

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté de Technologie
Département Ingénierie des Systèmes Electriques

Mémoire de Master

Présenté par
Mr SAYAH Rayane Abdelaziz

Filière : Electronique
Spécialité : Electronique des Systèmes Embarqués

***Implémentation d'un système de sécurité
dans l'automobile***

Soutenu le 07/07/2022 devant le jury composé de :

HARRAR	Khaled	MCA	Université de Boumerdès	Président
BELKACEM	Samia	MCA	Université de Boumerdès	Rapporteur
MECHID	Samira	MAA	Université de Boumerdès	Examineur
AKLIOUAT	Hacene	MAA	Université de Boumerdès	Examineur

Remerciements

Je voudrais d'abord remercier ALLAH, de m'avoir guidé pendant toutes mes années d'étude et de m'avoir donné la volonté, la patience et le courage pour terminer mon travail, après mes parents qui quoi que je dise, je ne saurai pas capable de les remercier comme il se doit.

Je remercie sincèrement mon encadreur, Dr Samia BELKACEM, pour m'avoir encouragé à réaliser ce travail, son soutien, sa disponibilité et ses conseils hors de l'ordinaire.

Je remercie aussi les membres de jury, Dr HARRAR Khaled président de jury, Mme MECHID Samira et Mr AKLIOUAT Hacene, d'accepter d'évaluer mon travail.

Je remercie les membres du club scientifique Electro avec lesquels j'ai passé un peu de temps, où j'ai travaillé avec des membres passionnés, spécialisés dans le domaine de la robotique et l'automatique et d'autres spécialités, ça me fait un grand plaisir !

Ils ont été plus que des collègues, et je les ai côtoyés avec plaisir.

Dédicace

*Je dédie ce travail à mes très chers parents,
vous n'avez cessé de me soutenir et de
m'encourager avec votre tendresse et affection
tout au long de mon parcours.*

*À mes chères sœurs, à ma grande famille
et à tous mes amis.*

SAYAH Rayane Abdelaziz

Liste des abréviations

ABS:	Anti-lock braking system
ADAS:	Advanced driver assistance systems
CNN:	Convolutional Neural Network
CPU:	Central Processing Unit
GPS:	The Global Positioning System
GNSS :	Géolocalisation et Navigation par un Système de Satellites
IOT:	Internet of Things (IoT)
NMEA:	National Marine Electronics Association
NTHSA:	National Highway Traffic Safety Administration
PPP:	Precise Point Positioning
PCB:	printed circuit board
RTK:	Real time Kinematic
RTOS:	Real Time Operating System
STI :	Systèmes de Transport Intelligents
V2V:	Vehicle to vehicle
V2P:	Vehicle to pedestrians
V2I:	Vehicle-to-infrastructure

تصميم وتطبيق نظام مدمج في السيارات هو موضوع هذا العمل. الهدف من هذا النظام هو سلامة السيارة وكذلك السائق ينقسم نظام السلامة هذا إلى ثلاثه أجزاء:

(1) نظام الحماية من الحرائق، (2) نظام التتبع في الوقت الفعلي، (3) نظام لفتح الباب مع التعرف على الوجه. ويستطيع النظام إطفاء حريق من خلال أجهزة استشعار أو كاميرات باستخدام الذكاء الاصطناعي في الوقت الفعلي، وكذلك إدارة عدد من المكالمات الهاتفية عن بعد لإعلام المديرين بجميع أنواعهم، من خلال الرسائل الرقمية المخزنة على بطاقة الذاكرة. يجعل من الممكن تحديد موقع السيارة في أي مكان في العالم باستخدام واجهة متصلة وأيضا موقع ويب. يتم إجراء نموذج أولي في كل من جوانب الأجهزة والبرمجيات. أعطى هذا النظام أداء جيدا ومعدل أخطاء منخفض للغاية مع الحفاظ على تكلفة تصنيع معقولة جدا.

الكلمات المفتاحية: الأمان، التحكم في الوصول، مكافحة الحرائق، الاتصال المدمج، المتحكم الدقيق، الرسائل، الذاتية.

Abstract

The design and realization of an embedded system in cars are the subject of this work. The objective of this system is the safety of the car as well as the driver this safety system is divided into three parts: 1) Fire protection system, 2) Real-time navigation system, 3) Unlock system with facial recognition. The system is able to extinguish a fire through sensors or cameras using artificial intelligence in real time, and also to manage a number of phone calls remotely to inform managers, through digitized messages stored on a memory card. And also, the system can locate the car anywhere in the world using a connected interface and website. A prototype is made in both hardware and software aspects. This system gave a good performance and a very reduced rate of errors while keeping a very affordable manufacturing cost.

Keywords: Security, Access control, Embedded connectivity, Microcontroller, Messaging, Autonomy.

Résumé

La conception et la réalisation d'un système embarqué dans l'automobile font l'objet de ce travail. L'objectif de ce système est la sécurité de l'automobile ainsi que le conducteur ce système de sécurité est divisé en trois parties : 1) Système anti-incendie, 2) Système de navigation en temps réel, 3) Système de déverrouillage avec la reconnaissance faciale. Le système est capable d'éteindre un incendie à travers des capteurs ou des caméras à l'aide de l'intelligence artificielle en temps réel, et aussi de gérer un certain nombre d'appels téléphoniques à distance pour informer les responsables de tous genres, à travers des messages stockés sur une carte mémoire. Il permet de localiser la voiture partout dans le monde en utilisant une interface connectée et aussi un site web. Un prototype est réalisé dans ses deux aspects matériels et logiciels. Ce système a donné une bonne performance et un taux très réduit d'erreurs tout en gardant un coût de fabrication très abordable.

Mots-clés : Sécurité, Contrôle d'accès, Anti incendie, Connectivité embarquée, Microcontrôleur, Messagerie, Autonomie.

Table des matières

Remerciements	ii
Dédicace	iii
Liste des abréviations	iv
ملخص	v
Abstract	v
Résumé	v
Table des matières	vi
Liste des figures	ix
Liste des tableaux	xi
Introduction générale	1
Chapitre 1 Introduction aux systèmes et connectivité embarquée dans l'automobile	3
1. Introduction	3
2. La sécurité	3
3. Système embarqués (Embedded System)	3
4. Qu'est-ce réellement qu'un système embarqué ?	4
5. Architecture de système embarqué	4
6. Problèmes de sûreté et de sécurité pour les systèmes embarqués	5
7. Conception des systèmes embarqués	6
7.1. Procédure de conception	6
7.2. Niveaux d'abstraction	7
8. Etapes de conception	7
8.1. Besoin (ou Exigences)	7
8.2. Spécification	8
8.3. Conception architecturale	9
8.4. Conception de composants matériels et logiciels	9
8.5. Intégration du système	9
9. Conception de circuits imprimés	9
9.1. Qu'est-ce que la conception de circuits imprimés ?	9
9.3. Routage de circuits imprimés	11
9.4. Exporté le fichier GBR	12
9.5. Intégration de la conception électronique & Mécanique	13

10.	Véhicule connectée.....	14
10.1.	Positionnement	14
10.2.	Interface Homme-machine	14
10.3.	Un système de gestion des informations d'identification de sécurité pour les communications V2X.....	16
10.4.	V2V Sécurité Véhicule Communication	16
11.	L'intelligence Artificielle & les véhicules autonomes.....	18
11.1.	Systèmes informatiques pour la conduite autonome	18
11.2.	Applications de conduite autonome.....	18
11.2.1.	La détection d'objets.....	18
11.2.2.	La détection de voie.....	19
12.	Conclusion.....	20
	Chapitre 2 : Implémentation électronique.....	21
1.	Introduction	21
2.	Qu'est-ce qu'un microcontrôleur ?	21
3.	Le microcontrôleur AVR.....	22
4.	Le microcontrôleur ARM.....	22
4.1.	Les meilleures cartes de développement ARM	23
5.	Système de base.....	24
6.	Implémentation d'un système de navigation	25
6.1.	Placement des composants pour le système de navigation.....	25
6.2.	Routage du circuit de système navigation	26
6.3.	Implémentation du système anti incendie.....	28
6.4.	Placement des composants pour le système anti incendie	28
6.5.	Routage circuit de système anti incendie.....	29
7.	Le circuit imprimé du system	31
7.1.	Placement des composants pour le système global	31
7.2.	Routage de circuit global	32
7.3.	Présentation du circuit global	33
7.4.	Présentation 3D du circuit global	34
7.5.	Intégration de la conception électronique et mécanique.....	34
8.	Conclusion.....	35
	Chapitre 3 Résultats & Discussion	36
1.	Introduction	36

2.	Interface home machine	36
3.	Système anti incendie	37
3.1.	Système anti incendie à base d'un capteur	38
3.2.	Présentation 3D du circuit de système anti incendie	39
4.	Système de navigation IOT (internet of things)	40
4.1.	Présentation 3D du circuit de navigation.....	42
5.	Logiciels requis	43
6.	Les packages importants	44
7.	Système de déverrouillage avec la reconnaissance faciale.....	45
7.1.	Test de la reconnaissance faciale.....	48
8.	Système de conduite autonome	49
8.1.	La détection automatique des voies et des objets.....	49
8.2.	Détection des voies.....	52
9.	Le système de sécurité.....	54
9.1.	Conception de circuit imprimé	54
9.2.	Conception des design 3D	54
10.	Realisation du prototype.....	55
11.	Conclusion.....	57
	Conclusion générale	56
	Bibliographie	60

Liste des figures

Figure 1. 1 : Système embarqué (Embedded System)	4
Figure 1. 2 : Architecture des systèmes embarqués.....	5
Figure 1. 3 : Les niveaux d'abstraction	8
Figure 1. 4 : Etapes de Fabrication d'un PCB	10
Figure 1. 5 : CPU NVIDIA pour la conduite autonome [7]	10
Figure 1. 6 : Placement des composants.....	11
Figure 1. 7 : ROUTAGE DE CIRCUITS IMPRIMÉS.....	12
Figure 1. 8 : Gerber.....	12
Figure 1. 9 : PCB fabriqué.....	13
Figure 1. 10 INTÉGRATION DE LA CONCEPTION ÉLECTRONIQUE ET MÉCANIQUE [8]	13
Figure 1. 11 : véhicule connecté Interface Homme-machine	15
Figure 1. 12 : Système de communication.....	16
Figure 1. 13: Communications entre les véhicules connectés	17
Figure 1. 14 :Détection d'objets [7]	19
Figure 1. 15 : Détection des voies.....	20
Figure 2. 1 : Architecture de Uc AVR.....	22
Figure 2. 2 : Architecture Uc ARM.....	23
Figure 2. 3 : Qu'elle que Uc ARM.....	23
Figure 2. 4 : Block diagramme	24
Figure 2. 5 : Raspberry pi + camera pi	24
Figure 2. 6: Placement des composants pour le système de navigation.....	25
Figure 2. 7 : Circuit de navigation sur plaque de bakélite	25
Figure 2. 8 : Routage circuit de système navigation.....	26
Figure 2. 9 : ESP 32 WIFI + BLU	27
Figure 2. 10 : GPS module	27
Figure 2. 11 : Placement des composants pour le système anti incendie.....	28
Figure 2. 12 : Circuit de système anti incendie sur plaque d'essai.....	28
Figure 2. 13 : : Routage circuit de système anti incendie	29
Figure 2. 14 : Microcontrôleur ATmega328.....	29
Figure 2. 15 : Actionneur Linéaire	30
Figure 2. 16 : Placement des composants pour le système global	31
Figure 2. 17 : Circuit de système global	31
Figure 2. 18 : Routage de circuit global.....	32
Figure 2. 19 : Circuit global	32
Figure 2. 20 : Présentation 3D du circuit global.....	33
Figure 2. 21 : Présentation 3D électronique et mécanique	34

Figure 3. 1 Interface home machine	36
Figure 3. 2 : Communication IHM et le système	37
Figure 3. 3 : Utilisation du système	37
Figure 3. 4 : Principe de fonctionnement système anti incendie	38
Figure 3. 5 : Détection de la flamme avec AI.....	39
Figure 3. 6 : Présentation 3D du circuit (système anti incendie).....	39
Figure 3. 7 : Principe de fonctionnement GPS	40
Figure 3. 8 : Résultats de localisation	41
Figure 3. 9 : Présentation 3D du circuit (système de navigation).....	42
Figure 3. 10 : Commande window.....	45
Figure 3. 11 : Les étapes de la reconnaissance faciale	45
Figure 3. 12 : Création fichier face recog	46
Figure 3. 13: Fichier qui contient les programmes importants	46
Figure 3. 14 : Data base	47
Figure 3. 15 : L'apprentissage supervisé	47
Figure 3. 16 : Test de l'étape de la reconnaissance faciale	48
Figure 3. 17 : Supervised fait partie du machine Learning et AI	49
Figure 3. 18 : La détection automatique des objets	49
Figure 3. 19 : Architecture CNN	50
Figure 3. 20 :Tensorflow & keras.....	50
Figure 3. 21 : Système d'exploitation linux	51
Figure 3. 22 : Etapes à suivre sur linux pour la conduite autonome	52
Figure 3. 23 : Sommation des pixels	53
Figure 3. 24: Principe de fonctionnement du conduit autonome	53
Figure 3. 25 : Platform IO +vs Code c'est l'équivalent d'ide Arduino.....	54
Figure 3. 26 : Conception PCB & 3D Design.....	55
Figure 3. 27 : LED stop de véhicule	55
Figure 3. 28 : Montage du système sur prototype.....	56

Liste des tableaux

Tableau 2. 1: La différence entre ARM & AVR	21
Tableau 2. 2 : Connexion entre les pins d'ESP32 et GPS module	26

Introduction générale

Système embarqué est un système électronique et informatique autonome dédié à une tâche précise. Il fonctionne avec une précision, il ressemble souvent à une carte rectangulaire avec fond vert, truffé de composants de toutes les formes et de petits fils.

Le système embarqué est à la fois un matériel et un logiciel utilisé en informatique et en électronique, avec des applications dans de nombreux domaines.

Logé à l'intérieur d'une machine, d'un appareil ou d'un véhicule, il se voit assigner une tâche précise qu'il se doit d'exécuter. Il s'agit d'un système autonome qui doit réaliser son travail en tenant compte de plusieurs contraintes : sa taille, le temps d'exécution dont il dispose et l'énergie.

La sécurité à une importance primordiale sur tout dans les véhicules, La voiture connectée contribuer également à plus de confort et de sécurité grâce au réseau de communication sans fil le véhicule est relié à des serveurs ou il peut aller puiser des informations toujours à jours qui serrant utile pour le conducteur [1].

La connectivité mondiale croissante, les exigences variables des consommateurs en matière de disponibilité constante des données, et les environnements d'entreprise équipés de capteurs constamment opérationnels entraînent la création, la collecte et l'analyse de quantités exponentielles de données. D'ici 2025, IDC estime qu'il y aura 79,41 zettaoctets de données créées et que près de 30 % deux d'entre elles devront être traitées en temps réel par des systèmes en temps réel [2].

L'arrivé des systèmes embarqué dans l'industrie automobile, les véhicule connecté et autonome ont devenue tendance.

Plusieurs compagnies automobiles travaillent sur le concept d'alimenter la prochaine génération de véhicules autonomes. Par exemple Parker offre jusqu'à 1,5 téraflops de performances pour les systèmes de cockpit d'IA autonome basés sur Deep Learning et offre les performances graphiques brutes et les fonctionnalités nécessaires pour alimenter plusieurs écrans haute résolution, tels que les écrans d'instruments de cockpit et les panneaux d'info divertissement embarqués.

L'objectif de ce travail est de concevoir et réaliser un dispositif de sécurité embarqué dans l'automobile basé sur l'intelligence du système dans une carte électronique.

- Système de sécurité à base des capteurs :
 1. Anti incendie
 2. Navigation et info divertissement
- Système de sécurité à base l'intelligence artificiel et vision par ordinateur :
 1. Détection de flamme
 2. La reconnaissance facile
 3. Conduite autonome

Le mémoire se présente sous quatre chapitres :

Dans le premier chapitre on donne une introduction aux systèmes et connectivité embarqué dans l'automobile en générale.

Le deuxième chapitre sera consacré à la conception électronique et l'implémentation électronique.

Le troisième chapitre est dédié à la présentation des résultats et les tests avec discussions.

Un quatrième chapitre est dédié à l'implémentation logiciel.

Enfin une conclusion générale est présentée.

Chapitre 1 Introduction aux systèmes et connectivité embarquée dans l'automobile

1. Introduction

La prochaine génération d'applications de sécurité des véhicules reposera sur la connectivité. Il est également largement admis que la connectivité sera une technologie habilitante clé pour la conduite autonome. Cette connectivité peut inclure véhicule à véhicule (**V2V**), Communications véhicule à infrastructure (**V2I**) et véhicule à piéton (**V2P**).

Dans ce chapitre on va décrire le système embarqué, les véhicules connectés et l'intelligence artificielle dans les véhicules autonomes.

2. La sécurité

La sécurité est la capacité d'une entité à protéger les ressources dont elle porte Responsabilité de protection.

Dans un système embarqué, cette responsabilité de protection peut s'appliquer aux ressources à l'intérieur ou aux ressources du système global auquel le système embarqué est connecté ou dans lequel il est subsumé. Comme nous le verrons plus loin dans ce chapitre, les propriétés protectrices d'un composant ou d'un système sont incorporées dans son politique de sécurité.

3. Système embarqués (Embedded System)

Un système embarqué est un produit électronique qui contient un microprocesseur (Un ou plusieurs) et un logiciel pour exécuter une fonction constitutive au sein d'une entité plus grande. [3]

Aujourd'hui, l'électronique embarquée est de plus en plus utilisée dans notre quotidien et dans de multiples domaines d'applications :

- Automobile
- L'aviation
- Télécommunication

4. Qu'est-ce réellement qu'un système embarqué ?

Il s'agit d'un système électronique et informatique autonome intégré à un objet qui est conçu pour résoudre un problème ou une tâche spécifique en utilisant un microprocesseur combiné avec du matériel électronique. La partie logicielle d'un système embarqué est entièrement programmée dans la partie matérielle, ce qui les rend indissociables l'une de l'autre contrairement à un environnement de travail classique de type PC [3].



Figure 1. 1 : Système embarqué (Embedded System)

5. Architecture de système embarqué

Les systèmes embarqués se composent de matériel et de logiciels. Ils sont souvent utilisés dans des applications qui nécessitent à la fois de détecter quelque chose de physique dans l'environnement et de contrôler quelque chose en réponse.

Un exemple simple est un système CVC avec un contrôleur de thermostat intégré.

