



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : 102/M2/2020

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Réseaux et technologies de l'information et de la
communication

Composition des services dans l'internet des objets

Par :

DJAFRI WALID

Soutenu le 30 septembre 2020, devant le jury composé de :

Nom et prénom

Grade

Président

Meklid abdessalam

Grade

Rapporteur

Nom et prénom

Grade

Examineur

ملخص:

اصبحت تكنولوجيا عاملا أساسيا في تحسين عديد من مجالات المدنية والصحية والعسكرية ومن أهم تكنولوجيات الجديدة إنترنت الأشياء التي تطورت سريعا لتدخل في جميع المجالات المذكورة سلفا
نقوم في هذا المشروع بدراسة معمقة حول إنترنت الأشياء ليتم بعده ذلك تطبيقها على نظام مركب لدفيئة زراعية لتحكم في عوامل مهمة دخلها كالحرارة ورطوبة وإضاءة عبر تطبيق اندرويد.
مع هذا النظام يمكن للمزارع الاطلاع على تفاصيل المناخ داخل الدفيئة الزراعية والتحكم بها باستمرار ومن أي مكان فيه اتصال بإنترنت وهذا لضمان إنتاج جيد وضمان راحة العميل.

Résumé :

La technologie est devenue un facteur fondamental dans l'amélioration de nombreux domaines civils, sanitaires et militaires, et l'une des nouvelles technologies les plus importantes est l'Internet des objets, qui s'est rapidement développé pour intervenir dans tous les domaines susmentionnés.

Dans ce projet, nous menons une étude approfondie sur l'Internet des objets, suivie de son application à un système complexe pour une serre agricole pour contrôler des facteurs importants tels que la température, l'humidité et l'éclairage, via une application Android.

Avec ce système, l'agriculteur peut voir les détails du climat à l'intérieur de la serre et le contrôler en continu et de n'importe où avec une connexion Internet, ceci pour assurer une bonne production et assurer le confort du client.

Abstract :

Technology has become a fundamental factor in improving many areas of civil, health and military, and one of the most important new technologies is the Internet of Things, which has rapidly developed to intervene in all the aforementioned areas.

In this project, we conduct an in-depth study on the Internet of Things, to be followed by its application to a complex system for an agricultural greenhouse to control important factors such as temperature, humidity and lighting, through an Android application.

With this system, the farmer can see the details of the climate inside the greenhouse and control it continuously and from anywhere with an Internet connection, and this is to ensure good production and ensure the customer's comfort.

Remerciement

*Avant toute chose, je remercie **Dieu** le tout puissant de m'avoir donné toute cette force, qui m'a permis d'arriver à bon port.*

*Je remercie s'adressent particulièrement mon encadreur **Meklid Abdesslam** qui a bien voulu diriger ce sujet. Pour son aide, ses encouragements jusqu'à la finalisation de ce travail.*

*Je remercie également **les membres du jury** qui m'honorent en acceptant le jugement pour mon travail.*

*Je remercie tous **les enseignants** du département d'Informatique pour la qualité de l'enseignement tout au long de notre cursus universitaire.*

*Je remercie mes **familles** qui ont su m'apporter leur soutien inlassable pendant toutes ces années de scolarité.*

*Je remercie toutes **les personnes** qui m'ont aidé de près ou de loin à accomplir notre travail.*

Merci, à tous...

Dédicace

Je dédie cet humble travail à mes chers parents

À mon frère et mes sœurs à ma grand-mère et à toute la famille

À tous mes professeurs

À tous mes amis et collègues

Je n'oublie pas les gens qui m'ont aidé

Table des matières

Table des matières	I
Liste des Figures	IV
Liste des tableaux	VI
Introduction générale	1
CHAPITRE 1 : Internet des objets	3
1. Introduction.....	3
2. Définition de l'internet des objets.....	3
3. L'évolution de l'écosystème de l'internet des objets.....	4
4. L'IoT en tant que réseau de réseaux.....	5
5. Architecture de l'internet des objets.....	5
6. Domaines d'application.....	7
6.1 La domotique en milieux urbains.....	7
6.2 L'énergie.....	7
6.3 Le transport.....	7
6.4 La santé.....	7
6.5 L'industrie.....	7
6.6 L'agriculture.....	8
7. Fonctionnement de l'IdO.....	8
7.1 Technologies de l'IdO.....	8
7.2 Protocoles de fonctionnement de l'IdO.....	9
7.2.1 CoAP (Constrained Application Protocol)	9
7.2.2 MQTT (Message Queue Telemetry Transport)	10
8. Les Réseaux de Capteurs Sans Fil (RCSFs)	10
8.1 Définition.....	10
8.2 Architecture d'un nœud capteur.....	11
9. Architecture des Réseaux de Capteurs Sans Fil.....	11
9.1 Architecture de communication.....	11
9.2 Pile protocolaire.....	12
9.3 Les plans de gestion de la pile protocolaire.....	13
9.3.1. <i>Plan de gestion de l'énergie</i>	13
9.3.2. <i>Plan de gestion de la mobilité</i>	13
9.3.3. <i>Plan de gestion des tâches</i>	13
9.4 Les couches de la pile protocolaire	13
10. Les caractéristiques d'un système d'IoT.....	15
11. Conclusion.....	16
CHAPITRE 2 : Composants d'un objet connecté	17
1. Introduction.....	17
2. Microcontrôleur	17
3. Cartes Arduino.....	19
4. Module de connectivité.....	21
5. Les capteurs	22
5.1 Définition d'un capteur.....	22
5.2 Classification des capteurs.....	23

5.3	Caractéristiques des capteurs.....	24
5.4	Exemples de capteurs.....	24
6.	Les cartes d'essai.....	26
7.	Les câbles.....	27
8.	LED'S.....	29
9.	Les résistances.....	29
10.	Les Plateformes D'IoT.....	29
10.1	La plate-forme AWS IoT (Amazon Web Service).....	30
10.2	Google Cloud Platform.....	31
10.3	IBM Cloud Platform.....	33
10.4	Comparaison des plateformes.....	35
11.	Les réseaux IoT	38
11.1	Comment choisir un réseau ?.....	38
11.2	Les réseaux personnels.....	39
11.3	Les réseaux du bâtiment.....	39
11.4	Les réseaux LPWAN (Low Power Wide Area Network).....	40
12.	Conclusion.....	41
CHAPITRE 3 : Serres Intelligentes.....		42
1.	Introduction.....	42
2.	Serres Intelligentes.....	42
2.1	Définition.....	42
2.2	Les différents types de serres.....	43
2.2.1	Serres tunnel.....	43
2.2.2	Serres multi-chapelle.....	44
2.2.3	Serres en verre.....	45
2.3	Structure et matériaux pour la serre.....	45
2.4	Conditions environnementales dans les serres.....	45
2.5	Contrôle des conditions environnementales.....	46
2.5.1	Contrôle de lumière.....	46
2.5.2	Gaz carbonique.....	46
2.5.3	Humidité de l'air.....	46
2.5.4	Température.....	47
2.6	Systèmes de contrôle.....	47
2.6.1	Systèmes d'irrigation.....	47
2.6.2	Système d'ouverture des serres.....	48
2.6.3	Surveillance de l'environnement.....	48
3.	Travaux IdO actuelles pour les serres.....	49
4.	Conclusion.....	50
CHAPITRE 4 : Conception et réalisation de système des serres		51
1.	Introduction.....	51
2.	Objectifs et principe de la solution proposée.....	51
2.1	Objectif du projet.....	51
2.2	Principe de la solution proposé.....	52

3. Architecture globale du système.....	53
3.1 Modules du système.....	53
3.1.1 Interface physique de capture et commande.....	53
3.1.2 Module de Perception et Reconnaissance de situations.....	54
3.1.3 Module de décision.....	54
3.2 Principe de fonctionnement.....	55
4. Gestion de serre.....	55
4.1 Gestion Température.....	55
4.2 Gestion Humidité.....	56
4.3 Gestion d'éclairage.....	56
5. Diagramme de classes.....	57
5.1 Classe Centre d'alerte.....	59
5.2 Classe Equipement.....	59
5.3 Classe capteur.....	60
5.4 Classe évènement.....	60
5.5 Classe SmartFarm.....	60
5.6 Classe GreenHouse.....	61
6. Diagramme d'activité.....	61
7. Conclusion.....	62
CHAPITRE 5 : Développement et Tests.....	63
1. Introduction.....	63
2. Cloud Storage « ThinkSpeak ».....	63
2.1 Fonctions de ThingSpeak.....	63
2.2 Les étapes pour utiliser ThingSpeak.....	63
3. Montage et programmation de matériel.....	66
3.1 DHT22 : Capteur de température et d'humidité.....	66
3.2 Capteur de luminosité.....	68
3.3 Capteur d'humidité du sol.....	69
3.4 L'affichage sur LCD.....	71
3.5 Actionneurs.....	72
3.6 Tests de capteurs ensemble.....	72
3.7 ESP8266 (Module Wifi).....	73
3.8 Data Storage Cloud.....	74
4. Développement et présentation de l'application.....	76
4.1 Vue générale de la structure du projet.....	76
4.2 Interfaces de l'application.....	76
5. Conclusion.....	80
Conclusion générale.....	81
Références.....	82

Liste des Figures

Figure 1.1 - L'Ido connecte des objets en utilisant des capteurs et Internet [1].....	3
Figure 1.2 - L'évolution d'IdO entre 2003 et 2020 [2].....	4
Figure 1.3 - L'I0dO est en quelque sorte un réseau de réseaux [2].....	5
Figure 1.4 - Architecture d'IdO [3].....	6
Figure 1.5 - Divers domaines d'applications de l'IdO.....	8
Figure 1.6 - Les étiquettes RFID [17].....	9
Figure 1.7 - Fonctionnement du protocole CoAP [15].....	10
Figure 1.8 - Fonctionnement du Protocole MQTT [18].....	10
Figure 1.9 - Composants d'un nœud capteur.....	11
Figure 1.10 - Architecture de communication d'un RCSF.....	12
Figure 1.11 - Architecture protocolaire d'un RCSF [21].....	13
Figure 2.1 - Microcontrôleur.....	17
Figure 2.2 - Carte Arduino UNO.....	19
Figure 2.3 - Carte Arduino Yun.....	19
Figure 2.4 - Arduino NANO.....	20
Figure 2.5 - Carte Seeeduino.....	20
Figure 2.6 - Arduino Leonardo.....	20
Figure 2.7 - Netduino.....	21
Figure 2.8 - Carte Arduino MEGA.....	21
Figure 2.9 - Capteur PIR.....	24
Figure 2.10 - Capteur ultrason hc-sr04.....	24
Figure 2.11 - Capteur PIR GROVE.....	25
Figure 2.12 - Capteur ultra son ADAFRUIT VL53L0X.....	25
Figure 2.13 - Capteur TMP36.....	25
Figure 2.14 - Capteur CTN 10K.....	25

Figure 2.15 - Capteur DHT 1.....	26
Figure 2.16 - Capteur pro-Grove.....	26
Figure 2.17 - Capteur GPS Shield.....	26
Figure 2.18 - Breadboard 400.....	27
Figure 2.19 - Breadboard PCB – Single.....	27
Figure 2.20 - Câble alimentation Arduino pour piles 9V.....	27
Figure 2.21 - Câble USB vers fiche DC JACK 2.1mm – 1m.....	28
Figure 2.22 - Câble antenne d'adaptation RP SMA vers uFL--u.FL-IPX-IPEX-RF.....	28
Figure 2.23 - Câble USB.....	28
Figure 2.24 - Les LED.....	29
Figure 2.25 - Une résistance.....	29
Figure 2.26 - Architecture AWS IoT.....	31
Figure 2.27 - Architecture Google Cloud plateforme.....	32
Figure 2.28 - architecture Ibm wason.....	34
Figure 2.29 - Types réseaux sans fil.....	38
Figure 3.1 - Exemple d'une serre agricole.....	43
Figure 3.2 - Une serre tunnel.....	43
Figure 3.3 - Serres multi-chapelle.....	44
Figure 3.4 - Serres en verre [39].....	45
Figure 3.5 - Les éléments principaux qui composent un système de serre intelligente..	48
Figure 4.1 - Éléments du système.....	53
Figure 4.2 - Architecture global du système.....	54
Figure 4.3 - Logigramme de fonctionnement de la température.....	55
Figure 4.4 - Logigramme d'humidité.....	56
Figure 4.5 - Logigramme d'éclairage.....	57
Figure 4.6 - Diagramme des classes.....	58
Figure 4.7 - Classe Centre d'alerte.....	59
Figure 4.8 - Class Equipement.....	59
Figure 4.9 - Classe capteur.....	60
Figure 4.10 - Classe évènement.....	60

Figure 4.11 - Class SmartFarm.....	60
Figure 4.12 - Classe GreenHouse.....	61
Figure 4.13 - Diagramme d'activité.....	61
Figure 5.1 - Création de compte ThingSpeak.....	64
Figure 5.2 - Création un nouveau canal.....	64
Figure 5.3 - Les détails du nouveau canal.....	65
Figure 5.4 - Obtenir L'API Thingspeak.....	65
Figure 5.5 - Montage du système final.....	66
Figure 5.6 - DHT2.....	67
Figure 5.7 - Montage Arduino et DHT22.....	67
Figure 5.8 - Montage Arduino et LDR.....	69
Figure 5.9 - Le circuit de capteur humidité du sol.....	70
Figure 5.10 - Montage Arduino et LCD.....	71
Figure 5.11 - Montage Arduino et actionneur.....	72
Figure 5.12 - Affichage des informations de système.....	73
Figure 5.13 - Montage Arduino et ESP8266-01.....	73
Figure 5.14 - Vue générale de la structure du projet.....	76
Figure 5.15 - Interface de chargement.....	77
Figure 5.16 - Interface d'accueil et menu.....	77
Figure 5.17 - Interface de suivi de la météo.....	78
Figure 5.18 - Interface de contrôle de la pompe.....	78
Figure 5.19 - Interface de contrôle de la lampe.....	79
Figure 5.20 - Interface de contrôle du ventilateur.....	79

Liste des tableaux

Tableau 1 - Description du système en couche pour les réseaux de capteurs.....	14
Tableau 2 - Comparaisons entre les Platform.....	37
Tableau 3 - Les composants du système.....	66
Tableau 4 - Les canaux ThingSpeak.....	75

Introduction générale

Depuis la fin des années 1980, Internet a évolué de manière spectaculaire. La dernière étape est l'utilisation de ce réseau mondial pour la communication avec des objets ou entre objets, évolution nommée Internet des Objets (IoT pour Internet of Things). L'évolution de l'IoT est rapide : depuis 2014, le nombre d'objets connectés est supérieur au nombre d'humains connectés et il est prévu que 50 milliards d'objets seront connectés en 2020.

Nous présenterons les caractéristiques essentielles de l'Internet des Objets. Il commence par introduire une vue d'ensemble, avec les nombreux types d'applications où l'IoT introduit de l'intelligence : réseaux, santé, maison, ville, industrie, télévision, automobile, etc. Puis, l'architecture générale est détaillée avec ses différents niveaux : détection et identification des informations, construction du réseau, traitement de l'information et actions nécessaires pour l'application, auquel correspond le schéma en couches : niveau composants, niveau réseau, niveau support (service et application) et niveau application.

L'ensemble des domaines concernés est assez large pour innover et créer de la valeur. Ce sujet offre un panorama de l'ensemble des éléments de la chaîne de valeurs : passerelles multitechnologies entre objets et réseau pour connecter le maximum d'objets, infrastructures de télécommunications dédiées à l'IoT, plateformes spécifiques, intégration de technologies pour des solutions sur mesure, opérateurs du service de l'objet connecté [37].

- Les composants matériels et leur environnement de développement sont illustrés. Ils comprennent : les identificateurs, capteurs, afficheurs, actuateurs au niveau des objets ;
- Les microcontrôleurs ou processeurs et cartes bas coût sans ou avec OS léger pour les objets simples, des processeurs avec OS comme Linux, Android, IOS et les ressources du nuage pour les objets complexes ;
- Les réseaux, qui peuvent être « courte distance », « longue distance » ou « cellulaires ». Les différentes technologies de réseau utilisables sont détaillées en fonction des contraintes : portée, débit, coût, sécurité, etc.

Dans le domaine de l'agriculture, la serre agricole pourra à son tour profiter des avantages de l'Internet des objets afin de se transformer à une serre intelligente.

En effet, les serres agricoles intelligentes sont capables de s'adapter à une certaine évolution de l'environnement à l'aide des entrées et des sorties mais aussi des moyens de communication et même les interfaces électroniques de traitement.

L'usage d'un moyen de communication sans fil est l'outil très primordiale et idéale pour améliorer le confort, la sécurité des biens et des personnes ainsi que la réduction de la consommation de l'énergie que ce soit dans le côté économique ou bien le côté gaspillage.

De ce fait, l'agriculture de demain sera automatisée et la production agricole s'appuiera sur la notion de la serre agricole intelligente qui va gérer non seulement les critères d'humidité, de lumière et d'arrosage mais aussi la facilité du contrôle et d'accès par l'agriculteur qui lui permettra certainement une production plus optimale sur tous les plans.

Internet des objets permet aux agriculteurs de prendre le contrôle de leurs champs agricoles à distance en ce qui concerne des paramètres tels que :

- L'humidité.
- La lumière nécessaire à la production agricole.
- L'arrosage.
- La température.

A travers notre application, nous voulons rendre la gestion des informations une tâche facile vis-à-vis les agriculteurs qui n'auront qu'à installer notre application sur leurs Smartphones.

Ce travail, réalisé dans le cadre de notre projet de fin d'études en Master, est décrit dans ce mémoire qui est structuré en cinq chapitres :

- Nous consacrons le premier chapitre à quelques définitions et généralités liées au domaine de l'Internet des objets.
- Le deuxième chapitre est consacré à la présentation des composants d'un objet connecté.
- Le troisième chapitre présente les serres agricoles et leur développement en agriculture, leur utilisation, ces composants, ces types sont les plus utilisés et l'effet du climat sur les serres.
- Le quatrième chapitre, nous présentons les objectifs et principe de la solution proposée et l'architecture global du système.
- Le chapitre cinq est réservé à notre application développée. Nous allons donc représenter les différentes étapes de ce dernier développement ainsi que l'environnement de programmation.
- Enfin, il résume la conclusion générale des cinquièmes chapitres de cette thèse, mesure les résultats obtenus et donne des perspectives pour ce projet.

CHAPITRE 1 :

Internet des objets

1. Introduction

L'internet of things, ce nom peut paraître un peu abstrait à première vue, voir même insignifiant pour certains, mais c'est une révolution technologique qui est déjà en marche et qui fait de plus en plus partie de notre quotidien. La traduction française, qui est "l'internet des objets" peut vous donner quelques pistes de compréhension du sujet, mais pas totalement en réalité. La vie futur, l'entreprise de demain, connectée et autonome ne peut s'envisager sans l'IoT (Internet of Things), c'est vous dire l'importance de cette révolution. Dans cet chapitre, je vais essayé de vous expliquer l'IoT de manière simplifiée.

2. Définition de l'internet des objets

L'Internet des objets (IdO) est une « *infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution* ». En réalité, la définition de ce qu'est l'Internet des objets n'est pas figée. Elle recoupe des dimensions d'ordres conceptuel et technique.

D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel.

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée (adresse IP, protocoles smtp, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wi-Fi [14].

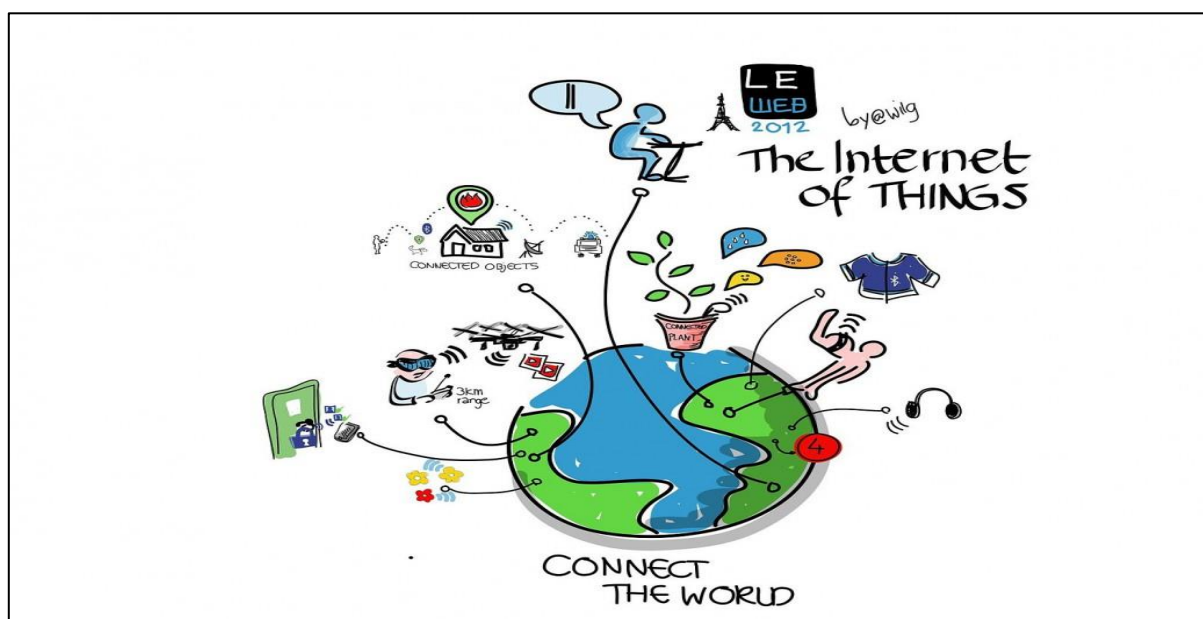


Figure 1.1 - L'Ido connecte des objets en utilisant des capteurs et Internet [1]

3. L'évolution de l'écosystème de l'internet des objets

Les premiers objets connectés n'apparaissent que dans les années 1990. Il s'agit de grille-pain, machines à café ou autres objets du quotidien. En 2000, le fabricant coréen LG est le premier industriel à parler sérieusement d'un appareil électroménager relié à internet, Les années 2000 verront les premières expérimentations d'appareils connectés à Internet. Ils l'utilisent notamment pour consulter des informations de manière automatique.

En 2003, la population mondiale s'élevait à environ 6,3 milliards d'individus et 500 millions d'appareils étaient connectés à Internet. Le résultat de la division du nombre d'appareils par la population mondiale (0,08) montre qu'il y avait moins d'appareil connecté par personne. Selon la définition de Cisco IBSG, l'Ido n'existait pas encore en 2003 car le nombre d'objets connectés était faible.

En raison de l'explosion des Smartphones et des tablettes, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards.

C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 (1,84 pour être exact) pour la première fois de l'histoire.

En affinant ces chiffres, Cisco IBSG a situé l'apparition de l'ido entre 2008 et 2009 (voir Figure 1.2).

En ce qui concerne l'avenir, Cisco IBSG estime que 50 milliards d'appareils seront connectés à Internet d'ici à 2020. Il est important de noter que ces estimations ne tiennent pas compte des progrès rapides d'Internet ni des avancées technologiques, mais reposent sur les faits avérés à l'heure actuelle [2]

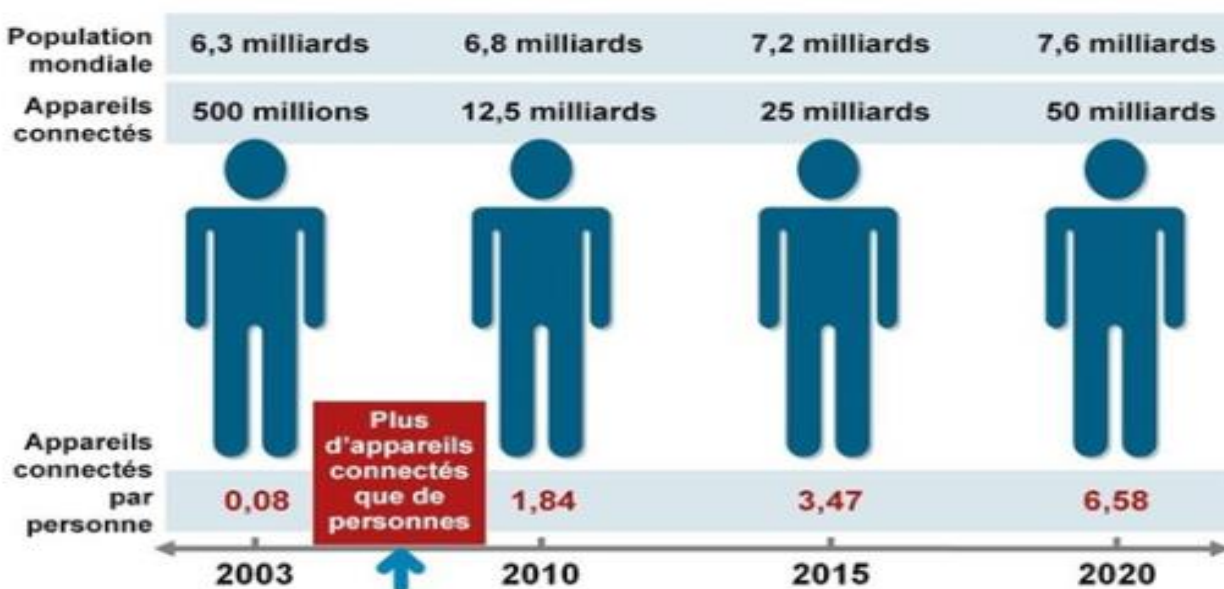


Figure 1.2 - L'évolution d'IdO entre 2003 et 2020 [2]

4. L'IoT en tant que réseau de réseaux

Actuellement, l'IoT se compose d'un ensemble hétérogène de réseaux spécialisés disparates. Prenons l'exemple des voitures récentes. Plusieurs réseaux permettent de contrôler le fonctionnement du moteur, les dispositifs de sécurité, les systèmes de communication, etc. Les bâtiments commerciaux et résidentiels sont également équipés de différents systèmes de contrôle dédiés aux dispositifs de chauffage, d'aération et d'air conditionné et au téléphone, à la sécurité ou encore à l'éclairage. Grâce à l'évolution de l'IoT, ces réseaux, et bien d'autres encore, seront connectés à des fonctions évoluées de sécurité, d'analyse et de gestion (voir figure 2). C'est ainsi que l'IoT nous aidera encore mieux à réussir ce que nous entreprenons.

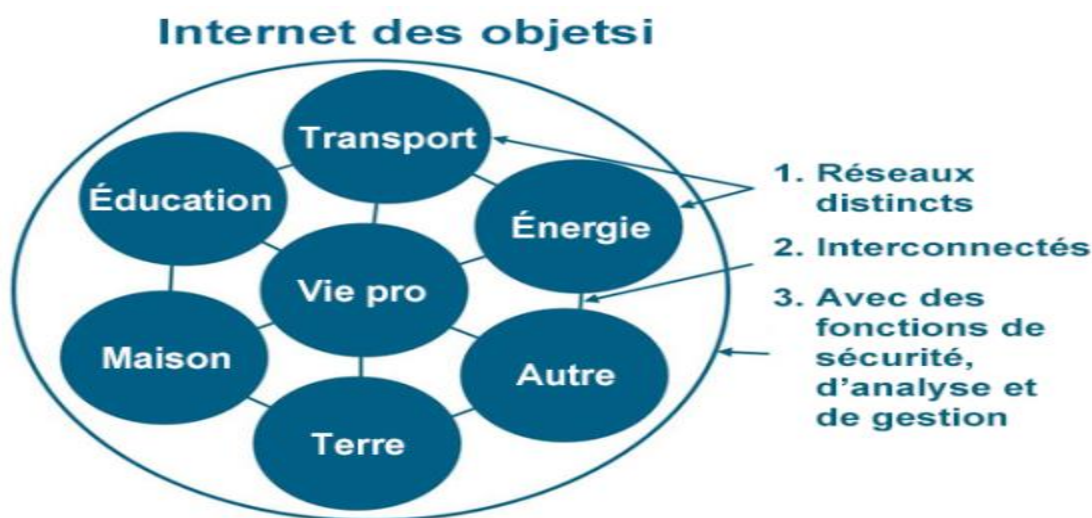


Figure 1.3 - L'IoD est en quelque sorte un réseau de réseaux [2].

