



Université ABBES LAGHROUR Khenchela  
Faculté des Sciences et de la Technologie  
Département de Génie Industriel  
جامعة عباس لغرور خنشلة  
كلية العلوم والتكنولوجيا  
قسم الهندسة الصناعية



N° Série : .....

# Mémoire de fin d'étude

*Pour l'obtention du diplôme de Master*

**Filière : Télécommunications**

**Spécialité : Systèmes des Télécommunications**

## THEME

**Systeme de communication entre les  
véhicules (VNET) en utilisant la  
communication sans fil et les méthodes de  
l'intelligence artificielle**

*Réalisé par : -MEBARKI Hana*

*-GHEDIR Ouided*

**Soutenu le : 15/09/2020** *Devant le jury composé de:*

*Mr. DJAMAI Djemouai  
Mr. DOUAK Fouzi  
Mr. BEDRA Sami*

*Président  
Encadreur  
Examineur*

*Université Abbes Laghrour-Khenchela  
Université Abbes Laghrour-Khenchela  
Université Abbes Laghrour-Khenchela*

*Promotion 2019/2020*

*Je dédie ce modeste travail :*

*A mon très cher père*

*Tu as toujours été à mes côtés pour me soutenir et m'encourager.*

*Je continuerai mon chemin seulement pour tu rendre fière de moi.*

*A ma très chère mère*

*Quoi que je fasse ou que je dise, je ne saurai te remercier comme il se doit.*

*Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.*

*A mes belles sœurs : Djahira, Majda et leurs familles.*

*A mes chers frères ; Fouzi, Emad et leurs familles.*

*A le bonheur de notre famille : Foud, Wassim, Sadjed*

*A ma famille, mes proche et a ceux que me donnée de l'amour et la vivacité.*

*A tous mes amis qui m'ont encouragé, et à qui je souhaite plus de succès*

*A tous ceux que j'aime*

*Merci.*

*Hana*

*Je dédie ce modeste travail :*

*Premièrement à ma chère mère.*

*Tu présentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leur étude.*

*À mes sœurs Sarah et Soumia pour l'attention, l'aide et le soutien qu'elles m'ont apporté. En témoignage de l'attachement et l'affection que je porte pour vous.*

*À tous mes membres de ma famille, petits et grands.*

*À tous ceux qui me sont chers.*

Ouïed

# *Remerciements*

---

*Avant tout je tiens mes remerciements à ALLAH tout puissant qui nous a donné la santé, le courage, et la volonté pour réaliser ce travail.*

*En second lieu, nous tenons à remercier notre encadreur Monsieur **Douak Fouzi**, pour nous donner l'opportunité d'enrichir nos connaissances pas seulement dans ce travail mais dans tous notre parcours et aussi nous a guidé par ses remarques pertinentes, ses critiques constructives et ses conseils tout au long de la réalisation de ce travail, sans lui ce travail n'aurait pas vu le jour.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.*

*Nous désire exprimer nos plus vives et sincères reconnaissance à tous nos enseignants qui nous permettent d'acquérir plus de connaissances durant notre parcours.*

*Nous tenons à remercier particulièrement tous les responsables de département Génie industriel ainsi que nos collègues.*

*Nos plus chaleureux remerciements pour tous ceux qui de près et de loin  
Ont contribué à la réalisation de ce travail.*

## Résumé

Les réseaux véhiculaires (VNET) font partie des systèmes de transport intelligents (STI). Ces derniers tentent de répondre à la question de savoir comment équiper les véhicules de moyens de communication sans fil pour éviter les accidents et la collision, pour avoir des impacts positifs sur l'économie, l'énergie et l'environnement.

Notre but, à travers ce travail est de présenter l'étude théorique et pratique pour la réalisation d'un système de communication entre les véhicules (VNET) qui dépend de la communication sans fil et les méthodes de l'intelligence artificielle, où le système est basé sur des circuits électrique de différents composants nécessaires entouré sur les unités de traitement : Arduino (Uno ou Mega), et carte Raspberry *Pi*. On considère que chaque véhicule se comporte comme un réseau de communication sans fil (nRF, GSM, GPS). Cette communication est très utile pour suivre les mouvements d'une véhicule et l'échange d'information entre véhicule-véhicule et véhicule-infrastructure en temps réel (position géographique, accident, danger, les panneaux de signalisations, message d'urgence ...).

On trouve aussi un système d'intelligence artificielle basé sur la carte Raspberry *Pi* 3 et une caméra situé à l'avant de chaque véhicule pour la détection et reconnaissance automatique des panneaux de signalisations (TSR).

**Mots clés :** VNET, Intelligence artificielle, Arduino, nRF, GPS, GSM, Raspberry *Pi* 3, TSR.

## Abstract

Vehicular networks (VNET) are part of intelligent transportation systems (ITS). These latter attempts to answer the question of how to equip vehicles with wireless communication means avoiding accidents and better manage congestion; which have positive impacts on the economy, energy, and the environment.

Our goal, through this work, is to present the theoretical and practical study about the realization of a communication system between vehicles (VNET) which depends on the wireless communication and the artificial intelligence methods, where the system is based on electrical circuits of various necessary components that are surrounded by processing units: Arduino (Uno or Mega) and Raspberry *Pi* board. Each vehicle is considered to behave as a wireless communication network (nRF, GSM, GPS). This communication is very useful for attracting the movements of a vehicle by exchanging information (geographical position, accident, danger, warning signs, emergency message, etc.) between vehicle to vehicle and vehicle to infrastructure in real-time.

In addition, an artificial intelligence system based on a Raspberry *Pi* 3 and a camera, which is located at the front of each vehicle for the detection and automatic recognition of traffic signs (TSR) is realized.

**Keywords:** VNET, Artificial Intelligence, Arduino, nRF, GPS, GSM, Raspberry *Pi* 3, TSR.

## ملخص

تعد شبكات المركبات (VNET) جزءاً من أنظمة النقل الذكية (ITS) تحاول هذه الأخيرة الإجابة على سؤال حول كيفية تجهيز المركبات بوسائل الاتصال اللاسلكي لتجنب الحوادث وإدارة الازدحام بشكل أفضل ؛ التي لها آثار إيجابية على الاقتصاد والطاقة والبيئة.

هدفنا من هذا العمل هو تقديم دراسة نظرية وتطبيقية لتحقيق نظام اتصال بين المركبات (VNET) الذي يعتمد على الاتصال اللاسلكي وأساليب الذكاء الاصطناعي حيث يركز النظام على مجموعة من الدوائر الكهربائية المختلفة التي تحتوي على العديد من العناصر الكهربائية الضرورية حيث نتحكم فيها بواسطة لوحات معالجة مثل Arduino و Raspberry. تعتبر كل مركبة كشبكة اتصالات لاسلكية (nRF و GSM و GPS) ، يعتبر هذا الاتصال مهم جداً لمتابعة حركة السيارات من خلال تبادل المعلومات (الموقع الجغرافي، حوادث المرور، والخطر، إشارات المرور، ورسالة الطوارئ، وما إلى ذلك) بحيث يكون هذا الاتصال بين المركبات والمركبات وكذلك بين المركبات و البنية التحتية في وقت حقيقي.

بالإضافة إلى نظام ذكاء اصطناعي يعتمد على Raspberry Pi 3 وكاميرا موجودة في مقدمة كل مركبة للكشف والتعرف التلقائي على إشارات المرور (TSR).

**الكلمات المفتاحية:** VNET, وسائل اتصالات لاسلكية, الذكاء الاصطناعي Raspberry Pi, nRF, Arduino, GSM, GPS, الكشف والتعرف التلقائي.

---

# *Liste des tableaux*

---

<b>Tableau I. 1.</b> Les statistiques des accidents de route dans l'Algérie (2011-2019). .....	14
<b>Tableau II. 1.</b> Tram du protocole ESB. ....	35
<b>Tableau III. 1.</b> La division de base de données en trois ensembles : apprentissage, validation, et test. ....	45
<b>Tableau III. 2.</b> Architecture du modèle MLP. ....	63
<b>Tableau III. 3.</b> Meilleurs paramètres du modèle MLP. ....	66
<b>Tableau III. 4.</b> Meilleurs paramètres du modèle SVM. ....	68
<b>Tableau IV. 1.</b> Affectation des broches GPS. ....	81
<b>Tableau IV. 2.</b> Commandes et les rôles d'un programme GPS. ....	83
<b>Tableau IV. 3.</b> Commandes AT dédiées au service SMS. ....	86
<b>Tableau IV. 4.</b> Connexions des broches LCD de communication. ....	88

# *Liste des figures*

---

<b>Figure I. 1.</b> Un exemple d'un réseau VNET.....	6
<b>Figure I. 2.</b> Modèle des réseaux VNETs et son mode de communications.....	8
<b>Figure I. 3.</b> Exemples de sécurités d'applications des réseaux VNET.....	13
<b>Figure II. 1.</b> Schéma d'une onde électromagnétique plane se propageant dans l'axe X. Les composantes électrique ( $E$ ) et magnétique ( $H$ ) de l'onde sont représentées dans les axes Y et Z.....	18
<b>Figure II. 2.</b> Une cellule. ....	20
<b>Figure II. 3.</b> Structure hiérarchique des cellules.....	20
<b>Figure II. 4.</b> Structure cellulaire du réseau GSM. ....	21
<b>Figure II. 5.</b> Structure générale d'un réseau GSM. ....	22
<b>Figure II. 6.</b> La constellation de satellites du système de navigation GPS. ....	24
<b>Figure II. 7.</b> Exemple de positionnement des 27 satellites lors d'une orbite terrestre. ....	24
<b>Figure II. 8.</b> Le segment de contrôle. ....	25
<b>Figure II. 9.</b> Segment utilisateur.....	25
<b>Figure II. 10.</b> Calcul de position à l'aide de la méthode de trilatération.....	26
<b>Figure II. 11.</b> Principe de la trilatération. ....	27
<b>Figure II. 12.</b> Positionnement en 3D. ....	28
<b>Figure II. 13.</b> Architecture d'un nœud capteur.....	29
<b>Figure II. 14.</b> Carte Arduino Uno.....	30
<b>Figure II. 15.</b> Les composants d'Arduino Mega 2560. ....	31
<b>Figure II. 16.</b> Description de l'Interface Arduino IDE.....	32
<b>Figure II. 17.</b> Structure d'un programme Arduino.....	33
<b>Figure II. 18.</b> La carte Raspberry <i>Pi</i> 3.....	33
<b>Figure II. 19.</b> Connexion ordinateur et Raspberry <i>Pi</i> .....	34
<b>Figure II. 20.</b> (a) Module nRF24L01 (b) Module nRF24L01-PA-LNA.....	35
<b>Figure II. 21.</b> Types de communications à base de RF24. ....	36
<b>Figure II. 22.</b> Le module GSM SIM 800L. ....	36
<b>Figure II. 23.</b> Schéma de fonctionnement AT COMMANDS. ....	37
<b>Figure II. 24.</b> Module GPS GY-NEO6MV2. ....	38

<b>Figure III. 1.</b> Schéma général d'un système de détection et reconnaissance. ....	43
<b>Figure III. 2.</b> Les formes et les couleurs des panneaux. ....	44
<b>Figure III. 3.</b> Panneaux signalisation. ....	44
<b>Figure III. 4.</b> Nombre des échantillons dans chaque classe. ....	45
<b>Figure III. 5.</b> Etapes du Descripteur HOG. ....	46
<b>Figure III. 6.</b> Applications de descripteur de HOG sur deux images de teste, dimensions des images [16×16], Cellule [8×8]. ....	47
<b>Figure III. 7.</b> Schéma d'un neurone biologique. ....	48
<b>Figure III. 8.</b> Représentation du modèle d'un neurone artificiel. ....	48
<b>Figure III. 9.</b> Structure d'un réseau de neurone multicouche. ....	49
<b>Figure III. 10.</b> Fonction de transfert : (a) du neurone "seuil"; (b) du neurone "linéaire", (c) "sigmoïde", et (d) du neurone "tangente hyperbolique" ....	51
<b>Figure III. 11.</b> Cas linéairement séparable. ....	54
<b>Figure III. 12.</b> Transformer l'espace de départ vers un espace de dimension supérieure. ....	54
<b>Figure III. 13.</b> Séparation entre trois classes en utilisant la méthode 1vsR. ....	56
<b>Figure III. 14.</b> Image de test. ....	58
<b>Figure III. 15.</b> Résultats de l'application de la méthode de seuillage simple sur les images de test. ....	58
<b>Figure III. 16.</b> Image normalisé de la méthode MSER sur les images de test. ....	59
<b>Figure III. 17.</b> Résultats de l'application de la méthode de segmentation MSER sur les images de test. ....	59
<b>Figure III. 18.</b> Résultats de l'application de la méthode de segmentation MSER et filtrage morphologique sur les images de test. ....	60
<b>Figure III. 19.</b> Résultats de l'application de la méthode d'étiquetage des composants connexes sur les images de test. ....	61
<b>Figure III. 20.</b> Détection des panneaux de signalisations sur les images de test. ....	61
<b>Figure III. 21.</b> Redimensionnés des images en blocs de dimension [16×16], [32×32], [48×48], et [64×64]. ....	63
<b>Figure III. 22.</b> Précision ( <i>Acc</i> ) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [16×16], le descripteur HOG et le classifieur MLP. ....	64

<b>Figure III. 23.</b> Précision ( <i>Acc</i> ) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [32×32], le descripteur HOG et le classifieur MLP. ....	64
<b>Figure III. 24.</b> Précision ( <i>Acc</i> ) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [48×48], le descripteur HOG et le classifieur MLP. ....	65
<b>Figure III. 25.</b> Précision ( <i>Acc</i> ) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [64×64], le descripteur HOG et le classifieur MLP. ....	65
<b>Figure III. 26.</b> Matrice de confusion sur l'ensemble de test (HOG-MLP). ....	66
<b>Figure III. 27.</b> Précision ( <i>Acc</i> ) en fonction de $C$ et $\gamma$ . La dimension de l'image (a) [16×16], (b) [32×32], (c) [48×48], et (d) [64×64], le descripteur HOG et le classifieur SVM.....	67
<b>Figure III. 28.</b> Matrice de confusion sur l'ensemble de test (HOG-SVM). ....	68
<b>Figure III. 29.</b> Etude comparative entre HOG-MLP et HOG-SVM. ....	68
<b>Figure IV. 1.</b> Architecture globale du système VNET. ....	71
<b>Figure IV. 2.</b> Schéma globale du système de communication équipé dans les véhicules.....	72
<b>Figure IV. 3.</b> Véhicule équipé par un ultrason et une communication radiofréquence.....	73
<b>Figure IV. 4.</b> Le prototype de l'infrastructure pour tester notre système. ....	73
<b>Figure IV. 5.</b> Organigramme proposé de la communication (a) véhicules à infrastructure et (b) véhicule à véhicule.....	74
<b>Figure IV. 6.</b> Système de reconnaissance automatique des panneaux de signalisations, (a) Raspberry et camera, (b) Système de transport intelligent. ....	75
<b>Figure IV. 7.</b> Schéma synoptique d'un système de message d'urgence.....	76
<b>Figure IV. 8.</b> Organigramme proposé du système de message d'urgence. ....	77
<b>Figure IV. 9.</b> Branchement de radiofréquence nRF24L01 avec la carte Arduino Uno.....	78
<b>Figure IV. 10.</b> Circuit électrique de module radiofréquence nRF24L01 avec la carte Arduino Uno. ....	79
<b>Figure IV. 11.</b> Schéma du capteur ultrason avec la carte Arduino Uno.....	79
<b>Figure IV. 12.</b> Circuit électrique du capteur ultrason avec la carte Arduino Uno.....	80
<b>Figure IV. 13.</b> Module GPS (GY-NEO6MV3). ....	81
<b>Figure IV. 14.</b> Branchement de module GPS avec Arduino Mega. ....	81
<b>Figure IV. 15.</b> Circuit électrique du module GPS avec la carte Arduino Mega.....	82

<b>Figure IV. 16.</b> (a) Branchement du module GSM avec une carte Arduino Mega.....	84
<b>Figure IV. 17.</b> Circuit électrique du module GSM avec la carte Arduino Mega. ....	85
<b>Figure IV. 18.</b> Schéma récapitulatif du système GSM.....	85
<b>Figure IV. 19.</b> Schéma pratique du Module SIM800L.....	86
<b>Figure IV. 20.</b> Branchement d'écran LCD avec Arduino Mega. ....	88
<b>Figure IV. 21.</b> Circuit électrique d'écran LCD avec Arduino Mega.....	88
<b>Figure IV. 22.</b> Détection et reconnaissance automatique des panneaux de signalisations, (a) Obligation de tourner à droite, (b) Arrêt obligatoire, (c) Obligation d'aller tout droit. ....	91
<b>Figure IV. 23.</b> Assemblage de radiofréquence nRF24L01 avec la carte Arduino Uno et l'afficheur LCD 16x2. ....	92
<b>Figure IV. 24.</b> Circuit électrique de module nRF24L01 connecté avec la carte Arduino Uno et l'afficheur LCD 16x2. ....	92
<b>Figure IV. 25.</b> Assemblage du robot mobile 1. ....	94
<b>Figure IV. 26.</b> Assemblage du robot mobile 2. ....	95
<b>Figure IV. 27.</b> Communication véhicules- infrastructure.....	95
<b>Figure IV. 28.</b> Communication véhicule-véhicule. ....	96
<b>Figure IV. 29.</b> Robot mobile 1 après l'assemblage. (a) Face de haut, (b) Face de l'avant droit, (c) Face en avant, (d) Face en arrière. ....	98
<b>Figure IV. 30.</b> Robot mobile 2 après l'assemblage. (a) Face de haut, (b) Face de l'avant droit, (c) Face en avant.....	99
<b>Figure IV. 31.</b> Prototype de l'infrastructure, (a) indication de présence ou absence des véhicules dans la zone de couverture de radiofréquence: Leds et afficheur LCD, (b) module nRF24L01.....	100
<b>Figure IV. 32.</b> Test d'affichage 1. ....	101
<b>Figure IV. 33.</b> Test d'affichage 2. ....	101
<b>Figure IV. 34.</b> Test d'affichage 3. ....	102
<b>Figure IV. 35.</b> Montre les coordonnées géographiques du véhicule à l'aide du logiciel Arduino IDE. ....	103
<b>Figure IV. 36.</b> Montre la position du véhicule à l'aide de Google Maps. ....	103
<b>Figure IV. 37.</b> Expérimental du Système de message d'urgence. ....	104
<b>Figure IV. 38.</b> Affichage des positions géographiques (latitude et longitude).....	105
<b>Figure IV. 39.</b> Texte SMS reçu envoyé par le système d'alerte de véhicule. ....	105

# *Liste des symboles*

---

**ASCII** : American Standard Code for Information Interchange.

**BTS Base** : Transmitter Station.

**BS** : Base Station.

**BSC** : Base Station Controller.

**CA** : Autorité de Confiance.

**CC** : Courant Continu.

**CRC** : cyclic redundancy check.

**DHCP** : Dynamic Host Configuration Protocol.

**DSRC** : Dedicated short Range Communication.

**ECC**: Etiquetage en composantes connexes.

**E/S** : Entrée/Sortie.

**EEPROM** : Electrically Erasable Programmable Read-Only Memory.

**ESB** : Enhanced Shock Burst.

**ECC** : Etiquetage en composantes connexes.

**FM** : Frequency modulation.

**GFSK** : Gaussian Frequency-Shift Keying.

**GPS** : Global Positioning System.

**GSM** : Global System for Mobile Communications.

**GPIO** : General Purpose Input/Output.

**GPRS** : General Packet Radio Service.

**GPSS** : Global Positioning System Fix Data.

**HDMI** : High Definition Multimedia Interface.

**HLR** : Home Local Register.

**HOG** : Histogram of Oriented Gradients.

**Http** : HyperText Transfer Protocol.

**IDE** : Integrated Development Environment.

**IEEE** : Electrical and Electronics Engineers.

**IMEI** : International Mobile Equipment Identity.

**IP** : Internet Protocol.

**ITS** : Intelligent Transportation System.

**LCD** : Liquid Crystal Library.

**LED** : Light Emitting Diode.

**MAC** : Media Access Control.

**Mbps** : Mega bits par secondes.

**ME** : Mobile Equipement.

**MISO** : Master Input Slave Output.

**MOSI** : Master Output Slave Input.

**MLP** : Multi-Layer Perceptron.

**MS** : Mobile Station.

**MSER** : Maximally Stable Extremal Regions.

**MMS** : Multimedia Message Service.

**NMEA** : *National Marine Electronics Association.*

**NTIC** : Nouvelles technologies de l'information et de la communication.

**NS** : Network Sub-system.

**OBU** : On-Board Unit.

**PIN** : Personal Identification Number.

**PDU** : Protocol Description Unit.

**PWM** : Pulse Width Modulation.

**RAM** : Random Access Memory.

**RF** : Radiofréquences.

**RGB** : Red Green Blue.

**RSU** : Road Side Unit.

**SCK** : SPI Serial Clock.

**SIM** : Subscriber Identity Module.

**SPI** : Serial Peripheral Interface.

**SSH** : Serveur Secure Shell.

**SVM** : Support Vector Machine.

**SMS** : Short Message Service.

**STI** : Système de transport intelligent.

**TA** : Terminal Adaptateur.

**TCP** : Transmission Control Protocol.

**TE** : Terminal Equipement.

**TSR** : Traffic Signs Recognition.

**VLR** : Visitor Location Register.

**VNET** : Réseaux Véhiculaires.

**V2I** : Vehicle-to-Infrastructure.

**V2V** : Vehicle-to-Vehicle.

**UART** : Universel Asynchronous Receiver Transmitter.

**USB**: Universal Serial Bus.

**Wi-Fi**: Wireless Fidelity.

---

# *Table des matières*

---

<b>Introduction Générale.....</b>	<b>1</b>
 <b>Chapitre I</b> <b>Communication entre les véhicules</b>	
I.1. Introduction .....	5
I.2. Réseaux Véhiculaires .....	5
I.2.1. Définition d'un réseau VNET.....	5
I.3. Le système des réseaux VNETs.....	6
I.3.1. Architecture et composants .....	6
I.3.2. Les modes de communication dans VNETs.....	7
I.3.3. Caractéristiques .....	8
I.3.4. Les type de messages.....	10
I.3.4.1. Message lié à la sécurité .....	10
I.3.4.2. Message à valeur ajoutée .....	10
I.3.5. Applications des réseaux VNET.....	10
I.3.5.1. Alerter en cas d'accidents.....	11
I.3.5.2. Alerter en cas de ralentissement anormal .....	11
I.3.5.3. La conduite collaborative .....	12
I.3.5.4. Hot spot sur les autoroutes.....	12
I.3.5.5. Gestion des espaces libres dans les parkings.....	12
I.4. Les principaux types de la collision et les accidents.....	13
I.5. Conclusion.....	16

## Chapitre II

### Communication sans fil

II.1. Introduction.....	18
II.2. Communication radiofréquence.....	18
II.2.1. Radiofréquence dans les systèmes VNET.....	19
II.3. Communication GSM.....	19
II.3.1. Notion de réseaux cellulaires.....	19
II.3.2. Taille des cellules.....	20
II.3.3. Structure du réseau GSM.....	21
II.3.4. Les équipements du réseau GSM.....	22
II.3.5. GSM dans les systèmes VNET.....	23
II.4. Système de localisation GPS.....	23
II.4.1. Différents segments du système GPS.....	23
II.4.1.1. Segment spatial.....	23
II.4.1.2. Segment de contrôle.....	24
II.4.1.3. Segment utilisateur.....	25
II.4.2. Principe de fonctionnement.....	26
II.4.2.1. Mesure de la position.....	26
II.4.2.2. Mesure de la vitesse.....	28
II.4.3. Localisation dans les systèmes VNET.....	29
II.5. Architecture d'un nœud capteur.....	29
II.5.1. Unité de traitement.....	29
II.5.1.1. Carte Arduino.....	30
II.5.1.1.1. Partie Hardware (Le matériel).....	30
II.5.1.1.2. Partie logiciel (Software).....	32
II.5.1.2. Carte Raspberry <i>Pi</i> .....	33
II.5.1.2.1. Partie Hardware.....	33
II.5.1.2.2. Partie logiciel.....	34
II.5.2. Les unités de communication.....	34
II.5.2.1. Module radio fréquence.....	34
II.5.2.2. Module GSM.....	36

II.5.2.3. Module GPS .....	38
II.6. Conclusion .....	39

## Chapitre III

### Traitement d'image et Intelligence artificielle

III.1. Introduction .....	41
III.2. Détection et reconnaissance automatique .....	41
III.3. Reconnaissance des systèmes TSR .....	42
III.4. Phase de calibrage .....	43
III.4.1. Description des bases de données .....	43
III.4.2. Extraction des caractéristiques .....	45
III.4.3. Apprentissage .....	47
III.4.3.1. Réseaux de neurones multicouches MLP .....	47
III.4.3.1.1. Fonctions de transfert .....	50
III.4.3.1.2. Mise en œuvre des réseaux neuronaux .....	51
III.4.3.1.3. Algorithmes d'apprentissage .....	52
III.4.3.2. Machine à vecteurs de support .....	53
III.4.3.3. Technique de validation .....	55
III.4.3.3.1. Validation simple .....	55
III.4.3.4. Classification multiclasse .....	55
III.4.3.4.1. Une-contre-reste (1vsR) .....	56
III.5. Phase en temps réel .....	57
III.5.1. Segmentation .....	57
III.5.1.1. Seuillage fixe .....	57
III.5.1.2. Régions extrêmes maximales stables (MSER) .....	57
III.5.1.3. Application des méthodes de segmentation sur les images de test ..	58
III.5.2. Détection .....	60
III.6. Classification TSR par réseaux de neurones artificiels et machine à vecteurs de support	62
III.6.1. Évaluation des performances .....	62
III.6.1.1. Taux de classification .....	62
III.6.1.2. Matrice de confusion .....	62
III.6.2. Descripteur HOG .....	62

_____ III.6.3. Classification par le perceptron multicouche (Multi Layer Perceptron MLP)	63
_____ III.6.4. Classification par SVM (Machines à vecteurs de support)	67
III.7. Conclusion	69

## Chapitre IV

### Réalisation pratique

IV. 1. Introduction	71
IV. 2. Architecture globale du système VNET	71
_____ IV.2.1. Communication véhicules à infrastructure	74
_____ IV.2.2. Communication véhicule à véhicule	75
_____ IV.2.3. Système de message d'urgence	76
IV. 3. Fonctionnement du système	77
_____ IV.3.1. Modules nRF24L01	78
_____ IV.3.2. Capteurs HC-SR04	79
_____ IV.3.3. Modules GPS	80
_____ IV.3.3.1. Rôle du module GPS dans le projet	81
_____ IV.3.3.2. Branchement du module GPS	81
_____ IV.3.3.3. Programmation de Module GPS	82
_____ IV.3.3.4. Exemple de programme	83
_____ IV.3.4. Modules GSM	84
_____ IV.3.4.1. Rôle du module GSM dans le projet	84
_____ IV.3.4.2. Branchement du module GSM	84
_____ IV.3.4.3. Commandes AT	86
_____ IV.3.4.4. Code source	86
_____ IV.3.4.5. Exemple de programme	87
_____ IV.3.5. Affichage à cristaux liquides (LCD)	87
_____ IV.3.5.1. Le rôle de l'affichage à cristaux liquides (LCD) dans le projet	87
_____ IV.3.5.2. Branchement de l'écran LCD	87
_____ IV.3.5.3. Programmation de l'écran LCD	89
_____ IV.3.5.4. Exemple de programme	89
_____ IV.3.6. Raspberry Pi 3	90
_____ IV.3.6.1. Connexion à distance	90

_____	IV.3.6.1.1. Réserveion d'adresse IP.....	90
_____	IV.3.6.1.2. Système de détection et reconnaissance automatique.....	90
_____	IV.3.7. Schéma d'assemblage .....	91
_____	IV.3.7.1. Prototype infrastructure .....	91
_____	IV.3.7.2. Prototype communication véhicule-véhicule et véhicule- infrastructure.....	92
_____	IV. 4. Réalisation finale et tests .....	95
_____	IV.4.1. Les tests d'affichage de l'infrastructure.....	101
_____	IV.4.2. Teste du système de message d'urgence.....	102
_____	IV.4.2.1. Parité logiciel .....	102
_____	IV.4.2.2. Parité matériel .....	104
_____	IV. 5. Conclusion.....	105
_____	<b>Conclusion générale.....</b>	<b>108</b>
_____	<b>Annexe: Budget prototype.....</b>	<b>111</b>
_____	<b>Bibliographie .....</b>	<b>113</b>

# *Introduction générale*

---

## Introduction générale

L'avancé spectaculaire des technologies de télécommunication sans fil peut contribuer à promouvoir la vie quotidienne; ou le monde a été changé entièrement. Cette évolution technologique répond aux exigences de l'individu, afin d'accéder à la donnée à partir des endroits différents à n'importe quel moment.

Des applications intelligentes issues de l'avancé de l'intelligence artificielle est créditée de changer la vie ; notamment dans les maisons, les sociétés et les véhicules ..., etc. Dans ce contexte, une des applications vise à renforcer la prévention routière, et à munir nos voitures et nos routes de capacités permettant de rendre la route plus sécurisée (les informations sur le trafic, les accidents, les dangers, les déviations possibles et les informations météorologiques,..., etc.). Ce type d'application permet de définir le concept des systèmes de transports intelligents (ITS, Intelligent Transportation System), son rôle est d'améliorer la sécurité, l'efficacité et la convivialité dans les systèmes de transport routier, à travers l'utilisation des nouvelles technologies de l'information et de la communication (NTIC) [1]. Parmi ces systèmes, on trouve les systèmes des réseaux véhiculaires (VNET).

Les systèmes VNET désignent les applications des nouvelles technologies d'intelligence artificielle et de communication sans fil dans le domaine des transports. Les véhicules ont été rendus intelligents, grâce à l'ajout de nouvelles fonctions afin d'appréhender leur environnement, en basant sur des informations venant des capteurs. Ces fonctions permettent aux véhicules de communiquer entre eux grâce à des interfaces radiofréquence, pour traiter les informations captées et prendre des décisions sur le comportement du véhicule ; où le traitement se fait par un ordinateur embarqué. Les différents types de communication radiofréquence utilisés dans la communication des véhicules sont appelés «Vehicle-to-Vehicle Communication, V2V » [2] ; ou ce utilisés dans des unités installées au bord de la route V2I (Vehicle-to-Infrastructure Communications) [3].

Un des problèmes majeurs des systèmes ITS est de rendre la voiture intelligente apte pour comprendre l'environnement où elle circule, notamment de connaître les panneaux de signalisations,...etc. Notre travail est divisée en deux phases, la première est le calibrage pour trouver les meilleurs paramètres de notre modèle hors ligne, et une deuxième phase qui est en temps réel permet de détecter et d'identifier les panneaux de signalisations, en basant sur l'exploitation des paramètres extraits dans la première phase. Pour les deux phases, la reconnaissance automatique des objets capturés est passée essentielle par trois étapes (i) la segmentation, telles que régions extrêmes maximales stables (MSER) [4], et le seuillage fixe

[5, 6], (ii) la détection : étiquetage en composantes connexes (ECC) [7], (iii) et les techniques de caractérisation et de classification, telles que, Histogramme de gradients orientés (HOG) [8], et les techniques d'intelligence artificielle [9]: Machine à vecteurs de support (SVM), Réseaux de neurones multicouches (MLP) [10].

Le but commun de toutes ces techniques est d'obtenir un taux de classification élevé ou une meilleure identification de notre objet (panneaux de signalisations routière). Ce sont toutes ces considérations qui ont guidé ce travail. Par conséquent l'objectif est de proposer des méthodes de reconnaissance pour des applications spécifiques, notamment dans le domaine ITS. Nous montrerons que les algorithmes proposés, comme techniques sur lesquelles notre travail est articulé, jouissent des propriétés très intéressantes pour atteindre objectif souligné.

En Algérie, les taux d'accidents et de décès liés à la circulation routière sont les plus élevés en comparaison avec d'autre pays ; les technologies VNET soit inexistante, soit non accessible chez nous. Pour ces raisons nous avons proposé de réaliser dans notre projet de fin d'étude un système de communication entre les véhicules (VNET) à base de communication sans fil (radiofréquence, GPS, GSM) et les méthodes de l'intelligence artificielle en lien avec la carte Raspberry Pi et une caméra. Lorsque le véhicule se trouve dans une situation dangereuse, notre système transmet des notifications au conducteur en lui permettant instantanément de prendre des décisions afin d'adapter la conduction avec la scène.

Notre mémoire est organisé en quatre chapitres :

Dans **Le premier chapitre**, on va expliquer quelques notions de bases liées à notre projet; définition du réseau VNET, l'architecture et les composants; les modes de communication dans VNETs ainsi que les caractéristiques. Également on va citer les différents types des messages échangés dans un réseau VNET et aussi les applications, enfin on présentera les principaux types de la collision et les accidents.

Dans **Le deuxième chapitre**, on va présenter les technologies de capteurs sans fil tels que la radiofréquence, GSM (Global System for Mobile Communications), et le système de positionnement GPS (Global Positioning System).

**Le troisième chapitre** est dédié à la présentation d'un système de reconnaissance de panneaux routiers (TSR- Traffic Signs Recognition) basé sur les méthodes de l'intelligence artificielle. On va utiliser le descripteur de caractéristique HOG (Histogram of Oriented Gradients) combiné avec deux méthodes de classification : perceptron multicouche MLP (Multi-Layer Perceptron) et machine à vecteurs de support SVM (Support Vector Machine)

pour calculer les caractéristiques et les descripteurs des objets extraits à l'étape de détection et ensuite l'identification des types de panneaux de signalisation.

Dans **Le quatrième chapitre**, on va présenter et expliquer la conception et la réalisation matérielle et logicielle de notre travail, on parlera de la réalisation d'un système de communication entre les véhicules (VNET) en utilisant les différents types de communication sans fil et les méthodes de l'intelligence artificielle.

Enfin, la dernière partie de ce mémoire est consacrée à la **conclusion générale** et quelques **perspectives**.

# *Chapitre I*

---

## *Communication entre les véhicules*

## **I.1. Introduction**

Les accidents de la route deviennent un phénomène courant dans notre pays, tous les jours un grand nombre d'accidents se produit dans différentes régions du pays. Le problème majeur de cette affaire ne reçoit pas le signal approprié et directions. Mais maintenant les gouvernements, constructeurs automobiles et consortium d'industriels, ont fixé la réduction des accidents de la route comme une priorité majeure. Afin de réussir ce challenge, une idée novatrice a été de rendre les véhicules et les routes plus intelligents par le biais des communications sans fil. En effet, les réseaux sans fil ont connu ces dernières années un essor spectaculaire, ils s'imposent aujourd'hui de façon indéniable. Leur apparition d'un côté et l'avancement des technologies de la communication et de l'information de l'autre, a donné lieu à la naissance de ce qu'on appelle les Systèmes de Transport Intelligents (STI) qui ont comme objectif de rendre plus efficace le système routier.

Les réseaux véhiculaires (VNET) sont considérés parmi les technologies récentes de communication sans fil, de nos jours les véhicules ne sont plus que de simples moyens de transport, ils sont dotés d'une source d'intelligence par leur interaction avec l'environnement routier grâce à des équipements informatiques embarqués dans les véhicules, les véhicules peuvent communiquer les uns avec les autres et avec des équipements fixes ou mobiles.

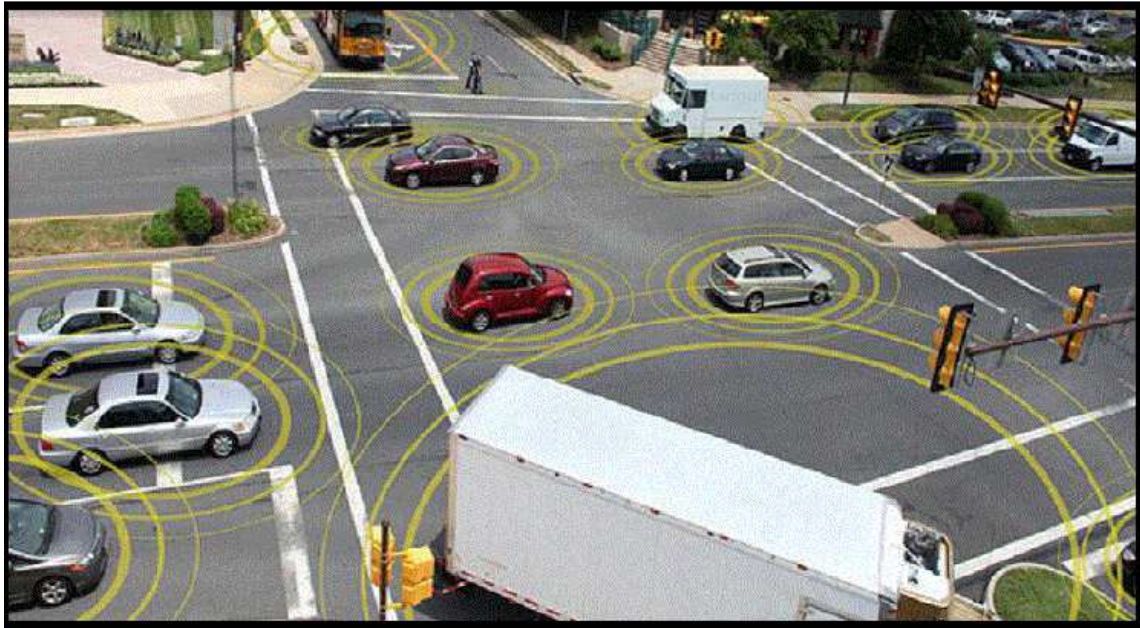
Dans ce chapitre en va voir les différents types des réseaux entre les véhicules, leurs modes de communications, leurs caractéristiques, et l'objectif de cette technologie. Ensuite, les services et les applications offertes dans ces réseaux.

## **I.2. Réseaux Véhiculaires**

### **I.2.1. Définition d'un réseau VNET**

Les VNETS sont des réseaux pour lesquelles les nœuds de communications entre des véhicules. Ces réseaux améliorent particulièrement la sécurité du trafic sur la route et globalement, la sécurité du système de transport grâce à des applications dites de sécurité. Ils améliorent aussi l'expérience de conduite grâce à des applications dites de confort. Les communications se déroulent d'une part entre les infrastructures placées le long des rues et les véhicules et d'autres parts entre les véhicules entre eux. Il existe plusieurs types d'applications selon l'information à transmettre et selon le contexte. Nous présentons ici quelques-unes des applications liées à la sécurité du trafic. Il s'agit de : l'application d'avertissement de la violation des feux de signalisation, l'application d'avertissement à l'approche d'un tournant, l'application d'avertissement coopératif d'une collision imminente, l'application d'avertissement de changement de voie, etc. Toutes les applications que nous

venons de citer ont pour caractéristique commune qu'elles dépendent des variables publiées par le véhicule qui initie la communication. En effet lorsqu'un véhicule reçoit une alerte ou un avertissement, il exploite les informations publiées par l'émetteur telles que sa vitesse, sa position géographique, son accélération, sa direction ou encore son rayon de transmission. Grâce à ces informations et le type d'alerte reçue, le conducteur du véhicule récepteur peut réagir efficacement pour éviter un accident ou une situation dangereuse [11].



**Figure I. 1.** Un exemple d'un réseau VNET.

### **I.3. Le système des réseaux VNETs**

#### **I.3.1. Architecture et composants**

Un réseau VNETs est constitué principalement de trois entités [12]:

##### **TA (Trusted Authority)**

Dites CA en français (autorité de confiance). C'est une source d'authenticité de l'information, elle assure la gestion et l'enregistrement de toutes les entités sur le réseau (RSU et OBU). La TA est sensée connaître toutes les vraies identités des véhicules et au besoin les divulguer pour les forces de l'ordre. Aussi, la TA dans certains travaux se charge de la délivrance et l'attribution des certificats et des pseudonymes de communications

##### **RSUs (RoadSide Unit)**

Ces entités sont les subordonnés des TA. Elles sont installées au bord des routes. Elles peuvent être principalement, des feux de signalisation, des lampadaires ou autres. Leur principale responsabilité est de soutenir la TA dans la gestion du trafic et des véhicules.

Elles représentent des points d'accès au réseau et aux différentes informations sur la circulation.

### **OBU (On-Board Unit)**

Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres). Leurs rôles sont d'assurer la localisation, la réception, le calcul, le stockage et l'envoi des données sur le réseau. Ce sont des émetteurs-récepteurs qui assurent la connexion du véhicule au réseau.

### **I.3.2. Les modes de communication dans VNETs**

Dans les réseaux VNETs, on trouve principalement, les entités fixes qui constituent l'infrastructure (RSU et TA) et les entités mobiles (les véhicules). Pour pouvoir échanger les différentes informations et données liées à la sécurité et au confort des usagers de la route, ces différentes entités doivent établir des communications entre elles. Pour cette raison, on distingue deux types de communications véhicule à véhicule (V2V) et véhicule à infrastructure. Comme illustré dans la Figure I.2 [12].

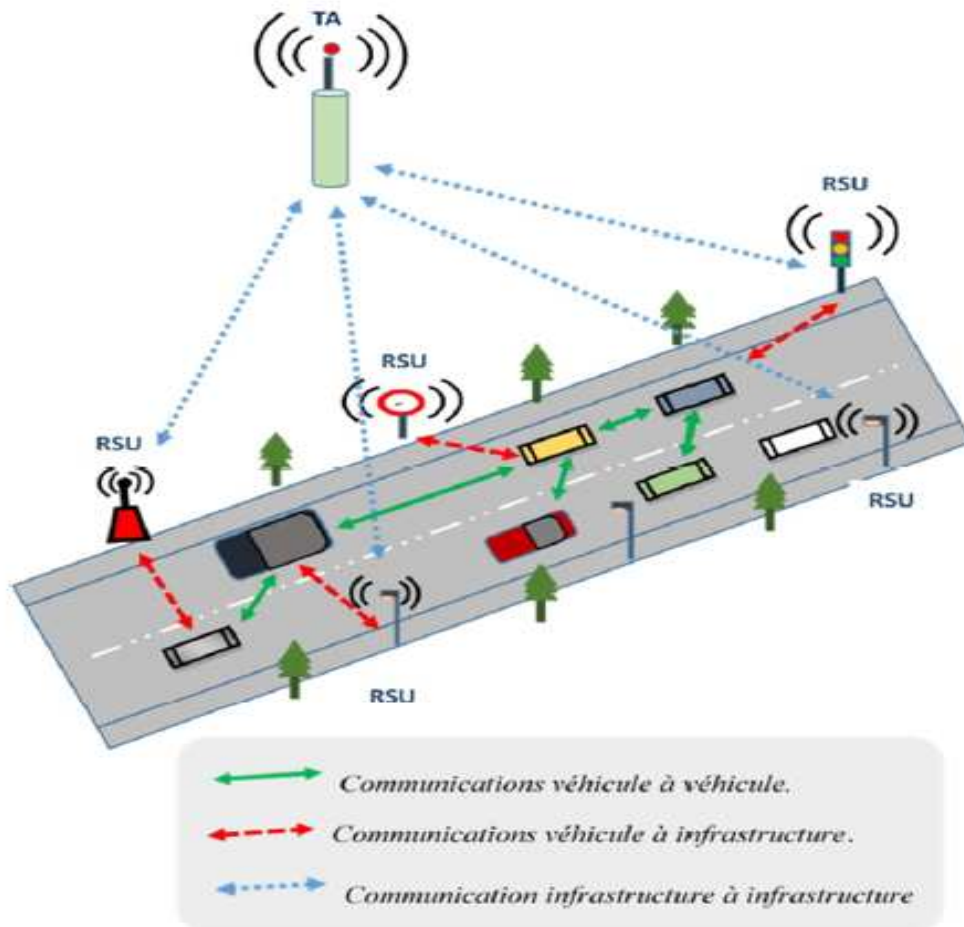
#### ✓ **Communication véhicule-à-véhicule (V2V)**

Ce type de communication fonctionne à l'aide des dispositifs installés dans les véhicules appelés OBU (On-Board Unit), suivant une architecture décentralisée.

La communication entre deux véhicules se fait directement, en mode Ad hoc inter véhiculaire. Ils n'ont pas besoin de faire appel aux infrastructures pour pouvoir communiquer entre eux. À condition que chaque véhicule soit à la portée de l'autre (Zone radio). Sinon, ils font appel à d'autres véhicules, qui vont jouer le rôle d'un pont(Intermédiaire) pour eux. Ce type de transmission est appelée communication à multi sauts.

#### ✓ **Communication véhicules à infrastructure**

La communication véhicule à infrastructure (V2I) est aussi appelée une communication en mode infrastructure. Ce mode de communication est assuré grâce aux différentes entités du réseau VNETs. En effet, les OBUs (On-Board Unit) des véhicules, les RSUs (Road Side Unit) placés aux bords des routes et même les TA (Trusted Authority) contribuent tous entre eux pour assurer les communications dans le réseau véhiculaire. Ce mode de communication assure une connectivité relativement forte par rapport à la communication en mode V2V (véhicule à véhicule). Comme il assure une meilleure utilisation des ressources du réseau. Ainsi, elle permet aux véhicules de bénéficier de plus de fonctionnalités et services comme l'accès à l'Internet et les informations météorologiques.



**Figure I. 2.** Modèle des réseaux VNETs et son mode de communications.

### I.3.3. Caractéristiques

Les réseaux VNETs ils se distinguent avec certaines caractéristiques, qui retiennent une grande attention, à savoir :

#### ➤ Capacité et autonomie d'énergie

Parmi les spécificités des réseaux VNETs, la grande capacité d'énergie caractérisant les véhicules qui les détiennent de leurs systèmes d'alimentation. Ce qui n'est pas le cas des autres types d'appareils des réseaux sans fil, qui incitent une grande préoccupation par rapport à la limite de leurs batteries.

#### ➤ Communication, traitement et stockage

Les véhicules sont caractérisés par la diversification de leurs modes de communications [12], car ils sont équipés de différentes interfaces (WIFI, Bluetooth, Radio et autres). Ceci implique une grande capacité de traitement et de stockage des données, qui est assurée grâce aux nouvelles technologies et les nombreuses puces électroniques très puissantes qui sont installées dans les véhicules.

