Master Thesis

Study of the Development of an IoT-based sensor platform for E-Agriculture

Master's degree in Industrial Engineering

Andrea Saba Sasot

15 Jan 2020

Universitat Politècnica de Catalunya
Department of Engineering Electronics
ESEIAAT (Escola Superior d'Enginyeries Industrials
Aerospacials i Audiovisuals de Terrasa)

Supervised by
José Antonio Soria Pérez

Documents

1. **Report**
2. Attachments
3. Budget

Contents

3

1 Introduction **1**
1.1 Background . . . 1
1.2 Aim of the project . . . 2
1.3 Scope of the project . . . 2
1.4 Specifications and Requirements . . . 3
1.5 Project deliverables . . . 4
1.6 Motivation of this work . . . 4
1.6.1 Overview of the manuscript . . . 4

2 E-Agriculture and Internet of Things **7**
2.1 Overview . . . 7
2.1.1 Application fields . . . 7
2.1.2 Major concerns . . . 10
2.1.3 Agriculture and Nutrition . . . 12
2.1.4 IoT and Agriculture . . . 14
2.2 The IoT Architecture . . . 18
2.2.1 The Things . . . 19
2.2.2 Data acquisition systems . . . 19
2.2.3 Edge Systems . . . 21
2.2.4 Server/cloud computing . . . 22
2.3 *M2M* and IoT Communication . . . 23
2.3.1 M2M Communication systems . . . 23
2.3.2 Protocols and message mechanisms . . . 24
2.4 IoT third-party commercial platforms . . . 25
2.4.1 ThingSpeak . . . 26
2.4.2 The Things Network . . . 27
2.5 Web server development and programming tools . . . 28
2.6 Agriculture sensors nodes and irrigation systems . . . 30
2.6.1 Arduino Boards . . . 31
2.7 Discussion . . . 33
2.7.1 Communication node-gateway . . . 33
2.7.2 Communication gateway-server . . . 36
2.7.3 Server side selection . . . 37
2.7.4 Node components selection . . . 37

2.8 Software and hardware selection 40

3 Platform Design and Development 43
3.1 Overview 43
3.2 E-agriculture platform layout 43
3.3 Network configuration 44
3.4 SQL database 46
3.5 Design of web interface in *localhost* 48
3.5.1 Front end 49
3.5.2 Back end 52
3.6 Node-RED layer 52
3.7 Gateway application program 53
3.8 Sensor/Actuator node 55
3.8.1 Node hardware system 55
3.8.2 Node application program 56
3.9 Discussion 58

4 Test, Set-up and Operating Results 59
4.1 Overview 59
4.2 Platform and system set-up 59
4.3 1st. Demo: Sensor node to *localhost* 61
4.4 2nd. Demo: *localhost* to sensor node 63

5 Conclusions and Future Work 65
5.1 Conclusions 65
5.2 Future Work 65

Acknowledgments 69

Bibliography 75

List of Figures

1.1 IoT-based sensor platform for Smart Agriculture layout 2

2.1 Smart home sketch [24] 8
2.2 Wireless blood pressure monitor [29] 9
2.3 Smart city implementations in Barcelona [32] 10
2.4 Smart Factory KL [37] 10
2.5 Barriers to IoT Success [39] 11
2.6 Undernourisment in the world [53] 13
2.7 IoT system for banana fields [60] 16
2.8 Precision fertilizer [61] 16
2.9 Cow geoposition device [63] 17
2.10 IoT general data process [64] 18
2.11 The Raspberry-pi embedded platform (Model 3b+) [65] 20
2.12 LG-01 Dragino product [66] 21
2.13 IPv4 Address Structure [74] 25
2.14 ThingSpeak IoT platform [76] 26
2.15 The Things Network platform [78] 27
2.16 A high-level overview of LAMP's determining components [80] 29
2.17 Arduino NANO 33 board [84] 31
2.18 Waspmote sensor node [85] 32
2.19 ESP8266 module [86] 32
2.20 LG01 in an IoT Network [66] 36
2.21 Arduino UNO board [88] (Left), LoRa shield and antenna [89] (Right) 38
2.22 SoilWatch (Left) [90], Adafruit SHT-30 sensor [91] (Right) 38
2.23 JT180 water pump [92] 39
2.24 Node's independent energy system [94] 40

3.1 Developed IoT-based agriculture architecture 43
3.2 Steps to connect Dragino wirelessly to the router [66] 45
3.3 Dragino's network IP address 45
3.4 Server's PC pinging Dragino's IP address 45
3.5 phpMyAdmin user interface 46
3.6 Database structure 46
3.7 Relation between the scripts implementing the web page 49
3.8 Landing page 50
3.9 "Your field" page 50
3.10 "Sensor records" page 51

3.11 “System settings” page 51
3.12 Back end flow chart 52
3.13 Platform’s Node-RED layer (crop section) 53
3.14 Simplified Dragino work flow 54
3.15 Node’s electronic circuit sketch 55
3.16 Pin Mapping For Lora [102] 56
3.17 Simplified Arduino work flow 57

4.1 Gateway connections 60
4.2 Node connections 60
4.3 Port forwarding using a Virgin router 61
4.4 System network sketch 61
4.5 Moisture sensor test 62
4.6 Moisture sensor results 62
4.7 Temperature and humidity test 63
4.8 Temperature and humidity sensor results 63
4.9 Irrigation system test 64

List of Tables

2.1 IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units) [11] 8
2.2 Communication protocol selection for the communication node-gateway 33
2.4 Pros and cons of LoRa technology 35
2.5 Platform's Software and hardware selection 40

3.1 "SensorActuator_Data" table structure 47
3.2 "SystemStatus" table structure 47
3.3 "UserDefined" table structure 48

1 Introduction

1.1 Background

Due to climate change and global warming, developing new ways of growing ecologic food while managing natural resources with sustainability, has become a main challenge for food producers in the world [1].

According to [2], by 2050 the world's population is expected to increase by 25%. Therefore, food companies will have to adapt in order to meet this demand by increasing food production by 50%, while competing in the global market. Because of this, developing new ways of harvesting food based on improved information systems and more efficient computing technologies will be essential to achieve this goal [1].

Since the first-interconnected device was created in 1982 [3], the field of sensor embedded computing has spread rapidly over the last decades, opening the door to the Internet-of-Things (IoT) . This concept is based on a system of computing devices embedded into physical objects, with the capacity to transfer data over a network, and the Internet, without requiring human interaction [4, 5, 6]. This technology has become increasingly popular both in our daily and professional lives. Smart homes, autonomous cars, Industry 4.0, or smart cities are just a few examples.

The agricultural sector has been one of the last sectors to introduce IoT solutions to improve their processes in order to become more efficient and environmentally sustainable. Smart farming is a farming management concept using modern technology to increase the quantity and quality of agricultural products, by deploying technologies such as IoT, data management, drones, GPS (*Global Positioning System*), among others [7]. For example, [8] highlights the necessity of using data analysis techniques to increase the efficiency of food production and to prevent pests. All in all, the result is the reduction of the use of water, fertilizers, energy, and pesticides, obtaining amore sustainable harvest, improving productivity and saving time. However, many small and medium size agriculture companies are still reluctant to implement new technologies, considering not worth it the cost of the infrastructure and/or its complexity [9].

1.2 Aim of the project

The aim of this project is to develop a custom IoT-based sensor platform with a basic network topology, using agriculture sensors and an irrigation system which can be controlled and monitored via Internet.

1.3 Scope of the project

In order to fulfill the objectives of this project, our proposed platform for Smart Agriculture will include a single sensor node, a gateway, a server application and a web page (Fig. 1.1). Its Iot functionality will be demonstrated according to the following behavior:

- The sensor node includes the controller board, the sensors, and the actuators. Its main function is to read sensor values and actuate the irrigation valves. When connected to the Internet, it also process user commands to send the data necessary for monitoring the plants and control the irrigation remotely.
- The gateway will act as a router, permitting the dialogue both ways between sensor node and a web-based server application, wirelessly.
- The server will contain the database corresponding to sensor data and user information, and it will host the web application page. It will be hosted in a laptop computer.
- The web application will allow the user to interact with the platform remotely after login to the server . Some basic operations (such as reading sensor data, activating the irrigation system, as well as configuring hardware platform settings) will be implemented to demonstrate platform suitability to deploy smart agriculture in larger fields in the future.
- Aspects related to the design of electronic circuit for sensor signal conditioning, network topology, system integrity, security and privacy, are beyond the scope of this work and will not be considered.

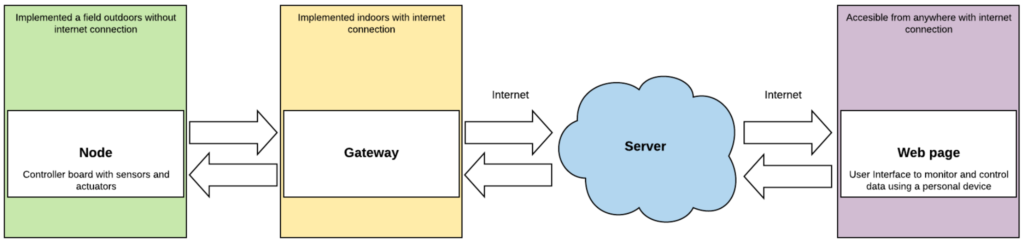


Figure 1.1: IoT-based sensor platform for Smart Agriculture layout

1.4 Specifications and Requirements

- **Sensor specifications**: Selected sensors will monitor basic parameters of the field: temperature, humidity and soil moisture. As they operate outdoors, they must be robust enough to endure adverse weather conditions, providing the necessary electronics for handling analog signals easily, or I2C/SPI digital protocols when necessary.
- **Irrigation System:** It must include the actuators and valves to irrigate the field in accordance with the information gathered from the sensors. The system must also include hoses and pipes connected to a near water supply service and/or a pump connected to a water tank.
- **Node board and CPU (*Central Processor Unit*):** The controller board must permit an easy wiring and assembly of sensor circuits, and include a CPU with the necessary specifications and resources for developing the local functions - sensor readings, irrigation and transmission of data. A Its wireless mechanism must communicate over long distances (5Kms) with short data-frames. The LoRa (*Long Range*) system is a specification for low-power devices suited for both LAN and WAN networks (*Local and Wide Area Networks*) suited for agriculture applications.
- **Gateway hub**: Its firmware must be LoRa compatible in order to communicate with the sensor node. It must provide connection to the Internet, either through Ethernet socket (and/or WiFi) as well as firmware support for providing access to web databases by means of M2M (*Machine-to-machine*) connectivity protocols. For these reasons, an OS-based (*Operating System*) platform is highly recommended for developing the necessary scripts. The gateway will stay indoors close to the server computer.
- **Server**: the IoT-based platform must be developed from scratch under LAMP (*Linux*, *Apache*, *MySQL* and *PHP*) web service stack: Apache is necessary to install a server in the computer, *MySQL* to develop the databases, and PHP to support the extension *phpMyAdmin,* an interface which facilitates the management of databases and the access of users to the web page and the platform. The data table must contains the necessary fields to store the different data, including sensor information sent from the node, which is read by the web application, as well as commands written by the user to change node behavior.
- **Web page** : The application must introduce some user-friendly mood to data display meaningful "easy-to-read", and to allow sensors and actuators to be operated easily by users.
- **Scalability:** Although the platform is restricted to just one node, the scripts should be, as far as possible, flexible enough to grow the network in the future, by adding more nodes.

1.5 Project deliverables

In this work, the following deliverables will be developed to achieve the proposed goals:

- **Smart-Agriculture Platform:** This involves the choice and assembly of the physical elements that make up the IoT system for the application at hand, including: the sensors and actuators that would normally be placed outdoors, the hardware units for both the node and the gateway, and the powering system. In this sense, although some elements may be considered independently, vendors offer IoT solutions based on closed products which may be taken into account.
- **Hardware programs**: This involves the implementation of scripts for managing configuration and sensor readings on one side, and the transmission of data through the gateway, on the other side.
- **Server application:** This stage involves both the design and implementation of the web application page accessed by users and the intermediate database for interfacing between sensor/actuator node and web page, respectively.

1.6 Motivation of this work

When I enrolled my master's degree in Industrial Engineering, I had little knowledge about electronics. Over the years, I had the possibility to course different subjects based on electronics and automation which made me realize how boundless this technology is. Since then, I have been amazed about how this technology can help our daily lives in many ways.

However, I never had the time and resources to start a project by myself during the degree. For this reason, the possibility of developing my master's thesis in this direction was more than attractive. I wanted to develop something useful, and different ideas came to my mind, but my tutor's one was the one I liked the most: IoT and Agriculture.

The relation of my family background with agriculture showed me the great benefits of eating what you harvest. I know this can be sometimes a luxury considering the amount of time you have to invest, and the space, knowledge and resources it involves. Therefore, one of my main goals in this project is to take the firsts steps to simplify these tasks and share this knowledge with people interested in smart agriculture.

1.6.1 Overview of the manuscript

In this chapter, the general guidelines and the goals of this project have been presented. Chapter 2 introduces the main concepts about IoT technology. It reviews the state-of-the-art of both fields and its growth forecast. It also discusses the selection of devices for designing E-agriculture applications. More precisely, the electronics of computational elements, sensor network typologies and communication systems commonly used in Iot-based platforms will be analyzed and discussed in order to start defining the aforementioned system.

In chapter 3 the design and development of the platform is explained in much detail. The scope and specifications have been outlined in sections 1.3 and 1.4.

Chapter 4 shows preliminary test results and operating behavior of the developed platform. Specially, the capacity of the platform to give users remote access to read data from the sensor node and control the irrigation via Internet will be demonstrated. A brief instruction manual explaining the end user how to use the platform is also provided.

Finally, chapter 5 discusses what our next steps should be for improving the current platform. This considers the development new functionalities for the current web page and sensor/actuator node, as well as the addition of more elements to extend the platform to other agriculture applications.

2 E-Agriculture and Internet of Things

2.1 Overview

The *Internet-of-Things* (IoT) is a concept related to small computing devices, which can be embedded to any object -from mechanical and digital machines to objects, animals or people- to provide them with unique identifiers (UId) and give them the ability to transfer data over a network and the Internet, with no need of human interaction [4, 5, 6]. The vision is that, by connecting objects, people and spaces, it is possible to introduce changes which can be only beneficial: increasing the productivity and efficiency of the environment while reducing consumption of resources, improving decision making by predictive analysis, fast response or reduction of human errors, are just a few.

Since the first contemporary vision of IoT was coined by Weiser in 1991 [10], embedded systems, wireless sensor networks as well as industrial automation have been the fields boosting the deployment of IoT in the world. But in the last decade, this scope has also included to real-time analytics, big-data or even machine learning, among other disciplines [4].

Many markets have shown the benefits of embracing IoT in their workplace. As shown in Table [11], the number of IoT endpoint units has grown considerably in the last two years and by 2020 is expected to increase 21%. across global industries.

2.1.1 Application fields

The application fields in IoT are often classified into four main groups: consumer, commercial, industrial and infrastructures/spaces [12, 13, 14]. A growing portion of IoT devices are for consumer use, including wearable technology, home appliances or medical care, all having remote monitoring capabilities [15].

Iot-based devices can be part of smart home (or automated) platforms that controls lightning, heating and security systems, among others [16, 17, 18] (Fig. 2.1). In this sense, log-term benefits can inlude, for instance, energy savings by automatically ensuring that lights and electronics are turned on and off only when necessary. *Amazon Echo*, *Google Home* or *Samsung's Smarthings Hub* [19] offer commercial standalone

Table 2.1: IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units) [11]

Segment	2018	2019	2020
Utilities	0.98	1.17	1.37
Government	0.40	0.53	0.70
Building Automation	0.23	0.31	0.44
Physical Security	0.83	0.95	1.09
Manufacturing & Natural Resources	0.33	0.40	0.49
Automotive	0.27	0.36	0.47
Healthcare Providers	0.21	0.28	0.36
Retail & Wholesale Trade	0.29	0.36	0.44
Information	0.37	0.37	037
Transport	0.06	0.07	0.08
Total	3.96	4.81	5.81

platforms to connect different smart home products. In addition, to the commercial systems, the are many non-propietary open-source ecosystems, including: Home Assistant, OpenHAP or Domoticz, among others [20, 21].

Another key application of smart home is to provide assistance by accommodating technology to elderly individuals and people with disabilities. For example, devices controlled by voice can assist users with sight and mobility limitations. To provide such features, Iot-based devices are equipped with sensors to monitor for medical emergencies, such as falls or seizures [22, 23]. Smart home applied in this way can provide users more freedom and a higher quality of life.

Figure 2.1: Smart home sketch [24]

The *Internet of Medical Things* (IoMT) is a branch of IoT related to the collection of medical data from people for research purposes. The information is not only used to monitor and alert patients, but also it plays an important role in the prevention and managing of chronic diseases [25]. Monitoring devices can range from blood pressure and heart rate to more advanced systems, such as: pacemakers, electronic wristbands and specialized implants, among others [26] (Fig. 2.2). In the later case, recent advances in *MEM* technology (*Michro-Electro Mechanical* devices) have enabled point-of-care medical diagnosis, where portability and low system-complexity is essential [27]. A report developed in 2015 by Goldman Sachs concluded that IoT devices cloud save more than $300 billion in annual healthcare expenditure in the USA [28]. For this reason, it is not surprising that IoMT is also extended to healthcare insurance industries nowadays [25].

Figure 2.2: Wireless blood pressure monitor [29]

In the building and home automation sector, IoT devices manage the mechanical and electronic systems of several types of buildings - whether private or public, industrial, institutions and/or residencies [26] (Fig. 2.3). In this context, three sub-areas are covered: 1) energy management; 2) surveillance; and 3) sensor integration [30], In the first, IoT-based devices helps to deploy energetically-efficient driven environments, optimizing consumption, whereas in surveillance real-time monitoring is used to track the behavior of occupants. Finally, the integration of sensors and computational nodes in the built environment helps knowing how resources could be designed and/or be used in the future [31].

Similarly, Smart cities are urban areas where IoT devices are used to manage assets, resources and service efficiently. In this application, data collected from citizens and devices can be analyzed and used for monitoring and managing traffic and transportation systems, power plants, water supply networks, crime detection, schools, hospitals and other community services [33, 34].

In industrial applications, IoT devices acquire and analyze data from connected equipment, manufacture locations and people. Measurements, automated controls, plant optimization as well as health and safety management and other manufacturing functions are provided by networked sensors [26] (Fig. 2.4). The principle is that such a highly-integrated cyber-space allows production plants to be optimized in real-time, designing new products with batches adapted to market demand [35]. The vision is that such scenario could generate so much business value that will

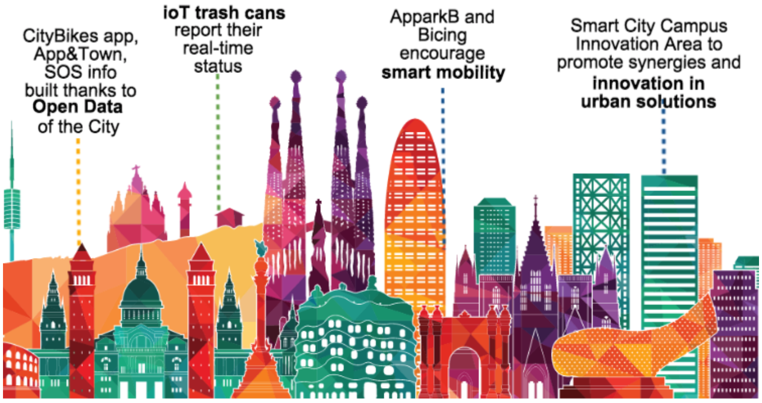


Figure 2.3: Smart city implementations in Barcelona [32]

eventually lead to the so called *Fourth Industrial Revolution* - also referred to as *Industry 4.0*. The potential growth for deploying IoT in this sector may generate $12 trillion of global GDP[1] by 2030 [36].