5. Architecture de l'internet des objets

On distingue deux types d'objet :

- **Les objets passifs** : ils utilisent généralement un tag (puce RFID, code barre 2D). Ils embarquent une faible capacité de stockage (de l'ordre du kilo-octet) leur permettant d'assurer un rôle d'identification. Ils peuvent parfois, dans le cas d'une puce RFID, embarquer un capteur (température, humidité) et être réinscriptibles.
- **Les objets actifs** : ils peuvent être équipés de plusieurs de capteurs, d'une plus grande capacité de stockage, être doté d'une capacité de traitement ou encore être en mesure de communiquer sur un réseau.

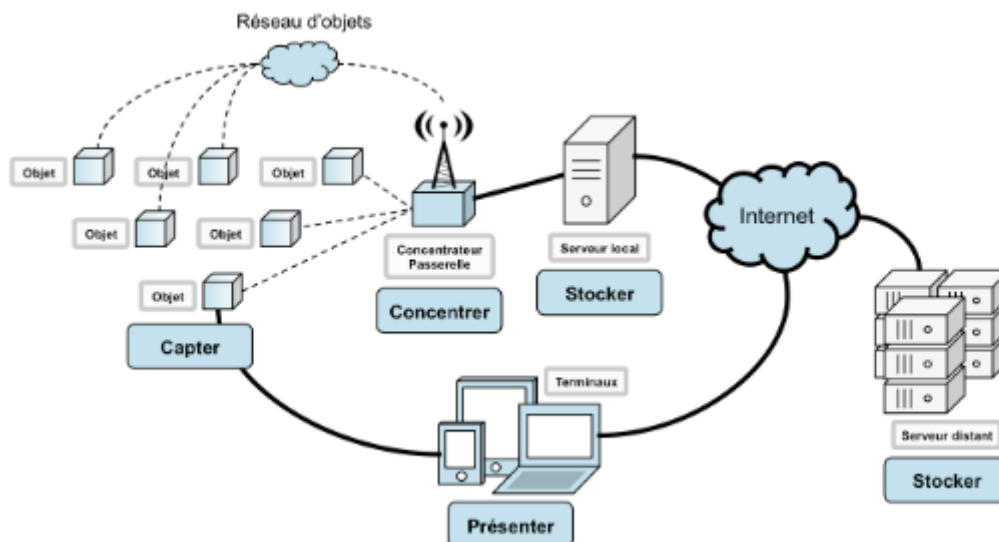


Figure 1.4 - Architecture d'IdO [3]

Précisons le rôle des différents processus présentés sur ce schéma :

- **Capter** désigne l'action de transformer une grandeur physique analogique en un signal numérique.
- **Concentrer** permet d'interfacer un réseau spécialisé d'objet à un réseau IP standard (e.g. WiFi) ou des dispositifs grand public.
- **Stocker** qualifie le fait d'agréger des données brutes, produites en temps réel, méta taguées, arrivant de façon non prédictible.
- Enfin, **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.

Deux autres processus n'apparaissent pas sur le schéma, car ils sont à la fois transverses et omniprésents :

Le **traitement des données** est un processus qui peut intervenir à tous les niveaux de la chaîne, depuis la capture de l'information jusqu'à sa restitution. Une stratégie pertinente, et commune quand on parle d'Internet des objets, consiste à stocker l'information dans sa forme intégrale. On collecte de manière exhaustive, « big data », sans préjuger des traitements qu'on fera subir aux données. Cette stratégie est possible aujourd'hui grâce à des architectures distribuées type NoSQL, capables d'emmagasiner de grandes quantités d'information tout en offrant la possibilité de réaliser des traitements complexes en leur sein (Map/Reduce par exemple).

La **transmission des données** est un processus qui intervient à tous les niveaux de la chaîne. Deux réseaux, supports des transmissions, cohabitent généralement :

- **Réseau local de concentration.** On utilise alors des technologies comme ANT, ZigBee, Z-wave, NFC ou Bluetooth LE.
- **Réseau WAN**, permettant d'interconnecter les réseaux spécialisés et de les interfacer avec des fermes de serveur. On utilise alors WiFi, les réseaux cellulaires (GSM, UMTS, LTE) ou encore les connexions physiques standard (Ethernet, fibre optique). Ces réseaux sont généralement connectés à Internet.

6. Domaines d'application

Les potentialités offertes par l'IdO et son aspect ubiquitaire permettent de développer de nombreuses applications. Cependant, seules quelques applications sont actuellement déployées. L'utilisation de l'IdO permettra le développement de plusieurs applications intelligentes à l'avenir qui toucheront essentiellement : la domotique, les villes, le transport, la santé et l'industrie. Dans ce qui suit, nous citons brièvement des exemples d'applications de l'IdO .

6.1 La domotique en milieux urbains

L'un des domaines d'application de l'IdO les plus intéressants concerne la mise des équipements domestiques sur réseau. Cela permet d'abord d'avoir un contrôle global des différents équipements techniques d'une maison depuis une même interface (une tablette ou un téléphone intelligent par exemple), mais aussi, il offre la possibilité de contrôler à distance ces équipements via la mise à disposition d'API sur le web.

Le champ d'application de l'IdO s'étale pour toucher les villes (smart cities), l'IdO permettra une meilleure gestion de tous les réseaux qui alimentent ces villes intelligentes (eaux, électricité, gaz, etc.). Des capteurs peuvent être utilisés pour améliorer la gestion des parkings et du trafic urbain, diminuer les embouteillages et les émissions en CO₂.

6.2 L'énergie

L'IdO propose des possibilités de gestion en temps réel de l'infrastructure de distribution de l'énergie, comme les réseaux électriques intelligents (smart grid). Cela contribuera d'une manière sûre à faciliter la maintenance et le contrôle de la consommation et la détection des fraudes.

6.3 Le transport

Grâce à l'IdO la communication inter véhicule et entre véhicules et infrastructures routière sera possible. De ce fait, ce domaine de l'IdO peut se considérer comme un prolongement des systèmes de transport intelligents qui a comme objectifs non seulement le renforcement de la sécurité routière et l'aide à la conduite, mais également, l'efficacité de la gestion du trafic, économie du temps, de l'énergie et le confort des conducteurs.

6.4 La santé

Ce domaine de l'IdO assurera le contrôle et le suivi des signes cliniques des patients par la mise en place des réseaux personnels de surveillance, ces réseaux seront constitués de biocapteurs posés sur le corps des patients ou dans leurs lieux d'hospitalisation ou même dans leurs domiciles. Cela facilitera la télésurveillance des patients et apportera des solutions pour l'autonomie des personnes à mobilité réduite.

6.5 L'industrie

Le déploiement de l'IdO dans l'industrie sera certainement un grand support pour le développement de l'économie et le secteur des services, puisque l'IdO permettra d'assurer un suivi total des produits, de la chaîne de production, jusqu'à la chaîne logistique et de distribution en supervisant les conditions d'approvisionnement. Cette traçabilité de bout en

bout facilitera la lutte contre la contrefaçon, la fraude et les crimes économiques transfrontaliers.

6.6 L'agriculture

L'IdO présentera des outils de choix pour la supervision de l'environnement des cultures, ce qui permettra une meilleure aide à la décision en agriculture. L'IdO servira non seulement à optimiser l'eau d'irrigation, l'usage des intrants et la planification des travaux agricoles, mais aussi, cette technologie peut être utilisée pour lutter contre la pollution (l'air, le sol et les eaux) et améliorer la qualité de l'environnement en général.



Figure 1.5 – Divers domaines d'applications de l'IdO.

7. Fonctionnement de l'IdO

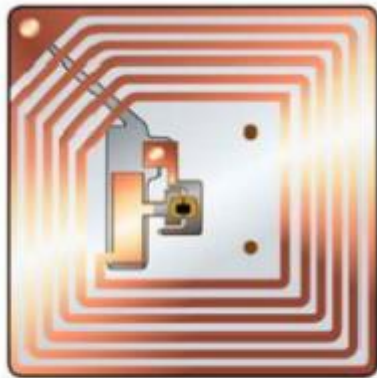
7.1 Technologies de l'IdO

L'Internet des objets (IdO) permet l'interconnexion de différents objets intelligents via l'Internet. Son fonctionnement est basé sur l'utilisation de plusieurs technologies. Nous citons, entre autres [17] :

— **Radio Frequency Identification (RFID)** : le terme RFID englobe toutes les technologies qui utilisent les ondes 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 radio. 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. Les étiquettes peuvent avoir différentes formes (Figure 1.6) et peuvent être passives ou actives.

— **Réseau de Capteurs Sans Fil (RCSF)** : c'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. 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 inclure divers types de capteurs et des actionneurs.

— **Machine to Machine (M2M)** : c'est l'association des technologies de l'information et de la communication avec des objets intelligents dans le but de donner à ces derniers les moyens d'interagir sans intervention humaine avec le système d'information d'une organisation ou d'une entreprise.



Étiquette passive



Étiquette active

Figure 1.6 – Les étiquettes RFID [17].

7.2 Protocoles de fonctionnement de l'IdO

La mise en application à une large échelle du concept d'IdO apparaît largement tributaire d'une standardisation de la communication entre Objets dite M2M [17].

Au niveau de la couche de Liaison, le standard IEEE 802.15.4 est plus adapté que l'Ethernet aux environnements industriels difficiles.

— Au niveau réseau, le standard IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) a réussi à adapter le protocole IPv6 aux communications sans fil entre nœuds à très faible consommation.

— Au niveau routage, l'IETF a publié en 2011 le standard Routing Protocol for Lossy Networks (RPL) .

— Au niveau de la couche application, les protocoles les plus connus sont le protocole Constrained Application Protocol (CoAP) qui utilise le protocole de transport User Datagramm Protocol (UDP), et le protocole Message Queuing Telemetry Transport (MQTT) qui utilise le protocole de transport Transfert Control Protocol (TCP).

7.2.1 CoAP (Constrained Application Protocol) : est un protocole de la couche d'application pour les applications IdO. Il définit un protocole de transfert Web basé sur les fonctionnalités HTTP, et est lié à UDP (et non TCP) par défaut qui le rend plus approprié pour les applications IdO. En outre, CoAP modifie certaines fonctionnalités HTTP pour répondre aux exigences de l'IdO telles que la faible consommation d'énergie et le fonctionnement en présence de liens à perte [17].

7.2.2 MQTT (Message Queue Telemetry Transport) : représente un protocole de messagerie idéal pour les communications IdO et M2M. MQTT utilise le modèle de publication/souscription pour offrir une flexibilité de transition et une simplicité d'implémentation. Il convient aux périphériques à ressources limitées qui utilisent des liens peu fiables ou à faible bande passante. MQTT est construit en haut du protocole TCP. Il se compose de trois composants ; abonnés, éditeurs et courtiers. De nombreuses applications utilisent MQTT telles que les soins de santé, la surveillance, le compteur d'énergie et la notification de Facebook [17].

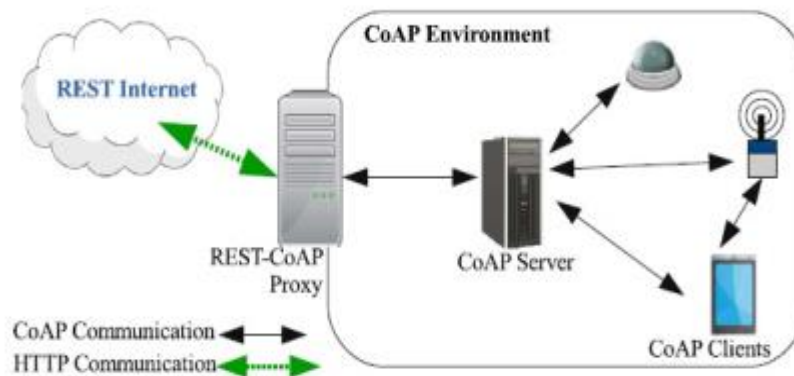


Figure 1.7 – Fonctionnement du protocole CoAP [15].

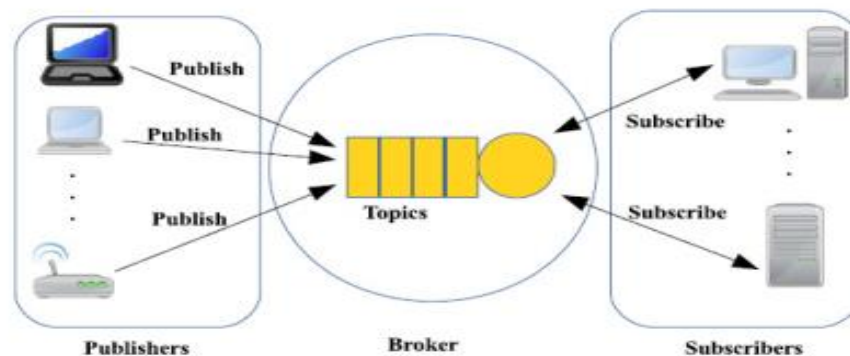


Figure 1.8 – Fonctionnement du Protocole MQTT [18]

8. Les Réseaux de Capteurs Sans Fil (RCSFs)

8.1 Définition

Un Réseau de Capteurs Sans Fil (RCSF) est un ensemble de dispositifs très petits, nommés nœuds capteurs, variants de quelques dizaines d'éléments à plusieurs milliers. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil

8.2 Architecture d'un nœud capteur

Un nœud capteur est composé de quatre unités principales, qui sont présentées dans la figure 1.9

- **Unité de capture (Sensing unit)** : elle est généralement composée de deux sous-unités, les capteurs et les convertisseurs analogique-numérique (appelé CAN). Les signaux analogiques produits par les capteurs basés sur le phénomène observé sont convertis en signaux numériques par le CAN, puis transmis à l'unité de traitement [20].
- **Unité de traitement (Processing unit)** : est composée de deux interfaces, une interface avec l'unité de capture et une autre avec l'unité de transmission. Cette unité est également composée d'un processeur qui est la partie la plus importante d'un nœud capteur, et une petite unité de stockage. Elle gère les procédures permettant au nœud capteur de collaborer avec les autres nœuds pour effectuer les tâches de capture, et stocker les données collectées [20].
- **Unité de transmission (Transceiver unit)** : est composée d'un module radio (émetteur/récepteur). Elle est responsable de toutes les émissions et réceptions de données sur un medium sans fil [20].
- **Unité d'énergie (Power unit)** : C'est un élément important dans l'architecture d'un nœud capteur, puisqu'elle permet de garder le fonctionnement du nœud capteur au sein d'un réseau pour une durée limitée. Cette unité se présente généralement sous la forme de batteries afin de fournir une énergie pour toutes les autres unités [20].

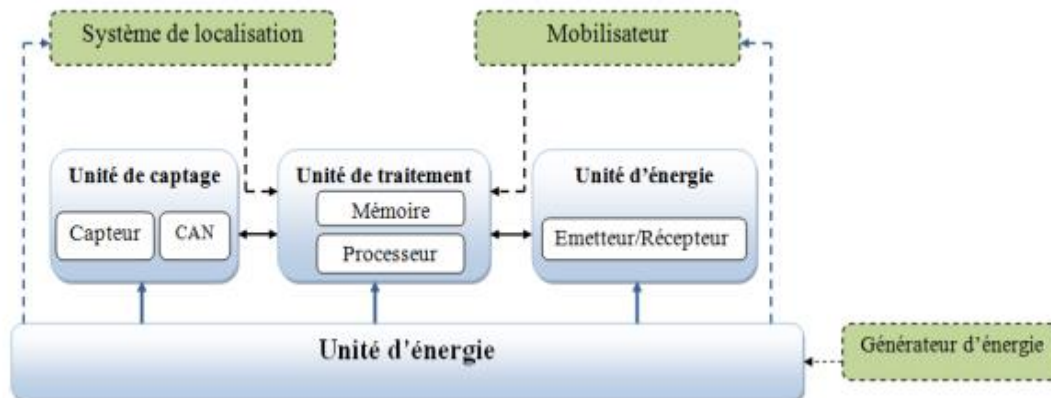


Figure 1.9 – Composants d'un nœud capteur.

9. Architecture des Réseaux de Capteurs Sans Fil

9.1 Architecture de communication

Un RCSF est constitué d'un ensemble de nœuds capteurs permettant de capturer des événements, traiter et de transmettre les informations recueillies à un nœud puits (sink), qui à son tour les envoie à un centre de traitement. Les nœuds capteurs ainsi que le nœud sink peuvent être fixes ou mobiles dans une zone de capture .

- **Nœuds capteurs** : leur type, leur architecture et leur position géographique dépendent de l'exigence plication en question. Leur énergie est souvent limitée puisqu'ils sont alimentés par des batteries [21].
- **Nœud Sink** : c'est un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un RCSF plus ou moins large et à charge un peu élevée, on peut trouver deux ou plusieurs sinks, et ce pour alléger la charge du réseau [21].
- **Centre de traitement des données** : c'est le centre vers lequel, les données collectées par le sink, sont envoyées (voir Figure 1.10) [21].

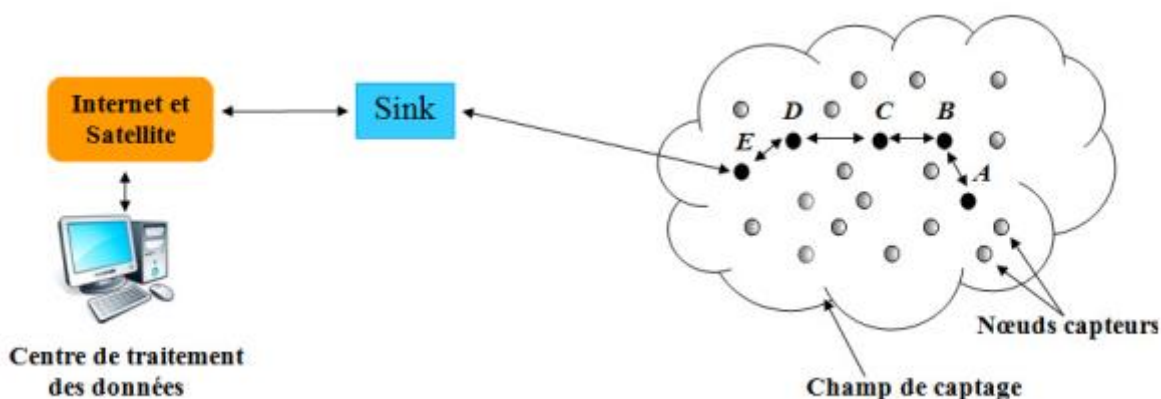


Figure 1.10 – Architecture de communication d'un RCSF

9.2 Pile protocolaire

La pile protocolaire utilisée par la station de base ainsi que tous les autres capteurs du réseau est illustrée par la figure 1.11. La pile protocolaire comprend la couche application, la couche transport, la couche réseau, la couche liaison de données, la couche physique, le plan de gestion de l'énergie, le plan de gestion de la mobilité et le plan de gestion des tâches.

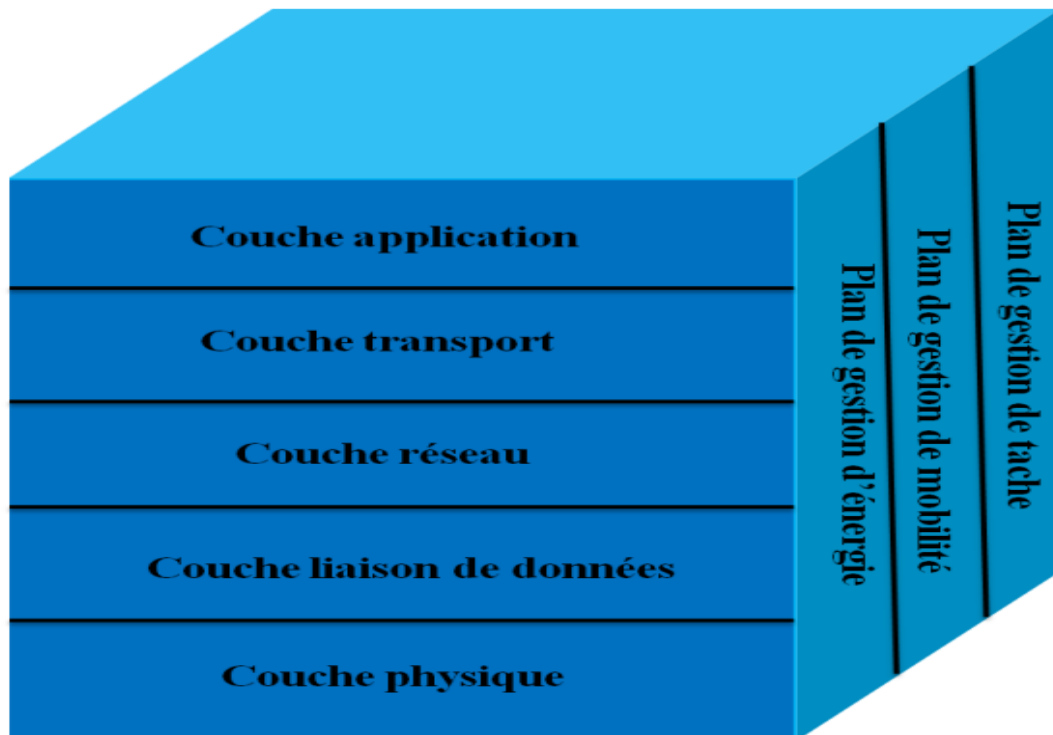


Figure 1.11 – Architecture protocolaire d'un RCSF [21].

9.3 Les plans de gestion de la pile protocolaire

9.3.1. Plan de gestion de l'énergie

L'objectif principal à ce niveau consiste à gérer l'énergie consommée par les capteurs. Un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués ou de limiter l'écoute inutile générée par la couche MAC. Ainsi, si le nœud possède un niveau d'énergie faible, il peut signaler aux autres participants au réseau le fait qu'il ne participe plus aux tâches de routage, pour consacrer le reste de son énergie au captage [21].

9.3.2. Plan de gestion de la mobilité

Ce niveau assure la détection et l'enregistrement de tous les mouvements des nœuds, afin de garder une trace de la route vers l'utilisateur final, et de conserver une cartographie à jour des nœuds voisins. Ceci est nécessaire pour équilibrer l'exécution des tâches et la consommation d'énergie [22].

9.3.3. Plan de gestion des tâches

Lors d'une opération de captage dans une zone géographique donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme. Cela dépend de plusieurs paramètres tels que la nature du capteur, le niveau d'énergie et la zone dans laquelle il a été déployé. À cette fin, ce niveau d'équilibrage et de distribution des tâches sur les différents nœuds a pour but d'assurer un travail coopératif et efficace [23].

9.4 Les couches de la pile protocolaire

La plupart du temps, les réseaux de capteurs sont modélisés selon le modèle en couches OSI (*Open System Interconnection*). De ce modèle, cinq couches concernent particulièrement les réseaux de capteurs.

Couche	Description
Application	Cette couche permet un interfaçage avec les applications, c'est le niveau le plus proche des utilisateurs.
Transport	Elle est chargée du transport des données, en passant par plusieurs étapes : le découpage en paquets, le contrôle de flux, la conservation de l'ordre des paquets et la gestion des erreurs de transmission.
Réseau	Elle permet de gérer l'acheminement de données via le réseau et intègre notamment le protocole de routage.
Liaison de données	<p>Cette couche assure plusieurs fonctionnalités à savoir :</p> <ul style="list-style-type: none"> - Elle spécifie comment les données sont expédiées entre deux nœuds ; - Elle est responsable du multiplexage des données, du contrôle d'erreurs, et de l'accès au canal ; - Elle assure une liaison point à point ou point à multipoint ; - Elle se décompose en une couche « lien logique » qui fournit une interface entre la couche liaison et la couche réseau, et en une couche « accès au médium » qui contrôle la radio.
Physique	Elle précise les spécifications matérielles de la communication, qu'elle soit filaire (câblage, connecteur...) ou, en ce qui concerne les réseaux de capteurs, sans fil (fréquences porteuses, modulation, etc.). Cette couche doit de plus assurer des techniques d'émission, de réception et de modulation de données de manière robuste

Tableau 1 - Description du système en couche pour les réseaux de capteurs

10. Les caractéristiques d'un système d'IoT

Les caractéristiques fondamentales de l'IoT sont les suivantes [24] :

- **Inter-connectivité** : tout peut être interconnecté avec l'infrastructure globale d'information et de communication.
- **Services liés aux objets connectés** : L'IoT est capable de fournir des services liés aux objets dans les limites des contraintes, telles que la protection de la vie privée et la cohérence sémantique entre les choses physiques et leurs objets virtuels associés. Afin de fournir des services liés aux choses (i.e. objets) dans les contraintes sur les choses, les technologies dans le monde physique et le monde de l'information vont changer.
- **Hétérogénéité** : Les périphériques de l'IoT sont hétérogènes en fonction des plates-formes matérielles et des réseaux. Ils peuvent interagir avec d'autres appareils ou plates-formes de services via différents réseaux.
- **Changements dynamiques** : L'état des dispositifs change dynamiquement, par exemple dormir et se réveiller, être connecté et / ou déconnecté ainsi que le contexte des dispositifs, y compris l'emplacement et la vitesse. De plus, le nombre d'appareils peut changer de façon dynamique.
- **Énorme échelle** : Le nombre de périphériques qui doivent être gérés et qui communiquent entre eux sera d'au moins un ordre de grandeur supérieur à celui des périphériques connectés à Internet. Encore plus critique sera la gestion des données générées et leur interprétation à des fins d'application. Cela concerne la sémantique des données, ainsi que la gestion efficace des données.
- **Sécurité** : Comme nous gagnons des avantages de l'IoT, nous ne devons pas oublier la sécurité. En tant que créateurs et destinataires de l'IoT, nous devons concevoir des mécanismes assurant la sécurité. Cela inclut la sécurité de nos données personnelles et la sécurité de notre bien-être physique. La sécurisation des points de terminaison, des réseaux et des données qui les traversent signifie la création d'un paradigme de sécurité qui évoluera.
- **Connectivité** : La connectivité permet l'accessibilité et la compatibilité du réseau. L'accessibilité se met sur un réseau alors que la compatibilité fournit la capacité commune de consommer et de produire des données.

11. Conclusion

Jusqu'ici, nous avons présenté une vision générale de l'IoT, la définition, les technologies utilisées et les domaines d'application. De ce qu'on a vu, on peut dire que l'IoT est conçue pour offrir une meilleure qualité de vie par l'automatisation des gestes quotidiens en fonction des besoins et des attentes de l'utilisateur final.

CHAPITRE 2 :

Composants d'un objet connecté

1. Introduction

Pour rendre les objets physiques connectés à Internet, nous avons besoin d'un ensemble de matériel et d'étages qui collecteront les informations et l'état de chacun et les enverront pour traitement. Dans cette unité, nous proposons un ensemble de matériel comprenant le type Arduino, des capteurs et des étages adaptés au processus.

Microcontrôleur

Cartes Arduino

Les capteurs

Les cartes d'essai

Les câbles

LED'S

Les résistances

Les Plateformes D'IoT

Les réseaux IoT

2. Microcontrôleur

Un microcontrôleur peut être vu comme un ordinateur situé dans un circuit intégré unique qui est dédié à l'exécution d'une tâche ou à l'exécution d'une application spécifique.

Il contient les éléments suivants : de la mémoire, des périphériques d'entrée / sortie programmables ainsi qu'un processeur. Les microcontrôleurs sont principalement conçus pour des applications embarquées et sont fortement utilisés dans les appareils électroniques à commande automatique tels que les téléphones portables, les appareils photo, les fours à microondes, les machines à laver, etc. [25]



Figure 2.1 - Microcontrôleur

Le microcontrôleur est composé principalement de **quatre parties [25]** :

Un microprocesseur qui va prendre en charge la partie traitement des informations et envoyer des ordres. Il est lui-même composé d'une unité arithmétique et logique (UAL) et d'un bus de données. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur

Une mémoire de données (RAM ou EEPROM) dans laquelle seront entreposées les données temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.

Une mémoire programmable (ROM), qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même. Il existe différents types de mémoires programmables que l'on utilisera selon l'application.

Notamment :

- **OTPROM** : programmable une seule fois mais ne coûte pas très cher.
- **UVPRM** : on peut la ré-effacer plusieurs fois grâce aux ultraviolets.
- **EEPROM** : on peut la ré-effacer plusieurs fois de façon électrique comme les mémoires flash.

