

**République Algérienne Démocratique et Populaire.
Ministère de L'Enseignement Supérieur et de la Recherche
Scientifique.
UNIVERSITE SAAD DAHLEB DE BLIDA**

Faculté des sciences

Département Informatique



**MEMOIRE DE FIN D'ETUDES
Pour l'obtention
Du Diplôme de Master en Informatique
Option : Ingénierie Des Logiciels**

THEME

**Gestion précise d'une exploitation agricole
connectée**

Réaliser par : Bahi Abderrahmane et Boukabous Rayan

Promotrice : Mme. Cherfa Imane (USDB1)

Remerciements

On remercie ALLAH, le Tout-Puissant, le Donneur de tout, de nous avoir donné le savoir et la faculté de pouvoir poursuivre nos études.

Au terme de ce travail, nous tenons à remercier sincèrement nos chers parents pour leurs sacrifices, leur soutien et leurs encouragements tout au long de notre parcours scolaire.

Nous remercions vivement notre encadrante, Dr. Cherfa Imene, pour son soutien, sa patience, sa disponibilité et ses précieux conseils, qui nous ont permis de réaliser ce projet.

Notre profonde gratitude aux membres du jury, qui nous honorent par leur évaluation de ce mémoire de fin d'études.

Enfin, nous remercions tous les enseignants qui ont contribué à notre formation, ainsi que toutes les personnes qui nous ont aidés de près ou de loin dans l'élaboration de ce sujet

Dédicaces

À mes chers parents,

Les mots ne suffisent pas pour exprimer ma gratitude pour tous vos sacrifices, encouragements et votre soutien toutes ces années.

À mon frère et à ma sœur.

À toute ma famille et à mes amis.

Bahi Abderrahmane

Dédicace

Je souhaite exprimer ma profonde gratitude à mon frère Nazim, pour son soutien indéfectible et ses encouragements constants tout au long de ce parcours. tes conseils ont été inestimables.

Je tiens également à remercier mes parents, pour leur soutien inconditionnel, leur patience et les nombreux sacrifices consentis pour me permettre d'atteindre mes objectifs. Votre appui a été essentiel à la réalisation de ce travail.

Je suis reconnaissant envers mes amis, pour leur camaraderie, leur soutien et les moments partagés. Votre présence a grandement enrichi cette expérience.

Enfin, je remercie sincèrement les membres de la société LiliumPharma, avec qui j'ai eu l'opportunité d'apprendre énormément. Votre expertise, votre soutien et votre dévouement ont été une source d'inspiration et ont grandement contribué à mon développement professionnel.

Résumé :

L'irrigation agricole est l'un des principaux consommateurs d'eau douce sur la planète. Une mauvaise gestion de l'eau crée des problèmes tels que le gaspillage d'eau et la diminution de la productivité agricole en termes de qualité et de quantité. Il est donc crucial de mettre en œuvre des techniques d'irrigation intelligente. L'Internet des objets (IoT) offre un important potentiel pour optimiser l'irrigation.

Cependant, certains systèmes d'irrigation peuvent rencontrer des problèmes tels que des boucles d'irrigation infinies et l'absence de surveillance du système dans le cas de systèmes autonomes ou une forte dépendance humaine dans certains systèmes de surveillance à distance. En tenant compte de ces scénarios, ce mémoire présente la conception et la réalisation d'un système d'irrigation basé sur la technologie IoT et une application web, garantissant une surveillance et un contrôle à distance en temps réel des données environnementales et du processus d'irrigation, avec la mise en œuvre d'un modèle matériel pour assurer l'efficacité et la fiabilité du système.

Mots clés :

Internet des objets (IoT), Agriculture de précision, Système d'irrigation, Gestion de l'eau, Application web.

Abstract

Agricultural irrigation is a major consumer of freshwater on the planet. Inadequate water management leads to problems like water waste and declining agricultural productivity (both quality and quantity). This necessitates the implementation of smart irrigation techniques, where the Internet of Things (IoT) offers significant potential for optimization.

While some irrigation systems can face issues such as infinite irrigation loops and lack of system monitoring in the case of autonomous systems or high human dependence for remote monitoring, this thesis presents the design and implementation of a smart irrigation system based on IoT technology and a web application. This system ensures real-time remote monitoring and control of environmental data and the irrigation process, while employing a hardware model to guarantee system efficiency and reliability.

Keywords:

Internet of Things (IoT), Precision Agriculture, Irrigation System, Water Management, Web Application.

ملخص

يعتبر ري المحاصيل الزراعية من أكثر القطاعات استهلاكاً للمياه العذبة على الكوكب. أين يؤدي سوء إدارة المياه إلى مشاكل مثل إهدار المياه وانخفاض إنتاجية الزراعة (من حيث الجودة والكمية). لذلك، من الضروري تنفيذ تقنيات الري الذكي، حيث توفر إنترنت الأشياء إمكانية مهمة لتحسين عملية الري.

على الرغم من أن بعض أنظمة الري يمكن أن تواجه مشاكل مثل حلقات الري اللانهائية وغياب مراقبة النظام في حالة الأنظمة المستقلة أو الاعتماد البشري الكبير في بعض أنظمة المراقبة عن بعد، فإن هذه الأطروحة تقدم تصميم وتنفيذ نظام ري ذكي يعتمد على تقنية إنترنت الأشياء وتطبيق ويب. يضمن هذا النظام المراقبة والتحكم عن بعد في الوقت الفعلي لبيانات البيئة وعملية الري، مع تطبيق نموذج أجهزة لضمان كفاءة النظام وموثوقيته.

الكلمات المفتاحية:

إنترنت الأشياء (IOT) ، الزراعة الدقيقة، نظام الري، إدارة المياه، تطبيق الويب.

Table des matières

Introduction générale	1
Chapitre 1 : État de l'art.....	2
Introduction.....	2
Définition d'un Système de systèmes (SoS).....	2
L'internet des objets (IoT).....	2
Définition	2
Exemples d'application de l'IoT	2
Les Composants d'une architecture Ido.....	7
Travaux connexes sur l'agriculture intelligente	11
Applications de l'IoT en agriculture intelligente.....	11
Technologies et solutions existantes	12
Discussion	20
Conclusion	21
Chapitre 2 : Conception du Système.....	22
Introduction.....	22
Conception Matérielle.....	22
Diagramme de blocs internes du système à réaliser	22
Algorithme d'irrigation proposé	23
Analyse et Conception logicielle	25
Analyse.....	25
Conception	27
Conclusion	33
Chapitre 3 : Réalisation du système	34
Introduction.....	34
Réalisation Matériel	34
La carte de développement ESP-32	34
Capteur d'humidité et de température DHT11	36
Capteur capacitif d'humidité du sol.....	37
Module de relais de déclenchement de bas niveau	38
Mini pompe à eau submersible 3-6V	40
Branchement global de système.....	41
Développement du la carte Esp 32.....	42
Réalisation logicielle :.....	43
Environnement et technologies de développement :.....	43
Présentation des applications	46
Résultats	60
Discussion des résultats	61

Conclusion	61
Conclusion générale	62
Références	63

Liste des figures

Figure 1 : Les applications de l'Internet des Objets Médicaux [25]	4
Figure 2 : Illustration des différentes applications de l'IoT dans les villes intelligentes [8].....	6
Figure 3 : Applications de l'ido dans l'Agriculture [7].....	7
Figure 4 : exemple Composants d'une architecture IoT [9]	8
Figure 5 : Comparaison des protocoles de communication Ido [10]	9
Figure 6 : Comparaison des protocoles de communication Ido [9]	10
Figure 7:Comparaison entre des protocoles dans la couche d'application [10]	11
Figure 8:Algorithme d'irrigation automatique [13]	14
Figure 9:composants du système proposé par [14].....	15
Figure 10:Schéma-bloc et organigramme du système d'irrigation de surface automatique [15].....	16
Figure 11:Schéma fonctionnel du système [16].....	17
Figure 12:le système Hydroponique de [21].....	19
Figure 13 : schéma block interne du système à réaliser.....	22
Figure 14 : Organigramme du système du microcontrôleur	24
Figure 15 : Diagramme de cas d'utilisation.....	25
Figure 16 : Diagramme de séquence de Cas d'utilisation Consulter les valeurs actuelles des mesures d'environnement	27
Figure 17 : Diagramme de séquence de Cas d'utilisation Consulter l'historique de changement des valeurs d'environnement dans le temps.....	28
Figure 18 : Diagramme de séquence de Cas d'utilisation Lancer le processus d'irrigation	29
Figure 19 : Diagramme de classe	30
Figure 20 : Carte de développement ESP32 38 broches [26]	35
Figure 21 : Capteur d'humidité et de température DHT11 [26]	36
Figure 22 : branchement de capteur DHT11	37
Figure 23 : Capteur capacitif d'humidité du sol [26]	37
Figure 24 : branchement de capteur capacitif d'humidité de sol.....	38
Figure 25 : Module de relais 5VDC [26]	39
Figure 26 : branchement de Module Relais	40
Figure 27 : Mini pompe à eau submersible 3-6V [26].....	40
Figure 28 : Branchement du mini pompe à eau	41
Figure 29 : Le branchement global de système avec les différents composants.....	41
Figure 30 : partie du code de processus Irrigation	43
Figure 31 : Architecture MVT [24]	44
Figure 32 : Connecter et inscrire	48
Figure 33 : page de Tableau de bord	49
Figure 34 : système de notification	50
Figure 35 : page d'historique des données.....	51
Figure 36 : page des terrains d'irrigation.....	52
Figure 37 : page des paramètres.....	53
Figure 38 : les pages dans l'interface mobile	53
Figure 39 : Tableau de Bord Application React	55
Figure 40 : Interface Accès aux détails des capteurs	57
Figure 41 : Interface configuration et accès à base de données	59
Figure 42 : fonctionnement du système complet réalisé.....	60
Figure 43 : fonctionnement du système avec pompe activé	61

Liste des tableaux

Tableau 1 : Dictionnaire de données	33
---	----

Liste d'abréviation

AC: Alternating Current
ADC: Analog-to-Digital Converter
AMQP: Advanced Message Queuing Protocol
API: Application Programming Interface
ARM: Advanced RISC Machine
BLE: Bluetooth Low Energy
CAN: Controller Area Network
COAP: Constrained Application Protocol
COM: Communication Port
CSS: Cascading Style Sheets
DAC: Digital-to-Analog Converter
DC: Direct Current
DDS: Data Distribution Service
DEVKIT: Development Kit
DHT: Digital Humidity and Temperature
ESP: Espressif
GND: Ground
GPIO: General-Purpose Input/Output
GPRS: General Packet Radio Service
GSM: Global System for Mobile Communications
HTML: Hypertext Markup Language
HTTP: Hypertext Transfer Protocol
I2C: Inter-Integrated Circuit
IA: Intelligence Artificial
IDE: Integrated Development Environment
iOS: iPhone Operating System
IOSeeds: Internet of Seeds
IOSR: Input Output Software Routine
IoT: Internet of Things
JSON: JavaScript Object Notation
KNN: K-Nearest Neighbors
LAN: Local Area Network
LCD: Liquid Crystal Display
MCU: Microcontroller Unit
MQTT: Message Queuing Telemetry Transport
MSE: Mean Squared Error
MVC: Model-View-Controller
MVT: Model-View-Template
NC: Normally Closed
NO: Normally Open
ORM: Object-Relational Mapping
PCB: Printed Circuit Board
PWM: Pulse-Width Modulation
QSPI: Quad Serial Peripheral Interface
REST: Representational State Transfer
RFID: Radio-Frequency Identification

ROM: Read-Only Memory
RTC: Real-Time Clock
SMS: Short Message Service
SOS: System of systems
SQL: Structured Query Language
SRAM: Static Random-Access Memory
SSE: Server sent event
STI: Serial Transfer Interface
TCP: Transmission Control Protocol
UART: Universal Asynchronous Receiver-Transmitter
UDP: User Datagram Protocol
USB: Universal Serial Bus
VCC: Voltage Common Collector
VS: Voltage Source
WAN: Wide Area Network
WAPI: WLAN Authentication and Privacy Infrastructure
WFA: Wi-Fi Alliance
WPA: Wi-Fi Protected Access
WPAN: Wireless Personal Area Network

Introduction Générale

L'eau, la ressource la plus importante pour la vie sur la planète, selon [1], depuis la quantité globale de l'eau qui existe dans le monde, environ 70% est consommée dans les processus d'irrigation dans l'agriculture. Ce dernier, face à l'accroissement démographique et aux problèmes de sécurité nutritionnelle, connaît une demande croissante dans le monde, où notre pays, l'Algérie par exemple, a intensifié le travail dans le domaine de l'agriculture ces dernières années. La consommation en eau dans ce secteur est d'environ 7 milliards de m³/an, sachant que la consommation globale nationale (consommation d'eau potable par la population, besoins des secteurs industriel et agricole) est de 10,6 milliards de m³/an [2].

Face au gaspillage très probable de l'eau, les mauvaises pratiques d'irrigation, telles qu'un mauvais timing d'arrosage, entraînent une série de problèmes en cascade. Cela se traduit par une diminution du rendement des cultures, une baisse de la productivité du secteur en termes de qualité et de quantité, des dommages environnementaux, une érosion des sols et une réduction de la circulation d'air dans la terre. L'effet net est un impact négatif important à la fois sur la société et sur l'environnement [1].

Ainsi, une approche plus précise est nécessaire pour le contrôle de l'irrigation. Une solution à explorer est la technologie de l'Internet des objets (en anglais Internet of Things, IoT), qui consiste en des applications exploitant des données de capteurs pour permettre le suivi, le contrôle, la précision et les processus de décision intelligents dans l'arrosage agricole. En exploitant les solutions existantes, il est possible d'utiliser des systèmes embarqués d'arrosage automatique basés sur des algorithmes qui doivent être combinés et supervisés, ou des systèmes d'irrigation supervisés par les agriculteurs mais posant une forte dépendance à ces derniers et ne tenant pas compte de l'absence d'une ou plusieurs parties du système.

Dans ce rapport, en exploitant les technologies et solutions existantes, nous proposons la réalisation d'un système d'irrigation basé sur les technologies du web et l'efficacité de l'IoT, tout en assurant le suivi, le contrôle à distance et en temps réel d'un système d'irrigation. Nous prenons en considération l'absence de superviseur humain dans le système et la collecte historique des données pour des analyses futures en vue d'améliorer les profits pour un tel système.

Ce mémoire est structuré en trois chapitres principaux. Le chapitre 1 présente l'état de l'art, fournissant une revue des technologies et des méthodes actuelles. Le chapitre 2 se concentre sur l'analyse et la conception du système, divisée en deux parties : la conception matérielle et la conception logicielle. Enfin, le chapitre 3 détaille la réalisation du système, également en deux parties : la réalisation matérielle et la réalisation logicielle. Notre mémoire est clôturée par une conclusion générale dans laquelle nous présentons un bilan et des perspectives.

Chapitre 1 : État de l'art

Introduction

Les systèmes dispersés peuvent exprimer davantage de leur valeur s'ils travaillent ensemble, ce qui est le cas pour l'Internet des Objets (Ido). Ainsi, pour que les valeurs physiques des sous-systèmes de contrôle soient interconnectées et exploitables dans un grand système, ce chapitre présente le modèle de référence concernant l'architecture, quelques technologies de communication et les domaines d'application de l'Internet des Objets. Il aborde également leurs applications dans le domaine de l'agriculture, ainsi que les systèmes réalisés dans ce contexte pour le suivi et le contrôle des systèmes d'irrigation.

Définition d'un Système de systèmes (SoS)

Un Système de Systèmes (SoS) dans le domaine de l'ingénierie peut être défini comme un ensemble de systèmes autonomes interconnectés et coordonnés pour satisfaire une capacité ou des fonctions spécifiques que les systèmes individuels ne pourraient pas réaliser seuls. Ces systèmes interconnectés peuvent provenir de divers secteurs tels que le spatial, militaire, transport, télécommunications, santé, énergie, etc. L'ingénierie des systèmes de systèmes nécessite des processus de développement, des outils et des méthodes pour concevoir, produire et déployer ces systèmes complexes. Les progiciels utilisés doivent être capables de gérer les compatibilités entre les différents systèmes et éléments qui les composent, ainsi que de suivre les exigences fonctionnelles tout au long du cycle de vie du système [3].

L'Internet des objets (IoT)

L'Internet des Objets (IoT) représente une révolution technologique qui connecte les objets du quotidien à Internet. Cette section présente une définition détaillée de l'IoT, nous illustrerons également ces concepts avec des exemples concrets d'applications IoT, montrant comment cette technologie transforme divers secteurs tels que la santé, la domotique, l'agriculture et les villes intelligentes.

Définition

L'Internet des objets (Ido) est un exemple puissant de système de systèmes. Fait référence au réseau d'appareils interconnectés, de capteurs et de logiciels qui permet la communication et le transfert de données entre eux. Il permet la collecte et l'analyse de grandes quantités de données, qui peuvent être utilisées pour automatiser des tâches dans divers domaines. L'Ido a le potentiel de révolutionner des industries telles que les villes intelligentes et les transports intelligents, et il est devenu une étape importante dans la transformation numérique du monde [4].

Exemples d'application de l'IoT

Il existe plusieurs exemples d'applications de l'IoT dans notre monde qui devient de plus en plus interconnecté. Nous détaillons quelques exemples ci-dessous.

- **Applications des Soins de Santé : l'Internet des Objets Médicaux (Io TM)**

L'Internet des Objets Médicaux (Io TM) est un domaine émergent qui transforme la prestation des soins de santé grâce aux technologies de l'Internet des Objets (IoT). Cette technologie peut être utilisée de multiples façons, comme le montre l'exemple dans la figure 1.

Les appareils IoT trouvent de nombreuses applications dans les établissements de santé, notamment la télésurveillance, c'est-à-dire la surveillance des patients à domicile plutôt qu'à l'hôpital. Les données collectées par les appareils IoT peuvent être analysées et utilisées pour la détection précoce des maladies. Les capteurs IoT ont joué un rôle vital pendant la pandémie de COVID-19 en aidant les soignants à mieux surveiller les paramètres essentiels et à intervenir en cas de changements.

Les capteurs sont la clé de l'utilisation des dispositifs IoT dans les soins de santé. Ils servent de passerelle entre le monde physique et celui de l'information via la collecte de diverses données. Ces capteurs médicaux peuvent mesurer la température, la respiration, le rythme cardiaque, le poids, la conductance cutanée, le taux de glucose, etc. Ces informations sont transmises aux patients, professionnels de la santé et autres parties prenantes via des réseaux de capteurs sans fil.

Les récents progrès technologiques liés à [5] [6] l'IoT continueront à façonner l'avenir des soins de santé. Mais la nature sensible de ces informations médicales souligne l'importance cruciale de la sécurité et du respect de la vie privée. Des défis subsistent, tels que la forte consommation d'énergie liée à l'IoT et les risques liés à la sécurité des nombreux appareils utilisés. Toutefois, l'utilisation de l'IoT dans la santé reste prometteuse et ouvre la voie à de nouveaux progrès [7].

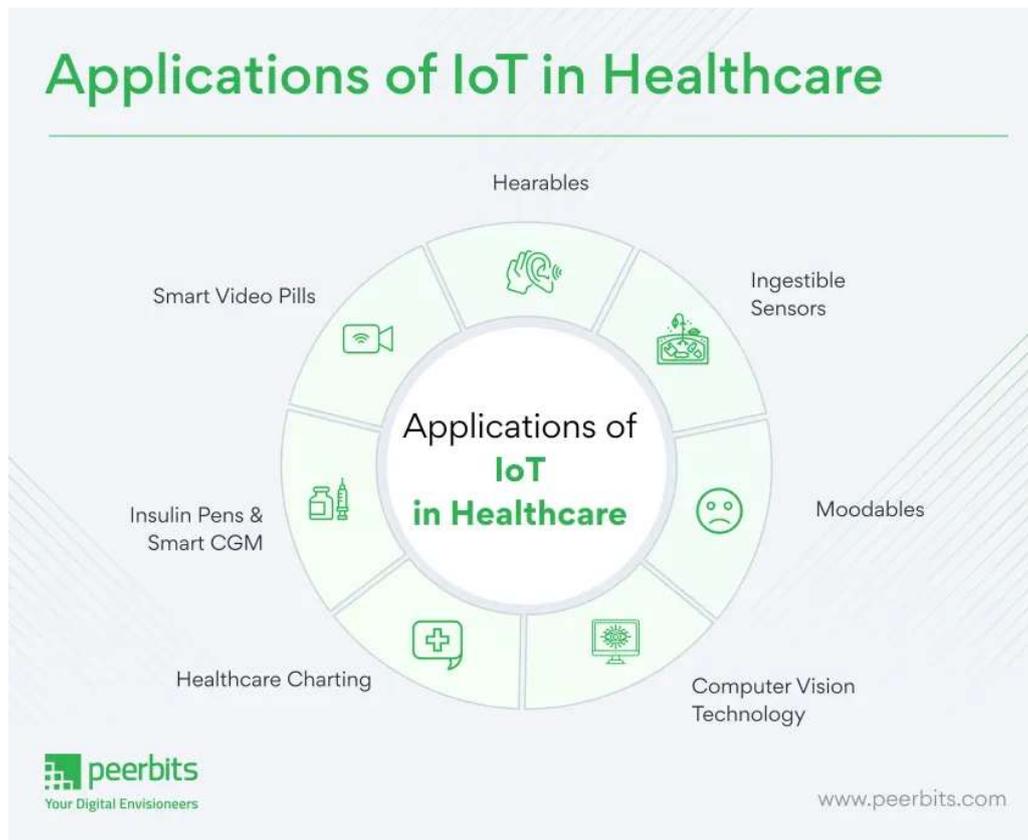


Figure 1 : Les applications de l'Internet des Objets Médicaux [25]

- **L'industrie 4.0**

L'Industrie 4.0, également connue sous le nom d'Industrie du Futur, est la quatrième révolution industrielle qui se concentre sur l'intégration de nouvelles technologies pour optimiser les systèmes de production et les processus industriels

Elle transforme radicalement la fabrication en intégrant l'IA, le « cloud computing » et les robots. Ces technologies exploitent les données des capteurs connectés pour optimiser la production, améliorer la qualité et réduire les coûts, propulsant le secteur manufacturier vers une ère connectée et intelligente [7].

- **Les maisons intelligentes (Smart Homes)**

L'essor des maisons intelligentes, propulsées par l'Internet des objets (IoT), transforme notre manière d'interagir avec notre habitat. Si le concept trouve ses racines dans les années 1970 avec le protocole X10, ce n'est que récemment qu'il a connu une adoption généralisée. L'automatisation de la maison reste au cœur de cette évolution, visant à améliorer la commodité et l'efficacité.

Les applications des maisons intelligentes sont multiples. Elles permettent:

- Surveillance et détection : caméras intelligentes, détecteurs de mouvement et d'ouverture, détecteurs de fumée et de monoxyde de carbone.
- Contrôle d'accès : serrures intelligentes, serrures intelligentes connectées à des caméras.

- Automatisation pour dissuader les intrusions : programmation de l'éclairage, simulateurs de présence.
- Assistance en cas d'urgence : boutons d'urgence, intégration avec les services d'urgence [7].

- **Les Villes intelligentes (Smart Cities)**

Les villes intelligentes, ou "Smart Cities" en anglais, utilisent l'Internet des objets pour résoudre différents problèmes urbains. Imaginez une ville équipée de capteurs et d'appareils sans fil qui collectent des données pour améliorer la gestion de la ville. Ces systèmes sont complexes car ils impliquent de nombreux appareils, technologies et services avec une diversité d'applications comme le montre la figure 2.

