



Mémoire de Projet de fin d'étude

Préparé par

Abdelmonim MTEKLILT
Fatima-ezzahra HOUCHAIMI

Pour l'obtention du diplôme

Master Sciences et Techniques
Electronique, Signaux et Systèmes Automatisés
(E.S.S.A)

Intitulé

**Etude et développement d'un outil de
communication et diagnostique pour le Module
électronique avant de la voiture BMW**

Encadré par :

Pr. Hicham GHENNIoui (FST Fés)

Mr. Mounir AMRAoui (LEAR Corporation)

Soutenu le 15 Juin 2016, devant le jury composé de :

Pr. Hicham GHENNIoui: Encadrant

Pr. Farid ABDI : Examineur

Pr. Mouhcine RAZI: Examineur

Remerciements

Au moment où s'achève ce projet de fin d'étude, nous tenons à adresser nos sincères remerciements et nos profonds respects au Mr. Mounir AMRAOUI pour nous avoir confié ce projet et assuré un encadrement de qualité tout au long de sa réalisation.

Nos remerciements s'adressent également au Pr. Hicham GHENNIQUI, notre encadrant à FST Fès (FSTF), qui par ses indications nous a permis de venir à bout de diverses difficultés.

Nous tenons à remercier également M. Tarik ADANI, Manager du département industrialisation & test engineering, de nous avoir accueilli comme stagiaires au sein de son département.

Nos remerciements vont aussi au corps de personnels d'entreprise pour leur disponibilité à nous offrir toutes les conditions de travail dont nous avons besoin durant la période de la réalisation de ce projet.

Nous tenons à remercier vivement les membres de jury qui ont accepté de juger notre travail.

Notre pensée va enfin à tous nos Professeurs tout au long de cette bonne formation pendant deux ans, car sans eux nous n'aurons pas à la hauteur de ce niveau et nous tenons à remercier nos amis, collègues, et étudiants pour toutes les expériences que nous avons vécu ensemble, d'une manière ou d'une autre, nous avons constitué un environnement favorable pour la réalisation de ce projet de fin d'étude.

Dédicaces

Nous tenons à dédier ce travail à tous ceux qui nous ont aidés de près ou de loin, à la réalisation du projet.

A nos chers parents : Pour leurs sacrifices, leurs patiences et leurs soutiens moraux et matériels, aussi leurs conseils qui nous avons guidé tout au long de notre chemin d'étude.

A notre Encadrant M. Mounir AMRAOUI au sein de l'entreprise, pour le temps qu'il a consacré et pour les précieuses informations qu'il nous a prodigué avec intérêt et compréhension.

A tous nos professeurs, Pour leurs aides et soutiens pendant toute notre période de formation.

A nos chers amis, que ce travail reste le symbole des moments passés ensemble.

Et à vous chers lecteurs.

Résumé

Tout d'abord le domaine du « testing » occupe une importance particulière dans le secteur industriel. En plus, en général la notion du « testing » ne se limite pas dans le fait de vérifier l'état des systèmes utilisés dans l'industrie, mais elle s'étend à la remise en cause des technologies existantes et suscite la critique de l'existant et l'essor vers de nouvelles techniques qui résolvent les problèmes et les faiblesses techniques.

Notre projet permet d'accélérer ce processus de développement de tests en électronique au sein de la société « Lear corporation Rabat ». Notre objectif est alors de faire des tests de diagnostics puissants permettant de prouver que les cartes fonctionnent correctement ou de diagnostiquer la cause exacte du défaut. Lorsque, la carte ne fonctionne pas correctement, La détection rapide des anomalies, et aussi l'identification des défauts qui sont des éléments clé dans notre projet car elles permettent d'isoler avec précision les défauts au niveau « hardware ».

Enfin, ce niveau exceptionnel de constatation de défaut réduit le nombre des cartes électroniques à réparer. Il permet aussi d'économiser du temps et surtout d'améliorer la fiabilité des cartes réparées.

Abstract

The field-testing has a special importance in the industrial sector. The concept of testing are not limited to the effect checking the condition of the systems used in the industry in general, but it extends to the questioning of existing technologies and arouses criticism of the existing and the development towards new technologies that solve technical problems and weaknesses.

Our project helps to accelerate the industrial development process in the company "Lear Corporation Rabat" and opens the doors to new ways.

In addition, it aims at making powerful diagnostic tests designed to prove that the cards work correctly or diagnose the exact cause of the fault when the card is not working properly. The rapid detection of anomalies and the fault identification are the key elements of our project. It can isolate precisely the defects in the hardware level. This exceptional level defect finding, reduces the number of the electronic circuit boards to repair and it saves time unparticular improve the reliability of the repaired cards.

Table des matières

Remerciements	1
Dédicaces	2
Résumé	3
Abstract	4
Liste des figures	8
Liste des tableaux.....	10
Liste des abréviations.....	11
Introduction générale	12
Chapitre 1 : Présentation de la société et description du projet	13
1.1 Présentation générale de LEAR Corporation.....	14
1.1.1 Aperçu historique	14
1.1.2 Les marchés principaux de LEAR	14
1.1.3 Les clients de LEAR	14
1.2 Lear Corporation- Technopolis Rabat.....	15
1.2.1 Présentation	15
1.2.2 Organisation de la société	16
1.2.3 Organigramme de la société	16
1.2.4 Organisation de la société	17
1.3 Produits et processus de production à Lear Rabat	18
1.3.1 Les différents produits.....	18
1.3.2 Processus de fabrication	19
1.4 Description du projet et méthodologie de travail.....	22
1.4.1 Problématique.....	22
1.4.2 Cahier des charges.....	23
1.4.3 Description du Projet.....	23
1.4.4 Plan d'action	24
1.4.5 Méthodologie de travail	27
Conclusion	28
Chapitre 2 : Spécifications fonctionnelles et techniques	29
2.1 Importance de test	30
2.2 La voiture cible BMW I8	31
2.3 Fonctionnalité de la carte PL7 (Platform 7 Front electronic Module).....	32
2.4 Hardware de la carte électronique PL7	33
2.4.1 Body (Le microcontrôleur SPC5646C)	34
2.4.2 Gateway (Le microcontrôleur MPC5668G)	36

2.4.3 ECL : Le microcontrôleur MC9S08SG32.....	37
2.4.4 Les driver	38
2.4.5 Diagramme de la carte PL7.....	39
2.4.6 Les connecteurs.....	41
2.5 Mise en œuvre la communication avec la Carte	42
2.5.1 Description de BUS CAN	42
2.5.2 Le choix de BUS CAN	43
2.5.3 Le protocole de communication CAN.....	43
2.5.4 Le CANcaseXL	48
2.5.5 La connexion de la carte.....	49
Conclusion	49
Chapitre 3 : Algorithme et méthode de diagnostic du module PL7 FEM.....	50
3.1 Procédure du diagnostic de la carte PL7	51
3.1.1 Présentation de la station de test	51
3.1.2 Description des étapes du diagnostic.....	53
3.2 Les Diagrammes de système du diagnostic.....	55
3.2.1 Diagramme de communication par protocole CAN	55
3.2.2 Diagramme de système de diagnostic général	57
Conclusion	57
Chapitre 4 : Conception et développement du système de diagnostic	58
4.1 Conception par le langage UML	59
4.1.1 Diagramme de cas d'utilisation général	60
4.1.2 Diagramme d'activité générale	61
4.1.3 Diagramme de séquence générale.....	63
4.2 Conception de deux manières de test.....	65
4.2.1 Test détaillé	65
4.2.2 Test Complet	69
4.3 Réalisation et tests de vérification du système de diagnostic	73
4.3.1 Environnement du travail.....	73
4.3.2 Développement.....	77
4.4 Phase de Test et Validation	78
4.5 Les interfaces graphiques.....	79
4.5.1 Interface de Test.....	79
4.5.2 Interface de statistique	80
4.5.3 Interface d'analyse hardware.....	81
4.5.4 Interface de configuration.....	81

Conclusion	83
Chapitre 5 : Etude technico-économique	84
5.1 Les dépenses actuelles	85
5.2 La pertinence du projet	86
Conclusion	86
<i>Conclusion générale</i>	87
<i>Webographie</i>	88
<i>Annexes</i>	89
Annexe A : Le fichier des fonctions	90
Annexe B: Connecter	91
Annexe C : Lancer les tests :	92
Annexe D : Schéma électrique de CAN :.....	93
Annexe E : Les données techniques de Cancasexl.....	94

Liste des figures

Figure 1 : Clients de LEAR Corporation.....	15
Figure 2 : Organigramme de la société Lear.....	16
Figure 3 : Direction de la société.....	17
Figure 4 : Zone de production	17
Figure 5 : Rework.....	18
Figure 6 : Schéma de processus assemble	23
Figure 7 : Les tâches du projet	25
Figure 8 : Diagramme de gant	26
Figure 9 : Démarche du projet par Cycle en V.....	27
Figure 10 : Processus design.....	28
Figure 11 : Coût de l'électronique embarquée exprimé sous forme de pourcentage par rapport au coût total des automobiles dans le monde de 1950 à 2030.....	30
Figure 12 : Les symboles du Module FEM.....	33
Figure 13 : Les composants électroniques	34
Figure 14 : Microcontrôleur MPC564.....	35
Figure 15 : Block diagramme MPC5668G.....	37
Figure 16 : Block Diagramme MC9S08SG32.....	38
Figure 17 : Diagramme de la carte FEM	40
Figure 18 : Les connecteur de la carte PL7.....	41
Figure 19 : Couplage physique du réseau CAN	42
Figure 20 : Débit	43
Figure 21 : Domaines d'application.....	44
Figure 22 : le bus CAN dans l'automobile.....	44
Figure 23 : Block de bus CAN protocole	45
Figure 24 : Les connecteurs de CAN.....	45
Figure 25 : Formats de trames	46
Figure 26 : Cancase XL.....	48
Figure 27 : Connexion de la carte PL7	49
Figure 28: Block diagramme Rework.....	52
Figure 29 : Diagramme de communication du protocole CAN	56
Figure 30 : Diagramme du système de diagnostic	57
Figure 31 : Diagramme de cas d'utilisation générale	60
Figure 32 : Diagramme d'activité générale	62
Figure 33 : Diagramme de séquence générale.....	64
Figure 34 : Diagramme De cas d'utilisation de test détaillé.....	65
Figure 35 : Diagramme d'activité de test détaillé	67
Figure 36 : Diagramme d'activité de test détaillé	68
Figure 37 : Diagramme de cas d'utilisation de test complet.....	69
Figure 38 : Diagramme d'activité de Test complet	71
Figure 39 : Diagramme de séquence de « Test complet»	72
Figure 40 : Environnement matériel	73
Figure 41 : Driver CAN	76
Figure 42 : Diagramme de MVC	77
Figure 43 : Interface de Test.....	80
Figure 44 : Interface de statistique	80
Figure 45 : Interface Hardware	81
Figure 46 : Interface de configuration.....	81

Figure 47 : Interface de test détaillé	82
Figure 48 : Exemple de test complet.....	82
Figure 49: Station de test	85

Liste des tableaux

Tableau 1 : Les différents produits.....	19
Tableau 2 : Processus de fabrication.....	22
Tableau 3 : Tests de vérification.....	79
Tableau 4 : Le coût actuel d'analyse sur la station Rework.....	85
Tableau 5 : La pertinence du projet	86

Liste des abréviations

- PCB : Printed circuit board
- CAN : Controller Area Network
- VB : Visual Basic
- BDC : Body Domain Controller
- DUT : Drivers Under Test
- EOL : End of Line
- PL3 : Platform 3
- PL7 : Platform 7
- FEM : Front Electronic Module.
- AOI : Automated Optical Inspection
- FKT : Functional Test
- ICT : In Circuit Test
- OBP : On Board Programming
- UML : Unified Modeling Language
- NOK : Non Ok
- ID : Identification
- ISO : International Standards Organisation

Introduction générale

Notre formation en Master Electronique Signaux et Système Automatisé au sein de la Faculté des Sciences et Techniques de Fès (FSTF) doit être se clôturer par un projet de fin d'étude qui est une phase de manipulation et d'application de différentes études et connaissances acquises durant la formation.

Nous avons effectué le stage au sein de l'entreprise <<Lear Corporation Rabat>> d'une durée de quatre mois du 01/02/2016 jusqu'au 01/06/2016, nous avons choisi comme sujet « Etude et développement d'un outil de communication et diagnostique pour le Module électronique avant de la voiture BMW ».

Notre objectif est de développer une application avec une interface graphique pour l'analyse et le diagnostic. Cette dernière doit communiquer avec la carte via CAN afin de reproduire le défaut. Nous devons faire la mesure et donner le feed-back des résultats par rapport à une spécification connue.

Ce document est organisé de la façon suivante,

- Le premier chapitre est consacré au contexte général du projet. Nous présentons l'organisme d'accueil, « Lear corporation » et ses lignes de production. Nous introduisons également dans ce chapitre la problématique étudiée dans ce projet.
- Le deuxième chapitre est dédié à l'analyse fonctionnelle et technique.
- Le troisième chapitre concerne le développement de l'algorithme utilisé dans le diagnostic.
- Le quatrième chapitre introduit la conception de l'application. Nous présentons dans ce chapitre l'environnement matériel et logiciel pour pouvoir réaliser les interfaces graphiques.
- Et finalement dans le dernier chapitre, nous fournissons une étude technico-économique basique afin de mettre en valeur notre projet.

Chapitre 1 : Présentation de la société et description du projet

1.1 Présentation générale de LEAR Corporation

1.1.1 Aperçu historique

Lear Corporation est une société américaine spécialisée dans la fabrication et la distribution d'équipements intérieurs automobiles. A la fin de juin 2009, elle était le deuxième fabricant de sièges automobiles au monde. En 2006, elle possédait 242 unités de production dans 33 pays, employant ainsi environ 90 000 personnes et avait un chiffre d'affaire 17.8 milliards USD. Son siège est situé à SOUTHFIELD au Michigan, Etats-Unis et son nom apparaît sur la liste Fortune 500.

Lear a pris de l'expansion dans les années 1980 et 1990 grâce à une série d'acquisitions. Elle tentait de devenir un distributeur complet d'équipements intérieurs automobiles (Sièges, circuits électriques, produits du plancher, garnitures intérieurs, etc...).

Cette stratégie a subi d'importants revers à la fin de l'année 2005 lorsque les fabricants automobiles ont décidé de ne pas se fier à un seul fournisseur pour les équipements intérieurs. En juin 2009, Lear a manifesté son intention de déposer son bilan. Elle a annoncé au 7 juillet 2009 qu'elle demandait le placement sous la protection du chapitre 11 de la loi américaine sur les faillites. Elle a été particulièrement affectée par la politique de baisse de production en véhicules mise en place par ses clients General Motors, Ford et Chrysler. Cette faillite est la 13ème plus grosse en termes de capitalisation depuis le début de l'année aux Etats-Unis.

1.1.2 Les marchés principaux de LEAR

Lear Corporation intervient principalement dans,

- composants électronique pour automobiles,
- câblage automobile,
- et siège automobile.

1.1.3 Les clients de LEAR

La nouvelle installation de Lear à Rabat fournit du matériel électronique du premier ordre au niveau de l'industrie, notamment des chargeurs de batterie et des modules de point d'accès à des constructeurs automobiles européens tels que,



Figure 1 : Clients de LEAR Corporation

1.2 Lear Corporation- Technopolis Rabat

Lear exploite déjà trois usines au Maroc, dont deux sont localisées à Tanger. Elles sont spécialisées dans la production de systèmes de câblage électrique et de sièges automobiles. La troisième se trouve à Rabat. Elle fournit des produits électroniques tels que des chargeurs de batterie et des modules d'interface aux constructeurs automobiles européens tels que BMW, Renault et Volkswagen.



1.2.1 Présentation

"Lear Electronics Technology Corporation" a décidé de s'installer à Technopolis Rabat-Salé, à travers une unité de production dont la réalisation nécessitera 140 millions de dirhams. La convention relative à cette implantation a été signée à Casablanca, par M. Alexandre Brue, PDG de Lear Maroc et M. Mohamed Lasry, administrateur directeur général du pôle offshore et technologies du groupe Medz, filiale de CDG-Développement.

La cérémonie de signature s'est déroulée en présence de M. Ahmed Réda Chami, ministre du Commerce, de l'Industrie et des Nouvelles Technologies, Anas Alami, directeur général de la CDG, Mariano De Tores, PDG de la division électrique et électronique de Lear Europe et Afrique et Fathallah Sijilmassi, directeur général de l'Agence marocaine de Développement des Investissements (AMDI).

La société Lear a annoncé l'ouverture officielle de sa nouvelle installation électronique à Rabat Technopolis le 18 Mai 2011. La construction de l'usine a débuté en Avril 2010 et s'est achevée en Décembre de la même année. L'installation occupe une superficie de 14.340 m² ; 8200 m² pour la production, 2380m² dédiée au stockage de la matière première et produits finis et 3760 m² pour l'administration et autres. Elle emploie plus de 400 personnes. A l'horizon 2013, la nouvelle unité compte passer à 700 emplois, et multiplier par 4 ses investissements.

Lear est un leader mondial en matière de systèmes de gestion électronique et électrique complets destinés à l'industrie automobile partout dans le monde. La nouvelle installation de Rabat s'intéresse à l'activité EPMS : Electrical : (Power Management Systems). Elle permet de fournir des gammes de systèmes de distribution d'énergie appelés boîtes de jonction intelligentes **SJB** (Smart Junction Boxes). Ce système gère de façon intelligente la distribution d'énergie et contrôle les différentes fonctionnalités électroniques et électriques de la voiture.

1.2.2 Organisation de la société

Un des points forts de Lear corporation est sa souplesse d'organisation. Pour chaque client, Lear met en place une ou plusieurs lignes de production et à chaque pôle d'activités correspond une organisation en termes de gestion, industrialisation, ingénierie maintenance, logistique et qualité. Les moyens mis en place permettent de répondre au mieux à toute demande.

1.2.3 Organigramme de la société

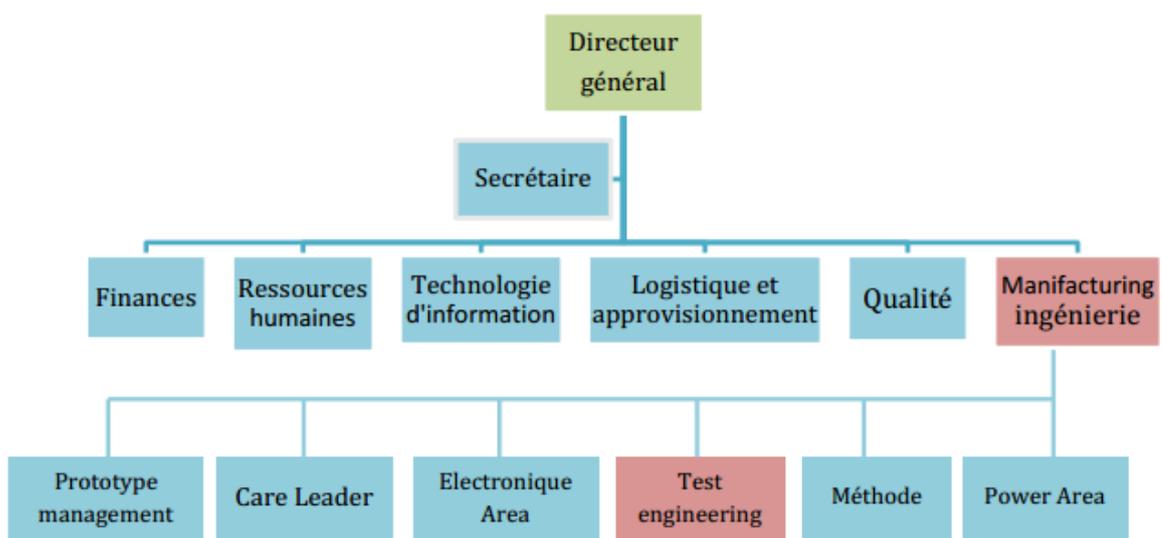


Figure 2 : Organigramme de la société Lear

1.2.4 Organisation de la société

La société se devise en trois grandes zones :

i) Zone bureaux

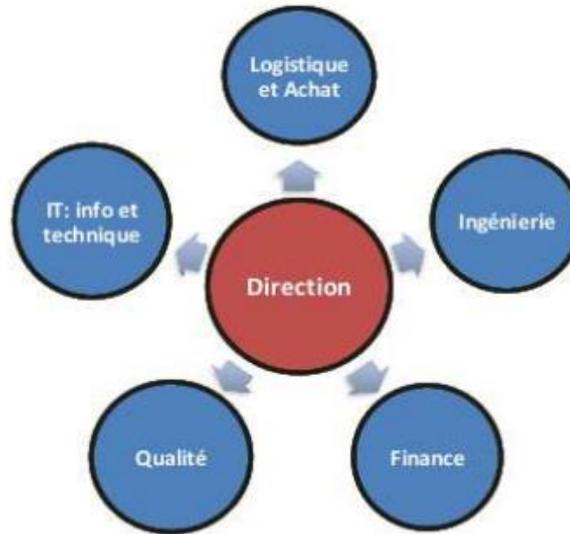


Figure 3 : Direction de la société

ii) Zone de production

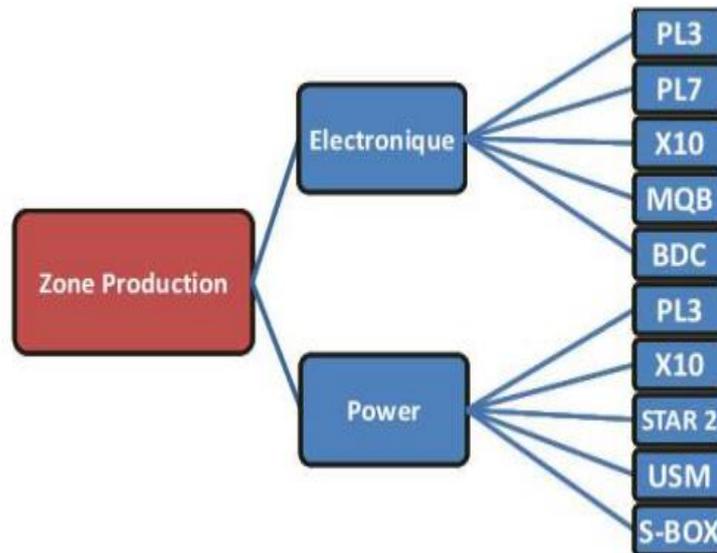


Figure 4 : Zone de production

iii) Rework

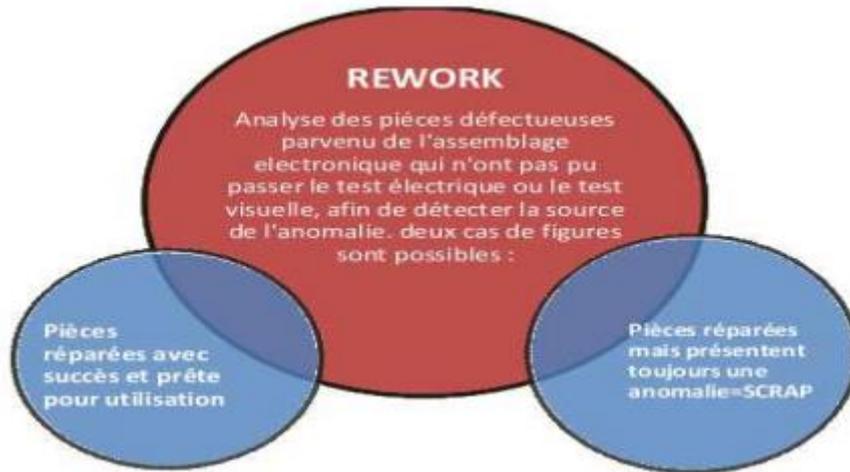
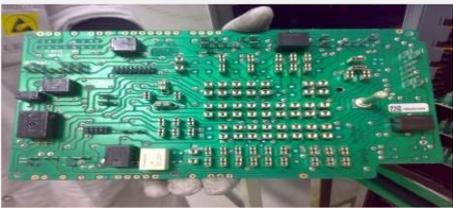


Figure 5 : Rework

1.3 Produits et processus de production à Lear Rabat

1.3.1 Les différents produits

A travers un processus de fabrication bien précis, l'usine Lear de Rabat fabrique principalement quatre produits et conquiert par conséquent le marché européen.

