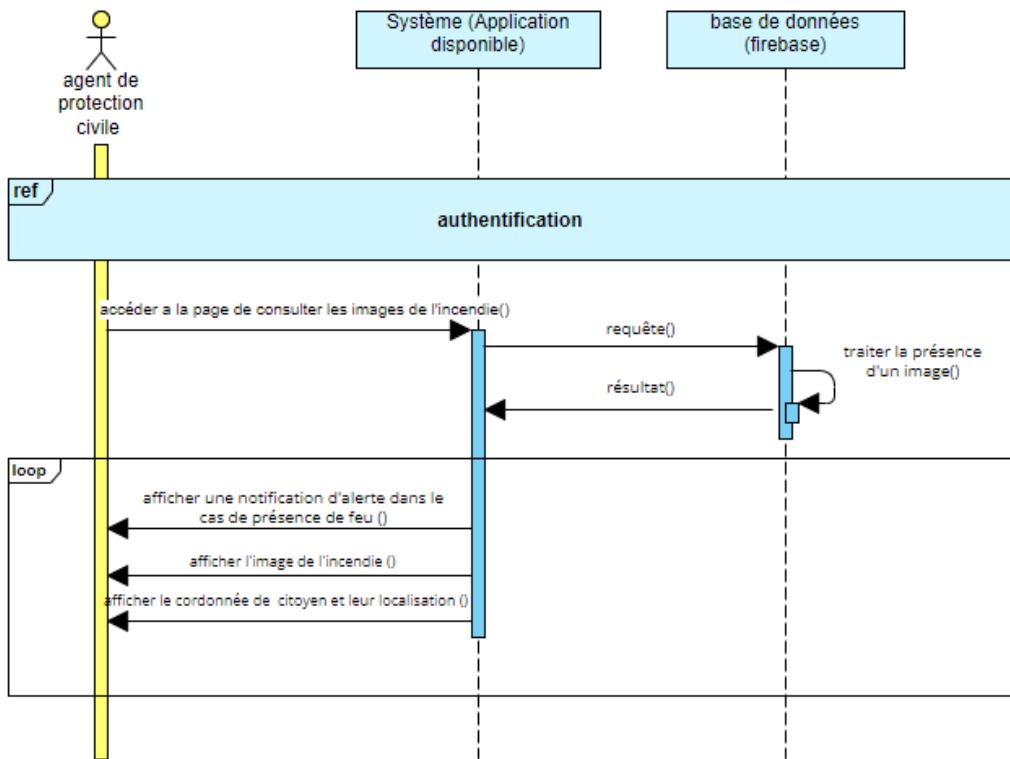


Figure 2.24: Diagramme de séquence « consulter les alerte et l'image de l'incendie »

2.8 Conclusion

Tout au long de ce chapitre nous avons présenté la conception matérielle et logicielle de notre système. Cette modélisation sera exploitée lors de la réalisation, ce qui fera l'objet du dernier chapitre. Dans ce qui suit nous montrons les résultats de la réalisation de notre système.

Chapitre 3 : Implémentation et tests de validation

3.1 Introduction

Une fois la phase de conception est terminée, on aborde le dernier volet du rapport pour exposer la phase de réalisation de notre système. On présente dans cette partie, les environnements logiciels utilisés et les outils de programmation lors de la réalisation du système et le développement des applications. On détaille par la suite l'architecture et on finit par exposer quelques interfaces afin d'exposer les principales activités du système.

3.2 Architecture globale de solution proposée

Le diagramme suivant (figure 3.1) illustre le principe de fonctionnement global à implémenter par le système.

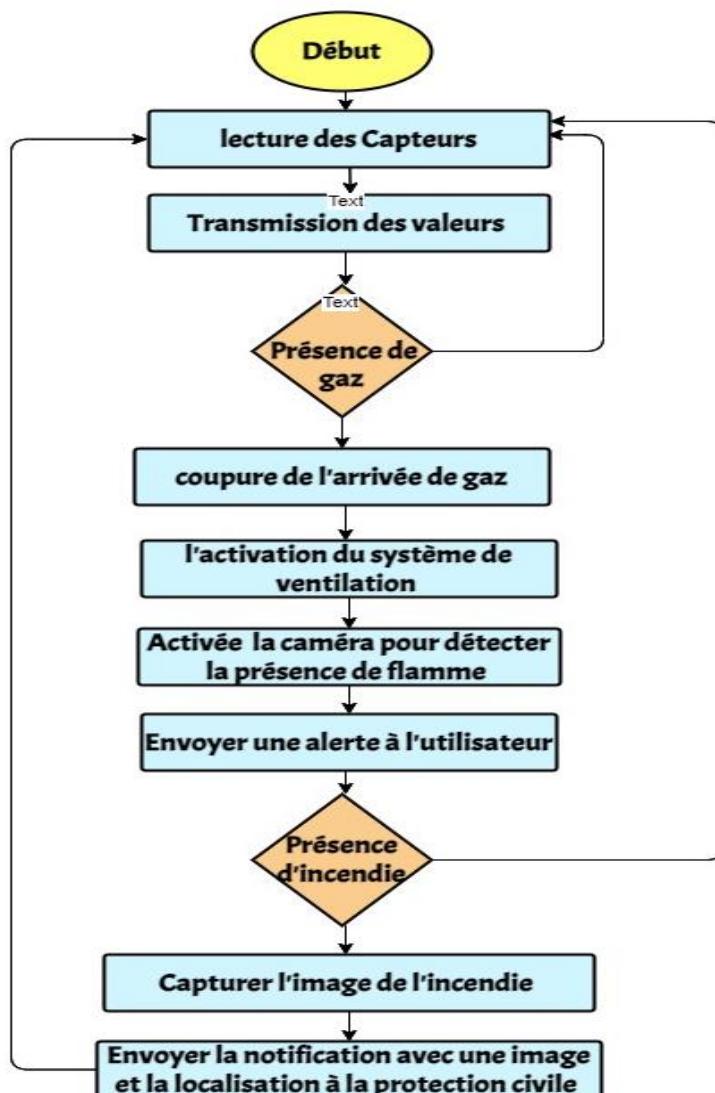


Figure 3.1: Mode de fonctionnement global du système

3.2.1 Architecture plate de la solution proposée

Dans cette partie nous allons représenter notre architecture plate de la solution proposée ainsi que les différentes parties comme indique-la (figure 3.2)

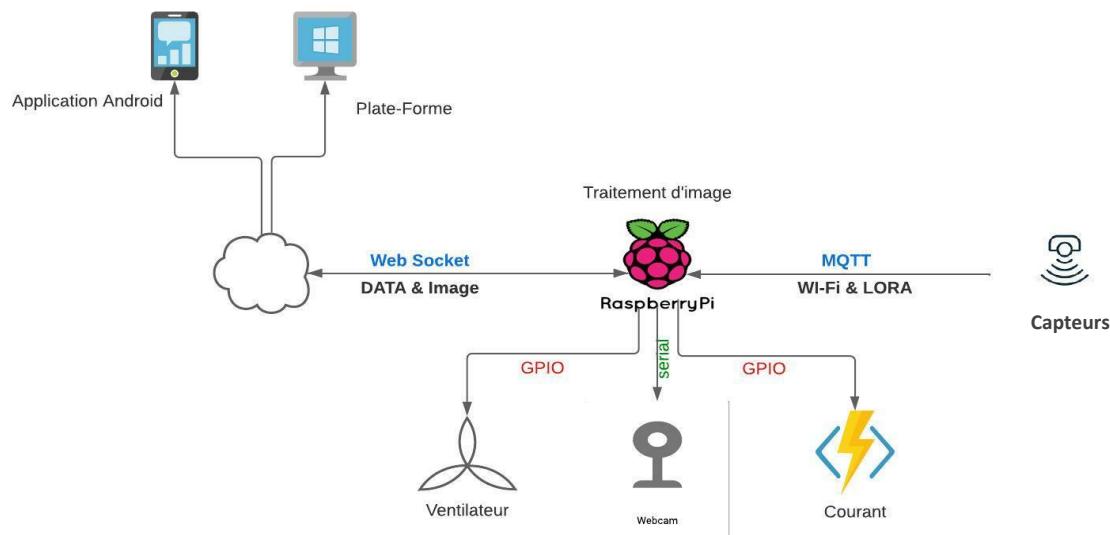


Figure 3.2 : Architecture plate de la solution proposée

3.2.2 Architecture par couche de la solution proposée

Dans cette session Nous allons décrire l'architecture du projet couche par couche, nous avons choisi l'architecture à trois couches (figure 3.3)

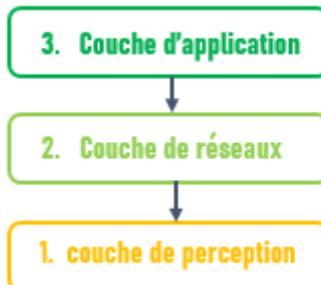


Figure 3.3: Architecture par couche de la solution

L'architecture de notre système se compose de 3 niveaux qui communiquent entre eux pour connecter le monde physique avec le monde virtuel des réseaux et du Cloud.

Ses niveaux sont :

- ❖ **La couche perception**

Cette couche est responsable de convertir des signaux analogiques en données numériques et vice versa. L'étape initiale de tout système IOT englobe un large éventail d'objets qui agissent comme un pont entre les mondes réel et numérique.

➔ Cette couche contient un ensemble des capteurs et des Actionneurs.

❖ ***La couche de réseaux***

Permet d'analyser, diffusées et stockées les données de capteurs et aussi faire la liaison entre ces objets intelligents à d'autres objets intelligents, c'est la couche responsable de la transmission des données reçus de la couche de perception à une base de données, serveur, ou d'un centre de traitement.

➔ les principales technologies utilisées pour réaliser cette couche dans notre projet sont : les technologies cellulaires 2G/3G/4G, wifi, MQTT...

❖ ***La couche d'application***

La couche application est ce avec quoi l'utilisateur interagit. C'est ce qui est chargé de fournir des services spécifiques à l'utilisateur.

➔ Cette couche contient une plateforme et une application mobile qui permet d'afficher les données.

3.3 Environnement logiciel et outils de programmation

Avant d'entamer la partie pratique, nous passons tout d'abord à la description des logiciels et des outils de développement que nous avons utilisé afin de développer notre système. La description de tous ces outils est bien mentionnée ci-dessous :

➤ ***Node RED***

Node-RED est un outil de développement basé sur les flux pour la programmation visuelle développée à l'origine par IBM pour connecter des périphériques matériels, des API et des services en ligne dans le cadre de l'Internet des objets

➤ ***Mosquitto Broker :***

Mosquitto est un serveur MQTT Open Source (Broker) que l'on peut installer sur un Raspberry Pi mais aussi sur presque tous les systèmes d'exploitation (MacOs, Windows, Linux...).

