

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJIMOKHTAR – ANNABA
BADJIMOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de l'Ingénierie

Département : Electronique

Domaine : Sciences et technologies

Filière : Electronique

Spécialité : Electronique des systèmes

Embarqués

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

Etat de l'art de « Smart Parking », Systèmes de gestion de parking intelligent

Présenté par : BAH ASTAN

Encadrant : U.B.M. Annaba

Jury de Soutenance :

Président	Grade	U.B.M. Annaba	rimaS assuomneB
Encadrant	Grade	U.B.M. Annaba	habaR lekaL
Examineur	Grade	U.B.M. Annaba	meziH tiA

Année Universitaire : 2020/2021

DEDICACE

Je dédie ce travail aux personnes qui sont chères à mon Cœur :

*A commencer par ma mère **KAMISSA KONE** et à mon père **GAYE BAH**
pour leur amour inconditionnel et leur soutien sans faille.*

***A tous mes parents** sans qui, je ne serais pas là où j'en suis maintenant.*

Pour tous leurs sacrifices, leurs prières, leurs amours et leurs soutiens.

***Mes Frères** pour leurs appuis et leurs encouragements permanents.*

*A toute **ma Famille** et surtout à **ma FAMILLE D'ALGERIE** pour leur
soutien tout au long de mon parcours et tous les bons moments passé
ensemble.*

*A Mes **Ami(e)s** ainsi que toutes **les personnes** qui m'ont soutenu et aidé
tout au long de mes études.*

Remerciements

*Après avoir rendu grâce à Allah le Tout Puissant et Le Tout-Miséricordieux de m'avoir donné la force et le courage pour accomplir ce stage de fin d'étude ainsi que tout au long de mes années études, je tiens à remercier l'ancien Monsieur le chef du département Electronique **Pr. Abdelghani REDJATI** de m'avoir accueillis au département et la responsable de Filière Electronique **Dr. Hafis TOUFIK** de m'avoir permis de suivre ce cursus de « Master Electronique des Systèmes Embarqués ». Je remercie aussi les personnes qui m'ont aidé pour mener à bien ce projet de stage et pour l'élaboration de ce mémoire, à commencer par mon encadreur **Pr. Lakel RABAH** pour sa disponibilité, son aide, ses conseils et encouragements. Je remercie également les membres de jury de m'avoir fait l'honneur en acceptant d'examiner et de juger mon travail Messieurs : **Pr. rimaS assuomneB** et **Dr. AmeziH ti** . Je remercie très vivement tous mes professeurs, mes promotionnaires, ami(e)s et tous le cadre scientifique de l'université Badji Mokhtar-Annaba qui mont aider à mener à bien ce projet de stage de fin d'étude, en dépit de la crise sanitaire mondiale liée au COVID-19.*

Résumé

Aujourd'hui, de nombreuses régions métropolitaines ont connu une croissance explosive du nombre de visiteurs et de clients en raison de la revitalisation urbaine, de l'extension des services de transport en commun dans les zones suburbaines et de la tendance générale à une mobilité accrue de notre société. En conséquence, il y a trop de véhicules sur la route et des places de stationnement insuffisantes. Cela a conduit à la nécessité d'un système de gestion de stationnement efficace appelé smart parking ou parking intelligent.

Le parking intelligent regroupe les technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications permettant d'améliorer le trafic dans la ville c'est-à-dire à circuler plus librement, guide les résidents et les visiteurs au stationnement disponible, prévoir avec précision l'occupation des points/véhicules en temps réel, aussi Permet des décisions intelligentes à l'aide de données, notamment des applications qui informe en temps réel le client et des rapports d'analyse historiques .Le parking intelligent joue un rôle majeur dans la création d'un meilleur environnement urbain en réduisant considérablement le temps de recherche pour stationner ce qui par conséquence réduit la consommation de carburant nous permettant de faire des économies créant moins d'embouteillage dans les routes et par la même occasion une réduction des émissions de CO2 et autres polluants.

Le but de ce travail est de réaliser un prototype d'un parking intelligent, en se basant sur (consultation, confort, gestion du temps et énergie), ces scénarios seront automatisés via un microcontrôleur (Arduino UNO) des différents capteurs ainsi qu'une application mobile qui nous permettra de mener des actions à distance.

Mots clés :

Parking intelligent, ville intelligente, Internet des objets, Arduino UNO, Application mobile.

Abstract

Today, many metropolitan areas have experienced explosive growth in the number of visitors and customers due to urban revitalization, the expansion of transit services to suburban areas, and the general trend towards increased mobility. Of our society. As a result, there are too many vehicles on the road and insufficient parking spaces. This has led to the need for an efficient parking management system called smart parking or intelligent parking.

The smart car park brings together electronic, automatic, computer and telecommunications technologies to improve traffic in the city, that is to say to move more freely, guides residents and visitors. parking available, accurately predict point / vehicle occupancy in real time, also Enables intelligent decisions using data, including applications that notify customer in real time and historical analysis reports, Parking intelligent plays a major role in creating a better urban environment by drastically reducing the search time to park which consequently reduces fuel consumption allowing us to save money creating less congestion on the roads and at the same time occasion a reduction in CO2 emissions and other pollutants.

The goal of this work is to make a prototype of an intelligent parking lot, based on (consultation, comfort, time and energy management), these scenarios will be automated via a microcontroller (Arduino UNO) of the various sensors as well as a mobile application that will allow us to carry out actions remotely.

Key words:

Smart Parking, Smart City, Internet of Things, Arduino UNO, Mobile App.

الملخص

اليوم، شهدت العديد من المناطق الحضرية نموًا هائلًا في عدد الزوار والعملاء بسبب التنشيط الحضري، وتوسيع خدمات النقل إلى مناطق الضواحي، والاتجاه العام نحو زيادة التنقل. نتيجة لذلك، يوجد عدد كبير جدًا من المركبات على الطريق وأماكن وقوف السيارات غير كافية. وقد أدى ذلك إلى الحاجة إلى نظام فعال لإدارة مواقف السيارات يسمى وقوف السيارات الذكي أو وقوف السيارات الذكي.

يجمع موقف السيارات الذكي بين التقنيات الإلكترونية والأوتوماتيكية والكمبيوتر والاتصالات السلوكية واللاسلكية لتحسين حركة المرور في المدينة، أي التحرك بحرية أكبر، وإرشاد السكان والزوار. تتوفر مواقف للسيارات، والتنبؤ بدقة بنقطة / إشغال السيارة في الوقت الفعلي، وتمكين أيضًا قرارات ذكية باستخدام البيانات، بما في ذلك التطبيقات التي تخطر العميل في الوقت الفعلي وتقارير التحليل التاريخي، يلعب نظام Parking Intelligent دورًا رئيسيًا في خلق بيئة حضرية أفضل من خلال تقليل وقت البحث بشكل كبير عن ركن السيارة مما يقلل بالتالي من استهلاك الوقود مما يسمح لنا بتوفير المال مما يؤدي إلى تقليل الازدحام على الطرق وفي نفس الوقت يؤدي إلى انخفاض انبعاثات ثاني أكسيد الكربون والملوثات الأخرى. الهدف من هذا العمل هو إنشاء نموذج أولي لموقف سيارات ذكي، استنادًا إلى (الاستشارة والراحة وإدارة الوقت والطاقة)، سيتم أتمتة هذه السيناريوهات عبر متحكم دقيق (Arduino UNO) لمختلف أجهزة الاستشعار بالإضافة إلى جهاز محمول التطبيق الذي سيتيح لنا تنفيذ الإجراءات عن بُعد.

الكلمات الدالة:

مواقف السيارات الذكية، المدينة الذكية، إنترنت الأشياء، Arduino UNO، تطبيق الهاتف المحمول.

Liste des Tableaux

Tableau 1: Caractéristique techniques du modem GSM/GPRS type M10..... 13

Liste des Figures

Figure 1: Stationnement intelligent	1
Figure 2: Exemples sur l'internet des objets	2
Figure 3: La mobilité intelligente.....	3
Figure 4: Stationnement au bord de la route	3
Figure 5: Stationnement intelligent des véhicules	6
Figure 6: carte microcontrôleur Arduino UNO	10
Figure 7: module GSM SIM900	11
Figure 8: Description module GSM	12
Figure 9: les broches du module GSM.....	14
Figure 10: Servomoteur	15
Figure 11: Signal de commande variant de 0 à 180°	15
Figure 12: Afficheur LCD 2 x 16.....	16
Figure 13: Configuration des pines.....	16
Figure 14: Capteur de détection IR (infrarouge).....	17
Figure 15: Capteur de mouvement	18
Figure 16: circuit du DS1307	19
Figure 17: Module RTC DS1307	19
Figure 18: Boitier de 8 broches du DS1307	20
Figure 19: L'interface de logiciel Arduino IDE	21
Figure 20: Les composants d'une barre d'action	21
Figure 21: Schéma fonctionnel des commandes AT	21
Figure 22: Interface du logiciel PROTEUS.....	23
Figure 23: Interface ISIS	24
Figure 24: Interface du logiciel PROTEUS ARES	24
Figure 25: L'interface Virtual Serial Ports	26
Figure 26: Interface de Blynk.....	27
Figure 27: Fonctionnement de Blynk	28
Figure 28: Insertion de l'équipement Arduino dans l'interface ISIS	31
Figure 29: Schéma électronique complet entier de notre prototype smart Parking	32
Figure 30: SMART PARKING (insertions des moniteurs séries).....	33
Figure 31: Prototype : configuration et insertion de la bibliothèque dans le module GSM	34
Figure 32: Organigramme de fonctionnement	39
Figure 33: Résultat de la simulation sous PROTEUS.....	39
Figure 34: résultat du premier scenario deux véhicules entrés	40
Figure 35: les rotations du servomoteur quand une voiture est détecté.....	41
Figure 36: Résultat de l'envoi du GSM au début de la simulation.....	41
Figure 37: Résultat de l'envoi du GSM en cas d'intrusion après l'heure fixé	42
Figure 38: Résultat de l'envoi du GSM des coordonnées GSP du parking	42
Figure 39: réservation à partir de l'application	43
Figure 40: Résultat après la réservation sur l'application	43
Figure 41: Annulation de la réservation.....	44

Table des matières

Introduction générale.....	9
Chapitre I : Etat de l'art.....	0
1. Introduction.....	1
1.1. Problématique de stationnement.....	1
1.2. Quelques définitions.....	1
1.2.1. Internet des objets.....	1
1.2.2. La mobilité intelligente.....	2
1.2.3. Stationnement.....	3
1.2.4. Politique de stationnement [3].....	3
1.2.5. Les différents types de stationnement [4].....	4
1.3. Smart parking.....	5
1.4. Les défis et avantages de la technologie parking intelligent.....	6
1.4.1. Les besoins du stationnement.....	6
1.4.2. Avantages de la technologie smart parking.....	6
1.5. Conclusion.....	7
Chapitre II : Description du fonctionnement « smart parking ».....	8
1. Introduction :.....	9
2. Partie matérielle.....	9
2.1. Le microcontrôleur Arduino UNO.....	9
2.2. Le module de communication GSM.....	11
2.2.1. Description du matérielle.....	11
2.2.2. Mise sous tension.....	13
2.2.3. LEDs.....	13
2.2.4. Caractéristiques techniques.....	13
2.3. Le système GPS :.....	14
2.4. Servomoteur :.....	14
2.5. L'afficheur LCD (Liquid Crystal Display).....	15
2.6. Capteur de détection d'obstacle infrarouge IR :.....	17
2.7. Le capteur de mouvement PIR SENSOR3.....	17
2.8. Horloge à temps réel RTC DS1307 :.....	18
3. Partie logicielle.....	20
3.1. Arduino IDE :.....	20
3.2. Les commandes AT.....	21
3.3. LES LOGICIELS DE SIMULATION ELECTRONIQUE.....	22

3.3.1.	Le logiciel PROTEUS	22
3.3.2.	L'interface ISIS	23
3.3.3.	L'interface ARES.....	24
3.3.4.	Virtual Serial Ports.....	25
3.4.	L'application mobile Blynk :.....	26
4.	Conclusion	28
Chapitre III : Simulation du fonctionnement du système		30
1.	Introduction.....	31
2.	Connexion des différents équipements sous PROTEUS.....	31
3.	Le principe de fonctionnement du prototype.....	34
4.	Organigrammes de fonctionnement du prototype.....	35
5.	Résultats	39
6.	Conclusion	44
Références bibliographiques.....		46

Introduction générale

Depuis des années, on a constaté dans plusieurs villes, notamment les grandes cités appelées mégapoles, que les conducteurs avaient beaucoup de problèmes pour trouver une place de parking facilement. Cela entraîne un embouteillage dans les villes, et d'après une étude, Un problème de pollution se pose également, les automobilistes qui tournent dans la ville pour chercher une place de parking polluent la ville sans se rendre compte. Le temps de recherche entraîne aussi l'énerverment des usagers ce qui n'aide pas à trouver une place, En effet un nombre d'heures sont perdues tous les jours pour chercher une place de parking dans le monde. Les Parking devraient offrir aux clients suffisamment d'espace pour garer leur voiture, car la voiture joue un rôle énorme dans le transport. Et pour cela, nous avons besoin d'un nouveau système, un système qui peut aider à gérer et réduire la circulation routière. Un système qui aide les clients à gagner du temps à la recherche d'une place de stationnement. La nécessité de ces options conduit à l'émergence d'un nouveau terme appelé « Stationnement intelligent »

En raison de l'importance des appareils mobiles intelligents dans la société d'aujourd'hui, de plus en plus de personnes utilisent le site web pour résoudre les problèmes quotidiens.

Cette technologie offre de nouveaux moyens pour optimiser l'utilisation des espaces de stationnement.

Notre projet consiste à réaliser un parking intelligent dans le but d'informer l'utilisateur sur la disponibilité d'une place libre de stationnement et de faire la réservation via une application mobile, auquel nous pouvons y accéder sur internet en temps réel. Amener l'utilisateur directement sur une place libre offre plusieurs avantages tel que : réduire le temps, économiser le carburant et diminuer les émissions de la pollution

Le stationnement intelligent (smart parking) a été conçu pour diverses raisons, pour réserver leur stationnement à l'avance, d'identifier les entrées/sorties des voitures, ou prédire très précisément où ils vont probablement trouver une place, et pour collecter des données à l'aide de capteurs, actionneurs et autres objets physiques, connectés à Internet via différents protocoles d'échange d'informations et de communication, pour assurer la surveillance, et la gestion.

Pour mener à bien notre travail, nous l'avons organisé en trois chapitres comme suite : dans le premier chapitre, qui fait l'objet d'une généralité sur le parking intelligent, la définition, les besoins et avantages.

Dans le deuxième chapitre, nous allons présenter l'aspect matériel et logiciel des différents composants électroniques utilisés pour réaliser un parking intelligent et détailler leurs fonctionnements.

Le troisième chapitre concerne la programmation et simulation de prototype réalisé, la présentation des essais et les commentaires des résultats obtenus.

Une conclusion, dans laquelle on met en perspective les développements futurs dans le domaine des transports à l'intérieur des grandes villes, vient clôturer ce travail de fin des études en Master.

Chapitre I :

Etat de l'art

1. Introduction

Avec les nouvelles technologies les chercheurs veillent à améliorer les conditions de vie des gens, La recherche concerne d'abord le confort et la facilité de mouvement et de vitesse dans la vie quotidienne comme aller au travail, faire du shopping, etc. Le Smart parking (stationnement intelligent) est un résultat d'une ville intelligente, pour que cela aide à résoudre les besoins et problèmes tels que la facilité de mouvement et d'accès au travail et réduire la pollution et aide le conducteur à trouver des espaces libres avec la possibilité de réservation par téléphone mobile.

Dans notre premier chapitre, nous offrons la solution de réduire le temps de recherche d'espace libre et les problèmes de stationnement, et nous allons conclure ce chapitre par une conclusion.

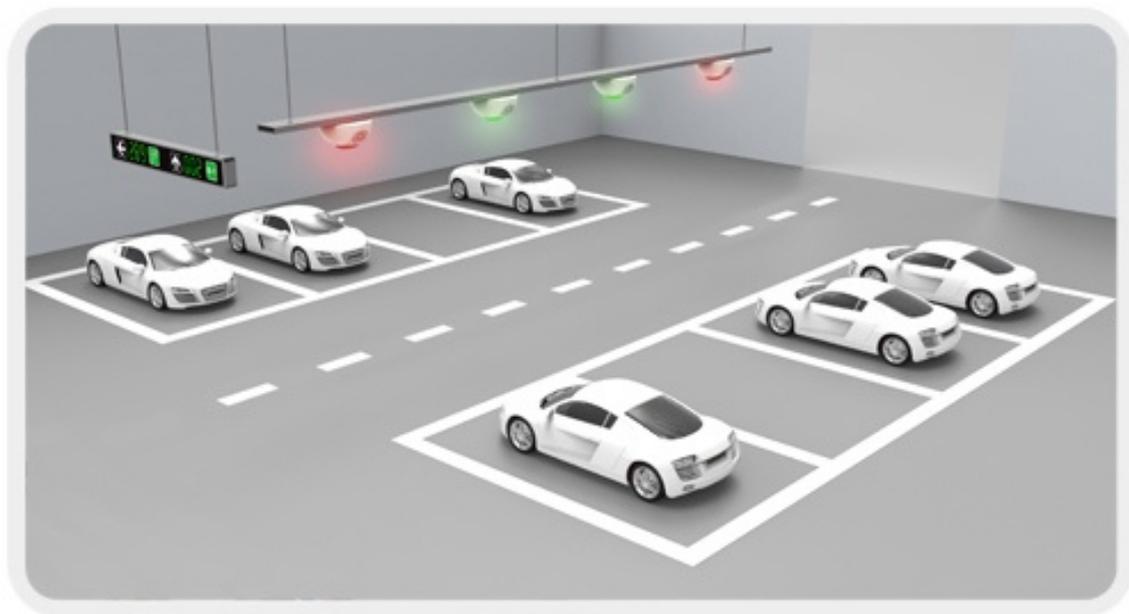


Figure 1: Stationnement intelligent

1.1. Problématique de stationnement

Les différents moyens de transport sont essentiels actuellement, mais avec le nombre de ces moyens, les embouteillages et les problèmes de circulation augmentent, beaucoup de voiture sont mal garés ce qui entraîne les accidents, les parkings sont mal gérés en plus de manque d'espaces de stationnement, sans prendre en compte la perte de temps. Pour réduire ces problèmes on a pensé au stationnement intelligent, qui va gérer la facilité et la fiabilité de mouvement dans les villes.

