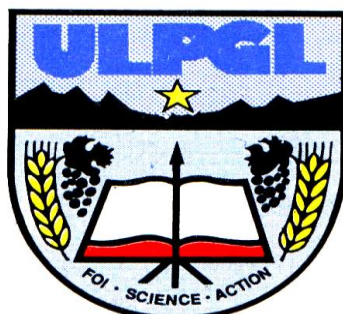


UNIVERSITE LIBRE DES PAYS DES GRANDS LACS

FACULTE DE SCIENCES ET TECHNOLOGIES

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE



BP. 368 GOMA

www.ulpgl.net

**CONCEPTION ET REALISATION D'UN
SYSTEME AUTOMATISE POUR LA
GESTION DU PARKING D'UN AEROPORT**

<< cas de l'aéroport de Goma >>

Par **ASHUZA CIZUNGU Justin**

Travail présenté en vue de l'obtention du Diplôme de
Bachelor en Sciences de l'Ingénieur

Option : Génie Electrique

Directeur : Prof. AKWIR NKIEDIEL Alain

Encadreur : Msc Ir. Daniella MUGHOLE KALIMUMBALO

ANNEE ACADEMIQUE 2024 - 2025



EPIGRAPHE

« L'innovation est ce qui distingue un leader d'un suiveur. »

Steve Jobs





DEDICACE

À mon très cher frère, AKILIMALI CIZUNGU Innocent

ASHUZA CIZUNGU Justin





REMERCIEMENTS

En premier lieu, nous rendons grâce à Dieu pour ses innombrables bienfaits.

Nous exprimons notre profonde reconnaissance à la Faculté des Sciences et Technologies pour la qualité de l'enseignement et de l'encadrement dont nous avons bénéficié tout au long de notre formation académique.

Nous adressons l'expression de notre profonde reconnaissance au **Professeur AKWIR NKIEDIEL Alain**, notre Directeur, pour la rigueur de ses orientations, la richesse de ses observations et l'intérêt constant qu'il a porté à ce travail, contribuant ainsi de manière déterminante à son aboutissement. Nous exprimons également notre haute considération à **Msc Ir Daniella MUGHOLE KALIMUMBALO**, notre Encadreur, pour la qualité de son encadrement, la pertinence de ses suggestions et son accompagnement soutenu tout au long de cette recherche.

Nous exprimons notre respectueuse gratitude à nos très chers parents, CIZUNGU MURHULA Cesaire et MWAVITA MIHIGO, pour leur dévouement, leurs sacrifices consentis et leur soutien indéfectible qui ont constitué le socle de notre parcours académique.

De plus, nous tenons à remercier nos ami(e)s et camarades qui ont été à nos côtés tout au long de ces années, pour leur affection, leur solidarité et leur soutien constant.

Nous adressons enfin notre profonde reconnaissance à nos frères et sœurs Luc, Ephraïm, Du miel, Béatrice, Akili, Archange, Patient, Elodie, Joséphine pour leurs encouragements constants, ainsi qu'à toutes les autres personnes qui, de près ou de loin, nous ont aidés dans la réalisation de ce travail.

ASHUZA CIZUNGU Justin



RESUME

L'évolution technologique permet aujourd'hui d'alléger certaines tâches répétitives et critiques, notamment dans le domaine des transports et de la logistique, où la gestion des flux de véhicules influence directement la fluidité des infrastructures. Dans le cadre de notre travail, nous nous sommes intéressés à la gestion du stationnement au sein des aéroports, en concevant un système automatisé pour le parking de l'aéroport de Goma, capable de gérer les entrées et sorties dans un environnement fortement fréquenté. Ce système doit gérer les réservations, fluidifier le trafic et assurer une attribution fiable des places sans intervention humaine constante.

Pour atteindre cet objectif, nous avons mis en œuvre une architecture basée sur le microcontrôleur ESP32, équipée de capteurs infrarouges pour détecter la présence des véhicules et pilotant des servomoteurs pour automatiser les barrières. Une interface web permet aux usagers de réserver une place et à l'administration de superviser l'occupation du parking en temps réel. Étant donné que cette approche repose sur la précision des capteurs infrarouges (sensibles aux interférences ou à l'éblouissement) et sur une synchronisation continue via le cloud, sa fiabilité peut être ponctuellement affectée dans des environnements extérieurs très imprévisibles.

Mots-clés : *Système automatisé, gestion de parking, ESP32, capteurs infrarouges, Liquid Crystal display, servomoteurs, application web, IoT, réservation en ligne.*



ABSTRACT

Technological advancements today allow for the automation of certain repetitive and critical tasks, particularly in the field of transportation and logistics, where vehicle flow management directly impacts the efficiency of infrastructure. In this work, we focused on airport parking management by designing an automated system for the Goma Airport parking lot, capable of regulating vehicle entry and exit in a highly frequented environment. This system is intended to manage reservations, streamline traffic, and ensure reliable allocation of parking spaces without constant human intervention.

To achieve this objective, we implemented an architecture based on the ESP32 microcontroller, equipped with infrared sensors to detect vehicle presence and controlling servomotors to automate the barriers. A web interface allows users to reserve parking spaces and enables the administration to monitor parking occupancy in real time. Since this approach relies heavily on the precision of infrared sensors (which can be affected by interference or glare) and continuous cloud synchronization, its reliability may be occasionally compromised in highly unpredictable outdoor environments.

Keywords: Automated system, parking management, ESP32, infrared sensors, Liquid Crystal Display, servomotors, web application, IoT, online reservation.



TABLE DES MATIERES

EPIGRAPHE.....	i
DEDICACE	ii
REMERCIEMENTS.....	iii
RESUME	iv
TABLE DES MATIERES	vii
SIGLES ET DES ABREVIATIONS.....	x
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
INTRODUCTION GENERALE	1
0.1. Contexte	1
0.2. Identification et formulation du problème	2
0.3. Questions de recherche	3
0.4. Formulation des hypothèses.....	3
0.5. Justification du choix du sujet et motivations.....	4
0.6. Énoncé des objectifs de recherche	5
0.7. Méthode et délimitation du travail.....	6
0.8. Subdivision du travail	7
Chapitre 1 : REVUE DE LA LITTERATURE ET GENERALITES SUR LES TECHNOLOGIES.....	8
1.1 Introduction partielle.....	8
1.2 Revue de la littérature	8

1.2.1	Revue empirique	Erreur ! Signet non défini.
1.2.2	Revue théorique	8
1.3	Technique du travail (matériels et logiciels).....	35
I.	Matériels	35
1.3.1	La carte programmable Arduino UNO	35
1.3.2	Capteur infrarouge (IR sensor) :	36
1.3.3	Servo Moteur :	36
1.3.4	LCD (<i>Liquid Crystal Display</i>) avec I2C:.....	37
1.3.5	Carte ESP32 :	37
1.3.6	LED :	38
1.3.7	Fils de connexion	38
1.3.8	Plaque d'essai.....	39
II.	Logiciels.....	39
1.3.9	Proteus :	39
1.3.10	Arduino IDE :	40
1.3.11	Visual Studio Code :	40
1.4	Conclusion partielle :	15
Chapitre 2 : CONCEPTION DU SYSTEME DE PARKING AUTOMATISE.....		16
2.1	Introduction partielle.....	16
2.2	Architecture globale du système.....	16
2.2.1	L'unité de traitement et de commande.....	18
2.2.2	L'unité de détection	18
2.2.3	L'unité de stockage	18
2.2.4	L'unité d'authentification	19
2.2.5	L'unité d'affichage.....	19
2.2.6	L'unité d'alimentation	19
2.2.7	L'unité de consultation à distance.....	19
2.3	Schéma bloc du système	Erreur ! Signet non défini.



2.4	Description des composants essentiels	20
2.4.1	Microcontrôleur ESP32	20
2.4.2	Servo Moteur :	23
2.4.3	Capteur infrarouge (IR sensor) :	27
2.4.4	LCD avec I2C :	29
2.5	Application web de gestion du parking	30
2.5.1	Technologies utilisées	32
2.5.2	Schéma de principe et câblage	32
2.6	Conclusion partielle	34
Chapitre 3 : REALISATION DU SYSTEME DE PARKING AUTOMATISE		35
3.1	Introduction partielle.....	35
3.2	Fonctionnement du système.....	40
3.3	Prototype proposé	43
3.4	Application Web	46
3.5	Estimation du coût des matériels	54
3.6	Conclusion partielle	55
CONCLUSION GENERALE		56
BIBLIOGRAPHIE.....		58
ANNEXES.....		61



SIGLES ET DES ABREVIATIONS

- IOT : Internet of things
- IDE : Integrated Development Environment
- IR : Infrarouge
- LCD : Liquid Crustal display
- LED : Light Emitting Diode
- PWM : Pulse-width Modulation
- ULPGL : Université libre des pays des grands lacs
- USB : Universel Serial Bus
- Wi-Fi : Wireless fidelity





LISTE DES TABLEAUX

<i>Tableau 2-1 Caractéristique d'un servomoteur.</i>	<i>23</i>
<i>Tableau 2-2 Caractéristique du capteur IR</i>	<i>27</i>
<i>Tableau 2-3 Caractéristique du LCD</i>	<i>29</i>
<i>Tableau 3-1 Estimation du cout pour la mise en œuvre de ce système.....</i>	<i>54</i>



LISTE DES FIGURES

<i>Figure 1-1 Illustration du parking automatique</i>	12
<i>Figure 1-2 Carte Arduino</i>	36
<i>Figure 1-3 Capteur IR</i>	36
<i>Figure 1-4 Servomoteur</i>	37
<i>Figure 1-5 LCD</i>	37
<i>Figure 1-6 ESP32</i>	38
<i>Figure 1-7 Led</i>	38
<i>Figure 1-8 Fils de connexion</i>	39
<i>Figure 1-9 Plaque d'essai</i>	39
<i>Figure 2-1 architecture du système</i>	17
<i>Figure 2-2 Schéma bloc du système</i>	Erreur ! Signet non défini.
<i>Figure 2-3 schéma bloc du système</i>	Erreur ! Signet non défini.
<i>Figure 2-4 ESP32</i>	20
<i>Figure 2-5 Constitution d'un Servomoteur</i>	24
<i>Figure 2-6 fonctionnement d'un servomoteur</i>	26
<i>Figure 2-7 Schéma bloc d'un capteur</i>	27
<i>Figure 2-8 Principe de fonctionnement d'un capteur IR</i>	28
<i>Figure 2-9 Schéma bloc de LCD</i>	30
<i>Figure 2-10 Schéma du Circuit de la simulation sous proteus</i>	32
<i>Figure 2-11 Schéma de Câblage sous Fritzing</i>	33
<i>Figure 3-1 Diagramme de séquence du système</i>	41
<i>Figure 3-2 Schémas de principe sous Proteus</i>	42
<i>Figure 3-3 prototype physique</i>	43
<i>Figure 3-4 entrer du premier véhicule</i>	44
<i>Figure 3-5 Parking plein</i>	45
<i>Figure 3-6 Parking plein avec réservation</i>	46



<i>Figure 3-7 Page d'inscription.....</i>	<i>47</i>
<i>Figure 3-8 Page d'authentification</i>	<i>48</i>
<i>Figure 3-9 Page d'accueil.....</i>	<i>49</i>
<i>Figure 3-10 Aperçu du Parking</i>	<i>49</i>
<i>Figure 3-11 Page de réservation</i>	<i>50</i>
<i>Figure 3-12 Page de profil.....</i>	<i>51</i>
<i>Figure 3-13 Page d'authentification</i>	<i>52</i>
<i>Figure 3-14 Tableau de bord</i>	<i>53</i>
<i>Figure 3-15 Tableau de bord</i>	<i>53</i>
<i>Figure 3-16 L'ouverture manuelle de la barrière.....</i>	<i>54</i>



INTRODUCTION GENERALE

Le développement rapide des technologies numériques, notamment dans le domaine des systèmes embarqués et de l'Internet des objets (IoT), a profondément transformé notre manière de concevoir les systèmes d'information et d'automatisation. Grâce à ces technologies, il est désormais possible de connecter des objets physiques à Internet afin de collecter, analyser et exploiter des données en temps réel. Cela permet aux dispositifs automatiques d'interagir avec leur environnement, de prendre des décisions autonomes et de fournir des services plus efficaces dans une variété de domaines [1]. Parmi les applications les plus prometteuses de l'IoT figurent la domotique, la gestion industrielle, la santé, la logistique et plus récemment, la gestion des infrastructures urbaines telles que les parkings.[2]

0.1. Contexte

L'urbanisation croissante et l'augmentation exponentielle du nombre de véhicules posent un véritable défi en matière de stationnement dans les villes modernes et les zones à forte activité. Dans les aéroports en particulier, où la mobilité et la fluidité du trafic sont cruciales, la gestion des parkings devient une priorité. Les infrastructures aéroportuaires accueillent quotidiennement un grand nombre de passagers, de visiteurs et de véhicules de service, ce qui peut engendrer une saturation rapide des parkings, des embouteillages à l'entrée, des pertes de temps considérables pour les usagers, ainsi qu'une insatisfaction générale[3].

Dans le cas de l'aéroport international de Goma, situé dans la province du Nord-Kivu en République démocratique du Congo, ces enjeux prennent une dimension encore plus significative. En tant que principal point d'entrée aérien de la région orientale du pays, l'aéroport de Goma connaît une fréquentation croissante, tant sur le plan national qu'international. Toutefois, son infrastructure de stationnement reste encore largement manuelle, peu optimisée, et sujette à de nombreuses inefficacités telles que l'absence de



signalisation sur les places libres, la gestion non automatisée des entrées et sorties, ou encore l'impossibilité de connaître en temps réel l'état du parking.

Face à cette situation, la mise en place d'un système automatisé de gestion du parking s'impose comme une solution innovante et adaptée. Ce système intégrerait des capteurs de détection (ultrasoniques ou infrarouges), des microcontrôleurs programmables tels ESP32, des actionneurs, des interfaces d'affichage en temps réel (écrans LCD), ainsi qu'une application connectée via wifi. Cette application, développée sous une plateforme adaptée, permettrait de consulter à distance l'état d'occupation des places, de réserver une place, et même d'effectuer un paiement en ligne pour réserver une place. L'ensemble du système permettrait ainsi de fluidifier la circulation, d'automatiser les entrées/sorties, et de moderniser efficacement l'infrastructure existante

Ce mémoire s'inscrit donc dans une démarche de modernisation technologique locale en proposant la conception et la réalisation d'un prototype fonctionnel d'un système automatisé de gestion de parking, spécifiquement adapté au contexte de l'aéroport de Goma. Il s'appuie sur les principes de l'IoT et des systèmes embarqués afin d'apporter une réponse concrète à un besoin réel d'optimisation de la gestion du stationnement dans un cadre urbain à forte fréquentation.

