الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département de Génie Électrique



Mémoire de Fin d'Études de Master

Filière : Électronique

Spécialité : Instrumentation

Thème

Système Intelligent de Contrôle de Feu De Circulation d'une Intersection avec une gestion Isolée

Préparé et présenté par

Melle. Aya Djihad HEBBACHE

Melle. Samah TEBBAL

Proposé et Dirigé par :

Prof. Mounir BOUHEDDA

Dr. Hamza BENYEZZA

Promotion 2023

Travail préparé au sein du Laboratoire des Systèmes Électroniques Avancés (LSEA) de l'Université de Médéa, ALGÉRIE

Dédicaces

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie.

Je dédie ce modeste travail :

A mes très chers parents, source de vie, d'amour et d'affection Mon père ***Mahfoud****, et ma mère* ***Nadia****, que je n'arriverai jamais à exprimer ma gratitude et ma reconnaissance, pour ses amours ses soutiens tout au long de mes études et leurs prières, leur encouragement et qu'aucune dédicace ne pourra compenser leur sacrifice pour mon bien être et mon bonheur.*

À la famille Tebbal surtout Mes sœurs et Mes frères : ***Amira****,* ***Meriem****,* ***Israa****,* ***Mustapha*** *et* ***Yazid*** *sources de joie et de bonheur.*

À mes amis et collègue : ***Wafaa, Walid***

En particulier à ma chère amie avant d'être un binôme ***Hebbache Aya Djihad*** *qui n'a jamais manqué de travailler.*

Nous dédions ce travail à notre encadreur, le Pr. ***Bouhedda Mounir****, ainsi qu'à notre co-encadreur, le* ***Dr. Benyezza Hamza****, pour leur patience incommensurable, leur disponibilité sans faille et leurs conseils précieux qui ont grandement contribué à la réalisation de ce travail. Qu'ils trouvent ici l'expression de notre profonde gratitude.*

À vous cher lecteur.

Et à tous ceux qui nous ont soutenus pour l'achèvement de ce projet avec réussite.

Samah

Dédicaces

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie.

Ce travail, et bien au-delà, je le dois à mes très chers parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots.

Mon père : ***Abdelkader*** *et Ma mère:* ***Zohra***

A mes sœurs : ***Saraa*** *et* ***Fatima zohra*** *et* ***Imane***

A mes frères : ***Mohammed, Ahmed, Imaad*** *et* ***Nour Eddin***

À mes amis et collègue : ***Wafaa, Saida , Khaoula , Rima, Ahlem ,Rihab***

En particulier à ma chère amie avant d'être un binôme ***Tebbal Samah*** *qui n'a jamais manqué de travailler.*

À notre encadreur ***Pr. Bouhedda.Mounir*** *et le sous encadreur* ***Dr. Benezza Hamza*** *pour ses immenses patiences, ses grandes disponibilités et ses conseils qui ont contribué grandement à la réalisation de ce travail.*

Aya

Remerciements

Nous tenons à remercier au premier lieu le Dieu tout puissant de nous avoir accordé la force et le courage de mener ce travail à terme.

Nous souhaitons exprimer nos plus sincères remerciements à notre encadreur, le Prof. Mounir BOUHEDDA de l'Université YAHIA FARES, pour sa disponibilité, sa lecture attentive, ses suggestions et remarques précieuses, et surtout pour la confiance illimitée qu'il a placée en nous tout au long de ce projet. Nous tenons à lui témoigner notre profonde reconnaissance et notre respect le plus sincère.

Nous tenons à exprimer au Dr. Hama BENYEZZA notre sincère gratitude pour ses orientations et précieuses aides.

Nous présentons également nos sincères remerciements à toute l'équipe pédagogique du département de génie électrique de l'université de Médéa, une équipe efficace qui nous apprise énormément de choses durant les cinq années de formation

Nous remercions particulièrement nos familles pour leurs soutiens moraux Pour ce travail, merci de nous avoir encouragés,

Nous remercions également nos ami(es), nos collègues de classe, qui nous ont encouragés tout au long de ce projet et nous ont beaucoup aidés.

Enfin, nos remerciements aux membres de jury qui ont fait l'honneur d'accepter de juger ce modeste travail.



نظام ذكي للتحكم في إشارات المرور في تقاطع معزول

ملخص - الهدف الرئيسي لهذا المشروع هو تطوير نظام ذكي للتحكم في إشارات المرور المتكيف باستخدام المنطق الضبابي. في البداية، يتم دراسة الحل والتحقق منه باستخدام بيئة الماتلاب. ثم يتم استخدام لغة برمجة بايثون لتنفيذ هذا المقترح. لضمان التحكم في الوقت الحقيقي في إشارات المرور، يتم استخدام راسبيري باي بالتعاون مع نود-رید، مما يتيح إنشاء واجهة إنسانية سهلة الاستخدام. يتم إنشاء التواصل بين خادم الحساب وجهاز التحكم في التقاطع عن طريق بروتوكول المساقط المترابطة (إم-كيو-تي-تي)، الذي يسهل التثبيت من خلال ضمان نقل معلومات لاسلكي فعال. يقدم هذا النهج المتكامل حلاً واعدًا لتحسين إدارة حركة المرور من خلال السماح بتكيف ذكي لإشارات المرور بناءً على الظروف في الوقت الحقيقي، مما يساهم في تحسين تدفق حركة المرور وتقليل الازدحام على الطرق.

الكلمات المفتاحية - إدارة تكيفية ذكية، إشارات المرور، المنطق الضبابي، راسبيري باي، نود-رید، واجهة إنسانية-آلية، اتصال إم-كيو-تي-تي، بايثون.

Intelligent Traffic Light Control System for an Isolated Intersection

ABSTRACT - The main objective of this study is to develop an intelligent adaptive traffic light control system using fuzzy logic. Initially, the solution is studied and validated using the Matlab environment. Then, Python programming language is employed to implement this approach. To ensure real-time control of the traffic lights, a Raspberry Pi is used in conjunction with Node-Red, enabling the creation of a user-friendly human-machine interface. The communication between the calculation server and the intersection control device is established through the MQTT protocol, which facilitates installation by ensuring efficient wireless transmission of information. This integrated approach offers a promising solution to improve traffic management by allowing intelligent adaptation of traffic lights based on real-time conditions, thereby contributing to smoother traffic flow and reduction of congestion on the roads.

KEYWORDS - Intelligent adaptive management, Traffic lights, Fuzzy logic, Raspberry Pi, Node-Red, Human-Machine Interface, MQTT communication.

Système intelligent de contrôle de feu de circulation d'une intersection isolée

RESUME - L'objectif principal de cette étude consiste à développer un système de contrôle adaptative intelligente des feux de circulation en utilisant la logique floue. Dans un premier temps, la solution est étudiée et validée en utilisant l'environnement Matlab. Ensuite, le langage de programmation Python est employé pour implémenter cette approche. Pour assurer le contrôle en temps réel des feux de circulation, un Raspberry Pi est utilisé en conjonction avec Node-Red, permettant ainsi la création d'une interface homme-machine conviviale. La communication entre le serveur de calcul et le dispositif de contrôle des feux d'intersection est établie grâce au protocole MQTT, qui facilite l'installation en assurant une transmission sans fil efficace des informations. Cette approche intégrée présente une solution prometteuse pour améliorer la gestion du trafic en permettant une adaptation intelligente des feux de circulation en fonction des conditions en temps réel, contribuant ainsi à une meilleure fluidité du trafic et à une réduction des embouteillages sur les routes.

MOTS-CLES - Gestion adaptative intelligente, Feux de circulation, Logique floue, Raspberry Pi, Node-Red, Interface Homme-Machine, Communication MQTT.

Sommaire

Sommaire I
Liste d'abréviations V
Liste de figures III
Liste des tableaux V
Introduction générale 1
Chapitre I : Généralités sur le feu de circulation
I.1. Introduction 3
I.2. Terminologie 3
I.2.1. Déplacement 3
I.2.2. Mobilité 3
I.2.3. Circulation routière 4
I.3. Feux de signalisation 5
I.3.1. Définition 5
I.3.2. Historique 5
I.3.3. Système de gestion des feux tricolores 6
I.3.4. Technologie de transport intelligente 6
I.4. Défirent Signaux Lumineux 6
I.4.1. Signal piéton 6
I.4.2. Signaux tricolores modaux 6
I.4.3. Signaux tricolores directionnels 7
I.4.4. Signaux d'anticipation directionnels 7
I.4.5. Signal d'arrêt 7
I.4.6. Signal d'arrêt piétons 8
I.5. Dans la perspective de l'utilisateur 8
I.6. Caractéristique du carrefour à feux étudié 9
I.7. Domaines d'emploi de feux de cirretculation 10
I.8. Intérêt et les inconvénients des feux de signalisation 10
I.8.1. Intérêt 10
I.8.2. Inconvénients 10
I.9. Différents feux de circulation intelligentes au monde 11
I.9.1. Singapour 11
I.9.2. Oslo 11

I.9.3. Londres 12
I.9.4. San Francisco 12
I.10. Conclusion 13

Chapitre II : Outils Software et Hardware utilisés

II.1 Introduction 14
II.2 Logique floue 15
II.2.1 Définition 15
II.2.2 Historique de la logique floue 15
II.2.3 Domaines d'application 16
II.2.4 Théorie de la logique floue 16
II.2.5 Structure d'un contrôleur flou 20
II.2.6 Interface de défuzzification 22
II.2.7 Exemple d'application 24
II.2.8 Avantages et les inconvénients du réglage par logique floue 26
II.3 Node-RED 26
II.3.1 Présentation de Node-RED 26
II.3.2 Composition de Node Red 27
II.4 Protocole MQTT 30
II.4.1 Introduction 30
II.4.2 Message Queuing Telemetry Transport 32
II.4.3 Mode de fonctionnement 33
II.4.4 Logiciel Mosquitto 33
II.5 Raspberry 34
II.6 Conclusion 35

Chapitre III: Réalisation pratique

III.1 Introduction 37
III.2 Position de problème 37
III.3 Cahier de charge 38
III.3.1 Optimisation du trafic 39
III.3.2 Réduction des temps d'attente 39
III.3.3 Sécurité routière accrue 39
III.4 Moyens hardware et software utilisés 39
III.4.1 Moyens Hardware 39
III.4.2 Moyens Software 39
III.5 Principe de contrôle 40
III.6 Contrôleur par logique floue 41
III.6.1 Système Flou 1 41

III.6.2 Système Flou 2 44
III.6.3 Tests de contrôleur 46
III.7 Fonctionnement global de système 47
III.8 Interface de supervision 48
III.8.1 Lancement de Node-RED sur Raspberry Pi 49
III.8.2 Programmation avec Node-RED 50
III.8.3 Traitement des données 50
III.8.4 Contrôle des feux 52
III.9 Réalisation 55
III.9.1 Schéma synoptique global de système 55
III.9.2 Etapes de réalisation 55
III.9.3 Prototype de réalisation 57
III.10 Évaluation 58
III.11 Conclusion 59
Conclusion 60
Références bibliographiques 62

Liste des Abréviations

D : débit de véhicule

IA : Intelligence Artificielle

IOT: Internet of Things

MQTT: Message Queuing Telemetry Transport

NR : Node -Red

NV : Nombre de véhicules

Ph : La phase

RLF : Régulateur par Logique Floue

SF1 : Système Floue1

SF2 : système Floue2

USA: United States of America

Liste des Figures

Chapitre I

Figure I.1. Deux voies d'aller et deux voies de retour 4
Figure I.2. Un carrefour 4
Figure I.3. Feu de circulation tricolores 5
Figure I.4. Signal piéton 6
Figure I.5. Signaux tricolores modaux. 7
Figure I.6. Signaux tricolores directionnels. 7
Figure I.7. Signaux d'anticipation directionnels. 7
Figure I.8. Signal d'arrêt. 8
Figure I.9. Signal d'arrêt piétons 8
Figure I.10. Zone fonctionnelle d'un carrefour simple 9
Figure I.11. Feu de signalisation à Singapour. 11
Figure I.12. Feu de circulation et feux de piéton à Oslo 12
Figure I.13. feux de circulation à la matinée en Londres 12
Figure I.14. Feux de circulation à San Francisco 13

Chapitre II

Figure II.1 : Présentation des Ensembles Floue 17
Figure II.2 : classification des personnes selon leur taille 18
FigureII.3 : Variable linguistique 'qualité du service 19
Figure II.4 : Différentes formes des fonctions d'appartenance 20
Figure II .5 : Structure interne d'un Régulateur de la logique floue 21
Figure II.6: Défuzzification par centre de gravité. 23
Figure II.7: Défuzzification par valeur maximum 23
Figure II.8: Ensembles flous pour exemple d'inférence Mamdani 25
Figure II.9: Nœuds de Node-RED les nœuds de traitement et les nœuds de sortie 27
Figure II.10 : composition d'un flux sur Node-Red 28
Figure II.11: nœuds de Dashboard 29
Figure II.12: Eléments de tableau de bord (Dashboard) 29
Figure II.13 : Représentation de la palette de GPIO 30
Figure II.14: Exemple de communication machine-à-machine suivant le protocole MQTT 32
Figure II.15: : exemple sur communication machine-à-machine suivant le protocole MQTT 33
Figure II.16 : Carte Raspberry Pi 3 34
Figure II.17: Brochage de Raspberry 35

Chapitre III

Figure III.1: Intersection à quatre chemins 37
Figure III.2: Phases de croisement à quatre voies 38
Figure III.3: Structure du contrôleur par la logique floue 41
Figure III.4: système floue1 42
Figure III.5 : présentation le nombre des véhicules dans l'intersection 42
Figure III.6: présentation le débit des véhicules dans l'intersection 42
Figure III.7: Variable sortie de la phase 43

Figure III.8 : système floue 2 44
Figure III.9: présentation le nombre des véhicules dans l'intersection 44
Figure III.10: présentation le débit des véhicules dans l'intersection 45
Figure III.11 : la durée de temps 45
Figure III.12: Mise en place des matériels du système. 48
Figure III.13: Représentation de démarrage de Node-Red. 49
Figure III.14 : Plateforme de développement Node-RED. 49
Figure III.15: Installation des nouvelles palettes 50
Figure III.16: Représentation de la palette "JSON" 51
Figure III.17 : Extraction d'une variable d'un objet 51
Figure III.18: Représentation de la palette d'affichage de phase. 51
Figure III.19: Flux de des données pour une voie. 52
Figure III.20 : Représentation de minuterie numérique. 53
Figure III.21: Flux d'affichage de temp. 53
Figure III.22: Configuration de nœud LEDs. 53
Figure III.23 : Flux d'affichage de l'état des LEDs pour chaque phase. 54
Figure III.24: L'affichage des états des feux dans l'interface graphique……..…...54
Figure III.25: Interface de supervision 54
Figure III.26: Programme final de Node RED.... 55
Figure III.27: Schéma synoptique de système. 55
Figure III.28: Montage des LEDs. 57
Figure III.29 : Représentations de circuit de la réalisation de système. 57
Figure III.30: Montage totale de système. 58
Figure III.30. Temps d'attente moyen basé sur le flux de véhicules pour l'algorithme AIT et le contrôleur flou 58

Liste des Tableaux

Chapitre II

Tableau II.1 : Représente exemple de quelques règles de conduite…….……...15
Tableau II.2: Représente les opérateurs floues.....20

Chapitre III

Tableau III.1 : Règles flous qui régissant le système flou1 43
Tableau III.2: présentation des règles floue.....46
Tableau III.3 : présenté une base de données d'entrée pour les sexes cycle..... 46
Tableau III.4: présenté les résulta des cycles..... 47
Tableau III.5: Etats des feux dans chaque phase.....52
Tableau III.6: présente Connexion des pins de LEDs avec le GPIO.....56
Tableau III.7. Temps d'attente moyen pour de multiple tests..... 58

Introduction Générale

La gestion efficace du trafic routier est devenue une préoccupation majeure dans les zones urbaines, où la congestion routière est une réalité omniprésente. Les feux de circulation jouent un rôle crucial dans la régulation du trafic, mais leur contrôle inefficace peut entraîner des embouteillages, des retards considérables et une détérioration de la qualité de vie des usagers de la route. Dans ce contexte, l'intelligence artificielle (IA) et les systèmes électroniques basés sur l'Internet des objets (IoT) offrent des solutions innovantes pour optimiser le contrôle des feux de circulation et atténuer les effets néfastes de la congestion.

La logique floue joue un rôle essentiel dans notre approche de contrôle intelligent des feux de circulation. En utilisant des concepts tels que les degrés d'appartenance et les variables linguistiques, la logique floue permet de modéliser et de gérer de manière adaptative les variations de flux de trafic, les heures de pointe et les conditions météorologiques. Elle offre ainsi la possibilité d'ajuster dynamiquement les séquences de feux de circulation en fonction du niveau de congestion, du volume de trafic et des priorités des différents flux, contribuant ainsi à une réduction significative de la congestion et à une meilleure fluidité du trafic.

Cette thèse se concentre sur le contrôle intelligent d'un feu de circulation d'une intersection isolée, visant à améliorer la fluidité du trafic et à réduire les problèmes de congestion. Nous proposons d'utiliser des technologies telles que MQTT (Message Queuing Telemetry Transport) et Node-RED, combinées à la logique floue en utilisant la programmation en Python, pour concevoir un système électronique intelligent capable d'optimiser les séquences de feux de circulation en temps réel selon les données reçues relatives à l'intersection.

La première partie de ce mémoire sera consacrée à une introduction générale, dans laquelle nous présenterons le contexte, la problématique et les objectifs de la recherche. Nous mettrons en évidence l'importance du contrôle des feux de circulation dans la gestion du trafic et les effets néfastes de la congestion.

Le document du mémoire contient trois chapitres. Le premier chapitre fournira des généralités sur le trafic routier, en expliquant les facteurs qui contribuent à la congestion et les approches traditionnelles de contrôle des feux de circulation. Nous soulignerons les limites de ces approches classiques et la nécessité d'adopter des méthodes plus avancées pour une gestion plus efficace du trafic. Le deuxième sera dédié à la présentation des moyens hardware et software

utilisés dans notre système électronique. Nous décrirons le fonctionnement de l'IoT, les principes de MQTT, Node-RED, la logique floue et la programmation en Python.

Dans le dernier chapitre, nous présenterons la méthodologie de l'implémentation et test. Nous démontrerons l'efficacité de notre système dans l'optimisation du contrôle des feux de circulation d'une intersection isolée. Nous analyserons les données recueillies, les performances du système et les avantages par rapport aux approches traditionnelles.

Enfin, le document est achevé par en une conclusion générale et des perspectives futures. Nous ferons le bilan de notre recherche, soulignerons les contributions apportées et discuterons des opportunités et des défis liés à une implémentation plus large de notre système dans des environnements réels.

Chapitre I

Généralités sur les feux de circulation

I.1 Introduction

Les systèmes traditionnels de contrôle des feux de circulation manquent souvent d'efficacité dans la gestion de la circulation urbaine mais depuis toujours, la technologie est auservice de l'homme. On n'oublie pas aussi que Le franchissement de plusieurs routes peut êtreune source sérieuse d'insécurité. La plupart des accidents et des décès sur les routes sont créés au carrefour, Alors ce chapitre permet d'identifier et de définir les principaux concepts relatifs au sujet.

I.2 Terminologie

I.2.1 Déplacement

C'est le mouvement de changement d'un lieu à un autre Il s'agit d'une simple voie d'accès réalisée sur la voie publique entre deux emplacements différents, chacun étant caractérisé par une activité, peuvent être effectués par un ou plusieurs modes de transport.

I.2.2 Mobilité

C'est la moyenne des trajets par personne et par jour. La mobilité varie en fonction de la superficie de la ville. Elle correspond de façon linéaire au niveau de vie (mesuré en fonction durevenu R) et la puissance du moteur Mָ (elle-même liée au revenu). Différents indicateurs peuvent être utilisé pour caractériser la mobilité d'une population par le temps et l'argent sont des indicateurs clés pour les familles, qui cherchent à maximiser leur temps et à minimiser leurs coûts. Le nombre moyen de trajets par jour et par personne est également important car il reflète l'interaction sociale et la fréquence des échanges avec les autres. De plus, la répartition des voyages entre les modes de transport reflète l'intérêt des consommateurs pour les différents moyens de transport. Les distances parcourues et la nature des connexions effectuées sont également des indicateurs clés qui reflètent l'attrait relatif des lieux et la possibilité d'échanger entre des lieux qui peuvent être éloignés ou situés à l'extérieur de la zone agglomération. Enfin, il est important de noter que la variabilité entre les types d'utilisation ou les ménages est caractérisée par leur situation d'habitation, leur place dans l'échelle sociale ou même dans le cycle de vie, ce qui est tout aussi important que les niveaux moyens de ces indicateurs [1].

