

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mouloud Mammeri Tizi-Ouzou  
Faculté de Génie Electronique et Informatique  
Département Informatique



En vue d'obtention du diplôme de Master Académique en  
informatique  
Option : Réseaux, Mobilité et Systèmes Embarqués

*Thème : Boite noire pour véhicule*

Réalisé par :  
Achour brahim  
Brahimi cherif

Dirigé par :  
Mr LAGHROUCHE Mourad  
Président de jury :  
Mr DAOUI Mehammed

Examineurs :  
Mme SAAD-BELKADI Malika  
Mr BENAMANE Kamal

## *Sommaire*

<i>Introduction Générale</i> .....	9
<b>Chapitre 1 : état de l'art</b>	
1. Introduction .....	12
2. Les Accidents de route. ....	12
2.1 Cause d'accident de routes et facteur de risque.....	12
2.2 Conséquence des accidents de route.....	13
2.3 Population exposé.....	13
2.4 Statistique des accidents de route en Algérie .....	14
2.5 Mesures à prendre en considération pour améliorer la sécurité .....	14
3. L'électronique Automobile .....	15
3.1. Les composants électroniques dans un véhicule .....	15
3.1.1Fonctionnement de Système ESP .....	15
3.1.2Le Fonctionnement du boîtier ABS .....	16
3.2 Le multiplexage dans un véhicule .....	18
3.2.1Définition de Multiplexage .....	18
3.2.2L'intérêt du multiplexage .....	19
3.2.3Adaptation des boîtiers (calculateurs) au multiplexage .....	19
3.2.4Les types des réseaux multiplexé .....	20
3.3. Le bus CAN (Controller Area Network) .....	21
3.3.1 Protocole CAN.....	21
3.3.2 Caractéristiques physiques du Bus .....	21
3.3.3 Structure des trames CAN .....	22
3.3.4 Les Identifiants CAN .....	24
3.3.5Fonctionnement détaillé de l'arbitrage .....	24
3.4. Le Diagnostics automobile (OBD).....	25
3.4.1 Les Protocoles OBD II .....	25
3.4.2 Le connecteur OBD-II .....	26
3.4.3 Les Modes de fonctionnement de L'OBD-II.....	26
3.4.4Le principe de fonctionnement des PID : .....	27

4. Conclusion .....	28
---------------------	----

## ***Chapitre 2 : la connectivité automobile***

1. Introduction .....	30
2. La connectivité automobile .....	30
2.1 La voiture connectée .....	30
2.2 Domaine de la connectivité automobile .....	30
2.3 Les protocoles sans fil automobile .....	31
2.4 Comparaison entre les protocoles sans fil automobile .....	32
3. Le protocole ZigBee .....	32
3.1 La norme IEEE 802.15.4 .....	33
3.2 Les couches du protocole ZigBee .....	33
3.3 Les équipements ZigBee .....	33
3.4 Les topologies du réseau ZigBee.....	34
4. Interface ZigBee (Le module XBee) .....	35
4.1 Présentation .....	35
4.2 Différentes versions des modules Xbee .....	35
4.3 Mode de fonctionnement des modules Xbee .....	35
4.4 Paramétrage des modules XBEE .....	36
4.5 Adressage des modules XBEE .....	36
4.6 Les modes de communication .....	36
5. Conclusion .....	37

## ***Chapitre 3 : moyen et conception***

1. Introduction .....	39
2. Objectifs du système .....	39
3. Fonctionnalité du système .....	38
4. Structure du système .....	40
4.1 Composant requis .....	41
4.2 Module de traitement.....	42
4.2.1 Présentation de la carte Arduino ADK .....	42

4.2.2 Alimentation de la carte Arduino ADK.....	42
4.2.3 Gestion des mémoires dans la carte.....	42
4.2.4 Les broches de la carte Arduino ADK.....	42
4.3Module d'acquisition.....	43
4.3.1 Bus CAN Shield .....	43
4.3.2 Capteur d'alcoolémie MQ 135 .....	46
4.4Module de Communication sans fil.....	47
4.4.1Présentation de Xbee Pro Séries 1 .....	47
4.4.2Caractéristique .....	47
4.5Module d'affichage .....	48
4.5.1 La SDK Android (Software Development Kit).....	48
4.5.2 L'IDE Eclipse .....	48
4.5.3 Le Plugin ADT pour Eclipse .....	48
4.5.4 SQLITE.....	49
4.5.5 Smartphone .....	49
5. Conception .....	50
5.1 Les graphes d'appels .....	50
5.2 Le graphe de flux de données .....	51
5.3 Le graphe de Raffinement successives .....	52
6. Conclusion .....	54

#### ***Chapitre 4 : Réalisation et Test***

1. Introduction .....	56
2. Principe de fonctionnement du système .....	56
3. Réalisation matérielle .....	57
3.1 Carte Arduino Mega Adk et CAN bus shield .....	57
3.2 Carte Arduino Mega Adk et Xbee Pro Series1 .....	58
3.3 Carte Arduino Mega Adk et MQ135 .....	59
3.4 Carte Arduino Mega Adk et Smart Phone .....	59

3.5 Schème Global du branchage dans le Système embarqué véhicule .....	60
3.6 Schème Global du branchage dans le Système embarqué borne de police .....	60
4. Présentation des applications du système .....	61
4.1 lancement de l'application .....	61
4.2 système embarqué dans un véhicule .....	61
4.3 système embarqué dans borne de police .....	62
5. Elaboration du programme .....	63
5.1 Communication intra-système .....	63
5.2 Communication entre les deux systèmes .....	64
5.3 Application Android dans borne de police .....	64
5.4 Application Android dans le véhicule .....	66
6. Test .....	68
6.1 Présentation d'ULM 327 .....	68
6.2 Véhicule testé .....	69
6.3 Test N°1: Volkswagen Passat .....	69
6.4 Test N°2: Volkswagen Golf 5 .....	72
6.5 Test N°3: Skoda Fabia .....	74
7. Conclusion .....	78
Conclusion générale .....	80
<b>Liste des tableaux :</b>	
Tableau 1 : Nombre de Morts dans les accidents de route en Europe .....	13
Tableau 2 : Comparaison entre CAN L et CAN H .....	21
Tableau 3 : Présentation de quelque PID .....	27
Tableau 4: Décodage de trame de réponse en mode 01 avec pid 0x00.....	28
Tableau 5: Comparaison entre les différents protocoles de communication sans fil.	32
Tableau 6 : comparatif entre Xbee et Xbee Pro .....	35
Tableau 7: Modes de fonctionnement du MCP2515 .....	44
Tableau 8: les filtres et masque d'acceptation du MCP2515 .....	46
Tableau 9: Les broches d'alimentation entre carte ADK et CAN.....	57

Tableau 10 : les broches de communication entre carte ADK et CAN bus shield...	57
Tableau 11 : Les broches d'alimentation entre ADK et Xbee Pro S1 .....	58
Tableau 12 : Les broches de communication entre ADK et Xbee Pro S1.....	58
Tableau 13 : Les broches d'alimentation entre ADK et MQ 135.....	59
Tableau 14 : La broche de communication entre MQ 135 et ADK .....	59
Tableau 15: information sur le véhicule du teste N°1 .....	69
Tableau 16 : Information sur le véhicule du Test N°2 .....	72
Tableau 17: Information sur le véhicule de teste N°3 .....	74

### Liste des Figures :

Figure 1: Exemple de fonctionnement de Boitier ESP.....	16
Figure 2: Capteur ABS passif.....	17
Figure 3: Exemple de câblage classique sans multiplexage .....	18
Figure 4: Exemple de câblage avec multiplexage .....	18
Figure 5: Boitier multiplexé .....	19
Figure 6: Schéma d'interconnexion des deux protocoles CAN et VAN chez PSA..	20
Figure 7 : L'architecture CAN dans un véhicule .....	21
Figure 8 : trame de Donnée .....	22
Figure 9 : fonctionnement d'arbitrage .....	24
Figure 10 : Connecteur OBD II .....	26
Figure 11: Les protocoles de communication sans fils automobile .....	31
Figure 12 : Topologie Zigbee .....	32
Figure 13 : structure générale du système .....	40
Figure 14: la carte Arduino Mega ADK.....	42
Figure 15: Sparkfun Can Bus Shield .....	43
Figure 16 : block diagramme du micro contrôleur MCP2515 .....	44
Figure 17 : Capteur d'alcoolémie MQ 135.....	46
Figure 18 : le circuit du capteur MQ135 .....	46
Figure 19: Xbee Pro Series 1 .....	47

Figure 20 : les branches de l'xbee pro séries 1 .....	47
Figure 21 : équipement Android.....	48
Figure22: Le graphe d'appels pour le système embarqué dans le véhicule .....	50
Figure 23 : Le graphe d'appels pour le système embarqué dans la borne de police.	51
Figure 24: Le graphe de flux de données pour le système embarqué véhicule.....	51
Figure 25: Le graphe de flux de données pour le système embarqué dans la borne de police.....	52
Figure 26: décomposition d'une tache en sous taches.....	52
Figure 27: le graphe de Raffinement successives pour le système embarqué dans le véhicule.....	53
Figure 28: le graphe de Raffinement successives pour le système embarqué dans la borne de police .....	54
Figure 29:Schémas fonctionnel de système .....	56
Figure 30: schéma de brochage du CAN Bus Shield avec Arduino ADK.....	57
Figure 31: Le Brochage CAN Bus Shield et OBD2.....	58
Figure 32: schéma de brochage du Xbee Pro Series1 et ADK.....	58
Figure 33: schéma de brochage du MQ135 et ADK .....	59
Figure 34: Câble Mini USB.....	59
Figure 35 : Schéma de brochage général coté véhicule .....	60
Figure 36 : Schéma général du brochage borne de police.....	60
Figure 37:Confirmation de communication entre l'application Android et l'Arduino ADK via le câble USB .....	61
Figure 38: Présentation de l'application Android information véhicule .....	61
Figure 39: Exemple de l'application lors de réception d'un rapport coté police ....	62
Figure 40:Présentation des opérations de l'application Android .....	62
Figure 41:Schéma général des buffers existant dans le système .....	64
Figure 42:Diagramme de classe pour l'application borne de police .....	65
Figure 43: Présentation de la Classe open Accessoire test.....	66

Figure 44:Présentation class main .....	67
Figure 45: Interface ELM327 .....	68
Figure 46: présentation du logiciel EOBD facile .....	68
Figure 47 : Volkswagen Passat .....	69
Figure 48 : emplacement prise OBD dans la Passat.....	70
Figure 49: photo prise lors du teste effectué .....	70
Figure 50: Résultat du Test 1.....	71
Figure 51 : capture plus claire des résultats obtenus .....	71
Figure 52 : Volkswagen Golf 5 .....	72
Figure 53 : branchement de la prise obd dans la Golf 5 .....	72
Figure 54: image réel du test réaliser sur la Golf 5 .....	73
Figure 55 : Résultat obtenu du test .....	73
Figure 56 : Skoda Fabia.....	74
Figure 57 : branchement effectué dans le test .....	74
Figure 58 : image réel du résultat obtenu .....	75
Figure 59 : capture d'écran d'un dépassement de la limite réglementaire de la vitesse dans une zone urbaine .....	76
Figure 60 : capture d'écran d'un dépassement de la limite réglementaire du taux d'alcoolémie .....	76
Figure 61 : réception du rapport dans la borne de police .....	77
Figure 62 : résultats obtenus.....	77

# Introduction générale

## Introduction générale

A la fin du 19<sup>ème</sup> siècle une invention technologique majeure est apparue et elle a considérablement modifié les sociétés de nombreux pays au cours du 20<sup>ème</sup> siècle. Cette invention est l'automobile.

L'automobile s'est ainsi progressivement imposée dans les pays développés comme le principal mode de transport pour la circulation des individus et des marchandises et elle a favorisé le développement des échanges économiques et culturels et a conduit au développement massif de nouvelles infrastructures (routes, autoroutes, parkings...)[1].

Cependant l'inconvénient majeur est l'accident qu'elle peut provoquer à tout moment, qui est généralement causé par plusieurs facteurs tels que l'excès de vitesse, l'alcool au volant...etc.

Afin de réduire les accidents de route, des systèmes de sécurité intégrés rendent la conduite plus fiable (freinage, direction assistée, etc..) et mieux visible (feux de croisement etc...), mais aussi, renforcer la protection offerte en cas de survenue d'un accident (airbags, ceinture de sécurité). Ainsi la sécurité routière crée de nombreux domaines de recherche qui induit aux développements des nouvelles technologies (aide à la navigation, régulation automatique de la vitesse, alerte en cas de véhicule arrêté sur la voie, détection du taux d'alcoolémie etc) ,dans ces recherches, le véhicule « intelligent » est amené à remplacer le conducteur défaillant.[2]

De nos jours, les systèmes de sécurité automobile contribuent à produire des véhicules plus Sûrs [3]. Mais restent coûteux.

L'objectif de notre travail consiste à réaliser une boîte noire pour récupérer des données d'un véhicule (tel que la vitesse, régime moteur, taux d'alcoolémie etc...), en temps réel et de les envoyer à une base de police en cas de dépassement de certaine limite. Cette boîte noire va contribuer à renforcer la sécurité par des mesures dissuasives envers les conducteurs qui ne se conforment pas à la réglementation.

Ce mémoire s'organise en quatre chapitres, à travers lesquels nous décrivons le travail effectué pour la conception et la réalisation de notre système :

- ✓ **Chapitre I :** état de l'art, dans lequel nous abordons les accidents de route, le fonctionnement et l'architecture électronique dans un véhicule et le diagnostic automobile.
- ✓ **Chapitre II :** Connectivité automobile, dans lequel nous étudions les différents protocoles de communication sans fil utilisées dans l'automobile et nous présentons la norme zigbee.
- ✓ **Chapitre III :** moyens et conception, dans lequel nous présentons l'objectif, la structure et les composants du système puis nous décrivons la phase de conception.

- ✓ **Chapitre IV** : réalisation pratique et tests. Dans lequel nous décrivons notre système ainsi son brochage électronique détaillé, Nous présentons également des exemples de codes et quelques interfaces graphiques constituant l'application et présentant des tests pratiques.

# Chapitre I : Etat de l'art

## 1. Introduction

Le transport routier présente des avantages tant pour les nations que pour les particuliers dans la mesure où il facilite la circulation des marchandises et des personnes. Il permet un meilleur accès à l'emploi, aux marchés, à l'éducation, aux loisirs et aux soins, ce qui a directement et indirectement un effet positif sur le confort des populations. Cependant, l'intensification de la circulation routière est l'origine des accidents de la route.

La technologie permet désormais d'envisager des systèmes de sécurité visant à éviter les accidents ou à les réduire grâce à la compatibilité liée entre les composants automobiles et ces systèmes.

Ce chapitre aura donc pour objectif de présenter les accidents de route en premier lieu puis donnera un aperçu sur le fonctionnement de l'électronique automobile, et en présentant l'un de ses bus de communication qui est le bus CAN et se terminera par l'étude de diagnostic automobile OBD.

## 2. Les Accidents de route :

Les accidents de la circulation provoquent chaque année plus d'1,2 million de décès et 20 à 50 millions de traumatismes. Dans la plupart des régions, cette épidémie continue à prendre de l'ampleur. Le taux de mortalité attribuable aux accidents de la route est plus élevé dans les pays à revenu faible ou intermédiaire (21,5 et 19,5 pour 100 000 habitants respectivement) que dans les pays à revenu élevé (10,3 pour 100 000 habitants). Plus de 90 % des décès surviennent dans les pays à revenu faible ou intermédiaire [4]. Les accidents de la circulation restent une cause importante de décès, de traumatisme et de handicap.

### 2.1 Cause d'accident de routes et facteurs de risque :

Les causes principales des accidents de route sont :

#### ❖ Comportement de l'utilisateur de la route :

Le comportement de l'utilisateur de la route est dans la plupart des cas la cause principale des accidents de la circulation.

#### ❖ Fatigue, baisse de vigilance et somnolence :

On constate qu'une conduite consécutive de deux heures peut conduire à une baisse de vigilance, ce qui peut être une cause d'accident. Ce facteur enregistre un pic d'accident à quatre heures du matin et un autre entre treize et seize heures. [5]

#### ❖ La Vitesse :

La vitesse joue un rôle aggravant en cas d'accident. L'énergie du choc est également plus importante (proportionnelle au carré de la vitesse, une vitesse 20 % plus élevée, provoquera un choc 44 % plus violent). La probabilité d'accident mortel est proportionnelle à la puissance 4 de la vitesse moyenne (en ville, à la puissance 2,5).[6]

La vitesse provoque notamment :

- un risque de perte d'adhérence plus important, notamment en cas de coup de volant.
- un allongement de la distance d'arrêt.

### ❖ Substances influençant le comportement :

Certaines substances psycho actives ou drogues et certains médicaments en particulier, les hypnotiques et les tranquillisants, influencent le comportement en ralentissant les réflexes et en diminuant la vigilance.

### ❖ Les infrastructures routières :

Des infrastructures routières mal adaptées (zones dangereuses non balisées, intersections sans visibilité...) peuvent aggraver le risque d'avoir un accident.

## 2.2 Conséquence des accidents de route :

Les accidents de la route provoquent non seulement des décès (comme le montre le tableau1) mais aussi de graves traumatismes, on estime qu'environ 20 à 50 millions de traumatismes surviennent chaque année dans le monde et ceux-ci sont une cause importante de handicap.

Il est ressorti d'une étude menée en Turquie [4] qu'environ 13 % des 95 000 personnes ayant survécu à un accident de la route en 2005 en avaient gardé des séquelles handicapantes, alors qu'en Inde, quelque 2 millions de personnes présentent un handicap suite à un accident de la route.

Les accidents de la route ont aussi de lourdes conséquences sur l'économie de nombreux pays, notamment ceux à revenu faible ou intermédiaire, qui luttent souvent pour satisfaire d'autres besoins de développement.

Pays	Nombre de tués											
	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
France	8 160	7 654	6 059	5 530	5 318	4 709	4 620	4 275	4 273	3 992	3 970	3 645
Allemagne	6 977	6 842	6 600	5 842	5 361	5 091	4 949*	4 477*	4 152			
Pologne	5 534	5 827	5 640	5 712	5 444	5 243	5 583	5 437				
Espagne	5 517	5 347	5 394	4 741	4 442	4 104	3 823	3 100				
Royaume-Uni	3 598	3 431	3 658	3 368	3 201	3 172	2 946	2 538				
Pays-Bas	993	987	1 028	804	817	730	709	677	644	640		
Suisse	544	513	546	510	409	370	384	357	349	327		

**Tableau 1** : Nombre de Morts dans les accidents de route en Europe de 2001 à 2012

## 2.3 Population Exposé :

Près de la moitié des personnes qui meurent dans un accident de la route sont des piétons, des cyclistes ou des usagers de deux-roues motorisés des « usagers vulnérables » cette proportion étant plus élevée dans les pays pauvres.

Les accidents de la route concernent toutes les tranches d'âge mais ils touchent plus particulièrement les jeunes.

## 2.4 Statistique des accidents de route en Algérie :

En 2014, 24 388 accidents ont été déplorés, faisant 3 984 morts et 44 546 blessés soit 67 accidents par jour avec 11 morts et 122 blessés, a indiqué le responsable de la sécurité routière au commandement de la gendarmerie nationale, qui présentait le bilan des accidents de la circulation.[7]

La comparaison entre les deux périodes 2014 et 2015 donne les résultats suivants :

- **Nombre de mort :** durant les 9 premiers mois de 2015 2.949 personnes sont mortes dans des accidents de la route contre 3.129 durant la même période de 2014, soit une baisse de 180 cas (-5,75%).
- **Nombre de blessés :** le nombre de blessés, enregistré durant la même période de l'année 2015, s'élève à 29.479 cas, soit également une baisse comparativement à celui enregistré durant la même période en 2014 (35.586 cas), représentant un recul de 6.107 cas (-17,16%).
- **Nombre d'accidents signalés :** quant aux nombre d'accidents signalés, il est passé de 19.262 en 2014, à 16.241 en 2015, soit une baisse de 3.021 accidents (-15,68%).  
Le bilan fait également une moyenne de 11 décès, de 130 blessés et 70 accidents enregistrés quotidiennement durant les neufs premiers mois de 2015.

## 2.5 Mesures à prendre en considération pour améliorer la sécurité :

- interdiction du port d'écouteurs, oreillettes ou casque audio en conduisant. Elle s'ajoute à l'interdiction de l'usage du téléphone portable au volant.
- l'obligation pour les motos, cyclomoteurs, scooters d'avoir une plaque d'immatriculation.
- l'augmentation nombre de radar.
- l'interdiction de circulation pour les véhicules ayant consommé une durée de vie de plus de 15 ans.
- Encourager l'achat de véhicules équipés de technologie de sécurité visant à améliorer la sécurité routière.
- Mettre en place un service chargé de repérer sur le réseau routier les différentes dégradations de la chaussée et capable d'intervenir rapidement pour la réhabilitation.
- La mise en place d'un système d'information et de communication capable d'informer à temps réel les usagers de la route sur la situation climatique et les mesures de sécurité à prendre.

### 3. L'électronique Automobile :

L'électricité, l'électronique et le logiciel automobile permettant de compléter les principaux composants mécaniques et hydrauliques (direction, freinage, suspension...) tous les modules de la voiture sont rendus plus intelligents donc le couplage des fonctions par l'électronique est possible. L'intégration de l'électronique dans les véhicules a permis de donner de nouvelles fonctionnalités telles que l'amélioration du confort et la sécurité.

#### 3.1. Les composants électroniques dans un véhicule :

Pratiquement tous les véhicules présentent une architecture électronique qui se compose de plusieurs périphériques, qui se communiquent entre eux et ces derniers se décomposent en trois groupes :

- **les actionneurs** : moteurs électriques, électro-aimants, sirènes, klaxons, ...  
Ces actionneurs sont généralement intégrés dans des pièces mécaniques.
- **les capteurs** : capteurs de pluie, de lumière, de position, de tension, de vitesse etc., ces capteurs sont pour la plupart équipés de petites parties électroniques.
- **les systèmes complets (boîtiers)** : Ces systèmes sont des ensembles électroniques complets, ils reçoivent ou émettent des informations soit en interrogeant un actionneur ou en recevant une information d'un capteur ce qui permet de gérer les défauts des ces capteurs et actionneurs, et ils peuvent aussi partager des informations entre eux par le bus de communication, parmi ces systèmes : tableau de bord, afficheurs, calculateurs d'airbags, calculateurs d'ABS, calculateurs ESP, surveillance de la pression des pneus, boîtier de climatisation etc...