**➤ Environnement de communication**

Les réseaux VNETs sont caractérisés par la grande diversité de leurs environnements qui sont déployés dans la nature à grande échelle. Passant du milieu urbain qui présente différents obstacles (immeubles) qui peuvent réduire la qualité de transmission radio, à un environnement autoroutier affecté principalement par les très grandes vitesses des véhicules.

**➤ Modèle de mobilité**

Le modèle de mobilité des réseaux VNETs est lié à la diversité environnementale et les infrastructures routières. Mais dans une certaine mesure, il est possible de prévoir l'évolution des déplacements des véhicules grâce à leurs vitesses, leurs directions et surtout la connaissance des cartes routières. Car les déplacements des véhicules sont structurés par les routes et les rues.

Le mode de mobilité des réseaux VNETs est affecté par la vitesse des véhicules et leurs déplacements aléatoires, qui peuvent réduire considérablement les durées de communications et leurs comportements face à des obstacles.

**➤ Modèle de communication**

Les réseaux VNETs ont été déployés principalement pour des raisons de sécurité routière (message d'alerte). Ils doivent relier une source d'alerte vers une multitude de destinations (véhicules ou infrastructures), par exemple, ce type de communication nécessite principalement un modèle de transmission en Broadcast. Néanmoins, les entités ont parfois besoin de communiquer en mode Unicast.

**➤ Topologie et connectivité**

Les réseaux VNETs sont caractérisés par une connectivité irrégulière et relativement faible, liée directement à la vitesse des véhicules, leurs déplacements aléatoires et leurs comportements face à des obstacles, qui peuvent réduire considérablement les durées des communications. En effet un véhicule peut rapidement rejoindre ou quitter un groupe de véhicules, ce qui rend les changements de topologie très fréquents et très dynamiques, constitués de plusieurs groupes séparés, ceci entraîne une réorganisation de la topologie du réseau.

**➤ Technologies de communications**

Pour mettre en place les différentes communications entre les entités du réseau VNET, diverses technologies ont été conçues, pour offrir les différents services et augmenter la portée des communications et des bandes passantes. Ainsi, une norme de communication appelée DSRC (Dedicated short Range Communication) a été adoptée. Sa couche physique est basée sur la norme IEEE 802.11a. Plus tard, l'IEEE s'inspira de cette norme pour créer la norme

actuellement utilisée 802.11p. Cette norme définit essentiellement les services de sécurité et le format des messages.

### **I.3.4. Les type de messages**

Les différents messages échangés dans les réseaux VNETs peuvent être facilement classés, selon leurs utilités et leurs contenus, en deux grands types de message: message lié à la sécurité et message à valeur ajoutée.

#### **I.3.4.1. Message lié à la sécurité**

L'objectif principal de la naissance des réseaux VNETs était la sécurité des usagers de la route. Ainsi on trouve deux types de messages de sécurités :

##### **a) Message beacon**

Ce type de message contient souvent des informations relatives à l'identité et à l'état actuel du véhicule (Position, vitesse, direction et autres), il est diffusé périodiquement et est utilisé principalement pour permettre l'identification du voisinage. Ce type de message joue un rôle primordial dans la plupart des protocoles de routage et de sécurité.

##### **b) Message d'alerte (d'urgence)**

Ce type de message est envoyé pour prévenir les autres véhicules de différentes urgences et des catastrophes sur la route (accident, congestion de la circulation, information météorologique, passage d'un véhicule de secours et autres), afin qu'ils aient plus de temps pour agir. Ce type de message aide à améliorer la circulation et la sécurité routière.

#### **I.3.4.2. Message à valeur ajoutée**

Ce type de message peut contenir n'importe qu'elle autre information ou donnée. Il peut contenir des informations sur des services. Comme l'endroit des restaurants ou des hôtels. Il peut aussi contenir des données multimédias ou n'importe quelle donnée ou information, qui peut améliorer le confort des usagers de la route [12].

### **I.3.5. Applications des réseaux VNET**

Les principales applications des réseaux IVC peuvent être classées en trois catégories [13]: (a) applications de sécurité routière, (b) applications d'aide à la conduite, et (c) applications de confort. Nous détaillons ci-après ces catégories et donnons ensuite des exemples d'applications.

- Applications pour la sécurité routière : la sécurité routière est devenue une priorité dans la plupart des pays développés. Cette priorité est motivée par le nombre croissant d'accidents sur ses routes associé à un parc de véhicules de plus en plus important. Afin d'améliorer la sécurité des déplacements et faire face aux accidents routiers, les IVC offrent la possibilité de prévenir les collisions et les travaux sur les routes, de

détecter les obstacles (fixes ou mobiles) et de distribuer les informations météorologiques;

- ✚ Applications pour les systèmes d'aide à la conduite et les véhicules coopératifs: pour faciliter la conduite autonome et apporter un support au conducteur dans des situations particulières : aide aux dépassements de véhicules, prévention des sorties de voies en ligne ou en virage, etc. Nous pouvons citer également le cas des compagnies de transports utilisant les IVC dans un but de productivité pour réduire la consommation de carburant;
- ✚ Applications de confort du conducteur et des passagers : en particulier les services de communication et d'informations des utilisateurs comme l'accès mobile à l'Internet, la messagerie, le chat inter-véhicules, les jeux en réseaux, etc. Dans la suite de cette partie nous nous limitons à la description de quelques services et exemples d'application des systèmes de communication véhicule à véhicule.

#### **I.3.5.1. Alerter en cas d'accidents**

Ce service permet, dans le cas d'un accident, d'avertir les véhicules se dirigeant vers le lieu de l'accident que les conditions de circulation se trouvent modifiées et qu'il est nécessaire de redoubler de vigilance. Il est nécessaire, également, en cas de densité réduite de véhicule de pouvoir conserver l'information pour pouvoir la retransmettre si un véhicule entre dans la zone de retransmission. Les messages de sécurité devront être émis à des périodes régulières. Ainsi les nœuds désignés pour la retransmission des messages émettront des alertes à instants réguliers. Les messages devront être de taille réduite pour être transmis le plus rapidement possible. Les messages devront comporter les coordonnées du lieu de l'accident et les paramètres de la zone de retransmission.

#### **I.3.5.2. Alerter en cas de ralentissement anormal**

Ce service permet d'avertir les automobilistes de situations de circulation particulières. L'information quel que soit la nature des difficultés de circulation renseigne l'automobiliste qu'il est nécessaire de ralentir.

Le message d'alerte est émis par un véhicule détectant les difficultés de circulation (freinage important par exemple, déclenchement des feux de détresse, pluie). Un véhicule banalisé effectuant des travaux peut également être à l'origine du message d'alerte (Figure I.2). Comme pour le message d'alerte informant d'un accident, le message d'alerte informant d'un ralentissement doit être transmis aux autres véhicules de façon efficace et rapide.

### I.3.5.3. La conduite collaborative

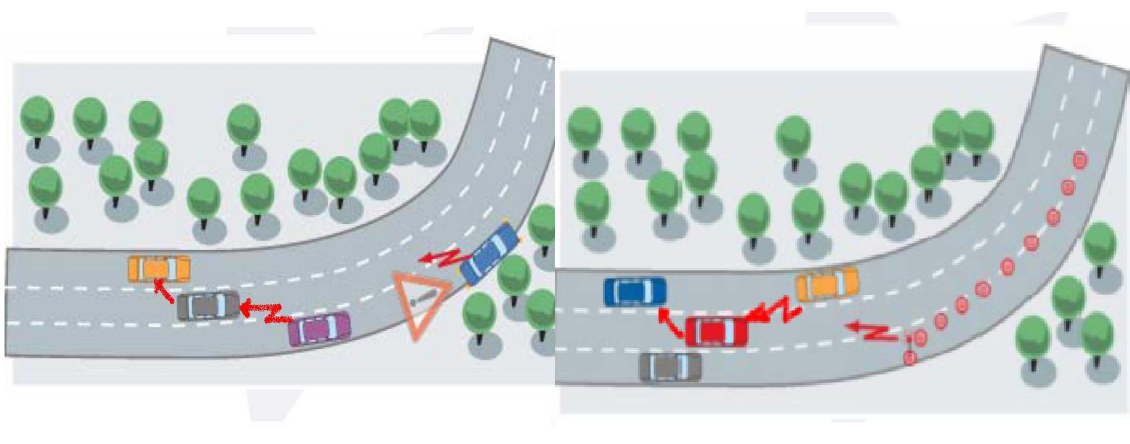
La conduite collaborative est un concept qui améliore considérablement la sécurité du transport routier, en plus de réduire le nombre de victimes lors d'accidents impliquant des véhicules automobiles. Cette innovation est basée sur un échange de renseignements entre des véhicules munis d'instruments (capteurs par exemple) leur permettant de percevoir ce qui les entoure et de collaborer en groupes formés dynamiquement. Ces groupes de véhicules, ou réseaux ponctuels, peuvent élaborer une stratégie de conduite collective qui exigerait peu ou pas d'interventions de la part des conducteurs. Depuis les dernières années, différentes architectures de véhicules automatisés ont été proposées, mais la plupart d'entre elles n'ont pas, ou presque pas, attaqué le problème de communication inter véhicules.

### I.3.5.4. Hot spot sur les autoroutes

Aujourd'hui, les personnes peuvent accéder à des sites web un peu avant qu'ils prennent le train, par exemple pour télécharger des films. En voiture, on peut imaginer d'acheter du contenu, au niveau d'une station essence, d'une gare ou même en pleine autoroute (en passant d'une voiture à une autre jusqu'au point d'accès le plus proche). Les passagers dans la voiture pourront ainsi jouer en réseaux, télécharger des fichiers MP3, envoyer des cartes à des amis, etc.

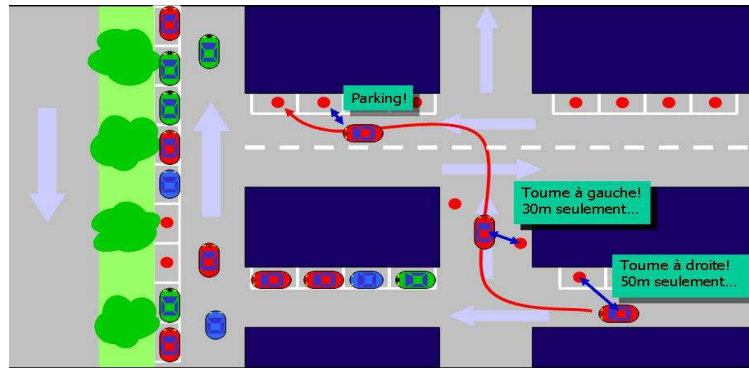
### I.3.5.5. Gestion des espaces libres dans les parkings

Ce service permet de rassembler des informations sur la disponibilité de l'espace de stationnement dans les parkings et de coordonner entre automobilistes afin de les guider aux espaces libres (ex: projet Smart Park) [13].

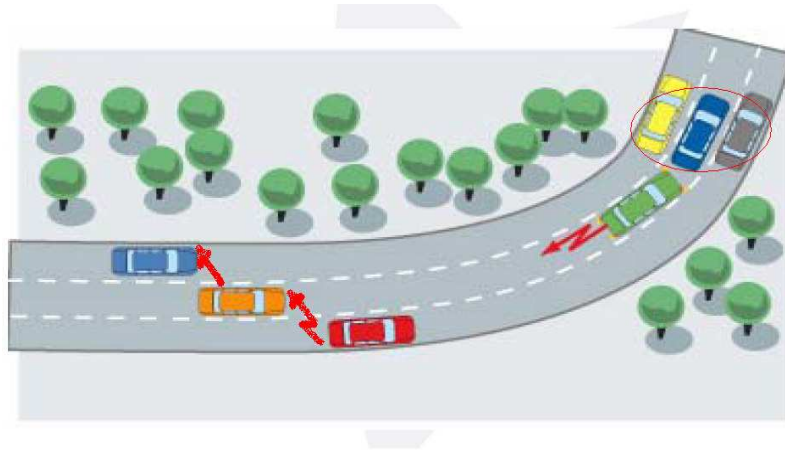


(a) véhicule en panne.

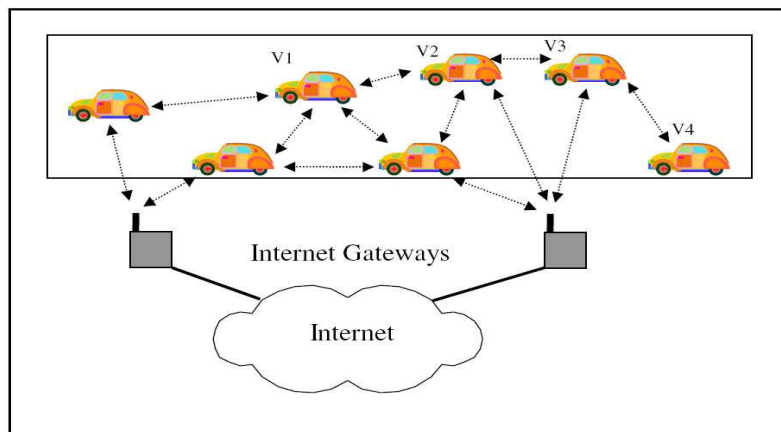
(b) travaux sur les routes.



(c) parking intelligent.



(d) risque de collision.



(e) accès à Internet.

**Figure I. 3.** Exemples de sécurités d'applications des réseaux VNET.

#### I.4. Les principaux types de la collision et les accidents

Accidents de la route est en train de devenir un problème majeur dans notre pays, Les accidents de voiture peuvent se produire en tout temps.

Accidents de la circulation se produit lorsque les véhicules entrent en collision les uns avec les autres, Certains des principaux types d'accidents se trouve sont collisions arrière, les

collisions d'impacts latéraux, retournement du véhicule, accidents de voiture unique, plusieurs véhicules tractions.

**Tableau I. 1.** Les statistiques des accidents de route dans l'Algérie (2011-2019).

<i>Année</i>	<i>Nombre d'accidents</i>	<i>Décès</i>	<i>Blessure</i>
2011	42788	2293	50958
2012	49426	2222	61226
2013	56136	2294	68734
2014	64294	2519	80225
2015	64813	2332	79828
2016	59902	2118	73337
2017	57328	1882	69754
2018	46244	1779	65959
2019	60790	1771	73534

- ❖ **Les collisions arrière :** Ces types d'accidents de la circulation sont souvent causés par une soudaine décélération (ralentissement ou freinage). Dans certains cas, un autre conducteur suit de trop près ou accélère à une vitesse plus élevée que la voiture devant elle. Whiplash (Coup de fouet) est une blessure fréquente qui se produit dans une collision arrière et affecte généralement les conducteurs et les passagers de la voiture touchée.
- ❖ **Collisions latérales :** Collisions latérales peuvent causer des blessures graves. Souvent appelée «T-bone» ou «collisions de broadside», accidents d'impact latéral se produisent lorsque le côté d'un véhicule est touché. Elle peut être influencée par l'avant ou à l'arrière d'un autre véhicule ou dans certains cas, un objet fixe. Les dommages au véhicule sont souvent graves et les conducteurs ou les passagers du côté affecté du véhicule habituellement soutenir des blessures bien pires qu'ils ne le feraient dans un autre type d'accident.
- ❖ **Collisions Sideswipe :** Les collisions se produisent lorsque Sideswipe deux voitures qui sont parallèles contact. Dans de nombreux cas, le dommage est que mineur, que les voitures viennent «chipé» les uns des autres. Les blessures et les dommages sont généralement minimes, à moins que l'un des conducteurs perde le contrôle du véhicule à la suite de la collision.

- ❖ **Retournement du véhicule :** véhicules d'accidents sont extrêmement renversement dangereux et effrayant. Un retournement se produit lorsque le véhicule se retourne littéralement sur le côté ou sur le toit. Tout véhicule peut être impliqué dans un accident de renversement, mais les voitures avec un centre de gravité élevé comme les VUS (véhicules utilitaires sport) sont particulièrement sujettes à ce type d'accident. Souvent causés par des virages serrés à grande vitesse, les accidents de retournement peuvent entraîner des blessures graves, notamment des lésions de la moelle épinière et des traumatismes crâniens.
- ❖ **Les collisions frontales :** Ces types de collisions sont souvent fatals. Les collisions frontales sont exactement ce qu'ils sonnent comme si elles se produisent lorsque les extrémités avant des deux véhicules ont une influence les uns des autres.
- ❖ **Accidents de voiture unique :** Les accidents impliquant un seul véhicule sont également fréquents. Ils se produisent lorsqu'un véhicule heurte un objet tel qu'un poteau, un arbre, une bouche d'incendie ou d'un mur. Dans certains cas, ils peuvent impliquer les piétons et les autres passants innocents. collisions voiture unique peut entraîner des blessures conducteur et passagers, les blessures aux piétons, et souvent d'importants dégâts matériels.
- ❖ **Les collisions de véhicules multiples :** Les collisions de véhicules multiples sont parfois appelés «pileups» et se produisent souvent sur des routes très fréquentées telles que les routes et les autoroutes. Ils peuvent comporter de nombreux véhicules et être le plus dangereux. Les véhicules peuvent être touchés plusieurs fois et il peut être difficile d'échapper. Il est également difficile de déterminer la faute dans ces cas.
- ❖ **Accidents Hit-and-run :** Un accident où un départ du conducteur la scène est connu comme un accident avec délit de fuite. Il peut être très difficile de déterminer l'identité du conducteur ou du véhicule [14].

Dans ces types d'accidents, il y a plusieurs méthodes pour l'éviter, dans notre travail nous avons basé sur un système de communication entre les véhicules (VNET) qui dépend de la communication sans fil (Chapitre II), cette communication est très utile pour suivre les mouvements d'un véhicule est basé sur un échange d'information entre véhicule-véhicule et véhicule-infrastructure en temps réel, et peut-être plus facile de réduire et éviter la collision et aussi de déclenché une alerte si la situation est risque. On trouve aussi un système d'intelligence artificielle (Chapitre III).

## **I.5. Conclusion**

Depuis quelques années, le développement des nouvelles technologies a favorisé une évolution du système de transport. Cette évolution vise à rendre le transport plus sûr, plus efficace, plus fiable et plus écologique, sans avoir nécessairement à modifier matériellement l'infrastructure existante. La gamme des technologies en jeu comprend l'informatique, les technologies de capteurs, les systèmes de contrôle et télécommunications.

Le développement des nouvelles technologies de communications et l'avancement des applications des STI ont favorisé un excellent stade pour l'évolution des réseaux véhiculaires. Dans ce chapitre nous avons donné un aperçu sur les réseaux de communication entre les véhicules. Par la suite, nous avons tout d'abord identifié les applications développées dans un réseau sans fil véhiculaire, et les avons classés en trois catégories : confort, gestion du trafic routier et sécurité du trafic routier. En raison de la criticité des messages envoyés par les applications de sécurité routière, nous nous intéressons à cette catégorie. En effet, ces applications ont des contraintes temporelles fortes et un impact direct sur la vie des usagers. Nous nous focalisons plus particulièrement sur l'application d'alerte de danger local qui diffuse des messages d'alerte.

## *Chapitre II*

---

### *Communication sans fil*

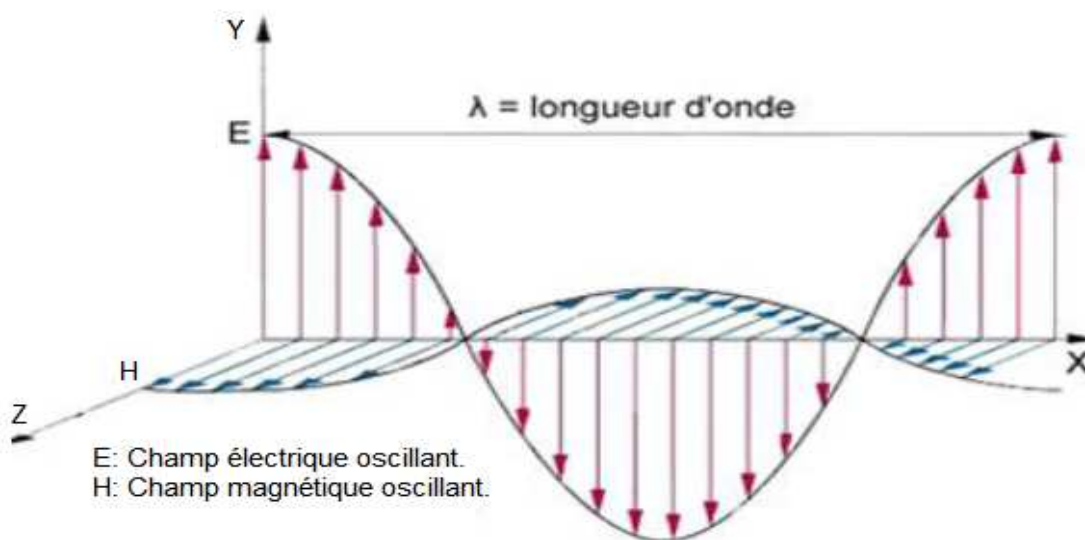
## II.1. Introduction

Les communications sans fils avec leurs multiples architectures représentent une solution vers des systèmes de communication entre les véhicules et aussi entre les véhicules et les infrastructures. Ces technologies ont pour vocation d'améliorer la sécurité routière avec la prévention des collisions, la signalisation des accidents et d'échanger des informations entre les véhicules.

Dans ce chapitre, nous avons présenté les technologies de capteurs sans fils tels la radiofréquence, GSM (Global System for Mobile Communications), et le système de positionnement GPS (Global Positioning System). Ces systèmes permettent également de développer et d'évaluer des applications de communication entre les véhicules. Ensuite, nous présenterons l'architecture d'un nœud capteur basé sur une communication sans fil et une étude technologique du matériel, ainsi l'état de l'art des nœuds capteurs existants commercialisés par différents constructeur. Ces types de capteurs seront par la suite utilisés dans notre travail.

## II.2. Communication radiofréquence

L'onde électromagnétique ou rayonnement est constitué(e) des champs électrique ( $E$ ) et magnétique ( $H$ ) dont l'alternance dans le temps assure la propagation dans l'espace par un phénomène d'induction mutuelle (Figure II.1). Selon la distance à l'antenne émettrice et selon les dimensions relatives de celle-ci par rapport à la longueur d'onde du rayonnement [15].



**Figure II. 1.** Schéma d'une onde électromagnétique plane se propageant dans l'axe X. Les composantes électrique ( $E$ ) et magnétique ( $H$ ) de l'onde sont représentées dans les axes Y et Z.

Les radiofréquences (RF) sont des ondes appartenant au spectre électromagnétique et qui s'étendent, en termes de fréquence, d'une centaine de *kHz* à une dizaine de *GHz*. Cette bande de fréquence est utilisée par les antennes radiofréquence (exemple : module nRF24, avec une fréquence de *2.4GHz*), téléphones portables, le Wi-Fi, les fours micro-ondes, etc. Une onde RF est, comme toute onde électromagnétique, composée d'un champ électrique et magnétique qui oscillent en phase, perpendiculaire l'un par rapport à l'autre et perpendiculaire à la direction de propagation (Figure II.1).

### II.2.1. Radiofréquence dans les systèmes VNET

Diverses technologies de communication sans fil sont proposées pour les systèmes VNET:

- ✚ -Des communications à courte portée comme le Wi-Fi ou comme les modules radiofréquence (nRF).
- ✚ Des communications à plus longue portée comme le GSM ou la technologie 3G.

### II.3. Communication GSM

Le GSM (Global System for Mobile communications), est un système cellulaire et numérique de télécommunication mobile. Il a été rapidement accepté et a vite gagné des parts de marché. L'utilisation du numérique pour transmettre les données permet des services et des possibilités élaborées par rapport à tout ce qui a existé.

Actuellement le réseau GSM en ALGERIE compte des millions d'utilisateurs. Le <<portable>> est devenu un produit de consommation courante. Ce petit trésor de technologie ouvre la porte à de nombreuses applications électronique sans fil à celui qui sait l'interfacer avec un PC ou un microcontrôleur. Il devient possible via l'envoi et la réception de SMS piloter et surveiller un processus quelconque. La distance n'est désormais plus un souci puisque le réseau GSM couvre la plus part des territoires nationale et mondiale.

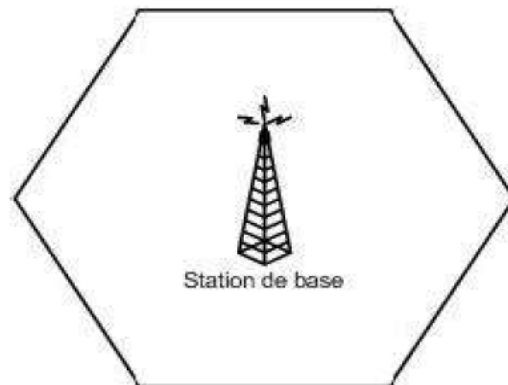
L'architecture fonctionnelle du GSM peut être divisée entre la Station Mobile (Mobile Station - MS), la Station de Base (Base Station - BS) et le sous-système réseau (Network Subsystem- NS). Le MS est gérée par l'utilisateur, le sous-système BS contrôle la liaison radio avec le MS, et le NS commutent les appels entre le mobile et les autres qui sont fixes ou les utilisateurs du réseau de mobiles ainsi que la Mobility Management.

#### II.3.1. Notion de réseaux cellulaires

Dans un réseau GSM, le territoire est découpé en petites zones appelées **cellules**.

Chaque cellule est équipée d'une **station de base** fixe munie de ses antennes installées sur un point haut (château d'eau, immeuble,...).

Les cellules sont dessinées hexagonales mais la portée réelle des stations dépend de la configuration du territoire arrosé et du diagramme de rayonnement des antennes d'émission. Dans la pratique, les cellules se recouvrent donc partiellement [16].

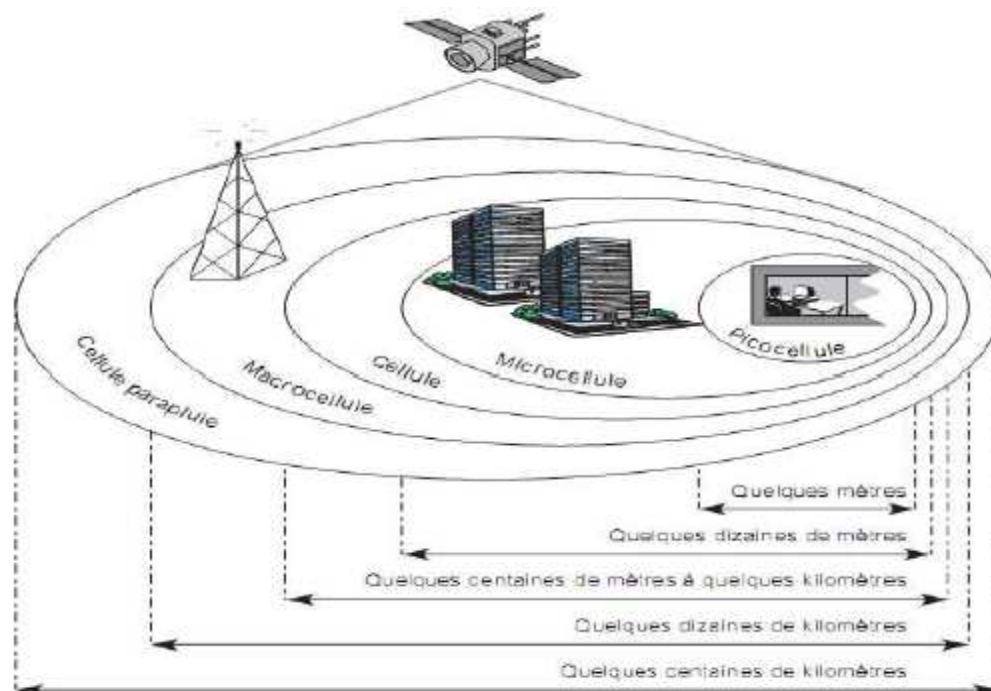


**Figure II. 2.** Une cellule.

### II.3.2. Taille des cellules

On distingue quatre niveaux hiérarchiques de cellules [17]:

- Pico-cellule couvrant une petite surface comme l'intérieur d'un bureau
- Micro-cellule couvrant la surface d'une petite cité
- Macro-cellule pouvant avoir une couverture de plusieurs kilomètres
- Cellule globale couvrant une région pouvant atteindre le tiers du globe grâce aux satellites.

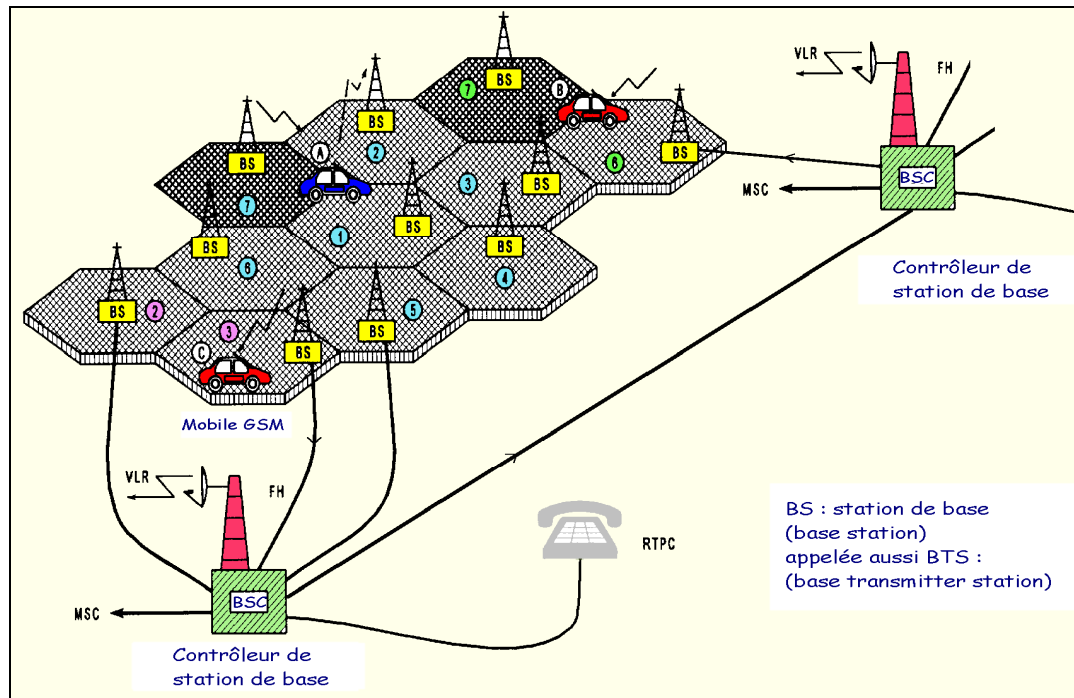


**Figure II. 3.** Structure hiérarchique des cellules.

### II.3.3. Structure du réseau GSM

Lorsqu'on téléphone à partir d'un mobile GSM:

- Le mobile transmet par radio la communication vers la station de base de sa cellule.
- La conversation est ensuite acheminée de façon plus classique (câble, fibre optique ...) vers le correspondant s'il est raccordé au réseau téléphonique filaire, ou à sa station de base s'il est équipé d'un mobile.
- Cette station de base transmet finalement la conversation par radio au correspondant.



**Figure II. 4.** Structure cellulaire du réseau GSM.

Même si deux personnes se trouvent dans la même cellule et se téléphonent, la conversation ne passe jamais directement d'un GSM à l'autre.

Au cours d'un déplacement, il est possible qu'on sorte d'une cellule. Il est nécessaire alors de changer de station de base tout en maintenant la communication : c'est le **transfert intercellulaire** ou **handover**.

Pour gérer ce transfert :

- Le téléphone GSM mesure en permanence la force du signal radio reçu de la station de base et écoute aussi régulièrement les stations de base des cellules voisines
- Lorsqu'il constate qu'il reçoit mieux une autre station de base que celle avec laquelle il échange les signaux, il en informe sa station de base
- La station de base décide alors de passer le relais à la station de base voisine et met en œuvre la procédure de handover.

Ce processus oblige tous les mobiles GSM à écouter les stations de base des cellules voisines en plus de la station de base de la cellule dans laquelle il se trouve.

### II.3.4. Les équipements du réseau GSM

Les fonctions mises en œuvre dans le réseau GSM sont celles requises dans tout réseau de mobiles comme la numérotation, l'acheminement vers un usager mobile, le transfert de cellules. etc... [16].

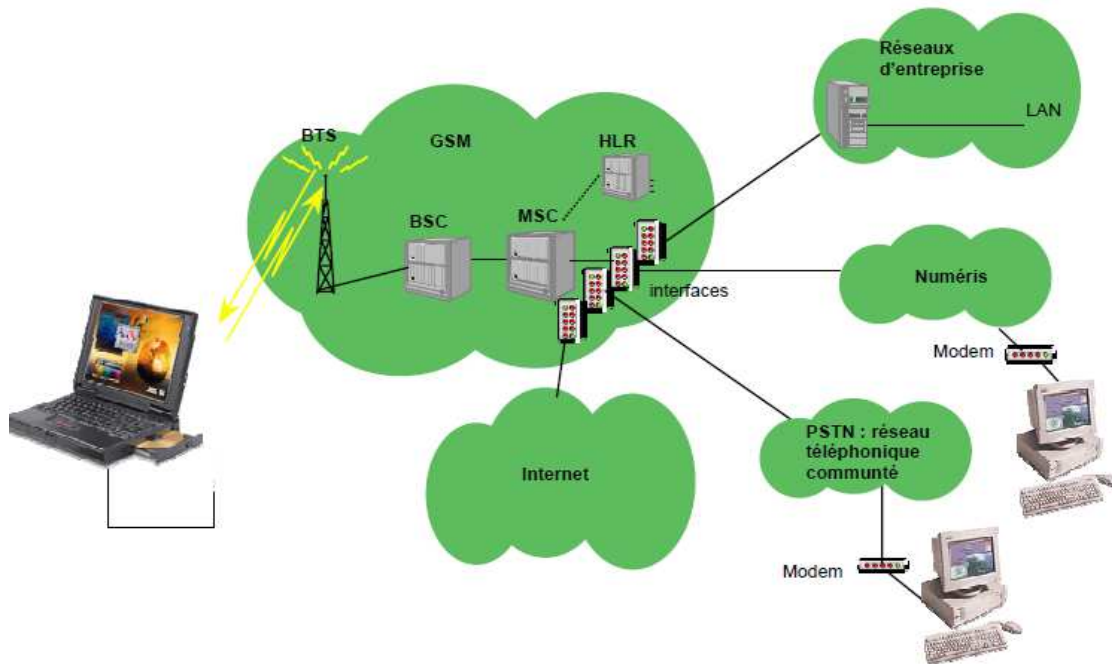


Figure II. 5. Structure générale d'un réseau GSM.

Le téléphone GSM ou **station mobile** est caractérisée par deux identités :

- Le numéro d'équipement, **IMEI** (International Mobile Equipment Identity) mis dans la mémoire du mobile lors de sa fabrication.
- Le numéro d'abonné **IMSI** (International Mobile Subscriber Identity) se trouvant dans la carte **SIM** (Subscriber Identity Module) de l'abonné.
- Le système de communication radio est l'équipement qui assure la couverture de la cellule et comprend :
  - Les stations de transmission de base **BTS** (Base Transmitter Station).
  - Le contrôleur de stations de base **BSC** (Base Station Controller) qui gère entre 20 et 30 BTS et possède son registre d'abonnés visiteurs **VLR** stockant les informations de l'abonné liées à sa mobilité.

Le commutateur de services mobiles **MSC** est un autocommutateur qui assure les fonctions de commutation nécessaires en aiguillant les conversations vers la MSC du correspondant ou

vers d'autres réseaux (téléphonique, Internet, Numéris ...) à travers des interfaces appropriées.

Le registre des abonnés nominaux ou **HLR** (Home Local Register) est une base de données utilisée pour la gestion des abonnés mobiles et contenant deux types d'informations :

- Les informations d'abonnés, le numéro d'abonné (IMSI).
- Les informations sur la localisation de l'abonné, permettant aux appels entrant dans le réseau d'être acheminés jusqu'à ce mobile.

### **II.3.5. GSM dans les systèmes VNET**

Dans un système VNET on trouve un système d'alerte communicant qui est basé sur la technologie du GSM ce dernier doit transmettre des messages d'avertissement en cas de détection d'une situation critique comme par exemple une panne de véhicule, une forte densité de trafic et des conditions dangereuses sur les routes. Les systèmes d'alerte existants sont constitués de deux composantes, la détection et la reconnaissance de l'obstacle. Ils sont basés sur des technologies des capteurs et des dispositifs informatiques installés dans les véhicules ainsi que sur l'infrastructure routière.

## **II.4. Système de localisation GPS**

Le système de localisation GPS est un système de positionnement global. Il est conçu pour fonctionner dans toutes les conditions météorologiques. Il permet d'obtenir précisément la vitesse et la position tridimensionnelle d'un récepteur quelque soit sa localisation, de façon continue et instantanée [18]. Le système de localisation GPS est constitué de trois segments : le segment spatial, le segment de contrôle et le segment utilisateur.

### **II.4.1. Différents segments du système GPS**

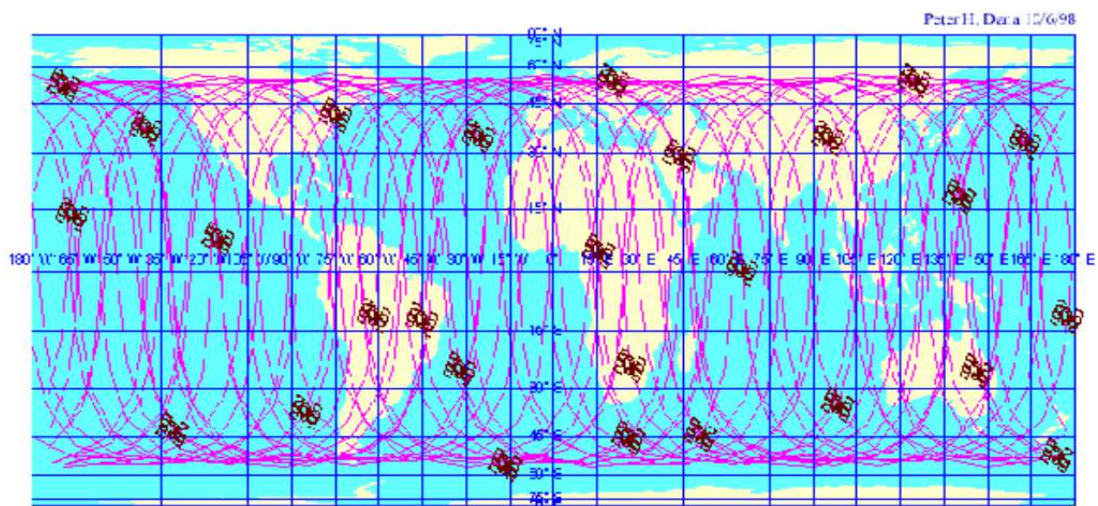
#### **II.4.1.1. Segment spatial**

Le segment spatial du système de navigation GPS est constitué par la constellation (voir Figure II. 6) comprend 24 satellites opérationnels qui gravitent autour de la terre avec une période orbitale de 12 heures. Les 24 satellites sont repartis sur 6 plans orbitaux avec une inclinaison multiple de  $55^\circ$  par rapport à l'équateur terrestre [19].



**Figure II. 6.** La constellation de satellites du système de navigation GPS.

La figure suivante représente un exemple des traces au sol sur 24 heures des satellites:



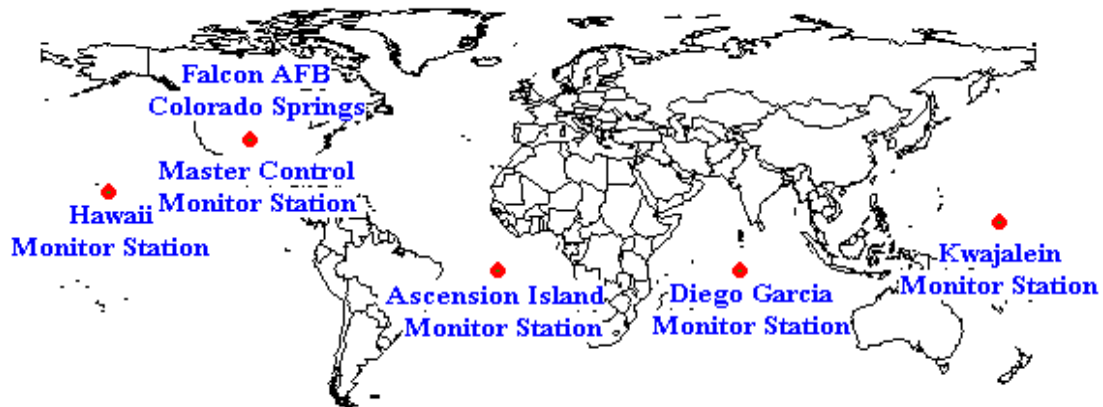
**Figure II. 7.** Exemple de positionnement des 27 satellites lors d'une orbite terrestre.

Chaque orbite contient 4 satellites espacés de  $60^\circ$  (formant un angle de  $55^\circ$  avec l'équateur) et sont à une altitude de 20180 (km) sur laquelle chaque satellite effectue, approximativement, 2 rotations en 24 heures. La disposition spatiale des satellites est conçue de telle sorte qu'en chaque point de la terre, plus de 4 satellites sont en vue. Au minimum quatre satellites sont nécessaires pour avoir un positionnement en trois dimensions (en général longitude, latitude et altitude) et la vitesse d'un récepteur.

#### II.4.1.2. Segment de contrôle

En ce qui concerne le deuxième type de segment, il s'agit d'un réseau de 5 différentes stations de poursuite de la composante de contrôle au sol qui ont pour principale fonction de suivre toutes les secondes la trajectoire des satellites GPS. La station principale gère toutes les

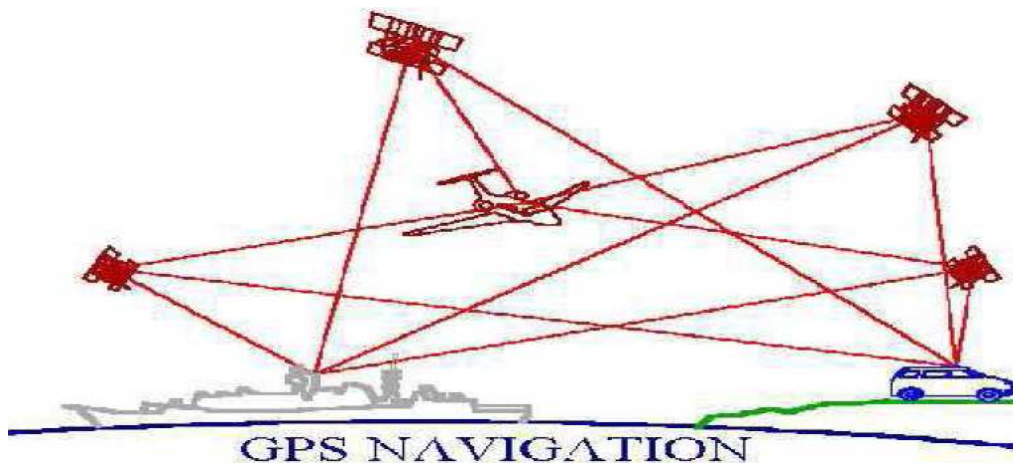
transmissions ainsi que le calcul des éphémérides et les corrections des horloges satellite qui sont régulièrement communiquées aux satellites, à leur tour, les transmettent aux utilisateurs. Le récepteur GPS utilise l'ensemble de ces informations pour calculer et corriger sa position. La Figure II.8 illustre les cinq différentes stations de surveillance dispersées dans le monde dont une station maître qui calcule les corrections à apporter aux messages des satellites (Hawaï, Ascension, Diego Garcia, Kwajalein, Colorado Springs).



**Figure II. 8.** Le segment de contrôle.

#### II.4.1.3. Segment utilisateur

La partie utilisateur comprend l'ensemble des récepteurs passifs civils et militaires (voir figure II.9). En utilisant des informations transmises par les satellites en vue pour se positionner, le récepteur est capable de calculer sa position, sa vitesse et le temps universel. Il faut, au minimum, 4 satellites en vue pour qu'un récepteur puisse recevoir quatre signaux nécessaires pour calculer la position en 3D et le temps. Il faut noter que l'utilisation des signaux GPS est gratuite et nous ne payons que les récepteurs qui sont de plus en plus petits et bon marché [20].

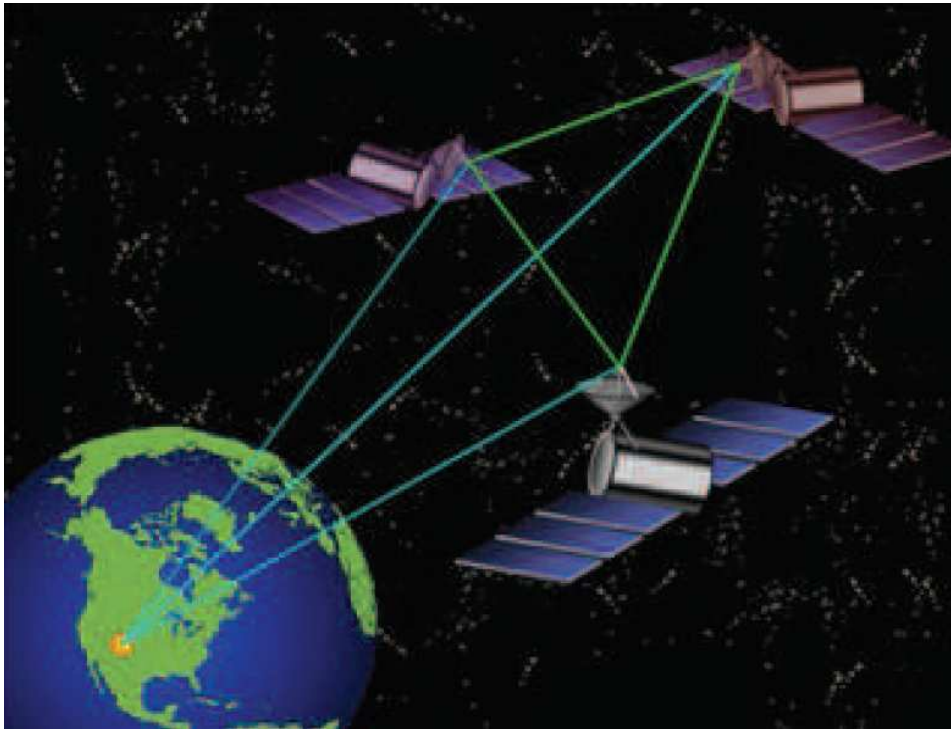


**Figure II. 9.** Segment utilisateur.

## II.4.2. Principe de fonctionnement

### II.4.2.1. Mesure de la position

Le positionnement GPS est basé sur le principe de la triangulation (voir figure II.10) : un lieu géométrique est déterminé par l'intersection de trois sphères dans l'espace. Dans le cas des systèmes GPS, chaque sphère est déterminée par son centre qui a la position d'un satellite dans l'espace et son rayon donné par la distance entre le récepteur et le satellite.