0.2. Identification et formulation du problème

0.2.1. Identification du problème

Au cours de notre étude de terrain menée à l'aéroport international de Goma, nous avons constaté plusieurs dysfonctionnements dans la gestion actuelle du stationnement. Le parking souffre régulièrement de situations de saturation, entraînant des files d'attente importantes à l'entrée et des pertes de temps pour les usagers. De nombreux conducteurs ont du mal à localiser rapidement une place libre, ce qui accentue la congestion. Par ailleurs, l'absence de système automatisé de contrôle d'accès rend difficile la gestion efficace des entrées et sorties, et la traçabilité des véhicules stationnés demeure limitée, ce qui peut poser des problèmes de sécurité dans un environnement aussi sensible qu'un aéroport.



0.2.2. Formulation du problème

Nous faisons face à une inadéquation manifeste entre la demande croissante en matière de stationnement à l'aéroport de Goma et les capacités actuelles de gestion, qui reposent encore sur des procédures majoritairement manuelles. Cette situation génère une inefficacité dans la régulation des flux de véhicules, une insatisfaction croissante des usagers, ainsi que des risques accrus en termes de sécurité et de surveillance. De plus, l'absence de système automatique permettant une supervision en temps réel et une interaction numérique avec les usagers limite considérablement les possibilités d'optimisation.

0.3. Questions de recherche

Pour mener à bien nos recherches, nous nous fixons de répondre aux questions suivantes :

- Comment la mise en place d'un système automatisé peut-elle améliorer la gestion du stationnement au parking de l'aéroport de Goma ?
- Quels dispositifs technologiques peuvent être mis en place pour améliorer le contrôle d'accès, la détection des véhicules et l'orientation des usagers dans le parking ?
- Dans quelle mesure l'intégration d'une application mobile connectée peut-elle faciliter la réservation des places, le suivi en temps réel des disponibilités et le paiement des frais de stationnement ?

0.4. Formulation des hypothèses

En se référant aux questions liées à nos recherches, notre travail sera centré sur les hypothèses suivantes :

- La mise en place d'un système de gestion automatisé permettrait d'améliorer l'organisation du parking de l'aéroport de Goma en réduisant les congestions et le désordre liés à la gestion manuelle.



- L'intégration de dispositifs technologiques tels que des capteurs de détection de véhicules, l'écran d'affichage, des barrières automatiques commandées par microcontrôleur et une interface de supervision (application web) contribuerait à améliorer le contrôle d'accès, la détection des véhicules ainsi que l'orientation des usagers à l'intérieur du parking.
- L'intégration d'une application web connectée pourrait permettre aux utilisateurs de consulter en temps réel les places disponibles, de réserver une place à distance et d'effectuer les paiements, améliorant ainsi la rapidité, la sécurité et l'efficacité du service de stationnement.

0.5. Justification du choix du sujet et motivations

0.5.1. Motivation et intérêt pour le sujet

Le choix de ce sujet découle du constat sur le terrain à l'aéroport de Goma, où la gestion manuelle du parking provoque une désorganisation, files d'attente et frustration chez les usagers. En tant qu'étudiants en génie électrique et informatique, passionnés par les systèmes automatiques, nous avons vu dans cette problématique une occasion concrète d'appliquer nos compétences en électronique embarquée, IoT et développement d'applications. Ce projet représente pour nous un défi motivant et une réponse technologique adaptée à un besoin réel de notre environnement.

0.5.2. Pertinence scientifique du sujet

La gestion automatisée des parkings est un thème d'actualité dans la recherche, notamment face à la croissance urbaine et aux défis de mobilité. Plusieurs travaux ont déjà exploré les apports des capteurs IoT, de la communication sans fil et des interfaces web. Notre étude s'inscrit dans cette dynamique en proposant une solution adaptée au contexte local, démontrant comment les technologies automatiques peuvent améliorer concrètement la gestion des services publics en Afrique, même avec des contraintes spécifiques.

0.5.3. Pertinence sociale du sujet





Sur le plan social, la problématique abordée concerne directement le bien-être des citoyens et la qualité des services offerts dans un espace public stratégique : l'aéroport. Une meilleure gestion du parking signifie moins de stress pour les usagers, un gain de temps considérable, une meilleure sécurité et une image modernisée de l'aéroport. La société profite de ce système en ce sens que celui-ci permet la réduction de l'émission du CO₂ qui est un élément majeur du réchauffement climatique. Car en ordonnant le stationnement les véhicules ne font plus beaucoup des tours dans le parking ainsi la quantité du CO₂ dans ce parking est réduit à moitié et la santé des clients est préservée.[4]

0.6. Énoncé des objectifs de recherche

0.6.1. L'objectif général

L'objectif principal de notre travail est de concevoir et de mettre en place un système de stationnement automatisé destiné à améliorer la gestion du parking de l'aéroport international de Goma, en combinant la détection automatique des véhicules, le contrôle d'accès, et une application web pour la réservation.

0.6.2. Les objectifs spécifiques

Dans le cadre de cette recherche, nous visons à :

- Réaliser le montage physique du prototype en intégrant les différents composants électroniques tels que l'ESP32, les capteurs infrarouges, le servomoteur et l'écran LCD ;
- Mettre en œuvre la programmation du microcontrôleur ESP32 afin d'assurer la détection des véhicules, le contrôle de la barrière et la communication avec l'application web ;
- Développer et connecter l'application web au système embarqué pour permettre la consultation, la réservation et la supervision du parking à distance ;
- Effectuer des tests pratiques afin de vérifier le bon fonctionnement, la fiabilité et les performances du système réalisé ;
- Concevoir, simuler et évaluer un prototype fonctionnel intégrant les technologies IoT.



0.7. Méthode et délimitation du travail

0.7.1. Méthode et technique du travail

Dans cette section, nous allons détailler les méthodes de recherche utilisées pour récolter les données, dans le but de bien orienter ce travail vers son approche scientifique autant sur le plan théorique, que sur le plan pratique ; il a été crucial de recourir à une série de méthodes, techniques et outils d'accompagnement aux investigations qui seront décrits dans le chapitre 1 de ce travail.

- **La méthode expérimentale :** C'est une démarche scientifique qui consiste à vérifier la validité d'une hypothèse au moyen d'épreuves répétées, au cours desquelles on modifie un à un les paramètres de situation afin d'observer les effets induits par ces changements[5].
- **La technique documentaire :** qui a permis pour des raisons de recherches, d'exploiter différents ouvrages, travaux, rapports, journaux et articles pour une connaissance approfondie de ce sur quoi nous allons travailler et arriver à terme de notre travail. Cette technique nous a été également d'une grande utilité dans l'élaboration de la structure du travail en s'appuyant également à l'internet ;
- **L'entrevue de recherche :** a permis de recueillir des informations auprès des responsables et les travailleurs de l'aéroport pour comprendre l'organisation du futur système de gestion de stationnement.

0.7.2. Délimitation du travail

Notre travail porte sur la « conception et réalisation d'un système automatisé pour la gestion du parking de l'aéroport international de Goma ». Il se limite à la conception, au prototypage et à la simulation d'un système automatisé capable de gérer les entrées et sorties des véhicules, de détecter en temps réel la présence de voitures dans le parking et d'estimer le nombre de places disponibles. Ce système intègre des capteurs, des modules de communication et une application web destinée aux usagers.



Nous délimitons notre étude selon deux axes principaux :

- **Du point de vue spatial** : L'aéroport international de Goma comprend plusieurs zones d'activités (arrivées, départs, zone administrative, etc.), mais notre étude se concentre uniquement sur la zone de stationnement destinée aux passagers et visiteurs. Les parkings réservés au personnel ou à la logistique ne sont pas inclus dans cette recherche.

0.8. Subdivision du travail

Ce mémoire est structuré en trois chapitres, chacun abordant une dimension spécifique de notre travail de recherche :

- Le premier chapitre sera intitulé « **Revue de la littérature et généralités sur les technologies** ». Ce chapitre présentera les principales recherches et réalisations antérieures en lien avec les systèmes intelligents de gestion de parking. Nous y décrirons les approches utilisées, les technologies mobilisées (capteurs, IoT, systèmes embarqués, applications web), ainsi que les limites identifiées dans différents contextes. Ce chapitre nous permettra d'ancrer notre sujet dans une dynamique scientifique existante et d'identifier notre valeur ajoutée.
- Le deuxième chapitre est intitulé « **Conception du système de parking automatisé** ». Détaillera la démarche de conception de notre solution, en présentant l'architecture matérielle du système de parking automatisé et le flux de travail logiciel utilisant Arduino IDE, Visual Studio Code et Proteus.
- Le troisième chapitre est intitulé « **Réalisation du système de parking automatisé** ». Sera dédié à la présentation des résultats obtenus en relation avec notre question de recherche. Nous y aborderons également les mesures nécessaires pour mettre en place cette solution à l'aéroport de Goma.

Chapitre 1 : REVUE DE LA LITTERATURE ET GENERALITES SUR LES TECHNOLOGIES

1.1 Introduction partielle

La création d'un système automatisé pour gérer un parking est aujourd'hui un sujet important pour beaucoup de villes et de grandes infrastructures. Dans un aéroport, où il faut assurer la fluidité de la circulation, la sécurité et une bonne utilisation de l'espace, l'utilisation de solutions automatisées devient indispensable.

1.2 Revue de la littérature

1.2.1 Revue théorique

Dans cette revue de littérature, nous allons voir ce qui a déjà été fait dans le domaine des parkings automatisés. Nous parlerons des technologies utilisées, des méthodes mises en place et des résultats obtenus ailleurs. Nous présenterons aussi les équipements nécessaires, comme les capteurs, les systèmes électroniques, les modules de communication et les applications web ou mobiles, qui rendent possible une gestion rapide et efficace.

Cette partie nous aidera à comprendre ce qui existe déjà et à choisir les solutions les plus adaptées pour le cas particulier de l'aéroport international de Goma.

- **Conceptions et réalisation d'un système de parking intelligent basé sur la technologie RFID et les capteur IR :** L'auteur de ce travail a proposé un système de parking intelligent afin de réduire le temps perdu dans la recherche des places libres. Pour ce faire, il a conçu et réalisé une maquette basée sur la technologie RFID et les capteurs infrarouges, implémentée sur un microcontrôleur Arduino Méga, permettant l'authentification, la détection des véhicules et l'affichage en temps réel des places disponibles[6].

- **Réalisation d'un parking intelligent** : l'auteur avait soulevé le problème de la congestion urbaine et l'impact écologique de l'émission de gaz à effet de serre ; vu ce problème il s'était fixé l'objectif de la réalisation d'une maquette d'un parking intelligent pour une perspective de l'implémenter dans le cas réel afin de diminuer ces problèmes. Il s'était appuyé sur les réseaux de capteur sans fil avec le système des cartes programmables Arduino ainsi que plusieurs capteurs (ultrason, DHT11) et plusieurs actionneurs (servo moteur, moteur à courant continue).[7]

En ce qui concerne la conception et la réalisation d'un système automatisé pour la gestion du parking de l'aéroport international de Goma, nous proposons d'utiliser différents outils technologiques tels que le module Wi-Fi ESP32, un servomoteur, un écran LCD ainsi que des capteurs infrarouges (IR).

Grâce à ces composants, il sera possible de mettre en place un système automatisé capable de gérer efficacement l'entrée, la sortie et le suivi des véhicules stationnés.

Une barrière motorisée, actionnée par un servomoteur, permettra le contrôle automatisé des accès. Des capteurs IR détecteront la présence des véhicules à l'entrée et à la sortie, ainsi que sur chaque place de stationnement, afin de connaître en temps réel l'état d'occupation du parking. Un écran LCD affichera le nombre de places disponibles, facilitant ainsi l'orientation des conducteurs vers les emplacements libres.

En complément, une application web connectée au système permettra aux usagers de consulter à distance l'état du parking, de localiser les places disponibles, de réserver un emplacement à l'avance et d'effectuer un paiement en ligne. Cette plateforme, accessible via Internet, renforcera l'efficacité du dispositif en offrant une interaction directe entre les utilisateurs et le système automatisé.

Notre approche se distingue des travaux précédemment réalisés, qui se limitaient souvent à des maquettes ou à la gestion simple des entrées et sorties. Dans notre projet, la gestion comprendra à la fois le contrôle des accès, la supervision en temps réel de toutes les places et l'interaction numérique avec les usagers grâce à l'application. Cette solution complète est spécifiquement

adaptée aux besoins d'un environnement aéroportuaire à forte fréquentation, comme celui de l'aéroport international de Goma.

1.2.1.1 Internet des objets

L'Internet des objets est un paradigme assez vaste et admet des acceptions multiples selon les acteurs et les environnements technologiques considérés. Schématiquement, l'internet des objets est une extension de l'Internet actuel à tous les objets pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à Internet. L'utilisation des étiquettes radios (puces RFID) permet, entre autres, une telle extension. Associées à un objet, ces puces peuvent être lues par un dispositif, mobile ou non et des informations relatives à l'objet seront retrouvées grâce à l'Internet [32]. Le but de l'Internet des objets est en effet d'étendre à des entités matérielles ou logicielles, les fonctionnalités offertes par l'Internet dans le domaine de la télécommunication afin de leur permettre d'échanger entre elles ou avec des humains toutes sortes d'informations ou de données. La figure ci-contre nous présente un ensemble d'objets connectés.



Figure 1-1 objets connectés [32]

A. Système automatisé pour la gestion du parking :

Un système automatisé pour la gestion du parking d'un aéroport utilise des technologies avancées afin d'optimiser le contrôle des accès, la surveillance des places disponibles et l'orientation des usagers. Ce type de dispositif permet d'améliorer considérablement l'expérience des passagers et visiteurs tout en rendant la gestion du stationnement plus efficace. Un parking automatisé représente une solution innovante pour gérer efficacement les espaces de stationnement qui tire parti de technologies avancées pour améliorer l'efficacité et l'expérience des utilisateurs[8].

Ce système s'appuie sur des capteurs automatisés qui détectent en temps réel l'occupation des places, ce qui permet aux conducteurs de trouver rapidement un emplacement libre[9].

La connectivité de l'Internet des objets (IoT) facilite la communication entre les différents composants du système, créant ainsi un réseau automatique. Les applications web dédiées offrent aux utilisateurs la possibilité de réserver des places, de payer électroniquement et d'accéder à des informations en temps réel, simplifiant ainsi le processus de stationnement[10]. Les systèmes de guidage dirigent les conducteurs vers les espaces libres, réduisant le temps passé à chercher une place. En utilisant l'analyse des données, le parking intelligent vise à recueillir des informations sur l'utilisation du parking, permettant ainsi une gestion plus efficace des espaces[11].