Figure 2.4: Smart Factory KL [37]

2.1.2 Major concerns

Despite the shared belief in the potential of IoT, there are still some barriers to overcome in order to accept this technology more widely. The main one, and most important, relates to security aspects (see Table 2.5). Fundamentally there are four levels of concerns that IoT devices and networks must address in this sense, all of them related to transmitted and stored data [38].

- **Confidentiality**. Unauthorized parties cannot have access to data.

Because of limitations in the way that services and nodes operate over a network, quite often user information is visible and vulnerable to privacy violation [40]. In general, attackers exploit flaws in device hardware and program code to enter digital systems with unauthorized access, among which, the main technique used today is known as *SQL frame injection* [41].

[1] *Gross Domestic Product*

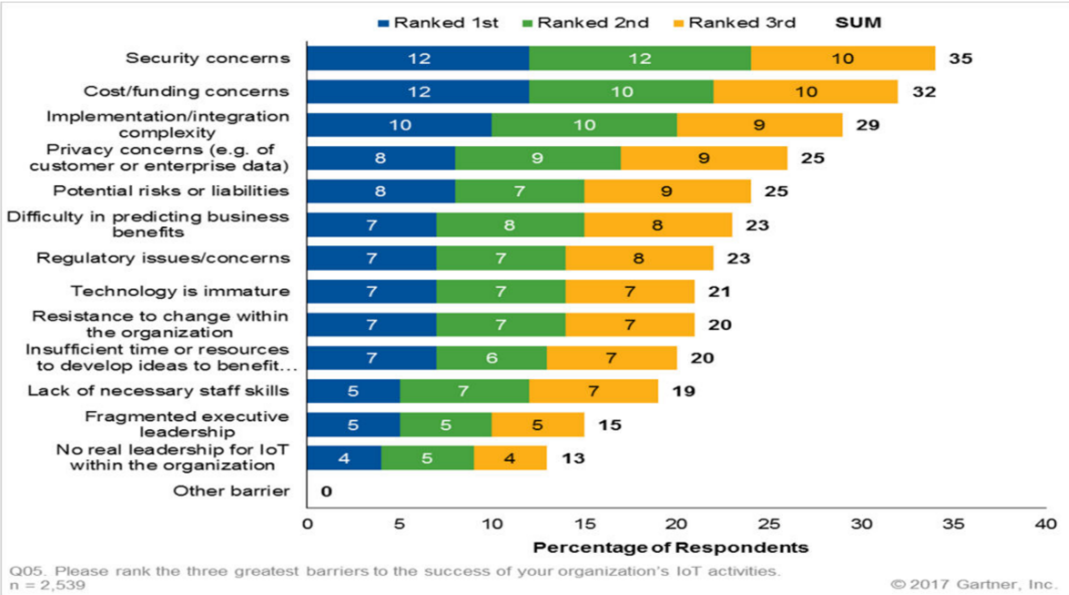


Figure 2.5: Barriers to IoT Success [39]

- **Integrity**. Intentional and unintentional corruption of data needs to be detected.

Corruption of data can be caused by malfunctions in network devices during the transmission of data. Fortunately, most of the existing HTTP-based protocols in IoT, such as MQTT among others, solve such problems by incorporating handshake and acknowledge mechanisms to check message integrity. But a major concern in this sense is the unwanted encryption of files. *Ransomware* is a type of malware from cryptovirology that threatens to publish victim's data or perpetually block access to it unless a ransom is paid [42]. In these cases, using antivirus programs or security policies to block known malware from launching will help to prevent the infection, but will not protect against all attacks [43, 44]. Therefore, having a proper backup solution is critical to defend against malware.

- **Non-repudiation**. A sender cannot deny having sent a given message.

This is one of the reasons why current web providers nowadays force users, first to register to their platform, and then *login* to use a given service.

- **Availability**. Data should be available to authorized parties even under external attacks.

This principle is particular important when *Distributed-Denial-of-Service* attack (DDoS) are produced. It means that if a perpetrator seeks to make a machine, a network or a service from it unavailable by disrupting services and making them unavailable, data should not be lost. This kind of menace is generally accomplished by flooding the targeted in an attempt to overload systems in order to avoid legitimate requests of being fulfilled [45].

Despite the huge development in the last decades, it cannot be omitted that IoT still presents significant uncertainty. In this sense, a study conducted by Ericsson in 2017 regarding the adoption of IoT identified a “clash” between IoT and companies’ governance structures [46]. One of the reasons that have been mentioned is that IoT suffers from platform fragmentation and lack of technical standards [47, 48, 49]. For all this reasons and the lack of historical precedence, it is generally accepted that IoT technology still is in its early stages of development.

2.1.3 Agriculture and Nutrition

Agricultural development is considered a main resource to feed the population and crucial in the economical growth of many countries, accounting for one-third of global GDP worldwide [50]. By the end of the 20th century, although global demand for agricultural products and growth population rates were rising, it did so less rapidly than in previous decades, raising the fear that the world could not be able to grow enough food and commodities so that future populations are adequately fed.

At that time, the FAO association indicated that low population growth rates combined with high levels of food consumption per person, beyond which further rises are limited, was the main cause. In this sense, demand for agricultural products was expected to decrease from 1.6% a year in 1999 to 1.4% by 2015, as world population increased [51], whereas the number of undernourished people could even decrease from 815 million to 610 by 2015 - and even to 443 million by 2030.

However, such decline has not been as expected in the 21st century. By 2018, 820 million people were still hungry, with the sub-Saharian region having the highest prevalence of undernourishment (21%), followed by the South-Asia region (16%) and slowly increasing in Middle East and Latin America (8%) and the Caribbean (7%) (see Fig. 2.6). This has undermined the immense challenge of achieving zero-hunger target by 2030. The uneven pace of economic recovery combined with the poor economic performance of many countries after the 2008/09 global economic downturn seems to undermine efforts to end hunger and undernourishment [52].

Episodes of financial stress, elevated trade tensions among developing countries and tightening financial conditions are contributing to uncertainty. Hungry has increased in countries where the economy has contracted or has slowed down, and 52 countries out of 65 where adverse impact of economic slowdowns and downturns have been severe rely on primary commodity exports and imports.

This year, a new indicator specifying the prevalence of food insecurity has been introduced: the *Food Insecurity Experience Scale* (FIES). While severe food insecurity is associated with with hunger, people experiencing moderate food insecurity face uncertainties about their capacity to obtain food, and have been forced to compromise on the quality and/or quantity of the food they eat. Considering both groups,

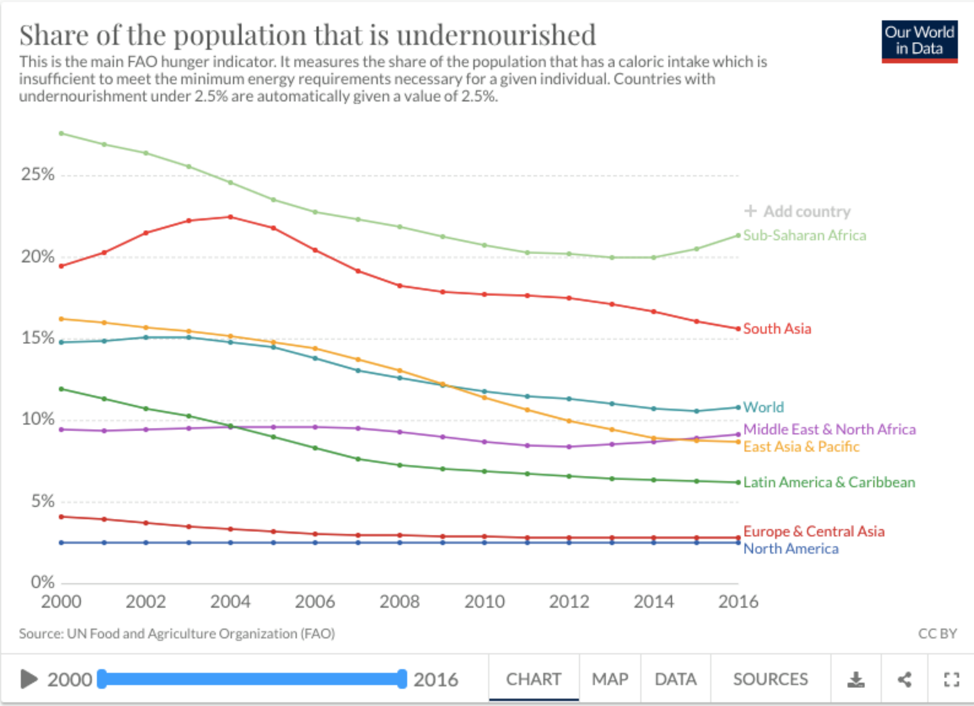


Figure 2.6: Undernourisment in the world [53]

it is estimated that over 2 billion people do not have regular access to safe, nutritious and sufficient food, including 8% of population in Northern America and Europe.

In order to palliate these problems, increasing global food production at least by 50% will be required by 2050 [2]. In this sense, globalization in food and agriculture has demonstrated to hold promises as well as present problems - it may have led to progress in reducing poverty in certain areas but, on the other hand, it has enabled the rise of multinational food companies with the power of demotivating farmers in many countries [51].

Safeguarding food security and nutrition in the longer term will only be possible by developing legal and administrative frameworks to ward off the threats while distributing benefits equally among the people. In this sense, it is critical to already have policies to counteract adverse economical effects when they arrive while avoiding cuts in essential services at all costs - including food. This is only possible by fostering the pro-poor people by more inclusive structural transformations, as it has been claimed that growth in the agriculture sector is two to four times more effective in raising incomes among the poorest than any other sector [50, 54].

Traditionally, three have been the major actions for growing crop production: expanding the land area, increasing the frequency with witch it is cropped (often related to irrigation) and boosting yields. However, a careful examination suggests that the world may be near of reaching the ceiling of what it can bear, and this view is not possible in all countries:

1. **Less new agricultural land will be opened than in the past**: At global level, there is enough unused potential land (up to 2.8billion ha) with varying degrees for the rain fed of arable crops, which is twice as much as it is currently farmed. But only a fraction of this area is realistically available in a foreseeable future, since much is needed to preserve forest cover and to support infrastructural development.
2. **Irrigation is vital to the world's food supplies**: The role of irrigation is expected to increase further, probably expanding the irrigation area to 402 million ha by 2030, whereas the net increase in irrigated land is predicted to be much less than 40% of that achieved since the late 1960s. In this sense, water resources will be a major constrain in regions of Asia and the Near East and North Africa which will need to increase water usage efficiency significantly.
3. **Yield** growth will continue to be the predominant factor in the future. It is estimated that 80% of future increases in crop production will have to come in the form of intensification: higher yields, increased multiple cropping and shorter harvest periods

In general, agriculture accounts for 70% of water usage, and generates 25% of greenhouse gas emissions and waste [50]. In addition, the impact of climate change are felt by farmers around the globe in unprecedented ways, as extreme whether is affecting their productivity, cutting crops yields and livestock and driving down incomes. With such scenario, the agriculture sector will continue to be among the most vulnerable, specially in areas with high concentration of poor people, and advanced technology will be necessary to deal with such problems and improve food security.

2.1.4 IoT and Agriculture

In addition to advances in biotechnology and organic agriculture, the Internet-of-Things is another technology that will play a significant role in increasing agricultural productivity and satisfy the growing food demand. Not surprisingly, the IoT market in the agricultural sector was outstripped with $16 million in 2017 and, with a growing *Compound Annual Growth Rate* (CGAR) of 14.5%, it is expected to reach $50 million by 2025 [55].

As a consequence, new terms defining technological concepts within the agriculture sector have spread lately. *Smart agriculture* is just considered roughly any food growing practice performed with the help of IoT technology, whereas *E-agriculture* is regarded to the conceptualization and design of ways of using information and communication technologies in the rural domain, with a primary focus on agriculture [56]. *Climate-smart Agriculture* (CSA), on the other hand, is an approach to help people who wants to develop agricultural methods that respond effectively to climate change, by pursuing three main objectives - increasing productivity, adaption to climate change and reduction of greenhouse gas emissions, wherever possible [57].

For example, collecting crop data from sensors - such as temperature, rainfall, humidity, wind speed, pest infestation or soil content among others - serves to automate farming techniques, to take informed decisions to improve quality and quantity, to minimize risk and water waste or to reduce the effort required to manage crops, among other farming tasks. Similarly, the use of image analysis technology and agricultural drones, form part in *Geographic Information Systems* (GIS). In this application, the land is mapped digitally and combined together with both statistical and geodetic data for analyzing the soil and deciding, for instance, what and where to plant.

In general, the agriculture industry comprises several players that provide solutions for some, but not all, the stages involved in the agriculture chain. This is because the economical feasibility vary from one agriculture processes to other and from one region to another. The global IoT market in agriculture is generally segmented by system and by application.

By system, products are classified into automation and control systems, sensing and monitoring devices, communication systems, hardware and software. Applications, on the other hand, ranges from precision farming, monitoring of *livestock, fish* or farm, to smart greenhouses. In 2017, software was the leading market as it consists of numerous tools to control hardware, such as computational elements, sensor units - soil moisture, temperature, light, humidity, pressure - which are crcuial for a wide range of precision farming, livestock, greenhouse and farming applications.

Some of the main market players of the global IoT market in agriculture include: Libelium [58] designs and manufactures custom hardware for wireless sensors commonly used in IoT applications; Growlink [59] is an agriculture technology company that offer tight wireless solutions for integrating control, data collection, monitoring of crops in a closed platform; Microsoft have developed FarmBeats, which is another platform of smart agriculture, used in combination with Azure (Microsoft's IoT platform), and end-to-end approach between sensors and the cloud. Other companies committed to smart agriculture with more or less degree are: IBM, Cisco Systems - Inc, Telit, Dragino, Decisive Farming, Farmers Edge -Inc, among others.

By application, precision farming dominated the IoT agriculture market in 2017 accounting for around 45%. This application consists in taking measurements in crops and livestock and act in near real-time with the goal of optimizing production and saving energy. The livestock monitoring segment, on the other hand, which includes the use of GPS and sensors, is projected to be the fastest growing sector in the future.

- **Case study 1: Smart Strawberries Crop Increases the Quality and Reduces the Time from Farm to Market** (Fig. 2.7). In Colombia, local organization Red Tecnoparque Colombia has deployed a wireless sensors network with Libelium technology to monitor their banana crops [60]. As illustrated in figure, Libelium's sensors allow producers to monitor key parameters including humidity, temperature, soil moisture, soil temperature, trunk diameter, fruit diameter, pluviometer and solar radiation.

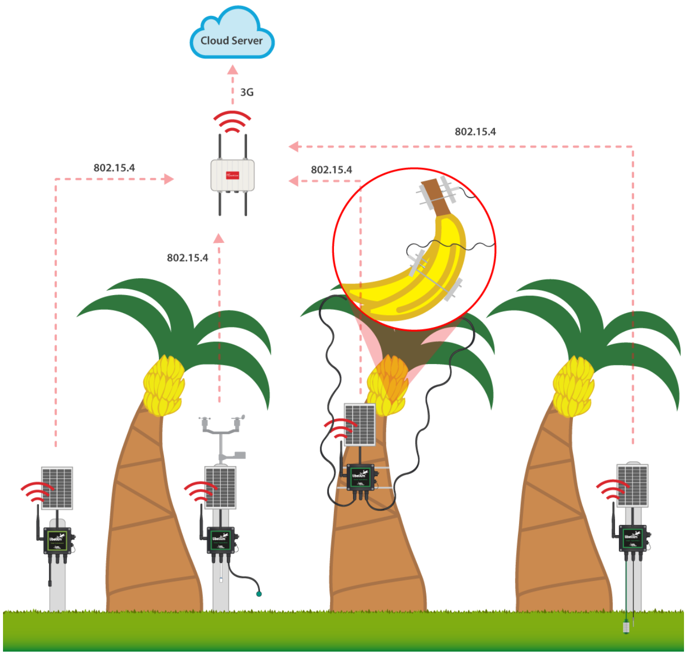


Figure 2.7: IoT system for banana fields [60]

- **Case study 2: Use only the amount of fertilizer they need** (Fig. 2.8). Vodafone's Precision Farming designed a GPS Farming device which is installed in the vehicle spreading the fertilizer and sends data to a server, therefore, the farmer can see where every kilogram has been deposited [61].

Figure 2.8: Precision fertilizer [61]

- **Case study 3: Data regarding the location, well-being and health of the cattle** (Fig. 2.9). Accent system provided a tracker to place into a cow to give its geoposition in real time. With this device, farmers save money detecting problems such as a cow falling ill to separated from the herd and prevent the spread of disease [62].

Figure 2.9: Cow geoposition device [63]

By region, Asia-Pacific accounted for the largest IoT agriculture market in 2017, basically due to the rise in population coupled with food demand, also explained in section 2.1.3. But another important factor has been the technological development of countries such as China, India, South Korea and Japan, among others.

In general, many of these technological solutions are out of reach for many farmers who operate as medium or small companies. The biggest barrier is the cost of infrastructure, where the investment can reach several thousands of euros. So this may be one of the reasons hampering advances in agriculture in some regions. Nevertheless, increase in smartphone users and the penetration of Internet among farmers boost their awareness about latest developments and hope to drive market growth in the future.

This chapter reviews the different elements involved in an IoT-based platform. Section 2.2 gives a close look on how a network architecture should be to make possible IoT-based applications in agriculture. The elements that make up such a platform are discussed in subsequent sections with a focus to those that have been selected for this project. Section 2.4 presents some case studies of current IoT platforms. These platforms are become very popular recently and users can design and configure its their own remote web interface for sending sensor data after creating an account. They are the starting point of discussion for the current IoT platform of this project. Section 2.6 deals with the sensor elements and the computational units which are responsible of picking up data from the physical world, and the gateways which are necessary to provide access to the Internet. Section 2.5 addresses the software available to develop server-based applications in the cloud, including programs for managing databases and data-frames, web design and different communication protocols. The last section (2.7) discusses our section for this project. In order to create a prototype at the reach of any consumer, components have been selected from different providers, so that updating software with new functionalities and scaling the network, without paying too much, can be possible in a foreseeable future.Thanks to this, not only farmers but also consumers that do not want to make a profit, but who share an interest in agriculture and technology, will be able to make use of this new and efficient ways of production.

2.2 The IoT Architecture

One of the biggest challenges when developing IoT is the diverse nature of the things that it aims to connect. In order to develop a robust and integrated solution, the network's architecture needs to serve its design purpose. Although each IoT system is generally different, its general data process flow remains roughly the same [64].

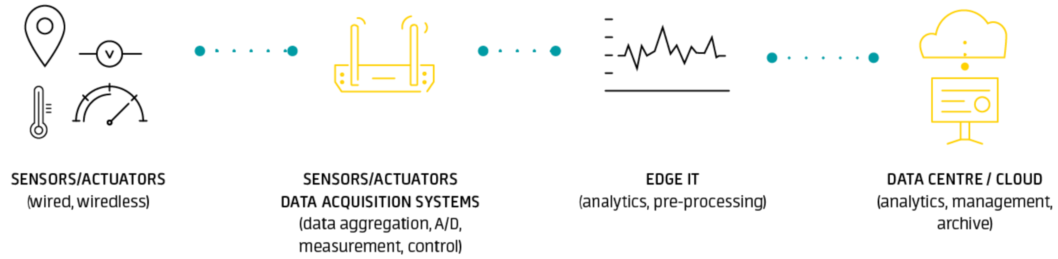


Figure 2.10: IoT general data process [64]

Figure 2.10 illustrates the main four layers involved in IoT data processes.

- The first layer is basically all the things to connect: the embedded sensors that sense the environment and the actuator which transform data into physical actions.
- The second layer consists of the gateway and data acquisitions systems. They acquire, process, and filter the data collection for the following layers, simplifying the communication and implementing security mechanisms.
- The third layer is based on edge analytics system for further processing and data enhancement, such us machine learning.
- The fourth layer is the data center, designed to store, process, and analyze the information. This layer usually includes interfaces for people to interact, control and monitor with the system in real time.