**Dans ce qui suit nous allons présenter quelque système complet tel que :**

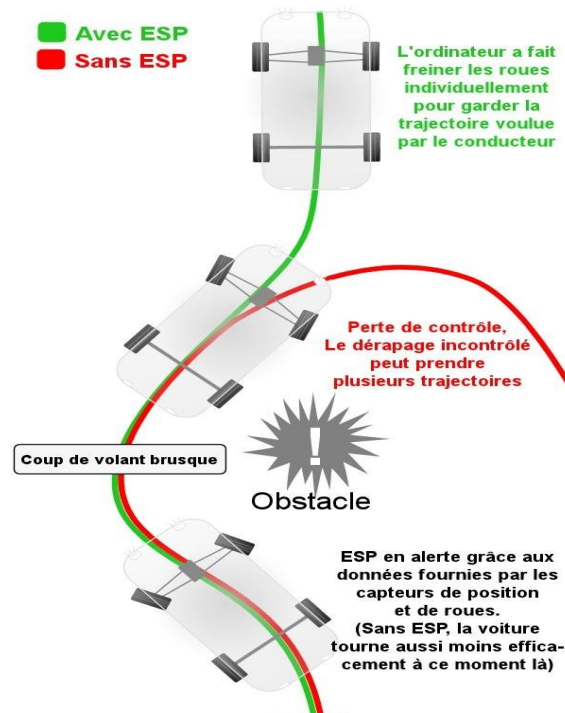
- ❖ Le calculateurs ESP
- ❖ Le calculateurs ABS

#### 3.1.1 Fonctionnement de Système ESP (Electronic Stability Program) :

Développé par Bosch pour Mercedes et BMW dans les années 1990, ce système corrige les écarts de trajectoire en agissant sélectivement sur les freins. Si la voiture tourne plus que le conducteur ne le souhaite parce que l'essieu arrière dérive vers l'extérieur, la roue avant extérieure au virage est freinée. À l'inverse, si elle tourne moins que ne le voudrait le conducteur parce que l'essieu avant glisse vers l'extérieur, c'est la roue arrière intérieure qui est freinée. À chaque fois, l'ESP se sert de la roue freinée comme d'un pivot pour ramener la voiture sur la trajectoire voulue.[8]

Selon Bosch, l'ESP pourrait éviter 80 % des accidents consécutifs à un dérapage non contrôlé.

La figure 1 montre un exemple sur le fonctionnement ESP



**Figure 1:** Exemple de fonctionnement de Boitier ESP

### Utilité du système ESP :

- Freinage droit malgré des différences d'adhérence sur le sol. Exemple : une ou plusieurs roues sur le verglas.
- Aide au braquage.
- Eviter la perte de contrôle : en donnant un coup de volant brusque.

### 3.1.2 Le Fonctionnement du boitier ABS :

Les capteurs ABS (Antilock Braking System) ou capteurs de vitesse de roue, se trouvent dans les dents des roues et détectent la vitesse de rotation de chacune d'elles, puis transfère cette information à l'unité de contrôle du système ABS. Si l'unité de contrôle du système ABS capte qu'il se produit une baisse importante de vitesse sur une ou plusieurs roues, ce qui pourrait provoquer leur blocage, elle intervient rapidement en modulant la pression de freinage de chaque roue individuellement. Ainsi les roues ne se bloquent pas et garantissent un freinage sûr, ce qui permet au conducteur de conserver le contrôle du véhicule.[9]

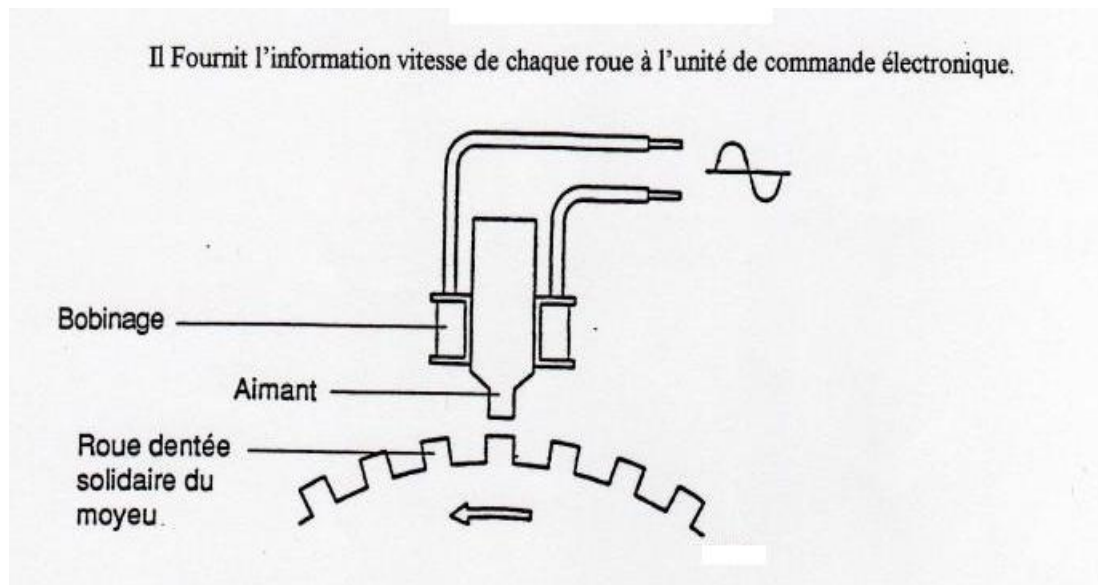
Le système ABS utilise non seulement l'information que lui envoient ses capteurs mais aussi d'autres systèmes, comme la boîte de vitesses.

Les capteurs de vitesse de rotation de roue sont différenciés en raison de leur fonctionnement en capteurs actifs et passifs. Une classification claire n'est pas définie.

- Si un capteur est activé par l'application d'une alimentation en tension et génère alors un signal de sortie, ce capteur est désigné comme actif.
- Si un capteur travaille sans alimentation en tension supplémentaire, ce capteur est caractérisé comme étant passif.

### Capteurs ABS passifs pour les roues dentées :

Les capteurs ABS passifs sont habituellement plus grands, ils ont une moindre précision et ne commencent à fonctionner que lorsque la roue atteint une vitesse rotationnelle minimale établie. Ils se composent d'une bobine qui entoure un noyau magnétique et un aimant permanent. Le flux magnétique qui se produit entre les dents et les creux de la roue induit une tension sinusoïdale de sortie qui est proportionnelle à la vitesse des variations que détecte le capteur. Comme le montre la figure 2



**Figure 2:** Capteur ABS passif

### Capteurs ABS actifs :

Les capteurs ABS actifs portent ce nom car ils ont besoin d'un apport externe d'énergie pour commencer à fonctionner.

La réponse que nous obtenons du capteur est un signal carré dont la fréquence est proportionnelle à la vitesse de rotation de la roue.

### 3.2 Le Multiplexage dans un véhicule :

L'augmentation constante des fonctions électroniques présentes dans les véhicules (ABS, climatisation, navigation, radar...) a fait que tous ces systèmes ont besoin d'un grand nombre d'informations pour fonctionner. Certaines de ces informations peuvent être communes à deux systèmes et parfois dépendent de l'état d'un autre système [10].

#### 3.2.1 Définition de Multiplexage :

Le multiplexage consiste à faire circuler plusieurs informations entre divers équipements sur un seul canal de transmission.

#### le câblage classique sans multiplexage:

dans le câblage classique sans multiplexage certains capteurs ont des liaisons avec plusieurs boîtiers (calculateurs) ou existent en deux exemplaires en raison de leur localisation, les liaisons entre boîtiers sont de plus en plus nombreuses. comme le montre la figure 3

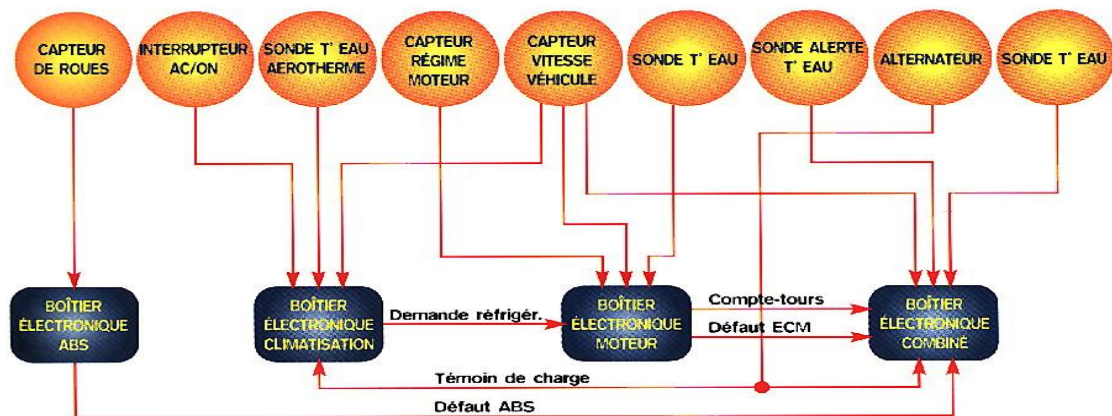


Figure 3: Exemple de câblage classique sans multiplexage

#### Le câblage avec multiplexage :

Le multiplexage diminue le nombre de capteurs et de liaisons entre boîtiers car chacun fournit aux autres, par l'intermédiaire du bus, les informations qu'il reçoit. Comme le montre la figure 4

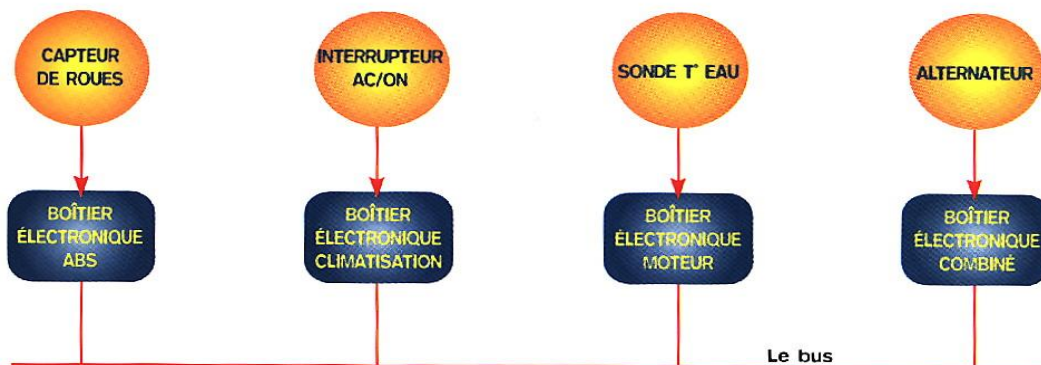


Figure 4: Exemple de câblage avec multiplexage

### 3.2.2 L'intérêt du multiplexage :

- La diminution du nombre de fils électriques
- La diminution du nombre de calculateurs et de capteurs
- La mise en commun d'informations
- La rapidité des informations circulant sur un bus
- Réduction de certains coûts de fabrication
- Facilité du diagnostic

### 3.2.3 Adaptation des boîtiers (calculateurs) au multiplexage :

Un boîtier multiplexé en général se compose de quatre interfaces comme le montre la figure 5.

#### ➤ L'étage d'entrée du boîtier :

Il transforme les signaux analogiques des capteurs en signaux numériques exploitables par le microprocesseur (ex : signal délivré par la thermistance d'eau).

#### ➤ L'étage de sortie du boîtier : (appelé étage de puissance) :

Il transforme les ordres, fournis par le microprocesseur sous forme de signaux numériques, en signaux analogiques destinés aux actionneurs.

#### ➤ L'étage de calcul : le microprocesseur, c'est le composant « intelligent » du boîtier.

#### ➤ L'interface de multiplexage :

- elle permet la communication entre le boîtier et le bus.
- les messages qui transitent par l'interface de multiplexage sont numériques et portent le nom de trames.[11]

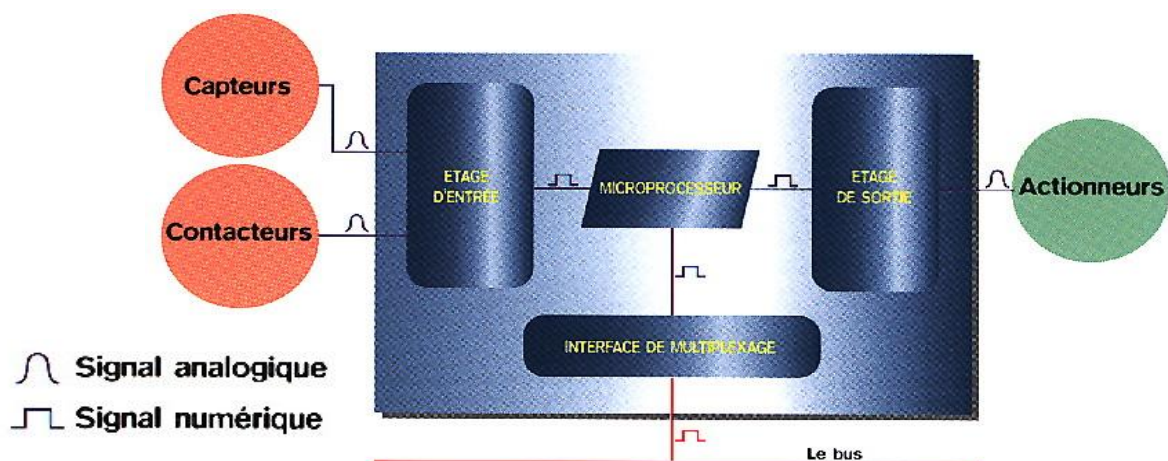


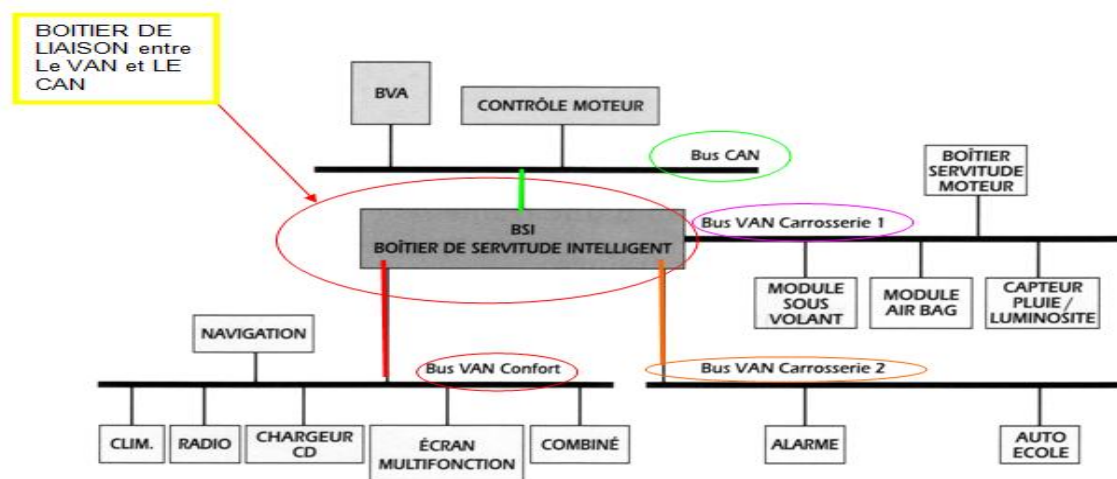
Figure 5: Boîtier multiplexé

### 3.2.4 Les types des réseaux multiplexés :

Le réseau est l'ensemble des boîtiers (calculateurs) qui communiquent entre eux, ce dernier se compose d'une architecture et d'un protocole.

- L'architecture du réseau est la disposition matérielle des boîtiers qui peut être : en étoile (VAN), en série (CAN) etc.
- Le protocole est utilisé pour la gestion de la communication entre les boîtiers (arbitrage, trame, horloge, débit).

Les constructeurs automobiles français Renault et Peugeot ont normalisé un type de réseau multiplexé le VAN (abréviation de Véhicule Area Network), Le constructeur d'équipement BOSCH a produit en même temps un standard appelé CAN (Controller Area Network). Pour le constructeur PSA le réseau VAN est couplé avec le réseau CAN (pour la gestion moteur) à l'aide d'un calculateur central appelé « BSI » Boîtier de servitude intelligent [12], comme montre la figure 6.



**Figure 6:** Schéma d'interconnexion des deux protocoles CAN et VAN chez PSA

- Le BSI sert de passerelle entre les deux standards de communication
- Le VAN s'adapte bien aux équipements de confort et de carrosserie
- Le CAN convient pour des échanges rapides : moteur et sécurité

### 3.3 Le bus CAN(Controller Area Network) :

Le protocole CAN (*Control Area Network*) est un protocole de communication série qui supporte des systèmes temps réel avec un haut niveau de fiabilité. Ses domaines d'application s'étendent des réseaux moyens débits aux réseaux de multiplexages faibles coûts.

La figure 7 montre une architecture de type CAN dans un véhicule

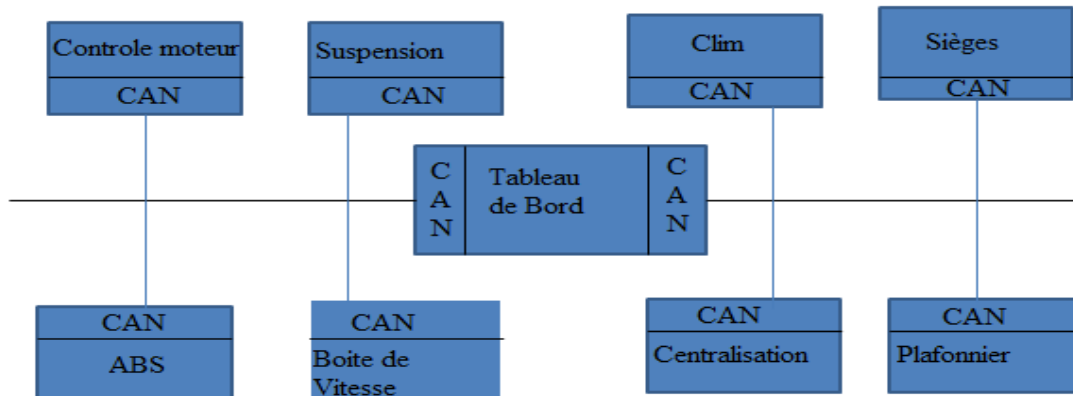


Figure 7: Architecture CAN dans un véhicule

#### 3.3.1 Protocoles CAN :

Le protocole CAN comporte deux sous spécifications qui diffèrent uniquement au niveau de la longueur de l'identifiant. La version A définit des ID de 11 bits et la version B définit des ID de 29 bits. On appelle ces trames respectivement des trames standards ("Standard Frames") et des trames étendues ("Extended Frames") [13].

#### 3.3.2Caractéristiques physiques du Bus CAN :

Le bus CAN utilise deux fils (une paire torsadée ou non) dont les désignations sont CAN L (Low) et CAN H (High), Il existe deux configurations du bus suivant le mode de travail souhaité qui sont low speed et high speed.

Le tableau 2 montre les caractéristiques des deux modes :

Parameters	CAN low speed	CAN high speed
Debit	125kb	125kb/s à 1 Mb/s
Nombre des nœuds sur le bus	2 à 20	2 à 30
Niveau dominant ( bit a 0 )	CAN H= 4V CAN L=1V	CAN H=CAN L=2V
Niveau récessif ( bit a 1 )	CAN H= 1.75V CAN L=3.25V	CAN H=CAN L=2.5V
Tensions d'alimentation	5V	5V

Tableau 2 : Comparaison entre CAN L et CAN H

### 3.3.3 Structure des trames CAN

Il existe cinq formats de trames CAN :

- trame de données
- trame de requête
- trame de gestion d'erreur
- trame de surcharge
- espace entre trames

#### 1) Trame de données :

Une trame de données se décompose en 7 champs différents :

- le début de trame SOF (*Start Of Frame*), 1 bit dominant.
- le champ d'arbitrage, 12 bits (identificateur).
- le champ de commandes, 6 bits.
- le champ de données, 0 à 64 bits (informations).
- le champ de contrôle CRC (*Cyclic Redundancy Code*), 16 bits
- le champ d'acquittement 2 bits.
- le champ de fin de trame EOF (*End Of Frame*), 7 bits récessifs.



**Figure 8 :** trame de Donnée

- **SOF (Start Of Frame) :** Commence toujours par un bit dominant (bit à 0), la ligne étant précédemment au repos. Ce bit ne sert qu'à synchroniser les horloges internes des récepteurs sur celle de l'émetteur : bit de start.
- **Champ d'arbitrage :** composé de 12 bits, les 11 premiers indiquent l'identité du contenu du message, et servent également à l'arbitrage (gestion des priorités), le dernier bit (RTR : Remote transmission Request bit) permet de coder la nature du message :
  - bit à 0 (dominant) : trame de données.
  - bit à 1 (récessif) : trame de requête.
- **Champ de commande :** Champ de commande constitué de 6 bits :
  - Le premier bit IDE (Identifiant Extension bit) est un bit dominant permettant de spécifier qu'il s'agit d'une trame standard.
  - Le deuxième bit est réservé et définit comme dominant.
  - Les 4 derniers bits permettent de coder le nombre d'octets du champ de données.
- **Champ de données :** Ce sont les données transmises par la trame de données. Il peut contenir de 0 à 8 octets (64 bits), ou chaque octet est transmis avec le bit de poids fort en premier.

- **Champ de contrôle** (Cyclic redundancy code) : Le champ CRC est composé de la séquence de 15 bits suivi d'un délimiteur (1bit récessif). La séquence de CRC permet de vérifier l'intégrité des données transmises. Les bits utilisés dans le calcul du CRC sont ceux du SOF, de champ d'arbitrage, du champ Control et du champ de données.
- **Champ d'acquittement** : Ce champ d'acquittement est composé de 2 bits :
  - Un bit d'acquittement (l'ACK Slot) à l'état 0 (dominant) si le calcul du code de vérification des données est correct, sinon en cas d'erreur le bit est à l'état 1 (récessif).
  - Un bit délimiteur d'acquittement, est toujours à l'état haut 1 (récessif)
- **Champ fin (end of frame)** : Chaque trame de données est terminée par une séquence de 7 bits récessifs.