MQTT (figure 3.4) est un protocole de communication très rapide et léger particulièrement bien adapté à la domotique et aux objets connectés. Il facilite la communication entre objets connectés (M2M) tout en économisant la batterie

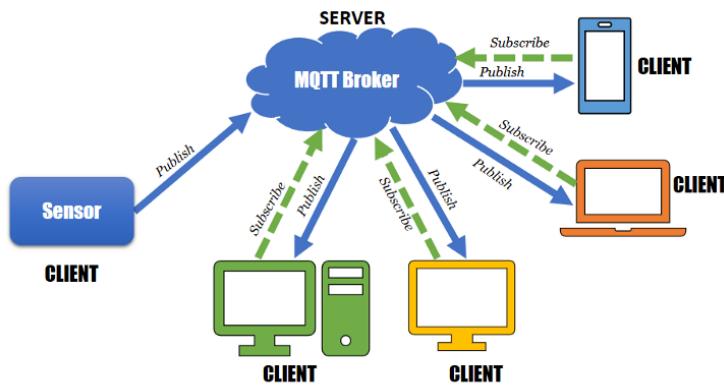


Figure 3.4 : Serveur MQTT

➤ Tensorflow

Tensorflow c'est une bibliothèque open source qui compatible avec python c'est une technologie qui rend l'intelligence Artificielle et le Machine Learning plus rapide mais aussi plus simple. [4]

Ci-dessous vous pouvez voir le diagramme actuel de la plateforme TensorFlow (figure 3.5)

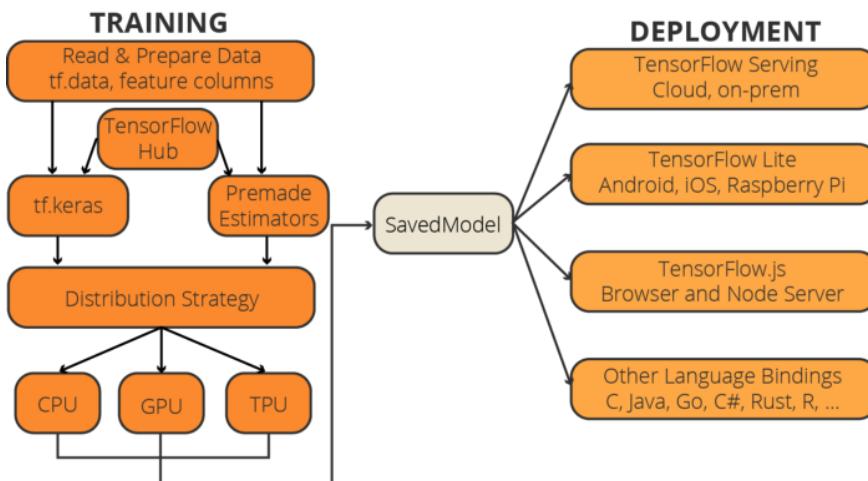


Figure 3.5: diagramme actuel de la plateforme TensorFlow. [5]

➤ Firebase

Filetasse est le nom d'une plateforme mobile de Google qui facilite la création de back-end à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web. Firebase offre différents services et outils que les développeurs exploitent le plus dans le cadre de développement

- ❖ Authentification
- ❖ Cloud Firestore
- ❖ Database Realtime
- ❖ Storage
- ❖ Hosting

➤ ***Android studio***

Il s'agit d'un environnement de développement pour développer des applications mobiles Android. Il est basé sur « IntelliJ IDEA » et utilise le moteur de production « Gradle ». Il peut être téléchargé sous les systèmes d'exploitation « Windows, MacOs, Chrom OS et Linux ».

➤ ***React JS***

React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état [6].

➤ ***Fritzing***

Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon. [7]

➤ ***EasyEDA***

EasyEDA est une suite d'outils EDA basée sur le Web qui permet aux ingénieurs hardware de concevoir, simuler, partager - publiquement et en privé - et discuter de schémas, simulations et cartes de circuits imprimés. [8]

3.4 Tests des capteurs à utiliser

3.3.1 Mise en marche du capteur de gaz MQ9

On a commencé par brancher le capteur MQ-9 avec la carte esp32 comme le montre la figure 3.6 et le tableau 3.1 :

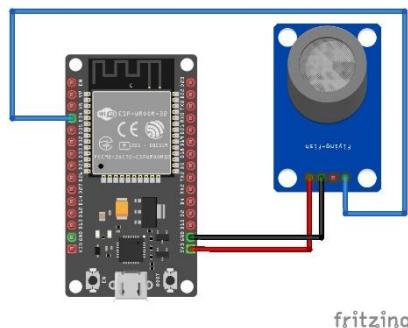


Figure 3.6: Capteur de gaz MQ9 avec la carte ESP32

Branchement (Tableau 3.1)

Capteur MQ9	Carte ESP32
VCC	3.3 v
GND	GND
A0	D32
D0	NC

Tableau 3.1 : Branchement de capteur MQ-9 avec ESP32

Etape de mesure de gaz :

Nous commençons par mettre le capteur dans un milieu ne contenant que de l'air, par la suite nous nous basons sur 100 mesures du capteur, pour calculer la moyenne des valeurs acquises. La tension correspondante est déduite de cette valeur en la multipliant par **3.3/4096**.

La résistance du capteur à l'air libre est déduite par la relation **(3.3 – tension) / tension**. Selon la documentation du capteur, R0 peut être déduit par la relation **R0=R_air/10**

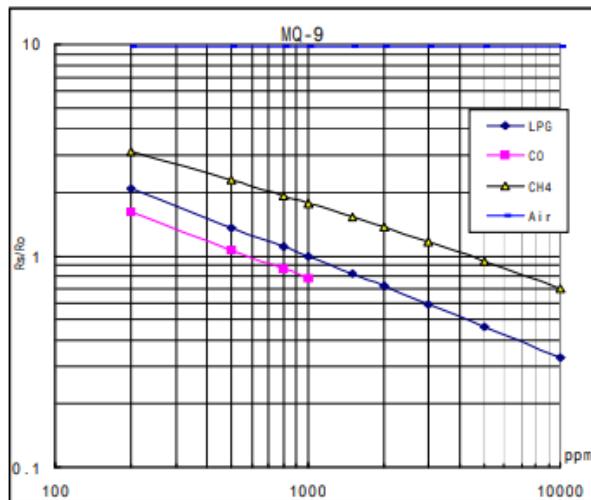


Figure 3.7: Courbe d'étalonnage de capteur MQ9

A ce stade, nous pouvons déposer le capteur dans n'importe quel environnement et calculer le ratio R/R0 (sachant que **R= (3.3-tension/tension)**). De plus, en nous référant aux courbes (figure 3.7), on a calculé l'équation de la droite linéaire « ax+b » de la concentration du gaz LPG en fonction de RS/R0 et on obtient l'équation suivante :

$$X = -(Y - 1.3) / 0.0003$$

Avec :

X : ppm « concentration de gaz LPG »

Y : RS/R0 « R0 une constante, RS résistance variable sensible de l'ensemble des gaz »

Etant donné que l'ESP32 admet 18 convertisseurs analogiques numériques avec une résolution de 12 bits, on a utilisé le pin 34 (**ADC1-6**) pour la lecture des valeurs analogiques. Après conversion les valeurs mesurées en « **ppm** » sont publiées au broker (l'adresse IP de la carte Raspberry) à travers le protocole MQTT où le topic est « **gaz/ppm** ».

```
client.publish("temp", (String)Temperature);
client.publish("hum", (String)Humidity);
client.publish("gas/ppm", (String)ppm); // Line highlighted with a red box
client.publish("smoke", (String)smoke);
```

Figure 3.8 : Envoi de valeur de gaz vers firebase

3.3.2 Mise en marche du capteur de fumée MQ2

Dans cette partie nous branchons le capteur MQ-2 avec la carte ESP32 pour la détection de fumée comme la présente la figure 3.9 :

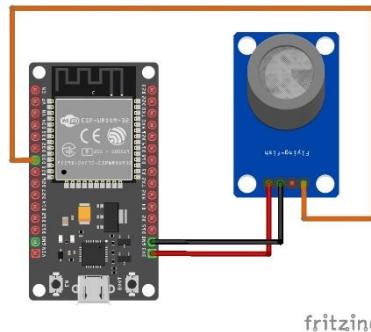


Figure 3.9: Capteur MQ2 avec la carte ESP32

Branchement (Tableau 3.2):

Capteur MQ-2	Carte ESP32
VCC	3.3v
GND	GND
A0	Pin 33
D0	NC

Tableau 3.2 : Branchement de capteur MQ-2 avec ESP32

Etape de mesure de fumée :

Le Capteur de fumée MQ2 appartient à la famille des capteurs MQ, ce pour cela les étapes de mesure et de calcule sont similaires aux calculs faits pour le capteur MQ9

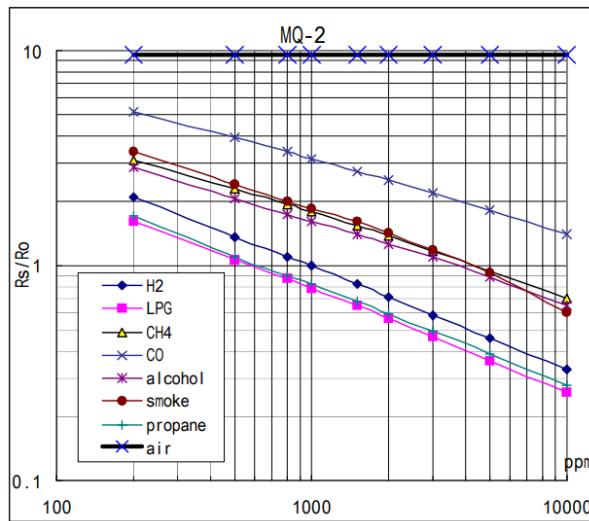


Figure 3.10: Courbe d'étalonnage du capteur MQ2

Selon le diagramme donné du capteur MQ2 on a calculé l'équation de la droite linéaire « **ax+b** » de fumée en fonction de RS/R0 :

$$X = - (Y - 1.2) / 0.00006$$

Avec :

X : ppm « concentration de gaz LPG »

Y : RS/R0 « R0 une constante, RS résistance variable sensible de

Comme on a fait pour le capteur MQ9 ce dernier a été branché au pin 33(**ADC1 CH5**) pour la lecture des valeurs analogiques, après la conversion les valeurs issues de ce capteur sont publiées au broker sur le topic « **smoke** ».

```
client.publish("temp", (String)Temperature);
client.publish("hum", (String)Humidity);
client.publish("gas/ppm", (String)ppm);
client.publish("smoke", (String)smoke);
```

Figure 3.11: L'envoie de valeur de fumée vers firebase

3.3.3 Mise en marche du capteur de flamme

Ce capteur lorsqu'il détecte une flamme, il déclenche un front montant sur le pin D4 à laquelle il est connecté comme le montre la figure suivante :

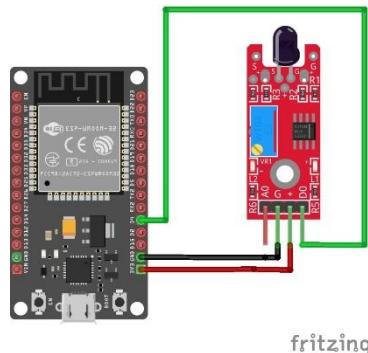


Figure 3.12: Branchement du capteur de flamme avec ESP32

3.3.4 Mise en marche du capteur DHT22

D'abord nous avons installé la bibliothèque <DHT.h> sur Arduino IDE, puis nous avons branché le capteur à la pin **D13** de la carte ESP32 comme suit (figure 3.13).

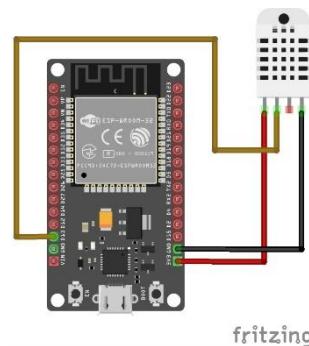


Figure 3.13: Branchement de capteur DHT22 avec la carte ESP32

Après la réalisation du montage et l'écriture de code convenable le résultat est affiché au moniteur série comme suit (figure 3.14):

```
Humidity: 70.60% Temperature: 26.50°C 79.70°F
Humidity: 70.60% Temperature: 26.50°C 79.70°F
```

Figure 3.14 : le résultat sur le moniteur série

3.3.5 Mise en marche de l'afficheur LCD

Pour cette partie, nous utiliserons un écran LCD 16×2 I₂C, l'avantage d'utiliser un LCD I₂C est que le câblage est vraiment simple. Nous avons juste besoin de câbler les broches SDA et SCL. Le rôle d'afficheur LCD dans notre projet est d'afficher la valeur de la température et de l'humidité.

