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 : Système embarqué

Thème

Contrôle flou des feux d'intersection basé sur l'Internet des Objets

Présenté par :

Mr. Ibrahim METALI

Proposé et Dirigé par :

Dr. Mounir BOUHEDDA

Mr. Hamza BENYEZZA

Promotion 2019

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

ملخص

يهدف هذا المشروع إلى تطبيق نظام تحكم ذكي يعتمد على المنطق الضبابي لمفترق الطرق من خلال تطوير واجهة رسومية للإشراف والتحكم من خلال استخدام مفهوم إنترنت الأشياء (IoT). يتم تنفيذ مشروعنا من خلال اللوحات الالكترونية التي تعطي حلول في التواصل ومعالجة البيانات (Raspberry)، وتوصيل أجهزة الاستشعار أو المشغلات ومعالجة البيانات (Arduino) والاتصالات (Module nRF24L01)، يتم تطوير حلول البرمجيات في بيئة برمجة مرئية "Node Red" لتطبيق IoT الخاص بنا باستخدام لغات البرمجة Python وJavaScript و C.

الكلمات المفتاحية: الاتصال، التحكم، الذكاء، إنترنت الأشياء (IoT)، مفترق الطرق، المنطق الضبابي، Node Red.

Abstract

This project aims to implement an intelligent control system based on the fuzzy logic of an intersection by developing a graphical interface of supervision and control through the concept of the Internet of Things (IoT). The implementation of our project is done through hardware solutions for interfacing and processing (Raspberry), connection of sensors or actuators and data processing (Arduino) and communication (Module nRF24L01), the software solutions are developed in a visual programming environment "Node Red" for our IoT application using Python, JavaScript and C programming languages.

Keywords: Communication, Control, Intelligent, Internet of Things (IoT), Intersection, Fuzzy Logic, Node-Red.

Résumé

Ce projet a pour objectif de mettre en œuvre un système de contrôle intelligent basé sur la logique floue d'une intersection en développant une interface graphique de supervision et contrôle par l'utilisation du concept de l'internet des objets (IoT). La mise en place de notre projet se fait grâce à des solutions matérielles pour l'interfaçage et traitement (Raspberry), connexion des capteurs ou actionneurs et traitement des données (Arduino) et la communication (Module nRF24L01), des solutions logicielles sont développés dans un environnement de programmation visuelle « Node Red » pour notre application IoT en utilisant les langages de programmation Python, JavaScript et C.

Mots clé : Communication, Contrôle, Intelligent, Internet des objets (IoT), Intersection, logique floue, Node-Red.

Remerciements

« Celui qui ne remercie pas les gens ne remercie guère Allah »

Toute la gratitude à Allah, je remercie Allah le Tout-Puissant, le Miséricordieux, de m'avoir appris ce que j'ignorais, de m'avoir donné la santé, la patience et tout ce qui m'était nécessaire pour l'achèvement de cette mémoire.

L'achèvement de cette mémoire ne serait pas possible sans le soutien, la patience et l'encouragement de plusieurs personnes bien particulières. Je voudrais saisir cette occasion pour exprimer ma gratitude à ceux qui m'ont aidé de mille et une façons.

Je tiens à remercier Monsieur BOUHEDDA Mounir, mon encadreur, Docteur à l'université de Yahia Fares Médéa, pour m'avoir proposé ce sujet, pour sa gentillesse, sa disponibilité tout le temps, ses explications et suggestions et surtout pour son aide précieuse tout au long de la réalisation de ce travail.

Je tiens à exprimer mes remerciements à Monsieur Benyezza Hamza doctorant à l'université de Yahia Fares Médéa, pour son aide, sa disponibilit, tout le temps.

Je remercie également les membres de jury qui ont acceptés d'honorés par leur présence ma soutenance afin de juger nos travail.

Mes remerciements les plus profond et chaleureux avec tout mon amour et toute ma gratitude vont à vous mes chers parents, je vous remercie de m'avoir encouragée, guidée avec vos conseils et surtout votre patience tout au longue de ce travail et aussi pour avoir fait de moi ce que je suis aujourd'hui. Merci maman, merci mon père, c'est vraiment fort peu devant ce que vous m'avez donné durant ce parcours. Je tiens à vous présenter toutes mes excuses et particulièrement si ce travail m'avait quelque peu empêchée de veiller sur vous, dans un temps de forte pression et de stress.

Je dois une mention particulière d'une profonde gratitude à ma famille qui m'a toujours soutenue, en particulier, mes sœurs, mes frères.

Enfin je n'oublie guère mes amis et tous ceux qui m'ont aidé de loin ou de près, MERCI à toutes et à tous.

Dédicaces

A mon très cher père

Mr. Metali Abdelfattah

Tu as rempli ton devoir envers tes enfants, tu nous as mis dans le droit chemin. Tu nous as appris la simplicité, la politesse, le respect des autres et l'honnêteté. Nous sommes fiers de toi.

A ma très chère mère

Ce travail ***est*** *le fruit de tes efforts, des longues années de sacrifices aux quels tu as consenti.*

Je ne trouverais jamais assez de mots pour t'exprimer toute ma gratitude et mon affection.

A mes chères sœurs

A mes chers frères

En témoignage de mon affection fraternelle et profonde estime.

Je vous souhaite beaucoup de bonheur et de réussite.

Restons unis et solidaires.

A toute ma famille

En témoignage de ma gratitude et l'expression de mon affection la plus sincère, je vous dédie ce travail.

A Mes Chers Amis

Bellarbi Abdel Rahim et Kadri Mohamed

Gherabli Rafik et Dechicha Farid

Khaldi Yacine et Cheriet Mohamed

A tous mes amis de "Systèmes Embarqués"

A tous mes amis d'Instrumentation"

A tous mes amis de "Réseaux Électriques"

Table de matière

Introduction générale 2

Chapitre I : Généralités sur les systèmes de transport intelligents

I.1.Introduction 3
I.2.Système de transport intelligent (STI) 3
I.3.Fonctionnement des STI 5
I.3.1. Fonctions des composantes des STI 5
I.3.2.Facteur humain 6
I.4.Applications des STI 6
I.5.Avantages de STI 11
I.6.STI dans les pays développés 11
I.6.1. Singapour 11
I.6.2. New York (États Unis d'Amérique) 13
I.6.3.Londres (Royaume-Uni) 15
I.7.Supervision dans les STI 16
I.8.Gestion des feux de circulation 17
I.8.1. Fonctionnement d'une intersection 17
I.8.2. Contrôleur des feux de circulation 18
I.8.3. Contrôle des feux de circulation 18
I.8.4. Systèmes de régulation du trafic 19
I.9.Conclusion 20

Chapitre II : Solutions hardwares et softwares

II.1.Introduction 21
II.2.Notion sur l'internet des objets (IoT) 21
II.2.1.Internet des objets 21
II.2.2.Qu'est-ce qu'un Objets ? 21
II.2.4.Applications de l'Internet des Objets 22
II.2.5.Technologies de l'internet des objets 26
II.2.6.Architecture de l'internet des objets 29
II.3.Outil de développement (Node Red) 30
II.3.1.C'est quoi le Node Red ? 30
II.3.2.Editeur Node Red 30

Table de matière

II.4.Raspberry 35
II.4.1.Fiche technique de Raspberry 35
II.5.Arduino 36
II.5.1.Caracteristiques principales 37
II.5.2.Shield Arduino 38
II.6.Logique floue 39
II.6.1.Théorie de la logique floue 39
II.6.2.Fonctions d'appartenance 39
II.6.3.Opérateur de la logique floue 40
II.6.4.Structure d'une commande par logique floue 41
II.7.Conclusion 45

Chapitre III : Réalisation pratique

III.1.Introduction 46
III.2.Cahier de charge 46
III.2.1.Collecte de données 46
III.2.2.Traitement et affichage des données 46
III.2.3.Action 46
III.3.Solutions hardware et software 47
III.4.Communication entre les solutions matérielles 47
III.5. Contrôleur flou 48
III.5.1.Fonction d'appartenance 48
III.5.2.Variables linguistiques 50
III.5.3.Règles floues 51
III.5.4.Algorithme du contrôleur flou 51
III.6. Communication avec le Module nRF24L01 53
III.6.1. Présentation 53
III.6.2.Liaison spi 54
III.6.3.Pins de module nRF24L01 54
III.6.4.Principe de fonctionnement 55
III.7.Communication 58
II.7.1.Transmission avec nRF24L01 58

Table de matière

III.7.2.Réception avec nRF24L01 59
III.7.3. Communication série 59
III.8. Fonctionnement du système global 60
III.9. Interface de supervision 61
III.10.Demarrage de Node Red sur Raspberry 61
III.11.Programmation avec Node Red 62
III.11.1. Réception des données 63
III.11.3. Traitement des données 64
III.11.2.Visualisation des données 66
III.11.4. Contrôle des feux de circulation 67
III.11.5.Afficher l'état des feux 70
III.12.Supervision avec application de l'IoT 72
II.13. Prototype réalisé 73
III.14. Conclusion 74
Conclusion générale 752
Référence bibliographie 76

Liste des abréviations

BLE : Bluetooth Low Energy.

CIGT : Centre d'Ingénierie et de Gestion du Trafic.

CRC : Contrôle de Redondance Cyclique.

GPIO: General Purpose Input Output.

GPS: Global Positioning System.

GSM: Global System for Mobile.

IDE: Integrated Development Environment.

IdO: Internet des Objets.

IHM: Interface Homme-Machine.

IoT: Internet of Things.

JSON: JavaScript Object Notation.

RFID: Radio Frequency Identification.

SCATS: Sydney Coordinated Adaptive Traffic System.

SCOOT: Split Cycle Offset Optimization Technique.

SMS: Short Message System.

SPI : Serial Peripheral Interface.

STI : Système de Transport Intelligent.

WIMAX: Worldwide Interoperability for Microwave Access.

WSN : Wireless Sensor Network.

Liste des figures

Figure I.1 : centres d'ingénierie et de gestion du trafic 6

Figure I.2 : Modèle d'intersection routière 7

Figure I.3 : Application de limites de vitesse variables 8

Figure I.4 : Caméra de surveillance 9

Figure I.5 : Problème de congestion aux autoroutes 9

Figure I.6 : Exemple d'un panneau de messages variable 10

Figure I.7 : Camera J Eyes 12

Figure I.8 : Feu de circulation tricolore 13

Figure I.9 : panneaux de signalisation électroniques 13

Figure I.10 : Route dans la ville de New York 14

Figure I.11 : Modèle de camera de feu rouge 15

Figure I.12 : Heathrow pod 16

Figure I.13 : Modèle d'intersection généralement utilisé 17

Figure I.14 : Diagramme des phases 17

Figure II.1 : Quelques exemples d'objets connectés. 22

Figure II.2 : Application de l'IoT dans le transport 23

Figure II.3 : Application de l'IoT au smart cities 23

Figure II.4 : Exemple d'un Robot connecté [https://www.fanuc.eu/uk/en] 24

Figure II.5 : Utilisation de IoT pour superviser et contrôler la serre 25

Figure II.6 : Internet des objets dans la domotique 26

Figure II.7 : Technologies de communication sans fils et leurs débits 27

Figure II.8 : Modèle architecture de l'internet des objets 29

Figure II.8 : Editeur « Node Red » 31

Figure II.9 : Comment construire un flux (flow) 31

Figure II.10 : Nœuds d'entrée utilisées dans Node Red 32

Figure II.11 : Nœuds de traitement utilisées dans Node Red 32

Figure II.12 : Nœuds De sortie utilisées dans Node Red 33

Liste des figures

Figure II.13 : Palette utilisée dans le tableau de bord (Dashboard) 34

Figure II.14 : Eléments de tableau de bord (Dashboard) 34

Figure II.15 : Carte Raspberry Pi 3 B+ 35

Figure II.16 : Fonctions des broches de GPIO Rpi 3 B+ 36

Figure II.17 : Carte Arduino Uno 37

Figure II.18 : Composants principale de l'Arduino 38

Figure II.19 : Shields utiliser avec Arduino 38

Figure II.21 : Formes des fonctions d'appartenance 40

Tableau II.1 : Opérateurs de Zadeh et de probabilités 40

Figure II.22 : Opérateurs ET, OU 41

Figure II.23 : Opérateur NON 41

Figure II.27 : Structure d'un régulateur logique floue 42

Figure II.28 : Degré d'appartenance d'un variable dans chaque catégories 43

Figure II.29 : Défuzzification par centre de gravité 44

Figure II.30 : Défuzzification par valeur maximum 45

Figure III.1 : Communication entre les composants électroniques. 48

Figure III.2 : Structure du contrôleur par la logique floue 48

Figure III.3 : Présentation de nombre de véhicule 49

Figure III.4 : Présentation de débit 49

Figure III.5 : Présentation de variable de sortie (Temps) 49

Figure III.6 : Organigramme de l'algorithme du contrôleur flou 52

Figure III.7 : modules nRF24L01 54

Figure III.8 : Communication maitre/esclave 54

Figure III.9 : Pins de module nRF24L01 et leurs fonctions 54

Figure III.10 : Fréquence du canal 56

Figure III.11 : Structure des paquets ShockBurst améliorée 56

Figure III.12 : Transmission des données par Arduino 59

Liste des figures

Figure III.13 : Réception des données par l'Arduino 59

Figure III.14 : Mise en place des matériels du système. 60

Figure III.15 : Environnement de programmation Node Red 61

Figure III.16 : Installation de gestionnaire des palettes 62

Figure III.17 : Installation des nouvelles palettes 62

Figure III.18 : Flux de système de supervision et contrôle 63

Figure III.19 : Palette Serial 63

Figure III.20 : Configuration du nœud Serial 64

Figure III.21 : Flux de réception et affichage des données 64

Figure III.22 : Configuration du nœud JSON 65

Figure III.23 : Différence entre la forme JSON et la forme objet JavaScript 65

Figure III.24 : Extraction d'un variable d'un objet 66

Figure III.25 : Configuration de jauge 66

Figure III.26 : Flux de réception des données 67

Figure III.27 : Visualisation des données dans l'interface graphique 67

Figure III.28 : Flux de contrôle les feux de circulation 68

Figure III.29 : Etats des feux dans la phase 01 69

Figure III.30 : Fonction en JS pour afficher les images et temporiser les feux 69

Figure III.31 : Visualisation des phases 70

Figure III.32 : Configuration de nœud Led 70

Figure III.33 : Flux d'affichage de l'état des Leds pour chaque phase 70

Figure III.34 : Fonction JS pour obtenir l'état des leds 71

Figure III.35 : Visualisation des états des feux dans l'interface graphique 71

Figure III.36 : Interface de supervision 71

Figure III.37 : Nœud FRED 72

Figure III.38 : Exemple d'envoi des données au FRED 72

Figure III.39 : Interface graphique avec l'outil FRED 73

Liste des figures

Figure III.40 : Prototype réalisé.....73

Figure III.41 : maquette du feu rouge.....74

Liste des tableaux

Tableau II.1 : Opérateurs de Zadeh et de probabilités**.....40**

Tableau III.1 : Règles floues **.....51**

Tableau III.2 : Etats des feux dans chaque phase**.....688**

Introduction générale

Introduction générale

Les progrès continus des technologies de la microélectronique, système embarqué et des réseaux de capteurs permettent maintenant d'envisager le déploiement de services sécurisés et optimisés distribués sur des réseaux d'objets communicants intelligents interconnectés : c'est la vision de l'Internet des Objets. Nous assistons actuellement au déploiement d'une nouvelle génération d'objets interconnectés dotés de capacités de communication de détection et d'activation (réseaux de transport d'information sans fil, RFID, WSN, etc.) pour de nombreuses applications. Ainsi l'interconnexion d'objets dotés de capacités avancées de traitement va conduire à une révolution en termes de création et de disponibilité de service et va profondément changer notre façon d'agir sur notre environnement. En bref le monde physique fusionne progressivement avec le monde virtuel[1].

Les futures villes intelligentes seront confrontées à une multitude de défis liés au logement, à la sécurité, à l'emploi et à la fourniture de services publics, y compris la mobilité urbaine. La mobilité urbaine devrait s'aggraver au cours du 21e siècle dans les pays en développement où le nombre de propriétaires de voitures est en augmentation. Cela entraînera des embouteillages supplémentaires du trafic, susceptibles d'entraîner des coûts supplémentaires pour les automobilistes, ainsi qu'une augmentation de la pollution atmosphérique et sonore, du stress et des pertes en accidents de la route[2]. Donc il est nécessaire de mettre en place une nouvelle infrastructure capable de résoudre ces problèmes également appelés systèmes de transport intelligents. Il est intelligent grâce à l'intégration des nouvelles technologies de détection et transmission, de surveillance, de stockage et de traitement intelligent qui permettent l'exploitation rapide et continue des données qui donnent des meilleurs résultats en matière de gestion des trafics et la sécurité et surtout la réduction de pollution.

Notre objectif consiste à proposer un système de contrôle intelligent des feux de circulation à base de l'internet des objets, ce système est constitué d'une interface de supervision et contrôle intelligent des feux à base de la logique floue.

Des microcontrôleurs de type Arduino sont utilisés pour la collecte des données et l'envoie des actions. Une carte Raspberry est utilisée pour l'exécution de l'algorithme de contrôle intelligent ainsi que l'interface Homme-Machine (IHM) réalisé sous la plateforme Node-Red d'IBM. La communication entre les différentes parties Hardwares est effectuée par des modules de communication radiofréquence ainsi que le port série USB. Le développement des différents programmes a nécessité l'utilisation de trois langages qui sont : Python, langage C et JavaScript.

Le mémoire réalisé dans ce cadre est réparti sur trois chapitres. Le premier chapitre présente des généralités sur le système de transport intelligent. Dans le deuxième chapitre, nous parlerons sur

les notions fondamentales des solutions théoriques est techniques à utiliser dans notre projet. Le troisième chapitre est consacré à la description de notre système développé en abordant les deux volets : l'implémentation hardware et le développement des programmes.

Et enfin, Nous terminerons notre mémoire avec une conclusion générale et des perspectives.

Chapitre I :

Généralités sur les systèmes de transport intelligents

I.1.Introduction

Dans le domaine de la mobilité, la technologie apporte progressivement et rapidement dans les années dernières, une "intelligence" qui améliore la sécurité, la sûreté et l'exploitation de l'information. On parle alors de systèmes de transport intelligents. Il consiste en l'application des nouvelles technologies au domaine des transports (mémoire, communication, traitement de l'information, capacités sensorielle). On trouve les STI dans plusieurs champs d'activité : optimisation de l'utilisation des infrastructures de transport, amélioration de la sécurité et de la sûreté, développement des services ainsi que développement durable en favorisant le report de la route vers les autres modes plus respectueux de l'environnement.

Nous entamerons ce chapitre par la définition de quelques notions relatives aux domaines des systèmes de transport intelligents.

I.2.Système de transport intelligent (STI)

Un système de transport intelligent (STI) est un système interactif de collecte, de traitement et de diffusion d'information appliqué aux transports, basé sur l'intégration des technologies de l'information et de la communication aux infrastructures et aux véhicules utilisés, de manière à améliorer la gestion et l'exploitation des réseaux de transport et des services aux utilisateurs qui y sont associés. Tous les modes de transport (routier, aérien…etc.) sont visés par ces applications pour la sécurité ou la régulation des flux de la circulation pour l'information des usagers des transports.

Les technologies utilisées dans les systèmes de transport intelligents varient, allant de systèmes de gestion basiques comme les systèmes de gestion des carrefours à feux, les panneaux à messages variables, les radars automatiques ou la vidéo surveillance aux applications plus avancées qui intègrent des données en temps-réel avec retour d'informations de nombreuses sources, comme les informations météorologiques, les systèmes de navigation embarqués informant des temps de parcours en temps réel. De plus, les techniques prédictives sont développées pour permettre une modélisation avancée et une comparaison avec une base regroupant des données historiques de référence[3].

Quelques technologies typiquement implantées dans les STI sont décrites comme suite :

- **Communication sans fil**

Diverses technologies de communication sans fil sont proposées pour les systèmes de transport intelligent pour collecter et transférer des données. Des communications à courte portée (protocole IEEE 802.11) et des communications à plus longue portée (WIMAX, GSM, etc…)[3].

- **Technologie de localisation**

La localisation des voyageurs, des marchandises et des véhicules sont des informations cruciales pour permettre le développement d'une mobilité plus intelligente, mieux optimisée et plus respectueuse de l'environnement.

Les services et technologies permettant de connaître et d'exploiter ces informations se développent très rapidement, on parle par exemple sur le GPS, la téléphonie mobile[3].

- **Les technologies de capteurs**

L'électronique est indispensable pour contrôler, gérer, améliorer, voire innover, les systèmes embarqués d'un véhicule. Ces systèmes ont recours à des capteurs pour renseigner l'intelligence des systèmes sur l'environnement, ces capteurs recueillent les données aussi bien en temps réel qu'en temps différé le plus précis et le plus accessible possible[3].

Les capteurs utilisés dans le système de transport intelligent sont : les capteurs sans fil, la boucle électromagnétique et les capteurs vidéo...etc.

- **Analyse de données**

L'analyse de données consiste à examiner des ensembles de données afin de tirer des conclusions sur les informations qu'ils contiennent, de plus en plus à l'aide de systèmes et de logiciels spécialisés[4]. Les technologies et techniques d'analyse de données sont largement utilisées dans les STI pour permettre aux organisations de prendre des décisions plus éclairées [5], optimiser les horaires de transport en prévoyant l'impact de l'entretien, des travaux routiers, de la congestion et des accidents et fournir des informations en temps réel au voyageurs.

I.3.Fonctionnement des STI

Les STI fonctionnent au moyen de technologies d'information et de contrôle qui représentent le noyau technique des STI et qui permettent d'en assurer les principales fonctions, néanmoins les facteurs humains sont également d'une importance vitale, et peuvent être très complexes[6].