## 2) trame de requête (Remote frame) :

Une trame de requête est constituée de la même manière qu'une trame de données sauf que le champ de données est vide. Dans le champ d'arbitrage, le bit de RTR est récessif. Par conséquent si deux nœuds émettent chacun une trame possédant le même identificateur (c'est à dire qu'un nœud émet une trame de données et l'autre une trame de requête), l'arbitrage sur le bit de RTR va donner la priorité à la trame de données.

## 3) trame de gestion d'erreur :

Les erreurs de transmission sont détectées par le décodage du champ de contrôle, et sont spécifiées au niveau du champ d'acquittement.

On peut dans le protocole du bus CAN détecter cinq types d'erreurs :

- **Erreur de bit**: Un nœud envoyant un bit sur le bus écoute en même temps les bits qu'il reçoit. Il considère comme une erreur de bit lorsque le bit envoyé est différent du bit reçu.
- **Erreur de contrôle** : Une erreur de CRC est détectée lorsque le CRC calculé par un récepteur est différent de la valeur du CRC contenu dans la trame.
- **Erreur de forme** : est détectée lorsqu'un bit qui devrait être à une certaine valeur est à une valeur différente (un délimiteur par exemple).
- **Erreur de Bit stuffing** : quand un transmetteur détecte 5 bits consécutifs de même valeur dans les bits à transmettre, il ajoute automatiquement un bit de valeur opposée.
- **Erreur d'acquittement** : Le transmetteur détecte une erreur d'acquittement lorsqu'il ne reçoit pas de bit dominant pendant l'ACK Slot.

## 4) Trame de surcharge :

La trame de surcharge indique aux autres nœuds qu'une station est surchargée. Elle est formée de deux champs :

- le Champ de surcharge avec six bits dominants,
- le délimiteur de surcharge avec huit bits récessifs.

## 5) espace entre trames (inter-frame space) :

Elle sépare les trames de données ou de requêtes entre elles. Il s'agit d'une suite de plusieurs bits récessifs.

### 3.3.4 Les Identifiants CAN (ID) :

Les trames de données transmises par un nœud sur le bus ne contiennent ni une quelconque adresse du nœud expéditeur ou du nœud destinataire. C'est plutôt le contenu du message, sa signification qui est précisée par un identificateur (ID). Chaque nœud recevant un message regarde si celui-ci est intéressant pour lui grâce à l'ID. Si c'est le cas, il le traite, sinon, il l'ignore.

Cet unique ID indique aussi la priorité des messages. Plus la valeur est faible, plus le message sera prioritaire. Si deux nœuds ou plus cherchent à avoir accès au bus en même temps, c'est celui de plus haute priorité qui gagne. Les messages de priorité inférieure seront automatiquement retransmis lorsque le bus sera libre.[14]

### 3.3.5 Fonctionnement détaillé de l'arbitrage :

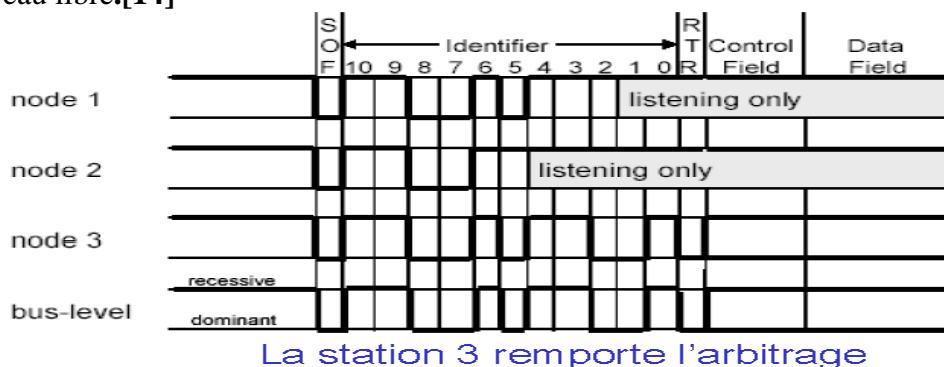
Dans un système typique, certains paramètres vont changer plus rapidement que d'autres. Ce sera par exemple la vitesse d'un moteur, tandis qu'un paramètre plus lent pourra être la Température de l'habitacle.

Il est donc naturel que les paramètres qui varient le plus soient transmis le plus souvent et par conséquent doivent avoir une plus grande priorité.

Dans les applications en temps réel, ceci nécessite non seulement une vitesse de transmission importante, mais aussi un mécanisme d'allocation du bus efficace qui soit capable de traiter les cas où plus d'un nœud cherchent à transmettre en même temps. Pour déterminer la priorité des messages, le CAN utilise la méthode CSMA/CD avec la capacité supplémentaire de l'arbitrage non destructif afin d'offrir une disponibilité maximale du bus.

La valeur de chaque ID, et donc la priorité de chaque type de messages, est assignée durant la conception du système. Un certain nombre de standards ont été développés selon les domaines d'utilisation du bus CAN pour fixer la priorité des ID, si plusieurs nœuds débutent leur transmission de trame en même temps, le premier qui présente un bit récessif (bit à 1) perd l'arbitrage dans le cas où un autre nœud présente un bit dominant (comme le montre la figure 16)

Tout ce passe donc comme si le message de plus haute priorité était le seul à être transmis. Lorsqu'un nœud perd l'arbitrage, il devient automatiquement un récepteur du message en cours de transmission, et il n'essaiera de retransmettre son message que lorsque le bus sera à nouveau libre.[14]



La station 3 remporte l'arbitrage

Figure 9 : fonctionnement d'arbitrage

### 3.4 le Diagnostics automobile (OBD) :

La norme OBD (On Board Diagnostic) a été mise en place au départ par la CARB (Californian Air Resources Board).son objectif est d'informer le conducteur d'un dysfonctionnement de son véhicule, au moyen d'un témoin lumineux placé sur le tableau de bord.

OBD permet la lecture des Diagnostic Trouble Codes standardisés, ou **DTC**, générés par le système informatique embarqué, ainsi que les informations temps réel en provenance des capteurs connectés aux calculateurs de bord.

En cas de dysfonctionnement de l'un de ces éléments, le système enregistre celui-ci sous forme de codes normalisés permettant au réparateur d'identifier le défaut.

La norme prévoit aussi l'effacement des DTC enregistrés par les calculateurs lorsque la maintenance a été effectuée [15-16].

Il existe plusieurs normes de l'OBD :

- **OBD I** (On Board Diagnostic) : Diagnostic embarqué première génération (avant 1996),
- **OBD II** : Deuxième génération, depuis 1996 (obligatoire aux Etats-Unis).
- **EOBD** (Européen On Board Diagnostic) : est la version Européenne de l'**OBD II**, depuis 2001 (pour les véhicules essence) et 2003 (Diesel).

#### 3.4.1 Les Protocoles OBD II :

Il existe cinq différents protocoles de communication disponibles sous la spécification OBD-II. Comme d'habitude, les fabricants ont tendance à mettre en avant leurs préférences d'où la variation des spécifications du protocole. Voici un bref aperçu de chacun des protocoles.

- **SAE J1850 PWM**: Ce protocole est généralement utilisé sur les véhicules Ford.
- **SAE J1850 VPW**: véhicules GM utilisent généralement cette version.
- **ISO 9141-2**: Pour les véhicules Chrysler européens, ou véhicules de l'Asie.
- **ISO 14230 KWP2000**: Dit protocole Keyword 2000, il est également utilisé sur Chrysler, véhicules européens, ou les véhicules asiatiques.
- **ISO 15765 CAN**: Ce protocole a été chargé dans tous les véhicules vendus aux États-Unis à partir de 2008 et plus tard. Toutefois, les véhicules européens peuvent avoir aussi un CAN.

### 3.4.2 Le connecteur OBD 2 :

Le connecteur obd2 permet l'accès au système de diagnostics embarqués, il possède 16 broches, dont leur utilisation est liée au protocole de communication utilisé par le constructeur comme le montre la figure 10.

- Protocoles PWM/VPW J1850 : utilise les broches 2 et 10
- Protocoles ISO 15765 (CAN) : utilise les broches 6 et 14
- Protocoles ISO 9141/14230 : utilise les broches 7 et 15

Les broches 1, 3, 8, 9, 11, 12, 13, 16 sont libres et les broches 4 et 5 représentent la masse.

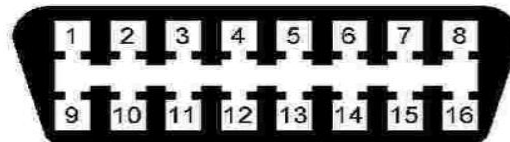


Figure 10 : connecteur OBD II

### 3.4.3 Les Modes de fonctionnement de L'OBD-II [17]:

Il y a dix modes de fonctionnement décrits dans la dernière spécification OBD-II. Ils sont comme suit:

**Mode1** : Ce mode retourne les valeurs courantes de certains capteurs tels que :

- le régime moteur
- la vitesse du véhicule
- les températures du moteur (air, liquide de refroidissement)

La norme OBD prévoit 135 paramètres pour le Mode 1.

**Mode 2** : Ce mode retourne les données instantanées d'un défaut.

**Mode 3** : Ce mode retourne les codes défauts enregistrés.

**Mode 4** : Ce mode permet d'effacer les codes défauts enregistrés

**Mode 5** : Ce mode retourne les résultats des autodiagnostic effectués sur les sondes à oxygène.

**Mode 6** : Ce mode retourne les résultats des autodiagnostic effectués sur les systèmes non soumis à la surveillance constante.

**Mode 7** : Ce mode retourne les codes défauts non confirmés (utile après une réparation pour vérifier que le code défaut ne réapparaît pas).

**Mode 8** : Ce mode retourne les résultats des autodiagnostic effectués sur d'autres systèmes.

**Mode 9** : Ce mode retourne les informations concernant le véhicule tel que:

- le VIN numéro d'identification du véhicule.

**Mode 10 :** Ce mode retourne les codes défauts permanents. Les codes utilisés sont identiques à ceux du mode 3 et 7. Contrairement aux modes 3 et 7, ces codes ne peuvent pas être effacés à l'aide du mode 4. Seuls plusieurs cycles routiers sans apparition du problème effaceront les défauts.

### 3.4.4 Le principe de fonctionnement des PID :

La norme OBDII prévoit plusieurs PID (Parameter Identifier) pour chaque mode et chaque PID correspond à un paramètre, Par exemple la norme OBDII prévoit le régime moteur pour PID 0x12, toute fois Un fabricant peut personnaliser un PID particulier qui n'est pas défini dans la norme OBD-II.

#### 3.4.4.1 Les PIDs standard :

Un petit aperçu dans le tableau 3 montre les PID OBD-II standards tels que définis par la SAE J1979. La réponse attendue pour chaque PID est donnée, ainsi que des informations sur la façon de traduire la réponse en données significatives. Tous les véhicules prennent en charge tous les PIDs.

Les modes 1 et 2 sont essentiellement identiques, sauf que le mode 1 fournit des informations en cours, alors que le mode 2 donne un aperçu des mêmes données prises au moment où le dernier code de diagnostic a été établi.

Mode (hex)	PID (hex)	Octets de données retournées	Description	Valeur min	Valeur max	unités	Formule
01	00	4	PIDs supportés [01-20]				Bit codé [A7..D0] == [PID 0x01..PID 0x20]
01	04	1	La valeur de la charge du moteur calculée	0	100	%	$A * 100/255$
01	0C	2	Régime moteur	0	16,383.75	Rpm	$((A * 256) + B) / 4$
01	0D	1	Vitesse du véhicule	0	255	Km/h	A

**Tableau 3 :** Présentation de quelque PID

#### Exemple Mode 0x01 PID 00:

Une demande de ce PID renvoie 4 octets de données. Chaque bit, de MSB à LSB, représente l'un des 32 prochaines PIDs et indique les PIDs prise en charge.

Par exemple, si la réponse du véhicule est BE1FA813 il peut être décodé comme suit:

Hex	B				E				1				F			
<b>Binaire</b>	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1
<b>PID Supporter</b>	Oui	Non	Oui	Oui	Oui	Oui	Oui	Non	Non	Non	Non	Oui	Oui	Oui	Oui	Oui
<b>N° PID</b>	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10

A				8				1				3			
1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	1
Oui	Non	Oui	Non	Oui	Non	Non	Non	Non	Non	Non	Oui	Non	Non	Oui	Oui
11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20

**Tableau 4:** Décodage de trame de réponse en mode 01 avec pid 0x00

Donc, PIDs pris en charge sont:

01, 03, 04, 05, 06, 07, 0C, 0D, 0E, 0F, 10, 11, 13, 15, 1C, 1F et 20

#### 4. Conclusion

Nous arrivons à présent à la fin de ce premier chapitre dans lequel nous avons abordé les accidents de route tout en présentant les causes et les conséquences ainsi quelque statistique sur les accidents en Algérie, puis nous avons parlé sur l'électronique automobile et ses divers périphériques, et nous avons présenté le bus CAN puis on a terminé par l'étude de diagnostic automobile qui est l'OBD.

Cet étude nous a permet de comprendre quelques technologies automobiles qui serviront d'aide à réaliser notre système.

Avant de savoir quelles solutions technologiques apportées, il faut d'abord identifier les besoins réels du système étudié et ses possibles évolutions, pour cela nous avons jugés utile de présenter la communication sans fil auxquelles nous auront à faire, cela sera l'objet du prochain chapitre.

# Chapitre II : La connectivité automobile

### 1. Introduction :

Les objets qui nous entourent sont dotés d'une capacité à communiquer, traiter des informations, prendre des décisions, envoyer des données et des commandes. Ces fonctionnalités rendent ces objets « intelligents » dans le sens où ils sont capables d'opérer de manière autonome.

L'internet des objets repose sur des communications entre machines (M2M), sans intervention humaine dans ces échanges. Pour ce faire, ces objets utiliseront des communications sans fil (ZigBee, Bluetooth, RFID etc....).elle a pour but de créer un réseau d'objets interconnectés tels que les Smartphones, les télévisions ou les voitures [18].

Dans ce chapitre nous allons présenter la connectivité automobile, en précisant ses domaines et les protocoles de communication sans fil qu'elle utilise, puis nous allons faire un comparatif entre ces derniers, ce qui nous permettra de choisir le protocole qui convient à notre projet, en terminant par une présentation de la norme ZigBee et du module Xbee.

### 2. La connectivité automobile :

Les technologies numériques appliquées à l'automobile avaient pour principal objectif de réduire le nombre d'éléments mécaniques. Ainsi, il a été possible de réduire considérablement le poids des véhicules et de réaliser des économies non négligeables. Plusieurs autres applications sont en cours d'expérimentation, notamment l'amélioration de l'expérience de conduite, la protection des passagers et la connectivité avec le monde extérieur.[19]

#### 2.1 La voiture connectée :

Ce type de véhicule intègre des systèmes prenant en charge une large gamme de technologies de communication, permettant de résoudre toutes sortes de défis relatifs au confort et à la protection du conducteur et de ses passagers, ainsi qu'à la sécurité du véhicule. Par ailleurs, les opérations de maintenance et réparation ont été considérablement simplifiées et peuvent désormais être anticipées dans certains cas.

#### 2.2 Domaine de la connectivité automobile :

Les domaines concernés par la connectivité sans fil automobile sont :

- **La Télématic (la télécommunication et l'informatique):** la télématic couvre un large espace de tâches devant être exécutées par le véhicule. La pression des pneus est déjà transmise sans fil (généralement avec Zigbee), ainsi que les données des systèmes « sans clés ». Tous ces éléments permettent de réduire au maximum le coût et le poids. À l'avenir, cette technologie ne se limitera pas au simple confort du conducteur et de ses passagers. L'accès aux données télématic permet de suivre les déplacements des flottes de véhicules et de limiter les vols.[20] D'autre part, les compagnies d'assurance disposeront également d'outils plus précis pour évaluer les circonstances des accidents. Il sera possible de transférer des informations de diagnostic embarqué, de sorte que l'entretien sera facilité.

- **La technologie ADAS (Système avancé d'assistance à la conduite) :** qui est intégrée dans de nombreux véhicules est la première étape d'un processus menant à des véhicules totalement autonomes. Les radars embarqués servent déjà à prévenir les collisions et des dispositifs de changement de voie commencent à entrer en service. Avec les communications V2V (Vehicle-to-Vehicle) et V2I (Vehicle-to-Infrastructure), des données précieuses pourront bientôt être transférées sans fil entre des véhicules relativement proches (ainsi qu'avec les feux de signalisation, les panneaux de limitation, etc.) pour anticiper les dangers de la route, les embouteillages et autres problèmes de ce type. [19]
- **divertissement et communications :** la connectivité dans les voitures, permettra aux smartphones de se connecter avec le véhicule, d'où la facilité de partager des documents multimédias. De même, les autres appareils portables électroniques (tels que les tablettes) seront utilisés par les passagers à l'arrière pour se connecter à Internet.

### 2.3 Les protocoles sans fil automobile :

D'ici 2020, plus de 250 millions de voitures connectées seront en circulation. Ces véhicules utiliseront différents protocoles de communication sans fil, notamment WLAN, LTE, Bluetooth et ZigBee (voir la Figure 11 pour plus de détails). [21]



**Figure 11:** Les protocoles de communication sans fils automobile

#### ❖ Le Bluetooth :

Le Bluetooth est une nouvelle technologie de transmission sans fil. Son but est de permettre la communication à courte distance entre plusieurs appareils, et sans le moindre câble, en utilisant les ondes radio , Elle peut remplacer par exemple les câbles entre ordinateurs, tablettes, imprimantes, scanners, claviers, souris etc.....

❖ **Le wifi :**

Le Wifi est une technologie de réseau informatique sans fil mise en place pour fonctionner en réseau interne, elle est devenue un moyen d'accès à haut débit à Internet. Il est basé sur la norme IEEE 802.11.

❖ **Le ZigBee :**

Cette technologie a pour but la communication de courte Distance telle que le propose déjà la technologie Bluetooth, tout en étant moins chère et plus simple.

**2.4 Comparaison entre les protocoles sans fil automobile :**

Le tableau suivant montre un comparatif entre les trois protocoles sans fils généralement utilisé dans l'automobile moderne

Protocole	Zigbee	Bluetooth	Wi-Fi
IEEE	802.15.4	802.15.1	802.11a/b/g
Besoins mémoire	4-32 Kb	250 Kb +	1 Mb +
Durée de vie	Années	Jours	Heures
Nombre de nœuds	65 000+	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11-54 Mb/s
Portée	100 m	10 m	100 m

**Tableau 5:** Comparaison entre les différents protocoles de communication sans fil

Toutes les caractéristiques du protocole ZigBee sont bien adaptées aux systèmes embarqués. En effet, le protocole ZigBee se distingue des autres protocoles par ses faibles besoins en mémoire et en énergie, ce qu'est favorable pour son implémentation. De plus, il présente une durée de vie très importante qu'est de l'ordre de plusieurs années, ainsi qu'un très large nombre de nœuds à supporter dans son réseau. Enfin, ce protocole convient parfaitement aux applications nécessitant une faible vitesse de transfert de l'ordre de 250 Kb/s.

**3. le protocole zigbee :**

ZigBee est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur le standard IEEE 802.15.4 pour les réseaux à dimension Personnel (Wireless Personal Area Networks: WPANs).

On retrouve ce protocole dans des environnements embarqués où la consommation est un critère de sélection. Les nœuds Zigbee sont conçus pour fonctionner plusieurs mois (jusqu'à deux ans pour les moins Consommant) en autonomie complète grâce à une simple pile alcaline de 1,5V.

### 3.1 La norme IEEE 802.15.4 :

Un PAN (Personal Area Network) est composé d'un ensemble de nœuds ayant pour rôle de collecter et de transmettre l'information vers un organe central du réseau, qui sera en charge de traiter les informations ou de jouer le rôle de passerelle. Ce dernier a pour nom « puits » ou « coordinateur ».

La norme IEEE 802.15.4 décrit les couches basses d'un réseau PAN, elle propose une couche physique et une couche liaison de données adaptées aux applications à faible débit dont l'autonomie énergétique est une contrainte forte. [22]

IEEE 802.15.4 définit deux types d'entités pouvant participer à un réseau: les RFD (Reduced Function Devices) et les FFD (Full Function Devices) :

- Les RFD sont des nœuds aux fonctionnalités réduites, très économes en énergie, et pouvant être connectés à des capteurs ou à des actionneurs, il est considéré comme un dispositif d'extrémité (end device).
- Les FFD sont des nœuds ayant la capacité de router ou de coordonner le réseau. Dans chaque réseau IEEE 802.15.4, l'un des FFD est le coordinateur du réseau. Le rôle du coordinateur est entre autres d'initier la formation de la topologie du réseau.

### 3.2 Les couches du protocole ZigBee :

Le protocole ZigBee peut être vu comme trois couches séparées

- **Couche Application :** Contrôler l'application qui tourne dans les équipements ZigBee  
Par exemple : capture de température
- **Couche de la pile ZigBee :** Les protocoles qui vérifient le routage des données à la bonne destination tout en assurant la sécurité de transmission
- **Le standard IEEE 802.15.4 :**
  - La sous-couche MAC (*Medium Access Control*) : responsable de l'adressage (où les données vont être transmises et d'où les données sont reçues)
  - La sous-couche PHY (*Physical layer*) : responsable de la transmission radio

### 3.3 Les équipements ZigBee :

Il existe 3 équipements zigbee :

- **Le coordinateur du réseau ZigBee :** Ce type d'équipement unique dans un réseau assure les fonctions telles que l'authentification, l'initiation de la communication, la sécurité et l'ajout des nœuds au réseau etc... Il doit être actif en permanence pour répondre à tout moment aux requêtes des autres éléments du réseau. Il est donc alimenté à plein temps.
- **Le routeur ZigBee :** Ces nœuds permettent la transmission de messages. Ils sont indispensables pour étendre le réseau par acheminement des trames d'un nœud à un autre. Ils permettent aussi aux autres modules de s'enregistrer sur le même réseau.

- **Le terminal ZigBee :** Les terminaux sont utilisés pour communiquer avec le monde extérieur, par exemple, les terminaux peuvent être connectés à des capteurs, des écrans etc. Ils peuvent se mettre en veille pour réduire d'avantage leurs consommations d'énergie. les terminaux ne peuvent pas communiquer directement avec les autres terminaux, mais ils communiquent à travers les routeurs ou le coordinateur du réseau.