De tels systèmes sont souvent appelés «**temps réel**» ou «**événementiel**» car ils doivent réagir en temps opportun aux événements du monde réel. Selon le type et la criticité de l'application, il est important de réagir aux événements à un moment précis. Les systèmes critiques pour la sécurité comme les freins antiblocage (ABS) dans une voiture doivent réagir en quelques millisecondes. De tels systèmes sont appelés «**temps réel dur**», ce qui signifie qu'ils doivent respecter les délais requis ou qu'une défaillance s'est produite. Les systèmes avec des délais plus souples sont appelés «**temps réel souple**». Dans la plupart de ces cas, ces systèmes fonctionnent sur une sorte de système d'exploitation en temps réel (RTOS) [4].

Voici un exemple simple d'architecture de système embarqué :

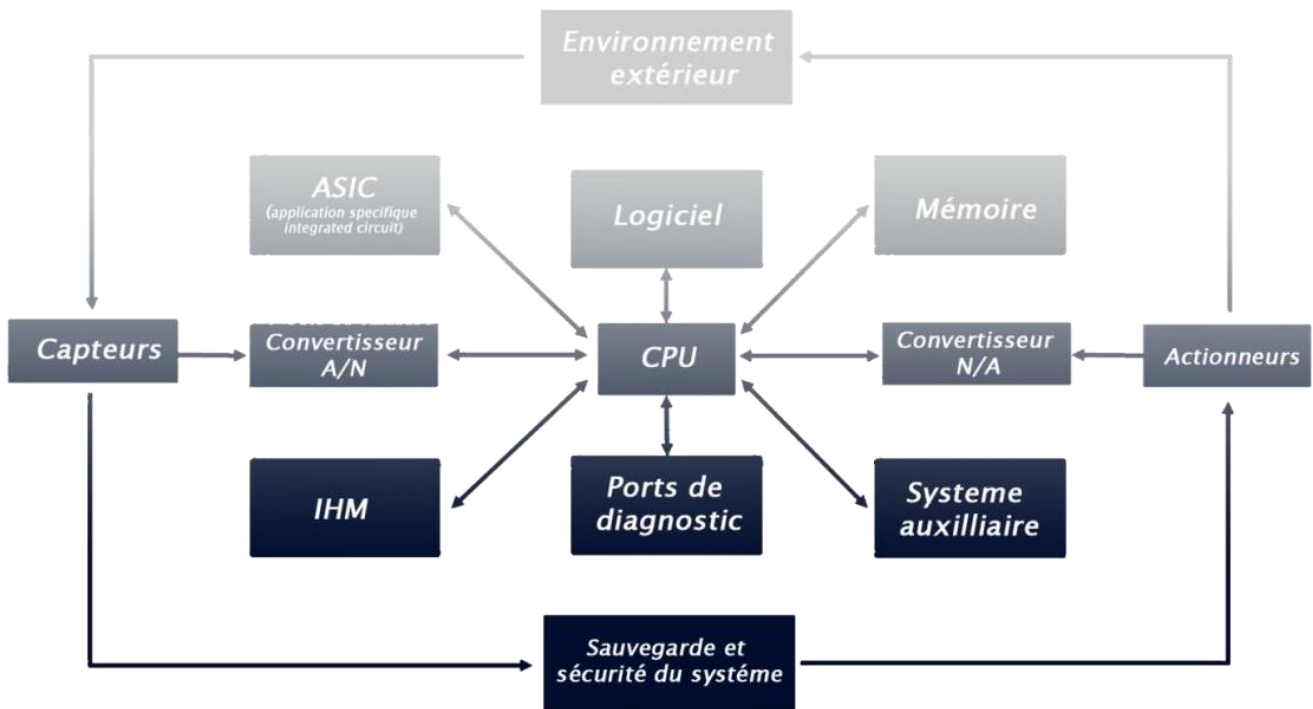


Figure 1. 2 : Architecture des systèmes embarqués

6. Problèmes de sûreté et de sécurité pour les systèmes embarqués

Les systèmes embarqués jouent un rôle important dans les dispositifs critiques pour la sécurité et la sûreté. Les produits tels que les stimulateurs cardiaques et les contrôleurs de frein ABS ne peuvent pas échouer. Des vies dépendent du bon fonctionnement de ces appareils, tout le temps. Des échecs très médiatisés.

La sécurité est désormais une préoccupation majeure à mesure que de plus en plus de systèmes embarqués sont mis en ligne, connectés via Internet (Internet des objets). Cette interconnexion crée de nouvelles fonctionnalités et opportunités et ouvre la porte aux attaques. Les appareils non sécurisés ne sont pas sûrs, la sécurité et la sûreté doivent donc être considérées de la même manière dans ces appareils critiques [4].

6.1. Comment rendre les systèmes embarqués sûrs et sécurisés ?

La seule façon de s'assurer que les systèmes embarqués sont sûrs et sécurisés est de vers les tests. Plus précisément, par une rigueur vérification et la validation tout au long du cycle de

développement. Cela inclut avant le développement du logiciel, au début de l'analyse des besoins et jusqu'à la fin de la vie.

Il existe des normes spécifiques qui doivent être suivies par les industries où la sécurité est une préoccupation majeure, notamment les suivantes :

- Automobile (ISO 26262)
- Aviation (DO-178C)
- Équipement médical (IEC 62304)
- L'automatisation industrielle (IEC 61508)
- Chemin de fer (EN 50128)

Les fabricants de ces appareils doivent suivre les normes et prouver par des audits qu'ils font preuve de diligence raisonnable et ont répondu à toutes les préoccupations en matière de sécurité et de sûreté. Ce type de développement matériel et logiciel prend du temps et coûte cher à construire et à tester, mais il est essentiel pour garantir un comportement correct lors du déploiement [4].

7. Conception des systèmes embarqués

Les méthodes de conception actuelles des systèmes embarqués découlent d'une évolution des méthodologies de conception de deux mondes qui étaient relativement distincts jusqu'à tout récemment le monde du logiciel et le monde du matériel. Dans ces deux mondes, les techniques utilisées actuellement résultent d'élévations des niveaux d'abstraction des informations à manipuler. Cela a permis de concevoir des systèmes de plus en plus complexes et performants.

7.1. Procédure de conception

Une procédure est nécessaire pour la conception (organisation)

La vue d'ensemble du processus de conception des systèmes embarqués vise les deux objectifs suivants :

- Tout d'abord, elle nous donnera une introduction aux différentes étapes de conception des systèmes embarqués avant de les approfondir plus en détail.

La compréhension de la méthodologie assure qu'on va respecter les paramètres importants.

- Deuxièmement, cela nous permettra d'examiner la méthodologie de conception elle-même.

7.2. Niveaux d'abstraction

A chaque niveau d'abstraction, on doit analyser le système pour déterminer ses caractéristiques actuelles et l'améliorer pour prendre en compte les détails manquants.

Top-down Design (conception descendante)

On part du plus haut niveau d'abstraction et on descend vers le plus détaillé (we start with abstraction of the system and then the details are created).

Bottom-up Design (conception ascendante)

On part des composants de base et on remonte vers le système (we start with components to build a system)

La technique Top-down est plus utilisée pour concevoir de nouveaux systèmes.

La technique Bottom-up est utilisée dans l'ingénierie inversée, par exemple lorsqu'on essaie de faire le design d'un système qui existe déjà.

Nous avons besoin de la technique Bottom-up parce que nous n'avons pas une idée parfaite de la façon dont les étapes ultérieures du processus de conception se dérouleront.

Une conception réaliste utilise les deux techniques.

8. Etapes de conception

Les étapes de conception sont détaillées ci-après :

8.1. Besoin (ou Exigences)

Les exigences peuvent être fonctionnelles ou non fonctionnelles. Nous devons saisir les fonctions de base du système embarqué, mais la description fonctionnelle n'est souvent pas suffisante.

Les exigences non fonctionnelles typiques incluent :

- Performance
- Coût
- Dimensions et poids physiques
- Consommation d'énergie
- Maquette

Sur la figure ci-dessous on donne les étapes nécessaires à la conception d'un système embarqué :

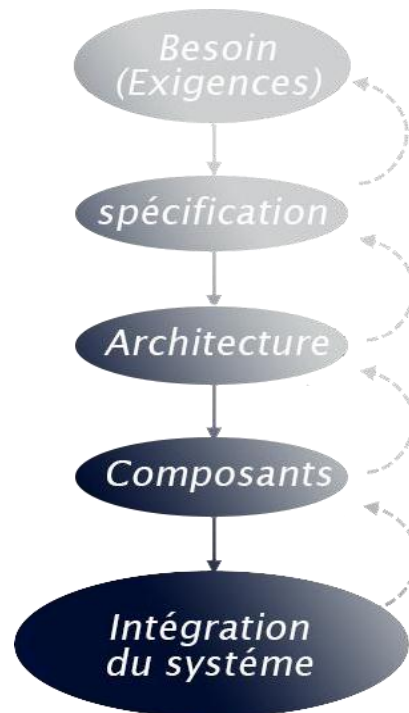


Figure 1. 3 : Les niveaux d'abstraction

Expression des besoins

Une description précise de ce que veut l'utilisateur (client) et de ce qu'il espère obtenir

- **Besoins fonctionnels**

Sortie en fonction des entrées et des paramètres

- **Besoins non-fonctionnels**

- Temps nécessaire pour calculer la sortie
- Taille, poids et forme
- Consommation
- Fiabilité
- Sécurité...

8.2. Spécification

Le cahier des charges est plus précis, il sert de contrat entre le client et les concepteurs. La spécification doit être suffisamment compréhensible pour que quelqu'un puisse vérifier qu'elle répond aux exigences du système et aux attentes globales du client. Il doit également être sans

ambiguïté. La spécification doit être rédigée avec soin afin qu'elle reflète fidèlement les exigences du client et le fasse d'une manière qui puisse être clairement suivie lors de la conception.

8.3. Conception architecturale

La spécification ne dit pas comment le système fait les choses, seulement ce que le système fait. L'objectif de l'architecture est de décrire comment le système implémente ces fonctions.

De nombreux détails de mise en œuvre du système doivent être expliqués en deux schémas fonctionnels : un pour le matériel et un autre pour le logiciel.

8.4. Conception de composants matériels et logiciels

L'effort de conception des composants construit ces composants conformément à l'architecture et aux spécifications. Les composants comprendront en général à la fois du matériel (μ c, FPGA, cartes, etc.) et des modules logiciels. Certains des composants seront prêts à l'emploi.

Le récepteur GPS est un bon exemple de composant spécialisé qui sera néanmoins un composant standard préconçu. Nous pouvons également utiliser des modules logiciels standards. Un bon exemple est la base de données topographique.

8.5. Intégration du système

Les composants construits sont assemblés pour voir comment le système fonctionne. Si nous ne déboguons que quelques modules à la fois, nous sommes plus susceptibles de découvrir les bogues (bugs) simples et de les reconnaître facilement. Ce n'est qu'en corrigeant tôt les bogues simples que nous pourrions découvrir les bogues les plus complexes ou les plus obscurs.

9. Conception de circuits imprimés

Dans cette section nous donnons les éléments essentiels à la conception de circuits imprimés.

9.1. Qu'est-ce que la conception de circuits imprimés ?

La conception de circuits imprimés permet de donner une forme physique à vos circuits électroniques. Grâce au logiciel de topologie, le processus de conception de circuits imprimés

combine le placement et le routage des composants pour définir la connectivité électrique sur un circuit fabriqué [5].

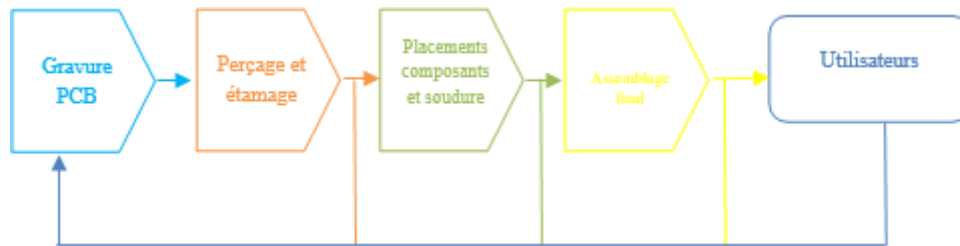


Figure 1. 4 : Etapes de Fabrication d'un PCB

Bref, tous ces éléments que nous observons plus facilement sur un CPU ou un GPU impactent aussi la construction de leurs cartes de support, ce qui oblige à obtenir des PCB de plus en plus fiables. Et ce sujet n'est pas si simple, car durant les différentes étapes de la conception d'une carte mère par exemple, le PCB passera entre les mains de plusieurs équipes et subira plusieurs tests, cependant les données et résultats de ces étapes n'étaient pas forcément liées il y a encore peu. Il faut dire que le besoin a changé avec le développement des applications plus sensibles comme le Big Data, l'automobile connectée ou les services en cloud, en passant par les IA et le machine Learning [6].

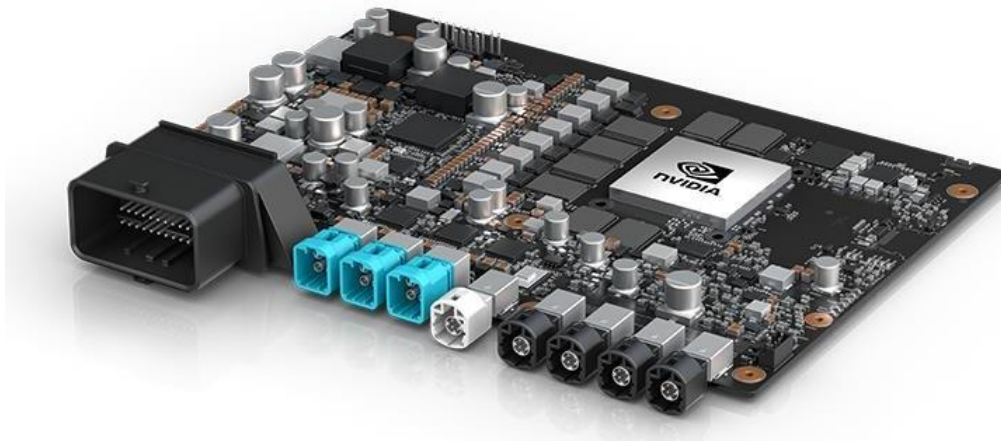


Figure 1. 5 : CPU NVIDIA pour la conduite autonome [7]

Conception circuits imprimés-PCB

Les défis technologiques ont forgé l'expérience des concepteurs de plaquettes de circuits imprimés (PCB). [8]

- Conception pcb multicouches (jusqu'à 32 couches)
- Technologie HDI

- Utilisation de divers matériaux
- Conception pcb flexible et rigidflex
- Circuits à très faible bruit pour des applications capables de répondre aux besoins en imagerie médicale
- Traçage de signaux différentiels haute fréquence (mémoire DDR3, multigigabit)
- Interfaçage complexe de type die bonding sur circuits imprimés
- Conception de circuit pouvant aller à des températures cryogéniques

9.2. PLACEMENT DES COMPOSANTS

Le processus de conception de circuits imprimés commence par la schématique ou on crée le schéma de notre projet. Suivant, on passe à l'étape qui consiste à placer les composants dans un espace très limité, on doit être créative pour résoudre les problèmes [5].

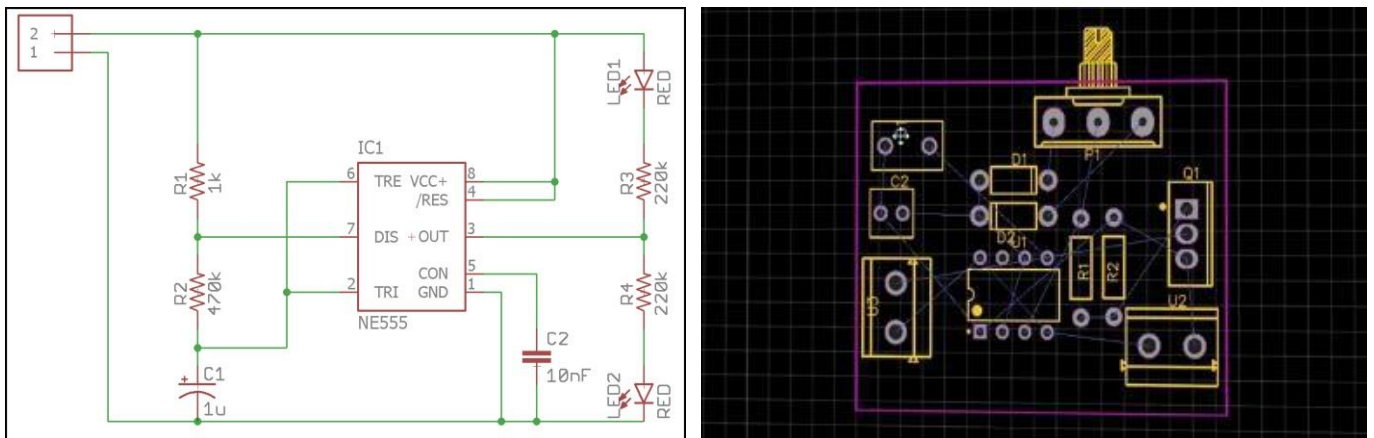


Figure 1. 6 : Placement des composants

9.3. Routage de circuits imprimés

Le routage de circuits imprimés est une étape importante durant la conception du circuit imprimé.

Il permet la connectivité électrique entre les composants d'un circuit imprimé manuellement ou automatiquement. Il n'existe pas deux circuits imprimés identiques. Le routage rend les conceptions uniques [5].

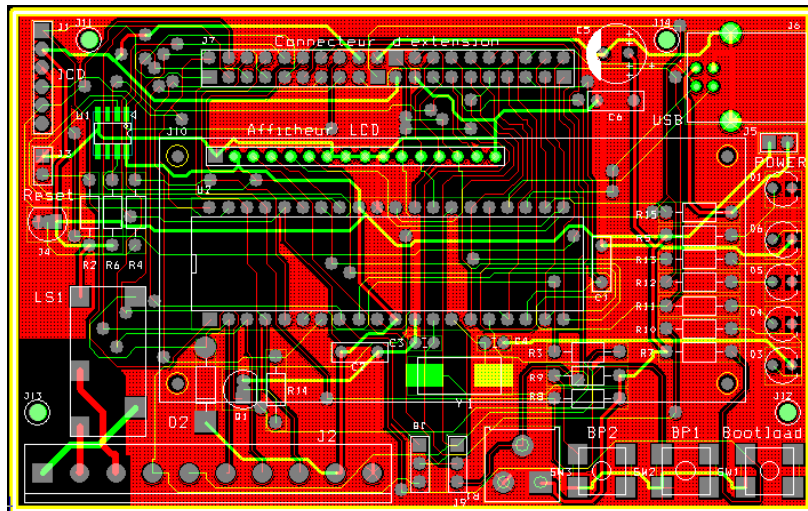


Figure 1. 7 : ROUTAGE DE CIRCUITS IMPRIMÉS

9.4. Exporté le fichier GBR

Un fichier GBR est un fichier Gerber qui stocke des conceptions de cartes de circuits imprimés. La plupart des programmes de conception de circuits imprimés peuvent exporter des données vers un fichier Gerber.

