

Royaume du Maroc

Université Mohammed Premier

Ecole Nationale des Sciences Appliquées, Oujda

**MEMOIRE DE PROJET DE FIN D’ETUDE**

Pour l’obtention du diplôme

**D’INGENIEUR D’ETAT**

**Spécialité :** ***Génie Electrique***

**Par**

**EL HIMRI HOUSSAM & KHEDROUF Omar**

Sous le thème

**Etude de performance de l’utilisation de FPGA dans les unités de contrôles électroniques ECU**

*Encadré par :*

Pr. EL MEHDI Abdelmalek

Mr. BABAOUI Mohamed

*Devant le jury composé de:*

*Pr.* XXXXXXXXXXXXXX

*Pr.*XXXXXXXXXXXXXX

*Pr.* XXXXXXXXXXXXX

*Soutenu le XX Juin 2022*

**Année universitaire 2021/2022**

***2021 / 2022***



# Dédicace

*Après avoir rendu grâce à DIEU le tout puissant, d’avoir fait de moi ce que je suis aujourd’hui.*

*Je dédie ce travail*

*À ma très chère et douce mère, celle qui m'a donné la vie, qui m’a transmis l’amour, le symbole de tendresse. Sa prière et sa bénédiction m’ont été d’un grand secours pour mener à mes études. Puisse Dieu, vous combler de santé, de bonheur et vous procurer une longue vie.*

*À mon père, homme de valeurs et de principes dans la vie et synonyme d’amour et de sacrifice. Nul sentiment ne peut exprimer ma reconnaissance envers lui.*

*À mon cher frère Sami et ma chère sœur Samah,**vos affections et vos encouragements ont toujours été pour moi les plus précieux. Je prie Dieu de vous procurer santé, bonheur et longue vie.*

*À mes enseignants et mes encadrants, vos efforts, vos précieux conseils, et l’attention avec lesquelles vous m’avez toujours entouré m’ont beaucoup touché. Je vous serais reconnaissant toute ma vie, et j’espère ne jamais vous décevoir.*

*À tous mes amis, je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes des amis sur qui je peux compter.*

# Remerciement

C’est avec le plus grand honneur que je réserve cette page de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l’élaboration de ce travail.

Mes sincères remerciements sont destinés d’abord à **Mme. EL ALAMI Nora**, team leader, pour m’avoir accueilli comme stagiaire au sein de son équipe, pour ses directives, ses encouragements et surtout pour sa disponibilité tout au long de la période du stage.

J'adresse, également, ma profonde gratitude à **Mr. BABAOUI Mohamed** pour le soutien technique et moral, ainsi d’avoir mis à ma disposition, son temps et son riche savoir-faire tout au long de la période du stage. Son implication, sa créativité et sa clairvoyance m’inspirent et m’infligent de donner le meilleur de moi-même.

Mes sincères remerciements et reconnaissances sont également exprimés à **Mr. ELMEHDI Abdelmalek,** mon encadrant pédagogique, pour son temps consacré au suivi des différentes étapes de mon projet de fin d’études. Je le remercie pour ses directives, ses conseils et ses encouragements. J’espère avoir été digne de la confiance qu’il m’ait accordée et que ce travail est finalement à la hauteur de ses espérances

Je profite de ces quelques lignes pour présenter également mes profonds remerciements et mon estime au corps professoral de **l’ENSA OUJDA** et aux enseignants du département de la **filière Génie Electrique**, plus particulièrement notre chef de filière **Mr. EL HAFYANI Mohamed Larbi**, qui ont ménagé tous leur temps et efforts pour nous doter d’une si riche formation.

J’exprime tous mes remerciements aux honorables membres de jury pour l’intérêt qu’ils portent à mon travail et pour l’honneur qu’ils font en acceptant de juger mon travail.

Que tous ceux et celles qui ont contribué de près ou de loin à l’accomplissement de ce travail trouvent l’expression de mes remerciements les plus chaleureux.

# Résumé

L’étude présentée dans ce rapport s’inscrit dans le cadre de Projet de Fin d’Etudes effectué au sein de Capgemini Engineering, afin d’obtenir le titre d’ingénieur d’état en Génie Electrique.

Les véhicules modernes deviennent de plus en plus confortables en raison des différentes technologies, au cœur de ces technologies, l’ADAS est la plus populaire.

ADAS ou Systèmes d’Aide à la Conduite, sont des systèmes destinés à aider le conducteur dans le processus de conduite. Lorsqu’ils sont conçus avec une interface homme-machine sûre, ils devraient augmenter la sécurité de la voiture et plus généralement la sécurité routière ou aussi aider le conducteur pendant le stationnement et d’autres options. La plupart des accidents de la route ont eu lieu en raison de l’erreur humaine, les systèmes ADAS sont devenus une nécessité pour éviter les accidents et donner au conducteur une expérience confortable en conduisant.

La mise en œuvre de ce travail a suivi plusieurs phases dont les plus importantes sont l’étude faite sur certaines fonctions ADAS (BSD, LKA, ACC), suivie de leurs modélisations, le test et le développement d’une IHM du modèle en question, et une étude sur la communication entre le modèle et l’IHM, ceci en suivant une méthodologie adoptée au préalable et qui est détaillée ainsi que le déroulement du travail dans ce rapport.

Le but de ce projet est de redévelopper ces fonctions par l’outil FPGA dans le but d’améliorer les performances des calculateurs ECUs du véhicule, et bien sûr mettre le point sur l’efficacité de cet outil.

**Mot clés :** ADAS, FPGA, ECU, test, IHM

# Abstract

The study presented in this report is part of a final year project carried out within Capgemini engineering, in order to obtain the title of state engineer in "Electrical and Embedded Systems Engineering".

Modern vehicles are becoming more and more comfortable due to different technologies, at the heart of which the ADAS is the most popular.

ADAS or Driving Assistance Systems, are systems designed to assist the driver in the driving process. When designed with a safe human-machine interface, they should increase the safety of the car and more generally road safety or also help the driver during parking and other options. Since most road accidents have occurred due to human error, ADAS systems have become a necessity to avoid accidents and to give the driver a comfortable driving experience.

The implementation of this work has known several phases of which the most important are the study on a certain ADAS functions (BSD, LKA, ACC), followed by its modeling and the simulation of the model, test and development of a HMI of the model in question, and a study on the communication between the model and the HMI, this following a methodology adopted beforehand and which is detailed as well as the course of the work in this report.

The purpose of this project is to redevelop these functions by the FPGA tool in order to improve the performance of the vehicle's ECUs, and of course to review the effectiveness of this tool.

**Keywords:** ADAS, FPGA, ECU, test.

# Table des matières

[Dédicace 3](#_Toc105425267)

[Remerciement 4](#_Toc105425268)

[Résumé 5](#_Toc105425269)

[Abstract 6](#_Toc105425270)

[Table des matières 7](#_Toc105425271)

[Liste des figures 12](#_Toc105425272)

[Liste des tableaux 14](#_Toc105425273)

[Liste des abréviations 15](#_Toc105425274)

[Introduction générale 17](#_Toc105425275)

[Chapitre I : Présentation de l’organisme d’accueil et du contexte général du projet 18](#_Toc105425276)

[I. Introduction : 19](#_Toc105425277)

[II. Présentation de l’organisme d’Accueil 19](#_Toc105425278)

[1. Groupe CAPGEMINI ENGINEERING: 19](#_Toc105425279)

[1.1. Aperçu général : 19](#_Toc105425280)

[1.2. Historique : 19](#_Toc105425281)

[1.3. Secteurs d’activités : 20](#_Toc105425282)

[1.4. Implantation dans le monde : 22](#_Toc105425283)

[2. Capgemini Engineering Maroc 22](#_Toc105425284)

[2.1. Présentation générale : 22](#_Toc105425287)

[2.2. Organigramme Capgemini Engineering : 23](#_Toc105425288)

[a. Organigramme du Département Système: 24](#_Toc105425293)

[b. Centre des services du pôle modélisation : 24](#_Toc105425294)

[3. Clients et partenaires : 25](#_Toc105425295)

[III. Présentation du projet : 26](#_Toc105425296)

[1. Contexte général et problématique : 26](#_Toc105425297)

[2. Cahier des charges : 27](#_Toc105425298)

[3. Méthodologie de travail : 28](#_Toc105425299)

[4. Planification du projet : 29](#_Toc105425300)

[IV. Conclusion : 31](#_Toc105425301)

[Chapitre II : Etat de l’art : Les systèmes embarqués dans les systèmes ADAS 32](#_Toc105425302)

[I. Introduction : 33](#_Toc105425303)

[II. Architecture générale du véhicule : 33](#_Toc105425304)

[1. Systèmes embarqués 33](#_Toc105425305)

[1.1. Définition : 33](#_Toc105425306)

[1.2. Composition : 34](#_Toc105425307)

[1.3. Communication – multiplexage : 36](#_Toc105425308)

[1.4. Contexte : 37](#_Toc105425309)

[2. Calculateur ECU : 38](#_Toc105425310)

[2.1. Définition : 38](#_Toc105425313)

[2.2. Architecture ECU : 38](#_Toc105425314)

[III. Systèmes d’Aide à la Conduite (ADAS) : 39](#_Toc105425315)

[1. Définition : 39](#_Toc105425316)

[2. Niveaux d’autonomie : 40](#_Toc105425317)

[3. Classification des systèmes ADAS : 42](#_Toc105425318)

[4. Exemples des fonctions ADAS : 42](#_Toc105425320)

[4.1. Assistant de maintien de voie LKA (Lane Keeping Assist): 42](#_Toc105425322)

[4.2. Détection des angles morts BSD (Blind Spot Detection): 44](#_Toc105425323)

[4.3. Adaptive Cruise Control ACC (Adaptive Cruise Control) : 45](#_Toc105425324)

[4.4. Freinage d'urgence automatique AEB (Automatic Emergency Breaking) : 46](#_Toc105425325)

[4.5. Assistance aux manœuvres de stationnement (Park Assist) : 47](#_Toc105425326)

[5. Métier de conception fonctionnelle ADAS 48](#_Toc105425327)

[IV. Conclusion : 48](#_Toc105425328)

[Chapitre III : Conception et rédaction des spécifications 49](#_Toc105425329)

[I. Introduction : 50](#_Toc105425330)

[II. Environnement logiciel : 50](#_Toc105425331)

[1. Outils de modélisation et simulation des fonctions ADAS : 50](#_Toc105425332)

[1.1. MATLAB : 50](#_Toc105425333)

[1.2. Simulink : 51](#_Toc105425334)

[1.3. Carte FPGA : 52](#_Toc105425335)

[1.4. Langage HDL : 53](#_Toc105425336)

[2. Cycle en V 54](#_Toc105425337)

[2.1. Définition : 54](#_Toc105425338)

[2.2. Phases du cycle en V : 54](#_Toc105425339)

[3. Conception basée sur modèle (Model-Based Design) : 56](#_Toc105425340)

[3.1. Définition : 56](#_Toc105425341)

[3.2. Tests In the Loop : 58](#_Toc105425342)

[3.3. Analogie avec le projet : 58](#_Toc105425343)

[a. Fonction BSD : 58](#_Toc105425344)

[4. Architecture générale : 61](#_Toc105425345)

[III. Acquisition des informations : 62](#_Toc105425346)

[1. Etude sur les capteurs : 62](#_Toc105425347)

[1.1. Capteur Ultrason : 62](#_Toc105425348)

[1.2. Radar : 63](#_Toc105425349)

[IV. Diagrammes : 64](#_Toc105425350)

[1. Identification des besoins : 64](#_Toc105425351)

[1.1. Diagramme bête à cornes : 64](#_Toc105425352)

[1.2. Diagramme de pieuvre : 65](#_Toc105425353)

[1.3. Diagramme de contexte : 66](#_Toc105425354)

# Liste des figures

[Figure 1: Domaines d’activités du groupe Capgemini Engineering 22](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425355)

[Figure 2: Capgemini Engineering Maroc à Casanearshore 23](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425356)

[Figure 3: Organigramme du département Système 24](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425357)

[Figure 4 : Marques automobiles de PSA 26](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425358)

[Figure 5 : Etapes de la démarche PDCA 28](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425359)

[Figure 6 : Diagramme de Gantt 30](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425360)

[Figure 7: Calculateur du véhicule 34](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425361)

[Figure 8: Communication capteur – calculateur – actionneur 36](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425362)