I.3.1. Fonctions des composantes des STI

Les STI sont le fruit de la révolution des technologies d'information et de la communication qui marque l'ère numérique. Ils comprennent un vaste éventail de fonctions de soutien aux utilisateurs, allant de simples alertes d'information jusqu'aux systèmes de contrôle hautement perfectionnés[6].

- **Collecte de données**

  La planification stratégique nécessite une collecte de données précise, complète et rapide avec une observation en temps réel. Les données sont collectées via divers périphériques matériels qui constituent la base d'autres fonctions STI. Ces dispositifs sont des identifiants automatiques de véhicule, des localisateurs automatiques de véhicule basés sur GPS, des capteurs, une caméra, etc. Le matériel enregistre principalement les données telles que le trafic, la surveillance, la vitesse et le temps de déplacement, la localisation, le poids du véhicule, les retards, etc. les serveurs généralement situés dans un centre de collecte de données qui stocke de grandes quantités de données pour une analyse ultérieure[7]

- **Utilisation de l'information**

  Au niveau infrastructurel ça représente la détection des incidents, la gestion de la demande et la surveillance de la congestion[6].

- **Communication**

  La communication rapide et en temps réel des informations étant la clé de la maîtrise de la mise en œuvre des STI, cet aspect consiste à transmettre les données collectées du terrain au CIGT, les données collectées sont utilisées pour prendre des décisions : envoyer des informations aux voyageurs, signaler une alerte ou commander les dispositifs des STI (ouvrir ou fermer les ponts par exemple). Les autres méthodes de communication sont les communications dédiées à courte portée (Wifi par exemple) et les interfaces longue portée et à moyenne portée (Wi MAX, GSM, 4G)[7].

- **Informations aux voyageurs**

  Les systèmes de conseil de voyage sont utilisés pour informer les usagers en déplacement des mises à jour de transport. Le système fournit des informations en temps réel telles que le temps de trajet, la vitesse de déplacement, les retards, les accidents de la route, les changements de route, les déviations, les conditions de la zone de travail, etc. Ces informations sont fournies par un large éventail de dispositifs électroniques tels que les panneaux à messages variables, la radio de signalisation routière, internet, SMS, cellule automatisée[7].

I.3.2.Facteur humain

Pour que les STI fonctionnent correctement, l'aspect humain doit être pris en compte, en attachant plus d'importance à l'interface entre l'homme et la machine (IHM). Les êtres humains font partie des STI, non seulement en tant que conducteurs, mais aussi en tant que planificateurs, voyageurs et autres, en utilisant divers modes de transport[6].

I.4.Applications des STI

- **Centre d'ingénierie et de gestion du trafic (CIGT)**

  Les réseaux routiers urbains sont gérés par des centres d'ingénierie et de gestion du trafic (CIGT) (la figure I.1). Le rôle de tels organismes, généralement attachés à une zone géographique bien particulière, est de coordonner au mieux les différents éléments routiers, et de faire face aux situations de la vie de tous les jours (travaux, accidents, gestion du trafic routier et des pics …etc.)[8].



Figure I.1 : centres d'ingénierie et de gestion du trafic

Les missions des CIGT sont multiples :

- Recueillir les données en provenance directe des routes afin de remplir un rôle de superviseur et d'agir en cas de problème.
- Gérer le trafic en cas d'imprévus et Informer les usagers et assurer le suivi des événements.
- Assurer du maintien et du bon fonctionnement de l'ensemble des équipements du réseau[3].

❖ **Intersections**

Le champ d'application des STI en milieu urbain est très large : en premier lieu, ces derniers agissent sur les intersections (la figure I.2), en se chargeant d'appliquer une stratégie de changement des feux de circulation. Cette gestion des feux va représenter un aspect essentiel de la fluidité du trafic routier dans une ville[8]. Les travaux insistent également sur la gestion des voies spéciales, où les transports en commun, taxi et pistes cyclables peuvent jouer un rôle important.



Figure I.2 : Modèle d'intersection routière

❖ **Ronds-points**

Certains travaux abordent la gestion des ronds-points, en se basant sur les priorités existantes. Dans de nombreux pays, il est, par exemple, commun de donner la priorité aux usagers étant déjà à l'intérieur d'un rond-point. Ce qui en ressort apparaît comme une évidence : soit deux flots, chacun étant sur une voie respectivement au cœur du rond-point et sur les côtés, la priorité doit être donnée au flot de la plus grande importance afin

d'obtenir une gestion optimale du trafic. Les feux quant à eux améliorent grandement la fluidité du trafic, mais uniquement en cas de débit suffisamment élevé[8].

❖ **Limites de vitesse variables**

Récemment, certaines juridictions ont commencé à expérimenter des limitations de vitesse variables (La figure I.3) en fonction de la congestion routière et d'autres facteurs. En règle générale, ces limites de vitesse ne changent que dans des conditions défavorables, au lieu de s'améliorer dans de bonnes conditions. Les premiers résultats ont montré une fluidité de la circulation et une diminution du nombre d'accidents [7].



Figure I.3 : Application de limites de vitesse variables

❖ **Sécurité et sûreté**

La sécurité routière est un thème transversal dans le domaine des transports, elle concerne l'ensemble des connaissances, dispositifs et mesures visant à éviter les accidents de la route ou à en atténuer les conséquences. Les STI peuvent, de ce fait, contribuer directement ou indirectement à la sécurité routière[3]. Ainsi se développent les systèmes de surveillance et d'alerte qui permettent d'alerter les usagers d'une difficulté de conduite[3]. La figure (I.4) présente une caméra de surveillance utilisée dans les routes.



Figure I.4 : Caméra de surveillance

❖ **Gestion de congestion**

La congestion constitue un problème majeur pour tous les réseaux de transport (la figure I.5), l'augmentation de l'éfficacité des systèmes de transport existants est un des principaux objectifs des programmes de STI partout dans le monde. Il est possible de réduire la congestion en dotant les réseaux d'instruments qui permettront d'améliorer leur exploitation en temps réel comme les capteurs qui permettront alors de contrôler ces systèmes[6].



Figure I.5 : Problème de congestion aux autoroutes

❖ **Information des voyageurs**

Dans le domaine de l'information des voyageurs, un service peut être qualifié d'ambiant, s'il renseigne le voyageur en fonction de son environnement direct. Il s'agit donc de

services capables d'adapter l'information en fonction des spécificités de l'utilisateur, de sa localisation et selon ses besoins afin de l'aider à réaliser son voyage dans les meilleures conditions possibles[6]. La figure (I.6) montre l'utilisation des panneaux de messages variables pour informer les voyageurs.



Figure I.6 : Exemple d'un panneau de messages variable

- **Protection de l'environnement**

Les systèmes de transport intelligents ont été identifiés par la communauté des spécialistes comme étant des outils susceptibles de réduire la consommation d'énergie, la pollution et les émissions de gaz à effet de serre dus aux déplacements des personnes et des marchandises[3].

- **Facteurs de confort**

Les usagers de tout système de transport doivent se sentir à l'aise, confortable et en confiance. Le contrôle des vitesses, la régulation des accès, les alertes préalables d'incidents et les instructions sur les itinéraires de substitution peuvent faciliter les déplacements routiers et les rendre moins stressants. Les installations comme les systèmes multimédias qui fournissent à la fois des services de divertissement et de navigation peuvent en faire autant. Les usagers des transports en commun s'attendent aussi à des standards élevés de confort, de commodité et de services[6].

I.5.Avantages de STI

- **Rendre plus confortable l'expérience du voyageur**

  Avec la mobilité intelligente l'expérience de voyage est considérablement simplifiée. Elle est également plus confortable et plus agréable pour les usagers.

  Cela passe notamment par les supports numériques et mobiles, qui servent de moyen de paiement ou de terminal d'information. Ces supports permettent de délivrer des informations aux voyageurs presque en temps réel sur tous les aspects de leur trajet. Ils peuvent faire un choix informé sur le mode de transport à emprunter, selon le coût du voyage, la durée prévue, les risques de retards, l'état du trafic ou l'affluence[9].

- **Réduire les risques liés à la mobilité**

  Les systèmes de communication embarqués permettent de rapporter les accidents et incidents très rapidement et donnent la possibilité aux autorités de réagir rapidement. Les aides à la conduite, qui sont déjà de plus en plus présentes, permettent à la fois de faciliter la conduite et d'éviter la perte d'attention du conducteur. Enfin, les systèmes de dissuasion tels que les radars et les caméras de surveillance avec traitement automatisé des images permettent de réduire les infractions routières et donc l'insécurité[9].

- **Avantages sur le plan de l'environnement**

  Le transport est une source importante d'émissions de gaz à effet de serre, et les STI peuvent aider à les réduire en assurant la fluidité de la circulation. Il est aussi possible de gérer et de surveiller les zones environnementales par le biais des STI. Certes, ils peuvent être utilisés pour signaliser les zones infectées ou même pour surveiller la qualité de l'air[6].

I.6.STI dans les pays développés

I.6.1. Singapour

En tant que nation insulaire aux ressources naturelles limitées, Singapour est confrontée aux défis d'une population urbaine croissante et du manque d'espace physique disponible. Le système STI de Singapour est un exemple frappant de solutions de mobilité intelligente réussies et tient en partie à ce que Singapour soit devenue l'une des grandes villes les moins congestionnées au monde[10].

Les solutions technologiques clés utilisées dans le cadre des STI comprennent :

- **Système de surveillance et de conseil pour les autoroutes**

  Surveille le trafic et améliore la sécurité sur les autoroutes pour une réponse rapide aux incidents. Il fournit des informations en temps réel sur les lieux des incidents et les temps de parcours sur les autoroutes[10]. Il existe aussi un Réseau de caméras de surveillance « Junction Electronic Eyes » (JEyes) (la figure I.7) permettant de surveiller les conditions de circulation dans les principales jonctions signalées. Cela aide à dissuader le stationnement et le chargement / déchargement illégaux sur les routes principales[10].



Figure I.7 : Camera J Eyes

- **Système de guidage de stationnement**

  Fournit des informations en temps réel sur les parcs de stationnement disponibles, assurant ainsi une utilisation efficace des installations de stationnement.

- **Système de détermination de la liaison verte**

  Le système de détermination de la liaison verte contrôle les feux de circulation (la figure I.8) et ajuste la durée de la lumière verte sur la base du volume de trafic en temps réel. Il fournit également un lien entre les jonctions adjacentes afin de minimiser le nombre d'arrêts des véhicules[10].



Figure I.8 : Feu de circulation tricolore

- **Panneaux de signalisation électroniques**

  Panneaux de signalisation à LED qui indiquent les virages interdits à certaines périodes. La figure (I.9) présente quelques panneaux de signalisation électronique.



Figure I.9 : panneaux de signalisation électroniques

I.6.2. New York (États Unis d'Amérique)

La ville de New York (la figure I.10) possède l'un des STI les mieux établis au monde, contribuant à réduire les encombrements, à améliorer la sécurité et à atténuer les impacts environnementaux négatifs liés aux transports.



Figure I.10 : Route dans la ville de New York

Programme STI relatif aux routes visant à améliorer la mobilité, les opérations, les systèmes véhiculaires, l'environnement et la sécurité grâce à l'utilisation de nouveaux systèmes et technologies de transport[10].

- **Centre de gestion du trafic**

  Il contient 86 caméras de vidéosurveillance et vise à optimiser l'efficacité de l'assistance aux opérations, à la maintenance et à l'ingénierie du trafic[10].

- **Système de contrôle de la circulation automobile**

  Le système de contrôle de la circulation fait organiser les routes et réduire la congestion, il contrôle 6 600 feux de circulation aux intersections et surveille 4 000 détecteurs de boucle dans la ville[10].

- **Système avancé de sécurité des piétons**

  Il comprend des capteurs de poids sur les trottoirs qui, lorsqu'ils sont piétinés, allument un panneau de passage pour piétons suspendu pour avertir les conducteurs que quelqu'un utilise le passage pour piétons.

- **Système d'information avancé pour les voyageurs**

  Ce système permet aux utilisateurs de visualiser les conditions de circulation en temps réel grâce à 78 caméras fournissant à la fois la diffusion d'images et la vidéo en continu[10].

- **Bureau d'intervention d'urgence**

  Il traite des urgences liées au trafic.

- **Caméras de feu rouge**

  Ils aident à constater les violations des feux de circulation[10].(voir la figure I.11).



Figure I.11 : Modèle de camera de feu rouge

I.6.3.Londres (Royaume-Uni)

Les investissements dans les STI ont été réalisés dans les domaines suivants : Système de transport intelligent de surface (SITS) : Ce projet vise à gérer efficacement l'espace de chaussée, les pistes cyclables et les autoroutes de la ville.

- **Technique d'optimisation de décalage de cycle fractionné (SCOOT)**

  Permet d'optimiser le minutage des feux de circulation dans le grand Londres. Il a été prouvé qu'il réduisait les retards jusqu'à 12% à chaque jonction, où qu'elle soit installée[10].

- **Signalisation routière numérique**

  Fournit aux personnes des informations en temps réel sur les trajets empruntant les itinéraires principaux pour Londres[10].

- **Heathrow pod**

  Un système de transport en commun rapide (la figure I.12) et zéro émission. Le voyage à travers les nacelles s'est avéré bénéfique, car il n'a plus besoin de voyager dans un bus, auparavant responsable de l'émission de 100 tonnes de dioxyde de carbone chaque année[10].



Figure I.12 : Heathrow pod

- **Optimisation des feux de circulation**

  Optimisez les horaires des feux de circulation en fonction du volume de véhicules sur les routes[10].

I.7.Supervision dans les STI

Au centre d'ingénierie et de gestion de trafic, la supervision est la fonction qui consiste à indiquer et à commander l'état d'un appel, d'un système ou d'un réseau[11]. La supervision inclut donc plusieurs activités :

- Surveiller les routes et l'environnement et assurer la sécurité routière.
- Visualiser les états des feux de circulation et les débits de mobilité dans les routes ainsi que la diffusion des caméras surveillance…etc.
- Analyser les informations obtenues par les équipements de STI installées dans la ville (capteurs, détecteurs…etc.) ou les informations de la météo
- Prendre la décision : la diffusion des informations ou consignes aux usagers (panneaux à message variable, signaux directionnels...), faire contrôler les feux de circulation ou intervenir en cas d'un incendie ou accident.

I.8.Gestion des feux de circulation

I.8.1. Fonctionnement d'une intersection

Le modèle d'intersection classique est représenté dans la figure (I.13). Il est composé d'un ensemble de quatre directions ou routes possibles. Ce modèle est celui utilisé dans la plupart des travaux de la littérature, mais rien n'interdit de l'étendre. Chaque direction peut être décomposée en une direction entrante et une direction sortante, ou bien des deux. Une direction entrante permet aux véhicules, répartis sur une ou plusieurs voies, de traverser l'intersection vers une direction sortante, qui fait le lien avec une intersection voisine. Sur chaque direction entrante, les véhicules tournant à gauche utilisent la voie la plus à gauche, tandis que la voie la plus à droite est généralement utilisée pour les véhicules allant tout droit ou tournant à droite[12].(E(Est), O(Ouest, N(Nord), S(Sud))

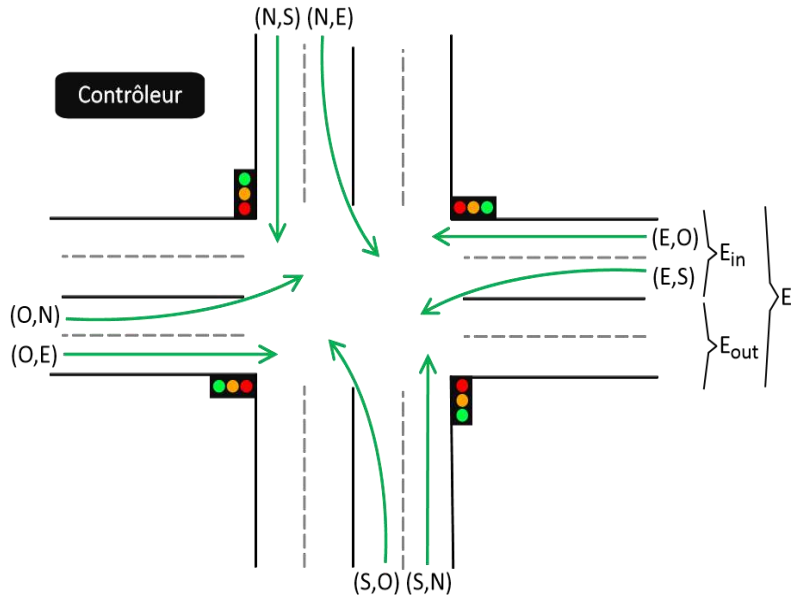


Figure I.13 : Modèle d'intersection généralement utilisé

La figure (I.14) montre le diagramme des phases.

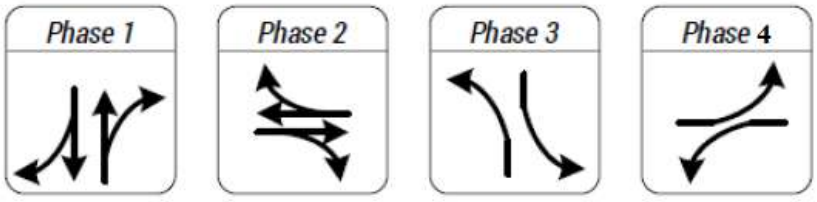


Figure I.14 : Diagramme des phases

I.8.2. Contrôleur des feux de circulation

De nos jours, les feux de circulation d'une intersection sont généralement gérés par une boite de contrôle, qui va posséder plus ou moins de propriétés en fonction des constructeurs. Typiquement, une boite est rattachée à une seule intersection et possède les éléments principaux suivants[12] :

- Une unité d'énergie.
- Une unité de détection, connectée à des éléments de contrôle (détecteurs).
- Une unité de contrôle, donnant l'ordre d'enclenchement des feux.
- Une unité d'avertissement rapide, réagissant en cas d'erreur critique (par exemple : orange clignotant sur l'ensemble des feux.).
- Une unité de gestion des conflits, qui est programmée avec les combinaisons des feux verts autorise (matrice de conflits) et qui vérifie les données envoyées par l'unité de contrôle : elle fait appel à l'unité précédente en cas d'erreur ou de faute constatée sur l'un des feux.
- Une unité d'administration, pour prendre le contrôle du carrefour (par la police par exemple).

I.8.3. Contrôle des feux de circulation

- **Première génération**

  Contrôle à temps fixe : en fonction de l'heure et parfois du jour, le système va utiliser un plan de feux prédéfini. Exemple : une configuration stricte est appliquée de 12h30 à 14h, tandis qu'une plus souple et équitable pour l'ensemble des voies d'un carrefour est appliquée à 3h du matin. Classiquement, trois configurations existent : matin, après-midi, et "reste". On peut imaginer quelque chose de plus complet prenant en compte des plans spécifiques aux heures de pointe[8].

- **Deuxième génération**

  Contrôle à temps dynamique - réactif : des détecteurs sont utilisés afin de recueillir les données du trafic périodiquement (plusieurs minutes ou cycles). Ces données sont ensuite utilisées pour mettre en place un plan de feux en réponse aux informations reçues. Cette méthode est simple à mettre en place, mais nécessite toutefois une très bonne connaissance

du système afin d'être efficace. Il s'agit de la première méthode à être apparue aux États-Unis à la fin des années 1980 avec l'apparition des UTCS (Urban traffic control software)[12].

- **Troisième génération**

Contrôle à temps réel - adaptatif : reprend le même principe que la deuxième génération mais cette fois-ci en temps réel. Ce type de contrôle programme dynamiquement les plans de feux en se servant des paramètres recueillis sur le terrain. Il est possible de calculer des valeurs telles que le temps de cycle, les phases ou encore leur ordonnancement. L'opération est effectuée de manière adaptative, c'est à dire en réaction directe au trafic. L'avantage de cette méthode est qu'elle peut s'adapter à de multiples situations, mais reste la plus compliquée à mettre en place[8].

I.8.4. Systèmes de régulation du trafic

- **SCOOT (Split Cycle Offset Optimization Technique)**

Est un système de contrôle développé par le TRL (Trafic research laboratory, Angleterre). À l'aide de détecteurs placés sur le terrain, SCOOT se base notamment sur un indice de performance afin de générer des plans de feux en fonction de la demande des utilisateurs. Cet indice est calculé par rapport au délai d'attente moyen, à la longueur des files d'attente et des arrêts sur le réseau. Cet aspect dynamique est réalisé à l'aide de mesures et de décisions entre les équipements sur le terrain et un centre de contrôle[12].

- **SCATS (Sydney Coordinated Adaptive Traffic System)**

Le système de contrôle de trafic adaptatif coordonné de Sydney a été mis au point par la Roads and Traffic Authority (RTA) de la Nouvelle-Galles du Sud, en Australie, à la fin des années 1970. SCATS est une stratégie de contrôle de signal hiérarchique, cyclique et sensible au trafic. Les opérateurs de système peuvent choisir entre différentes combinaisons d'objectifs, notamment minimiser les arrêts, minimiser les retards et maximiser le débit. Semblable à SCOOT, SCATS ajuste le temps de cycle, les fractionnements et les décalages en réponse à la demande de trafic en temps réel afin de minimiser les arrêts et les retards[12].

I.9.Conclusion

Le domaine des systèmes de transport intelligents est vaste, riche et décisif que ce soit dans la vie quotidienne ou encore dans le contrôle du trafic routier. Dans ce chapitre, nous avons évoqué quelques généralités sur les systèmes de transport intelligents et les technologies utilisées dans ce domaine et comment le STI faire le confort, la sécurité et peuvent de résoudre les problèmes liés à la mobilité. Nous avons montré aussi la gestion des feux de circulation et les systèmes de régulation utilisées dans le monde.

Chapitre II :

Solutions hardwares et softwares

II.1.Introduction

Actuellement l'internet des objets est la tendance dans le monde de l'informatique et système embarqué. L'Internet des Objets (l'IoT) repose sur l'idée que tous les objets peuvent être connectés à Internet et sont donc capable d'émettre de l'information et de recevoir des commandes. L'IoT est capable d'être appliqué dans tous les domaines de notre vie.