L'un des usages de ces technologies est la gestion du trafic. Les villes densément peuplées souffrent souvent d'embouteillages. Les villes intelligentes utilisent des technologies de communication pour collecter des données en temps réel et adapter la circulation en conséquence. On parle alors de "Systèmes de trafic intelligents" (STI) qui collectent des données en temps réel grâce aux appareils IoT pour optimiser la circulation.

Un autre domaine d'application de l'IoT est la gestion intelligente des déchets. Les systèmes actuels sont souvent surchargés dans les zones urbaines. L'objectif est d'utiliser des appareils IoT pour optimiser la collecte des déchets et réduire leur impact sur l'environnement. La nécessité de processus plus écoénergétiques et d'environnements plus sains motive le développement de ces technologies. On peut transformer des poubelles et des bacs de recyclage en appareils IoT, équipés de capteurs qui signalent quand il faut les vider, optimisant ainsi les collectes [7].

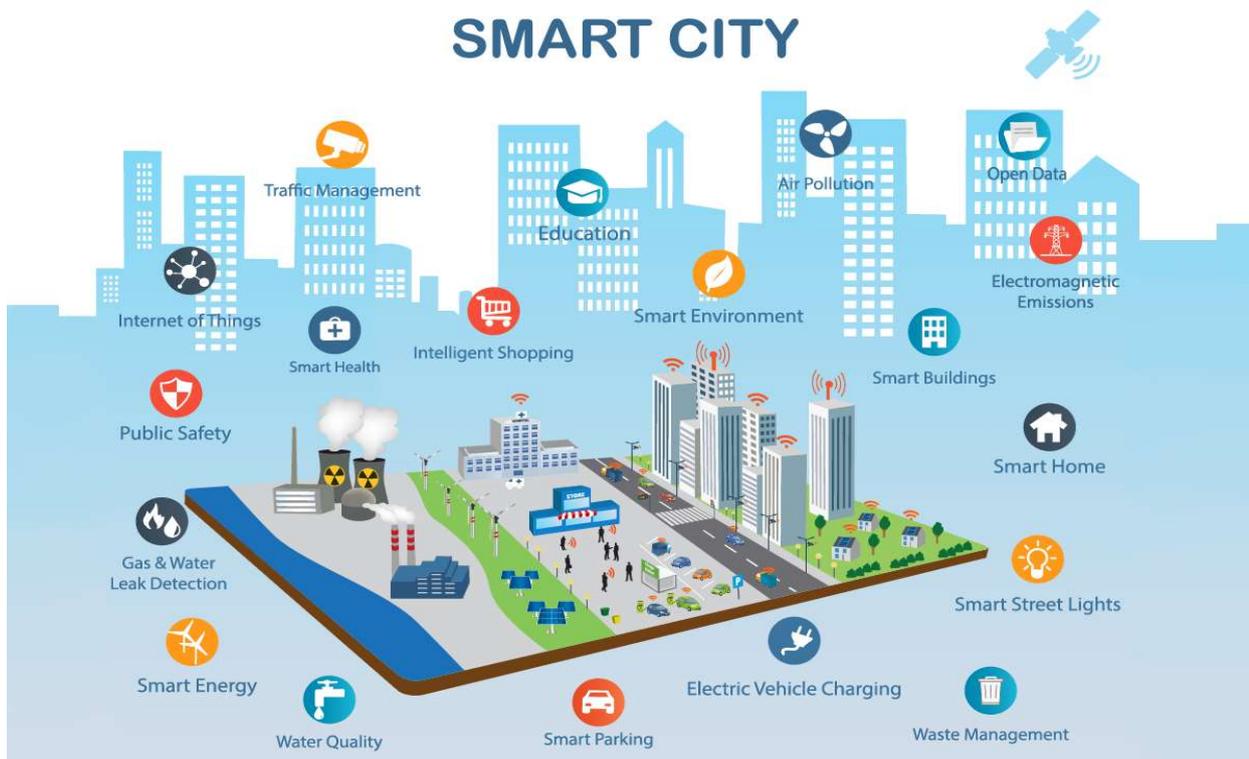


Figure 2 : Illustration des différentes applications de l'IoT dans les villes intelligentes [8]

- **L'IoT dans l'agriculture**

Cette section traite du rôle des techniques IoT dans le domaine agricole. Le rendement des cultures dépend de nombreux paramètres tels que le type de plante, la quantité d'eau, la température ambiante, le type et la température du sol, les nutriments présents, l'ensoleillement, etc. Pour mesurer ces paramètres dans différents environnements, on utilise des capteurs à faible coût. Ces capteurs sont légers, ils observent les conditions environnementales mais beaucoup sont incapables de se connecter à Internet pour stocker les données collectées.

Afin de recevoir les données des capteurs et effectuer leur traitement initial, de petits ordinateurs peu énergivores appelés "passerelles" (Gateway) sont utilisés. Une passerelle est un appareil qui collecte les données d'une variété de capteurs et les transfère vers Internet, où elles sont ensuite stockées sur un serveur ou un cloud. Différents protocoles peuvent être utilisés pour établir la communication entre les capteurs et la passerelle, tels que ZigBee, MQTT, etc., principalement conçus pour les appareils à faible coût. La passerelle utilise des modules Wi-Fi (ou GPRS) pour la connexion Internet.

Les données collectées sont stockées dans une base de données ou sur un cloud, puis traitées à l'aide du machine Learning ou du Big Data pour soutenir la prise de décision.

Les systèmes sont conçus de manière à ce que l'utilisateur final puisse contrôler les appareils (actionneurs) tels que la pompe à eau, la pompe à brouillard, etc., à distance. Les systèmes IoT permettent aussi un contrôle automatique : si l'humidité du sol est insuffisante, la pompe à eau se déclenche, si l'ensoleillement est faible, l'éclairage est activé automatiquement pour une croissance optimale des plantes. Ce type d'automatisation, issu de l'apprentissage profond (deep Learning) ou du Big Data, réduit le travail physique et augmente la production.

Les techniques IoT s'appliquent non seulement à l'agriculture traditionnelle, mais aussi aux cultures hydroponiques, où les plantes poussent dans des solutions nutritives, sans sol. L'IoT assure un suivi et un contrôle efficaces de ces nutriments, optimisant la croissance des plantes [7],

La figure 3 illustre des chemins d'application de ces technologies pour améliorer le produit agricole.

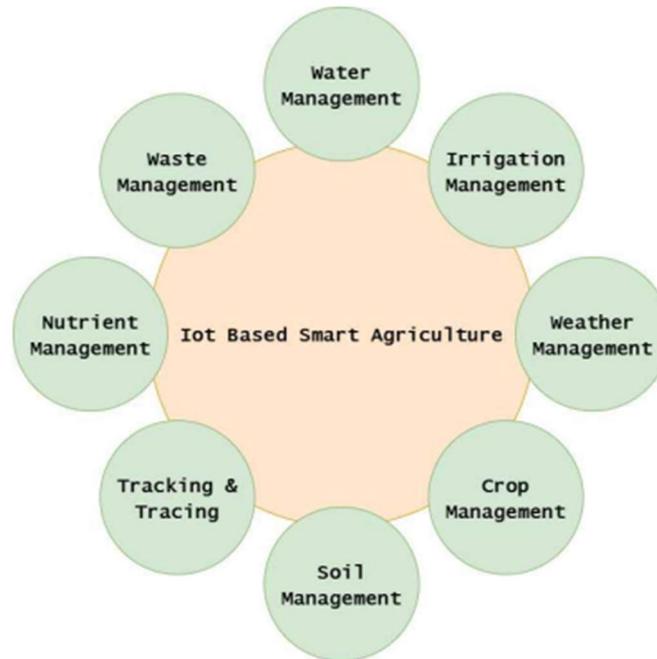


Figure 3 : Applications de l'ido dans l'Agriculture [7]

Les Composants d'une architecture Ido

Il n'y a pas de consensus global sur l'architecture de l'IoT, partagé par le monde entier et les chercheurs. Diverses architectures ont été suggérées. Selon certains chercheurs, l'architecture de l'IoT comprend trois niveaux, tandis que d'autres soutiennent l'idée. Architecture à quatre niveaux. Selon eux, en raison de l'évolution de l'IoT, l'architecture en trois niveaux ne peut pas répondre aux exigences des applications. Afin d'améliorer les caractéristiques de qualité du système, certains chercheurs ont proposé une architecture à cinq niveaux. Dans cette partie, nous examinons les composants du système pour une architecture proposée par [9] est exprimée dans la figure ci-dessous (voir Figure4)

➤ La couche de sensation physique

La couche de détection physique, également appelée la couche de perception ou la couche de dispositif, constitue le fondement de l'architecture de l'Internet des objets (IoT). Cette couche est responsable de l'interaction directe avec le monde physique à travers des senseurs et des actuateurs.

Son rôle est d'identifier les objets et de collecter des informations à leur sujet. De nombreux types de capteurs, comme les RFID, les codes-barres 2D et les capteurs génériques, peuvent être

attachés aux objets pour collecter des informations. Le choix du type de capteur dépend des besoins de l'application. Les informations collectées par ces capteurs peuvent concerner la localisation, les variations de l'environnement (air, etc.), le mouvement, les vibrations, etc.

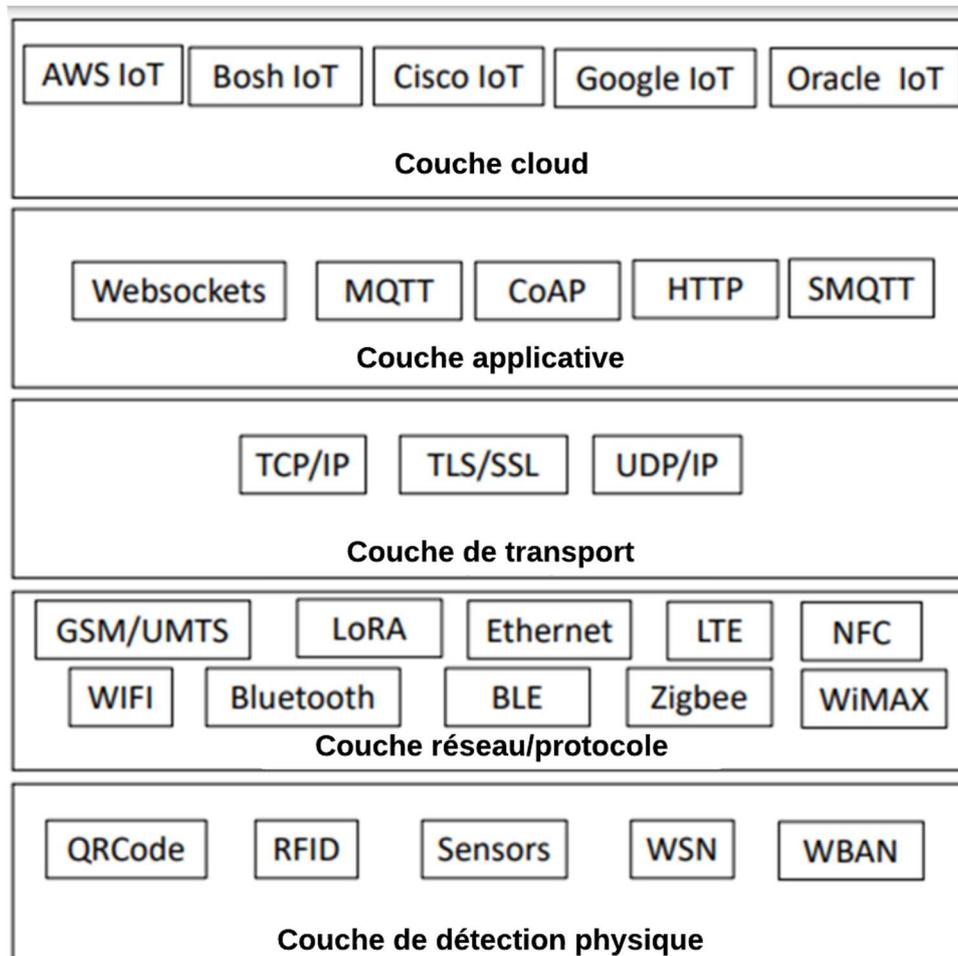


Figure 4 : exemple Composants d'une architecture IoT [9]

➤ Le niveau de réseau et de protocole

Les protocoles de communication jouent un rôle crucial dans les systèmes IoT en facilitant la communication et l'échange de données entre les appareils et les autres parties distantes du réseau. D'après certaines recherches, les protocoles de communication de l'IoT sont séparés en deux niveaux distincts : le réseau utilisant des capteurs et le réseau de passerelles [9]. Donc, il existe deux niveaux de protocoles pour les réseaux de capteurs

1. Technologies de communication des Réseaux basées sur les capteurs

Filaire: RS-485, Modbus, CAN bus

Sans's fil: ZigBee, LoRa WAN, Bluetooth Low Energy (BLE), Wi-Fi, 6LoWPAN

Fonction : Collecte des données des capteurs et transmission au passerelle

2. Technologies de communication des Réseaux de passerelles

- Ethernet

- 3G/4G/5G
- 6LoWPAN
- Satellite
- Wi-Fi

Fonction : Transmettre les données du réseau de capteurs vers Internet, un LAN ou un autre réseau.

Les figures 5 et 6 présentent les caractéristiques, ainsi que les bénéfices et les inconvénients des protocoles de communication de données les plus appropriés.

Technologie	Compétence	Débit de données	Portée
RFID	Authentification, Surveillance, Identification, Gestion de l'information	Jusqu'à 640 kbps	1-10 m
NFC	Communication, Sécurité	106 - 424 kbps	≤ 10 cm
6LowPan	Protection et Sécurité	20-250 kbps	10-30 m
BLE	Communication, Réduction de la consommation d'énergie	1 mbps	100 m
ZigBee	Communication, Surveillance	20-250 kbps	10-100 m

Figure 5 : Comparaison des protocoles de communication Ido [10]

Protocole de Communication	Normes	Protocole de Chiffrement	Consommation d'Énergie	Avantages	Inconvénients
6LoWPAN	IEEE 802.15.4	AES	Faible	Faible traitement	Absence d'authentification
RPL	IETF RPL	AES	Faible	Faible traitement	Vulnérabilité à de nombreuses attaques
NFC	ISO/IEC 14443	RSA, DSA	Faible	Simplicité de déploiement	Portée limitée
Bluetooth	IEEE 802.16	AES, ECDH	Moyenne/Très faible (BLE)	Faible consommation	Suivi de la vie privée/identité
Wi-Fi	IEEE 802.11i/e/g	AES	Élevée	Mobilité et efficacité	Portée limitée
ZigBee	IEEE 802.15.4	AES	Faible	Dispositifs à faible coût et faible consommation d'énergie	Transmission unique de la clé non protégée
WiMax	IEEE 802.16	RSA	Moyenne	Prend en charge l'authentification	Mobilité limitée
3G/4G/5G	UMTS/LTE	RSA, 3DES	Moyenne	Portabilité	Limitation de la batterie

Figure 6 : Comparaison des protocoles de communication Ido [9]

➤ La couche de transport

On trouve deux fonctions essentielles pour les applications qui interagissent sur a réseau à travers le transport layer :

- **Protocole de contrôle de transmission (TCP)** : Le protocole est axé sur les connexions. Avant de transmettre les données, il établit une relation fidèle entre le transmetteur et le destinataire. TCP assure la vérification des erreurs, la livraison en ordre et la retransmission des paquets. Il est préférable d'utiliser des applications telles que le téléchargement de fichiers, la navigation sur internet et l'envoi d'emails qui nécessitent une livraison de données fiable.
- **Protocole de données d'utilisateur (UDP)** : Il simplement envoie des paquets de données (datagrammes) de manière autonome, sans avoir à établir d'abord une connexion. UDP priorise la vitesse sur la cohérence. Il convient parfaitement pour des applications où la perte de données occasionnelle est tolérable, comme le gaming en ligne ou le streaming en temps réel d'audio/vidéo.

➤ La couche Application

Elle permet aux utilisateurs de visualiser, contrôler et gérer les appareils. Dans cette partie l'IoT impose des exigences spécifiques aux protocoles d'application, notamment en termes de faible consommation d'énergie et de faible capacité des appareils. Pour répondre à ces besoins, plusieurs protocoles ont été développés, et ils sont comparables selon les besoins en fonction de leurs caractéristiques exprimées dans la figure 7:

- **AMQP (Advanced Message Queuing Protocol)** : un protocole flexible et performant pour la messagerie et le streaming de données.
- **COAP (Constrained Application Protocol)** : un protocole léger conçu pour les appareils à ressources limitées.

- **DDS (Data Distribution Service)** : un protocole robuste pour la publication et l'abonnement aux données en temps réel.
- **MQTT (Message Queuing Telemetry Transport)** : un protocole simple et efficace pour la communication entre clients et serveurs.

Protocole/Paramètre	CoAP	MQTT	XMPP	AMQP	DDS
Sécurité	DTLS	SSL	SSL	SSL	SSL, DTLS
Messagerie	Requête/Réponse	Publication/Abonnement	Requête/Réponse Publication/Abonnement	Publication/Abonnement	Publication/Abonnement sans courtier
Transport	UDP	TCP	TCP	TCP	TCP, UDP
QoS	Confirmable : Accusé de réception Non-confirmable : sans accusé de réception	3 niveaux : Sans accusé de réception livré au moins une fois exactement une fois	Pas de support pour QoS	Support pour QoS	23 politiques : Sécurité, fiabilité, durabilité, priorité, etc.
Taille de l'en-tête du format de message	4	2	Pas d'en-tête Utilise XML Stanza	8	-
Fiabilité	Peu fiable	Fiable	Fiable	Fiable	Très fiable

Figure 7: Comparaison entre des protocoles dans la couche d'application [10]

➤ Couche de données et de services cloud

Le développement d'applications pour l'Internet des Objets (IoT) peut s'avérer ardu. De nombreux composants matériels, langages de programmation et techniques de communication sont utilisés. Cela complique la gestion de l'infrastructure complète nécessaire à l'exécution de l'application, y compris le matériel et les logiciels.

Pour répondre à cette problématique, les grandes entreprises de l'IoT ont développé des Framework cloud. Ces Framework facilitent le développement d'applications IoT en agissant comme des ensembles d'instructions. Ils gèrent en arrière-plan les aspects techniques complexes de la communication, vous libérant ainsi pour vous concentrer sur la fonctionnalité réelle de votre application.

Travaux connexes sur l'agriculture intelligente

L'agriculture intelligente, ou agriculture de précision, intègre les technologies modernes telles que l'Internet des Objets (IoT), et le big data pour optimiser la production agricole et la gestion des ressources. Cette section examine les travaux connexes et les recherches récentes dans le domaine de l'agriculture intelligente. Nous analyserons les différentes approches développées pour améliorer l'efficacité et la rentabilité des pratiques agricoles.

Applications de l'IoT en agriculture intelligente

Les changements climatiques et les événements naturels extrêmes ont une influence significative sur le développement des cultures et la productivité agricole. Heureusement, grâce à l'Internet des objets (IoT), il est possible de recueillir une multitude de données environnementales à l'aide de capteurs variés et de les stocker dans un système centralisé d'information [11].

A. Contrôle du sol et des plantes

- Des dispositifs mesurant l'humidité, la température et d'autres paramètres du sol sont utilisés pour élaborer des programmes d'arrosage efficaces et faciliter la prise de décisions.

B. Gestion de l'irrigation et optimisation de l'utilisation de l'eau

- Le niveau d'eau dans les réservoirs est surveillé en temps réel. La détection des fuites contribue également à réduire le gaspillage d'eau.

C. Préservation des cultures

- L'Internet des objets (IoT) joue un rôle crucial dans la détection des maladies et des parasites grâce à l'utilisation de capteurs. Lorsque le niveau de nuisibilité dépasse un seuil prédéfini, les agriculteurs reçoivent des alertes afin qu'ils puissent prendre les mesures nécessaires.

D. Amélioration des cultures

- En utilisant une base de données centralisée accessible via le cloud, les agriculteurs ont accès à des informations sur le moment optimal pour semer, combattre les maladies et récolter.

Technologies et solutions existantes

- **Système d'irrigation automatisé pour la surveillance des serres [12]**

-Ils ont utilisé un système connecté avec des capteurs et des actionneurs à l'intérieur d'une serre pour surveiller et contrôler les processus d'irrigation.

-Des capteurs d'humidité mesurent la concentration d'humidité dans le sol. Cette information est transmise à un microcontrôleur via un convertisseur analogique-numérique (ADC).

-Le microcontrôleur analyse les données et évalue les besoins en eau des plantes en se basant sur les niveaux d'humidité du sol. Il active ensuite l'électrovanne via un circuit relais afin de fournir de l'eau aux plants en utilisant un système **d'irrigation goutte à goutte**.

-De plus, des capteurs de température et de lumière surveillent les éléments environnementaux tels que la température, l'humidité et les niveaux de lumière, qui peuvent avoir un impact sur la croissance des plantes. Un écran LCD affiche les valeurs environnementales en temps réel.

L'irrigation goutte-à-goutte offre plusieurs avantages :

- Meilleure gestion de l'eau pour les plantes.
- Réduction des maladies et des nuisibles.
- Diminution des coûts d'exploitation et d'engrais.
- Lutte contre l'érosion du sol.
- Adaptable à de nombreux terrains.

Résultats :

-Selon les estimations, l'irrigation manuelle de ce champ consomme environ 41 litres d'eau. Grâce à l'utilisation du drip setup, il faut environ 30 litres d'eau pour arroser tout le terrain, ce qui permet d'économiser environ 27% de l'eau. En réduisant l'utilisation de l'eau à 20 litres grâce aux deux systèmes mentionnés précédemment, à savoir l'irrigation goutte à goutte et le système automatique, on réalise une économie d'eau d'environ 48,78%.

-la pratique de capteurs économiques et la simplicité des circuits font de ce projet un produit peu coûteux.

- **Un système d'irrigation intelligent basé sur l'IOT utilisant de la terre Prévision de l'humidité et de la météo [13]**

Ce travail présente un système d'irrigation basé sur une technologie open source qui prédit les besoins d'irrigation en tenant compte des paramètres du sol et des données de prévisions météorologiques.

L'application surveille l'ensemble de la ferme à distance emplacement. Économie d'énergie L'algorithme est utilisé dans le nœud pour économiser de l'énergie. Basé sur l'arborescence Le protocole est utilisé pour la collecte de données du nœud à la base gare. Système ayant deux nœuds, un nœud qui recueille toutes les valeurs des paramètres environnementaux et du sol et l'autre se compose d'une caméra pour capturer des images et surveiller les cultures.

Les Composants du système

- Arduino UNO Atmega328.
- Node ESP8266 Wi-Fi soc.
- Capteur d'humidité de sol F-28.
- Plateforme fournit des prévisions météorologiques.

- Cloud plateforme Thingspeak¹.

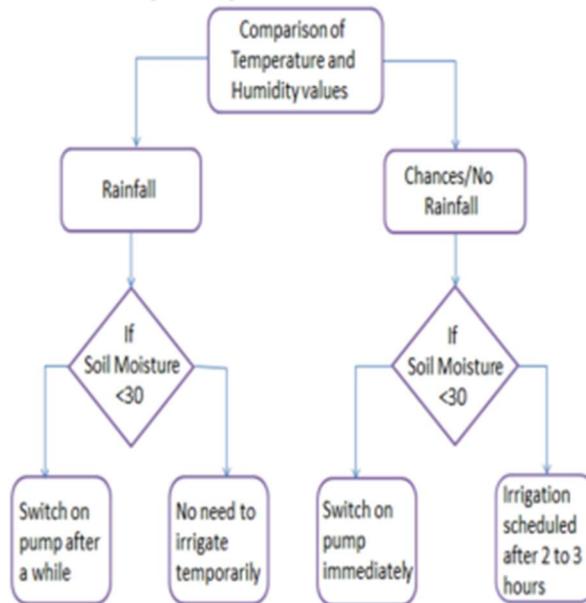


Figure 8:Algorithme d'irrigation automatique [13]

- **Système d'irrigation intelligent basé sur l'IoT [14]**

Ce travail présente l'utilisation d'un capteur d'humidité du sol et d'un capteur d'humidité relative et de la température connectés et implémentés via la plateforme cloud Thingspeak. Le système permet de surveiller et de visualiser les variables environnementales sur le tableau de bord Web Thingspeak grâce à l'exécution d'un code MATLAB. Ce code permet également le contrôle automatique d'une pompe à eau équipée d'un moteur à courant continu.