Putting any IoT system to work means joining resources together - including hardware, software, electronics and systems - no matter how heterogeneous they all may be, to form an integrated, reliable and cost-effective solution. The resulting efficiency and applicability of the system will largely depend on the quality of the selected elements in the infrastructure. Therefore, some considerations about the elements of the different stages are made in the following subsections to guide the selection of the devices used in this study. Sensor and actuator considerations are addressed in section 2.6.

2.2.1 The Things

As the basis of any IoT system, embedding sensors to small computational units provide its essence: the data. To pick up information within an object an interact with the environment, sensors and actuators are required. These elements are normally embedded together with the micro-controller in the printed circuit board (PCB) or connected separately as another object through the available pins. In agriculture applications sensors obtain parameters in crops such as temperature, soil moisture and pH, exposure to sunlight, soil moisture, among others.

Being in close collaboration with sensors, actuators use the data to generate physical actions. For example, a watering system can analyze the situation based on the input provided by the sensors in real-time and open the water valves in the places where soil humidity is below a threshold value. This process, of course, is automated and happens without human intervention.

Another necessary feature of sensor nodes is that the computational element should not only be able to communicate bidirectionally with the nearest gateways but also being able to recognize other nodes in the vicinity, and talk to each other to share information and collaborate to leverage the value of the whole deployment. Achieving this in battery-operated devices is not an easy task, as operating in this way requires lots of computing power and consumes precious energy and bandwidth. So a robust architecture can operate efficiently only if communication protocols are secure and lightweight. This is why M2M mechanisms take part in IoT platforms (refer to section 2.3).

2.2.2 Data acquisition systems

Although gateways are near the sensor, its function has to be considered apart, as it is crucial for the processes of data collection and transfer to edge and cloud-based systems. Given the massive volume of data that dozens of node may generate in the platform, tools for the aggregation, metadata filtering, selection and transportation of data are often necessary.

As intermediate devices, gateways and data acquisition systems provide the bridge to the rest of the elements involved in the architecture and to Internet. Which means that they convert sensor data into formats that are easily transferable and usable, minimizing the amount of information that needs to be forwarded, affecting positively to transmission costs and response times.

Another important feature of gateways is security. Because they manage direction flow in both directions, by using proper encryption programming and security tools, they can prevent IoT cloud data leaks as well as malicious external attacks on IoT nodes.

2.2.2.1 Raspberry-Pi

Raspberry-Pi is the brand of a series of small single-board computers (Fig. 2.11) which have been created to promote teaching of basic computer science in schools and developing countries in the last decade. All models feature a Broadcom system-on-a-chip (SoC) with an integrated ARM processor and a *Graphic Processor Unit* (GPU).

The boards feature one to five USB ports, a tip-ring-sleeve jack for audio output and lower-level digital GPIO pins which also support common protocols, such as Serial Peripheral Interface (SPI), *Inter-Integrated Circuit* (I2C) or asynchronous RS232 serial port.

The processor speed ranges from 1.4GHz (Model Pi3B+) to 1.5 GHz, for the Pi4, with up to 4GBytes of RAM memory. The board is managed by an operating system stored in a SDHC micro-memory card, among which *Raspbian* is generally used. It also supports HDMI video output, 8P8C Ethernet, WiFi 802.11n and Bluetooth, which can be programmed through the Python integrated environment. These features also makes the Raspberry-Pi a good candidate for developing gateway or even and edge system applications based on IoT.

Since the Raspberry-pi can be compared to a board computer, it can be a good candidate to be used as a gateway hub. Nevertheless, some sort of lightweight wireless connectivity is needed to provide connectivity and data transfer with node sensors. The way to do this, is to acquire an additional board, very often named as shields compatible with its 34-pin connector, implementing some of the radio-frequency system listed in subsection 2.3.1, and install the drivers.

Figure 2.11: The Raspberry-pi embedded platform (Model 3b+) [65]

2.2.2.2 Dragino

Like the Raspberry-pi, the Dragino (Fig.2.12) is another low-cost computer platform which has full Ethernet and 802.11/b/g/n WiFi capabilities [66]. It runs under a Linux-based OpenWrt operating system, which adds router features to the platform. Solving, therefore, the connectivity problem of computer-based sensor nodes to the Internet and enhancing micro-controller developments, such as the Arduino.

It can be used in many applications such as remote control of robots, data logging, web development for data presentation, mesh networking, WiFi and many more. The manufacturer provides two different firmware: *Arduino Yun* and *Mesh IoT firmware*. The first, is derived from the official Arduino Yun platform, as this board has been used to implement the router, specially in the version products LG01-P, LG01-S and OLG01. The second one include support for mesh networking which is very helpful for developing IoT-based applications with several nodes.

All the products incorporate LoRa, the long-range transceiver LoRa which has become widely used as a method for communicating sensor data among end nodes and computer platforms in IoT applications

Figure 2.12: LG-01 Dragino product [66]

2.2.3 Edge Systems

While not being an essential component of the platform, edge end devices bring benefits in IoT architectures of large value. Whether transfer of data to the cloud or data-center is slow, edge systems can provide quicker response and flexibility in the process of analysis of IoT, when the network is operating offline. For this reason, edge computing has seen a significant increase in IoT ecosystems, and specially in the agriculture sector.

As well as gateways, they are normally located near the physical nodes, but providing actionable intelligence, forwarding to the Internet only the larger chunks of data which really need the processing power of a server-based computer. This task can sometimes are implemented in the gateway, when it has enough processing capability. Security is generally enhanced if programmed in this stage, and therefore minimizing network exposure and contributing to leveraging business resources.

2.2.4 Server/cloud computing

Servers are generally considered the brain of the whole IoT body. They implement the remote services requested by users: data storing, monitoring and remote control of sensors based on web interfaces and additional analytical and management tools, among others. In this sense, both data-centers and cloud-based systems serve this purpose.

On one hand, web applications can be uploaded or programmed remotely in the data-center, depending on the service given by the web domain provider or, on the other hand, built-in functions can be configured if a cloud-based server is chosen. Analyses of massive volumes by means of statistical tools or even machine learning methods, which nodes or edge systems would never be able to support due to power consumption, can be tasks programmed in such systems.

2.2.4.1 Cloud vs. Data-center

The main difference between data-centers and cloud platforms is that the first refers to *on-premise* hardware development whereas the later refers to *off-computing* software. While both have their benefits, it ultimately corresponds to developers which one fits best their needs.

For many small companies with limited budget and lack of resources, using the cloud makes much more sense as it gives immediate operability without the need of a significant investment. The scalability of the cloud allows to add or reduce storage capacity and resources when needed since, in principle, it does not require investment on-premise hardware. As such, cloud platforms can reduce the cost of ownership and simplify the environment. Flexibility, ease of use, development speed, and agility or even software update maintenance are other attractive characteristics of the cloud.

On the other hand, although maintaining an in-house data-center may seem expensive, this can sometimes be a better option. For example, businesses using the cloud depend on external services for cyber-security, reliability and performance, whereas using an in-house data-center lets you have full control of such features. This is preferable when running mission-critical applications or needing of strict compliance requirements. In the end, flexibility, workload and security needs will dictate whether a data-center or cloud is the best choice.

2.3 *M2M* and IoT Communication

M2M stands for *Machine-to-Machine* communication which consists in the automatic exchange of data between two devices. Such systems can range from serial connection - such as, RS232, I2C and SPI -, industrial wired communications (such as DeviceNet, Ethernet/IP, Modbus or ProfiNet) to wireless communications - such as Zigbie, Bluetooth, WiFi, GPS, LoRa - among others.

The idea of connecting and communicating two devices to each other, without human interaction, is not new and has been evolving not only in the sense of technological developments but also concept-wise. As the number of connected IoT devices increases, this view of technology is in constant need of dedicated solutions for its proper development. In a world of disparate device standards, the demand for easy and effective communication systems has become stronger.

Just like IoT, M2M allows virtually any sensor to communicate, bringing the possibility to monitoring themselves and responding automatically to changes in the environment with reduced human involvement. The difference is that IoT generally refers to wireless (or even cellular) communications whereas M2M refers to two devices - whether wireless or not - communicating to one another. For example, MQTT is a form of M2M protocol that enables a publish/subscribe messaging model in an extremely lightweight way. It is useful for connecting to remote locations where a small code footprint is required and/or network bandwidth is at a premium [67]. Other examples of data protocol used in IoT are LWM2M, CoAP, AMQP or HTTP, between others.

Wireless M2M has been dominated by the cellular market since 2005 when the 2G networks came out. For such reason the cellular market has tried to brand M2M as an inherited cellular product by offering M2M data plans. However, M2M is only one part of this market and shouldn't thought of as a cellular-only area. So, essentially, M2M networks operate similarly to LAN or WAN networks with the difference that are mainly used in sensors, machines and controls.

2.3.1 M2M Communication systems

To make possible getting digital data across a network, received data has to be evaluated continuously in the context of the progress of conversation. So the communication mechanism must follow specific rules with the syntax, semantics, and synchronization, as well as tools for error recovering so that sender and receiver can understand each other [68].

Seven main wireless mechanisms coexist in the market of IoT applications [69]:

- Satellite. It is the medium used for GPRS, 2G, 3G and 4G mobile systems. the information is transmitted from the publisher to the next antenna and goes to the client through satellite.

- WiFi. It is a popular low-power wireless communication system for internet access used in local and wide area networks (WLAN/LAN).
- Radio Frequency. It uses radio wave signals which travels through the air medium and carry the data. In general, data is encrypted before being broadcasting using an encryption method to protect information- frequency modulation (FM), frequency shift-keying (FSK) or phase-shift keying (PSK). The receiver must demodulate the received signal to get the data. The radio frequency refers to the oscillation signal used in the modulation/demodulation stage.
- RFID. It uses electromagnetic fields to identify and track tags attached to objects. It used to develop sensor applications rather than in communication tasks.
- Bluetooth is a wireless for connecting various devices, exchanging data over relatively short distances with significant rate eliminating wires.
- Near Field Communication (NFC) is a wireless technology used two communicate devices within short ranges (4cm) used in contact-less payment applications, electronic tickets and mobile payment.
- Zigbee/LoRa. Zigbee and LoRa are both low-power wireless methods also based on spectrum modulation techniques (RF). LoRa includes mechanisms in the protocol to create wide-area networks.

The selection of one mechanism is conditioned by the application needs and goals. Section 2.7 addresses this matter for the current study.

2.3.2 Protocols and message mechanisms

In addition to the medium, another characteristic has to do with the *language* they use when messages are exchange. The nature of communication, the exchanged data and any state-dependent behavior is defined in the protocol. In the digital systems, the protocols contain communication algorithms and data structures[67].

2.3.2.1 Network protocols

In an scenario where communicating devices share the medium but may have individual operating systems and different hardware, transmission is not necessarily reliable. In a network system, its software modules interface with the framework used by the operating system of the machine in charge of the networking functionality. When developing the protocol algorithm in a portable programming language, the software used may be operating-system independent.

By the time the Internet was developed, abstraction layer had proved to be the best approach for both compiler and operating system design. So given the similarities between programming languages and communication protocols, the networking programs were decomposed into cooperating protocols.

For these reasons, networking systems do not use a single protocol to handle transmission. Instead, they use a set of cooperating tools, which is named as *protocol suite.* Some of the best known protocol suites are TCP/IP, IPX/SPX, or X.25, among other. All they follow the *Open System Interconnection* model (OSI)[70] and different protocols included in the TCP/IP, some of which are used in IoT applications [71], are:

1. Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL)
2. Identification (ex: EPC, uCode, IPv6, URIs)
3. Discovery (ex: Physical Web, mDNS, DNS-SD)
4. Data Protocols (ex: MQTT, CoAP, AMQP, HTTP)
5. Device Management (ex: TR-069, OMA-DM)
6. Semantic (ex: JSON-LD, Web Thing Model)
7. Multi-layer Frameworks (ex: Alljoyn, IoTivity, Weave, Homekit)

The TCP/IP protocol suite has mechanisms for the management of machine identification, where the most common is known as IPv4/IPv6 [72]. The IP address is a numerical label assigned to each device to identify the device connected to the network [73]. Each IPv4 is unique and uses a 32-bit word format, which can be automatically or manually configured. Its structure consists of four octets separated by a period, as illustrated in figure 2.13.

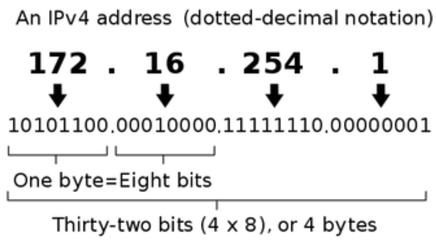


Figure 2.13: IPv4 Address Structure [74]

2.4 IoT third-party commercial platforms

An IoT platform is a multi-layer web-based technology that enables remote and straightforward provisioning for helping the developer to manage his connected devices within the IoT universe. It connects the user's hardware to the cloud, no matter how diverse, providing flexible middle-ware for the different connectivity options, enterprise-grade security mechanisms and data-processing functionality, among others.

In an heterogeneous IoT ecosystem, this middle-ware is intended to support integration of any connected device and blend third-party applications to it. Currently, there are many different IoT commercial platform in the market with different prices depending on the features which are hired. In what follows, two of the most popular free-license platforms are reviewed briefly.

2.4.1 ThingSpeak

ThingSpeak [75] (Fig. 2.14) is an open source IoT analytics platform server that allows to store and retrieve data from devices using HTML and MQTT protocols, via a LAN network, and permits to aggregate, visualize and analyze live data streams in the cloud. Sensors as well as location tracking are some of the applications that can be developed with this platform.

This platform use different channels to specify remote the sensor nodes, and the variables are represented in different fields. Since it has integrated support from MATLAB software - from Mathworks - registered users can analyze and visualize uploaded data from their devices.

However, the free license is limited to four channels and eight fields, which means that only a total of four devices, with eight sensors each, can be monitored in this platform. The update interval of each field is also limited to 15 seconds, with a maximum of 255 characters in a single post, and 3 million messages per year.

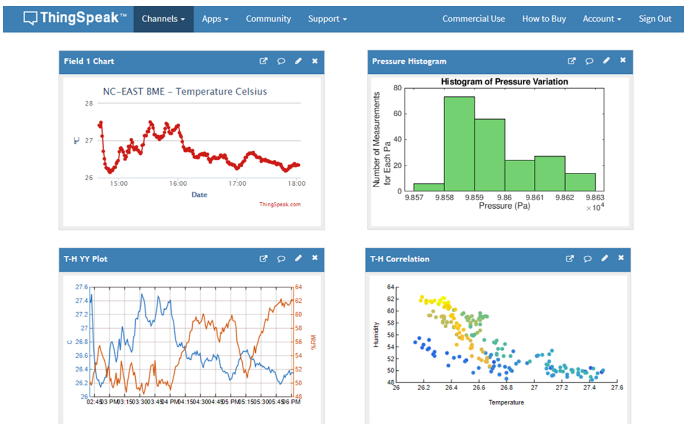


Figure 2.14: ThingSpeak IoT platform [76]

By purchasing additional services, ThingSpeak offers data visualization in real time, to use MATLAB functions in the server to analyze the data, run data analytics based on schedules or events, develop web software or automatically act on your data through third-party services like Twitter. However, this platform does not apply any encryption on IoT device data.

2.4.2 The Things Network

The Things Network (TTN, Fig. 2.15) is considered the first open-source decentralized network server for developing IoT around the world, by providing tools which allow users to build an low-cost applications, sharing resources, and featuring maximum security and scalability[77].

Currently, this platform supports LoRaWAN wireless technology, which outperform other systems for the long range (from 5 to 15kms) using low power, which can increase battery life to months or even years, but also using low bandwidths (51byte/message) is possible. The LoRaWAN specification is built on top of the LoRa technology, - developed by the LoRa Alliance -, and adds networking capabilities to the sensor nodes spread in the field.

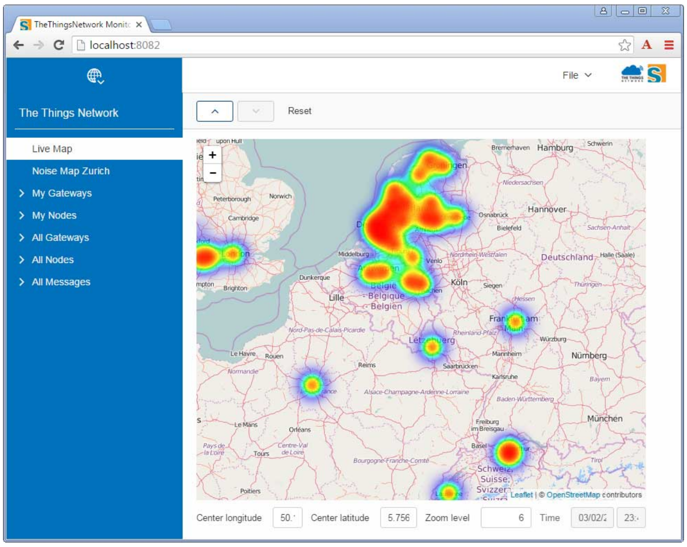


Figure 2.15: The Things Network platform [78]

To connect a device to the TNN infrastructure, the sensor unit needs to have a LoRaWAN module, either on board, as a shield or wired, which has to be registered to the platform beforehand. These modules are normally connected to the node controller through synchronous protocols such as SPI or I2C, and are talked via RS232 serial USB port. Proprietary devices of TTN, in this sense are the Things Node and the UNO Node, but other certified LoRaWAN can be operated with this platform. The Arduino SDK library enables to send messages by typing a couple lines of code.

As mentioned in section 2.2.2 gateways form the bridge between the device radio protocols and the Internet. Since LoRaWAN operates on unlicensed bands (see section 2.7.1.1) running any custom LoRaWAN gateway is completely legal, as long as the hop frequency of each country is respected.

As LoRaWAN is a Non-IP protocol, it requires some sort of routing and preprocessing before any message can be sent. However, there are many products containing both the IP stack and the LoRA transceiver in one single device which contains the necessary tools. So these gateways only have to forward packets to the Internet and can be placed outdoors adding more access points, in case land coverage is needed. This is the case with the LG01 router from Dragino [79] and whose cost is about 65€ per unit.

One of the limitations is that each device can only operate at one single frequency channel so the bandwidth may be constrained in certain applications.

2.5 Web server development and programming tools

An alternative to developing an IoT services in the cloud is to create one from scratch using other web server tools such as the LAMP service stack (Fig. 2.16). LAMP stands for Linux, Apache, MySQL and Php/Perl and contains a set of free and open-source web development tools.

The modularity of this package may vary, but this particular has become popular because it is sufficient to host a wide variety of site frameworks. Its characteristics are the following characteristics:

- Linux is a UNIX-based free open-source operating system on which LAMP is based. There are different LAMP versions that can be used in other OS platforms, such as Windows (WAMP) and macOS (MAMP).
- Apache implements the LAMP's web server. It is considered one of the first and most reliable servers, powering around 53% of websites worldwide[81] and embracing the philosophy of free and open-source software as well. In the last distributions, Apache supports a wide variety of features as compiled modules which extends the core functionality of Apache. Server-side programming, authentication schemes, encryption or even new other server alternatives, such as Nginx are a few. Thanks to these tools website owners serve content and deliver digital files when people surf the Internet.
- MySQL is a component used to give to web pages the profile of a database-driven application using Structure Query Language (SQL). This language is used to view information in databases as well as changing their content. The statements, or line codes, used in such operations are called SQL queries.
- Php is a server side scripting language designed for web development. When web pages are designed in HTML code execution is done at the user's browser (client-side) whereas with PHP execution takes place at the server side, which allows web pages to be implemented dynamically. This feature is necessary when developing web pages for e-commerce applications or to dialogue with databases, such as MySQL or Oracle among others.

