

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE**



**Mémoire De Fin d'études En Vue D'obtention
D'un Master Académique**

Filière : Informatique

Spécialité : Réseaux, Mobilité et Systèmes Embarqués

**Conception d'un système IoT de bout en bout pour le suivi
temps réel des processus de production, de géolocalisation et
de surveillance**

Présente par :

Mlle. HANAFI Imane

Mlle. YAHIA Samia

Proposé et encadré par :

MR.RAMDANI Mohamed

Présenté le 12/10/2020 devant la commission de jury composée de :

Président : FILALI Idir

Examineur : SADI Samy

Promotion : 2019/2020



Avant-propos



Ce travail a été réalisé, après huit mois d'efforts, à l'UNIVERSITE **MOULOUD MAMMERI DE TIZI-OUZOU**.

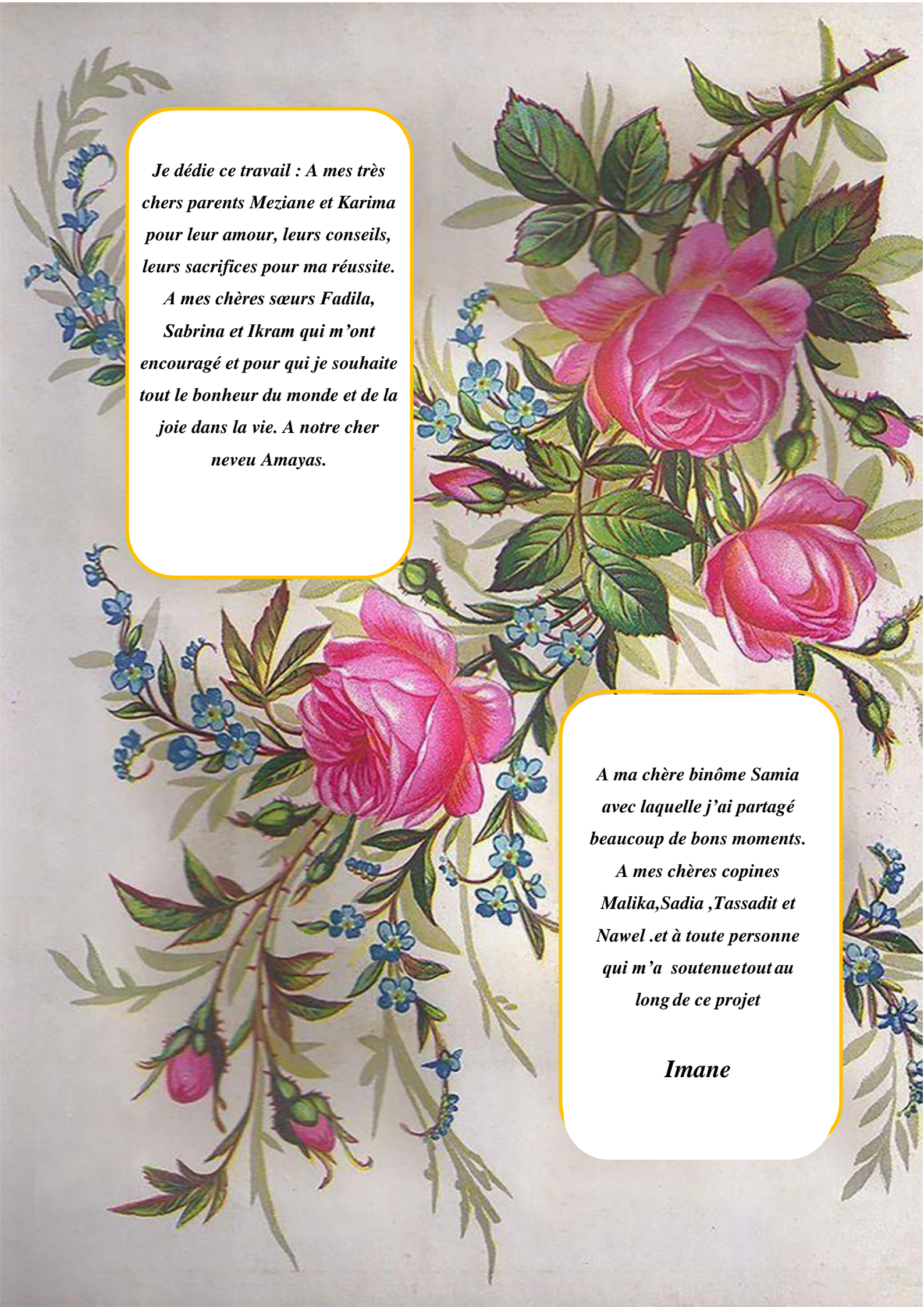
Nous remercions Dieu le Tout-Puissant, pour nous avoir donné la force, la patience, qui nous a doté d'intelligence, et nous a maintenu en santé pour mener à bien cette année d'étude et de mener à terme ce présent travail.

Nous nous faillirons à la tradition si nous n'exprimons pas ici notre gratitude envers les responsables de l'université pour les moyens mis à notre disponibilité, en particulier au niveau du département d'informatique pour le suivi et la disponibilité. Il en est de même pour nos enseignants qui nous ont assuré un apprentissage de qualité.

Nous tenons à remercier notre encadreur, monsieur RAMDANI Mohamed, pour sa confiance. Nous voudrions bien lui exprimer notre gratitude pour les discussions scientifiques précieuses et le temps consacré à nous orienter dans nos travaux.

Nous remercions vivement monsieur FILALI Idir, pour l'honneur qu'il nous fait en acceptant de présider le Jury de soutenance. Il en est de même pour monsieur SADI Samy, pour l'intérêt et le temps pris pour examiner ce travail.

Nous sommes enfin reconnaissantes à tous ceux qui ont contribué, de près ou de loin, à l'accomplissement de ce travail par leurs encouragements.



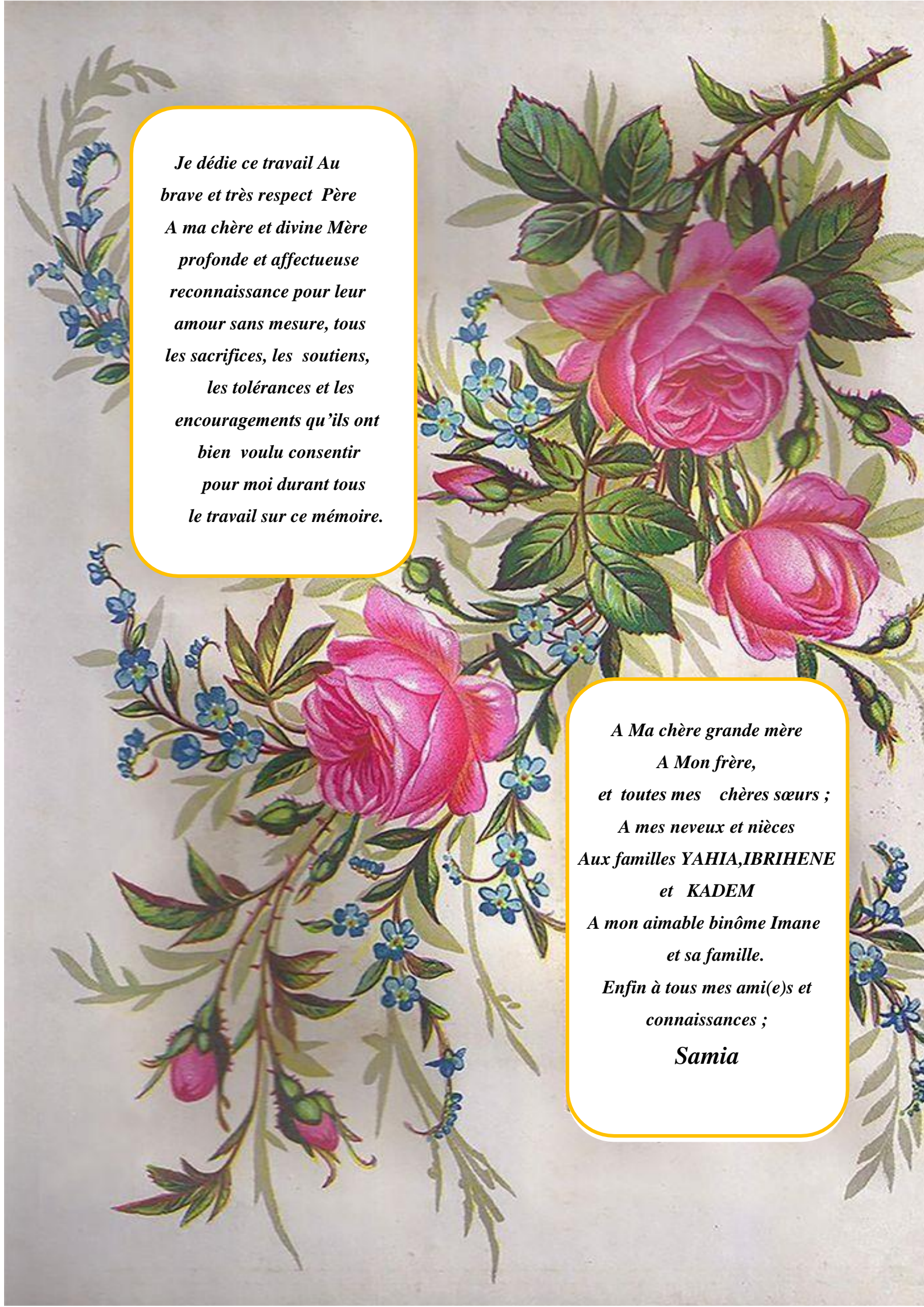
Je dédie ce travail : A mes très chers parents Meziane et Karima pour leur amour, leurs conseils, leurs sacrifices pour ma réussite.

A mes chères sœurs Fadila, Sabrina et Ikram qui m'ont encouragé et pour qui je souhaite tout le bonheur du monde et de la joie dans la vie. A notre cher neveu Amayas.

A ma chère binôme Samia avec laquelle j'ai partagé beaucoup de bons moments.

A mes chères copines Malika, Sadia, Tassadit et Nawel .et à toute personne qui m'a soutenu tout au long de ce projet

Imane



*Je dédie ce travail Au
brave et très respect Père
A ma chère et divine Mère
profonde et affectueuse
reconnaissance pour leur
amour sans mesure, tous
les sacrifices, les soutiens,
les tolérances et les
encouragements qu'ils ont
bien voulu consentir
pour moi durant tous
le travail sur ce mémoire.*

*A Ma chère grande mère
A Mon frère,
et toutes mes chères sœurs ;
A mes neveux et nièces
Aux familles YAHIA, IBRIHENE
et KADEM
A mon aimable binôme Imane
et sa famille.
Enfin à tous mes ami(e)s et
connaissances ;
Samia*

Résumé

La technologie de l'Internet des Objets (IoT) a attiré beaucoup d'attention ces dernières années en particulier au niveau des entreprises. L'IoT consiste principalement à connecter des objets physiques à l'Internet. Le Web of Things (WoT) est un IoT plus spécifique qui vise à apporter des technologies et des normes du Web à l'IoT. L'émergence de l'IoT et du WoT offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel, un processus qui n'était pas possible auparavant.

De nos jours, il existe de nombreuses plateformes et applications pour l'IoT. Cependant, et au mieux de notre connaissance, ils se limitent généralement dans leur champ d'application à un simple schéma de stockage et de récupération des données. Dans une tentative de profiter de cette occasion et utiliser les techniques existantes dans le monde de l'IoT, ce mémoire présente un modèle d'un système IoT de bout en bout pour le suivi temps réel des processus de production, de géolocalisation et de surveillance. Basé sur la programmation matérielle et afin d'exposer les fonctionnalités des objets connectés et utiliser les fonctionnalités de la carte arduino uno dans les systèmes embarqués, nous avons réalisé une application web pour le suivi temps réel des grandeurs environnementales notamment la température et l'humidité, en utilisant un capteur de température et d'humidité DHT-22 plus une carte Arduino UNO pour l'acquisition des données. Pour le fonctionnement de notre application web on a développé une partie backend en utilisant NodeJS pour le développement du serveur et postgresql pour le stockage des données du système et une partie frontend en utilisant ReactJS. Notre application offre des services à l'entreprise notamment l'affichage en temps réel des données captées (température et humidité) afin d'aider l'utilisateur à visualiser le changement des données acquises par le capteur sous forme de graphes en temps réel. L'affichage de l'historique des données captés du dernier jour, de la dernière semaine, ou du dernier mois, qui aiderons l'utilisateur à établir des statistiques et des études stratégiques et de faire des prédictions au sein de l'entreprise en vue de garantir une bonne performance, de réduire les risques et d'améliorer la productivité en termes de quantité et de qualité.

Abstract

Internet of Things (IoT) technology has gained a lot of attention in recent years especially at the level of companies. The IoT is mainly about connecting physical objects to the Internet. The Web of Things (WoT) is a more specific IoT that aims to bring technologies and standards from the web to the IoT. The emergence of IoT and WoT offers great potential for the development of new services and applications connecting the physical world to the virtual world, a process that was not possible before. Today, there are many platforms and applications for the IoT. However, for the best of our knowledge, they are generally limited in scope to a simple scheme of data storage and retrieval. In an attempt to take advantage of this opportunity and used existing techniques in the IoT world, this project presents a model of a complete end-to-end IoT system for real-time monitoring of production, geolocation and surveillance. Based on hardware programming, and in order to expose the functionalities of connected objects and use the functionalities of the arduino uno card in on-board systems, we have produced a web application for real-time monitoring of environmental quantities, in particular temperature and humidity in using a DHT-22 temperature and humidity sensor with an Arduino UNO board.

Sommaire

Remerciement

Dédicaces

Résumé

Sommaire

Liste des figures

Liste des abréviations

Introduction générale01

Chapitre I – Concepts fondamentaux de l'IOT

I.1 Introduction.....03

I.2 Etat de l'art de l'IoT.....04

I.2.1 Définition L'internet des objets.....04

I.2.2 Les différents processus de l'internet des objets.....05

I.2.3 Fonctionnement de l'IoT.....05

I.2.4 Les objets connectés.....06

I.2.5 Les avantages de l'internet des objets.....15

I.2.6 Les inconvénients de l'IoT.....15

I.2.7 La sécurité dans l'IoT16

I.2.8 Les challenges de l'IoT.....17

I.2.9 Les protocoles de communication dans l'IoT.....18

I.2.10 Les moyens de communication dans l'IoT19

I.3 Domaines d'application.....21

I.4 Conclusion.....25

Chapitre II : les services web et l'IoT

II.1 Introduction26

II.2 L'apport des langages web dans l'IoT.....26

II.1.1 Services Web.....26

Sommaire

II.1.2 Intergiciel.....	29
II.1.3 WoT <i>web of things</i>	30
II.3 La problématique	30
II.4 Objectifs et contributions.....	31
I.5 La conclusion.....	32
Chapitre III – Méthodes et matériels	
III.1 Introduction.....	33
III .2 Rôle de NodeJS dans les applications embarquées.....	33
III.2.1 La naissance de NodeJS.....	33
III.2.2 NodeJS et l’IoT.....	34
III.2.3 Systèmes embarqués.....	35
III .3 Modules et Node-Package-Manager.....	37
III.3.1 Les frameworks de NodeJS.....	37
III .4 Communication temps réel avec Express et Socket.io.....	39
III .4.1 ExpressJS.....	40
III .4.2 Socket.io.....	40
III.5 Module Arduino	40
III.5.1 Définition.....	40
III.5.2 Les gammes de la carte Arduino.....	41
III.6 RASPBERRY PI 3.....	45
III.6.1 Histoire.....	45
III.6.2 Définition.....	45
III.6.3 Equipement essentiel.....	45
III.6.4 Les spécifications Entrées / Sorties.....	46
III.6.5 L’intérêt du Raspberry PI 3.....	46
III.6.6 Les améliorations sur la carte Raspberry.....	47

Sommaire

III.7 Comparaison des cartes.....	48
III.8 Conclusion.....	48
Chapitre IV – Réalisation et implémentation de système	
IV.1 Introduction.....	49
IV.2 Outils utilisés	49
IV.3 Les technologies utilisées	52
IV.4 L'architecture globale du système	55
IV.4.1 Câblage du système.....	55
IV.4.2 implémentation du système.....	56
IV.4.3 Présentation des interfaces.....	59
IV.5 Conclusion.....	61
Conclusion générale.....	62
Perspectives.....	63
Bibliographie	

Liste de figures

Figure I.1 : nombre d'appareils connectés [2]	4
Figure I.2 : le maillage des réseaux [2]	4
Figure I.3 : l'internet des objets [68]	5
Figure I.4 : la lampe DAL [7]	7
Figure I.5 : bracelet connecté [8]	7
Figure I.6 : capteur GPS UBLOX NEO-6M [9]	7
Figure I.7 : capteur DHT-22[75]	8
Figure I.8 : capteur d'empreinte. [11]	8
Figure I.9 : Module de caméra OV7670 [12]	9
Figure I.10 : capteur de pression BMP 280[13]	9
Figure I.11: capteur de mouvement PIR HC-SR501 [15]	10
Figure I.12 : lecteur RFID RC522 avec ses accessoires [16]	10
Figure I.13 : Le microcontrôleur [17]	11
Figure I.14 : Architecture interne d'un microcontrôleur [17]	12
Figure I.15: le protocole HTTP [21]	18
Figure I.16 : services de la santé [23]	22
Figure I.17 : Services de la domotique [1]	22
Figure I.18 : Services du Smart Grid [1]	23
Figure I.19 : IoT dans les transports [1]	23
Figure I.20 : IoT dans l'industrie [1]	24
Figure I.21 : IoT dans l'agriculture [26]	24
Figure I.22 : l'IoT dans le militaire [28]	25
Figure II.1: les services du web [69]	27
Figure II.2 : Architecture classique des services Web [29]	28
Figure II.3 : Architecture Middleware [29]	29

Liste de figures

Figure III.1 : les trois vies de JavaScript [62].....	34
Figure II.2 : l'exécution non bloquante [62].....	35
Figure III.3 : Architecture générale et modes de fonctionnement [32].....	36
Figure III.4 : sailsJS. [71].....	38
Figure III.5 : MeanJS [72].....	38
Figure III.6 : electronJS [70].....	39
Figure III.7 : ExpressJS [73].....	40
Figure III.8 : Socket.io [74].....	40
Figure III.9 : Description de la carte Arduino-Mega2560 [39].....	41
Figure III.10 : Description de la carte Arduino-Nano [39].....	42
Figure III.11 : Description de la carte Arduino-Uno [39].....	44
Figure III.12 : Raspberry pi 3[40].....	45
Figure III.13 : Tableau comparatif entre quelques cartes raspberry et arduino [91][92][93][94].....	48
Figure IV.1 : Sublime Texte [42].....	49
Figure IV.2 :L'IDE arduino [44].....	50
Figure IV.3 : Node Package Manager [49].....	51
Figure IV.4 : Git Hub [51].....	52
Figure IV.5 : TeamViewer [53].....	52
Figure IV.6 : PostgreSQL [53].....	53
FigureIV.7 : ReactJS [56].....	53
Figure IV.8 :NodeJS [58].....	54
Figure IV.9 : ExpressJS [61].....	54
Figure IV.10 : architecture globale du système.....	55
Figure IV.11 : câblage du système.....	56
Figure IV.12 : Montage du capteur DHT22.....	56

Liste de figures

Figure IV.13 : client ReactJS.....	57
Figure IV.14 : serveur NodeJS.....	57
Figure IV.15: La fonction d'insertion sur la BDD.....	58
Figure IV.16: La table data.....	58
Figure IV.17: interface d'authentification.....	59
Figure IV.18: données de capteur DHT-22.....	60
Figure IV.19: Les données récupérées de la BDD.....	61

Liste des abréviations

ADSL Signifie **A**symetric **D**igital **S**ubscriber **L**ine

ASIC Application Specific Integrated Circuits

AVR Automatic Voltage Regulator

BDD Base De Données

CAN Controller Area Network

CLI Command line Interface

COAP Constrained Application Protocol

CPU Central Processing Unit

EEPROM Electrically Erasable Programmable Read Only Memory

EPROM Erasable Programmable Read Only Memory

FPGA Field Programmable Gate Array

GSM Global System for Mobile Communications

GPS Global Positioning System

HTTP Hypertext Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

IDE Integrated Development Environment

IHM Interface Homme Machine

IoT Internet Of Things

IP Internet Protocol

IPv4 Internet Protocol Version 4

IPv6 Internet Protocol Version 6

I2C Inter-Integrated Circuit

JSON JavaScript Object Notation

LCD Liquid Crystal Display

LED Light Emitter Diode

MLI Modulation de Largeur d'Impulsions

Liste des abréviations

MMS Multimedia **M**essaging **S**ervice

MQTT **M**essage **Q**ueuing **T**elemetry **T**ransport

MVC **M**odel **V**iew **C**ontroller

NFC **N**ear **F**ield **C**ommunication

NPM **N**ode **P**ackage **M**anager

OTPROM **O**ne **T**ime **P**rogrammable **R**ead **O**nly **M**emory

PC **P**ersonal **C**omputer

PCB **P**rinted **C**ircuit **B**oard

PROM **P**rogrammable **R**ead **O**nly **M**emory

PPS **P**ulse **P**ar **S**econde

PWM **P**ulse **W**idth **M**odulation

SCCB serial **c**ameral **c**ontrol **b**us

SIM **S**ubscriber **I**dentify/**I**dentification **M**odule

SMA **S**ub-**M**iniature **V**ersion **A**

SMS **S**hort **M**essage **S**ervice

SOA **S**ervice-**O**riented **A**rchitecture

SOAP **S**imple **O**bject **A**ccess **P**rotocol

SPI **S**erial **P**eripheral **I**nterface

SSP **S**ecure **S**imple **P**airing

TCP/IP **T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol

TIC **T**echnologie d'**I**nformation et **C**ommunication

TV **T**ele**V**ision

UART **U**niversal **A**synchronous **R**eceiver **T**ransmitter

UDDI **U**niversal **D**escription, **D**iscovery, and **I**ntegration

URI **U**niform **R**esource **I**dentifier

URN **U**niform **R**esource **N**ame

Liste des abréviations

USB Universal Serial **B**us

UTC Coordinated Universal Time

UV-EPROM Ultra-Violet Erasable Programmable Read Only Memory

VCC Virtual Channel Connection

WIFI Wireless Fidelity

VGA Video Graphics Array

WOT Web Of Things

WSN Wireless Sensor Network

WSDL Web Services Description Language

XML Extensible Markup Language

XML-RCP Extensible Markup Language Remote Procedure Call

Introduction

Introduction

Générale

Générale

Introduction générale

L'informatisation de plus en plus généralisée des activités humaines représente un effort continu qui a pris naissance avec l'avènement des ordinateurs. L'homme a de plus en plus besoin d'avoir recours aux machines, aux robots et aux objets capables de fournir des services avec peu, voire sans aucune intervention humaine. Par ailleurs, les ordinateurs, les appareils électroniques et surtout l'informatique et l'Internet deviennent une nécessité de la vie quotidienne.