I.2.3 Circulation routière

C’est le mouvement de véhicules dans une route, Elle est mesurée par le débit qui est le nombre de véhicules pendant une période de temp et on peut parler de plusieurspoint atteint la circulation routière.

- **Voie :** C'est le sentier, la route, par laquelle on va d'un endroit à un autre ; c'est une zone aménagée pour se déplacer dans la ville. (Voie urbaine) entre les localités Ouen milieu rural(routes). Les canaux de communication constituent donc les moyens de connexion dans l’espace ; la voie permet un trafic bidirectionnel, mais parfois, en particulier dans la ville. Augmenter la capacité du réseau routier en une seule direction. Il y a un trottoir (1,5 m est un minimum souhaitable) éventuellement plantés d'arbres (5m de largeur minimale) [1] .comme la figure I.1 suivante qui a deux voies d’aller et deux voies de retour.



Figure I.1. Deux voies d'aller et une voies de retour

- **Route** Itinéraire pour passer d'un lieu à un autre, Il s'agit d'une route destinée à la connexion entre les localités et le service des zones rurales Cela permet à l'homme de passerde sa maison à d'autres régions plus ou moins éloignées [1].
- **Carrefour** Cela permet à l'homme de se déplacer de son domicile à d'autres zones plus ou moins isolées aussi un ensemble de carrefours constitue un réseau [2].telle que représente la figure I.2.



Figure I.2. Carrefour

I.3 Feux de signalisation

I.3.1 Définition

C’est un feu de trois couleurs (vert, rouge, orange) qui se trouve partout où des gens ont besoin de traverser la rue. Ces feux doivent signaler quand passer, quand ralentir et quand s'arrêter
Les dispositifs de contrôle des feux de circulation ont permis de limiter les accidents urbains et de simplifier la circulation. Ils ontun cycle fixe prédéfini et ne prennent pas en considération les situations réelles de trafic. Comme les comptages de véhicules et de piétons. Les embouteillages sont l'un des problèmes majeurs des villes mobilisées avec une population importante. Ce problème estlié à une hausse significative du nombre de véhicules. La figure I.3 représente un feu de circulation tricolores [3].



Figure I.3. Feu de circulation tricolores

I.3.2 Historique

Le premier dispositif de signalisation à destination des usagers de la route a été le corps humain, plus précisément le corps du policier. C’est d’abord par l’usage de ses bras et de sa voix qu’il fut appelé à effectuer certaines opérations de régulation de la circulation, qui était encore à l’époque pédestre et hippique. Face à l’inefficacité de ces gestes, qui étaient fréquemment ignorés par leurs destinataires, le policier s’équipa d’outils pour augmenter la portée de son action. Le premier de ceux-ci, installé à Londres en 1868, fut un sémaphore artificiel constitué de bras articulés. Il était animé par la force musculaire d’un policier. Ensuite, il fut équipé de lampes de différentes couleurs destinées à améliorer leur visibilité nocturne, alimentées au gaz et allumées par le même policier. Mais le dispositif londonien eut un destin tragique. Non seulement il provoqua la panique d’un détachement de cavalerie, laquelle entraînera la mort de deux policiers, mais il connut également en 1869 une explosion qui atteindra mortellement le policier venu allumer les lampes. Un appareil modifié fut installéen 1872. Ces différents événements firent naître une certaine méfiance quant à la généralisationde tels dispositifs dans la capitale britannique. C’est pourquoi, à la veille de la première guerre mondiale, seule une poignée de feux de signalisation, alimentés au kérosène, y étaient en

service. Un autre sémaphore routier fut également installé à Toledo, en Ohio, en 1908, mais son succès fut mitigé en raison de sa faible visibilité nocturne. En 1930, il n'existait plus aucun dispositif de ce type aux Etats-Unis[4].

I.3.3 Système de gestion des feux tricolores

Le système de gestion des feux de circulation est un système utilisé pour surveiller et contrôler l'écoulement à la croisée des chemins, en assurant le bon mouvement des véhicules sur les routes et la sécurité en partageant au fil du temps l'utilisation du même espace entre les flux contradictoires. Il contribue également à la gestion du flux de la demande [5].

I.3.4 Technologie de transport intelligente

Les systèmes de transport intelligents sont équipés de technologies de communication sans fil pour collecter et transférer des données. Ces technologies comprennent des communications à faible portée telles que le protocole IEEE 802.11 et des communications à longue portée comme WIMAX et GSM. La technologie de localisation est également importante pour le développement d'une mobilité plus intelligente, plus efficace et plusécologique. Les services et les technologies qui permettent de localiser les passagers, les marchandises et les véhicules se développent rapidement, notamment grâce au GPS et à la téléphonie mobile. Les technologies de capteurs sont également indispensables pour le contrôle, la gestion, l'amélioration et l'innovation des systèmes embarqués dans un véhicule. Ces systèmes utilisent des capteurs pour recueillir des données environnementales en temps réel de macière précise et accessible [6].

I.4 Différents signaux lumineux

I.4.1 Signal piéton

C'est un signal constitué de deux feux vert et rouge, normalement disposés dans cet ordre de droite à gauche (figure I.4); éventuellement ils peuvent être disposés l'un au-dessus de l'autre, ou le vert est en bas [7].



Figure I.4. Signal piéton

I.4.2 Signaux tricolores modaux

Ils sont composés de trois feux verts, jaune, rouge, dans cet ordre de bas en haut, munis chacun d'un même pictogramme. Le feu vert peut être remplacé par un feu jaune clignotant, les

signaux se dénommant alors respectivement telle que la figure I.5 [7].

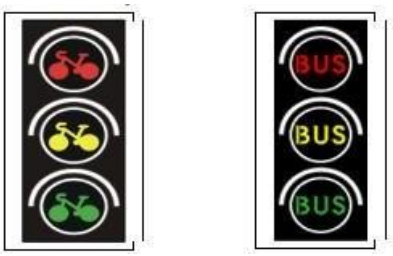


Figure I.5. Signaux tricolores modaux

I.4.3 Signaux tricolores directionnels

Ils sont destinés chacun à l'ensemble des véhicules qui ont pour destination la direction indiquée par la flèche, ou l'une des directions indiquées. En aucun cas le feu vert ne peut être remplacé par un feu jaune clignotant tel que la figure I.6 suivante[7].

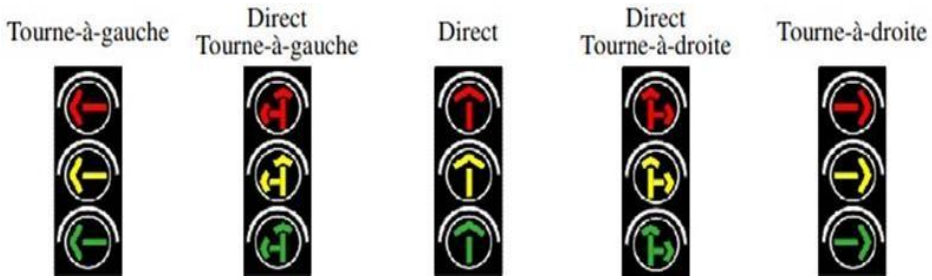


Figure I.6. Signaux tricolores directionnels.

I.4.4 Signaux d'anticipation directionnels

Ils sont composés d'un feu jaune clignotant et sont obligatoirement associés à unensemble de feux tricolores circulaires (vert sur le feu du bas). II est recommandé de les associeraux signaux comme indiqué figure. Ils sont munis d'un pictogramme en forme d'une ou deux flèches comme (figure I.7) [7].

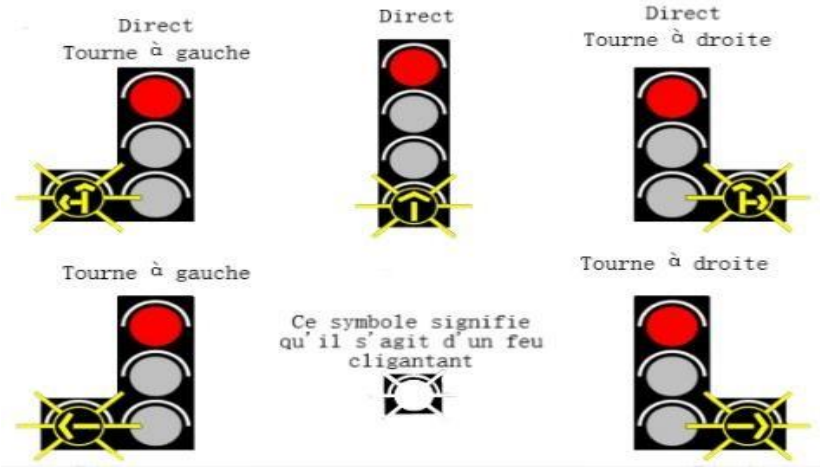


Figure I.7. Signaux d'anticipation directionnels

I.4.5 Signal d'arrêt

Il est composé d'un feu circulaire rouge clignotant. Éventuellement, deux de ces signaux peuvent être assemblés ou rappelés, et clignoter en synchronisme ou en alternance. Il est destiné à interdire momentanément la circulation à tout véhicule routier, devant un obstacle ou un danger particulier (passage à niveau, traversée de voies exclusivement réservées aux véhicules des services réguliers de transport en commun, pont mobile, avalanche…). Il peut être employé pour favoriser le débouché sur la voie publique des véhicules prioritaires des pompiers comme indiquez ci-dessus sur la figure I.8 [8].



Figure I.8. Signal d'arrêt.

I.4.6 Signal d'arrêt piétons

C'est pour piétons Il est composé d'un pictogramme rouge fixe figurant un piéton surmontant un pictogramme rouge clignotant portant la mention STOP. Il est destiné à interdirela traversée par les piétons des sites exclusivement réservés aux véhicules des services régulièrsde transport en commun (figure I.9) [8].



Figure I.9. Signal d'arrêt piétons

I.5 Dans la perspective de l'utilisateur

En tant qu'utilisateur de feux de circulation, notre principale préoccupation est la sécurité routière. Nous devons respecter les feux de signalisation pour éviter les accidents et maintenir un environnement routier sécurisé. Les feux de circulation fonctionnent en conformité avec le code de la route, et il est essentiel de connaître et de respecter les règles qui leur sont associées. Lorsque nous approchons d'une intersection, nous devons être attentifs aux autres véhicules et aux piétons qui peuvent traverser lorsque le feu change de couleur. La patience et l'anticipation sont également importantes, car les feux de circulation peuvent entraîner des périodes d'attente, en particulier aux heures de pointe. Nous utilisons les feux comme repères pour la navigation, en identifiant les intersections et en suivant les directions appropriées.

Nous devons également être conscients des feux spéciaux destinés aux véhicules d'urgence, aux transports en commun ou aux véhicules prioritaires. Enfin, n'oublions pas que les feux de circulation sont également importants pour les piétons. Nous devons respecter les feux de signalisation spécifiques aux piétons lorsque nous traversons la route. En suivant ces conseils et en étant attentifs à notre environnement, nous contribuons à assurer la sécurité de tous les usagers de la route.

I.6 Caractéristiques du carrefour à feux étudié

La figure I.13 représente les zones fonctionnelles d'un carrefour, Dans chaque intersection, nous pouvons identifier trois zones fonctionnelles :

- Une zone de stockage, placée : en amont de la zone de conflit, constitue l'entrée empruntée par les véhicules
- Une zone de conflit, relative à l'espace de croisement des routes ; i.e. la ressource critique partagée par tous les véhicules qui traversent l'intersection
- Une zone de sortie, placée en aval de la zone de conflit, permet le soulagement de cette zone.

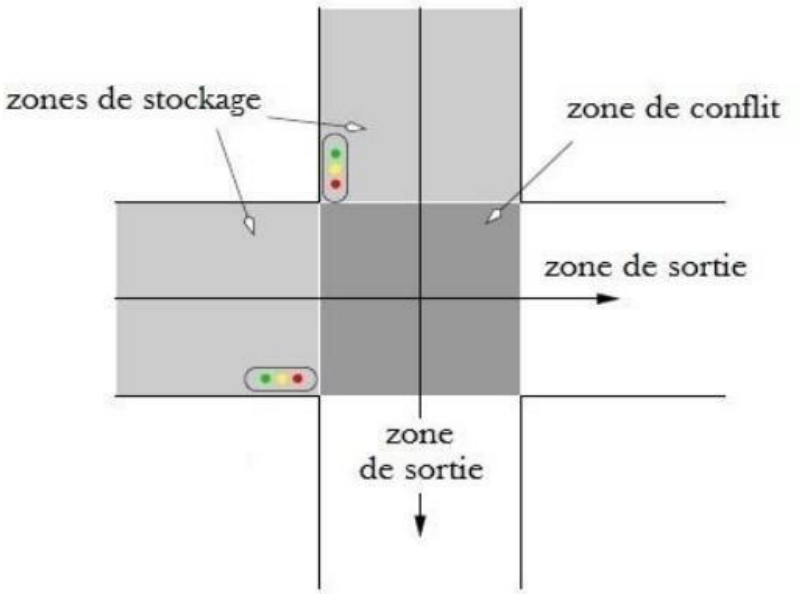


Figure I.10. Zone fonctionnelle d'un carrefour simple

Le caractère classique de la signalisation lumineuse concerne tant les couleurs à utiliser que l'ordre dans lequel elles apparaissent. En effet, l'éclairage des lumières est séquentiel, ce qui exige une procédure ordonnée, dans ce cas l'ordre vert-jaune-rouge (Lannoy, 2001). Un vocabulaire technique commun, désignant le fonctionnement de la signalétique tricolore, est également employé par toute la communauté d'ingénieurs concernés par le domaine :

- **Condition des feux :** Le feu qui contrôle le passage libre (feu vert), toléré (feu orange) ou interdit (feu rouge) de la circulation automobile.
- **Phase :** Durée pendant laquelle un incendie traverse les trois états, c'est-à-dire qu'il exécuteun signal lumineux.
- **Cycle** Période au cours de laquelle tous les feux exécutent leur phase permettant successivement l'admission de tous les courants du véhicule dans cette jonction.
- **Plan de feux** Durée et séquence de l'allumage des trois séries de lumières au cours d'un cycle ; une même intersection peut suivre en alternance plusieurs plans d'incendie[10].

I.7 Domaines d'emploi de feux de circulation

Le domaine d'emploi des feux de circulation, également connus sous le nom de feux de signalisation, est la régulation du trafic routier. Ils sont utilisés pour contrôler la circulation des véhicules et des piétons aux intersections, en indiquant quand s'arrêter et quand avancer.

Les feux de circulation sont utilisés dans les zones urbaines, les carrefours, les intersections complexes, les passages pour piétons et d'autres endroits où la régulation du trafic est nécessaire. Leur objectif principal est d'améliorer la sécurité routière en réduisant les risques de collisions et de conflits entre les véhicules et les piétons.

I.8 Intérêt et les inconvénients des feux de signalisation

I.8.1 Intérêt

Le passage à des systèmes de feux de signalisation adaptatifs a plusieurs avantages potentiels. Tout d'abord, cela pourrait entraîner une diminution du bruit et de la consommationd'énergie liés au freinage des feux. De plus, la plupart des véhicules passent en réalité au début du feu vert, donc moins d'engrenages seraient nécessaires à la fin du vert, ce qui pourrait améliorer la fluidité de la circulation et entraîner une diminution des arrêts. Enfin, cela pourraitégalement réduire le nombre de passages à niveau en rouge, ce qui améliorerait la sécurité pourles piétons et les cyclistes.

I.8.2 Inconvénients

Lors de l'évaluation des fentes sur les routes courbes, il peut y avoir des erreurs. En effet,la courbure peut fausser la perception de la distance et donc des fentes. De plus, dans le cas de deux voies de circulation, la direction inverse peut être moins visible en raison de la présence d'autres véhicules, ce qui peut entraîner des risques d'accidents. Enfin, il est possible que certains piétons ne respectent pas le signal rouge, ce qui peut également causer des problèmes

de sécurité.

I.9 Différents feux de circulation intelligentes au monde

I.9.1 Singapour

Singapour est réputée pour son système de gestion du trafic avancé. Les feux de signalisation à Singapour sont sophistiqués et utilisent des technologies avancées pour optimiser la circulation. Par exemple, le système de contrôle des feux de signalisation à Singapour est basé sur des algorithmes de synchronisation intelligents pour améliorer l'efficacité de la circulation et réduire les embouteillages.



Figure I.11. Feu de signalisation à Singapour

I.9.2 Oslo (Capitale Norvégienne connectée)

À Oslo, les feux de signalisation sont similaires à ceux utilisés dans d'autres pays européens. Ils utilisent des feux rouges, jaunes et verts. Cependant, la Norvège est connue pour être pionnière dans l'utilisation de feux de signalisation intelligents. Par exemple, certains feux de signalisation sont équipés de capteurs qui détectent la présence de véhicules et ajustent les durées des feux en fonction du trafic réel.



Figure I.12. Feu de circulation et feux de piéton à Oslo

I.9.3 Londres (Smart City on co-construction)

À Londres, les feux de signalisation suivent généralement la convention britannique. Ils utilisent des feux rouges, jaunes et verts, mais la disposition peut varier. Les feux peuvent être positionnés horizontalement plutôt que verticalement. Dans certains cas, des feux de signalisation à compte à rebours sont également utilisés, indiquant le temps restant avant le changement de signal.



Figure I.13. Feux de circulation à la matinée en Londres

I.9.4 San Francisco

San Francisco utilise généralement des feux de signalisation standard avec des feux rouges, jaunes et verts. Cependant, en raison de sa topographie unique et de la densité de la circulation, certaines intersections de San Francisco sont équipées de feux de signalisation spéciaux pour les tramways et les lignes de bus, afin de faciliter la coordination entre les différents modes de transport.



Figure I.14. Feux de circulation à San Francisco

I.10 Conclusion

Dans ce chapitre nous avons abordé un aperçu complet et détaillé des feux de signalisation et de leur rôle essentiel dans la régulation du trafic et la sécurité routière. Nous avons exploré la terminologie associée, y compris les concepts de déplacement et de mobilité. De plus, nous avons examiné les différents types de signaux lumineux utilisés, tels que les signaux piétons, les signaux tricolores modaux et directionnels, ainsi que les signaux d'arrêt.

Nous avons également abordé les aspects pratiques et fonctionnels des feux de signalisation, en mettant l'accent sur leur utilité pour les usagers de la route. De plus, nous avons discuté de la problématique liée à leur utilisation et examiné les caractéristiques spécifiques d'un carrefour à feux étudié. Les domaines d'application des feux de circulation ont été explorés, soulignant leur importance dans la gestion du trafic dans différents contextes.

Enfin, nous avons évalué l'intérêt et les inconvénients des feux de signalisation, en reconnaissant leur contribution à la sécurité routière, mais aussi les éventuelles limitations ou critiques qu'ils peuvent susciter.

Chapitre II

Outils Software et Hardware utilisés

II.1 Introduction

La conception et la mise en œuvre des systèmes d'informations avancés, de logiciels d'aide à la décision, de postes de supervisions de procédés industriels (avec les problèmes de communications homme machine qui les accompagnent) se caractérisent par la constante manipulation d'informations dont beaucoup sont subjectives, imprécises, vagues, incertaines.

Mener à bien cette intégration de l'homme dans son environnement informatique, exploiter correctement les savoirs, automatiser certaines tâches, tout cela requiert la formalisation et la mécanisation de méthodes humaines pour de raisonnement empiriques ou naturel, la rationalisation de procédures de choix [1].

Généralement, nous sommes confrontés au besoin de modéliser la connaissance des problèmes, qui rompt avec la tradition des soi-disant sciences objectives, qui s'intéresse essentiellement à modéliser l'univers physique[2].