**Figure II. 10.** Calcul de position à l'aide de la méthode de trilatération.

La position de chaque satellite étant véhiculée par les messages de navigation et peut être décodée par le récepteur GPS. La distance entre l'antenne du récepteur et le satellite est calculée à partir du temps de vol du signal GPS. De cette manière et avec l'utilisation de trois satellites, le récepteur GPS peut déterminer sa position horizontale.

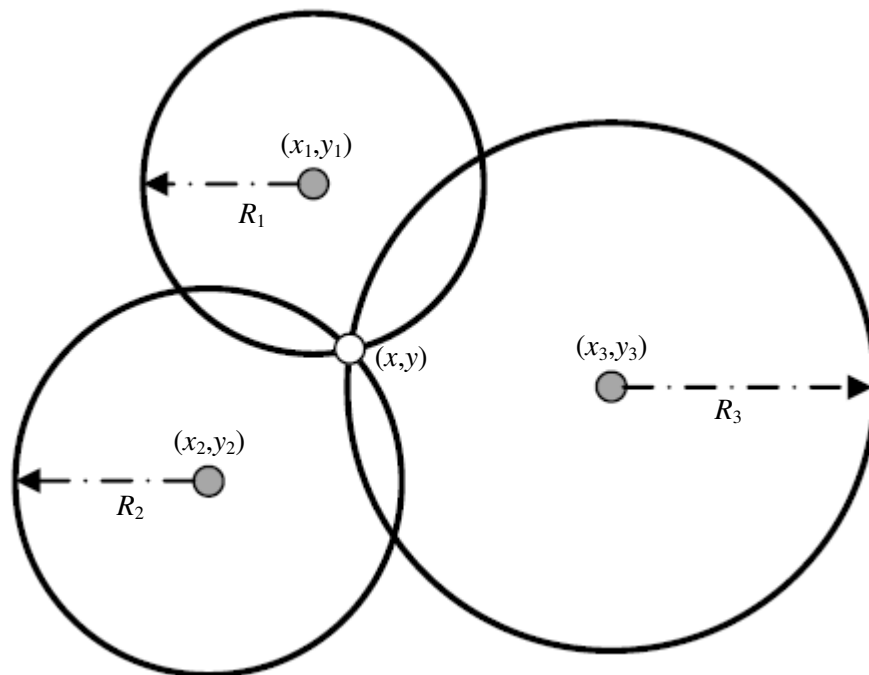
La trilatération est la méthode la plus basique et intuitive. Cette méthode calcule la position d'un nœud par l'intersection de trois cercles, comme le montre la Figure II.11. Cette méthode est basée sur la résolution d'un système de trois équations à trois inconnues par rapport à trois satellites de positions connues (équations II.1, II.2, II.3). Par exemple, si on considère que la position inconnue d'un récepteur en deux dimensions (2D) est  $(x, y)$  et que les coordonnées des trois satellites sont  $(x_i, y_i, i = 1, 2, 3)$ , nous obtenons le système de trois équations à résoudre ci-dessous :

$$R_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (\text{II.1})$$

$$R_2^2 = (x - x_2)^2 + (y - y_2)^2 \quad (\text{II.2})$$

$$R_3^2 = (x - x_3)^2 + (y - y_3)^2 \quad (\text{II.3})$$

ou  $R_i$ : La distance entre le nœud à localiser et le  $i^{\text{ème}}$  satellites.



**Figure II. 11.** Principe de la trilatération.

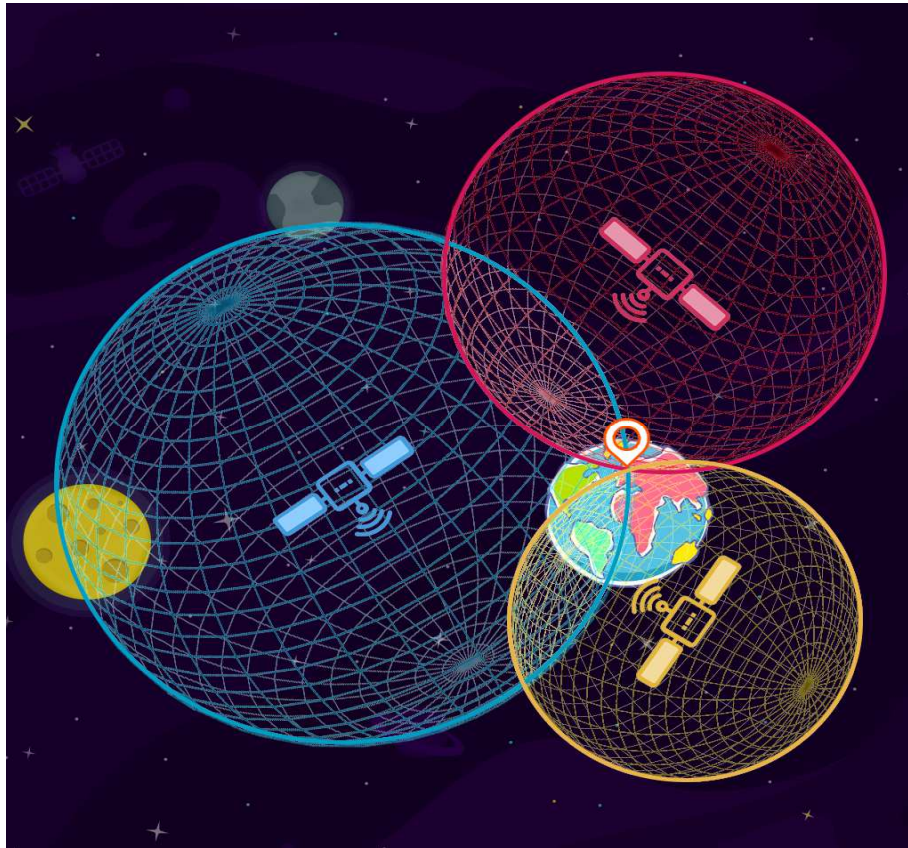
En soustrayant l'équation (II.3) des deux premières équations, on obtient un système linéaire avec deux équations à deux inconnus  $(x, y)$  qui peut donner une seule solution [21]:

$$x(x_3 - x_1) + y(y_3 - y_1) = \frac{1}{2}(x_3^2 - x_1^2 + y_3^2 - y_1^2 + R_1^2 - R_3^2) \quad (\text{II.4})$$

$$x(x_3 - x_2) + y(y_3 - y_2) = \frac{1}{2}(x_3^2 - x_2^2 + y_3^2 - y_2^2 + R_2^2 - R_3^2) \quad (\text{II.5})$$

Dans ce système, les distances  $R_i$  sont calculées par le temps de vol ou la durée du trajet reliant le satellite au récepteur ( $T_i$ ) [ $R_i = c \cdot T_i$ ,  $c \approx 3 \cdot 10^8 \text{ m/s}$ ] que met le signal GPS pour faire le trajet entre le récepteur et les satellites. Si le temps  $T_i$  est mesuré avec précision, la résolution de ce système suffit pour déterminer la position du récepteur. Cependant, en réalité, le temps de vol est entaché d'erreur de synchronisation entre l'horloge du récepteur et celle du satellite. Cette incertitude sur le temps est considérée comme une quatrième inconnue, d'où la nécessité d'une quatrième équation (donc d'un quatrième satellite) pour qu'un récepteur puisse déterminer sa position. C'est pour cette raison que le nombre minimal de 4 satellites, est fixé pour le positionnement 2D.

La figure suivante représente un exemple du positionnement en 3D.



**Figure II. 12.** Positionnement en 3D.

Si plus de quatre satellites sont observés, la précision et la fiabilité du positionnement seront plus élevées. La position fait référence à un système de coordonnées géocentriques. Les coordonnées tridimensionnelles obtenues sont exprimées dans le système de coordonnées utilisé pour le calcul des positions des satellites. Il est à noter que ce système de coordonnées à des difficultés de localisation provient de la perturbation ou de l'interruption causée par les obstacles (bâtiments, rochers, tunnels, interférences électroniques, forêts, conditions météorologiques,...) [22].

#### **II.4.2.2. Mesure de la vitesse**

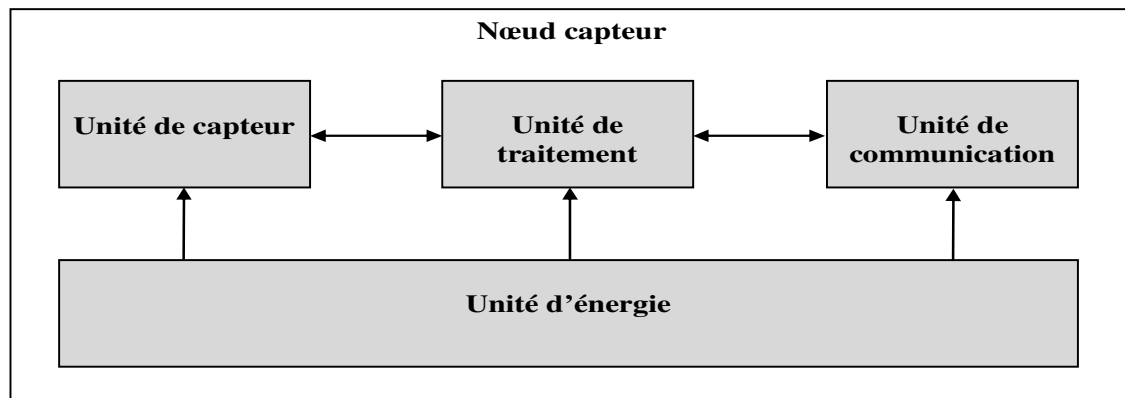
Le système GPS permet de déterminer la vitesse à laquelle se déplace l'utilisateur. Cette évaluation fait par mesure de doppler le signal provenant d'un satellite GPS. En effet, le signal perçu par le récepteur GPS n'a pas exactement la même fréquence que lorsqu'il est généré par le satellite. Suivant le principe Doppler, le rapport des fréquences est en fonction des positions et des vitesses relatives de satellite et de l'utilisateur. Si  $f$  est la fréquence du signal généré dans le satellite et  $f'$  la fréquence à laquelle il est perçu par le récepteur, on a  $f' = f$ , la vitesse du satellite est calculé à partir de l'éphéméride transmise et les coordonnées sont calculées lors de la détermination de la position.

### II.4.3. Localisation dans les systèmes VNET

Les systèmes de positionnement (i.e. récepteur GPS) constituent la principale solution pour localiser les véhicules. Il s'agit d'une solution raisonnable car les récepteurs GPS peuvent être facilement installés dans les véhicules. Les récepteurs GPS ont un niveau raisonnable de précision pour la plupart des applications. De plus, le fonctionnement du GPS n'est pas toujours robuste [19]. Il est fonction de l'environnement et de la visibilité des satellites entraînant des problèmes dans le fonctionnement des systèmes de communication entre les véhicules VNET.

### II.5. Architecture d'un nœud capteur

Les quatre composants principaux d'un nœud capteur sont représentés suivant l'architecture normalisée de la figure II.13:



**Figure II. 13.** Architecture d'un nœud capteur.

Pour effectuer un choix judicieux du matériel composant un nœud capteur, il est essentiel de passer par une étude technologique de ce dernier.

Nous commencerons par l'unité de traitement, ensuite l'unité de communication. Pour l'unité d'énergie, nous avons utilisé une alimentation à courant continu. Concernant les unités de capture, elles dépendent du type d'application, dans le chapitre IV, nous présenterons les capteurs utilisés dans notre travail.

#### II.5.1. Unité de traitement

Le choix de l'unité de traitement est fondamental dans la conception d'un nœud capteur vu qu'elle joue un rôle important dans la gestion de consommation d'énergie du nœud et dans la vitesse d'exécution de l'algorithme de commande du réseau et comporte la mémoire programme qui doit être assez grande pour pouvoir implémenter des programmes assez volumineux et complexe.

Parmi les différentes unités de traitement existantes pouvant être utilisées dans un réseau de capteur sans fil, nous présenterons dans la suite deux types des cartes essentielles dans notre travail : Arduino, Raspberry Pi:

### II.5.1.1. Carte Arduino

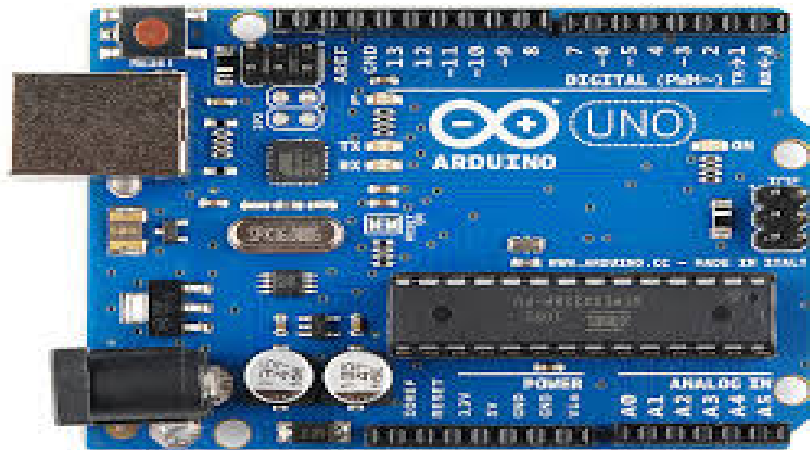
Le système Arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, pour programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le cout de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique [6].

Ce système est constitué de deux parties (matériel et logiciel):

#### II.5.1.1.1. Partie Hardware (Le matériel)

Les cartes électroniques dont les schémas sont en libre circulation sur internet. Dans la section suivante, nous détaillons deux types de cartes Arduino (UNO, et Mega).

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. Sa mémoire de 32KB et son grand nombre d'E/S font de ce circuit compatible. Ce type de carte est un élément idéal pour les systèmes embarqués ou pour notre projet qui nécessite des applications multitâches. L'Arduino Uno peut se programmer avec le logiciel IDE Arduino.



**Figure II. 14.** Carte Arduino Uno.

Les spécifications techniques de cette carte Arduino Uno sont les suivantes :

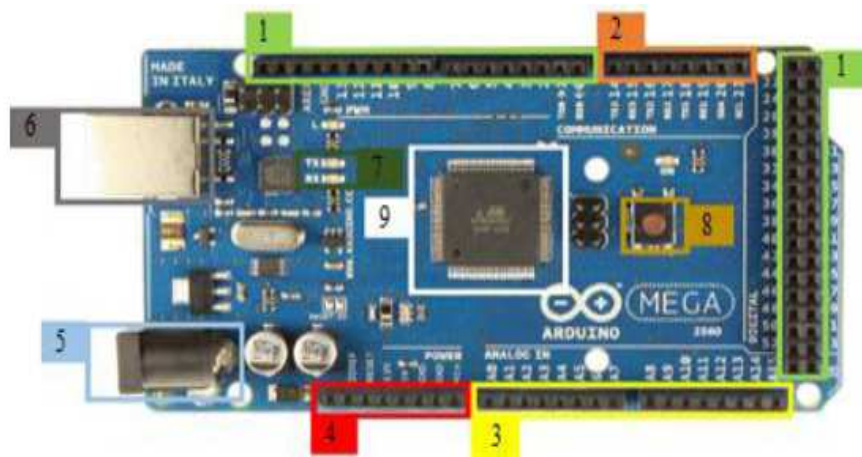
- ❖ Microcontrôleur Atmel ATmega328.
- ❖ Voltage opérationnel (au niveau logique) : 5V.
- ❖ Pins d'entrées/sorties digitales : 14 (dont 6 disposent d'une sortie PWM).
- ❖ Pins d'entrée analogique : 6 (utilisables en broches E/S numériques).
- ❖ Courant direct par pin d'entrée/sortie : 40mA.
- ❖ Mémoire Flash : 32KO (ATmega328) dont 2KO sont utilisés par le bootloader.

- ❖ SRAM : 2KO (ATmega328).
- ❖ EEPROM : 1KO (ATmega328).
- ❖ Vitesse d'horloge : 16MHz.

La carte Arduino UNO peut être alimentée par l'USB ou par une alimentation externe. La source est sélectionnée automatiquement.

La tension d'alimentation extérieure (hors USB) peut venir soit d'un adaptateur AC-DC ou de piles. L'adaptateur peut être connecté grâce à un 'jack' de 2.1mm positif au centre. Le raccordement vers un bloc de piles peut utiliser les bornes GND et *Vin* du connecteur d'alimentation (POWER). La carte peut fonctionner à l'aide d'une tension extérieure de 7 à 12volts.

Le schéma qui suit montre les différents composants de la carte Arduino Mega2560 :



**Figure II. 15.** Les composants d'Arduino Mega 2560.

1. Entrées/Sorties digitales (54 pin).
2. Pin de communication.
3. Entrées/Sorties Analogique (peuvent aussi servir comme E/S digitales, 16 pin).
4. Broches d'alimentation pour le montage (5V, 3,3V, GND...).
5. Entrée d'alimentation externe pour la carte.
6. Port USB pour la communication entre le PC et la carte.
7. LED indiquant la communication avec l'ordinateur (Tx, Rx)
8. Botton RESET.
9. Microcontrôleur ATmega2560.

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino (IDE) peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

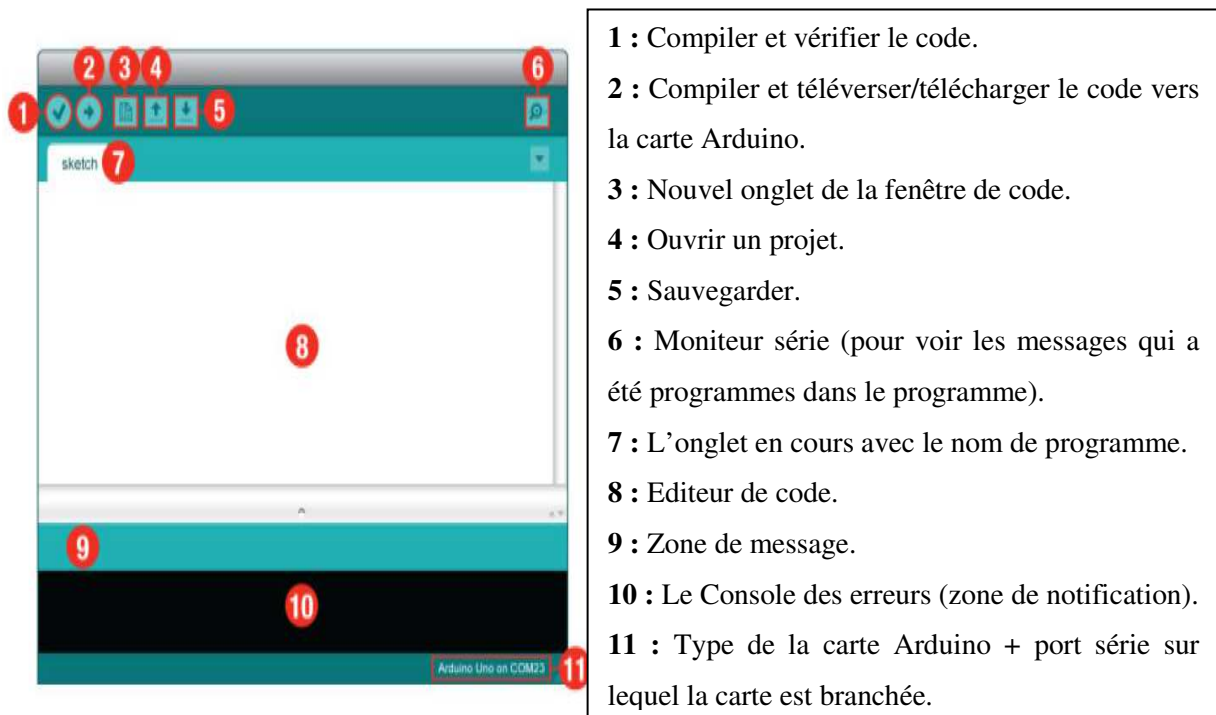
### II.5.1.1.2. Partie logiciel (Software)

Le logiciel IDE Arduino est gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

Pour commander un Arduino sur PC on installe le logiciel IDE Arduino (par exemple la version 1.6.5 dans notre cas) qui est une application Java et qui a l'interface montrée dans la figure II.16. IDE Arduino est une interface souple et simple est exécutable sur n'importe quel système d'exploitation. Arduino basé sur la programmation en C ou C++ [6].

Le logiciel Arduino a pour les fonctions principales :

- De pouvoir écrire et compiler des programmes pour la carte Arduino.
- De se connecter avec la carte Arduino pour y transférer les programmes.
- De communiquer avec la carte Arduino.



**Figure II. 16.** Description de l'Interface Arduino IDE.

La structure d'un programme IDE Arduino est comporte trois parties [6] :

- (1) La partie déclaration des variables (optionnelle).
- (2) La partie initialisation et configuration des entrées/sorties : la fonction **setup ()**.
- (3) La partie principale qui s'exécute en boucle : la fonction **loop ()**.

```

Fichier  Édition  Croquis  Outils  Aide
DigitalReadSerial $
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);      // delay in between reads for stability
}
11
Arduino/Genuino Uno on COM14

```

Figure II. 17. Structure d'un programme Arduino.

## II.5.1.2. Carte Raspberry Pi

### II.5.1.2.1. Partie Hardware

La carte Raspberry Pi 3 (Figure II.18) est une ordinateur mono-carte pouvant se connecter à un téléviseur, à un clavier et disposant d'une connectivité Wi-Fi et Bluetooth, possède 4 ports USB, un port micro-SD, un connecteur d'E/S 40 broches et un port HDMI. Cette version est physiquement identique à la version Pi 2 la rendant compatible avec les boîtiers et autres accessoires de la version Pi 2. La version 3 est basée sur un processeur ARM Cortex-A53 64 bits quatre cœurs à 1,2 GHz (environ 10 fois plus rapide que le Pi 1 et 50% plus performante que le modèle Pi 2) et possède 1 GB de mémoire RAM [6].

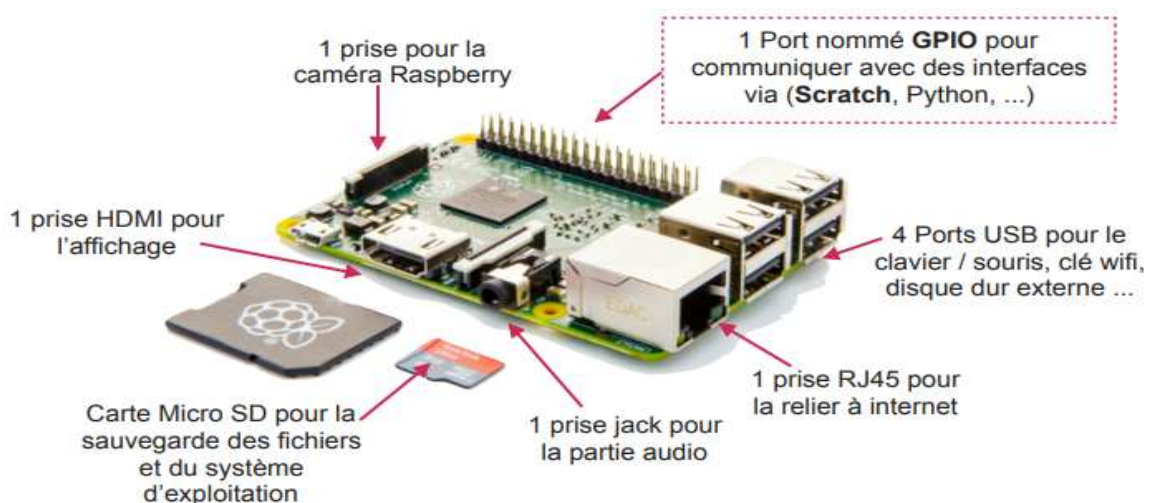
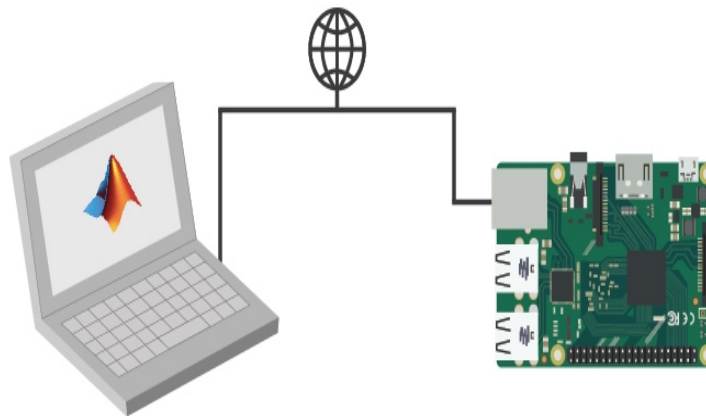


Figure II. 18. La carte Raspberry Pi 3.

### II.5.1.2.2. Partie logiciel

Le support package MATLAB-SIMULINK pour Raspberry *Pi* permet d'écrire des programmes MATLAB qui communiquent avec la carte Raspberry *Pi* et acquièrent des données à partir des ports *GPIO* de la carte, des caméras et autres appareils connectés. MATLAB étant un langage interprété de haut niveau, il permet de facilement prototyper et affiner des algorithmes pour notre projet basée sur le Raspberry *Pi*. MATLAB inclut des milliers de fonctions mathématiques et de tracés que nous pouvons utiliser dans le cadre de la programmation de Raspberry *Pi*, couvrant des domaines tels que le traitement vidéo et le traitement d'image, l'optimisation, les statistiques et le traitement du signal.



**Figure II. 19.** Connexion ordinateur et Raspberry *Pi*.

Avec le support package MATLAB pour Raspberry *Pi*, la carte Raspberry *Pi* est connectée à un ordinateur exécutant MATLAB. Le traitement est effectué sur l'ordinateur avec MATLAB.

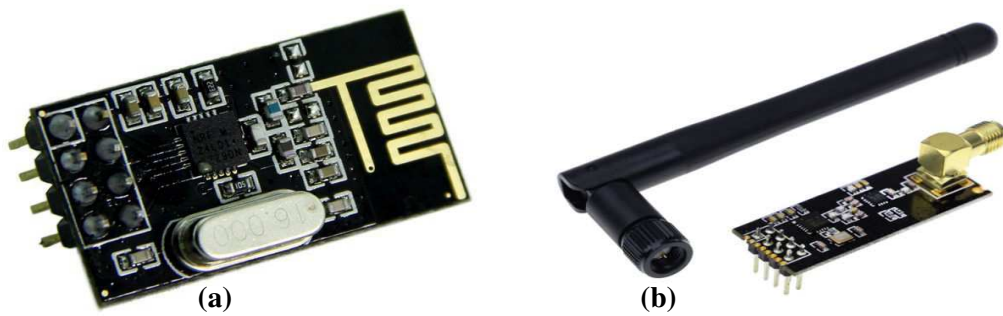
## II.5.2. Les unités de communication

L'unité de communication comporte trois modules : radiofréquence, GSM, GPS. Pour chaque type il faut choisir les liaisons ou les composants de communication séries compatibles avec l'unité de traitement.

### II.5.2.1. Module radio fréquence

Dans notre travail, nous avons utilisé le module nRF24L01, ce type de famille RF24 (avec et sans antenne) est des émetteurs/récepteurs qui utilisent la bande de fréquence 2.4 *Ghz*. Ils possèdent une antenne intégrée et fonctionnent en modulation *GFSK* avec 125 canaux possibles.

La vitesse de fonctionnement maximale est de 2 *Mbps* (Mega bits par secondes), avec une faible consommation électrique. S'il est utilisé dans un espace ouvert et avec un débit inférieur, sa portée peut atteindre 100 mètres [23].



**Figure II. 20.** (a) Module nRF24L01 (b) Module nRF24L01-PA-LNA.

Une liaison SPI (Serial Peripheral Interface) s'établit entre un maître et un esclave, ce qui permet d'avoir un échange de données par série entre ces derniers. L'échange de données peut s'effectuer dans les deux sens. Cette liaison s'effectue par l'intermédiaire de trois fils notés MOSI (Master Output Slave Input), MISO (Master Input Slave Output) et SCK (SPI Serial Clock).

La consommation d'énergie de ce module est d'environ 12 mA pendant la transmission, ce qui est encore plus bas qu'une seule LED. La tension de fonctionnement du module est de 1.9 à 3.6V, mais la bonne chose est que les autres broches tolèrent 5V logique, donc nous pouvons facilement le connecter à un Arduino sans utiliser de convertisseurs de niveau logique.

Le module RF24 peut être considéré comme un bus SPI dont les connexions ne sont pas filaires mais hertziennes. Ce module intègre un protocole spécial appelé : «Enhanced Shock Burst» ou bien ESB. Les trames du protocole ESB contiennent un préambule radio, l'adresse de destination, un en-tête, la charge utile, et un CRC. La charge utile peut être d'une longueur fixe ou variable, sa taille est renseignée dans 6 des 9 bits de l'en-tête.

**Tableau II. 1.** Trame du protocole ESB.

Préambule	Adresse	En-tête	Charge utile	CRC
1 octet	3-5 octets	9 bits	0-32 octet (s)	1-2 octet (s)

Le module peut utiliser 125 canaux différents, ce qui donne la possibilité d'avoir un réseau de 125 modems fonctionnant indépendamment en un seul endroit. Chaque canal peut avoir jusqu'à 6 adresses, ou chaque unité peut communiquer avec jusqu'à 6 autres unités en même temps. Le module RF24 permet de construire les trois réseaux de capteur, comme représentée sur la figure suivante:

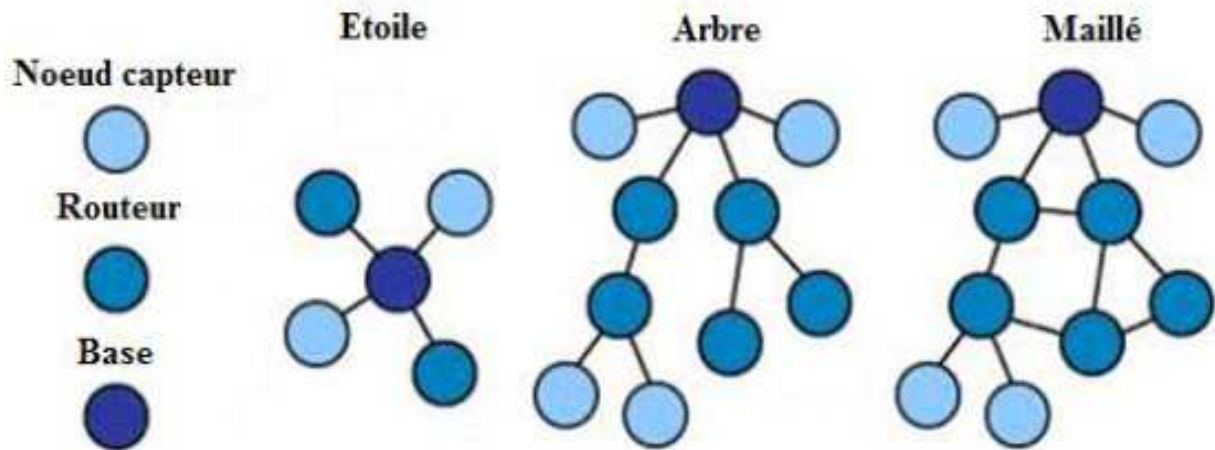


Figure II. 21. Types de communications à base de RF24.

### II.5.2.2. Module GSM

Dans notre projet, nous avons utilisé le module GSM SIM800L représenté sur la figure II.22. Ce module est l'un des plus petits modules GSM du monde avec une taille de  $2.2cm \times 1.8cm$ . C'est un module puissant qui démarre automatiquement et recherche automatiquement le réseau. Il inclut notamment le Bluetooth 3.0+EDR et la radio FM (récepteur uniquement). Il nous permettra d'échanger des SMS, de passer des appels [24].

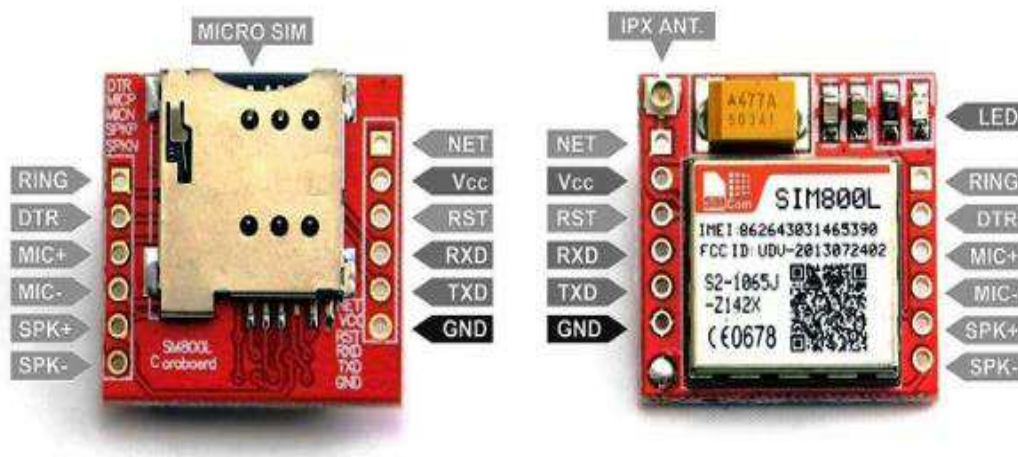


Figure II. 22. Le module GSM SIM 800L.

#### Caractéristiques

- Puce : SIM800L ;
- Tension : 3.7-4.2V ;
- Courant de crête : 2A ;
- Quadri-bande 850/900/1800/1900 MHz ;
- Le module d'alimentation démarre automatiquement, réseau de référencement ;

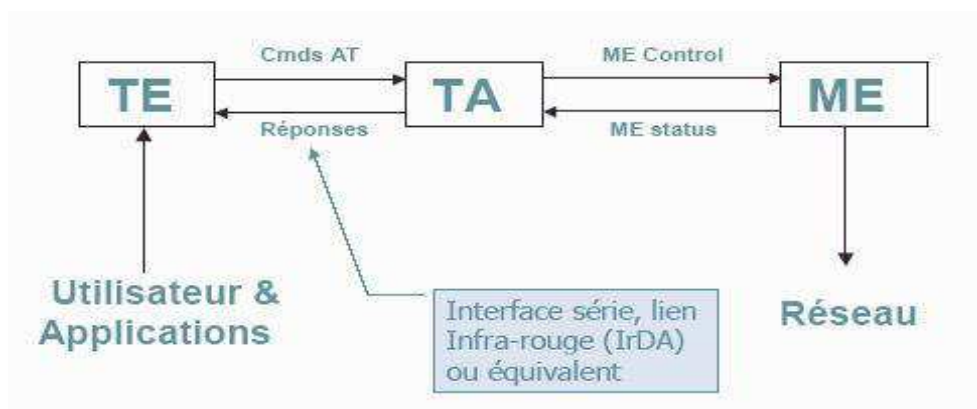
- Emettre et recevoir des appels vocaux à l'aide d'un casque ou un haut-parleur 8Ω externe et microphone électret ;
- Envoyer et recevoir des messages SMS ;
- Envoyer et recevoir des données GPRS (TCP/IP, http, etc.);
- Les feux de signalisation à bord sont complètement allumés. Il clignote lentement lorsqu'il y a un signal, il clignote rapidement lorsqu'il n'y a pas de signal;

### 🚦 Protocole de communication

On utilise le module GSM SIM800L pour envoyer les alertes vers le site web ou une station de service, et pour envoyer les messages SMS. Le module SIM800 est un téléphone GSM simple, sans clavier, écran, micro ni hautparleur mais possédant une liaison série de type UART (Universal Asynchronous Receiver Transmitter), qui permet de le connecté à n'importe quel microcontrôleur ou carte de développement, carte Arduino Uno ou Mega.

### 🚦 Les commandes AT

Les commandes AT sont définies dans la norme "GSM 07.07.AT" est l'abréviation de 'ATtention'. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes *ASCII*). Les commandes permettent la gestion complète du module [25].



**Figure II. 23.** Schéma de fonctionnement AT COMMANDS.

La commande ne dépasse pas 556 caractères

- ME (Mobile Equipment) : téléphone portable.
- TE (Terminal Equipment) : peut-être un ordinateur ou un microcontrôleur.
- TA (Terminal Adaptateur) : assure la liaison entre le ME et le TE.
- TA et ME forment une seule entité, par exemple un téléphone portable standard ou un terminal GSM contient dans son boîtier à la fois le TA et le ME.

Le TE forme une entité à part, par exemple il peut s'agir d'un ordinateur qui dispose d'un port série ou un circuit électronique basé sur un microcontrôleur qui implante un port série.

### II.5.2.3. Module GPS

Pour que les véhicules sont localisés, nous avons pensé à équipé notre système par un module de type GY-GPS6MV2. Ce récepteur GPS puissant et peu coûteux est basé sur le célèbre et haut module GPS u-Blox NEO6M. Il est livré avec une petite batterie pour le démarrage à chaud, et il y a également une mémoire EEPROM intégrée. Pour offrir une meilleure réception du signal, il y a une antenne externe en céramique qui se connecte à la carte via un connecteur U.FL. La Figure II.24 présente ce module.



**Figure II. 24.** Module GPS GY-NEO6MV2.

Le récepteur GPS peut fonctionner de 3.3V à 5V, ainsi toutes ces cartes de 5V Arduino (Arduino-Uno, Méga) et le contrôleur 3.3V qui incluent dans l'Arduino et Raspberry Pi, fonctionnent parfaitement avec ce module GPS. Si on veut lui envoyer des commandes de configuration pour l'interface avec la ligne série (UART) et donc relier Tx (Arduino) avec Rx (GPS), il faudra abaisser la tension de 5V vers 3,3V à l'aide d'un pont diviseur de tension. Les caractéristiques de ce module sont les suivantes [26] :

- ✓ EEPROM pour enregistrer les données de configuration lorsqu'elles sont mises hors tension.
- ✓ Antenne en céramique.
- ✓ Tension de fonctionnement : 3.3V à 5V DC.
- ✓ Interface de communication : UART TTL, 9600 bps.
- ✓ Indicateur de signal LED.
- ✓ VCC : Alimentation du module GPS.
- ✓ RX : Données reçues dans (entrée dans le module GPS).
- ✓ TX : transmission de données (sortie du module GPS).
- ✓ GND : Terre (Logique et mise à la terre).

- ✓ Dimension : 36 x 26 mm.
- ✓ Poids : 22g.

## II.6. Conclusion

Dans ce chapitre, nous avons donné un aperçu sur les différents types de communication sans fil. Commençant par la définition d'une communication radiofréquence. Puis, nous avons présenté le système GSM, nous avons essayés de présenter le système GSM en se basant principalement sur son structure et architecture. Nous présentons ensuite le système GPS d'une façon générale, nous avons assimilé son principe de positionnement et on a également vu son fonctionnement et ces différentes applications.

Avant d'aborder la description des différents éléments de notre projet, nous avons présenté les parties essentielles d'une unité de communication (nRF24, GSM, GPS) et de traitement par les deux cartes électroniques programmables (Arduino, Raspberry Pi), éventuellement les modules utilisés pour la réalisation d'un système de communication entre les véhicules.

Dans le prochain chapitre, nous allons nous intéresser aux techniques de détection et reconnaissance automatique des panneaux de signalisations en utilisant l'intelligence artificielle, ces dernières techniques sont utilisées pour améliorer les systèmes de communication entre les véhicules. Dans le chapitre IV, nous avons donné plus de détails sur l'utilisation de l'intelligence artificielle et leurs liaisons avec les technologies de communication sans fil.

## *Chapitre III*

---

# *Traitement d'image et Intelligence artificielle*

### **III.1. Introduction**

De nos jours, le nombre de véhicules circulant sur la route ne cesse d'augmenter, ce qui accroît le nombre d'accidents et les problèmes de sécurité routière. Pour pallier ces problèmes, il est nécessaire de trouver des nouvelles technologies de l'information et de la communication aux domaines des transports afin d'améliorer la sécurité routière et de réduire le nombre d'accidents.

Parmi les services qu'offrent les systèmes de transport intelligents, on trouve l'aide à la conduite. Un conducteur peut être distrait de sa tâche principale, la conduite, ce qui occasionne un manque de vigilance vis à vis de la signalisation courante. Cette situation augmente le risque d'accident. En effet, manquer un panneau de limitation de vitesse ou d'interdiction de doubler génère une situation à risque en plus du fait d'exposer le conducteur à des sanctions. Afin de réduire ce risque, nous avons proposé un système de reconnaissance de panneaux routiers (TSR-Traffic Signs Recognition) basé sur l'intelligence artificielle. TSR assiste le conducteur soit en le tenant informé de la signalisation liée à la route parcourue, soit de réguler la vitesse d'un véhicule autonome par exemple.

Dans ce chapitre, nous nous sommes concentrés sur l'étape de détection des panneaux de signalisation, en évaluant leurs performances sur un ensemble d'images de scène de trafic (mode offline), et en temps réel à l'aide d'une carte Raspberry Pi et une caméra (mode online).

Dans le cadre de notre travail, nous nous intéressons, plus particulièrement, à la segmentation, détection, et la reconnaissance automatique, où nous concevons et mettons en œuvre une méthode efficace pour identifier les panneaux de signalisation. Nous allons utiliser le descripteur de caractéristique HOG (Histogram of Oriented Gradients) combiné avec deux méthodes de classification : perceptron multicouche MLP (Multi-Layer Perceptron) et machine à vecteurs de support SVM (Support Vector Machine) pour calculer les caractéristiques et les descripteurs des objets extraits à l'étape de détection et ensuite l'identification des types de panneaux de signalisation.

### **III.2. Détection et reconnaissance automatique**

La détection et la reconnaissance automatiques des panneaux de signalisation est importante dans les systèmes avancés d'assistance aux conducteurs, ont d'un grand intérêt ces dernières années. La détection des panneaux de signalisation à un impact direct sur la sécurité du conducteur et les dommages peuvent être facilement produits en raison de leur ignorance. Les

systèmes automatiques développés pour aider le conducteur basés sur la détection et la reconnaissance des signes peuvent par conséquent corriger les comportements de conduite les plus dangereux.

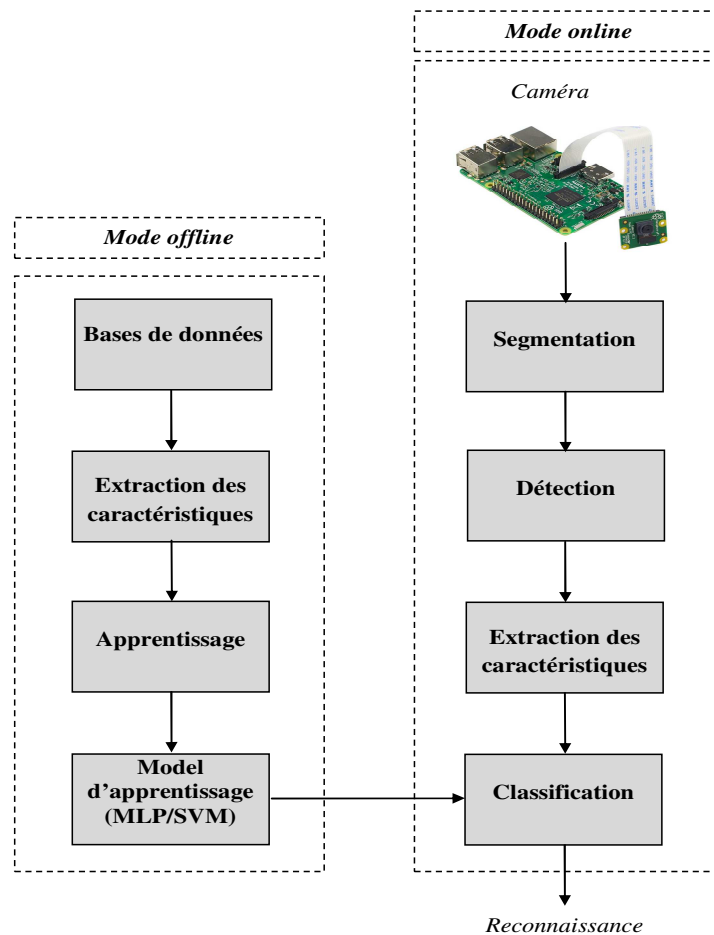
Pour identifier les signes la plupart des chercheurs divisent la tâche en trois étapes séquentielles : segmentation, détection et reconnaissance. Le rôle de la première étape est de déterminer la zone des panneaux dans la scène de la route, tandis que la détection sélectionne les zones qui ont une forme appropriée de panneau de signalisation. Ensuite, l'étape de reconnaissance identifie l'information des panneaux de signalisation extraits.

Les approches d'identification des signes peuvent être classées en trois classes principales : (i) des méthodes géométriques basées sur la forme géométrique pour détecter les signes, (ii) des méthodes colorimétriques qui sont basées sur la couleur dans la phase de détection, (iii) et il existe également des méthodes qui combinent l'apprentissage avec les méthodes des deux classes précédentes [5].

Dans notre travail nous avons utilisé les méthodes colorimétriques, l'espace de couleur le plus utilisé est RVB (Plans : Rouge, Vert, Bleu). Dans ces approches, les images sont d'abord converties en un espace colorimétrique désigné, ensuite un algorithme de seuillage basé sur la détection des couleurs dominantes, puis une méthode de filtrage pour la suppression des informations inutiles et extraire les différents objets, par la suite on applique une technique de description des caractéristiques (HOG), ces régions sont ensuite classifiées par un algorithme de reconnaissance (MLP ou SVM).