Au fil du temps, on a intégré l'ordinateur dans différents objets. En effet, avec le Web, ces objets peuvent se connecter et communiquer entre eux. Du coup, les machines liées à l'Internet sont conçues pour exécuter, puis penser, et aujourd'hui, elles apprennent à percevoir, sentir et réagir. C'est dans ce contexte que nous orientons notre recherche sur le concept de lier les objets à l'Internet, connu de nos jours comme « Internet of Things » (IoT).

Nous entendons parler de l'IoT. En effet, en raison de ses applications, l'IoT a retenu beaucoup d'attention autant dans le milieu universitaire que dans l'industrie et plusieurs autres domaines. Dans notre travail de recherche, nous tenterons d'apporter une contribution à cet égard en proposant une solution aux problèmes rencontrés lors de l'application de l'IOT dans divers domaines dans le monde notamment au niveau des entreprises.

Pour répondre au besoin de l'utilisateur qui cherche à obtenir le bon service au bon moment et au bon endroit, on pense qu'il existe plusieurs défis à la mise en œuvre réussie de l'Internet of Things (IoT). En tant qu'informaticiens, notre souci est d'arriver à combiner le monde physique et le monde virtuel, autrement dit, de travailler sur la standardisation de ce nouveau système en développant de nouvelles applications et de nouveaux environnements ainsi que de nouveaux protocoles de communication. Par exemple, le système RFID (identification par radiofréquence), le protocole ZigBee, le Cloud Computing... Après l'analyse de ces travaux et de bien d'autres, on observe que les recherches effectuées jusqu'à présent sont très modestes face à une présence de plus en plus forte de données et de services qui ne sont pas exploités à leur plein potentiel. De plus, la zone de couverture de leur système est très limitée géographiquement, et leur approche utilise toujours le principe client-serveur qui présente des failles connues comme la saturation, la déconnexion et autres, ce qui met en péril tout le système. Ce travail de recherche vise donc à concevoir et implémenter un système IoT de bout en bout pour le suivi temps réel des processus de production, de géolocalisation et de surveillance. L'idée derrière ce modèle est d'apporter, à l'aide de services Web, une solution liée à l'IoT.

Introduction générale

Selon le géant de la réseautique Cisco, les appareils connectés ont atteint le nombre de 50 milliards en 2020 et intègrent notre vie personnelle et professionnelle dans ses différentes facettes. D'où l'importance de mettre l'accent sur la nécessité de mettre en place des systèmes d'IoT qui répondent à nos besoins. Pour ce faire, notre travail est devisé en quatre chapitres, organisé comme suit :

Dans le premier chapitre, nous allons présenter les concepts fondamentaux de l'internet des objets. Il est donc nécessaire de définir l'État de l'art sur IoT afin de comprendre les technologies utilisées dans ce domaine et son fonctionnement. A la fin de ce chapitre, nous allons évoquer en détail les domaines d'applications de l'IoT.

Le deuxième chapitre, présente les services web et l'IoT. Nous commençons par présenter l'apport des langages Web dans l'IoT. Puis, nous exposerons la problématique, ensuite nous citerons les objectifs et contribution de nos recherches.

Le troisième chapitre présente une description générale du matériel et méthodes dédiés à la conception du système proposé dans ce projet. En première partie, nous décrirons le Rôle de NodeJS dans les applications embarquées, par la suite, nous détaillerons les outils utilisés par NodeJS pour son fonctionnement notamment les Modules et NPM (Node Package Manager) et nous décrivons par la suite la communication temps réel avec Express et Socket.io. En deuxième partie de ce chapitre, nous présenterons l'environnement de développement intégré (IDE) de l'Arduino, afin d'exploiter les ressources d'une carte d'acquisition Arduino UNO, ainsi nous décrivons brièvement comment connecter des périphériques à celle-ci.

La phase de la conception et de la simulation du système proposée sera présentée au quatrième chapitre. Selon l'étude que nous avons effectuée au parcours des deux premiers chapitres, nous avons proposé un schéma conceptuel pour accomplir le système, afin d'obtenir les résultats désirés de ce projet via une application installée sur un navigateur web. Enfin, ce document se termine par une conclusion générale résumant le bilan des travaux effectués afin de mieux assimiler son contenu et l'ouverture aux travaux futurs.

Chapitre

CU9b1rL6

N°01

14 01

I.1 Introduction

L'évolution importante de l'Internet en terme de services offerts aux utilisateurs et du nombre d'entités connectées, rend ce réseau mondial indispensable dans tous les domaines de la vie quotidienne (économie, communication, militaire, etc.).

Depuis sa création en 1969, Internet était « **L'Internet des ordinateurs** » où il regroupait des milliards de connexions et d'échanges entre des équipements informatiques et offrait aux utilisateurs un ensemble de services (navigation web, téléchargements, etc.) ; jusqu'à sa migration vers « **L'Internet des personnes** » grâce à la tendance des services de messageries et des réseaux sociaux où des flux de données sont échangés entre des personnes connectées. A présent, les équipements d'accès à Internet varient de grosses stations de travail à des miniatures appareils mobiles intelligents (Tablettes, Smartphone, etc.) équipés de plus en plus par des capteurs et d'actionneurs permettant l'interaction avec l'environnement physique. L'ouverture de l'Internet et l'accessibilité du réseau permettent d'y connecter une multitude d'objets pour une multitude d'usages, cela est connu sous le nom de « **L'Internet des objets** » [1].

Aujourd'hui, il y a plus d'appareils intelligents que de personnes. De plus en plus, beaucoup de personnes sont connectées à Internet, d'une manière ou d'une autre, 24 heures sur 24. Un nombre toujours plus élevé de personnes possèdent et utilisent trois, quatre appareils intelligents, voire plus. On compte notamment des smartphones, des systèmes de surveillance de l'activité physique et de l'état de santé, des liseuses et des tablettes. Actuellement en 2020, chaque utilisateur possède en moyenne 6,58 appareils intelligents, comme le montre la Figure I.1 [2].

Les réseaux numériques modernes rendent tout cela possible. Le monde a rapidement été couvert par les réseaux qui permettent aux appareils numériques de s'interconnecter et de transmettre des données. Imaginant le maillage de réseaux comme une peau numérique qui entoure la planète, comme illustré dans la Figure I.2. Grâce à elle, les terminaux mobiles, les capteurs électroniques, les instruments de mesure électroniques, les dispositifs médicaux et les indicateurs peuvent tous se connecter. En effet, ils assurent une fonction de surveillance, communiquent, évaluent les données et dans certains cas, s'adaptent automatiquement aux informations collectées et transmises [2].

Nous vivons actuellement une transformation numérique, à mesure que la société adopte des appareils technologiques et des réseaux informatiques, qui continuent de gagner du terrain dans le monde entier si bien que les bénéfices économiques de la numérisation se multiplient. La transformation numérique consiste à appliquer la technologie numérique pour créer les bases de l'innovation dans les entreprises, le secteur de l'industrie et tous les aspects de la société humaine [2].

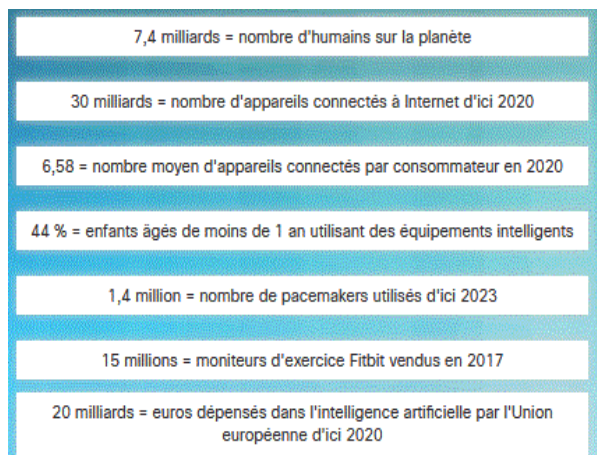


Figure I.1 : nombre d'appareils connectés [2].

Figure I.2 : le maillage des réseaux [2].

I.2 L'état de l'art de l'IoT

I.2.1 Définition L'internet des objets

Le terme « internet of things » (en français internet des objets) est né en 1999 au centre, MIT (Massachusetts Institute of Technology), fondé par Kevin Ashton, un chercheur britannique, pionnier dans son domaine (IDO) [3].

Internet des objets (IoT) désigne l'interconnexion de millions d'appareils et de capteurs intelligents connectés à Internet. Ces derniers collectent et partagent des données qui seront utilisées et analysées par plusieurs organismes, dont des entreprises, des villes, des gouvernements, des hôpitaux et des particuliers. L'avènement de l'IoT a été possible grâce à l'apparition des nouvelles technologies électroniques telles que les capteurs, les actionneurs, diverses cartes DIY Do It Yourself et des réseaux sans fil. Des objets jusqu'à présent inanimés peuvent désormais être équipés d'un capteur intelligent ou plusieurs qui collectent des données et les transferts à un réseau [67].



Figure I.3 : l'internet des objets [68].

I.2.2 Les différents processus de l'internet des objets

- **Capter** : désigne l'action de transformer une grandeur physique en un signal numérique ;
- **Concentrer** : permet d'interfacer un réseau spécialisé d'objets à un réseau IP standard (exemple : wifi) ;
- **Stocker** : qualifie le fait d'agréger des données brutes, produites en temps réel, méta taguées, arrivant de façon non prédictible.
- **Présenter** : indique la capacité de restituer les informations de façon compréhensible par l'homme, tout en lui offrant un moyen d'agir et/ou d'interagir [5].

I.2.3 Fonctionnement de l'IoT

L'internet of things (IoT) désigne diverses solutions techniques qui permettent d'identifier des objets, capter, stocker, traiter, et transférer des données dans les environnements physiques, L'IoT permet aussi l'interconnexion des différents objets intelligents via l'internet. De plus, pour son fonctionnement, plusieurs systèmes technologiques sont nécessaires, parmi lesquels RFID, WSN, M2M, TCP/IP, technologies mobiles, etc...

➤ RFID (Radio Frequency Identification)

Le terme RFID englobe toutes les technologies qui utilisent les ondes de la radio pour identifier automatiquement des objets ou des personnes. C'est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui

émet des ondes radiophonique.

Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance. [6]

➤ **WSN (Wireless Sensor Network)**

C'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. En effet, chaque nœud possède une capacité de traitement et peut contenir différents types de mémoires, un émetteur-récepteur RF et une source d'alimentation, comme il peut aussi tenir compte des divers capteurs et des actionneurs. Comme son nom l'indique, le WSN constitue alors un réseau de capteurs sans fil qui est une technologie nécessaire au fonctionnement de l'IoT. [6]

➤ **M2M (machine-to-machine)**

Le machine to machine comme son nom l'indique est un réseau dans lequel des objets communiquent avec d'autres objets, il est composé de capteurs ayant des capacités de détections et de mesure d'événements, et munis d'éléments de communication qui donne à un administrateur la capacité de manipuler, observer et réagir aux événements et aux phénomènes dans un environnement spécifique à partir d'une application web/mobile. Le réseau machine to machine est constitué de capteurs, d'actionneurs d'objets connectés, et de Gateway qui jouent le rôle d'un hub pour ses objets [6].

I.2.4 Les objets connectés

Définition

Un objet connecté est un appareil composé de capteurs qui envoient des données vers une application mobile ou un service web pour de multiples domaines d'application. Les objets connectés s'appuient sur les innovations technologiques majeures telles que le Cloud computing, et l'avancée en général des techniques informatiques. De plus en plus performants, ils peuvent être reliés à des services pour mesurer, partager, suivre, analyser les données quotidiennes de l'individu et lui donner des conseils. Depuis peu, les objets connectés commencent à prendre part dans notre vie quotidienne à l'image de : Ordinateurs, téléviseurs, cameras IP, les smartphones, les montres connectées et bien d'autres [6].



Figure I.4 : la lampe DAL [7].



Figure I.5: bracelet connecté [8]

Les différents éléments d'un objet connecté

❖ Les capteurs

Les capteurs sont des éléments qui transforment une grandeur physique en information électrique. Ils rendent compte de l'état du système à un moment donné. L'information est délivrée sous la forme d'un signal qui sera exploité par une interface programmable. Citons quelques exemples de capteurs :

➤ capteur GPS UBLOX NEO-6M [90]

Le module u-blox NEO-6M Global Positioning System (GPS) est un capteur GPS simple doté d'une antenne, d'un adaptateur, d'une batterie et d'un port d'extension d'antenne intégré sur une carte de circuit imprimé. Par ailleurs, il suffit de connecter quatre fils pour que ce module GPS soit opérationnel avec un Microcontrôleur Arduino par exemple.



Figure II.6 : capteur GPS UBLOX NEO-6M [9]

➤ **Capteur de température et d'humidité DHT-22 [76]**

Est un capteur numérique, à faible coût permettant de mesurer de manière efficace la température et l'humidité de l'air ambiant grâce à sa combinaison de deux en un d'un capteur d'humidité capacitif et d'une thermistance. De plus il se connecte facilement à votre microcontrôleur préféré grâce à un signal numérique sur un seul fil. Le capteur est calibré et ne nécessite pas de composants supplémentaires pour pouvoir être utilisé. [75]

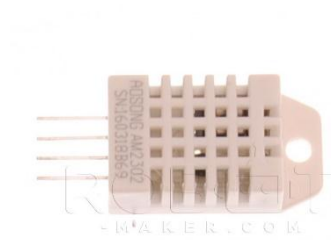


Figure II.7 : capteur DHT-22[75]

➤ **Capteur d'empreinte digitale**

La biométrie regroupe les techniques d'identification des personnes passant par analyse des traits corporels propres à chaque individu. En effet, la capture de l'image d'une empreinte digitale consiste à trouver les lignes tardes par les crêtes et les vallées. Par ailleurs la plupart des techniques de reconnaissance d'empreintes digitales sont fondées sur l'optique. En revanche, ces systèmes sont chers et peu maniables, ce qui en restreint l'usage. [10]



Figure I.8: capteur d'empreinte. [11]

➤ **Module de camera OV7670 [88]**

Le module Caméra OV7670 est un petit capteur d'images, à fonctionnement basse tension, qui offre toutes les fonctions d'une puce unique caméra VGA et un processeur d'image. Grâce à la commande de bus SCCB, le capteur peut émettre toutes les données de cadre, d'échantillonnage, et de diverses résolutions en 8 bits.

L'image VGA obtenue peut atteindre jusqu'à un maximum de 30 images par seconde. D'ailleurs, les utilisateurs peuvent totalement contrôler la qualité d'image, le format de données et le mode de transmission Interface SCCB standard compatible avec une interface I2C. [12]



Figure I.9 : Module de caméra OV7670 [12]

➤ **Capteur de pression BMP280 [81]**

C'est un capteur environnemental notamment utilisé pour capter la température et la pression barométrique et aussi une mise à niveau des capteurs BMP085 [78] /BMP180 [79]/BMP183 [80].Ce capteur est idéal pour toutes sortes de détection météorologique et peut même être utilisé à la fois en I2C qu'est un protocole utilisé par nombreux composants peut être implémenté sur n'importe quel microcontrôleur et SPI qu'est un bus bidirectionnel basé sur la communication maître-esclave. Ce module de précision de Bosch constitue la meilleure solution à faible coût pour mesurer la pression barométrique avec une précision absolue de ± 1 hPa et une température avec une précision de $\pm 1,0$ ° C. Comme la pression change avec l'altitude et que les mesures de pression sont bonnes, nous pouvons également l'utiliser comme altimètre avec une précision de ± 1 mètre. Le capteur est soudé sur une carte PCB et livré avec un régulateur 3.3V et une adaptation de niveau logique afin de pouvoir l'utiliser avec un microcontrôleur en logique 3V ou 5V sans souci. [13]

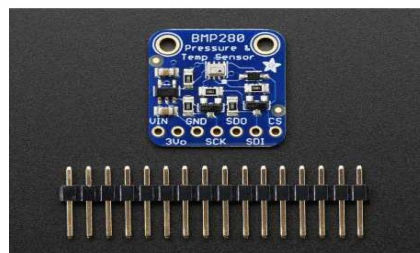


Figure I.10: capteur de pression BMP 280[14]

➤ **Capteur de mouvement PIR HC-SR501 [77]**

Est un capteur capable de détecter les mouvements. Il détecte les rayonnements infrarouges dans son champ de vision et déduit une présence ou un mouvement des humains et des animaux dans un rayon de 7 mètres. Ce capteur peut servir dans de nombreuses applications notamment dans un système d'alarme [15].



Figure I.11: capteur de mouvement PIR HC-SR501 [15]

➤ **Module RFID RC522 [89]**

La technologie RFID (radio frequency identification) est utilisée dans différents domaines, comme la traçabilité, les antivols ou l'identification d'une carte d'accès. Le lecteur RFID est capable de lire des informations contenues dans une puce spéciale (un tag). En effet, il génère un champ magnétique qui permet l'activation et l'alimentation du tag, lorsqu'il se trouve suffisamment proche (quelques centimètres). Le tag peut ensuite échanger des informations avec le lecteur (comme son numéro d'identifiant, mais il possède aussi une mémoire de quelques kilo-octets). Les tags RFID sont souvent intégrés à un porte-clés ou une carte au format carte bancaire, mais on peut aussi les rencontrer dans les objets du quotidien comme les jouets des enfants [16], la figure ci-dessous représente un capteur RFID.

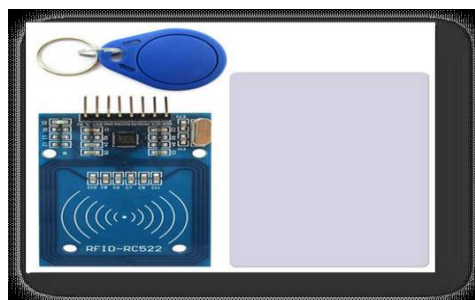


Figure I.12: lecteur RFID RC522 avec ses accessoires [16]

❖ Le microcontrôleur

Définition d'un microcontrôleur

Un microcontrôleur se présente sous la forme d'un circuit intégré qui contient en interne, l'équivalent de la structure complète d'un micro-ordinateur. C'est à dire qu'il rassemble dans un même boîtier un microprocesseur, plusieurs types mémoires et des périphériques de communication (Entrées-Sorties) [5].