La dernière partie correspond aux ressources auxiliaires. Celles-ci sont généralement :

- Ports d'entrées / sorties parallèle et série.
- Des Timers pour générer ou mesurer des signaux avec une grande précision temporelle.
- Des convertisseurs A/N pour traiter les signaux analogiques.

3. Cartes Arduino

C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur, et un logiciel, incontestable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur [9]. Il existe plusieurs modèles d'Arduino parmi eux :

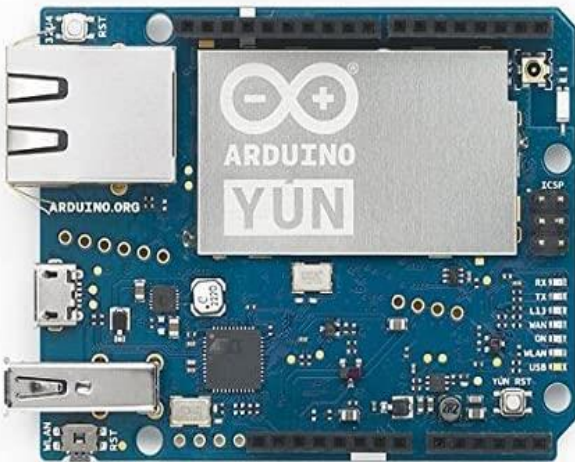
Carte Arduino UNO :



- Alimentation : via port USB ou 7 à 12 V sur Connecteur
- Microprocesseur : ATmega328
- Mémoire flash : 32 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 [26]

Figure 2.2 - Carte Arduino UNO

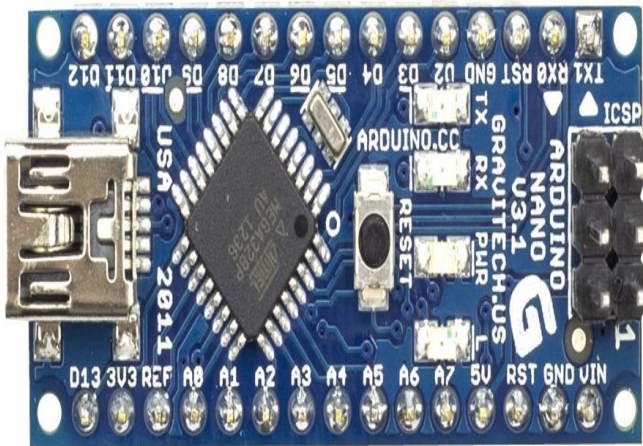
Arduino Yun



- mémoire RAM : 64 Mb DDR2
- mémoire flash : 32 kB
- 20 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 micro USB
- 1 port Ethernet, 1 port USB et 1 port micro-
- Dimensions : 72 x 53 x 17 mm [26]

Figure 2.3 - Carte Arduino Yun

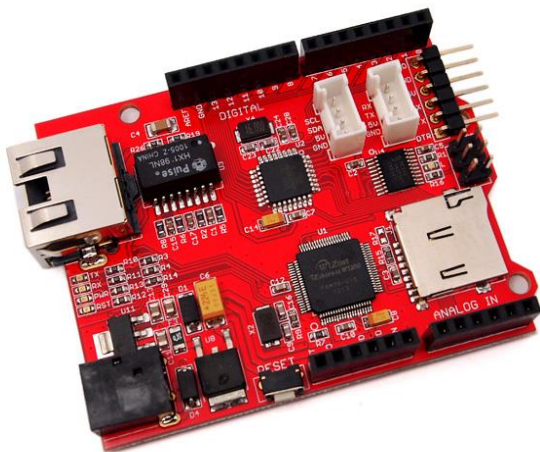
Arduino Nano



- alimentation : via port USB 6 à 20 V non régulée sur broche 30
- microprocesseur : ATmega328
- mémoire flash : 32 kB
- 14 broches d'E/S dont 6 PWM
- 8 entrées analogiques 10 bits
- intensité par E/S : 40 mA
- cadencement : 16 MHz
- gestion des interruptions
- fiche USB : mini-USB B
- dimensions : 45 x 18 x 18 mm [26].

Figure 2.4 - Arduino NANO

Arduino Seeduino



- Cette carte présente l'avantage de s'utiliser directement avec les capteurs, ce qui permet de réaliser vos prototypes sans soudure.
- alimentation : port USB
 - microprocesseur : ATmega328P
 - mémoire flash : 32 kB
 - 14 broches d'E/S dont 6 PWM
 - 6 entrées analogiques 10 bits
 - cadencement : 16 MHz
 - bus série, I2C et SPI
 - fiche micro-USB
 - dimensions : 70 x 54 x 13 mm [26]

Figure 2.5 - Carte Seeduino

Arduino Leonardo



- Microcontrôleur : ATmega32u4.
- Operating Voltage : 5V.
- Input Voltage (Recommended) : 7-12V.
- Input Voltage (limites) : 6-20V.
- Digital I/O Pins : 20.
- PWM Channel : 7.
- Analog Input Channels: 12.
- DC Current per I/O Pin: 40 mA.
- DC Current for 3.3V Pin: 50 mA.

Figure 2.6 -Arduino Leonardo

Netduino

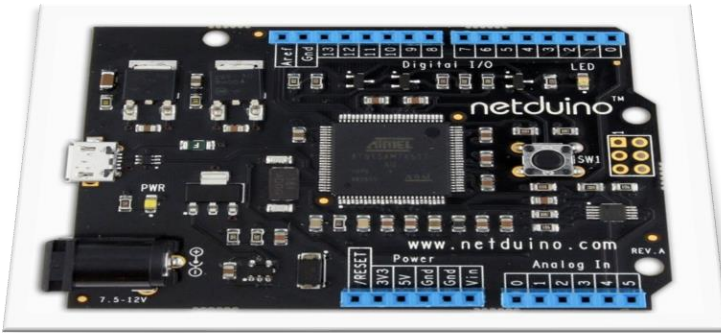


Figure 2.7 - Netduino

Carte de développement dont la puissance de calcul globale est supérieure à celle de l'Arduino. Il existe plusieurs modèles disponibles, y compris Netduino 2[26] :

- Vitesse : 120MHz, Cortex-M3 32 bits
- Mémoire Flash : 192 KB
- RAM : 60 Ko
- I_{max} : 25 mA par broche (Résistance Max)
- Alimentation : 3.3V mais 5V support

Carte Arduino MEGA



Figure 2.8 - Carte Arduino MEGA

- alimentation : via port USB ou 7 à 12 V sur connecteur alim
- microprocesseur : ATmega2560
- mémoire flash : 256 kB
- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- intensité par E/S : 40 mA
- cadencement : 16 MHz
- gestion des interruptions
- fiche USB B
- dimensions : 107 x 53 x 15 mm [26]

4. Module de connectivité

Deux modes de connectivité sont distingués :

- **connectivité sans fil** : Un réseau sans fil permet à moins deux terminaux de communiquer entre eux sans aucune liaison filaire, mais uniquement grâce aux signaux radioélectriques. Exemples: Wi-Fi, Bluetooth, Infrarouge, GSM/3G.
- **connectivité filaire** : Le réseau filaire est un réseau qui comme son nom l'indique est un réseau que l'on utilise grâce à une connexion avec fil. Ce réseau utilise des câbles Ethernet pour relier des ordinateurs et des périphériques grâce à un routeur ou à un commutateur.

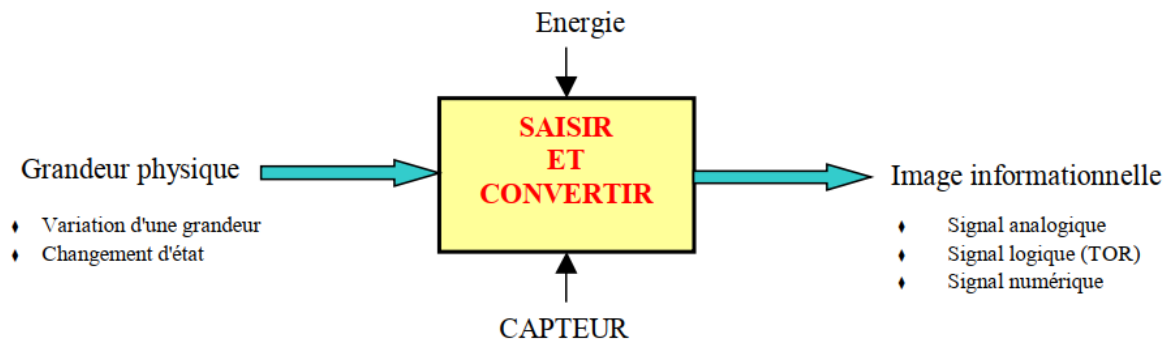
Exemples : L'Ethernet.. 3.

5. Les capteurs

5.1 Définition d'un capteur

Un capteur est un organe de prélèvement d'informations qui élabore, à partir d'une grandeur physique, une autre grandeur physique de nature différente (généralement électrique) représentative de la grandeur prélevée, et utilisable à des fins de mesure

Représentation fonctionnelle :

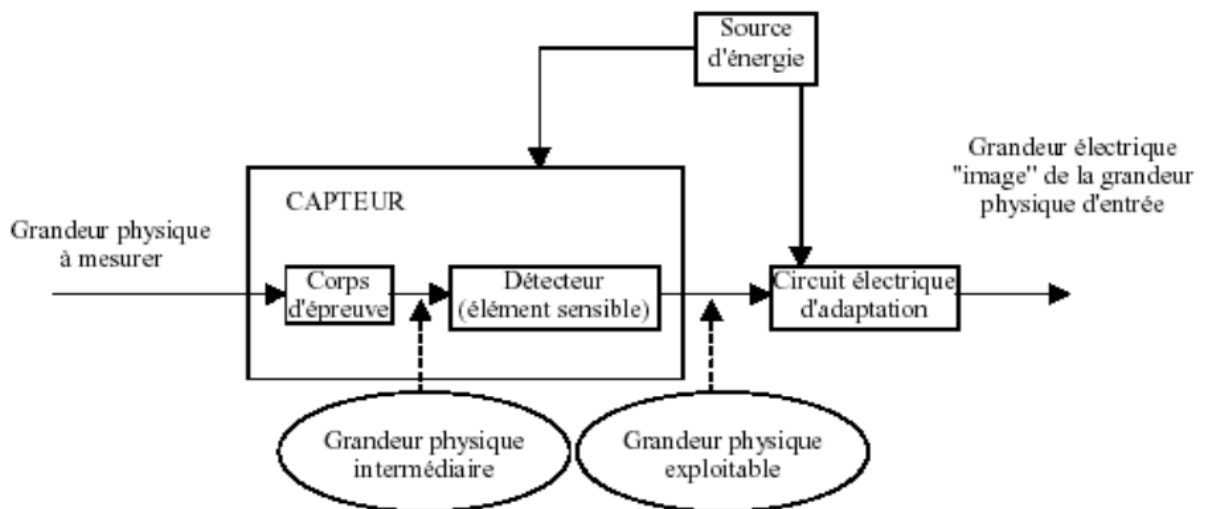


Rôle du capteur :

Parmi les informations de toutes natures issues de notre environnement, on distingue les grandeurs physiques associées à des événements climatiques, géométriques ou encore lumineux ou temporels. Le rôle du capteur est de rendre exploitable ces différentes grandeurs physiques en vue de leur traitement ultérieur

- Mesure de présence : indique la présence d'un "objet" à proximité immédiate ;
- Mesure de position, de déplacement ou de niveau : indique la position courante d'un objet animé d'un mouvement de rotation ou de translation ;
- Mesure de vitesse : indique la vitesse linéaire ou angulaire d'un "objet" ;
- Mesure d'accélération, de vibrations ou de chocs ;
- Mesure de débit, de force, de couples, de pressions ;
- Mesure de température, d'humidité

Éléments constitutifs d'un capteur :

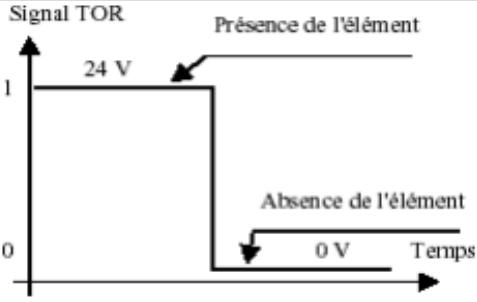
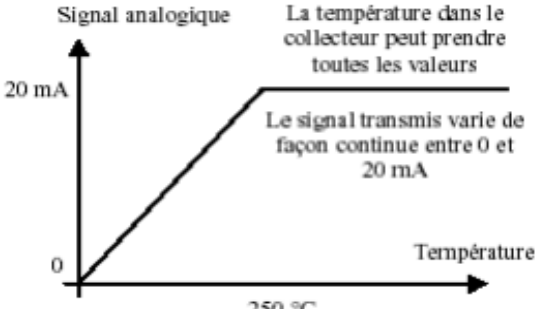
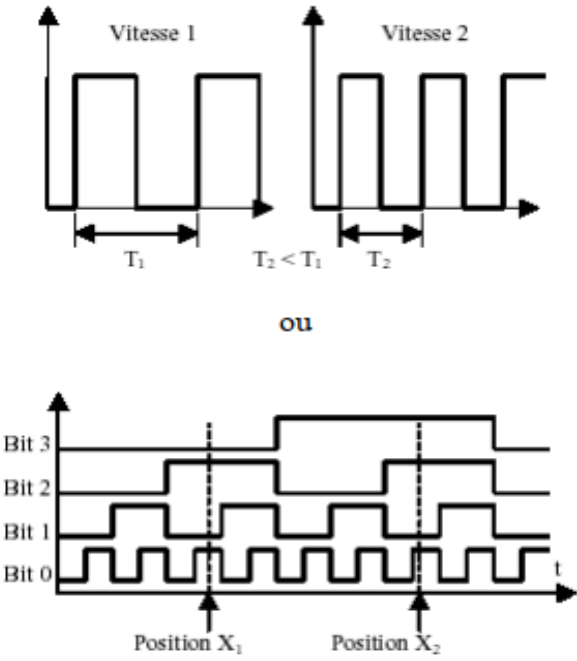


5.2 Classification des capteurs

La grandeur de sortie du capteur peut varier :

- De manière binaire (information vraie ou fausse), c'est le **capteur Tout Ou Rien (TOR)** ;
- De façon progressive (variation continue), c'est le **capteur analogique** ;
- D'échelon de tension ou de courant, c'est le **capteur numérique**.

Exemples :

<p>Capteur TOR</p>	
<p>Capteur Analogique</p>	
<p>Capteur Numérique</p>	

5.3 Caractéristiques des capteurs

Les capteurs sont distingués par les caractéristiques suivantes [41] :

- **Étendue de mesure (Plaine Echelle)** : L'étendue de mesure est la différence entre la limite supérieure et la limite inférieure de la grandeur mesurable par un capteur.

Lorsque le capteur

fournit une valeur de la grandeur entre 0 et le maximum, ce maximum est appelé « Plaine Echelle ».

- **Résolution** : plus petite variation de grandeur mesurable par le capteur.
- **Rapidité** : temps de réaction du capteur. La rapidité est liée à la bande passante.
- **Précision** : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- **Sensibilité** : représente la variation du signal de sortie par rapport à la variation du signal d'entrée.
- **Linéarité** : représente l'écart de sensibilité sur l'étendue de mesure.
- **Finesse** : Elle permet d'estimer l'influence que peut avoir le capteur et de son support ou de ses liaisons sur la grandeur à mesurer. Par exemple, dans le cas d'un capteur de température, une capacité calorifique importante réduit sa finesse.
- **Fréquence de résonance** : Un capteur possède une réponse qui peut dépendre de la fréquence de la grandeur mesurée. Lorsqu'il existe une fréquence à laquelle la réponse est particulièrement élevée, celle-ci est appelée fréquence de résonance.

5.4 Exemples de capteurs

Il existe plusieurs modèles type de capteur pour chaque modèle. Nous présentons les exemples suivants :

Capteur de mouvements PIR :



Figure 2.9 - Capteur PIR

Ce module est un détecteur de mouvement basé sur un capteur à infrarouges passifs.

- Alimentation : 5 à 9 Vcc
- Consommation : < 55uA
- Angle de détection : < 100°
- Distance de détection : 7 mètres
- Température -30°C à 80°C
- Dimensions : 28 x 13 x 13 mm
- Prix : 4.3€ [27].

Capteur de distance ULTRASON HC-SR04



Figure 2.10 - Capteur ultrason hc-sr04

- Plage de détection : 2cm à 4m
- Angle de détection idéal : 15°
- Alimentation : 5V
- Consommation : 15mA [27]

Capteur d'Humidité DHT11

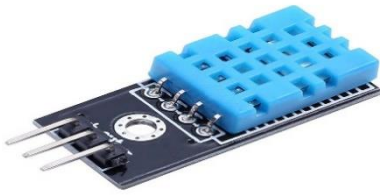


Figure 2.15 - Capteur DHT 1

- Alimentation : 5V
- Consommation : 0.5 mA en nominal / 2.5 mA maximum
- Etendue de mesure température : 0°C à 50°C ± 2°C
- Etendue de mesure humidité : 20-90%RH ±5%RH [27]

Capteur de Température et humidité Pro - Grove

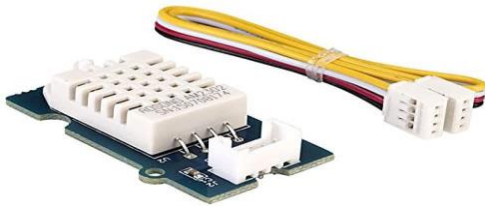


Figure 2.16 - Capteur pro-Grove

- Compatible de l'interface Grove
- Plage de Température : 0°C à 50°C ± 0.3°C
- Plage de d'humidité : 20% à 90% RH ± 2% RH[27]

Capteur de position GPS Ada fruit Ultimat GPS Shield

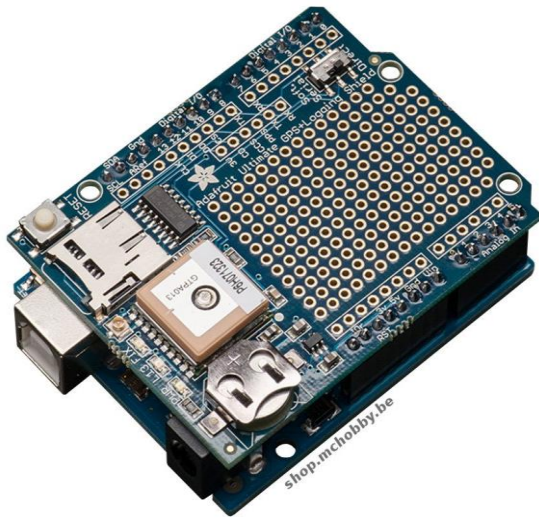


Figure 2.17 - Capteur GPS Shield

- Sensibilité de -165 DBm, mises à jour à 10 Hz, 66 canaux
- Faible puissance, consommation à 20 mA
- Compatible Arduino UNO, Leonard, Durmilanove
- Slot μSD pour stockage de données
- Module RTC pour datation avec batterie incluse (jusqu'à 7 ans de durée)
- LED pour signaler le stockage de données
- Sortie PPS
- Antenne interne et connecteur u.FL pour antenne externe
- Zone de prototypage [27]

6. Les cartes d'essai

Une plaque d'essai est d'une très grande utilité pour réaliser des montages électroniques sans soudure, en association notamment avec une carte à microcontrôleur telle que les cartes Arduino. Il est donc capital de bien savoir les utiliser et de comprendre les principes d'utilisation. Rien de très compliqué, mais il faut bien comprendre. La plaque d'essai s'utilise avec des straps, bouts de fils en cuivre monobrin (c'est important !!) de taille et de longueurs différentes. Les extrémités des straps doivent dénudés sur 1 cm environ [28].

Breadboard 400

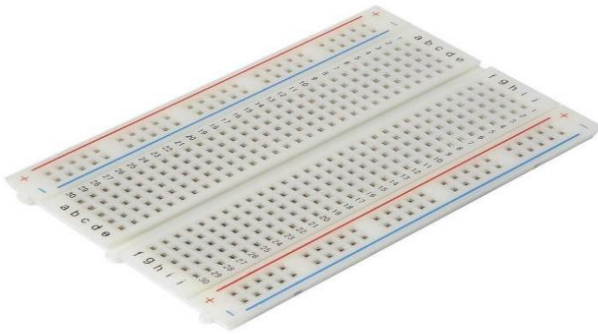


Figure 2.18 - Breadboard 400

Une **platine Labdec** (appelé en anglais **breadboard**, solderless breadboard, protoboard ou plug-board) est un dispositif qui permet de réaliser le **prototype d'un circuit électronique** et de le **tester**. L'avantage de ce système est d'être totalement réutilisable, car il ne nécessite **pas de soudure**. [27]

Adafruit Perma-Proto Half-sized Breadboard PCB – Single

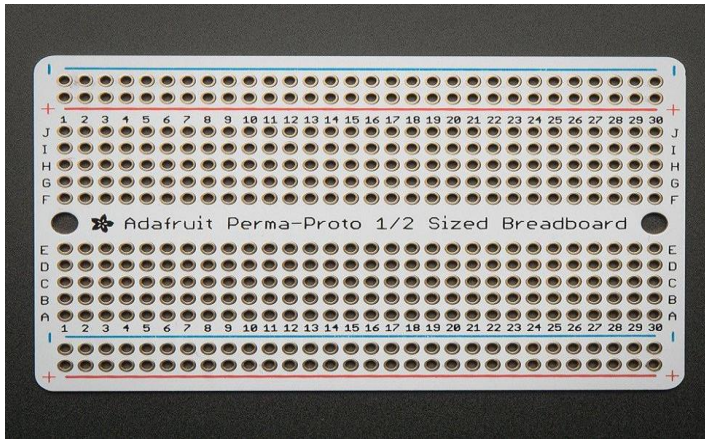


Figure 2.19 - Breadboard PCB – Single

Cette plaque à trou Perma-Proto sur le modèle de la breadboard 400 points est idéale pour reproduire vos projets testés sur une breadboard car les liaisons sont identiques. Il suffit donc de reproduire votre montage sur la plaque à trou Perma-Proto et de souder les composants dessus pour reproduire votre montage à l'identique [27].

7. Les câbles

Un câble regroupe généralement plusieurs fils conducteurs au sein d'une même gaine qui va les protéger, ces câbles sont utilisés pour relier les différents composants et peut être utilisé pour le transport d'énergie électrique mais aussi pour la transmission de données.

Câble alimentation Arduino



Figure 2.20 - Câble alimentation Arduino pour piles 9V

Ce câble permet d'alimenter simplement votre kit Arduino (Duemilanove, UNO ou encore MEGA 2560) à l'aide d'une pile 9V type 6LR61. Le câble dispose d'un connecteur type clip pile d'un côté et d'une fiche Power jack 2.1mm. Idéal pour donner toute l'autonomie énergétique à votre montage à moindre frais.

Câble USB vers fiche DC JACK



Figure 2.21 - Câble USB vers fiche DC JACK 2.1mm - 1m

Il existe deux standards pour alimenter une carte en 5V, L'USB et la fiche en DC Jack 2.1mm. Cet adaptateur est donc idéal pour alimenter une carte avec un connecteur DC Jack 2.1mm à partir d'une prise USB en 5V DC.

Le câble mesure 1m, il est en 22 AWG et il dispose d'une fiche USB A d'un côté et d'un connecteur DC Jack 2.1mm de l'autre avec le +5V DC relié au centre et la masse à l'extérieur.

Caractéristiques :

- ✓ Poids : 23g
- ✓ Longueur : 3.3 feet / 1m
- ✓ Taille de câble : 22AWG

Câble antenne d'adaptation RP



Figure 2.22 - Câble antenne d'adaptation RP SMA vers uFL--u.FL-IPX-IPEX-RF

Câble RF pigtail adaptateur U.FL/UFL vers RP SMA Femelle (Reverse SMA - **SMA inversé**) avec câble coaxial RG178, impédance 50 ohms, 15cm de longueur.

Le connecteur U.FL est aussi appelé UFL, IPEX, IPAX, IPX, MHF, AMC et UMCC.

Ce câble U.FL / UFL est notamment compatible avec le matériel WIFI et la connectique SMA inversé, **RP-SMA**.

Câble USB



Figure 2.23 - Câble USB

Ces câbles USB Sont nécessaire pour connecter facilement en USB les cartes Arduino. Il existe plusieurs types de câble selon le matérielle que nous avons utilisé. Par exemple USB « 2.0 câble type A/B ». Utiliser pour connecter Genuino Uno, Genuino Mega 2560, Genuino 101 ou toute carte avec le port USB femelle A de votre ordinateur. La longueur du câble est d'environ 178cm [27].

8. LED'S

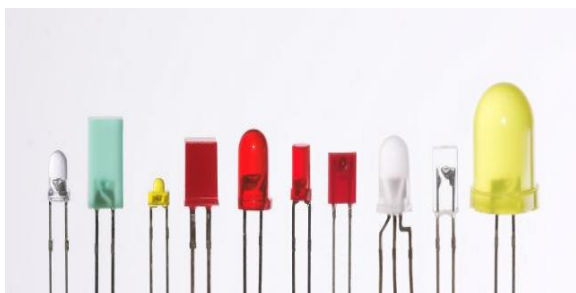


Figure 2.24 - Les LED

Une LED (en français : DEL : diode électroluminescente) est un composant électronique et optique, qui en étant traversé par du courant électrique, émet une lumière d'une intensité diffuse. Les LED consomment peu d'électricité [30]

9. Les résistances

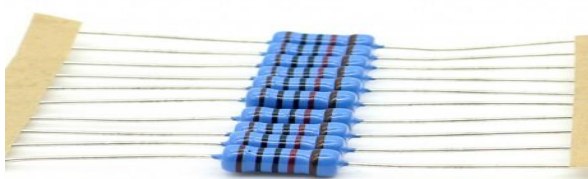


Figure 2.25 - Une résistance

La résistance électrique traduit la propriété d'un composant à s'opposer au passage d'un courant électrique (l'une des causes de perte en ligne d'électricité). Elle est souvent désignée par la lettre R et son unité de mesure est l'ohm. Elle est liée aux notions de résistivité et de conductivité électrique. On utilise une résistance de 560 Ω pour la sécurité des matérielle électronique [27]

10. Les Plateformes D'IoT :

Une plateforme IoT permet de faciliter la connexion de toutes sortes d'objet au réseau ainsi que le développement d'applications pour contrôler et gérer ces objets. Toute la complexité de la connectivité est tâche confiée à plate-forme IoT.

La définition d'une plateforme IoT varie en fonction des éditeurs, une plateforme IOT permet de :

- Connecter les objets à un SI.
- Authentifier et sécuriser les objets connectés.
- Gérer les données et les commandes à envoyer et à recevoir.
- Collecter, visualiser et analyser des données.
- Intégration avec d'autres services Web.

La plate-forme IoT garantit une intégration transparente avec différents matériels en utilisant une gamme de protocoles de communication populaires, en appliquant différents types de topologie (connexion directe ou passerelle) et en utilisant des SDK (Software Development Kit) si nécessaire.

Grâce aux interfaces d'intégration fournies par la plate-forme, on peut également alimenter les données IoT collectées dans des systèmes spécifiques de visualisation /stockage de données analytiques / données et transmettre des données à des périphériques connectés (configuration, notifications) ou entre eux (contrôles, événements) en utilisant différents types d'applications utilisateur.

Une plate-forme IoT est aussi souvent appelée middleware IoT, ce qui souligne son rôle fonctionnel comme celui d'un médiateur entre le matériel et les couches applicatives.

Les meilleures plates-formes IoT peuvent être intégrées à presque tous les périphériques connectés et se fondre dans les applications utilisées par l'appareil [31].