La question de la représentation et de l'utilisation de la connaissance est au cœur d'une discipline scientifique relativement nouvelle et controversée qu'on appelle "intelligence artificielle (IA)". Encore récemment, la discipline a eu un impact limité sur les applications industrielles, parce qu'elle portait uniquement sur la manipulation symbolique du savoir plutôt que sur la modélisation numérique traditionnellement utilisée en sciences de l'ingénieur. Plus récemment, nous avons observé un retour numérique de ces problèmes d'IA, y compris les réseaux neuronaux et la logique floue[3].

Les réseaux neuronaux offrent une représentation implicite (black box) des connaissances, très similaire au processus de reconnaissance des systèmes automatiques, alors que la logique floue est plus cohérente avec l'intelligence artificielle symbolique, qui propose la notion de raisonnement, où la connaissance est explicitement encodée, bien que, la logique floue a permis de faire le lien entre la modélisation numérique et la modélisation symbolique, Cela a permis le développement industriel à partir d'algorithmes très simples pour convertir le savoir symbolique en entités numériques et vice versa. La théorie des ensembles flous produit aussi un traitement original de l'incertitude fondé sur la pensée séquentielle, qui permet de traiter formellement l'ignorance partielle et l'inconsistance des systèmes d'information [2].

II.2 Logique floue

II.2.1 Définition

La logique floue est une extension de la logique booléenne créée par Lotfi ZADEH en 1965 sur la base de sa théorie mathématique des ensembles floues qui est une généralisation de la théorie classique des ensembles, l'introduction du concept de degré dans la vérification d'une condition, permettant ainsi qu'une condition se trouve dans un état autre que vrai ou faux, la logique floue confère une très grande souplesse au raisonnement qui l'utilise, Ceci permet de prendre en considération les inexactitudes et les incertitudes[4].

L'un des intérêts de la logique floue dans l'officialisation du raisonnement humain est que les règles sont exprimées en langage naturel. Le Tableau II.1 donne quelques règles sur un tableaux de conduite qu'un conducteur dans des conditions normales et réglementaires. Intuitivement, il semble que le cerveau apprécie de façon approximative des variables d'entréecomme cet exemple, ce qui correspond au degré de vérification d'un état logique flou[4].

Tableaux II.1. Représente exemple de quelques règles de conduite.

Si le feu est rouge…	Si ma vitesse estélevée …	Et si le feu estproche …	Alors je freine fort.
Si le feu est rouge	Si ma vitesse estfaible …	Et si le feu est loin …	Alors je maintiens mavitesse.
Si le feu est orange …	Si ma vitesse estmoyenne	Et si le feu estloin…	Alors je freinedoucement.
Si le feu est vert …	Si ma vitesse estfaible	Et si le feu estproche …	Alors j'accélère.

II.2.2 Historique de la logique floue

- **1965** : le concept de la logique floue est introduit par Pr. Lotfi ZADEH (Université de Berkeley), où il définit des ensembles flous et des opérateurs associés ; théorie des ensembles flou
- **1970 :** les premières applications reposaient sur le concept d'expertise qui permet de quantifier le flou sur la base de connaissances acquises précédemment ; Systèmes spécialisés d'aide à la décision en logique floue et de diagnostic dans l'orientation médicale, commerciale, professionnelle etc.
- **1974-1975 :** la première application industrielle apparaît ; le professeur Mamdani développait une stratégie de contrôle des procédés et les résultats obtenus ont été

appliqués au fonctionnement d'une machine à vapeur. Réglementation imprécise d'une chaudière à vapeur effectuée par Mamdani à Londres.

- **1978-1980 :** la société Danoise F.L. Smidth a réalisé le contrôle d'un four à ciment. il est maintenant que la première application industrielle réelle de la logique floue est en train d'émerger.
- Au Japon, la logique floue a connu son vrai boom à la fin des années 1980 dont la recherche est non seulement théorique mais aussi très applicable dans plusieurs secteurs[5]

II.2.3 Domaines d'application

Les systèmes flous ont été employés dans une grande variété d'applications industrielles, de gestion et médicales Le contrôle flou se révèle être utilisé dans beaucoup de domaines[5].

- Contrôle des métros grâce à un fonctionnement plus confortable et économique et uneprécision d'arrêt accrue. (1987 à Sendai, Japon).
- Commande de tubes broyeurs pour la fabrication du ciment (première réalisation en1979 au Danemark).
- Production du fer et de l'acier, purification de l'eau, chaînes et robots de fabrication,opérateurs, unités arithmétiques, micro-ordinateurs.
- Contrôle des lave-linge, avion, drone, machine industrielle, …

II.2.4 Théorie de la logique floue

II.2.4.1 Ensemble floues

La logique floue est fondée sur la théorie des ensembles flous, qui ne sont que la généralisation de la théorie des ensembles classiques. Pour faire une métaphore en langage ensembliste, la théorie des ensembles classiques n'est qu'un sous-ensemble de la théorie desensembles flous (Figure II.1) [5], [6].

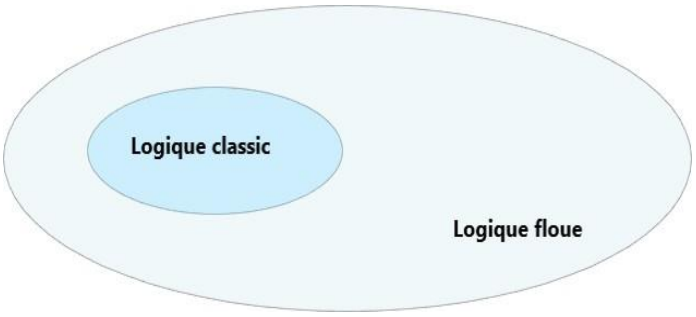


Figure II.1. Présentation des Ensembles Flous

Soit A un ensemble flou défini sur un univers de discours X, il est caractérisé par une fonction d'appartenance $\mu_A(x)$ qui prend des valeurs entre [0, 1].

$$A = \{x, \mu_A(x) / x \in X\} \quad \text{(II.1)}$$

Autrement, en logique Floue, un ensemble flou « A » dans un univers de discours « X » est défini par une fonction d'appartenance « μ_A » qui à tout élément « x » appartenant à «X», est associé un nombre « $\mu_A(x)$ » compris entre 0 et 1.

Un ensemble flou A sur un univers de discours X est un ensemble de couples ordonnés :

1. *A* est tel que tout *x* en est membre à un degré entre 0 et 1
2. *A* est entièrement caractérisé par sa fonction d'appartenance.

Concrètement, *A* est un attribut qualitatif (valeur floue ou linguistique) que l'on associe avec les valeurs précises d'une variable numérique *x* [5].

Exemple

Un exemple simple d'ensembles flous est la classification des personnes selon leur taille en trois ensembles : petite, moyenne et grande. La façon d'établir cette classification est présentée par la figure II.2.

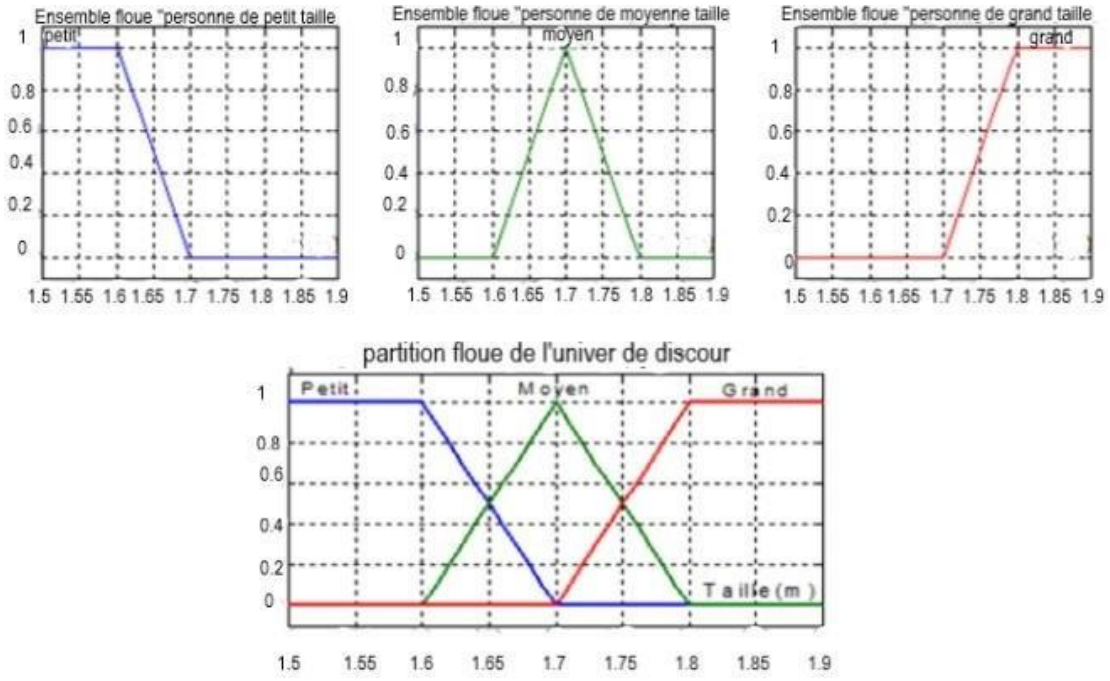


Figure II.2. Classification des personnes selon leur taille

Cette figure montre que la différence entre ces trois groupes ne varie pas soudainement, mais progressivement. Par exemple, une personne du taille 1m60 appartient l'ensemble "petit"Avec une valeur μ=0.75 de la fonction d'appartenance et à l'ensemble "moyens" avec une valeur μ=0.25 et une valeur grande μ=0. La figure II.2 donne le degré d'appartenance d'une personne, selon sa taille, à un certain ensemble flou**.** Pour cette raison, ce type de figure s'appelle une fonction d'appartenance

On peut ainsi résumer la terminologie dans l'illustration suivante :

- Variable linguistique : Taille
- Valeur d'une variable linguistique : Petit, Moyen, Grand, ...
- Ensembles flous : "petit", "moyen", "grand", ...
- Plage de valeurs : (0, 1.60,1.8, ...
- Fonction d'appartenance : $\mu_E(x) = m \quad (0 \leq m \leq 1)$
- Degré d'appartenance : m

II.2.4.2 Variables linguistiques

Une variable linguistique est une variable dont les valeurs sont des mots ou des expressions exprimés dans un langage naturel ou artificiel [1]. Soit V une variable (qualité de service, quantité de pointe, etc.), X l'intervalle de valeurs de la variable et TV un ensemble fini ou infini de sous-ensembles flous (Figure II.3).

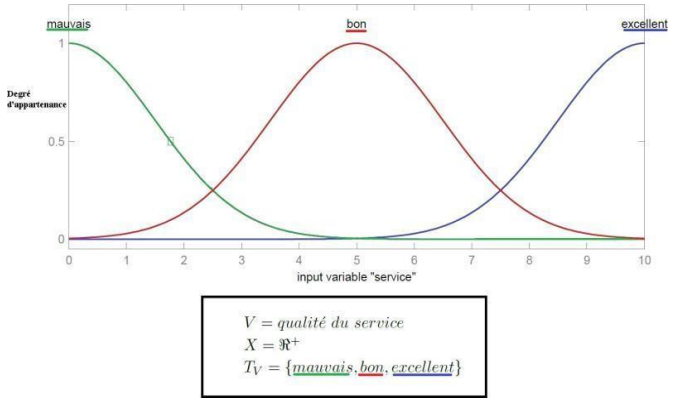


Figure II.3 : Variable linguistique ’qualité du service

II.2.4.3 Fonction d’appartenance

Le concept d'appartenance est primordial en théorie des ensembles : il signifie si un élément fait partie d'un ensemble ou pas. Par exemple, l'entier 7 appartient à l'ensemble {6;7; 9} et l'entier 5 n'appartient pas à l'ensemble {6; 7; 9}. Une fonction d'appartenance*(également appelée fonction indicatrice ou encore fonction caractéristique)* est une fonction qui explicite l’appartenance ou non à un ensemble A [5].

II.2.4.4 Exemple

Soit F la fonction caractéristique de l'ensemble A= {6; 7; 9}, et *x* un entier quelconque :

$$\mu_A(x) = \begin{cases} 0 \text{ si } x \notin A \\ 1 \text{ si } x \in A \end{cases} \qquad \text{(II.2)}$$

Cette notion d'appartenance est très importante, parce que la logique floue repose sur la notion d'appartenance floue. Ceci signifie simplement qu'on peut appartenir, par exemple, à 0.8 (80%) d'un ensemble. Contrairement à la théorie des ensembles classiques où comme nous venons de le voir l'appartenance est soit 0 (n'appartient pas) ou 1(appartient)[5]. La figure II.4 montre différentes formes des fonctions d’appartenance

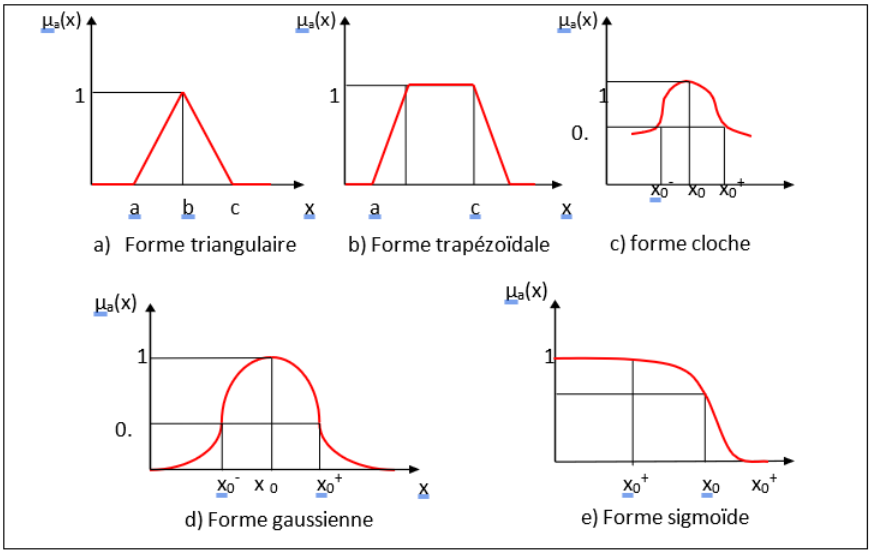


Figure II.4. Différentes formes des fonctions d'appartenance

II.2.4.5 Opérateurs flous

Afin de pouvoir manipuler aisément les ensembles flous, nous redéfinissons les opérateurs de la théorie des ensembles classiques afin de les adapter aux fonctions d'appartenance propres à la logique floue permettant des valeurs strictement entre 0 et 1.

Contrairement aux d´définitions des propriétés des ensembles flous qui sont toujours les mêmes, la d´définition des opérateurs sur les ensembles flous est choisie, `a l'instar des fonctions d'appartenance. Voici les deux ensembles d'opérateurs pour le complément (NON), l'intersection (ET) et l'union (OU) utilisés le plus couramment [4]. Le tableau II.2 représente les opérateurs flous, le complément (NON), l'intersection (ET) et l'union (OU).

Tableaux II.2. Opérateurs flous

Dénomination	Intersection ET	Réunion OU	Complément NON
Symbole	$\mu_{A\cap B}(x)$	$\mu_{A\cup B}(x)$	$\mu_{\bar{A}}(x)$
Opérateurs de Zadeh : MIN/MAX	Min ($\mu_A(x)$; $\mu_B(x)$)	Max ($\mu_A(x)$; $\mu_B(x)$)	$1-\mu_A(x)$
Probabiliste : PROD/PROBOR	$\mu_A(x) \times \mu_B(x)$	$\mu_A(x)+\mu_B(x)-\mu_A(x)\times\mu_B(x)$	$1-\mu_A(x)$

II.2.5 Structure d'un contrôleur flou

La structure d'un contrôleur flou comporte trois grandes parties :

- Une interface de fuzzification,
- Un mécanisme d'inférence,
- Une interface de Défuzzification.

La figure II.6 représente, à titre d'illustration la structure d'un régulateur flou à deux entrées et une sortie : ou X1et X2 représentent les variables d'entrée, et Xr celle de sortie ou la commande [1].

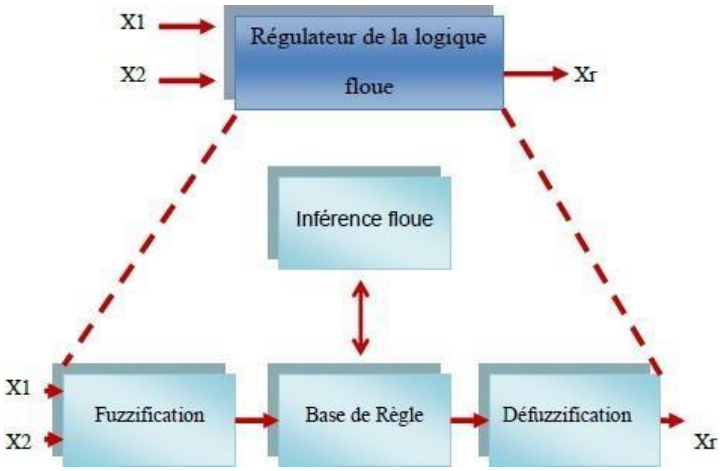


Figure II.5. Structure interne d'un Régulateur de la logique floue

II.2.5.1 Interface de fuzzification

Le bloc de fuzzification définit les intervalles de valeurs pour les fonctions d'appartenance à partir des valeurs des variables d'entré, exécute une fonction de fuzzification qui convertit les données d'entrée Cela devrait être fait dans un domaine habituellement normalisé par l'intervalle [1 ,1] afin de faciliter le calcul[1].

II.2.5.2 Base de règles

Le bloc de base de connaissances inclut les connaissances dans le domaine d'applicationet le résultat de commande attendu. Il se compose de "base de données" et "langue de commande (flou) base de règles". La base de données effectue des définitions qui sont nécessaires pour établir les règlesde commande et manipuler les données floues dans un RLF. La base de règles représente la stratégie de commande et l'objectif souhaité à travers

lesrègles de contrôle de langue [1].

II.2.5.3 Mécanisme d'Inférence floue

Le bloc inférence est le cœur d'un RLF**,** il possède la capacité de simuler les décisions humaines et de déduire les actions de commande floue à l'aide de l'implication floue et des règles d'inférence dans la logique floue. Le traitement numérique des règles d'inférence qui permet d'obtenir le résultat linguistique ou flou du régulateur se fait par différentes méthodes,on cite principalement les trois méthodes suivantes [7].

II.2.5.4 Méthode d'inférence max-min (méthode de Mamdani)

La méthode d'inférence max-min est réalisée à la condition de l'opérateur "AND" par la formulation du minimum. Le résultat de chaque règle, introduit par « THEN », lie le facteur d'appartenance de la prémisse avec la fonction d'appartenance de la variable de sortie est réalisé par la formation du minimum. Finalement l'opérateur « OR » qui lie les diverses règles est réalisé par la formation du maximum [7].

II.2.5.5 Méthode d'inférence max-produit (méthode de Larsen)

La méthode d'inférence du produit max est effectué, au niveau de l'état, l'opérateur «AND » par la formation du produit. La condition dans chaque règle, introduite par « THEN» se fait par la formation du produit. L'opérateur « OR », qui lie les différentes règles, se concrétise par la formation du maximum[1]

II.2.5.6 Méthode de Sugeno

L'opérateur « AND » est réalisé par la formation du minimum, la conclusion de chaque règle floutée présente une forme polynomiale. La sortie est égale à la moyenne pondérée de la sortie de chaque règle floue[1].

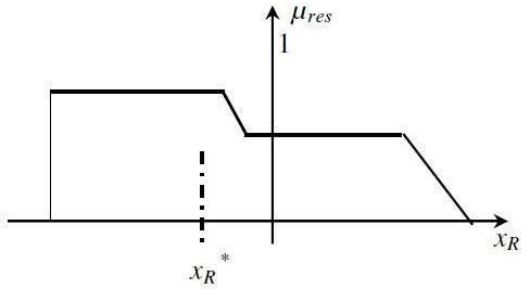
II.2.6 Interface de défuzzification

Les méthodes d'inférences fournissent une fonction d'appartenance résultante $\mu_{res}(X_r)$ pour la variable de sortie (X_r)Il s'agit donc d'une information floue. Les actionneurs actuels, utilisées dans les boucles de commande ne s'accompagnent pas de ce genre de décision, il convient de la transformer en une grandeur de commande précise, il faut prévoir une transformation de cette information floue en une information déterminée. Cette transformation est appelée défuzzification. Les méthodes de défuzzification les plus

couramment utilisées sont les trois méthodes expliquées ci-après [7]

II.2.6.1 Méthode de centre de gravité

La méthode de défuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante $\mu_{res}(x_r)$. Dans ce contexte, il suffit de calculer l'abscisse (x_r). La figure II.7 montre le principe de défuzzification [7].