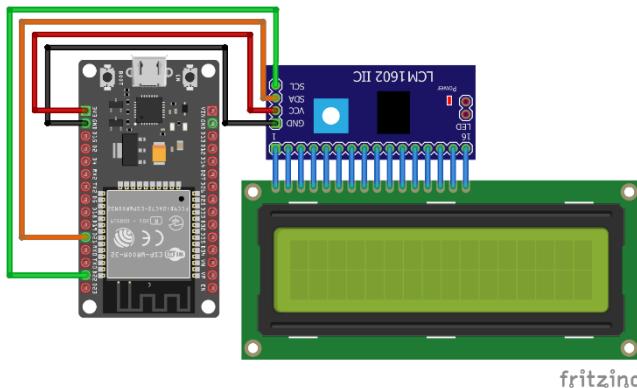


Figure 3.15: Branchement d'afficheur LCD avec la carte ESP32 via l'adaptateur I₂C

3.5 Implémentation de l'unité d'émission/réception

3.5.1 Esp32 en tant que « publisher »

L'objectif de cette partie est de présenter comment publier un message sur un « **topic** » **MQTT**, en utilisant ESP32 et les bibliothèques Arduino IDE. Pour l'ESP32, nous allons utiliser une bibliothèque **MQTT** appelée « **MQTTClient.h** ».

```
#include <MQTT.h>
#include <MQTTClient.h>
#include <WiFi.h>
```

Figure 3.16 : Bibliothèque MQTT

La façon la plus simple d'installer la bibliothèque est d'utiliser le gestionnaire de bibliothèque Arduino IDE, comme le montre la figure 3.17 :

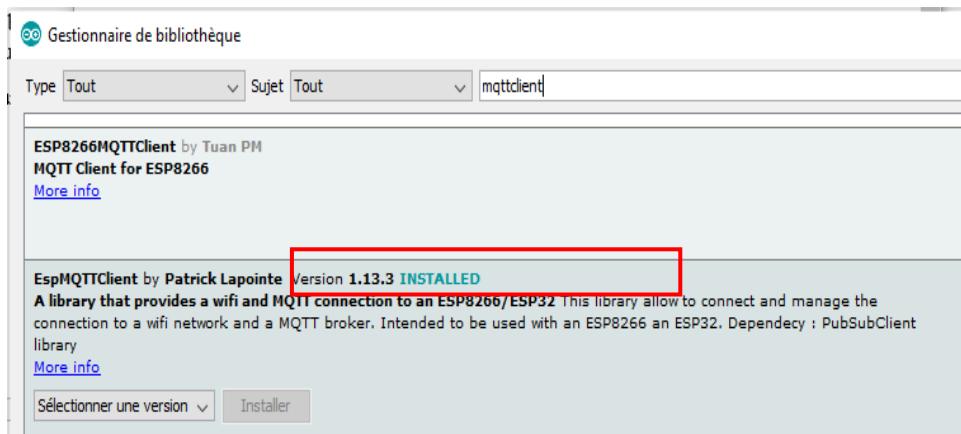


Figure 3.17: gestionnaire de bibliothèque Arduino IDE

Ainsi, nous devons inclure certaines bibliothèques dans notre code. Nous avons besoin de la bibliothèque WiFi appelé « **wifi.h** », afin de pouvoir connecter l’ESP32 à un réseau WiFi

```
#include <MQTT.h>
#include <MQTTCClient.h>
#include <WiFi.h>
```

Figure 3.18: Bibliothèque Wifi

Nous déclarons les identifiants et les informations nécessaires pour se connecter au réseau WiFi, nous avons besoin de l’SSID et du mot de passe.

Comme d’habitude, nous lançons le moniteur série pour afficher et suivre l’évolution de notre programme. Nous nous connectons également au réseau WiFi, afin que nous puissions atteindre le courtier MQTT et publier des messages.

Ensuite, il faut spécifier l’adresse et le port du serveur MQTT.

```
IPAddress ipaddress(192, 168, 43, 246);
WiFiClient net;
MQTTCClient client;
```

Enfin, pour publions les valeurs des capteurs sur un « **Topic** », nous appelons la méthode « **Publish** », en passant comme paramètres d’entrée le nom du « **Topic** » et les valeurs à publier, entant que chaîne de caractères.

```
client.publish("temp", (String)Temperature);
client.publish("hum", (String)Humidity);
client.publish("gas/ppm", (String)ppm);
client.publish("smoke", (String)smoke);
```

3.5.2 Configuration de la Raspberry PI en tant que « subscriber»

❖ Installation et mise en marche de la carte raspberry pi

La carte Raspberry utilise comme support la mémoire la carte SD. Cette carte contient le système d'exploitation ainsi que tous les programmes ou fichiers personnels que nous souhaitons stocker dessus. Nous avons utilisé une carte de type MicroSD de classe 10 ou plus et de taille 32 GB pour garantir une rapidité de réponse. Pour installer le système nous avons utilisé un logiciel SD CardFormatter comme la montre la figure 3.19.

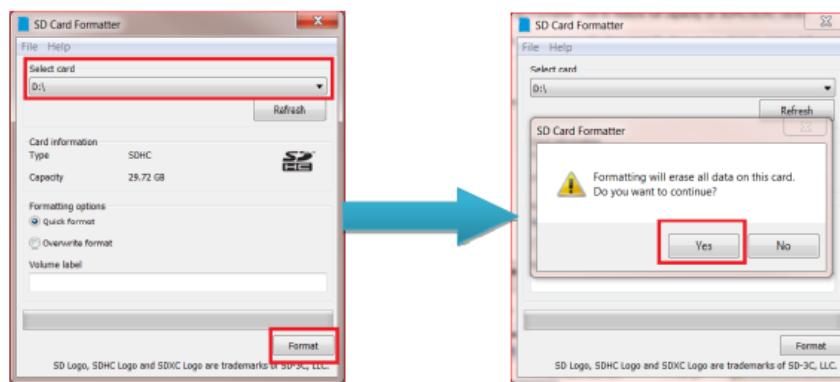


Figure 3.19 : Formatage de la carte microSD

Nous passons maintenant à l'installation du système. Nous avons installé tout d'abord Raspberry PI Imager. Ensuite, nous avons choisi le système d'exploitation convenable et le périphérique que nous avons gravé dessus et enfin cliquer sur **ÉCRIRE** comme le montre la figure 3.20.



Figure 3.20 : Raspberry PI Imager

Pour accéder à la carte Raspberry à distance d'après un ordinateur connecté sur le même réseau local il faut activer le protocole SSH.

Ensuite nous cherchons l'adresse IP de la carte Raspberry en écrivant '**ifconfig**' dans le LXTerminal du système d'exploitation. La connexion entre la carte et le PC se fait grâce au logiciel **VNC Viewer** qui est préinstallé sur le Raspbian. Finalement l'écran de la carte Raspberry est partagé sur notre PC.

❖ ***Installation et mise en marche du logiciel Node-RED :***

Pour installer **Node-RED** il faut entrer la commande suivante :

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-
installers/master/deb/update-nodejs-and-nodered)
```

Puis, nous redémarrons notre Pi pour que le démarrage automatique prenne effet :

```
pi@raspberry:~ $ sudo reboot
```

Lorsque Notre Pi est de retour, nous pouvons tester l'installation en saisissant l'adresse IP de notre Pi dans un navigateur web suivi du numéro de port **1880** :

```
http://192.168.43.246:1880
```

❖ ***Installation et mise en marche d'un broker Mosquitto :***

Etape	Commande	Description
1	pi@raspberry:~ \$ sudo apt update && sudo apt upgrade	mettre à jour le système
2	pi@raspberry:~ \$ sudo apt install -y mosquitto mosquitto-clients	Installer mosquitto broker
3	pi@raspberry:~ \$ sudo systemctl enable mosquitto.service	Démarrage automatique de Mosquitto au démarrage de Raspberry Pi
4	pi@raspberry:~ \$ mosquitto -v	tester la version de mosquitto

Tableau 3.3: Installation et mise en marche d'un broker Mosquitto

La dernière commande affiche la version installée et la figure 3.21 affiche le rendu de cette commande.