1.2. Quelques définitions

1.2.1. Internet des objets

L'Internet des objets, parfois écrit IdO ou IOT (Internet of things), désigne l'ensemble des infrastructures et technologies mises en place pour faire fonctionner des objets divers par le

eux. Tous ces termes techniques peuvent porter à confusion; résumons donc en disant que la mobilité intelligente donne à chaque usager l'opportunité de devenir acteur de sa mobilité, il n'a plus à la subir comme cela a été le cas pendant bien trop longtemps. [2]

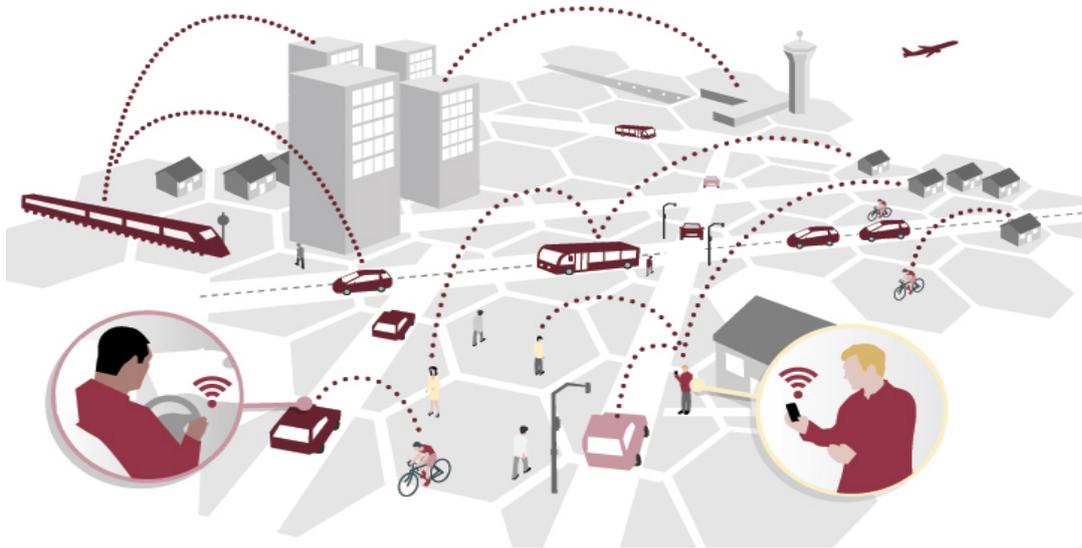


Figure 3: La mobilité intelligente

1.2.3. Stationnement

L'immobilisation momentanée d'un véhicule est ce qui définit un stationnement, c'est soit avec paiement ou gratuit. Le véhicule doit être garé au bord de la route pour qu'il ne perturbe pas les usagers et ne cause pas de problèmes et que le conducteur soit à l'aise avec sa voiture.



Figure 4: Stationnement au bord de la route

1.2.4. Politique de stationnement [3]

La politique de stationnement est une réglementation mise en place à l'intérieur des parkings d'entreprise afin de réguler le stationnement des employés.

Cette réglementation a pour but d'optimiser la gestion du parking en mettant en place plusieurs solutions et fonctionnalités pour répondre aux problématiques de parking rencontrées par l'entreprise.

La politique de stationnement est un outil qui vient s'inscrire dans le plan de mobilité des entreprises (anciennement plan de déplacement des entreprises).

Elle va permettre de réguler le trafic à l'intérieur mais également à l'extérieur du parking, on vise donc avec cet outil une gestion globale de la mobilité.

1.2.5. Les différents types de stationnement [4]

- **Stationnement public**
Il est convenu d'appeler stationnement public, tout stationnement, souvent hors-rue, accessible à tous sans réserve. Les stationnements loués à des individus sont considérés publics lorsqu'ils peuvent être loués à quiconque au moment de refaire le contrat de location.
- **Stationnement privé**
Il est convenu de catégoriser comme stationnement privé, les espaces de stationnement qui sont réservés (ou interdits) à une catégorie spécifique d'utilisateurs (employés, résidents, propriétaires, etc.).
- **Stationnement payant**
Un stationnement est dit payant, lorsque l'acte de stationnement est autorisé moyennant une charge monétaire. Le montant exigé peut être fixe ou peut varier selon : le temps de stationnement, le moment de la journée ou la saison.
- **Stationnement gratuit**
Un stationnement est dit gratuit, s'il n'exige pas un paiement à la suite d'un acte de stationnement.
- **stationnement pour personnes handicapées**
Un espace de stationnement est dit pour personnes handicapées, s'il est exclusivement réservé pour des personnes à mobilité réduite. Généralement, il respecte les géométries exigées par les normes en vigueur.
- **Stationnement incitatif**
Un stationnement incitatif est un espace de stationnement pour automobiles, généralement située périphérie d'une agglomération et qui a pour but d'inciter les automobilistes à accéder à leurs centres d'intérêt en transport en commun. Il peut être intérieur ou extérieur, payant ou gratuit.
- **Stationnement de longue durée**
Un stationnement est dit de longue durée, lorsqu'il est conçu pour un roulement plus lent (périodes de stationnement plus longues). Il est généralement présent dans des endroits où l'on trouve des voyageurs (aéroports, gares, ports...).
- **Stationnement de courte durée**

Un stationnement de courte durée est prévu pour un roulement rapide (période de stationnement plus court). Il est généralement présent sur rue dans les centres d'affaires.

➤ **Stationnement temporaire**

Un espace de stationnement est dit temporaire, lorsqu'il est créé afin de corriger ou d'accommoder, pour une durée déterminée, une contrainte temporaire au stationnement (chantier, inondation...).

➤ **Stationnement sur rue**

Un espace de stationnement hors rue est un espace de stationnement qui n'est pas situé le long d'une voie de circulation automobile.

➤ **Stationnement en parallèle (ou longitudinal)**

Le stationnement en parallèle est un mode de stationnement qui consiste à garer une automobile Parallèlement à un trottoir. Ce type de stationnement est essentiellement réservé aux stationnements sur rue. Il est considéré comme étant le type de stationnement sur rue le plus sûr et donc le plus recommandé.

➤ **Stationnement à angle**

Le stationnement à angle est un type de stationnement qui consiste à garer une automobile de biais, de sorte à former un angle de 90°, 75°, 60°, ou 45° avec la voie de circulation connexe.

1.3. Smart parking

Le stationnement intelligent, ou smart parking est une application de technologies moderne qui permet d'éliminer beaucoup besoins en peu de temps, et ceci est limité à la facilité de circulation et à la sécurité routière, aux places de réservation et aux différents moyens de paiement avec amélioration et la rapidité ce faisant. [4]

Le principe consiste à équiper chaque place de stationnement d'un capteur intelligent capable de détecter la présence d'un véhicule et d'informer en temps réel que la place est libre ou occupée. Le capteur est complètement autonome et ne nécessite donc aucune infrastructure à proximité, ce qui réduit les coûts d'investissement et surtout de maintenance. Ils peuvent également réserver leur place pour éviter qu'un autre conducteur ne prenne. Ils existent aussi des applications pour les téléphones portables qui faciliteront les déplacements des conducteurs dans les grandes villes.

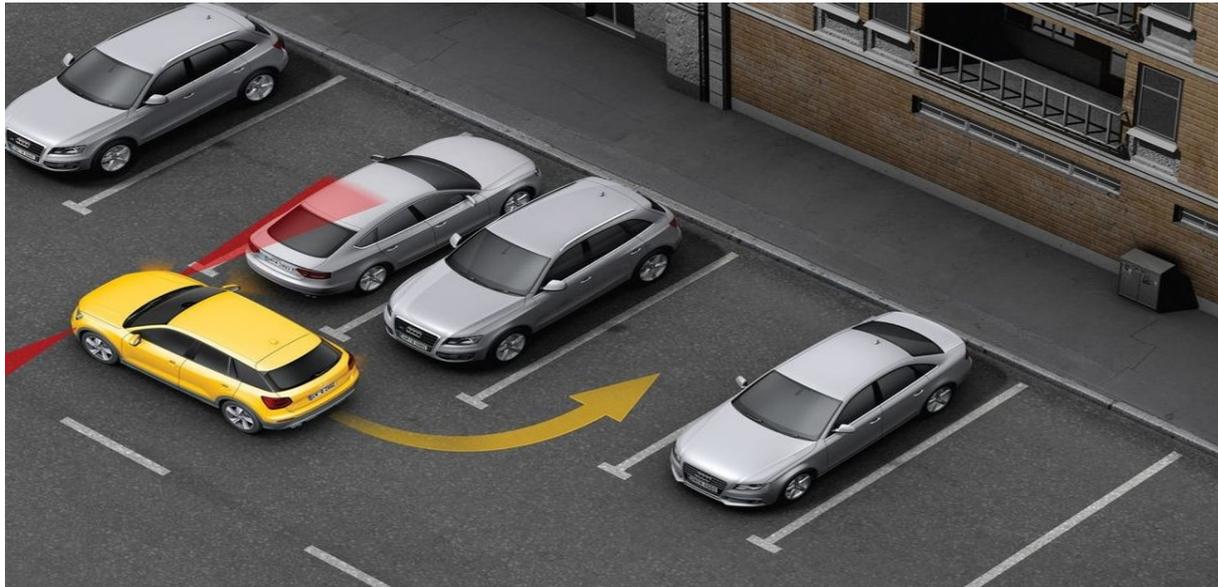


Figure 5: Stationnement intelligent des véhicules

1.4. Les défis et avantages de la technologie parking intelligent

1.4.1. Les besoins du stationnement

Une technologie de stationnement intelligente qui aidera à optimiser l'utilisation de l'espace de stationnement, à améliorer l'efficacité des opérations de stationnement et à faciliter la circulation.

- **Protection contre la pollution:** Avec le nombre de voitures qui augmente et par conséquent nous avons du mouvement dans les villes, la poursuite du trafic cause la pollution de l'air due au dégagement de gaz toxiques et la faciliter de stationnement aide à minimiser la circulation et donc à réduire la pollution.
- **La facilité de mobilité :** minimiser la gestion du trafic vous aide à naviguer facilement et ainsi économiser ton temps.
- **Exploitation des terres et des espaces vides comme lieux de stationnement.**
- **Manque de sécurité :** Le véhicule s'arrête à des endroits aléatoires et sans surveillance et constitue un danger pour le véhicule et le conducteur.
- **Apporter et fournir un service aux touristes:** Toute nouvelle personne dans la ville a du mal à trouver une place pour arrêter sa voiture.

1.4.2. Avantages de la technologie smart parking

- Obtenir des informations précises sur les endroits occupés ou non occupés en temps réel.
- Guider et transmettre les données du stationnement disponible et non occupé.
- La facilité d'utilisation des places de stationnement.
- Augmenter l'activité et se déplacer plus librement dans la ville en utilisant les technologies modernes.
- Assurer la sécurité du trafic pour les conducteurs et les utilisateurs.
- Avantage et profit du temps de recherche d'espace libre de stationnement

- Le stationnement intelligent joue un rôle clé dans la réduction de la pollution et la réduction de l'utilisation de l'essence et l'émission de gaz toxiques.
- Le stationnement intelligent permet une surveillance et une gestion améliorées et en temps réel de l'espace de stationnement disponible, ce qui entraîne une génération de revenus significative.
- Travailler sur la communication d'informations aux usages avant, pendant et après le stationnement. [4]

1.5. Conclusion

Le stationnement intelligent joue un rôle majeur dans le monde car il contribue à réduire la congestion et de nombreux problèmes avec la facilité de mobilité en ville et en maintenant le rythme de travail des employés pour qu'ils réservent la place de leur voiture via les communications et les téléphones intelligents. Les principaux facteurs favorisant le stationnement intelligent sont principalement les problèmes d'habitabilité urbaine, la mobilité des transports et la durabilité de l'environnement.

Dans ce chapitre nous avons donné une idée sur le smart parking, sa définition, ces besoins et avantages.

Chapitre II :
Description du
fonctionnement « smart
parking »

1. Introduction :

La conception de notre parking va porter sur deux aspects : l'aspect matériel et l'aspect logiciel.

La partie matérielle concerne la gestion des entrées et sorties à savoir : la gestion des barrières et l'actualisation de l'affichage du compteur indiquant le nombre de places encore disponibles ainsi qu'un détecteur de présence pour assurer la sécurité du parking en envoyant en temps réel un message d'alerte sur le portable du propriétaire via un module de transmission GSM.

Alors que la partie logicielle porte sur la gestion de l'application en actualisant son contenu en permanence et à la gestion des communications entre d'une part le parking et l'application et d'autre part la gestion des communications entre les conducteurs et l'application.

2. Partie matérielle

2.1. Le microcontrôleur Arduino UNO

L'Arduino UNO est une carte électronique basée autour d'un microcontrôleur Atmega328P du fabricant ATMEL. Il existe plusieurs autres types comme : le MEGA, DUO...etc. Le programme est stocké et exécuté par le microcontrôleur pour les différentes tâches.

On a spécialement choisi cette carte parce qu'elle est facile à utiliser et parfaitement adéquate pour notre projet actuel et en plus son prix sur le marché est assez abordable.

- **Présentation de la carte Arduino UNO**

L'intérêt principal des cartes ARDUINO est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple à travers un port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I2C, exploitation de servomoteurs.[5]

- **Description du matérielle**

L'Arduino UNO est une carte électronique composée de plusieurs parties qui contribuent à son fonctionnement dont on va citer les principales tout en expliquant leurs rôles.

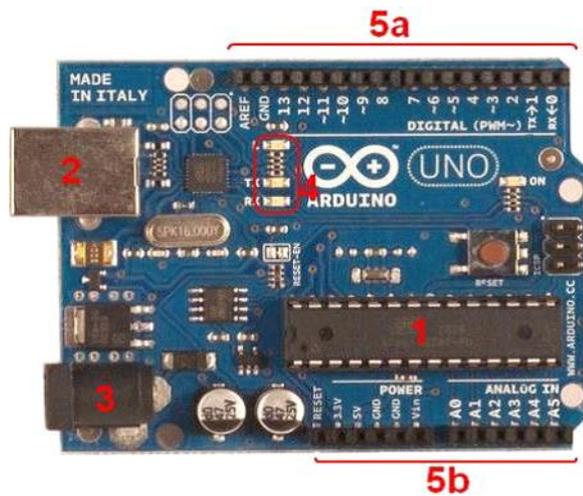


Figure 6: carte microcontrôleur Arduino UNO

1: le microcontrôleur

Le cerveau de la carte. C'est lui qui va recevoir le programme que nous allons créer et va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir faire des choses, dont nous avons besoins. Il est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée :

- Le processeur : C'est le composant principal du microcontrôleur. C'est lui qui va exécuter le programme que nous lui donnerons à traiter. On le nomme souvent le CPU.
- La mémoire Flash : c'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible (c'est la même qu'une clé USB par exemple).
- RAM : c'est la mémoire dite "vive", elle va contenir les variables de notre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur (comme sur un ordinateur).
- EEPROM : c'est le disque dur du microcontrôleur. On peut y enregistrer des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.
- Les registres : c'est un type de mémoire utilisé par le processeur.
- La mémoire cache : c'est une mémoire qui fait la liaison entre les registres et la RAM

2 et 3 : Alimentation

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en 2) ou bien par une alimentation externe (en 3) qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. Pas de danger de tout griller, il faut respecter l'intervalle de 7V à 15V.

4 : Visualisation

Les trois "points blancs" entourés en rouge (4) sont des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.

Nota : Quand on branche la carte au PC, elle clignote quelques secondes.

Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une, pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

5a et 5b : la connectique

La carte Arduino ne possédant pas de composants qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte (en 5a et 5b).

2.2. Le module de communication GSM

Pour notre application nous avons besoin d'une communication de longue distance pour recevoir les informations sur notre consommation et recevoir des alertes même en étant loin. Pour cela on a opté pour le module GSM/GPRS. Le module GSM/GPRS de chez SeedStudio est une carte d'interface compatible avec Arduino. Elle permet d'envoyer et recevoir des SMS, des données ou des communications vocales depuis le réseau mobile. Le module est basé sur le circuit SIM900 de la société SIMCOM. Il est contrôlé via les commandes AT depuis une carte Arduino. Le module est livré avec une antenne patch déportée. Un connecteur au dos de la platine est prévu pour recevoir une carte SIM ainsi qu'une pile Lithium CR1220. La communication entre le module et une carte Arduino est réalisée par la liaison série asynchrone : UART ou une liaison série logicielle.



Figure 7: module GSM SIM900

2.2.1. Description du matériel

Il existe plusieurs modules GSM : SIM900, SIM800, Shield V2 ...etc. Le GSM permet à une carte Arduino de se connecter à Internet, effectuer /recevoir des appels vocaux et envoyer et recevoir des messages SMS. Le Shield utilise un modem radio M10 de Questel. Il est possible de communiquer avec le Shield en utilisant des commandes AT qui sont des jeux de commande textuelle. La bibliothèque GSM dispose d'un grand nombre de méthodes de communication avec le Shield.[6]

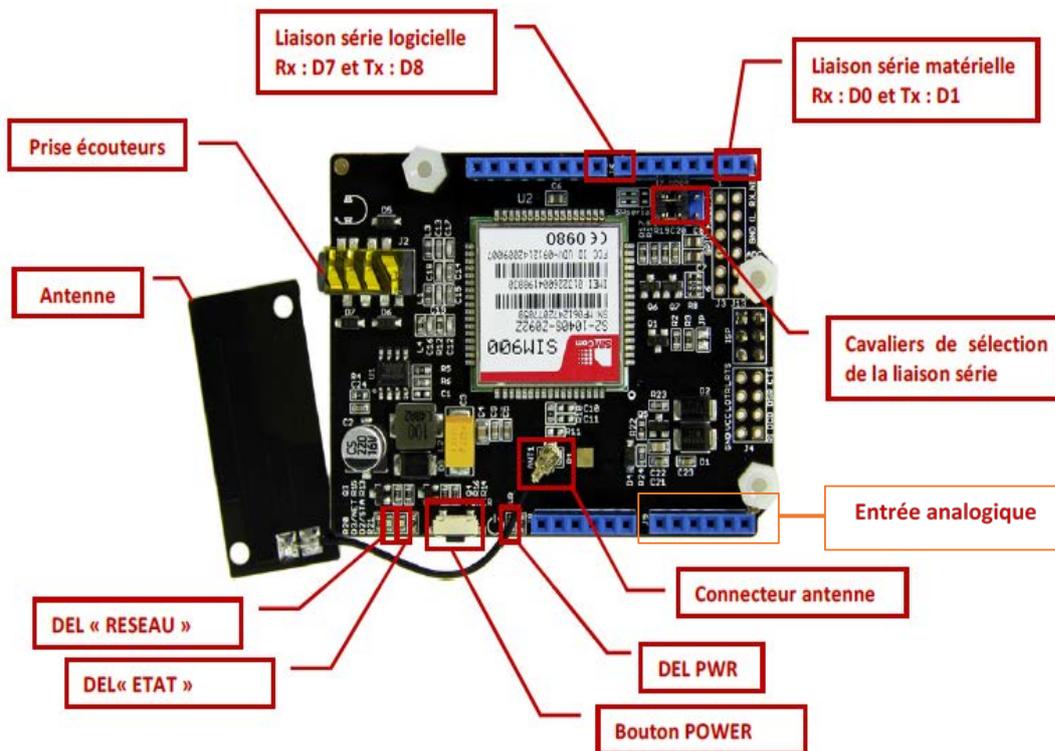


Figure 8: Description module GSM

Le Shield contient :

- Une antenne intégrée.
- 6 Broches d'entrées analogiques : pour brancher des sondes analogiques de mesures et y programmer des seuils minimum et maximum et en cas de franchissement des seuils on reçoit un SMS.
- 14 Broches numériques : Entrées/Sorties.
Entrées digitales ou numériques : pour brancher des détecteurs du type : Intrusion, présence, contact de porte, détecteur de fumée, Gaz, inondation, thermostat, Hygrométrie, ...etc.
Les sorties sont activées en envoyant au modem GSM un SMS codé, un relais de puissance est alors activé pour commander tous matériels (équipements électriques) : Portes, portails, chauffages, climatisation, volets, moteurs, pompes, ...etc.
- Deux petits boutons : RESET (est lié à la broche de réinitialisation Arduino) et POWER (allume et éteint le modem).
- Prise audio intégrée.
- Des indicateurs embarqués (voyants d'état) : On, Statut et Net

On : indique que le bouclier est alimenté.