### III.3. Reconnaissance des systèmes TSR

On présente dans cette partie l'approche proposée pour la reconnaissance des systèmes TSR (voir la Figure III.1). Notre méthode basée sur deux phases : la phase de calibration pour trouver les meilleurs paramètres de notre modèle (seuil de détection, l'architecture de classifieur MLP ou SVM), cette dernière est une phase hors ligne (mode offline), et une autre phase en temps réel (mode online) qui permet de détecter et identifier les panneaux de signalisation, avec l'utilisation des paramètres trouvés dans la phase de calibration.



**Figure III. 1.** Schéma général d'un système de détection et reconnaissance.

Les différentes étapes de notre méthode proposée sont détaillées dans sections suivantes.

### III.4. Phase de calibrage

#### III.4.1. Description des bases de données

Les panneaux de signalisation routière sont des éléments de la signalisation routière. Ils désignent à la fois le dispositif sur lequel sont implanté un signal routier et le signal proprement dit.

Placés sur le côté des routes, ils ont pour fonction [27]:

- de rendre plus sûre la circulation routière en informant les usagers d'éventuels dangers qu'ils peuvent rencontrer ;
- de faciliter cette circulation, en indiquant par exemple les directions à suivre ;
- d'indiquer ou de rappeler diverses prescriptions particulières de police ;
- de donner des informations relatives à l'usage de la route.

Pour comprendre le sens des panneaux, il faut d'abord remarquer leurs couleur et formes. La Figure III.2 une représentation récapitulatif comment s'articulent les formes et les couleurs des panneaux.

Toutefois, la signalisation n'a et ne saurait avoir le caractère d'une garantie assurée par les pouvoirs publics aux usagers de la route contre les aléas et les dangers de la circulation. Ces usagers circulent toujours à leurs risques et périls. Pour ces raisons, notre travail est basé sur la reconnaissance automatique des panneaux de signalisation routière réalisée pour le but d'aider les conducteurs dans la route et augmenter leurs sécurités.

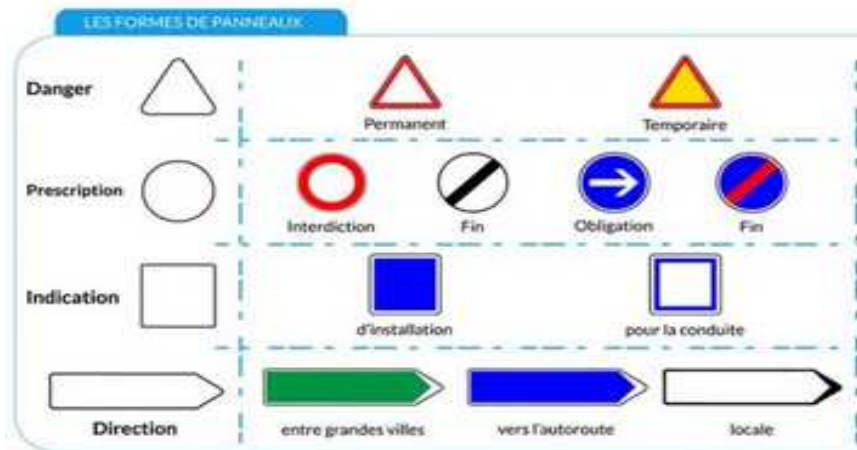


Figure III. 2. Les formes et les couleurs des panneaux.

La validation de toutes les méthodes utilisées dans ce travail est réalisée sur une base de donnée réelle liées à signalisation routière. Cette base de données consiste de 34 classes des panneaux de signalisation plus une classe de l'environnement (arrière-plan) [28, 29]. La figure suivante représente les différents panneaux de signalisation utilisée comme base de données dans la suite de notre mémoire.

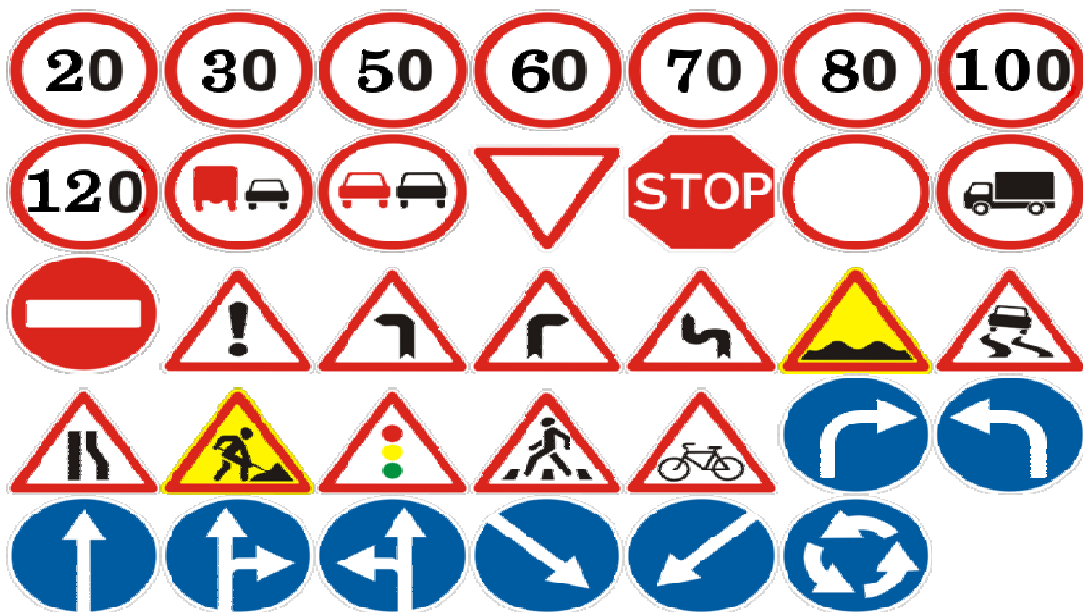
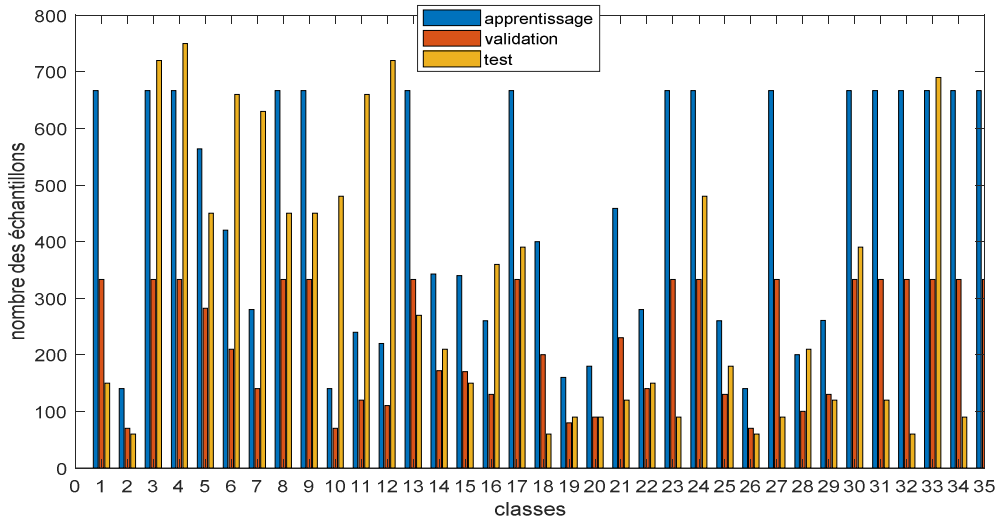


Figure III. 3. Panneaux signalisation.

Comme le montre la figure III.3, pour la phase de détection nous avons besoin seulement de détecté les deux couleurs : Rouge et Bleu.

La figure III.4 représente le nombre des échantillons dans l'ensemble d'apprentissage, validation, et l'ensemble de test, respectivement dans chaque classe. Le Tableau III.1 illustre le nombre total des échantillons dans les trois ensembles : apprentissage, validation, et test.



**Figure III. 4.** Nombre des échantillons dans chaque classe.

**Tableau III. 1.** La division de base de données en trois ensembles : apprentissage, validation, et test.

Base de données	Nombre total
Apprentissage	15959
Validation	7972
Test	10740

### III.4.2. Extraction des caractéristiques

La technique de l'histogramme de gradients orientés HOG introduite en 2005 [8], comme un descripteur de fonctionnalités de l'image. L'idée essentielle derrière l'histogramme de gradient orienté c'est que l'apparence locale et la forme d'objet dans une image peut être décrite par la distribution d'intensité des gradients ou de direction des contours. HOG est utilisé pour calculer l'histogramme de gradient orienté en zone locale de bloc extrait de l'étape précédente de détection.

Les étapes de l'algorithme sont présentées comme suit:

**Prétraitement de l'image:** Convertir les plans de couleurs extraites en gris.

**Calcul du gradient:** calcul du gradient de l'image dans le sens horizontal et vertical des blocs à l'aide de l'opérateur *Sobel edge*. La formule de calcul est la suivante:

$$G_x(i, j) = X(i+1, j) - X(i-1, j) \quad (\text{III.1})$$

$$G_y(i, j) = X(i, j+1) - X(i, j-1) \quad (\text{III.2})$$

où  $X(i, j)$  est la valeur du pixel,

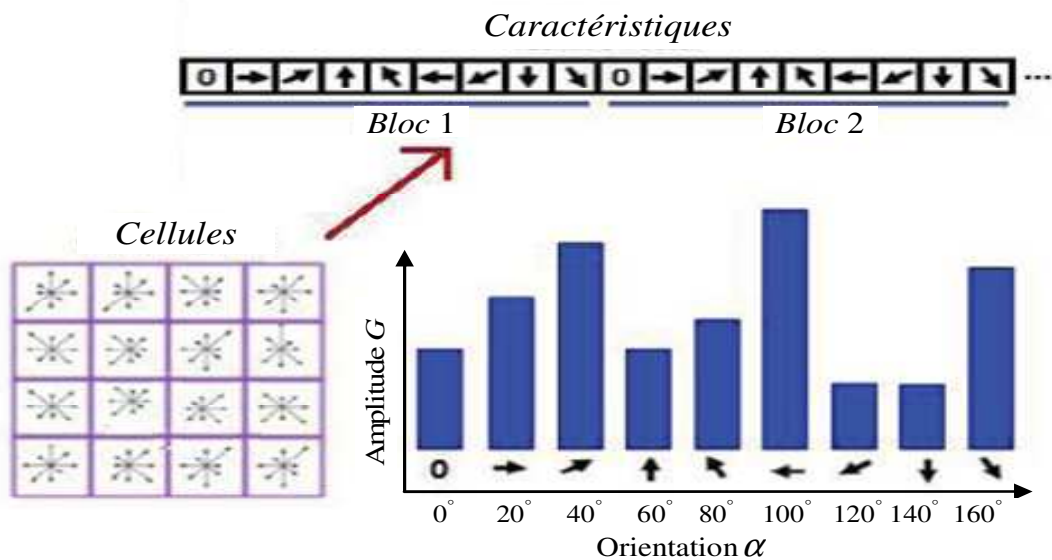
$G_x(i, j)$ ,  $G_y(i, j)$  sont respectivement des gradients dans les directions verticale et horizontale du pixel  $X(i, j)$ .

Par la suite en calcul l'amplitude ( $G$ ) et la direction (orientation  $\alpha$ ) du gradient comme suit :

$$G(i, j) = \sqrt{G_x^2(i, j) + G_y^2(i, j)} \quad (\text{III.3})$$

$$\alpha(i, j) = \tan^{-1}\left(\frac{G_x(i, j)}{G_y(i, j)}\right) \quad (\text{III.4})$$

Le principe du descripteur HOG, consiste dans un premier temps, à diviser l'image de l'objet à caractériser en cellules qui sont elles mêmes regroupées en blocs qui se recouvrent (Figure III.5). Dans chaque cellule, les orientations des gradients de l'ensemble des pixels sont calculées et collectées dans un histogramme monodimensionnel. Ensuite, ces histogrammes sont normalisés localement en intensité au niveau de chaque bloc. Plusieurs versions existent pour effectuer cette normalisation. Certaines agissent sur la normalisation des histogrammes en considérant uniquement les gradients des orientations [30], tandis que d'autres comptabilisent aussi le module du gradient au lieu des occurrences seules [10, 26].



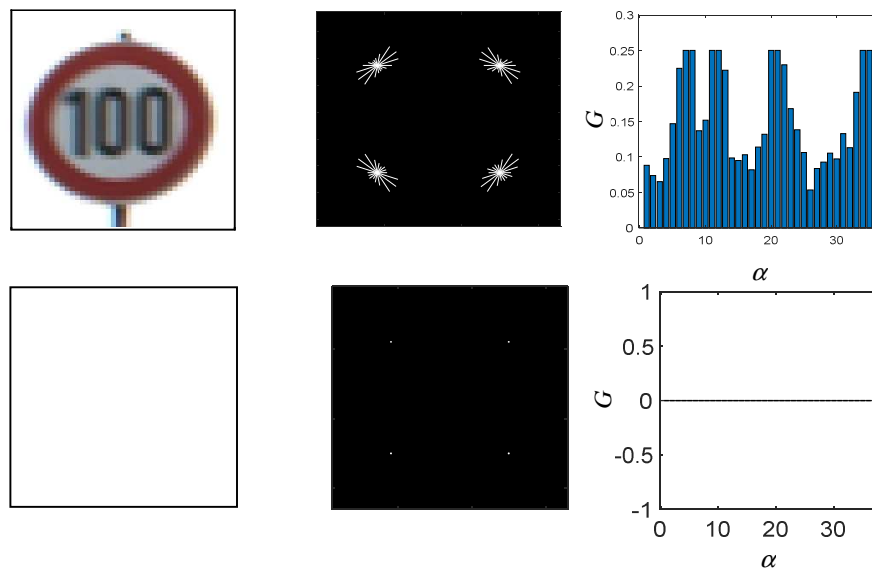
**Figure III. 5.** Etapes du Descripteur HOG.

L'histogramme formé à partir des orientations du gradient des pixels à l'intérieur de chaque cellule est quantifié en 9 bits (intervalles) qui sont de même tailles couvrant 180 ou 360 degrés d'orientation.

Le descripteur HOG présente certains avantages clés par rapport aux autres méthodes de descripteur :

- ✓ Comme il fonctionne sur les cellules localisées, il montre invariance aux transformations géométriques et photométriques de tels changements (l'aurait seulement apparaître dans les régions spatiales plus grandes).
- ✓ Le descripteur HOG est donc particulièrement adapté pour la détection des panneaux de signalisation dans les images.

La figure suivante illustre un exemple de l'application de descripteur HOG sur deux images :



**Figure III. 6.** Applications de descripteur de HOG sur deux images de teste, dimensions des images [16×16], Cellule [8×8].

### III.4.3. Apprentissage

Il est clair que la fiabilité d'un système de détection et de reconnaissance des panneaux de signalisation repose en grand porté sur le choix de la méthode d'extraction des caractéristiques et la méthode de classification.

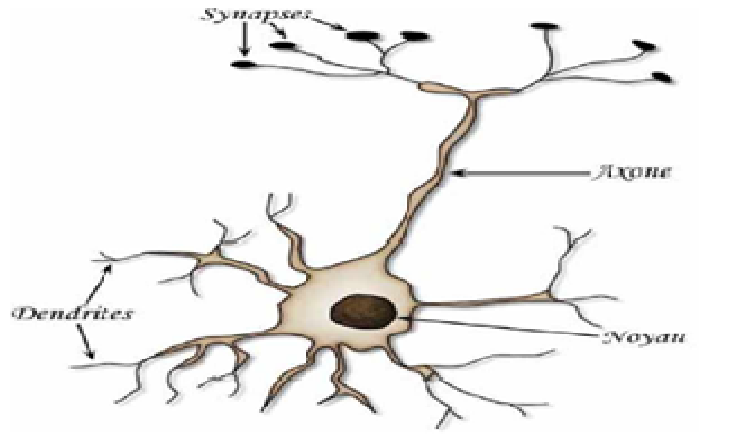
L'étude que nous avons parité nous a permis de faire le choix des combinaisons HOG-MLP et HOG-SVM respectivement pour la détection et la classification.

#### III.4.3.1. Réseaux de neurones multicouches MLP

Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie (voir la Figure III.7) [31].

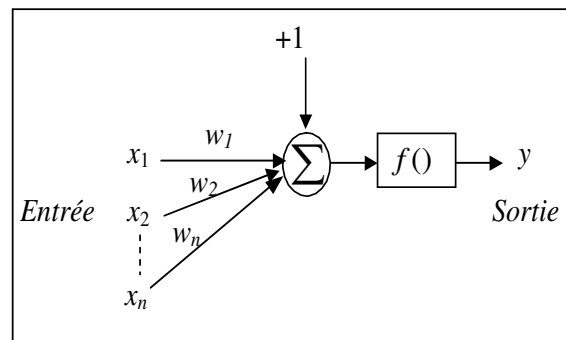
La structure d'un neurone se compose de trois parties :

- La soma : ou cellule d'activité nerveuse, au centre du neurone.
- L'axone : attaché au soma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.
- Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones.



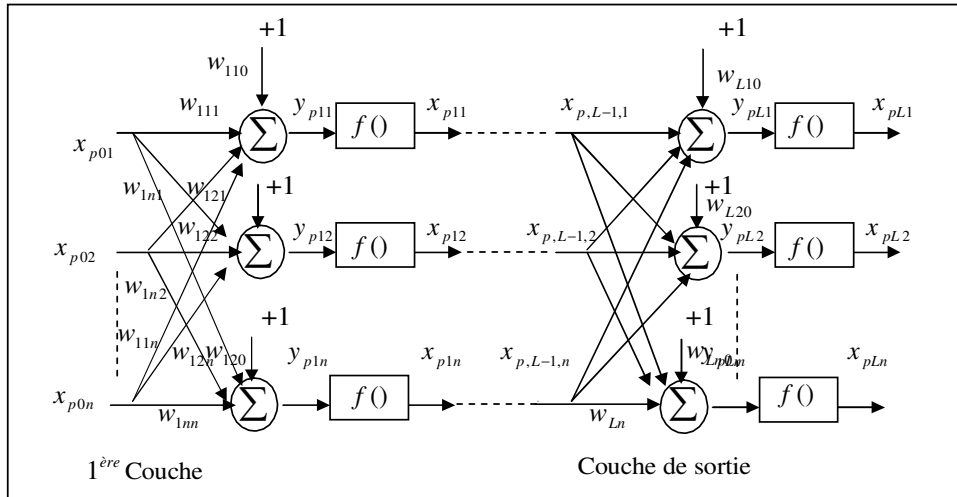
**Figure III. 7.** Schéma d'un neurone biologique.

On peut faire un parallèle entre des modèles mathématiques et certaines informations que l'on connaît à propos du neurone biologique, la figure suivante est une représentation du modèle d'un neurone.



**Figure III. 8.** Représentation du modèle d'un neurone artificiel.

La généralisation d'un neurone est les réseaux de neurones, comme représenté sur la figure III.9. Un réseau de neurones est un maillage de plusieurs neurones, généralement organisé en couches. Pour construire une couche de  $k$  neurones, il s'agit simplement de les assembler comme à la figure III.9. Les  $n_i$  neurones d'une même couche sont tous branchés aux  $n_i$  entrées. On dit alors que la couche est totalement connectée.



**Figure III. 9.** Structure d'un réseau de neurone multicouche.

La topologie d'un tel réseau est formée de plusieurs couches de neurones sans communication à l'intérieur d'une même couche (figure III.9).

- une couche en entrée qui représente les entrées auxquelles sont transmises les données à traiter en provenance d'une source extérieure au réseau ;
- une ou plusieurs couches cachées effectuant le traitement spécifique du réseau ;
- une en sortie qui délivre les résultats.

L'apprentissage est supervisé, c'est-à-dire que l'on présente au réseau, en même temps, une forme et son modèle. L'apprentissage dans ce type de structure consiste à appliquer des couples (entrées, sorties désirées) à l'entrée du réseau.

Une sortie réelle est calculée pour chaque neurone de la  $j^{ème}$  couche. Ce calcul est effectué de proche en proche de la couche d'entrée vers la couche de sortie, celle-ci est appelée "propagation d'avant". Ensuite l'erreur est calculée puis propagée dans le réseau, donnant lieu à une modification des poids.

On considère un réseau comportant une couche d'entrée à  $n$  neurones, une couche de sortie à  $m$  neurones et il comporte une à plusieurs couches cachées.

Supposons qu'on dispose d'un ensemble d'apprentissage composé de  $k$  pair de vecteurs :

$$(x_1, o_1), (x_2, o_2), \dots, (x_k, o_k)$$

avec :

$$x_p = (x_{p,0,1}, x_{p,0,2}, \dots, x_{p,0,n})^t \in R^n$$

Vecteur d'entrée.

$$O_p = (O_{p,1}, O_{p,2}, \dots, O_{p,m})^t \in R^m$$

Vecteur des sorties désiré.

$$y_p = (y_{p,L,1}, y_{p,L,2}, \dots, y_{p,L,n})^t \in R^n$$

Vecteur des sorties réel du réseau.

où :

$w_{j,k,i}$  : La connexion entre le neurone  $k$  de la couche  $j-1$  et le neurone  $i$  de la couche  $j$ .

$y_{p,j,k}$  : L'entrée totale du neurone  $k$  pour l'échantillon  $p$  de la couche  $j$ .

$w_{j,k,0} = \theta_{j,k}$  : Le poids fictif du neurone  $k$  de la couche  $j$  correspondant à un biais dont l'entrée est fixé à 1.

L'entrée totale du  $k$  nœud pour la couche  $j$  est :

$$y_{p,j,k} = \sum_{i=0}^n w_{j,k,i} x_{p,j-1,i} \quad (\text{III.5})$$

La sortie de ce nœud sera :

$$x_{p,j,k} = f(y_{p,j,k}) \quad (\text{III.6})$$

où  $f$  est une fonction de transfert.

#### III.4.3.1.1. Fonctions de transfert

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone. Les quatre les plus utilisées sont les fonctions "seuil", "linéaire", "sigmoïde", et "tangente hyperbolique" [32].

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne dépasse pas le seuil, la fonction retourne alors la valeur 0 (on peut interpréter ce 0 comme signifiant faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Utilisée dans le contexte d'un neurone, cette fonction permet de prendre des décisions binaires.

La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

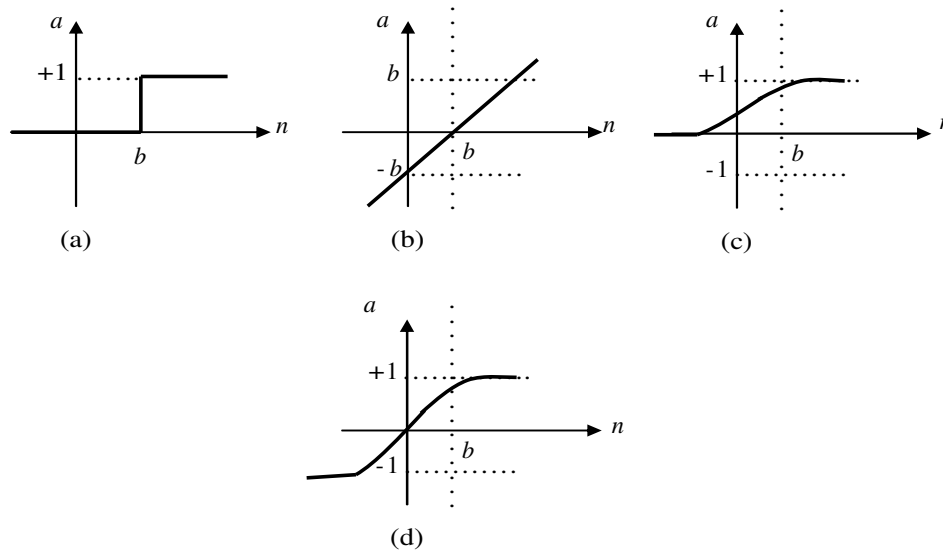
$$a = n \quad (\text{III.7})$$

Les fonctions de transferts sigmoïde (Equation III.8) et tangente hyperbolique (Equation III.9) sont données par les équations suivantes :

$$a = \frac{1}{1 + e^{-n}} \quad (\text{III.8})$$

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (\text{III.9})$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de  $b$  (biais), respectivement. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement, que la fonction "tangente hyperbolique" est une version symétrique de la sigmoïde.



**Figure III. 10.** Fonction de transfert : (a) du neurone “seuil”; (b) du neurone “linéaire”, (c) “sigmoïde”, et (d) du neurone “tangente hyperbolique”

En général, même fonction d'activation  $f$  (tangente hyperbolique) pour toutes les unités des couches cachées son équation est donnée par l'équation (III.9).

#### III.4.3.1.2. Mise en œuvre des réseaux neuronaux

Nous allons suivre une démarche reprise par Wierenga et Kluytmans [33] qui est composée de quatre étapes principales :

##### Étape 1 : fixer le nombre de couches cachées

Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue. Une seconde couche cachée prend en compte les discontinuités éventuelles.

##### Étape 2 : déterminer le nombre de neurones par couches cachées

Chaque neurone supplémentaire permet de prendre en compte des profils spécifiques des neurones d'entrée. Un nombre plus important permet donc de mieux coller aux données présentées mais diminue la capacité de généralisation du réseau. Ici non plus il n'existe pas de règle générale mais des règles empiriques.

##### Étape 3 : choisir la fonction d'activation

Nous considérerons la fonction tangente hyperbolique pour le passage de la couche d'entrée à la couche cachée. Le passage de cette dernière à la couche de sortie sera soit linéaire, soit non linéaire (tangente hyperbolique) selon nos types de variables.

#### Etape 4 : choisir l'apprentissage

L'apprentissage par rétro-propagation nécessite la détermination du paramètre d'ajustement des poids synaptiques à chaque itération.

La détermination du critère d'arrêt est aussi cruciale dans la mesure où la convergence peut passer par des minima locaux.

##### III.4.3.1.3. Algorithmes d'apprentissage

La rétro-propagation consiste à rétro-propager l'erreur commise par un neurone à ses synapses et aux neurones qui y sont reliés. Pour les réseaux de neurones, on utilise habituellement la rétro-propagation du gradient de l'erreur [34].

L'algorithme de rétro-propagation du gradient change les poids d'un réseau dont l'architecture est fixée par l'opérateur. Cet algorithme repose sur la minimisation de l'erreur quadratique entre les sorties calculées et celles souhaitées.

Le terme rétro-propagation du gradient provient du fait que l'erreur calculée en sortie est transmise en sens inverse vers l'entrée [35].

L'erreur commise sur le  $k^{\text{ème}}$  nœud de sortie est [35]:

$$\delta_{pk} = O_{pk} - x_{pk} \quad (\text{III.10})$$

Par conséquent l'erreur totale (pour tous les nœuds) est [35]:

$$E_p = \frac{1}{2} \sum_{k=1}^m \delta_{p,k}^2 = \frac{1}{2} \sum_{k=1}^m (O_{p,k} - x_{p,l,k})^2 \quad (\text{III.11})$$

Pour minimiser  $E_p$  (erreur quadratique), on calcule son gradient par rapport à chaque poids  $w$ , puis on modifie les poids dans le sens inverse du gradient.

La fonction de mise à jour des poids est en fonction du calcul du gradient. Ainsi, les poids sur la couche des sorties sont mis à jour de la façon suivante [35]:

$$w_{l,k,j}(t+1) = w_{l,k,j}(t) + \Delta_p w_{l,k,j}(t) \quad (\text{III.12})$$

$$\Delta_p w_{l,k,j}(t) = \mu (O_{p,k} - x_{p,l,k}) f'(y_{p,l,k}) x_{p,l-1,j} \quad (\text{III.13})$$

ou :  $\mu$  est le pas (taux) d'apprentissage.

La convergence du réseau par rétropropagation est un problème crucial car il requiert de nombreuses itérations. Pour pallier à ce problème, un paramètre est souvent rajouté pour accélérer la convergence. Ce paramètre est appelé le momentum.

Le momentum est un moyen efficace pour accélérer l'apprentissage et aussi pour pouvoir sortir des minimums locaux.

La règle de mise à jour des poids devient alors :

$$w_{l,k,j}(t+1) = w_{l,k,j}(t) + \mu e_{p,l,k} x_{p,l-1,j} + m[w_{l,k,j}(t) - w_{l,k,j}(t-1)] \quad (\text{III.14})$$

avec  $e_{p,l,k} = \mu(O_{p,k} - x_{p,l,k})f'(y_{p,l,k})x_{p,l-1,j}$

$m$  : est la constante du momentum.

Généralement, le pas d'apprentissage  $\mu$  et le momentum  $m$  doit être compris entre 0 et 1 pour assurer la convergence de l'algorithme vers une solution optimale.

**Le résumé l'algorithme de Rétropropagation est donnée par les étapes suivantes :**

**Étape (1):** Appliquer un vecteur d'entrée  $x_p = (x_{p,0,1}, x_{p,0,2}, \dots, x_{p,0,n})^t$  aux nœuds d'entrées puis initialiser les poids du réseau à de petites valeurs aléatoires;

**Étape (2):** Exécuter l'échantillon d'apprentissage à travers le réseau;

**Étape (3):** Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées;

**Étape (4):** Mise à jour les poids de la couche de sortie et couches cachées;

**Étape (5):** Répéter ce processus jusqu'à ce que l'erreur  $E_p$  devienne acceptable (aller à l'étape 2).

Il n'existe pas de règles permettant de déterminer le nombre de couches cachées dans un réseau donné  $n_i$ , le nombre de neurones dans chacune d'elles, le pas d'apprentissage  $\mu$ , et le momentum  $m$ . La solution est d'optimiser ces paramètres d'une façon empirique.

#### III.4.3.2. Machine à vecteurs de support

Le classificateur SVM (en anglais Support Vector Machine, SVM) [36] a été conçu pour une séparation de deux ensembles de données. Il est considéré donc comme un classificateur binaire.

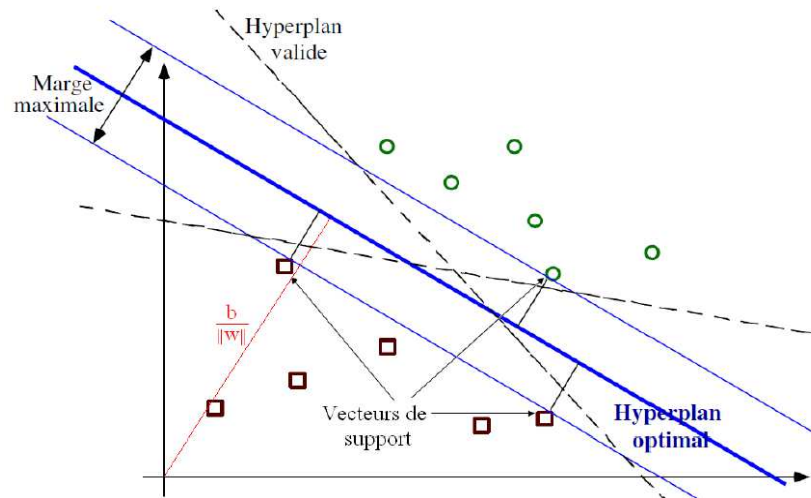
Les SVM ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyper paramètres, leurs garanties théoriques, et leurs bons résultats en pratique.

Le problème consiste à transformer l'espace dans lequel on travaille en un autre espace dit de grandes dimensions et à l'intérieur duquel nous établirons une séparation en différentes classes. Le passage d'un espace à l'autre s'effectue suivant quelques transformations mathématiques utilisant des fonctions non linéaires, encore appelées noyaux (ou fonction kernel) [37].

S'il existe un hyperplan qui sépare les deux classes, les points de l'hyperplan sont décrits par l'équation :

$$x_i \omega + b = 0 \quad (\text{III.15})$$

Où  $\omega$  est la normale au plan et  $|b|/\|\omega\|$  la distance entre l'hyperplan et l'origine, comme représenté dans la figure suivante:



**Figure III. 11.** Cas linéairement séparable.

Pour estimer la classe d'un nouveau point  $x$ , on calcule [37]:

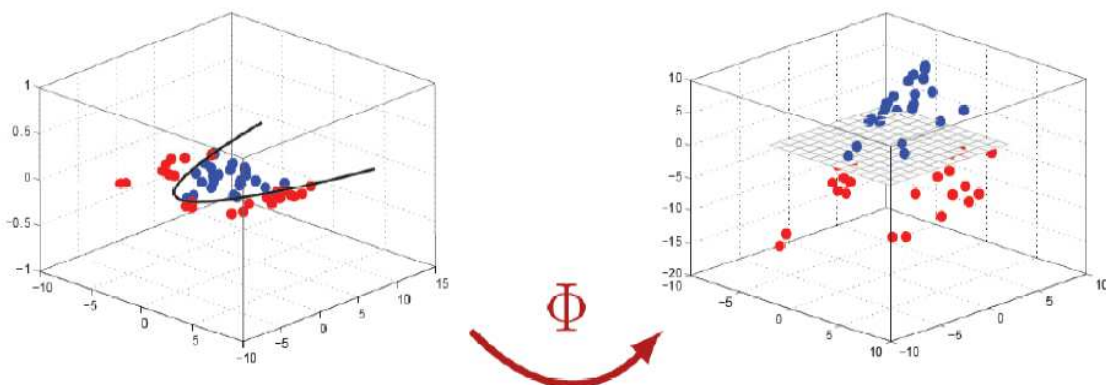
$$f(x) = \text{sign}(x \cdot \omega^0 + b^0) = \text{sign}\left(\sum_{VS} \alpha_i^0 y_i(x_i, x) + b^0\right) \quad (\text{III.16})$$

où  $\omega^0, b^0$  sont des constantes qui vérifiant l'hyperplan optimal,  $VS$ : est l'ensemble des vecteurs de support  $0 \leq \alpha_i^0 \leq C$ .

Les  $\alpha_i^0$  sont les multiplicateurs de Lagrange. Le paramètre de régularisation  $C$  contrôle le niveau d'erreur dans la classification (sert à fixer le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la marge).

Pour généraliser cette méthode dans le cas où la fonction de décision n'est pas linéaire, l'idée est la transformation vers un autre espace de dimension suffisamment grande en utilisant une fonction  $\Phi$ , tel qu'il existe une fonction  $\mathbf{K}$  de noyau.

$$\mathbf{K}(x_i, x) = \Phi(x_i)^T \cdot \Phi(x) \quad (\text{III.17})$$



**Figure III. 12.** Transformer l'espace de départ vers un espace de dimension supérieure.

Pour estimer la classe d'un point  $x$ , il suffit de calculer la fonction [38]:

$$f(x) = \text{sign}\left(\sum_{\forall s} \alpha_i^0 y_i k(x_i, x) + b^0\right) \quad (\text{III.18})$$

Pour choisir le noyau  $\mathbf{K}$  : les principaux noyaux utilisés dans les applications sont [39, 40]:

➤ **Le noyau RBF Gaussien:**

$$\mathbf{K}(x_i, x) = \exp\left(-\gamma \|x_i - x\|^2\right) \quad (\text{III.19})$$

$\gamma$  : Paramètre inversement proportionnel à la largeur de la gaussienne ( $\sigma$ ), avec  $\gamma = \frac{1}{2\sigma^2}$ .

➤ **Le noyau Polynomial de degré  $d$ :**

$$\mathbf{K}(x_i, x) = (x_i^T \cdot x + 1)^d \quad (\text{III.20})$$

➤ **Le noyau Linéaire:**

$$\mathbf{K}(x_i, x) = (x_i^T \cdot x) \quad (\text{III.21})$$

Nous allons donc appliquer cette approche pour la classification des panneaux de signalisation. Dans notre travail, nous avons choisi le noyau RBF, donc les paramètres à optimiser dans le cas de ce noyau sont :  $C$  et  $\gamma$ , ces paramètres sont optimisés de façon empirique avec la technique de validation simple.

### III.4.3.3. Technique de validation

Dans la littérature scientifique, il existe plusieurs techniques pour l'optimisation des meilleurs paramètres de l'algorithme de classification (MLP, SVM) : Validation simple, Monte-Carlo validation Croisé,  $k$ -fold cross-validation. Dans notre cas, nous nous sommes intéressés à la technique de validation simple.

#### III.4.3.3.1. Validation simple

La technique de validation consiste tout simplement à diviser les données disponibles en deux ensembles (les ensembles d'apprentissage et de validation), sans qu'une donnée ne soit commune. Un nombre non négligeable de données est bien entendu nécessaire dans l'ensemble de validation pour estimer correctement le critère de généralisation, donc pour optimiser les paramètres de classifieur (MLP ou SVM). Souvent on garde 2/3 des données dans l'ensemble d'apprentissage et en réserve 1/3 pour la validation. Les ensembles d'apprentissage et de validation sont tirés aléatoirement parmi les  $N$  données disponibles.

#### III.4.3.4. Classification multiclasse

Les réseaux de neurones MLP ou machines à vecteur support SVM sont dans leur origine bi-classes (classification binaire). Cependant, les problèmes du monde réel sont dans la plupart des cas multiclasse (exemple : la discrimination entre plusieurs panneaux de signalisation).

Dans ce cas, on ne cherche pas à affecter un nouvel exemple à l'une de deux classes mais à l'une parmi plusieurs, c.-à-d. que la décision n'est plus biclassée et un seul hyperplan ne suffit plus.

Les méthodes des classifications multiclassées, réduisent le problème multiclassé à une composition de plusieurs hyperplans biclassés permettant de tracer les frontières de décision entre les différentes classes [41, 42]. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous ensembles représentant chacun un problème de classification biclassés. Pour chaque problème un hyperplan de séparation est déterminé par la méthode MLP ou SVM biclassés. On construit lors de la classification une hiérarchie des hyperplans biclassés qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple. On trouve dans la littérature deux méthodes plus utilisées pour la décomposition : une contre reste, une contre une. Dans notre travail nous intéressent à la méthode une-contre-reste (1vsR) :

#### III.4.3.4.1. Une-contre-reste (1vsR)

Selon la formulation de Vapnik [36], elle consiste à déterminer pour chaque classe  $k$  un hyperplan  $H_k(x)$  la séparant de toutes les autres classes. Cette classe  $k$  est considéré comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1), ce qui résulte, pour un problème à  $K$  classes. Un hyperplan  $H_k$  est défini pour chaque classe  $k$  par la fonction de décision suivante :

$$k^* = \arg_{1 \leq k \leq K} \text{Max}(f_k(x)) \quad (\text{III.22})$$

avec  $f_k(x)$  : la fonction de décision ( $k$  classe) de la sortie MLP ou SVM.

La Figure III.13 représente un exemple d'un cas de séparation de 3 classes.

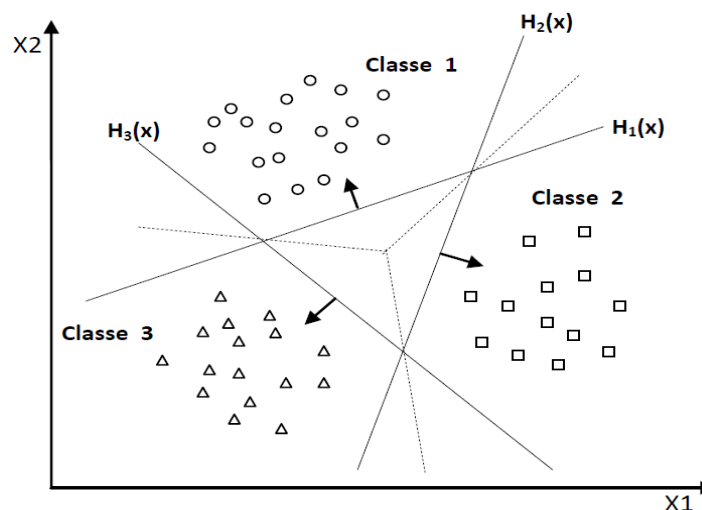


Figure III. 13. Séparation entre trois classes en utilisant la méthode 1vsR.

### III.5. Phase en temps réel

#### III.5.1. Segmentation

La caméra de Raspberry Pi prend des photos à intervalles de temps réguliers. On effectue alors une segmentation, qui permet d'obtenir une image ne contenant que du blanc et du noir (panneaux de signalisation/arrière-plan). On définit pour cela un seuil et on compare la luminance de chaque pixel à ce seuil. Dans un cas on remplace ce pixel par du blanc, dans l'autre par du noir. L'image binaire obtenue est noir aux endroits qui n'ont pas des panneaux de signalisations, et blanche où il ya eu un panneau de signalisation.

Dans ce mémoire on a besoin de deux couleurs : le rouge et le bleu (couleurs des panneaux), donc on doit extraire la couleur rouge pour les panneaux qui indiquent un danger ou une interdiction et la couleur bleu pour les panneaux qui indiquent une obligation. Il existe différentes méthodes de segmentation, dans ce mémoire nous avons proposé d'utiliser deux types de segmentation :

- ✚ Seuillage fixe [5].
- ✚ Segmentation par la méthode régions extrêmes maximales stables MSER (maximally stable extremal regions) [4].

##### III.5.1.1. Seuillage fixe

La méthode consiste à prédéfinir des seuils d'après l'histogramme de l'image qui extraire la couleur rouge et bleu. Si les coefficients obtenus est supérieurs ou égaux à des seuils  $TH_1$ ,  $TH_2$ ,  $TH_3$ ,  $TH_4$ , ces coefficients mis à 1, sinon 0 (arrière-plan). On varie le seuil de 10 jusqu'à 250.

Pour la séparation des pixels en rouge (image binarisé  $I_R$ ) et bleu ( $I_B$ ), en utilise les deux équations suivantes :

$$I_R = \begin{cases} si & X_R - X_V > TH_1 \text{ et } X_R - X_B > TH_2 & 1 \\ ailleurs & & 0 \end{cases} \quad (III.23)$$

$$I_B = \begin{cases} si & X_B - X_R > TH_3 \text{ et } X_B > TH_4 & 1 \\ ailleurs & & 0 \end{cases} \quad (III.24)$$

##### III.5.1.2. Régions extrêmes maximales stables (MSER)

La méthode MSER est basée sur la transformation de l'espace de couleur RVB (rouge vert bleu) vers un autre espace normalisé rouge/bleu  $\Omega_{RB}$ . Les valeurs les plus élevée de cette normalisation sont utilisée pour la détection de la couleur rouge ou bleu simultanément, les valeurs les plus bas pour les autres couleurs.

L'image normalisé Rouge/Bleu est donné par l'équation suivante :

$$\Omega_{RB} = \max\left(\frac{R}{R+V+B}, \frac{B}{R+V+B}\right) \quad (\text{III.25})$$

L'image binarisé rouge ou bleu ( $I_{RB}$ ) est définie par l'équation III.26 :

$$I_{RB} = \begin{cases} si & \Omega_{RB} > TH & 1 \\ ailleurs & & 0 \end{cases} \quad (\text{III.26})$$

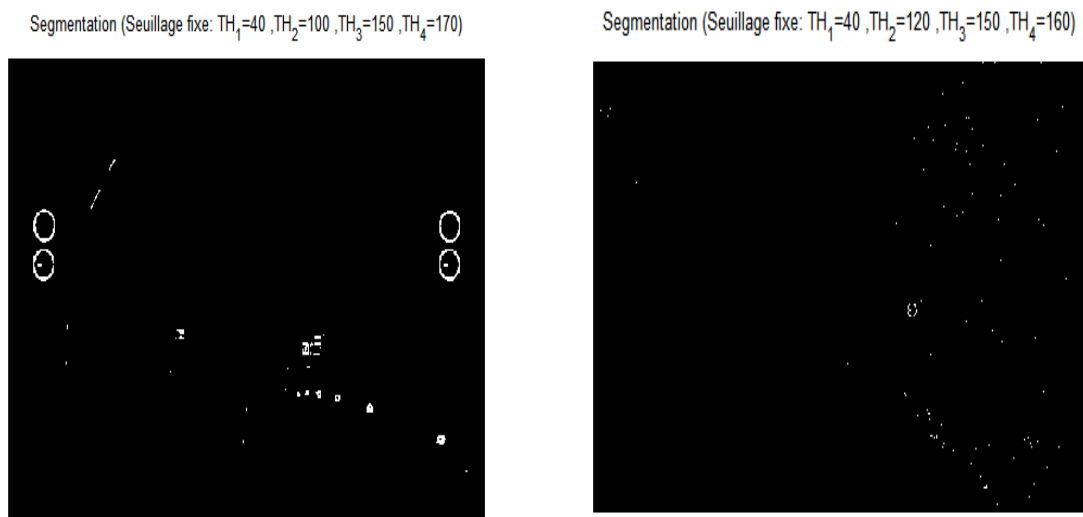
### III.5.1.3. Application des méthodes de segmentation sur les images de test

Nous avons appliqués les deux méthodes de segmentation sur les mêmes images pour noter la différence entre elles (voir Figure III.14).



**Figure III. 14.** Image de test.

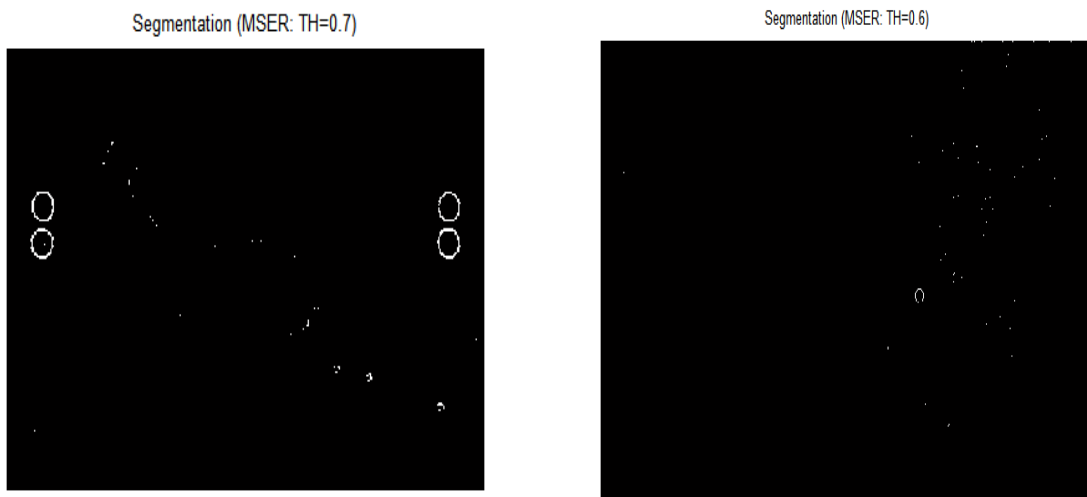
Les figures suivantes (III.15 à III.17) représentent la segmentation des images de test par les deux méthodes : seuillage fixe et segmentation par MSER.



