

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingéniorat

Département : Electronique

Domaine : Sciences et Techniques

Filière : Electronique

Spécialité : Instrumentation

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Conception d'un système de gestion d'une cité intelligente

Présenté par : *Baia-Rasso Abir*

Encadrant : A.REDJATI

Maitre de conférences

UBM Annaba

Jury de Soutenance :

k. MANSOURI	Professeur	UBM Annaba	Président
A. REDJATI	Maitre de conférences	UBM Annaba	Encadrant
S. GHERBI	Professeur	UBM Annaba	Examineur

Année Universitaire : 2020/2021

ملخص

تتطور البشرية من حيث المكان والزمان والمادة بمعدل غير مسبوق فقد أدى هذا التطور الى ظهور مصطلح جديد يعرف بالمدينة الذكية اذ اصبحت نموذجًا حضريًا جديدًا.

ظهر هذا المفهوم عند مفترق طرق لاتجاهين قويين التحضر المتسارع للبشرية والثورة الرقمية تعمل تقنيات المعلومات والاتصالات على تجهيز المدن بهدف توفير أدوات للمساعدة في اتخاذ القرار وبالتالي تحسين الأداء الحضري.

بحيث يتم توصيل الخدمات العامة والخاصة من أجل تحسين IOT تدمج المدينة الذكية أيضًا بروتوكول إنترنت الأشياء جميع عمليات الوصول ذات الصلة والتحكم فيها وتسهيلها.

يجب أن تشجع المدينة الذكية مواطنيها وشركاتها وإداراتها على تخيل أنماط جديدة للتنظيم والمشاركة والاتصال والإنتاج... لتطوير الديناميكية الحضري.

يتعلق هذا المشروع بتصميم وإنجاز نموذج أولي لمدينة ذكية يعتمد على متحكم Arduino Mega كلوحة اتصالات بالإضافة إلى ESP 32 كعنصر ذكي. يتكون نظامنا من ثلاثة أجزاء أساسية موقف السيارات الذكي، والري التلقائي والإضاءة العامة الذكية التي تعمل بلوحة شمسية. بالإضافة إلى جهاز تحكم عن بعد باستخدام تطبيق Blynk لمراقبة الري ومواقف السيارات.

كلمات مفتاحية :

الإضاءة العامة . المدينة الذكية، موقف السيارات الذكي، الري التلقائي، Blynk ، ArduinoMega ، ESP 32 ،
IoT الذكية،

Abstract

Humanity is evolving in terms of space, time and matter at an unprecedented rate, so, for a few years now, the Smart City has been a new urban ideal.

This concept appeared at the crossroads of two strong trends, the exponential urbanization of humanity and the digital revolution, information and communications technologies are equipping cities with the objective of providing tools to help decision thus improving urban functioning.

The smart city also integrates the Internet of Things (IOT) protocol so that public and private services will be connected in order to optimize, control and facilitate all related access.

The Smart City must encourage its citizens, its businesses, its administrations to imagine new modes of organization, sharing, communication, production ... to develop urban dynamism.

This project concerns the design and realization of a prototype of a smart city based on an Arduino Mega microcontroller as a communication and control board as well as ESP 32 as an intelligent element. Our system consists of three essential parts: intelligent parking, automatic irrigation and intelligent public lighting powered by a solar panel. As well as a remote control via the Blynk application for watering and parking monitoring.

Key words: Iot, smart city, ESP32, Arduino Mega, intelligent parking, automatic irrigation, intelligent public lighting, solar panel, Blynk.

Résumé

L'humanité évolue sur le plan des espaces, du temps et de la matière à un rythme jamais atteint, ainsi, de puis quelques années, la Smart City constitue un nouvel idéal urbain.

Ce concept est apparu à la croisée de deux tendances fortes, l'urbanisation exponentielle de l'humanité et la révolution numérique, les technologies de l'information et des communications équipent les villes dans l'objectif de fournir des outils d'aide à la décision améliorant ainsi le fonctionnement urbain.

La ville intelligente intègre aussi le protocole de l'internet des objets (l'Iot) pour que les services publics et privés seront connectés dans le but d'optimiser, contrôler et faciliter tous les accès y liée.

La Smart City doit encourager ses citoyens, ses entreprises, ses administrations à imaginer de nouveaux modes d'organisation, de partage, de communication, de production... pour développer le dynamisme urbain.

Ce projet concerne la conception et la réalisation d'un prototype d'une ville intelligente à base d'un microcontrôleur Arduino Mega comme une carte de communication et contrôle ainsi que l'ESP 32 comme un élément intelligent. Notre système constitué de trois parties essentielles : un parking intelligent, irrigation automatique et éclairage public intelligent alimenté par un panneau solaire. Ainsi qu'une commande à distance via l'application Blynk pour la surveillance de l'arrosage et du parking.

Mots clés : smart city, parking intelligent, irrigation automatique, éclairage public intelligent, Blynk, Arduino Mega, ESP32, Iot.

Remerciement

Je remercie en premier lieu ; Dieu le tout puissant, le seigneur de tous les temps, pour son aide et sa grande miséricorde, car c'est grâce à lui que je suis arrivée à accomplir et à présenter ce modeste travail.

*Je tiens à exprimer ma profonde gratitude à mon encadreur **Mr REDJATI ABDELGHANI**, pour avoir accepté de diriger ce travail. Je lui témoigne toutes mes reconnaissances pour ses conseils, ses orientations et sa patience.*

Grand merci à toute personne ayant aidé de près ou de loin à l'accomplissement de ce modeste travail.

Dédicaces

Je dédie ce modeste travail

En premier lieu à mes très chers parents

*Qui m'ont soutenu avec leurs prières tout au long de mon parcours
scolaire et universitaire*

Ainsi que mes chères sœurs Ferial et Malika,

Et toute la famille,

*« **BAIARASSO** » et « **NOUACER** »*

Sans oublier la famille universitaire,

En particulier Le département d'Electronique

*Le Chef de département, Les enseignants et aussi tous les
Responsables et le personnel pour tout le savoir qu'ils m'ont*

Transmis durant tous le cycle universitaire.

Enfin à tous mes amis qui m'ont soutenu et encouragé.

A mes très chères amies Char Rania, Bekouche Sabrina et Ibtissem.

Liste des tableaux

Tableau II.1 : Les Caractéristiques de la carte Arduino Mega 2560.....	15
Tableau II.2 : Les Caractéristiques du module ESP32.....	17
Tableau II.3 : les caractéristiques de capteur d'humidité du sol.....	22
Tableau II.4 : Caractéristiques de LDR.....	25
Tableau II.5 : Caractéristiques du module HC-SR04.....	27
Tableau II.6 : contrôle du sens de rotation d'un moteur à CC.....	29
Tableau II.7 : Caractéristiques de L293D.	30
Tableau II.8 : Brochage LCD I2C avec ESP32 et Arduino MEGA.....	31
Tableau II.9 : Caractéristiques Du servomoteur.....	33

Figure I.1 : Tableau programmatique d'une ville intelligente selon l'agence européenne Dédale consacrée à la culture, aux technologies et à l'innovation sociale.....	5
Figure I.2 : Six critères principaux de la Ville intelligente.....	9
Figure I.3 : Des exemples sur l'internet des objets.....	10
Figure I.4 : Smart parking.....	11
Figure I.5 : Eclairage intelligent.....	12
Figure I.6 : Iot based smart irrigation system.....	12
Figure II.1 : Architecture de la carte MEGA 2560.....	15
Figure II.2 : Carte ESP32.....	16
Figure II.3 : Schéma fonctionnel du l'ESP32.....	18
Figure II.4 : Brochage du module ESP32.....	18
Figure II.5 : Exemple programme Blynk avec ESP32.....	20
Figure II.6 : Capteur d'humidité	22
Figure II.7 : Capteur d'humidité capacitif.....	23
Figure II.8 : Photorésistance LDR.....	24
Figure II.9 : Capteur infrarouge.....	25
Figure II.10 : Capteur ultrason avec brochage.....	26
Figure II.11 : Séquence de fonctionnement du module HC-SR04.....	28
Figure II.12 : Brochage de L293D.....	29
Figure II.13 : Afficheur LCD I2C.....	31
Figure II.14 : Servomoteur SG90.....	32
Figure II.15 : Illustration du signal de contrôle Source.....	33
Figure II.16 : Mini pompe à eau.....	34
Figure II.17 : Panneau solaire.....	34
Figure III.1 : Schéma globale du système.....	38

Figure III.2 : schéma synoptique.....	39
Figure III.3: Organigramme de fonctionnement.....	40
Figure III.4 : Schéma électronique du parking intelligent.....	41
Figure III.5: Montage sur plaque d'essai (Breadboard).....	42
Figure III.6 : Schéma synoptique.....	44
Figure III.7: Organigramme de fonctionnement.....	45
Figure III.8 : Schéma électronique d'arrosage automatique.....	46
Figure III.9: Montage sur plaque d'essai (Breadboard).....	47
Figure III.10 : Chargeur de batterie automatique entre 14-15 volts avec un courant maximal de 3 ampères.....	49
Figure III.11 : Schéma synoptique.....	51
Figure III.12: Organigramme de fonctionnement.....	52
Figure III.13: Schéma électronique d'un suiveur solaire.....	53
Figure III.14: Chargeur de batterie 12V.....	54
Figure III.15: Convertisseur DC-DC.....	56
Figure III.16: Schéma électronique.....	57
Figure III.17: le mode station d'un ESP32.....	58
Figure III.18: le mode station d'un ESP32.....	59
Figure III.19: présentation de l'application Blynk.....	63
Figure III.20: présentation de l'interface de l'application Blynk.....	64
Figure III.21: Application Blynk : Parking avec l'ESP32.....	64
Figure III.22: Application Blynk : Système d'irrigation avec l'ESP32.....	66

Liste des abréviations

TIC : Technologie de l'Information et de Communication.

IoT: Internet of Things.

URL: Uniform Resource Locator.

SIG: Système d'Information Géographique.

WIFI: Wireless Fidelity.

IP: Internet Protocol.

URL: Uniform Resource Locator.

PWM: Power Width Modulation.

IdO : Internet des Objets.

MLI : Modulation de Largeur d'Impulsion.

IIoT : Internet Industrial of Things (Internet Industriel Des Objets).

CAD : Computer Aided Construction.

IDE : Integrated Developpement Environement.

LDR: Light Dependent Resistor.

LCD: Liquid Crystal Display.

I2C: Inter Integrated Circuit.

SDA: Serial Data Line.

SCL: Serial Clock Line.

IR: InfraRed.

PV: Panneau photovoltaïque.

UART: Universal Synchronous/Asynchronous Receiver Transmitter.

EEPROM: Electrically-Erasable Programmable Read-Only Memory.

SPI: Serial Peripheral Interface.

CPU: Central Processing Unit.

ADC: Analog to Digital Converter.

ROM: Read Only Memory.

RTC: Réseau Téléphonique Commuté.

SDK: Software Development Kit.

LED: Light-Emitting Diode.

GND: Ground.

TTL: Transistor-Transistor Logic.

DC: Direct Current.

CC: Courant Continu.

DIP: Dual Inline Package.

Liste des tableaux	
Liste des figures	
Liste des abréviations	
Table des matières	

Introduction générale

Chapitre I: Etat de l'art

I.1. Introduction	04
I.2. Définition de la ville intelligente	04
I.3. Évolution du concept de "Smart City "	06
I.4. Les objectifs de la smart city	07
I.5. Les caractéristiques de la smart city	07
a. Une économie intelligente	08
b. Des citoyens intelligents	08
c. Une gouvernance intelligente	08
d. Une mobilité intelligente	08
e. Un environnement intelligent	08
f. Un mode de vie intelligent	08
I.6. Définition de l'internet des objets	09
I.7. Domaines d'applications	10
I.8. Matériels utilisés dans les villes intelligentes	11
I.9. Conclusion	12

Chapitre II : Architecture et spécifications

II.1. Introduction	14
II.2. Spécifications	14
II.2.1. Carte Arduino Mega 2560	14
• Caractéristiques de la carte Arduino Mega	14
• Architecture de la carte Mega 2560	15
• Alimentation	15
II.2.2. Module wifi ESP32	16
• Présentation	16
• Les caractéristiques du module ESP32	17
• Brochage du module ESP32	18
• Programmation de l'ESP32	19
II.3. Plateformes de développement	20
II.3.1. Arduino IDE	20
II.3.2. Proteus Professionnel	21
II.3.3. Isis	21
II.3.4. Fritzing	21
II.4. Composants électroniques	21
II.4.1. Capteurs	21
1. Capteur d'humidité du sol (ou l'hygromètre)	21
• Présentation	21
• Caractéristiques du capteur d'humidité	22
• Branchement du capteur d'humidité sur un Arduino Mega	24
2. Capteur de lumière (photorésistance)	24
• Présentation	24

• Caractéristiques de la photorésistance.....	25
3. Capteur infrarouge.....	25
• Présentation.....	25
• Caractéristiques du capteur infrarouge.....	26
• Branchement du capteur infrarouge sur un Arduino Mega.....	26
4. Module ultrasonique de détection d'obstacle.....	26
• Présentation.....	26
• Caractéristiques du module HC-SR04.....	27
• Principe de fonctionnement du module HC-SR04.....	28
II.4.2. Autres composants.....	28
1. Circuit L293D.....	28
• Présentation.....	28
• Caractéristiques de L293D.....	30
2. Ecran LCD avec I2C.....	30
• Présentation.....	30
• Caractéristiques de l'afficheur LCD I2C.....	31
• Branchement de l'afficheur LCD avec Arduino Mega.....	31
3. Servomoteur.....	32
• Présentation.....	32
• Commande d'un servomoteur.....	32
• Caractéristiques d'un servomoteur.....	33
• Branchement du servomoteur avec Arduino Mega.....	33
4. Mini pompe à eau DC 3-6V.....	34
5. Panneau solaire.....	34
II.5. Conclusion.....	35
Chapitre III : Conception et réalisation	
III.1. Introduction.....	37
III.2. Liaison directe.....	37
III.2.1. Parking intelligent.....	38
a. Présentation.....	38
b. Schéma synoptique.....	39
c. Organigramme.....	40
d. Schéma électronique.....	41
e. Fonctionnement.....	42
f. Conclusion.....	42
III.2.2. Arrosage automatique.....	43
a. Présentation.....	43
b. Schéma synoptique.....	43
c. Organigramme.....	45
d. Schéma électronique.....	46
e. Fonctionnement.....	47
f. Conclusion.....	47
III.2.3 Eclairage publique intelligent.....	48
a. Suiveur solaire.....	48
b. chargeur solaire.....	49

c. Convertisseur DC-DC.....	49
d. Eclairage public.....	50
e. Schéma synoptique.....	51
f. Organigramme de poursuite en double axe	52
g. Schéma électronique	53
1. Suiveur solaire	53
2. Chargeur de batterie 12V	54
3. Convertisseur DC-DC.....	56
4. Eclairage public.....	57
i. Conclusion	57
III.3. Liaison WIFI	58
1. Les différents modes de fonctionnement de l'ESP32	58
2. Les applications de contrôle pour IoT	59
a. Application Web.....	59
1. HTML	60
2. CSS	60
3. JavaScript.....	60
4. Les plateformes IoT Cloud	61
b. Application mobile	61
c. Application Blynk.....	62
3. Configuration de l'application Blynk pour ESP32.....	62
3.1. L'utilisation d'application Blynk pour le Smart Parking	64
3.2. L'utilisation d'application Blynk pour l'arrosage automatique.....	66
III.4. Essais et résultats.....	68
1. Parking intelligent.....	68
a. Liaison direct.....	68
b. Liaison wifi.....	69
2. Irrigation automatique.....	70
a. Liaison direct	70
b. Liaison wifi.....	71
3. Eclairage public intelligent.....	72
III.5. Problèmes rencontrés durant notre travail	73
III.6. Conclusion	74
Conclusion générale	
Bibliographie	
Annexe	

Introduction générale

La ville est considérée comme l'endroit le plus important pour un rassemblement humain, c'est un organe vital, complexe et aussi un lieu de concertation de plusieurs activités et mobilité intense pour des individus afin d'assurer leurs différents besoins de vie.

Le concept de «ville intelligente» qui répond aux nouvelles exigences et aux changements : sociaux, technologiques ...etc. vécus. Ce concept émerge depuis plus d'une dizaine d'années. Affectant un large champ de disciplines telles que l'économie, l'ingénierie, le social ou l'environnement. C'est une notion plutôt récente qui représente une nouvelle approche de développement urbain en mettant en avant l'intégration de nouvelles technologies d'information et de communication (TIC), l'Internet des objets (IoT), les partenariats public-privé, le capital social et humain dans la gestion de la ville dans le but de répondre aux nécessités de la dite ville de façon efficiente.

Dans les dernières années; un nouveau terme est apparu reliant l'intelligence d'une ville à un protocole de communication appelé : IOT 'Internet des Objets' Ce dernier relie les appareils électroniques (autres que les ordinateurs et les Smartphones) à Internet pour un suivi et une gestion efficace des activités quotidiennes. Dans le domaine du développement urbain, les TIC et l'IOT sont des éléments constitutifs importants de la création d'une infrastructure intelligente pour administrer, servir et accompagner la population urbaine en constante augmentation. Cette révolution facilite la création d'objets intelligents permettant des avancées dans de multiples domaines, l'un des domaines les plus affectés par l'émergence de l'IOT est la ville intelligente.

La ville intelligente est une communauté qui comprend ces enjeux et qui est proactive dans l'élaboration de solutions aux problématiques, en raison de la disponibilité des cartes à microcontrôleurs comme l'Arduino, Raspberry Pi et ESP32, avec leur propre plateforme open Source, qui a ouvert la voie aux inventeurs en électronique et les experts pour développer des systèmes intelligents dans le but de garantir le confort et améliorer la qualité et le mode de vie.

Dans ce travail, nous avons procédé à la réalisation pratique, sous forme d'une maquette de ville intelligente, en utilisant l'Arduino comme une carte de communication et de contrôle, et le module ESP32 comme un élément intelligent, en plus quelques capteurs (infrarouge, capteur d'humidité,...). En détaillant le fonctionnement de chaque partie du système (parking intelligent, arrosage automatique, éclairage public...), les algorithmes et les technologies utilisées, et expliquer les circuits proposés et le matériel utilisé.

Notre travail se résume autour de trois chapitres principaux :

1. Dans le premier chapitre nous allons présenter l'état de l'art sur la technologie actuelle des villes intelligentes.
2. Le deuxième chapitre présente l'aspect architectural d'une ville intelligente et donne les spécifications des besoins ainsi que le matériel et équipement utilisé dans la conception de ville intelligente.
3. Le troisième chapitre consiste sur la conception et la réalisation pratique du système proposé. La conclusion synthétise le travail réalisé et donne un aperçu des perspectives qui puissent être développées.

Chapitre 1 :

Etat de l'art

I.1. Introduction

Ces dernières années nous avons constaté une accélération du développement de nouvelles technologies. En effet, l'accroissement de la population, notamment urbaine et les problématiques environnementales accélèrent les investissements en recherche et innovation. Ceci a donné naissance à une nouvelle notion qui est celle de la Smart City, ou ville intelligente en français. L'idée initiale est d'utiliser de nouvelle technologie pour un meilleur développement de la ville.

Dans ce chapitre nous tentons d'appréhender et de présenter les principaux éléments du concept de ville intelligente, soit sa définition, son évolution, ses objectifs, ses caractéristiques, ses avantages et inconvénients. Ainsi que l'IOT qui est un des outils de la smart city.

I.2. Définition de "Ville intelligente"

La ville intelligente c'est un ensemble de solutions technologiques visant à l'optimisation des aires urbaines par le traitement et l'analyse des données. Une ville devient « intelligente » lorsqu'elle peut croiser et centraliser des données et des processus technologiques pour optimiser le fonctionnement de l'ensemble de ses services dans une perspective de développement économique durable, de résilience et de qualité de vie élevée. L'enjeu est de répondre aux difficultés qu'ont les infrastructures des grandes villes occidentales pour faire face à la croissance continue de leur population et de leurs activités.

Alors, l'expression « ville intelligente », s'emploie pour une ville qui développe les technologies de l'information et de la communication (TIC) pour améliorer la qualité des services urbains ou encore réduire ses coûts et intègre divers dispositifs physiques connectés au réseau, constituant l'Internet des objets (IOT), pour optimiser l'efficacité des opérations et des services urbains et se connecter aux citoyens. Ainsi, une ville intelligente devrait être capable de gérer des infrastructures communicantes entre elles, adaptables, durables et plus efficaces, automatisées pour améliorer la qualité de vie des citoyens, dans le respect de l'environnement.

Une ville intelligente est une ville qui investit dans son capital humain et social ainsi que dans ses infrastructures de communication traditionnelles (transport) et modernes (TIC), dans le but d'améliorer la qualité de vie de ses habitants et de dynamiser sa croissance économique, tout en ayant une gestion durable de ses ressources et en utilisant des moyens encourageant la participation citoyenne dans la création de la ville. Caragliu et al. (2012) [1]

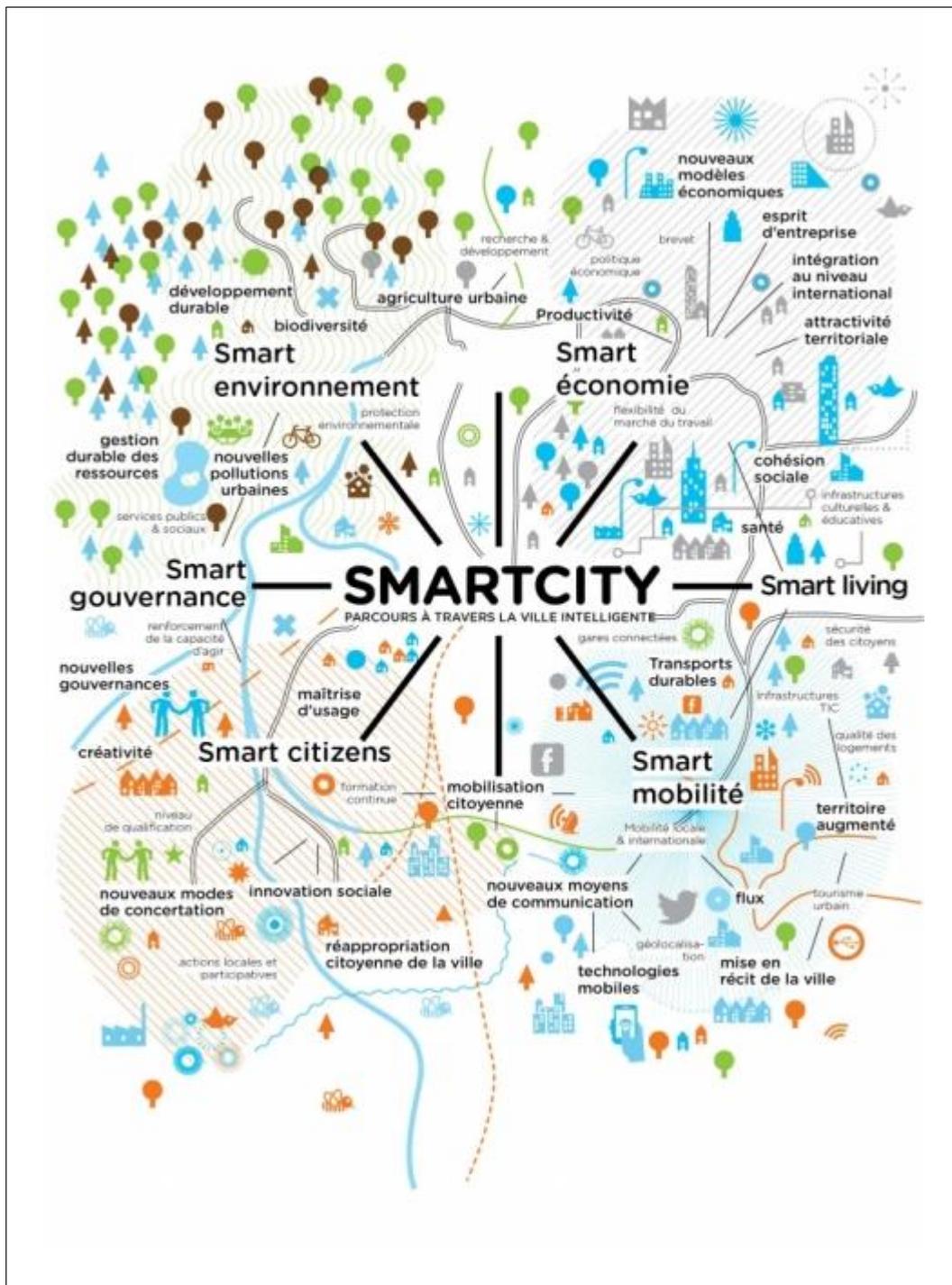


Figure I.1 : Tableau programmatique d'une ville intelligente selon l'agence européenne Dédale consacrée à la culture, aux technologies et à l'innovation sociale [2].

I.3. Évolution du concept de "Smart City "

La smart city n'est pas un concept récent ou lié à l'apparition des nouvelles technologies: il s'agit d'abord d'une ville conçue pour optimiser les coûts, respecter l'environnement et augmenter le bien-être et le confort de ses habitants.

-Les smart cities apparaissaient déjà avant l'émergence des machines: l'ensemble palatial de l'Alhambra à Grenade, érigé entre les XIIIe et XVe siècles, a ainsi été conçu dans un souci d'optimisation et de bonne gestion des ressources naturelles disponibles (maîtrise de l'eau, permaculture, implantation des bâtiments en fonction des conditions d'ensoleillement etc.), en faisant l'ancêtre des villes intelligentes modernes.

-En 1880, Alexander Graham BELL, a tenté une première utilisation d'une forme de fibre optique pour la communication en inventant le « photophone ».