Figure 1-2 Illustration du parking automatique.[12]

B. Objectifs et avantages

Mettre en place un système automatisé de gestion du parking à l'aéroport international de Goma a plusieurs objectifs et avantages. L'idée principale est de rendre le stationnement plus simple, plus rapide et mieux organisé, tout en améliorant la sécurité et le confort des usagers.

i. Objectifs d'un système de stationnement automatisé

Ce système vise avant tout à faciliter la vie des voyageurs, visiteurs et employés de l'aéroport. Plus concrètement, il permettra de :

- ✓ **Mieux utiliser les places disponibles** : en indiquant en temps réel celles qui sont libres.
- ✓ **Réduire le temps perdu à chercher une place** : car le conducteur sera directement guidé vers un emplacement vide.
- ✓ **Améliorer la circulation** : à l'entrée et à l'intérieur du parking, en évitant les files d'attente et les embouteillages.
- ✓ **Renforcer la sécurité** : grâce à un contrôle automatique des entrées et sorties, ce qui permettra aussi de mieux suivre les véhicules qui se garent.
- ✓ **Protection de l'environnement** : en limitant le temps que les voitures passent à tourner inutilement, et donc en réduisant la pollution.

ii. Avantages

Un tel système apporte aussi de nombreux avantages visibles dans le quotidien des usagers :

- ✓ **Gain de temps** : Les conducteurs trouvent rapidement une place et ne stressent plus à l'idée de tourner longtemps.
- ✓ **Plus de sécurité** : Le système surveille les entrées, sorties et affiche des informations en direct pour éviter les problèmes.
- Meilleure organisation** : Les gestionnaires de l'aéroport auront des données précises sur l'utilisation du parking, ce qui leur permettra de mieux planifier.
- ✓ **Bonne image de l'aéroport** : Une infrastructure moderne et connectée donne confiance aux passagers et aux partenaires.

iii. Les inconvénients

- Le coût élevé de la construction ou de l'installation

Le coût d'avoir un système de gestion du stationnement sain et fonctionnel est généralement élevé. Cela est dû aux différents composants qui entrent dans le fonctionnement du système. Des composants tels que la fonctionnalité statistique, la billetterie automatisée et les rapports statistiques, et bien d'autres, rendent tout cela coûteux. Certaines organisations peuvent ne pas être en mesure de se le permettre.

- Entretien régulier

Le système est automatisé ; cependant, il nécessite encore plusieurs entretiens réguliers. Il s'agit de s'assurer que le système fonctionne parfaitement. L'entretien pourrait être une fois tous les mois.

- Fonctionnement

Beaucoup de gens ne sont pas habitués au système automatisé. En conséquence, il peut être difficile pour eux de faire usage, provoquant ainsi d'autres complications pendant le stationnement

C. Fonctionnement

Le fonctionnement d'un système automatisé pour la gestion du parking de l'aéroport de Goma repose sur un ensemble d'éléments qui travaillent ensemble pour rendre le stationnement simple et rapide.

- ❖ **Capteurs de détection (IR) :** Chaque place de parking est équipée d'un capteur qui détecte si une voiture est présente ou non. Les informations sont envoyées automatiquement vers le système et affichées sur un écran placé à l'entrée du parking. Ainsi, les conducteurs savent immédiatement quelles places sont libres avant même d'entrer.
- ❖ **Collecte et affichage des données :** Les données collectées par les capteurs sont mises à jour en temps réel et présentées de manière claire sur un écran ou à travers l'application web. Cela évite aux conducteurs de tourner inutilement à la recherche d'une place, ce qui fait gagner du temps et réduit les bouchons dans le parking.
- ❖ **Système de guidage :** En plus des écrans à l'entrée, des panneaux ou indicateurs lumineux peuvent guider les conducteurs vers les emplacements libres. L'application permet aussi de vérifier à distance s'il reste de la place avant même d'arriver à l'aéroport, ce qui est très utile pour les passagers pressés.
- ❖ **Intégration avec d'autres services :** Le système peut être relié à d'autres infrastructures de l'aéroport, comme la billetterie ou les services de réservation. Cela permet une meilleure organisation globale et aide à fluidifier les mouvements des véhicules, surtout aux heures de pointe.



1.3 Conclusion partielle :

Ce premier chapitre a permis d'établir les fondements théoriques de notre travail en explorant les fondements des systèmes de parking automatisés. Nous avons défini les concepts clés liés à la gestion automatisée du stationnement, examiné les différentes technologies utilisées et mis en évidence les outils électroniques et logiciels fondamentaux qui animent ce domaine, de la carte ESP32 aux applications IoT et aux applications web.

Ce cadre théorique étant posé, le chapitre suivant se concentrera sur *la conception du système de parking automatisé*. Nous détaillerons le choix des composants, la conception structurelle et fonctionnelle des installations et l'architecture complète du système électronique. L'objectif sera de passer de la théorie à la pratique et de préparer la phase de réalisation de notre prototype fonctionnel.



Chapitre 2: CONCEPTION DU SYSTEME DE PARKING AUTOMATISE

2.1 Introduction partielle

Les systèmes automatisés aident aujourd'hui à mieux gérer les équipements publics. Ces systèmes rassemblent différentes technologies pour réaliser des fonctions automatiques qui améliorent les services. Notre projet réunit des composants électroniques et une application web. Tous ces éléments travaillent ensemble pour créer une solution pratique qui facilite le stationnement. Cette conception répond directement aux problèmes observés sur le terrain.

2.2 Architecture globale du système

Notre système de gestion de parking est organisé autour de trois systèmes principaux qui travaillent ensemble :

Les trois unités de base :

- L'unité qui traite et commande (le cerveau du système)
- Système de détection (les yeux du système)
- Système de stockage (la mémoire du système)

Les unités complémentaires :

- Système motorisé (pour gérer les entrées)
- Système d'affichage (pour informer les usagers)
- Système de signalisation
- Système d'authentification

Le système fonctionne comme un ensemble coordonné où chaque partie a un rôle précis. L'unité de détection repère les véhicules, l'unité de traitement analyse la situation et prend les décisions, puis l'unité de stockage garde en mémoire toutes les informations. Les autres unités aident à contrôler l'accès et à communiquer avec les utilisateurs.

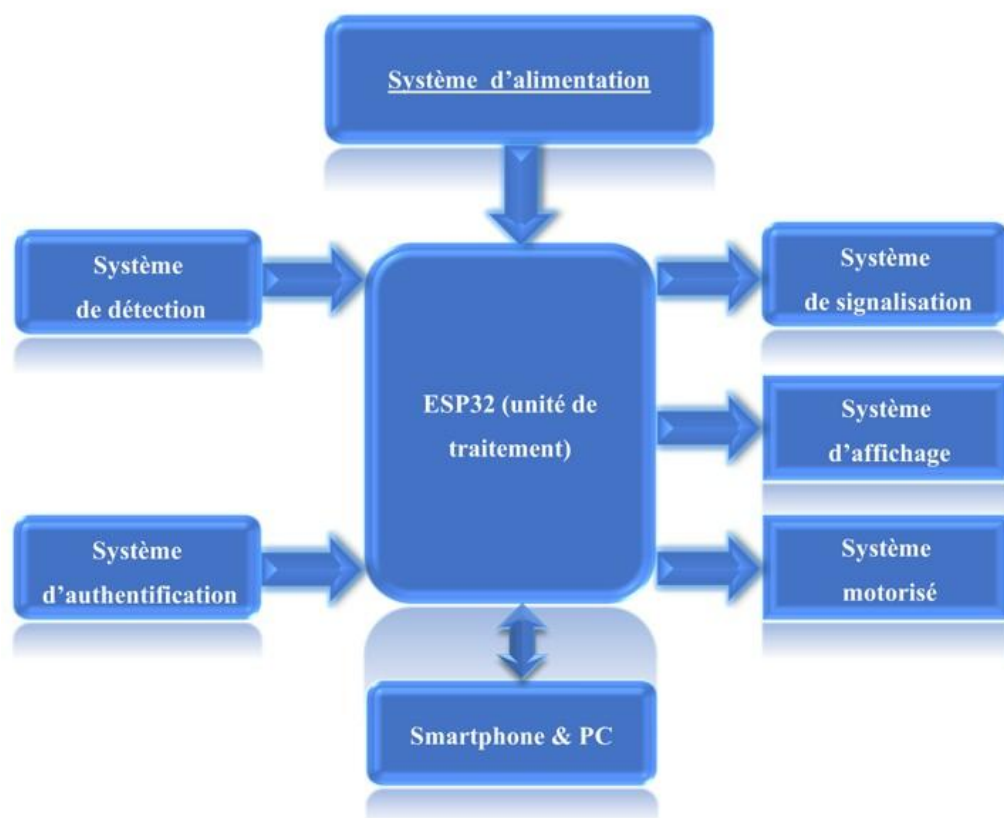


Figure 2-1 architecture du système

2.2.1 L'unité de traitement et de commande

Cette unité constitue le cœur du système. Elle est représentée par la carte Arduino et EP32 qui assure la coordination et le contrôle de toutes les autres unités. Elle reçoit les signaux provenant des capteurs, les analyse et exécute les actions correspondantes : allumer ou éteindre les LED, mettre à jour la base de données, activer la barrière d'accès, ou encore envoyer les informations à l'écran d'affichage et à l'application web. En d'autres termes, elle agit comme le cerveau du système, garantissant une communication fluide entre les différents modules du parking automatisé.

2.2.2 Système de détection

Système de détection est constitué essentiellement des capteurs infrarouges (IR) placés au niveau de chaque emplacement de stationnement ainsi qu'aux points stratégiques du parking. Son rôle principal est de détecter la présence ou l'absence d'un véhicule sur une place donnée.

2.2.3 Système de stockage

Ce Système agit comme la mémoire du système. il correspond à la base de données dans laquelle sont enregistrées toutes les informations relatives au parking, notamment :

- Le nombre total et le nombre de places disponibles,
- L'historique des entrées et sorties de véhicules,
- Les informations d'identification des usagers.

Il permet au système de stocker, d'extraire et de mettre à jour les données en temps réel pour assurer un fonctionnement optimal et une traçabilité complète des activités du parking.



2.2.4 Système d'authentification

Système d'authentification est chargé de contrôler l'accès des véhicules au parking en vérifiant l'identité des usagers à l'aide d'un code d'accès. Il constitue un élément essentiel de la sécurité du système, car il permet de s'assurer que seuls les usagers autorisés peuvent entrer dans le parking.

2.2.5 Système d'affichage

Ce système est chargé de communiquer les informations essentielles aux usagers et aux gestionnaires du parking. Il regroupe plusieurs dispositifs permettant de visualiser clairement l'état du parking, à savoir :

- Un écran LCD intégré au prototype, qui affiche en temps réel le nombre de places libres et occupées ;
- Une application web permettant de consulter à distance, depuis un smartphone ou un ordinateur, la disponibilité des places.

Ces dispositifs d'affichage facilitent la gestion du parking et permettent aux conducteurs d'obtenir rapidement les informations nécessaires avant même d'entrer sur le site.

2.2.6 Système d'alimentation

Ce système d'alimentation assure l'apport énergétique nécessaire au bon fonctionnement de l'ensemble du système de stationnement automatisé. Il joue un rôle fondamental, car sans une alimentation stable et adaptée, les différentes unités ne pourraient pas fonctionner correctement.

2.2.7 L'unité de consultation à distance

L'unité de consultation à distance permet aux usagers et aux gestionnaires du parking d'accéder aux informations du système sans être physiquement présents sur le site. Elle repose principalement sur l'application web connectée au serveur backend via Internet.



2.3 Description des composants essentiels

2.3.1 Microcontrôleur ESP32

L'ESP32 est un système sur puce (SoC) intégrant les technologies Wi-Fi 2,4 GHz et Bluetooth, conçu avec la technologie TSMC 40 nm basse consommation. Il offre d'excellentes performances énergétiques et radioélectriques, tout en garantissant robustesse, fiabilité et polyvalence pour les applications embarquées et IoT.[21]

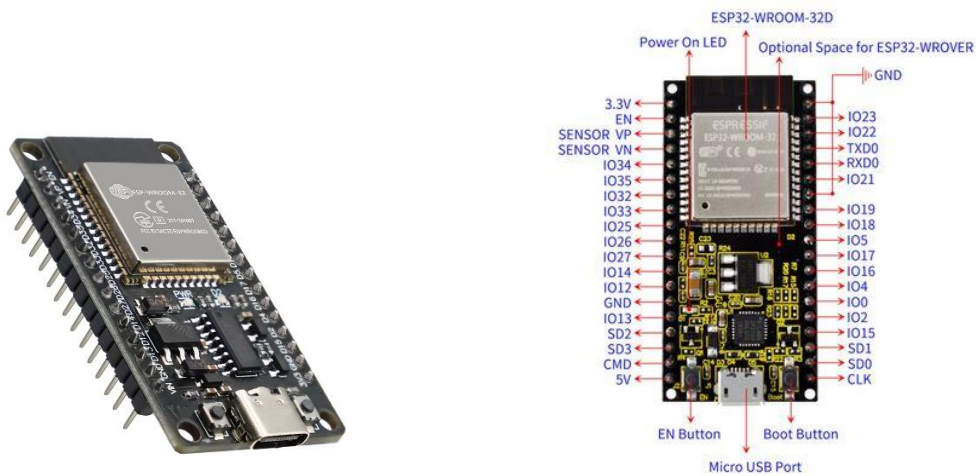


Figure 2-2 ESP32.[22]

a. Rôle

Il assure trois fonctions essentielles :

- La centralisation des données issues des capteurs infrarouges installés sur chaque emplacement ainsi qu'aux points d'entrée et de sortie du parking ;
- La communication sans fil avec l'application web grâce au module Wi-Fi intégré, permettant l'affichage en temps réel de l'état du parking ;
- Commande des actionneurs : contrôle la barrière motorisée, les LEDs de signalisation et autres périphériques.

**b. Caractéristiques techniques****Connectivité sans fil :**

- Wi-Fi: 802.11 b/g/n:
 - 802.11b est une norme plus ancienne avec une vitesse maximale de 11 Mbps.
 - 802.11g offre jusqu'à 54 Mbps
 - 802.11n est plus rapide, avec des débits allant jusqu'à 600 Mbps, et fonctionne à la fois sur les bandes de 2,4 GHz et 5 GHz.
- Bluetooth : v 4.2 BR/EDR et BLE jusqu'à v 5.0 et v 5.1 ;

Processeurs :

- CPU : Xtensa double-cœur (ou simple-cœur), microprocesseur LX 32 bits, fonctionnant à 160 ou 240 MHz et fournissant jusqu'à 600 DMIPS ;
- Coprocesseur ultra basse consommation (ULP) ;
- Mémoire : 32 Mo SRAM ;

Gestion énergétique :

- Alimentation :
 - 5V via le port USB ou la broche VIN
 - Tension interne de fonctionnement : 3,3 V
- Consommation : ~80 mA en activité, < 20 μ A en veille profonde
- Fonctionnement 24h/24 avec optimisation énergétique





Interfaces de périphériques

L'ESP32 utilisé dans ce projet est une carte de développement à 38 broches, offrant une grande flexibilité pour l'interfaçage avec les capteurs et les actionneurs. Il dispose notamment de :

- 38 broches physiques, dont jusqu'à 34 broches GPIO programmables ;
- Plusieurs broches dédiées à l'alimentation (VIN, 3,3 V, 5V, GND) ;
- 3 interfaces UART, utilisées pour la communication série avec des modules externes (RFID, paiement, débogage) ;
- 4 interfaces SPI pour l'échange rapide de données ;
- 2 interfaces I²C, notamment pour la communication avec l'écran LCD ;
- ADC 12 bits jusqu'à 18 canaux, permettant la lecture des signaux analogiques ;
- 2 DAC 8 bits, utiles pour la génération de signaux analogiques ;
- Des sorties PWM matérielles pour le contrôle de moteurs et de LEDs.

c. Sécurité :

- Standard de sécurité supportant complètement IEEE 802.11, incluant WPA/WPA2 et WAPI de WFA ;
- Secure boot (démarrage sécurisé) ;
- Chiffrement de la Flash ;
- 1024-bit OTP, jusqu'à 768 bits pour les clients ;
- Accélération matérielle du chiffrement : AES, SHA-2, RSA, Cryptographie sur les courbes elliptiques (ECC), Générateur de nombres aléatoires (RNG).

d. Gestion optimisée de l'énergie :



- Consommation ultra-basse en mode veille adaptée aux périodes de faible activité ;
- Réveil automatique par interruption GPIO ou minuterie intégrée ;
- Fonctionnement continu 24h/24h, indispensable pour une infrastructure aéroportuaire.

