

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Electronique



## Mémoire de Projet de Fin d'Études

Présenté par

Ouahiba CHERIR

Amal MEKDOUR

Pour l'obtention du diplôme de Master en Electronique

Spécialité : Electronique des Systèmes Embarqués

Thème

# Gestion intelligente des parkings par commandes vocales

Proposé par :

Promoteur :

Fayçal YKHLEF, Maître de Recherche A, CDTA, Alger.

Co-promoteur :

Farid YKHLEF, Professeur, Université Blida 1, Blida.

Année Universitaire 2022-2023

## REMERCIEMENTS

---

Nous débutons ce mémoire en exprimant notre gratitude envers Dieu Tout-Puissant pour son aide et son soutien dans la réalisation de ce. Sans lui, nous n'aurions jamais réussi.

Nous tenons à remercier tout d'abord Monsieur **YKHLEF Fayçal**, Maître de Recherche A au CDTA, qui nous a proposé ce thème, et accepté de superviser ce mémoire. Son rôle a été crucial dans la réussite de ce projet grâce à ses orientations et son assistance. Nous tenons également à exprimer notre gratitude envers **Dr. BOUCHAFFRA Djamel**, Directeur de Recherche, madame **BENZABA Wahiba**, Ingénieur de recherche conseillé, et madame **BOUTALEB Ratiba**, ingénieur de recherche, qui nous ont apporté une aide et des conseils précieux pendant notre séjour au CDTA.

Nous tenons particulièrement à remercier notre co-encadreur, M. **YKHLEF Farid**, Professeur à l'Université SAAD DAHLAB, qui a cru en nos capacités et a prédit le succès de ce projet. Nous remercierons chaleureusement les membres du jury, pour avoir consacré une partie de leur temps précieux pour discuter et évaluer notre travail.

## DEDICACE

---

Je dédie ce mémoire à mes **chers parents**

Pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

À mes frères « **IMAD** » et « **TARAK** » pour leur appui et leur encouragement,

À mes sœurs « **KHALIDA** » et « **FAHIMA** » pour leurs encouragements permanents, et leur soutien moral, et toute ma famille pour leur soutien tout au long de mon parcours universitaire.

À **monsieur Ismail**, magasinier de département électronique au USDB qui nous a beaucoup aidés et s'est donné du mal pour assurer le succès de réalisation de nos projets à l'université.

Ma binôme **Ouahiba** pour les bons moments passés ensemble, et à mes camarades qui m'ont aidé « **Abdelbaki, Amine, Walid, Sid Ahmed, Yacine, Karima** » Je leur souhaite beaucoup de succès dans leur vie.

A tous mes amis, et à l'ensemble des étudiants de la promotion de 2ème année Master électronique des systèmes embarqués 2022/2023.

**Amal**

## DEDICACE

---

Je dédie ce mémoire à mes chers parents

**Ma mère**, qui a œuvré pour ma réussite à travers son amour, son soutien, les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie.

**Mon père**, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Merci pour les nobles valeurs, l'éducation et le soutien permanent venant de toi.

**Mes frères et sœurs** pour leurs encouragements permanents, et leur soutien moral.

À **monsieur Ismail**, magasinier de département électronique au USDB qui nous a beaucoup aidés et s'est donné du mal pour assurer le succès de réalisation de nos projets à l'université. Et sans oublier **HAMMOUDA Rafik**, **BOUCHIBA Amine** et **BERKAN Lahcene Walid**, qui m'ont beaucoup aidé dans ma carrière universitaire.

Ma binôme **Amal** pour les bons moments passés ensemble

A mes meilleures amies qui ont été toujours à mes côtés surtout dans les moments difficiles. A tous ceux qui, par un mot, m'ont donné la force de continuer.

Et tous ceux qui j'aime et qui m'aiment.

A tous mes collègues de la promotion de génie électronique 2022/2023

**Ouahiba**

## RESUME

---

**ملخص الهدف** من هذا المشروع هو استخدام تقنية التعرف على الصوت من أجل جعل الإجراءات الرئيسية لوكلاء مواقف السيارات تشتغل بطريقة أوتوماتكية. النظام الذي نقترحه يتعرف في الوقت الحقيقي على الأوامر المنطوقة من وكيل نقطة التحكم ويقوم بتنفيذ المهام التالية: (١) فتح وإغلاق الحاجز الرئيسي، (٢) تشغيل وإيقاف الإضاءة تلقائيًا في موقف السيارات، و(٣) تشغيل وإيقاف الإنذار في حالة السرقة. تعتمد منهجية هذا المشروع على مرحلتين رئيسيتين: (١) كشف صوت تصفيق اليدين باستخدام تقنية التعرف على الأصوات المفاجئة، و (٢) التعرف على الأوامر المنطوقة باستخدام واجهة برمجة التطبيقات (API) من Google. تم تصميم البرمجيات باستخدام لغة Python. وتم استخدام أردوينو لتنفيذ الأوامر. تحصلنا على نتائج مشجعة.

**الكلمات الدالة:** التعرف على الصوت، الاصوات المندفعة، إدارة وقوف السيارات، بايثون، أردوينو

---

### Résumé :

L'objectif de ce projet est d'utiliser la technologie de la reconnaissance vocale pour automatiser les actions principales des agents de parking. Le système que nous proposons reconnaît en temps réel les commandes prononcées par l'agent du poste de contrôle et exécute les tâches suivantes : (i) ouverture et fermeture de la barrière principale, (ii) allumage et extinction automatique de l'éclairage dans le parc, et (iii) déclenchement et arrêt d'alarmes en cas de vol. La méthodologie adoptée dans ce projet repose sur deux phases principales : (i) la détection des claquements de mains en se basant sur une technique d'identification des sons impulsifs, et (ii) la reconnaissance des commandes vocales à l'aide de l'API (Application Programming Interface) Google. Le logiciel a été conçu en utilisant le langage Python. Arduino a été utilisé pour exécuter les commandes. Les résultats obtenus sont encourageants.

**Mots clés :** Reconnaissance vocale, sons impulsifs, gestion de parkings, Python, Arduino

---

**Abstract:** The objective of this project is to use voice recognition technology to automate the main actions of parking agents. The system we propose recognizes in real-time the commands pronounced by the control agent and performs the following tasks: (i) opening and closing of the main barrier, (ii) automatic light control in the parking area, and (iii) activation and deactivation of alarms in the event of theft. The methodology adopted in this project is based on two main phases: (i) detection of hand claps based on an impulsive sound identification technique, and (ii) recognition of voice commands using Google's Application Programming Interface (API). The software was designed using the Python language. Arduino was used to execute the commands. The results obtained are promising.

**Keywords:** Voice recognition, impulsive sounds, parking management, Python, Arduino.

---

## LISTE DES ACRONYMES ET ABREVIATIONS

---

**ASR:** *Automatic Speech Recognition*

**CNN:** *Convolutional Neural Network*

**DTW:** *Dynamic Time Warping*

**HMMs:** *Hidden Markov Models*

**IR:** *Infra Rouge*

**IA:** *Intelligence Artificielle*

**IOT:** *Internet of Things*

**LCD:** *Liquid Crystal Display*

**LPR:** *License Plate Recognition*

**RNN:** *Recurrent Neural Networks*

**SR:** *Speech Recognition*

## LISTE DES FIGURES

---

<b>Figure 2.1:</b> Parking intelligent [2] .....	3
<b>Figure 2.2:</b> Capteurs d'identification des places disponibles [3] .....	4
<b>Figure 2.3:</b> Ancien stationnement [3] .....	6
<b>Figure 2.4:</b> Parking surface .....	7
<b>Figure 2.5:</b> Parking souterrain [6] .....	7
<b>Figure 2.6:</b> Parking à étages [6] .....	8
<b>Figure 2.7:</b> Parking souterrain public .....	9
<b>Figure 2.8:</b> Parking surface privé .....	9
<b>Figure 3.1:</b> Méthode globale de la gestion intelligente des parkings par commandes vocales .....	14
<b>Figure 3.2</b> Etapes de détection des sons impulsif .....	16
<b>Figure 3.3:</b> Tâches à exécuter .....	21
<b>Figure 3.4:</b> Schéma de connexion de la barrière, du module relais et de l'Arduino ....	22
<b>Figure 3.5:</b> Schéma de connexion du buzzer et de l'Arduino .....	22
<b>Figure 3.6:</b> Schéma de connexion de la lampe, du transistor et de l'Arduino .....	23
<b>Figure 4.1:</b> Jupyter Notebook .....	25
<b>Figure 4.2:</b> (a) Carte Arduino Uno. (b) Interface d'utilisateur de l'Arduino .....	25
<b>Figure 4.3:</b> Wo Mic Client .....	26
<b>Figure 4.4:</b> Wo Mic .....	26
<b>Figure 4.5:</b> Option de Wo Mic client .....	27
<b>Figure 4.6:</b> Option de Wo Mic .....	27
<b>Figure 4.7:</b> Application Lesser AudioSwitch .....	28
<b>Figure 4.8:</b> Qt Designer .....	29
<b>Figure 4.9:</b> Conversion « ui » vers « py » .....	29
<b>Figure 4.10:</b> Exemple de « firmata » .....	30
<b>Figure 4.11:</b> Téléversement de programme « firmata » .....	31
<b>Figure 4.12:</b> Interface de la gestion intelligente par commande vocale .....	32
<b>Figure 4.13:</b> Liste des choix de microphones .....	32
<b>Figure 4.14:</b> Choix des microphones .....	33

<b>Figure 4.15:</b> Affichage de variance .....	33
<b>Figure 4.16:</b> Configuration du seuil de la variance .....	34
<b>Figure 4.17:</b> Liste des tâches.....	34
<b>Figure 4.18:</b> (a) Bouton de démarrage (b) Affichage.....	34
<b>Figure 4.19:</b> Affichage de Arduino .....	35
<b>Figure 4.20:</b> Graphe du seuil de variance .....	35
<b>Figure 4.21:</b> Message généré suite à la détection d'un son impulsif .....	35
<b>Figure 4.22:</b> (a) Message affiché. (b) Modification de la couleur du cercle de la barrière. ....	36
<b>Figure 4.23:</b> Action d'ouverture de la barrière.....	36
<b>Figure 4.24:</b> (a) Affichage du message de la barrière. (b) Changement de la couleur du cercle.....	37
<b>Figure 4.25:</b> Fermeture de la barrière .....	37
<b>Figure 4.26:</b> (a) Affichage du message de l'alarme. (b) Changement la couleur du cercle de l'alarme.....	37
<b>Figure 4.27:</b> Montage électronique incluent le buzzer Commettant une alarme .....	38
<b>Figure 4.28:</b> (a) Affichage du message d'alarme. (b) Désactivation de l'alarme.....	38
<b>Figure 4.29:</b> (a) Affichage du message d'éclairage. (b) Changement la couleur du cercle.....	38
<b>Figure 4.30:</b> Activation de l'éclairage .....	39
<b>Figure 4.31:</b> (a) Affichage du message relatif à l'éclairage. (b) Changement de la couleur du cercle de l'éclairage .....	39
<b>Figure 4.32:</b> Extinction de l'éclairage.....	39
<b>Figure 4.33:</b> (a) Affichage de l'exécution de la commande. (b) Changement de toutes les couleurs .....	40
<b>Figure 4.34:</b> Exécution de toutes les tâches .....	40
<b>Figure 4.35:</b> (a) Affichage de l'arrêt. (b) Changement de toutes les couleurs .....	41
<b>Figure 4.36:</b> Arrêter toutes les tâches .....	41
<b>Figure 4.37:</b> (a) Message correspondant à un problème de la reconnaissance vocale. (b) réécriture du mot incorrect.....	41

<b>Figure 4.38:</b> Affichage du message correspondant à un problème de déconnexion du microphone.....	42
<b>Figure 4.39:</b> Affichage du message correspondant à problème de connexion avec google speech recognition.....	42
<b>Figure 4.40:</b> (a) Bouton d'arrêt. (b) Affichage du message d'arrêt .....	43
<b>Figure 4.41:</b> Interface de la gestion intelligente du parking en langue Anglaise .....	43
<b>Figure 4.42:</b> Interface de la gestion intelligente en langue Française.....	44

## LISTE DES TABLEAUX

---

<b>Tableau 4-1 : Évaluation des personnes. ....</b>	<b>45</b>
---	-----------

# TABLE DES MATIERES

---

Remerciements	(i)
Dédicaces	(ii)
Résumé	(iii)
Liste d'abréviations et acronymes	(iv)
Liste des figures	(v)
Liste des tableaux	(vi)
Table des matières	(vii)

## **CHAPITRE 1 : INTRODUCTION..... 1**

1.1 Motivations .....	1
1.2 Contributions .....	1
1.3 Impact de projet .....	2
1.4 Organisation du mémoire .....	2

## **CHAPITRE 2 : GENERALITES SUR LA GESTION INTELLIGENTE DES PARCS DE STATIONNEMENT.....3**

2.1 Introduction .....	3
2.2 Gestion intelligente des lieux de stationnement.....	3
2.3 Parcs de stationnement à travers les âges .....	5
2.4 Type de parkings .....	6
2.5 Etat d'art sur la gestion intelligente des parkings .....	10
2.6 Conclusion.....	12

## **CHAPITRE 3 : SOLUTION PROPOSEE POUR LA GESTION DES PARKINGS ..... 13**

3.1 Introduction .....	13
3.2 Méthodologie globale de gestion .....	13
3.3 Conclusion.....	23

## **CHAPITRE 4 : RESULTATS ET INTERPRETATIONS ..... 24**

4.1 Introduction .....	24
4.2 Logiciels et langages de programmation utilisés.....	24
4.3 Connexion entre Python et Arduino .....	29
4.4 Implémentation de la solution proposée .....	31
4.5 Evaluation du système .....	44
4.6 Conclusion.....	45

<b>CHAPITRE 5 : CONCLUSION ET TRAVAUX FUTURES.....</b>	<b>47</b>
4.7 Conclusion.....	47
4.8 Travaux futurs .....	48

# Chapitre 1 : Introduction

## 1.1 Motivations

L'élaboration d'une application de gestion intelligente des parcs de stationnement offre l'avantage de libérer l'agent de contrôle de certaines tâches, lui permettant ainsi de se concentrer sur des responsabilités essentielles. Grâce à cette solution, l'agent peut consacrer son temps et ses efforts à d'autres aspects importants de la gestion du parking, améliorant ainsi l'efficacité globale du parc.

L'intelligence artificielle (IA), l'internet des objets (IO) et l'électronique des systèmes embarqués sont des éléments indispensables au développement de telles applications.

L'IA joue un rôle crucial dans la conception de systèmes de gestion intelligents des parkings. Elle permet l'analyse et l'interprétation des données en temps réel, l'optimisation de l'utilisation de l'espace de stationnement, et l'amélioration de la gestion du parc.

A titre d'exemple, la technologie de reconnaissance des commandes vocales permet à l'agent de contrôle d'interagir avec l'ordinateur de gestion des parcs de stationnement et d'automatiser l'exécution des tâches et actions. Cette exécution doit se faire en temps réel et être compatible avec un ordinateur de moyennes performances.

De plus, la sécurité du parking est renforcée grâce à la possibilité de déclencher des alarmes en cas d'urgence ou de vol par le biais de commandes vocales. Cela dissuade efficacement les actes criminels et contribue à prévenir toute tentative de vol ou d'intrusion. De plus, en éliminant la nécessité pour l'agent de contrôle de quitter sa position ou d'effectuer des actions manuelles, le risque d'accidents ou de blessures est considérablement réduit, assurant ainsi un environnement plus sûr pour les utilisateurs du parking.

## 1.2 Contributions

L'objectif de ce projet est d'utiliser les technologies de la reconnaissance de la voix en vue d'automatiser les actions principales des agents des parkings. Le système que nous proposons reconnaît en temps réel les commandes de l'agent du poste de contrôle et exécute les tâches suivantes : (i) ouverture et fermeture de la barrière principale, (ii) gestion de l'éclairage du parc, et (iii) déclenchement des alarmes en cas de vol. Le

système fonctionne en temps réel et permet à l'agent de communiquer avec l'ordinateur principal du parc et exécute des actions.

Nous avons développé un logiciel de détection des claquements de mains et de reconnaissance des commandes vocales. La détection des claquements de main permet à l'agent de contrôler d'initier une interaction avec l'ordinateur principal du parc en lui donnant un signal d'action. Ensuite, les commandes vocales sont identifiées et traduites en actions correspondantes.