Les fichiers gerber continent des modèles 3D détaillée de PCB ce fichier doit être envoyé à l'industrie de fabrication de pcb comme jlcpcb, pcbway, etc.

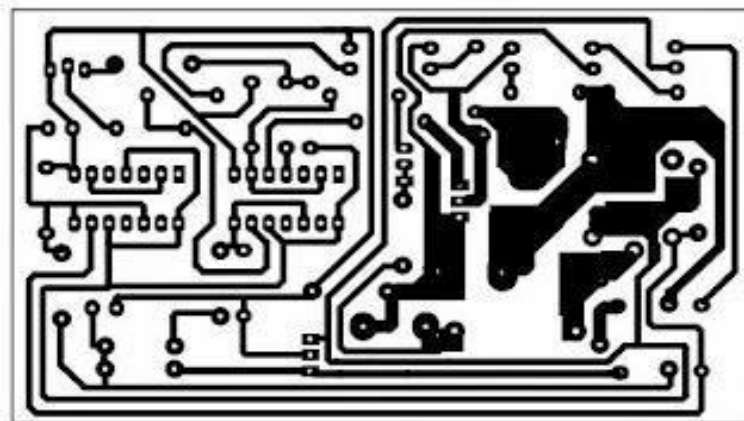


Figure 1. 8 : Gerber

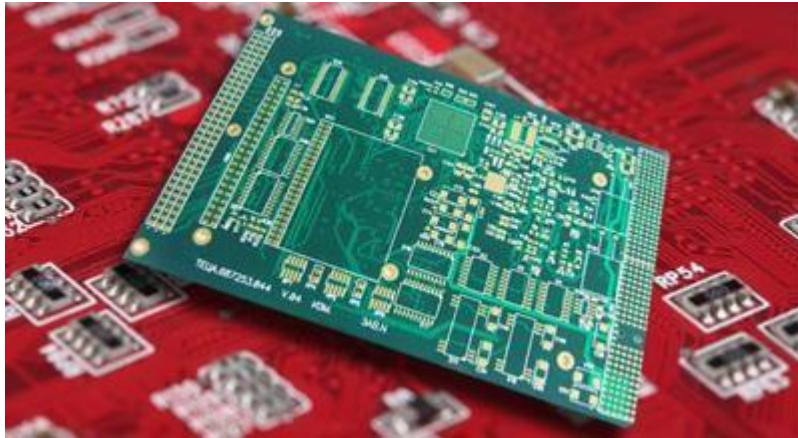


Figure 1. 9 : PCB fabriqué

9.5. Intégration de la conception électronique & Mécanique

La majorité des conceptions électroniques font partie d'un produit mécanique. Pour la conception mécanique des ingénieurs utilise solidworks ou blender pour réaliser un modèle 3D compatible pour protéger la conception électronique. Un projet électronique bien fini il doit tenir compte de la conception mécanique.

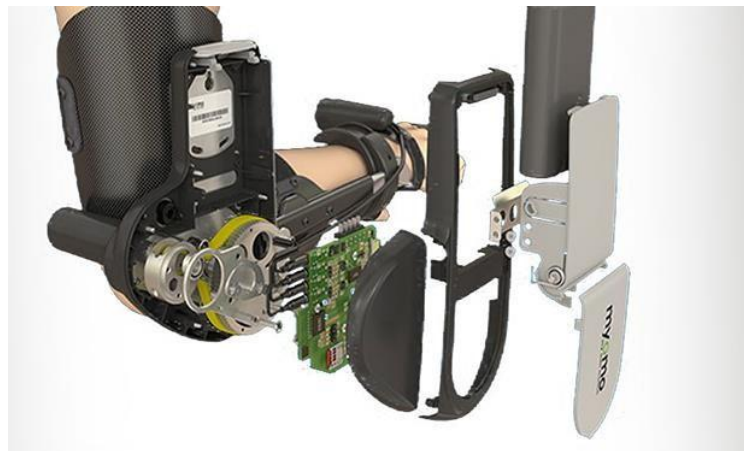


Figure 1. 10 INTÉGRATION DE LA CONCEPTION ÉLECTRONIQUE ET MÉCANIQUE [8]

10. Véhicule connectée

Les voitures connectées nous orientent vers des trajets plus sûrs et plus pratiques et alimentent une industrie des données de plusieurs milliards de dollars.

10.1. Positionnement

Les avantages de la connectivité dans les STI, y compris la sécurité et la commodité, découlent d'informations partagées entre les véhicules connectés, les autres participants au trafic connectés, et les infrastructures routières. Un ensemble essentiel d'informations partagées comprend la position (emplacement) et la vitesse car ils permettent à l'appareil connecté de connaître la présence et prédire le comportement d'autres trafics pertinents, même en s'appuyant uniquement sur les données échangées en l'absence d'autres capteurs. Cela permet en outre à un véhicule système pour avertir le conducteur humain, voire initier automatiquement des actions correctives.

Systèmes de positionnement par satellite, y compris l'American Global Positioning System (GPS) et également d'autres systèmes mondiaux de navigation par satellite (GNSS), fournir un emplacement et une vitesse référencés à l'échelle mondiale qui sont souvent suffisamment précis pour de nombreuses applications STI, mais leur performance souffre dans le ciel obstrué.

Intégration d'autres données pertinentes pour le positionnement grâce à la fusion de capteurs, comme celle des capteurs inertiels, télémétriques et de vision, mais aussi des cartes, améliore le positionnement robuste dans divers environnements. Application de technologies satellitaires avancées algorithmes de positionnement, tels que GPS différentiel (DGPS), cinématique en temps réel (RTK) et positionnement précis des points (PPP) [9].

10.2. Interface Homme-machine

L'intention principale de ce chapitre est de décrire ce qu'est l'interaction homme-machine (HMI) est, pourquoi il est important dans le contexte automobile et comment les véhicules connectés peut en bénéficier. Chaque appareil destiné à être utilisé par des personnes, y compris les voitures, devrait fournir des moyens qui permettent une interaction réussie. C'est tout ce qu'IHM est à propos : comment concevoir une interface qui permet une interaction intuitive, simple et opportune avec un appareil. Selon la nature de la tâche, certains de ces aspects peuvent ont une priorité plus élevée que les autres. Cependant, tous les trois sont extrêmement

importants dans les véhicules, car la conduite est une activité complexe nécessitant les mains et les yeux qui posent charge à la fois physique et cognitive des conducteurs. Dans une telle situation, il faut pour qu'une IHM fournisse des informations adéquates aux conducteurs sans affecter négativement leur tâche principale de conduire. Les informations présentées aux conducteurs peuvent être grossièrement divisé en deux catégories

: info divertissement et sécurité. L'info divertissement est surtout concerné avec des fonctionnalités pratiques dans les véhicules, telles que **la navigation, la musique, la climatisation, téléphone, etc.** Dans ce chapitre, nous nous intéressons principalement à l'IHM pour les applications, puisque c'est dans ce domaine que les véhicules connectés peuvent apporter plus. La communication V2X permet un tout nouvel horizon de détection qui n'est pas réalisable avec l'un des capteurs de véhicule actuellement disponibles, tels que **radar, lidar, caméra, etc.** Cela permet de concevoir de tous nouveaux éléments liés à la sécurité et Advanced Driver Assistance Systems (ADAS) par rapport à ce qui était possible avant [9].



Figure 1. 11 : véhicule connecté Interface Homme-machine

Dans tous les cas, l'IHM correspondante doit expliquer efficacement la situation sur la route et inciter les conducteurs à effectuer les actions adéquates (comme éviter un obstacle ou un freinage préventif). Ceci peut être réalisé en utilisant n'importe quel type de modalités (et leurs combinaisons) qui influencent les sens de base des gens : la vue, l'ouïe, le toucher, l'odorat et le goût. Dans ce chapitre, nous aborderons quelques exemples représentatifs applications liées à la sécurité et analyser comment différentes modalités peuvent être utilisées pour créer une IHM réussie. Enfin, nous explorerons brièvement comment l'IHM et les véhicules convergent dans le domaine de la conduite autonome [9].

10.3. Un système de gestion des informations d'identification de sécurité pour les communications V2X

Le système de communication V2X exige que les utilisateurs puissent faire confiance aux informations leur est présenté. A cette fin, chaque destinataire doit être en mesure de savoir si les messages reçus sur l'interface radio proviennent d'une source fiable et n'ont pas été altéré pendant la transmission. Cette relation de confiance doit être établie comme dès que deux véhicules reçoivent des messages l'un de l'autre. Dans le même temps, les utilisateurs se soucient sur la vie privée et n'acceptera probablement pas le système s'il permet le suivi de singuliers dispositifs. Assurer à la fois la sécurité et la confidentialité dans la plus large mesure raisonnable et possible est le défi majeur et l'objectif de conception du SCMS de sécurité présentée dans ce chapitre. Il a été conçu pour les communications V2V et V2I et a été développé par le Crash Avoidance Metrics Partnership (CAMP) dans le cadre d'un Accord de coopération avec l'USDOT [9].

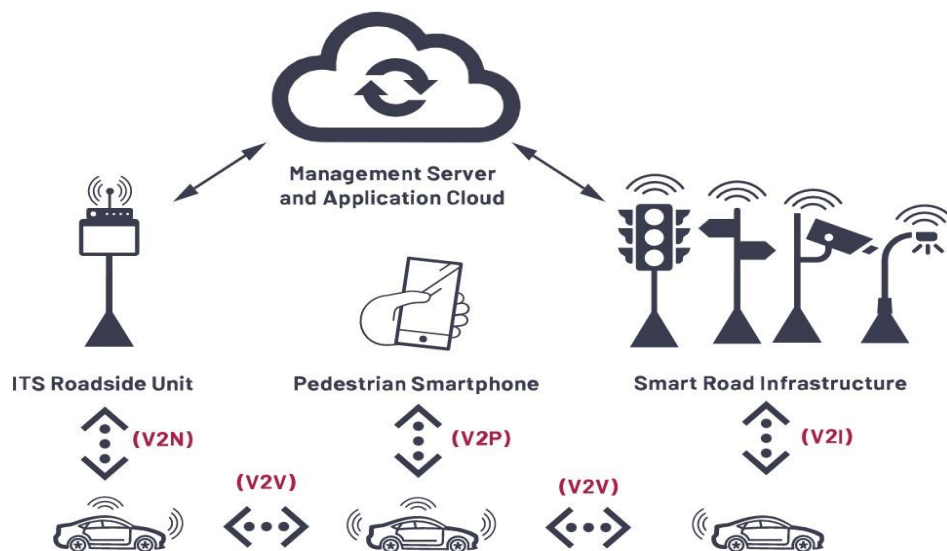


Figure 1. 12 : Système de communication.

10.4. V2V Sécurité Véhicule Communication

La National Highway Traffic Safety Administration (NTHSA) s'est intéressée à La communication V2V comme prochaine étape pour faire face aux taux croissants de décès d'accidents liés aux véhicules. Les technologies d'évitement des collisions d'aujourd'hui

dépendent à bord des capteurs comme la caméra et le radar pour fournir une entrée de sensibilisation aux applications de sécurité.

Ces applications avertissent le conducteur d'un danger imminent ou parfois même agir au nom du conducteur. Cependant, même des technologies comme celles-ci ne peuvent pas « prédire » un accident qui pourrait se produire à cause d'un véhicule qui n'est pas très proche ou pas dans la ligne de visée vers le véhicule hôte.

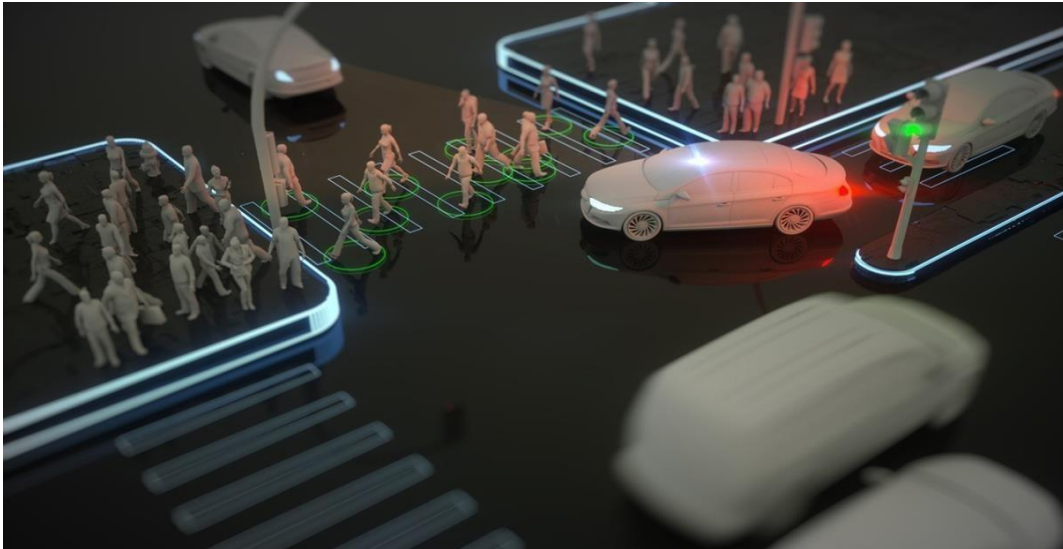


Figure 1. 13: Communications entre les véhicules connectés

Une technologie qui peut voir à travers un autre véhicule ou des obstacles comme des bâtiments et prévoir un danger peut combler ces lacunes et réduire considérablement les accidents. Les communications V2V peuvent fournir aux véhicules la capacité se parler et donc voir dans les coins et à travers les obstacles une distance plus longue par rapport aux capteurs embarqués actuels. On estime que V2X les communications résolvent jusqu'à 80 % des accidents non altérés. La NHTSA travaille à la normalisation de Communications V2V et éventuellement rendre obligatoire la diffusion des données du véhicule (par ex. Coordonnées GPS, vitesse, accélération) sur DSRC via V2V. Un véhicule a besoin d'un Unité embarquée (OBU) pour établir la communication V2V avec d'autres véhicules également équipés d'OBU ou de communication V2I avec l'infrastructure de trafic équipés d'unités routières (RSU). En général, un OBU dispose d'une radio DSRC pour l'émission et la réception, un récepteur GNSS, un processeur et plusieurs interfaces (par ex. CAN, Ethernet, GPS) pour obtenir les données du véhicule. Message essentiel dans La communication V2V est appelée messages de sécurité de base (BSM). BSM est une émission message généralement transmis fréquemment jusqu'à 10

fois par seconde. Contenu de BSM comprend des informations sur le véhicule telles que la vitesse du véhicule, l'emplacement et l'état des freins.

Les applications de sécurité utilisent les données des véhicules distants (RV) de BSM et du véhicule hôte (HV) des interfaces OBU telles que CAN et GNSS pour prédire un potentiel crash et alerter le conducteur. Les messages V2V pourraient également être fusionnés avec des capteurs embarqués tels que radar, LiDAR et caméra pour améliorer le niveau de confiance de détection de véhicule pour des applications de sécurité ou même pour la conduite autonome pour certain degré [9].

11. L'intelligence Artificielle & les véhicules autonomes

La vision par ordinateur est utilisée pour aider les machines à interpréter rapidement les données afin qu'elles puissent voir leur environnement et prendre rapidement des décisions en fonction de cette entrée visuelle. Ces machines sont souvent essentielles pour assurer le maintien de la production ou des processus critiques. Les systèmes en temps réel permettent de garantir que de telles machines sont capables de traiter ces données en temps quasi réel [2].

11.1. Systèmes informatiques pour la conduite autonome

Un véhicule autonome implique de multiples sujets, comme les systèmes informatiques, l'apprentissage automatique, la communication, robotique et ingénierie des systèmes, intégrer différentes technologies et innovations.

11.2. Applications de conduite autonome

De nombreux algorithmes sont déployés dans le système informatique pour la détection, la perception, la localisation, la prédiction et le contrôle. Dans cette partie, nous présentons l'état de l'art des algorithmes y compris la détection d'objets, la détection de voie, la localisation et la cartographie, la prévision et la planification, et le contrôle des véhicules [10].

11.2.1. La détection d'objets

La détection d'objets est une tâche de vision par ordinateur qui fait aujourd'hui partie intégrante de nombreuses applications grand public telles que les systèmes de surveillance et de sécurité, la reconnaissance de texte mobile et le diagnostic de maladies à partir d'IRM/TDM. La détection d'objets est également l'un des composants essentiels pour prendre en charge la conduite autonome. Les véhicules autonomes s'appuient sur la perception de leur

environnement pour assurer des performances de conduite sûres et robustes. Ce système de perception utilise des algorithmes de détection d'objets pour déterminer avec précision des objets tels que des piétons, des véhicules, des panneaux de signalisation et des barrières à proximité du véhicule. Les détecteurs d'objets basés sur l'apprentissage en profondeur jouent un rôle essentiel dans la recherche et la localisation de ces objets en temps réel [10].



Figure 1. 14 :Détection d'objets [7]

11.2.2. La détection de voie

La détection de voie dans les scènes de conduite est un module important pour les véhicules autonomes et systèmes de l'assistance avancée à la conduite. Ces dernières années, de nombreux détecteurs de voie ont été proposés.

Effectuer une détection de voie précise dans le temps réel est une fonction cruciale de l'assistance avancée à la conduite (ADAS), car il permet aux VA de se conduire eux-mêmes dans les voies de circulation correctement pour éviter les collisions, et il prend en charge la décision et la voie de planification de trajectoire ultérieures Départ [10].



Figure 1. 15 : Détection des voies

12. Conclusion

Dans ce chapitre, nous avons présenté un système de sécurité pour les véhicules. Nous avons expliqué à quelles exigences un tel système doit répondre et comment le concept présenté les remplit. Nous avons montré les bases de conception d'un système embarqué pour les véhicules connectés et autonome et nous avons expliqué comment ses caractéristiques de conception ciblant un travail à haut rendement.

Chapitre 2 : Implémentation électronique

1. Introduction

Les systèmes embarqués dans l'automobile font partie des systèmes électroniques complexes, ce qui fait qu'ils doivent avoir une puce puissante. Cette puce est généralement le microcontrôleur ou le microprocesseur d'où le choix de la puce est vraiment important.

2. Qu'est-ce qu'un microcontrôleur ?

Un microcontrôleur est un circuit intégré (IC) unique qui est comparable à un petit ordinateur autonome et il est conçu pour effectuer les tâches spécifiques des systèmes embarqués. Un microcontrôleur contient une unité de traitement mais une petite quantité de mémoire (ROM, RAM, etc.), quelques IO ports pour les périphériques, une minuterie, etc. AVR et ARM relèvent de la famille des microcontrôleurs. Mais ARM peut être utilisé à la fois comme microcontrôleur ou comme microprocesseur. Le microcontrôleur ARM et le microcontrôleur AVR diffèrent les uns des autres en termes d'architecture et d'ensembles d'instructions, de vitesse, de distribution, de mémoire, de consommation d'énergie, de largeur de bus, etc.