**Figure III. 15.** Résultats de l'application de la méthode de seuillage simple sur les images de test.



**Figure III. 16.** Image normalisé de la méthode MSER sur les images de test.

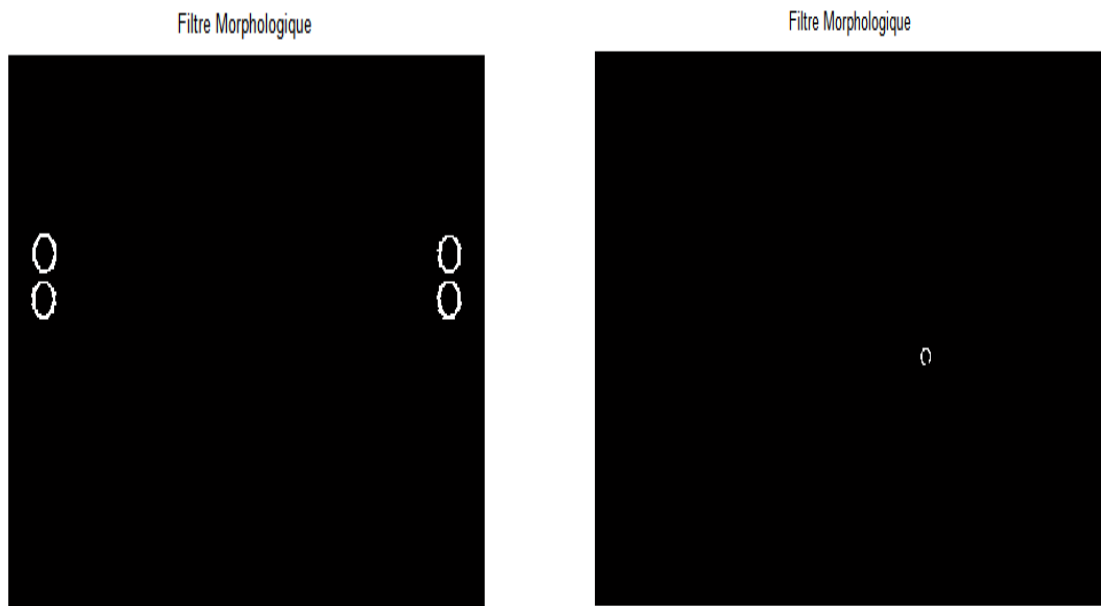


**Figure III. 17.** Résultats de l'application de la méthode de segmentation MSER sur les images de test.

La deuxième méthode basée sur MSER est la plus simple et la plus optimale par rapport à la méthode de seuillage simple, car nous modifions un seul seuil ( $0.5 \leq TH \leq 0.9$ ), ce qui la rend efficace et ses résultats sont dans un grand pourcentage correct, contrairement à la première méthode nous devons changer quatre seuils (chaque seuil entre 10 et 250), ce qui rend difficile à implémenter cette méthode pratiquement avec l'utilisation d'un Raspberry Pi et une caméra. Dans la suite de notre travail en utilise la méthode MSER, comme une méthode de segmentation.

Pour améliorer notre méthode de segmentation, nous proposons ensuite d'appliquer sur les images binaires une stratégie de filtrage morphologique. Le filtrage consiste à réduire de

manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image. Dans notre cas, nous allons effectuer une érosion (Ouverture morphologique). Cela nous permet de supprimer les pixels isolés qui ne correspondent pas à notre objet de couleur. Ensuite nous effectuons une dilatation qui nous permet de renforcer les groupes denses de pixels (donc notre objet). La figure suivante représente un exemple de filtrage morphologique.

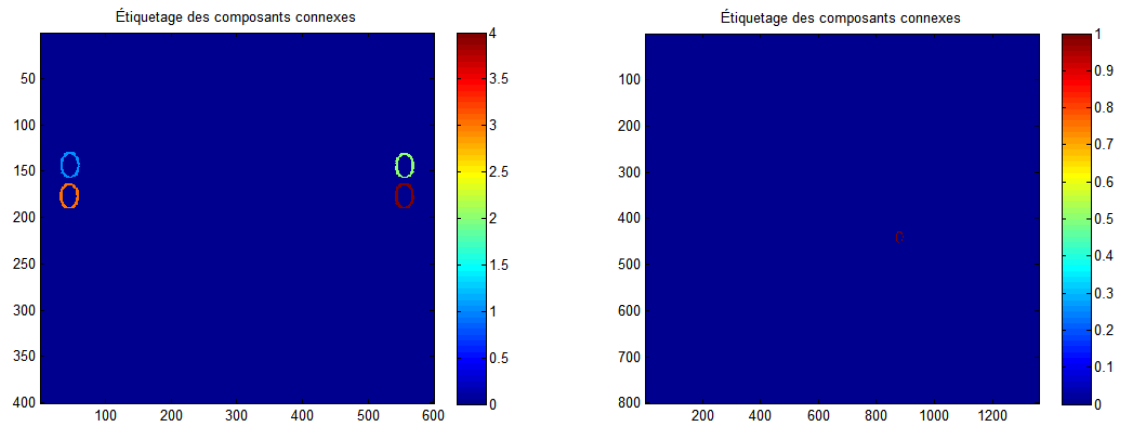


**Figure III. 18.** Résultats de l'application de la méthode de segmentation MSER et filtrage morphologique sur les images de test.

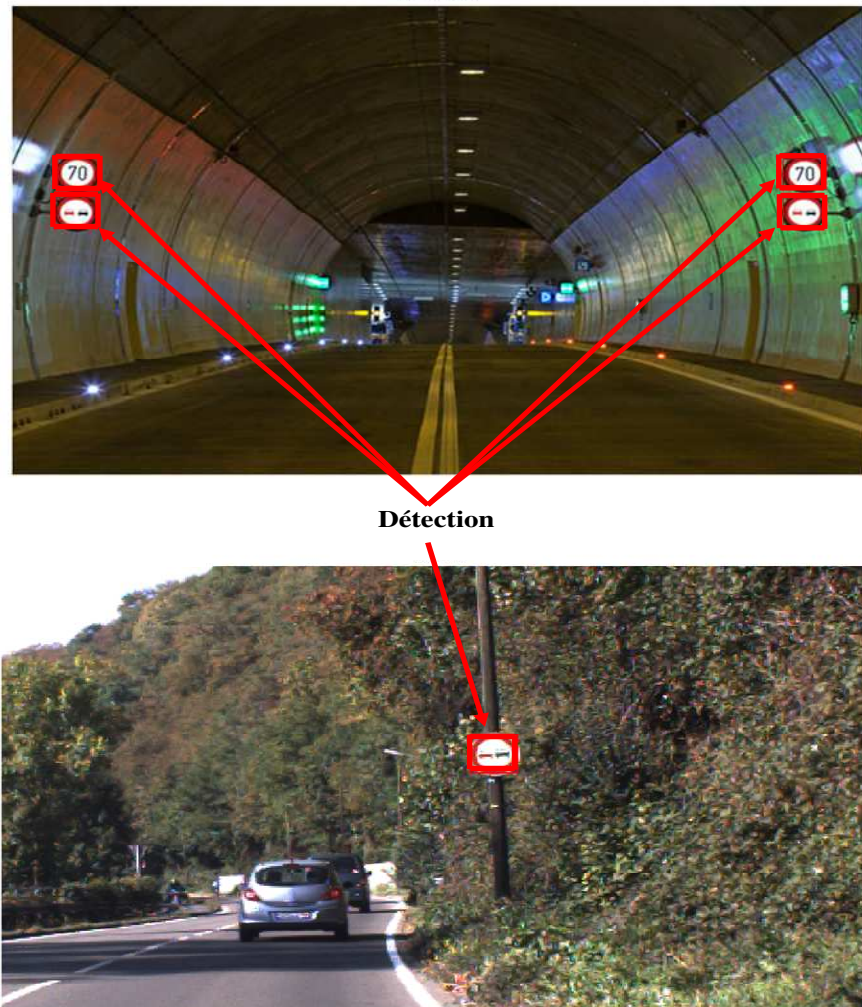
### III.5.2. Détection

Pour la séparation des régions détectées par la segmentation en utilise la méthode de l'étiquetage en composantes connexes (ECC). Cette technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe.

Un exemple de traitement ECC sur deux images de test est donné à la Figure III.19. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé. Chaque région étiquetée (Figure III.20) est ensuite classifiée par les méthodes de classification : réseaux de neurones artificiels et machine à vecteur de support (MLP, SVM).



**Figure III. 19.** Résultats de l'application de la méthode d'étiquetage des composants connexes sur les images de test.



**Figure III. 20.** Détection des panneaux de signalisations sur les images de test.

### III.6. Classification TSR par réseaux de neurones artificiels et machine à vecteurs de support

La classification est l'attribution d'une classe spécifique à un objet donné. Cette attribution a besoin d'un certain degré d'abstraction pour pouvoir extraire des généralités à partir des exemples dont on dispose. Dans cette partie nous avons utilisé deux types de classificateurs (MLP et SVM) pour faire la reconnaissance des panneaux de signalisations TSR (Traffic Sign Recognition).

Un système de classification opère en deux phases consécutives : une phase de conception et une phase de généralisation. La première phase consiste à choisir l'architecture optimale du modèle pour pouvoir s'adapter à différentes conditions d'entrée, l'application de cette phase est en mode hors ligne. Une fois l'apprentissage du modèle est terminé, il passe en phase de généralisation pour réaliser la classification en temps réel.

#### III.6.1. Évaluation des performances

##### II.6.1.1. Taux de classification

Dans le domaine de la classification, nous avons besoin d'un critère de précision. Dans notre cas nous avons choisi de calculer le taux de classification, il est donné par la formule suivante:

$$Acc(\%) = \frac{P}{N} \times 100 \quad (\text{III.27})$$

$P$  : Echantillons bien classés.

$N$  : Nombre d'échantillons des classes cibles.

##### II.6.1.2. Matrice de confusion

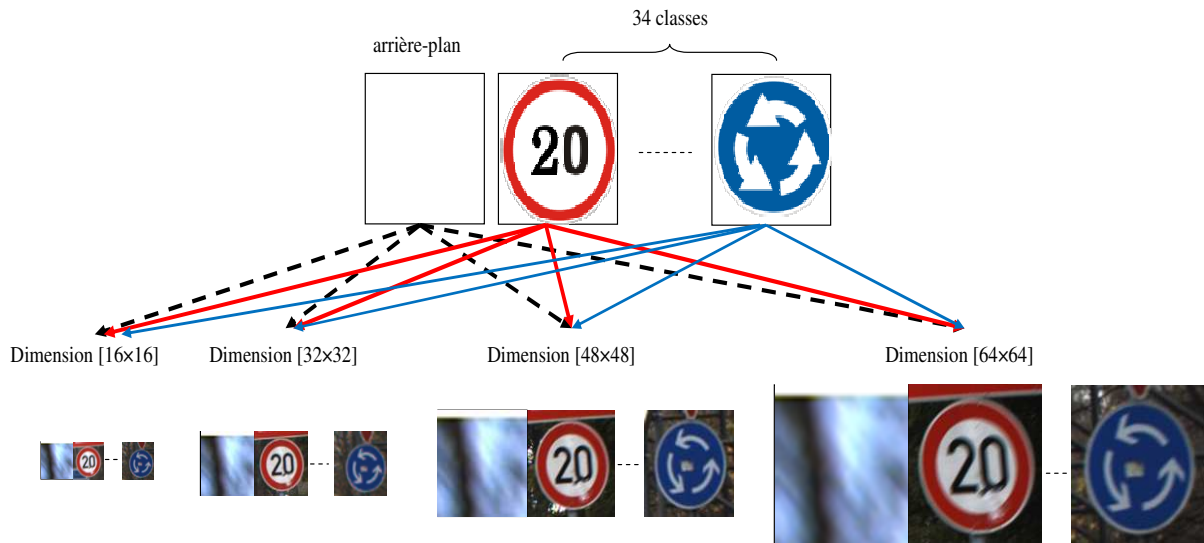
La matrice de confusion, dans la terminologie de l'apprentissage supervisé, est un outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence).

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classifier correctement. Un modèle sans erreurs aura ses résultats concentrés sur la diagonale de sa matrice de confusion.

#### III.6.2. Descripteur HOG

Avant de passer à la phase de classification (MLP, SVM), une étape de caractérisation par l'utilisation de descripteur HOG.

L'étape de caractérisation est décrite comme suit : L'image est redimensionnée dans notre approche suivant quatre tailles de bloc :  $16 \times 16$ ,  $32 \times 32$ ,  $48 \times 48$ , et  $64 \times 64$  pixels (Figure III.21), ensuite un descripteur HOG est effectuée sur chaque bloc.



**Figure III. 21.** Redimensionnés des images en blocs de dimension  $[16 \times 16]$ ,  $[32 \times 32]$ ,  $[48 \times 48]$ , et  $[64 \times 64]$ .

### III.6.3. Classification par le perceptron multicouche (Multi Layer Perceptron MLP)

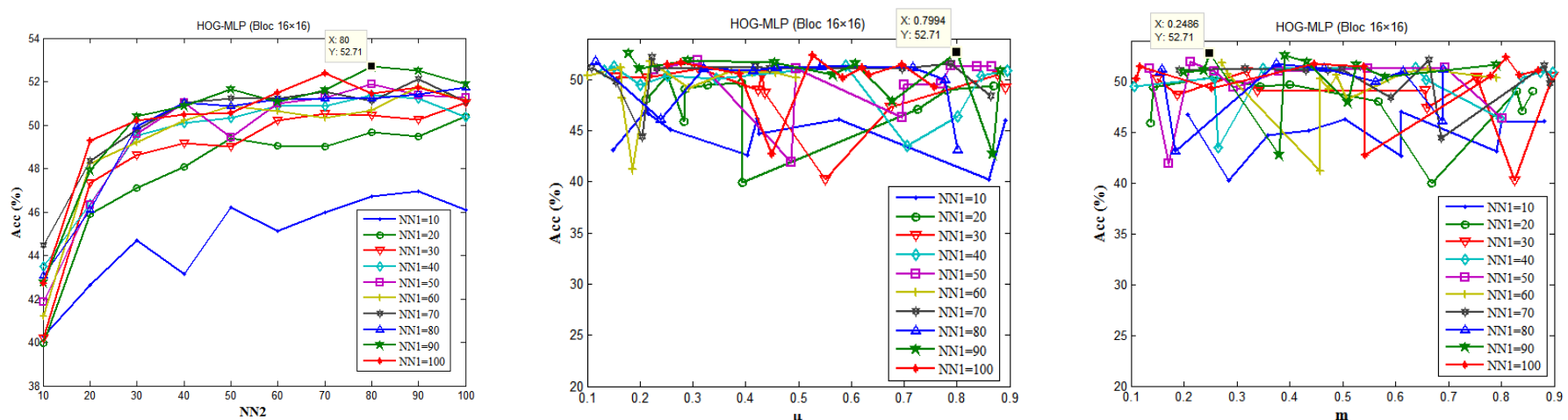
Pour la sélection des modèles, nous avons opté pour l'utilisation de la combinaison HOG et MLP. L'estimation de la meilleure architecture de l'MLP est obtenue par la technique de validation simple.

La variation des paramètres du modèle MLP est donnée par le tableau suivant:

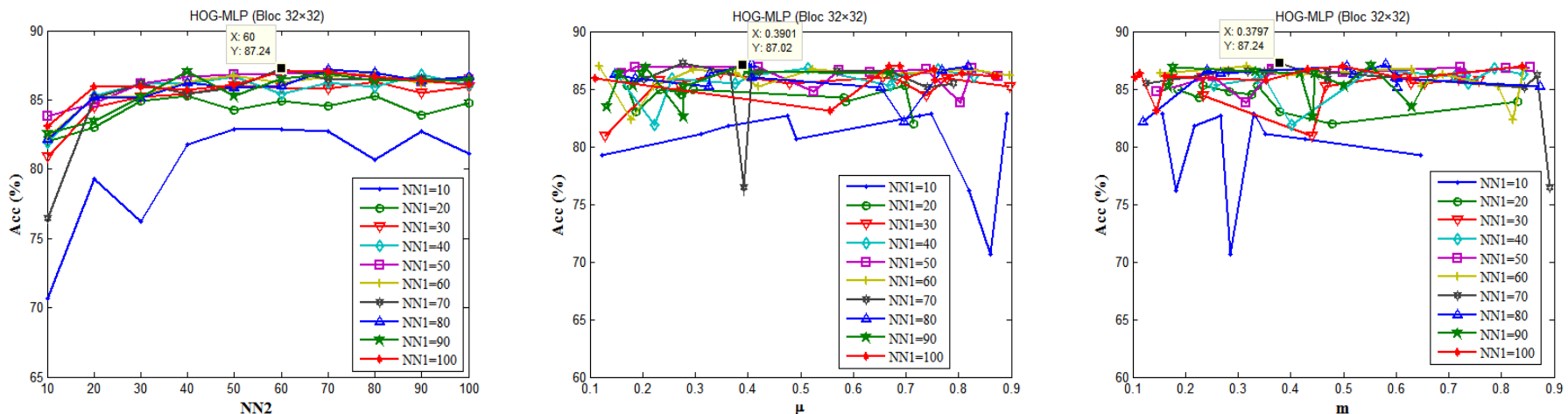
**Tableau III. 2.** Architecture du modèle MLP.

	Minimum	Maximum
Nombre de neurone dans le couche cachée 1 (NN1)	10	100
Nombre de neurone dans le couche cachée 2 (NN2)	10	100
Taux d'apprentissage ( $\mu$ )	0.1	0.9
Momentum ( $m$ )	0.1	0.9

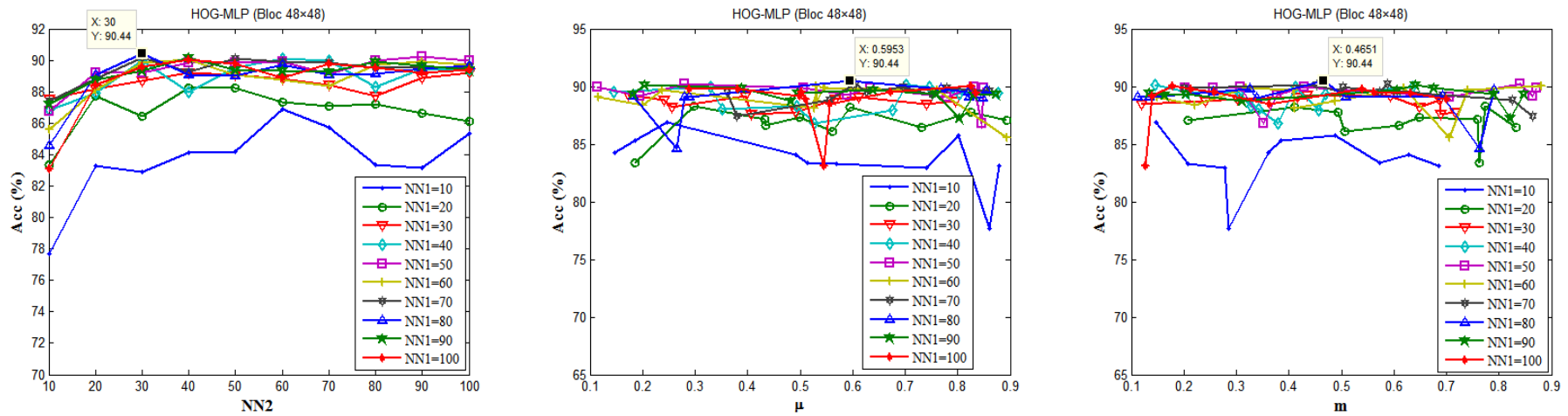
Pour les différents redimensionnements des images et pour chaque combinaison des paramètres de l'MLP, les précisions ( $Acc$ ) sur la base de validation sont sauvegardées en fonction de NN1, NN2,  $\mu$  et  $m$ , comme montres les figures suivantes (Figure III.22 à III.25).



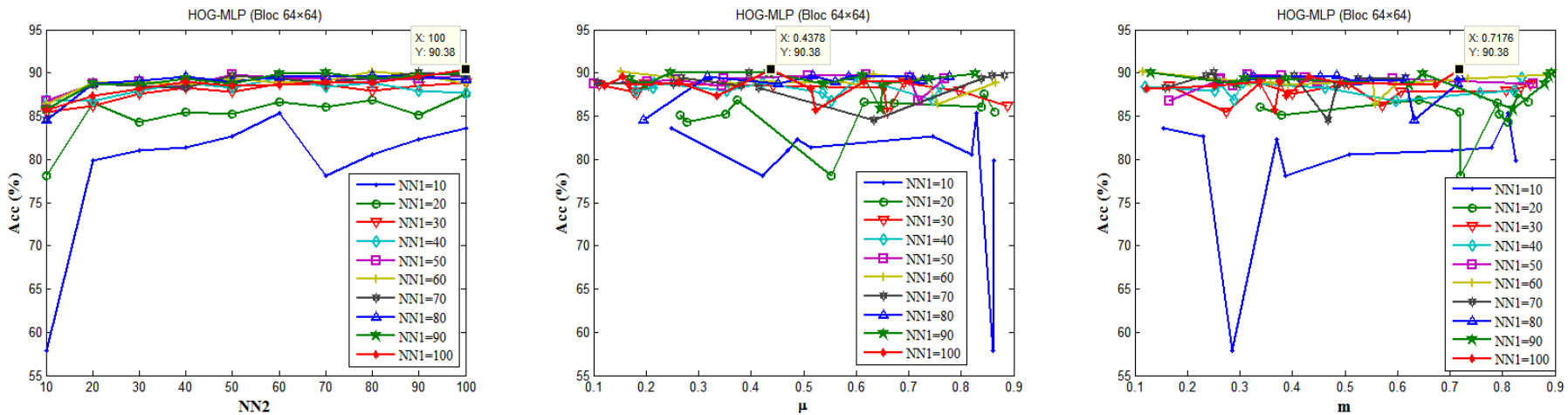
**Figure III. 22.** Précision (*Acc*) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [16×16], le descripteur HOG et le classifieur MLP.



**Figure III. 23.** Précision (*Acc*) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [32×32], le descripteur HOG et le classifieur MLP.



**Figure III. 24.** Précision (*Acc*) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [48×48], le descripteur HOG et le classifieur MLP.



**Figure III. 25.** Précision (*Acc*) en fonction des nombres de neurones dans les couches cachées, taux d'apprentissage, momentum, en utilisant : la dimension de l'image [64×64], le descripteur HOG et le classifieur MLP.

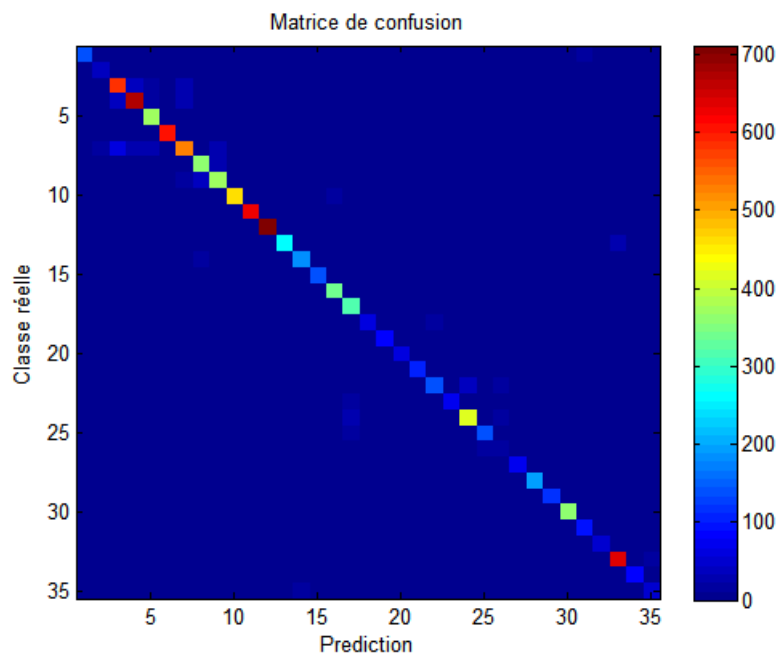
Dans la figure III.22, nous avons représenté toutes les valeurs de précision, pour le nombre de neurones dans la première couche cachée est 90, dans la deuxième couche cachée 80, le taux d'apprentissage est 0.8 et le momentum 0.25, le réseau donne une précision maximale égale à 52.71% sur l'ensemble de validation.

Dans les figures III.23, III.24 et III.25 nous avons simulé la même expérience mais avec des redimensionnements des images : [32×32], [48×48] et [64×64]. Le tableau III.3, illustre les meilleurs paramètres pour chaque dimension de l'image. Il est clair que pour un bloc [48×48] le modèle donne une précision maximale.

**Tableau III. 3.** Meilleurs paramètres du modèle MLP.

	Bloc [16×16]	Bloc [32×32]	Bloc [48×48]	Bloc [64×64]
NN1	90	80	80	100
NN2	80	60	30	100
$\mu$	0.7994	0.3901	0.5953	0.4378
$m$	0.2486	0.3797	0.4651	0.7176
Acc(%)	52.71	87.24	<b>90.44</b>	90.38

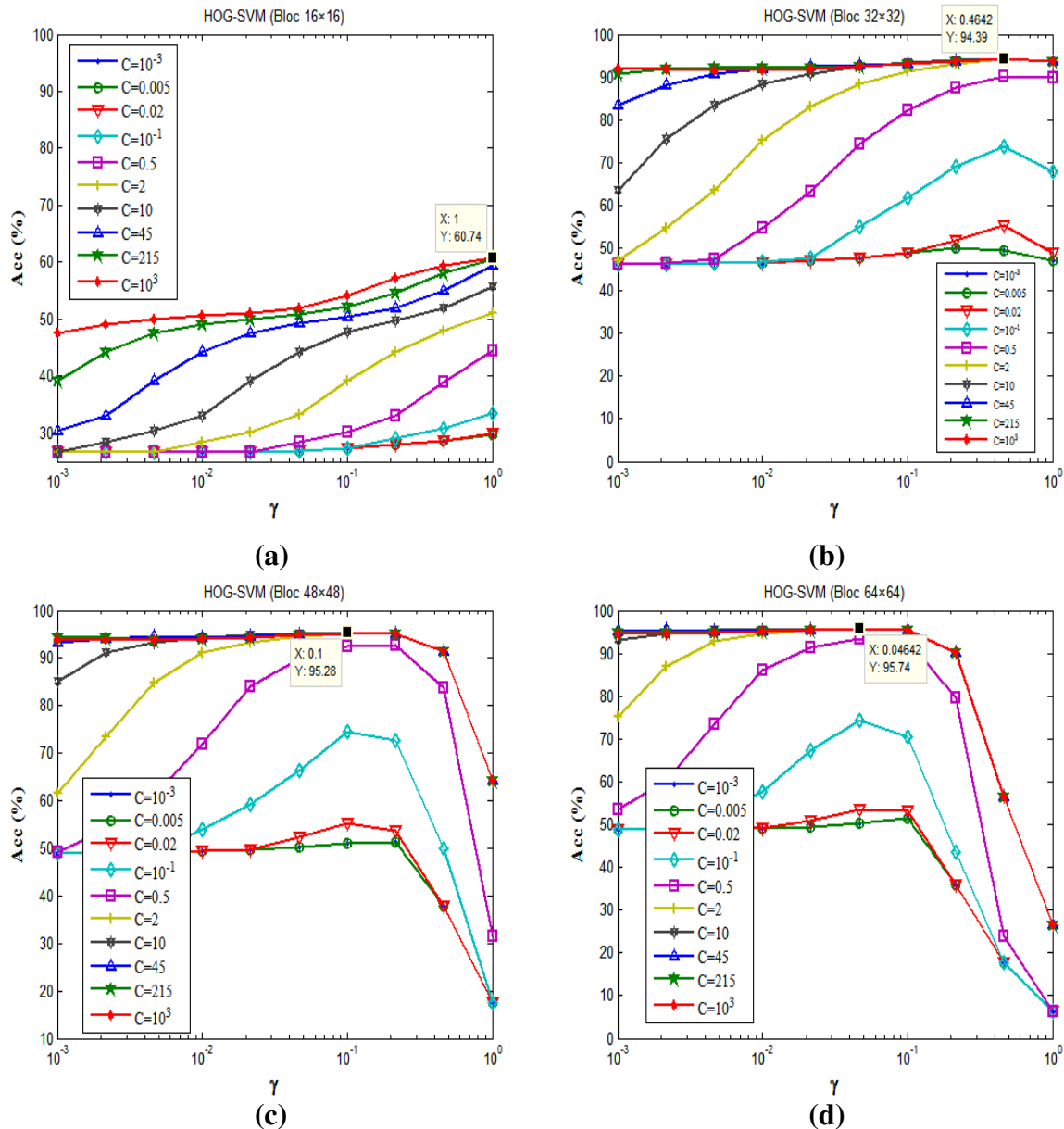
Une fois le meilleur modèle est obtenu (le nombre de neurones optimale=80, 30, le coefficient d'apprentissage=0.5953 et le momentum=0.4651), nous utilisons l'ensemble de test pour tester les performances du classificateur MLP. La Figure III.26 représente la matrice de confusion pour l'ensemble de test, la précision sur le test est 88.26%.



**Figure III. 26.** Matrice de confusion sur l'ensemble de test (HOG-MLP).

### III.6.4. Classification par SVM (Machines à vecteurs de support)

La figure III.27 représente le taux de classification en fonction des paramètres à optimiser:  $C$  et  $\gamma$ , avec l'utilisation de 4 blocs différents [16×16], [32×32], [48×48], et [64×64].



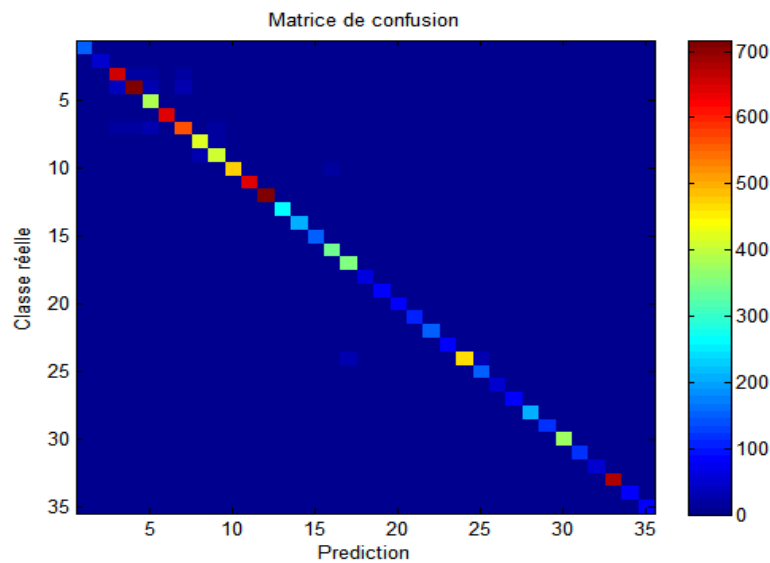
**Figure III. 27.** Précision ( $Acc$ ) en fonction de  $C$  et  $\gamma$ . La dimension de l'image (a) [16×16], (b) [32×32], (c) [48×48], et (d) [64×64], le descripteur HOG et le classifieur SVM.

Le tableau III.4 illustre les combinaisons des paires ( $C$ ;  $\gamma$ ) qui donne les meilleures précisions sur l'ensemble de validation. Le meilleur taux de classification est obtenu avec les combinaisons : Bloc [64×64],  $C=2$ , et  $\gamma=0.0464$ .

Une fois le meilleur modèle est obtenu, nous utilisons l'ensemble de test pour tester les performances du classificateur HOG-SVM. La figure III.28 représente la matrice de confusion sur l'ensemble de test. La précision sur l'ensemble de test est 94.96%.

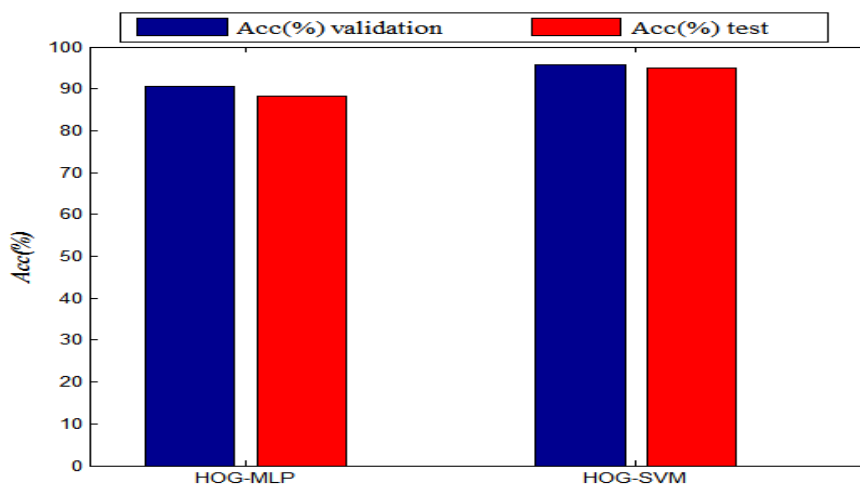
**Tableau III. 4.** Meilleurs paramètres du modèle SVM.

	Bloc [16×16]	Bloc [32×32]	Bloc [48×48]	Bloc [64×64]
$C$	1000	10	10	2
$\gamma$	1	0.4642	0.1	0.0464
Acc(%)	60.74	94.39	95.28	<b>95.74</b>



**Figure III. 28.** Matrice de confusion sur l'ensemble de test (HOG-SVM).

La figure suivante présente une comparaison entre les meilleurs résultats obtenus par les deux méthodes de classification (MLP, SVM) sur les ensembles de validation et de test. Nous remarquons que le classificateur SVM donne des résultats significativement meilleurs que la méthode MLP.



**Figure III. 29.** Etude comparative entre HOG-MLP et HOG-SVM.

### III.7. Conclusion

L'intérêt de la classification des images en utilisant l'intelligence artificielle cela fait partie de système d'alarme de notre projet. Ils permettent des solutions à certaines problématiques comme l'amélioration de la sécurité routière.

Dans ce chapitre, nous avons expliqué les deux phases essentielles pour la détection et reconnaissance automatique des panneaux de signalisations (TSR); plus précisément : la phase de calibrage et la phase en temps réel. D'après les résultats obtenus, l'algorithme basé sur le descripteur HOG avec un bloc de dimension [64×64] et le classifieur machine à vecteurs de support (SVM) semble être le plus efficace pour la reconnaissance TSR. Les meilleures combinaisons entre la segmentation, détection, et la classification sont examinés dans le chapitre IV.

Nous sommes intéressés par la méthode de reconnaissance TSR avec l'utilisation de la carte Raspberry Pi 3 et une communication radiofréquence qui sera une partie de notre système de communication entre les véhicules (VNET) dans le chapitre qui suit.

## *Chapitre IV*

---

### *Réalisation pratique*

## IV. 1. Introduction

Dans les chapitres précédents on a travaillé sur le côté théorique du projet, en revanche dans ce dernier chapitre on travaille sur le côté hardware. En effet, dans ce chapitre nous allons décrire les composants utilisés et leurs fonctions dans le système VNET. La conception matérielle et logicielle de notre travail sera présentée et expliquée, ainsi que les différents tests effectués. Ce chapitre présente le prototypage de notre système de communication entre les véhicules (VNET) en utilisant les différents types de communication sans fil et les méthodes de l'intelligence artificielle.

## IV. 2. Architecture globale du système VNET

Dans notre travail, on considère que chaque véhicule comporte comme un réseau de communication sans fil (nRF, GSM, GPS), qui contient plusieurs nœuds capteurs dont chacun dispose de sa propre fonction, d'une base et d'une interface pour l'utilisateur (conducteur).

Dans les sections suivantes, nous allons d'abord détailler l'architecture globale de ce système à réaliser. Ensuite, nous allons expliquer le fonctionnement général du système en l'occurrence celui des différentes unités composant les nœuds capteurs et le rôle de chaque élément.

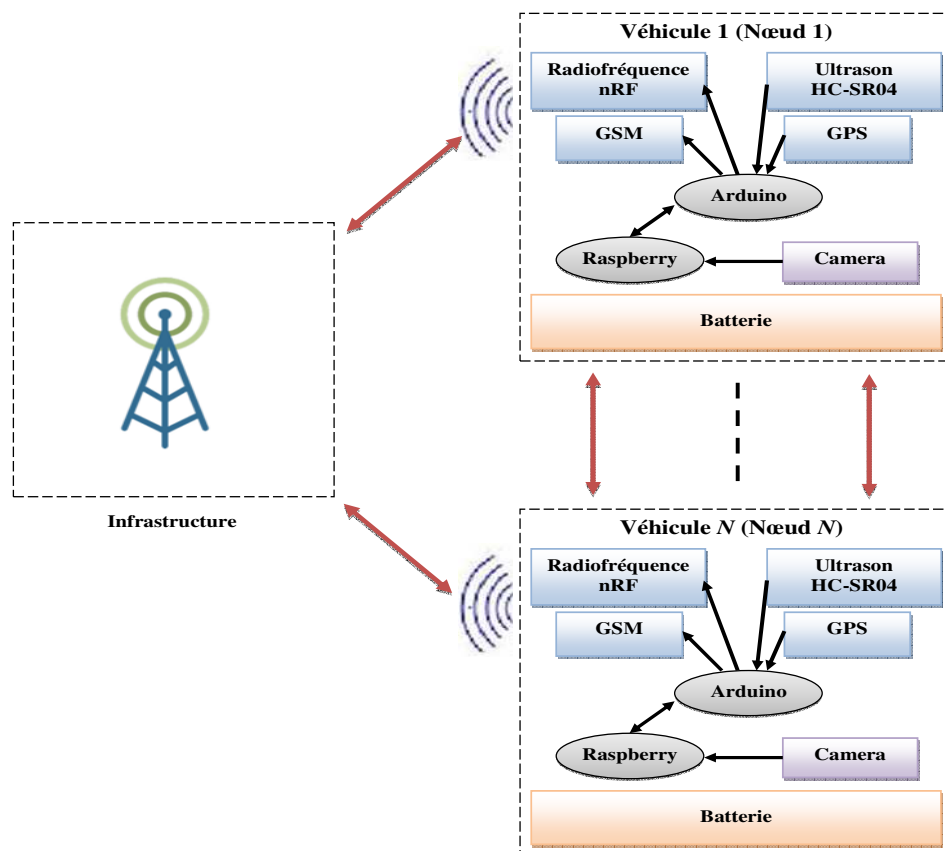
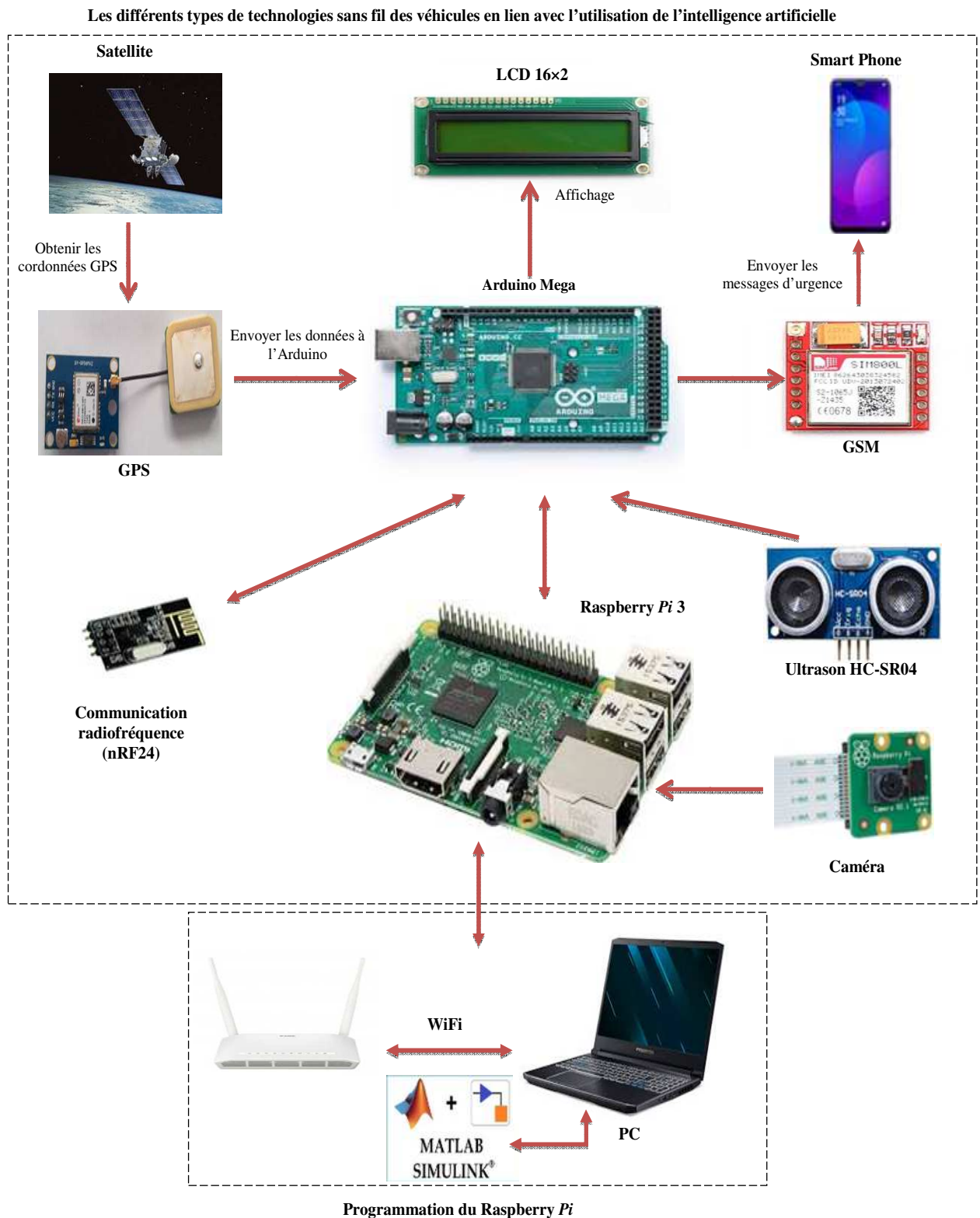


Figure IV. 1. Architecture globale du système VNET.

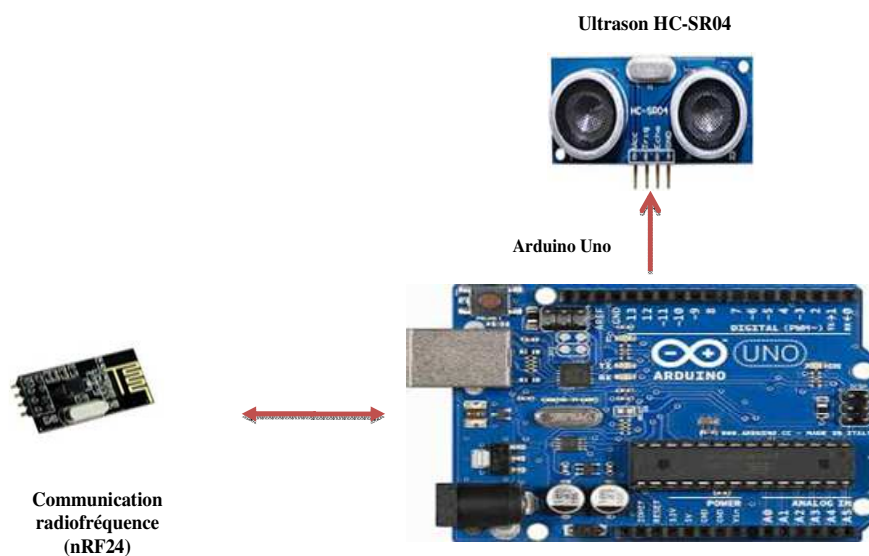
Les éléments de base constituant un nœud capteur sont illustrés dans la figure IV.1. Chaque véhicule est équipé de différents types de technologies sans fil en lien avec l'utilisation de l'intelligence artificielle (Figure IV.2).



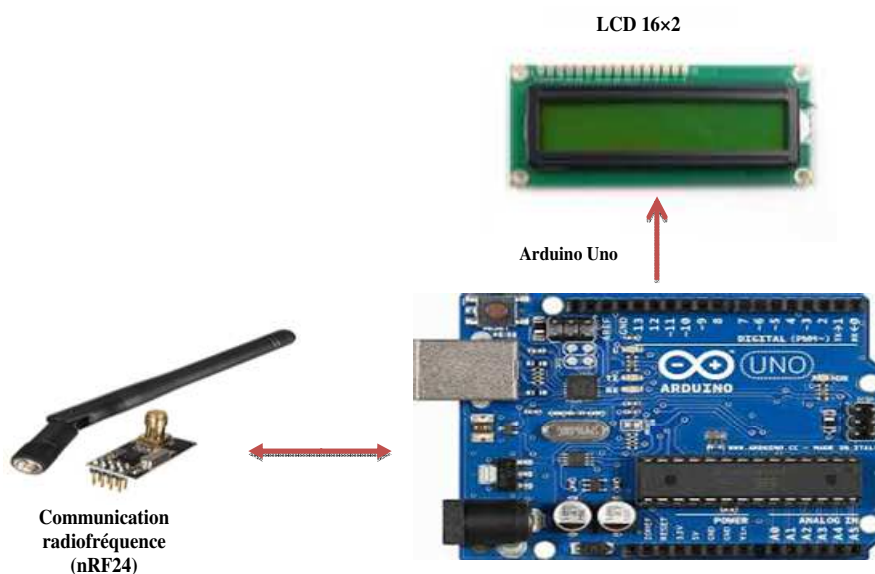
**Figure IV. 2.** Schéma globale du système de communication équipé dans les véhicules.

Pour tester le fonctionnement de notre projet de communication entre les véhicules, nous avons proposé d'utiliser un prototype, constitué de deux robots mobiles et infrastructure. Le premier robot décompose de tous les éléments essentiels de notre système (Arduino Mega, afficheur, ultrason, nRF, GPS, GSM) comme représentée dans la Figure IV.2, tandis que le deuxième robot décompose d'Arduino Uno, radiofréquence et ultrason. Ce dernier est utilisé pour tester la transmission et la réception entre les différentes véhicules (Figure IV.3).