[Figure 9 : Les systèmes embarqués dans l'automobile 37](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425363)

[Figure 10: Calculateur ECU 38](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425364)

[Figure 11 : Architecture typique d’une ECU 39](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425365)

[Figure 12 : Les niveaux d'autonomie des ADAS 41](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425366)

[Figure 13: Classification des ADAS 42](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425367)

[Figure 14 : Fonction d'assistance de maintien de voie (LKA) 43](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425368)

[Figure 15: La fonction détection des angles morts (BSD) 44](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425369)

[Figure 16 : La fonction régulation de vitesse (ACC) 45](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425370)

[Figure 17: La fonction de freinage automatique d'urgence (AEB) 46](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425371)

[Figure 18 : La fonction d'assistance de stationnement (Park Assist) 47](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425372)

[Figure 19 : Logo de MATLAB 51](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425373)

[Figure 20: Carte FPGA 52](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425374)

[Figure 21: Langage de description matériel VHDL 53](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425375)

[Figure 22: Cycle en V 54](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425376)

[Figure 23: Model Based Design 57](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425377)

[Figure 24: Modélisation de la fonction BSD 59](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425378)

[Figure 25: Code VHDL de la fonction BSD 59](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425379)

[Figure 26: Test Bench fonction BSD 60](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425380)

[Figure 27 : Implémentation du code dans la carte FPGA (pas d’obstacle) 60](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425381)

[Figure 28: Implémentation du code dans la carte FPGA (avec obstacle) 61](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425382)

[Figure 30: Entrées / Sorties des différentes fonctions 61](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425383)

[Figure 31: Principe du capteur ultrason 62](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425384)

[Figure 32: Effet Doppler 64](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425385)

[Figure 33: Diagramme bête à cornes 65](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425386)

[Figure 34: Diagramme de pieuvre 66](#_Toc105425387)

[Figure 35: Diagramme de contexte 67](file:///C:\Users\YAHYA\Desktop\Rapport%20PFE.docx#_Toc105425388)

# Liste des tableaux

[Tableau 1: Historique du groupe Altran 20](#_Toc105594102)

[Tableau 2 : Liste des tâches et leurs durée d'éxécution 31](#_Toc105594103)

[Tableau 3: Exemple de capteurs 35](#_Toc105594104)

[Tableau 4: Exemple d'actionneurs 36](#_Toc105594105)

# Liste des abréviations

**ADAS:** Advanced Driver Assistance System

**FPGA:** Field Programmable Gate Array

**ECU:** Electronic Control Unit

**IHM:** Interface Homme Machine

**BSD:** Blind Spot Detection

**LKA:** Lane Keeping Assist

**ACC:** Adaptive Cruise Control

**PLM:** Product Lifecycle Management

**DSP:** Digital Signal Processor

**CAN:** Controller Area Network

**LDW:** Lane Departure Warning

**AEB:** Automatic Emergency Breaking

**HDL**: Hardware Description Language

**MBD:** Model-Based Design

**MIL:** Model In Loop

**HIL:** Hardware In Loop

**SIL:** Software In Loop

# Introduction générale

Un système embarqué est défini comme un système [électronique](https://fr.wikipedia.org/wiki/%C3%89lectronique_(technique)) et [informatique](https://fr.wikipedia.org/wiki/Informatique) autonome, souvent [temps réel,](https://fr.wikipedia.org/wiki/Syst%C3%A8me_temps_r%C3%A9el) spécialisé dans une tâche précise. Le terme désigne aussi bien le [matériel informatique](https://fr.wikipedia.org/wiki/Mat%C3%A9riel_informatique) que le [logiciel](https://fr.wikipedia.org/wiki/Logiciel) utilisé.

L’automobile est un des secteurs qui utilise énormément de technologies embarquées. Aujourd’hui, les véhicules sont devenus des systèmes plus complexes dont les fonctions les plus importantes sont contrôlées par l’ordinateur. Parmi les systèmes les plus élaborés, nous trouvons les systèmes ADAS. Ces systèmes sont conçus pour être intégrés dans les véhicules afin de renforcer la quantité et la qualité des informations transmises au conducteur.

Capgemini Engineering, étant un des leaders mondiaux du conseil en innovation et ingénierie avancée, s’est inscrite dans une démarche d’amélioration continue de l’ensemble de ses secteurs et de ses activités y compris l’automobile.

C’est dans ce contexte que s’inscrit donc ce projet de fin d’études qui consiste à faire une étude des performances d’utilisation des FPGA dans les unités de contrôle électroniques ECUs, par développer certaines fonctions ADAS par l’outil FPGA et conclure l’efficacité de ce dernier par rapport aux autres outils.

Le présent manuscrit est organisé en trois chapitres dont le premier a pour but de présenter l’organisme d’accueil ainsi que le contexte général du projet, le deuxième chapitre se concentre sur l’état de l’art de tout ce qui concerne les systèmes embarqués et les systèmes ADAS, pendant que le troisième s’occupe des différents outils et spécifications utilisés, alors que le dernier sera consacré pour les résultats obtenus.

# Chapitre I : Présentation de l’organisme d’accueil et du contexte général du projet

## Introduction :

La première partie de ce présent chapitre est dédiée à la découverte de l’environnement dans lequel s’est déroulé ce projet de fin d’études, elle sert principalement à définir de plus en détails la présentation générale de l’organisme d’accueil « Capgemini Engineering ». Dans la deuxième partie nous allons présenter le contexte du projet, à ce stade, nous annoncerons dans un premier temps la problématique suivie de la démarche utilisée pour l’aborder, et clôturer par une planification, qui permettra d’atteindre nos objectifs tout en respectant les délais.

## Présentation de l’organisme d’Accueil

### Groupe CAPGEMINI ENGINEERING:

#### Aperçu général :

Le groupe Capgemini Engineering Maroc est un leader mondial du conseil en innovation et ingénierie avancée, Capgemini Engineering propose à ses clients d’innover autrement en les aidant à développer ou en développant pour eux les produits et les services de demain. La Groupe les accompagne sur l’intégralité de la chaine de valeur du cycle de vie d’un projet, de l’idée à l’industrialisation. Capgemini Engineering intervient depuis plus de 30ans auprès des grands acteurs de nombreux secteurs : automobile, aéronautique, spatial, défense et naval, ferroviaire, énergie, industriel et électronique, sciences de la vie, télécoms et médias, logiciel, finance et secteur public.

#### Historique :

Capgemini Engineering est le leader mondial incontesté des services d’ingénierie et de R&D, a été créé en 1982 et dirigé par Dominique Cerutti. Le groupe offre à ses clients une proposition de valeurs uniques pour relever leurs défis de transformation et d’innovation. Capgemini Engineering accompagne ses clients, du concept à l’industrialisation, pour développer les produits et les services de demain. Capgemini Engineering a généré 2,9 milliards d’euros de chiffre d’affaires en 2018, avec près de 47 000 employés dans plus de 30 pays.

Tableau 1: Historique du groupe Altran

|  |  |
| --- | --- |
| Année | Evénements |
| 1982 | Création d’Altran Technologies S.A en France |
| 2004 | Altran s’implante en Asie et crée Altran Pr[i]me, une entité spécialisée dans le management de projets d’innovation de large envergure. |
| 2012 | Dans le cadre du Plan de Performance 2012, PSA Peugeot Citroën choisit Altran comme partenaire stratégique |
| 2013 | Création d’Altran au Maroc |
| 2018 | Création d’un centre d’ingénierie Automobile à Casablanca une nouvelle joint-venture entre Altran et Magna entreprise canadienne spécialisée dans l’équipement automobile et la sous traitance sous le nom de MG2. Altran annonce le plan "The High Road, Altran 2022". Ce plan vise 14,5% de marge et une augmentation du chiffre d'affaires à 4 milliards d'euros en 2022, en pariant sur les ruptures technologiques. |
| 2020 | En avril 2020, Altran est devenue filiale de Capgemini et est renommée Capgemini Engineering |

#### Secteurs d’activités :

Capgemini Engineering intervient depuis plus de 35 ans auprès des grands acteurs de nombreux secteurs : Automobile, Aéronautique, Spatial, Défense & Naval, Rail, Infrastructure & Transport, Industrie & Biens de consommation, Sciences de la vie, Communications, Semi-conducteur & Electronique, Logiciel & Internet, Finance & Secteur public.

Le groupe Altran couvre toutes les phases du cycle de vie d’un projet, de sa définition (veille technologique, études de faisabilité technique, définition des stratégies, etc.) à sa concrétisation (conception, mise en œuvre et validation de solutions, etc.).L’activité s’organise autour de trois domaines :

• Conseil en technologie, et en R&D : est la capacité de mettre en œuvre le savoir-faire, les méthodes techniques et scientifiques des consultants ingénieurs pour aboutir à des projets innovants.

• Conseil en organisation et en systèmes d’information : permet de fournir aux entreprises la capacité de rester compétitives face aux contraintes de croissance du marché, de rentabilité et de législation. Les entreprises qui font appel à Capgemini Engineering sur ce point cherchent à faciliter leurs prises de décisions, et leur agilité organisationnelle.

• Conseil en stratégie et management : permet aux entreprises de maîtriser leur environnement et d'anticiper les évolutions du secteur dans lequel elles travaillent.



Figure 1: Domaines d’activités du groupe Capgemini Engineering

#### Implantation dans le monde :

L’empreinte internationale d’Altran offre un avantage unique sur le marché actuel. En phase avec une planète de plus en plus connectée. Le Groupe est conçu pour offrir des services d’innovation et d’ingénierie de premier ordre à ses clients, tout en leur permettant de bénéficier des meilleures pratiques mondiales dans tous les secteurs d’activités. Le groupe Altran est implanté dans 30 pays dans le monde, répartis entre l’Europe, l’Asie, l’Amérique, et l’Afrique (Maroc et Tunisie).

### Capgemini Engineering Maroc



#### Présentation générale :

À travers son implantation à Casablanca en avril 2013, Capgemini Engineering a souhaité disposer d’une plateforme Near-shore afin d’accompagner le développement international du groupe dans les secteurs de l’automobile, de l’aéronautique et du transport. Il s’agit en effet d’accompagner les clients de Capgemini Engineering dans leur stratégie d’innovation, d’optimisation de coût et des délais.



Figure 2: Capgemini Engineering Maroc à Casanearshore

L’entité marocaine a également pour ambition d’être un acteur de proximité au service des grands comptes clients de Capgemini Engineering installés sur le territoire national. Capgemini Engineering Maroc s’intéresse notamment à l’évolution des secteurs de l’automobile, de l’aéronautique et des énergies renouvelables.

Enfin, Capgemini Engineering s’appuie sur la stratégie offshoring mise en place par le gouvernement marocain offrant des avantages optimisant fortement la composante compétence/coût.

#### Organigramme Capgemini Engineering :

Dans le cadre de la nouvelle stratégie de Capgemini Engineering, les équipes opérationnelles sont reparties sur quatre directions, dont chacune propose des solutions adéquates aux besoins des clients :

▪ Direction Ressources Humaines (DRH)

▪ Direction Finance (DF)

▪ Direction Programme (DP)

▪ Direction Technique (DT)



##### Organigramme du Département Système:

Dans le cadre de la mission relative à ce projet de fin d’études, notre stage s’est déroulé au sein de la Direction Technique, Département Ingénierie Système, équipe Modélisation. Et nous avons eu l’occasion de travailler avec les différents pôles qui se collaborent pour la réalisation pour la réalisation du projet.