Statut : s'allume lorsque le modem est sous tension et que des données sont transférées vers / depuis le réseau GSM / GPRS.

Net : clignote lorsque le modem communique avec le réseau radio.

- 5 Broches d'alimentation.

2.2.2. Mise sous tension

La mise sous tension du Shield est réalisée de manière matérielle via le bouton « POWER ». Cette mise sous tension peut être réalisée également de manière logicielle en appliquant un niveau logique haut sur la broche D9.

2.2.3. LEDs

Les indications données par les 3 LEDs sont :

La LED PWR de couleur verte indique, lorsqu'elle est allumée, que le Shield est sous tension et lorsqu'elle est éteinte le Shield est hors tension.

2.2.4. Caractéristiques techniques

Le M10 est un modem quadri-bande GSM / GPRS avec les caractéristiques techniques qui montré sont sur le *Tableau 2*.

Tableau 1: Caractéristique techniques du modem GSM/GPRS type M10

Fréquence	GSM 850 MHz, GSM 900 MHz DCS 1800 MHz et PCS 1900 MHz
Protocoles	TCP/UDP et HTTP via une connexion GPRS
La vitesse maximale de transfert de données sur les liaisons montantes et descendantes GPRS	85,5 kbps
Tension de fonctionnement	5V
Puissance	Il est recommandé d'alimenter la carte avec une liaison externe pouvant fournir entre 0.7 et 1 A
Dimension	100 x 53 mm

- **Branchement**

La broche 2 est connectée à la broche TX du M10

La broche 3 à la broche RX.

La broche PWRKEY du modem est connectée à la broche 7 d'Arduino

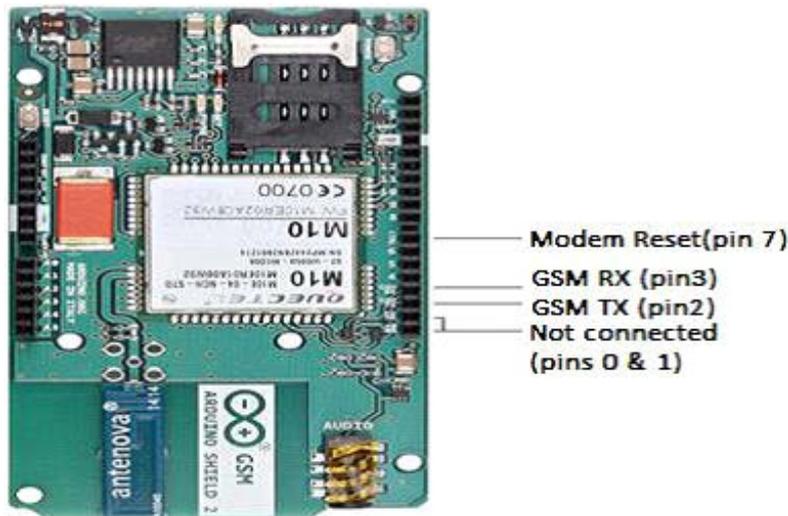


Figure 9: les broches du module GSM

2.3. Le système GPS :

Le Système de Positionnement Global (GPS) est un système de navigation par satellites destiné à fournir la position, la vitesse et l'heure n'importe où à la surface de la terre ou dans ses environs.

La précision du positionnement par GPS va de 100 mètres à quelques mètres en temps réel. Elle peut atteindre quelques millimètres pour les applications de géodésie en temps différé. Bien qu'il ait des milliers d'utilisateurs civils du GPS à travers le monde, le GPS a été créé et est contrôlé par le département de la défense américain (U.S Department of Defense).[7]

Potentiel d'utilisation :

Le système GPS permet de calculer la position tridimensionnelle (latitude, longitude, altitude) d'un utilisateur, de manière continue et instantanée, en tout endroit sur terre, aussi il permet de déterminer sa vitesse et sa direction. De plus, le système GPS fournit une information temporelle, c'est le temps de GPS.

Conçu à l'origine pour des fins de navigation militaire, le système GPS a vite été utilisé pour des fins de localisation et de positionnement tant pour les civils que pour les militaires.

Le système GPS est une solution potentielle à presque toutes les applications nécessitant une référence spatiale (coordonnées géo référencées) telles que la navigation, la gestion de transport, la circulation aérienne, la géodésie, l'hydrographie, la foresterie, et bien d'autres encore.

2.4. Servomoteur :

Un servomoteur est un moteur électrique spécifique pour un déplacement angulaire. Un servomoteur permet de se positionner précisément sur un angle. La plage de l'angle est généralement de 0 à 180 degrés.

Un servomoteur contient un moteur à courant continu, un réducteur à roues dentées à axes parallèles et une électronique de commande.

L'alimentation et la commande se font par un câble de trois fils, un commun, un fil d'alimentation et un fil de commande. Les couleurs sont conventionnelles pour un constructeur.[8]



Figure 10: Servomoteur

Caractéristiques du SG90 :

- Dimensions : 22 x 11.5 x 27 mm
- Poids : 9g
- Matériel principal: Plastique
- Couleur principale: Bleu
- Vitesse : 0.12 s / 60° sous 4.8V - 0,10 s / 60° sous 6.0V
- Couple de décrochage : 1.2Kg/cm sous 4.8V
- Tension d'alimentation : 4.8V à 6V
- Amplitude : de 0 à 180°

Exemple de signal de commande pour un servomoteur pouvant varier de 0 à 180°.

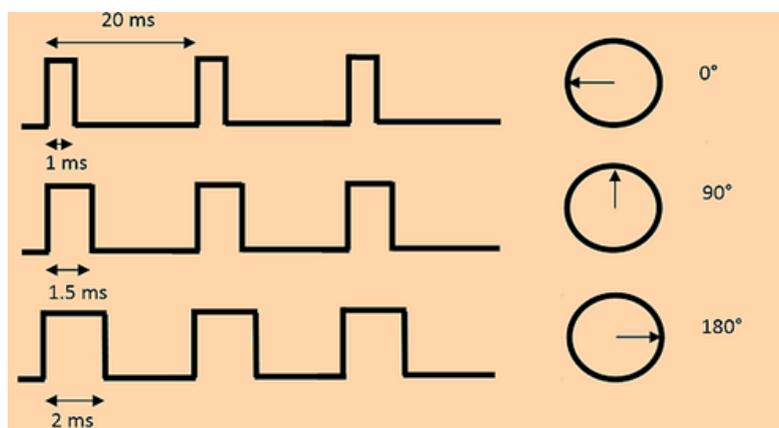


Figure 11: Signal de commande variant de 0 à 180°

2.5. L'afficheur LCD (Liquid Crystal Display)

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage de paramètres de fonctionnement. Grâce à la commande par un microcontrôleur ces afficheurs permettent de réaliser un affichage de messages aisés. Ils permettent également de créer ses propres caractères.

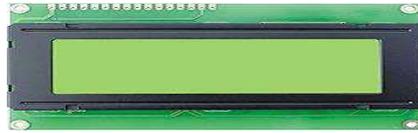


Figure 12: Afficheur LCD 2 x 16

- **Fonctionnement d'un afficheur LCD :**

L'affichage comporte d'autres éléments que l'afficheur LCD seul. Il s'agit bien évidemment de:

- Le rétroéclairage qui permet d'éclairer l'écran afin de permettre la visibilité des valeurs.
- Le contrôleur LCD qui permet la jonction avec l'extérieur (ici notre microprocesseur Arduino Uno) afin de recevoir les données à afficher et commande l'affichage
- Et le circuit intégré de commande qui se charge de la gestion du module

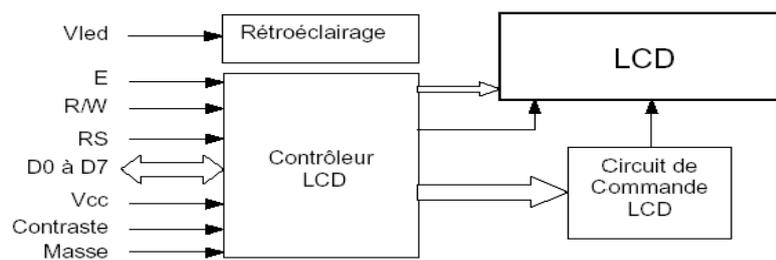


Figure 13: Configuration des pins

- **Rôle des différentes broches de l'afficheur LCD :**

- VCC, Masse : alimentation de l'afficheur LCD. Un afficheur LCD s'alimente en 0V 5V.
- Contraste : entrée permettant de régler le contraste de l'afficheur LCD. Il faut appliquer une tension continue réglable (entre 0V et 5V) à l'aide d'un potentiomètre.
- Vled : différence de potentiel permettant de commander le rétro éclairage.
- E : entrée de validation (ENABLE), elle permet de valider les données sur un front descendant. Lorsque E=0 alors le bus de données est à l'état haute impédance.
- RS : Register Select cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (RS=0) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (RS=1).
- R/W : entrée de lecture (R/W=1) et d'écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.
- D7...D0 : bus de données bidirectionnelles, il permet de transférer les instructions ou les données à l'afficheur LCD.

- **La mise en œuvre d'un afficheur LCD :**

Un afficheur LCD sera commandé par un microcontrôleur.

Il faut donc penser aux mises en œuvre :

- Matérielle : connexion des broches du microcontrôleur à l'afficheur LCD,
- Logicielle : utilisation de sous programmes permettant de commander l'afficheur LCD (initialisation, effacement de l'afficheur, affichage d'un caractère, affichage d'une variable,)

2.6. Capteur de détection d'obstacle infrarouge IR :

Le concept de base de la détection d'obstacles IR (infrarouge) est de transmettre le signal IR (rayonnement) dans une direction et un signal est reçu au récepteur IR lorsque le rayonnement IR rebondit à partir d'une surface de l'objet.[9]



Figure 14: Capteur de détection IR (infrarouge)

Caractéristiques:

- Tension de fonctionnement: 3 - 5V DC
- Type de sortie: sortie de commutation numérique (0 et 1)
- Trous de vis de 3 mm pour un montage facile
- Taille de la planche: 3,2 x 1,4 cm
- Distance de détection: 2 ~ 20 cm
- Angle de détection: 35 ° degré

2.7. Le capteur de mouvement PIR SENSOR3

Le capteur de mouvement PIR (Passive InfraredSensor) est un senseur électronique qui mesure la lumière infrarouge (IR) rayonnant à partir d'objets dans son champ de vision. Ils permettent de déterminer si un être humain ou (un animal) est entrée où sorti du champ de détection du module. Ils sont très souvent utilisés dans les systèmes d'alarmes ou de détection de présence pour leur faible coût et leur efficacité.[10]

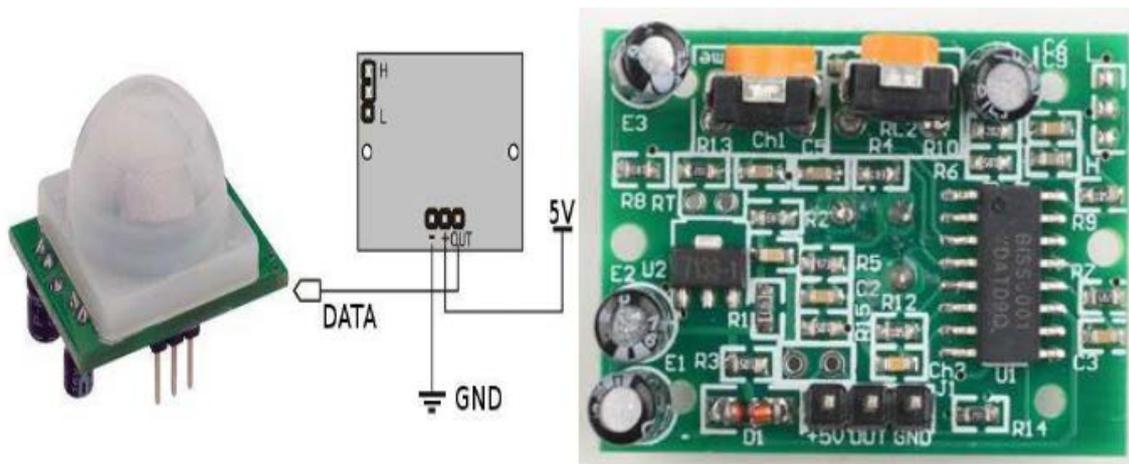


Figure 15: Capteur de mouvement

Les applications d'un capteur PIR sont multiples:

- Détection de mouvement et activation d'une œuvre interactive.
- Détection de passage/intrusion.
- Commande d'une chatière automatisée.
- Détecteur pour commande d'installation domotique

Le fonctionnement d'un capteur PIR n'est pas compliqué à comprendre. Il possède 3 pins : l'alimentation, la masse et le pin data qui va nous intéresser. Par défaut, celui-ci est à l'état bas. Quand le capteur détectera un mouvement cette sortie passera à l'état haut pendant une durée comprise entre 5s et 2min réglable grâce à un des deux petits potentiomètres situés à l'arrière du capteur, l'autre permettant de régler la sensibilité de 3 à 7m. Pour les détails techniques, ce capteur est basé sur un circuit-intégré BISS001, en regardant le datasheet on s'aperçoit que c'est un CI spécialisé pour cette application, qu'il opère entre 3 et 5V et qu'il consomme très peu de courant, il est donc idéal pour une utilisation à piles.[10]

Caractéristiques du capteur PIR sont:

- Dimensions: 32 x 24 x 27H mm
- Voltage: 5-12VDC □ Output: 3,3V TTL
- Détection Distance: 3-7mt (approx, ajustable)
- Delay Time: 5-200s (ajustable)
- Trigger: L: non répétable trigger -H: répétable trigger

2.8. Horloge à temps réel RTC DS1307 :

Le module DS1307 est une horloge temps réel (aussi appelé "RTC", "Real Time Clock"). C'est une horloge numérique autonome qui donne l'heure quand on la lui demande. Ce genre d'horloge est très utile dans des projets de mesure de grandeurs physiques avec horodatage par exemple. Ce module RTC est capable de gérer l'heure (heures, minutes, secondes) et la date (jours, mois, année) tout en s'occupant des mois de 30 ou 31 jours, des années bissextiles, etc. Le calendrier intégré dans le module DS1307 est valable de l'an 2000 à l'an 2100, ce qui devrait être suffisant pour la plupart des projets. PS Le module dérive de quelques secondes

par jours en moyenne. Cela dépend de la température ambiante et de la qualité du quartz d'horloge.[11]

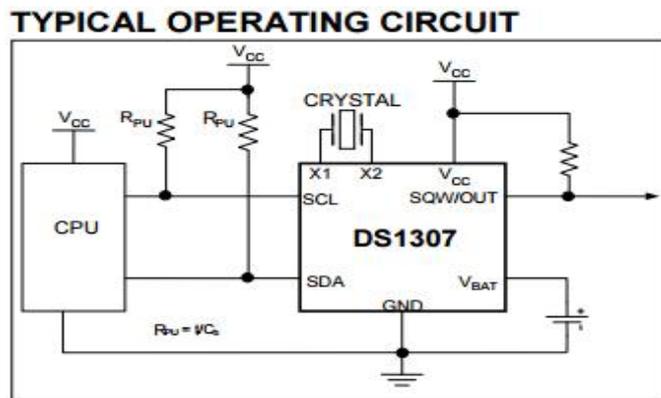


Figure 16: circuit du DS1307

La communication avec le microcontrôleur maître se fait via un bus I2C. Le module dispose de tout le nécessaire pour garder en mémoire l'heure en cas de coupure d'alimentation grâce à une batterie externe. Une simple pile bouton permet de garder l'heure et la date à jour durant plusieurs années sans alimentation. Le module DS1307 ne dispose pas de fonctionnalité "alarme" contrairement à d'autres modules RTC plus haut de gamme.

