TUNISIAN REPUBLIC
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF TUNIS EL MANAR

HIGHER INSTITUTE OF COMPUTER SCIENCE

FINAL STUDIES PROJECT REPORT

Presented in order to obtain

National Diploma Of Applied Licence in Science and Technology

Mention: Industrial Data Processing

Specialty: Embedded Systems

By

Faicel Ben Saïd

Mohamed Neji Ghazouani

Design and Implementation of a Smart Irrigation System

Academic supervisor:	**Mrs.Faten Ben Aicha**	Assistant Professor
Academic supervisor	**Mr.Mohamed Laaraiedh**	Assistant Professor

elaborated within Higher Institute Of Computer Science

University Year: 2018 - 2019

TUNISIAN REPUBLIC
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF TUNIS EL MANAR

HIGHER INSTITUTE OF COMPUTER SCIENCE

FINAL STUDIES PROJECT REPORT

Presented in order to obtain

National Diploma Of Applied Licence in Science and Technology

Mention: Industrial Data Processing

Specialty: Embedded Systems

By

Faicel Ben Saïd

Mohamed Neji Ghazouani

Design and Implementation of a Smart Irrigation System

Academic supervisor:	**Mrs.Faten Ben Aicha**	Assistant Professor
Academic supervisor	**Mr.Mohamed Laaraiedh**	Assistant Professor

elaborated within Higher Institute Of Computer Science

University Year: 2018 - 2019

I authorize the students to submit their internship report for evaluation.

Academic supervisor, **Faten BEN AICHA**

I authorize the students to submit their internship report for evaluation.

Academic supervisor, **Mohamed Laaraiedh**

Dedications

For the project is dark and full of terrors, i needed light to guide me through this darkness. The light can come in many shapes: family, friends, a television series or even a cat.
For a special someone who i may never speak to after this project, i want to say "Valar Morghulis".

Faicel Ben Saïd

This work is dedicated to my family and especially my sisters Sarah and Sondos who significantly contributed to my success. I would like to also to my close friends for their help and support. I also want to dedicate this project to my aunt's soul. Finally, a reply from that special someone "Valar dohaeris".

Mohamed Neji Ghazouani

ACKNOWLEDGEMENTS

First of all, we would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our professors and academic supervisors Mrs.Faten Ben Aicha, we couldn't do it without your smile, as well for Mr.Mohamed Laaraiedh, you have been an inspiration to this project from the beginning. It was a real privilege and a great honor for us to work under your supervision, share your knowledge and most of all for wasting your precious time.

Furthermore we would also like to acknowledge with much appreciation all of those whom supported us somehow along the way. Family, friends, werewolf people, director Mrs.Monia Najjar... Thank you all, you are awesome.