Nous donnons en détail en quoi ils diffèrent les uns des autres.

Tableau 2. 1: La différence entre ARM & AVR

	AVR Microcontrôleur	ARM Microcontrôleur
Width	8/32 bit	32-bit and also available in 64-bit
Speed	1 clock/ instruction cycle	1 clock / instruction cycle
Memory	Flash, SRAM and EEPROM	Flash, SDRAM and EEPROM
ISA	RISC architecture	RISC architecture
Memory Architecture	Modified	Modified architecture
Power consumption	Low	Low
Manufacturer	Atmel	Apple, Samsung Electronics and TI
Communication Protocols	UART, USART, SPI, I2C	UART, USART, LIN, I2C, CAN, USB, Ethernet, I2S, DSP, SAI

3. Le microcontrôleur AVR

Le microcontrôleur AVR est fabriqué par la société Atmel en 1996. Il est basé sur RISC Instruction set Architecture (ISA) et également appelé Advanced Virtual RISC. AT90S8515 était le microcontrôleur initial appartenant à la famille AVR. Le microcontrôleur AVR est la catégorie de contrôleur la plus populaire et il est bon marché. Il est utilisé dans de nombreuses applications robotiques [15].

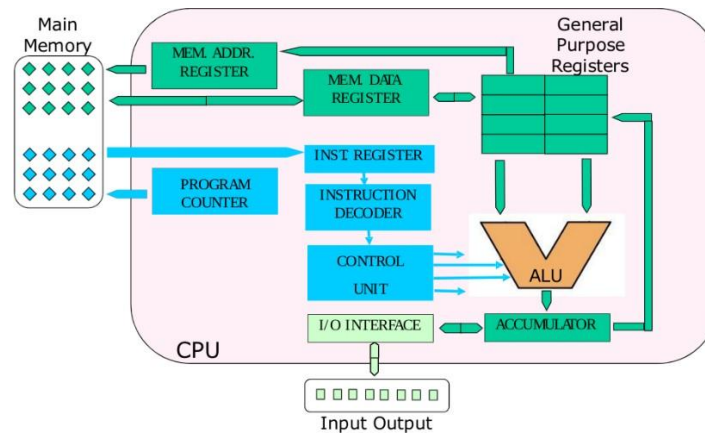


Figure 2. 1 : Architecture de Uc AVR

4. Le microcontrôleur ARM

Le microcontrôleur ARM a été introduit par l'organisation informatique Acron et est fabriqué par Apple, Nvidia, Qualcomm, Motorola, ST Microelectronics, Samsung Electronics et TI, etc. Il est basé sur l'architecture de jeu d'instructions RISC (ISA) et également appelé micro-contrôleur RISC avancé. C'est le microcontrôleur le plus populaire et la plupart des industries l'utilisent pour les systèmes embarqués car il fournit un large éventail de fonctionnalités et est bon pour produire des appareils avec d'excellentes apparences [15].

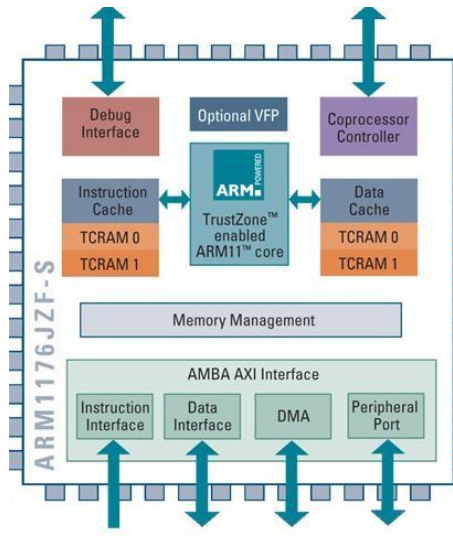


Figure 2. 2 : Architecture Uc ARM

4.1. Les meilleures cartes de développement ARM

Dans la Figure2.3 nous montrons les microcontrôleurs ARM les plus connues et les plus utilisés aujourd'hui comme STM, Atmel, Microchip.



Figure 2. 3 : Qu'elle que Uc ARM

5. Système de base

Dans notre voiture, nous utilisons Raspberry pi comme principal puce de traitement. Open CV a été installé dans le système d'exploitation et avec la caméra Raspberry, la vision par ordinateur et l'apprentissage automatique sont utilisés pour évaluer la sortie. Cette sortie est transmise à Uc ARM et le Uc est relié à L298 qui permet de contrôler les moteurs. Ils sont tous alimentés par power Bank comme le montre la figure (2.4).

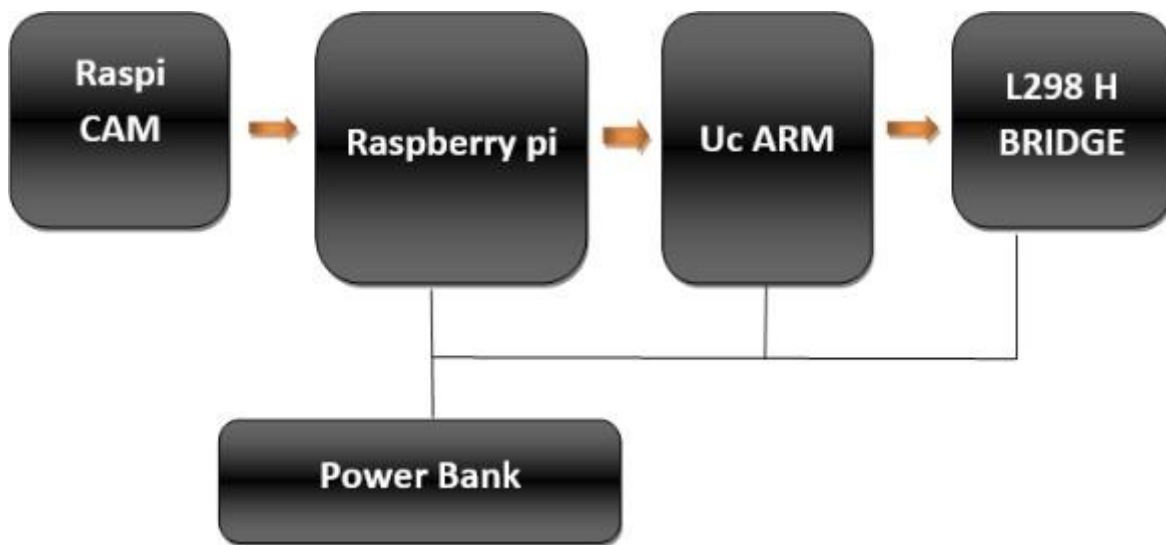


Figure 2. 4 : Block diagramme

Le Raspberry Pi est un nano-ordinateur mono-carte à processeur ARM de la taille d'une carte de crédit conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi.

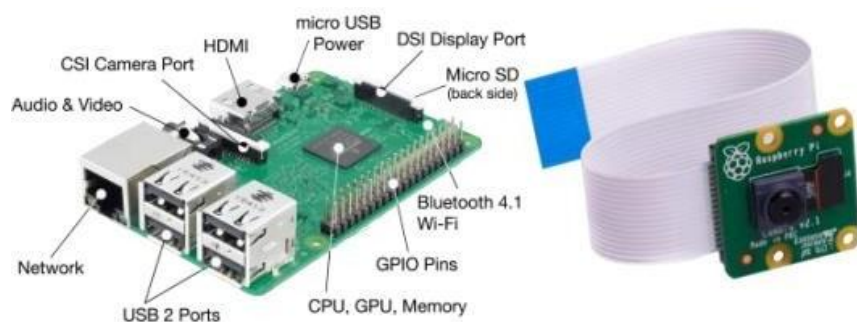


Figure 2. 5 : Raspberry pi + camera pi

6. Implémentation d'un système de navigation

Dans cette section, nous allons donner expliquer les étapes d'implémentation d'un système de navigation.

6.1. Placement des composants pour le système de navigation

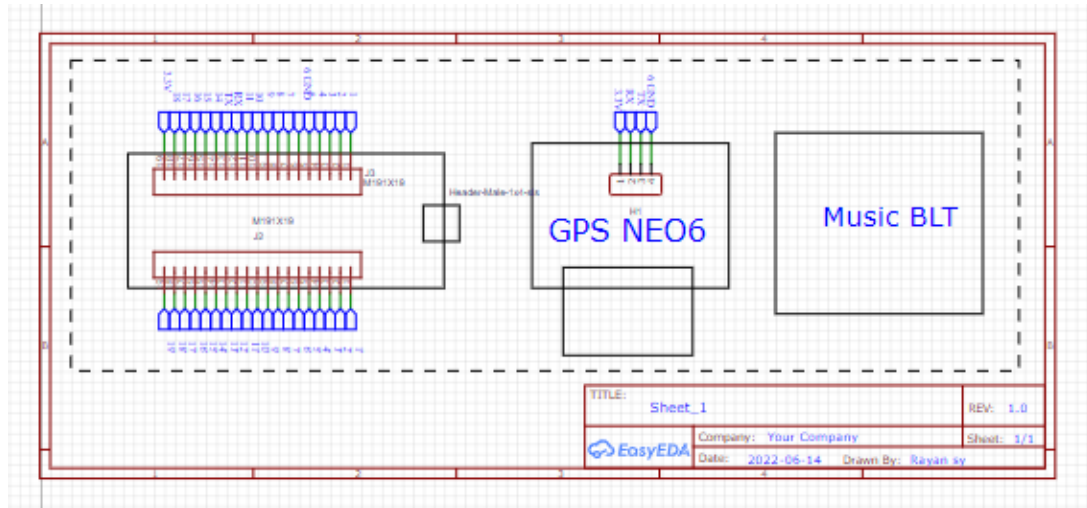


Figure 2. 6: Placement des composants pour le système de navigation

Dans ce schéma on a (Esp32, GPS Module & Module Bluetooth speaker) qui sont intégrés ensemble c'est le système de navigation connectée (iot)



Figure 2. 7 : Circuit de navigation sur plaque de bakélite

Composants requis

- ESP32
- GPS Module
- Fils de cavalier
- Breadboard

Tableau 2. 2 : Connexion entre les pins d'ESP32 et GPS module

ESP32 Pin	GPS Pin
3.3v	3.3v
	TX
	RX
GND	GND

6.2. Routage du circuit de système navigation

Le routage est une étape importante dans le PCB Design il permet de connecter les pins du système entre eux sellant le tableau (3.3v avec 3.3v), ainsi de suite

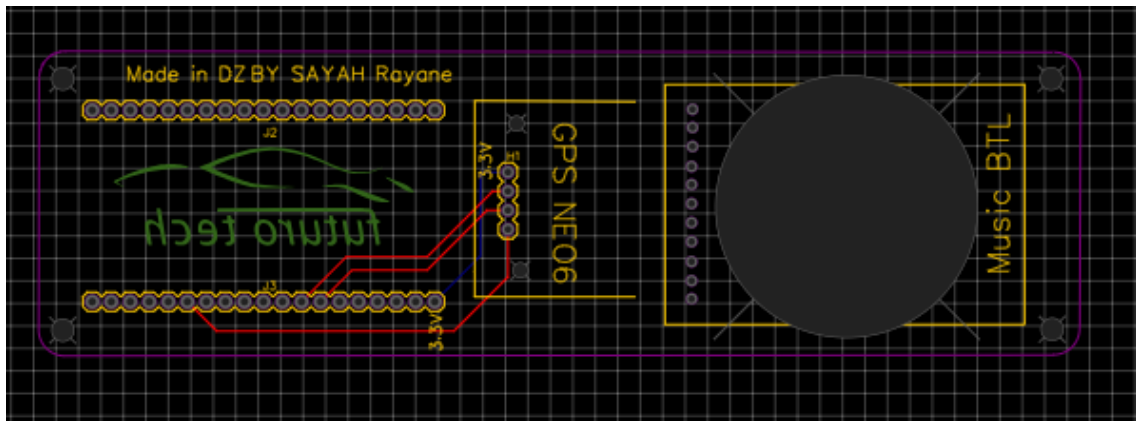


Figure 2. 8 : Routage circuit de système navigation

Esp 32 microcontrôleur est une série de microcontrôleurs de type système sur une puce (SoC), basé sur l'architecture Xtensa LX6 de Tensilica (en), intégrant la gestion du Wi-Fi et du Bluetooth en mode double, et un DSP. C'est une évolution d'ESP8266.

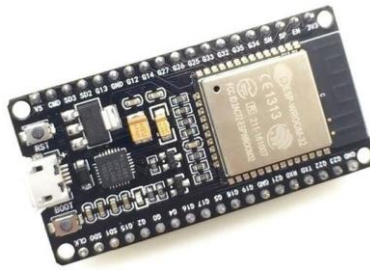


Figure 2. 9 : ESP 32 WIFI + BLU

- **Le module NEO-6M**

Le module NEO-6M est livré avec un boîtier de dimensions de 16 x 12,2 x 2,4 mm. Il dispose de 6 moteurs de positionnement Ublox offrant des performances inégalées. C'est un récepteur GPS de bonne performance avec une architecture compacte, une faible consommation d'énergie et des options de mémoire fiables. Il est idéal pour les appareils mobiles fonctionnant sur batterie compte tenu de son architecture et de ses besoins en énergie.

Temps de localisation est inférieur à 1 seconde et il lui permet de trouver les satellites presque instantanément. La sortie est au format des normes NMEA, qui peuvent être décodées pour trouver les coordonnées et l'heure de l'emplacement.

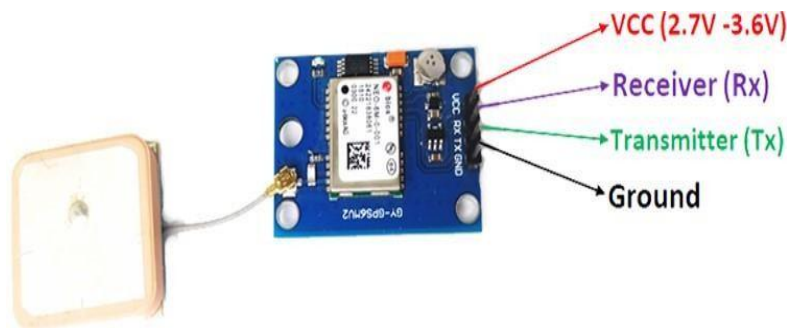


Figure 2. 10 : GPS module

- Alimentation : 2.8V à 5V
- Interface : RS232 TTL
- EEPROM intégré et antenne externe
- Débit en bauds par défaut : 9600 bps

6.3. Implémentation du système anti incendie

Dans cette section, nous allons expliquer la méthode d'implémentation du système anti incendie :

6.4. Placement des composants pour le système anti incendie

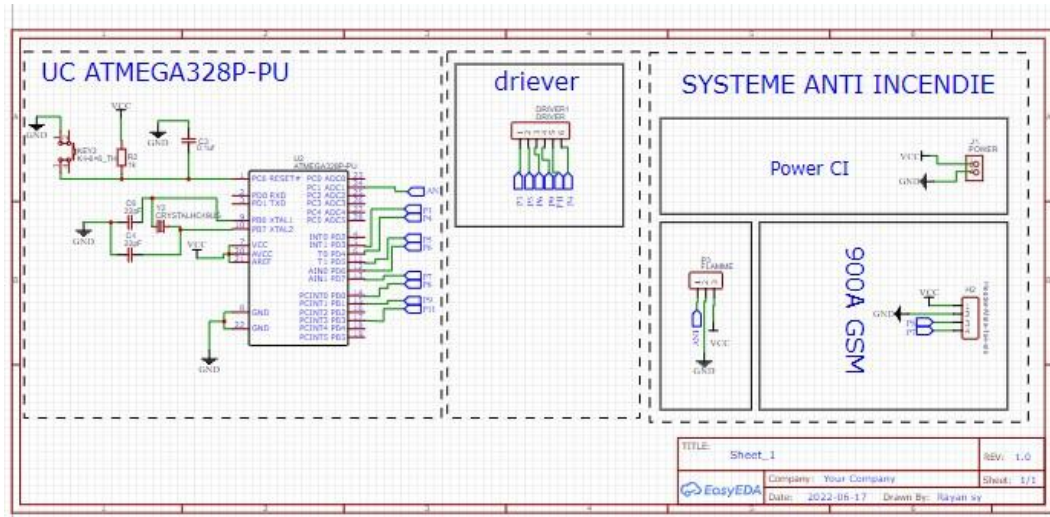


Figure 2. 11 : Placement des composants pour le système anti incendie

Dans ce schéma on a Atmega, GSM Module, capteur de flamme, l'alimentation & draver pour contrôler l'actionneur qui va ensuite exciter l'extincteur sont intégrés ensemble c'est le système anti incendie connecté (IoT).



Figure 2. 12 : Circuit de système anti incendie sur plaque d'essai

Composants requis

- ATmega 328p
- Modem GSM
- Actionneur Linéaire
- Détecteur de flamme
- Extincteur
- Breadboard

6.5. Routage circuit de système anti incendie

Le microcontrôleur atmega est connecté avec les pins du GSM 900A, pins de L298n et les pins des autres composant quartz, bouton poussoir résistance etc.

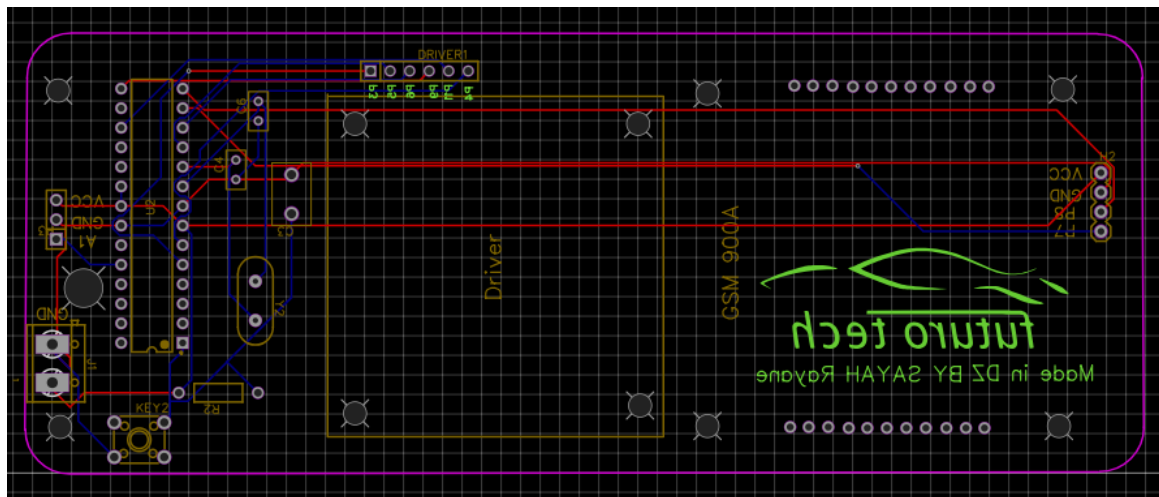


Figure 2. 13 : : Routage circuit de système anti incendie

• ATmega 328p

C'est un microcontrôleur ATMEL de la famille AVR 8bits.