Le module DS1307 dispose cependant d'une sortie "base de temps" permettant d'avoir un signal logique à une fréquence fixe (1 Hertz par exemple) pour faire fonctionner un circuit ou un compteur externe. Cela peut être utile dans certaines applications.[11]

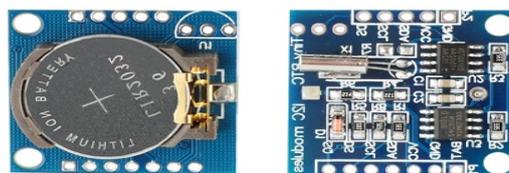


Figure 17: Module RTC DS1307

- **Principe d'utilisation du module :**

Le module DS1307 se comporte comme une petite mémoire externe I2C contenant l'heure et la date à des emplacements fixes. La plage mémoire du module est d'une taille impressionnante de 64 octets. Ce n'est pas énorme, mais il ne faut pas beaucoup de mémoire pour stocker une date et une heure. Lire la mémoire du module revient à lire l'heure et la date courante. Écrire la mémoire du module revient à mettre à jour l'heure et la date.[11]

- **Brochage du DS1307 :**

L'horloge temps réel DS1307 est commercialisée dans un boîtier de 8 broches classiques, comme illustré dans le schéma suivant :

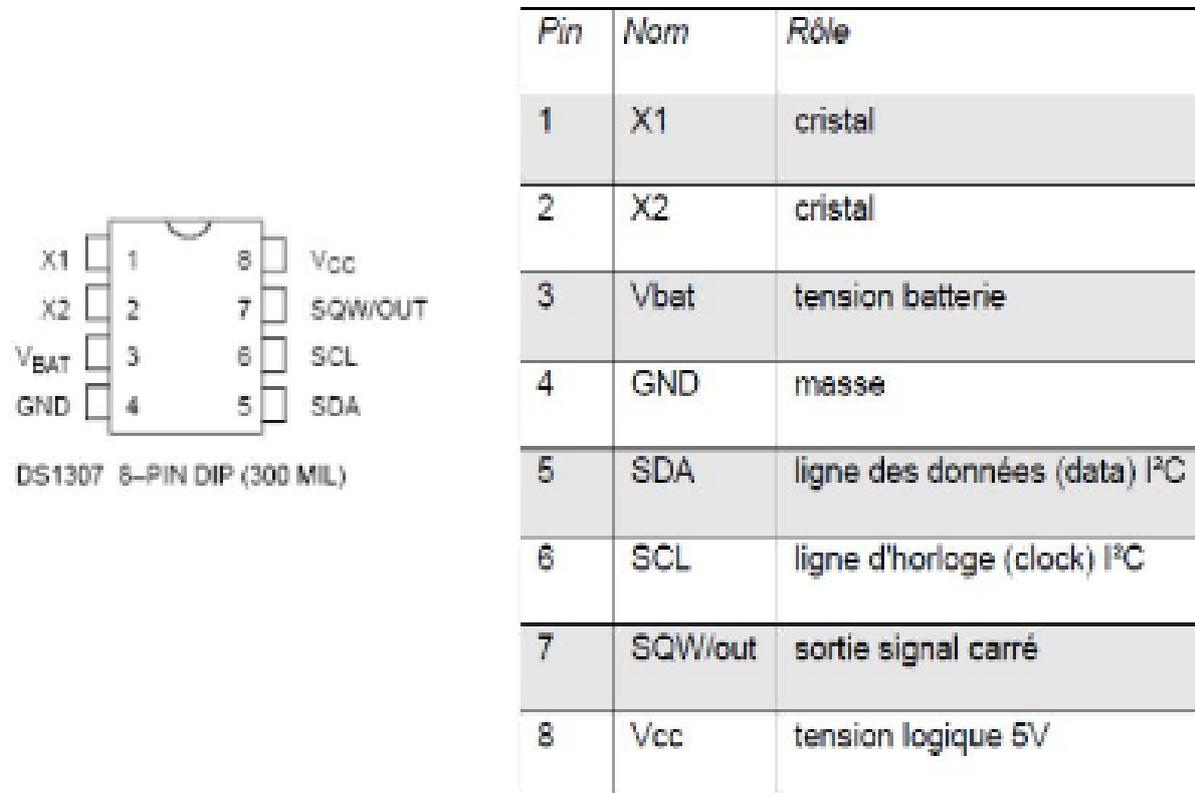


Figure 18: Boîtier de 8 broches du DS1307

3. Partie logicielle

3.1. Arduino IDE :

L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino, dans un langage simple qui écrit en java et basé sur le modèle du langage de traitement. L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. Ce logiciel peut être utilisé avec n'importe quel tableau Arduino.

Le logiciel Arduino a pour fonctions principales :

- Pouvoir écrire et compiler des programmes pour la carte Arduino
- Se connecter avec la carte Arduino pour y transférer les programmes
- Communiquer avec la carte Arduino

Le logiciel Arduino intègre également un Terminal Série (fenêtre séparée) qui permet d'afficher des messages textes reçue de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facile des programmes, permet d'afficher sur PC l'état de variables, de résultats de calculs ou de conversions Analogiques-Numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.[5]

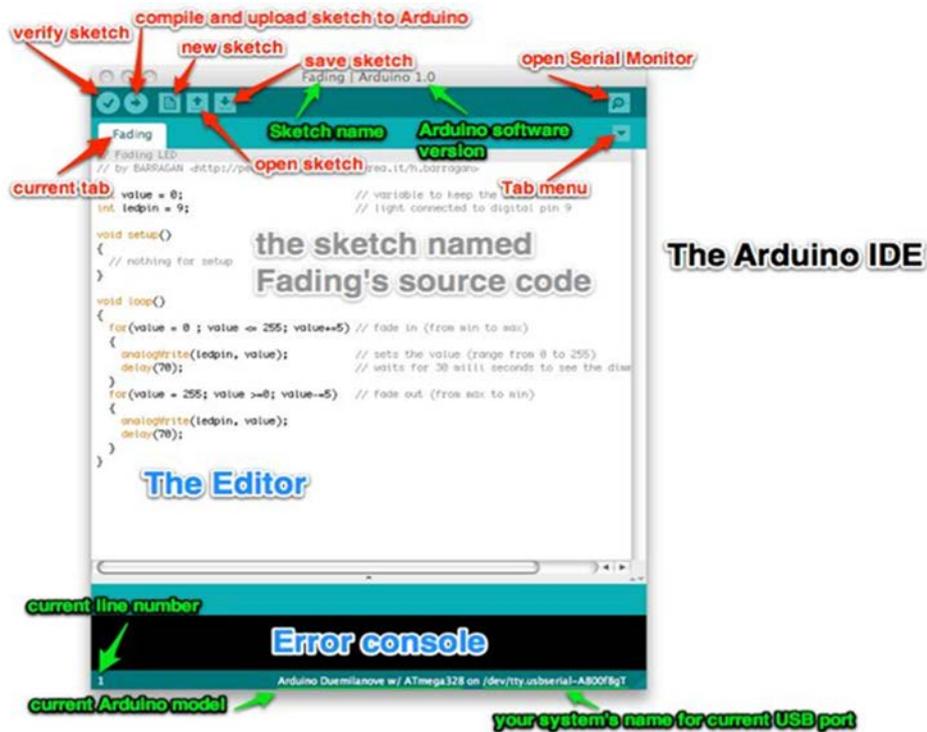


Figure 19: L'interface de logiciel Arduino IDE

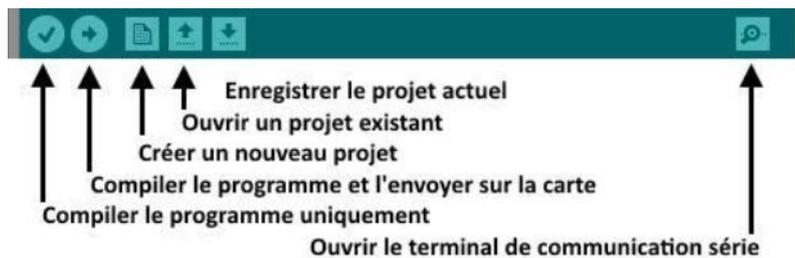


Figure 20: Les composants d'une barre d'action

3.2. Les commandes AT

Les commandes AT sont des jeux de commandes textuelles permettant de gérer la plupart des modems ou des modules GSM. Ces commandes commencent toujours par les lettres « AT » et se terminent obligatoirement par un retour chariot.

AT : Code d'attention. C'est le préfixe d'une ligne de commande qui indique au modem qu'une commande ou une séquence de commandes va être envoyée.[12]

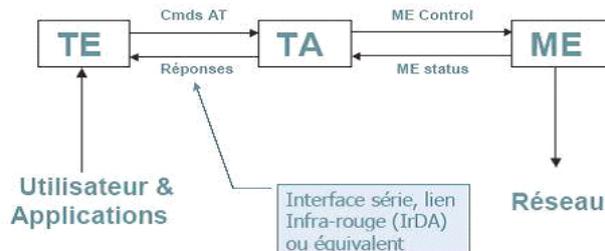


Figure 21: Schéma fonctionnel des commandes AT

- TE (Terminal Equipment) : envoi et affiche les Commandes
- ME (Mobile Equipment)
- TA (Terminal Adaptator) : interface entre l'utilisateur et le mobile

Les types des commandes AT et les réponses

- **Commande de test : AT+<X>=?**
Pour la commande test, le ME renvoie la liste des paramètres et des plages de valeurs définies avec la commande d'écriture correspondante par processus internes.
- **Commande pour lire : AT+<X> ?**
Elle renvoie la valeur actuellement définie du paramètre ou des paramètres.
- **Commande pour écrire : AT+<X>=<...>**
Cette commande définit les valeurs de paramètre défini par l'utilisateur.
- **Commande d'exécution : AT+<X>**
La commande d'exécution lit les paramètres non variables affectés par les processus internes dans le GSM.

3.3. LES LOGICIELS DE SIMULATION ELECTRONIQUE

Il existe plusieurs logiciels de simulation des circuits électronique comme PROTEUS par exemple Multisim qui est un logiciel de simulation qui est plutôt performant dans la simulation analogique et numérique de circuits intégrés. Par contre s'il s'agit de simuler un microprocesseur, Multisim présente des faiblesses c'est pourquoi pour la simulation de notre prototype, PROTEUS est la meilleure option.

3.3.1. Le logiciel PROTEUS

PROTEUS est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans PROTEUS permettent la CAO (Conception Assisté par Ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : ISIS et ARES [13]. Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle.

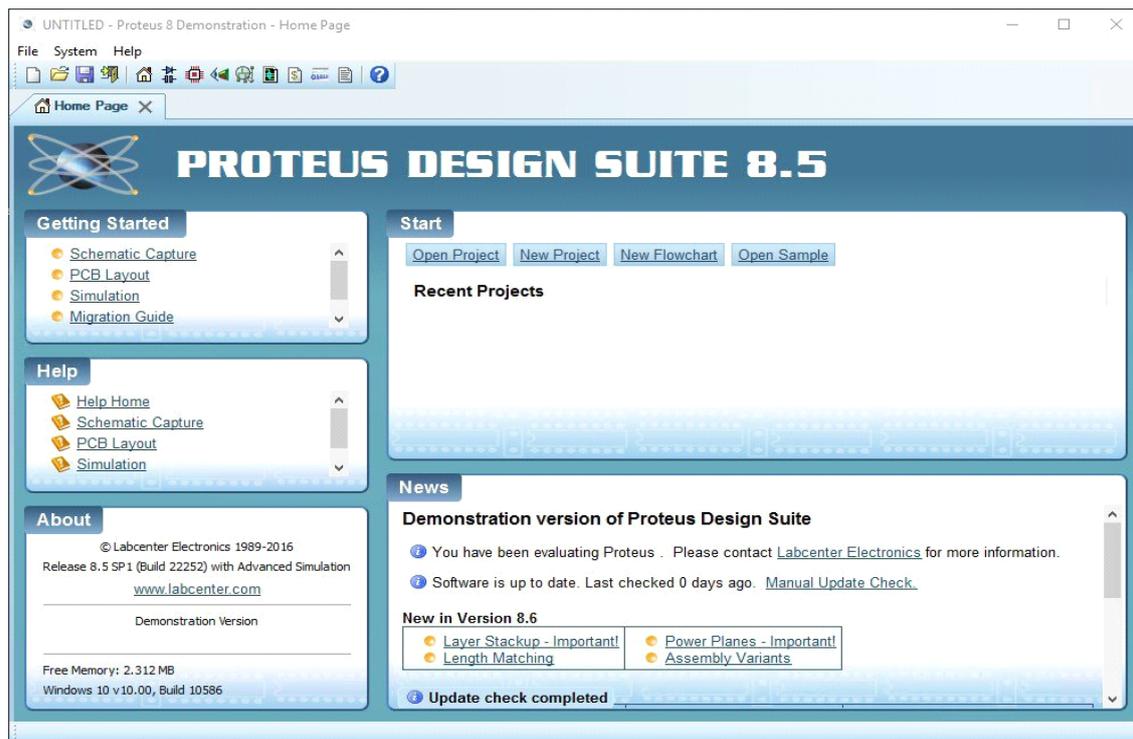


Figure 22: Interface du logiciel PROTEUS

Outre la popularité de l'outil, PROTEUS possède d'autres avantages :

- Pack contenant des logiciels facile et rapide à comprendre et prendre en main,
- Le support technique est performant,
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

3.3.2. L'interface ISIS

Ce sont les sigles d'Intelligent Schematic Input System et c'est le programme qui permet de dessiner sur le plan du circuit électrique, avec toute sorte de composants comme des résistances, des bobines, des condensateurs, des sources d'alimentation ou même des microprocesseurs. Le logiciel ISIS de PROTEUS est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits [14].

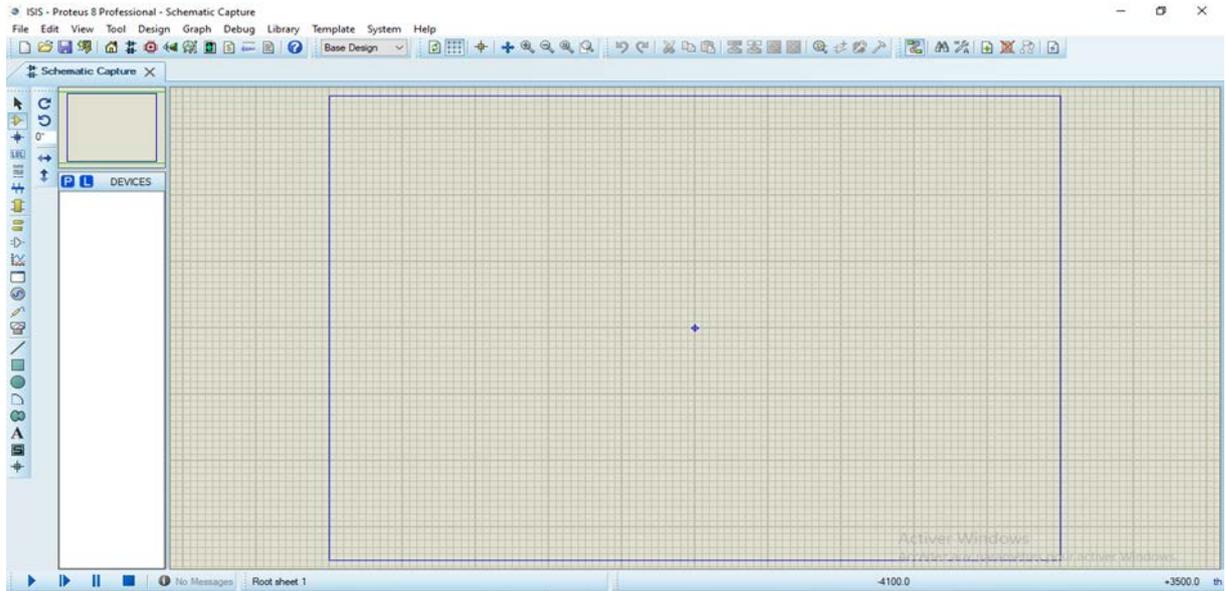


Figure 23: Interface ISIS

3.3.3. L'interface ARES

Le logiciel ARES est un outil d'édition et de routage qui complètement parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficiente lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement.

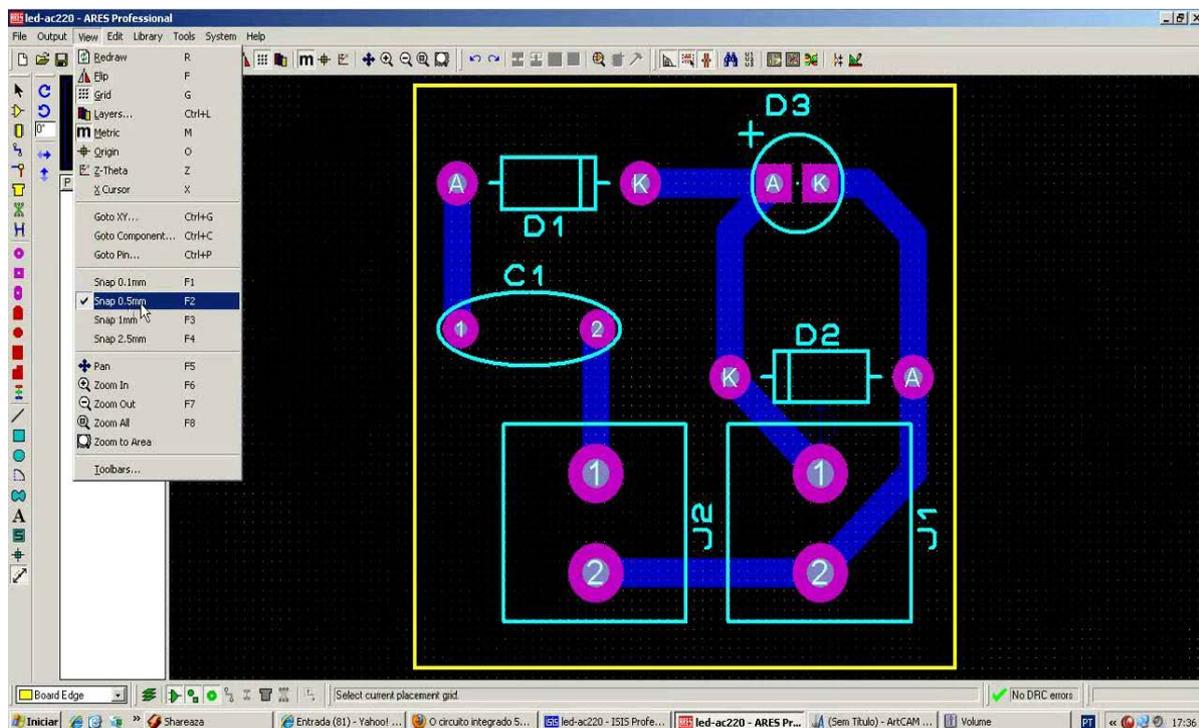


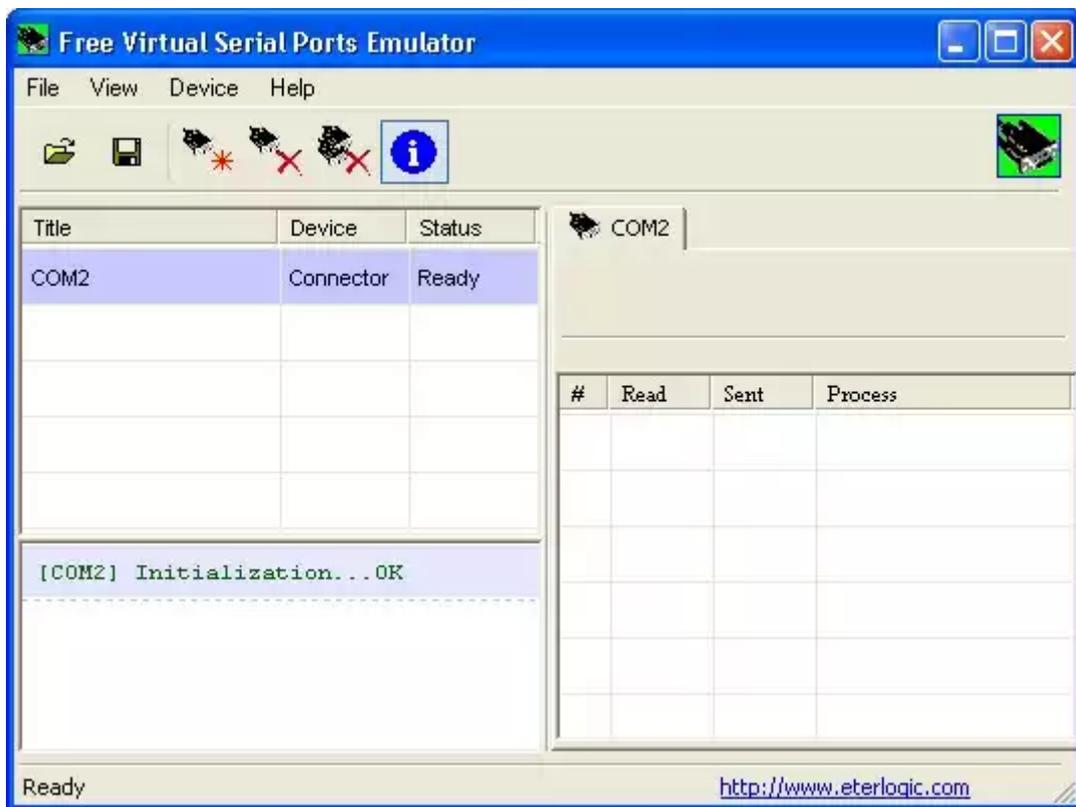
Figure 24: Interface du logiciel PROTEUS ARES

3.3.4. Virtual Serial Ports

L'utilitaire Virtual Serial Ports Emulator (VSPE) est un outil destiné aux personnes ayant de très bonnes bases en programmation. En effet, ce logiciel permet d'émuler, de créer, de tester et de déboguer des ports virtuels. De cette manière, des applications n'auront plus besoin de passer par un port de série, mais pourront utiliser un port virtuel. L'intérêt étant que plusieurs utilisateurs pourront donc y accéder simultanément.