Dans ce chapitre, nous présenterons les notions fondamentales de l'IoT et les bases théoriques et techniques nécessaires pour réaliser notre projet.

II.2.Notion sur l'internet des objets (IoT)

II.2.1.Internet des objets

L'Internet des objets (IdO, ou en anglais « Internet of Things » IoT), est un nouveau concept ayant prouvé rapidement son succès. L'IdO correspond à un ensemble d'objets physiques et intelligents connectés qui communiquent via de multiples technologies avec diverses plateformes de traitement de données, en lien avec les vagues du Cloud et du Big data. Les objets intelligents peuvent être considérés comme des machines, des appareils, des capteurs, des actionneurs, des véhicules, tous dispositifs équipées de capacités de communication filaires ou sans fil, de telle sorte qu'il y ait une fusion entre le monde réel (physique) et le monde numérique (virtuel)[13].

L'objectif principal de son développement consiste à rendre certaines tâches faciles pour les utilisateurs et les environnements intelligents, en utilisant la fonctionnalité de contrôle à distance des choses. Cette technologie est d'abord appliquée aux capteurs et aux actionneurs, ensuite, elle est étendue à différents systèmes, à savoir : les réseaux intelligents, les maisons intelligentes, les transports intelligents et les villes intelligentes[13].

II.2.2.Qu'est-ce qu'un Objets ?

L'apparition du concept de l'IdO est due à la variété des équipements et des objets utilisés dans notre vie quotidienne (voir figure I.1) : ordinateurs, capteurs, actionneurs, smartphones, véhicules connectés, smart homes, etc. Les objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés. Un objet connecté est un matériel électronique qui peut communiquer avec un Smartphone, une tablette tactile et/ou un ordinateur, ses

caractéristiques pouvant évoluer au cours du temps (position, niveau de batterie, etc.). Il peut envoyer et recevoir des informations, via une liaison sans fil, Bluetooth ou Wifi, etc. D'autres définitions s'accordent à dire qu'un objet connecté possède des capacités de calcul, d'acquisition (capteur) et d'action (actionneur). Les objets identifiés par RFID (Radio Frequency Identification) sont exclus de cette définition, car une puce RFID classique ne peut pas être considérée comme un dispositif de calcul, celle-ci représente un identifiant.

L'intérêt principal d'un objet connecté est l'interactivité, la possibilité de récupérer des informations, ou d'envoyer des statistiques, de garder le contact, etc.[13].



Figure II.1 : Quelques exemples d'objets connectés.

II.2.4.Applications de l'Internet des Objets

- **Transport**

  L'Internet des Objets (IoT) a le potentiel pour transformer l'industrie des transports (la figure II.2) en modifiant profondément la façon de rassembler les données. Les données collectées à partir de ces équipements peuvent alors être analysées par les organismes de transport pour[14] :

  - Améliorer l'expérience du passager avec des transports plus fiables, de meilleurs services à la clientèle, une communication et des informations plus précises[14].
  - Augmenter la sécurité, grâce aux capteurs de données qui détectent toutes les anomalies dans la vitesse des trains, les températures de la chaussée, l'état des pièces des avions, ou encore le nombre de voitures qui attendent à une intersection et gerber les trafics routiers[14].

- Améliorer la performance opérationnelle, en surveillant de manière proactive les infrastructures critiques et créer des processus plus efficaces, pour réduire les coûts d'exploitation et améliorer la capacité du système[14].

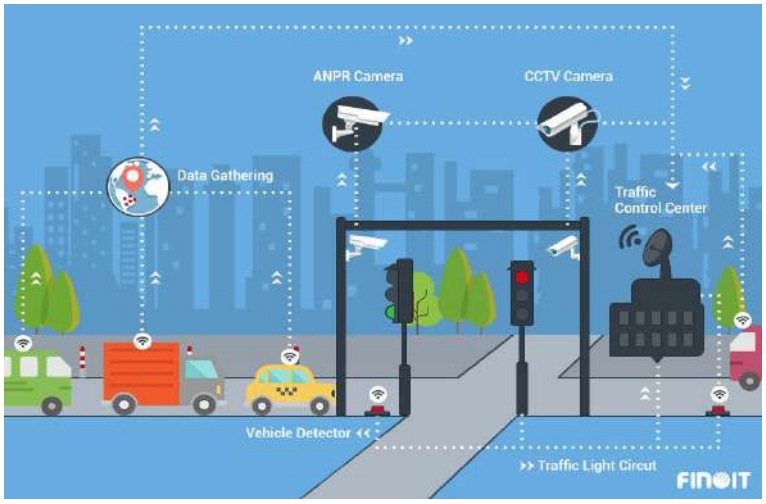


Figure II.2 : Application de l'IoT dans le transport

❖ **Smart Cities**

Efficace, innovant et participatif, le concept smart cities offre à ses habitants une qualité de vie maximale avec une consommation de ressources minimale grâce à une combinaison intelligente des infrastructures (énergie, transport, communication) aux différents niveaux hiérarchiques (ville, quartier, bâtiment)[15] comme elle est montrée dans la figure(II.3).

Grâce à des capteurs (éléments physiques intégrés dans l'IoT), les données produites et récoltées (de manière sécurisées) peuvent être réutilisées par la population pour développer de nouveaux services et participer à l'amélioration de la vie urbaine[15].



Figure II.3 : Application de l'IoT au smart cities

❖ **Industrie**

L'IoT industriel est un système comprenant des capteurs intelligents, des machines, des outils, des plates-formes logicielles, des serveurs et des applications en nuage. Des capteurs intelligents sont déployés à chaque étape de la fabrication pour des applications spécifiques. Ces réseaux de capteurs envoient en permanence des données à la passerelle IoT (qui joue le rôle de concentrateur entre les périphériques IoT et le Cloud), qui les reçoit et les transmet au serveur d'applications Cloud pour traitement et analyse. Des programmes d'application sophistiqués sont développés pour gérer de grandes quantités de données au sein d'un réseau sécurisé. Ils sont accessibles via les applications Smartphone[16].

La précision des étapes de processus peut être améliorée à un niveau supérieur grâce à l'automatisation de la machine. Des outils d'automatisation tels que les automates programmables sophistiqués qui sont utilisés avec des réseaux de capteurs intelligents connectés à un système de Cloud central qui collecte une énorme quantité de données. Des logiciels et des applications spécialement conçus permettent d'analyser les données et leur comportement afin de les améliorer. L'automatisation industrielle améliore la précision et l'efficacité : réduit les erreurs, facile à contrôler et accessible à distance via des applications[16].



Figure II.4 : Exemple d'un Robot connecté [https://www.fanuc.eu/uk/en]

❖ **Agriculture Intelligente**

L'agriculture intelligente a pour objectif d'augmenter la productivité des systèmes agricoles et améliorer la sécurité alimentaire. Cet objectif requiert l'utilisation de nouvelles technologies, telles que l'imagerie satellitaire et l'informatique.

Il s'appuie sur l'utilisation des capteurs qui vont s'occuper de récolter les informations utiles sur l'état du sol, taux d'humidité, taux des sels minéraux, etc. et envoyer ces informations au fermier pour prendre les mesures nécessaires garantissant la bonne production[1]. Comme elle est présentée dans la figure (II.5) le fermier peut utiliser une tablette par exemple comme une interface pour collecter les données envoyés par des capteurs superviser la serre et contrôler des dispositifs comme les électrovannes, les ventilos…etc.



Figure II.5 : Utilisation de IoT pour superviser et contrôler la serre

❖ **Domotique**

La domotique est l'ensemble des techniques de l'électronique permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, caméras, volets roulants, porte de garage, lumières…etc.) (La figure II.6).

La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics…etc.[1].



Figure II.6 : Internet des objets dans la domotique

II.2.5.Technologies de l'internet des objets

- **Capteurs**

Les objets du quotidien ou les appareils équipés de capteurs, par exemple, possèdent des capteurs de mesure qui enregistrent les états physiques ou chimiques. Ils mesurent la température, la pression, la luminosité, l'humidité, le pH ou le mouvement. Afin de rendre les résultats de mesure utilisables numériquement, ils les traduisent en signaux électriques. Le capteur de luminosité d'un smartphone mesure l'intensité lumineuse de l'environnement. Avec ces informations, l'écran peut s'adapter au niveau de luminosité[17].

- **Réseau de capteurs sans fil (RCSF)**

Les progrès de la technologie des communications sans fil et de l'électronique ont permis la mise au point de nœuds de capteurs multifonctionnels, économiques et à faible coût, de petite taille et qui communiquent sans lien sur de courtes distances. Ces nœuds de capteurs minuscules et généralement simples se composent d'unités de détection, de traitement de données et de composants communicants. Un grand nombre de ces nœuds déployés sur de grandes zones peuvent collaborer les uns avec les autres. Pour être rentables, les nœuds de capteurs fonctionnent souvent avec des réserves d'énergie très limitées[18]. Il répond aux exigences des applications IoT (fiabilité, économique et fournir des informations en temps réel).

- **RFID (Radio Frequency Identification)**

  Cette technologie permet l'identification sans contact d'un objet au moyen d'ondes électromagnétiques. Pour permettre au lecteur de le reconnaître et de le localiser, l'objet est muni d'une étiquette radio et d'un code indubitable. Les systèmes RFID ont une portée allant jusqu'à 100 mètres. Un exemple d'application est dans l'industrie de la logistique, où les conteneurs peuvent être mieux localisés par RFID lorsqu'ils sont expédiés[17].

- **Cloud**

  Ces réseaux virtuels de stockage et de traitement de données sont également essentiels pour l'infrastructure d'un Internet des objets à maillage large. Par exemple, le Cloud permet d'externaliser le stockage d'objets en réseau ou d'augmenter leur capacité de stockage[17].

  Pour une grande partie de l'Internet des objets, la tête (ou plutôt le cerveau) du système est dans le nuage. Les capteurs et les appareils collectent des données et effectuent des actions, mais le traitement / la commande / l'analyse (ou «fonctions intelligentes») se produit généralement dans le Cloud[19].

Dans la communication, il existe plusieurs technologies de communications longue portées et courte portées (Zigbee, Bluetooth, Wifi, Z-wave, NFC, Zigfox, LoRa…etc.). On présente quelques technologies de communication sans fils utilisé dans l'internet des objets et le débit de chaque technologie (Voir la figure III.7).

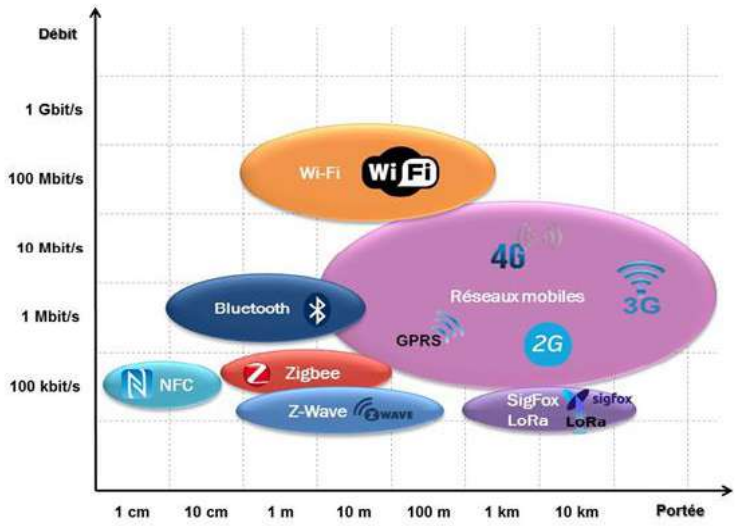


Figure II.7 : Technologies de communication sans fils et leurs débits

- **Bluetooth Low Energy (BLE)**

Le Bluetooth Low Energy est très largement utilisé dans le monde. Quasiment tous les smartphones sont équipés de cette techno, fréquemment utilisée pour faire communiquer les wearables. Il a une portée de 60 mètres en terrain dégagé et consomme environ 20 fois moins d'énergie que le Wifi. La dernière version de cette techno est plus adaptée à l'IoT et dispose d'une portée deux fois supérieure à celle de son aînée.

Ce réseau de courte portée permet de transporter nettement moins d'infos que le Wifi : 1 mbps seulement[20].

- **Zigbee**

ZigBee et ses différents profils industriels sont basés sur le protocole IEEE802.15.4, une technologie de réseau sans fil standard. Il est destiné aux applications nécessitant des transferts de données limités à des taux de transfert faibles dans un rayon de 100 m, généralement dans une maison ou un bâtiment. Il présente des avantages dans les systèmes complexes nécessitant un fonctionnement à faible consommation, des niveaux de sécurité élevés, une extensibilité élevée, un nombre de nœuds élevé et peut prendre en charge les réseaux de capteurs et de contrôle sans fil dans les applications IoT[20].

  - Fréquence : 2,4 GHz
  - Portée : moins de 100m, avec débits de données 250kbps.

- **Wifi**

La connectivité Wifi est un choix évident pour les développeurs, en particulier dans les environnements domestiques et les réseaux locaux. Il permet un transfert rapide des données et peut gérer de grandes quantités de données. Mais sa consommation électrique risque d'être trop élevée pour de nombreuses applications IoT[20].

  - Fréquences : bandes 2,4 GHz et 5 GHz
  - Portée : environ 50m
  - Débits de données : 150 à 200 Mbits / s est typique, 600 Mbits / s maximum

- **Réseaux cellulaires mobiles**

Toute application IoT nécessitant des opérations sur de longues distances peut tirer parti de Cellular GSM, 3G, 4G. Cellular convient à de gros volumes de données, mais le coût et la consommation d'énergie pour la gestion de gros volumes de transfert de données risquent

d'être trop élevés pour la plupart des applications IoT. Cellular convient aux projets utilisant des données faibles et pilotés par capteurs, transférés sur Internet[20].

- Fréquences : 900, 1800, 1900, 2100 MHz
- Portée : 35 km max pour le GSM
- Débits de données : moins de 170kps GPRS, moins de 384kbps EDGE, moins de 2Mbps, moins de 10Mbps HSP, 3-10Mbps LTE

II.2.6.Architecture de l'internet des objets

L'architecture la plus basique est une architecture à trois couches. Il a été introduit au tout début de la recherche dans ce domaine. Il comporte trois couches, à savoir les couches de perception, de réseau et d'application[21] (Voir la figure II.8).

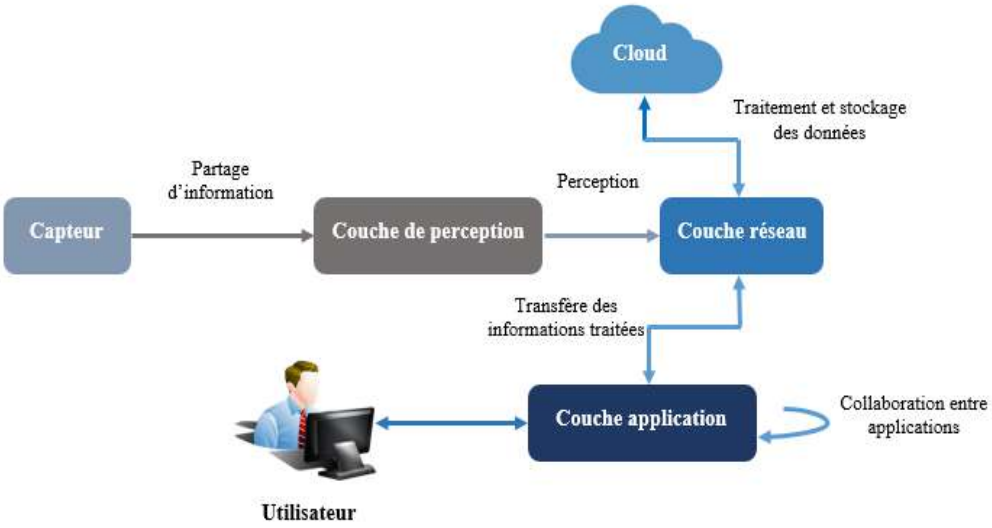


Figure II.8 : Modèle architecture de l'internet des objets

❖ **Couche de perception**

C'est la couche physique, qui possède des capteurs pour détecter et collecter des informations sur l'environnement. Il détecte certains paramètres physiques ou identifie d'autres objets intelligents dans l'environnement[21].

- **Couche réseau**

  Cette couche est responsable de la connexion à d'autres objets intelligents, périphériques réseau et serveurs. Ses fonctionnalités sont également utilisées pour la transmission et le traitement des données de capteurs[21].

- **Couche application**

  Elle est chargée de fournir des services spécifiques à l'application à l'utilisateur. Il définit diverses applications dans lesquelles l'Internet des objets peut être déployé, telles que les maisons intelligentes, les villes intelligentes et la santé intelligente[21].

II.3.Outil de développement (Node Red)

II.3.1.C'est quoi le Node Red ?

En 2013, IBM a développé ce projet en tant qu'open source, afin de connecter le matériel et les périphériques à des services et logiciels Web. De cette manière, les périphériques IoT sont connectés et contrôlent le monde réel.

Node-RED est un outil puissant pour la construction d'applications Internet des objets (IoT), l'accent étant mis sur la simplification du câblage des blocs de code pour la réalisation de tâches. Il utilise une approche de programmation visuelle permettant aux développeurs de connecter ensemble des blocs de code prédéfinis, appelés «nœuds» (« nodes » en anglais), pour effectuer une tâche. Les nœuds connectés, généralement une combinaison de nœuds d'entrée, de traitement et de sortie, forment ensemble un « flux » (flow en anglais).

C'est un outil de développement basé sur les flux dans lequel l'exécution est construite sur Node.js. Les flux créés sont stockés dans JSON. En stockant dans JSON, nous pouvons partager, importer et exporter dans le monde réel. Une bibliothèque de flux en ligne vous permet de partager vos meilleurs flux avec le monde.

II.3.2.Editeur Node Red

- **Différentes parties de Node-Red**

  La figure (II.8) présente l'éditeur « Node Red ».A gauche les modules/fonctions appelés « Nodes» installées. D'autres modules sont à disposition sur internet Au milieu les tâches/programmes appelé « Flow » c'est notre implémentation En haut toutes les tâches

qui tournent simultanément sur notre machine/CPU. A droite on trouve la fenêtre info /Debug.

Une fois édité, on peut charger le flow dans la machine (Deploy). Lors de l'activation, il se peut que nous recevions des messages d'erreur, manque d'information etc. à corriger.

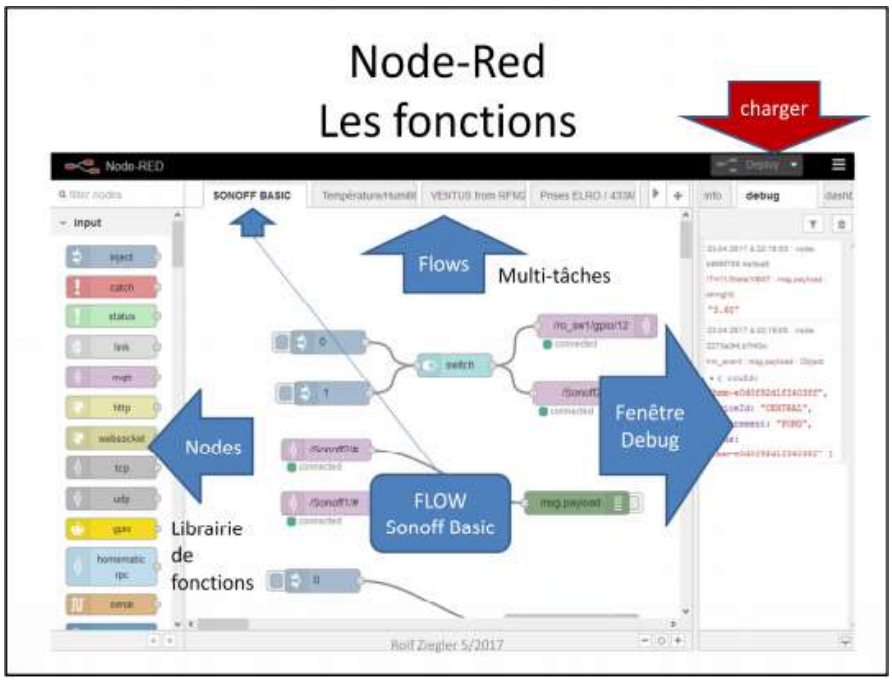


Figure II.8 : Editeur « Node Red »

Les nœuds connectés comprennent les nœuds d'entrée, les nœuds de traitement et les nœuds de sortie (la figure II.9).



Figure II.9 : Comment construire un flux (flow)

❖ **Nœuds d'entrée**

Il vous permet d'entrer les données d'entrée dans le flux. Ils ont un carré gris sur le côté droit que vous pouvez connecter aux autres nœuds. En connectant ce nœud à un autre nœud, vous pouvez envoyer les données d'entrée.

Vous pouvez également connecter des services tels que Twitter, Google, des sockets Web et série, et envoyer ces données sous forme d'entrées aux autres nœuds. Sinon, vous pouvez

entrer manuellement les données d'entrée à l'aide du nœud appelé «injecter»[22]. (Voir la figure II.10).