Figure 2. 14 : Microcontrôleur ATmega328

Les principales caractéristiques sont :

- **FLASH** =mémoire programme de 32Ko
- **SRAM** =données (volatiles) 2Ko
- **EEPROM** =données (non volatiles) 1Ko
- **Digital I/O**(entrées-sorties Tout Ou Rien) = 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)
- **Timers/ Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)

• **Actionneur Linéaire**

Un actionneur linéaire électrique se compose d'un moteur à courant continu, d'une série d'engrenages (réducteur), de deux interrupteurs de fin de course et d'une vis mère comprenant un écrou.

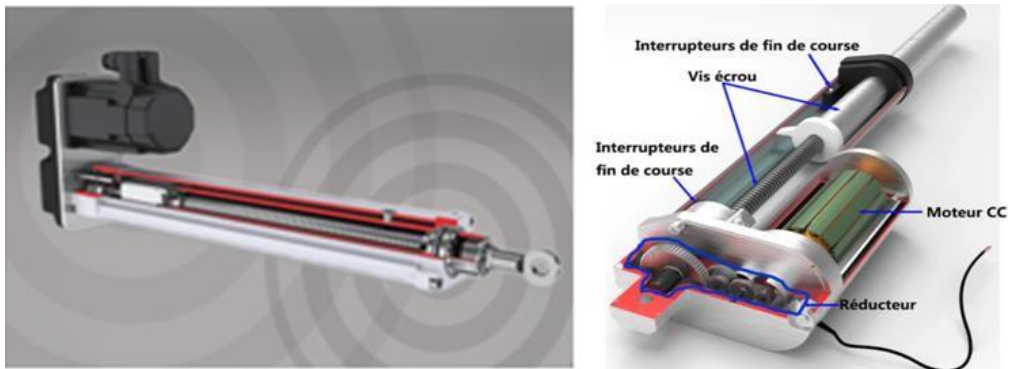


Figure 2. 15 : Actionneur Linéaire

7. Le circuit imprimé du system

Pour la conception du circuit imprimé il existe plusieurs logiciels et sites en ligne. Dans notre projet nous avons utilisé **easyeda** ou **altium designer** pour la conception des circuits imprimés.

7.1. Placement des composants pour le système global

Sur la figure ci-dessous nous montrons le placement des composants concernant notre projet.

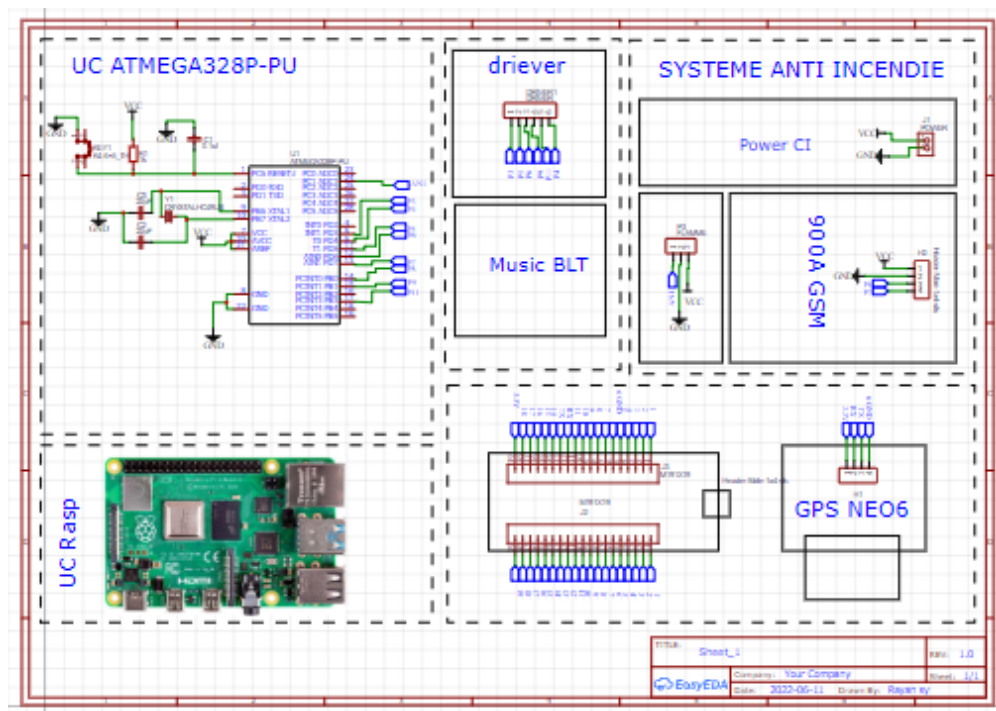


Figure 2. 16 : Placement des composants pour le système global

Dans ce schéma tous les composant du système sont intégrés ensemble afin d'obtenir le système global qui se trouve sur la figure ci-dessous :



Figure 2. 17 : Circuit de système global

7.2. Routage de circuit global

Ici on a connecté tous les pins du système global, les deux couches de PCB sont utilisées pour relier beaucoup de composant avec une surface réduite.

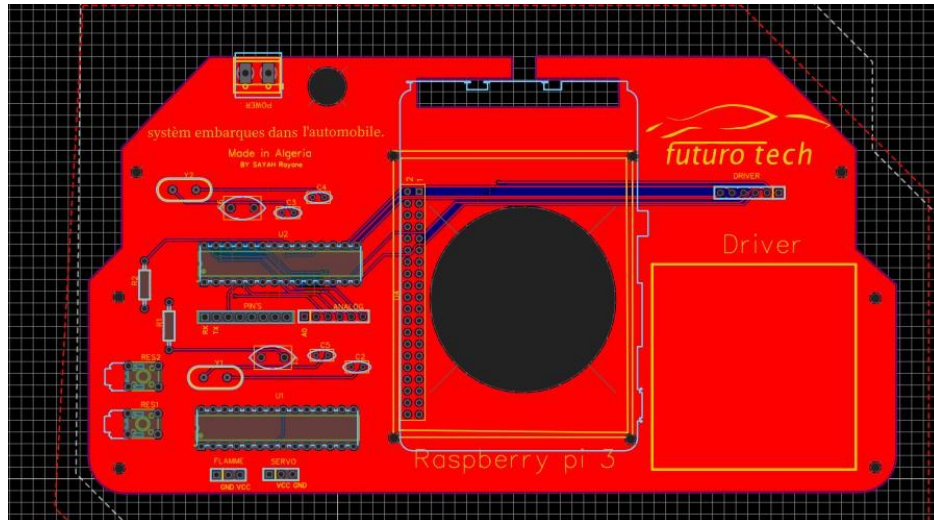


Figure 2. 18 : Routage de circuit global

7.3. Présentation du circuit global

Sur la figure (2.19) nous donnons le circuit global.



Figure 2. 19 : Circuit global

Dans cette plaque a bakélite on a implémenté tous le système (anti incendie, navigation, autonomie).

7.4. Présentation 3D du circuit global

Sur la figure ci-dessous, nous donnons la représentation en 3D du circuit global.

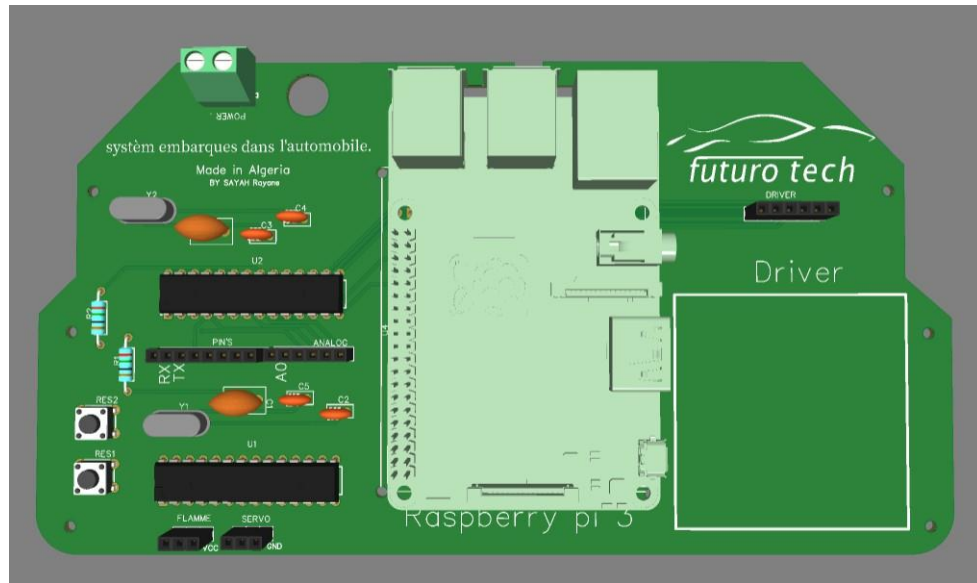


Figure 2. 20 : Présentation 3D du circuit global

Tous l'intelligence de système tient dans ce circuit imprimé dotée d'un microcontrôleur et d'autre composants qui font l'intégralité du système

7.5. Intégration de la conception électronique et mécanique

L'intégration de la conception électronique et mécanique est une étape essentielle pour avoir un projet bien fini on utilise des logiciels de 3D design pour la conception il existe plusieurs, dans notre cas nous utilisons blender et SolidWorks pour notre projet.

On peut faire la conception électronique sur SOLIDWORKS Electrical pour intégrer les deux conceptions ensemble, SOLIDWORKS Electrical est le seul package de conception schématique qui peut être lié à l'environnement mécanique 3D SOLIDWORKS. Cette fonctionnalité signifie que vous pouvez automatiquement acheminer et obtenir les longueurs de vos fils, câbles et faisceaux.

Le complément SOLIDWORKS Electrical 3D permet d'acheminer automatiquement les fils, les câbles et les faisceaux dans l'environnement 3D [11].

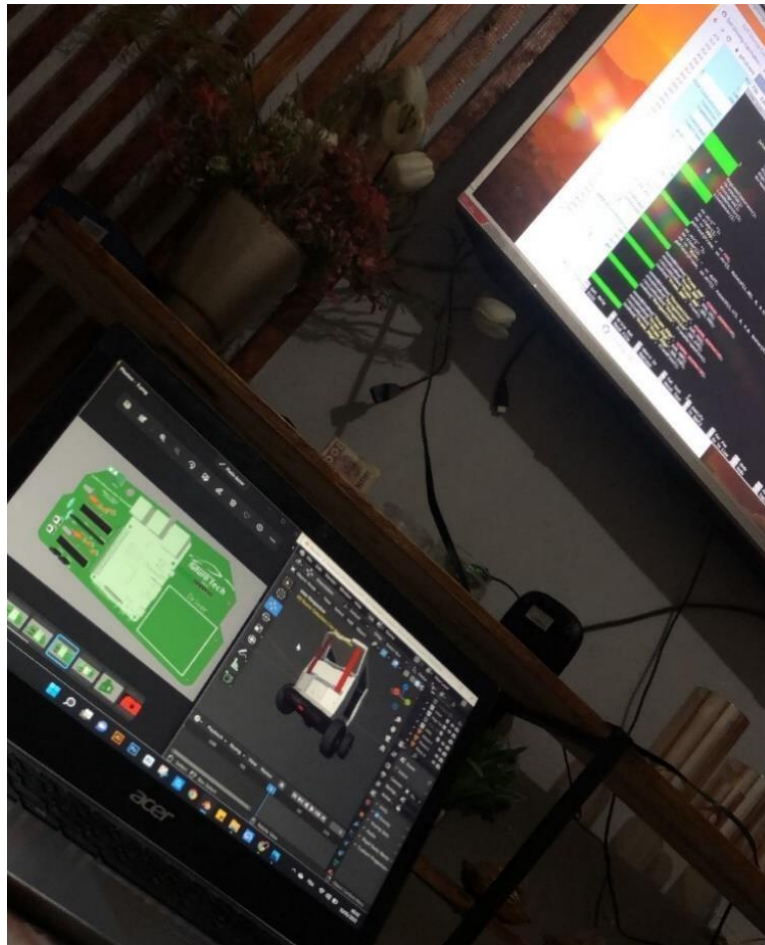


Figure 2. 21 : Présentation 3D électronique et mécanique

8. Conclusion

Dans ce chapitre nous avons étudié la partie électronique de notre projet. Nous avons vu les composants et les cartes de manière détaillée ce qui nous a permis de réaliser notre circuit sur la plaque de bakélite.

Chapitre 3 Résultats & Discussion

1. Introduction

Le but de notre projet est de réaliser un système de sécurité embarqué dans l'automobile basé sur l'intelligence artificielle dans une carte électronique.

Ce chapitre est la suite de la partie électronique, un site web a été créé pour suivre le véhicule en temps réel, une interface a été mise en œuvre pour interfacer à un écran tactile, ce dernier permet l'accès à des applications comme par exemple l'alerte pour les zones de danger, l'information routière & comme les applications qui compte des capteurs du véhicule.

2. Interface home machine

L'automobile change elles sont désormais remplies de composants électroniques, elles ont un grand écran de tableau de bord, des applications logicielles pour la gestion et l'accès aux différentes fonctionnalités et applications, un logiciel qu'il suffit de mettre à jour pour ajouter de nouvelles fonctions.

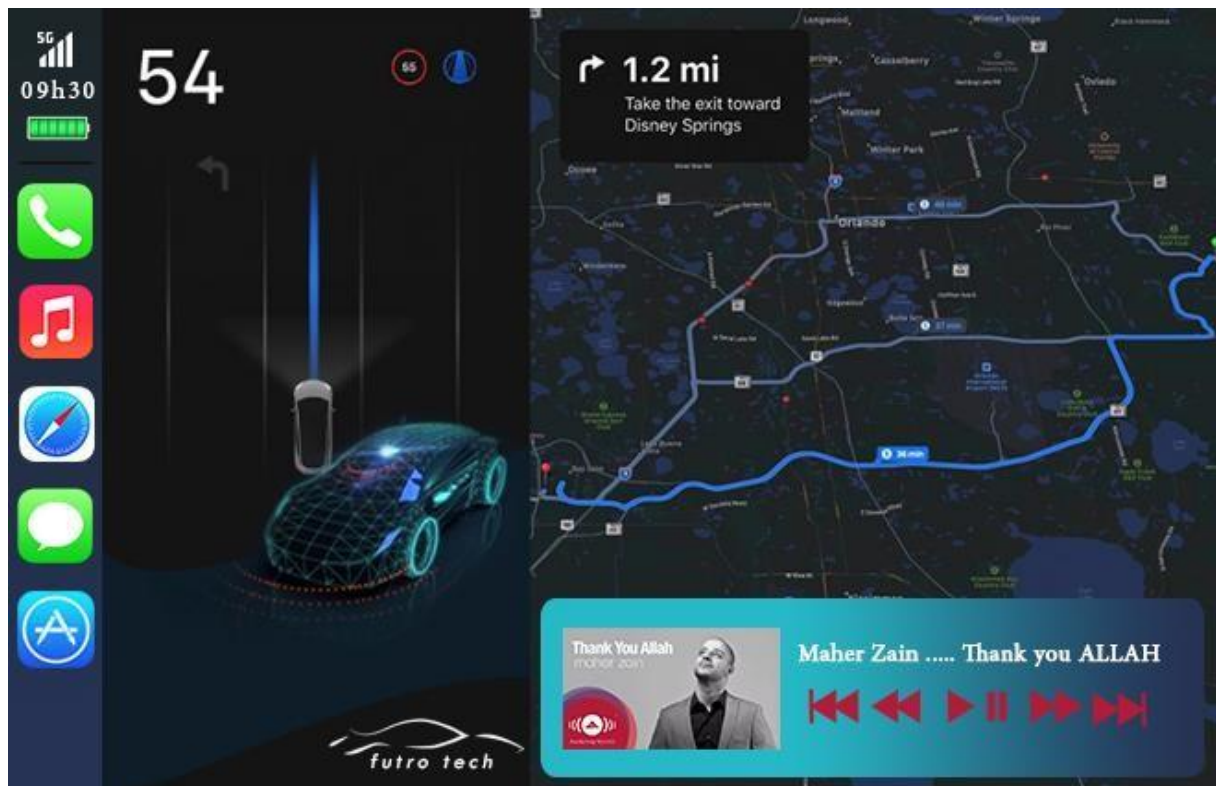


Figure 3. 1 Interface home machine

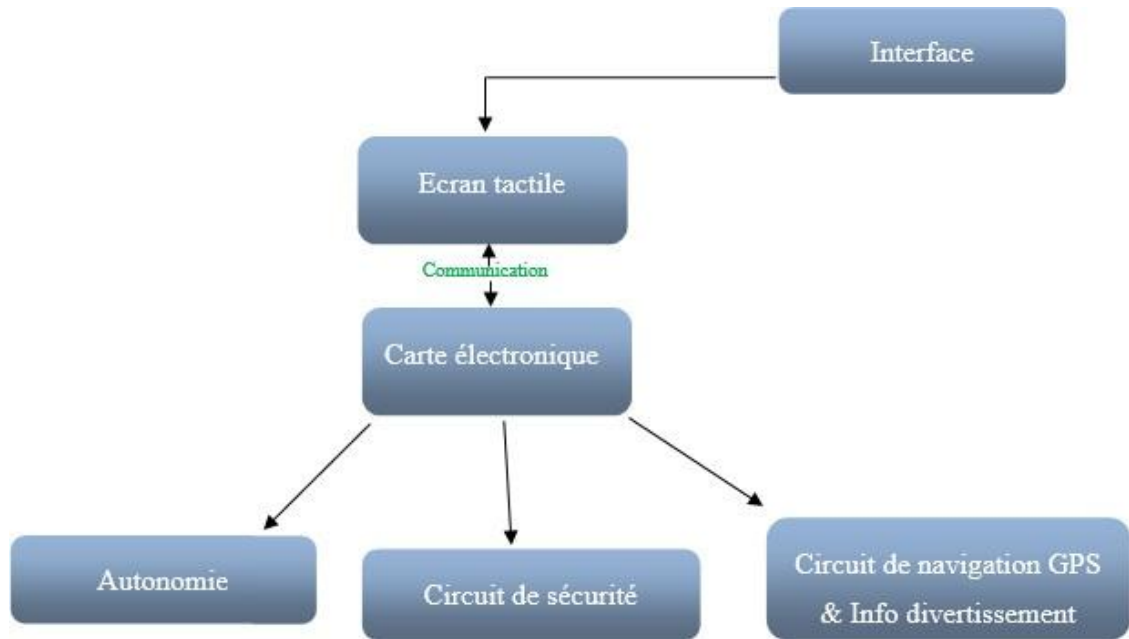


Figure 3. 2 : Communication IHM et le système

Cette interface permet d'interfacer à un écran tactile qui est connecté avec le CPU du système

3. Système anti incendie

Un système anti incendie il permet de détecter un départ de feu dans véhicule, et de gérer la sécurisation des personnes qui se trouvent dedans. Techniquement le système anti incendie détecte le feu à travers un capteur de flamme ou une caméra qui va détecter le feu à base de l'intelligence artificiel une fois le feu est détecter l'extincteur à l'intérieur de la voiture s'excite pour étendre le feu en quelle que seconde.

