الجمهورية الجزَائِرية الديمقرَاطية الشعبية

People's Democratic Republic of Algeria

وزَارة التعليم العَالي و البحث العلمي

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

جَامعة عمَار ثليجي بِالأَغوَاط

AMAR TELIDJI LAGHOUAT UNIVERSITY

كلية العلوم

FACULTY OF SCIENCE

DEPARTMENT OF MATHEMATICS AND INFORMATICS

MASTER THESIS

Domaine : Mathematics and Computer science
Filiere : Computer science
Option : Networks, Systems and Distributed Applications

BY : SACI ABDELHAMID

Theme

INTERNET OF THINGS SUPPORT TO SMART AGRICULTURE CASE OF DEPLOYMENT AND DATA COLLECTION

Supervised by : Cherroun Hadda

Academic Year 2021/2022

Abstract

Nowadays, the agriculture is living its digital transformation, Internet Of Things is one of technologies that have pushed the agriculture to this transformation, IoT provides many services to manage the critical agriculture phases, In this paper we address to design and implement an optimized IoT network support to smart agriculture. Its objective is to collect efficient data on weather inside a farm, IoT node deals with three different types of data: temperature, humidity and soil moisture. Our designed IoT relies on set of sensors and communication equipment that communicate using a single-hop architecture using MQTT protocol.

Key Words: Internet Of Things, IoT, WSN, Sensors, Smart Agriculture, Network, NodeMcu, Data Collection.

Résumé

Aujourd'hui, l'agriculture vit sa transformation digitale, l'Internet des Objets est l'une des technologies qui ont poussé l'agriculture à cette transformation, l'IoT fournit de nombreux services pour gérer les phases critiques de l'agriculture, Dans ce document, nous abordons la conception et la mise en œuvre d'un réseau IoT optimisé pour l'agriculture intelligente. Son objectif est de collecter des données efficaces sur les conditions météorologiques à l'intérieur d'une ferme, noeud IoT traite de trois types de données différents : la température, l'humidité et l'humidité du sol. Notre IoT conçu repose sur un ensemble de capteurs et d'équipements de communication qui communiquent à l'aide d'une architecture à mono saut utilisant le protocole MQTT.

Mots Clés: Internet des Objets, IoT, WSN, capteurs, Agriculture Intelligente, Réseau, NodeMcu, Collection de données.

الملخص

في الوقت الحاضر، تعيش الزراعة تحولها الرقمي، إنترنت الأشياء هي واحدة من التقنيات التي دفعت الزراعة إلى هذا التحول، توفر إنترنت الأشياء العديد من الخدمات لإدارة مراحل الزراعة الحرجة، في هذه الورقة نتناول تصميم وتنفيذ شبكة إنترنت الأشياء لدعم الزراعة الذكية. هدفها هو جمع بيانات فعالة عن الطقس داخل المزرعة، تتعامل عقدة إنترنت الأشياء مع ثلاثة أنواع مختلفة من البيانات: درجة الحرارة والرطوبة ورطوبة التربة. تعتمد تقنية إنترنت الأشياء المصممة لدينا على مجموعة من أجهزة الاستشعار ومعدات الاتصال التي تتواصل باستخدام بنية أحادية القفزة باستخدام بروتوكول.

Contents

List of Figures

List of Abbreviations

IoT Internet of Things

Leach Low Energy Adaptive Clustering Hierarchy

HEED Hybrid Energy Efficient Distributed

MCU Microcontroller Unit

MQTT Message Queuing Telemetry Transport

SNR Signal to Noise Ratio

WSN Wireless Sensor Network

T-MAC Time-out Medium Access Control

TDMA Time Division Multiple Access

CH Cluster Head

Introduction

Context

Healthy, sustainable and inclusive food systems are critical to achieve country's development goals. Agricultural development is one of the most powerful tools to end extreme poverty, boost shared prosperity, and feed a projected 9.7 billion people by 2050. Growth in the agriculture sector is two to four times more effective in raising incomes compared to other sectors. Agriculture is also crucial to economic growth. in 2018, it accounted for more than 25% of global gross domestic product (GDP) in some least developing countries [1].

But agriculture-driven growth, poverty reduction, and food security are at risk: Multiple shocks – from COVID-19 related disruptions to extreme weather, pests and conflicts – are impacting food systems, resulting in higher food prices and growing hunger.

Our research work is part of a sustainable project called *Dz-Smart-Agriculture* that aims harnessing the Algerian Agriculture by means of usage and deployment of advanced Information Technologies. This latter includes Internet of Things IoT , Machine Learning and its advanced techniques and sustainable energies.

The Project functionalities are illustrated in Figure 1. Its grantees are essentially farmers, agriculture actors: politics, leaders and researches in smart agriculture domains.

The goals of the project are to monitor crop health through different analysis, the main ones are:

1. Deployment of optimized IoT networks that can gather practical Data,
2. Creation of many largest Datasets from real in-sit environment,
3. improving the crops outcomes,
4. Providing farms by real practical information in order to well manage their farms outcomes,
5. Prediction of diseases,
6. Remote farms Monitoring and controlling,
7. Provide effective decision support tools for decision makers.

Our contribution in this substantial project is the involvement in the design of an optimized Internet of Things (IoT) that allows the data collection.

[1] https://www.worldbank.org/en/topic/agriculture/overview#1 seen on 08/05/2022

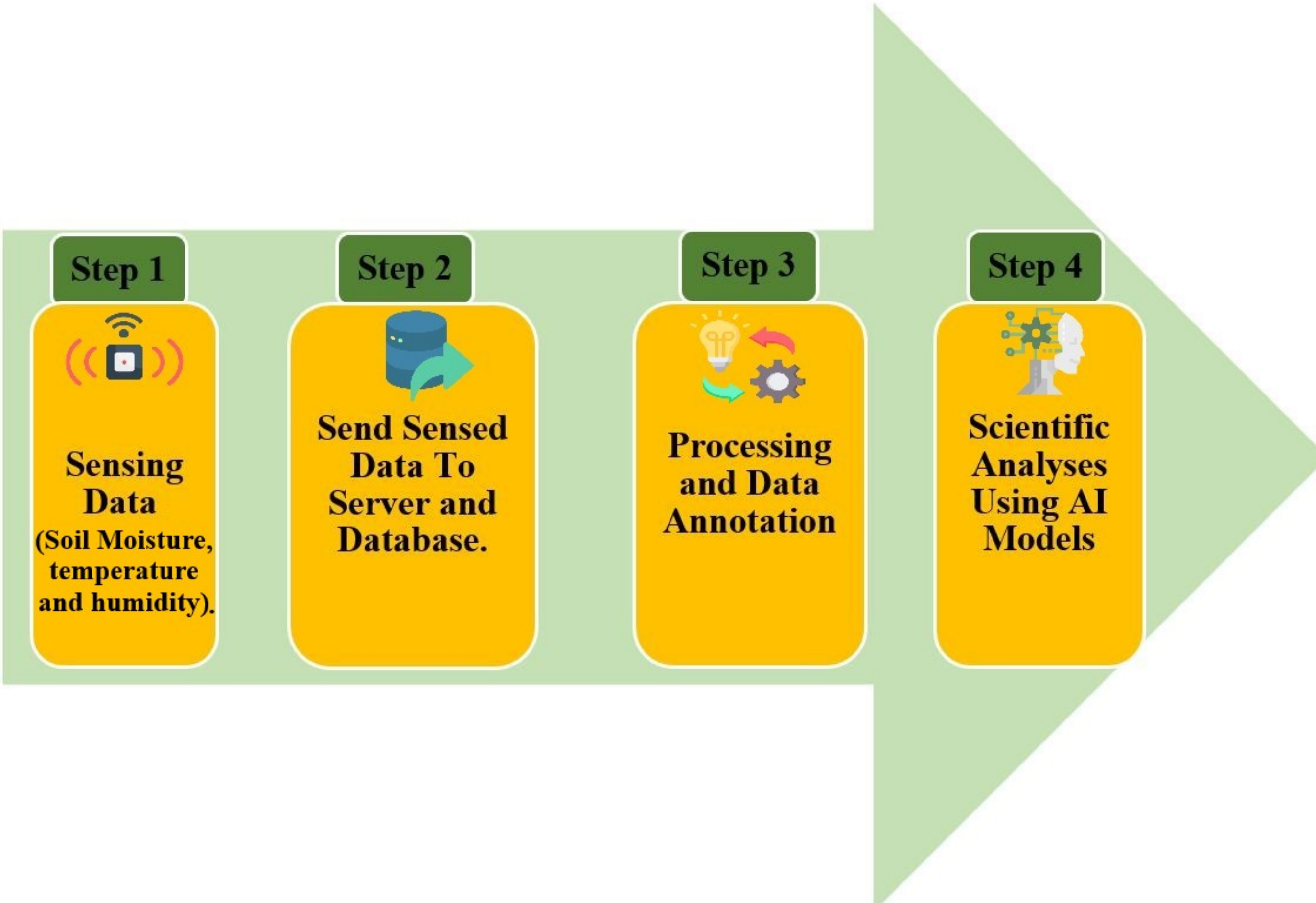


Figure 1: Main Steps of DZ-Smart-Agriculture

In our investigation, we will focus on the tuning of features and parameters to achieve efficient IoT that respects project constraints. In addition we will focus on collecting data related to weather information and those related to the irrigation.

In fact, we targeted an Olive Farm. Let us notice that the olive tree is resistant to drought, but responds greatly to the supply of water by any method. Well irrigated trees tend to produce higher yields, while the annoying phenomenon of Alternate Bearing can be mitigated through a rational and well-designed irrigation system. In general, olive trees that are cultivated for oil need less irrigation than those cultivated for table olives.

However for both simulation, full irrigation increases pumping costs, promotes unnecessary vegetative growth, can reduce flowering, and increases pruning costs [2].

Most olive tree diseases are caused by bacteria, viruses and fungi. These diseases are Crown Gal, Verticillium Cercospora, Olive fly, leaf spot, Peacock spots and Olive anthracnose.

[2] https://wikifarmer.com/olive-tree-water-requirements/

Thesis structure

The remaind of this manuscript is divided in four chapters.

1. The first chapter is dedicated to recall some background, concepts and preliminaries on IoT and WSN.

2. In the second chapter we presented a review on some related work.

3. In the third chapter we describe our proposed design in which we proposed architectures, topologies and protocols that can be used in this project.

4. The fourth chapter is dedicated for the implementation phase of the project. This chapter illustrates the chosen software, hardware and the scripting part.

Chapter 1

Generalities

In this introductory chapter we recall some concepts related to Internet Of Thing (IoT) and Smart Agriculture. First, we recall some definitions and preliminaries on IoT , their design, features and briefly their applications. Secondly, we focus deeply on the targeted domain of application so called Smart Agriculture.

1.1 Internet of Things conception and features

Internet of things (IoT) is one of the buzzwords in the IT industry. This technology aims to transform our real objects into intelligent virtual objects starting from your car until your teeth brush which make us not connected to Internet but surrounded by it giving us not only control of things around us, but also keeping us informed on the state of the things.

The Internet of Things (IoT) is the network of physical objects—devices, instruments, vehicles, buildings and other items embedded with electronics, circuits, software, sensors and network connectivity that enables these objects to collect and exchange data [1]. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency and accuracy. The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Kevin Ashton is a British technology pioneer who is known for inventing the term "the Internet of Things" to describe a system where the Internet is connected to the physical world via ubiquitous sensors. IoT is able to interact without human intervention.