Contents

Introduction **1**

1 Project context **2**

Introduction . . . 3

1.1 IoT and Smart Farming . . . 3

1.1.1 IoT . . . 3

1.1.2 Smart Farming . . . 5

1.2 Problematic . . . 5

1.3 Existing Solutions . . . 6

1.4 About the Project . . . 9

1.4.1 Project Specification . . . 9

1.4.2 Objectives . . . 9

Conclusion . . . 9

2 Architecture and UML Modeling **10**

Introduction . . . 11

2.1 Requirements analysis . . . 11

2.1.1 Functional Requirements . . . 11

2.1.2 Non-functional Requirements . . . 11

2.2 System architecture . . . 11

2.3 System Modeling . . . 12

2.3.1 Use Case diagram . . . 13

2.3.2 Sequence diagram . . . 14

Conclusion . . . 15

3 Work Environment **16**

Introduction . . . 17

3.1 Embedded IOT Protocols Environment . . . 17

3.1.1 Sigfox . . . 17

3.1.2 NB-IoT . . . 17

3.1.3 Lora and Lorawan . . . 17

3.1.4 Comparison . . . 18

3.2 Hardware Environment . . . 19
3.2.1 Raspberry Pi 2 Model B . . . 19
3.2.2 Arduino . . . 21
3.2.3 DHT11 . . . 25
3.2.4 Soil moisture sensor . . . 26
3.2.5 DC motor . . . 28
3.2.6 Relay . . . 29
3.2.7 RFM95 Lora module . . . 29
3.2.8 Dragino Lora/GPS hat . . . 31
3.3 Software Environment . . . 32
3.3.1 Visual studio . . . 32
3.3.2 SQL Server Management Studio 2017 . . . 34
3.3.3 Star UML 2 . . . 34
3.3.4 Arduino IDE . . . 34
3.3.5 Device Explorer . . . 34
Conclusion . . . 34

4 Solution Implementation **35**
Introduction . . . 36
4.1 Gateway implementation . . . 36
4.1.1 Gateway Configuration . . . 36
4.1.2 Gateway implementation to The Things Network . . . 37
4.2 Nodes Implementation . . . 38
4.2.1 Sensor node implementation . . . 38
4.2.2 Sensor/Actuator Node Implementation . . . 42
4.3 Microsoft Azure cloud implementation . . . 44
4.3.1 TTN to Azure Iot Hub . . . 44
4.3.2 Mobile application to Azure Cloud . . . 46
4.4 Mobile Application implementation . . . 47
4.4.1 Mobile Application frontend . . . 47
4.4.2 Mobile Application backend . . . 52
Conclusion . . . 53

Conclusion and Perspectives **54**

Bibliography **55**

Annexes **57**

List of Figures

1.1 IoT basic 4 stages Architecture 4
1.2 Plantix Logo 6
1.3 Phyt'eau interface 7
1.4 Senya interface 7
1.5 Rachio 3 8
1.6 Orbit B-hyve 57915 8

2.1 System Architecture 12
2.2 General Use Case Diagram 13
2.3 General Sequence Diagram 14

3.1 Raspberry Pi 2 20
3.2 Arduino boards 22
3.3 The Arduino Mega 2560 Rev3 23
3.4 The Arduino Nano 24
3.5 DHT11 25
3.6 Capacitive sensor 25
3.7 The thermistor(NTC) operating mode 26
3.8 The Soil moisture sensor 27
3.9 The 755 motor 28
3.10 The Relay 29
3.11 The RFM95 30
3.12 The Dragino Lora Shield 31

4.1 "main.cpp" Configuration 36
4.2 Dragino Lora Shield with Raspberry Pi 2 37
4.3 gateway Lunching 37
4.4 TTN Status 38
4.5 Sensor Node 38
4.6 Soil Moistre Node 39
4.7 Personal interface 39
4.8 Device registration 40

4.9 Access keys . . . 40
4.10 Config.h . . . 41
4.11 Arduino Code . . . 41
4.12 Exadecimal String Format . . . 42
4.13 Payload Format . . . 42
4.14 Original Format . . . 42
4.15 Actuator Node . . . 43
4.16 DHT11 Node . . . 43
4.17 DHT11 Data . . . 44
4.18 TTN Azure Bridge . . . 45
4.19 Device Explorer Monitor . . . 45
4.20 Azure Iot Hub Devices . . . 45
4.21 SQL management studio Test . . . 47
4.22 sign in View . . . 48
4.23 Sign Up View . . . 49
4.24 Home View . . . 50
4.25 Settings View . . . 51
4.26 About View . . . 52
4.27 Arduino Mega 2560 Pin Mapping . . . 57
4.28 Arduino Nano Pin Mapping . . . 58
4.29 RFM 95 Pin Mapping . . . 59