```

pi@raspberrypi: ~
Setting up libdlt2:armhf (2.18.6-1) ...
Setting up libwebsockets16:armhf (4.0.20-2) ...
Setting up mosquitto (2.0.11-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service →
/lib/systemd/system/mosquitto.service.
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rpil+deb11u2) ...
pi@raspberrypi:~ $ sudo systemctl enable mosquitto.service
Synchronizing state of mosquitto.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mosquitto
pi@raspberrypi:~ $ mosquitto -v
1639226740: mosquitto version 2.0.11 starting
1639226740: Using default config.
1639226740: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1639226740: Create a configuration file which defines a listener to allow remote
access.
1639226740: For more details see https://mosquitto.org/documentation/authenticat
ion-methods/
1639226740: Opening ipv4 listen socket on port 1883.
1639226740: Error: Address already in use
1639226740: Opening ipv6 listen socket on port 1883.
1639226740: Error: Address already in use

```

Figure 3.21 : Rendu de commande Mosquitto -v

Pour activer l'accès à distance afin de pouvoir communiquer avec d'autres appareils IoT, nous devons modifier/créer un fichier de configuration selon les étapes suivantes (tableau 3.4)

étape	Commande	Description
1	pi@raspberry:~\$ sudo nano /etc/mosquitto/mosquitto.conf	ouvrir le fichier « mosquitto.conf »
2	listener 1883 allow_anonymous true	à la fin du fichier ajouter les lignes suivantes
3	pi@raspberry:~\$ sudo reboot	Redémarrage de la carte raspberry pi

Tableau 3.4 : Les étapes d'activation l'accès à distance

Après avoir installé **Node-RED** et mosquitto broker, nous pouvons maintenant tester notre MQTT « **subscriber** »

❖ **Raspberry pi en tant que « subscriber »**

La carte Raspberry pi en tant que Broker offre plusieurs services la figure 3.22 présente le diagramme de flux des données régit par l'environnement Node-RED.

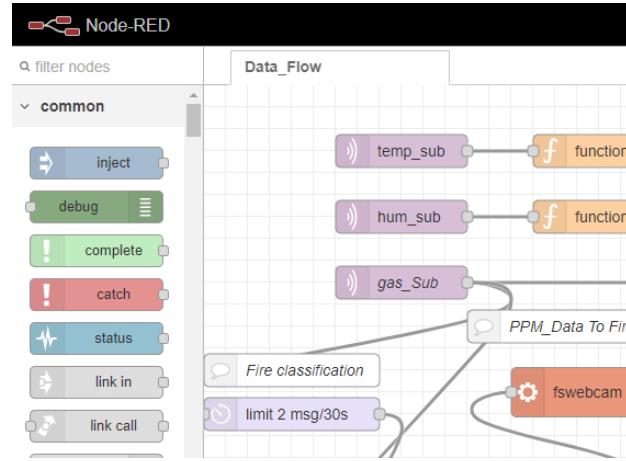


Figure 3.22 : Diagramme de flux des données

Les flows des données commencent par l'écoute du serveur MQTT pour recevoir les données envoyées par les capteurs, comme le montre la figure 3.23

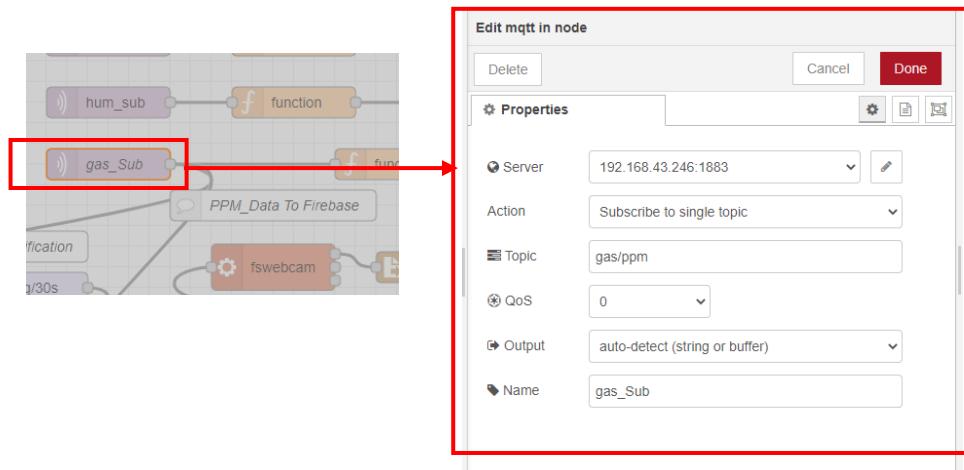


Figure 3.23 : Configuration de node MQTT

3.6 Unité de détection d'incendie par vision artificielle

Dans cette partie nous présentons les flows des données qui englobent le traitement d'images pour la détection d'un foyer de flamme à l'aide d'une caméra Raspberry pi (figure 3.24).

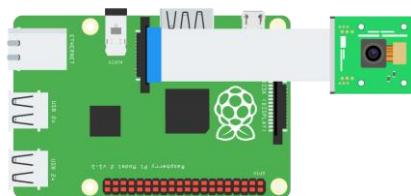


Figure 3.24 : Branchement de Camera avec la carte Raspberry Pi

3.5.1 Transfer Learning « apprentissage par transfert »

Le Transfer Learning, ou « apprentissage par transfert », désigne les techniques permettant de transférer les connaissances de plusieurs sources vers d'autres problèmes à régler. Il s'agit des compétences apprises à partir de tâches antérieures servant à améliorer de nouvelles tâches similaires. [8]

L'intelligence artificielle (IA), et plus spécifiquement le Deep Learning, nécessite un grand nombre de données pour faire fonctionner à leur meilleur niveau les réseaux de neurones artificiels. Pour améliorer la qualité de l'apprentissage d'un logiciel, il est possible de mutualiser des savoirs d'un modèle à un autre.

Le Transfer Learning permet de réaliser des bénéfices importants, en termes de temps et de techniques d'apprentissage. Une base de données gratuite sur le web va enrichir et améliorer la qualité des réseaux de neurones d'un logiciel.

➔ Il est possible de faire du Transfer Learning avec ses différentes techniques « **Tensorflow, Pytorch, Keras** » de Deep Learning afin d'améliorer leur capacité. [9]

3.5.2 Classifications des images en utilisant « Tensorflow »

Dans cette partie on a utilisé le modèle pre-trained « MobileNet » créé par google, MobileNet est une architecture de réseau efficace et un ensemble de deux hyper-paramètres pour construire de très petits modèles à faible latence qui peuvent être facilement adaptés aux exigences de conception pour les applications de vision mobile et embarquée.

Et pour rendre le modèle plus rapide et plus accessible on a utilisé l'outil « Teachable machine », dans cet outil, il y a trois types de projets parmi lesquels nous pouvons choisir : les images, sons et postures, comme l'indique la figure 3.25

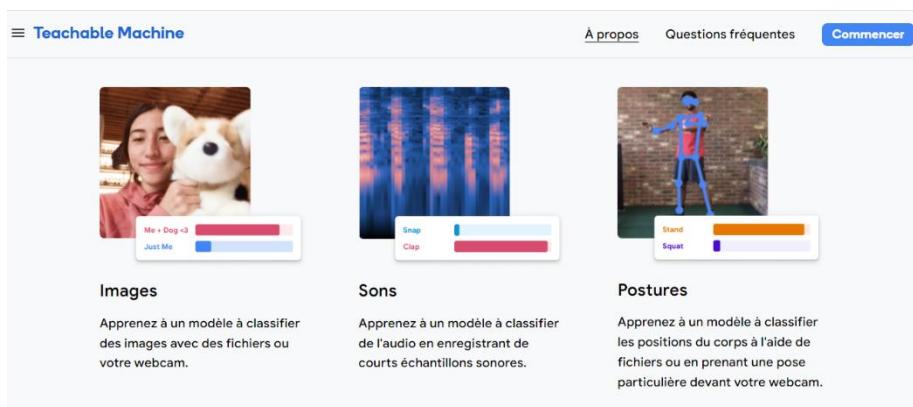