Some preliminary IoT applications have been already developed in healthcare, transportation, and automotive industries as shown in figure 1.2. IoT technologies are at their infant stages; however, many new developments have occurred in the integration of objects with sensors in the Internet. The development of IoT involves many issues such as infrastructure, communications, interfaces, protocols, and standards. The objective of this paper is to give general concept of IoT , the architecture and layers in IoT , some basic terms associated with it and the services provided [1].

The design of an IoT architecture involves many factors such as networking, communication, processes etc. In designing the architecture of IoT , the extensibility, scalability, and operability among devices should be taken into consideration. Due to the fact that things may move and need to interact with others in real-time mode. A critical requirement of an IoT is that the things in the network must be connected to each other. IoT system architecture must guarantee the operations of , which connects the physical and the virtual worlds [1].

In general IoT networks are service driven, we should talk about designing a Service Oriented Architecture.

1.1.1 Service oriented architecture

The Service Oriented Architecture (SOA) treats a complex system as a set of well-defined simple objects or subsystems. Those objects or subsystems can be reused and are maintained individually. Therefore, the software and hardware components in an IoT can be reused and upgraded efficiently. Due to these advantages, SOA has been widely applied as a mainstream architecture [1]. Figure 1.1 illustrates the four layers of an SOA.

Sensing layer

IoT is expected to be a wide spread physical inner-connected network, in which things are connected continuously and can be controlled from anywhere. In the sensing layer, the smart systems on tags or sensors are able to automatically sense the environment and exchange data among devices. Things can be uniquely identified and the surrounding environments can be monitored for various purposes and applications. Every object in IoT holds a digital identity and can be easily tracked in the digital domain. The technique of assigned unique identity to an object is called a universal unique identifier (UUID). The identifiers might contain names and addresses. A UUID is a 128-bit number used to uniquely identify some object or entity on the Internet [1].

Network layer

The network layer in IoT , connects all things and allows them to be aware of their surroundings. Via the network layer, things can share data with the connected things, which is crucial to intelligent event management and processing in IoT . For the sharing of data and to provide services by a device a strong network is essential. The network should also automatically discover and map things. Things need to be assigned roles automatically to deploy, manage, and schedule the behavior of things and should be able to switch to any roles at any time as required. This enables devices to perform tasks collaboratively [1].

Service layer

Service layer enables the services and applications in IoT . It is a cost effective platform where software and hardware can be reused. The services in the service layer run directly on the network to effectively locate new services for an application and retrieve data dynamically about services. Most of specifications are undertaken by various standards developed by different organizations. A universally accepted service layer is important for IoT . A practical service layer consists of a minimum set of applications, application programming interfaces (APIs), and protocols supporting required applications and services. All of the service-oriented activities, such

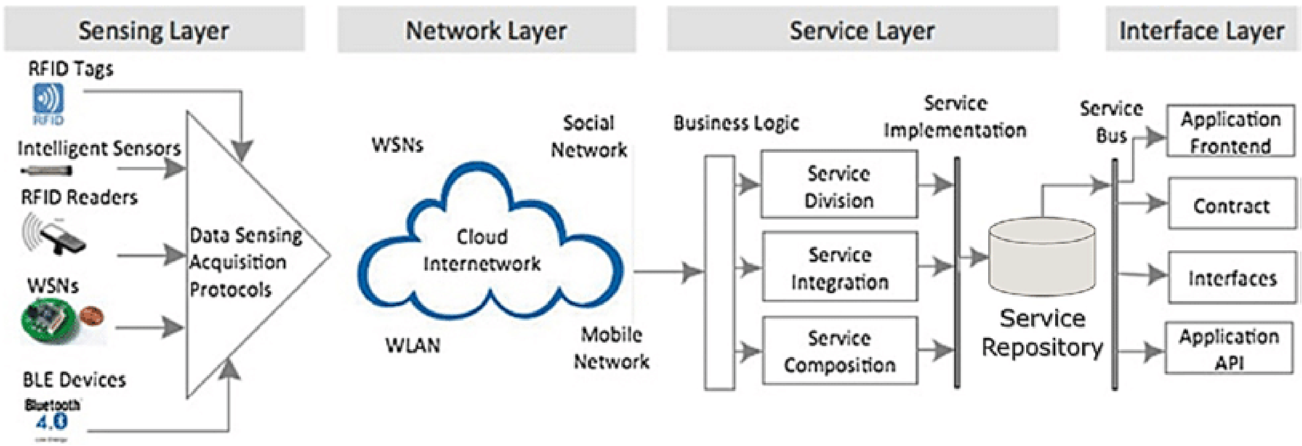


Figure 1.1: SOA Layers [1]

as information exchanging and storage, management of data, search engines and communication, are performed at the service layer [1].

Interface layer

In IoT , a large number of devices are connected. These devices belong to different people and hence do not always imply with the same standards. The compatibility issue among the things must be solved for the interaction among things. Compatibility involves in information exchanging, communication and events processing. There is a strong need for an effective interface mechanism to simplify the management and interconnection of things. Basically interface layer works in the application frontend or API (Application Program Interface) [1].

1.2 IoT applications

There are many possibilities for having interconnected "things" that can interact with each other over the Internet. IoT can be used for all types of applications ranging from connecting all the devices in your house to create a smart home or even connecting all the government and civic services in a city to create a smart city so we can say that the IoT applications are endless. See Figure 1.2. We can cite some of them [13] hereafter.

- **Smart home** Nowadays homes and offices use IoT technologies. Various electronic gadgets and heating, ventilation, and air conditioning systems such as lights, fans microwave ovens, refrigerators, heaters and air conditioners are embedded with sensors and actuators to utilize the energy sufficiently, monitor and control amount of heating, cooling and level of light, room light sense the presence of human beings. They turn on when you enter, when fire or smoke detected at home, wireless smoke and carbon monoxide sensors sound alarms and also alert by phone or email and adds more comfort in life, which in turn minimize the cost and increases energy saving [14].
- **Smart city** On a broader scale, IoT technologies can be employed to make cities more efficient. The goal of smart cities is to leverage the IoT to improve the lives of citizens by improving

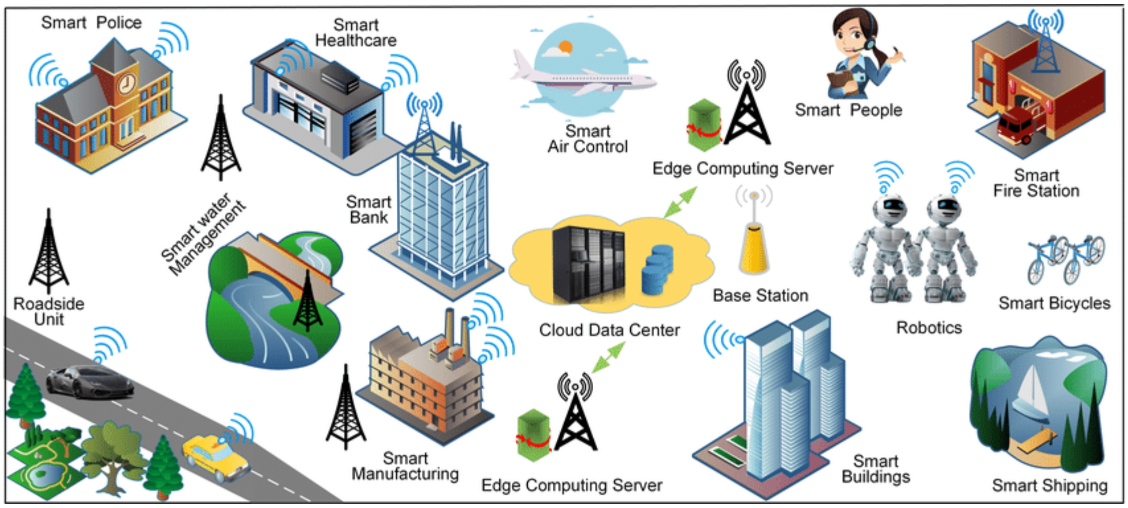


Figure 1.2: IoT applications [2]

traffic control, monitoring the availability of parking spaces, evaluating air quality and even providing notification when trash containers are full[14].

- **Health tracking** The IoT is used in healthcare domain to improve the quality of human life by assisting basic tasks that humans must perform through application. Sensors can be placed on health monitoring equipment used by patients. The information collected by these sensors is made available on the Internet to doctors, family members and other interested parties in order to improve treatment and responsiveness. Additionally, IoT devices can be used to monitor a patient's current medicines and evaluate the risk of new medications in terms of allergic reactions and adverse interactions. With the use of sensors and the technology stated above we can track the person's body temperature, heart beat rate, blood pressure, etc. In case of emergency, the individual and their personal doctor will be notified with all the data collected by the sensors. This system will be very useful to senior citizens and disabled people who live independently [14].

- **Supply-chains** Internet of Things monitors every stage of supply chain from purchasing of raw materials from the suppliers by the manufactures, production, distribution, storage, product sales and after sales services. This will help to maintain the stock required for continuous sale, which in turn results in customer satisfaction and increased sales. According to Cisco's economic analysis, IoT will generate 1.9 trillion dollar from supply chain and logistics over the next decade. By using this we can also diagnose if the machines require repair and maintenance [14].

- **Natural disaster monitoring** Natural disasters such as earth quake, landslides, forest fire, volcanoes, flood, etc. can be predicted by using wireless detection sensors. These detection intimate the respective authorities to take the precautions before the disaster occurs[14].

Due to the importance of Smart Agriculture in this current work we will give more details about it in the following section.

1.3 Smart agriculture

Figure 1.3: Smart Agriculture [3]

In the field of smart agriculture IoT provides a wide range of applications such as soil and plant tracking, crop growth observation, and selection, assistance for irrigation assessment, and monitoring of the agriculture environment. In smart agriculture Internet of Things (IoT) technology is applied in diagnostics and control. To optimize agriculture, the implementation of IoT in the field has increased the productivity and effectiveness of farmers. It may help to determine field variables such as soil quality and plant biomass. It can also be used to test and monitor variables including temperature, soil moisture, and crop diseases. Besides, IoT can be used to track crop growth and yield influencing factors. Farmers can also figure out which crops are most suited for which conditions and can rotate crop accordingly .

IoT applications support farmers during crop planting, irrigation, crop processing, harvesting and post harvest, crop storage and transportation, and many other benefits in agricultural IoT systems. Soil moisture sensors, humidity sensors, leaf moisture sensors, solar radiation sensors, infrared light sensors, and rainfall predictors are among the field sensors used in IoT -based systems. In IoT scenarios, sensors can be installed in a variety of locations, including greenhouses, seed banks, cold rooms, agricultural machinery, transportation systems, and livestock, and the data collected can be processed in the cloud for monitoring and control [6].

In this chapter, we reported some concepts and background necessary to our investigations mainly IoT and their applications.

Chapter 2

Related work

This section provides a thorough review of the related literature to the deployment of IoT and WSN as a support to smart agriculture. In this review we have chosen three papers that dealt with three (see figure 2.1) categories of applications: Data Collection, Monitoring/Controlling, Diseases Prediction.

