

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

Département d'Informatique



Mémoire de fin d'étude

En vue de l'obtention du diplôme de Master Académique en informatique

Option : Réseaux, Mobilités et Systèmes Embarqués.

Thème

Conception et réalisation d'un système de diagnostic et géolocalisation d'un véhicule

Réalisé par:

HADJI Djamel.

TOUCHERIFT Hacene.

Dirigé par :

Mr LAGHROUCHE Mourad.

Promotion 2016/2017

Remerciements

Nous remercions avant tout notre Dieu le tout puissant, le miséricordieux pour la force et le bon esprit qui nous a donné pour réaliser ce travail.

Nous tenons à exprimer notre gratitude et nos remerciements au membre de jury. Veuillez accepter dans ce travail notre sincère respect et notre profonde reconnaissance.

Nous tenons également à remercier infiniment notre promoteur M. LAGHROUCHE Mourad pour nous avoir offert les conditions nécessaires et nous avoir guidés dans l'élaboration de ce travail et contribuer largement à sa réalisation avec la patience et le dynamisme qui le caractérise et aussi son soutien tout au long de notre projet.

Nous remercions également Mr DAOUI pour son accueil chaque fois que nous avons sollicité son aide, ainsi que pour ses multiples conseils.

Nous profitons aussi de ce mémoire pour exprimer nos plus vifs remerciements envers tous les professeurs qui nous ont apportés du soutien durant nos études et envers tous nos amis qui ont été toujours près de nous avec leurs encouragements, critiques et conseils.

Enfin, notre reconnaissance va à ceux qui ont plus particulièrement assuré le soutien affectif de ce travail : nos familles.

Dédicaces

Je dédie ce modeste travail

A mes chers parents

Pour leur soutien, leur patience, leur sacrifice

et leur amour, vous méritez tout éloge,

vous qui avez fait de moi ce que je suis maintenant.

J'espère être l'image que vous avez fait de

moi, que Dieu vous garde et vous bénisse.

Je dédie aussi ce travail à mon cher frère et ma chère sœur,

pour leur affection et leur encouragement qui ont toujours

été pour moi des plus précieux.

Que ce travail soit pour vous le gage de mon profond amour

A tous mes amis et mon binôme

A tous ceux qui nous ont aidés.

H.Djamel

Dédicaces

Je dédie ce modeste travail

A mes chers parents

Pour leur soutien, leur patience, leur sacrifice

et leur amour, vous méritez tout éloge,

vous qui avez fait de nous ce que nous sommes maintenant.

J'espère être l'image que vous avez fait de

moi, que dieu vous garde et vous bénisse.

Je dédie aussi ce travail à mes chers frères et sœurs,

pour leur affection et leur encouragement qui ont toujours

été pour moi des plus précieux.

Que ce travail soit pour vous le gage de notre profond amour

A tous mes amis et mon binôme

A tous ceux qui nous ont aidés.

T.Hacene

Sommaire

Introduction générale :.....	1
Chapitre 1 : Généralités sur l'électronique automobile.....	3
I. Introduction :.....	4
II. L'électronique automobile :	4
II.1. Les composants électroniques dans un véhicule :	4
II.2. Le multiplexage:	5
II.2.1. Définition du multiplexage :.....	5
II.2.2. Le multiplexage dans un véhicule:	5
II.2.3. Historique sur le multiplexage :.....	7
II.2.4. Les avantages du multiplexage :.....	8
II.2.5. Adaptation des calculateurs :.....	8
II.2.6. Les types du réseaux multiplexés:.....	9
II.3. Le bus CAN (Controller Area Network):.....	10
II.3.1. Introduction:.....	10
II.3.2. Définition du bus CAN:	10
II.3.3. Les signaux du bus CAN :	11
II.3.4. Caractéristiques du bus CAN :	12
II.3.5. Les trames dans CAN:.....	15
II.4. La protocole OBD :.....	22
II.4.1. Définition :	22
II.4.2. Historique de l'OBD :.....	22
II.4.3. Protocoles de communication OBD-II :	23
II.4.4. Le connecteur OBD-II:.....	23
II.4.5. Modes de diagnostic de l'OBD-II :	24
III. Conclusion :	25
Chapitre 2 : La connectivité automobile et la géolocalisation.....	26
I. Introduction :.....	27
II. La connectivité automobile :	27
II.1. Véhicule connecté :	27
II.2. Technologies de communication automobile:.....	28
II.3. Utilisateurs du véhicule :	28
II.4. Les protocoles sans fil automobile :	29
II.4.1. Le Bluetooth :	29
II.4.2. Normes Bluetooth :	30
II.4.3. Mise en œuvre (communication Bluetooth) :	30
III. La géolocalisation :	31

III.1. Présentation de la géolocalisation :	31
III.2. Historique de la géolocalisation :	31
III.3. Les techniques de géolocalisation :	32
III.3.1. La géolocalisation par GSM :	32
III.3.2. La géolocalisation par satellite GPS:.....	33
III.3.2.1. Introduction:.....	33
III.3.2.2. Les segments du GPS :.....	33
III.3.2.3. Le fonctionnement d'un GPS :.....	34
IV. Autres systèmes de traçage :	36
V. Conclusion :	36
Chapitre 3: Conception du système de surveillance et diagnostic	37
I. Introduction :	38
II. Objectif du système :	38
III. Structure globale du système :	38
IV. Outils nécessaires pour notre système:	39
V. Implémentation matérielle:	47
VI. Conception :	51
VI.1. Introduction :	51
VI. 2. Conception Matérielle :	52
VI.2.1. Les graphes d'appels :	52
VI.2. 2. Le graphe de Raffinement successives :	53
VI .3. Conception Logicielle :	56
VII. Conclusion :	61
Chapitre 4: Realisations et tests.....	370
I. Introduction :	63
II. Principe de fonctionnement du système :	63
III. Les interfaces IHM de notre application :	64
IV. Les tests :	67
V. Conclusion :	73
Conclusion générale	74
Bibliographie :	76
Annexe :	77

Liste des figures :

Figure I. 1 : Exemple de câblage classique sans multiplexage.....	5
Figure I. 2 : exemple du câblage avec multiplexage.	6
Figure I. 3 : exemple de la vitesse du véhicule non multiplexée.....	6
Figure I. 4 : exemple de la vitesse du véhicule multiplexée.....	7
Figure I. 5 : Boîtier multiplexé.....	8
Figure I. 6 : Le réseau CAN automobile.	10
Figure I. 7 : Debit max 1Mbits/s.	11
Figure I. 8 : Debit max Mbits/s.	11
Figure I. 9 : le codage NRZ.....	12
Figure I. 10 : la technique Bit Stuffing.....	13
Figure I. 11 : Le Bit Timing.	13
Figure I. 12 : Exemple de bit timing.....	14
Figure I. 13 : Trame de données.....	15
Figure I. 14 : Champ d'arbitrage.....	16
Figure I. 15 : Champ de contrôle.....	16
Figure I. 16 : Champ de données.....	17
Figure I. 17 : Champ de CRC.....	18
Figure I. 18 : Champ d'acquiescement.	18
Figure I.19: Trame de requête.....	18
Figure I. 20 : Trame de gestion d'erreurs.....	19
Figure I. 21 : Trame d'erreur active.....	19
Figure I. 22 : Trame d'erreur passive.....	20
Figure I. 23 : Trame de surcharge.	20
Figure I. 24 : Période d'inter-trame.	21
Figure I. 25 : Exemple d'arbitrage au niveau champ d'arbitrage.....	22
Figure I. 26 : Le connecteur OBD-II.....	23
Figure II. 1: Les protocoles sans fil automobile.....	29
Figure II. 2 : cas d'un seul satellite.....	35
Figure II. 3 : cas de deux satellites.....	35
Figure II. 4 : cas de trois satellites.....	35
Figure II. 5: le positionnement à l'aide des trois satellites.....	36
Figure III. 1: Structure générale de notre système.....	38
Figure III. 2 : La carte Arduino Mega 2560.....	40
Figure III. 3 : Le CAN-Bus Shield.....	40
Figure III. 4 : Le câble OBD-II à DB9.....	41
Figure III. 5 : Le module SIM908.....	42
Figure III. 6: Bluetooth HC-05.....	42
Figure III. 7 : ELM 327 Bluetooth.....	43
Figure III. 8 : La manière schématique sur de fonctionnement de l'interface ELM 327.....	44
Figure III. 9 : Le logo du SDK Android.....	45
Figure III. 10 : Le logo d'IDE Android Studio.....	46
Figure III. 11 : Un Smartphone.....	46
Figure III. 12: Le schéma de brochage entre Arduino et Bus CAN Shield.....	48
Figure III. 13 : Brochage Arduino Bluetooth HC-05.....	49
Figure III. 14 : Le schéma de brochage entre Arduino et SIM 908.....	50
Figure III. 15 : le graphe général pour concevoir un système embarqué.....	51

Figure III. 16 : le graphe d'appels pour le système embarqué dans le véhicule.....	52
Figure III. 17 : Décomposition d'une tâche en sous tâches.	53
Figure III. 18 : Le graphe de Raffinement successives pour le système embarqué dans le véhicule..	54
Figure III. 19 : Le graphe de Raffinement successives pour le système embarqué concernant la..... géolocalisation.....	55
Figure III. 20 : Méthodologie de conception.....	56
Figure III. 21 : le diagramme des cas d'utilisation de notre système.	57
Figure III. 22 : Diagramme de séquence du cas d'utilisation « démarrage du système ».	57
Figure III. 23 Diagramme de séquence du cas d'utilisation « Demande d'informations en temps réel ».	58
Figure III. 24 : Diagramme de séquence du cas d'utilisation Demande de la position du véhicule.....	59
Figure III. 25 : Diagramme de classes du système.	61
Figure IV 1 Principe de fonctionnement du système.	63
Figure IV 2 Interface principale de l'application.	64
Figure IV 3 Les options à choisir, L'autorisation de connectivité Bluetooth et la visualisation des informations temps réel.....	65
Figure IV 4: Enregistrement des données en temps réel.	65
Figure IV 5 SMS contenant longitude et latitude.....	67
Figure IV 6 : Interface de géolocalisation Google Map.	67
Figure IV 7 : Volkswagen Passat.	68
Figure IV 8 : Résultats obtenus lors du test.....	69
Figure IV 9 : Renault Symbole.....	70
Figure IV 10 : Hyundai Accent.	71
Figure IV 11 : Résultats du test final avec l'onglet des choix.....	72
Figure IV 12 : Résultats du test final.....	72
Liste des tableaux :	
Tableau 1 : les différences entre le CAN Low speed et le CAN High speed.....	12
Tableau 2 Longueur du Bus et Débit.....	14
Tableau 3 Nombre d'octets qui seront obtenus dans le champ de données en fonction des bits DLC..	17
Tableau 4 : alimentation du Bus CAN Shield par Arduino mega2560.	47
Tableau 5: La communication entre le Bus CAN Shield avec Arduino mega2560.	47
Tableau 6: alimentation du HC-05 par Arduino mega 2560.	48
Tableau 7: La communication du HC-05avec Arduino mega 2560.....	48
Tableau 8: L'alimentation du SIM 908 par Arduino mega 2560.....	49
Tableau 9: La communication entre SIM 908 avec Arduino mega 2560.....	49
Tableau 10: Informations sur le véhicule du teste N°1.	68
Tableau 11: Informations sur le véhicule du Test N°2.....	70
Tableau 12: Information sur le véhicule du Test N°3.	71

INTRODUCTION GENERALE

Introduction générale :

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

L'automobile est l'un des plus importants domaines d'application des systèmes embarqués et actuellement, plus de 30% de la valeur d'une voiture est composée de systèmes électroniques embarqués.

L'automobile s'est ainsi progressivement imposée dans les pays développés comme le principal moyen 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...).

Cependant l'inconvénient majeur est l'accident qu'elle peut provoquer à tout moment, qui est généralement causé par plusieurs facteurs tels que le manque de diagnostic ...etc.

L'outil indispensable dans tout véhicule est l'outil de géolocalisation, le plus connu est appelé «GPS», celui-ci est doté d'un récepteur qui permet au conducteur de se localiser et l'aider à s'orienter qui permettra au propriétaire de connaître l'emplacement de son véhicule, en cas de vol. Il peut aussi voir les déplacements de son véhicule en temps réel.

De nos jours, les systèmes de sécurité automobile contribuent à produire des véhicules plus sûrs, mais restent coûteux.

L'objectif de notre travail consiste à réaliser un système pour récupérer des données d'un véhicule (tel que la vitesse, régime moteur,etc.), en temps réel. Ce système va contribuer à renforcer la sécurité par des mesures dissuasives envers les conducteurs qui ne se conforment pas à l'état du véhicule (faire un diagnostic au véhicule) ainsi notre système nous permettra de concevoir et développer un localisateur de véhicule. C'est un système utilisant un récepteur GPS pour la localisation et un module GSM pour la transmission de la position à l'aide des SMS. Ce localisateur va permettre à son utilisateur, muni d'un smartphone, de repérer son véhicule volé et voir sa position sur une carte.

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** : Généralités sur l'électronique automobile, dans lequel nous abordons le fonctionnement et l'architecture électronique dans un véhicule et le diagnostic automobile.
- ✓ **Chapitre II** : La connectivité automobile et géolocalisation, dans lequel nous étudions les différents protocoles de communication sans fil utilisées dans l'automobile et nous présentons le Bluetooth et il expliquera le principe de la géolocalisation GPS/GSM.

- ✓ **Chapitre III** : Conception du système de diagnostic et géolocalisation, 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, Nous présentons quelques interfaces graphiques constituant l'application et des résultats des tests pratiques.

Chapitre I:

Généralités sur l'électronique automobile

I. Introduction :

Le transport routier présente des avantages ou 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 le fonctionnement de l'électronique générale automobile, notamment le multiplexage dans le véhicule et le bus de communication le plus connu qui est le bus CAN et se terminera par l'étude de diagnostic automobile OBD.

II. L'électronique automobile :

Pratiquement il n'existe aucun domaine de l'automobile qui n'échappe à l'électronique. Le nombre de composants électroniques présents ne cesse d'augmenter et en font des véhicules de hautes-technologies – plus sûrs, plus confortables et plus respectueux de l'environnement. En 2015, l'électronique représentera environ 40 % de la valeur d'une automobile (30 % en 2010).

II.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 se qu'il 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...

II.2. Le multiplexage:

II.2.1. Définition du multiplexage :

Le multiplexage est une technique qui consiste à faire passer plusieurs informations à travers un seul support de transmission. Elle permet de partager une même ressource entre plusieurs utilisateurs [1].

II.2.2. Le multiplexage dans un véhicule:

L'augmentation du nombre de calculateurs et de capteurs dans les automobiles entraîne une hausse du nombre de connexions et de longueur de câbles (et donc de pannes). La multiplication de ces fonctions électroniques aujourd'hui nécessite un moyen robuste, fiable et économique pour échanger en temps réel des informations sur des distances moyennes ; les nombreux câbles nécessaires doivent, en outre, être contenus dans un espace très réduit. Cette dernière solution ne peut être que le multiplexage.

❖ Un simple exemple de « câblage classique (avant le multiplexage) » :

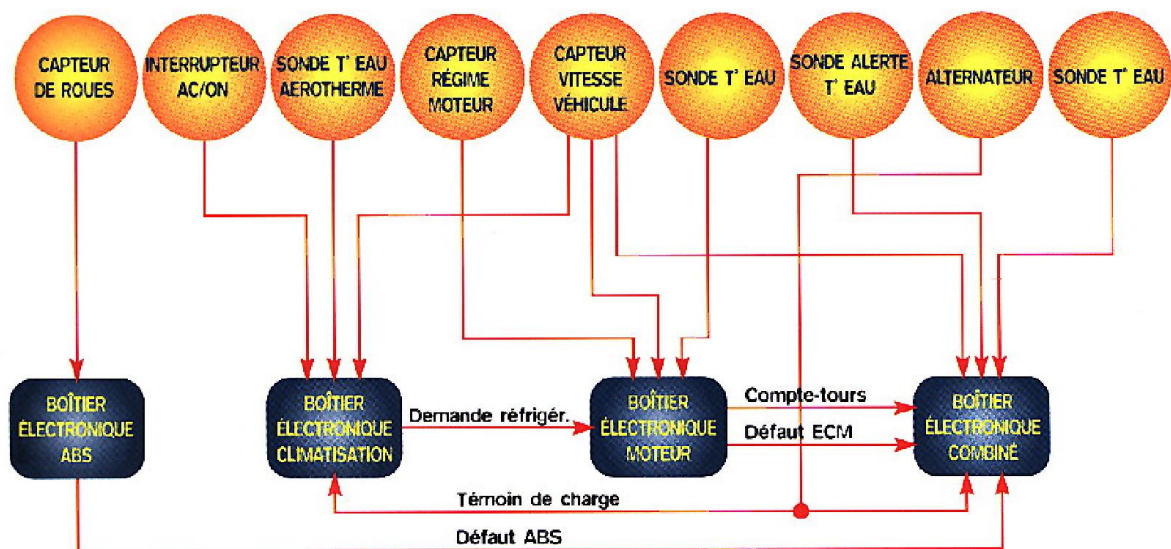


Figure I. 1 : Exemple de câblage classique sans multiplexage.

- Certains capteurs ont des liaisons avec plusieurs calculateurs ou existent en deux exemplaires en raison de leur localisation.
- Les liaisons entre boîtiers sont de plus en plus nombreuses.

❖ Un simple exemple après multiplexage :

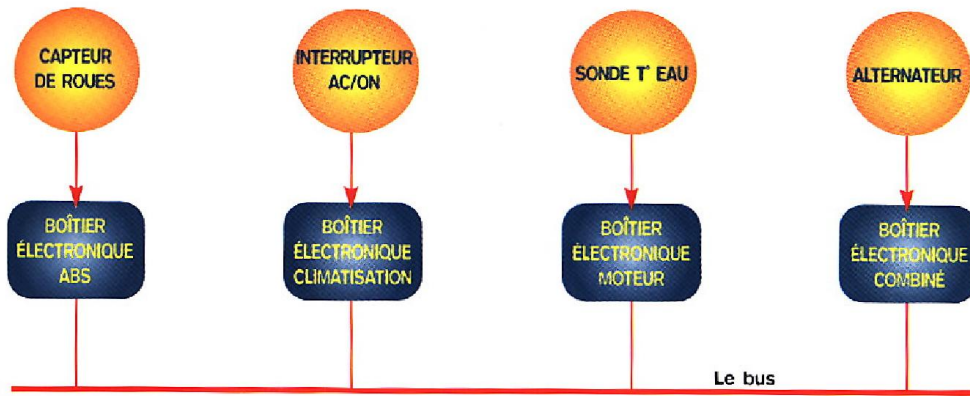


Figure I. 2 : exemple du câblage avec multiplexage.

- Diminution du nombre de capteurs et de liaisons entre boîtiers car chacun fournit aux autres, par l'intermédiaire du bus, les informations qu'il reçoit dans une liaison filaire : c'est le partage des trames.

A titre exemple :

✚ Sur une voiture non multiplexée, les informations circulent comme cela:

Si on prend l'information **vitesse** du véhicule, elle est fournie par un capteur de vitesse sur la boîte qui doit donner l'information au calculateur d'injection, au calculateur d'ABS, au bloc compteur au calculateur de climatisation etc. ...

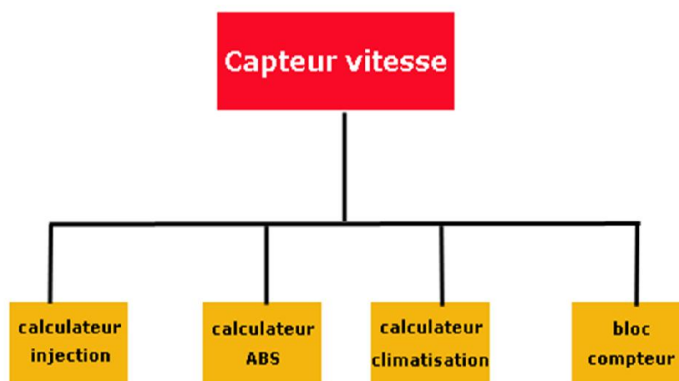


Figure I. 3 : Exemple la vitesse du véhicule non multiplexée.

Le capteur donne sa valeur à chaque calculateur, grâce à des liaisons filaires, impliquant de grandes longueurs de câbles.

✚ Sur un véhicule multiplexé :

Le calculateur d'ABS fournit l'information sur le réseau multiplexé aux différents calculateurs qui en ont besoin comme la vitesse (la Figure I.4 illustre un exemple).

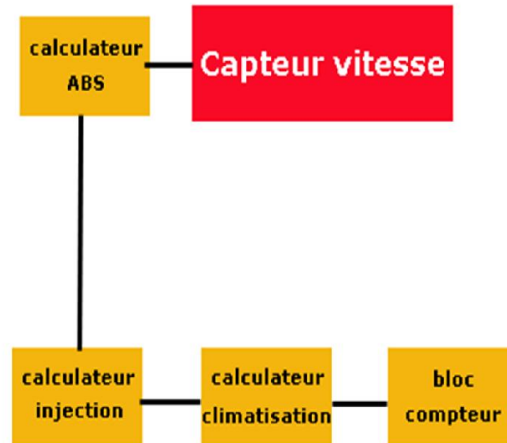


Figure I. 4 : Exemple la vitesse du véhicule multiplexée.

II.2.3. Historique sur le multiplexage :

Actuellement, tous les véhicules comportent, au moins un réseau multiplexé.

Le développement du multiplexage dans le véhicule est passé par des étapes suivantes :

- En 1983 Bosch dépose, pour l'industrie, un brevet d'un Réseau Local de Contrôle appelé CAN (Controller Area Network)
- En 1986, PSA après un travail avec Renault, dépose à la norme le Réseau Local Véhicule (VAN : Véhicule Area Network)
- En 1989, PSA teste ces premiers composants multiplexés
- En 1992, BMW avec la 850i et Mercedes avec sa Classe S teste réellement le CAN
- En 1994, 1400 Citroën XM Pallas sortent équipées du tableau de bord entièrement multiplexé en VAN
- En 1994, Audi A4 avec gestion moteur multiplexée
- En 1998, fabrication en série sur la 206 (VAN : autoradio, chargeur cd, navigation, écran multifonction)
- En 1999, fabrication en série sur la 406 et la Xsara Picasso.