### 3.4 Les topologies du réseau ZigBee

Il existe quatre types de topologies zigbee comme le montre la figure 12 :

- **La topologie Point à point :** chaque nœud pourra communiquer avec n'importe quel autre nœud du réseau grâce à la collaboration des nœuds intermédiaires sollicités afin de relayer les paquets jusqu'à la destination.
- **La topologie étoile :** Cette topologie impose que les communications s'établissent directement et uniquement entre le coordinateur et les autres nœuds
- **La topologie en arbre :** Dans une topologie en arbre, le coordinateur est connecté soit à des routeurs soit à des nœuds terminaux.
- **La topologie en maille :** est une modification de la topologie en arbre, les nœuds sont interconnectés avec d'autres nœuds de sorte que de multiples voies permettent de connecter chaque nœud. Les connexions entre les nœuds sont mises à jour dynamiquement et optimisées par une table de routage intégrée dans la maille.

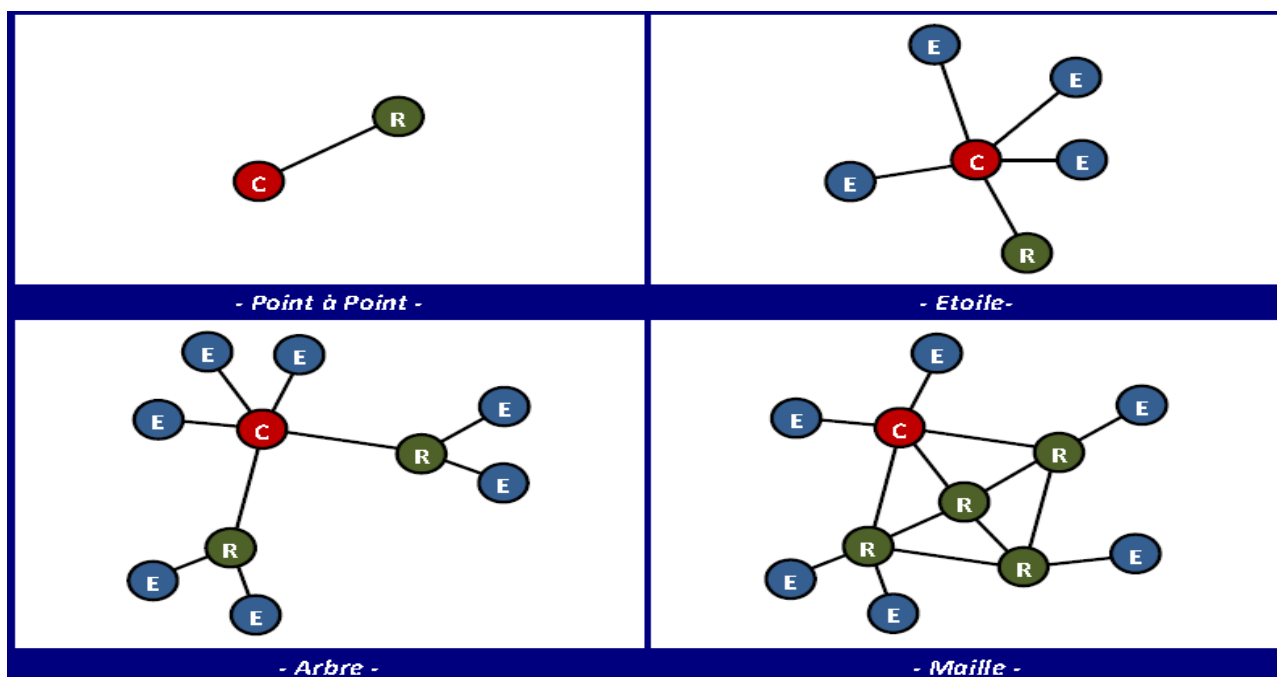


Figure 12: Topologie Zigbee

#### 4. Interface ZigBee (Le module XBee) :

##### 4.1 Présentation :

Les modules de communication Xbee™ sont des modems radiofréquences utilisant le Protocole de communication Zigbee. Fabriqués par Digi International, ils ont été Certifiés par la communauté industrielle ZigBee Alliance en 2006.

##### 4.2 Différentes versions des modules Xbee :

Il existe trois types de modules Xbee :

- **Xbee S1** (série 1).
- **Xbee S2** (série 2).
- **Xbee Pro.**

Bien que la consommation ou la portée soient sensiblement les mêmes, les modules S1 et S2 ne sont pas compatibles entre eux. D'autre part, les modules S1 ne permettent que les topologies réseau « point à point » et « étoile ».

Totalement compatibles avec les modules S2, les modules « pro » offrent des meilleures performances notamment la portée mais au détriment de la consommation.

Les Xbee Pro ont une portée plus étendue que les Xbee simple avec une même vitesse de transmission comme le montre le tableau 6.

Performances	Xbee	Xbee Pro
Porté intérieur / milieu urbain	Jusqu'à 30 m	Jusqu'à 100 m
Porté extérieur champ libre	Jusqu'à 100 m	Jusqu'à 1,6 Km
Vitesse de transmission	250 kbps	250 kbps

**Tableau 6** : Comparatif entre Xbee et Xbee Pro

##### 4.3 Mode de fonctionnement des modules Xbee :

Les modules Xbee peuvent être configurés pour répondre à certaines besoin de fonctionnement :

- I. Mode transparent** : Il s'agit du mode par défaut utilisé par les modules Xbee. Toute donnée reçue sur la broche RX du module Xbee est automatiquement transmise par radio.
- II. Mode API (Application Programming Interface)** : ce mode permet d'accéder à toutes les possibilités de mise en réseau des modules.
- III. Mode command** : ce mode permet de configurer le module, ses entrées, ses sorties, son adresse, l'adresse de destination de ses messages, etc.

#### 4.4 Paramétrage des modules XBEE :

Les modules Xbee peuvent être utilisés en mode transparent sans aucune modification de leur configuration. Cependant pour modifier certains paramètres (canal radio utilisé, numéro de réseau, vitesse du port série...).

Pour configurer le module Xbee, il faut lancer le logiciel X-CTU qui va lire tous les paramètres du XBee.

Les paramètres importants sont :

- **PAN ID (Personal Area Network)** : Identifiant du réseau personnel. Cet identifiant doit être le Même pour les modules XBee qui doivent appartenir au même réseau.
- **SH (Serial Number High)** : Bits de poids fort (32 bits) du numéro de série du module XBee.
- **SL (Serial Number Low)** : Bits de poids faible (32 bits) du numéro de série du module Xbee.
- **DH (Destination Address High)** : Bits de poids fort du numéro de série du module XBee avec Lequel la communication est faite.
- **DL (Destination Address Low)** : Bits de poids faible du numéro de série du module XBee avec lequel la communication est faite.
- **BD (Baud Rate)** : Vitesse de transmission en bit/s.

#### 4.5 Adressage des modules XBEE :

Il y a 2 types d'adressage possible :

- **Adressage réseau de 16 bits** : une adresse de 16 bits est assignée à un nœud lorsqu'il rejoint le réseau. Cette adresse est unique mais elle n'est pas statique.
- **Adressage réseau de 64 bits** : Chaque nœud contient une adresse unique de 64 bits qui l'identifie.

#### 4.6 Les modes de communication :

##### MODE UNICAST:

Dans ce mode de fonctionnement, le module récepteur, envoie un "ACK" à celui qui a émis le paquet de données. Si l'émetteur ne reçoit pas ce "ACK", il renvoie jusqu'à 3 fois ce paquet.

##### MODE BROADCAST:

Dans ce cas il n'y a pas de "ACK", envoyé par le récepteur, ni de répétition d'envoi par l'émetteur.

Tous les modules reçoivent et acceptent les paquets de données.

Pour envoyer des données sans tenir compte de l'adresse destinataire sur 16 ou 64 bits, il faut positionner l'adresse destinataire: DH = 00 00 00 00 et DL = 00 00 FF FF.

### **5. Conclusion :**

Dans ce chapitre nous avons introduit la notion des objets connectés en se basant sur la connectivité automobile puis nous avons présenté le protocole sans fil zigbee.

Les différents concepts traités dans les deux premiers chapitres nous aideront à mieux comprendre notre mode d'opération et les notions fondamentales pour bien réaliser notre application.

Dans le prochain chapitre nous allons présenter la conception de notre application en spécifiant son objectif et les moyens utilisés pour sa réalisation.

# Chapitre III : Moyen et conception

### **1. Introduction :**

La conception de toute solution logicielle doit être traitée avec précision et détail, précédée d'une analyse profonde et bien réfléchie, car elle est le reflet du futur système avant même sa concrétisation.

Dans le but d'avoir une meilleure analyse et de rendre la conception du logiciel plus complète nous allons étudier tous les outils et le matériel qui nous serviront à la réalisation de notre application. Puis nous allons passer à la phase conception.

### **2. Objectifs du système :**

Réalisation d'un système qui permet de récupérer des données en temps réel à partir d'un véhicule (Vitesse, régime moteur, taux d'alcool) et faire des vérifications sur le dépassement illégal des limites réglementaire, chaque dépassement engendra une génération de rapport construit de l'état de la conduite puis ce rapport sera transmis vers une borne de police la plus proche. Cette dernière collecte des rapports en provenance des véhicules et elle permet leur ajout à sa base de données.

### **3. Fonctionnalité du système :**

Le système peut être divisé en deux parties.

#### **Première partie : (Côté véhicule)**

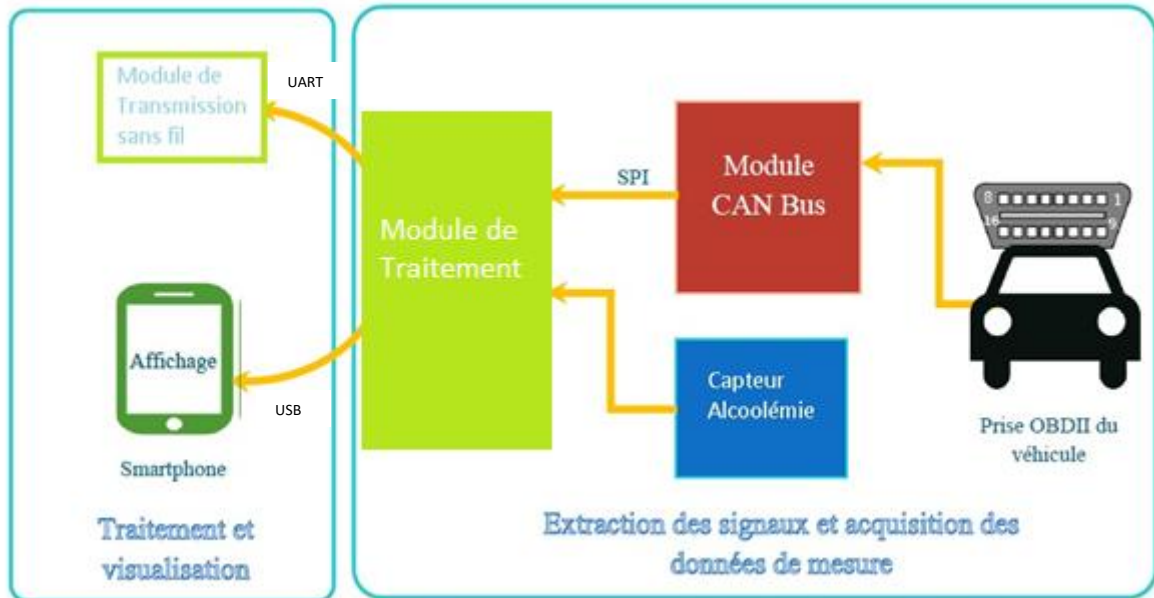
- l'extraction de plusieurs signaux à partir d'un véhicule qui utilise le protocole CAN, les signaux seront ensuite transmis vers un module de traitement.
- extraction de valeur de taux d'alcoolémie à l'aide d'un capteur, les valeurs seront transmises ensuite vers le module de traitement.
- affichage des valeurs extraites en temps réel dans une application Android.
- vérification de la conformité des valeurs acquises au règlement routier et en cas de dépassement il y aura une construction ou une mise à jour de rapport.
- le rapport doit contenir plusieurs informations sur le véhicule tel que : immatriculation, taux d'alcoolémie, vitesse (moyenne et maximale), régime moteur , durée de dépassement.
- détection d'un système embarqué policier (borne de police ) , et transmission du rapport à l'aide d'un module de communication sans fil vers ce système .

#### **Deuxième partie : (Côté borne de Police)**

- informer tous les véhicules de sa présence.
- réception de rapport en cas de dépassement réglementaire.
- affichage du rapport dans une application Android.
- sauvegarde du rapport dans une base de données.
- recherche et suppression des rapports existants dans la base de données.

#### 4. Structure du système

La figure suivante montre la structure globale du système à réaliser :



#### Système embarqués dans le véhicule



#### Système embarqués dans la borne de police

**Figure 13:** Structure générale du système

#### 4.1 Composant requis :

Avant d'entrer dans la conception du système, il faut faire le point sur les composants que nous allons utiliser.

D'après la structure du système, il exigera quatre types de modules :

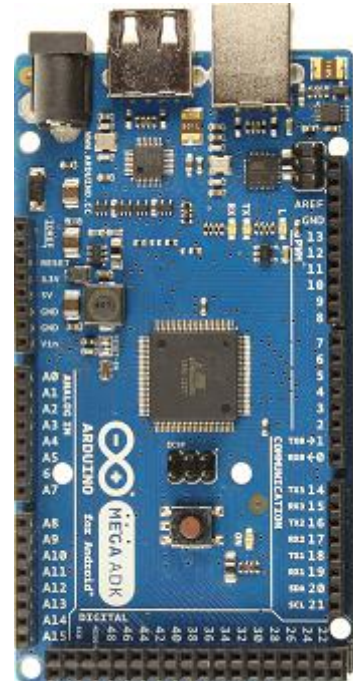
- **Module de traitement** : qui sera la carte ARDUINO MEGA ADK aura pour fonction d'exécuter le programme principal dont chaque composant a son programme et on a opté pour ce module car :
  - possède plusieurs broches dont USB qui sont nécessaires pour notre projet.
  - faciliter de brochage et compatibilité avec les autres modules.
  - espace mémoire considérable pour notre programme.
  - langage de programmation facile.
- **Module de communication** : après l'étude faite dans le chapitre deux on a constaté que le protocole zigbee convient mieux à être notre choix pour le module de communication grâce à son interface XBEE pro s1.
- **Module d'affichage** : l'aspect affichage sera présenté sous forme d'application android, ce qui nécessite des équipements android (Smartphone, tablette).
- **Module d'acquisition** :
  - Le CAN bus shield permet d'interfacer le bus CAN d'un véhicule afin d'extraire des paramètres souhaités, puis il transmet ces données vers le module de traitement.
  - Le MQ135 est un capteur sensible à l'alcoolémie, il permet la récupération le taux d'alcoolémie dans le véhicule.

## 4.2 Module de traitement :

### 4.2.1 Présentation de la carte Arduino Mega ADK:

L'Arduino MEGA ADK est une carte électronique basée sur le microcontrôleur ATmega2560, Elle dispose d'une interface hôte USB pour connecter avec des téléphones Android. Elle dispose de 54 broches numériques d'entrée / sortie, 16 entrées analogiques, 4UART, une prise d'alimentation, et un bouton de remise à zéro.

La principale différence entre une carte Arduino Mega ADK et une carte Arduino Mega2560 est que l'ADK propose en plus une interface USB compatible avec les téléphones Android. [23]



### 4.2.2 Alimentation de la carte Arduino Mega ADK:

Figure 14: La carte Arduino Mega ADK

La carte Arduino MEGA ADK peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée.

Les broches d'alimentation sont les suivantes :

- VIN : La tension d'entrée à la carte Arduino quand elle est alimentée par une source externe.
- 5V : Cette broche délivre un 5V régulé par le régulateur de la carte.
- 3V3 : Une alimentation de 3,3 volts.
- GND : La masse.
- IOREF : Cette broche fournit la référence de tension avec laquelle le microcontrôleur fonctionne.

### 4.2.3 Gestion mémoires dans la carte :

La carte Arduino MEGA ADK dispose de 256 Ko de mémoire flash pour le stockage de code, 8 Ko de SRAM et 4 Ko de mémoire EEPROM.

### 4.2.4 Les broches de la carte Arduino Mega ADK:

- **54 broches numériques** : Chacune des 54 broches numériques sur la carte Arduino MEGA ADK peut être utilisée comme une entrée ou une sortie.
- **4 UARTS (RX TX)** sur les broches : [0 1], [19 18], [17 16], [15 14].

- **Interruptions externes:** sur les broches : 2 ,3 ,18 ,19 ,20 ,21.
- **PWM:** sur les broches 2 à 13 et 44 à 46.
- **SPI:** sur les broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)
- **USB Host:** communique avec l'Arduino via le bus SPI. Donc, il utilise les broches suivantes : 7 (RST), 50 (MISO), 51 (MOSI), 52 (SCK).
- **LED: 13.** Il est équipé d'une LED connectée à la broche numérique 13.
- **16 entrées analogiques.**
- **AREF.** Tension de référence pour les entrées analogiques.
- **Réinitialiser.** La mise de la ligne à LOW permet de réinitialiser le microcontrôleur.

#### 4.3 Module d'acquisition :

##### 4.3.1 Bus CAN Shield:

**4.3.1.1 Présentation :** Ce composant permet de connecter le bus CAN du véhicule à la carte de développement, il utilise le microcontrôleur MCP2515. Il interroge les différents capteurs liés au réseau Can pour obtenir des informations, il peut également stocker des données ou d'interfacer un écran. [24]

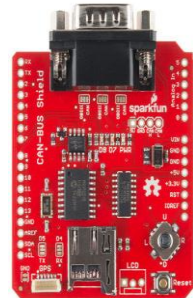


Figure 15 : Sparkfun Can Bus Shield

##### 4.3.1.2 Caractéristiques :

- CAN v2.0B jusqu'à 1 Mb / s
- Interface SPI à vitesse haute
- Connexion CAN via le connecteur sub-D d'une manière standard.
- L'alimentation peut se fournir à la carte par sub-D
- Socket pour le module GPS EM406.
- Titulaire de la carte Micro SD.
- Connecteur pour LCD de série.
- Bouton de réinitialisation.
- Deux indicateurs LED.

##### 4.3.1.3 Méthodes de communication:

En raison de toutes les différentes fonctionnalités matérielles disponibles sur le bus can shield, il existe différentes méthodes de communication.

- **SPI :** Le MCP2515 et la microSD peuvent communiquer avec l'Arduino via les lignes SPI.
- **Entrées analogiques :** Le joystick est connecté aux broches A1-A5 sur l'Arduino.
- **port séries :** L'écran LCD et GPS peuvent communiquer avec l'Arduino à l'aide des ports série.

**4.3.1.4 Le microcontrôleur MCP2515 :**

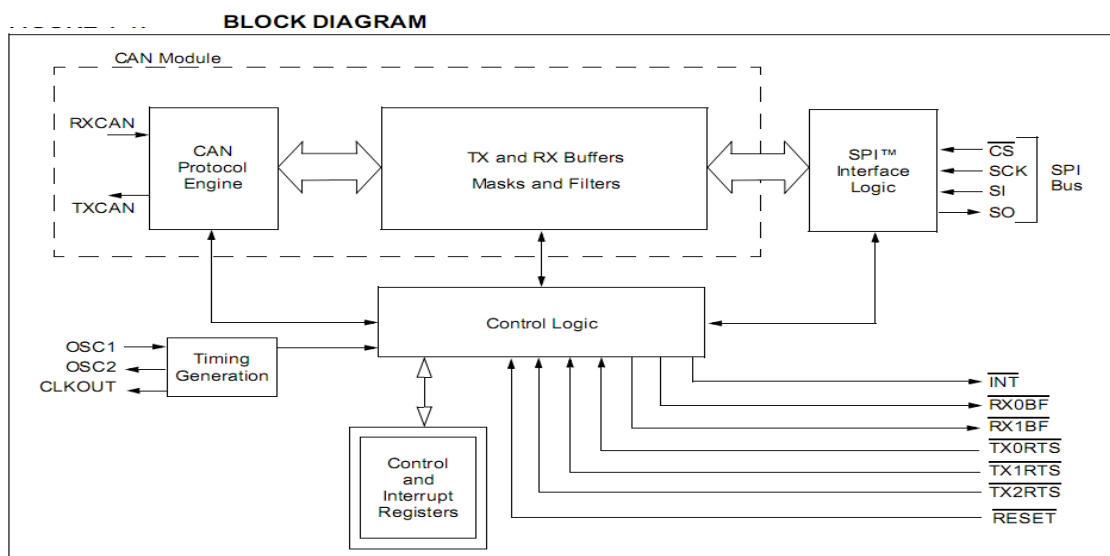
Ce circuit intégré met en œuvre la spécification CAN en version B. Cela signifie que le contrôleur est capable d'émettre et de recevoir les trames de données standards et étendues. Cette spécification est capable de communiquer jusqu'à 1Mb / s (cela dépend de la longueur du bus). le MCP2515 est capable de communiquer avec microprocesseur via l'interface SPI. Le MCP2515 à deux masques et six filtres d'acceptation qui sont utilisés pour filtrer les messages indésirables. [25]

**Modes de fonctionnement du MCP2515 :** le MCP2515 définit cinq modes comme le montre le tableau suivant :

Mode	La description
Configuration	Ce mode permet au développeur de mettre à jour les paramètres de la mcp2515 (configuration des masques et filtres, registre de contrôle)
Ordinaire	Le mode de fonctionnement standard. Tous les filtres et les masques sont appliqués.
En veille	Mode de faible puissance. SPI reste actif et le MCP2515 se réveille s'il y a une activité sur le bus CAN
Ecoute	Dans ce mode le MPC1525 reçoit tous les messages qui circulent sur le bus CAN, il permet de détecter la vitesse de transmission de bus.
Bouclage	Tous les messages envoyés sont filtrés puis retournés par les buffers de réception. utilisées dans le développement et les tests.

**Tableau 7:** Modes de fonctionnement du MCP2515

**L'architecture de mcp2515 :** Le contrôleur MCP2515 se compose de trois parties principales. Chaque partie couvre les différentes fonctions de ce contrôleur CAN comme le montre la figure suivante :



**Figure 16 :** Block Diagramme du micro contrôleur MCP2515

### **La transmission et réception de message par le MPC2515 :**

Le MCP2515 à trois buffers d'émission et deux buffers de réception, et deux masques d'acceptation et un total de six filtres de réception.