List of Tables

3.1 Embedded IOT Protocols Review 18
3.2 Cards review 20
3.3 Arduino Boards review 22

Abreviations list

- **AI** = **A**rtificial **I**ntelligence
- **API** = **A**pplication **P**rogramming **I**nterface
- **EEPROM** = **E**lectrically **E**rasable **P**rogrammable **R**ead- **O**nly **M**emory
- **GPIO** = **G**eneral **P**urpose **I**nput / **O**utput
- **GPRS** = **G**eneral **P**acket **R**adio **S**ervice
- **GPS** = **G**lobal **P**ositioning **S**ystem
- **GSM** = **G**lobal **S**ystem for **M**obile communication
- **HTTP** = **H**yper **T**ext **T**ransfer**P**rotocol
- **IDE** = **I**ntegrated **D**evelopment **E**nvironment
- **IOT** = **I**nterne **O**f **T**hings
- **IT** = **I**nformation **T**echnology
- **LPWAN** = **L**ONG **P**ower **W**ide **A**rea **N**etwork
- **LoRa** = **L**ONG **R**ange
- **LoRaWAN** = **L**ONG **R**ange **W**ide **A**rea **N**etwork
- **MQTT** = **M**essage **Q**ueuing **T**elemetry **T**ransport
- **NB- IOT** = **N**arrow **B**and- **I**nterne **O**f **T**hings
- **NoSQL** = **N**ot **O**nly **S**ructured **Q**uery **L**anguage
- **PWM** = **P**ulse **W**idh **M**odulation
- **RDBMS** = **R**elational **D**ata **B**ase **M**anagement **S**ystem
- **RFM** = **R**adio **F**requency **M**odule
- **SF** = **S**preading **F**actor
- **SPI** = **S**erial **P**eripheral **I**nterface

- **SQL** = **S**ructured **Q**uery **L**anguage
- **SRAM** = **S**tatic **R**andom **A**ccess **M**emory
- **TTN** = **T**he **T**hings **N**etwork
- **UML** = **U**nified **M**odeling **L**anguage

Introduction

We all learned things from other people who were willing to share their knowledge with us in the same way we share our knowledge with others. In this sense human society can be viewed as a network of "knowledge" the power of which can be appreciated by the fact that most of the things we know actually come from our participation in this network. In the same way that humans possess knowledge about the world so do another more precise, reliable, relentless aspects (like IoT , Big Data , Artificial Intelligence etc) which they store as database information. So imagine the possibilities that would arise if this knowledge could be shared in the same way as humans and used properly.

The IoT system is a network of objects where all participants are connected to the Internet and act as data providers, receivers or manipulators that takes advantage of the collective service provided by other participants. The dynamic nature of the network and the fact that its power increases as more peers join makes the IoT community a very attractive model and has been used successfully in the major domains.

Typically different domains will have different use for the same type of information. The problem in sharing this info is how to integrate the different uses so that the participants can communicate. For instance, smart is everywhere but it is not totally available to everyone because smart is expensive. In this context, our project aims to design and implement a low cost IoT network for the benefit of smart and precise agriculture.

This report is organized in four chapters, the first chapter is a context chapter that presents a description of the project. In Second chapter we set the functional and non-functional requirements and model our solutions behaviour using UML. Furthermore, In chapter three, we set our work environment. Finally, the last chapter summarizes the project implementation.

Chapter 1

Project context

Plan

Introduction 3

1 IoT and Smart Farming 3

2 Problematic 5

3 Existing Solutions 6

4 About the Project 9

Conclusion 9

Introduction

1.1 IoT and Smart Farming

1.1.1 IoT

"To invent, you need the Internet, communication, good imagination and a pile of things." –Ammar Rayes

The Internet of Things has been many years in the making. Indeed, the concept of using sensor devices to collect data and then transfer it to applications across a network has been around for several decades. However, interconnected sensors and actuators are being altered by market shifts and technology trends in recent years. The combination of low-cost hardware and high-speed networking technologies, both wired and wireless, have enabled a new generation of compact sensor devices with ubiquitous connectivity across the wider Internet. These systems are facilitating real-time data collection, sharing and providing unprecedented visibility and control of assets, personnel, operations and processes. The further use of cloud-based computing & storage facilities is introducing even more advanced data analysis capabilities, ushering in a new era of intelligent decision-making, control, and automation. Broadly, these new paradigms are termed as the Internet of Things.In short, IoT is a system of system [1].