II.2.4. Les avantages du multiplexage :

- Moins de capteurs et/ou de nombres de liaisons avec les boîtiers.
- Le poids et les coûts diminuent.
- Enrichissement de fonctions sans surcoût important :
 - ✓ Faire allumer les feux de croisement lorsque le capteur de pluie détecte une averse (évolution d'un logiciel)
 - ✓ Mise en action des feux de détresse lors d'une forte décélération
- Les méthodes répondent à une norme ISO donc fiabilité accrues (théoriquement)
- Les constructeurs « protègent » leur réseau d'APV car la plupart des interventions sur les systèmes multiplexés nécessitent l'utilisation d'outils de diagnostic particuliers :
 - ✓ Méthodes de recherche de pannes complexes + téléassistance.
 - ✓ Téléchargement de mise à jour.
 - ✓ Apprentissage lors d'une installation ou d'un changement de composants multiplexés (ex : autoradio) : c'est le télé-codage.

II.2.5. Adaptation des calculateurs :

a. Boîtier multiplexé :

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

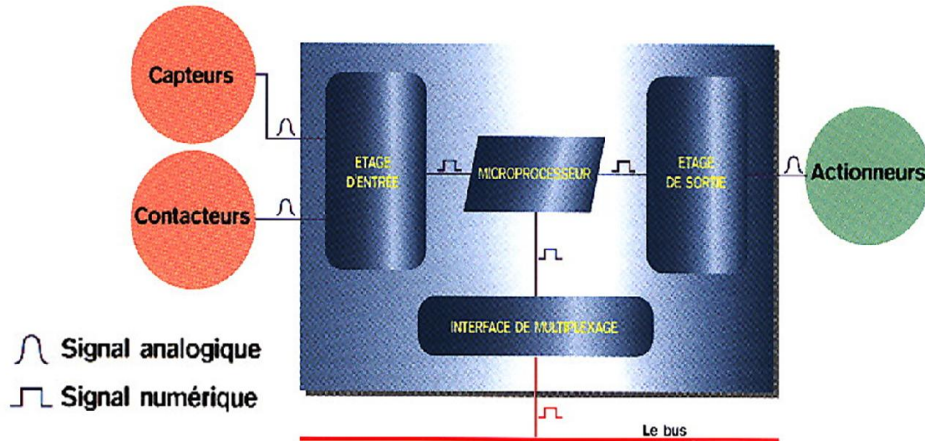


Figure I. 5 : Boîtier multiplexé

L'interface de multiplexage se charge des communications avec le bus.

- **L'étage d'entrée du boîtier** : Il transforme (**il code**) les signaux analogiques des capteurs en signaux numériques exploitables par le microprocesseur (Convertisseur Analogique Numérique CAN).
- **L'étage de sortie du boîtier** : Il transforme les ordres, fournis par le microprocesseur sous forme de signaux numériques, en signaux analogiques destinés aux actionneurs (Convertisseur Numérique Analogique CNA)

- **L'étage de calcul** : le microprocesseur, c'est le composant « intelligent » du boîtier. Il possède des mémoires qui peuvent être : ROM, RAM, EEPROM (mémoire morte pouvant être reprogrammées (de plus en plus utilisée).
- **L'interface de multiplexage** :
 - Elle permet la communication entre le boîtier et le bus CAN
 - Les messages qui transitent par l'interface de multiplexage sont numériques et portent le nom de **trames**
 - Ces trames sont découpées en plusieurs champs

b. Le signal : analogique ou numérique ?

Le signal analogique est l'image de ce qu'il mesure, son amplitude et parfois sa fréquence évolue avec le temps.

Le signal numérique est un signal codé qui utilise la numérotation binaire, c'est à dire qu'il ne peut prendre que 2 valeurs (ex : tension ou pas tension, lumière ou pas lumière).

II.2.6. Les types des réseaux multiplexés:

Trois types principaux des réseaux multiplexés existent sur le marché : **le CAN** (Control Area Network), **le VAN** (Véhicule Area Network), **le J 18/50 [2]** :

- Le CAN est un réseau à haute vitesse (500 à 1 000 kilobits par seconde) développé à l'origine par Bosch et les constructeurs allemands. Ce dernier convient pour des échanges rapides : moteur et sécurité,
- Le VAN offre un gros avantage par rapport au CAN. Lorsqu'un émetteur envoie une information, seul le récepteur concerné l'acquiesce, alors que, avec le CAN, tous les sous-ensembles acquiescent le message, même s'il ne leur est pas destiné. Le VAN fonctionne également très bien en mode dégradé. Le VAN s'adapte bien aux équipements de confort et de carrosserie.
- Le J 18/50, d'origine américaine, est un protocole à faible vitesse (41,6 kilobits par seconde) pour la carrosserie (vitres, alarmes, rétroviseurs, etc.)

II.3. Le bus CAN (Controller Area Network):

II.3.1. Introduction:

Depuis les années 1960 la longueur de câbles utilisés dans l'automobile ne cesse de croître pour dépasser 2000 m en 1995. Le nombre des connexions atteint 1800 à cette même date, donc la fiabilité et la sécurité sont menacés.

Les normes en matière de pollution et de consommation d'énergie obligent les constructeurs à multiplier les capteurs et actionneurs intelligents dans leurs véhicules accélérant ce processus de multiplication des câbles et connexion depuis une vingtaine d'années.

Le besoin de sécurité accrue (ABS, ESP, AIR-BAG...) et la demande de confort (mémoire des réglages de conduite, climatisation régulée par passager, système de navigation...) ne font que renforcer cette tendance.

La société BOSCH développe dès le début des années 1980 une solution de multiplexage des informations circulant à bord de la voiture. Le bus CAN apparaîtra et sera normalisé dans les années qui suivent (dès 1983).

Les composants CAN se démocratisent et investissent d'autres secteurs de l'électronique embarqué (médical, produits numériques, systèmes électrotechnique...) [3]

II.3.2. Définition du bus CAN:

Le bus CAN (*Control Area Network*) est un bus de communication série, principalement conçu pour mettre en réseau diverses cartes électroniques.

Il est utilisé dans l'automobile pour faire dialoguer les divers systèmes de contrôle ou de commande : ABS, système de freinage, de suspension, etc...

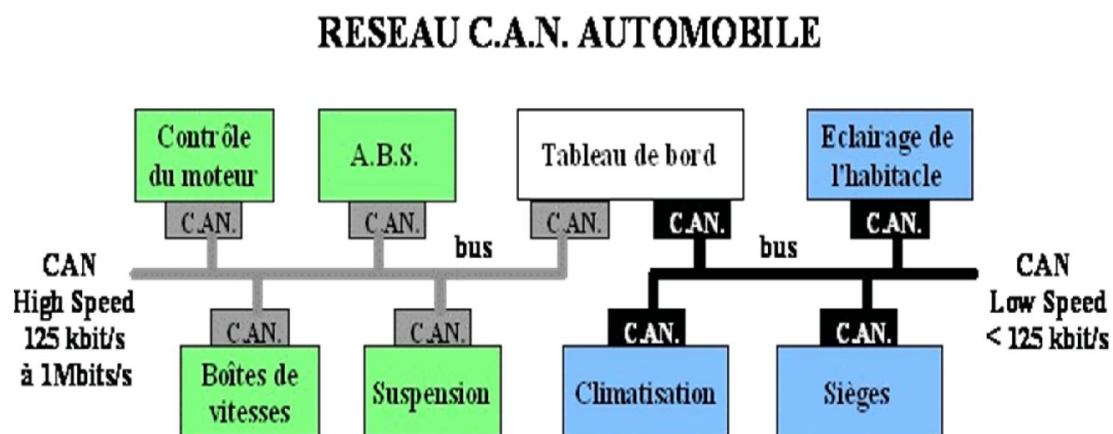


Figure I. 6 : Le réseau CAN automobile.

En pratique, il y a trois bus CAN différents dans une voiture, à des débits différents :

- un bus très rapide pour gérer la sécurité (freinage, ABS, détection chocs, airbags...);
- un bus à vitesse moyenne pour gérer le moteur (commandes et capteurs);
- un bus lent pour gérer tous les accessoires (lampes, boutons, ...etc.);

La structure du protocole du bus CAN possède implicitement les principales propriétés suivantes :

- hiérarchisation des messages.
- garantie des temps de latence.
- souplesse de configuration.
- réception de multiples sources avec synchronisation temporelle.
- fonctionnement multi-maîtres.
- détections et signalisations d'erreurs.
- retransmission automatique des messages altérés dès que le bus est de nouveau au repos.
- distinction d'erreurs : d'ordre temporaire ou de non-fonctionnalité permanente au niveau d'un nœud, déconnexion automatique des nœuds défectueux.

II.3.3. Les signaux du bus CAN:

La transmission des données est effectuée sur une paire filaire différentielle. La ligne est donc constituée de deux fils. Et les niveaux logiques (récessifs et dominants) sont obtenus par la différence de potentiel entre les deux voies CAN L et CAN H.

Les niveaux de tension sur CANL et CANH dépendent du type Low Speed ou High Speed du bus. Ces niveaux de tension correspondent à un codage dit NRZ (No Return to Zero : il n'y a jamais de courant nul sur la ligne. La masse n'est plus utilisée et les niveaux logiques correspondent à 2 niveaux de tensions distincts).

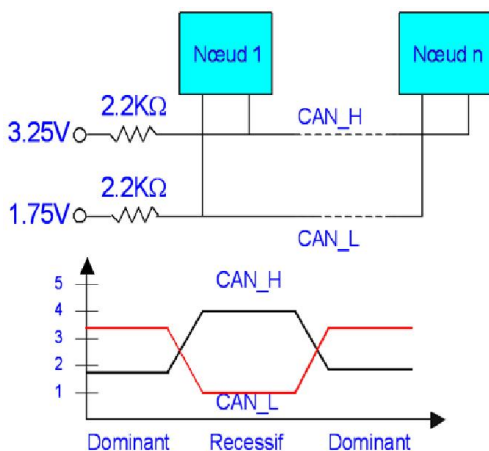


Figure I. 8 : Debit min ≤ 125 Kbps.

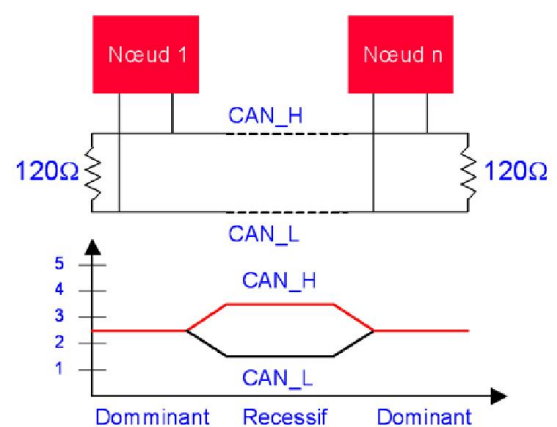


Figure I. 7 : Debit max 1Mbps

Le tableau ci-dessous résume les principales différences entre les deux types de bus notamment sur les débits supportés :

Paramètres	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
Courant de sortie (mode émission)	>1 mA sur 2,2 kohm	25 à 50 mA sur 60 ohm
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
Caractéristique du câble	30pF entre les câbles de lignes	2*120 (ohm)
Tensions d'alimentation	5V	5V

Tableau 1 : les différences entre le CAN Low speed et le CAN High speed.

II.3.4. Caractéristiques du bus CAN [3] :

a. Le codage NRZ : bits dominants et récessifs :

La succession de bits transitant sur le bus est codé avec la méthode du NRZ (Non Return To Zero). Pendant la durée totale du bit, le niveau de tension de la ligne est maintenu, c'est à dire que pendant toute la durée durant laquelle un bit est généré, sa valeur reste constante qu'elle soit dominante ou récessive.

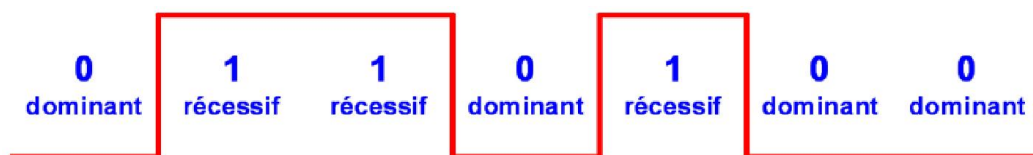


Figure I. 9 : le codage NRZ

b. Le bit stuffing:

Une des caractéristiques du codage NRZ est que le niveau du bit est maintenu pendant toute sa durée. Cela pose des problèmes de fiabilité si un grand nombre de bits identiques se succèdent. La technique du Bit Stuffing impose au transmetteur d'ajouter automatiquement un bit de valeur opposée lorsqu'il détecte 5 bits consécutifs dans les valeurs à transmettre.

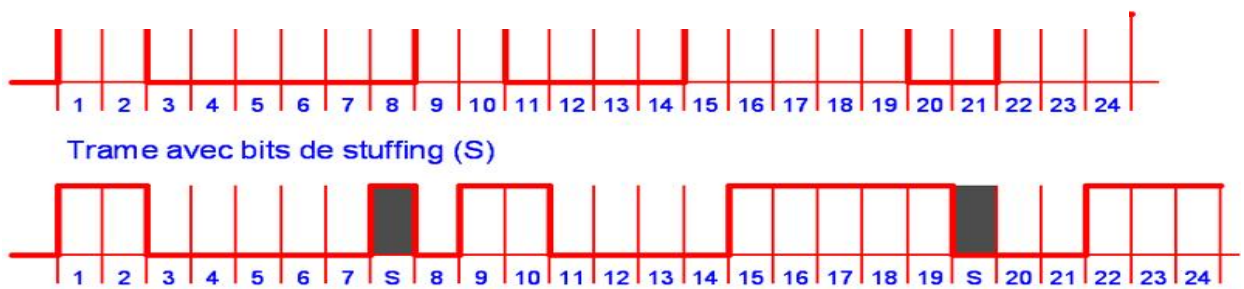


Figure I. 10 : la technique Bit Stuffing.

c. Le bit timing :

On définit la plus petite base de temps reconnue sur un bus CAN comme étant le *Time Quantum*. Cette base de temps est une fraction de l'horloge de l'oscillateur du bus. Un bit dure entre 8 et 25 quantum.

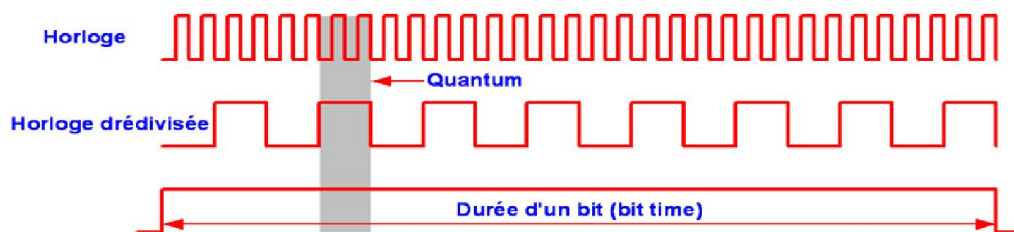


Figure I. 11 : Le Bit Timing.

Exemple de bit timing: lecture d'un bit ISO11898: High Speed CAN 250 Kbps

- 1 bit correspond à 32 coups d'horloge
- La lecture du bit devra être faite au 20ème coup d'horloge.

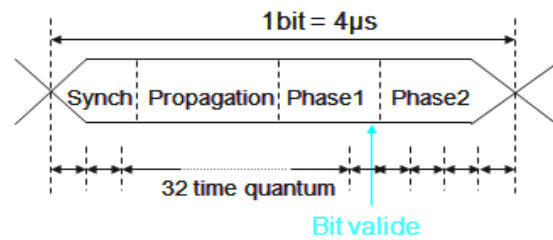


Figure I. 12 : Exemple de bit timing.

d. Longueur du bus et débit:

La longueur du bus dépend des paramètres suivants :

1. Le délai de propagation sur les lignes physiques du bus.
2. La différence du quantum de temps défini précédemment, du aux différences de cadencement des oscillations des nœuds.
3. L'amplitude du signal qui varie en fonction de la résistance du câble et de l'impédance d'entrée des nœuds.

Débit	Longueur	Longueur d'un bit
1 Mbit/s	30 m	1 µs
800 kbit/s	50 m	1,25 µs
500 kbit/s	100 m	2 µs
250 kbit/s	250 m	4 µs
125 kbit/s	500 m	8 µs
62,5 kbit/s	1000 m	16 µs
20 kbit/s	2500 m	50 µs
10 kbit/s	5000 m	100 µs

Tableau 2 Longueur du Bus et Débit.

Pour une longueur de bus supérieure à 200 mètres il est nécessaire d'utiliser un optocoupleur, et pour une longueur de bus supérieure à 1 kilomètre il est nécessaire d'utiliser des systèmes d'interconnexion tels que des répéteurs ou des ponts. N'importe quel module connecté sur un bus CAN doit pouvoir supporter un débit d'au moins 20 kbit/s.

II.3.5. Les trames dans CAN:

L'information sur le bus est envoyée sous la forme de trames au format fixé. Quand le bus est libre, n'importe quel nœud peut commencer à envoyer une trame.

En fait le protocole CAN 2.0 comporte deux sous-spécifications qui diffèrent au niveau de la longueur de l'ID :

- La version CAN 2.0A définit des ID de 11 bits appelée trame standard,
- La version CAN 2.0B définit des ID de 29 bits appelée trame étendue [4].

Dans le protocole CAN on trouve cinq types de trames qui sont :

- Trame de données.
- Trame de requête.
- Trame de gestion d'erreur.
- Trame de surcharge.
- Période d'inter-trame.

La structure générale d'une trame de données CAN avec la version CAN 2.0A est :

- ID (11 bits) correspond au champ d'arbitrage sans le bit RTR (les détails viennent juste dans la section suivante).
- Message (8 octets) correspond au champ de données.
- Plein d'autre chose pour contrôler les erreurs,

Voici un aperçu sur chaque trame citée précédemment avec la norme standard d'ID de 11 bits (CAN 2.0A) [3] :

a. Trame de données:

Une trame de données se décompose en 7 champs différents comme les montre la figure I.13:

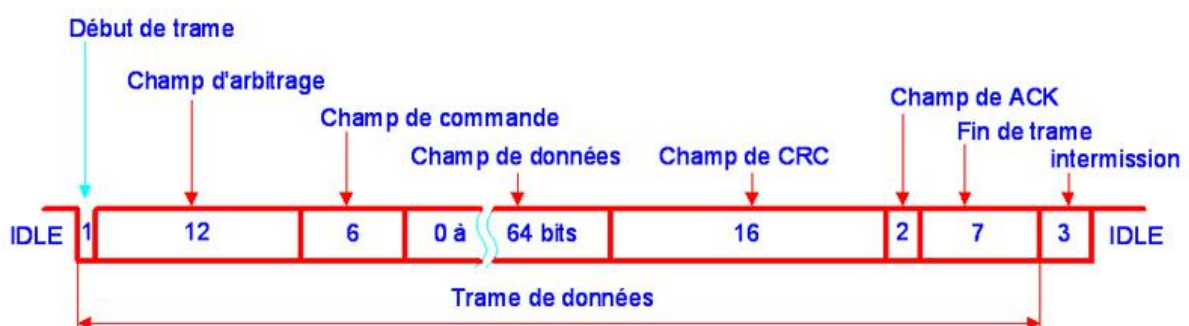
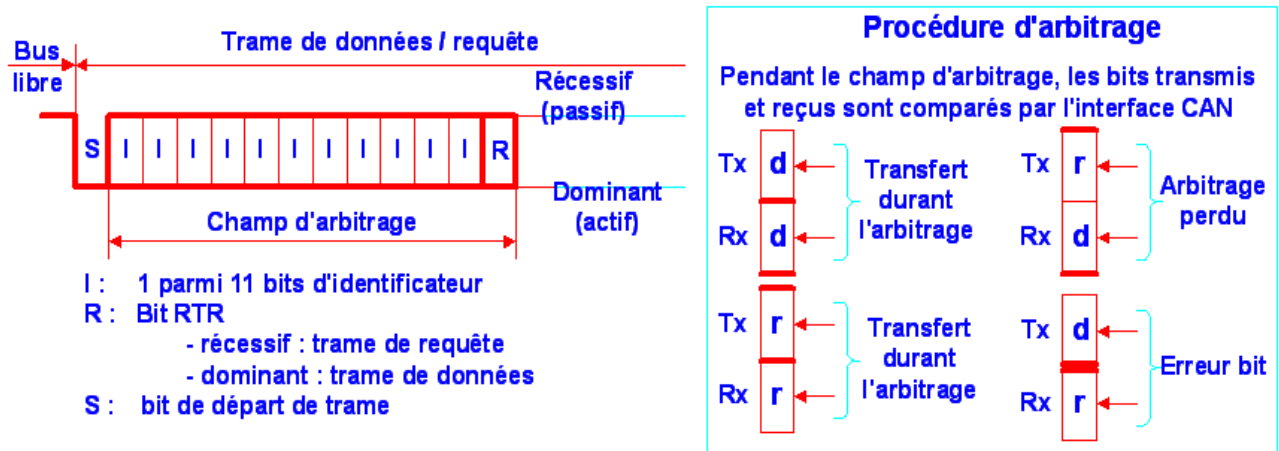


Figure I. 13 : Trame de données.

- **début de trame** : 1 bit à 0 (dominant).
- **champ d'arbitrage 12 bits (ou 30 bits)**: 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.

Figure I. 14 : Champ d'arbitrage.

- **champ de contrôle (commande) (6 bits)** : Il est constitué de 6 bits.

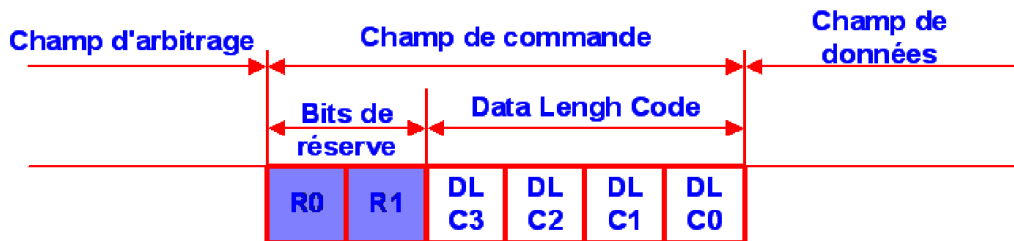


Figure I. 15 : Champ de contrôle.

Il y a *deux Bits de réserves* : Les deux premiers bits (émis dominants en trame 2.0A) sont en réserve d'usages ultérieurs et permettent d'assurer des compatibilités futures ascendantes (notamment celles de la trame dite étendue CAN 2.0B). Les contrôleurs CAN doivent être aptes à traiter toutes combinaisons de tous les bits du champ de commande.