Les composants du système:

¹ ThingSpeak est un service en ligne de plateforme d'intégration et d'analyse IoT [5]

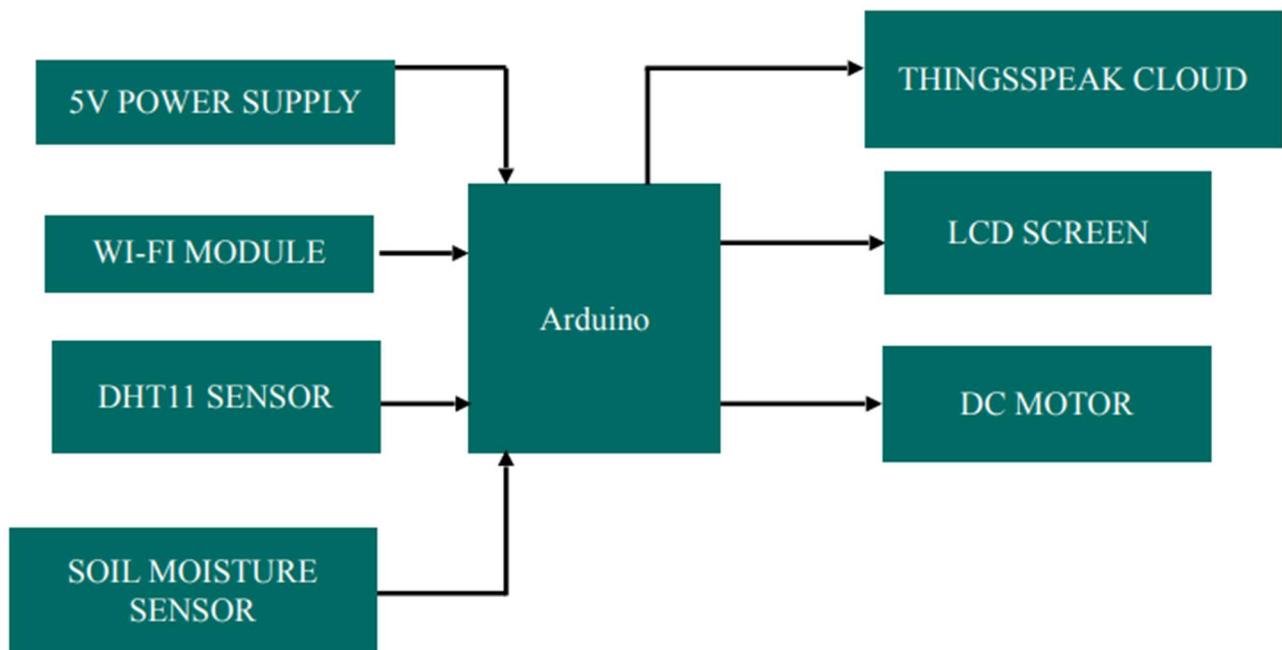


Figure 9: composants du système proposé par [14]

- **Automatisation du système d'irrigation par bassin à capteur d'humidité du sol [15]**

Ce travail présente l'automatisation de l'irrigation d'un champ agricole (60m * 14m) grâce à un réseau sans fil de capteurs. Le système utilise une pompe à eau automatisée et contrôlée par le service cloud Croplytics². L'article mentionne également l'utilisation d'un système solaire pour alimenter tous les besoins énergétiques du dispositif.

² Croplytics est un service en ligne de plateforme d'intégration et d'analyse IoT dans le demain d'agriculture [6]

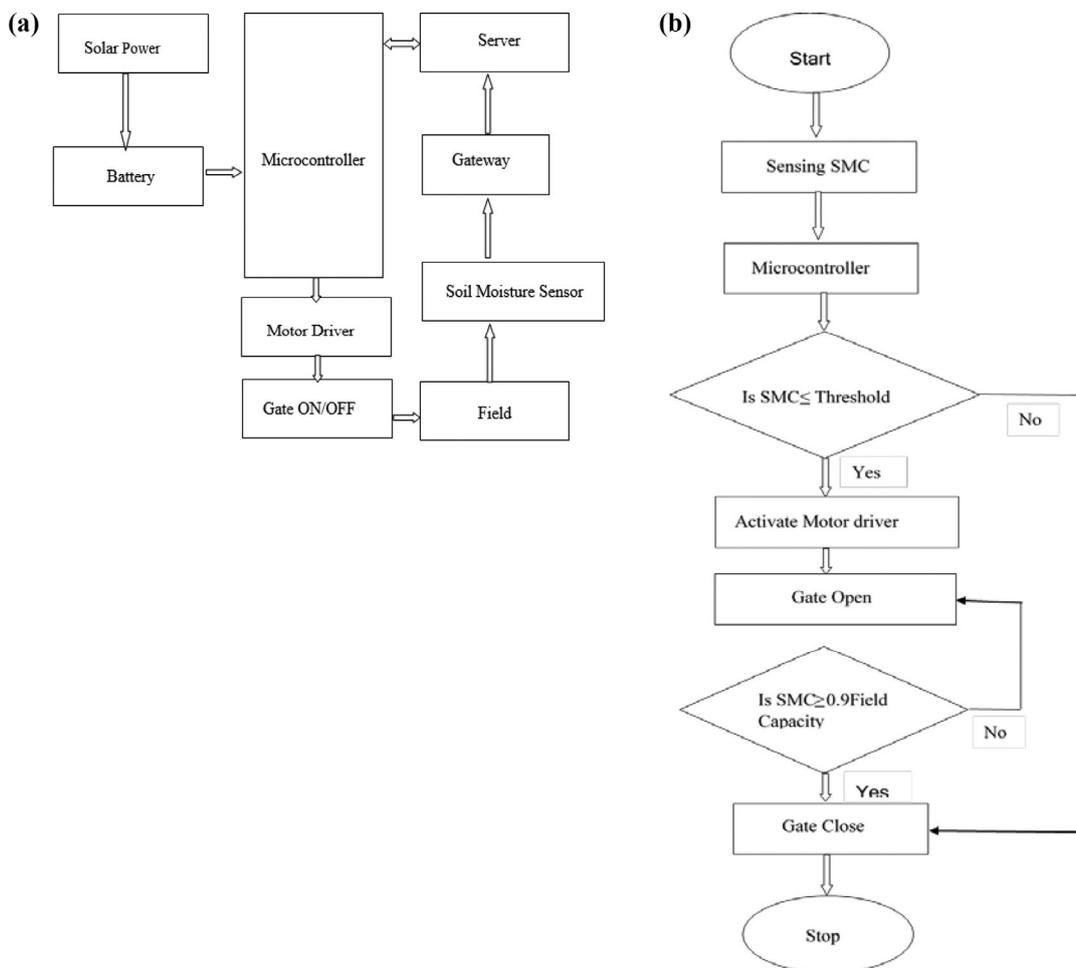


Figure 10: Schéma-bloc et organigramme du système d'irrigation de surface automatique [15]

Le système utilise la méthode d'irrigation de surface à travers une porte qui s'ouvre et se ferme en fonction de l'irrigation du champ. La quantité totale d'eau appliquée lors de chaque événement a été mesurée à l'aide d'un débitmètre à ultrasons.

La communication sans fil entre les capteurs d'humidité du sol et la passerelle est établie à l'aide du module LoRa et un module GSM est utilisé pour la communication entre la passerelle et les services cloud.

-Le système d'irrigation automatisé a été testé en conditions de sol nu avec 3 configurations de capteurs d'humidité.

Résultats :

La position optimale des capteurs :

Un manque d'humidité important : une profondeur de 37.5 cm, avec une prise d'eau de 25%.

Un faible manque d'humidité : une profondeur de 7,5 cm, avec une prise d'eau de 75%.

-Amélioration de l'efficacité d'irrigation jusqu'à 86.6%.

- **Développement d'un système d'irrigation hybride pour une agriculture intelligente utilisant des capteurs IoT et l'apprentissage automatique [16]**

-Le système utilise différents capteurs pour mesurer l'humidité du sol, le niveau de précipitations, la vitesse du vent, le niveau de l'eau, la température et l'humidité d'air (capteur DHT11). Il utilise également des modèles de machine Learning tels que K-voisins les plus proches (KNN), Naïve Bayes, Forêt aléatoire, et Régression logistique pour analyser les données en utilisant le service cloud ThinkSpeak.

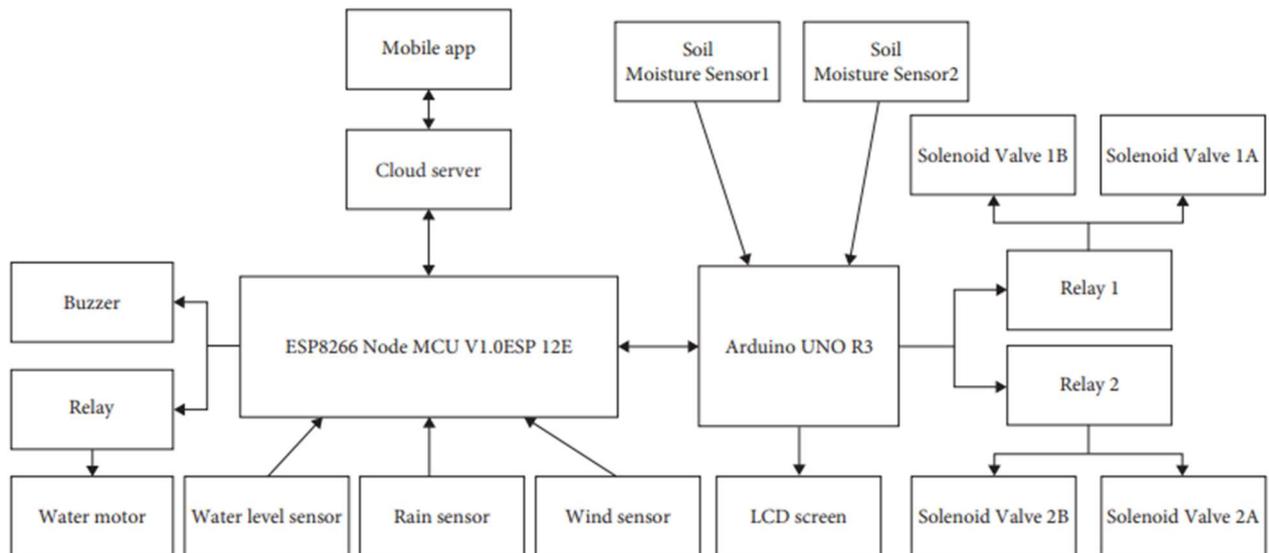


Figure 11: Schéma fonctionnel du système [16]

-Une application Android est développée avec la plateforme Blynk³ pour visualiser les paramètres environnementaux et pour le contrôle à distance. Les données collectées sont analysées avec différents algorithmes d'apprentissage automatique.

-Microprogramme développé en C/C++ avec l'IDE Arduino.

-Pour l'analyse des données, Jupiter Notebook et Anaconda sont utilisés.

Les résultats du taux de précision et de l'erreur carrée moyenne (MSE) sont les suivants :

-K-plus proche voisin (KNN) : 99,3% / 1,66.

-Naïve Bayes et Forêt aléatoire : 99,8% / 0,16.

-Régression logistique : 99,5% / 0,5.

- **Solutions innovantes pour une agriculture durable : systèmes d'irrigation pilotés par l'iot [17]**

Ce système d'irrigation utilise des capteurs connectés pour surveiller l'environnement et piloter l'irrigation. Il permet de contrôler une pompe à distance, de recevoir des notifications par SMS et d'analyser les données collectées via le service web Adafruit⁴. Le microprogramme du système est développé avec l'IDE Arduino, et utilise divers capteurs pour mesurer l'humidité du sol, la température, les précipitations et le niveau d'eau. La connectivité filaire et le module GSM garantissent une communication fiable. Ce système permet d'économiser l'eau, d'améliorer le rendement agricole et de simplifier la gestion des cultures.

- **Conception et mise en œuvre d'un système de surveillance de l'humidité du sol et d'irrigation basé sur ARM et IoT [18]**

Le système d'irrigation connecté utilise un modèle ARM (le Raspberry Pi 3B), qui centralise le traitement des données et le contrôle de l'ensemble.

Le modèle ARM dispose d'une puissance de traitement supérieure, permettant d'envisager des fonctionnalités plus avancées, le modèle a été programmé en Python.

³ Blynk est une plateforme IoT pour les smartphones qui est utilisée pour contrôler des microcontrôleurs via Internet [27]

⁴ Adafruit IO est un service basé sur le cloud spécialement conçu pour les projets IoT. Il sert de pont entre les appareils connectés et le cloud [28]

Le système exploite les technologies de communication Socket et I2C.

L'application web développée avec Flask, HTML et Bootstrap offre la surveillance et le contrôle à distance du système.

Le système peut fonctionner en mode automatique, déclenchant l'irrigation en fonction de seuils prédéfinis d'humidité et de température. De plus, les utilisateurs reçoivent des notifications par email pour les informer de l'état du système.

- **Système d'irrigation intelligent basé sur l'IoT et la logique floue [19]**

Cet article présente un système d'irrigation intelligent basé sur l'IoT et la logique floue. Des capteurs et des actionneurs assurent une surveillance continue des conditions environnementales telles que la température de l'air, l'humidité et l'humidité du sol. Le système irrigue le champ à l'aide d'une électrovanne pilotée à distance. Toutes les données collectées par le microcontrôleur sont envoyées vers une base de données cloud pour analyse et traitement statistique.

Un algorithme de calcul flou qui s'exécute au niveau du microcontrôleur, responsable de l'irrigation de l'eau, est proposé comme suit :

- a. Si plus de 3 capteurs d'humidité du sol indiquent un sol sec :
La vanne servo s'ouvre **complètement**.
- b. Si le capteur de température indique du froid :
La vanne servo se **ferme**, quel que soit l'état d'humidité du sol.
- c. Si moins de 4 capteurs d'humidité du sol indiquent un sol normal, et que le capteur de température indique du chaud et que le capteur d'humidité ambiante indique du sec :
La vanne servo s'ouvre **partiellement**.
- d. Si moins de 4 capteurs d'humidité du sol indiquent un sol sec, et que le capteur de température indique du chaud et que le capteur d'humidité ambiante indique du sec :
La vanne servo s'ouvre **complètement**.
- e. Si moins de 3 capteurs d'humidité du sol indiquent un sol sec (quel que soit l'état de la température ou de l'humidité ambiante) :
La vanne servo s'ouvre **partiellement**.

Le texte original ne précise pas explicitement l'action pour le cas où 4 capteurs d'humidité du sol indiqueraient un sol normal. Dans ce cas La vanne servo doit se **ferme**, quel que soit l'état d'humidité du sol.

- **Smart & Green : Un Cadre d'Internet des Objets pour l'Irrigation Intelligente [20]**

Cet article propose le cadre IoT Smart&Green, qui met en œuvre un plan de gestion de l'irrigation basé sur :

Les informations sur le système d'irrigation fournies par les utilisateurs.

Les données météorologiques et d'humidité du sol proches des champs surveillés.

Les techniques de suppression des valeurs aberrantes.

Fonctionnalités principales

Prétraitement des données d'humidité du sol avec suppression des valeurs aberrantes pour obtenir un besoin d'eau d'irrigation plus précis.

Prévision de l'humidité du sol pour les champs sans capteurs dédiés.

Optimisation de la consommation d'eau en minimisant le besoin d'eau d'irrigation tout en garantissant un développement sain des cultures.

Résultats

-L'utilisation de données réelles d'humidité du sol permet d'économiser jusqu'à 90,4% d'eau par rapport à la méthode ne tenant compte que de la météo et des cultures.

-Les techniques de suppression des valeurs aberrantes (Zscore et MZscore) appliquées aux données réelles permettent d'économiser 89,3% d'eau.

-Pour l'approche basée sur le potentiel matriciel, ces mêmes techniques permettent d'économiser en moyenne 20,7% et 14,1% d'eau par rapport à la méthode sans correction.

-La prédiction de l'humidité du sol et son traitement par Zscore, MZscore et Chauvenet permettent d'économiser entre 56,4% et 90% d'eau par rapport à la méthode d'équilibre hydrique sans capteurs.

- **Optimisation et contrôle de l'agriculture hydroponique grâce à l'IOT [21]**

Ce projet explore l'utilisation de l'hydroponie, une technique de culture hors sol, comme solution potentielle aux défis de l'agriculture moderne. La fragmentation des terres agricoles, l'utilisation d'engrais et de pesticides chimiques font partie des obstacles croissants auxquels est confrontée la production alimentaire. Le système permette aux cultivateurs de surveiller et de contrôler leur système à distance, contrôlant et ajustant automatiquement les niveaux de nutriments, la température et l'éclairage.

Le système utilise des capteurs d'humidité et du niveau pH du sol, ainsi que des capteurs de température et d'éclairage, avec une carte Arduino ATmega328.

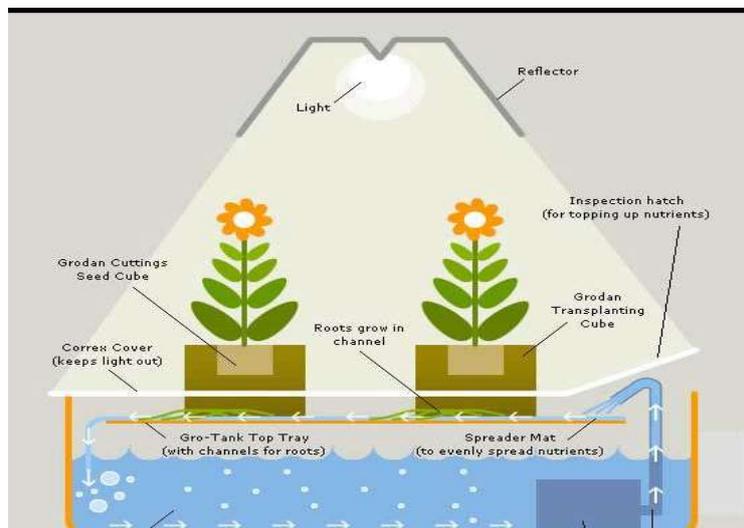


Figure 12: le système Hydroponique de [21]

Le système hydroponique proposé offre une solution plus efficace en termes de ressources que les méthodes traditionnelles en pleine terre. Cependant, ce système présente également des inconvénients tels qu'une plus grande sensibilité aux maladies, un coût d'installation plus élevé et la nécessité d'une maintenance régulière.

Discussion

Dans cette section, nous discutons les conclusions que nous avons tirées des travaux connexes. Nous examinerons les différentes technologies et équipements déployés pour la collecte de données et le contrôle des systèmes agricoles. De plus, nous analyserons les divers algorithmes d'irrigation développés pour optimiser l'utilisation de l'eau et améliorer l'efficacité des cultures. En comparant et en contrastant ces approches, nous mettrons en lumière les défis et les opportunités dans l'adoption d'une solution précise pour l'agriculture.

Matériel à utiliser

- Les cartes Arduino (fréquemment utilisées dans les projets [13] [14] [16]) avec un module de connectivité externe (ESP8266, module GSM...) peuvent être remplacées par une carte de développement ESP32. Cette dernière offre un nombre plus élevé de broches de connexion tant numériques qu'analogiques, un microcontrôleur plus puissant que celui de l'Arduino Uno, une carte réseau intégrée Wi-Fi et BLE, et un prix moins cher que l'Arduino Uno. Ces facteurs rendent la carte ESP32 la plus adaptable pour nos projets en termes de coût et de qualité.
- Les valeurs environnementales qui doivent être prises en compte lors du processus d'irrigation sont l'humidité du sol, la température et l'humidité de l'air afin d'éviter l'irrigation par temps très froid pour préserver les plantes, comme cela a été proposé dans [19]. D'autres valeurs telles que les prévisions météorologiques et les capteurs de pluie ont été utilisées, mais le cas de pluie ou l'état de l'air sont satisfaits par les valeurs d'humidité du sol, de température et d'humidité de l'air. Cependant, il reste un inconvénient de forte dépendance à ces deux capteurs. Il est donc conseillé d'avoir une disposition redondante de ces capteurs selon les besoins. Les capteurs de pH et de nutriments du sol sont plus adaptés pour le suivi de l'utilisation des engrais.
- Le capteur d'humidité du sol est très important dans les systèmes d'agriculture intelligente et dans le contexte du contrôle de l'irrigation qui est basé sur la valeur d'humidité du sol, mais les capteurs **résistifs** utilisés par [13] [14] doivent être remplacés par des capteurs de type **conductif**. Ces derniers coûtent un peu plus cher, mais ils fournissent des valeurs plus précises et plus fiables, ainsi qu'une durée de vie plus longue. De plus, les capteurs résistifs de base rendront le sol malsain pour les plantes avec le temps [22] .

Les services cloud

- Les services cloud qui offrent la manipulation des composants IoT sont également populaires et faciles à déployer pour les fonctionnements de la base d'un système IoT, tels que la collecte de données depuis les capteurs et le contrôle des actionneurs, avec des interfaces simples et modifiables. Cependant, généralement, ils disposent de frais pour continuer l'utilisation du système ou pour obtenir certaines fonctionnalités.
- Des services cloud tels que Thingspeak, Adafruit, Croplytics, Blynk offrent un plan gratuit pour commencer, mais avec le temps, les données des capteurs deviendront plus volumineuses, et dans le cas de Thingspeak, par exemple, le nombre de messages échangés par jour est limité à 8 200 messages par jour avec un intervalle de 15 secondes entre chaque message. Ainsi, ces services expriment des limites en termes d'utilisation, de scalabilité et de modifiabilité.

Protocole de connexion de la passerelle et du serveur

- Les technologies telles que le GSM, le LoRa, etc., peuvent couvrir de plus grandes zones, mais nécessitent plus de connaissances techniques pour le déploiement ainsi que pour la maintenance. Le BLE (Bluetooth Low Energy) consomme moins d'énergie mais couvre des zones de petite taille. Nous proposons d'utiliser une puce intégrée Wi-Fi dans l'ESP32 qui couvre une plage de 50m à 200m, ce qui convient aux zones agricoles de taille moyenne et aux serres intelligentes, car elle est plus facile à déployer et adaptée aux projets économiques. Si des plages de données plus importantes sont nécessaires, il est proposé d'utiliser soit un répéteur de signal Wi-Fi, soit des modules de connexion LoRa ou GSM compatibles avec l'ESP32.

Algorithmes d'irrigation

- Les algorithmes d'irrigation automatique appliqués dans [15] [13], ainsi que la logique floue de [19], expriment le problème de dépendance à un seuil de valeur et aux valeurs transmises par les capteurs, qui peuvent tomber en panne et déclencher une boucle d'irrigation infinie. Et cette valeur peut être modifiée selon les besoins des agriculteurs, mais il n'est pas pratique de modifier le code du microcontrôleur à chaque fois, ce qui souligne une limitation en termes de connaissances techniques nécessaires pour les agriculteurs.
- Dans le cas du contrôle en temps réel par l'utilisateur du système, il arrive qu'après avoir donné l'ordre d'ouvrir la vanne (pompe), une ou plusieurs parties du système tombent en panne (notamment un problème de réseau, surtout dans les zones agricoles). La vanne ne se ferme que si l'utilisateur le vérifie à nouveau.