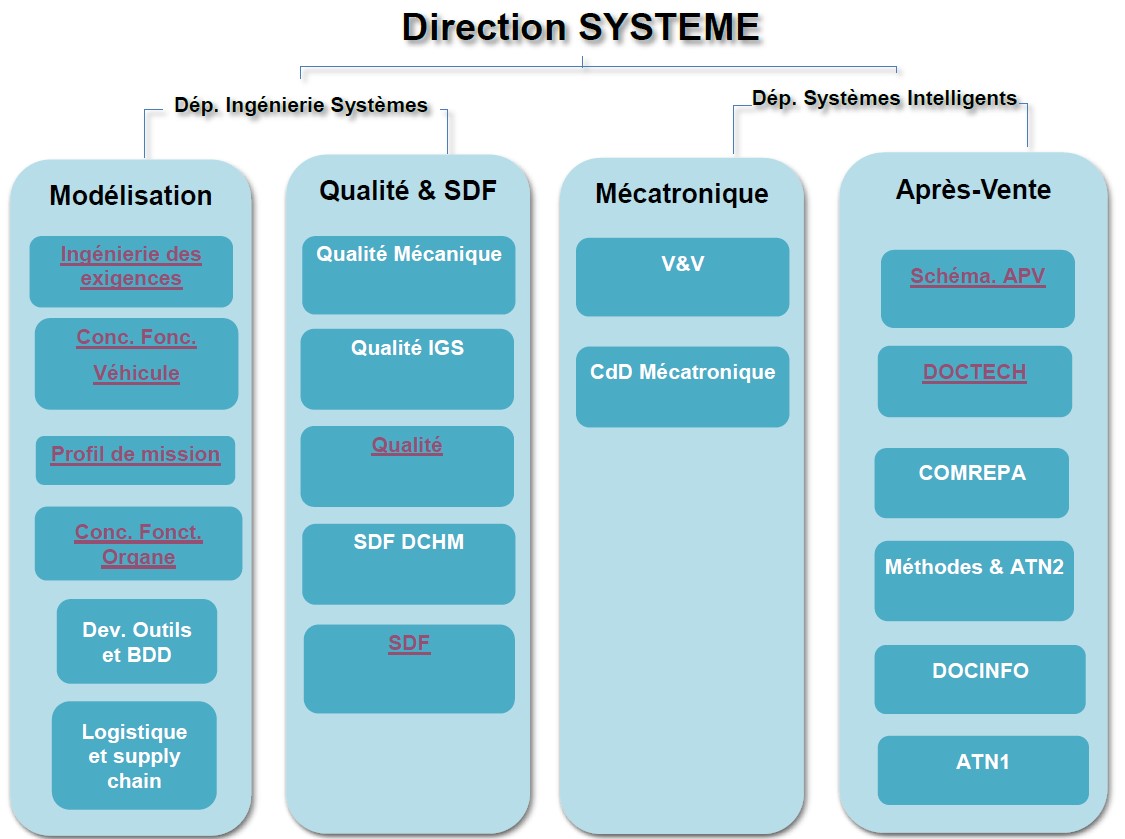


Figure 3: Organigramme du département Système

La direction Système intervient avant la phase de production, elle s’occupe de tout ce qui est R&D, solutions techniques… Elle se compose de quatre pôles, Modélisation, Qualité & SDF, Mécatronique et Après-Vente.

##### Centre des services du pôle modélisation :

Il se compose également de six centres de services :

* **Ingénierie des exigences** : Cette équipe s’occupe de la réception des exigences clients, le tri et la suppression des exigences non faisables, la reformulation et la traduction des exigences, ainsi que, la migration des exigences vers la plateforme PLM (Product Lifecycle Management).
* **Conception Fonctionnelle du Véhicule** : Le travail de cette équipe est basé sur l’analyse fonctionnelle des fonctions ADAS (Advanced Driver Assistance Systems) se sont des fonctions d’aide à la conduite. Lors de cette conception fonctionnelle, chaque fonction du véhicule doit être recensée, caractérisée, ordonnée, hiérarchisée et valorisée afin d’effectuer un dimensionnement correct des caractéristiques des fonctions du véhicule.
* **Profil de mission** : Elle s’occupe de la représentation statistique d’un scalaire distribué sur une population donnée, il sert à caractériser l’usage des véhicules, de ses fonctions et de ses composants en fonction des zones de commercialisation et des environnements. Cette étape est nécessaire pour une conception optimisée.
* **Conception Fonctionnelle Organe** : Même principe que la conception fonctionnelle du véhicule, sauf que cette équipe s’occupe de la modélisation organique et non pas la modélisation fonctionnelle.
* **Dev outil et BDD** : Ce centre de service assure la production des tâches de la conversion automatique de document entrant (spécifications comme exemple) en plan de test (ou bibliothèque de tests), ainsi que, la réalisation de banc de validation HIL (Hardware in the loop) qui se présente comme une approche qui consiste à tester un produit final dans un environnement aussi proche que possible de l’environnement réel. Le produit à tester est ainsi placé dans une boucle matérielle liant les commandes aux contrôles et simulant les liens et les interactions qui existent physiquement.
* **Logistique et supply chain** : Le rôle des fonctions achat-logistique est d’assurer la continuité et la fiabilité des flux de marchandises, depuis l’entrepôt de fournisseur, jusqu’à la réception par le demandeur.

### Clients et partenaires :

Le client majeur, auquel la société a rendu service durant notre période de PFE au sein de Capgemini Engineering Maroc est PSA Peugeot Citroën. PSA Peugeot Citroën est un des principaux constructeurs automobiles européens avec 11,8 % de parts de marché en Europe en 2014. Il construit son développement sur 3 marques, au rayonnement mondial et aux personnalités distinctes dans le cadre de stratégies internationales coordonnées et d’un plan produit cohérent. Le Groupe PSA est un constructeur automobile français qui exploite les marques automobiles Peugeot, Citroën, DS, OPEL et Vauxhall.

Il construit son développement sur 3 marques, au rayonnement mondial et aux personnalités distinctes dans le cadre de stratégies internationales coordonnées et d’un plan produit cohérent. Groupe PSA est un constructeur automobile français qui exploite les marques automobiles Peugeot, Citroën, DS, OPEL et Vauxhall.

Figure 4 : Marques automobiles de PSA

## Présentation du projet :

### Contexte général et problématique :

Récemment, dans la recherche automobile et dans le domaine des transports intelligents, plusieurs projets intéressants ont été menés afin de diminuer le nombre d’accidents, Lors du développement de ces projets, de nouveaux systèmes d’aide à la conduite ont été proposés.

Cela est dans le but d’assurer la sécurité des usagers de la route, pour améliorer continuellement l’expérience de conduite de leurs conducteurs, ces entreprises développent toujours plus d’équipements innovants, s'appuyant sur [les dernières technologies existantes](https://www.ornikar.com/code/cours/mecanique-vehicule/technologie-assistance).

Dans ce cadre et afin d’opérer avec excellence les projets et satisfaire les clients notamment le principal client PSA et garantir une bonne qualité, Capgemini Engineering a lancé des projets qui visent à atteindre l’excellence en termes de qualité, coût et délai.

C’est dans cette perspective que s’inscrit mon projet qui a pour objectif d’étudier certaines fonctions ADAS qui sont déjà développées et les reprogrammer par la carte FPGA, ce qui permet de viser une meilleure performance. Les résultats ou les bénéfices de cette étude seront visibles clairement dans certains critères selon la fonction concernée (fiabilité, temps de réponse, coût…).

### Cahier des charges :

**Sujet :** Etude de performance de l’utilisation de FPGA dans les unités de contrôles électroniques ECU (modélisation et simulation (Avec possibilité de réalisation).

Les unités de contrôle électronique (ECU) permettent de gérer en permanence les fonctions embarquées [l'assistance au freinage, la suspension active, les fonctionnalités de direction, les sacs gonflables, etc.] et ses contraintes en utilisant un microcontrôleur puissant exécutant du code surveillant un ensemble de capteurs et contrôlant une séquence d'actionneurs.

L’objectif du sujet est de comprendre l’architecture EE des véhicules afin de détecter les contraintes dans les fonctions critiques et penser à l’optimisation en implémentant FPGA en tant que cœur de traitement des ECUs au lieu des microcontrôleurs et faire une étude comparative en faisant la simulation et la réalisation d’un prototype montrant le résultat en cas de présence de matériel

Il s’articule autour du cahier des charges suivant :

* Etudier l’architecture EE du véhicule (précisément les ECUs).
* Se reposer sur le choix des fonctions ADAS qui seront étudiées.
* Etudier et développer chaque fonction à l’aide de l’outil MATLAB / SIMULINK.
* Etudier la carte FPGA et voir son impact dans le cas de son utilisation dans une ECU comparé à celui obtenu lors de l’utilisation d’un microcontrôleur.
* Modéliser et simuler les résultats (implémentation dans la carte)
* Effectuer une étude comparative entre les deux approches.
* Conclure sur l’efficacité des FPGA comme étant le cœur de traitement des ECUs.

### Méthodologie de travail :

Afin de réaliser ce projet, j’ai choisi d’adopter la démarche PDCA qui nous a permis de mener à bien les tâches demandées.

**PDCA :** une démarche d’amélioration continue ou de résolution de problème, symbolisée par la roue de Deming.

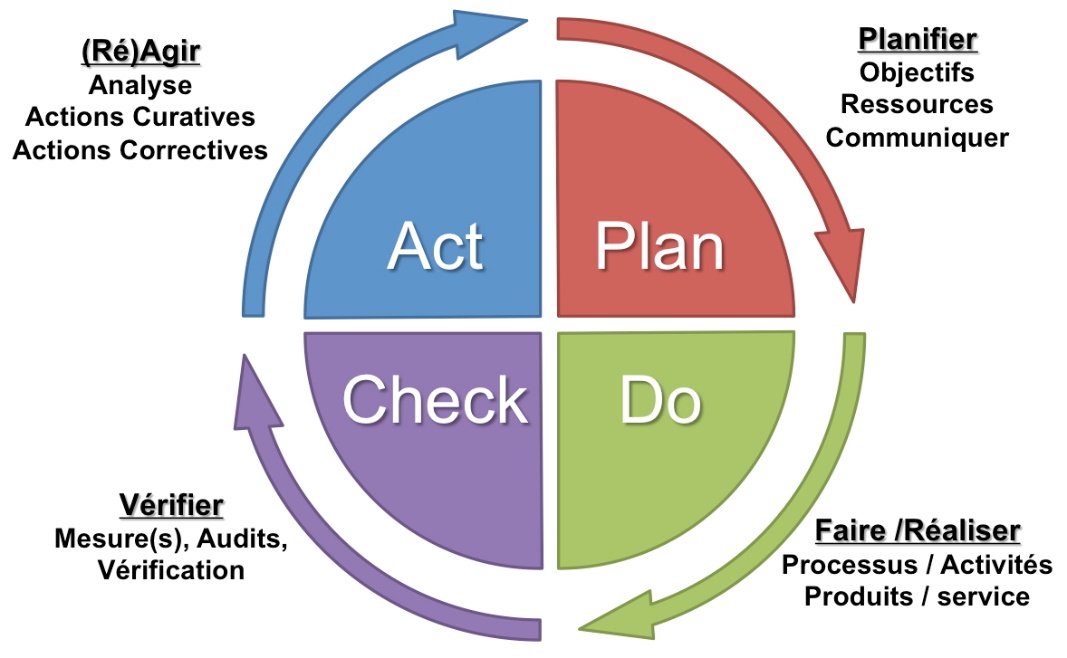


Figure 5 : Etapes de la démarche PDCA

Il signifie :

* P : Plan = planifier

Cette étape est très importante, car elle consiste à bien **définir le sujet ou le problème**, afin d’identifier des solutions pérennes.

Cette étape est finalisée par un plan d’actions, incluant leur planification et les acteurs.

* D : Do = réaliser

Cette étape consiste en la **mise en œuvre des actions** définies précédemment.

* C : Check = vérifier

Il s’agit de **vérifier l’efficacité des actions** menées. Ceci peut se faire par le biais de mesures, d’indicateurs, ou d’observations. Un délai peut être défini selon la nature de l’action.

Des ajustements doivent être réalisés, si nécessaire en revenant à l’étape P, lorsque des actions se révèlent inefficaces.

* A : Act = assurer et améliorer

Cette étape permet de finaliser la démarche afin d’**assurer la pérennité des résultats** des actions mises en œuvre. Il s’agit le plus souvent d’élaborer ou mettre à jour des documents, tels que procédures, processus, guides de bonnes pratiques, ou formulaires.

Il s’agit également d’identifier des **améliorations**, en revenant à l’étape P pour les mettre en œuvre.

 L’ISO 9001 est structuré suivant le cycle du PDCA.

### Planification du projet :

Après avoir défini la problématique et la démarche à suivre, il est nécessaire d’élaborer un planning détaillé des tâches à réaliser afin de respecter les délais prédéfinis pour le stage et assurer son bon déroulement. Le diagramme GANTT, dans la page suivante (*figure 6)*, présente l’ensemble des tâches à effectuer pour la réalisation du projet avec une prévision de la durée de chacune.