2.3.2 Servo Moteur :

Servomoteur peut être défini comme un dispositif électromécanique utilisé pour contrôler précisément la position angulaire, la vitesse ou le couple d'un mécanisme. Il est généralement composé d'un moteur (souvent un moteur électrique), d'un capteur de position, et d'un circuit de commande, le tout réuni dans un même boîtier[17].

a. Caractéristiques techniques du servomoteur

Tableau 2-1 Caractéristique d'un servomoteur.

Paramètre	Valeur/Description
Modèle	SG90
Poids	9g
Tension d'alimentation	4,5V – 5V
Connecteur	Marron (GND), Rouge (VCC), Orange (Signal)
Consommation	3,7 mA au repos - 170 et 250 mA en rotation
Signal de commande	PWM (1 à 2 ms toutes les 20 ms)
Position 180°	2,5ms
Position 0°	0.5ms

b. Constitution d'un Servomoteur

La constitution d'un servomoteur comprend plusieurs éléments essentiels :

- Moteur électrique ;
- Réducteur (Engrenages) ;
- Capteur de position (Potentiomètre) ;
- Amplificateur de puissance ;
- Boîtier et arbre de sortie ;
- Électronique de commande ou contrôle.

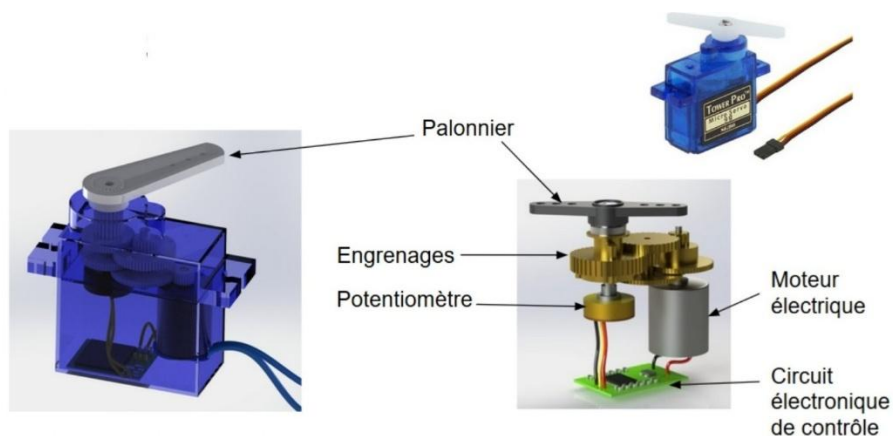


Figure 2-3 Constitution d'un Servomoteur[30].

La description des composants d'un servomoteur[17] :

- **Moteur électrique**



Dans un servomoteur, c'est lui qui génère la puissance nécessaire pour produire le mouvement rotatif ou linéaire du mécanisme contrôlé.

– **Réducteur de vitesse**

Comme son nom l'indique, le réducteur de vitesse réduit la vitesse de rotation du moteur tout en augmentant le couple. Il permet d'obtenir des mouvements plus lents et plus puissants, essentiels pour certaines applications.

– **Encodeur ou potentiomètre**

Il s'agit du capteur de position qui mesure la position actuelle (en temps réel) de l'axe du moteur. Il fournit un retour d'information au système de commande pour comparer la position réelle à celle souhaitée, assurant ainsi un positionnement précis.

– **Contrôleur**

C'est l'unité de traitement qui reçoit le signal de commande (la consigne) et le compare à la position actuelle du moteur, ajustant la puissance transmise au moteur pour corriger toute déviation et atteindre la position cible.

– **Amplificateur de puissance**

Ce composant amplifie le signal envoyé par le contrôleur pour le transmettre au moteur, fournissant la puissance nécessaire pour effectuer les mouvements demandés.

– **Source d'alimentation**

Fournit l'énergie électrique nécessaire pour faire fonctionner le moteur et les autres composants électroniques du servomoteur.

c. Principe de fonctionnement d'un servomoteur

La plupart des servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. [31]



Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur.

Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

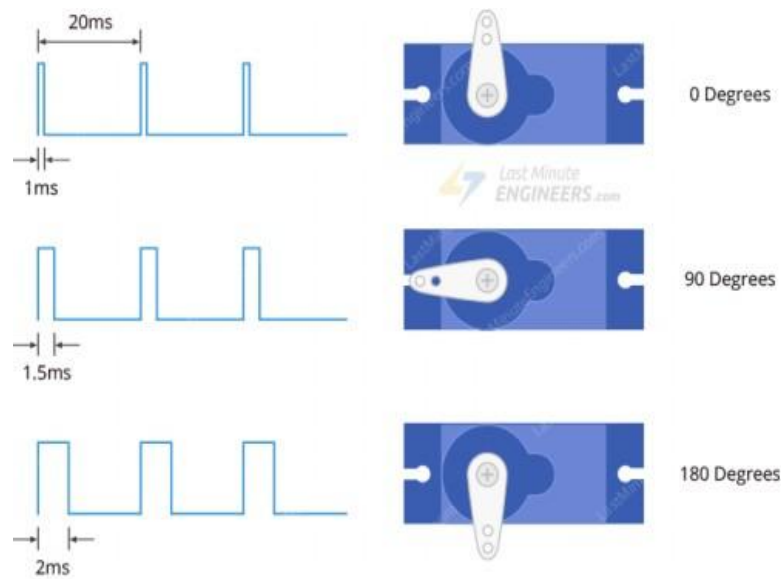


Figure 2-4 fonctionnement d'un servomoteur.[31]

- Si l'impulsion est élevée pendant 1 ms, l'angle d'asservissement sera nul.
- Si l'impulsion est élevée pendant 1,5 ms, alors le servo sera à sa position centrale.
- Si l'impulsion est élevée pendant 2 ms, le servo sera à 180 degrés.
- Des impulsions comprises entre 1 ms et 2 ms déplaceront l'arbre de servo sur les 180 degrés de sa course.

2.3.3 Capteur infrarouge (IR sensor) :

Un est un dispositif capable de transformer une grandeur physique (telle que le mouvement et la lumière.) en une autre grandeur physique manipulable.[32]

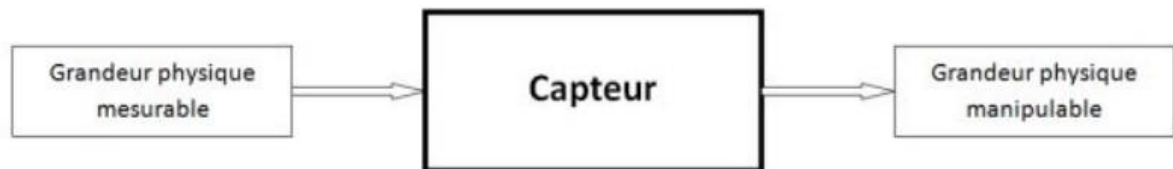


Figure 2-5 Schéma bloc d'un capteur.[32]

a. Caractéristiques du détecteur

Tableau 2-2 Caractéristique du capteur IR

Paramètre	Valeur
Alimentation	3,3V – 5V DC
Distance de détection	2 à 30 cm selon réglage
Sortie	Numérique (0 ou 1)
Consommation de Courant	20mA

b. Principe de fonctionnement

Le capteur infrarouge fonctionne sur le principe d'échange d'un rayonnement lumineux invisible à l'œil humain. Il est constitué principalement de deux éléments : un émetteur infrarouge et un récepteur infrarouge.[32]

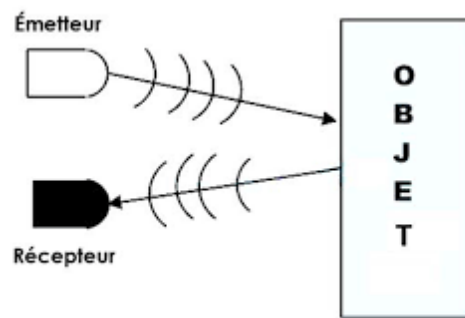


Figure 2-6 Principe de fonctionnement d'un capteur IR.

– **L'émetteur infrarouge**

Le capteur commence par utiliser son émetteur, qui est en réalité une petite LED infrarouge. Quand elle s'allume, elle envoie un faisceau de lumière infrarouge invisible à l'œil humain. Cette lumière part droit devant elle, vers l'objet ou la surface à détecter.

– **Réflexion du rayonnement sur l'objet**

Lorsque le faisceau infrarouge émis par la LED rencontre un objet, une partie de cette lumière est réfléchi. Le comportement de la réflexion dépend de plusieurs facteurs : la distance de l'objet, sa couleur, sa texture et son pouvoir réfléchissant.

– **Le récepteur infrarouge**

Le récepteur, généralement une photodiode ou un phototransistor, est positionné de manière à recevoir le rayonnement réfléchi. Il convertit cette lumière incidente en un signal électrique proportionnel à l'intensité de la lumière reçue.

2.3.4 LCD avec I2C :

Est un dispositif d'affichage utilisant la propriété optique des cristaux liquides pour afficher du contenu. Il est souvent employé pour fournir des informations visuelles dans les projets électroniques.[33]

a. Caractéristiques

Tableau 2-3 Caractéristique du LCD

Paramètre	Valeur
<i>Liquid Crystal Display</i>	
Alimentation	4.5V~5.5V DC
Consommation en courant	2.0mA
20 colonnes pour 4 lignes	20x4 caractères
Taille d'un caractère	2.95mm x 4.35mm (L x l)
<i>Module I2C (PCF8574)</i>	
Données	SDA
Horloge	SCL
Adresse I2C	0x27 (souvent) ou 0x3F
Broches disponibles	VCC, GND, SDA, SCL

b. Principe de fonctionnement

Un LCD 4x20 est un afficheur capable de montrer 4 lignes de 20 caractères.

Normalement, il utilise 8 ou 4 fils de données, mais avec un module I2C (PCF8574) fixé derrière, la communication se fait seulement par 2 fils : SDA et SCL.[32]

Principe :

- Le microcontrôleur (Arduino) envoie des commandes sous forme de bits via le protocole *I2C*.
- Le module *PCF8574* convertit ces informations et les transmet au LCD.
- Le LCD affiche les caractères grâce à son contrôleur interne *HD44780*.
- Le protocole I2C permet d'adresser l'écran à l'aide d'une *adresse* (souvent *0x27* ou *0x3F*).

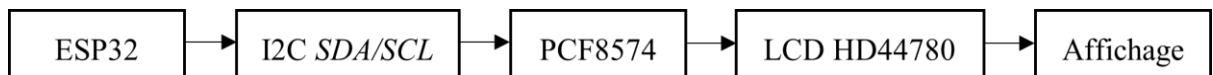


Figure 2-7 Schéma bloc de LCD.

2.4 Application web de gestion du parking

L'application web constitue l'interface principale entre les usagers et le système de parking automatisé. Elle permet de consulter en temps réel l'état du parking, notamment le nombre de places disponibles et occupées, ainsi que d'effectuer des réservations à distance. Elle intègre également une fonctionnalité de paiement en ligne, facilitant le règlement des frais de stationnement et améliorant la fluidité du service.



Pour les gestionnaires, l'application offre une plateforme de supervision permettant de suivre l'occupation du parking, l'historique des réservations et l'état général du système, contribuant ainsi à une meilleure gestion des ressources.

Cahier des charges de l'Application Web

Le volet logiciel de notre solution, appelé « AeroPark », doit être capable de gérer et de centraliser toutes les interactions numériques entre les usagers, l'administration de l'aéroport et le parking physique. Il permettra donc :

- À l'utilisateur (client), de créer un compte sécurisé, de consulter en temps réel la disponibilité des places de parking via une cartographie interactive, et d'effectuer une réservation en ligne pour une durée déterminée.
- Au système, de générer un code d'accès unique après validation de la réservation et du paiement, permettant au client de s'identifier à l'entrée du parking.
- De lire, de traiter et d'afficher instantanément les changements d'état du parking (Libre, Occupé, Réserve) grâce à une communication en temps réel avec les capteurs du site.
- À l'administrateur, d'accéder à un tableau de bord sécurisé pour : superviser l'occupation globale, gérer les utilisateurs inscrits, suivre l'historique des réservations et des paiements.
- À l'administrateur, d'enregistrer manuellement les clients (qui se présentent sans réservation) de manière purement administrative, sans interférer avec la détection physique des capteurs.
- À l'administrateur, de déclencher une ouverture d'urgence de la barrière à distance via une commande manuelle depuis l'interface web.



2.4.1 Technologies utilisées

L'application repose sur une architecture client-serveur. Le backend est développé avec FastAPI (Python) et communique avec l'ESP32. Les données sont stockées dans Firebase Firestore, l'authentification est assurée par Firebase Authentification, et la communication en temps réel est réalisée via WebSocket. Le frontend est développé en HTML, CSS et JavaScript.

2.4.2 Schéma de principe et câblage

- La Figure 2-10 présente le schéma électronique du système de parking automatisé simulé sous Proteus. Elle montre l'interconnexion entre l'Arduino UNO, les capteurs infrarouges de détection des places et du passage des véhicules, l'écran LCD d'affichage et le servomoteur de commande de la barrière, permettant ainsi d'assurer la gestion automatique des entrées, des sorties et de l'occupation des places.

