



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique
Université El-wancharissi de Tissemsilt



Faculté des Sciences et de la Technologie
Département de Sciences et de la Technologie

Mémoire de fin d'études pour l'obtention du diplôme
de Master académique en

Filière : **Électronique**

Spécialité : **Instrumentation**

Présentée par : **FERNANE ABDERAZAK**

CHEMLAL OMAR

Thème

COMMANDE À DOMICILE À L'AIDE DE L'ASSISTANT VOCAL

Soutenu le 12/07/2021

Devant le Jury :

Bachir NAIL	Président	M.C.B	Univ-Tissemsilt
Aoued MEHARRAR	Examineur	M.C.A	Univ-Tissemsilt
Mostefa HAMDANI	Encadreur	M.C.B	Univ-Tissemsilt

Année universitaire : 2020-2021



ملخص

الهدف من عملنا هو دراسة إمكانية تشغيل / إيقاف تشغيل الأجهزة الكهربائية بناءً على الكلام (الأمر). يُحدث هذا النظام تطوراً في نظام الأتمتة الذي يدمج (الإنسان والآلة) كواجهة تنظيمية. يتكون من نظام تحكم يعتمد على نقل الإشارات وفقاً للإدخال الذي يتم تحديده بناءً على الأوامر الصوتية التي يقدمها المستخدم عبر الميكروفون.

أظهرت النتائج التجريبية أن هذا النظام يدمج معالجة البيانات التي تقدم نفسها في شكل شبكات عصبية وإلكترونيات بواسطة متحكمات دقيقة.

تسمح لنا النتائج التي تم الحصول عليها بمعرفة وظائف خوارزميات الذكاء الاصطناعي والتيها باستخدام المتحكمات الدقيقة.

الكلمات المفتاحية: ذكاء اصطناعي، شبكات عصبية، متحكم دقيق، معالجة صوتية.

RÉSUMÉ

L'objectif de notre travail est d'étudier la possibilité d'allumer / éteindre les appareils électriques en fonction de parole (commande). Ce système crée une évolution dans le Système d'automatisation et qui intègre (l'homme-machine) comme une interface règlementaire. Il se compose d'un système de contrôle basé sur les microcontrôleur raspberry pi et arduino pour transmettre les signaux en fonction de l'entrée étant sélectionné sur la base des commandes vocales données par l'utilisateur via microphone.

Les résultats expérimentaux ont montré que ce système intègre informatique qui ce presente sou forme de réseaux de neurones et l'électronique par les microcontrôleur.

Les résultats obtenus nous permettent de connaître la fonctionnalité des algorithmes de intelligence artificiel et leur mécanisme avec les microcontrôleur.

Keywords : Artificial intelligence, Neural networks, microcontroller, Voice processing.

ABSTRACT

The objective of our work is to study the possibility of turning on / off electrical devices based on speech (command). This system creates an evolution in the Automation System which integrates (man-machine) as a regulatory interface. It consists of a control system based on the raspberry pi and arduino microcontroller to transmit signals according to the input being selected based on voice commands given by the user via microphone.

The experimental results have shown that this system integrates data processing which presents itself in the form of neural networks and electronics by microcontrollers.

The results obtained allow us to know the functionality of artificial intelligence algorithms and their mechanism with microcontrollers.

Keywords. Cloud Computing, Load Balancing, Fuzzy logic, Clustering, K-means

DÉDICACE

Je dédie ce modeste travail

À mes parents

Aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler que Dieu leur procure bonne santé et longue vie.

*À ceux que j'aime beaucoup et qui m'ont soutenue durant toute ma vie mes frères
Sans oublier mes grands parents ainsi que toute ma chère famille et mes amis.*

À tous ceux qui ont contribué de près ou de loin pour que ce travail soit possible. Je vous dis merci

Abderazek

Lieu, le 4 juillet 2021.

DÉDICACE

Je dédie ce modeste travail

À mes parents

Aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler que Dieu leur procure bonne santé et longue vie.

*À ceux que j'aime beaucoup et qui m'ont soutenue durant toute ma vie mes amis
Sans oublier mes grands parents ainsi que toute ma chère famille.*

À tous ceux qui ont contribué de près ou de loin pour que ce travail soit possible. Je vous dis merci

À tous mes collègues de la spécialité Instrumentation

En fin, je le dédie à ma très chère Rania qu'elle est toujours à mes côtés

Omar

Lieu, le 4 juillet 2021.

REMERCIEMENT

En premier lieu, nous tenons à remercier notre dieu, notre créateur, pour le courage et la patience qu'il nous a donné pour accomplir ce travail.

Nous remercions très sincèrement le président de jury et les membres de jury ayant acceptés d'évaluer ce travail.

Nous remercions notre encadreur Mr Hamdani Mostapha pour ses orientations ses conseils, sa disponibilité à tous épreuve, pour sa gentillesse et sa patience, ses remarques fructueuses. Tout notre respect et notre gratitude, Merci.

Nous exprimons notre profonde gratitude et nos remerciements à nos professeurs pour la qualité de l'enseignant qu'ils nous ont prodigué au cours de ces cinq années passées à l'universitaire de Tissemsilt.

Nous remercions tous nos collègues de la promotion Électronique.

Nos derniers remerciements et qui ne sont pas les moindres, vont à tous ceux qui ont contribué de près ou de loin pour l'aboutissement de ce travail.

Abderazek et Omar

Lieu, le 4 juillet 2021.

Table des matières

Table des matières	VII
Liste des tableaux	XI
Table des figures	XII
Liste Des Algorithmes	XIV
Introduction Générale	1
1 Introduction	1
2 Contexte	3
3 Problématique	3
4 Contributions	3
5 Organisation du mémoire	4
I CONTEXTE ET ÉTAT DE L'ART	5
CHAPITRE 1 INTELLIGENCE ARTIFICIELLE ET SYSTÈME DE COMMANDE	6
Intelligence artificielle Et Système de commande	6
1.1 Introduction	7
1.2 L'intelligence artificielle	7
1.2.1 Origines de l'IA	8
1.2.2 Objectifs de l'IA	8
1.2.3 Fonctionnement de L'IA	8
1.2.4 Qu'est-ce qu'une prise de décision ?	9
1.2.5 Que fait une machine pour décider ?	10
1.2.6 Programmation sans et avec AI	10
1.3 Applications de l'IA	12
1.3.1 Traitement du langage naturel	12

1.3.2	Systèmes experts	12
1.3.3	Systèmes de vision	12
1.3.4	Reconnaissance de la parole	12
1.3.5	Robots intelligents	13
1.4	Domaines d'utilisation de l'IA	13
1.4.1	Domaines d'applications de l'IA dans les Smart homes	14
1.5	Réseaux de neurones	15
1.5.1	Les réseaux de neurones formels	16
1.6	Types des réseaux de neurones artificiels (RNA)	18
1.6.1	Les Réseaux Proactifs (monocouche)	18
1.6.2	Les Réseaux Proactifs Multi Couches (PMC)	19
1.6.3	Les Réseaux récurrents	19
1.7	Apprentissage	20
1.7.1	Apprentissage supervisé	23
1.7.2	Apprentissage non supervisé	24
1.8	Topologies des RNA	25
1.8.1	Tache des neurones Artificiels	26
1.9	Avantages et Inconvénients des réseaux de neurones	26
1.9.1	Avantage des réseaux de neurones	26
1.9.2	Inconvénients des réseaux de neurones	27
1.10	Réseau de neurones convolutifs	27
1.10.1	Architecture globale de CNN	28
1.10.2	Couche convolutive	28
1.10.3	Couche de pooling	29
1.10.4	Couche de correction (ReLu)	29
1.10.5	Avantages et inconvénients du CNN	30
1.11	Réseaux de neurones récurrents (RNN)	30
1.11.1	Fonctionnement général	31
1.11.2	Variantes d'apprentissage	33
1.12	Domotique	34
1.13	Traitement Vocal	34
1.13.1	Système de reconnaissance vocale dépendant du locuteur	35
CHAPITRE 2 MICROCONTRÔLEUR		37
Microcontrôleur		37
2.1	Introduction	38

2.2	Description d'un Microcontrôleur	38
2.2.1	Utilisation	40
2.2.2	Types de Microcontrôleur	40
2.3	Avantages des microcontrôleurs	40
2.4	Le choix d'un microcontrôleur	41
2.5	Présentation Du Raspberry Pi 3	41
2.5.1	Définition	41
2.5.2	Choix du Raspberry Pi	42
2.5.3	Les différentes parties d'un RaspberryPi 3	42
2.5.3.1	Fiche technique	43
2.5.3.2	Les spécifications Entrées / Sorties	44
2.5.3.3	Les ports GPIO :	44
2.5.4	L'intérêt du Raspberry PI 3 :	45
2.5.5	Les différentes utilisations du Raspberry Pi 3	45
2.5.5.1	L'utilisation MultiMedia	45
2.5.5.2	La domotique	46
2.5.5.3	Les systèmes embarqués	46
2.5.5.4	Un ordinateur, tout simplement	46
2.5.6	Avantage de Raspberry Pi 3	46
2.5.7	Performances et contraintes du Raspberry pi	47
2.6	Présentation d'Arduino	48
2.6.1	Définition	48
2.6.2	Caractéristiques de la carte Arduino UNO	48
2.6.3	Les différents composants de la carte Arduino UNO	49
2.6.3.1	Les sources d'alimentation de la carte Arduino UNO	49
2.6.3.2	Les ports de communications	50
2.7	Composants utilisés	51
2.7.1	Relais	51
2.7.2	Transistor	52
2.7.2.1	Transistor bipolaire	52
2.7.2.2	Transistor BC547	53
2.7.3	Module Bluetooth HC-05	53
II PARTIE EXPÉRIMENTALE		55
CHAPITRE 3 CONCEPTION ET MISE EN ŒUVRE		56

Conception et Mise en œuvre	56
3.1 Introduction	57
3.2 Schéma Bloc Globale	57
3.3 Systèmes Globale	57
3.4 Langage de Programmation Python	58
3.4.1 Installation des outils et premier programme Python	58
3.4.2 Côté logiciel	59
3.4.2.1 Python Pip	59
3.4.2.2 Installation sur RasPi	60
3.4.3 Présentation de «IDE » Arduino	61
3.4.4 Programmer avec Arduino	62
3.4.4.1 Structure d'un programme Arduino	62
3.5 Approche proposée	63
3.6 Conversion d'un Signal vocale à un texte	65
3.6.1 Class Bidirectional GRU	65
3.6.2 Class CNN	66
3.6.3 Class Speech Recognition	67
3.6.4 Code Principale	67
3.7 Explication du Code Arduino	69
CHAPITRE 4 RÉALISATION DE LA CARTE DE COMMANDE	71
Simulation et résultats	71
4.1 Introduction	72
4.2 Côté matériel (Hardware)	72
4.2.1 Équipements	72
4.2.2 Schémas électroniques globale du système	74
4.3 Fonctionnement	76
4.4 Réalisation de la carte PCB	77
4.5 Carte PCB 2D	79
4.6 Carte PCB 3D	80
4.7 Essai pratique	83
4.7.1 Le Cas D'allumage de la lampe	84
4.7.2 Le Cas d'éteindre la lampe	85
4.8 Perspectives	87
Bibliographie	88

Liste des tableaux

1.1	Programmation sans et avec IA	11
1.2	Comparaison entre neurone Biologique et artificiel	16
1.3	Comparaison entre neurone Biologique et artificiel	21
1.4	Exemple de différents types d'apprentissages en mode supervisé	24
1.5	Exemple de différents types d'apprentissages en mode non supervisé	25
1.6	Avantages et inconvénients du CNN	30
2.1	Fiche Technique Raspberry Pi	43
4.1	Équipements Générale	73

Table des figures

1.1	Décision de la machine	10
1.2	l'intelligence artificielle faible et forte	11
1.3	Panorama des domaines de l'intelligence artificielle	13
1.4	Relations entre l'intelligence artificielle et l'intelligence ambiante	14
1.5	Différence entre Neurone Biologique / Neurone Artificiel	15
1.6	Neurone artificiel	16
1.7	Structure générale du neurone formel	17
1.8	Réseau proactif monocouche (Perceptron)	18
1.9	Perceptron Multi-couches (MLP)	19
1.10	Réseau récurrent avec neurones cachés	20
1.11	Expérience de Pavlov	22
1.12	Exemple d'apprentissage	23
1.13	Topologies des réseaux de neurones artificiels	25
1.14	Réseau neuronal convolutif	28
1.15	Architecture simple de CNN	28
1.16	Exemple de max-pooling	29
1.17	Schéma d'un réseau de neurones récurrents à droite la version « dépliée » de la structure.	31
1.18	Représentation compacte des RNN	31
1.19	Représentation dépliée des RNN	32
1.20	Représentation dépliée hiérarchisée des RNN	33
1.21	Mécanisme du système de reconnaissance vocale	35
2.1	Schéma type de tout appareil programmable	39
2.2	Raspberry pi 3	42
2.3	Les différentes parties de Raspberry Pi 3	42
2.4	Les ports GPIO de Raspberry Pi 3	45
2.5	Arduino Uno	48
2.6	Caractéristiques Techniques De La Carte Arduino Uno	49

2.7	Description des entrées/sorties de la carte Arduino Uno	50
2.8	Schéma Complet Des Broches	51
2.9	Relais	52
2.10	Différent Types De Transistor	53
2.11	Module Bluetooth HC-05	54
3.1	Schéma Bloque Globale du Système	57
3.2	Système Globale RPi et Arduino	58
3.3	Installation Du pip Python Validé	60
3.4	Logiciel Du Raspberry Pi	61
3.5	L'interface visuelle du IDE Arduino	62
3.6	Structuration d'un programme Par Arduino	63
3.7	Organigramme du déroulement du programme	64
3.8	Organigramme de contrôle	65
4.1	Aspect réel des composants utilisé	72
4.2	Alimentation	74
4.3	Schéma D'Entrée	75
4.4	Schéma de Sortie	75
4.5	Schéma électronique générale	76
4.6	Carte PCB avec un Arrière-plan Gris	77
4.7	Carte PCB avec un Arrière-plan Blanc	78
4.8	Carte PCB avec un Arrière-plan Noir	78
4.9	Carte PCB en 2ème Dimension	79
4.10	Carte PCB en 2ème Dimension avec composant Visible	79
4.11	Phase Arriérer De la Carte PCB en 2ème Dimension	80
4.12	Carte PCB en 3ème Dimension 'Coté Gauche'	80
4.13	Carte PCB en 3ème Dimension 'Coté Droit'	81
4.14	Typon imprimé dans la plaque de cuivre	81
4.15	Baignade de la plaque dans le perchlorure de fer	82
4.16	Circuit électronique de la plaque	82
4.17	L'interface de la plaque Électronique	83
4.18	Premier Essai démonstrative	83
4.19	Exécution avec Cmd pour la commande ON	84
4.20	Allumage de la lampe	84
4.21	Exécution avec Cmd pour la commande OFF	85
4.22	Lampe éteinte	85

LISTE DES ALGORITHMES

CB-GRU Class Bidirectional GRU	65
C-CNN Class CNN	66
C-SR Class Speech Recognition	67
C-P Code Principale	67
C-Arduino Code Arduino	69

GLOSSAIRE

IA Intelligence Artificielle	7
RN Réseau de Neurones	15
RNA Réseaux de Neurones Artificiels	18
PMC Réseaux Proactifs Multi Couches	18
MLP Multi-Layers Perceptron	18
RNN Réseaux de neurones récurrents	30
TAL Traitement Automatique des Langues	30
CNN Convolutional Neural Networks	27

INTRODUCTION GÉNÉRALE

Sommaire

1	Introduction	1
2	Contexte	3
3	Problématique	3
4	Contributions	3
5	Organisation du mémoire	4

1 Introduction

LA maison est un endroit particulièrement important pour tout le monde, c'est là que nous vivons et revenons. Certaines personnes (en particulier les personnes âgées) passent beaucoup de temps à la maison, donc l'habitat a une grande influence sur la qualité de vie.

La fonction principale de la maison est de protéger les résidents des effets des changements climatiques et de fournir un abri équipée de services pour une vie confortable, Ces fonctions ont maintenant évolué pour aboutir à fournir un environnement intelligent et confortable en toutes saisons et assurer la sécurité des personnes et des biens.

Améliorer le sentiment de sécurité et de confort à la maison semble être une tâche de grande importance sociale. Ces dernières années, la technologie a été utilisée pour créer des maisons intelligentes. La Smart House est définie comme une

maison équipée de technologies de télécommunications électroniques, automatiques, informatiques et environnementales, visant à fournir aux résidents une assistance dans diverses situations de la vie domestique [1].

Le mot domotique vient de « Domus », le domicile en latin, associé au suffixe « tique » qui fait référence à la technique. La domotique regroupe donc l'ensemble des technologies de l'électronique, de l'informatique et des télécommunications qui sont utilisées dans les domiciles pour rendre ceux-ci plus « intelligents ». Elle vise à intégrer différents systèmes pouvant être connectés entre eux et à des réseaux internes et externes de communication. Parmi ces fonctions, on trouve notamment l'économie et la gestion d'énergie, l'information et la communication, la maîtrise du confort, la sécurité et l'assistance. Câblée ou fonctionnant par ondes radio, la domotique investit notre univers quotidien pour nous faciliter la vie. Dans ses applications les plus évoluées, la domotique met en réseau et coordonne le fonctionnement de différents types d'équipements ménagers, de travail et de loisir. Les applications possibles de la domotique concernent aussi bien la programmation, la surveillance, que le contrôle à distance [2].

Les maisons intelligentes, en revanche, sont souvent décrites de manière restrictive en termes techniques ou fonctionnels pour leur capacité à améliorer le confort, la commodité, la gestion de l'énergie et la sécurité. Les études sur les maisons intelligentes qui s'intéressent aux problèmes de contrôle et de sécurité le font généralement pour améliorer la convivialité des technologies de maison intelligente ou pour éliminer les obstacles pour une adoption plus large.

Techniquement la maison intelligente et aussi appelé domotique Elle consiste à mettre en place des réseaux reliant différents type d'équipements (électroménager, éclairage, hifi, équipement domotique, etc) Ainsi, elle regroupe tout un ensemble de services permettant l'intégration des technologies modernes.

Le concept de domotique fait l'objet de nombreuses définitions. De la maison intelligente à la maison communicante, toutes les gradations sont possibles. De fait, la domotique concerne l'application à l'habitat de toutes les technologies dites nouvelles pouvant s'intégrer à ce contexte [3].

2 Contexte

Cette maquette, à l'échelle d'un Play mobile, permettrait de présenter certaines fonctionnalités de la domotique à travers trois scénarios : serrure codée, contrôle par Bluetooth et détection d'intrus. Ces scénarios seront automatisés via un micro contrôleur exécutant des programmes informatiques. Nous nous sommes alors lancés dans ce vaste projet, soit la fabrication de la carte pour commande de l'assistant vocale et la programmation de nos scénarios.

3 Problématique

Dans le monde numérique d'aujourd'hui, La technologie à modéré dans la vie quotidienne par une façon obligatoire ou pour le divertissement, et ce système de commande est automatiquement une procédure standard que nous avons décidé d'étudier pour qu'elle soit unique et personnalisé ,facile et plus principalement confortable c'est pour ça qu'on a utilisé la reconnaissance vocale dans le fonctionnement de ce système.

Les différents algorithmes de l'intelligence artificielle pour arriver à placer la reconnaissance vocale dans ce système, ensuite un transfert de signal qui permet l'application dans la maison.

4 Contributions

C'est dans ce contexte générale que se propose cette étude ayant pour objectif principal l'étude de la possibilité d'utiliser l'électronique et l'informatique pour but :

- Utilisation d'un ensemble de théorèmes dans l'électronique celle du courant et de la tension.
- Implémentation des algorithmes de l'intelligence artificielle.
- Utilisation des fonctions mathématiques avec bien sûr le traitement signal.

5 Organisation du mémoire

Ce mémoire est organisé en deux parties. La première partie a pour objectif la présentation des concepts, méthodes et outils nécessaires pour répondre à notre problématique. Elle comporte deux chapitres.

- **Chapitre I** : Ce chapitre présente une étude bibliographique sur l'intelligence artificiel et le réseaux de neurones en générale et aussi les synthèses la reconnaissance vocale.
- **Chapitre II** : Notions théoriques sur le microlocalement (arduino raspberry pi) et l'étude des composants utilisé.
- **Chapitre III** : ce chapitre représente le coté matérielle et l'application des méthodes utilisées lors de cette étude par logiciels et simulation.
- **Chapitre IV** : ce chapitre concerne la réalisation de la carte fonctionnelle de notre système et la discussion de nos résultats.

Première partie

CONTEXTE ET ÉTAT DE L'ART

CHAPITRE 1

INTELLIGENCE ARTIFICIELLE ET SYSTÈME DE COMMANDE

Sommaire

1.1	Introduction	7
1.2	L'intelligence artificielle	7
1.3	Applications de l'IA	12
1.4	Domaines d'utilisation de l'IA	13
1.5	Réseaux de neurones	15
1.6	Types des réseaux de neurones artificiels (RNA)	18
1.7	Apprentissage	20
1.8	Topologies des RNA	25
1.9	Avantages et Inconvénients des réseaux de neurones	26
1.10	Réseau de neurones convolutifs	27
1.11	Réseaux de neurones récurrents (RNN)	30
1.12	Domotique	34
1.13	Traitement Vocal	34

1.1 Introduction

L'intelligence artificielle implique deux idées de base. Premièrement, il s'agit d'étudier les processus de pensée des êtres humains. Deuxièmement, représenter ces processus via des machines (comme des ordinateurs, des robots, etc...).

L'application de l'intelligence artificielle nécessite des connaissances en perception, raisonnement, traitement de l'information et apprentissage. Elle apporte chaque jour une part de ses découvertes dans différents domaines techniques et non techniques, tels que l'économie, la sociologie, la médecine, l'agriculture, l'industrie générale et toutes les infrastructures technologique.

1.2 L'intelligence artificielle

Selon le père de l'intelligence artificielle John McCarthy, il s'agit de «la science et l'ingénierie de la fabrication des machines intelligentes, en particulier de programmes informatiques intelligents » [4].

L'Intelligence Artificielle (IA) est une discipline de l'informatique qui à pour but de créer des machines intelligentes, en "opposition" avec l'intelligence naturelle des êtres vivants. Le terme à beaucoup évolué au fil du temps, il englobe dorénavant toutes les idées visant à permettre à une machine de pouvoir émuler les capacités cognitives de l'Homme, et de les surpasser. Ce terme "d'intelligence artificielle" voit le jour en 1956 après les nombreux travaux débutés pendant la Seconde Guerre Mondiale et constitue l'un des plus récents champs d'études parmi les sciences et l'ingénierie [5].

Par conséquent, ce sera un ordinateur ou une machine avec un programme qui a des performances similaires à l'intelligence humaine et peut même être amplifié par la technologie.

Ces machines sont en mesure de :

- Raisonner
- Traiter de grandes quantités de données
- Discerner des modèles indétectables par l'œil d'un humain Comprendre et analyser ces modèles

- Interagir avec l'Homme
- Apprendre progressivement
- Améliorer continuellement ses performances [6]

1.2.1 Origines de l'IA

Les origines de l'IA remontent à la mythologie grecque, où des histoires mentionnent un homme mécanique capable de mimer le comportement humain. Toutefois, la quête pour le développement de l'IA semble devenir possible pendant la Seconde Guerre mondiale, lorsque les scientifiques de nombreuses disciplines, notamment des domaines émergents de la neurosciences et de l'informatique, ont travaillé ensemble pour s'atteler à la question des machines intelligentes [7].

Les premières traces de l'IA remontent à 1950 dans un article d'Alan Turing intitulé "Computing Machinery and Intelligence" dans lequel le mathématicien explore le problème de définir si une machine est consciente ou non. De cet article découlera ce que l'on appelle aujourd'hui le Test de Turing qui permet d'évaluer la capacité d'une machine à tenir une conversation humaine.

L'officialisation de l'intelligence artificielle comme véritable domaine scientifique date de 1956 lors d'une conférence aux États-Unis qui s'est tenue au Dartmouth College. Par la suite, ce domaine atteindra de prestigieuses universités comme celles de Stanford, du MIT, ou encore d'Édimbourg [8].

1.2.2 Objectifs de l'IA

Créer des systèmes experts : les systèmes qui présentent un comportement intelligent, apprennent, démontrent, expliquent et conseillent leurs utilisateurs.

Mettre en œuvre l'intelligence humaine dans les machines : créer des systèmes qui comprennent, pensent, apprennent et se comportent comme des humains.

1.2.3 Fonctionnement de L'IA

Les machines dotées d'une intelligence artificielle peuvent mémoriser le comportement. Ce type de travail de mémoire leur permet de résoudre les problèmes plus tard

et de prendre des mesures correctives face à telle ou telle situation. Cet apprentissage se fait à l'aide de bases de données et d'algorithmes. Cette tâche complexe aide la machine à mesurer l'importance du problème, à examiner les solutions possibles et les situations passées similaires afin de faire la bonne chose.

C'est en réalité un système de statistiques sophistiqué et très performant qui conduisent la machine à prendre une décision ou à avoir le comportement attendu. Pour mesurer son degré d'intelligence, une machine est soumise au test de Turing. Ce test porte le nom de l'inventeur de l'IA, Alan Turing. Ce mathématicien britannique fut l'un des premiers à se demander, en 1950, si une machine était capable de penser. Le test de Turing consiste à converser avec la machine et à lui demander de créer quelque chose avec des critères précis qu'elle se doit de respecter [9].

1.2.4 Qu'est-ce qu'une prise de décision ?

La prise de décision est un processus cognitif complexe, différent de la réaction instinctive et immédiate, visant à la sélection d'un type d'action parmi différentes alternatives. Ce processus est théoriquement basé sur des critères de choix, et sur une analyse des enjeux et des options et conduit à un choix final.

Dans cette définition, nous apprenons donc qu'une prise de décision est un processus qui met en place à la fois l'utilisation des connaissances apprises dans certains environnements pour choisir une action parmi tout un panel, et également l'apprentissage de part ces mêmes environnements qui permettra lors de prochaines prises de décision d'avoir une expérience plus grande afin de choisir l'action qui apportera le meilleur ratio entre le gain et les conséquences [5].

De plus il n'est pas nécessaire d'apporter une réponse de vive voix, mais il est très important que la prise de décision se fasse dans un état conscient. Ce pendant ce n'est pas toujours par la réflexion et le passage vers notre soi conscient que la décision est prise. Certaines décisions sont prises par notre inconscient, les exemples donnés le plus fréquemment sont l'arrêt à un feu rouge lorsque nous sommes au volant d'un véhicule, ou le fait de s'éloigner d'un feu lorsqu'il commence à heurter notre corps. Nous ne réfléchissons pas dans ces cas-là à la décision que nous devons prendre, ainsi nous prenons des décisions sur le coup, pourtant les processus ont bien été mis en place et se sont déroulés de manière inconsciente. On peut donc distinguer deux types

de prises de décision : [5]

- les prises de décision conscientes, où nous réfléchissons et où nous passons consciemment par toutes les étapes de la prise de décision.
- les prises de décision inconscientes, où nous ne nous rendons pas compte de ce processus mais qui se produit tout de même, et qui apportent une réponse.