FigureII.6. Défuzzification par centre de gravité.

L'abscisse de centre de gravité peut être déterminée à l'aide de la relation générale (II.3).

$$x_R^* = \frac{\int_{-1}^{1} x_R \mu_{res}(x_R) dx_R}{\int_{-1}^{1} x_R(x_R) dx_R} \qquad \text{(II.3)}$$

II.2.6.2 La méthode de maximum

La défuzzification par centre de gravité exige en général une envergure de calcul assez importante. Par conséquent, il sera utile de disposer d'une méthode de défuzzification plus simple. Quelquefois, on fait appel à la méthode par valeur maximum. Comme signal de sortie x_{R^*}, on choisit l'abscisse de la valeur maximale dela fonction d'appartenance résultante μ_{res} (x_R). Comme la montre par la figure II.8 [7].

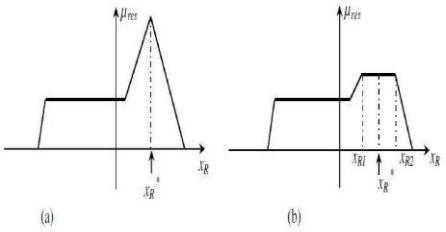


Figure II.7. Défuzzification par valeur maximum

Lorsque $\mu_{res}(x_R)$ est écrêtée, toutes valeurs entre x_{R1} et x_{R2} peuvent être utilisées. Afin d'éviter cette indétermination, on prend la moyenne des abscisses du maximum (figure II.8). Cependant cette méthode présente un grand inconvénient c'est que le signal de sortie x_{R^*}, saute si la dominance change d'une fonction partielle à une autre[7]

II.2.6.3 Méthode de la moyenne des maximas

Cette méthode génère un signal de commande qui représente la valeur moyenne de tous les maximums, dans le cas où il existe plusieurs valeurs pour lesquelles la fonction d'appartenance résultante est maximale[1].

II.2.6.4 Défuzzification

Après avoir combiné les règles, il faut maintenant produire un chiffre net comme sortie. Dans ce cas-ci, la sortie doit être la vitesse du ventilateur. La technique la plus populaire est làméthode du centroïde on cherche le centre de gravité du polygone obtenu [8]. Le centre de gravité n'a pas besoin d'être calculé des façons très précises. On peut approximer en considérant des petites surfaces.

II.2.7 Exemple d'application

Dans cet exemple, on va montrer l'application des quatre étapes d'un contrôleur flou[9].

1) Fuzzification des variables d'entrée
2) Évaluation des règles
3) Agrégation des sorties des règles
4) Défuzzification

On prend l'exemple d'un système de contrôle d'un ventilateur de maison, ayant 2 entrées (température et humidité) et une sortie (vitesse du ventilateur). Des règles possibles sont [9]:

- Si X est A1 ou Y est B1 ALORS est C1
- Si X est A2 ou Y est B2 ALORS z est C2
- Si X est A3 ALORS z est C3

Où *x* est la température, *y* est l'humidité et *z* est la vitesse du ventilateur. Les ensembles flous sont :

- A_1 = faible, A_2 = moyenne, A_3 = élevée
- B_1 = sec, B_2 = humide
- C_1 = lent, C_2 = moyenne, C_3 = rapide

Les variables linguistiques ainsi que leur fonction d'appartenance sont données par la figure II.9.

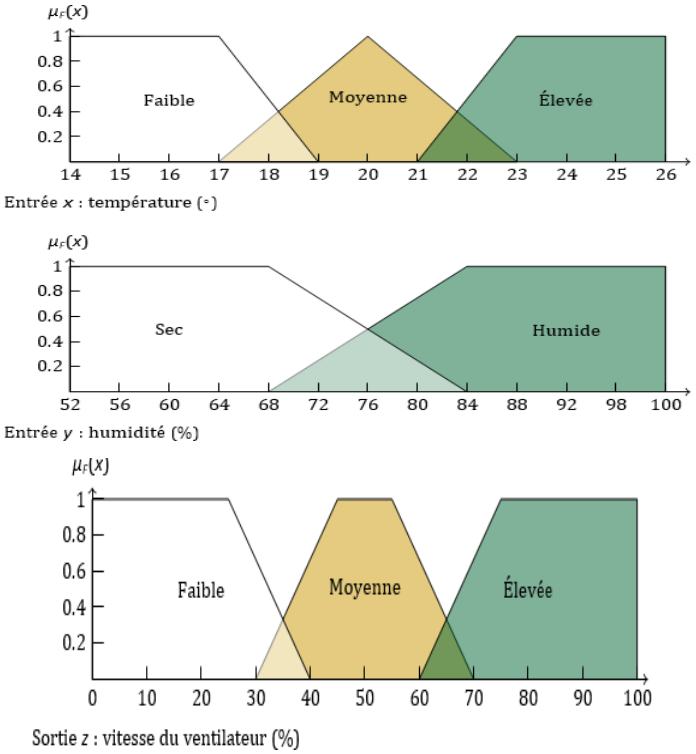


Figure II.8. Ensembles flous pour exemple d'inférence Mamdani

Supposons qu'il fait 18°C, et que l'humidité est de 80%. On applique ces entrées sur les fonctions d'appartenance pour déterminer l'appartenance à chaque variable. Une température de 18°C correspond à une appartenance de 0.5 à l'ensemble Faible ($\mu_{A1} = 0.5$) et une appartenance de 0.33 à l'ensemble moyenne ($\mu_{A2} = 0.33$).

Une humidité de 80% correspond à une appartenance de 0.25 à l'ensemble sec ($\mu_{B1} = 0.25$) et une appartenance de 0.75 a` l'ensemble humide ($\mu_{B2} = 0.75$)[20].

Évaluation des règles : Il faut évaluer les règles en fonction des entrées obtenues. On applique les opérateurs flous correspondants pour combiner les règles. Si une règle a plusieurs antécédents, un opérateur flou est utilisé pour obtenir un seul chiffre qui représente le résultat. Ce résultat est ensuite applique´ a` la fonction d'appartenance de la conséquence. Le résultat peut être produit par deux méthodes : coupure ou mise à l'échelle [9]. Si on reprend l'exemple précédent, on applique les valeurs d'appartenance à la règle 1 :

Si x est A_1 (0.5) OU y est B_1 (0.25), Alors z est C_1 (?)

- ***Méthode de maximum***

$$\mu C1 = max\,[\mu A1, \mu B1] = 0.5 \qquad (II.10)$$

Si on utilise la méthode de maximun, l'appartenance est 0.5. Pour la règle 2 :

$$Si\ x\ est\ A2\ (0.33) ET\ y\ est\ B2\ (0.75) ALORS\ z\ est\ C2\ (?) \qquad (II.11)$$

- ***Méthode de minimum***

$$\mu C1 = min[\mu A2, \mu B2] = 0.33 \qquad (II.12)$$

1) Si on utilise la méthode de minimum, l'appartenance est 0.33.
2) Alors, on obtient :

SI x est A_1 (0.5) OU y est B_1 (0.25) ALORS z est C_1 (0.5) SI x est A_2 (0.33) ET y est B_2 (0.75) ALORS z est C_2 (0.33)SI x est A_3 (0.0) ALORS z est C_3 (0.0)

II.2.8 Avantages et les inconvénients du réglage par logique floue

La commande floue présente plusieurs avantages. Elle est généralement robuste, ce qui garantit un bon fonctionnement même en présence de perturbations. Elle est également auto-adaptative, c'est-à-dire qu'elle peut s'adapter automatiquement aux variations du système à réguler. De plus, contrairement à d'autres méthodes de commande, elle ne nécessite pas de modèle mathématique du système à réguler. La théorie de la commande floue est simple et peut être appliquée à des systèmes complexes. Toutefois, malgré ses avantages, la commande floue présente certains inconvénients tels qu'une précision de réglage souvent peu élevée, une cohérence des inférences non garantie à priori, une technique de réglage totalement empirique et une dépendance directe des performances à l'expertise de l'utilisateur. En outre, il n'est pas possible de prouver la stabilité de la boucle de commande et il y a une possibilité d'apparition de cycles limites en raison du fonctionnement non linéaire[5].

II.3 Node-RED

II.3.1 Présentation de Node-RED

Node-RED est un outil polyvalent conçu spécifiquement pour créer des applications IoTen mettant l'accent sur la facilité d'utilisation grâce à la programmation visuelle. Grâce à son interface glisser-déposer, les développeurs peuvent assembler des flux de travail complexes à partir d'éléments de nœud simples, ce qui leur permet de développer et de déployer rapidement des solutions IoT sur diverses plates-formes. En tirant parti de Node.js, Node-RED tire parti d'un modèle basé sur les événements hautement évolutif, permettant une communication efficace entre les appareils tout en garantissant une latence minimale [15]

II.3.2 Composition de Node Red

Les nœuds de Node-Red sont composés d'un fichier HTML et d'un fichier JavaScript. Le fichier HTML indique à quoi le nœud doit ressembler, combien d'entrées / sorties doit utiliser. Le fichier JavaScript est responsable de la façon dont le nœud devrait fonctionner .Node-RED offre une variété des nœuds préinstallés prêt à utilise la figure II.2 illustre ces différents nœuds.

Il existe 7 nœuds d'entrée de base dans l'installation par défaut. Ces nœuds couvrent les mécanismes de communication de base utilisée principalement par les applications IoT. Cela s'étale a des protocoles Internet fondamentaux tels que UDP et TCP jusqu'à un niveau plus élevé tels que HTTP et les mécanismes MQTT de publication / souscription (publish/subscribe)[22]



Figure II.9 : Nœuds de Node-RED les nœuds de traitement et les nœuds de sortie (la figure II.9)



Figure II.10: composition d'un flux sur Node-RED

II.3.2.1 Nœuds d'entrée (input nodes)

Ces nœuds nous permettent d'entrer les données d'entrée dans le flux. Ils possèdent un carré gris sur le côté droit qui peut être connecté aux autres nœuds. En connectant ce nœud à un autre nœud, vous pouvez envoyer les données d'entrée[22].

II.3.2.2 Nœuds de sortie (output nodes)

Output nodes permettent d'envoyer les données vers le monde extérieur, vers d'autres services, par exemple MQTT. Il y a un carré gris sur le côté gauche auquel ils peuvent être connectés à un nœud d'entrée ou à une fonction. Si on ne souhaite pas envoyer les données à des nœuds de messagerie, on peut simplement déboguer ces messages en utilisant l'onglet «deploy» dansle coin supérieur droit de l'éditeur[22].

Nœuds de traitement :

Les nœuds de traitement sont utilisés pour traiter les données dans lesquelles ils ont un point de terminaison d'entrée avec un ou plusieurs points de terminaison de sortie. Ils sont utilisés pour transformer les types de données, écrire les codes personnalisés et déclencher un message

II.3.2.3 Tableau de bord (Dashboard)

Node-RED Dashboard est un module qui fournit un ensemble de nœuds dans Node-RED pour créer rapidement un tableau de bord de données en temps réel.il est connu aussi sous forme d'une palette qui se compose de plusieurs nœuds comme présente dans la figure II.11.[22].

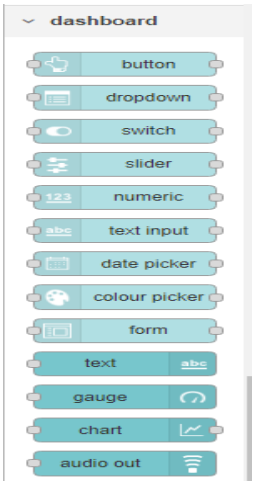


Figure II.11: nœuds de Dashboard

Les nœuds de tableau de bord de Node-RED fournissent une large gamme de composants d'interface utilisateur, tels que des graphiques, des jauges, du texte de base, des curseurs et des entrées. Ces composants peuvent être utilisés pour créer des tableaux de bord IoT debase. Cependant, il y aura des moments où vous aurez besoin de quelque chose de plus personnalisé. Le nœud Modèle est la solution pour cela. Il existe dans node-RED une catégorie de nœuds du tableau de bord (Dashboard) comme le montre la figure (II.13).



Figure II.12 : Éléments d'un tableau de bord (Dashboard)

II.3.2.4 GPIO

GPIO (General Purpose Input/Output) est l'acronyme utilisé pour désigner les broches

d'entrée/sortie à usage général sur une carte électronique, comme un microcontrôleur ou un Raspberry Pi. Ces broches GPIO peuvent être configurées pour agir soit en tant qu'entrées, soit en tant que sorties, permettant ainsi à l'appareil de communiquer avec d'autres composants électroniques

Lorsqu'une broche GPIO est configurée en tant qu'entrée, elle peut recevoir des signaux provenant de capteurs, de boutons ou d'autres sources externes. Lorsqu'elle est configurée en tant que sortie, elle peut envoyer des signaux à des actionneurs, tels que des LED, des moteurs ou d'autres dispositifs électroniques.

La configuration et la manipulation des broches GPIO dépendent du matériel et du langage de programmation utilisé. Sur un Raspberry Pi, par exemple, vous pouvez accéder aux broches GPIO en utilisant des bibliothèques spécifiques telles que RPi.GPIO en Python.

Les broches GPIO offrent une grande flexibilité et sont largement utilisées dans le domaine de l'électronique et de l'informatique embarquée pour contrôler et interagir avec divers composants et dispositifs externes. La palette "GPIO" est représenter sur la figure II.14 suivantes.



Figure II.14 : Représentation de la palette de GPIO

II.4 Protocole MQTT

II.4.1 Introduction

MQTT est un protocole de transport de messagerie de publication / abonnement (publish/subscribe) Client Server à la fois extrêmement simple et léger idéal pour l'Internet des objets. MQTT est une abréviation du <Message Queue Telemetry Transport>, autrement dit MQTT est un protocole M2M (Machine-to-Machine) open source

L'efficacité de ce protocole est de plus en plus approuvée dans de célèbres applications comme la messagerie sur le réseau social Facebook. Maintenant, son application pour les applications fondées sur les communications M2M dans l'IoT est largement investiguée.

[22].

MQTT a été Créé en 1999 par le Dr Andy Stanford-Clark d'IBM (International Business Machines Corporation), et Arlen Nipper d'Arcom (maintenant Eurotech), en août 2010, IBM apublié le protocole MQTT V3.1 spécification, qui a été utilisé dans le développement de futures normes. Eclipse, et plus précisément le projet Eclipse Paho, a été le premier à déplacerMQTT sur son chemin de normalisation. En octobre 2014, MQTT est devenu une norme OASIS [22].lorsque leur cas d'utilisation était de créer un protocole avec les objectifs suivants: [22]

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

Par la suite, la version 3.1 a été libérée de droits en 2010, et le protocole est standardisé par l'OASIS en 2014 dans sa version 3.1.1. [22]. Ces objectifs sont toujours au cœur de MQTT tandis que l'attention s'est déplacé des systèmes intégrés propriétaires aux cas d'usage de l'Internet des objets. Il y a souvent une confusion quant à la signification appropriée de l'abréviation MQTT. La réponse courte est que MQTT n'a officiellement plus d'acronyme, c'est juste MQTT. La réponse longue histoire raconte que l'ancien acronyme signifiait "Transport de télémétrie MQ" (Message Queuing Telemetry Transport), alors que MQ fait référence à "MQ Séries", un produit développé par IBM qui prend notamment en charge MQTT. Le protocole a été nommé après quand il a été inventé 1999.Il devient un standard ISO (ISO/IEC 20922) en 2016 [22].

L'efficacité du protocole MQTT a été approuvée pour certaines applications comme : les applications médicales et industrielles. Il s'est avéré même qu'il représente une bonne solution pour les applications mobiles dans des environnements contraints car il présente une faible empreinte mémoire, une faible consommation d'énergie avec une meilleure distributionde l'information aux destinataires. Car ses qualités font de lui un excellent candidat pour les communications au sein de l'Internet des Objets , mais il a également été utilisé avec succès par Facebook pour être au cœur de leur système de messagerie à cause de sa performance , comme il est utilisé par Amazon web services dans ses passerelles pour permettre la diffusion des données à plusieurs abonnés sur un sujet donné . La norme MQTT décrit le comportement attendu des brokers et clients MQTT, ainsi que les détails du format binaire employé pour les échanges [22].

II.4.2 Message Queuing Telemetry Transport

Le protocole Publish/Subscribe asynchrone fonctionne avec TCP/IP. C'est un protocole M2M (Machine to Machine) largement utilisé dans l'IOT. Le protocole MQTT déjà mentionnée auparavant est un protocole à message extrêmement léger. C'est pour cette raison qu'il est adopté dans l'écosystème IOT. Presque toutes les plates-formes IOT prennent en charge, le protocole MQTT peut envoyer et recevoir des données à partir des objets intelligents. Trois niveaux de qualité de service sont prévus pour la transmission des messages :

- Fire and Forget : le message est envoyé une fois et aucun acquittement n'est exigé.
- Delivered at least once : le message est envoyé au moins une fois et un acquittementest exigé.
- Deliveredexactly once : un mécanisme en quatre temps assure la délivrance dumessage une seule et unique fois aquitement si oui ou non.

La figure II.15 représente un Exemple de communication machine-à-machine suivant le protocole MQTT.

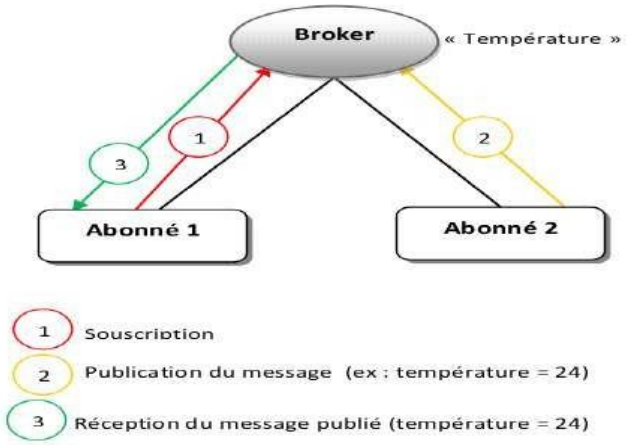


Figure II.14 : Exemple de communication machine-à-machine suivant le protocole MQTT.

Du point de vue architectural, les nœuds capteurs sont des publieurs qui se connectent tous à une entité centrale dite broker. Chaque message est publié au niveau du broker dans une rubrique qui convient au type de la donnée contenue dans le message. Les abonnés peuvent

souscrire à plusieurs rubriques. A chaque fois qu'un nouveau message est publié dans l'une des rubriques d'intérêts, il sera immédiatement diffusé aux abonnés intéressés. Comme illustre dans la figure15. La figure II.16 ci-dessous illustre un deuxième exemple sur communication machine-à-machine suivant le protocole MQTT.

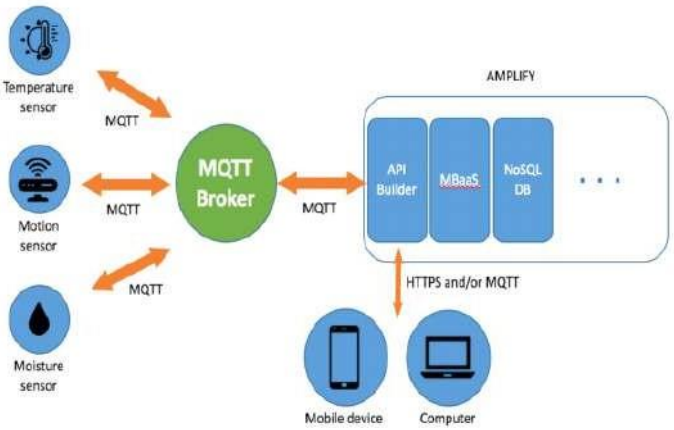


Figure II.15 : exemple sur communication machine-à-machine suivant le protocole MQTT.