1.1.1.1 IoT Architecture

IoT architecture varies from solution to another, based on its type which we intend to build [2]. IoT as a technology majorly consists of four main components, over which an architecture is framed.

- Sensors
- Devices
- Gateway
- Cloud

The following is the basic 4 Stage Architecture of IoT example:

- **Stage 1:**
  - **Sensors:** Sensors collect data from the environment or object under measurement and turn it into useful data.

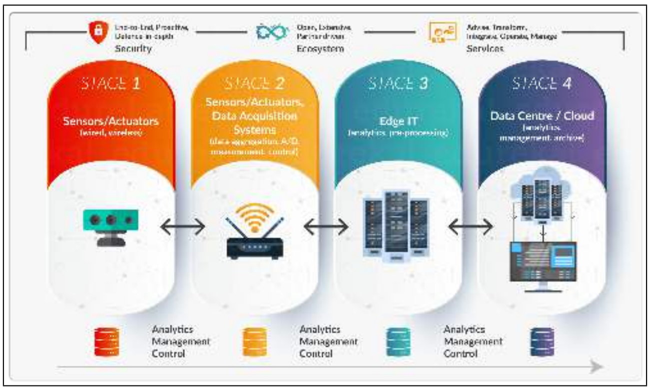


Figure 1.1: IoT basic 4 stages Architecture

  - **Actuators:** Actuators can also intervene to change the physical conditions that generate the data. An actuator might, for example, shut off a power supply, adjust an air flow valve or move a robotic gripper in an assembly process.
  - **Sensing/Actuating** stage covers everything Example: Industrial devices to robotic camera systems, water-level detectors, air quality sensors and heart rate monitors.

- **Stage 2:**

  The stage 2 systems often sit in close proximity to the sensors and actuators.

  For Example: a pump might contain a half-dozen sensors and actuators that feed data into a data aggregation device that also digitizes the data. This device might be physically attached to the pump. An adjacent gateway device or server would then process the data and forward it to the Stage 3 or Stage 4 systems.

- **Stage 3:**

  - Once IoT data has been digitized and aggregated, it's ready to cross into the realm of IT.
  - However, the data may require further processing before it enters the data centre.
  - This is where edge IT systems, which perform more analysis, come into play.
  - Edge IT processing systems may be located in remote offices or other edge locations, but generally these sit in the facility or location where the sensors reside closer to the sensors, such as in a wiring closet.

- **Stage 4:**

  - The data from **Stage 3** is forwarded to physical data centre or cloud-based systems, where more powerful IT systems can analyze, manage and securely store the data.

- It takes longer to get results when you wait until data reaches **Stage 4**, but you can execute a more in-depth analysis, as well as combine your sensor data with data from other sources for deeper insights.
- **Stage 4** processing may take place on-premises, in the cloud, or in a hybrid cloud system, but the type of processing executed in this stage remains the same, regardless of the platform.

1.1.2 Smart Farming

Smart Farm is all about intelligent irrigation and smarter farming. It involves the incorporation of information and communication technologies into machinery, equipment and sensors for use in and has a real potential to deliver a more productive and sustainable agricultural production, based on a more precise and resource-efficient approach [3]. In this sense, smart farming is strongly related to three interconnected technology fields:

- Management Information Systems: Planned systems for collecting, processing, storing and disseminating data in the form needed to carry out a farm's operations and functions.
- Precision Agriculture: Management of spatial and temporal variability to improve economic returns following the use of inputs and reduce environmental impact.
- Agricultural automation and robotics: The process of applying robotics, automatic control and artificial intelligence techniques at all levels of agricultural production, including farmbots and farmdrones.