La détection des claquements de mains se base sur une technique d'identification des sons impulsifs. Cependant, la reconnaissance des commandes vocales est effectuée à l'aide de la reconnaissance vocale de Google. Le circuit Arduino est utilisé pour exécuter les commandes.

### **1.3 Impact de projet**

Le travail présenté dans ce projet de fin d'étude revêt une grande importance pour le développement des technologies industrielles dans notre pays. Il propose un module essentiel pour la gestion intelligente des parkings, ce qui a un fort impact socio-économique en Algérie. Plusieurs clients potentiels, tels que les supermarchés, les banques, les sociétés, les universités, les écoles, les lycées et les ambassades, pourraient être intéressés par notre solution. De plus, ce système peut être facilement intégré dans un projet de cité intelligente (Smart City).

### **1.4 Organisation du mémoire**

**Le mémoire est organisé comme suit :**

**Le deuxième chapitre :** présent des généralités sur la gestion intelligente des parcs de stationnement. Un état de l'art sur la gestion intelligente des parkings est aussi fourni dans ce chapitre.

**Le troisième chapitre :** expose un aperçu sur la solution proposée.

**Le quatrième chapitre :** se concentre sur la présentation des résultats expérimentaux.

**Le cinquième chapitre :** comprend une conclusion et des perspectives.

# Chapitre 2 : Généralités sur la gestion intelligente des parcs de stationnement

## 2.1 Introduction

Dans ce chapitre nous présentons des généralités sur la gestion intelligente des lieux de stationnement. Nous présentons un aperçu général sur l'évolution des parcs de stationnement à travers les âges. Nous abordons également les différentes catégories de parking. Nous mentionnons quelques travaux de recherche sur les méthodes de gestion intelligente des parkings. Nous terminons ce chapitre par une conclusion.

## 2.2 Gestion intelligente des lieux de stationnement

La conception des parcs de stationnement intelligents consiste à utiliser une multitude de technologies en vue d'une gestion efficace des parkings. L'IA, les capteurs, et les caméras de surveillance sont des exemples de telles technologies [1].

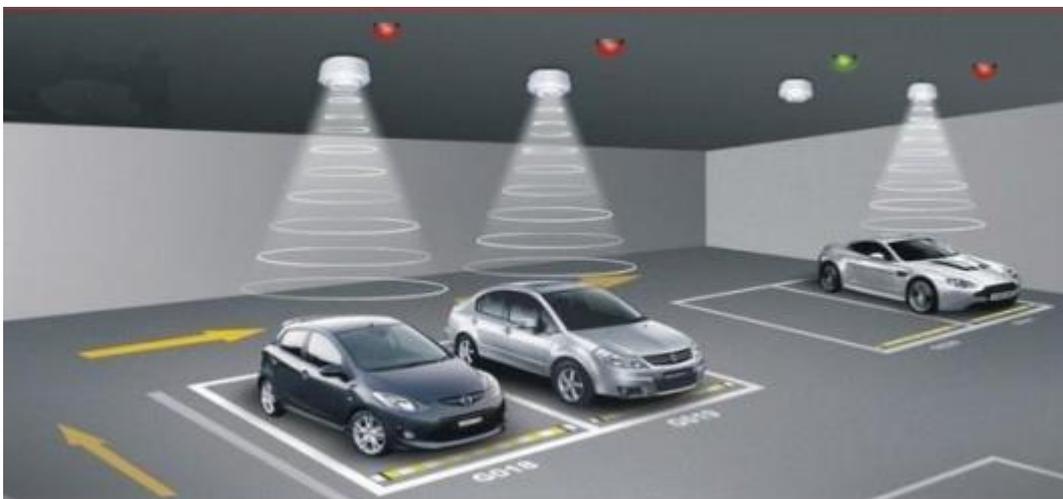
Un exemple d'utilisation de ces technologies pour automatiser les parcs de stationnement est illustré sur la Figure 2.1. Les capteurs sont installés sur la route, au centre de chaque emplacement, afin de détecter les places de stationnement disponibles et occupées [2].



Figure 2.1: Parking intelligent [2]

Ce que nous entendons par gestion efficace, c'est la capacité de gérer les processus (les étapes nécessaires pour accomplir un projet) ou les activités (les actions spécifiques nécessaires pour exécuter les processus) afin d'atteindre les meilleurs résultats possibles, tout en réduisant les pertes. Cela implique l'utilisation appropriée de technologies de pointes telles que l'IOT et l'IA. Par exemple, nous pouvons citer le développement d'applications en ligne qui facilitent la recherche et l'information sur les places de stationnement disponibles à proximité pour les conducteurs. Cela permet d'éviter de perdre du temps à chercher une place de stationnement, de réduire la congestion routière et d'offrir plus de confort aux conducteurs [1].

L'IA peut également être utilisée à l'intérieur des parkings pour assurer une gestion efficace. Nous mentionnons les techniques de reconnaissance des plaques d'immatriculation qui contribuent à garantir que seules les personnes autorisées utilisent les places de stationnement, ainsi que l'ouverture et la fermeture automatique des barrières. Des caméras de sécurité peuvent également être installées dans les parkings pour assurer la sécurité des véhicules et des utilisateurs, ainsi que pour détecter tout comportement suspect [1]. De même, des capteurs de position peuvent être utilisés pour identifier les places occupées et disponibles (Figure 2.2).



**Figure 2.2:** Capteurs d'identification des places disponibles [3]

La gestion efficace permet d'offrir une meilleure assistance aux agents de sécurité afin qu'ils puissent accomplir leurs tâches de manière plus efficace, notamment en assurant la sécurité générale du parking.

## 2.3 Parcs de stationnement à travers les âges

Au fil des siècles, la gestion des parkings a connu une évolution significative afin de répondre aux besoins changeants des communautés et des propriétaires de véhicules. Initialement, les propriétaires de voitures se garaient dans des zones de stationnement désignées ou des espaces informels dans la rue. Cependant, avec la croissance exponentielle de la circulation automobile, les problèmes de congestion et de sécurité routière ont augmenté, exigeant une réglementation de la gestion des parkings.

Il y a bien longtemps, lorsque les gens ont commencé à se déplacer en utilisant autre chose que leurs pieds, ils ont ressenti le besoin de disposer d'un espace spécifique pour stocker leurs moyens de transport. L'exemple le plus rudimentaire de stationnement ancien remonte à l'utilisation d'un rail par les cavaliers pour relier leurs chevaux fatigués. Avec le temps, l'émergence des voitures a nécessité davantage d'espace pour leur stockage. Cela a conduit à la création d'écuries, l'équivalent médiéval des garages et des parkings, où les véhicules pourraient être protégés et abrités. Pour cette raison, les places de stationnement spécialement conçues pour les automobiles ont commencé à apparaître au début du 20<sup>ème</sup> siècle, tel que des parkings (en surface, à étages et souterrain) [3] (Figure 2.3).

La City and **Suburban Electric Carriage Company** [4], fabricant de véhicules électriques, a introduit le premier parking fermé à plusieurs étages à Londres. En 1901, l'entreprise a érigé un bâtiment de sept étages pouvant accueillir 100 voitures. Par la suite, deux autres parkings ont été construits, offrant respectivement 230 et 200 places de stationnement. De plus, le premier parking souterrain a été édifié à Barcelone sous la Casa Mila, une création de l'architecte **Antoni Gaudi** [5] .

Il s'agissait de la première tentative de mécanisation du processus de stationnement. Les voitures étaient élevées aux niveaux supérieurs à l'aide d'ascenseurs de service, Cependant, des voituriers étaient utilisés pour garer une voiture dans la bonne place de stationnement [3].



**Figure 2.3:** Ancien stationnement [3]

Les avancées technologiques ont permis de développer des systèmes de gestion de stationnement de plus en plus sophistiqués et automatisés, tels que les applications de stationnement mobiles et les systèmes de stationnement intelligents, pour offrir une meilleure expérience de stationnement aux conducteurs. Ainsi, la gestion des parkings est devenue une industrie importante dans de nombreuses villes, avec de nombreuses entreprises spécialisées dans la fourniture de services de stationnement et de gestion de parkings [1].

## **2.4 Type de parkings**

Les parkings occupent une place primordiale au sein de nos villes en fournissant des espaces de stationnement pratiques et sécurisés pour les véhicules. Ils jouent un rôle essentiel en facilitant la mobilité et en évitant les problèmes associés au stationnement. Il existe une grande variété de types de parkings, chacun étant conçu pour répondre à des besoins spécifiques et proposant des fonctionnalités différentes.

### **2.4.1 Parking en surface**

Le parking de surface est un type de stationnement qui se trouve à la surface du sol, sans étages supplémentaires. Ces parkings sont généralement situés en bordure de la route ou dans des zones spécialement aménagées pour le stationnement public, souvent dans des zones densément peuplées [6] (voir Figure 2.4).



**Figure 2.4:** Parking surface

### 2.4.2 Parking souterrain

Le stationnement souterrain se situe sous la surface du sol, généralement sous les bâtiments ou les espaces publics. Des rampes sont utilisées pour se déplacer entre les différents niveaux. La présence d'ascenseurs ou d'escaliers permet aux conducteurs de se déplacer entre les niveaux du parking. Ce type de parking est largement utilisé dans les zones urbaines où l'espace au sol est limité [6].

La figure 2.5 montre un niveau parmi les niveaux du parking souterrain.



**Figure 2.5:** Parking souterrain [6]

### 2.4.3 Parking à étages

Les parkings à étages sont des structures de stationnement à plusieurs niveaux conçus pour stationnement des véhicules. Ils fonctionnent de manière similaire aux parkings souterrains, mais sans nécessiter de travaux de creusement importants. Ils sont équipés de rampes permettant aux conducteurs de monter ou descendre les étages en

voiture, ainsi que d'ascenseurs et d'escaliers pour les piétons [6]. La figure 2.6 montre un parking à étages :



**Figure 2.6:** Parking à étages [6]

Tous ces types de parkings peuvent être classés en deux catégories principales : les parkings publics et les parkings privés. Dans les deux cas, leur objectif est de fournir des espaces de stationnement pour les véhicules. Les parkings, qu'ils soient souterrains, à plusieurs niveaux ou en surface, peuvent varier en taille. Certains parkings peuvent également offrir des services supplémentaires tels que la vidéosurveillance, l'éclairage et d'autres technologies. Il est important de respecter les règles spécifiques à chaque parking, comme la durée maximale de stationnement et les tarifs applicables.

Découvrons maintenant ces deux catégories principales.

- **Parking public**

Les parkings publics sont des emplacements désignés pour le stationnement des véhicules, accessibles à tous les usagers de la route. Ils sont gérés par des autorités publiques telles que les municipalités. Ils peuvent être trouvés dans des zones résidentielles, des centres commerciaux, des parcs, ainsi que les lieux touristiques ou dans d'autres espaces. Il existe également des parkings privés accessibles au public, tels que ceux des supermarchés, même s'ils sont situés sur des terrains privés. Dans ces cas, n'importe qui est autorisé à y garer sa voiture [7]. La figure 2.7 montre un exemple sur parking souterrain public



**Figure 2.7:** Parking souterrain public

- **Parking privé**

Il existe des parkings privés qui appartiennent à des personnes, des entreprises ou d'autres entités privées. L'accès à ces parkings est limité aux utilisateurs spécifiques et n'est pas ouvert au grand public. Ces parkings requièrent généralement une autorisation spéciale pour accéder et offrent souvent un niveau de sécurité plus élevé que les parkings publics [7].

À titre d'exemple des parkings, nous présentons la figure 2.8 qui montre un parking surface privé.



**Figure 2.8:** Parking surface privé

## 2.5 Etat d'art sur la gestion intelligente des parkings

Dans cette section, nous présentons un état de l'art sur les méthodes de gestion des parcs de stationnement dans la littérature. Nous mettons en avant les solutions proposées pour améliorer la gestion des parkings en utilisant l'IA et l'IoT.

**Shailja Sharma et al** [8] ont proposé un système de gestion du stationnement pour améliorer la recherche des places de stationnement disponibles. Il s'agit d'une application mobile qui utilise la reconnaissance vocale. Ce système comprend des capteurs infrarouges (IR) pour connaître le nombre de places de parking disponibles en temps réel. L'application marquera "vrai" sur les places occupées par des voitures et marquera "faux" sur les places vides. Lorsque le conducteur enregistre une réservation en fournissant des informations à l'aide de sa voix, le système analyse les données vocales pour extraire les informations. Une fois la réservation terminée, le conducteur effectue le paiement en ligne, et l'application considère la place comme réservée (en le marquant "vrai"). La période de réservation dure deux heures, et si le conducteur ne se présente pas pendant ce temps, sa demande sera annulée et il perdra 80% du montant payé.

**Toshihiro KOJIMA** [9] a conçu un système de gestion du stationnement qui facilite la recherche des véhicules garés et la détection des véhicules volés. Le système utilise la reconnaissance vocale pour améliorer le confort de l'utilisateur. Lorsqu'une voiture entre dans le parking, une caméra capture l'image de sa plaque d'immatriculation, qui est analysée pour extraire le numéro d'immatriculation. Ces informations sont ensuite stockées. Le conducteur choisit un emplacement de stationnement, détecté par un capteur, et la caméra de l'intérieur du parking capture une image de la plaque d'immatriculation. L'emplacement et les informations de la plaque d'immatriculation sont enregistrés. Lorsque le conducteur souhaite récupérer sa voiture, il prononce le numéro de plaque d'immatriculation, qui est analysé pour vérification et affichage de l'emplacement sur un écran. La serrure de stationnement est ouverte si le numéro correspond, permettant au conducteur de récupérer sa voiture. Le système permet également la détection des voitures volées en comparant les informations de plaque d'immatriculation avec une liste réservée aux voitures volées. Si une correspondance est trouvée, la serrure reste verrouillée et la voix de l'utilisateur est enregistrée, fournissant des preuves pour les enquêtes criminelles.

**Amara Aditya et all** [10] ont proposé un système de stationnement intelligent pour les smart cities, qui utilise la technologie IoT. Ce système est basé sur le développement d'une application mobile permettant aux utilisateurs de vérifier la disponibilité des places de stationnement dans un parking et de réserver une place de stationnement. Il utilise des capteurs IR pour détecter la présence de véhicules dans les places de stationnement, ainsi que la technologie RFID (Radio Frequency Identification) pour identifier les véhicules qui ont réservé une place de stationnement. Le système utilise ces dispositifs pour collecter en temps réel des données sur la disponibilité des places de stationnement dans un parking. Ces données sont ensuite envoyées vers le cloud pour être analysées, afin de proposer un emplacement de stationnement approprié à l'utilisateur.

Lors de l'évaluation de leur travail, ils ont affirmé que le système de stationnement intelligent proposé est précis et efficace, avec un taux de reconnaissance de 91,25%.

**M. Venkata Sudhakar et all** [11] ont développé une application mobile permettant de détecter les places de stationnement disponibles dans un parking intelligent. L'application mobile fournit également des alertes en cas d'incendie ou de fuite de gaz. Un écran LCD est installé à l'entrée du parking pour afficher le nombre de places disponibles. Des capteurs infrarouges sont utilisés pour détecter la présence d'un véhicule à l'entrée du parc de stationnement. Un système de reconnaissance automatique de plaques d'immatriculation License Plate Recognition « LPR » est utilisé pour identifier le numéro d'immatriculation du véhicule et enregistrer la date et l'heure de son accès.

Les résultats présentés dans l'article montrent que le système fonctionne efficacement dans différentes situations.

**Singh et all**, [12] ont introduit un système de parking automatisé qui utilise la technologie Bluetooth comme moyen d'accès. Ce système de parking automatisé repose sur un mécanisme qui permet de garer et de récupérer les voitures sans intervention humaine. Lorsqu'un utilisateur souhaite stationner son véhicule, le système utilise un lecteur Bluetooth pour récupérer l'identifiant Bluetooth de l'utilisateur, et cet identifiant est ensuite stocké dans une base de données. Le microcontrôleur Advanced RISC Machine « ARM » compare l'identifiant récupéré avec les numéros stockés dans la base de données. Si l'identifiant correspond à un nouveau véhicule, le processus de stationnement est activé, et le véhicule est automatiquement déplacé vers un emplacement de stationnement disponible. Si l'identifiant correspond à un véhicule déjà

garé, cela indique que l'utilisateur souhaite récupérer son véhicule. Dans ce cas, le processus de récupération est déclenché, et le système se charge de déplacer automatiquement le véhicule vers une zone de récupération.