2.1 Disease prediction

The authors proposed a smart agriculture system to provide information on soil moisture, humidity and temperature of the potato field [4]. To monitor these critical factors, they instrumented a potato field with wireless sensors. This monitoring can predict the risk of developing of phytophthora disease which make it easier for farmers to take appropriate decisions to treat the field, or parts of it, with fungicide only when absolutely needed.

Their global deployed architecture ,under the project LOFAR Agro,[1] is illustrated in Figure 2.2. In this paper researchers used sensor nodes in the field to measure relative humidity (RH) and temperature (T) once per minute. Ten samples are encoded in a single packet, which is send – directly, or through multiple hops to a gateway at the edge of the field. The gateway uses standard Wi-Fi to forward the sensor data over the LOFAR backbone network to the Agro server, which logs the data, filters out erroneous readings, and hands the accumulated data to the Phytophthora decision support system (DSS) server. This decision support system combines the field data with a detailed weather forecast to determine the treatment policy, which is then provided to the farmer.The normal farming routine requires that a treatment advice is available every morning when the activities for the day are planned.

The sensor node is based on the TNOde platform3. A TNOde together with a battery is packed into a waterproof PVC-based casing see figure 2.3. The 7 cm antenna is placed on top and the micro-climate sensor is connected by a cable (I2C) at the bottom. Experiences has shown that the radio range is dramatically reduced when the potato crop is flowering and leaves cover the (ground-based) antennas. Therefore the nodes are installed on poles at a height of 75 cm. The potato plants grow to about 100 cm, however, we had to include a safety margin to ensure that the nodes could not be hit by farming equipment attached to a tractor. The cost of a node is about 250

[1] http://www.lofar.org/p/Agriculture.html

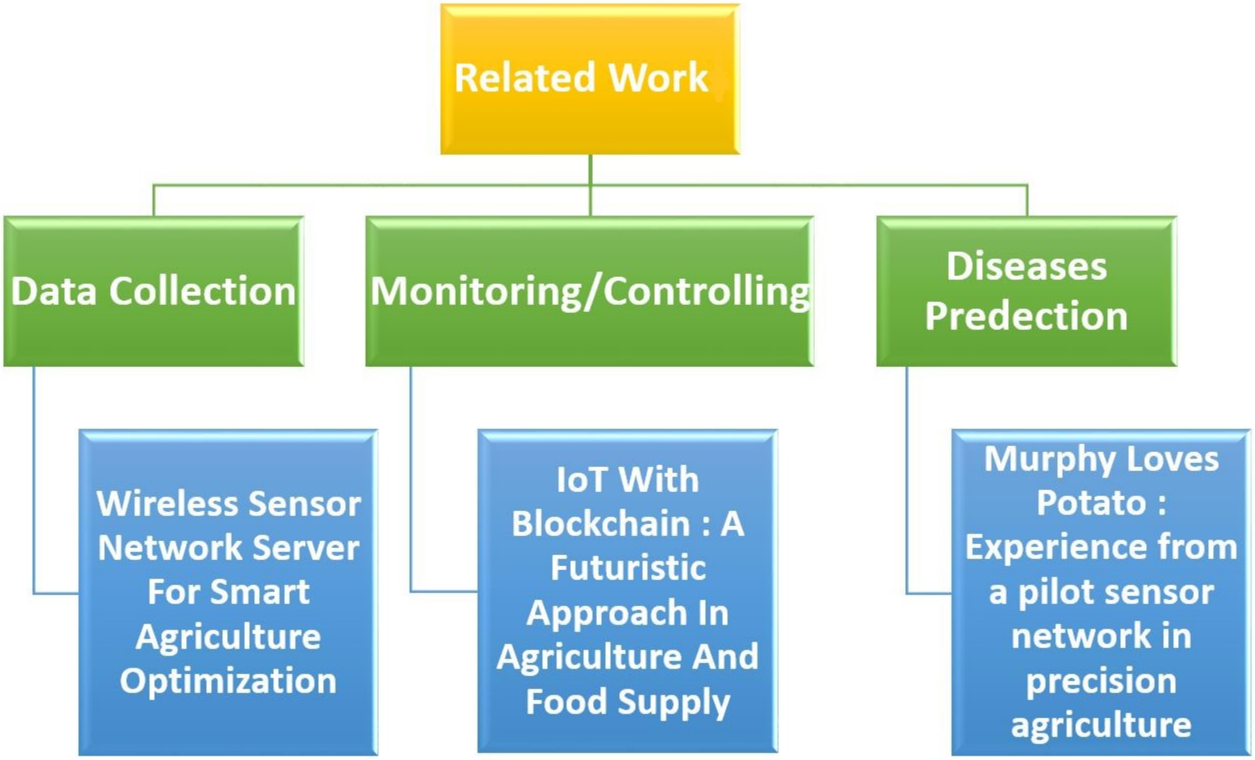


Figure 2.1: Related Work.

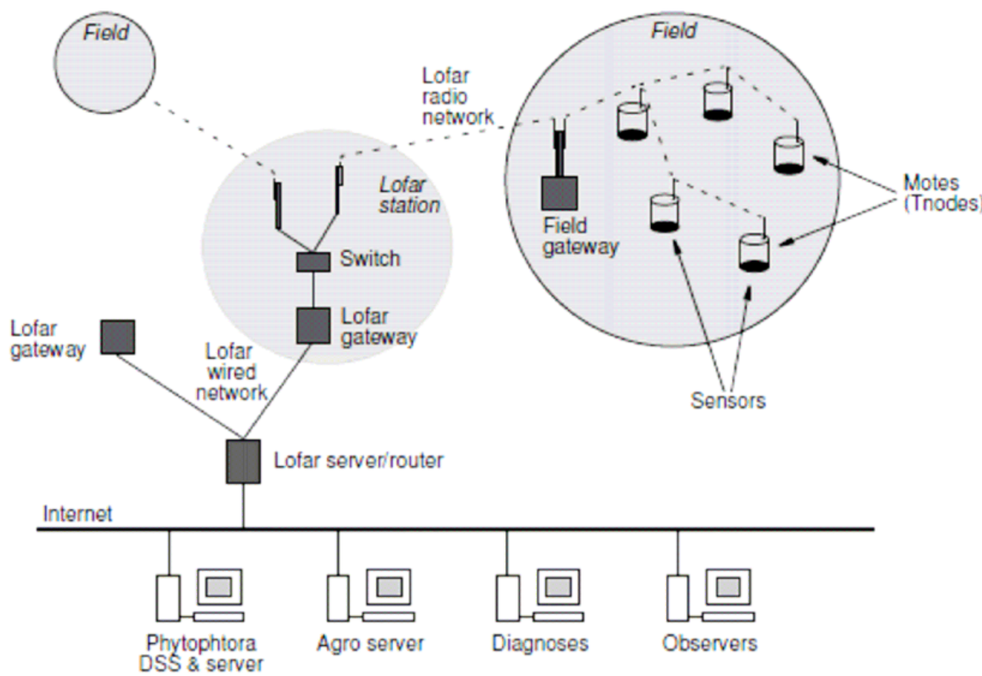


Figure 2.2: LOFAR-agro setup [4]

CPU	ATmega128L (8-bit, 8MHz) 128KB Flash memory 4KB DRAM memory
Storage	4Mbit EEPROM memory
Radio	Chipcon CC1000 868 MHz, FSK 76.8 kbps, output power -20 to 5 dBm
I/O	RS232, 3 LEDs 5 general purpose I/O pins
Sensor	Sensirion SHT75 T $\pm 0.3\,^{\circ}C$, RH $\pm 1.8\%$
Battery	Sonnenschein SL-770/T C-cell, 3.6V, 7.2Ah

Figure 2.3: LOFAR-agro TNode performance [4].

euro.The gateway at the edge of the field is based on a Star gate from Crossbow , which is equipped with a 400 MHz X-Scale processor, a 256MB Compact Flash card for backup storage, a PCMCIA Wi-Fi card connecting to the LOFAR backbone network, and one TNOde communicating with the sensor nodes. This TNOde is connected to a five-meter-high antenna to ensure that most nodes in the field can be reached directly. The gateway is powered by a solar panel in combination with a rechargeable battery to ensure operation around the clock. The total cost of the gateway is around 1500 euro. All TNOdes run a customized TinyOS version which handles the minor differences between the Mica2 and TNOdes hardware layout. The image for the field sensors includes the following components: T-MAC For the medium access control layer. T-MAC is tightly integrated with the power management component of TinyOS and puts the radio and CPU to sleep whenever there is no traffic to send or receive.

2.2 Data collection

Ardiansyah et al. has designed a system for data collection based on wireless sensor network. This system can be manifested by designing a WSN server. This WSN server used to collect data from sensors and store it in a database [5].

The researchers developed their wireless sensor network on 6 phases as shown in Figure 2.4. To perform this network they used the following hardware :

- **NodeMCU-ESP8266** Micro controller to process sensed data .
- **Soil moisture Sensor YL-69** It will sens the data and send it to the microcontroller

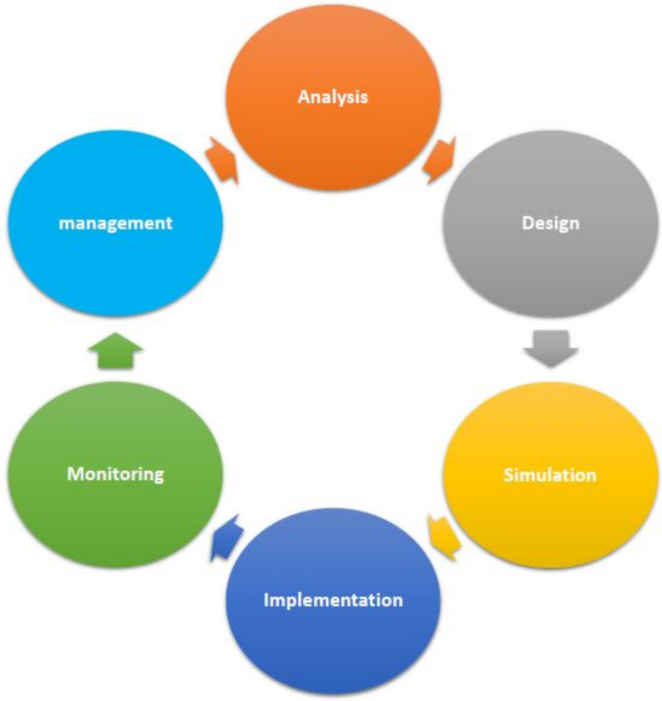


Figure 2.4: Network developments life cycle.

- **Connection** For the connection they used a bread-board and some jumper cables to connect all the components as shown in figure 2.5 [5].

To program the NodeMcu to read the senT data from the sensor and communicate correctly with the database they used C/C++ programming languages and the Arduino interface. The resulted model design of the WSN is shown in Figure 2.6 [5].

The WSN server model for smart farming optimization using moisture sensors and an IoT platform to send data from the sensors to the server. The sensed data can be consulted from the website trough internet connection. The data in the server can be accessed remotely and provide a real time data from the field. The sensed data shows a promising readings [5].

2.3 Monitoring and controlling

In this paper the authors combined between the blockchain technology and the IoT to create a smart model to reorganize the food production and supply chain starting by the farm ending to the client and including all the stake holders as shown in Figure 2.7.In this paper we are interested only by the IoT architecture [6].