Produit	Explication	Clients
PL3 : 	Il s'agit d'un smart Junction box destiné à BMW-MINI Cooper. Selon les fonctions affectées à chaque variante, on distingue plusieurs niveaux : High, Low, Service, High DCDC et service DCDC.	BMW MINI-COOPER 
PL7 : 	Le PL7 représente des modèles d'intégration d'une haute technologie, en utilisant moins de composants en comparaison avec le PL3 afin de réduire les dimensions de la Junction box, le cout et le délai d'assemblage.	BMW 

<p>STAR2 :</p> 	<p>La distribution de l'alimentation et la protection des cartes électroniques</p>	<p>Mercedes</p> 
<p>MQB :</p> 	<p>Le produit est un gateway avec trois différentes variantes.</p>	<p>Volkswagen, land-rover</p> 

Tableau 1 : Les différents produits

1.3.2 Processus de fabrication

Notre sujet est basé sur la carte électronique PL7, alors nous nous intéressons à l'enchaînement de production de cette carte et au processus de fabrication de toutes les étapes décrites ci-dessous. [1]

ETAPES	MACHINE	FONCTIONNEMENT
<p>1 - Laser machine</p>		<p>La première opération consiste à imprimer un code barre sur la carte PCB, pour sauvegarder tous les informations ou bien l'historique de la carte.</p>
<p>2- Sérigraphie</p>		<p>DEK est une machine de sérigraphie entièrement automatique qui permet de déposer la pâte à braser sur le PCB à l'aide des raclettes et d'un pochoir, avec un temps de cycle de juste de 18 secondes. [2]</p>

<p>3- cyber optique</p>		<p>Machine d'inspection de la pâte à braser d'un temps de cycle de 17 secondes. Cette technologie permet de contrôler intégralement le dépôt de pâte à braser : bonne quantité, bon emplacement et bonne épaisseur.</p>
<p>4- Placement des composants à grande vitesse modulaire</p>		<p>La Machine PANASONIQUE qui permet de placer avec une grande précision tous les types de composants actuellement câblés sur des cartes CMS. Elle est commandée par un micro-ordinateur, et elle utilise une caméra permettant de détecter l'emplacement correcte des composants.</p>
<p>5- Four de refusions (soudage)</p>		<p>Un four à tapis multizones (zones de préchauffage, de refusions et de refroidissement) que l'on peut régler indépendamment, et qui permet de fusionner la pâte à braser et à créer des joints brasés entre les connexions des composants et les PCB.</p>
<p>6- Test optique automatique à analyse 3D</p>		<p>Test AOI OMRON VT-WIN2 est une machine de test optique automatique à analyse 3D. elle permet de mettre en évidence la présence de défauts, en comparant le graphisme de la couche interne avec celui d'un circuit correct.</p>

7- Test les circuits ouverts, les court-circuités, et la programmation des microcontrôleurs



BOARD TESTER SPEA
Test ICT Une machine est destinée au test des cartes électroniques, elle consiste à tester, individuellement, les uns après les autres, les différents composants que comporte la carte. L'accès aux composants se fait via un lit à clous. Elle est divisée en deux fonctionnalités l'un pour tester les circuits ouverts, les court-circuités, de vérifier que les bons composants sont aux bonnes places, et l'autre pour programmer le microcontrôleur pour la carte contrôlé, c'est-à-dire de changer le programme dans la mémoire de la carte électronique.

8- Control Manuel

Observation Manuel

Tester la carte manuellement.

9- Vernissage



On charge la carte manuellement, une machine de vernissage sélectif automatique. Elle dépose de manière reproductible une couche de vernis homogène à des endroits bien définis de la carte.

10- Séchage		<p>Un four à tapis multizones qui permet de sécher rapidement le vernis qu'on a appliqué sur les PCB.</p>
11- Encapsulation de la carte		
12-Test fonctionnel		<p>La machine FKT met la carte dans un environnement pareil de la voiture. Elle applique sur la carte un ensemble des tests différents qui doit passer correctement avant les clients.</p>

Tableau 2 : Processus de fabrication

1.4 Description du projet et méthodologie de travail

1.4.1 Problématique

Pour « Lear Corporation Rabat », l'étape du test est trop nécessaire pour vérifier la qualité et le bon fonctionnement des cartes électroniques. Ce projet consiste à développer une nouvelle application très fiable afin d'accélérer la production d'une façon efficace et précise.

Chez Lear corporation rabat Technopolis, la machine FKT déjà présentée pour tester le fonctionnement des cartes, selon le schéma suivant :

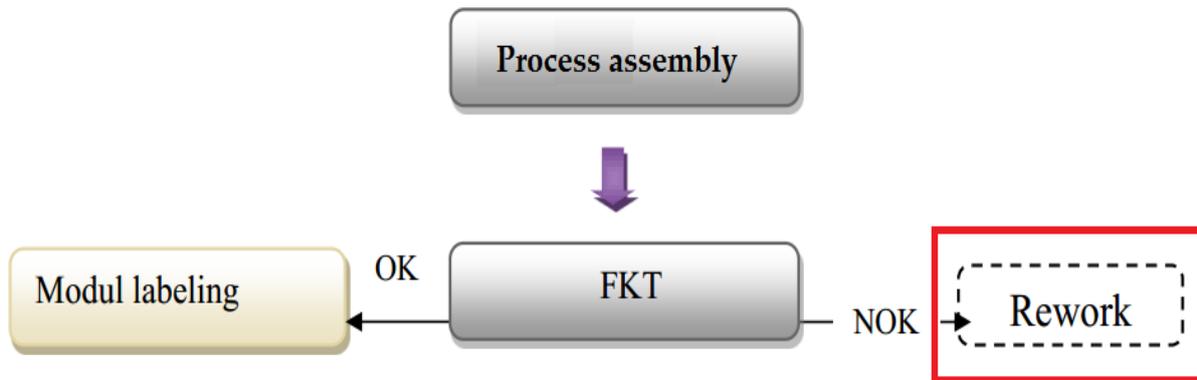


Figure 6 : Schéma de processus assemble

Dans le **Rework** les techniciens ont testé le fonctionnement des cartes d'une manière plus détaillée de A à Z (via une application sur PC), celle-ci est conventionnellement a besoin d'un traitement manuelle pour détecter le problème. Cependant, notre projet consiste à mettre en place une application automatique pour l'analyse à l'aide d'une interface graphique contenant plus d'options, afin de faciliter le test et d'accélérer la production.

1.4.2 Cahier des charges

L'objet majeur de ce stage est de développer une application avec une interface graphique pour l'analyse et le diagnostic. Cette application doit,

- assurer la communication avec la carte via le BUS CAN,
- donner le feed-back des résultats par rapport à une spécification connue,
- faire la mesure en utilisant des méthodes statistiques,
- et établir le rapport de diagnostic.

1.4.3 Description du Projet

Le PL7 est une ligne de production de carte électronique pour la marque BMW, ce produit passe par plusieurs étapes d'assemblage, avant d'arriver au test électrique qui se devise en deux parties :

- Le test de continuité
- Le test fonctionnel

Une fois le test est complet, la carte continue son chemin pour être envoyée au client final. Mais comme dans tous les processus d'assemblage et de test, cette ligne génère sa part de non-conformité, qui devrait être diagnostiqué et analysé pour déduire les causes racine, de ce fait la nécessité de développer une application avec interface graphique pour l'analyse et le diagnostic.

1.4.4 Plan d'action

Pour établir notre plan d'action, nous avons utilisé le logiciel Gantt Project :

-Les taches :

Nous spécifions et distinguons les différentes taches à réaliser dans une durée bien déterminée. Ainsi chacun de nous a consacré une étape et la réaliser tout seul pour l'un complète l'autre et ça montre un esprit d'équipe d'un succès projet bien fait.

Ci-dessous la capture des différentes taches à faire :

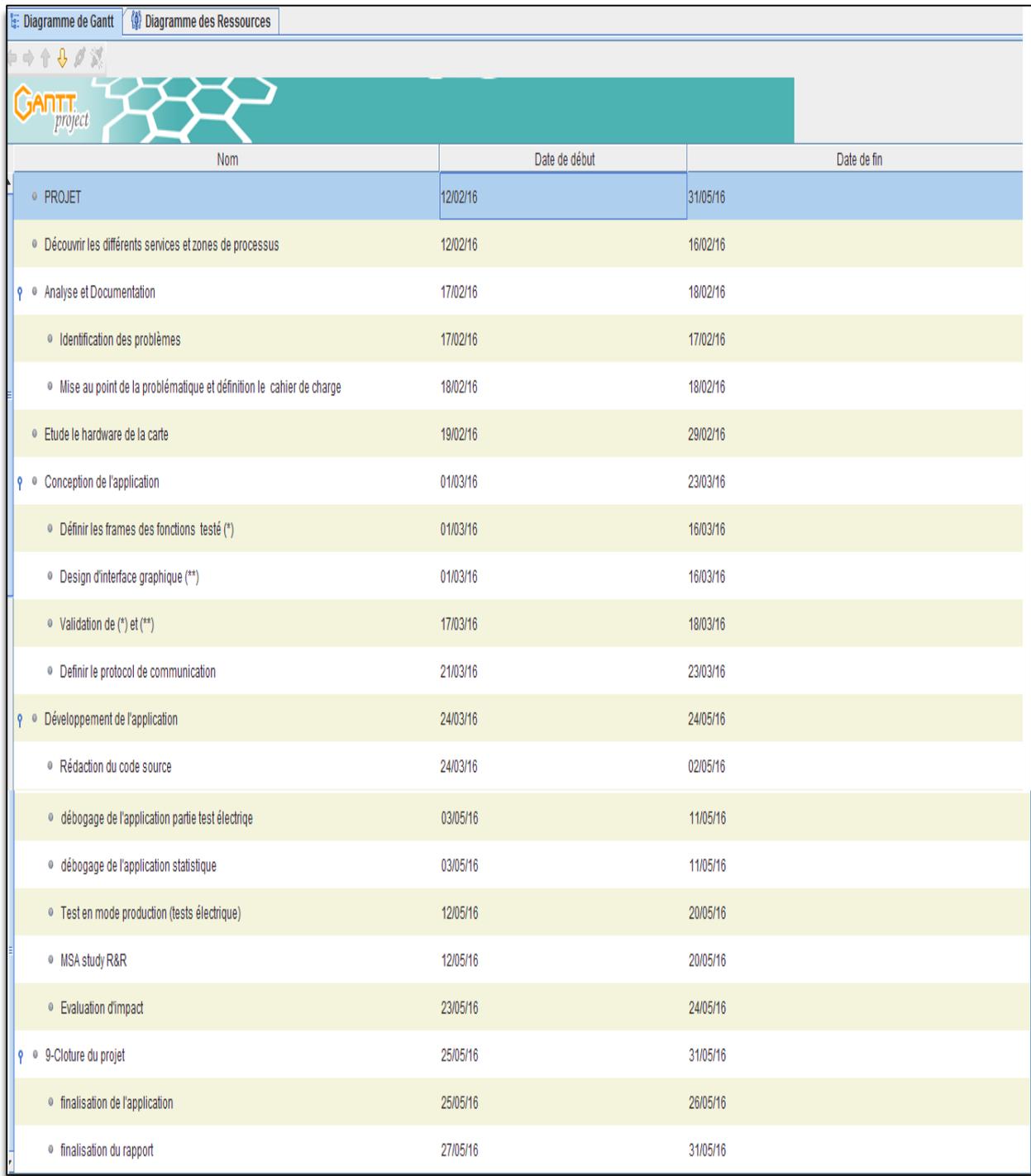


Figure 7 : Les tâches du projet

-le diagramme de Gantt :

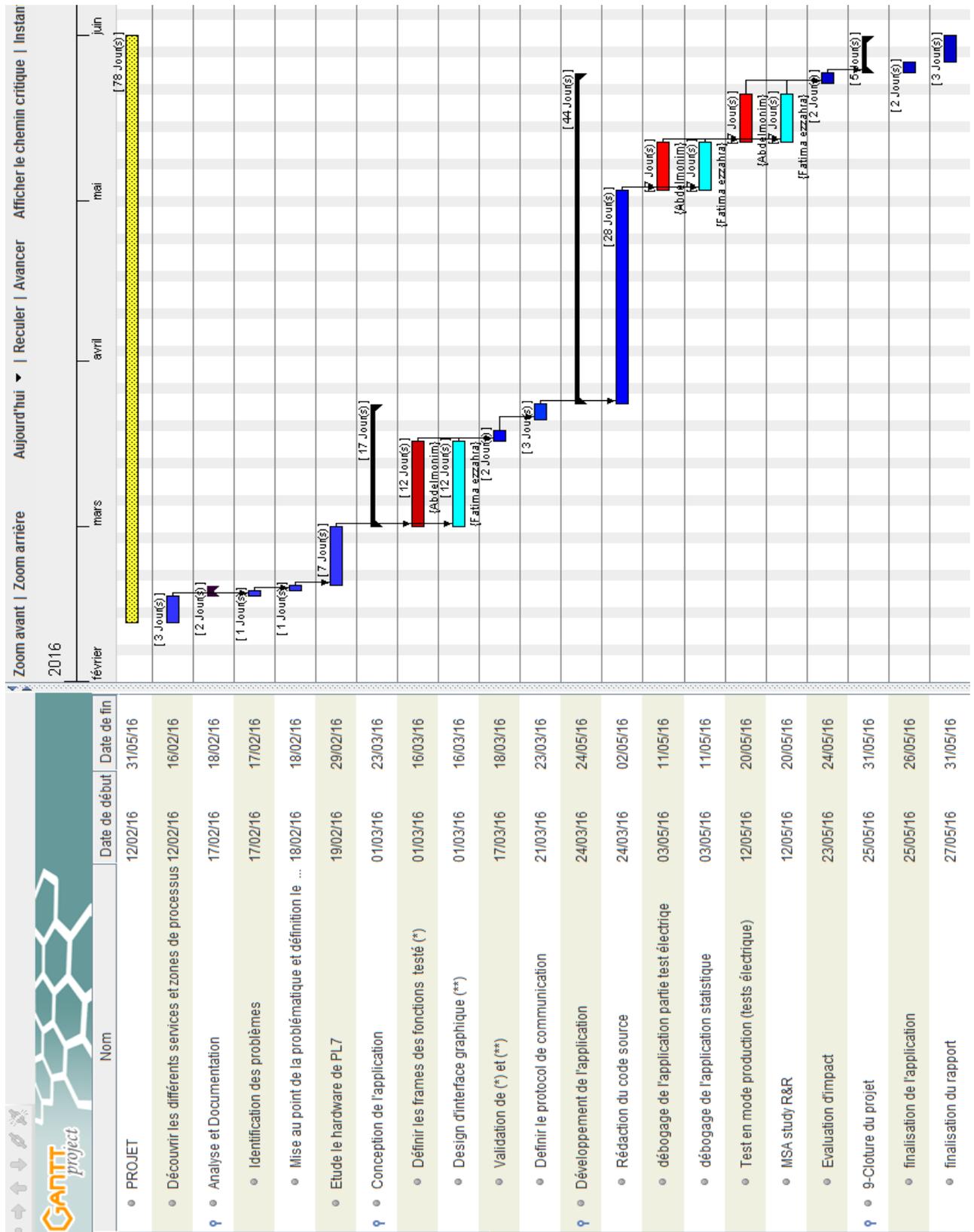


Figure 8 : Diagramme de gant

1.4.5 Méthodologie de travail

La réalisation de l'application doit se réaliser au travers d'une démarche de conception peut se représenter par un cycle en V :

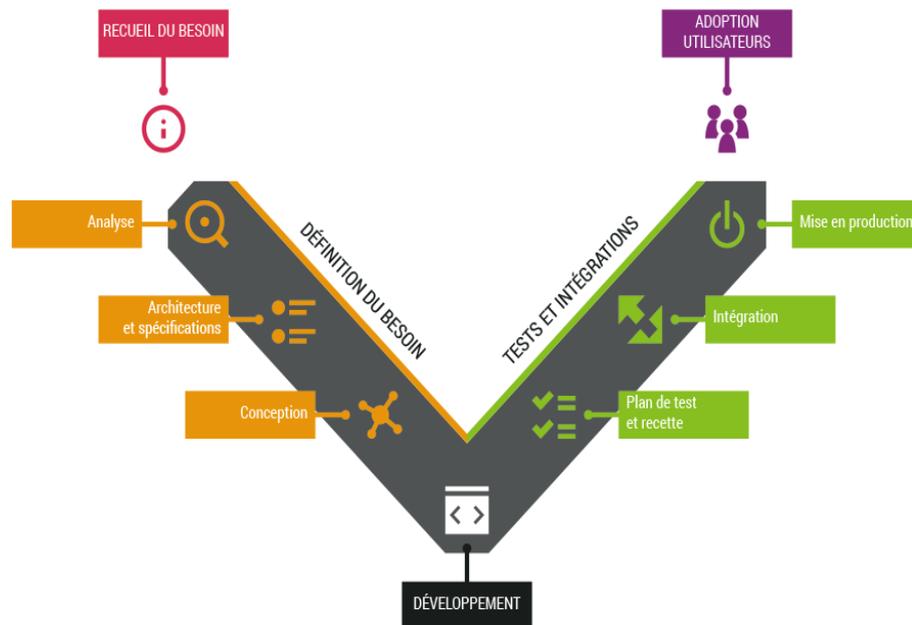


Figure 9 : Démarche du projet par Cycle en V

- **Processus design**

La démarche proposée est composée de cinq étapes :

- Analyse des besoins
- Conception
- Réalisation
- Test et Vérification
- Mise en production

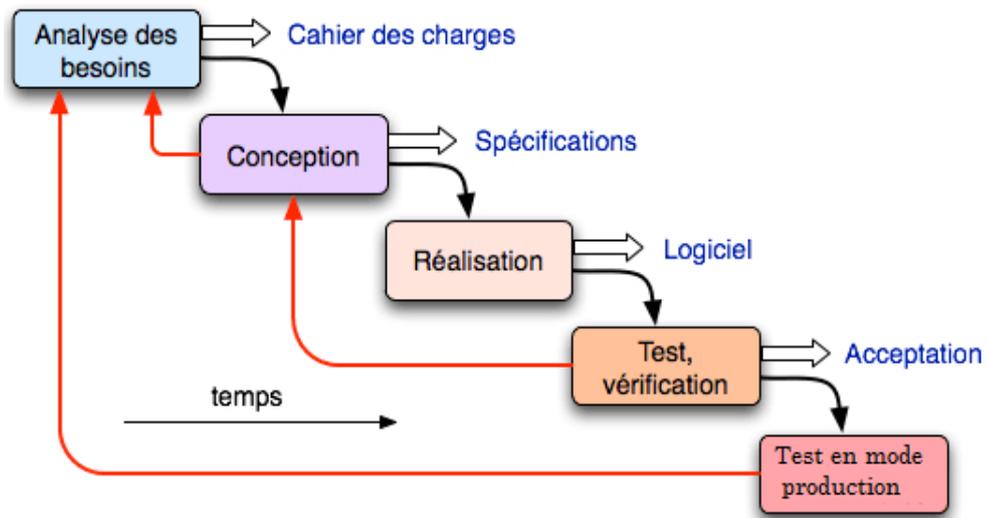


Figure 10 : Processus design

Conclusion

Dans ce chapitre, nous avons présenté la société et nous avons introduit notre projet d'une façon générale. Nous présentons dans le chapitre suivant les spécifications techniques et fonctionnelles de la carte PL7 (FEM).

Chapitre 2 : Spécifications fonctionnelles et techniques

Nous commençons par la phase de spécifications techniques et fonctionnelles. C'est la première étape du processus de développement qui permet de décrire le hardware de la carte PL7 et sa fonctionnalité, puis nous faisons une étude technique spécifique sur la communication de la carte avec l'environnement extérieur.

2.1 Importance de test

La technologie avancée des voitures modernes a une grande différence avec celle des modèles du passé. En effet, dans de nombreux domaines, les systèmes mécaniques conventionnels ont été remplacés par des circuits électroniques complexes. Cela a permis de réaliser de nombreux progrès en termes d'efficacité opérationnelle et d'apporter de nouvelles fonctionnalités dont nous bénéficions tous. Face à l'intégration davantage d'électronique embarquée, les ingénieurs de test doivent faire face à de nouveaux défis.

Sous bien des aspects, les voitures modernes comme BMW sont plus proches des ordinateurs que les précédents véhicules. Dans certains cas, elles doivent traiter plus de 25 Go de données par heure pour exécuter les tâches qui leur sont demandées. La Figure 1 illustre l'évolution du pourcentage du prix des circuits électroniques par rapport au coût total de la voiture au cours des décennies, et cette tendance va se poursuivre. [3]

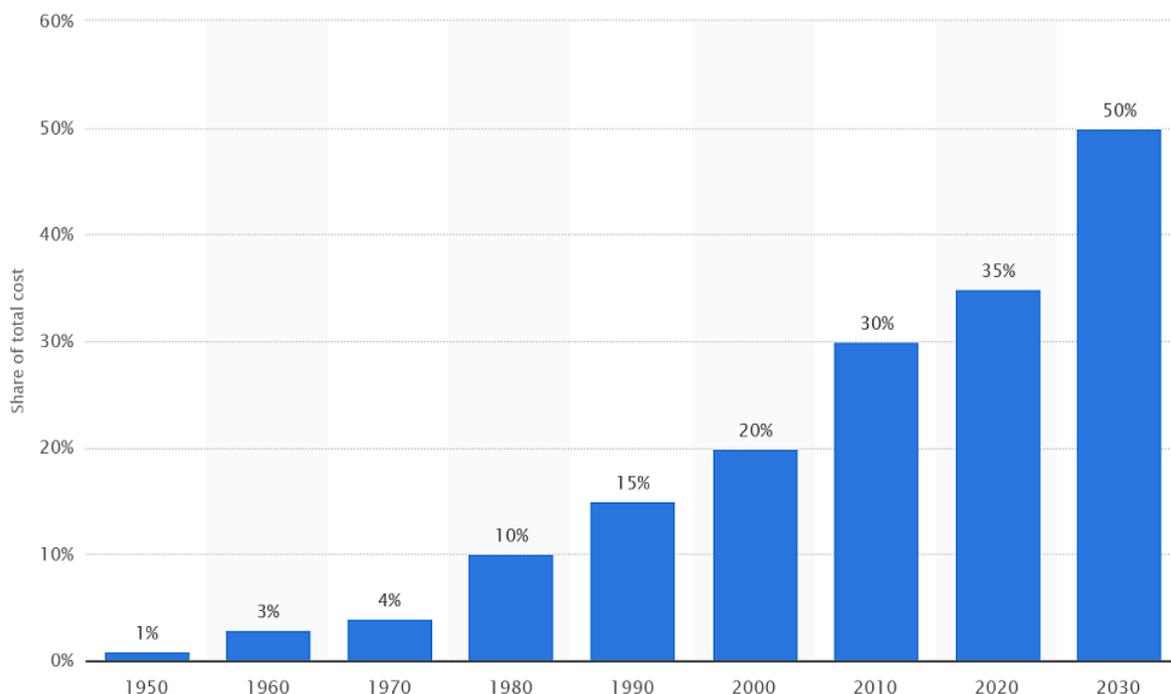


Figure 11 : Coût de l'électronique embarquée exprimé sous forme de pourcentage par rapport au coût total des automobiles dans le monde de 1950 à 2030

Les processus de test sont majoritairement utilisés en production et en maintenance. Le marché demande des solutions toujours plus efficaces aux meilleurs coûts, capables de s'adapter à des cadences de production en constante progression.

2.2 La voiture cible BMW I8

La BMW i8 est une voiture de sport issue du concept car hybride Vision Efficient Dynamics. Présentée lors du salon de l'automobile de Francfort 2009, elle concilie technologie hybride et performances de super car classique.[4]



La nouvelle marque se caractérise comme suite :

-Motorisation : Combine un moteur électrique, une batterie et un moteur thermique.

1. Les roues arrières sont entraînées par un moteur trois cylindres essence TwinPower Turbo de 1,5 L et développant 170 kW (231 ch) pour un couple maximal de 320 Nm. La puissance est transmise aux roues à l'aide d'une boîte automatique à six rapports.
2. Moteur électrique d'une puissance de 96 kW (131 ch) et d'un couple maximal de 250 Nm entraîne les roues arrière via une boîte automatique à deux rapports
3. La combinaison des deux moteurs via une gestion intelligente des flux d'énergie donnent une puissance cumulée de 266 kW (362 ch).
4. Accélération de 0 à 100 km/h en moins de 4,4 secondes pour une vitesse maximale de 120 km/h, au-delà, le moteur essence vient l'épauler il est possible de rouler jusqu'à 250 km/h

-Consommation Carburant et Autonomie Batterie :

1. Consommation de Carburant : selon le cycle de conduite européen, la i8 affiche une consommation record de 2,5 litres/100 km.
2. Batterie : lithium-ion haute tension à refroidissement liquide d'une capacité énergétique utile de 5 kWh. Elle autorise une autonomie en mo de tout électrique pouvant aller jusqu'à 35km.
3. Se recharge en moins de 3 heures tandis que sur le boîtier mural BMW i), une recharge complète prend moins de 2 heures.

-Lear Rabat a contribué à la production de la BMW i8 en fournissant des terminaux, des connecteurs, et des cartes électroniques telle que le « PL7 card » et le S-BOX « switch Box » dont actuellement Lear Rabat est le seul fournisseur au niveau mondial.

- PL7 (Platform 7) : Permet la gestion des systèmes électroniques de la voiture. Il s'agit d'un smart Junction box destiné à BMW. Le PL7 représente des modèles d'intégration d'une haute technologie, en utilisant moins de composants en comparaison avec le PL3 afin de réduire les dimensions de la Junction Box, le cout et le délai d'assemblage.
- S-Box ou Switch Box: C'est une boîte de commutation des relais de forte puissance pour les Batteries haute tension, équipé d'un circuit de commande pour des mesures de précisions.