-En 1950, application de la vision « cybernétique » à la planification locale et à la gestion urbaine par un ensemble d'urbanistes américains dans quelques cités américaine set aussi l'apparition de premières cartographies automatiques (SIG).

-En 1969 le département de défense des Etats Unis d'Amérique crée le réseau Arpnet «Internet».par la suite, le développement des technologies de l'audiovisuel, de télécommunication et de l'informatique à partir des années 1990 (explosion numérique), en convergence avec le phénomène de l'urbanisation massive, ont donné naissance à une nouvelle révolution urbaine moderne reposant sur la notion de l'informatique.

-En 2003, la Corée du Sud a commencé à mentionner dans ses documents officiels l'existence d'une stratégie reposant sur la notion de l'informatique omniprésente (ubiquité), et le projet de ville ubiquitaire hyper connectée de Songdo était déjà en construction. Un plan de 12 autres cités a été lancé deux ans après, et c'est sous le nom de « U-Cities » quelles villes intelligentes ont officiellement été inscrites au programme du gouvernement en 2006.

(Lahoulou et al., 2017) [3]

-En 2008, IBM rejoint le mouvement avec son initiative « smarter cities » et en 2009, il a rendu populaire cette expression après avoir identifié les villes comme un marché potentiel prometteur, y voyant une possibilité de développement des TIC. (Coulée and Dieuzeide, 2018) [4]

Aujourd'hui, le concept fait la tendance et de plus en plus concrétisé dans de nombreuses villes au monde. Il fait aussi l'objet de plusieurs conférences, débats, études universitaires et écrits et de nombreuses conventions internationales ont été signées. Rudolf GIFFINGER, expert en recherche analytique sur le développement urbain et régional à l'Université

technologique de Vienne, a défini dans le rapport de son étude « Smart cities : Ranking of European medium sizes cities », la notion de « ville intelligente » ainsi que ses six caractéristiques essentielles, rapport devenu une référence pour toutes les recherches concernant le concept. (Centre of Regional Science, 2007) [5]

I.4. Les objectifs de la Smart City

La Smart City à divers objectifs, elle permet tout d'abord une optimisation des coûts, une organisation plus efficace de la ville et une meilleure maîtrise permettant une augmentation du bien-être des habitants.

- La Smart city est un outil de gouvernance, de digitalisation et de gestion. (LAVATINE.A, 2018) [6]
- La Smart city a pour objectif de transformer la ville en un territoire vivant et dynamique et d'améliorer l'aménagement urbain et l'habitat ainsi que promouvoir l'économie collaborative et la solidarité sous toutes ses formes. Elle développe de nouveaux projets et des stratégies visant à améliorer la qualité de vie de ses citoyens. Elle développe aussi des outils de planification énergétique et crée des nouveaux modes de collaboration entre acteurs de la ville (collectivité, opérateurs énergétiques, aménageurs, bailleurs, promoteurs, exploitants, acteurs privés, universités, habitants, ...).
- La ville intelligente est donc un enjeu indispensable de la transition énergétique pour contribuer à la lutte contre le réchauffement climatique, diminuer l'épuisement des ressources naturelles et favoriser l'accès aux conditions de vie satisfaisantes au plus grand nombre. (Delft, 2019) [7]
- Elle offre grâce à des modèles dynamiques, pourvus d'interfaces, adapter le privilège à chaque citoyen de connaître, d'analyser et d'influer sur les données de son environnement. (RAFAI, 2018) [8]

I.5. Les caractéristiques de la smart city

- La transformation intelligente des villes sous l'impulsion des nouvelles technologies a progressivement intégré des aspects de la vie urbaine aussi variées que l'économie, l'éducation, la démocratie, les infrastructures, les transports, l'environnement, la sécurité et la qualité de vie. (Breux and Diaz, 2017) [9]

L'Union européenne s'inspire de la vision de Rudolf Giffinger, expert en recherche analytique sur le développement urbain et régional à l'université technologique de Vienne, et réfère en

la matière, dont les travaux indiquent que la ville intelligente doit être performante dans six domaines.

a. Une économie intelligente

C'est la compétitivité économique de la ville. Elle se mesure à travers des facteurs comme l'innovation, l'esprit d'entreprise, la productivité, la flexibilité du marché du travail ou encore l'intégration sur le marché nationale et internationale.

b. Des citoyens intelligents

Autrement dit le capital humain et social de la ville. Il est question, bien sûr, du niveau de qualification de la population mais aussi de sa pluralité, de son ouverture d'esprit, de sa créativité, de la qualité des interactions sociales ou de la participation à la vie publique.

c. Une gouvernance intelligente

C'est-à-dire un mode d'administration de la ville transparent, transversal – et partagé – intégrant la participation du citoyen.

d. Une mobilité intelligente

Giffinger met l'accent sur l'accès local et international à la ville, l'existence d'infrastructures connectées exploitant les TIC (technologies de l'information et de la communication) et de systèmes de transports innovants, durables et sûrs.

e. Un environnement intelligent

On parle ici d'écologie et de gestion des ressources. La ville intelligente doit favoriser un environnement de qualité (espaces verts, qualité de l'air), gérer de façon durable ses ressources et œuvrer à la protection de l'environnement. Les éco-quartiers sont des exemples, localisés, d'un environnement géré intelligemment.

f. Un mode de vie intelligent

Il regroupe des facteurs liés à la qualité de vie : culture, santé, sécurité, habitat, éducation, tourisme, cohésion sociale, etc.

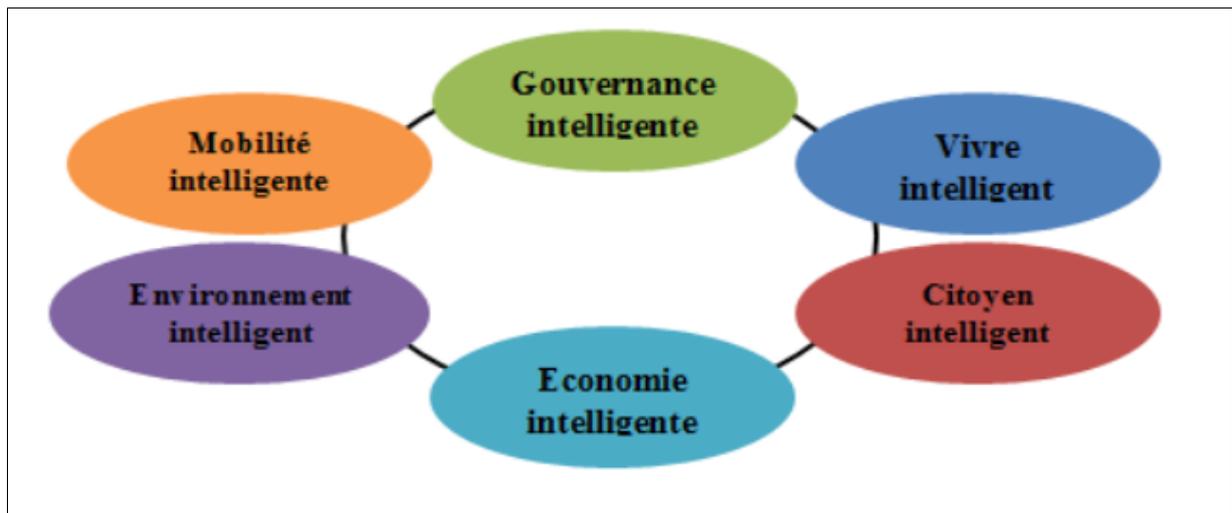


Figure I.2 : Six critères principaux de la Ville intelligente.

1.6. Définition de "Internet des objets"

L'Internet des objets (IOT) désigne l'utilisation d'appareils et de systèmes connectés intelligemment pour exploiter les données recueillies par des capteurs et des actionneurs intégrés dans des machines et d'autres objets physiques et aussi de pouvoir récupérer, stocker, transférer et traiter les données s'y rattachant sans discontinuité entre les mondes physiques et virtuels. Il existe de nombreuses applications de l'Internet des objets dans le monde réel, allant de l'IdO grand public et de l'IdO d'entreprise à l'IdO de fabrication et industriel (IIoT). Les applications IoT couvrent de nombreux domaines, y compris l'automobile, les télécommunications, l'énergie et plus encore. (Gsm, 2014) [10]

Dans le segment grand public, par exemple, les maisons intelligentes équipées de thermostats intelligents, d'appareils intelligents et de systèmes de chauffage, d'éclairage et d'appareils électroniques connectés peuvent être contrôlés à distance via des ordinateurs, des Smartphones ou d'autres appareils mobiles. Dans une ville intelligente, les capteurs et les déploiements IoT, tels que les lampadaires intelligents et les compteurs intelligents, peuvent aider à réduire le trafic, économiser l'énergie, surveiller et résoudre les problèmes environnementaux et améliorer l'assainissement. (Simard, 2018) [11]

Le but de l'Internet des objets est en effet d'étendre à des « choses », c'est-à-dire à des entités matérielles ou logicielles, les fonctionnalités offertes par l'Internet dans le domaine de la communication afin de leur permettre d'échanger entre elles ou avec des humains toutes sortes d'informations ou de données. Dans le succès de l'Internet, deux facteurs clés ont joué

un rôle essentiel : l'adressage qui permet de repérer un utilisateur par son adresse IP (Internet Protocol) ou plus couramment par son adresse Web (l'URL ou Uniform Resource Locator) et l'interopérabilité qui permet, grâce à un ensemble de protocoles qui se sont progressivement imposés dans le monde Internet, à des usagers quelconques de communiquer et d'échanger des données et des services, où qu'ils soient dans le monde et quelle que soit l'origine des équipements qu'ils utilisent [12].



Figure I.3 : Des exemples sur l'internet des objets.

I.7. Domaines d'applications

Une large gamme de services intelligents peut être déployée dans le cadre d'initiatives de villes intelligentes. Ceux-ci peuvent être regroupés dans les domaines suivants:

- Transport : Réduire les embouteillages et rendre les déplacements plus efficaces, sécurisés et sûrs.

- Environnement : Gérer et protéger les ressources de la ville, contrôler les coûts et ne fournir que la quantité d'énergie ou d'eau requise, tout en réduisant les déchets.
- Bâtiment : Pour améliorer la qualité de vie dans les bâtiments de la ville (par exemple, la maison, les bâtiments commerciaux, etc.).
- Education : Pour accroître l'accès aux ressources éducatives, améliorer la qualité de l'éducation et réduire les coûts.
- Tourisme : Pour améliorer l'accès aux sites culturels et améliorer les services de divertissement.
- Soins de santé : Pour améliorer la disponibilité des services de soins de santé, fournir des soins de santé et de prévention et devenir plus rentable.
- Sécurité publique : Utiliser des informations en temps réel pour anticiper et répondre rapidement aux urgences.
- Agriculture : Pour améliorer la gestion de l'agriculture.
- Gestion de la ville : pour rationaliser la gestion de la ville et fournir de nouveaux services de manière efficace.

I.8. Matériels utilisés dans les villes intelligentes



Figure I.4 : smart parking.



Figure I.5 : poteaux intelligents.

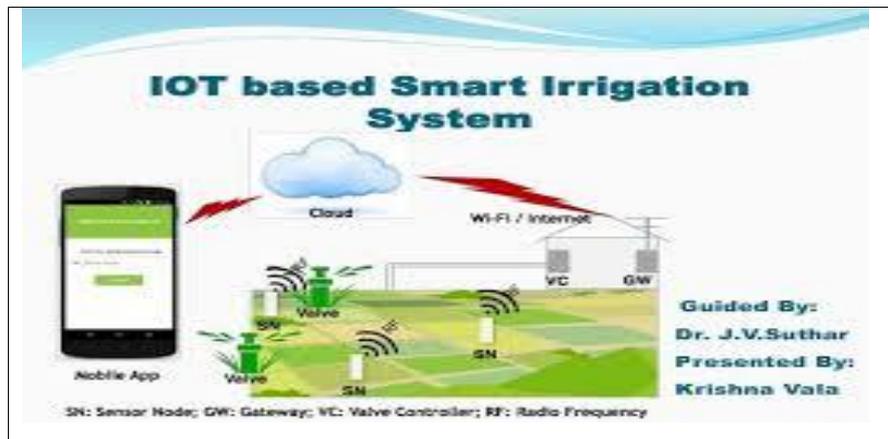


Figure I.6: Iot based smart irrigation system.

I.9. Conclusion:

A travers ce chapitre, nous constatons qu'il n'existe pas une seule définition de smart city. Pour qu'une ville soit intelligente; il est impératif d'avoir les six critères principaux de Rudolf Giffinger ainsi que l'utilisation des technologies de l'information et de communication dans ses infrastructures, services et mode de vie des citoyens pour répondre aux besoins de sa population, mais aussi pour améliorer la qualité de vie. En effet, la transformation numérique d'une ville doit permettre de répondre au désir de résider au sein d'une ville intelligente, propre et dynamique, qui devient éco responsable et permet un enrichissement financier grâce à une meilleure gestion des coûts et des ressources.

Chapitre 2 :

Architecture et spécifications

II.1. Introduction

Ce chapitre est consacré à la présentation du matériel et logiciels utilisés pour développer notre système et ainsi que les différentes plateformes d'exécution de ses différentes parties.

II.2. Spécifications

II.2.1. Carte Arduino Mega 2560

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM (Power Width Modulation (ou Modulation de Largeur d'Impulsion ' MLI'), 16 entrées analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complexes. Elle est aussi compatible avec MATLAB.

Caractéristiques principales :

- Alimentation via port USB ou d'un connecteur jack.
- Mémoire flash : 256 KB.
- d'un connecteur ICSP (programmation "in-circuit"),
- et d'un bouton de réinitialisation (reset).

Les Caractéristiques de la carte Arduino Mega

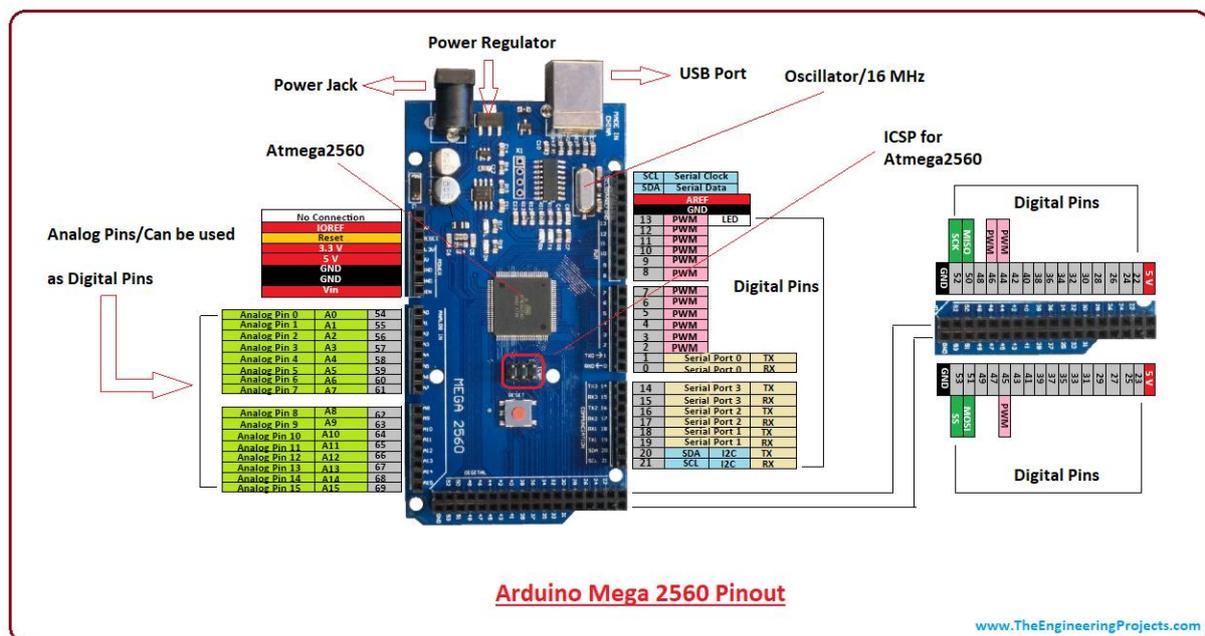
Microcontrôleur	Atmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée – 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le bootloader

Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Tableau II.1 : Les Caractéristiques de la carte Arduino Mega 2560.

Architecture de la carte MEGA 2560 :

La carte Arduino MEGA 2560 est composée de plusieurs éléments essentiels qui sont présentés dans la figure suivante :



II.2.2. Module WiFiESP32

Présentation

L'ESP32 est une carte à base de microcontrôleur avec des modules Wifi et Bluetooth intégrés. Très simple d'utilisation il est léger et possède une capacité de mémoire et de calculs supérieurs à l'Arduino. Ce qui en fait une carte idéale pour l'apprentissage de la programmation, le développement d'objets connectés ou de serveur.

La base, pour créer des objets connectés, est de les connecter à un réseau comme le réseau WiFi. Il intègre un émetteur-récepteur WiFi ce qui lui permet de se connecter à des réseaux existants ou bien de mettre en place son propre réseau. Grâce à cela, il est possible de créer un serveur qui héberge une interface Web permettant de piloter le microcontrôleur ESP32 à distance.

L'ESP32 contenant un microprocesseur double cœur fonctionnant en 32 bits. Le microcontrôleur ESP32 possède 48 GPIO dont 25 sont disponibles sur les broches. Sur ces 25 broches:

- 15 peuvent être utilisées pour des entrées analogiques
- 25 peuvent être utilisées comme sorties PWM
- 9 comme capteurs capacitifs.
- Certaines sont réservées pour les protocoles de communication série (SPI, I2C, Serial).

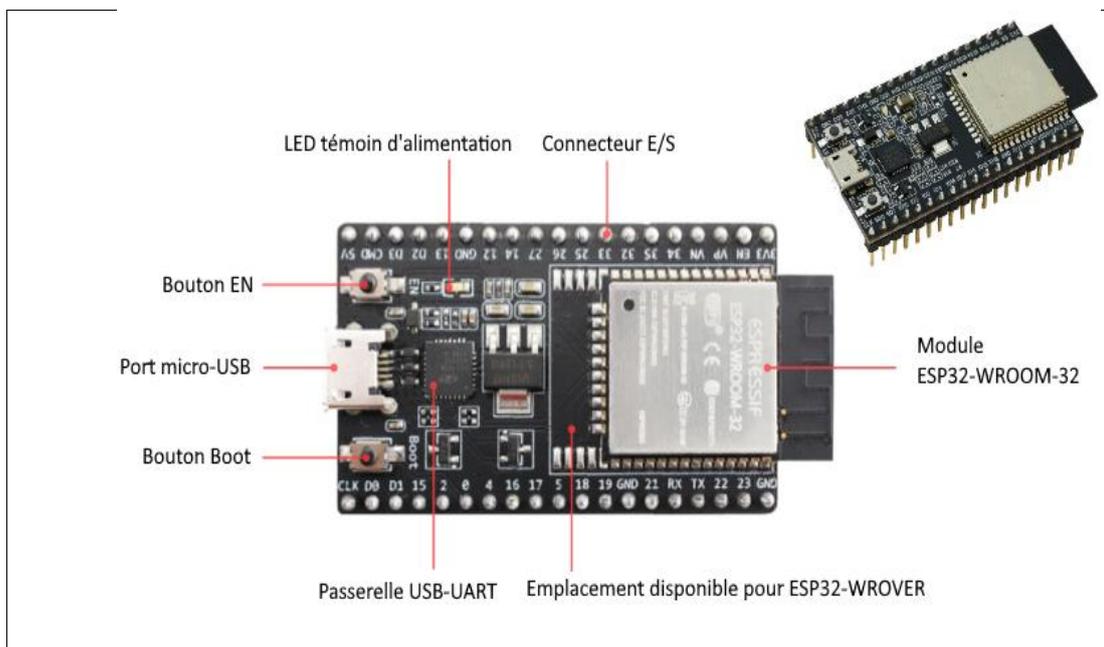


Figure II.2 : Carte ESP32.

Les Caractéristiques du module ESP32

Caractéristique	ESP32
CPU	Xtensa Dual-Core 32-bit LX6 avec 600 DMIPS
Vitesse du WiFi	802.11n jusqu'à 150 Mbps
Protocole WiFi	802.11 b/g/n (2.4 Ghz)
GPIO	36
Bluetooth	OUI
CAD	Deux canaux DAC 8 bits
ADC	RAS 12 bits
Canaux ADC	8 canaux
Référence ADC V	1100mV
SPI/I2C/I2S/UART	4/2/2/3
Modes WiFi	Station/SoftAP/SoftAP+Station/P2P
Capteur tactile	OUI (8-Channel)
Capteur de température	OUI
Capteur à effet Hall	OUI
SRAM	520 kB (8 kB de SRAM dans RTC)
FLASH (externe)	4Mbytes (également disponible plus haut)
ROM	448 kB ROM pour le démarrage et les fonctions de base
Protocoles de réseau	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
Interface périphérique	UART/SDIO/SPI/I2C/I2S/IR Télécommande GPIO/ADC/DAC/Touch/PWM/LED
Plage de température de fonctionnement	-40°C ~ +85°C
Tension de fonctionnement	2,5V ~ 3,6V
Courant de fonctionnement	Moyenne : 80 mA

Tableau II.2 : Les Caractéristiques du module ESP32.

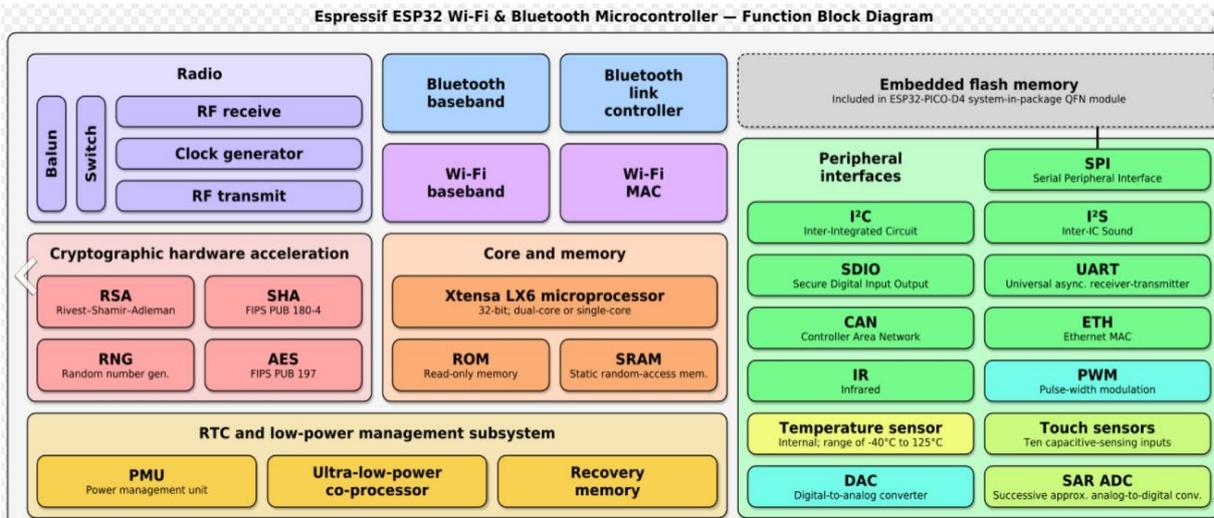


Figure II.3 : Schéma fonctionnel du l'ESP32.

Brochage du module ESP32

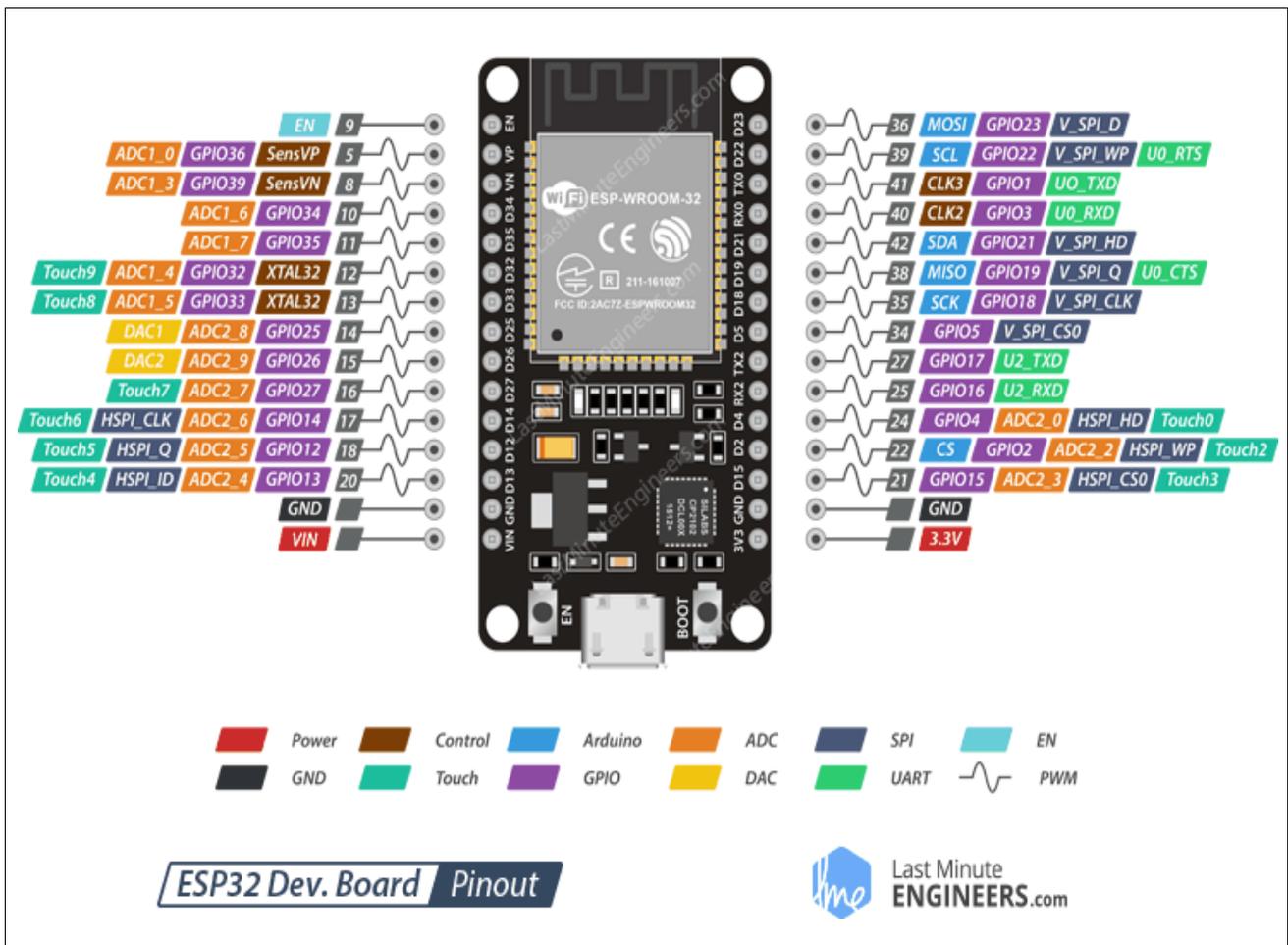


Figure II.4 : brochage du module ESP32.