10.1 La plate-forme AWS IoT (Amazon Web Service)

AWS IoT est une plate-forme qui permet de connecter les appareils aux services AWS et à d'autres appareils. Elle sécurise les données et interactions, traite les données transmises par les différents appareils et déclenche des actions en conséquence. Nous distinguons les fonctionnalités suivantes :

- **Kit SDK pour les appareils AWS IoT** : Pour aider les équipements matériels ou les applications mobiles à connecter facilement et rapidement. Le kit SDK pour les appareils AWS IoT permet la connexion et l'authentification des appareils, ainsi que l'échange de messages avec AWS IoT, via les protocoles MQTT, Web Sockets ou HTTP. Ce kit SDK prend en charge les langages C, JavaScript et Arduino.
- **Passerelle AWS IoT** : La passerelle AWS IoT permet aux appareils de se connecter en toute sécurité et de communiquer efficacement avec AWS IoT. La passerelle permet l'échange de messages selon un modèle publication/souscription qui permet des communications de type 1 à 1 ou 1 à plusieurs. Dans le modèle AWS IoT de communication 1 à plusieurs, un appareil connecté peut diffuser des données à plusieurs abonnés sur un sujet donné.
- **Authentification et autorisation** : AWS IoT assure l'authentification mutuelle et le chiffrement sur tous les points de connexion, si bien qu'aucun échange de données ne se produit entre les appareils et AWS IoT sans que les identités n'aient été vérifiées. AWS IoT prend en charge la méthode AWS d'authentification (appelée « SigV4 ») ainsi que l'authentification basée sur le certificat X.509.
- **Registre** : Un registre identifie chaque appareil et permet d'assurer le suivi de ses métadonnées. Le registre affecte une identité unique à chaque appareil, selon un format cohérent, quel que soit le type d'appareil et le mode de connexion utilisés. Il intègre également des métadonnées qui décrivent les caractéristiques de l'appareil. Le registre stocke les métadonnées des appareils sans frais supplémentaires et pour une durée illimitée tant que la consultation ou mettez à jour le registre au moins une fois tous les 7 ans.
- **Versions Shadow** : Avec AWS IoT, Avoir la possibilité de créer une version virtuelle persistante, ou version « shadow », qui correspond au dernier état de chaque appareil et permet aux applications et aux autres équipements de lire les messages de l'appareil et d'interagir avec ce dernier. Les versions Shadow conservent le dernier état transmis et celui souhaité pour chaque appareil, et ce, même si l'appareil est déconnecté. La version Shadow vous permet de conserver l'état de chacun de vos appareils pendant un an gratuitement.
- **Moteur de règles** : Le moteur de règles permet de concevoir des applications AWS IoT qui collectent, traitent et analysent les données transmises par les appareils connectés afin de déclencher des actions. Le moteur de règles évalue les messages entrants publiés dans AWS IoT et les convertit afin de les transmettre à un autre appareil ou à un service de Cloud, selon les règles métier qui est définies. Une même règle peut déclencher une ou plusieurs actions en parallèle [32].

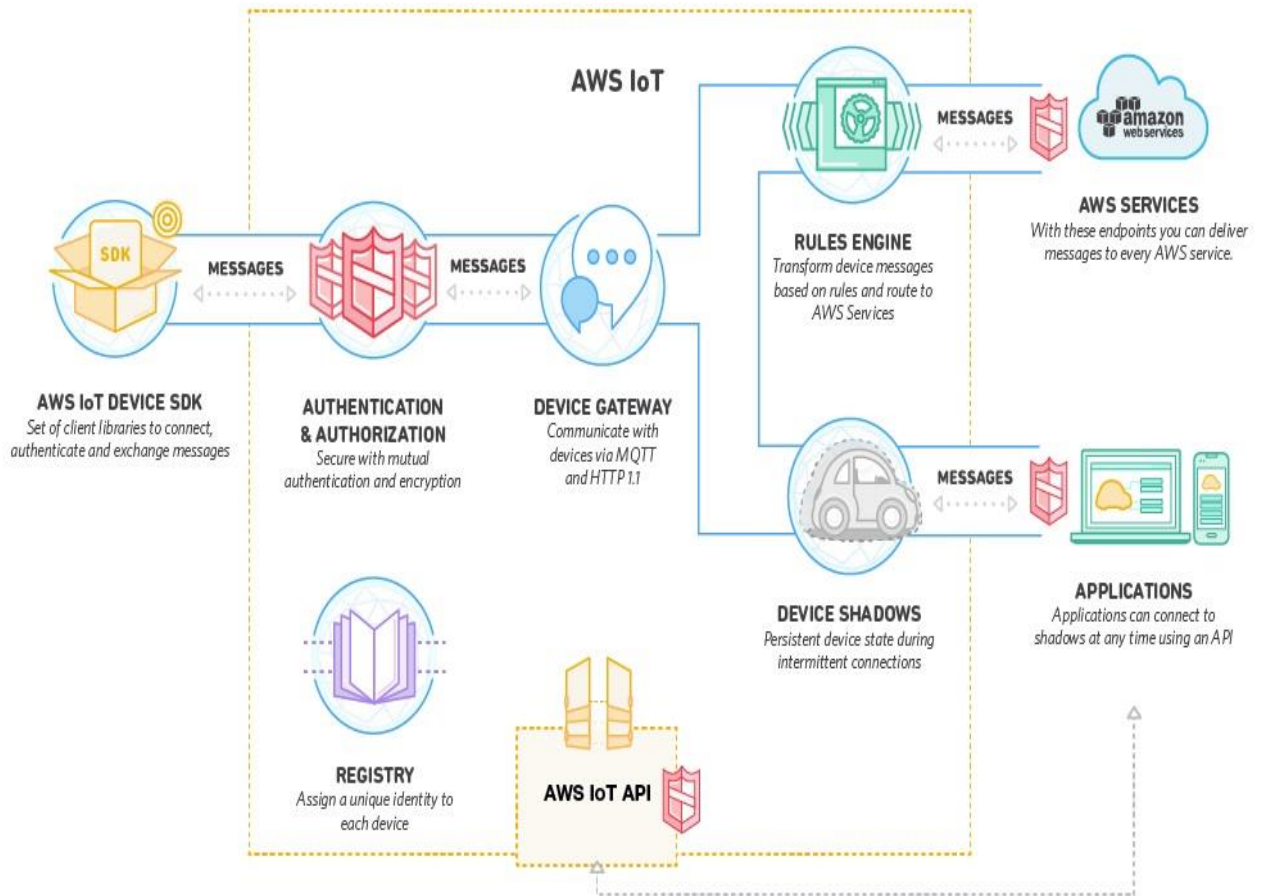


Figure 2.26 - Architecture AWS IoT

10.2 Google Cloud Platform

Cette infrastructure permet de gérer les flux de données alimentés par des millions d'appareils intelligents dans Internet of Things. L'architecture de ce type de traitement de flux en temps réel doit traiter de l'importation, du traitement, du stockage et de l'analyse de données de centaines de millions d'événements par heure.

➤ **Devices** : Les périphériques sont des dispositifs physiques qui interagissent avec le monde et recueillent des données. En général, ils peuvent être considérés en deux groupes : les dispositifs contraints et standard.

1. **Les périphériques contraints** : peuvent être très petits et avoir très peu de ressources en termes de calcul, de stockage, ... etc. Ils peuvent communiquer uniquement via des réseaux qui ne peuvent pas accéder directement à la plate-forme, par exemple via Bluetooth Low Energy (BLE).

2. **Les appareils standards** : ressemblent plus à de petits ordinateurs. Ils peuvent acheminer des données directement sur des réseaux vers la Plateforme passé par une certaine forme de périphérique de passerelle.

➤ **Ingestion (Capacité) :**

1. Cloud Pub/Sub : est un système de messagerie qui peut agir comme un absorbeur de d'objets, à la fois pour les flux de données entrantes ainsi que les modifications apportées à l'architecture de l'application. Même les périphériques standards peuvent avoir une capacité limitée à stocker et à réessayer d'envoyer des données de télémétrie.

2. Stackdriver Monitoring and Stackdriver Logging : Stackdriver Monitoring fournit des statistiques de séries chronologiques sur les principaux indicateurs de santé et peut vous alerter dès que des problèmes surviennent dans la flotte de périphériques. Stackdriver Logging peut collecter des données de démarrage et d'exécution sur les applications exécutées sur l'appareil et peut servir de source clé pour l'analyse interne des applications.

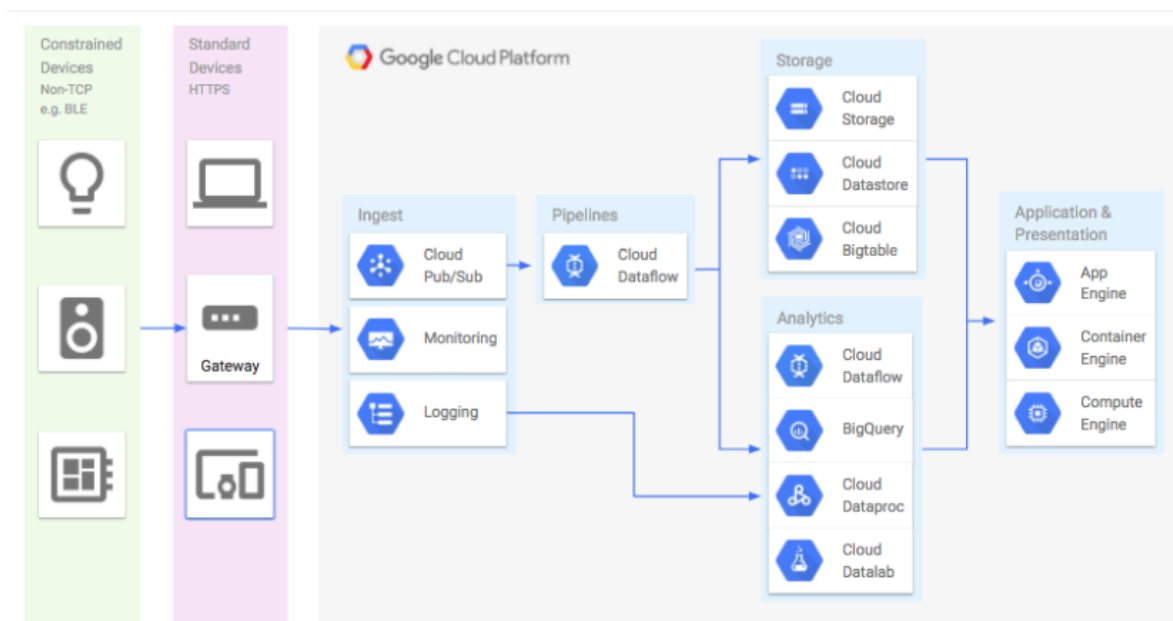


Figure 2.27 - Architecture Google Cloud plateforme

➤ **Pipelines :** gère les données après son arrivée sur la Plateforme, de la même façon que les pièces sont gérées sur une ligne d'usine. Cela inclut des tâches telles que :

1. Transformation des données : convertir les données dans un autre format, par exemple, convertir une tension de signal de périphérique capturée en une unité étalonnée de mesure de la température.

2. Agrégation et calcul des données : En combinant des données, vous pouvez ajouter des contrôles, tels que la moyenne des données sur plusieurs périphériques afin d'éviter d'agir sur un seul périphérique, ou de vous assurer que la dispose de données exploitables si un seul périphérique est hors ligne.

3. Enrichissement des données : combiner les données générées par l'appareil avec d'autres métadonnées sur le périphérique ou avec d'autres ensembles de données, comme les données météorologiques ou de trafic, pour une analyse ultérieure.

4. Déplacement de données : stocker les données traitées dans un ou plusieurs lieux de stockage final.

5. Cloud Dataflow : est conçu pour exécuter toutes ces tâches de pipeline sur les données de lot et de flux.

➤ **Storage** : les données du monde physique se présentent sous diverses formes et tailles. La Plateforme offre une gamme de solutions de stockage, depuis des blocs de données non structurés avec Google Cloud Storage, tels que des images ou des flux vidéo de caméras connectées, le stockage structuré d'entités avec Google Cloud Datastore et des bases de données chronologiques de haute performance avec Google Cloud Bigtable.

➤ **Analytics** : est l'endroit où vous extrayez la valeur de l'information à partir des données brutes ou traitées. Bien que puisse effectuer certaines analyses pendant la diffusion dans le pipeline Cloud Dataflow, la majeure partie du traitement analytique concerne les données accumulées dans différents systèmes de stockage. Souvent, la valeur des analyses IoT provient de la combinaison de données du monde physique avec des données provenant d'autres sources, telles que des données de relation client ou des systèmes d'information en ligne.

1. BigQuery : fournit un entrepôt de données entièrement géré avec une interface familière, similaire à SQL.

2. Cloud Dataflow : Cela signifie que vous pouvez appliquer une nouvelle logique de traitement à de nouvelles données dans le pipeline et également réappliquer la nouvelle logique aux données historiques maintenant stockées.

3. Cloud Datalab : fournit une Table de travail de données interactif pour explorer les ensembles de données avec une approche de bloc-notes qui vous permet de combiner code, commentaire et graphiques.

4. Cloud Dataproc : est une solution de gérée l'exécution de Hadoop et Spark, que vous pouvez utiliser pour exploiter l'étendue et la portée de ces écosystèmes.

➤ **Applications and presentation** : la réalisation de la valeur des données nécessite une action. Cette action peut être une réponse automatisée par une application longue durée ou la présentation de données à un décideur humain. Pouvoir utiliser plusieurs services variés pour héberger des applications sur la plateforme, y compris Google Container Engine pour les applications à base de conteneurs, les environnements standards et flexibles Google App Engine en tant que plateforme gérées pour les applications évolutives et Google Compute Engine, offrant des VM à haute performance pour un maximum la flexibilité [33].

10.3 IBM Cloud Platform

IBM Watson IoT Platform fournit aux applications un accès puissant aux terminaux et aux données IoT afin de vous aider à composer rapidement des applications d'analyse, des tableaux de bord de visualisation et des applications IoT mobiles.

Watson IoT Platform vous permet d'effectuer des opérations de gestion de terminaux puissantes, de stocker des données de terminal et d'y accéder, et de connecter une grande diversité de terminaux et de terminaux de passerelle. Watson IoT Platform fournit une communication sécurisée vers et depuis vos terminaux à l'aide de MQTT et de TLS.

Watson IoT Platform communique avec vos applications et vos terminaux à l'aide de l'API

Watson IoT Platform et du protocole de messagerie Watson IoT Platform. Le tableau de bord

Watson IoT Platform se connecte en tant qu'interface utilisateur initiale pour simplifier

les opérations au sein de la plateforme. Les données de terminal peuvent être stockées ou utilisées à l'aide de solutions d'analyse.

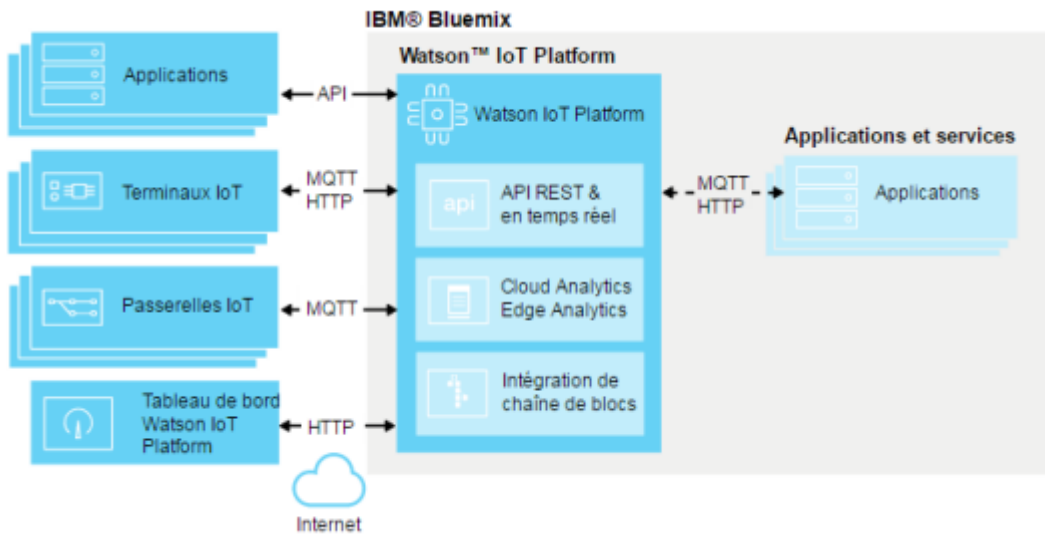


Figure 2.28 - architecture Ibm waston

➤ **Organisations** : Lorsque vous vous enregistrez auprès de Watson IoT Platform, vous recevez un ID d'organisation. Votre ID d'organisation est un identificateur unique composé de six caractères pour votre compte. Les organisations s'assurent que vos données ne sont accessibles que par vos terminaux et vos applications. Après l'enregistrement, les terminaux et les clés d'API sont liés à une organisation unique. Lorsqu'une application se connecte au service à l'aide d'une clé d'API, elle s'enregistre à l'organisation qui est associée à la clé d'API utilisée. Pour votre sécurité, la communication entre les organisations n'est pas possible. Le seul moyen de transmettre des données entre deux organisations consiste à créer une application au sein de chaque organisation qui communiquera avec les applications situées dans l'autre organisation.

➤ **Terminaux** : Un terminal peut être n'importe quel élément doté d'une connexion à Internet et qui peut insérer des données via une commande push dans le service Cloud. Toutefois, les terminaux ne peuvent pas communiquer directement avec d'autres terminaux, mais ils acceptent des commandes émises par des applications et envoient des événements à des applications. Les terminaux dans Watson IoT Platform sont identifiés par un jeton d'authentification unique. Les terminaux doivent être enregistrés avant de pouvoir se connecter à Watson IoT Platform.

➤ **Passerelles** : Les passerelles sont des terminaux spécialisés qui possèdent les fonctions combinées d'une application et d'un terminal, ce qui leur permet de servir de points d'accès pour d'autres terminaux. Les terminaux qui ne peuvent pas se connecter directement à Internet peuvent accéder au service Watson IoT Platform en se connectant d'abord au terminal de passerelle. Les passerelles doivent être enregistrées avant de pouvoir se connecter au service.

- **Applications** : Une application peut être n'importe quel élément doté d'une connexion à Internet et qui interagit avec les données des terminaux et contrôle le comportement de ces terminaux. Les applications s'identifient dans Watson IoT Platform à l'aide d'une clé d'API et d'un ID d'application unique. Contrairement aux terminaux, les applications individuelles n'ont pas besoin de s'enregistrer avant de pouvoir se connecter à Watson IoT Platform. Toutefois, elles doivent utiliser une clé d'API valide qui a été enregistrée.
- **Événements** : Les événements constituent le mécanisme par lequel les terminaux publient des données sur Watson IoT Platform. Les terminaux contrôlent le contenu de leurs messages et affectent un nom à chaque événement envoyé. Watson IoT Platform utilise les données d'identification qui sont associées à chaque événement reçu afin de déterminer le terminal qui l'a envoyé. Cette architecture empêche les terminaux de simuler les droits d'accès d'autres terminaux. Les applications peuvent traiter des événements en temps réel et voir la source de l'événement et les données contenues dans ce dernier. Les applications doivent être configurées pour définir les terminaux et les événements auxquels ils sont abonnés.
- **Commande** : Les commandes représentent le mécanisme qui permet aux applications de communiquer avec des terminaux. Seules les applications peuvent envoyer des commandes, et ces dernières sont envoyées à des terminaux spécifiques. Le terminal doit déterminer l'action à exécuter dès la réception d'une commande donnée. Les terminaux peuvent être conçus pour écouter n'importe quelle commande ou pour s'abonner à une liste de commandes spécifiée [34].

10.4 Comparaison des plateformes

D'abord, On va présenter une liste des principales caractéristiques qui sont importantes pour toute plate-forme logicielle IoT, ensuite la comparaison des mesures dans laquelle ces principales caractéristiques ont été mises en œuvre dans les plates-formes logicielles actuelles IoT.

- **Gestion d'appareils** : La gestion des périphériques est l'une des caractéristiques les plus importantes attendues de toute plateforme logicielle IoT. La plate-forme IoT doit tenir à jour une liste des périphériques qui lui sont connectés et suivre leur état de fonctionnement.

Il doit être en mesure de gérer les mises à jour de la configuration, du microprogramme et de fournir des rapports d'erreur au niveau du périphérique et de la gestion des erreurs. À la fin de la journée, les utilisateurs des périphériques devraient être en mesure d'obtenir des statistiques de niveau de périphérique individuel.

- **L'intégration** : Le soutien à l'intégration est une autre caractéristique importante attendue d'une plate-forme logicielle IoT. L'API devrait permettre d'accéder aux opérations importantes et aux données qui doivent être exposées à partir de la plateforme IoT. Il est courant d'utiliser les API REST pour atteindre cet objectif.
- Protocoles de collecte de données : Un autre aspect important qui nécessite une attention est les types de protocoles utilisés pour la communication de données entre les composants d'une plate-forme logicielle IoT. Une plate-forme IoT peut avoir besoin d'être mise à l'échelle à des millions ou même des milliards de périphériques. Des protocoles de communication légers devraient être utilisés pour permettre une faible consommation d'énergie ainsi qu'une faible fonctionnalité de bande passante réseau.

➤ **Analyse de données** : Les données collectées à partir des capteurs connectés à une plateforme IoT doivent être analysées de manière intelligente afin d'obtenir des informations significatives. Il existe quatre principaux types d'analyse qui peuvent être effectués sur les données IoT :

- Analyse en temps réel.
- Analyse en lots.
- Analyse prédictive.
- Analyse interactive.

Les analyses en temps réel effectuent une analyse en ligne (à la volée) des données en continu. Les opérations d'exemple comprennent les agrégations basées sur des fenêtres, le filtrage, la transformation et ainsi de suite.

L'analyse par lot exécute des opérations sur un ensemble de données accumulées. Ainsi, les opérations par lot se déroulent à des périodes de temps planifiées et peuvent durer plusieurs heures ou plusieurs jours.

L'analyse prédictive se concentre sur les prédictions basées sur diverses techniques statistiques et d'apprentissage automatique.

Les analyses interactives exécutent des analyses exploratoires multiples sur les données en flux et en lots.

➤ **visualisation de données** : Avec une visualisation rapide et facile, on peut voir les données en quelques secondes et déduire des informations significatives à partir des visuels.

La visualisation des données est un terme général qui décrit tout effort visant à aider les gens à comprendre l'importance des données en les plaçant dans un contexte visuel. Les modèles, les tendances et les corrélations qui pourraient passer inaperçus dans les données textuelles peuvent être exposés et reconnus plus facilement avec le logiciel de visualisation de données.

➤ **Tarifification** : Action d'établir le prix d'un service fourni par une plateforme IOT. Le client ne paie que ce qu'il utilise. Les tarifs sont calculés en fonction du nombre de messages publiés sur la plateforme IoT (coût de publication) et du nombre de messages envoyés par la plateforme IoT vers les appareils ou applications (coût d'envoi).

➤ **Sécurité de l'information** : Les mesures de sécurité de l'information requises pour exploiter une plate-forme logicielle IoT sont beaucoup plus élevées que les applications logicielles générales et les services. Des millions de dispositifs connectés à une plateforme IoT signifient que nous devons anticiper un nombre proportionnel de vulnérabilités. Généralement, la connexion réseau entre les périphériques IoT et la plateforme logicielle IoT devrait être cryptée avec un mécanisme de cryptage fort pour éviter les écoutes potentielles [24]

	IBM Platform Watson	Google Cloud Platform	AWS Platform
Gestion d'appareils	Oui	Oui	Oui
Protocoles de collecte de données	MQTT, HTTPS	Thread	MQTT, HTTP1.1, WebSockets
Analyse de données	Analyse en temps réel	Analyse en temps réel	Analyse en temps réel
Visualisation de données	Oui(portail Web)	Oui (tableau de bord Google IoT)	Oui (tableau de bord AWS IoT)
Tarifification	BM affiche trois tarifs : gratuit pour les start-up et les "amateurs", 24,15 \$ par mois pour les "moyennes" entreprises, 175,35 \$ pour les grande	/	Gratuit jusqu'à 250 000 messages par mois, pendant un an. Puis 5 \$ par million de messages
Sécurité de l'information	Link Encryption (TLS), Authentication (IBM Cloud SSO), Identity management (LDAP)	Link Encryption (TLS)	Link Encryption (TLS), Authentication (SigV4, X.509)

Tableau 2 - Comparaisons entre les Platform

11. Les réseaux IoT

Depuis les années 2010, la multiplication des objets connectés grand public et professionnels a entraîné la création ou la transformation de multiples réseaux dédiés : les réseaux attachés à la personne, « indoor » pour les bâtiments et « outdoor » à l'échelle des territoires, des pays ou même mondiaux [36].

11.1 Comment choisir un réseau ?

Les réseaux IoT se sont focalisés sur la performance énergétique pour augmenter :

- La durée de vie des objets sur batterie,
- La distance de communication pour diminuer les coûts d'infrastructure
- Le coût de service pour permettre de connecter plus d'objets.

Un réseau sera choisi en fonction de nombreux critères tels que : disponibilité, déploiement, maturité, espérance de vie, débit de données, distance et type de communication, objets fixes ou mobiles, localisation des objets, niveau de sécurité, type d'opérateur de réseau, durée de vie des batteries, disponibilité des modules, type de plateforme IoT, coût (OPEX/CAPEX), maintenance, interopérabilité....

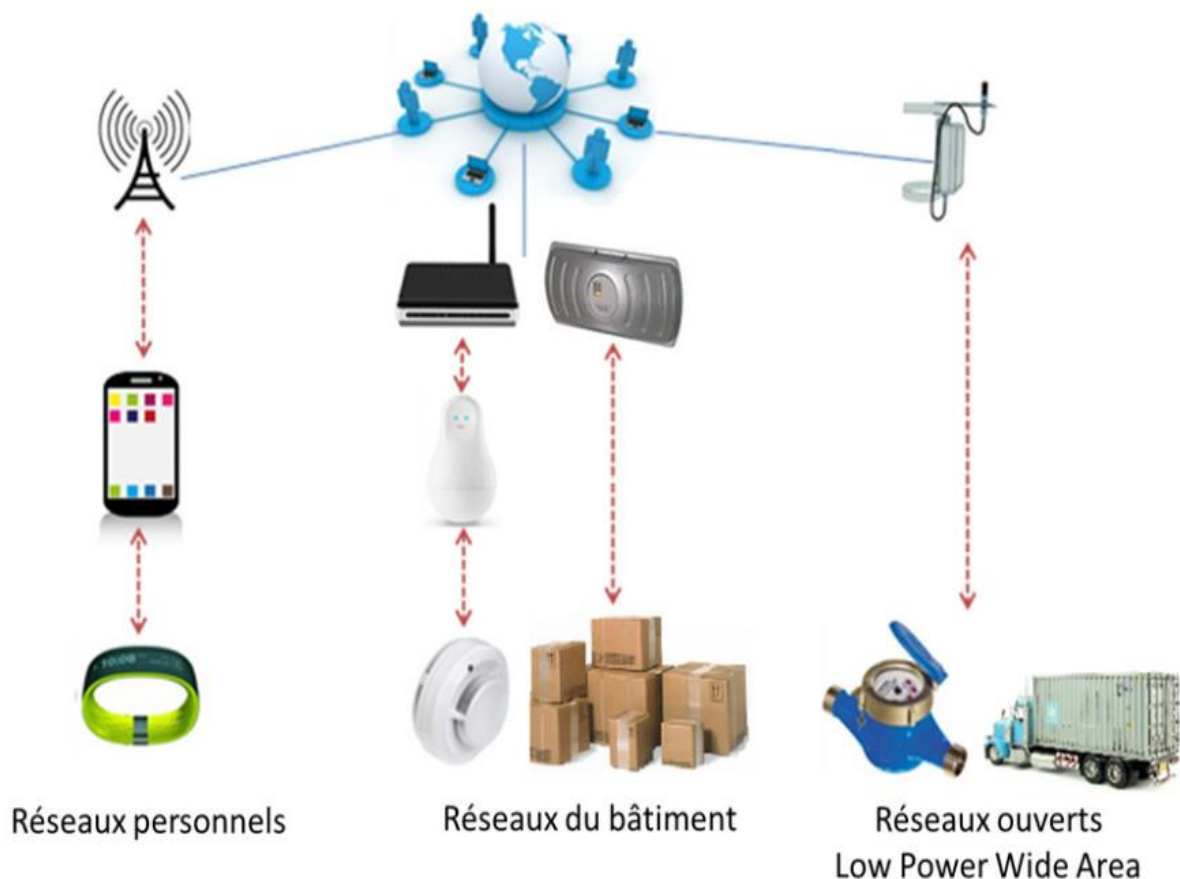


Figure 2.29 - Types réseaux sans fil

11.2 Les réseaux personnels

Les réseaux personnels Ce sont les réseaux grand public attachés à la personne, orientés autour du sport & bien-être, des loisirs, des médias, des réseaux sociaux et de l'efficacité personnelle. Ces réseaux sont donc très souvent liés aux smartphones ou à des dispositifs miniaturisés et portables. Les principaux réseaux sont :

Bluetooth (BLE, BLE Smart, etc...) qui permet une connectivité facile à quelques mètres en utilisant un protocole de communication très largement diffusé et disponible nativement sur de très nombreux appareils

ANT+ qui est un protocole propriétaire, avec de nombreux partenaires industriels et très fortement lié au sport et au bien-être [36].