#### **A. Le buffer de transmission :**

Le MCP2515 implémente trois tampons d'émission (txb0 txb1txb2 ). Chacun de ces tampons occupe 14 octet de mémoire SRAM.

- **Le premier octet (TX nctrl) :** est un registre de contrôle associé au tampon de message. Les informations contenues dans ce registre détermine les conditions dans lesquelles le message sera transmis et indique l'état de la transmission du message.
- **Cinq octets (les registres d'identification TXBnSIDH, XBnSIDL, TXBnEID8, TXBnEID0, TXBnDLC) :** Sont utilisés pour maintenir la norme standard et étendue des identifiants CAN.
- **Les huit derniers octets (les registres de données TXBnD0-TXBnD7) :** contient les huit octets de données possibles du message à transmettre

#### **B. les Buffer de réceptions :**

Le MCP2515 comprend deux tampons de réception (RXB0 RXB1) avec des filtres d'acceptation multiples pour chacun. Il y a aussi un buffer d'assemblage séparé (MAB) qui agit comme un troisième tampon de réception.

Le MAB est toujours engagé à la réception du message suivant du bus. Le MAB rassemble tous les messages reçus. Ces messages seront transférés vers les tampons RXBn si l'acceptation critères de filtrage est satisfaite.

Les buffers de réception sont organisés de la même manière que pour le buffer démission au lieu décrire il faut charger tous les registres.

Lorsque le message est reçu, un front descendant apparaît sur la broche INT qui génère une interruption externe au microcontrôleur.

#### **C. Les filtres d'acceptation des messages reçus :**

Les filtres d'acceptation et les masques sont utilisés pour déterminer si un message dans le tampon de message d'assemblage devrait être chargé dans l'un des tampons de réception. Une fois qu'un message valide est reçu dans le MAB, les champs du message d'identification sont comparés aux valeurs de filtre. S'il y a une correspondance, ce message sera chargé dans le tampon de réception appropriée.

Les masques filtrants sont utilisés pour déterminer quels bits dans l'identifiant sont examinés avec les filtres. Une table de vérité est indiquée dans le tableau 7 indiquant comment chaque bit du l'identificateur est comparé aux masques et aux filtres afin de déterminer si le message doit être chargé dans un tampon de reception. Le masque détermine

essentiellement les bits aux quel les filtres d'acceptation seront appliqués. Si un bit de masque est réglé sur un zéro, ce bit sera automatiquement accepté, quel que soit le bit de filtre.

**TABLE 4-2: FILTER/MASK TRUTH TABLE**

Mask Bit n	Filter Bit n	Message Identifier bit	Accept or Reject bit n
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

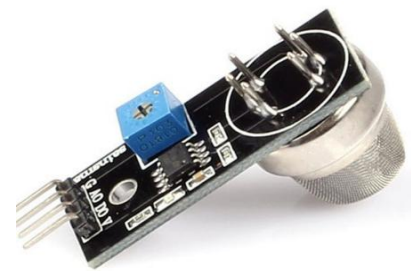
**Tableau 8:** Les filtres et masque d'acceptation du

### 4.3.2 Capteur d'alcoolémie MQ 135 :

#### 4.3.2.1 Présentation de MQ 135 :

Le MQ135 est un capteur qui permet de mesurer la qualité de l'air. Le MQ135 est sensible aux principaux polluants présents dans l'atmosphère. Ce capteur est sensible au CO<sub>2</sub>, à l'alcool, au Benzène, à l'oxyde d'azote (NO<sub>x</sub>) et à l'ammoniac (NH<sub>3</sub>).

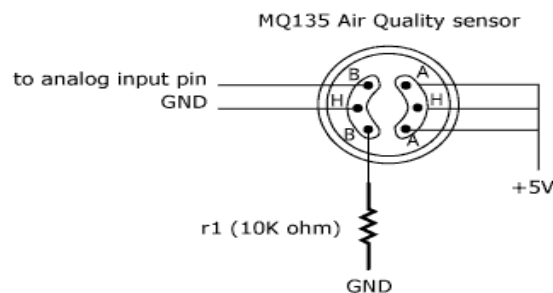
Ce capteur utilise le dioxyde d'étain (SnO<sub>2</sub>) dont la conductivité électrique varie en fonction de la présence de polluant(s). Comme tous les capteurs de la série MQ, il faudra procéder à une calibration dans une atmosphère de référence pour mesurer la présence d'un polluant en particulier. [26]



**Figure 17 :** Capteur d'alcoolémie MQ 135

#### 4.3.2.2 Le circuit :

La figure 18 montre le circuit du capteur MQ 135



**Figure 18 :** Le circuit du capteur MQ135

**4.3.2.3 Calibration :** Le calibrage consiste à étalonner le capteur en l'exposant à un gaz d'alcool connu de 0.4mg / L.

**4.3.2.4 Calcul de taux d'alcool dans le souffle :**

- Il existe une formule simple utilisée pour calculer l'alcoolémie (BAC) qui est :

$$\text{BAC} = \text{souffle mg / L} * 0,21$$

**Fonctionnement :**

Le capteur MQ 135 capte la valeur de l'alcool et l'envoie à l'Arduino MEGA ADK par la branche A0, après avoir reçu cette valeur analogique elle sera convertis par une formule dans le programme.

**4.3Module de Communication sans fil:**

**4.4.1 Présentation de Xbee Pro Séries 1 :**

Proposés depuis quelques années par la société Digi International, les modules Xbee sont des modems radio très élaborés, fonctionnant sur la bande ISM (Industrie Science et Médical), c'est-à-dire sur une fréquence de 2,4 GHz. Les modules Xbee vont au-delà du simple modem puisqu'ils permettent de constituer de véritables réseaux sans fil et supportent de ce fait la notion d'adressage.



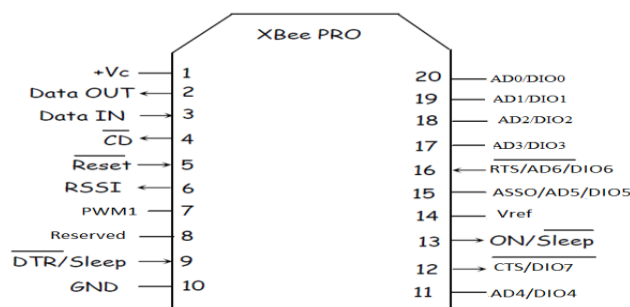
**Figure 19:** Xbee Pro Series 1

Ces modules peuvent être utilisés avec un ordinateur ou un microcontrôleur mais ils peuvent aussi fonctionner seuls. Ils disposent de six entrées analogiques et de huit entrées Numériques..

**4.4.2Caractéristique :**

- La portée théorique à l'intérieur est de 100 m et de 1500 m en extérieur.
- Alimentation entre 2,8 et 3,4 V.
- La consommation en émission est de 137 mA jusqu'à 210 mA.
- La consommation en réception est 50 mA.
- En mode "sleep" la consommation est inférieure à 10 mA.
- Le protocole utilisé est le 802.15.4 de la norme ZigBee.

La figure 20 montre les broches de l'Xbee Pro



**Figure 20 :** Les branches de l'XBEE pro séries 1

#### 4.5 Module d'affichage :

Le développement d'une application Android nécessite les composants suivant :

- Environnement de développement Eclipse.
- SDK Android (Software Development Kit).
- Plug in ADT (Android Development Tools).
- Smart Phone.

La figure 21 montre un équipement Android



**Figure 21:** Equipement Android

##### 4.5.1 La SDK Android (Software Development Kit):

SDK signifie Software Development Kit, c'est un ensemble d'outils d'aide à la Programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique. La SDK Android permet donc de développer des applications pour Android.

##### 4.5.2 L'IDE Eclipse :

Eclipse est un environnement de développement intégré, libre, extensible, universel, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

Pour développer notre application, nous avons utilisé Eclipse Mars.

##### 4.5.3 Le Plugin ADT pour Eclipse :

Google fournit un Plugin pour Eclipse, nommé ADT (Android Development Tools). Sa fonction principale est de créer un pont entre Eclipse et Android.

#### 4.5.4 Le SQLite database :

Les bases de données pour Android sont fournies à l'aide de SQLite. L'avantage de SQLite est qu'il s'agit d'un SGBD très compact et par conséquent très efficace pour les applications embarquées. Il est possible d'effectuer des opérations sur une base de données, comme créer des tables, supprimer des entrées, etc. Afin d'effectuer toutes ces opérations, on passe par un langage de requête.

L'exécution des requêtes est effectuée dans le même processus que l'application Android et une base de données est réservée à l'application créatrice, seule cette application créatrice peut y accéder.

#### 4.5.5 Smartphone :

Smartphone est un terme d'origine anglo-saxonne, signifiant littéralement « téléphone Intelligent ».

Selon le principe de l'ordinateur, il peut exécuter divers applications grâce à un système d'exploitation conçu spécialement pour mobiles, et donc, en particulier, fournir des fonctionnalités de plus de celles de téléphones mobiles classiques comme : l'agenda, la télévision, le calendrier, la navigation sur le web, la consultation et l'envoi du courrier électronique, la dictaphone/magnétophone, la calculatrice, la boussole, le gyroscope, la messagerie vocale, etc.

Il est possible de personnaliser son Smartphone en y installant des applications additionnelles telles que des jeux ou des utilitaires via un magasin d'applications en ligne différent pour chaque système d'exploitation comme Google Play sur Android.

## 5. Conception :

Après avoir présenté le matériel nécessaire dans de ce chapitre, nous allons, suivre une méthode de conception qui aura pour rôle la formalisation des étapes préliminaires du développement de notre système. La phase de conception permet de décrire de manière plus ou moins précise le fonctionnement futur du notre système et de faciliter sa réalisation.

Un système est un ensemble de composants qui sont combinés ensemble pour exécuter une tâche compliquée d'une manière coordonnée. Le composant est utilisé ici dans un sens très large, y compris les logiciels, le matériel informatique, le matériel numérique, l'alimentation, les capteurs et les actionneurs.

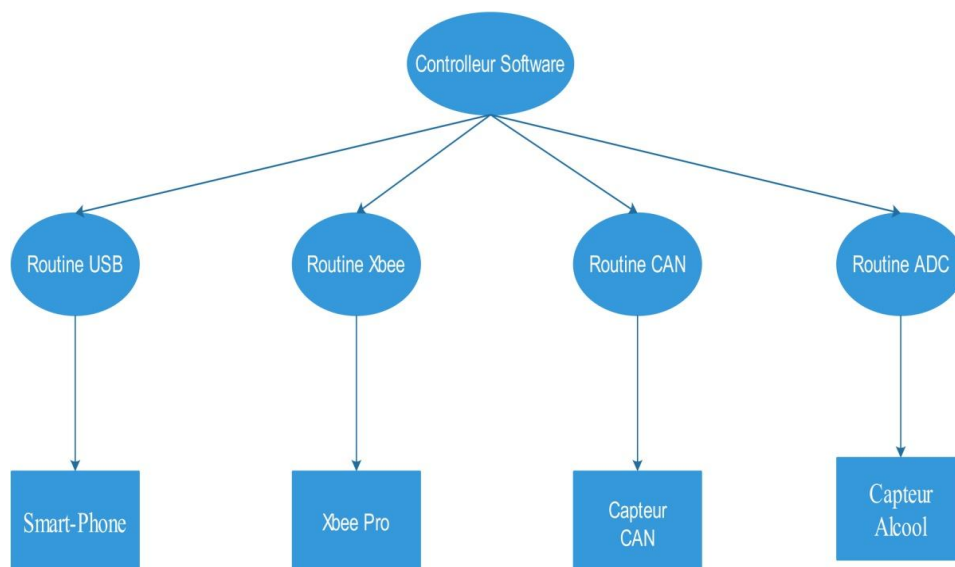
Le système est composé des composants et des interfaces. D'une manière récursive, on peut concevoir un composant en connectant des composants plus simples, avec des interfaces. En utilisant ce modèle, on peut envisager la conception de haut en bas comme le processus en commençant par un gros composant, puis subdiviser chaque composant dans les sous-simples reliés entre eux avec des interfaces.

Ce processus se termine lorsque chaque composant est si simple qu'il peut être construit ou acheté. Le test peut alors procéder de façon ascendante en connectant les composants réels ainsi que des interfaces réelles.[27]

### 5.1 Les graphes d'appels :

Les graphes d'appels sont un moyen graphique pour définir la façon dont les modules de logiciel et matériel sont interconnectés, les rectangles représentent les composants matériels et les ovales montrent des modules logiciels. les flèches de la routine d'appel sont dirigées vers les modules appelé. Les ports d'entrées et sorties sont organisées en groupes et placés en bas du graphique.

La figure suivante représente le graphe d'appels pour le système embarqué dans le véhicule :



**Figure22:** Le graphe d'appels pour le système embarqué véhicule

La figure suivante représente le graphe d'appels pour le système embarqué dans la borne de police :

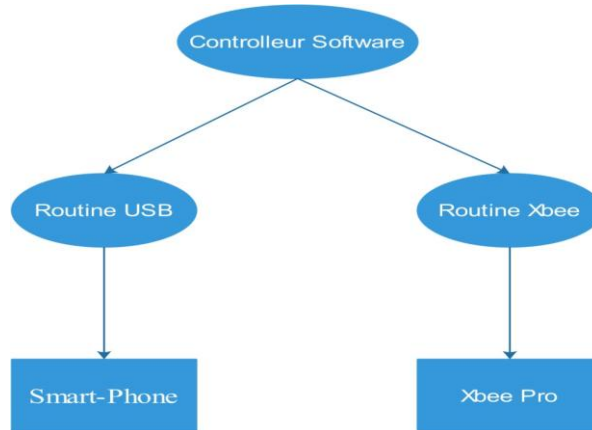


Figure 23 : Le graphe d'appels pour le système embarqué dans la borne de police

### 5.2 Le graphe de flux de données :

Le graphe de flux de données est un schéma fonctionnel du système montrant le flux d'informations. Les flèches se pointent de la source vers la destination. Les rectangles représentent les composants matériels et les ovales sont des modules logiciels.

Nous utilisons des données graphiques de flux dans la conception de haut niveau, car ils décrivent le fonctionnement global du système, tout en cachant les détails de la façon dont il fonctionne.

La figure suivante représente le graphe de flux de données pour le système embarqué dans le véhicule :

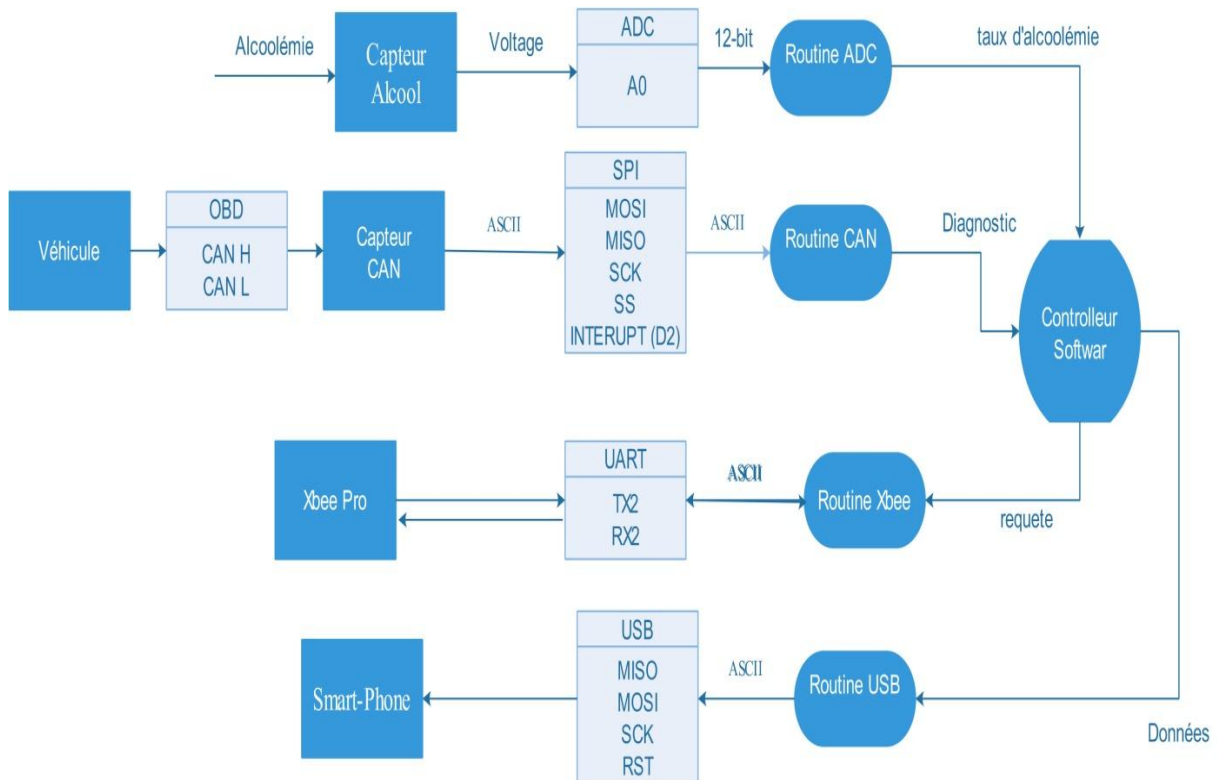
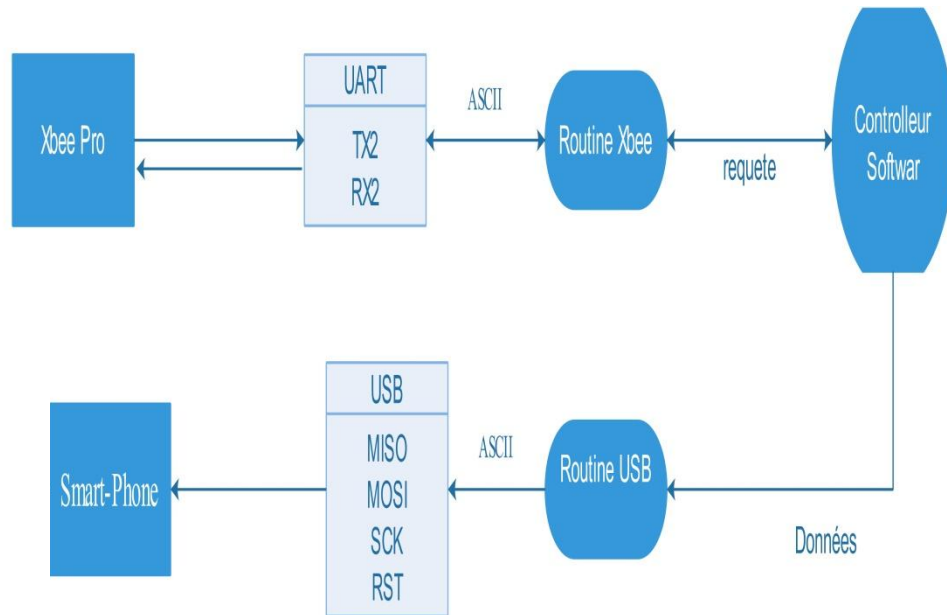


Figure 24: Le graphe de flux de données pour le système embarqué véhicule

La figure suivante représente le graphe de flux de données pour le système embarqué dans la borne de police :



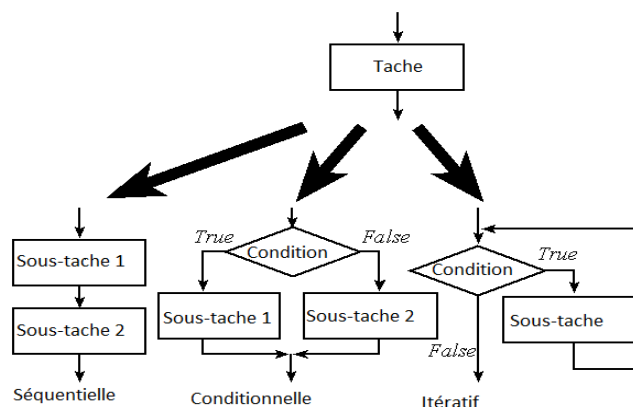
**Figure 25:** Le graphe de flux de données pour le système embarqué borne de police

### 5.3 Le graphe de Raffinement successives :

Raffinement successif, affinement progressif, et la décomposition systématique sont trois termes équivalents à une technique pour convertir un énoncé de problème dans un algorithme logiciel.

Nous commençons par une tâche puis on la décompose en un ensemble de sous-tâches plus simples. Ensuite, les sous-tâches sont décomposées en sous-sous-tâches encore plus simples. Nous faisons des progrès aussi longtemps que chaque sous-tâche est plus simple que la tâche elle-même.