Programmation de l'ESP32

Il existe une variété de plates-formes de développement qui peuvent être équipées pour programmer l'ESP32 :

- En script Lua, avec le firmware NodeMCU.
- En C, avec le SDK d'Espressif.
- En C, avec l'IDE Arduino.
- En JavaScript, avec le firmware Espruino.
- En MicroPython, avec le firmware MicroPython.
- ou l'une des plates-formes répertoriées sur [WiKiPedia](#) .

Pour la programmation de l'ESP32, nous allons utiliser l'environnement de développement d'Arduino car cet IDE présente beaucoup d'avantages (Gratuit, open-source, Multiplateforme ...), il faut installer le dernier IDE Arduino, ensuite, télécharger le contenu du référentiel ESP32-arduino et librairie qui vont permettre de compiler pour l'ESP 32. Pour vérifier l'installation correcte de l'esp32 dans Arduino IDE nous avons testé l'exemple simple de LED interne déjà disponible dans Arduino IDE pour la carte de développement ESP32 pour l'IoT. Ces exemples sont installés lors de l'installation de la bibliothèque ESP32 dans Arduino IDE.

Étapes à suivre

- Connexion de la carte à l'ordinateur.
- Sélection de notre carte (fenêtre outil >> Cartes : clique sur le module Dev -type de carte)
- Sélection du port auquel notre esp32 est connecté (fenêtre outil >> port)
- Télécharger l'exemple et vérifier sur l'esp32 le clignotement de LED (Fichier >> Exemple >> Basics >> Blink).

```
18 // by COLBY NEWMAN
19
20 This example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
34   delay(1000); // wait for a seco
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED of
36   delay(1000); // wait for a seco
37 }

```

Bibliothèque non valide trouvée dans C:\Program Files (x86)
Bibliothèque non valide trouvée dans C:\Program Files (x86)
Bibliothèque non valide trouvée dans C:\Program Files (x86)

1 ESP32 Dev Module sur COM7

Figure II.5: exemple programme Blynk avec ESP32.

II.3. Plateformes de développement

II.3.1. Arduino IDE

La partie software de ce projet s'articule autour de l'environnement de développement intégré d'Arduino IDE (Integrated Développement Environement) qui offre une interface minimale pour rédiger des programmes appelés « Sketch », les compiler et les transférer dans la carte Arduino à travers une liaison USB, intégrant aussi un moniteur de port série pour l'affichage des données.

II.3.2. Proteus professionnel

Proteus Professional est une suite logicielle pour l'électronique. Développé par Lab center électroniques, le logiciel inclus dans Proteus Professional permet la CAD (Computer Aided Construction) dans le domaine de l'électronique. Deux logiciels

Les principaux composants de cette suite logicielle sont: (ISIS, ARES, PROSPICE)

Cette suite logicielle est bien connue dans le domaine de l'électronique. De nombreuses entreprises et organisations de formation utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus Professional a d'autres avantages :

- Pack contenant un logiciel facile et rapide à comprendre et à utiliser.
- Le support technique est performant.
- L'outil de prototypage virtuel permet de réduire les coûts matériels et logiciels lors de la conception d'un projet.

II.3.3. Isis

Est un éditeur de schéma qui intègre un simulateur analogique / logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

II.3.4. Fritzing

Fritzing est un logiciel libre de conception de circuit imprimé permettant de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon¹, ce logiciel est destiné aux non-professionnels de l'électronique.

Il est adapté aux débutants ou confirmés en électronique pour faire rapidement des circuits simple, et est également un bon outils didactiques pour apprendre à bidouiller en électronique par la pratique.

II.4. Composants électroniques

II.4.1. Capteurs

Est un dispositif électronique qui transforme une grandeur physique analogique d'entrée (température, pression, déplacement,...) en une autre grandeur de nature électrique (tension, courant, charge).

1. Capteur d'humidité du sol (ou l'hygromètre)

Présentation

La sonde en forme de fourche à deux conducteurs exposés, agit comme une résistance variable (tout comme un potentiomètre) dont la résistance varie en fonction de la teneur en eau du sol. Cette résistance est inversement proportionnelle à l'humidité du sol:

- Plus il y a d'eau dans le sol signifie une meilleure conductivité et se traduira par une résistance plus faible.
- Moins il y a d'eau dans le sol signifie une conductivité médiocre et se traduira par une résistance plus élevée.

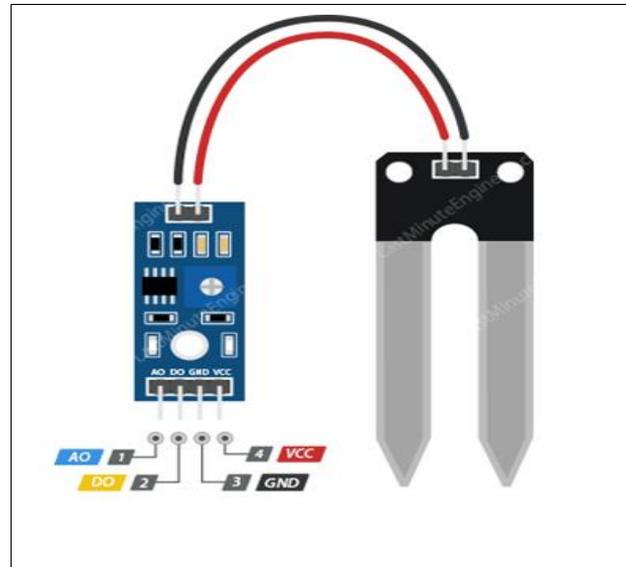


Figure II.6 : capteur d'humidité du sol.

Il est généralement utilisé pour détecter l'humidité du sol. Ainsi, il sert à construire un système d'arrosage automatique ou pour surveiller l'humidité du sol des plantes. Ce dernier dispose d'un potentiomètre intégré pour le réglage de la sensibilité de la sortie numérique (D0), d'une LED d'alimentation et d'une LED de sortie numérique.

Caractéristiques du capteur d'humidité

Le tableau résume les caractéristiques du capteur d'humidité du sol:

VCC	3,3v ou 5v
GND	Masse (0v)
A0	Analogue Output Interface 0 - 200 dans l'eau 200 - 400 milieu d'arrosage parfait 400 - 700 milieu d'arrosage moyen 700 – 950 sol sec 1023 dans l'air
D0	digital output interface (0 ou 1)
<i>Dimension</i>	20mm x 60mm
<i>Courant</i>	35 mA

Tableau II.3 : les caractéristiques de capteur d'humidité du sol.

Note

Un problème communément connu avec ces capteurs est leur courte durée de vie lorsqu'ils sont exposés à un environnement humide. La puissance appliquée à la sonde accélère constamment le taux de corrosion de manière significative.

Solution

Pour surmonter cela, nous vous recommandons de ne pas alimenter le capteur en permanence, mais de ne le mettre sous tension que lorsque vous prenez les mesures.

1- Un moyen simple d'y parvenir est de connecter la broche VCC à une broche numérique d'un Arduino et de la régler sur **HIGH** ou **LOW** selon vos besoins.

De plus, la puissance totale consommée par le module (avec les deux LED allumées) est d'environ 8 mA, il est donc possible d'éteindre le module par une broche numérique sur un Arduino.

2- Ou on utilise un capteur d'humidité du sol capacitif **v1.2**. la figure ci-dessous présenté un capteur de type capacitif :



Figure II.7 : Capteur d'humidité du sol capacitif.

Branchement du capteur d'humidité sur Arduino Mega

On branche donc :

- **AO** (sortie analogique) du capteur sur la broche **A0** de l'arduino.
- **DO** (sortie numérique) broche donne la sortie numérique du circuit comparateur interne.
(On n'a pas utilisé la broche numérique)
- Le **VCC** du capteur au **5V** de l'arduino.

- Le **GND** du capteur au **GND** de l'arduino.

2. Capteur de lumière (photorésistance)

Présentation

La photorésistance **LDR (Light Dependent Resistor)** est un dipôle dont la résistance dépend de la lumière qu'il reçoit, plus il reçoit de lumière, moins il devient résistif, c'est-à-dire plus de courant peut le traverser. La partie sensible du capteur est une piste de sulfure de cadmium en forme de serpent : l'énergie lumineuse déclenche une augmentation de porteurs libres dans ce matériau, de sorte que sa résistance électrique diminue à priori.



Figure II.8 : Photorésistance LDR.

Caractéristiques de la photorésistance

Le tableau ci-dessous résume les caractéristiques de la photorésistance

Résistance à la lumière	1 K Ω
Résistance dans l'obscurité	10 K Ω
Tension Max	150 V
Puissance Max	100 MW

Tableau II.4 : Caractéristiques de LDR.

3. Capteur infrarouge (IR)

Présentation

Nous avons utilisé dans notre projet un capteur infrarouge FC-51 qui joue le rôle du capteur de barrière. Lorsqu'il détecte l'obstacle, le voyant vert de la carte s'allume et l'interface de sortie émet un signal de niveau bas continu.

La distance peut être ajustée par potentiomètre, la gamme est de 2-30 cm.

Le facteur clé affectant la distance de détection est la réflectivité et la forme de la cible. La distance de détection des objets noirs est la plus petite, mais elle est la plus éloignée pour le blanc. Plus la surface des objets est petite, plus la distance de détection est réduite.

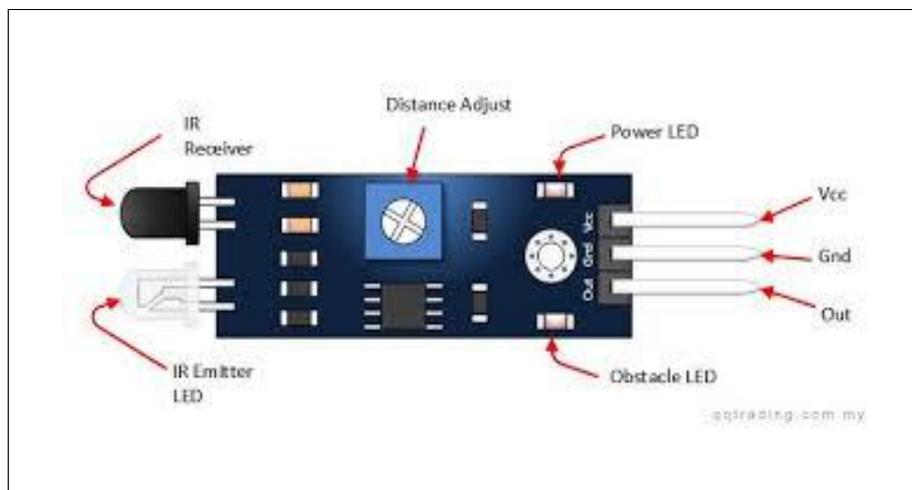


Figure II.9: Capteur infrarouge.

Caractéristiques de capteur IR

- Paire infrarouge émetteur et récepteur.
- Distance Effective gamme 2 ~ 30 cm.
- tension de fonctionnement de 3.3 V-5 V.
- Plage de détection du capteur peut être ajustée via un potentiomètre.
- Peut être utilisé pour robot obstacle d'évitement, suivi de lignes et noir et blanc etc..

Branchement du capteur infrarouge sur arduino MEGA

On branche donc :

- **OUT** du **IR** sur la broche **digital** de l'arduino.
- Le **VCC** du **IR** au **5V** de l'arduino.
- Le **GND** du **IR** au **GND** de l'arduino.

4. Module ultrasonique de détection d'obstacle

Présentation

Le détecteur HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet. Peu importe l'intensité de la lumière, la température ou le type de matière, le capteur pourra facilement détecter s'il y a un obstacle devant lui. Tout de fois, il peut être contraint sur certains types de couleurs tels que le noir (contraste), ou encore sur la matière comme le textile. Son champ de vision est de 90° environ selon l'environnement. Si une impulsion de plus de 10µS et détecter, alors le capteur envoie une série de 8 impulsions à ultrason de 40kHz et attends le réfléchissement du signal. Ensuite, en ayant en tête la vitesse du son, il effectue un rapide calcul pour déterminer la distance.

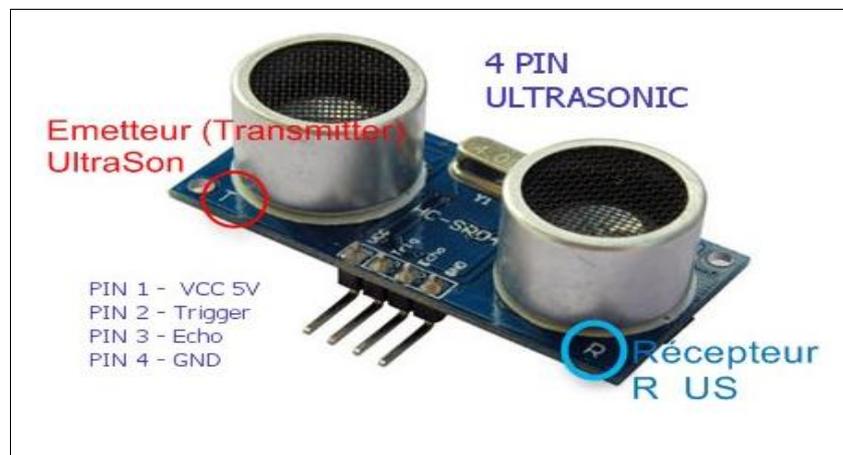


Figure II.10 : Capteur ultrason.

Caractéristiques du module HC-SR04

Alimentation	3,3 ou 5V
Consommation en utilisation	15 mA
Gamme de distance	2 cm à 5 m
Résolution	0.3 cm
Précision	1 cm
Angle de mesure	< 15°
Fréquence	40 kHz
Déclenchement	impulsion TTL positive de 10µs
	impulsion positive TTL proportionnelle à la

Signal écho	distance.
Calcul	$\text{distance (cm)} = \text{impulsion } (\mu\text{s}) / 58$
Dimensions	45 x 20 x 18 mm

Tableau II.5 : Caractéristiques du module HC-SR04.

Principe de fonctionnement du module HC-SR04

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10 μs sur la broche 'Trig Input'; cela déclenche la mesure. En retour la sortie 'Output' ou 'Echo', va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet.

Voici une représentation graphique de la séquence de fonctionnement du module :

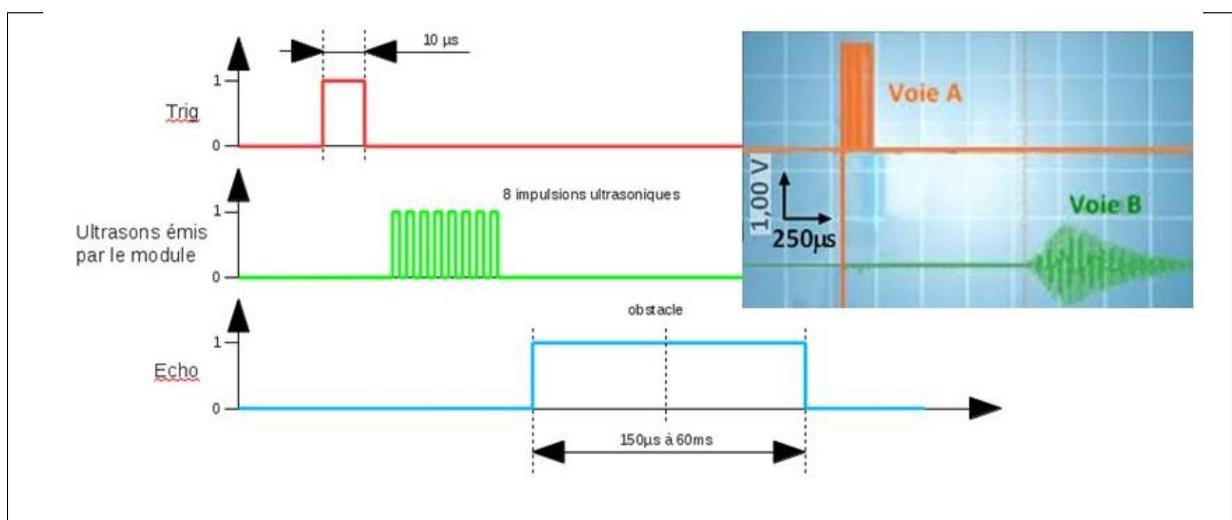


Figure II.11 : Séquence de fonctionnement du module HC-SR04.

II.4.2. Autres composants

1. Circuit L293D

- **Présentation**

Le **L293D** est un double pont-H, il est utilisé pour commander la rotation de deux moteurs dans les deux sens en raccordant les sorties de façon appropriées, Le L293 est un circuit

intégré monolithique, à haut voltage, grand courant et 4 canaux ; cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC alimentés jusqu'à 36 volts et il est capable de piloter une paire de moteurs à courant continu ou un moteur pas à pas. Le circuit peut fournir un maximum de 600 mA par canal. En utilisant différentes combinaisons de Input 1 et Input 2, il devient possible de démarrer, stopper ou inverser le courant ainsi commander la vitesse (les branches responsables enable 1 enable 2, les pattes 1 et 9 du L293d).

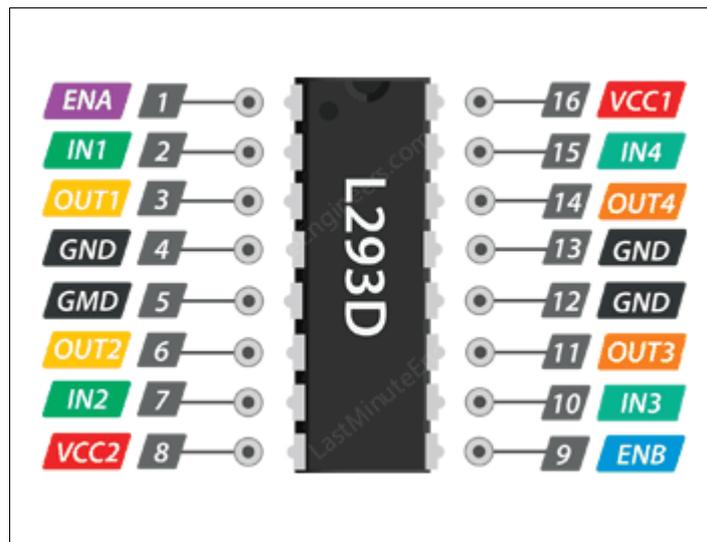


Figure II.12: Brochage de L293D.

Le principe de fonctionnement du circuit est simple, il suffit de brancher les deux moteurs sur les broches (OUT1, OUT2) pour le moteur 1 et (OUT 3, OUT4) pour le moteur 2.

Le circuit est alimenté par une double alimentation ; VCC1=5V pour la commande et VCC2 =12V pour l'alimentation des moteurs.

Pour activer les deux parties du circuit, un niveau logique haut doit être appliqué aux entrées *ENA* et *ENB*.

Les entrées logiques de contrôle IN1 et IN2 pour la commande du moteur 1 (OUT1, OUT2) et IN3 et IN4 pour la commande du moteur 2 (OUT 3, OUT4) déterminent le sens de rotation des moteurs.

Le sens de rotation d'un moteur peut être contrôlé en appliquant soit un niveau logique haut (5 volts) soit un niveau logique bas (0V) à ces broches. Le tableau ci-dessous illustre comment cela est fait.

<i>EN 1</i>	<i>EN 2</i>	<i>Sens de rotation</i>
0	0	Moteur éteint
1	0	Avant
0	1	Arrière
1	1	Moteur éteint

Tableau II.6 : Contrôle du sens de rotation d'un moteur à CC.

- **Caractéristiques de L293D**

Nombre de pont-H	2
Courant Max régime continu	600 mA (x2)
Courant de pointe Max < 2ms	1200 mA
Vs Max Alim moteur	36 V
Vs Max Alim logique	7 V
Nombre de Broche	16 DIP
Perte de tension	1.3 à 1.4v (typical)

Tableau II.7: Caractéristiques de L293D.

2. Ecran LCD avec un I2C

- **Présentation**

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement les caractéristiques apparaissent sombres sur fond clair.

I2C (Signifie : Inter-Integrated Circuit, en anglais)

Est un bus série synchrone bidirectionnel half-duplex, où plusieurs équipements, maîtres ou esclaves, peuvent être connectés au bus.

La connexion est réalisée par l'intermédiaire de deux lignes :

- **SDA** (Serial Data Line) : ligne de données bidirectionnelle.
- **SCL** (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle.

Il ne faut également pas oublier la masse qui doit être commune aux équipements.

Le nombre maximal d'équipements est limité par le nombre d'adresses disponibles, 7 bits d'adressage et un bit R/W (lecture ou écriture), soit 128 périphériques, mais il dépend également de la capacité (C_B) du bus.

Note

Ces modules sont actuellement fournis avec une adresse I2C par défaut soit 0x27 ou 0x3F. Pour déterminer quelle version vous avez vérifié l'adaptateur I2C tableau noir sur la face inférieure du module. S'il y a 3 jeux de patins étiquetés A0, A1, A2 et puis l'adresse par défaut sera 0x3F. S'il n'y a pas pds l'adresse par défaut sera 0x27. Le module est équipé d'un potentiomètre de réglage de contraste sur la face inférieure de l'écran. Cela peut nécessiter de réglage pour l'écran pour afficher le texte correctement. Si la pression est appliquée à la carte fille I2C il est possible de se plier et de venir en contact avec le module LCD.



Figure II.13 : Afficheur LCD I2C.

- **Caractéristiques de l'afficheur LCD I2C**
 - Adresse I2C : 0x20-0x27 (0x20 par défaut)
 - Rétro-éclairé : (bleu avec couleur char blanc).
 - Tension d'alimentation : 5V.
 - Interface: I2C / TWI x1, interface Gadgeteer x2.
 - Contraste ajustable.

- **Branchement de l'afficheur LCD sur arduino MEGA**

On branche donc :

- SDA du LCD sur la broche SDA de l'arduino.
- SCL du LCD sur la broche SCL de l'arduino.
- Le VCC du LCD au 5V de l'arduino.
- Le GND du LCD au GND de l'arduino.

Carte	I2C / TWI pins
Arduino Mega 25600	20 (SDA), 21 (SCL)
Esp 32	D21 (SDA), D22 (SCL)

Tableau II.8: Brochage LCD I2C avec ESP32 et Arduino MEGA.

3. Servomoteur

- **Présentation**

Un servomoteur (souvent abrégé en « servo », provenant du latin *servus* qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi. Le servomoteur intègre dans un même boîtier, la mécanique (moteur et engrenage), et l'électronique, pour la commande et l'asservissement du moteur. La position est définie avec une limite de débattement d'angle de 180 degrés, mais également disponible en rotation continue.

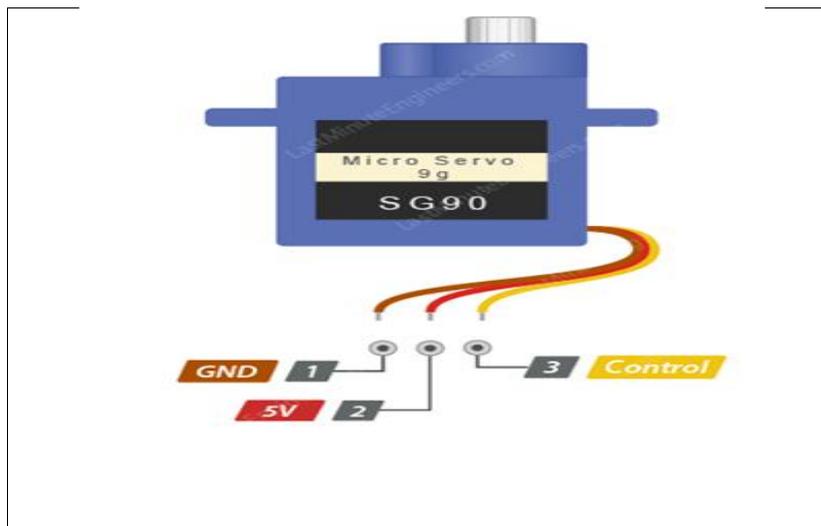


Figure II.14 : Servomoteur SG90.

- **Commande d'un servomoteur**

La commande du moteur est basée sur le signal PWM d'une fréquence égale à 50 Hz (période de 20 ms). Contrairement à la variation de la vitesse d'un moteur à CC dont on peut ajuster le rapport cyclique du 0% à 100%, le servomoteur n'autorise qu'une variation allant de 1 ms à 2 ms : D'une autre façon un rapport cyclique variant de 1 ms /20 ms=5% à 2 ms/20 ms=10%. Si on utilise le formalisme 0-180° alors:

- Si l'impulsion est élevée pendant 1 ms, l'angle d'asservissement sera nul (0°).