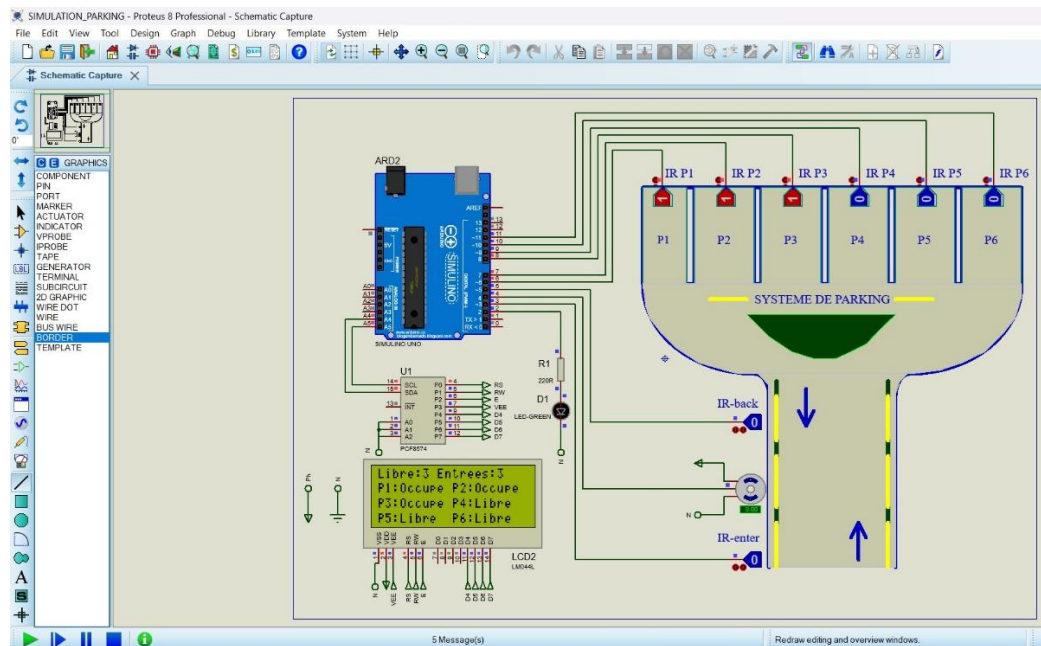


Figure 2-8 Schéma du Circuit de la simulation sous proteus.

- La Figure 2-11 présente le schéma de câblage du système de parking automatisé. L'alimentation 220 V est convertie en 5 V continu pour alimenter l'Arduino UNO et les modules. Des détecteurs infrarouges détectent la présence des véhicules aux emplacements ainsi qu'à l'entrée et à la sortie. L'Arduino traite ces informations pour déterminer l'état des places. Un écran LCD affiche le nombre de places disponibles et un servomoteur commande l'ouverture et la fermeture de la barrière.

La figure 2-11 illustre ainsi l'architecture globale partie matérielle du système de gestion automatique du parking.

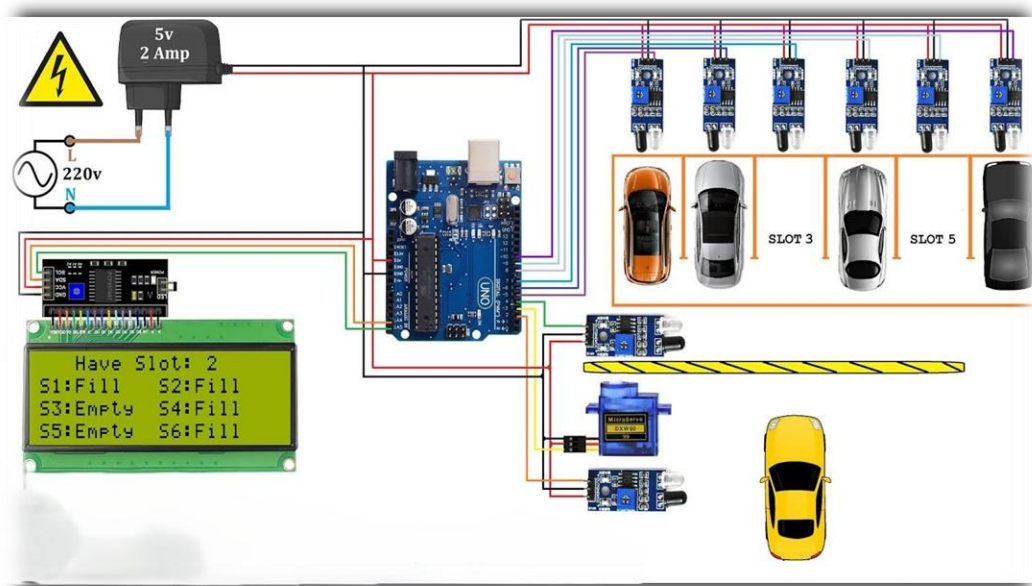


Figure 2-9 Schéma de Câblage sous Fritzing.[34]



2.5 Conclusion partielle

Ce chapitre a été consacré à la conception du système. Il nous a permis de définir l'architecture générale du système ainsi que le rôle de chacun de ses composants, depuis les dispositifs de détection jusqu'à l'unité de traitement et aux éléments de commande.

Les principaux choix techniques ont été orientés vers la simplicité, la fiabilité et la facilité d'intégration. Les capteurs infrarouges assurent la détection des véhicules, tandis que l'ESP32 joue le rôle d'unité centrale de traitement et de communication. Le servomoteur permet le contrôle automatique de la barrière et l'écran LCD informe les usagers sur la disponibilité des places. Enfin, la connectivité sans fil garantit l'échange des données avec l'application web, permettant une consultation à distance de l'état du parking.

Ainsi, ce travail de conception constitue une base solide pour la réalisation du prototype. Le chapitre suivant sera consacré à la réalisation concrète du système, en abordant les étapes de montage, de câblage, de programmation et de test, afin de vérifier le bon fonctionnement du système dans des conditions proches de la réalité.



Chapitre 3: REALISATION DU SYSTEME DE PARKING AUTOMATISE

3.1 Introduction partielle

Dans ce chapitre nous passons à l'étude de la réalisation du projet. Cette étape consiste à mettre en œuvre concrètement les éléments matériels et logiciels afin d'évaluer le fonctionnement du système, sa faisabilité technique ainsi que sa rentabilité dans un contexte réel, notamment celui de l'aéroport de Goma.

3.2 Matériels et logiciels

Le succès de ce projet repose sur l'intégration harmonieuse de plusieurs technologies complémentaires, allant de la conception logicielle au contrôle matériel.

I. Matériels

3.2.1 La carte programmable Arduino UNO

L'Arduino Uno est une carte électronique programmable qui sert à contrôler différents dispositifs comme des moteurs, des voyants lumineux, des capteurs ou d'autres systèmes automatisés. Elle est basée sur un microcontrôleur ATmega328 fonctionnant à une fréquence de 16 MHz. La carte dispose de 14 broches d'entrée/sortie numériques (dont 6 peuvent envoyer des signaux PWM pour contrôler par exemple la vitesse d'un moteur), de 6 entrées analogiques, d'une connexion USB, d'une prise d'alimentation, d'un bouton de réinitialisation et d'un résonateur céramique à 16 MHz. Elle contient tout ce qu'il faut pour faire fonctionner le microcontrôleur : il suffit de la brancher à un ordinateur via un câble USB ou de l'alimenter avec un adaptateur secteur ou une batterie pour commencer à l'utiliser.[13]



Figure 3-1 Carte Arduino. [14]

3.2.2 Capteur infrarouge (IR sensor) :

Un capteur infrarouge est un instrument électronique utilisé pour détecter certaines caractéristiques de son environnement en émettant ou en détectant un rayonnement infrarouge. Les capteurs infrarouges mesurent ou observent également la chaleur d'un objet et détectent son mouvement. Un circuit de capteur infrarouge est l'un des modules de base les plus courants des appareils électroniques.[15]

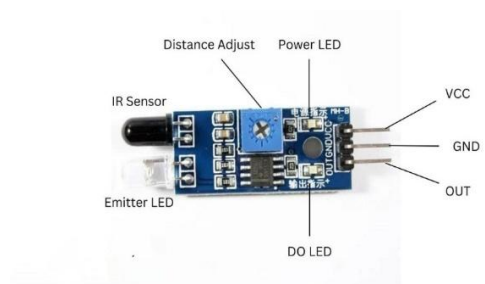


Figure 3-2 Capteur IR.[16]

3.2.3 Servo Moteur :

Servomoteur peut être défini comme un dispositif électromécanique utilisé pour contrôler précisément la position angulaire, la vitesse ou le couple d'un mécanisme. Il est généralement

composé d'un moteur (souvent un moteur électrique), d'un capteur de position, et d'un circuit de commande, le tout réuni dans un même boîtier.[17]

Le servomoteur fonctionne en boucle fermée : il compare en temps réel la position actuelle avec la position souhaitée, puis ajuste la commande du moteur pour corriger toute différence. On parle donc d'un système asservi en position.[17]



Figure 3-3 Servomoteur.[18]

3.2.4 LCD (Liquid Crystal Display) avec I2C:

Est un dispositif d'affichage utilisant la propriété optique des cristaux liquides pour afficher du contenu. Il est souvent employé pour fournir des informations visuelles dans les projets électroniques.[19]

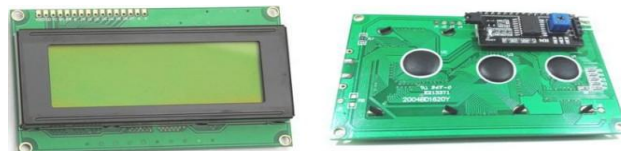


Figure 3-4 LCD.[20]

3.2.5 Carte ESP32 :

L'ESP32 est un système sur puce (SoC) intégrant les technologies Wi-Fi 2,4 GHz et Bluetooth, conçu avec la technologie TSMC 40 nm basse consommation. Il offre d'excellentes performances énergétiques et radioélectriques, tout en garantissant robustesse, fiabilité et polyvalence pour les applications embarquées et IoT.[21]



Figure 3-5 ESP32.[22]

3.2.6 LED :

La LED est un composant électronique qui produit de la lumière lorsqu'elle reçoit un courant électrique. Elle est très utilisée comme voyant lumineux dans différents projets électroniques, pour indiquer par exemple l'état de fonctionnement d'un système ou signaler une information particulière.



Figure 3-6 Led.[23]

3.2.7 Fils de connexion

Ce sont des fils que nous utilisons pour réaliser les connexions de différents composants lors de la réalisation du prototype de notre système. On en trouve généralement à des couleurs différentes.



Figure 3-7 Fils de connexion[24]

3.2.8 Plaque d'essai

La plaque d'essai, sert à accueillir des composants électroniques qui peuvent être reliés entre eux à l'aide de câbles flexibles. Elle est utilisée par les professionnels pour tester et ajuster le fonctionnement des circuits avant leur production en série sur des cartes imprimées.[25]

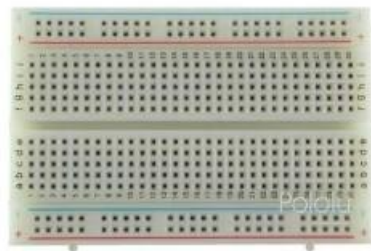


Figure 3-8 Plaque d'essai[26]

I. Logiciels

3.2.9 Proteus :

Proteus est un logiciel de conception assistée par ordinateur (CAO) qui permet de créer des schémas électroniques, de simuler leur fonctionnement et de tester virtuellement les circuits avant leur réalisation physique.[27]

Dans le cadre d'un projet comme la gestion automatisée du parking, Proteus est utilisé pour simuler le câblage des capteurs, de l'Arduino et des autres composants afin de vérifier le bon fonctionnement du système avant le montage réel.

3.2.10 Arduino IDE :

L'Arduino IDE (Integrated Development Environment) est un logiciel qui permet d'écrire, compiler et envoyer des programmes (appelés *sketches*) vers les microcontrôleurs. Il dispose d'une interface simple qui inclut un éditeur de texte, un compilateur et un outil de communication via USB.[28]

Ce logiciel est essentiel pour programmer et tester le système automatisé de gestion de parking, car il permet de contrôler les capteurs, l'actionneur et l'affichage en temps réel.

3.2.11 Visual Studio Code :

Visual Studio Code est un éditeur de code source gratuit et open source développée par Microsoft, qui facilite le développement et l'optimisation des projets informatiques.[29]

Il permet aux développeurs d'écrire, d'éditer et de déboguer du code dans de nombreux langages de programmation comme Python, JavaScript, C++, Java, et bien d'autres.

3.3 Fonctionnement du système

Afin de comprendre la dynamique du système, nous présentons le diagramme de séquence global. Ce diagramme a pour objectif de visualiser l'enchaînement chronologique des interactions entre les trois entités majeures de notre solution :

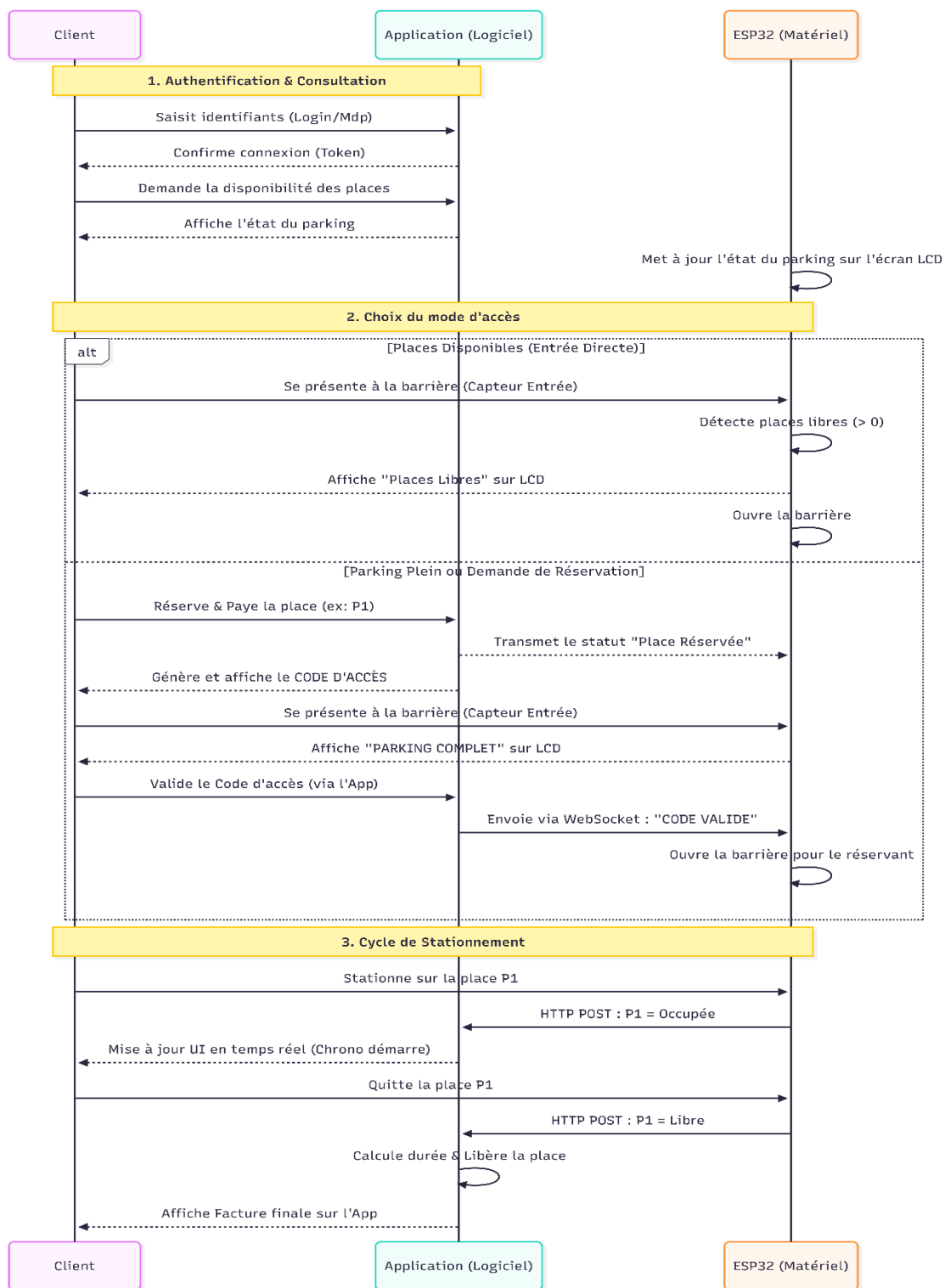


Figure 3-9 Diagramme de séquence du système.