Dans cette partie je vais expliquer comment marche le système :



Figure 3. 3 : Utilisation du système

3.1. Système anti incendie à base d'un capteur

Sur la figure (3.4) nous donnons l'organigramme qui explique le principe de fonctionnement du système anti incendie.

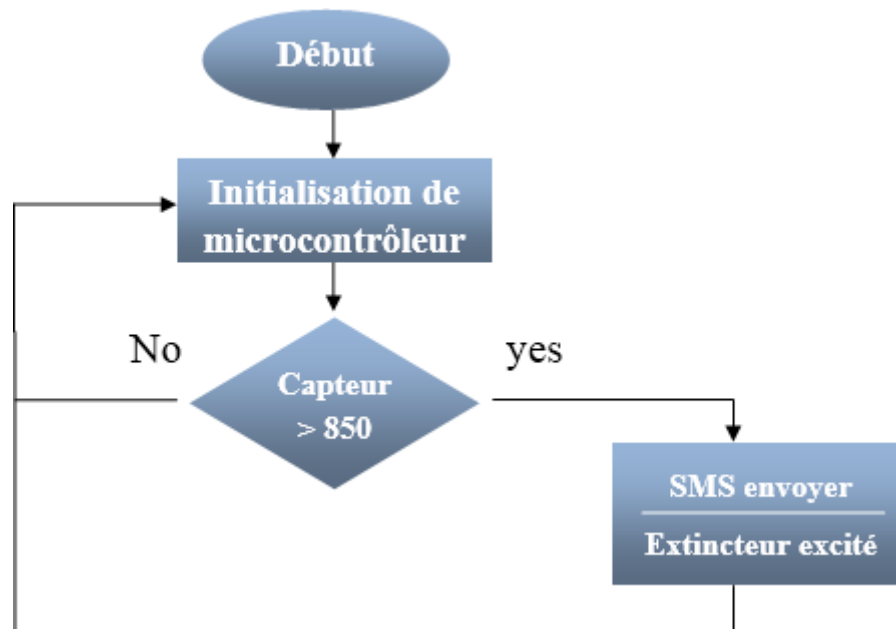


Figure 3. 4 : Principe de fonctionnement système anti incendie

Si le système détecte le feu, la condition (capteur > 850) sera validée et l'extincteur s'excite et on reçoit un message en même temps indiquant l'état de véhicule.

3.1. Système anti incendie à base du AI

La recherche dans le domaine de la détection du feu et de la fumée devient un sujet de plus en plus traité. Les algorithmes conventionnels utilisent exclusivement des modèles basés sur des vecteurs de caractéristiques. Ces vecteurs restent difficiles à définir et dépendent largement du type de feu observé. Ces méthodes donnent des résultats avec un faible taux de détection. Une approche innovante pour résoudre ce problème est d'utiliser un algorithme permettant de déterminer automatiquement les caractéristiques utiles pour classifier le feu et la fumée. Dans cette partie, nous utilisons AI pour identifier le feu et la fumée d'une vidéo en temps réel.

Ces résultats indiquent clairement que l'utilisation des AI pour la détection du feu et de la fumée est très prometteuse.

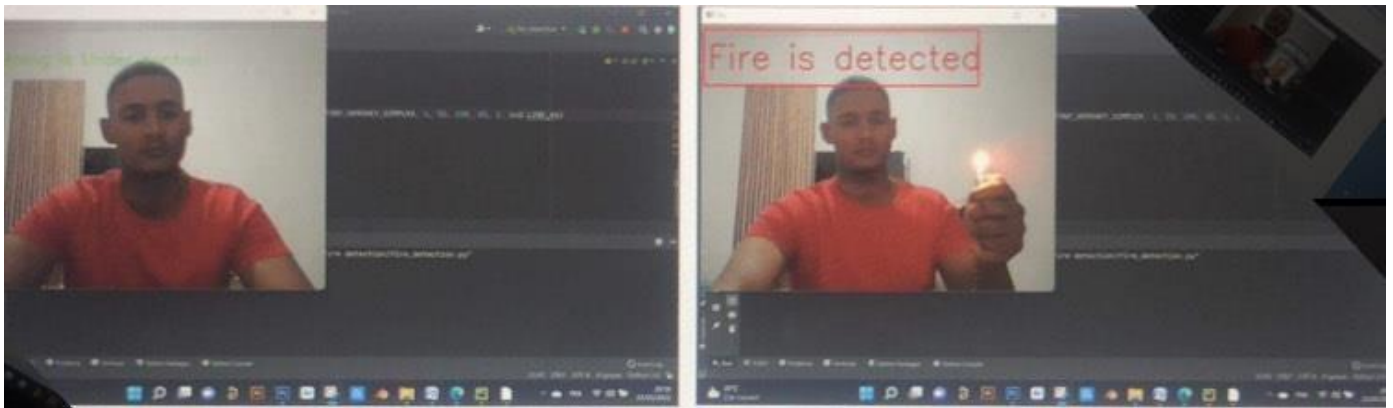


Figure 3. 5 : Détection de la flamme avec AI

Avec camera est à travers qu'elle que code sur python, On réussit à détecter la flamme tout comme on fait avec le capteur de flamme sauf qu'ici on a utilisé la vision par ordinateur (computer vision).

Donc le résultat dès que la caméra détecte la flamme on rassoit un message d'alerte sur le téléphone est l'extincteur s'excite directement.

3.2. Présentation 3D du circuit de système anti incendie

Sur la figure ci-dessous nous donnons le circuit imprimé du système anti incendie à 3D.

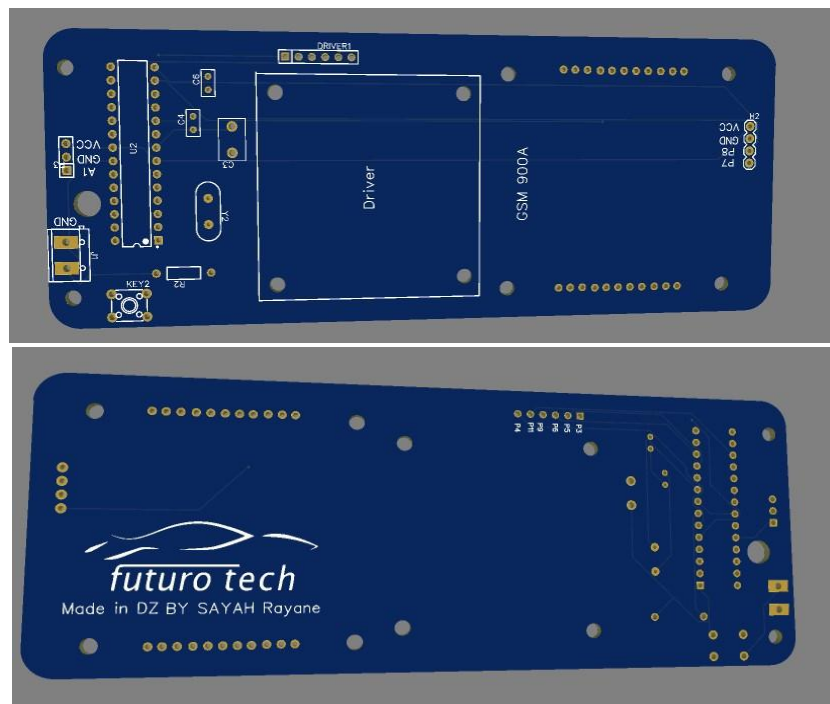


Figure 3. 6 : Présentation 3D du circuit (système anti incendie)

4. Système de navigation IOT (internet of things)

- Système de suivi des véhicules en temps réel

Le système de suivi de véhicule est un système qui peut être utilisé pour suivre n'importe quel type de véhicule via un (GPS).

Grâce à ce système, le véhicule peut être suivi à tout moment par serveur, téléphone ou tout autre système de réseau informatique.

- Services Info-divertissement

Musique en streaming, des appels téléphonique Bluetooth, assistante vocale etc.

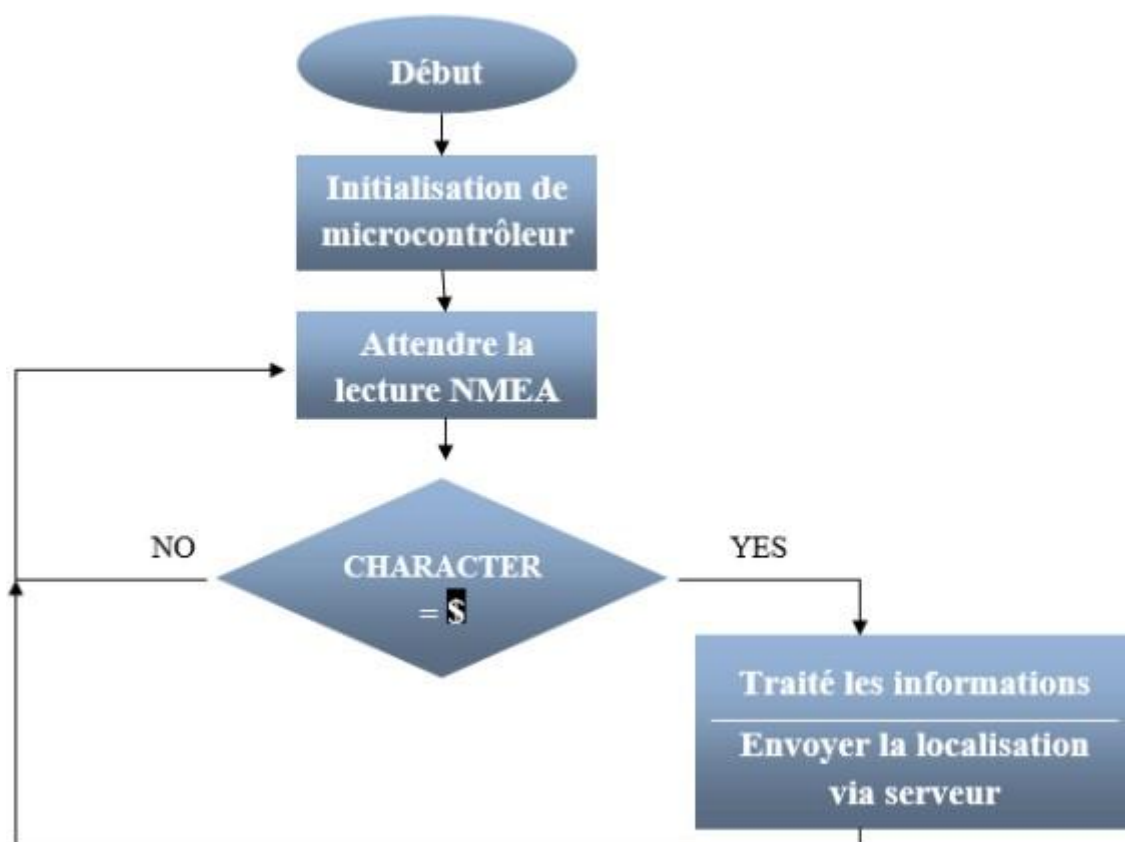


Figure 3. 7 : Principe de fonctionnement GPS

Le Module GPS reçoit les coordonnées de localisation de satellites en lectures NMEA qui contient des informations chiffrées comme ce si :

\$GPGGA, 064036.289, 4836.5375, N, 00740. 9373, E, 1, 04, 3.2, 200.2, M, , , , 0000*0E . Les deux premiers caractères correspondent à l'identifiant du récepteur : ici GP pour Global Positioning System. Les trois lettres suivantes correspondent à l'identifiant de la trame : GGA pour GPS Fix et Date. C'est la trame la plus courante.

Eléments de la trame GGA

Décomposons maintenant cette trame selon les premiers éléments qui la composent :

GPGGA : type de la trame.

064036.289 : heure d'envoi de la trame, ici 06h 40min 36,289s (UTC).

4836.5375, N : latitude Nord, ici 48°36,5375' (en DM, degrés minutes).

00740.9373, E : longitude Est, ici 7°40,9373' (en DM également).

1 : type de positionnement (1 pour le positionnement GPS).

04 : nombre de satellites utilisés.

3.2 : précision horizontale.

200.2, M : altitude, ici 200 mètres.

Le microcontrôleur traite ces informations de localisation selon l'algorithme qui y est stocké, et extrait longitude et latitude, puis indique la localisation en temps réel dans maps, sur une application ou dans l'interface homme machine [12].

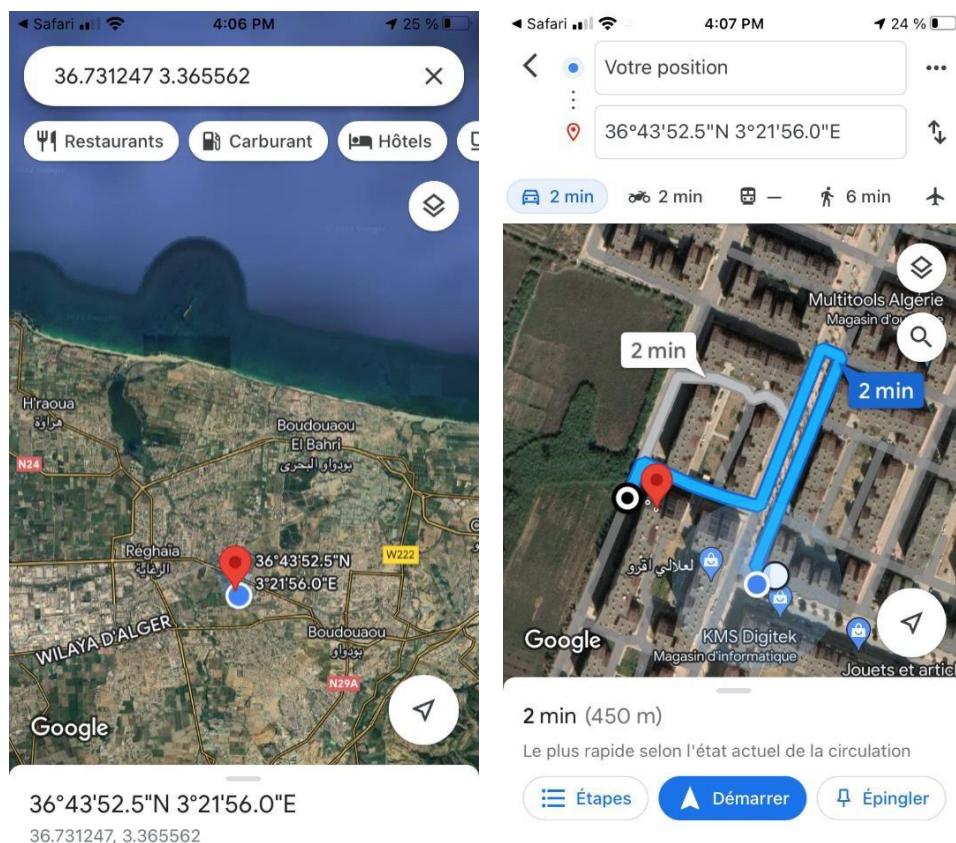


Figure 3. 8 : Résultats de localisation

Dans cette photo j'ai localisé ma voiture a 2 minutes de chez moi à travers le site en temps real.

4.1. Présentation 3D du circuit de navigation

Sur la figure ci-dessous nous donnons la présentation 3D du système de navigation.

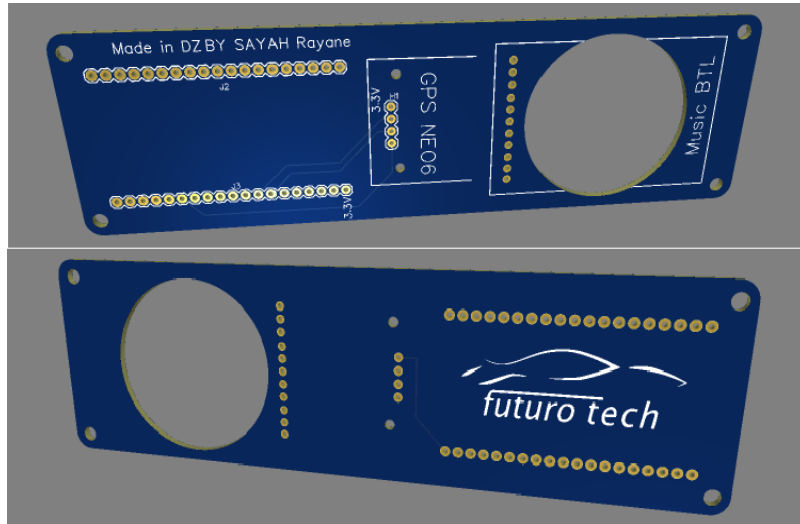


Figure 3. 9 : Présentation 3D du circuit (système de navigation)

5. Logiciels requis

Les logiciels nécessaires à notre implémentation sont :

Python

Python est un langage de programmation à usage général, ce qui est une autre façon de dire qu'il peut être utilisé pour presque tout. Plus important encore, il s'agit d'un langage interprété, ce qui signifie que le code écrit n'est pas réellement traduit dans un format lisible par ordinateur au moment de l'exécution.



Platform IO

Platform IO est un outil professionnel multiplateforme, inter-architecture, multi-Framework, pour les ingénieurs de systèmes embarqués et pour les développeurs de logiciels qui écrivent des applications pour des produits embarqués.



VS Code

Visual Studio Code est un éditeur de code en termes simples. Visual Studio Code est « un éditeur libre qui aide le programmeur à écrire du code, aide au débogage et corrige le code à l'aide de la méthode intelli-sense ».



PyCharm

PyCharm est un environnement de développement intégré (IDE) utilisé en programmation informatique, créé pour le langage de programmation Python. Lorsque vous utilisez PyCharm sur Databricks, PyCharm crée par défaut un environnement virtuel Python, mais vous pouvez configurer pour créer un environnement Conda ou utiliser un environnement existant.



Tous d'abord il faut installer les logiciels précédant puis installer les packages qu'on va utiliser pour réaliser ce projet.

6. Les packages importants

Les packages nécessaires à notre réalisation sont :

6.1. Open cv

Open Computer Vision est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.



6.2. Dlib++

Dlib est une boîte à outils C++ moderne contenant des algorithmes et des outils d'apprentissage automatique permettant de créer des logiciels complexes en C++ pour résoudre des problèmes du monde réel.



6.3. CMake

CMake est un outil pour gérer la construction du code source. À l'origine, CMake a été conçu comme un générateur pour divers (dialectes of Makefile), aujourd'hui CMake génère des systèmes de construction modernes tels que Ninja ainsi que des fichiers de projet pour les IDE tels que Visual Studio et Xcode.