Les microcontrôleurs améliorent l'intégration et le coût lié à la conception et à la réalisation d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré. Nous parlons alors de « système sur puce » (en anglais « système on chip ») [5].

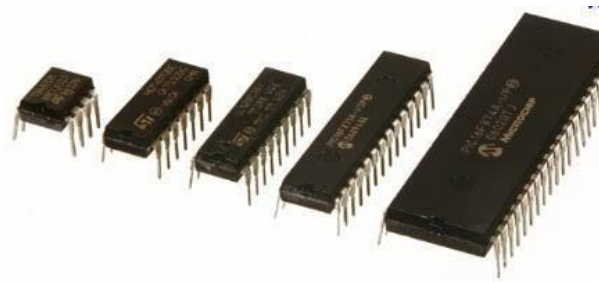


Figure I.13 : Le microcontrôleur [5]

Domaine d'application d'un microcontrôleur

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, dans de multiples industries et application, notamment dans les domaines de l'automatisation, de la fabrication, de la robotique, de l'automobile, de l'éclairage, de l'énergie intelligente, de l'automatisation industrielle et des communications de l'internet des objets [5].

Les différents éléments composant le microcontrôleur

Un microcontrôleur est constitué de plusieurs éléments qui servent à des tâches bien précises comme les montrent la figure ci-dessus :

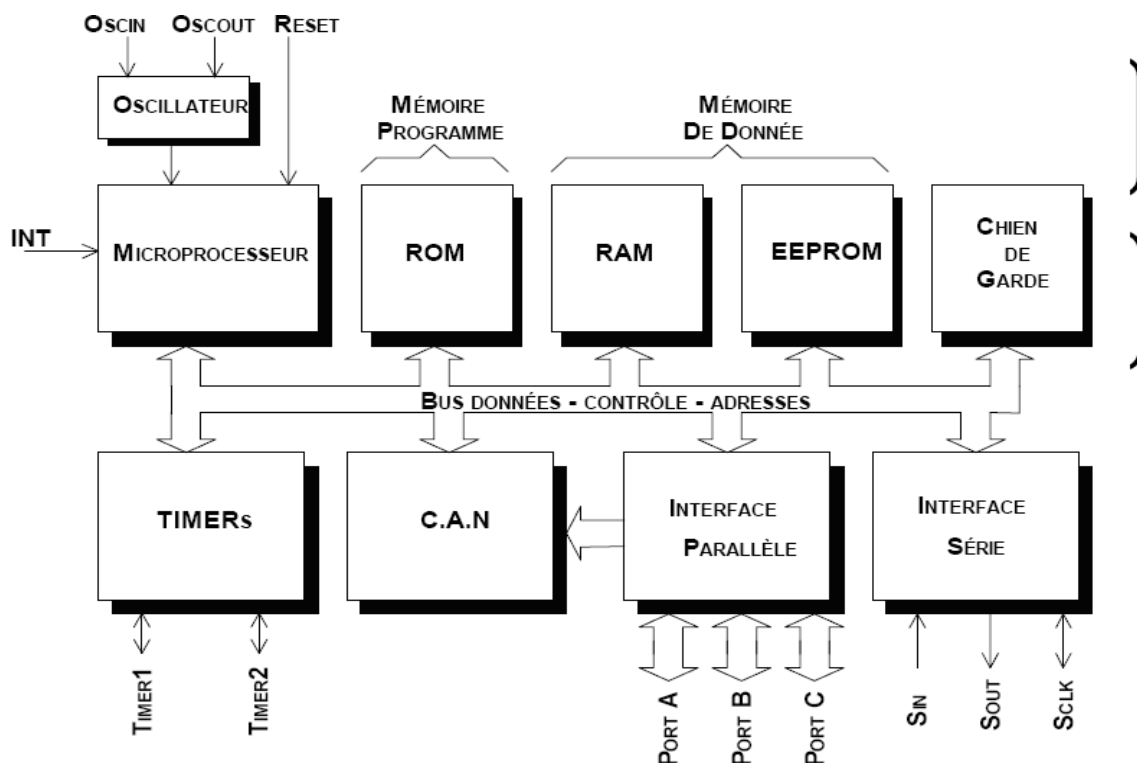


Figure I.14: Architecture interne d'un microcontrôleur [5].

➤ Le processeur ou CPU Central Processing Unit

Il est non seulement le cœur du microcontrôleur mais aussi l'équivalent du microprocesseur avec une puissance généralement moindre, la vocation n'étant pas la même. C'est cette unité centrale qui exécute le programme et le pilote ainsi que tous les autres éléments. Elle dispose généralement de deux connexions avec l'extérieur, une pour une horloge interne et l'autre pour un Oscillateur [5].

➤ **Les mémoires** : Afin de stocker un programme, des données temporaires ou des données persistantes, le microcontrôleur dispose de plusieurs types de mémoires, ayant chacune des caractéristiques particulières. Parmi ces types on trouve :

a) **Mémoires programmes** : Ce dispositif contient les instructions du programme qui doit exécuter le microprocesseur. En effet, ce type de mémoire (appelée mémoire morte), est accessible uniquement en lecture. En revanche, sa programmation nécessite une procédure particulière et un matériel adéquat. Il en existe différents types selon leur mode de programmation :

- **La mémoire morte(ROM)** : est une mémoire dont le contenu est conservé même en cas de coupure de courant. Elle contient le programme qui va exécuter

le processeur.

- **PROM** : mémoire morte programmable électriquement une seule fois par le développeur (appelée aussi OTPROM)
 - **EPROM & EEPROM** : Se sont de type mémoire morte effaçable électriquement et programmable dont à différence entre eux et que la première nécessite de retirer l'appareil et le soumettre à un rayonnement ultra-violet pour effacer la mémoire (aussi appelé UVEPROM). Par contre, la mémoire EEPROM peut être effacée par un simple courant électrique sans qu'il soit nécessaire de la retirer de l'appareil qu'il la contient.
 - **La mémoire flash (flash ROM)** : est une variété de la mémoire EEPROM rapide et effaçable pour un secteur complet, et non par une case individuelle. On trouve plusieurs interfaces pour les adresser. C'est ce type de mémoire qui est souvent utilisé dans les microcontrôleurs pour enregistrer le programme qui va gérer le fonctionnement du système technique.
- b) **Mémoire de données** : Ce dispositif permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats d'Operations, états des capteurs...). En effet, ces mémoires sont accessibles en écriture et en lecture. On en trouve 2 types [5] :
- **La mémoire vive**: Elle permet le stockage des données temporairement qui sont nécessaire à l'exécution du programme de gestion du système technique. Elle est dite « volatile » car ces données ne seront pas disponible si le système technique est mis hors tension.
 - **La mémoire morte**: Non-volatile (données conservées en cas de coupure de l'alimentation) ayant un temps d'écriture assez élevé (quelques ms) par rapport au temps de lecture qui est assez faible (quelques ns).
- **Les bus** : Un bus est un ensemble de lignes, transportant des informations codées binaires. Chacune de ces lignes est affectée d'un poids binaire. C'est par l'intermédiaire de ces lignes que s'effectuent les échanges entre les différents éléments du système. On distingue 3 types de bus : le bus de données, le bus d'adresse et le bus de contrôle [5].

- **Les périphériques** : Sont des circuits électroniques intégrés au microcontrôleur capable d'effectuer des tâches spécifiques. On peut mentionner entre autres [5]:
- **Les convertisseurs analogiques-numériques(CAN)** : donnent un nombre binaire à partir d'une tension électrique.
 - **Les convertisseurs numériques-analogiques (CNA)** :
Donnent une tension électrique à partir d'un nombre.
 - **Les générateurs de signaux à modulation de largeur d'impulsion (MLI, ou en anglais, PWM pour pulse Width Modulation).**
 - **Les timers** : Compteurs d'impulsions d'horloge interne ou d'événements externes.
 - **Les chiens de garde (Watch Dog).**
 - **Les comparateurs** : Comparent deux tensions électriques
 - **Les contrôleurs de bus de communication (UART, IIC, SSP, CAN, flextray, USB, Ethernet, etc.)**
 - **Les entrées/sorties**: Permettent au microcontrôleur de communiquer avec le monde extérieur. C'est donc là qu'elles vont être connectées aux claviers, aux afficheurs, aux poussoirs, aux moteurs et relais, etc. [5].

Critères de choix d'un microcontrôleur

Les critères principaux que nous devons retenir pour choisir un microcontrôleur sont les suivants :

- La disponibilité des circuits de famille sur le marché.
- Le prix des circuits doit être à la portée.
- La programmation de la mémoire morte doit être facile.
- Il est important de considérer la vitesse maximale, la quantité RAM ou ROM ;
- La consommation d'énergie ;
- Et enfin, les outils de développement doivent être aussi peu coûteux que possible [5].

Les types d'objets connectés

- **Objet connecté simple** : un objet connecté qui peut émettre ou transmettre des données.
- **Objet connecté intelligent** : un objet connecté qui peut effectuer des tâches spécifiques traiter ou analyser des données. Ce type d'élément est équipé de capteurs, microcontrôleur, semi-conducteurs et des logiciels qui font ces tâches. Sa capacité réside dans le fait de pouvoir traiter les informations et produire des résultats après le traitement et l'analyse.
- **Objet autonome** : il a toutes les caractéristiques d'un objet connecté simple et intelligent. De plus, il peut faire plus de tâches comme analyser ses données, étudier son environnement pour agir à la fin [5].

I.2.5 Les avantages de l'internet des objets

Les principaux bénéfices que peuvent avoir un système doté d'un système IOT sont:

- **Amélioration de la productivité** : l'IoT permet la surveillance, le monitoring et le contrôle de différentes tâches, ce qui optimise les différentes opérations qui augmentent la productivité et l'efficacité
- **Analyses prédictives** : grâce à la collecte de nombreuses données, les nouvelles technologies de l'IoT permettent d'examiner les patrons récurrents et contribuent à l'analyse prédictive qui peut être principalement utilisée en maintenance. Ces informations précises vont servir à améliorer les tâches et les services existants
- **Rapidité d'action** : les données permettent de suivre en temps réel et même à distance les systèmes mis en place. Elles facilitent l'optimisation des interventions de maintenance, mais aussi donnent un avantage stratégique à l'entreprise dans le suivi de l'évolution des marchés.
- **Diminution des erreurs humaines** : grâce à la complémentarité des technologies comme l'intelligence artificielle, l'IoT permet de minorer les erreurs humaines dues à des tâches répétitives [18].

I.2.6 Les inconvénients de l'IoT

Les enjeux des cyberattaques sont élevés pour les consommateurs lorsqu'il est question d'IoT, et il est essentiel de comprendre et de contrôler les nouveaux types de

conséquences. Citons par exemple :

- Une attaque contre un appareil médical peut induire des dommages physiques, voir la perte d'une vie.
- Le brouillage d'un capteur utilisé pour des compteurs intelligents pourrait provoquer une panne de courant et générer divers inconvénients causés par le manque d'électricité.
- Le piratage d'une voiture sans chauffeur risquerait de se traduire par une perte de contrôle du véhicule sur une autoroute bondée.
- L'exploitation des faiblesses d'un appareil IoT pourrait permettre à un individu malveillant d'accéder à des données hautement sensibles telles que des renseignements bancaires, personnels ou de santé.
- Un piratage de système dans une smart home peut engendrer un dysfonctionnement total à l'intérieur de la maison (ouverture des portes, lancer les matériels de la maison sans contrôle) [19].

I.2.7 La sécurité dans l'IoT

La sécurité est l'élément le plus important qui pose des obstacles devant l'adaptation des êtres humains des applications de l'IoT suite aux menaces concernant la vie privée des utilisateurs.[1].En effet ,sécuriser un système informatique implique directement la garantie des objectifs suivants :

✓ L'Authentification

En contrôlant et identifiant les nœuds au sein d'un réseau, l'authentification permet la coopération entre eux. Cette phase précède toutes les autres phases, et on ne peut jamais assurer une confidentialité et intégrité de donnée si l'authentification est mal gérée, et un simple attaquant peut rejoindre le réseau et injecte des messages erronés. L'utilisation d'un code d'authentification de message MAC3 permet d'assurer l'authentification de l'origine et l'intégrité de message. [1]

✓ L'intégrité

Les données reçues par le nœud récepteur doivent être identiques à celles envoyées par le nœud émetteur sans aucune altération. Pour chaque message envoyé, une empreinte digitale est générée par une fonction de hachage pour assurer l'intégrité. [1]

✓ La confidentialité

Cette propriété est assurée par l'utilisation de clés cryptographiques (symétrique ou asymétrique). Elle consiste à préserver le secret du message échangé et de ne pas le révéler aux adversaires. [1]

✓ La disponibilité

Le réseau doit être disponible à tout moment et autorise les parties communicantes à utiliser du medium quand c'est nécessaire. [1]

✓ La fraîcheur

Les données échangées doivent être actuelles, et ne sont pas des réinjection des échanges précédents interceptés par des attaquants.

Les propriétés citées précédemment constituent un défi majeur devant l'Internet des Objets avant que ce paradigme devient une réalité. La vulnérabilité des objets de l'IoT aux attaques de sécurité est due à l'impossibilité d'appliquer des standards de sécurité classiques à cause de la densité importante du réseau et sa topologie dynamique, la limite des nœuds en ressource (stockage, calcul, énergie) et le type de communication (sans-fil) [1]

I.2.8 Les challenges de l'IoT

Dans l'IoT, tout objet doit être identifiable, capable de communiquer et d'interagir avec l'environnement auquel il appartient. Pour satisfaire ces contraintes, de nouvelles techniques doivent être mises en place avec les caractéristiques des objets connectés. Parmi les challenges les plus importants de l'IoT nous citons :

- **Déploiement de l'IPv6**

Le nombre d'entités à connecter dans l'IoT augmente d'une manière exponentielle. Et vu la limite de l'IPv4 en termes d'espace d'adressage, il est primordial de recourir à un adressage IPv6. De plus, la taille des identifiants du RFID (64 ou 96 bits) est une contrainte pour l'intégration de ces derniers dans IPv6. [1]

- **L'auto-configuration**

L'auto-configuration et l'adaptation avec l'environnement est un défi majeur de l'IOT vu que le nombre important d'objets à connecter et leur hétérogénéité et la limite de la configuration manuelle. [1]

I.2.9 Les protocoles de communication dans l'IoT

Les protocoles de communication permettent d'effectuer l'échange d'information sur un moyen de communication physique [20].

- **HTTP** : est un protocole omniprésent dans l'Internet, qui permet d'échanger des données, par exemple lors de la consultation de sites Web.

Il est cependant :

- Assez verbaux.
- plus adapté au API haut niveau.
- Communication entre applications.
- Plus lourds (données).
- Sécurité : HTTPS pour version sécurisée [20]

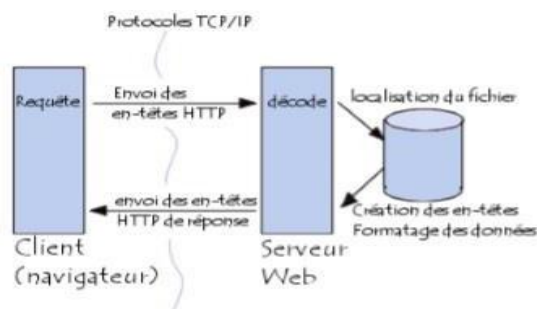


Figure I.15: le protocole http [21].

- **MQTT** : est un protocole alternatif très utilisé dans l'Internet des Objets. En effet, MQTT (pour Message Queuing Telemetry Transport) est un protocole adapté aux connexions qui intègrent une partie mobilité entre les clients et le serveur qui va stocker les données [20].
- **COAP** (Constrained Application Protocol): est un protocole de transport de données utilisé dans l'Internet des Objets. Ce protocole s'adresse davantage à des environnements contraints qui possèdent peu de bande passante pour échanger des données. En effet, le nombre d'octets est limité par envoi et la surcharge du protocole est très faible. C'est donc un protocole très utile pour les environnements contraints avec des équipements à faible capacité de traitement et de transfert [20].

I.2.10 Les moyens de communication dans l'IoT :

Les moyens de communication dans l'internet des objets sont devisés en trois parties selon la portée.

Les technologies à courte portée :

➤ Le Bluetooth :

- Haut débit, faible portée, quasiment intégré dans tous les appareils du quotidien.
- Si la sécurité n'est pas bien configurée alors l'objet s'expose à des failles de sécurités.
- Adapté aux technologies telles que les oreillettes pour la communication par voix aux raccordements d'équipements à un ordinateur. [20] [22]

➤ Zigbee :

- Technologie peu consommatrice en énergie et s'intègre à bas coût dans les équipements.
- Couverture réseau limitée à une faible zone autour de la passerelle. Achat d'appareils spécifiques car la technologie n'est pas disponible dans les smartphones et ordinateurs.
- Adapté aux applications de contrôle de commandes dans les contextes bureautiques et domotiques [20] [22].

➤ Le protocole NFC (Near Field Communication) :

- Technologie facile à utiliser et à mettre en place. Idéal pour échanger de courtes informations.
- Contrairement au Bluetooth, la technologie n'est pas encore disponible dans les objets du quotidien tels que les smartphones.
- Adapté pour les situations de badge (paiement sans contact, titre de transport, contrôle d'accès...) [20] [22].

Les technologies à moyenne portée :

➤ Zwave :

- Adapté aux objets alimentés par batterie et communiquent à bas débit.

- Technologie adaptée à des besoins très spécifiques. Achat d'appareils spécifiques.
- Spécifiquement conçu pour les applications de domotique [20] [22].
 - Wi-Fi :
- Très Haut débit, qualité du signal assurée, connexion simple et rapide à la passerelle.
- Non adapté aux objets alimentés uniquement par batterie. Couverture réseau limitée à une faible zone autour de la passerelle. Une passerelle Wi-Fi mal configurée expose le réseau à des failles de sécurité (man in the middle).
- Adapté aux applications de domotique, au contexte indoor [20] [22].

Les technologies à longue portée :

- Cellulaire (GSM, 3G, 4G) :
 - ❖ GSM(Global System for Mobile communications) :
 - Fonctionnalité multimédia (SMS, MMS), accès internet et introduction de la carte SIM
 - Portée faible en zone rurale
 - Appels vocaux, messages courts, navigation (partielle) [22].
 - ❖ 3G :
 - Sécurité élevée, itinérance internationale.
 - Consommation élevée d'énergie, couverture de réseau faible, coût élevé de licence de spectre.
 - Vidéoconférence, TV sur mobile, GPS [20] [22].
 - ❖ 4G :
 - Très haut débit, haute disponibilité, très bonne couverture géographique, gestion de la mobilité.
 - Consommation importante en énergie, prix de l'abonnement important [22].
 - Applications à très haut débit et très grandes volumes de données (voix, vidéo). Adapté à la télésurveillance, aux smartphones (IoT critique) [20] [22].

➤ Réseaux radio bas-débit (LORA, SIGFOX) :

❖ LORAWAN :

- Bas débit, faible puissance, prix bas pour l'abonnement longue portée, économie de la batterie (>10ans).
- Utilisation d'un modem spécifique, ne gère pas la mobilité des objets, délai de livraison d'une donnée à l'objet très variable (jusqu'à plusieurs jours).
- Adapté à la télémétrie à bas débit. Ce qui englobe plus généralement l'IoT des masses [20] [22].

❖ SIGFOX :

- Très bas débit, faible puissance, prix bas pour l'appareil et l'abonnement ,longue portée, économie de la batterie (> 10 ans).
- Dépendance à une technologie propriétaire, utilisation d'un modem spécifique, ne gère pas la mobilité des objets, délai de livraison d'une donnée à l'objet très variable (jusqu'à plusieurs jours).
- Uniquement adapté à la télémétrie à très bas débit [20] [22].

❖ Fibre optique :

- Fourni le plus haut débit du marché et très stable car insensible aux perturbations électromagnétiques. Prix de l'abonnement pas plus cher que celui de l'ADSL [20] [22].
- Déploiement national en cours. Raccordement à la fibre optique difficile et couteux pour une mise en production en masse.
- Adapté aux applications à fort volume de données (téléchargement de fichiers, TV en 3D...) [20] [22].

I.3 Domaines d'application

➤ **Domaine médical ou E-Health**

L'Internet des Objets joue un rôle important dans le domaine médical. Un réseau de miniatures capteurs avalés ou implantés sous la peau, peut être déployé pour assurer la surveillance des patients (température, pression du sang, activités de respiration, rythme

cardiaque, etc.) indépendamment de leurs environnements. Ce réseau communique l'état de santé du patient au médecin en temps réel via une application mobile qui joue la passerelle entre le réseau corporel et l'internet [1].