There is a wide range of additional tools that add powerful utilities that complement LAMP. Some interesting, although not the only ones,are:

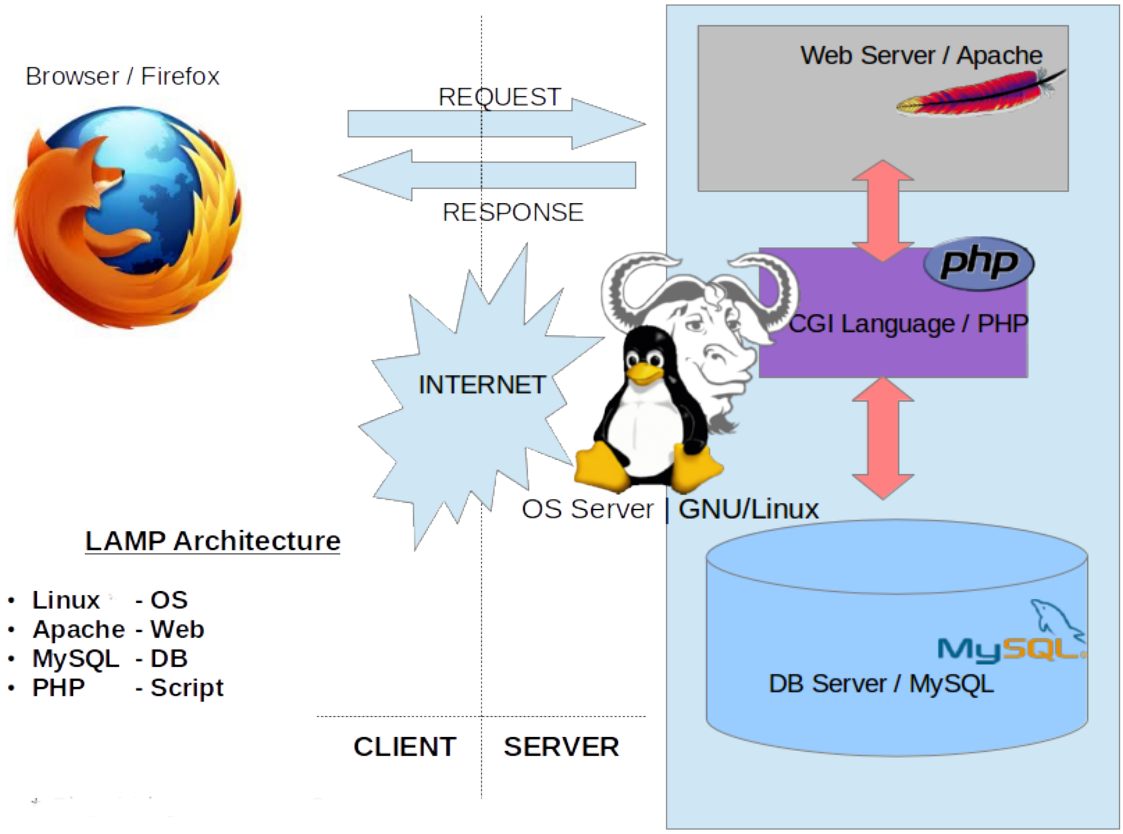


Figure 2.16: A high-level overview of LAMP's determining components [80]

- *Graphic User Interfaces* (GUI) development tools simplify the creation of web pages by allowing the designer to arrange graphical elements from the the front-end side of the page-text, images, videos, buttons and other control elements - by combining HTML, CSS and JavaScript programming languages and the use of drag-and-drop editors..

- *Node-RED* is a free flow-based development tool for visually wiring together hardware devices to APIs and web services and software as part of the Internet of Things. As such, it is a flow-based development tool. It can also be installed in hardware devices such as Raspberri, Arduino, Android or Beaglebone Black.

- *phpMyAdmin* is a free and open source administration tool, written in PHP language, with the goal to handle the administration of MySQL over the web and simplify the remote management of databases. Very frequent SQL operations can be executed through the interface - tables, columns, relations, indexes, users, permissions, etc - while still having the ability to execute SQL statements from the command line. phpMyAdmin supports additional features, such as: import/export data in various formats (CSV, SQL, XML, PDF, XLSS, LaTex and others), Administration of multiple servers, graphic layout in various format, support for MariaDB and many more.

- DNS support tools. DNS stands for Domain Name System, and DNS services translate domains names into IP addresses [82]. This feature is necessary to make the web domain accessible every time the server machine hosting it is being turn off, without having to store the old IP. There are different license-free services, such as Duck DNS or no-IP, which are able to translate a PC's IP address into a subdomain in order to make it accessible from the Internet.

2.6 Agriculture sensors nodes and irrigation systems

In agriculture applications, when monitoring the crop conditions to obtain the conditions of the land and control the water irrigation system, sensors and actuators must be embedded as part the computing devices. In [83] the five basic factors in order to grow vegetables adequately are light and temperature, air, water, soil, and energy. A brief explanation of each of the element could be found below:

- **Light and temperature**: The duration of the solar exposure influence the plant activity, determining the beginning and the duration of the vegetative phases - germination, development, flowering, fruiting and ripening. So in order to controlling the yield, knowing the amount of solar exposure hours, light intensity and the temperature reached is important.
- **Air**: Proper ventilation of the field is crucial to clean air and plants. A poor ventilation drives into too humidity excess and, therefore, the appearance of fungi. On the other hand, too much ventilation could dry the soil or break branches and stems. The effects of the wind in agriculture are many. On the positive side it transports not only pollen and organic matter but also cold or hot air masses and clouds. But this phenomenon is more related to the damage it causes, mainly, those which are visible: damage of crops, fall of leaves, fruits or even knocking down trees.
- **Water**: Agricultural water is used in irrigation, pesticide, fertilizer applications, crop cooling and frost control. Plants require certain amount of water, preferably provided by the rain. If the amount of rain water is not enough, irrigating systems must be taken into consideration. Additionally, good drainage systems must be installed.
- **Soil**: The soil is classified considering its texture and the size of the particles, and can be sandy, silty and clayey. Sandy soils have the thickest texture among all three. This influence the type of vegetable planted. In this sense, wheat, beans, potatoes or rape-seed crops among others, grow very good in clay soils. Heavier and wetter soils are more suited for barley, whereas carrots and beets adapt better to sandy soils. Implementing good practices are also important for improving the quality of the soil. The best way is to work the

land with a hoe. Some fertilizer in the form of decomposition material before the cultivation period is also of great help.

- **Solar Energy**: It can be used for saving money, increasing self-reliance and reducing pollution. by reducing the farm's electricity. Solar heat collectors can be used to control crop temperature as well as warm homes, livestock buildings and greenhouses.

It is important to note that many of the previous characteristics vary on each season and the place the crop is planted. It is necessary to pay attention to critical moments such us the warmest, coldest or rainiest days in order to take control of the most critical situations. Therefore, knowing: temperature, humidity, soil moisture and solar radiation is basic for determining and controlling the above characteristics

These field's conditions are controlled through sensors and actuators connected to an embedded system which contains the micro-controller (MCU).

2.6.1 Arduino Boards

Arduino is currently one of the most popular solutions and offers diverse boards with different micro-controllers. It is an open-source platform aimed at developing prototypes for people with little knowledge in electronics, based on a easy-to-use hardware and software . The different boards are able to read inputs - whether in analog or digital format - and turn it into a control action (or output) - activating a motor, turning on a led, or publishing something online or written in a file. This is done by sending a set of instructions to the processing unit on the board.

Over the years, Arduino have been used for markers, scientists and programmers worldwide to build low cost instruments, prove physics principles or just to get started in programming and robotics. As soon as it reached a wider community, its products have evolved from the the simple and low-cost UNO board, to units with high processing power or even IoT capabilities such as the NANO 33 IoT (Fig. 2.17). A main feature is that it uses a high-level programming - know as *wiring* - which abstracts from low-level machine code, what makes it a clear, simple and easy-to-use programming language. However, this comes at the price of reducing processing speed and increasing consumption power compared to other platforms.

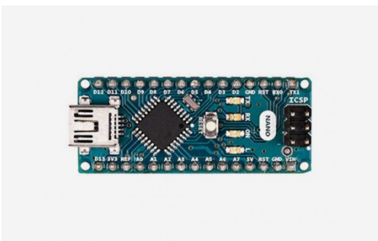


Figure 2.17: Arduino NANO 33 board [84]

2.6.1.1 Other platforms

- Waspmote

Waspmote (Fig. 2.18) is another open-source wireless sensor platform from Libellium (Zaragoza - Spain) focused on the implementation of robust and low-consumption nodes, also known as "motes", to be completely autonomous and battery powered with a lifetime from one to five years.

As Arduino, Waspmote follows a modular philosophy designed for different application fields related to IoT, including agriculture sensors. It also feature different low-power RF transceiver, including LoRa, SigFox or Zigbee.

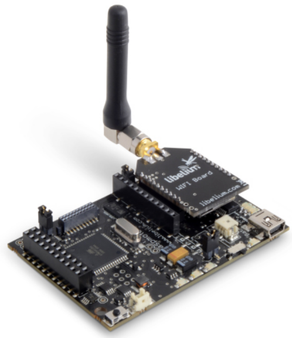


Figure 2.18: Waspmote sensor node [85]

- ESP32866

This board uses a low-cost WiFi integrated circuit with full implementation of the TCP/IP stack and micro-controller capability manufactured by Espressif Systems (Shangai - China). This unit allows to connect to a WiFi network and access directly to the Internet without a laptop computer.

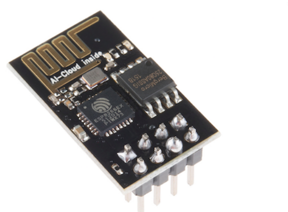


Figure 2.19: ESP8266 module [86]

Its very low price (less than 5€) and the fact that it uses very few external components, which make this board really small, attracted many hardware programmers to explore this module, translating the original Chinese documentation. Nowadays this module has significant library support, including for SQL. In this sense, when the connected via WiFi, SQL queries can be instantiated to a server from this small module.

2.7 Discussion

After the review done in the previous sections about the elements that can form an IoT infrastructure it's time to indicate the selected ones for this work by analyzing its pros and cons.

According to the stages and IoT architecture can have (see section 2.2), the current platform contains a node (which gathers the "Things"), a gateway which deploys the data acquisition system and the server. Elements for implementing the edge analytics of the IoT layer is discarded due to the size of the current system and the scope of the project, since this project is intended as a development example of a smart-agriculture IoT system rather than a real application

One important decision has to do with the communication systems, and their medium, of the whole platform. In this application, this affect to two main stages: the transferring of data between the node and the gateway, and between the gateway and the server trough the Internet.

Another element whose selection needs to be discussed is the data acquisition system, and the software tools that make up the server application, as this last stage is developed in a laptop computer. Finally, another important hardware element is related to the elements of the sensors node: including computing platform, sensors and actuators, and the transceiver for communicating with the gateway.

2.7.1 Communication node-gateway

In agriculture applications where sensors are normally deployed in a field (outdoors), whereas the gateways that collects the data stay in indoor locations, generally wireless systems are used. Among this group, Table 2.2, summarizes the suitability of the main systems used in engineering applications.

Table 2.2: Communication protocol selection for the communication node-gateway

Communication protocol	Compatibility with the project
Satellite	This is the preferred option in areas of very difficult access, due to geographical profile of the ground, where no sort of communication infrastructure - whether wired or not,or mobile 3G or 4G is deployed. This system normally has high monthly cost and high power consumption. It's rarely used in e-agriculture applications unless it is strictly necessary.

Communication protocol	Compatibility with the project
WiFi	This wireless system makes more sense in the transferring of data gateway-server rather than node-gateway, as WiFi is most used for Internet access instead of point-to-point sensor data transfer. When the first issue is predominant, one consideration to take into account is the availability of access to web services in rural areas, which is not always as easy as one would.
Radio Frequency (RF)	This communication shall suit better in the the node-gateway transfer as it generally consumes little energy compared to other other mechanisms, such as Bluetooth, and need of less network infrastructure requirements. In this group, another choice is the modulation mechanism used by the this wireless system, which can be LoRa, Zigbee, SigFox or even standard RF modulation methods.
RFID	This system is intend for object identification applications and does nothing to do with this project.
Bluethooth	This system is used in device-to-device where removing wires is a key point in the application, and it could be suited in agriculture applications. Although it may be compatible with the node-gateway communication, this mechanism is generally intended for short distances spending significant power to achieve higher rates compared to other RF As such it is not considered in this project
Near Field Communication (NFC)	This communication protocol may not be compatible with the communication node-gateway considering the small distance needed between devices

Taking into account the considerations in table 2.2, the communication from the node to the gateway in agriculture applications is best using RF technology.

2.7.1.1 LoRa Radio Frequency

For this project LoRa has been chosen. LoRa (short for Long Range) is relatively recent wireless technology which enables low-power wireless wide-area networking. Technically, it is a modulation scheme for modulating RF signals - using license-free sub-gigahertz radio frequency bands like 433 MHz, 868 MHz (Europe) and 915 MHz (Australia and North America) [87] - to encode digital information using chirped multi-symbol format, enabling long-range transmissions (more than 10 km in rural areas) with low consumption. However, it also refers to support modulation for networking applications, including micro-controllers, gateways for IoT applications

While people sometimes refer to LoRa when they really mean LoRaWAN, actually, this last system adds Media Access Protocol (MAC) in top of the LoRA layer, also built using Semtech 's LoRA. LoRaWAN, however, is rarely used in private networks (such as industrial applications). Instead, it is suited for public wide area networks because all transmission channels are tuned to the same frequency.

In this sense, its is much better to have only one network of devices in the required area to avoid collision problems, without having to hop to a sub-band near the reference frequency. Because all the gateways in the network are tied back to the same server, the server decides which one should respond to a transmission. So, in a large network, any transmission is heard by multiple receivers, the server tells one gateway to respond and the others simply ignore the transmission.

This process avoids downlink and uplink collisions because a single gateway is always transmitting. What is not known so much, is that there is a way to use underlying LoRAWAN technology - which is LoRA - without using LoRaWAN. For instance, Link Lab's Symphony Link uses a proprietary MAC layer on top of the Semtech's transceivers SX1272 and SX1276 with additional features to securely connect IoT devices to the cloud.

Table 2.4: Pros and cons of LoRa technology

Pros	Cons
Long distance range	Non-international
Low power consumption	Slow
Low cost	Small data sizes
Include networking capabilities	
No need of an established Internet network	
Low infrastructure required	

There are plenty of companies using LoRa chips for other protocols. For this reason, LoRa is a good and serious candidate for using it to develop communications in Smart Agriculture scenarios. The pros and cons are summarized in Table 2.4.

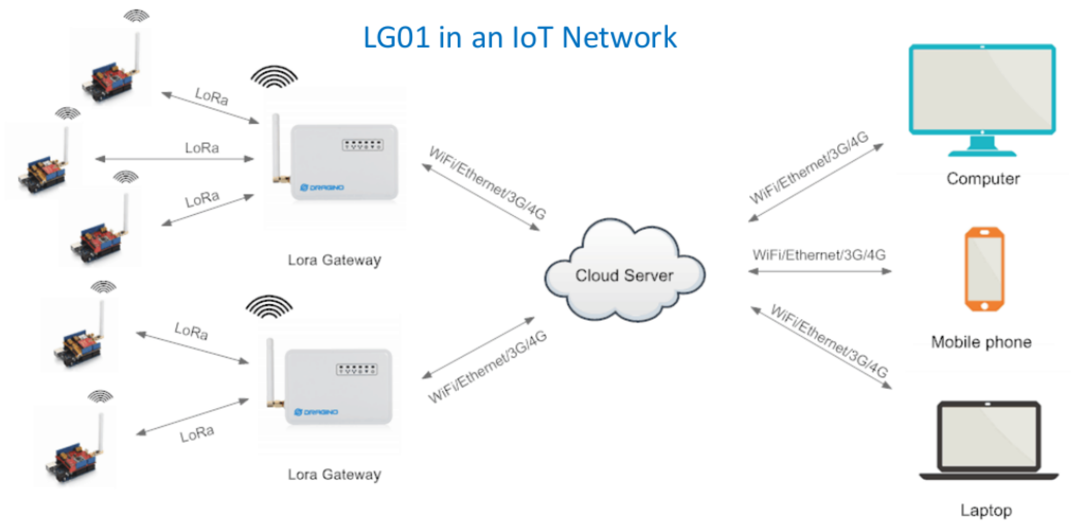
Other technologies compete with LoRa, these are Sigfox, LTE-M, Wize technology or narrowband IoT, among others. However, LoRa and also Sigfox gateways seem to dominate the market with many products which can be purchased for Arduino board controllers and Raspberry. All these reasons, and the fact that it is license-free, have made LoRa the choice of this project.

2.7.2 Communication gateway-server

The communication gateway-server is necessary so that the user can monitor the state of the field anywhere. So the the gateway is an element that needs Internet access for sending results and receive commands from the user. To this purpose, the gateway can use WiFi connection if some web service access is near the crop field or, otherwise, 3G/4G cellular transceivers will be necessary.

The Dragino company offers gateway solutions, such as the LG01-S , which features LoRa technology and allows the user to bridge sensor networks through LoRa to an IP network via WiFi, Ethernet, 3G or 4G cellular (Fig. 2.20). Therefore, it fulfills the system requirements and demonstrate the agriculture application by creating a bridge to send information between the sensors and the server [66].

Figure 2.20: LG01 in an IoT Network [66]



The node communicates with the Lora Gateway bidirectionally . It consists of an Arduino controller board with a LoRa shield plugged on it. In the server side, an application can be developed allowing the user to send and receive information from any personal device.

2.7.3 Server side selection

In this project the LAMP service stack for developing the web application is chosen. Unlike commercially available third IoT-based platforms, creating a customized platform, developers obtain full flexibility and no restrictions. Although this option can be time consuming , in a long term basis could help saving resources and cost.

In addition to the basic LAMP package, other tools have been used to speed development by simplifying transmission tasks and to including debugging tools. These are:

- The front end of the web page is developed under HTML5, CSS and JavaScript programming languages.
- Node-RED blocks are used to implement communication flow between the server and the gateway.
- phpMyAdmin administrates sensor databases and user commands.
- No-IP is used as a service to obtain a DNS for managing IP addresses.

2.7.4 Node components selection

On the basis of using LoRa communication and Dragino LG01, the Arduino platform have several shield which allow to develop LoRa-based custom applications relatively fast.

In addition, the Arduino UNO is a robust board and the most used and documented board of whole Arduino family [88]. It is a micro-controller board base on the ATmega328P, it has 14 digital input/output pins, 6 analog inputs, a USB connection and a power jack [88], which makes it an adequate option for this project. Therefore, this micro-controller platform was used to demonstrate the IoT capability of the platform of this project.

When this project was started, Libelium's micro-controller was also considered, as this company sells several motes compatible with LoRa and Sigfox communications. Unlike LoRa, Sigfox is not license-free and an annual pay must be done to maintain this license. However, the cost of Libelium was significantly above the solution based on the Arduino UNO and its LoRa shield, and was out of reach for this project. Nevertheless, it is worth mentioning that Libelium (and particularly the Waspmote and LoRa modules) has other important feature advantages, which should be taken into account when developing agriculture applications. Among which, it stands out the robustness of the platform in front of undesired weather conditions and the battery life.

According to the project's scope and time limitations, the platform uses a single node with two sensors and one actuator., The environmental variables measured are temperature, humidity and soil moisture. Since they are placed outdoors, a basic

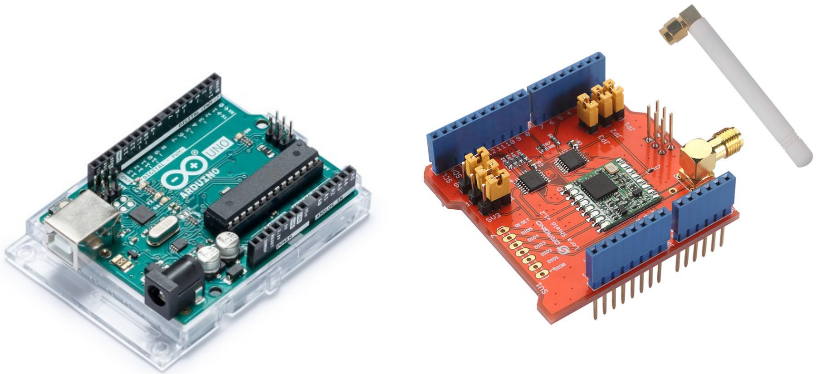


Figure 2.21: Arduino UNO board [88] (Left), LoRa shield and antenna [89] (Right)

criteria so that they can endure adverse weather conditions has been taken into account.