2.3 Fonctionnalité de la carte PL7 (Platform 7 Front electronic Module)

Lear rabat Technopolis livre différents équipements d'automobile, plusieurs lignes de productions sont placées, par lesquelles passent les différents produits : PL3, PL7, MQB, X10 et TSMC/JLR. [5]

Le PL7 fait un processus d'assemblage des différents composants et faire des tests nécessaires seulement pour la voiture BMW I8 de telle sorte à diminuer le coût et le délai d'assemblage.

FEM (Front Electronic Module) est un module de commande électronique permet de gérer des systèmes électroniques dans la voiture, on trouve ci-dessous la plupart des fonctionnalités de la carte sous forme des symboles et leur signification :

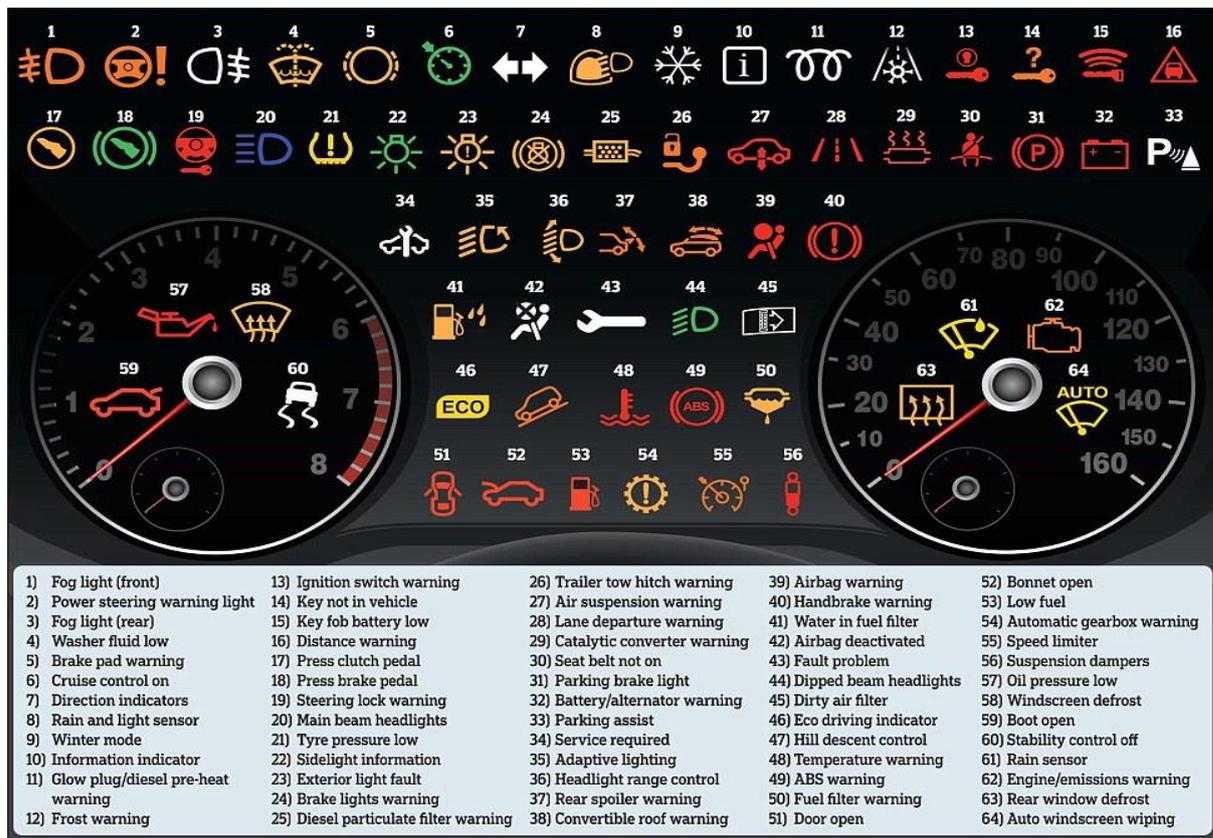


Figure 12 : Les symboles du Module FEM

-> Dans notre projet, l'application doit afficher le résultat des certaines fonctions sous forme des symboles, afin de permettre à l'utilisateur d'avoir une idée sur le test.

2.4 Hardware de la carte électronique PL7

La ligne PL7 permet de distribuer une carte électronique pour la voiture BMW I8, cette carte comme toutes les cartes contient de nombreux composants électroniques, on cite parmi eux :



Figure 13 : Les composants électroniques

Ces composants permettent de gérer le système électronique dans la voiture, nous nous intéressons à l'étude des trois microcontrôleurs existant dans la carte. Ces derniers sont de la famille Freescale [6]. On trouve deux types : SPC56XX 32-bit «body et Gateway ». Il existe un troisième type : « ECL » de type MC9S08SG32 8-bit :

- Body : Le microcontrôleur central de la carte responsable de la plupart des fonctions « SPC5646 ».
- Gateway : Microcontrôleur responsable de la gestion de l'ensemble des protocoles de communication « SPC5668 ».
- ECL : C'est le Co body Microcontrôleur contrôlé par le Body.

2.4.1 Body (Le microcontrôleur SPC5646C)

Le microcontrôleur MPC5646C est un module On-chip de la famille freescale disposant les caractéristiques suivantes :

- 32-bit core Power Architecture,
- Allant jusqu'à 120 MHz,
- Contient unité des virgules flottantes (FPU),
- Supporte Nexus 3+.
- 3 MB de mémoire Flash on-chip,
- 256 KB SRAM on-chip,
- Deux convertisseurs analogique-numérique 12 bits (ADC),

- 199 (BGA) pins (I/O),
- 8 modules interfaces séries (DSPI),
- 10 modules de communication série (LINFlex),
- 1 contrôleur FlexRay double canal,
- 1 Contrôleur Fast Ethernet (FEC),
- Régulateur tension on-chip (VREG).

La figure suivante présente l'architecture du microcontrôleur :

MPC564xB/C Block Diagram

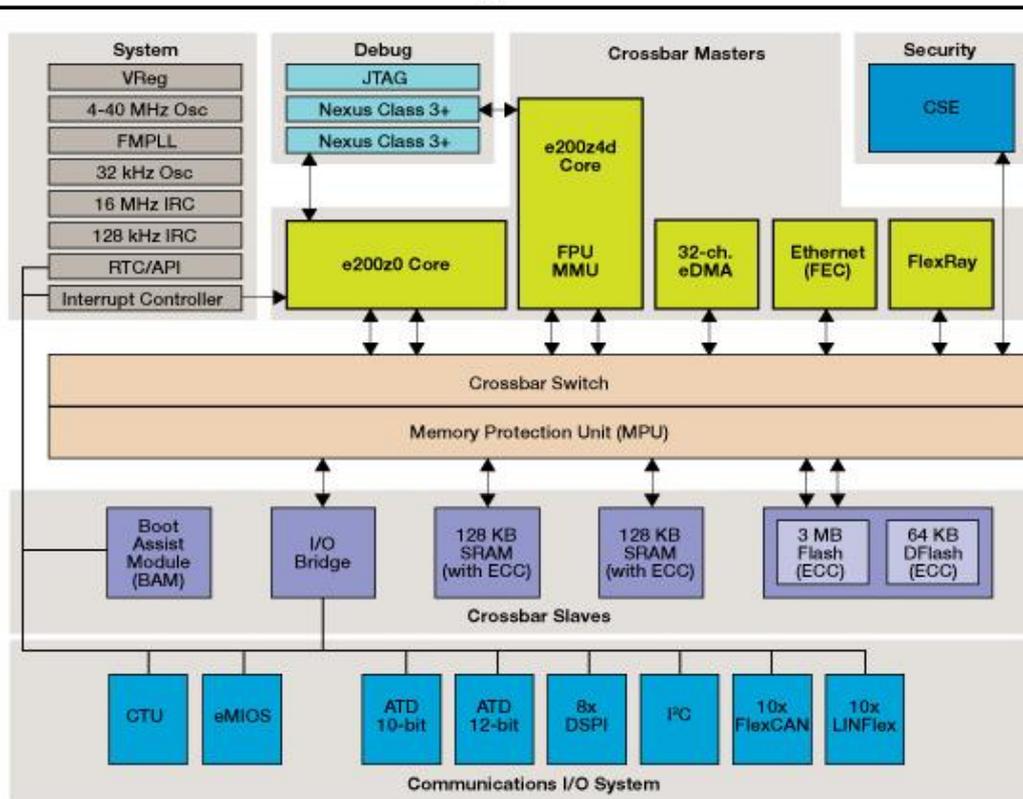


Figure 14 : Microcontrôleur MPC564

BODY est un microcontrôleur de type SPC5646C considéré comme le cœur de la carte BDC, il rend possible la gestion de la plupart des périphériques de la carte. La famille des microcontrôleurs MPC564xB - C 32-bit est conçue pour la prochaine génération des modules de commande de carrosserie automobile haut de gamme (BCM) et des applications passerelles. En outre, il offre un niveau supérieur d'intégration des périphériques afin de satisfaire le besoin croissant, l'amélioration des ensembles de fonctionnalités et aussi offrir plus d'espace mémoire.

Du point de vue sécurité, la SPC564xB – C comporte un moteur cryptographique des services innovants (CSE). Le CSE est un ensemble de fonctionnalités matérielles de chiffrement qui permettent la transmission des informations de manière sécurisée et fiable. La famille SPC564xB - C dispose également d'un dual-core Power Architecture R avec une capacité de traitement de près de 300 DMIPS, il peut fonctionner en mode veille pour aider à une consommation d'énergie réduite. Il a aussi une large gamme des périphériques de communication pour interfacer l'ensemble des sous-systèmes avec lesquels un BCM communique.

2.4.2 Gateway (Le microcontrôleur MPC5668G))

Le microcontrôleur MPC5668G est un module On-chip de la famille freescale disposant des caractéristiques suivantes :

- Conforme à la catégorie Power Architecture CPU (e200z650) 32 bits,
 - 32 Ko mémoire cache,
 - La vitesse d'exécution statique : 116 MHz.
- 2MB en mémoire flash on-chip,
- 592 KB de mémoire ECC SRAM on-chip,
- 16 canaux des Contrôleurs d'accès direct à la mémoire,
- Fast Ethernet controller : support 10-Mbps et 100-Mbps,
- Contrôleur d'interruption (INTC) prend en charge 316 vecteurs d'interruption externes (22 sont réservés),
- Convertisseur analogique-numérique (ADC),
- Interface communication (I2C),
- Module d'interface communication Serial (ESCI),
- Module CAN (FlexCAN) Version 2.0B ,
- Contrôleur FlexRay Dual-channel,
- Contrôleur JTAG.

La figure suivante présente l'architecture du microcontrôleur :

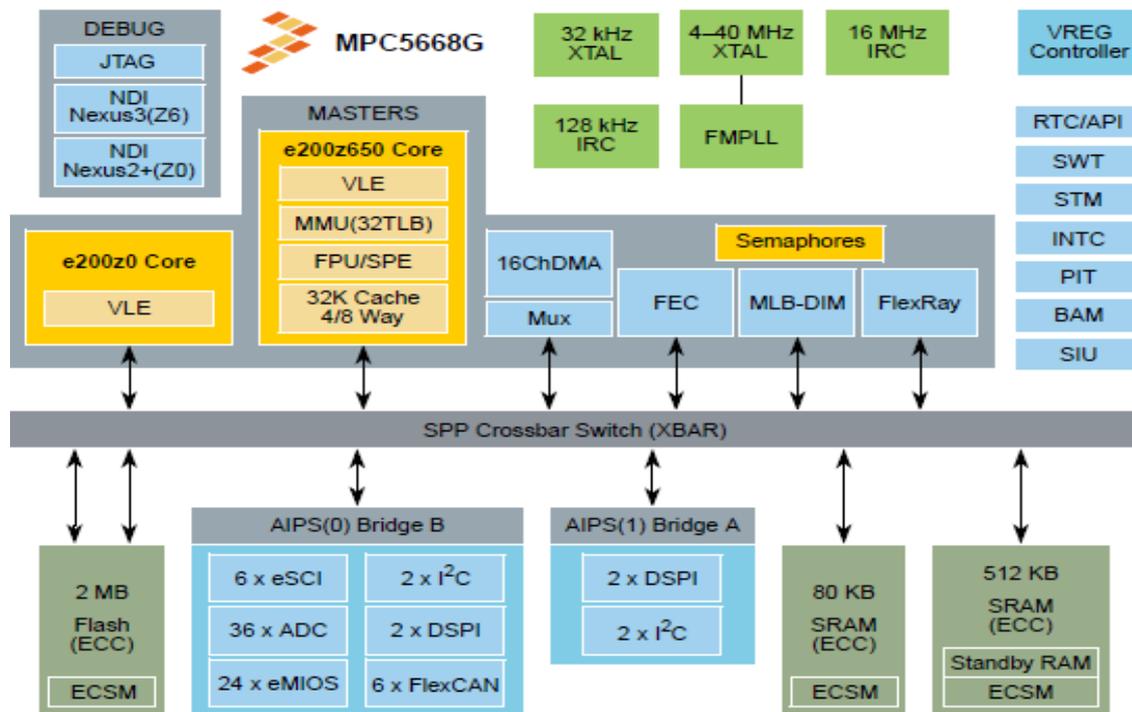


Figure 15 : Block diagramme MPC5668G

Le Microcontrôleur MPC5668G 32-bit de la technologie Power Architecture R dual-core, concentre à un seul microcontrôleur tous les protocoles de communication possibles qu'on peut trouver dans une voiture. L'architecture dual-core fournit le débit et la performance dont on a besoin pour maintenir en temps réel le fonctionnement prévu. Le Microcontrôleur MPC5668G est un Gateway responsable de la gestion de tous les types et les protocoles de communication utilisés dans le domaine automobile tels que CAN, Ethernet, Flexray permettant l'échange d'information entre le module électronique et l'environnement extérieur.

2.4.3 ECL : Le microcontrôleur MC9S08SG32

Le microcontrôleur MC est un module On-chip de la famille freescale dispose des caractéristiques suivantes :

- 8-Bit HCS08 Central Processor Unit (CPU) • 40-MHz HCS08 CPU,
- Prise en charge de 32 sources d'interruption,
- Oscillateur : cristal ou céramique varie de 31,25 kHz à 38,4 kHz ou 1 MHz à 16 MHz,

- Convertisseurs analogique-numérique (ADC) 10-bit,
- Contrôleur SPI. Environnement extérieur,
- Protocole I2C,
- Contrôleur SCI.

La figure suivante présente l'architecture du microcontrôleur :

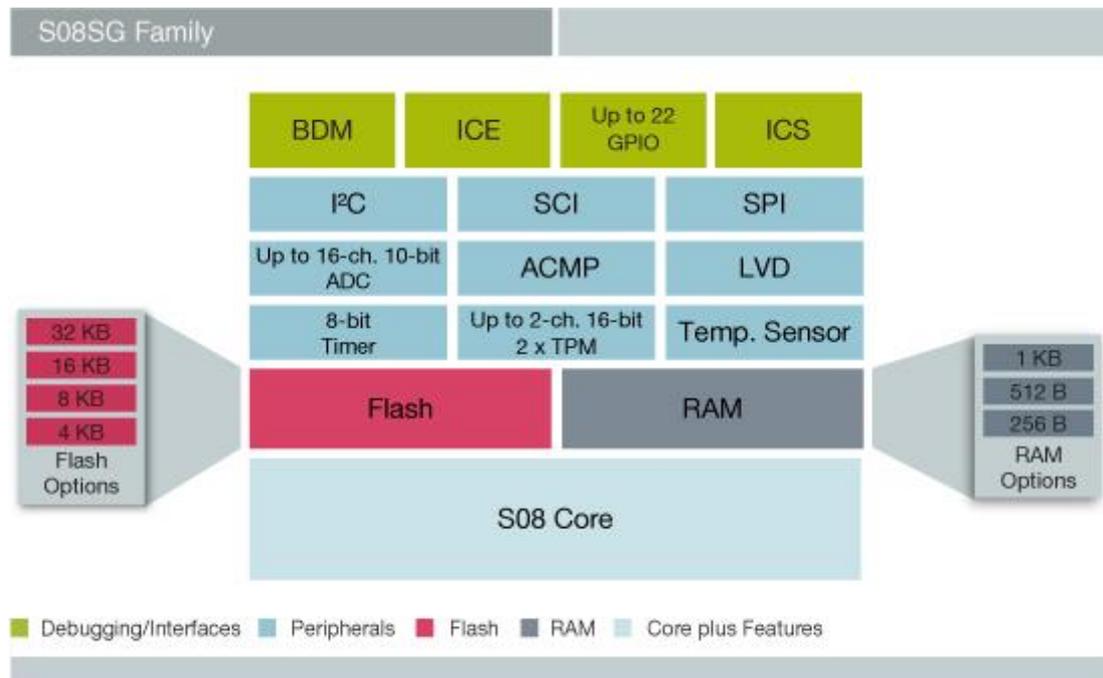


Figure 16 : Block Diagramme MC9S08SG32

Les microcontrôleurs de MC9S08SG32 sont des membres de la famille HCS08 8-bits, faible coût, haute performance. Les dispositifs à haute température de la série MC9S08SG32 ont été qualifiés de fonctionner jusqu'à 150 C. Tous les microcontrôleurs de la famille utilisent le noyau HCS08 amélioré avec une variété de modules, taille de la mémoire, types de mémoire, et les types de paquets.

2.4.4 Les driver

La carte PL7 dispose de plusieurs composantes électroniques qui donnent capacité de la carte BDC à contrôler des périphériques internes de la voiture.

- Les drivers contrôlés par BODY :
 - Mirror/Washer Heater : Contrôle les moteurs des essuie-glaces,
 - Washer Pumps : Contrôle la pompe de jet d'eau,

- LED Outputs : Contrôle les ampoules d'éclairages,
 - LED Outputs Aux : Contrôle les ampoules d'éclairages auxiliaire,
 - AIRBAG : Contrôle le système de sécurité « airbag »,
 - Front Wiper (speed I/I I) : Contrôle la vitesse des essuie-glaces avant,
 - Rear Wiper I : Contrôle la vitesse des essuie-glaces arrière,
 - Capteur de carburant : Permet de savoir le niveau de carburant dans le réservoir de la voiture.
- Les drivers contrôlés par Gateway ZGW :
- CAN : Protocole de communication CAN,
 - LIN : Protocole de communication LIN,
 - Flexray : Protocole de communication Flexray,
 - Ethernet : Protocole de réseau local à commutation de paquets.
- Les drivers contrôlés par ECL :
- ECL Supply,
 - CAS. FailSave Hardware.

2.4.5 Diagramme de la carte PL7

Le diagramme suivant présente un schéma simplifié de la liaison entre trois microcontrôleurs et des autres fonctions, ce diagramme permet de donner une idée générale sur l'architecture de la carte PL7.

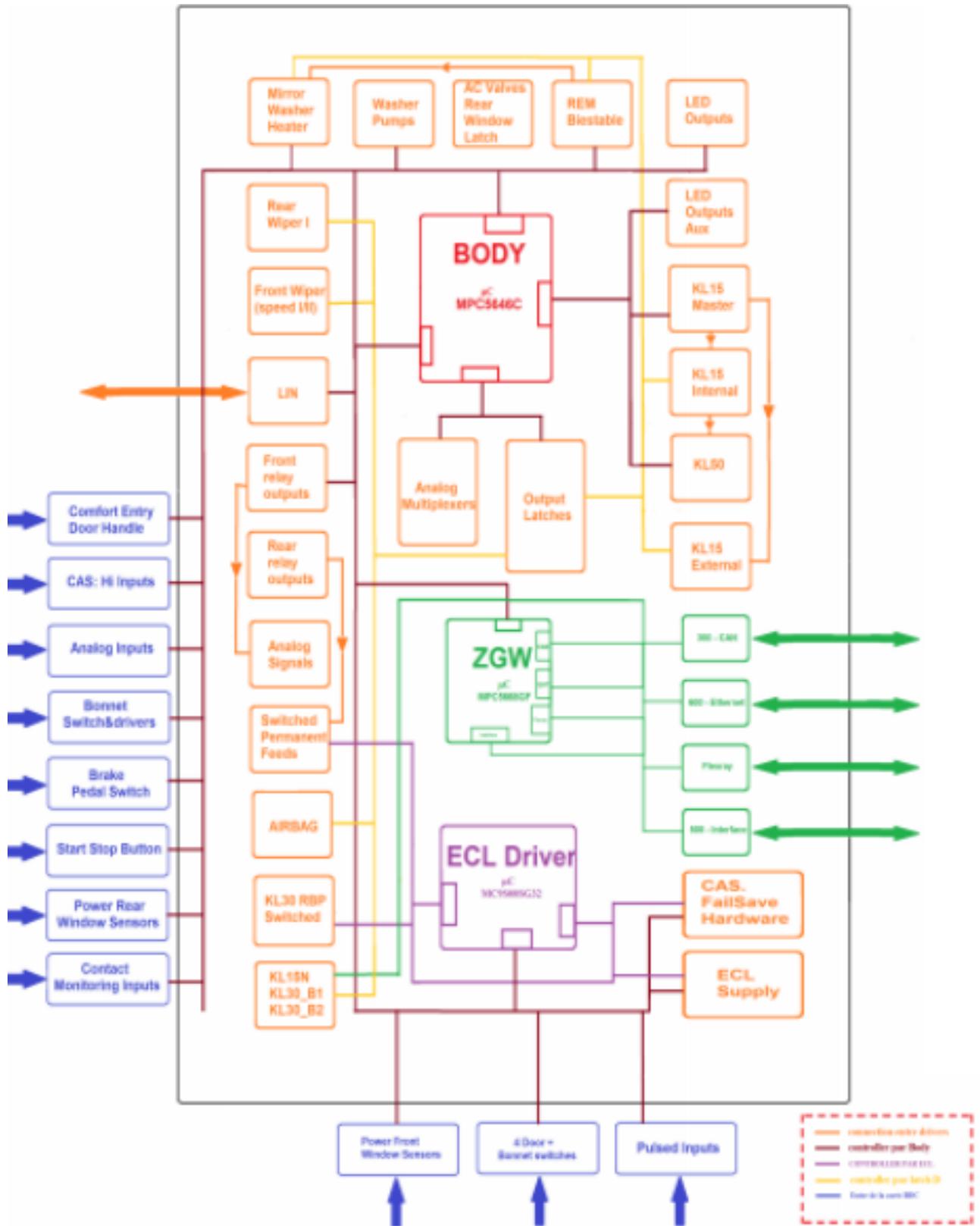
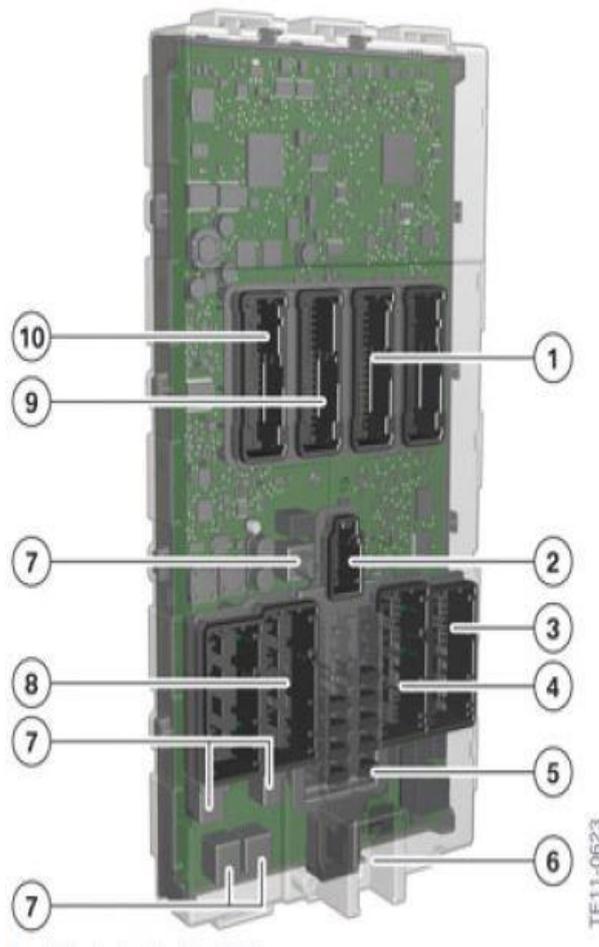


Figure 17 : Diagramme de la carte FEM

2.4.6 Les connecteurs

La carte PL7 dispose des 10 connecteurs pour communiquer avec l'environnement extérieur comme le montre le schéma suivant :



Index	Explanation
1	Connector (54-pin)
2	Connector (12-pin)
3	Connector (42-pin)
4	Connector (54-pin)
5	Fuses
6	Connector (1-pin, B+ supply)
7	Relay
8	Connector (54-pin)
9	Connector (54-pin)
10	Connector (54-pin)

Figure 18 : Les connecteur de la carte PL7

Au niveau de notre projet, nous aurons besoin d'effectuer la communication en utilisant le protocole CAN. Pour ce faire, nous utilisons le quatrième connecteur pour l'alimentation et le neuvième pour le transfert et la réception des données et les autres sont liés à des charges fictives pour simuler les tests fonctionnels.