II.4.3 Mode de fonctionnement

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

Ce pattern consiste à organiser les messages par sujets (topics) et à gérer leur distribution selon le principe d'abonnement. Ainsi, toute application peut publier ses messages sur les topics de son choix, tandis que les applications intéressées par les messages d'un topic donné peuvent s'abonner pour recevoir tous les nouveaux messages publiés sur ce topic[12].

II.4.4 C'est quoi le logiciel Mosquitto

Mosquitto est un logiciel open source qui implémente le protocole MQTT (Message Queuing Telemetry Transport). MQTT est un protocole de messagerie léger, conçu pour la communication entre des appareils connectés à des réseaux à faible bande passante.

Mosquitto agit en tant que broker MQTT, ce qui signifie qu'il sert d'intermédiaire entre les appareils qui publient des messages et ceux qui les reçoivent. Les appareils clients utilisent le protocole MQTT pour se connecter à Mosquitto et échanger des messages avec d'autres clients. Ce logiciel offre des fonctionnalités telles que la publication et la souscription de messages, la gestion des sujets (topics) MQTT, la prise en charge de différents niveaux de qualité de service (QoS) pour les messages, l'authentification des clients, le chiffrement des communications à l'aide du protocole SSL/TLS, et bien plus encore.

Mosquitto est utilisé dans de nombreux scénarios d'Internet des objets (IoT) où des appareils ont besoin de communiquer entre eux de manière efficace et fiable. Il est écrit en langage C et est disponible sur plusieurs plateformes, notamment Linux, Windows, macOS, et d'autres.

II.5 Raspberry

Raspberry Pi (figure II.17) est une série de petits ordinateurs à carte unique (SBC) développés au Royaume-Uni par la Raspberry Pi Foundation en association avec Broadcom Le projet



Figure II.16 : Carte Raspberry Pi 3

Raspberry Pi était à l'origine axé sur la promotion de l'enseignement de l'informatique de base dans les écoles. Le modèle original est devenu plus populaire que prévu, se vendant en dehors de son marché cible pour des utilisations telles que la robotique. Il est largement utilisé dans de nombreux domaines, tels que la surveillance météorologique, en raison de son faible coût, de sa modularité et de sa conception ouverte. Il est généralement utilisé par les amateurs d'informatique et d'électronique, en raison de son adoption des normes HDMI et USB[15]

Les interfaces matérielles pour Raspberry Pi 2 et Raspberry Pi 3 sont exposées via l'en-tête **J8** de 40 broches sur la carte. Les fonctionnalités incluent :

- 24x - Broches GPIO
- 1x - UARTs série (RPi3 inclut uniquement mini UART)
- 2x - Bus SPI
- 1x - Bus I2C
- 2x - Broches d'alimentation 5V
- 2x - 3.3V broches d'alimentation
- 8x - Broches au sol

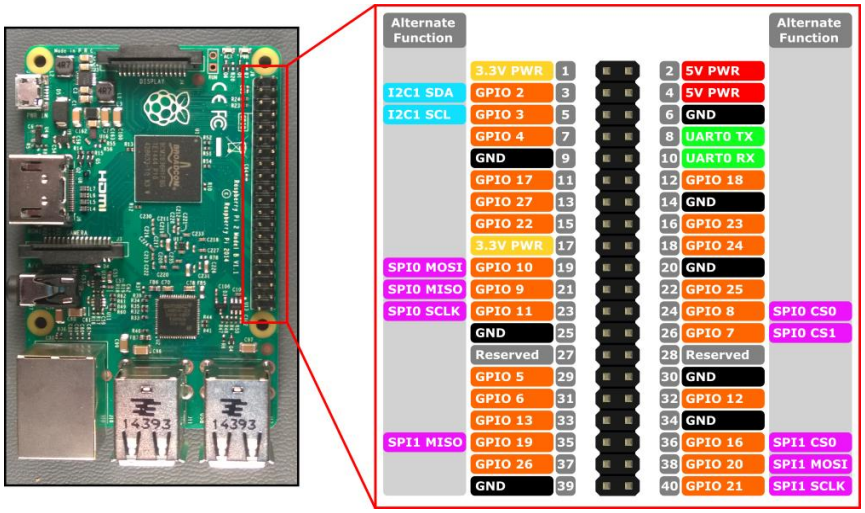


Figure II.17 : Brochage de Raspberry

II.6 Conclusion

Ce chapitre a exploré différentes solutions matérielles et logicielles utilisées dans le domaine de l'informatique et de l'automatisation. Nous avons examiné plusieurs technologies clés, notamment la logique floue, Node-RED, Raspberry Pi, MQTT et Mosquitto, qui offrent des fonctionnalités essentielles pour la mise en œuvre de systèmes intelligents et interconnectés.

La logique floue s'est révélée être une approche puissante pour traiter les concepts vagues et incertains. Elle trouve des applications dans divers domaines, allant du contrôle de systèmes à l'intelligence artificielle et à la prise de décision. En utilisant des ensembles flous et des règles d'inférence, la logique floue permet de gérer des informations ambiguës de manière efficace.

Node-RED, quant à lui, offre un environnement de développement visuel qui simplifie la création de flux de données et l'automatisation des tâches. Grâce à une interface conviviale

basée sur des nœuds interconnectés, il permet de configurer et de connecter facilement différents appareils matériels et logiciels, offrant ainsi une grande flexibilité pour la réalisation de projets d'automatisation.

Raspberry Pi, un ordinateur monocarte abordable et polyvalent, se positionne comme une plateforme matérielle idéale pour les projets d'automatisation et d'Internet des objets (IoT). Avec sa faible consommation d'énergie et sa capacité à prendre en charge divers capteurs, actionneurs et interfaces de communication, il permet la mise en place de systèmes intelligents et connectés à moindre coût.

Enfin, MQTT et Mosquitto jouent un rôle central dans la communication entre les appareils IoT. MQTT, un protocole de messagerie léger basé sur le modèle de publication/souscription, permet un échange de données efficace et fiable entre les nœuds d'un réseau IoT. Mosquitto, un broker MQTT open-source, assure le routage des messages et la gestion des abonnements, facilitant ainsi la mise en place de réseaux de communication sécurisés et évolutifs.

Chapitre III

Réalisation pratique

III.1 Introduction

Dans la suite de ce projet, nous concentrons sur la partie pratique y compris l'architecture et le fonctionnement du système de feu de circulation basé sur Node-RED, Raspberry Pi et Python. Cette partie pratique permet de démontrer l'efficacité et l'impact de notre approche dans la gestion du trafic et la sécurité routière. Nous présentons également les différentes étapes de développement, les défis rencontrés et les résultats obtenus. Ce projet démontre l'application pratique de ces technologies émergentes dans le domaine de la gestion intelligente du trafic, ouvrant ainsi la voie à de nouvelles possibilités d'amélioration des infrastructures urbaines et de la mobilité des citoyens.

III.2 Position de problème

La figure III.1 représente l'intersection considérée. Cette dernière est composée de 8 voies et 4 phases ou chaque phase comporte 2 voies. Si une phase est en vert, les feux des deux voies correspondantes sont au vert et toutes les autres voies sont en rouge.

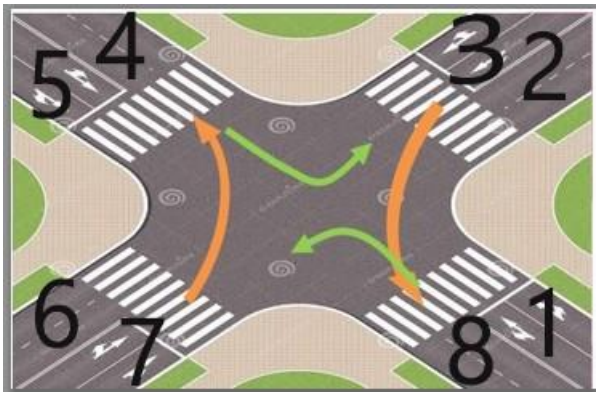


Figure III.1. Intersection à quatre chemins avec numérotation des voies

Dans ce modèle montre carrefour qui est typiquement utilisé dans la littérature afin de valider un modèle : une intersection composée de quatre directions avec un nombre fixé de voies. Ici, les voies pour tourner à gauche sont séparées des voies allant tout droit ou à droite, ces deux derniers

Alors on va expliquer les quatre phases :

- La première phase : Les véhicules transportent du nord au sud (à droite), du nord à l'ouest (virage), du sud au nord (à droite) et du sud à l'est (virage).
- La deuxième Phase : les véhicules circulent d'ouest en est (droit), l'ouest Vers le sud (tournant), d'est en ouest (droit) et d'est en nord (tournant).
- La troisième phase est celle des véhicules qui circulent du nord vers l'est (tournant) et du sud à l'ouest (tournant).
- Le quatrième Phase est des véhicules qui se déplacent d'ouest en nord (tournant) et de l'est au sud (tournant).

La figure III.2 montre les quatre phase qui peuvent exister dans un cycle ou dans chaque phase, on peut remarquer que deux voies passent en vert et les autres en rouge.

Phase	Phase 1	Phase 2	Phase 3	Phase 4
Dessin				
Voies en vert	1 et 5	2 et 6	3 et 7	4 et 8

Figure III.2. Phases de croisement à quatre voies

III.3 Cahier de charge

Le but principal d'un système intelligent de contrôle de feu de circulation d'une intersection isolée est d'améliorer la gestion du trafic et la sécurité routière. Voici les objectifs spécifiques que ce système vise à atteindre :

III.3.1 Optimisation du trafic

Le système cherche à optimiser la fluidité du trafic en ajustant les cycles de feux de circulation en fonction des conditions actuelles du trafic. En optimisant la synchronisation des feux de circulation, il permet de réduire les temps d'attente des véhicules, d'atténuer la congestion et d'améliorer l'efficacité globale de l'intersection.

III.3.2 Réduction des temps d'attente

En adaptant les durées des feux de circulation en temps réel en fonction du flux de trafic détecté, le système vise à réduire les temps d'attente des véhicules aux feux rouges. Cela contribue à améliorer la fluidité du trafic, à réduire les retards et à offrir une expérience de conduite plus agréable aux usagers de la route.

III.3.3 Sécurité routière accrue

En utilisant des données de détection du trafic, le système peut identifier les moments où des conditions de trafic potentiellement dangereuses se produisent, telles qu'une accumulation excessive de véhicules ou des comportements de conduite imprudents. Il peut alors ajuster les feux de circulation pour réduire les risques d'accidents et améliorer la sécurité des usagers de la route.

III.4 Moyens hardware et software utilisés

La solution proposée utilise des plusieurs moyens hardware et software

III.4.1 Moyens Hardware

- Carte Raspberry pi 3 B+.
- Les relais
- Les LEDs
- Carte mémoire.

- Les fils électriques

III.4.2 Moyens Software

- Outil de développement « Node Red » basé sur le langage « JavaScript ».
- Langage « Python ».

III.5 Principe de contrôle

Le contrôle des feux de l'intersection est basé sur quatre étapes essentielles

A) Détection du trafic : Le système utilise une base de données, pour collecter des données sur le trafic dans l'intersection. Ces bases de données peuvent détecter la présence de véhicules, leur vitesse, leur nombre, etc.

B) Traitement des données : Les données collectées sont ensuite transmises à un module de traitement. Ce module analyse les données en temps réel pour obtenir des informations sur le trafic actuel

C) Prise de décision : Sur la base des données analysées, un système de contrôle centralisé prend des décisions sur la durée des feux de circulation pour chaque voie de l'intersection. Des algorithmes avancés peuvent être utilisés pour optimiser ces décisions en fonction de critères tels que la fluidité du trafic, la réduction des temps d'attente et la sécurité routière.

D) Contrôle des feux de circulation : Une fois les décisions prises, le système intelligent contrôle les feux de circulation de manière adaptative. Il ajuste les cycles de feux, la durée du feu vert, le temps de transition entre les feux, etc., en fonction des besoins actuels du trafic détectés par les capteurs et des décisions prises par le système de contrôle centralisé.

III.6 Algorithme de fonctionnement d'un cycle

Le pseudo-algorithme suivant donne une idée sur le fonctionnement du contrôleur intelligent à implémenter

1) Début

2) Lecteur les données : Nombre de véhicule dans chaque voie (NV), et le débit dans chaque voie (D)

3) Calcul les degrés d'urgence de chaque phase

4) Détermination de phase correspondant au degré d'urgence maximale
5) Si le degré d'urgence est supérieur ou égal à 0,6, activer cette phase et calculer son temps vert en fonction du D et NV correspondant
6) Une phase peut passer en vert au maximum deux fois selon les conditions établies
7) Allume les LEDs

III.7 Contrôleur par logique floue

Notre contrôleur a deux logique floues reçoit les données relatives aux nombres des véhicules (NV) dans l'intersection et les débit des véhicules (D) et délivre à ses sortie la phase à allumer en vert et par conséquent les phases à allumer en rouge ainsi que la durée de la phase en vert.

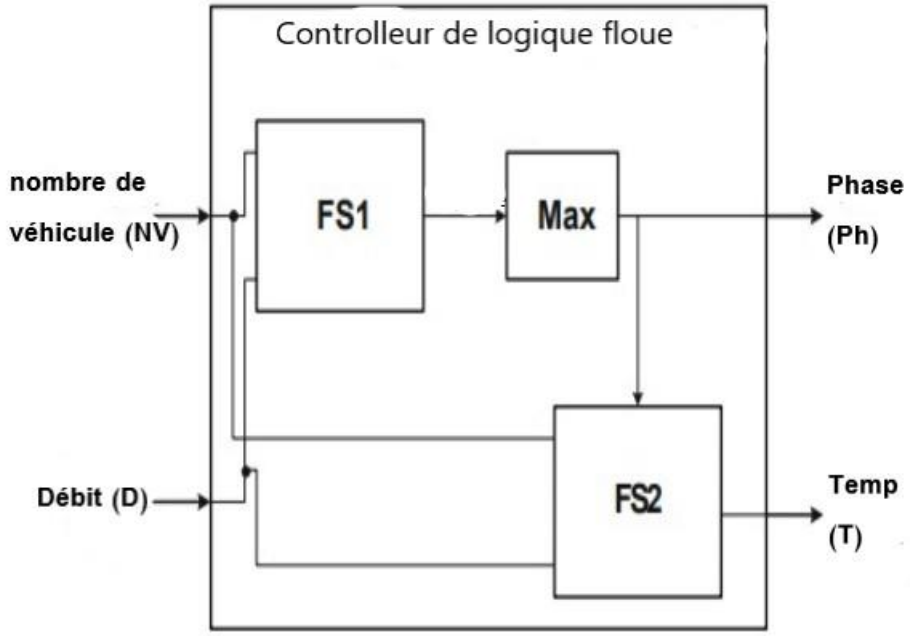


Figure III.3. Structure du contrôleur par la logique floue

III.7.1 Système Flou 1

Le système floue 1 permet de calculer la phase, il possède deux entrées (débit(véh/min), nombre des véhicules) et une sortie (la phase), Comme la figure III.4 ci-dessus.

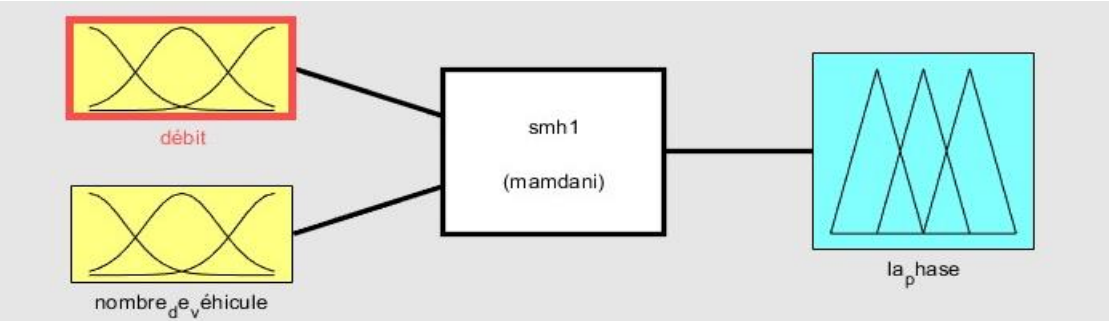


Figure III.4. Système floue1

III.7.1.1 Fonction d'appartenance

En premier lieu on va définir l'univers de discours de chaque variable :

- ✓ Débit (véh/min) (D) : Entré 1 : sept classes.
- ✓ Nombre de véhicule (NV) : Entré 2 : sept classes.
- ✓ La phase (ph) : Sortie : sept classes.

Les figures (III.5, III.6) représentent les fonctions d'appartenance de la variable d'entrée.

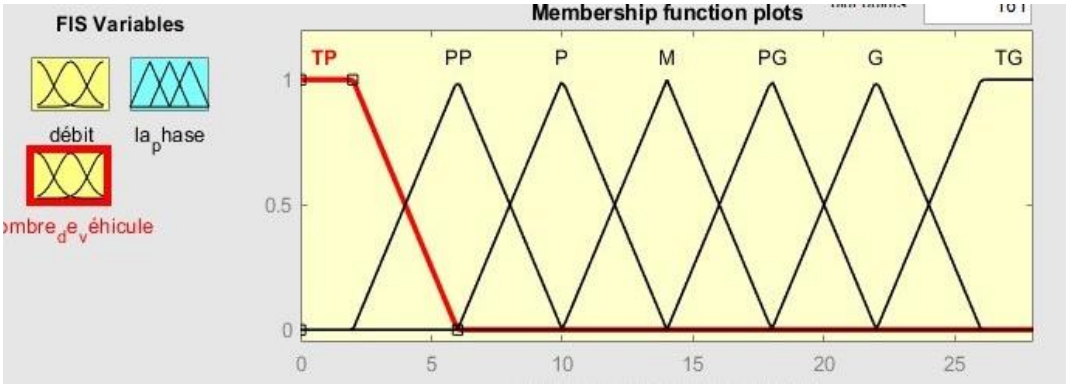


Figure III.5. Présentation le nombre des véhicules dans l'intersection

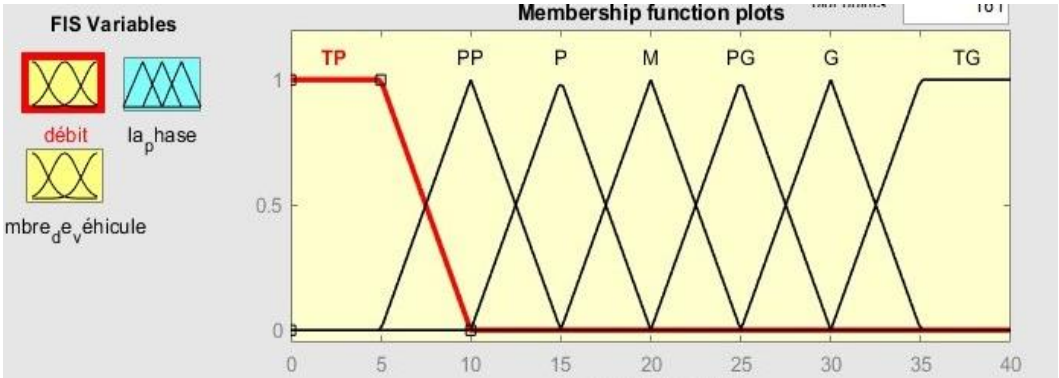


Figure III.6. Présentation le débit des véhicules dans l'intersection

La figure III.7 représente les fonctions d'appartenance de la variable de sortie.