Figure 3.25 : Type de projet dans l'outil "Teachable machine"

Alors on a utilisé le premier type ‘image’, puis on a ajouté notre « Dataset » à classifier qui est composée de deux classes la première est un ensemble d’images des maisons, usines et des restaurants sans incendies et l’autre classe avec incendies.

- ➔ Le modèle est maintenant qualifié à prédire si une image appartient à la classe 1 ou 2 avec une précision de 90% au 100%. La figure 3.26 Présente un exemple de reconnaissance d’une flamme par ce modèle.

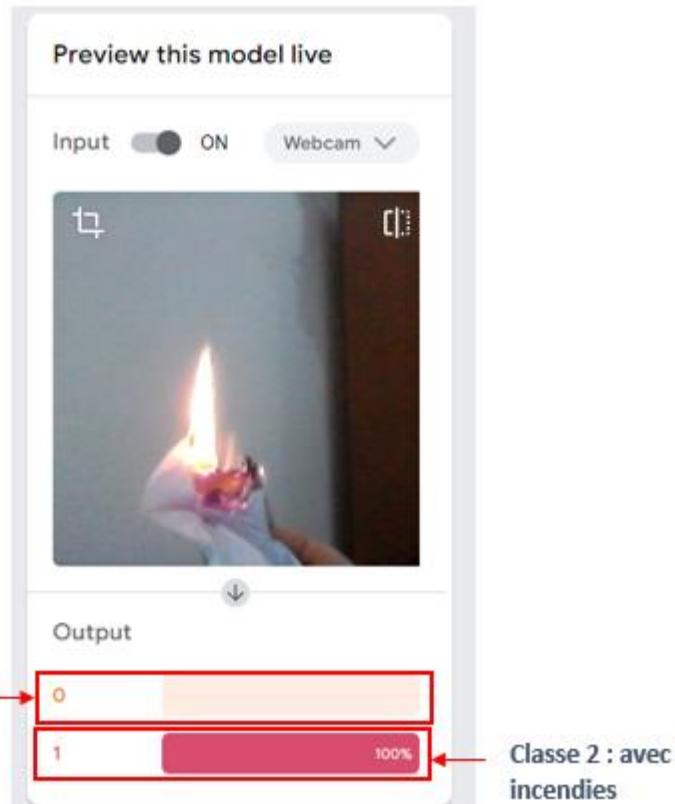


Figure 3.26 : Exemple de reconnaissance d'une flamme

3.5.3 Intégration du modèle de reconnaissance de flamme dans Node-RED

Après avoir reçu les valeurs des données de concentration de gaz (GPL) et de fumée. La caméra est activée si une condition est validée sur la concentration de GPL et la valeur de

fumée dans l'aire pour capturer des images (voir figure 3.27) et les envoyer vers le modèle de classification des images « Tensorflow » que nous avons présenté précédemment.

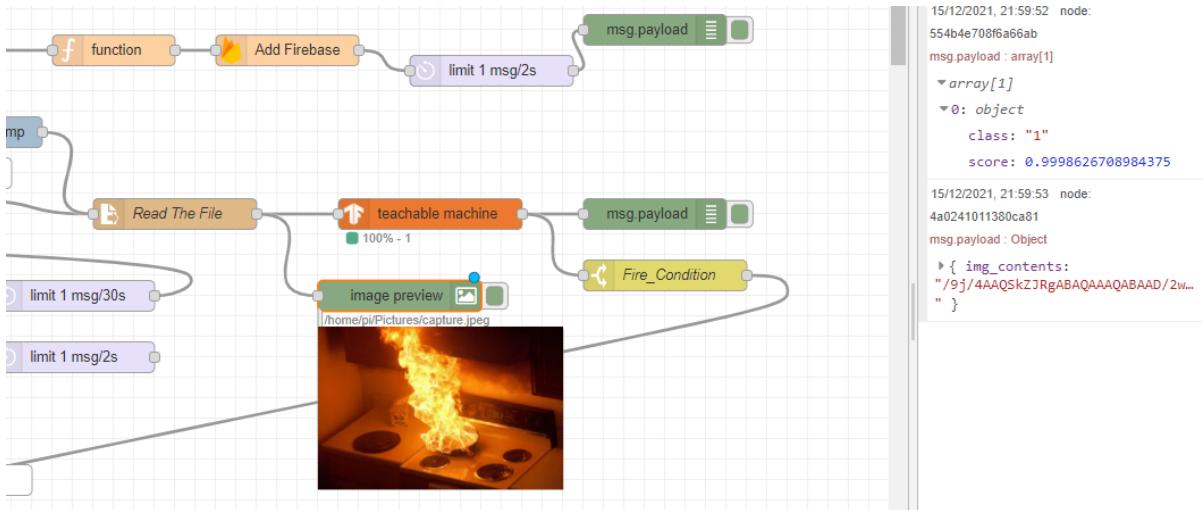


Figure 3.27 : Modèle Tensorflow dans Node-Red

Le modèle computer vision va décider la présence d'un incendie ou non, si oui le système convertit l'image capturée de l'incendie sous forme ‘base64’ (figure 3.28) puis l'envoie vers le cloud puis l'application Android et la plateforme web la récupère.



Figure 3.28 : Conversion d'image à base64 et l'envoi vers le Cloud



Figure 3.29 : Format d'image dans le Cloud Firebase

3.7 Unité de protection contre les incendies

Les accidents en cas de fuite de gaz sont le plus souvent l'intoxication, l'asphyxie et l'explosion, ce qui entraîne des risques de brûlures graves. Ce pour cela que nous avons doté notre système des mécanismes permettant de protéger le lieu contre les conséquences de la fuite de gaz. Ainsi, cette solution de protection se compose d'un système de ventilation et un autre pour la coupure de gaz.

3.6.1 Système de ventilation

Le ventilateur est activé si une condition est validée sur la concentration de gaz. Ce dernier est commandé par la carte Raspberry pi via un relais comme le montre la figure 3.30 et le Tableau 3.5

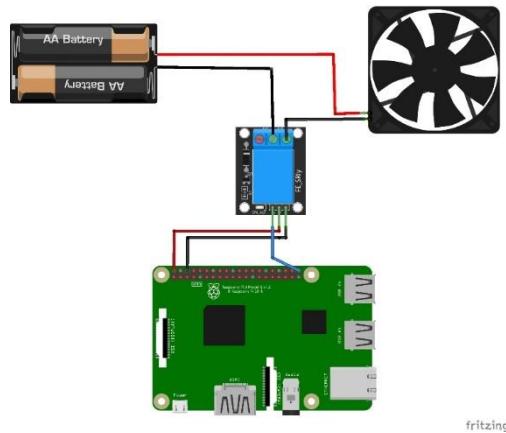


Figure 3.30 : Branchement de ventilateur avec la carte Raspberry Pi

Ventilateur	Relais	Raspberry Pi
-	VCC	5V
-	GND	GND
-	IN1	GPIO 21
VCC	NO (naturel open)	-
GND	COM	-

Tableau 3.5 : Brochage de ventilateur avec la carte Raspberry pi

Pour contrôler notre ventilateur on a utilisé un système qui puisse à la fois délivrer suffisamment de courant et être contrôlé par une sortie de la carte Raspberry pi, dans notre projet on a utilisé un relais qui permet l'ouverture ou la fermeture d'un circuit électrique de puissance à partir d'un ordre émis par la carte Raspberry Pi selon le schéma de la figure

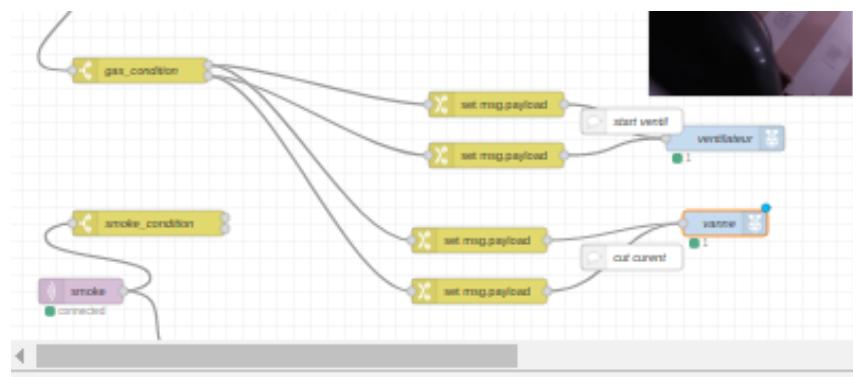


Figure 3.31: L'ouverture et la fermeture de ventilateur

3.6.2 Dispositif de coupure de l'arrivée de gaz

Afin de rester en sécurité en cas de fuite de gaz, on a envisagé d'installer un dispositif pour couper l'arrivée de gaz en cas de fuite. Ce système se compose d'une électrovanne qui se situe dans l'installation de gaz de ville comme indiqué dans la figure 3.32.

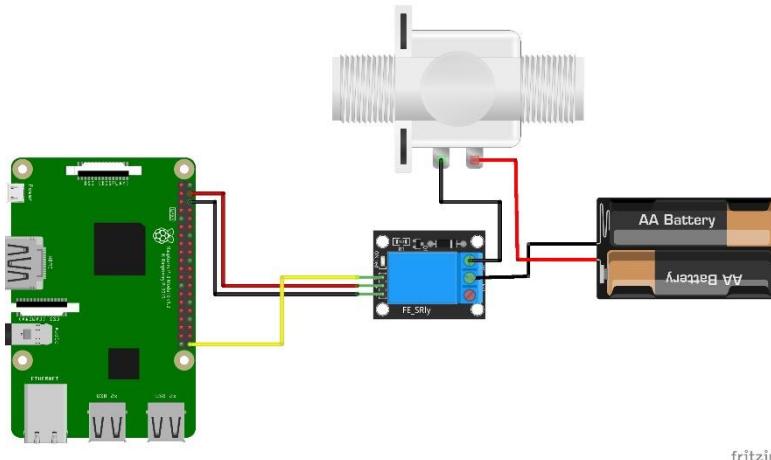


Figure 3.32: système de coupure de gaz

En effet, les électrovannes sont des vannes électriques à 2 positions. Elles sont utilisées seulement pour l'ouverture et la fermeture d'un réseau. Dans notre solution on a pensé à utiliser une électrovanne de gaz à un thermostat de sécurité pour assurer l'arrêt et la circulation du gaz, comme elle n'est pas disponible actuellement nous l'avons remplacée par une électrovanne d'eau dans notre prototype.

3.8 Unité d'alerte

Si un des capteurs détecte la présence de fumées, monoxyde de carbone ou une fuite de gaz un ensemble des actions d'alerte se déclenche pour signaler la présence d'une fuite de gaz ou début de feu.

- ✓ Alerte visuelle et sonore : représentée sous forme d'une combinaison sonore et lumineuse
- ✓ Alerte par notification: Si la concentration de gaz LPG est plus élevée une notification était déclenchée sur l'application mobile pour notifier l'utilisateur qu'il y a une fuite de gaz (figure 3.32).

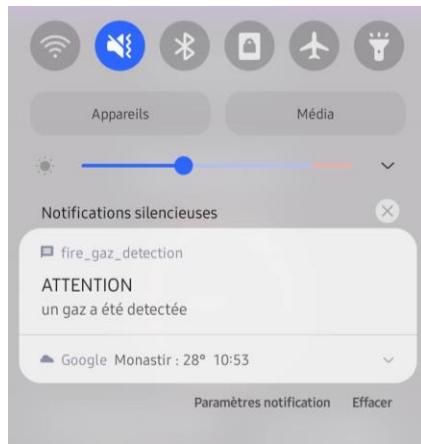


Figure 3.33: Notification d'alerte de l'application mobile