- The soil moisture sensor is the SoilWatch10 from Pino Tech [90]. It is a capacitive sensor which measures relative water content, and it is waterproof. Since the sensing principle is based on capacitive variation, it does not require exposure of the metal electrodes. Therefore, it lasts longer than the common resistive sensors.
- The Adafruit SHT-30 is the Mesh-protected Weather-proof Temperature/Humidity sensor selected for measuring air's temperature and humidity. The mesh encasing the sensor makes it weatherproof and keeps water from seeping into the body of the sensor, and allows air to pass through [91].

Figure 2.22: SoilWatch (Left) [90], Adafruit SHT-30 sensor [91] (Right)

Both sensors are prepared to connect the UNO board and manufacturers include practical use examples, providing the required materials and libraries to accelerate implementation and set up.

To complement the platform , an irrigation system consisting on a water pump, tube and water tank is also implemented. The kit model used is the JT180 from Jovtop, which includes a tube to transport water. The pump comes with two power wires (+V and ground), which just have to be connected to power to turn it on.

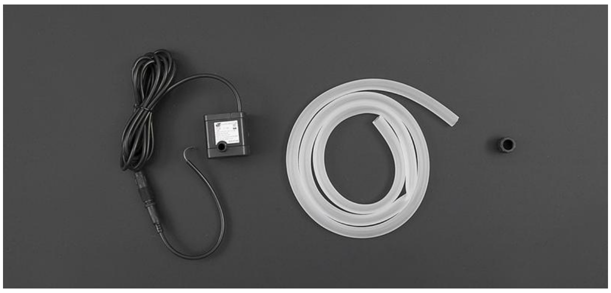


Figure 2.23: JT180 water pump [92]

In addition, in order to connect all the components together and test the electronic circuit, a preliminary breadboard has been used to connect jump wires and pin headers. As this device demands significant power when turned on, external circuitry is needed to actuate the valves and control the DC motor, since the I/O digital outputs cannot drive enough current and the Arduino board may be damaged. One possibility is to would be to use the L293N driver.

At the time of reporting this project this mechanism wasn't installed yet, as the delivery time of the supplier was delayed significantly and the valve system could not be neither set up nor prove.

Finally, separate elements for a powering system were purchased for feeding the node with the necessary voltage and current to operate correctly (Fig. 2.24). The kit consisted of several elements, including: a solar panel, a battery and additional components in order to manage the energy system:

- Adafruit Medium 6V 2W Solar panel [93]. It is a waterproof, scratch resistant, and UV resistant solar panel to cover the node's electrical power requirements.
- Adafruit USB / DC / Solar Lithium Ion/Polymer charger [94]. It is a charger to transfer the energy between the Solar panel, battery and micro-controller.
- Jack Adapter Cable [95]. For connecting the solar panel to the polymer charger.
- 2000mAh 3.7V Lipo Battery Pack [96]. Lipo Battery to store the energy obtained from the solar cells.
- PowerBoost 500 Basic - 5V USB Boost 500mA from 1.8V+ [97]. This device is a DC-to-DC boost voltage converter that transform the 1.8V from the solar cells to the 5V-DC voltage required by the Arduino micro-controller board.
- Adafruit Micro-Lipo Charger for LiPo/LiIon Batt w/MicroUSB Jack - v1 [98]. This device serves for charging the Lipo battery from a microUSB cable when there are no sunny days.

As with the power driver, the node's energy system could not finally be included for the same reasons.

Figure 2.24: Node's independent energy system [94]

2.8 Software and hardware selection

All the software, and hardware elements chosen for this project are summarized in Table 2.5

Table 2.5: Platform's Software and hardware selection

Product	S/H	Description
Linux	Software (Server)	Included in LAMP, it will be the operating system of the server.
Apache	Software (Server)	Included in LAMP, it will be the server to hold the web page.
MySQL	Software (Server)	Included in LAMP, it will be the server's database to store the information.
Php	Software (Server)	Included in LAMP, it will be the programming language to develop the server's functionality.
PhpMyAdmin	Software (Server)	Administration tool to manage MySQL database for development purposes.
CSS, HTML and JavaScript	Software (Web page)	Programming languages to develop the web site from scratch for the user to interact with the server.

Product	S/H	Description
Node-RED	Software (Data acquisition)	Layer implemented to simplify the communication between the gateway and server.
No-IP	Software (DNS services)	To translate the server's IP address into a subdomain.
LG01-S IoT Gateway	Hardware	Gateway which works as a bridge to bring data from the node to the Internet and vice versa.
LoRa Shield and antenna	Hardware	Shield to provide Arduino's micro controller with the capacity to communicate using LoRa protocol.
Arduino UNO	Hardware	Micro controller to read the data from the sensors, control the actuator, and manage the node's data.
Breadboard	Hardware	Device to connect weldless the node's components in order to test the electronic circuit.
Temperature and humidity sensor	Hardware	Adafruit SHT-30 Mesh-protected Weather-proof Temperature/Humidity Sensor .
Moisture sensor	Hardware	SoilWatch 10 3V - Soil moisture sensor.
Water pump actuator	Hardware	Immersible pump and water tube for the irrigation system.
Jump wires and pin headers	Hardware	Breadboard wires and pin headers to connect the different node elements.
Ethernet cables	Hardware	Ethernet cables to connect the server and gateway to the router (optional).
External power source	Hardware	External power source to fed the valve
L293N driver	Hardware	To control the current sent to the irrigation valve

3 Platform Design and Development

3.1 Overview

In this chapter, the design and development of the IoT platform of this project is illustrated, taking into consideration the products and technologies selected in the previous chapter.

It starts explaining the platform layout, including the different elements and its interconnections (section 3.2). Next sections take a closer look to each element's deployed hardware programs (sections 3.3 to 3.8), in order to achieve its communication with the rest of devices in the network. Additionally, it will include the node's electronic diagram for wiring the sensors and actuators within the breadboard (section 3.8.1).

The chapter concludes with an overview of the possibilities and limitations of the proposed solution in the current stage (section 3.9).

3.2 E-agriculture platform layout

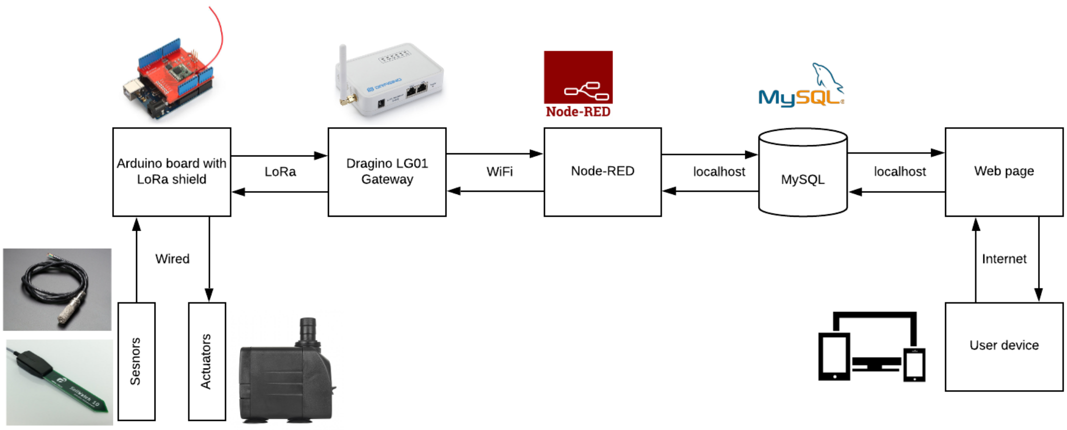


Figure 3.1: Developed IoT-based agriculture architecture

The developed IoT platform (Fig. 3.1) is based on an interconnected node, a gateway and server. The node contains an Arduino controller wired to two sensors devices -

a soil temperature/humidity sensors and a mosisture sensor- and and an irrigation valve.

The Arduino has attached a LoRa shield with its antenna, in order to communicate wirelessly with the gateway. The gateway, on the other hand, uses WiFi communication with the server, and operates as a bridge between the node and the server, which is based on one MySQL database.

In order to easily handle the commands and information sent from the gateway to the database, an intermediate Node-RED layer has been implemented. This tool serves to manipulate and refine the information in the system and improves the developer's debug experience. The MySQL database is in charge of holding and storing the data from the node, the gateway and the user.

Finally, the website presents the data and sends the user's requirements to MySQL database. Users access all this information from any personal computer with internet connection, and they can read the sensors, actuate the valves and change system settings.

3.3 Network configuration

In this project, the server is implemented in a laptop PC with a LAMP service stack application, which has three layers: the Node-RED program, the MySQL database and the web page. Since all three run in the same computer, rather than a server machine, to talk to each other they use an IP address hostname, which by default is "*localhost*". Localhost is a hostname that means "this computer", and it is used to access the network services that are running on the host [99].

Internet connection of the network is established wirelessly after login the Dragino platform through the WiFi transceiver to a home router, and so does the laptop computer. When connected, the Dragino and the PC receive a network IP address so that they can be identified within the internal network among different devices using the same router.

To wirelessly connect Dragino to the router, some changes need to be applied in Dragino's configuration. Check Dragino's manual [66] to access its configuration website and apply the changes shown in image 3.2.

Once the changes are saved, navigating to "Status" and "Overview", a new IPv4 address should appear, as illustrated in image 3.3. In this example, the router's IP address is 192.168.0.1, and Dragino obtained the IP address 192.168.0.20.

To ensure the connection has been established between Dragino and the server, the server's PC should successfully ping Dragino's IP address using the terminal (see figure 3.4)

It is important to notice that in a server's network using dynamic addressing, different IPs are assigned every time the Dragino, the home router, or the computer is

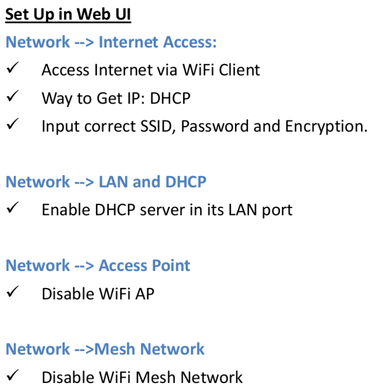


Figure 3.2: Steps to connect Dragino wirelessly to the router [66]

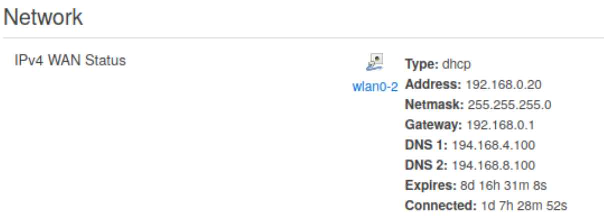


Figure 3.3: Dragino's network IP address

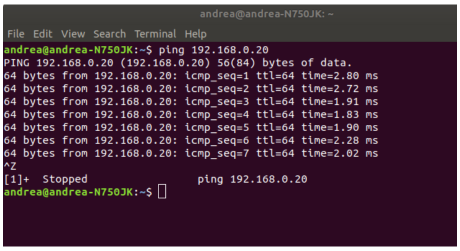


Figure 3.4: Server's PC pinging Dragino's IP address

turned on and off, causing problems to the different scripts involved in the gateway and the host PC to connect and access the internal network.

To solve this, and simplify the access to the server, a DNS name (*Dynamic Name System*) is required. The company No-ip offers the possibility to give a different hostname to the computer for free. This is done by, creating an account in their website, a dynamic host, and forwarding port 80 through the router configuration as stated in their guide [100]. When this service is installed, the IP address is detected automatically and associated to the new sub domain. Therefore, the gateway script just need to fix this sub domain path, to access regardless of the IP address from anywhere.

On the other hand, since the Arduino node cannot access to the Internet because of its remote location in the field, sending data to the gateway is done through the

LoRa wireless communication system. This feature is embedded to the Arduino by plugging the shield and antenna.

3.4 SQL database

One main element in the server is the database, which manages and stores relevant data from the whole platform. It has been developed using the MySQL toolkit and administrated using phpMyAdmin, in order to simplify its management and avoid long instances both from users and from the gateway hub.

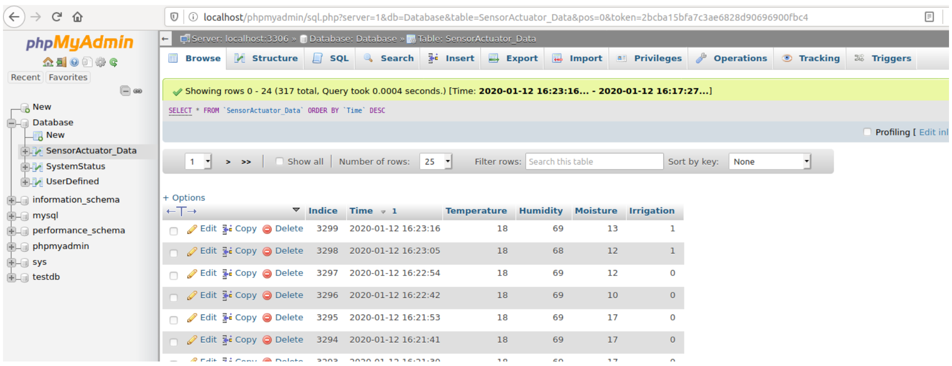


Figure 3.5: phpMyAdmin user interface

It is accessed through the server's IP address. As illustrated in Fig. 3.1, both the RED-Node application and the website use SQL instances to access the database. Any client requires a user name and a password, and different clients can have different privileges. For instance, both RED-Node and the web page can read, update and add data in the MySQL tables.

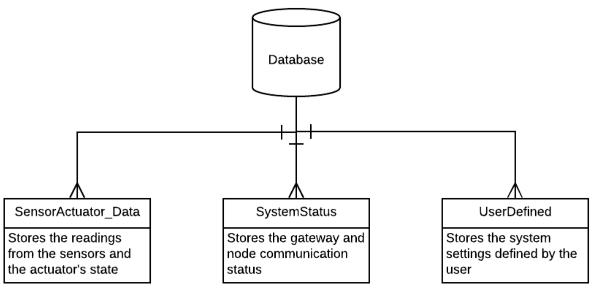


Figure 3.6: Database structure

The database is divided in three different tables: 1) *SensorActuator_Data*, 2) *SystemStatus* and 3) *UserDefined*, (Fig. 3.6). The first one (Table 3.1) is responsible of gathering and storing information from the node, including readings from the sensors and state of the valves (switched on or off). Each reading is made according to the frequency rate configured by the user (*Frequency* field in the *UserDefined* table, 3.3). This structure allows having previous sensor readings so that the user can observe and track trends and plot the data.

Table 3.1: "SensorActuator_Data" table structure

Data name	Data type	Description
Id	Integer	Gives a unique id number to each row. Starting from "1", the id value auto increments each time a row is added.
Time	Time	Automatically records the time when rows with measurements are added.
Temperature	Double	Stores the temperature readings from the sensor [degrees].
Humidity	Double	Stores the humidity readings from the sensor [%].
Moisture	Double	Stores the soil moisture readings from the sensor [%].
Irrigation	Boolean	Stores the irrigation system status (ON = 1, OFF = 0). ON means the water pump is activated, and deactivated otherwise.

The *SystemStatus* (Table 3.2) monitors the state of the gateway and the operation of the sensor node. They serve to enable (or disable) platform options according to the device status and prevent faulty operations, for instance, trying to connect the irrigation valve when the node is disabled. Unlike table 3.1, the *SystemStatus* just stores one row of data which is updated under user request.

Table 3.2: "SystemStatus" table structure

Data name	Data type	Description
Time	Time	Automatically records the time when the row has been updated.
GatewayStatus	Integer	0 means the gateway has been disconnected ; 1 gateway connected but not receiving any answer from the node; and 2 communication between gateway and sensor node is enabled.
NodeStatus	Integer	The status of the node is 0 when it is disconnected from communication or 2 if it is correctly communicating with the gateway.

Finally, the *UserDefined* (table 3.3) stores platform settings requested by the user through the web page.

Table 3.3: "UserDefined" table structure

Data name	Data type	Description
Time	Time	Automatically records the time when the row has been updated.
Manual	Boolean	Irrigation user request. In *Manual Mode* (Manual = 1) the user turns ON/OFF the irrigation system. In Automatic Mode (Manual = 0) irrigation is turned ON and OFF according to the threshold level from the "*Humidity*" entry of this table.
Irrigation	Boolean	When *Manual* = 1, this value stores if the user wants to switch on the irrigation system (Irrigation = 1) or wants to switch it off (Irrigation = 0).
Humidity	Integer	This value stores the soil moisture threshold. When *Manual* = 0, the irrigation system will turn when the moisture readings are below this value, and above this value the irrigation system will turn off.
Frequency	Float	This value fixes the sampling rate to which new measurements are added in Table 3.1
Connection	Boolean	This value stores the connecting sensor option selected by the user.

It is important to notice that in this project has not been intended to create a table to register users, therefore, the website is not prepared to sign up new users.

3.5 Design of web interface in *localhost*

Apache is a free and open-source cross platform used worldwide that offers many compiled modules which can extend the core of a laptop computer to server functionalities, including - authentication schemes, and server-side programming with languages such as Perl, Python and PHP - offering the possibility to host a web domain in the localhost computer, which can be accessed via an IP address.

In this project, both the server domain and the web page are developed from scratch using different languages (Fig. 3.7). The web page can be divided into a *front end* and *back end*. The front end implements all the elements the end user will see and interact with, and is developed using HTML language (*Hyper Text Mark-up Language*), CSS (*Cascade Style Sheets*) and Javascript languages.

HTML serves to define the structure of the web content, while CSS takes care of the web page style and Javascript implements all the functionalities of controls, buttons and dialog boxes, among others. On the other hand, the back end is implemented with Php and makes the connections between the SQL database and the Javascript program of the web page.

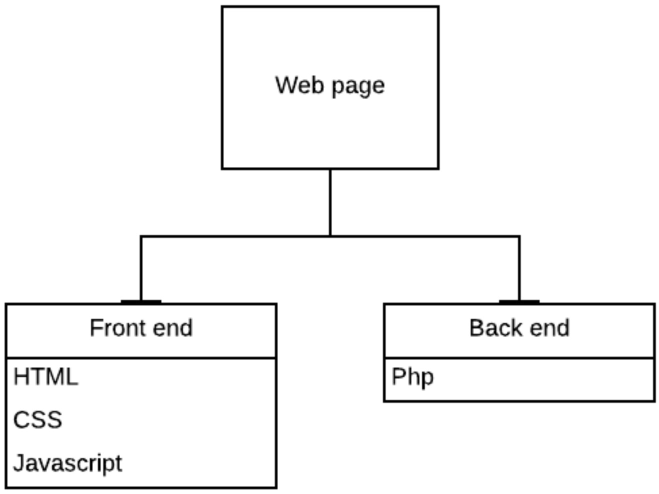


Figure 3.7: Relation between the scripts implementing the web page

3.5.1 Front end

The user interface is the tool by which the user interacts with the machine. In this system, the client should interact with the hardware elements of the platform just through the website. Therefore, it is important to design a robust, easy to use and capable web interface. To achieve this, the front-end has taken into account an strategy with a minimalist and responsive web design.

A responsive web design means making a website that can adapt to the size of different screens. This feature is very important, since nowadays people use computing devices with screens of different sizes and amount of pixels: from laptop computers, smart phones, tablets, among others. Therefore, it is important that the information displayed on the website adapts to each screen, allowing the user to read it without struggling. This characteristic been achieved using the CSS flex property, which sets a flexible length on the implemented items.