4 bits DLC : Les 4 derniers bits du champ de commande (champ DLC - *Data Length Code*) indiquent le nombre d'octets qui seront contenus dans le champ de données

Nombre d'octet	DLC 3	DLC 2	DLC 1	DLC 0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

Tableau 3 : Nombre d'octets qui seront obtenus dans le champ de données en fonction des bits DLC

Tel que le bit **d** signifie dominant et le bit **r** signifie récessif.

- **champ de données (0 à 8 octets)** : Le champ de données est l'endroit où se trouvent les données utiles transmises. Il peut être composé de 0 octet minimum à 8 octets maximum transmis avec le MSB (*Most Significant Bit*) en tête. De 0 à 8 inclus, cela fait 9 valeurs donc 4 bits du DLC pour définir le nombre de données contenues.

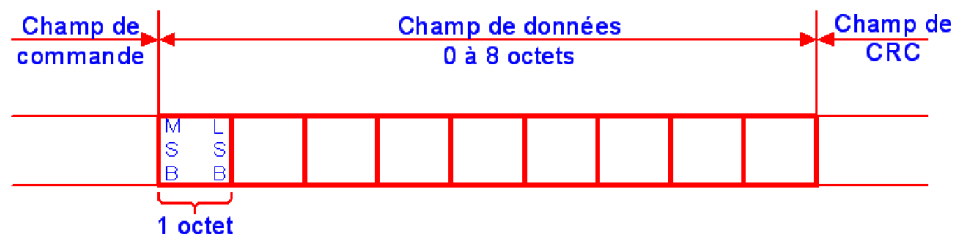


Figure I. 16: Champ de données.

- **champ de CRC (16 bits)** : 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.

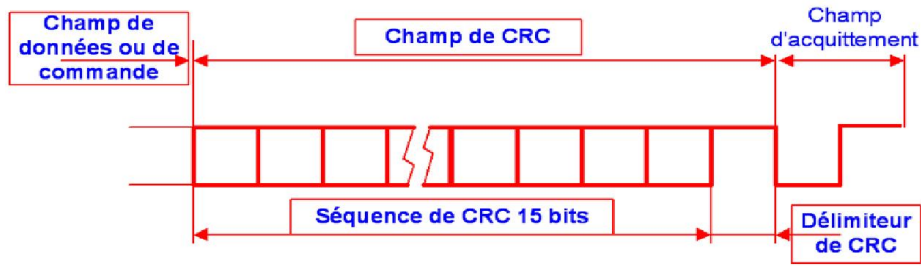


Figure I. 17: Champ de CRC.

- champ d'acquittement (2 bits) :** Ce champ d'acquittement est composé de 2 bits :
 - Un bit d'acquittement (l'ACK Slot) est dominant si le calcul du code de vérification des données est correct, sinon en cas d'erreur le bit est récessif.
 - Un bit délimiteur d'acquittement, est toujours à l'état haut 1 (récessif).

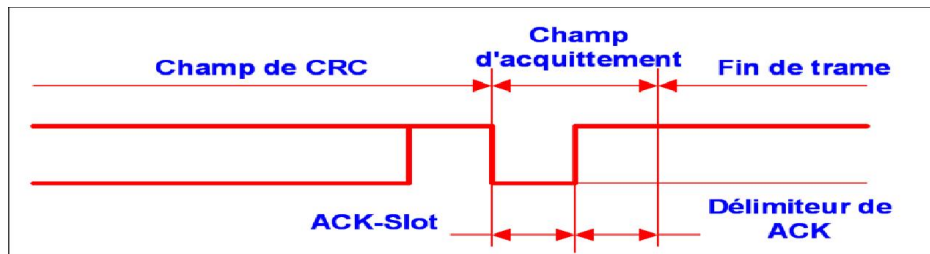


Figure I. 18: Champ d'acquittement.

- fin de trame (7 bits) :** La trame de donnée se termine par un drapeau formé par une séquence de 7 bits récessifs.

b. Trame de requête:

Une trame de requête est constituée de la même manière qu'une trame de données sauf que le champ de données est vide.



Figure I.19: Trame de requête.

c. La trame de gestion d'erreurs :

Une trame d'erreur permet de signaler aux autres nœuds CAN la présence d'une erreur (Passive ou Active).

La trame d'erreur est constituée de deux champs principaux : le drapeau d'erreur (06 bits dominants/récessifs) et le délimiteur de champ (08 bits récessifs).

Ci-dessous voici la façon dont se construit une trame d'erreur :

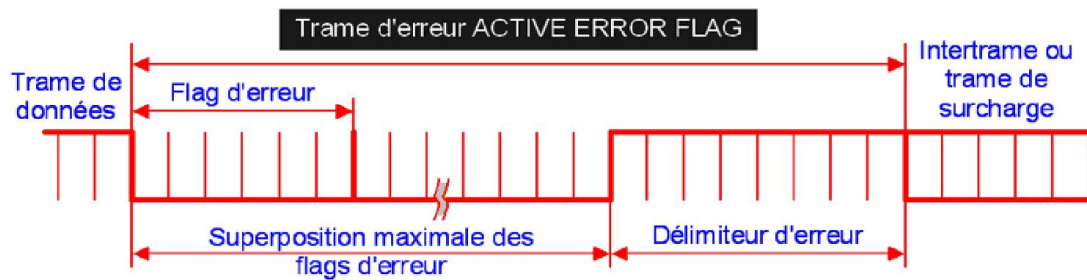


Figure I. 20 : Trame de gestion d'erreurs.

Donc on a deux sortes de trames d'erreur:

✓ La trame d'erreur active:

La trame d'erreur active est formée de six (06) bits dominants consécutifs pour le drapeau (Flag error active) suivi de huit (08) bits récessifs pour le délimiteur (Error Delimiter).

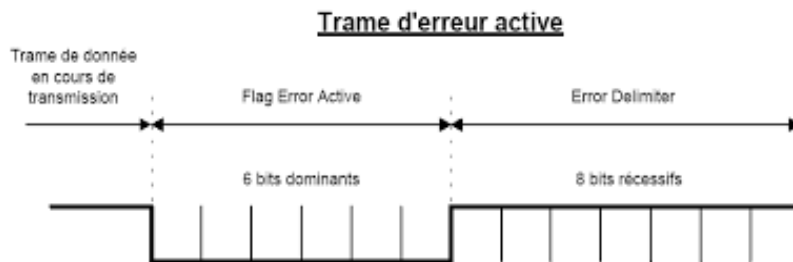


Figure I. 21 : Trame d'erreur active.

✓ La trame d'erreur passive:

La trame d'erreur passive est formée de six bits récessifs pour le drapeau (Flag error active) et de huit bits récessifs pour le délimiteur (Error Delimiter).

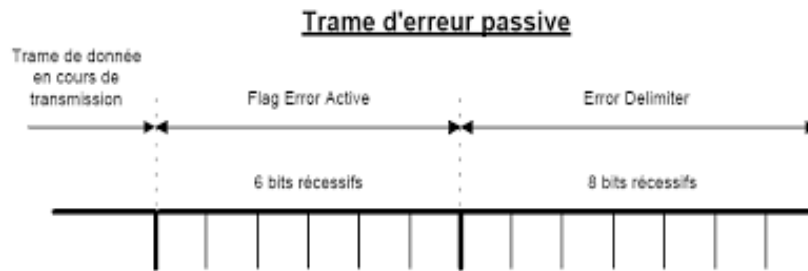


Figure I. 22: Trame d'erreur passive.

d. Trame de surcharge:

Dans le but de ne pas encombrer le bus indéfiniment, seules deux trames de surcharge peuvent être générées de manière consécutive. Elle se compose de deux champs : le champ des flags de surcharge et le champ délimiteur de surcharge.

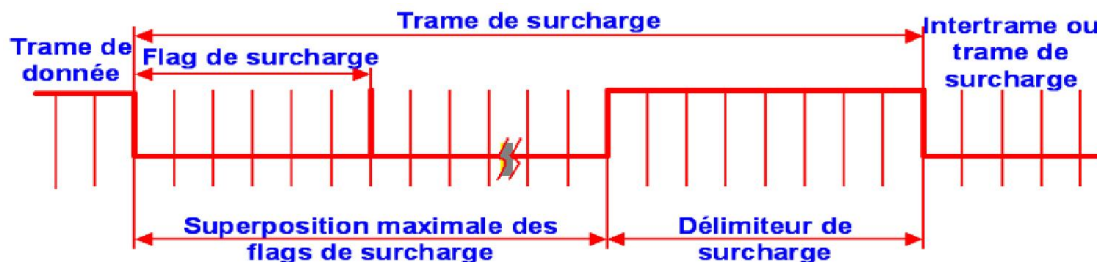


Figure I.23: Trame de surcharge.

Le champ des flags de surcharges est composé de 6 bits dominants consécutifs. Il détruit le champ intermission de l'intertrame.

Le champ délimiteur de surcharge est composé de 8 bits récessifs consécutifs. Après le passage d'un flag de surcharge, l'entité examine le bus jusqu'à ce qu'elle détecte une transition qui indique le passage d'un bit dominant à un bit récessif. À ce moment-là, chaque entité sur le bus a terminé d'envoyer son flag de surcharge et elles envoient donc 7 bits récessifs consécutifs pour former le délimiteur de surcharge.

e. Période d'inter-trame:

Les trames de données et de requêtes sont séparées des autres trames par une inter-trame, alors que les trames d'erreur et de surcharges ne sont pas séparées par cette inter-trame. Celui-ci se compose de 2 ou 3 champs selon les cas : le champ intermission, le champ bus libre et pour les unités qui ont été émettrices du message précédent, le champ suspension transmission.

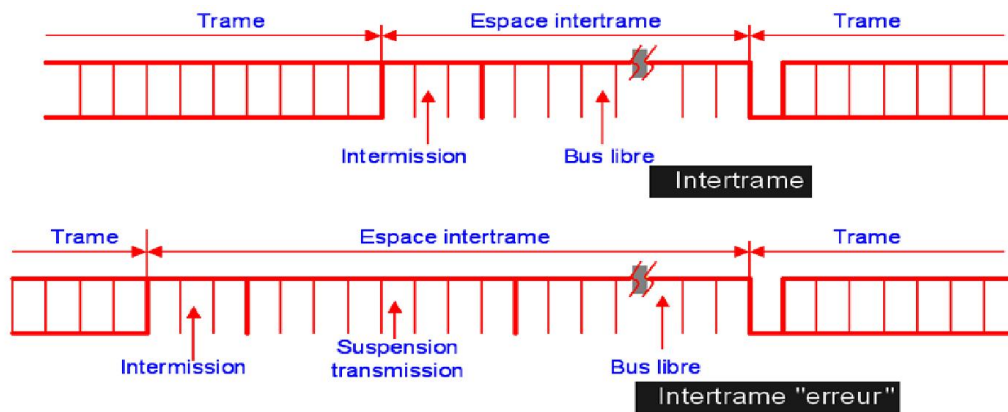


Figure I. 24: Période d'inter-trame.

Le champ intermission est composé de 3 bits récessifs. Pendant l'intermission, aucune trame de type données ou requêtes n'est autorisée à démarrer, seule la trame de surcharge peut circuler.

La durée du champ bus libre peut être arbitrairement choisie. Pendant ce laps de temps, la ligne est « libre » et n'importe quelle entité peut accéder au bus. Si un message était en attente lors de la précédente transmission, alors celui-ci pourra démarrer dès le premier bit suivant l'intermission. Ce bit sera interprété comme le SOF (Start Of Frame) de la nouvelle transmission.

Après qu'une entité avec un statut « Error passive » ai transmis un message, elle envoie le champ suspension transmission composé de 8 bits récessifs avant de démarrer une nouvelle transmission ou de reconnaître que le bus est « libre ». Si pendant ce même laps de temps une transmission démarre (par une autre entité) alors la première entité deviendra réceptrice de ce message.

II.3.6. Principe de l'arbitrage:

Le procédé d'attribution du bus est basé sur le principe de "l'arbitrage bit à bit", selon lequel les nœuds en compétition, émettant simultanément sur le bus, comparent bit à bit l'identificateur de leur message avec celui des messages concurrents. Les stations de priorité moins élevée perdront la compétition face à celle qui a la priorité la plus élevée.

Les stations sont câblées sur le bus par le principe du "ET câblé". En cas de conflit c'est à dire émission simultanée, la valeur 0 écrase la valeur 1.

Dans l'exemple ci-dessus, trois stations émettent simultanément :

- la station 2 perd la compétition puis la station 1 ;
- seule la station 3 pourra transmettre.

Lors de l'arbitrage bit à bit, dès qu'une station émettrice se trouve en état récessif et détecte un état dominant, elle perd la compétition et arrête d'émettre. Tous les perdants deviennent automatiquement des récepteurs du message, et ne tentent à nouveau d'émettre que lorsque le bus se libère.

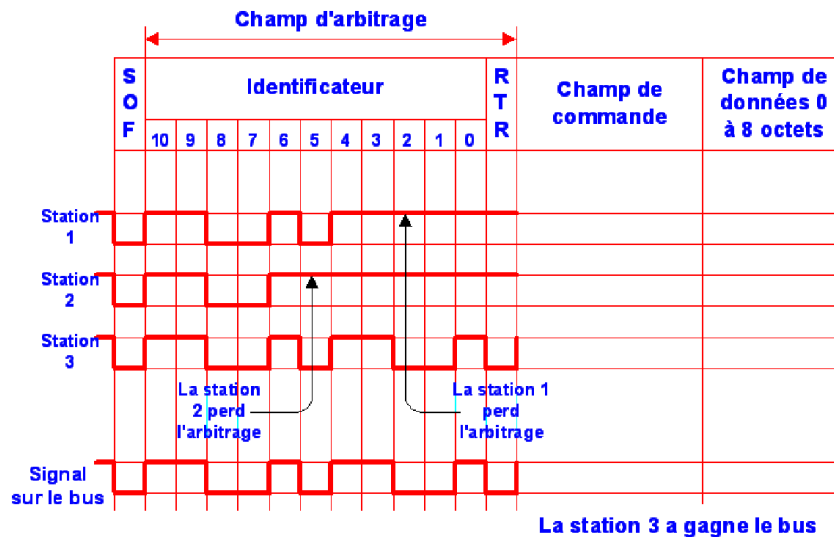


Figure I. 25: Exemple d'arbitrage au niveau champ d'arbitrage.

II.4. Le protocole OBD :

II.4.1. Définition :

L'OBD pour « On Board Diagnostic » est un système électronique intégré au calculateur moteur du véhicule et il surveille en permanence tous les composants et systèmes du véhicule, concernés par les gaz d'échappement.

Si un défaut apparaît, celui-ci est signalé au conducteur par l'intermédiaire d'un témoin de défaut MIL (témoin de défaut).

Simultanément, le défaut est enregistré dans la mémoire des défauts du calculateur moteur et peut être lu et effacé au moyen de tout appareil de diagnostic du commerce.

Cette prise est donc en connexion directe avec le calculateur moteur aussi appelé ECU (unité de contrôle électronique) [4].

II.4.2. Historique de l'OBD :

Le système **OBD** est né aux États-Unis, et plus précisément en Californie. Afin de réduire la pollution atmosphérique croissante générée par le trafic routier dans les mégapoles de Los Angeles et de San Francisco, les autorités compétentes ont instauré dès 1988 des valeurs limites d'émission de gaz plus strictes pour les véhicules essence, et ont exigé des automobilistes un contrôle supplémentaire de leurs véhicules. En 1994, tous les autres États des États-Unis ont mis cette réglementation en application. En 1996, **OBD-II** est entré en vigueur, et les valeurs limites d'émission ont baissé à nouveau. À partir de 2001, OBD-II a été en partie reprise en Europe sous l'appellation OBD ou **EOBD** [5].

II.4.3. Protocoles de communication OBD-II :

Un véhicule conforme à la **norme OBD-II** peut utiliser n'importe lequel des cinq **protocoles de communication**, 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.

II.4.4. Le connecteur OBD-II:

Le connecteur OBD-II 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.

- 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.

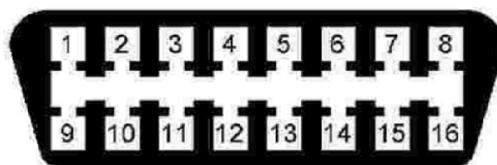


Figure I. 26: Le connecteur OBD-II

II.4.5. Modes de diagnostic de l'OBD-II :

Quel que soit le protocole de communication utilisé, la norme OBD définit **10 modes de diagnostic** décrits ci-dessous. Les constructeurs ne sont pas tenus de supporter tous ces modes. Chaque constructeur peut définir des modes additionnels au-dessus de #9 quand il intègre des fonctions supplémentaires dans le calculateur moteur et qu'elles doivent être aptes au diagnostic. Tous les véhicules ne vont pas supporter tous les **PIDs**, et il peut y avoir des PIDs conçus par le fabricant qui ne sont pas définis dans le standard OBD2 [6].

Mode1:

Retourne les valeurs courantes de :

- régime moteur
- vitesse
- températures
- etc.

Mode 2 :

Retourne les données gelées de défauts.

Mode 3 :

Ce mode retourne les codes défauts enregistrés.

Ces codes défauts ont été standardisés pour toutes les marques de véhicule et classés en 4 catégories :

- P0xxx : défauts standards liés au système de propulsion (moteur et transmission)
- C0xxx : défauts standards liés au châssis
- B0xxx : défauts standards liés à la carrosserie
- U0xxx : défauts standards liés aux réseaux de communications

Mode 4 :

Ce mode permet d'effacer les codes défauts enregistrés et d'éteindre le voyant moteur orange.

Mode 5 :

Ce mode retourne les résultats des auto-diagnostic effectués sur les sondes à oxygène/lambda. Il s'applique principalement aux véhicules essence.

Mode 6 :

Ce mode retourne les résultats des auto-diagnostic effectués sur les systèmes non soumis à surveillance constante.

Mode 7 :

Ce mode retourne les codes défauts non confirmés.
Les codes utilisés sont identiques à ceux du mode 3.

Mode 8 :

Ce mode retourne les résultats des auto-diagnostic effectués sur d'autres systèmes (il est très peu utilisé en Europe).

Mode 9 :

Ce mode retourne les informations concernant le véhicule telles que:

- le VIN numéro d'identification du véhicule
- les valeurs de calibration

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 de conduite sans apparition du problème effaceront les défauts.

Remarque :

- Les **codes défauts du système de diagnostic OBD2** sont des codes stockés par le système informatique du véhicule et correspondent à un problème enregistré. Ces codes identifient une zone d'erreur et sont destinés à vous servir de guide vers le point précis où se trouve le problème sur le véhicule.
- Il existe deux types de codes défauts : les **codes défauts standards** et les **codes défauts constructeurs**. La liste des codes génériques n'étant pas toujours suffisante, les constructeurs peuvent ajouter de nombreux codes spécifiques à leurs modèles. Les codes défauts standards ont été définis afin que n'importe quel appareil de diagnostic soit en mesure de les lire et de les décoder grâce à un format standardisé.

III. Conclusion :

Nous arrivons à présent à la fin de ce premier chapitre dans lequel nous avons abordé l'électronique automobile et ses divers périphériques, et nous avons présenté le bus CAN puis on a terminé par l'outil de diagnostic automobile qui est l'OBD.

Cette étude nous a permis 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 jugé utile de présenter les communications sans fil (tels que Bluetooth, GPS, GSM) auxquelles nous aurons à faire, cela sera l'objet du prochain chapitre.

Chapitre II :

La connectivité automobile et la géolocalisation

I. Introduction :

L'Homme a toujours eu besoin de localiser les objets et de se situer dans l'environnement confronté à ce sujet, il a cherché des repères dans le ciel (étoiles et constellations), a cherché et créé des repères sur la planète (fleuves, sommets des montagnes...), a tracé des cartes, des itinéraires, a découvert et créé divers instruments (la boussole, le théodolite...).

L'interconnexion 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 (Bluetooth, RFID ...etc.), elle a pour but de créer un réseau d'objets interconnectés tels que les Smartphones, les télévisions ou les voitures.

Dans ce chapitre nous allons présenter la connectivité automobile, en précisant ses domaines et les protocoles de communication sans fil qu'elle utilise, on se basant sur la norme Bluetooth, puis nous allons définir la géolocalisation et ses techniques, en s'appuyant sur les deux techniques les plus utilisées qui sont le GPS et GSM, nous détaillerons ensuite le système de positionnement par satellite le plus connu, qui est le GPS.

II. La connectivité automobile :

II.1. Véhicule connecté :

Une voiture connectée est une voiture disposant d'un système de communication embarqué qui offre un accès à Internet. Cette connexion s'effectue via les réseaux de téléphonie mobile grâce à un équipement intégré au véhicule, ou en utilisant le smartphone du conducteur ou d'un passager. Très souvent, la voiture connectée est dotée d'un réseau local sans fil permettant de partager cet accès entre divers équipements à bord du véhicule.

Elle peut également se connecter à l'infrastructure du réseau routier ou aux autres véhicules circulant à proximité. Ainsi, les utilisateurs d'une voiture connectée bénéficient d'une gamme étendue de services et d'applications destinés en priorité au conducteur, mais également aux passagers.

Ces innovations impliquent une multiplicité d'acteurs : constructeurs et équipementiers automobiles, souvent en partenariat avec des fabricants d'équipement électronique et des entreprises de développement de logiciels et d'applications, mais aussi multinationales de l'Internet ou de l'industrie informatique et opérateurs de téléphonie mobile.

Comme pour tout objet connecté, l'aspect sécurité est essentiel. Il s'agit de protéger la voiture connectée contre la cybercriminalité et les accidents. Ce point est d'autant plus critique que les automobiles évoluent en intégrant des systèmes robotiques qui visent à les transformer en voiture autonome.

II.2. Technologies de communication automobile:

Les technologies de communication mises en œuvre dans les voitures connectées empruntent les réseaux de téléphonie mobile pour établir une connexion à Internet. On distingue trois modes de connectivité.

- connectivité embarquée (embedded) : le véhicule est équipé d'une carte SIM spéciale;
- connectivité intégrée : utilisation de la connexion du smartphone du conducteur ou d'un passager, en intégrant les applications grâce à un écran tactile ou des commandes vocales ;
- connectivité partagée (teethered) : la connexion du smartphone est partagée entre l'utilisateur et l'automobile.

II.3. Utilisateurs du véhicule :

➤ Conducteur :

Les applications et services destinées au conducteur sont variés. Par exemple :

- navigation et guidage GPS, informations routières et prévision de la circulation;
- planification de la maintenance et dépannage du véhicule;
- détection d'accident et appel automatique;
- détection des piétons (par leur smartphone ou par wifi);
- clé électronique sur smartphone pour ouvrir et démarrer la voiture.