IoT-Based Agriculture Protocol for the Smart Model

IoT nodes are ideal for cluster farms because they doesn't consume too much energy and can be further reduced through an efficient clustering protocol. Therefore, this research proposed a new clustering protocol IoT-based agriculture [6].

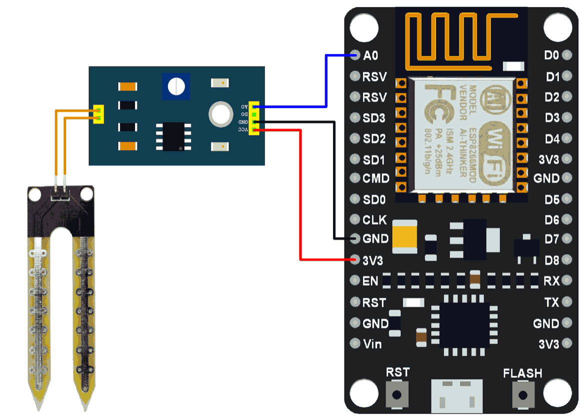


Figure 2.5: NodeMcu and soil moisture sensor connection [5].

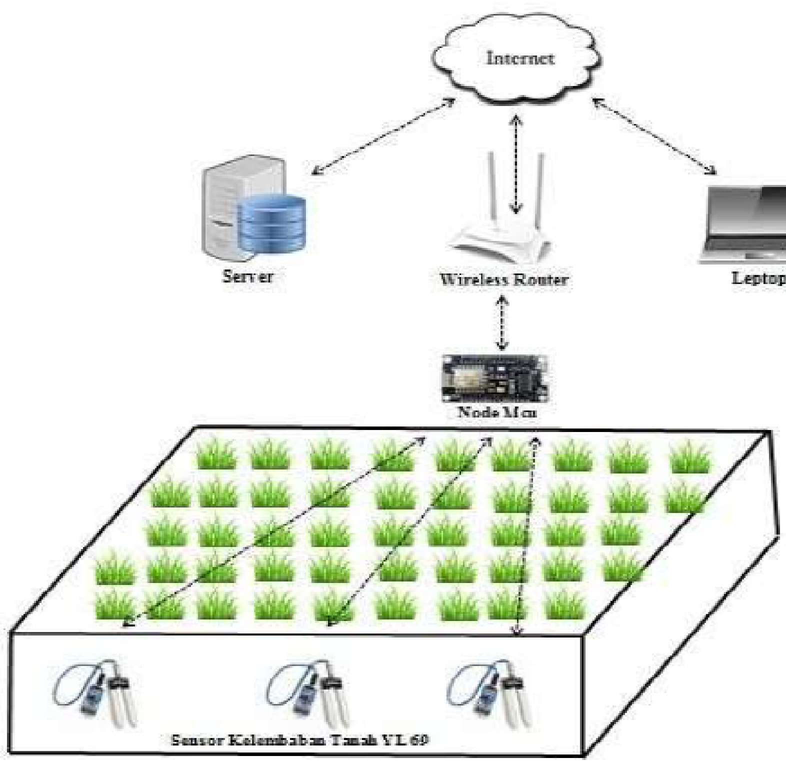


Figure 2.6: WSN Design [5].

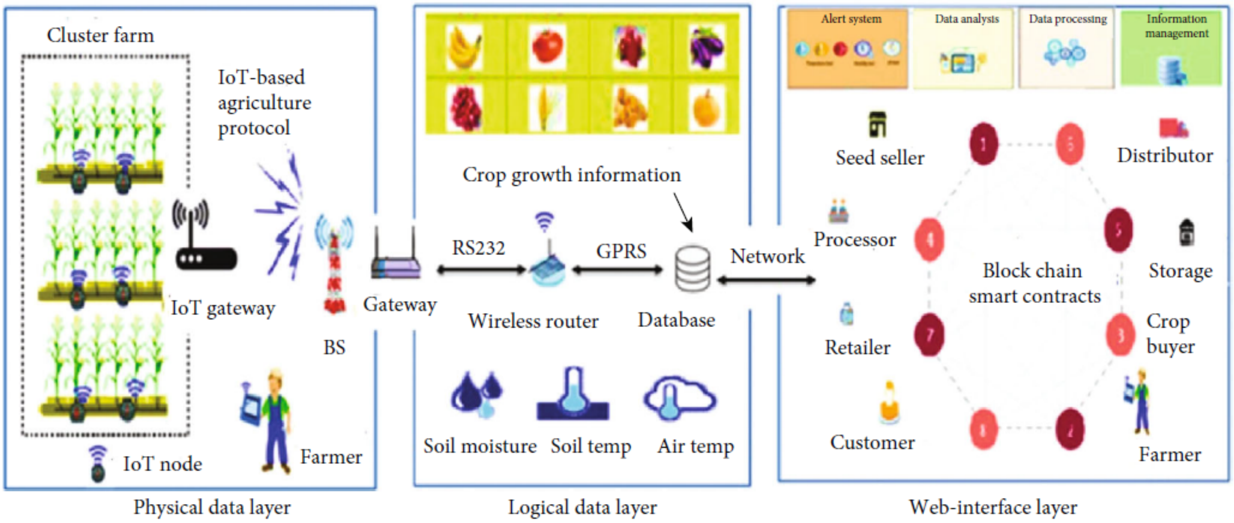


Figure 2.7: IoT and blockchain smart model [6].

Clustering mechanism

(i) IoT nodes are installed on a cluster farm randomly.

(ii) IoT nodes send hello messages to the base station. with local information.

(iii) The initial number of clusters is calculated by taking the optimal values to vary with the node density before the node starts to expire, and the smaller clusters become larger clusters.

(iv) The base station and sink are installed outside the cluster farm as presented in Figure 2.8.

IoT nodes are divided into groups in a cluster farm, and these groups are known as clusters; each cluster has a head node. Each cluster's nodes sense data and send it to a single head node; they do not communicate with other head nodes [6].

In this paper, they have provided complete solution for agriculture and food supply chain by integrating IoT with the blockchain. In addition, they have developed an energy efficient routing protocol for the proposed system to extend system life by reducing energy consumption. A novel approach has been applied in this research and developed a futuristic smart model for agriculture and the food supply chain, which offers an innovative way for farmers to acquire information on crops. IoT can provide farmers with information on crop yields, soil temperature, pest infestation, and soil quality that is essential for high crop production and provides precise data that can be used to improve farming techniques [6].

2.4 Summary

To summarize all the previous reviews that we've seen in this chapter we have to cover these important points: Deployed protocols, sensed data and costs.
Table 2.1 summaries the reviewed work.

- **Deployed protocols** One of the most important factors in the efficiency of any network is protocols we can see that clearly their role in the previous papers in terms of latency, number

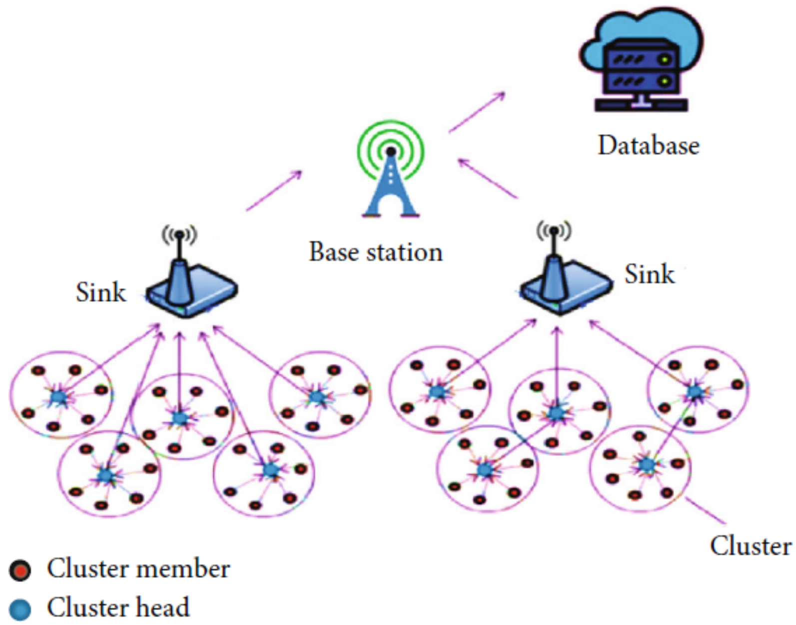


Figure 2.8: IoT-based agriculture clustering protocol [6].

Work	Targeted Application	Deployed Protocoles	Size	Sensed Data	Cost
Langendoen et al. [4]	Diseases Prediction	T-MAC	Medium	Soil Moisture, Temperature, humidity	28000.00 euro +
Ardiansyah et al.[5]	Data Collection	HTTP	Small	Soil Moisture	/
Awan et al.[6]	Monitoring and Controlling	Iot-based leach	Large	Soil Moisture, Temperature, humidity plus other values	/

Table 2.1: Related Work Summary.

of transmissions and the most important one network lifetime. In paper [6] researchers used their improved version of leech (Low Energy Adaptive Clustering Hierarchy) protocol what increased the network stability and decreased the power consumption mostly into the half. As they have a big density of IoT node they had the possibility to use Heed (Hybrid Energy Efficient Distributed) protocol in steed of leech which can offer them more covered zones and it can save power by shorting the distances of transmissions.

In paper [5] authors doesn't mentioned the protocols used for the communication; if they used the NodeMcu by their default configuration so they are using HTTP protocol, which is not very bad choice, but they were able to use other lighter protocols dedicated to IoT to get better results.

In paper [4] Langendoen et al. Used the T-MAC(Time-out MAC) protocol which a very simple and efficient protocol.The novel idea of this protocol is to reduce idle listening by transmitting all messages in bursts of variable length, and sleeping between bursts.

- **Sensed data** We can see from these reviews that there is a consensus on the sensed data for agriculture applications which are soil moisture, temperature and humidity because they are the basic factors needed in these applications.
- **Costs** IoT systems have various costs from the cheap to the very expensive as the WSN used in paper [4]. The cost is not the point , we can have an IoT system with reasonable cost, the point is how we do to choose the right architecture for the system.

From these reviews we can see clearly the important role IoT and WSN play in the smart Agriculture and precision agriculture especially in increasing the quantity and the quality of the crop, and how the right technical choices such as the used architecture and the protocols can affect the system performance.

2.5 Conclusion

In this chapter we have reviewed some work that have dealt with the deployment of IoT as support for some smart agriculture applications mainly dedicated to remote monitoring, data collection and disease prevision. The lessons learned from this modest literature review are numerous. First, the deployed protocols are chosen from those keeping energy. Recently the Blockchain paradigms are deployed to ensure security and data durability. Furthermore, the cost of such IoT is very low compared to hard solutions.

Chapter 3

Conception of the targeted IoT

In this chapter, we will present our approach that deals with the design and deployment of IoT based network for a targeted smart agriculture application.

In fact this network can help the agriculture in plant tracking, irrigation assistance, decision making and we can also use the generated/collected database from the IoT as an input for many a smart system.

3.1 Motivation

Let us recall that the main aim of this project is to deploy IoT network that can help the farmers in plant tracking, irrigation assistance, decision making. furthermore, we can also use the generated/collected database from the IoT as an input for many smart systems, smart systems based on machine learning. They can predict the possible fungal diseases that can be caused by the weather changes and the irrigation program, to protect the crop from fungal's spread and also to decrease the used amount of fungicides to get a more secure product for the consumer.