1.2.5 Que fait une machine pour décider ?

La machine va collecter des données, le plus possible, les classer et les interpréter, les évaluer et décider ou éventuellement aider l'humain à décider, ce que l'on appelle l'aide à la décision.

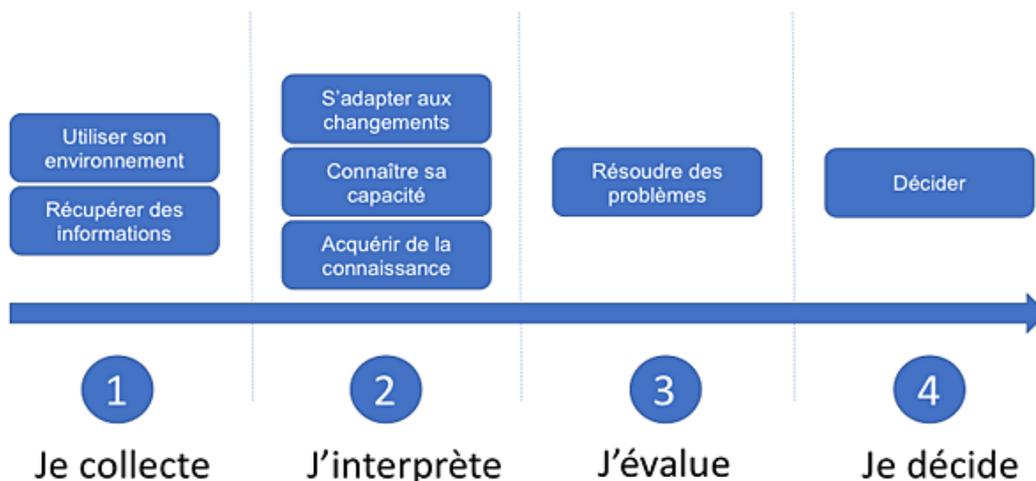


FIGURE 1.1 – Décision de la machine

1.2.6 Programmation sans et avec AI

La segmentation du champ de l'IA, place curieusement au même niveau des outils génériques comme le deep learning et la machine learning et ses propres applications comme la vision artificielle ou le traitement vocale.

La robotique intègre de son côté tous les autres champs du schéma plus quelques autres qui lui sont spécifiques comme les capteurs, les matériaux, la mécanique, les moteurs électriques et autres batteries [10].

La programmation sans et avec AI est différente des manières suivantes : [4]

Tableau 1.1 – Programmation sans et avec IA [4]

Programmation sans IA	Programmation avec IA
Un programme informatique sans IA peut répondre aux questions spécifiques auxquelles il est destiné résoudre.	Un programme informatique avec IA peut répondre aux questions génériques qu'elle est censée résoudre.
La modification du programme conduit à Changement dans sa structure.	Les programmes d'IA peuvent absorber de nouvelles modifications en mettre hautement pièces indépendantes de La modification du programme conduit à changement dans sa structure. Informations ensemble. Par conséquent, vous pouvez modifier même une infime information de programme sans affecter sa structure.
La modification n'est ni simple ni rapide. Cela pourrait conduire à nuire au programme.	Modification rapide et facile du programme.

Les promesses non tenues des débuts de l'IA ont amené à distinguer d'une part les machines qui non seulement mettraient en œuvre des raisonnements semblables aux raisonnements humains, mais auraient également une réelle conscience d'elles-mêmes : c'est ce qu'on appelle l'intelligence artificielle forte.

D'autre part les machines qui rendent de nombreux services aux humains en simulant l'intelligence humaine : c'est l'intelligence artificielle faible (figure 1.2).



FIGURE 1.2 – l'intelligence artificielle faible et forte [7]

1.3 Applications de l'IA

L'IA a été dominante dans divers domaines tels que :

1.3.1 Traitement du langage naturel

Il est possible d'interagir avec l'ordinateur qui comprend le langage naturel parlé par les humains [4].

1.3.2 Systèmes experts

Certaines applications intègrent des machines, des logiciels et des des informations pour donner un raisonnement et des conseils. Ils fournissent des explications et des conseils aux les utilisateurs.

1.3.3 Systèmes de vision

Ces systèmes comprennent, interprètent et comprennent les entrées visuelles sur l'ordinateur. Par Exemple,

- ◇ Un avion d'espionnage prend des photographies qui sont utilisées pour déterminer l'espace informations ou carte des zones.
- ◇ Les médecins utilisent un système expert clinique pour diagnostiquer le patient.
- ◇ La police utilise un logiciel informatique capable de reconnaître le visage d'un criminel avec le portrait stocké réalisé par un artiste médico-légal [4].

1.3.4 Reconnaissance de la parole

Certains systèmes intelligents sont capables d'entendre et de comprendre la langue en termes de phrases et leurs significations pendant qu'un humain lui parle. Il peut gérer différents accents, mots d'argot, bruit en arrière-plan, changement du bruit humain dû au froid etc.

1.3.5 Robots intelligents

Les robots sont capables d'exécuter les tâches données par un humain. Ils ont des capteurs pour détecter des données physiques du monde réel telles que la lumière, la chaleur, la température, le mouvement, le son, bosse et pression. Ils ont des processeurs efficaces, plusieurs capteurs et d'énormes mémoire, pour montrer l'intelligence.

De plus, ils sont capables d'apprendre de leur erreur et ils peuvent s'adapter au nouvel environnement [4].

1.4 Domaines d'utilisation de l'IA

Avant de nous centrer sur l'IA en éducation nous présentons différents domaine sd'utilisation de l'IA.

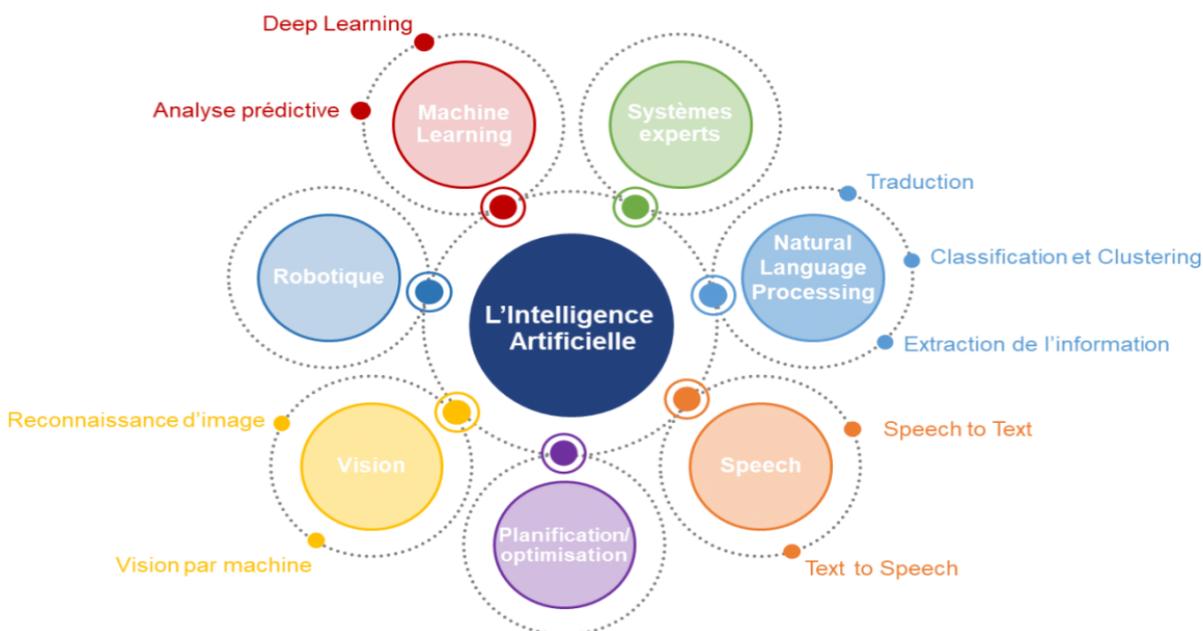


FIGURE 1.3 – Panorama des domaines de l'intelligence artificielle [11]

La figure 1.3 ci-dessus présente sept domaines où intervient l'IA.

L'IA est souvent assimilée à un robot ,ou plutôt un système embarqué dans un corps, cependant l'IA est un ensemble de technique qui intervient dans bien des domaines sous une forme algorithmique,notamment comme dans le domaine Speech, un domaine dans lequel est l'outil Google Traduction ou reverso.

1.4.1 Domaines d'applications de l'IA dans les Smart homes

Les dernières avancées dans les méthodes d'intelligence artificielle ont produit des applications qui utilisent les informations généralement incertaines et imprécises obtenues à partir de l'environnement perceptif, par exemple, la modélisation de situations et l'apprentissage automatique, et le raisonnement temporel du traitement des événements dans l'environnement perceptif, ou l'analyse de comportement des habitants. La représentation des connaissances est une question largement discutée en intelligence artificielle, qui est d'une grande importance pour l'intelligence environnementale.

La figure 1.4 montre les relations existant entre l'intelligence ambiante et l'intelligence artificielle.

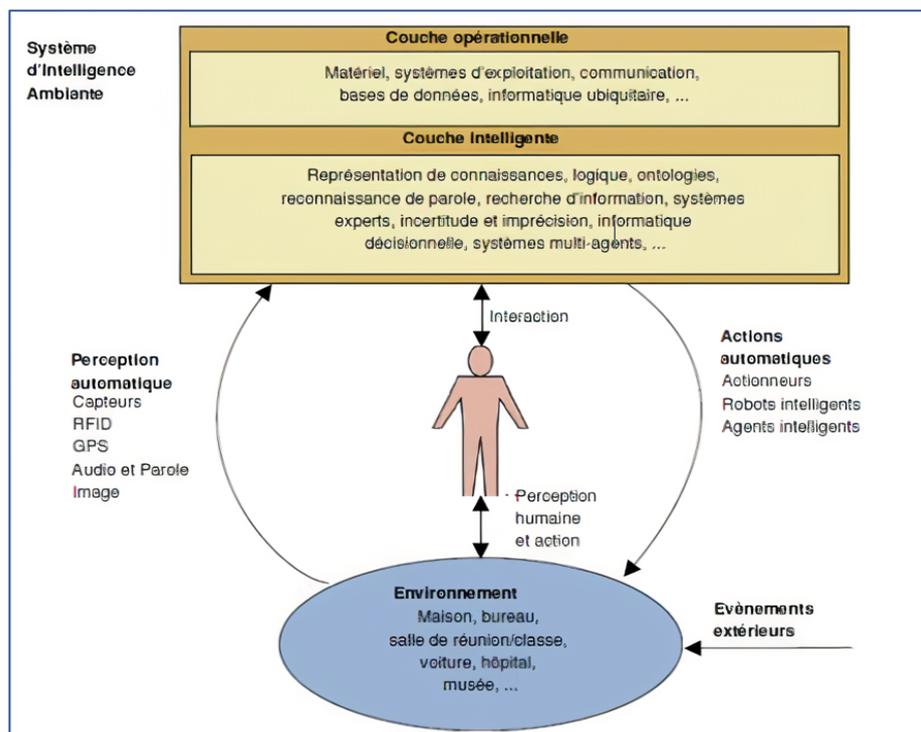


FIGURE 1.4 – Relations entre l'intelligence artificielle et l'intelligence ambiante [12]

La section suivante énumère quelques domaines d'application de l'IA dans les Smart Homes, cependant, cela ne devrait pas être considéré comme une liste exhaustive de possibilités, mais plutôt une opportunité d'offrir certains concepts sur les différents niveaux d'applications pouvant être déployés et qui attendent d'être explorés.

1.5 Réseaux de neurones

Un réseau de neurones (Réseau de Neurones (RN)) est un assemblage de constituants élémentaires inter-connectés (appelés «neurones» en hommage à leur modèle biologique), qui réalisent chacun un traitement simple mais dont l'ensemble en interaction fait émerger des propriétés globales complexes. Chaque neurone fonctionne indépendamment des autres de telle sorte que l'ensemble forme un système massivement parallèle. L'information est stockée de manière distribuée dans le réseau sous forme de coefficients synaptiques ou de fonctions d'activation, il n'y a donc pas de zone de mémoire et de zone de calcul, l'une et l'autre sont intimement liés.

Un réseau de neurone ne se programme pas, il est entraîné grâce à un mécanisme d'apprentissage. Les tâches particulièrement adaptées au traitement par réseau de neurones sont : l'association, la classification, la discrimination, la prévision ou l'estimation, et la commande de processus complexes.

Les réseaux de neurones artificiels consistent en des modèles plus ou moins inspirés du fonctionnement cérébral de l'être humain en se basant principalement sur le concept de neurone. [13]

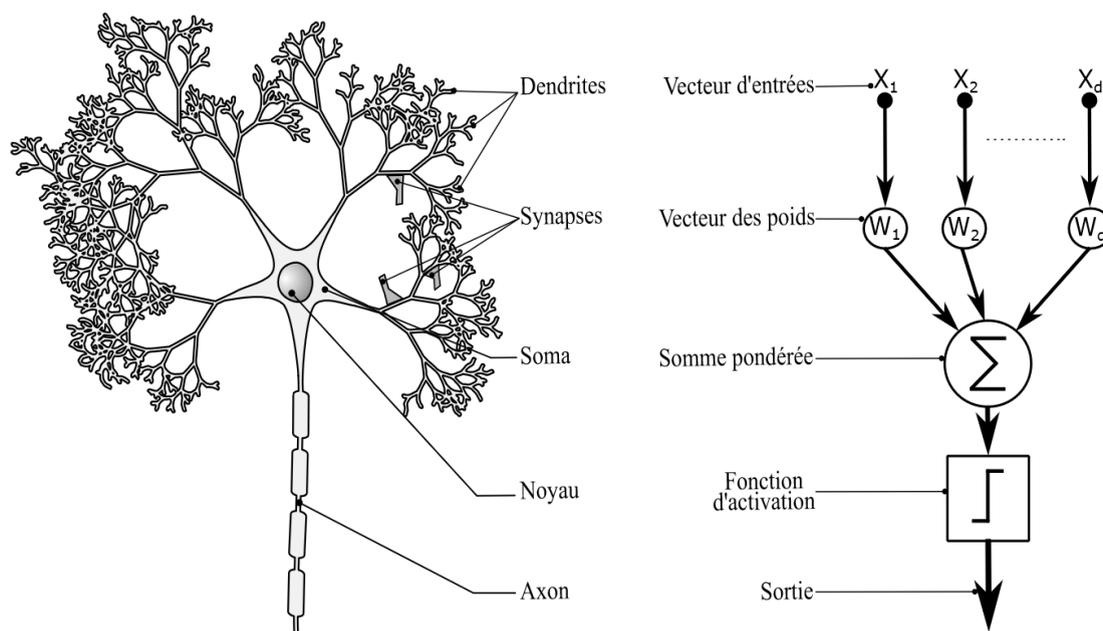


FIGURE 1.5 – Différence entre Neurone Biologique / Neurone Artificiel [14]

Cette comparaison est montrée dans Tableau 1.2 [15]

Tableau 1.2 – Comparaison entre neurone Biologique et artificiel

Système nerveux	Réseau neurone Artificiel
Neurone	Traitant élément, noeud, neurone artificiel, neurone abstrait.
Le corps cellulaire (soma)	Niveau de l'activation, fonction de l'activation, fonction du transfert, la fonction de la sortie
Axone	La communication avec d'autre neurone.
Synapse	poids multiplicatifs.

1.5.1 Les réseaux de neurones formels

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'oeuvre de MM. Mac Culloch et Pitts. Ils présentent un modèle assez simple pour les neurones et explorent les possibilités de ce modèle.

Le neurone représenté par la figure 1.6 se compose d'une cellule possédant plusieurs entrées et une sortie.

Le neurone artificiel moderne fait la somme des entrées multipliées par leur poids, additionne un biais (b qui permet de s'assurer que le résultat reste entre 0 et 1) et lui applique ensuite une **fonction d'activation** qui est une fonction non linéaire comme **une sigmoïde** qui génère une valeur comprise entre 0 et 1, ou entre -1 et +1 générant une valeur statistique facile à exploiter dans le reste du réseau de neurones

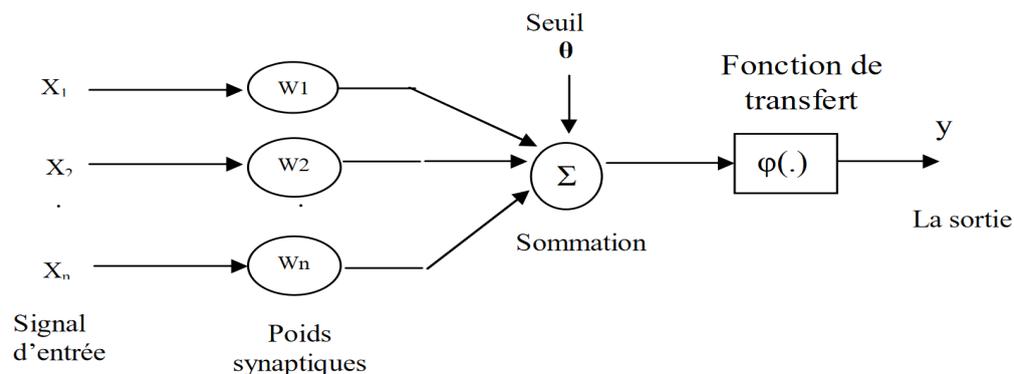


FIGURE 1.6 – Neurone artificiel [16]

Avec :

$$y = f\left(\sum_{i=1}^n W_i x_i + b\right) \quad (1.1)$$

y : La sortie du neurone.

f : La fonction d'activation (ou de transfert).

x : Entrée du neurone.

w : Poids synaptique du neurone.

b : Biais.

Par définition un neurone formel (figure 1.6) est une fonction algébrique non linéaire, paramétrée, à valeurs bornées [17].

Il peut être caractérisé par :

- La nature de ses entrées $(x_1, x_2, \dots, x_i, \dots, x_n)$, qui peuvent être les entrées du réseau ou les sorties d'autres neurones du même réseau.
- La fonction d'entrée totale définissant le pré-traitement (combinaison linéaire) effectué sur les entrées comme $\sum W_i \cdot x_i + b$, dont W_i est le poids synaptique attaché à l'entrée i et le b désigne le seuil d'activation (biais)
- Sa fonction d'activation, ou d'état f , définissant l'état interne du neurone en fonction de son entrée totale. Cette fonction peut prendre plusieurs formes figure
- Sa fonction de sortie calculant la sortie du neurone en fonction de son état d'activation.

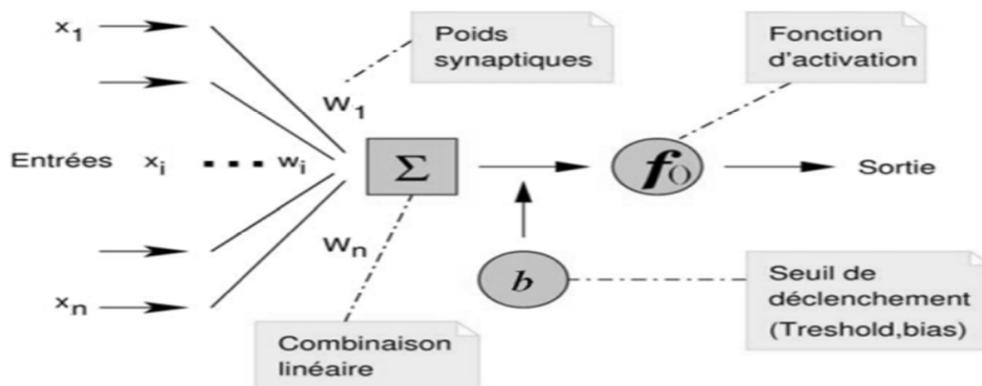


FIGURE 1.7 – Structure générale du neurone formel [17]

1.6 Types des réseaux de neurones artificiels (RNA)

Pour concevoir un réseau de neurones, nous devons établir des connexions entre les neurones.

Tous les Réseaux de Neurones Artificiels (RNA) utilisent la connexion directe pour acheminer l'information de l'entrée vers la sortie. La connexion récurrente permet d'acheminer l'information de la sortie des neurones des couches supérieures vers les entrées des neurones précédents. Les réseaux de neurones qui doivent choisir un neurone gagnant utilisent la connexion latérale pour établir une relation entre les neurones de sortie et la maintenir. Nous avons deux types principaux de connexion : **proactifs**, **récurrente** [18].

1.6.1 Les Réseaux Proactifs (monocouche)

Le Perceptron Multi Couches (Réseaux Proactifs Multi Couches (PMC)), en anglais Multi-Layers Perceptron (Multi-Layers Perceptron (MLP)), est le réseau de neurones le plus utilisé pour des fonctions d'approximation, de diagnostic et de prédiction.

L'architecture des PMC est conçue de manière que le flux de données se propage dans un seul sens (des entrées vers la ou les sorties) en passant par les couches intermédiaires du réseau. Ce modèle utilise souvent une règle d'apprentissage supervisé. L'algorithme d'apprentissage le plus utilisé ou employé par rMLP est la rétro-propagation du gradient où, la propagation des informations effectuée dans le sens inverse pour l'adaptation des paramètres libres [14].

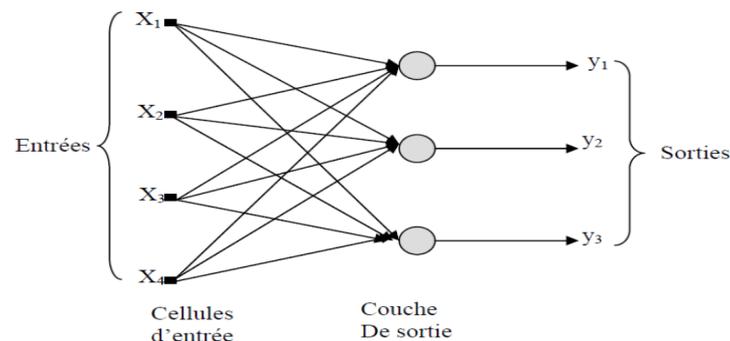


FIGURE 1.8 – Réseau proactif monocouche (Perceptron) [15]

1.6.2 Les Réseaux Proactifs Multi Couches (PMC)

Le PMC a une structure bien particulière : ses neurones sont organisés en couches successives (figure. 1.9). Chaque neurone d'une couche reçoit des signaux de la couche précédente et transmet le résultat à la suivante, si elle existe. Les neurones d'une même couche ne sont pas inter-connectés. Un neurone ne peut donc envoyer son résultat qu'à un neurone situé dans une couche postérieure à la sienne.

L'orientation du réseau est fixée par le sens unique de propagation de l'information, de la couche d'entrée vers la couche de sortie. Pour les réseaux considérés, les notions de couches d'entrée et de sortie sont donc systématiques. Dans ce qui suit un RNA à une couche cachée est utilisé (figure. 1.9).

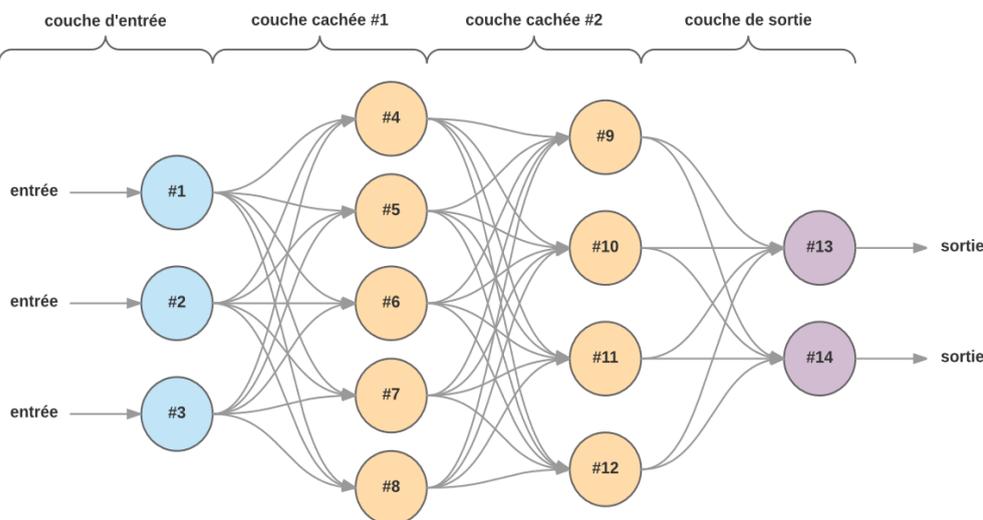


FIGURE 1.9 – Perceptron Multi-couches (MLP) [19]

1.6.3 Les Réseaux récurrents

Les réseaux récurrents se distinguent des réseaux pro-actifs par le fait qu'ils contiennent au moins une boucle de contre-réaction des neurones de sortie vers les entrées ou au moins d'une couche vers une couche précédente, adjacente ou non.

La figure 1.10 Présente un exemple d'un réseau de neurone récurrent qui à deux entrés, trois neurones dans la couche cachée et deux neurones dans la couche de sortie. Les connexions de rétroaction de ce réseau proviennent aussi bien des neurones de la couche cachée que des neurones de la couche de sortie [15].

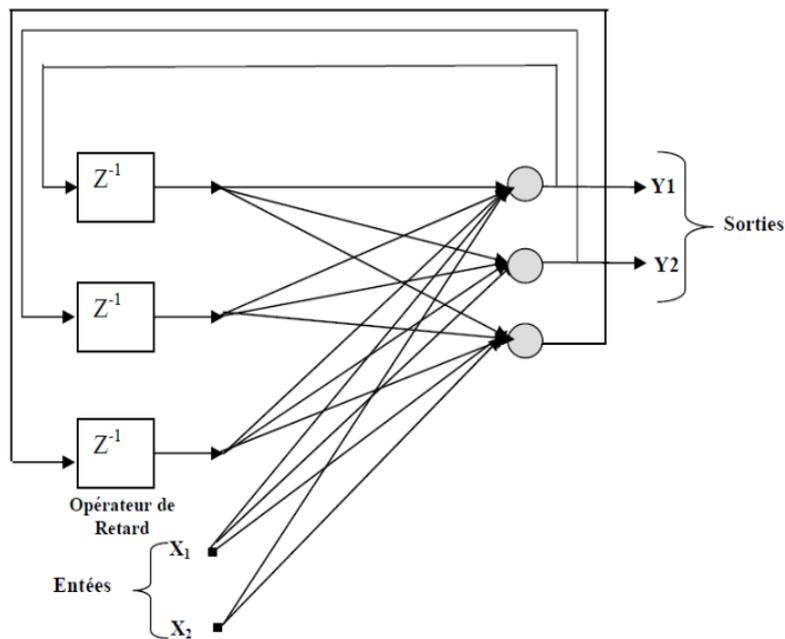


FIGURE 1.10 – Réseau récurrent avec neurones cachés [15]

1.7 Apprentissage

L'apprentissage est la fonction la plus intéressante des réseaux de neurones. C'est le fondement inspirant de cette approche. Un réseau de neurones peut ajuster son fonctionnement dans un environnement défini basé sur un algorithme d'apprentissage.