L'exploitation des données collectées sur le véhicule permet de proposer des assurances personnalisées et des services d'évaluation des performances écologiques du conducteur.

Pour y pouvoir se servir et traiter ces données on utilise des protocoles sans fils.

➤ Type du conducteur :

Quand on possède une voiture, il est très courant qu'on ne soit pas la seule personne à la conduire. Qu'il s'agisse d'un conjoint ou d'un ami, par exemple, il arrive souvent qu'on soit amené à laisser quelqu'un d'autre s'en servir. Cependant, il faut savoir que selon les contrats d'assurance, on n'est pas forcément couvert quand on laisse quelqu'un d'autre prendre le volant.

On fait toujours la différence entre plusieurs types de conducteurs.

Il existe premièrement le **conducteur principal**, qui comme son nom l'indique est celui qui conduit le plus souvent le véhicule.

Il existe ensuite une distinction entre **conducteur secondaire** et **conducteur occasionnel**. Le premier est une personne qui conduit le véhicule de façon régulière. Quant au conducteur occasionnel, il s'agit de n'importe quel individu qui se sert de la voiture de façon ponctuelle.

Ce qui nous a permis de faire une étude sur la façon d'utiliser un véhicule par chacun de ces conducteurs afin de mieux les classer tout en utilisant un système de traçage.

II.4. 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 (la Figure II.1 présente plus de détails) [7].



Figure II. 1: Les protocoles sans fil automobile.

II.4.1. Le Bluetooth :

Bluetooth est une technologie de réseau personnel sans fils , c'est-à-dire une technologie de réseaux sans fils d'une faible portée permettant de relier des appareils entre eux sans liaison filaire. Contrairement à la technologie infrarouge, la technologie Bluetooth utilise les ondes radio (bande de fréquence des 2.4 GHz) pour communiquer, les appareils Bluetooth ne nécessitent pas d'une ligne de vue directe pour communiquer, ce qui rend plus souple son utilisation et permet notamment une communication d'une pièce à une autre, sur de petits espaces [8].

II.4.2. Normes Bluetooth :

Le standard Bluetooth se décompose en différentes normes :

- IEEE 802.15.1 définit le standard Bluetooth 1.x permettant d'obtenir un débit de 1 Mbit/sec ;
- IEEE 802.15.2 propose des recommandations pour l'utilisation de la bande de fréquence 2.4 GHz (fréquence utilisée également par le WiFi). Ce standard n'est toutefois pas encore validé ;
- IEEE 802.15.3 est un standard en cours de développement visant à proposer du haut débit (20 Mbit/s) avec la technologie Bluetooth ;
- IEEE 802.15.4 est un standard en cours de développement pour des applications Bluetooth à bas débit [8].

II.4.3. Mise en œuvre (communication Bluetooth) :

La communication avec le protocole Bluetooth est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est « maître/esclave ». Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves.

Pour son utilisation, elle se passe en plusieurs étapes :

1. Le maître se met en mode « reconnaissable »
2. L'esclave trouve le maître et demande à s'y connecter
3. Le maître accepte la connexion
4. Les périphériques sont alors appariés (ou associés)
5. La communication peut commencer

Afin d'échanger des données, les appareils doivent être appariés. L'appairage se fait en lançant la découverte à partir d'un appareil et en échangeant un code. Dans certains cas, le code est libre, et il suffit aux deux appareils de saisir le même code. Dans d'autres cas, le code est fixé par l'un des deux appareils (appareil dépourvu de clavier, par exemple), et l'autre doit le connaître pour s'y raccorder. Par la suite, les codes sont mémorisés, et il suffit qu'un appareil demande le raccordement et que l'autre l'accepte pour que les données puissent être échangées.

III. La géolocalisation :

III.1. Présentation de la géolocalisation :

La géolocalisation ou géo référencement est un procédé permettant de positionner un objet sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Cette opération est réalisée à l'aide d'un terminal capable d'être localisé grâce à un système de positionnement par satellites et un récepteur GPS par exemple, ou par d'autres techniques; de plus le terminal est en mesure de publier, en temps réel ou de façon différée, ses coordonnées géographiques latitude/longitude. Les positions enregistrées peuvent être stockées au sein du terminal et être extraites ultérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. La transmission en temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM, GPRS, UMTS, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet à la plateforme de visualiser la position du terminal au sein d'une carte. La plateforme est le plus souvent accessible depuis internet [9].

III.2. Historique de la géolocalisation :

A l'origine, la géolocalisation a été conçue pour des besoins de l'armée américaine. En 1993, Bill Clinton décide d'ouvrir cette technique au grand public. Des services sont alors apparus permettant de localiser des objets, des personnes, de gérer des flottes de véhicules sur un plan ou une carte à l'aide de coordonnées géographiques.

Depuis quelques années, la géolocalisation s'est démocratisée et est très utilisée notamment pour la navigation routière.

Ces positions sont obtenues à l'aide de systèmes de radionavigation par satellites (GNSS), le terminal le plus connu du grand public est le GPS (Global Positioning System). Ce sont ces services de navigation routière qui ont très fortement dynamisé le marché au début de la géolocalisation, et qui continuent de représenter la majorité du marché en valeur.

Il existe de nombreuses technologies de localisation et nous ne les nommerons pas tous ses procédés.

Aujourd'hui, nous avons choisi de nous concentrer sur deux besoins différents dans le domaine de la géolocalisation [10].

III.3. Les techniques de géolocalisation :

III.3.1. La géolocalisation par GSM :

GSM pour « Global System for Mobile Communications » est une norme numérique de deuxième génération pour la téléphonie mobile.

Avec cette technique de géolocalisation, on trouve la position de l'objet désiré en se basant sur certaines informations relatives aux antennes GSM auxquels ce dispositif est connecté. La précision va dépendre, ici, de l'environnement, du milieu où se situe l'appareil. En milieu urbain, la densité d'antennes est supérieure par rapport au milieu rural.

Il existe plusieurs techniques de géolocalisation par GSM qui sont comme suit [11] :

a. Différentiel temps ou EOTD (Enhanced Observed Time Différence) :

Cette technologie, repose sur un système relativement simple. Le téléphone mobile du salarié itinérant émet un signal qui est récupéré et renvoyé par la station mobile la plus proche. La différence de temps entre l'émission et la réception de l'onde GSM permet à un serveur externe de localiser l'emplacement du téléphone portable sur son réseau.

b. Identification de cellule (Cell ID) :

Cette technologie est la plus utilisée, elle permet de localiser l'utilisateur qui se situe dans une zone couverte par le réseau grâce à l'identification de la cellule de son antenne mobile. Cette technique est très rapide puisque la géolocalisation peut être effectuée en moins de 3 secondes. En revanche, elle est relativement imprécise puisqu'il y a une marge d'erreur :

- de 200 à 300 mètres en ville lorsque le réseau est dense
- de plus ou moins 10 kilomètres en campagne ou dans une zone rurale

c. La triangulation :

Cette technique de localisation fait appel à l'antenne hertzienne du téléphone mobile de l'utilisateur. Trois relais émetteurs et récepteurs envoient des informations qui sont ensuite traitées et analysées. Les relais changent en fonction des déplacements du téléphone portable.

Pour être opérationnelle, ce type de triangulation implique que la carte SIM du mobile soit munie d'une application spécifique.

Par rapport à la géolocalisation Cell ID, cette technique est beaucoup plus précise puisque la marge d'erreur est relativement réduite :

- 120 à 130 mètres en ville lorsque le réseau est dense
- environ 5 kilomètres en campagne ou dans une zone rurale

En revanche, le temps de localisation est de 4 à 6 secondes en moyenne.

III.3.2. La géolocalisation par satellite GPS:

III.3.2.1. Introduction:

Le GPS est un système de navigation à base de satellites conçu pour fournir instantanément des informations de position, de vitesse et de temps pratiquement à n'importe quel endroit sur terre, n'importe quelle heure et par n'importe quelles conditions météorologiques. Le GPS permet un positionnement instantané avec une précision qui va d'une centaine de mètres à quelques mètres (95% des cas).

Les principales étapes dans le développement du système GPS ont été :

- 1973 : début du programme sous l'égide du Département de la défense des USA.
- 1974-1983 : essais des premiers récepteurs, lancement des premiers satellites du block I.
- 1985 : Le système de navigation s'ouvre aux applications civiles.
- 1986 : poursuite de l'évolution du système avec les lancements des satellites du block II
- 1990 (25 mars) : mise en fonction de la SA.
- 1991 : la SA est momentanément désactivée durant la première guerre du Golfe ce qui entraîne une fantastique promotion du GPS.
- 1995 (17 juillet) : USA annonce que le système GPS est pleinement opérationnel.
- 1997 : lancement des satellites du block IIR.
- 2000 (1^{er} mai) : la SA est désactivée.

Bien que développé dans un but essentiellement militaire, les applications civiles du GPS ont connu depuis des années un essor considérable. Parmi les utilisations civiles actuelles du GPS, on peut citer :

- La navigation en générale (course pédestre, marine, **automobile**,...)
- Les relevés géométriques et topographiques.
- Etc.,...

III.3.2.2. Les segments du GPS :

Du point de vue fonctionnel, le système GPS est composé de trois segments principaux : le segment spatial, le segment de contrôle et le segment utilisateur.

a. Le segment spatial:

C'est la constellation des satellites du GPS. Le segment spatial est le cœur du système, il est composé de 24 satellites, répartis de telle sorte que chaque zone de la terre est couverte par 4 satellites. Ces satellites évoluent en orbite haute soit à plus de 19.000 kilomètres au-dessus de la surface de la terre. Le fait d'être positionné à une telle altitude permet aux signaux de couvrir une très large zone.

Les orbites de ces satellites sont organisées de telle façon qu'un récepteur GPS sur terre reçoive toujours les signaux de trois satellites au minimum à la fois et à un instant donné. Ces satellites sont en orbite à plus de 11.000 km/h ce qui leur permet également de faire le tour de la terre toutes les 12 heures.

b. Le segment de contrôle:

C'est la partie qui permet de piloter et de surveiller le système. Il contrôle les satellites GPS en les suivant et en leur fournissant des corrections d'orbites et de temps. Dans le Monde, il existe cinq stations de contrôle au sol qui sont situées tout autour de la terre.

En fait, il y a quatre stations automatiques de surveillance et une seule station principale de contrôle. Les stations automatiques reçoivent constamment des données provenant des satellites et renvoient ces informations à la station principale de contrôle. La station principale de contrôle corrige et met à jour (éphémérides, paramètres d'horloge ...) les données reçues puis renvoie les informations aux satellites via deux antennes situées sur deux autres sites différents.

c. Le segment utilisateur :

C'est l'ensemble des utilisateurs civils ou militaires qui ne font que recevoir et exploiter les informations reçues des satellites via leur récepteur GPS.

Le segment utilisateur regroupe les utilisateurs militaires (marine, armée de terre, de l'air), la police ou la gendarmerie. Dans le domaine civil, le segment utilisateur regroupe les navigateurs maritimes, les pilotes, les chasseurs, pêcheurs, randonneurs, les sportifs, les conducteurs d'engin etc.

III.3.2.3. Le fonctionnement d'un GPS :

Le GPS fonctionne grâce à un moyen que vous connaissez sûrement : **la triangulation**.

La triangulation permet grâce à trois satellites fixes qui connaissent leur position exacte de déterminer la position d'un point précis sur Terre. Chaque satellite du réseau GPS possède une horloge atomique d'une grande précision.

Principe de triangulation [12] :

- ✚ À un instant t un satellite émet son signal vers le récepteur GPS qui le reçoit à un instant tI . Le GPS calcule ensuite grâce à une opération simple $tI - t = T$ qui est le temps qu'a mis le signal pour arriver. Connaissant la vitesse du signal (300 000 km/h) le GPS, grâce à la relation $distance (d) = temps \times vitesses$ détermine une sphère de centre le satellite et de rayon d qui sont ses positions susceptibles.

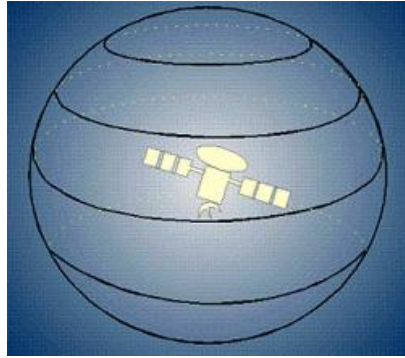


Figure II. 2 : cas d'un seul satellite.

- ✚ En renouvelant ce mécanisme avec un deuxième satellite il détermine une deuxième sphère qui coupe la 1^{ère} en un plan sur lequel est situé le GPS.

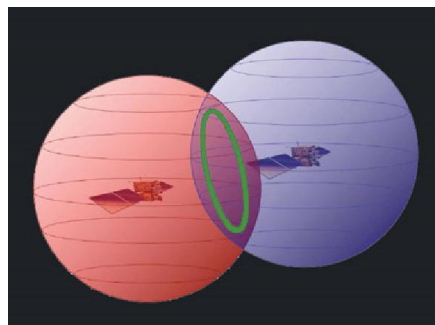


Figure II. 3 : cas de deux satellites.

- ✚ Pour terminer, un troisième satellite est nécessaire afin de trouver sa position exacte sur ce plan. En effet lorsque la troisième sphère coupe le plan elle donne deux points.

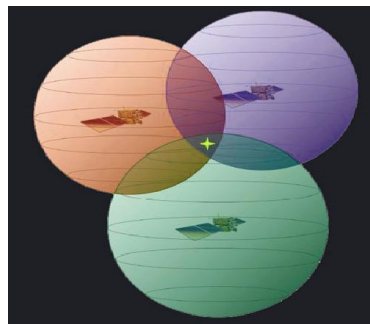


Figure II. 4 : cas de trois satellites.

- ✚ Dans le cas où l'utilisateur se situe à la surface de la Terre seul un des deux points est cohérent. Ainsi on peut déduire sa position exacte en éliminant le point donnant un résultat incohérent.

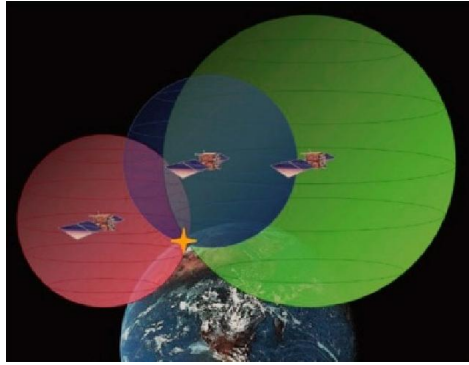


Figure II. 5: le positionnement à l'aide des trois satellites.

IV. Autres systèmes de traçage :

A côté des systèmes de guidage ou de géolocalisation en temps réel, il existe des solutions permettant simplement de localiser un objet sur demande :

- **La balise RFID** : ce petit boîtier (qui fait parfois à peine la taille d'une carte de crédit), contient une radio étiquette (ou « tag RFID »). Son utilisation est conseillée dans des zones hostiles (murs épais en béton, métal, proximité de l'eau...) mais peu étendues.
- **Le traceur** : ce petit boîtier permet de localiser un objet à tout moment, sur demande. Il est essentiellement utilisé comme moyen de prévention contre le vol. Certains modèles, très perfectionnés, sont couplés avec des systèmes d'alerte, par exemple envoi d'un SMS en cas de démarrage de véhicule ou de choc, possibilité de couper le moteur à distance...

V. Conclusion :

Dans ce chapitre nous avons introduit la notion de la connectivité automobile puis nous avons présenté le protocole sans fil Bluetooth, ainsi nous avons abordé les techniques de la géolocalisation d'automobile en focalisant principalement sur la technologie GPS et son fonctionnement.

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:

Conception du système de diagnostic et géolocalisation

I. Introduction :

La conception de toute solution logicielle doit être traitée avec précision et détail, précédée d'une analyse profonde et bien réfléchi, 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 logiciels et matériels qui nous serviront à la réalisation de notre application ainsi la réalisation matérielle de tous les modules utilisés durant notre projet. Puis nous allons passer à la phase de conception.

II. Objectif du système :

Dans ce travail nous allons réaliser un système qui permet la géolocalisation et la récupération des informations du véhicule en temps réel (vitesse, régime moteur,...etc.).

III. Structure globale du système :

La Figure III. 1 montre la structure globale du système à réaliser :

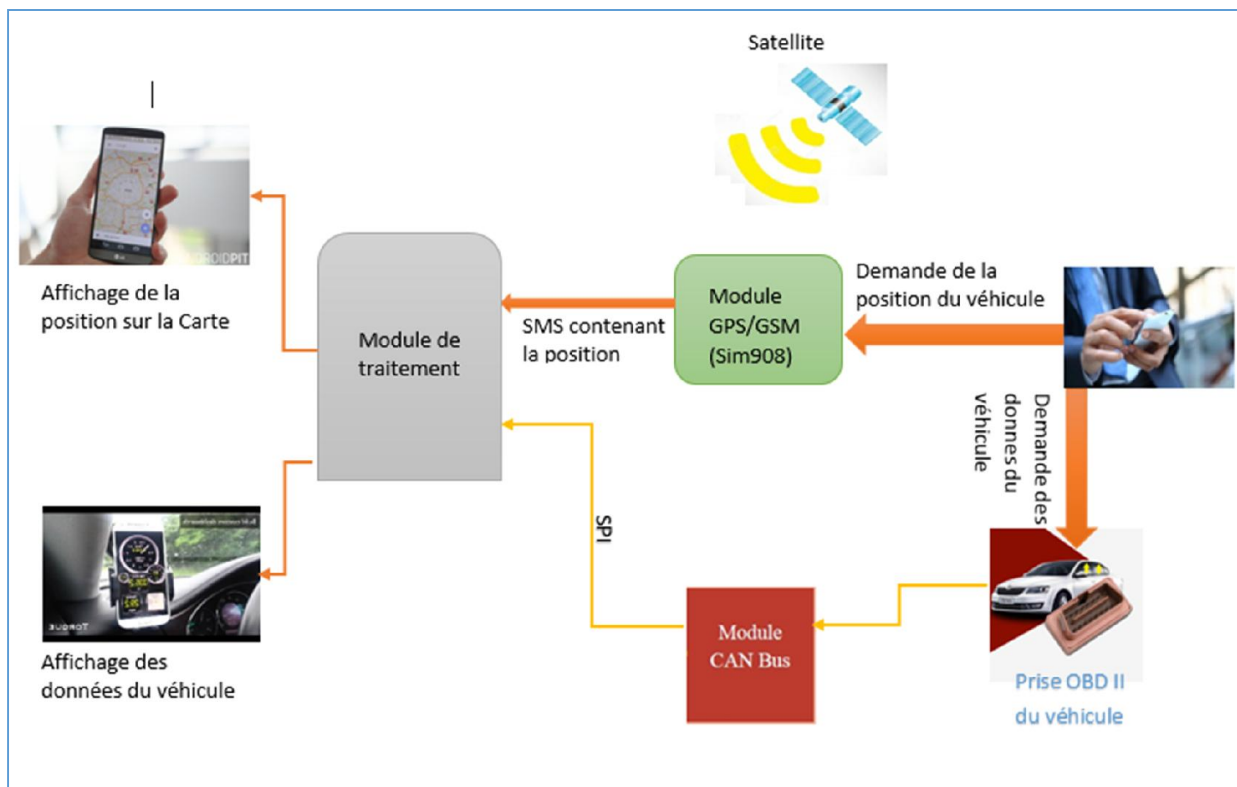


Figure III. 1: Structure générale de notre système.

Le système représenté dans la figure III.1 consiste à concevoir et développer une application pour :

- pouvoir récupérer les données en temps réel d'un véhicule (la vitesse, le régime, ...etc.) à partir de la prise OBD-II en utilisant le module Bus CAN (interface d'acquisition ELM327) et les afficher sur un smartphone sous forme d'un tableau de bord du véhicule.
- Ainsi localiser le véhicule à n'importe quel moment et à n'importe quel endroit à partir du module SIM908 avec le module de traitement qui est avec, ce dernier envoie la position sous forme de message (SMS), qui contient des coordonnées : latitude et longitude par exemple et afficher la position sur une carte cartographique sur le smartphone. Une fois le message reçu, le propriétaire peut visualiser l'endroit où se trouve son véhicule.

IV. Outils nécessaires pour notre système:

Avant d'entrer dans la conception du système, il faut faire le point sur les outils matériels et logiciels que nous devons utiliser.

Ce que nous avons besoin pour notre système sont les outils suivants :

- **La carte Arduino utilisée** : il s'agit de la carte Mega 2560 a pour rôle comme module de traitement.
- **Le module Bus Can Shield** : qui a pour rôle comme un module d'acquisition.
- **Câble OBD-II à DB9** : permet de récupérer les signaux d'un port OBD-II et les envoyer au module de traitement.
- **Le Module SIM 908** : il s'agit du module quadri-bande GSM/GPRS/GPS.
- **Bluetooth HC-05** : qui joue un rôle comme module de communication entre le calculateur moteur du véhicule et le Smartphone.
- **Le module de communication ELM 327 Bluetooth** : qui est une interface de diagnostic automobile, qui supporte tous les protocoles OBDII possédant un Bluetooth.
- **La SDK Android** : kits nécessaire pour développement Android.
- **IDE Android Studio** : Logiciel gratuit de Google pour développer des applications Android.
- **Un Smartphone** : pour tester et affichage de notre application.

Dans ce qui suit nous allons voir en détails les définitions et les fonctions exactes de chacun des outils précédents.

a. Présentation de la carte Arduino Mega 2560:

L'Arduino Mega qu'on a utilisée est une carte microcontrôleur basé sur l'ATmega 2560. Il dispose de 54 broches numériques d'entrée / sortie, 16 entrées analogiques, 4 UART (ports série matériels), d'une connexion USB, une prise d'alimentation et un bouton de réinitialisation. Il contient tout le nécessaire pour soutenir le microcontrôleur, il suffit de le connecter à un ordinateur avec un câble USB ou batterie pour commencer [13].