6.4. Tensorflow

TensorFlow est un outil open source d'apprentissage automatique développé par Google.

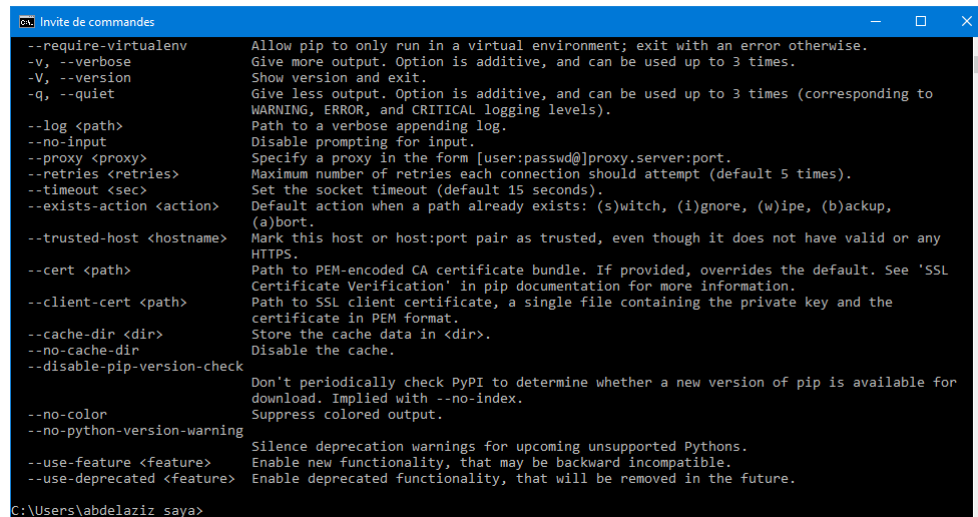


6.5. Keras

Keras est une bibliothèque logicielle open-source qui fournit une interface Python pour les réseaux de neurones artificiels. Keras sert d'interface pour la bibliothèque TensorFlow.



Ces package ou bibliothèques pour les installer il faut passé par le terminal cmd commande Window



```

C:\Users\abdelaziz saya>
--require-virtualenv Allow pip to only run in a virtual environment; exit with an error otherwise.
-v, --verbose Give more output. Option is additive, and can be used up to 3 times.
-V, --version Show version and exit.
-q, --quiet Give less output. Option is additive, and can be used up to 3 times (corresponding to
WARNING, ERROR, and CRITICAL logging levels).
--log <path> Path to a verbose appending log.
--no-input Disable prompting for input.
--proxy <proxy> Specify a proxy in the form [user:passwd@]proxy.server:port.
--retries <retries> Maximum number of retries each connection should attempt (default 5 times).
--timeout <sec> Set the socket timeout (default 15 seconds).
--exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
(a)bort.
--trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any
HTTPS.
--cert <path> Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL
Certificate Verification' in pip documentation for more information.
--client-cert <path> Path to SSL client certificate, a single file containing the private key and the
certificate in PEM format.
--cache-dir <dir> Store the cache data in <dir>.
--no-cache-dir Disable the cache.
--disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for
download. Implied with --no-index.
--no-color Suppress colored output.
--no-python-version-warning Silence deprecation warnings for upcoming unsupported Python3.
--use-feature <feature> Enable new functionality, that may be backward incompatible.
--use-deprecated <feature> Enable deprecated functionality, that will be removed in the future.

```

Figure 3. 10 : Commande window

7. Système de déverrouillage avec la reconnaissance faciale

La technologie de reconnaissance faciale peut être utilisée dans une large gamme d'applications telles que l'authentification d'identité, le contrôle d'accès et la surveillance. Les intérêts et les activités de recherche en reconnaissance faciale ont considérablement augmenté au cours des dernières années [13].

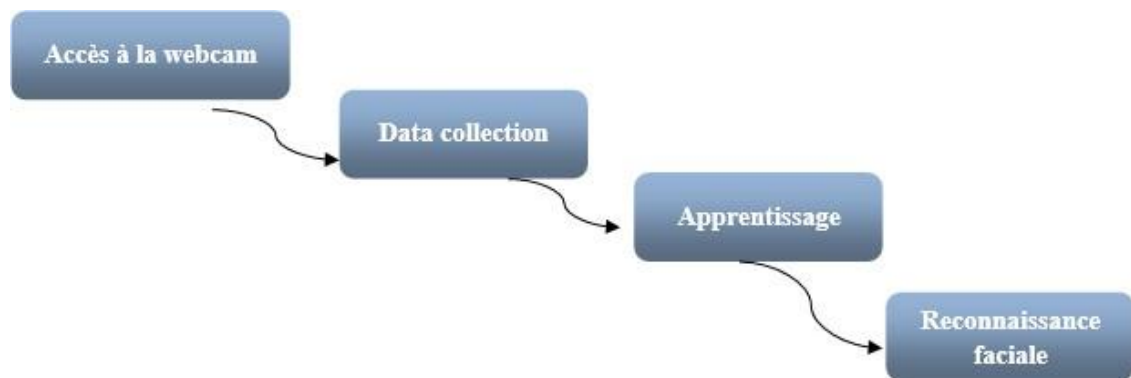


Figure 3. 11 : Les étapes de la reconnaissance faciale

Pour réaliser la reconnaissance faciale il faut suivre les étapes suivantes :

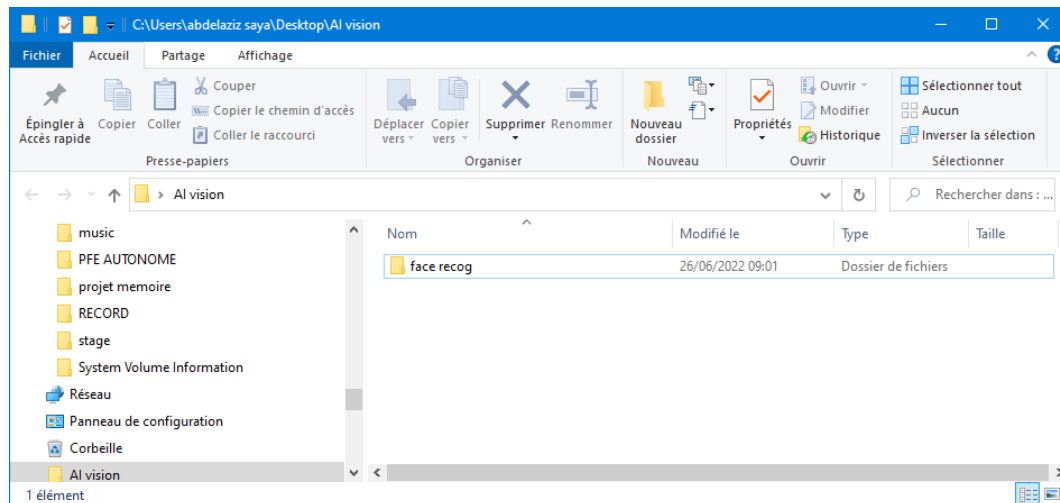


Figure 3. 12 : Création fichier face recog

Crée un fichier face recog tous les programme doit être créé à l'intérieur de ce fichier

- Code pour accéder à la webcam open cv
- Code Identification de visage
- Code pour l'apprentissage automatique (training)
- Code Reconnaissance faciale
- Fichier Data base

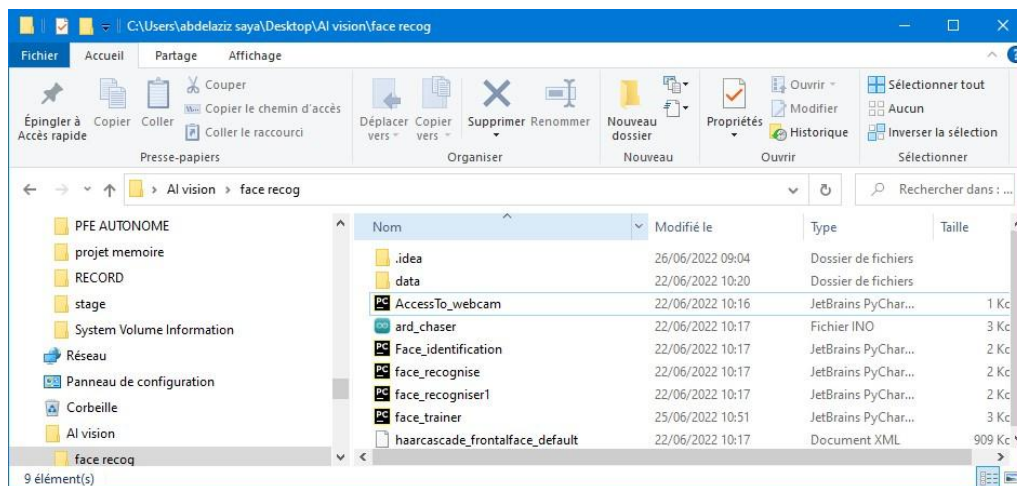


Figure 3. 13: Fichier qui contient les programmes importants

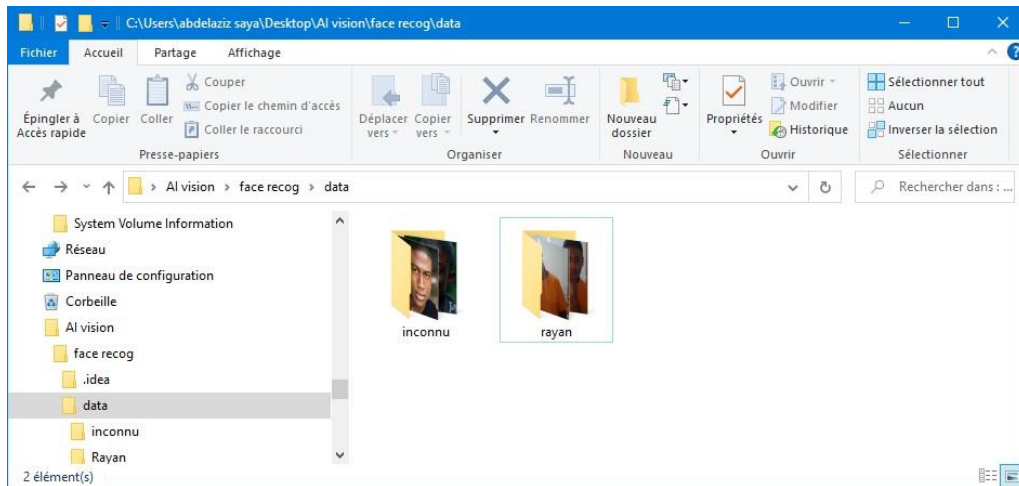


Figure 3. 14 : Data base

Il important de créer les deux fichiers à travers c'est deux fichier python peut classifier entre visage reconnu et l'inconnu on lui donne des input et output c'est le concept de L'apprentissage supervisé photo ci-dessous Figure 4.5 montre un exemple sur L'apprentissage supervisé.

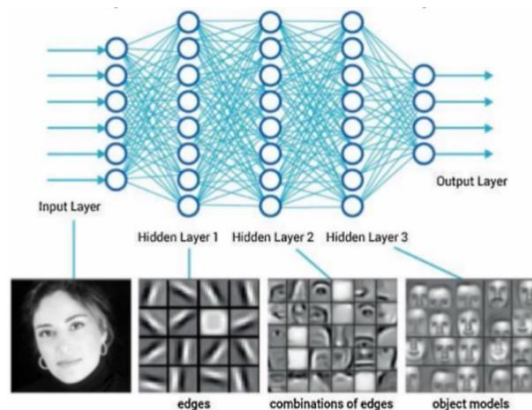


Figure 3. 15 : L'apprentissage supervisé

Les bibliothèques utilisées

- Open cv
- Numpy
- Pillow
- Dlib++
- C Make

7.1. Test de la reconnaissance faciale

Sur la figure ci-dessous nous donnons l'étape de test de la reconnaissance faciale.

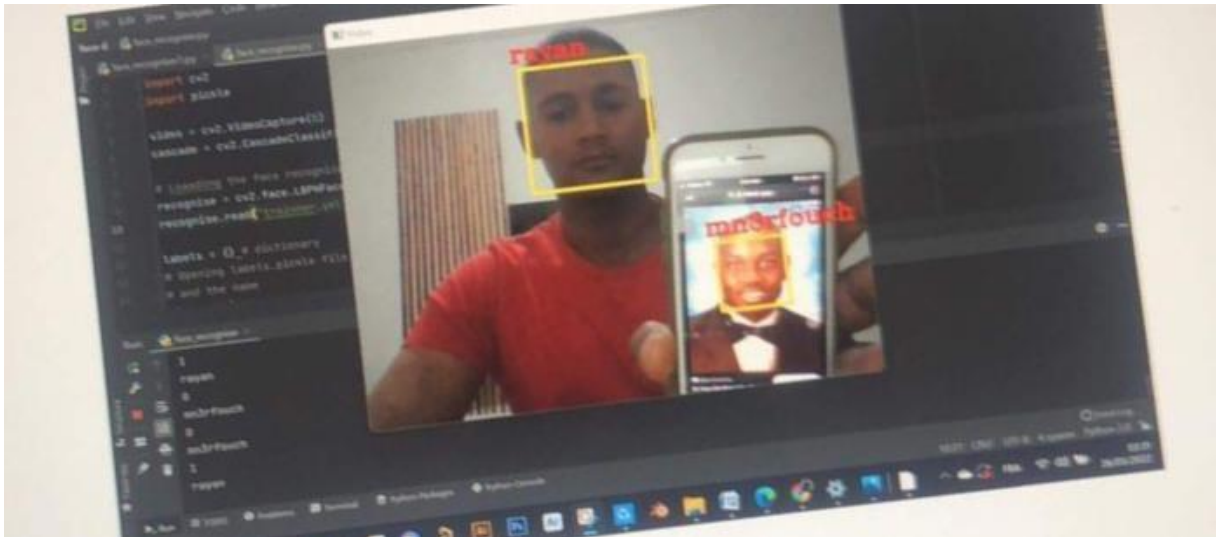


Figure 3. 16 : Test de l'étape de la reconnaissance faciale

Ce que on voit dans la photo c'est la reconnaissance faciale, ce qui fait qu'on apprend à la machine de reconnaître un visage est sa c'est fait grâce à machine Learning à travers les étapes suivant :

- **Collection des données**
- **Apprentissage (Learning)**
- **Face recognition**

L'apprentissage supervisé, également connu sous le nom d'apprentissage automatique supervisé (Supervised machine Learning), est une sous-catégorie de l'apprentissage automatique (machine Learning) et de l'intelligence artificielle. Il est défini par son utilisation d'ensembles de données étiquetés pour former des algorithmes permettant de classer les données ou de prédire les résultats avec précision.

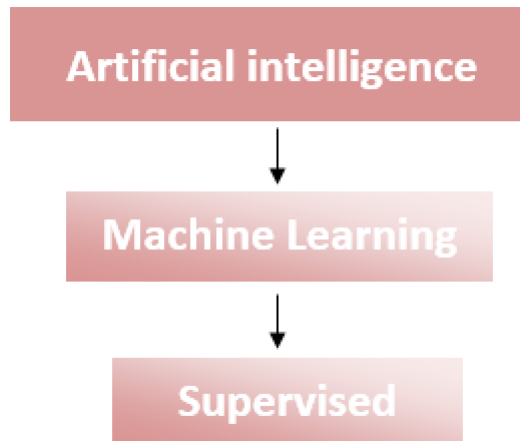


Figure 3. 17 : Supervised fait partie du machine Learning et AI

8. Système de conduite autonome

Dans cette section, nous allons introduire la partie concernant la conduite autonome.

8.1. La détection automatique des voies et des objets

La détection automatique des voies est une technologie cruciale qui permet aux voitures autonomes de se positionner correctement dans un environnement de conduite urbaine à plusieurs voies de nombreuses méthodes de vision par ordinateur ont été développées dans le but de résoudre le problème de détection de voie.

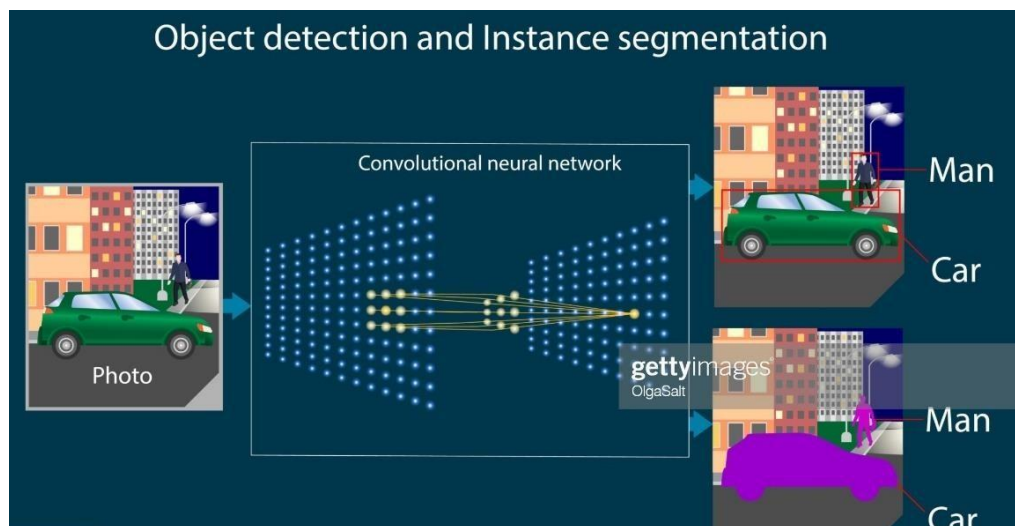


Figure 3. 18 : La détection automatique des objets

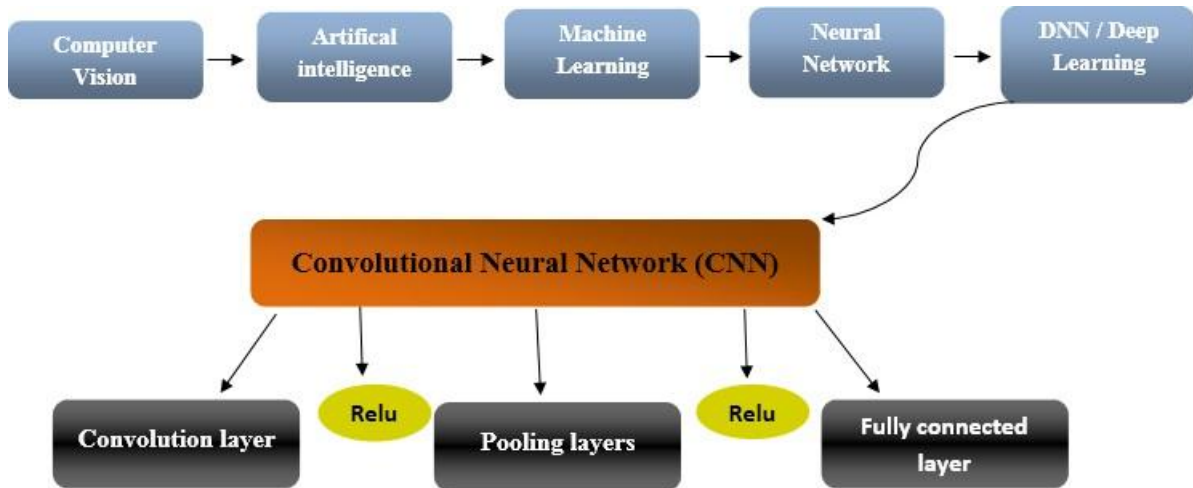


Figure 3. 19 : Architecture CNN

La convolution est une opération mathématique simple généralement utilisée pour le traitement et la reconnaissance d'images. Sur une image, son effet s'assimile à un filtrage.