**P. DharmaReddy et al**, [13] ont conçu un système intelligent pour améliorer la gestion du stationnement. Le système fournit des informations en temps réel sur les places de stationnement disponibles. Ces informations sont accessibles via le site web ou l'application mobile associée au système.

Ils ont mis en place un système qui utilise une caméra pour détecter les emplacements libres dans le parking. Les images capturées par la caméra sont traitées à l'aide d'une technique de traitement d'images, ce qui permet au système de déterminer quels emplacements sont occupés par des voitures et lesquels sont disponibles. Les informations sur les emplacements disponibles sont affichées sur un écran tactile LCD. Lorsqu'un emplacement est occupé, l'écran LCD affiche une case de couleur rouge pour indiquer qu'il est pris ; et lorsque l'emplacement est libre, le numéro correspondant est affiché sur l'écran LCD.

Les utilisateurs peuvent réserver un emplacement de stationnement en envoyant un Short Message Service « SMS » avec le numéro d'emplacement de leur choix. Lorsque le contrôleur, géré par le système, reçoit cette information via le réseau Global System for Mobile Communications « GSM », il marque l'emplacement comme réservé et le garde libre jusqu'à ce que le véhicule réservant arrive. L'évaluation montre que le système a été conçu et testé avec succès.

## **2.6 Conclusion**

Dans ce chapitre, nous avons présenté une vue globale sur la gestion intelligente des lieux de stationnement. Nous avons présenté les parcs de stationnement à travers les âges. Nous avons également présenté les différents types de parking. Nous nous sommes focalisés sur l'état d'art de la gestion intelligente des parkings. Nous avons constaté qu'il existe différents systèmes de gestion des parkings, mais nous n'avons pas trouvé beaucoup de systèmes qui ont assisté les agents de parking. C'est pourquoi nous souhaitons concevoir un système de gestion de stationnement intelligent qui fonctionne à l'aide de commandes vocales prononcées par l'agent.

# Chapitre 3 : Solution proposée pour la gestion des parkings

## 3.1 Introduction

Dans ce chapitre, nous présentons notre méthodologie pour la gestion globale. Nous aborderons la solution proposée pour la reconnaissance des commandes vocales et l'exécution des tâches spécifiques. Enfin, nous concluons ce chapitre par une conclusion.

## 3.2 Méthodologie globale de gestion

Dans cette section, nous présentons notre méthodologie qui est subdivisée en trois points : (i) un aperçu général, (ii) la partie logicielle et (iii) la partie matérielle.

### 3.2.1 Aperçu général

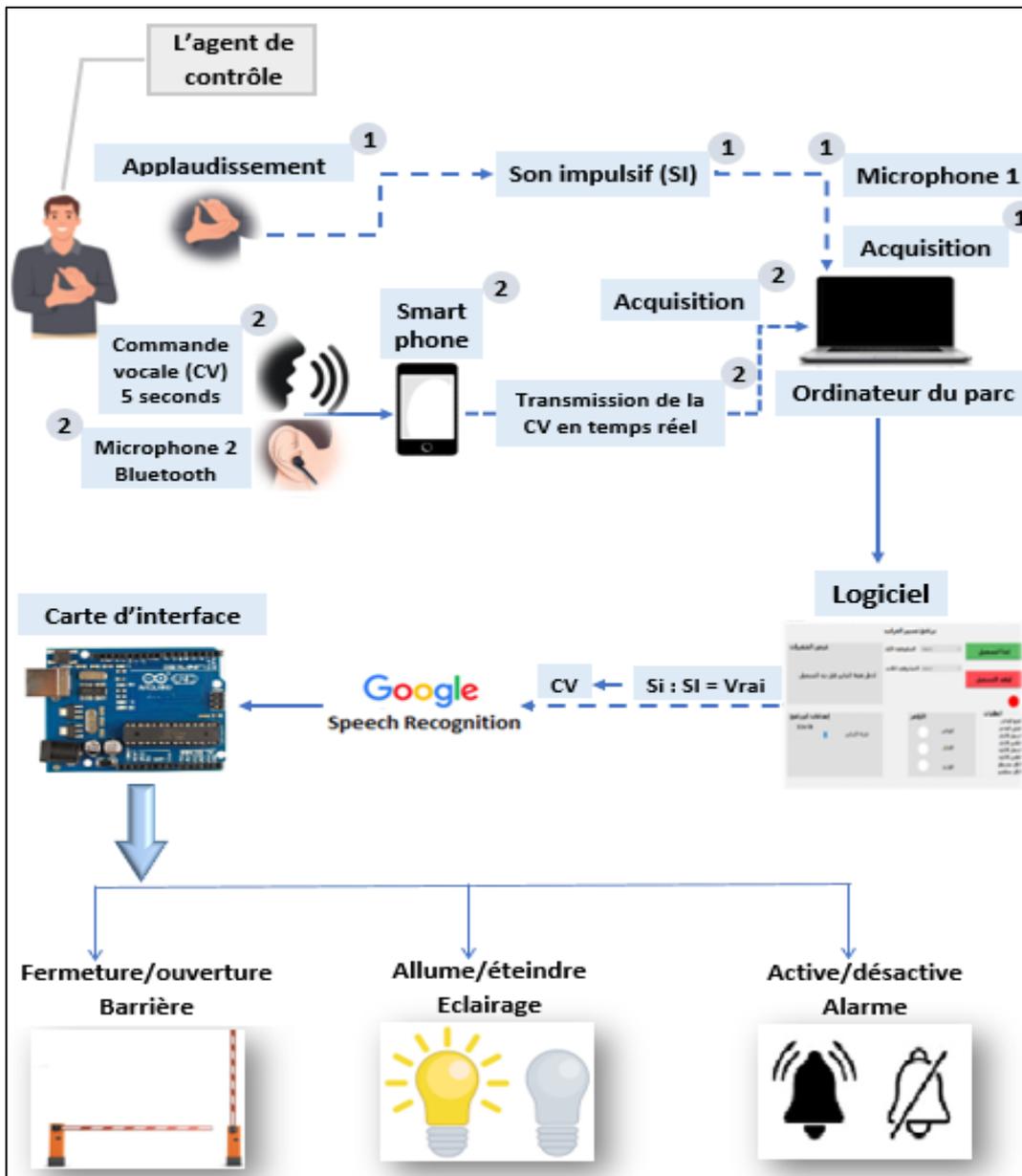
Les actions qu'un agent peut exécuter pour gérer le parking comprennent : (i) l'ouverture de la barrière, (ii) la fermeture de la barrière, (iii) l'allumage de l'éclairage, (iv) l'extinction de l'éclairage, (v) l'activation de l'alarme, (vi) la désactivation de l'alarme. Toutes ces actions peuvent être exécutées ou arrêtées à la fois.

Nous utilisons deux microphones dans notre système. Le premier est fixe et enregistre les sons de l'environnement d'une manière permanente afin de détecter d'éventuels événements impulsifs. Le deuxième microphone est placé près de l'oreille de l'agent pour enregistrer ses commandes. Ce microphone est mobile et peut être déplacé rapidement vers différents endroits. Il est important de maintenir une distance appropriée entre le deuxième microphone et les lèvres de l'agent pour assurer une bonne reconnaissance.

Lorsqu'un son impulsif tel que les claquements des mains est capté par l'algorithme de détection à travers le microphone 1 de l'ordinateur (**étape 1 sur la Figure 3.1**), le programme affiche "Je vous entends" et le deuxième microphone est activé. L'agent prononce la commande vocale désirée et l'API Google Speech Recognition convertie la voix en texte. Une comparaison est effectuée entre la commande vocale et les actions disponibles dans le programme afin d'exécuter la commande souhaitée. Cette dernière est envoyée à la carte d'interface (Arduino) entre le PC et le matériel (la lampe, la Barrière et le buzzer), qui se charge d'exécuter les actions requises.

Par exemple, si la commande vocale est "Ouvre la barrière", la barrière s'ouvrira après quelques secondes.

La figure (3.1) représente la méthode globale de la gestion intelligente des parkings par commandes vocales.



**Figure 3.1:** Méthode globale de la gestion intelligente des parkings par commandes vocales

### 3.2.2 Partie logicielle

La partie logicielle est divisée en deux étapes : (i) la détection des sons impulsifs, et (ii) la reconnaissance des commandes vocales.

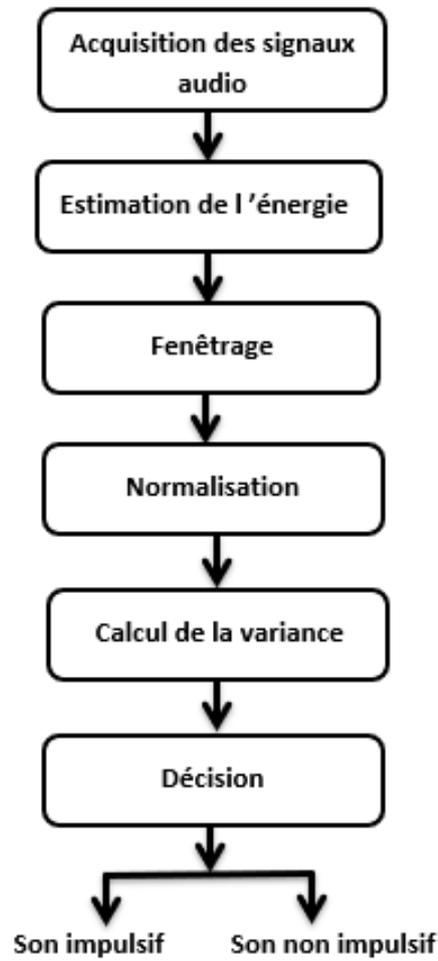
### 3.2.2.1 Détection des sons impulsifs

La détection des sons impulsifs consiste à identifier avec précision le moment où il se produit un changement brusque dans la pression acoustique du son. Dans la littérature, différentes approches sont décrites pour détecter les sons impulsifs, notamment la détection par seuillage et la détection par classification [14].

La détection par classification est une méthode largement utilisée pour identifier les événements sonores. Cette approche repose sur l'utilisation des techniques d'apprentissage afin de déterminer la présence ou l'absence d'un événement sonore spécifique. Cependant, cette méthode peut être sujette à des fausses alertes et présente une complexité algorithmique élevée. Plusieurs méthodes de reconnaissance peuvent être employées, notamment les Gaussian Mixture Models (GMMs), les Hidden Markov Models (HMMs) et les Support Vector Machines (SVMs). A. Rabaoui [14] a proposé une approche basée sur des fonctions exponentielles combinées à un modèle SVM à une seule classe pour estimer le rapport de vraisemblance généralisé. Cette solution permet d'effectuer la classification des sons en utilisant également les SVMs.

La détection par seuillage est une méthode qui compare une caractéristique du signal audio à un seuil pour déterminer la présence d'un événement sonore. Cette approche est moins complexe que la détection par classification. A. Dufaux a proposé une méthode de détection des sons impulsifs basée sur la variance des séquences d'énergie [14]. Cette méthode est conçue pour fonctionner en temps réel et dans des environnements bruités. La méthode de détection des sons impulsionnels que nous avons utilisée dans notre travail, repose sur la détection en temps réel des événements acoustiques impulsifs en utilisant l'évolution de la variance des séquences d'énergies à court terme du signal audio. Cependant, cette méthode doit relever deux contraintes majeures, le choix des paramètres de la méthode et le niveau de pression acoustique capté par le microphone, qui affecte la qualité de l'onde acoustique enregistrée [15].

Pour résoudre ces contraintes, il est important de sélectionner des paramètres algorithmiques appropriés en fonction de la plateforme d'implémentation et de régler la sensibilité du capteur pour obtenir de bonnes performances de détection. Ces paramètres doivent également être ajustés pour obtenir une efficacité de détection optimale. Les différentes étapes de la méthode de détection des sons impulsifs sont illustrées dans la figure 3.2.



**Figure 3.2** Etapes de détection des sons impulsif

- **Acquisition des signaux audio**

L'acquisition des signaux audio en temps réel fait référence à l'enregistrement immédiat des signaux sonores au fur et à mesure de leur occurrence.

- **Estimation de l'énergie**

L'estimation de l'énergie d'un signal numérique est une méthode qui permet de calculer la quantité d'énergie contenue dans un signal audio échantillonné à une fréquence donnée. Cette estimation est obtenue en calculant la somme des carrés des échantillons du signal sur une période donnée. Cette méthode est utilisée dans la détection des sons impulsifs pour détecter les événements acoustiques impulsifs en temps réel. La formule de calcul de l'estimation de l'énergie est :

$$e(k) = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n + kN) \quad (3.1)$$

$x(n)$  : un signal audio échantillonné à la fréquence  $F_e$

$k$  : est l'indice du bloc ( $k = 0, 1, \dots \dots \dots +\infty$ )

$n$  : est l'indice d'échantillon

$e(k)$  : est l'énergie du  $k$ ème bloc calculée à partir du signal  $x(n)$

$N$  : Indique la longueur du  $k$ ème bloc

○ **Fenêtrage de la séquence d'énergie**

C'est une phase importante de la méthode de détection des événements impulsifs. La séquence d'énergie est obtenue en estimant l'énergie d'un signal audio à l'aide de l'équation (3.1). Le fenêtrage de la séquence d'énergie consiste à créer une séquence de puissance  $e_{win}$  de longueur  $L$ . La formule de calcul du fenêtrage de la séquence d'énergie est :

$$e_{win}(j) = e(i) \quad (3.2)$$

Les indices «  $i$  » et «  $j$  » varient en fonction de la valeur «  $k$  ».

Pour sauvegarder la séquence d'énergie dans une fenêtre dynamique  $e_{win}(j/k)$ , les valeurs de " $j$ " vont de 0 à  $L-1$  et les valeurs de " $k$ " vont de 0 à  $+\infty$ .

Selon la valeur de " $k$ ", deux états peuvent être distingués (i) et (ii) :

i) Régime transitoire  $k < L$   $i, j = 0$  à  $k-1$

Le régime transitoire est une phase initiale dans laquelle les composantes de la séquence d'énergie sont initialisées pour les valeurs de " $k < L$ ". Pendant cette phase, la variation des indices " $i$  et  $j$ " est effectuée simultanément. Cette phase a pour but de conserver les  $L$  valeurs récentes de l'énergie dans une fenêtre dynamique. Cette fenêtre est mise à jour à chaque nouveau bloc  $k$  et permet de conserver les  $L$  composantes.

ii) Régime permanent  $k \geq L$   $i = k - L + 1$  à  $k$  et  $j = 0$  à  $L - 1$

Le régime permanent est décrit comme la phase dynamique du système de détection où la séquence d'énergie est mise à jour pour les valeurs de  $k$  et où la décision est prise sur la nature de la séquence captée (impulsive ou non).

Pour chaque nouvelle valeur de  $k$ , les  $L$  composantes récentes de la séquence d'énergie sont calculées en utilisant l'équation (3.2). Cette phase permet de conserver les  $L$  valeurs récentes de l'énergie dans une fenêtre dynamique mise à jour à chaque nouveau bloc  $k$ . En utilisant cette fenêtre, le système peut décider si la séquence captée est impulsive ou non en utilisant des seuils prédéfinis.

- **Normalisation de la séquence d'énergie**

La normalisation de la séquence d'énergie est cruciale pour la méthode de détection, car elle permet l'analyse de la variance de la séquence d'énergie, qui est un critère efficace pour détecter les augmentations soudaines du niveau de signal. Pour normaliser la dynamique de la séquence d'énergie, le vecteur  $e_{win}(j)$  doit être normalisé et transformé dans une gamme de valeurs comprises entre 0 et 1. La formule de calcul de la normalisation est :

$$e_{norm}(j) = \frac{e_{win}(j) - \min_j(e_{win}(j))}{\max_j(e_{win}(j)) - \min_j(e_{win}(j))} \quad (3.3)$$

- **Calcul de la variance des fenêtres normalisées**

La variance des fenêtres normalisées est calculée dans l'étape suivante après la normalisation de la séquence d'énergie. Cette étape consiste à calculer la variance de la séquence d'énergie normalisée en excluant la dernière composante.

La formule de calcul de la variance est :

$$\text{var}(k) = \frac{1}{L-1} \sum_{j=0}^{L-2} [e_{norm}(j) - \bar{e}_{norm}(k)]^2 \quad (3.4)$$

$\bar{e}_{norm}(k)$  représente la moyenne des premières valeurs L-1 de  $e_{norm}(j)$

Cette étape est importante pour détecter les augmentations soudaines des niveaux de signal.