Conclusion

L'IoT ajoute à l'agriculture traditionnelle un nouveau potentiel en permettant la communication entre les différentes parties du système, le suivi des valeurs physiques du terrain et le contrôle des composants à distance et en temps réel. Dans ce chapitre, nous avons présenté les domaines d'application de l'IoT, les technologies utilisées et leur application dans le domaine de l'agriculture et de l'irrigation intelligente. Nous avons également examiné les systèmes déjà réalisés et les points à considérer dans les prochaines étapes pour réaliser notre système, tels que le choix des composants physiques (capteurs, microcontrôleur, etc.), l'architecture du système, les technologies d'échange et le traitement des données, afin de maximiser les avantages d'un tel système et d'optimiser l'utilisation de l'eau dans l'irrigation agricole.

Chapitre 2 : Conception du Système

Introduction

Dans le chapitre précédent, nous avons exploré les technologies, les besoins et les différents systèmes d'irrigation. Dans ce chapitre, nous exploiterons les résultats de l'état de l'art pour proposer notre propre système d'irrigation. Nous allons présenter dans ce qui suit la conception matérielle puis logicielle de notre système, ainsi que l'algorithme d'irrigation proposé.

Conception Matérielle

Dans cette section, nous détaillons la conception matérielle du système d'irrigation précis. Nous présenterons le diagramme de blocs internes SysML (System Modeling Language), illustrant les composants et leur interconnexion au sein du système. En outre, nous décrivons l'algorithme d'irrigation proposé, qui utilise les données collectées pour optimiser la distribution de l'eau.

Diagramme de blocs internes du système à réaliser

Le système proposé est composé de quatre parties :

- La première partie est le réseau de capteurs et d'actionneurs (pompe à eau) connectés à une carte ESP32 qui permet la collecte des données de température, d'humidité de l'air et d'humidité du sol, ainsi que l'irrigation par la pompe.
- La deuxième partie est le microcontrôleur qui effectue le prétraitement et la transmission des données des capteurs, contrôle la pompe à eau et communique avec le serveur via la carte Wi-Fi intégrée.
- La troisième partie consiste en le serveur, qui joue le rôle d'intermédiaire du système, assurant la gestion des utilisateurs des microcontrôleurs et des données des capteurs.

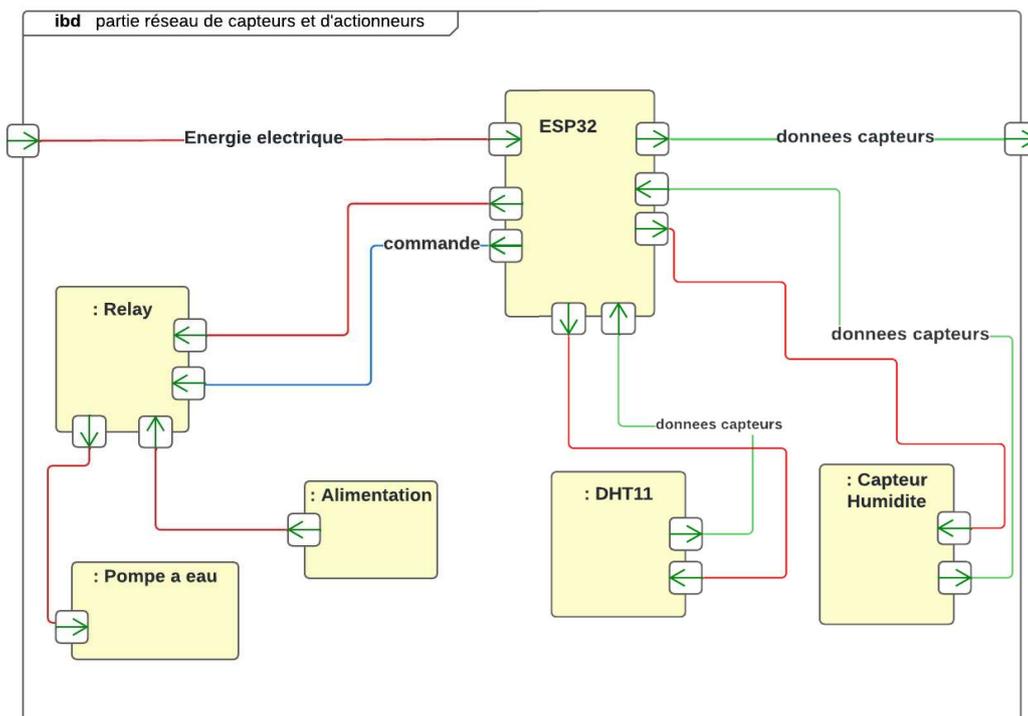


Figure 13 : schéma block interne du système à réaliser

- La quatrième partie est l'interface web, qui permet d'accéder aux données des capteurs en temps réel et de contrôler l'irrigation.

Algorithme d'irrigation proposé

Lorsque le système est alimenté, l'ESP lit les valeurs de température, d'humidité de l'air et d'humidité du sol. Ensuite, il établit une connexion avec le serveur. En cas d'échec de la connexion, il passe au processus d'irrigation. Sinon, il envoie les mesures des capteurs avant de synchroniser son état avec celui défini par le serveur. Il peut alors changer les valeurs du temps d'irrigation maximum, le seuil d'humidité du sol, et la fréquence des mises à jour en fonction des indications du serveur.

Selon les instructions du serveur, l'ESP commence l'irrigation. Ensuite, il vérifie si toutes les communications avec le serveur sont réussies. Sinon, il commence l'irrigation.

Le processus d'irrigation se déroule comme suit :

1. Il vérifie à nouveau les mesures des capteurs avant d'allumer la pompe à eau.
2. Ensuite, il surveille les valeurs environnementales jusqu'à ce que la pompe soit éteinte si l'une des conditions suivantes est remplie :
 - Le seuil d'humidité est dépassé,
 - Le temps d'irrigation maximum est atteint,
 - La température n'est pas appropriée pour continuer l'irrigation ($\leq 0^{\circ}\text{C}$).

Ensuite, il met à jour l'état de la pompe sur le serveur s'il y a une connexion Wi-Fi établie.

Ces étapes garantissent la synchronisation en temps réel avec le serveur, le contrôle à distance et permettent au microcontrôleur d'interagir en cas de perte de connexion avec le serveur. Si l'utilisateur ne surveille pas le système lors de l'activation de la pompe, ces précautions optimisent la consommation d'eau et minimisent les risques de sur-irrigation.

La figure 14 représente l'organigramme global de fonctionnement du microcontrôleur.

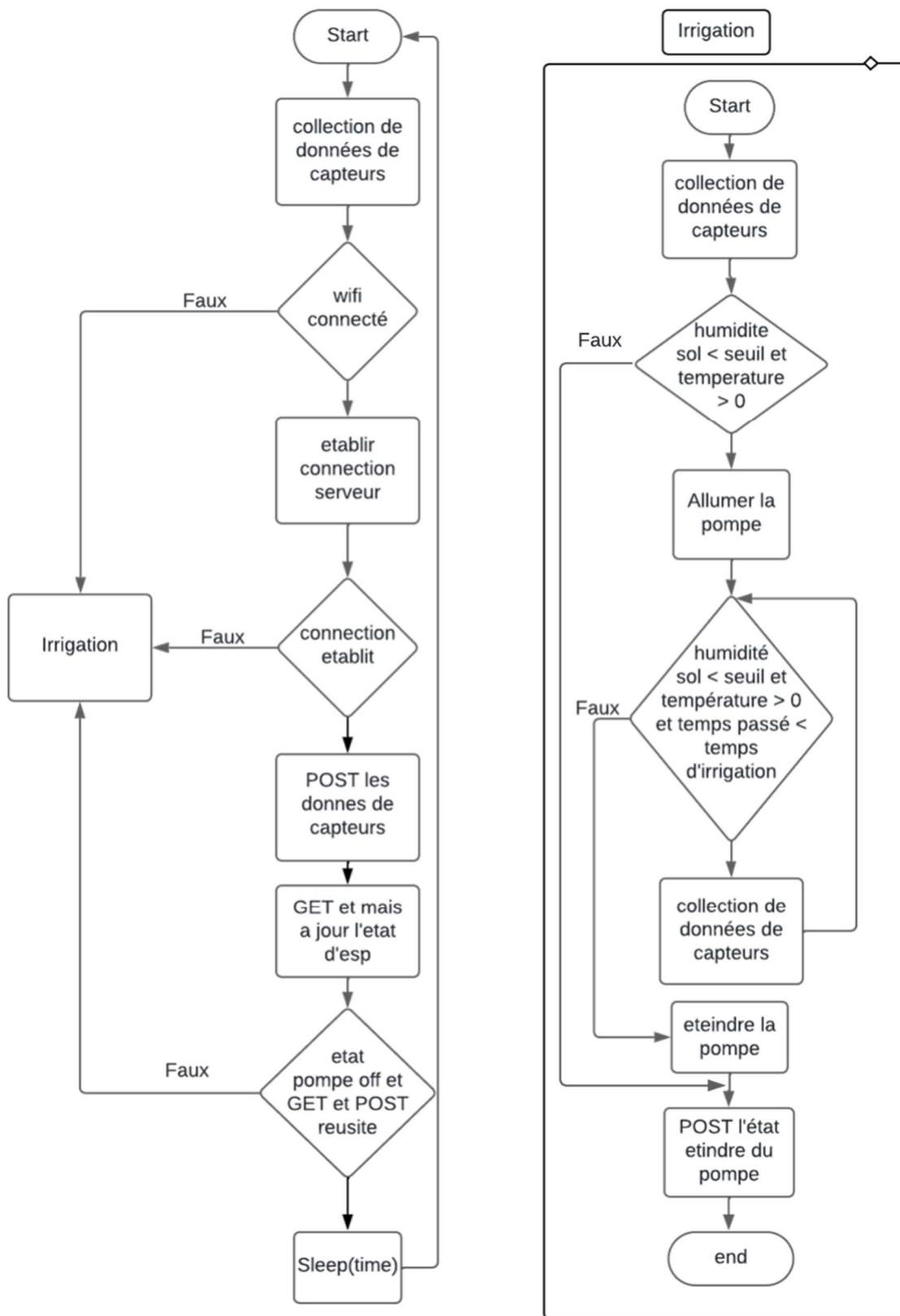


Figure 14 : Organigramme du système du microcontrôleur

Analyse et Conception logicielle

Cette section se concentre sur l'analyse et la conception logicielle du système d'irrigation précis. Nous y présenterons dans la phase d'analyse les différents cas d'utilisation (use cases) pour illustrer les interactions entre les utilisateurs et le système. Ensuite, dans la phase de conception, nous détaillerons le diagramme de séquence, montrant l'ordre des interactions dans divers scénarios, et nous exposerons le diagramme de classe, qui définit les structures et les relations des objets nécessaires au bon fonctionnement du système. Ensemble, ces éléments fournissent une vue complète et cohérente de la conception logicielle, assurant une implémentation efficace et alignée avec les objectifs du projet.

Analyse

Identification des acteurs du système

- **Client**

L'utilisateur principal du système est le client. Une fois authentifié, il peut gérer ses terrains d'irrigation présentés par les modules ESP32 connectés, consulter les mesures environnementales, commencer le processus d'irrigation, et consulter l'historique des changements des valeurs mesurées par les capteurs. Ceci est illustré dans la figure ci-dessous (voir Figure4).

Diagramme de cas d'utilisation

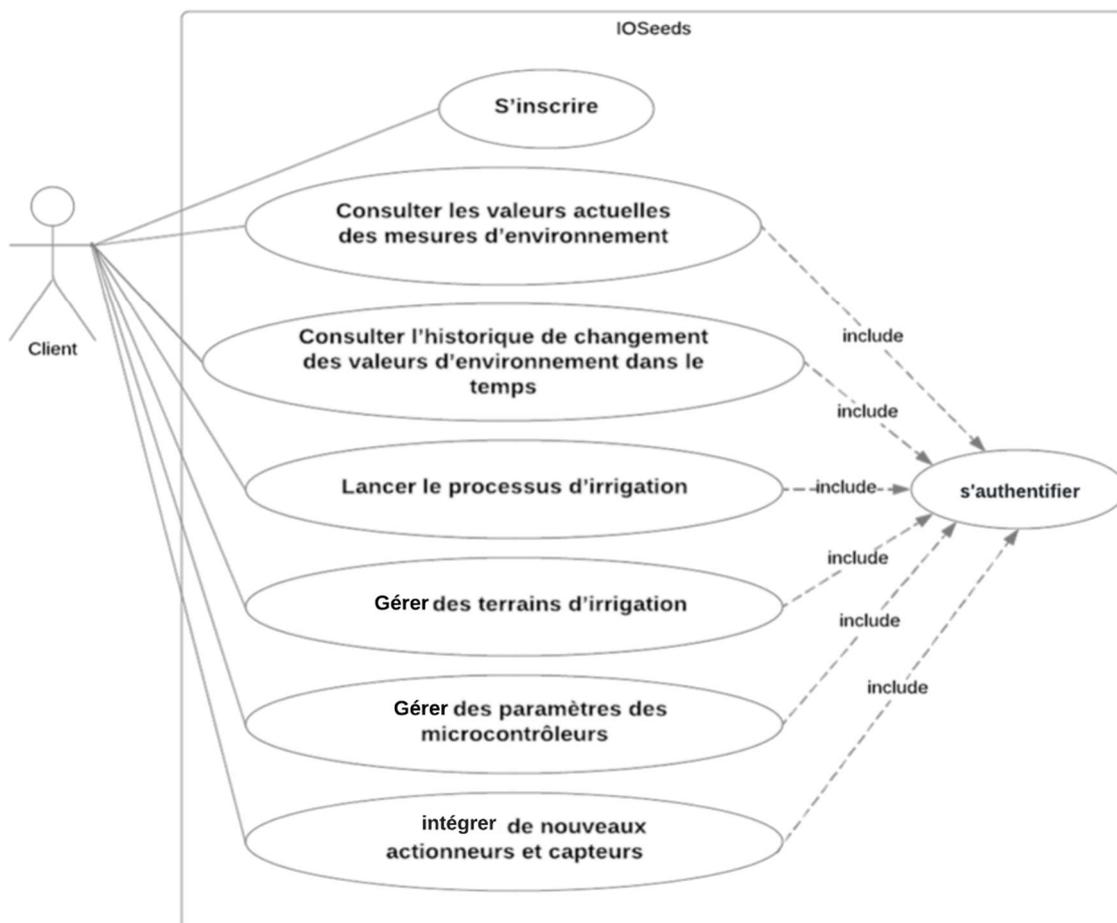


Figure 15 : Diagramme de cas d'utilisation

Description des cas d'utilisation

Dans ce qui suit nous présentons une petite description des cas d'utilisation :

- **S'inscrire** : Le client crée un nouveau compte pour la gestion de ses terrains d'irrigation, chacun identifié par le microcontrôleur associé.
- **Consulter les valeurs actuelles des mesures d'environnement** : Après l'authentification et l'initialisation du module ESP32, le client peut obtenir en temps réel les valeurs d'humidité du sol, de température et d'humidité de l'air, ainsi que l'état actuel de la pompe à eau concernée.
- **Consulter l'historique de changement des valeurs d'environnement dans le temps** : Le client peut consulter l'historique des valeurs environnementales du terrain d'irrigation sélectionné pour des analyses futures et une meilleure prise de décision.
- **Lancer le processus d'irrigation** : Le client peut commencer le processus d'irrigation en temps réel en activant la pompe à eau du terrain concerné pour une durée et un seuil d'humidité spécifiés par lui.
- **Gestion des terrains d'irrigation** : Le client peut gérer plusieurs terrains d'irrigation identifiés par les microcontrôleurs connectés, en ajoutant, supprimant et sélectionnant le terrain à suivre.
- **Gestion des paramètres des microcontrôleurs** : Les paramètres des microcontrôleurs, comme le seuil d'humidité, le temps maximal pour le processus d'irrigation, et la fréquence de mise à jour des valeurs des capteurs, peuvent être modifiés par l'utilisateur.
- **Intégration de nouveaux actionneurs et capteurs** : Selon ses besoins, le client peut intégrer d'autres actionneurs et capteurs pour améliorer le système d'irrigation.

Conception

Diagrammes de séquences

Diagramme de séquence de Cas d'utilisation Consulter les valeurs actuelles des mesures d'environnement

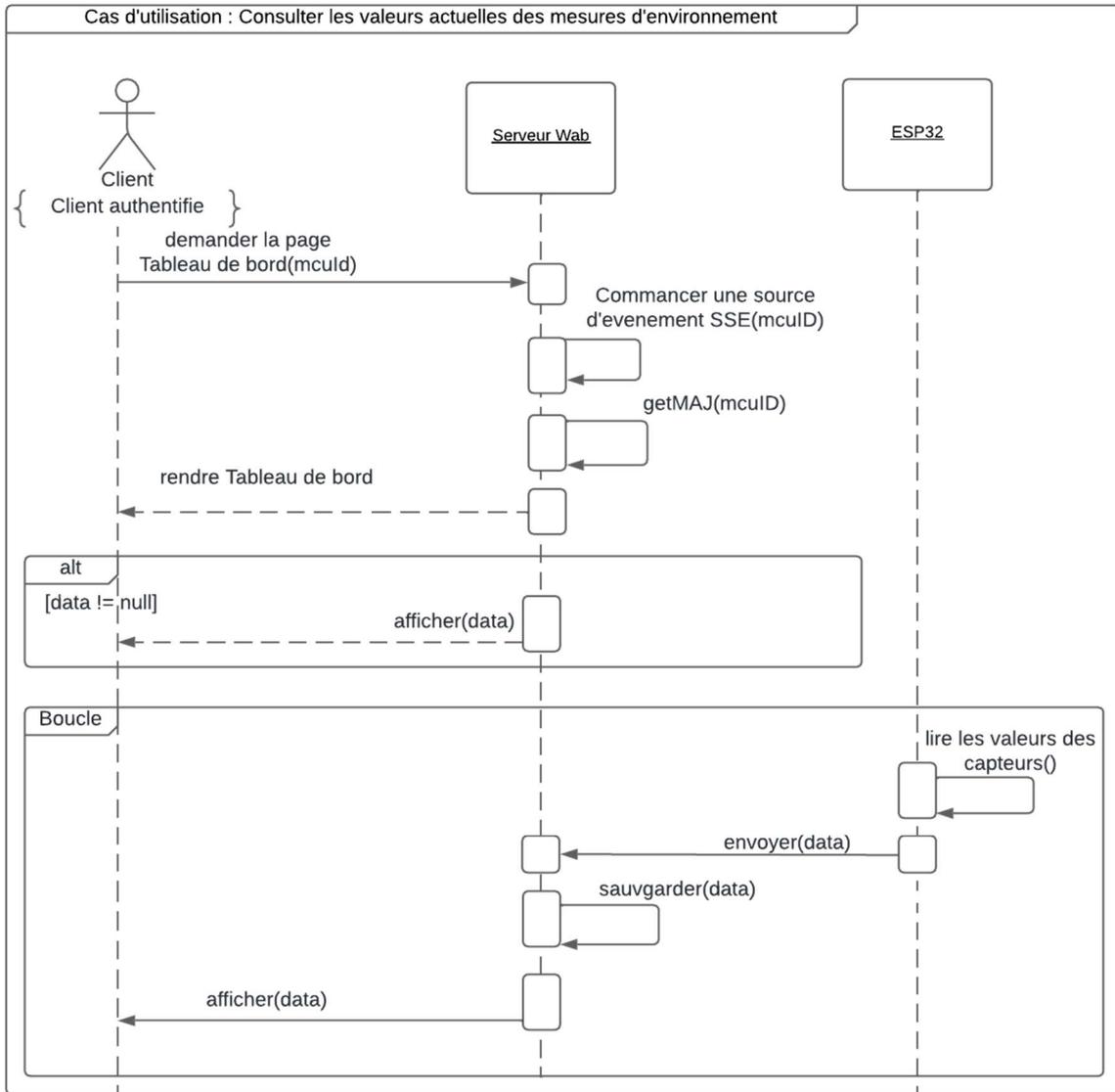


Figure 16 : Diagramme de séquence de Cas d'utilisation Consulter les valeurs actuelles des mesures d'environnement

Diagramme de séquence de Cas d'utilisation Consulter l'historique de changement des valeurs d'environnement dans le temps

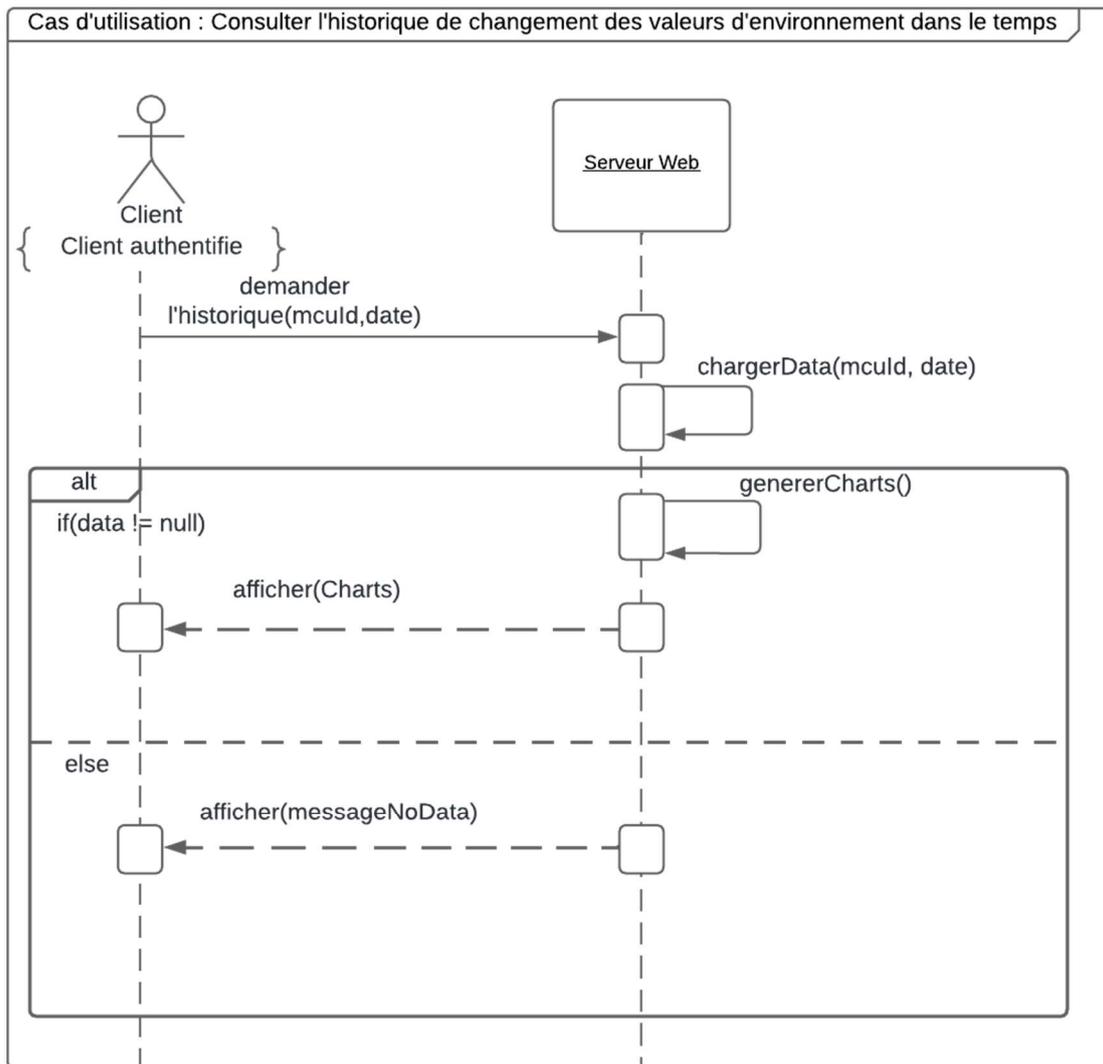


Figure 17 : Diagramme de séquence de Cas d'utilisation Consulter l'historique de changement des valeurs d'environnement dans le temps