Smart Farming applications do not target only large, conventional farming exploitation, but could also be new levers to boost other common or growing trends in agricultural exploitation, such as family farming and organic farming.

1.2 Problematic

The IoT is of a great importance in Tunisia, unfortunately it is limited due to lack of resources, financing and specialized human resources. As a result, it affects the evolution of every aspect.

The city of Menzel Bou Zelfa. Located forty kilometers south east of Tunis and fifty kilometers north of Nabeul, thirty four villages and 14453 hectares of farming land. The agricultural city is known as the capital of oranges.

Being a farmer in such area these days is a challenge with all the difficulties ahead such as the lack of water resources especially during summer, also the augmentation of the salinity of the water table and the rise of production costs compared to lower selling prices. Interestingly, approximately 70 percent of the total volume of water withdrawals in the world are used for irrigation and that's precisely where most of the water waste happens. Around 60 percent of the water meant to be used for irrigation is lost, either due to evapotranspiration, land runoff or simply inefficient and primitive usage methods. This, in turn, brings to light the importance of smart irrigation powered by the internet of things [4].

1.3 Existing Solutions

The existing systems of irrigation are primitive and depends entirely on the full presence of manpower and the use of technology is very limited and usually none existent. Yet on a national and international scale exists a variation of solutions that consists either on image processing or data gathering and monitoring. Here we state the most important:

Plantix is a free, simple yet reliable application that identifies through photos the diseases and pests that attacks plants and gives access to relative information to prevent. Unidentified technologies [5].

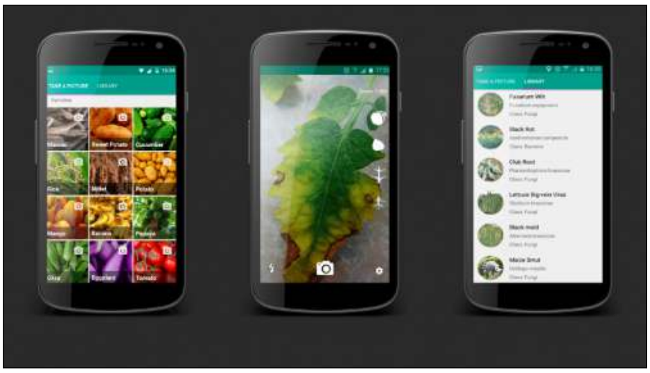


Figure 1.2: Plantix Logo

Phyt'eau is a platform to measure up agroclimatic conditions of the farm to achieve a precise irrigation [6].

Figure 1.3: Phyt'eau interface

Senya is a Web/Mobile platform for record keeping of the farm and also a dashboard that allows temperate management of your crops and visualization of activities and input inventory within the farm [7].

Figure 1.4: Senya interface

Rachio 3 is the best smart sprinkler controller you can buy because, quite simply, it does exactly what it's supposed to. It connects to either your 2.4GHz or 5GHz Wi-Fi network and is certified to work with Alexa. Download the app on iPhone or Android. It even uses satellite information to skip programmed times based on rain and wind data. If you want the total package, this is the one you want [8].

Figure 1.5: Rachio 3

B-hyve is a four-station controller, which has apps for both iOS and Android. It features weather-sensing technology, which enables smart watering options based on slope, soil type, amount of sun, live weather feeds, and more. There's a built-in override if you really need to water your lawn or garden and it may even make you eligible for rebates from your local water service [9].

Figure 1.6: Orbit B-hyve 57915

Some of these solutions is very expensive, the farmers avoid the risk of using such systems because they consist on replacing the existing structure, even though these solutions are well tested and promise a production increase the farmers still do not trust technology. That is why this kind of technology is needed to make a positive impact on every farmer's most invaluable resources, time, water, structure and hopefully trust.