Figure I.16: services de la santé [23]

➤ Domotique ou Smart Home

L'IOT permet l'accès à un monde ambiant, où les objets domestiques collaborent entre eux pour améliorer l'habitat humain. Ce concept permet de gérer tous les équipements (tâches ménagères, verrouillage des portes / fenêtres, vidéo surveillance, etc.) en-site ou à distance via un réseau de communication. [1]



Figure I.17: Services de la domotique [1]

➤ Réseaux électriques intelligents ou Smart Grid

Un réseau électrique intelligent est un réseau amélioré grâce à l'intégration des énergies renouvelables et des TIC en garantissant la communication bidirectionnelle entre les différents acteurs du réseau et en temps réel. Il vise à garantir la disponibilité, l'efficacité, le coût réduit et la sécurité de l'énergie fournie [1].

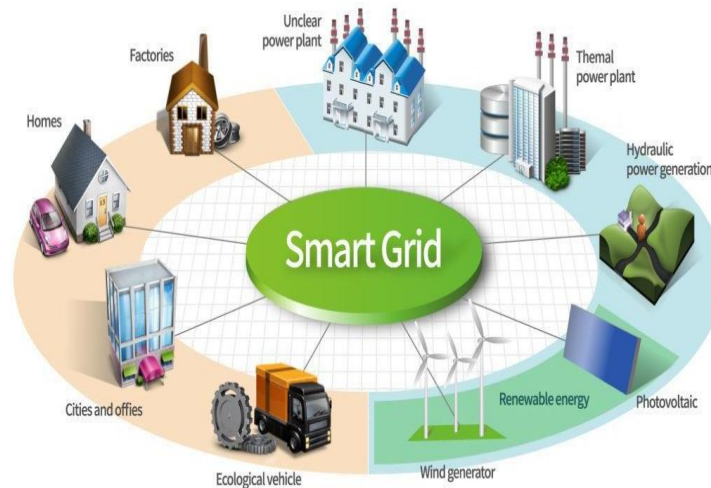


Figure I.18 : Services du Smart Grid [1]

➤ Transport

Les cartes touristiques peuvent être équipées par des balises à base de la technologie NFC2, et permettre au Smartphone de récupérer via le web les informations qui peuvent intéresser les touristes (hôtels, restaurants, événements) [24].



Figure I.19 : IOT dans les transports [1]

➤ L'industrie

Le problème des pièces non homologues résultant de l'utilisation des pièces qui ne répondent pas aux exigences (Suspected Unapproved Parts) particulièrement dans le secteur de l'aéronautique peut être résolu à l'aide de l'IOT en marquant chaque pièce avec des tags contenant des identifiants et permettant la récupération automatique des informations relatives à la pièce [1].

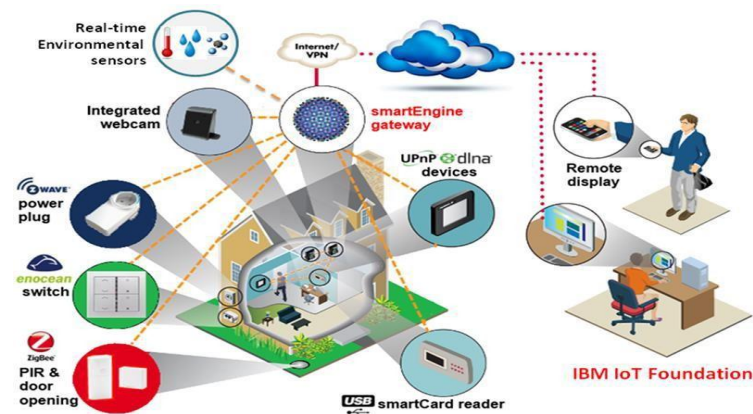


Figure I.20 : IOT dans l'industrie [1]

➤ L'agriculture

Depuis plusieurs années, les agriculteurs utilisent des outils informatiques pour accomplir différentes tâches, administratives d'abord, puis techniques (robots de traites, outils d'aide aux réglages des machines, cartes satellites pour épandage d'azote, prévision météorologique, cours des matières premières, ...). Mais depuis quelque temps, on commence à pouvoir connecter « sans fil » tous ces systèmes les uns avec les autres, à les faire fonctionner ensemble pour obtenir un service supplémentaire. A l'inverse d'un outil attelé mécaniquement au tracteur [25].

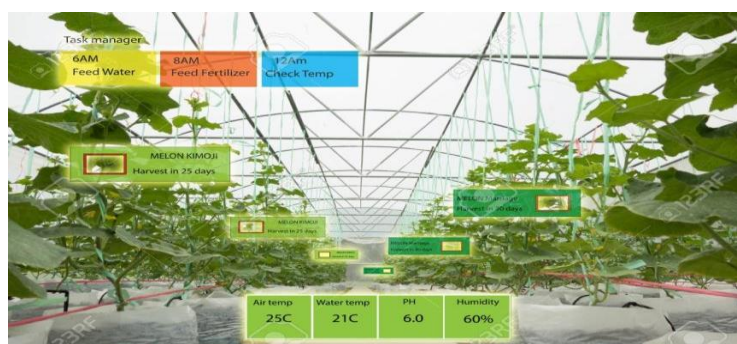


Figure I.21 : IOT dans l'agriculture [26].

➤ Le militaire

Les drones militaires font l'objet d'un engouement croissant. Contrairement à certaines idées reçues, les systèmes opérationnels de drones sont des systèmes relativement coûteux. Pour remplacer ou compléter les avions pilotés, les missiles de croisière ou les satellites. Les drones sont de plus en plus utilisés pour des missions dangereuses (reconnaissance à basse altitude, désignation d'objectif, suppression de la défense aérienne adverse) ou de très longue durée, les « pilotes » de ces engins pouvant, du fait qu'ils restent au sol, se relayer. Les Anglo-Saxons parlent de « missions 3D », pour dull, dirty and dangerous (missions ennuyeuses, sales et dangereuses). Les drones peuvent aussi remplacer ou compléter les satellites pour des missions de reconnaissance stratégiques ou d'alerte avancée (détection de départ de missiles), lorsque les qualités de permanence sur zone et de transmission en temps réel sont des critères essentiels. En effet, ils sont capables d'assurer une surveillance permanente de zones ciblées et de transmettre en temps réel leurs données, ce qui n'est pas toujours le cas des satellites. En revanche, les drones doivent respecter les traités relatifs à la légitimité de pénétration des espaces aériens nationaux, ce dont s'affranchissent les satellites. On distingue de différentes catégories de drones militaires, en fonction de leur taille et de leur masse, de leur rayon d'action et de la nature des charges utiles qui les équipent [27]



Figure I.22 : l'IOT dans le militaire [28].

I.4 La conclusion

Dans ce premier chapitre, nous avons présenté brièvement l'état de l'art de l'IoT, les domaines d'applications de l'IoT et l'apport des langages web sur l'IoT. Le chapitre suivant sera consacré aux services web et IoT.

Chapitre

N°02

II.1 Introduction

En quelques années, les services web sont devenus le nouveau point de convergence technologique de l'industrie du logiciel. En effet, Un bon nombre d'éditeurs de serveurs d'applications, d'outils de développement, ont placé ces services web au cœur de leur stratégie. Dans ce chapitre, nous décrivons L'apport des langages web dans l'IoT ainsi les technologies liées aux services web ensuite nous définirons Intergiciel et le WoT pour *web of things*.

II.2 L'apport des langages web dans l'IOT

Le service Web représente la technologie permettant l'échange de données entre les différentes plateformes et leurs applications, bien qu'elles puissent être hétérogènes. En effet l'échange de données nécessite l'utilisation d'un logiciel intermédiaire, l'Intergiciel. Ce dernier est un logiciel qui offre une interface d'échange de données entre les objets dans l'Internet. Finalement, un objet peut, non seulement être intégré dans l'Internet, mais aussi sur le Web, plus précisément sur une extension de l'IOT nommée le WOT. [29]

II.2.1 Services Web

Étant donné que « les objets connectés sans services Web associés meurent », dans notre modèle, nous recourons aux services Web pour contourner les problèmes de normes liés à l'interconnexion, ce qui permet un accès souple à des plateformes logicielles et matérielles via Internet. [29]

➤ Définition des services Web

Les services Web sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires Ils fonctionnent sous le format XML, et peuvent être conçus sur deux grands modèles : XML-RCP et SOAP. La différence entre les deux est que le modèle SOAP est conçu à partir des objets que l'on veut transmettre en langage informatique sans prendre en compte de langage de programmation spécifique. Le principe reste le même : le client dispose d'une application à laquelle il transmet une demande. Cette application est traitée à distance. Les web services peuvent être mis au point pour présenter des systèmes de classement, des modèles de calculs, des catalogues ou des données clientèle par exemple. Et des protocoles Internet standards. Les services Web sont des

compléments aux programmes et applications existants, développés à l'aide de langages tel que Visual Basic, C, C++, Java ou autre, et servent de pont pour que ces programmes communiquent entre eux. Plusieurs autres définitions des services Web peuvent être retenues. De ce qui précède, on peut conclure que le service Web est un service électronique, offert par des technologies universelles développées en fonction des protocoles réseautiques mondiaux (essentiellement d'Internet) qui agit comme infrastructure de communication. [29]



Figure II.1 : les services du web [69]

➤ Architecture du service Web

Les services Web permettent, à travers un espace de communication, d'échanger différentes informations. Dans cette section, on tente de comprendre comment cet échange de données se fait en présentant un modèle d'architecture classique pour les services Web appelés SOA (Architecture orientée service). Comme le montre la Figure 2.1 ci-dessous, le modèle SOA met en interaction trois acteurs : un client, un fournisseur et un intermédiaire. [29] Le modèle SOA fait intervenir les trois acteurs suivants :

- 1) **le client** : qui est le demandeur du service représenté par l'application, qui va chercher et consommer le service;
- 2) **le fournisseur**: qui est le propriétaire du service offert dans la plateforme d'accueil;

3) **l'annuaire des services** : un intermédiaire entre le client et le fournisseur, qui offre au fournisseur la possibilité de publier ses services, et au client la capacité de localiser les services répondant à ses besoins. Le fonctionnement des services Web se base sur un modèle composé de trois phases fondamentales (la publication, la découverte et l'interaction) [29].

- **Publication**

Après avoir structuré sémantiquement son service, le fournisseur passe ensuite à la publication de ce service afin d'être découvert puis utilisé par d'autres utilisateurs ou machines. La publication du service se fait par l'UDDI (Universal Description, Discovery, and Integration), un annuaire de services fondé sur XML. Toutefois, avant qu'il soit publié, le service doit être décrit, cette description se faisant dans le langage WSDL (Web Service Description Language) [29].

- **Découverte**

Une fois la publication du service est faite, le client peut alors interroger l'annuaire (UDDI) en utilisant des mots clés permettant l'obtention d'un ensemble de descriptions (WSDL) qui contiennent toutes les informations nécessaires à l'invocation du service [29]

- **Interaction**

Aussitôt que la description du service publiée dans l'annuaire correspondant à la requête lancée par le client, ce dernier aura une interaction directe avec le service par l'intermédiaire de SOAP (Simple Object Access Protocol) qui invoque les descriptions du service Web [29].

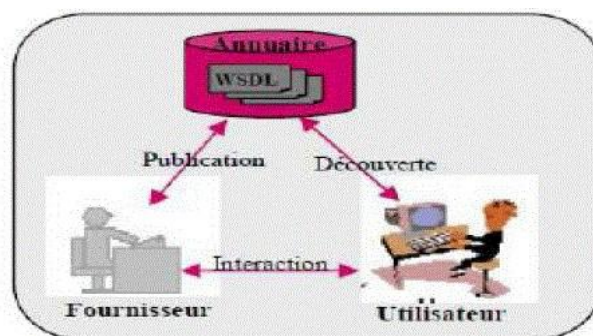


Figure II.2: Architecture classique des services Web [29]

II.2.2 Intergiciel

➤ Définition de L'intergiciel

Un intergiciel est un logiciel de communication qui permet aux processus exécutés sur une ou plusieurs machines d'interagir à travers un réseau. » [29].

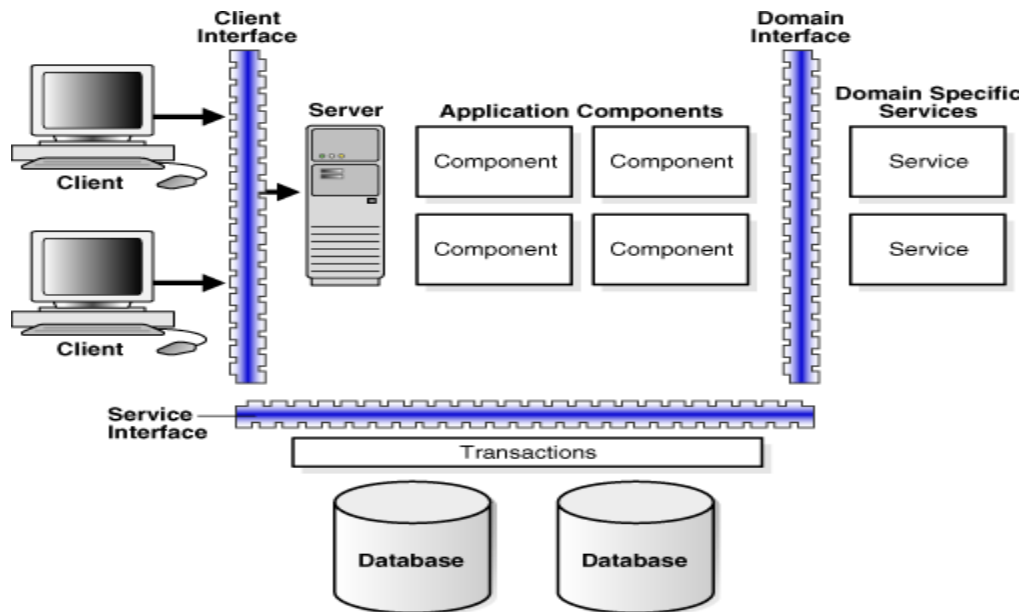


Figure II.3: Architecture Middleware [29]

➤ Utilité de l'intergiciel

L'intergiciel est une classe de logiciels permettant la communication entre des applications qui n'étaient pas conçues pour dialoguer entre elles. Il homogénéise l'accès à l'ensemble des services proposés par le réseau. Les intergiciels sont utilisés pour supporter des applications complexes et distribuées. Ils permettent aussi de lier deux ou plusieurs logiciels entre eux afin que les applications correspondantes puissent échanger des données. L'intergiciel rend le développement d'une application plus facile en fournissant une abstraction de programmation commune, en masquant l'hétérogénéité et la distribution du matériel et du système d'exploitation, et en masquant les détails de programmation de haut niveau.[29]

II.2.3 WoT web of things:

➤ **Définition de WOT:**

L'IoT est considéré parmi les domaines de recherche les plus actifs. Avec l'exploration des différents points de vue et l'analyse des différentes méthodes, on constate que le développement rapide et la propagation phénoménale de l'IOT, en citant comme exemple la renaissance du Web of Things (WOT), qui permet principalement de réduire les barrières entre le monde physique et le monde virtuel. Le WOT propose de faire connecter les dispositifs physiques par les protocoles et les normes du Web et de les utiliser comme des ressources dans le développement d'applications Web. Le WOT peut se définir comme étant « un concept informatique qui décrit un avenir où les objets du quotidien sont entièrement intégrés avec le Web. Les systèmes informatiques qui permettent la communication avec le Web en sont une condition préalable. Ces dispositifs intelligents seraient alors en mesure de communiquer les uns avec les autres en utilisant les standards du Web existants. » [29]

➤ **Spécifications de WOT**

D'après les définitions du WoT, la solution d'inclure l'architecture du Web classique à des objets physiques connectés à l'Internet rend leurs contributions plus intelligentes et efficaces dans notre vie physique et virtuelle. Ainsi, le WOT permet d'exploiter des données enregistrées par des ressources embarquées grâce aux services Web, tels que le protocole HTTP et l'URI, qui présentent des spécifications pour le WOT. En outre, l'URI (Uniform Resource Identifier) est une chaîne de caractères structurée permettant d'identifier de manière unique une ressource dans un espace de nom donné. Cette ressource peut être désignée soit par un URN (Uniform Resource Name) ou par une URL (Uniform Resource Locator). [29]

II.3 La problématique

Une entreprise est une organisation ou une unité institutionnelle, mue par un projet décliné en stratégie, en politiques et en plans d'action, dont le but est de produire et de fournir des biens ou des services à destination d'un ensemble de clients ou d'usages, en réalisant un équilibre de ses comptes de charges et de

produits. Pour se faire, une entreprise mobilise, consomme et fait appel à des ressources (matérielles, humaines, financières, immatérielles et informationnelles) ce qui la conduit à devoir coordonner ces fonctions. Par conséquent, la gestion de cette dernière n'est pas facile car il faut contrôler la totalité des fonctionnalités offertes. En premier, parmi les différents départements et directions dans une entreprise, la direction des ressources humaines figure sans nul doute parmi les plus complexes. Pour cause, gérer la dimension humaine s'avère une tâche tordue dans la mesure où, malgré l'existence des règlements et des systèmes de gestion de base, chaque personne avec qui on traite est différente, à la fois gérant, négociateur et médiateur. Aussi, la gestion de la production est l'ensemble des activités qui participent à la conception, la planification des ressources (matérielles, financières, ou humaines), leur ordonnancement, l'enregistrement et la traçabilité des activités de production, le contrôle des activités de production de l'entreprise. L'objectif est d'améliorer de façon continue la gestion des flux des stocks inclus dans la chaîne de travail qui débute avec les fournisseurs et se termine chez les clients intermédiaires ou finaux. L'ensemble de ces activités doit être réalisé dans le respect des procédures établies (implicitement ou explicitement) par l'entreprise et tenir compte à la fois de la qualité de ses produits ou de ses services, mais aussi de la sécurité de ses salariés ou de son environnement. [30]

Une bonne partie de l'activité d'une entreprise consiste à communiquer, enregistrer, rechercher de l'information, effectuer des calculs, et créer des documents. La plupart des entreprises aujourd'hui sont informatisées, l'informatique est un outil indispensable dans l'entreprise pour pouvoir la gérer. Les entreprises doivent aujourd'hui être vigilantes, constamment attentives au mouvement de leur environnement et la veille est une systématisation du processus de repartage, de validation, d'analyse, de synthèse et de diffusion de l'information utile. [31]

II.4 Objectif et contribution

L'objectif principal de notre travail consiste à présenter un système IoT de bout en bout pour le suivi temps réel des processus de production, de géolocalisation et

de surveillance .Un système capable de favoriser et mettre en œuvre les fonctionnalités de l'Internet of Things (IoT). Pour réaliser notre projet de recherche et parvenir à proposer notre modèle, nous avons choisi de l'aborder sous l'angle des services Web et de l'internet of things (IoT).

Dans la présente recherche, nos principales contributions sont les suivantes :

- Proposition d'un système IoT de bout en bout pour le suivi temps réel des processus de production, de géolocalisation et de surveillance;
- Développement d'une application web pour le suivi temps réel des données environnementales notamment la température et l'humidité pour faciliter la visualisation données capté par la carte arduino uno en utilisant le capteur DHT22 de température et d'humidité afin d'exploiter les différents services et technologies du monde de l'IoT.

II.5 Conclusion

Un service web n'est qu'une transaction accessible par l'échange de documents XML entre deux URL .Ce dernier offre une réelle souplesse d'utilisation et permet l'accélération du développement d'application. En adoptant les services web, on facilitera l'interopérabilité et la réutilisabilité du code. Dans ce présent chapitre nous avons décrit l'apport du web dans l'IoT ainsi ces services offert .Dans le chapitre suivant sera consacré au matériel et méthodes utilisé dans notre mémoire.

Chapitre

N°03

III.1 Introduction

L'objectif principal de notre travail consiste à la conception d'un système IoT complet de bout en bout pour le suivi temps réel des processus de production, essentiellement la température, l'humidité et la pression, de géolocalisation et de télésurveillance. Pour cela, nous avons utilisé les modules d'Arduino et Raspberry qui nous faciliteront notre tâche en leur associant une carte GSM avec différents capteurs de température/humidité, pression Le premier critère de choix des modules Arduino et raspberry réside dans la simplicité de programmation de ces derniers et leurs pertinentes performances, ce qui réduit énormément le coût de la réalisation. Et pour le développement de notre application temps réel, nous avons utilisé NodeJS pour permettre de développer rapidement des applications avec d'importants volumes de données en temps réel dans les systèmes embarqués.