Diagramme de séquence de Cas d'utilisation Lancer le processus d'irrigation

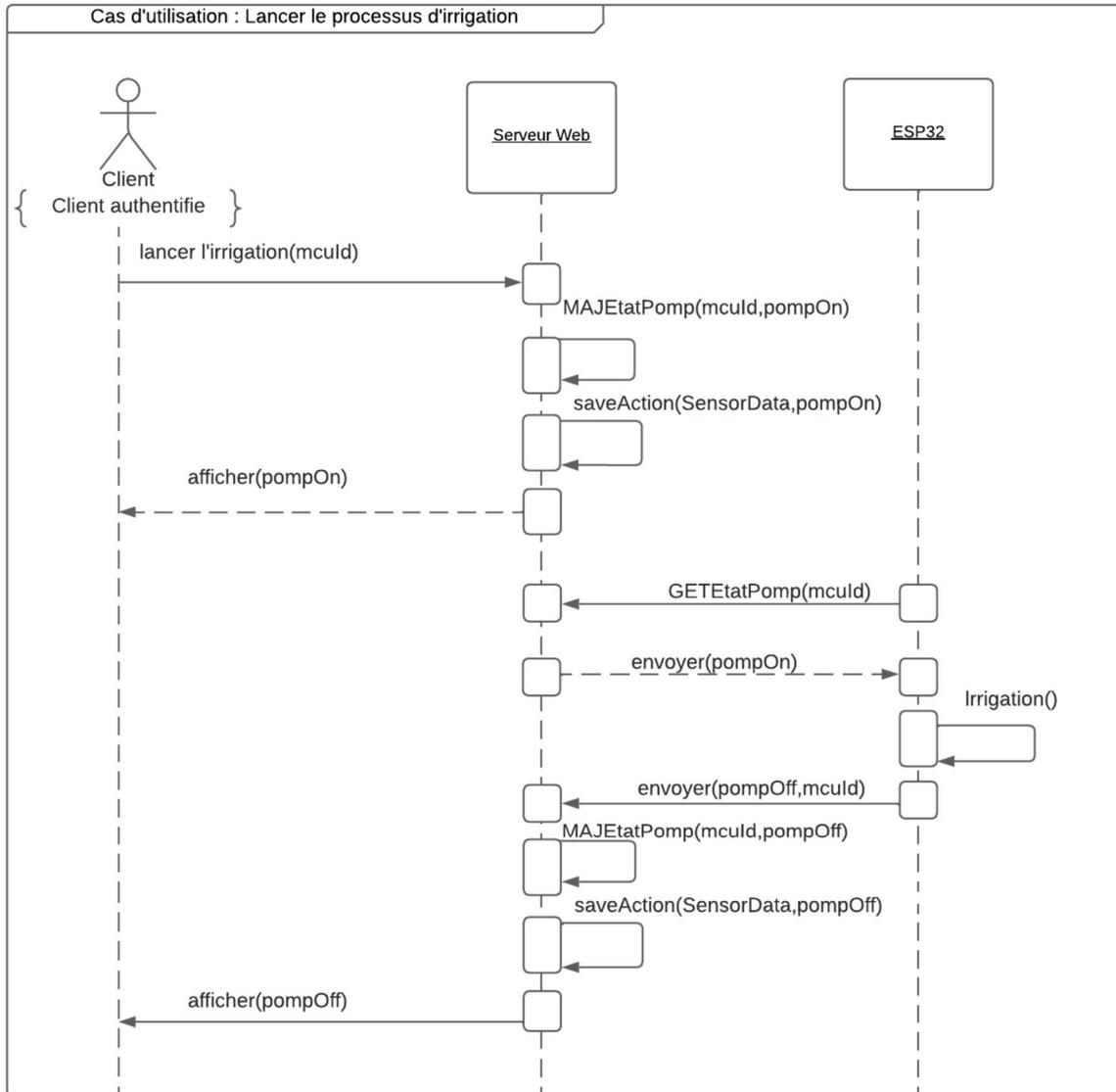


Figure 18 : Diagramme de séquence de Cas d'utilisation Lancer le processus d'irrigation

Diagramme de classe

Le diagramme de classes de notre système d'irrigation se compose de quatre classes principales : **User**, **MCU**, **SensorData**, et **Action**. Chacune de ces classes joue un rôle crucial dans la structure et le fonctionnement du système. La classe **User** représente les utilisateurs du système. La classe **MCU** représente le microcontrôleur central qui gère les opérations du système. La classe **SensorData** encapsule les données collectées par les capteurs du système, tels que l'humidité du sol, la température, et d'autres paramètres environnementaux. Tandis que la classe **Action** définit les actions à entreprendre par le système.

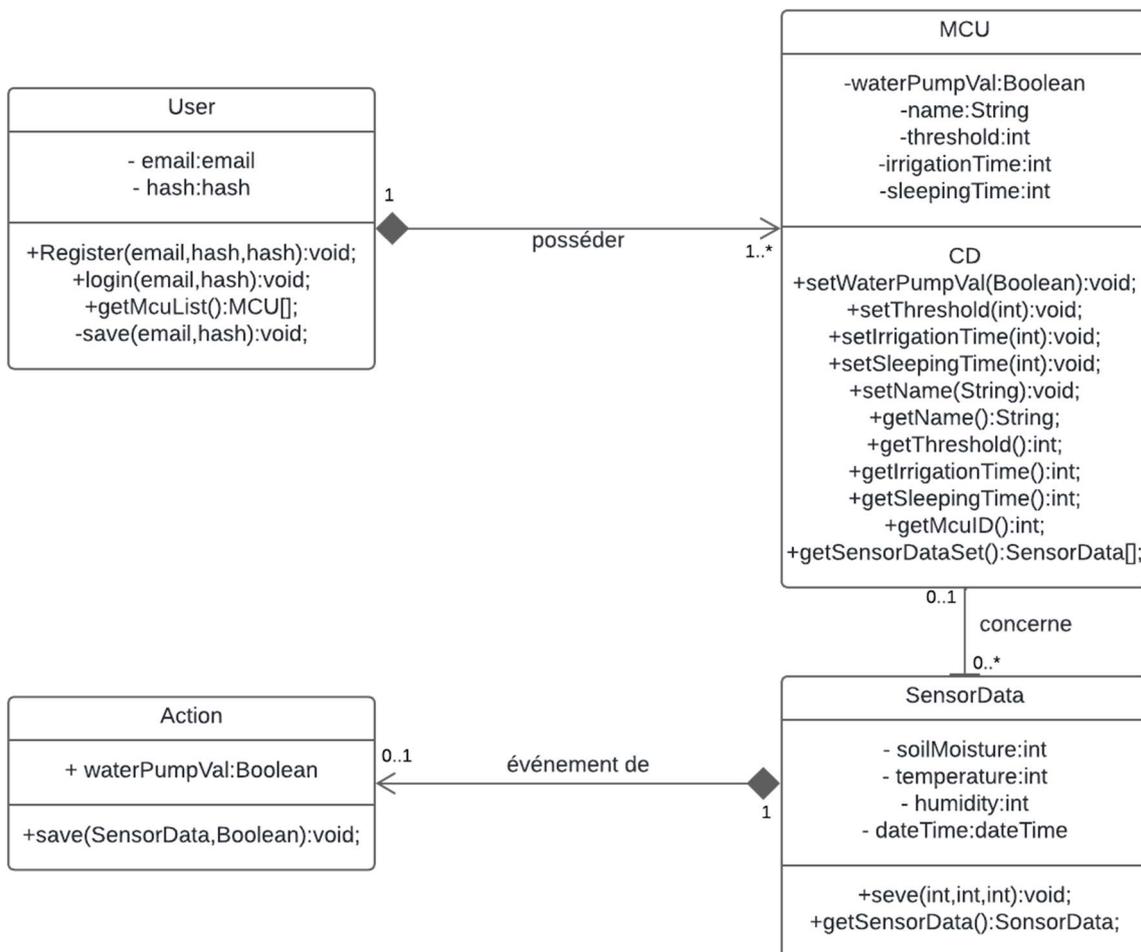


Figure 19 : Diagramme de classe

Les attributs de classes ainsi que les différentes méthodes sont détaillés dans le dictionnaire de données suivant :

		Code	Signification
Class User	Class	User	Contient la logique essentielle pour la gestion des utilisateurs et le système d'authentification permettant le contrôle de leurs systèmes.
	Attributs	email	L'email dans le rôle de nom d'utilisateur.

		hash	Le mot de passe.
	Méthodes	Register	Inscrire un nouvel utilisateur.
		Login	Authentifie le nom d'utilisateur et le mot de passe.
		getMcuList	Récupère la liste des objets MCU gérés par l'utilisateur.
		save	Crée un objet de type User.
Class MCU	Class	MCU	Stocke les objets microcontrôleurs ainsi que leur état interne et permet leur manipulation et contrôle. Elle représente l'état prévu du microcontrôleur sur le serveur.
	Attributs	waterPumpVal	L'état actuel de la pompe à eau (ON ou OFF).
		name	Le nom de l'objet MCU (ou de terrain d'irrigation).
		threshold	Le seuil d'humidité du sol.
		irrigationTime	Le temps maximal avant que l'irrigation ne s'arrête.
		sleepingTime	Le taux de mise à jour des variables des capteurs.
	Méthodes	setWaterPumpVal	Changer l'état de la pompe à eau.
		setThreshold	Modifier le seuil d'humidité du sol.
		setIrrigationTime	Modifier le temps maximal d'irrigation.
		setSleepingTime	Modifier le taux de mise à jour.
		setName	Modifier le nom du module ESP32 (terrain d'irrigation).
		getName	Récupérer le nom du MCU.
		getThreshold	Récupérer le seuil d'humidité.
getIrrigationTime		Récupérer le temps maximal d'irrigation.	
		getSleepingTime	Récupérer le taux d'actualisation.

		getMcuID	Récupérer la référence du MCU (terrain d'irrigation).
		getSensonDataSet	Récupérer la table des données des capteurs pour ce terrain pour une date donnée.
		C	Créer un objet de type MCU.
		D	Supprimer l'objet MCU.
Class SensorData	Class	SensorData	Stocke les données mesurées des capteurs envoyées par le microcontrôleur, ainsi que les dernières mises à jour et les données historiques pour la visualisation des données et les futures analyses.
	Attributs	soilMoisture	L'humidité du sol.
		Temperature	La température de l'air.
		Humidity	L'humidité de l'air.
		dateTime	Date et heure où l'instance SensorData a été fournie par le microcontrôleur.
	Méthodes	save	Sauvegarde une nouvelle instance.
getSensorData		Récupère la dernière actualisation des données des capteurs.	
Class Action	Class	Action	Cette classe est destinée à stocker les actions des utilisateurs pour démarrer ou arrêter le processus d'irrigation, l'exécution du processus par le microcontrôleur, ainsi que les valeurs des capteurs correspondant à l'action. Ces données historiques peuvent être utilisées pour l'entraînement en apprentissage automatique et pour de futures analyses.
	Attributs	waterPumpVal	La valeur fournie pour la pompe (ON ou OFF).

	Méthodes	save	Sauvegarde l'instance action faite pour ces valeurs des capteurs.
--	----------	------	---

Tableau 1 : Dictionnaire de données

Conclusion

Ce chapitre a présenté le modèle conceptuel pour notre système global avec les différents composants, interactions et algorithmes proposés pour le fonctionnement et le développement du système qui suivra dans le chapitre de réalisation suivant.

Chapitre 3 : Réalisation du système

Introduction

Ce chapitre est conçu pour présenter la réalisation de notre système d'irrigation en deux niveaux. Le niveau matériel, qui comporte les différents composants et capteurs. Et le niveau logiciel, avec l'application web développée pour gérer le système.

Réalisation Matériel

La carte de développement ESP-32

La carte de développement NodeMCU-32 à 38 broches ESP32 est un outil extrêmement puissant pour le prototypage rapide de projets de l'Internet des objets (IoT). Il intègre le module S-i-p ESP-WROOM-32, basé sur le Soc ESP32, sur une seule carte.

Cette carte permet l'utilisation de divers protocoles de communication sans fil tels que : Wi-Fi, Bluetooth, BLE (Bluetooth Low Energy).

NodeMCU-32 est spécialement conçu pour être utilisé sur une carte de prototypage.

Il peut être alimenté directement depuis le port micro-USB ou depuis une source externe 5V ou 3V, grâce à un régulateur de tension intégré.

La plateforme ESP32 est compatible avec le développement dans différents langages de programmation, Framework et bibliothèques, tels que :

Arduino (langage C++), Micro Python, Lua, Espied (Espressive IoT Development Framework) ...

Ce microcontrôleur est idéal pour le contrôle de la pompe, la collecte de données de capteur, l'interaction avec le serveur et le contrôle automatique en mode hors ligne.

Une image réelle de ce microcontrôleur est présentée dans la figure 20.

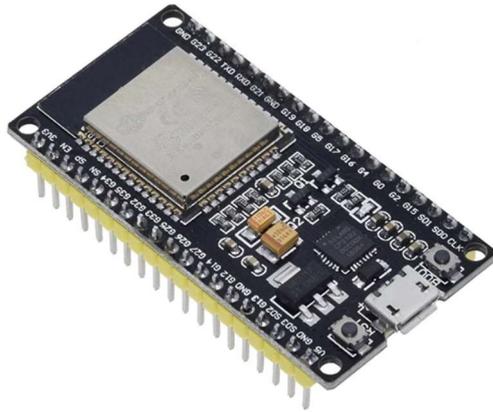


Figure 20 : Carte de développement ESP32 38 broches [26]

SPÉCIFICATIONS TECHNIQUES

Tension d'alimentation (USB) : 5 V DC (avec régulateur de tension)

Puce USB-série : CP2102 (le pilote CP210x nécessaire pour programmer l'ESP32 via USB)

Tension d'entrée/sortie : 3,3 V CC

Carte : ESP32 DEVKIT V2 (Espressive)

Soc : ESP32 (ESP32-D0WDQ6)

Processeur : Tensilica Xtensa LX6 double cœur (32 bits)

Vitesse d'horloge : jusqu'à 240 MHz

Wi-Fi : 802.11 b/g/n/e/l (802.11n @ 2,4 GHz jusqu'à 150 Mbps)

Bluetooth: v4.2 BR/EDR et Bluetooth Low Energy (BLE)

Mémoire : (448 Ko de ROM, 520 Ko de SRAM, 16 Ko de SRAM en RTC, QSPI Flash/SRAM, 4 Mo)

Broches : 38

Broches numériques GPIO : 24 (certaines broches d'entrée uniquement)

Broches PWM : 16

Convertisseur numérique-analogique DAC : 2 (8 bits)

UART : 2

Antenne : sur PCB

Sécurité : normes IEEE 802.11, y compris WFA, WPA/WPA2 et WAPI

Dimensions : 55*28mm

Capteur d'humidité et de température DHT11

Parmi les capteurs de température et d'humidité les plus utilisés pour les systèmes d'irrigation IoT figurent les capteurs d'humidité capacitifs à faible coût de la série DHT, y compris le DHT11. Apprécies pour leur capacité à mesurer simultanément la température et l'humidité, ces capteurs sont disponibles dans des versions plus précises comme le DHT22. D'autres modèles, tels que l'AM2315, offrent une large plage de températures (-40°C à 125°C) et une précision de $\pm 0,1\%$. Cependant, comme le souligne l'étude, les systèmes d'irrigation IoT ne nécessitent généralement pas les plus hauts niveaux de précision du marché. Le DHT11 (voir la figure 21), étant le capteur le plus utilisé dans cette étude, nous l'avons trouvé le plus adapté à notre projet. [23]

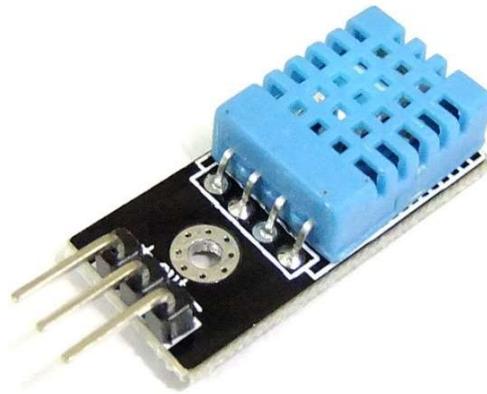


Figure 21 : Capteur d'humidité et de température DHT11 [26]

- **Caractéristiques**

Broches utiles : VCC, GND et Data

Alimentation : +5 V (3,5 – 5,5 V)

Plage de température de fonctionnement : 0 à 50°C

Précision : +/- 2°C

Plage de fonctionnement de l'humidité : 20 % à 96 %

Précision : +/- 5 % HR

- Branchement

-Le branchement de pin signal de DHT11 avec GPIO pin 14 (ou pin 2, 12...) de l'esp32 est défini par la Library qui lire les signales analogues de capteur comme le montre la figure 22.

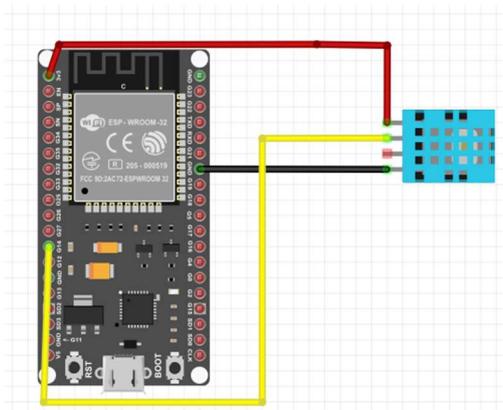


Figure 22 : branchement de capteur DHT11

Capteur capacitif d'humidité du sol

Dans la conception d'un système d'irrigation de précision, le capteur d'humidité du sol joue un rôle crucial. Deux types principaux dominent le marché : capacitif et résistif. Cependant, comme le souligne une étude pertinente, les capteurs capacitifs (voir la figure 23) sont le meilleur choix pour ce projet en raison de leur [22]:

Sensibilité accrue : Ils peuvent détecter même les changements mineurs de l'humidité du sol avec une plus grande précision.

Durée de vie prolongée : Leur conception offre une résistance supérieure à la dégradation, ce qui conduit à un fonctionnement plus long.

Fiabilité améliorée : Ils fournissent des lectures cohérentes et fiables, minimisant ainsi le risque de dysfonctionnement.

Ce capteur fonctionne en générant une sortie analogique allant de 0 à 3 volts. Celle-ci est ensuite convertie par un convertisseur numérique-analogique (DAC) en une valeur numérique généralement comprise entre 0 et 1023 (bien que certains modèles puissent avoir de légères variations). Dans votre programme, vous pouvez ensuite mapper cette valeur numérique sur une échelle conviviale de 0 à 100 %, représentant le pourcentage d'humidité du sol.



Figure 23 : Capteur capacitif d'humidité du sol [26]

- **Caractéristiques**

Broches : Trois broches - Sortie analogique, VCC (alimentation) et GND (masse)

Précision : $\pm 3 \%$

Température de fonctionnement : -40°C à 85°C

Dimensions : 99 mm x 16 mm

- **Branchement**

-Le connexion de l'esp ADC pin 32 avec l'analogique pin de capteur est pour éviter le conflit avec les pin active lors d'utilisation du carte wifi.

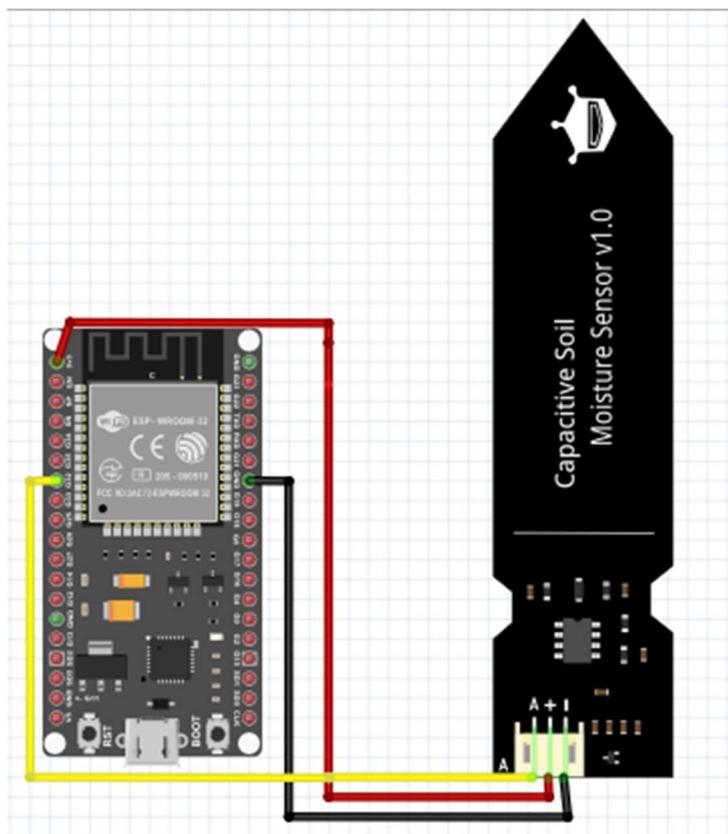


Figure 24 : branchement de capteur capacitif d'humidité de sol

Module de relais de déclenchement de bas niveau

Le relais est un interrupteur contrôlable qui isole la partie commande de la partie alimentation. Il est crucial de contrôler les actionneurs d'un système qui dépendent d'une source d'énergie externe différente des 3,3 V de l'ESP, comme c'est le cas pour une pompe à eau (qui fonctionne entre 3,5 et 6 V dans notre cas) et il est même compatible avec une pompe à eau de tension plus élevée (220 V), une image réelle de ce module est présentée sous la figure 25.



Figure 25 : Module de relais 5VDC [26]

- **Caractéristiques**

Charge maximale : AC 250V/10A, DC 30V/10A

Courant de déclenchement : 5 mA

Tension de fonctionnement : 5 V

Taille du module : 50x26x18,5 mm (L x l x H)

Trois trous de fixation, diamètre 3,1 mm

Interfaces :

DC+ : Alimentation positive (VCC)

DC : Alimentation négative (GND)

IN : peut-être un relais de contrôle de haut niveau ou de bas niveau

NO : Interface de relais normalement ouverte

COM : Interface de relais commune

NC : Interface de relais normalement fermée

- **Branchement**

La connexion de l'esp pin 17 avec le pin de signal dans le relais est aussi pour éviter le conflit avec les pin active lors d'utilisation du carte wifi comme le montre la figure 26.

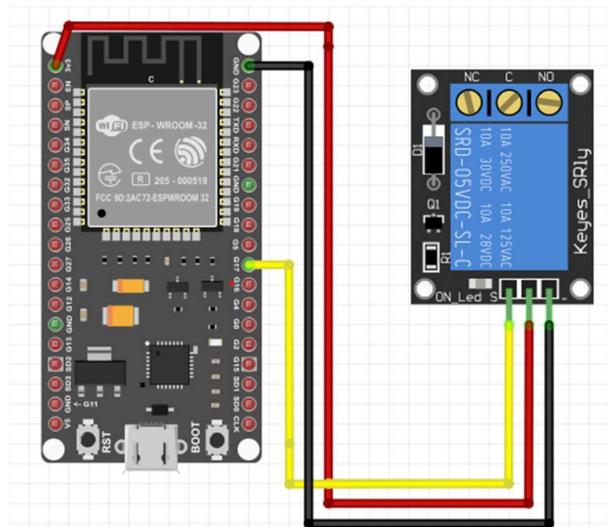


Figure 26 : branchement de Module Relais

Mini pompe à eau submersible 3-6V

Une mini pompe à eau est contrôlée par le système via le microcontrôleur à l'aide du module de relais (voir l'image réelle sous la figure 27).