La conduite Autonome



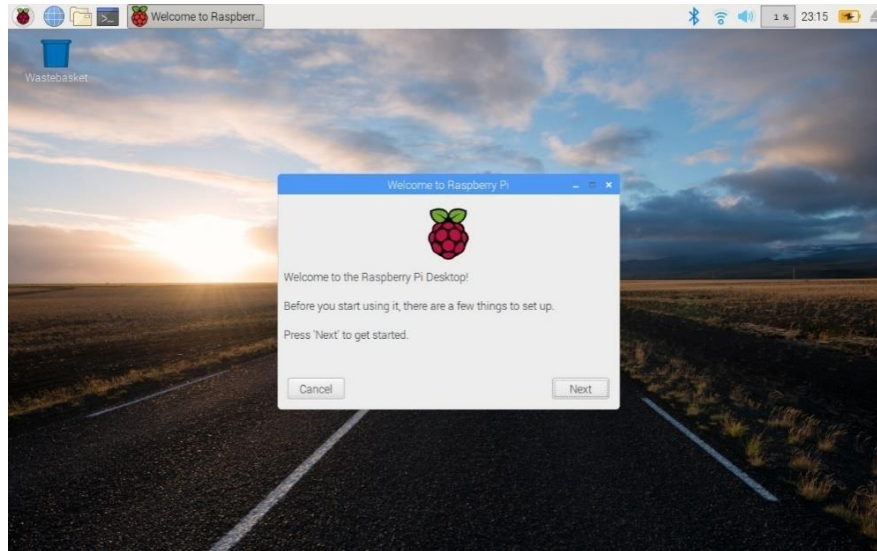
Figure 3. 20 :Tensorflow & keras

Il est important d'utiliser ces deux bibliothèques qui marchent ensemble ils sont très recommandés sur tout pour CNN Convolutionnal neural network

Cette fois on va écrire le code ou coder dans un système d'exploitation linux de Raspberry-Pi.

Donc la première partie : il faut installer Raspbian.

Raspbian : est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry Pi.



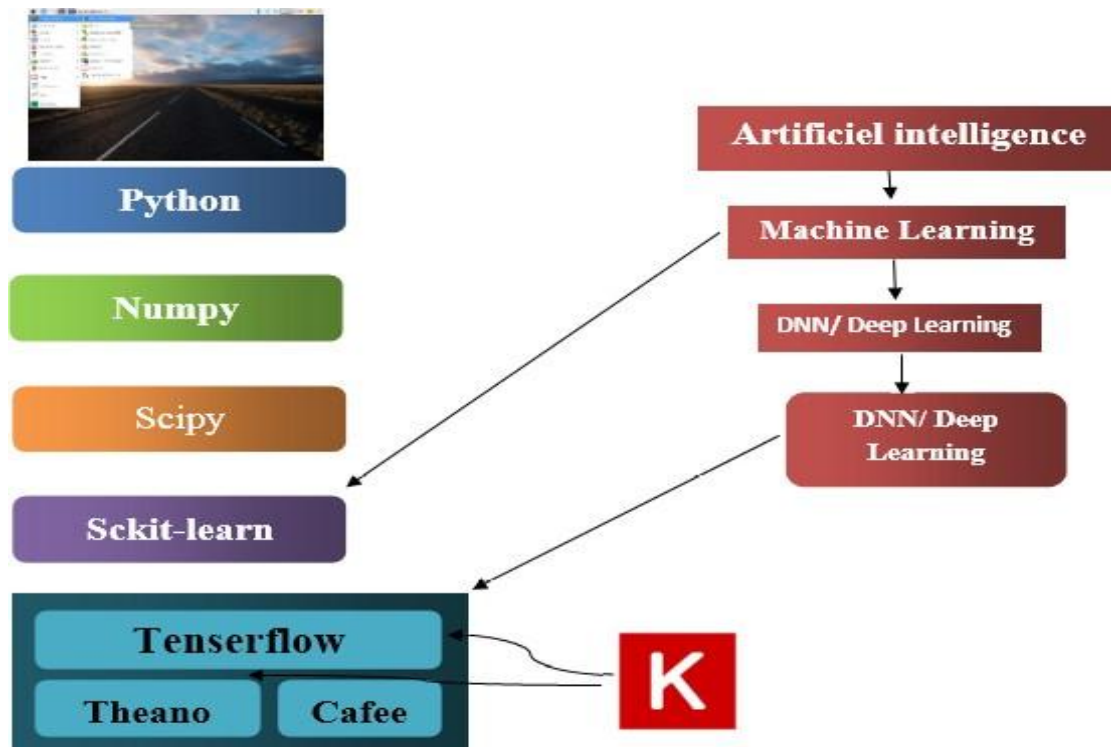


Figure 3. 22 : Etapes à suivre sur linux pour la conduite autonome

Ce qu'on voit ici c'est pour simplifier après télécharger python et IDE on passe à l'installation des bibliothèques comme Numpy, Scipy et sckit-learn ce sont les bibliothèques responsables de l'apprentissage automatique en suit Tensorflow, Theano et Cafee ce sont les bibliothèques responsables de Deep Learning et parmi eux la fameusekeras.

Après qu'on programmer notre Raspberry Pi on programmer le Uc ARM qui est Arduino duo est les connecté avec le câble sériel qu'on la caméra-pi envoi les infos au raspbrry pi, le raspbrry téléverse les infos au Uc qui est connecte à un H Bridge contrôleur de moteur.

8.2. Détection des voies

Pour la détection des voies, après traitement de l'image, nous avons trouvé la distance des pixels blancs (255). Pour cela, les pixels sont traités comme des vecteurs pour retrouver leur Positions.

Si la distance des pixels blancs augmente les commandes appropriées sont envoyé au microcontrôleur pour prendre le virage à droite. Si la distance diminue il indique au microcontrôleur pour prendre le tourner à gauche.

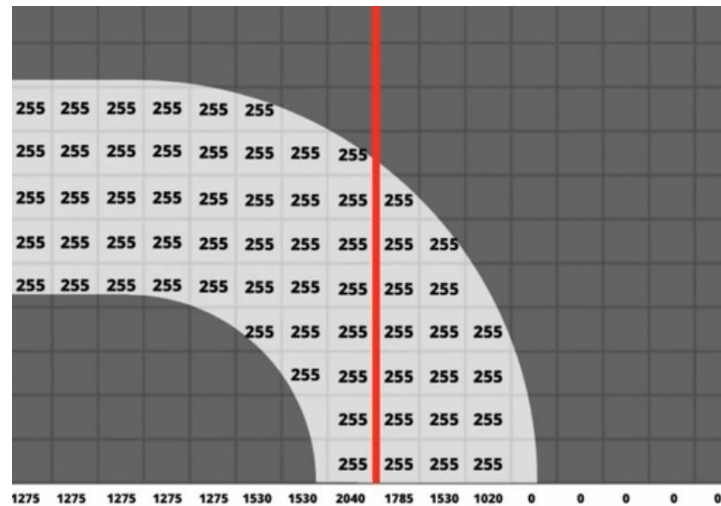


Figure 3. 23 : Sommmation des pixels

Les 255 représente les pixels en blanc le 0 en noire les line de la route sont en blanc donc notre système traite les lignes blanches en pixels (255).

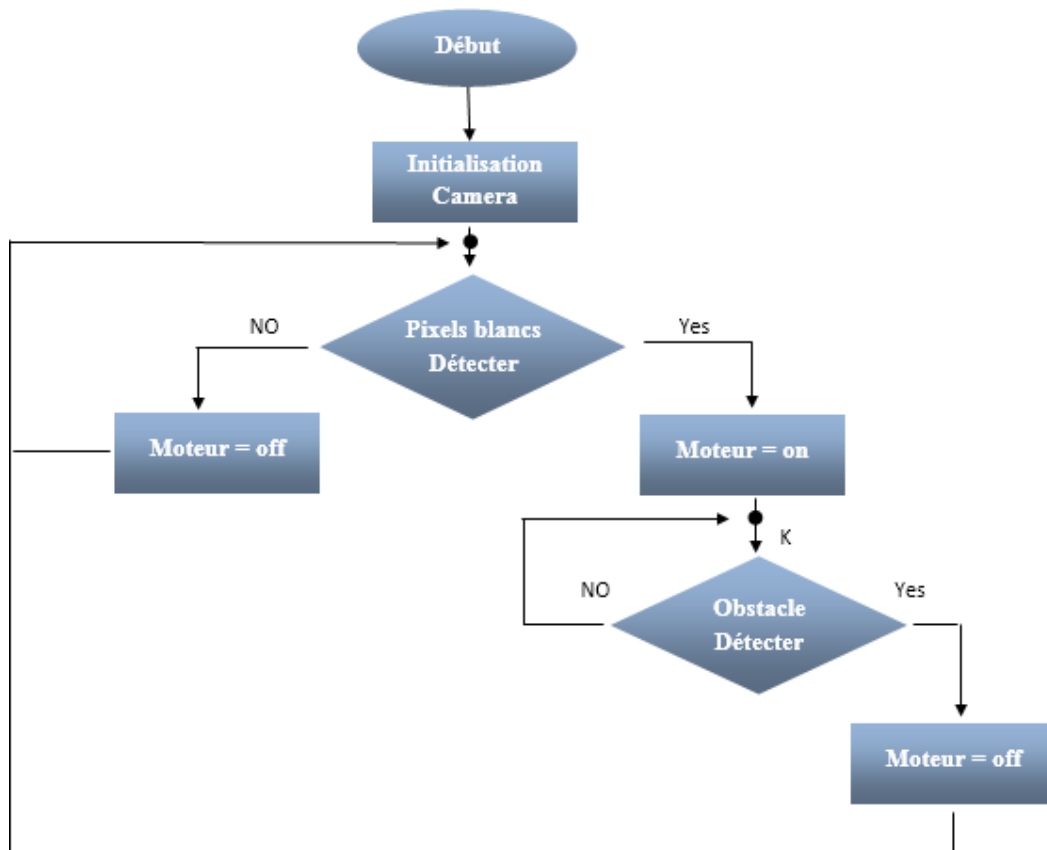


Figure 3. 24: Principe de fonctionnement du conduit autonome

Si la caméra de la voiture détecte les lignes ou les voix de la route la voiture roule et si elle détecte un obstacle elle s'arrête si non elle ne démarre pas.

9. Le système de sécurité

Par contre dans cette partie y a beaucoup de hardware que du software du coup pour écrire notre programme il suffit d'installer un éditeur texte.

De ma part j'ai utilisé platform IO + vs code ils sont l'équivalent de Arduino ide sauf qu'utilisé ce dernier permet de programmer plus rapide logiciel est plus rapide on trouve beaucoup de bibliothèques intégrées et des types de carte comme esp32 etc. sans les télécharger.



Figure 3. 25 : Platform IO +vs Code c'est l'équivalent d'ide Arduino

9.1. Conception de circuit imprimé

Pour les circuits imprimés nous avons utilisé **EasyEDA**. **EasyEDA** : est une suite d'outils EDA basée sur le Web qui permet aux ingénieurs en matériel de concevoir, simuler, partager - publiquement et en privé - et discuter de schémas, de simulations et de cartes de circuits imprimés. Les autres fonctionnalités incluent la création d'une nomenclature, de fichiers Gerber et de fichiers Pick and place et de sorties documentaires aux formats PDF, PNG et SVG.



9.2. Conception des design 3D

Pour les conceptions 3D j'ai utilisé **blender**

Blender : est un ensemble d'outils logiciels d'infographie et d'animation 3D gratuits et open-source utilisés pour créer des courts métrages et des films d'animation animés, des effets visuels, des modèles imprimés en 3D, des applications 3D interactives, de la réalité virtuelle et des jeux informatiques.



Sur la figure ci-dessous nous montrons notre conception du PCB en 3D.

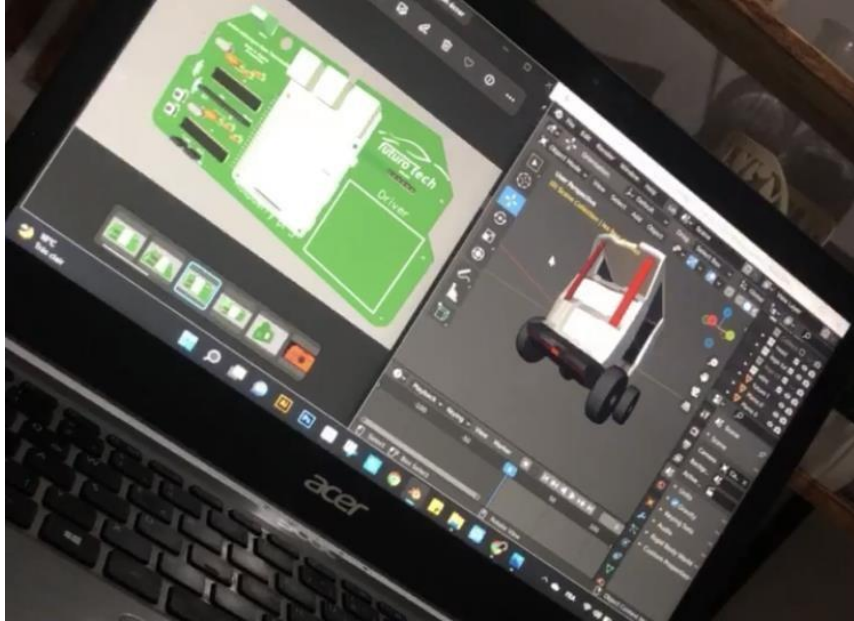


Figure 3. 26 : Conception PCB & 3D Design

10. Realisation du prototype

Sur la figure (3.27) des LEDs sont ajoutés au projet. Ces LEDs s'allument lorsque la voiture est arrêtée, et s'éteignent lorsque le véhicule se trouve en état de marche.



Figure 3. 27 : LED stop de véhicule

Sur la figure ci-dessous nous donnons l'implémentation du circuit réalisé.



Figure 3. 28 : Montage du système sur prototype

Le système peut être implémenté sur n'importe véhicule comme par exemple ce prototype de robot livreur, comme il peut être beaucoup plus volumineux et l'implémenté sur des grands véhicules.

11. Conclusion

Dans ce chapitre nous avons expliqué en détail la partie software et hardware de notre projet, qui représente un système de sécurité très complexe implémenté dans les véhicules. Il s'agit du système anti incendie et le système de déverrouillage.

Conclusion générale

Les systèmes embarqués sont de nos jours largement utilisés dans de multiples domaines d'applications. Leur rôle est d'exercer une tâche précise de manière autonome, sans intervention humaine souvent en temps réel,

Le système possède également de nombreuses autres contraintes : Espace mémoire, Sécurité, Puissance CPU, Autonomie, dimensions physiques etc. Ils sont dans la majeure partie du temps.

Dès l'apparition des premières automobiles, la sécurité a été une préoccupation majeure des ingénieurs et des pouvoirs publics. Dans ce mémoire, nous nous sommes intéressés au problème d'implémenter des systèmes complexe basé sur deep Learning et AI sur une puce, notre travail consiste à implémenter un système de sécurité dans les véhicules.

L'apprentissage en profondeur a déjà été intégré à la plupart des pipelines à la pointe de la technologie. Cette évolution a entraîné une augmentation massive de la précision des systèmes et a entraîné la saturation du critère de référence « standard » actuel.

D'après les résultats obtenus, nous pouvons affirmer qu'on peut implémenter des systèmes complexes basés sur le deep Learning et l'intelligence artificiel sur une puce.

L'utilisation du CNN dans la reconnaissance faciale, détection de flamme et même dans la conduite autonome a donné un très bon résultat et bonne performance.

Dans le futur nous pouvons améliorer notre projet :

- Réduire la taille du PCB en utilisant un seul CPU puissant.
- Une meilleure détection basée sur le Deep Learning.

En fin, un ingénieur en système embarqué doit avoir une double compétence en électronique et en logiciel, où il doit maîtriser les langages de programmation Embedded C, C++, python etc.

Bibliographie

- [1] <https://www.wsp.com/fr-GL/secteurs/vehicules-connectes-et-autonomes>. Consulté le 28/02/2022
- [2] <https://www.intel.fr/content/www/fr/fr/robotics/real-time-systems.html>. Consulté le 12/02/2022
- [3] Xiao, Perr, Desining Embedded Systems and the Internet of Things (IOT), Edition Wiley, 2018.
- [4] <https://fr.parasoft.com/blog/what-is-an-embedded-system/> . Consulté le 12/02/2022
- [5] <https://www.autodesk.fr/solutions/pcb-design-software>. Consulté le 12/02/2022
- [6] <https://www.comptoir-hardware.com/actus/cartes-meres/42643-des-pcb-avec-le-moins-de-defaults-possible-un-des-nouveaux-besoins-du-hardware.html>. Consulté le 12/02/2022
- [7] <https://www.nvidia.com/fr-fr/self-driving-cars/> . Consulté le 12/02/2022
- [8] <https://www.usherbrooke.ca/grams/expertises/conception-circuits-imprimes-pcb> Consulté le 12/02/2022
- [9] http://www.aptustech.com/services/integration_meca/ . Consulté le 12/02/2022.
- [10] Radovan, Miucic, Connected Vehicles: Intelligent Transportation Systems 2018, Wayne State University, Edition springer, 2018.
- [11] Weisong Shi, Liangkai Liu, Computing Systems for Autonomous Driving. Edition springer, 2021.
- [12] <https://www.geeksforgeeks.org/difference-between-avr-and-arm/> . Consulté le 12/02/2022
- [13] <https://www.goengineer.com/solidworks/electrical-design/electrical> Consulté le 12/02/2022
- [14] H. A. Abdallah Dafallah, "Design and implementation of an accurate real time GPS tracking system," The Third International Conference on e-Technologies and Networks for Development (ICeND2014), 2014, pp. 183-188, doi: 10.1109/ICeND.2014.6991376.
- [15] Guodong Guo, S. Z. Li and Kapluk Chan, "Face recognition by support vector machines," Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 2000, pp. 196-201, doi: 10.1109/AFGR.2000.840634.
- [16] <https://embeddedschool.in/distinguish-between-avr-and-arm-microcontroller/> . Consulté le 12/02/2022