Avec lui, vous pourrez donc créer un port parallèle virtuel, partager des données physiques pour plusieurs applications, exposer le port à un réseau local via le protocole TCP, etc.

Dans notre projet ce logiciel nous permet d'octroyer un port virtuel afin de pouvoir communiquer avec notre application Blynk vu qu'en réalité notre prototype né pourrait pas communiquer directement avec l'application du ce port virtuel permettra à l'application de se connecter.



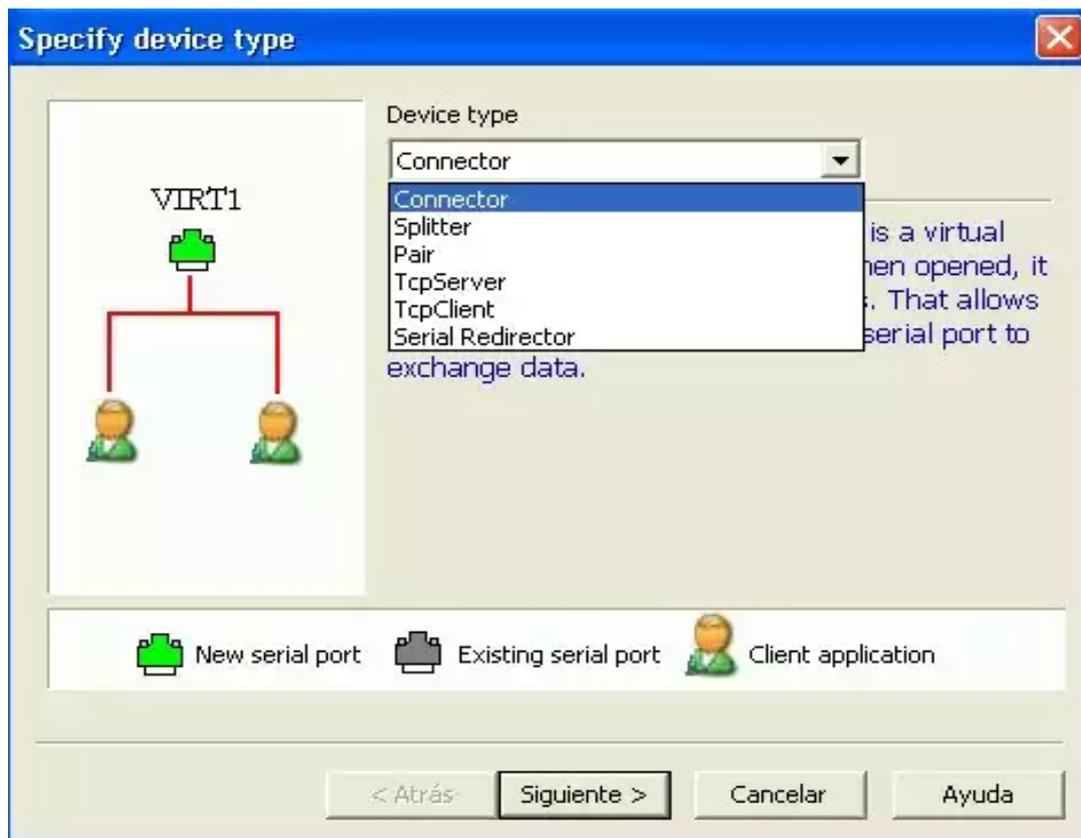


Figure 25: L'interface Virtual Serial Ports

3.4. L'application mobile Blynk :

Blynk a été conçu pour l'Internet des Objets. Il peut contrôler un hardware à distance, il peut afficher des données de capteur, il peut stocker des données, les visualiser et faire beaucoup d'autres trucs.

Il y a trois composants majeurs dans la plateforme:

- **Application Blynk** : Permet de créer de fantastiques interfaces pour vos projets utilisant différents widgets qu'on fournit.
- **Serveur Blynk** : Responsable de toutes les communications entre le smartphone et le hardware. Il peut utiliser le nuage (Cloud en anglais) Blynk ou faire tourner son Serveur privé Blynk localement. C'est open-source, ça peut facilement gérer des milliers de périphériques.
- **Bibliothèque Blynk** : Pour toutes les plateformes hardware populaire, active la communication avec le serveur et traite toutes les commandes entrantes et sortantes.[15]



Figure 26: Interface de Blynk

- **Fonctionnalité :**

Tous les hardwares et périphériques supportés Connexion au nuage via:

- Ethernet
- Wifi
- Bluetooth...
- Collection de widgets faciles à utiliser.
- Manipulation des broches directes sans code à écrire.
- Facilité d'intégrer et ajouter de nouvelles fonctionnalités en utilisant les broches virtuelles.
- Communication Périphérique-à-Périphérique en utilisant le widget Bridge envoi d'emails, de notifications push,... etc.[15]

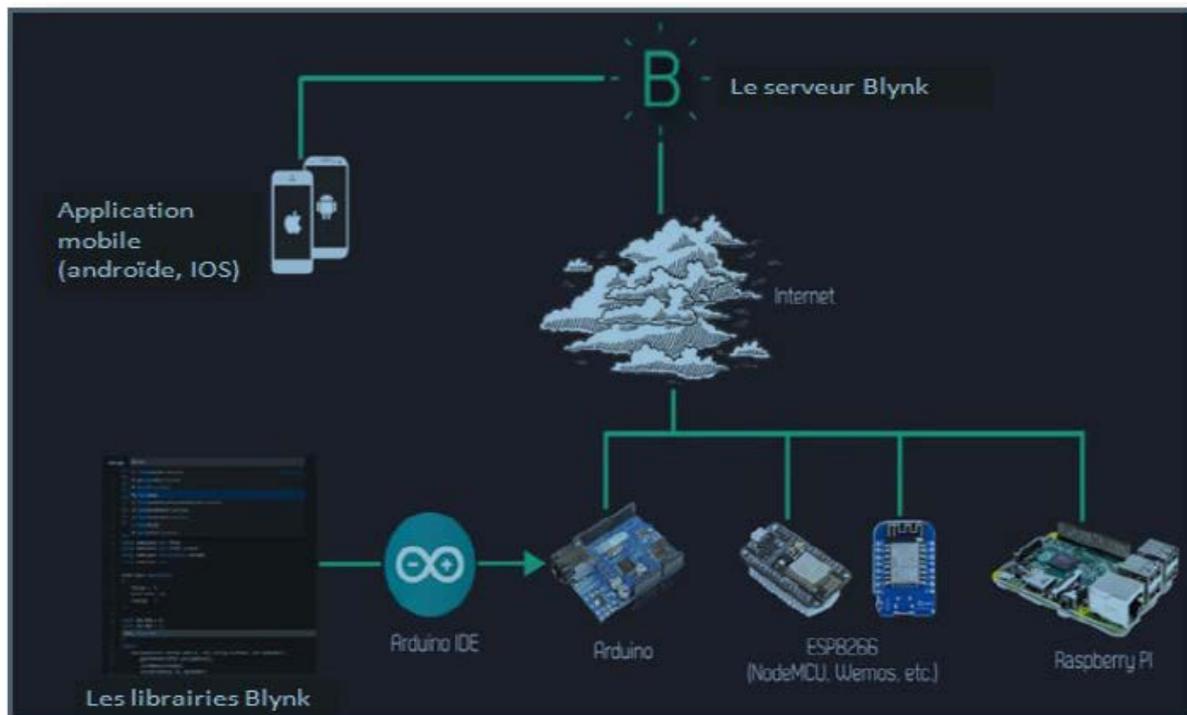


Figure 27: Fonctionnement de Blynk

L'application est composée de deux parties :

- La partie design : Cette partie est composé d'un écran constitué de :
 - Un afficheur LCD : Qui permet d'afficher les données de notre simulation c'est à dire le nombre total de place, le nombre de place libre et le nombre de place réservé ;
 - Un bouton RESERVER : C'est un bouton cliquable qui une fois cliqué donne une impulsion au niveau de notre microordinateur et nous permettant alors de réserver une place dans notre parking ;
 - Un Timer : Le timer nous indique le temps pendant lequel la réservation est d'actualité. Une fois le délai dépasser la réservation sera annulé ;
 - Un moniteur: Qui nous affiche si les informations sur la réservation.
- La partie Codes : Dans cette partie on a utilisé des codes pour donner des instructions aux différents éléments de notre application.

4. Conclusion

Ce chapitre nous a permis de comprendre les grands blocs de fonctionnement du prototype du parking intelligent, de connaître les notions de bases et les caractéristiques des composants et modules que nous utilisons dans le projet. De comprendre leurs modes de fonctionnement ainsi que les logiciels utilisés. Les équipements choisis pour la réalisation du prototype ont été

sélectionnée en se basant sur beaucoup de critères comme le coût des équipements ou bien la performance du matériel.

Chapitre III :

Simulation du

fonctionnement du système

1. Introduction

Dans le chapitre précédent on a essayé de voir le model général du notre prototype du parking intelligent et présenté les équipements utilisés. Dans ce chapitre on va voir la conception du model en utilisant les différents équipements et nous procédons à la simulation du prototype sous l'interface PROTEUS, voir les fonctions de chaque équipement au sein du projet. Pour cela nous commençons par faire le câblage sous l'interface PROTEUS et ensuite nous passerons aux codes dans l'IDE Arduino.

2. Connexion des différents équipements sous PROTEUS

Les équipements choisis et décrits dans le chapitre précédent vont être raccorde dans l'interface PROTEUS comme suite :

On ne peut pas commencer la conception ou la simulation de notre prototype sans avoir un processeur pour traiter l'information afin d'automatiser notre parking. Donc pour cela on a simulé sur proteus on utilisant une carte Arduino UNO. On a procédé comme suite :

- Ouvrir le logiciel PROTEUS
- Se rendre sur ISIS
- Cliquer sur P (Pick Device) qui nous permet de sélectionner les équipements pour notre simulation
- Rechercher ARDUINO qui est le microcontrôleur qui nous permet le traitement de données et valider la sélection avec OK.

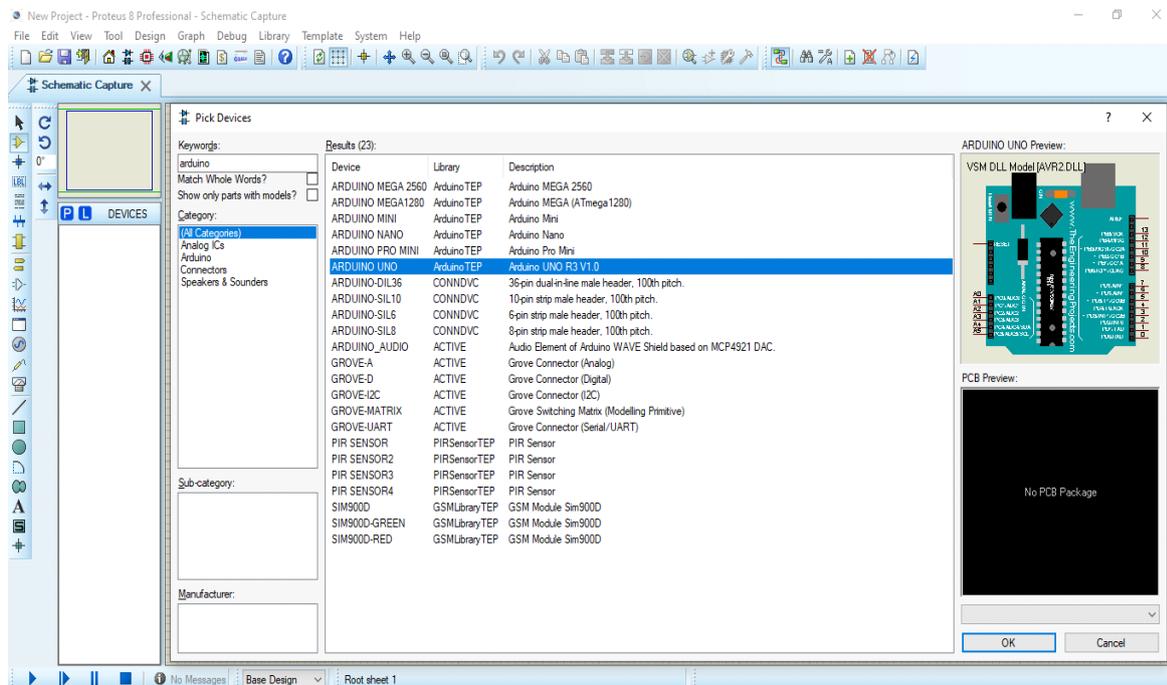


Figure 28: Insertion de l'équipement Arduino dans l'interface ISIS

Et on procède de la même façon pour sélectionner tous les équipements dont on a besoin pour notre simulation c'est-à-dire :

- Arduino UNO (le micro-Controller)
- SERVO MOTEUR (qui nous sert de barrière)
- GPS (module GPS)
- LOGICTOGGLE (bouton pour simuler les mouvements)
- Le LM044L (L'écran LCD pour l'affichage de données et l'heure en temps réel composé de 4 lignes et 20 colonnes)
- Le SIM900D (qui est le module GSM)
- PIR SENSOR3 (le capteur de mouvement)
- IR SENSOR (des capteurs infrarouges p)
- MINRES 10k (des résistances de 10 k ohm pour le calibrage du LCD)
- COMPIM (Port virtuel pour connecter avec l'application mobile Blynk)
- RTC (REAL TIME CLOCK)

Après avoir passé l'étape de la sélection des équipements on procède au câblage comme suite :

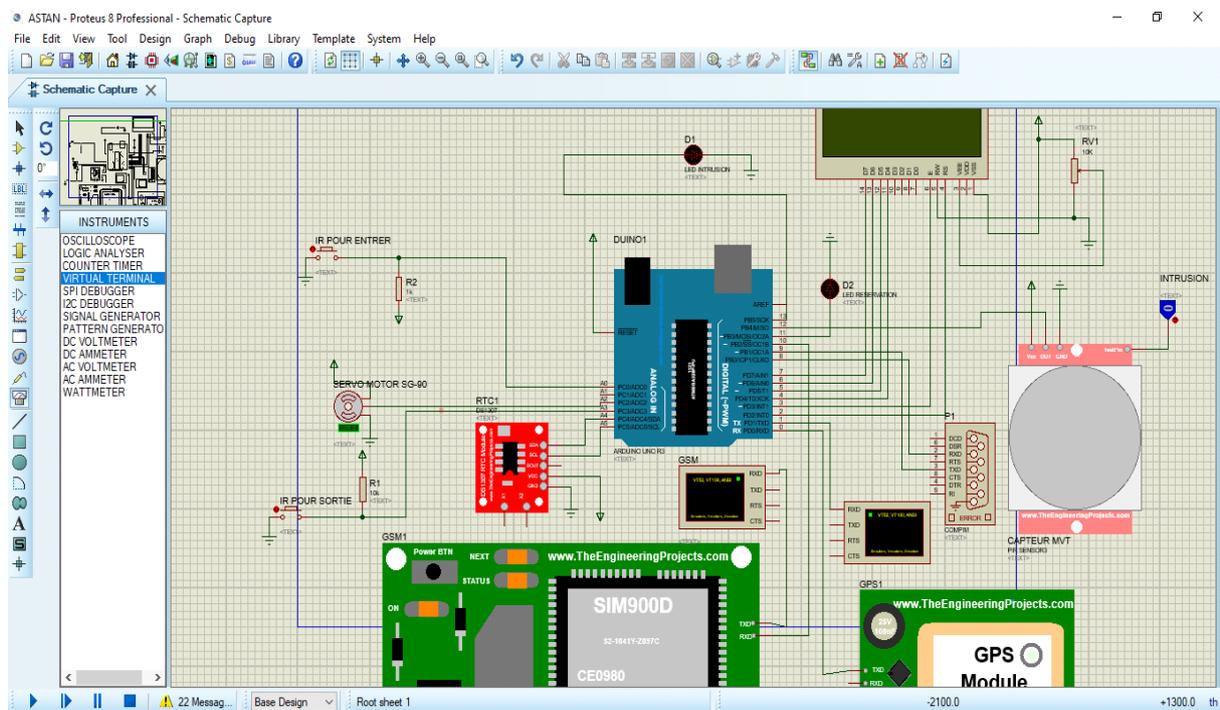


Figure 29: Schéma électronique complet entier de notre prototype smart Parking

Pour le GSM et l'Arduino on a ajouté des moniteurs série en procédant comme suite :

- Clic droit sur l'interface de simulation
- Aller sur Place
- Ensuite sur Virtual instrument
- Et sélectionner Virtual terminal