Pour l'infrastructure, on affiche la position du premier véhicule par les coordonnées GPS et pour l'induction des panneaux de signalisation avec l'utilisation de l'intelligence artificielle (Figure IV.4).



**Figure IV. 3.** Véhicule équipé par un ultrason et une communication radiofréquence.

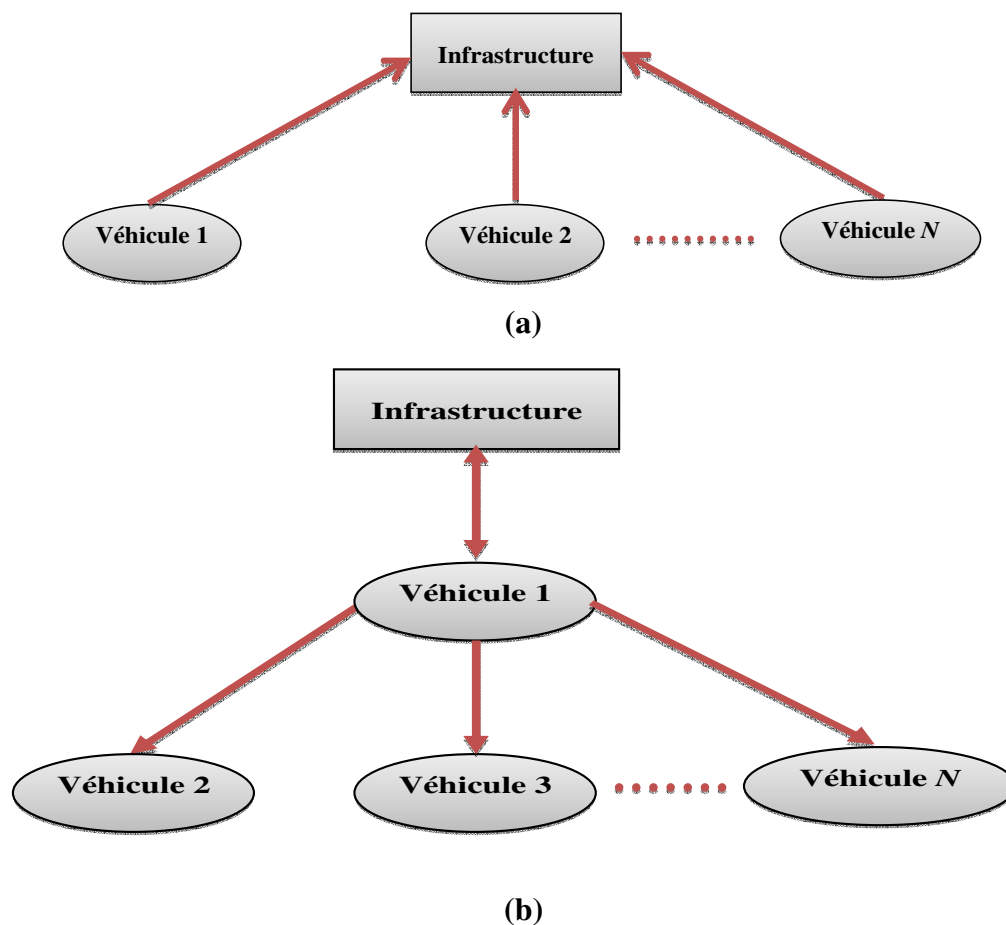


**Figure IV. 4.** Le prototype de l'infrastructure pour tester notre système.

Notre système de communication entre les véhicules (VNET) est divisé en trois blocs: communication véhicules à infrastructure, communication véhicule-à-véhicule et le système d'alerte (message d'urgence).

#### IV.2.1. Communication véhicules à infrastructure

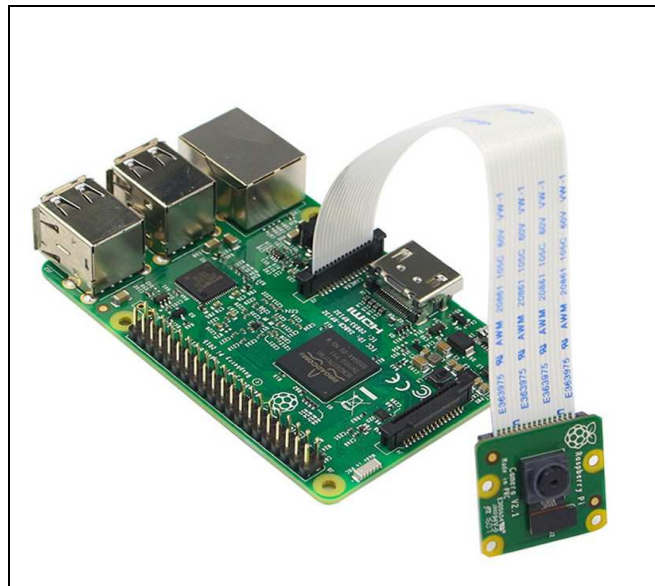
Dans notre mémoire, la communication véhicule à infrastructure est basé sur la transmission radiofréquence, nous avons utilisé un module nRF24L01 dans l'infrastructure et les véhicules, ce dernier fonctionne comme un émetteur et récepteur en même temps. Cette communication est très utile pour suivre les mouvements d'un véhicule est basé sur un échange d'information entre les véhicules et l'infrastructure en temps réel à tout moment dans une zone limité et il permet également de connaître le nombre de véhicule connectées dans cette zone. Le nRF24L01 déterminer quel véhicule est entré en premier dans cette zone, après avoir identifié et connu le véhicule numéro 1 de la zone, l'infrastructure l'informera ou celle-ci commencera à envoyer des informations aux autres véhicules (accident, danger, les panneaux de signalisations, ...). Le GPS situé dans les véhicules il nous permet de connaître la position et la direction du véhicule de plus, nous pouvons voir s'il ya une collision dans cette zone.



**Figure IV. 5.** Organigramme proposé de la communication (a) véhicules à infrastructure et (b) véhicule à véhicule.

### IV.2.2. Communication véhicule à véhicule

Dans ce bloc la communication entre les véhicules est réalisé en utilisant la technologie radiofréquence (nRF24L01) qui nous permet de la diffusion des informations comme la distance entre une véhicule et l'autre, cette distance est calculée par le capteur ultrason HC-SR04 qui est situé à l'avant de chaque véhicule, et dans le cas d'un arrêt brusque du véhicule. Dans ce bloc on trouve aussi un système d'intelligence artificielle (un Raspberry Pi 3 et une caméra) situé à l'avant de chaque véhicule pour la reconnaissance automatique des panneaux de la route et on transmette cette reconnaissance à la voiture suivante pour l'indiquer à travers le nRF24L01, comme représenté dans la figure suivante.



(a)

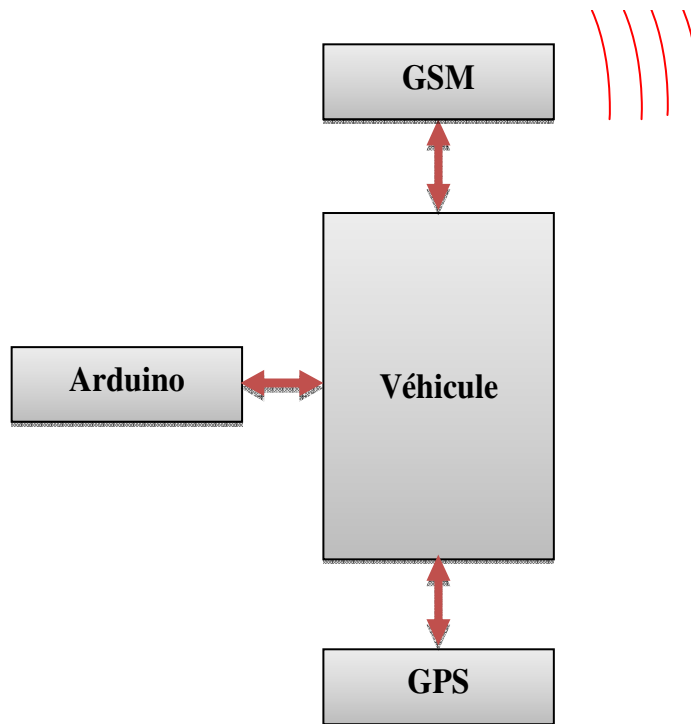


(b)

**Figure IV. 6.** Système de reconnaissance automatique des panneaux de signalisations, (a) Raspberry et camera, (b) Système de transport intelligent.

### IV.2.3. Système de message d'urgence

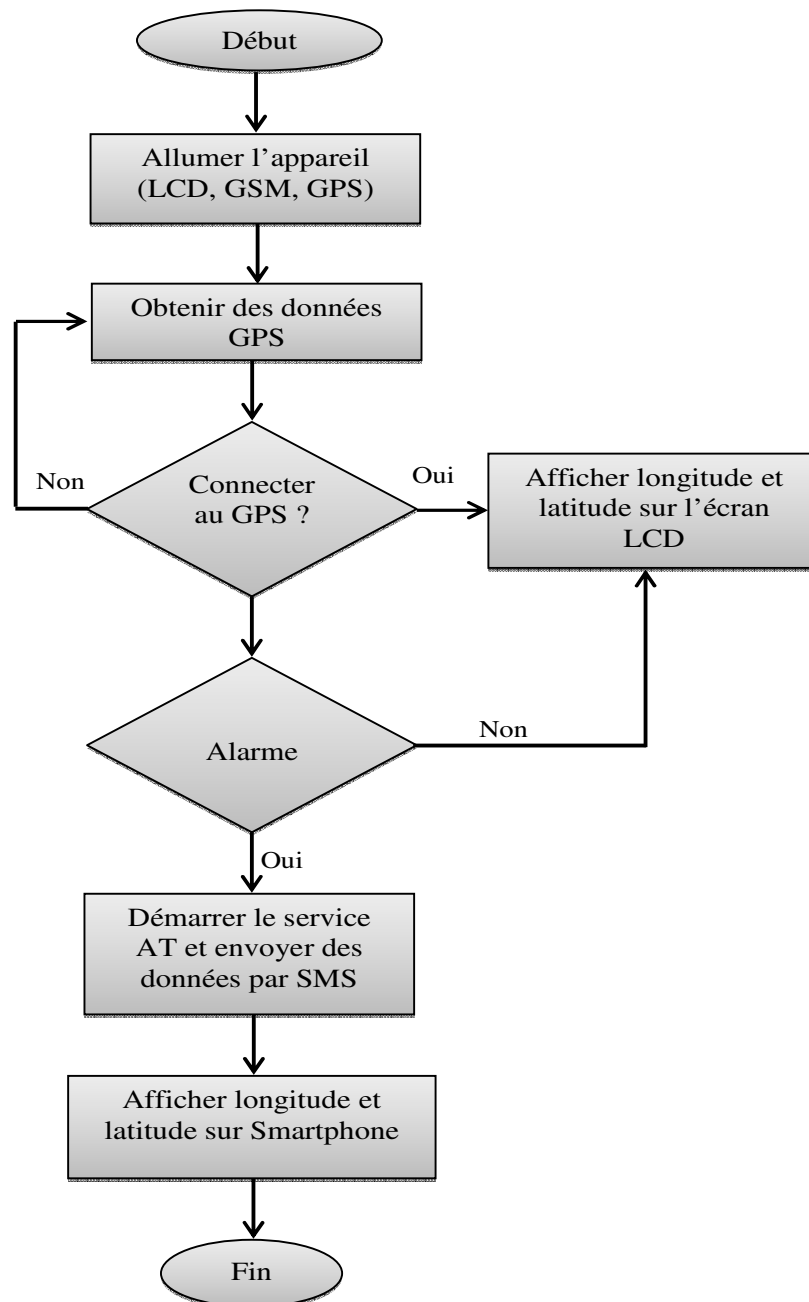
Pour générer et envoyer un message d'urgence, le véhicule doit connaître sa propre position via un système de positionnement GPS qui utilise le service par satellite. Une fois sa position connue, le véhicule a besoin d'une unité de traitement Arduino Mega qui peut prendre son emplacement pour envoyer la chaîne de données requise (les positions géographiques : latitude et longitude, les informations de conducteur). Une fois le message d'urgence généré, notre système doit transmettre ce message en utilisant le module GSM à un ou plusieurs contacts et aux services adaptés à la situation par exemple la charge des problèmes médicaux urgents, les pompiers, ...etc. Ce message est généré et envoyé en cas de risque ou d'accident. La figure suivante représente un schéma synoptique de notre système de message d'urgence.



**Figure IV. 7.** Schéma synoptique d'un système de message d'urgence.

La Figure IV.8 illustre l'organigramme de fonctionnement du système de message d'urgence basé sur un Arduino. Un interrupteur a pour but de couper ou d'alimenter notre circuit électrique composé de l'Arduino et les différents shields (nRF, GPS, GSM et LCD). La broche *TX* du GPS est connectée au microcontrôleur de l'Arduino et les broches *TX* et *RX* du GSM sont connectées aux ports série pour communiquer avec le GSM SIM800L. Le récepteur GPS localise latitude et longitude du véhicule à l'aide du service satellite. Si le GSM est prêt, les données de position sont envoyées sous forme de SMS comme un message

d'urgence en cas de risque ou accident, et en même temps l'écran LCD affiche l'emplacement sous forme de latitude et de longitude.



**Figure IV. 8.** Organigramme proposé du système de message d'urgence.

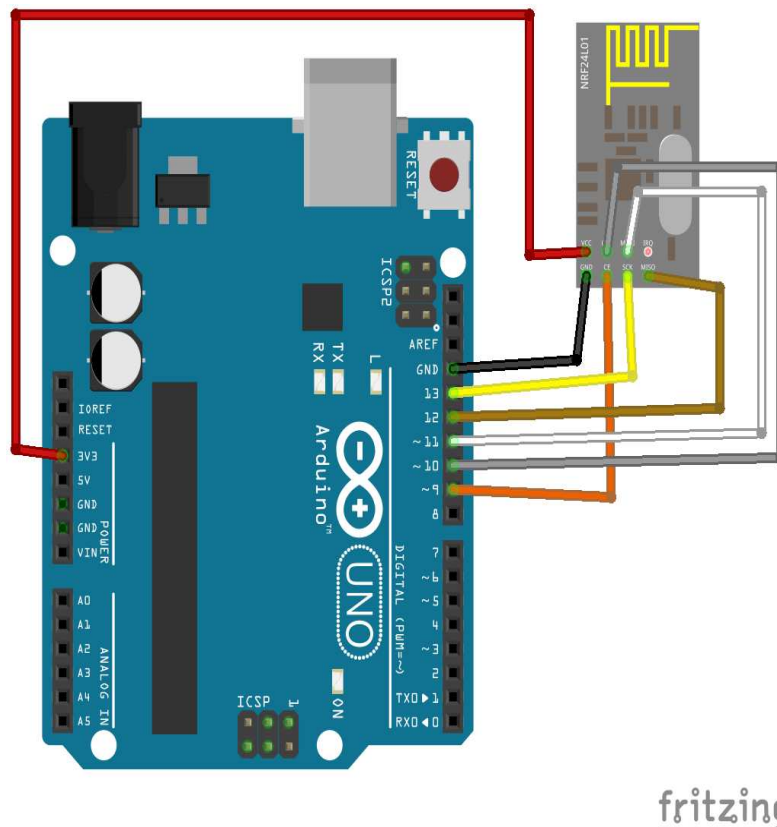
### IV. 3. Fonctionnement du système

Afin de réaliser le prototype de notre système de communication entre les véhicules, il est important d'avoir : infrastructure, et deux robots mobile. Pour ce système nous avons besoin un émetteur, un récepteur radiofréquence, un capteur de distance, système de localisation GPS, un système GSM, et un système de reconnaissance automatique à base de Raspberry Pi.

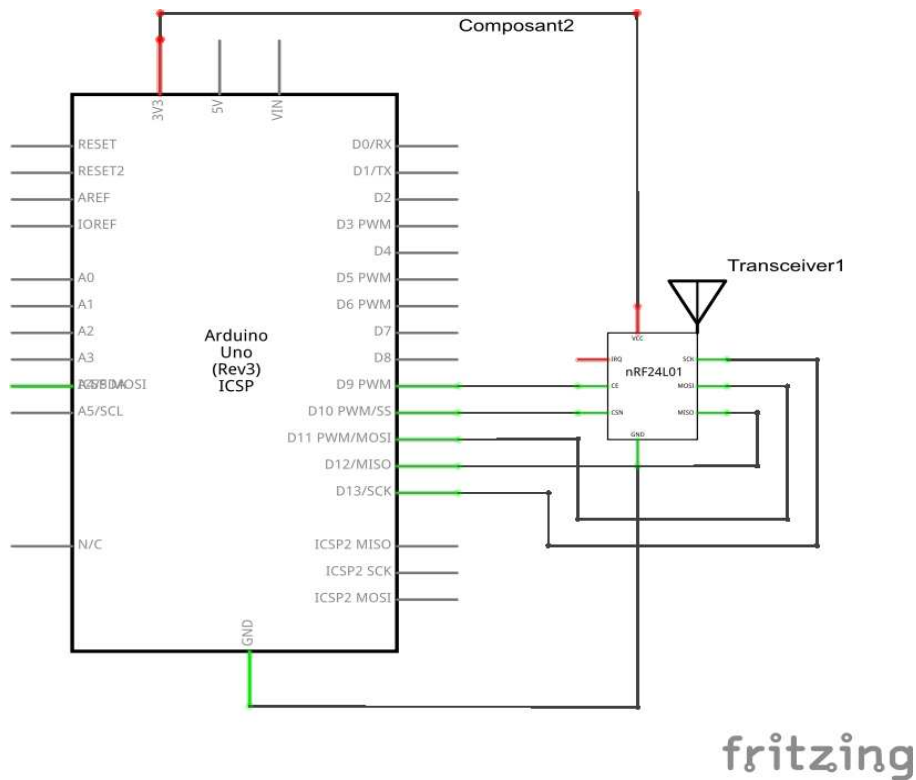
La réalisation matérielle est faite en premier lieu, chaque module de notre montage est réalisé et testé séparément, les montages sont d'abord construits sur des «breadboard» ou cartes de montage expérimental. Après les avoir expérimentés et adoptés séparément, nous les avons regroupés et réalisés sur multi circuits (module GPS, module GSM, Afficheur LCD 16x2, capteur HC-SR04, transmission radiofréquence, Raspberry Pi avec une camera). Ces derniers sont faits en utilisant le logiciel de dessin des schémas électroniques «*fritzing*».

#### IV.3.1. Modules nRF24L01

Ces modules sont des *transceiver*, c'est à dire qu'ils peuvent émettre et réceptionner en radio fréquence de 2.4 GHz. Le module nRF24L01 sert à transmettre les informations entre les véhicules et l'infrastructure ou vice versa en utilisant un Arduino Uno ou Mega, cette communication nous permet de connaître le nombre de véhicules connectés dans une zone limitée et de détecter le premier véhicule dans la couverture de cette zone, et l'échange d'information entre les véhicules comme la distance entre une voiture et la voiture suivante et aussi l'indique qu'il y a un panneau. Un exemple de l'émetteur ou récepteur nRF24L01 est illustré dans les deux figures suivantes. La première figure représente le branchement de nRF, et la deuxième représente le circuit électrique de montage réalisé.



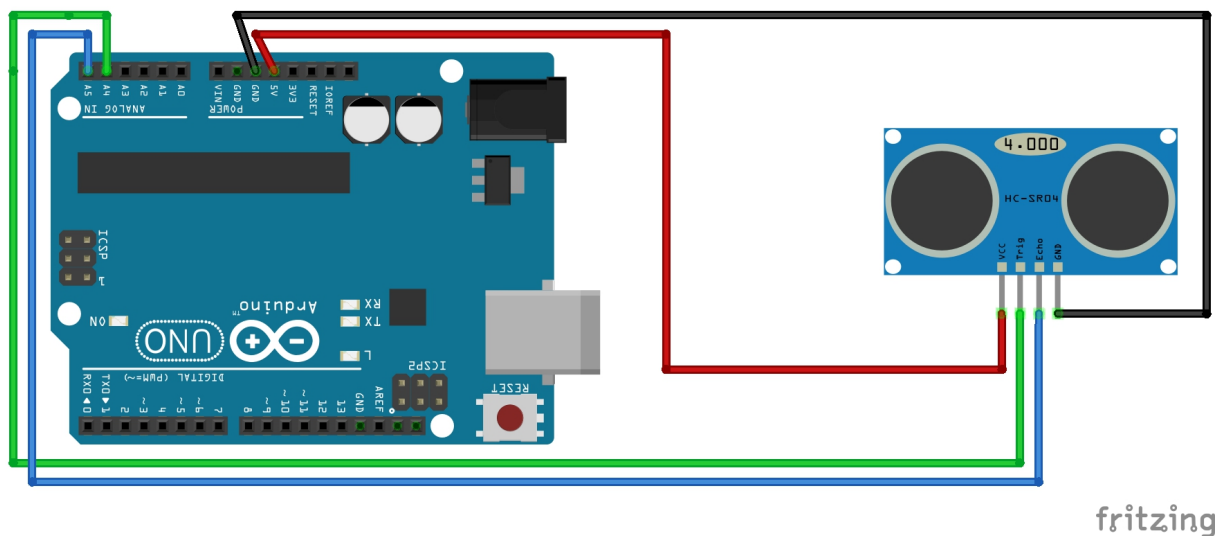
**Figure IV. 9.** Branchement de radiofréquence nRF24L01 avec la carte Arduino Uno.



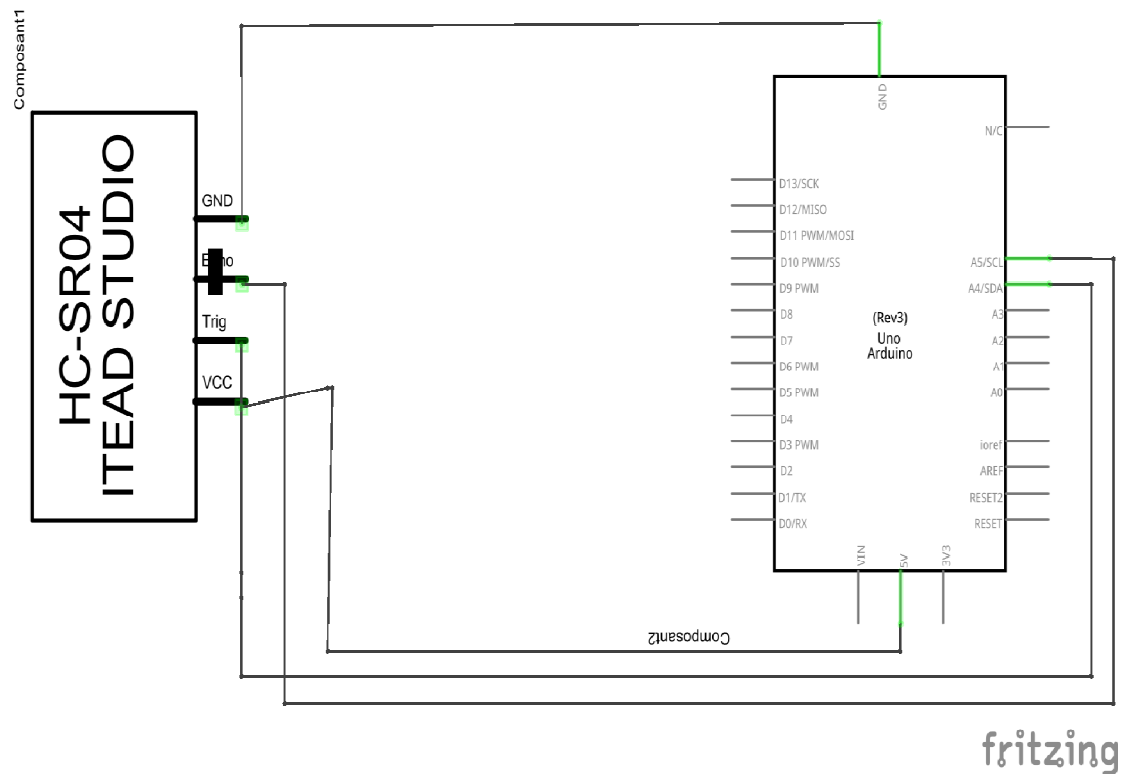
**Figure IV. 10.** Circuit électrique de module radiofréquence nRF24L01 avec la carte Arduino Uno.

### IV.3.2. Capteurs HC-SR04

Nous avons placé un capteur ultrason HC-SR04, il est sur le côté avant de notre prototype (robot mobile) qui permet de calculer la distance entre deux véhicules. Les capteurs ultrasons fonctionnent en mesurant le temps de retour d'une onde sonore émise par le capteur. La vitesse du son étant à peu près stable, on en déduit la distance par rapport l'obstacle. Pour cette partie, nous avons assemblé la carte Arduino Uno et le capteur sonar à Ultrason HC-SR04, comme montré dans les figures IV.11 et IV.12.



**Figure IV. 11.** Schéma du capteur ultrason avec la carte Arduino Uno.



**Figure IV. 12.** Circuit électrique du capteur ultrason avec la carte Arduino Uno.

### IV.3.3. Modules GPS

Global Positioning System (GPS) est un système de navigation par satellite composé d'au moins 24 satellites. Le GPS fonctionne dans toutes les conditions météorologiques, partout dans le monde, 24h/24h.

Il existe plusieurs Shields Arduino GPS mais dans notre projet nous avons utilisé le module GPS avec la référence GY-NEO6MV3 avec une antenne externe, donc nous devons le souder (voir figure IV.13). Ce module prend en charge le débit de 4800 bps à 230400 bps avec un baud par défaut de 9600 est utilisé pour recevoir les données des satellites. Les données sont envoyées à l'Arduino qui lit les coordonnées en extrayant la chaîne \$GPGGA (Global Positioning system fix data).

Il y a une LED sur le module GPS GY-NEO6MV3 qui indique l'état de la position fixe. Il clignotera à différents taux selon l'état dans lequel il se trouve (voir la figure IV.13).

- ✓ Pas de clignotement : signifie qu'il recherche des satellites.
- ✓ Clignote toutes les 1 s: signifie que la correction de position est trouvée.

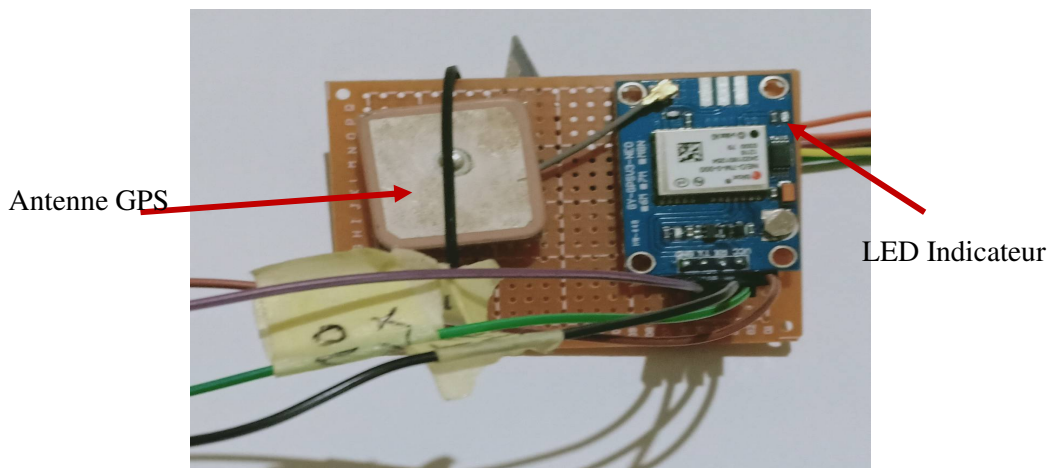


Figure IV. 13. Module GPS (GY-NEO6MV3).

#### IV.3.3.1. Rôle du module GPS dans le projet

GPS à bord du véhicule calcule en temps réel la position de ce dernier et renvoie l'information à infrastructure pour détecter le nombre des véhicules existant dans cette zone et sa direction, et envoie les coordonnées du véhicule par le réseau GSM en cas d'alarme, et aussi l'affichage de la position sur l'afficheur LCD 16x2.

#### IV.3.3.2. Branchement du module GPS

Le GPS utilise 4 pins, VCC et GND et ensuite TX et RX qui sont connectés à un Arduino. Ces deux derniers pins sont respectivement le signal de transmission et celui de réception. Les Figures IV.14 et IV.15 représentent le branchement et le circuit électrique de module GPS avec l'arduino Mega.

Tableau IV. 1. Affectation des broches GPS.

Nom de broche	Rôle de la broche
GND	La broche de terre (0V).
TX (émetteur)	Utilisé pour la communication série.
RX (récepteur)	Utilisé pour la communication série.
VCC	Alimente le module GPS (+5V).

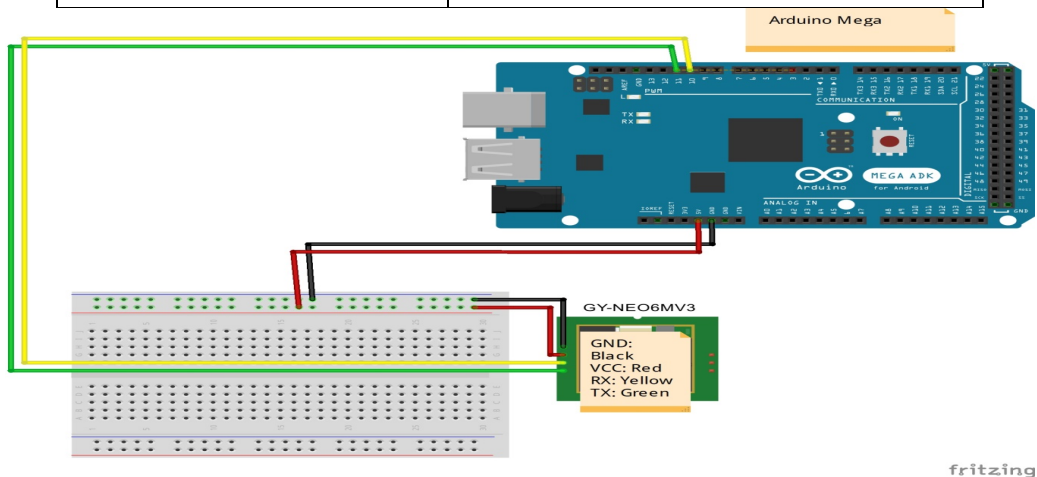
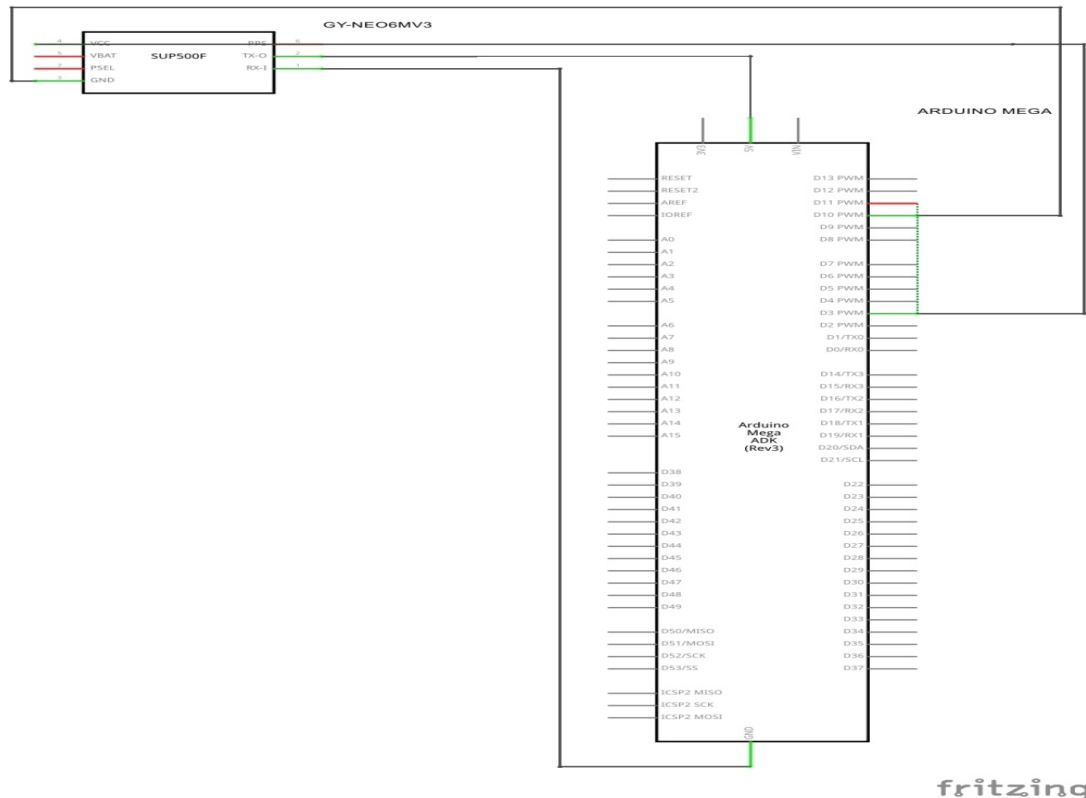


Figure IV. 14. Branchement de module GPS avec Arduino Mega.



**Figure IV. 15.** Circuit électrique du module GPS avec la carte Arduino Mega.

#### IV.3.3.3. Programmation de Module GPS

Nous allons utiliser la bibliothèque `#include "TinyGPS++.h"` qui extrait automatiquement les coordonnées GPS à partir des trames NMEA.

La bibliothèque `#include "SoftwareSerial.h"` a été développée pour permettre la communication série sur d'autres broches numériques de l'Arduino de manière logicielle. Il est possible de gérer plusieurs ports séries logiciel avec des vitesses allant jusqu'à 155200 *bps* cependant un seul peut recevoir des données à la fois.

Lorsque l'on ouvre le moniteur série (loupe en haut à droite de logiciel IDE Arduino), on voit apparaître des trames de type NMEA.

- ✓ Toutes les trames commencent par le signe \$,
- ✓ Toutes les trames finissent par le symbole \* + checksum sur 2 chiffres en hexadécimal. Le checksum permet de vérifier l'intégrité des données transmises,

Les trames qui nous intéressent pour la localisation sont les trames qui commencent par \$GPGGA. (GP =GPS et GGA=localisation).

Le module communique en 9600 *bauds* et envoie les informations de localisation toutes les 1s.

Le tableau IV.2 ci-dessous présente les commandes utilisés et leurs rôles dans la programmation d'une carte Arduino avec un module GPS. Dans la sous section suivante, on

donne un exemple d'un programme en utilisant un Arduino et le module GPS (GY-NEO6MV3).

**Tableau IV. 2.** Commandes et les rôles d'un programme GPS.

Commande	Rôle
#include<TinyGPS++.h	Appel de la bibliothèque
TinyGPSPlusgps	Création de l'instance de l'objet (GPS)
Gps.encode(data)	Transmet les caractères reçus à la bibliothèque pour analyse
Gps.location.isUpdated()	Localisation mise à jour
Gps.location.lat()	Renvoie la latitude sous la forme d'un nombre de type double
Gps.location.lng()	Renvoie la longitude sous la forme d'un nombre de type double
Gps.altitude.meters()	Renvoie l'altitude en mètres sous la forme d'un nombre de type double
Gps.speed.kmph()	Renvoie la vitesse en <i>Km/h</i> sous la forme d'un nombre de type double
Gps.satellites.value()	Renvoie le nombre de satellites utilisés sous la forme d'un nombre de type unsigned long

#### IV.3.3.4. Exemple de programme

```
#include "TinyGPS++.h"
#include "SoftwareSerial.h"
SoftwareSerial serial_connection(11, 10); //RX=pin 10, TX=pin 11
TinyGPSPlus gps; //This is the GPS object that will pretty much do all the gruntwork with the NMEA data
void setup() {
  Serial.begin(9600); //This opens up communications to the Serial monitor in the Arduino IDE
  serial_connection.begin(9600); //This opens up communications to the GPS
  Serial.println("GPS Start"); //Just show to the monitor that the sketch has started
}
void loop() {
  while(serial_connection.available()) //While there are characters to come from the GPS
  {
    gps.encode(serial_connection.read()); //This feeds the serial NMEA data into the library one char at a time
    if(gps.location.isUpdated()) //This will pretty much be fired all the time anyway but will at least reduce it to only after a package of NMEA data comes in
    //Get the latest info from the gps object which it derived from the data sent by the GPS unit
    Serial.println("Satellite Count:");
    Serial.println(gps.satellites.value());
    Serial.println("Latitude:");
    Serial.println(gps.location.lat(), 6);
    Serial.println("Longitude:");
    Serial.println(gps.location.lng(), 6);
    Serial.println("");
  }
}
```

### IV.3.4. Modules GSM

Il existe plusieurs Shields Arduino GSM/GPRS de différentes bandes de fréquence permettant la communication avec le monde entier quel que soit le type de cette communication (SMS, voix, données et fax). Dans ce projet, nous avons choisi le GSM SIM800L. Réunissez 850/900/1800/1900MHz, il peut transmettre la voix, le SMS et les informations sur les données avec la consommation de puissance faible. Il permet d'échanger des messages avec la carte Arduino Mega grâce aux commandes AT.

#### IV.3.4.1. Rôle du module GSM dans le projet

Le module GSM envoie les coordonnées par SMS pour prévenir le centre de suivi routier (protection civile, la gendarmerie nationale) de différentes urgences et des catastrophes sur la route en cas de risque de circulation ou d'accident, afin qu'il puisse suivre le véhicule, pour l'objectif d'une intervention rapide et un contrôle de la situation, surtout si le conducteur est seul sur la route et qu'il n'y a personne pour l'aider, car ce système est efficace dans de tels cas.

#### IV.3.4.2. Branchement du module GSM

Le module SIM800L se met directement sur la carte Arduino (voir les figures IV.16 et IV.17). La carte Arduino Mega et le SIM800L sont alimentés par une tension de 5V. Le module GSM communique avec la carte Arduino par des commandes AT qui permettent l'échange des SMS entre la carte et le GSM (voir la commande figure IV.18).

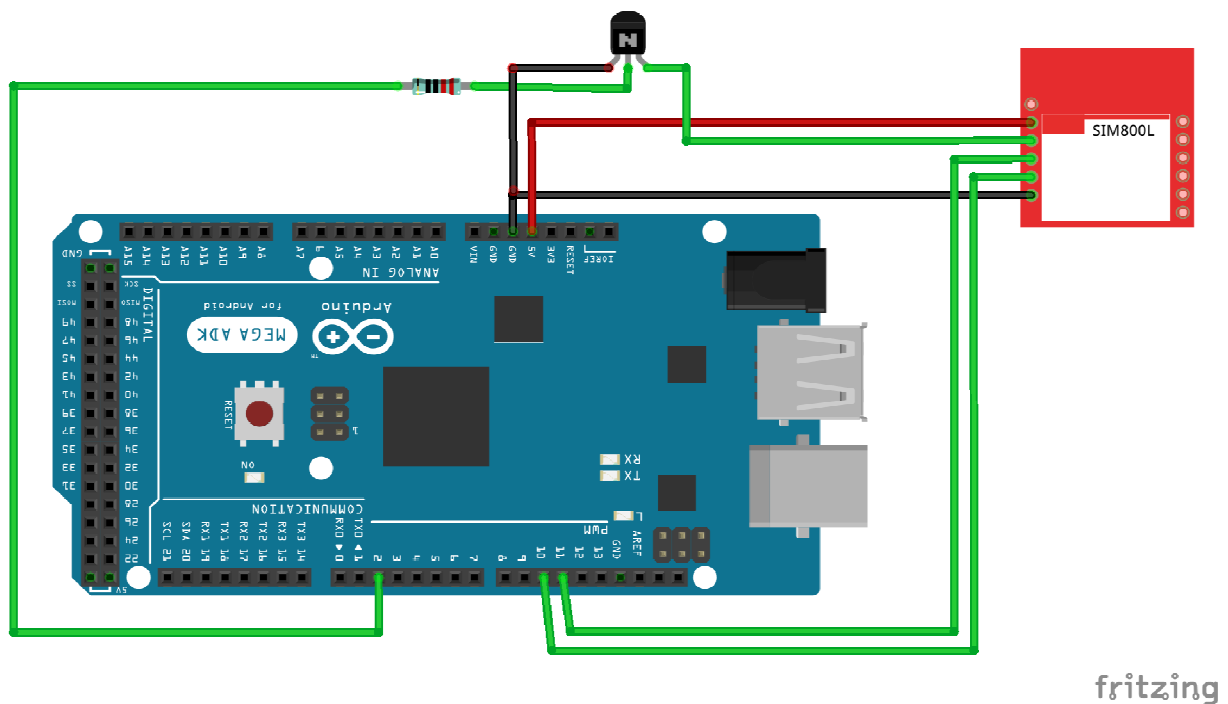


Figure IV. 16. (a) Branchement du module GSM avec une carte Arduino Mega.

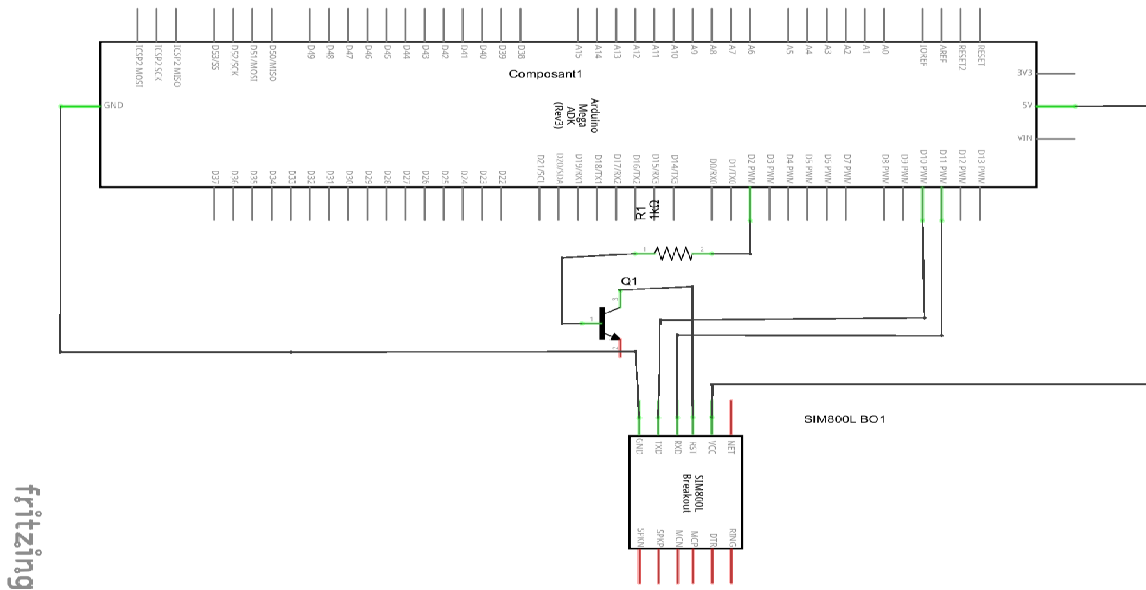


Figure IV. 17. Circuit électrique du module GSM avec la carte Arduino Mega.

La figure suivante représente les éléments de base de notre système de communication GSM :

- **Smart Phone (Utilisateur):** Permet de recevoir les messages SMS par le module GSM.
- **Module GSM:** Permet de recevoir ou d'envoyer des SMS et communiquer avec la carte Arduino Mega par une communication série (UART), les commandes utilisées dans cette connexions ce sont des AT commands.
- **Arduino Mega:** Permet de communiquer avec le module GSM, lire les messages et envoyer des commandes et exécuter les différents processus comme les systèmes d'alertes.
- **Alimentation :** Le module GSM et la carte Arduino Mega sont alimentés par une tension continue de 5V.

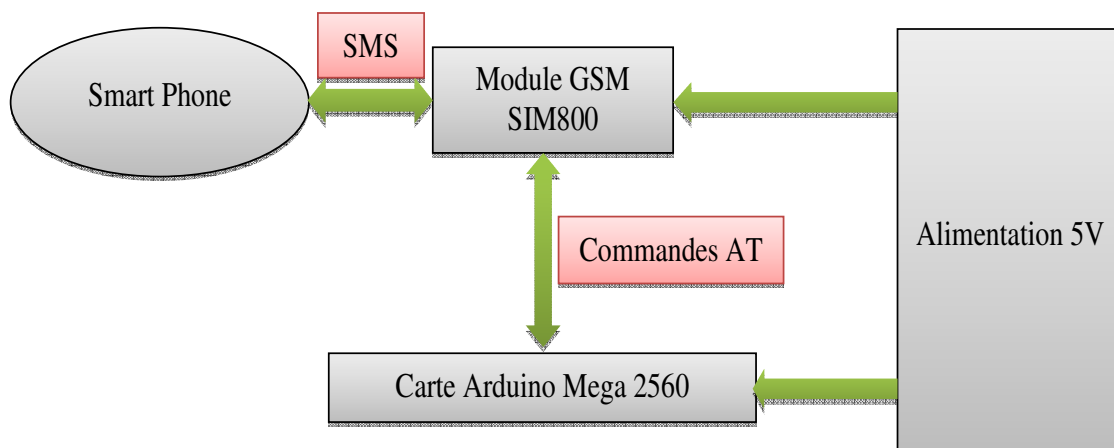
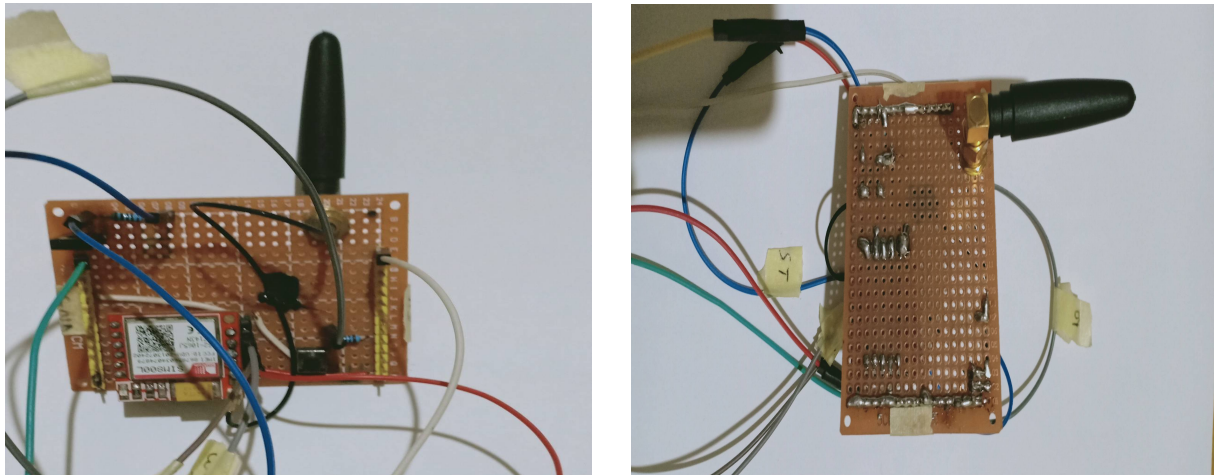


Figure IV. 18. Schéma récapitulatif du système GSM.