- **Décision**

Après avoir calculé la variance de la séquence d'énergie normalisée, l'étape suivante consiste à la comparer à un seuil "Th". Lorsque la variance est inférieure à ce, un événement impulsif est détecté. Sinon, aucun son impulsif n'est détecté.

Le système de détection des sons impulsifs utilise des seuils prédéfinis pour décider si la séquence captée est impulsive ou non. Ces seuils sont établis expérimentalement en utilisant des événements impulsifs à trois niveaux d'intensités différents (faible, moyen et fort) dans des conditions acoustiques à faible bruit d'environnement. Les seuils de décision optimaux sont trouvés en utilisant ces événements impulsifs et en cherchant un seuil global de décision. Le seuil de décision repose sur une méthodologie spécifique proposée par A. Hattali et A. Mousselmal [16]. Cette méthodologie n'a pas été élaborée ou développée dans notre étude, mais nous avons utilisé le seuil de décision qui y était présenté.

### 3.2.2.2 Reconnaissance des commandes vocales

#### A) Vue globale

La reconnaissance de la parole englobe la capacité humaine à comprendre et à interpréter la parole, ainsi que le processus automatisé de conversion de la parole en texte écrit. La reconnaissance automatique de la parole (Automatic Speech Recognition : ASR) est un processus automatisé qui utilise des algorithmes et des modèles statistiques pour analyser les signaux audio et transcrire les mots prononcés en texte [17].

Les principaux algorithmes utilisés dans l'ASR comprennent les modèles de Markov cachés nommés HMMs, la déformation dynamique du temps nommée Dynamic Time Warping (DTW) et les réseaux de neurones, notamment les réseaux de neurones convolutifs (Convolutional Neural Networks : CNN) et les réseaux de neurones récurrents (Recurrent Neural Networks : RNN) [18].

L'algorithme DTW est une méthode qui permet de trouver la meilleure correspondance entre deux séquences de données. Dans le contexte de l'ASR, la DTW est utilisée pour trouver la meilleure correspondance entre un énoncé oral et un ensemble de mots ou de phrases connus.

Les HMMs sont utilisés pour modéliser les transitions temporelles des phonèmes et des mots dans la parole [19]. Les CNN sont souvent utilisés pour extraire les caractéristiques acoustiques à partir du signal vocal, en particulier dans les tâches de reconnaissance d'acoustique de base [20]. Les RNN, en particulier les LSTMs (Long Short-Term Memory), sont utilisés pour modéliser les dépendances temporelles à long terme et capturer les relations contextuelles entre les phonèmes et les mots [21].

La reconnaissance des commandes vocales est une branche spécialisée de la reconnaissance de la parole qui se concentre sur l'identification et l'exécution de commandes spécifiques prononcées par un utilisateur. Les systèmes de reconnaissance des commandes vocales sont conçus pour comprendre et répondre aux instructions verbales.

Les packages de ASR disponibles sur Python se divisent en deux catégories : (i) les packages en ligne et (ii) les packages hors ligne. Les packages en ligne nécessitent une connexion internet pour effectuer la ASR, tandis que les packages hors ligne peuvent fonctionner sans connexion internet. Parmi les outils de ASR hors ligne, on peut citer CMU Sphinx, Kaldi, Julius, Mozilla DeepSpeech et Pocketsphinx [22]. En ce qui concerne les

packages en ligne, on peut mentionner Google Speech Recognition, Google Cloud Speech-to-Text, Amazon Transcribe, Microsoft Azure Speech Services et IBM Watson Speech to Text. Ce sont tous des packages de la reconnaissance vocale qui permettent de transcrire des fichiers audios en texte. Ces packages sont réputés pour leur précision et leur fiabilité, et peuvent être utilisés à diverses fins [23].

## **B) Approche proposée**

Dans notre étude, nous avons analysé les deux types de packages et nous avons constaté que les packages en ligne fournissent des résultats nettement meilleurs que ceux fonctionnant hors ligne.

Nous utiliserons la technologie de ASR en Arabe. Nous utiliserons la bibliothèque "SpeechRecognition" qui propose plusieurs fonctions de reconnaissance vocale, notamment `recognize_amazon`, `recognize_bing`, `recognize_sphinx`, `recognize_tensorflow`, `recognize_azure`, `recognize_ibm`, `recognize_google_cloud` et `recognize_google` [24].

Parmi ces fonctions, `recognize_google_cloud` utilise l'API de reconnaissance vocale de Google Cloud Speech-to-Text. Pour utiliser la fonction `recognize_google_cloud`, une clé payante d'API Google Cloud valide est requise. En revanche, `recognize_google` est un moteur de reconnaissance vocale gratuit qui se base sur l'API Google Speech Recognition. Cette fonction envoie les données audio à Google pour effectuer la reconnaissance vocale. Cependant, cette commande nécessite une connexion Internet active [24]. Elle est pratique pour la reconnaissance vocale en temps réel et offre généralement une bonne précision. C'est pourquoi nous avons opté pour Google speech recognition comme choix final.

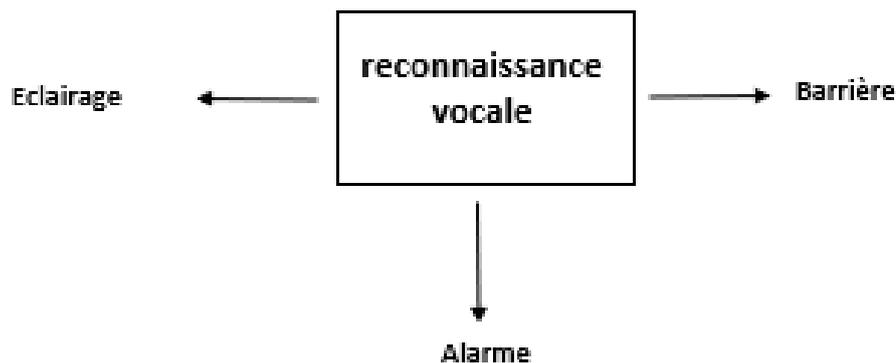
### **3.2.3 Partie matérielle**

Nous avons choisi d'utiliser l'Arduino Uno comme carte d'interface entre le PC et l'environnement extérieur dans notre projet. L'Arduino Uno a été sélectionné en raison de sa convivialité et de sa popularité auprès des débutants. Cela facilite le travail de l'agent de contrôle, qui peut désormais effectuer différentes procédures et actions à distance simplement en utilisant la voix depuis sa position. Grâce à l'utilisation de l'Arduino Uno, les réponses sont rapides et l'agent peut communiquer avec l'ordinateur

principal du parc pour effectuer diverses tâches. Nous avons identifié trois tâches principales pour l'exécution des commandes vocales :

- ✓ Ouverture et fermeture la barrière principale.
- ✓ Allumage et extinction de l'éclairage du parc.
- ✓ Déclenchement et arrêt de l'alarme.

Chaque commande vocale correspond à une tâche spécifique qui est exécutée. La Figure 3.3 illustre les tâches à effectuer après la reconnaissance vocale.

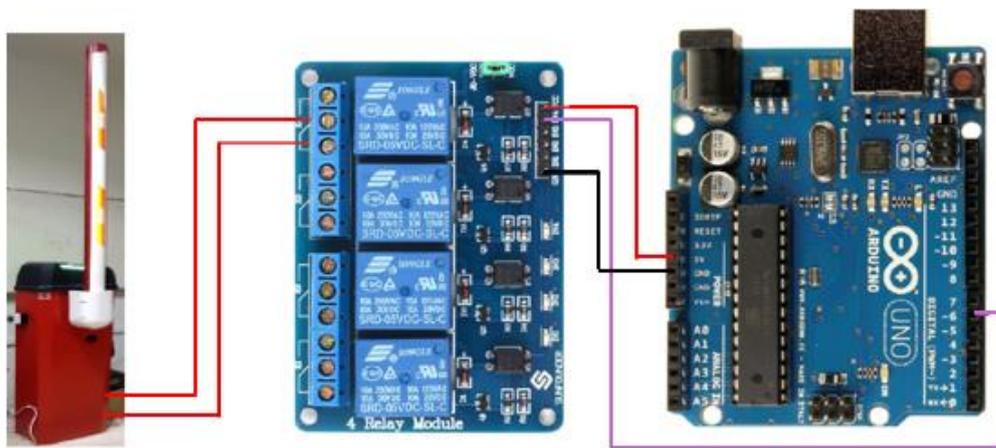


**Figure 3.3:** Tâches à exécuter

### 3.2.3.1 Barrière

Nous utilisons un module de relais 5V pour contrôler l'ouverture et la fermeture de la barrière en raison de sa capacité à commuter des charges électriques avec un signal de commande à faible tension. Pour connecter le module relais à l'Arduino, nous avons relié les entrées du relais (GND, 5V, IN) à l'Arduino. Plus précisément, nous avons connecté l'entrée IN (INPUT) du relais à la broche 6 « numérique » de l'Arduino, et les sorties du relais (COM, NO) à la barrière.

La connexion entre la barrière, le module relais et l'Arduino est représentée dans la figure 3.4.

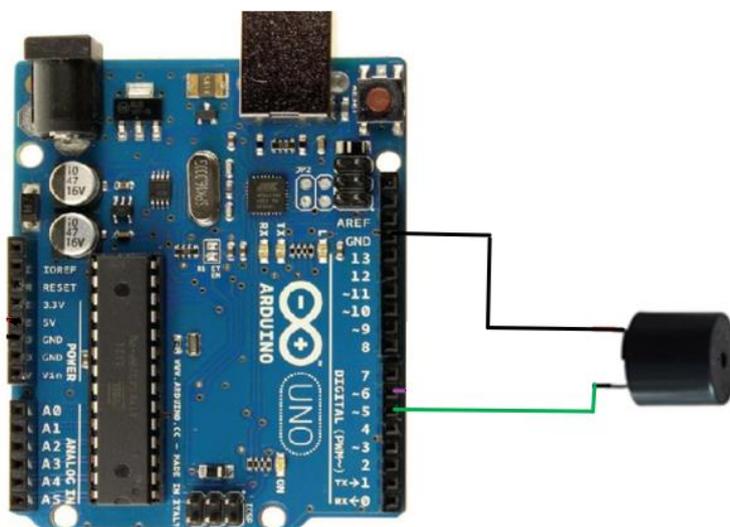


**Figure 3.4:** Schéma de connexion de la barrière, du module relais et de l'Arduino

### 3.2.3.2 Alarme

Pour faire fonctionner le buzzer, nous le connectons à l'Arduino de la manière suivante : la patte inférieure du buzzer est connectée à la broche GND (masse) et la patte supérieure est reliée à la broche 5 de l'Arduino. Afin d'activer ou désactiver le buzzer, il faut fournir une tension de 5V ou 0V.

Le schéma de connexion entre l'Arduino et le buzzer est représenté dans la figure 3.5.



**Figure 3.5:** Schéma de connexion du buzzer et de l'Arduino

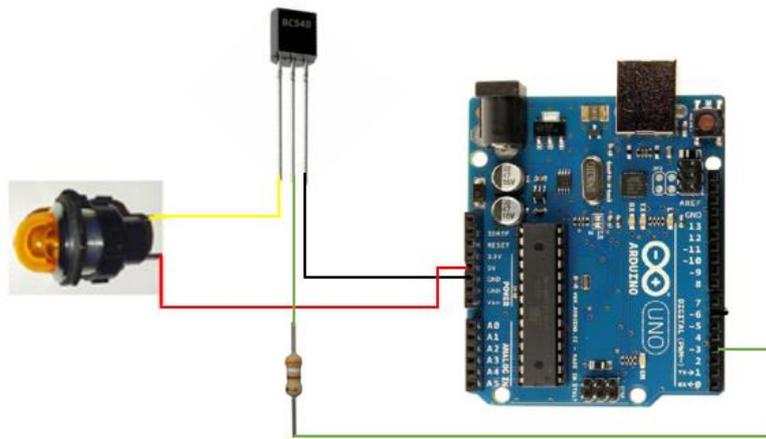
### 3.2.3.3 Eclairage

Pour la lampe, nous utilisons un support de lampe accompagné d'une lampe fonctionnant sous 5 volts et 100 mA. De plus, nous utilisons un transistor bipolaire de type NPN BC548,

car le courant disponible sur la broche numérique de l'Arduino (pin 3) est de 40 mA, tandis que la lampe nécessite un courant de 100 mA.

Nous avons connecté la première patte de la lampe aux 5 volts de l'Arduino et la deuxième patte de la lampe au collecteur (C) du transistor. L'émetteur (E) du transistor est relié à la masse (GND) de l'Arduino et la base (B) du transistor est connectée à une résistance, puis au transistor lui-même et à la broche 3 de l'Arduino.

Le schéma de connexion entre la lampe, le transistor et l'Arduino est représenté dans la figure 3.6.



**Figure 3.6:** Schéma de connexion de la lampe, du transistor et de l'Arduino

### 3.3 Conclusion

Dans ce chapitre, nous avons présenté la solution proposée pour la gestion des parkings. Nous avons détaillé la méthodologie globale de gestion tenant compte la détection des sons impulsifs, la reconnaissance vocale ainsi que la partie matérielle. Dans le chapitre suivant, nous présentons les résultats expérimentaux ainsi que leurs interprétations.

# Chapitre 4 : Résultats et interprétations

## 4.1 Introduction

Dans ce chapitre, nous présenterons les résultats obtenus par l'application de la solution proposée pour la gestion intelligente des parkings. Nous présentons trois sections importantes : (i) les logiciels et langages de programmation utilisés, (ii) la connexion entre Python et Arduino, et (iii) l'implémentation de la solution. Nous terminons ce chapitre par une conclusion.

## 4.2 Logiciels et langages de programmation utilisés

Dans cette partie, nous présenterons le langage Python ainsi que l'éditeur Jupyter. Nous aborderons également le logiciel Arduino, les applications Wo Mic et Lesser AudioSwitch, ainsi que la conception des interfaces graphiques avec Qt Designer.

### 4.2.1 Langage python

Guido van Rossum [25] est le créateur de Python. Il a publié la première version de Python en 1991. Python est l'un des langages de programmation de haut niveau les plus utilisés. Il s'agit d'un langage de programmation open source qui offre de nombreuses bibliothèques de support. Ces bibliothèques constituent l'un des principaux avantages de Python, ce qui en fait une plateforme capable d'évoluer et de se développer.

### 4.2.2 Editeur de programmation

Il existe de nombreux éditeurs de programmation Python. Nous citons : PyCharm, Visual Studio Code, Sublime Text, Jupyter Notebook, Spyder IDE (environnement de développement intégré).

Nous avons choisi d'utiliser Jupyter Notebook (Voir la Figure 4.1). Qui est un éditeur de programmation enregistré avec l'extension "ipynb". C'est une application web open source [26]. L'un de ses avantages est sa capacité à exécuter du code à l'aide de cellules individuelles. Chaque cellule peut contenir du code indépendant, ce qui vous permet de tester et de modifier facilement des parties spécifiques du code sans avoir à exécuter l'ensemble du notebook à chaque fois. Cela accélère le processus de correction d'erreurs. Cette fonctionnalité vous permet également de mieux comprendre chaque morceau de

code. De plus, Jupyter facilite la création et l'affichage direct des graphiques dans les notebooks.