Figure 27 : Mini pompe à eau submersible 3-6V [26]

- **Caractéristiques**

Tension : DC 3-6V

Courant : 100-200mA

Levée maximale : 30 - 80 cm / 11,8 - 31,5 pouces

Débit : 1,2-1,6 L/minute

Diamètre intérieur de l'entrée d'eau : 5 mm

Diamètre intérieur de la sortie d'eau : 4,5 mm

Diamètre extérieur : 7,5 mm

- Branchement illustré dans la figure 28:
 - Le relais ouvre et coupe le circuit de la pompe et l'alimentation externe, selon le contrôle de l'ESP.
 - L'alimentation de 3.7V pour éviter le problème de grounds commune qui va interrompre le travail des capteurs si on utilise 7.4V .

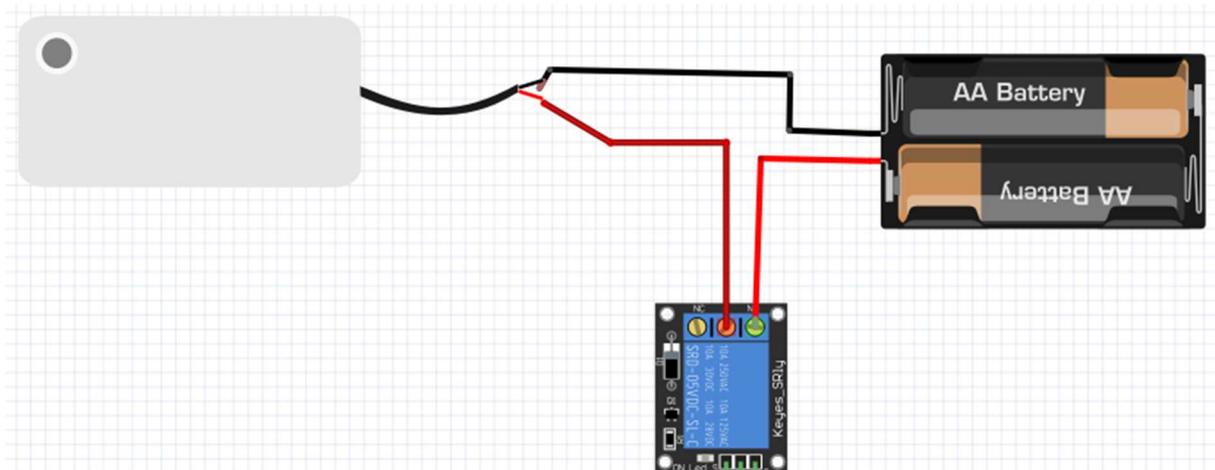
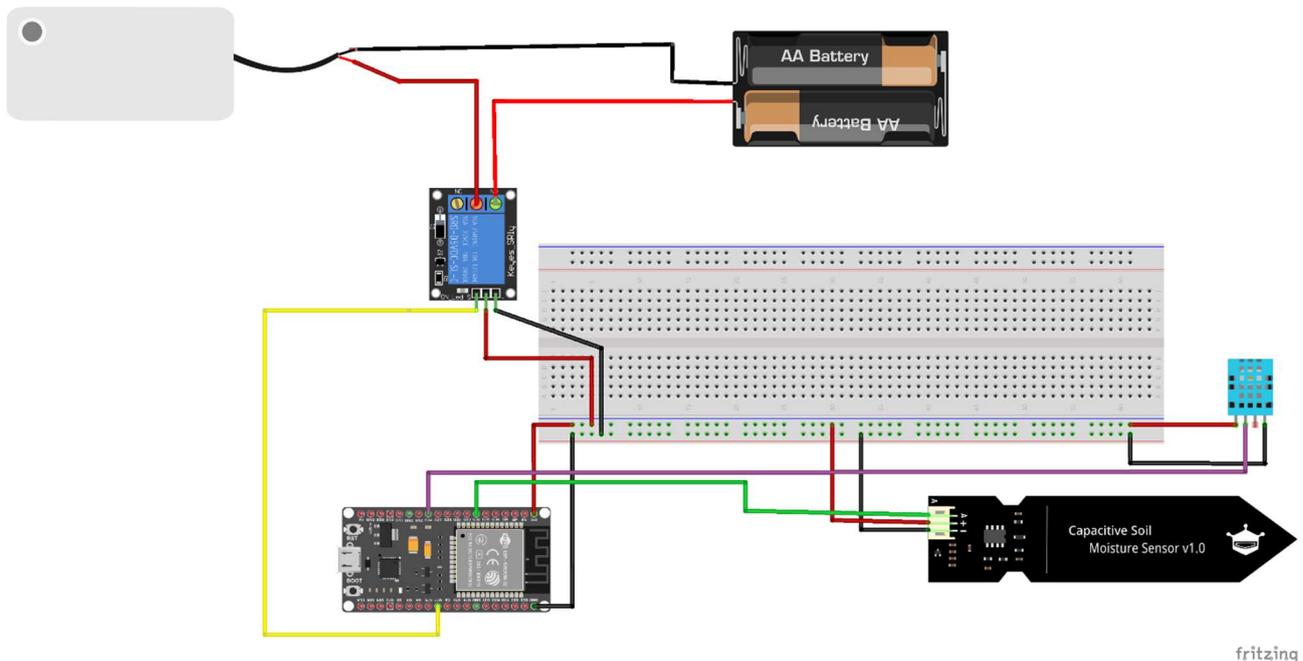


Figure 28 : Branchement du mini pompe à eau

Branchement global de système

En combinant tous les aspects du branchement des différents composants, le branchement global est montré dans la figure 29.



fritzing

Figure 29 : Le branchement global de système avec les différents composants

Développement de la carte Esp 32

- **Arduino IDE :**

L'IDE Arduino est un environnement de développement open source basé sur le Framework Arduino, qui permet de programmer des microcontrôleurs avec des fichiers d'extension (.ino) en langage C++ et de les téléverser sur le Firmware de l'ESP32. Il facilite également l'intégration des bibliothèques nécessaires pour d'autres composants.

- **Bibliothèques utiliser :**

- **HTTPClient.h** : une bibliothèque efficace d'Arduino permettant à l'ESP32 d'échanger des requêtes HTTP avec le serveur en utilisant des données au format JSON pour les mesures des capteurs et pour synchroniser les variables d'état internes de l'ESP.
- **DHT.h** : une bibliothèque de capteurs unifiée d'Adafruit qui permet de capturer et de convertir les mesures analogiques du DHT11 en un format approprié.
- **Wifi. H** : contient la logique nécessaire pour créer et connecter un client Wifi basé sur le module Wifi de l'ESP32.
- **ArduinoJson.h** : fournit les fonctions essentielles pour la sérialisation et la désérialisation des données échangées avec le serveur.

Une partie de l'implémentation du module ESP32 pour le fonctionnement de l'algorithme d'irrigation proposé est illustrée dans la figure 30.

```

146 void irrigation(){
147     //calculer les valeurs
148     int soilMoisture = 0;
149     soilMoisture = map(analogRead(sensorPin), wetValue, dryValue , 100, 0);
150     float temperature = dht.readTemperature();
151
152     if((soilMoisture <= threshold )&&(temperature > 0)){ digitalWrite(RELAY_PIN, LOW);//on
153     unsigned long lastTime = millis();
154     while((soilMoisture < threshold )&&(((irrigationTime*1000) + lastTime) > millis())&&(temperature > 0)){
155         //for the soil sensor to read again
156         delay(2000);
157         temperature = dht.readTemperature();
158         if (isnan(temperature)) {
159             Serial.println("Failed to read from DHT sensor!");
160             temperature = -1;}
161         Serial.println("Irrigation loop");
162         soilMoisture = map(analogRead(sensorPin), wetValue, dryValue , 100, 0);
163     }
164     digitalWrite(RELAY_PIN, HIGH);//off
165     if((WiFi.status()== WL_CONNECTED) && (mcuID != 0)){
166         http.begin(url + "api/mcu/" + String(mcuID)+"/");
167         http.addHeader("Content-Type", "application/json");
168         String jsonOutput;
169         const size_t CAPACITY = JSON_OBJECT_SIZE(1);
170         StaticJsonDocument<CAPACITY> doc;
171         JsonObject object = doc.to<JsonObject>();
172         object["waterPumpVal"]=false;
173         serializeJson(doc, jsonOutput);
174         Serial.println(String(jsonOutput));
175         int code = http.PUT(String(jsonOutput));
176         if(code > 0){

```

Output Serial Monitor x

Figure 30 : partie du code de processus Irrigation

Réalisation logicielle :

Environnement et technologies de développement :

- **Python :**

Python est un langage de programmation très polyvalent et facile à lire.

Il permet différents styles de programmation comme procédural, orienté objet et fonctionnel. Il est utilisé beaucoup dans le développement web, la science des données, l'intelligence artificielle et l'automatisation. Ses nombreuses bibliothèques et Framework en font un outil puissant pour les développeurs.

- **Django :**

Django est un Framework web en Python qui permet de développer rapidement des applications avec un design propre.

Il suit le modèle MTV (modèle-Template-vue) et intègre un ORM (Object Relationnel Model) pour interagir avec les bases de données.

Django offre des fonctionnalités prêtes à l'emploi comme l'authentification, une interface d'administration et la gestion des formulaires.

Avec une communauté active et une documentation complète, Django est très apprécié des développeurs. [24]

- MTV (modèle-Template-vue) :

L'architecture de Django diffère légèrement du modèle MVC traditionnel. La particularité de Django est qu'il gère automatiquement la partie contrôleur (gestion des requêtes des clients, des permissions sur les actions, etc.). Ainsi, on parle plutôt d'une architecture MVT : Modèle-Vue-Template illustré dans la figure 31.

Cette architecture conserve les notions de modèle et de vue, tout en introduisant un nouvel élément : le Template. Un Template, ou "gabarit" en français, est un fichier HTML qui sera récupéré par la vue et envoyé au visiteur. Avant cet envoi, le Template est analysé et exécuté par le Framework, semblable à un fichier contenant du code. Django offre un moteur de Template puissant, permettant d'afficher des variables dans le code HTML, d'utiliser des structures conditionnelles (if/else), des boucles (for), etc. [24].

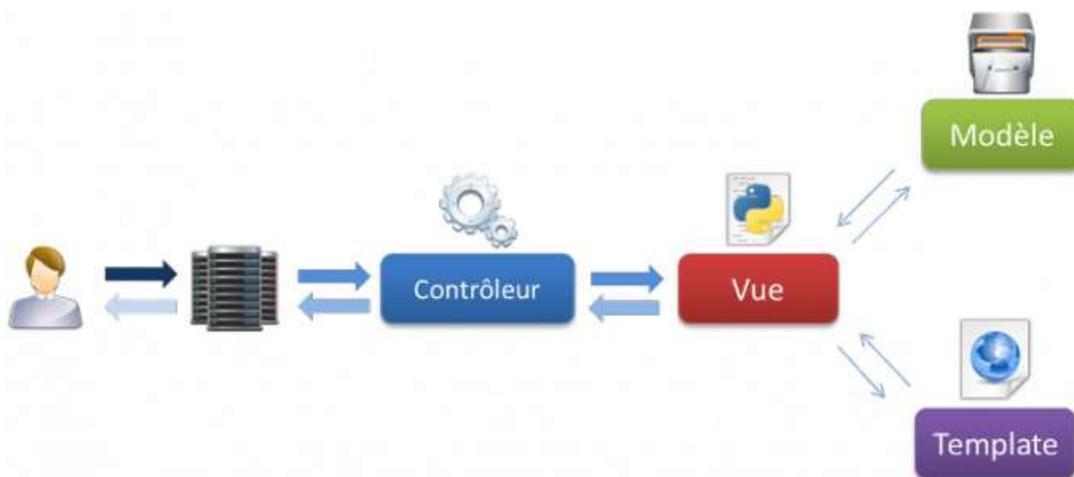


Figure 31 : Architecture MVT [24]

- Object Relationnel Model :

Un modèle, est défini sous forme de classe où chaque attribut représente un champ de la base de données. Django se charge de créer la table correspondante dans la base de données et de gérer la liaison entre la base et les objets de la classe. Ainsi, il n'est plus nécessaire d'écrire des requêtes SQL pour interagir avec la base de données. Le Framework permet de représenter chaque entrée de la table comme une instance de la classe, ce qui facilite grandement l'accès et la modification des données.

- Django REST Framework :

Django REST Framework est un outil puissant pour créer des API web avec Django.

Il propose des fonctionnalités comme la sérialisation, l'authentification et les « viewsets », simplifiant ainsi le développement d'API.

Il permet de créer des solutions d'API flexibles et robustes, en offrant des comportements standards et personnalisés.

Avec une documentation riche et une communauté active, il est le choix privilégié pour les API RESTful.

- **React :**

React Native est une bibliothèque JavaScript développée par Facebook qui permet de créer des applications mobiles pour iOS et Android en utilisant une base de code unique. En tirant parti de composants natifs, React Native offre des performances comparables à celles des applications entièrement natives. Les fonctionnalités comme le Hot Reloading facilitent un développement rapide et efficace, permettant aux développeurs de voir instantanément les modifications apportées au code. Des applications populaires comme Facebook, Instagram, Airbnb, et Skype ont été créées avec React Native, démontrant sa robustesse et sa fiabilité. Avec une communauté active et une vaste bibliothèque de modules, React Native simplifie l'intégration des API tierces et des fonctionnalités complexes. Bien que certaines performances puissent être légèrement en deçà des applications 100% natives pour des tâches spécifiques, les avantages en termes de réduction des coûts et de temps de développement sont significatifs. En somme, React Native est une solution puissante et flexible pour le développement d'applications mobiles modernes.

- **Vite.js :**

Vite est un outil de développement, la communauté React bénéficie d'une expérience de développement encore plus fluide et efficace. Vite se démarque par sa compilation à la volée et son chargement de modules ESM (ECMAScript Modules), offrant des temps de démarrage ultra-rapides et une recompilation instantanée.

- **JavaScript :**

JavaScript est un langage de programmation qui apporte la logique aux pages web.

Il permet des actions côté client, comme le prétraitement des données et la validation de formulaires et le chargement dynamique de contenu.

- **Bootstrap :**

Bootstrap est un Framework frontend très utilisé pour créer des sites web réactifs et mobiles.

Il offre des composants CSS et JavaScript comme des grilles, des boutons et des barres de navigation.

Grâce à ses modèles préconçus et ses nombreuses options de personnalisation, Bootstrap accélère le développement web.

- **Visual Studio Code :**

Visual Studio Code est un éditeur de code léger et open-source créé par Microsoft.

Il supporte de nombreux langages de programmation et offre des fonctionnalités comme le débogage, le contrôle de version et les extensions.

L'interface personnalisable et l'écosystème riche de VS Code en font un outil très populaire parmi les développeurs.

- **Web Socket:**

WebSocket est une technologie de communication bidirectionnelle en temps réel qui permet aux navigateurs web et aux serveurs d'établir une connexion persistante et à faible latence. En utilisant WebSocket, les applications web peuvent envoyer et recevoir des données instantanément, ce qui les rend idéales pour les fonctionnalités en temps réel telles que les chats en ligne, les mises à jour en direct et les jeux multijoueurs. Avec WebSocket, les développeurs peuvent créer des applications web interactives et dynamiques qui offrent une expérience utilisateur immersive et réactive. En permettant une communication fluide entre le client et le serveur, WebSocket simplifie la mise en œuvre de fonctionnalités complexes et améliore les performances des applications web modernes. En combinant la puissance de WebSocket avec d'autres technologies web telles que React et Vite, les développeurs peuvent créer des applications web hautement performantes et évolutives pour répondre aux besoins changeants des utilisateurs.

- **Protocole HTTP :**

HTTP (HyperText Transfer Protocol) est le protocole utilisé pour transférer des données sur le web.

Il définit les méthodes de communication entre les navigateurs web et les serveurs, comme les méthodes GET, POST, PUT et DELETE.

HTTP est la base de l'échange d'informations sur le web, assurant la connexion entre les utilisateurs et les sites web.

Présentation des applications

Au cours de la réalisation de notre projet, nous avons effectué deux prototypes utilisant des technologies différentes. Nous allons présenter chacun des prototypes et montrer les avantages de chacun.

Prototype 1

Dans un premier temps, pour assurer le contrôle à distance et le suivi en temps réel des valeurs physiques et des composants du système, tout en garantissant l'accès, la modifiabilité et la fiabilité du système d'irrigation, nous présentons l'application web IOSeeds. Développée pour répondre aux besoins fonctionnels et non fonctionnels, elle est accessible sur toutes les plateformes et s'adapte à toutes les tailles d'écran, garantissant une expérience utilisateur optimale.

Technologies

- Django REST Framework, HTML, CSS, JavaScript : Utilisés pour construire et interagir avec l'application web.
- ESP32 : Module matériel interagissant avec le système via le protocole HTTP.

Objectifs

- Mise en œuvre de l'algorithme d'irrigation : Développer et implémenter l'algorithme d'irrigation proposé pour assurer la fiabilité et la disponibilité du système d'irrigation, même en cas d'absence d'une ou plusieurs parties du système.
- Interaction avec le système matériel : Assurer une communication efficace avec le matériel réalisé pour garantir un contrôle et une surveillance optimaux du processus d'irrigation.

Connecter et inscrire

Pour utiliser le système, l'utilisateur doit d'abord s'authentifier. Ou Il se connecte ou s'inscrit sous l'interface illustrée dans la figure 32. Dans le processus de création de nouveaux utilisateurs, une nouvelle instance de type MCU (microcontrôleur) sera créée. Le système d'authentification est basé sur le module « django.contrib.auth » du Framework Django.

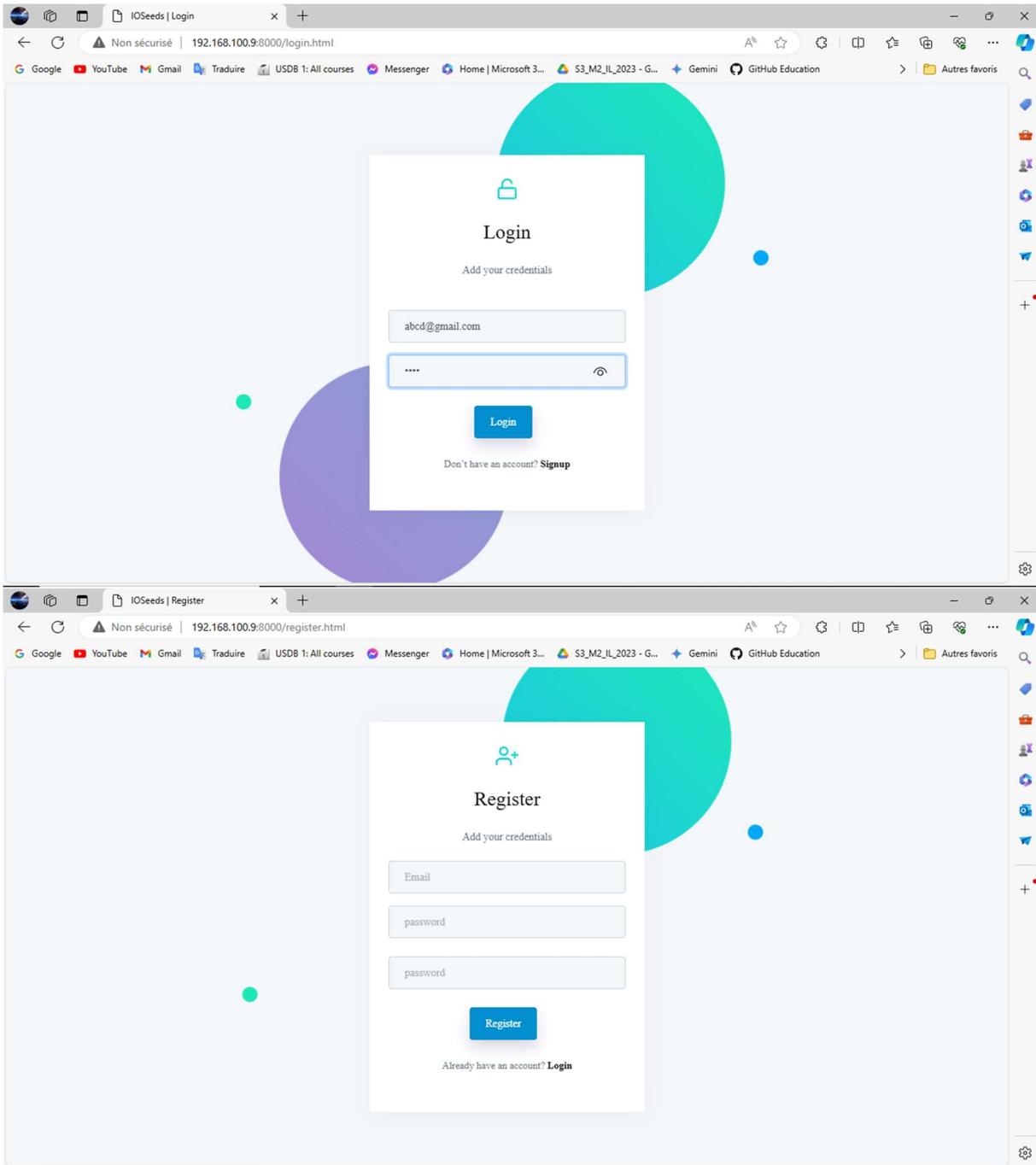


Figure 32 : Connecter et inscrire

Tableau de bord

Après l'authentification et l'initialisation du microcontrôleur, le tableau de bord présente les valeurs des mesures de trois capteurs : l'humidité et la température de l'air, l'humidité du sol, ainsi que l'état actuel de la pompe, avec un taux d'actualisation défini par l'utilisateur (par exemple, toutes les 5, 10 ou 60 secondes...). Il permet également le contrôle à distance de la pompe avec les interfaces sous la figure 33.