La figure suivante représente le schéma pratique du module SIM800L.



**Figure IV. 19.** Schéma pratique du Module SIM800L.

#### IV.3.4.3. Commandes AT

Les commandes AT sont des instructions utilisées pour contrôler le module GSM. AT est l'abréviation d'Attention. Chaque ligne de commande commence par "AT" ou "at". C'est pourquoi les commandes du GSM sont appelées commandes AT. Le Shield GSM et les téléphones mobiles prennent en charge les commandes AT liées à l'envoi des SMS comme représenté dans le tableau suivant.

**Tableau IV. 3.** Commandes AT dédiées au service SMS.

Commandes	Rôle
AT+CMGS	Envoyer le SMS
AT+CMGF	Sélection le format du SMS (PDU ou texte)
AT+CMGW	Ecrire un SMS dans la mémoire
AT+CMGD	Supprimer le SMS
AT+CMGC	Envoyer une commande
AT+CNMI	Indication concernant un nouveau SMS

#### IV.3.4.4. Code source

La bibliothèque GSM SoftwareSerial.h est incluse dans Arduino IDE. Avec le Shield Arduino GSM, cette bibliothèque permet à une carte Arduino de faire la plupart des opérations que nous pouvons effectuer avec un téléphone GSM : passer et recevoir des appels vocaux, envoyer et recevoir des SMS et vous connecter à Internet via un réseau GPRS. L'utilisation de cette librairie se fait comme suit `#include<SoftwareSerial.h>` au début du programme. Un exemple est présenté dans la sous section suivante.

### IV.3.4.5. Exemple de programme

```
#include<SoftwareSerial.h>
SoftwareSerialmySerial(10, 11);
voidsetup()
{
mySerial.begin(9600); // Setting the baud rate of GSM Module Serial.begin(9600); // Setting the
baud rate of Serial Monitor (Arduino)
delay(100);
}
voidloop()
{
if (Serial.available(>0)
switch(Serial.read())
{
case 's':
SendMessage();
break;
}
if (mySerial.available(>0)
Serial.write(mySerial.read());
}
voidSendMessage()
{
mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
delay(1000); // Delay of 1000 milli seconds or 1 second
```

### IV.3.5. Affichage à cristaux liquides (LCD)

Pour afficher les paramètres de notre prototype, un écran LCD parallèle est utilisé. C'est un écran de 16×2 caractères. Il comporte 16 broches pouvant afficher un total de 32 caractères (16 colonnes et 2 lignes).

#### IV.3.5.1. Le rôle de l'affichage à cristaux liquides (LCD) dans le projet

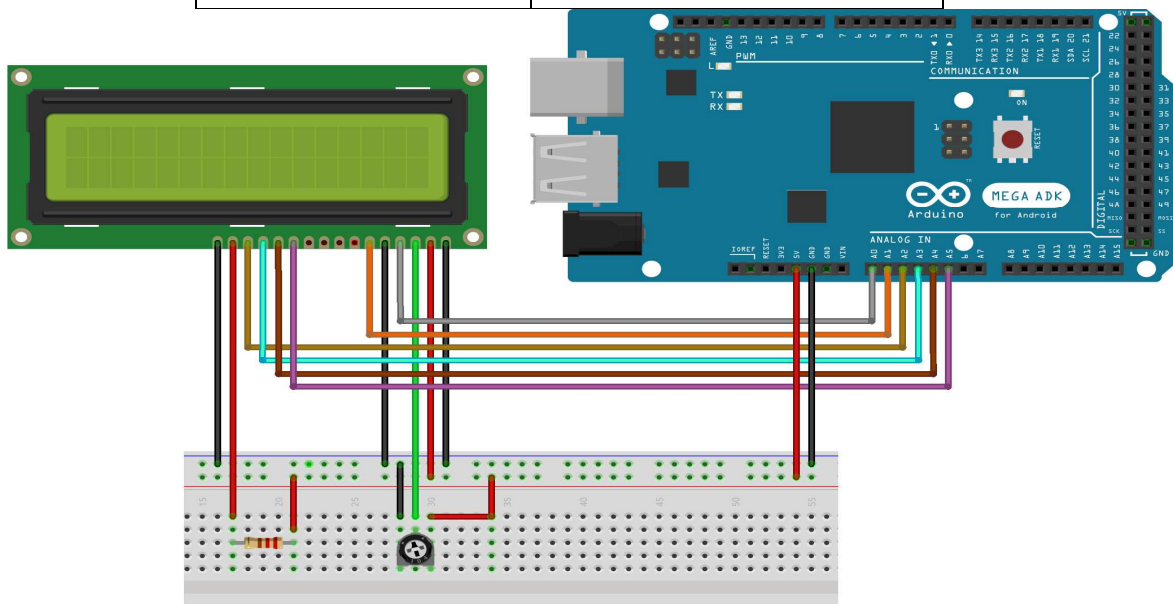
Dans ce projet nous avons utilisé deux afficheur LCD, une dans le véhicule pour afficher les coordonnées GPS et les panneaux de signalisations, l'autre se trouve dans l'infrastructure pour afficher la position et la situation des véhicules dans la zone de couverture de l'infrastructure.

#### IV.3.5.2. Branchement de l'écran LCD

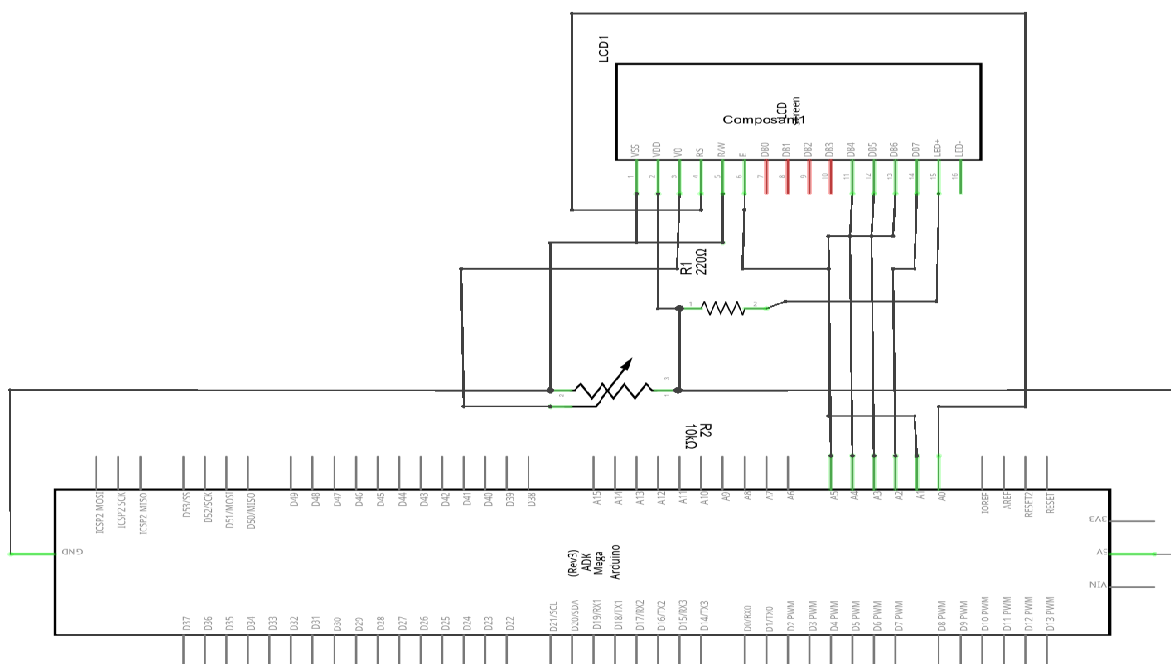
Nous pouvons connecter les broches de communication de l'écran LCD à toutes les broches d'entrées/sorties de l'Arduino. Dans ce projet, ils sont connectés comme indiqué dans le tableau IV.4. Le branchement et le circuit électrique de l'écran LCD avec l'Arduino Mega sont représentés dans les figures IV.20 et IV.21.

**Tableau IV. 4.** Connexions des broches LCD de communication.

Broches LCD	Broches d'Arduino
RS	A0
EN	A1
D4	A2
D5	A3
D6	A4
D7	A5



**Figure IV. 20.** Branchement d'écran LCD avec Arduino Mega.



**Figure IV. 21.** Circuit électrique d'écran LCD avec Arduino Mega.

fritzing

### IV.3.5.3. Programmation de l'écran LCD

L'IDE Arduino inclut la bibliothèque LiquidCrystal, un ensemble de fonctions qui facilite l'interface avec l'écran LCD parallèle. La bibliothèque LiquidCrystal possède une quantité impressionnante de fonctionnalités, notamment le clignotement du curseur, le défilement automatique du texte, la création de caractères personnalisés et la modification du sens d'impression du texte. Pour commencer il faut d'abord, inclure la bibliothèque LiquidCrystal par `#include<LiquidCrystal.h>`. Ensuite il faut, initialisez un objet LCD comme suit: LiquidCrystallcd (A0, A1, A2, A3, A4, A5). Les arguments pour l'initialisation de l'écran LCD représentent les broches Arduino connectées à RS, EN, D4, D5, D6 et D7, dans cet ordre. Dans la configuration, nous appelons la fonction `begin()` de la bibliothèque pour configurer l'écran LCD avec la taille du caractère. L'écran utilisé est un écran (16x2). 16 représente le nombre de colonnes et 2 le nombre de lignes. La fonction qui définit le nombre de lignes et de colonnes est : `lcd.begin(16, 2)`. Les positions à l'écran sont indexées en commençant par (0,0) en haut à gauche. Le premier argument de `setCursor()` spécifie quel numéro de colonne et le second spécifie quel numéro de ligne. Par défaut, l'emplacement de départ est (0,0). Ainsi, si nous appelons la fonction `print()` sans d'abord modifier l'emplacement du curseur, le texte commence dans le coin supérieur gauche. Dans la sous section suivante en représente un exemple de la programmation IDE Arduino avec un écran LCD.

### IV.3.5.4. Exemple de programme

```
#include<LiquidCrystal.h>
LiquidCrystallcd(A0, A1, A2, A3, A4, A5); // (rs en d4 d5 d6 d7)
void setup () {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Mebarki Hana");
  lcd.setCursor(0, 1);
  lcd.print("Ghedir Ouided");
  delay(10000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" VNET Systems:");
  lcd.setCursor(0, 1);
  lcd.print("GPS, GSM, RF");
  delay(10000);
}
```

### IV.3.6. Raspberry Pi 3

#### IV.3.6.1. Connexion à distance

Pour pouvoir se connecter à distance à notre Raspberry, il faut le connecter à un réseau local et faire appel au protocole SSH. Le serveur SSH est activé par défaut sur le Raspberry *Pi*.

Sous Linux, le client étant intégré dans la plupart des distributions. Par contre, sous Windows, nous devons installer «Putty» qui est un client SSH. Ensuite, il suffit d'entrer l'adresse IP du Raspberry dans Putty pour pouvoir se connecter.

##### IV.3.6.1.1. Réserveation d'adresse IP

Un serveur DHCP distribue une adresse IP aux clients selon la disponibilité des adresses de la plage définie. Cependant, il est possible de faire en sorte qu'un hôte ait toujours la même adresse IP attribuée, c'est ce que l'on appelle de la réserveation d'adresse. On doit donc indiquer dans le serveur DHCP (Dynamic Host Configuration Protocol) que telle adresse IP est réservée à telle adresse MAC.

Ou bien on alloue une adresse IP statique à notre Raspberry pour une liaison filaire par la configuration suivante :

-On ouvre le fichier de configuration des cartes réseaux :

```
sudo nano /etc/network/interfaces
```

-Puis indiquons à notre carte réseau que nous souhaitons passer en static :

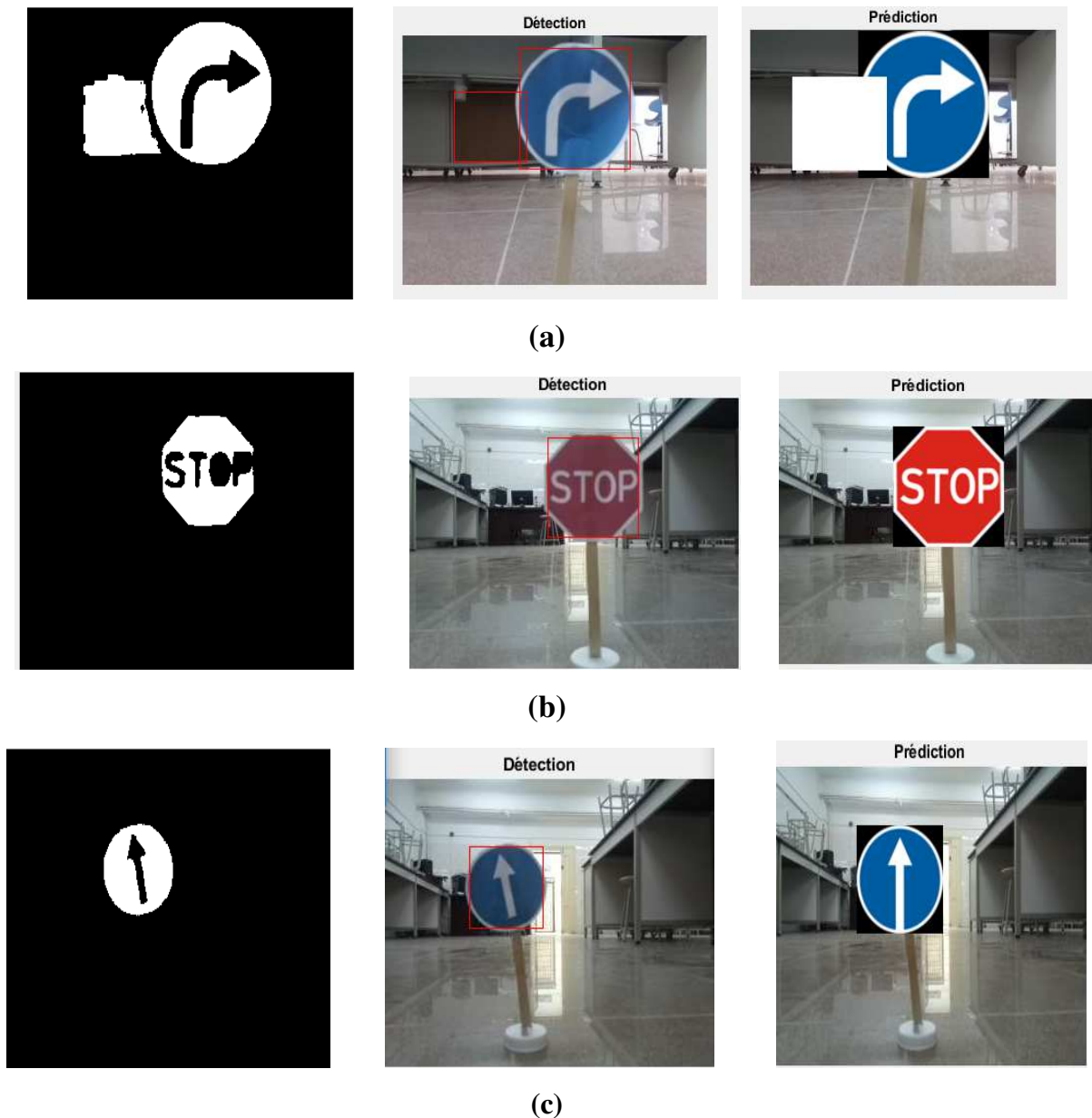
```
auto eth0
iface eth0 inet static
address 192.168.1.111
netmask 255.255.255.0
gateway 192.168.1.1
```

##### IV.3.6.1.2. Système de détection et reconnaissance automatique

Dans cette partie nous sommes intéressés par la méthode de détection et reconnaissance TSR (Traffic Signs Recognition) en temps réel avec l'utilisation de la carte Raspberry *Pi 3* connecté avec une camera.

Dans le chapitre précédant, nous avons expliqué en détail les deux phases essentielles pour la détection et la reconnaissance automatique des panneaux de signalisations (TSR); plus précisément : la phase de calibrage et la phase en temps réel. D'après les résultats obtenus, nous avons trouvé que l'algorithme basé sur la segmentation MSER, la détection ECC, avec l'intermédiaire d'un descripteur HOG (bloc de dimension [64×64]) et le classifieur machine à vecteurs de support (SVM) semble être le plus efficace pour l'étape de reconnaissance TSR en temps réel.

Afin de démontrer la méthode décrit précédemment dans le chapitre III, nous avons fait des tests qui ont été prise avec la RaspCam. La Figure IV.22 montre la détection de quelques panneaux de signalisation qui nous avons utilisé.

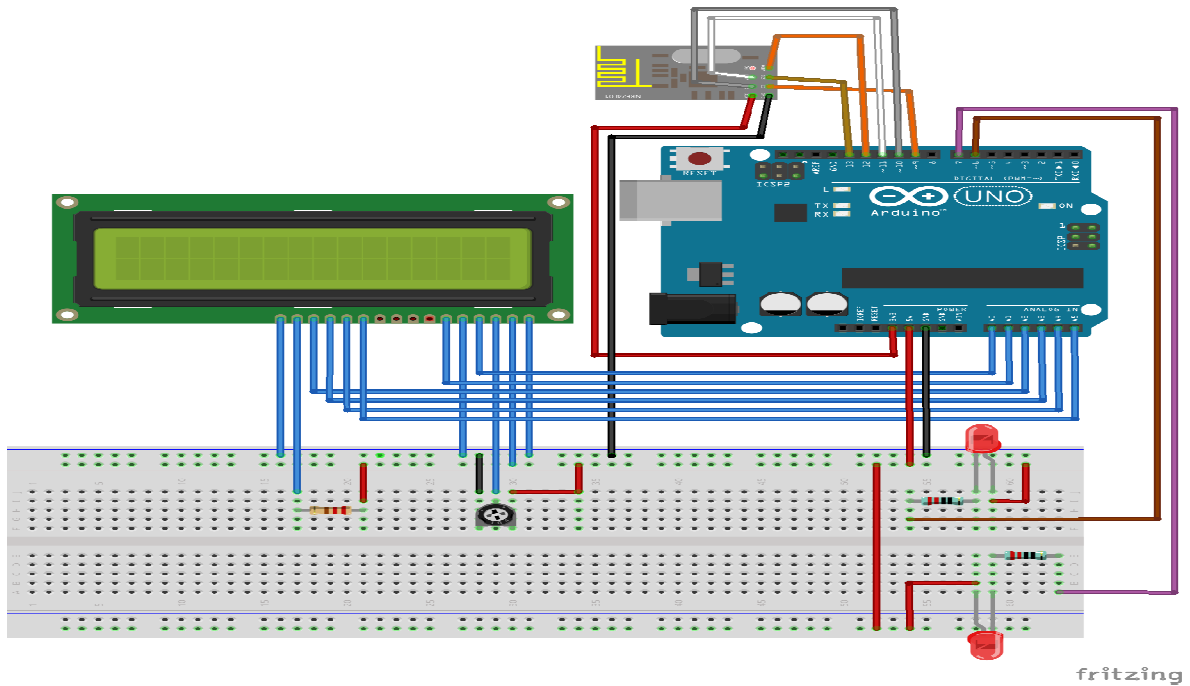


**Figure IV. 22.** Détection et reconnaissance automatique des panneaux de signalisations, (a) Obligation de tourner à droite, (b) Arrêt obligatoire, (c) Obligation d'aller tout droit.

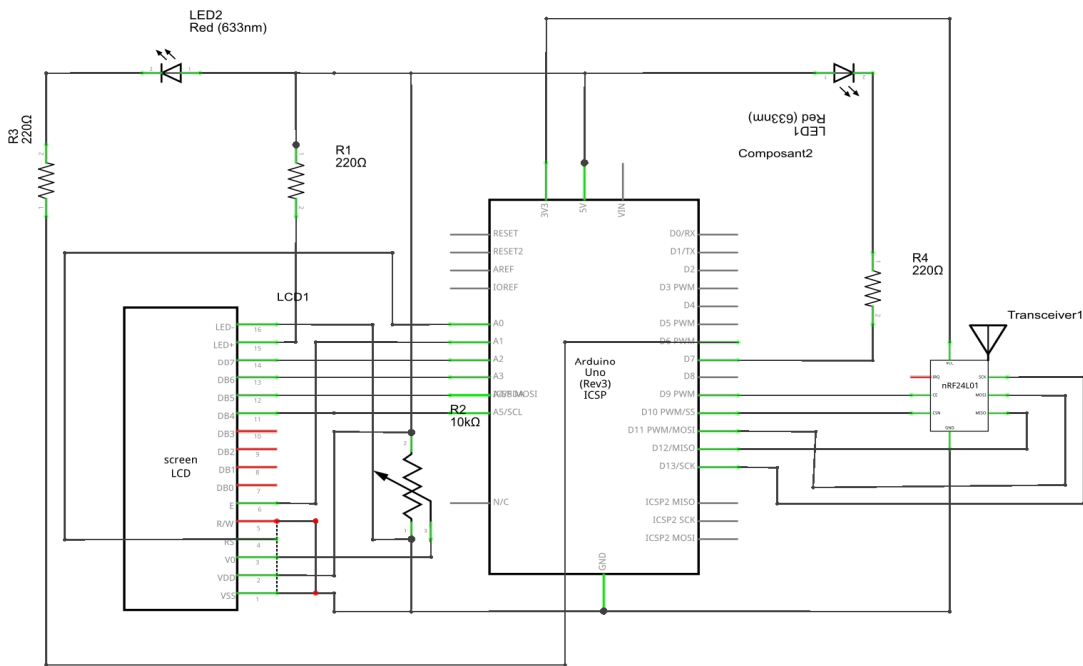
### IV.3.7. Schéma d'assemblage

#### IV.3.7.1. Prototype infrastructure

Pour visualisés les différentes informations émis par les véhicules, on utilise les composants: Arduino Uno, l'afficheur LCD 16x2, nRF24, des résistances et des leds d'indication comme montré dans les figures IV.23 et IV.24.



**Figure IV. 23.** Assemblage de radiofréquence nRF24L01 avec la carte Arduino Uno et l’afficheur LCD 16x2.



**Figure IV. 24.** Circuit électrique de module nRF24L01 connecté avec la carte Arduino Uno et l’afficheur LCD 16x2.

#### IV.3.7.2. Prototype communication véhicule-véhicule et véhicule-infrastructure

Dans cette section, nous faisons des assemblages des montages précédents que nous avons utilisés pour réaliser le prototype de notre projet de communication véhicule-véhicule et véhicule-infrastructure. Afin d’être le plus fidele possible avec le système réel, nous avons

utilisés deux robots mobiles pour l'objectif de tester l'échange d'informations entre les véhicules lui-même et véhicule-infrastructure en utilisant la communication radiofréquence.

Pour le premier véhicule (robot mobile 1) nous avons opté pour les composants suivants:

- ❖ Cartes Arduino Mega et Raspberry Pi 3 qui jouent le rôle d'unité de traitement pour notre prototype.
- ❖ Une caméra qui permet de prendre des photos ou des vidéos de l'environnement.
- ❖ Un écran LCD qui est le principal élément de visualisation de l'état du système (les panneaux de signalisations, la distance, les coordonnées GPS, ...).
- ❖ Ultrason (HC-SR04) pour la détection d'obstacle ou de mesuré la distance.
- ❖ Le pont-H L298N, son rôle sera de commander les quatre moteurs courant continu de notre prototype.
- ❖ Quatre moteurs à courant continu (5V) qui permettent de déplacer le prototype.
- ❖ Les différents shields de communication sans fil (nRF24L01, GSM, GPS).

Pour le deuxième véhicule (robot mobile 2) nous avons utilisé les composants suivants:

- ❖ Une carte Arduino Uno.
- ❖ Ultrason HC-SR04.
- ❖ La radiofréquence nRF24L01.
- ❖ Shield L298N et deux moteurs à courant continu qui permettent de déplacer notre robot.

Les figures IV.25 et IV.26 illustrent l'assemblage finale des deux robots mobiles utilisés dans notre prototype. On prend en considération les facteurs suivants :

- Pour le déplacement des robots mobiles, nous avons appliquée une simple commande sur les moteurs à courant continu par la carte Arduino (Uno ou Mega): avant, droite, gauche, arrêt.
- Pour tester le système d'alerte de notre prototype, un bouton poussoir déclenche ce dernier.
- L'échange des informations entre le Raspberry *Pi* et l'Arduino Mega est à travers une communication parallèle, par l'intermédiaire de deux résistances en série.
- En utilise le toolbox sous Simulink-Matlab de Raspberry *Pi* (simulink support package for raspberry Pi hardware) pour créer un système embarqué.

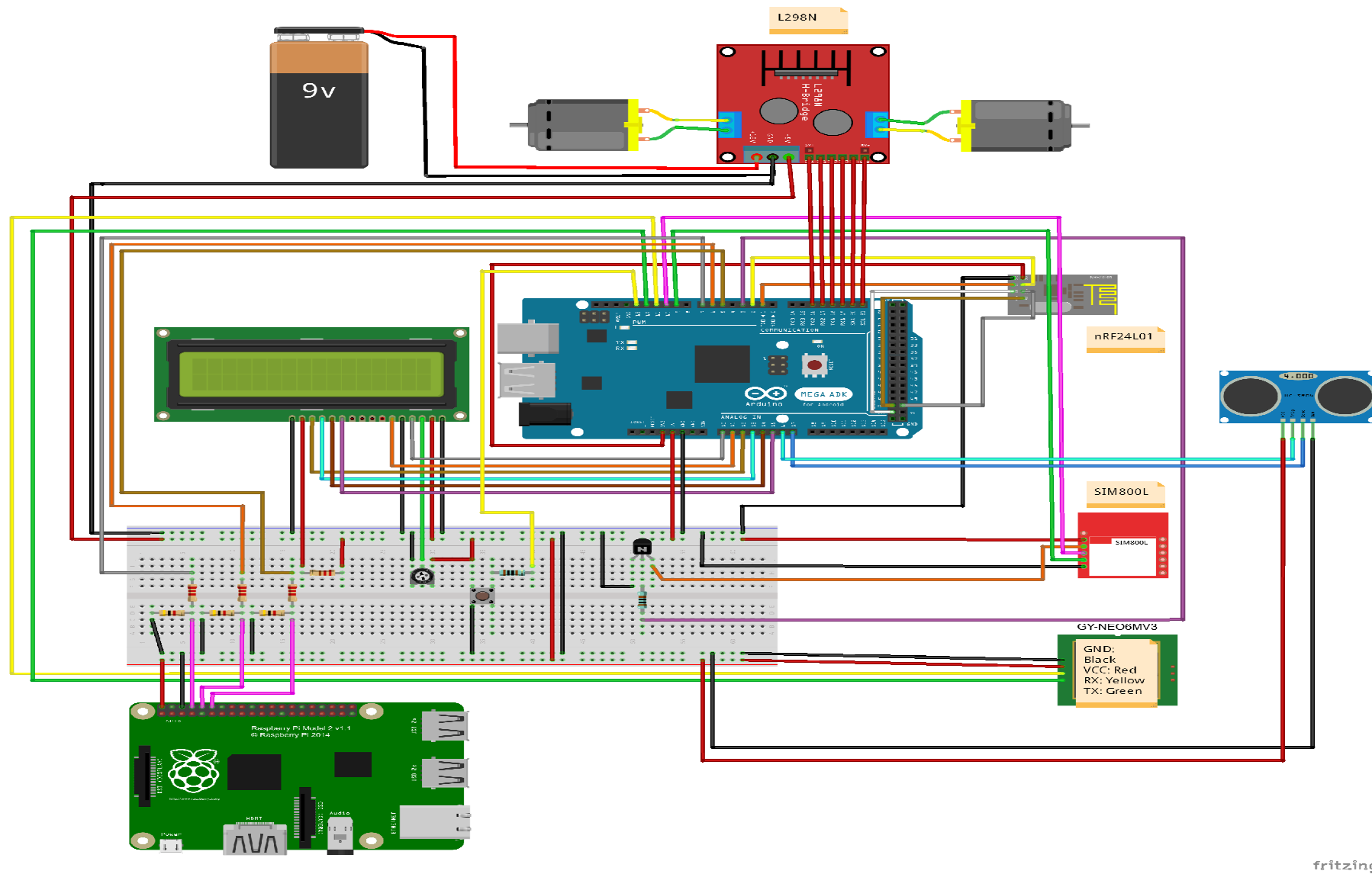


Figure IV. 25. Assemblage du robot mobile 1.

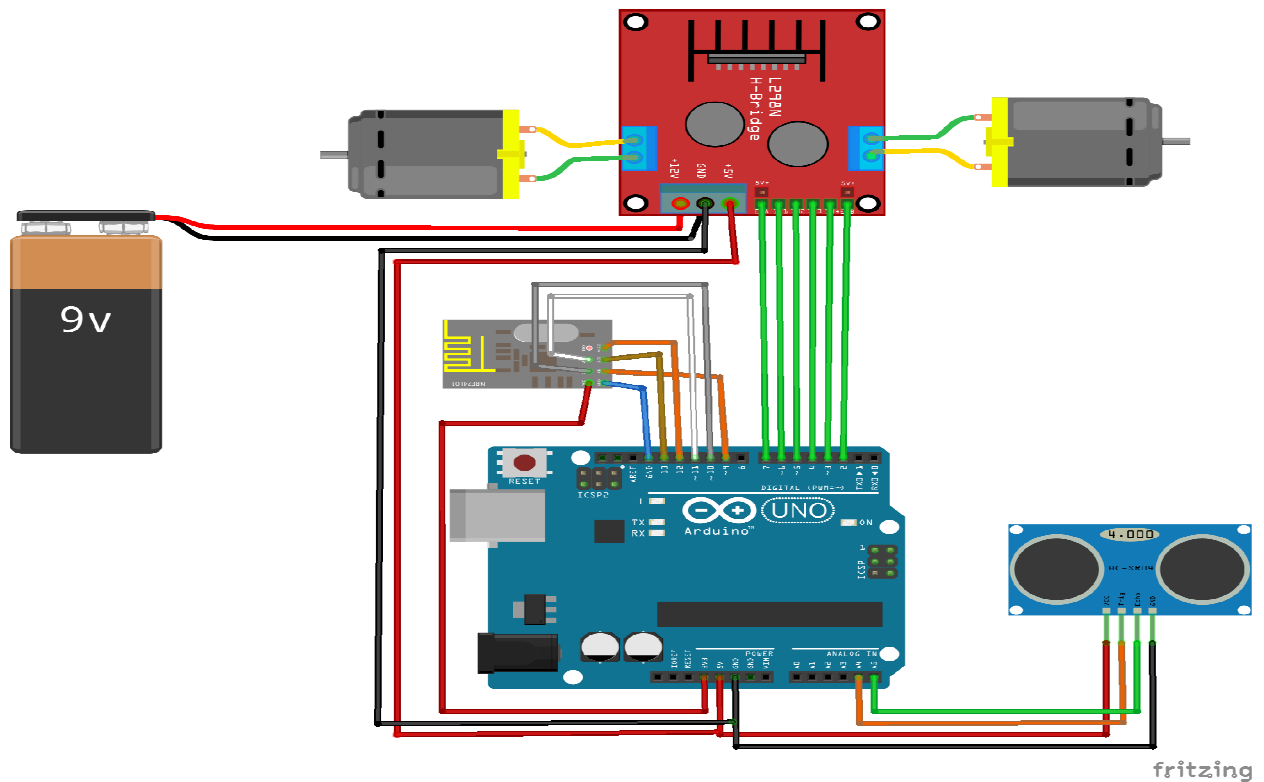


Figure IV. 26. Assemblage du robot mobile 2.

#### IV. 4. Réalisation finale et tests

pour tester le fonctionnement de notre prototype du système VNET, nous avons utilisé une maquette pour expliquer l'objectif de notre projet, dans cette dernière nous avons placé deux robots mobiles et infrastructure (voir les figures IV.27 et 28).

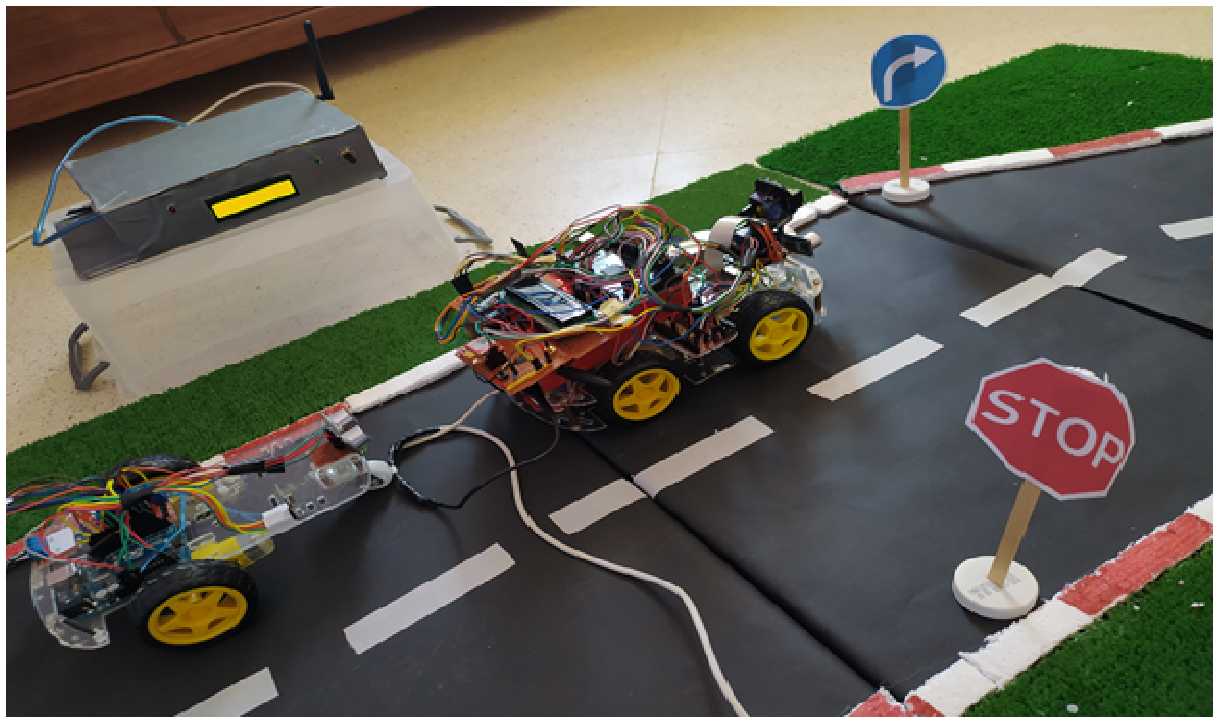
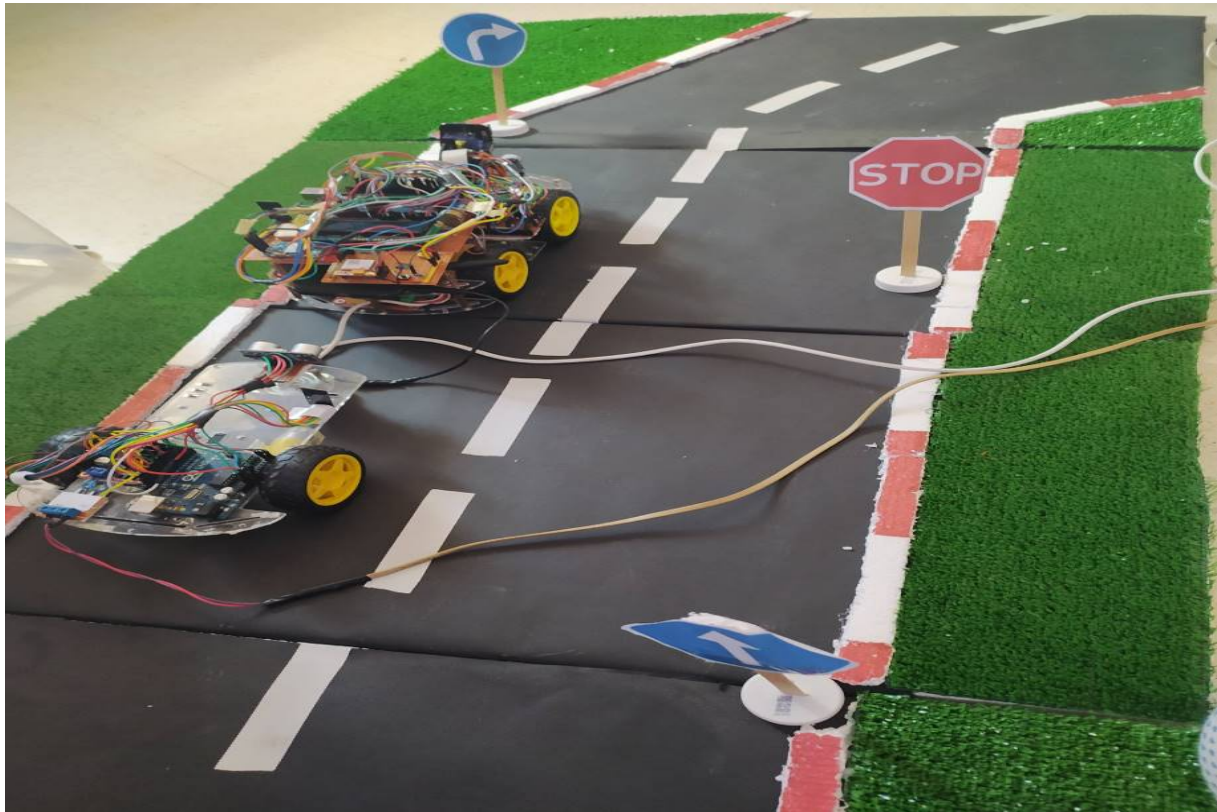
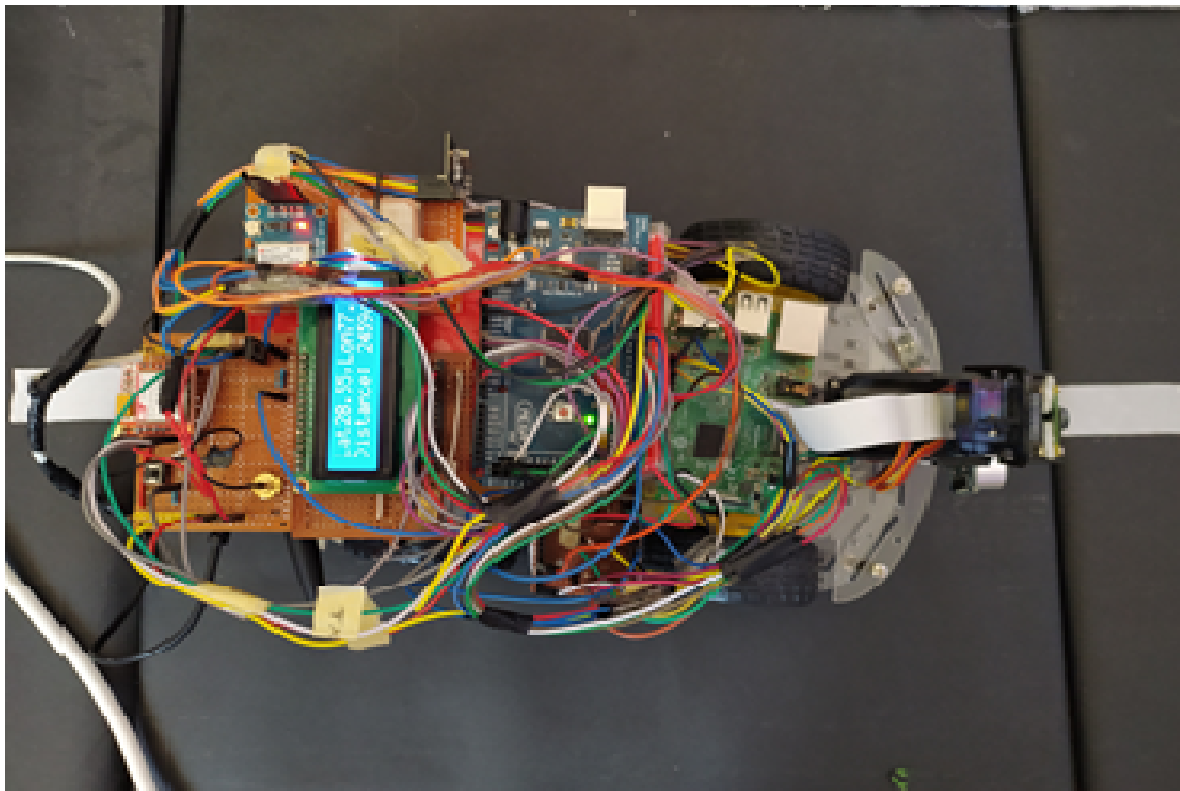


Figure IV. 27. Communication véhicules- infrastructure.

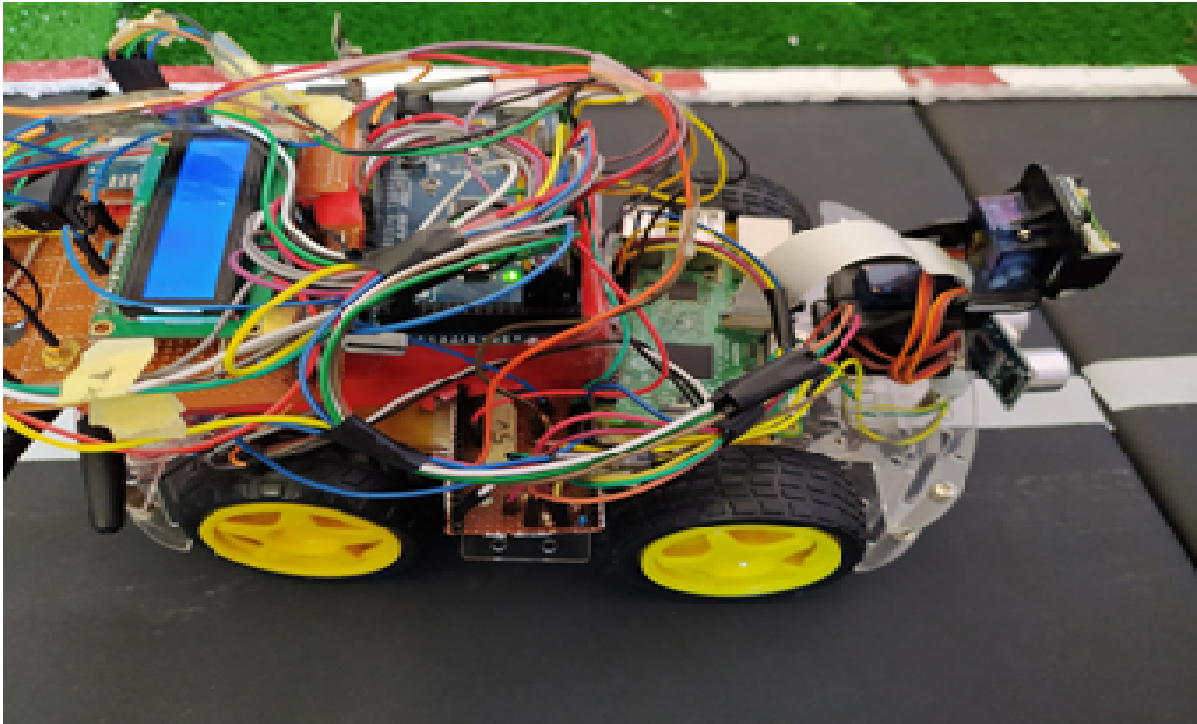


**Figure IV. 28.** Communication véhicule-véhicule.

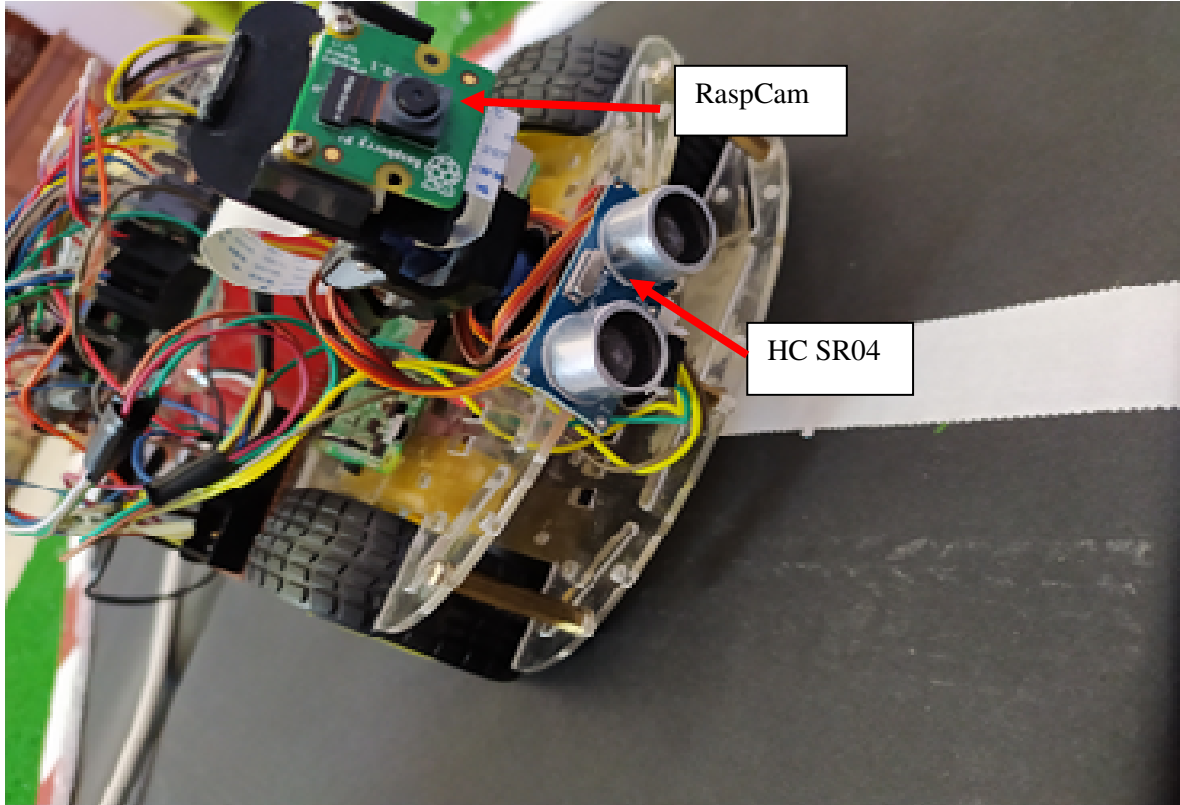
Le premier robot décompose de tous les éléments essentiels de notre système (Arduino Mega, afficheur, ultrason, nRF, GPS, GSM) comme représentée dans la figure IV.29.