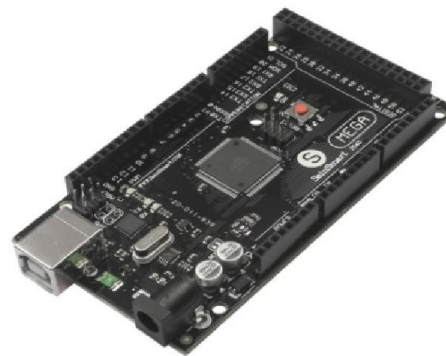


Figure III. 2 : La carte Arduino Mega 2560

b. CAN-Bus Shield [14] :

➤ Description:

Ce bouclier est équipé d'un microcontrôleur Microchip MCP2515 CAN, lui-même équipé de l'émetteur récepteur MCP2551 CAN. Une fois fiché sur votre Arduino, il lui offre la possibilité de transmettre des informations à la façon d'un bus CAN (Controller Area Network), ce qui la rend beaucoup plus efficace. Le Shield CAN-BUS possède également deux emplacements dédiés intéressants : l'un pour connecter une carte mémoire micro SD, et l'autre pour connecter un module GPS EM-406 (non fournis). Il possède également d'une interface SPI ultra-rapide (10 MHz).

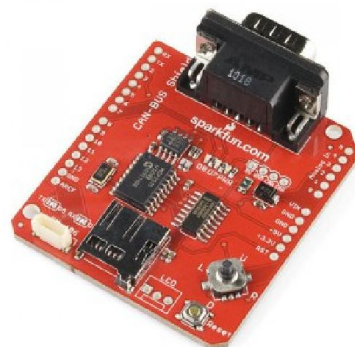


Figure III. 3: Le CAN-Bus Shield.

➤ **Spécifications techniques:**

- CAN v.2.0B: jusqu'à 1 Mb/s.
- Interface SPI High speed: 10 MHz
- Données et demandes de requête standards et étendues.
- Connexion CAN via connecteur standard sub-D 9 broches.
- Alimentation de l'Arduino via câble sub-D par fusible réamorçable.
- Protection contre les inversions de polarité.
- Emplacement pour module GPS EM506.
- Connecteur pour carte mémoire micro SD.
- Connecteur pour afficheur LCD série.
- Bouton Reset.
- Joystick de contrôle.
- 2 indicateurs LED.

c. Câble OBD-II à DB9 :

Ce câble permet d'accéder aux broches du connecteur OBDII de la voiture. Il possède un connecteur mal OBDII (16 broches) sur une extrémité et un connecteur série femelle DB9 (9 broches) sur l'autre extrémité, qui doit être branché au bus Can Shield dotée d'une connectique SUB-D 9 comme montre la figure III.4 suivante :

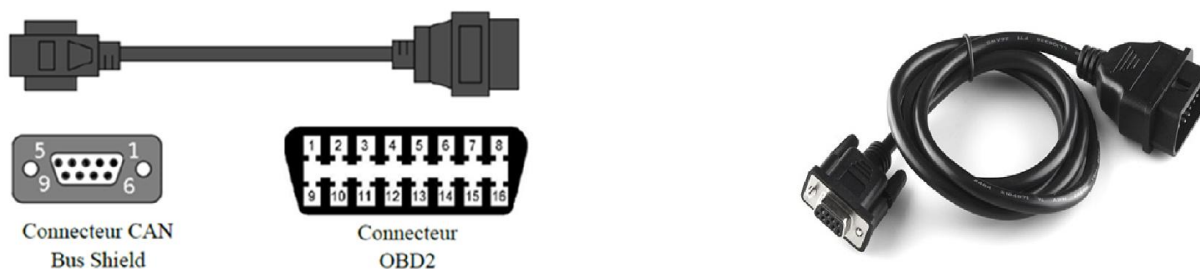


Figure III. 4 : Le câble OBD-II à DB9.

d. Module SIM908 :

➤ **Présentation du Module GSM/GPRS/GPS SIM 908 :**

Le SIM908 utilisé dans notre localisateur s'agit d'un module quadri-bande GSM / GPS travaillant sur les fréquences 850/900/1800/1900 MHz, chaque partie est dotée de sa propre prise d'antenne.

La partie GSM permet la gestion des SMS, des communications de type « voix » et des communications de type « data » (transfert de données). De plus, il peut être utilisé pour envoyer et recevoir des données après avoir été connecté au réseau (un accès direct à Internet) via le

GPRS. En outre, il peut réaliser un positionnement global en localisant son Position géographique avec le récepteur GPS.

Ce module est géré par un processeur AMR926EJ-S basé sur l'architecture de version 5 qui s'occupe de la communication téléphonique et des données (via le protocole TCP/IP) et dialogue (par l'intermédiaire d'une trame UART avec l'interface série TTL) avec la partie GSM. Le processeur gère également la carte SIM en 3V ou 1,8 V qui est à l'extérieur du module.

Le module dispose de deux ports série TTL : un principal (TX/RX) qui est utilisé pour transmettre les données d'un SMS et la transmission des données au cours d'une session TCP/IP en GPRS, mais permet aussi de recevoir des commandes du circuit (dans notre cas, celles provenant de la carte arduino mega 2560) qui peuvent être à la fois des commandes de type AT standards, commandes de type propriétaire AT étendues de SIMCOM. Celui du GPS (GPSTX/GPSRX) est plutôt utilisé pour communiquer la trame NMEA. Le module fonctionne avec une tension allant de 3,4 à 4,5 V et consomme un courant de 0,4 A avec le GSM en mode de transmission et le GPS en fonction.



Figure III. 5 : Le module SIM908.

e. Bluetooth HC-05 :

Le module HC-05 est un module Bluetooth SPP (Serial Port Protocol) facile à utiliser, conçu pour une configuration de connexion série sans fil transparente. Le module Bluetooth HC-05 peut être utilisé dans une configuration Master ou Slave, ce qui en fait une excellente solution pour la communication sans fil. Ce module bluetooth de port série est entièrement équipé de la modulation Bluetooth V2.0 + EDR (Enhanced Data Rate) de 3Mbps avec un émetteur-récepteur radio de 2,4 GHz et une bande de base. Il utilise CSR Bluecore 04 - Système externe Bluetooth à puce unique avec technologie CMOS et avec AFH (Adaptive Frequency Hopping Feature).



Figure III. 6: Bluetooth HC-05.

Remarque :

Comme on a cité dans le deuxième chapitre dans la section Bluetooth ce que concerne la notion de maître/esclave :

- Le rôle du module ne peut être configuré que par des commandes AT.
- Les modules esclaves ne peuvent pas lancer une connexion à un autre périphérique Bluetooth, Mais peut accepter les connexions.
- Le module Maître peut lancer une connexion à d'autres périphériques.

f. Le module de communication ELM327 :**➤ Définition :**

L'interface de diagnostic ELM327 qui supporte tous les protocoles OBD vous permet de lire les codes défauts moteur de votre voiture et d'effacer le voyant orange (MIL) du tableau de bord. L'original ELM327 est implémenté sur le microcontrôleur PIC18F2480 de Microchip Technology.

On a utilisé l'ELM 327 Bluetooth, cette dernière qui contient un module Bluetooth intégré peut se connecter à distance à votre Smartphone Android ou à votre PC pour diagnostiquer facilement et rapidement les problèmes de votre véhicule.

Remarque : Ce circuit intégré ELM327 Bluetooth est équivalent à l'interconnexion des éléments qu'on a utilisés: Le module de traitement (carte arduino) + Le module d'acquisition (Bus Can Shield) + Câble adaptateur DB9 + Bluetooth puisque ces composants qui font partie du système embarqué. (Vu que l'ELM327 fonctionne sur tous les protocoles par contre le Bus can Shield fonctionne juste sur le protocole CAN ISO 11898 des véhicules Allemand).



Figure III. 7 : ELM 327 Bluetooth.

➤ **Principe de fonctionnement de l'interface ELM 327 [15] :**

Le boîtier électronique contient 3 sous-ensembles principaux lui permettent de jouer son rôle d'interface :

- **Adaptation de tensions électriques:** Les réseaux embarqués dans les automobiles ont des niveaux de tensions qui requièrent des drivers spécifiques. Dans le cas d'un ELM327, comme ce boîtier supporte de nombreux protocoles, plusieurs drivers sont nécessaires.
- **La puce ELM327:** Le circuit intégré, qui par abus de langage donne son nom à tout le boîtier, est le cerveau de l'interface. Il est en charge de sélectionner le protocole et de convertir celui-ci en un protocole compréhensible par les PC modernes. Ce petit circuit (cadencé 4MHz) a pour unique rôle de faire la passerelle entre les protocoles.
- **Adaptation de tensions pour le PC:** La puce électronique n'étant elle-même pas capable de communiquer avec un ordinateur, il est nécessaire une fois de plus d'adapter les niveaux électriques avant d'envoyer le flux d'information.

L'image Figure III. 8 présente de manière schématique les principales fonctions d'une interface :

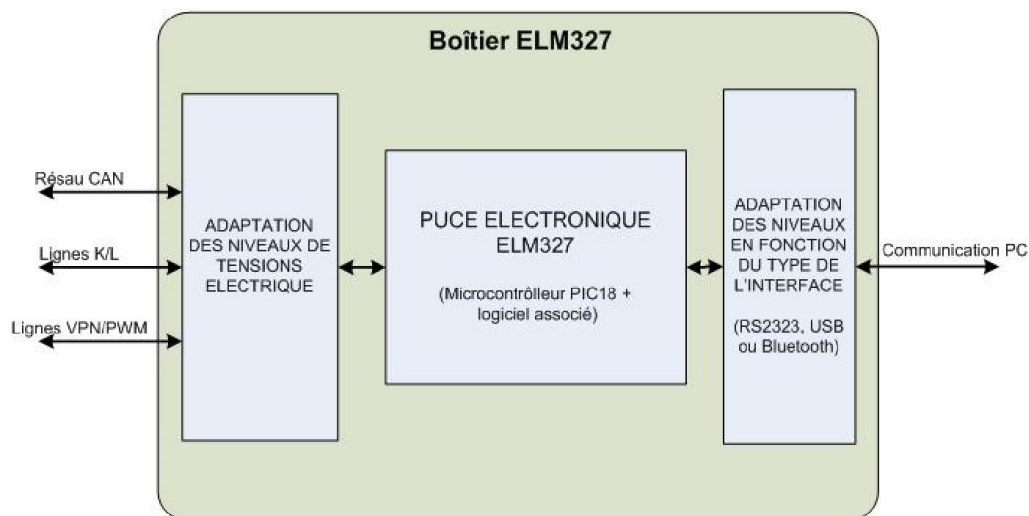


Figure III. 8 : La manière schématique sur de fonctionnement de l'interface ELM 327.

Après cette présentation, le terme d'interface trouve tout son sens. Le rôle des ELM327 est bien d'interfacer 2 mondes bien distincts : d'une part celui de l'automobile régi par ces standards imposant un niveau élevé de robustesse et un prix faible et d'autre part le monde de l'informatique qui est plus axé sur la modularité.

➤ **Le but d'utilisation de l'interface ELM327 :**

La collecte des informations avec le module CAN Bus Shield était performante mais limitée.

Après plusieurs testes effectués sur des véhicules on a déduit que le module d'acquisition BUS CAN Shield supporte seulement un seul protocole de communication OBD-II qui est le protocole CAN, la preuve la récupération des données avec cette dernière marche équitablement seulement sur des véhicules allemands par rapport aux autres véhicules, par contre lorsque on a testé sur un autre véhicule (exemple Citroen Berlingo) les résultats obtenues n'étaient pas vraiment exactes et il y avait des erreurs de collecte des informations.

Donc l'utilisation de module ELM 327 permet de récupérer directement les informations dans le véhicule sans passer par un circuit externe possédant les deux principaux modules (carte Arduino + CAN Bus Shield). Ce système s'applique sur tous les véhicules quel que soit le protocole de communication OBD qu'elle possède.

g. La SDK Android:

Le Kit de développement logiciel Android (Android SDK) contient les outils nécessaires pour créer, compiler et déployer les applications Android. La plupart de ces outils sont en ligne de commande. Le principal moyen de développer des applications Android est d'utiliser le langage de programmation Java.

Le SDK Android contient un débogueur appelé « Android debug bridge » ou aussi « adb », qui permet de connecter un appareil Android virtuel ou réel, dans le but de gérer le périphérique ou de déboguer votre application.



Figure III. 9: Le logo du SDK Android

h. IDE Android Studio :

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA.

Android Studio permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.



Figure III. 10 : Le logo d'IDE Android Studio.

i. Module d'affichage (Smartphone) :

Un smartphone désigne un téléphone intelligent qui peut exécuter divers logiciels ou applications grâce à un système d'exploitation spécialement conçu pour mobiles (Android, Linux, iOS,...), ces applications particulier permettent de fournir des fonctionnalités en plus de celles des téléphones mobiles classiques comme : navigation sur Internet, lecture des musiques et des films, localisation GPS, écran tactile,...etc. Un exemple d'un smartphone qu'on a utilisé durant notre mémoire s'agit d'un LG x power.



Figure III. 11: Un Smartphone.

V. Implémentation matérielle:

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

➤ **Carte Arduino Mega 2560 et CAN Bus Shield:**

- Pour l'alimentation:

Arduino mega 2560	CAN Bus Shield
Reset	Reset
VCC	VCC
GND	GND
Vin	Vin

Tableau 4 : alimentation du Bus CAN Shield par Arduino mega2560.

- Pour la communication:

Arduino mega 2560	CAN Bus Shield
MOSI (51)	P11
MISO (50)	P12
P8	P10
CLK	P13
INT0(2)	P2

Tableau 5: La communication entre le Bus CAN Shield avec Arduino mega2560.

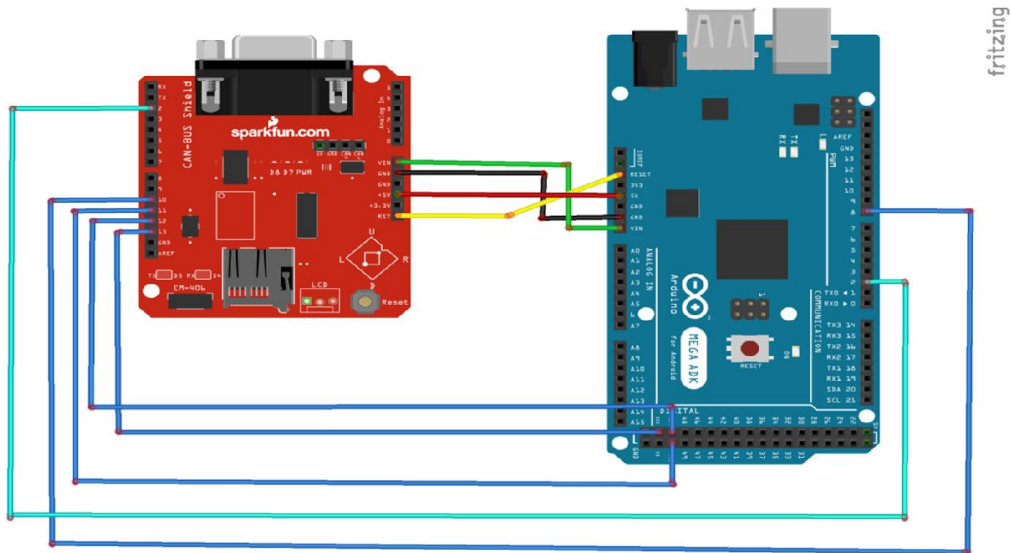


Figure III.12: Le schéma de brochage entre Arduino et Bus CAN Shield.

➤ **Carte Arduino Mega 2560 et Bluetooth HC-05:**

- Pour l'alimentation:

Arduino mega 2560	Bluetooth HC-05
VCC	VCC
GND	GND

Tableau 6: alimentation du HC-05 par Arduino mega 2560.

- Pour la communication :

Arduino mega 2560	Bluetooth HC-05
Rx	Tx
Tx	Rx

Tableau 7: La communication du HC-05 avec Arduino mega 2560.

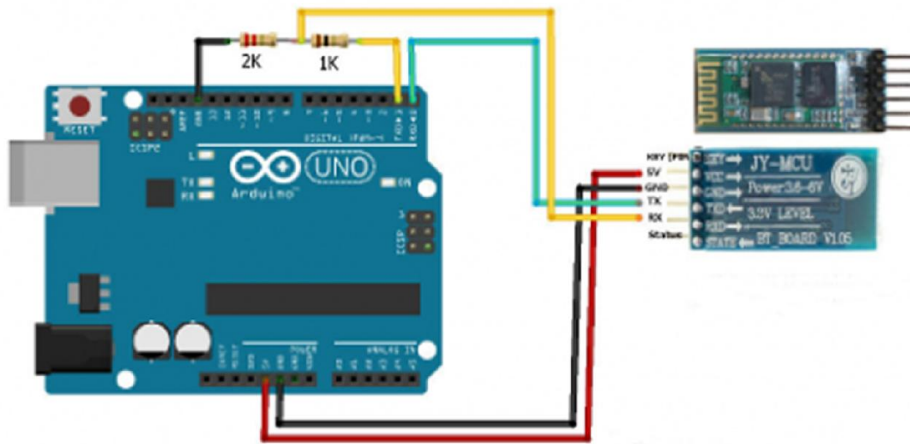


Figure III. 13 : Brochage Arduino Bluetooth HC-05.

Il ne reste alors plus que les broches Key et Led. La broche **Led** sert à connecter une LED de statut, vous pouvez la laisser déconnectée cela n'influencera pas le comportement du module. La broche **Key** sert à utiliser le mode « commande » du module. Avec cette dernière vous pourrez reconfigurer la voie série (vitesse, parité etc...) et d'autres options liées au Bluetooth (nom du module, mot de passe d'appairage, mode esclave/maître...). Cette broche est à connecter à n'importe quelle sortie numérique de l'Arduino (GPO).

➤ **Carte Arduino Mega 2560 et SIM908:**

- Pour l'alimentation :

Arduino mega 2560	SIM908
VCC	VCC
GND	GND
Vin	Vin

Tableau 8: L'alimentation du SIM 908 par Arduino mega 2560.

- Pour la communication :

Arduino mega 2560	SIM908
Tx	Rx
Rx	Tx

Tableau 9: La communication entre SIM 908 avec Arduino mega 2560.

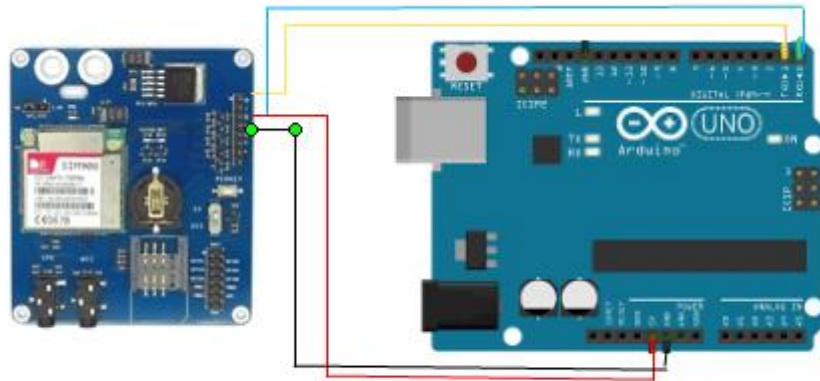


Figure III. 14 : Le schéma de brochage entre Arduino et SIM 908.

VI. Conception :

VI.1. Introduction :

La production de certaines applications exige de pouvoir générer des graphes au sens recherche opérationnelle du terme. C'est-à-dire des graphiques représentant des nœuds liés entre eux via des arcs orientés ou non.

Donc après avoir présenté les outils nécessaires pour notre système, nous allons, suivre une méthode de conception dans le domaine des systèmes embarqués 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.

La Figure III. 15 représente le graphe général pour concevoir un système embarqué:

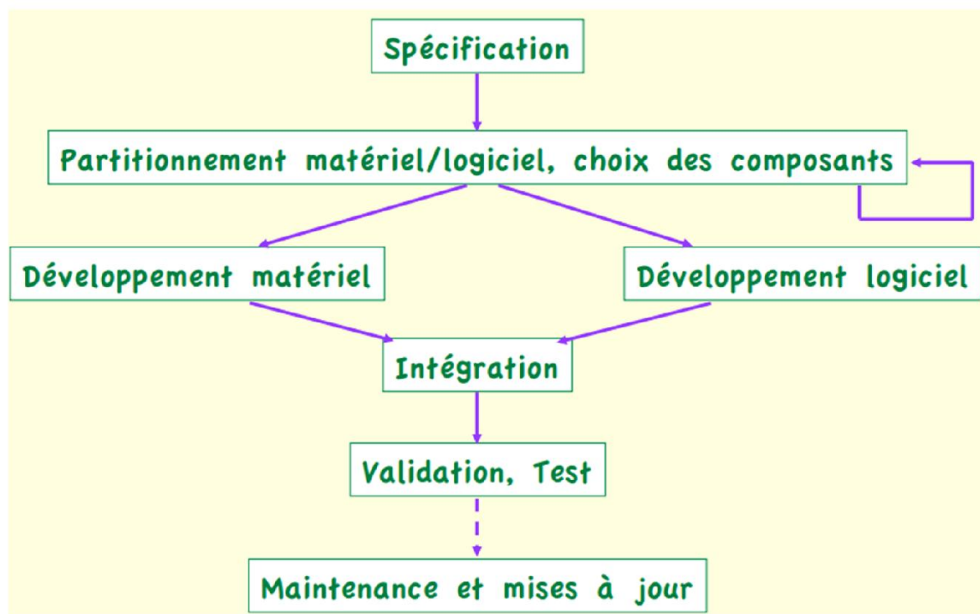


Figure III. 15: le graphe général pour concevoir un système embarqué.

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.

VI. 2. Conception Matérielle :

VI.2.1. Les graphes d'appels :

Les graphes d'appels sont des moyens graphiques 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 appelés. Les ports d'entrées et sorties sont organisées en groupes et placés en bas du graphe.