- Si l'impulsion est élevée pendant 1,5 ms, alors le servo sera à sa position centrale (90°).
- 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.

La durée des impulsions peut parfois varier selon les marques et elles peuvent être de 0,5 ms pour 0 degré et de 2,5 ms pour 180 degrés.

Le schéma ci-dessous montre l'illustration du signal de contrôle source du servomoteur :

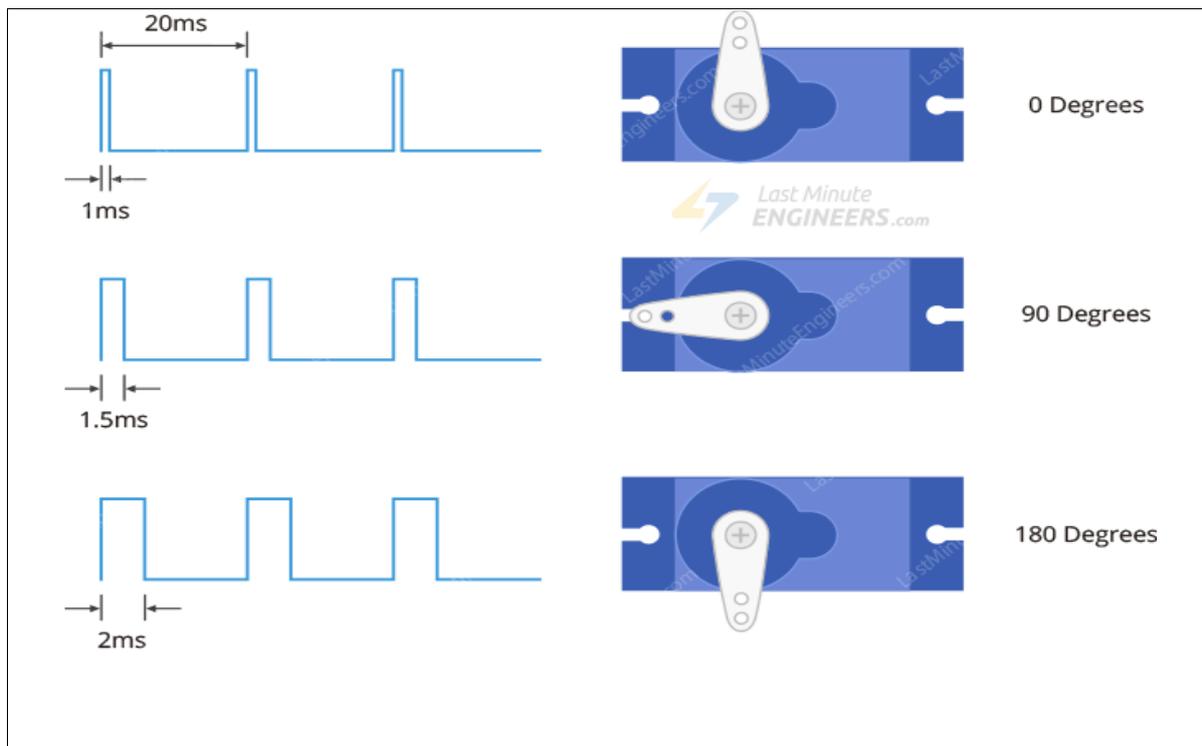


Figure II.15 : Illustration du signal de contrôle Source.

- **Caractéristiques du servomoteur**

Alimentation	4.8 à 6V
Angle de rotation	180°
Couple	1.3 Kg.cm
Vitesse	0.12. sec / 60°
Dimensions	23.2 x 12.5 x 22 mm
Poids	9g

Tableau II.9 : Caractéristiques du servomoteur.

- **Branchement du Servomoteur sur arduino Mega**

- Le fil du **Servo** sur la broche **PWM** de l'arduino.
- Le fil rouge du **Servo** au **5V** de l'arduino.
- Le fil marron du **Servo** au **GND** de l'arduino.

4. Mini pompe à eau DC 3-6V

Modèle immergée dans l'eau.



Figure II.16 : Mini pompe à eau.

5. Panneau solaire

Le panneau solaire (PV) est un élément d'une installation solaire destiné à recueillir l'énergie solaire pour la convertir en énergie électrique.



Figure II.17 : Panneau solaire.

II.5. Conclusion

Dans ce chapitre, nous avons donné une description détaillée du matériel utilisé pour notre projet et les caractéristiques techniques de chaque partie, ainsi que les logiciels utilisés.

Chapitre 3 :

Conception et réalisation

III.1. Introduction

Après avoir évoqué dans le chapitre précédent le matériel utilisé pour démontrer notre système, il y a lieu de présenter actuellement le volé pratique de notre système objet de discussion.

La partie conception concerne la simulation et la réalisation de trois parties principales de notre travail à savoir :

- Parking intelligent.
- Arrosage automatique.
- Eclairage public intelligent.

Dans ce chapitre, nous allons présenter l'architecture globale du système proposé qui sera déroulée consécutivement sur deux étapes essentielles

1^{ère} étape : liaison directe (montage et fonctionnement du système).

2^{ème} étape: liaison par Wifi (création de l'application : BLYNK et l'adopter pour commander notre système à distance).

III.2. Liaison directe

La figure ci-dessous représente le schéma global de notre projet, qui est divisé en trois sous-systèmes.

Le premier sous-système est un parking intelligent composé de deux capteurs infrarouge pour détecter la présence des véhicules, un servomoteur pour commander l'ouverture et la fermeture de la barrière, et un afficheur LCD I2C pour l'affichage du nombre de places disponibles dans le parking.

Le deuxième est un système d'irrigation automatique comprenant un capteur d'humidité enfoncer dans le sol, ainsi qu'une pompe émerger dans l'eau qui arrose le jardin quand le sol est sec, et un capteur ultrason pour surveiller le niveau d'eau dans le réservoir, qui travail en parallèle avec un pompe extérieur relia à la conduite d'eau qui remplit le réservoir lorsque le niveau est bas.

Le dernier sous-système représente la conception d'un éclairage public intelligent, concrétisé par un suiveur solaire composé d'un panneau photovoltaïque, quatre photorésistances (LDR) et deux servomoteurs qui orientent le panneau vers la source de lumière pour charger une batterie 12V. Un convertisseur DC-DC joue le rôle d'élevateur de tension quand celle-ci est inférieure à la tension de seuil voulue.

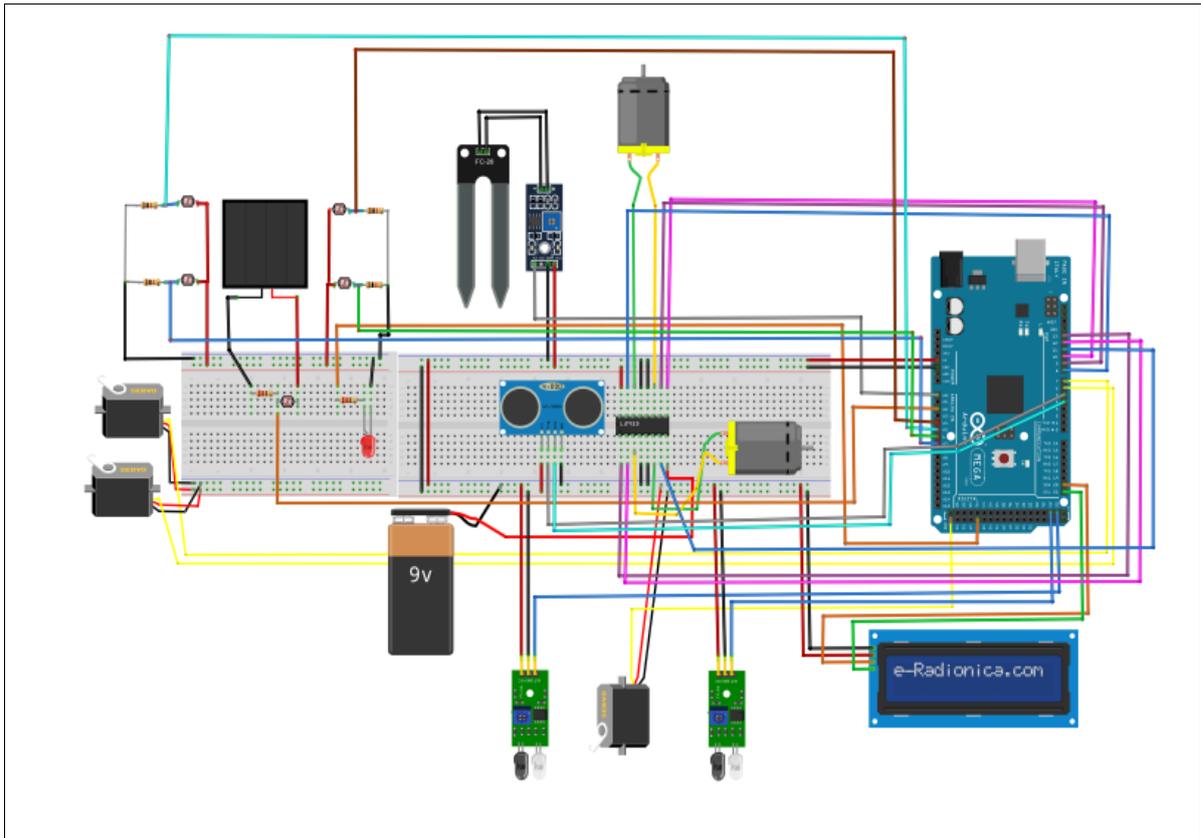


Figure III.1: Schéma globale du système.

III.2.1. Parking intelligent

a. Présentation

Dans cette partie du projet nous allons simuler le fonctionnement du parking automatisé avec une carte Arduino. Cette maquette reprend le fonctionnement général des systèmes automatisés qui permettent l'accès au parking.

La Maquette du parking contient huit (06) places et comprend :

- Barrière mécanique actionnée par un servomoteur (montée ou descente de la barrière)
- Capteur infrarouge à l'entrée du parking détecte la présence d'une voiture à l'entrée.
- Capteur infrarouge à la sortie du parking détecte le véhicule sortant.
- Ecran LCD I2C permet l'affichage du nombre de places disponibles dans un parking. Cela a pour but de comptabiliser le nombre de places disponible dans un parking.

b. Schéma synoptique

Le schéma synoptique de fonctionnement du parking est représenté par la figure suivante :

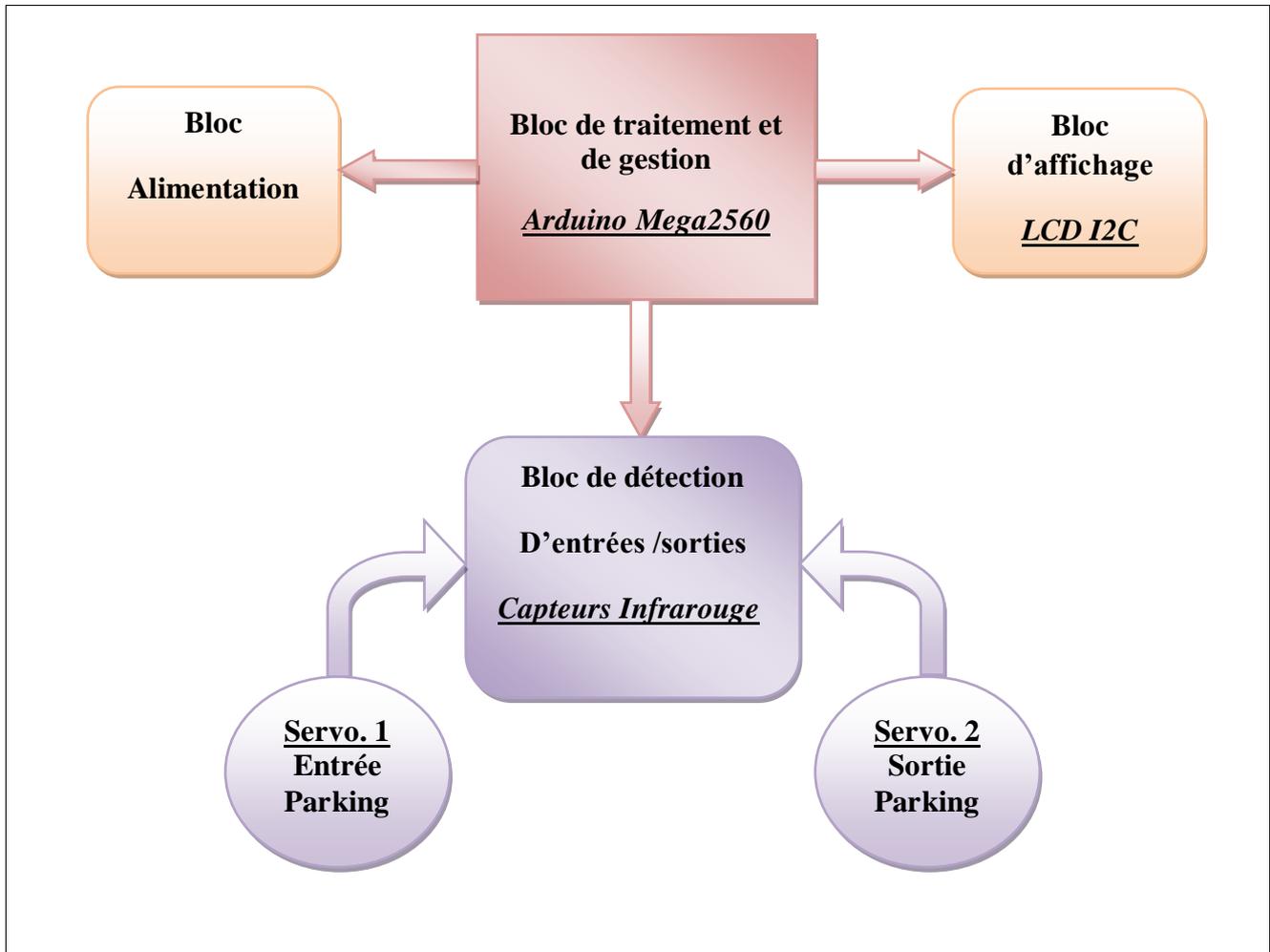


Figure III.2 : schéma synoptique.

- **Circuit de commande** : Assurant la synchronisation des différentes fonctions de service : il reçoit les données envoyées par les capteurs et la commande des actionneurs et assure aussi le comptage et décomptage du nombre d'entrées et sorties.
- **Capteurs** : Ils détectent la présence des véhicules à chaque entrée ou sortie du parking.
- **Actionneur** : Commande des servomoteurs.
- **Affichage** : Affiche le nombre d'entrées et des sorties de véhicules au parking.

C. Organigramme

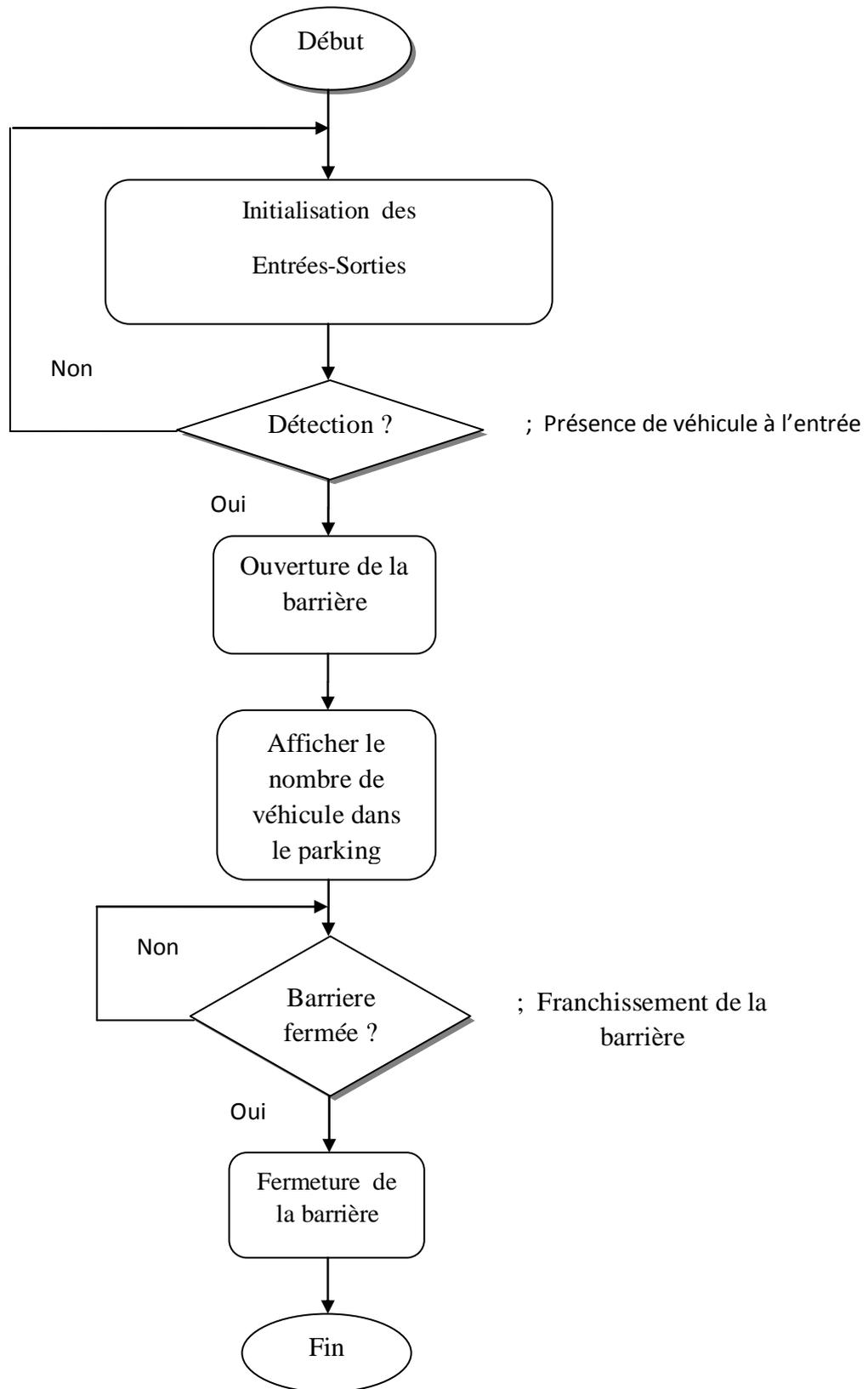


Figure III.3: Organigramme de fonctionnement.

A noter :

Dans le cas où un véhicule quitte le parking, le système comptabilise systématiquement le nombre de places libres, ce qui se traduit aussi par l’affichage du nombre de place disponible dans notre parking.

d. Schéma électronique

Une simulation par le logiciel Proteus, de notre parking est illustrée dans la figure suivante :

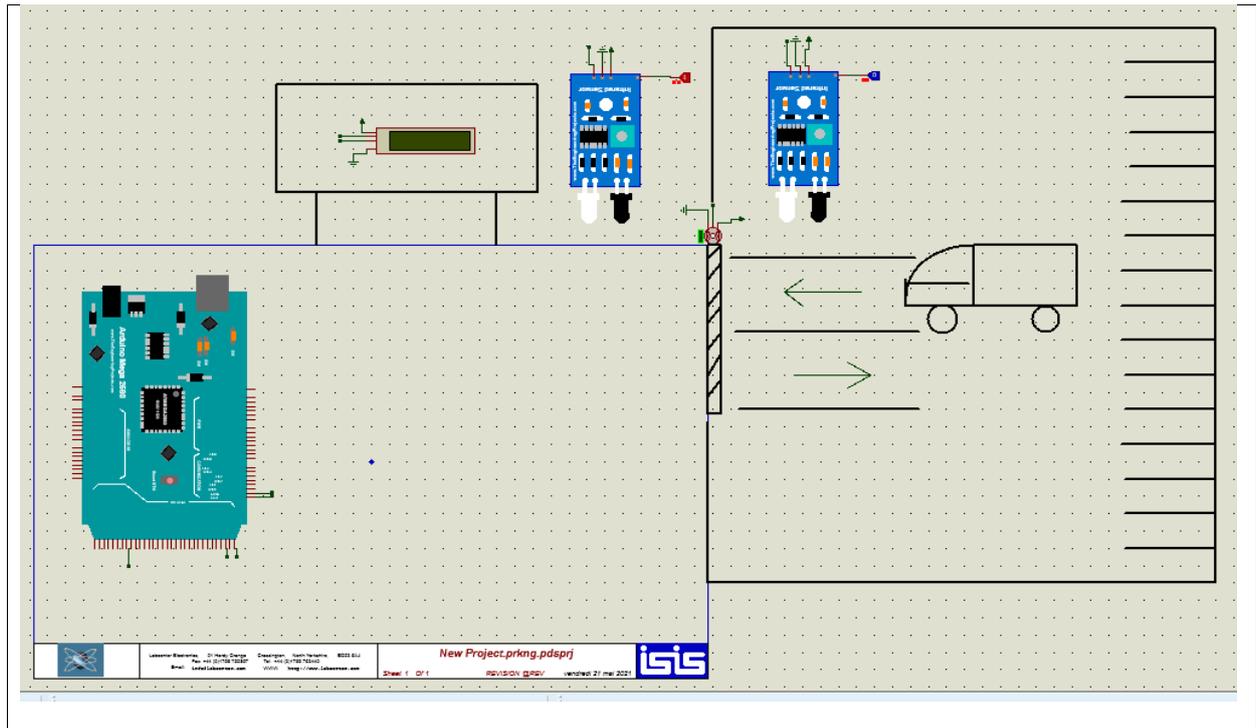


Figure III.4 : Schéma électronique du parking intelligent.

Pour tester le circuit ; on a monté les différents composants sur une plaque d’essai dont les connexions sont illustrées par la figure suivante et il a été utilisé cette fois ci le prototypage Fritzing à l’aide d’un Esp32.

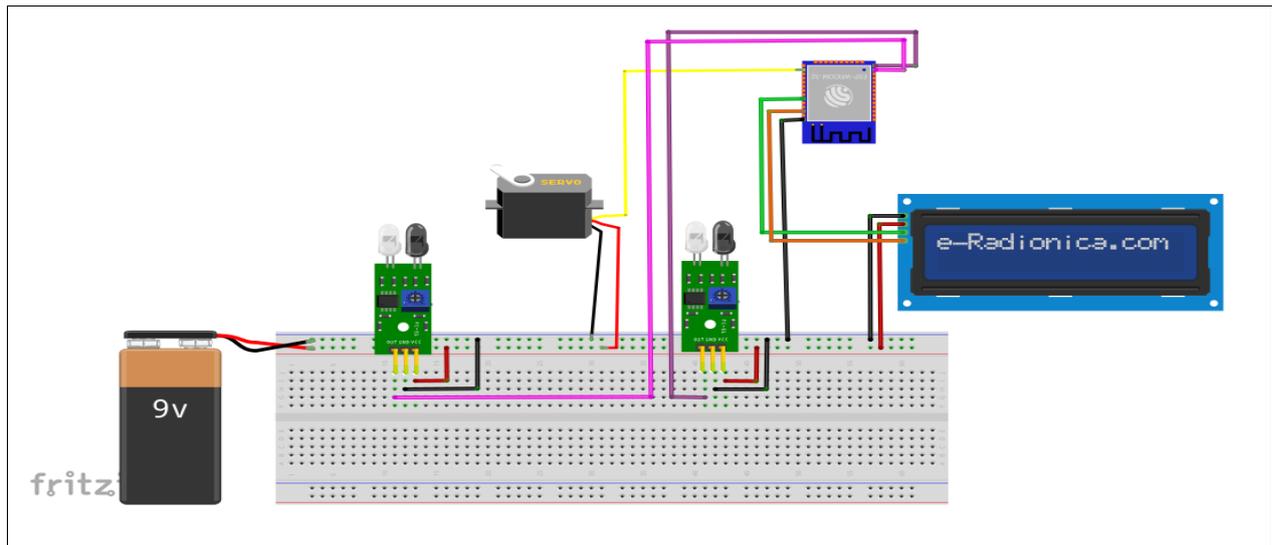


Figure III.5: Montage sur plaque d'essai (Breadboard).

e. Fonctionnement

Accès au parking

Le capteur infrarouge est situé à l'entrée du parking. Quand un véhicule est présent, la barrière s'ouvre automatiquement. Cette dite barrière se ferme directement après que le véhicule passe.

Le système de contrôle comptabilise chaque véhicule entrant et affiche le nombre de véhicules présents. Une fois que le parking est plein, aucun véhicule ne pourra accéder et un message sera affiché signalant que le parking est plein.

Pour permettre aux véhicules de sortir du Parking ; un second capteur est installé à la sortie.

Une fois que le véhicule a quitté le parking, le système de contrôle indique et affiche la disponibilité de la place libre.

f. Conclusion

Dans cette partie, nous avons exposé le fonctionnement du Parking intelligent et présenté sa description détaillée en donnant un synoptique et un schéma fonctionnel du système.

III.2.2. Arrosage automatique

a. Présentation

Compte tenu des imperfections qualifiantes les pratiques traditionnelles d'arrosage où la consommation d'eau s'avère irrationnelle, on a mis en œuvre une nouvelle modalité plus développée qui prend en considération l'humidité du sol à l'aide des capteurs. Notre système d'irrigation fonctionne comme suit : un capteur d'humidité est implanté dans le sol pour la mesure du taux d'humidité, le système d'arrosage comprend aussi un réservoir d'eau. Notons que l'arrosage est automatisé si le sol est humide.