Finalement, une sous-tâche est si simple qu'elle peut être converti en un code de logiciel. on peut décomposer une tâche de trois manières différentes, comme le montre la figure suivante :



**Figure 26:** Décomposition d'une tâche en sous tâches

La figure suivante représente Le graphe de Raffinement successives pour le système embarqué dans le véhicule :

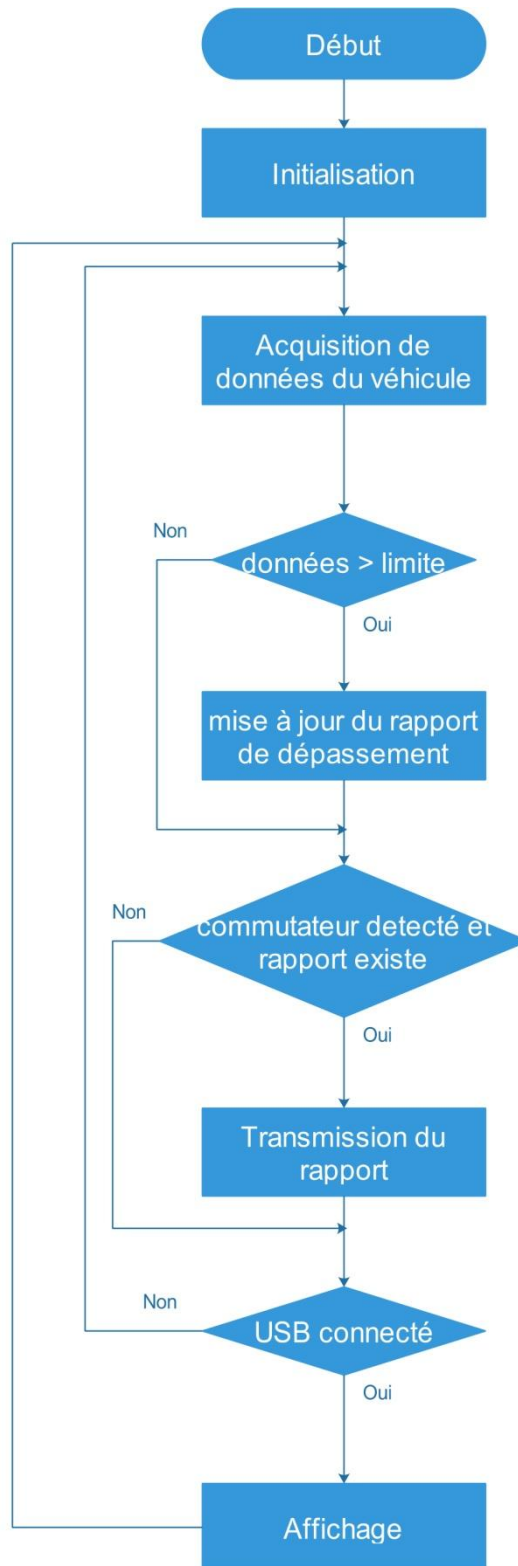
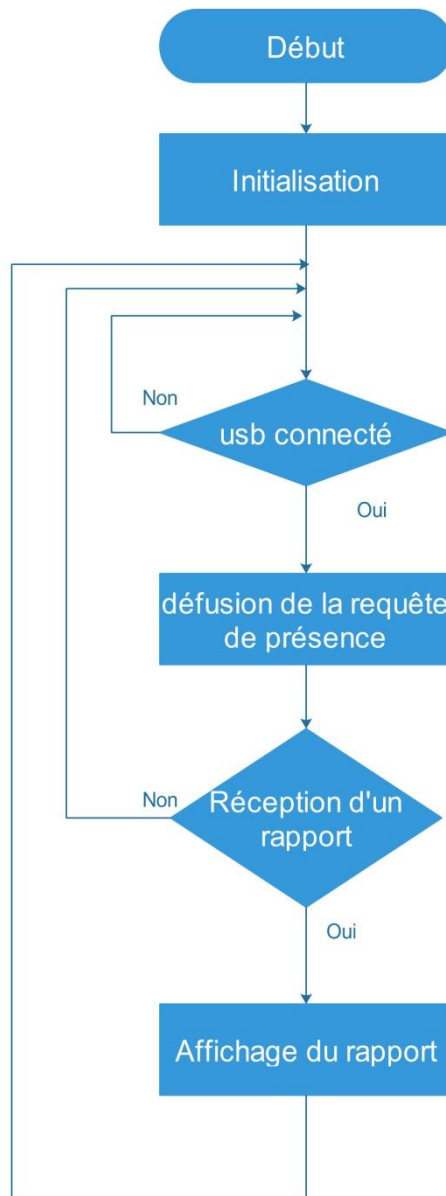


Figure 27 : Le graphe de Raffinement successives pour le système embarqué dans le véhicule

La figure suivante représente Le graphe de Raffinement successives pour le système embarqué dans la borne de police :



**Figure 28:** Le graphe de Raffinement successives pour le système embarqué dans la borne de police.

## 6. Conclusion :

Tout au long de ce chapitre, nous avons exposé les différents besoins auxquels doit répondre l'application ainsi que l'étude des différents concepts et moyens utilisés pour la réalisation de cette application.

Dans le chapitre suivant, nous allons aborder l'implémentation de notre système en pratique.

# Chapitre IV : Réalisation et Test

## 1. Introduction :

La réalisation est la phase la plus importante après celle de la conception. Dans cette phase nous allons transformer le modèle conceptuel établi en des composants logiciels formant notre système. Puis, nous nous intéresserons à décrire les différentes étapes de réalisation de l'application en présentant ses fonctionnalités via ses différentes interfaces. Pour finir, nous effectuons une série de tests.

## 2. Principe de fonctionnement du système :

La figure 29 montre le schéma fonctionnel de notre système qui permet de récupérer en temps réel le régime moteur, la vitesse et le taux d'alcoolémie à partir d'un véhicule et de comparer ces données acquises à des limites réglementaires, en cas de dépassement de ses limites des informations du véhicule (immatriculation, vitesse, régime moteur, etc.) seront envoyés au poste de police qui pourra les sauvegarder dans une base de données .



Figure 29 : Schéma fonctionnel de système

### 3. Réalisation matérielle :

Notre système est divisé en deux parties :

- Système embarqué véhicule
- Système embarqué borne de police

Pour la réalisation de ces parties nous avons utilisé les brochages suivant :

#### 3.1 Carte Arduino Méga Adk et CAN bus shield :

- Pour l'alimentation :

Arduino Adk	CAN Bus Shield
Reset	Reset
5V	5V
GND	GND
Vin	Vin

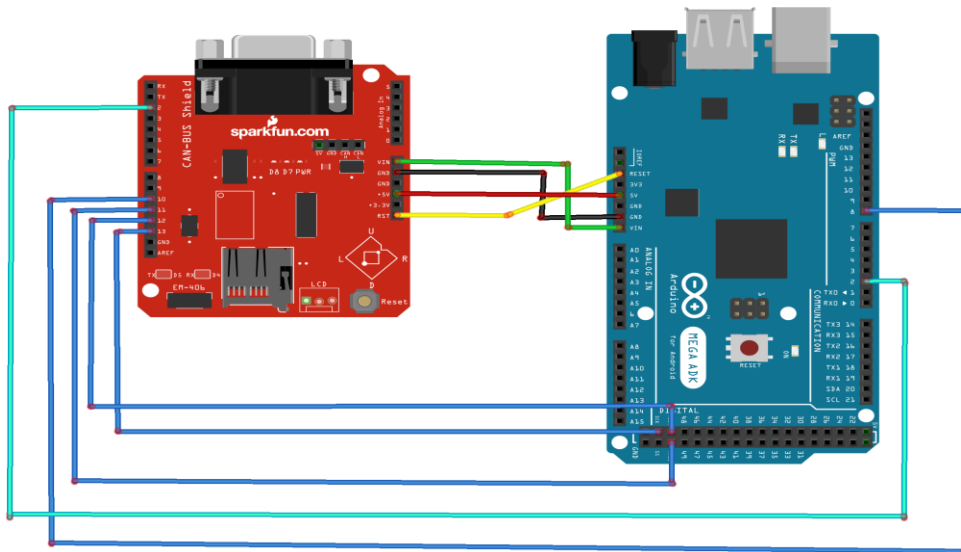
**Tableau 9:** Les broches d'alimentation carte ADK et CAN bus

- Pour la communication :

Arduino Adk	CAN Bus Shield
MOSI	P11
MISO	P12
P8	P10
CLK	P13
INT0	P2

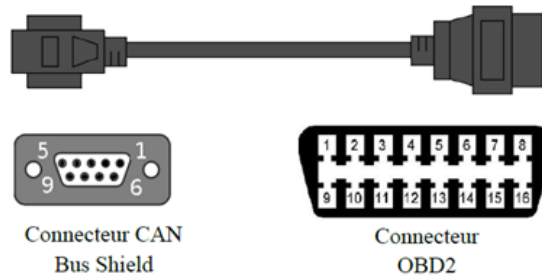
**Tableau 10:** Les broches de communication carte ADK et CAN Bus Shield

La figure 30 montre le schéma de brochage sur le quel nous avons établi les liaisons suivantes:



**Figure 30:** Schéma de brochage du CAN Bus Shield avec Arduino ADK

**CAN Bus Shield et la prise OBD:** dans le but d'extraire les informations, le Can bus Shield est branché au véhicule grâce au câble obd2-db9 comme le montre la figure 31



**Figure31:** Le Brochage CAN Bus Shield et OBD2

### 3.2 Carte Arduino Adk et Xbee Pro Series1 :

- Pour l'alimentation :

Arduino mega Adk	Xbee Pro Series1
3.3V	3.3V
GND	GND

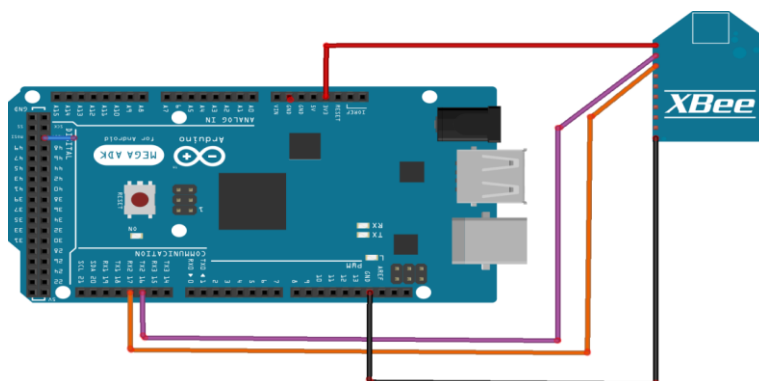
**Tableau 11 :** Les broches d'alimentation entre arduino mega ADK et Xbee Pro

- Pour la communication :

Arduino Adk	Xbee Pro Series1
RX2	RX
TX2	TX

**Tableau 12:** Les broches de communication entre ADK et Xbee Pro S1

La figure 32 montre le schéma de brochage sur le quel nous avons établi les liaisons suivantes:



**Figure 32:** Schéma de brochage entre Arduino MEGA ADK et XBEE PRO S1

### 3.3 Carte Arduino Mega Adk et MQ135 :

- Pour l'alimentation :

Arduino Adk	MQ135
5V	5V
GND	GND

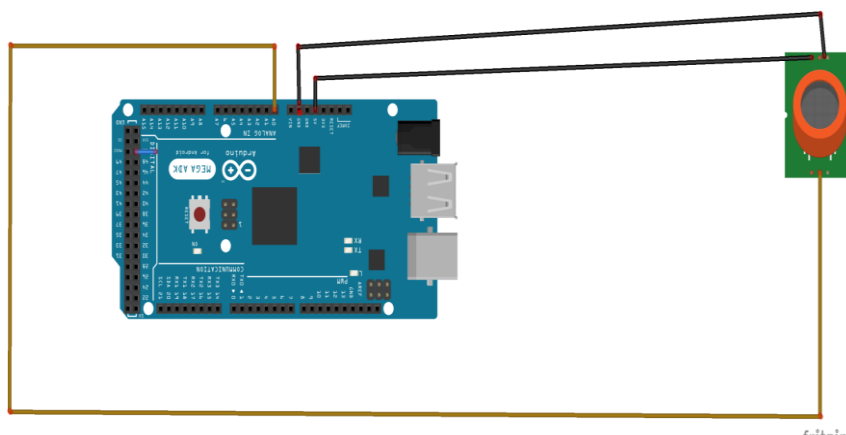
**Tableau 13 :** Les broches d'alimentation entre ADK et MQ 135

- Pour la capture de taux d'alcoolémie :

Arduino Adk	Xbee Pro Series1
A0	A0

**Tableau 14:** La broche de communication entre MQ 135 et ADK

La figure 32 montre le schéma de brochage sur le quel nous avons établi les liaisons suivantes:



**Figure 33:** Schéma de brochage du MQ135 et ADK

### 3.4 Carte Arduino Mega Adk et Smart Phone :

Le brochage ce fait à laide d'un Câble Mini USB qui permet de relier la carte Arduino Adk à un smart phone comme montre la figure 34.



**Figure 34:** Câble Mini USB

### 3.3 Schème Global du brochage dans le Système embarqué véhicule :

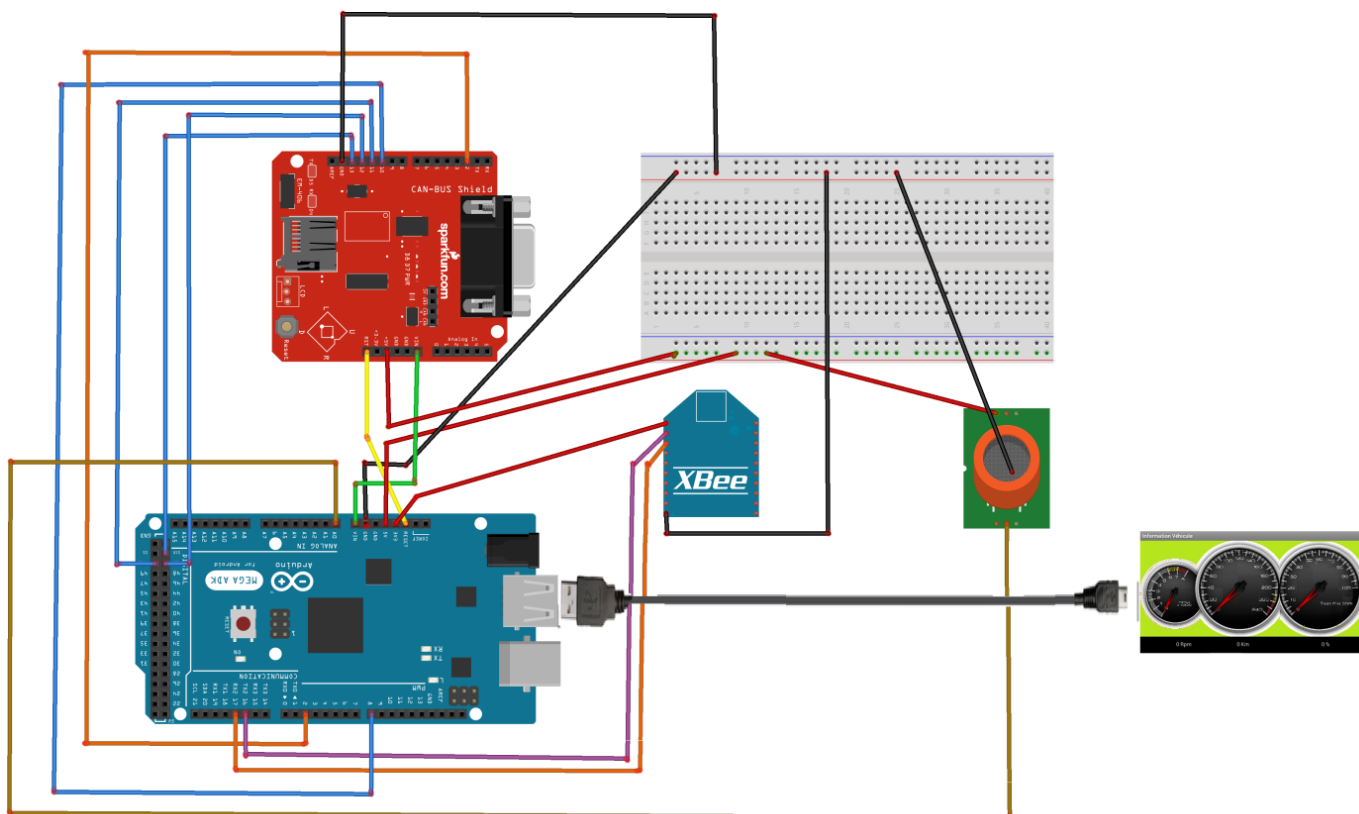


Figure 35 : Schéma de brochage général coté véhicule

### 3.4 Schème Global du brochage dans le Système embarqué borne de police :

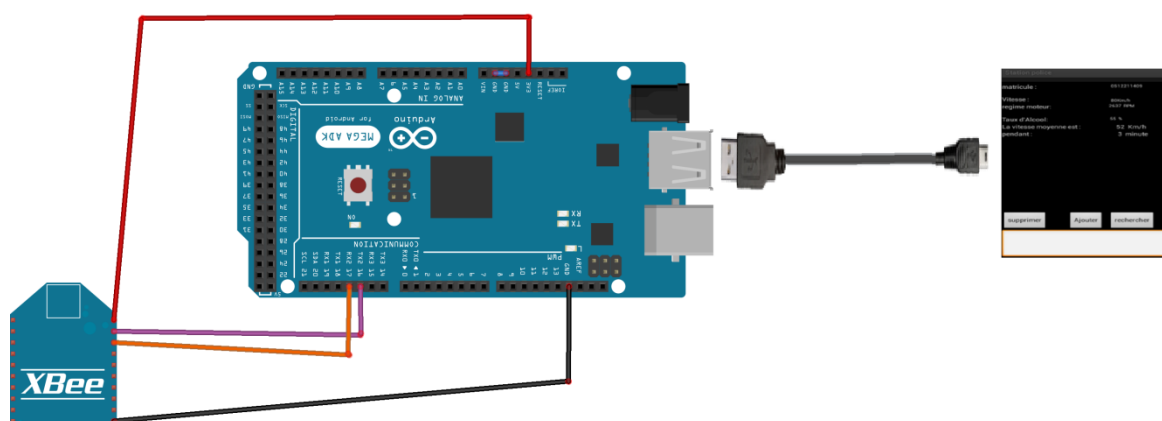


Figure 36 : Schéma général du brochage borne de police

#### 4. Présentation des applications du système :

De point de vue fonctionnel, notre système est subdivisé en deux parties :

- un système embarqué véhicule
- un système embarqué borne de police

4.1 Lancement de l'application : l'utilisateur doit accepter la communication USB comme le montre la figure 37.

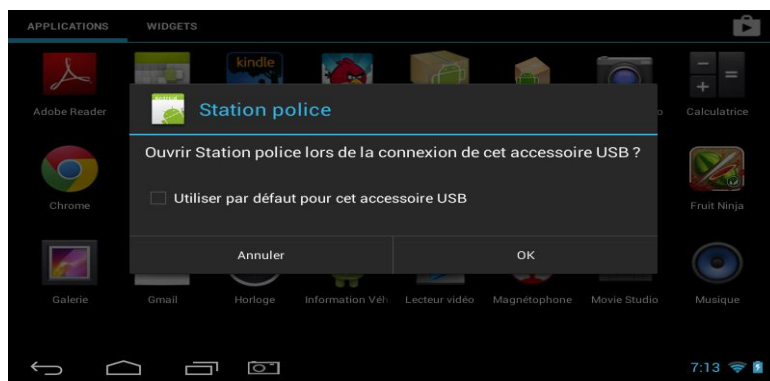


Figure 37: Confirmation de communication entre l'application android et l'arduino MEGA ADK via le câble USB

#### 4.2 Système embarqué véhicule :

Le conducteur peut visualiser des informations à propos de sa vitesse, son régime moteur et son taux d'alcoolémie en temps réel dans une interface graphique comme le montre la figure 38.

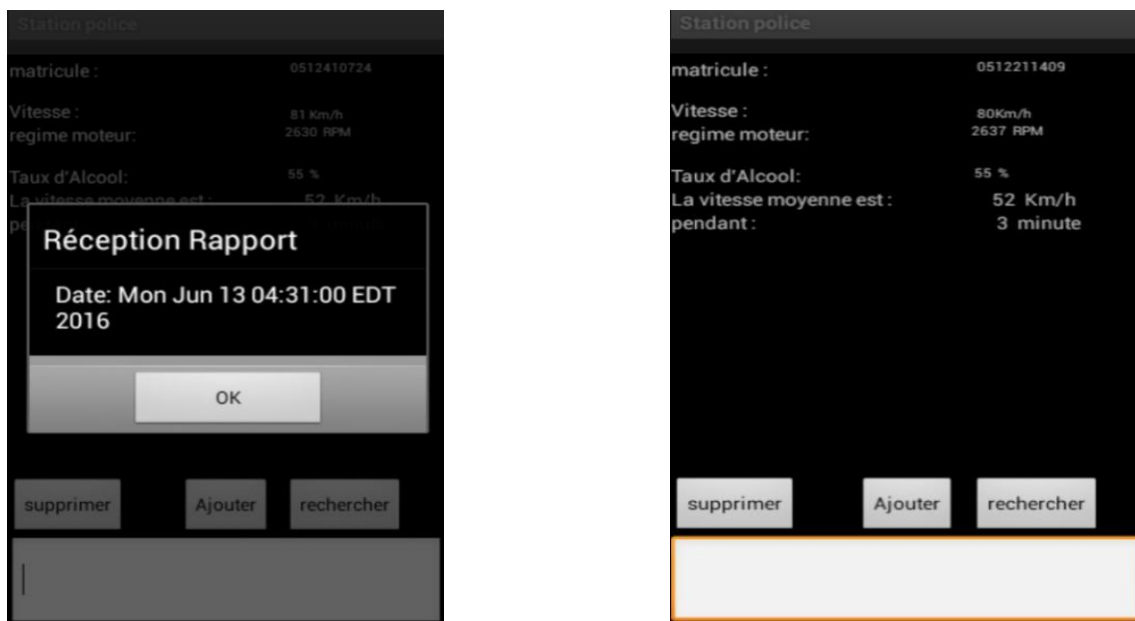


Figure 38: Présentation de l'application android Informations véhicule

### 4.3 Système embarqué dans la borne de police :

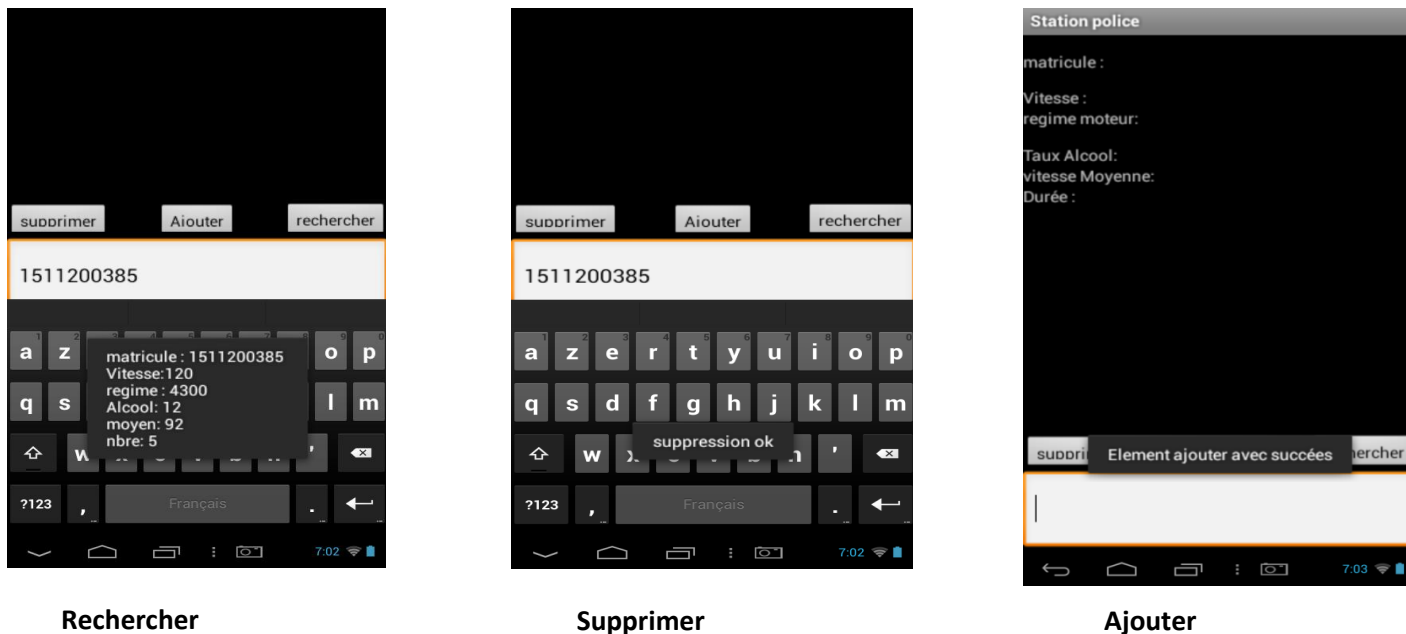
Le policier peut consulter un rapport de dépassement lors de sa réception, puis dans le cas où il n'arrive pas à pénaliser le conducteur il doit ajouter le rapport à la base de données.