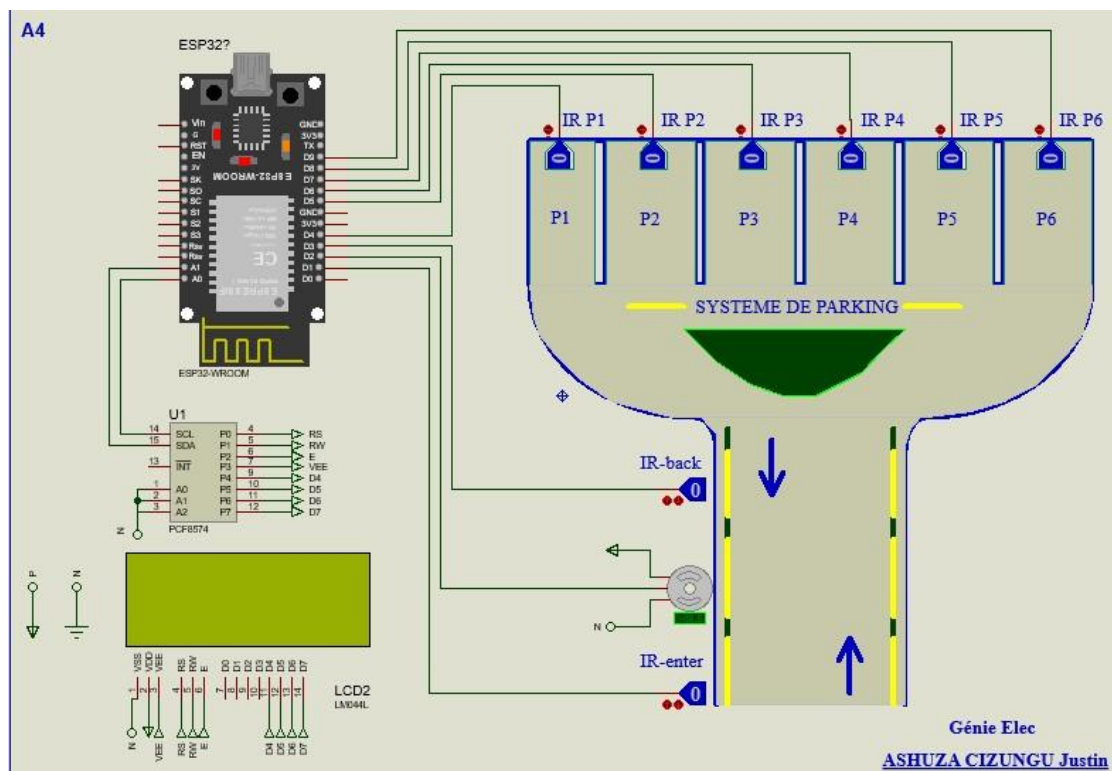
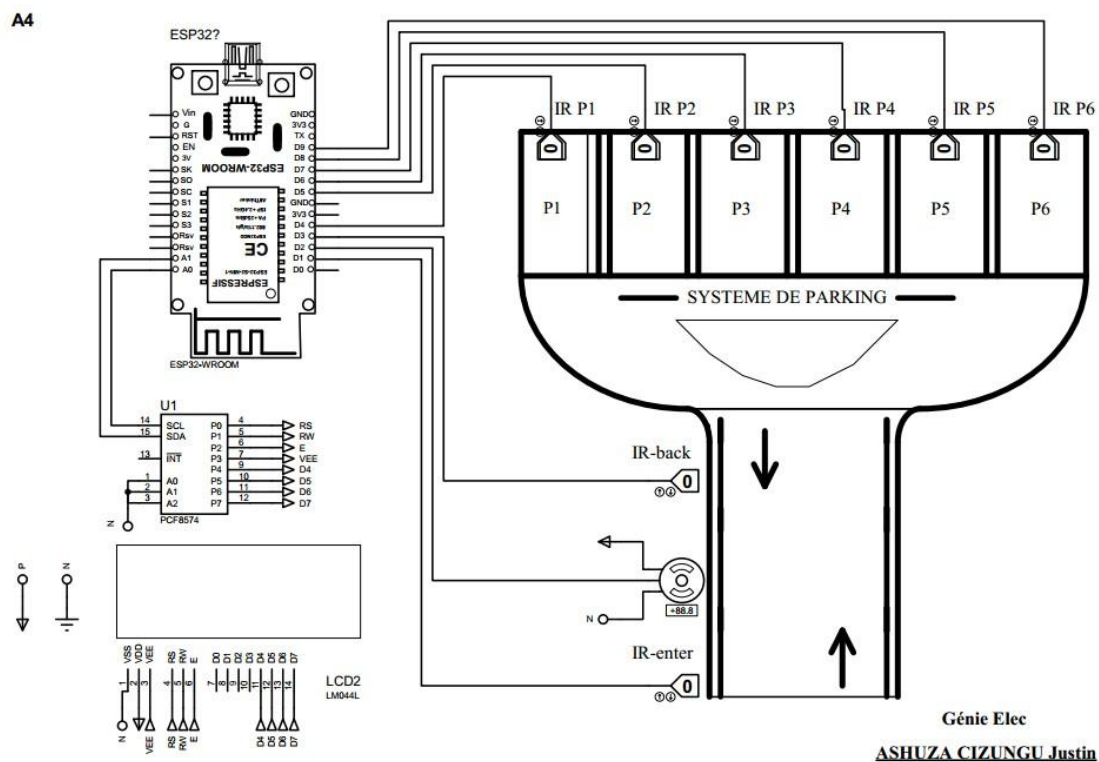


Figure 3-10 Schémas de principe sous Proteus.

3.4 Prototype proposé

Après la phase de simulation, nous avons procédé à la réalisation physique du système afin de concevoir un prototype répondant à la problématique de notre travail. Celui-ci a été réalisé à l'aide de différents composants électroniques. Le montage physique nous a permis d'effectuer plusieurs expérimentations ainsi que des tests sur le système. Les résultats obtenus ont permis d'évaluer l'efficacité du prototype et ont contribué à l'avancement du projet.



Figure 3-11 prototype physique

Dans cette section, nous présentons l'état actuel du parking. Ce dernier est encore vide, offrant six places de stationnement. Cette observation est clairement visible sur l'écran. Ainsi, nous pouvons constater l'absence de véhicules à cet instant.



Figure 3-12 entrer du premier véhicule

Dans cette section, nous constatons que le premier véhicule est déjà présent dans le parking et une place réservée. Cela est clairement visible sur l'écran P 1 Reserve et P5 Occupe. Cette indication signifie que le parking P1 est actuellement occupé. Ainsi, nous pouvons confirmer la présence d'un véhicule.



Figure 3-13 Parking plein

Dans cette section, nous observons que le parking est déjà plein. Par conséquent, aucun véhicule ne peut y entrer. Ce système a été conçu pour lutter contre l'encombrement. Ainsi, il garantit une gestion efficace de l'espace de stationnement. Il suffit la sortie d'un autre véhicule dans le parking pour qu'un autre véhicule puisse entrer.



Figure 3-14 Parking plein avec réservation

Dans cette section, le parking est entièrement occupé, y compris par des places déjà réservées à l'avance. Dans cette situation, aucun véhicule sans réservation ne peut entrer afin d'éviter tout encombrement. Les utilisateurs ayant réservé une place doivent saisir le code d'accès généré dans l'application pour confirmer leur accès. Ce code est ensuite vérifié par le système au niveau de la barrière. Si le code est valide, la barrière s'ouvre automatiquement malgré l'état complet du parking.

3.5 Application Web

i. Page d'inscription

La page d'inscription permet aux nouveaux utilisateurs de créer un compte afin d'accéder aux services du système de parking automatisé. L'utilisateur doit renseigner certaines informations personnelles telles que le nom, l'adresse e-mail et un mot de passe. Ces informations sont ensuite enregistrées dans la base de données du système. Cette fonctionnalité facilite la gestion

des usagers et permet d'identifier chaque utilisateur de manière unique, tout en assurant un accès sécurisé aux fonctionnalités telles que la réservation des places et le paiement.

AeroPark GOMA Accueil Réserver Connexion Inscription

Créer un compte

Rejoignez AeroPark GOMA pour réserver vos places

Nom complet

Adresse email

Mot de passe

Confirmer le mot de passe

S'inscrire

[Vous avez déjà un compte ? Se connecter](#)

Figure 3-15 Page d'inscription.

ii. Page d'authentification

Cette fonctionnalité a été mise en place pour des raisons de sécurité. L'utilisateur en charge de la gestion du parking aura besoin de s'authentifier pour pouvoir se connecter et utiliser les

fonctions de l'application. L'authentification se fait à l'aide d'un login et d'un mot de passe qui est inséré manuellement via l'interface.

Figure 3-16 Page d'authentification

iv. Page d'accueil

La page d'accueil de l'application constitue l'interface principale d'interaction avec l'utilisateur. Elle a été conçue pour offrir une lisibilité immédiate de l'état du parking grâce à un tableau de bord dynamique. L'utilisateur peut visualiser en temps réel le nombre de places disponibles, occupées ou réservées, facilitant ainsi sa prise de décision avant même son arrivée sur le site avec sa voiture.

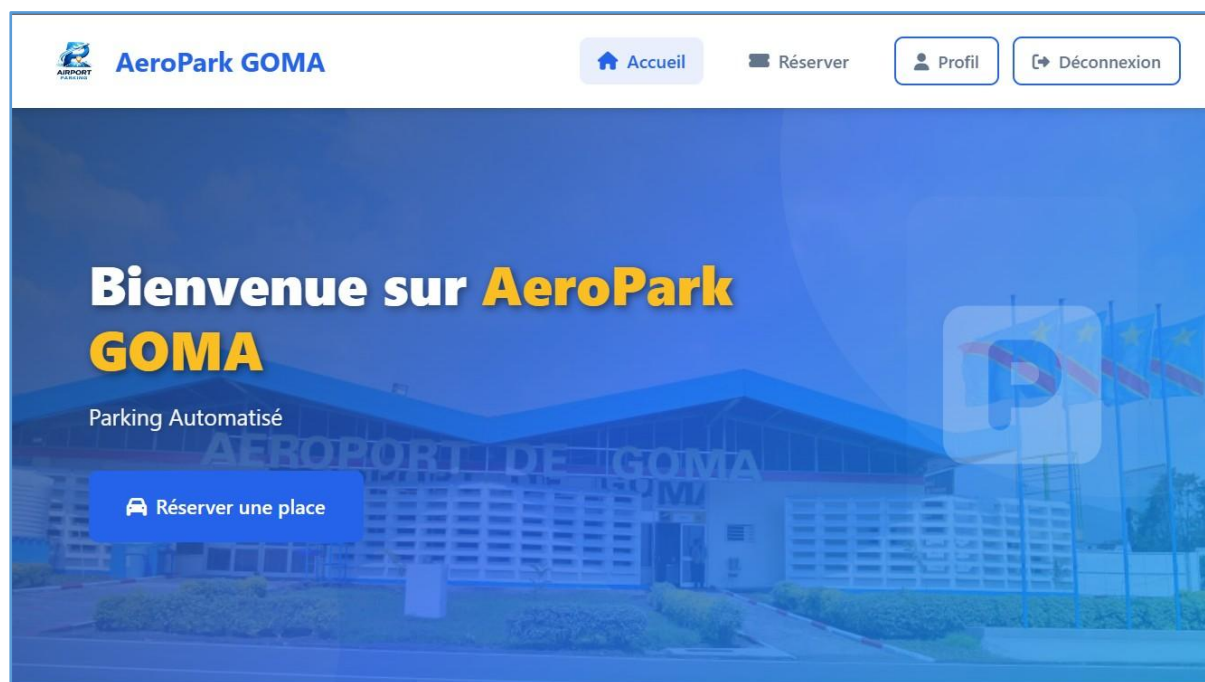


Figure 3-17 Page d'accueil



Figure 3-18 Aperçu du Parking

v. Page de réservation

La page de réservation constitue l'étape de transition entre l'espace virtuel et l'espace physique. Sur cette interface, l'utilisateur sélectionne la place de stationnement souhaitée ainsi que la durée de réservation. Le système calcule automatiquement le tarif applicable selon la devise définie par l'administrateur. Une fois la transaction validée, le serveur génère instantanément un code d'accès unique que le client devra utiliser pour ouvrir la barrière lorsque le parking est complet.

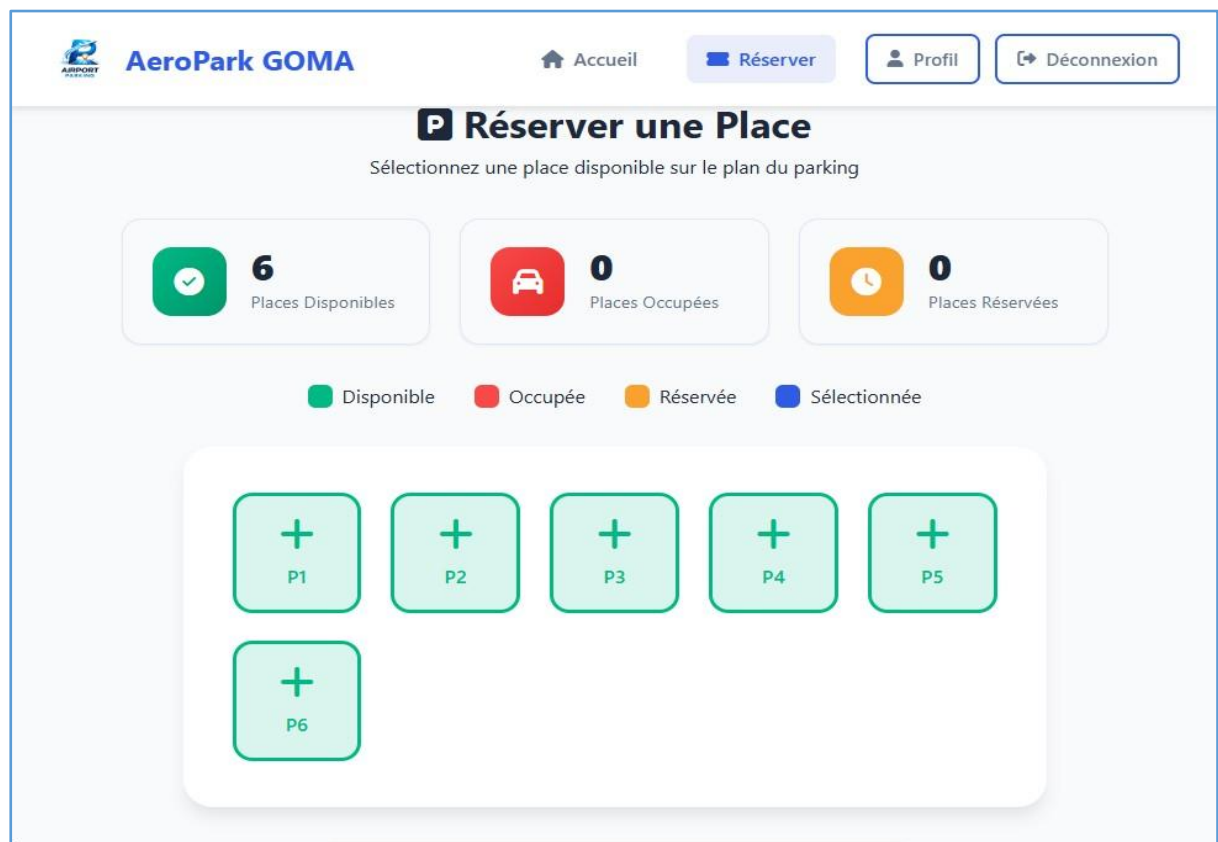


Figure 3-19 Page de réservation

vi. Page de profil

La page Profil permet à l'utilisateur de centraliser et de gérer l'ensemble de ses informations personnelles et de ses interactions avec le système. Comme on peut l'observer, cette interface affiche non seulement les données d'identification (nom, email, plaque d'immatriculation), mais fournit également Ces statistiques.