The website application consists of a *landing page*, and the main website itself, which is divided into three different pages:

Figure 3.8: Landing page

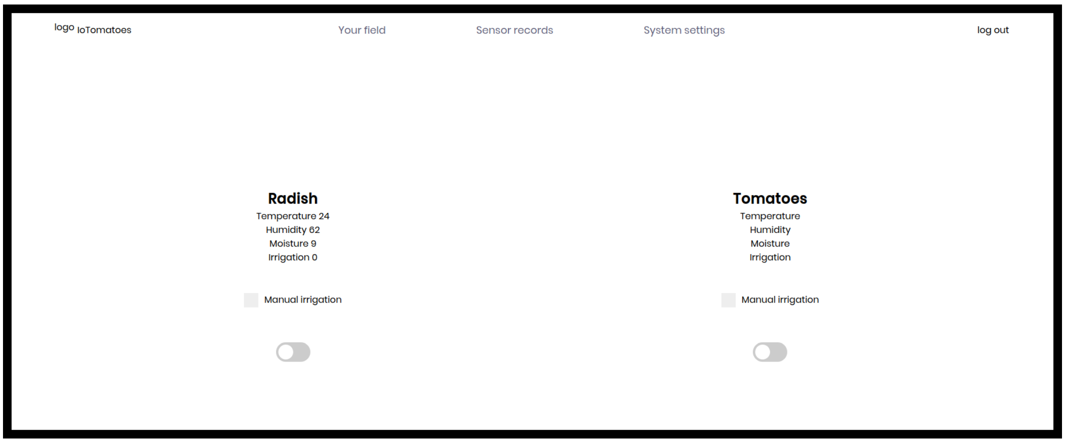


Figure 3.9: "Your field" page

- The landing page is the entry page the user sees when accessing to the website. In there, there is a portal to log in, where the user signs in with its user name and password. To make it more attractive, it has been designed with a background image and a semi-transparent log in portal (Fig. 3.8).

- After loging in, the user enters the *Your field* page containing the main menu where also information in real-time about current crop conditions is displayed (Fig. 3.9). This interface page has different sections. There is one section per node, named after the type of vegetable it is monitoring, where the user can know the temperature, humidity and soil moisture of vegetables and whether the irrigation system is activated (i.e: Radishes are at 24 Celsius degrees, 64% humidity in the atmosphere, 9% moisture in the soil and the irrigation system is off). The user can decide to use the irrigation system in manual or automatic

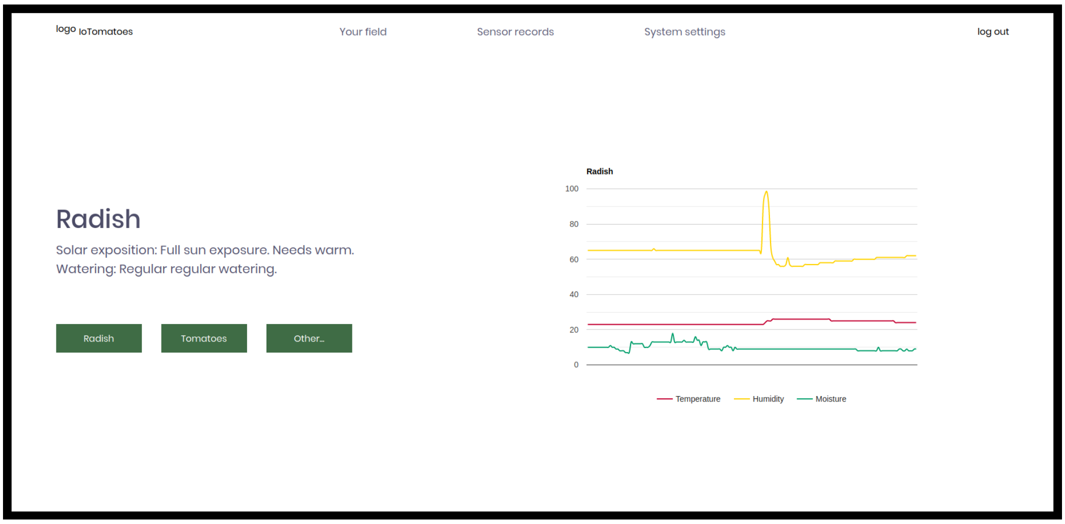


Figure 3.10: "Sensor records" page

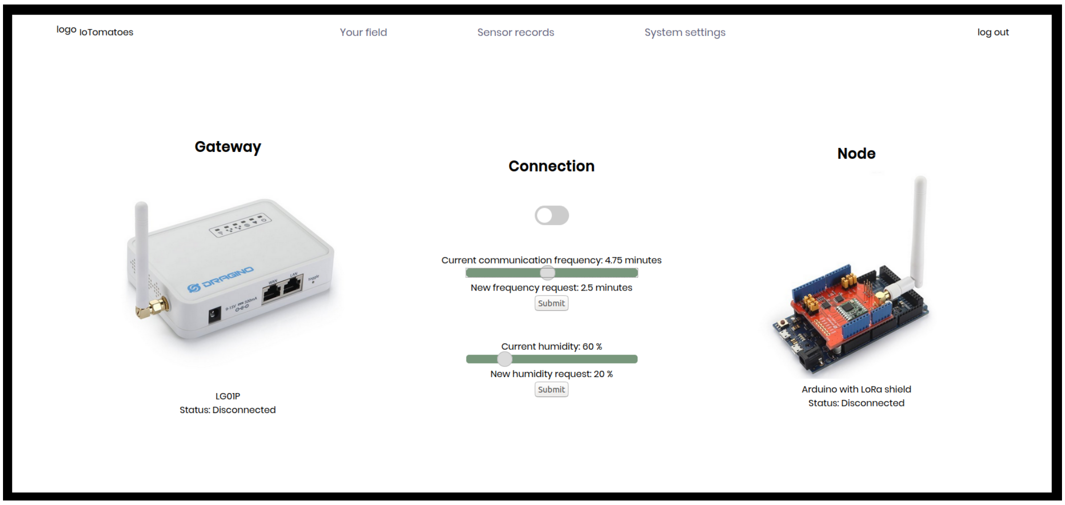


Figure 3.11: "System settings" page

mode by checking the *Manual irrigation* box. If manual mode is selected, an slider will be enabled to turn the irrigation system on or off.

- When the user access the *Sensor records* tab, the previous sensor readings are shown in a plot to track trends (Fig. 3.10).
- Finally, in the *System settings* section the user can change system parameters and check whether the node and gateway are operating correctly (Fig. 3.11). By means of the *Connection* slider, the user can cut communication with the sensor node but communication with the gateway is maintained .The user can also configure the sampling rate of the sensor, balancing the rate and the energy required by the system, and the threshold when the irrigation system will be turned on/off depending on the soil moisture.

3.5.2 Back end

Php is a general purpose programming tool, used in websites as a server side scripting language.

In this project, the website back end is basically in charge of accessing the database to read and write information, under the request of the website's front end (Fig. 3.12). When the web page wants to display sensor data in the user interface or send input information from the user to the node, an ajax request to the back end is sent through the JavaScript (front end). Ajax is a set of web development techniques that allow the web page to send and retrieve data from a server asynchronously, without interfering with the display and behavior of the existing page [101]. When the request is received by the back end, a MySQL connections with the server database is established. Depending on the request the actions taken vary from one instance to another: reading data, updating a row, or adding a new row in the sensor table are just a few.

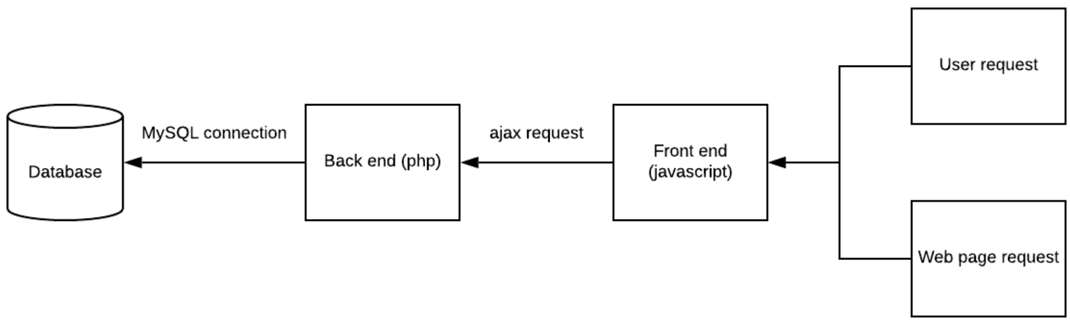


Figure 3.12: Back end flow chart

3.6 Node-RED layer

Node-RED has been deployed in the server's localhost as a layer between MySQL database and the gateway to simplify the transfer of SQL packets. While not being an indispensable element, it is a helpful debugging tool for monitoring data frames between the gateway and the database and detecting whether any communication issues occur.

In order to transmit the data between the gateway and Node-RED, both MQTT and HTTP protocols have been used. Figure 3.13 illustrates how the data coming through MQTT and HTTP protocols is managed in a crop section of the Node-RED layer.

- Sensor data flow is handled with an MQTT publishing node in the diagram on the top of Fig. 3.13 . The first node is a MQTT node, which connects to the broker (the server and port) and subscribes to messages from an specific topic. In other words, this node is listening for other devices to publish on a topic, using the specific server and port. In this example, the gateway publishes the sensors' results and actuator status on this topic. Once the data arrives, it goes to the second node "json", where the data is structured in order to be more comprehensive and easy to manage. Finally, the last element opens a MySQL connection and the data is sent to the server
- Initially, the communication between the gateway and Node-RED was planned to be through MQTT publishing and subscription, but the gateway did not behave as expected when connected to several topics. To fix this issue, HTTP requests have been implemented alternatively to MQTT in some cases, such as the sending of data from the server to the gateway (see bottom of Fig. 3.13). In this case, the *GetIrrigation* node creates an HTTP end-point for managing web services. When the gateway connects to the URL and publishes in for this node, Node-RED connects to the server's database, reads the requested information from the required table and send it to the gateway.

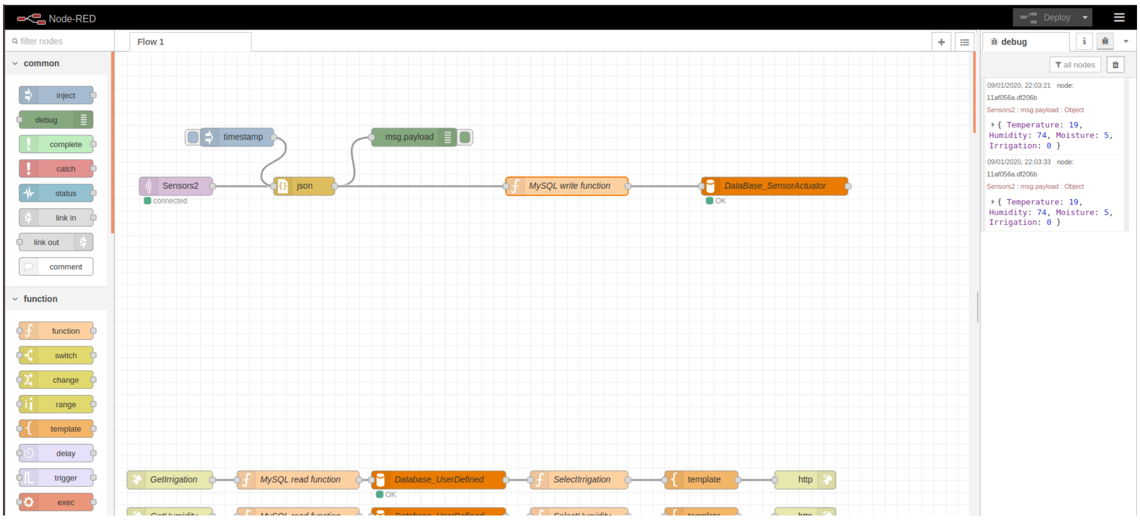


Figure 3.13: Platform's Node-RED layer (crop section)

3.7 Gateway application program

The selected hardware to operate as a gateway has been the Dragino LG01-S, and its mission is to bridge between the sensor node and the server application. The Dragino gateway is programmed in C language through Arduino IDE.. .

Figure 3.14 shows the general program flow corresponding to the gateway program. The hub initiates its LoRa interface, after powering up, and seeks for available

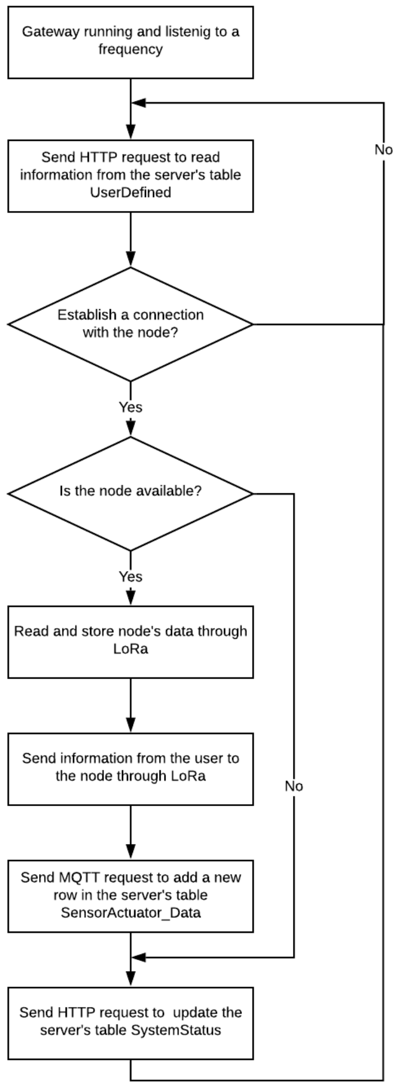


Figure 3.14: Simplified Dragino work flow

nodes at 433MHzin, the radio-frequency spectrum toed . All this is managed by the GitHub library of the LoRa transceiver RH_RF95 .

When the set up is completed, the gateway requests user commands through the HTTP socket stored in the server. This can be either activating the the irrigation system, configuring a new sampling frequency, modifying threshold values or disconnecting the sensor node from the gateway (see table 3.3).

When the user wants to connect with the sensor node and connection is established between the node and the gateway, it requests for availability of the node. If the node is not available, the gateway informs through HTTP publishing about the node's unavailable status. On the contrary, if the node is available, it will read and store the sensor's information coming from the node through LoRa. Afterwards, the gateway will send an answer to the node with user defined information related to the sensors and actuators, and will send the node's information to the server through MQTT. Finally, it will update the server with the node's available status.

3.8 Sensor/Actuator node

The installation of the sensor/actuator node comprises the physical connection of the different elements the hardware program implemented in the Arduino Board

3.8.1 Node hardware system

The first trial consists on a breadboard to verify electrical operation of the sensors, the actuators and the LoRa shield (Fig. 3.15).

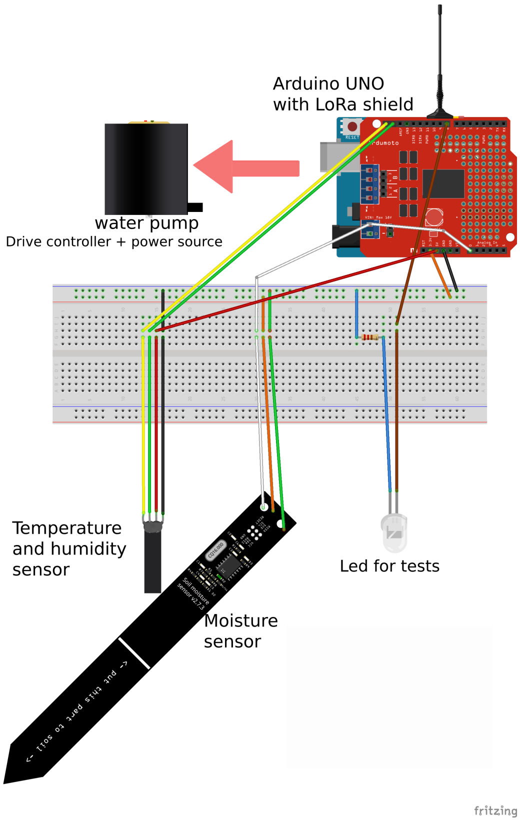


Figure 3.15: Node's electronic circuit sketch

The micro-controller is currently powered via a USB from the Laptop PC. However, on future developments this stage has to be implemented with an alternative energy system.

The SHT-30 Temperature/Humidity digital sensor is powered to 5V (red and black wires) which are supplied by the Arduino board, whereas its I2C interface is connected to SDA and SCL, respectively. The Soilwatch 10 moisture sensor uses 3 wires. The power terminals connect to 3.3V and the signal output (White) is connected to the analog pin A0. the green wire has been connected to ground, the brown wire to 3.3V output pin, and the white wire to the A0 (analog) Arduino input pin. The water pump has two wires: a brown wire for powering the actuator (4.5~12V DC) and a blue one for ground.Curently, the blue wire has been connected to ground and the orange wire to the 8th Arduino digital output pin, therefore, it will be easily switched on/off via software.However, the pump was not working using the pins, this is because the output from the pins is 5V and 0.04A, but the pump requires 0.1A when working at 4.5V. Therefore, it will be required to feed the pump with an external energy source (for instance 12V) and a current driver such as L293D shield or L298N drive controllers. Until its installation, a let for tests is going to be installed where the pump was supposed to be.

The LoRa shield is attached over the Arduino Board, and it is connected through the pins (Fig. 3.16).

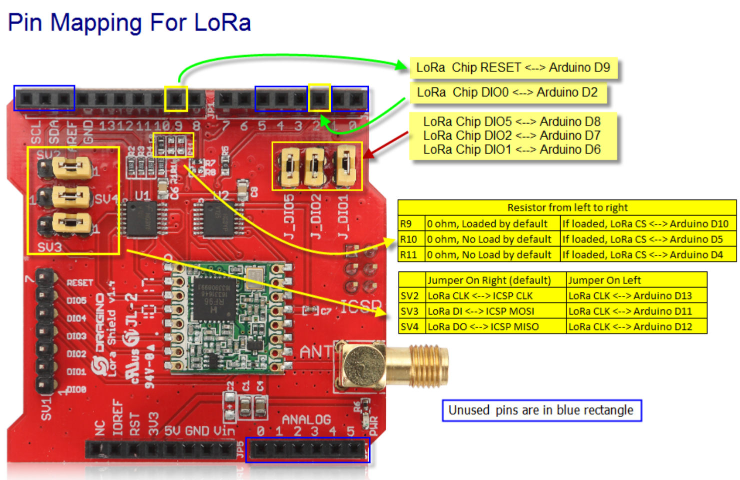


Figure 3.16: Pin Mapping For Lora [102]

3.8.2 Node application program

The hardware program is based on an been Arduino UNO platforms and correspond to the flow diagram of Fig. 3.17. Its main function is, on one hand,to read t sensor variables and control the actuator, and manage data transfer and sensor configuration according to received information from the user. This code is programmed in Processing language through the Arduino IDE.

As seen, once the micro controller is switched on, the Arduino opens aits LoRa connection, and initialize sensors and actuator. If no problems are found, the Arduino wlistens to 868MHz in the radio-frequency spectrum. This is managed by the library RH_RF95 “client node”.

When commands are received from the gateway, the node reads the sensor variables and send the data through LoRa to the gateway. This is done thanks to the Adafruit’s_SHT31 tempereture/humidity sensor library. This operation takes into account the configured sampling ratewhich can be configured through the web page.

Then, the node awaits for the gateway to recieve and the following information from the user:

- User’s selected communication frequency rate.
- User’s selected manual or automatic irrigation system.
- If manual, the irrigation status (ON or OFF).
- If automatic, the moisture threshold value to turn the irrigation ON/OFF.