2.5 Mise en œuvre la communication avec la Carte

Les tests qui s'effectuent sur la carte PL7 FEM nécessitent d'établir avant tout une communication avec les deux microcontrôleurs afin de recevoir les données liées au produit pour l'analyse et le diagnostic. Pour gérer cette communication, dans notre projet nous utilisons le protocole CAN pour envoyer les frames et recevoir le feedback et des résultats des fonctions.

2.5.1 Description de BUS CAN

Le bus CAN (Controller Area Network) est né du besoin de trouver une solution de communication série dans les véhicules automobiles, qui ont tendance à intégrer de plus en plus de commandes électroniques. Jusqu'à maintenant, tous les organes de commandes des véhicules échangeaient les données par l'intermédiaire de lignes dédiées. L'augmentation du nombre d'organe embarqué a contraint les équipementiers automobiles à développer une nouvelle architecture à base de bus réseaux. Le GIE Renault-PSA avec les partenaires comme Sagem, Valeo et autres ont développé le bus VAN (Véhicule Area Network), les puces contrôleurs sont fabriquées par MHS, SGS, TI En Allemagne, Bosch a développé, au milieu des années 80, le bus CAN ou "Controller Area Network" qui a fait l'objet d'une normalisation ISO 11898.

Avec le bus CAN, les stations ayant les mêmes droits (organes de commande, capteurs ou actionneurs) sont reliées par un bus série. Le protocole CAN de base leur permet d'échanger 2048 variables. Ce protocole, ainsi que les paramètres électriques de la ligne de transmission, sont fixés par la norme 11898. La transmission physique s'effectue soit une paire torsadée ou par liaison infrarouge, hertzienne ou par fibre optique [7]

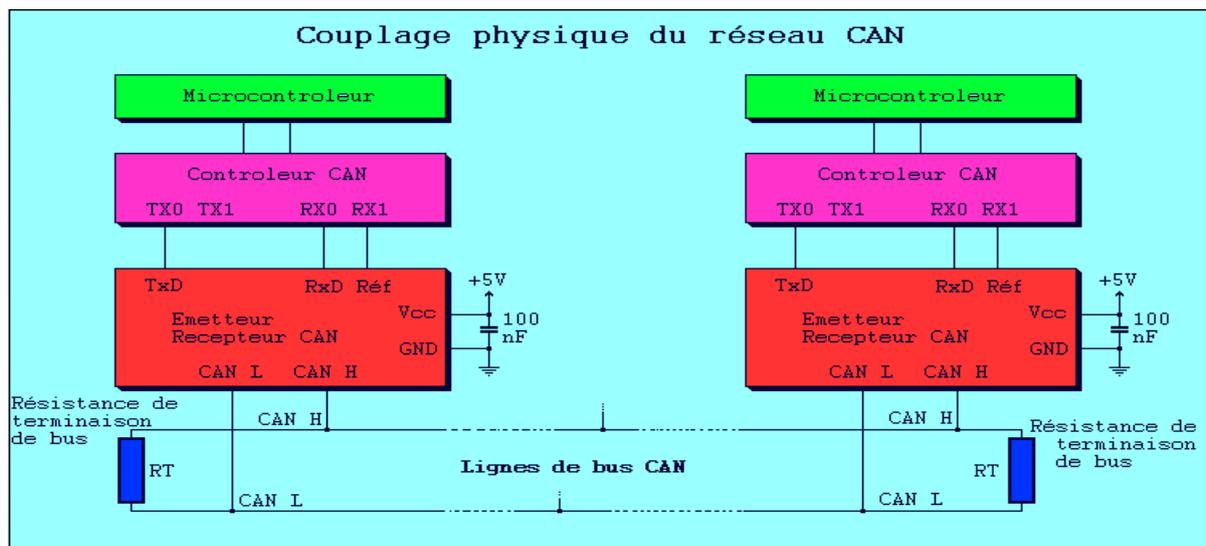


Figure 19 : Couplage physique du réseau CAN

2.5.2 Le choix de BUS CAN

Le bus CAN est une application d'une approche connue sous le nom de multiplexage, et qui consiste à raccorder à un même câble (un bus) un grand nombre de calculateurs qui communiqueront donc à tour de rôle. Cette technique élimine le besoin de câbler des lignes dédiées pour chaque information à faire transiter (connexion point-à-point).

Il s'agit d'un bus de terrain. Il sert à relier principalement des équipements du type capteurs, actionneurs etc. Il est principalement utilisé dans des milieux ayant des fortes contraintes. Plus spécifiquement, nous le trouvons sur les automobiles.

2.5.3 Le protocole de communication CAN

2.5.3.1 Principe de fonctionnement

Du type multi-maître, orienté messages courts, le bus CAN est bien adapté à la scrutation de variables émises par des stations déportées. La norme ISO 11898 spécifie un débit maximum de 1Mbit/s. La longueur maximum du bus est déterminée par la charge capacitive et le débit. Les configurations recommandées sont les suivantes :

Débit	Longueur
1 Mbit/s	40 m
500 Kbit/s	100 m
100 Kbit/s	500 m
20 Kbit/s	1000 m

Figure 20 : Débit

Le protocole est basé sur le principe de diffusion générale : lors de transmission, aucune station n'est adressée en particulier, mais le contenu de chaque message est explicité par une identification reçu de façon univoque par tous les abonnés. Grâce à cet identificateur, les stations, qui sont en permanence à l'écoute du réseau, reconnaissent et traitent les messages qui les concernent; elles ignorent simplement les autres.

L'identificateur indique aussi la priorité du message, qui détermine l'assignation du bus lorsque

plusieurs stations émettrices sont en concurrence. En version de base, c'est un nombre de 11 bits, ce qui permet de définir jusqu'à 2048 messages plus ou moins prioritaires sur le réseau. Chaque message peut contenir jusqu'à 8 octets de données, ce qui correspond par exemple à l'état de 64 capteurs. L'adressage par le contenu assure une grande flexibilité de configuration. Il est possible d'ajouter des stations réceptrices à un réseau CAN sans modifier la configuration des autres stations.

Domaines d'application :

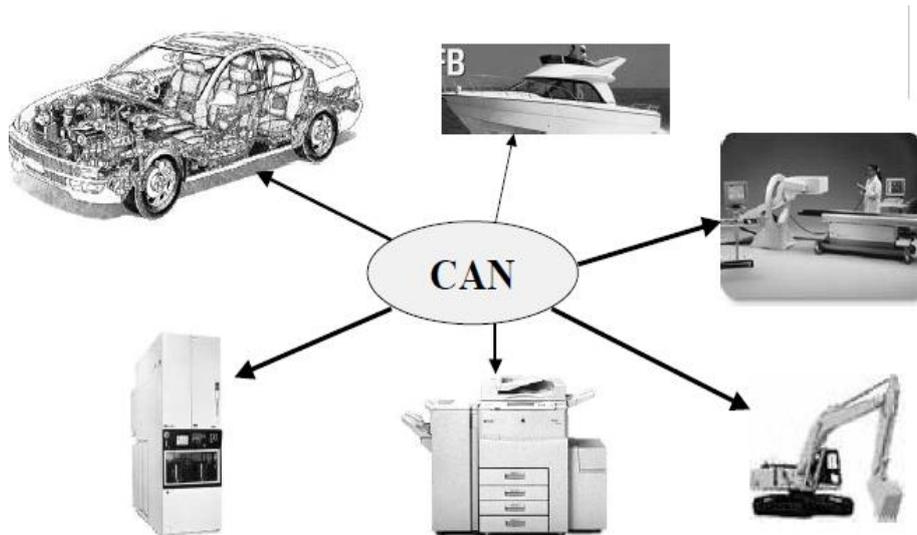


Figure 21 : Domaines d'application

Evolution du câblage dans l'automobile grâce à un bus de CAN :

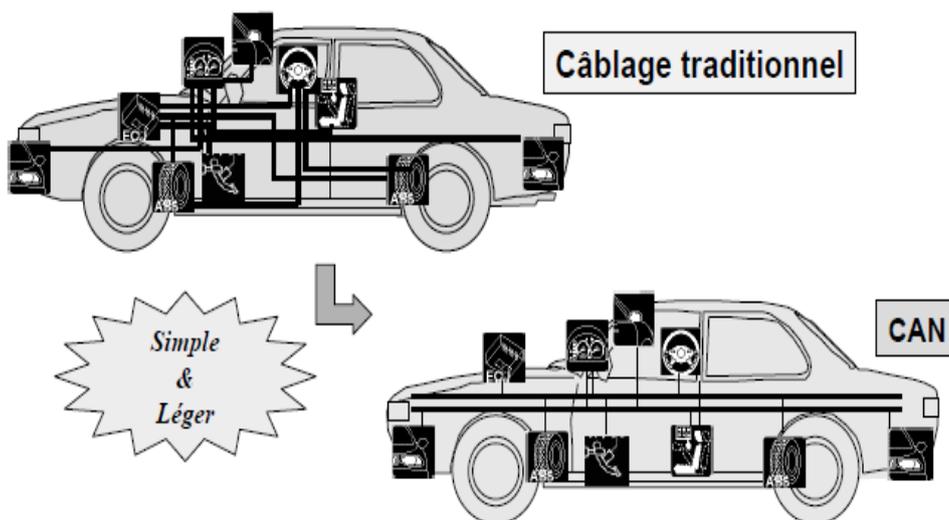


Figure 22 : le bus CAN dans l'automobile

Block de bus CAN protocole :

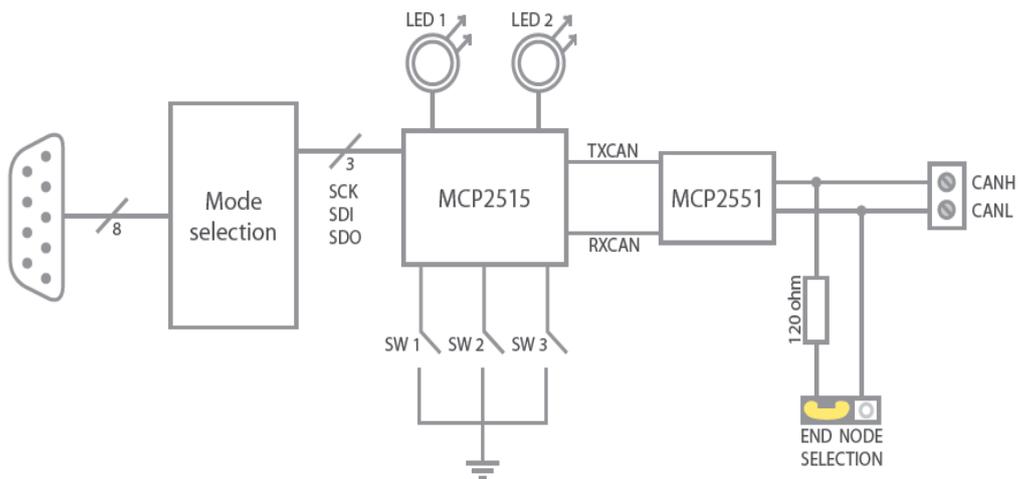
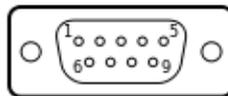


Figure 23 : Block de bus CAN protocole

Connecteurs :



Broche	Description
1	(Réservé)
2	CANL
3	Masse
4	(Réservé)
5	Blindage (optionnel)
6	Masse
7	CANH
8	(Réservé)
9	Alimentation externe (optionnel)

Figure 24 : Les connecteurs de CAN

2.5.3.2 Formats de trames de messages

La norme CAN définit deux formats de protocole : Standard (Version 2.0 A) et Extended (Version 2.0 B). La différence résulte seulement dans la longueur de l'identificateur (ID) qui est de 11 bits de base et 18 bits supplémentaires en mode Extended. Cette extension permet

l'augmentation du nombre de variables échangées, et le nombre de stations sur le réseau. Le nombre d'octets de données échangées à chaque trame reste inchangé.

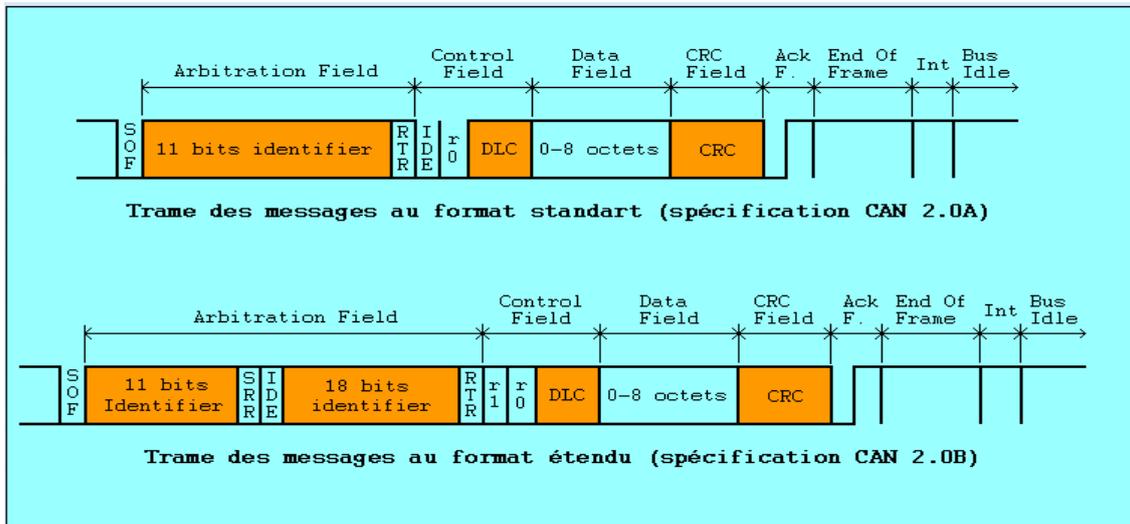


Figure 25 : Formats de trames

Une trame est composée des champs suivants :

- bit SOF (Start Of frame),
- zone d'arbitrage (11 bits),
- bit RTR (Remote Transmission Request) : détermine s'il s'agit d'une trame de données ou d'une trame de demande de message,
- bit IDE qui établit la distinction entre format standart (état dominant) et format étendu (état récessif),
- 1 bit réservé pour une utilisation future,
- 4 bit DLC : nombre d'octets contenus dans la zone de données,
- zone de données de longueur comprise entre 0 et 8 octets,
- zone CRC de 15 bits : (Cyclic Redundancy Code) ces bit sont recalculés à la réception et comparés aux bits reçus. S'il y a une différence, une erreur CRC est déclarée,

-zone ACK composé d'un bit à l'état récessif ainsi qu'un bit séparateur ACK. Le premier bit doit être forcé à l'état dominant par les stations ayant bien reçu cette trame,

- et zone EOF de 7 bits : (End Of Frame) permet d'identifier la fin de la trame.

Stuff bit : au cours de la construction d'une trame, si 5 bits consécutifs portent la même valeur 0 ou 1, un bit de valeur opposée est ajouté à la suite des 5 bits consécutifs.

Les contrôleurs CAN qui admettent le format étendu peuvent aussi émettre et recevoir des messages au format standard. En revanche, dès que l'on utilise sur le réseau des contrôleurs ne maîtrisant que le format standard, les messages étendus seront mal interprétés.

2.5.3.3 Détection des erreurs

La trame du protocole CAN intègre des mécanismes de détection d'erreurs. De plus, comme tous les nœuds de réseau surveillent simultanément le bus, ils détectent des différences entre bits reçus et bits émis. Dès qu'une erreur est détectée, la transmission en cours est interrompue par l'émission d'un indicateur d'erreur ("error flag"). L'émetteur peut donc recommencer à émettre son message.

Tout ce système de gestion des erreurs est complètement transparent pour le développeur et l'utilisateur. Le système est capable de gérer automatiquement ses conflits et ses erreurs en émettant des trames d'erreurs pour renseigner l'émetteur du message sur le type de faute qu'il a commis. Une station est capable de faire la distinction entre les perturbations temporaires et les défauts permanents. Les stations en défaut permanent sont déconnectées automatiquement du réseau.

Ce système de gestion d'erreur fait toute la puissance du réseau CAN, certains constructeurs démontrent que la probabilité d'erreur résiduelle reste inférieure $4.6 \cdot 10^{-11}$.

2.5.4 Le CANcaseXL

CANcaseXL est un équipement de société Vector qui donne la possibilité d'envoi et réception de données en utilisant le protocole CAN, écouter un bus CAN sans l'influencer, détecter et générer des frames d'erreur. En outre il est équipé d'une carte mémoire qui donne la possibilité de stocker les données. Les données sont transférées à un PC par USB. Le CANcaseXL sera utilisé avec notre application afin établir la communication avec la carte.[8]



Figure 26 : Cancase XL

2.5.5 La connexion de la carte PL7

La carte PL7 soit connectée au CANcaseXL à travers le connecteur 9 qui contient le CAN-H et CAN-L. Un connecteur est utilisé pour la transmission des frames et la réception des données, et l'autre connecteur contient les pins d'alimentation des microcontrôleurs comme montré dans la figure ci-dessous :



Figure 27 : Connexion de la carte PL7

Conclusion

Nous avons fait dans ce chapitre une étude technique sur la carte PL7(FEM) dont les microcontrôleurs, le protocole CAN et les connecteurs (partie de hardware). Nous avons également étudié les fonctionnalités de la carte. Nous développons dans le chapitre suivant un algorithme nécessaire à la réalisation de notre projet.

Chapitre 3 : Algorithme et méthode de diagnostic du module PL7 FEM

Le présent chapitre a pour objectif d'illustrer la procédure du diagnostic de la carte PL7 FEM pour établir dans un premier temps la communication CAN et dans un deuxième temps pour effectuer les tests fonctionnels. Une étude software préalable doit être élaborée avant de choisir le bon algorithme permettant d'améliorer la qualité du test du produit.

3.1 Procédure du diagnostic de la carte PL7

3.1.1 Présentation de la station de test

L'étape « End OF Line » est la dernière étape dans le processus de fabrication, qui a pour but d'effectuer un ensemble des tests électriques sur la carte, lorsque ces tests électrique sont réussis, alors dans ce cas , la carte continue son chemin vers le client, cependant dans le cas contraire la carte est renvoyée au Rework pour l'analyse, au moyen des tests manuels afin de détecter la source d'anomalie. Deux cas de figures sont possibles :

- Pièces non réparables, donc rejetées.
- Pièces réparées avec succès, donc renvoyées pour refaire le test automatique.

L'organigramme suivant décrit le processus de travail de la zone Rework :

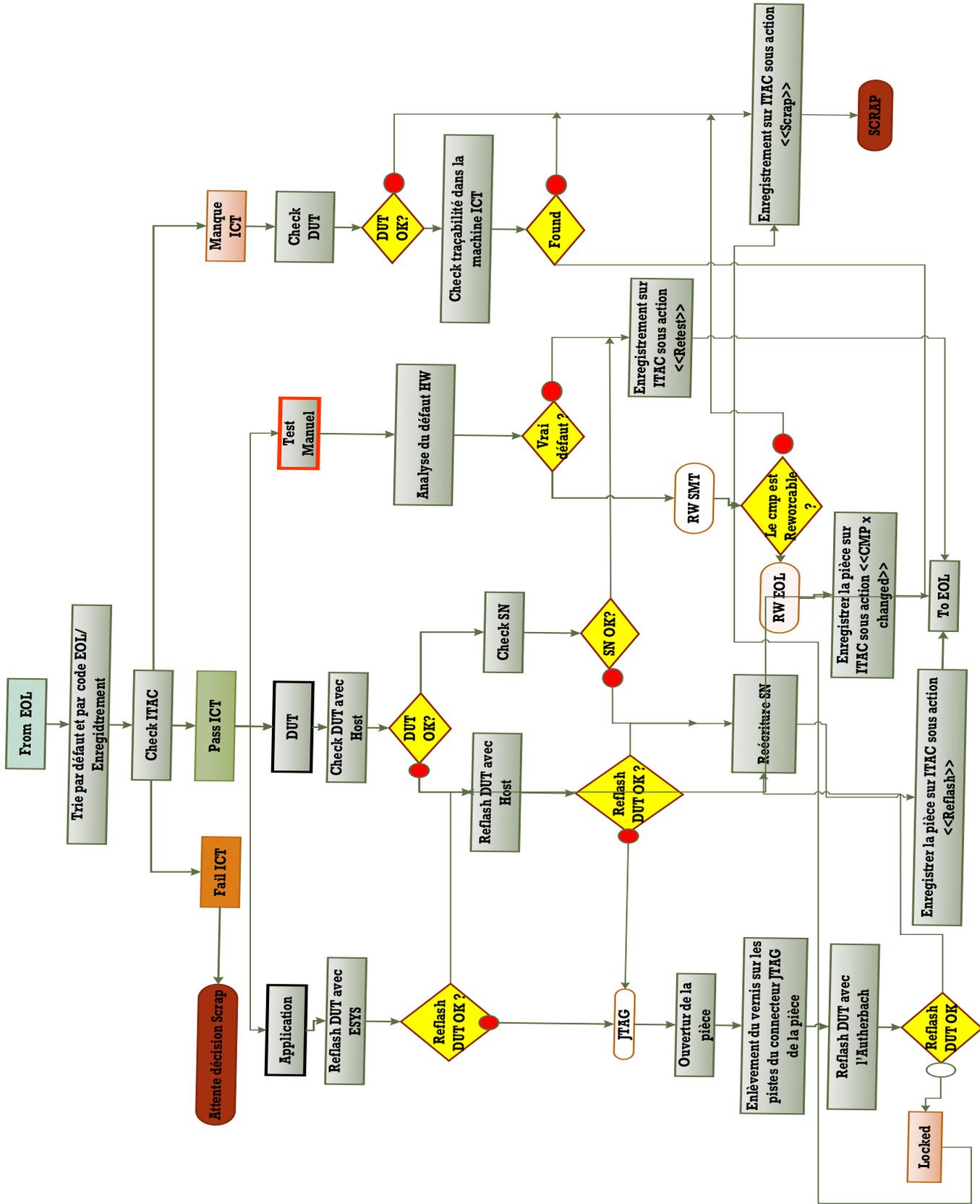


Figure 28: Block diagramme Rework

Les anomalies détectées lors de l'étape « End of Line » sont :

- Vrai défaut

Lorsque le défaut est détecté dans l'étape <<End of Line>>, il est alors aussi dans la station REWORK, ce qui montre qu'il y a un vrai défaut au niveau Hardware de la carte PL7, à cause d'une défaillance dans le processus ou bien du manque d'une telle étape.

- Faux défaut

Lorsque le défaut est détecté au test fonctionnel mais n'est pas confirmé au Rework, on dit qu'on a un faux défaut qui pourrait être dû à la machine de test automatique.

- Valeur limite

Lorsque le test de « End of Line » déclare une fonction comme NOK, et quand on teste la même fonction dans le Rework et on remarque que la valeur mesurée est très proche de l'intervalle de tolérance de fonction, alors dans ce cas on considère que le résultat de ce test est OK.

➤ Notre projet présente une nouvelle méthode permettant d'effectuer les tests électriques dans la station Rework pour le diagnostic qui consiste donc à isoler les problèmes de la carte et par la suite les repérer. En effet, notre application assure au technicien une analyse globale sur le hardware de la carte PL7 d'une manière automatique et plus rapide.

3.1.2 Description des étapes du diagnostic

Après la mise sous tension de la carte, l'application de notre système de **diagnostic** du produit exécute les étapes suivantes :

1. Mettre la carte sous tension

Cet étape consiste à régler la tension de l'alimentation à 12,5 V, à mettre les connecteurs à leur place adéquate et finalement brancher correctement le CANCEASE entre le PC et la carte.

2. Initialisation

Avant d'établir la communication par protocole CAN, nous proposons de passer par l'étape de l'initialisation afin d'améliorer la connexion. Cette phase permettra de rafraîchir l'interface graphique et surtout d'initialiser le Driver.

3. Connecter le PC avec la carte via le CANCEASE

Avant de commencer les tests, une communication entre le système de test et le produit doit être établie. C'est l'étape primordiale qui consiste à créer, à configurer, et à activer

les ports au niveau du programme pour pouvoir, par la suite lire les données à partir des microcontrôleurs ZGW et Body de la carte, et d'envoyer les frames à partir de notre application.

4. Check state (lire numéro de séries et la version DUT)

Après avoir réussi l'établissement d'une connexion CAN avec le produit, nous aurons besoin de lire le numéro de série à partir du microcontrôleur de la carte, ainsi de lire la version DUT afin de vérifier que le DUT existe dans le microcontrôleur. Des frames sont donc envoyés dans cette phase pour extraire les données correspondantes.

5. Téléchargement du fichier de fonctions

Au niveau de notre projet, nous avons choisis de rassembler toutes les fonctions dans un fichier qui permet d'ajouter ou de supprimer les fonctions dans tous le temps. Ce fichier contient les frames des fonctions ainsi que leurs valeurs analogiques et digitales, alors que cette étape est nécessaire afin de définir les fonctions pour lancer le test.

6. Test détaillé (fonction par fonction)

Après définies les fonctions, notre application permet à l'utilisateur de faire le test des fonctions une par une, et afficher le résultat du test.

7. Test complet

Notre programme doit parcourir toutes les fonctions qui existent dans le fichier pour récupérer le frame de chaque fonction afin de convertir et de l'envoyer au microcontrôleur, ce dernier contient le DUT qui doit répondre et envoyer les résultats sous forme des frames.