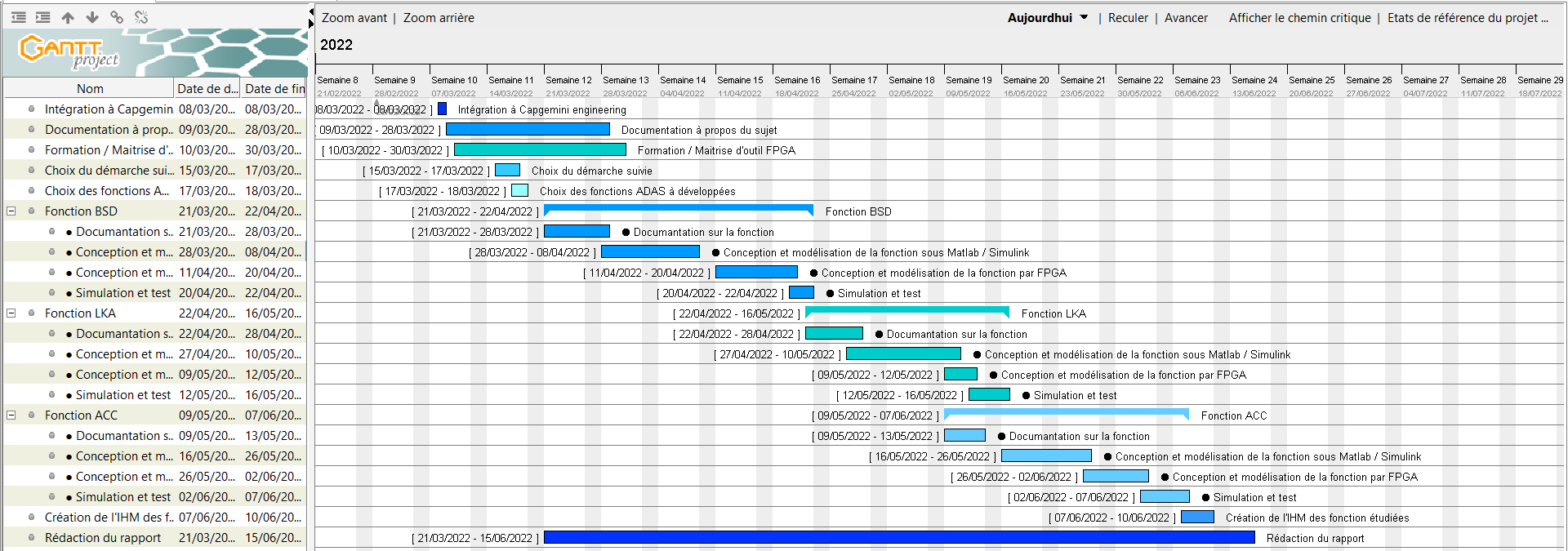


Figure 6 : Diagramme de Gantt

|  |  |  |
| --- | --- | --- |
| **Tâche** | **Durée de début** | **Durée de fin** |
| **● Intégration à Capgemini engineering** | mar 08-mars-22 | mar 08-mars-22 |
| Documentation à propos du sujet | mer 09-mars-22 | lun 28-mars-22 |
| Formation / Maitrise d'outil FPGA | jeu 10-mars-22 | mer 30-mars-22 |
| Choix de la démarche suivie | mar 15-mars-22 | jeu 17-mars-22 |
| Choix des fonctions ADAS à développées | jeu 17-mars-22 | ven 18-mars-22 |
| **● Fonction BSD** | lun 21-mars-22 | lun 28-mars-22 |
| Documentation sur la fonction |
| Conception et modélisation de la fonction sous Matlab / Simulink | lun 28-mars-22 | ven 08-avr-22 |
| Conception et modélisation de la fonction par FPGA | lun 11-avr-22 | mer 20-avr-22 |
| Simulation et test | mer 20-avr-22 | ven 22-avr-22 |
| **● Fonction LKA** | ven 22-avr-22 | jeu 28-avr-22 |
| Documentation sur la fonction |
| Conception et modélisation de la fonction sous Matlab / Simulink | mer 27-avr-22 | mar 10-mai-22 |
| Conception et modélisation de la fonction par FPGA | lun 09-mai-22 | jeu 12-mai-22 |
| Simulation et test | jeu 12-mai-22 | lun 16-mai-22 |
| **● Fonction ACC** | lun 09-mai-22 | ven 13-mai-22 |
| Documentation sur la fonction |
| Conception et modélisation de la fonction sous Matlab / Simulink | lun 16-mai-22 | jeu 26-mai-22 |
| Conception et modélisation de la fonction par FPGA | jeu 26-mai-22 | jeu 02-juin-22 |
| Simulation et test | jeu 02-juin-22 | mar 07-juin-22 |
| Création de l'IHM des fonctions étudiées | mar 07-juin-22 | ven 10-juin-22 |
| **● Rédaction du rapport** | lun 21-mars-22 | mer 15-juin-22 |

Tableau 2 : Liste des tâches et leurs durée d'éxécution

## Conclusion :

Ce chapitre introductif a été consacré essentiellement à la présentation de l’environnement dans lequel mon projet de fin d’études a été effectué. Il a aussi mis l’accent sur la présentation du contexte général de mon projet, ses objectifs et sa planification.

Pour entamer notre projet, une étude technique sur des différentes fonctions ADAS et ses principes sera présentée en détail dans le prochain chapitre.

# Chapitre II : Etat de l’art : Les systèmes embarqués dans les systèmes ADAS

## Introduction :

Ce chapitre met l'accent sur les différentes notions rencontrées lors du projet commençons par définir les systèmes ADAS, et citer les différents types et fonctions des ADAS, ensuite le principal outil utilisé qui est la carte FPGA.

La deuxième partie explique en détails la relation entre les FPGA et le domaine automobile précisément les systèmes ADAS, ce qui représente la base de notre projet.

## Architecture générale du véhicule :

### Systèmes embarqués

#### Définition :

Un système embarqué est un ensemble d’éléments informatiques et électroniques interagissant entre eux de façon autonome et complémentaire. Ces systèmes sont conçus de manière à pouvoir répondre spécifiquement aux besoins de leur environnement respectif.

Le terme « système » désigne l’ensemble des éléments qui constituent le système embarqué, souvent ces systèmes sont composés de sous-systèmes étant donné leur complexité.

Le terme « embarqué » représente la mobilité et l’autonomie du système en interaction directe avec son environnement dans l’exécution de tâches précises, afin de répondre à la finalité de celui-ci.

Contrairement aux systèmes classiques, les systèmes embarqués sont conçus pour réaliser des tâches bien précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de sécurité, indispensables selon l’utilisation du système.

Un système embarqué regroupe à la fois la partie software (logicielle) et la partie hardware (matériaux) étroitement liées afin de produire les résultats escomptés.

#### Composition :

* **Calculateur :** un boitier métallique considéré comme étant l’élément principal du système embarqué automobile où régit la mémoire, la carte-mère ou encore le traitement logiciel. Chacun des calculateurs automobiles sont dédiés au pilotage d’une ou certaines tâches bien précises, ainsi de nombreux calculateurs sont présents dans les véhicules formant son système électronique.

Figure 7: Calculateur du véhicule

* **Capteur :** un élément essentiel qui permet à transformer une grandeur physique (température, pression…) en un signal électrique afin de le transmettre au calculateur, son principal objectif est donc de renseigner le calculateur qui va pouvoir agir en temps réel avec l’environnement, c’est pourquoi, il envoie de façon constante des informations en continu au calculateur relié.

|  |  |
| --- | --- |
| **Information de grandeur physique** | **Signaux électriques** |
| Température | Sonde de température |
| Position / vitesse | Capteur PMH |
| Débit d'air | Débitmètre |
| Pression | Capteur pression |
| Vibrations | Capteur cliquetis |
| Angle | Capteur gyroscopique |
| Taux d'oxygène | Sonde lambda |
| Taux d'humidité | Capteur de pluie |
| Lumière | Capteur de luminosité |

Tableau 3: Exemple de capteurs

* **Actionneur :** Lorsque le traitement est réalisé par le calculateur, un signal électrique est transmis aux actionneurs permettant une action physique sur le véhicule.

Ces actionneurs (ou actuateurs) transforment le signal électrique reçu en énergie mécanique. Cette transformation d’énergie peut être réalisée par moteur, de façon magnétique, hydraulique ou optique.

|  |  |
| --- | --- |
| **Paramètre** | **Actionneur** |
| Temps d'injection | Injecteur |
| Etincelle de combustion | Bobine d'allumage |
| Suppression Nox (oxyde d'azote) | Vanne EGR |
| Pression turbo | Electrovanne turbo |
| Production de froid | Compresseur climatisation |
| Verrouillage des portières | Actionneur de porte |
| Mouvement des vitres | Moteur lève-vitre |

Tableau : Exemple d'actionneurs

#### Communication – multiplexage :

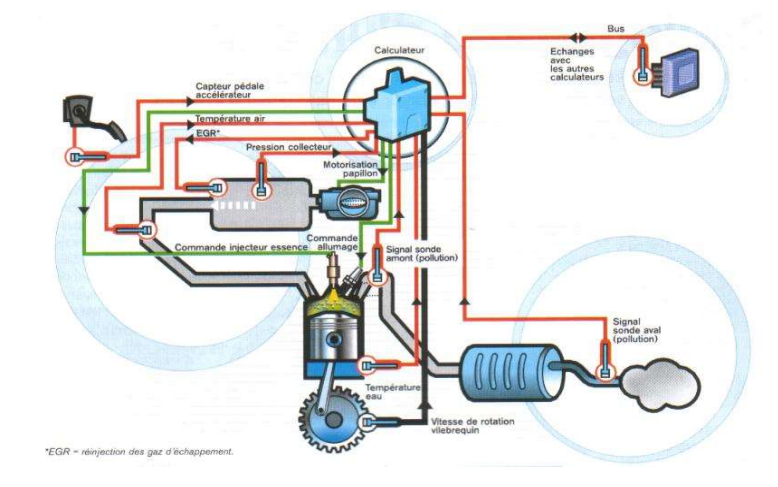
Tous ces composants échangent entre eux par l’intermédiaire de faisceaux, ils correspondent à de petits câbles permettant la transmission des signaux électriques contenant les informations recueillies et à transmettre.

Figure 8: Communication capteur – calculateur – actionneur

Cette figure illustre la communication et les échanges entre le calculateur et ses actionneurs et capteurs. En vert, les interactions avec les actionneurs comme la commande d’allumage et en rouge, les interactions avec les capteurs, correspondant tous à des faisceaux électriques.

#### Contexte :

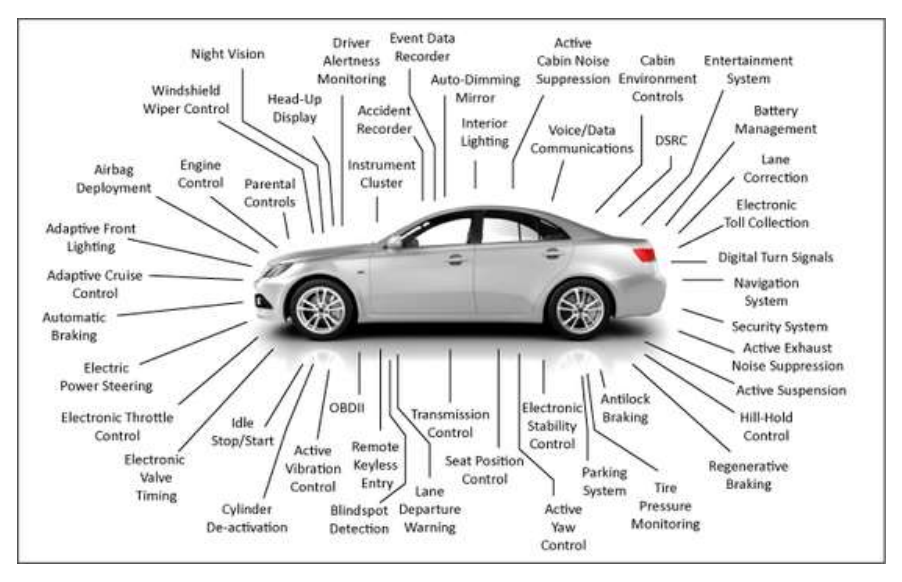
Aujourd’hui, un véhicule contient une grande quantité d’électronique et d’informatique : on retrouve plus de 100 capteurs, 30 à 50 calculateurs selon le type de véhicule et parfois près d’un million de lignes de codes pour les véhicules de dernière génération. Cette évolution s’explique par les demandes exigeantes des consommateurs et l’envie de différenciation des concurrents sur marché de l’automobile. S’ajoute à cela les contraintes économiques et écologiques où l’électronique embarquée répond à ces nouvelles attentes. De nouvelles fonctionnalités impliquent parfois une intégration électronique et informatique par le biais de systèmes embarqués. Voici une représentation des systèmes intégrés d’un véhicule moderne.