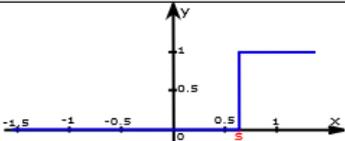
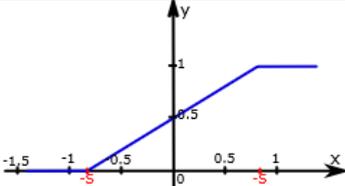
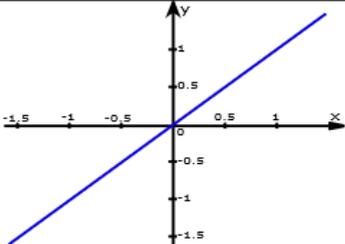
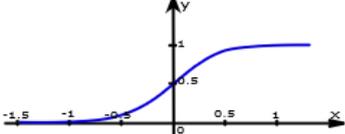
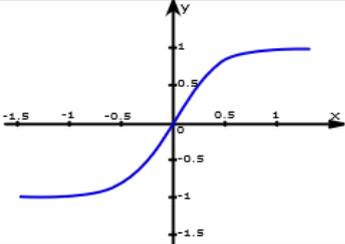
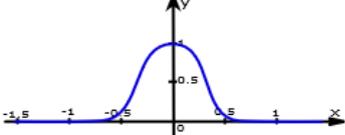
Qu'est-ce qu'apprendre ? Il est difficile de répondre à cette question car il n'existe pas de définition universellement adoptée. De multiples définitions dans la littérature expliquent différents concepts entourant le même concept.

D'une manière générale, les réseaux de neurones doivent être capables d'effectuer des fonctions spécifiques dans un environnement bien défini. Cette capacité est obtenue grâce à l'algorithme d'apprentissage adopté par le modèle neuronal. C'est la première phase du cycle de vie du réseau de neurones (passage de l'état d'ignorance à l'état de connaissance) [14].

En cas de changement de fonction ou d'environnement externe, il faut repasser par cette phase pour adapter le comportement du réseau de neurones aux nouveaux paramètres (transition entre deux états de savoir). L'apprentissage d'un réseau de neurones consiste à faire subir au système un ensemble de stimulations itératives. Par intuition, le réseau sera informé de mieux en mieux sur l'ensemble entrée. A chaque itération

d'apprentissage (alimentation du réseau de neurones par un stimulus), les neurones du réseau adaptent leurs traitements pour répondre aux besoins de la fonction ciblée [14].

Tableau 1.3 – Comparaison entre neurone Biologique et artificiel

<i>Fonction</i>	<i>Comportement</i>	<i>Formule mathématique</i>
<i>Pas unitaire</i>		$f(x) = \begin{cases} 0, & \text{si } x < S \\ 1, & \text{si } x \geq S \end{cases}$
<i>Linéaire Seuillée</i>		$f(x) = \begin{cases} 0, & \text{si } x \leq -S \\ m \cdot x + b, & \text{si } -S < x < S \\ 1, & \text{si } x \geq S \end{cases}$
<i>Identité</i>		$f(x) = x$
<i>Sigmoïde</i>		$f(x) = \frac{1}{e^{-x} + 1}$
<i>Tangente</i>		$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
<i>Gaussienne</i>		$f(x) = \frac{1}{\sqrt{2\pi}\sigma} * e^{-\frac{(x-m)^2}{2\sigma^2}}$

Les algorithmes d'apprentissages donnent de meilleurs résultats lorsqu'on leur fournit des exemples multiples et variés; ainsi le réseau peut assimiler toutes les connaissances. Ils existent différentes règles d'apprentissages parmi lesquelles on peut distinguer [20]

- ✓ La règle de Widrow-Hoff.
- ✓ La règle de Hbb
- ✓ La règle de Perceptron
- ✓ La règle de Grossbergen, ect... [20].

A la base, l'idée d'apprentissage a été inspirée de l'apprentissage par conditionnement observé chez l'animal.

Le physiologiste Pavlova introduit la notion de conditionnement classique expliquée par sa fameuse expérience de conditionnement du chien illustré sur la figure 1.11 [21].

Conditionnement classique : A- stimulus inconditionnel. B- stimulus neutre. C- Association de stimuli. D- Stimulus conditionnel.

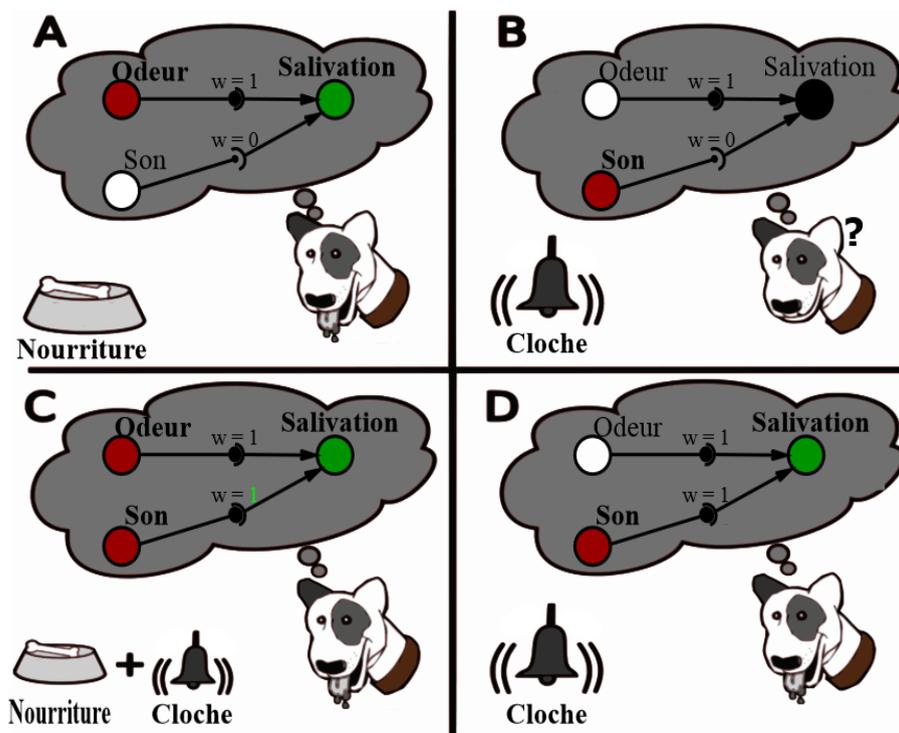


FIGURE 1.11 – Expérience de Pavlov [14]

L'exemple de la figure 1.12 est une partie de l'application de reconnaissance du caractère manuscrit. Il nous donne une idée sur la forme générale d'un apprentissage de réseau de neurones [22].

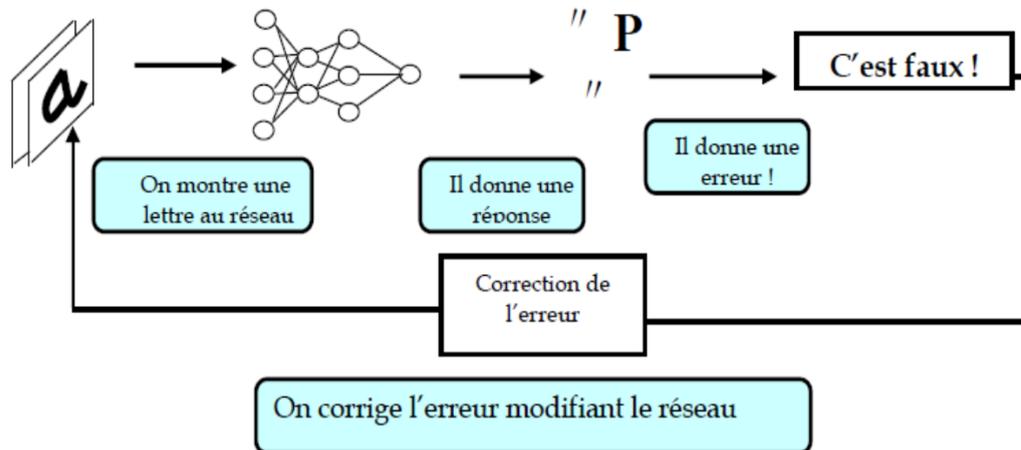


FIGURE 1.12 – Exemple d'apprentissage

Il existe plusieurs types de règles d'apprentissage qui peuvent être rassemblées en deux catégories qui sont l'apprentissage supervisé et l'apprentissage non supervisé.

1.7.1 Apprentissage supervisé

Apprentissage supervisé Dans ce type d'apprentissage, on cherche à imposer au réseau un fonctionnement donné en forçant les sorties des réseaux à prendre des valeurs bien données (choisie par l'opérateur) et ce en modifiant les poids synaptiques. Le réseau se comporte alors comme un filtre dont les paramètres de transfert sont ajustés à partir des couples entrée-sortie présentés.

L'adaptation des paramètres du réseau s'effectue à partir d'un algorithme d'optimisation, l'initiation des poids synaptiques étant le plus souvent aléatoire. Des exemples d'utilisations de différentes règles d'apprentissage, architectures et algorithmes qui sont reliés à ce type supervisé sont illustrés dans le tableau 1.4 [23].

Tableau 1.4 – Exemple de différents types d'apprentissages en mode supervisé [23]

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Supervisé	Perception Simple ou Multi-Couches	Correction d'erreur	Perceptron, Rétro- Propagation, Adaline, Madaline	Classification, Approximation de fonctions, Prédiction, Contrôle
	Récurrente	Boltzmann	Apprentissage de Boltzmann	Classification
	Multi-Couches non bouclés	Hebb	Analyse de discriminants linéaires	Analyse de données, Classification
	à compétition	Par compétition	LVQ	Catégorisation au sein d'une classe, Compression de données
	ART		ARTMap	Classification, Catégorisation au sein d'une classe

1.7.2 Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, seules les valeurs d'entrée sont disponibles dans ce type d'apprentissage non supervisé qui est appelé aussi « apprentissage par compétition ». Dans ce cas, les exemples présentés à l'entrée provoquent une auto-adaptation du réseau afin d'engendrer des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires. Des exemples d'utilisations de différentes règles d'apprentissage, architectures et algorithmes qui sont reliés à ce type supervisé sont illustrés dans le tableau 1.5 [23].

Tableau 1.5 – Exemple de différents types d'apprentissages en mode non supervisé [23]

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Non supervisé	Multi-couches non bouclé	Correction d'erreur	Projection de Sammon	Analyse de données
	Non bouclé ou à compétition	Hebb	Analyse en composantes	Analyse de données,
	à compétition	Par compétition	VQ	Catégorisation, Compression de données
	Carte de Kohonen		SOM	Catégorisation, Analyse de données
	ART		ART-1, ART-2	Catégorisation

1.8 Topologies des RNA

On classe généralement les réseaux de neurones en deux catégories : les réseaux faiblement connectés à couches que l'on appelle des réseaux "feedforward" ou réseaux directs et les réseaux fortement connectés que l'on appelle des réseaux "récurrents". Dans ces deux configurations, on retrouve des connexions totales ou partielles entre les couches. La figure Suivante présente les différentes topologies d'un Réseau de neurones artificiels [24].

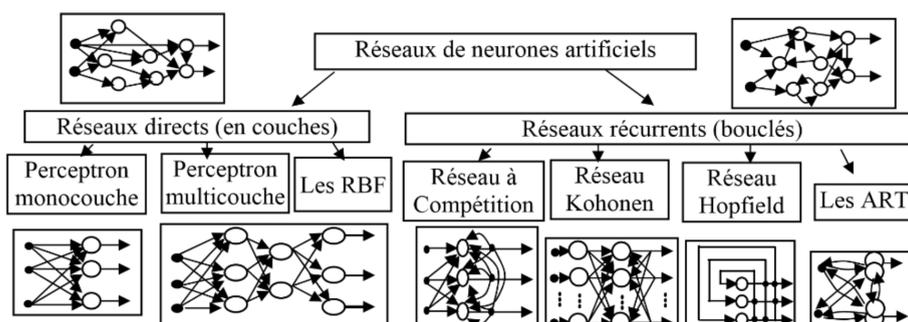


FIGURE 1.13 – Topologies des réseaux de neurones artificiels [24]

1.8.1 Tache des neurones Artificiels

Un réseau de neurone est conçu pour remplir une tâche que le concepteur définit, donc il est nécessaire de communiquer au réseau des données, et de récupérer des résultats de calcul. Afin de réaliser cette tâche, les neurones constituant un réseau sont répartis en trois sous-ensembles (couches) comme suit :

- Couche ou ensemble de neurones d'entrée : ils perçoivent les données du problème sous forme d'un vecteur codé nommé patron d'entrée.
- La recherche mathématique.
- Couche cachée ou ensemble de neurones cachés : cette couche est d'un intérêt particulier pour la couche ou ensemble de neurones de sortie : l'activation de ces neurones sera interprétée comme la réponse du réseau. Elle est sous forme d'un vecteur codé nommé patron de sortie qui va être décodé pour interpréter les résultats..

1.9 Avantages et Inconvénients des réseaux de neurones

1.9.1 Avantage des réseaux de neurones

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe
- Faculté d'apprentissage à partir d'exemples représentatifs, par "rétro-propagation des erreurs". L'apprentissage (ou construction du modèle) est automatique
- Résistance au bruit ou au manque de fiabilité des données
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en mathématiques, informatique statistique requise
- Comportement moins mauvais en cas de faible quantité de données
- Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multi-variables. [25]

1.9.2 Inconvénients des réseaux de neurones

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s)
- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence
- Le problème du sur-apprentissage (apprentissage au détriment de la généralisation)
- La connaissance acquise par un réseau de neurone est codée par les valeurs des poids synaptiques, les réseaux de neurones sont donc des boîtes noires où les connaissances sont intelligibles pour l'utilisateur. [25].

1.10 Réseau de neurones convolutifs

Les réseaux de neurones à convolution Convolutional Neural Networks (**CNN**) sont une catégorie de réseaux de neurones qui se sont avérés très efficaces dans des domaines tels que la reconnaissance vocale et la classification d'images.

Les réseaux convolutifs ont été introduits pour la première fois par Fukushima, il a dérivé une architecture du réseau nerveux hiérarchique inspirée par le travail de recherche de Hubel. Lecun, les a généralisés pour classer les chiffres avec succès et pour reconnaître les numéros de contrôle manuscrit par LeNet-5.

Ciresan a utilisé les réseaux convolutifs et réalisé les meilleures performances dans la littérature pour la reconnaissance d'objets multiples pour des bases de données d'images : MNIST, NORB, CIFAR10 et l'ensemble de données Image Net.

Les réseaux CNN se concentrent principalement sur le fait que l'entrée sera composée d'images. Cela permet de centrer l'architecture à mettre en place pour répondre au mieux à la nécessité de traiter un type de données spécifiques.

Les réseaux neuronaux convolutifs diffèrent des autres formes de réseaux neuronaux artificiels en ce sens. Qu'au lieu de se concentrer sur l'intégralité du domaine problématique, les connaissances sur le type spécifique d'entrées sont exploitées, cela permet à son tour de mettre en place une architecture réseau beaucoup plus performante [26].

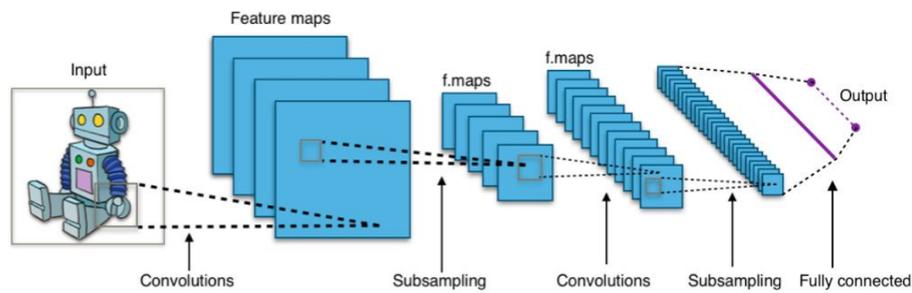


FIGURE 1.14 – Réseau neuronal convolutif [26]

1.10.1 Architecture globale de CNN

Les CNN sont composés de trois types de couches : des couches convolutives, des couches de regroupement et des couches entièrement connectées. Lorsque ces couches sont empilées, une architecture CNN a été formée. Une architecture CNN simplifiée pour la classification MNIST est illustrée à la figure 1.15 [26].

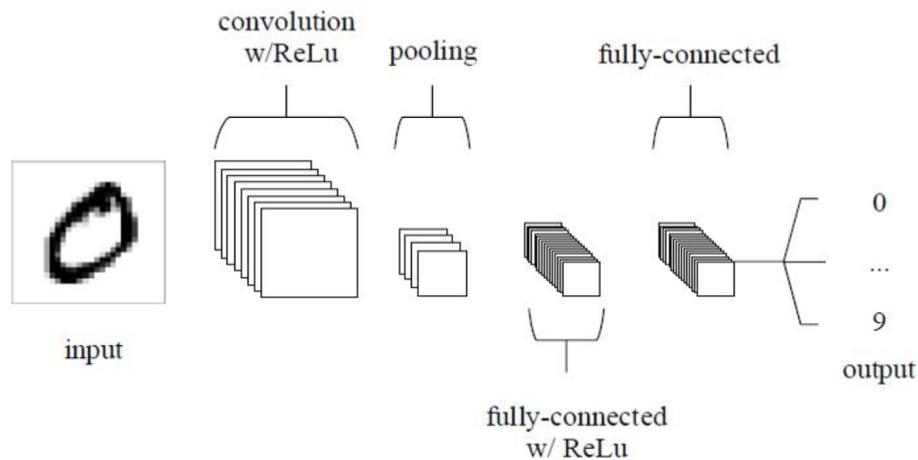


FIGURE 1.15 – Architecture simple de CNN [26]

1.10.2 Couche convolutive

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution

(aussi appelé volume de sortie) : la 'profondeur', le 'pas' et la 'marge'.

1. **'Profondeur'** de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).

2. **'Le pas'** contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand
3. **'La marge (à 0)'** ou 'zéro padding' : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zéro padding' marge permet de contrôler la dimension spatiale du volu souhaitable de conserver la même surface que celle du volume d'entrée. frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyperparamètre.

Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du hyperparamètre. Cette sortie. En particulier, il est parfois [27].

1.10.3 Couche de pooling

Après chaque couche convolutive, il peut y avoir une couche de pooling. Cette couche sous échantillonne le long de la dimensionnalité spatiale de l'entrée donnée, ce qui réduira davantage le nombre de paramètres au sein de cette activation. Il y a plusieurs façons de faire cette mise en commun, comme prendre la moyenne ou le maximum, ou une combinaison linéaire prise par des neurones dans le bloc. Par exemple, la Fig. 3 montre le max pooling sur une fenêtre 2×2 . [26].



FIGURE 1.16 – Exemple de max-pooling [26]

1.10.4 Couche de correction (ReLU)

C'est une couche pour améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie [26].

La fonction ReLu :

$$F(x) = \max(0, x) \quad (1.2)$$

- Cette fonction force les neurones pour que toutes les valeurs négatives soient mises à zéro [28].

1.10.5 Avantages et inconvénients du CNN

Tableau 1.6 – Avantages et inconvénients du CNN

Avantages	Inconvénient
Peut être implémenté dans diverses résolutions d'image.	Ne convient pas pour être simple.
L'informatique est tellement détaillée que le taux d'erreur est probablement faible.	Traiter assez longtemps.
Le réseau de neurones de convolution (CNN) est capable de résoudre des problèmes complexes ayant de nombreux paramètres à calculer.	L'informatique est très complexe, directement proportionnelle à la complexité des problèmes
Peut classer la forme du visage de données connues et inconnues.	Je ne peux pas décrire le visage avec une certaine position.

1.11 Réseaux de neurones récurrents (RNN)

Les Réseaux de neurones récurrents (RNN) sont des modèles capables de prendre en compte un contexte dans leur fonction de décision. Ils sont pour cela particulièrement adaptés à plusieurs tâches de Traitement Automatique des Langues (TAL), notamment celles qui consistent à prédire une information séquentielle, La plupart des modèles probabilistes utilisés pour ce type de tâches (même les tout premiers, historiquement) sont aussi capables d'utiliser un contexte dans leur fonction de décision.

Dans le cas des RNN, l'information contextuelle est donnée par une connexion en boucle. Cette connexion permet de prendre en compte à l'étape courante une ou plusieurs informations prédites dans une étape précédente [29].

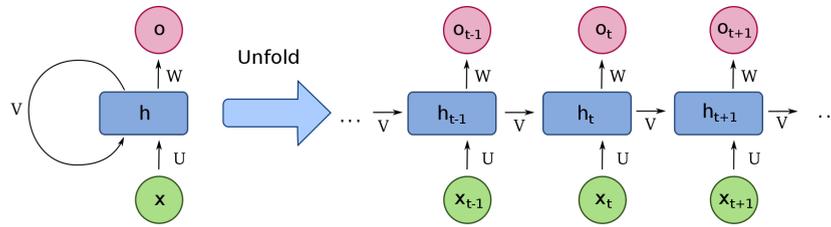


FIGURE 1.17 – Schéma d’un réseau de neurones récurrents à droite la version « dépliée » de la structure.

1.11.1 Fonctionnement général

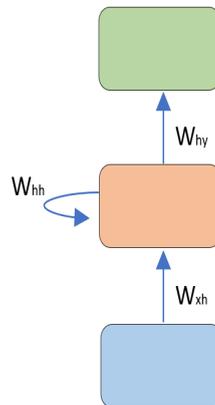


FIGURE 1.18 – Représentation compacte des RNN [30]

Les réseaux de neurones récurrents (illustrés à la figure 1.18) impliquent des boucles dans les graphes neuronaux. La principale motivation derrière cette architecture est de pouvoir manipuler une séquence de vecteurs d’entrée, chaque vecteur représentant un événement temporel, et pas seulement des données isolées sans signification temporelle. En élargissant la modélisation compacte de RNN en termes de temps (voir la figure 1.19, ce type de réseau peut être considéré comme une série chronologique de réseaux MLP connectés entre eux via leurs couches cachées respectives. La dépendance potentielle entre les événements de codage RNN est la séquence vectorielle d’entrée [30].

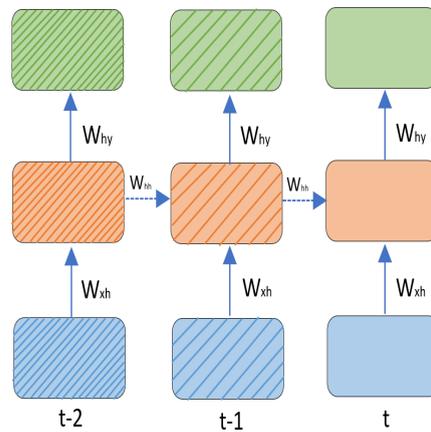


FIGURE 1.19 – Représentation dépliée des RNN [30]

Suivant cette modélisation, un RNN prend en entrée une séquence d'événements $x = (x_1, x_2, \dots, x_T)$ et définit la séquence d'états cachés $h = (h_1, h_2, \dots, h_T)$ pour produire la séquence de vecteurs de sortie $y = (y_1, y_2, \dots, y_T)$ en itérant de $t = 1$ à T :

$$h_t = H(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h) \quad (1.3)$$

$$y_t = W_{hy} \cdot h_t + b_y \quad (1.4)$$

Où T est le nombre total de vecteurs d'entrée, $W_{\alpha\beta}$ est la matrice de poids entre les couches α et β et b_β est le vecteur de biais de la couche β . La fonction H utilisée dans le cas des RNN est généralement la tangente hyperbolique (\tanh).

Étant conçu pour les réseaux non bouclés, l'algorithme de rétro-propagation du gradient n'est pas suffisant pour la prise en compte des liaisons temporelles exprimées à travers les formules (1.2) et (1.3). Une solution à ce problème consiste à considérer une représentation dépliée « hiérarchisée » des RNN. Dans la schématisation offerte dans la figure 1.20, l'échelle temporelle, représentée à travers les arcs diagonaux, porte maintenant une signification hiérarchique dans le sens où les couches cibles sont de plus haut niveau. C'est à travers cette représentation hiérarchisée que nous pouvons utiliser l'algorithme de rétropropagation du gradient généralisé aux réseaux de neurones non bouclés. Cette version est appelée rétropropagation du gradient à travers le temps [30].

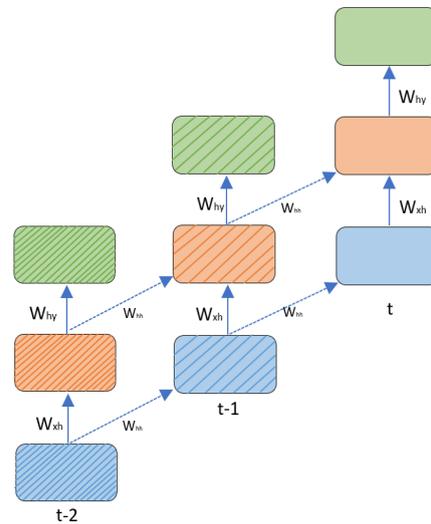


FIGURE 1.20 – Représentation dépliée hiérarchisée des RNN [30]

La particularité de cette représentation par rapport à un réseau de neurones non bouclé « classique » est l'existence de plusieurs paramètres partagés. Par exemple, une matrice de poids commune W_{hh} transmet l'information à travers les arcs diagonaux. En outre, les matrices de poids W_{xh} et W_{hy} , représentées par les arcs verticaux, sont partagées respectivement à travers le temps [30].

1.11.2 Variantes d'apprentissage

Un choix très important pour l'apprentissage des RNN concerne l'algorithme de propagation arrière. En effet, étant donnée leur architecture récurrente, pour apprendre proprement un RNN il faudrait utiliser l'algorithme dit de propagation arrière à travers le temps, censé permettre d'apprendre des contextes arbitrairement longs.

Cependant, on a montré que les RNN pour les modèles des langues apprennent mieux avec seulement 5 étapes passées. Ceci pourrait être dû au fait que, dans les tâches de TAL, l'information passée retenue en mémoire par un RNN grâce à sa connexion récurrente se disperse après quelques étapes seulement.

Aussi, du moins dans le cas des RNN de Jordan, garder en mémoire un contexte d'étiquettes prédites arbitrairement long ne donne pas en général une garantie d'amélioration des performances, un contexte plus long étant aussi plus bruité. Puisque l'algorithme de propagation arrière à travers le temps est plus coûteux que l'algorithme classique, on a préféré utiliser explicitement l'information contextuelle donnée à

la connexion récurrente dans les RNN, et utiliser l'algorithme de propagation arrière classique, sans perte de performance [29].