8. Affichage des résultats

Après l'envoi des frames, notre application doit recevoir le résultat sous forme des frames puis elle extrait la valeur correspondante et par la suite une conversion et une comparaison sont exécutées afin d'afficher le résultat de test.

9. Résultat statistique (CPK)

Le résultat statistique permet de faire une analyse importante de la qualité des mesures de test des fonctions. C'est le moment où l'entreprise peut vérifier qu'elle est en mesure de répondre à un cahier des charges. [9]

10. Résultats hardware

Le résultat hardware permet d'identifier le problème probable au niveau des composants électroniques, afin de réparer les composants défectueux.

11. Traçabilités

La traçabilité est aujourd'hui une technique indispensable qui permet de connaître toutes les informations nécessaires et suffisantes d'un produit. Elle jouera un rôle important dans la surveillance et l'appréciation de la qualité du test de l'application. Toutefois il ne suffit pas de constater qu'une carte PL7 n'a pas le même SN imprimé sur le label, il s'agit surtout de savoir à quelle étape, l'opération s'est arrêté. En effet, l'application doit générer un rapport de traçabilité pour chaque carte PL7 avec l'extension « .OK » pour les cartes qui ont réussi le test et « .NOK » pour les cartes ne l'ont pas réussi.

3.2 Les Diagrammes de système du diagnostic

Les diagrammes suivants exposent les étapes essentielles qu'il faut suivre pour réaliser la communication CAN avec le module PL7 FEM et d'effectuer l'analyse et le diagnostic de la carte :

3.2.1 Diagramme de communication par protocole CAN

Notre système de diagnostic doit réaliser avant tout une communication (envoie les frames et reçoit le feedback) avec le produit, cette étape consiste à ouvrir le driver, ouvrir les ports, définir le filtre d'acceptation des frames et l'activation des canaux. Ce processus de communication est illustré au niveau de la figure suivante :

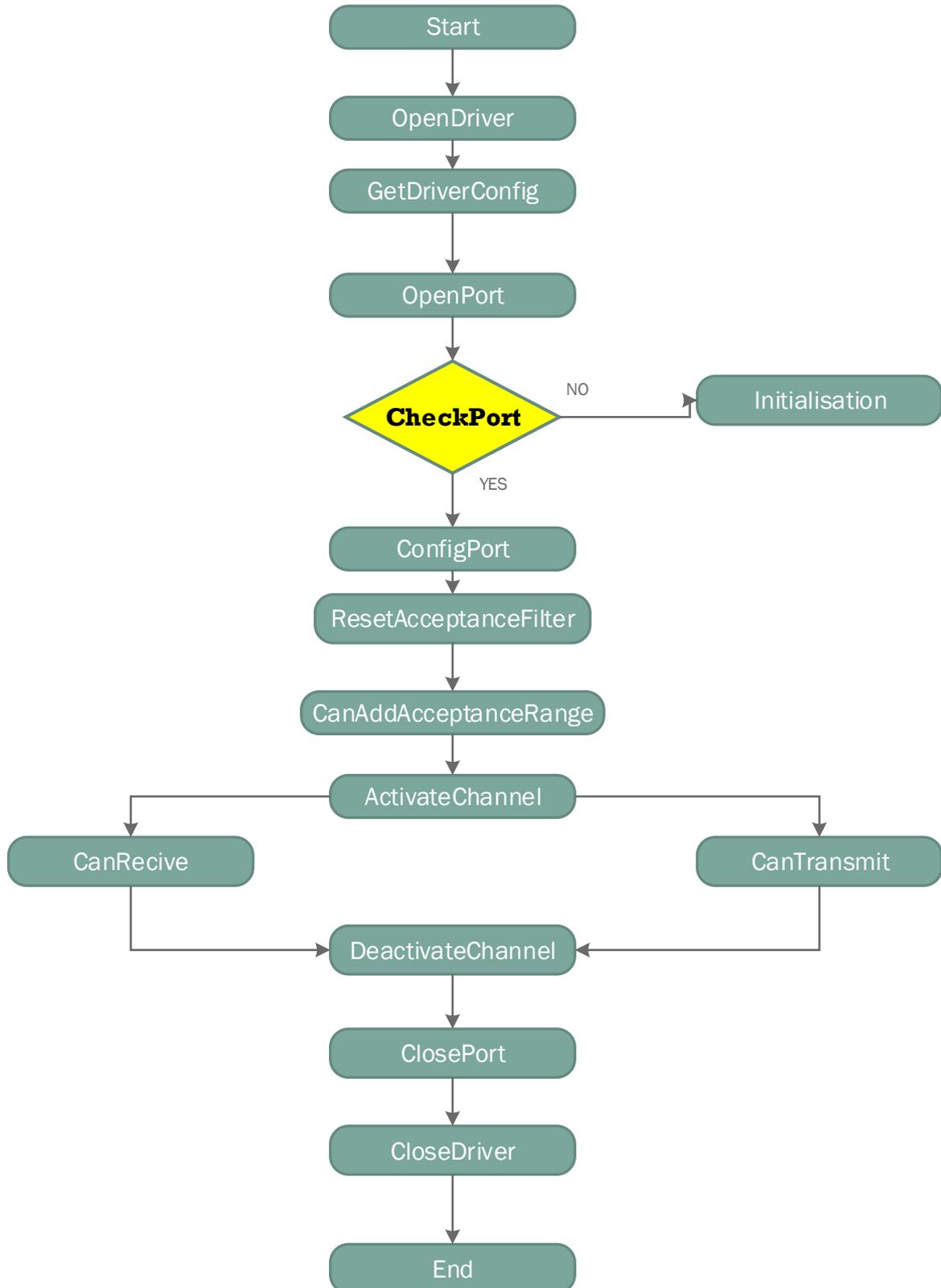


Figure 29 : Diagramme de communication du protocole CAN

3.2.2 Digramme de système de diagnostic général

Le diagramme suivant expose les étapes essentielles qu'il faut suivre pour réaliser une analyse et un diagnostic de la carte PL7 FEM :

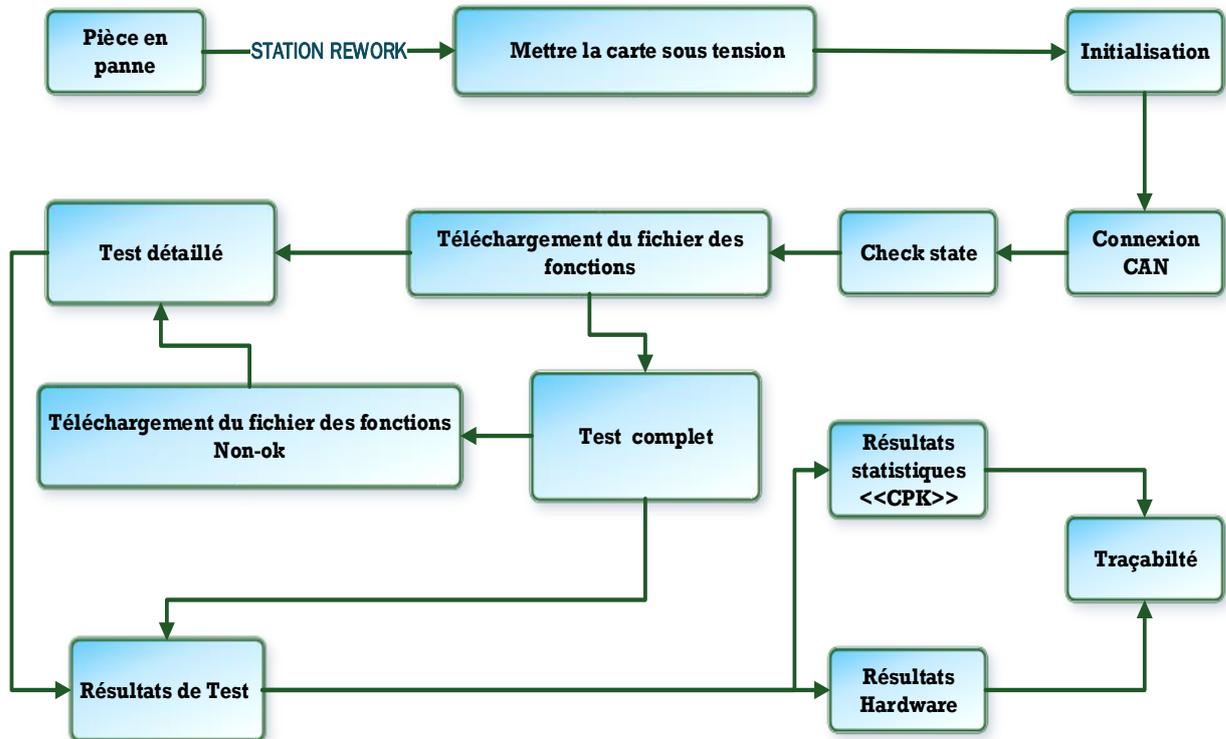


Figure 30 : Diagramme du système de diagnostic

Conclusion

Ce chapitre avait pour objet de décrire les étapes et l’algorithme à suivre pour effectuer les tests sur le produit. Nous présentons dans Le chapitre suivant le développement de l’application en se basant sur la conception par langage UML.

Chapitre 4 : Conception et développement du système de diagnostic

Dans le chapitre précédent, nous avons vu la procédure du système de diagnostic de la carte PL7 FEM par une étude software afin de choisir le bon algorithme qui décrit précisément les étapes et les règles techniques appliquées dans la phase de conception et du développement. Dans ce chapitre nous abordons la conception et le développement de notre système de diagnostic qui est divisé en trois parties essentielles :

- Conception par le langage UML (Unified Modeling Language) :

Nous présentons la conception générale du projet à travers les diagrammes UML suivants : cas d'utilisations, séquence et d'activité.

- Réalisation et test de vérification :

Dans cette deuxième partie, nous détaillons les deux manières de tests par les diagrammes d'UML, et nous expliquons notre développement par la méthode de MVC (Model-View-Controller).

- Les interfaces graphiques :

Dans cette dernière partie, nous présentons les captures d'écrans des différentes pages de l'application, en expliquant les rôles de chaque objet.

4.1 Conception par le langage UML

Dans cette partie, nous étudions le comportement de conception du système à travers le langage UML, UML est un langage de modélisation graphique unifié à base d'une représentation schématique, conçu pour fournir une méthode normalisée ainsi de visualiser la conception du système et qui permet de mieux comprendre le système à développer.

Il permet :

- de visualiser le système comme il devrait l'être,
- de spécifier les structures de données et le comportement du système,
- et de fournir un guide pour la construction du système.

Dans cette partie, nous allons présenter en détail la conception du projet à travers les diagrammes UML suivants :

- les diagrammes des cas d'utilisation,
- les diagrammes de séquence,

- et les diagrammes d'activités.

4.1.1 Diagramme de cas d'utilisation général

Un cas d'utilisation est une manière spécifique pour utiliser un système. Les acteurs sont à l'extérieur de celui-ci, ils modélisent tout ce qui interagit avec le système de diagnostic. Les principaux concepts de ce diagramme sont les acteurs, les cas d'utilisation et les sujets. Un sujet représente un système avec lequel les acteurs et les autres sujets interagissent (Figure 31).

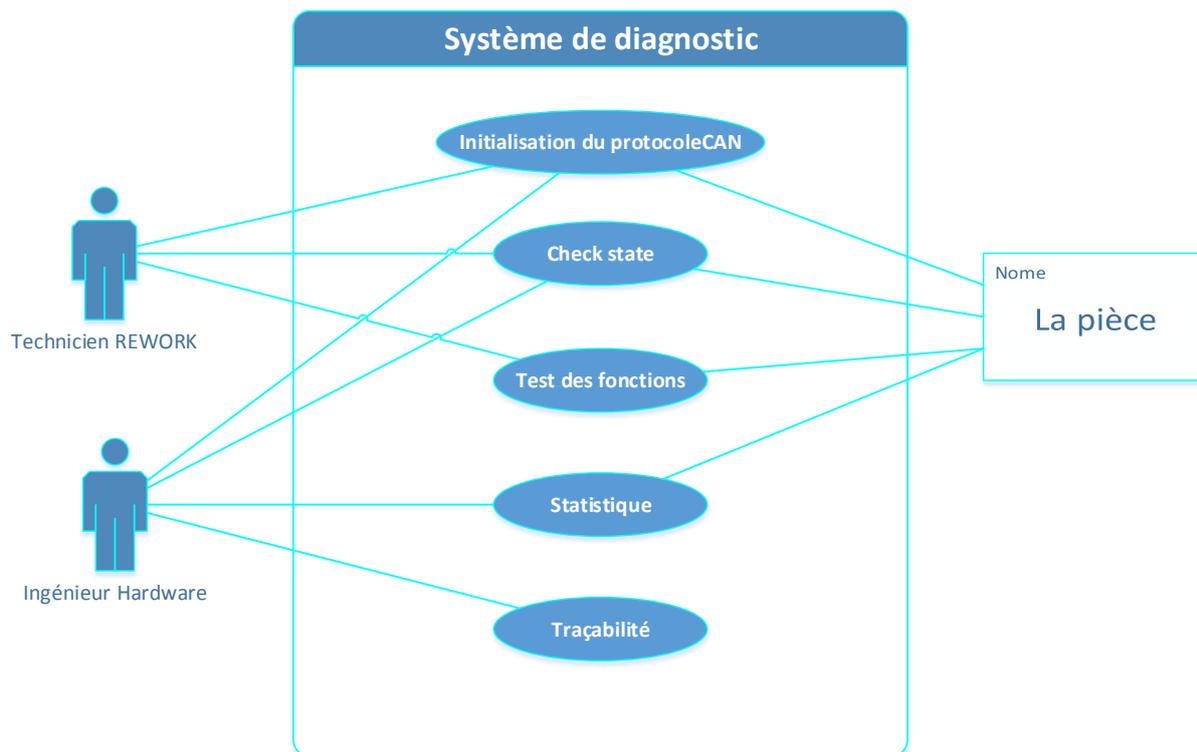


Figure 31 : Diagramme de cas d'utilisation générale

- **Identification des acteurs :**

Au niveau de ce paragraphe, nous présentons les différents acteurs susceptibles d'interagir avec le système de diagnostic. Mais tout d'abord, nous donnons une définition du concept acteur. Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Technicien Rework : son rôle consiste à analyser les produits défectueux par des tests automatiques soit test détaillé (fonction par fonction) ou bien test complet, afin de déduire les causes racine des défauts.

Ingénieur hardware : son rôle à suivre en temps réel la traçabilité de produit et à demander des valeurs de tests de la pièce sous forme de fichiers avec le numéro de série sur son nom et avec l'extension « .OK » pour les cartes qui ont réussi le test et «.NOK» pour les cartes ne l'ont pas réussi. Ainsi il a une idée sur la qualité des mesures de la partie statistique.

La pièce : c'est le produit en panne.

4.1.2 Diagramme d'activité générale

Dans cette sous-section, nous allons voir l'alternative visuelle des descriptions détaillées de cas d'utilisation. Il s'agit du diagramme d'activité. Celui-ci permet de modéliser notre processus interactif global pour notre système de diagnostic, et représenter le déclenchement d'événements des états du système .

Les événements des états de notre système ,

- initialisation de la connexion,
- lire le numéro de série,
- lire la version DUT,
- télécharger le fichier des fonctions,
- test détaillé,
- et test complet.

La figure suivante représente notre diagramme d'activité :

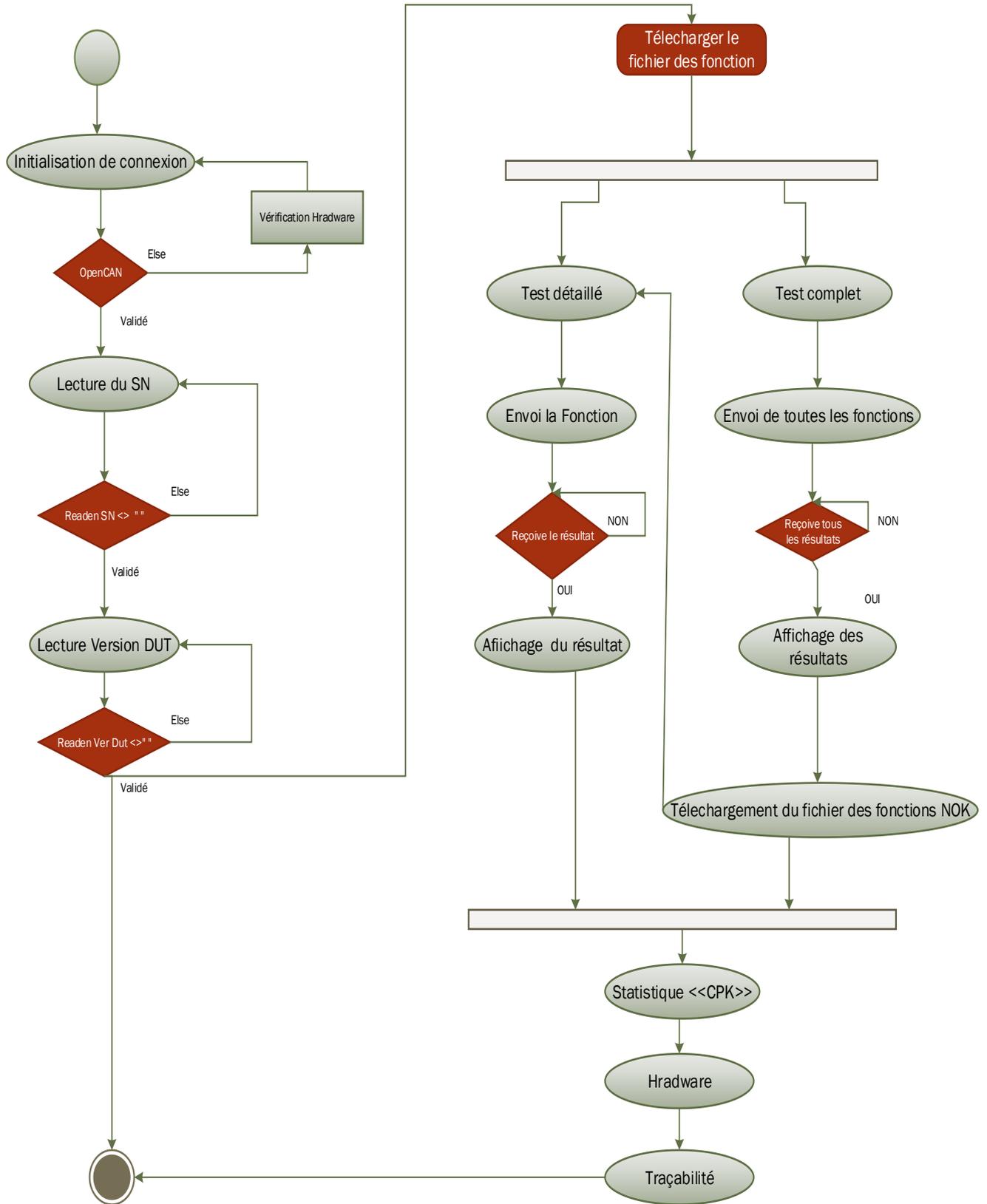


Figure 32 : Diagramme d'activité générale

4.1.3 Diagramme de séquence générale

Le diagramme « cas d'utilisation général » nous allons l'expliquer par l'utilisation du diagramme de séquence. Ce dernier est un outil de mise en évidence des interactions entre des objets de système, pour atteindre un objectif fixé par l'utilisateur.

Notre diagramme de séquence du système d'analyse peut être vu comme une implémentation d'un scénario. Le rôle de note diagramme est double :

- Définir précisément les traitements à mettre en œuvre pour répondre à la demande de l'utilisateur.
- Conforter, préciser, amender la structure des données de l'application. Tant du point de vue, les liaisons entre les sous-systèmes et la répartition des opérations dans le système.

La figure 33 illustre notre diagramme de séquences :

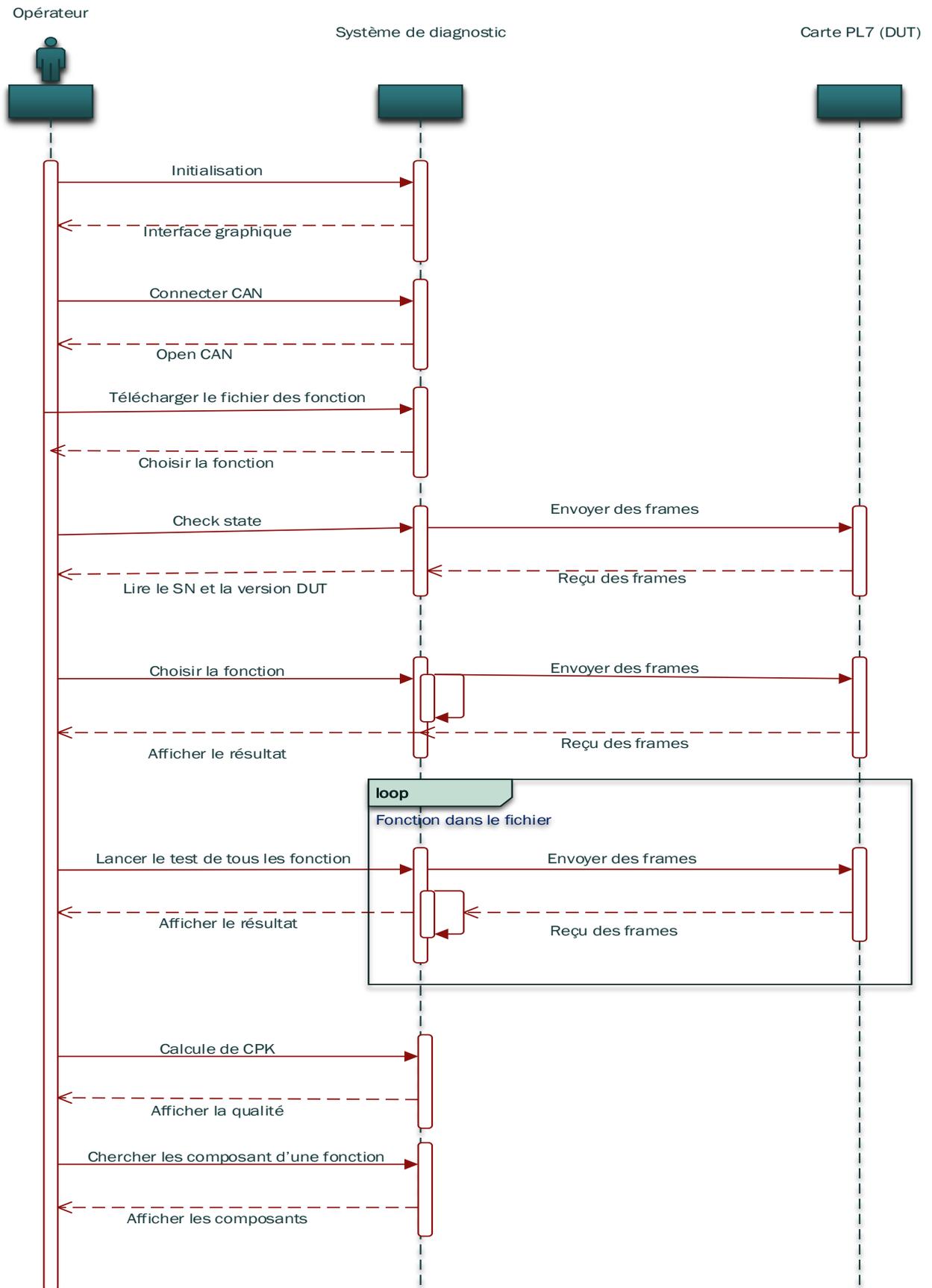


Figure 33 : Diagramme de séquence générale

4.2 Conception de deux manières de test

4.2.1 Test détaillé

4.2.1.1 Diagramme de cas d'utilisation de «Test détaillé»



Figure 34 : Diagramme De cas d'utilisation de test détaillé

- **But** : permet à l'opérateur d'analyser les fonctions des cartes par test détaillé qui donne le résultat de test « ok » ou « nok » et affiche les valeurs pour une meilleure analyse.
- **Les acteurs** : Les techniciens.
- **Description textuelle** : Dès que la carte défailante arrive à la station Rework pour l'analyse, le technicien doit appliquer sur celle-ci un ensemble des tests afin de recevoir le feedback. Ce test détaillé permet de visualiser les valeurs du test qui permet au technicien de décider le type de défauts (valeur limite, faux défaut, vrai défaut).
- **Les scénarios** :
 - ✓ Télécharger le fichier des fonctions : permet de télécharger un fichier dont des fonctions à tester, selon l'enchaînement suivant :
 1. Choisir le fichier des fonctions.
 2. Afficher le numéro, le nom et le chanel ID de la fonction.
 - ✓ Tester la fonction : permet de choisir la fonction à tester et le système va envoyer les frames convenable à travers le protocole de communication CAN.

1. Extraction du frame de fichier : Après avoir choisi le nom de fonction ainsi sa chanel ID, le système cherche le frame correspondant et l'extrait par le fichier.
 2. Après l'extraction du frame le système doit le convertir et l'envoyer à travers le protocole de communication CAN, afin de recevoir le résultat.
- ✓ Lire le résultat : permet à l'utilisateur de lire le résultat sous forme de symbole « ok » ou « nok » et d'afficher les valeurs.