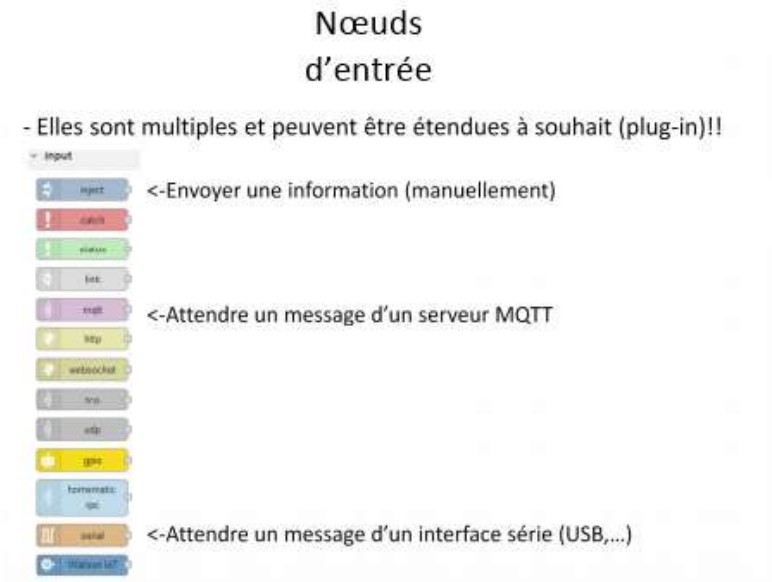


Figure II.10 : Nœuds d'entrée utilisées dans Node Red

❖ **Nœuds de traitement**

Les nœuds de traitement sont utilisés pour traiter les données dans lesquelles ils ont un point de terminaison d'entrée et 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. On montre les différents nœuds de traitement dans la figure (II.11).



Figure II.11 : Nœuds de traitement utilisées dans Node Red

❖ **Nœuds de sortie**

Il vous permet d'envoyer les données vers le monde extérieur, vers d'autres services, par exemple Twitter. Il y a un carré gris sur le côté gauche auquel ils peuvent être connectés à un noeud d'entrée ou à une fonction. Si vous ne souhaitez pas envoyer les données à des services tels que Twitter ou des nœuds de messagerie, vous pouvez simplement déboguer ces messages en utilisant l'onglet «deploy» dans le coin supérieur droit de l'éditeur[22].(voir la figure II.12)



Figure II.12 : Nœuds De sortie utilisées dans Node Red

❖ **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. (Voir la figure II.13)

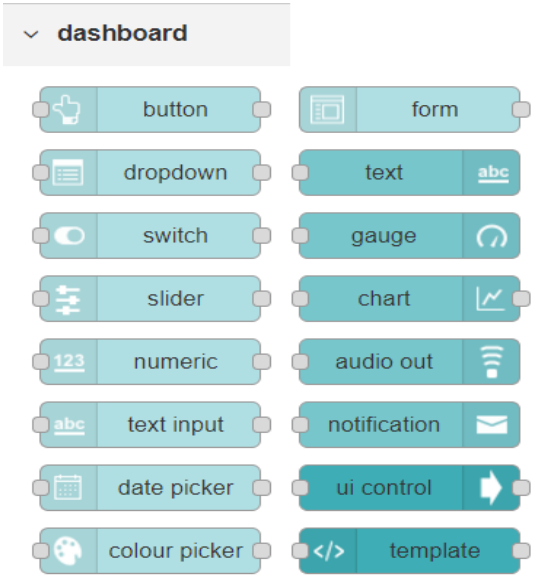


Figure II.13 : Palette utilisée dans le tableau de bord (Dashboard)

Les nœuds de tableau de bord de Node-RED fournissent un ensemble complet de composants d'interface utilisateur permettant de créer des tableaux de bord de base adaptés à l'Internet des objets (IoT) : graphiques, jauges, texte de base, ainsi que des curseurs et des entrées. Cependant, il y aura toujours des situations où vous aurez besoin de quelque chose de personnalisé. Le nœud de modèle est la solution et dans ce tutoriel, nous allons vous montrer un exemple d'utilisation rapide pour créer votre propre widget d'interface utilisateur. On montre les différents éléments dans le tableau de bord (Dashboard) dans la figure (II.14).

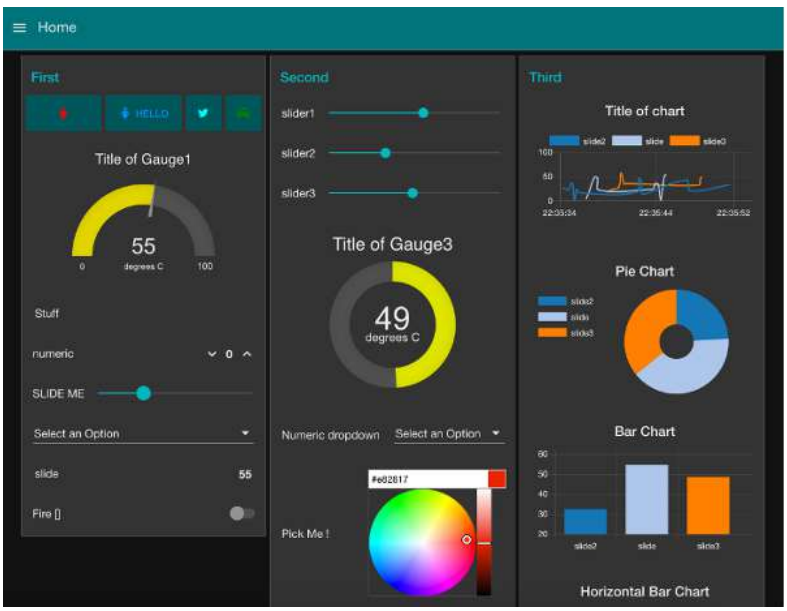


Figure II.14 : Eléments de tableau de bord (Dashboard)

II.4.Raspberry

Le Raspberry Pi est un nano-ordinateur de la taille d'une carte de crédit. C'est à dire que l'ensemble des fonctions d'un ordinateur, le microprocesseur, la mémoire RAM, la carte vidéo, la carte Ethernet, le wifi, le Bluetooth…sont tous inclus dans un seul et unique circuit intégré. La carte électronique à peine plus grande qu'un paquet de cigarettes est bâtie autour de ce composant. Le Raspberry Pi est livré sans système d'exploitation, sans clavier, ni écran mais possède toute la connectique pour pouvoir les brancher[23](voir la figure II.15).



Figure II.15 : Carte Raspberry Pi 3 B+

II.4.1.Fiche technique de Raspberry

- Alimentation à prévoir : 5 Vcc/maxi 2,5A.
- Un processeur Broadcom BCM2837B0 ARM Cortex-A53 64 bits quatre cœurs à **1,2** GHz et 1GB de mémoire RAM.
- Compatible Wifi 802.11 b/g/n/ac double bande 2,4 Ghz et 5 Ghz et Bluetooth 4.2 (Bluetooth Classique et BLE).
- 4 ports USB 2.0.
- Un port Ethernet RJ45 (10/100/300 Mbps).
- Une sortie vidéo HDMI.
- Une sortie audio Jack 3,5 mm en stéréo.
- Un Support pour cartes micro-SD.
- Un connecteur GPIO avec 40 broches d'E/S.

❖ **Port à usage générale entrées/sorties de Raspberry (GPIO)**

Le port GPIO à 40 broches est le moyen fourni par les créateurs du Raspberry Pi pour nous permettre d'accéder aux ports d'entrée/sortie du processeur BCM2837 (voir la figure II.16).

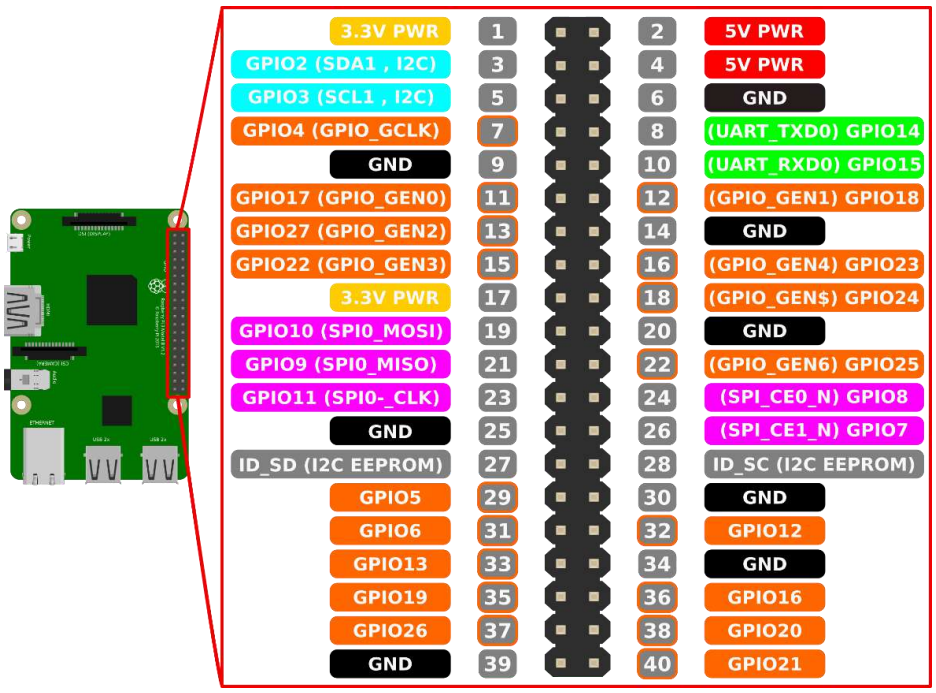


Figure II.16 : Fonctions des broches de GPIO Rpi 3 B+

Les ports du connecteur GPIO ont tous un rôle bien déterminé. On trouve des alimentations 5V et 3,3V ainsi que des **masses.** Les ports GPIO sont identifiés par un chiffre ; GPIO1, GPIO2, … Certains ont une fonction supplémentaire précisée entre parenthèses. Cette fonction n'empêche pas d'utiliser les GPIO de façon classique, comme entrée/sortie numérique[24].

Les GPIO utilisables comme entrée/sortie numérique sont au nombre de **26**. Ils ne fonctionnent qu'en tout ou rien, 0 ou 1, 0V ou 3,3V. Il n'y a pas de port analogique. Si vous avez besoin de plus de ports ou de ports analogiques, il faudra ajouter des cartes d'extension[24].

II.5.Arduino

Arduino est une plate-forme open-source utilisée pour la construction de projets électroniques. Arduino consiste à la fois en une carte de circuit physique programmable (souvent appelée microcontrôleur) et en un logiciel, ou IDE (environnement de développement intégré) exécuté sur un ordinateur, utilisé pour écrire et télécharger du code informatique sur la carte physique.

La plate-forme Arduino est devenue très populaire, l'IDE Arduino utilise une version simplifiée du C ++, ce qui facilite l'apprentissage de la programmation. Enfin, Arduino fournit un facteur de forme standard qui décompose les fonctions du microcontrôleur en un package plus accessible. L'Uno est l'un des conseils les plus populaires de la famille Arduino et un excellent choix (voir la figure II.17).



Figure II.17 : Carte Arduino Uno

II.5.1.Caracteristiques principales

- alimentation
  - via port USB (Programmer ou peur être alimenter).
  - 7 à 12 V sur connecteur Jack DC 5,5 x 2,1 mm
- Microcontrôleur ATMega328
- Un oscillateur à quartz 16 MHz.
- mémoire flash 32 Kb, mémoire SRAM 2 kb et mémoire EEPROM 1 Kb.
- 14 broches d'E/S dont 6 PWM (Pulse Width Modulation)
- 6 entrées analogiques 10 bits.
- bus série I2C et SPI.
- gestion des interruptions, bouton Reset.
- Régulateur de tension.

La figure (II.18) présente les composants de la carte Arduino.

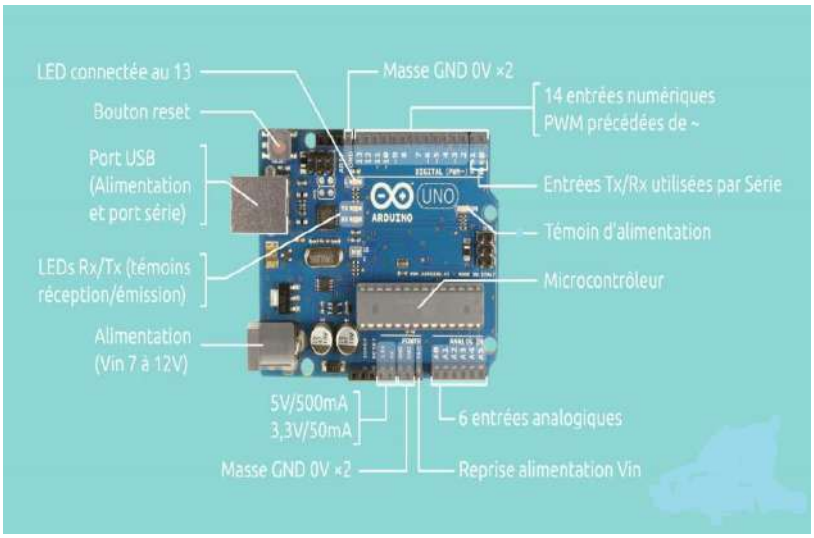


Figure II.18 : Composants principale de l'Arduino

II.5.2.Shield Arduino

Shields Arduino sont des cartes qui se branchent sans soudure aux cartes Arduino ou à d'autres Shields Arduino pour augmenter leurs capacités. Les Shields Arduino conservent l'esprit original d'Arduino, facile à produire et à utiliser. Les cartes Arduino (voir la figure II.19).

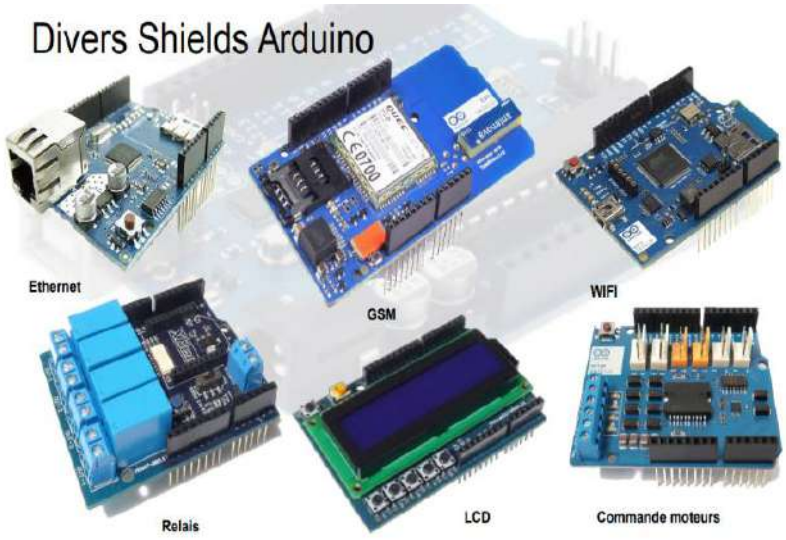


Figure II.19 : Shields utiliser avec Arduino

II.6.Logique floue

II.6.1.Théorie de la logique floue

La logique floue est un type de modélisation qui s'intéresse à la prédiction d'une variable catégorielle *Y* « subjective » au sens où elle n'est pas objectivable : elle dépend de l'observateur (l'individu est « grand », « moyen » ou « petit »). Ce cadre sort de la statistique classique dans lequel la valeur de la variable *Y* est objectivable (« l'individu mesure 176 cm »). L'application de la logique floue revient à tenter d'appliquer un raisonnement proche de la pensée humaine[25] :

- Les variables prédictives (comme la variable à prédire) sont catégorielles avec des modalités subjectives (« grand », « petit ») et non pas de données objectivables (176 cm). Ces variables catégorielles sont appelées « variables linguistiques ». Dans le cadre statistique usuel, la variable continue initiale (ici la taille en cm) peut être discrétisée pour donner des intervalles distincts, par exemple : « petit < 170cm < moyen < 180cm <grand ». La logique floue vise à prendre en compte les incertitudes qui existent au voisinage des seuils (due en partie à des principes de subjectivité)[25].
- Une donnée peut appartenir à plusieurs modalités d'une même variable (un individu de 165 cm peut être considéré comme petit mais aussi comme moyen). Les classes définies ne partitionnent donc pas l'ensemble des possibles car elles peuvent se recouper[25].
- La logique floue intègre un ensemble de règles permettant d'attribuer (d'une manière logique) une sortie à une entrée[25].

II.6.2.Fonctions d'appartenance

Les fonctions d'appartenance peuvent théoriquement prendre n'importe quelle forme, mais en général, les fonctions d'appartenance les plus utilisées sont définies par des formes géométriques ou des fonctions.

Les fonctions d'appartenance sont le plus souvent représentées par les fonctions triangulaire et trapézoïdale (la figure II.21), donc par des segments de droite, les fonctions sont alors dites linéaires par morceaux. Dans la plupart des cas, et en particulier pour le réglage par la logique floue, ces deux formes sont suffisantes pour délimiter des ensembles flous[25].

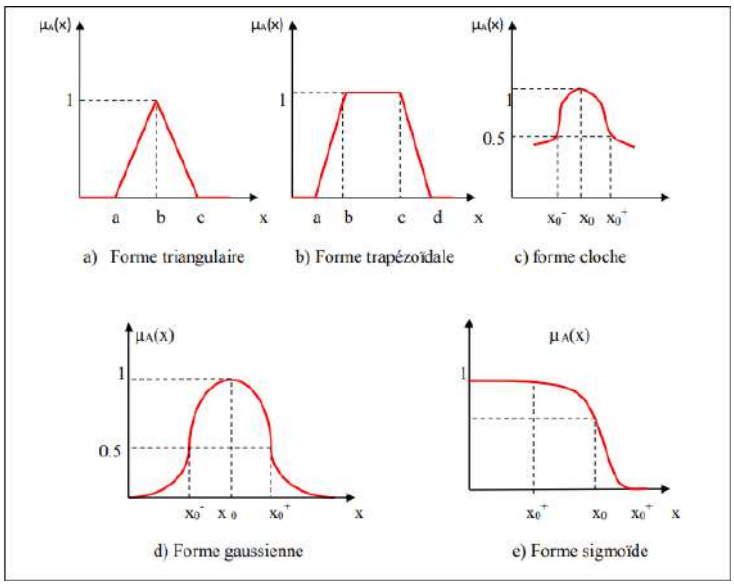


Figure II.21 : Formes des fonctions d'appartenance

II.6.3.Opérateur de la logique floue

On a résumé dans le tableau (II.1) les opérateurs de Zadeh et les opérateurs probabilistes

Tableau II.1 : Opérateurs de Zadeh et de probabilités

	Opérateurs floue de Zadeh	**Opérateurs probabilistes**
A ET B	$Min\ (\mu_A, \mu_B)$	$\mu_A \times \mu_B$
A OU B	$Max\ (\mu_A, \mu_B)$	$\mu_A + \mu_B - \mu_A \times \mu_B$
NON A	$1 - \mu_A$	$1 - \mu_A$

- μ désigne la fonction d'appartenance.
- Les figures (II.22) et (II.23) présente graphiquement les opérateurs ET, OU et NON

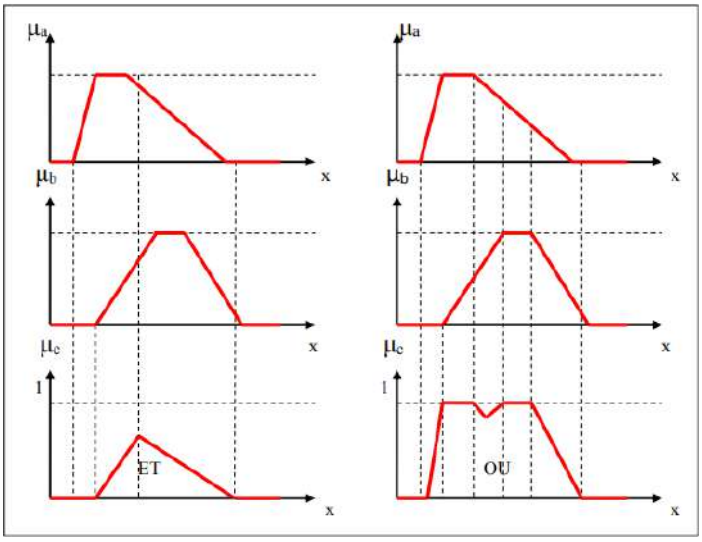


Figure II.22 : Opérateurs ET, OU

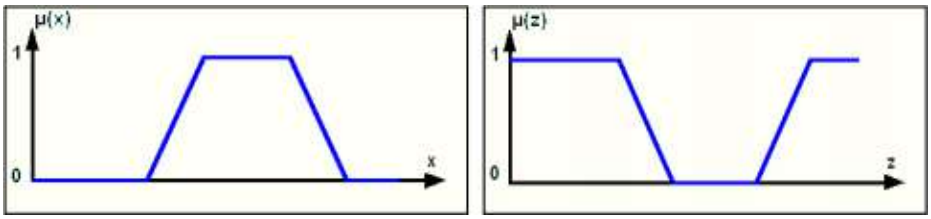


Figure II.23 : Opérateur NON

II.6.4.Structure d'une commande par logique floue

On peut distinguer trois parties principales constituant la structure d'un régulateur floue :

- La fuzzification.
- Un mécanisme d'inférence.
- La Defuzzification.

La figure (II.27) 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.

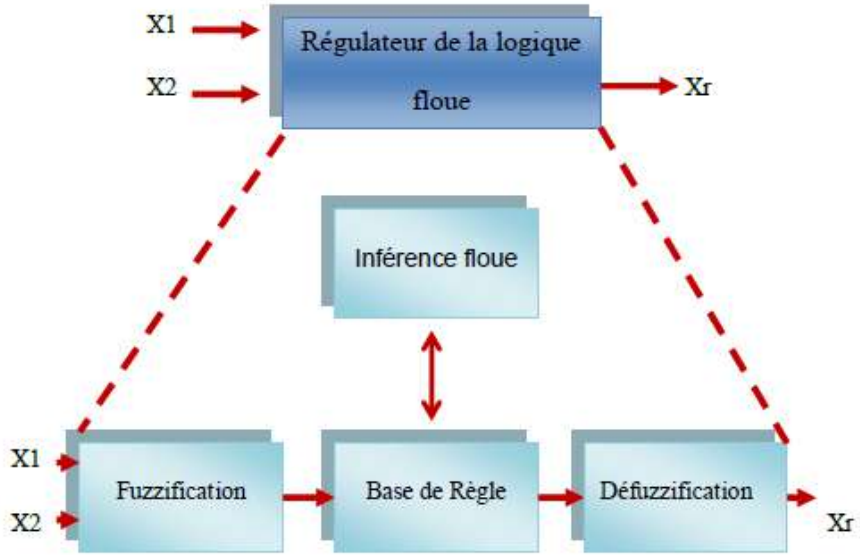


Figure II.27 : Structure d'un régulateur logique floue

❖ **Fuzzification**

Cette première étape consiste à transformer les variables (d'entrée et de sortie) en variables linguistiques :

- Pour chaque variable, on définit dans un premier temps l'univers du discours (i.e. la plage de valeurs que peut prendre la variable).
- La variable est ensuite découpée en catégories appelées variables linguistiques
- Une fonction (allant de 0% à 100%) permettant de définir pour chaque variable son degré d'appartenance (pourcentage de véracité) : « l'observation est dans telle catégorie » est affectée à chaque catégorie.