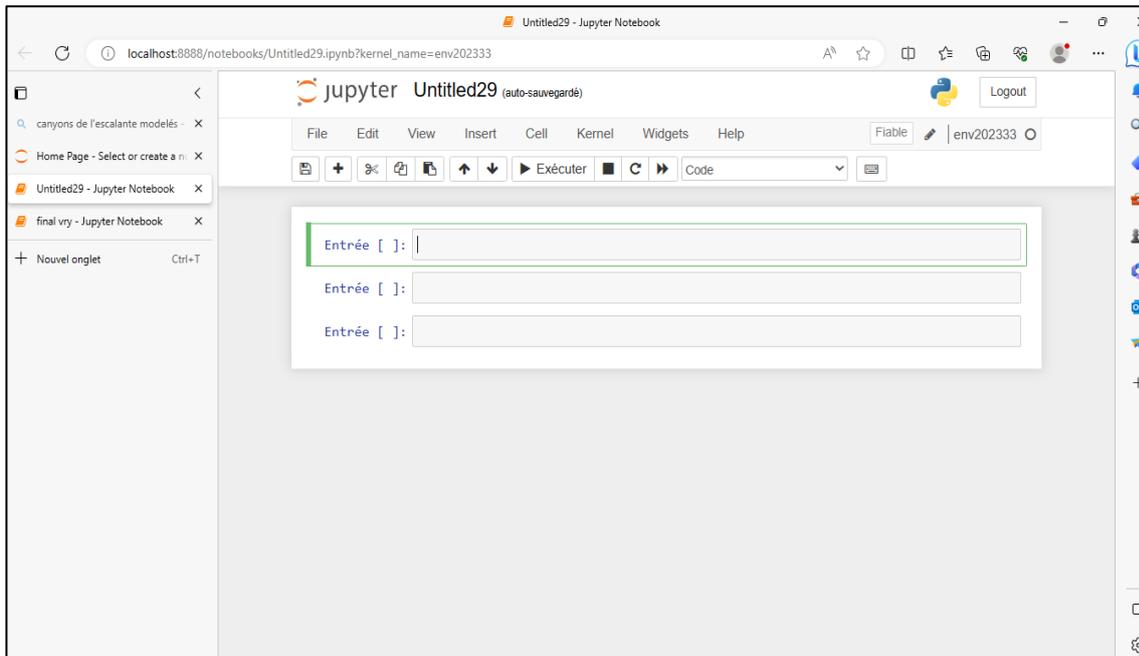
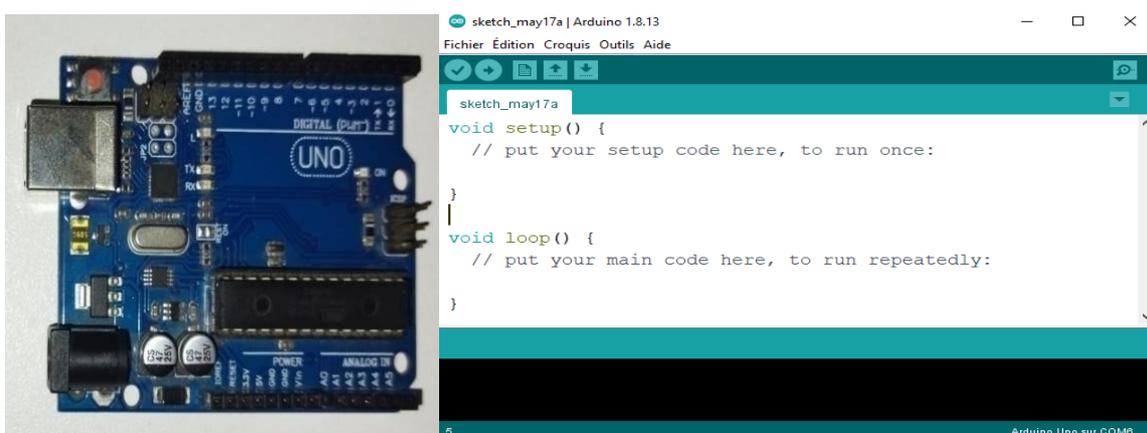


Figure 4.1: Jupyter Notebook

### 4.2.3 Arduino

Arduino est une plateforme électronique open-source basée sur du matériel et des logiciels faciles à utiliser. Elle est destinée à tous ceux qui réalisent des projets interactifs [27]. L'éditeur de programmation Arduino est en effet inspiré de plusieurs langages. Dont le C et le C++. Nous avons utilisé la version Arduino 1.8.13 (Figure 4.2)



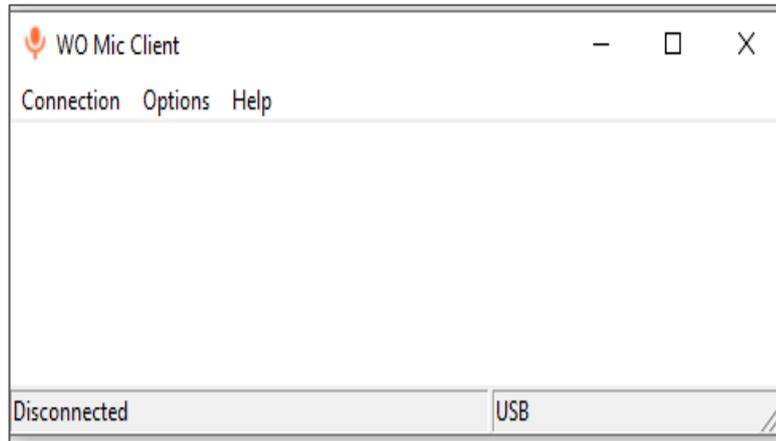
(a)

(b)

Figure 4.2: (a) Carte Arduino Uno. (b) Interface d'utilisateur de l'Arduino

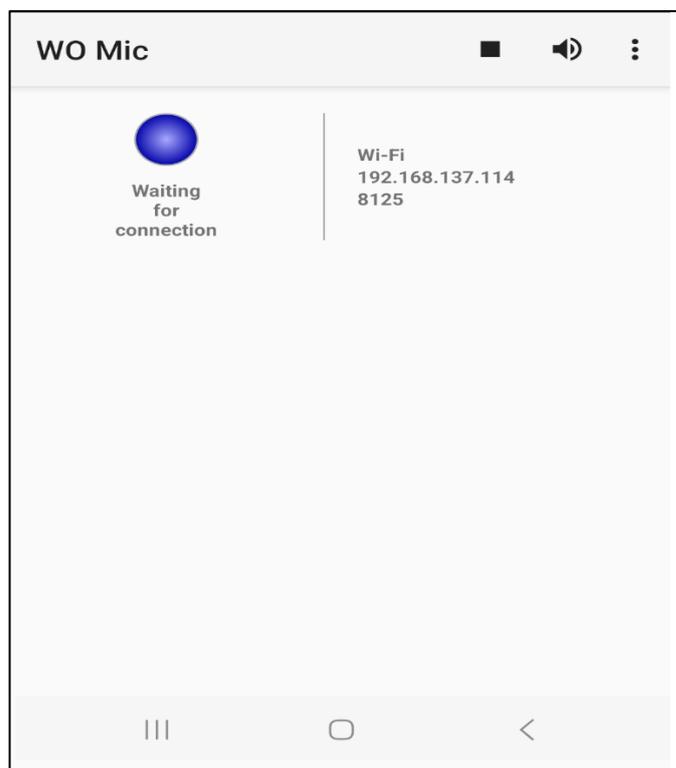
#### 4.2.4 Wo Mic

Wo Mic est un outil qui transforme un smartphone en microphone pour n'importe quel ordinateur. Pour cela, il existe deux plateformes différentes pour utiliser Wo Mic. La première plateforme est une application Windows appelée "WO Mic Client" qui s'installe sur l'ordinateur [14] (Figure 4.3).



**Figure 4.3:** Wo Mic Client

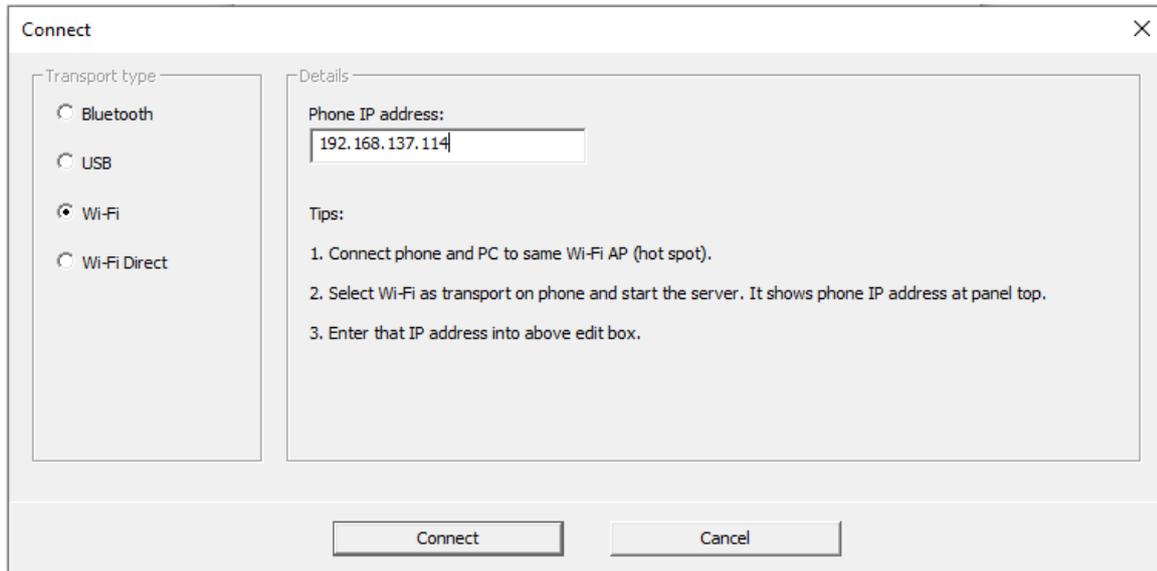
La deuxième plateforme est une application Android appelée "WO Mic" qui se télécharge sur le smartphone [14] (Figure 4.4).



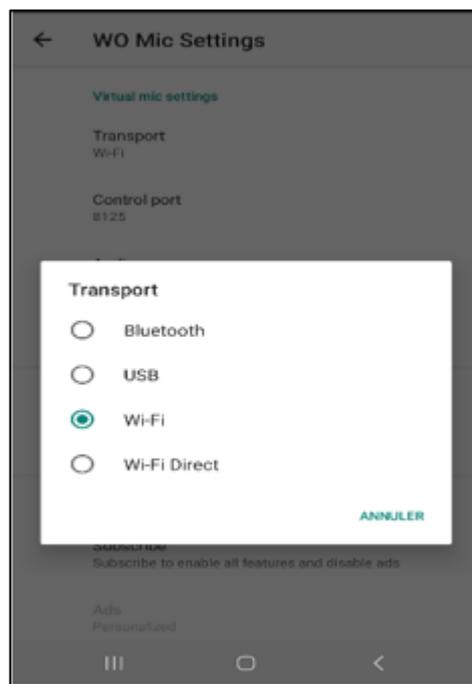
**Figure 4.4:** Wo Mic

Il existe quatre options de connexion entre l'ordinateur (Figure 4.5) et le smartphone (Figure 4.6) : "Bluetooth", "USB", " Wi-Fi" (Wireless Fidelity), et "Wi-Fi Direct"

Dans notre cas, nous avons utilisé le mode Wi-Fi. Nous devons créer un réseau en activant le point d'accès de l'ordinateur. Ensuite, il faut connecter ce réseau avec le smartphone.



**Figure 4.5:** Option de Wo Mic client



**Figure 4.6:** Option de Wo Mic

Pour améliorer notre travail, nous avons connecté des écouteurs Bluetooth au téléphone afin de les transformer en microphone pour le téléphone. Cela est encore plus utile

lorsque nous utilisons Wo Mic via USB pour aider l'agent de parking à se déplacer dans la salle de contrôle. Pour réaliser cette configuration, il faut installer l'application Lesser AudioSwitch sur notre smartphone (Figure 4.7).

Lesser AudioSwitch offre de nombreuses fonctionnalités utiles, notamment la transformation des écouteurs Bluetooth en microphone, la possibilité de changer la sortie audio entre les écouteurs et les haut-parleurs, ainsi que d'autres fonctionnalités [28].



**Figure 4.7:** Application Lesser AudioSwitch

#### 4.2.5 Conception des interfaces graphiques

La conception d'interfaces graphiques Utilisateurs (Graphical User Interface : GUI) peut être réalisée à l'aide de différents outils, tels que Tkinter, Qt Designer, Adobe XD, Qt Creator, Figma. Nous avons choisi Qt Designer en raison de ses avantages (voir la Figure 4.8) [29] :

- ✓ Il facilite le processus de création d'interfaces en permettant aux utilisateurs de faire glisser et déposer des éléments d'interface tels que des boutons, des zones de texte, des images.
- ✓ Qt Designer offre une grande flexibilité en permettant la personnalisation des éléments d'interface tels que les couleurs, les polices et les tailles, ce qui permet aux utilisateurs de créer des interfaces graphiques personnalisées.

- ✓ Un avantage de Qt Designer est qu'il permet de convertir facilement les fichiers enregistrés format "ui" (User Interface) en langages de programmation tels que C++ ou Python pour les intégrer dans les codes. Cela permet d'économiser beaucoup de temps.

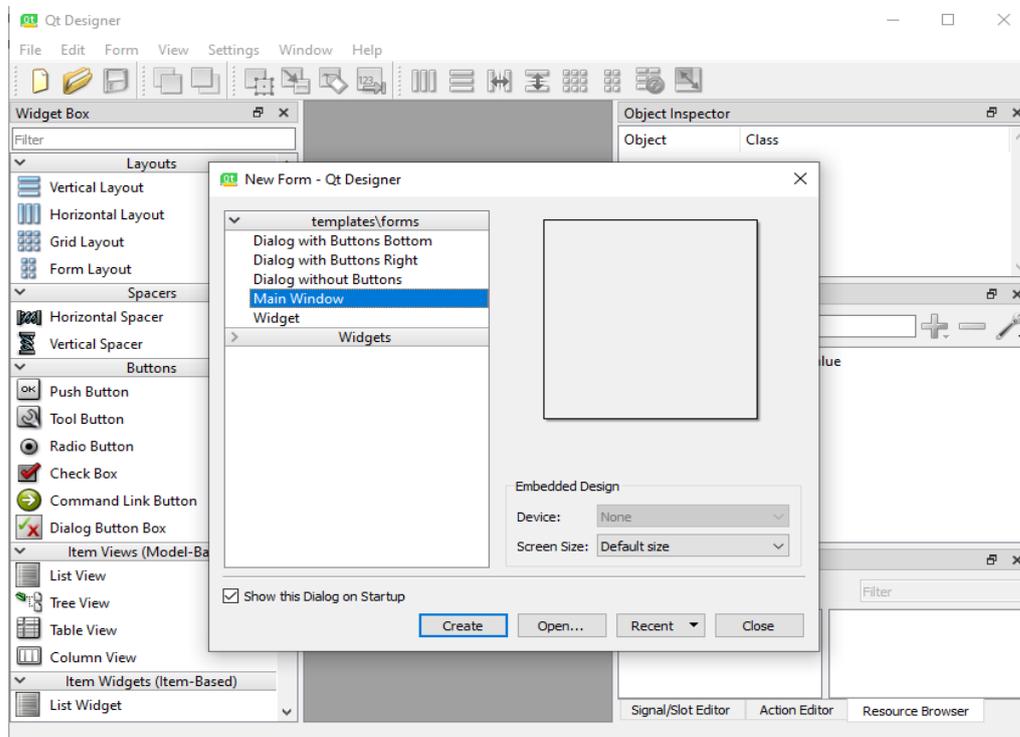


Figure 4.8: Qt Designer

Pour convertir une interface conçue avec Qt Designer en code Python, il suffit d'utiliser cette instruction (voir la Figure 4.9) :

```
Anaconda Powershell Prompt (anaconda w10)
(env202333) PS C:\Users\ELMOHIT\Desktop\QT> conda activate env202333
(env202333) PS C:\Users\ELMOHIT\Desktop\QT> cd C:\Users\ELMOHIT\Desktop\QT\
(env202333) PS C:\Users\ELMOHIT\Desktop\QT> Python -m PyQt5.uic.pyuic -x guienarab.ui -o guienarab.py
(env202333) PS C:\Users\ELMOHIT\Desktop\QT>
```

Figure 4.9: Conversion « ui » vers « py »

### 4.3 Connexion entre Python et Arduino

Il existe plusieurs façons de connecter Arduino et Python. Deux méthodes couramment utilisées [25]:

- La bibliothèque PySerial : est une bibliothèque générale pour la communication série.

- La bibliothèque PyFirmata : est une bibliothèque spécialement conçue pour communiquer avec les cartes Arduino en utilisant le protocole Firmata.

Dans notre projet, nous avons choisi La bibliothèque PyFirmata.

### 4.3.1 Bibliothèque « Firmata »

Pour téléverser la bibliothèque « Firmata » sur une carte Arduino, il est nécessaire d'installer la bibliothèque « Firmata » dans l'IDE Arduino. Pour ce faire, il faut suivre les étapes suivantes [25]:

- ✚ Ouvrir l'IDE Arduino.
- ✚ Aller dans le menu "Croquis" et sélectionner "Inclure une bibliothèque".
- ✚ Sélectionner "Gérer les bibliothèques".
- ✚ Rechercher "Firmata" et sélectionner "Firmata by Firmata team".
- ✚ Cliquer sur "Installer".