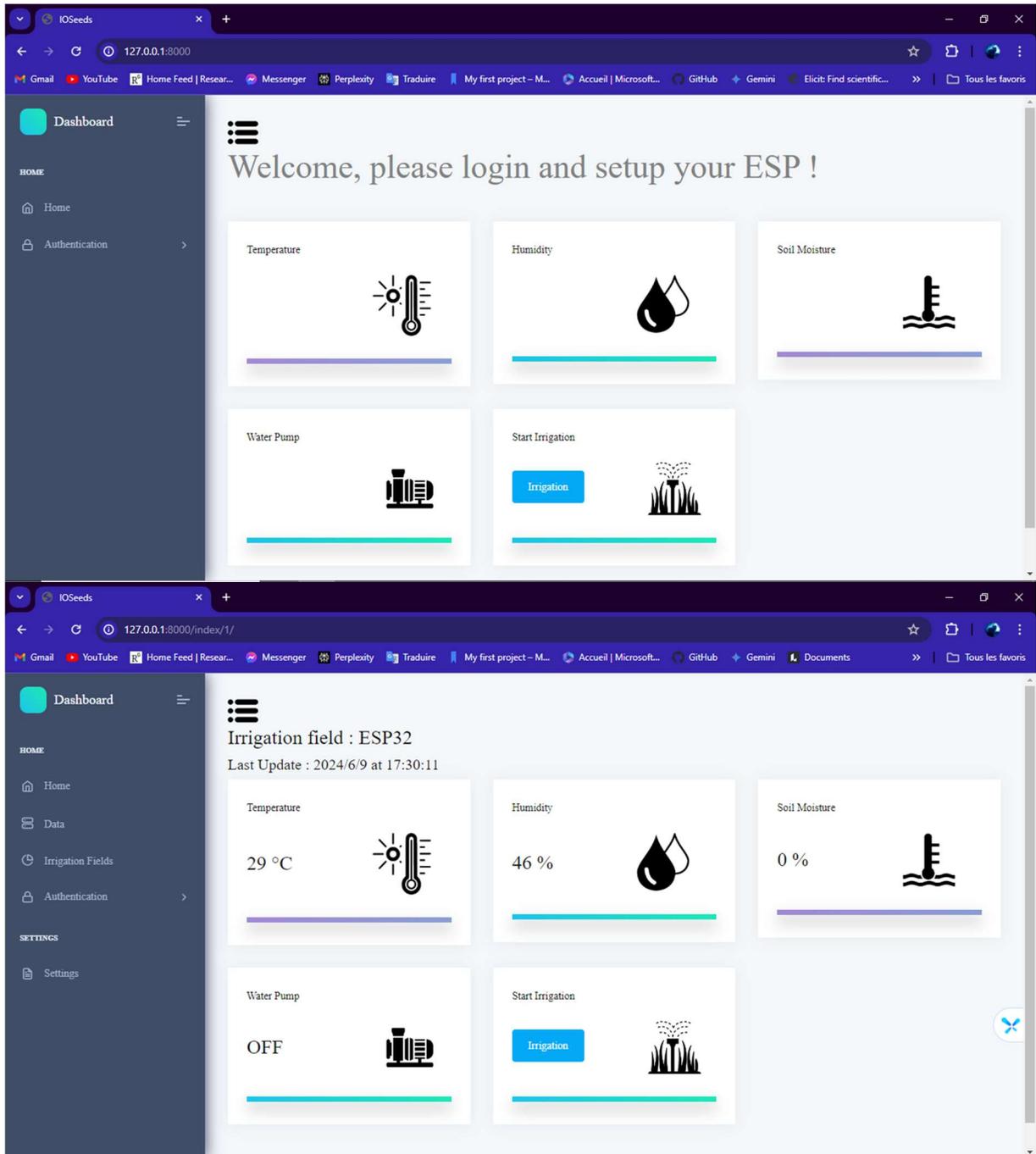


Figure 33 : page de Tableau de bord

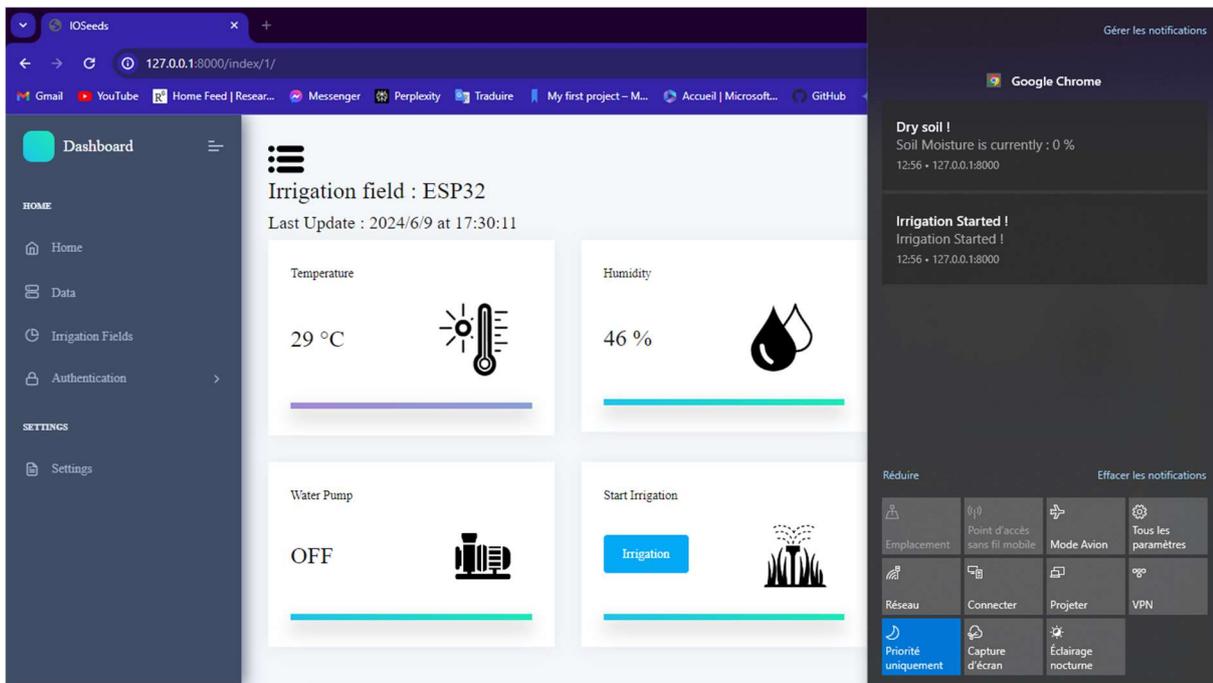


Figure 34 : système de notification

- Ce système permet aux utilisateurs d'être notifiés si l'humidité du sol est inférieure à la valeur qu'ils ont définie et si l'irrigation a été activée comme le montre la figure 34.
- Cette fonctionnalité est assurée grâce à la technologie SSE (Server-Sent Events) pour garantir la diffusion en temps réel des nouvelles valeurs du système aux utilisateurs.

Historique des données

Cette page permet de consulter l'historique des valeurs du système pour le microcontrôleur actuel et une date sélectionnée sous forme de graphiques de données pour suivre l'évolution de l'état du système dans le temps sous l'interface illustrée dans la figure 35.

Les graphiques sont créés à l'aide de la bibliothèque « Plot-Ly » du langage Python qui offre les fonctionnalités pour créer des graphiques de données interactifs et conviviaux.

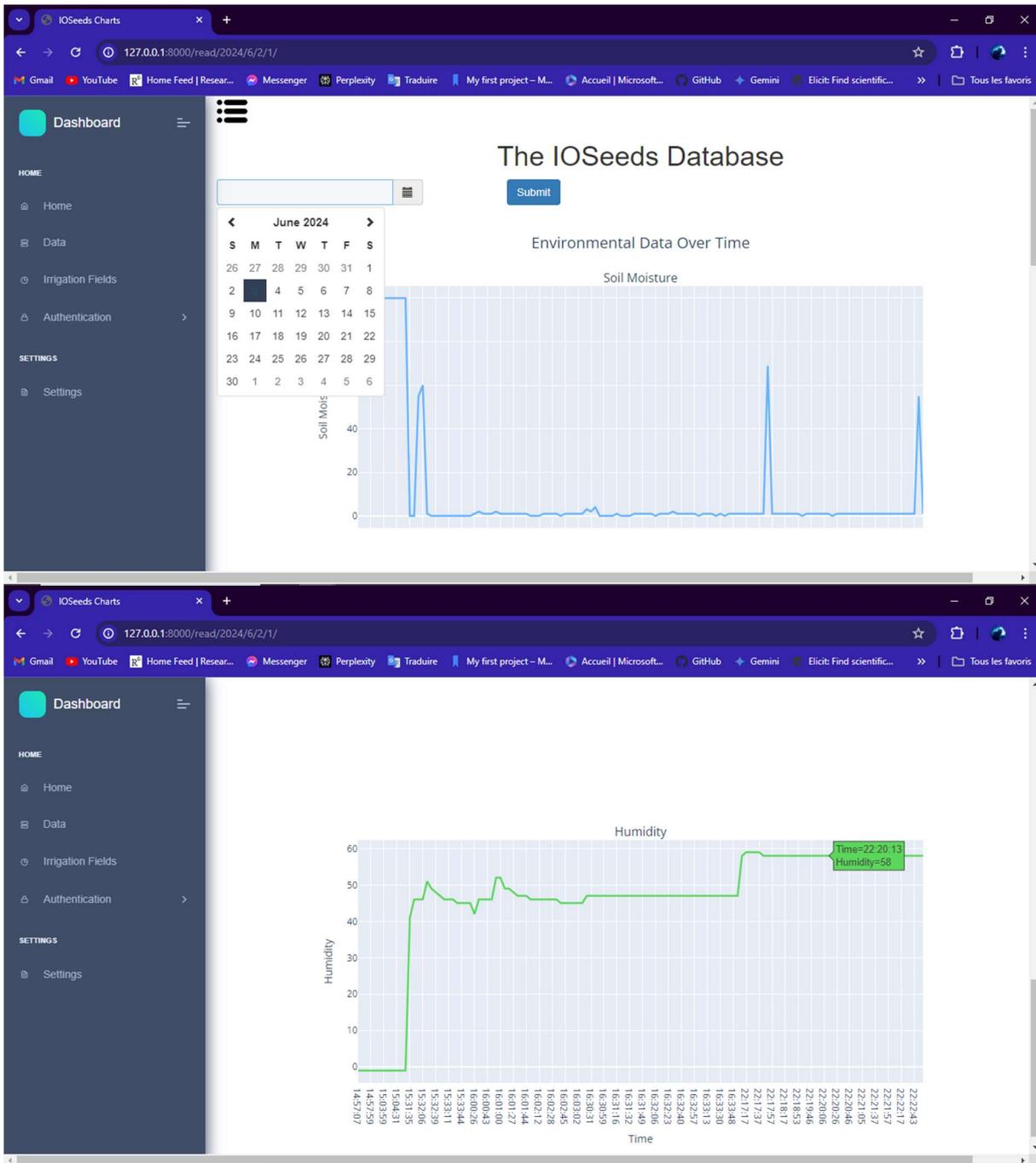


Figure 35 : page d'historique des données

Les terrains d'irrigation

L'utilisateur peut gérer plusieurs terrains d'irrigation présentés par l'entité microcontrôleur. Cette page permet la sélection, l'ajout et la suppression de ces derniers (voir la figure 36).

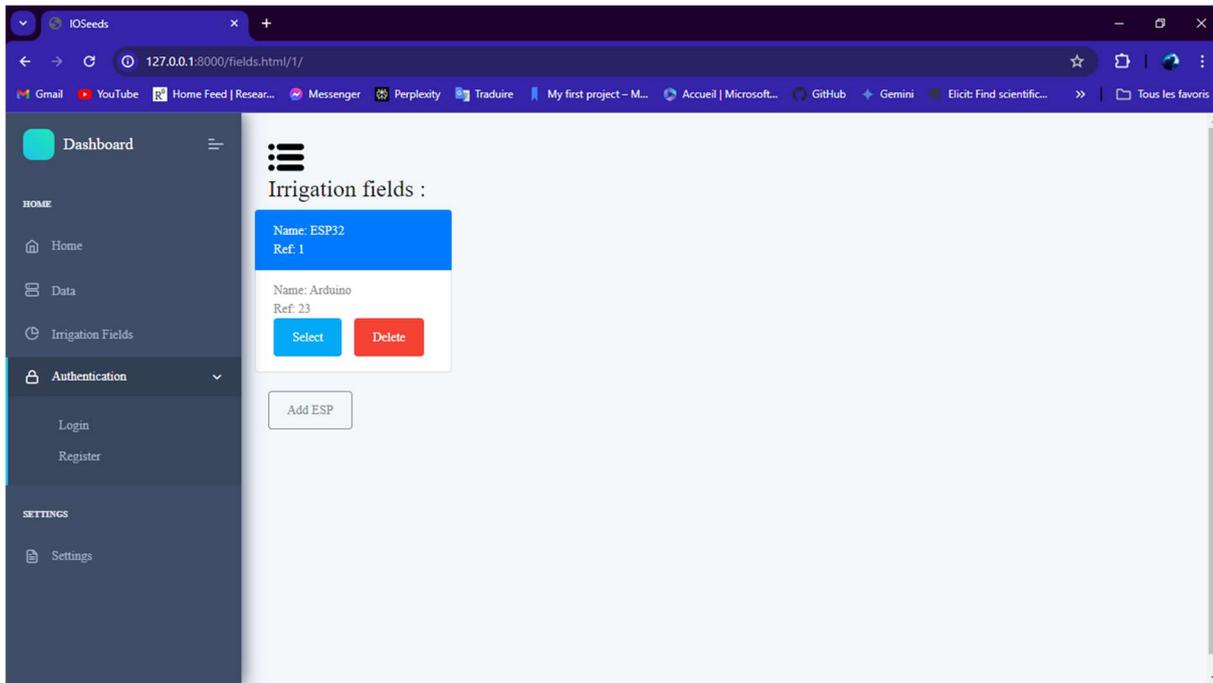


Figure 36 : page des terrains d'irrigation

Modification des valeurs d'état du microcontrôleur

Pour satisfaire les besoins du système proposé, la page sous la figure 37 des paramètres permet de vérifier et de modifier les valeurs d'état interne du microcontrôleur, qui sont :

- Le taux d'actualisation des valeurs des mesures physiques par l'ESP
- Le nom de l'ESP pour distinguer facilement entre les terrains d'irrigation
- Le temps maximal d'irrigation utilisé par l'ESP lors de l'exécution de l'algorithme d'irrigation proposé
- Le seuil d'humidité du sol pour le système de notification et pour le processus d'irrigation

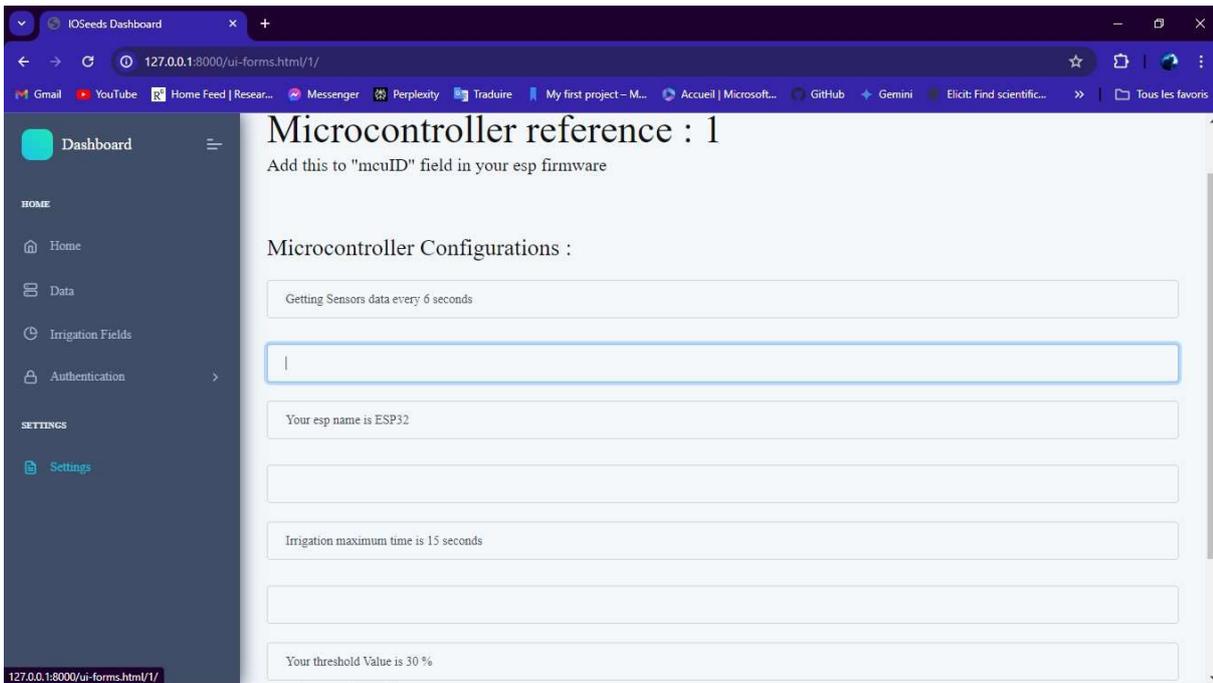


Figure 37 : page des paramètres

Interface mobile

L'intégrité du système sur toutes les plateformes et les tailles d'écran est possible avec notre application web, pour que l'application soit accessible par tous avec une bonne expérience utilisateur ,Un exemple d'utilisation avec l'interface mobile est illustré sous la figure 38.

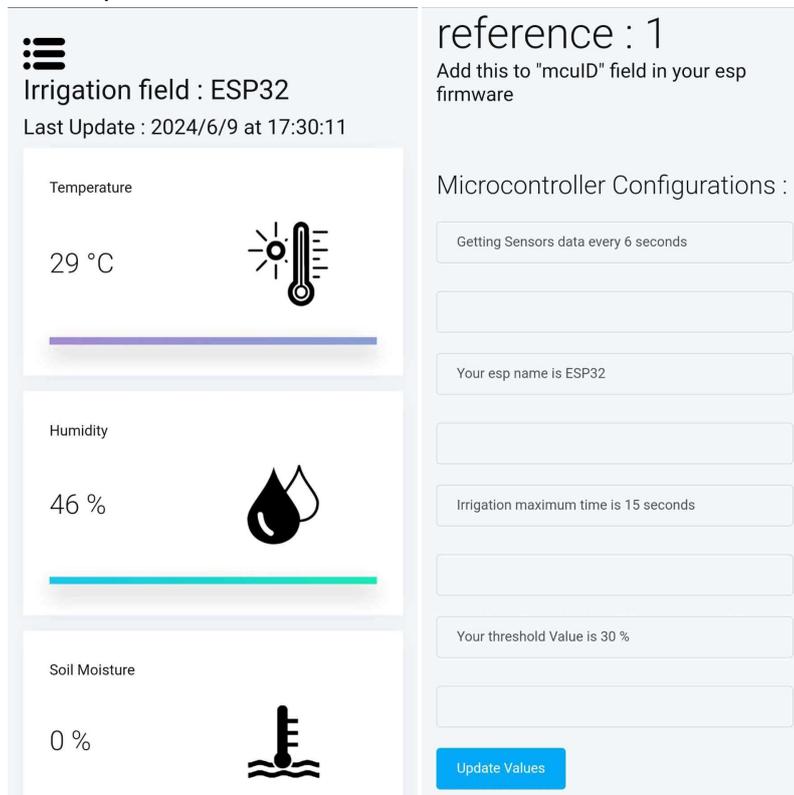


Figure 38 : les pages dans l'interface mobile

Prototype 2

Dans un second temps Ce prototype assure la supervision et le contrôle à distance des capteurs et des actionneurs en temps réel, en appliquant une connexion ouverte avec le module ESP32 via la technologie Web Socket.

Technologies

- Django REST Framework, React, Vite.js : Utilisés pour construire et interagir avec l'application web.
- ESP32 : Module matériel interagissant avec le système via une connexion Web Socket ouverte.

Objectifs

- Supervision et contrôle total : Permettre à l'utilisateur de superviser et de contrôler entièrement le système à distance.
- Scalabilité du système : Faciliter la modification matérielle des composants IoT utilisés par le système d'irrigation, notamment l'ajout de nouveaux capteurs et actionneurs selon les besoins.

Tableau de bord

Une fois l'application lancée, vous serez immédiatement redirigé vers le tableau de bord principal, offrant un accès centralisé à tous les microcontrôleurs, capteurs et appareils connectés. Ce tableau de bord affiche l'état actuel de chaque microcontrôleur, offrant ainsi une vue d'ensemble rapide de la santé de votre système. De plus, vous aurez la possibilité de changer le thème de l'application selon vos préférences personnelles, offrant ainsi une expérience utilisateur personnalisée et adaptée à vos besoins est exprimée dans la figure ci-dessous (voir Figure39).

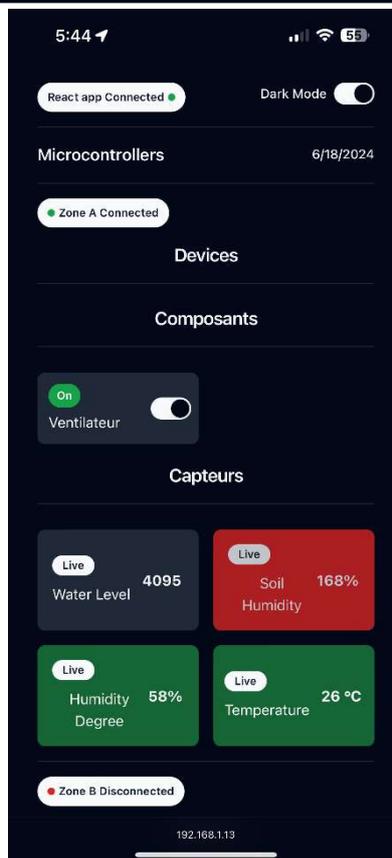
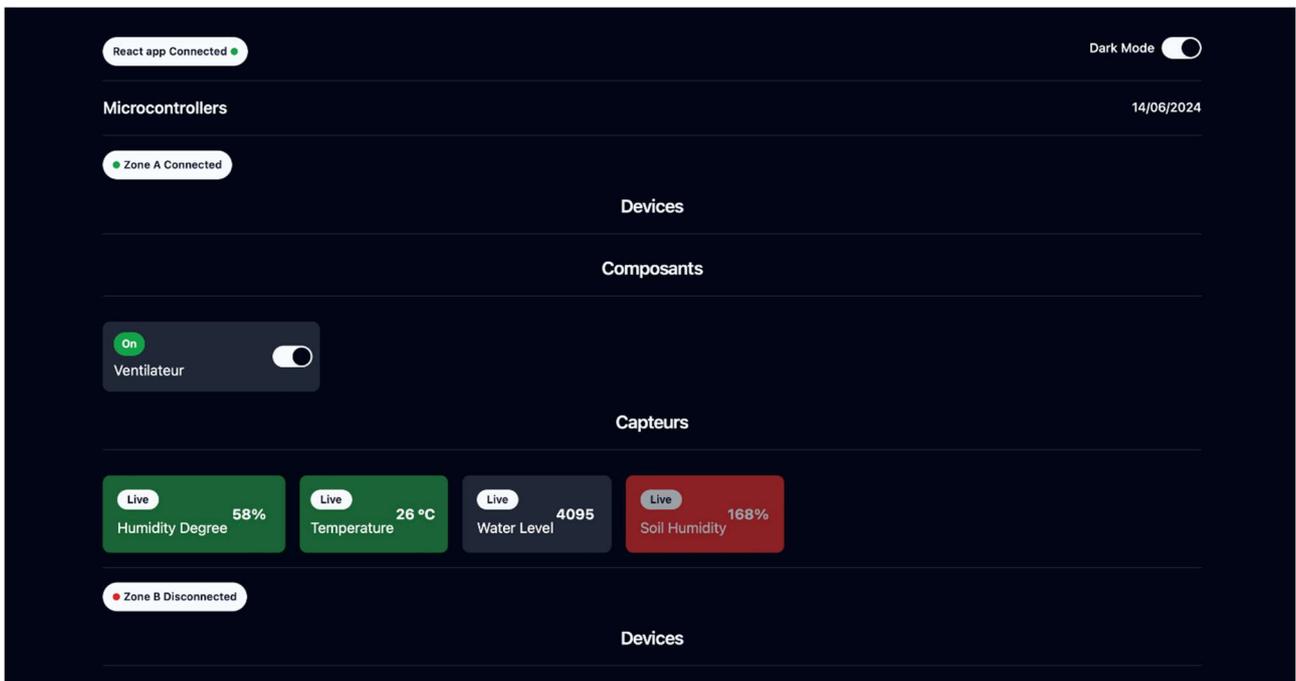


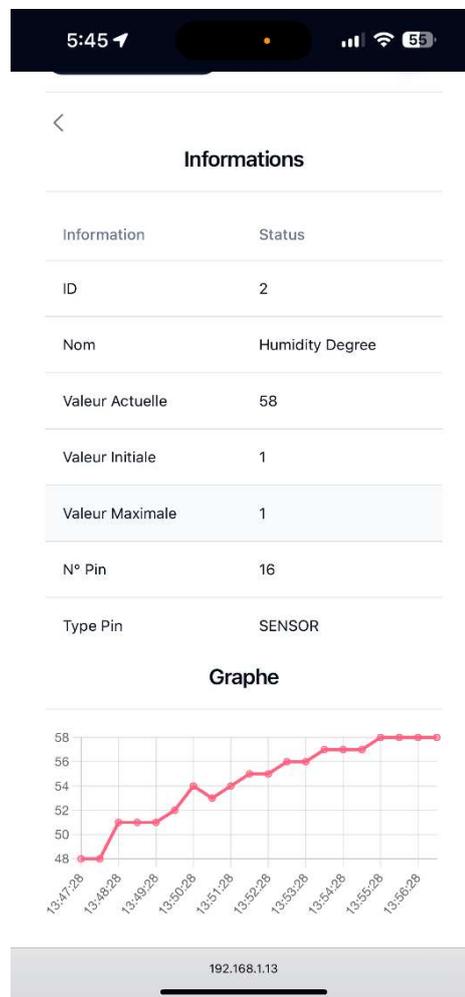
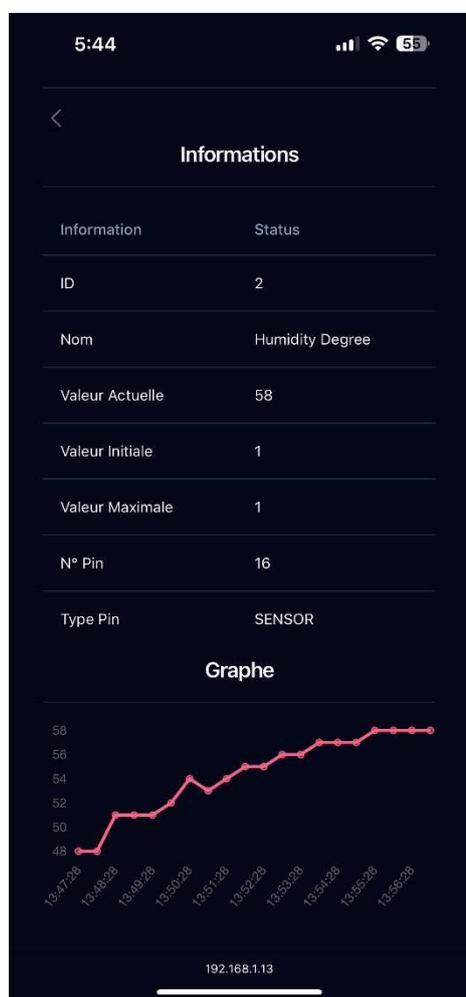
Figure 39 : Tableau de Bord Application React

Une fois dans l'interface de Django Admin, vous pouvez modifier les paramètres des broches, y compris leur numéro, leur état et d'autres configurations pertinentes, en fournissant ainsi un contrôle complet sur la configuration matérielle de votre système. Cette intégration transparente

vous permet de gérer efficacement votre infrastructure IoT, en combinant la flexibilité de Django Admin avec la puissance de contrôle des appareils connectés de votre application.