Cette étape est principalement réalisée sur la base d'observations statistiques (ou par apprentissage, supervisé ou non, pour regrouper les valeurs d'une variable en catégories homogènes) ou à dire d'expert. Les graphiques (figure II.28) sont tracés en prenant l'exemple de la taille.

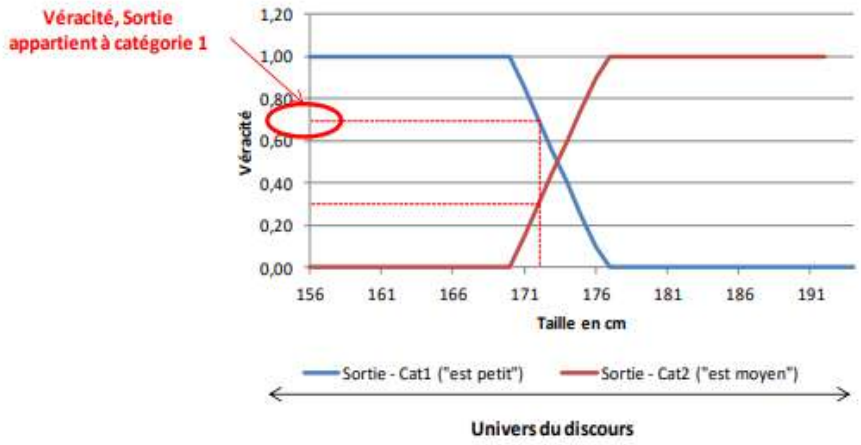


Figure II.28 : Degré d'appartenance d'un variable dans chaque catégories

- **Univers du discours :** taille en cm comprise entre 156cm et 194cm.
- **Variable linguistique :** nom de la variable de sortie (par exemple la taille).
- **Valeurs linguistiques :** « Cat1 » (petit), « Cat2 » (moyen).

La variable en sortie a une véracité de 70% pour la catégorie 1 et une véracité de 30% pour la catégorie2.

❖ **Inférence floue**

a. Construction d'un ensemble de règle

Sur la base des catégories précédemment réalisées, un ensemble de règles sont construites.

Une véracité pour chacune des règles est alors calculée. La construction de ces règles, principalement basée sur des « ET », des « OU » et des « SANS », se traduit mathématiquement de la sorte[25].

b. Mécanisme d'Inférence floue

Le bloc inférence est le cœur de la logique floue, 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 la sortie linguistique ou floue du régulateur se fait par différentes méthodes, nous nous nous intéressons à la méthode de Mamdani on cite principalement[25] :

- **Méthode d'inférence max-min (méthode de Mamdani)**

  La méthode d'inférence max-min est réalisée, au niveau de la condition l'opérateur « ET » par la formulation du minimum. La conclusion dans chaque règle, introduite par « ALORS », 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. Enfin l'opérateur «OU» qui lie les différentes règles est réalisé par la formation du maximum[25].

❖ **Défuzzification**

Les méthodes d'inférences fournissent une fonction d'appartenance résultante μres (Xr) pour la variable de sortie. 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[25]. Cette transformation est appelée défuzzification. Les méthodes de défuzzification les plus couramment utilisées sont :

- **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 μres (Xr).Dans ce contexte, il suffit de calculer l'abscisse (Xr) (la figure II.29)[25].

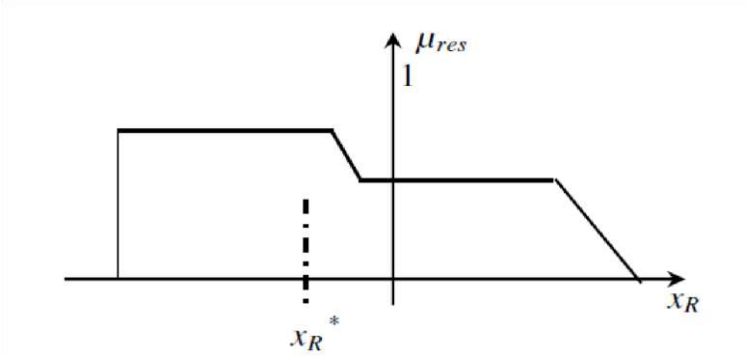


Figure II.29 : Défuzzification par centre de gravité

L'abscisse de centre de gravité peut être déterminée à l'aide de la relation générale :

$$x_R^* = \frac{\int_{-1}^{1} x_R . \mu_{res} . (x_R) . dx_R}{\int_{-1}^{1} \mu_{res} . (x_R) . dx_R} \qquad (2.6)$$

- **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 (voir la figure II.30). Comme signal de sortie xR*, on choisit l'abscisse de la valeur maximale de la fonction d'appartenance résultante μres (xR)[25].

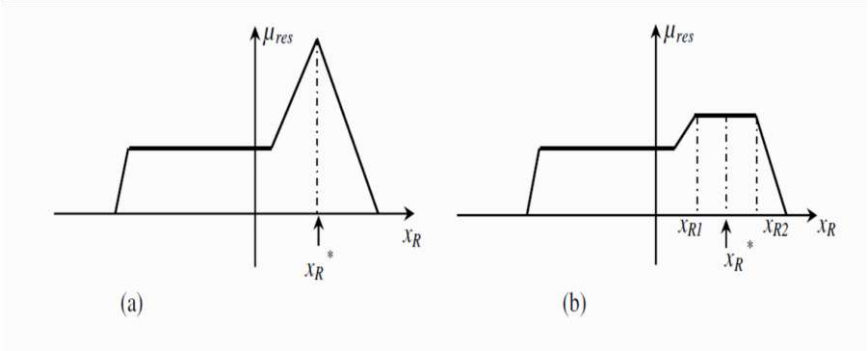


Figure II.30 : Défuzzification par valeur maximum

Lorsque μ_{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 (la figure II.10.b).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[25].

II.7.Conclusion

Dans le deuxième chapitre, nous avons débuté par un aperçu sur l'internet des objets, en présentant ses domaines d'application, ses technologies appliquées et son architecture. Nous avons également donnée une description rapide de la carte Raspberry et la carte Arduino et enfin nous avons présenté l'outil de programmation Node Red et terminé par la théorie de la logique floue.

Chapitre III :

Réalisation pratique

III.1.Introduction

Dans ce chapitre, nous présenterons l'implémentation des solutions matérielles et logicielles, le but de ce projet est de réaliser une interface homme-machine pour la supervision et le contrôle des feux de circulation à base de l'internet des objets.

Nous allons expliquer dans ce chapitre le cahier de charge, la méthode de communication entre les composants, on présentera aussi notre interface homme-machine.

III.2.Cahier de charge

Le but de ce projet est de réalisation d'une plateforme de gestion de transport à base de l'internet des objets, nous voulons contrôler une intersection en utilisant un contrôleur flou, ce type de contrôle programme dynamiquement le temps de feux vert qui varie en fonction des paramètres de l'intersection (nombre de véhicules et le débit de circulation pour chaque phase).

Les étapes de fonctionnement de notre projet sont comme suite :

III.2.1.Collecte de données

Cette opération consiste de l'agrégation des données des paramètres de l'intersection. Dans notre cas, nous avons créé une base de données contient les valeurs de nombre de véhicule et de débit de chaque phase. Ces données transmettent à la station de base pour le traitement.

III.2.2.Traitement et affichage des données

Dans cette étape, la présentation des données est importante pour mettre à jour l'IHM avec l'état de l'intersection à travers une interface graphique simple à utiliser à un accès local ou à distance. Le traitement des données collectées est effectué par un contrôleur intelligent afin d'exploiter le résultat pour le contrôle de l'intersection.

III.2.3.Action

Dans cette étape, l'action consiste à contrôler les d'intersection. De plus le fonctionnement des feux de circulation doit être affiché sur l'interface de supervision en temps réel.

III.3.Solutions hardware et software

Les solutions matérielles et logicielles utilisées dans notre projet sont introduites comme suite :

- **Solution matérielles**
  - Carte Raspberry pi 3 B+.
  - Carte Arduino.
  - Module nRF24L01.
- **Solutions logicielles**
  - Outil de développement « Node Red » basé sur le langage «JavaScript ».
  - Langage « C » pour programmer la carte Arduino.
  - Langage « Python ».

III.4.Communication entre les solutions matérielles

La figure (III.1) illustre la communication entre les matériels et les méthodes d'accès à l'interface de notre système.

Notre système est constitué de deux cartes Arduino (maitre et esclave) et une station de base (carte Raspberry). La communication entre le maitre et l'esclave est effectuée à travers le module nRF24L01 (communication par radio fréquence), Nous avons utilisé le port USB de Raspberry pour effectuer une communication série avec l'Arduino maitre.

L'objectif de ce montage est de communiquer le Raspberry avec l'Arduino l'esclave en passant par l'Arduino maitre car la communication par radio fréquence entre le Raspberry et l'Arduino n'est pas disponible dans le Node Red. Donc l'Arduino maitre est un composant intermédiaire entre l'esclave et la station de base.

Nous pouvons accéder à l'interface créée par Node Red en utilisant un réseau local ou à distance à travers le Cloud.

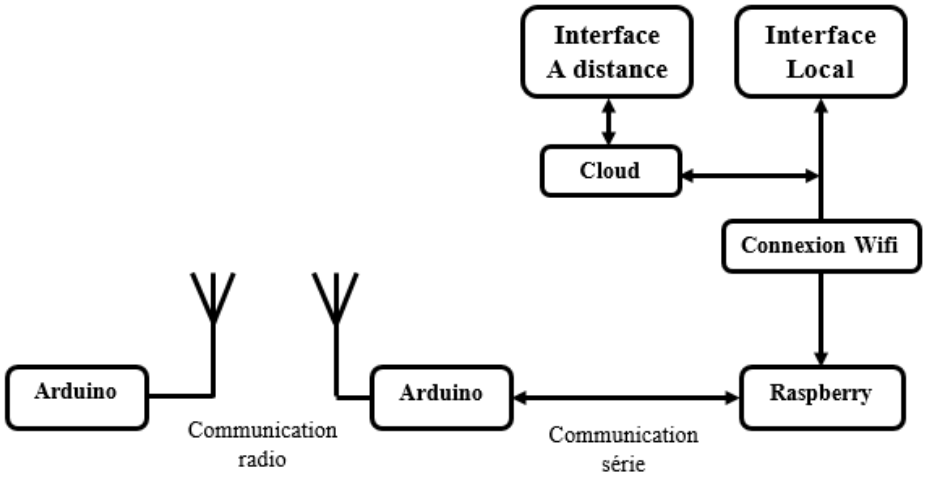


Figure III.1 : Communication entre les composants électroniques.

III.5. Contrôleur flou

L'algorithme de la logique floue permet de calculer le temps des feux verts pour chaque phase, en fonction de deux variables d'entrée : nombre des véhicules et le débit de circulation pour chaque phase (la figure III.2).



Figure III.2 : Structure du contrôleur par la logique floue

III.5.1.Fonction d'appartenance

Avant de définir les fonctions d'appartenance des entrées et la sortie, nous allons définir l'univers de discours de chaque variable :

- Entrée 01 (Nombre des véhicules) : Cinq classes.
- Entrée 02 (Débit (véh/min)) : Cinq classes.
- Sortie (Temps) : Cinq classes.

Les figures (III.3) et (III.4) illustrent les fonctions d'appartenance des variables d'entrée.

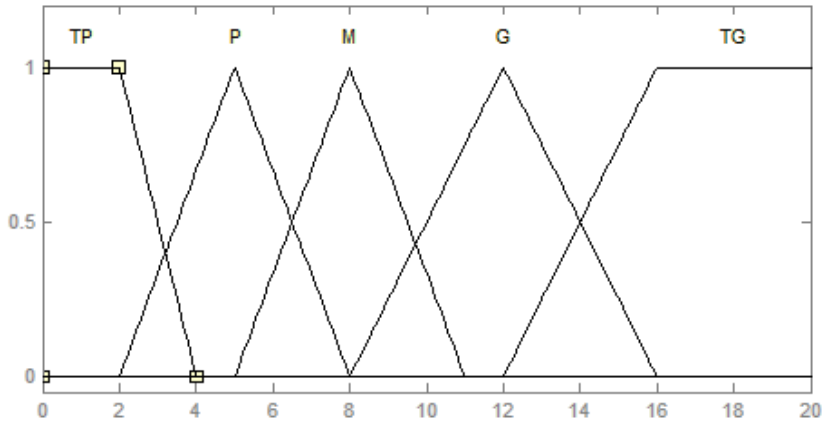


Figure III.3 : Présentation de nombre de véhicule

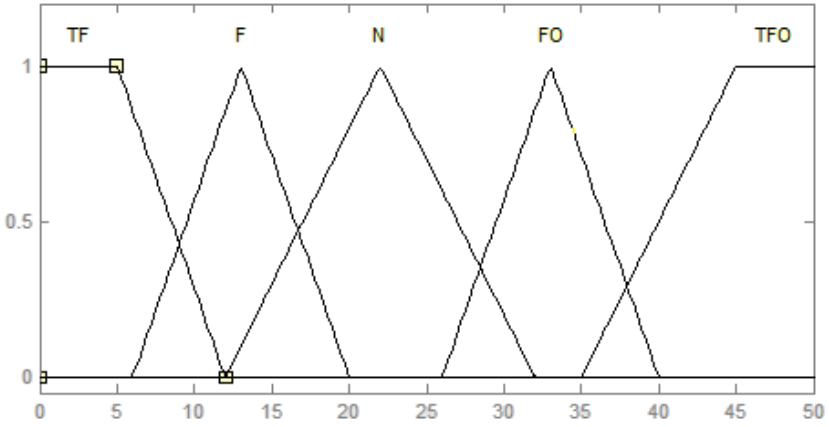


Figure III.4 : Présentation de débit

La figure (III.5) présente les fonctions d'appartenance de la variable de sortie (Temps)

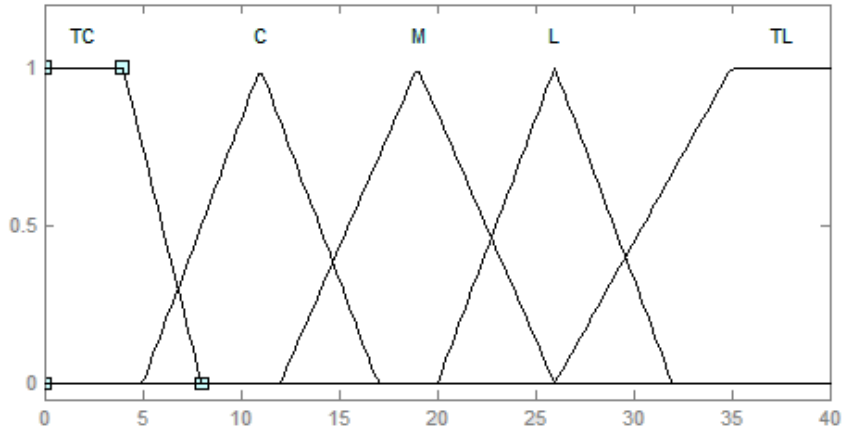


Figure III.5 : Présentation de variable de sortie (Temps)

III.5.2.Variables linguistiques

- **Entrée 01 : Nombre de véhicule**

  L'intervalle de nombre de véhicule est borné entre 0 et 20. Les variables linguistiques sont : {TP, P, M, G, TG}

  - **TP :** Très petit.
  - **P :** Petit.
  - **M :** Moyen.
  - **G :** Grand.
  - **TG :** Très grand.

- **Entrée 02 : Débit**

  L'intervalle de débit est borné entre 0 et 50. Les variables linguistiques sont :

  {TF, F, M, FO, TFO}

  - **TF :** Très faible.
  - **F :** faible.
  - **M :** Moyen.
  - **FO :** Fort.
  - **TFO :** Très fort.

- **Sortie : Temps**

  Intervalle : [0 40].

  Variables linguistiques : {TC, C, M, L, TL}

  - **TC :** Très court.
  - **C :** Court.
  - **M :** Moyen.
  - **L :** Long.
  - **TL :** Très long.

III.5.3.Règles floues

Notre contrôleur flou fonctionne selon les règles dans le tableau III.1.

Tableau III.1 : Règles floues

		Débit (véh/min)				
		TF	F	N	FO	TFO
Nombre des véhicules	TP	TC	TC	TC	C	M
	P	TC	TC	C	C	M
	M	TC	C	M	L	TL
	G	C	M	L	L	TL
	TG	M	L	L	TL	TL

III.5.4.Algorithme du contrôleur flou

Dans notre travaille, la structure du contrôleur a été écrit langage python. On utilise dans ce programme la méthode d'inférence Min-Max (Méthode de Mamdani). La figure (III.6) présente l'organigramme de cet algorithme.

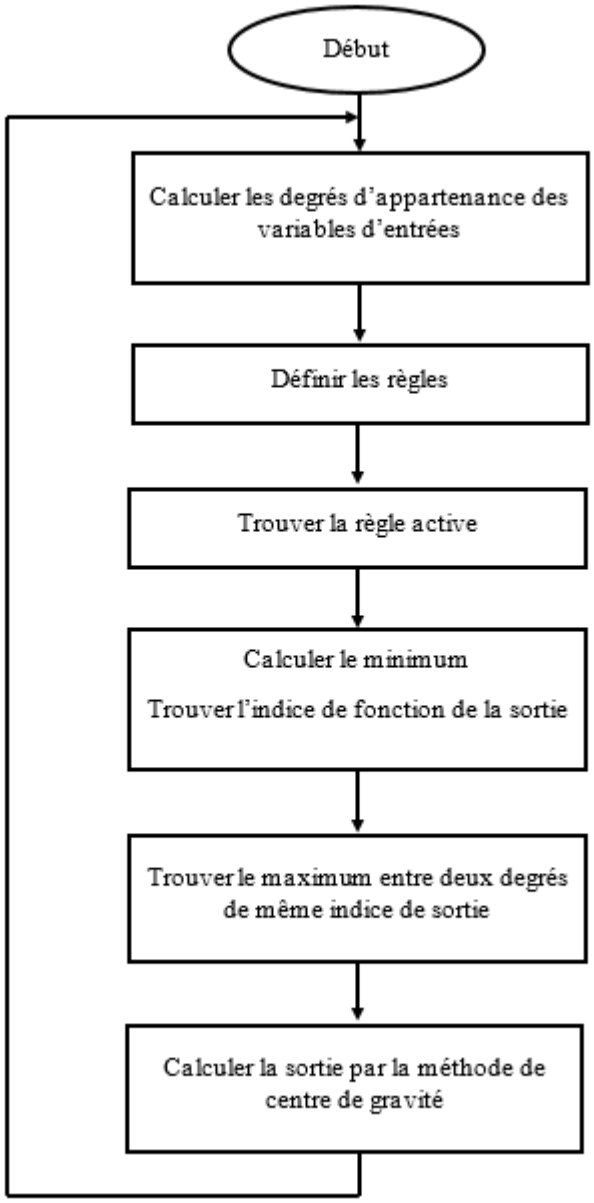


Figure III.6 : Organigramme de l'algorithme du contrôleur flou

- Les fonctions d'appartenance sont définies par des équations mathématiques comme suit :

$$Tri(x) = \begin{cases} 0 \quad pour\ x \leq a \\ \frac{x-a}{b-a} \quad pour \quad a \leq x \leq b \\ \frac{c-x}{c-b} \quad pour \quad b \leq x \leq c \\ 0 \quad pour \quad c \leq x \end{cases} \tag{3.1}$$

$$Tra(x) = \begin{cases} 0 \quad pour\ x \leq a \\ \frac{x-a}{b-a} \quad pour \quad a \leq x \leq b \\ 1 \quad pour \quad b \leq x \leq c \\ \frac{d-x}{d-c} \quad pour \quad c \leq x \leq d \\ 0 \quad pour \quad d \leq x \end{cases} \tag{3.2}$$

- Dans le programme, les variables linguistiques sont remplacées par des indices, par exemple : le variable linguistique « très petit » ou « TP » prend l'indice « 0 » et le variable linguistique « Moyen » ou « M » prend l'indice « 2 ».
- Le calcul des degrés d'appartenance (μ(x) et μ(y)) s'effectue pour chaque variable linguistique et on ajoute les résultats de chaque variable d'entrée dans un vecteur de [1x5] (on travaille avec deux variables d'entrée et 5 variables linguistiques pour chaque entrée).
- On construit une matrice de [5x5] en utilisant les vecteurs résultants de calcul de degrés d'appartenance.
- On stocker les indices des non nuls de la matrice résultante.
- On définit les règles utilisées dans une matrice.
- On compare la matrice de non nuls avec la matrice des règles définit pour trouver les positions des règles actives.
- On calcule le minimum entre les deux degrés d'une règle active et trouve l'indice de fonction de la sortie.
- On détermine le maximum entre les degrés qui ont le même indice.
- La sortie est calculée par la méthode de centre de gravité.

III.6. Communication avec le Module nRF24L01

III.6.1. Présentation

Le module nRF24L01 est un émetteur / récepteur développé par la société Nordic Semi-Conductor (figure III.7) qui utilise la bande de fréquence 2.4 GHz. Il fonctionne avec une tension d'alimentation entre 1.9V et 3.6V, mais ses signaux sont tolérants au 5 V et il peut donc être utilisé avec un Arduino Uno par exemple. Le débit maximum du nRF24L01 est de 2Mbps, mais à cette vitesse la portée n'est que de quelques mètres.