Figure 9 : Les systèmes embarqués dans l'automobile

### Calculateur ECU :



#### Définition :

L'ECU (Electronic Control Unit ou unité de commande électronique) est le cerveau du bloc moteur et est donc indispensable dans la voiture d'aujourd'hui. Pendant la conduite, le dispositif de contrôle collecte constamment des informations pour pouvoir contrôler d'autres composants.

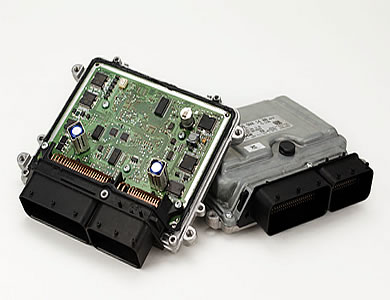


Figure 10: Calculateur ECU

#### Architecture ECU :

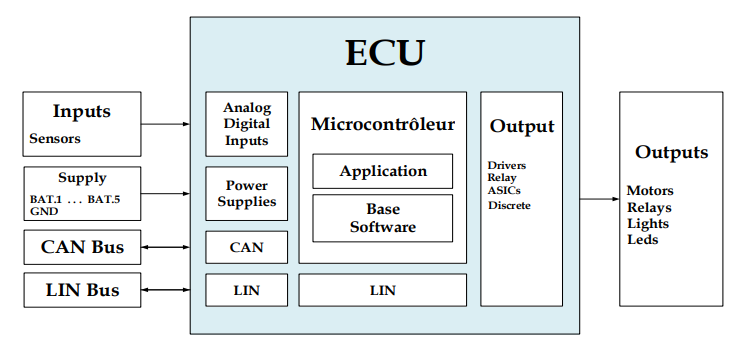
Selon le domaine auquel l'ECU est destinée, les fournisseurs proposent des architectures personnalisées qui conviennent le mieux aux fonctionnalités de ce domaine. Par exemple, un contrôleur de corps peut intégrer différents protocoles de réseau et offrir peu ou pas de prise en charge de l'accélération matérielle, tandis qu'un contrôleur de télématique intègre une interconnexion à grande vitesse et des blocs d'accélérateurs dédiés pour le traitement vidéo ou les interfaces radar. Typiquement, une ECU est composée d’un élément de traitement, d’une interface réseau et de la mémoire associée, ainsi que d’autres modules, comme illustré sur la figure 11. L'élément de traitement peut être un microcontrôleur (MCU), un processeur à usage général (GPP), un DSP (Digital Signal Processor) ou un contrôleur spécifique à une application. Celles-ci exécutent les tâches logicielles requises pour la fonctionnalité de l’ECU. L'élément de traitement intègre généralement des périphériques communs tels que des temporisateurs (Timers), des interfaces I2C/SPI pouvant être utilisés par le code d'application.

Figure 11 : Architecture typique d’une ECU

## Systèmes d’Aide à la Conduite (ADAS) :

### Définition :

Les systèmes d’aide à la conduite ADAS sont des systèmes électroniques présents dans les voitures ayant pour objectif d’automatiser, d’adapter, d’améliorer les conditions de sécurité et de conduite. Les fonctions de sécurité sont conçues pour éviter les collisions et les accidents, en proposant des technologies qui alerteront le conducteur de la présence de risques potentiels, ou qui lui permettront d’éviter des collisions par la mise en œuvre de systèmes de surveillance, et de prise de contrôle du véhicule.

Les systèmes d’assistance à la conduite comportent les fonctions de régulation de la vitesse ACC, de freinage automatique, et de détection de présence d’autres véhicules ou de dangers, le maintien du véhicule dans sa voie, la prise en compte des informations GPS sur la situation de la circulation, la connexion des téléphones et systèmes de communication, ou la visibilité dans les angles morts.

### Niveaux d’autonomie :

***Niveau 0 :*** *(***Pas d’automatisation)** Les systèmes automatiques peuvent envoyer des alertes et des avertissements, mais ils ne peuvent pas contrôler le véhicule. Le conducteur ne dispose logiquement d’aucune autonomie. Il a le contrôle total et exclusif des fonctions primaires du véhicule (les freins, la direction, l’accélération et la force motrice).

***Niveau 1 :*** (**Conduite Assistée)**  Le conducteur et les systèmes automatiques partagent le contrôle du véhicule.

Par exemple : Pendant l’activation de l’ACC (Régulateur de vitesse adaptatif) le conducteur contrôle le système de direction (Volant) en même temps la fonction ACC ajuste la vitesse, la responsabilité des manœuvres est en permanence conservée par l’humain, qui délègue une partie des tâches aux systèmes.

***Niveau 2 :*** (**Automatisation partielle)**  Les systèmes automatiques sont capables de contrôler complètement le véhicule (Accélération, Freinage et Volant). Le conducteur est responsable de surveiller la conduite et prêt à intervenir à tout moment. La fonction Park Assist est un bon exemple du Niveau 2 puisque la voiture gère seule tous les paramètres de guidage sous la supervision du conducteur, qui a la possibilité d’intervenir pour reprendre la main sur la trajectoire à tout instant. En cas d’accident, il est entièrement responsable du défaut du système car il n’a pas été suffisamment attentif à son environnement et à manquer à son devoir de supervision. Cette étape marque la limite juridique actuelle, pour laquelle il n’est pas encore possible de transférer la responsabilité à une machine même si sa fiabilité est supérieure aux décisions humaines.

***Niveau 3 :* (Automatisation conditionnelle)** Le véhicule est capable de réagir automatiquement aux situations qui exigent une action immédiate (Comme le freinage d’urgence). Le conducteur doit être prêt parfois à intervenir dans un bref délai.

***Niveau 4 :*** (**Automatisation de haut niveau)**  Semblable au niveau précédent, sauf que ce niveau ne demande pas l’attention du conducteur pour assurer la sécurité (il peut s’endormir, ou bien quitter le siège). Ce niveau n’est possible que dans certaines zones géographiques ou des situations d’embouteillage ; La voiture s’arrêtera en toute sécurité si le conducteur ne reprend pas le contrôle lorsqu’il en a besoin.

***Niveau 5 :*** (**Automatisation complète**) Ne demande aucune intervention humaine. Consiste en un système doué de capacités de conduite autonome totale et permanente, dans lequel l’humain n’intervient à aucun moment sinon pour indiquer sa destination et se laisser transporter.

Figure 12 : Les niveaux d'autonomie des ADAS

L’autonomie de l’automobile est assurée grâce à l’intervention de différentes fonctions d’aide à la conduite, basées sur l’intégration des différents capteurs permettent de percevoir l’environnement et communiquer l’information aux calculateurs de l’automobile.

Dans la suite, nous allons détailler les fonctions ADAS utilisées pour assurer l’autonomie des véhicules et les différents capteurs et technologies développées pour contrôler ces fonctions.

### Classification des systèmes ADAS :

Les systèmes ADAS peuvent être divisés en systèmes **passifs** et **actifs**.

* Les systèmes passifs avertiront le conducteur d'un danger potentiel et le laisseront au conducteur de braquer ou de freiner pour l'éviter.
* Les systèmes actifs, en revanche, prendront le contrôle du véhicule et dirigeront, accéléreront ou freineront si le conducteur ne le fait pas.

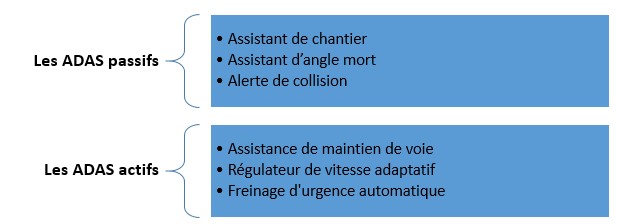


Figure 13: Classification des ADAS



### Exemples des fonctions ADAS :

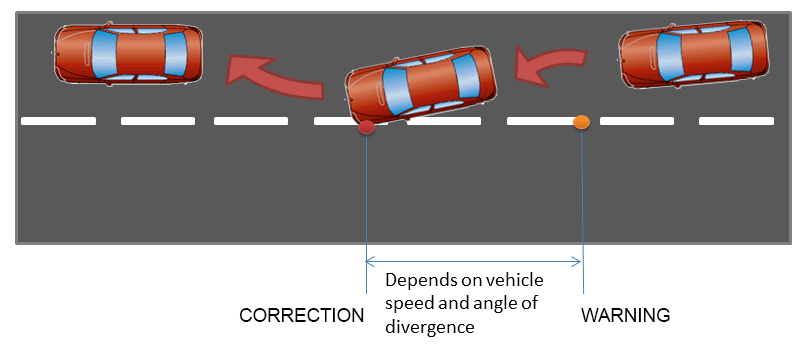


#### Assistant de maintien de voie LKA (Lane Keeping Assist):

Cette fonction est développée afin de minimiser la sortie involontaire de voie en cas d’une inattention momentanée du conducteur, les sous-fonctions utilisées dans l’assistance de maintien de voie sont : Lane Departure Warning et Lane Keeping Assistance (LDW/LKA).

La fonction LDW est un système passif informe le conducteur à travers des alertes quand le véhicule est en train de sortir de la voie. Ces alertes sont souvent sous forme d’une vibration du volant, effet acoustique ou/et messages affichés dans l’IHM.

La LKA est un système passif vise à rendre le véhicule à la voie si aucune action n’est tenue par le conducteur, en utilisant un contrôle de couple de l’arbre du système de direction assistée en boucle fermée.

La figure ci-dessous illustre un cas exemple de l’intervention de la fonction d’assistance de maintien de voie :

La fonction d’assistance de maintien de voie génère une consigne qui sera l’entrée du système de direction assistée, elle est basée principalement sur une caméra montée derrière le pare-brise avec un calculateur attaché à celle-ci (La transformation des données doit être rapide pour assurer la sécurité active).

Figure 14 : Fonction d'assistance de maintien de voie (LKA)

Cette caméra fournit au calculateur d’assistance de maintien dans la voie des images numérisées (environ 25 images/seconde). Le calculateur analyse les images numériques et décide s’il est nécessaire d’effectuer une correction de la direction dans les limites données et calcule aussi combien de temps cette action doit durer.

#### Détection des angles morts BSD (Blind Spot Detection):

Les systèmes de surveillance des angles morts détectent les véhicules dans l’angle mort du conducteur — les zones sur le côté et à l’arrière du véhicule hors de la vue du conducteur — et en avertissent le conducteur lorsqu’un véhicule est détecté. Des capteurs (caméra, ultrasons ou radar) surveillent les véhicules dans les angles morts d’un véhicule et alertent le conducteur avec des signaux lumineux soit dans les rétroviseurs latéraux, soit dans la porte. Si vous activez un clignotant pour changer de voie et qu’un véhicule est présent dans l’angle mort, le système peut vous avertir au moyen de signaux lumineux, sonores ou vibratoires.