- réception d'un rapport : lors de la réception d'un rapport un message d'alerte s'affiche, après sa validation les données du rapport s'affichent dans la page d'accueil comme le montre la figure 39, ensuite le policier peut effectuer les opérations de recherche, ajout, suppression. Comme le montre la figure 40



**Figure39:** Exemple de l'application lors de réception d'un rapport coté police

- En cliquant sur le Botton Ajouter, le rapport sera ajouté à la base de données et un message de confirmation affichera.
- La recherche et la suppression dans la base de données ce fait par Immatriculation



Rechercher

Supprimer

Ajouter

**Figure 40:**Présentation des opérations de l'application android borne de police

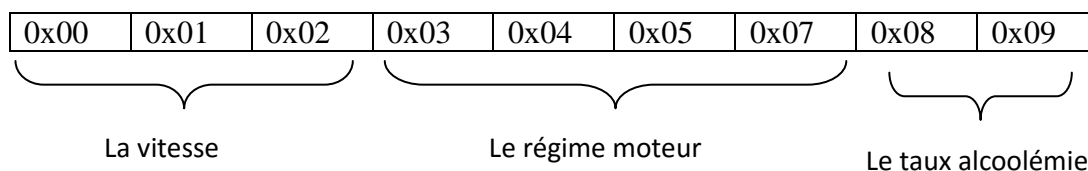
## 5. Elaboration du programme :

L'application programmé essentiellement en c et en java Android est devisé en plusieurs programmes qui communiquent à l'aide de plusieurs buffers d'échange de données, ces échanges servent a l'exécution des différents programmes de l'application, dans ce qui suit nous allons expliquer le fonctionnement de ces buffers.

### 5.1 Communication intra-système :

Il existe trois buffer de communication tel que :

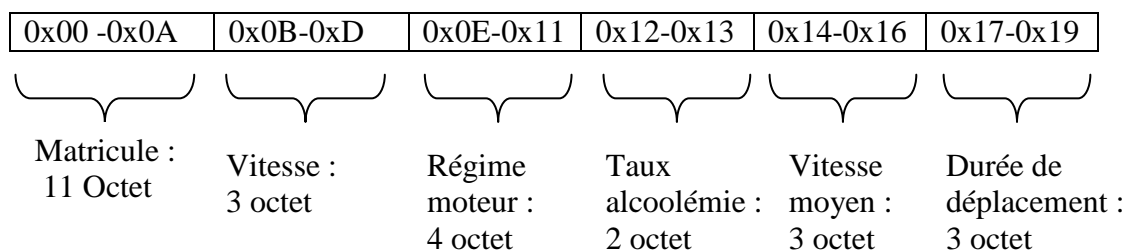
- Buffer 1 (Arduino Adk / tablette) : un buffer de 16000 octets est crée pour les échanges entre la tablette et l'Arduino Adk à laide de la liaison USB, notre programme principale utilise seulement 9 octet et cela pour transmettre les données sous le format suivant : la vitesse 3 octet, régime moteur 4 octet et taux d'alcoolémie 2 octet.



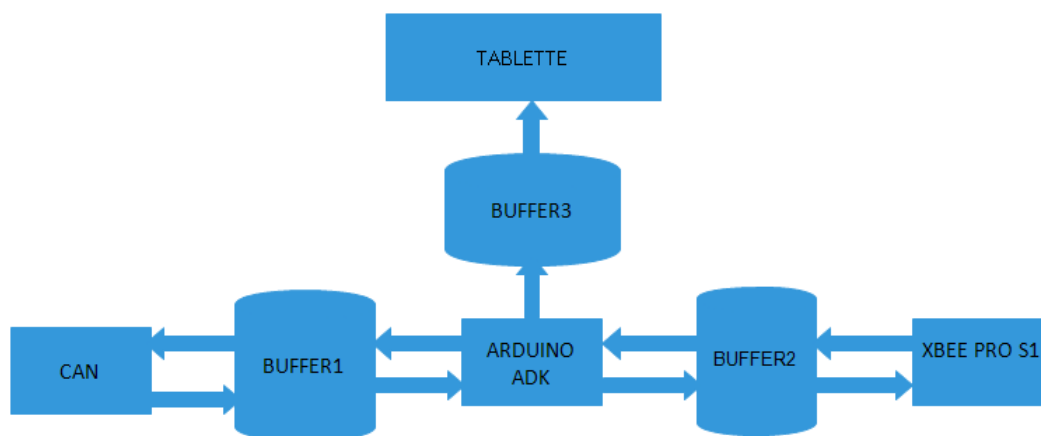
- Buffer 2(Arduino Adk / CAN BUS) : a la même taille d'une trame CAN il sert a transmettre les commandes CAN et récupérer les données résultantes



- Buffer 3(Arduino Adk / Xbee Pro Séries 1) : permet d'envoyer un rapport vers le composant Xbee, un rapport est composé de 25 octets de la façon suivante :



La figure 41 montre les différentes communications dans le système embarqué véhicule à l'aide des trois buffers décrits précédemment :



**Figure 41:** Schéma général des buffers existant dans le système

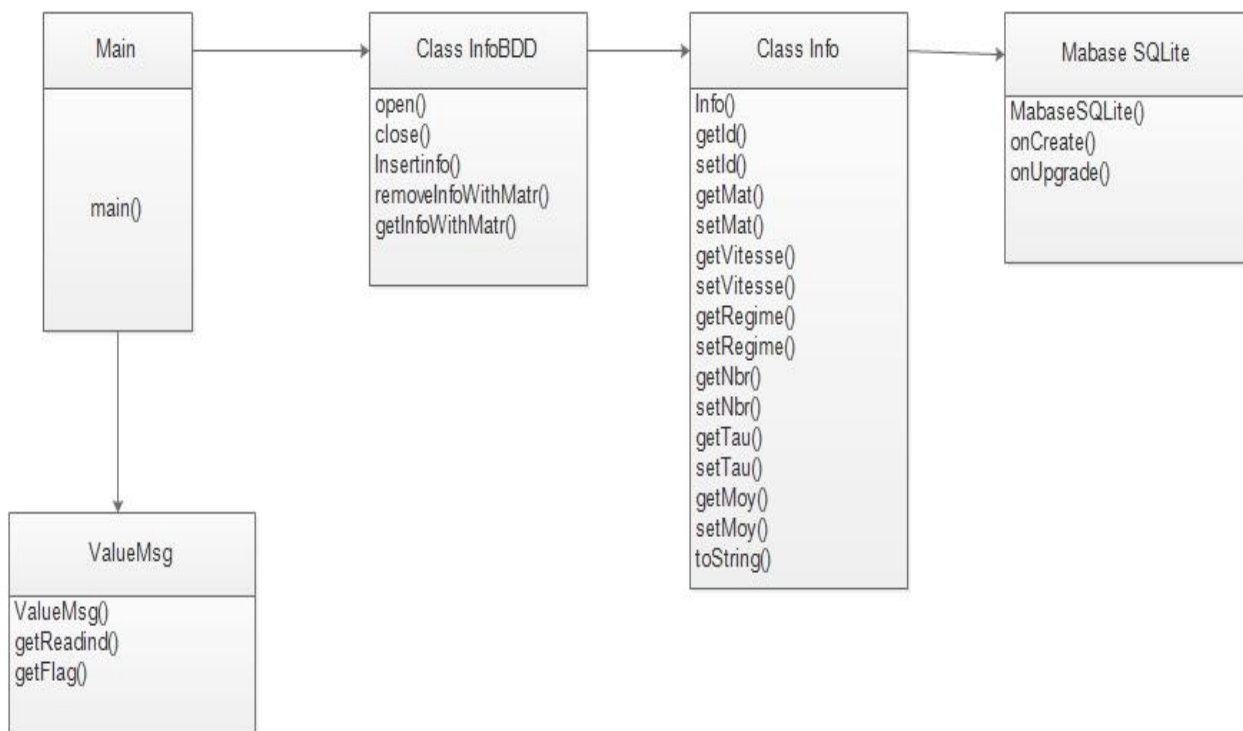
### 5.2 Communication entre les deux systèmes :

- Le système embarqué borne de police diffuse d'une façon périodique une trame de présence.
- quand le système embarqué véhicule est en état de dépassement de la réglementation routier et il détecte la présence de la borne de police il lui transmet un rapport.

### 5.3 Application Android borne de police :

Le diagramme de classes est un schéma statique. Il représente le point de vue statique d'une application, il est non seulement utilisée pour visualiser, décrire et documenter les différents aspects d'un système, mais aussi pour la construction de code exécutable de l'application logicielle.

La figure 42 montre le diagramme de classe de notre application dans le cas de la borne de police.



**Figure 42 :** Diagramme de classe pour l'application borne de police

**Présentation des Classes :** Le diagramme de classe de notre application se compose de cinq classes :

**1. Classe de création et manipulation de base de données :**

**Class Info :** Cette classe est très simple puisque dans notre cas, une information est définie par un identifiant, et les données du rapport.

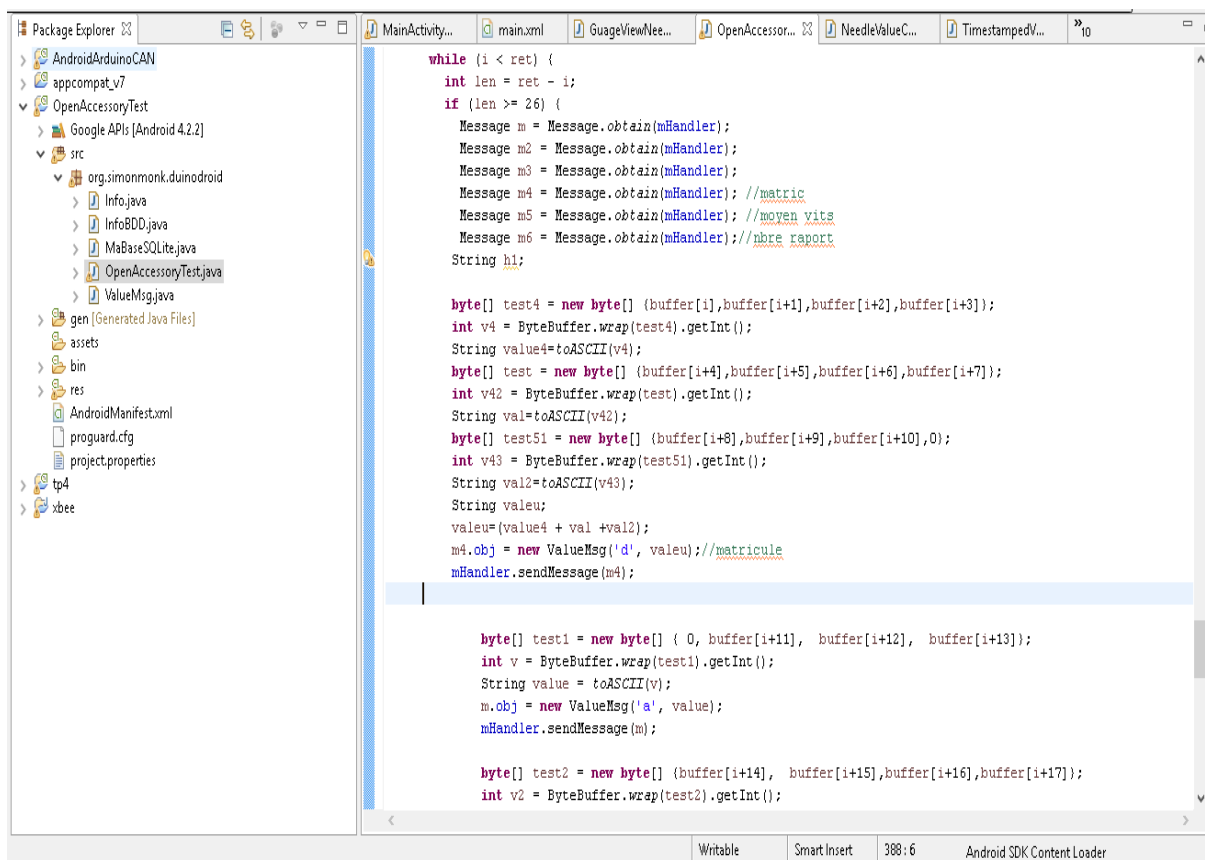
**Class MaBaseSQLite :** cette classe est étendue de SQLiteOpenHelper qui est une classe d'assistance pour gérer la création de bases de données. Cette classe va permettre de définir la table qui sera produite lors de l'instanciation de celle-ci.

**Class InfoBDD :** Elle va nous permettre de gérer l'insertion, la suppression, la modification des rapports dans la base de données.

**2. Classe d'affichage :**

**Class open Accessoire test :** est l'activité principale de l'application qui s'occupe de l'initialisation de la communication USB avec la carte Arduino MEGA ADK pour faire l'affichage des rapports reçu ,et elle fait appel à la classe Infobdd pour effectuer les différentes opérations sur la base de donnée, cette classe est présenté dans la figure 43.

**Class value msg** : cette classe permet de créer une structure qui contient les différentes valeurs des données échanger entre Arduino Adk et l'application android.



**Figure 43** : Présentation de la Classe open Accessoire test

### 5.4 Application Android dans le véhicule :

Se compose de deux types de classe :

#### 1. Classe d'affichage :

Comme dans l'application précédente il ya deux classes qui permettent de récupérer une donnée de l'arduino adk et l'afficher dans l'application.

**La class main** : permet d'initialiser la communication USB puis d'afficher les valeurs reçues. Cette classe est présentée dans la figure 44.

**Class value msg** : cette classe permet de créer une structure qui contient des différentes valeurs des données échanger entre Arduino ADK et l'application android.

## 2. Classe de variation de l'aiguille dans l'affichage

D'autres classes sont utilisées pour contrôler la variation de l'aiguille qui correspond à la vitesse et au régime moteur du conducteur

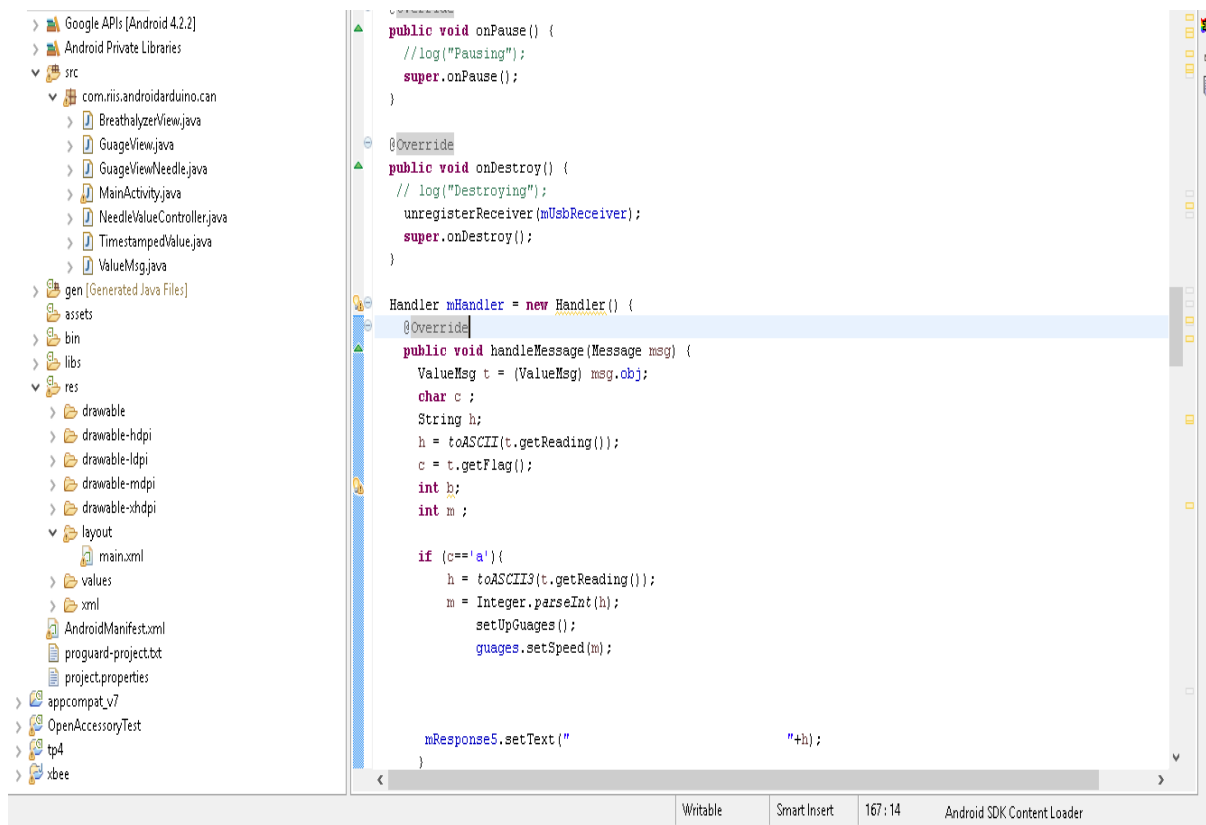


Figure 44:Présentation de la Classe main

## 6. Test :

Dans cette partie nous allons présenter quelques figures qui illustrent notre système mise en pratique avec les tests que nous avons fait sur différents véhicules.

### 6.1 Présentation d'ULM 327 :

Comme nous avons déjà parlé dans les chapitres précédents, il existe plusieurs protocoles OBD afin de voir le véhicule ayant le protocole CAN nous avons utilisé l'ULM327 qui est une interface plus courante et la plus pratique, elle supporte tous les protocoles OBD, tel que le KWP et le CAN. Comme le montre la figure 45.



### Logiciel EOBD facile :

Figure 45: Interface ELM327

Le logiciel EOBD-Facile développé par l'équipe d'outils OBD facile permet d'utiliser les interfaces de type ELM327. Il nous permettra ainsi de dialoguer avec les véhicules pour connaître leur spécification (protocole utilisé, emplacement de la prise OBD). L'interface EOBD facile est présentée dans la figure 46.

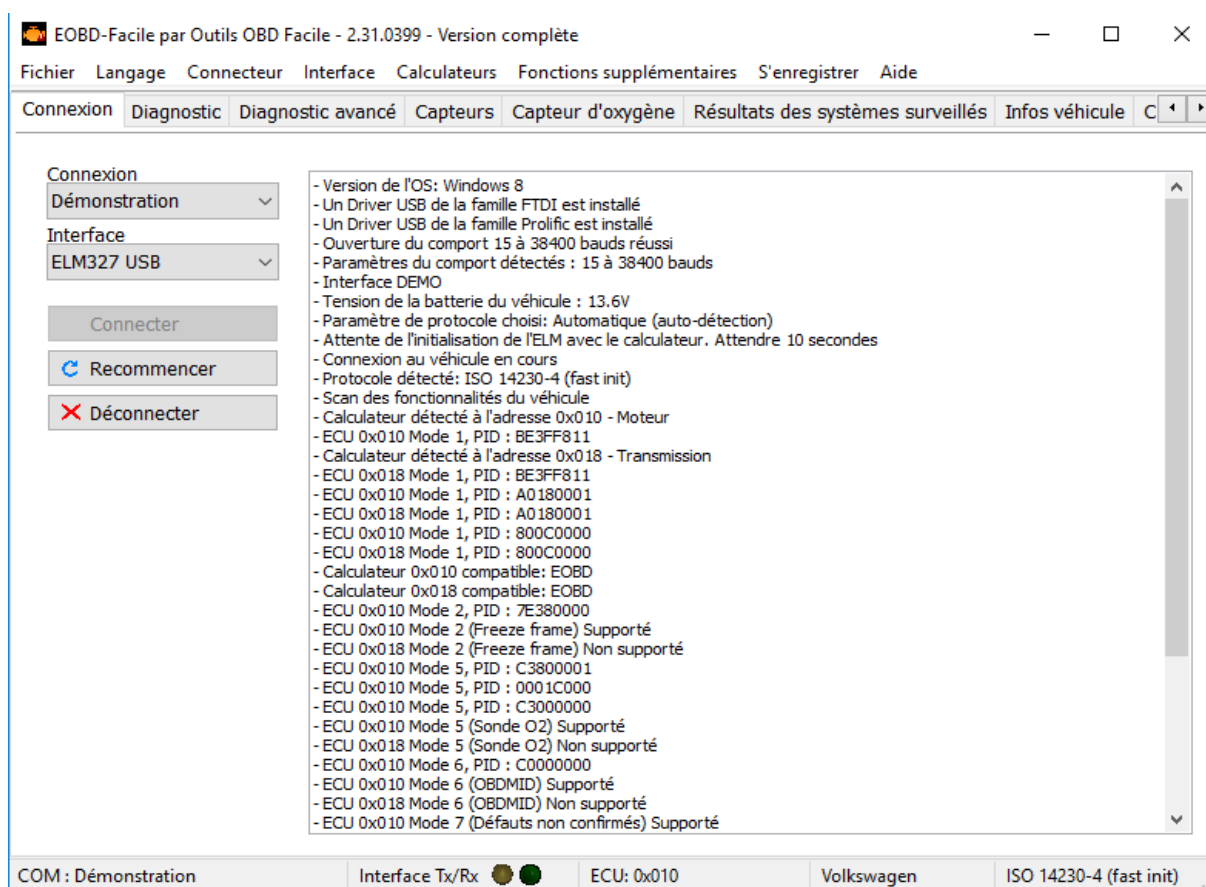


Figure 46: Présentation du logiciel EOBD facile

## 6.2 Véhicule testé :

Après avoir fait plusieurs tests, nous avons constaté que les marques Volkswagen et Skoda utilisent la communication bus CAN, qui est compatible avec notre système.