Ce module peut communiquer avec six autres modules. En utilisant cette fonctionnalité, nous pouvons l'utiliser dans les réseaux maillés et d'autres applications réseau.



Figure III.7 : modules nRF24L01

III.6.2.Liaison spi

La liaison SPI (Serial Peripheral Interface) est une liaison série synchrone (bidirectionnelle) qui opère en mode full- duplex. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un même bus, dans ce cas, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée Slave Select (SS) (Voir la figure III.8). Le module nRF24L01 peut être considéré comme un bus SPI dont les connexions ne sont pas filaires mais hertziennes.

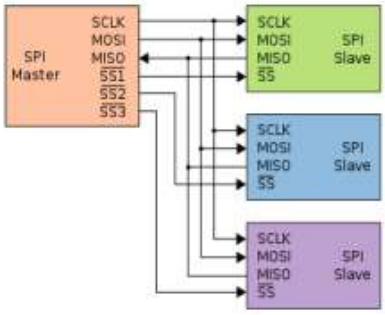


Figure III.8 : Communication maitre/esclave

III.6.3.Pins de module nRF24L01

La figure (III.9) montre les broches de module nRF24L01



Figure III.9 : Pins de module nRF24L01 et leurs fonctions

- GND est la broche de terre. Il est généralement marqué en entourant la broche dans un carré afin de pouvoir être utilisé comme référence pour identifier les autres broches.
- VCC fournit l'alimentation au module. Cela peut être n'importe où de 1,9 à 3,6 volts.
- CE (Chip Enable) est une broche active-HAUTE. Lorsque sélectionné, le nRF24L01 transmettra ou recevra, selon le mode dans lequel il se trouve actuellement.
- CSN (Chip Select Not) est une broche active-LOW et est normalement maintenue à l'état HIGH. Lorsque cette broche devient faible, le nRF24L01 commence à écouter les données sur son port SPI et les traite en conséquence.
- SCK (Serial Clock) accepte les impulsions d'horloge fournies par le bus maître SPI.
- MOSI (Master Out Slave In) est l'entrée SPI du nRF24L01.
- MISO (Master In Slave Out) est la sortie SPI du nRF24L01.
- IRQ est une broche d'interruption pouvant alerter le maître lorsque de nouvelles données sont disponibles pour traitement.

III.6.4.Principe de fonctionnement

- **Fréquence du canal**

  Le module émetteur-récepteur nRF24L01 transmet et reçoit des données sur une certaine fréquence appelée canal. De plus, pour que deux ou plusieurs modules d'émetteur-récepteur puissent communiquer l'un avec l'autre, ils doivent se trouver sur le même canal. Ce canal peut être n'importe quelle fréquence dans la bande ISM à 2,4 GHz ou, plus précisément, entre 2,4 GHz et 2,5 GHz[26].

  Chaque canal occupe une largeur de bande inférieure à 1 MHz. Cela nous donne 125 canaux possibles avec un espacement de 1 MHz. Ainsi, le module peut utiliser 125 canaux différents, ce qui permet d'avoir un réseau de 125 modems fonctionnant de manière indépendante à un endroit (Voir la figure III.10)[26].

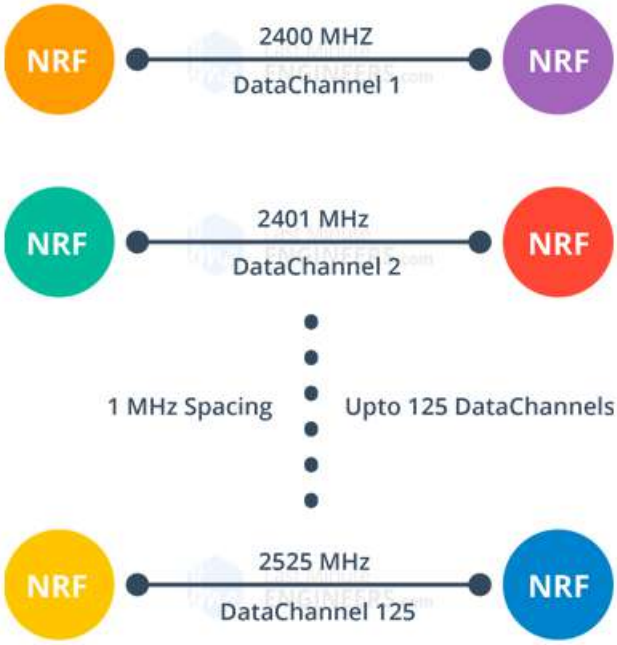


Figure III.10 : Fréquence du canal

❖ **Protocole de communication**

Le module émetteur-récepteur nRF24L01 utilise une structure de paquets appelée Enhanced ShockBurst. Cette structure de paquet simple est décomposée en 5 champs différents, illustrés dans la figure (III.11).

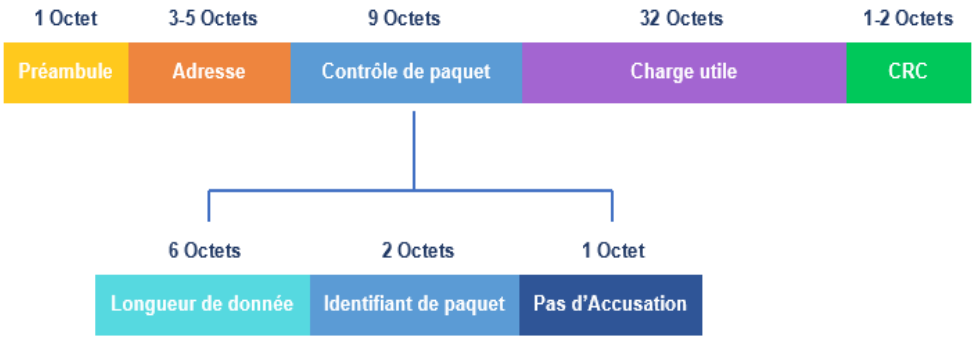


Figure III.11 : Structure des paquets ShockBurst améliorée

La structure d'origine de ShockBurst ne comprenait que les champs Préambule, Adresse, Charge utile et Contrôle de redondance cyclique (CRC). ShockBurst amélioré a apporté une fonctionnalité accrue pour des communications améliorées utilisant un champ de commande de paquets (PCF) récemment introduit. Cette nouvelle structure est excellente pour plusieurs raisons. Premièrement, il autorise les charges utiles de longueur variable avec un spécificateur de longueur de charge utile, ce qui signifie que les charges utiles peuvent varier de 1 à 32 octets[26].

Deuxièmement, il attribue à chaque paquet envoyé un identifiant de paquet, ce qui permet au périphérique récepteur de déterminer si un message est nouveau ou s'il a été retransmis (et peut donc être ignoré)[26].

Enfin et surtout, chaque message peut demander l'envoi d'un accusé de réception lorsqu'il est reçu par un autre appareil[26].

- **Transaction avec accusé de réception et interruption**

Ceci est un exemple de scénario positif. Ici, l'émetteur commence une communication en envoyant un paquet de données au récepteur. Une fois que tout le paquet est transmis, il attend (environ 130 µs) la réception du paquet d'accusé de réception (paquet ACK). Lorsque le destinataire reçoit le paquet, il envoie un paquet ACK à l'émetteur. À la réception du paquet ACK, l'émetteur émet un signal d'interruption (IRQ) pour indiquer que les nouvelles données sont disponibles[26].

- **Transaction avec paquet de données perdu**

Il s'agit d'un scénario négatif dans lequel une retransmission est nécessaire en raison de la perte du paquet transmis. Une fois le paquet transmis, l'émetteur attend la réception du paquet ACK. Si l'émetteur ne l'obtient pas dans le délai ARD (Auto-Retransmit-Delay), le paquet est retransmis. Lorsque le paquet retransmis est reçu par le récepteur, le paquet ACK est transmis, ce qui génère à son tour une interruption au niveau de l'émetteur[26].

- **Transaction avec accusé de réception perdu**

Il s'agit là encore d'un scénario négatif dans lequel une retransmission est nécessaire en raison de la perte du paquet ACK. Ici, même si le destinataire reçoit le paquet lors de la première tentative, en raison de la perte du paquet ACK, l'émetteur pense que le destinataire n'a pas du tout reçu le paquet. Ainsi, une fois le délai de retransmission automatique écoulé, il retransmet le paquet. Désormais, lorsque le destinataire reçoit le paquet contenant le même identifiant de paquet que précédemment, il le rejette et envoie à nouveau le paquet ACK[26].

L'ensemble du traitement des paquets est effectué automatiquement par le module nRF24L01 sans implication du microcontrôleur.

- **Multi-récepteur (multi-receiver)**

Le nRF24L01 fournit une fonctionnalité appelée Multireceiver (Multiple Transmitters Single Receiver). Chaque canal RF est divisé logiquement en 6 canaux de données

parallèles appelés canaux de données. En d'autres termes, un canal de données est un canal logique dans le canal RF physique. Chaque canal de données a sa propre adresse physique (adresse de canal de données) et peut être configuré.

III.7.Communication

Nous travaillons dans notre projet avec deux méthodes de communication, communication radio en utilisant le module nRF24L01 et communication avec un port série.

II.7.1.Transmission avec nRF24L01

- **Initialisation**

  Dans cette étape on va :

  - Ajouter les bibliothèques SPI, nRF24L01, RF24.
  - Définir des pins CSN et CE et appeler la bibliothèque « RF24 » pour créer le radio
  - Définir l'adresse de module de réception (esclave).
  - Initialiser les variables utilisés dans le programme et créer une base de données à transmettre.

- **Configuration**

  Les fonctions qu'on va utiliser dans cette étape s'effectue grâce à la bibliothèque «nRF24L01» :

  - Démarrage de communication : **radio.begin ().**
  - La vitesse de transmission des données : **radio.setDataRate ().**

    On peut travailler avec : 250Kb/s, 1Mb/s ou 2Mb/s.
  - Configuration du canal (lecture ou écriture) et spécification de l'adresse de l'esclave :

    radio.openWritingPipe (Adresse de l'esclave).

La figure (III.12) présente un exemple de transmission.

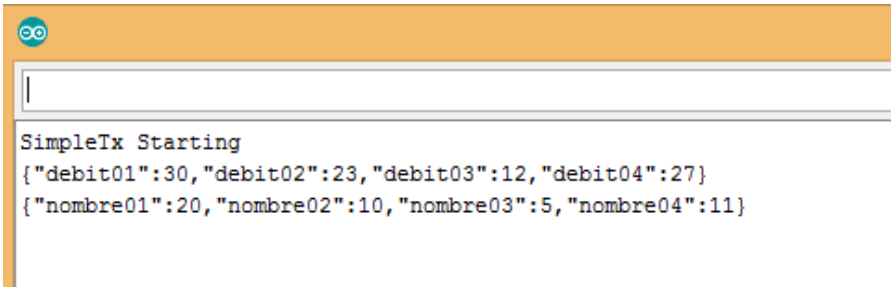


Figure III.12 : Transmission des données par Arduino

III.7.2.Réception avec nRF24L01

Dans le module de réception, nous appliquons les mêmes paramètres d'initialisation et de configuration du module de transmission sauf qu'on change le mode vers le mode lecture et préciser le canal en utilisant les deux fonctions suivantes :

- **radio.openReadingPipe (1, thisSlaveAddress)**
- **radio.startListening ()**

La réception des données est montrée dans la figure (III.13).

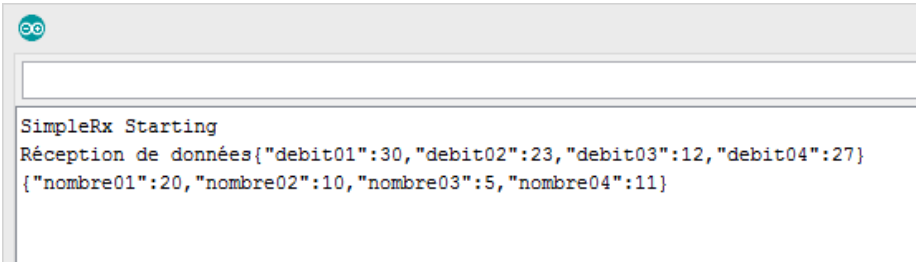


Figure III.13 : Réception des données par l'Arduino

III.7.3. Communication série

On utilise dans communication série les fonctions de la bibliothèque « SPI » pour le transfert ou la réception, nous suivons les étapes suivantes :

- Configuration du port série (démarrage et débit) : **Serial.begin (débit)**
- Nous utilisant la fonction « Serial.print » pour écrire dans le port série et la fonction « Serial.read » pour lire le port série.

III.8. Fonctionnement du système global

Nous présentons dans la figure (III.14) la mise en place des matériels pour créer notre système de

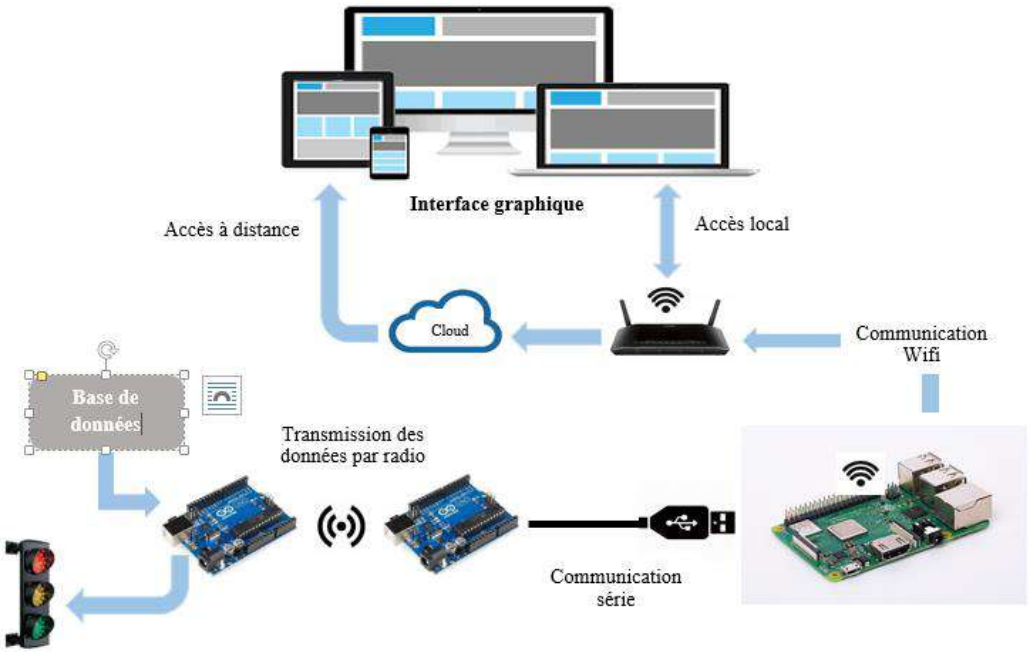


Figure III.14 : Mise en place des matériels du système.

- Un Arduino esclave exécute une tache de réception des commandes pour contrôler les feux. Ces données sont envoyées par le Raspberry à un Arduino maitre par un port série, après cette étape, l'Arduino maitre effectue une communication par radio fréquence pour transmettre les commandes à l'Arduino l'esclave. Chaque fin de cycle, notre esclave transmet les valeurs de nombre de véhicule et de débit de chaque phase à l'Arduino maitre à travers le module nRF24L01, puis le maitre envoie ces paramètres au Raspberry pour que le Node Red traite ces données.

- Lorsque les valeurs de nombre de véhicule et de débit sont disponibles sur le Node Red, le contrôleur flou exécuté dans Node Red en langage python utilise ces valeurs pour calculer le temps de feu vert de chaque phase. Ensuite, nous devons afficher les données (nombre de véhicule, débit et le temps calculé) dans notre interface.

- Chaque phase, Node Red envoie des valeurs à travers le port série pour que l'Arduino maitre transmet ces valeurs à l'Arduino l'esclave via une communication par radio fréquence. Ces valeurs sont utilisées pour contrôler les feux qui sont connectées avec notre esclave.

III.9. Interface de supervision

Dans notre projet, l'interface est application web accessible depuis un navigateur, Node Red est une solution très intéressante car ce dernier fait le lien entre les composants utilisés.

Grace à Node Red, nous pouvons construire des applications adaptées à l'internet des objets, on trouve un ensemble des blocs prédéfinit, configurable ce qui offre une programmation visuelle simple. La combinaison des nœuds connectées (nœuds d'entrée, nœuds de traitement et nœuds de sortie) forme un flux (flow en anglais). L'interface de supervision est créée par l'utilisation des nœuds de tableau du bord (Node Red Dashboard), nous pouvons afficher les données collectées grâce aux éléments graphiques de Node Red.

III.10.Demarrage de Node Red sur Raspberry

L'accès à l'interface de programmation se fait à partir d'un navigateur. Il est préférable d'utiliser le navigateur d'un ordinateur connecté sur le même réseau que Raspberry. 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.15).

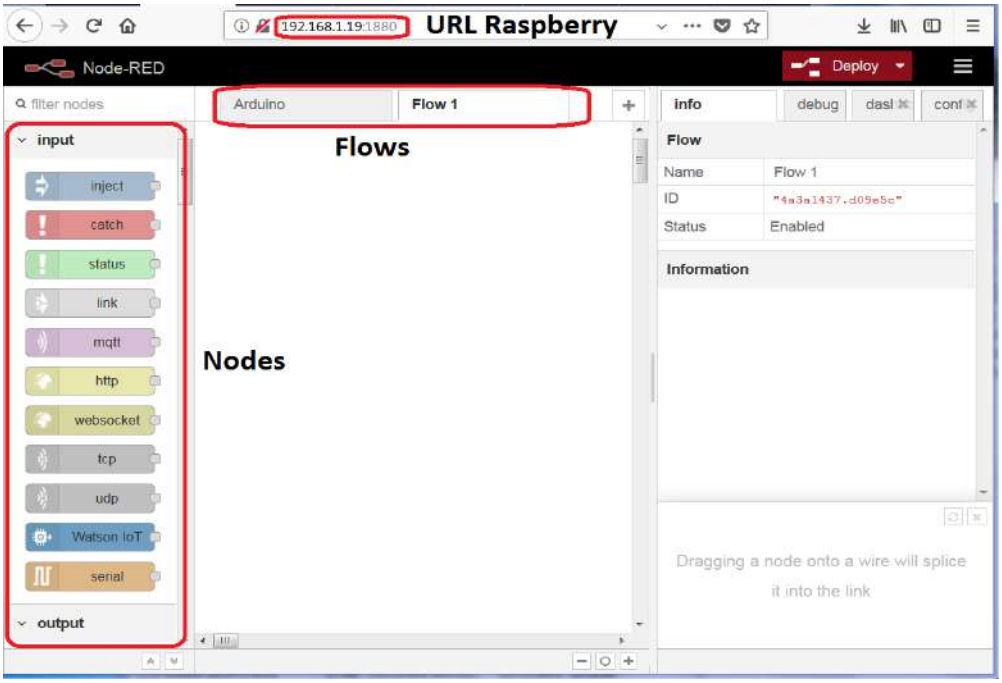


Figure III.15 : Environnement de programmation Node Red

Les nœuds (nodes) de base sont déjà bien fournis dans Node Red, mais on doit ajouter des nœuds de tableau de bord (Dashboard) pour créer l'interface homme-machine. Avant tout, on besoin d'installer le gestionnaire des nœuds (voir la figure III.16).

```
$ node-red-stop
$ sudo apt-get update
$ sudo apt-get install npm
$ cd /home/pi/.node-red
$ npm rebuild
$ node-red-start
```

Figure III.16 : Installation de gestionnaire des palettes

Après l'exécutions des commandes précédentes, nous pouvons installe des nouveaux nœuds, la figure (III.17) présente comment installer des nœuds.

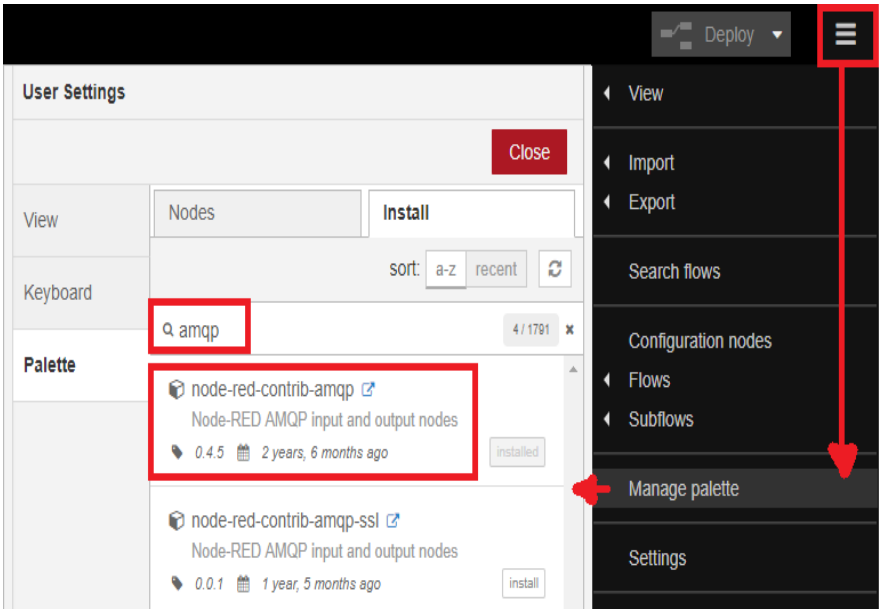


Figure III.17 : Installation des nouvelles palettes

III.11.Programmation avec Node Red

Notre flux est constitué de trois blocs principaux, un bloc reçoit les données envoyées via le port série, un bloc qui traite ces données en utilisant un nœud pour exécuter un programme développé sous Python et un bloc qui contrôle les feux de circulation (voir la figure III.18).