- ✓ Alerte par SMS : il s'agit d'une alerte déclenché sous forme d'un SMS pour éviter les problèmes de joignabilité (figure 3.34).



Figure 3.34 : SMS d'alerte

3.9 Développement de la plateforme Web

Dans cette partie nous illustrons les différentes fonctionnalités mises en œuvres à travers des imprimés écrans de notre plateforme Web.

3.9.1 Mode citoyen

❖ Interface Principale

C'est la première interface d'interaction entre l'utilisateur et la plateforme web, nous avons prévu une simple interface facile à utiliser.



Figure 3.35 : Interface principale

Bouton	Description
Home	Permet à l'utilisateur d'accéder à l'interface d'authentification
About Us	Offre à l'utilisateur des informations sur le système
Contact Us	Permet à l'utilisateur de contacter la propriétaire de site
Sign Up	Offre à l'utilisateur de s'inscrire dans la plateforme

Tableau 3.6: Tableau des boutons de l'interface principale

❖ *Interface d'authentification*

La figure suivante présente l'interface "S'authentifier". Elle assure la sécurité de notre application à travers l'authentification obligatoire de chaque utilisateur.

Nous avons prévu une simple interface de connexion qui permet à l'utilisateur de s'authentifier et d'accéder à l'interface de consultation de la plateforme "Dashboard" à travers son login et son mot de passe. En cliquant sur le bouton "Login", si les coordonnées de notre utilisateur sont justes, alors il accède à l'application; sinon un message d'erreur sera affiché.

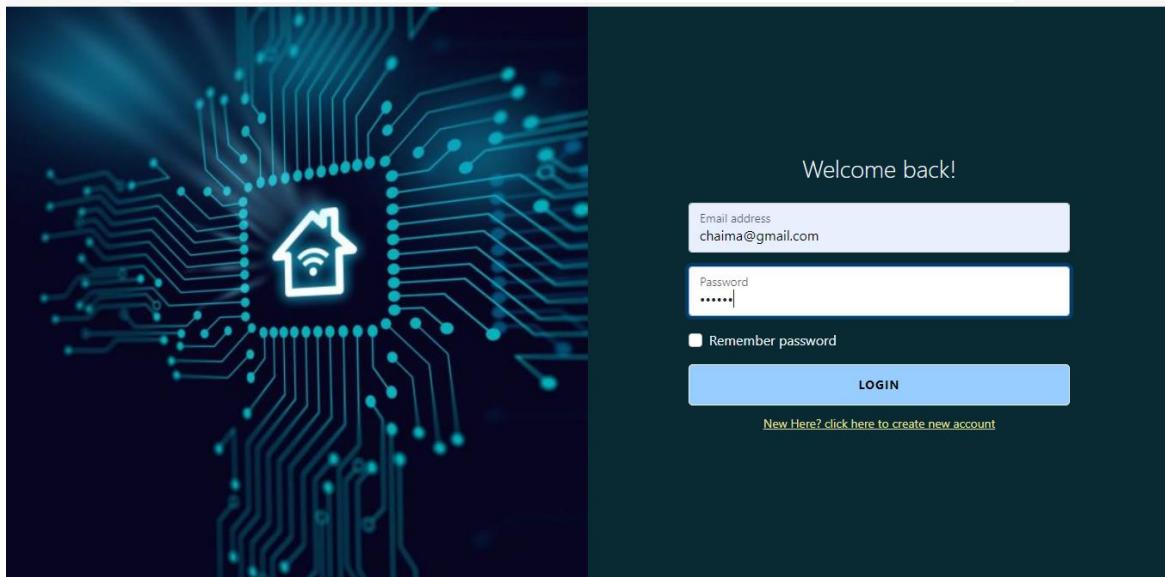


Figure 3.36: Interface d'authentification

❖ *Interface d'inscription*

L'interface de l'inscription permet aux nouveaux utilisateurs de faire un nouveau compte en remplissant ces informations nécessaires : Nom/prénom, email, etc (figure 3.37).

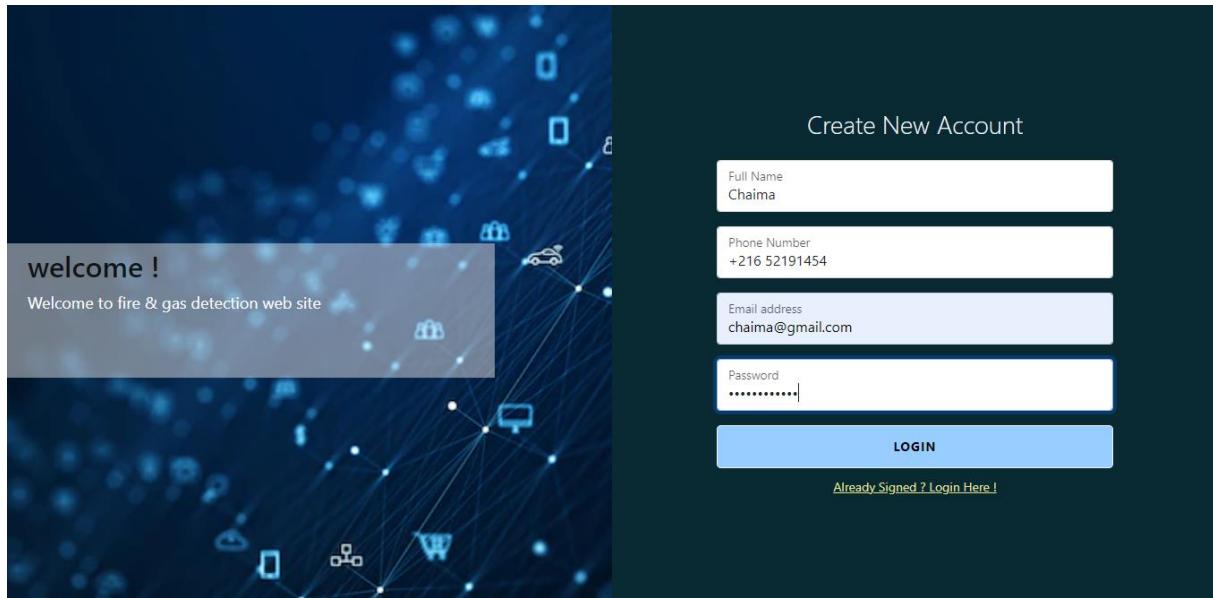


Figure 3.37: Interface d'inscription

❖ *Interface de consultation des données*

Une fois authentifié en tant qu'citoyen, il est redirigé vers son propre tableau de bord qui contient différentes interfaces comme le montre la figure 3.38. Elle nous présente un tableau de différentes mesures (concentration de gaz, Température, Humidité) et un diagramme de l'historique qui permet à l'utilisateur de consulter les valeurs issues des jours précédents et la possibilité de se déconnecter "LogOut".

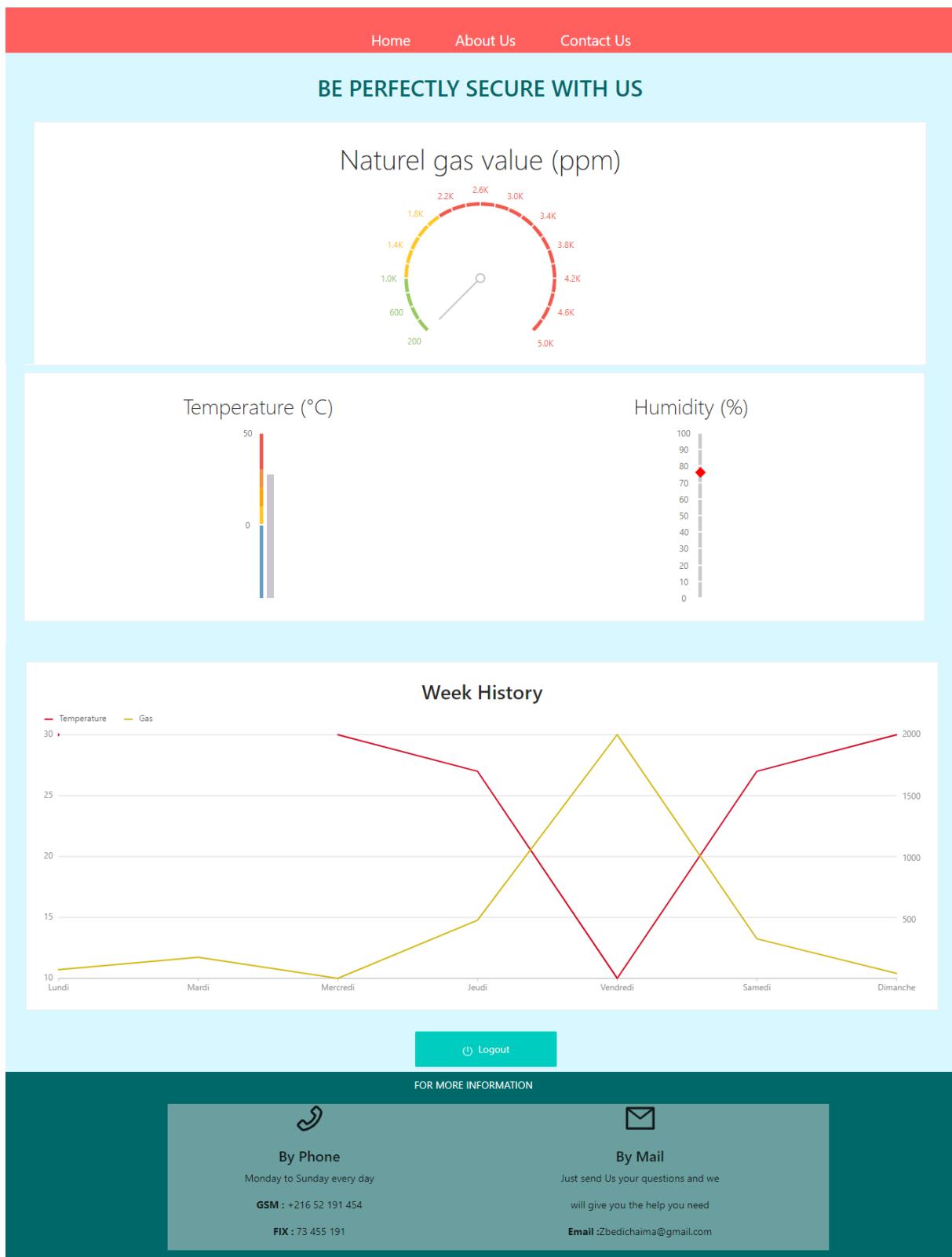


Figure 3.38: Tableau de bord de consultation pour un citoyen

3.9.2 Mode Protection Civile

❖ Interface d'authentification

C'est la première interface d'interaction entre l'agent de protection civile et la plateforme, l'interface d'authentification permet à l'agent de protection civile d'accéder au tableau de bord après le saisi du numéro de série et l'email (figure 3.39).

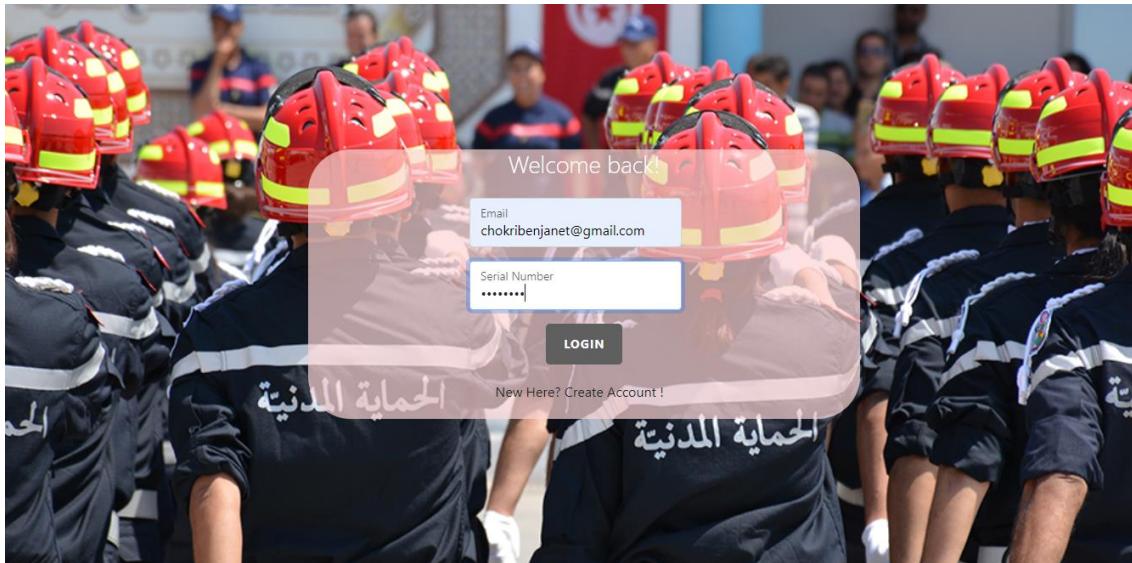


Figure 3.39: L'interface d'authentification pour un agent de protection civile

En cliquant sur le bouton "Login", si les coordonnées de notre utilisateur sont justes, alors il accède à l'interface suivante, sinon un message d'erreur sera affiché.

❖ Interface de consultation

Une fois authentifié, l'agent va accéder à la page de consultation présentée dans la Figure 3.40. Elle nous présente un tableau de différentes notifications reçues :

- ↳ **Coordonnées de citoyen** : afficher le nom & prénom de l'utilisateur avec son numéro de téléphone et la localisation
- ↳ **Données de système** : la valeur de température/humidité et la valeur de concentration de gaz pour connaître la source de feu.
- ↳ **L'image de l'incendie** : c'est l'image capturée par la caméra de système qui indique la présence de feu
- ↳ **Localisation Google Maps** qui permet à l'agent de visualiser la position de sur Google maps.

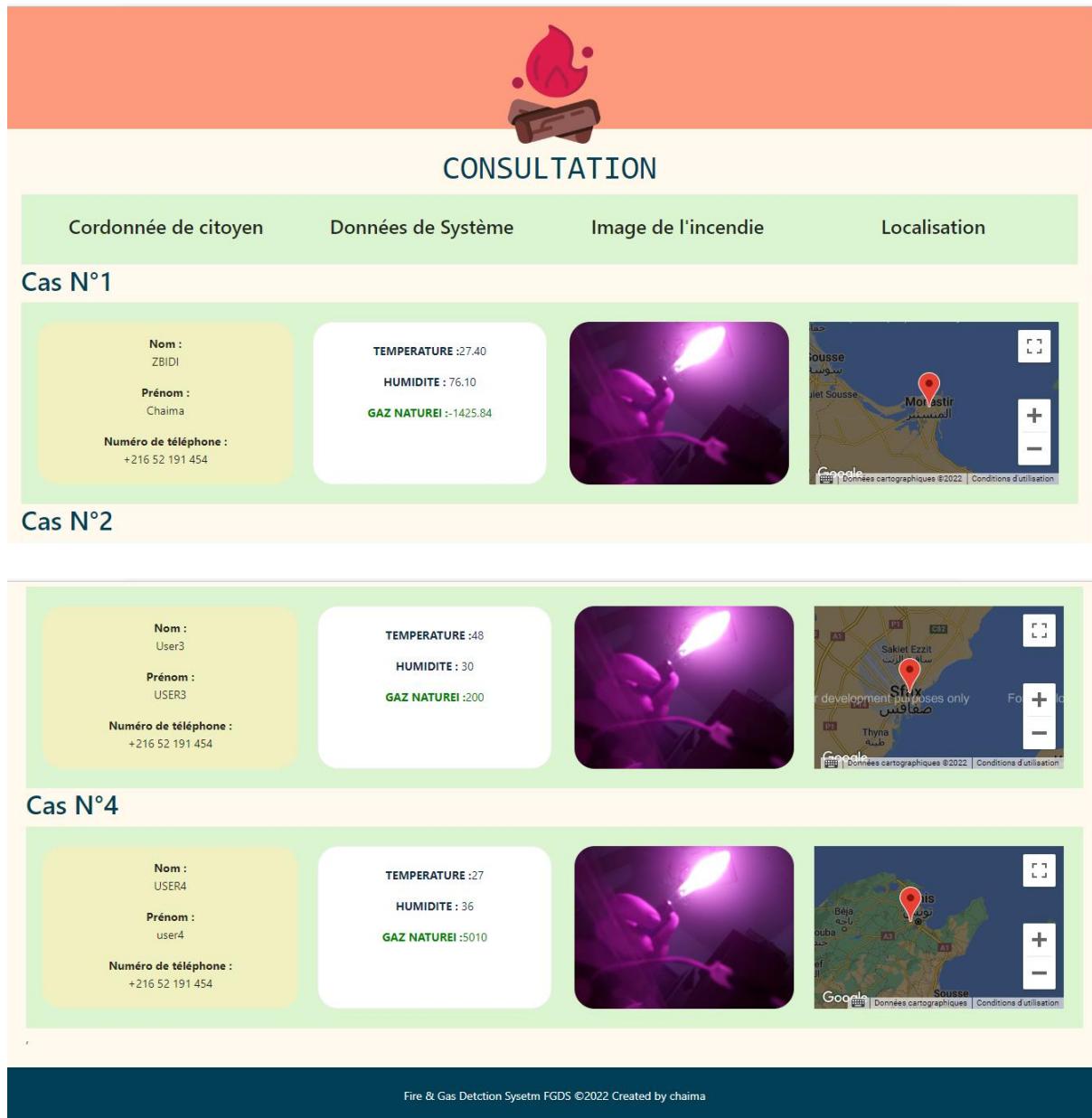


Figure 3.40 : Tableau de bord d'un agent de protection civile

3.10 Développement de l'application Android

Dans cette partie, nous explicitons la partie de développement de notre application Android. A cet égard, nous donnerons un aperçu des interfaces de l'application accompagné d'un descriptif et une partie traitement pour l'explication de « back end ».

3.10.1 Partie graphique

3.10.1.1 Interface principale

C'est la première interface d'interaction entre l'utilisateur et l'application, cette première interface nous amène vers deux autres interfaces à l'aide de deux boutons « user » et « protection » comme l'indique la figure suivante.