3.2 Global view of the proposed architectures

The sensors are deployed according to some standards chosen by the agriculture expert like wind, sun exposure...,that's why we have to choose the right topology in order to achieve a satisfying result by decreasing the collisions and latency and saving batteries to increase the sensor life for the most long time possible.

In order to guarantee the right connectivity, communication and the efficiency of our wireless sensor network we investigate two architectures: single-hop and multi-hop.

3.2.1 Single-hop architecture

We have proposed a single-hop star topology because it can give our network a higher performance in terms of lifetime, less transmissions and low latency [6].

IoT nodes are divided into groups in the farm. These groups are known as clusters; each cluster has a head node. Each cluster's nodes sense data and send it to a single head node. They

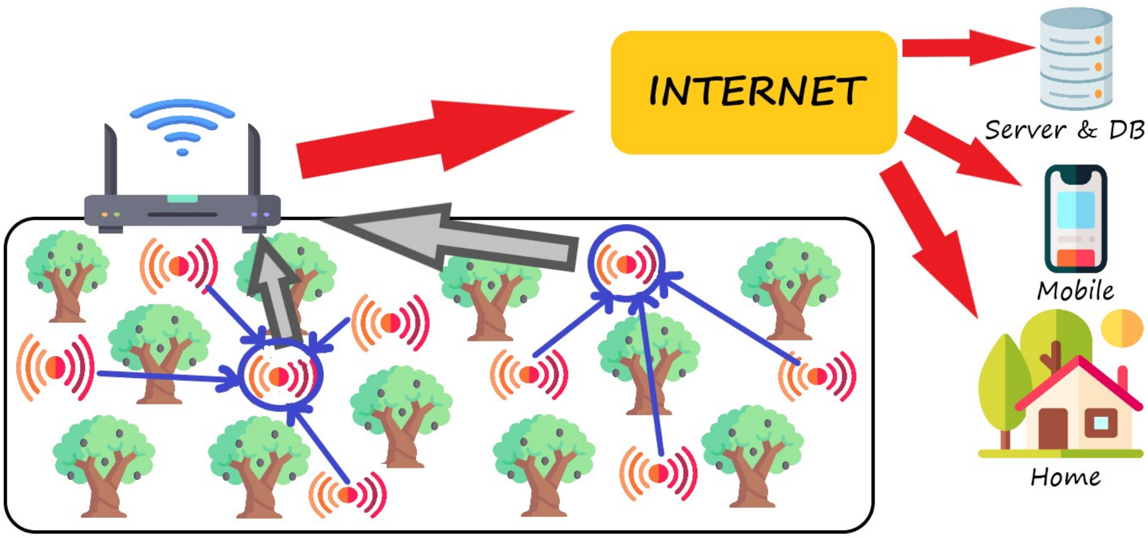


Figure 3.1: Proposed Single-hop star topology.

do not communicate with other head nodes. Nodes in clusters are distributed in such a way that one cluster node does not communicate with other cluster nodes, but only with the cluster heads of their respective clusters. Each cluster's Cluster Head (CH) is in charge of sharing sensed data with the gateway to achieve a satisfactory SNR. One of the most famous protocols used in such architectures is Leach [6].

Leach

Leach protocol is a TDMA based MAC protocol. The main aim of this protocol is to improve the lifetime of WSN by saving energy. Leach protocol consists of two phases: Set-up phase and Steady phase. Leach protocol is a typically a representation of hierarchical routing protocol. It is self-adaptive and self-organized. Leach protocol uses round as unit, each round is made up of cluster set-up stage and steady state storage for the purpose of reducing unnecessary energy costs [15].

Here are some important phases of Leach protocol:

- **Cluster Head (CH) selection** The CH is responsible for collecting data from member nodes and transmitting aggregated data to the gateway. It requires a lot of energy to do so, and the transmission process must be boosted with high power amplification. Two parameters are considered in the CH selection process: the history of nodes acting as CH and the optimal percentage of a node. The generation of a random number is used to make each node decision (between 0 and 1). The node will be selected as CH if the generated random number is threshold $T(n)$ for that round .

$$T(n) = \begin{cases} \frac{P}{1-p*(r \mod \frac{1}{P})} & n \in G \\ 0 & Otherwise \end{cases} \}$$

Where p represents the percentage of CH, r represents the number of rounds, and G represents the set of nodes that were not selected as CH in the previous $1/p$ rounds. Every node in the cluster has a $1/p$ chance of being a CH in each round in this situation. When a node is chosen as CH, it sends an advertisement message to its nearby nodes, inviting them to be CH. The advertisement message is accepted by the member nodes, and they enter the CH [6].

- **Data Transmission** Following CH selection, the data transmission schedule kicks off, with member nodes sending data to their assigned CH during their designated transmission time. Low-energy transmission was required for this type of transmission. Before transmission time is allocated, the member node can be turned off to save energy. The CH must keep the receiver to receive the full data and then combine all of the data into a single signal before transmitting it [6].

3.2.2 Multi-hop architecture

As shown in Figure 3.3 in multi-hop Wireless Networks there are one or more intermediate nodes along the path that receive and forward packets via wireless links [16].

Multi-hop wireless networks have several benefits: Compared to networks with single wireless link, multi-hop wireless networks can extend the coverage of a network and improve connectivity. Moreover, transmission over multiple “short” links might require less transmission power and energy than over “long” links. In addition, they enable higher data rates resulting in higher throughput and more efficient use of the wireless medium. Multi-hop wireless networks avoid wide deployment of cables and can be deployed in a cost-efficient way. In case of dense multi-hop networks several paths might become available that can be used to increase robustness of the network [16].

In wireless multi-hop networks, nodes communicate with each other using wireless channels and do not have the need for common infrastructure or centralized control. Nodes may cooperate with each other by forwarding or relaying each others’ packets, possibly involving many intermediate relay nodes. This enables nodes that cannot hear each other directly to communicate over intermediate relies without increasing transmission power. Such multi-hop relaying is a very promising solution for increasing throughput and providing coverage for a large physical area. By using several intermediate nodes, the sender can reduce transmission power thus limiting interference effects and enabling spatial reuse of frequency bands [16].

One of the very used protocols with this architecture is Hybrid energy efficient distributed (HEED) protocol. HEED protocol is made for increasing the network lifetime in IoT networks. This protocol considers two parameters for deciding the cluster heads, residual energy and node density see Figure 3.2 .

3.2.3 MQTT protocol

For the communication between the cluster head, the gateway and the database we are using the MQTT (Message Queuing Telemetry Transport) protocol. It is specifically designed for ”machine to machine” communication. MQTT protocol runs over TCP/IP and has a data packet size with low overhead (minimum 2 bytes) so that consumption of the power supply is also small enough. This protocol is a data-agnostic protocol that can transmit data in various forms such as

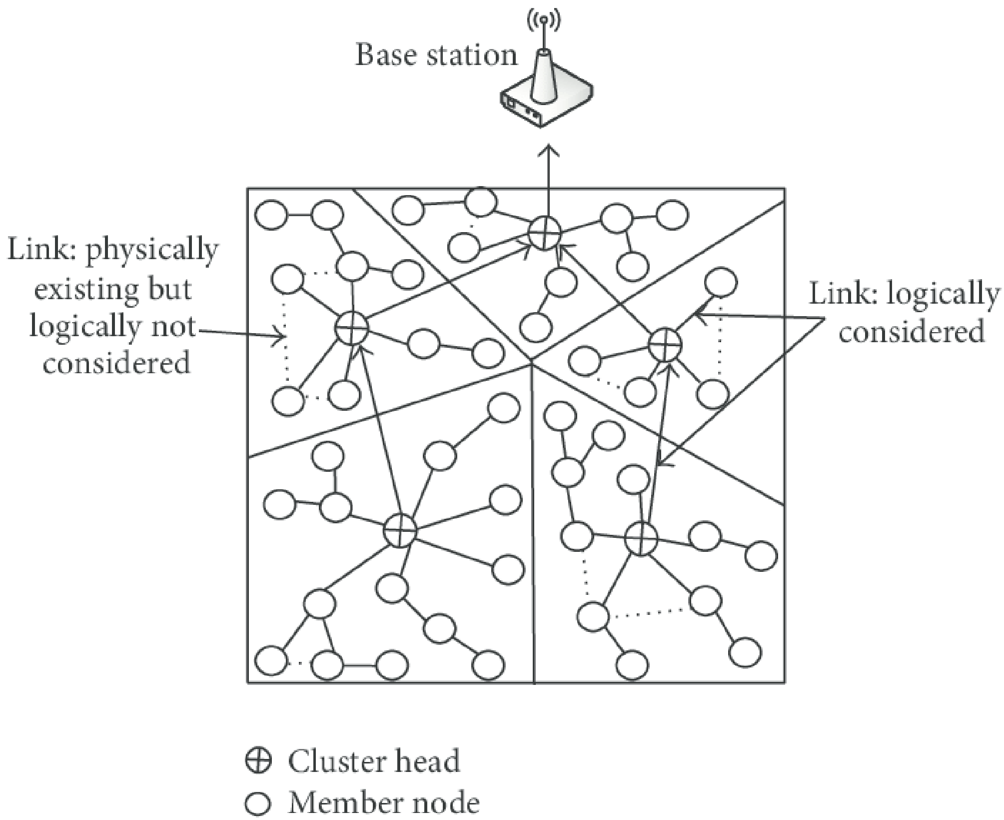


Figure 3.2: Heed Protocol [7]

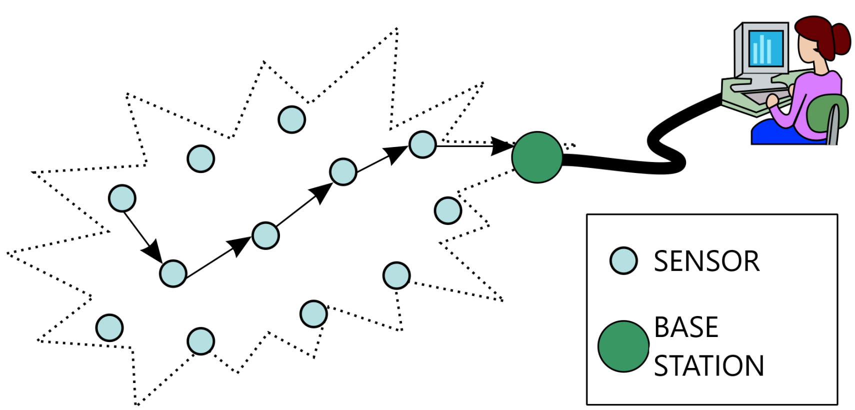


Figure 3.3: Multi-hop communication [8]

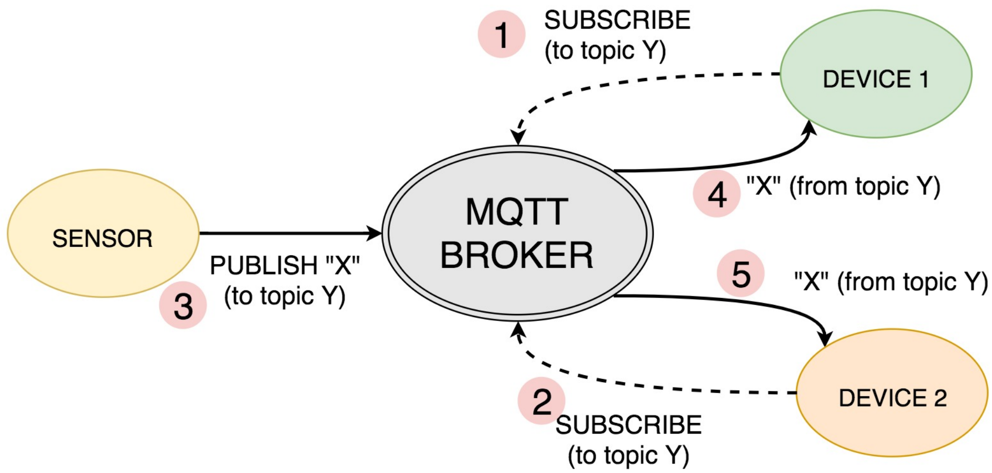


Figure 3.4: MQTT Architecture [9].

binary data, text, XML, or JSON and this protocol uses a publish/subscribe model rather than a client-server model [17].