11.3 Les réseaux du bâtiment

- ❖ BLE industriel, issu du Bluetooth grand public fiabilisé pour l'industrie, permet la création de réseaux denses dans les entrepôts ou le tertiaire pour suivre les biens et les capteurs, avec une distance de communication typique de 20 à 200m. Pouvant associer les étiquettes BLE avec des balises, des répéteurs et des routeurs, ces réseaux permettent de très nombreuses applications allant jusqu'à la géolocalisation à l'intérieur des bâtiments.
- ❖ RFID Active, comparable au BLE dans les milieux industriels contraints, permet de connecter et de géolocaliser des objets dans un bâtiment. Souvent complémentaire au BLE, la RFID active utilise un protocole non propriétaire libre de droit.
- ❖ Zigbee: réseau maillé du bâtiment, permettant de connecter les objets de proche en proche, basé sur un protocole propriétaire géré par la Zigbee Alliance, souvent utilisé en domotique pour remplacer le câblage électrique (lumières, capteurs de confort, capteurs d'alertes)
- ❖ Z-Wave: réseau maillé assez similaire à Zigbee, reposant aussi sur une alliance d'industriels
- ❖ ZWave Alliance EnOcean : réseau domotique propriétaire, ayant la caractéristique d'offrir certaines fonctions sans batteries avec capture de l'énergie (voir « energy harvesting » plus haut)
- ❖ Thread/ 6LoWPAN : réseau maillé du bâtiment, qui a pour vocation d'harmoniser la connexion des objets pour la maison connectée. C'est un protocole propriétaire basé sur la technologie 6LoWPAN (IPv6 Low power Wireless Personal Area Networks), et piloté par une alliance d'industriels emmenée par Google
- ❖ WiFi Hallow : autre réseau maillé pour la domotique, basé aussi sur de l'IPv6 et un protocole WiFi adapté, protocole propriétaire piloté par la WiFi Alliance
- ❖ Dash7 : réseau plus longue distance pour la domotique et l'automobile, permettant une meilleure pénétration dans les bâtiments à 433MHz et 868Mhz, protocole propriétaire piloté par la Dash7 Alliance.
- ❖ LoRa privé : réseaux cellulaire privé basé sur le protocole LoRaWAN, qui permet une communication longue distance en intérieur et extérieur de messages courts (remontée de capteurs ou localisation). L'intérêt de ce réseau est de permettre aux objets de sortir du mode « privé » en profitant des réseaux « publics » LoRaWAN avec un abonnement aux services d'opérateurs nationaux et internationaux [36].

11.4 Les réseaux LPWAN (Low Power Wide Area Network)

Ce sont les nouveaux réseaux (cellulaires ou non) dédiés aux objets connectés en extérieur. Ils sont idéaux pour remonter des informations simples de capteurs et de localisation vers des plateformes de gestion d'objets, sur le cloud ou sur des réseaux privés.

Ces réseaux sont basés sur des protocoles privilégiant :

- La très basse consommation (autonomie de 5 à 10 ans)
- La très longue distance de communication (plusieurs km)
- La bonne couverture à l'intérieur des bâtiments
- La très forte densité d'objets connectables Les faibles coûts d'opération

La contrepartie de ces avantages est le débit limité de communication et la limitation du nombre de communications bi-directionnelles (à partir de l'objet ou vers l'objet).

Les principaux réseaux LPWAN sont :

- Sigfox est le premier réseau cellulaire créé spécifiquement pour les objets connectés, la couverture est très forte en France et dans les pays limitrophes et se déploie très rapidement en Amérique du Nord et en Asie. C'est un réseau opéré par Sigfox ou ses partenaires en fonction des pays, et qui repose sur un protocole de communication UNB (Ultra Narrow Band) pour favoriser la portée et le nombre d'objets connectés sur une même antenne. Fin 2015, Sigfox annonçait que déjà plus de 7 millions d'objets étaient connectés à leur réseau et ce nombre ne cesse de croître.
- LoRa, réseau cellulaire concurrent de Sigfox dédié aussi aux objets connectés, qui a vu ses premiers déploiements massifs à partir de 2017. Il utilise un protocole de communication ouvert géré par la LoRa Alliance, qui permet aux utilisateurs de créer leur propre réseau, interne (ex : un site industriel voir LoRa privé dans la section ci-dessus) ou bien d'utiliser les différents réseaux nationaux et internationaux des opérateurs (Objenious de Bouygues Télécom et Orange en France).
- Wize EN13757-4/N
- 3GPP : LTE-M, NB-IOT, 5G
- Réseaux propriétaires ou marché : ingenu, RPMA, Wimax, Qowisio, Weightless [36].

12. Conclusion

Ce chapitre a fait l'objet d'une étude sur la technologie d'IoT. Nous avons présenté les différents composants. En particulier, nous avons étudié la communication et les plateformes de gestion des systèmes d'IoT qui facilitent, aux développeurs, la mise en œuvre de la communication, ainsi, elles permettent d'accélérer et de réduire le coût de développement des produits et des applications IoT.

CHAPITRE 3 :

Serres Intelligentes

1. Introduction

Selon l'Organisation des Nations Unies (ONU), le « principal facteur à l'origine de l'augmentation des besoins alimentaires ». Selon l'Organisation des Nations Unies pour l'alimentation et l'agriculture (FAO), la population mondiale devrait atteindre 9,6 milliards d'habitants d'ici 2050. Il sera donc difficile pour le secteur agricole de répondre aux besoins de la population vivant de l'alimentation.

Aujourd'hui, une autre difficulté que le secteur agricole doit faire face relative aux conditions météorologiques instables et le réchauffement de la planète qui a un impact négatif sur les cultures.

Les scientifiques cherchent des techniques et des moyens répondant aux besoins alimentaires suffisants et allant au-delà des menaces du changement climatique. L'agriculture intelligente utilise des technologies avancées telles que Big Data, l'Internet des objets, etc. Elle facilite l'automatisation de l'agriculture, la collecte des données sur le terrain et les analyses afin que les agriculteurs puissent prendre une décision précise en matière de production de cultures de haute qualité.

Dans ce chapitre, nous parlons des serres intelligentes.

2. Serres Intelligentes

2.1 Définition

La serre est un espace réduit qui peut être entre 1 m² Jusqu'à 100 m², elle est en général fermée ou partiellement ouverte, et destinée en général à la production agricole, ou une enceinte destinée à la et la protection des plantes en exploitant le rayonnement solaire elle est devenue un local industriel de production de la matière végétale ou l'on tente d'adapter l'environnement immédiat de la plante, de façon à améliorer sa productivité et sa qualité, en l'affranchissant du climat extérieur, du sol local et même des saisons [38].

Avantages :

- Production plus élevée grâce à la possibilité de contrôler les conditions climatiques de la culture et de favoriser la production à toutes les saisons, augmentation du rendement et de la qualité de la récolte,
- Précocité et retard de la production,



Figure 3.1 - Exemple d'une serre agricole

2.2 Les différents types de serres

2.2.1 Serres tunnel

La serre tunnel (Figure 3.2) est formée de beaucoup d'arches métalliques et bien plantée dans la terre, recouvertes d'un film souple en plastique (généralement blanc transparent ou jaune transparent) et c'est ce qui lui donne la forme d'un tunnel. Ce type est célèbre en Algérie.



Figure 3.2 - Une serre tunnel.

Caractéristique des serres tunnel :

- ✓ La hauteur varie généralement entre 2 et 2,50 mètres.
- ✓ La largeur se situe entre 3 et 5 mètres.
- ✓ Une surface au sol minimal de 6 m².
- ✓ Généralement existe deux portes.

2.2.2 Serres multi-chapelle :

Les serres Multi-chapelle (Figure 3.3) sont conçues pour être parfaitement assemblées grâce à des joints vissés qui simplifient le montage tout en absorbant idéalement les différentes forces qui ont un impact sur la structure. Elles sont très demandées pour leur robustesse et leur grande capacité d'adaptation aux dimensions et aux caractéristiques du terrain [40].



Figure 3.3 - Serres multi-chapelle

Caractéristique des serres multi-chapelle :

- 1) Matériaux utilisés dans la couverture :
 - film plastique.
 - matériaux semi
 - rigides.
 - plaque rigide.
- 2) Les éléments en aluminium ou fer :
 - Poteaux d'installation.
 - Arches en tôle d'aluminium.
- 3) Grande volume intérieur : Sa hauteur atteint 6 mètres et la superficie peut dépasser 100 m².
- 4) La nature de la structure permet la bonne exploitation du rayonnement solaire
- 5) Elles procurent une aération uniforme de l'installation Les différents modèles sont conçus pour résister à des conditions climatiques déterminées.
- 6) Aérations au faîtage ou latérales.

2.2.3 Serres en verre :

Les serres en verre (Figure 3.4) comme les serres multi-chapelle dans la structure la seule différence est dans la couverture où les plaques de verre sont utilisées dans ce type à la place du film plastique.



Figure 3.4 - Serres en verre [39].

2.3 Structure et matériaux pour la serre

Il en existe quatre types pour la structure [40] : le bois, le PVC, l'aluminium, et l'acier et le verre.

- ✓ **Le bois** : est le matériau le plus esthétique et le meilleur isolant thermique.
- ✓ **Le PVC** : est en outre un bon isolant, qui limite la condensation et permet des économies d'énergie, de plus, son entretien est aisé. Cependant il faut savoir que le PVC ternit avec le temps.
- ✓ **Le métal** (aluminium ou acier) : il n'est pas un très bon isolant mais il est utile pour la construction de très grandes serres car il est mécaniquement résistant.
- ✓ **Le verre** plus lourd, est néanmoins plus translucide et meilleur vecteur de luminosité

2.4 Conditions environnementales dans les serres

La progression des plantes dans les serres implique certaines conditions climatiques spéciales. Dans ce qui suit, on présente d'une manière générale ces conditions.

- **Lumière** : La plupart des légumes nécessitent au moins 8 heures de lumière par jour pour produire de manière satisfaisante. Dans des zones très nuageuses ou durant les courtes journées d'hiver, un éclairage supplémentaire devant être nécessaire.

- **Gaz carbonique** : Les serres commerciales utilisent couramment des générateurs de CO₂ pour maximiser leur production. Lors de la conception d'un système de CO₂, les rendements n'augmenteront que si le CO₂ est le « facteur limitant ». Cela signifie que si toutes les autres variables ne sont pas optimales (lumière, engrais, température / humidité, pH, etc.), les avantages d'une augmentation des niveaux de CO₂ ne seront pas obtenus .

- **Mouvement de l'air** : il est aussi un facteur important qui affecte la croissance de la plante, modifiant les transferts d'énergie, la transpiration et l'absorption de CO₂, ce qui affecte la taille des feuilles, ainsi que la croissance de la tige et le rendement. Le taux de photosynthèse peut être augmenté de 40 pourcents si la vitesse du vent augmente de 10 à 100 centimètres par seconde.

- **Humidité** : l'humidité de l'air et celle du sol sont deux facteurs importants pour la croissance des plantes

- **Température** : la température du milieu intérieur et extérieur, du sol et de l'eau doit également respecter certaine norme

2.5 Contrôle des conditions environnementales

2.5.1 Contrôle de lumière

Le contrôle de la lumière peut concerner les actions suivantes :

- **Renforcement de l'éclairage** : Durant les courtes journées d'hiver ou dans les zones très nuageuses, un éclairage supplémentaire est requis. Le renforcement de l'éclairage peut être réalisé par une source supplémentaire d'éclairage basé sur des lampes électriques ou par la lumière réfléchié du soleil. La lumière réfléchié est surtout importante dans le cas des jardins urbains.

- **Réduction de l'éclairage** : s'il y a trop de lumière dans la serre, on peut la diminuer.

2.5.2 Gaz carbonique

Le contrôle du gaz carbonique peut concerner l'enrichissement en CO₂ (méthode de la glace carbonique). Le taux de CO₂ peut être augmenté en utilisant des générateurs de CO₂. Lors de la conception d'un système de CO₂, les rendements n'augmenteront que si le CO₂ est le « facteur limitant ». Cela signifie que si tous les autres facteurs ne sont pas optimales (lumière, engrais, température / humidité, pH, etc.), l'augmentation de CO₂ n'aura pas d'effet.

2.5.3 Humidité de l'air

Le contrôle de l'humidité peut concerner les deux actions suivantes :

- **Diminuer l'humidité élevée** : Ceci peut être réalisé par aérer l'air humide vers l'extérieur. À chaque fois que la température extérieure est suffisamment chaude, la ventilation vers l'extérieur peut diminuer l'humidité.

- **Augmenter l'humidité** : il est possible d'augmenter l'humidité en arrosant simplement le sol et en utilisant des climatiseurs de type refroidisseur de marais. Une autre méthode consiste à utiliser un système de brumisation pour pulvériser le brouillard dans l'air. Cela aidera à refroidir la serre tout en ajoutant de l'humidité à l'air.

2.5.4 Température

Le contrôle de température peut concerner :

- **Refroidissement** : le refroidissement peut être réalisé par plusieurs méthodes : par évaporation, par toile d'ombrage et par le mouvement de l'air. Le principe de refroidissement par évaporation est simple. Lorsque les ventilateurs d'extraction expulsent de l'air à l'une des extrémités de la serre, ils aspirent de l'air humide à l'autre extrémité. Le déplacement de l'air humide peut conduire une évaporation de l'eau et par conséquent une absorption de la chaleur. Le refroidissement peut être effectué en utilisant un rideau d'ombrage interne qui peut être tiré par temps nuageux et étendu (manuellement ou avec de petits moteurs) les jours ensoleillés à la demande.

- **Chauffage** : on peut chauffer la serre par un petit système électrique contrôlé par thermostat. Ce système peut délivrer la chaleur uniquement lorsque cela est nécessaire. on peut également chauffer la serre en utilisant du gaz avec un chauffage central.

2.6 Systèmes de contrôle

La figure 3.5 présente les éléments principaux qui composent un système de serre intelligente. Il existe de nombreuses techniques utilisées pour contrôler l'environnement des serres. Dans ce qui suit, on présente certaines des techniques plus utilisées pour contrôler les facteurs environnementaux.

2.6.1 Systèmes d'irrigation

L'irrigation est l'opération consistant à apporter artificiellement de l'eau à des végétaux cultivés pour en augmenter la production, et permettre leur développement normal en cas de déficit d'eau induit par un déficit pluviométrique, un drainage excessif ou une baisse de nappe, en particulier dans les zones arides [40].

Tout système d'irrigation implique les opérations de pompage, de traitement, de distribution et d'entreposage / Récupération de l'eau.

On peut distinguer plusieurs techniques d'irrigation [40] :

- ❖ Manuelle (arrosoir, seau...), réservée aux très petites surfaces ;
- ❖ Par écoulement de surface, sous le simple effet de la gravité, au moyen de canaux et rigoles : irrigation gravitaire appelée aussi irrigation de surface, irrigation par sillons ou « à la raie » ;
- ❖ Par aspersion, technique qui consiste à reproduire la pluie ;
- ❖ Par micro aspersion, semblable à la précédente mais plus localisée donc plus économe en eau ;
- ❖ Par micro irrigation ou goutte à goutte, technique économe en eau et qui permet d'éviter le ruissellement, mais présente le grave inconvénient de charger à la longue les sols en sels qui en modifient les caractéristiques.
- ❖ Par infiltration, au moyen de tuyaux poreux enterrés, variante de la technique du goutte à goutte.
- ❖ Par inondation ou submersion (c'est la technique appliquée dans les rizières ; c'était aussi celle qui fertilisait l'Égypte par les crues du Nil)

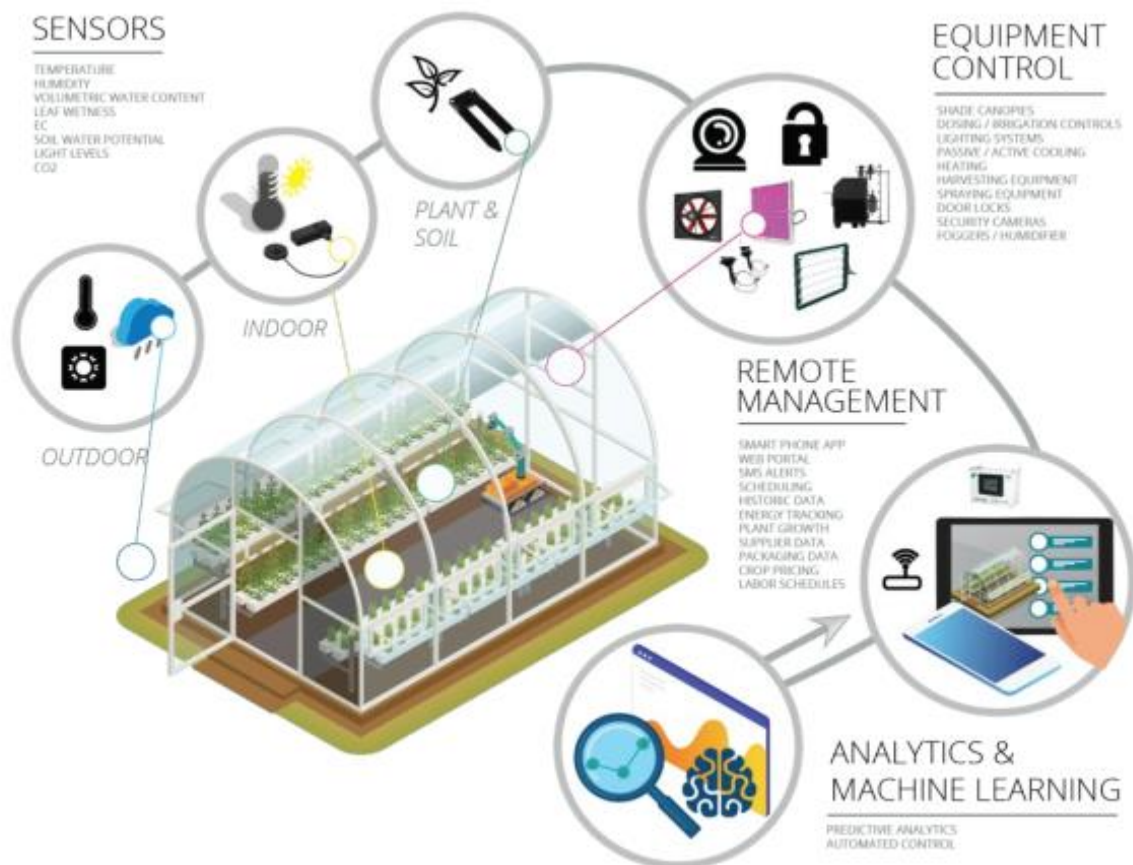


Figure 3.5 - Les éléments principaux qui composent un système de serre intelligente

2.6.2 Système d'ouverture des serres

Le système d'ouverture des serres permet d'aérer la serre afin d'ajuster la température, de l'humidité, de CO₂, etc. Un système de contrôle de température peut être réalisé en utilisant un capteur de température, qui se fait "lire" par un circuit électronique, qui lui décide à partir de quelle et jusqu'à quelle température il faut maintenir ouvert (Grâce à un petit moteur et un mécanisme de transformation du mouvement) [10].

2.6.3 Surveillance de l'environnement

La surveillance de l'environnement et de climat intérieur et extérieur des serres est une opération importante dans le cas des serres intelligentes automatisées. Elle est basée sur l'utilisation des divers capteurs à déployés à l'intérieur et à l'extérieur de la serre. Ces capteurs doivent être reliés à un système (filaire ou sans fil) de capture qui collecte en temps réel et d'une manière permanent les données fournies par ces capteurs.

3. Travaux IdO actuelles pour les serres

Dans cette section, nous présentons quelques travaux d'IdO ont été réalisé dans le cadre de systèmes de serres intelligentes. Il s'agit de [41] :

- **Kaa** : est une plate-forme IoT open source qui permet de relier différents capteurs, appareils connectés et installations agricoles. Elle permet de mettre en place des applications à utiliser par les agriculteurs et les producteurs de surveiller à distance le climat, les cultures et les équipements. Elle permet également de fournir des moyens d'analyse prédictive des cultures.
- **GetSenso** : est une application Cloud permet à l'utilisateur de surveiller à distance et en temps réel les conditions climatologiques de la serre via un ordinateur de bureau, un smart phone ou une tablette. GetSenso est basée sur la technologie de capteurs sans fil et IdO offre un paramètre de surveillance à faible coût dans une serre. La surveillance est basée sur les différents capteurs tel que celui de lumière, d'humidité, etc., placés dans la serre. GetSenso fournit également des informations permettant de définir les valeurs d'automatisation de la serre, l'agriculteur peut initier un processus d'automatisation tel que l'arrosage.
- **Bitponics** : est un projet de jardinage IdO open source destiné aux producteurs et aux agriculteurs qui planifient des rappels indiquant le moment où ils doivent prendre des mesures, des lectures de capteurs en temps réel, ainsi que la possibilité d'allumer et d'éteindre automatiquement les appareils connectés, ..., et tout cela à partir de n'importe quel navigateur Web.
- **Plantlink** : ce n'est pas un système IdO, mais un capteur d'humidité du sol connecté qui envoie à l'agriculteur un rappel d'arrosage (push, email, texto) quand cela lui convient le mieux. De cette manière l'agriculteur peut contrôler l'arrosage précisément en fonction de l'état du sol. Plantlink eut être relié directement avec la vanne et peut donc la contrôler automatiquement.
- **HarvestGeek** : est un système d'IdO de surveillance et d'automatisation des serres et jardins open source et communique selon le mode sans fil. Ce système fonctionne conjointement avec un système de perception qui s'appelle HarvestBot. HarvestGeek propose des modalités de notification par SMS, Facebook, Twitter ou par courrier électronique.

4. Conclusion

Le choix du type et de la structure de la serre dépend du climat de la région et du type de plante à cultiver dans cette serre.

Dans ce chapitre nous avons vu les types de serres agricoles les plus populaires, ensuite nous avons déterminé les paramètres les plus agissants sur la croissance de la culture sous serre, enfin nous avons défini les différents systèmes automatiques nécessaires à la gestion efficace des paramètres climatiques sous serre.

Pour fournir le meilleur climat possible aux plantes en s'appuyant sur une bonne connaissance de leurs caractéristiques et de leurs besoins à tous les stades de leur croissance et en choisir les meilleures technologies dans les serres.

CHAPITRE 4 :
Conception et réalisation
de système des serres

1. Introduction

La phase de conception est très importante pour fixer les choix techniques et de préparer l'implantation, elle décrit la solution (comment le problème est résolu) et elle doit servir de support pour l'implantation et la maintenance.

Dans ce chapitre, nous présentons les objectifs et principe de la solution proposée et l'architecture global du système

2. Objectifs et principe de la solution proposée

L'agriculture intelligente repose sur l'argent et l'application de la technologie dans l'agriculture et la production végétale. Les serres intelligentes doivent notamment relever les défis suivants :

- Changement permanent des conditions climatiques.
- Différentes conditions d'agriculture d'une plante à l'autre.
- Coût de production des maisons en plastique.
- Type de capteurs et leur nombre dans chaque maison en plastique.
- Connexion des serres à Internet.
- Les interfaces d'interaction homme/système inconfortables.

2.1 Objectif du projet

L'objectif dans ce projet est de créer un système de serre connectée et intelligente. Cette serre est conçue principalement pour établir les conditions nécessaires à la croissance de la plante, qui sont collectées à partir des capteurs préalablement placés dans la serre. Les données sont traitées immédiatement pour obtenir des données plus complexes que nous ne pouvons pas obtenir des capteurs. Les données de haut niveau constituent la base sur laquelle reposent les décisions et les prédictions, ce qui nous donne un aperçu de la situation dans la serre et le futur.

Ce système assure les fonctionnalités suivantes :

- Assure un contrôle automatisé et intelligent des serres pour établir les conditions climatologiques idéales et contrôler la quantité de matière organique à soumettre pour chaque espèce végétale
- Assure un contrôle assisté par ordinateur de ces serres où l'opérateur humain peut intervenir dans le processus de contrôle moyennant une interface appropriée.
- collecter l'ensemble des données des serres et de leur environnement à distances via un application Web,
- surveiller et établir des statistiques et des études stratégiques à distance et faire des prédictions à la base de ces données.

Notre système offre à l'utilisateur deux options de contrôle :

- Le contrôle manuel de la serre via un application
- Un autre contrôle automatique grâce au système intelligent.

Les services que notre système peut fournir (automatique ou manuelle) dans le cadre de contrôle sont :

- Arrosage des plantes et gestion de l'irrigation
- Gestion de ventilation
- Gestion de réservoir
- Gestion de l'éclairage

2.2 Principe de la solution proposé

Nous avons opté pour la décomposition de notre système en trois parties (figure .1)

- la première partie correspond au sous-système physique de capture et d'actionnement. Ce sous-système sert d'interface physique entre le reste du système et la serre. Il permet de collecter en permanence des données en temps réel sur la température, l'humidité, luminosité, ... de l'environnement intérieur de la serre. Il permet également de traduire des commandes du système en des actions réelles sur la serre. Par exemple : allumer la lumière. Ce système est mis en œuvre en utilisant la carte microcontrôleur Arduino, des capteurs physiques et des composants pour l'actionnement. Le langage de programmation pour ce système est une version simplifiée du langage C++ dédiée pour ce genre de cartes.

- La deuxième partie correspond à partie principale du système proposé. Cette partie est conçue pour l'abstraction des données des capteurs en identifiant les différentes situations contextuelles sur l'état de la serre et pour la prise des décisions convenables pour réaliser différents services. Cette partie implémente ainsi les fonctionnalités de perception, de décision, et d'action. Pour la réalisation de cette partie, nous avons opté pour l'introduction des réseaux de neurones pour reconnaître les situations de haut niveau impliquant le déclenchement des services appropriés et l'introduction des arbres de décision comme un outil de prise de décision. Cette partie est programmée en java et sensé exécuter sur un ordinateur proche à la serre.

-La troisième partie est implémentée comme un application web dynamique. Elle est donc réellement hébergée dans un serveur Internet. Elle permet de surveiller plusieurs serres et d'établir plusieurs genres de statistiques et ainsi que de prédiction.