La figure III.16 représente le graphe d'appels pour le système embarqué dans le véhicule :

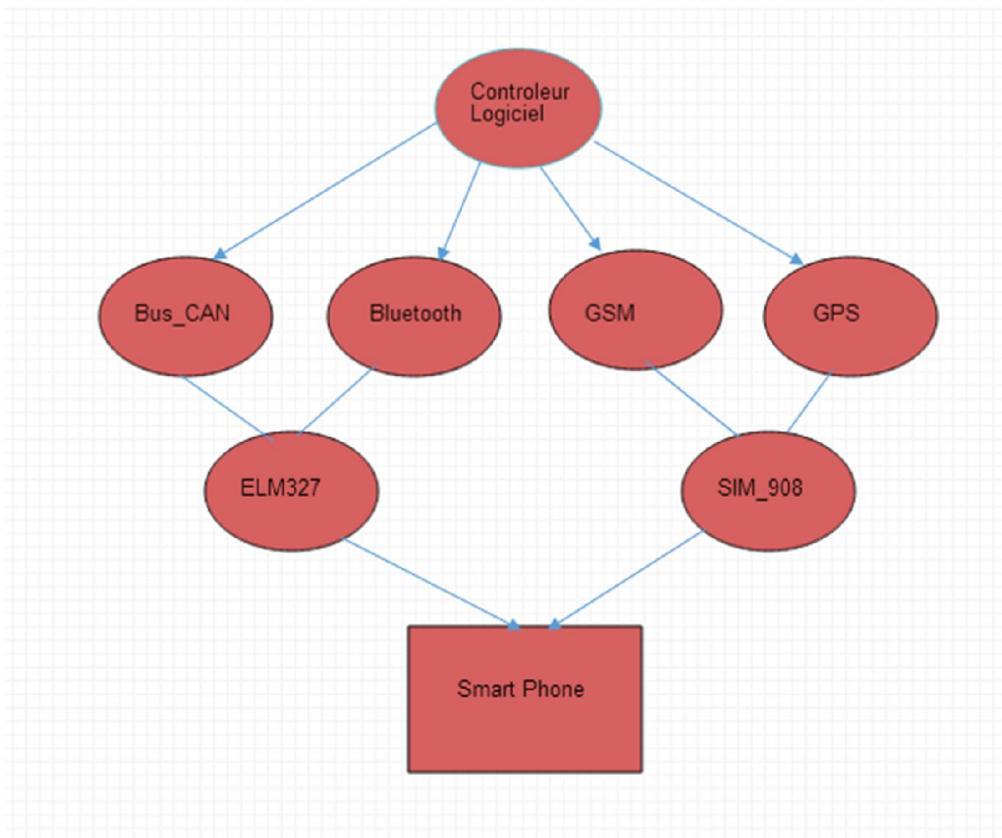


Figure III. 16 : le graphe d'appels pour le système embarqué dans le véhicule.

Remarque :

Le Bus CAN, Bluetooth, GSM, GPS, ELM 327 et SIM 908 dans ce graphe ne représentent pas des composants matériels mais du côté logiciels.

VI.2. 2. 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 convertie en un code de logiciel. On peut décomposer une tâche de trois manières différentes, comme le montre la figure suivante :

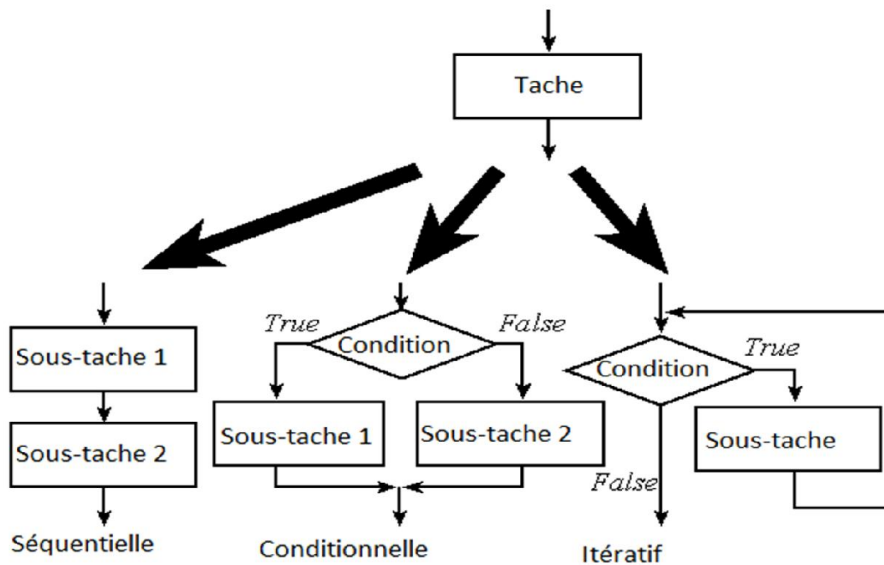


Figure III. 17 : Décomposition d'une tâche en sous tâches.

La figure III.18 représente le graphe de Raffinement successives pour le système embarqué dans le véhicule :

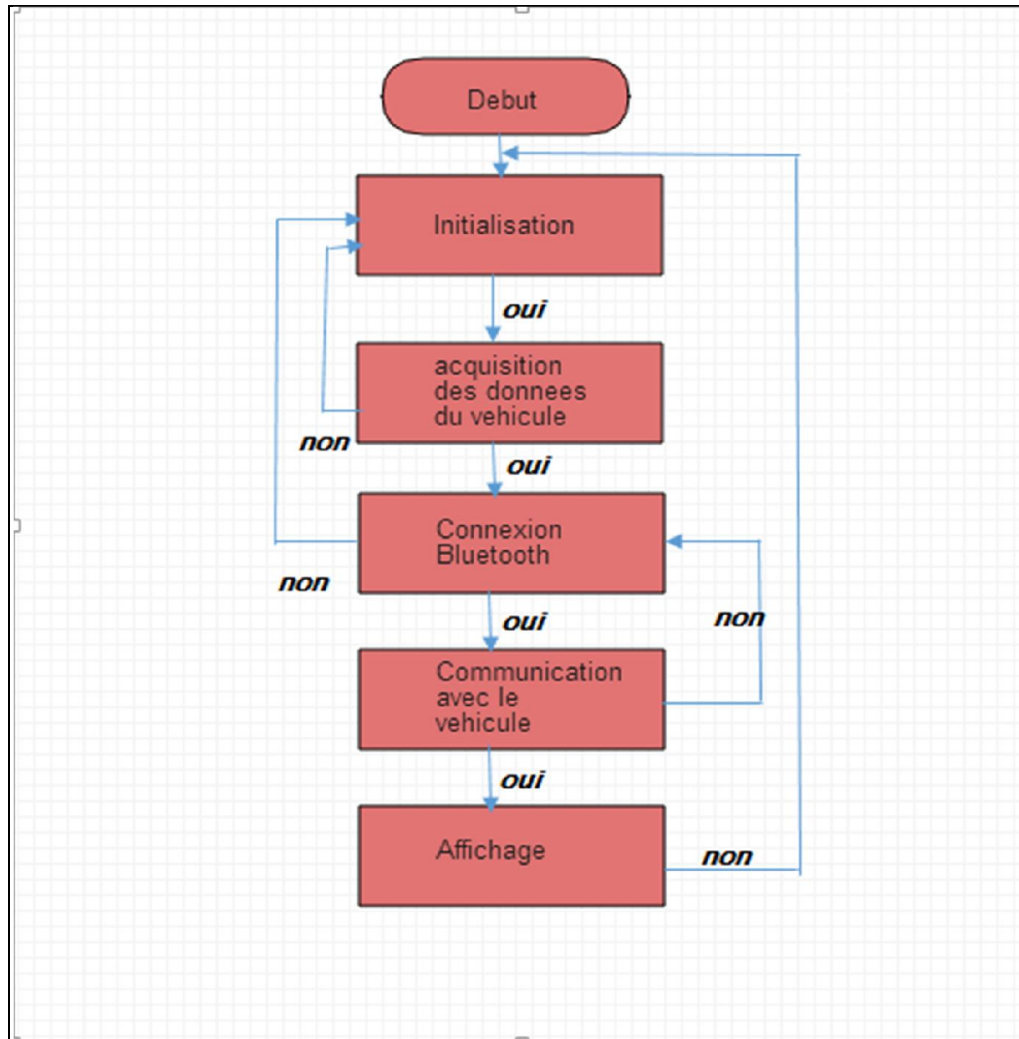


Figure III. 18 : Le graphe de Raffinement successives pour le système embarqué dans le véhicule.

La figure III.19 représente le graphe de Raffinement successives pour le système embarqué concernant la géolocalisation :

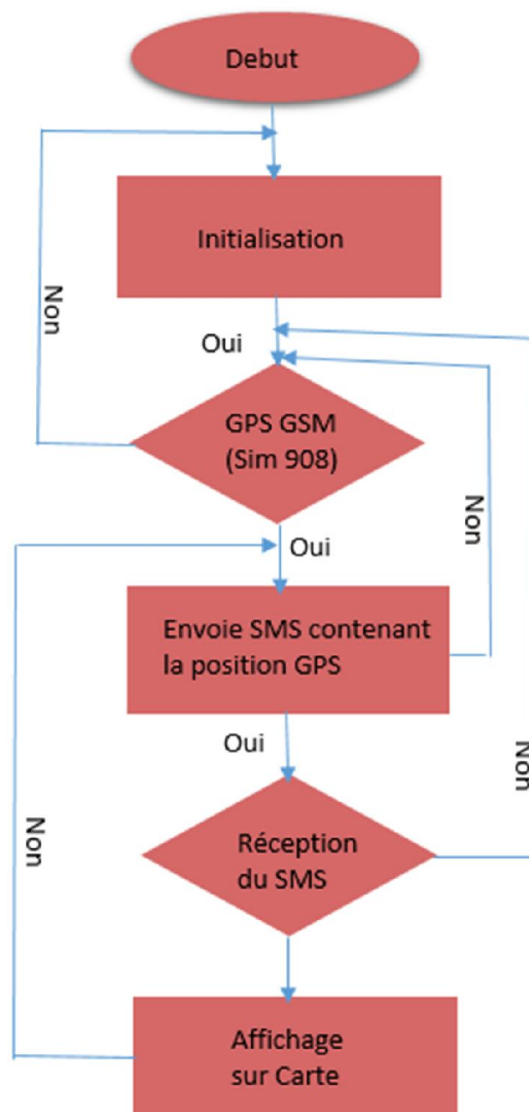


Figure III.19: Le graphe de Raffinement successives pour le système embarqué concernant la géolocalisation.

VI .3. Conception Logicielle :

VI.3.1. Méthodologie de conception :

Une méthodologie de conception correspond à un ensemble de méthodes que doit suivre le concepteur afin de réaliser son système, celui-ci consiste à la Spécification des besoins, la Conception, le Codage et les tests de validation.

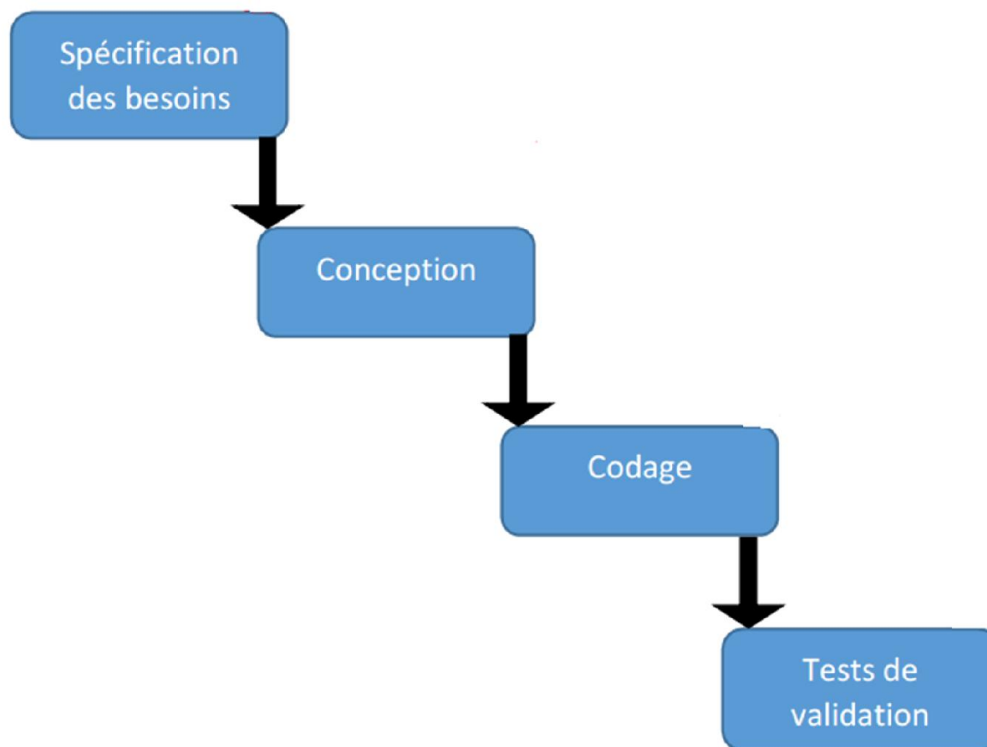


Figure III. 20: Méthodologie de conception.

Dans ce contexte, un langage de modélisation est souvent utilisé. En ce qui nous concerne, Pour la modélisation de la partie logicielle, nous allons nous inspirer du support graphique du langage UML afin de mieux expliquer les caractéristiques structurelles et comportementales de notre système.

Notre méthodologie peut être décrite par le schéma de la figure III.20 :

La figure III.21 donne le diagramme des cas d'utilisation de notre système.

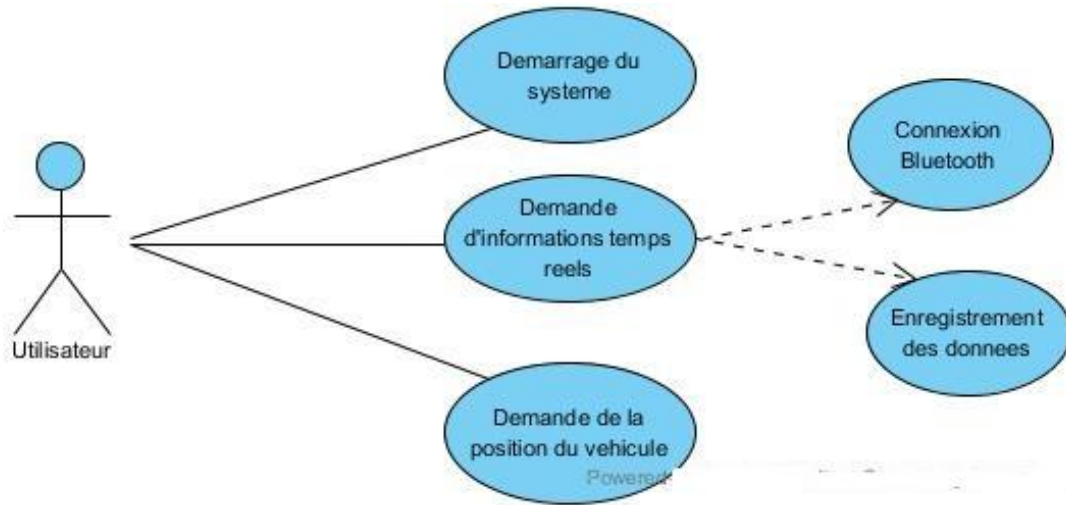


Figure III. 21: le diagramme des cas d'utilisation de notre système.

VI.3.2. Comportement du système logiciel :

Les diagrammes suivants vont nous permettre de voir avec précision l'interaction entre les différents modules du système logiciel dans les différents cas d'utilisation définis précédemment.

Le cas d'utilisation « démarrage du système » :

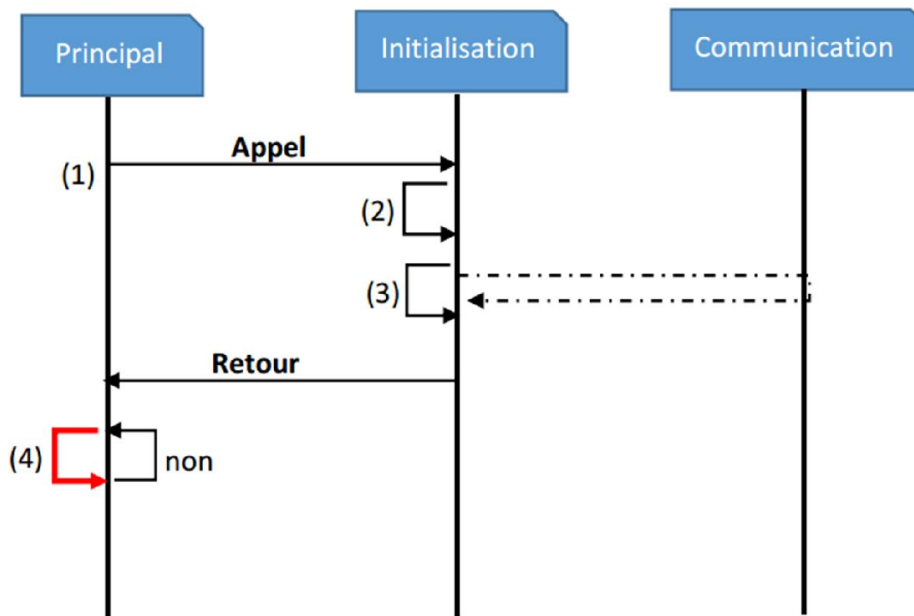


Figure III. 22: Diagramme de séquence du cas d'utilisation « démarrage du système ».

Description:

- (1) Lors de la mise en marche du système, le module « Principal » fait appel au module « Initialisation ».
- (2) Dans le module « Initialisation », l'appel est fait à la procédure qui initialise la liaison série.
- (3) Le même module fait appel à la procédure d'initialisation du système embarqué.
- (4) Le module « Principal » boucle pour attendre un évènement arrivant sur la liaison série, dans la boucle, un test est effectué pour savoir si l'évènement correspond à l'arrivée de la donnée désirée. Si c'est le cas, alors on fait appel au module, sinon on continue à scruter le port série.

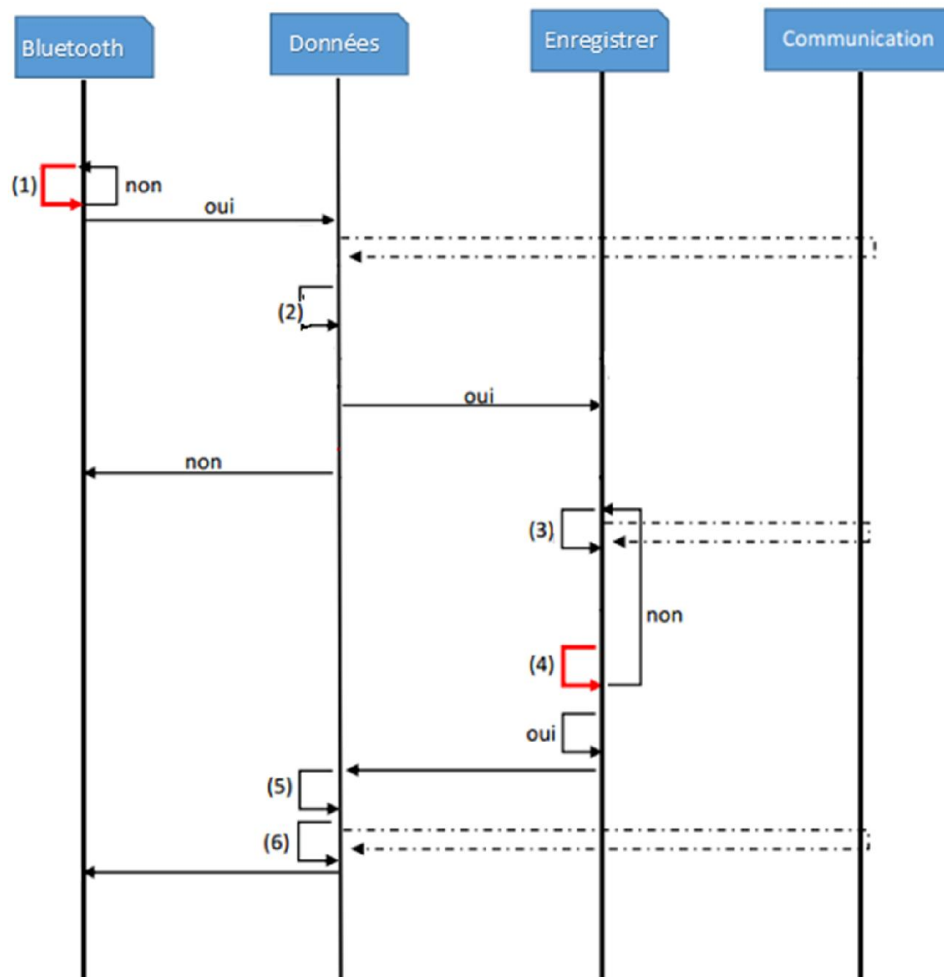
Le cas d'utilisation « Demande d'informations en temps réel » :

Figure III. 23 Diagramme de séquence du cas d'utilisation « Demande d'informations en temps réel ».

Description :

- (1) Le module « Bluetooth » demande une liaison entre les deux périphériques smartphone et système embarqués.
- (2) Une fois la liaison est établie la communication entre les deux modules débute via les commandes AT.
- (3) L'acquisition des données peut être enregistrée dans un fichier.
- (4) Si y a pas de données à recevoir il faut bien vérifier la communication Bluetooth.
- (5) On récupère le contenu reçu, s'il y a correspondance.
- (6) Fin de la communication.

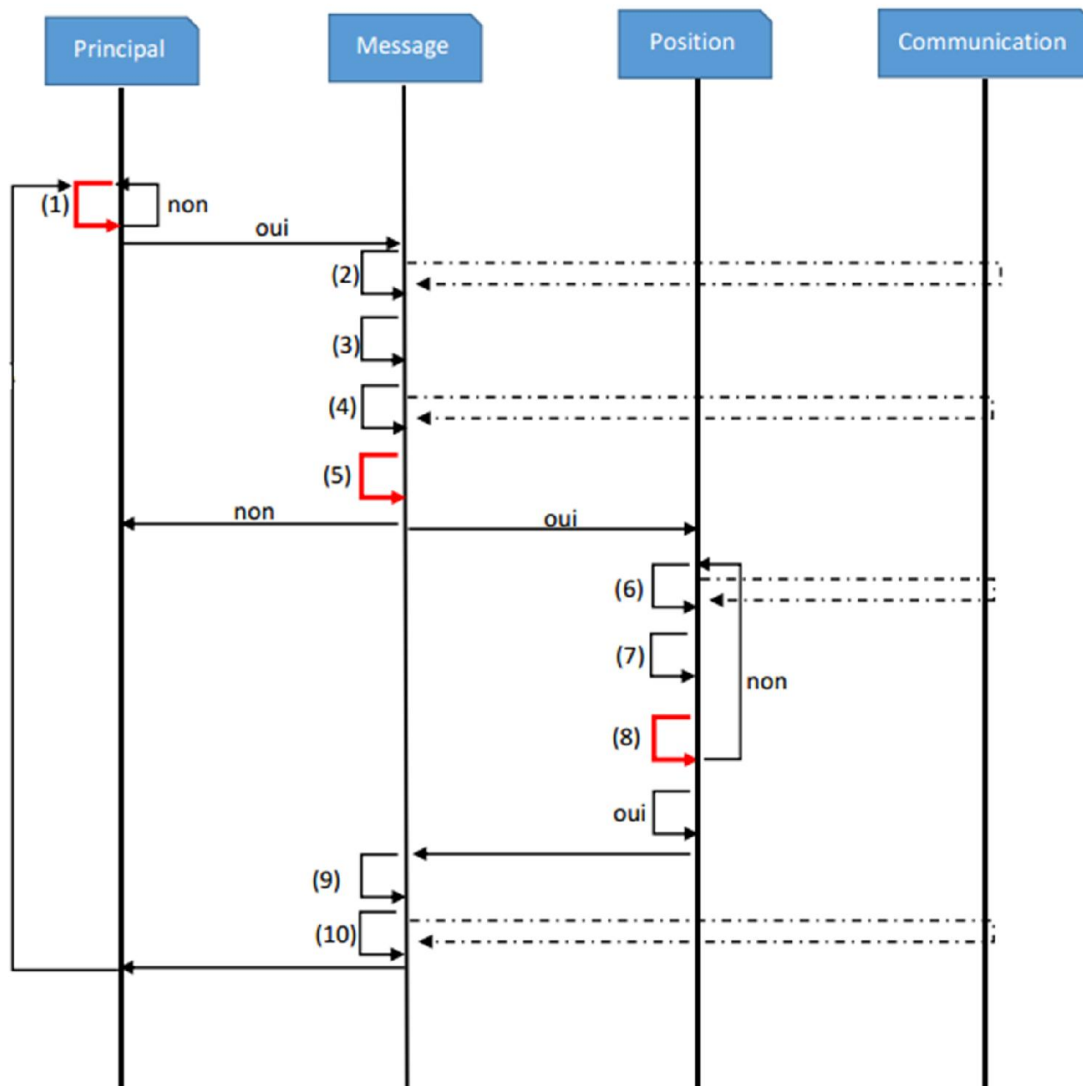
Le cas d'utilisation « Demande de la position du véhicule » :

Figure III.24 : Diagramme de séquence du cas d'utilisation Demande de la position du véhicule.