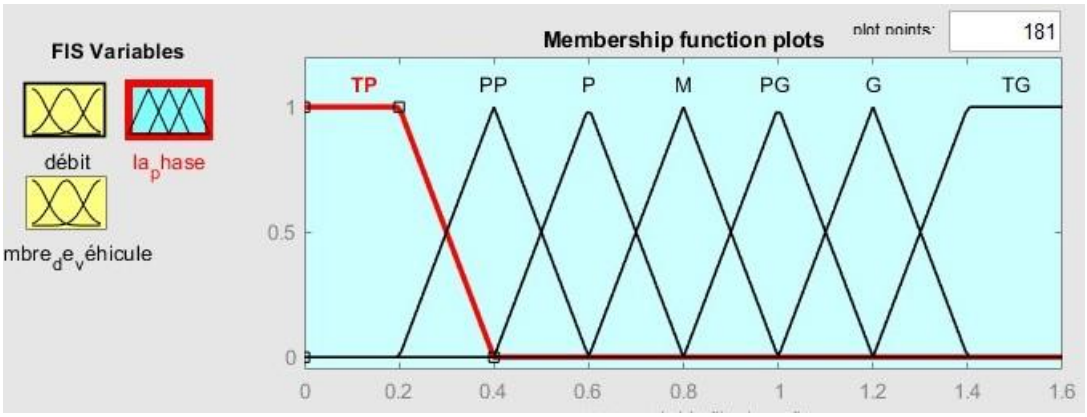


Figure III.7. Variable sortie de la phase

III.7.1.2 Variable linguistiques

- **Entré 1 : débit**

- Débit : [0 40]
- Débit : {TP, PP, P ,M ,PG, G, TG}
- TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grand

M : moyen

G : grand

TG : très grand

- **Entré 2 : nombre des véhicules**

- Nombre-de-véhicule : [0 28]
- Nombre-de-véhicule : {TP, PP, P, M, PG, G,TG}

TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grand

G : grand

TG : très grand

- **Sortie : la phase**

- la phase : [0 1.6]
- La phase : {TP, PP, P, M, PG, G, TG}

TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grand

G : grand

TG : très grand

III.7.1.3 Règles floues

Le tableau III.1. Donne les règles floues qui régissant le système flou1 (SF1)

		Débit (véh/min)						
		TP	PP	P	M	PG	G	TG
Nombre de	TP	TP	PP	PP	P	P	M	M
	PP	PP	PP	P	P	M	M	PG

	P	PP	P	P	M	M	PG	PG
	M	P	P	M	M	PG	PG	G
	PG	P	M	M	PG	PG	G	G
	G	M	M	M	PG	G	G	TG
	TG	M	PG	PG	G	G	TG	TG

III.7.2 Système Flou 2

Le système floue 2 détermine la durée de temps de la phase sélectionnée par le système floue 1. Il possède deux entrées le nombre de véhicule de la phase sélectionnée et leur débit et une seule sortie qui est la durée de temps vert (temps) comme voire sur la figure III.8

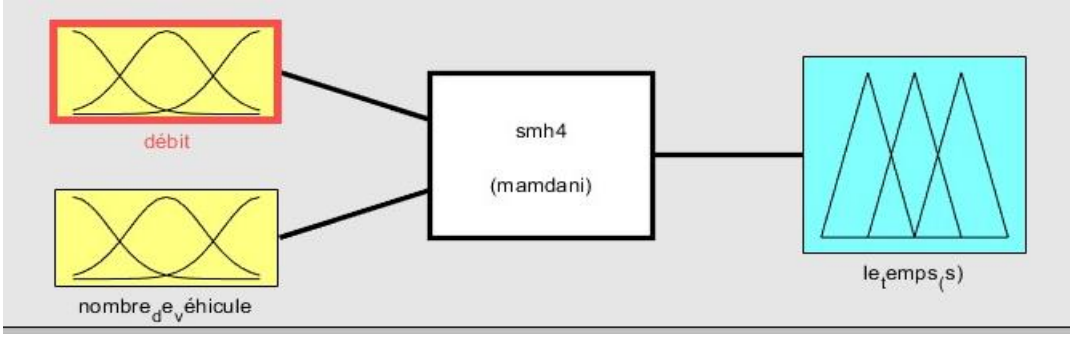


Figure III.8. Système floue 2

III.7.2.1 Fonction d'appartenance

En premier lieu on va définir l'univers de discours de chaque variable :

- ✓ débit /min (D): Entré1 : sept classes.
- ✓ Nombre de véhicule : Entré2 : sept classes.
- ✓ Le temps : Sortie : sept classes

Les figures (III.9, III.10) représentent les fonctions d'appartenance de la variable d'entrée. Et la figure III.11 représente fonction d'appartenance de variable de sortie

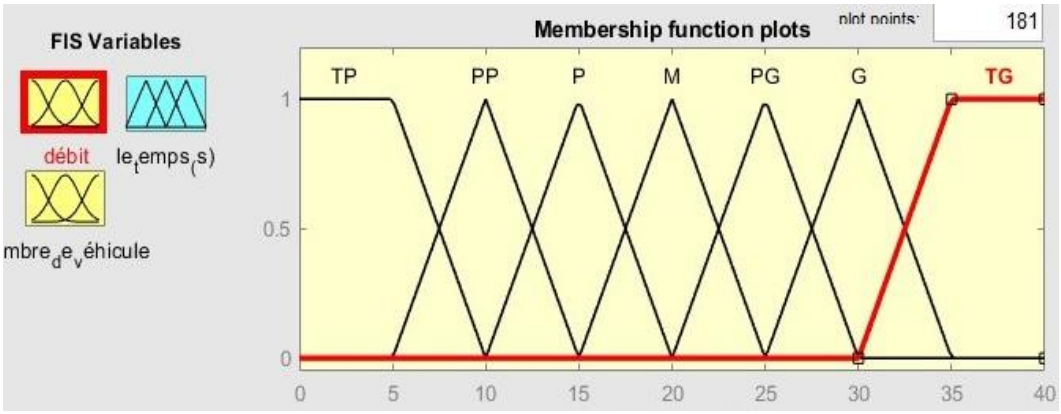


Figure III.9. Présentation le nombre des véhicules dans l'intersection

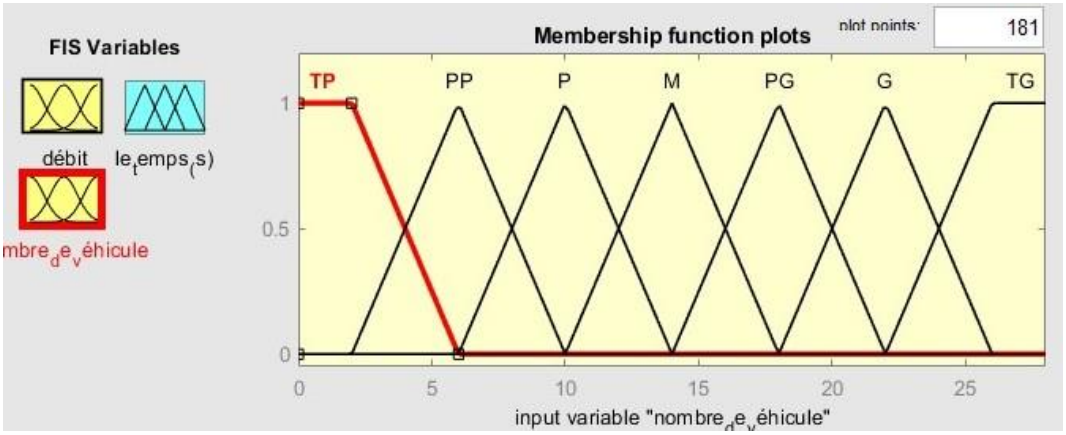


Figure III.10 : présentation le débit des véhicules dans l'intersection

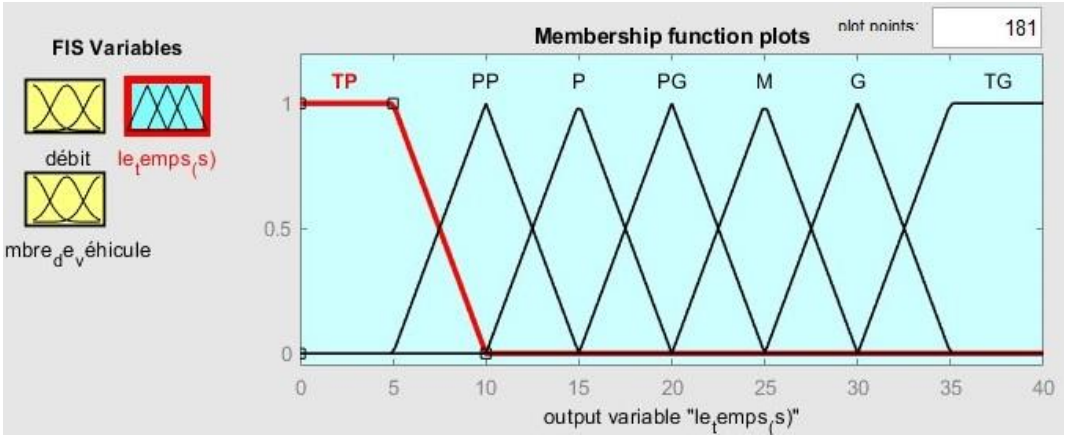


Figure III.11 : la durée de temps

III.7.2.2 Variable linguistiques

- **Entré 1 : débit**

- Débit : [0 40]
- Débit : {TP, PP, P, M, PG, G, TG}

TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grande

G : grand

TG : très grand

- **Entré 2 : nombre de véhicule**

- Nombre-de-véhicule : [0 28]
- Nombre-de-véhicule : {TP, PP, P, M, PG, G, TG}

TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grand

G : grand

TG : très grand

- **Sortie : la durée de temps : [0 40]**

- Temps : {TP, PP, P, M, TG, G, TG}

TP : très petit

PP : proche de petit

P : petit

M : moyen

PG : proche de grand

G : grand

TG : très grand

III.7.2.3 Règles floues

Le tableau III.2 représente les règles floues qui régissant le SF2.

Tableau III.2. Présentations des règles floue

		Débit (véh/min)						
		TP	PP	P	M	PG	G	TG
Nombre de véhicules	TP	TP	PP	PP	P	P	M	M
	PP	PP	PP	P	P	M	M	PG
	P	PP	P	P	M	M	PG	PG
	M	P	P	M	M	PG	PG	G
	PG	P	M	M	PG	PG	G	G
	G	M	M	M	PG	G	G	TG
	TG	M	PG	PG	G	G	TG	TG

III.8 Tests de contrôleur

En utilisant le langage de programmation Python en utilisant le package "simpful" pour la logique floue. Pour les données relatives au "Débits" et "Nombres de véhicules"

nous avons utilisés des données stockées dans fichier Excel Tableau III.3. Notre contrôleur accorde la priorité à la phase la plus urgente, et la durée du feu vert dépend du nombre de véhicules (NV) et du débit (D)

Tableau III.3 Présentation une base de données d'entrée pour les sexes cycle

Cycle1	Phase	Phase4	Phase1	Phase3	Phase 3	Phase 2	Phase 1			Durée cycle totale
	Durée de vert	30s	32s	27s	36s	29s	26s			180 s
Cycle 2	Les phase	Phase1	Phase 4	Phase1	Phase2	Phase3	Phase2	Phase4	Phase3	
	Durée de vert	28s	35s	36s	31s	20s	30s	32s	24s	236 s
Cycle3	Les phase	Phase4	Phase1	Phase1	Phase3	Phase4	Phase2	Phase1		
	Durée de vert	30s	29s	34s	19s	30s	21s	23s		186 s
Cycle 4	Les phase	Phase 1	Phase3	Phase4	Phase2	Phase4	Phase3	Phase1		
	Durée de vert	20s	32s	36s	30s	34s	29s	28s		209 s
Cycle 5	Les phase	Phase 4	Phase1	Phase3	Phase1	Phase3	Phase4	Phase2		
	Durée de vert	27s	22s	33s	30s	35s	26s	30s		203 s
Cycle 6	Les phase	Phase 4	Phase1	Phase3	Phase1	Phase3	Phase4	Phase 2		

	Durée de vert	28s	29s	20s	23s	35s	32s	36s		203 s

Le tableau III.4 donne les résultats de l'exécution du programme python indiquant les durées de temps vert attribuées à chaque phase du cycle de contrôle.

Tableau III.4. Présentation des résultats des cycles

		Voie1	**Voie2**	**Voie3**	**Voie4**	**Voie5**	**Voie6**	**Voie 7**	**Voie8**
Cycle1	**NV**	19	24	15	9	13	25	20	28
	D	27	23	25	1	37	19	17	18
Cycle2	**NV**	22	19	35	4	17	19	5	10
	D	7	23	18	17	34	28	6	0
Cycle3	**NV**	14	16	29	20	10	15	16	9
	D	31	11	3	10	35	3	19	0
Cycle4	**NV**	11	16	7	11	20	19	5	4
	D	17	25	15	4	34	16	14	26
Cycle5	**NV**	31	1	9	5	2	10	36	21
	D	35	22	15	2	35	18	16	8
Cycle6	**NV**	23	11	26	25	8	0	18	2
	D	38	17	14	12	10	20	3	5

III.9 Fonctionnement global de système

Le système de feux de circulation intelligent que j'ai développé repose sur un ensemble de contrôleurs d'intersection et une communication sans fil efficace. Chaque intersection est équipée d'un contrôleur dédié chargé de gérer les feux de circulation

spécifiques à cet emplacement. La communication entre le contrôleur principal et les contrôleurs d'intersection est établie grâce à une communication WIFI à base de MQTT, qui permet une transmission bidirectionnelle des instructions par radiofréquence. Le contrôleur principal, quant à lui, assume la responsabilité de la gestion globale du système. En utilisant les données enregistrées dans un fichier Excel, il analyse le trafic, prend des décisions éclairées et génère les instructions appropriées pour chaque contrôleur d'intersection. Ce système offre ainsi une coordination optimisée des feux de circulation, contribuant à une meilleure fluidité du trafic et à une réduction des temps d'attente aux intersections. Nous avons la figure III.12 représente le fonctionnement global de système ci-dessus.

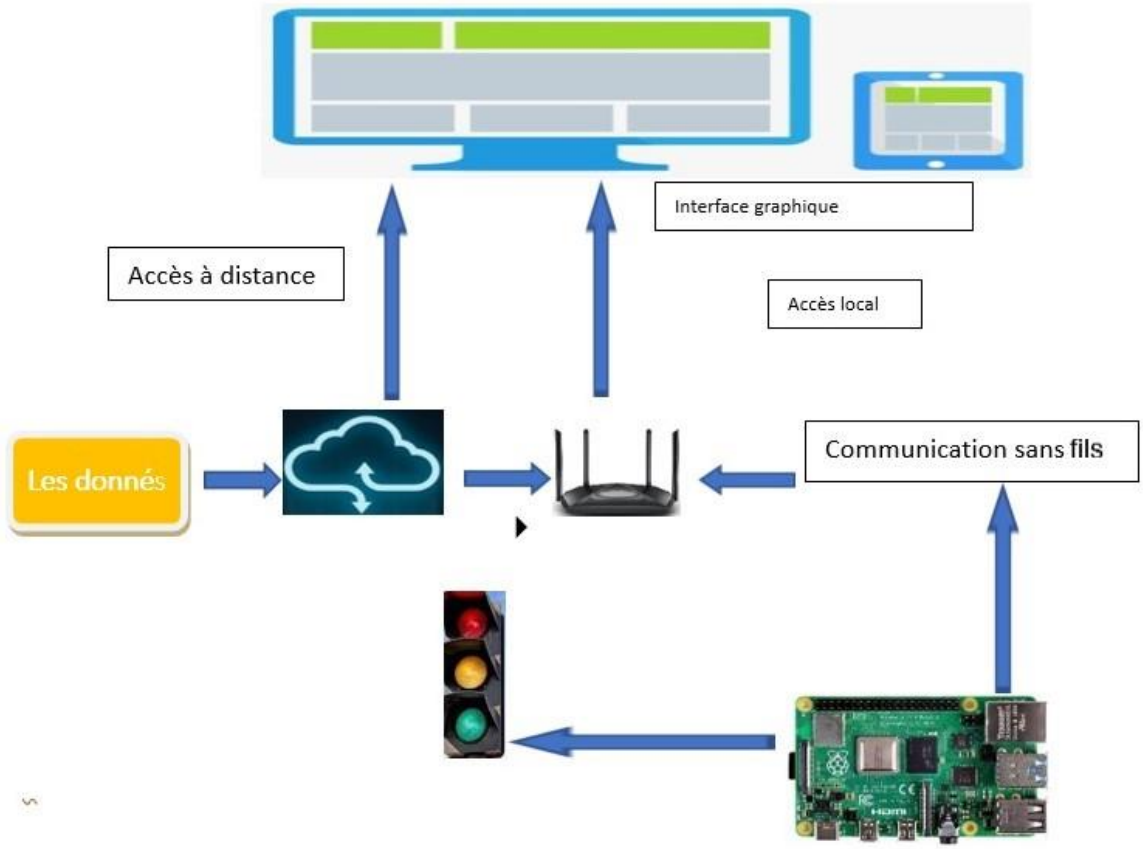


Figure III.12. Mise en place des matériels du système

Le programme développe sous Python est permet de recevoir les données transmises par le Raspberry Pi pour contrôler les LEDs. À chaque cycle terminé, Python reçoit les données de Raspberry via une communication wifi à base de MQTT. Ces données sont le nombre de véhicules et le débit pour chaque phase. Ensuite la durée de la phase sélectionnée par le

système Flou 1 est calculée par le système flou 2. Ces données sont utilisées pour gérer l'allumage et la temporisation des LEDs de l'intersection. Les états des LEDs ainsi que la durée de la phase qui est en vert sont affichés sur une IHM local (Raspberry) et au niveau du serveur.

III.10 Interface de supervision :

Dans notre projet, nous avons développé une interface sous la forme d'une application web accessible depuis un navigateur. Pour la mise en place de cette interface, nous avons opté pour l'utilisation de Node-RED, une solution extrêmement intéressante. Node-RED permet de créer des applications adaptées à l'Internet des objets (IoT) en facilitant la connexion entre les différents composants du système.

L'un des avantages de Node-RED réside dans sa programmation visuelle. En effet, il offre un ensemble de blocs prédéfinis et configurables, ce qui simplifie considérablement le processus de développement. En combinant les différents nœuds connectés tels que les nœuds d'entrée, de traitement et de sortie, nous créons ce que l'on appelle un flux (ou flow en anglais).

Pour la création de l'interface de supervision, nous utilisons spécifiquement les nœuds de tableau de bord de Node-RED (Node-RED Dashboard). Ces nœuds nous permettent d'afficher les données collectées à l'aide d'éléments graphiques et interactifs fournis par Node-RED. Cela nous offre une interface utilisateur conviviale et intuitive pour visualiser les informations et interagir avec le système de feux de circulation intelligent.

III.10.1 Lancement de Node-RED sur Raspberry Pi

On doit utiliser l'url donné par le terminal lors de lancement de « Node-RED» «http://ip_raspberry:1880 ». Comme elle est présenté dans la figure (III.13) et la figure (III.14).



Figure III.13. Représentation de démarrage de Node-Red

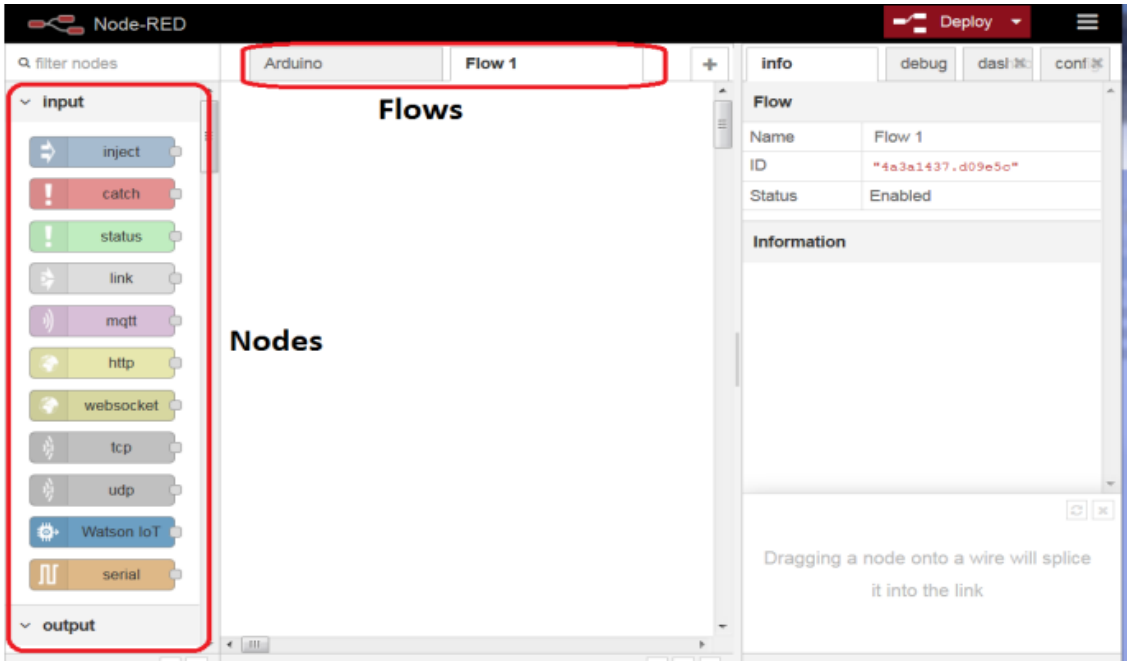


Figure III.14. Plateforme de développement Node-RED

Node-RED est livré avec un ensemble de nœuds de base qui offrent déjà un large éventail de fonctionnalités. Cependant, pour créer une interface homme-machine conviviale, il est nécessaire d'ajouter des nœuds de tableau de bord (Dashboard). Avant de pouvoir les utiliser, il est important d'installer le gestionnaire de nœuds requis.