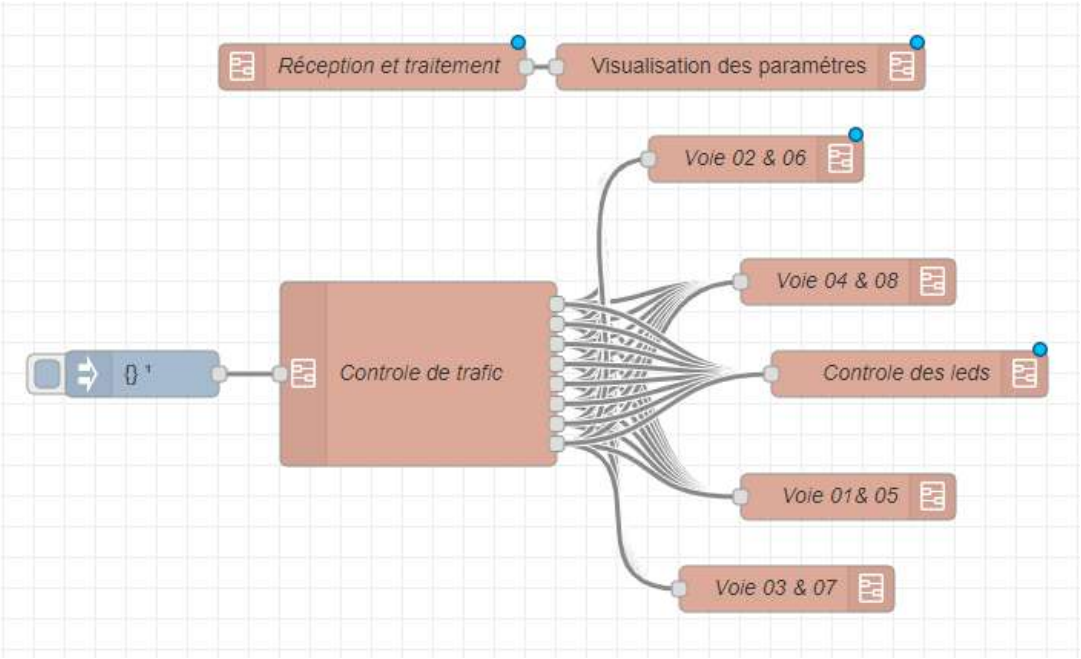


Figure III.18 : Flux de système de supervision et contrôle

III.11.1. Réception des données

La réception des données envoyées par l'Arduino maitre au Raspberry (débit de phase et nombre de véhicule) s'effectue par le port série, pour que le Node Red utilise le port USB, nous devons ajouter une palette « Serial » (voir la figure III.19)

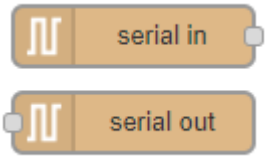


Figure III.19 : Palette Serial

Pour recevoir ou envoyer via un port USB local, nous cliquons sur le nœud pour configurer les paramètres de port série (spécifier le COM, le débit de transmission...etc.) (Voir la figure III.20).

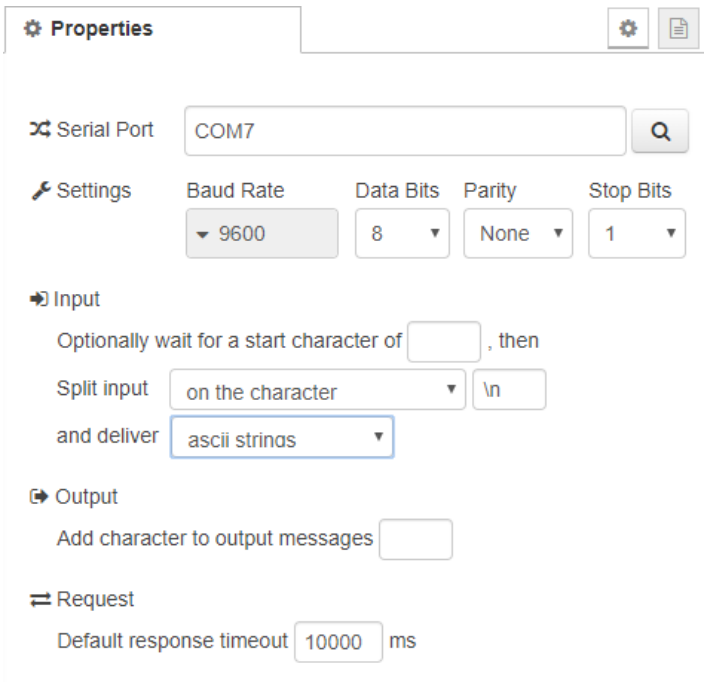


Figure III.20 : Configuration du nœud Serial

Le flux de réception des données est montré dans la figure (III.21)

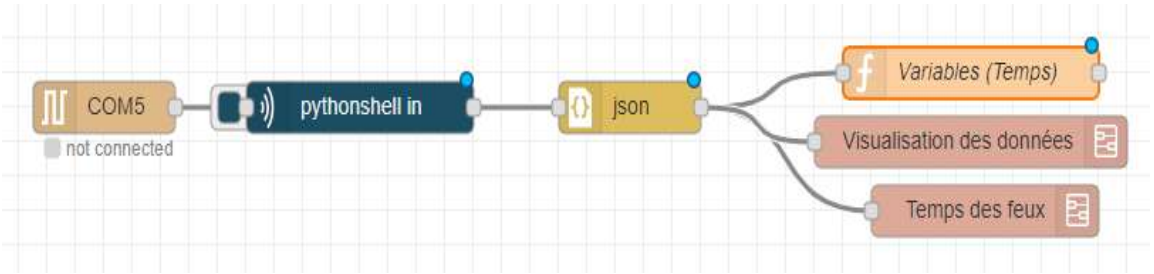


Figure III.21 : Flux de réception et affichage des données

III.11.3. Traitement des données

Les données reçus par Node Red à travers le port série sont traitées par le contrôleur flou écrit en langage « Python », les variables d'entrées du programme sont les valeurs de nombre de véhicule et le débit le temps de feu vert nécessaire pour chaque phase (en seconde)

Node Red fournit un nœud (ou palette) « Pythonshell », il est capable d'exécuter un script python à partir de notre environnement de programmation. L'entrée au nœud deviendra un argument du script python, la sortie du script sera envoyée à la sortie du nœud. Il faut juste spécifier le chemin d'accès au dossier de l'environnement virtuel dans la configuration du nœud.

La sortie de Pythonshell est un vecteur contient les paramètres de l'intersection et le temps du feu vert de chaque phase, ces données doit être visualisées dans notre interface, mais avant ça, on doit les séparer. Nous utilisons dans cette étape un nœud appelé « JSON ».

Ce nœud capable de convertir une chaîne JSON vers un objet JavaScript et l'inverse (les données en chaine JSON ont le type chaines de caractères « string »). La configuration de ce nœud est simple, pour le choix du mode de conversion, on clique sur le nœud (voir la figure III.22).

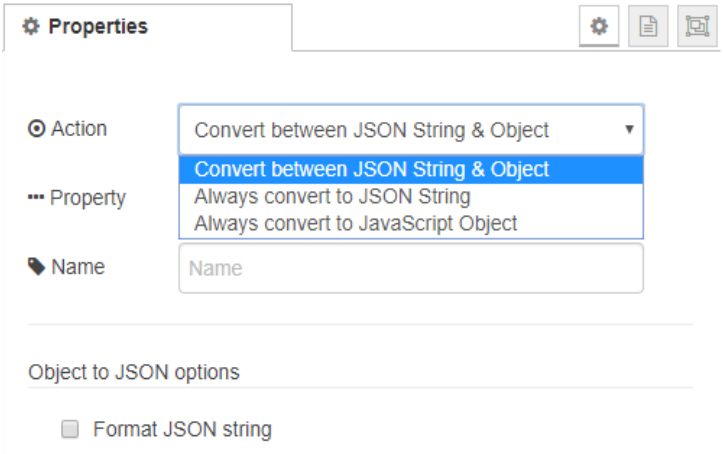


Figure III.22 : Configuration du nœud JSON

Nous faisons cette opération lorsque nous recevons des données car les données d'une chaine JSON ne sont pas extractibles (Les données sont une chaîne de caractère et sont entourées de guillemets et ne sont pas exploitables). Donc la conversion en JSON vers un objet JavaScript permet de séparer les données et avoir la possibilité de les exploiter, la figure (III.23) présente la différence entre les données reçus sans l'utilisation du nœud JSON et lorsqu'on ajoute le nœud JSON.



Figure III.23 : Différence entre la forme JSON et la forme objet JavaScript

III.11.2.Visualisation des données

Une fois que nous avons converti les données en chaine JSON vers un objet JavaScript, nous pouvons extraire des valeurs individuelles et séparées. Nous devons utiliser dans cette étape le nœud « Function » pour extraire et stocker les valeurs, la méthode d'extraction des valeurs est illustrée dans la figure (la figureIII.24).

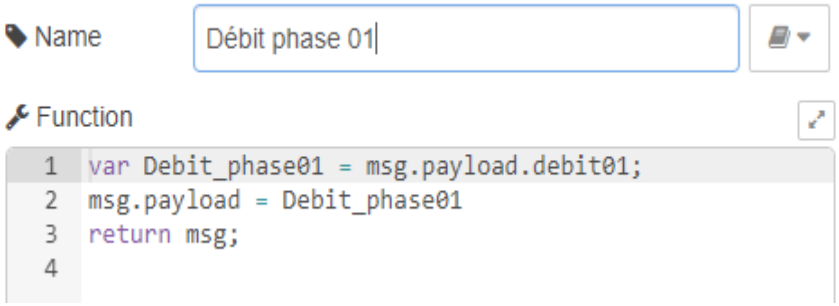


Figure III.24 : Extraction d'un variable d'un objet

Après l'extraction, nous pouvons afficher la valeur de débit ou nombre des véhicules dans notre interface en utilisant le nœud du tableau de bord « jauge » pour indiquer le débit des véhicules, la configuration des paramètres (Groupe, nom, la taille de la jauge, type de jauge, plage des valeurs) sont illustrées dans la figure (voir la figure III.25).

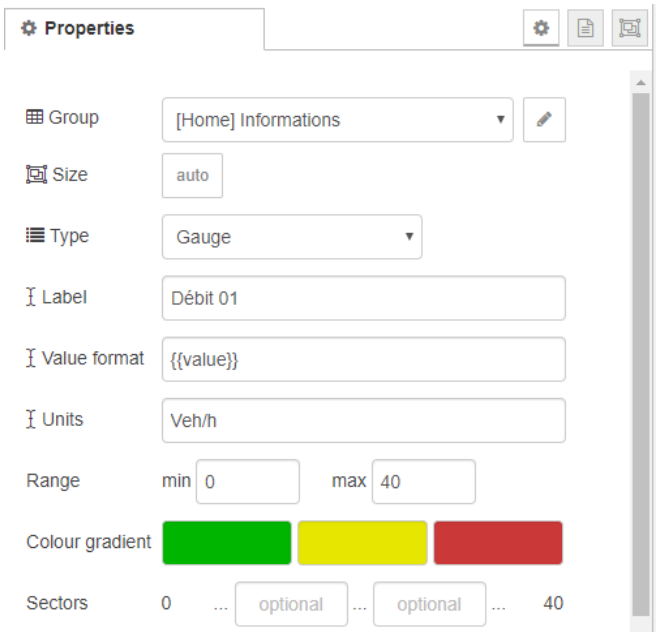


Figure III.25 : Configuration de jauge

Le flux qui visualise les données est montré dans la figure (III.26)

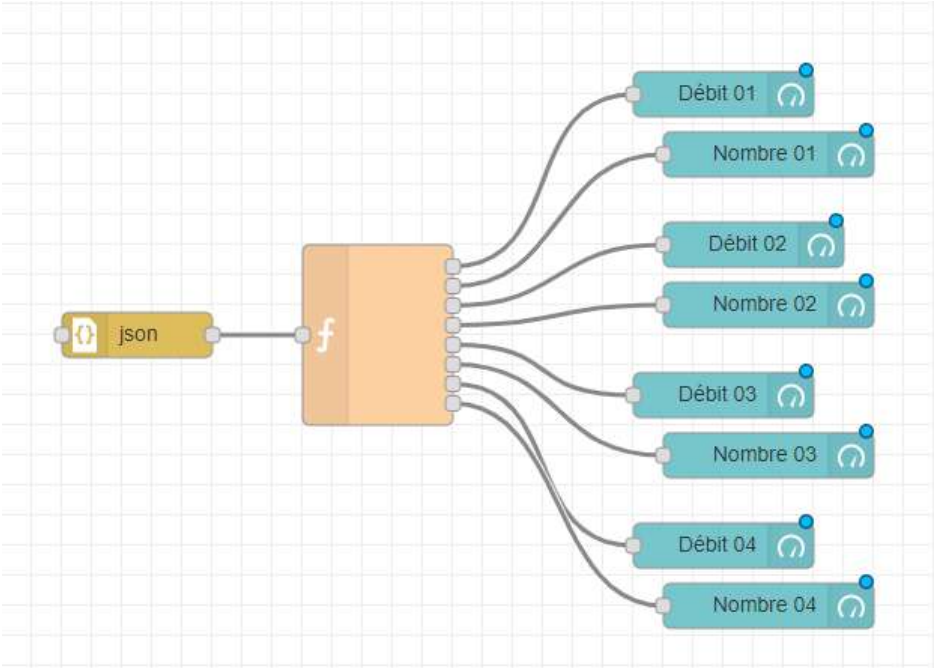


Figure III.26 : Flux de réception des données

Les données sont affichées dans l'interface comme suite (la figure III.27)



Figure III.27 : Visualisation des données dans l'interface graphique

III.11.4. Contrôle des feux de circulation

Pour contrôler les feux de circulation et gérer les séquences des phases, nous avons créé un flux en utilisant le nœud « Function », cette palette permet d'écrire et exécuter des fonctions en JavaScript, la figure (III.28) présente le flux de contrôle.

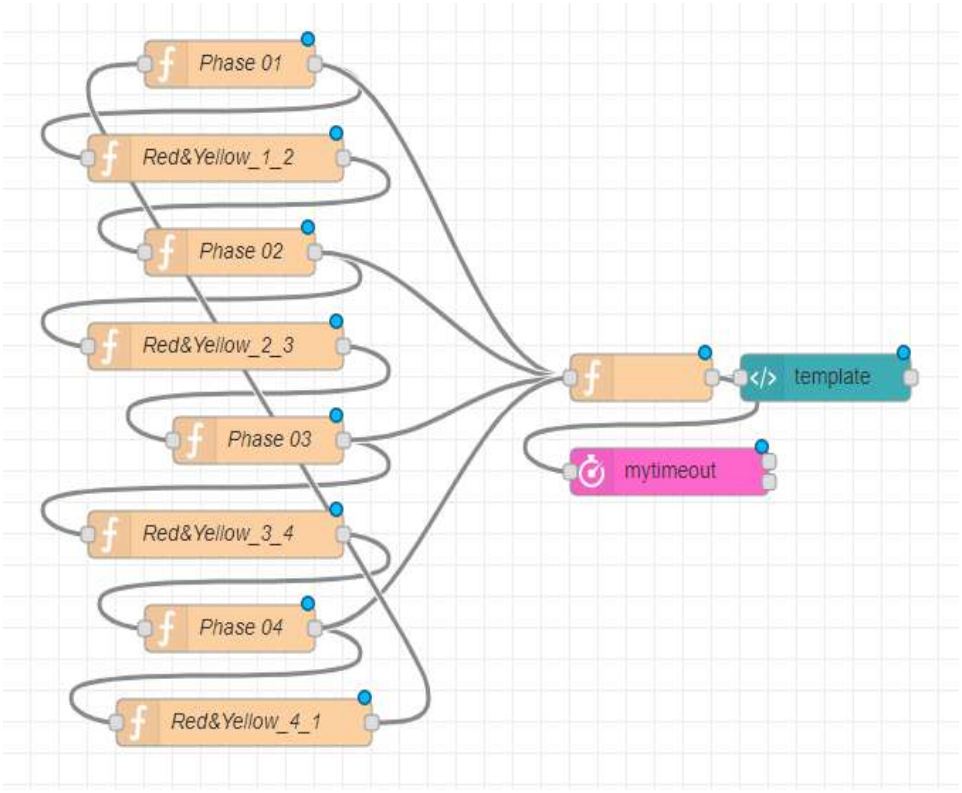


Figure III.28 : Flux de contrôle les feux de circulation

Le plan des feux est organisé selon le tableau (III.2).

Tableau III.2 : Etats des feux dans chaque phase

	Voies (02 et 06)		Voies (04 et 08)		Voies (01 et 05)		Voies (03 et 07)	
	Vert	**Rouge**	**Vert02**	**Rouge02**	**Vert03**	**Rouge03**	**Vert04**	**Rouge04**
Phase01	**Vrai**	**faux**	**Faux**	**Vrai**	**Faux**	**Vrai**	**Faux**	**Vrai**
Phase 02	**Faux**	**Vrai**	**Vrai**	**Faux**	**Faux**	**Vrai**	**Faux**	**Vrai**
Phase 03	**Faux**	**Vrai**	**Faux**	**Vrai**	**Vrai**	**Faux**	**Faux**	**Vrai**
Phase 04	**Faux**	**Vrai**	**Faux**	**Vrai**	**Faux**	**Vrai**	**Vrai**	**Faux**

Les fonctions sont exécutées en séquence dans une boucle fermée, chaque fonction envoie un message dans la sortie contient l'état des feux dans chaque phase et la donnée à envoyer dans le port série, un exemple d'une fonction présenté dans la figure (III.29)

```
function ph1(){
    msg.payload = {"com":0,"phase1":0,
                   "green1": true, "red1": false, "yellow1": false,
                   "green2": false, "red2": true, "yellow2": false,
                   "green3": false, "red3": true, "yellow3": false,
                   "green4": false, "red4": true, "yellow4": false};
    node.send(msg);
}
setTimeout(ph1,1000);
```

Figure III.29 : Etats des feux dans la phase 01

Nous pouvons afficher le temps des phases pour informer l'utilisateur. Il suffit d'extraire les valeurs de temps calculées par le contrôleur logique floue et séparées par le nœud JSON, la visualisation dans l'interface des valeurs de temps est effectuée par le nœud jauge.

Nous avons ajouté une section dans notre interface graphique pour afficher une image illustre le diagramme des phases routiers grâce au nœud « Template » (la figure III.28), à chaque phase active le nœud « function » envoie l'image des directions active au « Template ». La figure (III.30) illustre la manière d'affichage des images.

```
if (msg.payload.phase1 === true){
    msg.payload="https://pp.userapi.com/c850128/v850128775/12a7b0/i1zDJgPIS6g.jpg";
}

else if (msg.payload.phase2 === true){
    msg.payload="https://pp.userapi.com/c850128/v850128775/12a7a2/S-IhnOEaBr4.jpg";
}

else if (msg.payload.phase3 === true){
    msg.payload="https://pp.userapi.com/c850128/v850128775/12a7a9/5uiwJzWmUxI.jpg";
}

else if (msg.payload.phase4 === true){
    msg.payload="https://pp.userapi.com/c850128/v850128775/12a79b/Rb8WKRW1N0M.jpg";
}
else{
    return null;
}
return msg;
```

Figure III.30 : Fonction en JS pour afficher les images et temporiser les feux

L'affichage du temps et l'image dans notre interface est présenté dans la figure (III.31).



Figure III.31 : Visualisation des phases

III.11.5.Afficher l'état des feux

Le fonctionnement des feux de circulation doit être présenté dans notre interface graphique, nous devons extraire l'état de chaque feu envoyé par les fonctions (la figure III.28), Après nous pouvons afficher l'état des feux en utilisant le nœud « Led », nous pouvons configurer ce nœud en changent la couleur lors de l'activation et la désactivation (voir la figure III.32).



Figure III.32 : Configuration de nœud Led

Le flux de visualisation des Leds dans l'interface est montré dans figure (III.33).

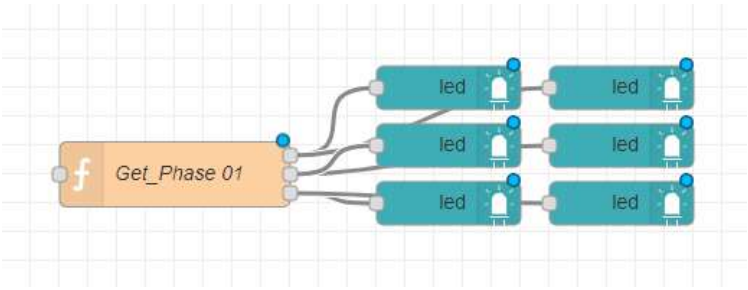


Figure III.33 : Flux d'affichage de l'état des Leds pour chaque phase

La figure (III.34) illustre la fonction d'extraction de l'état des leds

Function

```
1  var get_red1 = { payload: msg.payload.red1 };
2  var get_yellow1 = { payload: msg.payload.yellow1 };
3  var get_green1 = { payload: msg.payload.green1 };
4
5  return [ get_red1, get_yellow1, get_green1];
```

Figure III.34 : Fonction JS pour obtenir l'état des leds

Les leds sont affichées dans l'interface graphique comme suit (la figure III.35).