1.12 Domotique

Les développements technologiques des dernières décennies ont introduit progressivement l'électronique dans notre quotidien dans des proportions devenues importantes. Cette incursion permet aujourd'hui d'imaginer, de manière sociologiquement naturelle, l'immersion complète des personnes dans des environnements comme des cloisons ou du mobilier comportant des systèmes électroniques cachés avec lesquels il est possible d'interagir. L'immersion dans un environnement interactif peut avoir plusieurs intérêts pour la personne tel que :

- ◇ une interaction plus naturelle avec l'environnement.
- ◇ une facilité d'utilisation de certains systèmes électroniques ou informatiques élargissant l'accès à ces technologies.
- ◇ une meilleure expérience lors de l'utilisation de systèmes multimédia.
- ◇ une amélioration de la qualité de vie pour chacun.
- ◇ l'aide et le maintien à domicile des personnes dépendantes (personnes âgées, malades chroniques, personnes handicapées).
- ◇ le maintien d'une vie sociale pour les personnes dépendantes au travers des liens électroniques avec l'extérieur (famille, communautés).
- ◇ le suivi médical permanent à domicile des personnes.

1.13 Traitement Vocal

La parole étant l'un des moyens de communication les plus importants entre les humains, le traitement de la parole est devenu un sujet intéressant pour les chercheurs et les ingénieurs. La reconnaissance vocale est une technologie utilisée pour convertir le signal vocal en mots compréhensibles ou en séquences de mots compréhensibles. ASR est le domaine d'étude des signaux de parole et des moyens de traiter ces signaux. Pour obtenir ce signal, les humains utilisent leurs cordes vocales pour produire

une quantité de son ou de parole, puis ils enregistrent ce son à l'aide d'un microphone de haute qualité. Après cela, la parole passe par un système de reconnaissance vocale pour reconnaître et convertir ce signal en une série de mots (texte). Le schéma fonctionnel de la figure 1.21 illustre le mécanisme général de traitement de la parole [31].

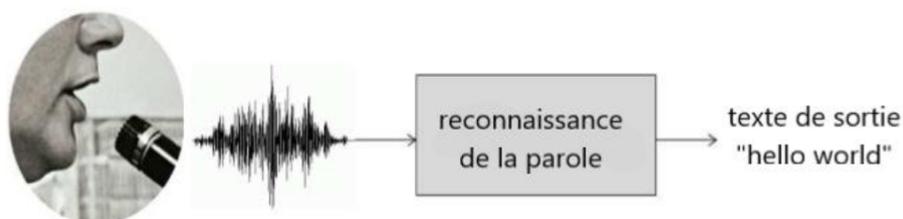


FIGURE 1.21 – Mécanisme du système de reconnaissance vocale [31]

La reconnaissance vocale est le processus de conversion de la parole en texte. Il est considéré comme un sous-domaine de la linguistique informatique, qui permet aux ordinateurs de comprendre le discours du langage naturel. C'est la connaissance et la recherche collectives du domaine linguistique, informatique et électrique. Les entreprises géantes de la technologie comme Google, Microsoft, Apple, IBM, Amazon, Baidu, etc. travaillent dans ce domaine depuis plusieurs décennies. Il existe essentiellement deux types de systèmes de reconnaissance vocale qui sont en pratique de nos jours :

- Système de reconnaissance vocale dépendant du locuteur
- Systèmes de reconnaissance vocale indépendants du locuteur [32].

1.13.1 Système de reconnaissance vocale dépendant du locuteur

Les systèmes dépendant du locuteur sont les systèmes qui ont été utilisés à des fins de dictée, dans lesquels un locuteur individuel lit et prononce le contenu donné à des fins d'apprentissage du système. De tels types de systèmes sont très précis pour un seul locuteur.

Ces systèmes sont plus fiables que les locuteurs indépendants, mais la limitation de tels systèmes est qu'ils ne sont capables de traduire que le discours d'une seule personne. Les systèmes dépendants du locuteur sont appris à traduire la parole d'un

seul locuteur uniquement, sa manière de parler, son débit et d'autres facteurs de la voix sont pris en compte par les systèmes de reconnaissance vocale dépendant du locuteur. Ces types de systèmes sont les meilleurs pour le système de reconnaissance vocale à utilisateur unique uniquement. Cela donnera une précision extrême pour traduire le discours d'un seul locuteur en texte. Pour utiliser ce type de système, chaque utilisateur doit former son propre système uniquement pour son utilisation[32].

CHAPITRE 2

MICROCONTRÔLEUR

Sommaire

2.1	Introduction	38
2.2	Description d'un Microcontrôleur	38
2.3	Avantages des microcontrôleurs	40
2.4	Le choix d'un microcontrôleur	41
2.5	Présentation Du Raspberry Pi 3	41
2.6	Présentation d'Arduino	48
2.7	Composants utilisés	51

2.1 Introduction

Les développements technologiques des dernières décennies ont introduit progressivement l'électronique dans notre quotidien dans des proportions devenues importantes. Cette incursion permet aujourd'hui d'imaginer, de manière socialement naturelle, l'immersion complète des personnes dans des environnements comme des cloisons ou du mobilier comportant des systèmes électroniques cachés avec lesquels il est possible d'interagir pour réaliser cela nous devons utiliser un microcontrôleur, alors l'objectif final étant de faciliter la vie des usagers.

2.2 Description d'un Microcontrôleur

Un microcontrôleur est un composant réunissant sur un seul et même silicium un micro-processeur, divers dispositifs d'entrées/sorties et de contrôle d'interruptions ainsi que de la mémoire, notamment pour stocker le programme d'application. Dédié au contrôle, il embarque également un certain nombre de périphériques spécifiques des domaines ciblés (bus série, interface parallèle, convertisseur analogique numérique, ...). Les microcontrôleurs améliorent l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré. On parle alors de "système sur une puce" (en anglais : "System On chip") [33].

Il existe plusieurs familles de microcontrôleurs, se différenciant par la vitesse de leur processeur et par le nombre de périphériques qui les composent. Toutes ces familles ont un point commun c'est de réunir tous les éléments essentiels d'une structure à base de microprocesseur sur une même puce. Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

- ✓ Un Microprocesseur (C.P.U.)
- ✓ Des Bus
- ✓ De la Mémoire de donnée (RAM et EEPROM)
- ✓ De la Mémoire programme (ROM, OTPROM, UVPROM ou EEPROM).
- ✓ Des Interfaces parallèles pour la connexion des entrées / sorties.

- ✓ Des Interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités.
- ✓ Des Timers pour générer ou mesurer des signaux avec une grande précision temporelle.

Sur la figure 2.1 nous présentons le schéma type d'un microcontrôleur [33].

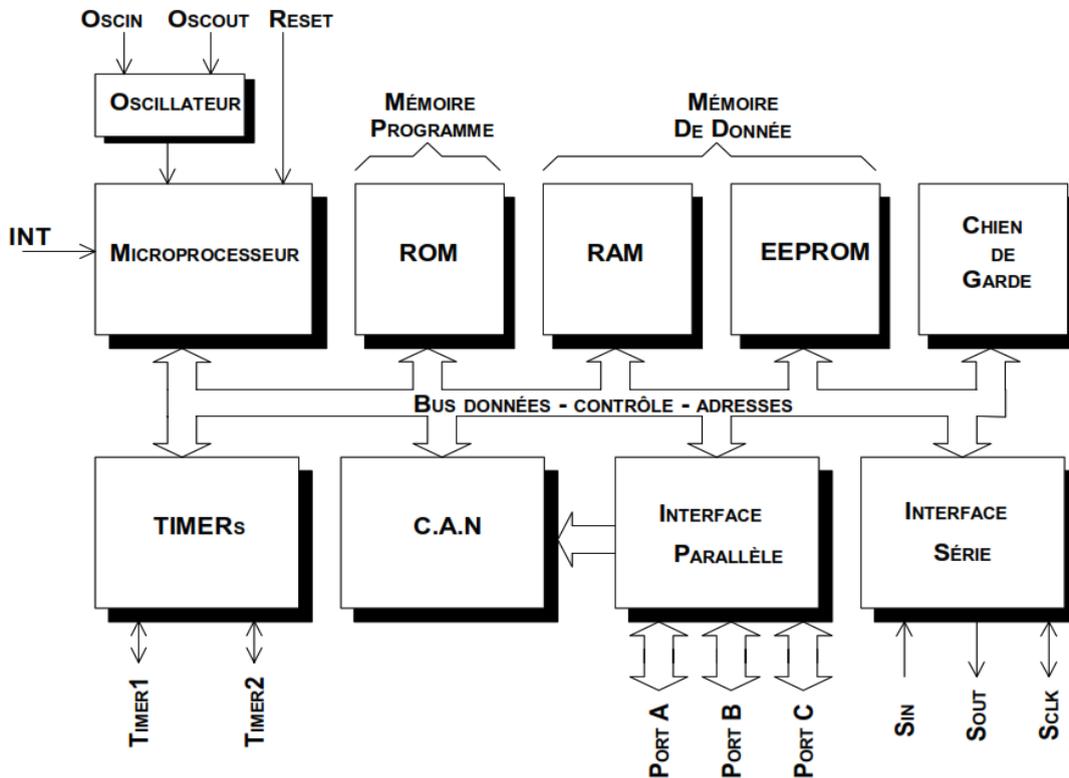


FIGURE 2.1 – Schéma type de tout appareil programmable

- Le processeur est le cœur du μC , c'est cette unité qui exécute le programme et pilote ainsi tous les autres éléments, elle dispose généralement de deux connexions avec l'extérieur, une pour son horloge et une pour sa réinitialisation ou reset
- Les mémoires sont chargées de stocker le programme qui sera exécuté ainsi que les données nécessaires et les résultats obtenus.
- Les périphériques ont pour tâche de connecter le processeur avec le monde extérieur dans les deux sens, soit le processeur fournit des informations vers l'extérieur, soit il en reçoit [34].

2.2.1 Utilisation

Toutes les solutions à base de composants programmables ont pour but de réduire le nombre de composants sur le circuit électronique et donc fiabiliser le circuit.

- Le microcontrôleur est en concurrence avec 3 types de technologies, logique câblée, réseaux de logique programmables (PAL, LCA,..), les microprocesseurs [35].

2.2.2 Types de Microcontrôleur

On trouve donc à l'heure actuelle au moins quatre types différents de microcontrôleur :

- Ceux sans aucune mémoire morte
- Ceux avec EPROM (programmable et effaçable à l'ultra violet)
- Ceux avec EEPROM
- Ceux avec un mélange de ces combinaisons (ROM -EEPROM ou UVPRM - EEPROM)

2.3 Avantages des microcontrôleurs

- Diminution de l'encombrement du matériel et du circuit imprimé.
- Simplification du tracé du circuit imprimé (plus besoin de tracer de bus !).
- Augmentation de la fiabilité du système :
 - ◊ Nombre de composants
 - ◊ Connexions composants/supports et composant circuit imprimé .
- Intégration en technologie MOS, CMOS, ou HCMOS diminution de la consommation.
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux, moins cher que les composants qu'il remplace.
- Environnement de programmation et de simulation évolués [35].

2.4 Le choix d'un microcontrôleur

Choisir un microcontrôleur dépend de la disponibilité c'est-à-dire les circuits de la famille doivent être facilement disponibles sur le marché amateur ,et aussi prix des circuits doit être à la portée de tous ,La programmation de la mémoire morte interne (celle qui contient le programme) doit être facile,Et enfin, les outils développement doivent être aussi peu coûteux que possible.

A l'heure actuelle, les circuits qui répondent le mieux à ces critères sont les microcontrôleur de la famille [Arduino](#) et [PIC de Microchip](#) et [Raspberry Pi](#).

Pour cela nous avons choisi l'Arduino parce que il est moins coûteux fiable et facile a programmé et avec un temps d'exécution rapide .

2.5 Présentation Du Raspberry Pi 3

2.5.1 Définition

Raspberry Pi est une petite carte informatique puissante, bon marché, piratable et orientée vers l'éducation lancée en 2012 figure 2.2. Il fonctionne de la même manière qu'un PC standard, nécessitant un clavier pour l'entrée de commande, une unité d'affichage et une alimentation.

Cet ordinateur de la taille d'une carte de crédit est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux-Debian et des logiciels compatibles. Mais également avec les OS Microsoft Windows : Windows 10 et Android.

La grande majorité des composants du système ses unités de traitement central et graphique, ses parties audio et de communication ainsi que sa puce mémoire de 256 Mo (modèle A) –512 Mo (modèle B), sont construits sur un seul composant.

La carte Raspberry Pi illustrée sur la figure 2.2 contient l'essentiel (processeur, puce graphique, mémoire de programme -RAM) et d'autres périphériques optionnels (diverses interfaces et connecteurs pour périphériques). Le processeur du Raspberry Pi est un système 32 bits, 700 MHz sur une puce, qui est construit sur l'architecture

ARM11 et peut être overclocké pour plus de puissance. La mémoire flash SD sert de disque dur au processeur de Raspberry Pi. L'unité est alimentée via le micro Connecteur USB tandis que la connectivité Internet peut se faire via un câble Ethernet / LAN ou via un dongle USB (connectivité WiFi) [36].

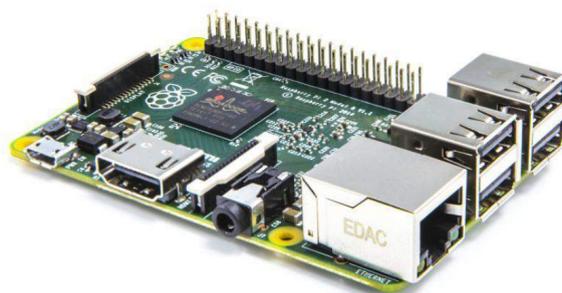


FIGURE 2.2 – Raspberry pi 3

2.5.2 Choix du Raspberry Pi

Tout d'abord, d'un point de vue économique, il faut savoir que les services d'hébergement web ne sont pas gratuits et qu'il faut sortir le porte-monnaie tous les mois/ans. Contrairement à la Raspberry qui demande juste une connexion.

De plus, en choisissant la Raspberry, vous avez la possibilité de modifier vos services à souhait (exemples : la taille du disque, l'hébergement de Database, etc.), ce qui n'est en général pas le cas chez les hébergeurs spécialisés [37].

2.5.3 Les différentes parties d'un RaspberryPi 3

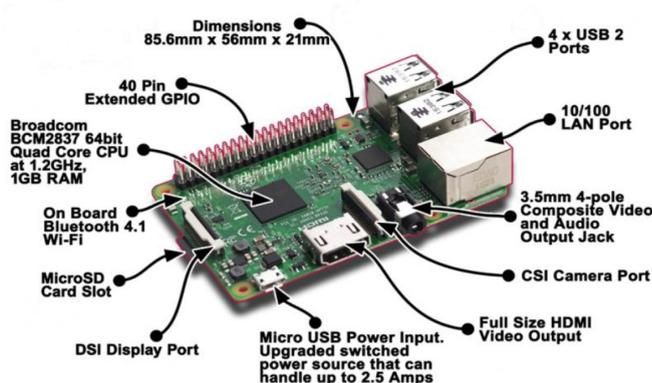


FIGURE 2.3 – Les différentes parties de Raspberry Pi 3 [31]

2.5.3.1 Fiche technique

Tableau 2.1 – Fiche Technique Raspberry Pi

Caractéristiques	
Développeur	Raspberry Pi Foundation
Fabricant	Newark Corporation (en), RS Components (en), Farnell element14 (en)
Type	Ordinateur à carte unique
Date de sortie	29 février 2012
Environnement	Linux (Debian, Fedora et ArchLinux), RISC OS, Windows IOT
Système d'exploitation	Linux (Raspbian, Pidora, et Arch Linux ARM gentoo), RISC OS, FreeBSD, NetBSD, Windows 10 IoT (uniquement compatible avec le Raspberry Pi 2B, 3B et 3B+), expérimental Windows 10 (arm édition) Plan 9
Alimentation	Micro-USB 5 V
Processeur	Broadcom BCM2835 - ARM1176JZF-S 700 MHz (modèle 1) ou 1 GHz (Modèle Zero) ¹ Broadcom BCM2836 - Cortex-A7 900 MHz (modèle 2) Broadcom BCM2837 - ARM Cortex-A53 1.2GHz (modèle 3B) Broadcom BCM2837B0 - ARM Cortex-A53 1.4GHz (modèle 3B+)
Stockage	Carte SD (A, B), Carte microSD (A+, B+, 2B, 3B et 3B+)
Mémoire	256 Mo (modèle A et A+) 256 Mo (modèle B rev 1) 512 Mo (modèle B rev 2 et B+) 1 Go (modèle 2, 3B et 3B+)
Carte graphique	Broadcom VideoCore IV ¹ @250MHz (modèles 1, 2 et Zero); @400MHz (modèle 3B et 3B+)
Connectivité	USB, Ethernet (modèle B, B+, 2B, 3B, 3B+) (RJ45), HDMI, RCA, Jack 3,5 mm, Micro USB Bluetooth (3B = 4.1, 3B+ = 4.2), Wifi (modèle 3B, 3B+ et Zero W), PoE (modèle 3B+)
Dimensions	85,60 mm × 53,98 mm × 17 mm (A, B, B+, 2B, 3B et 3B+), 65 mm × 53,98 mm × 17 mm (A+), 65 mm × 30 mm × 5 mm (PCB v1.2, PCB ,1.3 et W)
Masse	44,885 g (A, B, B+, 2B, 3B), 23 g (A+)
Site web	www.raspberrypi.org

2.5.3.2 Les spécifications Entrées / Sorties

- **textProcesseur ARM** : Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ».
- **Mémoire vive RAM** : C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement.
- **4 ports USB 2.0** : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques
- **1 port Ethernet** : C'est un port qui correspond au protocole international ETHERNET de réseau local à commutation de paquets.
- **1 port HDMI / 1 Composite** : « High Definition Multimedia Interface » permet de relier le Raspberry PI à un dispositif compatible : écran LCD ou un vidéoprojecteur...
- **1 sortie audio jack 3,5mm** : C'est une connectique audio-vidéo.
- **Connecteur SD / SDHC** : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe.
- **GPIO** : « General Purpose Input/Output » sont des ports d'Entrée/Sortie 40 broches.
- **Bluetooth/Wifi 802.11 b/g/n** [\[37\]](#).

2.5.3.3 Les ports GPIO :

Les ports GPIO (General Purpose Input/Output) sont des ports d'entrée/sortie très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée.

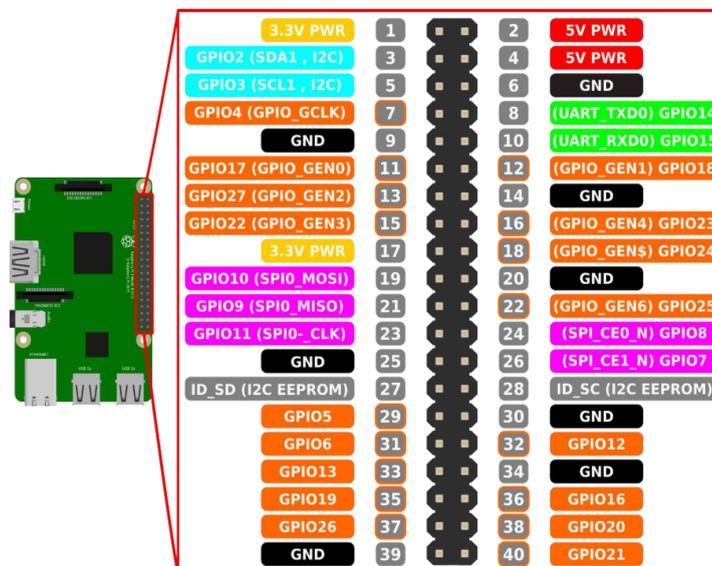


FIGURE 2.4 – Les ports GPIO de Raspberry Pi 3

2.5.4 L'intérêt du Raspberry PI 3 :

- Elle permet d'utiliser les fonctions classiques d'un ordinateur comme le traitement de texte, mais aussi de lire des vidéos et de jouer... et plus encore.
- Outre l'aspect économique, c'est surtout le côté ludique de l'apprentissage de la programmation qui est mis en avant.
- Cela encourage ainsi à la récupération, à la « bibouille », correspondant ainsi à l'esprit de ce projet open source, le Raspberry Pi fonctionnant sous Linux

2.5.5 Les différentes utilisations du Raspberry Pi 3

Les différentes utilisations du Raspberry Pi 3 Il est bien évidemment impossible de dresser une liste exhaustive des projets réalisables avec la framboise. Cependant, il est possible de distinguer un certain nombre d'usages globaux de celle-ci.

2.5.5.1 L'utilisation MultiMedia

Il s'agit là d'un des usages les plus développés du Raspberry Pi. Sous l'idée de MultiMedia pointe en fait surtout celle de média-center.

2.5.5.2 La domotique

Le problème avec la domotique c'est qu'il n'existe pas beaucoup d'interfaces complètes, et elles sont assez chères, mais le Raspberry Pi est présent pour résoudre ce problème. En effet, comme il est de taille réduite, il peut facilement s'intégrer dans une pièce. De plus, avec ses nombreuses sorties, notamment GPIO, il peut accueillir de nombreux modules pour communiquer avec des équipements domotiques.

2.5.5.3 Les systèmes embarqués

La framboise étant particulièrement petite, et possédant des entrées GPIO, elle est souvent utilisée dans le cadre de projets liés à l'électronique, en tant que contrôleur central d'un système informatique, etc. De nombreux projets émergent, comme le contrôle d'une voiture télécommandée avec caméra, des drones, ou même un amusant projet sur le piratage de drone.

2.5.5.4 Un ordinateur, tout simplement

On l'oublie souvent, mais avant tout, le Raspberry Pi est un ordinateur. Petit, bon marché, et souvent suffisant pour bien des utilisations. C'est sa raison d'exister. IL fournit une alternative aux personnes n'ayant pas les moyens d'acheter un ordinateur standard. Il permet aussi d'encourager l'apprentissage de la programmation.

2.5.6 Avantage de Raspberry Pi 3

- Petit en taille.
- Pas cher / Faible coût.
- Server Serveur et hébergement à bas prix, capacité à gérer le trafic web.
- Open Source.
- Peut être utilisé comme Super Computer.

2.5.7 Performances et contraintes du Raspberry pi

Les performances du Raspberry Pi seront comparées aux nœuds de capteurs sans fil suivants : [38, 39]

- **MicaZ** - est basé sur l'Atmel ATmega128L qui est un microcontrôleur basse consommation et exécute MoteWorks à partir de sa mémoire flash interne. La radio MICAz (MPR2400) IEEE 802.15.4 offre à la fois une sécurité à haute vitesse (250 kbps) et matérielle (AES-128).
- **TelosB** - regroupe tous les éléments essentiels pour les études en laboratoire dans une plate-forme unique comprenant : une capacité de programmation USB, un IEEE 802.15.4 radio avec antenne intégrée, un microcontrôleur basse consommation avec mémoire étendue et une suite de capteurs en option. Cette plate-forme offre une faible consommation d'énergie permettant une longue durée de vie de la batterie ainsi qu'un réveil rapide de l'état de veille.
- **Iris** - est utilisé pour activer WSN à faible puissance. Iris fournit aux utilisateurs une grande variété d'applications de détection personnalisées offrant jusqu'à trois fois une portée radio améliorée et deux fois la mémoire programme par rapport aux générations précédentes de MICA Motes.
- **Cricket** - est une version géolocalisée du populaire module processeur / radio basse consommation MICA2. Le Cricket Mote comprend tout le matériel standard MICA2 et un émetteur et un récepteur à ultrasons.
- **Lotus** - est basé sur le microcontrôleur NXP LPC1758, 32 bits ARM Cortex-M3. Une seule carte processeur peut être configurée pour exécuter simultanément l'application / le traitement du capteur et la pile de communications réseau / radio. Lotus, comme toutes les plates-formes de nœuds de capteurs mentionnées précédemment, a un débit de données de 250 kb / s.

2.6 Présentation d'Arduino

2.6.1 Définition

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout permettre à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine. Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions [40].

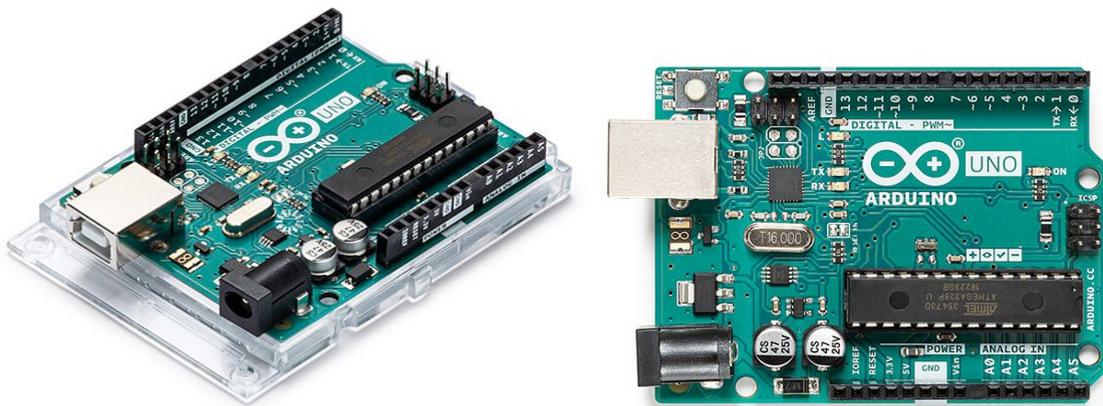


FIGURE 2.5 – Arduino Uno

2.6.2 Caractéristiques de la carte Arduino UNO

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. C'est la plus simple et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enfiler une série de modules complémentaires.

Le microcontrôleur ATmega328 contient un bootloader qui permet de modifier le programme sans passer par un programmeur. Cette carte est livrée avec un support en plastique mais sans cordon USB (voir articles conseillés).

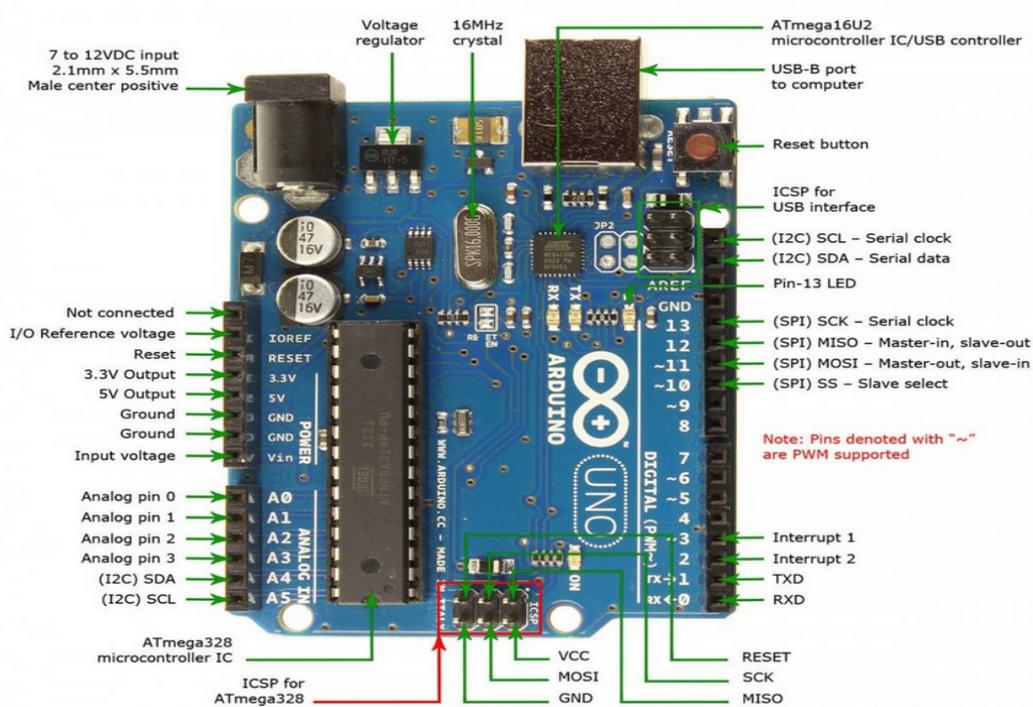


FIGURE 2.6 – Caractéristiques Techniques De La Carte Arduino Uno

2.6.3 Les différents composants de la carte Arduino UNO

2.6.3.1 Les sources d'alimentation de la carte Arduino UNO

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion

USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- 3.3V Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA, la figure 2.7 illustre les différentes parties.