Description :

- (1) Le module « Principal » scrute l'arrivée d'un nouvel évènement, dès qu'il s'agit d'un message il fait appel au module « Message ».
- (2) Suite à l'arrivée d'un message, le module « Message» fait appel à la procédure permettant de lire le message à partir de la carte SIM.
- (3) Une fois le message sauvegardé, on fait appel à une procédure pour en extraire le contenu du message(SMS).
- (4) Afin de ne pas encombrer la mémoire de la carte SIM, le module « Message » supprime le message après avoir atteint un certains nombres.
- (5) On récupère le contenu reçu, s'il y a correspondance, on fait appel au module «Position ». Sinon, on revient à la scrutation.
- (6) Le module « Position », récupère la trame NMEA et l'enregistre en mémoire.
- (7) On extrait la longitude et la latitude de la trame et on les sauvegarde en mémoire.
- (8) La procédure de test, compare les valeurs des coordonnées récupérées à la valeur nulle, dans le cas où les valeurs sont incorrectes une autre demande de trame est effectuée, sinon le module construit les coordonnées finales.
- (9) Retour au module « Message » où on construit le message en réponse à l'utilisateur.
- (10) On Affiche ensuite le message sous forme de Marker sur la carte GoogleMAP.

VI.3.3. Diagramme de classe du système logiciel :

On peut représenter la structure modulaire du système logiciel par le diagramme de classes de la figure III.25 suivante:

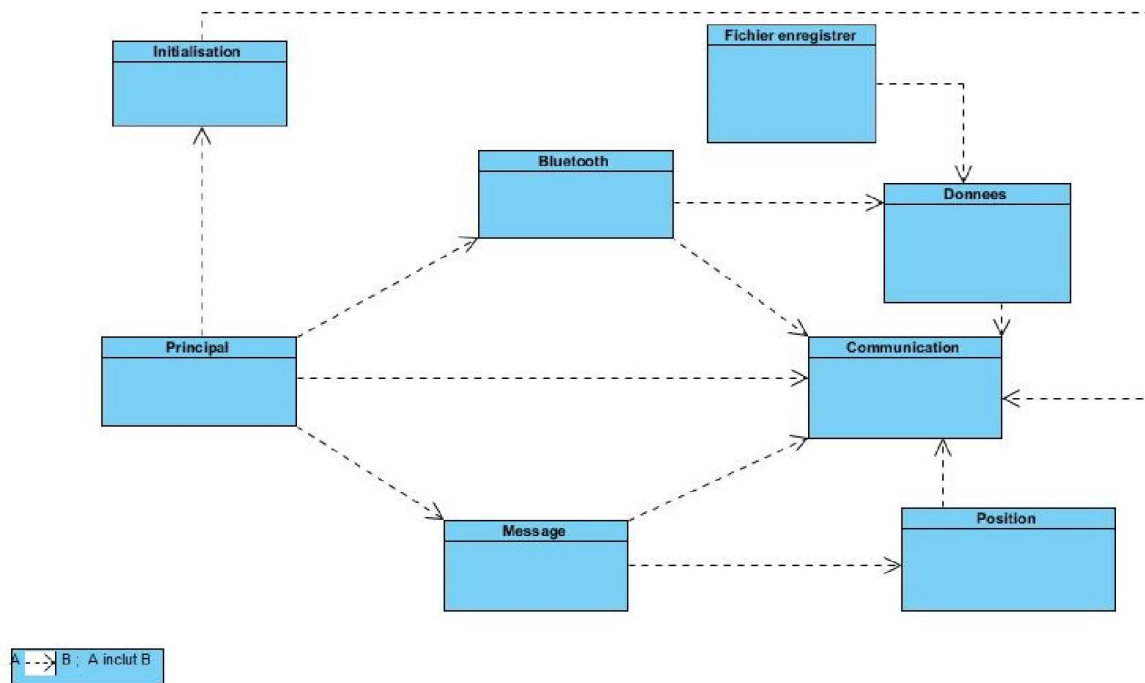


Figure III. 25: Diagramme de classes du système.

VII. 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 ainsi la conception pour la réalisation de cette application.

Dans le chapitre suivant, nous allons aborder la réalisation de notre système, les tests effectués, ainsi les interfaces IHM réalisées sous Android de notre application.

Chapitre IV:

Réalisation et test

I. 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.

II. Principe de fonctionnement du système :

La figure ci-dessous montre le schéma fonctionnel de notre système qui permet de récupérer en temps réel le régime moteur, la vitesse ...etc. d'un véhicule afin de lui faire un diagnostic et d'enregistrer ces données dans un fichier Excel, ainsi ce système permet de localiser le véhicule soit pour suivre un itinéraire spécifique, ou en cas de vol ou de perte de trouver le véhicule et le localiser.



Figure IV 1 Principe de fonctionnement du système.

Présentation des applications du système :

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

- un système embarqué véhicule (OBD-II).
- un système embarqué pour la localisation (GPS).

III. Les interfaces IHM de notre application :

- ✚ **Lancement de l'application** : l'utilisateur peut faire un choix en cliquant sur l'un des boutons en dépend du service désiré comme le montre la figure suivante :

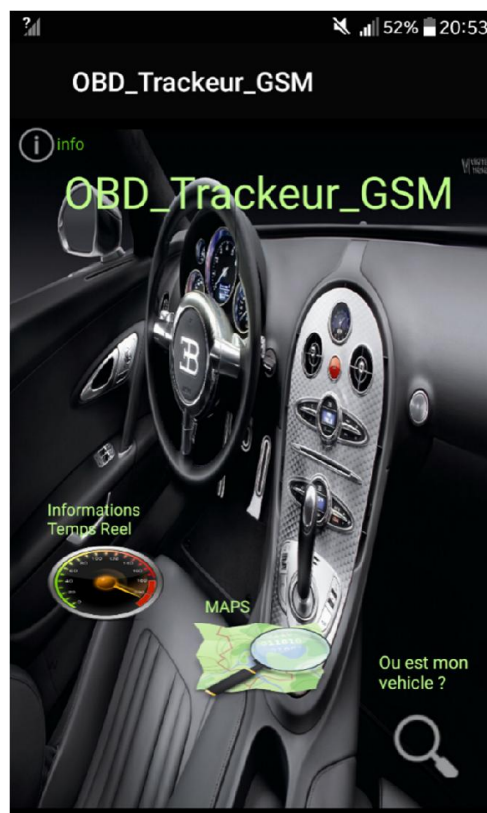


Figure IV 2 Interface principale de l'application.

Le conducteur peut visualiser des informations à propos de sa vitesse, son régime moteur, ...etc en temps réel dans une interface graphique comme le montre la figure ci-dessous tout en acceptant la communication Bluetooth avec l'interface de diagnostic, ainsi il peut accéder dans différentes fenêtres tout en choisissant l'option qu'il veut comme le montre les figures suivantes :

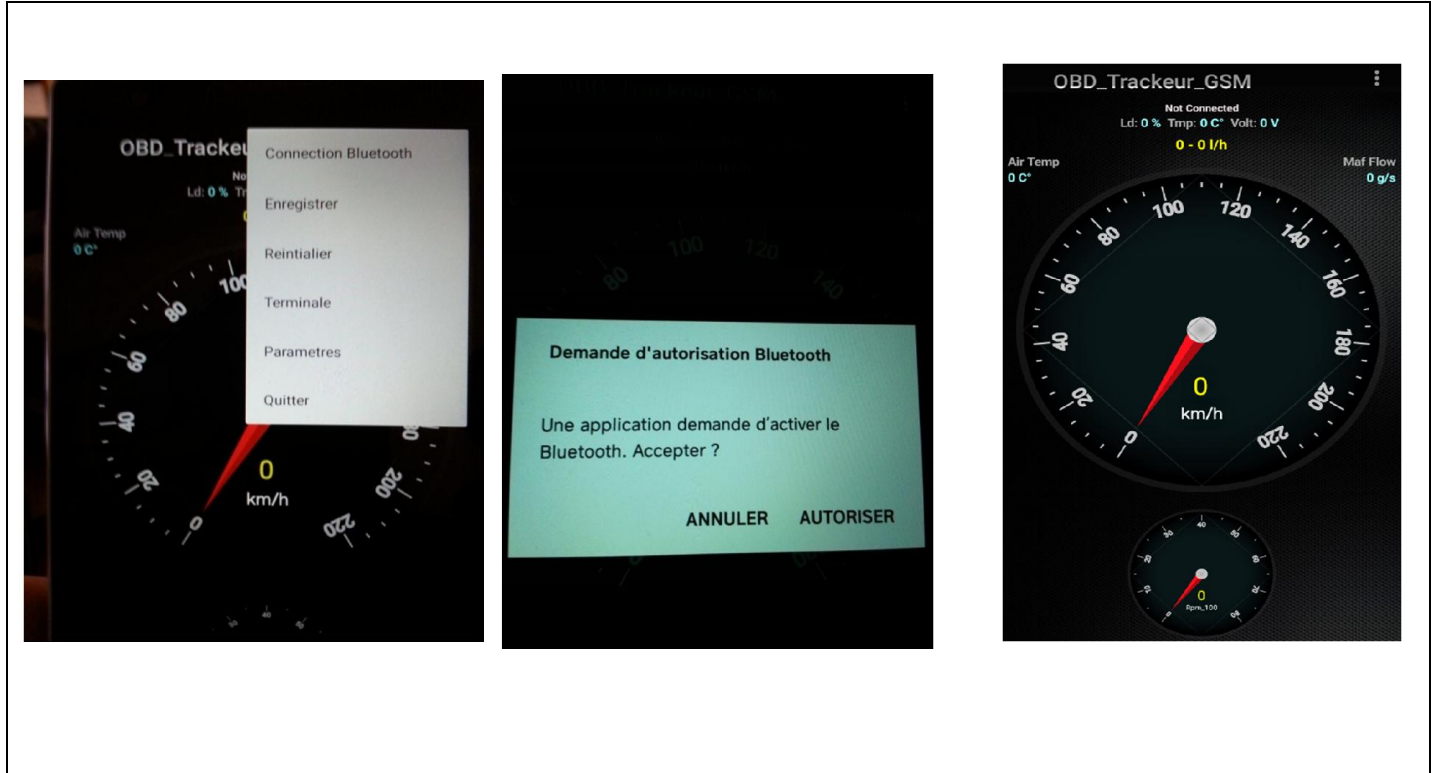


Figure IV 3 Les options à choisir, L'autorisation de connectivité Bluetooth et la visualisation des informations temps réel.

✚ Enregistrement des données en temps réel :

Ce service se fait sous cette interface et on choisit les données à enregistrer comme en travail juste avec le mode on à la Figure IV 4:



Figure IV 4 : Enregistrement des données en temps réel.

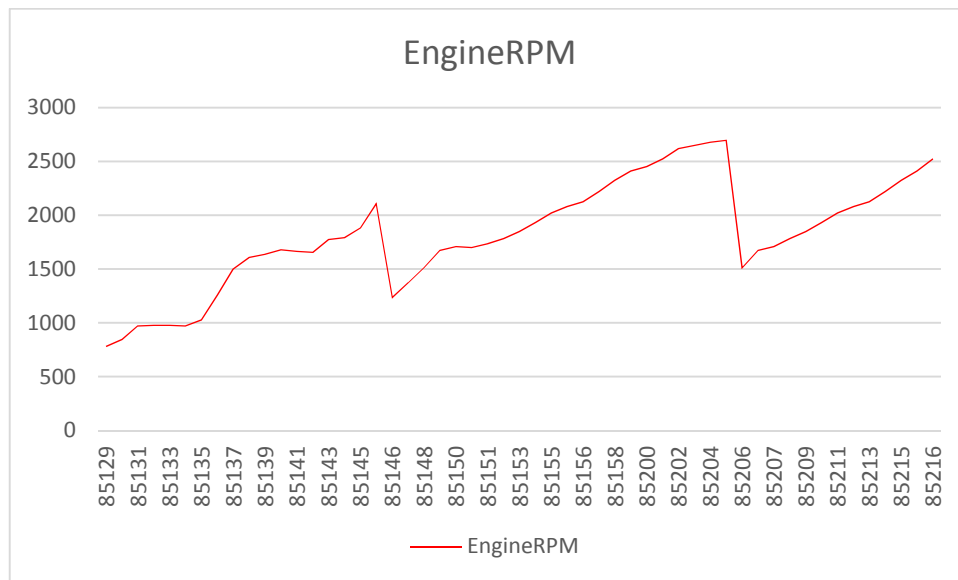
Le résultat est un fichier Excel qui contiendra les paramètres sélectionnés en haut juste y on a des résultats qui seront pas afficher à cause des capteurs du véhicules car ça dépend d'un constructeur à l'autre on a obtenu le fichier suivant :

A	B	C	D	E	F	G	H
EngineRPM	VehicleSpeed	stanceSinceCodesClear	FuelLevellnput	IntakeAirTemperature	keManifoldAbsolutePres	kuntimeSinceEngineStae	ehicleIdentificationNumber
961	0						
936	0						
961	0						
978	0						
978	0						
1132	0						
1163	0						
1266	0						
1372	0						
2453	0						
2525	0						
2618	0						
2679	0						
899	0	20276	NO DATA>	25	39	NO DATA>	53 52 36 42 34 47 2: 44
848	0	20276	NO DATA>	25	33	NO DATA>	53 52 36 42 34 47 2: 44
1299	0	20276	NO DATA>	25	31	NO DATA>	53 52 36 42 34 47 2: 44
859	0	20276	NO DATA>	25	29	NO DATA>	53 52 36 42 34 47 2: 44

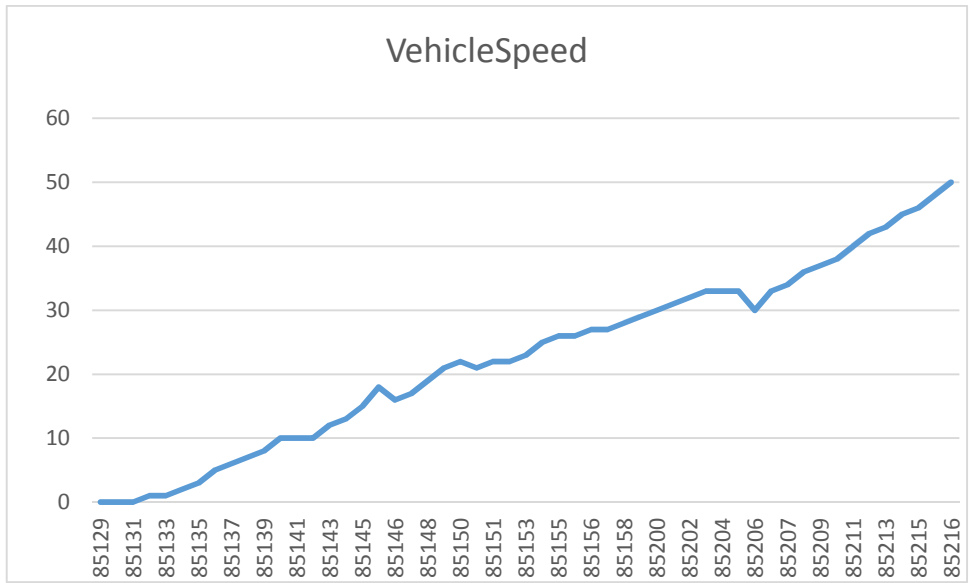
Figure IV 5 : Exemple d'un fichier Excel.

Ce dernier peut être utilisé afin de générer des graphes et d'étudier l'évolution de la vitesse par rapport au régime moteur et faire des différentes études sois de ce qui concerne le véhicule ou savoir le comportement du conducteur.

Remarque : 0851xx représente hhhmss (heure, minute, seconde).



Graphe 1 Représente la variation du régime moteur (rpm tr/min).



Graphe 2 Représente la variation de la vitesse (Km/h).

Google Map offre des services de localisation cette interfaces montre la carte et la position de notre véhicule :

Suite à la réception de la position dans un SMS comme le montre la Figure IV.5, nous avons visualisé la position du véhicule sur Google Map tout comme on peut tracer un itinéraire, la Figure IV 6 représente la position et l’itinéraire.

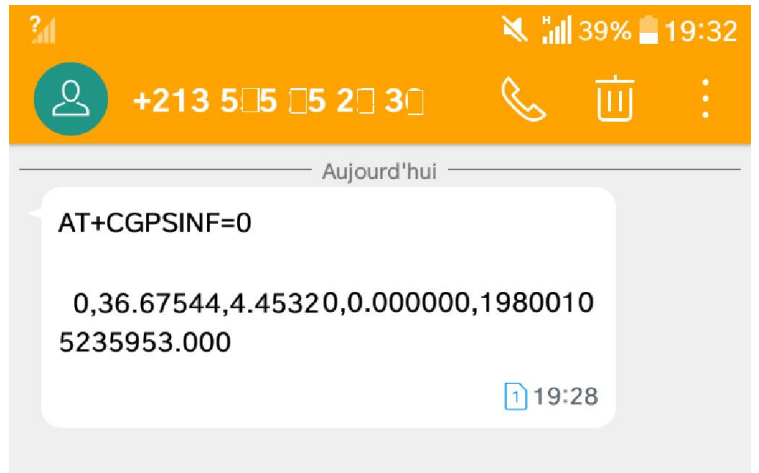
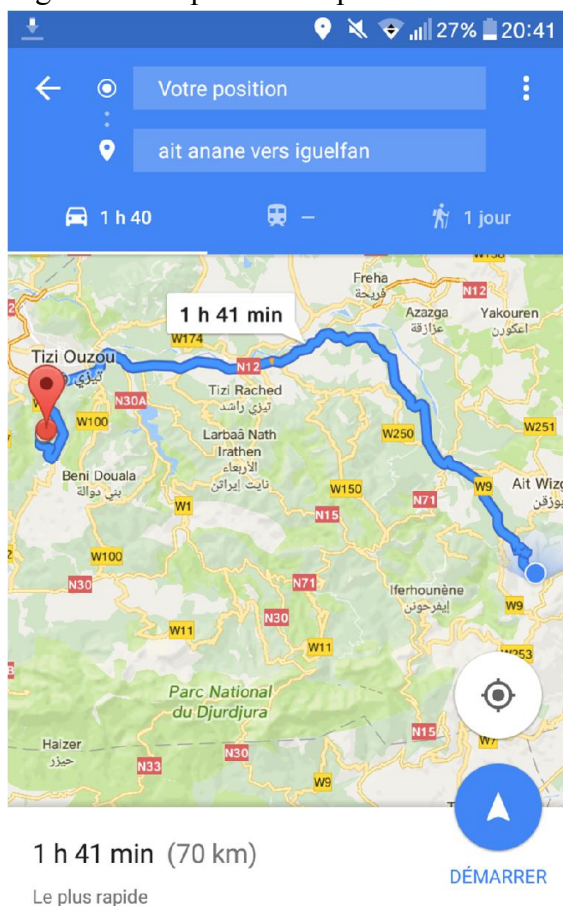


Figure IV 5 SMS contenant longitude et latitude.

Figure IV 6 : Interface de géolocalisation Google Map.

IV. Les tests :

a. Véhicules testés :

Après avoir fait plusieurs tests, nous avons constaté que les marques Volkswagen et Skoda utilisent la communication bus CAN, qui est compatible avec le Bus Can Shield mais vu la non disponibilité d'un véhicule allemand on a décidé de rendre notre système standard avec L'ELM327 on a fait des tests sur différents véhicules.

➤ Test N°1 : Volkswagen Passat :



Figure IV 7 : Volkswagen Passat.

Le tableau montre les principales caractéristiques d'une passat :

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Volkswagen	Passat	1.9 tdi	2012	Diesel	CAN 11bits 500 kb

Tableau 10: Informations sur le véhicule du teste N°1.

✚ Matériels Utilisés dans le test :

Pour le premier test nous avons utilisé la Carte Arduino MEGA2560, le CAN bus Shield et le logiciel arduino.

✚ Objectif du 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.

✚ Résultat du 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 figure suivante montre les données affichées dans le compteur minute véhicule et les résultats obtenus sont dans la figure ci dessous :

```
CAN-Bus Demo
CAN Init ok

Vehicle Speed:
2 km/h
RPM:
1129 tr/min
Vehicle Speed:
5 km/h
RPM:
1428 tr/min
Vehicle Speed:
8 km/h
RPM:
1623 tr/min
.....
```

Figure IV 8 : Résultats obtenus lors du test.

➤ **Test N°2 : Renault Symbole :**



Figure IV 9 : Renault Symbole.

Le tableau montre les caractéristiques de la Symbole :

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Renault	Symbole	Essence 1.6	2016	Essence	KWP FAST 29 bits

Tableau 11: Informations sur le véhicule du Test N°2.