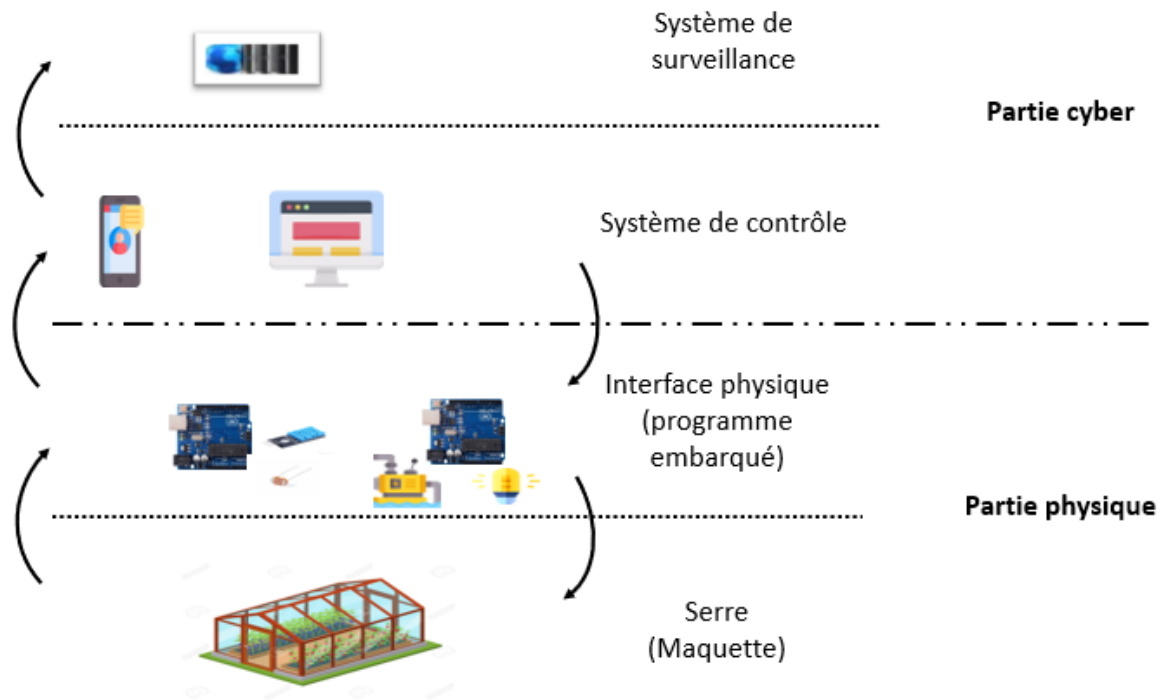


Figure 4.1 - Éléments du système

3. Architecture globale du système

3.1. Modules du système

3.1.1. Interface physique de capture et commande

Cette interface est composée de deux modules :

Module de capture des informations :

Composé d'une carte microcontrôleur et d'un ensemble de capteurs, ce module permet de collecter des données sur le climat intérieur de la serre, il s'agit de :

- La température et l'humidité de l'air
- La luminosité ou l'intensité lumineuse
- La densité du gaz
- L'humidité du sol

Module actionneur :

Composé également d'une autre carte microcontrôleur et d'un module relai et contrôle les composants suivants :

- Robinet Électrovanne pour contrôler le débit d'eau pour la régulation de l'irrigation.
- La lampe de lumière pour contrôler l'éclairage dans la serre.
- Ventilateur pour l'aération de la serre
- Pompe pour remplir le réservoir

Les actions à entreprendre par ce module à propos de ces composants sont : allumer(composant) et éteindre(composant).

Il est à noter que ces actions sont élémentaires. Les actions complexes sont gérées par le niveau supérieur.

3.1.2. Module de Perception et Reconnaissance de situations

Le Module de Perception et Reconnaissance de situations est le module qui permet de reconnaître les différentes situations à la base desquelles les décisions (du système ou l'opérateur humain) doivent être établies. Ces situations sont déduites à partir des données brutes fournies par module de capture.

Les différentes situations considérées dans le cadre de notre projet sont :

- Pour le sol : sec ou non sec.
- Pour la lumière : suffisant ou insuffisant.
- Pour le climat : chaud ou modéré

Le processus de calcul de situation se déclenche après la collecte des données requises pour une période déterminée.

3.1.3. Module de décision

Comme mentionné ci-dessus, l'un de nos objectifs est de créer un système intelligent capable de prendre des décisions de manière autonome. Donc ce module est crée pour fournir la suite des actions à entreprendre pour exécuter le service requis. Le mapping entre la situation courant déduite par le module de reconnaissance de situation et l'(les) action (s) à entreprendre est basé sur l'arbre de décision.

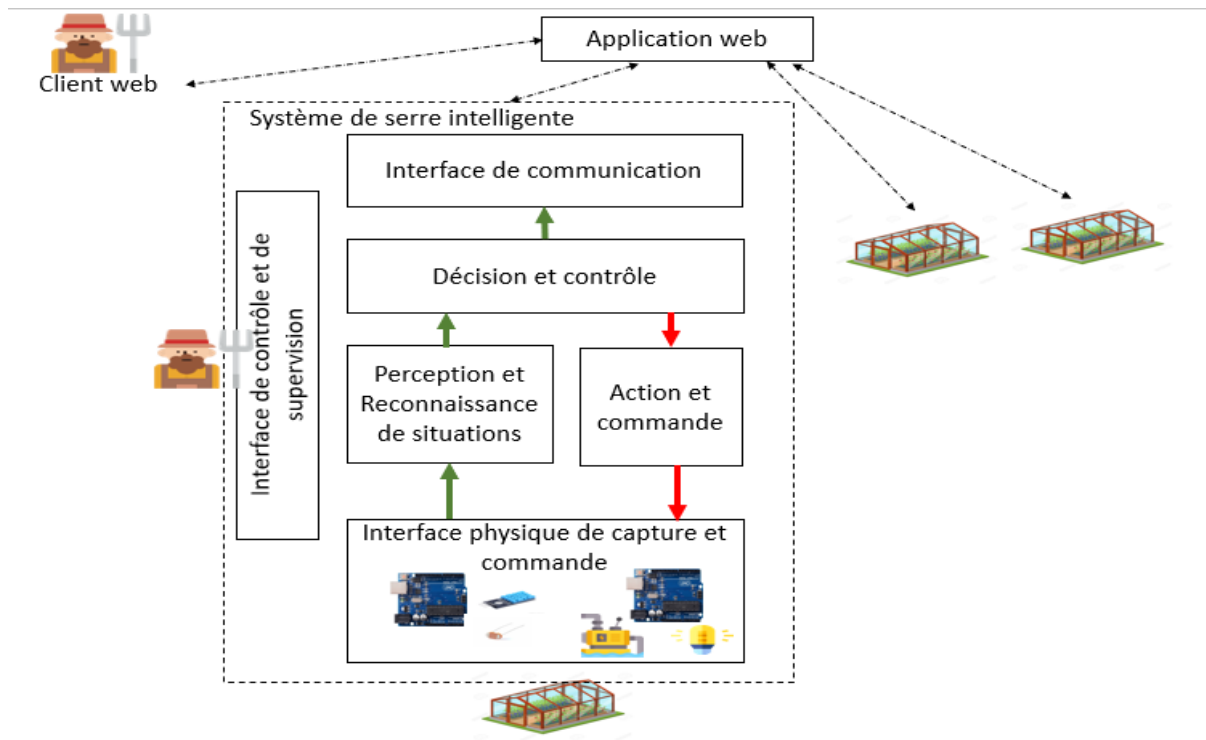


Figure 4.2 - Architecture global du système

3.2. Principe de fonctionnement

L'utilisateur du système est libre de basculer entre le contrôle automatique et manuel. Dans le cas de contrôle automatique, le fonctionnement du système est gouverné par les données acquises par les capteurs (via le module de capture).

À chaque fois où il y a un changement de situation le module décision se déclenche et détermine l'action à entreprendre à chaque fois. Ces actions concernent l'ouverture et la fermeture de l'arrosage, de l'éclairage, etc.

L'exécution des actions doit être effectuée d'une manière concurrente avec le reste du système.

Les données fournies et les décisions prises manuellement et automatiquement d'une durée de 24 heures seront fournies à l'application web.

Pour garder de la trace sur historique de son fonctionnement, les données obtenues et les actions entreprises par le système sont sauvegardées dans un journal. Ce journal peut se servir d'une base d'apprentissage ultérieur.

Le site web est conçu pour permettre le contrôle de plusieurs serres. Il étend également le système de contrôle local par des fonctionnalités relatives à l'analyse et aux statistiques et à la prédiction.

4. Gestion de serre

4.1 Gestion Température

La température de notre serre doit être comprise entre 32° et 34° pour une bonne croissance des plantes. Si la température atteint les 34° et plus, l'alarme sonore va être déclenchée, et la serre est commandée par moteur permettant une ventilation, sinon un chauffage électrique sera allumé automatiquement dès que la température est en dessous d'une valeur minimale fixé.

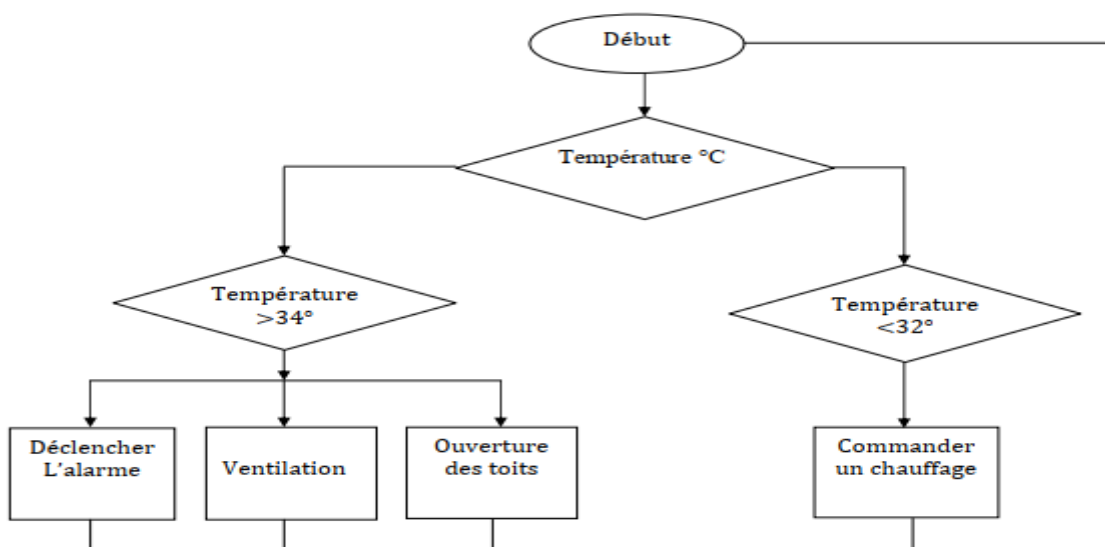


Figure 4.3 - Logigramme de fonctionnement de la température.

4.2 Gestion Humidité

La partie d'humidité de notre système est un autre type principal climatique, qui protège les occupants d'un milieu très humide ou bien trop sec. Si l'humidité descend sous un seuil, le système doit déclencher un arrosage automatique.

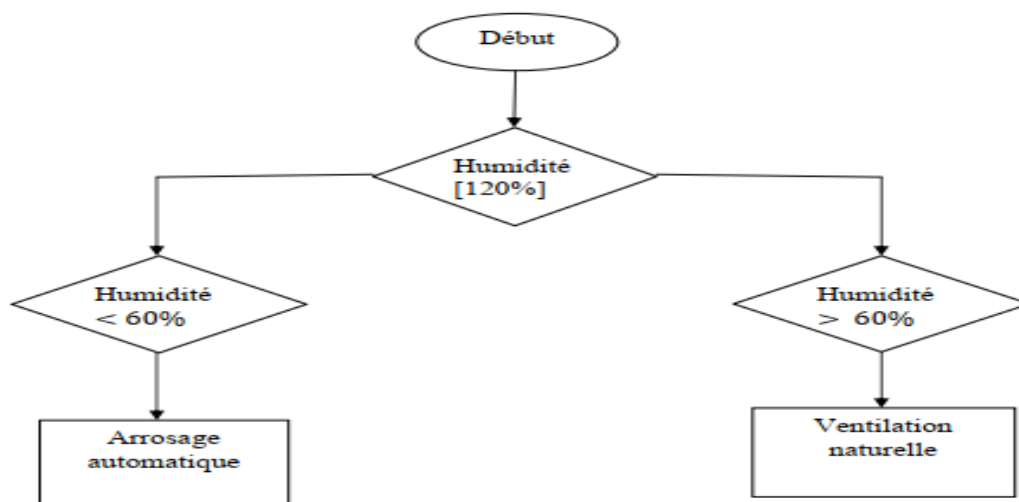


Figure 4.4 - Logigramme d'humidité

4.3 Gestion d'éclairage

Le service de Gestion d'éclairage se déroule selon le scénario suivant :

- Les capteurs (capteur de température - capteur d'humidité de l'air - capteur de luminosité) enregistrent les valeurs de lumière et d'humidité de l'atmosphère et température, puis envoyées à la carte Arduino.
- Le module reconnaissance situation désassemble les données et les affiche dans l'interface graphique sous forme de courbes, d'une part, d'autre part, la température moyenne, l'humidité moyenne de l'air et lumière sont utilisées comme entrées du réseau neuronal artificiel, qui reconnaît l'état de l'éclairage : élevé, modéré ou faible.
- Envoyer le résultat du réseau neurone artificiel à l'arbre de décision afin d'extraire la situation qui correspond aux données actuelles de la serre.
- En cas de contrôle automatique, module de décision envoie des demandes d'allumez la lampe et éteignez-la.

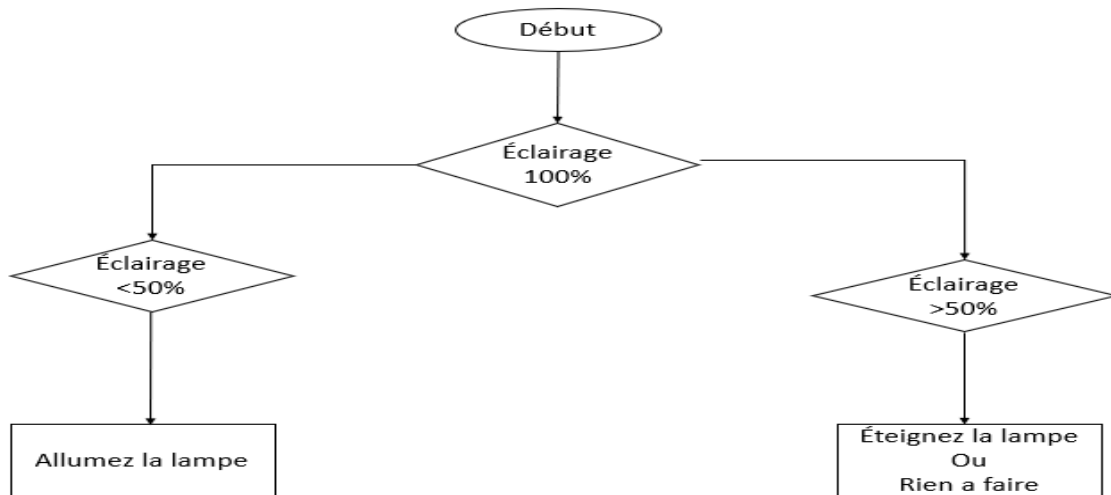


Figure 4.5 - Logigramme d'éclairage

5. Diagramme de classes

Le diagramme de classes décrit les types des objets qui composent un système et les différents types de relations statiques qui existent entre eux. Le diagramme de classes fait abstraction du comportement du système. Les classes qui composent notre système sont :

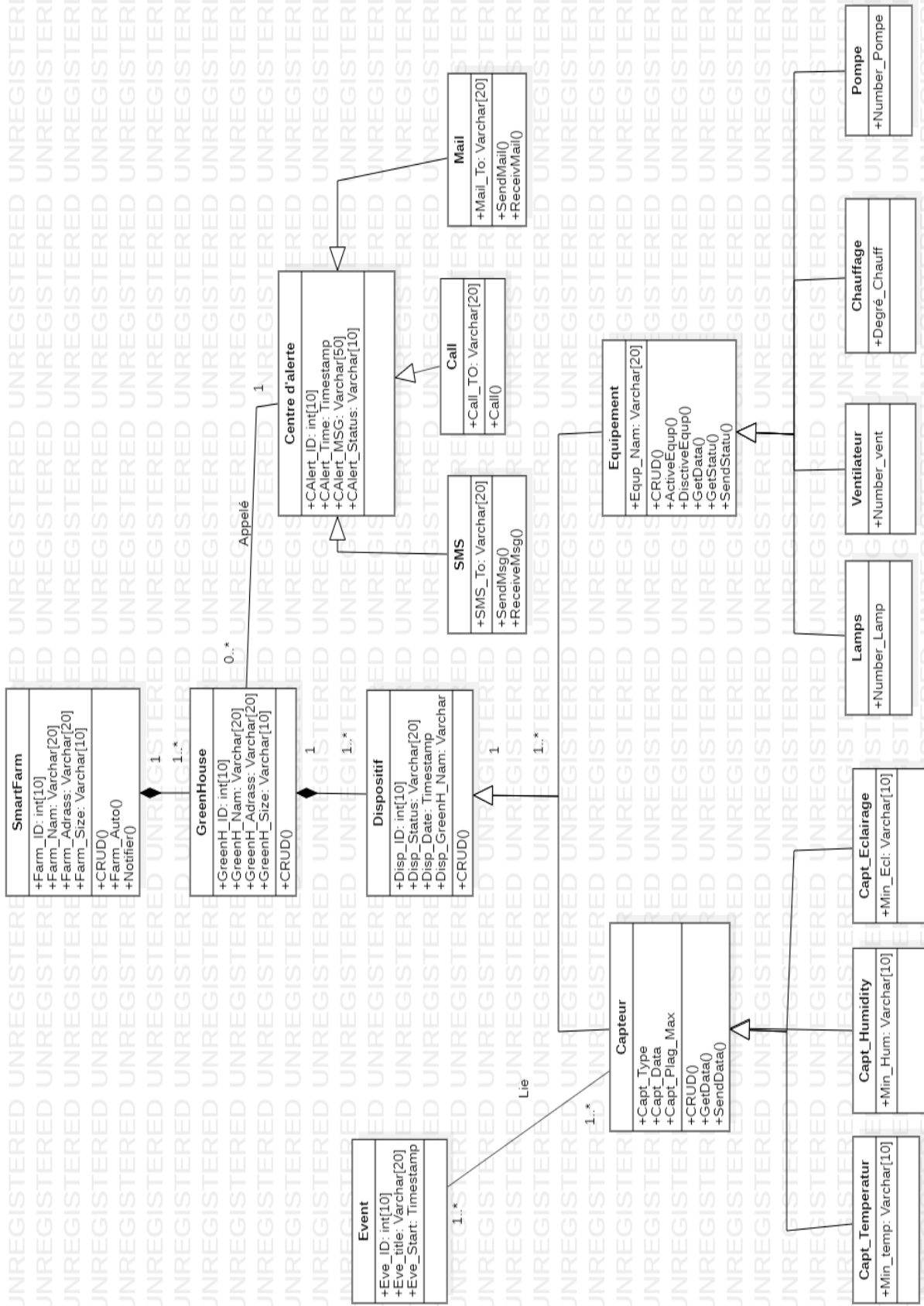


Figure 4.6 - Diagramme des classes

La représentation détaillée (les attributs et les fonctions) de chaque classe est décrite dans les figures suivantes :

5.1 Classe Centre d'alerte

C'est la class responsable d'envoi et de la réception des diverses Messages d'Alerte.

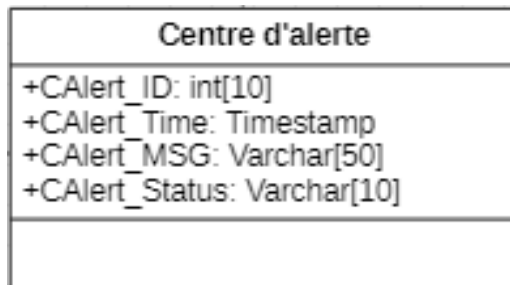


Figure 4.7 - Classe Centre d'alerte

SendMsg () : permet d'envoyer un Message d'Alerte.

Call () : permet de passer un appelle téléphonique.

ReceiveMsg () : permet de recevoir un Message envoyé par le Responsable au système.

5.2 Classe Equipement

C'est la class qui permet d'interagir entre la carte ESP32 (Actionneur) et le système, aussi le pilotage des équipements de serre.

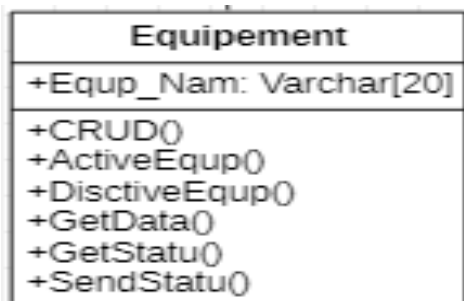


Figure 4.8 - Class Equipement

ActiveEqui () : permet d'activer l'équipement.

DesactiseEqui () : permet de désactiver l'équipement.

GetData () : permet de recevoir les données envoyer par le system.

GetStatut () : permet de récupérer le statut d'un équipement électrique.

SendStatut () : permet d'envoyer le statut d'un équipement au system (ON/OFF).

5.3 Classe capteur

C'est la classe qui permet d'interagir entre la carte ESP32 (Capteur) et le système, aussi la réception des différentes valeurs des paramètres climatiques.

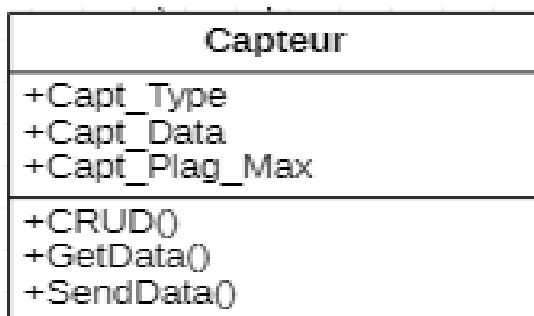


Figure 4.9 - Classe capteur

5.4 Classe évènement

C'est la classe qui permet de lie chaque information envoyer par la carte ESP32 (capteur) avec la date d'acquisition.

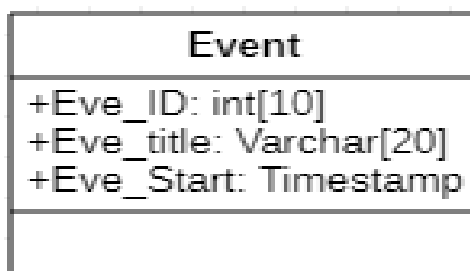


Figure 4.10 - Classe évènement

5.5 Classe SmartFarm

C'est la classe principale de notre system, elle se compose d'un à plusieurs de serre et permet de faire toutes les contrôles et les actions.

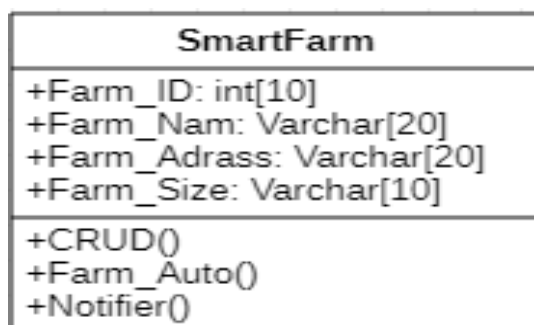


Figure 4.11 - Class SmartFarm

5.6 Classe GreenHouse

La classe qui représente une serre, elle se compose d'un ensemble des Dispositif (capteurs et équipements)

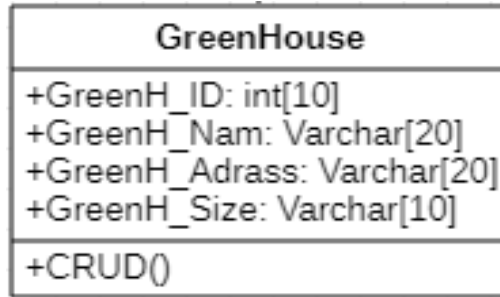


Figure 4.12 - Classe GreenHouse

6. Diagramme d'activité :

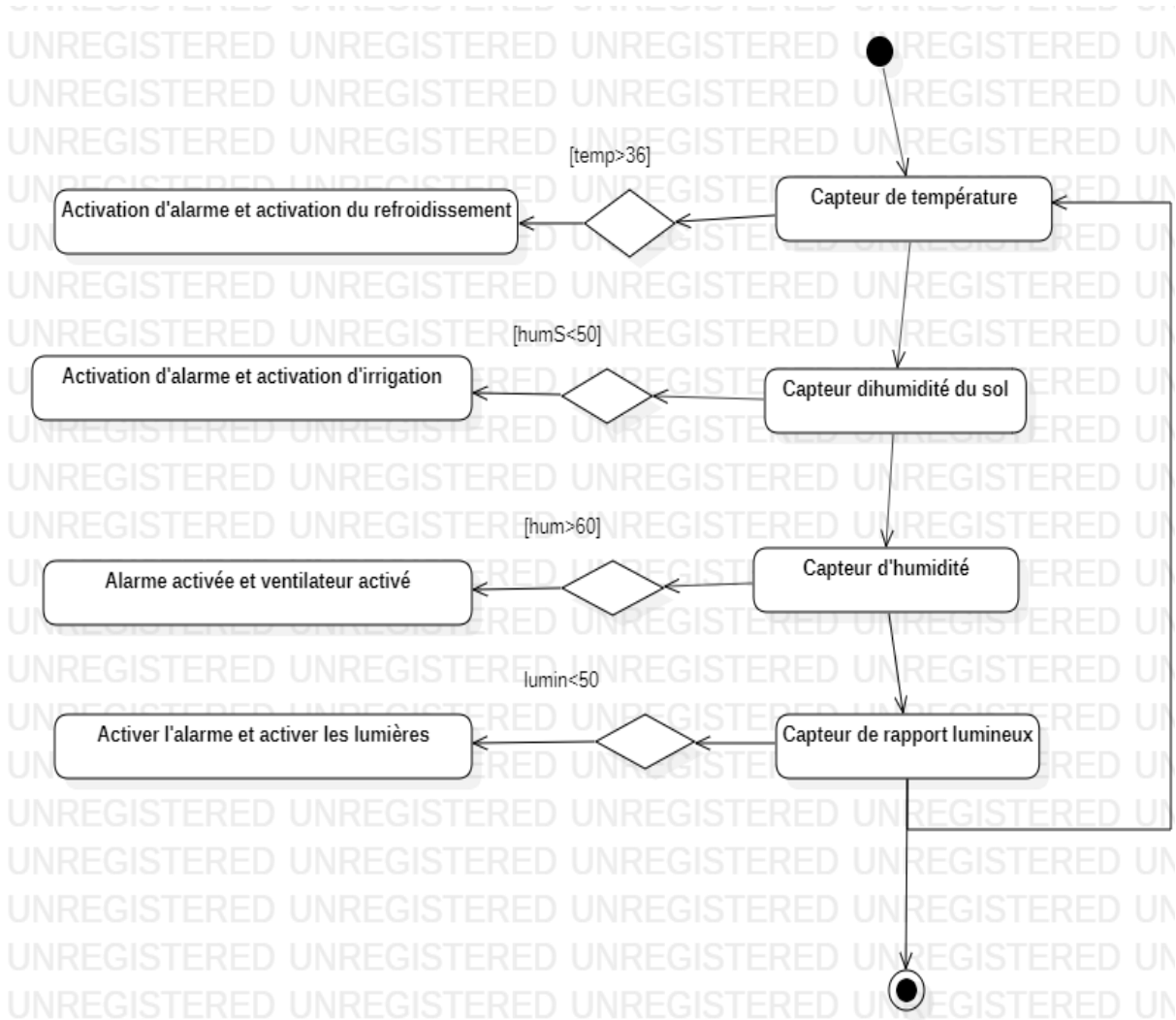


Figure 4.13 - Diagramme d'activité

7. Conclusion

Dans ce chapitre, nous avons présenté tous les détails de la conception de notre système de serres intelligentes. Nous avons présenté dans un premier lieu le principe de notre solution et l'architecture globale du système et expliqué ensuite le détail de chaque partie. Dans le chapitre suivant, nous présenterons les détails de l'implémentation.

CHAPITRE 5 :

Développement et Tests

6. Introduction

Cette partie de notre étude ne nous donnera pas de bons résultats si on néglige certains paramètres, donc le bon fonctionnement de notre système se base essentiellement sur une bonne démarche et une bonne réflexion de notre programme.

Le but de chapitre est d'énumérer les différentes étapes suivies pour arriver à réaliser ce projet depuis le montage jusqu'à la version finale du Système.

Nous commençons ce chapitre par la description du montage de notre maquette ; les outils utilisés pour la réalisation de système finale.

Nous décrirons par la suite les étapes du développement ainsi que la programmation en introduisant le code source.

Nous finirons par la description et l'illustration des interfaces en précisant leurs rôles et fonctionnements.