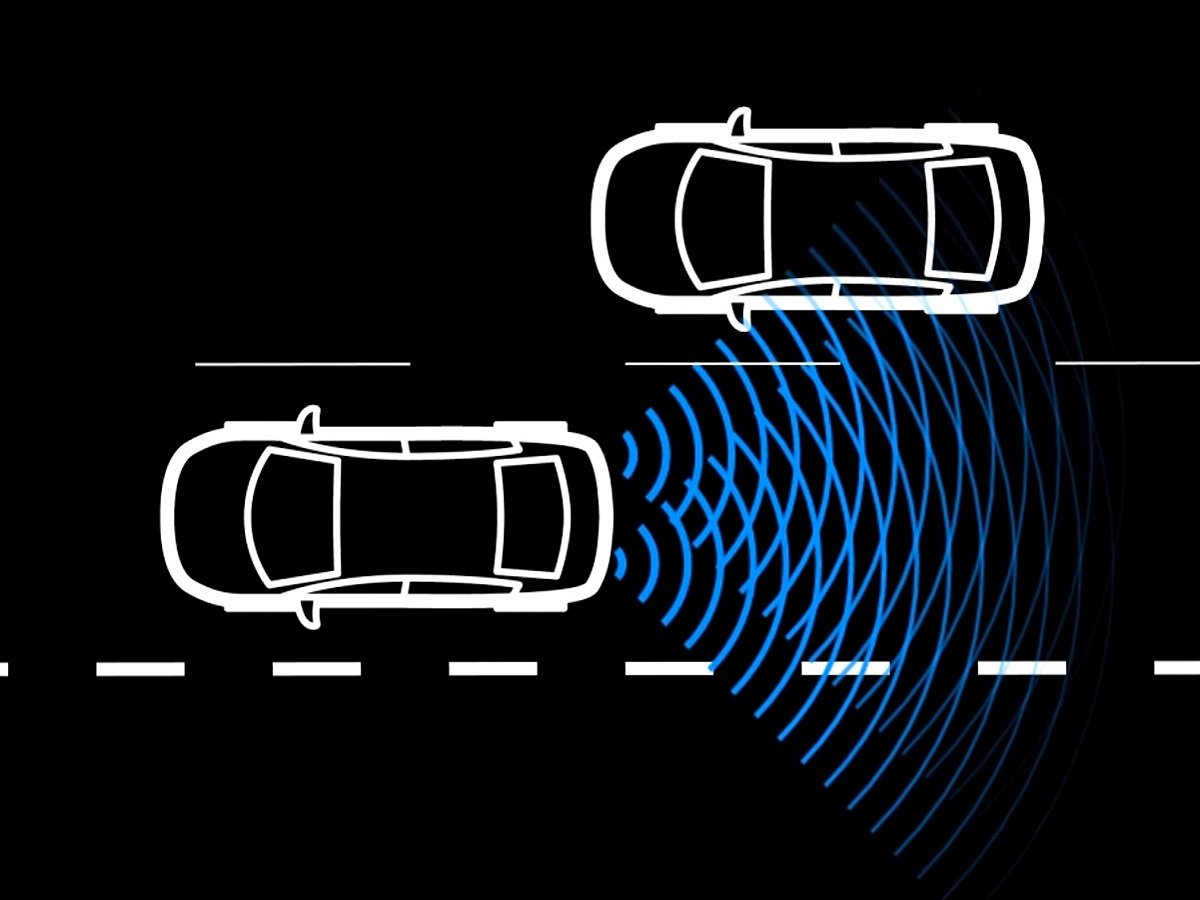
Une image peut apparaître sur le rétroviseur latéral vous indiquant qu’un véhicule se trouve dans votre angle mort, même si votre clignotant n’est pas activé. Cependant, le système n’émettra un signal sonore d’avertissement (comme un bip) que si vous utilisez votre clignotant.

Figure 15: La fonction détection des angles morts (BSD)

#### Adaptive Cruise Control ACC (Adaptive Cruise Control) :

Le régulateur de vitesse adaptatif est une version évoluée du [régulateur de vitesse.](https://voiture.kidioui.fr/lexique-automobile/regulateur-de-vitesse.html) Il ne permet pas uniquement de maintenir une vitesse de croisière : il est également capable de calculer et de conserver une distance de sécurité avec le véhicule qui le précède sur la même voie.

Ce dispositif utilise un capteur radar pour contrôler les obstacles mobiles situés dans la même file à l’avant du véhicule. L’ACC permet de conserver une vitesse constante définie tant qu’aucun autre véhicule n’est détecté. Lorsque l’ACC détecte devant lui un véhicule plus lent, il réduit la puissance du moteur et, le cas échéant, actionne les ralentisseurs et les freins pour conserver une distance de sécurité définie. Une alerte de distance se déclenche si l’intervention du chauffeur est nécessaire au maintien de cette distance.



Figure 16 : La fonction régulation de vitesse adaptative (ACC)

#### Freinage d'urgence automatique AEB (Automatic Emergency Breaking) :

Il actionne le système de freinage en cas de risque de collision imminente sans intervention du conducteur. En cas de risque de collision avec le véhicule qui précède, le système va alerter le conducteur de façon visuelle (système passive) ou freiner à la place du conducteur (système active) si celui-là ne réagit pas afin de réduire l’intensité de l’impact ou même d’éviter totalement la collision.

Ce système est actif à partir de 8Km/h, il détecte des véhicules jusqu’à 180Km/h également des piétons jusqu’à 160 Km/h.

AEB se décline en trois catégories :

-Système à basse vitesse - fonctionne dans les rues de la ville pour détecter d'autres véhicules devant votre voiture afin de prévenir les collisions et les blessures ne mettant pas la vie en danger.

- Système à plus grande vitesse - scanne jusqu'à 200 mètres à l'avance à l'aide d'un radar longue portée à des vitesses plus élevées.

-Système piéton - détecte le mouvement des piétons par rapport à la trajectoire du véhicule pour déterminer le risque de collision. Ce système utilise une caméra et un radar situé au parechoc qui vont analyser en permanence le Traffic.

#### 

#### 

Figure 17: La fonction de freinage automatique d'urgence (AEB)

#### Assistance aux manœuvres de stationnement (Park Assist) :

Le Park Assist est un système actif d’aide au stationnement qui, à la demande du conducteur, détecte un espace de stationnement et pilote automatiquement le système de direction pour un stationnement parallèle au trottoir en adoptant la trajectoire idéale.

Après activation par le conducteur, cet équipement mesure la taille des places dans une file de voitures garées et détecte un espace de stationnement Cette mesure est réalisée à l’aide de capteurs ultrasons, ceux des aides au stationnement avant et arrière aidés par deux capteurs latéraux supplémentaires à l’avant.

Le conducteur garde le contrôle de la mobilité du véhicule (embrayage, rapports de vitesses, freinage, accélérateur) et peut reprendre la main à tout moment en saisissant le volant. Pendant les phases d’entrée et de sortie de stationnement, le système fournit des informations visuelles et sonores au conducteur afin de sécuriser sa manœuvre.



Figure 18 : La fonction d'assistance de stationnement (Park Assist)

### Métier de conception fonctionnelle ADAS

Un ingénieur de spécification ADAS est un ingénieur d’étude automobile qui travaille sur les systèmes d’aide à la conduite tels que la détection des piétons, le freinage d’urgence, la reconnaissance des panneaux de signalisations et bien d’autres applications. Son travail consiste à analyser les besoins exprimés par les automobilistes, les fonctionnalités déployées sur les véhicules de la concurrence, ainsi que les critiques des journalistes automobiles. Cette base de données va nourrir la réflexion de l’ingénieur porté sur l’analyse fonctionnelle du système et détailler les prestations dans le cahier des charges.

Ainsi, avant de concevoir un système d’aide à la conduite, il faut comprendre ce que le client attend, afin de le traduire en spécifications techniques et en architecture système.

L’ingénieur d’étude ADAS peut alors définir l’architecture du contrôle / commande, et valider son fonctionnement via la simulation numérique (ex : Matlab Simulink) et physique (ex : banc de test, prototypage sur véhicule…).

## Conclusion :

Ce chapitre nous a permis de décrire l’architecture générale des véhicules, et précisément des calculateurs ECUs, qui nous intéressent dans notre étude. Il a permis aussi de définir les systèmes d’aide à la conduite et de démontrer l’intérêt de ces systèmes dans la sécurité routière ainsi de les classifier. Dans le chapitre suivant une présentation des outils utilisés dans ce projet sera affichée.

# Chapitre III : Conception et rédaction des spécifications

## Introduction :

Ce chapitre sera consacré aux différents outils utilisés lors de la réalisation du projet. En commençant par la présentation des langages de programmation et de conception du modèle et de l’interface. Ensuite les outils de test et le protocole utilisé pour faire la communication entre Matlab et Python ainsi les plans de tests et finalement la validation et les résultats.

## Environnement logiciel :

### Outils de modélisation et simulation des fonctions ADAS :

#### MATLAB :

Partout dans le monde, des millions d'ingénieurs et de scientifiques utilisent MATLAB pour analyser et concevoir les systèmes et produits de demain. MATLAB est présent dans des systèmes automobiles de sécurité active, des véhicules spatiaux, des appareils de surveillance médicale, des réseaux électriques intelligents et des réseaux mobiles. Il est utilisé dans les domaines de l'apprentissage automatique, le traitement du signal, la vision par ordinateur, les communications, la finance computationnelle, la conception de contrôleurs, la robotique et bien plus.

La plate-forme MATLAB est optimisée pour résoudre les problèmes scientifiques et techniques. Le langage MATLAB, basé sur les matrices, est le moyen le plus naturel au monde pour exprimer les mathématiques computationnelles. Les graphiques intégrés permettent de visualiser facilement les données afin d'en dégager des informations. Grâce à la vaste bibliothèque de boîtes à outils prédéfinies, on peut commencer directement par les algorithmes essentiels à notre domaine d’activité.

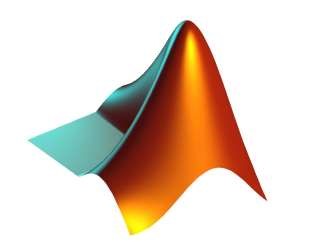


Figure 19 : Logo de MATLAB

#### Simulink :

Simulink est un environnement de diagramme fonctionnel destiné à la simulation multi domaine et à l'approche de conception par modélisation (Model Based Design). Il prend en charge la conception et la simulation au niveau système, la génération automatique de code, ainsi que le test et la vérification en continu des systèmes embarqués.

Simulink propose un éditeur graphique, un ensemble personnalisable de bibliothèques de blocs et des solveurs pour la modélisation et la simulation de systèmes dynamiques. Il est intégré à MATLAB, ce qui permet d'incorporer les algorithmes MATLAB dans les modèles et d'exporter le résultat des simulations vers MATLAB pour compléter les analyses.

L’environnement Simulink peut modéliser un système, simuler son comportement, décomposer le design avant son implémentation. Avec Simulink, il est possible de créer des diagrammes hiérarchiques de blocs pour la modélisation haut niveau d’un système, comme des Blocs Diagrammes de Fiabilité, de construire des simulations complètes, d’intégrer des composants comme un signal analogique, des communications numériques ou des logiques de contrôle.

Simulink peut modéliser des données simples ou multicanaux, des composants linéaires ou non. Simulink peut simuler des composants numériques, analogiques ou mixtes. Il peut modéliser des sources de signaux et les visualiser.

#### Carte FPGA :

La carte FPGA (ou Field-programmable gate array), ce qui peut être traduit en français par (réseau de [portes](https://www.futura-sciences.com/maison/definitions/maison-porte-10855/) programmables sur site) est un [circuit intégré](https://www.futura-sciences.com/tech/definitions/tech-circuit-integre-1875/) fait pour être (re)programmé par l'utilisateur après sa fabrication en utilisant un [langage informatique](https://www.futura-sciences.com/tech/questions-reponses/internet-meilleurs-langages-developpement-moment-10578/) spécifique, donc sans modifier le matériel.

Une FPGA est configurée en utilisant un langage de description matériel comme VHDL ou Verilog, des langages informatiques qui permettent de décrire un circuit électronique et son comportement.



Figure 20: Carte FPGA

#### Langage HDL :

Les FPGA utilisent un type de langage spécial, appelé **HDL** (Hardware Description Language). Il existe deux variantes principales : Verilog et VHDL. Presque toutes les suites de développement comme **Quartus** ou **Vivado** prennent en charge ces deux variantes. Le choix est donc une question de préférence. Ces langages sont ensuite « synthétisés », ce qui équivaut à l'opération de compilation pour les microcontrôleurs. L'outil de synthèse indique au FPGA comment relier les éléments logiques pour créer l'effet décrit par votre code. Lorsque vous débutez, n'oubliez pas un point essentiel : votre code est traduit en instructions matérielles et non en instructions CPU.

Pour simplifier le développement, les fournisseurs de FPGA proposent un catalogue de blocs de code fréquemment utilisés, pouvant facilement être inclus dans une conception. Sont inclus des blocs de communication, de RAM, de multiplicateur, etc. Vous pouvez les configurer et les ajouter à votre conception sans même avoir à vous soucier du code HDL impliqué, ce qui permet d'accélérer le développement.