Figure 3.41 : Interface Principale d'application mobile

Ces deux boutons permettent l'accès aux interfaces respectives pour le citoyen et pour l'agent de protection civile.

3.10.1.2 Mode citoyen « utilisateur normal »

❖ Interface d'authentification pour le citoyen

L'interface d'authentification (figure 3.42) permet à l'utilisateur d'accéder à son compte de l'application après le saisi de l'email et de son mot de passe.



Figure 3.42: Interface d'authentification

Le bouton « Login » permet à l'usager d'accéder à l'application après la vérification des données dans la base. Un nouvel utilisateur peut cliquer sur « New Here ?, create Account » pour s'inscrire dans l'application.

❖ *Interface d'inscription*

L'interface de l'inscription permet aux nouveaux utilisateurs de faire un nouveau compte en remplissant les informations nécessaire (Nom et Prénom, Email, Adresse, etc) comme l'indique la figure 3.43.



Figure 3.43: Interface d'inscription

Le bouton « Registre » permet d'enregistrer les coordonnées d'un nouveau citoyen dans la base de données. Après la création du compte et l'authentification on peut maintenant passer à l'interface de notre système.

❖ *Interface de consultation*

C'est l'interface de consultation des données du système (figure 3.44), nous trouvons dans cette interface la valeur de température, la valeur d'humidité et la valeur de concentration de gaz.

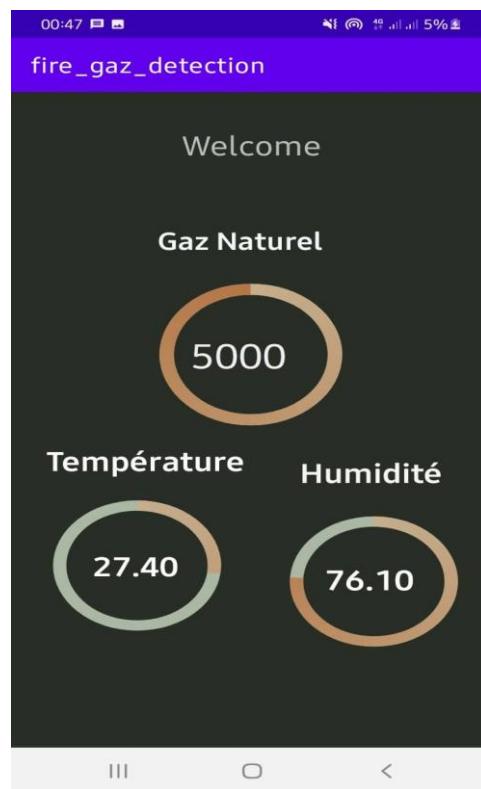


Figure 3.44 : Interface de consultation

3.9.1.3 Mode protection Civile

- ❖ **Interface d'authentification pour un agent de la protection civile**

L’interface d’authentification (figure 3.45) permet à l’agent de la protection civile d'accéder à son compte de l’application après la saisie du numéro de série et son nom.

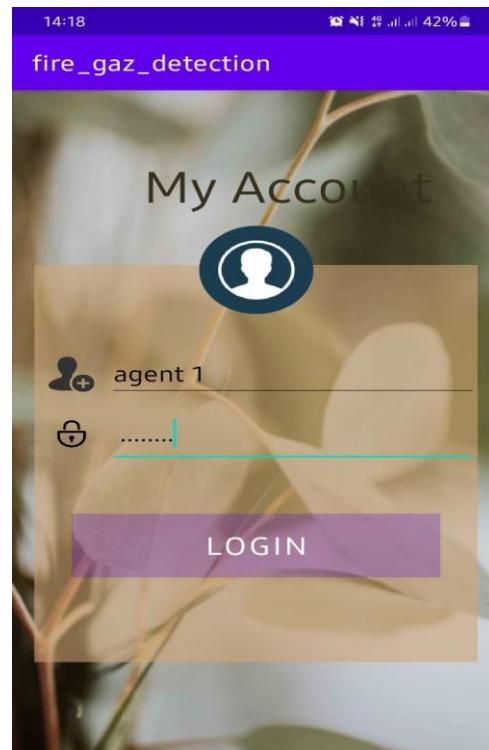


Figure 3.45: Interface d'authentification pour un agent de protection civile

Le bouton « login » permet à l'agent de protection civile d'accéder à l'interface pour consulter l'image de l'incendie et les coordonnées de l'emplacement où s'est produit l'incident (figure 3.46) après la vérification des données d'authentification.

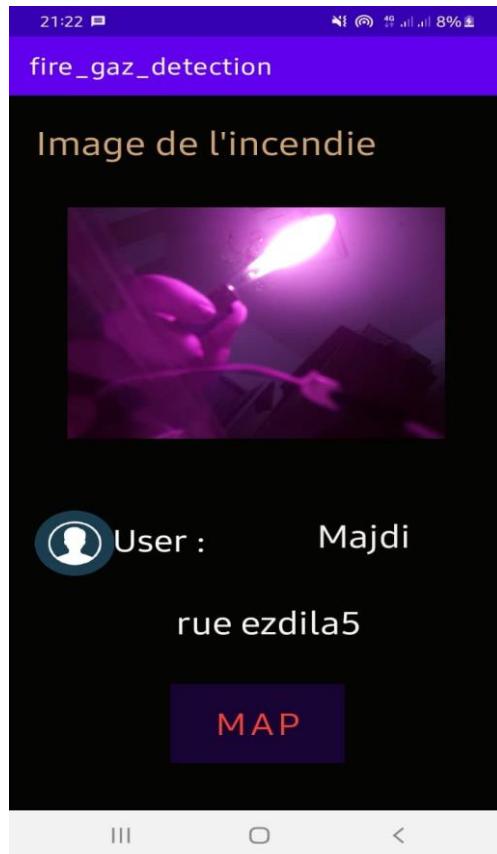


Figure 3.46 :Interface de consultation d'incendie

3.11 Partie Traitement

3.11.1 Mode citoyen

❖ *L'authentification*

L'authentification de l'utilisateur est assurée par Firebase authentication (figure 3.47) « cet outil fournit des SDK faciles à exploiter, des services back-end ou encore des bibliothèques d'interface utilisateur. Ces bibliothèques, nous permettent d'authentifier nos utilisateurs ».

The screenshot shows the Firebase Authentication console under the 'Users' tab. At the top, there's a search bar with placeholder text 'Recherchez par adresse e-mail, numéro de téléphone ou ID utilisateur'. To its right is a blue button labeled 'Ajouter un utilisateur'. Below the search bar is a table with columns: 'Identifiant', 'Fournisseurs', 'Date de création', 'Dernière connexion', and 'UID utilisateur'. The table contains four rows of user data:

Identifiant	Fournisseurs	Date de création	Dernière connexion	UID utilisateur
chaima@gmail.com	✉️	25 mars 2022	29 mai 2022	JwbJIDRLPrSxpA6S3Bve41UmTt63
zbedichaima@gmail.com	✉️	25 mars 2022	25 mars 2022	ph3yxI87hRY4s0dpqD25xk9j7Bi1
rhimmajd1@gmail.com	✉️	16 déc. 2021	16 déc. 2021	IzI1S0WU9EaYXaKHuanLdEi1zit2
user@gmail.com	✉️	11 déc. 2021	16 avr. 2022	TZGfOEzotKXqafS6oY5Y3b0faEk2

At the bottom of the table area, there are pagination controls: 'Lignes par page : 50', '1 – 4 of 4', and navigation arrows.

Figure 3.47: Authentification Firebase

❖ Récupération des données

La récupération de la concentration du gaz LPG, la température et l'humidité est assurée par « Firebase Realtime » (figure 3.48). Realtime Database « cet outil permet le stockage et la synchronisation de données.

The screenshot shows the Firebase Realtime Database console under the 'Données' tab. At the top, there's a warning message: 'Protégez vos ressources Realtime Database des utilisations abusives telles que la fraude à la facturation et le hameçonnage' with a 'Configurer App Check' button and a close 'X' button. Below the warning is a URL field containing 'https://fire-gaz-detection-default.firebaseio.com'. The main area displays a hierarchical list of data values:

```

https://fire-gaz-detection-default.firebaseio.com/
  HUMID: "76.10"
  LPG_PPM: "5000"
  SMK: "-20"
  TEMP: "27.40"
  
```

Figure 3.48: Realtime Database Firebase

Nous avons utilisé la fonction « ***OnDataChange*** » dans Android studio qui va écouter chaque changement de valeur dans la base et le récupérer.

Si la concentration de gaz LPG est supérieure à 3000 ppm (c à d 7%) une notification est envoyée pour notifier l'utilisateur qu'il y a une fuite de gaz (figure 3.49).



Figure 3.49: Notification d'alerte en cas de détection de gaz

3.10.2 Mode protection civile

❖ *L'authentification*

Concernant les agents de protection civile l'authentification est assurée par une base de données contenant les numéros de série des agents de protection civile en utilisant « Firebase authentication », nous utilisons le numéro de série en tant que mot de passe.

❖ *Récupération de l'image de l'incendie*

L'image capturés est stockée sous forme « base64 » dans Firebase Realtime comme nous avons précisé précédemment.

Lorsque une image est envoyée, une notification est déclenchée pour notifier les agents a été détectées (figure 3.50).



Figure 3.50: Notification d'alerte en cas de détection de flamme

3.12 Test de fonctionnement du système

Les tests sont des étapes primordiales à faire pour vérifier le fonctionnement de notre système, ainsi qu'à l'aide du test nous pouvons traiter certaines anomalies que nous ne pouvons pas les voir théoriquement sans test. Nous commençons par le branchement complet de différents capteurs avec la nodeMCU ESP32 et le branchement de camera et les actionneurs avec la carte raspberry pi comme montre le figure 3.51:

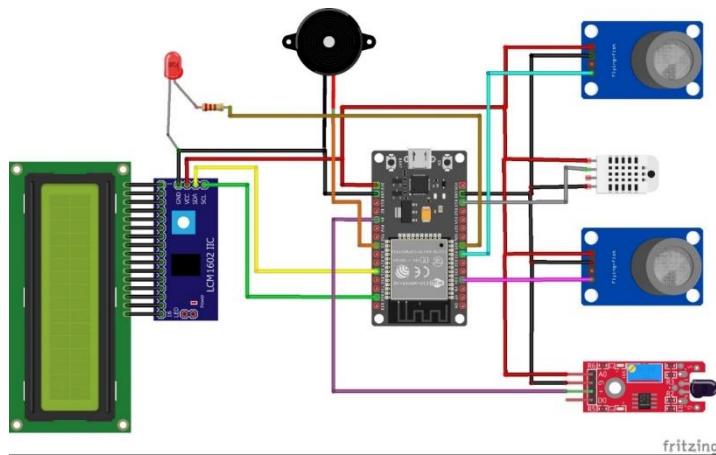


Figure 3.51 : Branchement des capteurs avec le nodeMCU

Maintenant nous allons expliquer les différents scénarios possibles pour notre système.

3.11.1 les scénarios de détection de gaz et de fumée

Condition 1 : présence de gaz avec une concentration élevée

Description : Le système est en train de surveiller les capteurs de gaz s'il détecte des taux

élevés un ensemble d'alarme est déclenché pour notifier l'utilisateur, puis une caméra sera ouverte pour capturer des images et les envoyer vers un modèle de classification d'image « Tensorflow » et le modèle computer vision va décider la présence d'incendie ou non. Si oui, le système envoie l'image capturée de l'incendie à l'agent de protection civile avec les coordonnées de l'emplacement. Ainsi que lors des détections de gaz un dispositif de protection se déclenche en coupant le gaz et en démarrant la ventilation pour éviter les risques d'étouffement ou d'explosion.

Condition 2 : présence de fumée sans gaz.

Description : Le système consulte les valeurs acquises du capteur MQ2, s'il détecte de la fumée une alarme déclenchée pour alerter l'utilisateur. À ce moment le dispositif de protection ferme le circuit.

3.11.2 les scénarios de détection de flamme

Condition : Détection des flammes dans les endroits non couverts par les caméras

Description : La détection de flamme peut se faire par traitement d'image ou par les capteurs de flamme dans les coins qui ne sont pas couverts par les caméras. En cas de détection de flamme une alarme sonore et visuelle est déclenchée.

3.13 Réalisation de la carte d'interfaçage

Le schéma de la carte d'interfaçage de la ESP32 est exposé par la figure 3.52

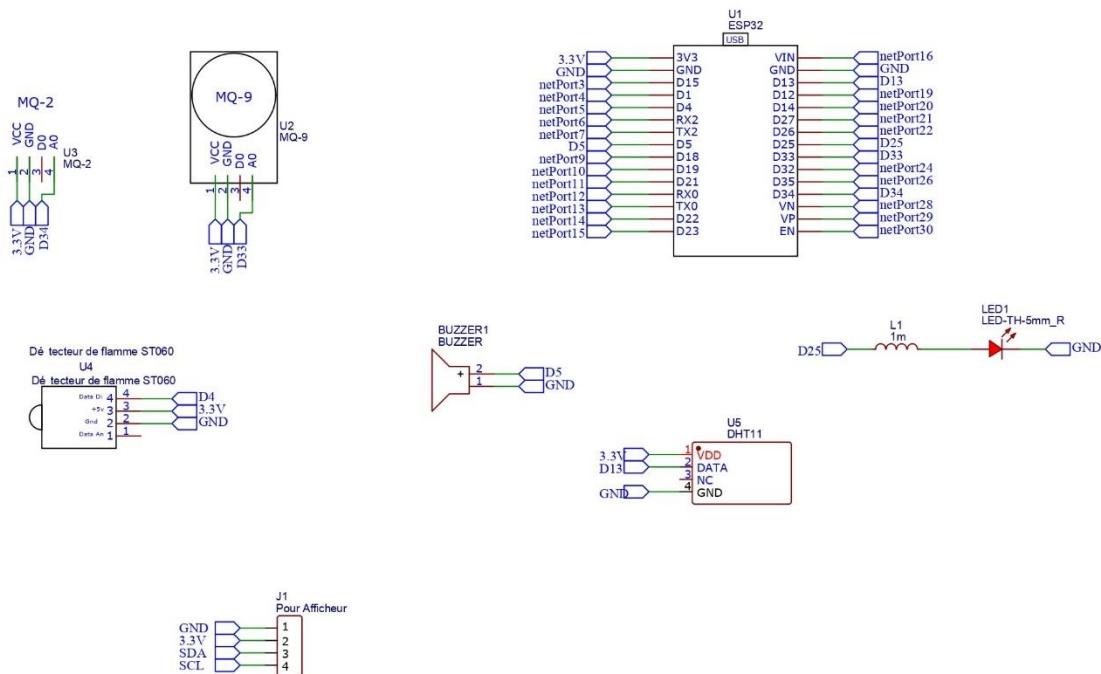


Figure 3.52 : Schéma de la carte sous EasyEDA

La figure 3.53 montre le routage de notre carte, pour améliorer la résistivité contre les perturbations externes, nous avons utilisé l'option de plan de masse

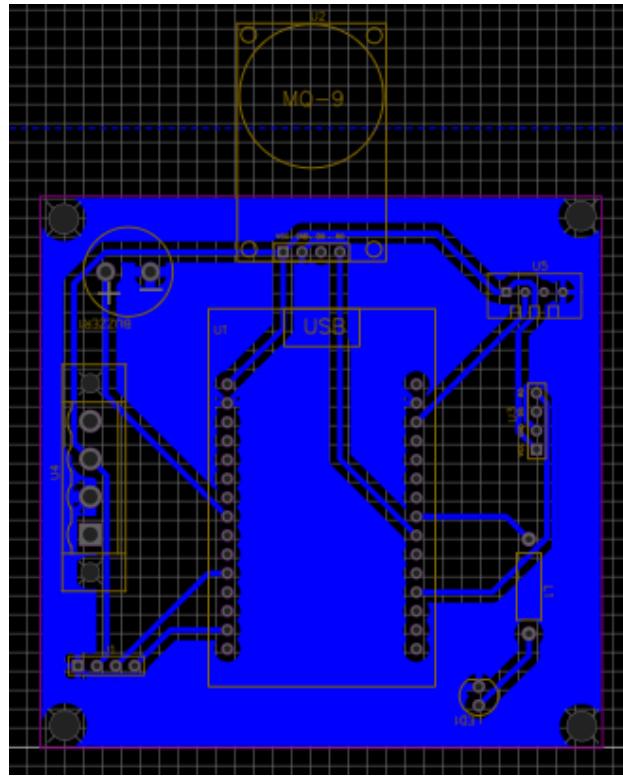


Figure 3.53 : Routage de carte électronique

La figure 3.54 montre la vue 3D de notre carte électronique

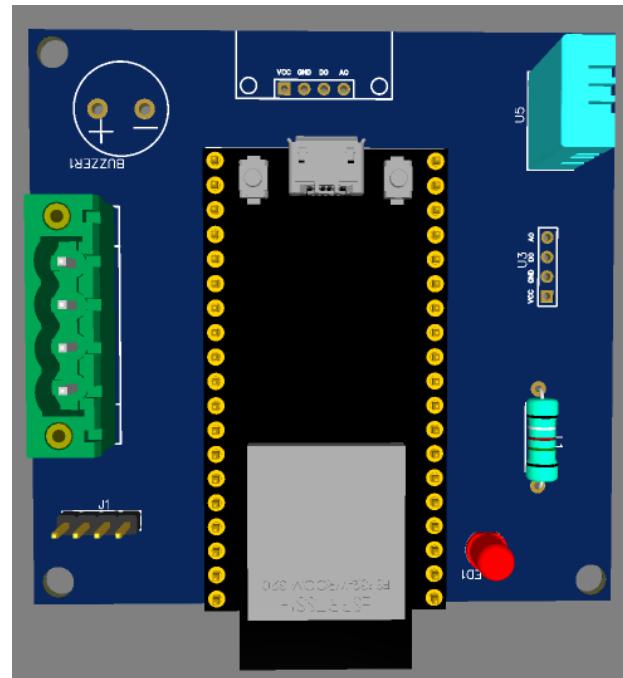


Figure 3.54 : La vue 3D de la carte électronique

Le prototype final du système de détection de gaz et de flamme par vision artificielle est illustré par la (figure 3.55 et figure 3.56)



Figure 3.55 : Prototype finale du système de détection de gaz et de flamme



Figure 3.56: Implémentation du système

3.15 Conclusion

Ce dernier chapitre a été consacré pour présenter l’application réalisée ainsi que les différents logiciels utilisés. On clôture avec la phase de tests de validation, à ce stade, nous avons réussi à atteindre nos objectifs en mettant en place un système de protection et de détection de gaz et de flamme par vision artificielle ainsi que l’émission en temps réel d’alerte en cas d’urgence en destination de l’utilisateur et la protection civile.

Conclusion générale et perspectives

Dans ce projet, notre objectif a été la réalisation d'une application pour la détection de gaz ainsi que la surveillance en temps réel de la part de l'utilisateur et la protection civile à travers une interface web et mobile avec la détection d'incendie par vision artificielle si elle est présente et de mettre en place un système de protection.

Concernant la démarche de réalisation du travail actuel, nous avons commencé tout d'abord par exposer des solutions existantes, dégager les failles et proposer la solution qu'on a apportée.

Par la suite, nous avons procédé à la conception des applications, du système ainsi que le choix des technologies à utiliser pour l'implémentation de la solution.

Ce stage nous a été très bénéfique sur le plan académique et personnel. Il nous a donné l'occasion de prendre la responsabilité pour réaliser plusieurs tâches afin d'accomplice notre projet. Il constitue désormais une expérience personnelle enrichissante et encourageante pour notre avenir.

Notre travail présente plusieurs perspectives d'amélioration. En particulier, nous souhaitons de travailler sur le protocole de communication « LoRa ».ainsi on peut au lieu de prendre des images successives on utilise un « live Stream »pour la détection de l'incendie. Et pour vraiment réaliser la notion d'alerte en temps réels il faut migrer à une infrastructure de réseaux cellulaire, évidement l'installation du réseau 5G va aider à appliquer un service fiable étant donné que le débit est énormément plus rapide.

Annexe A : Capteur DHT22

Le DHT22 comprend un composant de mesure de l'humidité de type résistif, une composante NTC de mesure de la température et un microcontrôleur 8 bits à l'intérieur, et fournit une sortie de signal numérique étalonnée. Il a une grande fiabilité et une excellente stabilité à long terme, grâce à la technique d'acquisition du signal numérique exclusif et la technologie de détection de la température et de l'humidité.

➤ Partie mesure de Température :

Pour la mesure de la température il y'a un capteur CTN ou thermistance. Cette dernière est un composant électronique dont la résistance électrique varie en fonction de la température. Les thermistances sont fabriquées par frittage de matériaux semi-conducteurs, tels que la céramique ou des polymères dans l'ordre de changement important dans la résistance à faible variation de température. Lorsque la température varie, ce qui entraîne un changement de résistance de la thermistance, d'où on trouve la température de l'environnement.

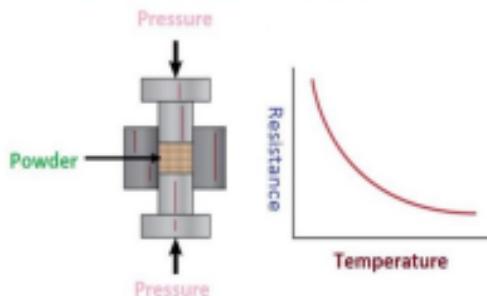


Figure 3.57 : DHT22 partie mesure de température.

➤ Partie mesure de Humidité :

Pour la mesure de l'humidité, il utilise un composant de mesure de l'humidité, qui a deux électrodes avec un substrat de maintien d'humidité entre elles. A chaque variation d'humidité dans l'environnement, la conductivité du substrat change ou la résistance entre les électrodes change. Par conséquence, on peut trouver l'humidité de l'environnement extérieur

Annexe B : Carte Raspberry pi 3

Caractéristique de la carte figure 3.39

- Alimentation à prévoir : 5 Vcc/maxi 2.5 A* via prise micro-USB (* intensité maxi si toutes les fonctions sont utilisées).
- CPU : Quad-core ARM Cortex-A53 1.2 GHz (Broadcom BCM2837)

- Processeur graphique ou GPU : Dual Core VideoCore IV Multimédia Co-Processor
- Wifi : 2,4 GHz, 802.11n (Broadcom BCM43438)
- Bluetooth : 4.1 (Broadcom BCM43438)
- Mémoire : 1 GB LPDDR2 (1024 Mo)
- USB : 4 ports USB 2.0
- Ethernet : port Ethernet 10/100 base T: RJ45
- Bus : SPI, I2C, série
- Support pour cartes : micro-SD
- Sorties audio : HDMI et jack 3,5 mm
- Sorties vidéo : HDMI
- Support des distributions : dédiées basées sur Linux et Windows 10
- Dimensions : 88 x 58 mm

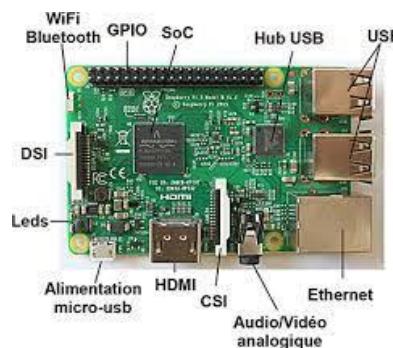


Figure 3.60: description des différents composants du Raspberry pi 3 modèle B

Dans la figure précédent on représente les composants de la carte Raspberry pi 3 modèle B. Ce qui concerne les connecteurs et ports disponible de cette carte, il est à noter que la carte Raspberry Pi donne accès à des entrées et sorties numériques appelées "General Purpose Input and Output" (GPIO) qui sont généralement contrôlées par le processeur ARM comme le montre la figure 3.40

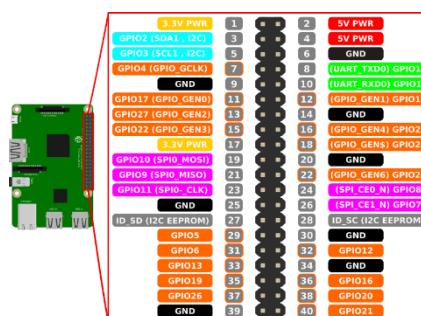


Figure 3.61 : les entrées et les sorties numériques de carte raspberry

Annexe C: La famille MQ

✓ Étalonnage de capteur MQ9

Avant d'utiliser le module, il faut calibrer ce capteur. Ce capteur estime la concentration de gaz inflammable à l'aide d'un rapport de résistance. Ce rapport comprend la résistance maximale du capteur à 1000 ppm R0 et la résistance interne RS. En air pur, après préchauffage, il faut télécharger le code d'étalonnage et attendre quinze minutes jusqu'à ce que R0 atteigne une valeur fixe. [11]

✓ Principe de fonctionnement de capteur MQ9

Lorsque l'air contient un capteur de monoxyde de carbone, de charbon, de gaz combustible liquéfié, la résistance du capteur diminue. Par conséquent, la sortie du capteur de gaz varie lorsque la concentration de gaz augmente. On peut régler la sensibilité en ajustant le potentiomètre.

✓ Principe de fonctionnement de capteur MQ2

Le principe de fonctionnement du capteur MQ-2 est basé sur la résistance. Lorsqu'il détecte de la fumée ou du feu, la valeur de sa résistance change. C'est ainsi que le gaz ou la fumée est détectée. Le capteur fonctionne sur une très basse tension d'environ 5 Volts. Le potentiomètre donné peut changer la sensibilité de la broche numérique, il tire un courant de 500mA. [12]

✓ Moniteur de la qualité de l'air de capteur MQ2

La surveillance de la qualité de l'air analyse plusieurs paramètres environnementaux de l'environnement et constitue donc la solution brillante pour contrôler les gaz de fumée spécifiques. Il utilise le capteur MQ-2 et est donc idéal pour détecter le méthane, l'alcool, la fumée et le propane.

Bibliographie

- [1] <https://itmsol.fr/services/objets-connectes-iot>
- [2] <https://www.guichetdusavoir.org/question/voir/32375>
- [3] <https://openlab.utbm.fr/docs/esp32/>
- [4] <https://www.lemondeinformatique.fr/actualites/lire-tout-savoir-sur-le-framework-machine-learning-tensorflow-75776.html>
- [5] <https://towardsdatascience.com/beginners-guide-to-tensorflow-2-x-for-deep-learning-applications-c7ebd0dcfbee>
- [6] <https://fr.wikipedia.org/wiki/React>
- [7] <https://www.encyclopedie.fr/definition/Fritzing>
- [8] <https://www.framboise314.fr/easyeda/>
- [9] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501859-transfert-learning-definition-exemples-et-avantages/>
- [10] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/>
- [11] <https://circuits-diy.com/mq9-carbon-monoxide-coal-liquefied-combustible-gas-sensor-module/>
- [12] <https://circuits-diy.com/mq-2-combustible-gas-smoke-sensor-module/>