MQTT requires two main software components:

- **MQTT Client** is to be installed on device.
- **MQTT Broker** serves to handle publish and subscribe data.

MQTT has 14 types of control signal, namely:

- **CONNECT**:Client request to connect to Server
- **CONNACK**:Connection Acknowledgement
- **PUBLISH**:A message which represents a new/separate publish
- **PUBACK**:QoS 1 Response to a PUBLISH message
- **PUBREC**:First part of QoS 2 message flow
- **PUBREL**:Second part of QoS 2 message flow
- **PUBCOMP**:Last part of the QoS 2 message flow

From those signals, there are only four main signals which are used directly by the client, namely PUBLISH, SUBSCRIBE, UNSUBSCRIBE, CONNECT. Other signals are part of the publish/subscribe mechanism [17].

3.3 Constraints & choices justification

3.1 reports a comparison between both proposed architectures. The salient observation is that single-hop based architecture will be efficient according to power consumption and cost.

Architecture	Network Density	Number of Transmissions	Protocol	Power Consumption	Cost
Single-Hop	Medium	Low	Leach	Low	Low
Multi-Hop	Very High	High	Heed	Very Low	High

Table 3.1: Single-Hop based Architecture Vs. Multi-Hop one.

3.3.1 Constraints

Before taking a decision about the adopted architecture, let us first recall some constraints that we have to respect. First, the allowed budget for the realisation of such IoT network is limited. In fact, if we aim that most farmers use such realisation to monitor their farms, the related cost need to be in the range of their generally low budgets.

In addition, the Algerian Market concerning those dedicated electronics has its limits. On one hand they do not provide specialized materiel with high precision. And on the other hand we are dealing with startups and little stores because there is no specialised suppliers.

3.3.2 Justification

In order to perform our IoT system we need to choose the suitable architecture, topology and protocols to achieve the targeted efficiency and performances.

In order to decide which is the right choice we need to discuss the following factors:

- **Covered zone** When we compare the Single-Hop and Multi-Hop architectures we find that the Multi-Hop architecture is dedicated to cover larger zones than the first one, because nodes in this topology can generate a logical link between the transmitter and the receiver passing through other nodes. While in Single-Hop architecture the covered zone is limited by the communication range of nodes.

- **Network lifetime** This is the most important factor which can determine the used architecture. When we consider the Multi-Hop Architecture we find that the power consumption is very low because the distance between IoT nodes is very short so they can pass the packets to each others until they achieve the final destination which is the CH, sink or base station. This fact requires networks with a very high density (high cost) and very high number of transmissions in stead of the Single-Hop architecture the power consumption is a little higher than the Multi-Hop architecture because the distance between the IoT nodes and the final destination is not short where the transmitter must be achieve the destination in one hop without passing by intermediate nodes what makes the distance longer. So the nodes needs more power to transmit their packets but the good point in this architecture that it reduces the number of transmissions and we can use it without the need to very big budgets because it doesn't need to high network density.

- **Number of transmissions** In Single-Hop architecture the number of transmissions is very low because there is no need to pass by intermediate IoT nodes what avoid the network

10	2F	00	04	M	Q	T	T	04	02	00	3C	00	23	token

Figure 3.5: Connection packet

10	02	flag	flag

Figure 3.6: Connection ACK Packet

to get saturated. Conversely,the Multi-Hop architecture needs to a very big number of transmissions what can cause a network saturation.

- **Latency** The latency in Multi-hop is much Higher than Single-Hop, packets must pass by intermediate nodes. Conversely, in Single-Hop packet have to pass directly to its final destination.
- **Cost** This factor has a direct relation with the density of the network whenever it increases the cost increase. In the proposed architectures we find that the Multi-Hop architecture is more dense than the Single-Hop architecture which makes the latter cheaper than its concurrent.

3.3.3 Theoretical measurements

In this section, we give some measurements on packets sizes.

Let n be the average number of connection attempts (n depends completely on the quality of the Internet), S is packet size and p is power needed to transmet or receive 1 bit and P is needed power to transmit or receive the whole packet.

$$P = n * S * p \tag{3.1}$$

Figures 3.5, 3.6, 3.7 and 3.8 shows the essential MQTT packets format. While connect packet's size is 392 bits, connack is 32 bits, publish is 240 bits, puback is 32 bits and disconnect is 16 bits.

- Power to send connect packet to server is:

$$P_{connect} = 392 * n * p \tag{3.2}$$

- Power to receive connack from server is:

$$P_{connack} = 32 * n * p \tag{3.3}$$

11	1C	00	07	D	A	T	A	S	E	T	00	11	S	O	I	L		M	O	I	S	T	U	R	E	1	0	2	4

Figure 3.7: Publish Packet

100	02	MSB	LSB

Figure 3.8: Publish ACK Packet

- Power to publish to server is:

$$P_{publish} = 240 * n * p \tag{3.4}$$

- Power to receive puback from server is:

$$P_{puback} = 32 * n * p \tag{3.5}$$

- Power disconnect from server is:

$$P_{disconect} = 16 * n * p \tag{3.6}$$

The needed power for sending data to server is the sum of all powers:

$$P_{Total} = P_{connect} + P_{connack} + P_{publish} + P_{puback} + P_{disconect} \Leftrightarrow P_{Total} = 712 * n * p \tag{3.7}$$

3.3.4 Conclusion

As a result for this section and according to what is required of our IoT system, the density of our network, the data that we need to collect,what hardware is available in our markets and according to the limited budget that we have; we find that the Single-Hop topology will help us to perform a performant IoT system dedicated for collecting agricultural data, monitoring and controlling farms and diseases detection, with some improvements in used protocols to get adapted to the hardware we use.

Chapter 4

Implementation

In this chapter we illustrate the deployed hardware and software tools, some practical details and issues that we've observed during the deployment phase.

4.1 Used tools

In order to deploy our IoT-system system our system we need some hardware and software tools. For this aim we've chosen the following modules, sensors, programming language and IDE.

4.1.1 Hardware

Figure 4.6 illustrates the connection of used tools in the implemented IoT node.

IoT nodes

IoT nodes are the main device in the network, these devices are providing the sensor data and sending it to the cloud. Our IoT nodes are mainly composed of:

ESP8266 NodeMcu For the light processing and communication we used the ESP8266 NodeMcu; This development board comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor [18]. This microprocessor operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi/Bluetooth and Deep Sleep Operating features make it ideal for IoT projects more characteristics are shown in Figure 4.2 [18].

DHT22 Temperature and humidity sensor The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use but requires careful timing to grab data [19]. The only real downside of this sensor is you can only get new data from it once every 2 seconds. See Figure 4.3 [19].

Its specifications are [19]:

- Supply voltage: 5V.

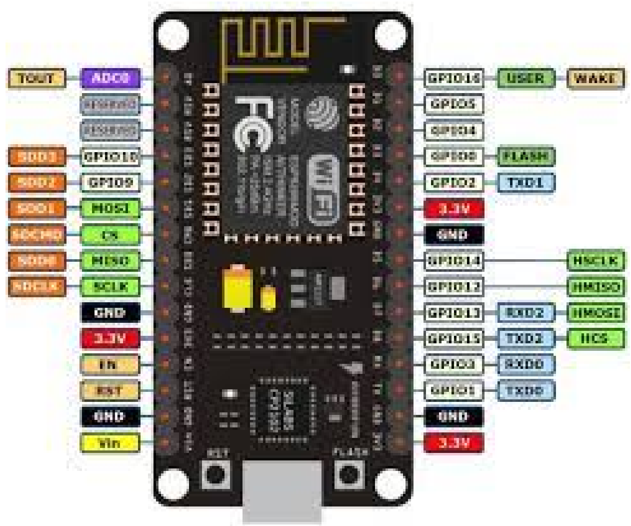


Figure 4.1: ESP8266 MCU [10]

	NodeMcu
Microcontriller	ESP8266 32-Bit
Size	49mm x 26mm
Clock Speed	80 MHz
USB to serial	CP2102
USB connector	Micro USB
Operating Voltage	3.3 V
Input Voltage	4.5-10V
Flash Memory/SRAM	4 MB / 64 KB
Digital I/O Pins	11
Analog Pins	1
ADC Range	0-3.3V
WiFi Built-In	802.11 b/g/n
Temperature Range	-40C-125C

Figure 4.2: ESP8266 characteristics

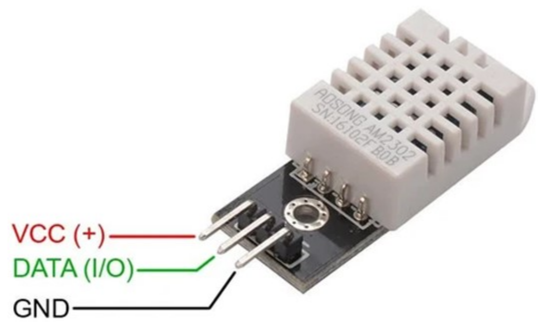


Figure 4.3: Temperature and Humidity Sensor DHT22

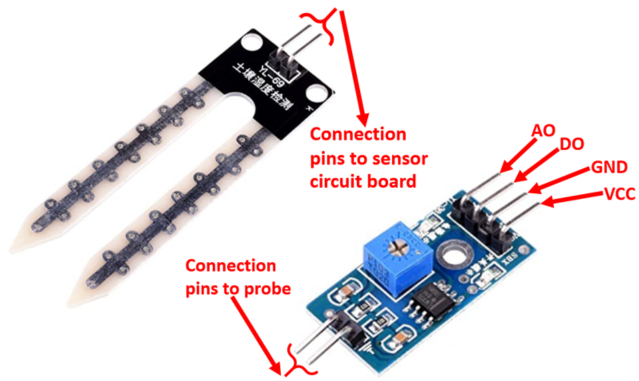


Figure 4.4: YL69 Soil Moisture Sensor [11]

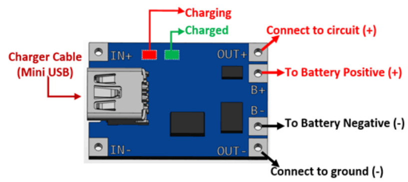
- Temperature range:-40 - 80°C
- Humidity range:0-100% RH resolution 0.1%RH error±2%RH.
- Sequence of the line: VCC, GND, S.