7. Cloud Storage « ThingSpeak »

ThingSpeak est une API et une application open source pour « Internet des objets », permettant de stocker et de collecter les données des objets connectés en passant par le protocole HTTP via Internet ou un réseau local. Avec ThingSpeak, l'utilisateur peut créer des applications d'enregistrement de données capteurs, des applications de suivi d'emplacements et un réseau social pour objets connectés, avec mises à jour de l'état. ThingSpeak peut être intégré aux plates-formes Arduino, Raspberry Pi, aux applications mobiles/Web, aux réseaux sociaux et aux analyses de données avec MATLAB.[31]

2.1. Fonctions de ThingSpeak

- ✓ API ouverte
- ✓ Collecte de données en temps réel
- ✓ Données de géolocalisation
- ✓ Traitement des données
- ✓ Visualisations de données

2.2. Les étapes pour utiliser ThingSpeak

- ❖ Etape 1 : Créez compte

Pour pouvoir télécharger les données sur ThingSpeak à des fins d'analyse et de traitement, vous devez créer votre compte.

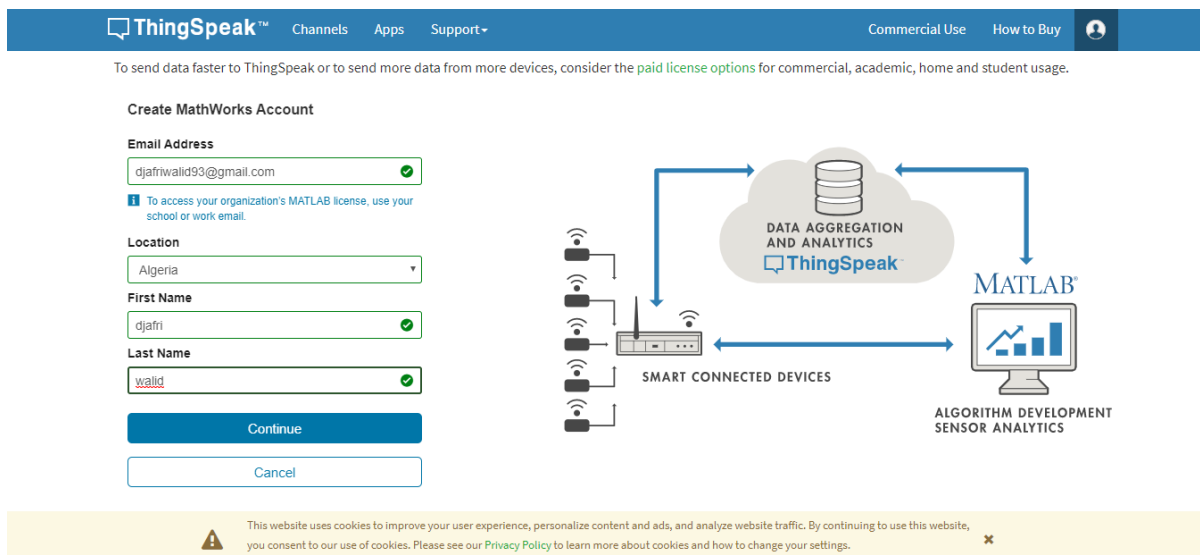


Figure 5.1 - Création de compte ThingSpeak

❖ Etape 2 : Créez un nouveau canal.

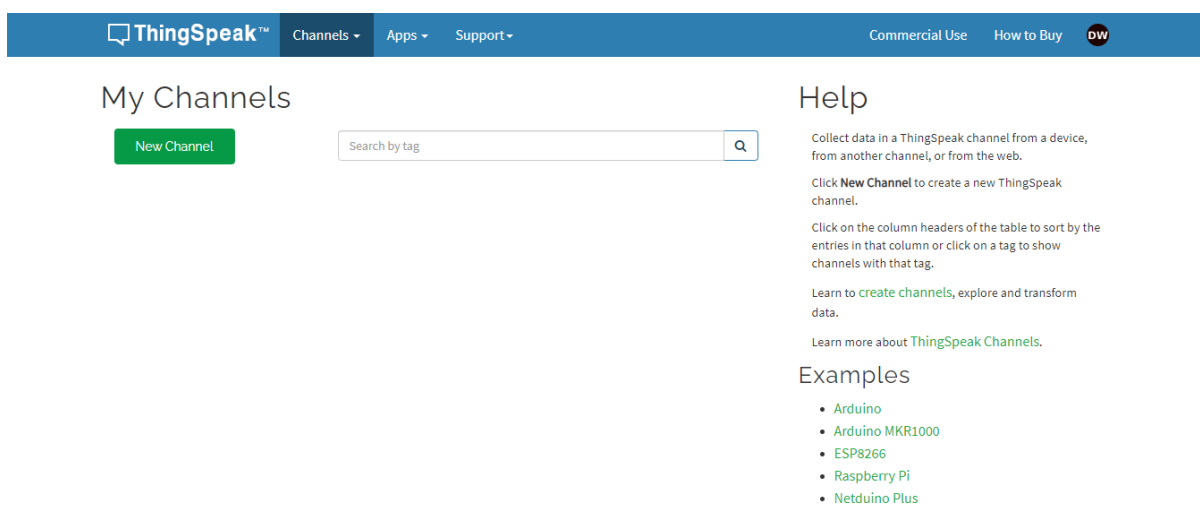


Figure 5.2 - Création un nouveau canal

Vous devrez saisir les détails du nouveau canal créé. Par exemple, vous pouvez utiliser le nom du projet sur lequel vous travaillez comme nom du canal. En cas de doute, vous pouvez laisser certaines options vides.

Les champs reflètent les données que vous allez télécharger. Par exemple, si vous mesurez la température de la pièce, vous pouvez définir le champ 1 comme Température de la pièce. À l'issue de cette opération, cliquez sur Enregistrer le canal.

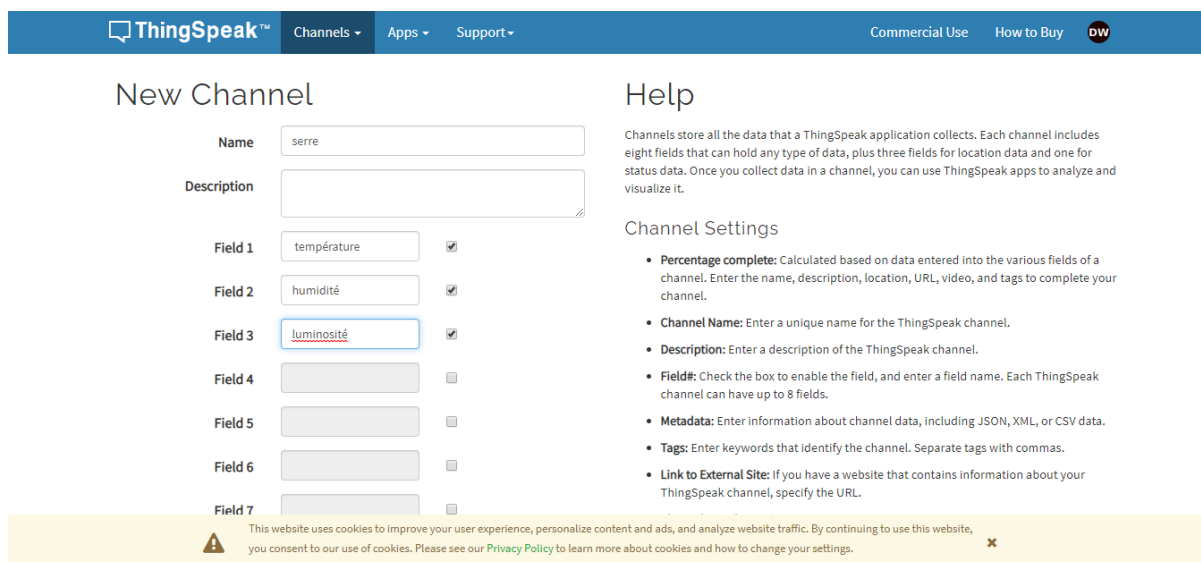


Figure 5.3 - Les détails du nouveau canal

❖ Etape 3 : Recherchez la clé API

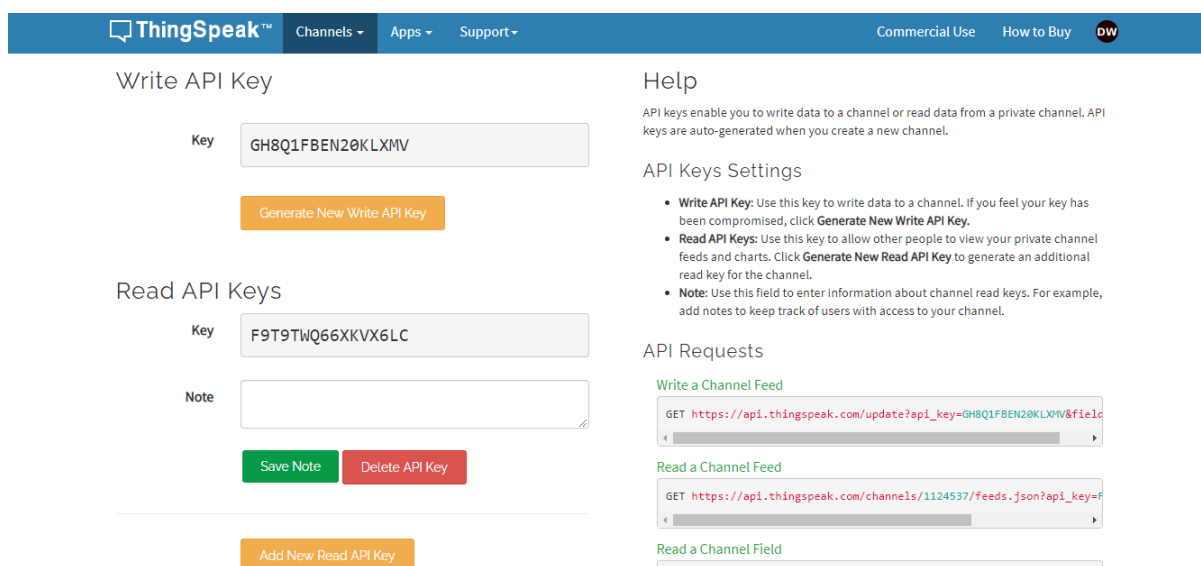


Figure 5.4 - Obtenir L'API Thingspeak

❖ Etape 4 : Programme de plate-forme Arduino

L'exemple de programme ci-dessous est adapté à l'utilisation d'une plate-forme Arduino, en association avec un module Wi-Fi.

```
char thingSpeakAddress [] = "api.thingspeak.com";
String writeAPIKey = "GX500IRSL529E1RY";
Const int updateThingSpeakInterval = 16 * 1000; //temps pour mettre à jour ThingSpeak
```


3. Montage et programmation de matériel

La figure ci-dessous représente le montage final de notre système composé de deux stations locale et distante. La carte ArduinoUno est entourée de plusieurs outils tels que le module WIFI ESP8266, les différents capteurs de température d'humidité et de luminosité ainsi que le capteur d'humidité du sol. Nous avons aussi l'alimentation en énergie à l'aide d'une batterie, des modules à relais et un LCD pour afficher.

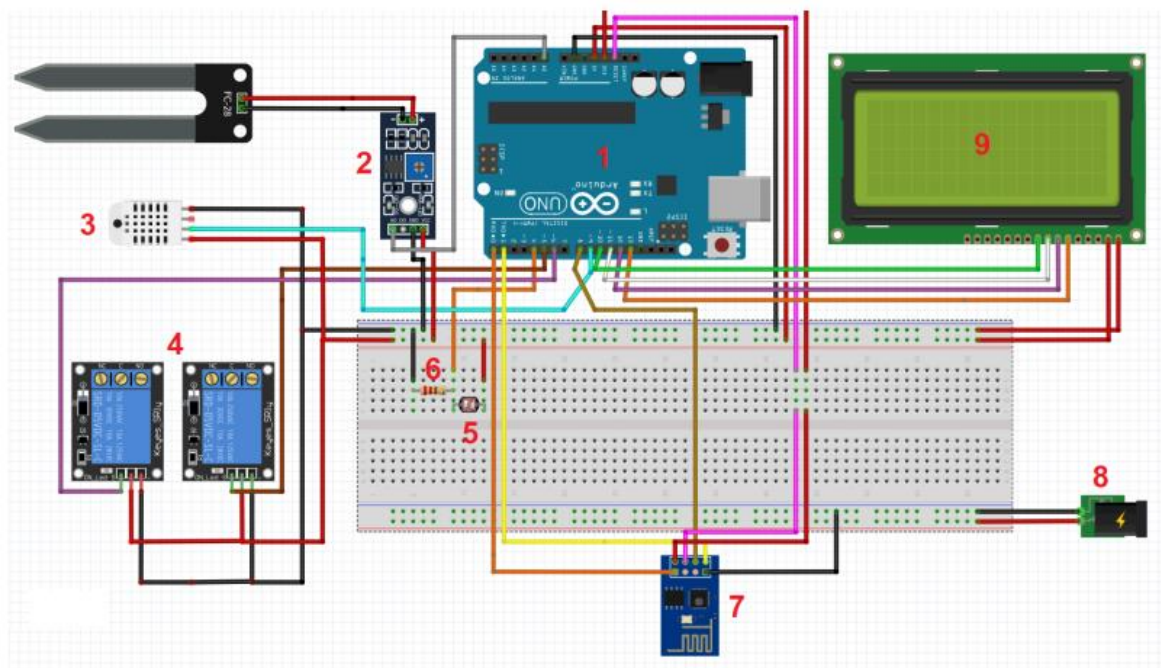


Figure 5.5 - Montage du système final

1	Arduino uno	6	Résistance
2	Capteur d'humidité du sol	7	Le module Wifi ESP8266
3	Capteur de température et d'humidité	8	Énergie
4	Les Modules à relais	9	LCD
5	Capteur de luminosité		

Tableau 3 - Les composants du système

3.1 DHT22 : Capteur de température et d'humidité

Le premier capteur à être installé et testé est le DHT 22, un capteur numérique d'humidité relative et de température qui, en utilisant un capteur d'humidité capacitif et une thermistance mesurant l'air environnant, crache un signal numérique sur la broche de données.

Le capteur doit être alimenté entre 3,3 V et 5 V (et fonctionnera de -40 ° C à + 80 ° C avec une précision de +/- 0,5 ° C pour la température et +/- 2% pour l'humidité relative.) Il est important de garder à l'esprit que sa période de détection est en moyenne 2 secondes (temps minimum entre les lectures).

Le DHT22 a 4 broches :

1. VCC (3 à 5 V)
2. DONNÉES (sortie de données)
3. NC (non connecté)
4. GND (Terre)

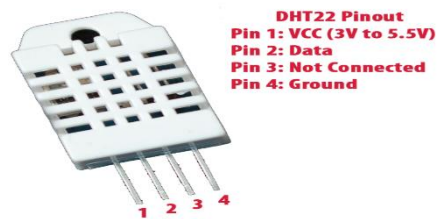


Figure 5.6 - DHT22

Considérant que vous utiliserez principalement le capteur sur des distances inférieures à 20m, une résistance de 10K ohms doit être connectée entre les broches Data et VCC. La broche de sortie doit être connectée à Arduino Pin5 comme indiqué sur le schéma précédent :

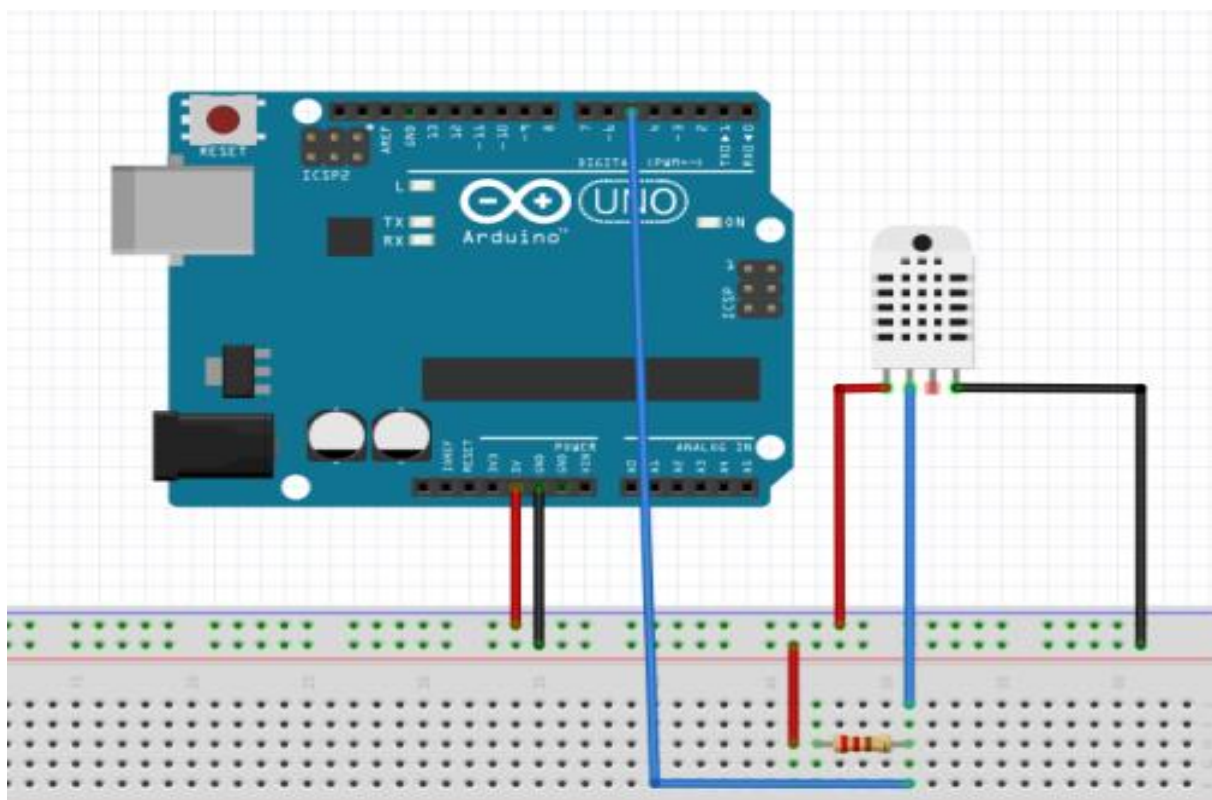


Figure 5.7 - Montage Arduino et DHT22

Une fois le capteur installé sur Arduino, il fallait télécharger la bibliothèque DHT depuis le référentiel GitHub : <https://github.com/adafruit/DHT-sensor-library> et l'installer dans le fichier de bibliothèque Arduino.

Le code source:

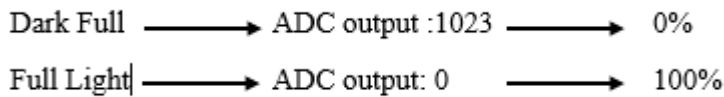
```
//Libraries
#include <DHT.h>;
//Constants
#define DHTPIN 7 // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor for normal 16mhz
Arduino
//Variables
int chk;
float hum; //Stores humidity value
float temp; //Stores temperature value
void setup()
{
  Serial.begin(9600);
  dht.begin();
}
void loop()
{
  delay(2000);
  //Read data and store it to variables hum and temp
  hum = dht.readHumidity();
  temp= dht.readTemperature();
  //Print temp and humidity values to serial monitor
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %, Temp: ");
  Serial.print(temp);
  Serial.println(" Celsius");
  delay(10000); //Delay 2 sec.
}
```

3.2 Capteur de luminosité

Un simple LDR (Light Dependent Resistor) peut être utilisée. Fondamentalement, ce que nous devrions faire est d'avoir un diviseur de tension où l'une des résistances est le LDR et le point central du diviseur devrait être utilisé comme une entrée analogique pour Arduino. De cette façon que la variation de la lumière, la résistance LDR varie également, la tension du point central du diviseur va également changer proportionnellement [33].

Pour les tests, nous allons utiliser un module LDR bon marché qui a le diviseur de tension intégré. Le module a 3 broches ("S" pour les données, "+" pour VCC et "-" pour GND). La broche "S" sera connectée à Arduino Pin Analog 0 (A0). Les broches "+" et "-" doivent être connectées respectivement à 5V et GND.

La fonction getLumen (LDR_PIN) lit quelques fois la sortie du capteur en calculant à la fin la moyenne de ces lectures. En outre, une fois que la sortie du convertisseur analogique-numérique d'Arduino est comprise entre 0 et 1023, nous devons ces valeurs pour obtenir les résultats suivants :



Voilà le montage :

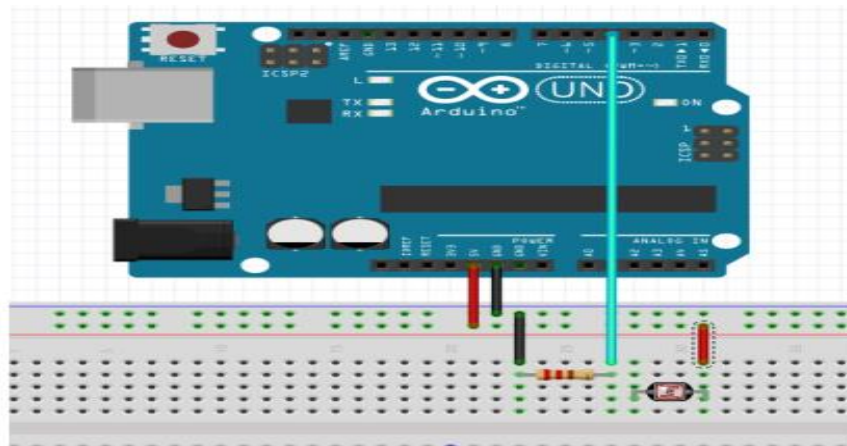


Figure 5.8 - Montage Arduino et LDR

Code source :

```
int getLumen(int anaPin)
{ intanaValue = 0;
for(int i = 0; i < 10; i++)
{
anaValue += analogRead(anaPin);
delay(50);
}
anaValue = anaValue/10;
anaValue = map(anaValue, 1023, 0, 0, 100); /
returnanaValue;
}
```

3.3. Capteur d'humidité du sol

Un capteur pour tester l'humidité du sol est très simple. Il a le même principe que décrit précédemment pour le capteur de luminosité. Un diviseur de tension à utiliser comme entrée de l'une des broches analogiques d'Arduino, mais à la place d'une "résistance dépendante de la lumière" nous aurons une "résistance d'humidité du sol dépendante". Le circuit de base est simple et peut être vu ci-dessous :

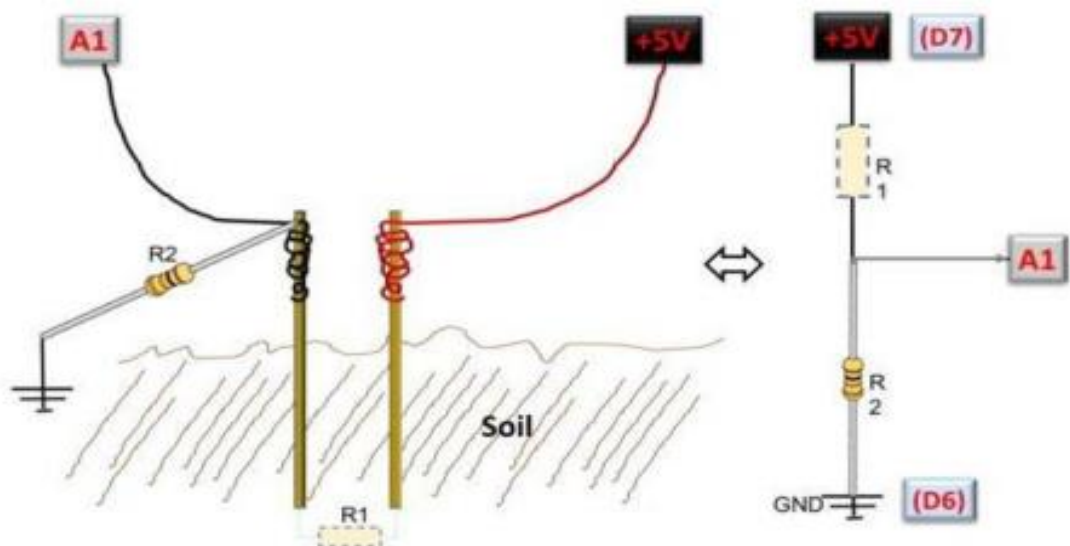


Figure 5.9 - Le circuit de capteur humidité du sol

Malheureusement, la réalité est un peu plus compliquée que cela (mais pas tellement). Un capteur simple comme décrit précédemment fonctionnerait très bien, mais pas pour longtemps. Le problème est que le fait d'avoir un courant constant circulant à travers les électrodes dans une seule direction va générer de la corrosion sur eux en raison de l'effet d'électrolyse. Une façon de le résoudre est de connecter les électrodes non à VCC et à la masse, mais à un port Arduino Digital. Pour ce faire, d'abord le capteur ne serait "excité" que lorsque la lecture devrait réellement se produire et que la direction du courant au-dessus des sondes pourrait être effectuée dans les deux directions, éliminant ainsi l'effet de l'électrolyse.

Un code de test simple :

```
const int analogInPin = A0;
int sensorValue = 0; // value read from the pot
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  delay(2);
}
```

3.4. L'affichage sur LCD

Pas toujours nous avons un moniteur série disponible pour analyser les sorties de nos capteurs. Par conséquent, un écran LCD sera ajouté au projet pour la surveillance locale. Le choix s'est porté sur un module LCD de haute qualité à 4 lignes de 20 caractères.

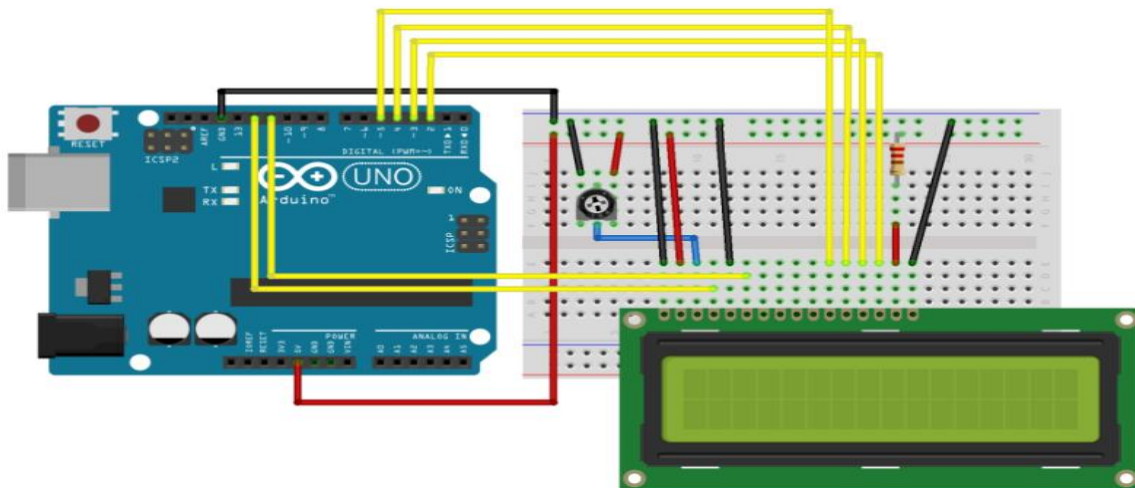


Figure 5.10 - Montage Arduino et LCD

```
Code source :
void setup() {
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
  lcd.setBacklight(HIGH);
  delay(500);
  lcd.setBacklight(LOW);
  delay(500);
}
```

3.5 Actionneurs

Jusqu'à présent, nous pouvons lire les données des capteurs et les afficher sur le moniteur série et sur l'écran LCD. Il est temps de faire quelque chose avec de telles données. Pensons aux actionneurs.

En regardant les données fournies par les capteurs, nous connaissons la température et l'humidité de l'air, la luminosité et le plus important comment est le sol. Avec ces données à portée de main, notre programme devrait calculer s'il est nécessaire d'irriguer, en allumant la pompe à eau ou la lampe électrique pour la chaleur. Pour cela, nous utiliserons un petit module relais 5V pour l'activation de la pompe et de la lampe. Le circuit du diagramme du module de relais peut être vu ci-dessous :



Figure 5.11 - Montage Arduino et actionneur

Pour la sortie, 3 broches : "NO", "Réf", "NC", qui sont : "Normal Ouvert", "Référence" et "Normal Fermé". Nous utiliserons la paire : NO et Réf (centre). Sur le schéma ci-dessus, "NO" est le terminal à connecter à "Live Mains" ou le positif direct de la source d'alimentation (12VDC pour la pompe et 220VAC pour la lampe). Le "Réf" sera connecté à la lampe ou à la pompe comme indiqué sur le schéma ci-dessus.