➤ **Test N°3 : Hyundai Accent :**



Figure IV 10 : Hyundai Accent.

Le tableau 11 montre les caractéristiques de l'accent :

Marque	Modèle	Motorisation	Année de production	NRJ	Protocole
Hyundai	Accent	Essence 1.5	2012 2016	Essence	ISO 9141-2 11 bits

Tableau 12: Information sur le véhicule du Test N°3.

✚ **Matériels Utilisés :**

Les matériaux utilisés dans les deux derniers véhicules sont :

Elm327, Une Carte Arduino MEGA 2560, un module sim908 et un Smartphone Android.

✚ **Objectif de nos tests:**

Ces tests récapitulent tout notre travail en rassemblant les résultats valides de nos tests précédents, il permet de vérifier :

- l'affichage des données acquises du véhicule dans l'application Android.
- La localisation en cas de vol ou de perte.
- l'affichage du rapport dans un fichier Excel.

b. Les figures suivantes représentent les résultats obtenus lors du test final :

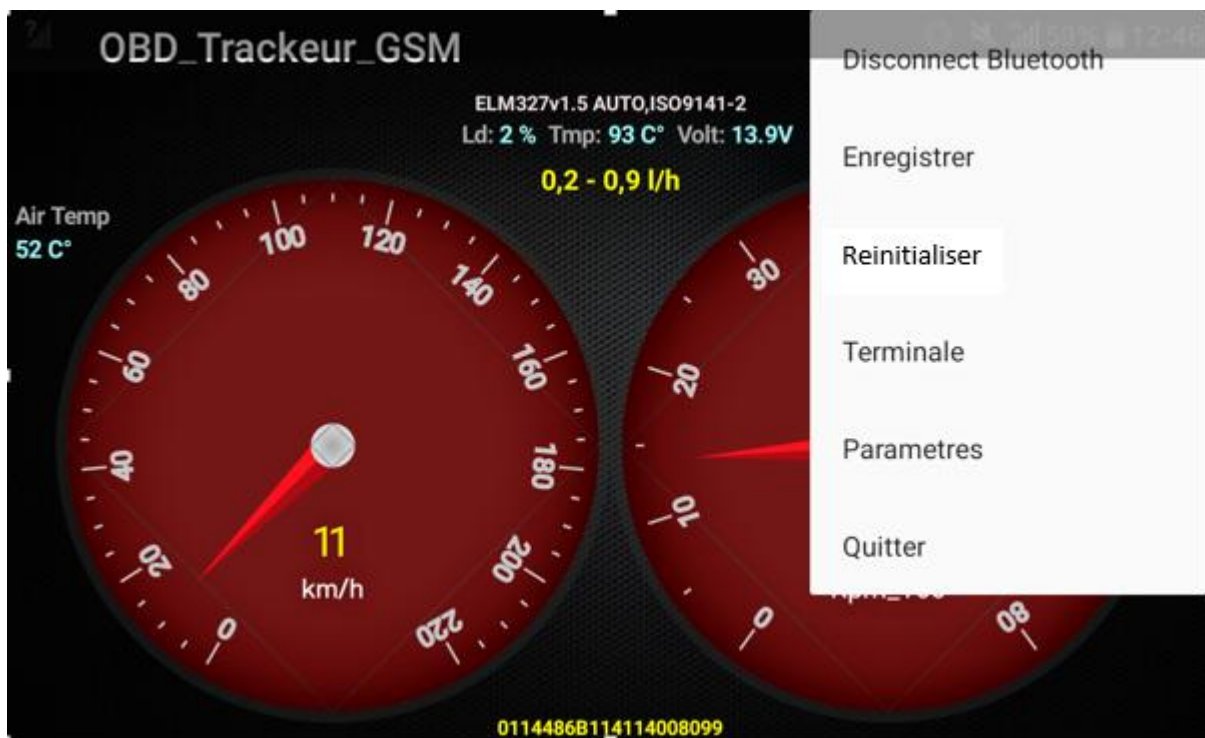


Figure IV 11 : Résultats du test final avec l'ongle des choix.

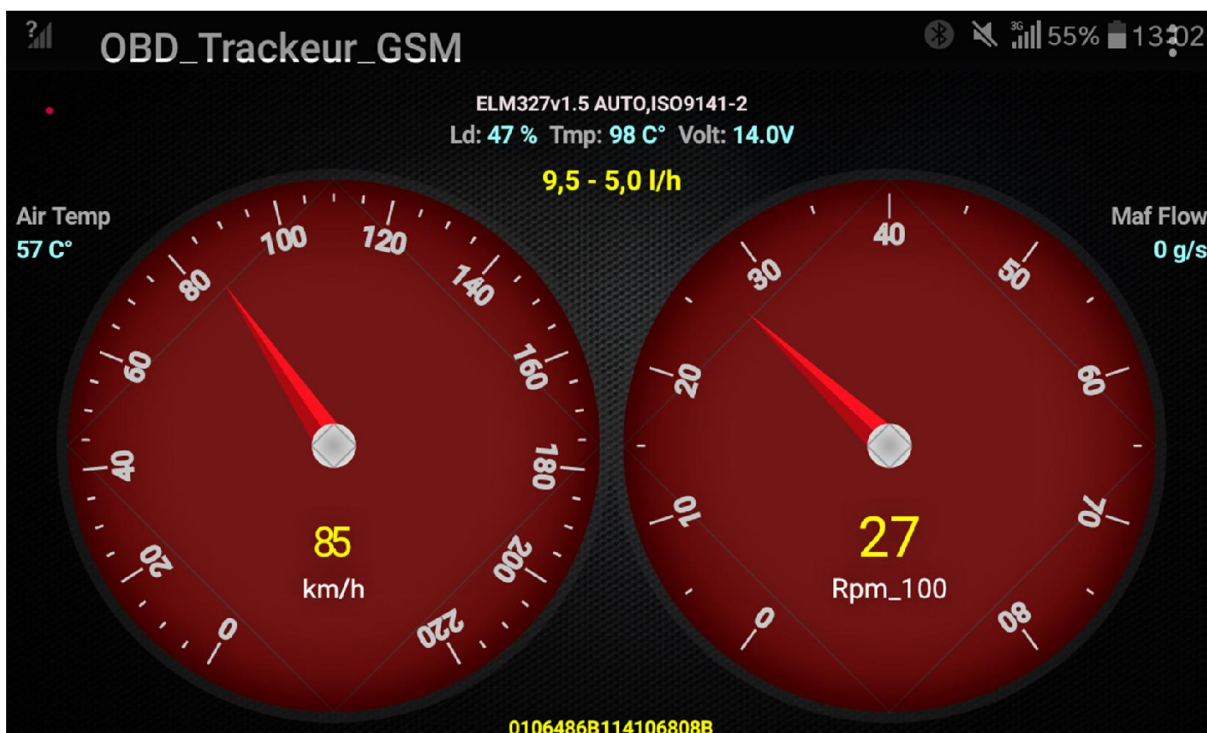


Figure IV 12 : Résultats du test final.

Ce dernier test nous a permis de valider notre travail dans lequel 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 un Smart Phone Android dans un véhicule.
- la localisation du véhicule.
- L'enregistrement des données.

IV. 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 générale

Conclusion générale :

Le principal objectif de ce travail est la conception et la réalisation d'un système de diagnostic automobile ayant pour fonction la récupération des données (vitesse, régime moteur, ...etc, juste dans notre système on a exploité le model uniquement), la vérification de leur conformité afin de mieux assurer la sécurité du conducteur et éviter les pannes non évidente, et ce via une communication sans fil basée sur la technologie « Bluetooth».et notamment de concevoir et développer un système permettant de localiser un véhicule en cas de vol. L'utilisation étendue des réseaux téléphoniques sans fil et la disponibilité de divers moyens de localisation.

Ce projet nous a permis de participer à une expérience intéressante qui nous a renforcé et enrichit 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'approfondissement de nos connaissances sur les cartes de développement (arduino, bus can shield) et des composants électronique (Bluetooth, Sim908).

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 pour permettre une meilleure identification des véhicules.
- Il peut aussi être programmé manière à ce que dès que le véhicule quitte un rayon prédéfini, un message soit envoyé vers l'utilisateur.
- améliorer l'application pour qu'elle permette de commander le véhicule à distance pour activer la climatisation automobile et le verrouillage des portes ou commander la vitesse du véhicule pour qu'elle ne dépasse pas les 120 km/h par exemple.

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 car la seule et unique chose qui pourra mettre une limite aux perspectives c'est bien notre imagination.

Bibliographie :

[1] : <http://www.panne-automobile.com/pourquoi-le-multiplexage-en-automobile.html>

[2]:<http://www.usinenouvelle.com/article/cablagele-multiplexage-a-la-conquete-de-l-automobilele-multiplexage-reduit-la-longueur-du-cablage-et-le-nombre-de-connexions-et-du-meme-coup-ameliore-la-fiabilite-des-circuits-electriques-des-vehicule.N80873>.

[3] : <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-Bus-CAN.htm>.

[4] : <http://www.outilsobfdfacile.fr/presentation-de-l-obd.php>

[5] : <http://www.conrad.fr/ce/fr/content/obd/OBD-das-elektronisches-Ueberwachungssystem-in-Ihrem-Fahrzeug>

[6] : <http://www.elm327.fr/norme-obd/modes-obd/>

[7] : <http://users.ece.utexas.edu/~valvano/Volume1/E-Book/>

[8] : <http://www.commentcamarche.net/contents/108-bluetooth-comment-ca-marche>

[9] : <https://fr.wikipedia.org/wiki/G%C3%A9olocalisation>

[10]:<http://www.orange-business.com/fr/blogs/relation-client/contact-multicanal/la-geolocalisation-definition-usages-et-limites>

[11] : <http://www.geolocalisation-vehicule.fr/localisation-par-gsm/#0>

[12] : <http://gps-tpe.e-monsite.com/pages/le-fonctionnement-du-gps/triangulation.html>

[13]:http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560

[14]: <https://www.generationrobots.com/fr/402252-shield-can-bus.html>

[15]: <http://www.outilsobfdfacile.fr/interface-diagnostic-elm-327.php>

[16]: https://fr.wikipedia.org/wiki/Android_Studio

Annexe

A. Les PIDs du protocole OBD-II :

➤ 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.

➤ 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

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			
Binaire	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1
PID Supporter	Oui	Non	Oui	Oui	Oui	Oui	Oui	Non	Non	Non	Non	Oui	Oui	Oui	Oui	Oui
N° PID	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

Donc, PIDs pris en charge sont:

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

B. Les commandes AT du GSM:

➤ Définition :

Les commandes AT ne sont rien d'autres que des instructions qui permettent l'accès aux fonctions d'un téléphone portable par l'intermédiaire d'un terminal. Spécifiées par un standard de télécommunication européen (ETS), ces commandes s'inspirent des commandes universelles permettant de piloter un modem. Les téléphones portables doivent être construits de manière à respecter les deux normes suivantes :

- GSM07.07 qui permet l'accès aux fonctions générales du téléphone ;
- GSM07.05 qui concerne la gestion des SMS.

Trois entités sont définies dans les textes officiels qui traitent le GSM :
TE : Terminal Equipment qui peut être un ordinateur ou un microcontrôleur (envoi et affiche les commandes).

TA : Terminal Adaptator qui est l'interface entre l'utilisateur et le mobile.
ME : Mobile Equipment qui correspond par exemple à un téléphone portable.

➤ Syntaxe d'une commande AT :

Le préfixe «AT» doit être mis au début de chaque commande, et pour terminer la ligne de commande on Entre <CR>.

On peut envoyer une commande AT de trois manières :

- **AT+CXXX=? (Commande de test)**, retourne la liste des paramètres utilisables avec la commande CXXX.
- **AT+CXXX? (Commande de lecture)**, retourne le ou les paramètres en cours associés à la Commande CXXX.
- **AT+CXXX=<xxx> (Commande d'écriture)**, applique le ou les paramètres <xxx> à la commande CXXX.

Une fois la commande envoyée, le téléphone doit y répondre favorablement ou non et la réponse retournée est de la forme :

<CR><LF>OK<CR><LF> si la commande est accepté.

-<CR><LF>ERROR<CR><LF> (accompagné éventuellement d'un message décrivant la nature de l'erreur), si la commande n'est pas reconnue, ou que le ME rencontre un problème lors de son exécution.

Il faut noter que le <CR> signifie *Carriage Return* (retour chariot) et le <LF> signifie *Line Feed* (Nouvelle ligne).

a. La norme GSM07.07 :

La norme GSM07.07 regroupe environ 80 commandes permettant d'accéder à toutes les fonctions du ME, le tableau suivant nous donne quelques exemples.

Commandes	Fonction
AT+CGSN	Identification numéro de série (IMEI)
AT+CGMI	Identification fabricant
AT+CPBS	Sélectionne un répertoire téléphonique
AT+CPBR	Lecture du répertoire téléphonique
AT+CSCS	Alphabet utilisé par le TE
AT+CCLK	Horloge
AT+CLIP	Présentation du numéro

Exemple de réponse : (la commande AT+CGSN) :

Commande de teste : AT+CGSN= ?

OK

Si erreur

+CME ERROR :<err>

Commande de lecture : AT+CGSN

+CGSN :<sn>

Retourne le numéro de série du ME nommé IMEI (International Mobile station Equipment Identity)

Si erreur

+CGMI ERROR :<err>

b. La norme GSM07.05 :

La norme GSM07.05 spécifie les commandes AT permettant la gestion des SMS comme on peut le voir sur les exemples du tableau suivant :

Commandes	Fonction
AT+CSMS	Sélection du service de messagerie
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSDH	Affiche en mode TEXT le paramétra SMS
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMGW	Écriture d'un SMS

Exemple de réponse : (la commande : AT+ CSCA)

Commande de teste : AT+CSCA= ?

OK

Si erreur

+CMS ERROR : <err>

Commande de lecture : AT+CSCA ?

+CSCA :<sca> , <tosca>

OK

Si erreur

+CMS ERROR : <err>

Commande d'écriture : Si mode TEXTE (+CMGF=1) : AT+CSCA=<sca> [, <tosca>]

OK

Définit l'adresse du centre de messagerie à utiliser pour l'envoi des SMS.

Si erreur

+CMS ERROR : <err>

Commande AT

AT + CGPSSTATUS

La description

La commande AT + CGPSSTATUS AT renvoie l'état du GPS.

L'information d'état suivante est affichée,

"Localisation inconnue", "Location Not Fix", "Location 2D Fix", Localisation 3D Fix ".

Exemples

AT + CGPSSTATUS?

+ CGPSSTATUS: Localisation inconnue

D'accord

Commande AT

AT + CGPSINF

La description

La commande AT + CGPSINF AT renvoie l'information de positionnement GPS actuelle.

Utilisation: AT + CGPSINF = ?

Si cette option est définie sur 0, les informations de localisation GPS suivantes sont renvoyées,

Exemples

AT + CGPSINF = 0

0,0.000000,0.000000,0.000000,19800106000016.000,0,0,0.000000,0.000000

D'accord

C. Le standard NMEA :

NMEA (National Marine Electronics Association) est une association américaine de fabricants d'appareils électroniques maritimes, basée à Severna Park au Maryland (États-Unis d'Amérique). Elle est à l'origine de nombreux standards et en particulier du Standard NMEA-0183.

NMEA 0183 est une spécification pour la communication entre équipements marins, dont les équipements GPS. Sous ce standard, toutes les données sont transmises sous la forme des caractères ASCII, tous imprimables, ainsi que les caractères [CR] Retour Chariot et [LF] Retour à la ligne. Les données sont transmises sous forme de trames.

Dans ce qui suit, le Standard NMEA est défini comme étant le protocole de transmission des données entre les équipements électroniques liés au GPS.

I- Format de la trame NMEA 0183 :

Chaque trame commence par le caractère \$

Suivi par un groupe de 2 lettres pour l'identifiant du récepteur. ID

Exemples:

- GP pour Global Positioning System.
- LC Loran-C receiver.
- OM Omega Navigation receiver.
- II Integrated Instrumentation (eg. AutoHelm Seatalk system).

ANNEXE

83

Puis un groupe de 3 lettres pour l'identifiant de la trame (MSG). Exemples :

- GGA : pour GPS Fix et Date.
- GLL : pour Positionnement Géographique Longitude -Latitude.
- GSA : pour DOP et satellites actifs.
- GSV : pour Satellites visibles.
- VTG : pour Direction (cap) et vitesse de déplacement (en noeuds et Km/h).
- RMC: pour données minimales exploitables spécifiques.

Suivent ensuite un certain nombre de champs séparés par une "virgule". La virgule permet la dé-concaténation des données dans le programme de traitement des données, calculateur, navigateur.

Et enfin un champ optionnel dit checksum précédé du signe *, qui représente le « OU exclusif » de tous les caractères compris entre \$ et * (sauf les bornes \$ et *), certaines trames exigent le checksum.

Suit la fermeture de la séquence avec un [CR][LF].

Une trame comporte au maximum 82 caractères.

II- Exemple de trame (GLL) :

\$GPGLL,4916.45,N,12311.12,W,225444,A

4916.46,N = Latitude 49 deg. 16.45 min. Nord.

12311.12,W = Longitude 123 deg. 11.12 min. West (ouest)

225444 = Acquisition du Fix à 22:54:44 UTC

A = Données valides

Pas de checksum

La trame manipulée par notre système :

0,357.469332,3643.701357,135.272675,20150530101952.000,42,11,0.000000,185.987579

(1) (2) (3) (4) (5) (6) (7) (8) (9)

(1) Type de la trame

(2) La longitude

(3) La latitude

(4) L'altitude

(5) Date et heure aaaammjjhhmmss

(6) TTFF (time to first fix) temps la 1ère fixation en seconde.

(7) nombre de satellites en vue

(8) Vitesse

(9) route sur le fond.

D. La technologie Machine To Machine (M2M).

Le M2M résulte de la convergence de trois familles de technologies : des objets intelligents reliés par des réseaux de communication avec un centre informatique capable de prendre des décisions. Les usages du Machine To Machine sont multiples : La gestion de flotte, la gestion de la chaîne d'approvisionnement (Supply Chain Management), la télésurveillance, la surveillance des biens et des personnes...etc.

I. Définition :

M2M consiste en l'utilisation d'un équipement (capteur, compteur, etc.) pour capturer un événement (température, mesure sismique, etc.) qui est relayé à travers un réseau de communication mobile, fixe ou hybride à une application (ex. du système d'information de l'entreprise qui utilise ces équipements M2M). Cette dernière traduit l'événement capturé en des informations significatives. Une solution M2M est le résultat d'une interaction continue entre les équipements M2M, les réseaux de communication et les applications.

II. Architecture :

Comme indiqué sur la figure C.1, le M2M permet un échange bidirectionnel d'informations entre l'équipement M2M et l'application. L'information reçue est ensuite traitée. Dans de nombreux cas, le M2M implique un groupe de « devices » semblables interagissant avec une application.

Dans certains cas, les équipements dans le groupe ne peuvent pas directement communiquer avec l'application du fait de leurs capacités limitées. Dans ces conditions, un équipement de médiation (GW, Gateway) est nécessaire, permettant de consolider la communication.

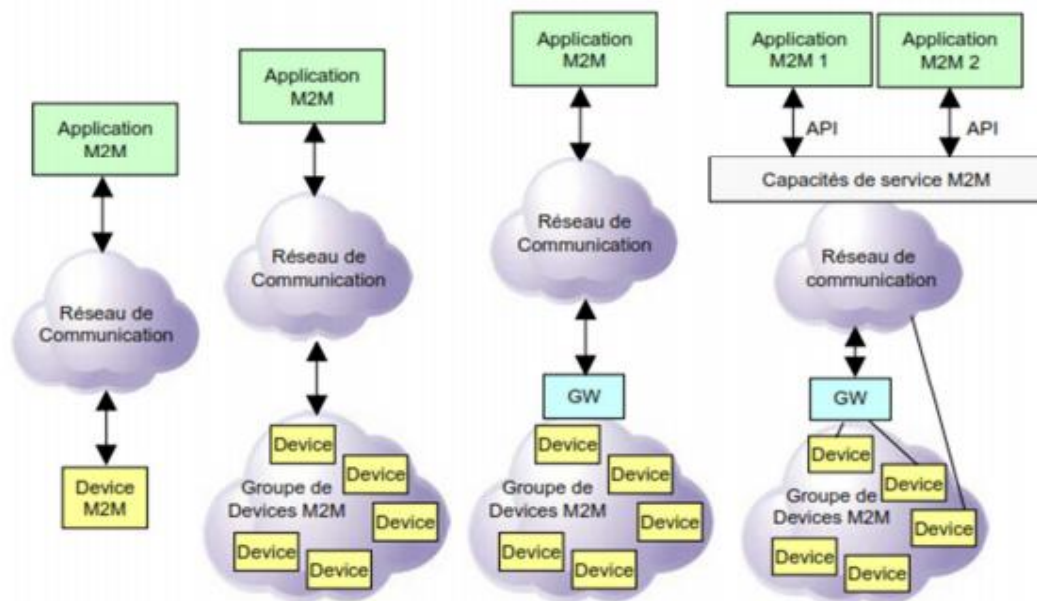


Fig.1. Architecture M2M.

III. Operations principales:

Dans toute solution M2M, il y a quatre opérations principales: la récolte de données, son transport et son traitement et le réveil du device afin qu'il puisse émettre un rapport de données non programmé.

- Réveil pour envoi de données (domaine des télécommunications) : Les devices M2M sont généralement programmés pour se réveiller à intervalle de temps fixe (e.g., toutes les heures), réaliser des mesures, s'attacher au réseau, établir une connexion de données, transférer leur rapport, puis libérer leur connexion et se détacher du réseau. Il peut arriver que l'application souhaite que le device M2M lui communique un rapport de données non programmé. L'application réveille donc le device par exemple par SMS et ce dernier transmet les données au serveur M2M.
- Récolte de données (domaine de l'électronique) : L'acquisition d'informations se fait grâce aux capteurs embarqués dans les devices M2M. Les évolutions technologiques dans ce domaine donnent naissance à des dispositifs de moindre taille, moins coûteux et moins consommateurs en énergie.
- Transport des données (domaine des télécommunications) : Plusieurs technologies de réseaux, radio ou filaire, peuvent coexister dans une même solution M2M. Le choix technologique dépendra de la couverture requise, du mode de connectivité, de la quantité de données à transmettre, de sa fréquence et du modèle économique.
- Traitement des données (domaine de l'informatique) : L'application reçoit les données, les traite et intègre les données résultantes dans le système d'information de l'entreprise.