Cette partie décrit la surveillance et l'arrosage du sol en utilisant deux (02) pompes, une émergée dans le réservoir pour l'arrosage et l'autre pour le remplissage du réservoir. Le contrôle du niveau d'eau de ce dernier est assuré par un capteur ultrason.

b. Schéma synoptique

Comme on va la montrer dans la figure ci-dessous, le schéma synoptique du système d'irrigation est constitué essentiellement de trois (03) parties et qui sont :

- Une carte renfermant une unité de mesure via des capteurs.
- Une carte de traitement et de contrôle.
- Un circuit renfermant l'actionneur muni d'indicateur.

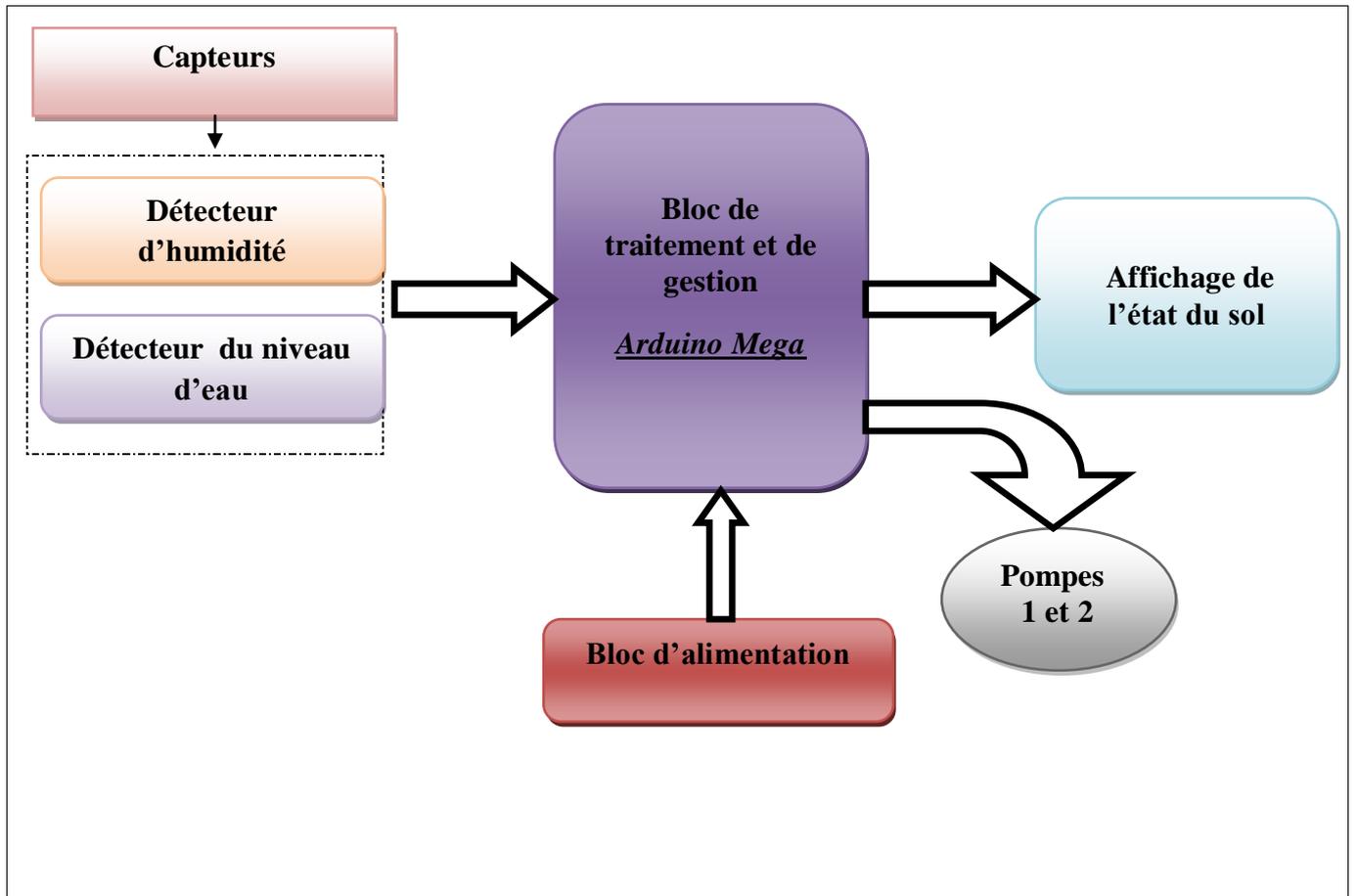


Figure III.6 : Schéma synoptique.

- L'unité de mesure comprend deux (02) types de capteurs à savoir : capteur d'humidité qui mesure le pourcentage d'humidité du sol, capteur ultrason qui permet de détecter l'existence de l'eau dans le réservoir avant le déclenchement de l'irrigation.
- L'unité de traitement est chargée de capter les informations délivrées par les différents capteurs. Il s'agit du microcontrôleur logé dans la carte Arduino MEGA, pour lequel le microcontrôleur assure la gestion de l'irrigation.
- L'actionneur piloté par un arduino selon le programme préétabli. Il s'agit de la commande de deux (02) pompes.
- L'indicateur de l'état du sol ; qui va détecter si le sol est sec ou bien humide.

c. Organigramme

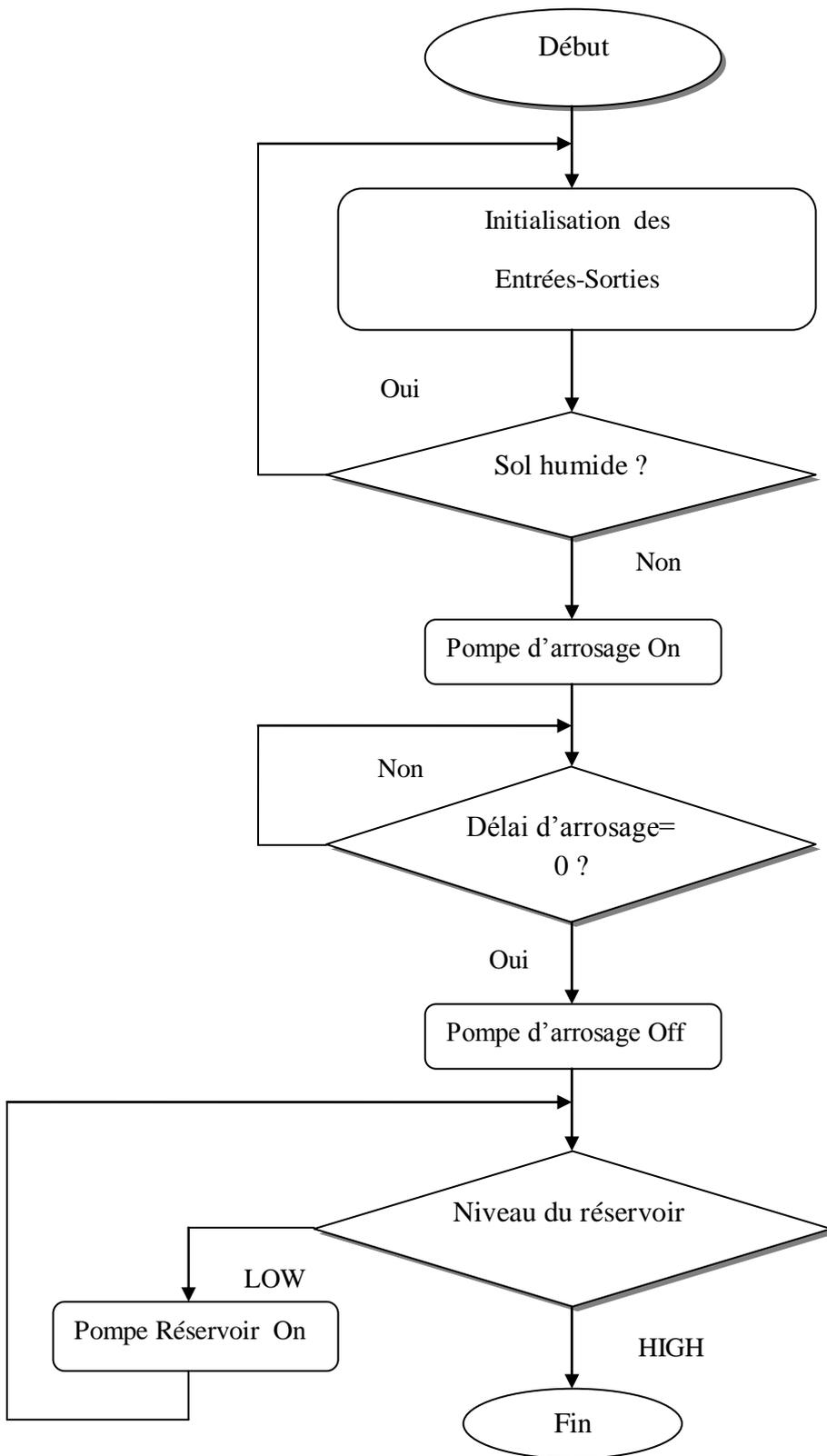


Figure III.7: Organigramme de fonctionnement.

d. Schéma électronique

La figure ci-dessous présente le schéma électronique d'arrosage automatique avec le logiciel Proteus :

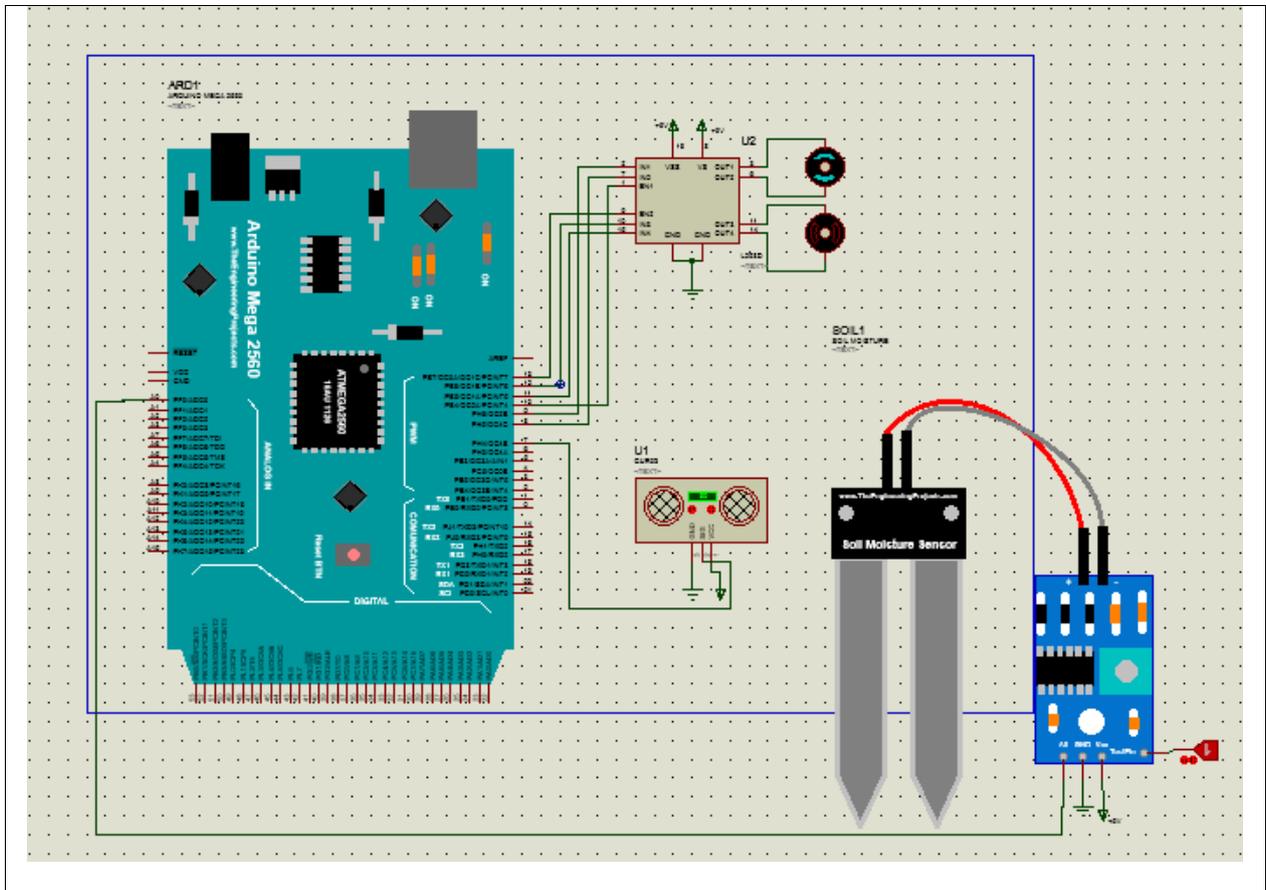


Figure III.8 : Schéma électronique d'arrosage automatique.

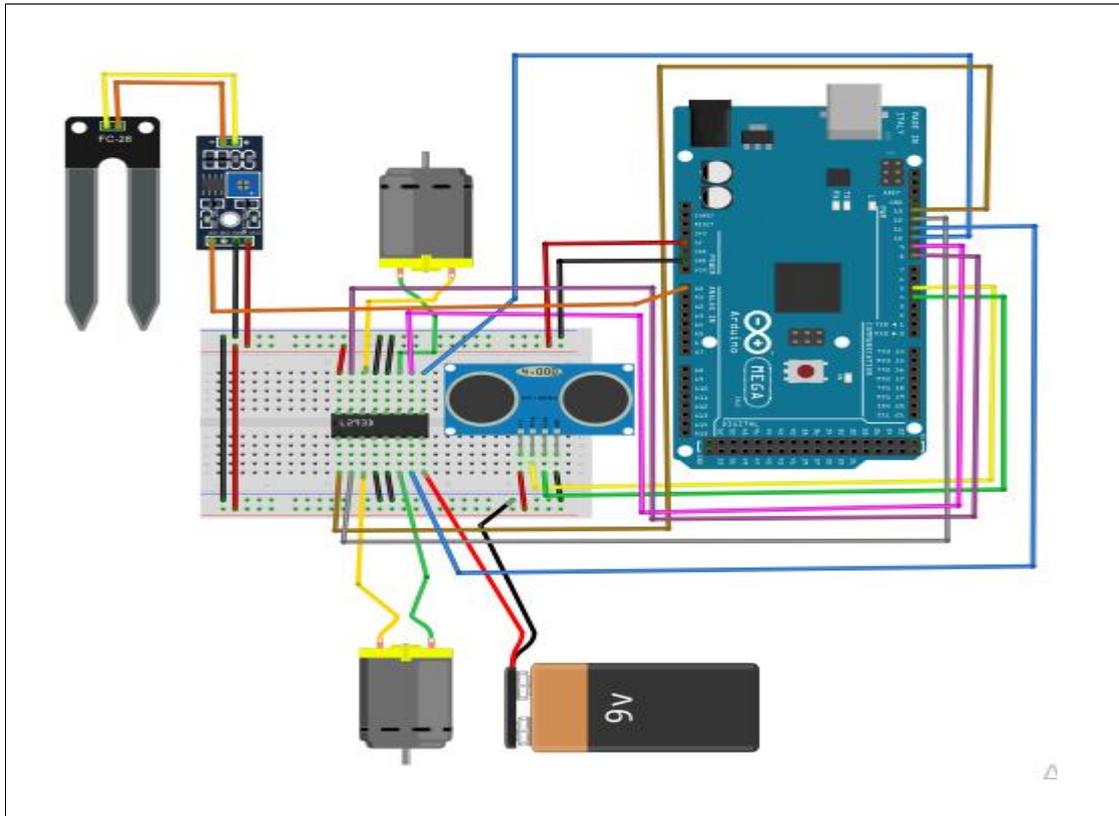


Figure III.9: Montage sur plaque d'essai (Breadboard).

e. Fonctionnement

Cette partie consiste à identifier les différentes entités intervenant dans notre système d'irrigation et qui sont constituées essentiellement d'un réservoir relié à une pompe émergée.

Cette pompe automatique assure le refoulement de l'eau vers le champ à arroser.

Un capteur d'humidité est implanté dans le sol lors de la détection d'un certain niveau d'humidité, le dispositif déclenche la pompe automatiquement.

Une fois le capteur ultrason détecte un manque d'eau dans le réservoir, le système déclenche la seconde pompe qui a pour rôle le remplissage de notre réservoir.

f. Conclusion

Dans cette partie, nous avons exposé le fonctionnement d'un système d'irrigation automatique et présenté sa description détaillée en donnant un synoptique et un schéma fonctionnel du système.

III.2.3 Eclairage public intelligent

a. Suiveur solaire

La dernière partie du projet porte sur l'étude et la réalisation d'un prototype du suiveur solaire à deux (02) axes: le panneau qui est orienté horizontalement et verticalement pour capter le maximum d'énergie solaire durant la journée et ce pour l'éclairage public. Ce système est commandé par une carte Arduino MEGA qui sert à manipuler les différents blocs du système à savoir :

* Capteur de lumière LDR

* Deux (02) Servomoteurs assurant l'orientation du panneau horizontalement et verticalement.

* LEDs utilisés pour la simulation de l'éclairage public.

Tracker double axe

Pour assurer un suivi complet de la course solaire (en azimut et en zénith), on utilise quatre capteurs LDR pour couvrir les changements (Nord-Sud) de la «déclinaison solaire» qui varie de $+23.45^\circ$ à -23.45° durant l'année.

Le principe est basé sur la différence d'éclairement des quatre LDR.

La figure représente l'organigramme de poursuite en double axe.

$V_{LDR_N} > V_{LDR_S}$, rotation du panneau PV vers le Nord. $V_{LDR_N} < V_{LDR_S}$, rotation du panneau PV vers le Sud. $V_{LDR_O} = V_{LDR_E}$ & $V_{LDR_N} = V_{LDR_S}$

Condition d'arrêt, le panneau est dans une position optimale.

VEst : Analaog input LDREst

VNord : Analaog input LDRNorth

VOst : Analaog input LDRWest

VSud: Analaog input LDRSouth

Vs : Tensin de seuil du panneau

V_{LDR_E}: Tension LDR Est

V_{LDR_N}: Tension LDR Nord

V_{LDR_O}: Tension LDR Ouest

V_{LDR_S}: Tension LDR Sud

S_EO : Servo Est_Ouest

b. chargeur solaire

Cette technique de charge est particulièrement intéressante car elle permet la transformation des rayons solaires (l'énergie lumineuse) en électricité (courant continu) sous l'effet photovoltaïque.

Exemple :

- **Principe de fonctionnement d'un chargeur de batterie automatique entre 14-15 volts avec un courant maximal de 3 ampères.**

Son fonctionnement est complètement automatique puisque il reste branché a une batterie, il entrera en fonction seulement si celle-ci est déchargée, alors qu'il se déconnectera automatiquement dès que la batterie sera chargée. Le dispositif est alimenté par un transformateur dont le secondaire est de 14-15 volts avec un courant de 3 Ampères minimum.

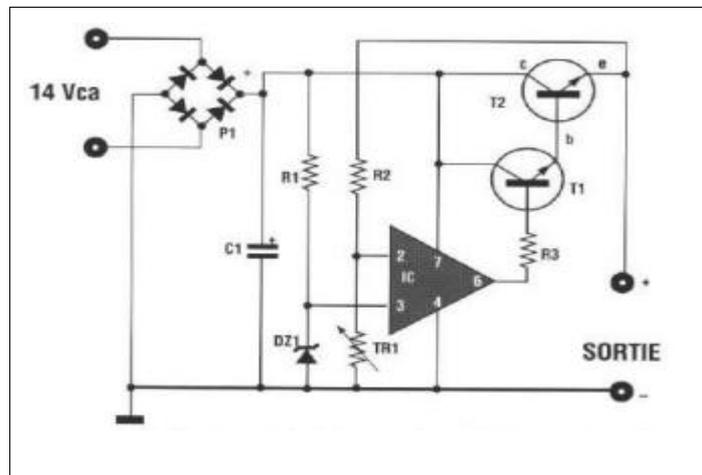


Figure III.10 : Chargeur de batterie automatique entre 14-15 volts avec un courant maximal de 3 ampères.

c. Convertisseur DC-DC

Pour exploiter la tension issue du panneau solaire, on doit faire une conversion DC-DC, plus un circuit de charge pour charger la batterie et cela pour alimenter notre système (éclairage public).

Le convertisseur continu- continu a pour fonction de fournir une tension continue variable à partir de tension continu fixe.

d. Eclairage public

A chaque détection de nuit, l'éclairage public sera mis en service systématiquement grâce à un dispositif de contrôle et ce en utilisant des capteurs adéquats.

Il faut noter que chaque poteau doit être équipé d'un petit panneau solaire et d'une batterie.

Cette dernière est chargée en permanence la journée par le panneau et le circuit de charge.

Les nouvelles technologies des lampes à LED permettent une consommation minimale et une très grande économie en énergie.

Afin d'augmenter la tension issue du panneau, un convertisseur DC- DC est indispensable.

e. Schéma synoptique

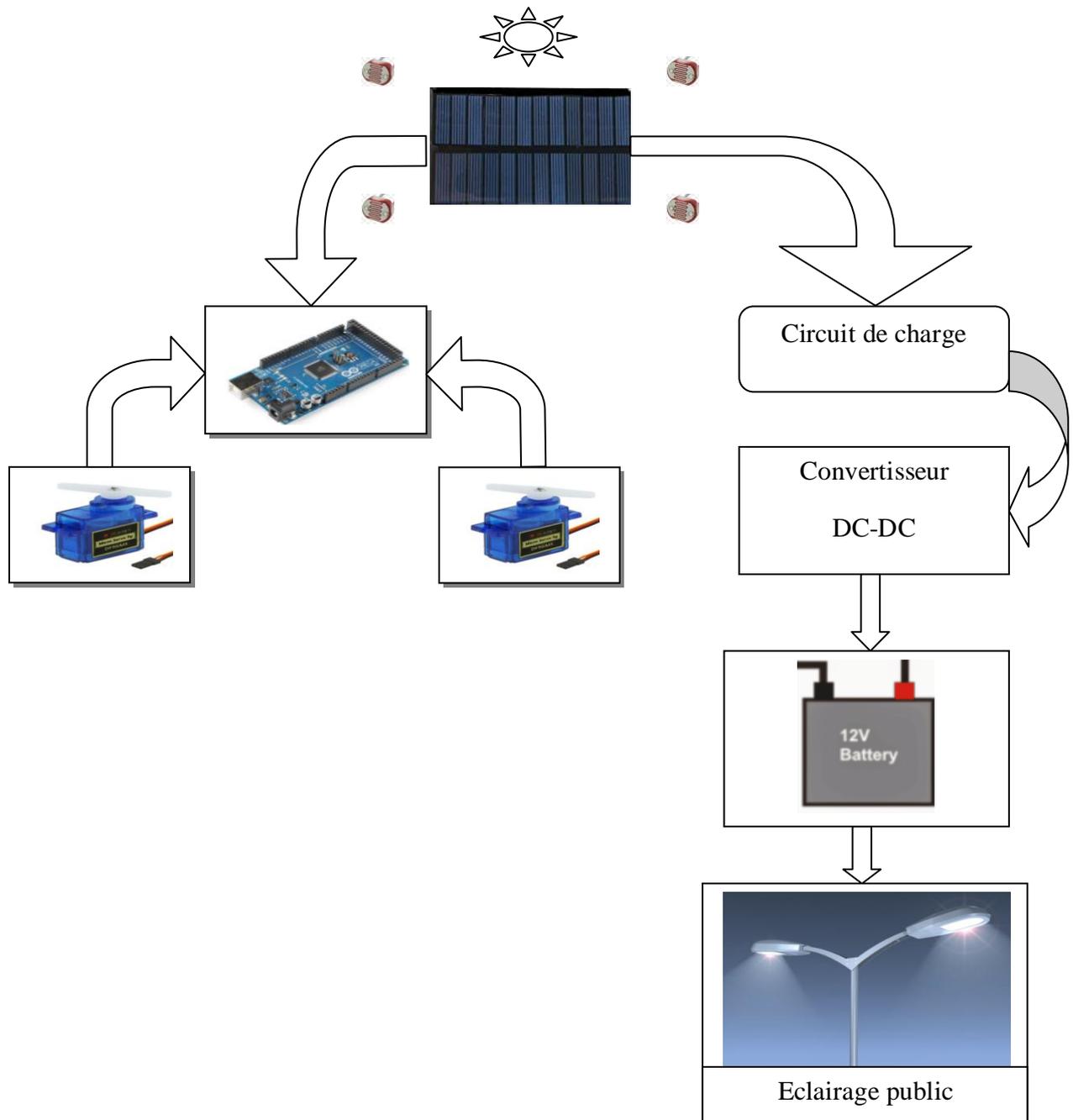


Figure III.11 : Schéma synoptique.

f. Organigramme de poursuite en double axe :

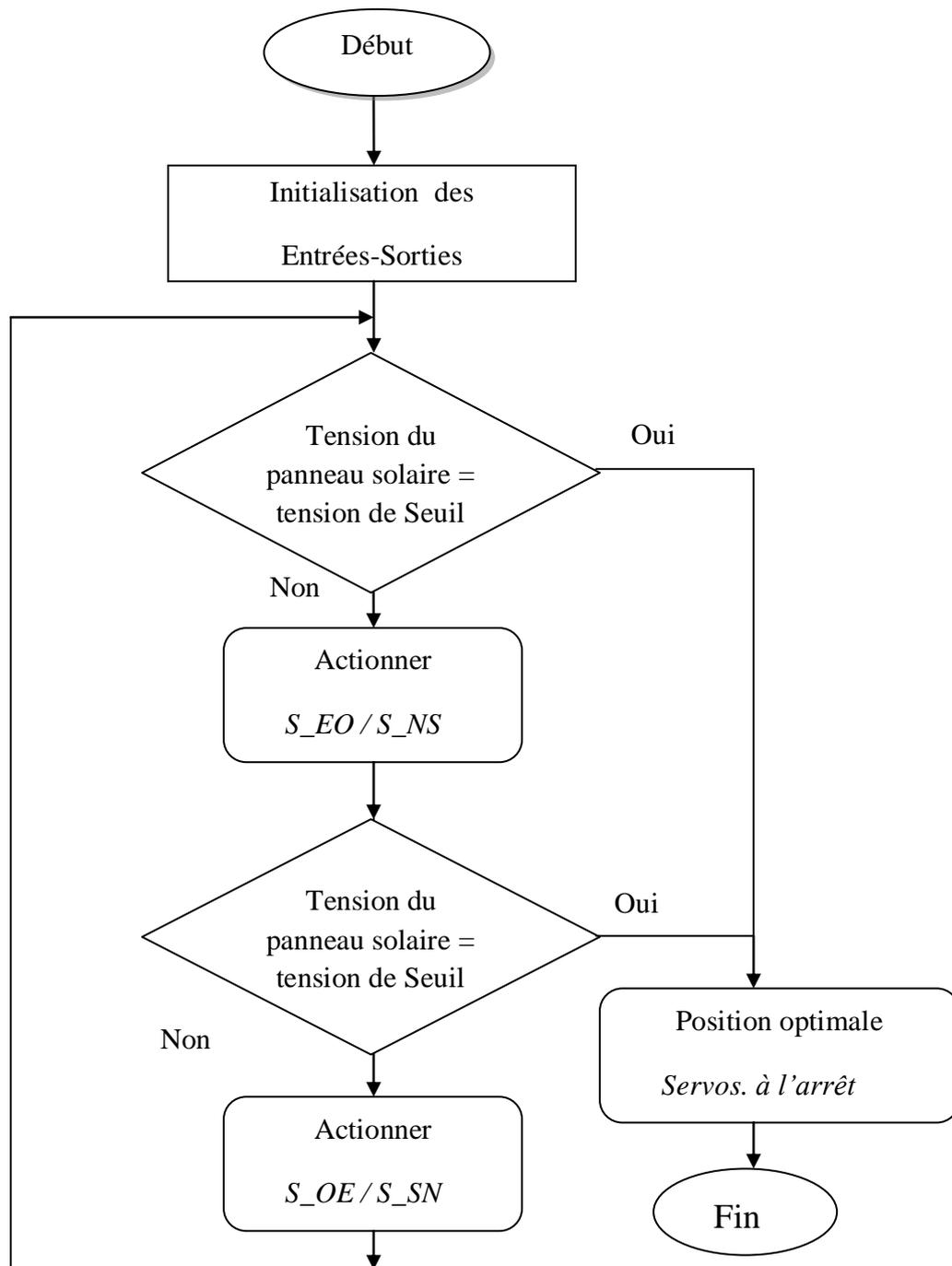


Figure III.12: Organigramme de fonctionnement.