Une fois la bibliothèque Firmata installée, il est possible de téléverser la bibliothèque « Firmata » sur la carte Arduino en suivant les étapes suivantes :

- ✚ Ouvrir l'exemple "StandardFirmata" dans le menu "Firmata" de l'IDE Arduino.

(Figure 4.10)

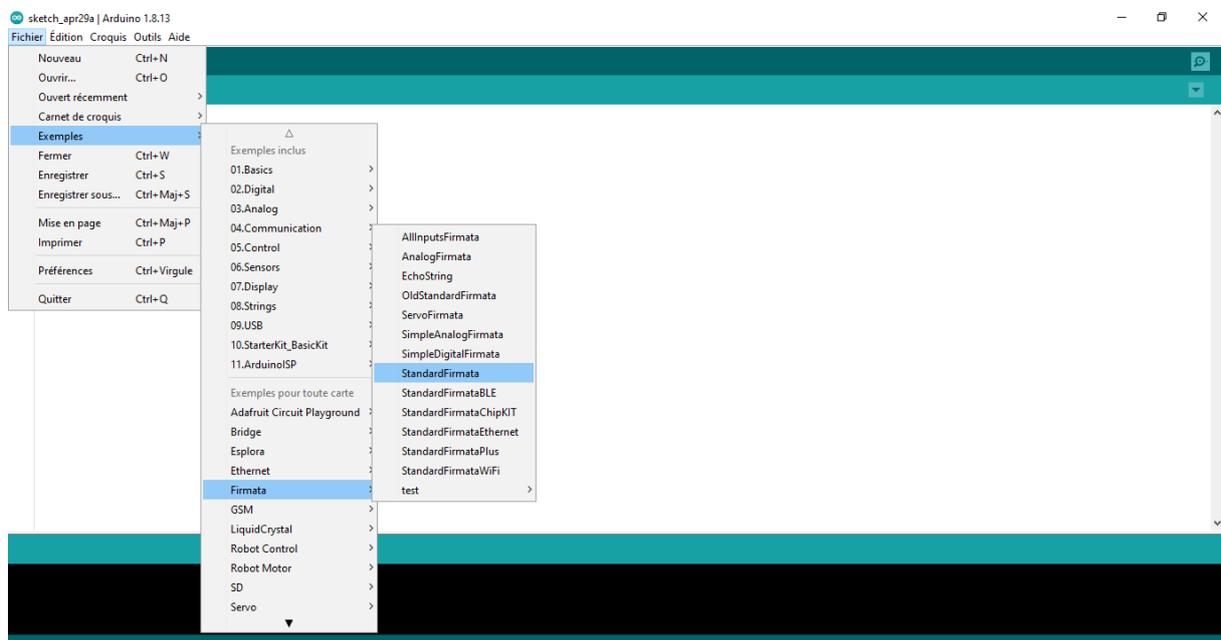


Figure 4.10: Exemple de « firmata »

- ✚ Vérifier que le bon port série et le bon type de carte sont sélectionnés dans le menu "Outils".

 Téléverser le programme sur la carte Arduino en cliquant sur le bouton "Téléverser" (Figure 4.11)



**Figure 4.11:** Téléversement de programme « firmata »

Une fois la bibliothèque « Firmata » téléversé sur la carte Arduino, il est possible de contrôler la carte Arduino à distance en utilisant une bibliothèque « PyFirmata » en Python.

### 4.3.2 Bibliothèque « PyFirmata »

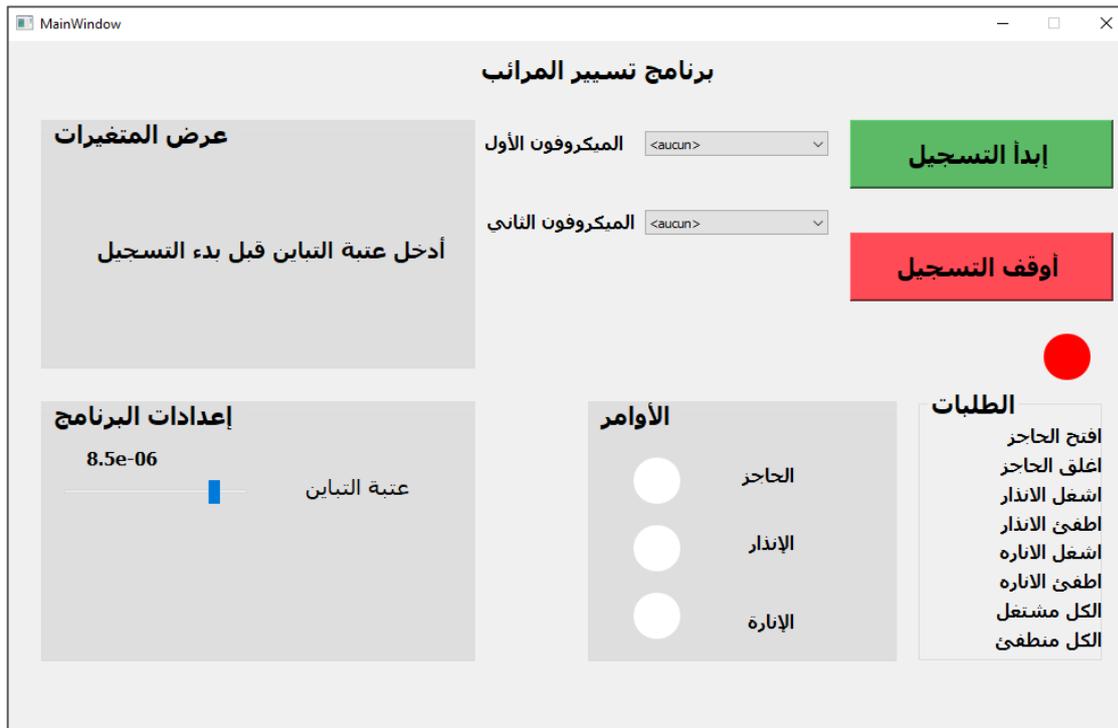
PyFirmata est une bibliothèque Python qui permet de contrôler des cartes électroniques programmables, telles que les cartes Arduino. PyFirmata utilise le protocole Firmata [25]. Nous devons exécuter cette commande (Pip install pyfirmata) dans notre terminal ou notre invite de commandes, à condition que nous ayons pip installé sur notre système. Une fois l'installation terminée, nous pourrons utiliser PyFirmata dans notre code Python en important la bibliothèque (import pyfirmata) et en initialisant une connexion avec notre carte programmable [25].

## 4.4 Implémentation de la solution proposée

Dans cette section, nous discuterons l'interface du projet.

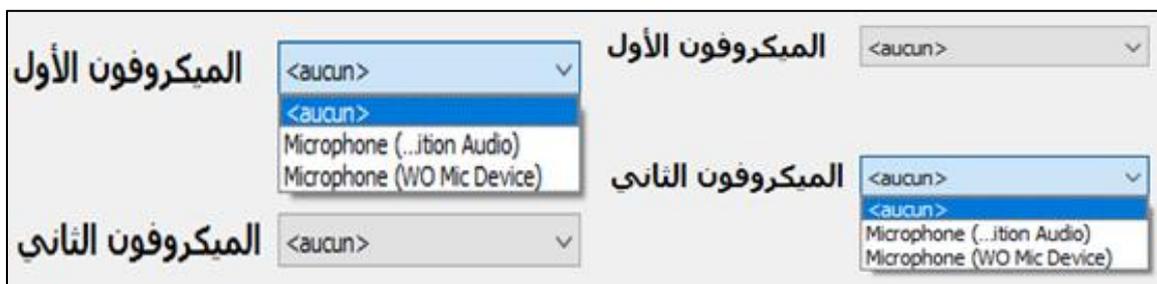
Pour commencer, nous avons connecté l'Arduino à l'ordinateur. Ensuite, nous avons connecté le téléphone à Wo Mic pour ajouter un autre microphone. Nous avons également connecté les écouteurs Bluetooth au smartphone pour utiliser le microphone

des écouteurs au lieu le microphone du téléphone. La Figure 4.12 présente les étapes de mise en œuvre de notre travail.



**Figure 4.12:** Interface de la gestion intelligente par commande vocale

La première étape consiste à sélectionner le premier et le deuxième microphone. En appuyant sur le "combo box", une liste des microphones connectés à l'ordinateur apparaît. Dans la Figure 4.13, seuls deux choix sont disponibles, le premier choix correspondant au microphone de l'ordinateur et le deuxième choix correspondant au microphone du téléphone.



**Figure 4.13:** Liste des choix de microphones

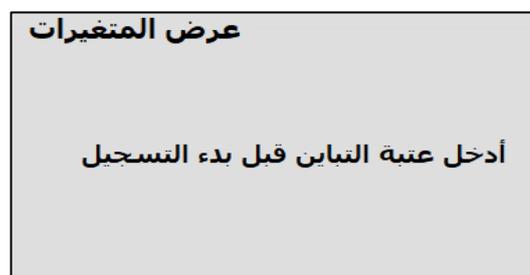
Si nous souhaitons augmenter le nombre de choix, il suffit de connecter un autre microphone à l'ordinateur.

Il est nécessaire de sélectionner deux microphones pour le fonctionnement de notre logiciel (Figure 4.14). Le premier est utilisé pour l'acquisition permanente et en temps réel des sons de l'environnement afin de détecter un éventuel son impulsif. Le deuxième microphone enregistre la voix pendant 5 secondes une fois qu'un son impulsif est détecté, tel qu'un claquement de mains. Pendant cette période, l'agent prononce la commande à exécuter.



**Figure 4.14:** Choix des microphones

Le logiciel nécessite l'introduction du seuil de détection des sons impulsifs " ادخل عتبه " (Veuillez saisir le seuil de détection des sons impulsifs avant de commencer l'enregistrement) (voir Figure 4.15).



**Figure 4.15:** Affichage de variance

Cela implique qu'il est nécessaire d'ajuster le seuil de variance à une valeur spécifique en fonction de la sensibilité du premier microphone choisi. Cette opération s'effectue en déplaçant le curseur du horizontalSlider (voir la Figure 4.16). Si le microphone est très sensible, il est nécessaire d'augmenter le seuil de variance, tandis que s'il est moins sensible, il faut réduire le seuil de variance.

Nous avons initialement fixé la valeur à  $8.5e-06$  car elle correspond à la sensibilité de notre microphone d'ordinateur que nous avons choisi d'utiliser (selon [16]).



**Figure 4.16:** Configuration du seuil de la variance

La Figure 4.17 précise la liste des commandes disponibles dans le système qui doivent être prononcées. Il y a huit tâches .



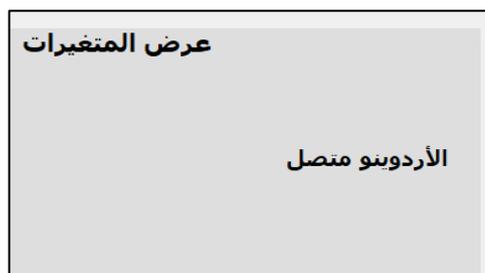
**Figure 4.17:** Liste des tâchés

À la fin de tout cela, nous appuyons sur le bouton " ابدأ التسجيل " (Démarrer l'enregistrement) pour démarrer le programme. Si l'Arduino n'est pas connecté, nous verrons sur l'affichage des variables le message " الاردوينو غير متصل " (l'Arduino n'est pas connecté) (Figure 4.18).



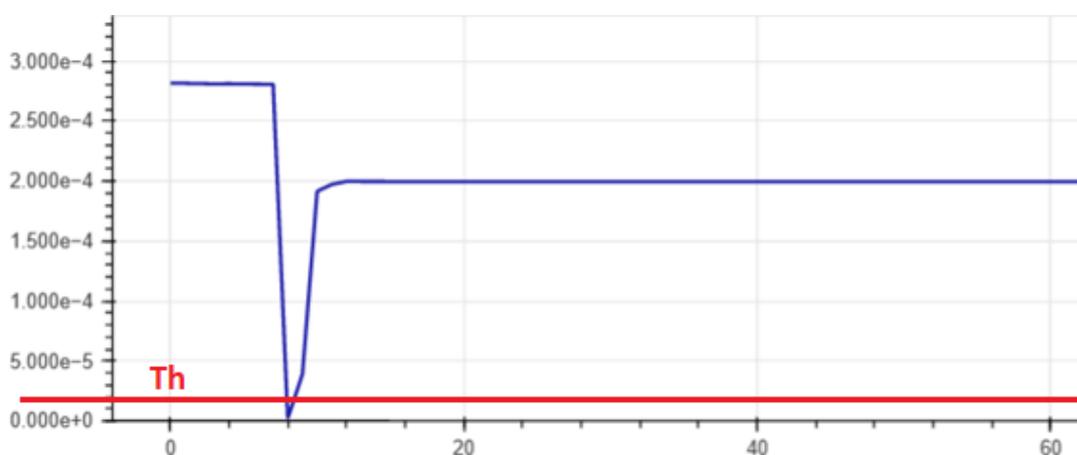
**Figure 4.18:** (a) Bouton de démarrage. (b) Affichage

Lorsque l'Arduino est connecté, le message "الاردوينو متصل" (l'Arduino est connecté) s'affichera, ce qui déclenchera l'enregistrement en temps réel (voir la Figure 4.19).



**Figure 4.19:** Affichage de Arduino

Lorsqu'un son impulsif est détecté (claquement de mains), cela signifie que le niveau sonore dépasse le seuil de variance (Voir la figure 4.20).



**Figure 4.20:** Graphe du seuil de variance

Ensuite, l'écran affiche le message "ماذا تريد مني يا سيدي" (que puis-je faire pour vous, monsieur), et le système le prononce également. De plus, le cercle situé à droite devient vert, et devant celui-ci s'affiche "أنا أسمعك" (Je t'écoute) (Figure 4.21), ce qui signifie que le système est en attente de l'instruction à exécuter. Cette instruction sera enregistrée à l'aide du deuxième microphone. La durée de l'enregistrement est de cinq secondes.



**Figure 4.21:** Message généré suite à la détection d'un son impulsif

Dans les cinq secondes, l'agent de stationnement aura prononcé l'instruction demandée. Si l'instruction prononcée correspond aux commandes du système, elle sera affichée. Voici les exemples que nous avons étudiés :

Le premier exemple concerne la barrière ; lorsque l'agent prononce la commande " افتح الحاجز " (ouvre la barrière), le message suivant s'affiche sur l'écran " لقد قلت افتح الحاجز " (Vous avez dit qu'ouvre la barrière) ; et le cercle correspondant devient également vert (Voir la Figure 4.22). Ensuite, l'Arduino exécute l'instruction pour ouvrir la barrière (voir la Figure 4.23).

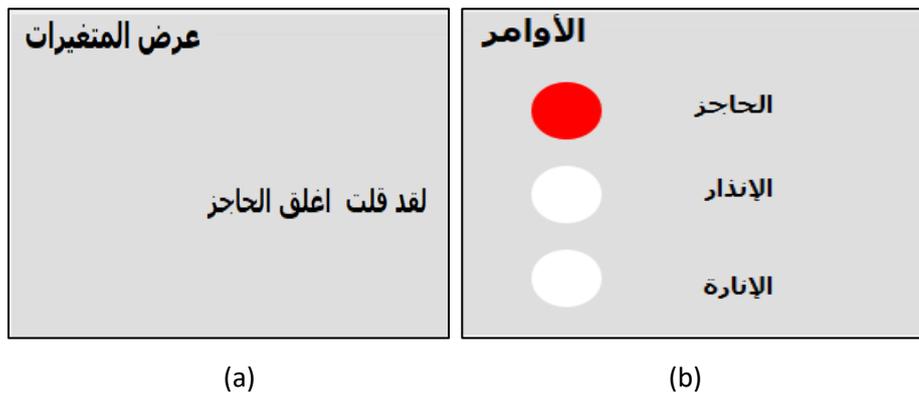


**Figure 4.22:** (a) Message affiché. (b) Modification de la couleur du cercle de la barrière.



**Figure 4.23:** Action d'ouverture de la barrière

Lorsque l'agent souhaite fermer la barrière, il lui suffit de claquer ses propres mains une fois de plus et de prononcer la demande (Figures 4.24 et 4.25).