III .2 Rôle de NodeJS dans les applications embarquées

III.2.1 La naissance de NodeJS

Dans les années 90, on parlait de DHTML (Dynamic HTML). On utilisait en fait les toutes premières versions de JavaScript pour faire des petits effets dans ses pages web par exemple, afficher une image lors d'un clic sur un bouton. C'était l'époque de Netscape et d'Internet Explorer 5.5 .Après dans les années 2000, on a commencé à utiliser le langage pour créer des interfaces côté client. C'est là que des bibliothèques comme jQuery ou Mootools sont apparues. Aujourd'hui, cet usage de JavaScript est très répandu et mature. On a pris l'habitude de manipuler le DOM (Document Object Model) pour affecter ses balises HTML en JavaScript et leur faire subir toutes sortes de traitements. Puis, aux alentours de 2010, JavaScript est entré dans une nouvelle ère. En effet, Google a commencé à rendre le langage beaucoup plus rapide avec l'apparition du navigateur Google Chrome. Avec celui-ci est né le moteur d'exécution V8 qui a considérablement permis d'accélérer l'exécution de code JavaScript. Des outils comme NodeJS sont ensuite apparus. [62]

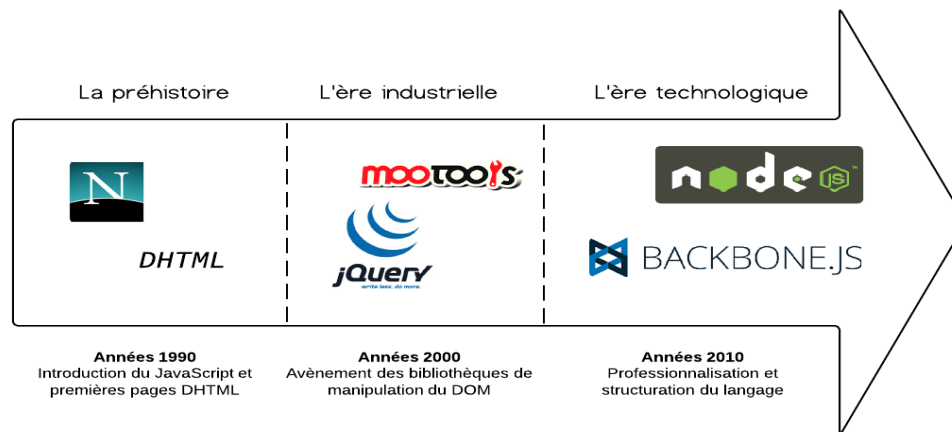


Figure III.1 : les trois vies de JAVASCRIPT. [62]

III.2.2 NodeJS et l'IoT

La caractéristique clé de toute application IoT est la collecte de données, la communication, l'analyse et l'action. NodeJS est un partenaire idéal pour toutes les fonctionnalités ci-dessus. Les sockets et le protocole MQTT sont bien adaptés aux NodeJS qui sont généralement utilisés pour la transmission de données continue dans les applications IoT. Les cartes de câbles pour applications IoT, telles que Intel Edison, BeagleBone Black et Raspberry Pi, peuvent facilement installer NodeJS comme environnement de programmation. NodeJS est livré avec le gestionnaire de packages NPM qui contient de nombreux modules IoT utiles, qui peuvent être utilisés pour un développement d'applications rapide et robuste. NodeJS est connu pour sa vitesse, son évolutivité et son efficacité, ce qui en fait le joueur clé pour les applications en temps réel gourmandes en données. Cela rend NodeJS bien adapté à l'IoT qui repose sur un trafic en temps réel gourmand en données. Cet environnement d'exécution JavaScript basé sur le moteur JavaScript V8 de Google Chrome utilise un modèle d'E / S non bloquant, piloté par les événements, qui le rend léger et efficace. Node.js est un endroit pour exécuter le logiciel d'application IoT (environnement d'exécution) construit à l'aide d'une technologie pour traduire JavaScript en code que les machines comprennent (Google V8) qui est capable de gérer beaucoup de choses qui se produisent à la fois (une entrée / sortie asynchrone pilotée par les événements).

➤ Moteur JavaScript V8 de Google Chrome

Node.js utilise le moteur d'exécution ultrarapide V8 de Google Chrome. Ce moteur V8 avait fait beaucoup parler de lui à la sortie de Google Chrome, car c'est un outil open source créé par Google qui analyse et exécute du code JavaScript très rapidement. Jusqu'à la sortie

de Chrome, la plupart des navigateurs lisaient le code JavaScript de façon peu efficace : le code était lu et interprété au fur et à mesure. Le navigateur mettait beaucoup de temps à lire le JavaScript et à le transformer en code machine compréhensible pour le processeur. Le moteur V8 de Google Chrome, qui est réutilisé ici par Node.js, fonctionne complètement différent. Très optimisé, il fait ce qu'on appelle de la compilation JIT (Just In Time). Il transforme le code JavaScript très rapidement en code machine. [62]

➤ Modèle d'E/S non bloquant

En effet, en JavaScript, presque toutes les opérations sont **non bloquantes**, cela inclut les requêtes HTTP, les opérations de lectures/écritures sur le disque, les appels à la base de données, etc. Les opérations s'exécutent de façon linéaire l'une après l'autre sans devoir attendre le résultat de l'opération précédente, une fonction de **callback** permet de récupérer la réponse de l'opération en question une fois achevée. [63]

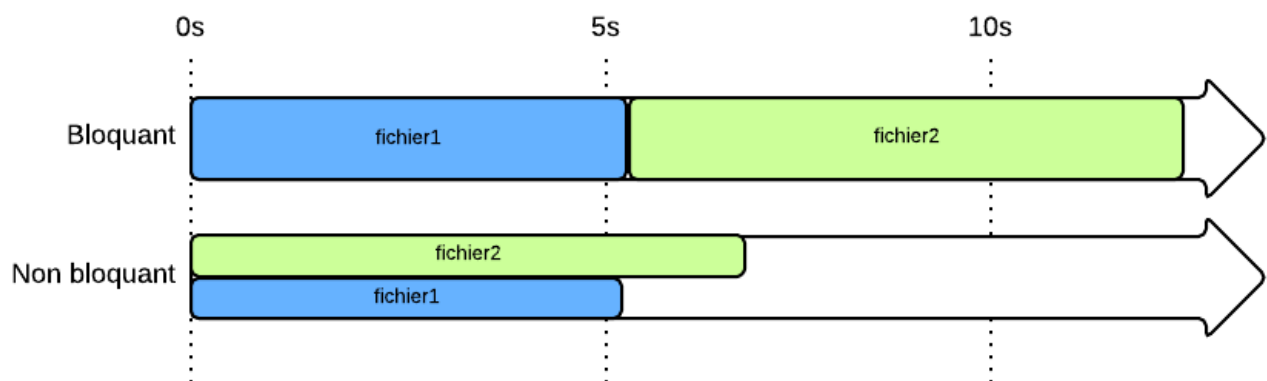


Figure III.2 : l'exécution non bloquante. [62]

III.2.3 Systèmes embarqués

➤ Définition des systèmes embarqués

Un système embarqué est un système complexe qui intègre du logiciel et du matériel, conçus ensemble afin de fournir des fonctionnalités diverses. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Par ailleurs, le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (*Personal Computer*). En effet, un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran d'ordinateur. Contrairement à un PC, l'interface IHM (Interface Homme machine) d'un

système embarqué peut être aussi simple qu'une diode électroluminescente LED (*Light Emitter Diode*) qui clignote ou aussi complexe qu'un système de vision de nuit en temps réel ; les afficheurs à cristaux liquides LCD (*Liquid Crystal Display*) de structure généralement simple sont couramment utilisés. Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables FPGA (*Field Programmable Gate Array*), des circuits dédiés à des applications spécifiques ASIC (*Application Specific Integrated Circuits*) ou des modules analogiques sont utilisés. Le logiciel a une fonctionnalité fixe à exécuter et spécifique à une application. L'utilisateur n'a pas la possibilité de modifier les programmes. Bien souvent, il n'a pas conscience d'utiliser un système à base des microprocesseurs. Les systèmes embarqués sont désormais utilisés dans des applications diverses telles que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, etc. [32]

➤ Caractéristiques

Les caractéristiques principales d'un système électronique embarqué sont :

- **Autonomes.** Une fois enfouis dans l'application, ils ne sont (le plus souvent) plus accessibles
- **Temps réel.** Les temps de réponses de ces systèmes sont aussi importants que l'exactitude des résultats.
- **Réactifs.** Il doit réagir à l'arrivée d'informations extérieures non prévues [32]

➤ Architecture générale et modes de fonctionnement

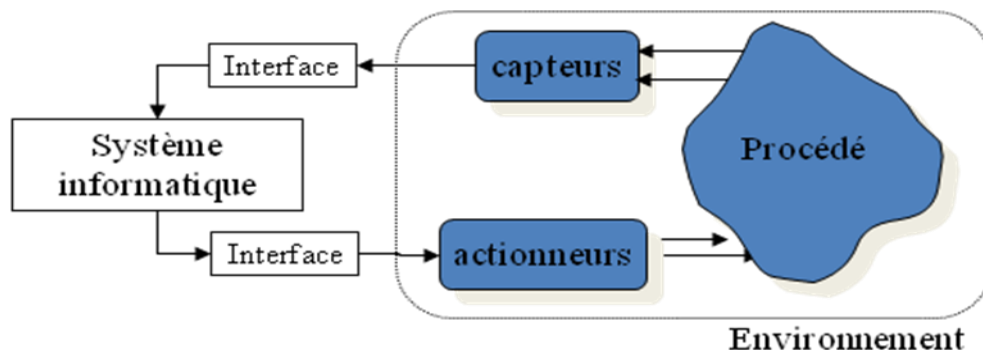


Figure III.3 : Architecture générale et modes de fonctionnement [32]

III .3 Modules et Node-Package-Manager

Malgré la taille minuscule du noyau de Node.js, ce dernier est très riche grâce à son extensibilité. Ces extensions de Node.js sont appelées modules. En effet, Il en existe des milliers qui offrent des fonctionnalités variées : de la gestion des fichiers uploadés à la connexion aux bases de données, en passant par des frameworks, des systèmes de templates et la gestion de communication temps réel avec le visiteur. Pour cela on aura besoin d'un outil qui permet de télécharger facilement tous les modules de la communauté Node.js.

NPM (Node-Package-Manager) est le gestionnaire de paquets par défaut et le plus populaire dans l'écosystème NodeJS. Il est principalement utilisé pour installer et gérer des modules externes dans un projet Node.js. Il est couramment utilisé aussi pour installer une large gamme d'outils CLI et exécuter des scripts de projet. NPM suit les modules installés dans un projet avec le fichier package.json qui réside dans le répertoire d'un projet et contient :

- Tous les modules nécessaires à un projet et leurs versions installées
- Toutes les métadonnées d'un projet, telles que l'auteur, la licence, etc.
- Les scripts pouvant être exécutés pour automatiser des tâches dans le cadre du projet

A mesure que nous créons des projets Node.js plus complexes, la gestion de nos métadonnées et dépendances avec le fichier package.json nous fournissent des builds plus prévisibles, puisque toutes les dépendances externes restent les mêmes. Le fichier gardera automatiquement la trace de ces informations ; si nous pouvons modifier le fichier directement pour mettre à jour les métadonnées de notre projet, nous aurons rarement besoin d'interagir directement avec lui pour gérer les modules. [33]

III.3.1 Les frameworks de NodeJS

Un framework d'application Web est une combinaison de bibliothèques, d'aides et d'outils qui permettent de créer et d'exécuter des applications Web sans effort. Il offre les bases de la création d'un site ou d'une application Web. Les frameworks web supportent la personnalisation, la flexibilité, l'extensibilité, la sécurité, la compatibilité avec d'autres bibliothèques, etc.....Parmi les frameworks de NodeJS nous trouvons :

✓ **SailsJS**

Est un framework de développement web MVC en temps réel pour Node.js construit sur Express. Son architecture MVC ressemble à celle de frameworks comme Ruby on Rails. Cependant, il est différent en ce sens qu'il prend en charge un style plus moderne et axé sur les données du développement d'applications Web et d'API. Il prend en charge les API REST générées automatiquement, l'intégration facile à WebSocket et est compatible avec n'importe quel front-end : Angular, React, iOS, Android, Windows Phone, ainsi que du hardware personnalisé. Ce framework a des caractéristiques qui prennent en charge les exigences des applications modernes. Sails est particulièrement adapté au développement de fonctions en temps réel comme le chat. [64]



Figure II.4 : sailsJS. [71]

✓ **Mean.io**

MEAN (en entier MongoDB, Express, AngularJS et NodeJS) est une collection de technologies open source qui fournissent ensemble un cadre de bout en bout pour construire des applications web dynamiques à partir de zéro. Il vise à fournir un point de départ simple et agréable pour l'écriture d'applications JavaScript fullstack en natif cloud, en commençant par le haut vers le bas. Il s'agit d'un autre framework Node.js construit sur Express. [64]



Figure II.5 :MeanJS [72]

✓ ElectronJS

Est un framework open-source, utilisé pour créer des applications bureautiques utilisant HTML, CSS et JavaScript. À l'intérieur, cela fonctionne grâce à Chromium et Node.js. L'un des principaux avantages de l'utilisation d'Electron est que, étant basé sur les technologies Web, il est multiplateforme, ce qui permet de déployer des applications pour Linux, MacOS et Windows, avec le même code. Il comporte également des éléments natifs tels que des menus et des notifications, ainsi que des outils de développement utiles pour le débogage et les rapports de panne. [65]



Figure II.6 : electronJS. [70]

III.4 Communication temps réel avec Express et Socket.io

Le Web n'est plus une simple succession de pages liées entre elles via des liens hypertextes. De plus en plus, ce sont de véritables applications que l'on retrouve, proposant des degrés d'interactivité et de réactivité de plus en plus importants, et nécessitant une communication maximale entre serveur et client. HTTP est un mode de transport unidirectionnel : le client envoie une requête, le serveur répond puis la connexion est fermée. Ceci pose évidemment un problème si on veut continuellement pousser des données vers le client. Express est une infrastructure d'applications Web Node.js minimaliste et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications Web et mobiles, et Socket.io permet d'établir une communication bidirectionnelle entre client et serveur afin de s'affranchir des limitations des navigateurs. [34][35]

III .4.1 ExpressJS

Est un *framework JavaScript* minimaliste qui permet d'accélérer le développement d'un serveur NodeJS. En offrant de nombreuses fonctionnalités, il simplifie ainsi l'écriture du serveur. En particulier, il permet de gérer aisément le routage et offre de nombreuses fonctionnalités supplémentaires à l'objet HTTP utilisé avec NodeJS. [36]

Figure II.7: ExpressJS. [73]



III .4.2 Socket.io

Est une bibliothèque JavaScript utilisable côté serveur avec Node.js, ou côté client dans un navigateur. Elle permet d'abstraire la communication temps réel en utilisant un mécanisme de transport permettant des connexions longues. Si les *Websockets* sont disponibles dans le navigateur c'est elles qui seront choisies, sinon plusieurs autres techniques sont essayées (socket Flash, long polling AJAX, ...). L'API fournie est très simple à apprendre et à comprendre, et elle permet une grande souplesse dans le développement d'applications temps réel en JavaScript. [34]



Figure II.8 : Socket.io [74].

III.5 module Arduino

III.5.1 Définition

Le module Arduino est une plate-forme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte. Pour programmer cette carte, on utilise l'logiciel IDE-Arduino. [37]

III.5.2 Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons l'Arduino-UNO, l'Arduino-Mini, l'Arduino-Nano, l'Arduino-Diecimila, et l'Arduino-Mega2560. Parmi Ces types, nous avons choisi trois cartes Arduino-Mega, Arduino-Uno et Arduino-Nano. Elles contiennent beaucoup de dispositifs qui seront détaillés par la suite. [37]

➤ La carte Arduino-Mega2560 [84]

La carte Arduino-Mega2560 est basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que l'Arduino-Uno. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Elle peut se programmer avec le logiciel IDE-Arduino. Le contrôleur ATmega2560 contient un *bootloader* qui permet de modifier le programme sans passer par un programmeur. Le logiciel est téléchargeable gratuitement [38].

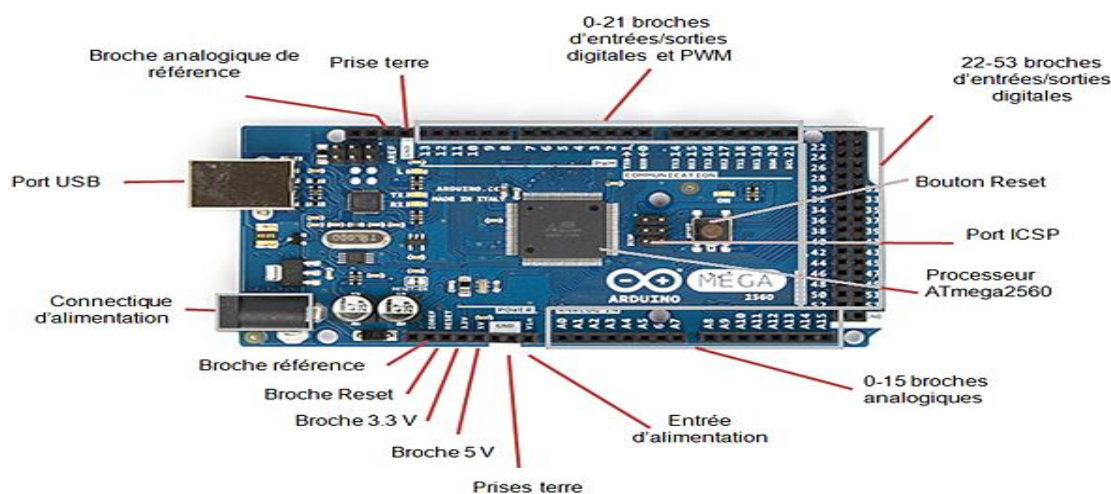


Figure III.9 : Description de la carte Arduino-Mega2560 [39]

Caractéristiques principales

- ✓ Version: Rev. 3
- ✓ Alimentation: 7 à 12 V sur connecteur alim.
- ✓ Microprocesseur: ATmega2560
- ✓ Mémoire flash: 256 kB

- ✓ Mémoire SRAM: 8 kB
- ✓ Mémoire EEPROM: 4 kB
- ✓ 54 broches d'E/S dont 14 PWM
- ✓ 16 entrées analogiques 10 bits
- ✓ Intensité par E/S: 40 mA
- ✓ Cadencement: 16 MHz
- ✓ 3 ports série
- ✓ Bus I2C et SPI
- ✓ Gestion des interruptions
- ✓ Fiche USB B
- ✓ Dimensions: 107 x 53 x 15 mm [38]

➤ **La carte Arduino-Nano [83]**

La carte Arduino-Nano est basée sur un ATmega328 cadencé à 16 MHz. Sa mémoire de 32 kB et son grand nombre d'E/S font de ce circuit compatible DIL30 un élément idéal pour les systèmes embarqués ou pour des applications robotiques nécessitant du multitâches. L'Arduino-Nano 3.0 peut se programmer avec le logiciel IDE-Arduino. Le contrôleur ATmega328 contient un *bootloader* qui permet de modifier le programme sans passer par un programmeur. Le logiciel est téléchargeable gratuitement [38].