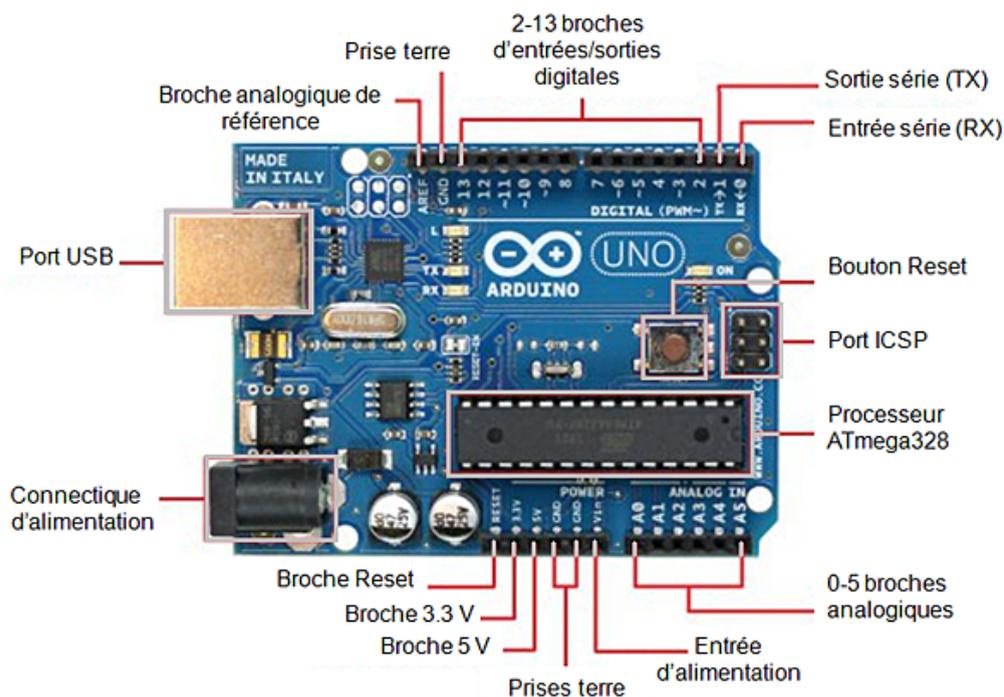


FIGURE 2.7 – Description des entrées/sorties de la carte Arduino Uno [41]

2.6.3.2 Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais

son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :

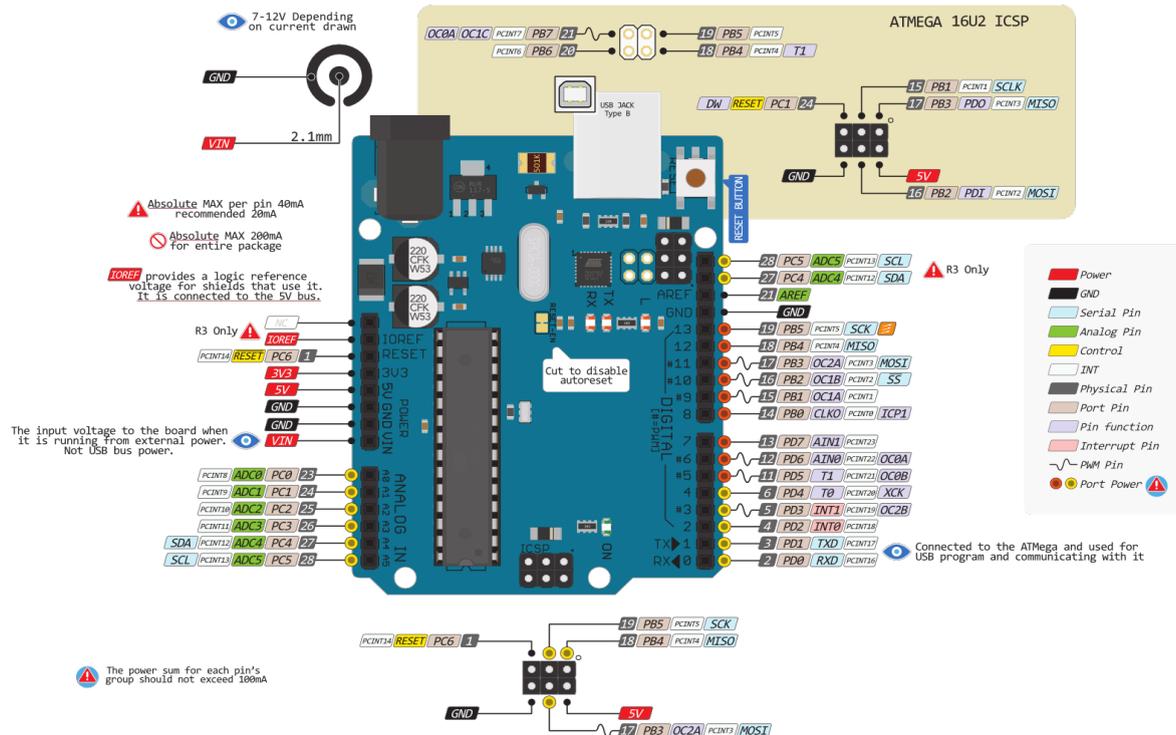


FIGURE 2.8 – Schéma Complet Des Broches

2.7 Composants utilisés

2.7.1 Relais

Un relais est un interrupteur électrique. De nombreux relais utilisent un électro-aimant pour actionner mécaniquement un commutateur, mais d'autres principes de fonctionnement sont également utilisés, tels que les relais à semi-conducteurs. Les relais sont utilisés lorsqu'il est nécessaire de contrôler un circuit par un signal séparé de faible puissance, ou lorsque plusieurs circuits doivent être commandés par un signal. Les premiers relais ont été utilisés comme amplificateurs dans les circuits télégraphiques longue distance : ils ont répété le signal venant d'un circuit et l'ont retransmis sur un autre circuit. Les relais étaient largement utilisés dans les centraux téléphoniques et les premiers ordinateurs pour effectuer des opérations logiques.

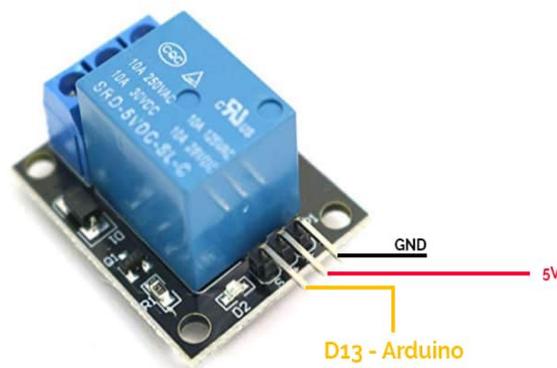


FIGURE 2.9 – Relais

2.7.2 Transistor

Un transistor est un composant électrique à semi-conducteur qui sert à amplifier le courant ou obtenir un courant quand il est commandé en tension. Il s'agit d'un tri-pole actif qui possède trois électrodes.

2.7.2.1 Transistor bipolaire

Il s'agit très certainement du type de transistor le plus employé, aussi bien dans le domaine des basses fréquences que des hautes fréquences, des faibles puissances que des hautes puissances. On l'appelle d'ailleurs tout simplement "Transistor".

Il en existe des dizaines de milliers de références, de type NPN ou PNP, mais il est tout de même possible de les classer par ordre de puissance, de fréquence, de gain. Ce qui avouons-le, permet de choisir plus facilement un remplaçant le jour où on ne trouve plus l'original... Un transistor bipolaire possède trois électrodes nommées E (émetteur), C (collecteur) et B (base).. Le transistor bipolaire peut être utilisé en base commune, en émetteur commun ou en collecteur commun, mais c'est en montage émetteur commun qu'on le retrouve le plus souvent utilisé (l'émetteur est commun à l'entrée et à la sortie) [42].

Exemples : 2N2222, 2N2907, BC107, BC557, etc...

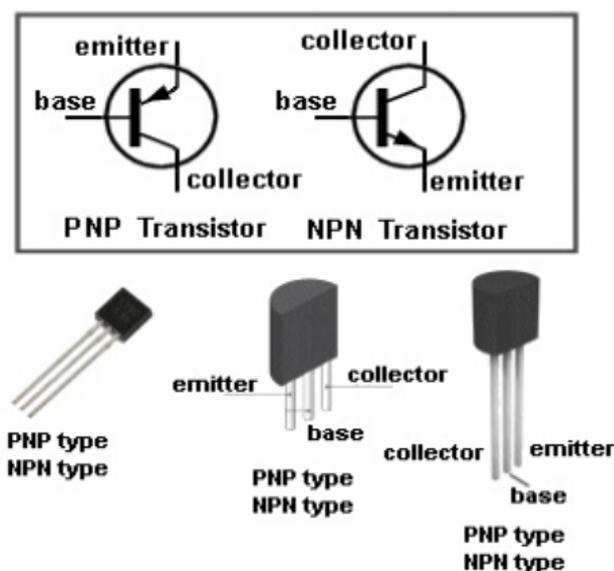


FIGURE 2.10 – Différent Types De Transistor

2.7.2.2 Transistor BC547

Le transistor BC547 est un petit transistor polyvalent et économique, idéal pour réaliser toutes sortes de circuits électroniques standard. C'est un transistor à jonction bipolaire NPN principalement utilisé à des fins d'amplification et de commutation. Il a un gain de courant maximal de 800. [43]

2.7.3 Module Bluetooth HC-05

Le Bluetooth est un protocole de communication sans fil, permettant l'échange bidirectionnel de données à une distance bien précise (15 à 20 mètres), en utilisant des ondes radio UHF sur une bande de fréquence de 2.4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant les liaisons filaires.

La norme « Bluetooth » fut mise au point en 1994 par le fabricant Suédois Ericsson à Lund. Le terme signifie littéralement en français : « dent bleue ».

Le module HC-05 n'est pas plus gros qu'un pouce. Il est en fait un montage d'un module Bluetooth sur un petit P.C.B, cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V et 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de LED de

signal, tout cela est déjà intégré.

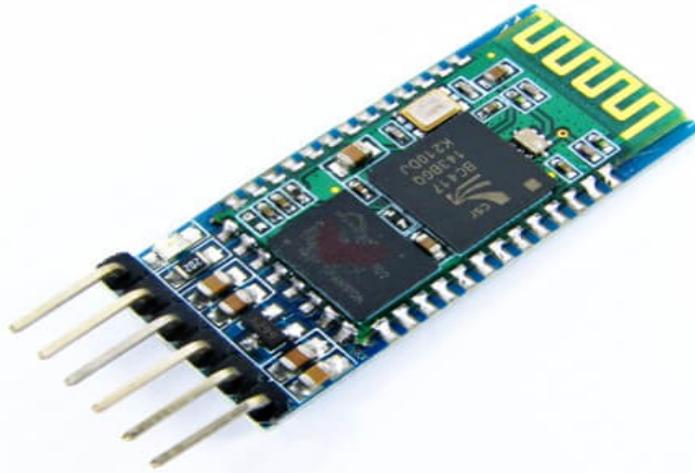


FIGURE 2.11 – Module Bluetooth HC-05

Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion.

- VCC broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino.
- GND masse. Typiquement connectée à la broche GND de l'Arduino
- RX broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino
- TX broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino
- State retourne 1 lorsque le module est connecté
- Key ou EN doit être alimentée pour entrer dans le mode de configuration et ne doit pas être connecté pour être en mode communication.

La particularité du module Bluetooth HC-05 est qu'il peut être utilisé en mode esclave (dans ce cas il est équivalent à un module HC-06 et s'utilise de la même manière) ou en mode maître, ce qui veut dire qu'il peut de manière autonome se connecter à un autre module Bluetooth (HC-06, HC-05 en mode esclave ou autre) sans action extérieure [44].

Deuxième partie

PARTIE EXPÉRIMENTALE

CHAPITRE 3

CONCEPTION ET MISE EN ŒUVRE

Sommaire

3.1	Introduction	57
3.2	Schéma Bloc Globale	57
3.3	Systèmes Globale	57
3.4	Langage de Programmation Python	58
3.5	Approche proposée	63
3.6	Conversion d'un Signal vocale à un texte	65
3.7	Explication du Code Arduino	69

3.1 Introduction

DANS ce chapitre, nous allons présenter en détail le principe du système de développement et le contenu de base des algorithmes, et concevoir chaque partie du système pour obtenir un schéma complet et précis.

L'objectif de ce chapitre est de mettre l'accent sur la description des différents algorithmes et code qu'on a utilisés pour l'application de ce projet.

3.2 Schéma Bloc Globale

Le système à base de la carte Raspberry et Arduino UNO, prend en charge la réception de l'entrée qui est un signal vocal ils le transfert, ensuite l'accord d'un traitement de données réglementaire par adresses et enfin la partie isolation pour appliqué l'ordre.



FIGURE 3.1 – Schéma Bloque Globale du Système

3.3 Systèmes Globale

Le système basé sur la carte Raspberry Pi est responsable de la réception des signaux sonores (voix) et de l'envoi de diverses informations entre les périphériques : il gère et maintient le circuit et les connexions de fonctionnement de tous les capteurs ou actionneurs, puis Arduino UNO connecte le relais qui est dans notre carte de commande réalisé pour commandé, sans oublier le passage par le traitement des classe de intelligence artificiel qui est implanté dans le Raspberry à laide de langage de programmation python.

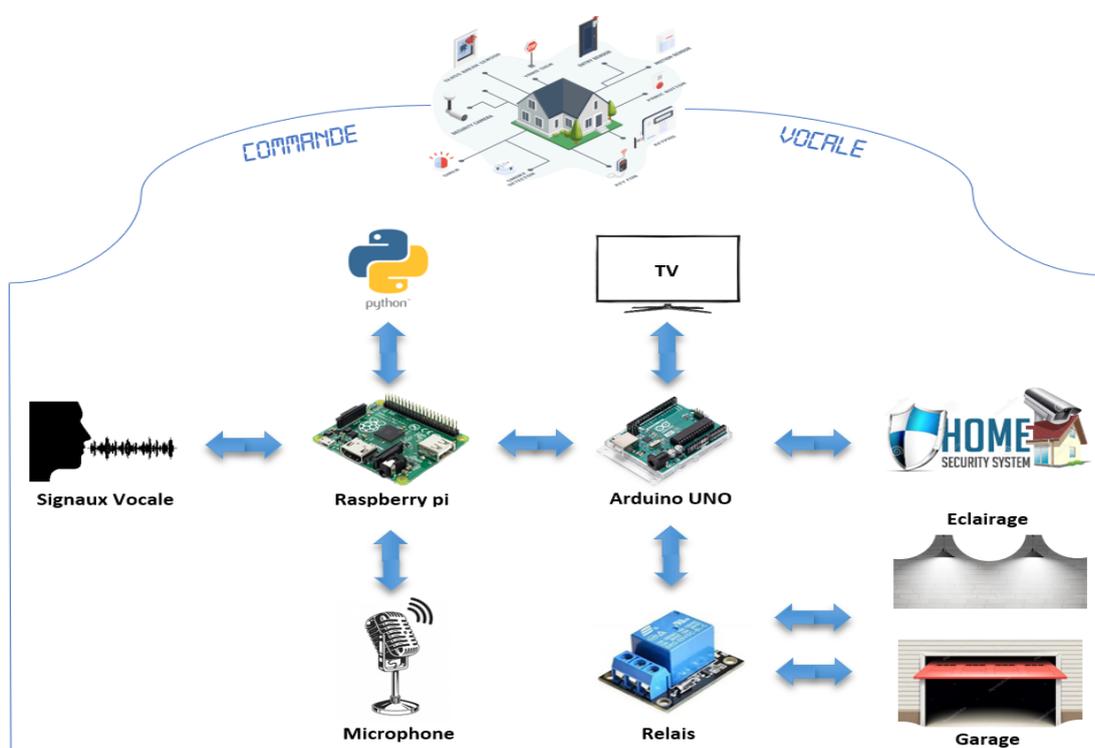


FIGURE 3.2 – Système Globale RPi et Arduino

3.4 Langage de Programmation Python

Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par **Guido van Rossum**, et publié pour la première fois en 1991. Python repose sur une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces significatifs. Il fournit des constructions permettant une programmation claire à petite et grande échelle. Python propose un système de typage dynamique et une gestion automatique de la mémoire. Il prend en charge plusieurs paradigmes de programmation, notamment orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard étendue et complète [45].

3.4.1 Installation des outils et premier programme Python

Afin de pouvoir développer en langage Python, vous devez installer les outils nécessaires :

1. Télécharger et installer le langage Python depuis [le site officiel Python](#).
2. Télécharger et installer un IDE Python : de nombreux choix s'offre à vous : Pycharm, PyScripter, Wing. Quant à moi je vous recommande wing, en raison de rapidité et de simplicité d'usage, en plus il est gratuit : [Télécharger Wing Personnel \[45\]](#).

3.4.2 Côté logiciel

3.4.2.1 Python Pip

Utilisation de base : [Dans le site de L'Installation de modules Python](#)

Dans la commande de windows PowerShell (Admin) et ont exécute les commandes suivant, Les outils standards de création de paquets sont tous conçus pour être utilisés à partir de la ligne de commande.

La commande suivante va installer la dernière version d'un module et ses dépendances depuis le Python Package Index :

```
1 python -m pip install SomePackage
```

Note

- Pour les utilisateurs POSIX (y compris Mac OS X et Linux), les exemples de ce guide supposent l'utilisation d'un environnement virtuel.
- Pour les utilisateurs de Windows, les exemples de ce guide supposent que l'option proposant de modifier la variable d'environnement PATH à été cochée lors de l'installation de Python.

Il est aussi possible de préciser une version minimum exacte directement depuis la ligne de commande. Utiliser des caractères de comparaison tel que >, < ou d'autres caractères spéciaux qui sont interprétés par le shell, le nom du paquet et la version doivent être mis entre guillemets :

```
1 python -m pip install SomePackage=1.0.4 # specific version
2 python -m pip install "SomePackage>=1.0.4" # minimum version
```

Normalement, si un module approprié est déjà installé, l'installer à nouveau n'aura aucun effet. La mise à jour de modules existants doit être demandée explicitement :

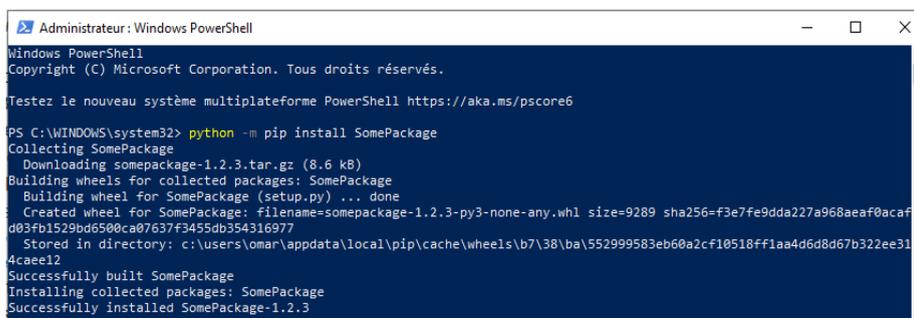
```
1 python -m pip install --upgrade SomePackage
```

Plus d'informations et de ressources concernant pip et ses capacités peuvent être trouvées dans le [Python Packaging User Guide](#).

La création d'environnements virtuels est réalisée grâce au module [venv](#). Installer des paquets au sein de l'environnement virtuel courant utilise les commandes montrées précédemment.

Voir aussi

[Guide Utilisateur de l'Empaquetage Python : Installer les paquets de la distribution Python.](#)



```
Administrateur: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> python -m pip install SomePackage
Collecting SomePackage
  Downloading somepackage-1.2.3.tar.gz (8.6 kB)
Building wheels for collected packages: SomePackage
  Building wheel for SomePackage (setup.py) ... done
  Created wheel for SomePackage: filename=somepackage-1.2.3-py3-none-any.whl size=9289 sha256=f3e7fe9dda227a968aaef0acaf
d03fb1529bd6500ca07637f3455db354316977
  Stored in directory: c:\users\omar\appdata\local\pip\cache\wheels\b7\38\ba\552999583eb60a2cf10518ff1aa4d6d8d67b322ee31
4caee12
Successfully built SomePackage
Installing collected packages: SomePackage
Successfully installed SomePackage-1.2.3
```

FIGURE 3.3 – Installation Du pip Python Validé

3.4.2.2 Installation sur RasPi

Pour installer le système d'exploitation Linux Raspbian, il faut qu'on utilise un logiciel qui s'appelle NOOBS (New Out Of Box Software). Ce logiciel gratuit est un système de récupération et d'installation rapide du Raspberry Pi.

L'utilisation d'une carte mémoire de capacité égale ou supérieure à 8 Gb est importante. Dans notre cas on va utiliser une carte originale Sandisk ultra 16 Gb.

Le formatage de carte mémoire est nécessaire et obligatoire. L'utilisation de logiciel comme SD Formatter est recommandée.

Nous avons maintenant une carte SD prête. On doit télécharger le logiciel NOOBS qui est disponible au site <https://www.raspberrypi.org/downloads/noobs>

Raspberry Pi Desktop

Compatible with:
PC and Mac



Debian Buster with Raspberry Pi Desktop

Release date: January 11th 2021
Kernel version: 4.19
Size: 2.948MB
[Show SHA256 file integrity hash:](#)

[Download](#)

[Download torrent](#)

FIGURE 3.4 – Logiciel Du Raspberry Pi

Cette version de logiciel contient le système d'exploitation Raspbian Linux.

Après le téléchargement on doit extraire le contenu de fichier compressé dans la carte mémoire qu'on a insérer dans le PC.

Une fois le transfert de fichier est terminé. On retire la carte mémoire de pc et on l'insert à notre Raspberry Pi.

Ensuite on branche le clavier, la souris, le câble HDMI, après qu'on vérifie le bon branchement de tous les équipements on alimente le Raspberry.

Remarque : l'insertion de carte mémoire est strictement interdite après l'allumage de RasPi. On branche le HDMI à un écran compatible comme un téléviseur.

- Au démarrage, NOOBS propose de choisir le système que nous souhaitons installer sur notre carte SD On choisit Raspbian et on clique sur Install.

L'opération prend quelque minute donc NOOBS nous invite à faire un peu de lecture.

Après le redémarrage de RasPi l'interface graphique de Raspbian s'apparie [46].

3.4.3 Présentation de «IDE » Arduino

Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel d'Arduino <https://www.arduino.cc/en/software>

L'IDE Arduino permet :

- D'éditer un programme : des croquis (sketch en Anglais),
- De compiler ce programme dans le langage « machine » de l'Arduino,
- De téléverser le programme dans la mémoire de l'Arduino,
- De communiquer avec la carte Arduino grâce au terminal,

A l'ouverture, l'interface visuelle du logiciel est présentée dans la figure 3.5 :

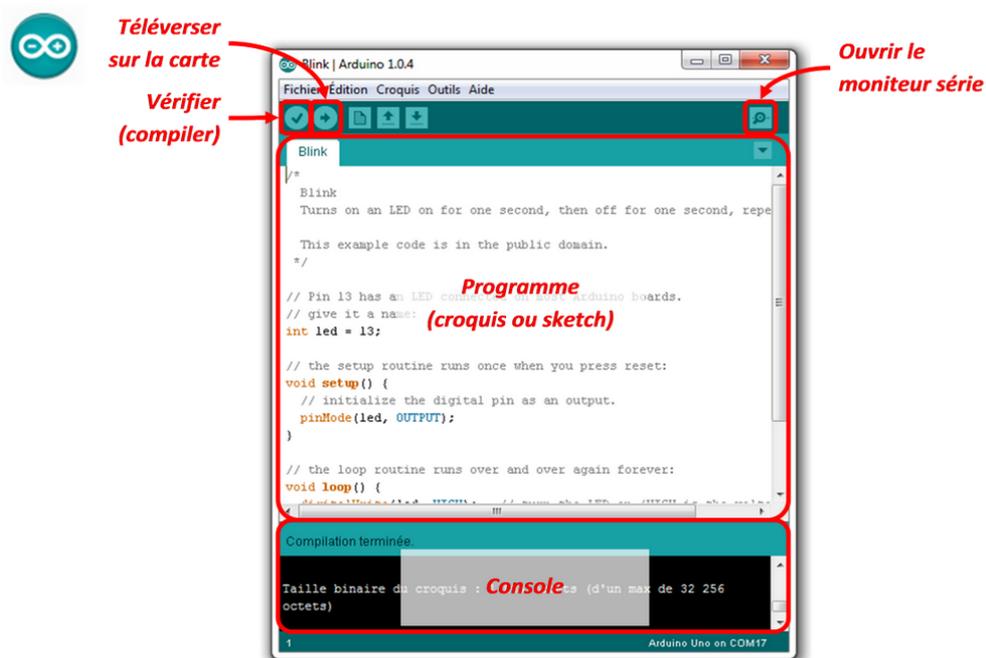


FIGURE 3.5 – L'interface visuelle du IDE Arduino

3.4.4 Programmer avec Arduino

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne, La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

3.4.4.1 Structure d'un programme Arduino

La structure d'exécution du code arduino est élaboré dans la figure 3.6

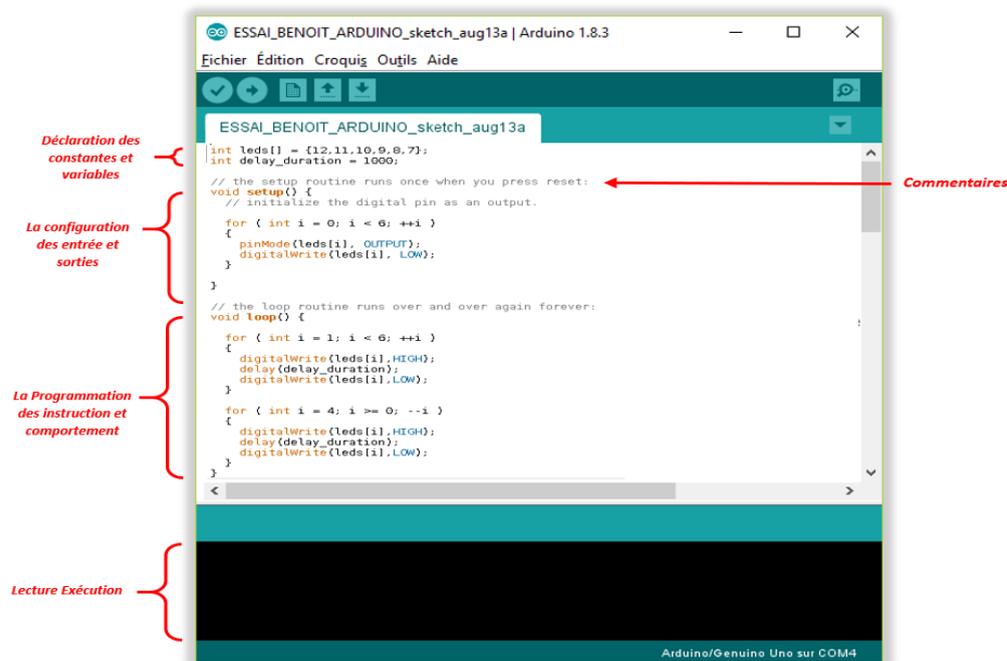


FIGURE 3.6 – Structuration d'un programme Par Arduino

3.5 Approche proposée

Pour cette solution, on développe conjointement la carte microcontrôleur et le logiciel. Le développement de la carte et du logiciel associé comprend trois phases :

- Développement de la carte et du programme simultanément par quelque Bibliothèques : `pyaudio` et `pytorch` et aussi `sqlLite3`....ect
- Association de différents code du site github
- Utilisation d'un émulateur pour tester cet ensemble (IDLE Python et Commande CMD windows PowerShell du windows).