**Figure 4.24:** (a) Affichage du message de la barrière. (b) Changement de la couleur du cercle

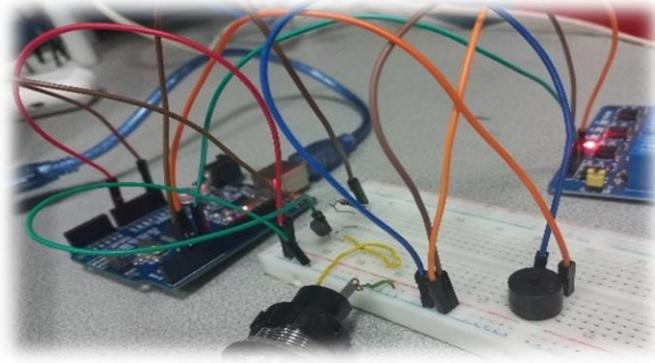


**Figure 4.25:** Fermeture de la barrière

Le deuxième cas concerne le fonctionnement du système d'alarme. Lorsque l'agent prononce la commande "اشغل الإنذار" (active l'alarme), l'écran affiche le message "لقد قلت اشغل الإنذار" (vous avez dit qu'active l'alarme), et le cercle de l'alarme change de couleur pour devenir vert (Voir la Figure 4.26). Ensuite, l'Arduino exécute l'instruction nécessaire pour activer l'alarme (voir la Figure 4.27).

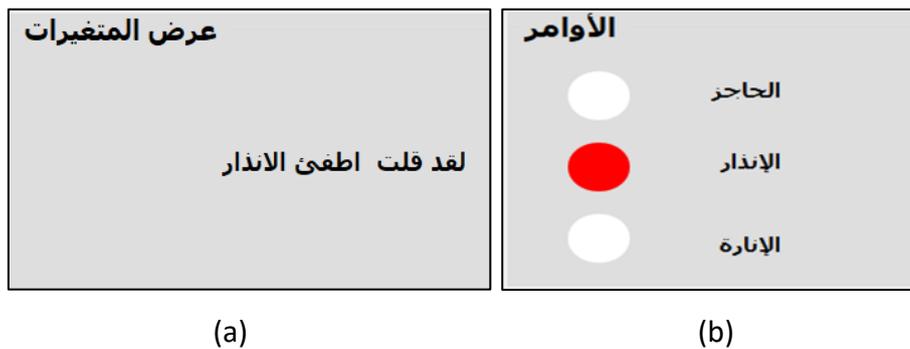


**Figure 4.26:** (a) Affichage du message de l'alarme. (b) Changement la couleur du cercle de l'alarme



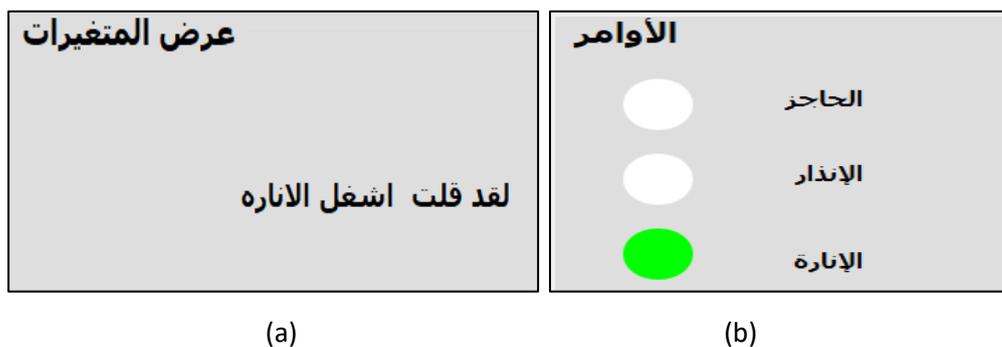
**Figure 4.27:** Montage électronique incluent le buzzer Commettant une alarme

Dans la Figure 4.28, on peut observer que lorsque l'agent prononce la commande " اطفى الانذار " (éteins l'alarme), le message " لقد قلت اطفى الانذار " (vous avez dit qu'éteins l'alarme) s'affiche à l'écran, et le cercle correspondant devient rouge. Ensuite, l'Arduino exécute l'instruction pour éteindre l'alarme.

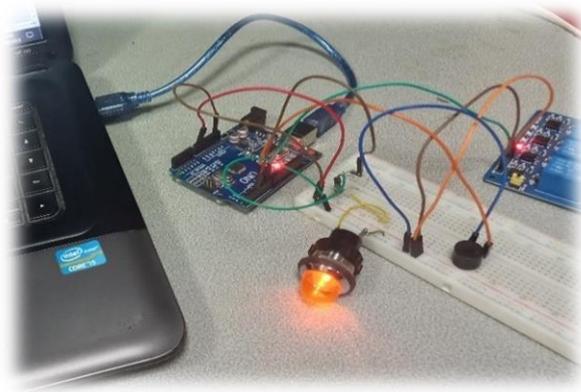


**Figure 4.28:** (a) Affichage du message d'alarme. (b) Désactivation de l'alarme

Dans le troisième exemple, nous nous intéressons à l'éclairage. Lorsque la commande " اشغل الانارة " (allume l'éclairage) est prononcée, l'écran affiche le message " لقد قلت اشغل الانارة " (Vous avez dit qu'allume l'éclairage), et le cercle correspondant passe également au vert (Figure 4.29). Par la suite, l'Arduino exécute l'instruction permettant d'allumer l'éclairage (Figure 4.30).



**Figure 4.29:** (a) Affichage du message d'éclairage. (b) Changement la couleur du cercle

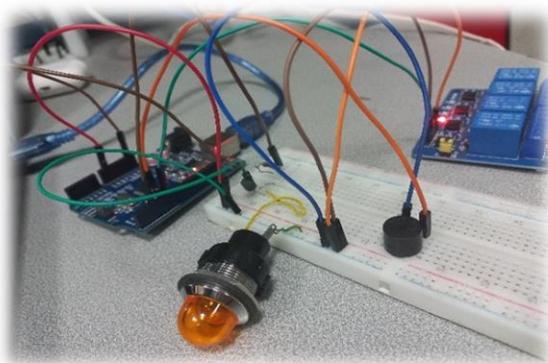


**Figure 4.30:** Activation de l'éclairage

La Figure 4.31 montre que lorsque l'agent prononce la commande " اطفى الاناره " (éteins l'éclairage), le message suivant s'affiche sur l'écran " لقد قلت اطفى الاناره " (vous avez dit qu'éteins l'éclairage"), et le cercle correspondant devient également rouge. Ensuite, l'Arduino exécute l'instruction pour éteindre l'éclairage (voir la Figure 4.32).



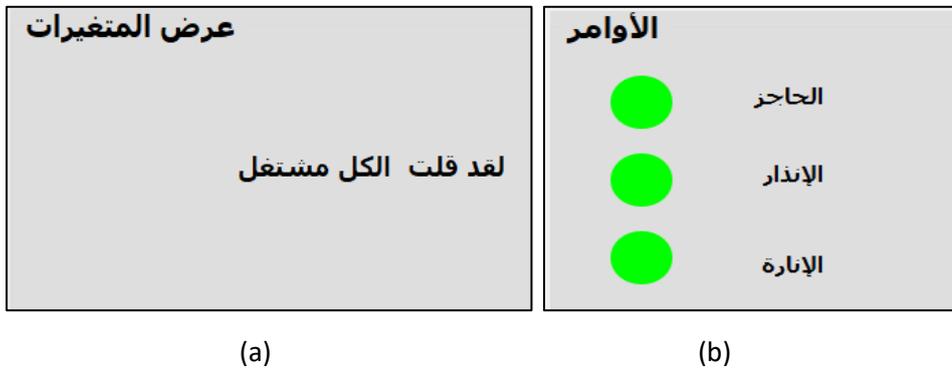
**Figure 4.31:** (a) Affichage du message relatif à l'éclairage. (b) Changement de la couleur du cercle de l'éclairage



**Figure 4.32:** Extinction de l'éclairage

Le quatrième exemple exécute toutes les tâches. Tout ce que l'agent a besoin de faire est de prononcer la commande " الكل مشغل " (tout est en marche), puis le message " لقد قلت "

"الكل مشغول" (vous avez dit que tout est en marche) apparaîtra sur l'écran des variables, et tous les cercles correspondant aux tâches deviendront verts (Voir la Figure 4.33). Ensuite, l'Arduino exécutera toutes les tâches simultanément (Voir la Figure 4.34).

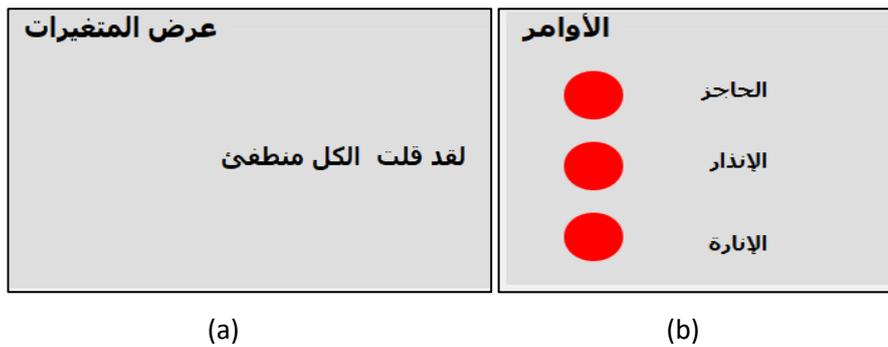


**Figure 4.33:** (a) Affichage de l'exécution de la commande. (b) Changement de toutes les couleurs



**Figure 4.34:** Exécution de toutes les tâches

Dans le dernier exemple, l'objectif est d'arrêter toutes les tâches, en prononçant « الكل منطفي » (arrête tout), le message « لقد قلت الكل منطفي » (vous avez dit qu'arrête tout) s'affiche sur l'écran des variables, et tous les cercles correspondant aux tâches deviennent rouges (Figure 4.35). Ensuite, l'Arduino exécute toutes les tâches simultanément (Figure 4.36).

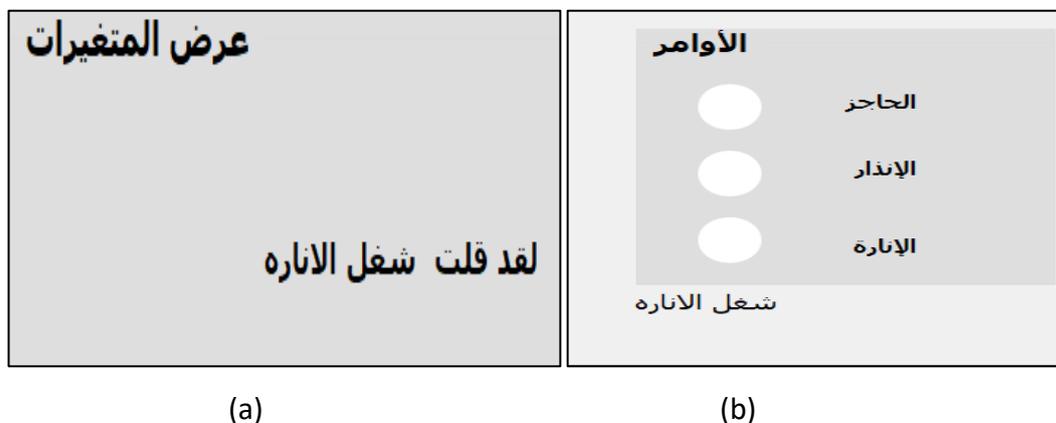


**Figure 4.35:** (a) Affichage de l'arrêt. (b) Changement de toutes les couleurs



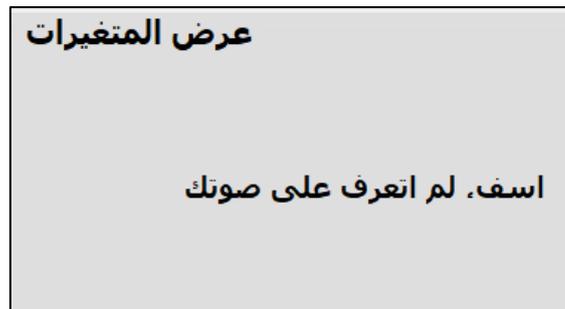
**Figure 4.36:** Arrêter toutes les tâches

Dans le cas où la tâche prononcée ne correspond pas aux tâches du système, elle sera affichée, et le système dira : « انا لا اعرف هذا الامر » (je ne connais pas cette commande) ; cela se produit en raison de la reconnaissance vocale. De plus, le programme réécrit le mot incorrect à la fin de l'interface afin que nous ne le répitions pas à nouveau, et le mot reste affiché devant nous jusqu'à ce que nous prononcions un autre mot incorrect. Il sera ensuite remplacé. (Figure 4.37)



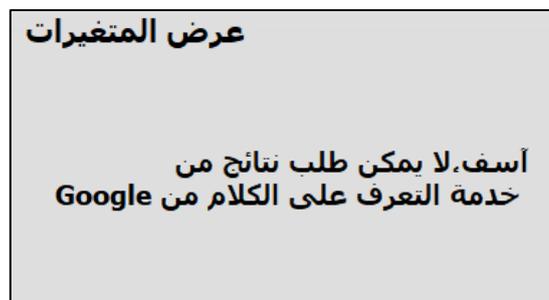
**Figure 4.37:** (a) Message correspondant à un problème de la reconnaissance vocale. (b) réécriture du mot incorrect

Si le logiciel ne reconnaît pas la parole, il affichera « اسف لم اتعرف على صوتك » (désolé, je n'ai pas pu reconnaître votre voix), ce qui est causé par une déconnexion du deuxième microphone de l'ordinateur. La Figure 4.38 illustre ce cas.



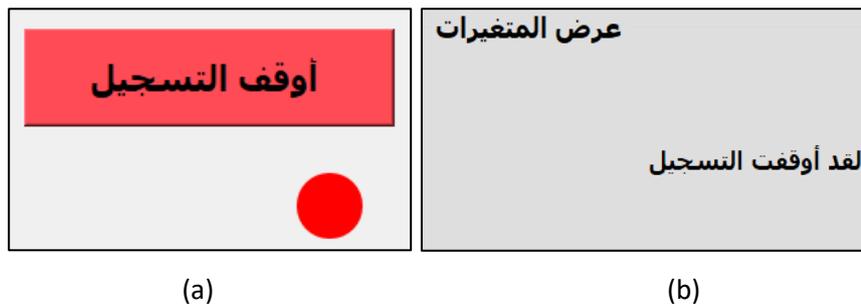
**Figure 4.38:** Affichage du message correspondant à un problème de déconnexion du microphone

La connexion internet est un élément essentiel pour le fonctionnement de notre système. Si la connexion est faible, cela entraînera une incapacité à reconnaître la parole et affichera : " Google من الكلام من خدمة التعرف على الكلام من Google " (désolé, impossible de demander des résultats au service de reconnaissance vocale de Google) (voir la Figure 4.39).



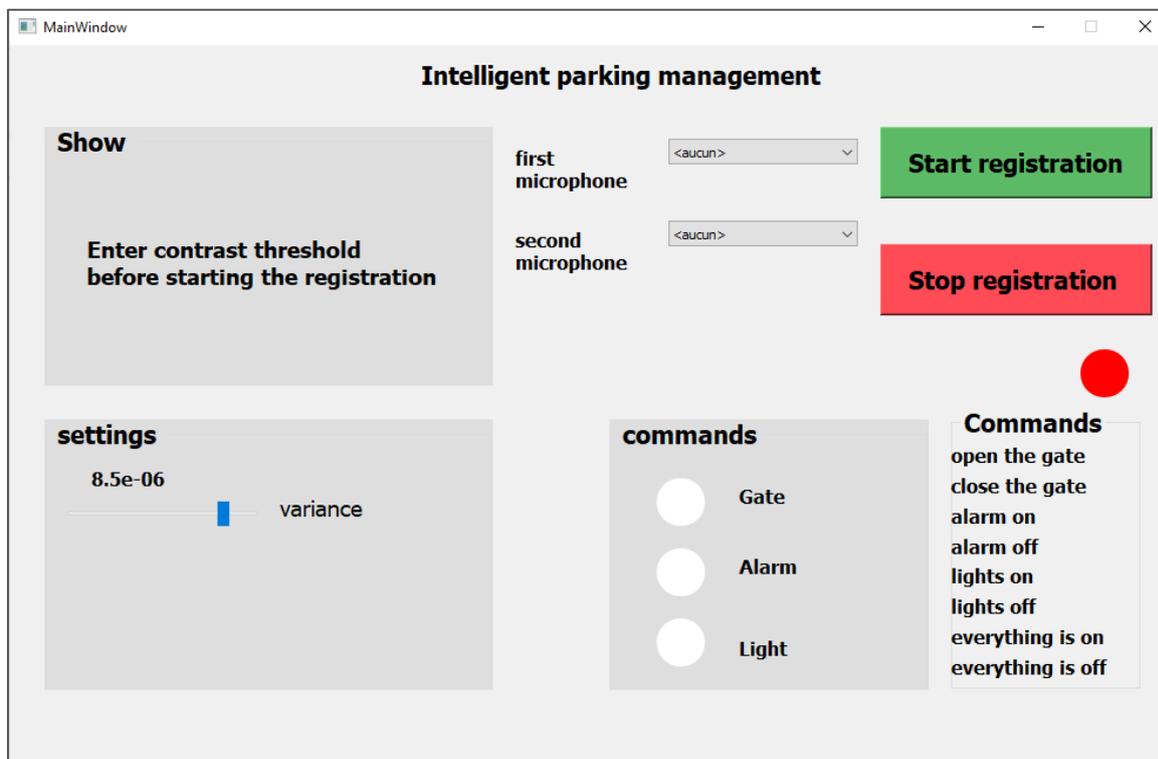
**Figure 4.39:** Affichage du message correspondant à problème de connexion avec google speech recognition

Lorsque l'agent de parc de stationnement souhaite arrêter le programme, il lui suffit d'appuyer sur le bouton اوقف التسجيل (arrête l'enregistrement). Sur l'écran, il affiche " لقد اوقفت التسجيل " (Vous avez arrêté l'enregistrement) et le cercle correspondant à l'acquisition en temps réel redevient rouge, ce qui entraîne la déconnexion avec l'Arduino (Figure 4.40).