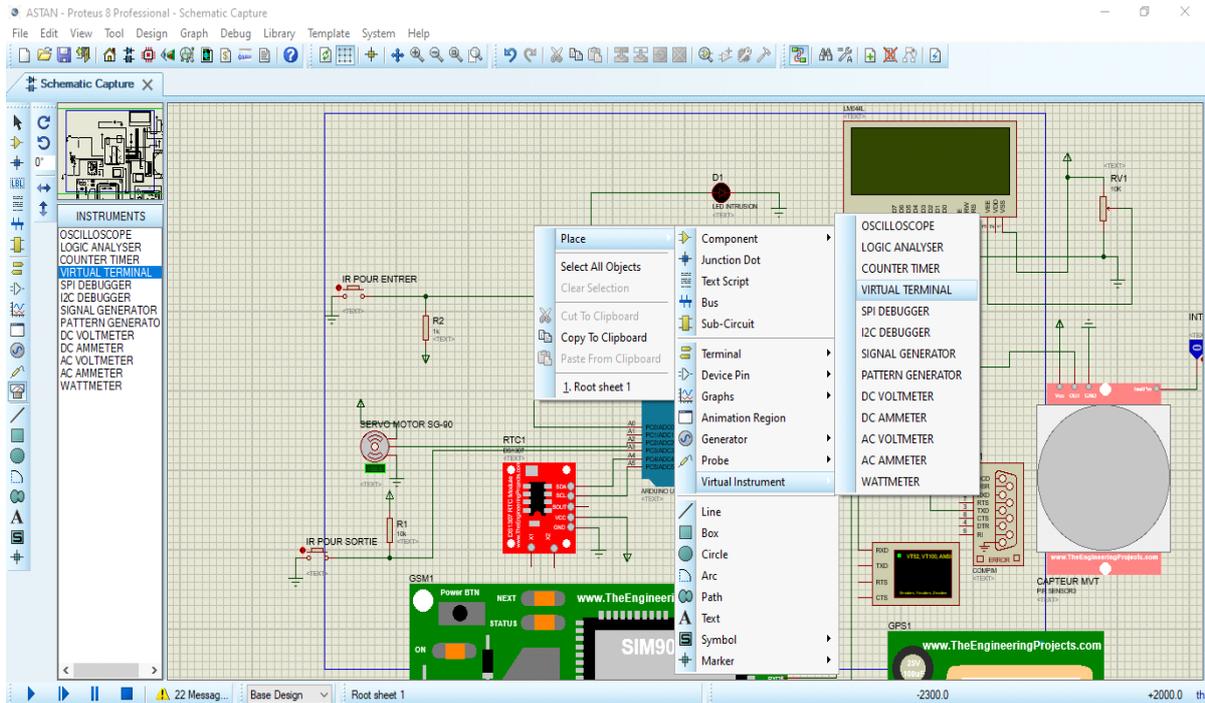


Figure 30: SMART PARKING (insertions des moniteurs séries)

De la même façon on ajoute le GROUND (terre) et POWER (la source d'alimentation des équipements)

(Clic droit >>>> Place >>>> Terminal >>>>> GROUND /POWER)

Après avoir fait le montage (câblage) on a installé la librairie dans le module GSM, le capteur de mouvement et le module GPS comme suite :

Double clic sur le module GSM on clique sur l'espace intitulé PROGRAM FILE et on cherche l'emplacement de la librairie préalablement télécharger .Delà même façon on procède pour installer les librairies pour le capteur de mouvement ainsi que le module GPS.

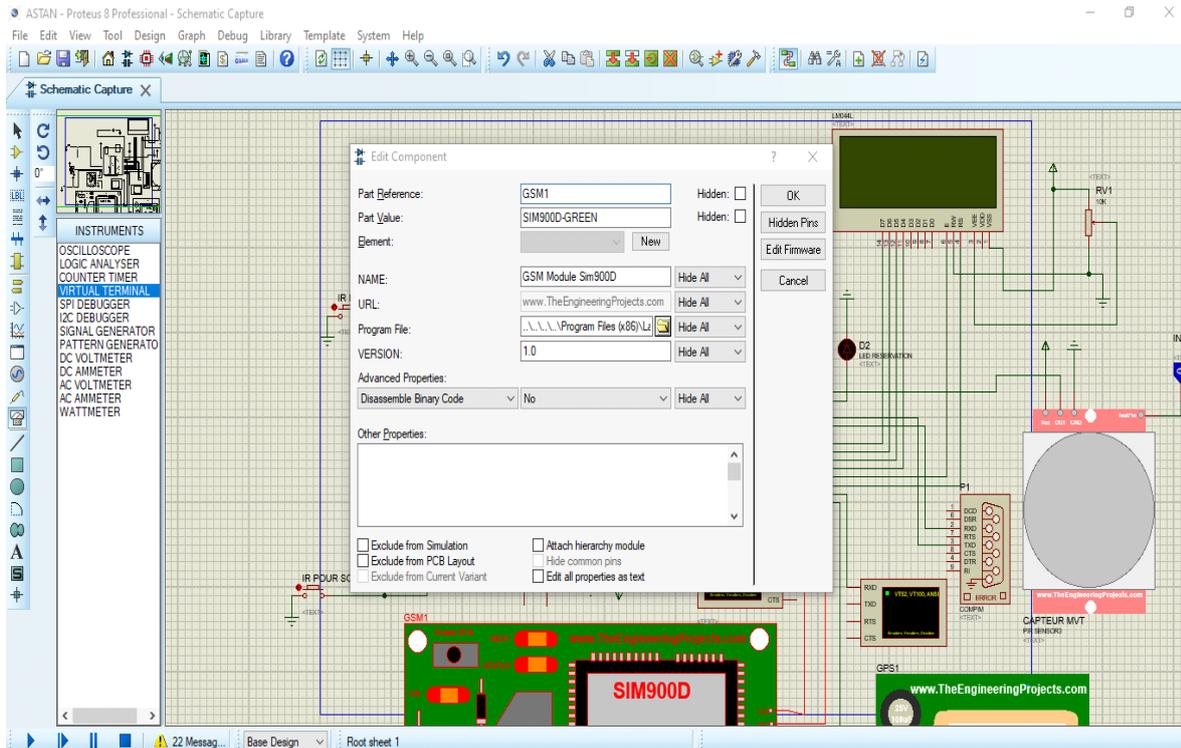


Figure 31: Prototype : configuration et insertion de la bibliothèque dans le module GSM

3. Le principe de fonctionnement du prototype

Le module RTC (Real Time Cloc) DS1307 envoie l'information contenant la date et l'heure sous le format suivant, par le pin A4 et A5 de l'Arduino :

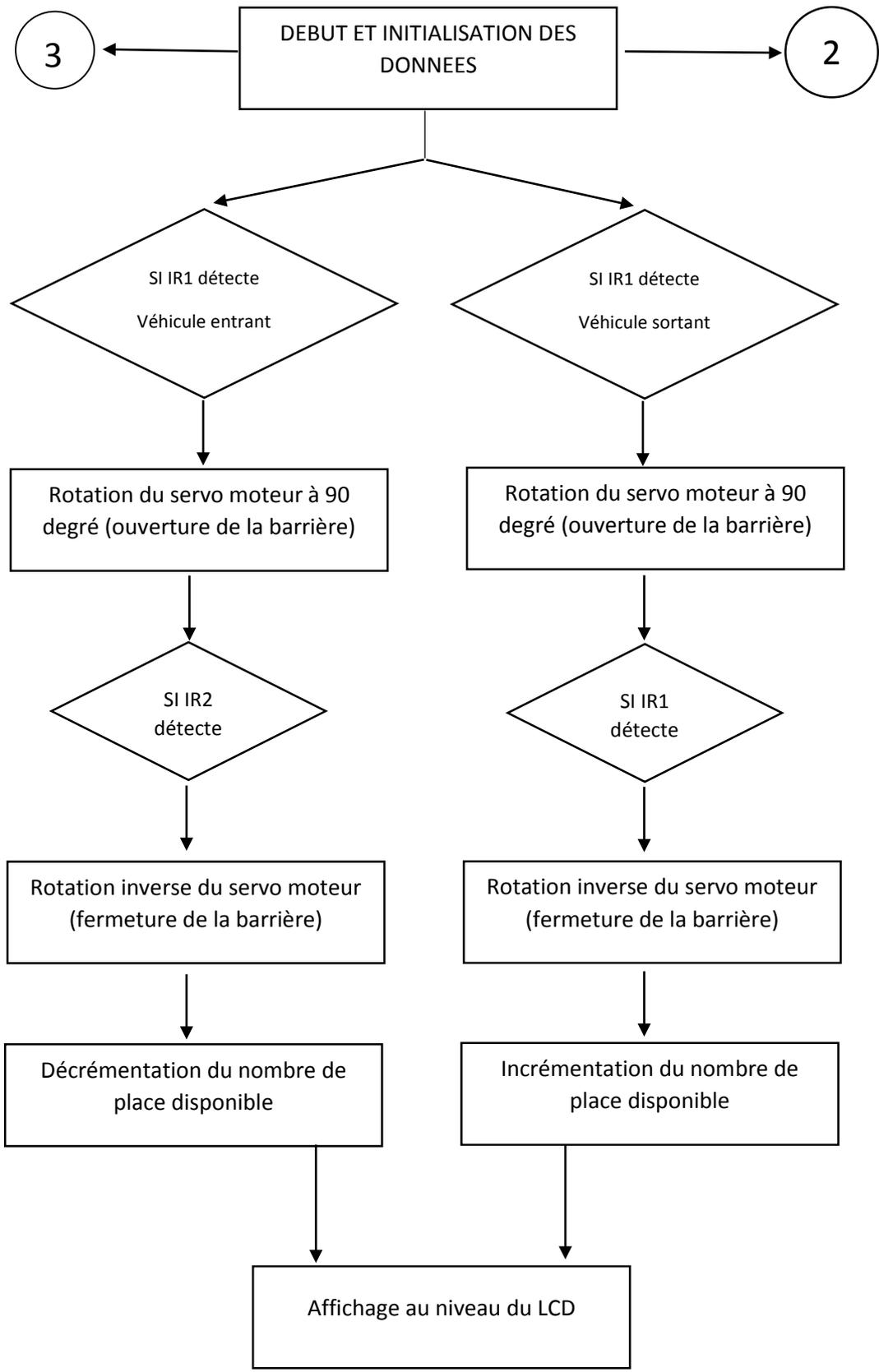
Time :	00 :00 :00
Date :	00/00/2000

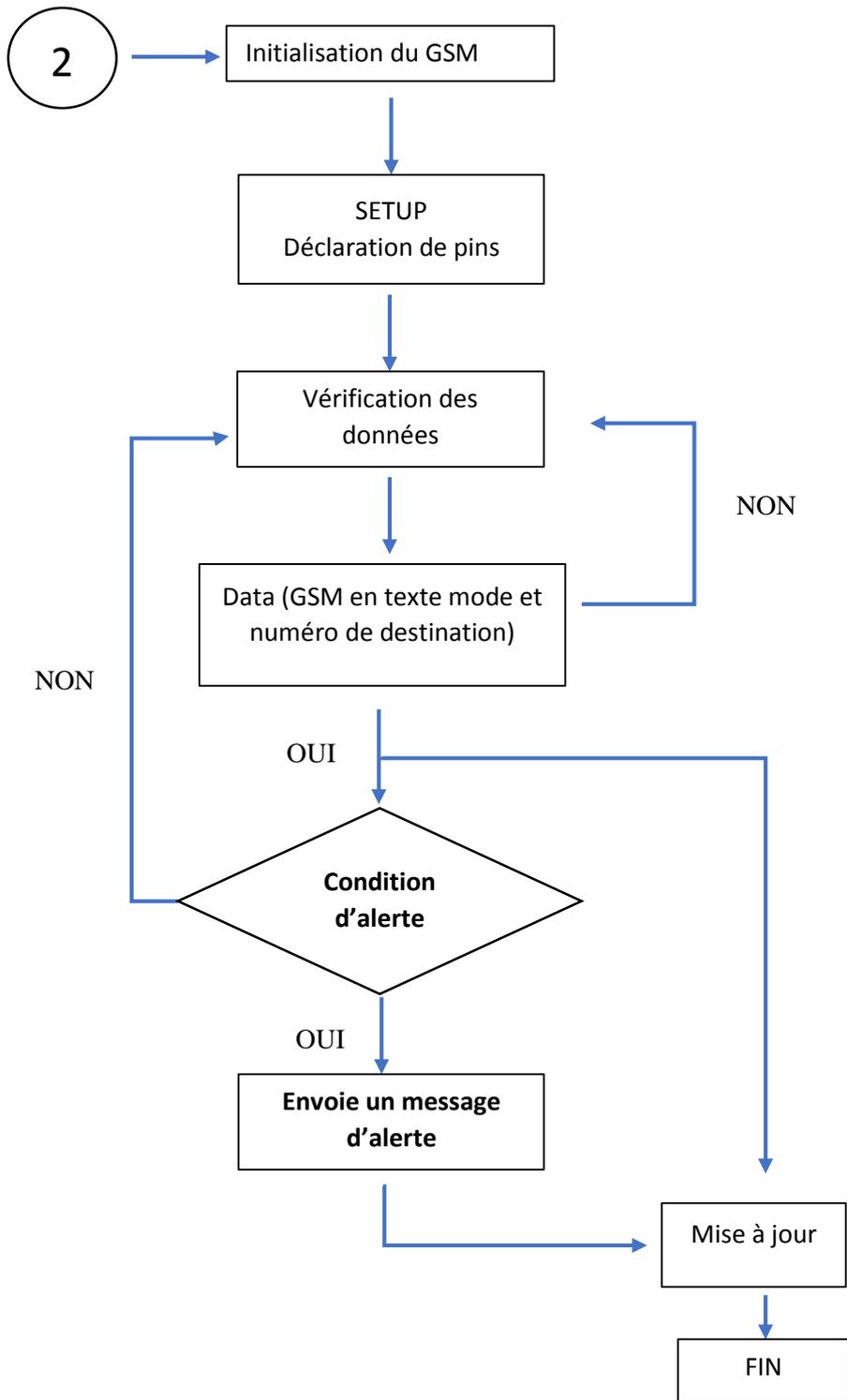
Le module RTC ne nous permet pas seulement d'avoir la date et l'heure car il nous permet aussi de conditionner la mise en marche de notre capteur de mouvement afin de pouvoir envoyer une ALERTE en cas d'intrusion indésirable dans le parking. Les capteurs infra-rouge permettent d'ouvrir la barrière pour laisser entrer ou sortir des véhicules et aussi d'incrémenter en cas d'entrée ou de décrémenter en cas de sortie le nombre de place disponible afin d'afficher le résultat sur le module LCD. Une fois que le résultat est affiché sur le LCD l'ARDUINO fait un test à travers le port virtuel COMPIIM pour savoir si notre application Blynk est connectée pour pouvoir envoyer des données sur cette dernière et/ou recevoir des commandes pour la réservation au niveau du parking d'une place. Si une réservation a été effectué alors l'arduino décrémente le nombre de place libre et attend un délai que dans notre simulation a été fixé de 10 secondes ce délais correspond à la durée d'attente d'un véhicule depuis le moment exacte de la réservation une fois ce délais dépassé la réservation sera annulé et un message sera envoyé au niveau de l'application pour informer que le délai d'attente est atteint et que la réservation est par conséquent annulé. Une fois que la réservation est annulée l'ARDUINO incrémente le nombre de place disponible.

Quant à notre détecteur de mouvement (PIR) il est muni d'un équipement nommé LOGICTOGGLE, cet équipement lui permet donc de simuler de mouvements que notre PIR détectera. Une fois qu'un mouvement est détecté l'ARDUINO vérifie l'heure exacte de la détection de mouvement et fait une concordance avec l'heure qu'on a fixé pour que notre GSM nous envoie un message d'alerte sur le numéro du propriétaire du PARKING disant que y'a une intrusion et une LED clignote. Une fois que le mouvement pendant cette tranche d'heure est arrêté le GSM envoie un autre message nous faisant savoir que le mouvement s'est arrêté et par conséquent y'a plus d'intrusion dans le parking. Ce dispositif permet au propriétaire du Parking d'être informé en temps réel sur la présence au sein du parking afin de mettre en place une meilleure sécurité et enfin de compte offrir une fiabilité sécuritaire à ses clients.

On a aussi doté le parking d'un GPS permettant de donner la localisation du parking afin de permettre aux clients de savoir l'emplacement du PARKING et en l'occurrence le chemin pour s'y rendre plus facilement faisant ainsi gagner du temps et ne pas faire des réservations fortuite.

4. Organigrammes de fonctionnement du prototype





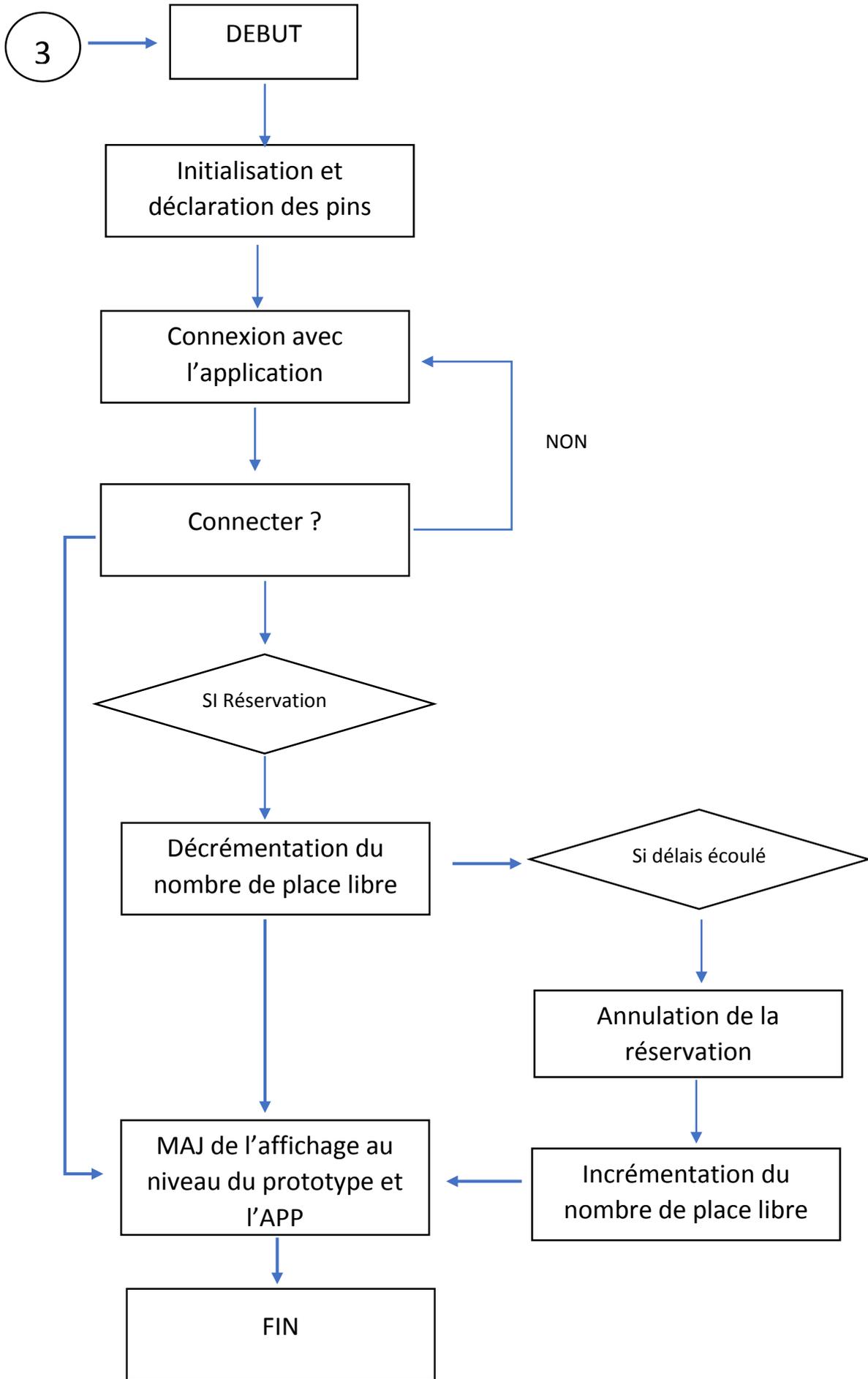


Figure 32: Organigramme de fonctionnement

5. Résultats

La simulation du prototype de notre Parking intelligent est donnée dans la figure 38. Une fois que la simulation est lancée on observe au niveau du LCD l'affichage qui nous indique le nombre de place total (TOTAL SPACE) qui ici est égale à 5 places disponible, le nombre de place disponible (HAVE SPACE) qui est également 5 places ce qui nous montre que y'a encore aucune voiture à l'intérieur du Parking et la place réservée (PLACE RESERVED) qui est égale à 0 car on a pas encore réservé.