3.6 Tests de capteurs ensemble

Maintenant, tous les capteurs sont prêts et testés individuellement, créons une fonction spécifique pour lire tous les capteurs en même temps.

Fonction pour lire l'information des capteurs :

```
void readSensors(void)
{
  tempDHT = dht.readTemperature();
  humDHT = dht.readHumidity();
  lumen = getLumen(LDR_PIN);
  soilMoist = getSoilMoist();
}
```

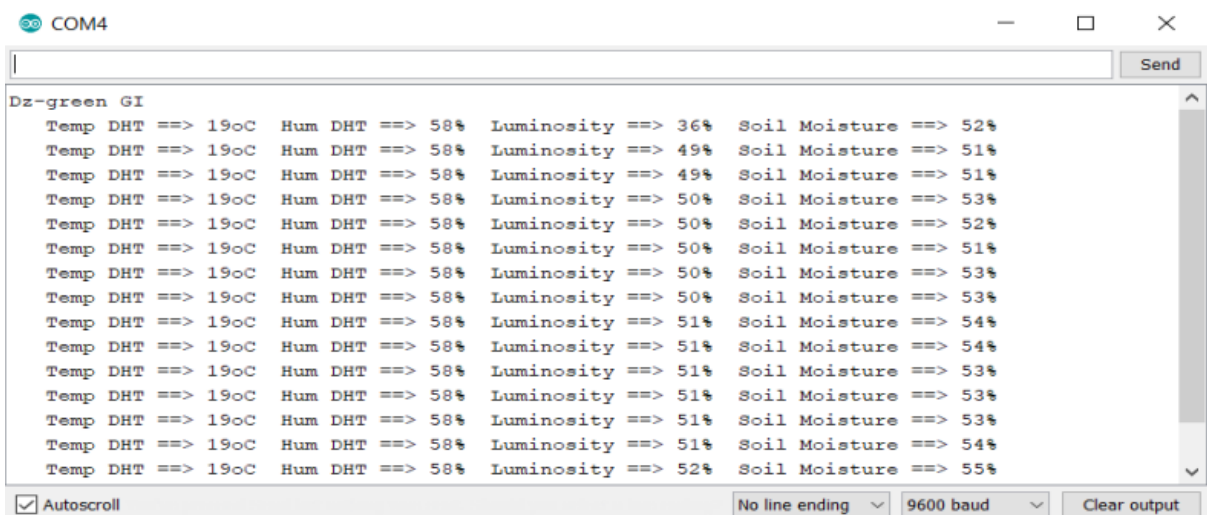


Figure 5.12 - Affichage des informations de système

3.7 ESP8266 (Module Wifi)

Pour se connecter à Internet, nous utiliserons ESP8266-01, un module simple, peu coûteux et facile à utiliser pour les projets d'objets Internet (IoT).

Le schéma électrique montre ici plus en détail comment connecter l'ESP8266.

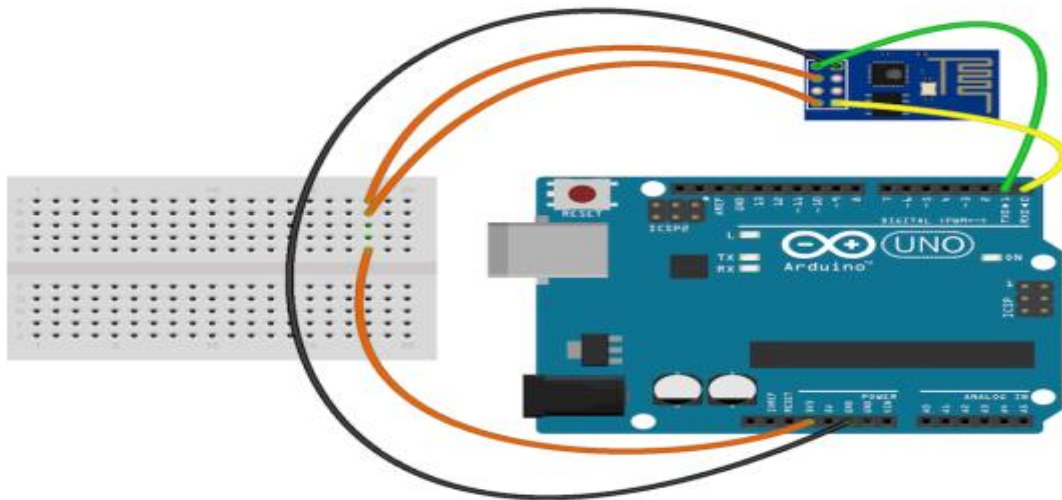


Figure 5.13 - Montage Arduino et ESP8266-01

Code pour les tests et la configuration ESP8266-01 :

```
#include<SoftwareSerial.h>
SoftwareSerial esp8266(2,3); //Rx ==> Pin 2; TX ==> Pin3
#define speed8266 9600 // <== ***** This is the speed that worked with my
ESP8266
#define CH_PD 4
voidsetup()
{
esp8266.begin (speed8266);
Serial.begin(speed8266);
```



```

reset8266(); // Pin CH_PD need a reset before start communication
Serial.println("ESP8266 Setup test - use AT coomands");
}
void loop()
{
while(esp8266.available())
{ Serial.write(esp8266.read());}
while(Serial.available())
{ esp8266.write(Serial.read()); }
}
void reset8266 ()
{
pinMode(CH_PD, OUTPUT);
digitalWrite(CH_PD, LOW);
delay(300);
digitalWrite(CH_PD, HIGH);
}

```

Connexion de l'ESP8266 à Internet :

```

void connectWiFi(void)
{
sendData("AT+RST\r\n", 2000, DEBUG); // reset
sendData("AT+CWJAP=\"YOUR USERNAME\", \"YOUR PASSWORD\" \r\n", 2000,
DEBUG);
//Connect network
delay(3000);
sendData("AT+CWMODE=1\r\n", 1000, DEBUG);
sendData("AT+CIFSR\r\n", 1000, DEBUG); // Show IP Adress
lcd.clear();
lcd.print("8266 Connected");
Serial.println("8266 Connected");
}

```

3.8 Data Storage Cloud

Le ThinkSpeak.com Toutes les données capturées telles l'humidité du sol, la température et l'humidité seront transférées sur le Cloud, en utilisant le service gratuit de "ThinkSpeak.com ».

Les fonctions pour transférer les données capturées :

```

void updateCmdThingSpeak(void)
{
digitalWrite(YELLOW_LED, HIGH);
for (int i = 0; i < 2; i++)
{
startThingSpeakCmd ();
cmd = msg ;
}
}

```

```

cmd += "&field1="; //champ 1 pour la température DHT
cmd += tempDHT;
cmd += "&field2="; //champ 2 pour l'humidité DHT
cmd += humDHT;
cmd += "&field3="; //champ 3 pour la luminosité LDR
cmd += lumen;
cmd += "&field4="; //Champ 4 pour les données sur l'humidité du sol
cmd += lampStatus;
cmd += "&field5="; // Champ 5 pour l'état de la POMPE
cmd += pumpStatus;

cmd += "&field6="; // champ 6 pour le statut LAMP
cmd += lampStatus;
cmd += "\r\n";
sendThingSpeakCmd();
delay (500);
}

```

Nous transférons toutes les données collectées et nous les stockons dans le Cloud.

Ceci est remarquable et utile pour une surveillance à distance, mais que se passe-t-il si, sur la base de ces données, nous voulons également allumer la pompe ou la lampe, indépendamment du programme automatique ?

Pour le faire, nous aurons également besoin de "télécharger" les données du Cloud et d'ordonner au contrôleur d'agir en fonction de ces commandes.

Nous allons créer des champs spécifiques sur notre canal ThingSpeak pour commander les actionneurs :

Field 7	0	La pompe devrait être éteinte
	1	La pompe doit être allumée
Field 8	0	La lampe devrait être éteinte
	1	La lampe devrait être allumée

Tableau 4 - Les canaux ThingSpeak

Code source :

```

void aplyCmd()
{
if (field7Data == 1) digitalWrite(PUMP_PIN, HIGH);
if (field7Data == 0) digitalWrite(PUMP_PIN, LOW);
if (field8Data == 1) digitalWrite(LAMP_PIN, HIGH);
if (field8Data == 0) digitalWrite(LAMP_PIN, LOW);
}

```

4. Développement et présentation de l'application

4.1 Vue générale de la structure du projet

Notre projet est opérationnel et peut être contrôlé depuis l'application. Vous pouvez surveiller les données sur le site ThingSpeak et envoyer des commandes en utilisant un navigateur, mais bien sûr, cette "solution web" ne peut pas être considérée comme une "élégante". La meilleure façon de mettre en œuvre une solution IOT complète est de développer une application complète qui affichera toutes les données, avec également des boutons pour activer les actionneurs.

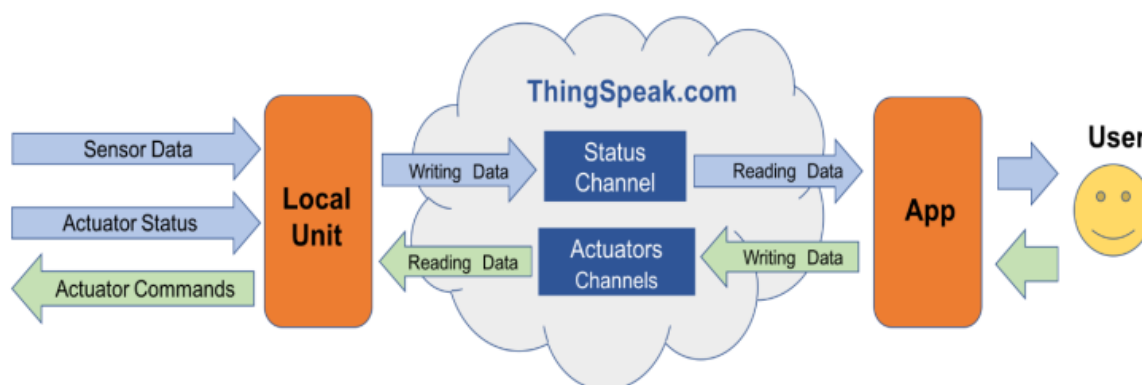


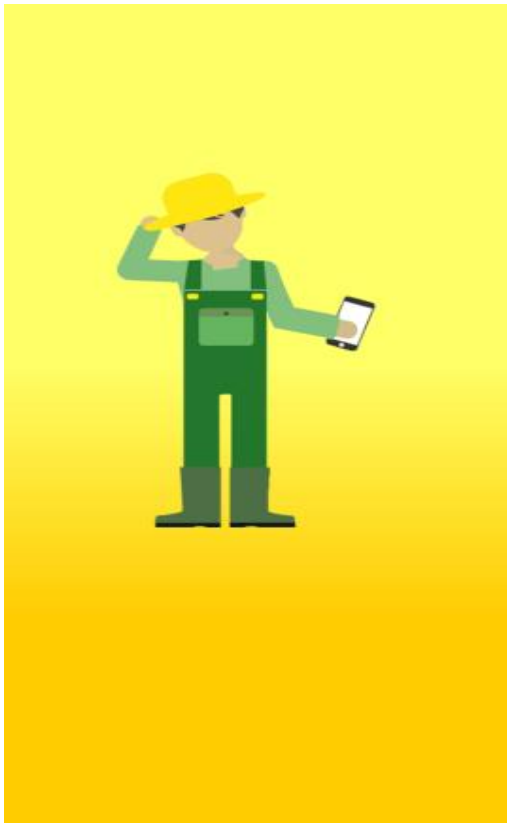
Figure 5.14 - Vue générale de la structure du projet

Le centre de notre projet IOT sera le ThingSpeak.com. L'unité locale (UNO / ESP-01) capturera les données provenant des capteurs et de l'état de l'actionneur, les enverra sur Internet, "writing" sur un canal d'état ThingSpeak.com spécifique. L'unité locale recevra également des données provenant d'Internet, les "reading" à partir de canaux d'actionneurs ThingSpeak spécifiques.

Une application saura également lire ces données à partir du canal d'état ThingSpeak.com et les afficher pour l'utilisateur. De même, l'utilisateur, sur la base de ces informations d'état, peut envoyer des commandes aux actionneurs, en les écrivant sur les canaux de l'actionneur ThingSpeak (voir le diagramme ci-dessus pour mieux comprendre le flux de données).

4.2 Interfaces de l'application

Dans ce qui suit, nous présentons notre application Android en exposant les différentes interfaces que nous avons réussie à mettre en place.



La figure 5.15 illustre l'Interface de chargement contenant le logo de l'application

Figure 5.15 - Interface de chargement.

La figure 5.16 illustre l'interface de la page d'accueil après avoir accédé à l'application, ensuite l'interface de menu qui apparue dans le but de choisir qu'elle genre d'information voulant-on savoir ou gérer.

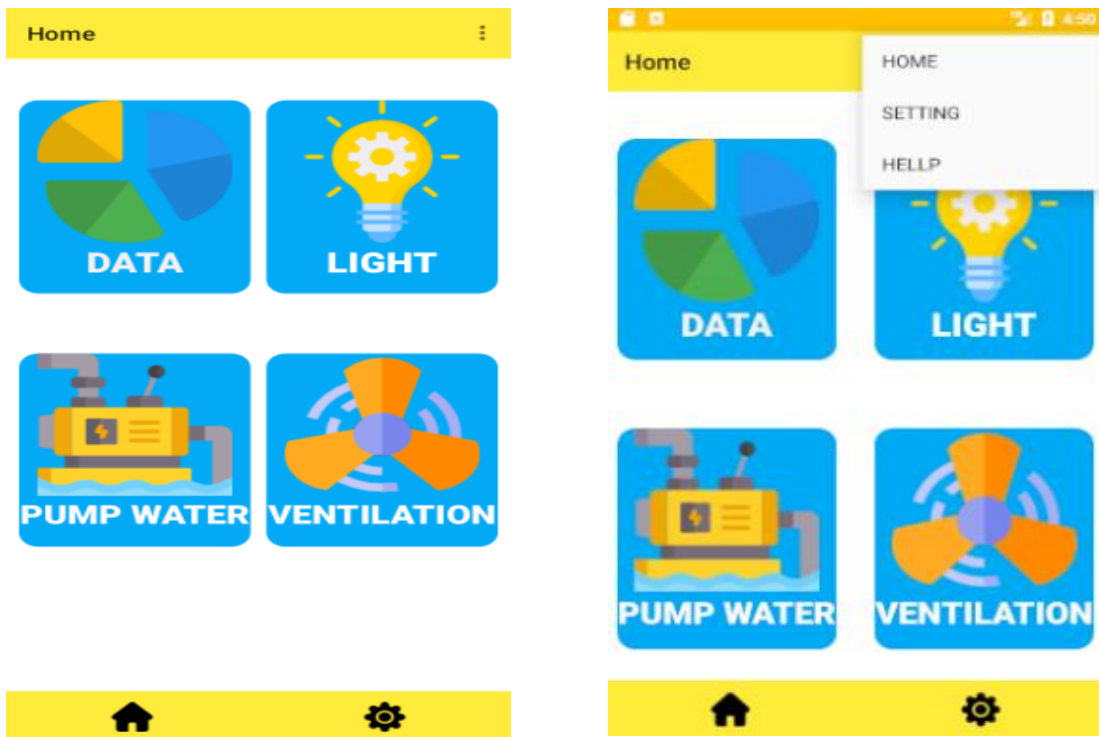


Figure 5.16 - Interface d'accueil et menu.

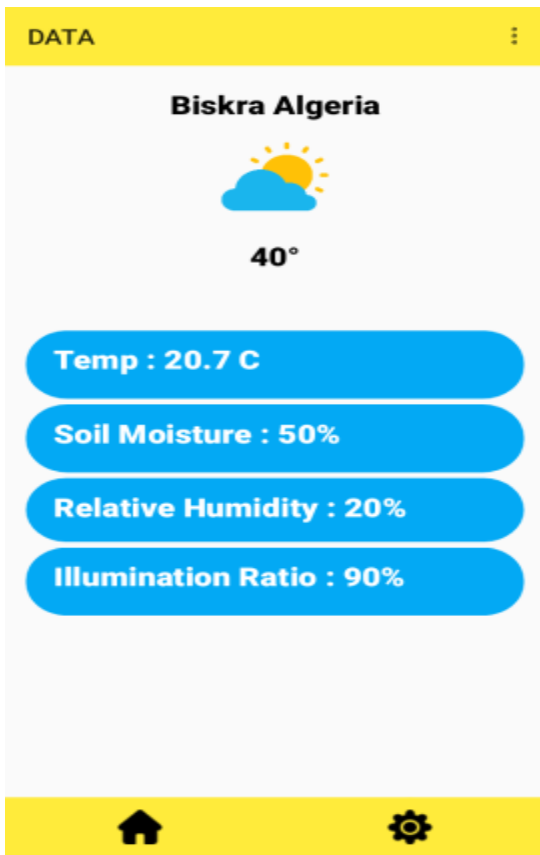


Figure 5.17 - Interface de suivi de la météo.

Cette figure montre l'interface de surveillance météorologique en direct : observations en direct de la température, de l'humidité, du rapport de lumière et de l'humidité du sol .

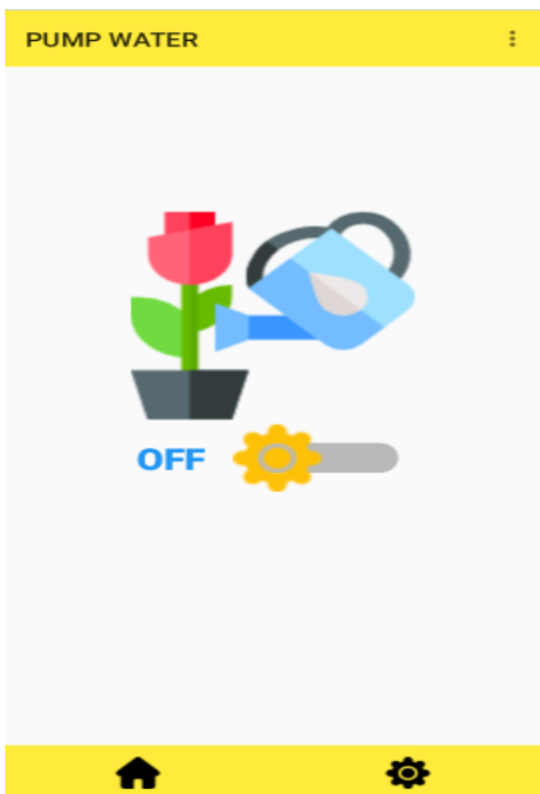
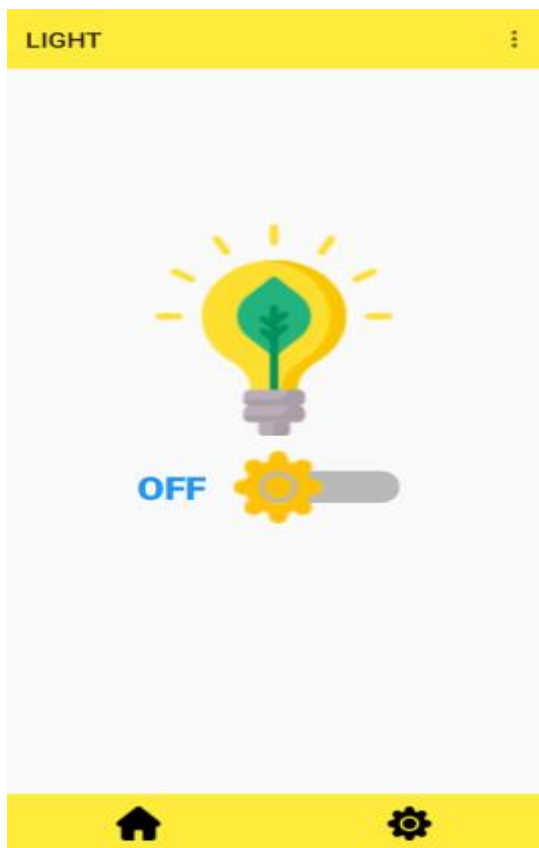


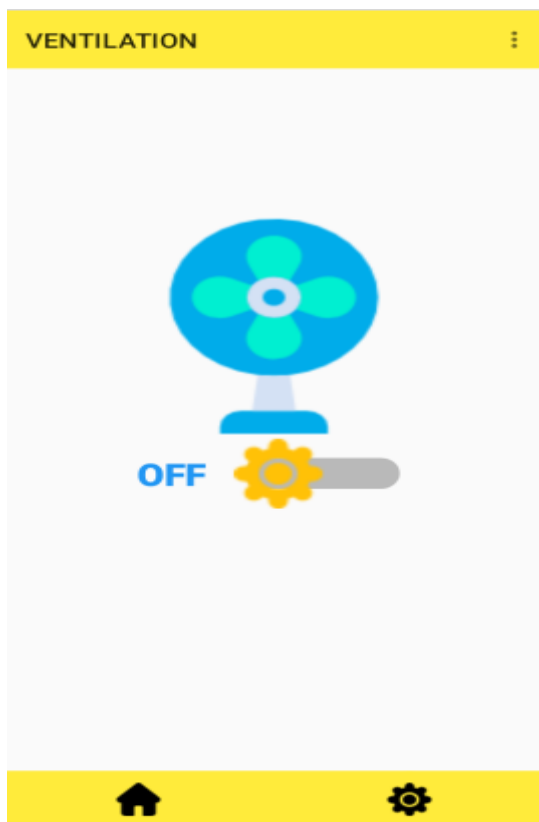
Figure 5.18 - Interface de contrôle de la pompe.

La figure illustre l'interface qui permet de contrôler la pompe d'arrosage.



La figure illustre l'interface qui permet de contrôler et de gérer la lumière nécessaire à la production en utilisant la lampe

Figure 5.19 - Interface de contrôle de la lampe.



La figure 5.20 illustre l'interface qui permet de contrôler la ventilation.

Figure 5.20 - Interface de contrôle du ventilateur.

5. Conclusion

Nous avons consacré ce chapitre pour la partie développement du système ainsi que pour les étapes de la réalisation de la serre agricole automatisée. Nous avons présenté en détails la réalisation de notre projet ou nous avons décrit le montage du matériel Arduino et notre programmation pas à pas. Par la suite nous avons décrit le principe de notre projet ainsi que son environnement de développement

Conclusion Générale

Nous avons pu découvrir l'importance de l'Internet des objets dans un futur proche et lointain car il contribuera au développement de tous les domaines de la vie tels que l'industrie, l'agriculture, la santé, etc.

Grâce à ce projet, nous avons également pu énumérer toutes les informations sur les composants de l'Internet des objets et tout ce qui y est lié à partir des réseaux et des composants tels que les capteurs et les sols.

Et nous avons choisi le domaine agricole pour appliquer cette technologie moderne en fournissant à une serre agricole l'Internet des objets pour le contrôler et le surveiller pour améliorer la production et collecter des informations et être contrôlé et surveillé via une application Android que le client transporte partout.

Ce fut un plaisir de chercher et découvrir tout ce qui touche à cette technologie et à son importance. J'espère que le concept d'Internet des objets se répandra en Algérie, car il ne concerne pas seulement les propriétaires d'un domaine, mais concerne tous les domaines et contribue à améliorer la vie en général.

Références

- [1] <https://nicholaskellettdotcom.files.wordpress.com/2016/04/1600px-internet-of-things-e1461017506926.jpg> . (consulté le 30 Janvier 2020).
- [2] <https://blog.octo.com/modeles-architectures-internet-des-objets/> . (Consulté le 30 Janvier 2020).
- [3] D.Evans. *L'Internet des objets. Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?*. Livre Blanc, Cisco IBSG, Etats-Unis, Avril 2011.
- [14] <http://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/>, (consulté le 30 Janvier 2020).
- [15] <https://www.supinfo.com/articles/single/7358-internet-of-things-iot> , (consulté le 30 Janvier 2020).
- [16] Y.Ait Mouhoub et F.Bouchebbah. *Proposition d'un mod`ele de confiance pour l'Internet des objets*. Mémoire master de l'université Abderrahmane Mira Bejaia, 21 Juin 2015.
- [17] Raouf Achour, Naima Makhloufi, Abdellah Boukerram, et al. *Authentification Dans L'iot*. PhD thesis, Université abderrahmane mira béjaia, 2017.
- [18] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things : A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials, Article*, 17(4) :2347–2376, 2015.
- [19] Sofiane, M. O. A. D., and Nizar Bouabdallah. "La consommation d'énergie dans les réseaux de capteurs sans fil." *Rapport de recherche, Institut De formation supérieure en informatique et communication IFSIC, Rennes* (2008).
- [20] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks, Article*, 38(4) :393–422, 2002.
- [21] Clément SAAD, « quelques contributions dans les réseaux de capteurs sans fil : localisation et routage », *Thèse de doctorat de l'université d'avignon, 2008*.
- [22] A.Ekici, Y.Gu, D.Bozdog, "Mobility-based communication in Wireless sensor Networks", *IEEE Communications Magazine*, 0163-0680/06, july 2006.
- [23] Y.W.Ma, J.L.Chen, Y.M.Huang, M.Y.Lee, "An Efficient Management System for Wireless Sensor Networks", *Journal Sensors* 10, 11400-11413; doi : 10.3390/s101211400, 2010.
- [24] Dr. Ovidiu Vermesan SINTEF, Norway, Dr. Peter FriessEU, Belgium, "Internet of Things– From Research and Innovation to Market Deployment", river publishers' series in communications, 2014.
- [25] *IoT Agenda*. [en ligne]. Disponible sur : < <http://internetofthingsagenda.techtarget.com/definition/microcontroller> > (Consulté le 5/04/2020).
- [26] Go tronic, <http://www.gotronic.fr/>, (Consulté le 8/04/2020).
- [27] <https://boutique.semageek.com/fr/2-arduino> , (Consulté le 8/04/2020)

- [28] <https://www.robot-maker.com/forum/tutorials/article/39-utiliser-une-plaque-dessai-ou-breadboard/>, (Consulté le 8/04/2020)
- [29] ARDUINO. PROTO SHIELD REV3 (UNO SIZE). [en ligne]. Disponible sur : < <https://store.arduino.cc/proto-shield-rev3-uno-size> >. (Consulté le 14/04/2020)
- [30] <https://www.linternaute.fr/dictionnaire/fr/definition/led/>, (Consulté le 15/04/2020)
- [31] Kaa, <http://www.kaaproject.org/what-is-iot/>, (Consulté le 20/04/2020).
- [32] Amazon web services, <https://aws.amazon.com/fr/iot/how-it-works/>, (Consulté le 20/04/2020).
- [33] Google Cloudplatform, <https://cloud.google.com/solutions/architecture/real-timestreamprocessing-iot>, (Consulté le 20/04/2020).
- [34] IBM Bluemix, <https://console.ng.bluemix.net/docs/services/IoT/iotplatform>(Consulté le 20/04/2020).
- [35] Miyuru Dayarathna, DZone, article, <https://dzone.com/articles/iot-software-platform-comparison>, (Consulté le 20/04/2020).
- [36] Les réseaux IoT, <https://www.connectwave.fr/techno-appli-iot/iot/les-reseaux-iot/>, (Consulté le 20/04/2020).
- [37] AL AGHA Khaldoun et ETIEMBLE Daniel. *Introduction à l'Internet des Objets*. l'université Paris Sud – Paris Saclay, 2018/08/10.
- [38] K. Mesmoudi, "Etude Expérimentale et Numérique de la Température et de l'Humidité de l'Air d'un Abri Serre Installé dans les Haut Plateaux d'Algérie, Région des Aurès," Thèse de Doctorat Physique Energétique, option énergétique Université de Batna, 2010.
- [39] <https://www.cmf-groupe.com/cmf-cultures/serres-verre/>, (Consulté le 20/04/2020).
- [40] Projet de fin d'étude d'un Système Automatisé « Cas d'une Serre » réalisé par Mr BENDIDANI Sohbi et Mr MILOUD ABID Aboubakr Essedik "Temouchent" (2017/2018).
- [41] Gupta, B. B., and Megha Quamara. "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols." *Concurrency and Computation: Practice and Experience* (2018): e4946.