Informations des capteurs et Historique

Une fois que vous cliquez sur un capteur spécifique, vous serez dirigé vers une page dédiée fournissant des informations détaillées sur ce capteur particulier, telles que son numéro de broche, son état actuel et d'autres détails pertinents. En outre, vous aurez accès à un graphique interactif affichant les valeurs précédentes capturées par ce capteur, vous permettant ainsi de visualiser facilement les tendances et les changements au fil du temps. Cette fonctionnalité vous offre un contrôle total sur les performances et le fonctionnement de chaque capteur, vous permettant de prendre des décisions éclairées pour optimiser votre système sous les interfaces illustrée dans la figure 40



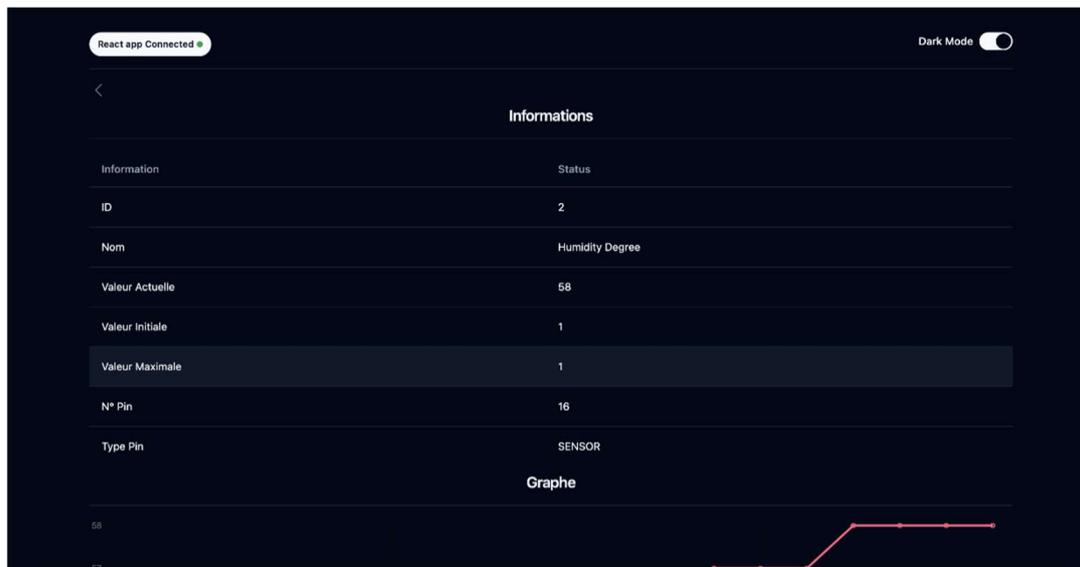


Figure 40 : Interface Accès aux détails des capteurs

Configurations Pin

Une fois dans l'interface de Django Admin, vous pouvez modifier les paramètres des broches, y compris leur numéro, leur état et d'autres configurations pertinentes, en fournissant ainsi un contrôle complet sur la configuration matérielle de votre système. Cette intégration transparente vous permet de gérer efficacement votre infrastructure IoT, en combinant la flexibilité de Django Admin avec la puissance de contrôle des appareils connectés de votre application présentée sous la figure 41.

Site administration

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#) [Change](#)
- Users [+ Add](#) [Change](#)

CONTROLLER

- Microcontroller pin values [+ Add](#) [Change](#)
- Microcontroller pins [+ Add](#) [Change](#)
- Microcontrollers [+ Add](#) [Change](#)

DJANGO APSCHEDULER

- Django job executions [+ Add](#) [Change](#)
- Django jobs [+ Add](#) [Change](#)

Recent actions

My actions

- ✖ Zone A - rayan - 12 - SWITCH - LOW - Microcontroller pin
- ✖ Zone A - 25 - 25 - SWITCH - HIGH - Microcontroller pin
- ✖ Zone A - 27 - 27 - SWITCH - LOW - Microcontroller pin
- ✖ Zone A - 26 - 26 - SWITCH - LOW - Microcontroller pin
- ✖ Zone A - 13 - 13 - SWITCH - HIGH - Microcontroller pin
- ✎ Zone B - test - 2 - SWITCH - 1 - Microcontroller pin
- + Zone B - test - 2 - SENSOR - 1 - Microcontroller pin
- ✎ Zone A - Soil Humidity - 33 - SENSOR - 4095 - Microcontroller pin
- ✎ Zone A - 33 - 33 - SENSOR - LOW - Microcontroller pin
- + Zone A - 33 - 33 - SWITCH - HIGH - Microcontroller pin

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

CONTROLLER

- Microcontroller pin values [+ Add](#)
- Microcontroller pins [+ Add](#)
- Microcontrollers [+ Add](#)

DJANGO APSCHEDULER

- Django job executions [+ Add](#)
- Django jobs [+ Add](#)

Select microcontroller pin value to change

[ADD MICROCONTROLLER PIN VALUE +](#)

Action: Go 0 of 100 selected

- MICROCONTROLLER PIN VALUE
- Soil Humidity - 4095 - 2024-06-14 22:01:28.454909+00:00
- Water Level - 4095 - 2024-06-14 22:01:28.335831+00:00
- Soil Humidity - 4095 - 2024-06-14 22:00:58.505367+00:00
- Water Level - 4095 - 2024-06-14 22:00:58.158402+00:00
- Soil Humidity - 4095 - 2024-06-14 22:00:28.565084+00:00
- Water Level - 4095 - 2024-06-14 22:00:28.152447+00:00
- Soil Humidity - 4095 - 2024-06-14 21:59:58.495635+00:00
- Water Level - 4095 - 2024-06-14 21:59:58.156973+00:00
- Soil Humidity - 4095 - 2024-06-14 21:59:35.640865+00:00
- Water Level - 4095 - 2024-06-14 21:59:35.432073+00:00
- Soil Humidity - 4095 - 2024-06-14 21:59:13.612314+00:00
- Water Level - 4095 - 2024-06-14 21:59:13.309310+00:00
- Soil Humidity - 4095 - 2024-06-14 21:58:44.039369+00:00
- Water Level - 4095 - 2024-06-14 21:58:43.840515+00:00
- Ventilateur - HIGH - 2024-06-14 21:58:26.039913+00:00
- Ventilateur - LOW - 2024-06-14 21:58:12.036471+00:00

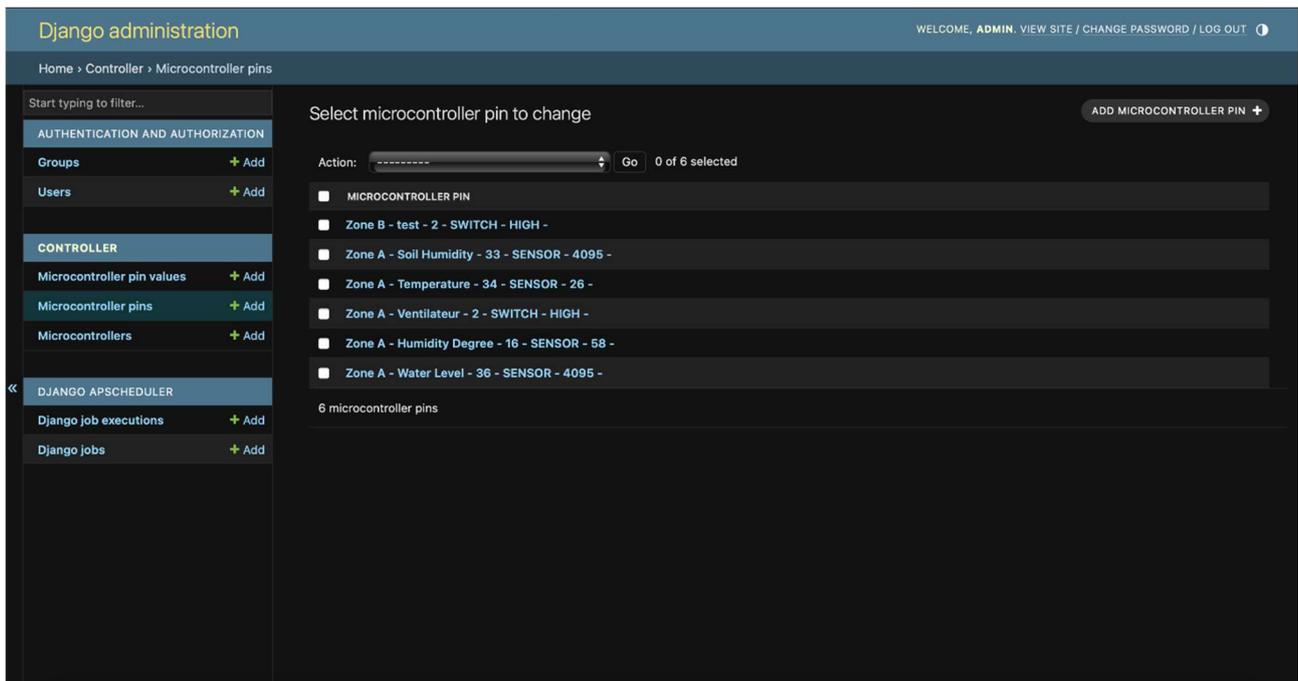


Figure 41 : Interface configuration et accès à base de données

Résultats :

Les figures 42,43 illustrent le montage complet du système matériel et logiciel réalisé.

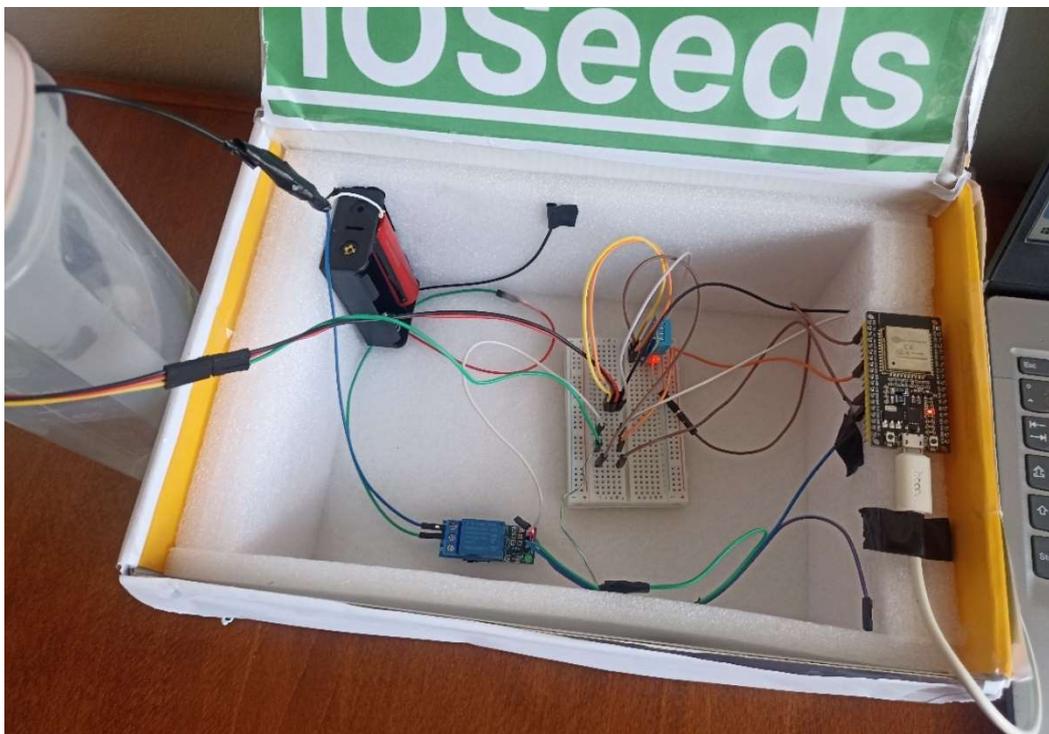


Figure 42 : fonctionnement du système complet réalisé



Figure 43 : fonctionnement du système avec pompe activé

Discussion des résultats

Notre système d'irrigation réalisé a satisfait les résultats suivants :

- Le suivi des valeurs des capteurs depuis le terrain et le contrôle de la pompe à eau à distance en temps réel, avec accès à la gestion et à la manipulation des valeurs internes des microcontrôleurs du système.
- L'algorithme d'irrigation proposé a permis de satisfaire les besoins du système en évitant les cas de sur-irrigation ou de boucle d'irrigation infinie, en déléguant la responsabilité de suivre le processus d'irrigation à l'ESP32 en cas d'absence de supervision humaine.
- Ce système a permis la collecte des données environnementales et des actions des superviseurs du système correspondants, ce qui prépare le terrain pour l'application d'intelligence artificielle pour des systèmes de prédiction et de recommandation dans le futur.

Conclusion

On a présenté dans ce chapitre la réalisation du système global d'irrigation basé sur le prototype d'un microcontrôleur et de ses différents composants électroniques et capteurs. Les deux applications web permettent une interaction et un contrôle efficaces pour répondre aux besoins du système d'irrigation réalisé.

Conclusion générale

Ce mémoire a présenté le développement d'un système d'irrigation précis, intégrant des technologies avancées pour optimiser la gestion de l'eau et améliorer l'efficacité des pratiques agricoles. Le travail réalisé a consisté en une analyse approfondie des besoins, suivie par la conception et l'implémentation du système. Nous avons détaillé la structure matérielle, proposé un algorithme d'irrigation efficace, et défini la partie logicielle, assurant une architecture robuste et cohérente.

Nous avons réussi à développer un système d'irrigation fonctionnel qui utilise les données des capteurs en temps réel pour prendre des décisions éclairées, réduisant ainsi le gaspillage d'eau et augmentant la productivité des cultures. Les tests effectués ont démontré l'efficacité et la fiabilité du système qui sert à :

L'agriculture de précision : Un arrosage précis basé sur les mesures d'environnement en temps réel permet de réduire le gaspillage d'eau et d'optimiser son utilisation.

Le contrôle à distance : La possibilité de contrôler et de surveiller l'irrigation à distance grâce à l'interface utilisateur réalisée.

L'actionnement de l'irrigation : Un actionneur d'irrigation (pompe) est contrôlé par le système pour fournir de l'eau en fonction des commandes de l'utilisateur ou des règles automatisées, tout en prenant en considération le cas d'absence d'une ou plusieurs parties du système pour éviter le gaspillage et la sur-irrigation.

Des décisions basées sur les données historiques : Les données historiques accessibles facilitent l'analyse et l'amélioration informée des programmes d'irrigation.

Pour l'avenir, plusieurs perspectives d'amélioration peuvent être envisagées. D'abord, l'intégration de techniques d'apprentissage automatique pourrait permettre une optimisation encore plus fine des schémas d'irrigation en fonction des conditions météorologiques et des données historiques. Ensuite, la mise en œuvre de meilleures mesures de sécurité pour protéger le système contre l'accès non autorisé. De plus, on pourra envisager une amélioration matérielle en utilisant des modules de connexion qui couvrent des espaces plus larges. Enfin, on pourra intégrer un module de détection d'anomalies pour mieux tirer parti des mesures d'environnement des capteurs, car ces derniers peuvent exprimer des détections de défauts périodiquement.

Références

- [1] R. Albandes, R. Machado, J. Lopes, J. Barbosa et A. Yamin, «An IoT Proposal for the Irrigation Management Exploring Context Awareness,» chez *2nd IFIP International Internet of Things Conference (IFIPIoT)*, Tampa, FL, United States, 2019.
- [2] *Près de 70% de la consommation d'eau accaparés par l'agriculture* | *Radio Algérienne*, 2018.
- [3] D. Andrieu, *Maîtriser l'ingénierie des systèmes de systèmes*, 2017.
- [4] «Internet of Things (IoT),» chez *Evolution of Wireless Communication Ecosystems*, John Wiley & Sons, Ltd, 2023, pp. 177-189.
- [5] ThingSpeak. [En ligne]. Available: <https://thingspeak.com/>. [Accès le 23 2024].
- [6] «croplytics by pragmatic,» [En ligne]. Available: <https://pragmatic.inosens.rs/product-name/croplytics/>. [Accès le 23 2024].
- [7] S. Terence et G. Purushothaman, «Systematic review of Internet of Things in smart farming,» *Transactions on Emerging Telecommunications Technologies*, vol. 31, p. e3958, June 2020.
- [8] educaclima, 23 2022. [En ligne]. Available: <https://educaclima.azurewebsites.net/do-you-live-in-a-smart-city/?lang=en>. [Accès le 5 3 2024].
- [9] H. Mrabet, S. Belguith, A. Alhomoud et A. Jemai, «A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis,» *Sensors*, vol. 20, 2020.
- [10] J. Kaur et K. Kaur, «Internet of Things: A Review on Technologies, Architecture, Challenges, Applications, Future Trends,» *International Journal of Computer Network & Information Security*, vol. 9, 2017.
- [11] K. Sivanraju, «IoT in Agriculture : Smart Farming,» *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 181-184, December 2018.
- [12] A. Sivagami, U. Hareeshvare, S. Maheshwar et V. S. K. Venkatachalapathy, «Automated Irrigation System for Greenhouse Monitoring,» *Journal of The Institution of Engineers (India): Series A*, vol. 99, pp. 183-191, March 2018.
- [13] S. Velmurugan, «An IOT based smart irrigation system using soil moisture and weather prediction,» 2020.
- [14] A. Dheer, M. Sharma et K. Tripathi, *Smart irrigation system based on IoT*.
- [15] M. Pramanik, M. Khanna, M. Singh, D. K. Singh, S. Sudhishri, A. Bhatia et R. Ranjan, «Automation of soil moisture sensor-based basin irrigation system,» *Smart Agricultural Technology*, vol. 2, p. 100032, 2022.
- [16] A. Kaur, D. P. Bhatt et L. Raja, «Developing a Hybrid Irrigation System for Smart Agriculture Using IoT Sensors and Machine Learning in Sri Ganganagar, Rajasthan,» *Journal of Sensors*, vol. 2024, p. 1–15, January 2024.
- [17] P. Vadityavath et I. Baddam, *INNOVATIVE SOLUTIONS FOR SUSTAINABLE AGRICULTURE: IoT-DRIVEN IRRIGATION SYSTEMS*, 2024.
- [18] P. Tan, E. T. Gebremariam, M. S. Rahman, H. Salman et H. Xu, «Design and Implementation of Soil Moisture Monitoring and Irrigation System based on ARM and IoT,» *Procedia Computer Science*, vol. 208, pp. 486-493, 2022.
- [19] G. Kokkonis, S. Kontogiannis et D. Tomtsis, «A Smart IoT Fuzzy Irrigation System,» *IOSR Journal of Engineering*, vol. 07, pp. 15-21, June 2017.
- [20] N. G. S. Campos, A. R. Rocha, R. Gondim, T. L. Coelho da Silva et D. G. Gomes, «Smart & Green: An Internet-of-Things Framework for Smart Irrigation,» *Sensors*, vol. 20, 2020.

- [21] V. V. Vegesna, S. C. Mathi, R. M Kaviya, J. K. Arasi, R. M. Sha et P. Dhivya, «Optimization and control of hydroponics agriculture using IOT,» *Asian Journal of Applied Science and Technology (AJAST) Volume*, vol. 1, p. 96–98, 2017.
- [22] S. Chowdhury, S. Sen et S. Janardhanan, *Comparative Analysis and Calibration of Low Cost Resistive and Capacitive Soil Moisture Sensor*, 2022.
- [23] L. Garc a, L. Parra, J. M. Jimenez, J. Lloret et P. Lorenz, «IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture,» *Sensors*, vol. 20, 2020.
- [24] M. , *Le fonctionnement de Django - D veloppez votre site web avec le framework Django • Biblioth que • Zeste de Savoir*, 2016.
- [25] U. Pisuwala, «Internet of Things in Healthcare: Applications, Benefits, and Challenges,» 18 10 2023. [En ligne]. Available: <https://www.peerbits.com/blog/internet-of-things-healthcare-applications-benefits-and-challenges.html>. [Acc s le 5 3 2024].
- [26] P. magasin, «Powertech magasin de vente de composants electronique,» [En ligne]. Available: <https://powertech-dz.net/>. [Acc s le 17 05 2024].
- [27] «blynk io,» [En ligne]. Available: <https://blynk.io/home-new>. [Acc s le 2 3 2024].
- [28] «Adafruit IO,» [En ligne]. Available: <https://io.adafruit.com/>. [Acc s le 2 3 2024].