La nécessité de déterminer avec exactitude les trajectoires empruntées par le soleil est primordiale, de manière à définir la meilleure façon avec laquelle sera effectué le suivi intelligent.

Un intérêt particulier a été porté à la technique de suivi active (double axe) ou asservi reposant sur le principe du repérage via des capteurs de lumière. C'est cette technique qu'on a adoptée et qui va être étudiée en vue d'une réalisation pratique.

g. Schéma électronique

1. Suiveur solaire

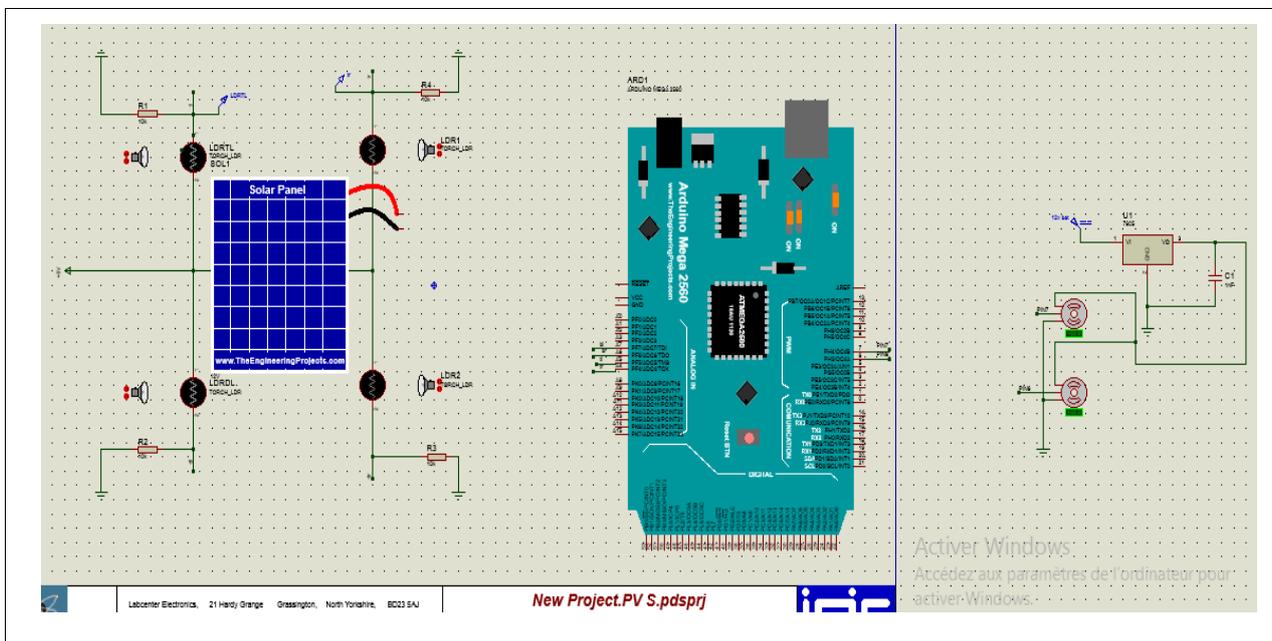


Figure III.13: Schéma électronique d'un suiveur solaire.

- **Fonctionnement**

Les quatre (04) capteurs (LDR) sont branchés aux entrées analogiques (A5, A7, A4, A6) de l'Arduino.

Et sont disposés d'une manière à ce que le panneau soit toujours bien orienté vers le soleil.

Le pilotage des deux(02) servomoteurs S_{EO} et S_{NS} dépend des tensions :

V_{LDR_E} , V_{LDR_N} , V_{LDR_O} et V_{LDR_S} aux entrées $VEst$, $VNord$, $VOst$ et $VSud$ du système de contrôle, ce qui provoque le mouvement du suiveur selon la position optimale du panneau photovoltaïque.

2. Chargeur de batterie 12V

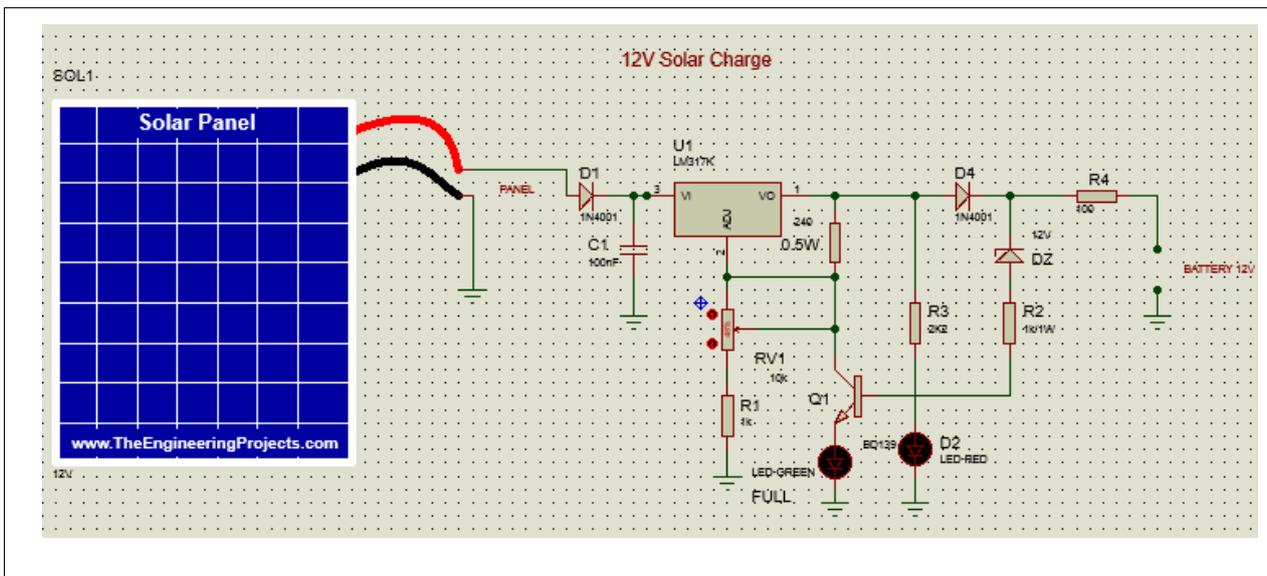


Figure III.14: Chargeur de batterie 12V.

- **Fonctionnement**

Ce circuit de chargeur de batterie automatique comprend principalement deux sections :

Une section d'alimentation et une section de comparaison de charge.

La tension d'alimentation principale normalement issue du panneau solaire est une tension continue qui doit être une tension supérieure à 12V (environ 14V).

La tension continue pulsée est appliquée à l'entrée d'un régulateur de tension variable le LM317 pour fournir une tension continue régulée.

La tension de sortie de ce régulateur de tension est variable de 1,2 V à 12 V et le courant de sortie maximal de ce circuit intégré est de 1,5 A. La tension de sortie de ce régulateur de tension est modifiée en faisant varier le pot 10k qui est connecté à la broche de réglage du LM317.

La sortie du régulateur de tension Lm317 est appliquée à la batterie via la diode D4 et la résistance R5. Ici, la diode D4 est utilisée pour éviter la décharge de la batterie en cas de panne de l'alimentation principale. Lorsque la batterie se charge complètement, la diode Zener DZ qui est connectée en polarisation inverse conduit. Maintenant, la base du transistor BD139 NPN reçoit le courant à travers la diode Zener afin que le courant total soit mis à la terre.

Dans ce circuit, la LED verte est utilisée pour indiquer la charge de la batterie. La résistance R3 est utilisée pour limiter le courant dans la LED verte.

- **Circuit de principe**

Si la tension de la batterie est inférieure à 12 V, le courant du LM317 IC traverse la résistance R4 et la diode D4 vers la batterie. À ce moment, la diode Zener ne conduit pas car la batterie prend tout le courant pour se charger.

Lorsque la tension de la batterie atteint 13,5 V, le flux de courant vers la batterie s'arrête et la diode Zener obtient la tension de claquage suffisante et permet au courant de la traverser.

Maintenant, la base du transistor reçoit le courant suffisant pour s'allumer afin que le courant de sortie du régulateur de tension LM317 soit mis à la terre via le transistor Q1. En conséquence, la LED rouge indique la pleine charge.

La tension de sortie du chargeur de batterie doit être inférieure à 1,5 fois celle de la batterie et le courant du chargeur doit être égal à 10 % du courant de la batterie. Le chargeur de batterie doit avoir une protection contre les surtensions, une protection contre les courts-circuits et une protection contre l'inversion de polarité.

3. Convertisseur DC-DC

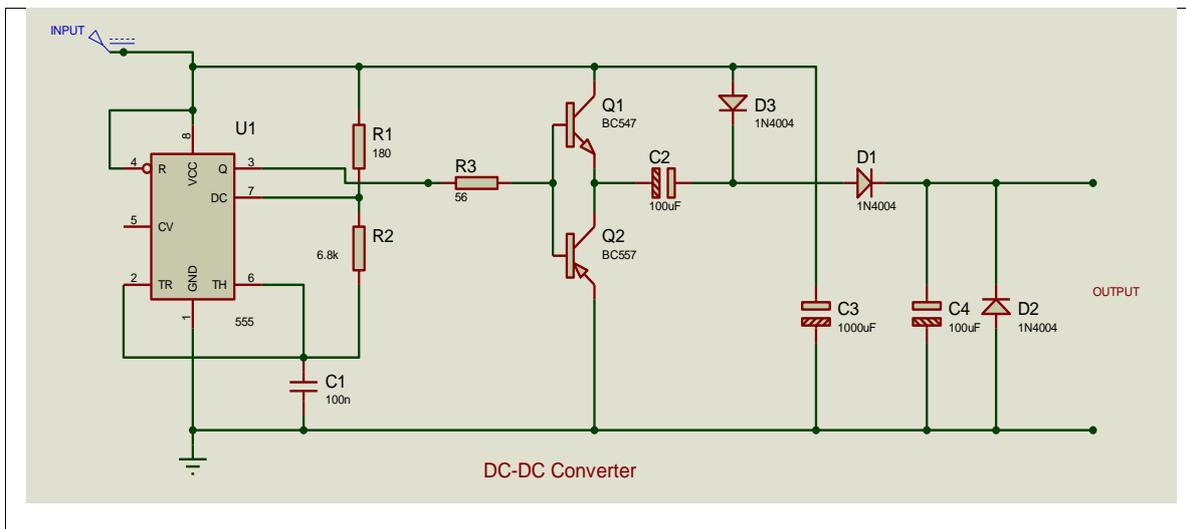


Figure III.15: Convertisseur DC-DC.

- **Fonctionnement**

Dans ce travail basé sur l'électronique de puissance, on essayera de réaliser un convertisseur DC-DC simple à l'aide de 555 Timer IC. La tension de sortie de ce circuit sera comprise entre 7,5 V et 35 V CC avec un courant de sortie de 60 mA.

Afin de garder le circuit convertisseur DC-DC utilisant 555 simple, nous n'avons utilisé aucune commande prédéfinie. Cela signifie que le niveau de la tension de sortie dépend de la charge. Théoriquement, le niveau de sortie est le double du niveau de la tension d'entrée. Mais en raison de certaines pertes dues aux transistors et aux diodes de redressement, un tel niveau n'est pas atteignable.

La chute à ces jonctions est constante de 0,6 V. Mais les pertes sont proportionnellement plus importantes avec une tension d'entrée de 6 V qu'avec une tension de 18 V.

Le 555 Timer IC configuré en mode multivibrateur génère un signal à une fréquence d'environ 10 kHz. Selon le niveau de sortie du CI, le transistor Q1 ou Q2 est activé. Il en résulte que le condensateur C2 est chargé pendant la demi-période. Ensuite, pendant l'autre demi-période, la charge du condensateur C2 est transférée au condensateur C3. Il en résulte une tension de sortie légèrement inférieure à deux fois la tension d'entrée.

Les diodes 1N4004 sont utilisées à cause du facteur de fréquence.

Le convertisseur consomme un courant de 5 mA plus deux fois le courant de sortie.

Simulation du circuit

La simulation pour le convertisseur DC-DC simple utilisant 555 Timer IC (7,5-35V) est effectuée à l'aide du logiciel Proteus et les résultats de sortie obtenus sont acceptables.

4. Eclairage public

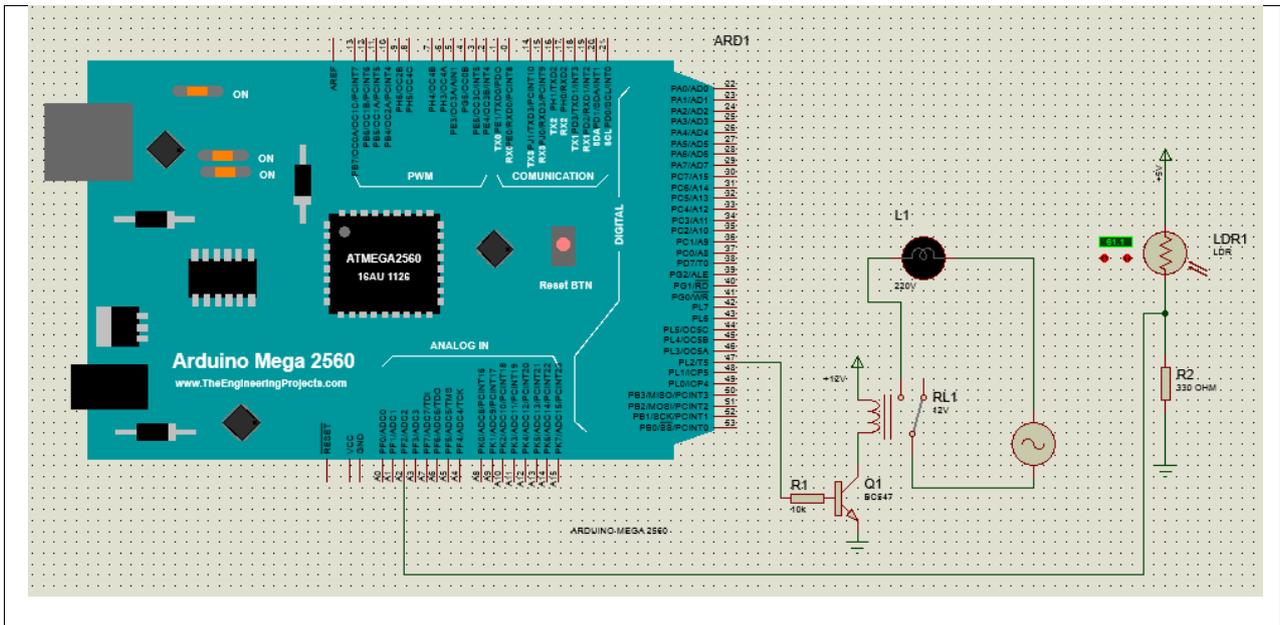


Figure III.16: Schéma électronique.

- **Fonctionnement**

Dans cette partie nous allons réaliser un système d'éclairage automatique avec Arduino. Il consiste à allumer une lampe LED lorsqu'il tombe la nuit, et l'éteindre dans le cas contraire. Il utilise principalement une photorésistance LDR.

i. Conclusion

Dans cette partie, nous avons réalisé un suiveur solaire à deux axes équipé d'un circuit de charge ainsi qu'un convertisseur DC-DC pour charger une batterie de 12v, et en donnant un synoptique et un schéma fonctionnel de chaque système.

III.3. Liaison WIFI

Afin de concrétiser la notion "ville intelligente" nous avons intégré l'Iot dans notre projet et cela par le biais de l'ESP32.

1. Les différents modes de fonctionnement de l'ESP32

L'ESP32 possède 2 modes WiFi possibles :

- **Le mode Station (STA) :** est utilisé pour connecter le module ESP32 à un point d'accès Wi-Fi. C'est le mode par défaut. L'ESP32 peut être utilisé comme :
 - **Un ordinateur :** qui serait connecté à notre routeur. Si ce dernier est relié à Internet, alors l'ESP32 peut accéder à Internet.
 - **Client :** c'est-à-dire faire des requêtes aux autres appareils connectés sur le réseau.
 - **Serveur :** c'est-à-dire que d'autres appareils connectés sur le réseau vont envoyer des requêtes à l'ESP32.

La figure suivante, présentée ci-dessous le mode station d'un ESP32:

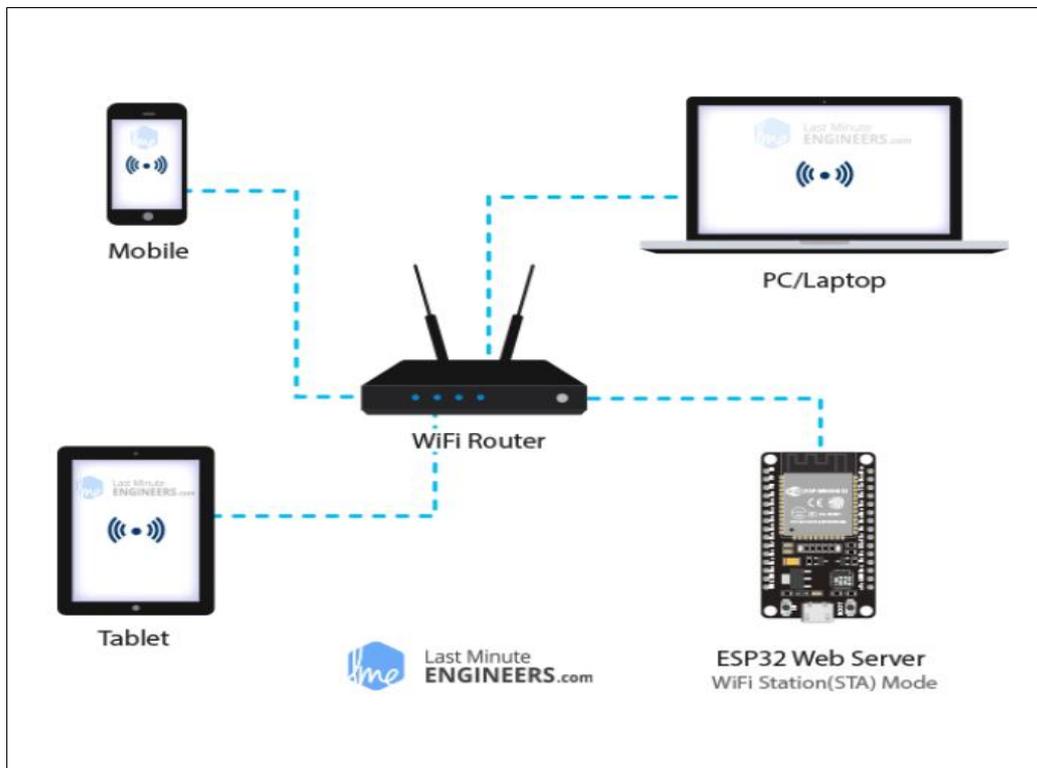


Figure III.17: le mode station d'un ESP32.

- **Point d'accès (Access Point)** : transforme l'ESP32 en point d'accès (AP), c'est à dire que d'autres appareils peuvent le connecter sur l'ESP32 (il donne une adresse IP à cinq clients au maximum). Ce mode de fonctionnement est aussi appelé point d'accès logiciel (soft-AP).



Figure III.18: le mode station d'un ESP32.

Note :

Dans ce mode l'ESP32 n'est pas connecté à internet. Et on doit créer une page HTML qui nous permette de saisir des paramètres de connexion à un réseau Wifi, ce mode est plus gourmand en calcul et en énergie (risque d'échauffement de la carte ESP32) puisque l'ESP32 doit simuler un routeur Wifi complet (Soft AP). La latence et le débit seront moins bon qu'avec un routeur classique.

2. Les applications de contrôle pour IoT

a. Application Web

Une application web est un logiciel applicatif d'où son nom, hébergé sur un serveur et accessible via les navigateurs Internet (Explorer, Mozilla Firefox, Chrome...). Contrairement à un logiciel classique, les applications web n'ont pas besoin d'être installées sur les ordinateurs de leurs utilisateurs, il leur suffit en effet de se connecter à l'application à l'aide de leurs navigateurs favoris pour pouvoir y accéder.

La création des applications web demande la maîtrise parfaite de différents langages de codage, comme le **HTML**, le **CSS**, **JavaScript**. En résumé, les applications web sont des pages vivantes qui réagissent à vos sollicitations et vous obéissent « au doigt et à l'oeil ». Cela nécessite donc deux (02) critères d'utilisation : simplicité et efficacité.

1. HTML

Le HTML (« HyperText Markup Language ») est un langage permettant de décrire les différents composants d'un document multimédia. Le HTML est tiré du XML, il est à la base de fonctionnement d'un site web. Il permet d'écrire et organiser le contenu de la page web (paragraphe, titres). Ainsi le langage HTML utilise des commandes (balises, marqueurs ou tags) pour le formatage des informations. Il est simple et compatible avec la plupart de navigateurs web. Nous avons adopté la version la plus récente qui est la version 5 « HTML5 » pour la conception de notre plateforme. Ceci nous offre beaucoup plus d'amélioration en termes de contenus, une meilleure gestion des formulaires.

2. CSS

Les feuilles de style en cascade ou CSS (« Cascading Style Sheets ») sont une extension de langage HTML. Les CSS se préoccupent de la mise en forme du contenu. Elles permettent :

- D'ajouter des propriétés à une commande HTML.
- Une mise à jour plus aisée des pages web d'un site
- Définir la typographie et l'agencement des pages web.

Actuellement nous sommes à la version 3 du CSS (CSS3) qui apporte beaucoup plus d'amélioration tant au niveau des bordures que des ombres.

3. JavaScript

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est -à-dire au niveau du navigateur et non du serveur Web. Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page Web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.

4. Les plateformes IoT Cloud

Sont des bases de données virtuelles en ligne qui stockent les informations de l'End-device et donne la visualisation de ces informations (tableaux, graphiques) pour les utilisateurs finaux.

b. Application mobile

Android est un logiciel de type système d'exploitation mobile, comme « Windows » est un système d'exploitation sur PC, c'est un système d'exploitation ouvert dont le code source est librement accessible ce qui permet à n'importe quel fabricant de l'intégrer dans son système gratuitement.

Android s'appuie sur deux piliers :

- le langage Java
- Le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur.

- **Application Blynk**

Blynk a été conçu pour l'Internet des objets. C'est une plate-forme avec des applications iOS et Android pour contrôler Arduino, Raspberry Pi, ESP32 et autres sur Internet. Il peut également :

-Contrôler le matériel à distance,

-Afficher les données des capteurs,

-Stocker des données, les visualiser et faire beaucoup d'autres choses intéressantes.

La plate-forme comprend trois (03) composants principaux :

A. Blynk App : nous permet de créer des interfaces incroyables pour notre projet en utilisant les divers widgets que nous fournissons.

B. Blynk Server : responsable de toutes les communications entre le smartphone et le matériel. Nous pouvons utiliser notre Blynk Cloud ou exécuter notre serveur Blynk privé localement. Il est open-source, pourrait facilement gérer des milliers d'appareils et peut même être lancé sur un Raspberry Pi.

C. Les bibliothèques Blynk : pour toutes les plates-formes matérielles populaires – permettent la communication avec le serveur et traitent toutes les commandes entrantes et sortantes.