If a reply occurs, internal node parameters are updated in order to fit to user’s requirements, including the irrigation system, which will be turned on or off

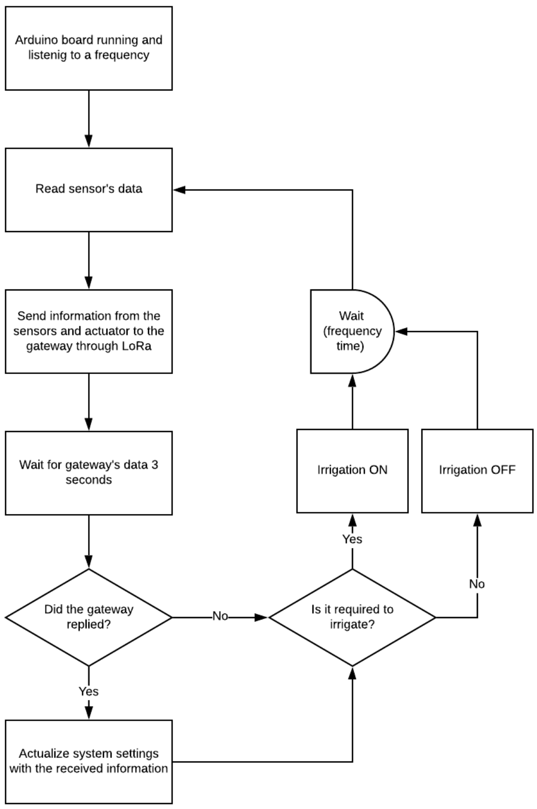


Figure 3.17: Simplified Arduino work flow

3.9 Discussion

In this section, the hardware programs of the different elements of the system as well as the node electronic system have been covered. At the point of writing this document, the system is capable of exchanging different information between the field node and the remote user.

In section 3.3, the server has obtained its IP and subdomain, allowing the rest of elements to communicate with it through the Internet. In section 3.4, a MySQL database and tables have been created in order to manage and store all the system's data, becoming the main part of the server. In section 3.5, the website has been designed to allocate the system's user interface, allowing the user to communicate with the platform. In section 3.6, a layer in between the hardware and the server has been developed in order to simplify data acquisition. In section 3.7, the gateway has been programmed in order to manage the information between the node and the server. Finally, in section 3.8, the node has been programmed to read the information from the sensors, manage the actuator, and change its settings depending on user's inputs.

On the other side, due to time limitations and the scope of the project, some important elements have not been covered. First of all, the irrigation system will require the implementation of additional elements, as explained in section 1.8.1. Secondly, the node's power management system has not been implemented. Even if the required components have been selected in chapter 2, due to time limitations they have not been wired to the node's electronic circuit. On the other hand, the application security management system is not strong enough. In order to deploy the system in a real scenario, further development in the security system is required.

4 Test, Set-up and Operating Results

4.1 Overview

To validate the platform implementation, this chapter shows its capacity to configure the sensor, obtain sensor data and activate the irrigation system after proper configuration.

First of all, this chapter will start with the required steps to set-up the system (see 4.2), and it will include different tests in order to confirm it has been properly configured (included in 4.3 and 4.4).

4.2 Platform and system set-up

The first step to set up the application is to switch on and connect the elements to the Internet.

1. Power on the gateway plugging it to an energy source. Connect it to the Internet via WiFi, has explained in Dragino manual (see [66]) or using an Ethernet cable, connected to Dragino WAN port, as illustrated in image 4.1. The lights will start blinking.

2. Turn on the PC where the server is allocated. Connect the PC to the Internet via WiFi or an Ethernet cable.

3. First of all, ensure all the elements of the node are correctly wired, it can be compared to chapter 3.8.1. Power on the micro-controller via USB to any energy source with USB port, then Arduino's board led should turn on. As illustrated in Figure 4.2, the Arduino micro-controller is connected to a PC via USB port and powering the rest of elements in the node.

In order for the website to be visible on the Internet, outside the router's internal network, the PC where the server is installed should have installed the dynamic DNS services offered by no-IP [100]. In addition, the router must be configured in order to forward port 80. The following steps will explain how to forward a port using a Virgin router.

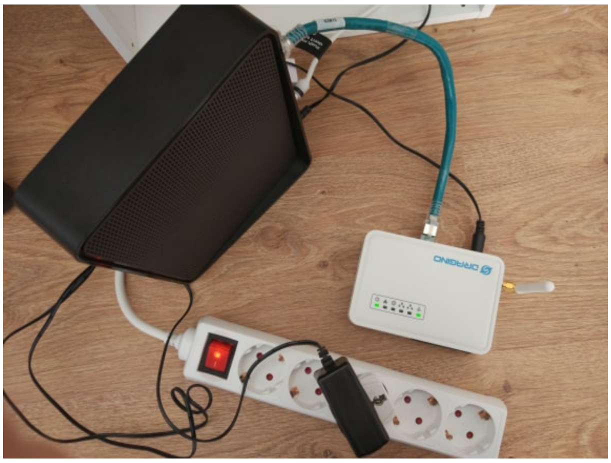


Figure 4.1: Gateway connections

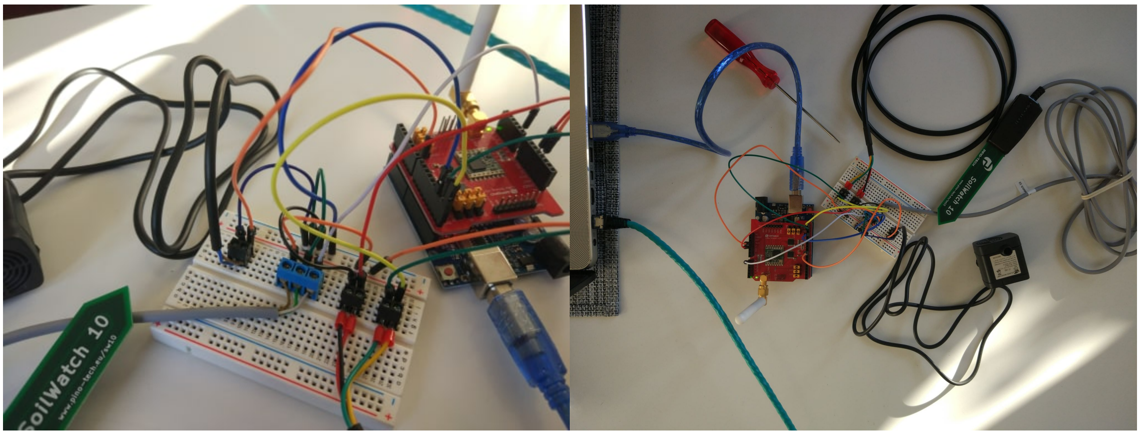


Figure 4.2: Node connections

1. Enter to the router configuration page using its IP address and password. For example, the IP address for this Virgin router is 192.168.0.1.

2. Go through the WiFi settings until finding Port forwarding settings. Using Virgin, the steps are: Advanced settings -> Security -> Port forwarding.

3. Create a new rule to forward port 80, as illustrated in figure 4.3 .

4. Enable the rule and save changes if required.

5. Finally, by typing “http://iotsaba.ddns.net/landingPage.html” the user will be able to access the web page from any device anywhere.

If the user has followed the previous steps, the obtained final system should be similar to the one illustrated in figure 4.4. However, the element’s IP addresses may vary.

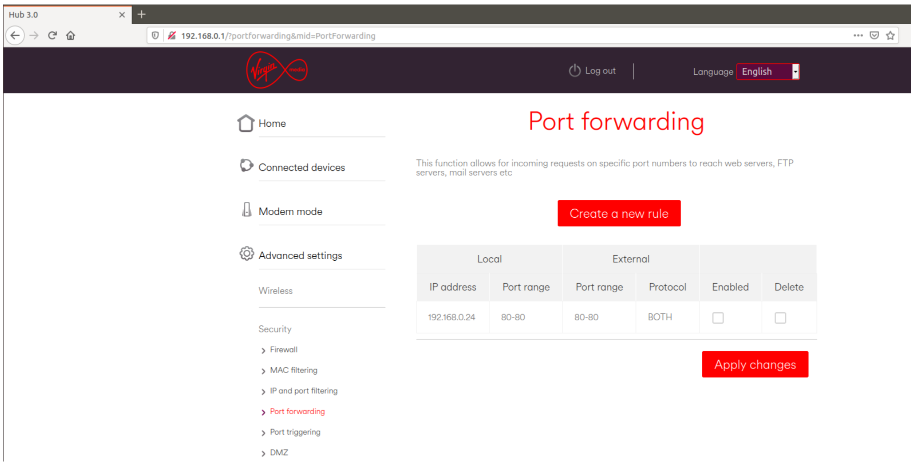


Figure 4.3: Port forwarding using a Virgin router

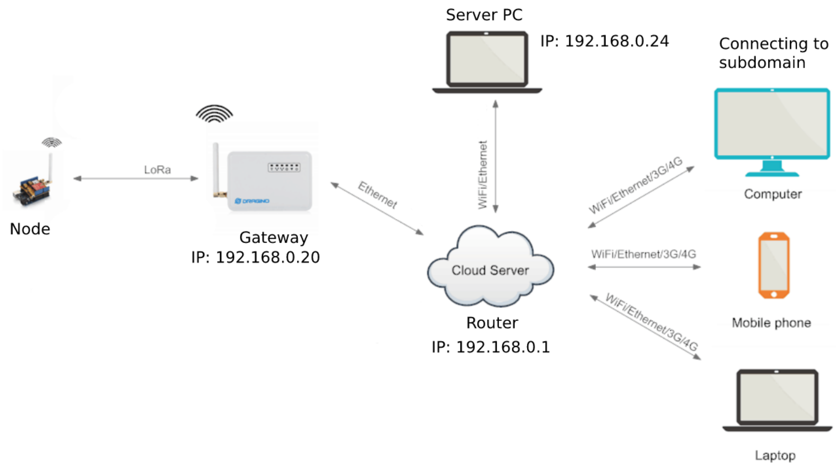


Figure 4.4: System network sketch

4.3 1st. Demo: Sensor node to *localhost*

In order to ensure the sensors are properly connected to the system and the platform is receiving their data, both sensors will be tested. Before proceeding with the tests, it is recommended to open the website and ensure the communication frequency in sysem settings is below 1 minute, so the user does not need to wait for values too long. In order to see the results from the tests, it is recommended to open the page "Sensor Records", where the results should be easy to see in the graph.

- First of all we will start testing the moisture sensor SoilWatch 10. In order to test the moisture sensor, the user will need a glass of water to submerge entirely the sensor, as illustrated in figure 4.5. The moisture values in website graph should go up to 100% moisture, as shown in figure 4.6. Once the sensor is outside the water, the values should decrease again.

Figure 4.5: Moisture sensor test

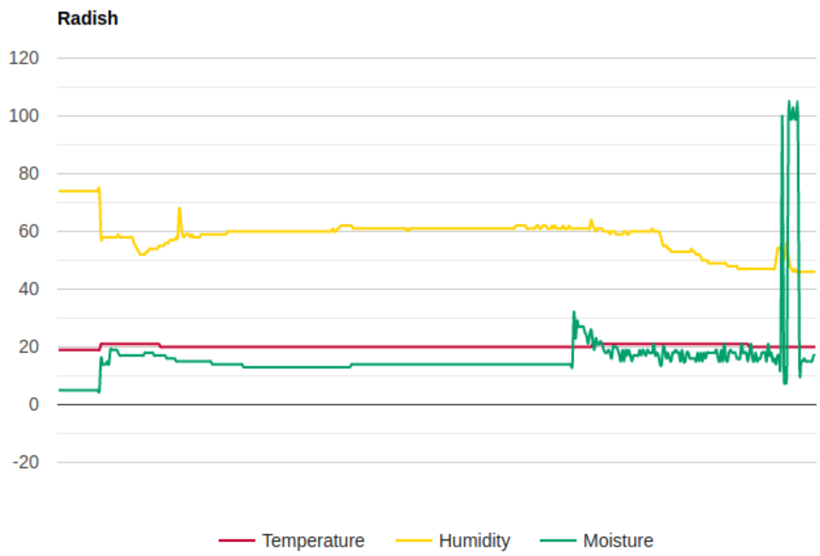


Figure 4.6: Moisture sensor results

- In order to test the temperature and humidity sensor Adafruit SHT-30, the user just needs to grab it between her/his hands for a minute as illustrated in figure 4.7. The humidity values in website graph should go up to 100% humidity, and the temperature should increase a little as shown in figure 4.8. Once the user stops grabbing the sensor, the humidity should decrease again, but the temperature will decrease slower.

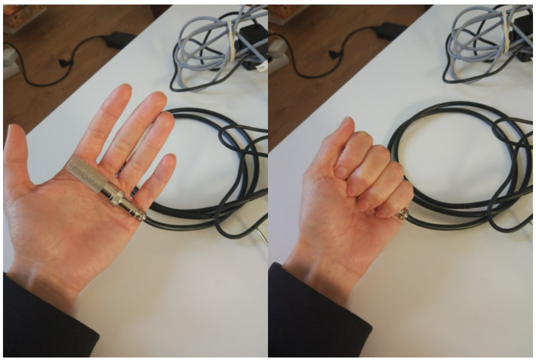


Figure 4.7: Temperature and humidity test

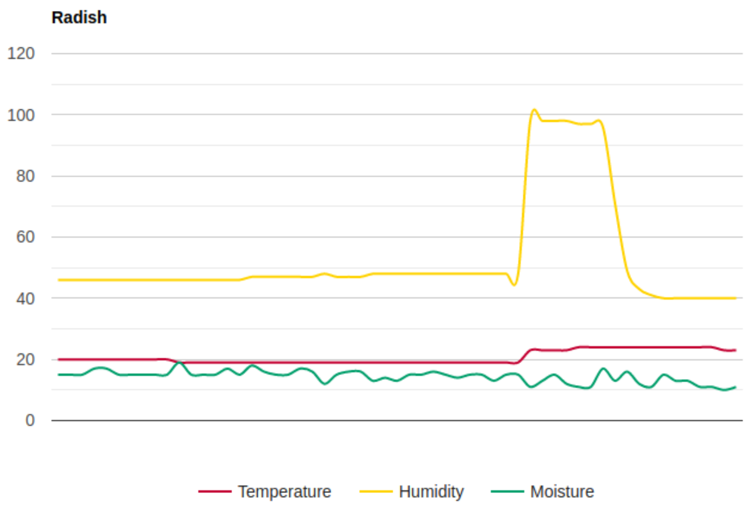


Figure 4.8: Temperature and humidity sensor results

4.4 2nd. Demo: *localhost* to sensor node

The user can check the node is responding to website inputs in different ways. In this example, we are going to manually control the irrigation system through the website page "Your field". One led is going to be used for this test, it will turn on when the irrigation system would be pumping water or off otherwise.

To use the irrigation system manually, the user should check the checkbox Manual Irrigation. At this moment, the slider below the checkbox should be enabled and the user will be able to switch the system on and off (Fig. 4.9)

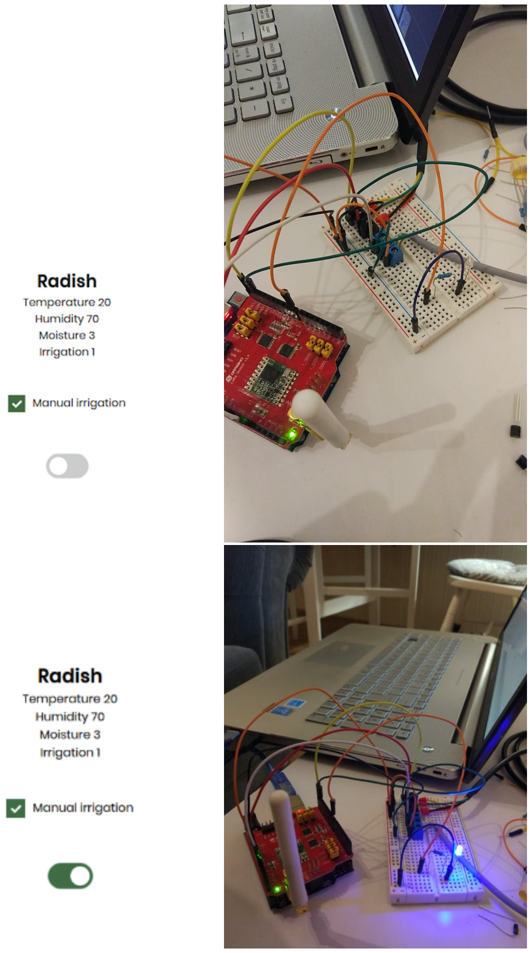


Figure 4.9: Irrigation system test

5 Conclusions and Future Work

5.1 Conclusions

In this work, a first prototype of a platform for E-agriculture has been successfully developed. The platform consists of a sensor node network connected to a server via Internet, and a web based server user interface.

The sensor node with the micro-controller included two sensor devices and one irrigation actuator. A basic website has been developed and hosted in a computer-based server, allowing the user to monitor the sensor readings and the actuator status, and to manage some of the system settings remotely, such as: the communication frequency rate between the node and the gateway, and the moisture threshold to start irrigating the field, among others. The hardware of the sensor node and gateway, and the software corresponding to the remote website and the database have been developed.

Chapter 2 discussed the theoretical and principles that guide current design of IoT applications worldwide. Trending technologies, software and protocols used in IoT projects today were reviewed. This lead to the selection of hardware components that made up the IoT platform of this study. Chapter 3, showed the design of the current E-agriculture platform and the strategy followed for programming the hardware of the architecture (sensor node and gateway) and the web software, in order to create a basic network architecture and transfer sensor data to the server.

A brief manual has been written to help the end user to set up this application in case he wants to test this work, as well as a troubleshooting section to verify communications from the localhost to the node and vice-versa.

5.2 Future Work

New tasks to be developed in a foreseeable future are presented in this section. The first goal this platform should aim is to deploy the current system in a real scenario of an agriculture application. In what follows, the necessary tasks to be carried out toward this goal in a near future are referred to as (ST, meaning Short-term), whereas long-term tasks (LT) refer to ideas to boost and scale the platform functionality once it is ready and operative in the field.

- **Powering system (ST)**. Adding a powering system based in the use of solar cell batteries is one necessary step ahead to make sensor node to operate autonomously in the field.
- **Irrigation valves (ST).** The necessary electronics to operate valve actuators have to be added. This rely on the purchase (or design) of a board which, once placed between the node and the valve, can drive the actuator input.
- **A more permanent node circuit (ST)**. At current time, sensors and actuators have been connected provisionally with the help of a breadboard. To gain in robustness, elements must be embedded into a single board, as far as possible, or to wire external elements properly to the Arduino board. Developing a custom Arduino board is possible, but requires recording the Arduino's bootloader in the Atmel's micro-controller (ATMEGA328P or 2560), which is only possible in UNO, MICRO, NANO and MEGA versions. The other alternative, is to change hardware platform, such as Opengarden or Waspmote Agriculture from Libelium. .
- **Weatherproof (ST)**. To protect the platform from extreme weather conditions, all electronics (batteries, PCBs and micro-controller) must be protected from the outside external agents. Therefore, a case must be designed in order to enclose and isolate all those elements.
- **Security and privacy (LT)**. Right now the platform lacks of any security or protection mechanism against attacks on people's privacy, because this matter was outside the scope of this project, but it is a major requirement of the current IoT system. Some possibilities are PHP extensions such as Mcrypt which provide basic level of data encryption when executing SQL queries, or to install Transport Layer Security and (TSL) to provide transport certificates with tools such as *Let's Encrypt*.
- **Real scenario tests (ST/LT - Beginning of spring)**. Once the platforms is ready enough to be implemented outside, it will be further tested in a real scenario. It s set up is planned for early spring, in a crop of Ballobar (Aragón, Spain). It is intended to tests its performance by planting two rows of the same vegetables, one row will be managed by the IoT platform and the second row will be irrigated using drip irrigation with timer (current installation).
- **Scalability (LT)**. When reliability is ensured in the IoT platform, the next logical step is to create a mesh network by adding several nodes and gateways. According to the nodes and gateways used for this platform, it seems worthwhile exploring The Things Network approach. The Dragino manual comes with recommendations and advises to configure the LG01 to operate with this infrastructure extending worldwide.
- **Server host (LT)**. Hosting a website in a laptop PC computer serve to learn the basics but real IoT applications are normally developed in data-centers or cloud platforms. Again, The Things Network provide tools in the cloud

to develop web interfaces for monitoring your sensors, although exploring the possibility to hold the website, for instance, in a Raspberry Pi could be a previous step to check your smart agriculture platform before moving it to the cloud.