YL-69 Soil moisture sensor This soil moisture sensor can read the amount of moisture in the soil. The sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance). It is made of corrosion-resistant material which gives it an excellent service life. This module includes an on-board voltage regulator which gives it an operating voltage range of 3.3 5.5V. See Figure 4.4.

Power supply For the power supply we're using:

- **18650 Li-on Battery:** The battery voltage is 4.2V and its capacity is 6800mah. See Figure 4.5a.
- **TP4056 Li-on charging module:** TP4056 is a charging module for Li-on batteries. The module works on 5V and contains a Micro USB input port more details are shown in Figure 4.5b.

(a) Samsung 18650 Li-on Battery



(b) TP4056 Li-on Charging Module [20]

Figure 4.5: Power Supply System

Water proof case For the aim to assemble all node components together in cases we used Famatel 3031 10x10c waterproof cases with IP55 protection which has a complete protection against water, dust and other harmful factors.

Connection The connection of the main node components is shown in Figure 4.6

Router+Gateway

In order to connect the IoT nodes to Internet, then send sensed data to the cloud and to offer to user a full real time monitoring and controlling and also to store the generated data in a database, we used a D-link 2750-u router and 3G Dongle plugged-in the USB port of the router because the farm where we are doing our test-bed is not covered neither with 4G nor ADSL(Until the day of the experimentations). See Figure 4.7.

Outdoor customer premises equipment (CPE)

Due to the fact that the used router wasn't able to cover all our sensing zone, so we decided to add a TP-Link WA-5210G outdoor CPE to our network to get more covered zone. This device can be used as Router, Access Point or Repeater to provide long distance wireless network. In this case we are using it as access point, see Figure 4.8.

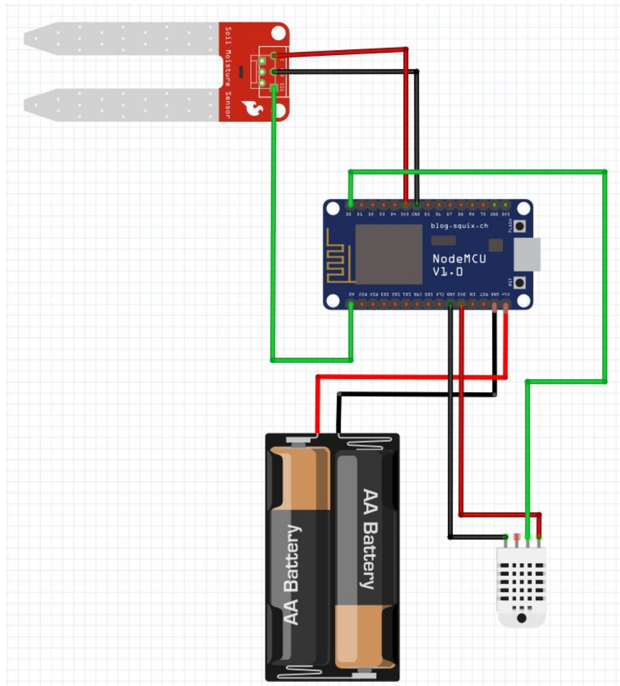


Figure 4.6: IoT Node Components Connection

Figure 4.7: Router+Dongle

Figure 4.8: IoT Node Components Connection

4.1.2 Software

Arduino language and IDE

In order to script and command the IoT node we have used the Arduino language.

Arduino code is written in C++ with an addition of special methods and functions. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language [21]. The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming (see Figure 4.9). It is where you code an validate your sketch before uploading it to the board you want to program. Arduino code is referred to as sketches [21].

Ubidots server

Ubidots is an Internet of Things (IoT) application builder with data analytics and visualization [22]. It turns sensor data into information that matters for business-decisions, machine-to-machine interactions, educational research, and an increased economization of global resources.

Ubidots delivers a secure experience for users with device friendly APIs accessed over HTTP/MQTT/TCP/UDP protocols that provide a simple connection for sending and retrieving data to/from IoT data performance optimized cloud [22]. For basic functionalities it is free with a limited number of devices, it offers also a free storage. For professional usage they offer services starts from 49$ to 499$.

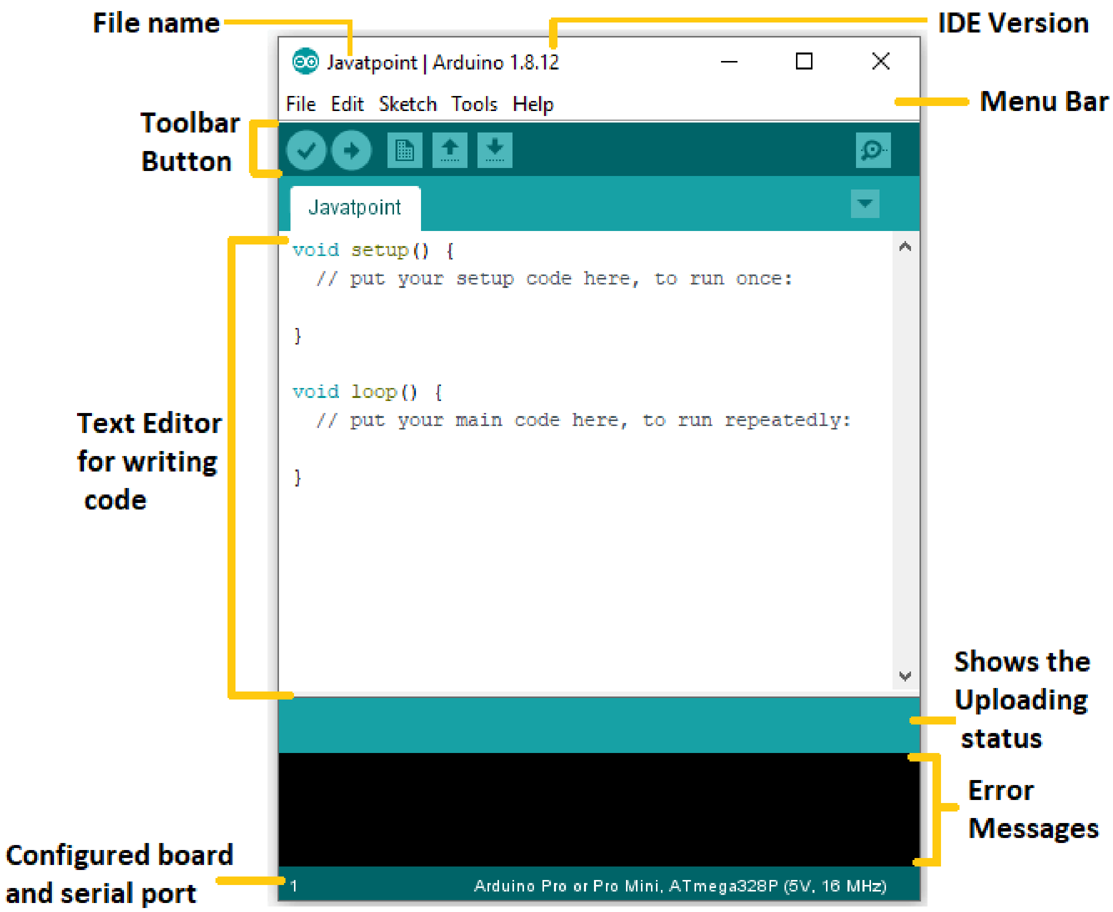


Figure 4.9: Arduino IDE Description [12]

4.2 Scripting

In this section we explain the uploaded code to IoT nodes by illustrating the used libraries dedicated for sensors programming and ESP8266 modules communication. Used libraries are:

- **ESP8266WiFi.h :**This library is responsible of ESP8266 connections.
- **DHT.h :** Concerned by temperature and humidity sensor; it translates the readings from electrical signals into readable values.
- **UbidotsESPMQTT.h :** Is a library that connects ESP8266 chip to Ubidots server using MQTT protocol

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#include "UbidotsESPMQTT.h"
```

The Arduino sketch contains two main functions:

- **setup() function :** Is called once, when the sketch starts. It's a good place to do setup tasks like setting pin modes or initializing libraries [23]. In this function we should write the static part and the starting parameters that should be done in the beginning of each run of the IoT node. In the setup() of uploaded sketch the conection to Access Point is established.

```
        WiFi.begin(ssid, password);
        while(WiFi.status()!= WL_CONNECTED){
            delay (1000);
            counter ++;
        if (counter == 20) {
            break;
            }
        }
```

- **loop() function :** Is called periodically and it is the heart of most sketches [23]. loop() executes three main tasks, the first is the connection to ubidots servers,

```
        if(!client.connected()){
            client.reconnect();
        }
```

reading sensed values,

```
    if (isnan(h) || isnan(t) ) {
        h = dht.readHumidity();
        t = dht.readTemperature();
    return;
    }
```

```

    float  soil = analogRead(moisture_pin);
    float value = (((1024 - soil) / 1024) * 100);
```

and sending the sensed data to server.

```
    client.add("soil_moisture", value);
    client.add("Temperature", t);
    client.add("Humidity", h);
    client.ubidotsPublish("Node_1");
```

After sending data to servers the IoT node should pass to deep sleep mode, this is the most power efficient option and the ESP chip only draws approximately 20µA. Using the function

```
     ESP.deepSleep(ESP.deepSleepMax());
```

where the function ESP.deepSleepMax() gets the maximum duration of deep sleep before the IoT node wakes up.

4.3 Empirical measurements

After many test beds and experimentations during the period of implementation we got the following results:

- IoT node stayed alive for consecutive 4 days of sending data each 5 minutes. The node generated three csv files of temperature, humidity and soil moisture, each file contains 1152 lines(see Figure 4.10).
- During the 4 days of sensing and sending data, IoT node send wrong temperature and humidity data during 1 hour, then it resumed sending correct data. the percentage of wrong data is:

  $$((60/5)/1152) * 100 = 1,04\% \quad (4.1)$$

  Which stays perfectly in the standards. As a result, when we configure the node on the maximum duration of deep sleep using the function "ESP.deepSleepMax()" which is around 3.5 hours, the IoT node can stay alive for more than 3 months without the need for charging.

4.4 Issues and requirements

As any project that deals with deployment of hardware and software, we faced some issues during the implementation of our project we can mention some of them in order to get some lessons and experiences.

- **Tools and hardware components availability** When we decided to start the realization of the IoT network we found ourselves very limited by the available hardware in the Algerian market which is still in the beginning and also the high pricing comparing to other international markets.