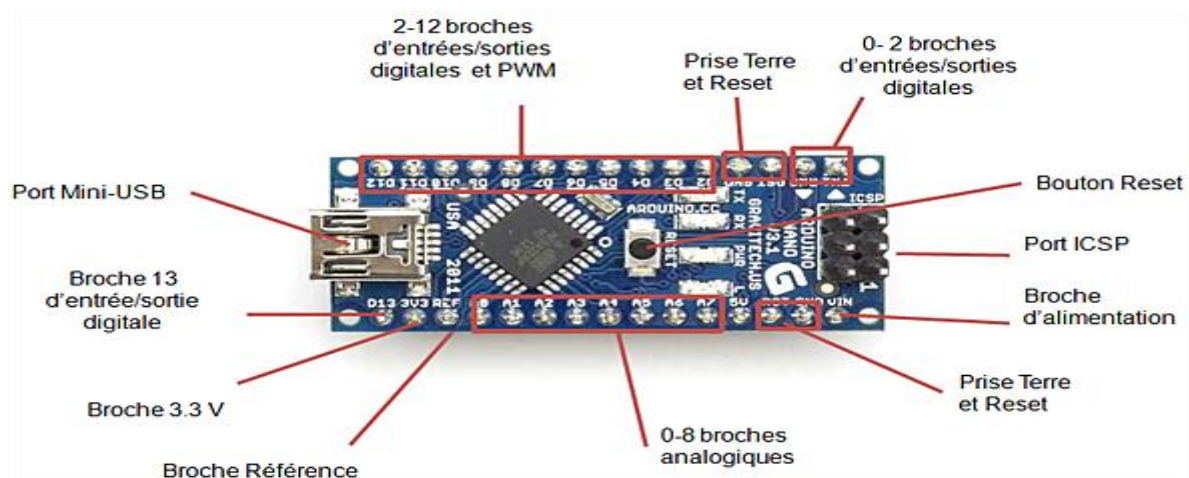


Figure III.10: Description de la carte Arduino-Nano [39]

Caractéristiques principales

- ✓ alimentation : via port USB ou 5 Vcc régulée sur broche 27 ou 6 à 20 V non régulée sur broche 30
- ✓ microprocesseur: ATmega328
- ✓ mémoire flash: 32 kB
- ✓ mémoire SRAM: 2 kB
- ✓ mémoire EEPROM: 1 kB
- ✓ 14 broches d'E/S dont 6 PWM
- ✓ 8 entrées analogiques 10 bits
- ✓ intensité par E/S: 40 mA
- ✓ cadencement: 16 MHz
- ✓ bus série, I2C et SPI
- ✓ gestion des interruptions
- ✓ fiche USB: mini-USB B
- ✓ boîtier DIL30
- ✓ dimensions: 45 x 18 x 18 mm
- ✓ Version d'origine fabriquée en Italie. [38]

➤ La carte Arduino-Uno [82]

La carte Arduino-Uno est basée sur un ATmega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Elle peut se programmer avec le logiciel IDE-Arduino. Le contrôleur ATmega328 contient un bootloader qui permet de modifier le programme sans passer par un programmeur. Le logiciel est téléchargeable gratuitement. [39].

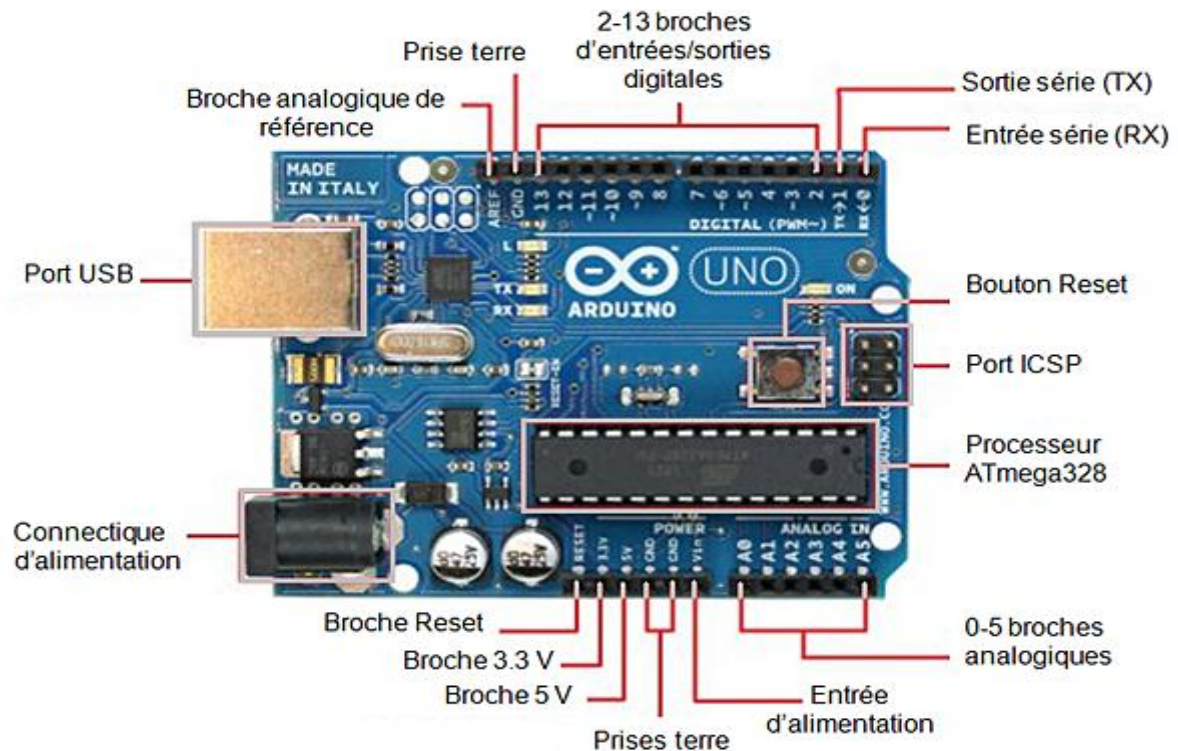


Figure III.11: Description de la carte Arduino-Uno [39]

Caractéristiques principales

- ✓ Version : Rev. 3
- ✓ alimentation : via port USB ou 7 à 12 V sur connecteur alim 5,5 x 2,1 mm
- ✓ microprocesseur: ATmega328
- ✓ mémoire flash: 32 kB
- ✓ mémoire SRAM: 2 kB
- ✓ mémoire EEPROM: 1 kB
- ✓ 14 broches d'E/S dont 6 PWM
- ✓ 6 entrées analogiques 10 bits
- ✓ intensité par E/S: 40 mA
- ✓ cadencement: 16 MHz
- ✓ bus série, I2C et SPI
- ✓ gestion des interruptions
- ✓ fiche USB B

- ✓ dimensions: 74 x 53 x 15 mm [38]

III.6 RASPBERRY PI 3 [85]

III.6.1 Histoire

Le Raspberry Pi (prononcé comme « *Raze* » « *Berry* » « *Paille* » en anglais) est un petit ordinateur de la taille d'une carte bancaire. Il a été conçu par une fondation éducative à but non-lucratif pour faire découvrir le monde de l'informatique sous un autre angle.

III.6.2 Définition

Le Raspberry Pi 3 est un nano-ordinateur mono-carte à processeur ARM conçu par le créateur de jeux-vidéos David Braben, dans le cadre de sa fondation Raspberry Pi 3. Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux-Debian et des logiciels compatibles. Mais également avec les OS Microsoft Windows : Windows 10 et Android. Il est fourni nu (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation du matériel de récupération. [40]

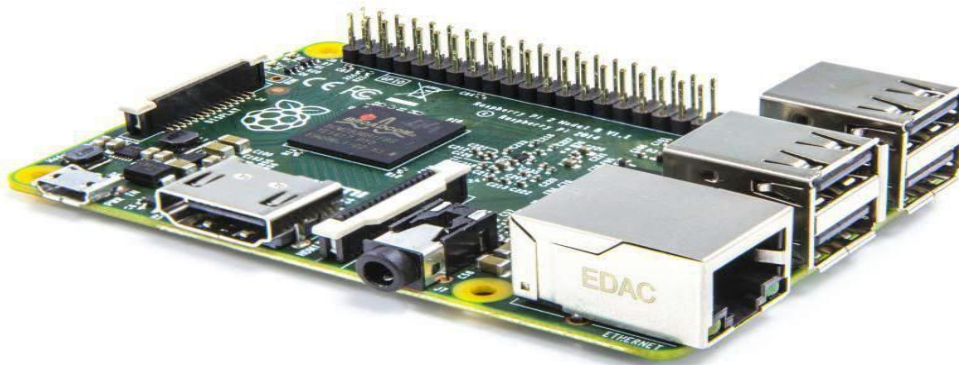


Figure III.12: Raspberry pi 3[40]

III.6.3 Equipement essentiel

Composé des périphériques indispensables pour l'installation d'un système d'exploitation.

- ✓ Clavier et souris en USB
- ✓ Carte SD (8GB minimum)
- ✓ Ecran avec port HDMI

- ✓ Câble HDMI
- ✓ La connexion à Internet
- ✓ Clavier et souris en USB
- ✓ Chargeur Micro-USB [40]

III.6.4 Les spécifications Entrées / Sorties

- **Processeur ARM** : Les architectures ARM sont des architectures de processeurs, A faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ».
- **Mémoire vive RAM** : C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement.
- **4 ports USB 2.0** : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques
- **1 port Ethernet** : C'est un port qui correspond au protocole international Ethernet du réseau local à commutation de paquets.
- **1 port HDMI / 1 Composite** : « *High Definition Multimedia Interface* » permet de relier le Raspberry PI à un dispositif compatible : écran LCD ou un vidéoprojecteur...
- **1 sortie audio jack 3,5mm** : C'est une connectique audio-vidéo.
- **Connecteur SD / SDHC** : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe.
- **GPIO**: « *General Purpose Input/Output* » sont des ports d'Entrée/Sortie 40 broches.
- **Bluetooth/Wifi 802.11 b/g/n** [40]

III.6.5 L'intérêt du Raspberry PI 3 :

- ✓ Elle permet d'utiliser les fonctions classiques d'un ordinateur comme le traitement du texte, mais aussi de lire des vidéos et de jouer... et plus encore.
- ✓ Outre l'aspect économique, c'est surtout le côté ludique de l'apprentissage de la programmation qui est mis en avant.
- ✓ Cela encourage ainsi à la récupération, à la « bibouille », correspondant ainsi à L'esprit de ce projet open source, le Raspberry Pi fonctionnant sous Linux [40].

III.6.6 les améliorations sur la carte Raspberry

➤ La Raspberry Pi 3 B+ [86]

Le modèle 3 B+, commercialisé deux ans après le Raspberry Pi 3, est une amélioration de ce dernier sur de nombreux points. Premièrement, le Raspberry Pi 3 B+ est équipé du SoC BCM2837B0 qui permet au CPU de passer d'une fréquence cadencée de 1.2 GHz à 1.4 GHz. Même si certaines personnes n'ont pas forcément besoin de ce gain de puissance, il peut se révéler très utile pour certains projets plus ambitieux. Au niveau de la connectivité, le Raspberry Pi 3 B+ propose une connexion Bluetooth Low Energy et une connexion Wi-Fi possible aussi bien sur la bande 2.4 GHz que sur la bande 5 GHz avec un débit bien meilleur que le Pi 3. Une des grandes améliorations du Raspberry Pi 3 B+ est effectivement ce connecteur pour l'extension Pi PoE HAT (Power over Ethernet) qui permet d'alimenter le Raspberry Pi avec une connexion Ethernet plutôt qu'une alimentation standard. [66]

Les caractéristiques :

- ❖ Un port Ethernet (Gigabit 300 Mbps)
- ❖ Un port HDMI
- ❖ Un connecteur micro-USB (pour l'alimentation)
- ❖ Un lecteur de carte SD
- ❖ 4 ports USB
- ❖ 1 connecteur pour la carte d'extension PoE HAT.

➤ La Raspberry Pi 3 A+ [87]

Plus récent que le Raspberry Pi 3 B+, le modèle 3 A+ se présente comme une version légèrement moins puissante mais aussi moins chère. Le Raspberry Pi 3 A+ intègre le même SoC Broadcom BCM2837B0 que le Pi 3 B+, à quatre cœurs cadencés à 1.4 Ghz (Cortex-A53 64 bit @ 1,4 GHz). La puissance est donc bien là. Mais l'A+ ne dispose toutefois que de 512 Mo de RAM et non de 1 Go comme le B+

Les caractéristiques

- ❖ 1.4GHz 64-bit quad-core ARM Cortex-A53;
- ❖ 512MB LPDDR2 SDRAM;

- ❖ WiFi 802.11 b/g/n/ac et Bluetooth 4.2 Low Energy;
- ❖ Boot USB renforcé;
- ❖ Gestion de la chaleur améliorée.

III.7 Comparaison des cartes

cartes caractéristiques	Raspberry pi3	Raspberry pi3 modèle b+	Raspberry pi3 modèle A+	Arduino uno	Arduino mega	Arduino nano
Fréquence processeur	1,2 GHz	1,4 GHz	1,4 GHz	16 Mhz	16 Mhz	16 MHz
Mémoire vive	1 GB	1 GB	512 MB	2 ko	8Ko	2Ko
Nombre de cœurs	4	4	4	1	1	1
Wi-Fi	802.11n	2.4GHz and 5GHz 802.11 b/g/n/ac	2.4GHz and 5GHz 802.11 b/g/n/ac	non	non	non
Core Type	Cortex-A53 64-bit	Cortex-A53 64-bit	Cortex-A53 64-bit	ATmega 328P	ATmega2560	ATmega328P

Figure III.13 : Tableau comparatif entre quelques cartes raspberry et arduino.

[91][92][93][94]

III.8 Conclusion

Dans ce chapitre, nous avons présenté les matériels et méthodes utilisés pour la conception de notre mémoire. Nous avons présenté dans un premier temps le Rôle de NodeJS dans les applications embarqués. Ensuite on a défini les Modules et Node-package-manager. Puis on a évoqué sur la Communication temps réel avec Express et Socket.io. A la fin on a défini les deux cartes arduino et raspberry .Le prochaine chapitre sera consacré à la partie réalisation et l'implémentation du système.

Chapitre

N°03

IV.1 Introduction

Le présent chapitre traite la phase de l'implémentation de notre application. Dans cette partie, nous allons développer une application IoT pour le suivi temps réel du processus de production notamment la température et l'humidité. Ce modèle intègre la contribution d'un objet intelligent (DHT22) au sein de l'Internet of Things. Nous débuterons, tout d'abord, par la description des technologies utilisées pour développer notre application. Ensuite, nous mentionnerons les outils technologiques utilisés. Finalement, nous donnerons un aperçu sur le travail réalisé en présentant les différentes interfaces de notre application web.

IV.2 Outils utilisés

➤ Sublime texte

Est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités. Depuis la version 2.0, sortie le 26 juin 2012, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares [41].



Figure IV.1 : Sublime Texte [42]

➤ L'IDE-Arduino

Le logiciel de programmation des cartes Arduino présente une application java, libre et multi plates-formes, servant d'éditeur de code et de compilateur qui peut transférer le programme (croquis ou sketch) à travers la liaison série asynchrone (RS-232, Bluetooth ou USB selon la carte). Il est également possible de se passer de l'interface Arduino, et de

compiler et uploader les programmes via l'interface en ligne de commande. Le langage de programmation Arduino est basé sur les langages c/c++ adaptés à la carte et ses entrées/sorties. Une fois chargé dans le microcontrôleur, le programme s'exécute. [43]

Download the Arduino IDE



Figure IV.2 :L'IDE arduino [44]

- **language C**

En particulier, utiliser le langage C pour programmer l'Arduino permet généralement de créer des programmes plus petits et davantage optimisés, avec un contrôle plus fin des tâches effectuées. Le C est adopté dans le monde entier pour la programmation de microprocesseurs, car il offre un bon compromis entre l'effort du développement et l'efficacité du programme, mais aussi, en raison de son histoire, il existe des bibliothèques optimisées, une documentation très complète et des manières de résoudre les problèmes.[45]

- **format JSON**

Ce format, provenant du monde JavaScript et représentant un objet, s'apparente au XML. Ses principales fonctions est de permettre le stockage des données textuelles, mais pas des données binaires (du moins pas sans les encoder, en base64 par exemple) de manière structurée. Comme ce format est très ouvert, il est pris en charge par de nombreux langages : JavaScript, PHP, Perl, Python, Ruby, Java,... Sa structure en arborescence et sa syntaxe simple lui permet de rester très "léger" et efficace. [46]

- **les bibliothèques**

Les bibliothèques Arduino sont un moyen pratique de partager du code tel que des pilotes de périphérique ou des fonctions utilitaires couramment utilisées. Le plus récent Arduino IDE dispose d'un gestionnaire de bibliothèque qui facilite l'installation de bibliothèques tierces soumises à Arduino pour utilisation. Pour le développement de notre

application on a utilisé la bibliothèque Adafruit qui permet le fonctionnement du capteur DHT22 [47]. Par exemple, Adafruit Unified Sensor Driver et Adafruit DHT Unified.

➤ NPM

NPM est l'abréviation de Node-Package-Manager, qui est un outil (programme) gérant les bibliothèques de programmation JavaScript pour NodeJS. Cet outil est réellement nécessaire pour le monde open source. Dans la communauté JavaScript, les développeurs partagent des composants de base, des bibliothèques de programmation ou même des Framework. Chacun de ces extraits de code peut dépendre de nombreux autres codes source ouverts. Heureusement, les outils de gestion de bibliothèque sont nés. Sinon, il faudra beaucoup d'efforts pour gérer ces bibliothèques. [48]



Figure III.3 : Node Package Manager [49]

➤ Git Hub

Est une société à but lucratif qui offre un service d'hébergement de référentiel il est basé sur le cloud. Essentiellement, il est beaucoup plus facile pour les individus et les équipes de l'utiliser pour le contrôle de version et la collaboration. En effet, l'interface de GitHub est suffisamment conviviale pour que même les codeurs débutants puissent profiter de ces services. Par ailleurs, Git est un système de contrôle de version distribué, ce qui signifie que l'ensemble de la base du code et de l'historique est disponible sur l'ordinateur de chaque développeur, cela permet des branchements et une fusion faciles. Il permet aussi de gérer d'autres types de projets livres. De plus, n'importe qui peut s'inscrire et héberger gratuitement un dépôt de code public, ce qui rend GitHub particulièrement populaire auprès des projets open-source. [50]



Figure III.4 : Git Hub [51]

➤ **TeamViewer**

TeamViewer est une solution complète de téléassistance, de contrôle et d'accès à distance compatible avec presque toutes les plateformes mobiles et de bureau, y compris Windows, MacOS, Android et iOS. TeamViewer permet de connecter à distance des ordinateurs ou appareils mobiles où qu'ils soient dans le monde et de les utiliser comme si on y est. De plus, connecter à distance aux serveurs, machines à usage commercial et appareils IDO de n'importe où et à n'importe quelle heure grâce à notre réseau mondial d'accès à distance [52].

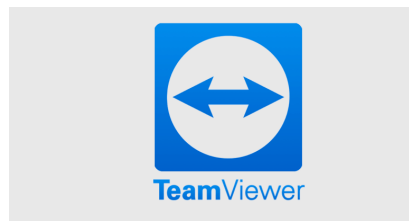


Figure IV.5 : TeamViewer [53]

IV.3 Les technologies utilisées

➤ **Système de gestion de base de données PostgreSQL**

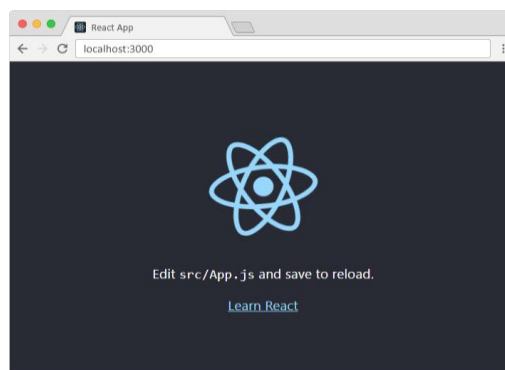
PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent avec d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises. PostgreSQL peut stocker plus de types de données que les types traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type etc. [41].



Figure IV.6 : PostgreSQL [53]

➤ ReactJS

Est une technologie Javascript, mais il ne s'agit pas d'un framework à proprement parler. En fait, il s'agit plus d'une librairie open source qui permet de construire des interfaces utilisateur dynamique. Et, de plus, cette librairie est maintenue un grand nombre de développeurs indépendants. React est basé sur Javascript. C'est par ailleurs une technologie très appréciée par développeurs web ces derniers temps. Elle permet de réaliser des applications web cross plateforme et ultra-performantes. Ce sont les équipes de Facebook qui sont à l'origine de cette plateforme. En effet, c'est en 2013 qu'est né React. Sa syntaxe, est basée sur la norme ECMAScript 6, elle permet de créer des composants JS pour la structuration de l'application. Puis, ainsi, concevoir la manière de développer une application. React s'est distingué par la simplicité de son starter kit et la souplesse de son architecture logicielle. Cette dernière, est basée sur des modules. En effet, React est aussi présent sur les principales plateformes de gestionnaires de paquets, comme Npm ou Yarn. En revanche, react est orienté application web. Pour cela, la création d'une interface connectée à n'importe quelle API de services REST est possible. Ce qui veut dire, à n'importe quelle technologie backend comme Node, Java, Symfony, ou encore WordPress. [55]



FigureIV.7 : ReactJS [56]

➤ NodeJS

NodeJS est un environnement open-source permettant de développer des applications multiplateformes. Basé sur le V8 de Google, le moteur d'exécution JavaScript utilisé dans Chrome, NodeJS traite donc les données en JavaScript. L'écriture se fait en JavaScript non plus uniquement côté client, mais également côté serveur. NodeJS vient ainsi concurrencer les autres langages serveurs que sont par exemple PHP ou Java. Si NodeJS peut effectivement être utilisé comme serveur sans avoir besoin d'une solution externe tels que Apache ou lighttpd, il reste avant tout une plateforme de développement JavaScript permettant de répondre à de nombreux besoins clients/serveurs grâce à la mise à disposition d'une API. [57]



Figure IV.8: NodeJS [58]

➤ Express.js

Il s'agit en fait d'un micro-Framework pour NodeJS. Il fournit des outils de base et robuste pour aller plus vite dans la création d'applications NodeJS. [59]. Express apporte une couche fine de fonctionnalités d'application Web fondamentales, sans masquer les fonctionnalités de Node.js qu'on connaît et apprécie. [60]



Figure IV.9: ExpressJS [61]

IV.4 L'architecture globale du système

La figure III.9 illustre l'architecture globale du system .En effet ,l'Arduino capte la donnée (température et humidité) grâce au capteur DHT22 puis ces données sont envoyées au serveur NodeJS sous format JSON en utilisant la communication série .Par la suite, le serveur envoie au même temps ces données à la base de données afin d'être sauvegardées pour une utilisation ultérieure notamment l'affichage de l'historique, et au client ReactJS afin d'afficher l'évaluation de la température et l'humidité en temps réel sous forme de graphs dans un navigateur web.