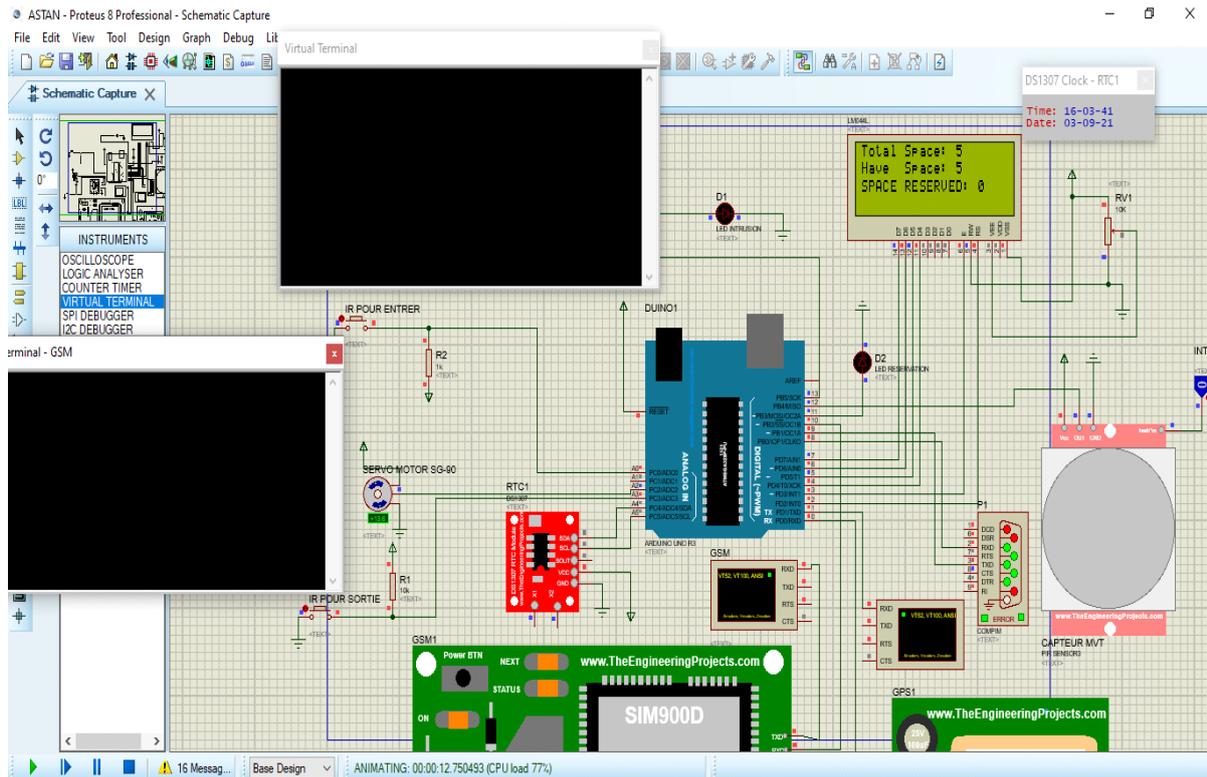


Figure 33: Résultat de la simulation sous PROTEUS

On a donc simulé un premier scénario c'est à dire on a fait entrer deux véhicules et voici le résultat. On a donc au niveau de l'affichage LCD (TOTAL SPACE =5 et HAVE SPACE = 3)

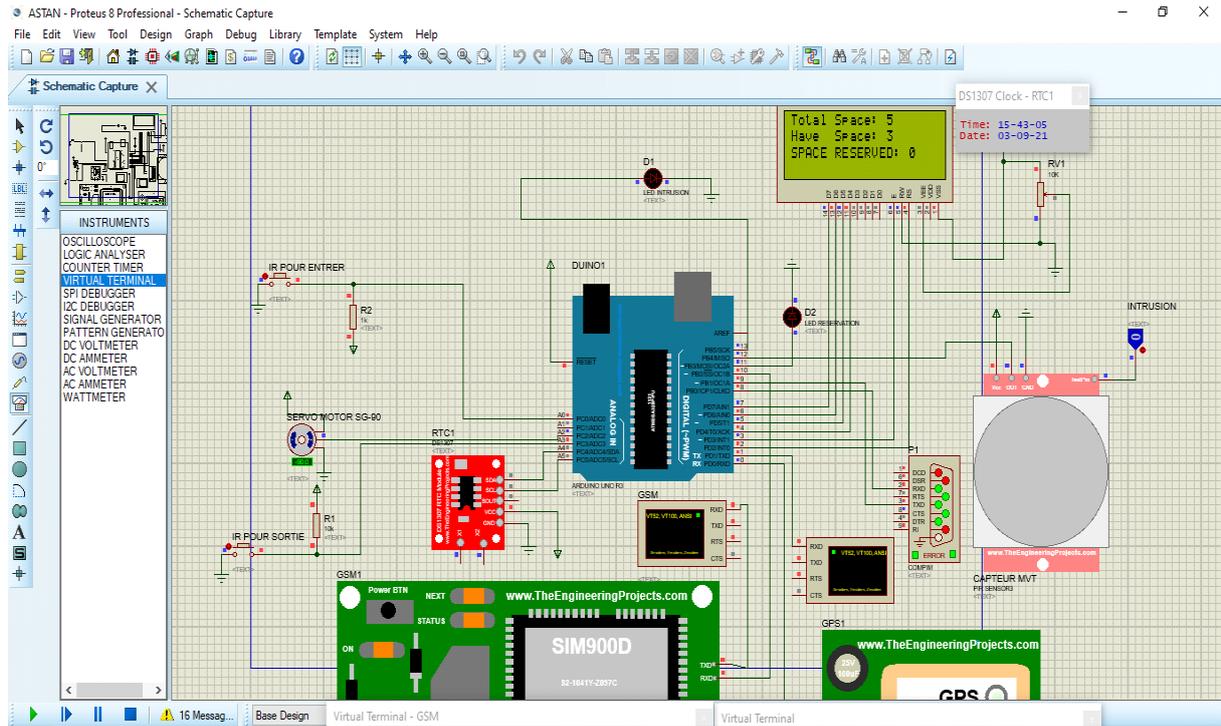


Figure 34: résultat du premier scenario deux véhicules entrés

Après que le détecteur infrarouge détecte un véhicule à l'entrée le servo moteur fait une rotation de pratiquement 90 degré ce que correspond en réalité à l'ouverture de la barrière. Cette barrière reste ouverte jusqu'à ce que le détecteur infra rouge qui est de l'autre côté de la barrière détecte cette même voiture ce qui déclenchera une rotation inverse du servo moteur qui le ramène a son état initial et correspondant donc à la fermeture de la barrière. Le même scenario se produit inversement quand un véhicule devrait sortir du parking. Ces scenarios se produisent à la seule condition que le nombre de place libre est supérieur à 0, si y'a plus de place libre le servo moteur ne fait pas de rotation c'est à dire la barrière reste fermé et l'afficheur LCD affiche "NO space available" autrement dit il n'y a plus de place libre.

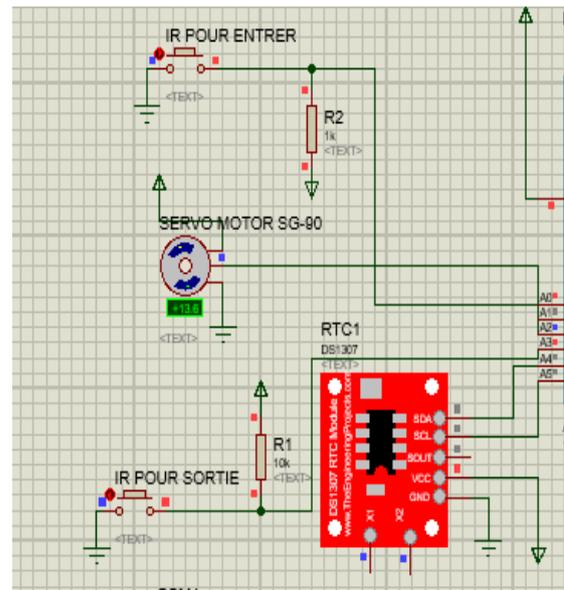
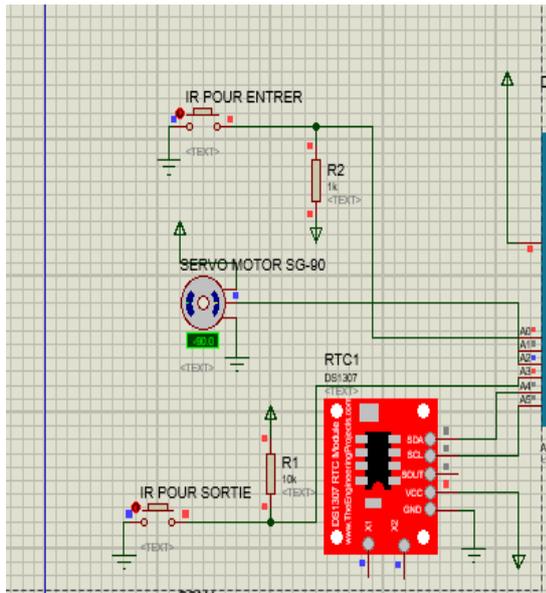


Figure 35: les rotations du servomoteur quand une voiture est détecté

Aussitôt après le lancement de la simulation le GSM vérifie la transmission en affichant sur son moniteur série qu'il est réglé en texte mode c'est à dire en mode envoie des messages textes. Ensuite il indique le numéro dans lequel il envoie les données écrit c'est à dire dans notre cas il envoie:

- ASTAN PARKING
- DATE
- HEURE

Comme dans la figure ci-dessous

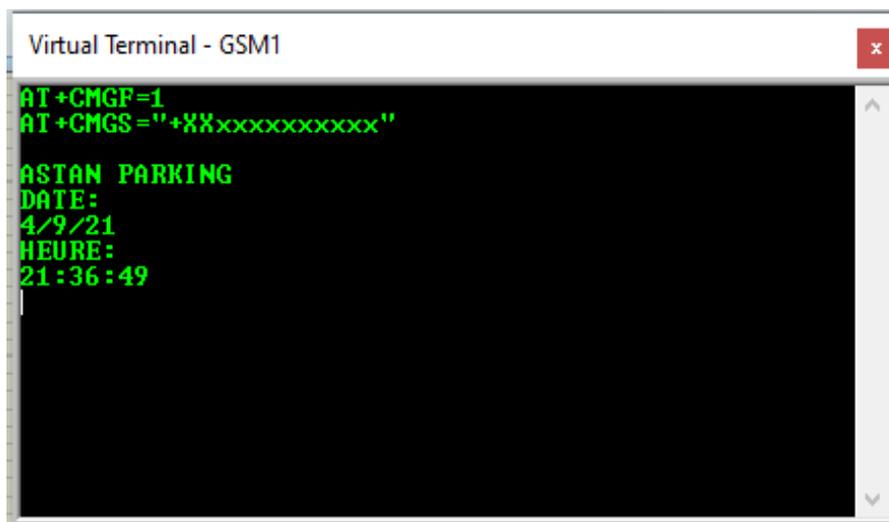


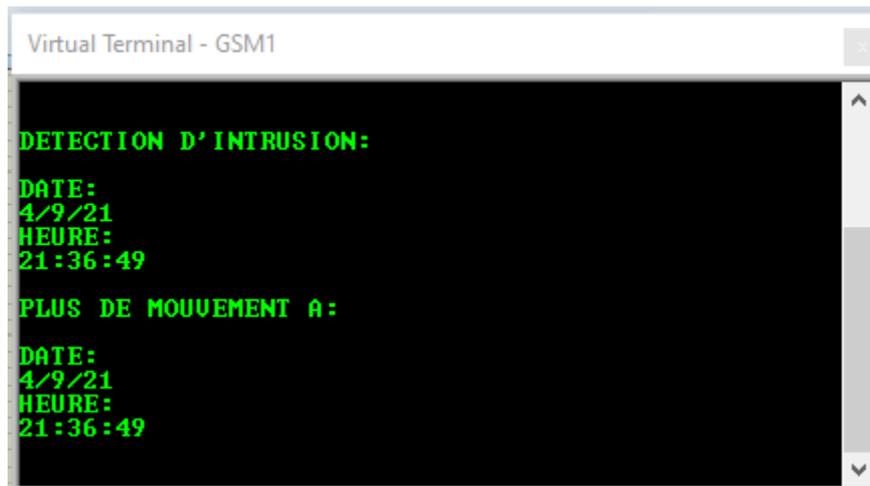
Figure 36: Résultat de l'envoi du GSM au début de la simulation

Après avoir vérifié la connexion et envoyer le message ci-dessus l'ARDUINO vérifie la condition d'alerte en cas d'intrusion dans le parking c'est à dire vérifier si le capteur de mouvement détecte une présence et en plus vérifier si cette présence est détecté après l'heure fixer pour la fermeture du parking, dans notre cas on a fixé cette heure à :

18h jusqu'à 6h.

On a donc simulé un mouvement en appuyant sur le bouton **logictoggle** pendant un moment et on a ensuite arrêté d'appuyer.

Le résultat est montré dans la figure ci-dessous

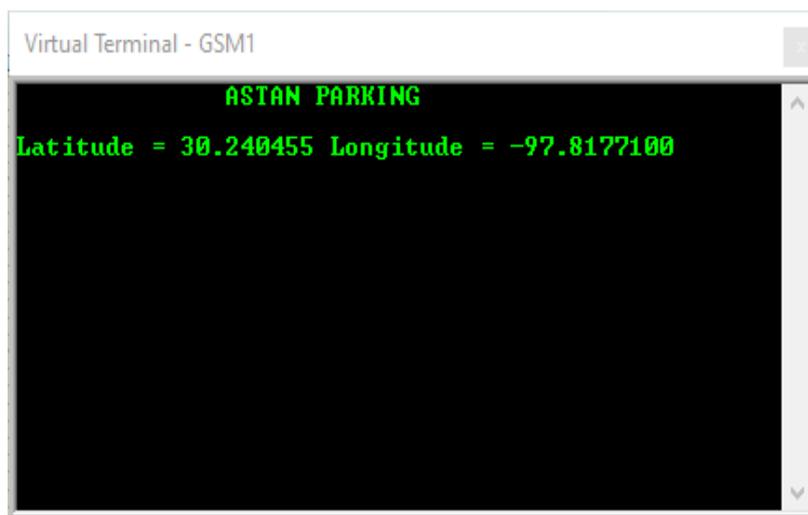


```
Virtual Terminal - GSM1
DETECTION D' INTRUSION:
DATE:
4/9/21
HEURE:
21:36:49
PLUS DE MOUVEMENT A:
DATE:
4/9/21
HEURE:
21:36:49
```

Figure 37: Résultat de l'envoi du GSM en cas d'intrusion après l'heure fixé

On a aussi simulé un GPS afin de nous donner les coordonnées afin de connaître l'emplacement du Parking.

Les résultats de la simulation



```
Virtual Terminal - GSM1
ASTAN PARKING
Latitude = 30.240455 Longitude = -97.8177100
```

Figure 38: Résultat de l'envoi du GSM des coordonnées GSP du parking

Une fois que notre prototype est simulé on a essayé de le connecter à notre application Blynk nommé SmartAstan comme cité plus haut notre application a pour de recevoir des données

nécessaires pour informer le client mais aussi de mener des actions au niveau du parking comme le fait de pouvoir réserver en appuyant sur le bouton ON de notre application Blynk .

Dans cette partie on remarque au niveau de notre afficheur LCD que RESERVED est égal à 1 ce qui est le résultat du fait que le client à réserver à partir de l'application une place dans le parking.