```
16.0,{},2022-05-21 02:59:21.824000+01:00
17.0,{},2022-05-21 02:54:30.008000+01:00
17.0,{},2022-05-21 02:49:38.002000+01:00
17.0,{},2022-05-21 02:44:47.188000+01:00
17.0,{},2022-05-21 02:39:54.034000+01:00
17.0,{},2022-05-21 02:35:01.910000+01:00
17.0,{},2022-05-21 02:30:10.192000+01:00
16.0,{},2022-05-21 02:25:18.277000+01:00
17.0,{},2022-05-21 02:20:26.648000+01:00
17.0,{},2022-05-21 02:15:34.796000+01:00
17.0,{},2022-05-21 02:10:42.936000+01:00
```

(a) Collected Temperature samples

```
40.0,2022-05-24 17:19:52.853000+01:00
46.0,2022-05-24 17:19:09.710000+01:00
39.0,2022-05-24 17:18:47.055000+01:00
38.0,2022-05-24 17:18:27.177000+01:00
40.0,2022-05-24 17:18:10.273000+01:00
41.0,2022-05-24 17:17:45.402000+01:00
39.0,2022-05-24 17:17:01.614000+01:00
42.0,2022-05-24 17:15:08.721000+01:00
40.0,2022-05-24 17:14:47.684000+01:00
41.0,2022-05-24 17:14:06.779000+01:00
44.0,2022-05-24 17:13:49.655000+01:00
42.0,2022-05-24 17:12:12.546000+01:00
```

(b) Collected Humidity samples

```
1024.0,2022-06-08 12:20:36.719000+01:00
1024.0,2022-06-08 12:20:25.055000+01:00
846.0,2022-06-07 15:29:49.541000+01:00
845.0,2022-06-07 15:29:37.787000+01:00
846.0,2022-06-07 15:29:26.114000+01:00
17.0,2022-06-07 15:05:58.950000+01:00
847.0,2022-06-07 15:05:30.602000+01:00
846.0,2022-06-07 15:05:03.442000+01:00
845.0,2022-06-07 15:04:51.743000+01:00
846.0,2022-06-07 11:40:05.220000+01:00
845.0,2022-06-07 11:39:50.905000+01:00
```

(c) Collected Soil Moisture samples

Figure 4.10: Collected Data

- **4G coverage** The zone where we installed and experimented our network is not covered by 4G so we couldn't use a 4G router to connect the IoT nodes to Internet, we decided to use a 3G dongle and connect it with simple router via USB. This technique connected the nodes with the server but if the 4G or 5G were available they will decrease the latency and save more power.

- **HTTP vs MQTT** During the experimentation phase we first used HTTP protocol for the communication between IoT nodes and Ubidots servers.Unfortunately, we discovered after few time that the nodes sometimes are making between 4 and 6 attempts to send data to server, by debugging the code we found that the node is crushing and sending exception codes, using the ESP Exception Decoder plug-in we discovered that when the node is sending HTTP data a stack-overflow is caused by the packet's size which exceed the stack's size. Therefore, we decided to use the MQTT protocol, the nodes send data from the first attempt what increased the network's life time.

- **ESP8266 Range** The covered distance by ESP8266 is between 15m and 20m which made us not able to apply architectures that requires communication between nodes.

- **UbidotsESPMQTT Library** After using UbidotsESPMQTT library we discovered some disadvantages, in case of disconnection due to technical problems from the server or internet provider the *reconnect()* function doesn't stop attempting connections until it connects because it contain a while loop conditioned with "not connected", for our network is a big problem which can consume energy for days without connecting. To solve this problem we limited the connections attempts to 20, to do this we added a variable named *counter* in the header file of the library then we added an other condition to the while loop in the cpp file which is "counter less or equal 20" and counter is incremented by one in each attempt.

- **Internet stability** Due to the internet interruption during the period of baccalaureate exam we lost a data of a whole week we were waiting to collect it.

Conclusion

The implementation of technologies in the agricultural field has shown its efficiency, the concept of IoT is one of these technologies, it offers to the stakeholders a complete dashboard in order to take the right decisions in the right time, therefore, the profit margin is increased and the final consumer is secure.

In this thesis we first proposed a design of an IoT network in which we describe the deployed architecture and protocols, used hardware and software in order to collect a real time data and use it as input in many Artificial Intelligent based systems which can help experts and farmers to protect their crops by predicting diseases. Finally, we a realized a network of IoT nodes equipped with sensors sensing three different types of data, the IoT network is realized with a low budget, it gathers data, predict diseases and control farms. The final results has shown a good efficiency compared to the cost.

Second shutter, we report the realization of the proposed IoT. The IoT nodes equipped with sensors that sense three different types of data related to the weather and soil. In the design we have respect many constraints and conditions. To reach an optimized IoT network. In fact this network is realized with a low budget and in a non-covered zone with Internet.

The outcome of that project has shown what we have learn from this investigations, many knowledge and known-how have been reached. In fact we have :

- acquired some knowledge about IoT and their deployment.
- raised some known-how about realization of IoT.

Due to the short reserved time for data gathering and due the instability of The Internet during the week of baccalaureate exams, IoT nodes can't collect the expected dataset size.

Appendix A

Illustrations an realization photos

FigureA.1 and A.2a illustrate

- the map of the farm used as the experimental site.
- the assemble node components respectively. For

  the cost of that note is 3750 DA (2000 DA for the MCU, 500 DA for soil humidity sensor, 700 DA for the rechargeable battery module and 450DA for the waterproof box)

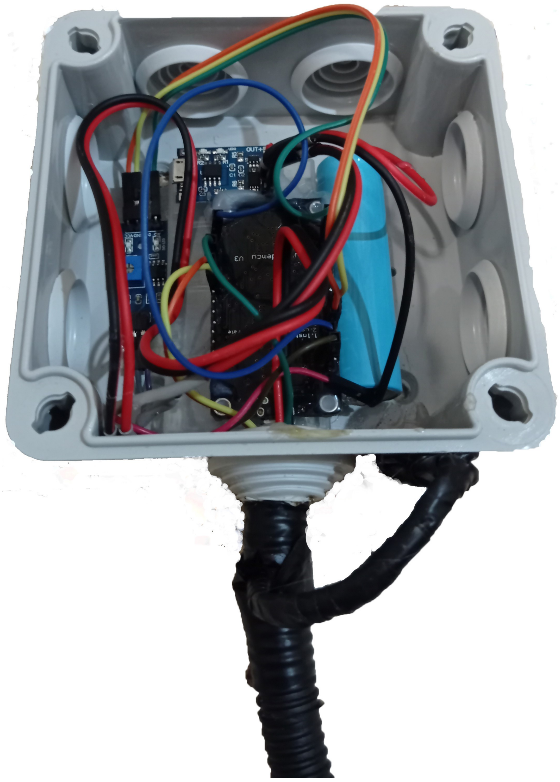
A.1 Experimental in-site farm

We have deployed our designed IoT network in a farm of olive and almond trees. This farm used the drip irrigation system. Olive trees age are between 8 and 12 years. While the almond trees are 4 year old. The Farm has a rectangle shape of (200m $\times 270m$).

For our purpose we need to have some information on the root systems of both type of trees to fix where the soil moisture sensors will be placed.

Concerning the root system of the olive trees it develops vertically until the third to fourth year of its life. Later on, the original root system is replaced by another is a flocculent root system formed in the olive tree neck, just below the soil surface. The root system development method is mainly determined by the nature of the soil. In the targeted Farm this root system is sited between 30cm to 100cm. While the almond trees root system is less deep. Roots are sited between 10cm and 50cm.

Figure A.1: The test bed farm map



(a) Assembled Node

(b) Assembled Node Installed in the Farm

Figure A.2: Node

Bibliography

[1] O. Bhat, P. Gokhale, and S. Bhat, "Introduction to IOT," *International Advanced Research Journal in Science, Engineering and Technology ISO*, vol. 3297, no. 1, 2007.

[2] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1761–1804, 2020.

[3] F. Khelifi, "Monitoring system based in wireless sensor network for precision agriculture," *Internet of Things (IoT): Concepts and Applications*, pp. 461–472, jan 2020.

[4] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes experiences from a pilot sensor network deployment in precision agriculture," *20th International Parallel and Distributed Processing Symposium, IPDPS 2006*, vol. 2006, no. May, 2006.

[5] D. Ardiansyah, A. S. Miftahul Huda, Darusman, R. G. Pratama, and A. P. Putra, "Wireless Sensor Network Server for Smart Agriculture Optimatization," in *IOP Conference Series: Materials Science and Engineering*, vol. 621, Institute of Physics Publishing, nov 2019.

[6] S. Awan, S. Ahmed, F. Ullah, A. Nawaz, A. Khan, M. I. Uddin, A. Alharbi, W. Alosaimi, and H. Alyami, "IoT with BlockChain: A Futuristic Approach in Agriculture and Food Supply Chain," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[7] A. Zeb, A. K. Muzahidul Islam, M. Zareei, I. Al Mamoon, N. Mansoor, S. Baharun, Y. Katayama, and S. Komaki, "Clustering analysis in wireless sensor networks: The ambit of performance metrics and schemes taxonomy," *International Journal of Distributed Sensor Networks*, vol. 12, no. 7, 2016.

[8] U. M. Pešović, J. J. Mohorko, K. Benkič, and Ž. F. Čučej, *Single-hop vs. Multi-hop – Energy efficiency analysis in wireless sensor networks*. Srbija, Beograd: Telekomunikacioni forum TELFOR 2010, 2010.

[9] "Le protocole MQQT et l'internet des objet (IOT) - éduscol STI."

[10] "Pilotez un NodeMCU ESP8266 à l'aide d'une interface web • AranaCorp." https://www.aranacorp.com/fr/pilotez-un-nodemcu-esp8266-a-laide-dune-interface-web/(seen on 2022-03-07).

[11] "Soil Moisture Sensor - How to use with Arduino - DIY Engineers." {https://www.diyengineers.com/2021/02/04/how-to-use-a-soil-moisture-sensor-with-arduino/}(seenon2022-03-08), urldate = 2022-03-08.

[12] "Arduino IDE - JavaTpoint." https://www.javatpoint.com/arduino-ide(seen on 2022-03-13).

[13] B. Dalal and S. Kukarni, "Wireless Sensor Networks-Applications," tech. rep., 20!!

[14] Soumyalatha and S. G. Hegde, "Study of Iot: Understanding IoT and aplications, issues and challenges," *International Journal of Advanced Networking & Application (IJANA)*, no. January, pp. 477–482, 2010.

[15] R. I. Tandel, "Leach protocol in wireless sensor network : A survey,"

[16] J. Paradells, T. Staub, R. D. Oliveira, J. Karlsson, F. Liu, R. Haarman, and P. Dely, "Chapter 5 : MULTI-HOP WIRELESS," *Wireless Networks*, pp. 1–68.

[17] R. A. Atmoko, R. Riantini, and M. K. Hasin, "IoT real time data acquisition using MQTT protocol," *Journal of Physics: Conference Series*, vol. 853, no. 1, 2017.

[18] "NodeMCU ESP8266 Pinout, Features & Datasheet." https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet (seen on 2022-02-02).

[19] "Overview — DHT11, DHT22 and AM2302 Sensors — Adafruit Learning System." https://learn.adafruit.com/dht (seen on 2022-03-02).

[20] "[Energie] TP4056 et batterie 18650 sur moteur électrique." https://forums.futura-sciences.com/electronique/904124-tp4056-batterie-18650-moteur-electrique.html(seen on 2022-03-13).

[21] "Everything You Need to Know About Arduino Code." https://www.circuito.io/blog/arduino-code/ (seen on 2022-03-02).

[22] "Ubidots IoT Application Development Platfrom, Sigfox Partner Network, The IoT solution book." https://partners.sigfox.com/products/ubidots-for-business (seen on 2022-03-03).

[23] "Arduino Sketches — Arduino Documentation — Arduino Documentation." https://docs.arduino.cc/learn/programming/sketches(seen on 2022-03-10).