4.2.1.2 Diagramme d'activité de test détaillé

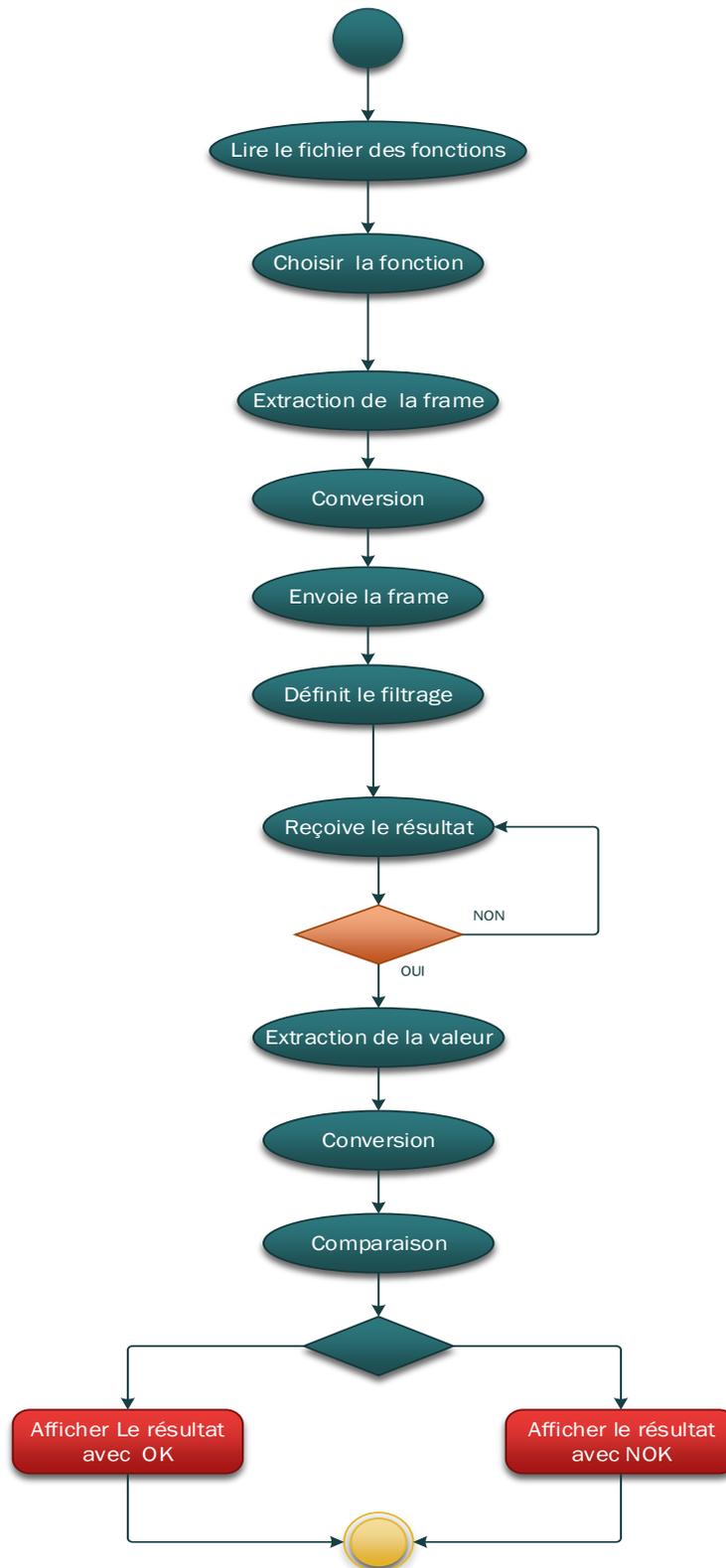


Figure 35 : Diagramme d'activité de test détaillé

4.2.1.3 Diagramme de séquence de « Test détaillé »

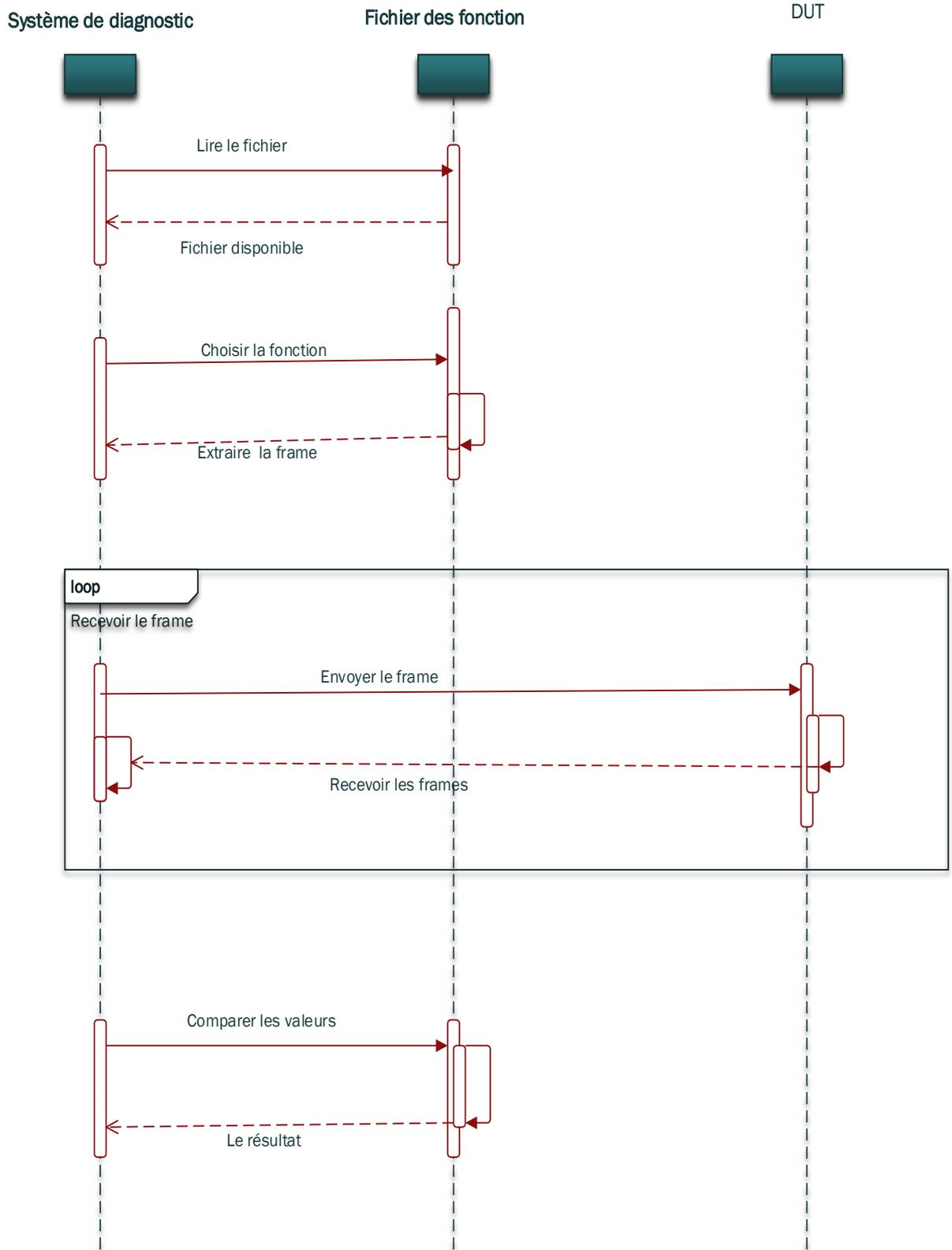


Figure 36 : Diagramme d'activité de test détaillé

4.2.2 Test Complet

4.2.2.1 Diagramme de cas d'utilisation

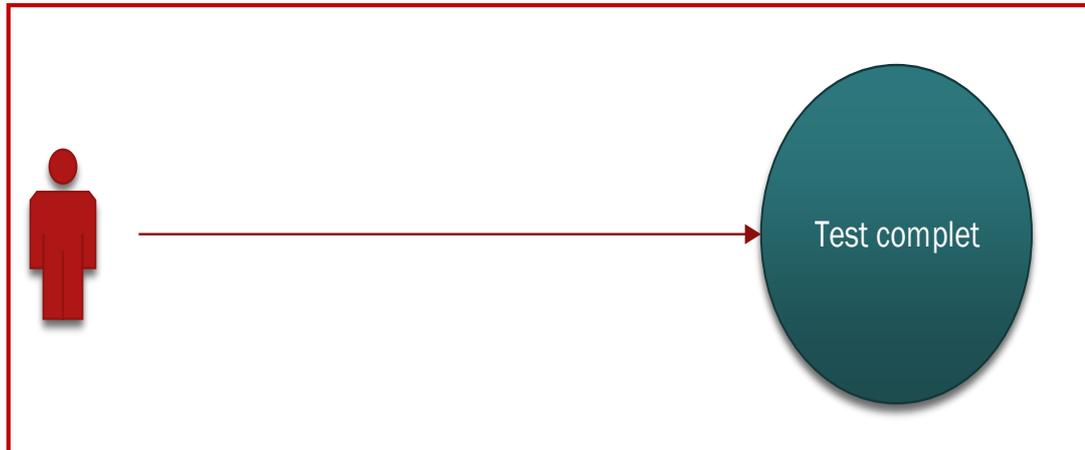


Figure 37 : Diagramme de cas d'utilisation de test complet

- **But :** Ce cas d'utilisation permet au technicien de tester toutes les fonctions d'après un fichier.
- **Les acteurs :** Les techniciens.
- **Description textuelle :** Lancement de ce test permet de parcourir toutes les fonctions du fichier et appliquer les tests .Afin, de générer un autre fichier qui contient les fonctions « Nok » et d'autre part de garder la traçabilité.
- **Les scénarios :**
 - ✓ Télécharger le fichier des fonctions : permet de télécharger un fichier qui a des fonctions à tester.
 - ✓ Lancer le test : le système doit parcourir tout le fichier et envoyer toutes les fonctions selon l'enchaînement suivant :
 1. Extraction de frame : le système doit extraire le frame de chaque fonction.
 2. Conversion de frame : le système doit convertir chaque frame qui s'adapte avec le protocole de communication CAN.
 3. Envoyer le frame : le système doit envoyer chaque fonction.
 4. Recevoir le résultat : le système doit recevoir tous les résultats.

5. Stockage des fonctions « Nok » : le système doit parcourir le fichier des fonctions et stocker en même temps chaque fonction « Nok » dans un autre fichier.
 6. Garder la traçabilité : le système doit garder la traçabilité sous forme de fichier avec l'extension « ok » ou « Nok ».
- ✓ Afficher le résultat : le système doit afficher tout le résultat de tests avec toutes les informations.

4.2.2.2 Diagramme d'activité de Test complet

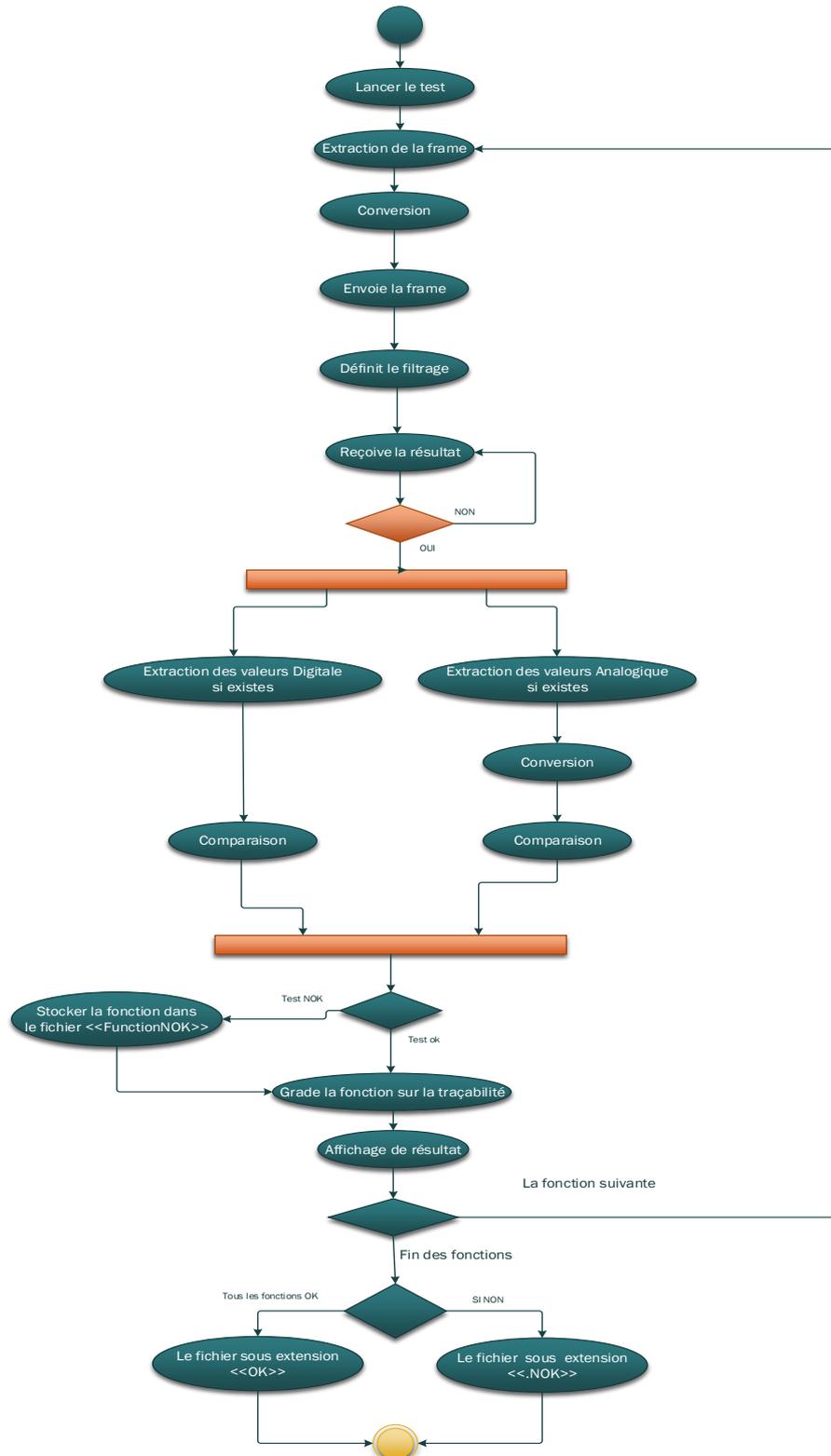


Figure 38 : Diagramme d'activité de Test complet

4.2.2.3 Diagramme de séquence de « Test complet »

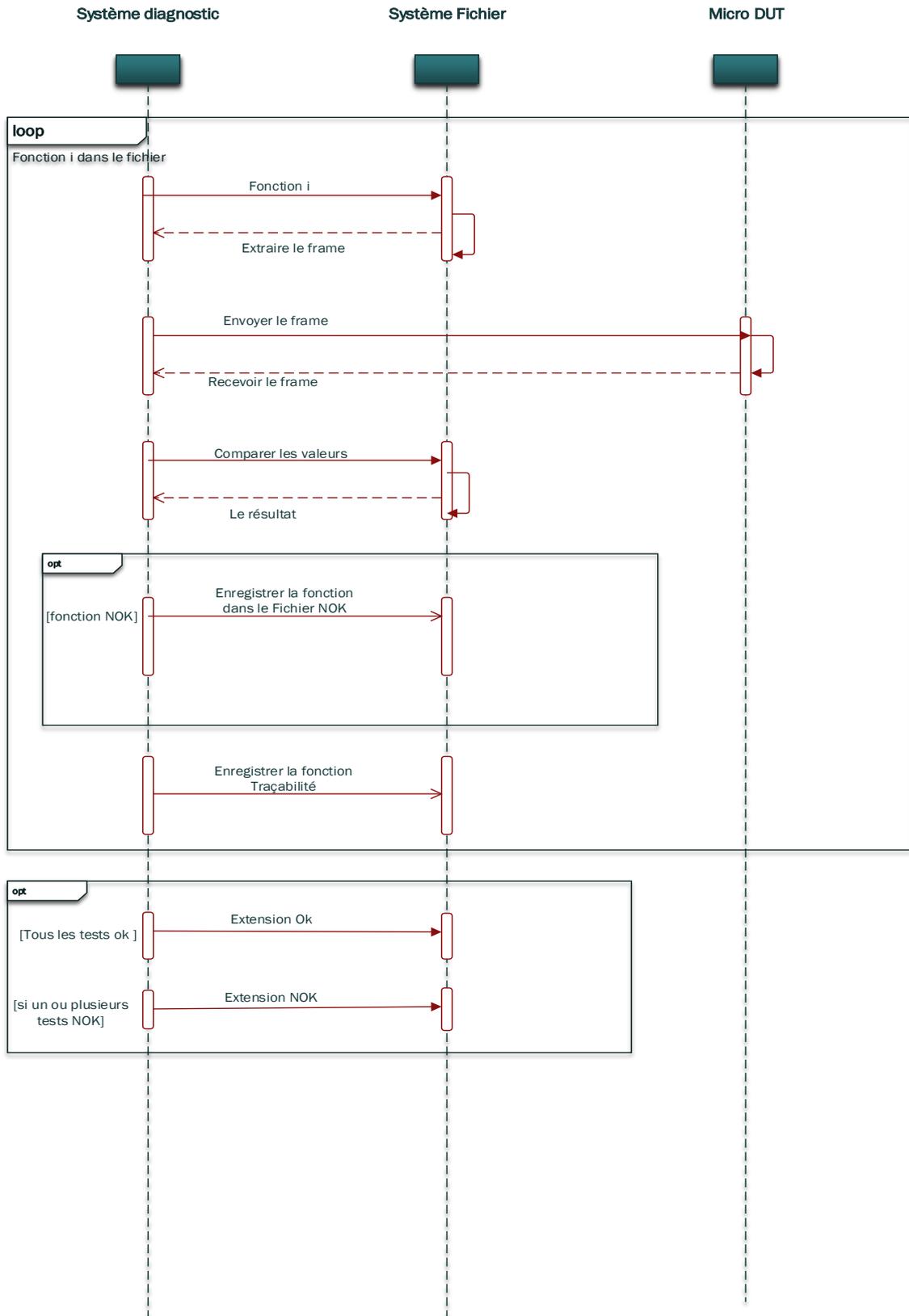


Figure 39 : Diagramme de séquence de « Test complet »

4.3 Réalisation et tests de vérification du système de diagnostic

4.3.1 Environnement du travail

Dans cette partie, nous mentionnons les caractéristiques des outils matériels et surtout les outils logiciels du développement dans notre application de diagnostic.

4.3.1.1 Environnement matériel

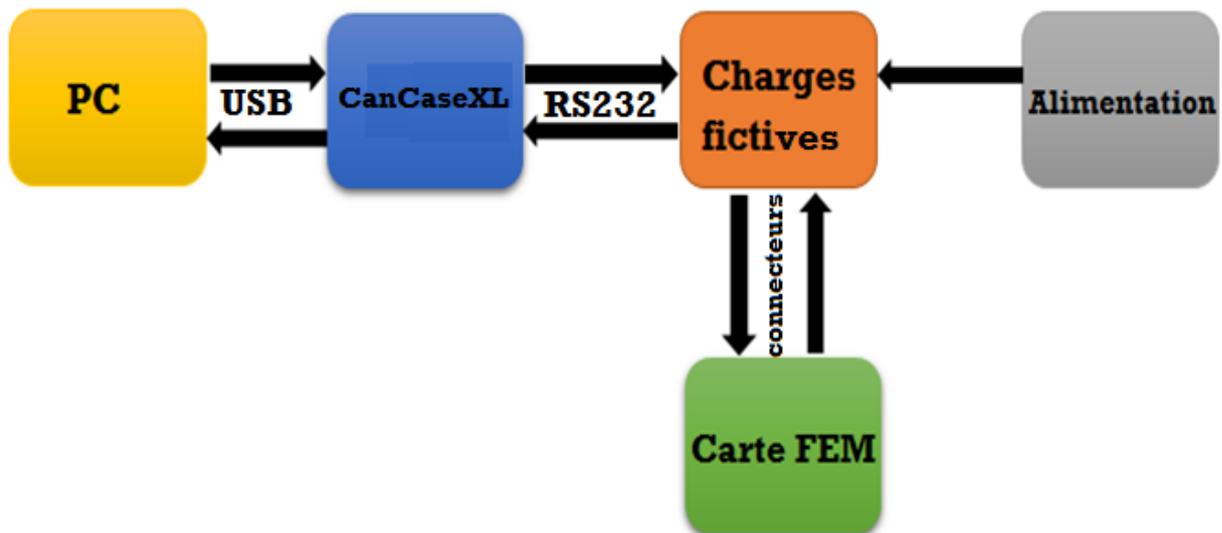


Figure 40 : Environnement matériel

L'ordinateur portable : L'application a été développée sur un ordinateur portable DELL qui se caractérise par :

Processus : Intel® Core™ i3-3217U CPU @ 1.80 GHz 1.80 GHz.

Mémoire installée (RAM) : 4.00 Go (3.89 Go utilisable).

Type de système : système d'exploitation 64 bits.

Disque dur : 500 Go.

Ecran : 15,6 pouces.

Le CanCasexl : un équipement de société « Vector » qui donne la possibilité d'envoi et réception de données en utilisant le protocole CAN/LIN.

Charge fictives : son rôle à transformer les frames arrive de CanCase à des signaux pour tester le produit.

Alimentation : Son rôle à fournir l'électricité aux autres appareils électriques est fixé sur 12.5 V.

Carte FEM : c'est le produit défaillant.

4.3.1.2 Environnement logiciel

Nous avons énuméré au cours de cette section les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

➤ **Langage de programmation :**

- Choix du langage de programmation :

Écrire des programmes nécessite l'utilisation d'un langage de programmation. Il y a toute une collection de langages disponibles. Ces langages ne sont pas, essentiellement équivalents. Chaque langage possède ses avantages et ses inconvénients. Afin de choisir un langage de programmation, un informaticien doit tenir compte d'un grand nombre de critères :

- Facilité d'apprentissage,
- Facilité d'utilisation pour un programmeur expérimenté,
- Rapidité d'exécution des programmes,
- Stabilité (absence de défaut),
- Possibilité de structuration modulaire du programme,
- Existence de bibliothèques de fonctions, de classes,
- Portabilité du langage sur différentes plateformes, etc.

Dans notre cas nous avons décidé de choisir le langage VB.NET pour les raisons suivantes :

- La disponibilité des bibliothèques du protocole CAN,
- La disponibilité des drivers du CANcaseXL,

- L'ensemble des programmes des machines de Test sont développés en VB.NET,
- **Architecture du Framework .NET**

En informatique, un Framework est un espace de travail modulaire. Le Framework .NET ne déroge pas à cette règle. C'est un composant logiciel à la base développé par Microsoft qui peut être ajouté au système d'exploitation Windows. Il existe maintenant des implémentations libres de .NET.

Ce Framework a pour but de faciliter la tâche des développeurs en proposant une approche unifiée à la conception d'applications. Il introduit également des facilités pour le développement, le déploiement et la maintenance d'applications. La stratégie de Microsoft est de développer un logiciel qui s'utilise sur un seul système d'exploitation Windows mais qui intègre une multitude de langages. Cette stratégie s'oppose à celle d'un autre Framework concurrent : Eclipse. Ce dernier logiciel fonctionne sur n'importe quel système d'exploitation mais intègre, par défaut, uniquement le Java. [11]

- **Langage VB.NET**

Comme nous l'avons déjà dit précédemment le langage VB.NET (aussi appelé VBx où x est supérieur à 6) est une évolution de Visual Basic (version inférieure à 7). Néanmoins, il y a une réelle rupture entre ces deux langages. VB6 avait déjà introduit la notion d'objet afin d'utiliser la programmation événementielle, cependant elle n'admettait pas la programmation objet en tant que tel. VB.NET est un véritable langage orienté objet qui fait apparaître l'héritage, la surcharge des fonctions qui n'était pas dans VB6. [12]

VB.NET étant un langage objet, il n'est donc pas étonnant de constater qu'il est très proche du langage Java. Les concepts de haut niveau de ces deux langages sont voisins. Cependant chaque langage a ses spécificités, VB.NET introduit notamment la notion de module et de Namespace absentes dans Java.

Un module est un type de référence similaire à une classe mais qui a des membres implicitement partagés. De plus, un module a la spécificité de ne jamais être instancié, il ne prend pas en compte l'héritage et ne peut pas s'appliquer à des interfaces. Un Namespace, quant à lui, est similaire à la notion de package en Java puisqu'il permet d'organiser d'autres entités.

En ce qui concerne les autres éléments de ce langage, ils respectent la philosophie objet. En effet, VB.NET possède des classiers qui peuvent être des classes, des interfaces. Ces classiers ont la possibilité de spécifier des attributs ainsi que des méthodes.