Fonctionnalités

- API et UI similaire pour tous les hardwares et périphériques supportés
- Connexion au cloud via:
 - Ethernet
 - WiFi
 - Bluetooth et BLE
 - USB (Serial)
 - ...
- 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
- Surveillance de l'historique des données via le widget **History Graph**
- Communication Périphérique-à-Périphérique en utilisant le widget **Bridge**
- Envoi d'emails, de tweets, de notifications push, etc.
- ... de nouvelles fonctionnalités sont constamment ajoutées !

3. Configuration de l'application Blynk pour ESP32

Après avoir téléchargé l'application Blynk sur un Smartphone, nous avons besoin de suivre les étapes suivantes:

1. Créer un compte où on peut utiliser notre compte Gmail actuel.
2. Après la création du compte, une fenêtre s'ouvrira et on clique sur **Nouveau projet**.

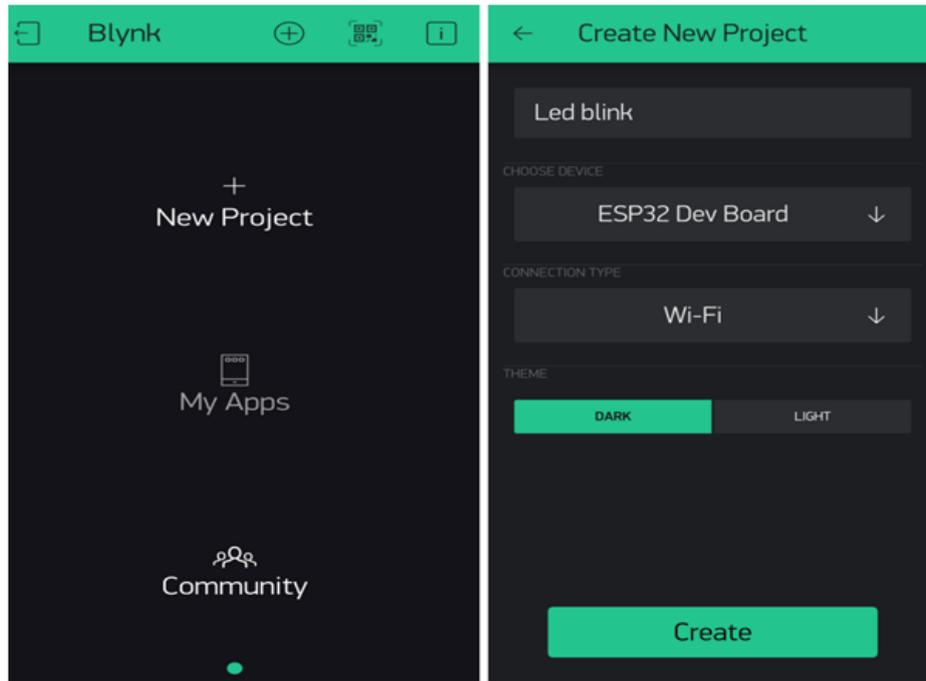


Figure III.19: présentation de l'application Blynk.

3. On donne un nom en fonction de notre choix et dans l'appareil, on choisit **ESP32 Dev Board** et dans le type de **connexion**, on choisit **wifi**, puis on clique sur Créer.
4. Une fenêtre apparaîtra qui montre notre **jeton d'authentification** qui sera sauvegardé par nos soins dès la réception de messagerie pour des éventuelles utilisations plus tard. Donc ; nous pouvons ouvrir notre e-mail pour vérifier la clé d'authentification et après on ajoute les widgets dont on a besoin.

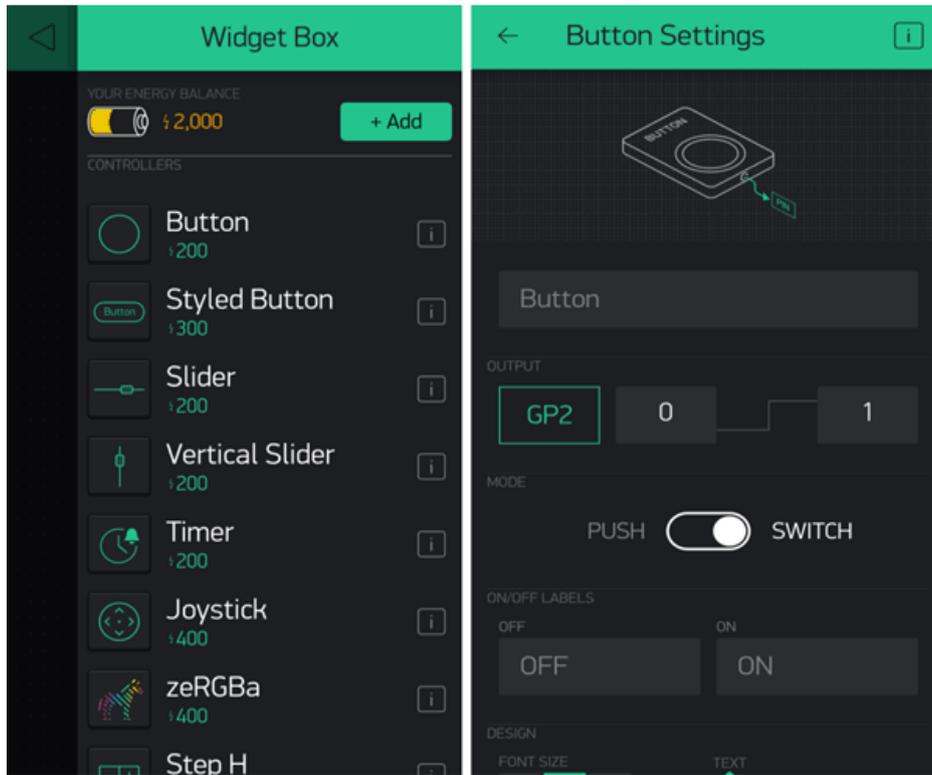


Figure III.20: présentation de l'interface de l'application Blynk.

3.1. L'utilisation d'application Blynk pour le Smart Parking

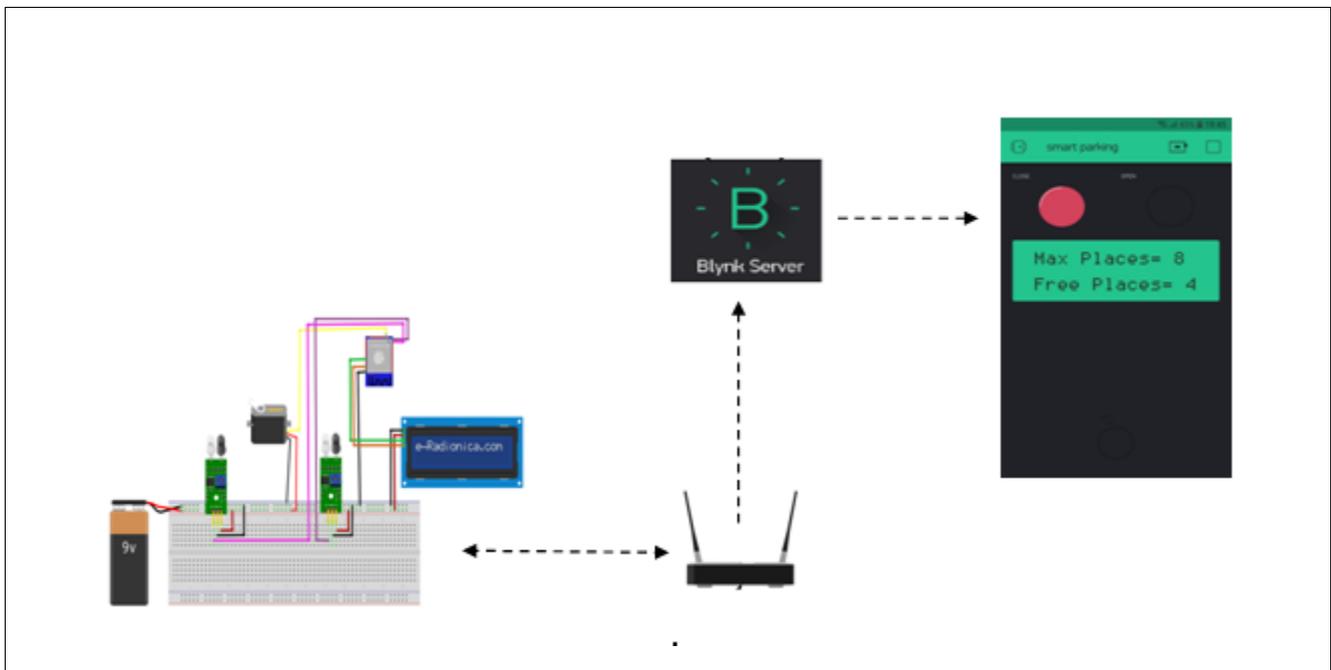
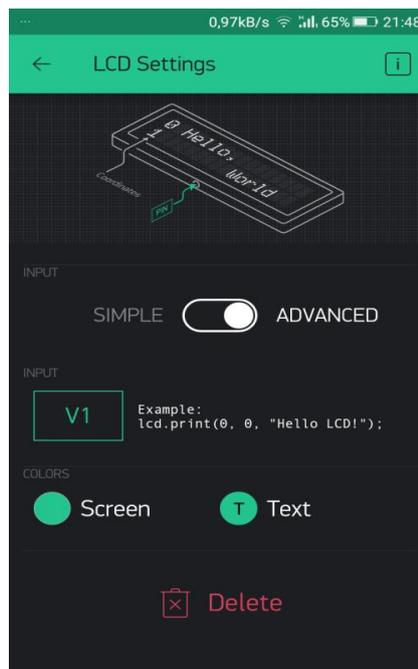


Figure III.21: Application Blynk : Parking avec l'ESP32.

L'objectif final d'une solution de stationnement intelligente intégrée à une ville intelligente est de fournir aux citoyens les outils nécessaires pour trouver facilement et rapidement des moyens d'améliorer leur vie grâce à l'utilisation de technologies abordables. Dans ce cas précis, l'objectif est de fournir aux utilisateurs finaux un outil simple pour accéder à la fente de stationnement la plus pratique disponible à un moment donné. En raison de l'utilisation croissante des technologies mobiles, la meilleure façon d'apporter la solution à l'utilisateur final est par le développement d'une application mobile. Sur cette partie ; on va procéder à l'exposition de la création de l'application Blynk et on va vous démontrer comment l'intégrer dans le parking intelligent où les étapes de travail par ordre se concrétisent dans les points suivants :

- Notre compte déjà créé avec l'application Blynk sera utilisé actuellement pour ouvrir un nouveau projet qui sera nommé : Smart Parking puis on va sélectionner un hardware (ESP32 Dev Board) ; on clique sur Create.
- Un message d'authentification sera reçu dans notre E-mail qui nous permettra de l'utiliser lors de la programmation par Arduino IDE.
- On rentre dans les widgets, on ajoute un widget de LCD.
- On accède aux paramètres de widget.
- On définit le code PIN sur la broche virtuelle (V1).
- On règle le mode sur Advanced.



Remarque :

Il existe deux façons d'activer et désactiver les choses avec Blynk :

1. Simple : Contrôle direct des broches d'entrées / sorties numérique sans codage.

2. Avancé : Utilisant des épingle virtuelles et un code simple.

- Les broches virtuelles ont été conçues pour permettre d'échanger des données entre la carte embarquée et l'application Blynk. Elle fonctionne comme un canal de communication.
- Le paramètre le plus important est PIN. La liste des broches reflète les broches physiques définies par notre hardware.

3.2. L'utilisation d'application Blynk pour l'arrosage automatique

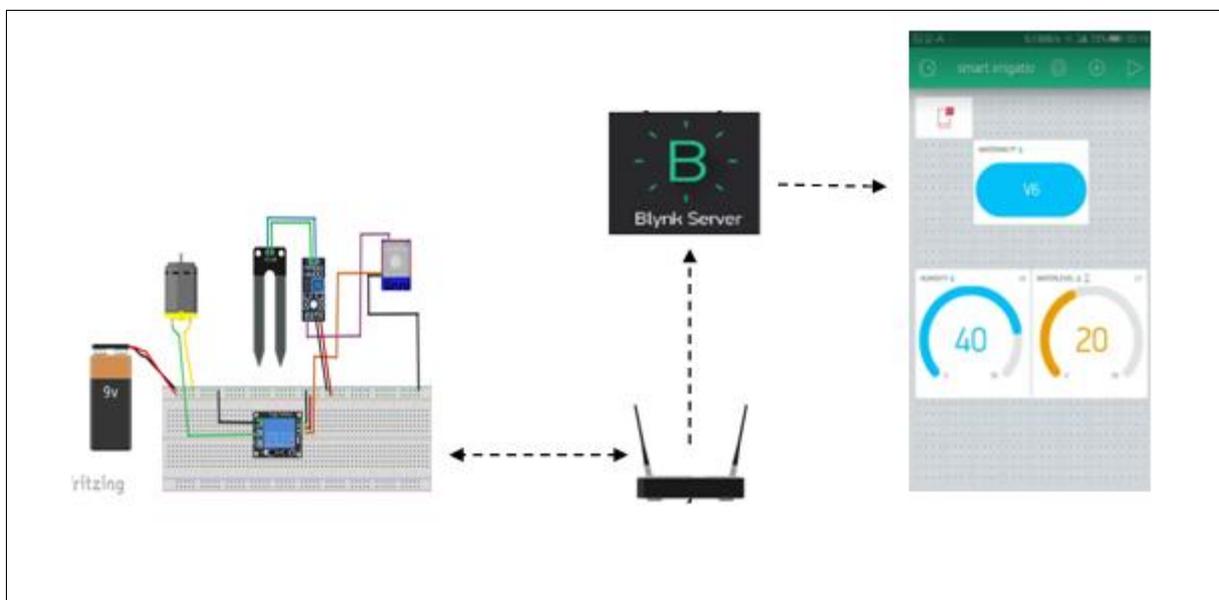


Figure III.22: Application Blynk : Système d'irrigation avec l'ESP32.

Le jardinage est un passe-temps très gratifiant. Cependant, en raison de nos modes de vie trépidants, les plantes de nos jardins peuvent ne pas toujours recevoir l'attention qu'elles méritent. Et si nous pouvions surveiller à distance leur état et leur apporter les soins dont ils ont besoin ? Il y a un moyen ! Dans cette partie du projet, nous allons créer un système de jardinage intelligent basé sur l'IoT à l'aide de l'application Blynk.

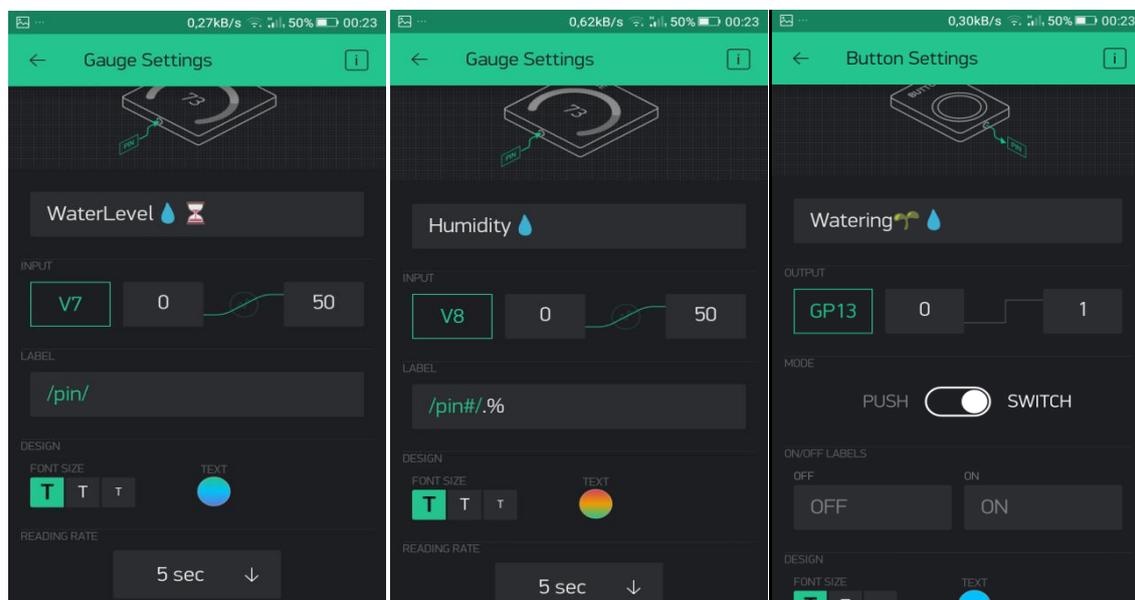
Ce projet de jardin intelligent est un système de surveillance de l'environnement des plantes qui surveille le niveau d'humidité du sol et le niveau d'eau. Ensuite, selon les conditions, le

système permet aux utilisateurs de livrer ce qui est nécessaire à distance via une application mobile. De plus, ce système peut être mis à l'échelle et étendu pour automatiser l'ensemble du processus de jardinage !

Dans cette configuration, nous surveillons les données du capteur et interfaçons un module de relais pour contrôler notre pompe. Grâce à l'application Blynk, nous pouvons surveiller à distance divers paramètres de notre jardin et activer les appareils selon les besoins.

Sur cette partie ; on va procéder à l'exposition de la création de l'application Blynk pour l'irrigation automatique où les étapes de travail par ordre se concrétisent dans les points suivants :

- Notre compte déjà créé avec l'application Blynk sera utilisé actuellement pour ouvrir un nouveau projet qui sera nommé : smart irrigation puis on va sélectionner un hardware (ESP32 Dev Board) ; on clique sur Create.
- Un message d'authentification sera reçu dans notre E-mail qui nous permettra de l'utiliser lors de la programmation par Arduino IDE.
- nous avons maintenant un canevas vide. Nous devons ajouter trois widgets de jauge pour surveiller le niveau d'eau, le niveau d'humidité du sol et un bouton pour contrôler le relais.
- On modifie les paramètres de chaque widget.



L'objet minuterie Blynk envoie des relevés d'humidité de l'eau et du sol toutes les cinq secondes. Ces données seront mises à jour dans les jauges de l'application Blynk.

III.4. Essais et résultats

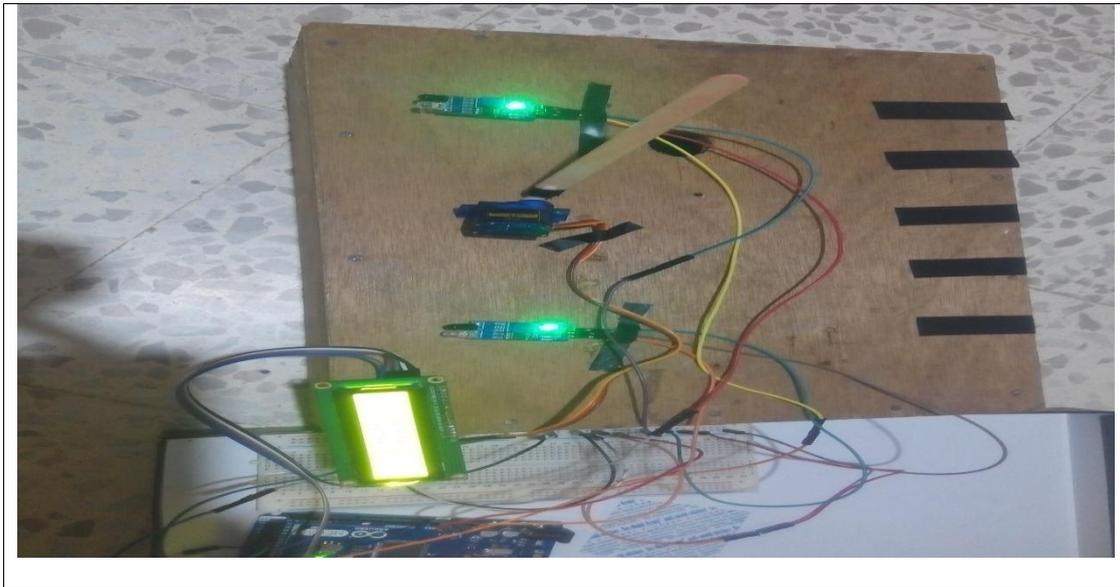
1. Parking intelligent

a. Liaison direct

- Entrée du Parking :

Détection de présence de véhicule :

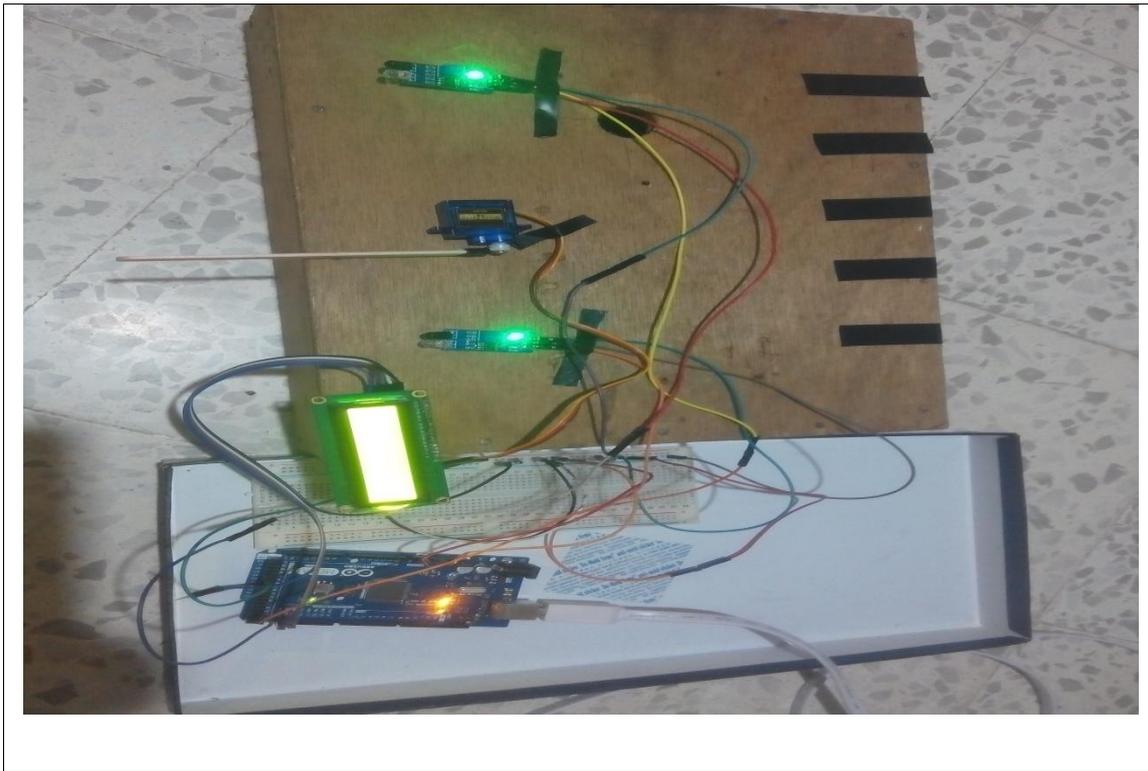
- Ouverture de la barrière.
- Décrémentation du nombre de places.



- Sortie du Parking :

Détection de présence de véhicule :

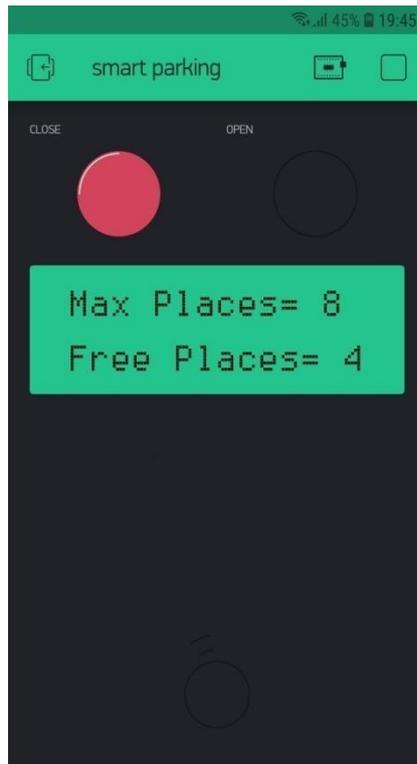
- Ouverture de la barrière.
- incrémentation du nombre de places.



b. Liaison wifi

Durant la journée, le système informatique contrôle en permanence le nombre d'entrée et de sorties de véhicules qui seront comptabilisés par la suite dans l'application Blynk sur le Smartphone.

Dans le cas où le Parking est saturé, un autre message sera envoyé à cette application, indiquant la non disponibilité de places dans le parking.