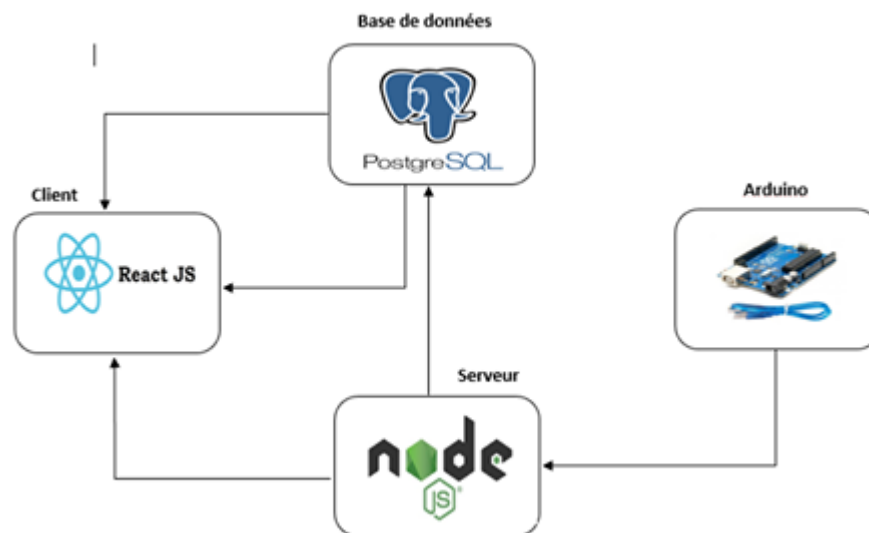


Figure IV.10: architecture globale du système.

IV.4.1 Câblage du système

Dans notre système nous avons utilisé une carte Arduino Uno, un capteur de température et d'humidité DHT22 et des fils, en voici un aperçu :

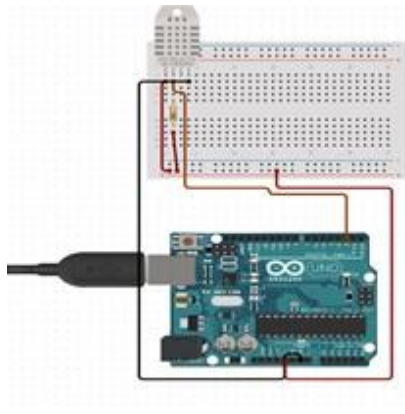


Figure IV.11 : câblage du système.

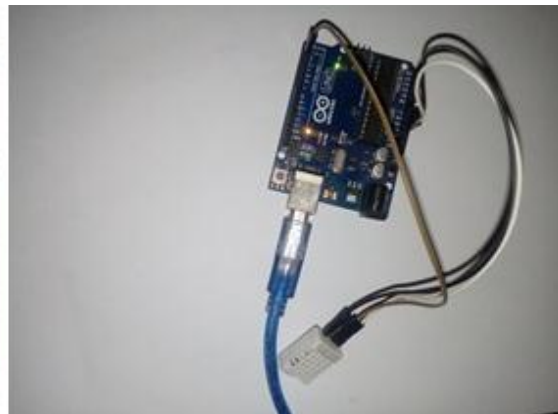


Figure IV.12 : Montage du capteur DHT22

IV.4.2 implémentation du système

➤ Branchement du capteur DHT22

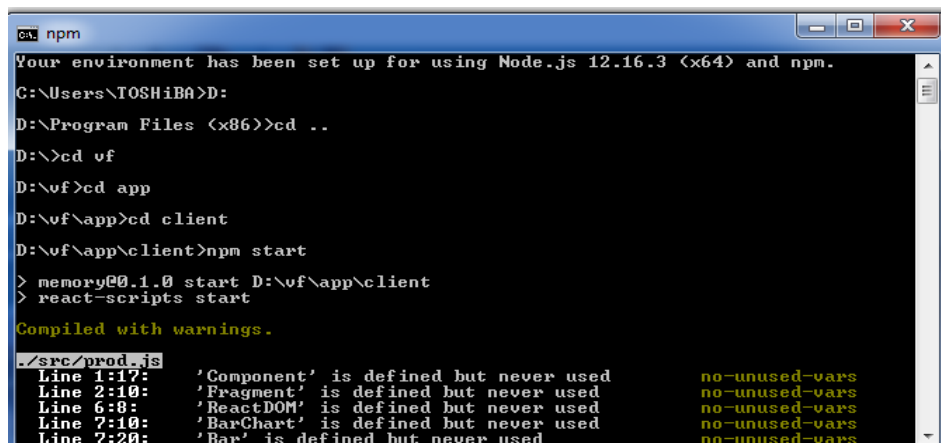
Le branchement du capteur DHT22 à la carte arduino UNO est simple. Dans notre cas, nous disposons de trois fils :

- Le fil blanc est le fil d'alimentation : branché à la borne 5V.
- Le fil noir est le fil de mise à la terre : branché à la borne GND.
- Le fil marron est le fil de transfert de données : branché à la borne qui sera réservée à cette fonction. Dans notre cas nous avons choisi arbitrairement le pin digital 2 (Figure III.17).

Pour lancer le projet il faut ouvrir l'invite de commande de NodeJS et saisir les deux commandes nécessaires pour lancer le serveur et le client comme suit :

➤ Lancement du client Reactjs :

La commande `npm start` permet de lancer le client, après un certains moment un nouveau onglet s'ouvrira dans le navigateur web par défaut (Google Chrome pour notre cas).



```

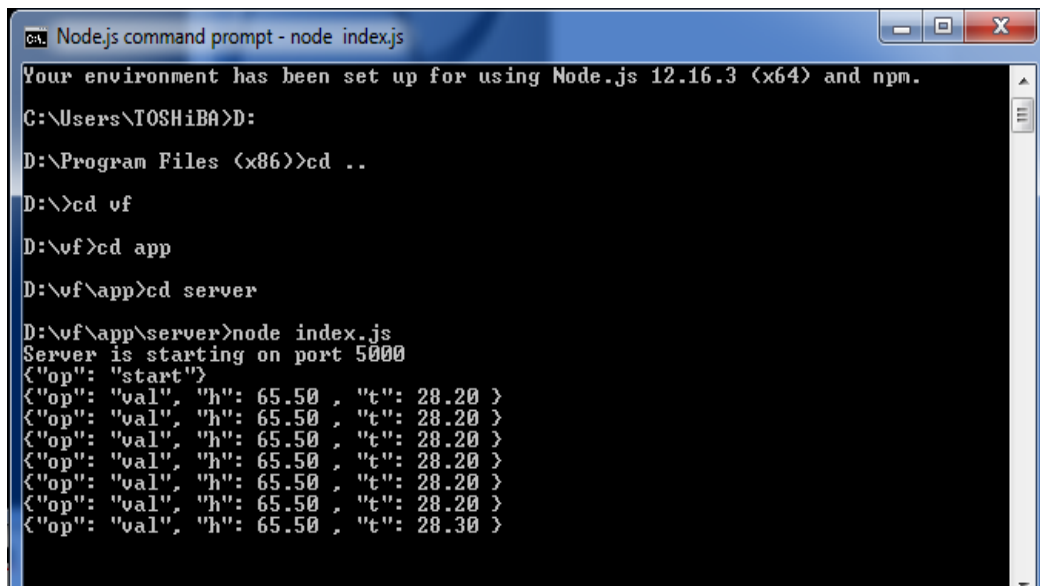
ca. npm
Your environment has been set up for using Node.js 12.16.3 (x64) and npm.
C:\Users\TOSHIBA>D:
D:\Program Files (x86)>cd ..
D:\>cd vf
D:\vf>cd app
D:\vf\app>cd client
D:\vf\app\client>npm start
> memory@0.1.0 start D:\vf\app\client
> react-scripts start
Compiled with warnings.
./src/prod.js
Line 1:17:  'Component' is defined but never used      no-unused-vars
Line 2:10:  'Fragment' is defined but never used          no-unused-vars
Line 6:8:   'ReactDOM' is defined but never used           no-unused-vars
Line 7:10:  'BarChart' is defined but never used            no-unused-vars
Line 7:20:  'Bar' is defined but never used                 no-unused-vars

```

Figure IV.13 : client Reactjs.

➤ Lancement du serveur NodeJS

Avec la commande `node index.js` notre serveur démarrera tout en affichant le port d'exécution (Server is starting on port 5000) et les données captées (température et humidité). Au même temps le serveur NodeJS est à l'attente d'une connexion client pour pouvoir lui transmettre les données en temps réel. Ces données ne seront pas perdues car elles seront envoyées vers une BDD PostgreSQL.



```

Node.js command prompt - node index.js
Your environment has been set up for using Node.js 12.16.3 (x64) and npm.
C:\Users\TOSHIBA>D:
D:\Program Files (x86)>cd ..
D:\>cd vf
D:\vf>cd app
D:\vf\app>cd server
D:\vf\app\server>node index.js
Server is starting on port 5000
{"op": "start"}
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.20 }
{"op": "val", "h": 65.50, "t": 28.30 }

```

Figure IV.14 : serveur NodeJS.

➤ Enregistrement sur la base de données

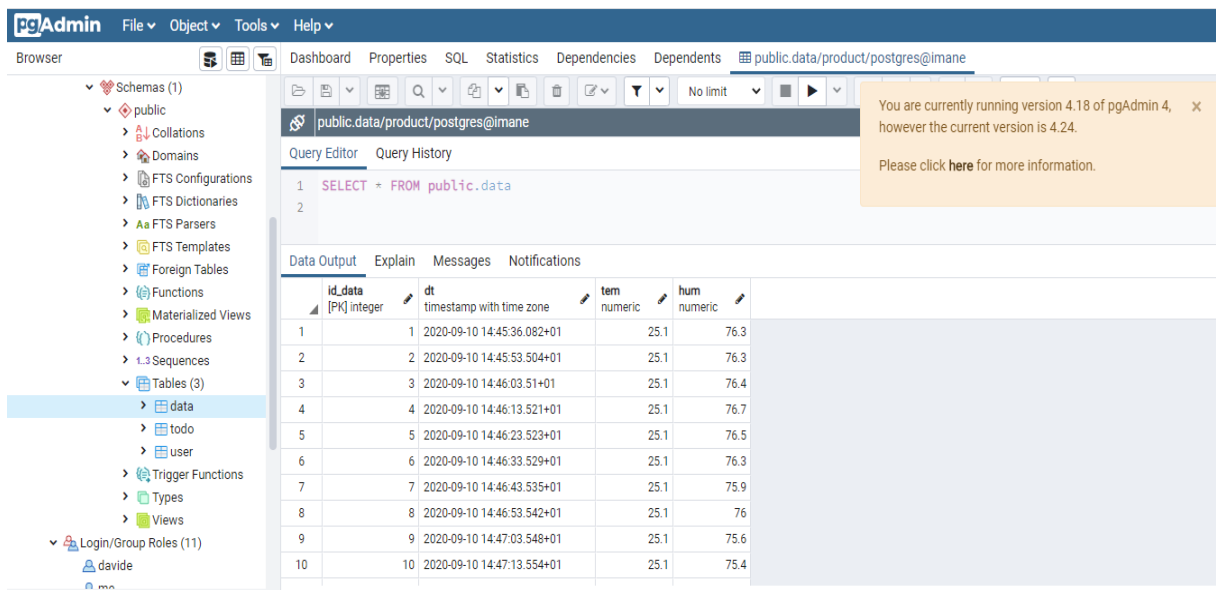
Les données sont envoyées à la BDD à l'aide de la fonction « insertValues » affiché sur la figure III.15. Afin d'être enregistrées dans la table data, la température se trouve dans le champ tem et l'humidité dans le champ hum et le champ dt pour référencer la date et l'heure du captage, ces données sont affichées sur la figure III.16.

```

60
61 //insertion des donnée capté dans la bdd
62 async function insertValues(t, h) {
63   const currentDateTime = new Date();
64   try {
65     await pool.query("INSERT INTO data(dt, tem, hum) VALUES ($1, $2, $3)", [currentDateTime, t, h]);
66   } catch (err) {
67     console.error("Erreur lors de l'insertion des données ::: " + err.message);
68   }
69 }

```

Figure IV.15: La fonction d'insertion sur la BDD.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Schemas (1)' tree is expanded to 'public', and the 'Tables (3)' list includes 'data', 'todo', and 'user'. The 'data' table is selected, and its contents are displayed in the 'Data Output' tab. The table has 10 rows of data. A notification box on the right indicates that the current version of pgAdmin 4 is 4.18, while the latest version is 4.24.

	id_data [PK] integer	dt timestamp with time zone	tem numeric	hum numeric
1	1	2020-09-10 14:45:36.082+01	25.1	76.3
2	2	2020-09-10 14:45:53.504+01	25.1	76.3
3	3	2020-09-10 14:46:03.51+01	25.1	76.4
4	4	2020-09-10 14:46:13.521+01	25.1	76.7
5	5	2020-09-10 14:46:23.529+01	25.1	76.5
6	6	2020-09-10 14:46:33.529+01	25.1	76.3
7	7	2020-09-10 14:46:43.535+01	25.1	75.9
8	8	2020-09-10 14:46:53.542+01	25.1	76
9	9	2020-09-10 14:47:03.548+01	25.1	75.6
10	10	2020-09-10 14:47:13.554+01	25.1	75.4

Figure IV.16: La table data

➤ Lancement de l'application sur http://localhost:3000/

Après avoir lancé le client, notre application web se lance sur le port 3000. L'interface nous permet de s'authentifier afin de permettre l'accès aux données captées par l'Arduino UNO à l'aide d'un capteur DHT-22.

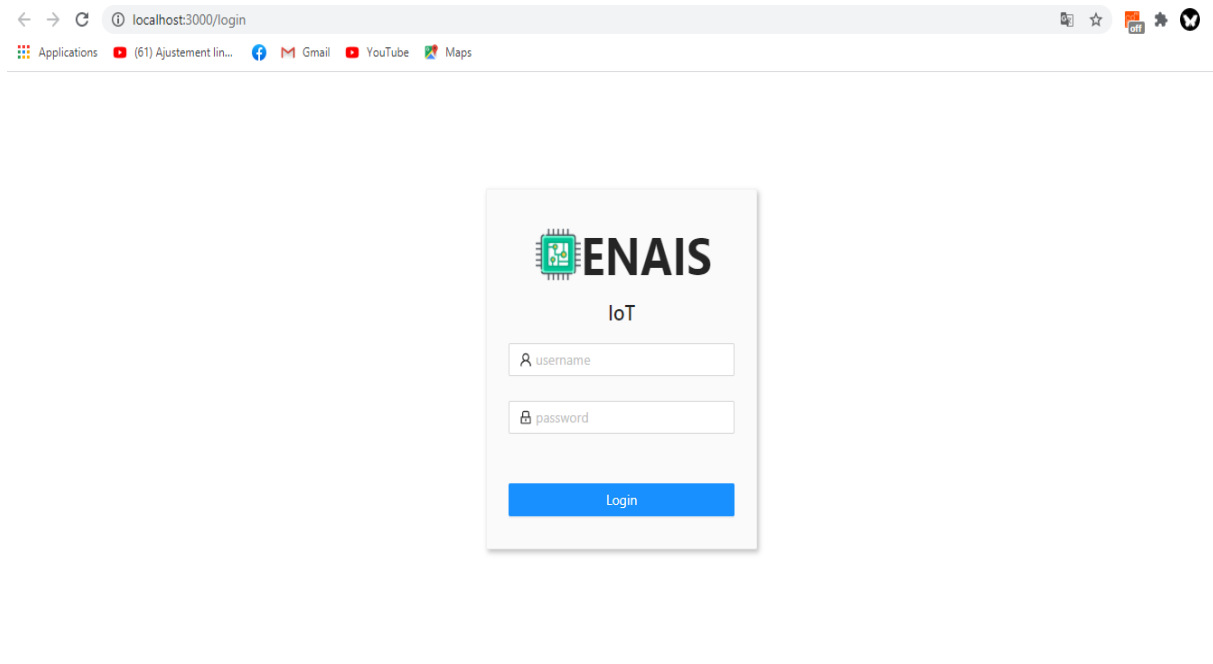


Figure IV.17:interface d'authentification.

IV.4.3 Présentation des interfaces :

L'interface d'accueil présente quatre espaces : un espace de « Real time », «historical», « Dashboard » et « Sittings », si l'utilisateur a déjà tapé le bon nom d'utilisateur et le bon mot de passe dans l'espace d'authentification présenté dans la figure III.14.

➤ L'interface « Real time » :

Sur cette interface l'utilisateur visualise le changement de la température et l'humidité en temps réel sous forme de graphes comme le montre la figure III.15. On pourra même voir à quelle heure cette donnée était captée.

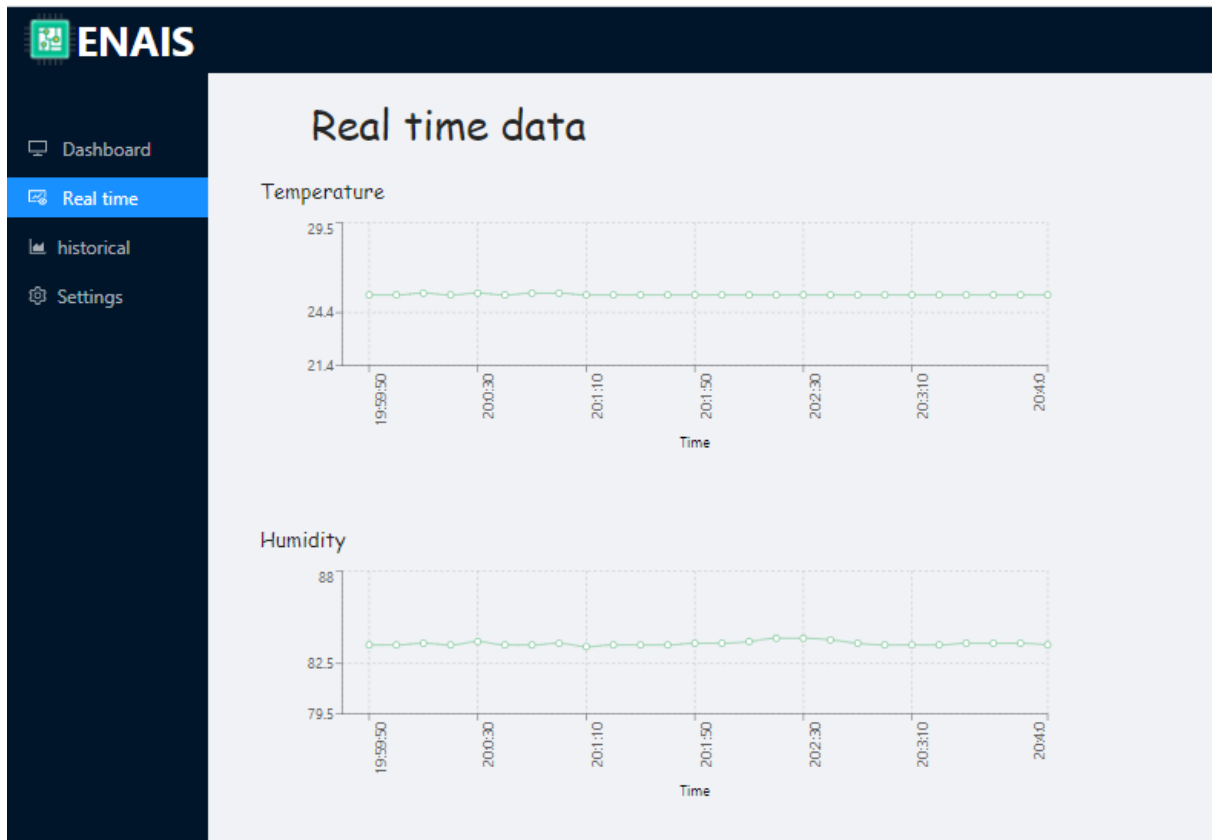


Figure IV.18:données de capteur DHT-22.