- **Include Edge analytics (LT)**. This is normally a final step when the platform has demonstrated robust and secure. Adding more add-ons to deeper process and analyze the data obtained from the fields. Using artificial intelligence to achieving real-time decision making based on previous data would be a powerful tool in order improve the efficiency of the system.

Acknowledgments

I would like to acknowledge the director of this project, Jose Antonio Soria Perez, and the Distributed Sensor Netowrks Group (DISEN), who provided knowledge, time, and helpful advice through all the process.

I would like to express my deepest appreciation to Joan Correns for his encouragement and invaluable help during all the project.

I wish to thank my parents for all their support and patience during my studies.

Last of all, I would like to thank all my colleges who contributed to this project with their ideas and expertise.

Bibliography

[1] Lewis A. How iot is minimising the damage of climate change on the agriculture industry, 05 2019. Viewed on 12 Nov 2019, <https://www.iot-now.com/2019/05/16/95838-iot-minimising-damage-climate-change-agriculture-industry/>.

[2] Food and Agriculture Organization of the United Nations. The state of food security and nutrition in the world, 2019. Viewed on 25 Oct 2019, <http://www.fao.org/state-of-food-security-nutrition/en/>.

[3] Carnegie Mellon University. The 'only' coke machine on the internet. https://www.cs.cmu.edu/∼coke/history_long.txt.

[4] Rouse M. internet of things (iot), 2019. Viewed on 25 Oct 2019, <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>.

[5] ITU. Internet of things global standards initiative. Viewed on 18 Nov 2019, <https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx>.

[6] Brown E. 21 open source projects for iot, 11 2019. Viewed on 25 Nov 2019, <https://www.linux.com/NEWS/21-OPEN-SOURCE-PROJECTS-IOT/>.

[7] FAO. Smart farming is key for the future of agriculture, 2017. Viewed on 10 Oct 2019, <http://www.fao.org/family-farming/detail/en/c/897026/>.

[8] Alonso T. Agricultura 4.0: la revolución iot en el campo español, 08 2018. Viewed on 02 Nov 2019, <https://hablemosdeempresas.com/grandes-empresas/iot-en-agricultura/>.

[9] Fernández J.M. Agricultura inteligente (parte 4): Barreras que ralentizan la expansión del iot en el sector, 01 2018. Viewed on 25 Oct 2019, <https://www.fertiberia.com/es/blog/2018/enero/agricultura-inteligente-4-barreras-iot/>.

[10] Palermo F. Internet of things done wrong stifles innovation, 07 2014. Viewed on 15 Nov 2019, <http://www.informationweek.com/strategic-cio/executive-insights-and-innovation/internet-of-things-done-wrong-stifles-innovation/a/d-id/1279157>.

[11] Gartner. Gartner says 5.8 billion enterprise and automotive iot endpoints will be in use in 2020, 08 2019. Viewed on 15 Oct 2019, <https://www.gartner.com/en/newsroom/press-releases>.

[12] Yasuhisa O., Abhijit M., and Naoto M. *MOS Devices for Low-Voltage and Low-Energy Applications*. John Wiley & Sons, 2017.

[13] Vongsingthong S. and Smanchat S. Internet of things: A review of applications & technologies. *Suranaree Journal of Science and Technology*, 2014.

[14] Business Insider. The enterprise internet of things market, 02 2015. Viewed on 02 Nov 2019, <https://www.businessinsider.com/the-enterprise-internet-of-things-market-2014-12?r=US&IR=T>.

[15] Trak.in. How iot's are changing the fundamentals of retailing, 05 2017. Viewed on 21 Nov 2019, <https://trak.in/tags/business/2016/08/30/internet-of-things-iot-changing-fundamentals-of-retailing/>.

[16] Won Min K., Moon Yeon S., and Park Hyuk J. An enhanced security framework for home appliances in smart home. *Human-centric Computing and Information Sciences*, 2017.

[17] Business Insider. How iot & smart home automation will change the way we live. Viewed on 02 Nov 2019, <https://www.businessinsider.com/iot-smart-home-automation?r=US&IR=T>.

[18] Greengard S. *The Internet of Things*. Cambridge, MA: MIT Press, 2015.

[19] Prospero M. Best smart home hubs of 2018. Tom's Guide, 09 2018. Viewed on 20 Nov 2019, <https://www.tomsguide.com/us/best-smart-home-hubs,review-3200.html>.

[20] Chinchilla C. What smart home iot platform should you use? Hacker Noon, 11 2018. Viewed on 21 Oct 2019, <https://hackernoon.com/what-smart-home-iot-platform-should-you-use-2554ea213df1>.

[21] Baker J. 6 open source home automation tools. opensource.com, 12 2017. Viewed on 16 Nov 2019, <https://opensource.com/tools/home-automation>.

[22] Aburukba R., Al-Ali A. R., Kandil N., and AbuDamis D. *Configurable ZigBee-based control system for people with multiple disabilities in smart homes*. IEEE, 2016.

[23] Mulvenna M. Views of caregivers on the ethics of assistive technology used for home surveillance of people living with dementia. *Neuroethics*, 2017.

[24] Kelley A. 5 home security upgrades you can make now. Viewed on 16 Nov 2019, <https://www.inverse.com/article/55773-5-home-security-upgrades-you-can-make-now>.

[25] Kricka L. J. History of disruptions in laboratory medicine: what have we learned from predictions? *Clinical Chemistry and Laboratory Medicine*, 06 2018.

[26] Ersue M., Sehgal A., Romascanu D., and Schönwälder J. Management of networks with constrained devices: Use cases. *ETF Internet Draft*, 07 2014.

[27] Grell M., Dincer C., Le T., and Lauri B. Autocatalytic metallization of fabrics using si ink, for biosensors, batteries and energy harvesting. *Advanced Functional Materials*, 2018.

[28] Stern C. Goldman sachs says a digital healthcare revolution is coming and it could save america 300 billion dolar, 06 2015. Viewed on 25 Nov 2019, <https://www.businessinsider.com/goldman-digital-healthcare-is-coming-2015-6?r=US&IR=T>.

[29] iot.do. Withings wireless blood pressure monitor. Viewed on 07 Nov 2019, <https://iot.do/devices/withings-wireless-blood-pressure-monitor>.

[30] Xiao-Feng X. Key applications of the smart iot to transform transportation, 2018. Viewed on 20 Nov 2019, <http://www.wiomax.com/what-can-the-smart-iot-transform-transportation-and-smart-cities/>.

[31] Severi S. M2m technologies: Enablers for a pervasive internet of things. *The European Conference on Networks and Communications*, 06 2014.

[32] Zigurat Global Institute of Technology. Smart city series: the barcelona experience, 02 2019. Viewed on 20 Nov 2019, <https://www.e-zigurat.com/blog/en/smart-city-barcelona-experience/>.

[33] Berg J. Sharing cities: A case for truly smart and sustainable cities. *MIT Press*, 11 2015.

[34] Musa S. Smart city roadmap, 01 2016. Viewed on 15 Nov 2019, <https://www.academia.edu/21181336/Smart_CityRoad_map>.

[35] Yang C., Shen W., and Want X. The internet of things in manufacturing: Key issues and potential applications. *IEEE Systems, Man, and Cybernetics Magazine*, 01 2018.

[36] Vashi S., Ram J., Modi J., Verma S., and Prakash C. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, pages 1645–1660, 2013.

[37] SmartFactory. The aim of smartfactorykl is to pave the way for the intelligent factory of tomorrow. Viewed on 04 Jan 2020, <https://smartfactory.de/en/industrie-4-0-plant/>.

[38] Zhou W., Zhang Y., and Liu P. The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. *IEEE Internet of Things Journal*, page 1606 1616, 06 2018.

[39] Gartner. Iot technology disruptions: A gartner trend insight report. *Gartner*, 06 2017.

[40] Zhang Z., Cheng Yi Cho M., and Wang C. Iot security: Ongoing challenges and research opportunities. *IEEE 7th International Conference on Service-Oriented Computing and Applications*, page 230 234, 2014.

[41] Khan M. and Salah K. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, page 395 411, 05 2018.

[42] Young A. and Yund M. Cryptovirology: extortion-based security threats and countermeasures. *IEEE Symposium on Security and Privacy*, page 129 140, 1996.

[43] Goodin D. Yo are infected, if you want to see your data again, pay us 300 dolar in bitcoins, 10 2013. Viewed on 04 Nov 2019, <https://arstechnica.com/information-technology/2013/10/youre-infected-if-you-want-to-see-your-data-again-pay-us-300-in-bitcoins/>.

[44] Naraine R. Cryptolocker infections on the rise; us-cert issues warning, 11 2019. Viewed on 25 Nov 2019, <https://www.securityweek.com/cryptolocker-infections-rise-us-cert-issues-warning>.

[45] CISA. Security tip (st04-015), 11 2009. Viewed on 10 Dec 2019 <https://www.us-cert.gov/ncas/tips/ST04-015>.

[46] Farley M. Why the consumer internet of things is stalling, 10 2013. Viewed on 21 Nov 2019, <https://www.forbes.com/sites/ciocentral/2016/09/13/why-the-consumer-internet-of-things-is-stalling/>.

[47] Lal Das P. Internet of things : the new government to business platform - a review of opportunities, practices, and challenges (english), 11 2017. Viewed on 19 Nov 2019, <http://documents.worldbank.org/curated/en/610081509689089303/Internet-of-things-the-new-government-to-business-platform-a-review-of-opportunities-practices-and-challenges>.

[48] Wieland K. Iot experts fret over fragmentation, 02 2016. Viewed on 20 Nov 2019, <https://www.mobileworldlive.com/mwc16-articles/iot-experts-fret-over-fragmentation/>.

[49] Qualcomm. Fragmentation is the enemy of the internet of things, 02 2016. Viewed on 20 Nov 2019, <https://www.qualcomm.com/news/onq/2016/02/19/fragmentation-enemy-internet-things>.

[50] The World Bank. Agriculture and food, 10 2019. Viewed on 23 Nov 2019, <https://www.worldbank.org/en/topic/agriculture/overview>.

[51] FAO. Executive summary. Viewed on 11 Oct 2019, <http://www.fao.org/3/y3557e/y3557e03.htm>.

[52] World Food Programme. 2019 - the state of food security and nutrition in the world (sofi): Safeguarding against economic slowdowns and downturns, 07 2019. Viewed on 05 Nov 2019, <https://www.wfp.org/publications/2019-state-food-security-and-nutrition-world-sofi-safeguarding-against-economic>.

[53] Max R. and Ritchie H. Hunger and undernourishment. *Our World in Data*, 2020. https://ourworldindata.org/hunger-and-undernourishment.

[54] Townsend R. Ending poverty and hunger by 2030 : an agenda for the global food system (english). *Washington, D.C. : World Bank Group*, 2015. http://documents.worldbank.org/curated/en/700061468334490682/Ending-poverty-and-hunger-by-2030-an-agenda-for-the-global-food-system.

[55] Allied Market Research. Global iot in agriculture market expected to garner 48.71 bn by 2025 at 14.7 Viewed on 07 Nov 2019, <https://www.globenewswire.com/news-release/2019/07/22/1885778/0/en/Global-IOT-in-Agriculture-Market-Expected-to-Garner-48-71-Bn-by-2025-at-14-7-CAGR-Says-AMR.html?fbclid=IwAR30T4zp0WEX2GLbDXeiANxDdYHjOqIVhyz4vgIYEk2ZHL7AmowLcx

[56] FAO. E-agriculture. Viewed on 13 Oct 2019, <http://www.fao.org/e-agriculture/>.

[57] FAO. Climate-smart agriculture. Viewed on 21 Oct 2019, <http://www.fao.org/climate-smart-agriculture/overview/en/>.

[58] Libelium. Viewed on 01 Dec 2019, <http://www.libelium.com/>.

[59] Growlink. Viewed on 15 Nov 2019, <https://growlink.com/>.

[60] Libelium. Improving banana crops production and agricultural sustainability in colombia using sensor networks, 10 2016. Viewed on 01 Dec 2019, <http://www.libelium.com/improving-banana-crops-production-and-agricultural-sustainability-in-colombia-using-sensor-networks>.

[61] Tomás J.P. Vodafone using precision agriculture to help new zealand farmers cut costs, 08 2017. Viewed on 28 Nov 2019, <https://enterpriseiotinsights.com/20170208/channels/fundamentals/vodafone-precision-agriculture-tag23-tag99>.

[62] Ligero R. 'connected cows' are destined to become a reality in farms, 03 2018. Viewed on 01 Dec 2019, <https://accent-systems.com/blog/connected-cows-reality-farms/>.

[63] Fox C. Dairymaster moomonitor+ first in class for monitoring cow behavior, 05 2019. Viewed on 02 Nov 2019, <https://www.independent.ie/business/farming/agri-business/companies/dairymaster-moomonitor-first-in-class-for-monitoring-cow-behavior-38095157.html>.

[64] Avsystem. What is iot architecture?, 07 2019. Viewed on 28 Nov 2019, <https://www.avsystem.com/blog/what-is-iot-architecture/>.

[65] Raspberry Pi. Products. Viewed on 03 Jan 2020, <https://www.raspberrypi.org/products/>.

[66] Dragino. *LG01 LoRa Gateway User Manual*. Dragino, 1.4 edition, 04 2018.

[67] TechTerms. Protocol, 03 2019. Viewed on 28 Nov 2019, <https://techterms.com/definition/protocol>.

[68] Irons-Mclean R. Iot and security standards and best practices, 01 2019. Viewed on 12 Nov 2019, <http://www.ciscopress.com/articles/article.asp?p=2923211&seqNum=6>.

[69] Mehl B. 6 communication protocols used by iot. Kisiblog, 05 2018. Viewed on 20 Dec 2019, <https://www.getkisi.com/blog/internet-of-things-communication-protocols>.

[70] Wikipedia. Osi model. Viewed on 03 Jan 2020, <https://en.wikipedia.org/wiki/OSI_model>.

[71] Harwood T. Iot standards and protocols, 01 2020. Viewed on 05 Jan 2020, <https://www.postscapes.com/internet-of-things-protocols/>.

[72] Sattar J. Abound. Identification protocols. *ResearchGate*, 01 2014.

[73] Wikipedia. Ip adress, 01 2020. Viewed on 10 Dec 2019, <https://en.wikipedia.org/wiki/IP_address>.

[74] Study.com. Ipv4 address: Structure, classes and types. Viewed on 25 Dec 2019, <https://study.com/academy/lesson/ipv4-address-structure-classes-and-types.html>.

[75] Mohan J. Build iot prototype rapidly with thingspeak, 06 2019. Viewed on 20 Nov 2019, <https://www.g2.com/products/thingspeak/reviews>.

[76] Libelium. Libelium adds compatibility with thingspeak as a new cloud partner for easy development of iot apps, 2018 10. Viewed on 03 Jan 2020, <http://www.libelium.com/libelium-adds-compatibility-with-thingspeak-as-a-new-cloud-partner-for-easy-development-of-iot-apps/>.

[77] Azazeal. Why do you need ttn?, 11 2018. Viewed on 29 Oct 2019, <https://www.thethingsnetwork.org/forum/t/why-do-you-need-ttn/20040>.

[78] bsi. The things network and eclipse scout. Viewed on 03 Jan 2020, <https://www.bsi-software.com/en/scout-blog/article/the-things-network-and-eclipse-scout.html>.

[79] Dragino. Lora questions - what is the difference between lg01 single channel gateway and a 8-channel lorawan gateway?, 04 2018. Viewed on 03 Nov 2019, <https://wiki.dragino.com/index.php?title=LoRa_Questions#What_is_the_difference_be
channel_LoRaWAN_gateway.3F>.

[80] Wikipedia. Lamp (software bundle), 11 2019. Viewed on 12 Oct 2019, <https://en.wikipedia.org/wiki/LAMP_(software bundle)>.

[81] Domatan G. What is apache? an in-depth overview of apache web server, 11 2019. Viewed on 20 Nov 2019, <https://www.hostinger.in/tutorials/what-is-apache>.

[82] Jackson B. 10 best free dns hosting providers, 10 2018. Viewed on 21 Nov 2019, <https://www.keycdn.com/blog/best-free-dns-hosting-providers>.

[83] Bueno M. *El huerto familiar ecológico*. Integral, 4 edition, 05 2004.

[84] Arduino. Arduino nano. Viewed on 05 Jan 2020, <https://store.arduino.cc/arduino-nano>.

[85] Libelium. New wifi module for waspmote to connect directly to cloud servers and iphone/android platforms, 02 2012. Viewed on 05 Jan 2020, <http://www.libelium.com/wifi_sensors_connect_to_cloud_iphone_android/>.

[86] Hampton C.R. Update the firmware in your esp8266 wi-fi module, 08 2015. Viewed on 05 Jan 2020, <https://www.allaboutcircuits.com/projects/update-the-firmware-in-your-esp8266-wi-fi-module/>.

[87] Wikipedia. Lora, 11 2019. Viewed on 15 Oct 2019, <https://en.wikipedia.org/wiki/LoRa>.

[88] Arduino. Arduino uno rev3. Viewed on 12 Sept 2019, <https://store.arduino.cc/arduino-uno-rev3>.

[89] Amazon. Dragino lora shield 915mhz, rfm95w wiless, compatible with arduino uno mega 2560 leonardo due, 3.3v or 5v low power consumption, antenna ipex. Viewed on 10 Sept 2019, <https://www.amazon.in/Dragino-Compatible-Arduino-Leonardo-Consumption/dp/B07HD1MH3J>.

[90] Pino tech. Soilwatch 10 soil moisture sensor. Viewed on 5 Oct 2019, <https://pino-tech.eu/soilwatch10/>.

[91] Adafruit. Sht-30 mesh-protected weather-proof temperature/humidity sensor - 1m cable. Viewed on 16 Oct 2019, <https://www.adafruit.com/product/4099>.

[92] Thepihut. Immersible pump and watertube. Viewed on 10 Oct 2019, <https://thepihut.com/products/immersible-pump-watertube>.

[93] Adafruit. Medium 6v 2w solar panel - 2.0 watt. Viewed on 05 Oct 2019, <https://www.adafruit.com/product/200>.

[94] Adafruit. Usb / dc / solar lithium ion/polymer charger - v2. Viewed on 05 Oct 2019, <https://www.adafruit.com/product/390>.

[95] Adafruit. 3.8 / 1.3mm or 3.5 / 1.1mm to 5.5 / 2.1mm dc jack adapter cable. Viewed on 05 Oct 2019, <https://www.adafruit.com/product/2788>.

[96] Pimoroni. Lipo battery pack - 2000mah. Viewed on 05 Oct 2019, <https://shop.pimoroni.com/products/lipo-battery-pack?variant=20429082247>.

[97] Adafruit. Powerboost 500 basic - 5v usb boost @ 500ma from 1.8v+. Viewed on 05 Oct 2019, <https://www.adafruit.com/product/1903>.

[98] Adafruit. Adafruit micro-lipo charger for lipo/liion batt w/microusb jack - v1. Viewed on 05 Oct 2019, <https://www.adafruit.com/product/1904>.

[99] Wikipedia. localhost, 02 2020. Viewed on 12 Oct 2019, <https://en.wikipedia.org/wiki/Localhost>.

[100] No-ip. Free dynamic dns : Getting started guide. Viewed on 10 Nov 2019, <https://www.noip.com/support/knowledgebase/getting-started-with-no-ip-com/>.

[101] Wikipedia. Ajax (programming), 12 2019. Viewed on 12 Oct 2019, <https://en.wikipedia.org/wiki/Ajax_(programming)>.

[102] Dragino. Lora shield, 10 2017. Viewed on 03 Jan 2020, <https://wiki.dragino.com/index.php?title=Lora_Shield>.