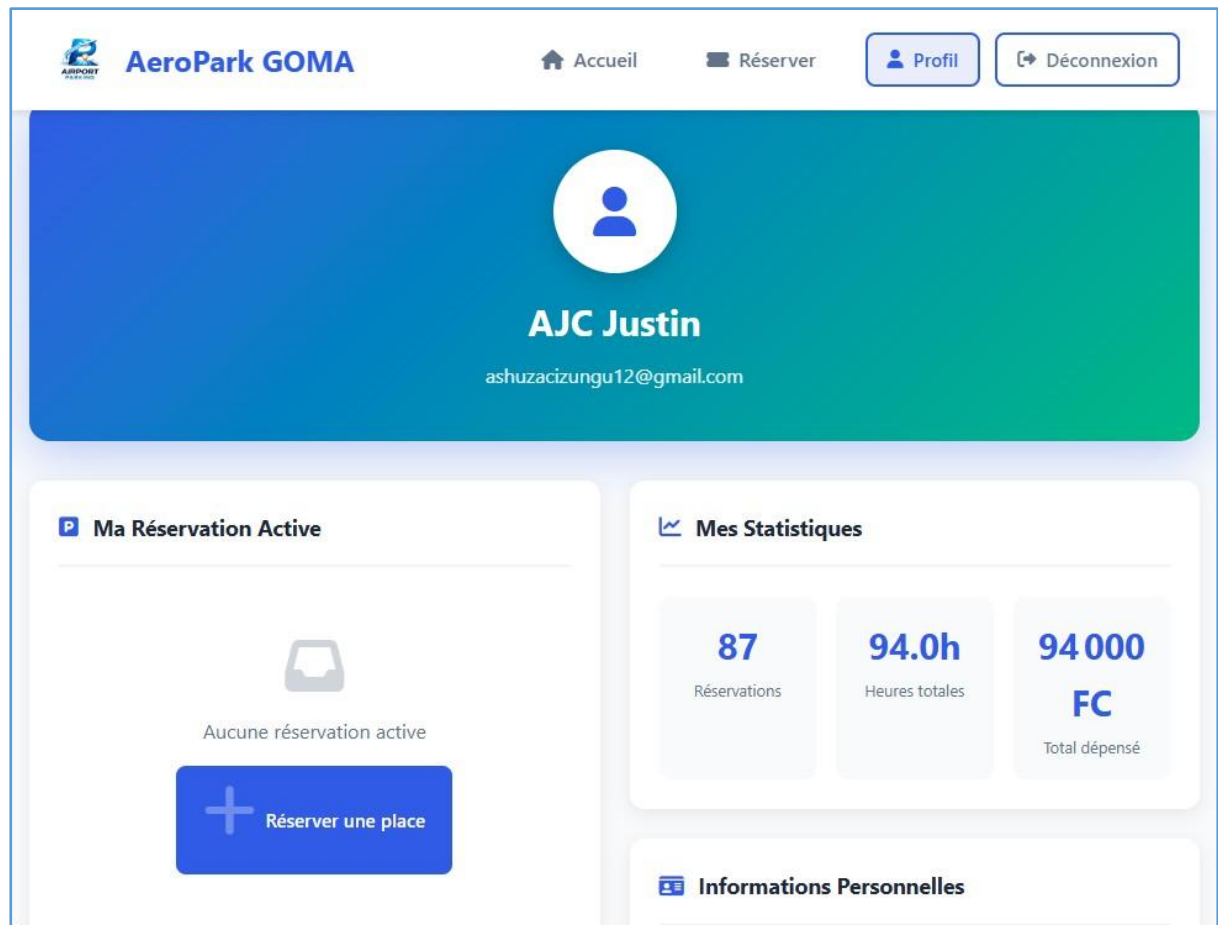


Figure 3-20 Page de profil

vii. Côté Administrateur

L'administration d'AeroPark constitue l'interface centrale de supervision et de contrôle du parking automatisé. Il offre une vue en temps réel des indicateurs clés tels que les places libres, réservées ou occupées, ainsi que les revenus générés. Grâce à une communication sécurisée avec la carte ESP32, il permet également un contrôle d'urgence des barrières. Le système intègre une gestion complète des réservations digitales et des clients physiques. Les statuts des places sont synchronisés dynamiquement avec la base de données Fire base. Un module dédié assure le suivi des utilisateurs et la traçabilité des paiements. Enfin, des paramètres avancés permettent la configuration sécurisée du système et la maintenance globale de l'infrastructure.

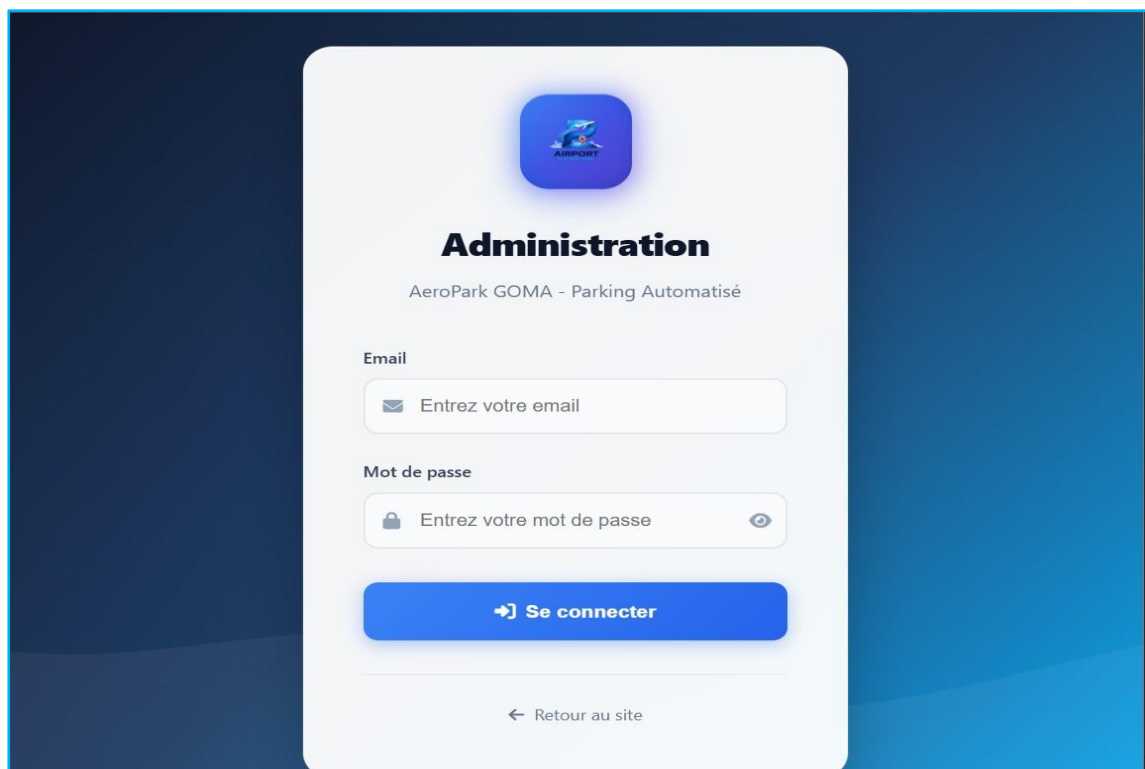


Figure 3-21 Page d'authentification

La figure 3-13 illustre l'interface de connexion, qui s'affiche dès l'ouverture de l'application. Elle offre à l'administrateur la possibilité de se connecter à son compte en utilisant son identifiant et son mot de passe.

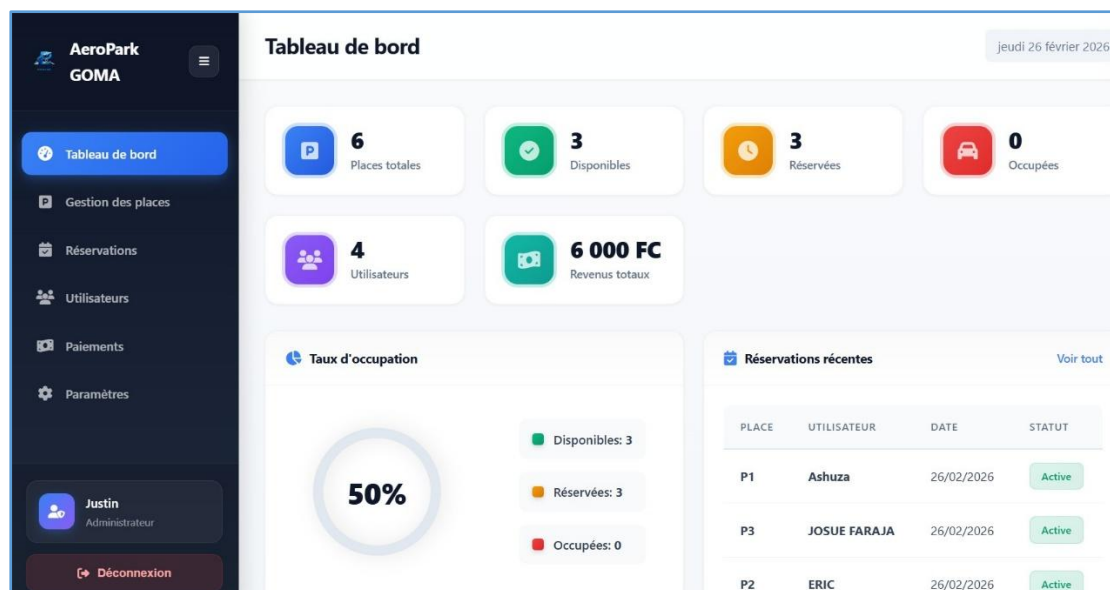


Figure 3-22 Tableau de bord

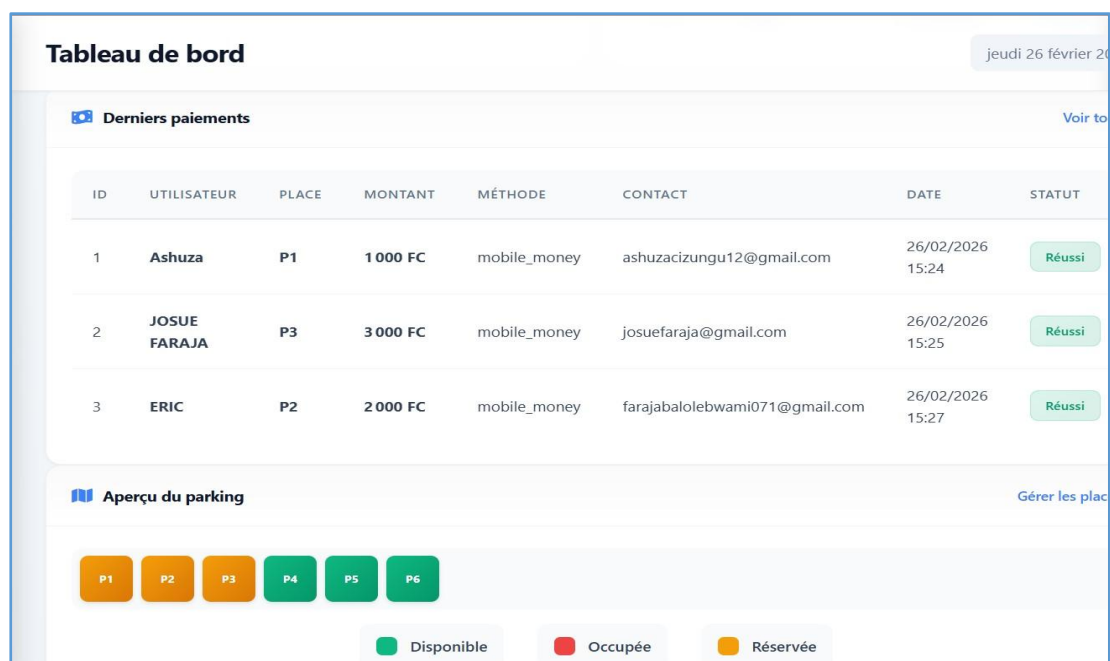


Figure 3-23 Tableau de bord

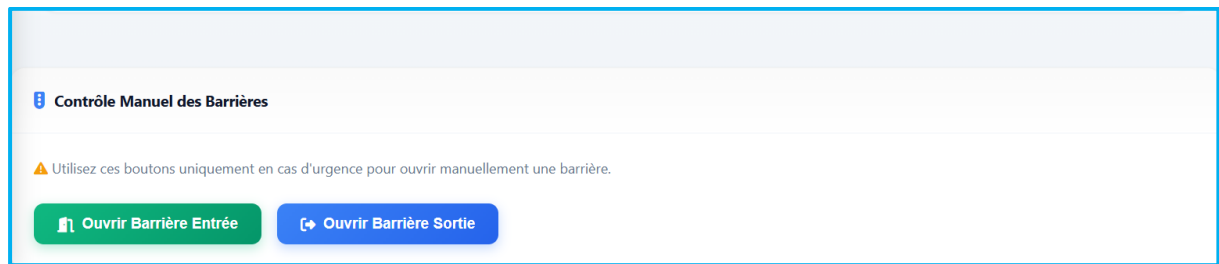


Figure 3-24 L'ouverture manuelle de la barrière

3.6 Estimation du coût des matériels

Tableau 3-1 Estimation du cout pour la mise en œuvre de ce système

Désignation	Quantité	Prix Unitaire	Prix Total
Module ESP32	1	\$16,00	\$16,00
Capteurs IR	8	\$5,00	\$40,00
Servomoteur SG90	1	\$5,00	\$5,00
Ecran LCD 20X4	1	\$10,00	\$10,00
Platine d'essai	1	\$3,00	\$3,00
Packet Fils Jump	1	\$4,00	\$4,00
Maquette	1	\$50,00	\$50,00
Module d'alimentation	1	\$2,50	\$2,50
Total			\$114,50

3.7 Conclusion partielle

Dans ce chapitre, nous avons présenté les résultats de la réalisation de notre système automatisé pour l'aéroport de Goma. Le premier défi a été de fiabiliser la détection physique des véhicules via l'ESP32 et de stabiliser le contrôle des barrières face aux interférences. Ensuite, l'intégration de ce matériel avec notre plateforme Cloud (Firebase et interfaces Web) a permis d'obtenir un prototype fonctionnel et synchronisé en temps réel. Ce prototype ayant été testé avec succès dans un environnement simulant les flux aéroportuaires, l'ensemble de ces résultats confirme que notre solution est viable et opérationnelle pour moderniser ce stationnement.