On reporte ci-dessous, dans le figure 3.7 l'organigramme de l'application globale. On note que certains points n'ont pas été implémenté par faute de temps, et parce qu'ils semblent plus élémentaires par rapport à ce qui a été déjà fait.

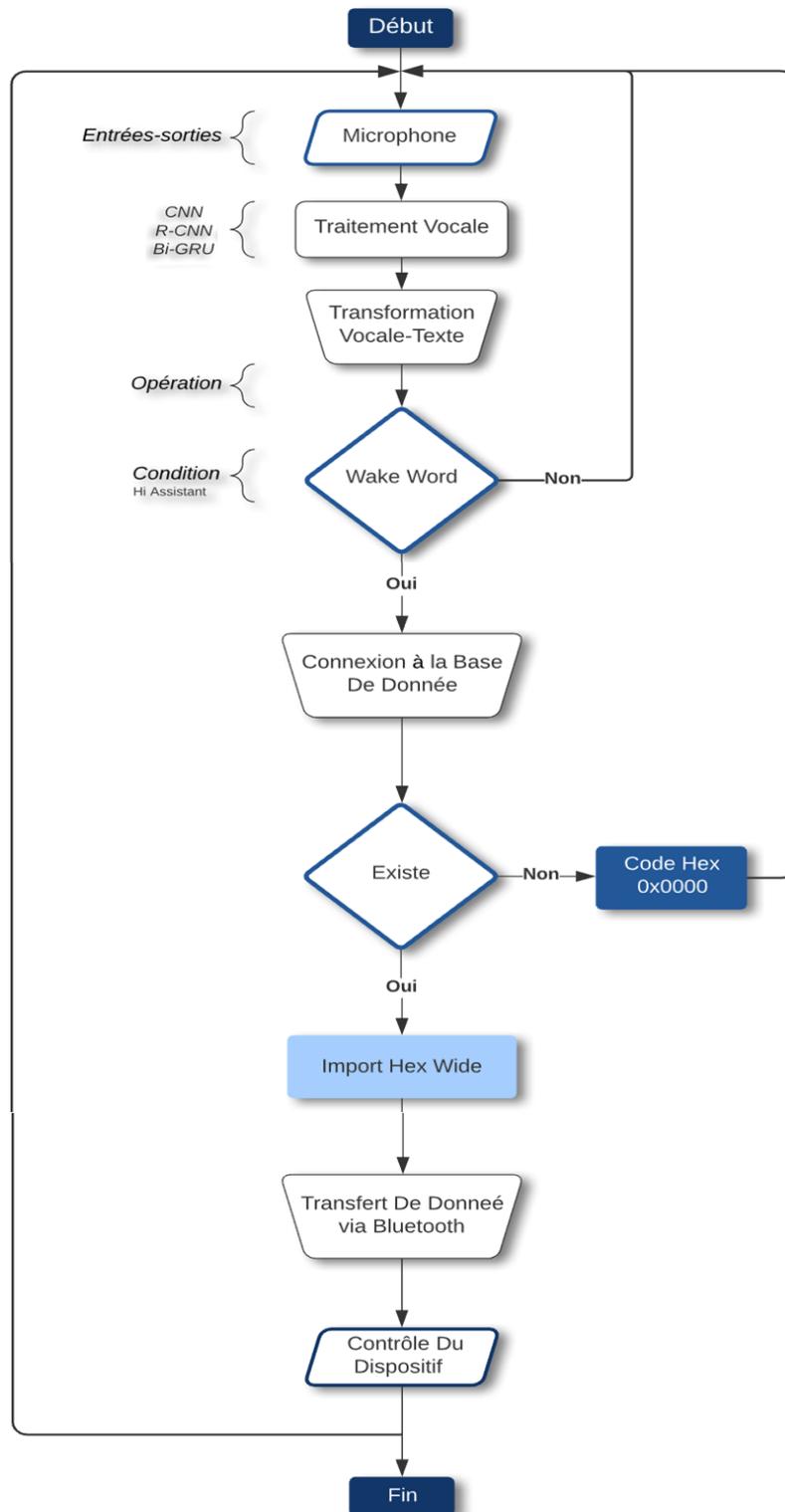


FIGURE 3.7 – Organigramme du déroulement du programme

Par la suite on a dans la figure 3.8 l’organigramme de contrôle qui déclare les adresses approuveront tout les commande des surface utilisé.

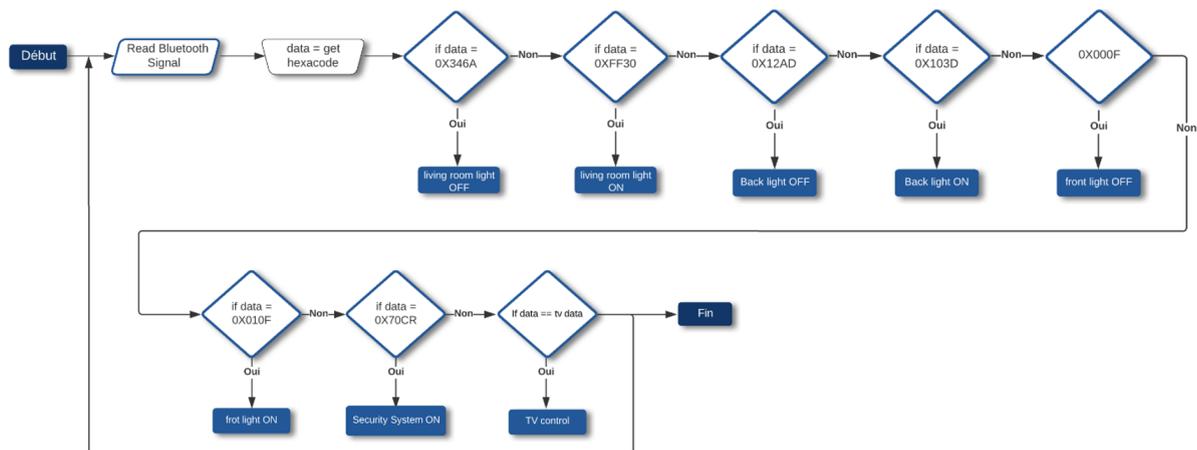


FIGURE 3.8 – Organigramme de contrôle

3.6 Conversion d'un Signal vocale à un texte

En utilisant le compilateur VSCode, nous écrivons un script contenant des classes pour traiter le signal, ces classes peuvent entrer le signal, former, tester et renvoyer un texte extrêmement proche de notre discours.

3.6.1 Class Bidirectional GRU

la Class Bidirectional GRU (**CB-GRU**) à moins d'opérations par rapport au LSTM et, par conséquent, ils peuvent être entraînés beaucoup plus rapidement que les LSTM le code et optimisé par une méthode qui traite le signale plus convenablement avec une contrôlabilité et précision avec les signaux.

```

1 class BidirectionalGRU(nn.Module):
2
3 def __init__(self, rnn_dim, hidden_size, dropout, batch_first):
4     super(BidirectionalGRU, self).__init__()
5     self.BIGRU = nn.GRU(input_size = rnn_dim, hidden_size =
6         ↪ hidden_size, num_layers = 1, batch_first = batch_first,
7         ↪ bidirectional=True
8     )
9
10    self.layer_norm = nn.LayerNorm(rnn_dim)
11    self.dropout = nn.Dropout(dropout)
12
13    def forward(self, x):
14        x = self.layer_norm(x)
15        x = F.gelu(x)
  
```

```

13     x, _ = self.BIGRU(x)
14     x = self.dropout(x)
15     return x
16

```

3.6.2 Class CNN

Class CNN (**C-CNN**) Consiste Les traits CNN textuels qui sont extraits et entraînés à partir des représentations vectorielles des mots (on parle alors on a des caractéristiques importantes). Ces traits sont appris par le réseau neuronal jusqu'à ce que le comportement désiré soit obtenu.

```

1  class ResidualCNN(nn.Module):
2
3  def __init__(self, in_channels, out_channels, kernel, stride, dropout, n_feats):
4      super(ResidualCNN, self).__init__()
5
6      self.cnn1 = nn.Conv2d(in_channels, out_channels, kernel, stride, padding =
7          ↪ kernel//2)
8      self.cnn2 = nn.Conv2d(out_channels, out_channels, kernel, stride, padding =
9          ↪ kernel//2)
10     self.dropout1 = nn.Dropout(dropout)
11     self.dropout2 = nn.Dropout(dropout)
12     self.layer_norm1 = CNNLayerNorm(n_feats)
13     self.layer_norm2 = CNNLayerNorm(n_feats)
14
15 def forward(self, x):
16
17     residual = x # (batch, channel, feature, time)
18     x = self.layer_norm1(x)
19     x = F.gelu(x)
20     x = self.dropout1(x)
21     x = self.cnn1(x)
22     x = self.layer_norm2(x)
23     x = F.gelu(x)
24     x = self.dropout2(x)
25     x = self.cnn2(x)
26     x += residual
27     return x # (batch, channel, feature, time)

```

3.6.3 Class Speech Recognition

Class Speech Recognition (C-SR)

```

1 class SpeechRecognitionModel(nn.Module):
2     #speech recognition Model inspired by deepspeech 2
3
4     def __init__(self, n_cnn_layers, n_rnn_layers, rnn_dim, n_class, n_feats,
5     ↪ stride=2, dropout=0.1):
6         super(SpeechRecognitionModel, self).__init__()
7         n_feats = n_feats//2
8         self.cnn = nn.Conv2d(1, 32, 3, stride=stride, padding=3//2) # cnn for
9     ↪ extracting heirachal feature
10
11     #n residual cnn layers with filter size of 32
12     self.rescnn_layers = nn.Sequential(*[
13         ResidualCNN(32, 32, kernel=3, stride=1, dropout=dropout, n_feats = n_feats)
14         for _ in range(n_cnn_layers)
15     ])
16     self.fully_connected = nn.Linear(n_feats*32, rnn_dim)
17     self.birnn_layers = nn.Sequential(*[
18         BidirectionalGRU(rnn_dim=rnn_dim if i==0 else rnn_dim*2,
19         hidden_size=rnn_dim, dropout=dropout, batch_first=i==0)
20         for i in range(n_rnn_layers)
21     ])
22     self.classifier = nn.Sequential(
23         nn.Linear(rnn_dim*2, rnn_dim),
24         nn.GELU(),
25         nn.Dropout(dropout),
26         nn.Linear(rnn_dim, n_class)
27     )
28
29     def forward(self, x):
30         x = self.cnn(x)
31         x = self.rescnn_layers(x)
32         sizes = x.size()
33         x = x.view(sizes[0], sizes[1] * sizes[2], sizes[3]) # (batch, feature, time)
34         x = x.transpose(1, 2) # (batch, time feature)
35         x = self.birnn_layers(x)
36         x = self.classifier(x)
37         return x

```

3.6.4 Code Principale

Code Principale (C-P)

```

1  from speech_recognition import speech_analizing
2  from responding import *
3  from database import database
4  import serial
5  import time
6
7  assistant = speech_analizing()
8  database = database()
9  connection = serial.Serial(port="COM4",baudrate=9600, timeout= 1)
10 connection.flushInput()
11 tabel_name = 'data'
12 while True :
13
14     print ("the system's wakedUp")
15     record = assistant.listen().lower()
16     print(record)
17     control_command = ''
18     wakeWord = ['assistant' ,
19                 'hi assistant',
20                 'hey assistant',
21                 'assistant ask',
22                 'hello assistant',
23                 'assistant ask for'
24                 ]
25     for item in wakeWord:
26         if item in record:
27             control_command = record[len(item)+1:]
28     print(control_command)
29     check_and_return_hex_code = database.manage_data(tabel_name,control_command)
30     print(check_and_return_hex_code)
31     request = check_and_return_hex_code
32     if request == "0x0000" :
33         assistant.answer(failure_message)
34     else :
35         try :
36             connection.write(request.encode('utf-8'))
37             # wakeButton= connection.read().decode('utf-8')
38         except :
39             assistant.answer(failure_transfer)
40
41
42     ## set tow new value to databases
43     # 0x0EF6      open garage
44     # 0x0EA4      close garage
45
46

```

3.7 Explication du Code Arduino

Code Arduino (C-Arduino) Dans cette étape, nous essayons de lire les données avec le modèle bluetooth et de contrôler nos objets à l'aide d'arduino uno et de certains composants électroniques à l'aide du code suivant

```
1 #include <IRremote.h>
2 #include <SoftwareSerial.h>
3
4 const int garage = 8;
5 const int living_room_light = 7;
6 const int back_light = 10;
7 const int front_light = 11;
8 const int security_sys = 12;
9 //IR to the pin 3;
10 //led to the pin 13;
11
12 int address = 32;
13
14 SoftwareSerial connection(0, 1);
15 IRsend irsend;
16
17 String data ;
18
19 void TV_control(byte data,int address){
20   irsend.sendNEC(data,address);
21 }
22
23 void setup() {
24   Serial.begin(9600);
25   delay(50);
26
27   pinMode(garage, OUTPUT);
28   pinMode(living_room_light, OUTPUT);
29   pinMode(back_light, OUTPUT);
30   pinMode(front_light, OUTPUT);
31   pinMode(security_sys, OUTPUT);
32
33 }
34
35 void loop() {
36   while (Serial.available()>0){
37     data = Serial.readString();
38
39     if(data== "0x346A" )    digitalWrite(living_room_light,LOW);
40     else if (data=="0xFF30") digitalWrite(living_room_light,HIGH);
41     else if (data=="0x0EF6") digitalWrite(garage,HIGH);
42     else if (data=="0x0EA4") digitalWrite(garage,LOW);
```

```
43     else if (data=="0x12AD") digitalWrite(back_light, LOW);
44     else if (data=="0x103D") digitalWrite(back_light, HIGH);
45     else if (data=="0x000F") digitalWrite(front_light, LOW);
46     else if (data=="0x01AF") digitalWrite(front_light, HIGH);
47     else if (data=="0x7070B04F") TV_control(0x7070B04F,address); //mute
48     else if (data=="0x7070A857") {
49         for (int i=0; i<6 ; i++) TV_control(0x7070A857,address); //sound low
50     }
51     else if (data=="0x707028D7") {
52         for (int i=0; i<6 ; i++) TV_control(0x707028D7, address); //sound up
53     }
54     else if (data=="0x707030CF") TV_control(0x707030CF, address); //turn off/on
55     else if (data=="0x70CF")digitalWrite(security_sys, HIGH); //security systeme
56 }
57 delay(50);
58 }
```

CHAPITRE 4

RÉALISATION DE LA CARTE DE COMMANDE

Sommaire

4.1	Introduction	72
4.2	Côté matériel (Hardware)	72
4.3	Fonctionnement	76
4.4	Réalisation de la carte PCB	77
4.5	Carte PCB 2D	79
4.6	Carte PCB 3D	80
4.7	Essai pratique	83
4.8	Perspectives	87

4.1 Introduction

CE chapitre décrit la phase de validation de notre approche. Il permettra d'évaluer et de valider nos stratégies proposées et Au cours de cela, nous allons présenter l'expérimentation de notre travaille, Après une étude générale des différents éléments constituant notre carte électronique on passe maintenant à la simulation et la réalisation physique de notre projet. Dans cette partie la programmation est prête on ce lance pratiquement par réalisation du Kit du développement.

4.2 Côté matériel (Hardware)

4.2.1 Équipements

Afin de réaliser la carte de contrôle, nous avons besoin de composants ajustés pour qu'il n'y ait pas de chute de tension, et en choisissant la disponibilité la plus rentable et relativement peu coûteuse, nous citons des cristaux 16 Mhz avec ces deux capacités et des condensateurs de filtrage électrique, un microcontrôleur ATMEGA avec boot-loader des diode pour empêcher le courant de revenir au transistor et aussi 4 Relais pour contrôler la commande et des LED pour visualiser les résultats, régulateur 5v, et enfin le conducteur électrique, l'aspect réel des composants est représenté dans la figure 4.1.

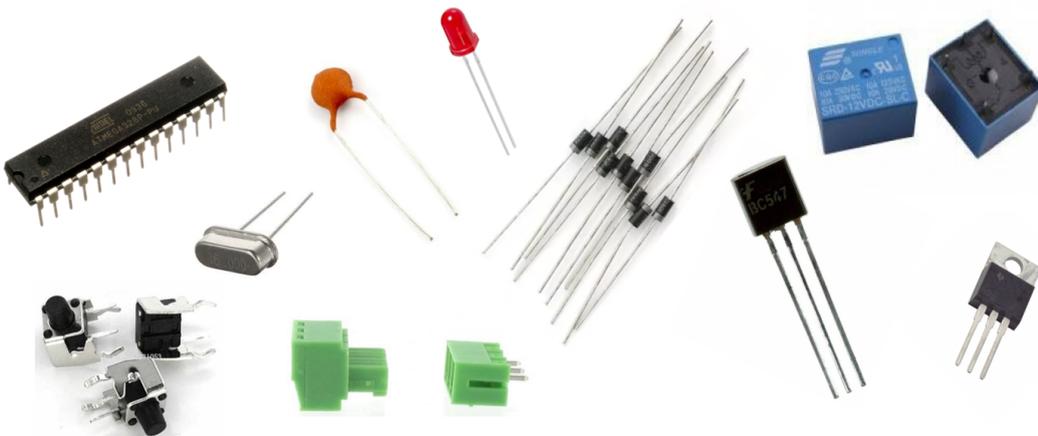


FIGURE 4.1 – Aspect réel des composants utilisé

La liste des composants détaillé est prescrite dans le tableau 4.1

Tableau 4.1 – Équipements Générale

ID	Nom	Désignateur	Quantité
1	Condensateur électrique 10 μ F	CAP-TH-BD5.0-P2.00-D0.8-FD	2
2	Condensateur céramique 22pF	CAP-TH-L3.5-W3.0-P2.54-D0.8	2
3	ATMEGA328P-PU Avec bootloader	DIP-28-L34.6-W7.3-P2.54-LS10.2-BL	1
4	Diode 1N4001	DO-41-BD2.4-L4.7-P8.70-D0.9-RD	4
5	Connecteur 2PIN	CONN-TH-EH508-140S02ASM1	4
6	Relais 12V/220V	RELAY-TH-Y3F-SS-109D	4
7	LED (1x Red,5x Green,1x White)	LED-TH-BD3.8-P2.54-FD	7
8	Transistor BC547	TO-92-3-L4.9-W3.7-P1.27-L	4
9	Résistance 6 x 600 ohm 5 x 10K ohm 2 x 220 ohm	RES-TH-BD2.4-L6.3-P10.30-D0.6	13
10	Régulateur 7805	TO-220-3-L10.0-W4.5-P2.54-L	1
11	SWITCH,6*6*5PLASTIC HEAD,260G, 0.25MM,DIP2	KEY-TH-2P-L6.0-W6.0-P5.00	1
12	Connecteur 3PIN	CONN-TH-EH508-140S03ASM1	1
13	Crystal 16MHz 7805	HC-49S-L11.0-W4.4-P4.88	1

4.2.2 Schémas électroniques globale du système

Les Schémas électroniques sont conçu ou devisé par des bloques ensuite ses derniers forme un seule schéma maître.

Nous avons une alimentation 12v pour aimanté le relais et convertir 12 à 5V en partie commande via un régulateur de tension le schéma d'alimentation est représenté dans la figure 4.2

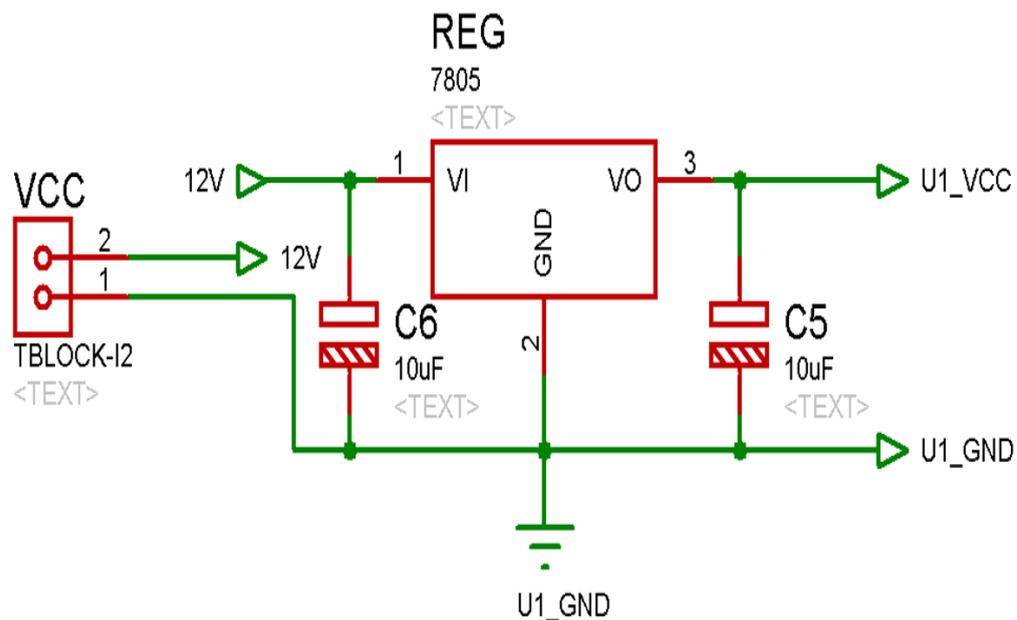


FIGURE 4.2 – Alimentation

Ensuite, il y'a Le schéma d'entrée élaboré dans la figure 4.3 qui représente deux parties :

- Le Bluetooth importe le signal on utilisent un codec, le SBC. Néanmoins, celui-ci peut avoir tendance à largement compresser le signal sonore.
- Le microcontrôleur traite ce signal à l'aide des partie mémoires et bien sur la partie programmation pour géré les ordres demandé.

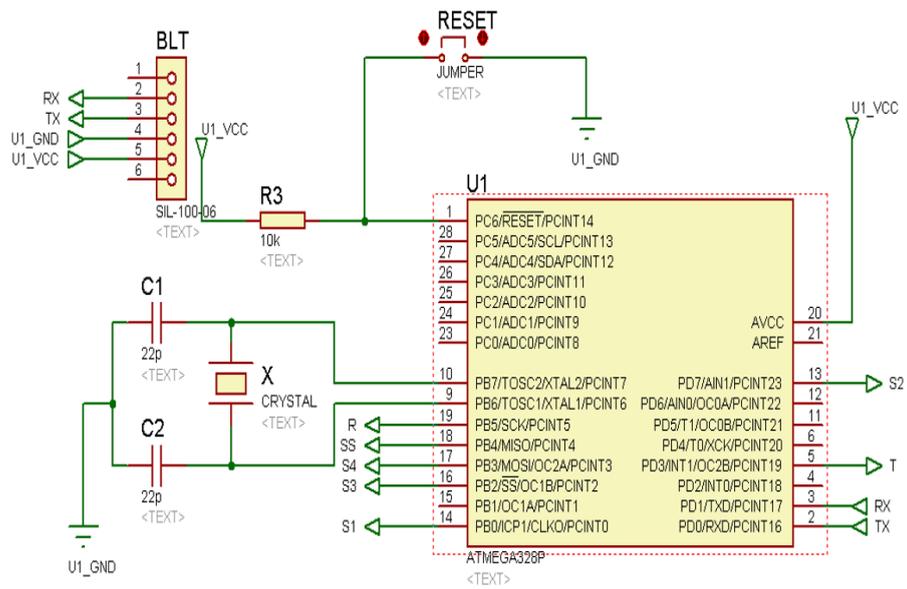


FIGURE 4.3 – Schéma D’Entrée

Le transistor qui est contrôlé de la base de données par un signal qui est lancé par le microcontrôleur ce transistor est alimenté par son émetteur avec 12 V et ainsi le collecteur du transistor et relie a la masse alors le transistor commande le relais ce dernier à une diode réversible de protection contre le retour du courant et aussi une LED pour qu’on sache si’il est en marche ou non, enfin par cela le relais sen charge de la commande finale, ceci est figuré dans la figure 4.4

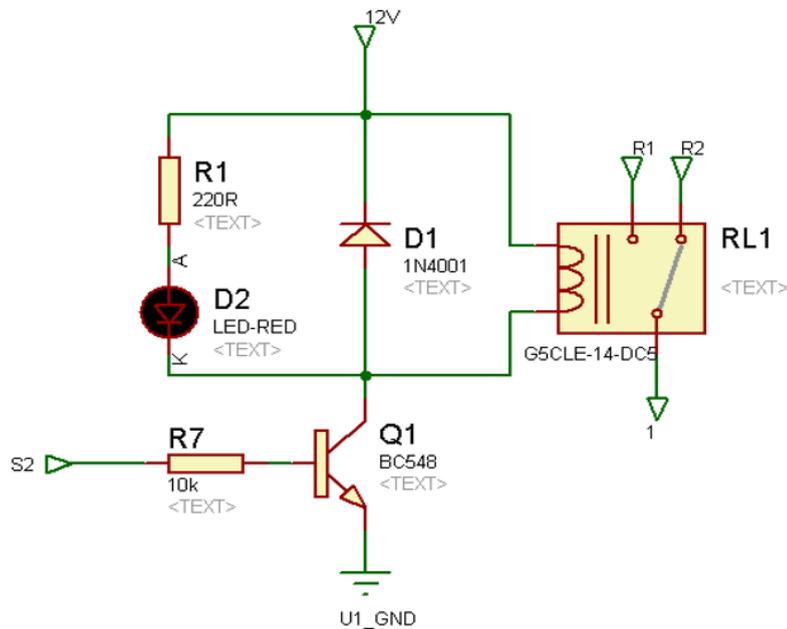


FIGURE 4.4 – Schéma de Sortie

Le schéma ci-dessous, illustré dans la figure 4.5, est extrait des différents blocs précédemment décrits. Nous avons conçu une structure optimale pour que le KIT de la carte soit petit et l'impression du PCB soit de petite forme.