En doit installer les nouveaux nœuds comme les figures III.14. Et III.15

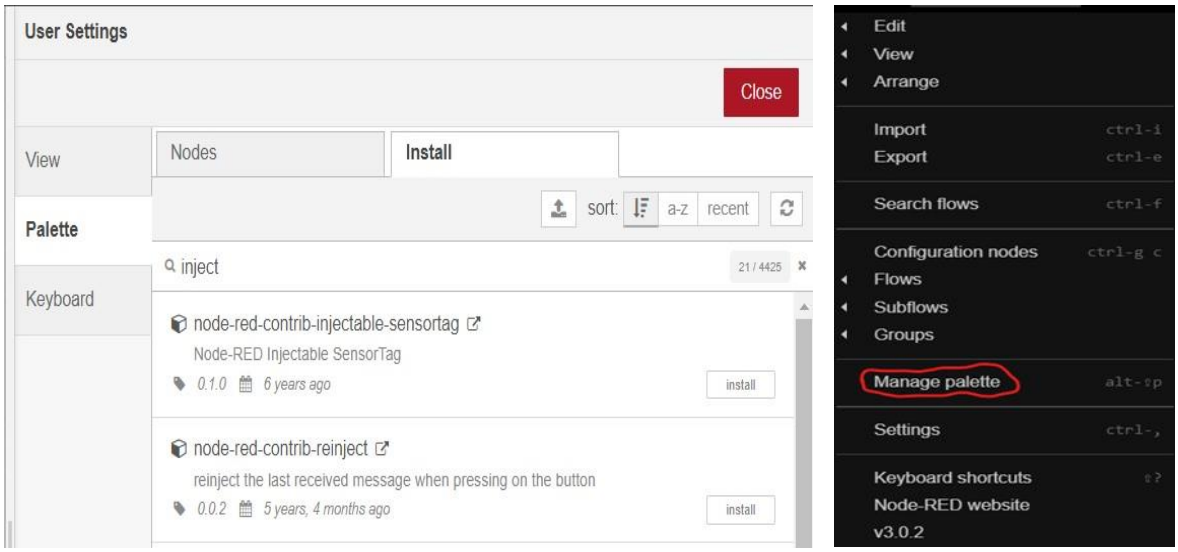


Figure III.15. Installation des nouvelles palettes

III.10.2 Programmation avec Node Red

Le flux de notre système est composé de deux composants principaux. Tout d'abord, il y a un bloc qui traite ces données en utilisant un nœud spécifique pour exécuter un programme développé en Python. Enfin, il y a un bloc qui contrôle les feux de circulation.

III.10.3 Traitement des données

Dans notre système, les données reçues par Node-RED via le port série sont traitées par un contrôleur flou écrit en Python. Les variables d'entrée du programme sont le nombre de véhicules et la durée du feu vert nécessaire pour chaque phase, exprimée en secondes. Pour exécuter le script Python à partir de notre environnement de programmation, nous utilisons le nœud (ou palette) "Python Shell" fourni par Node-RED. Ce nœud est capable d'exécuter un script Python, en prenant les entrées du nœud en tant qu'arguments pour le script, et en envoyant la sortie du script à la sortie du nœud. Il est simplement nécessaire de spécifier le chemin d'accès au dossier de l'environnement virtuel dans la configuration du nœud.

Pour séparer les données provenant de Python Shell, nous utilisons un nœud appelé "JSON" dans notre flux. Ce nœud a la capacité de convertir une chaîne JSON en plusieurs variable. Les données en format JSON sont représentées sous forme de chaînes de caractères. La configuration du nœud est assez simple. La palette "JSON" est représenter sur la figure III.16 suivantes.



Figure III.16. Représentation de la palette "JSON"

Nous utilisons le nœud "Function" pour chaque variable on doit déterminer leur a message dans cette étape. La méthode d'extraction des valeurs est illustrée dans la figure référencée.III.16.

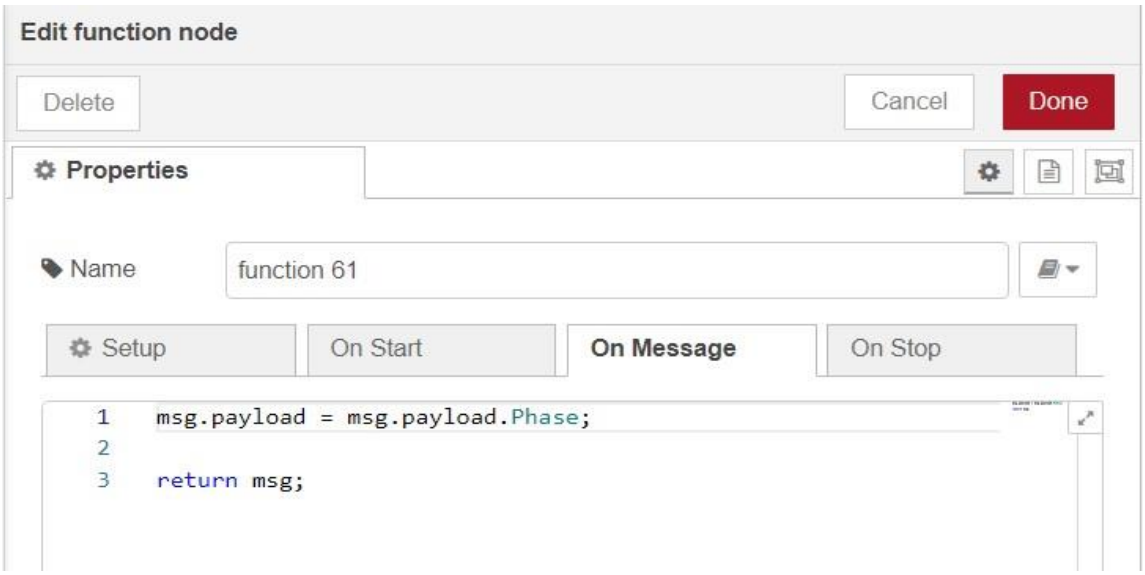


Figure III.17. Extraction d'une variable d'un objet

Une fois que les valeurs ont été extraites, nous pouvons les afficher dans notre interface en utilisant plusieurs nœuds. Pour afficher la phase en utilise le nœud « TEXT »la palette est représenter sur la figure III.18 ci-dessus.



Figure III.18. Représentation de la palette d'affichage de phase

Le flux de données pour une voie est représenté dans la figure III.19. Suivante

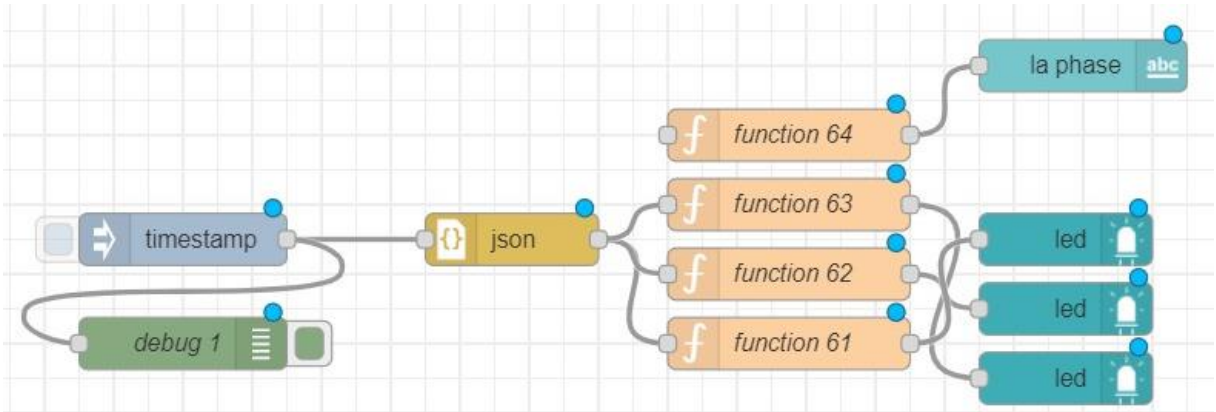


Figure III.19. Flux de des données pour une voie

III.10.4 Contrôle des feux

Pour assurer le contrôle des feux de circulation et gérer les séquences des phases, nous avons développé un flux en utilisant le nœud "Function". Ce nœud appartient à une palette qui permet d'écrire et d'exécuter des fonctions en JavaScript. Tel que la figure III.19 on mettre un nœud de « Function » pour chaque variable.

Le plan des feux est organisé selon le tableau (III.5)

Tableau III.5. Etats des feux dans chaque phase

	Voie (02 et 06)		Voie (04 et 08)		Voie (01 et 05)		Voie (03 et 07)	
	Rouge	Vert	Rouge	Vert	Rouge	Vert	Rouge	Vert
Phase 1	Vrai	Faux	Faux	Vrai	Faux	Vrai	Faux	Vrai
Phase 2	Faux	Vrai	Vrai	Faux	Faux	Vrai	Faux	Vrai
Phase 3	Faux	Vrai	Faux	Vrai	Vrai	Faux	Faux	Vrai
Phase 4	Faux	Vrai	Faux	Vrai	Faux	Vrai	Vrai	Faux

Les fonctions sont exécutées en séquence dans une boucle fermée. Chaque fonction envoie un message en sortie qui contient l'état des feux pour chaque phase, ainsi que les données sont envoyer avec une communication sans fils.

Pour informer l'utilisateur, il est possible d'afficher les durées des différentes phases. Cela peut être réalisé en extrayant les valeurs de temps calculées par le contrôleur logique flou et en les séparant à l'aide d'un nœud JSON. La visualisation de ces valeurs de temps dans l'interface est ensuite effectuée à l'aide d'un nœud « countdown timer ». Il est illustré dans la figure III.20 ci-dessus.

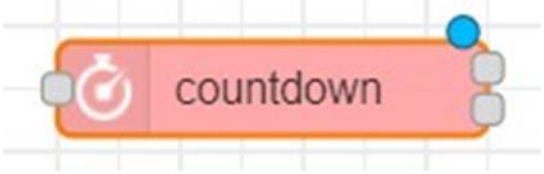


Figure III.20. Représentation de minuterie numérique

Le nœud "Change" permet de définir, modifier, supprimer ou déplacer les propriétés d'un message ou bien contexte du flux ou du contexte global. Ce nœud peut spécifier plusieurs règles qui seront appliquées dans l'ordre où elles sont définies. L'affichage de nœud « change » et des durées est illustré dans la figure III.21 ci-dessus

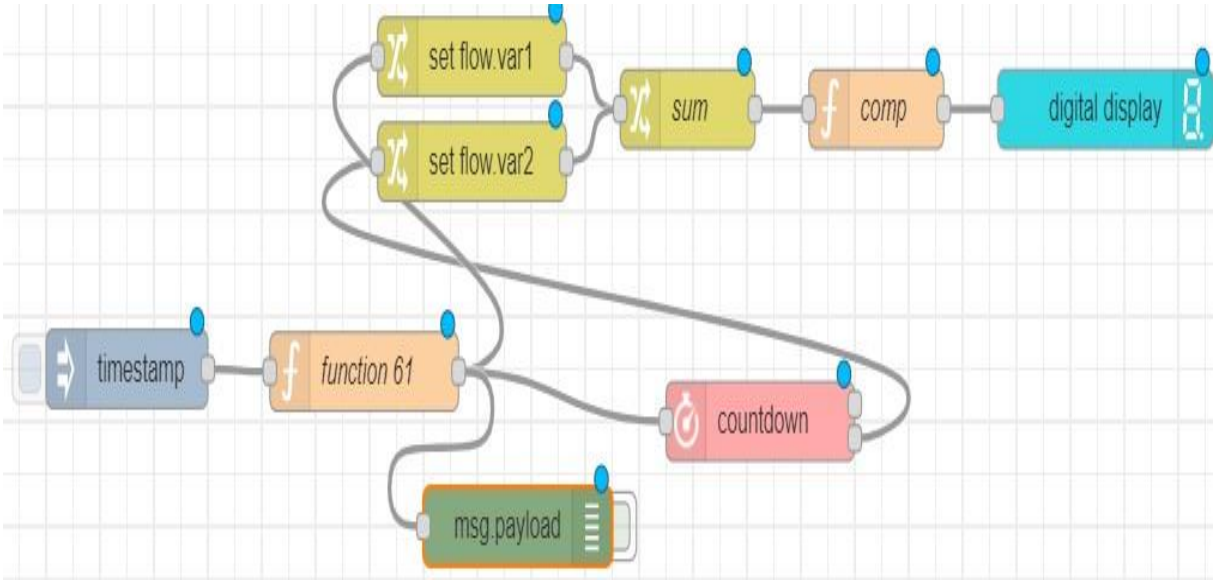


Figure III.21. Flux d'affichage de temp

Pour présenter le fonctionnement des feux de circulation dans notre interface graphique, nous devons afficher l'état de chaque feu qui est envoyé par les fonctions (voir figure III.22). Nous pouvons extraire cet état de chaque feu et l'afficher en utilisant le nœud "Led". Ce nœud

Peut-être configuré pour changer de couleur lors de l'activation et de la désactivation.



Figure III.22. Configuration de nœud LEDs

Le flux de visualisation des LEDs dans l'interface est montré dans figure III.23

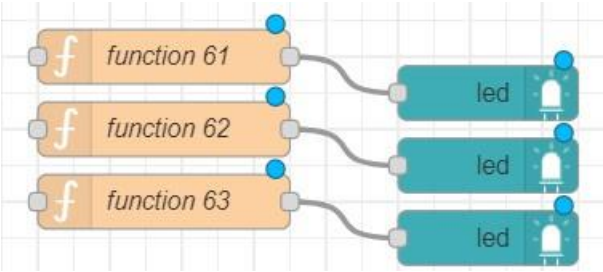


Figure III.23. Flux d'affichage de l'état des LEDs pour chaque phase

Les LEDs sont affichées dans l'interface graphique comme suit (la figure III.24)

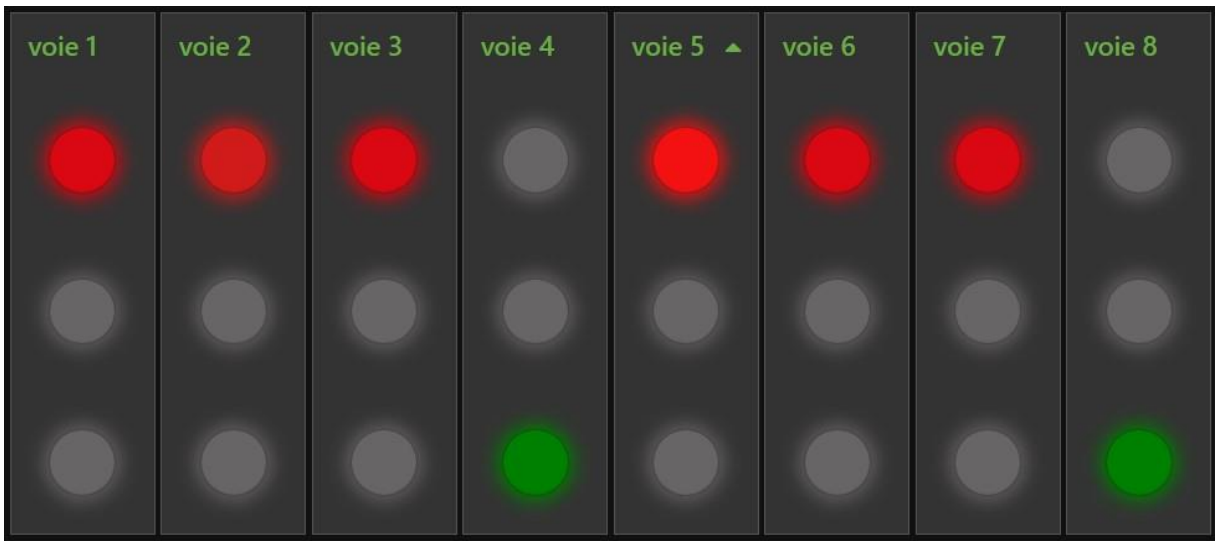


Figure III.24. L'affichage des états des feux dans l'interface graphique

Notre interface graphique est enfin présentée dans la figure III.25



Figure III.25. Interface de supervision

Le programme final de Node RED est présenté dans la figure III.26.

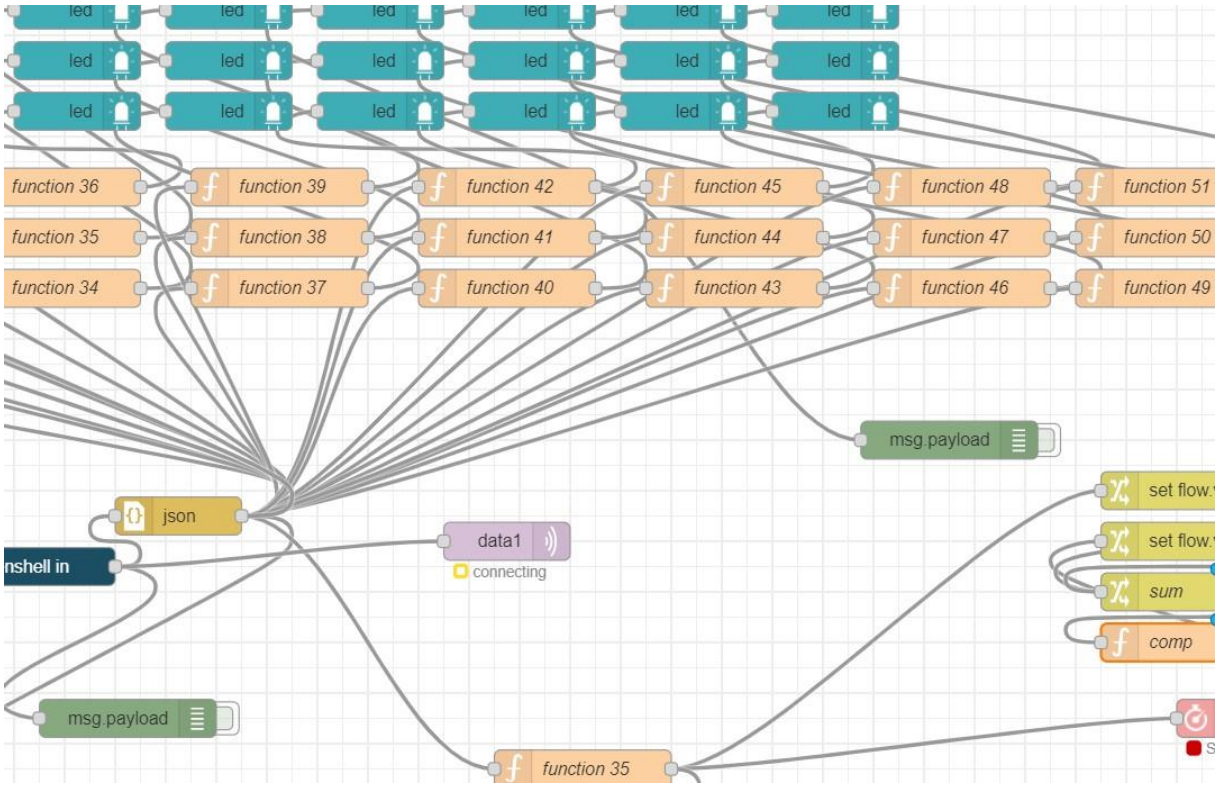


Figure III.26. Programme final de Node RED

III.11 Réalisation

III.11.1 Schéma synoptique global de système

La figure III.28 ci-dessus représente le schéma global de système

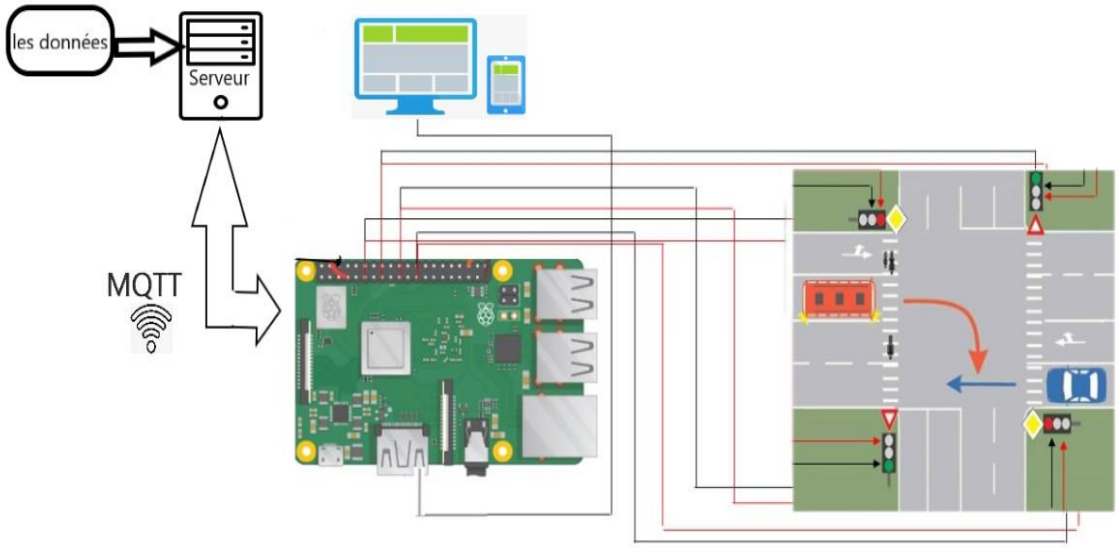


Figure III.27. Schéma synoptique de système

III.11.2 Etapes de réalisation

A) Configuration matérielle : On doit assurer d'avoir un Raspberry Pi avec les broches GPIO disponibles. En doit connecter nos LEDs à une broche GPIO du Raspberry Pi.