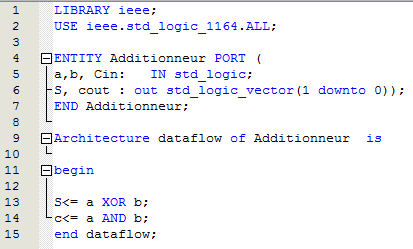
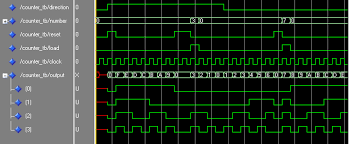


Figure 21: Langage de description matériel VHDL

### Cycle en V

#### Définition :

L’ingénierie de systèmes regroupe l’ensemble des activités de pilotage des projets de construction effective d’un système en s’appuyant sur sa décomposition architecturale produite par l’architecture de systèmes. Ce processus prend classiquement la forme d’un "cycle en V"

Le cycle en V est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et un flux ascendant, qui assemble le produit en vérifiant sa qualité.

#### Phases du cycle en V :

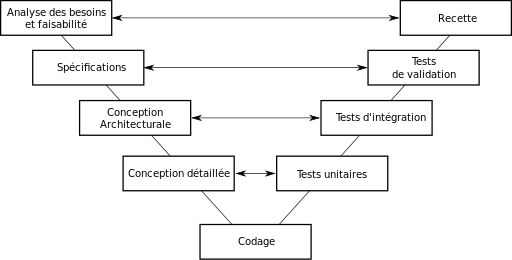


Figure 22: Cycle en V

La partie gauche du cycle en V consiste à créer et développer le système tandis que la partie droite consiste à vérifier le modèle développé par une série de tests pour chacune des étapes de modélisation jusqu’à la livraison d’une recette au client. Il est important de noter que les différentes phases sont souvent réalisées par des équipes ou des personnes différentes. Par exemple, les vérifications par les tests ne sont en général pas effectuées par les mêmes personnes que celles ayant réalisées le logiciel à tester : cela renforce la confiance dans le produit fini en évitant le biais de l’auto-vérification. Le plus souvent, les différents tests sont spécifiés par les équipes qui modélisent le problème et conçoivent la solution, mais ils sont exécutés par les équipes qui opèrent les vérifications et validation.

* ***Analyse du besoin et faisabilité*** *:* Le besoin du client est généralement vague ce qui suscite une écriture d’un cahier des charges plus détaillé. Le client sollicite un produit et il formule ses besoins à travers ce produit. Il définit un délai final de rendu. Le prestataire réalise une étude de faisabilité afin de savoir si la solution peut être conçue et rentable.
* S***pécifications*** : Avant d'entamer la conception, il est indispensable de savoir ce que l'on veut faire pour satisfaire les exigences du client, quelles sont les fonctionnalités à réaliser et les contraintes à respecter. Une spécification est un ensemble explicite d'exigences à satisfaire par un matériau, un produit ou un service. Si un matériau, un produit ou un service ne parvient pas à satisfaire à une ou plusieurs des spécifications applicables, il peut être désigné comme étant hors spécification.
* ***Conception architecturale*** : Cette phase regroupe l'ensemble des activités conduisant à l'élaboration de l'architecture du système. L'objet de la phase de conception préliminaire est donc de répondre à la question : COMMENT ?
* ***Conception détaillée*** : C'est l'ensemble des activités consistant à détailler les résultats de la conception architecturale tant sur le plan algorithmique que sur celui de la structure des données, jusqu'à un niveau suffisant pour permettre l’élaboration du modèle.
* ***Codage*** : Traduire les résultats de la conception détaillée en un programme.
* ***Tests unitaires*** : Activité ayant pour but de vérifier, que chaque composant du logiciel pris isolément donne des résultats conformes au dossier de conception détaillée.
* ***Tests d’intégration*** : Activité consistant à assembler et à tester progressivement les composants du logiciel identifiés lors de la conception préliminaire et contrôlés lors des tests unitaires.
* ***Tests de validation*** : La validation est l'activité conduisant à s'assurer essentiellement au moyen de tests qu'un système est conforme à ses spécifications et au manuel d'utilisation.
* ***Recette*** : C'est l'activité qui consiste pour un fournisseur à présenter au client le logiciel étudié et/ou réalisé, et, pour ce dernier, à contrôler que celui-ci est conforme à toutes les exigences contractuelles.

### Conception basée sur modèle (Model-Based Design) :

#### Définition :

Un modèle, c’est un ensemble d’équations, lois ou règles qui décrivent le comportement d’un système plus ou moins finement, selon les phénomènes à étudier. Les prototypes physiques peuvent être remplacés par des modèles virtuels, ajustés (calés) sur la base de données expérimentales pour se rapprocher le plus possible de la réalité. La simulation permet alors d’étudier le comportement du système dans tous types de situations (scénarios).

La Conception basée sur modèle (Model-Based Design : MBD) est une méthode de gestion de projet informatique qui permet d’améliorer le développement d’un système. Alors qu’un développement classique est basé sur le cycle en V, l’approche MBD s’intéresse sur le design du système tout en structurant et en enrichissant la communication entre les différentes équipes chargées du développement. Le coût de la modélisation est compensé par le gain en temps de développement, la convergence plus rapide de la solution et la réutilisation des modèles créés. Nécessitant un fort investissement initial, le Model-Based Design est essentiellement utilisé pour dimensionner les contrôleurs/régulateurs sur les projets complexes, à forte valeur ajoutée (aéronautique, automobile, industrie...).

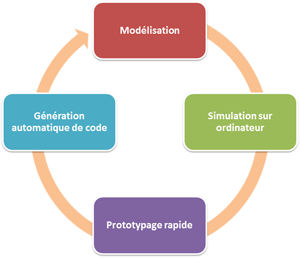


Figure 23: Model Based Design

#### Tests In the Loop :

Les solutions hardware in the loop (HIL), software in the loop (SIL) et model in the loop (MIL) permettent de valider les solutions développées en situation réelle simulée. Leur utilisation accélère la conception et améliore la maîtrise de la qualité, tout en réduisant le recours aux prototypes réels et aux essais physiques.

Voici une rapide définition des solutions de simulation « dans la boucle », par ordre chronologique :

* **MIL:** model in the loop. L’ensemble du système est modélisé, ce qui permet de simuler un environnement complet (véhicule par exemple) afin de tester les lois de commande et de corriger les erreurs mécaniques, électroniques et logicielles avant la fabrication de prototypes (validation fonctionnelle);
* **SIL:** software in the loop. Le code qui sera utilisé ensuite dans le calculateur est testé unitairement, afin de corriger les erreurs en simulant le fonctionnement du système réel (validation de la génération de code);
* **HIL:** hardware in the loop. Les lois de commande sont intégrées dans un calculateur physique relié à l’environnement de simulation, et parfois à une partie mécanique réelle, afin de tester l’implémentation matérielle du contrôle, les temps de réponse, etc. (validation de l’intégration).

#### Analogie avec le projet :

##### Fonction BSD :

* **MIL:** Comme on a dit, il s’agit d’une modélisation de l’ensemble du système. Nous avons modélisé le système global (BSD), avec la simulation de la fonction (figure 23)

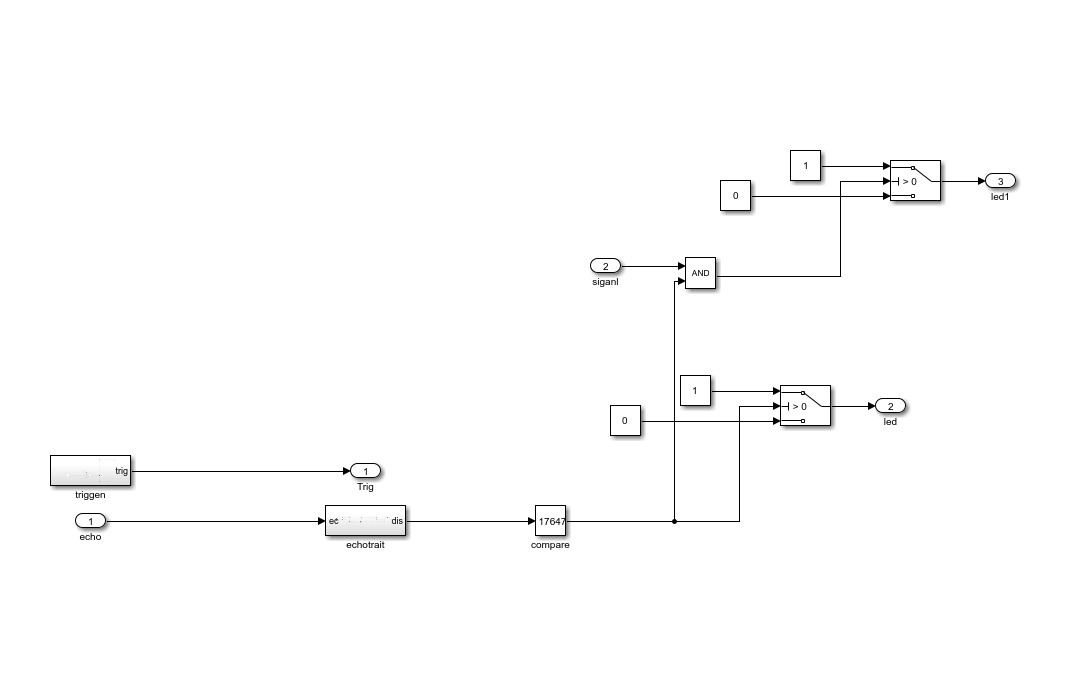


Figure 24: Modélisation de la fonction BSD

* **SIL:** La création du code VHDL de la fonction (figure 24) et la vérification du code par un test bench (figure 25)

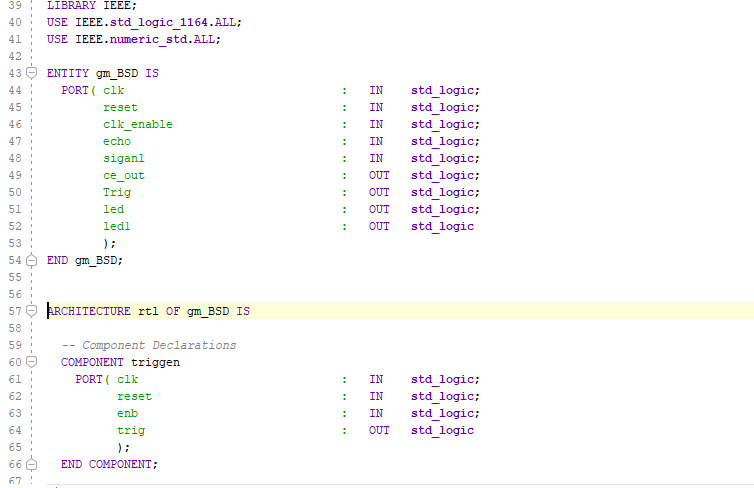


Figure 25: Code VHDL de la fonction BSD

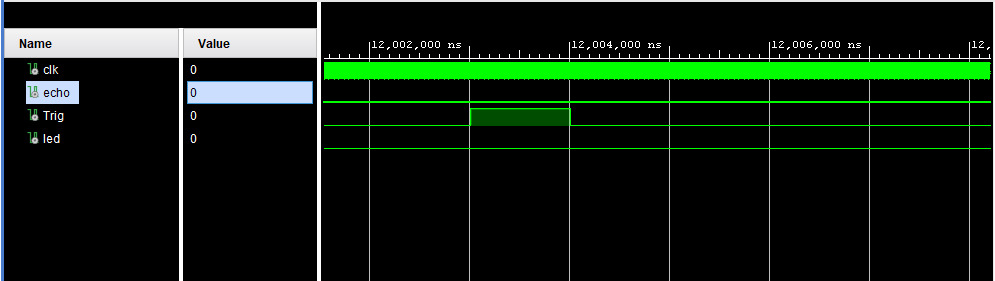


Figure 26: Test Bench fonction BSD

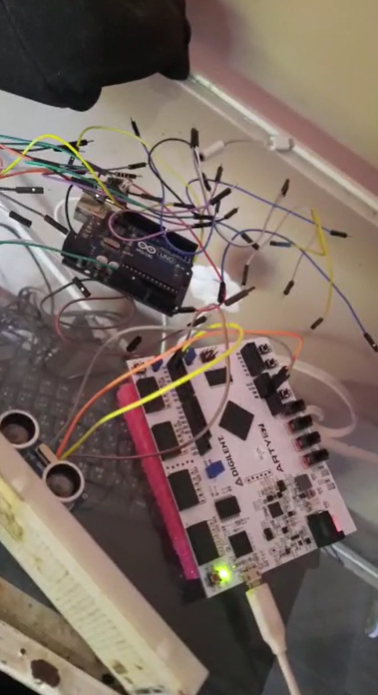
* **HIL:** On a effectué l’implémentation du code VHDL dans la carte FPGA et bien sûr, voir les résultats obtenus concrètement (figures 27-28)

Figure 27 : Implémentation du code dans la carte FPGA (pas d’obstacle)

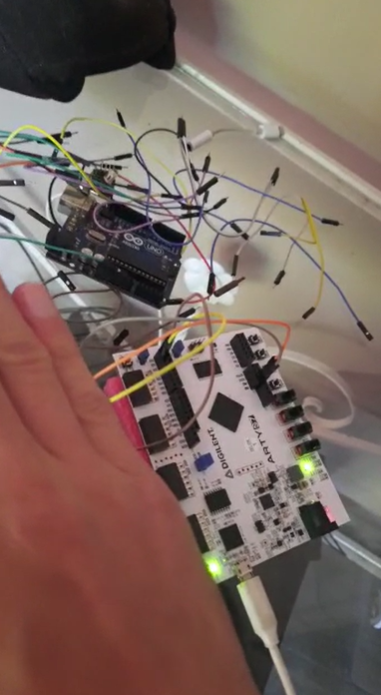


Figure 28: Implémentation du code dans la carte FPGA (avec obstacle)

### Architecture générale :

Avant de commencer la conception, on définit tout d’abord l’architecture générale sous forme d’un schéma qui illustre la conception de base du modèle (pour chaque fonction). Ils existent deux parties principales : les entrées et les sorties.