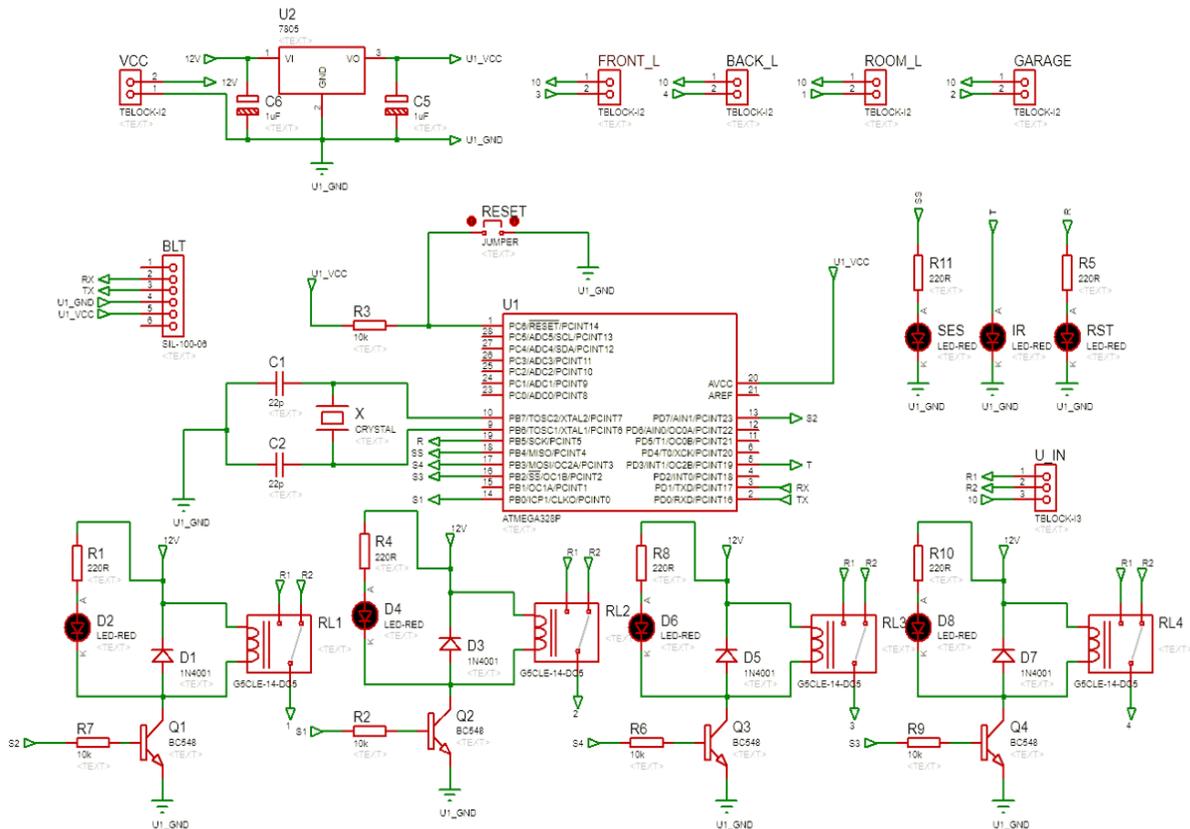


FIGURE 4.5 – Schéma électronique générale

4.3 Fonctionnement

Ce système reçoit tous les mots par voix de la langue anglaise il faut citer "wake word" pour l'exécuter, c'est la formule de lancement du système qui est dans ce cas "assistant, hi assistant, assistant ask, assistant ask for " quand le système c'exécute ont démarrent les commandes qu'on veut réaliser.

Ici, on veut démarrer la commande d'allumage des lampes d'une chambre et la cour de devant et le système de sécurité et aussi le téléviseur de la chambre Donc le Système ce connecte à la base de données qu'on à réalisé avec **Sql lite** il cherche la commande cité des paramètre précédant , s'il la trouve il revient au code **hexadécimale**.

Par Exemple pour allumer l'éclairage de la chambre on dit « hi assistance wake word » il est prêt pour qu'il reçoit le message donc on dit « turn on living room light » après il recherche dans la base de donnée il exerce le code 0xFF30 le système connecte le code à la classe api c'est une classe pour que le système connecte et comprend les données reliant entre le système de contrôle et le système et aussi le système de traitement , donc il envoie le code Hex par Bluetooth si la commande n'est pas défini il revient au code 0x0000 et il affiche le message d'erreur suivant «sorry i can't recognize you».

Ensuite on revient au système de contrôle et le système de traitement vocal qui lit le code qu'il reçoit du Bluetooth après vérification et il essaye d'exécuter la commande d'allumage de la lampe de la chambre par la procédure suivante : fournit 5 v à la sortie du brochage de la lampe et connecte avec le transistor qui lui contrôle l'allumage de la lampe avec l'ouverture du relais 220v qui allume la lampe.

4.4 Réalisation de la carte PCB

Ici, nous concevons notre PCB en utilisant le site Web <https://easyeda.com>

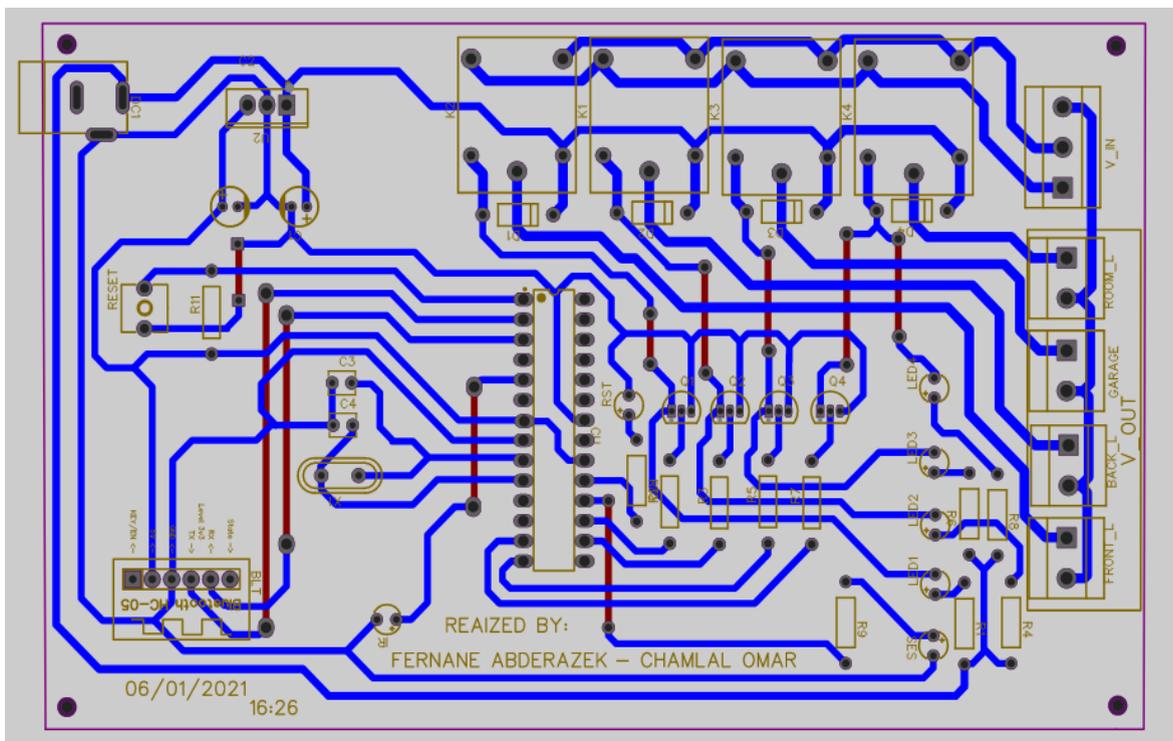


FIGURE 4.6 – Carte PCB avec un Arrière-plan Gris

Ce dernier est utilisé pour concevoir des schémas électroniques et des plaques PCB, nous pouvons également réaliser notre carte électronique et gravé dessus les composants utilisé dans le schéma à l'aide du typon finale.

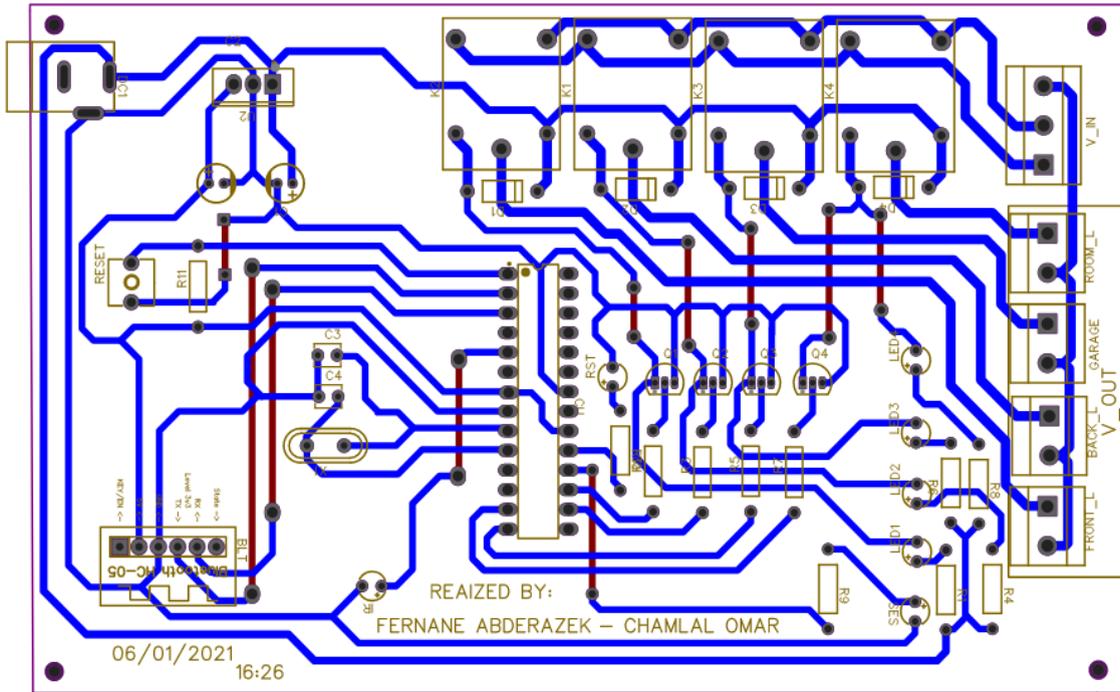


FIGURE 4.7 – Carte PCB avec un Arrière-plan Blanc

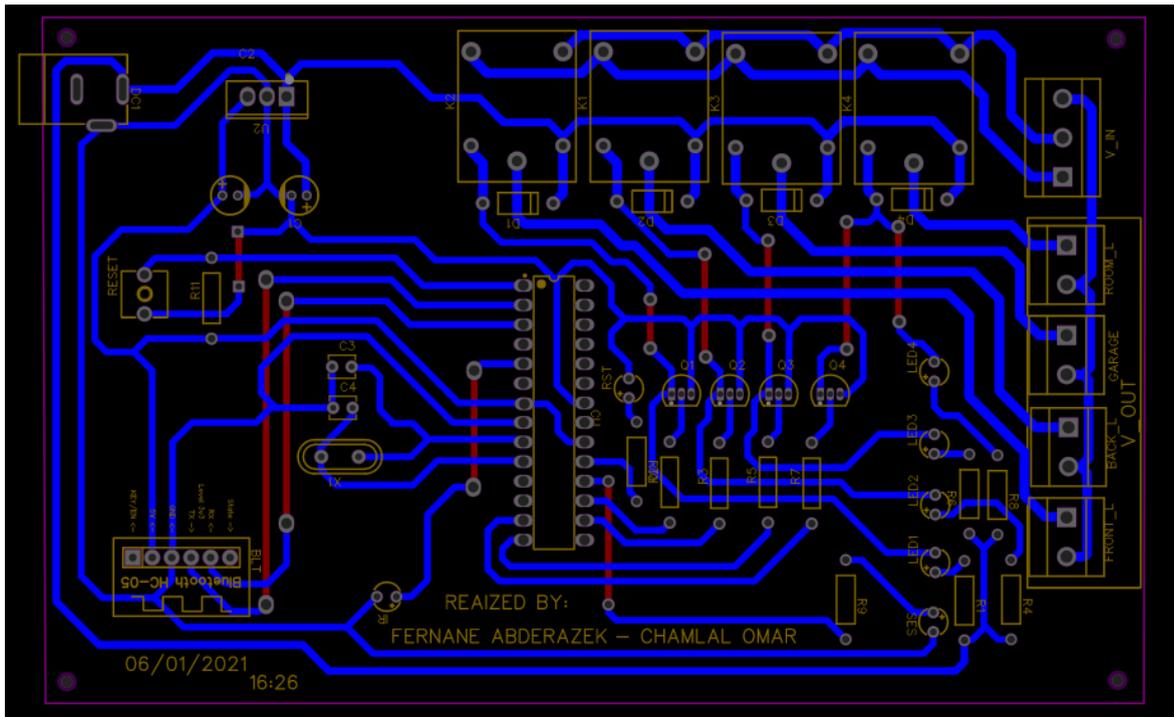


FIGURE 4.8 – Carte PCB avec un Arrière-plan Noir

4.5 Carte PCB 2D

En utilisant la vue 2D et 3D nous verrons notre carte électronique finale avec et sans composant, le résultat est représenté dans les figures 4.9 et 4.10.

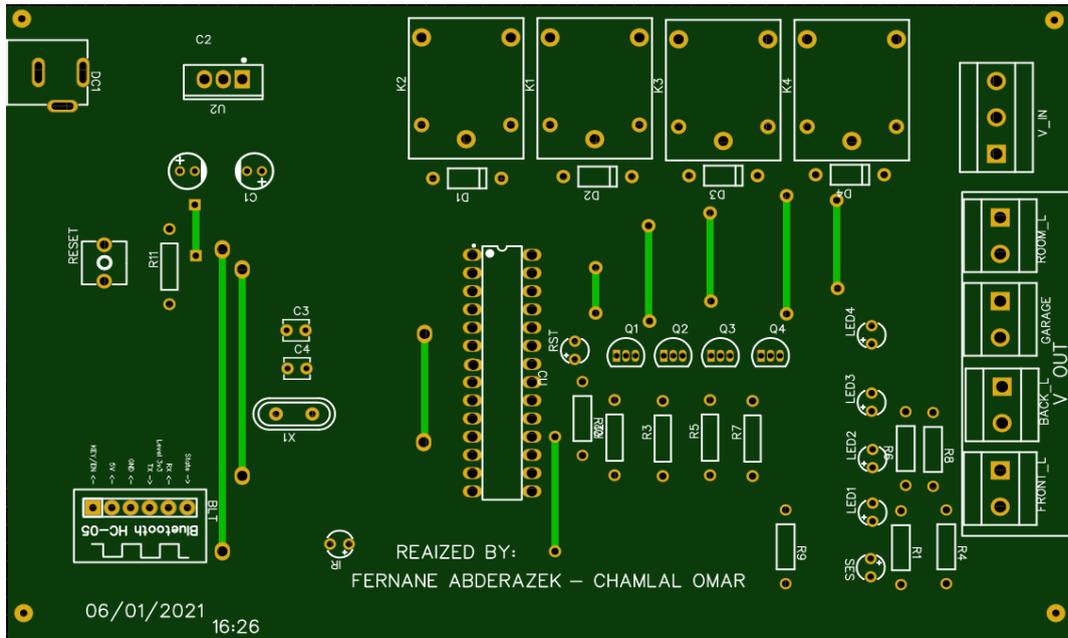


FIGURE 4.9 – Carte PCB en 2 Dimensions

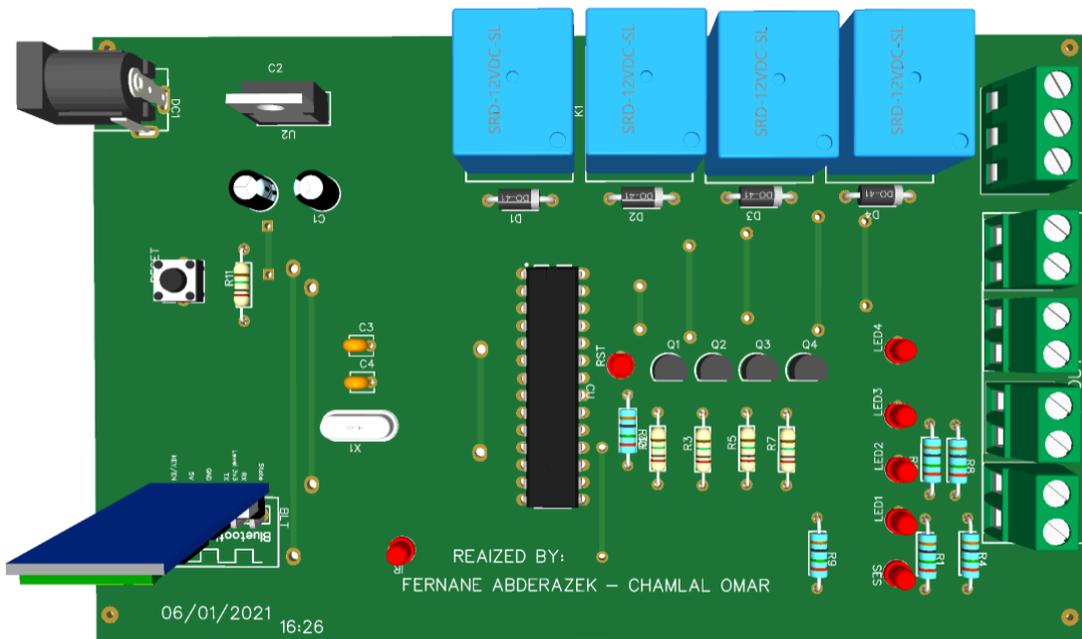


FIGURE 4.10 – Carte PCB en 2 Dimensions avec composant Visible

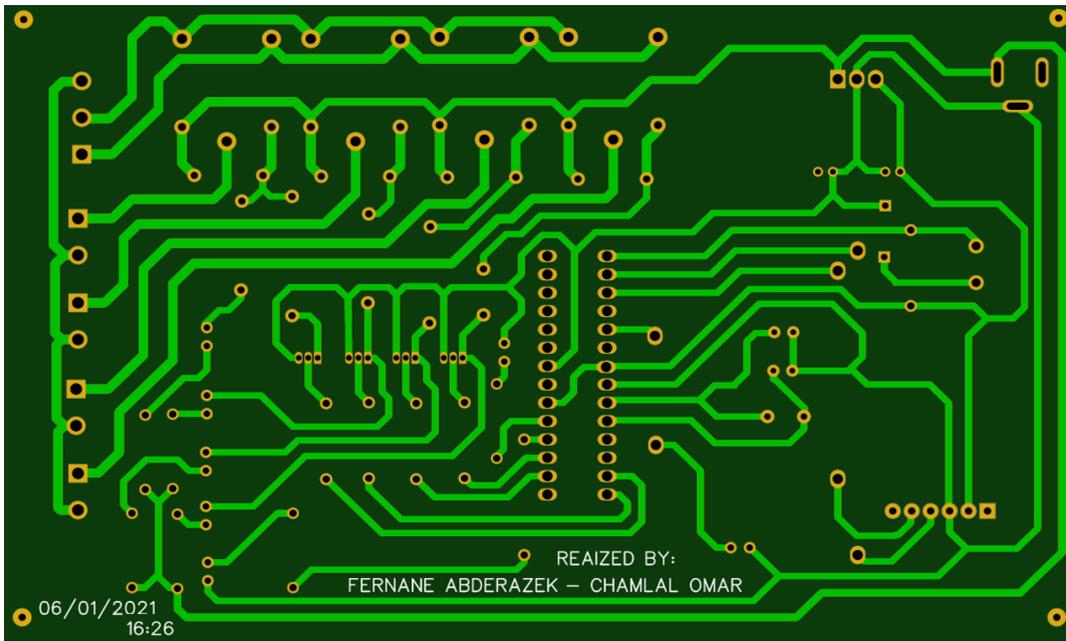


FIGURE 4.11 – Phase Arrière De la Carte PCB en 2 Dimensions

4.6 Carte PCB 3D

Vous pouvez trouver tout sur ce projet dans notre compte sur Easyeda

https://easyeda.com/DZ_TECH/home-controlling-using-microcontroller

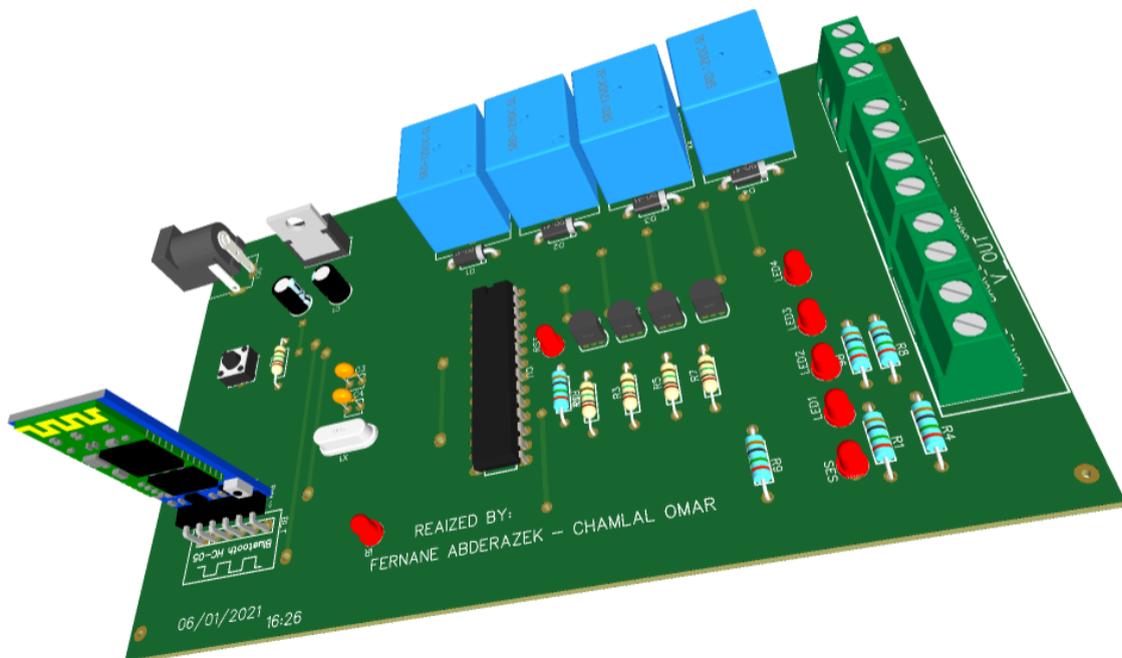


FIGURE 4.12 – Carte PCB en 3 Dimensions 'Coté Gauche'

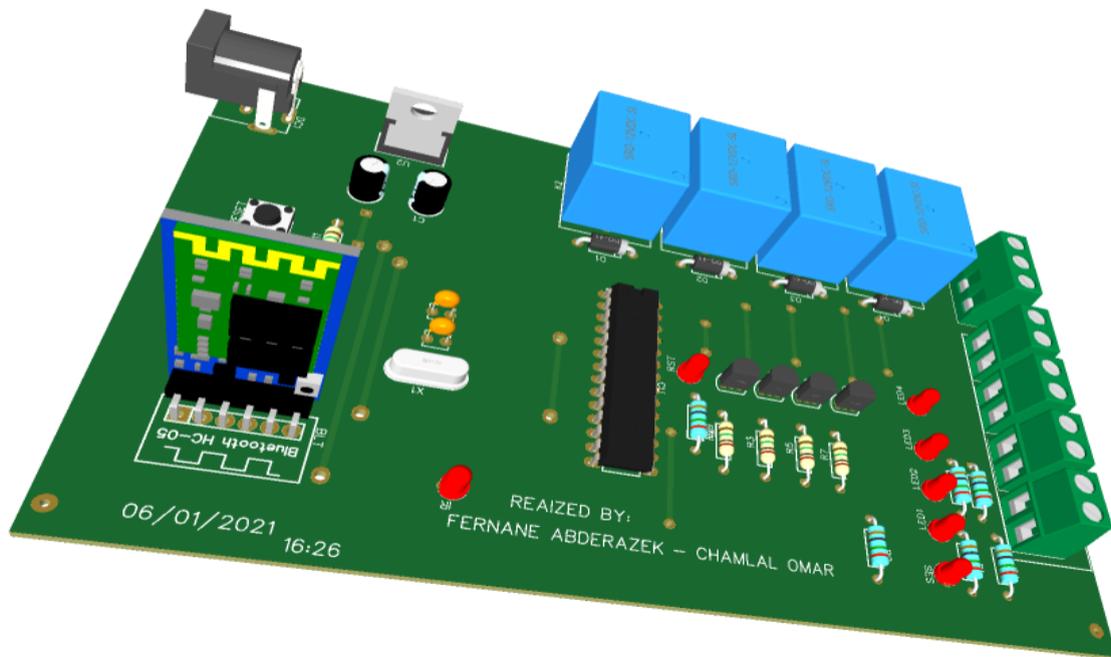


FIGURE 4.13 – Carte PCB en 3 Dimensions 'Côté Droit'

On a utilisé une méthode classique pour la réalisation de notre plaque PCB ,on débute par l'impression du typon crée précédemment, ensuite on place le typon sur la plaque de cuivre par la chaleur avec le fer à repassé le résultat est obtenu dans le figure 4.14

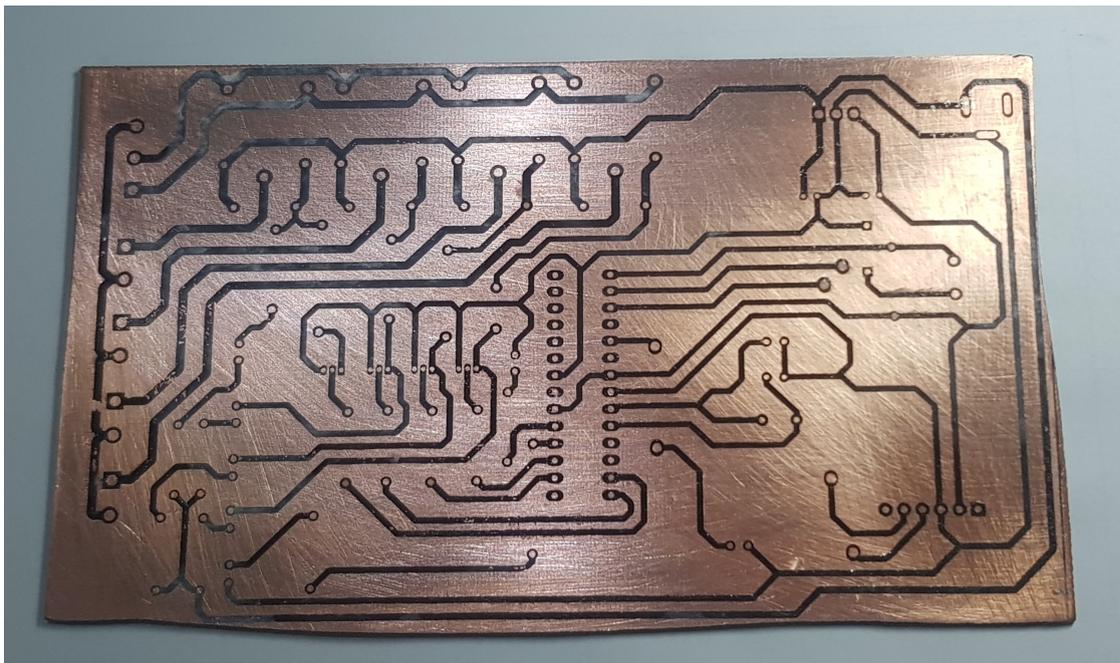


FIGURE 4.14 – Typon imprimé dans la plaque de cuivre

Par la suite, la plaque baigne dans le perchlorure de fer dans quelques minutes pour isoler les pistes de cuivre en interprétant le typon on voit l'application dans la figure 4.15.