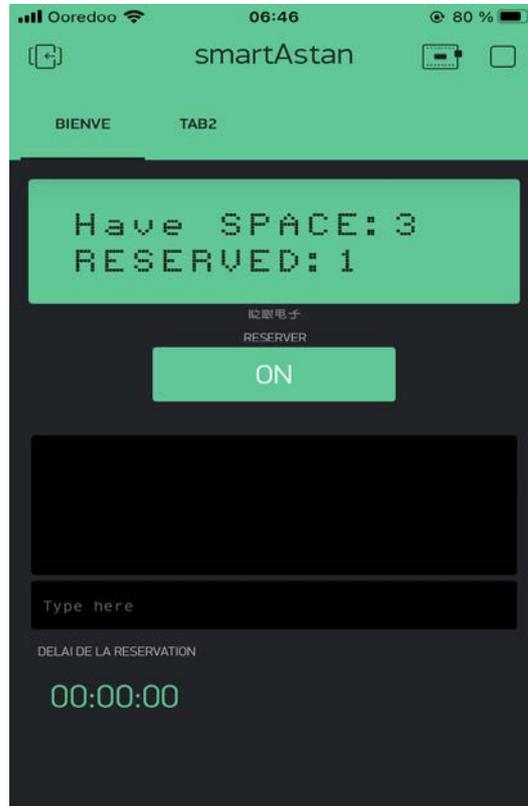


Figure 39: réservation à partir de l'application

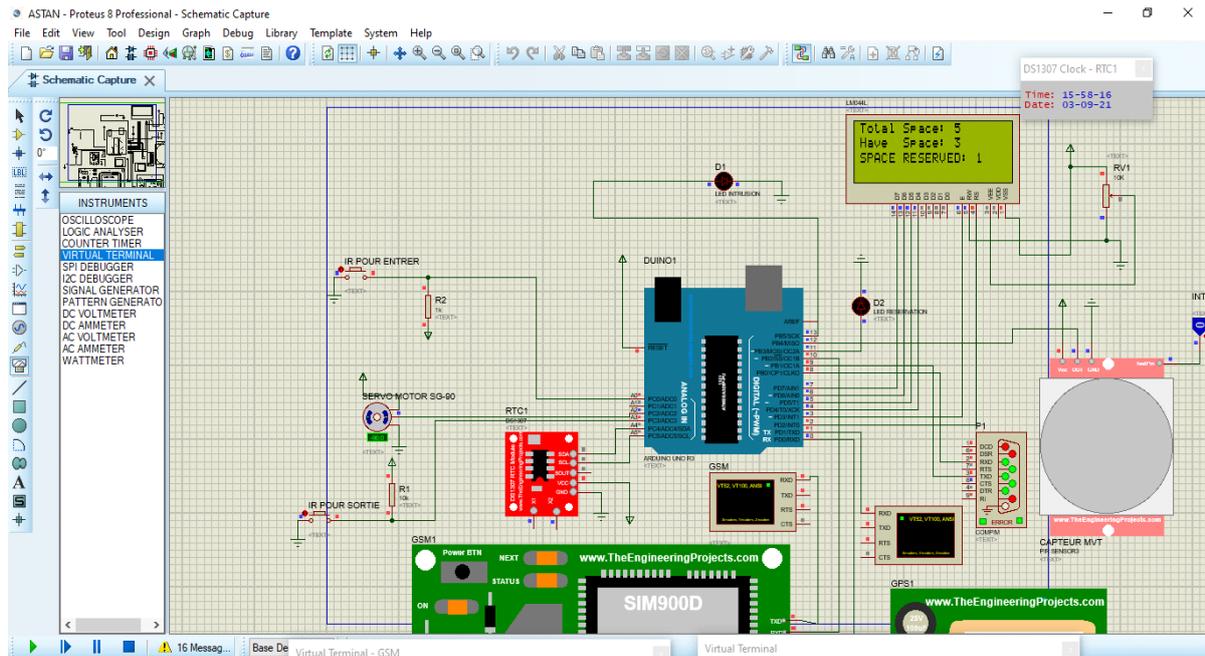


Figure 40: Résultat après la réservation sur l'application

Cette réservation à une durée limitée dans la simulation on a estimé cette durée à 10 secondes en utilisant un timer une fois ce délai dépassé. Mais si le client arrive au niveau du parking avant que le délai ne soit atteint alors la réservation sera annulée manuellement en appuyant sur le bouton OFF.

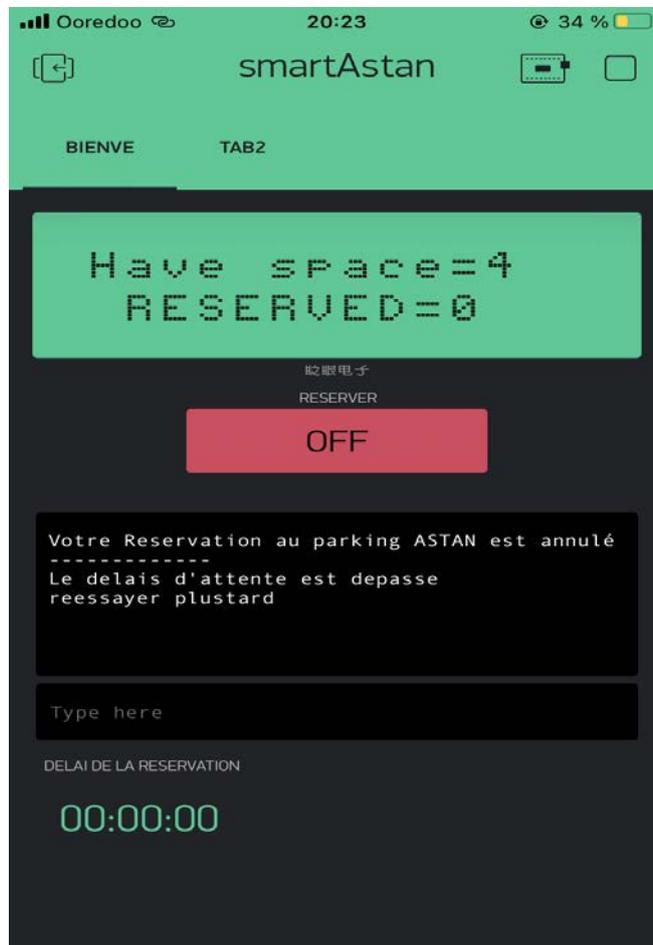


Figure 41: Annulation de la réservation

6. Conclusion

Dans ce chapitre on a vu la relation entre les différents équipements ainsi leur connexion avec Arduino sous PROTEUS. On a procédé à quelques configurations sur des équipements comme le module de transmission GSM le détecteur de présence PIR et le port virtuel pour pouvoir communiquer avec notre application Blynk. On a donc établi la communication entre Arduino et les modules de communications GSM et notre application. On a expliqué le principe de fonctionnement entier du prototype et on a procédé à la simulation. On a procédé à plusieurs tests pour témoigner de la fiabilité du processus ce qui nous a permis d'avoir des résultats plutôt convaincants.

Conclusion générale et perspectives

Dans ce projet l'objectif était de réaliser un parking intelligent dans le but d'informer le client sur la disponibilité d'une place libre de stationnement et de faire la réservation via une application mobile, auquel nous pouvons y accéder sur internet en temps réel. Amener le client directement sur une place libre offre plusieurs avantages tel que : réduire le temps, économiser le carburant et diminuer les émissions des polluants ; Et aussi d'informer le propriétaire sur l'état du parking offrant ainsi une sécurité maximale.

La réalisation de ce projet innovant a été un **vrai challenge**, car elle m'a permis de faire une recherche bibliographique sur un sujet nouveau et d'actualités, très applicatives et qui répond à un réel besoin du marché. Ce stage ma pousser à chercher des connaissances non seulement dans le domaine de l'électronique et des communications sans fil (GSM, WI-FI) mais aussi dans le domaine de l'informatique et des nouvelles technologies (par exemple les commandes de communication AT). Il m'a permis aussi d'acquérir de nouvelles compétences et d'utiliser de nouveaux équipements et bien d'autres. Pendant ce stage, j'ai également été confronté aux soucis que posent la conception et la simulation numérique d'un système électronique assez complexe.

Le prototypage virtuel par la simulation PROTEUS est un **moyen très puissant** il permet, grâce à sa bibliothèque riche, de créer un système model avec toutes les fonctionnalités désirées et faire dénombrables expérimentations tout en économisant le temps et l'énergie. Il nous a permis de faire une **conception complète du prototype** et d'avoir des **résultats très satisfaisants conformes à notre cahier des charges**.

Dans le but d'arriver à un démonstrateur final près à être commercialisé sur le marché je propose comme perspectives – pour améliorer et compléter ce projet – de ce travail les points suivants :

- La réalisation expérimentale du prototype électronique en version Béta et son intégration dans le marché pour la validation du prototype,
- L'utilisation d'une carte électronique plus performante,
- L'utilisation d'une mémoire externe pour stocker les données afin de créer une base de données destinée au recueil des Big Data afin d'optimiser les décisions du parking intelligent
- La création d'un réseau de parking intelligent pour couvrir un vaste territoire.

Références

bibliographiques

- [1] <https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/>
- [2] <https://blog.bepark.eu/fr/quest-ce-que-la-mobilit%C3%A9-intelligente>
- [3] <https://corporate.bepark.eu/fr/politique-de-stationnement-tout-ce-que-vous-devez-savoir#>
- [4] <http://dspace.univ-km.dz/xmlui/bitstream/handle/>
- [5] Carte ARDUINO UNO Microcontrôleur ATmega328 (<http://perso-laris.univ-angers.fr/~cotenceau/ArduinoCotenceau2016.pdf>)
- [6] MODULE GSM/GPRS SEEDSTUDIO Lycée Gustave Eiffel
- [7] 026-2007.pdf
- [8] <https://pecquery.wixsite.com/arduino-passion/le-servomoteur>
- [9] <http://qqtrading.com.my/ir-infrared-obstacle-detection-sensor-module-fc-5>
- [10] file:///C:/Users/pc/AppData/Local/Temp/ZerroukiMohamedAmine_NernasRiadh.pdf
- [11] <https://www.carnetdumaker.net/articles/utiliser-un-module-horloge-temps-reel-ds1307-avec-une-carte-arduino-genuino/>
- [12] Commande AT (https://www.technologuepro.com/gsm/commande_at.htm)
- [13] PROTEUS (<http://www.elektronique.fr/logiciels/PROTEUS.php>)
- [14] ISIS PROTEUS (http://projet.eu.org/pedago/sin/tutos/ISIS_PROTEUS.pdf)
- [15] **L'application mobile blynk** (<https://booteille.github.io/blynk-docs-fr/>)

Annexes

ANNEXE 1

LED GSM

DEL « PWR » (verte)	Eteinte	Le shield est hors tension
	Allumée	Le shield est sous tension
DEL « ETAT » (Rouge)	Eteinte	Le module SIM900 est hors tension
	Allumée	Le module SIM900 est sous tension
DEL « RESEAU » (Verte)	Eteinte	Le module SIM900 en veille
	Allumée : 64 ms Eteinte : 800 ms	Le module ne trouve pas de réseau
	Allumée : 64 ms Eteinte : 3000 ms	Le module a trouvé un réseau
	Allumée : 64 ms Eteinte : 300 ms	Communication GPRS

ANNEXE 2

Commandes AT dédiées au service SMS :

AT+CSMS	Sélection du service de messagerie
AT+CPMS	Sélection de ta zone mémoire pour le stockage des SMS
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CSAS	Sauvegarde du paramétrage
AT+CREG	Restauration du paramétrage par défaut
AT+CNMI	Indication concernant un nouveau SMS
AT+CMGL	Liste les SMS stockés en mémoire
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMSS	Envoie d'un SMS stocké en mémoire
AT+CMGW	Écriture d'un SMS
AT+CMGD	Efface un SMS

ANNEXE 3

CODE ARDUINO POUR LE FONCTIONNEMENT DU PROTOTYPE :

```
#define BLYNK_TEMPLATE_ID      "TMPLxxxxxx"
#define BLYNK_DEVICE_NAME     "Device"
#define BLYNK_AUTH_TOKEN      "a6ohbNg9ZlpZpZwGYsyye-wIYWw2yhR"
```

```
#include <Wire.h>
#include <LiquidCrystal.h> // initialize the library with the numbers of the interface pins
#include <SoftwareSerial.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
#include <Servo.h> //includes the servo library
#include <BlynkSimpleStream.h>
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(10, 11); // RX, TX
SoftwareSerial mySerial(8, 9);
#define BLYNK_PRINT SwSerial
char auth[] = BLYNK_AUTH_TOKEN;
WidgetLCD Lcd(V2);
uint8_t second, minute, hour, wday, day, month, year, ctrl;
int ledPin = 13; // choose the pin for the LED
int inputPin = 12; // choose the input pin (for PIR sensor)
int pirState = LOW; // we start, assuming no motion detected
int val = 0; // variable for reading the pin status
const int buttonPin = V1; // button1 is connected to pin 11
int buttonState = 0;

Servo myservo1;
int ir_s1 = A0;
int ir_s2 = A3;
int Total = 5;
int Space;

int RESERVED;

int flag1 = 0;
int flag2 = 0;
```

```
void setup() {
  pinMode(buttonPin, INPUT);

  mySerial.begin(9600); // Setting the baud rate of GSM Module
  Serial.begin(9600); // Setting the baud rate of Serial Monitor (Arduino)
  delay(100);

  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare sensor as input

  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);

  myservo1.attach(A2);
  myservo1.write(100);

  lcd.begin(20, 4);
  lcd.setCursor(0,0);
  lcd.print(" Car Parking ");
  lcd.setCursor(0,1);
  lcd.print(" System ");
  delay(1000);
  lcd.clear();

  Space = Total;
  RESERVED=0;

  // Debug console
  SwSerial.begin(115200);

  // Blynk will work through Serial
  // Do not read or write this serial manually in your sketch
```

```

Serial.begin(9600);
Blynk.begin(Serial, auth);

delay(400);
if (read_ds1307()) {
  Serial.println("AT+CMGF=1");          //Sets the GSM Module in Text Mode
                                     // Delay of 1000 milli seconds or 1 second
  Serial.println("AT+CMGS=\"+XXXXXXXXXXXX\"\\r"); // Replace x with mobile number
  Serial.println("ASTAN PARKING");
  print_time() ;
} else {
  Serial.println("Non detector DS1307");
}
delay(500);
}

void loop(){

if(analogRead (ir_s1) == LOW && flag1==0){
if(Space>0){flag1=1;
if(flag2==0){myservo1.write(0); Space = Space-1-RESERVED;}
}else{
  lcd.setCursor (0,0);
  lcd.print(" Sorry not Space ");
  lcd.setCursor (0,1);
  lcd.print(" Available ");
  delay (1000);
  lcd.clear();
}
}
}

```

```
if(analogRead (ir_s2) == LOW && flag2==0){flag2=1;
if(flag1==0){myservo1.write(0); Space = Space+1;}
}
```

```
if(flag1==1 && flag2==1){
delay (1000);
myservo1.write(100);
flag1=0, flag2=0;
}
```

```
lcd.setCursor (0,0);
lcd.print("Total Space: ");
lcd.print(Total);
```

```
lcd.setCursor (0,1);
lcd.print("Have Space: ");
lcd.print(Space);
lcd.setCursor (0,2);
lcd.print("SPACE RESERVED: ");
lcd.print(RESERVED);
```

```
// delay(3000);
// if (read_ds1307()) {
//print_time() ;
// } else {
// Serial.println("Non detector DS1307");
// }
// delay(1000);
```

```

val = digitalRead(inputPin); // read input value
if ((val == HIGH)&&(23<=hour<=6)) { // check if the input is HIGH
  digitalWrite(ledPin, HIGH); // turn LED ON
  if (pirState == LOW) {
    // we have just turned on
    Serial.println();
    Serial.println();
    Serial.println("DETECTION D'INTRUSION:");
    Serial.println();
    print_time();
    // We only want to print on the output change, not state
    pirState = HIGH;
  }
} else {
  digitalWrite(ledPin, LOW); // turn LED OFF
  if (pirState == HIGH){
    // we have just turned of
    Serial.println();
    Serial.println("PLUS DE MOUVEMENT A:");

    Serial.println();
    print_time();
    // We only want to print on the output change, not state
    pirState = LOW;
  }
}
// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed. If it is, the buttonState is HIGH:

```

```

if (buttonState == HIGH) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
  RESERVED=RESERVED+1;
  Space=Space-1;
  lcd.setCursor (0,1);
  Lcd.print(1, 0, "Have Space=");
  Lcd.print(11, 0,Space);
  Lcd.print(1, 1, "RESERVED: ");
  Lcd.print(9, 1,RESERVED);

}
else {
  // turn LED off:
  digitalWrite(ledPin, LOW);
  RESERVED=RESERVED-1;
  Space=Space+1;
  lcd.setCursor (0,1);
  Lcd.print(1, 0, "Have Space=");
  Lcd.print(11, 0,Space);
  Lcd.print(1, 1, "RESERVED: ");
  Lcd.print(9, 1,RESERVED);

  SwSerial.println("Votre reservation est annuler" );
  SwSerial.println("Car le delais d'attente est depasser" );
  SwSerial.println("Veuillez reessayer plutard" );
}

if ( (Serial.available(>0))
switch(Serial.read())
{case 's':
  SendMessage();

```

```

        break;
    }
}

bool read_ds1307()
{
    // Iniciar el intercambio de información con el DS1307 (0xD0)
    Wire.beginTransmission(0x68);
    // Escribir la dirección del segundero
    Wire.write(0x00);
    // Terminamos la escritura y verificamos si el DS1307 respondió
    // Si la escritura se llevo a cabo el metodo endTransmission retorna 0
    if (Wire.endTransmission() != 0)
        return false;
    // Si el DS1307 esta presente, comenzar la lectura de 8 bytes
    Wire.requestFrom(0x68, 8);

    // Recibimos el byte del registro 0x00 y lo convertimos a binario
    second = bcd2bin(Wire.read());
    minute = bcd2bin(Wire.read()); // Continuamos recibiendo cada uno de los registros
    hour = bcd2bin(Wire.read());
    wday = bcd2bin(Wire.read());
    day = bcd2bin(Wire.read());
    month = bcd2bin(Wire.read());
    year = bcd2bin(Wire.read());

    // Recibir los datos del registro de control en la dirección 0x07
    ctrl = Wire.read();

    // Operacion satisfactoria, retornamos verdadero
    return true;
}

```

```

uint8_t bcd2bin(uint8_t bcd)
{
    // Convertir decenas y luego unidades a un numero binario
    return (bcd / 16 * 10) + (bcd % 16);
}

void print_time()
{
    Serial.print("DATE: ");
    Serial.println();
    Serial.print(day);
    Serial.print('/');
    Serial.print(month);
    Serial.print('/');
    Serial.print(year);
    Serial.println();
    Serial.print("HEURE: ");
    Serial.println();
    Serial.print(hour);
    Serial.print(':');
    Serial.print(minute);
    Serial.print(':');
    Serial.print(second);
    Serial.println();
}

void SendMessage()
{
    mySerial.println("AT+CMGF=1");          //Sets the GSM Module in Text Mode
                                           // Delay of 1000 milli seconds or 1 second
    mySerial.println("AT+CMGS=\"+XXxxxxxxxxx\""); // Replace x with mobile number
}

```

```
if (val == HIGH)          //mySerial.available(>0)
{ Serial.write(mySerial.read());
  Serial.println("GSM CONNECTER");
  delay(500);
  if (val == HIGH)
  Serial.println("MOTION DETECTED"); // AT Command to receive a live SMS

}
}
```