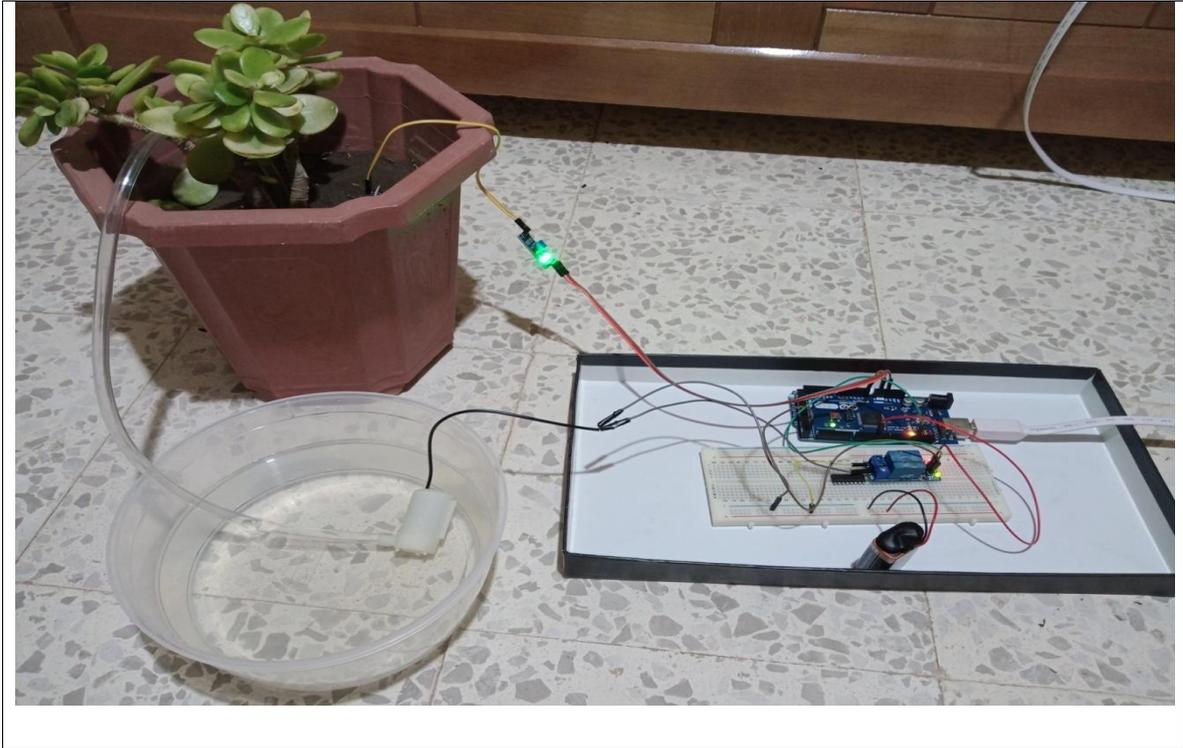


2. Irrigation automatique

a. Liaison direct

Un capteur d'humidité est implanté dans le sol lors de la détection d'un certain niveau d'humidité, le dispositif déclenche la pompe automatiquement.

Une fois le capteur ultrason détecte un manque d'eau dans le réservoir, le système déclenche la seconde pompe qui a pour rôle le remplissage de notre réservoir.

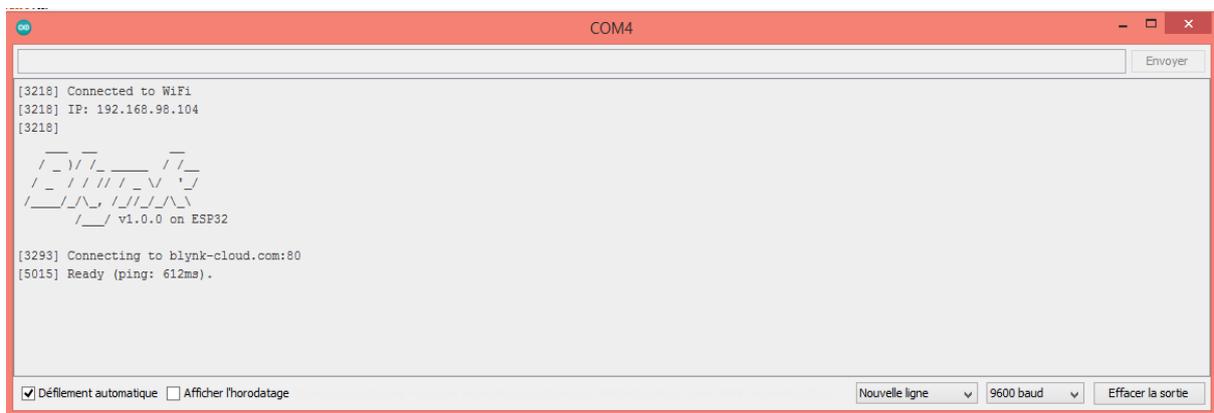
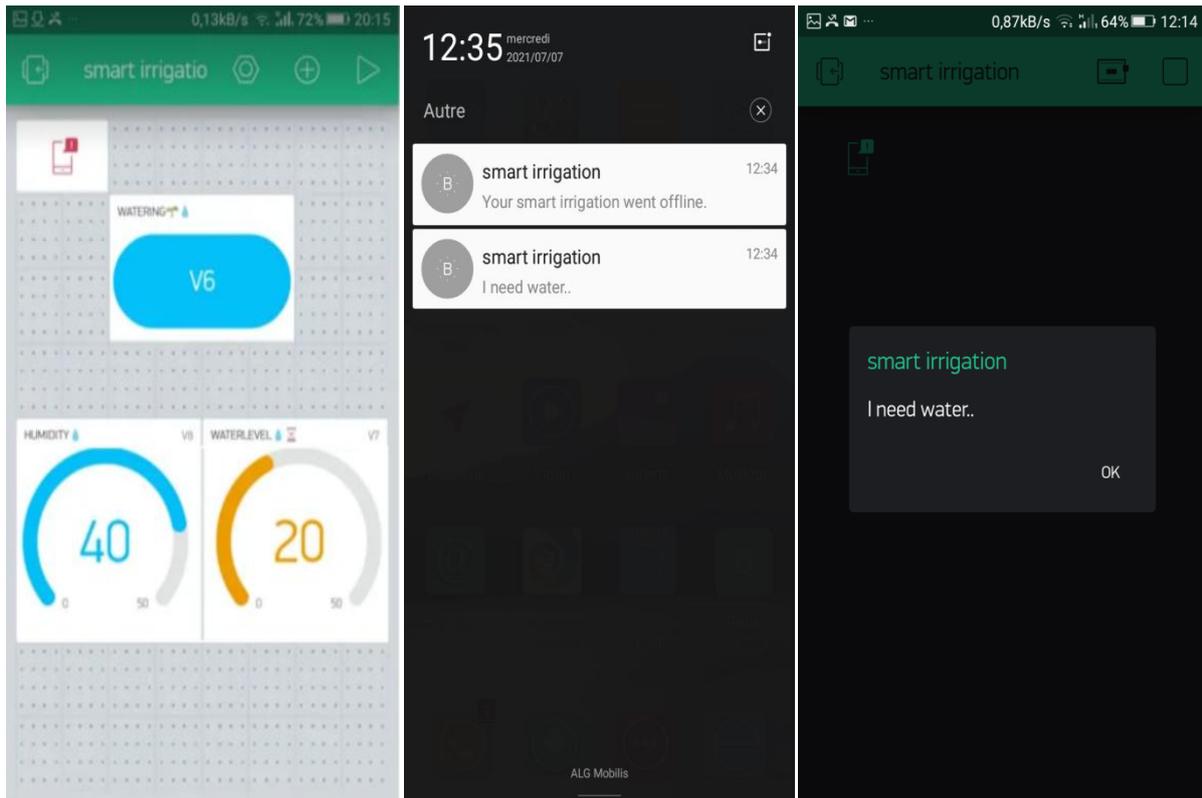


b. Liaison wifi

Si le niveau d'eau dans le réservoir est minimum, nous pouvons commander notre pompe externe d'eau à distance pour remplir le réservoir.

Maintenant, si le sol est sec et a besoin d'eau notre système donne l'ordre pour commander la pompe émergée dans le réservoir afin de donner la quantité d'eau suffisante pour arroser le sol.

On reçoit aussi une notification sur notre Smartphone quand le sol a besoin d'eau.



3. Eclairage public intelligent

Pour notre système d'éclairage public, nous avons adopté une application par liaison direct.

En présence de lumière de jour la batterie de 12v est chargée par le panneau photovoltaïque, dans le cas où la tension du panneau est inférieure à la tension de seuil le convertisseur DC-DC fait augmenter cette tension pour arriver à la tension voulue.

A la tombé de la nuit, l'éclairage public sera assuré par la batterie chargée.



```
COM3
|
| Envoyer
Analog reading = 17 - Sombre
Analog reading = 16 - Sombre
Analog reading = 356 - Peu de Lumineux
Analog reading = 85 - Sombre
Analog reading = 709 - Tres lumineux
Analog reading = 55 - Sombre
Analog reading = 665 - Tres lumineux
Analog reading = 753 - Tres lumineux
Analog reading = 818 - Tres lumineux
Analog reading = 10 - Sombre
Analog reading = 585 - Lumineux
Analog reading = 724 - Tres lumineux
Analog reading = 681 - Tres lumineux
Analog reading = 727 - Tres lumineux
Analog reading = 9 - Noir
Analog reading = 28 - Sombre
Analog reading = 12 - Sombre
Analog reading = 11 - Sombre
 Défilement automatique  Afficher l'horodatage
Nouvelle ligne 9600 baud Effacer la sortie
```

III.5. Problèmes rencontrés durant notre travail :

Lors de notre travail sur ce système, il a été rencontré par nos soins quelques difficultés et qui sont les suivantes :

- LCD I2C n'a pas pu fonctionner directement malgré que le montage a été fait normalement et le programme de fonctionnement étaient correcte , après des multiples recherches faites pour résoudre cet obstacle , il a été détecté que l'afficheur LCD I2C a deux broches de

contrôle qui doivent être reliées uniquement avec les PIN consécutivement D20 et D21 avec Arduino Mega à défaut, le LCD I2C ne pourra pas fonctionner.

- Lorsqu'on a voulu télécharger un nouveau sketch sur notre ESP32 ; on a obtenu ce message d'erreur « Une erreur fatale s'est produite : Échec de la connexion à l'ESP32 : Délai dépassé... Connexion... ». Cela signifie que notre ESP32 n'est pas en mode flash/téléchargement.

Après avoir sélectionné le bon nom de carte et COM port, on a procédé comme suit :

- On doit Maintenir enfoncé le bouton « **BOOT** » de notre carte ESP32

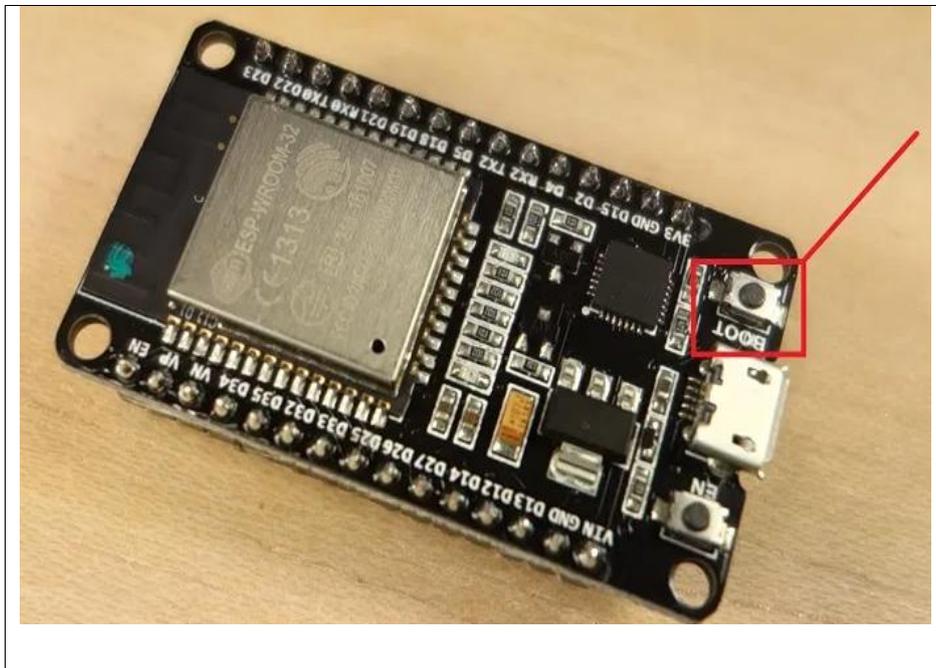


Figure III.23: problématique rencontrée sur le fonctionnement de l'Esp32.

- Nous Appuyons sur le bouton « **Télécharger** » dans l'IDE Arduino pour télécharger notre croquis :



- Après avoir vu le message « Connexion.... » dans notre IDE Arduino, on relâche le doigt du bouton " **BOOT** " :
- Après cela, nous devons voir le message « **Téléchargement terminé** »

- C'est ça. notre ESP32 devrait avoir le nouveau sketch en cours d'exécution. On appui sur le bouton « **ACTIVER** » pour redémarrer l'ESP32 et exécuter la nouvelle esquisse téléchargée.
- Nous devons également répéter cette séquence de boutons chaque fois que nous souhaitons télécharger un nouveau croquis.

III.6. Conclusion :

La phase de la réalisation est l'étape la plus importante. Dans ce contexte, nous avons présenté la deuxième partie de la pratique de notre projet par une brève description des applications et les utilisations de l'IOT sur la plateforme Blynk.

Conclusion générale :

Notre travail consiste à montrer les tendances récentes de la ville intelligente, ce « nouveau » modèle de ville est répandu dans le monde entier et qui investit dans son capital humain et social ainsi que dans ses infrastructures de communication traditionnelles (transport) et modernes (TIC), dans le but d'améliorer la qualité de vie de ses habitants et de dynamiser sa croissance économique, tout en ayant une bonne gestion de ses ressources et en utilisant des moyens encourageant la participation citoyenne dans la création de la ville et en mettant en place des infrastructures numériques (réseaux, objets connectés, données).

L'IOT, qui est une nouvelle technologie innovante et très pertinente, et un pilier des innovations en secteur de TIC et surtout dans le champ desancements des Smart City, qui aide le pays à digitaliser ses infrastructures et même les citoyens qui peuvent avoir une vie plus simple, rapide et efficace et nous permet de gagner beaucoup de temps et d'énergie.

C'est dans ce contexte que ce projet a été construit, et qui atteint avec succès et les résultats des différents tests sont très concluants. Ce projet nous a permis d'approfondir nos connaissances dans la réalisation pratique de circuits électroniques, la programmation de microcontrôleurs et à apprendre les différentes technologies employées dans la conception des villes intelligentes ou connectées. Ainsi que nous avons acquis une bonne maîtrise pour la création d'une application Android avec l'environnement *Blynk App* après avoir montré les éléments de contrôle de notre réalisation à distance ou par internet avec l'utilisation du module Wifi esp32.

Les perspectives entre dans cette nouvelle ère et apportera des réponses en termes d'innovation dans les relations entre habitants, villes, territoires et technologies.

Bibliographie

Sites WEB

[1*] <https://ovci-ulaval.weebly.com/quest-ce-quune-ville-intelligente.html>

[2] <http://www.smartcity.fr/europe/presentation>

[3*] http://www.applis.univtours.fr/scd/EPU_DA/2016PFE_Torres_Helene.pdf

Ouvrages et documents

[1] Caragliu, A., & Del Bo, C. (2012). Smartness and European urban performance: assessing the local impacts of smart urban attributes. *Innovation: The European Journal of Social Science Research*, 25(2), 97-113.

[3] LAHOULOU, MERABET & MERDJANA. 2017. *LA VILLE INTELLIGENTE : CONDITIONS, METHODES ET OUTILS D'UNE TRANSITION*. . master 2 mémoire, Université Mohamed Seddik BENYAHIA – Jijel Faculté des Sciences et de la Technologie

[4] COULÉE, C. & DIEUZEIDE, M. 2018. *Smart cities : quelles sont les principales menaces*. Master 120 crédits en ingénieur de gestion, à finalité spécialisée, Université catholique de Louvain

[5] CENTRE OF REGIONAL SCIENCE, V. U. 2007. *Smart cities :Ranking of European medium-sized cities*

[6] LAVATINE.A 2018. *SMART CITY INSTITUTE: STRATÉGIES ET PROJETS SMART CITY EN BELGIQUE onepoint.x weave*

[7] DELFT, T. 2019. Le projet City-zen: <http://www.cityzen-smartcity.eu/fr/objectifs/> .

[8] RAFAI, A. 2018. *Développement d'un système d'IoT (Internet of Things) pour le Smart Lighting sous la Plateforme BIM*. MASTER EN INFORMATIQUE, Université Larbi Ben M'hidi de Oum El Bouaghi

[9] BREUX, S. & DIAZ, J. 2017. *LA VILLE INTELLIGENTE Origine, définitions, forces et limites d'une expression polysémique*. Institut national de la recherche scientifique Centre - Urbanisation Culture Société.

[10] GSMA. 2014. Understanding the Internet of Things (IoT). [...] SIMARD. 2018. *Développement d'un système d'IoT (Internet of Things) pour le Smart Lighting sous la Plateforme IBM.* . Mémoire de fin d'études

[11] SIMARD. 2018. *Développement d'un système d'IoT (Internet of Things) pour le Smart Lighting sous la Plateforme IBM.* . Mémoire de fin d'études

[12] Jean-Pierre Hauet « L'Internet des objets deux technologies clés les réseaux de communication et les protocoles », Deuxième partie, REE N°5/2016.

Annexe

SN	COMPOSANTS	LA DESCRIPTION	QUANTITÉ
1	Résistance	180	1
2	Résistance	6,8 K Ω	1
3	Résistance	56	1
4	Condensateur	47 μ F, 63 V (condensateur électrolytique)	1
5	Condensateur	100 μ F, 63 V (condensateur électrolytique)	1
6	Condensateur	1000 μ F, 25V (condensateur électrolytique)	1
7	Condensateur	10 nF (disque en céramique)	1
8	NE555	555 Minuterie IC	1
9	BC547	Transistor PNP	1
dix	BC557	Transistor NPN	1
11	1N4004	Diode redresseur	3

Tableau 1: Les composants utilisé d'un le convertisseur DC-DC.

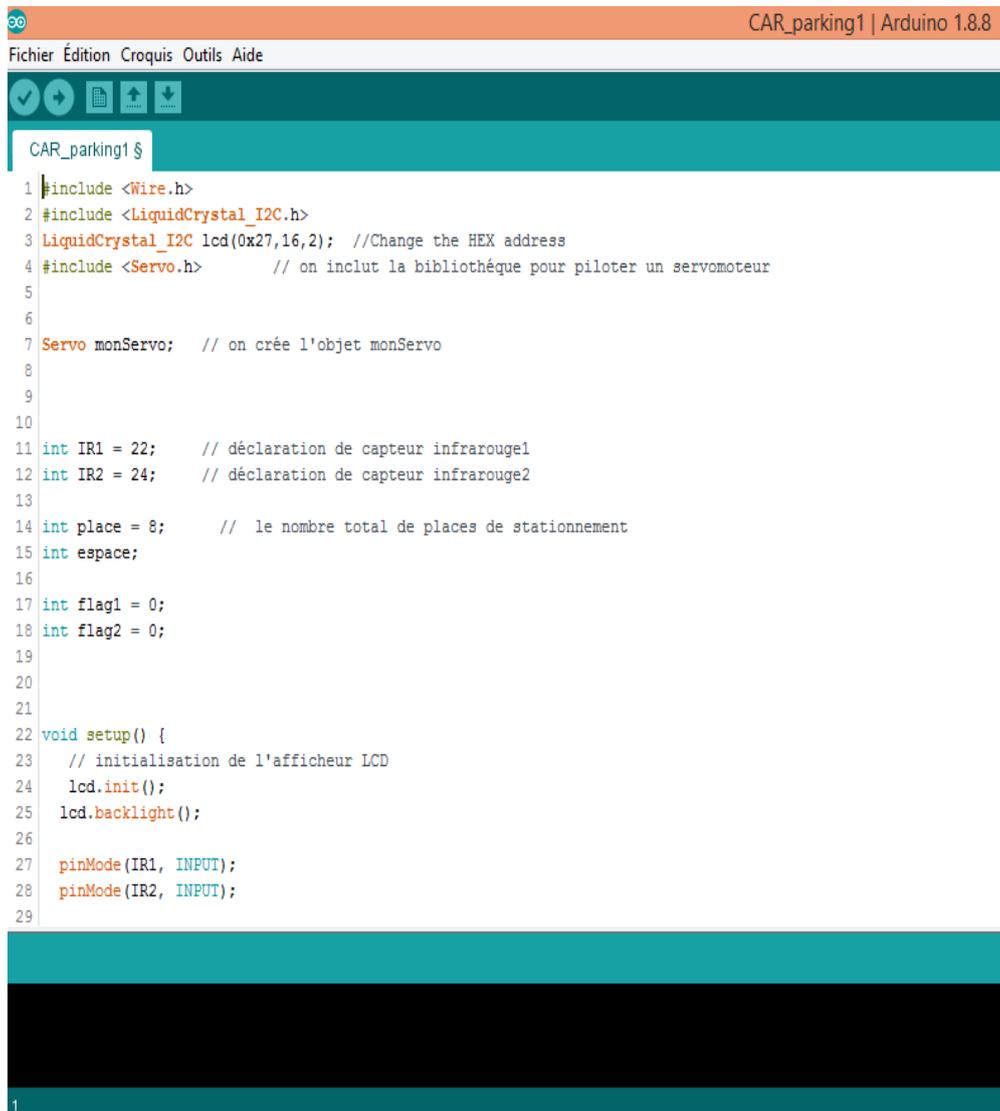
Coût du matériel

Composants	Prix
Arduino Mega 2560	3200 DA
Wifi ESP32	2400 DA
Panneau solaire	750 DA
Capteur infrarouge (x2)	900 DA
Capteur d'humidité du sol	450 DA
Servomoteur (x3)	1800 DA
LCD avec I2C	1200 DA
Shield L293D	200 DA
LDR (x5)	450 DA
Pompe à eau	1400 DA
Résistance (x6)	60 DA
Alimentation	200 DA
PTIX TOTAL : 13010 DA	

Tableau 2: Prix des composants.

Code Arduino

Parking intelligent



```

CAR_parking1 | Arduino 1.8.8
Fichier Édition Croquis Outils Aide
CAR_parking1 $
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27,16,2); //Change the HEX address
4 #include <Servo.h> // on inclut la bibliothèque pour piloter un servomoteur
5
6
7 Servo monServo; // on crée l'objet monServo
8
9
10
11 int IR1 = 22; // déclaration de capteur infrarouge1
12 int IR2 = 24; // déclaration de capteur infrarouge2
13
14 int place = 8; // le nombre total de places de stationnement
15 int espace;
16
17 int flag1 = 0;
18 int flag2 = 0;
19
20
21
22 void setup() {
23 // initialisation de l'afficheur LCD
24 lcd.init();
25 lcd.backlight();
26
27 pinMode(IR1, INPUT);
28 pinMode(IR2, INPUT);
29

```

```
29
30 // Initialisation du servo
31 monServo.attach(45); // la pin utilisé par le servomoteur PWM
32 monServo.write(100); // servo commandé dans sa position initiale
33
34
35 Serial.begin(9600);
36
37
38 lcd.setCursor(0,0);
39   lcd.print(" systeme ");
40   lcd.setCursor(0,1);
41   lcd.print("parking");
42   delay (2000);
43   lcd.clear();
44   espace = place;
45
46 }
47
48 void loop() {
49
50   if(digitalRead (IR1) == LOW && flag1==0)
51   {
52   if(espace>0)
53   {
54     flag1=1;
55     if(flag2==0){
56       monServo.write(0);
57       espace = espace-1;
```

```

CAR_parking1 $
57  espace = espace-1;
58  }
59  }
60  else {
61  lcd.setCursor (0,0);
62  lcd.print(" parking complet ");
63  lcd.setCursor (0,1);
64  lcd.print("  :(  ");
65  delay (3000);
66  lcd.clear();
67  }
68  }
69
70  if(digitalRead (IR2) == LOW && flag2==0)
71  {
72    flag2=1;
73    if(flag1==0)
74    {
75      monServo.write(0);
76      espace = espace+1;}
77  }
78
79  if(flag1==1 && flag2==1)
80  {
81    delay (1000);
82    monServo.write(100);
83    flag1=0, flag2=0;
84  }
85

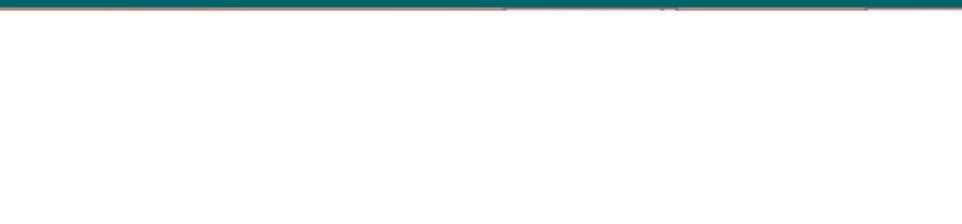
```



```

76  espace = espace+1;}
77  }
78
79  if(flag1==1 && flag2==1)
80  {
81    delay (1000);
82    monServo.write(100);
83    flag1=0, flag2=0;
84  }
85
86  lcd.setCursor (0,0);
87  lcd.print("nbr de place: ");
88  lcd.print (place);
89
90  lcd.setCursor (0,1);
91  lcd.print(" il reste: ");
92  lcd.print (espace);
93  }

```



Irrigation automatique

```
auto_arrosage2 | Arduino 1.8.8
Fichier Édition Croquis Outils Aide
Téléverser
auto_arrosage2$
1 // humidité
2
3 int humidity = A0;
4 int sensorValue;
5
6 // moteur pompe à eau
7
8 int in1 = 9;
9 int in2 = 8;
10
11
12 void setup() {
13
14   Serial.begin(9600);
15
16   pinMode(humidity, INPUT); // déclaration du capteur d'humidité
17
18
19   // shield
20
21   pinMode(in1, OUTPUT);
22   pinMode(in2, OUTPUT);
23
24
25 }
26
27 void loop() {
28   // humidité
29   sensorValue = analogRead(A0); // lecture humidité;
30
31   // -----
32   sensorValue = analogRead(A0); // lecture humidité;
33   Serial.println("humid"); // affichage humidité
34   Serial.println("sensorValue: ");
35
36   if (sensorValue > 700) // si sol sec (dans l'eau : 500, dans l'air:1023)
37   {
38     digitalWrite(in1, HIGH);
39     digitalWrite(in2, LOW);
40     delay(500);
41
42   }
43   else {
44     digitalWrite(in1, LOW);
45     digitalWrite(in2, LOW);
46   }
47
48 }
```