FIGURE 4.15 – Baignade de la plaque dans le perchlorure de fer

Enfin, le résultat final après nettoyage et pressage de la carte est illustré dans la figure 4.16

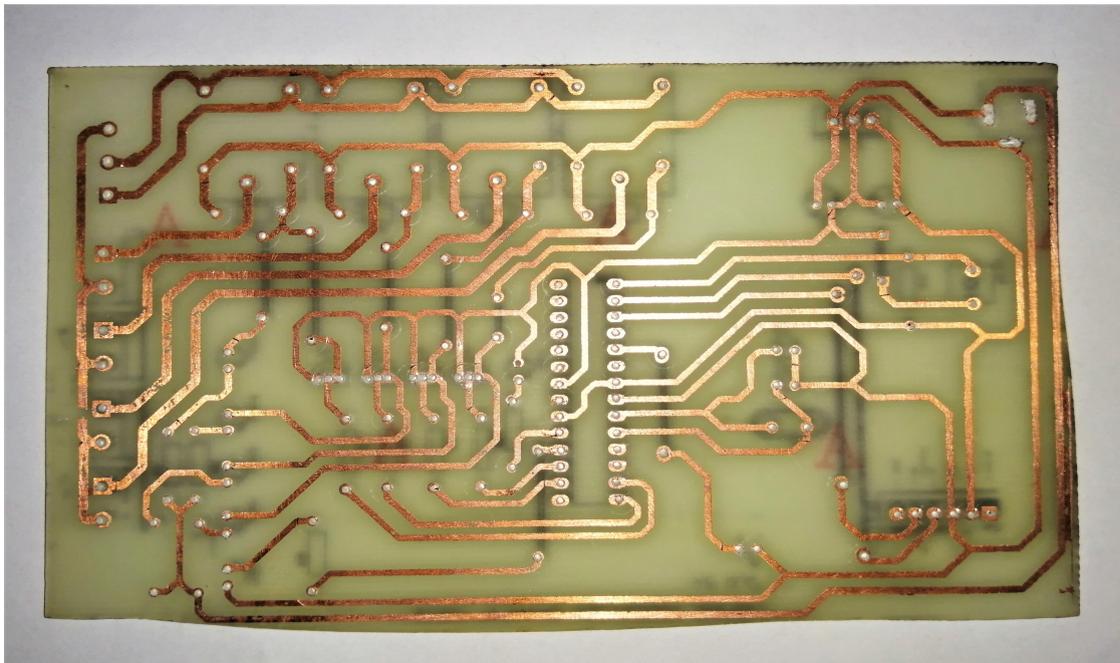


FIGURE 4.16 – Circuit électronique de la plaque

Le côté des composants apparaît dans la figure 4.17 cela est conçu avec du papier et verni à l'inverse de la plaque.

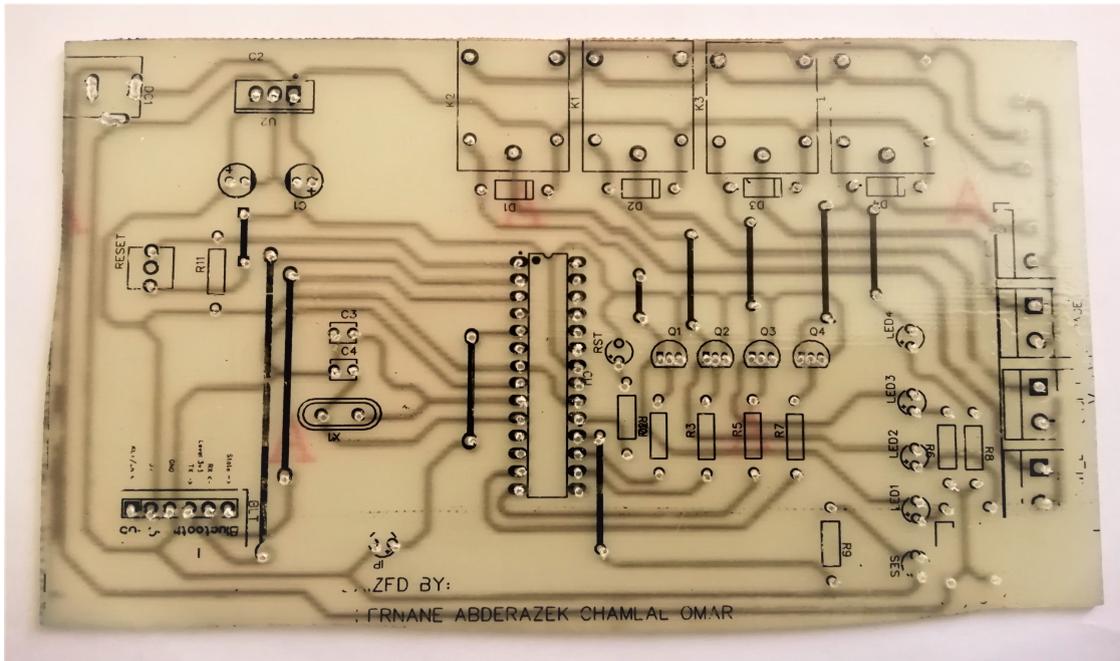


FIGURE 4.17 – L'interface de la plaque Électronique

4.7 Essai pratique

Dans ce cas on va planifier un essai pratique du système , on débute par la lampe de chambre (living room) on a arrivé a allumé cette lampe et encore l'éteindre et notre résultat et pourvu dans la Led qui est branché a la sortie du Microcontrôleur reliev par le transistor et en revanche on a branché un ampèremètre parallèle avec la LED pour voir la valeur de la tension qui franchi cette dernière, la description est vu dans la figure 4.18.

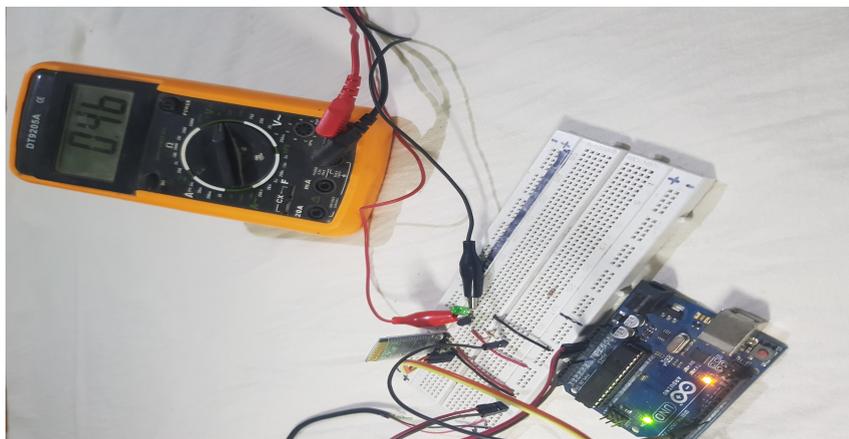
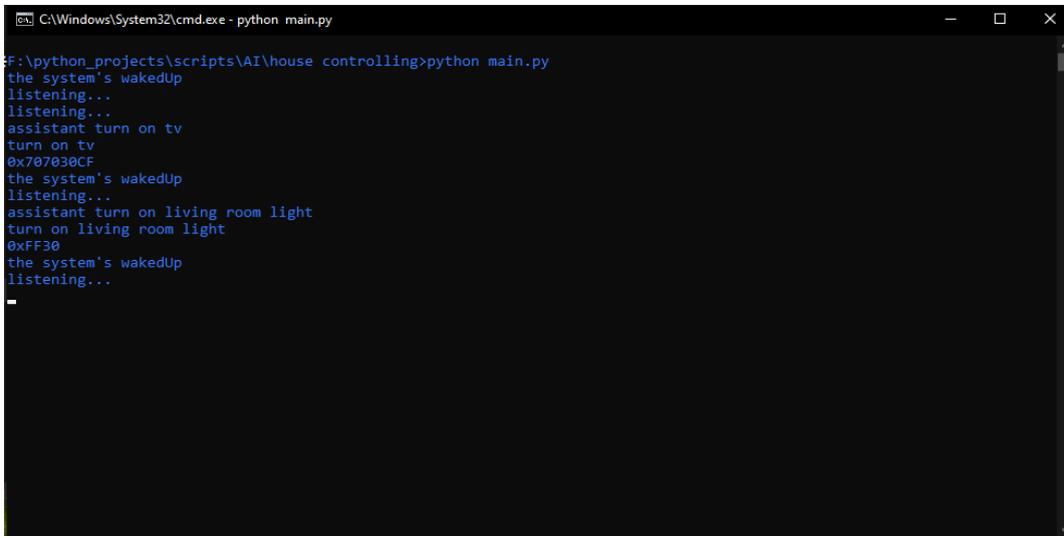


FIGURE 4.18 – Premier Essai démonstrative

4.7.1 Le Cas D'allumage de la lampe

Temporairement pour l'essayage on utilise le pc a la place du Raspberry Pi pour exécuter le code on relie nous commande avec l'arduino, ensuite on a implémenté le main du code python avec déclaration de la commande d'allumage de la lampe par exemple (hi assistante trun on living room light) cela est présenté dans la figure 4.19 ci-dessous :



```
C:\Windows\System32\cmd.exe - python main.py
F:\python_projects\scripts\AI\house controlling>python main.py
the system's wakedUp
listening...
listening...
assistant turn on tv
turn on tv
0x707030CF
the system's wakedUp
listening...
assistant turn on living room light
turn on living room light
0xFF30
the system's wakedUp
listening...
-
```

FIGURE 4.19 – Exécution avec Cmd pour la commande ON

L'obtention du résultat pratique dans le système est représenté dans la figure 4.20 :

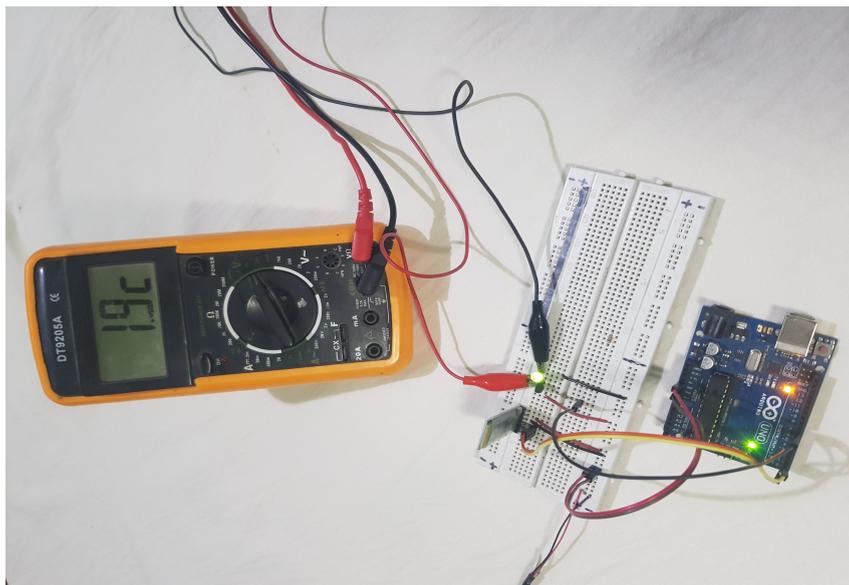
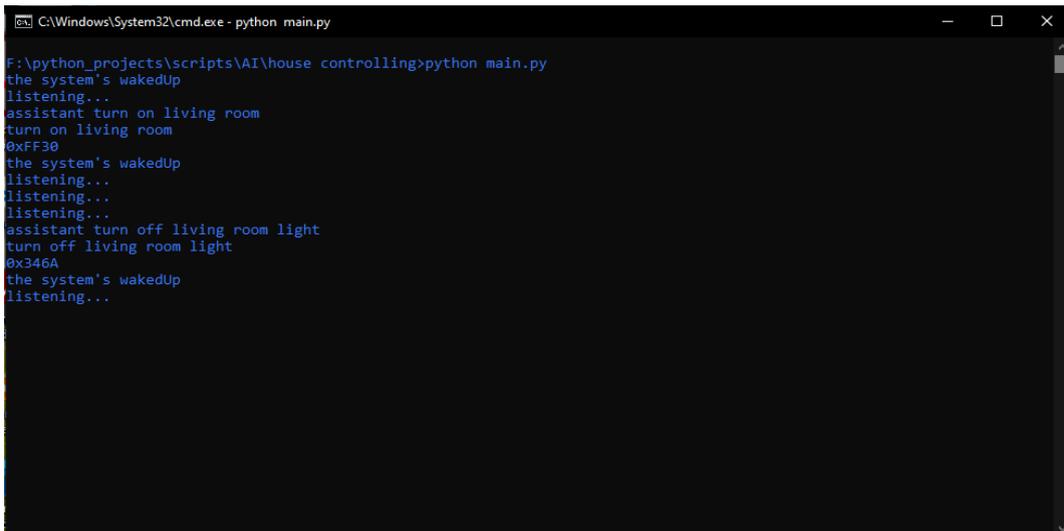


FIGURE 4.20 – Allumage de la lampe

4.7.2 Le Cas d'éteindre la lampe

Pour ce cas on déclare la commande pour que la lampe s'éteigne (assistant turn off living room light) on a obtenu la réponse qui est présentée dans la figure 4.21 suivante :



```
cmd.exe - python main.py
F:\python_projects\scripts\AI\house controlling>python main.py
the system's wakedUp
listening...
assistant turn on living room
turn on living room
0xFF30
the system's wakedUp
listening...
listening...
listening...
assistant turn off living room light
turn off living room light
0x346A
the system's wakedUp
listening...
```

FIGURE 4.21 – Exécution avec Cmd pour la commande OFF

Enfin, on voit les résultats dans le système électronique (Arduino) dans la figure 4.22 :

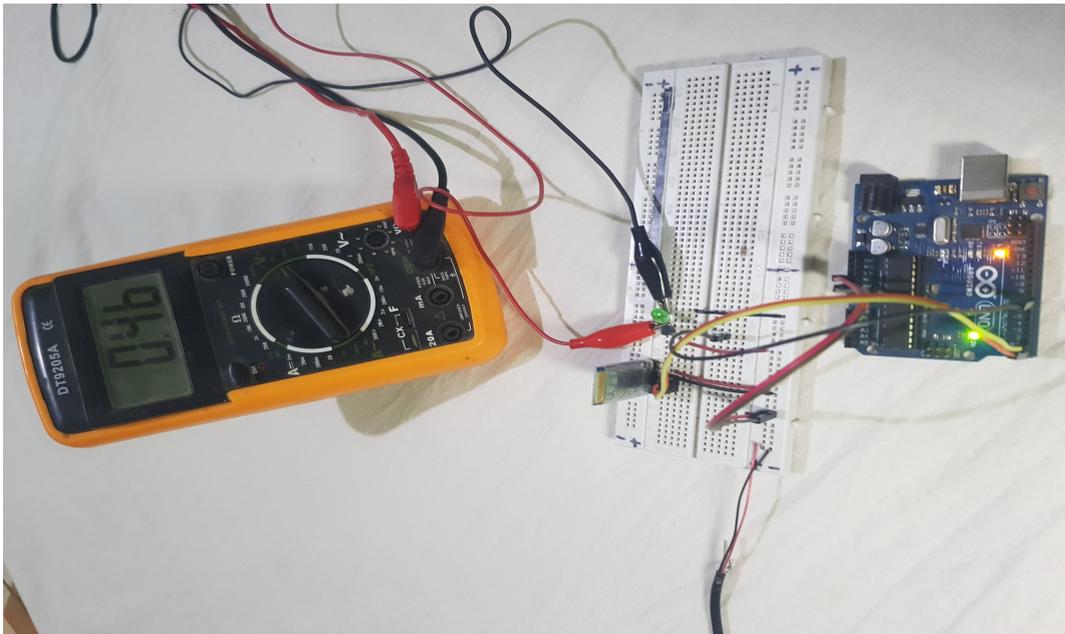


FIGURE 4.22 – Lampe éteinte

CONCLUSION GÉNÉRALE

Pour les travaux futurs, nous envisageons à considérer que les assistant Vocale jouent un rôle de plus en plus important car ce sont eux qui permettant de mesurer les effets des phénomènes de toutes natures qui agissent l'environnement de l'homme.

Le travail qui nous a été confié était basée principalement sur la réalisation d'un Assistant Vocale.

Pour réaliser ce projet, nous avons tout d'abord défini les besoins et les objectifs à atteindre selon les spécifications de notre détecteur. Ainsi, nous avons fait des recherches théoriques qui ont une relation directe à la compréhension de la conception de notre montage.

Par la suite nous avons passés à la partie pratique pour réaliser notre assistant vocale et nous avons fait les différents essais pour atteindre notre objectif et montré que le système marche convenablement.

Au cours de ce travail, nous avons employé plusieurs classe des réseaux de neurone que nous avons recherché qui sont la base de tout cette réalisation que nous conseillons a appliqué dans les différents domaine techniques.

Nous avons également rencontré des cas qui ne peuvent pas être étudiés théoriquement parce que cela se situe en dehors de cas idéal des commande du système et c'est ce que nous avons vu pour l'intelligence artificiel et les langage de programmation, mais grâce à les recherches approfondies et à l'étude de toutes les possibilités,

nous avons pu proposer des solutions sans recourir à des dépenses supplémentaires et maintenir de bonnes performances.

Nous avons commencé par préparer les programmes des réseaux de neurones nécessaires au bon fonctionnement et qui sont compatible avec le traitement vocale, puis nous avons créé la carte électronique lié à la base de données. Nous avons également utilisé plusieurs services pour surmonter les obstacles de intelligence artificiel c'est-à-dire modifier la structuration des programme, puis nous avons terminé le travail en faisant fonctionner notre assistant a laide du langage python qui est laissant entre tout les blocs.

Le rapport conclut qu'il existe plusieurs systèmes de commande vocale ciblant les maisons intelligentes, qui génère des effets positifs et négatifs tels que décrits dans les résultats. De plus, le rapport déclare qu'une implantation d'un système de commande vocale peut avoir un effet potentiel à long terme lorsqu'il est combiné avec la théorie du bourrin.

Ce projet nous a permis d'une part d'améliorer nos connaissances scientifiques, que ce soit du côté de programmation et apprentissage de nouveaux langages, en l'occurrence Python, et C, ou du côté réalisation matérielle autour des systèmes à microprocesseur, dans notre cas, on a utilisé un Raspberry Pi3.

4.8 Perspectives

Comme perspectives nous envisageons à cité les tache suivantes :

- L'Assistant vocale peut être programmé pour n'importe quel surface d'application.
- Cette Réalisation nous a permet de comprendre le fonctionnement de l'IA qui est la base technologique actuelle.
- La Représentation du Système électronique qui porte de grand empêchement technique bloquant la continuation de ce projet.
- Le travaille de groupe et aussi l'arme la plus performante pour continuer facilement le travaille et apprendre beaucoup de chose supplémentaire et essentiel par contre le travaille solitaire est moins rentable et fatigant.

BIBLIOGRAPHIE

- [1] S. ABDELLI and M. E. h. METAHRI, "Smart house," in *Université ABOU BEKR BEL-KAID Tlemcen*, 2017.
- [2] A. Reha, H. Ounayn, M. Kellili, N. El Abdi, O. Ismaili, M. Satar, M. Bouchouirbat, and A. Goucheq, "Conception et réalisation d'une maison intelligente," in *Colloque sur les Objets et systèmes Connectés*, 2019.
- [3] D. Reffas, "Système domotique avec raspberry pi," *Mémoire de master, Université de Beskra*, juillet 2019.
- [4] S. T. "Simplyeasylearning", "Artificial intelligence intelligent systems.pdf," *Artificial intelligence*, 2019.
- [5] A. PETRICEVIC and V. TALLOIR, "L'intelligence artificielle : une solution à la prise de décision quotidienne," in *ESGI École supérieure de informatique*, April 2019.
- [6] E. Commission, "Tout savoir sur l'intelligence artificielle." <https://experiences.microsoft.fr/business/intelligence-artificielle-ia-business/comprendre-utiliser-intelligence-artificielle/>, 2021.
- [7] I. Zara, "L'intelligence artificielle principe, outils et objectifs.," in *Université Badji Mokhtar Annaba.*, 2019.
- [8] E. Commission, "Histoire de l'intelligence artificielle." <https://siecledigital.fr/2018/08/20/histoire-intelligence-artificielle/>, Dec 2020.

-
- [9] futura sciences, "intelligence artificielle." <https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/>, Dec 2020.
- [10] O. Ezratty, *Les usages de l'intelligence artificielle*. Olivier Ezratty, 2018.
- [11] B. Barraud, "L'intelligence de l'intelligence artificielle," 2019.
- [12] M. I. I. Amezzane, "L'intelligence artificielle dans les smart homes," 2015.
- [13] Y. Djeriri, "Les réseaux de neurones artificiels," *mémoire fin d'étude, Sidi Bel Abbès, Algérie*, 2017.
- [14] M. Abadi, *Réalisation d'un réseau de neurones "SOM" sur une architecture matérielle adaptable et extensible à base de réseaux sur puce "NoC"*. PhD thesis, Université de Lorraine ; Université du Centre (Sousse, Tunisie), 2018.
- [15] H. Mezaache, *Les réseaux de Neurones formels Et Les systèmes Neuro-Flous Pour l'apprentissage par renforcement*. PhD thesis, Université de Batna 2, 2008.
- [16] N. Merzouka, *Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réel : une approche dans l'espace d'état*. PhD thesis, 2018.
- [17] G. Dreyfus, J. Martinez, M. Samuelides, M. Gordon, F. Badran, S. Thiria, and L. Héroult, *Réseaux de neurones*, vol. 39. Eyrolles Paris, 2002.
- [18] H. Chaoui, *Conception et comparaison de lois de commande adaptative à base de réseaux de neurones pour une articulation flexible avec non-linéarité dure*. PhD thesis, Université du Québec à Trois-Rivières, 2002.
- [19] Y. Bendaoud, "Prédiction des résistances mécaniques des bétons à base des ciments composés en utilisant les réseaux neurones artificiels," 2014.
- [20] M. Y. Ammar, *Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu*. PhD thesis, 2007.
- [21] A. Jauffret, N. Cuperlier, P. Gaussier, and P. Tarroux, "From self-assessment to frustration, a small step toward autonomy in robotic navigation," *Frontiers in neurorobotics*, vol. 7, p. 16, 2013.

-
- [22] D. Ali, *Intitulé : Etude de la classification supervisée des données environnementales à l'aide de réseaux de neurones de fonctions à base radiales*. PhD thesis, UNIVERSITE DE MOHAMED BOUDIAF M'SILA FACULTE DE TECHNOLOGIE, 2016.
- [23] E. M. Brakni, *Réseaux de neurones artificiels appliqués à la méthode électromagnétique transitoire InfiniTEM*. PhD thesis, Université du Québec en Abitibi-Témiscamingue, 2011.
- [24] M. Mohamed, *Approximation du modèle géométrique inverse d'un robot manipulateur par les réseaux de neurones artificiels*. PhD thesis, UNIVERSITE MOHAMED BOUDIAF-M'SILA, 2019.
- [25] A. Berkani, *Métaheuristique Hybride Réseaux de Neurones Artificiels-PSO du Recuit Simulé pour la Commande d'un Procédé Industriel Non-linéaire*. PhD thesis, Université de Batna 2, 2013.
- [26] M. OULMI and S. KALOUNE, *Classification d'objets avec le Deep Learning*. PhD thesis, université Akli Mouhand Oulhadje-Bouira, 2018.
- [27] H. Abdelaziz, I. Omar, A. OUAHAB, *et al.*, *La classification des images satellitaires par l'apprentissage profonde (deep learning)*. PhD thesis, Université Ahmed Draïa-Adrar, 2019.
- [28] B. Bilal, "Deep learning basé sur les méthodes de réduction pour la reconnaissance de visage," *MÉMOIRE DE MASTER*, 2019.
- [29] M. Dinarelli and I. Tellier, "Étude de réseaux de neurones récurrents pour l'étiquetage de séquences," in *Traitement Automatique des Langues Naturelles (TALN)*, 2016.
- [30] M. Bouaziz, *Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles*. PhD thesis, Université d'Avignon, 2017.
- [31] H. Abdel Hakim and T. Mohamed, *Reconnaissance automatique de la parole arabe continu*. PhD thesis, Université Saad Dahleb Blida 1, 2019.
- [32] RAGHAV, "Speech recognition using hidden markov model and artificial neural networks," *Lovely Professional University Phagwara, Punjab India*, vol. 11614243, November 2017.

-
- [33] S. BENYAB and A. BENADLA, *Réalisation d'un détecteur de câble secteur encastré*. PhD thesis.
- [34] K. Jlassi, "Microprocesseurs et microcontrôleurs," 2008.
- [35] "Microcontrôleur." <http://for-ge.blogspot.com/2015/07/microcontrôleur.html>, Juin 2021.
- [36] M. Schmidt, *Raspberry Pi : a quick-start guide*. Pragmatic Bookshelf, 2014.
- [37] A. Ahmed and D. M. Nabil, "Développement d'une application de télé-opération d'un système chauffage commande par internet," *Master, AUTOMATIQUE ET SYSTEMS Université Dr.Tahar Moulay de Saïda*.
- [38] M. Maurya and S. R. Shukla, "Current wireless sensor nodes (motes) : Performance metrics and constraints," *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 2, no. 1, pp. 45–48, 2013.
- [39] Memsic, "Powerful sensing solutions,"
- [40] A. MOUSSAOUI, "Conception et réalisation d'un bras manipulateur commandé par arduino mega 2560," *Université de Boumerdes*, 2017.
- [41] S. Reyvanth and G. Shirish, "Pid controller using arduino,"
- [42] "Transistor." <http://for-ge.blogspot.com/2015/04/transistor.html>, Dec 2021.
- [43] AP, "Astuces pratique." <https://www.astuces-pratiques.fr/electronique/le-transistor-bc547>, 2021.
- [44] Xukyo, "aranacorp." <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/>, 2018.
- [45] Y. Derfoufi, "Programmation en langage python," 2019.
- [46] H. Khaled, *Elaboration d'un serveur web à l'aide de Raspberry*. PhD thesis, BADJI MOKHTAR–ANNABA UNIVERSITY, 2019.