➤ **Driver CancaseXL**

Notre application de diagnostic doit communiquer avec la matériel CanCase, pour le fait il a besoin d'une interface logicielle qui joue l'intermédiaire entre le système et le périphérique externe, c'est le driver CancaseXL.

Voici une capture d'écran de configuration de notre l'application sur le driver :

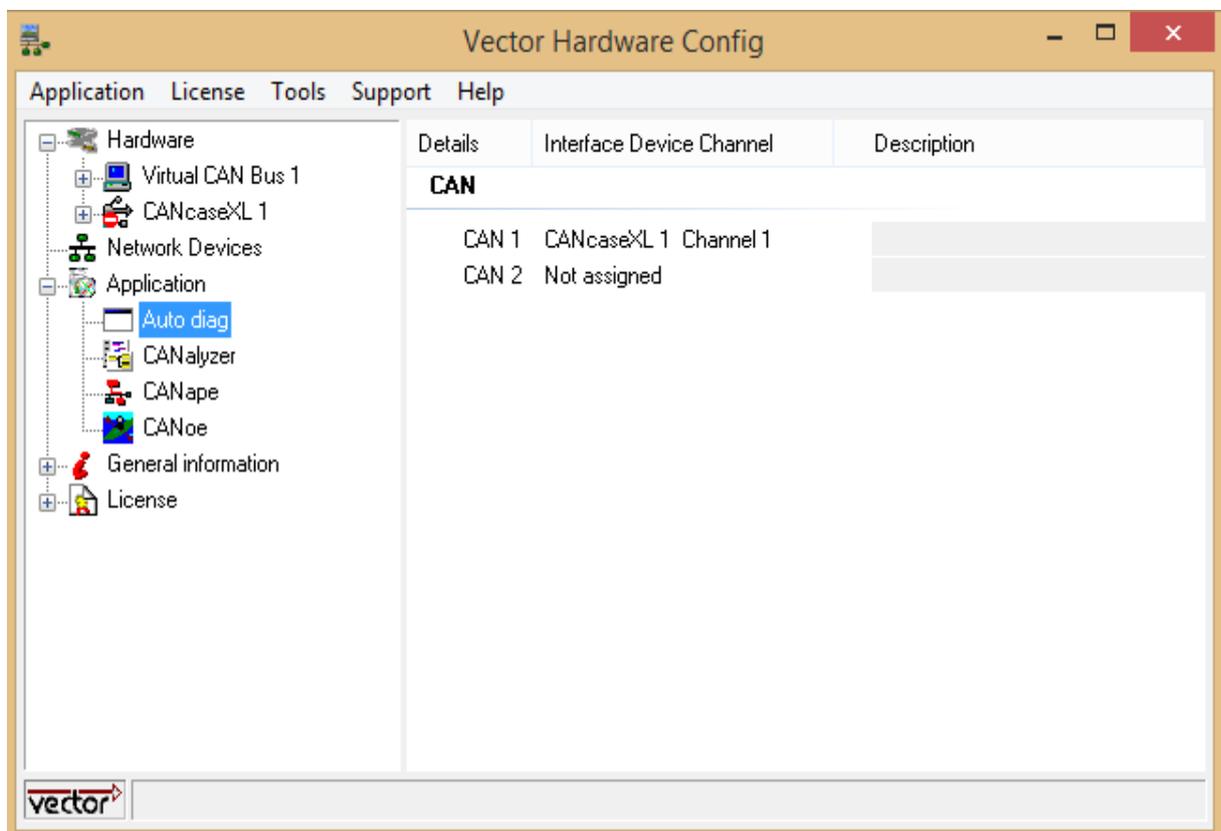


Figure 41 : Driver CAN

4.3.2 Développement

Pour le développement, nous avons appliqué le modèle Model-View-Controller (MVC). Ce paradigme divise l'IHM (Interface Homme Machine) en un modèle (M pour algorithmique le cœur de l'application) une vue (V pour la présentation, l'interface graphique) et un contrôleur (C pour la logique de contrôle, et la gestion des événements / synchronisation), chacun ayant un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate.

L'architecture MVC ne résout pas tous les problèmes. Elle fournit souvent une première approche qui peut être adaptée par la suite. Elle offre aussi un cadre pour structurer une application.

Ce patron d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.[14]

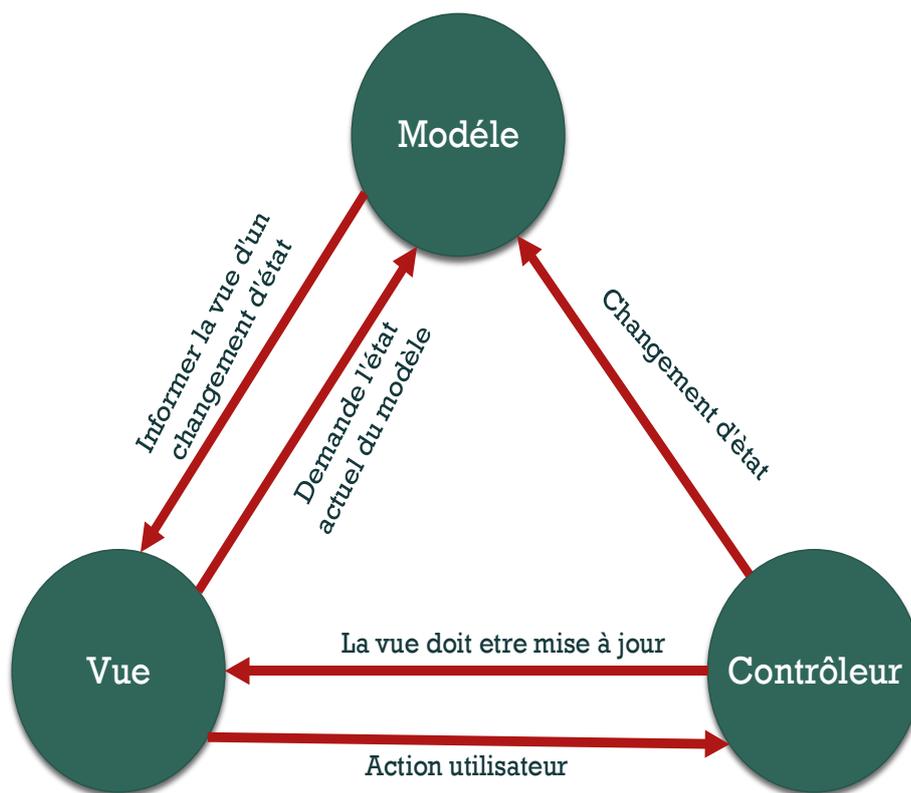


Figure 42 : Diagramme de MVC

La vue :

Ce que l'on nomme « la vue » est en fait une IHM (Interface Homme Machine). Elle représente ce que l'utilisateur a sous les yeux. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Le modèle :

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec des données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et il est responsable de leur intégrité. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur...). Il offre aussi des méthodes pour récupérer ces données.

Le contrôleur :

Il permet de faire le lien entre la vue et le modèle lorsqu'une action utilisateur est intervenue sur la vue. C'est cet objet qui a le rôle de contrôler les données et il prend en charge la gestion des événements de synchronisation. Il reçoit tous les événements de l'utilisateur et il enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier notifie la vue que les données ont changé pour qu'elle les mette à jour.

4.4 Phase de Test et Validation

Durant la réalisation de notre application, nous avons abordé deux grandes parties (matériel et logiciel), chacune génère sa partie des erreurs, alors il faut réaliser un ensemble de tests et de vérifications afin de trouver la bonne solution.

Dans cette partie, nous allons présenter une liste de modules de test, scénario de test et le résultat obtenu à partir du tableau suivant :

Module de test	Scénarios de test	Résultat obtenu
1. Application et driver	Configurer le driver.	Validé
2. Gestion des données	Lire les données du fichier des fonctions	Validé
	Extraction et affichage des fonctions sur l'interface graphique.	
3. Les connecteurs	Tester les connecteurs	Validé
4. Connexion Can	Activer et choisir le chanel adéquat	Validé
5. Lire le numéro de série	Envoyer les frames correspondants de SN	Validé
	Filtrage et réception des résultats	
6. Test de la fonction	Extraction du frame de fichier et son envoi.	Validé
	Réception de plusieurs frames par suite l'extraction du résultat	
7. Affichage le résultat	Conversion le feedback et le comparer avec les valeurs de fichier.	Validé

Tableau 3 : Tests de vérification

4.5 Les interfaces graphiques

L'interface graphique est une partie très importante pour notre application, qui devrait être convenable et conviviale offrant une certaine facilité à l'utilisateur au court des analyses.

Nous présentons maintenant un ensemble de captures d'écrans sur les principaux points de l'application :

4.5.1 Interface de Test

Au court de l'initialisation de l'application, une interface de test est affichée.

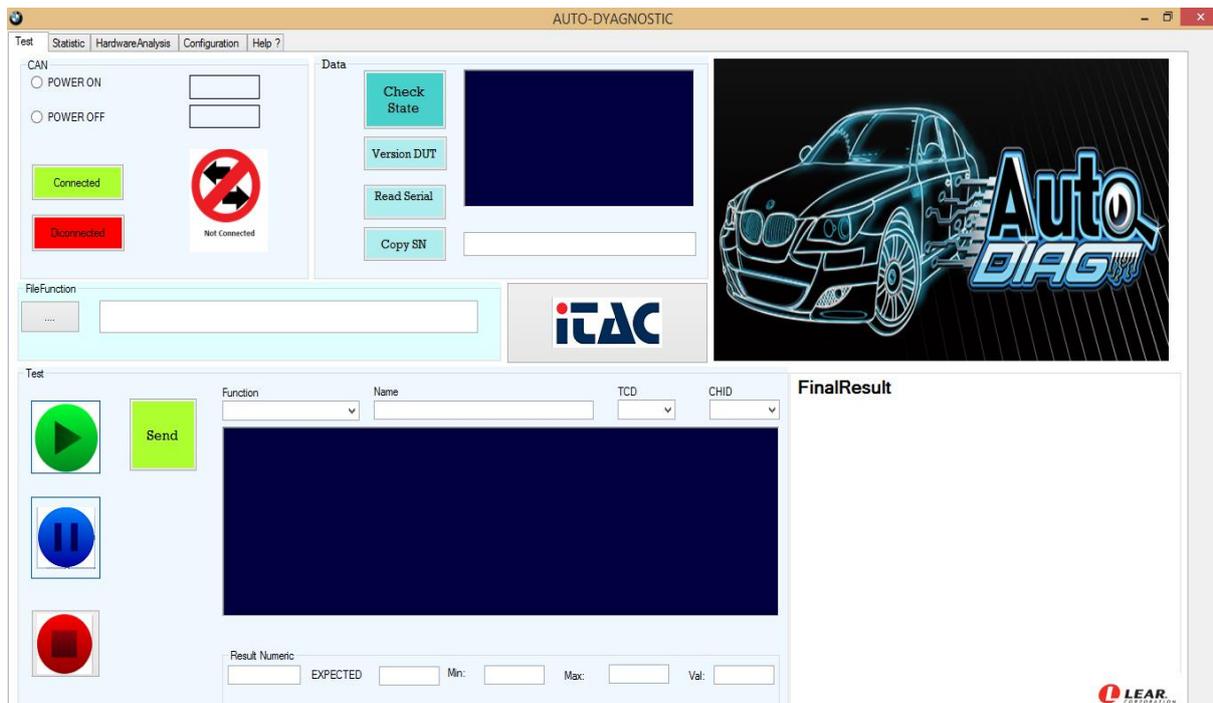


Figure 43 : Interface de Test

4.5.2 Interface de statistique

Après avoir connecté la pièce dans la première interface, celle-ci permet de faire les statistiques :

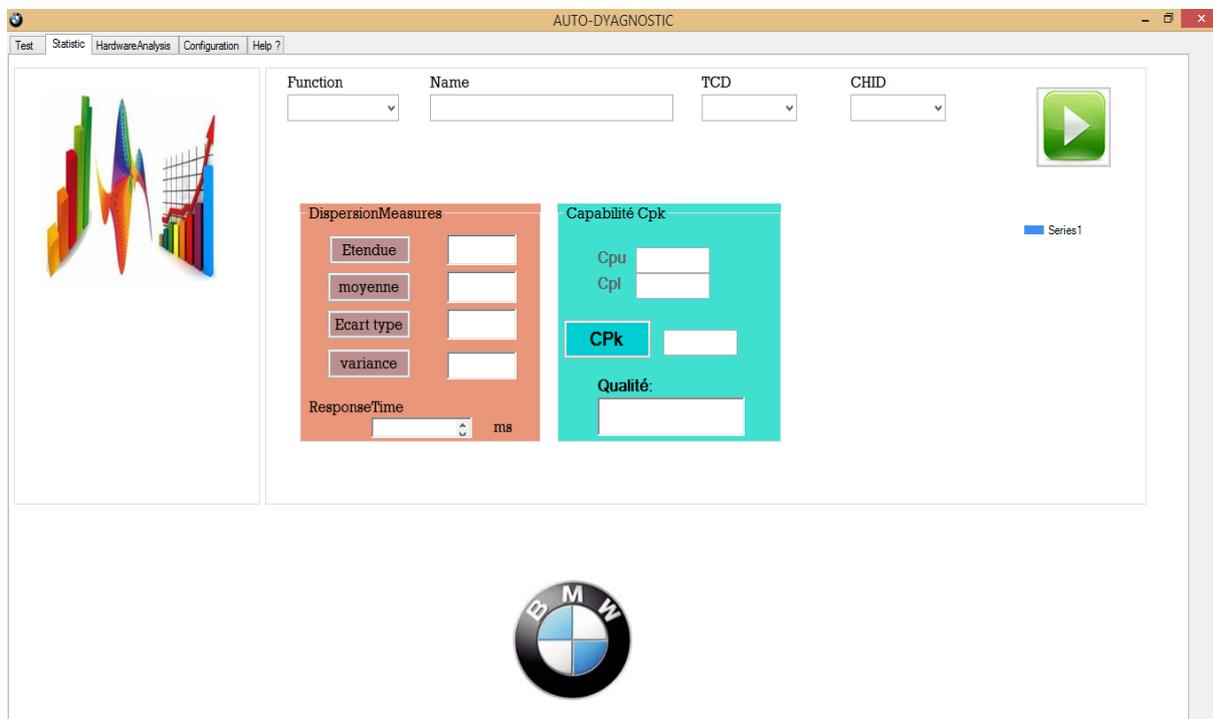


Figure 44 : Interface de statistique

4.5.3 Interface d'analyse hardware

Dès que le test complet est réalisé, la nouvelle interface permet d'afficher les composants des fonctions Nok pour le but d'isoler facilement le défaut hardware.

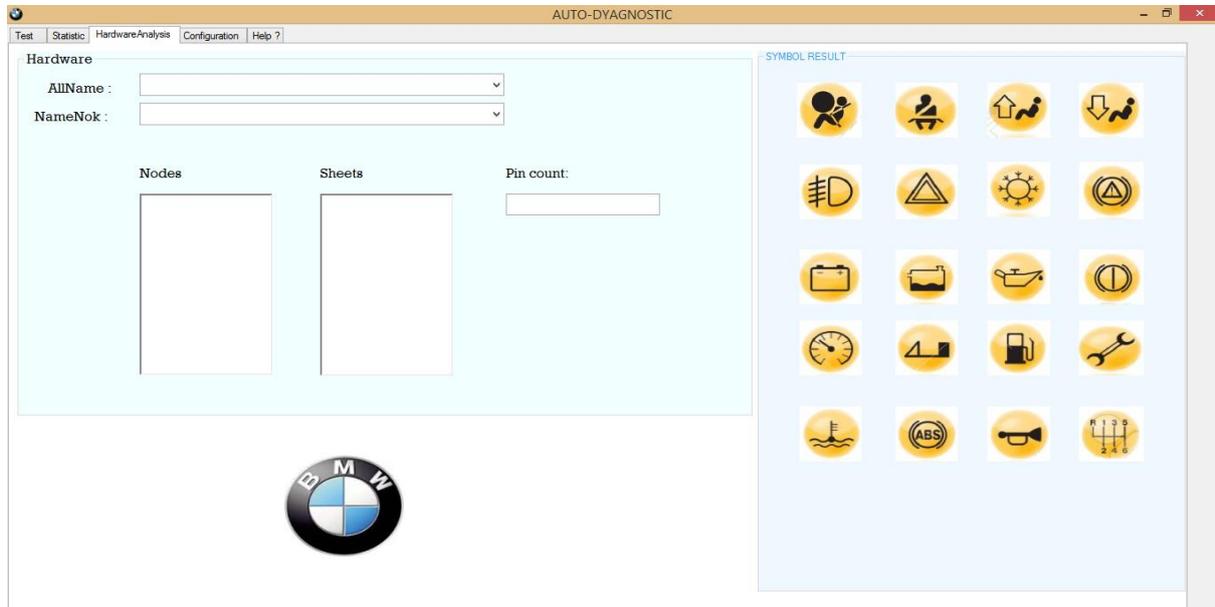


Figure 45 : Interface Hardware

4.5.4 Interface de configuration

Après avoir téléchargé le fichier des fonctions, cette interface permet d'ajouter les fonctions pour les tester.

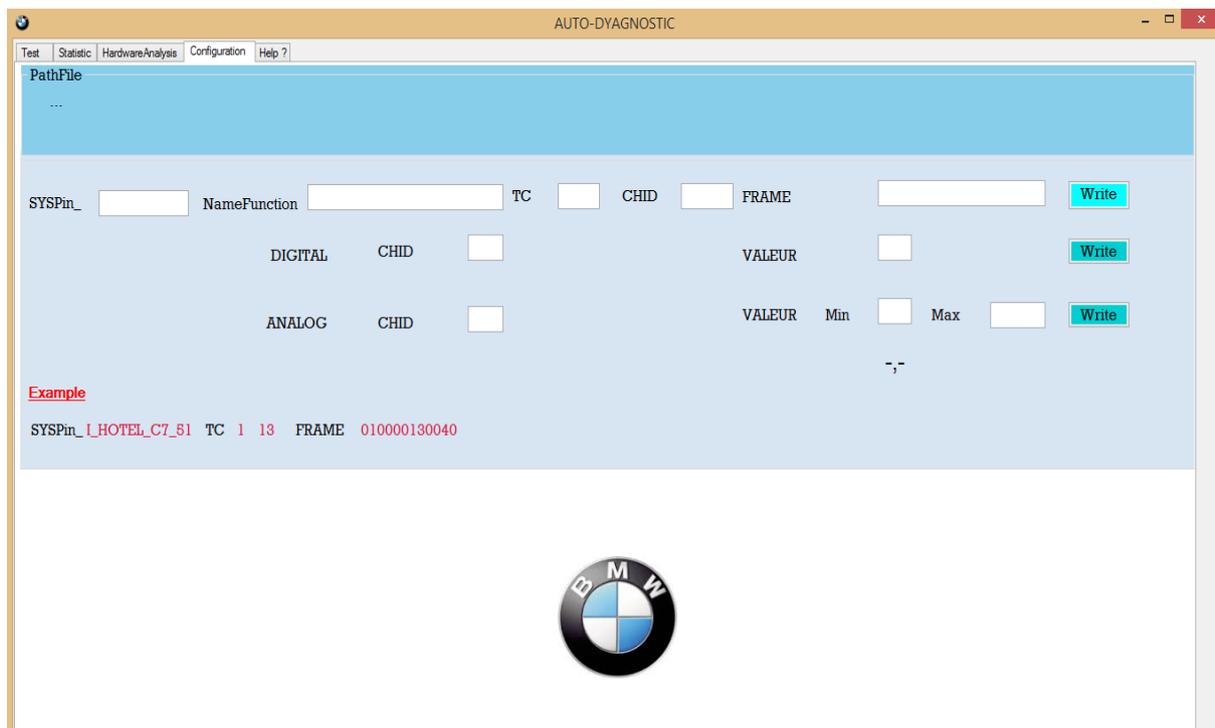


Figure 46 : Interface de configuration

✓ Exemple de test détaillé :

Nous avons testé la fonction « I_CENTRAL_LOCK_C7_27 » et le résultat est Ok.

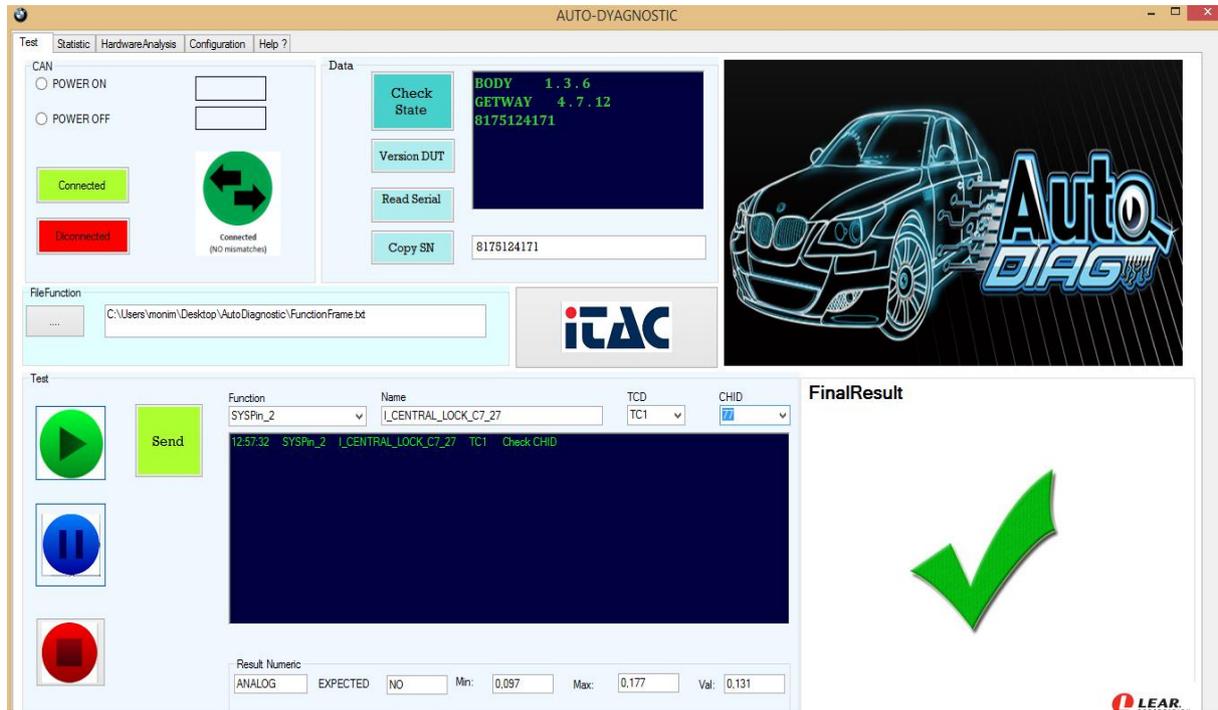


Figure 47 : Interface de test détaillé

✓ Exemple de test complet :

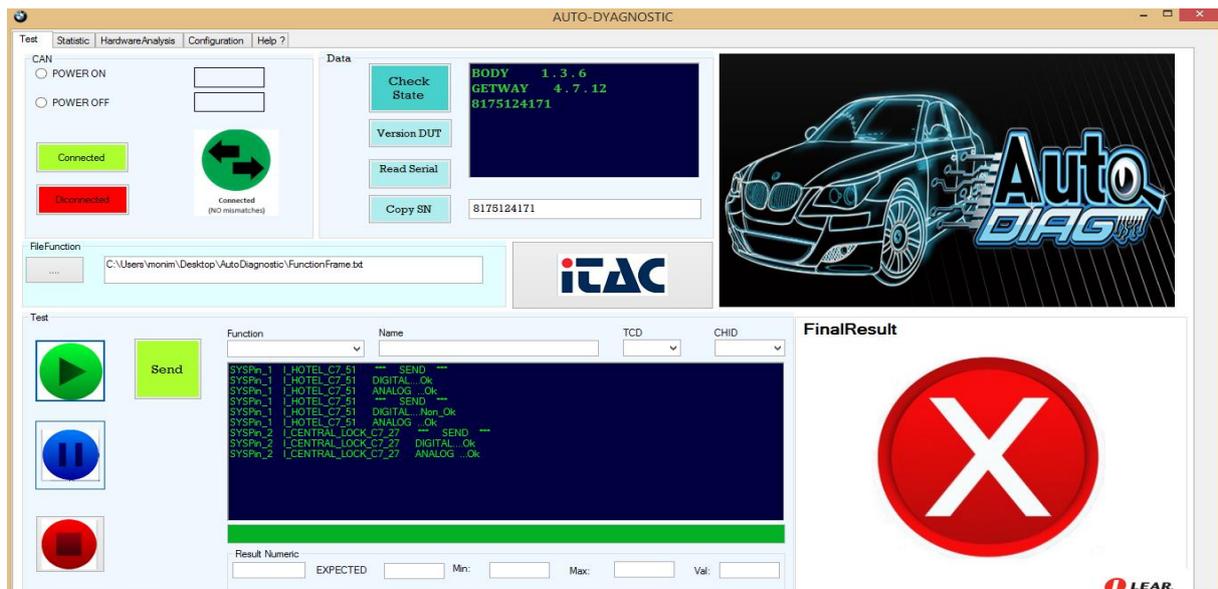


Figure 48 : Exemple de test complet

Conclusion

Nous avons réalisé, la conception générale du projet et nous avons précisé la conception de deux tests par les trois diagrammes de l'UML. Cette dernière, nous a aidés de développer notre application d'une manière efficace afin de mettre une interface très fiable.