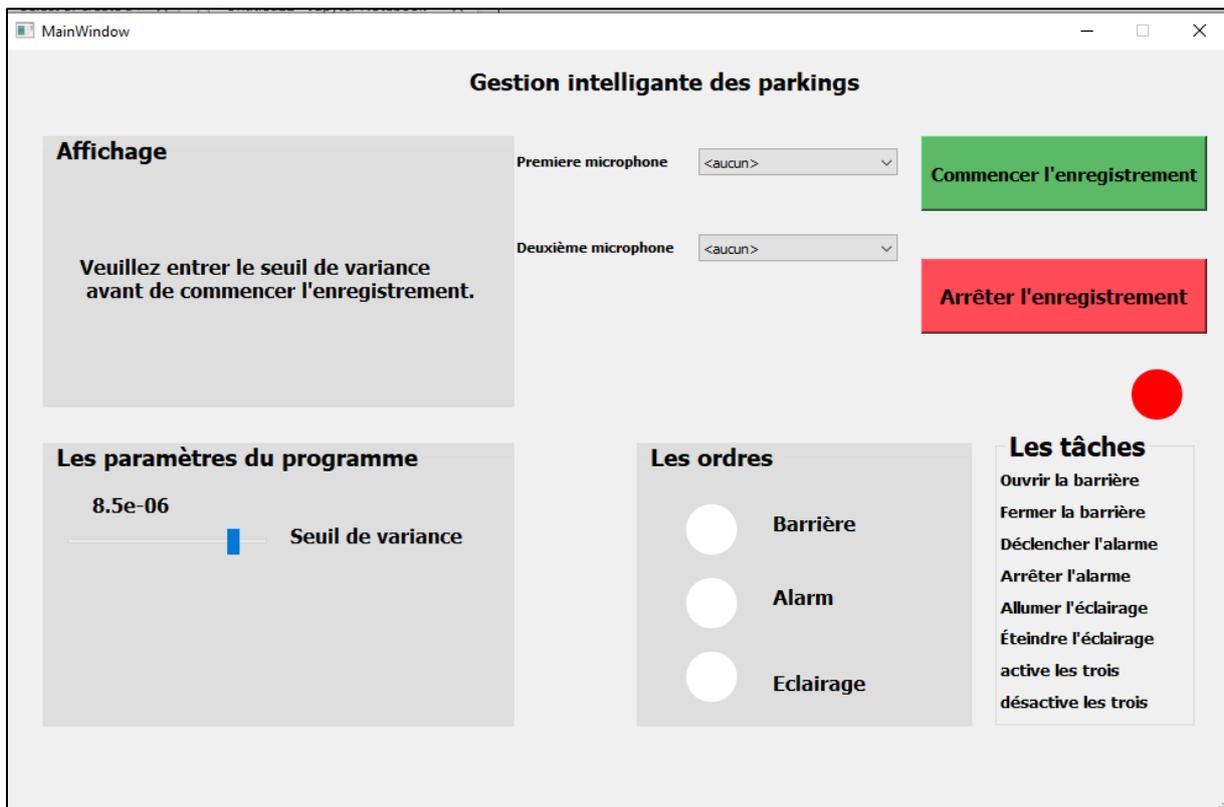


**Figure 4.40:** (a) Bouton d'arrêt. (b) Affichage du message d'arrêt

Nous avons également traduit l'interface en anglais (voir la Figure 2.41) et en Français (voir la Figure 2.42). Cela nous aide à garantir une communication efficace entre les agents de parking et le logiciel proposé.



**Figure 4.41:** Interface de la gestion intelligente du parking en langue Anglaise



**Figure 4.42:** Interface de la gestion intelligente en langue Française

## 4.5 Evaluation du système

Nous avons évalué le projet en utilisant quatre critères. Tout d'abord, nous avons évalué les performances du programme. Le temps d'initialisation du programme est de seulement deux secondes. En ce qui concerne la reconnaissance vocale, le temps nécessaire pour convertir la commande vocale en texte est de 5 secondes lorsque la qualité de l'internet est bonne. Cependant, cette action peut prendre plus de temps en cas de faible qualité de connexion. De plus, le temps d'exécution des tâches est compris entre 5 et 6 secondes.

Deuxièmement, nous avons évalué la précision du programme en vérifiant sa capacité à reconnaître et transcrire correctement les commandes vocales. Nous avons constaté que notre programme fonctionne correctement et commet peu d'erreurs en langue Arabe que les autres langues « Anglais et Français ».

Troisièmement, nous avons évalué la robustesse et la fiabilité du programme en le testant dans des environnements acoustiques difficiles, tels que la présence de bruit ou de

personnes qui parlent simultanément. Le programme a montré de bonnes performances dans l'exécution des commandes malgré ces problèmes environnementaux.

Finalement, nous avons également recueilli les commentaires des utilisateurs sur leur satisfaction à l'égard du programme en utilisant une échelle de notation de 1 à 5. Les résultats ont révélé que deux personnes ont attribué la note maximale, trois personnes ont donné la note de 4, et une personne a donné la note de 4,5. Cela représente une notation moyenne de 4,2 (Tableau 4.1).

**Tableau 4-1 :** Évaluation des personnes.

<b>Personnes</b>	<b>1<sup>ère</sup> personne</b>	<b>2<sup>ème</sup> personne</b>	<b>3<sup>ème</sup> personne</b>	<b>4<sup>ème</sup> personne</b>	<b>5<sup>ème</sup> personne</b>	<b>6<sup>ème</sup> personne</b>
Note	5	5	4	4	4	3.5

## 4.6 Conclusion

Dans ce chapitre, nous avons exposé les résultats expérimentaux de notre projet de gestion intelligente des parkings par le biais des commandes vocales. Nous avons commencé par présenter les différents logiciels et langages de programmation que nous avons utilisés. Ensuite, nous avons abordé la manière dont nous avons établi la connexion entre Python et Arduino. Nous avons ensuite détaillé l'implémentation du projet, en expliquant les différentes étapes et les choix techniques que nous avons faits. Enfin, nous avons procédé à l'évaluation du projet, en analysant ses performances et en recueillant les retours des utilisateurs.

Sur la base des résultats obtenus, nous pouvons conclure que le système donne de bons résultats en utilisant un smartphone avec un écouteur Bluetooth, et il est également possible d'éliminer l'utilisation des écouteurs Bluetooth.

Notre système présente les avantages suivants :

- ❖ Il fonctionne dans trois langues : l'Arabe, le Français et l'Anglais, ce qui facilite au l'agent de parking de choisir la langue qui lui convient.
- ❖ Facile à implémenter en temps réel.
- ❖ Adaptabilité aux conditions environnementales.
- ❖ Temps d'exécution court pour les tâches.
- ❖ Ne nécessitant pas beaucoup de dispositifs.

Notre solution présente les inconvénients suivants :

- ❖ Si le décalage temporel entre deux événements impulsifs consécutifs est inférieur à la taille de la séquence d'énergie  $L$ , le système ne sera pas en mesure de les détecter simultanément.
- ❖ Son utilisation nécessite une connexion Internet de haute qualité lors de la reconnaissance vocale.

# Chapitre 5 : Conclusion et travaux futures

## 4.7 Conclusion

L'objectif de ce projet est de mettre en œuvre un système de gestion intelligente des parkings en utilisant des commandes vocales. Notre système est capable de reconnaître en temps réel les commandes vocales émises par l'agent de contrôle et d'effectuer les tâches suivantes : (1) ouvrir et fermer la barrière principale, (2) allumer et éteindre automatiquement l'éclairage du parc, et (3) activer et désactiver l'alarme en cas de vol.

La méthodologie de ce projet se divise en deux étapes principales : la détection du son des claquements de mains et la reconnaissance des commandes vocales. Nous avons utilisé une méthode basée sur le seuillage pour la détection des sons impulsifs et l'API de Google speech recognition pour la reconnaissance des commandes vocales. Ces deux méthodes se sont révélées très efficaces en termes de complexité algorithmique et de ressources nécessaires. Nous avons utilisé Arduino pour exécuter les commandes.

Le chapitre trois de notre étude aborde plus en détail les aspects mentionnés. Le quatrième chapitre est dédié aux résultats expérimentaux et à leur interprétation. Nous avons utilisé plusieurs logiciels et outils, tels que l'éditeur de programmation Jupyter Notebook, Arduino, WO Mic, l'application Lesser AudioSwitch et Qt Designer pour la conception de l'interface utilisateur graphique.

Nous avons évalué les performances de la gestion intelligente des parkings, avec une durée d'initialisation du programme de 2 secondes. Pour la reconnaissance vocale, il fallait 5 secondes pour la reconnaissance de commande vocale et 5 à 6 secondes pour l'exécution des tâches. Nous avons sollicité l'avis d'un panel de six personnes qui ont évalué le projet, et les résultats de l'évaluation ont atteint une échelle de 4,2/5. En ce qui concerne la performance de l'algorithme de détection, nous avons trouvé le meilleur compromis entre fiabilité et performances en utilisant une taille de bloc N de 320, une taille de séquence d'énergie L de 50 et un seuil de détection Th de 0.0000085. Nous avons également proposé un module de réglage de la sensibilité du capteur.

Le projet est conçu pour fonctionner sur un ordinateur portable HP i5 avec le système d'exploitation Windows. Dans notre étude, nous avons utilisé un smartphone avec un

écouteur Bluetooth. Notre méthode présente l'avantage d'être une solution peu complexe, à la fois du côté logiciel et matériel. Elle est facilement adaptable aux conditions de l'environnement et facile à implémenter en temps réel.

## **4.8 Travaux futurs**

**Les travaux futurs consistent à :**

- Il faut développer la partie de reconnaissance des commandes vocales de manière à ce qu'elle ne nécessite pas d'accès à Internet et qu'elle prenne en charge le dialecte algérien plutôt que l'Arabe, l'Anglais et le Français.
- Dans les commandes vocales, nous avons utilisé 3 commandes, mais il est nécessaire d'augmenter le nombre de commandes.
- Pour notre projet, l'utilisation de l'Arduino pour activer ou désactiver (0 et 1) des tâches spécifiques a été réalisée. Pour rendre le projet plus compact et optimiser les tâches, il serait possible de remplacer l'Arduino par les ports séries ou parallèles de l'ordinateur, ou bien par une carte plus robuste comme la carte Controllino.

## Bibliographie

- [1] K. Hassoune, W. Dachry, F. Moutaouakkil and H. Medromi, "Smart parking systems: a survey," *International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1-6, 2016.
- [2] A. Boumergued and S. Nasri, "Conception et réalisation d'une application mobile pour un parking intelligent," Mémoire de Master, Département informatique, Université de Bordj Bou Arréridj, 2020.
- [3] S. Nahmar and I. F. Ferhah, "Interface de communication intelligente entre parkings," Mémoire de Master, Département d'électronique, Université Blida 1, 2022.
- [4] "British Parking Association(BPA)," [Online]. Available: <https://www.britishparking.co.uk/>. [Accessed 15 05 2023].
- [5] R. George, "britannica," 21 juin 2022. [Online]. Available: <https://www.britannica.com/browse/Politics-Law-Government>. [Accessed 10 05 2023].
- [6] S. E. Bensalah , "Optimisation de la gestion d'un parking automatisé," Mémoire de Master, Département d'électronique, Université Badji Mokhtar Annaba, 2019.
- [7] "Lepermislibre," [Online]. Available: <https://www.lepermislibre.fr/code-route/cours/differents- types-parking>. [Accessed 08 05 2023].
- [8] S. Hailja and M. Subhit, "Car parking system based on voice recognition," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 5, pp. 426-430, 2018.
- [9] K. Toshihiro, "Parking IoT system," *US Patents 2013/0339024 A1*, pp. 1-12, 2013.
- [10] A. Amara, A. Shahina, T. Rohit and V. K. Sri Krishna , "An IoT assisted Intelligent Parking System (IPS) for Smart Cities," *Procedia computer science*, vol. 218, pp. 1045-1054, 2023.

- [11] A. V. Anoora Reddy, M. Venkata Sudhakar, K. Mounika, M. V. Sai Kumar and T. Bharani, "Development of smart parking management system," *Matrielstoday:proceedings*, vol. 80, pp. 2794-2798, 2023.
- [12] H. Singh, C. Anand, V. Kumar and A. Sharma, "Automated Parking System With Bluetooth Access," *International Journal of Engineering and Computer Science*, 2014.
- [13] P. DharmaReddy, A. RajeshwarRao and A. Syed Musthak, "An intelligent parking guidance and information system by using image processing technique," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, 2013.
- [14] S. Ahmed Hamada, "Détection des sons impulsifs en vue de la mise en œuvre d'un système de surveillance audio," Mémoire de Master, Département d'électronique, Université Blida 1, 2017.
- [15] F. Ykhlef, S. Ahmed Hamada, F. Ykhlef, A. Derbal et D. Bouchaffra, «Real-Time Detection of Impulsive Sounds for Audio Surveillance Systems,» chez *JERI*, 2019.
- [16] A. Hattali and A. Mousselmal, "Détection et localisation en temps réel des événements sonores en vue de la conception d'un système de surveillance audio," Mémoire de Master, Département d'informatique, Université Blida 1, 2022.
- [17] V. Bhardwaj, M. Tahar Ben Othman, V. Kukreja, Y. Belkhier, M. Bajaj, A. Ur Rehman, M. Shafiq and H. Hamam, "Automatic Speech Recognition (ASR) systems for children a systematic literature review," *Applied Sciences*, p. 26, 2022.
- [18] C. Vimala and V. Radha, "A review on speech recognition challenges and approaches," *World of Computer Science and Information Technology Journal (WCSIT)*, vol. 2, no. 1, pp. 1-7, 2012.
- [19] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.
- [20] W. Jianxin, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology, Nanjing University*, vol. 5, pp. 1-31, 2017.
- [21] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 2020.
- [22] F. Boyer, "Reconnaissance de parole pour le français et intégration dans un système de compréhension du langage parlé," Thèse de Doctorat, Département d'informatique, Université de Bordeaux, 2021.

- [23] B. Iancu, "Evaluating Google Speech-to-Text API's performance for romanian e-learning resources," *Informatica Economica*, vol. 23, pp. 17-25, 2019.
- [24] Ö. Teker, *Profound Python Libraries*, 2022.
- [25] D. Pratik, *Python Programming for Arduino*, Royaume-Uni, Birmingham: Packt Publishing, 2015.
- [26] R. Castilla and M. Pena, "Jupyter notebooks for the study of advanced topics in fluid mechanics," *Computer Applications in Engineering Education*, pp. 1-13, 2023.
- [27] J. Nussey, "Arduino pour les nuls poche," Edi8, France, 2017.
- [28] "uptodown," [Online]. Available: <https://lesser-audioswitch.fr.uptodown.com/android>. [Accessed 17 05 2023].
- [29] "système fman," 2016. [Online]. Available: <https://build-system.fman.io/qt-designer-download>. [Accessed 17 05 2023].