## 6.3 Test N°1 : Volkswagen Passat



**Figure 47 :** Volkswagen Passat

Le tableau 15 montre les principales caractéristiques d'une passat

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Volkswagen	Passat	1.9 tdi	2007	Diesel	CAN 11bits 500 kb

**Tableau 15:** Information sur le véhicule du teste N°1

### 6.3.1 Emplacement prise OBD dans la passat :

La prise OBD se trouve sous le volant au coté gauche du conducteur comme le montre la figure 48.



Figure 48 : Emplacement prise obd dans la Passat

### 6.3.2 Matériels Utilisés dans le test :

Pour le premier test nous avons utilisé la Carte Arduino UNO, le CAN bus shield et le logiciel arduino.

### 6.3.3 Objectif de Test:

Comprendre la communication dans le protocole CAN avec l'identification des calculateurs, et les PID utilisés pour le régime moteur, vitesse, la température de refroidissement moteur etc. La figure 49 montre une photo prise lors du test effectué



Figure 49: Photo prise lors du test effectué

### 6.3.4 Résultat de Test N°1 :

Ce premier test nous a permis de récupérer la vitesse et de comprendre le fonctionnement du CAN avec l'identification de ses équipements, qui nous aidera à améliorer notre travail au prochain test.

La figures 50 montre l'équivalence entre les données affichées dans le compteur minute véhicule et les résultats obtenus.

La figure 51 montre nos résultats d'une façon plus claire.

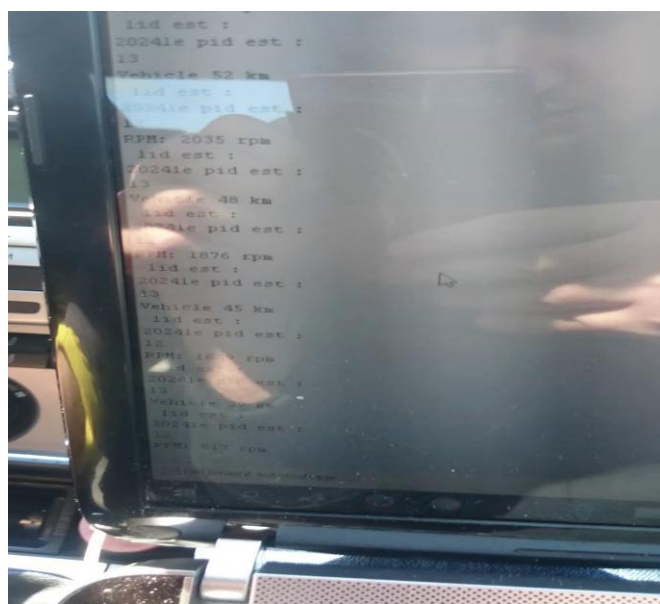


Figure 50: Résultat du Test 1

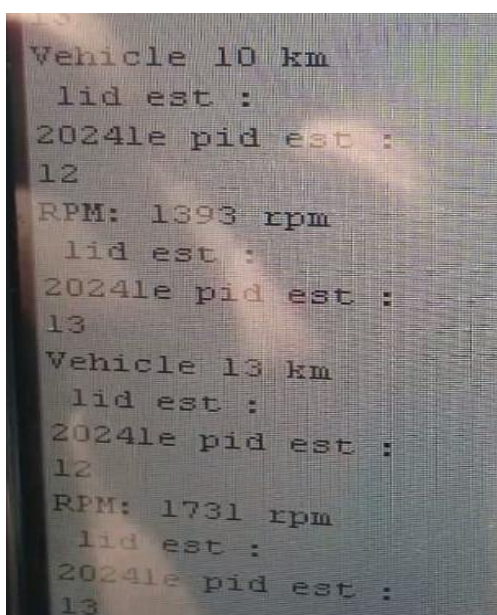


Figure 51 : Capture plus claire des résultats obtenus

**6.4 Test N°2 : Volkswagen Golf 5**



**Figure 52 : Volkswagen Golf 5**

Le tableau 16 montre les caractéristiques de la Golf 5

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Volkswagen	Golf 5	TDI	2015	Diesel	CAN 11bits 500 kb

**Tableau 16 : Information sur le véhicule du Test N°2**

**6.4.1 Emplacement de la prise obd :** la prise OBD se trouve sous le volant dans le côté gauche du conducteur comme le montre la figure 53



**Figure 53 : Branchement de la prise obd dans la golf 5**

**6.4.2 Matériels Utilisés : Carte Arduino UNO, CAN bus shield.**



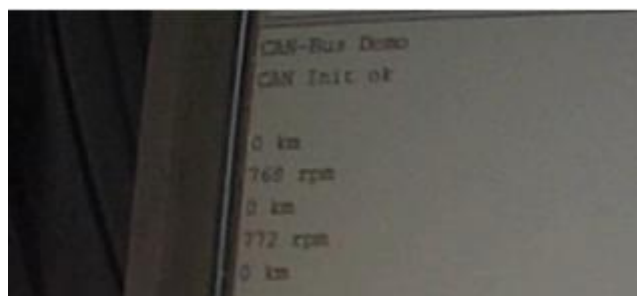
**Figure 54 :** Image réel du test 2 réalisé sur la golf 5

**6.4.3 Objectif de Test:**

Après avoir compris le principe des identifiants CAN, nous avons élaboré un nouveau programme qui permet le filtrage des identifiants grâce aux masques et filtres du module CAN BUS Shield vu en chapitre 3 section 4.3.1

**6.4.4 Résultat de Test N°2 :**

La figure 55 montre le résultat du test 2 qui permet de récupérer des données à partir de la Golf 5 d'une façon optimisée



**Figure 55 :** Résultat obtenu du test 2

**6.5 Test N°3 : Skoda Fabia**



**Figure 56 : Skoda Fabia**

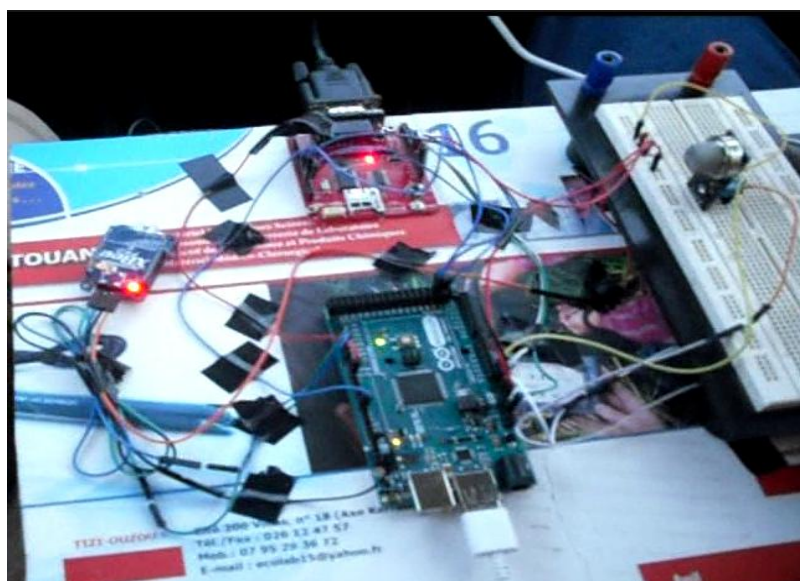
Le tableau 17 montre les principales caractéristiques de la Skoda Fabia

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Skoda	Fabia	1.6TDI	2014	Diesel	CAN 11bits 500 kb

**Tableau 17:** Information sur le véhicule de teste N°3

**6.5.1 Matériels Utilisés :**

Deux Cartes arduino MEGA ADK, une carte CAN bus shield, deux XBEE PRO S1, un capteur d'alcool MQ135, et une tablette et un Smartphone android. Comme le montre la figure 57.



**Figure 57 : branchement effectué dans le test**

### 6.5.2 Objectif de Test N°3:

Ce test récapitule tout notre travail en rassemblant les résultats valides de nos tests précédents, il permet de vérifier :

- l'affichage des données acquises de la skoda dans l'application android.
- la création et la transmission d'un rapport de dépassement de règlement routier.
- l'affichage du rapport dans la borne de police.

### 6.5.3 Résultat de Test N°3 :

La figure 58 nous montre un Comparatif entre valeur du compteur minute de véhicule et notre application android, se qui permet de confirmer l'exactitude des valeurs acquises du véhicule et de leur affichage.



Les valeurs de compteur minute (Fabia) durant le test 3

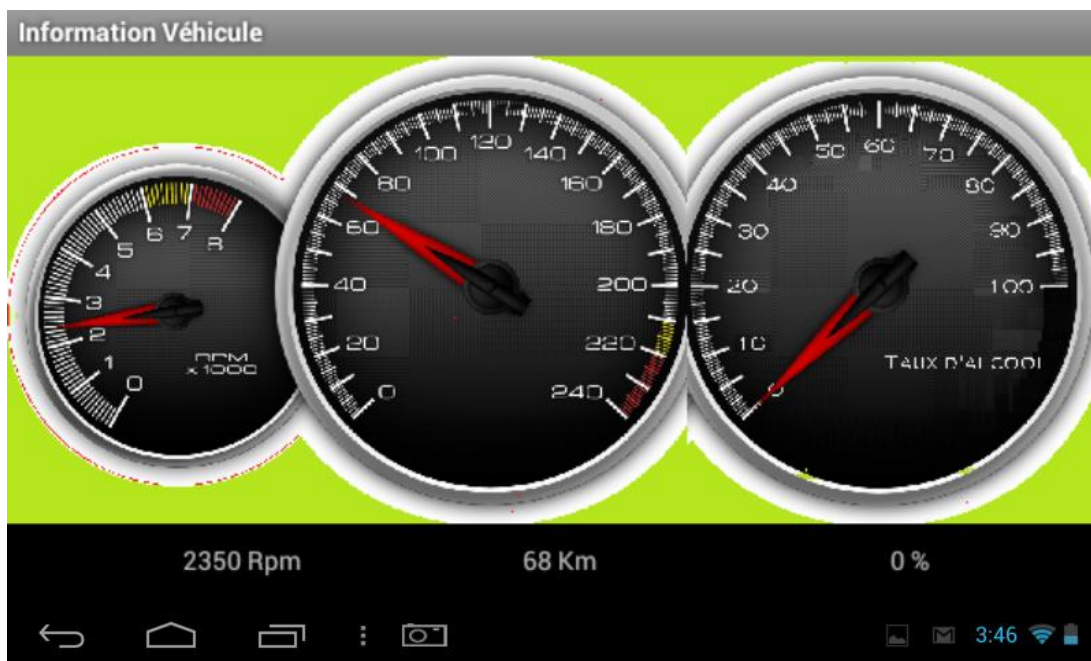


Les valeurs de compteur minute (application android) durant le même test

**Figure 58** : Comparaison de valeur entre le compteur minute et application android

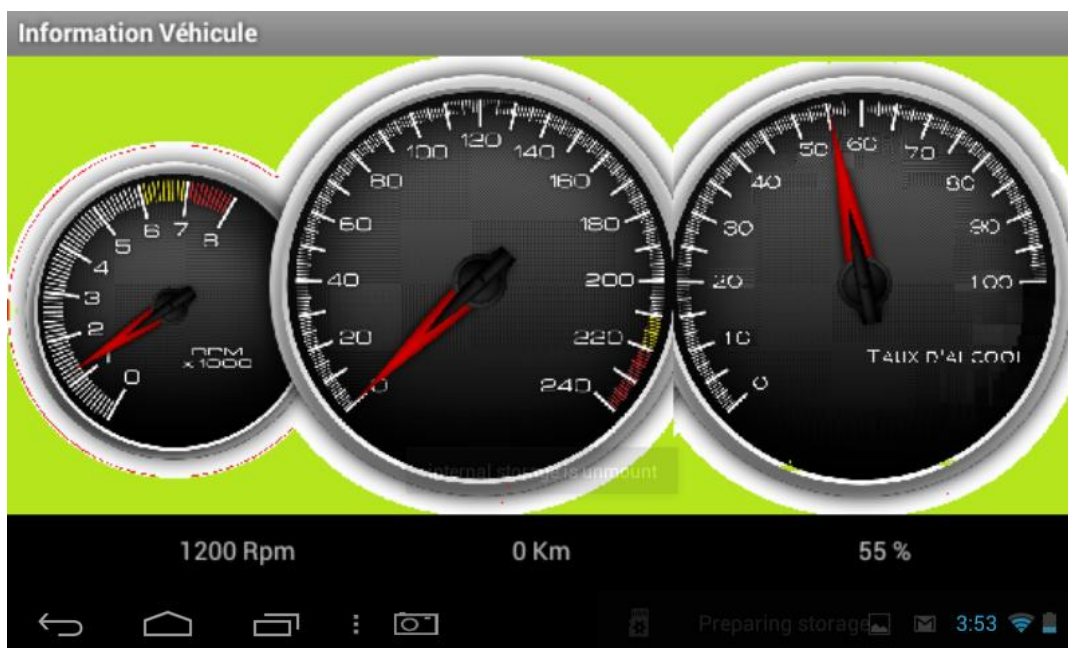
Afin de tester la communication entre le véhicule et la borne de police, durant le trajet parcouru le véhicule a franchi les limites réglementaire en ce qui concerne la vitesse autorisé dans une zone urbaine ( 40 km/h ) et le taux d'alcoolémie.

La figure 59 montre un dépassement de vitesse enregistré au cours du trajet effectué :



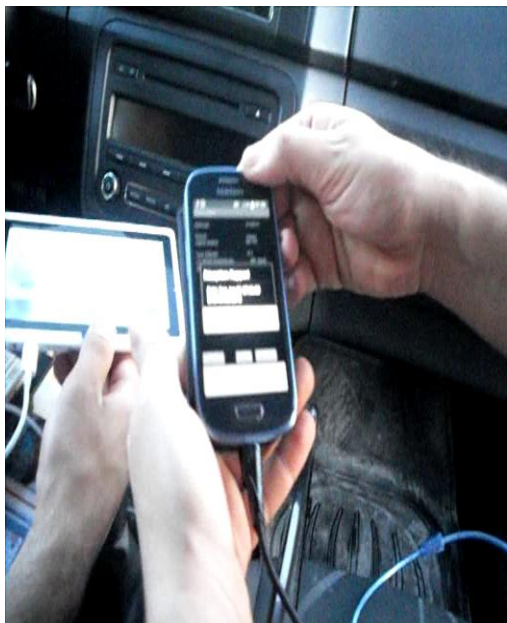
**Figure 59** : Capture d'écran d'un dépassement de la limite réglementaire de la vitesse dans une zone urbaine

La figure 60 montre un dépassement de taux d'alcoolémie enregistré au début du trajet



**Figure 60** : Capture d'écran d'un dépassement de la limite réglementaire du taux d'alcoolémie.

La deuxième partie du test consiste à brancher la borne de police et de vérifier la réception du rapport de dépassement du règlement routier qui a été enregistré durant la première partie du test. Les figures 61,62 montrent les résultats obtenus lors de l'alimentation de la borne de police.



**Figure 61** : Réception du rapport dans la borne de



**Figure 62** : Résultats obtenus

Ce dernier test nous a permis de valider notre travail dans le quel on a testé toutes les fonctionnalités réalisées dans notre projet notamment :

- l'acquisition des données à partir d'un véhicule.
- l'affichage des données acquises en temps réel sur une tablette Android dans un véhicule.
- la vérification de la conformité des données au règlement.
- la transmission des données et leurs affichages dans la borne de police.

### **7 Conclusion :**

Au cours de ce de dernier chapitre, nous avons décrit les étapes de réalisation et de test de notre système. Dans la première partie, nous avons procédé, d'une part, à une description fonctionnelle de notre système ainsi que les brochages matériels nécessaires à sa réalisation, et d'autre part, à la description de notre application en présentant quelques interfaces essentielles.

Ensuite, dans la deuxième partie, nous avons présenté quelques tests effectués durant notre travail ainsi que les résultats obtenus illustrés par quelques figures.

# Conclusion et perspectives

## **Conclusion générale :**

Le principal objectif de ce travail est la conception et la réalisation d'un système de sécurité routière ayant pour fonction la récupération des données (vitesse, régime moteur et taux d'alcoolémie), la vérification de leur conformité et, en cas de dépassement de certaines limites, l'envoi d'un rapport chargé des informations du véhicule (matricule, vitesse, régime moteur...etc.) vers une station de police, et ce via une communication sans fil basée sur le protocole « zigbee ».

Grace à ce système les différents facteurs qui causent les accidents peuvent être contrôlés par les autorités classiques et cela aura un effet positif sur les conducteurs qui seront obligés à appliquer le code de la route d'une façon stricte.

Ce projet nous a permis de participer à une expérience intéressante qui nous a renforcés et enrichis nos connaissances et ce à travers :

- l'acquisition d'une précieuse expérience relative au fonctionnement d'un véhicule.
- l'amélioration de nos connaissances sur le langage de programmation Java Android et le C.
- l'amélioration de nos connaissances en bases de données (SQLITE).
- l'approfondissement de nos connaissances sur les cartes de développement (arduino, bus can shield) et des composants électroniques (xbee pro s1, capteur MQ135).

En guise de perspectives, et malgré les diverses fonctionnalités qu'offre ce système, plusieurs travaux peuvent être envisagés pour poursuivre ce projet :

- Analyse du comportement du conducteur à partir du régime moteur qui permettra de développer un système de sécurité comme détection de fatigue et de stress (contrôle de vitesse...etc).
- Utilisation du même principe d'acquisition pour afficher d'autres paramètres liés au diagnostic automobile dans une application.
- Utilisation d'un décodeur VIN pour permettre une meilleure identification des véhicules.
- Utilisation de l'interface ELM 327 afin de supporter tous les protocoles OBD 2.

Pour conclure, on souhaite que ce modeste travail puisse répondre favorablement aux besoins des futurs utilisateurs et servir comme outil d'aide et de documentation pour les promotions à venir.

## **Bibliographie**

[1] Citation proposée : Rapport de situation sur la sécurité routière dans le monde : il est temps d'agir. Genève, Organisation mondiale de la Santé, 2009 disponible sur : [www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2009](http://www.who.int/violence_injury_prevention/road_safety_status/2009).

[2] LES HORAIRES ATYPIQUES disponible sur : [www.aismtcai.com/fichs/16108.pdf](http://www.aismtcai.com/fichs/16108.pdf)

[3] Relation entre la vitesse pratiquée et les accidents : Les modèles de Nilsson et Elvik disponible sur : <http://www.securiteroutiere.gouv.fr/content/download/33225/309812/version/1/file/2014+05+14+La+vitesse+et+les+accidents+-+Nilsson+et+Elvik.pdf>

[4] Statistiques - Algeriepolice disponible sur : <http://www.dgsn.dz/?ETAT-COMPARATIF-DES-ACCIDENTS,7538>

[5] ESP disponible sur : <http://www.autoservices.ch/mecanique/esp.htm>

[6]. INFORMATION TECHNIQUE disponible sur : <http://www.fae.es/files/product/pdf/info-tecnica-fran-cabs1-58.pdf>

[7] Dossier multiplexage disponible sur : [http://pedagogie2.ac-reunion.fr/MetiersMaintenanceAuto/autre/286\\_dossier\\_multiplexage.pdf](http://pedagogie2.ac-reunion.fr/MetiersMaintenanceAuto/autre/286_dossier_multiplexage.pdf)

[8] LABARBE David, Le multiplexage automobile, Numéro de révision 234 disponible sur : <http://www.geea.org/spip.php?article183>

[9] Le réseau CAN (Controller Area Network) Nicolas NAVET NRIA Lorraine disponible sur : <http://www.loria.fr/~nnavet>

[10] Gestion de bus CAN-protocole CAN Vincent Oberle disponible sur : [www.oberle.org/can-can.html](http://www.oberle.org/can-can.html)

[11] Gestion de bus CAN disponible sur : [www.oberle.org/can-rapport.pdf](http://www.oberle.org/can-rapport.pdf)

[12] OBD - Compteur Auto.Com disponible sur : <http://www.compteurauto.com/?page=seo&sp=obd>

[13] Modes et configuration de l'OBD (PID) - Outils OBD Facile disponible sur : <http://www.outilsobdfacile.fr/mode-et-pid-obd.php>

[14] Voitures connectées : les nouveaux défis du test électronique Par Bill Mckinley, Keysight Technologies et Matt Hodgetts, Microlease disponible sur : <http://www.actutem.com/voitures-connectees-les-nouveaux-defis-du-test-electronique/>

[15] Gartner, un cabinet américain spécialisé dans les analyses pour l'industrie informatique disponible sur : <https://www.bestpractices-si.fr/publications/etudes/gartner-big-data-source-d-emplois-it>

[16] RABIA Fatima Mélissa TAZIBT Celia Yasmine, « Déploiement d'un réseau de capteurs sans fil en technologie ZigBee », mémoire de master, 2015, UMMTO.

[17] ARDUINO MEGA ADK disponible sur :  
<https://www.arduino.cc/en/Main/ArduinoBoardMegaADK>

[18] CAN-BUS Shield - DEV-13262 - SparkFun Electronics disponible sur:  
<https://www.sparkfun.com/products/13262>

[19] MCP2515 - Stand-Alone CAN Controller with SPI™ Interface disponible sur :  
<http://ww1.microchip.com/downloads/en/DeviceDoc/21801d.pdf>

[20] Arduino Breathalyzer: Calibrating the MQ-3 Alcohol Sensor disponible sur:  
<http://nootropicdesign.com/projectlab/2010/09/17/arduino-breathalyzer/>

[21] Embedded Systems - Shape The World, Jonathan Valvano and Ramesh Yerraballi disponible sur <http://users.ece.utexas.edu/~valvano/Volume1/E-Book/>