➤ **L'interface « historical » :**

Au niveau de cette interface l'historique des données (température et humidité) est affiché selon le choix effectué par l'utilisateur, ce dernier pourra sélectionner 1/7/30 jours en appuyant sur l'un des boutons présenté en haut de la page, un graphe est affiché présentant l'historique comme le montre la Figure III.16.

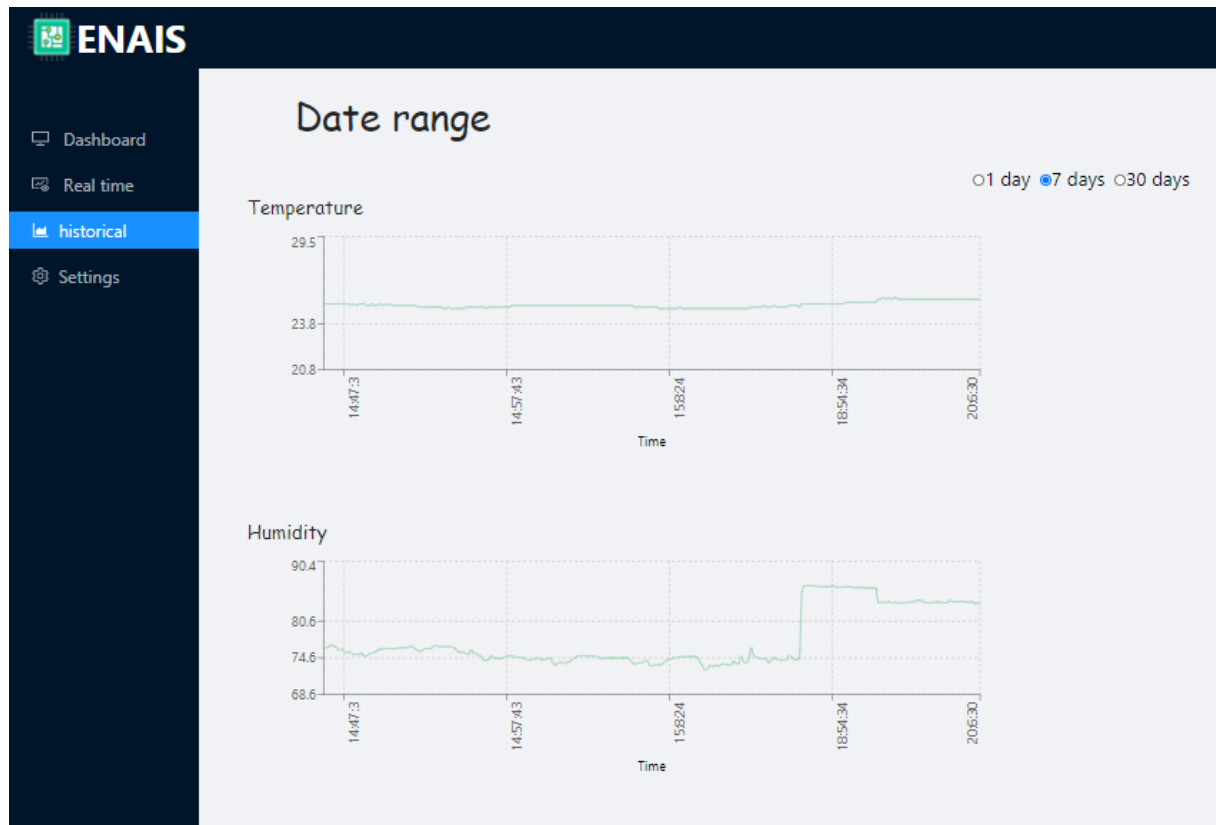


Figure IV.19:Les données récupérées de la BDD.

IV.5 Conclusion

Dans ce chapitre, nous avons présenté les détails d'implémentation et réalisation de notre objet connecté, ce qui nous a permis d'explorer les nouvelles technologies dans le domaine de l'Internet of Things, dont NODEJS ,REACTJS ,capteur DHT22 ,la carte ARDUINO et sa capacité de convertir tout objet à base d'un système électronique en un objet connecté, d'où nous avons pu récupérer des données émises par un capteur météorologique. Cette technologie est capable même d'embarquer un serveur Web à part entière, ce qui ouvre le champ pour le développement d'applications embarquées de communication et d'échange de données; Tout au long de ce chapitre, nous avons abordé les technologies utilisées, notre environnement de travail. Par la suite, nous avons expliqué notre architecture globale du système afin de présenter finalement les différentes principales parties d'implémentation de notre application.

Conclusion
conclusion

Générale
générale

Conclusion générale

L'émergence de l'Internet of Things (IoT) offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel. Par ailleurs, le problème abordé dans ce mémoire est comment exploiter ces services qui peuvent être offerts par différents appareils et différents objets, afin d'établir des systèmes intelligents pour répondre aux exigences et aux besoins des utilisateurs. Pour ce faire, nous avons mis en exergue une application web pour le suivi temps réel du processus de production notamment l'humidité et la température. En effet, ce modèle est destiné à favoriser les interactions automatiques entre des objets notamment la carte arduino uno et le capteur DHT22 destiné pour acquérir des données (température et humidité) par la mise en place d'une stratégie permettant la visualisation sur une interface web le changement de la température et l'humidité au sein d'une entreprise afin de prévenir les risques et d'optimiser la rentabilité de celle-ci. En revanche, pour développer cette application, plusieurs étapes progressives ont été entamées, puis finalisées par l'application globale, ou dans chaque étape, deux parties pour la conception sont faites ; une partie software : qui consistait en la programmation en langage C dans l'environnement Arduino IDE, Node.js et postgresql pour le frontend de notre application et ReactJS pour le backend ; et une partie hardware : qui consistait à une carte arduino uno et un capteur DHT22 qui est relié à cette dernière via des fils mâle-femelle. Et en fin à la réalisation pratique grâce à une interface web. Les résultats achevés montrent l'efficacité du système dans le captage et la sauvegarde en temps réel des données captées sur une base de données relationnelle et la visualisation sous forme de graph le changement de ces données. Ainsi, il reste encore des pistes de recherche à explorer pour peaufiner et optimiser notre modèle.

Perspectives

Comme nous pouvons le constater, notre projet pour l'avenir vise une autonomie complète des objets dans l'entreprise, dans le sens où ils peuvent collaborer sans intervention humaine, en restant fidèle à notre modèle actuel et en explorant d'autres pistes de recherche et de pratique sur les différents processus et concepts proposés. Malgré les réalisations que nous avons réussi à mener à bien, et la mise en place de notre application web pour le processus de production. Nous pensons qu'il reste encore du chemin à faire pour améliorer et explorer de nouvelles pistes, d'abord, l'approche de la géolocalisation qui permet de visualiser en temps réel le déplacement d'un véhicule en temps réel qui nécessite le développement et l'implémentation d'une application serveur/client capable de reformuler les besoins d'un objet en une requête envoyée dans le réseau en utilisant la technologie GPS. Ensuite, l'approche de surveillance et du contrôle d'accès dépend d'une répartition manuelle des objets selon une analyse visuelle de leur contribution ; nous prévoyons développer un outil automatique avec plus de critères capables de détecter si une personne peut accéder à l'entreprise ou pas grâce à un capteur d'empreinte digital et des caméras de surveillance. De plus, la sécurité est essentielle pour chaque entreprise. Des systèmes IoT sécurisés, résilients et adaptatifs nécessitent de bons partenariats, la protection des informations commerciales précieuses est bien nécessaire. En effet, il faut prendre en compte l'ensemble de ces éléments, ainsi que les usages qui induisent un niveau de risque, car la remontée ou la baisse de la température ou l'humidité dans une salle 6 dans l'entreprise présente des risques, pour cela l'adoption ou le développement d'un système de sécurité qui prévoit les risques auxquels tout système dans une entreprise doit se confronter comme la modification des données, l'usurpation d'identité et la non-intégrité des données. Nous pensons appliquer cette sécurité à l'aide d'une bibliothèque micro ECC installée sur l'IDE d'Arduino qui permet de chiffrer les données juste après leurs captages. Finalement, la version actuelle du modèle proposé se porte que sur le processus de production. Il serait intéressant de travailler sur un cas plus général notamment pour les processus de géolocalisation et de la surveillance pour un système IoT plus complet de bout en bout.

Webliographie

- [2] <https://82252856.netacad.com/courses> , 26/01/2020,
- [4]<https://mbamci.com/comprendre-linternet-des-objets-et-ses-differentes-applications/>, 26/01/2020.
- [7] <https://www.journaldulapin.com/tag/dal/> , 26/01/2020.
- [8] <https://savemoney.es/fr/asin/lemfo> , 26/01/2020.
- [9]<https://abra-electronics.com/wireless/gps/modules/mod-120-neo-6m-ublox-gps-module.html?sl=fr>, 26/01/2020.
- [10] <https://fr.slideshare.net/mo9rissat/identification-des-empreintes-digitales> , 29/01/2020.
- [11] http://www.abiova.fr/biometrie_empreinte_solo.asp ,consulté le 29/01/2020.
- [12] <https://www.robotshop.com/eu/fr/module-camera-ov7670-.html> 22/05/2020
- [13] <http://gilles.thebault.free.fr/spip.php?article47> ,22/05/2020
- [14]https://boutique.semageek.com/fr/1105-bmp280-capteur-de-pression-barometrique-temperature-altitude.html?search_query=BMP+280+pression&results=86 , 22/05/2020
- [15]https://www.robot-maker.com/shop/blog/27_Connecter-capteur-Infrarouge-HCSR501.html , 23/05/2020
- [18] <https://www.intesens.com/les-benefices-de-liot/> 26/01/2020
- [19] <https://www.cgi.com/fr/blogue/cybersecurite/les-menaces-liees-a-l-internet-des-objets> , 26/01/2020.
- [20]<https://openclassrooms.com/fr/courses/5079046-mettez-en-place-une-architecture-pour-objets-connectes-avec-le-standard-onem2m/5079156-retrouvez-vous-dans-la-jungle-des-standards> , 26/01/2020.
- [21] <https://www.commentcamarche.net/contents/520-le-protocole-http> , 26/01/2020.
- [22]<https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about-fr> , 26/01/2020
- [23] <https://francais.medscape.com/features/diaporama/33000077>, 26/01/2020
- [24] <https://www.journaldulapin.com/tag/dal/> , 26/01/2020.
- [25] <https://www.yara.fr/fertilisation/blog/agriculture-connectee/> ,26/01/2026.

- [26]https://fr.123rf.com/photo_88025683_concept-de-l-agriculture-intelligente-agronome-ou-agriculteur-utilisent-l-intelligence-artific.html?fromid=djk2MHI4cGNxOTV1ZmQ4WC9jMU1ZZz09 26/02/2020
- [27]<https://www.universalis.fr/encyclopedie/drones/3-les-systemes-de-drones-militaires/>, 26/01/2020
- [28] <https://www.middleeasteye.net/fr/en-bref/la-chine-> ,26/01/2020
- [30]<https://simplement-entrepreneurs.fr/10-problemes-rencontres-directeurs-rh/>,26/01/2020.
- [31] https://fr.m.wikipedia.org/wiki/Gestion_de_la_production , 26/01/2020.
- [32]-https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarquesintroduction.htm?fbclid=IwAR2p8JSqQe6N9nK6rQxuVf0Oy2hnVinlg8heIg08Wqc5ivGe_xV739DVJAA ,12/09/2020..
- [33]-https://www.digitalocean.com/community/tutorials/how-to-use-node-js-modules-with-npm-and-package-json-fr?fbclid=IwAR2p8JSqQe6N9nK6rQxuVf0Oy2hnVinlg8heIg08Wqc5ivGe_xV739DVJAA ,12/09/2020.
- [34]https://www.synbioz.com/blog/tech/le-web-temps-reel-avec-socketio?fbclid=N9nK6rQxuVf0Oy2hnVinlg8heIg08Wqc5ivGe_xV739DVJAA ,12/09/2020.
- [35]https://expressjs.com/fr/?fbclid=IwAR2p8JSqQe6N9nK6rQxuVf0Oy2hnVinlg8heIg08Wqc5ivGe_xV739DVJAA ,12/09/2020.
- [36]http://www.cril.univ-artois.fr/~boussemart/express/chapter01.html?fbclid=IwAR2p8JSqQe6N9nK6rQxuVf0Oy2hnVinlg8heIg08Wqc5ivGe_xV739DVJAA ,12/09/2020.
- [38] <https://arduino.cc> 20/05/2020
- [39] <https://www.editions-eni.fr/> 21/05/2020
- [42] <https://www.macg.co/logiciels/2017/09/editeur-de-code-sublime-text-3-est-finalise-99715> , 30/08/2020
- [44]-<https://www.locoduino.org/spip.php?article15> 30/08/2020.
- [45]-<https://francesco-balducci.developpez.com/tutoriels/programmer-arduino-en-c/> 30/08/2020.
- [46]-<https://www.alsacreations.com/article/lire/1675-json-stockage-leger-pratique->

donnees-multitypes.html , 30/08/2020.

[47]<https://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use/arduino-libraries> , 30/08/2020.

[48] <https://o7planning.org/fr/11925/qu-est-ce-que-npm> , 30/08/2020.

[49]<https://blog.lesieur.name/installer-tous-les-modules-npm-node-js-sous-windows/> 30/08/2020.

[50]<https://kinsta.com/fr/base-de-connaissances/base-de-connaissances-github/> 02/10/2020.

[51]<https://dev.to/ravirajthedeveloper/what-is-git-and-github-and-how-to-use-github-2mb1> , 30/08/2020.

[52] <https://www.teamviewer.com/fr/produits/teamviewer/> 30/08/2020.

[54]<https://www.postgresql.org/> 30/08/2020.

[55]<https://unscuzzy.com/react-js-cest-quoi/> 02/10/2020.

[56]<https://www.w3schools.com/react/> 30/08/2020.

[57]<https://www.alioze.com/developpement-nodejs-2017> 30/08/2020.

[58]https://hub.docker.com/_/node 30/08/2020.

[59]<https://openclassrooms.com/fr/courses/4664381-realisez-une-application-web-avec-react-js/4664388-decouvrez-lutilite-et-les-concepts-cles-de-react> 30/08/2020.

[60]<https://expressjs.com/fr/> 30/08/2020.

[61]<https://www.fiverr.com/greatwits/develop-backend-in-nodejs-and-expressjs> 30/08/2020.

[62]<https://openclassrooms.com/fr/courses/1056721-des-applications-ultra-rapides-avec-node-js/1056866-node-js-mais-a-quoi-ca-sert> 21/09/2020.

[63]<https://medium.com/@nizar.oukhchi/le-non-bloquant-avec-node-js-fac11fe9ae51> 21/09/2020.

[64]<https://www.hello-pomelo.com/les-meilleurs-frameworks-nodejs-en-2019/> 21/09/2020.

[65]<https://riptutorial.com/fr/electron> 21/09/2020.

[66]<https://www.raspberrypi-france.fr/guide/choisir-modele-raspberry-pi/> 21/09/2020.

- [67]<http://blogpeda.ac-poitiers.fr/lp2i-si/2013/11/07/arduino/> 01/10/2020.
- [68]<https://www.advenica.com/en/blog/2020-03-19/security-in-a-future-of-iot-based-on-5g>
01/10/2020.
- [69]<https://www.lebigdata.fr/services-web-definition> , 01/10/2020.
- [70]https://www.pinclipart.com/pindetail/oJwwRT_debugger-for-electron-next-js-examples-clipart/ 01/10/2020.
- [71]<https://blog.bef-technology.com/quest-ce-que-sailsjs/> 01/10/2020.
- [72]<http://meanjs.org/> 01/10/2020.
- [73]<https://medium.com/javascript-in-plain-english/3-express-js-features-you-need-to-know-8f78b0035f33> 01/10/2020.
- [74]<https://www.agendadulibre.org/events/9893> 01/10/2020.
- [75]- <https://www.robot-maker.com/shop/capteurs/31-dht22.html>. 04/10/2020
- [76] <https://www.sparkfun.com/products/10167> ,06/10/2020
- [77] <https://atisys-domotique.fr/capteurs/174-capteur-mouvement-detecteur-pir-infrarouge-hc-sr501.html> , 06/10/2020
- [78]<https://www.digikey.fr/product-detail/fr/bosch-sensortec/BMP085/828-1005-1-ND/1987010> , 06/10/2020
- [79]<https://www.alldatasheet.com/datasheet-pdf/pdf/1132068/BOSCH/BMP180.html> ,
06/10/2020
- [80]<https://www.mouser.fr/ProductDetail/Bosch-Sensortec/BMP183?qs=d72FGnIDsgRh64LIaYgRCw==>, 06/10/2020
- [81]<https://www.mouser.fr/ProductDetail/Bosch-Sensortec/BMP280?qs=QhAb4EtQfbUP9Z%252BCHM3Wyg==>,06/10/2020
- [82] <https://store.arduino.cc/arduino-uno-rev3> ,06/10/2020
- [83] <https://store.arduino.cc/arduino-nano> , 06/10/2020
- [84] <https://store.arduino.cc/arduino-mega-2560-rev3> 06/10/2020
- [85] <https://www.generationrobots.com/fr/402366-raspberry-pi-3-modele-b.html> ,
06/10/2020
- [86] <https://www.ldlc.com/fiche/PB00246555.html> , 06/10/2020
- [87] <https://www.ldlc.com/fiche/PB00261214.html> ,06/10/2020
- [88]<https://www.instructables.com/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/> , 06/10/2020

- [89] <https://www.moussasoft.com/product/module-rfid-rc522-lecteur-rfid> 89, 06/10/2020
- [90] https://ntas-store.cm/index.php?route=product/product&product_id=230 , 06/10/2020
- [91] <https://socialcompare.com/fr/comparison/raspberrypi-models-comparison> ,
06/10/2020
- [92] <https://microcontroleur.info/arduino-vs-raspberry-pi/> , 06/10/2020
- [93]<https://www.arduipianet.ma/comparatif-des-differentes-cartes-arduino/> , 06/10/2020
- [94] robot-maker.com/shop/arduino-officielles/107-arduino-nano-officiel-107.html ,
06/10/2020

Bibliographie

- [1] MESELLEM Ahmed, MOULA Mohammed Zinedine, Approche distribuée basée sur la diffusion pour la gestion des clés dans l'Internet des Objets, mémoire de master II, 2018.
- [3] Dechany, Maxime, L'impact de l'internet des objets sur le futur de la logistique et du transport: cas du transport routier », mémoire de Master, 2017-2018.
- [5] M.HANICHE Malika, m.TABRAIT, Nabila internet des objets dans le domaine de l'agriculture de demain, mémoire master II, 2019.
- [6] Mlle MEKRIOU RYMA, Mr MAZARI WALID Introduction à l'internet de l'objet et réalisation d'un système domotique, Mémoire de master, 2016.
- [16] Oubira Bilal, Mémoire, Conception et implémentation du système « Radio Frequency Identification» à l'aide d'une carte Arduino et lecteur RFID, master II, -2017.
- [17] M.HANICHE Malika, m.TABRAIT Nabila, internet des objets dans le domaine de l'agriculture de demain, , , mémoire master II, 2019.
- [29] Rabeb Saad, Modèle collaboratif pour l'Internet of Things (IoT), Mémoire master II, 2016,
- [37] DEHIMI Belkacem et AOUIA elmoueyed bilal nabih, Etude, adaptation et Réalisation d'un Système Electronique pour sauver de vie des Infarctus "Accident Cardiaque" Mémoire master II, année 2017/2018.
- [40] ADDA Ahmed. DJEBABRA Mohamed Nabil Développement d'une application de télé-opération d'un Système chauffage commande par internet, Mémoire master II, 2018.
- [41] Mr. MERZOUGUI Khelifa, Mr. BABA AHMED Adil, Proposition d'une extension Odoo pour la gestion des appels d'offres sur les engins. Cas d'application : L'entreprise algérienne, Mémoire master II 2015-2016.
- [43] BENIKHLEF Meriem et BENOTMANE Ikram, Conception et réalisation d'un système d'acquisition de données d'un capteur ultrasons HC-SR04 en utilisant un algorithme de filtre de Kalman, Mémoire master II, année 2019.