CONCLUSION GENERALE

Au terme de ce travail, nous avons développé un système automatisé de gestion de parking inspiré du contexte de l'aéroport de Goma, nous avons implémenté un prototype fonctionnel permettant d'automatiser le contrôle d'accès et la gestion des places de stationnement. Le système développé repose sur l'utilisation de capteurs infrarouges connectés à un microcontrôleur ESP32, assurant la détection des véhicules, le traitement logique des données et le contrôle des barrières d'entrée et de sortie par un servo moteur. Une application web a été mise en place comme interface de gestion, permettant la réservation des places et la supervision en temps réel des opérations.

Pour parvenir à ce résultat, nous sommes passés par une phase d'analyse au cours de laquelle les besoins fonctionnels et techniques du système ont été définis dans un cahier des charges. Nous avons ensuite procédé à la conception et à la modélisation de l'architecture du système, en mettant en œuvre une approche basée sur les principes des systèmes embarqués et de l'Internet des Objets (IoT). L'architecture retenue s'appuie sur une communication en temps réel via WebSockets et sur une API développée avec FastAPI, assurant la synchronisation entre le système physique et l'interface web. Après le choix des composants matériels et logiciels, nous avons réalisé l'implémentation du prototype, suivi de tests permettant de valider son bon fonctionnement et d'identifier certaines contraintes techniques, notamment liées aux interférences des capteurs et à la cohérence entre états physiques et logiques.

Le prototype proposé ne constitue pas un système complet, mais une base fonctionnelle démontrant la faisabilité d'une solution automatisée à coût maîtrisé pour la gestion d'un parking aéroportuaire. Conscients que ce travail ne saurait couvrir l'ensemble des exigences d'un système déployé en conditions réelles, nous restons ouverts à toute observation et amélioration susceptible d'enrichir cette approche.

De futures perspectives pourraient inclure l'intégration d'un système de reconnaissance automatique des plaques d'immatriculation, l'optimisation des mécanismes de sécurité, l'ajout de modules de paiement automatisé et ainsi que l'intégration de la vision par ordinateur pour la détection intelligente des véhicules, afin de rendre le système plus performant, autonome et conforme aux standards des infrastructures modernes.

BIBLIOGRAPHIE

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.
- [2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.
- [3] P. Bellavista, S. Member, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT Urban Scenarios," *IEEE Sens. J.*, vol. 13, no. 10, pp. 3558–3567, 2013, doi: 10.1109/JSEN.2013.2272099.
- [4] N. Sakib, A. S. M. Bakibillah, S. Susilawati, and A. S. Kamal, "Eco-Friendly Smart Car Parking Management System with Enhanced Sustainability," pp. 1–14, 2024.
- [5] P. Grelley, "Contrepoint — La méthode expérimentale," *Informations sociales*, vol. n° 174, no. 6, pp. 23–23, 2012, doi: 10.3917/inso.174.0023.
- [6] E. O. J. Christelle, "Conception et realisation d'un systeme de parking intelligent base sur la technologie rfid et les capteurs ir," *UNIVERSITE DE YAOUNDE I ECOLE NORMALE SUPERIEUR D'ENSEIGNEMENT TECHNIQUE D'EBOLOWA DEPARTEMENT DE DE GENIE INFORMATIQUE*, no. Dipet li, pp. 2019–2020, 2020.
- [7] Mr. G. E. Lounes, Mr. Alia, "Réalisation d'un parking intelligent," *Université A/Mira de Bejaia Faculté des Sciences Exactes Département d'Informatique*, 2017.
- [8] V. Bui, A. Alaei, and M. Bui, "On the Integration of AI and IoT Systems: A Case Study of Airport Smart Parking," *Internet of Things*, pp. 419–444, 2022, doi: 10.1007/978-3-030-87059-1_16.
- [9] H. Ben Rebah, "Internet of Things based Smart Parking System Monitoring using Mobile and Web Applications," *1st International Conference on Innovative Engineering*



- Sciences and Technological Research, ICIESTR 2024 - Proceedings*, no. May 2024, 2024, doi: 10.1109/ICIESTR60916.2024.10798322.
- [10] V. Kiruthika, M. Jagadeeswari, C. Snehaprabha, and H. Sreejaa, "IoT Based Smart Parking System," *AIP Conf. Proc.*, vol. 2914, no. 1, 2023, doi: 10.1063/5.0175916.
- [11] S. P. Collins *et al.*, *No Title 済無No Title No Title No Title*. 2021.
- [12] "shutterstock."
- [13] U. Manual, "Target Areas," *The Lancet*, vol. 300, no. 7770, p. 222, 1972, doi: 10.1016/S0140-6736(72)91649-2.
- [14] "Benelec."
- [15] P. Ajmera, "A Review Paper on Infrared sensor," *International Journal of Engineering Research & Technology*, vol. 5, no. 23, pp. 1–3, 2018.
- [16] "PCbasic."
- [17] "Omnifab."
- [18] "Wix."
- [19] D. S. Poltak Sihombing, "The LCD Interfacing and Programming."
- [20] H. Technology, "I2C Serial Interface 20x4 LCD Module," *Datasheet LCD 20X4 I2C Serial Interface*, pp. 1–26, 2019.
- [21] J. Crouzet, "De de en d e," vol. XXVII, pp. 1–2, 1983.
- [22] "esp32-wroom-32d-wifi-bluetooth." [Online]. Available: <https://www.direnc.net/esp32-wroom-32d-wifi-bluetooth-gelistirme-modulu-en>
- [23] "Otronic."
- [24] Jumper, "HTB1BPK_coCF3KVjSZJnq6znHFXaX." [Online]. Available: <https://aytoo.ma/produit/kit-fils-cables-jumper-40-x-200-mm/>
- [25] Erik Bartman, "Le grand livre d'arduino," 2015.
- [26] Plaque d'essai, "7." [Online]. Available: <https://boutique.3sigma.fr/prototypage/32-plaque-d-essai.html>
- [27] M. A. A. Mareai, "Basic Electronic Circuits by Proteus Simulating Software," 2024.

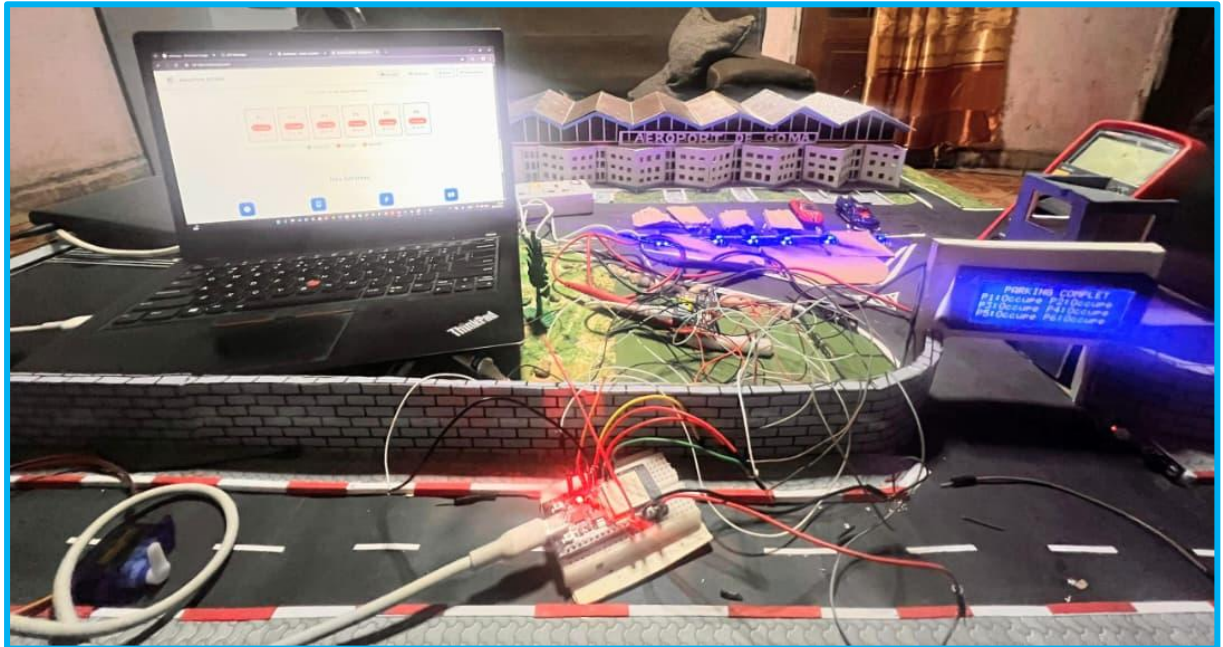
- [28] M. Fezari and A. A. D. Al Zaytoona, "Integrated Development Environment 'IDE' For Arduino," *ResearchGate*, no. 10, pp. 1–11, 2018.
- [29] J. Tan, Y. Chen, and S. Jiao, *Visual Studio Code in Introductory Computer Science Course: An Experience Report*, vol. 1, no. 1. Association for Computing Machinery, 2024. doi: 10.18260/1-2--48259.
- [30] Collegegujan, "servomoteur_."
- [31] Fabacademy, "Fonctionnement du servomoteur et interface avec Arduino", [Online]. Available: <https://fabacademy.org/2021/labs/talents/students/munzir-alradi/assignments/output-devices/>
- [32] Simon.L et Hippolyte.W, "Arduino : Premiers pas en informatique embarquée," *Le blog d'Eskimon*, pp. 1–639, 2020.
- [33] D. S. Poltak Sihombing, "The LCD Interfacing and Programming." [Online]. Available: <https://www.intechopen.com/chapters/80636>
- [34] Dfrobot, "[Online]. Available: <https://community.dfrobot.com/makelog-311989.html>

ANNEXES

I. Parking de l'aéroport de Goma



II. Illustrations des tests



II. Extrait du code source Arduino

```
// =====
// SECTION 1 : INCLUSION DES BIBLIOTHÈQUES
// =====

#include <WiFi.h>           // Gestion WiFi sur ESP32
#include <HTTPClient.h>      // Requêtes HTTP pour communiquer avec le backend
#include <ArduinoJson.h>     // Manipulation JSON pour envoyer/recevoir données
#include <WebSocketsClient.h> // WebSocket pour mise à jour temps réel
#include <ESP32Servo.h>      // Contrôle du servomoteur pour la barrière
#include <Wire.h>            // Communication I2C (LCD)
#include <LiquidCrystal_I2C.h> // Pilotage écran LCD I2C 20x4

// =====
// SECTION 2 : CONFIGURATION WIFI & BACKEND
// =====

const char* ssid    = "globaltech";           // Nom du réseau WiFi
const char* password = "Akim1234@@@";         // Mot de passe WiFi
const char* serverHost = "192.168.1.141";      // Adresse IP du serveur backend
const int  serverPort = 8000;                  // Port du serveur backend
const char* apiKey    = "aeropark-sensor-key-2024"; // Clé API pour authentification
```

```

// Construction dynamique des URLs d'API à partir du host et du port
String sensorUpdateUrl = "http://" + String(serverHost) + ":" + String(serverPort) + "/api/v1/sensor/update";
String parkingStatusUrl = "http://" + String(serverHost) + ":" + String(serverPort) + "/parking/status";

// =====
// SECTION 3 : CONFIGURATION MATÉRIELLE (GPIO + LCD I2C)
// =====

#define NB_PLACES 6

const int IR_PINS[NB_PLACES] = {13, 14, 25, 26, 27, 19}; // Capteurs IR des places
const int IR_ENTREE = 32; // Capteur barrière entrée
const int IR_SORTIE = 33; // Capteur barrière sortie
const int SERVO_PIN = 4; // Servo barrière

// Anti-rebond asymétrique
#define DEBOUNCE_LIBRE_VERS_OCCUPE_MS 500 // Détection rapide
#define DEBOUNCE_OCCUPE_VERS_LIBRE_MS 10000 // Résistance aux faux positifs

// Vote majoritaire pour confirmer qu'une place est libre
#define VOTES_CONFIRMATION_LIBRE 10
#define VOTE_CONFIRMATION_OK 9
#define VOTE_CONFIRMATION_INTERVALLE_MS 100

// LCD I2C 20x4
LiquidCrystal I2C lcd(0x27, 20, 4); // Adresse I2C 0x27

// =====
// SECTION 4 : LOGIQUE DE CONTRÔLE DE LA BARRIÈRE D'ENTRÉE
// =====

// Détecte une voiture à l'entrée, applique les vérifications et ouvre la barrière
// si l'accès est autorisé (places libres ou code valide).
// Vérifie verrou anti-rebond post-fermeture (3s)
bool verrouillee = (maintenant - derniereFermeture < 3000);
// Détection voiture par vote majoritaire 2/3 (FIXE B)
bool voitureEntree = false;

if (systemReady && !verrouillee && digitalRead(IR_ENTREE) == LOW)
{
    int voteEntree = 0;
    for (int v = 0; v < 3; v++)
    {
        if (digitalRead(IR_ENTREE) == LOW)
            voteEntree++;
    }
}

```

```

    delay(30);
}
voitureEntree = (voteEntree >= 2); // true si 2/3 lectures LOW
}
// Si une voiture est détectée
if (voitureEntree)
{
    // Compte les places libres selon état debounced
    int libres = 0;

    for (int i = 0; i < NB_PLACES; i++)
        if (etatPlaces[i] == LIBRE)
            libres++;

    bool autorise = false;

    if (libres > 0)
    {
        autorise = true; // Places disponibles → accès autorisé
        Serial.print("AUTORISÉ — Places libres : ");
        Serial.println(libres);
    }
    else if (codeValidatedViaWeb && (maintenant - codeValidationTime < 60000))
    {
        autorise = true; // Parking plein, mais code réservation valide
        Serial.println("AUTORISÉ via code application mobile.");
    } else
    {
        Serial.println("REFUSÉ — Parking complet."); // Refus si parking plein et pas de code
    }
    // Action si accès autorisé
    if (autorise)
    {
        ouvrirBarriere(); // Lever la barrière
        delay(5000);      // Maintenir ouverte 5s
        fermerBarriere(); // Refermer automatiquement
        codeValidatedViaWeb = false; // Invalide le code après usage
        derniereFermeture = millis(); // Mettre à jour verrou anti-rebond
        afficherLCD(); // Mise à jour affichage LCD
    }
}
}

```