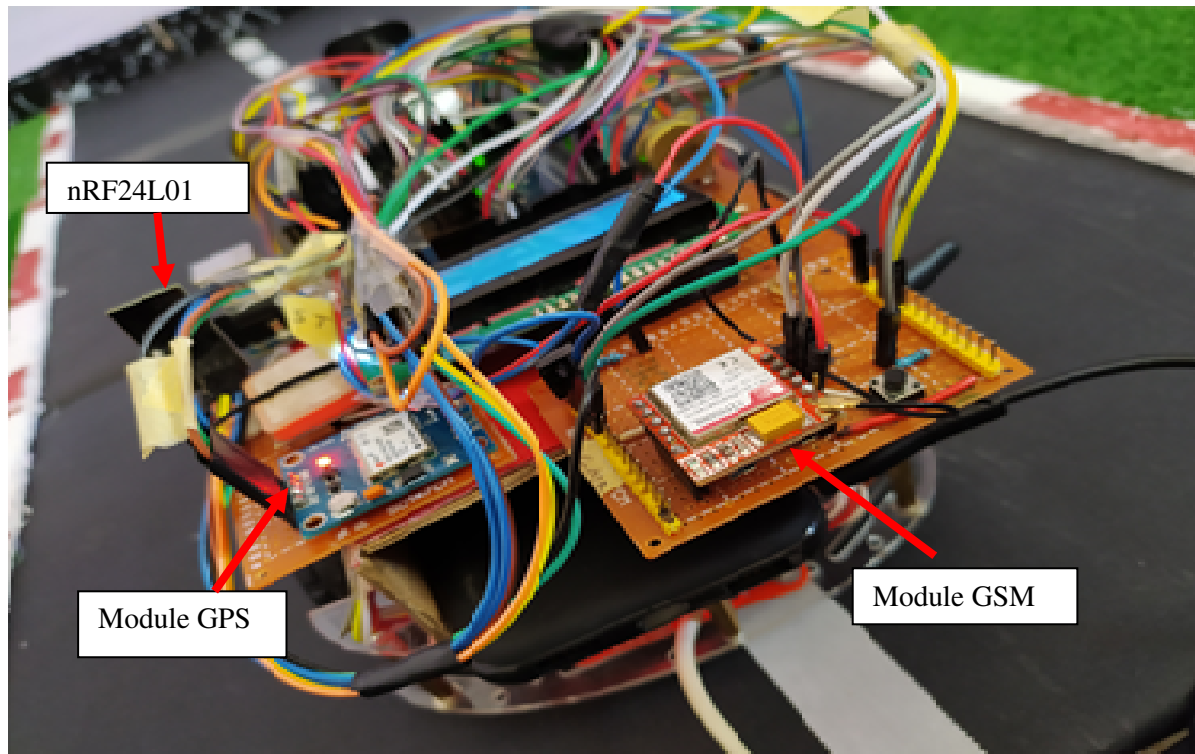
(a)



(b)



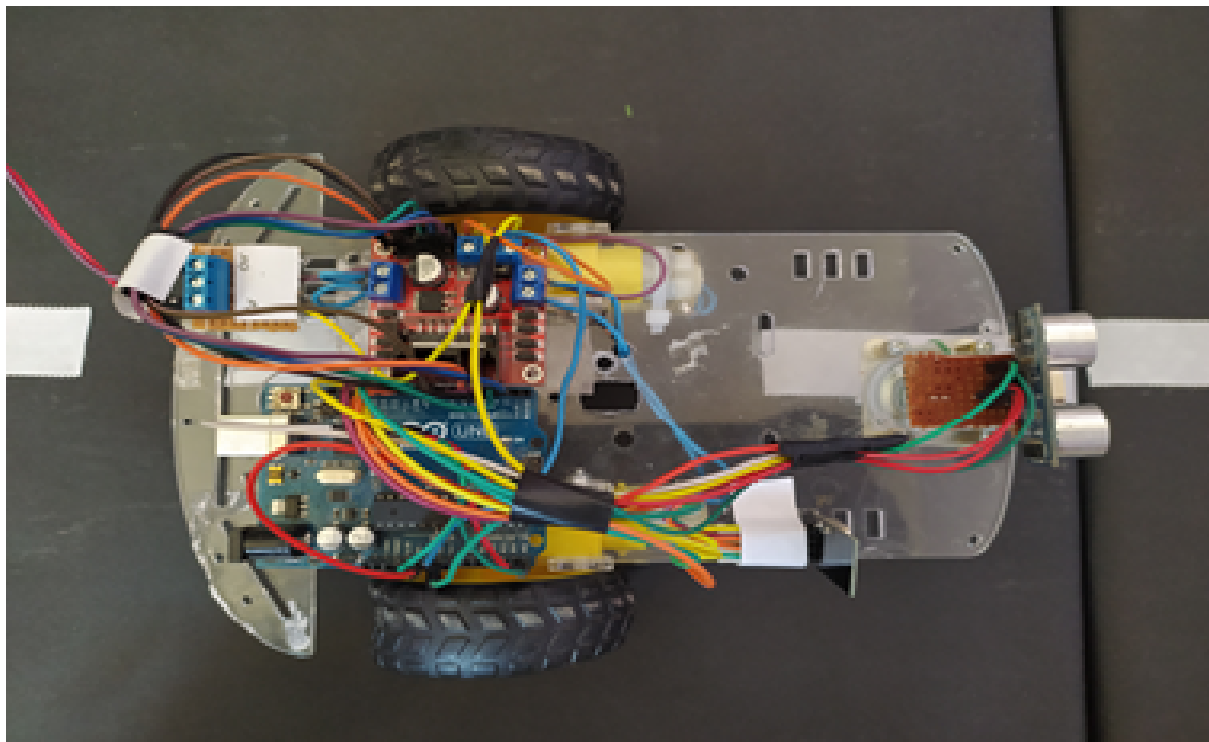
(c)



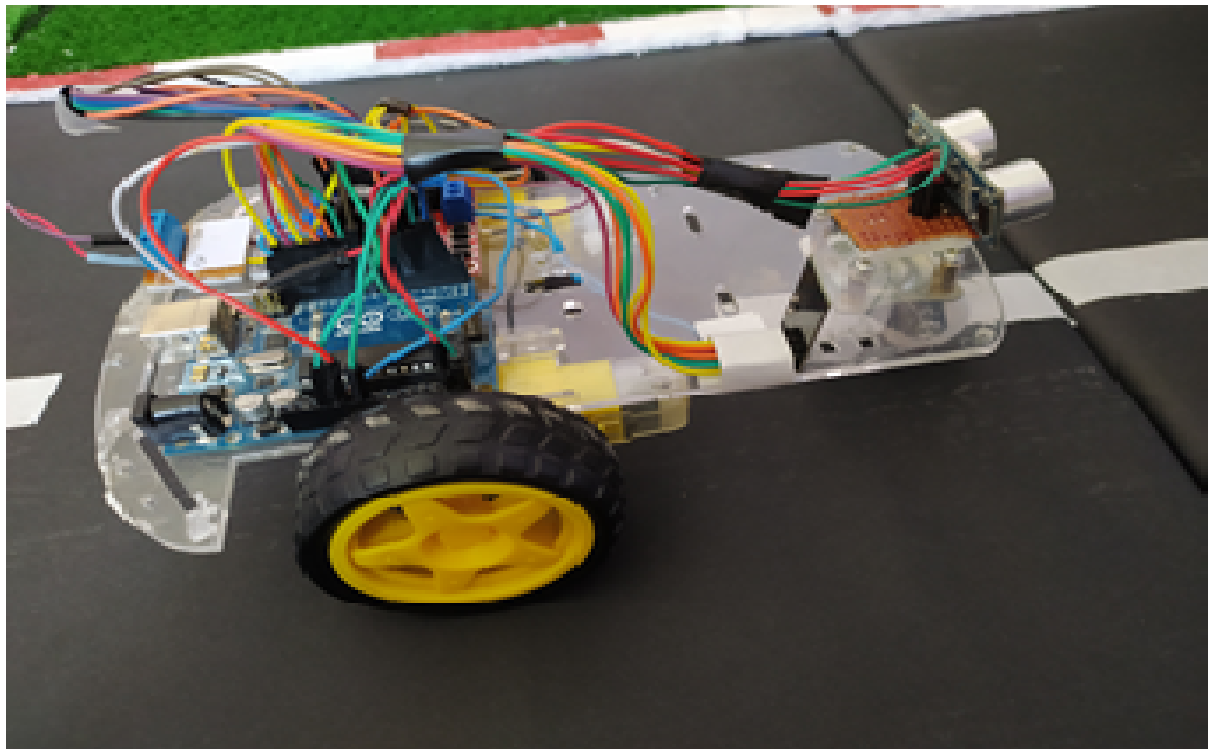
(d)

**Figure IV. 29.** Robot mobile 1 après l'assemblage. (a) Face de haut, (b) Face de l'avant droit, (c) Face en avant, (d) Face en arrière.

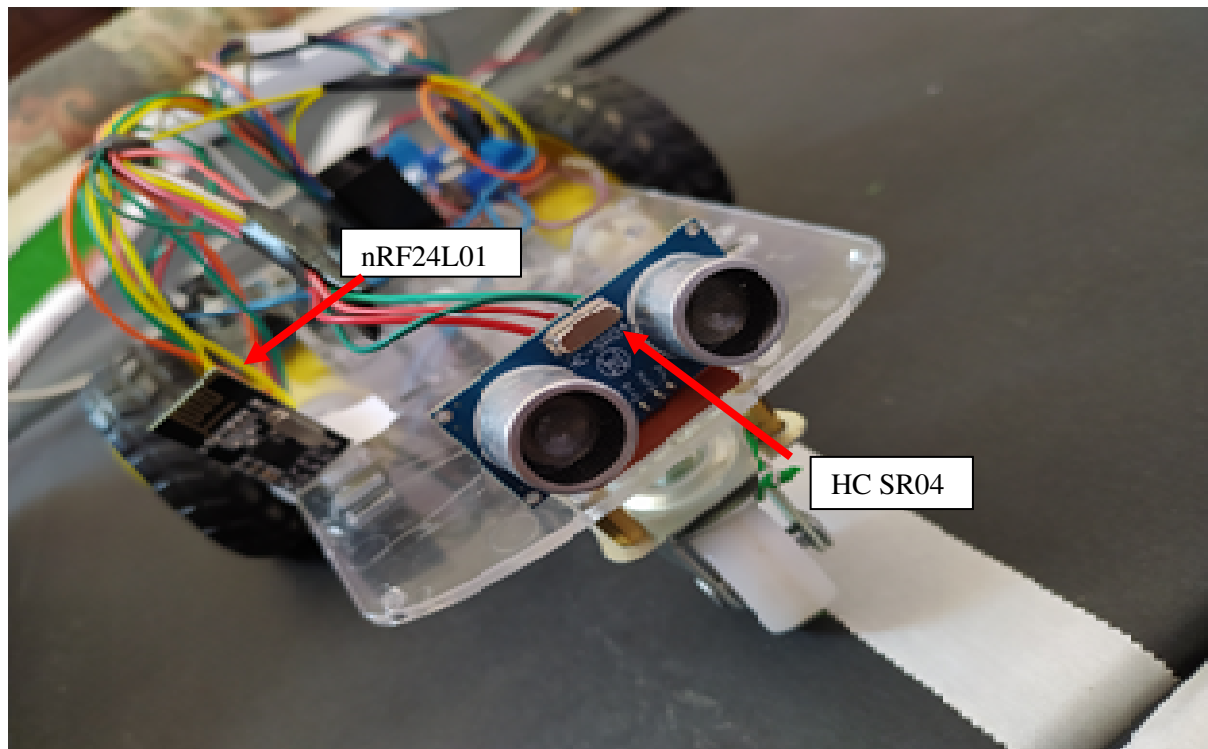
Le deuxième robot se décompose d'Arduino Uno, radiofréquence et ultrason. Ce dernier est utilisé pour tester la transmission et la réception entre les différents véhicules (Figure IV.30).



(a)



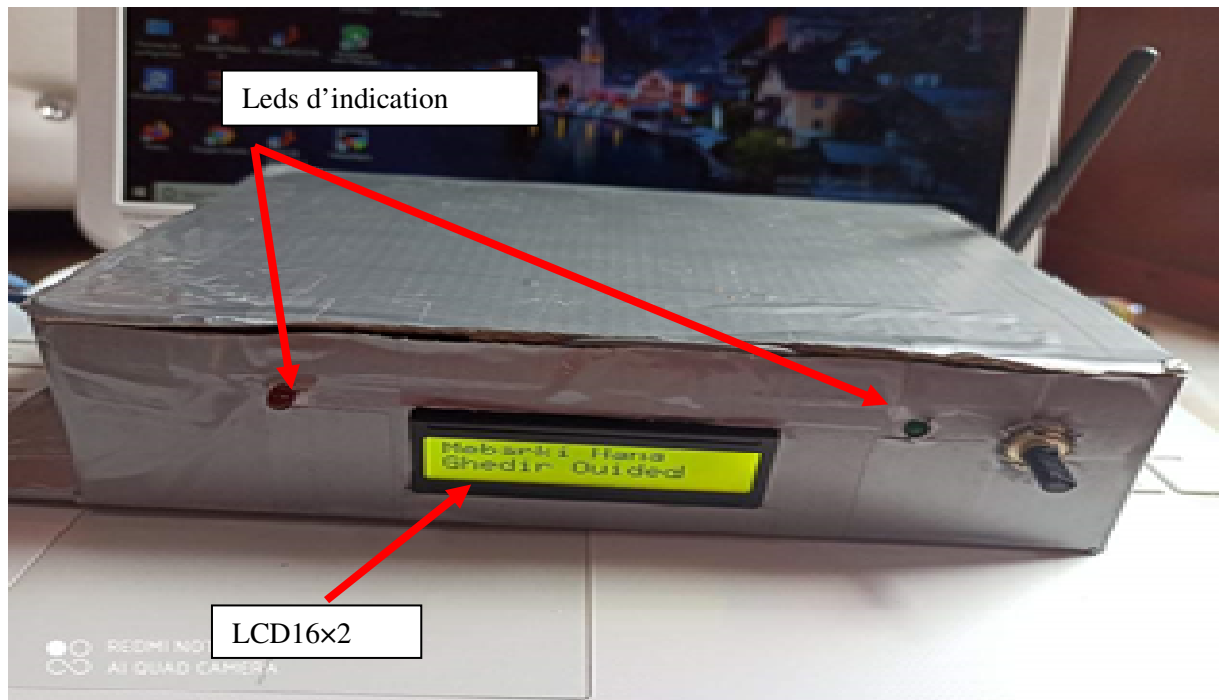
(b)



(c)

**Figure IV. 30.** Robot mobile 2 après l'assemblage. (a) Face de haut, (b) Face de l'avant droit, (c) Face en avant.

Pour l'infrastructure, on affiche la position du premier véhicule par les coordonnées GPS et ainsi l'induction des panneaux de signalisation avec l'utilisation de l'intelligence artificielle (Figure IV.31).



(a)



(b)

**Figure IV. 31.** Prototype de l'infrastructure, (a) indication de présence ou absence des véhicules dans la zone de couverture de radiofréquence: Leds et afficheur LCD, (b) module nRF24L01.

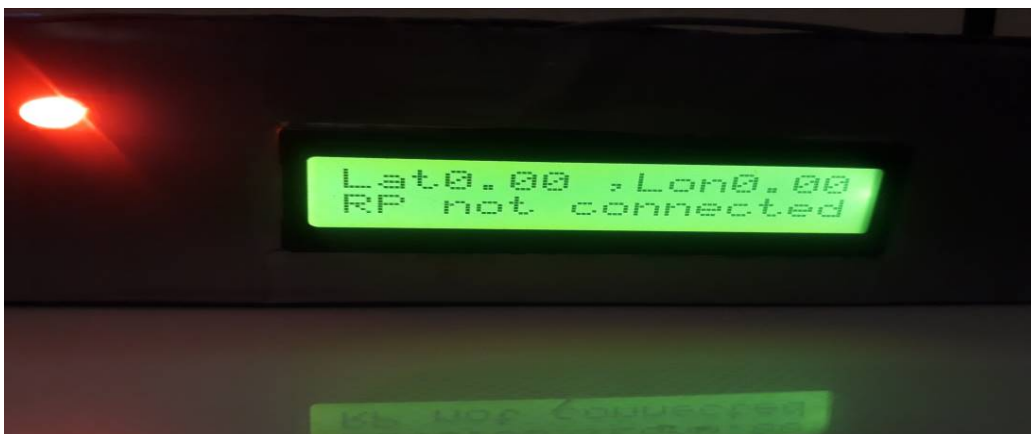
#### IV.4.1. Les tests d'affichage de l'infrastructure

- Dans le cas où il n'y a aucune voiture dans la zone de couverture de radiofréquence, LCD affiche :



**Figure IV. 32.** Test d'affichage 1.

- Dans le cas où le Raspberry *Pi* n'est pas connecté, LCD affiche :



**Figure IV. 33.** Test d'affichage 2.

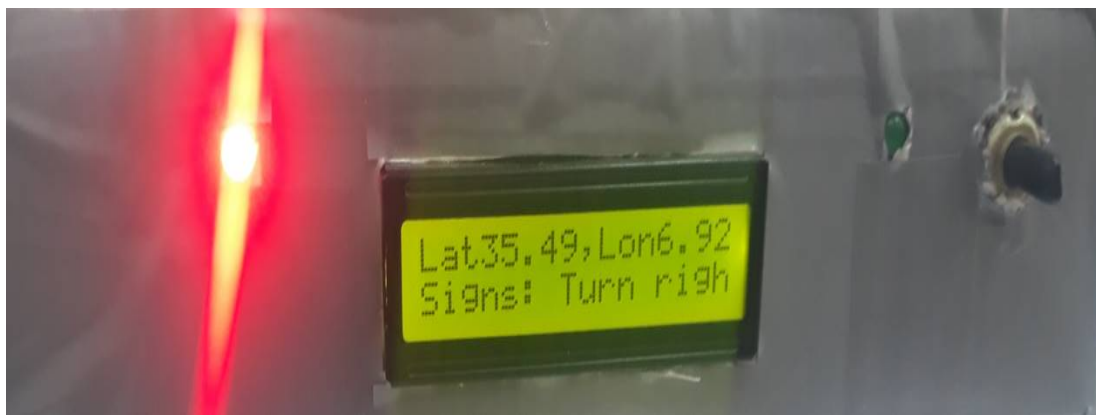
- Dans le cas où on trouve un panneau sur la route, LCD affiche les coordonnées GPS de la localisation et le nom du panneau :



(a)



(b)



(c)

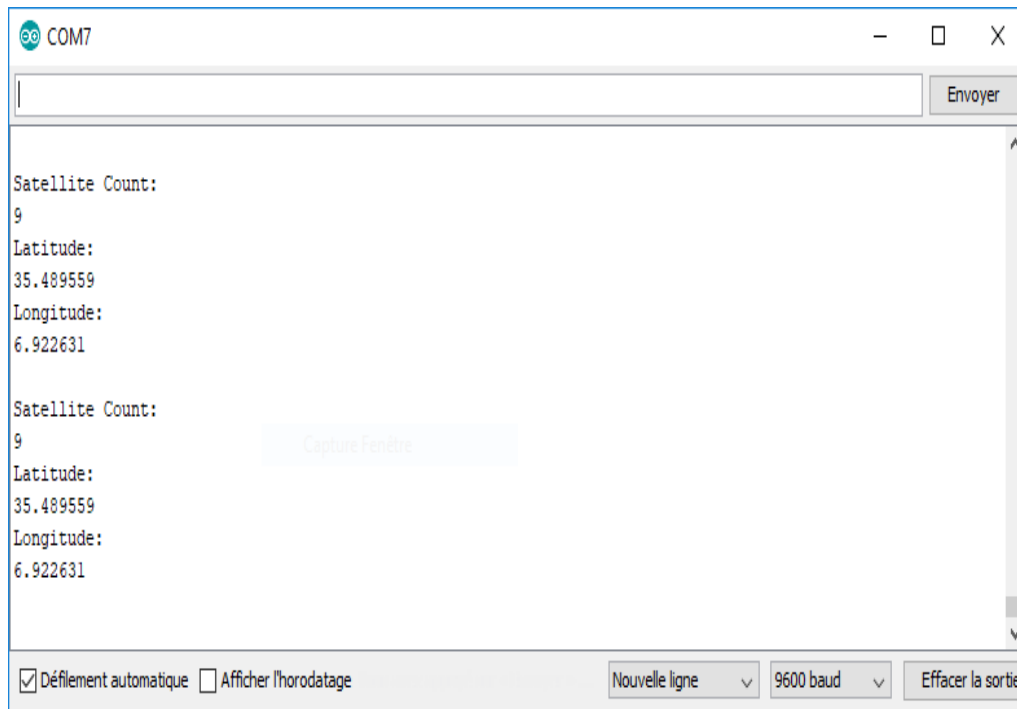
**Figure IV. 34.** Test d'affichage 3.

#### IV.4.2. Teste du système de message d'urgence

##### IV.4.2.1. Parité logiciel

###### ➤ Test de l'emplacement du véhicule sur le logiciel arduino IDE

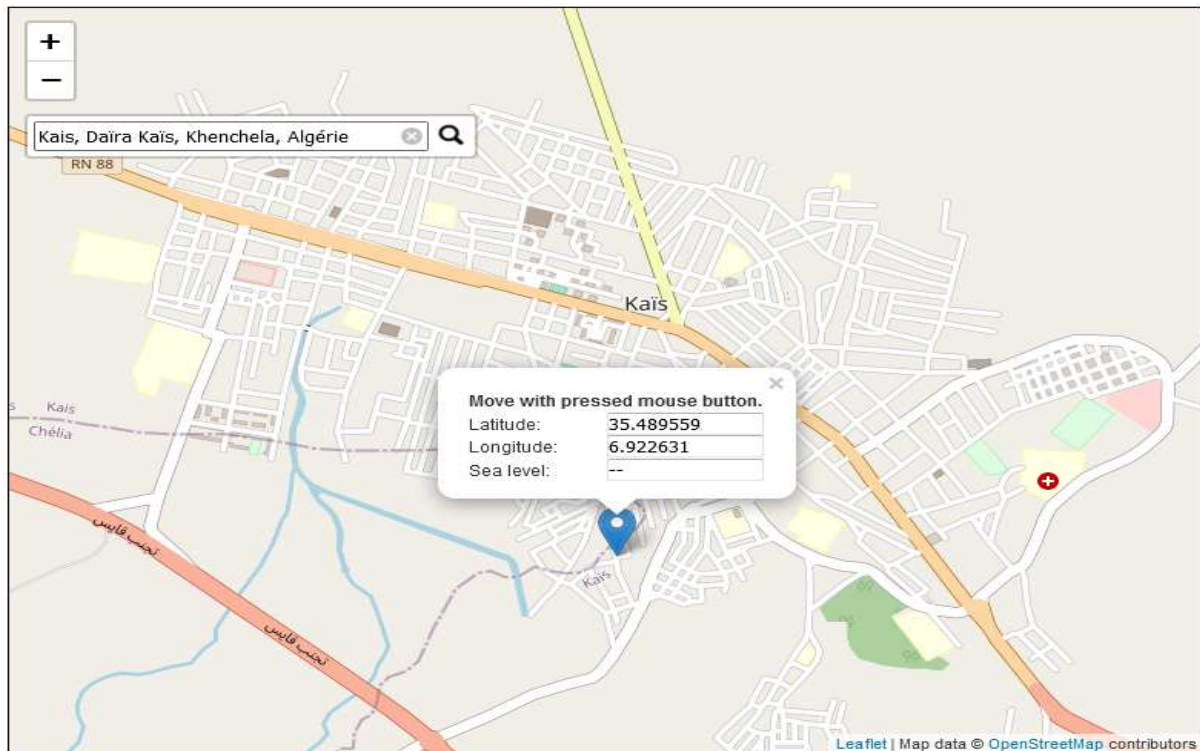
Le logiciel Arduino IDE est utilisé pour compiler le programme dans le microcontrôleur. Dans ce logiciel, le langage de programmation C a été utilisé pour le codage. Pour recevoir des données du satellite et envoyer des données dans le téléphone portable, le programme est écrit à l'aide du logiciel Arduino IDE.



**Figure IV. 35.** Montre les coordonnées géographiques du véhicule à l'aide du logiciel Arduino IDE.

### ➤ Test de l'emplacement du véhicule sur Google Map

Les informations de longitude et de latitude reçues ont été utilisées pour identifier l'emplacement du véhicule sur Google Map.



**Figure IV. 36.** Montre la position du véhicule à l'aide de Google Maps.

Figure IV.36 illustre l'emplacement du véhicule sur la carte Google Map. La bulle indique la position sous forme de latitude et de longitude du véhicule, l'endroit était «kais, daira kais, khenchela, algérie» dont la longitude 6.922631 et la latitude 35.489559.

#### IV.4.2.2. Parité matériel

En début, une alimentation CC +5V est fournie au GPS et GSM shield, Arduino méga avec microcontrôleur pour activer ces appareils. L'Arduino est utilisé pour contrôler le processus avec un récepteur GPS et un module GSM. Le récepteur GPS fonctionnant sur un débit de 9600 bauds est utilisé pour recevoir les données des satellites. Les données sont envoyées à l'Arduino qui lit les coordonnées en extrayant la chaîne \$GPGGA du récepteur GPS et transmises au GSM en termes de coordonnées (latitudes et longitude) .Le module GSM envoie les coordonnées par SMS, et utilisé un écran LCD 16x2 est également pour afficher les coordonnées, comme le montre les figures IV.37 et IV.38.

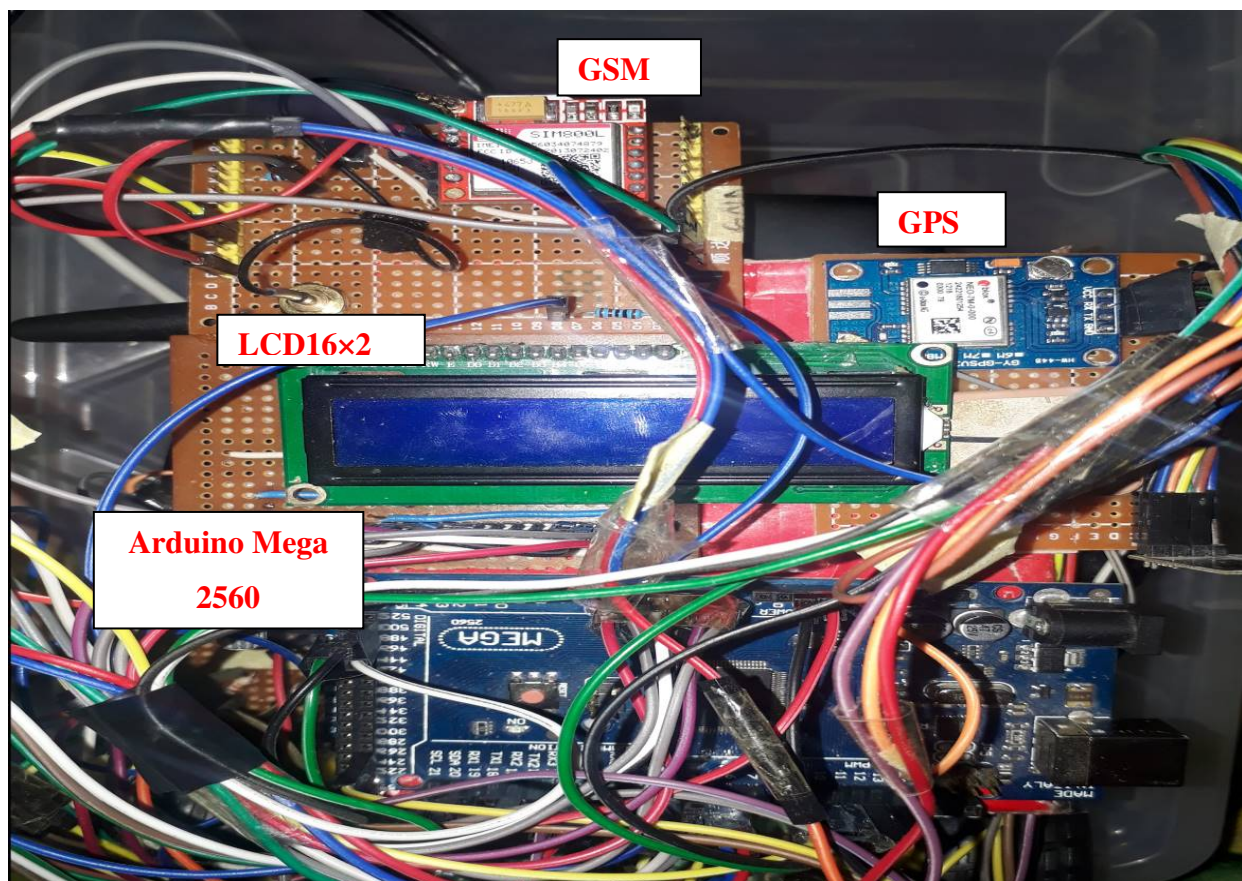


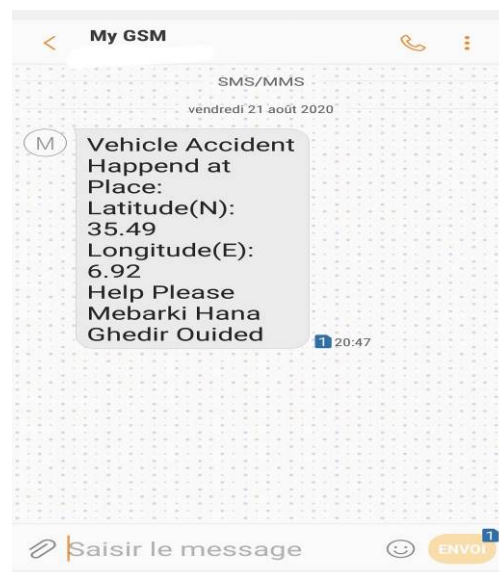
Figure IV. 37. Expérimental du Système de message d'urgence.



**Figure IV. 38.** Affichage des positions géographiques (latitude et longitude)

Aussi on peut utiliser LCD 16x2 pour afficher la distance entre les véhicules qui est calculée par le capteur ultrason HC-SR04 qui est situé à l'avant de chaque véhicule.

Afin de démontrer le fonctionnement du système de message d'urgence du véhicule avec succès, un téléphone portable a été configuré pour obtenir la position géographique du véhicule par SMS, le message reçu envoyé par le système d'alerte de véhicule est illustré sur la figure suivante :



**Figure IV. 39.** Texte SMS reçu envoyé par le système d'alerte de véhicule.

## IV. 5. Conclusion

Dans ce dernier chapitre on a projeté la lumière sur l'environnement de travail, et les composants électroniques qu'on a utilisé pour réaliser le prototype de système VNET ainsi que les schémas de chaque bloc. Ensuite, on a présenté l'efficacité et les performances de

notre algorithme de détection et reconnaissance automatique des panneaux de signalisation implémenté sur la carte Raspberry *Pi*. Enfin, on a fait plusieurs tests afin d'éclairer mieux le fonctionnement de notre projet.

## *Conclusion générale*

---

## Conclusion générale

La communication dans les réseaux VNET est encore un domaine très actif, mais à ce moment, peu de protocoles de communication sont spécifiquement conçus pour les réseaux véhiculaires. Il faut donc adapter des mécanismes et des protocoles spéciaux à ce contexte particulier et qui répondent aux exigences et spécificités de ces réseaux.

Pour cela, notre première contribution a porté sur l'utilisation des techniques d'intelligence artificielle : réseaux de neurones MLP et machine à vecteurs de support SVM par l'intermédiaires des méthodes de segmentation et détection pour la reconnaissance automatique des panneaux de signalisations. Le développement de cette approche a été fondé sur deux étapes. La première étape, est de réfère à la proposition de l'architecture optimale pour les méthodes MLP et SVM ; pour atteindre notre objectif, nous avons utilisé quatre types de bases de données avec les dimensions de bloc : 16×16, 32×32, 48×48 et 64×64 pixels, ensuite un descripteur HOG est effectué sur chaque bloc. Nous pouvons dire, que le choix de la base d'apprentissage est très important dans la conception d'un classifieur MLP ou SVM.

Dans la deuxième étape, nous avons testé la robustesse et la performance des méthodes de l'intelligence artificielle développées (MLP, SVM), afin de l'utiliser sur des scènes réelles.

Les résultats obtenus sur les bases de données avec le classifieur SVM sont nettement meilleurs par rapport à ceux obtenus avec les réseaux de neurones MLP. Les meilleures performances de classifieur SVM peut être obtenue avec l'utilisation des méthodes décrite comme suit : en utilisant la méthode de segmentation : Régions extrêmes maximales stables (MSER), la détection par la méthode d'extraction des composantes connexes (ECC) avec les filtres morphologiques, et l'extraction des caractéristiques par le descripteur HOG avec un bloc de découpage de 64×64 pixels. Ces meilleurs paramètres sont utilisés dans notre système en temps réel avec la carte Raspberry *Pi* et une caméra.

Notre deuxième contribution a porté sur la réalisation d'un système de communication entre les véhicules (VNET) en utilisant les différents types de communication sans fil (radiofréquence, GPS, GSM). Pour atteindre notre objectif et obtenir des résultats proches de la réalité, nous avons proposé d'utiliser un prototype, constitué de deux robots mobiles et infrastructure, le premier robot se décompose de tous les éléments essentiels de notre système (Arduino Mega, afficheur, ultrason, nRF, GPS, GSM), tandis que le deuxième robot se décompose d'Arduino Uno, radiofréquence et ultrason. Ce dernier est utilisé pour tester la transmission et la réception entre les différents véhicules.

Pour l'infrastructure, on affiche la position du premier véhicule par des coordonnées GPS et pour l'identification des panneaux de signalisation avec l'utilisation de l'intelligence artificielle.

Notre système (VNET) est utilisé pour échanger des données entre les véhicules, et aussi entre véhicules et infrastructure. Dans le but d'améliorer la sécurité et l'efficacité des transports routiers ainsi que l'amélioration du confort de l'utilisateur en lui offrant une multitude des services d'information (accident, danger, collision, les panneaux de signalisations,...).

Outre, ce que nous avons mentionné précédemment, nous avons également ajouté un système de message d'urgence, pour générer et envoyer ce message, le véhicule doit connaître sa propre position via un système de positionnement GPS qui utilise le service par satellite. Pour transmettre ce message nous avons utilisé le module GSM, les données de position sont envoyées sous forme de SMS à un ou plusieurs contacts et aux services adaptés à la situation par exemple, la charge des problèmes médicaux urgents, les pompiers, ...etc. Ce message est généré et envoyé en cas de risque ou d'accident.

Nous espérons que ce projet sera bénéfique à toute personne souhaitant réaliser ce genre de systèmes et qu'il pourra être achevé par d'autres étudiants dans le but d'améliorer et d'approfondir encore plus leurs connaissances.

### **Les perspectives**

Ce travail nous a permis de traiter des problèmes d'ordre pratique et de vérifier des connaissances théoriques acquises le long de notre formation, spécialement dans le domaine de télécommunication et aussi de comprendre et pratiquer les techniques d'intelligence artificielle. Grâce au travail continu, on a pu atteindre notre but et satisfaire le cahier de charge, mais cela ne veut pas dire qu'il est complet, nous proposons que le travail réalisé soit la base de toute série d'améliorations que nous n'avons pas eu la chance de les faire par manque de temps et de matériel. Ainsi, parmi les inconvénients de ce type de projet est le coût plus élevé des composants (voir Annexe : Budget prototype). Le nombre d'améliorations que peuvent être ajoutés sont :

- L'étude approfondie de la phase de segmentation et détection des panneaux de signalisations:
  - Dans le cas de segmentation nous avons opté d'utiliser la méthode MSER en suite un simple seuillage. Ces différentes méthodes peuvent certainement être améliorées, avec l'utilisation d'autres techniques de seuillage.

- Pour la détection de plusieurs panneaux dans la même scène nous avons utilisé l'algorithme de l'étiquetage en composantes connexes (ECC), dans le but d'accélérer la vitesse d'exécution et/ou d'augmenter l'efficacité de notre méthode de détection, par exemple en effectuant la méthode de détection d'objet YOLO v3 (You Only Look Once version 3).
- Etude approfondie sur les méthodes de classification:
  - Utilisation d'autre méthode pour déterminer les meilleurs caractéristiques des images.
  - Utilisation d'autre techniques d'intelligence artificielle pour la classification automatique des panneaux de signalisations : apprentissage profond (deep learning).
- Utilisation des protocoles de routage sur les réseaux Ad-Hoc de véhicules (VANET).
- Relier les réseaux de véhicules à l'internet afin de fournir une grande gamme de services, pour le passager, le constructeur automobile ou l'exploitant d'infrastructure.

## *Annexe : Budget prototype*

Capteurs	Caractéristiques	Quantité	Prix unitaire (DA)	Prix (DA)
Raspberry Pi	RaspPi 3	1	11000	11000
Caméra	8Mp	1	3300	3300
Arduino	Uno	2	2800	5600
Arduino	Mega	1	3800	3800
Batterie	PowerBank	1	2400	2400
Radiofréquence	nRF24L01 avec antenne	1	1800	1800
Radiofréquence	nRF24L01 sans antenne	2	900	1800
GSM	GSM800L	1	3600	3600
GPS		1	4500	4500
Robot mobile 1	Châssis+2 moteurs CC	1	1700	1700
Robot mobile 2	Châssis+4 moteurs CC	1	3700	3700
L298N		2	1800	3600
Afficheur	LCD 16x2	2	1200	2400
Ultrason	HS-04	2	1000	2000
Connecteurs	fils	50	25	1250
<b>Total</b>				<b>52450</b>

## *Bibliographie*

---

# Bibliographie

- [1] A. Daniel, A. Paul, A. Ahmad, S. Rho, "Cooperative Intelligence of Vehicles for Intelligent Transportation Systems (ITS)," *Wireless Pers Commun*, vol. 87, pp. 461-484, 2015.
- [2] Y. Liu, P. Watta, B. Jia, Y. L. Murphey, "Vehicle position and context detection using V2V communication with application to pre-crash detection and warning," *2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens*, pp. 1-7, 2016.
- [3] Y. Meraihi, "Routage dans les réseaux véhiculaires (VANET) cas d'un environnement type ville," *Magister en génie électrique, Option: Infotronique, Université de Boumerdes*, 2011.
- [4] J. Greenhalgh M. Majid, "Real-time detection and recognition of road traffic signs," *IEEE transactions on intelligent transportation systems*, vol. 13, pp. 1498-1506, 2012.
- [5] A. Behloul Y. Saadna, "A Fast and Robust Traffic Sign Recognition," *International Journal of Innovation and Applied Studies*, vol. 5, pp. 139-149, 2014.
- [6] A. Abidi S. Zeroual, "Etude et conception d'un détecteur d'objet en utilisant la communication sans fil, équipé d'un Raspberry Pi et une caméra," *Mémoire de Master, Option : Systemes de Télécommunications, Université Abbes Laghrour Khenchela*, 2019.
- [7] L. He, Y. Chao, K. Suzuki, K. Wu, "Fast connected-component labeling," *Pattern recognition*, vol. 42, pp. 1977-1987, 2009.
- [8] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," *IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886-893, 2005.
- [9] K. T. Islam, R. G. Raj, G. Mujtaba, "Recognition of traffic sign based on bag-of-words and artificial neural network," *Symmetry* vol. 9, pp. 1-21, 2017.
- [10] A. Salhi, B. Minaoui, M. Fakir, H. Chakib, H. Grimech, "Traffic signs recognition using HP and HOG descriptors combined to MLP and SVM classifiers," *International Journal of Advanced Computer Science and Applications*, vol. 8, pp. 526-530, 2017.
- [11] R. Engoulou, "Sécurisation des VANETs par la méthode de réputation des noeuds," *Thèse de doctorat. École Polytechnique de Montréal*, 2013.
- [12] K. Moghraoui, "Gestion de l'anonymat des communications dans les réseaux véhiculaires Ad hoc sans fil (VANETs)," *Thèse de doctorat. Université du Québec à Trois-Rivières*, 2015.
- [13] R. Meraihi, S. M. Senouci, D. E. Meddour, M. Jerbi, "Vehicle to Vehicle Communications: Applications and Perspectives," *Wireless Ad Hoc and Sensor Networks, Editor(s): Houda Labiod*, pp. 285-308, 2008.
- [14] S. Khan, "Collision avoidance system proposed by a model using NRF24L01 and infrared sensor," *Mémoire de Master, Option: Informatique et Ingénierie, Université BRAC*, 2018.
- [15] T. M. Saad, "Détecteur et brouilleur de téléphones mobiles, ," *Mémoire : Ingénieur CNAM En Electroniques, Institut des sciences appliquées et économiques, Université Libabaise*, 2010.
- [16] J.-P. Muller, "Le réseau GSM et le mobile," Version 07/2002.
- [17] L. Chamek, "Localisation des mobiles par une stratégie de prédiction," *Localisation des mobiles par une stratégie de prédiction, Mémoire de Magister, Option: Specification logiciels et traitement de l'information, Université M'hamed Bougara, Boumerdes*, 2011.

- [18] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, *Global positioning system: theory and practice*, 2012.
- [19] E. Kaplan, C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [20] R. B. Langley, "The evolution of the GPS receiver," *GPS World*, vol. 11, pp. 54-58, 2000.
- [21] M. Kara, "Réseau de capteurs sans fil: étude en vue de la réalisation d'un récepteur GPS différentiel à faible coût " *Thèse de doctorat Université Blaise Pascal-Clermont-Ferrand II*, 2009.
- [22] S. Malys, J. Slater, "Maintenance and enhancement of the World Geodetic System 1984," *In Proceedings of the 7th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1994)*, pp. 17-24, 1994.
- [23] L. Arouche, N. Mezhoud, "Etude de la géolocalisation dans le réseau GSM," *Mémoire de Master, Option: Télécommunication, Université Abderrahmane Mira Béjaïa*, 2012.
- [24] N. Cherchouri, M. Sadjji, "Etude et réalisation d'un système d'assistance des malades atteints de la sclérose en plaque (The Hacoving System)," *Mémoire de Master, Option: Imagerie et Appareillage Biomédical, Université M'Hamed Bouguara Boumerdes*, 2017.
- [25] H. Bendjafer, A. Midoun, "Système de contrôle par GSM," *Mémoire de Master, Option: Instrumentation Electronique, Université de AbouBekr Belkaid, Telemcen*, 2017.
- [26] J. Bégard, N. Allezard, P. Sayd, "Real-time humans detection in urban scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008.
- [27] B. Soheilian, Aurore Arlicot, N. Paparoditis, "Extraction de panneaux de signalisation routière dans des images couleurs," *Proceedings of the 17th French-speaking conference on Pattern Recognition and Artificial Intelligence on (RFIA), Caen, France*, 2010.
- [28] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323-332, 2012.
- [29] R. Ayachi, M. Afif, Y. Said, M. Atri, "Traffic signs detection for real-world application of an advanced driving assisting system using deep learning," *Neural Processing Letters*, vol. 51, pp. 837-851, 2020.
- [30] F. Suard, A. Rakotomamonjy, A. Bensrhair, A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," *IEEE Intelligent Vehicles Symposium*, pp. 206-212, 2006.
- [31] C. Touzet, "les réseaux de neurones artificiels, introduction au connexionnisme: cours, exercices et travaux pratiques," 1992.
- [32] M. Parizeau, "Réseaux de Neurones GIF-21140 et GIF-64326," *Université de Laval*, 2004.
- [33] B. Wierenga, J. Kluytmans, "Neural nets versus marketing models in time series analysis: A simulation study " *Marketing: Its Dynamics and Challenges*, pp. 1139-1153, 1994.
- [34] Y. Ammar, "Mise en oeuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition Batch/Continu," *Thèse de doctorat I.N.P. Toulouse*, 2007.
- [35] M. Parizeau, "Réseaux de neurones," *Thèse de doctorat. Université LAVAL*, 2004.
- [36] V. N. Vapnik, "Statistical Learning Theory," *Edition Wiley*, 1998.

- [37] O. Bousquet, "Introduction au Support Vector Machines (SVM) " *Centre de mathématiques appliquées, Ecole polytechnique de Palaiseau*, 2001.
- [38] J. Shawe-Taylor, N. Cristianini, "Support Vector Machines and other kernel based learning methods," *Cambridge University Press*, 2000.
- [39] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, "Choosing kernel parameters for support vector machines," *Machine Learning*, vol. 46, pp. 131–160, 2001.
- [40] T. Joachims, "Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning," *MIT Press Cambridge, MA, USA*, 1999.
- [41] L. Hamel, "Knowledge discovery with support vector machines," *Wiley Edition*, 2009.
- [42] A. Shigeo, "Support Vector Machines for Pattern Classification," *Springer-Verlag London Limited*, 2005.