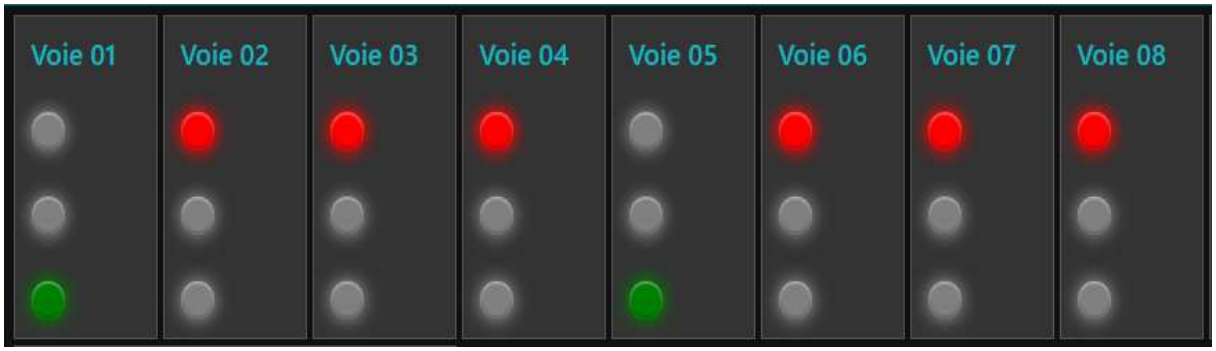


Figure III.35 : Visualisation des états des feux dans l'interface graphique

Notre interface graphique est enfin présentée dans la figure (III.36)

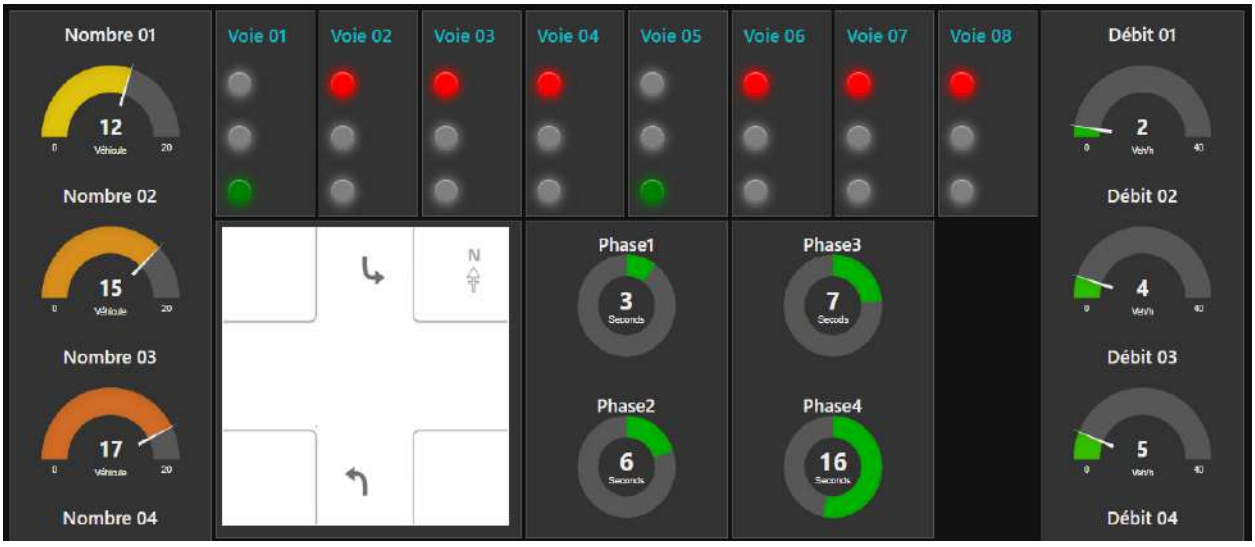


Figure III.36 : Interface de supervision

III.12.Supervision avec application de l'IoT

Dans cette étape, nous utilisons le Cloud pour accéder à distance à notre interface de supervision, nous utilisons dans notre application FRED.

FRED (La figure III.37) est un outil offre un service Cloud pour gérer Node Red à distance depuis n'importe quel réseau en utilisant une tablette ou un smart phone.



Figure III.37 : Nœud FRED

Pour envoyer des données au FRED et les visualiser dans l'interface graphique, nous utilisons le nœud de sortie « Fred out » (voir la figure III.38)

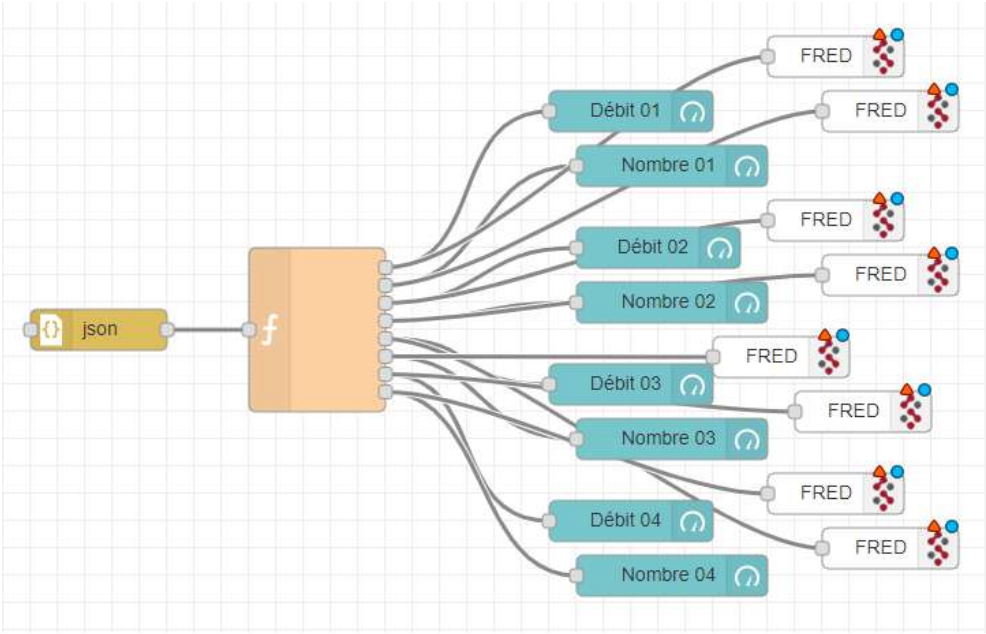


Figure III.38 : Exemple d'envoi des données au FRED

La figure (III.39) présente l'interface visualisée dans l'outil FRED

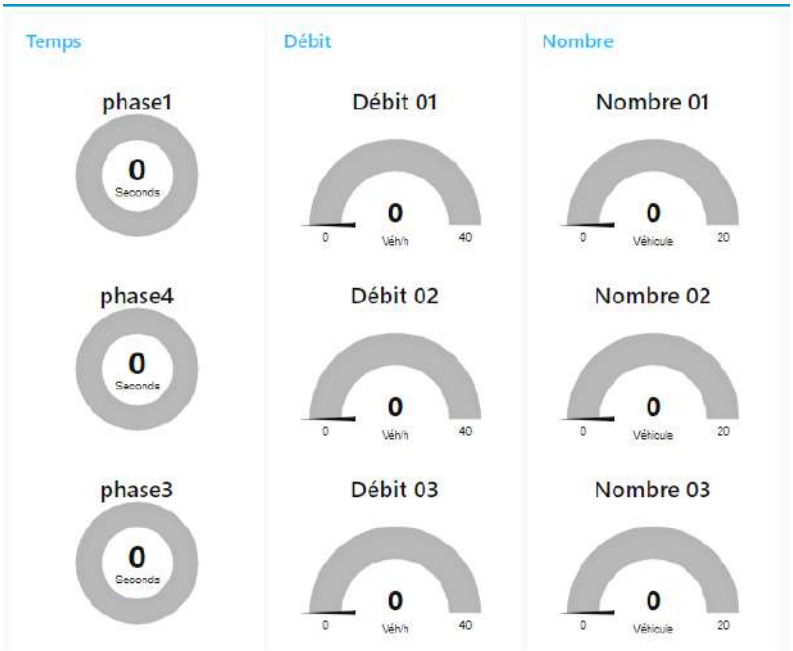


Figure III.39 : Interface graphique avec l'outil FRED

II.13. Prototype réalisé

Nous introduisons dans la figure (III.40) notre montage pratique, un Arduino maitre reçoit les données de Raspberry et les transmet vers l'Arduino esclave afin de contrôler les feux de circulation.

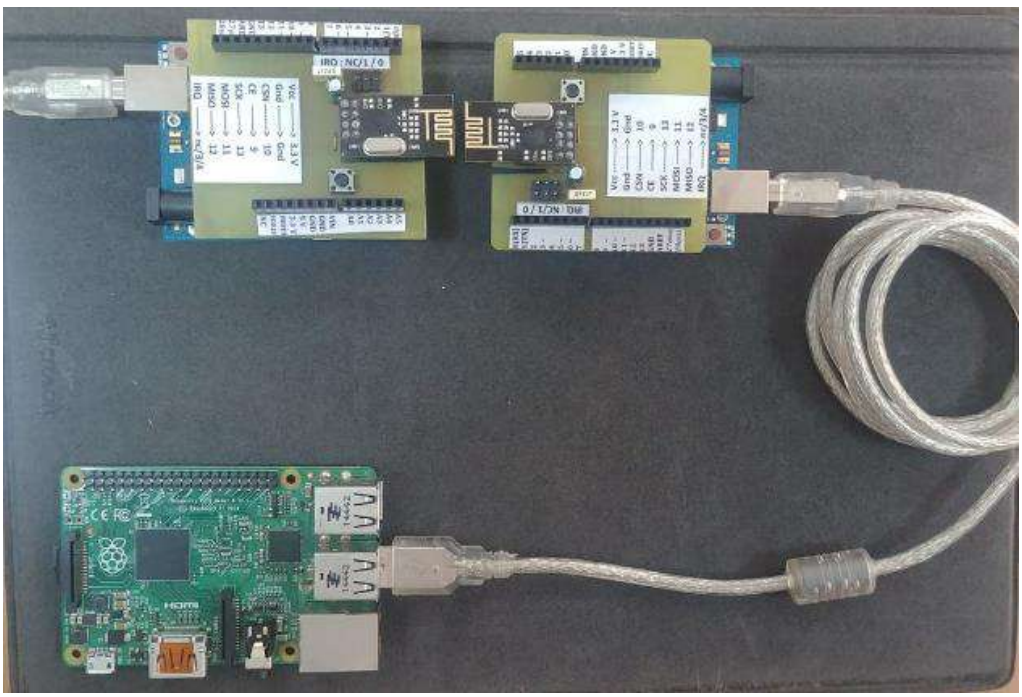


Figure III.40 : Prototype réalisé

Les feux de circulation sont introduits dans la figure (III.41)

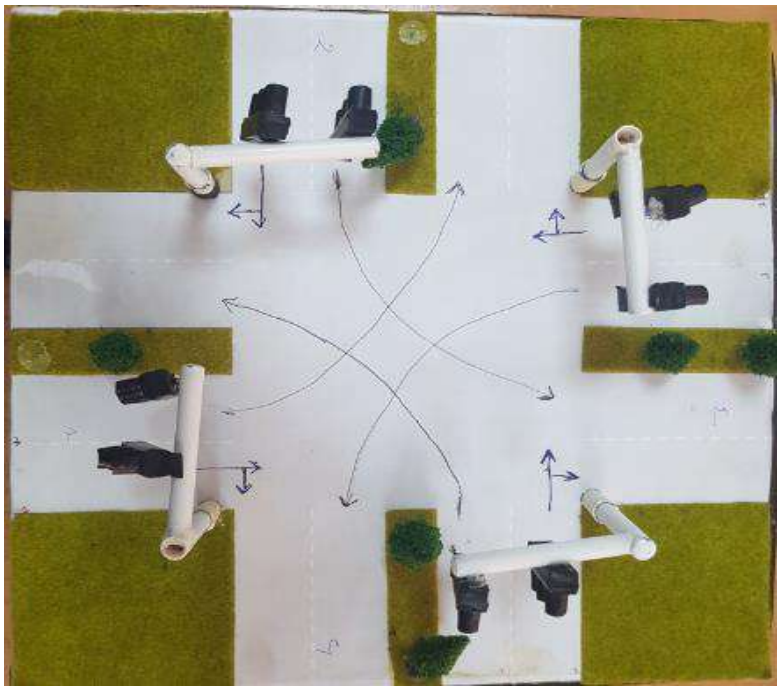


Figure III.41 : maquette du feu rouge

III.14. Conclusion

Dans ce dernier chapitre, nous avons présenté en détails notre réalisation pratique, nous commençons par la présentation de cahier de charge et les outils matériels et logiciels utilisés dans notre projet, une explication détaillée sur le contrôleur flou et la communication avec le module nRF24L01. Ensuite, nous avons expliqué le fonctionnement de flux créé dans l'environnement Node Red étape par étape.

Enfin, nous avons présenté notre interface de supervision et le prototype réalisé.

Conclusion générale

Conclusion générale

Dans ce mémoire, nous avons abordé les aspects de la gestion du trafic routier urbain. Nous avons commencé par une présentation des systèmes de transport intelligents, en donnant un aperçu général des technologies appliquées, les contextes des STI et les principes de cette tendance qui évolue quotidiennement. Nous avons ainsi présenté les stratégies utilisées par les pays développés pour appliquer des systèmes intelligents.

L'investissement dans la mobilité intelligente donnera des grands bienfaits tels que l'amélioration de la fluidité des trafics routiers, la réduction de la congestion dans les villes et augmentation du niveau de la sécurité et le confort pour les conducteurs et protection de l'environnement par la réduction des émissions de gaz à effet de serre. La gestion de trafic routier nécessite d'appliquer une stratégie intelligente afin d'améliorer leur efficacité. Nous avons développé dans notre projet un contrôleur à base de la logique floue des feux de circulation d'une intersection de manière intelligente par l'exploitation des données du trafic routier (Nombre de véhicule et le débit)

Pour une meilleure gestion de trafic routier, la supervision est une fonction indispensable pour assurer le bon fonctionnement des équipements d'un système de transport intelligent, nous avons introduit dans notre projet une interface de supervision et de contrôle des feux d'une intersection. Cette application est créée à base de l'internet des objets qui permet d'intégrer le monde physique avec le monde virtuel, les objets connectés capable d'émettre les informations issues de capteurs ainsi que les états des actionneurs (Feux de circulation). Les actions peuvent être données à partir de l'IHM développé sous Node-Red.

L'implémentation des systèmes liés au projet est effectuée à l'aide des composants électroniques pour le traitement et la communication (l'Arduino et le Raspberry et le module de communication par radio fréquence) qui sont compatibles avec des applications embarquées et les solutions Internet des Objets.

Les différents tests effectués ont montré la validité du travail réalisé. Comme perspective à notre travail, on propose d'étaler la solution développée à un réseau routier urbain ou la coopération.

Référence bibliographie

[1] B. Abderrahim and B. H. M. Imad, "Automatisation d ' une maison intelligente via une application Android", Mémoire de Master, Université Abou Bekr Belkaid Tlemcen, 2017.

[2] "Internet of Things (IOT) Applications in Intelligent Transport Systems (ITS) Part - I." [Online]. Available: https://www.masterbuilder.co.in/internet-of-things-iot-applications-in-intelligent-transport-systems-its-part-i/. [Accessed: 26-May-2019].

[3] S. Aicha, "Conception et réalisation d ' une plateforme de gestion d ' un réseau de transport, ", Mémoire de Master, Université Abou Bekr Belkaid Tlemcen, 2012.

[4] M. Rouse, "What is data analytics (DA)? - Definition from WhatIs.com." [Online]. Available: https://searchdatamanagement.techtarget.com/definition/data-analytics. [Accessed: 15-Apr-2019].

[5] S. Arora, "Top 14 Areas for Data Analytics Application." [Online]. Available: https://www.digitalvidya.com/blog/data-analytics-applications/. [Accessed: 15-Apr-2019].

[6] M. H. Kahina and M. K. Kahina, "Modélisation et simulation du trafic routier à l ' aide des réseaux de capteurs sans fils", Mémoire de Master, Université A/Mira de Béjaïa, 2013.

[7] M. Choudhary, "What is Intelligent Transport System and how it works?," *Geospatial world*, 2019. [Online]. Available: https://www.geospatialworld.net/blogs/what-is-intelligent-transport-system-and-how-it-works/. [Accessed: 07-Mar-2019].

[8] S. Faye, "Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil," ParisTech, 2012.

[9] Commission Européenne, *Mobilité 3.0 Ensemble pour la mobilité intelligente*. 2015.

[10] "5 Cities and Their Intelligent Transportation Systems | SmartMyCity." [Online]. Available: https://smartmycity.com/intelligent-transportation-systems/?fbclid=IwAR3qY3WVbvKwzsxLym-O7UckvmY0p7fHyrLeyxaSo0R9MLdAfxWDplXuN1c. [Accessed: 07-Mar-2019].

[11] "Système de contrôle commande - CETU," 2017. [Online]. Available: http://www.cetu.developpement-durable.gouv.fr/systeme-de-controle-commande-gtc-supervision-a1134.html. [Accessed: 08-Mar-2019].

[12] S. Faye, “Contrôle et gestion du trafic routier urbain par un réseau de capteurs sans fil,” ParisTech, 2014.

[13] D. Bendouda, “Contrôle des réseaux de capteurs et actionneurs sans fil pour la supervision”,Thèse de doctorat, Université Oran1 ahmed ben bella, 2018.

[14] “L ’ Internet des Objets dans les transports”, Alcatel Lucent Enterprise, 2018.

[15] S. Tronnolone, “Les domaines d’application de l’internet des objets (IoT),” 2017. [Online]. Available: https://gladiacteur.com/iot-linternet-des-objets/. [Accessed: 27-Mar-2019].

[16] Rajiv, “Applications of Industrial Internet of Things (IIoT) - RF Page,” 2018. [Online]. Available: https://www.rfpage.com/applications-of-industrial-internet-of-things/. [Accessed: 16-Jun-2019].

[17] “L’Internet des objets : à l’ère des objets connectés,” 2019. [Online]. Available: https://www.ionos.fr/digitalguide/sites-internet/droit-dinternet/internet-des-objets/. [Accessed: 02-Apr-2019].

[18] M. K. Muruganandam, D. B. Balamurugan, and D. S. Khara, “Design Of Wireless Sensor Networks For IOT Application : A Challenges and survey,” *Int. J. Eng. Comput. Sci.*, vol. 7, no. 03, pp. 23790–23795, 2018.

[19] C. McClelland, “What is the Cloud? How Does it Fit into the Internet of Things (IoT)? | IoT For All,” 2019. [Online]. Available: https://www.iotforall.com/what-is-the-cloud/. [Accessed: 07-Apr-2019].

[20] A. Shankar, “Overview of networking technologies used to build IoT solutions,” *Intelligent Tech Channels*, 2017. [Online]. Available: http://www.intelligenttechchannels.com/overview-of-networking-technologies-used-to-build-iot-solutions/. [Accessed: 02-Apr-2019].

[21] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *J. Electr. Comput. Eng.*, vol. 2017, pp. 1–25, Jan. 2017.

[22] “What is Node-RED? - iot4beginners.” [Online]. Available: https://iot4beginners.com/what-is-node-red/. [Accessed: 08-Apr-2019].

[23] “Qu’est-ce qu’un Raspberry Pi ? - Nouvelles Technologies.” [Online]. Available: https://www.nouvelles-technologies.net/quest-ce-quun-raspberry-pi/. [Accessed: 10-Apr-2019].

[24] “GPIO - Framboise 314, le Raspberry Pi à la sauce française....” [Online]. Available: https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/. [Accessed: 11-Apr-2019].

[25] Mr. Benyezza Hamza and M. T. Yassine, “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, UNIVERSITE Dr. YAHIA FARES DE MEDEA, 2017.

[26] “In-Depth: How nRF24L01 Wireless Module Works & Interface with Arduino.” [Online]. Available: https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/. [Accessed: 07-Jun-2019].

العنوان: التحكم الضبابي في أضواء حركة مرور التقاطع باستعمال أنترنت الأشياء

ملخص

يهدف هذا المشروع إلى تطبيق نظام تحكم ذكي يعتمد على المنطق الضبابي لمفترق الطرق من خلال تطوير واجهة رسومية للإشراف والتحكم من خلال استخدام مفهوم إنترنت الأشياء (IoT). يتم تنفيذ مشروعنا من خلال اللوحات الالكترونية التي تعطي حلول في التواصل ومعالجة البيانات (Raspberry)، وتوصيل أجهزة الاستشعار أو المشغلات ومعالجة البيانات (Arduino) والاتصالات (Module nRF24L01)، يتم تطوير حلول البرمجيات في بيئة برمجة مرئية "Node Red" لتطبيق IoT الخاص بنا باستخدام لغات البرمجة Python وJavaScript و C.

الكلمات المفتاحية: الاتصال، التحكم، الذكاء، إنترنت الأشياء (IoT)، مفترق الطرق، المنطق الضبابي، Node Red.

Title: Fuzzy control of intersection traffic light based on the Internet of Things

Abstract

This project aims to implement an intelligent control system based on the fuzzy logic of an intersection by developing a graphical interface of supervision and control through the concept of the Internet of Things (IoT). The implementation of our project is done through hardware solutions for interfacing and processing (Raspberry), connection of sensors or actuators and data processing (Arduino) and communication (Module nRF24L01), the software solutions are developed in a visual programming environment "Node Red" for our IoT application using Python, JavaScript and C programming languages.

Keywords: Communication, Control, Intelligent, Internet of Things (IoT), Intersection, Fuzzy Logic, Node-Red.

Titre: Contrôle flou des feux d'intersection basé sur l'Internet des Objets

Résumé

Ce projet a pour objectif de mettre en œuvre un système de contrôle intelligent basé sur la logique floue d'une intersection en développant une interface graphique de supervision et contrôle par l'utilisation du concept de l'internet des objets (IoT). La mise en place de notre projet se fait grâce à des solutions matérielles pour l'interfaçage et traitement (Raspberry), connexion des capteurs ou actionneurs et traitement des données (Arduino) et la communication (Module nRF24L01), des solutions logicielles sont développées dans un environnement de programmation visuelle « Node Red » pour notre application IoT en utilisant les langages de programmation Python, JavaScript et C.

Mots clé : Communication, Contrôle, Intelligent, Internet des objets (IoT), Intersection, logique floue, Node-Red.