Figure 30: Entrées / Sorties de chaque fonction

## Acquisition des informations :

Dans le but de garantir une meilleure détection des conditions de l’environnement de conduite, il est nécessaire de faire une étude sur les capteurs utilisés dans les systèmes ADAS.

### Etude sur les capteurs :

#### Capteur Ultrason :

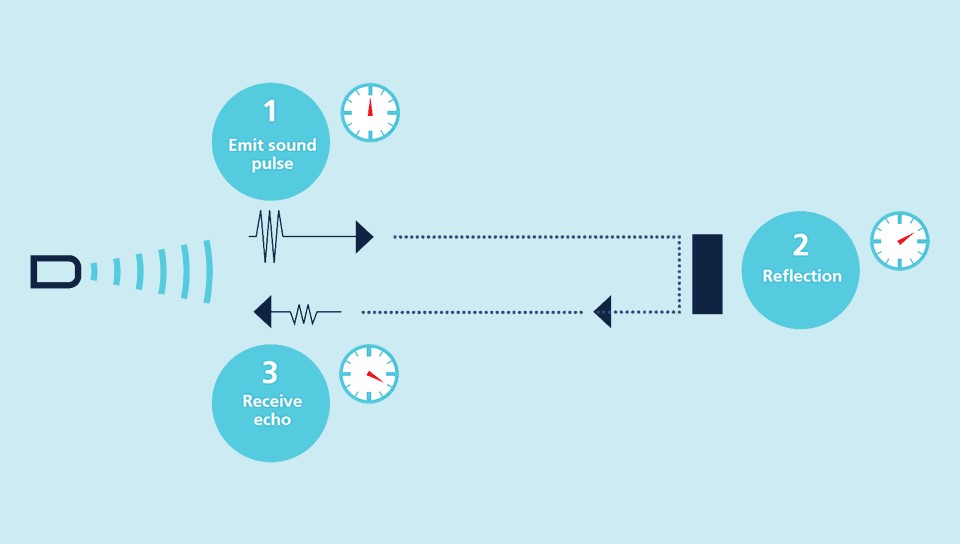
Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l’air à la vitesse du son. Lorsqu’elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d’écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l’émission du signal et la réception de l’écho.

Figure 31: Principe du capteur ultrason

#### Radar :

**Fonctionnement :** Le radar (RAdio Detecting And Ranging) transmet une onde électromagnétique à l’aide d’une antenne. Cette onde rebondit sur un objet dans le trajet du faisceau radar, revient et est reçue. Les échos radar sont générés par tous les matériaux électriquement conducteurs, en particulier par tous les véhicules qui composent la circulation routière. Par conséquent, le radar est particulièrement bien adapté en tant que principe de mesure de distance. En outre, le radar offre des avantages dans des conditions météorologiques défavorables (comme le brouillard ou la pluie), car il utilise une longueur d’onde plus grande que les méthodes optiques. D’autres principes de mesure de distance (tels que les appareils de mesure de distance optique) nécessitent des surfaces ayant de bonnes propriétés de réflexion. Les objets avec des réflecteurs optiques sales ou non visibles ne peuvent pas être détectés de manière fiable.

**Mesure de temps de propagation :** Pour toutes les méthodes radar, la mesure de distance est basée sur une mesure de temps de propagation directe ou indirecte pour le temps écoulé entre la transmission du signal radar et la réception de l’écho du signal. En réflexion directe, ce temps τ est égal à (deux fois) la distance au réflecteur et la vitesse de la lumière c **: τ = 2d/c.**

**Effet Doppler :** Pour un objet se déplaçant par rapport au capteur radar avec une vitesse relative vrel, l’écho du signal subit un décalage de fréquence fd par rapport au signal émis. Pour les vitesses différentielles énumérées ici, ce décalage est : **fd = −2 fc. vrel/c**

Avec : fc : fréquence porteuse du signal. , vrel : vitesse relative.

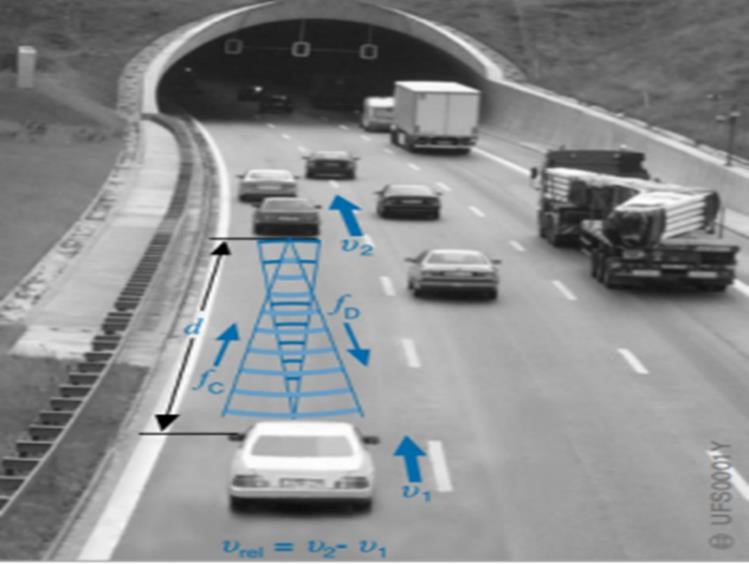


Figure 32: Effet Doppler

## Diagrammes :

### Identification des besoins :

#### Diagramme bête à cornes :

Un diagramme bête à cornes est un outil pour l’analyse fonctionnelle du besoin. C’est un schéma qui démontre si le produit est utile pour l’utilisateur, s’il répond à ses besoins.

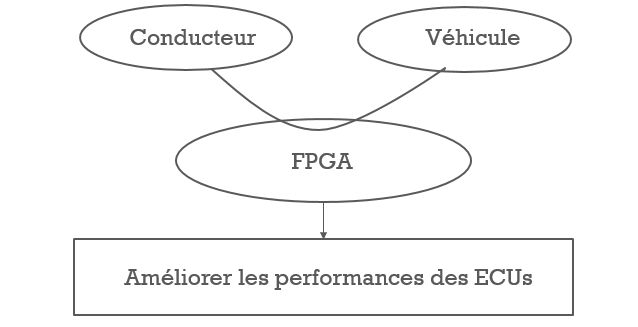


Figure 33: Diagramme bête à cornes

#### Diagramme de pieuvre :

Le diagramme pieuvre ou graphe des interactions est un schéma qui représente la relation entre un produit/service et son environnement. C’est un outil d’analyse utilisé dans le cadre de la méthode APTE. Le diagramme pieuvre permet de représenter les fonctions de service d’un produit. C’est-à-dire qu’il permet de voir quelles sont les fonctions essentielles et secondaires d’un produit et comment ces fonctions réagissent avec le milieu extérieur.

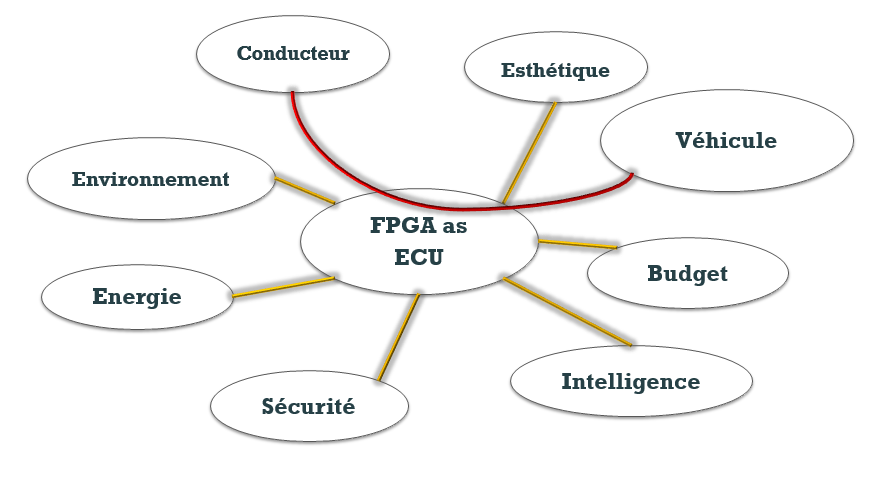


Figure 34: Diagramme de pieuvre

#### Diagramme de contexte :

Le diagramme de contexte permet de définir les limites de l'étude. Il place le diagramme dans son contexte en listant les acteurs ou éléments qui agissent ou interagissent avec lui. Le degré de raffinement de ce diagramme dépend de ce que l'on veut montrer.

Ce diagramme exprime l’environnement du système dans une situation donnée. Le diagramme de contexte permet donc de définir les frontières de l’étude et de préciser la phase du cycle de vie dans laquelle on situe l’étude (généralement la phase d’utilisation).

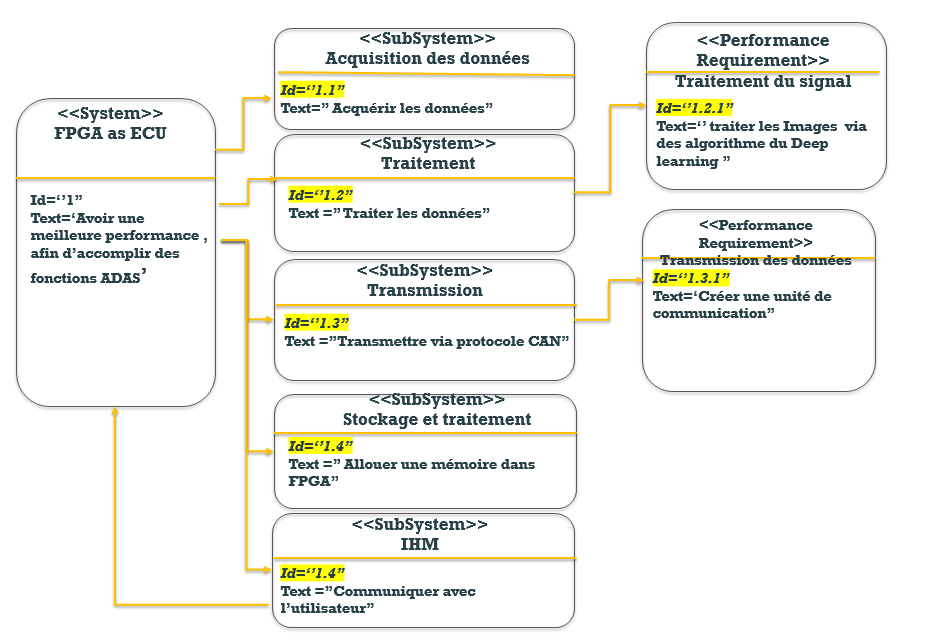


Figure 35: Diagramme de contexte