B) Installation de Node-RED : On doit assurer d'avoir Node-RED installé sur votre Raspberry Pi. Vous pouvez suivre les instructions officielles de Node-RED pour l'installer correctement.

C) Ajout du module GPIO dans Node-RED : Dans l'interface Node-RED, En doit ajouter un module pour accéder aux broches GPIO. En doit utiliser le module "Node-Red-contrib-gpio" qui fournit les nœuds nécessaires pour interagir avec les broches GPIO.

D) Configuration du nœud GPIO : Dans Node-RED, En doit ajouter un nœud "rpi-gpio out" à nos flux. Double-cliquez dessus pour ouvrir les paramètres. En doit configurer la broche GPIO que nous avons connectée à les LEDs (par exemple, GPIO17) et en doit sélectionner "Output" comme mode de fonctionnement.

E) Programmation en Python : En doit utiliser un nœud "exec" dans Node-RED pour exécuter du code Python. En doit ajoutez un nœud "exec" à nos flux et double-cliquez dessus pour éditer le code.

F) Connexion du nœud GPIO au nœud "exec" : Dans Node-RED, En doit relier la

sortie du nœud GPIO à l'entrée du nœud "exec" pour exécuter le code Python lorsque En doit activez la sortie GPIO.

G) Déploiement du flux : Une fois que nous avons configuré et connecté tous les nœuds, cliquez sur le bouton de déploiement dans Node-RED pour activer votre flux.

Tableaux III.6 présente Connexion des pins de LEDs avec le GPIO

LED	GPIO	Pin
R1	2	3
O1	3	5
V1	4	7
R2	17	11
O2	27	13
V2	22	15
R3	5	29
O3	6	31
V3	13	33
R4	16	36
O4	20	38
V4	21	40

R1 : Lamp rouge de voie 1**, O1:** Lamp orange de voie 1**, V1:** Lamp vert de voie 1.

R2: Lamp rouge de voie 2**, O2:** Lamp orange de voie 2**, V2:** Lamp vert de voie 1

R3 : Lamp rouge de voie 3**, O3:** Lamp orange de voie 3**, V3:** Lamp vert de voie 1

R4 : Lamp rouge de voie 4**, O4:** Lamp orange de voie 4**, V4:** Lamp vert de voie 1

III.11.3 Prototype de réalisation

Dans notre schéma pratique illustré par la figure III.28, Nous présentons notre

dispositif où un Python maître est utilisé pour recevoir les données en provenance d'un Raspberry Pi. Ces données sont ensuite transmises à un Python esclave dans le but de contrôler les feux de circulation.

En commence sur le montage des LEDs est illustré sur la figure III.27

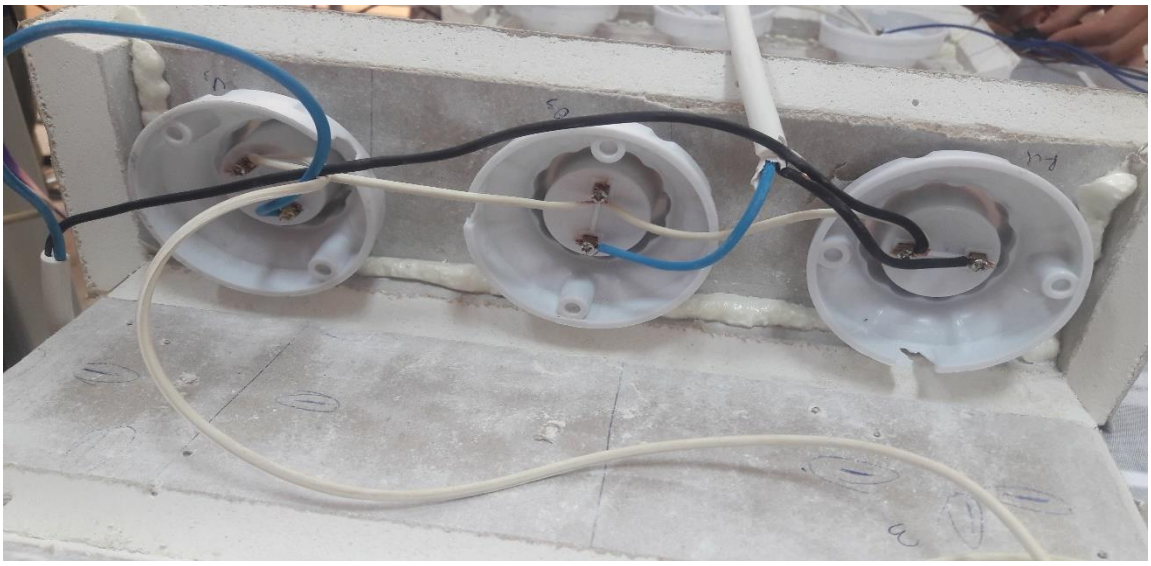


Figure III.27. Montage des LEDs

Le circuit de la réalisation de système est illustré sur la figure III.28 ci-dessus

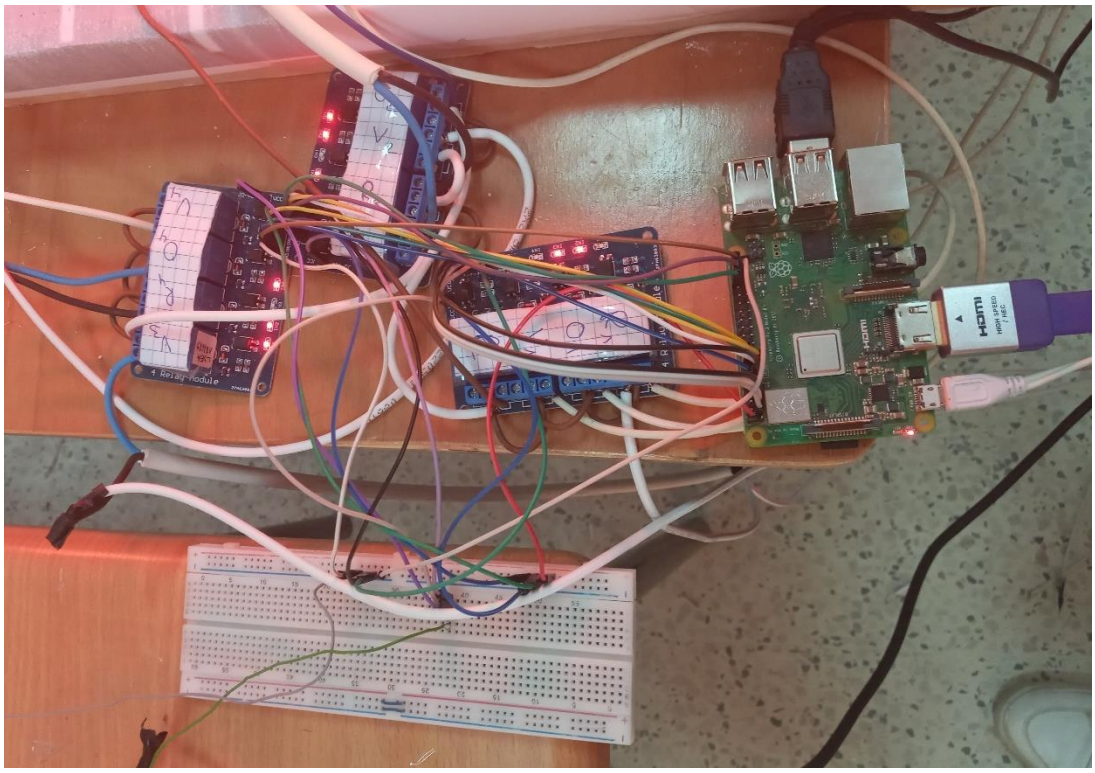


Figure III.28. Représentations de circuit de la réalisation de système

En doit présenter sur la figure ci-dessus le montage total de système

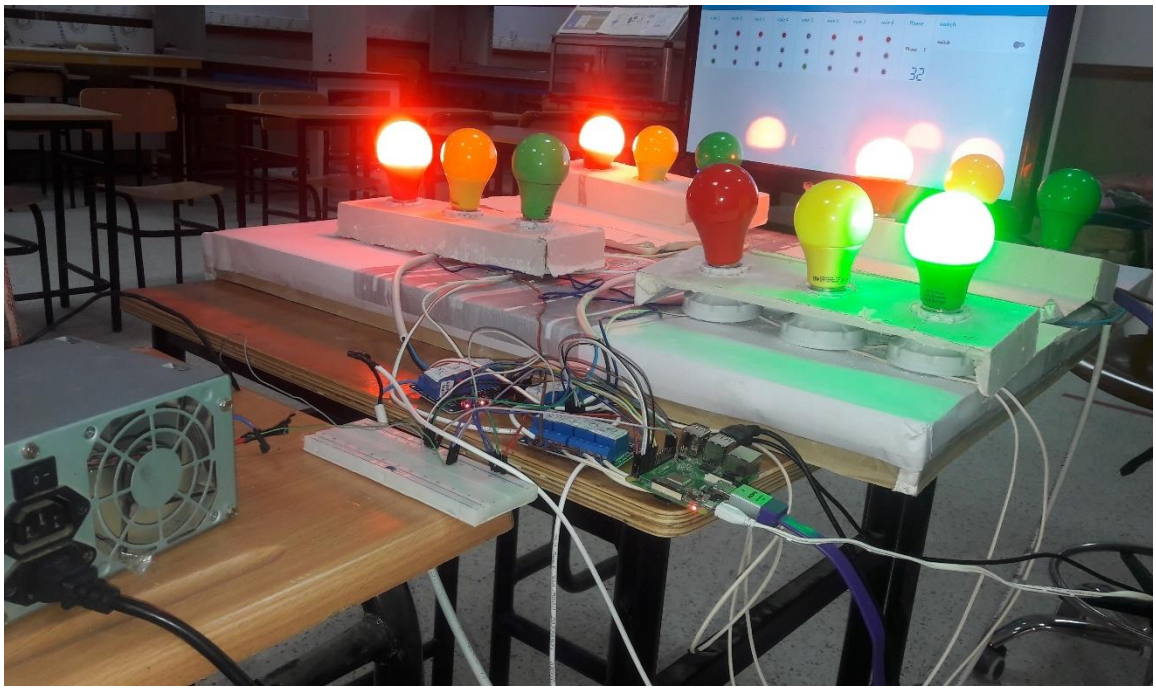


Figure III.29. Montage totale de système

III.12 Conclusion

Dans ce chapitre final, nous avons minutieusement exposé notre réalisation pratique, englobant les solutions matérielles que nous avons proposées. Nous avons mis en place une carte Raspberry pour le traitement des données et l'exécution de la transmission, ainsi qu'un module de communication par radiofréquence pour la réception des commandes. En parallèle, nous avons utilisé un Raspberry Pi pour créer une interface de supervision et de contrôle. De plus, nous avons exploré l'utilisation de l'outil Node-RED afin de concevoir une application IoT. Enfin, nous avons abordé le système de contrôle basé sur la logique floue, offrant ainsi une approche plus flexible et adaptative. Tout au long de ce chapitre, nous avons détaillé les différentes étapes et les technologies impliquées dans notre projet, aboutissant à une solution complète et fonctionnelle pour la gestion des feux de circulation.

Conclusion Générale

Ce mémoire met en évidence l'importance de la gestion efficace du trafic routier dans les zones urbaines, où la congestion routière est un problème majeur. Les feux de circulation jouent un rôle crucial dans la régulation du trafic, mais un contrôle inefficace peut entraîner des embouteillages et des retards importants, impactant la qualité de vie des usagers de la route. Pour relever ce défi, l'utilisation de l'intelligence artificielle (IA) et des systèmes électroniques basés sur l'Internet des objets (IoT) offre des solutions innovantes.

La logique floue se révèle essentielle dans cette approche de contrôle intelligent des feux de circulation. Grâce à des concepts tels que les degrés d'appartenance et les variables linguistiques, la logique floue permet de modéliser et de gérer de manière adaptative les variations de flux de trafic, les heures de pointe et les conditions météorologiques. Cette approche offre ainsi la possibilité d'ajuster dynamiquement les séquences de feux de circulation en fonction de la congestion, du volume de trafic et des priorités des différents flux, conduisant à une réduction significative de la congestion et à une meilleure fluidité du trafic.

Le mémoire se concentre sur le contrôle intelligent d'un feu de circulation d'une intersection isolée, visant à améliorer la fluidité du trafic et à réduire les problèmes de congestion. Pour cela, des technologies telles que MQTT (Message Queuing Telemetry Transport) et Node-RED, combinées à la logique floue programmée en Python, ont été utilisées pour concevoir un système électronique intelligent capable d'optimiser les séquences de feux de circulation en temps réel.

Lors de l'élaboration de ce mémoire, nous avons abordé divers aspects pertinents, notamment le contexte du trafic routier, les approches traditionnelles de contrôle des feux de circulation, les moyens matériels et logiciels utilisés dans le système électronique, ainsi que la méthodologie de mise en œuvre et de test. Les résultats ont démontré l'efficacité du système dans l'optimisation du contrôle des feux de circulation, avec des avantages par rapport aux approches traditionnelles.

En finalité, ce mémoire a apporté des contributions significatives en proposant une approche innovante basée sur l'intelligence artificielle et l'Internet des objets pour améliorer la gestion du trafic routier. Cependant, des perspectives futures restent à explorer, notamment l'implémentation plus large du système dans des environnements réels, ainsi que l'adaptation

du contrôle des feux de circulation à plusieurs intersections connectées et à des situations de trafic variables.

Dans l'ensemble, ce mémoire a ouvert la voie à de nouvelles possibilités de recherche et a souligné l'importance de l'innovation technologique pour résoudre les problèmes de congestion routière et améliorer la qualité de vie des usagers de la route dans les zones urbaines.

Références bibliogrpahiques

[1] Ibrahim Metali, "Contrôle flou des feux d'intersection basé sur l'Internet des Objets", Mémoire de Master de génie en système embarqué, Université de Médéa ,2019.

[2] aidéé carrière, "La gestion en temps réel d'un feu de circulation dans le contexte des systèmes de transports", mémoire de Master, HEC Montréal, Novembre 2005.

[3] F.Zahra Fagroud, Habib Benlahmar, "Vers un Feu Tricolore Intelligent basé sur l'Internet des objets et le traitement d'images", Université d'Aix-Marseille, Mar 2021, Marseille, France Mars, 2023.

[4] M. halbwachs, "Des outils de gestion du trafic et des feux de signalisation" , pp. 1–382, 1999

[5] Halim Deboub et Fairouz Zaoui, "Un contrôle intelligent des feux de circulation aux intersections urbaines", Mémoire de Master en Informatique, Université de Bouira, 2020

[6] Aicha Siba, "conception et réalisation d'une plateforme de gestion d'un réseaux de transport", Mémoire de Master en réseaux et systèmes de télécommunications, Université de Tlemcen, 2012.

[7] les arrêtés, "6 -ème PARTIE : Feux de circulation permanents," Journal officiel du 1er août 1991, Dec. 2018.

[8] Sadjida Bellatrèche, Roumaissa Ahmed Serier, "Contrôle Intelligent des Feux de Circulation d'une Intersection en Utilisant L'algorithme de colonie", Mémoire de Master en Systèmes intelligents et communicants , université Médéa, 2016.

[9] CEREMA, "Sécurité routière : Du nouveau sur la réglementation sur les feux tricolores cerema" , septembre 2021, Disponible au https://www.cerema.fr/fr

[10] Bassem Sammoud, "Contribution à la modélisation et à la commande des feux de signalisation par réseaux de Petri hybrides", Université de Technologie de Belfort-Montbeliard, France, 2010

[12] Pierre Calderan, "Complete guide to create control interfaces for IoT devices. Lots of practical examples for Raspberry Pi, Arduino, Android and more", June 2020.

[11] Hassina Chaib et Somia Taba, Mohamed Tahar Taba, "contrôle de feux de carrefours par arduino XBEE alimente par panneaux solaire", mémoire de master en systèmes des télécommunications et télécommunications, université Guelma, juillet 2019

[13] Hamza Benyezza, Yassine Toumi, "Réalisation d'un contrôleur intelligent des feux de circulation à base de la logique floue implémentée sur un API Siemens S7-300", Mémoire de Master en automatique des systèmes ,Université Médéa ,2016.

[14] Radim Belohlavek, George J. Klir, "Concepts and Fuzzy Logic", The MIT Press, 2011.

[15] Matt Richardson, Shawn Wallace, "Getting Started with Raspberry Pi" , Make Community, LLC, 2012

[16] L. A. Zadeh, "Fuzzy Logic," Computer (Long Beach Calif), vol. 21, no. 4, pp. 83–93, Apr. 1988, doi: 10.1109/2.53.

[17] T. J. Ross, Fuzzy Logic with Engineering Applications. Wiley, 2016.

[18] Hellali Lallouani, "Commande directe du couple basée sur la logique floue type-2 d'une machine asynchrone double étoile", Thèse de doctorat en Génie Électrique, Université de Msila ,2020.

[19] Feth Harrouche, "Application de la logique floue aux machines tournantes", mémoire magister en Mécanique de précision, 2000, Université de Sétif .

[20] E. P. Dadios, Fuzzy Logic - Algorithms, Techniques and Implementations. 2012. doi: 10.5772/2663..

[21] Karim Cherai et Saad Belaribi, "Conception et développement d'un jumeau numérique pour le contrôle et la Supervision des lits d'hôpital intelligents", Mémoire de Master en Informatique, université Médéa, 2020.

[22] Sofiane Bendamardji, "Conception et réalisation d'un système d'IoT pour la gestion des camions de transport des carburants de NAFTAL", Mémoire de Master en Instrumentation, université Médéa , 2022.

[23] Mohamed El Amine Aradj Taha, Laid Kahloul, "Usage du protocole MQTT dans une application de suivi", Mémoire de Master en Informatique, Université biskra, 2017.

[24] P R. Gala, S. Verma, U. Kumar, and H. Ojha, "A Survey of Intelligent Traffic Light Control Systems," Int. J. Comput. Appl., vol. 180, no. 21, pp. 31–36, Feb. 2018.