Chapitre 5 : Etude technico-économique

Le « Rework » est une station réservée pour analyser manuellement les pièces défectueuses parvenues de l'assemblage électronique qui n'ont pas pu passer le test électrique ou le test visuel. Notre application est dédiée pour cette station pour l'objectif de vérifier le fonctionnement des cartes PL7 (FEM). Lorsque la carte ne fonctionne pas correctement alors l'application va trouver le défaut d'une manière automatique. La détection rapide des anomalies et l'identification du défaut est un élément clé de notre application.

5.1 Les dépenses actuelles

Actuellement, la station connaît plusieurs problèmes sont,

- perte du temps d'analyse pour une seule pièce,
- probabilité de commettre des erreurs dans les calculs,
- mal jugement de la précision sur le défaut,
- l'utilisation manuelle du copier le « numéro de série », ça provoque la traçabilité de la pièce.



Figure 49: Station de test

➤ Le coût actuel d'analyse des cartes sur la station Rework:

Le temps d'analyse d'une carte	Equipe hardware	Le nombre des cartes par semaine	Le cout par semaine
1 heure	20 €	60 pièces	1200 €

Tableau 4 : Le coût actuel d'analyse sur la station Rework

5.2 La pertinence du projet

1. Minimiser le taux des déclarations de faux défaut.	Supposons que la machine FKT déclare un test Nok. A l'aide de notre nouvelle application a le refait dans la station Rework. L'application va déclarer automatiquement que le test est Ok. Ici, on constate qu'il y a un faux de défaut.
2. Evaluer la dispersion des mesures.	Représente la variabilité ou l'étendue des différentes valeurs.
3. Minimiser le temps d'analyse Rework.	Le fichier des fonctions Nok permet d'analyser les fonctions Nok d'une manière détaillée et plus rapide.
4. Rendre l'analyse plus efficace.	L'interface graphique permet d'afficher les valeurs et le résultat de test pour meilleure efficacité.
5. Supporter l'analyse FA des KM0 et retour client.	L'interface permet d'afficher les résultats sous forme des symboles FEM, qui supporte l'analyse FA.
6. Etude statistique du processus (évolution vers Cpk carte).	Permet d'avoir la qualité de mesures.
7. Possibilité d'analyse fonction par fonction.	C'est le test détaillé.
8. Lecture des résultats sous format décimal ainsi (affichage).	Affichage des résultats décimaux.
9. Sauvegarde des rapports d'analyse.	Permet de suivre l'analyse sous forme des rapports qui comportent toutes les informations du test.

Tableau 5 : La pertinence du projet

- Le coût d'analyse par notre application,
 - ✓ La carte est analysée entre 5 et 10 minute,
 - ✓ Le coût de notre application d'analyse par semaine : 200 €.
- Enfin, grâce à notre projet nous avons minimisé le coût de cette analyse par 1000 €.

Conclusion

D'après cette étude technico – économique, nous constatons que notre système a aidé la société de minimiser les pertes financière ainsi que les pertes du temps.

Conclusion générale

Au court du stage au sein de la société « Lear Corporation- Technopolis, Rabat », nous avons mis en pratique les différentes connaissances acquises durant notre formation. De plus, nous sommes confrontés aux difficultés réelles du monde du travail et du management d'équipes. Sur le plan technique, notre projet consisté à concevoir une application de test automatisé et ainsi une interface de commande et d'acquisition des données via le logiciel Visual Basic.

Concernant la partie conception et suivant le cahier des charges imposé par la société, nous avons défini les différents blocs nécessaires de l'application de diagnostic. Cette application nous a permis de réduire considérablement le temps de mesure et d'améliorer la fiabilité d'analyse. Ensuite, nous avons réalisé une Interface graphique pour afficher les informations aux utilisateurs ayant effectués les tests ainsi que les résultats de mesures. Cette opération a aussi pour objectif de garder la traçabilité des tests avec les détails des mesures concernant la carte électronique PL7 FEM produite par la société.

Dans la réalisation de cette partie, nous avons suivi une démarche qui consiste à satisfaire le cahier des charges en proposant une solution intégrée optimale. Effectivement, l'entreprise peut profiter de cet axe du projet pour généraliser l'interface diagnostic pour les différents produits.

Le travail a demandé un effort de réflexion et de recherche pour pouvoir réaliser ce projet, la chose qui nous a permis par la suite d'acquérir une double compétence dont la technique et la culture de bonnes pratiques et de découvrir de nouvelles technologies et de les approfondir dans un monde qui ne cesse d'accroître jour après jour.

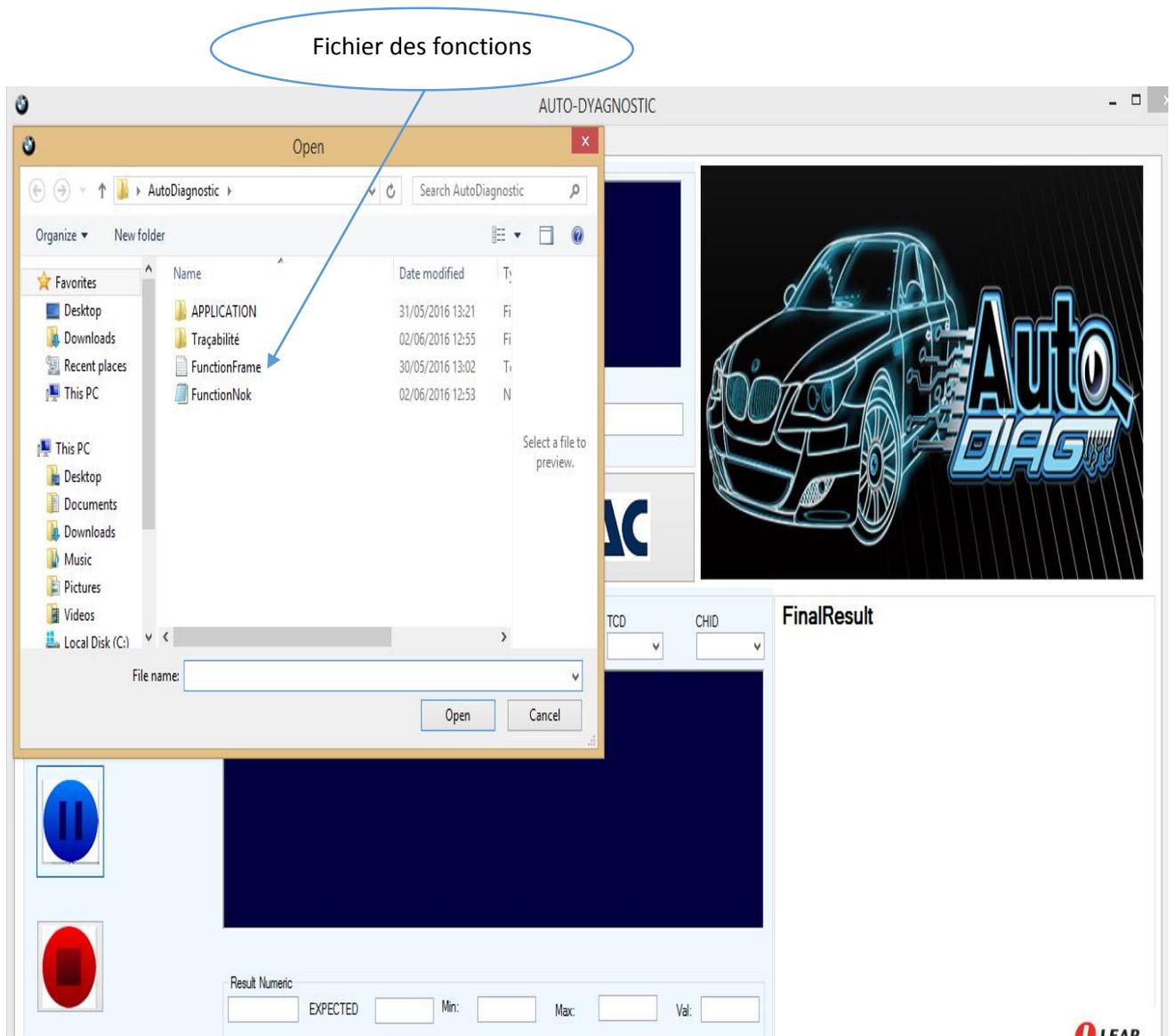
En fin, comme perspective nous savons que notre application a pour objectif de la fiabilité du test. Alors pour améliorer notre travail, nous proposons de réaliser un circuit intégré à base d'ARDUINO qui rassemble le protocole de communication avec le testeur manuel.

Webographie

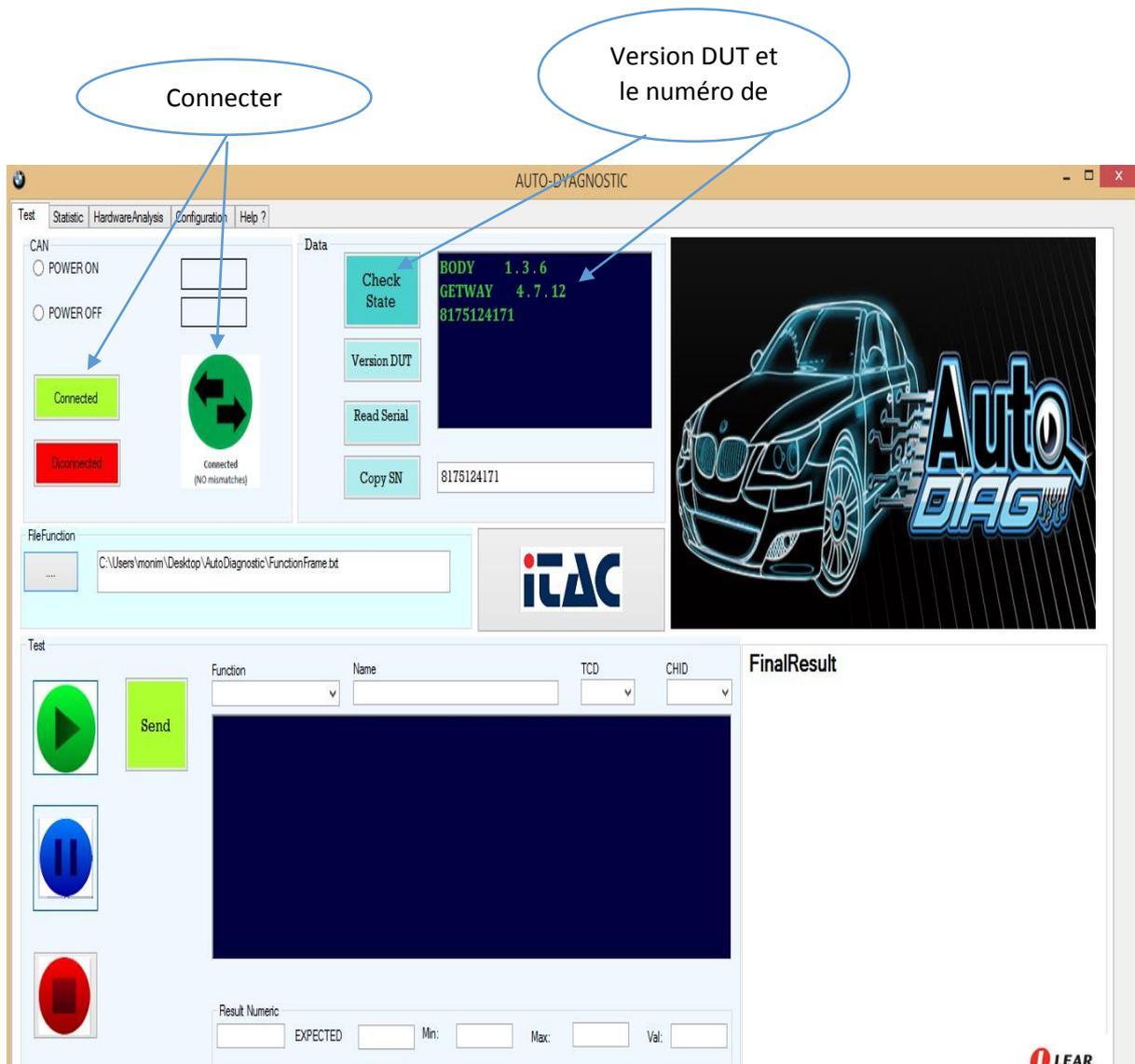
- [1] <http://www.technologuepro.com/cours-genie-electrique/cours-26-test-reparation-cartes-electroniques/> dernière visite le 20/03/2016 (page19).
- [2] <http://www.dek.com/> dernière visite le 26/03/2016 (page19).
- [3] <http://www.actutem.com/voitures-connectees-les-nouveaux-defis-du-test-electronique/> ; dernière visite le 26/04/2016 (page30).
- [4] <http://www.automobile-propre.com/voitures/bmw-i8/>; dernière visite le 01/05/2016(page31).
- [5] <http://mhhauto.com/Thread-Front-Electronic-Module-FEM-GW> ; dernière visite le 15/04/2016 (page32).
- [6] <http://www.freescale.com> ; dernière visite 25/04/2016 page(34).
- [7] <http://www.oberle.org/can-can.html> ; dernière visite le 16/05/2016 (page 42).
- [8] https://vector.com/vi_vt-system_en.html ; dernière visite 25/04/2016 (page48).
- [9] <http://chohmann.free.fr/qualite/cp.htm> ; dernière visite le 20/05/2016 (page56).
- [10] <http://www.fr.openclassrooms.com> ; dernière visite 25/04/2016.
- [11] https://fr.wikipedia.org/wiki/Framework_.NET dernière visite le 12/05/2016 (page 77).
- [12] <http://www.msdn.microsoft.com> ; dernière visite 25/05/2016 (page77).
- [13] <https://openclassrooms.com/courses/apprenez-a-programmer-en-vb-net/les-threads-5>; dernière visite 23/05/2016.
- [14] <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/mieux-structurer-son-code-le-pattern-mvc> ; dernière visite le 31/05/2016 (page 79).

Annexes

Annexe A : Le fichier des fonctions



Annexe B: Connecter

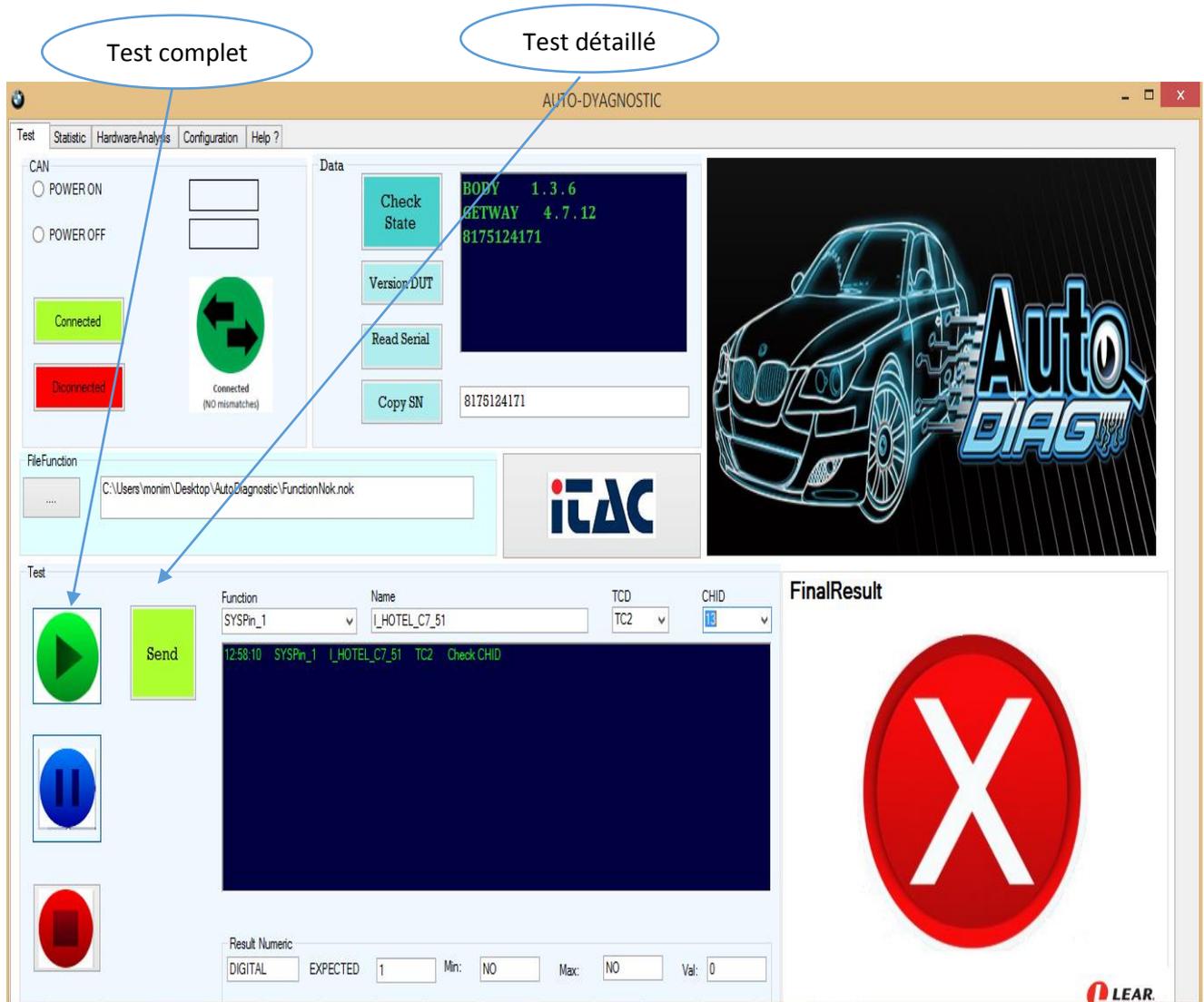


The screenshot shows the 'AUTO-DYAGNOSTIC' software interface. On the left, there are controls for CAN power (POWER ON/OFF) and a 'Connected' button. A callout bubble labeled 'Connecter' points to this button. In the center, there is a 'Data' panel with a 'Check State' button and a text area displaying:


```
BODY 1.3.6
    GETWAY 4.7.12
    8175124171
```

 A callout bubble labeled 'Version DUT et le numéro de' points to this text. Below the data panel are buttons for 'Version DUT', 'Read Serial', and 'Copy SN'. The 'Copy SN' button has the value '8175124171' next to it. On the right, there is a graphic of a car with the 'Auto-DIAG' logo. At the bottom, there is a 'Test' section with a 'Send' button and a 'FinalResult' area. The 'Result Numeric' section includes fields for 'EXPECTED', 'Min', 'Max', and 'Val'.

Annexe C : Lancer les tests :



The screenshot shows the 'AUTO-DYAGNOSTIC' software interface. At the top, there are menu options: Test, Statistic, Hardware Analysis, Configuration, and Help. The interface is divided into several sections:

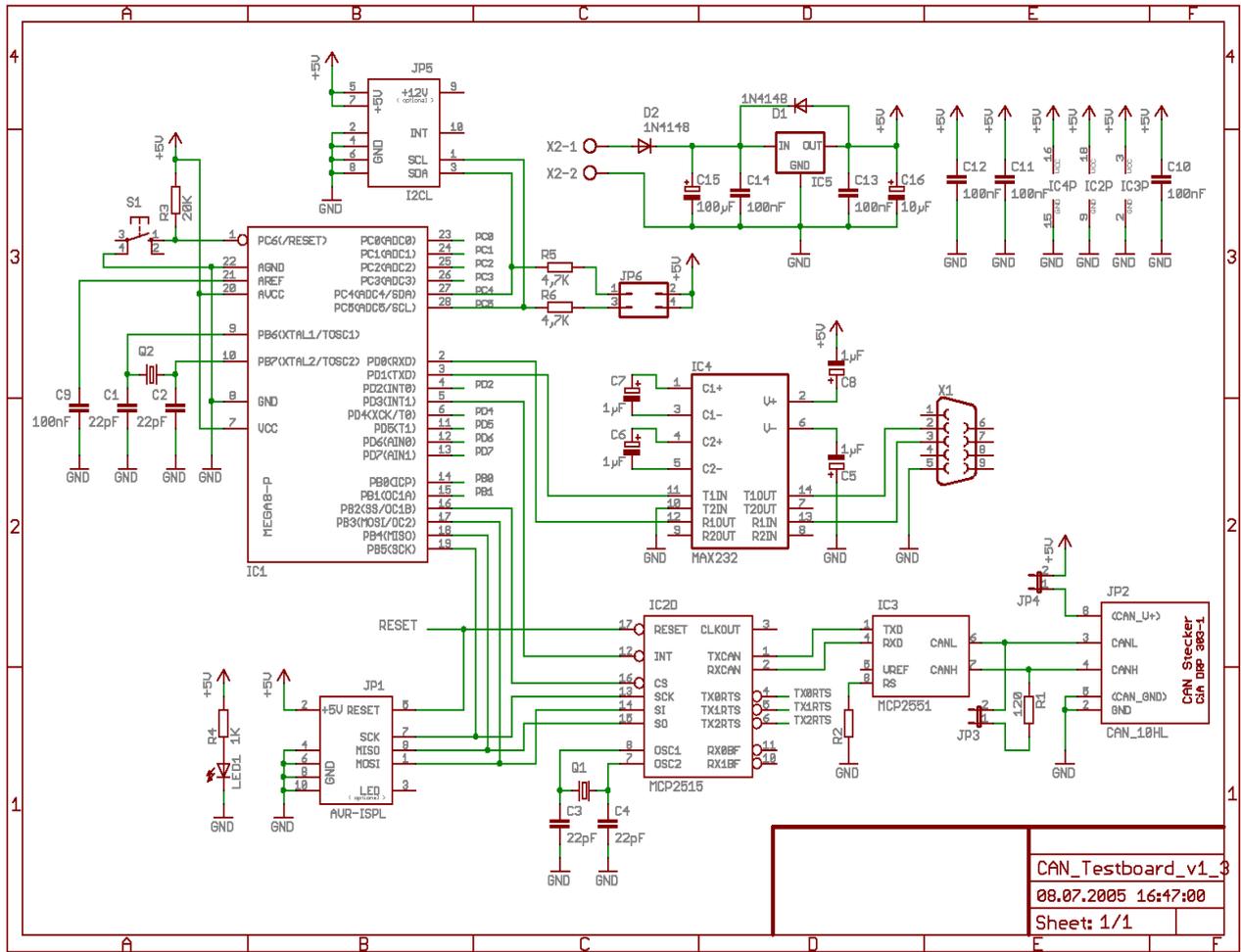
- Left Panel:** Contains CAN control options (POWER ON, POWER OFF), connection status (Connected, Disconnected), and a FileFunction field with the path 'C:\Users\monim\Desktop\AutoDiagnostic\FunctionNok.nok'.
- Center Panel:** Features a 'Data' window with a 'Check State' button and a terminal window displaying:


```
BOPY 1.3.6
GETWAY 4.7.12
8175124171
```

 Below this is a 'Version DUT' field, a 'Read Serial' button, and a 'Copy SN' button with the value '8175124171'.
- Right Panel:** Displays a graphic of a car with the 'Auto DIAG' logo and the 'itAC' logo.
- Bottom Section:** Includes a 'Test' area with a green play button (labeled 'Test complet'), a blue pause button, and a red stop button. A 'Send' button is also present. Below these are dropdown menus for 'Function' (SYSPin_1), 'Name' (L_HOTEL_C7_51), 'TCD' (TC2), and 'CHID' (18). A terminal window shows the command: '12:58:10 SYSPin_1 L_HOTEL_C7_51 TC2 Check CHID'. Below the terminal is a 'Result Numeric' section with fields for 'DIGITAL', 'EXPECTED' (1), 'Min' (NO), 'Max' (NO), and 'Val' (0).
- FinalResult:** A large red oval with a white 'X' indicates a failed test.

Two blue circles with arrows point to the green play button and the 'Test complet' label, and another blue circle with an arrow points to the 'Check State' button and the 'Test détaillé' label.

Annexe D : Schéma électrique de CAN :



Annexe E : Les données techniques de Cancasexl

Technical Data	
Area of application	Mobile, stationary
Microcontroller	ATMEL AT91 (ARM7 TDMI with 64 MHz)
CAN channels	Two independent channels (NXP SJA1000)
Identifier	11/29 bit
Transceiver (galvanic decoupling)	See CAN-, LIN-, J1708 piggybacks in bus transceiver overview
PC interface	USB 2.0 highspeed (USB 1.1 compatible)
Temperature range	Operation: -20..+70 °C Storage: -40..+85 °C
Driver libraries for	C++, C, .NET (C#, Visual Basic, Delphi)
Operating system	Windows XP/Vista (32 bit), Windows 7 (32/64 bit)
Error Frame/Remote Frame	Detection & Generation
Dimensions approx.	105 x 85 x 32 mm
Power consumption (typ.)	1 W
External power supply (optional)	7 V..33 V
Time stamp accuracy	1 µs
Average latency time in interface operating	300 µs