1.4 About the Project

Our solution consists on meeting the farmer half-way, in other words we are making a subsystem for the existing structure, we will enhance its productivity to the limit.

1.4.1 Project Specification

The project aims to design and implement a system which allows the user to monitor and control his entire farm from anywhere using the mobile application from any smart device or PC. The application displays environmental information such as temperature and humidity. The system also monitor and adjust soil moisture and check on real time pump status. The system also will provide a variation of options and it is all about precision agriculture.

The requested tasks are :

- Implementation of sensor nodes.
- Implementation of a Lora Gateway.
- Implementation of a Lora Server.
- Implementation of a Mobile application.

1.4.2 Objectives

Eventually, for our objectives we will be aiming to:

- Help farmers to avoid water wastage.
- Minimizing runoffs and other wastage.
- Neglecting the human errors.
- Saving energy, time, and valuable. resources

Conclusion

In this chapter, we defined the context of our project, reviewed some of the existing solutions and explained our perspectives and objectives. The farmers problems wont disappear in a single solution but we're gonna get them all one at a time.

Chapter 2

Architecture and UML Modeling

Plan

Introduction **11**

1 Requirements analysis **11**

2 System architecture **11**

3 System Modeling **12**

Conclusion **15**

Introduction

After defining the general context of our project. This chapter presents the functional and non-functional requirements of our application. The chapter subsequently present a global description of our solution and concludes by modeling the behaviour of our system with UML.

2.1 Requirements analysis

This section is about the functional and non-functional requirement of the system. In short, a functional requirement describes what a software system should do, while non-functional requirements place constraints on how the system will do so.

2.1.1 Functional Requirements

The system will provide the user with the following specific behaviors or functions:

- Obligatory authentication.
- Checking humidity level.
- Checking temperature level.
- Checking pump status.
- Receiving notification of high or low soil moisture level.
- Reminder for the manual mode.

2.1.2 Non-functional Requirements

Our system should meet the following non-functional requirements: • **Une maîtrise des différents aspects du projet :**

2.2 System architecture

The aim of the present designed architecture is to meet the previously mentioned requirements. We suggest 3 nodes:

- Two separate soil moisture and temperature nodes.
- A third node that provides air humidity, temperature and controls pump and provides its status.

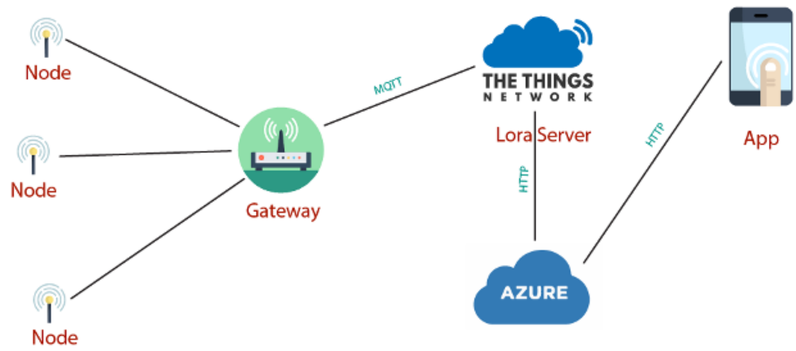


Figure 2.1: System Architecture

These nodes are connected to the gateway via Lora technology that send and receive information in order to process it. The actuator will action the water pump, start the system and also communicates with the gateway. The gateway itself is Lora powered and its role is to receive data and transfer them throughout Ethernet to a cloud server, then to a platform where the processing actually happens and the actions sent back to the actuator appears (Figure 2.1).

2.3 System Modeling

There were a variety of choices for this task, we finally decided to settle on a Use Case and a global sequence diagram of the system using UML language and a software named StarUML.

2.3.1 Use Case diagram

We used the use case diagram model to define the functionality of a system using actors and a set of actions, services, and functions that the system needs to perform. Numerous attempts were made to finally present the final version of the Use Case diagram in figure 2.2.

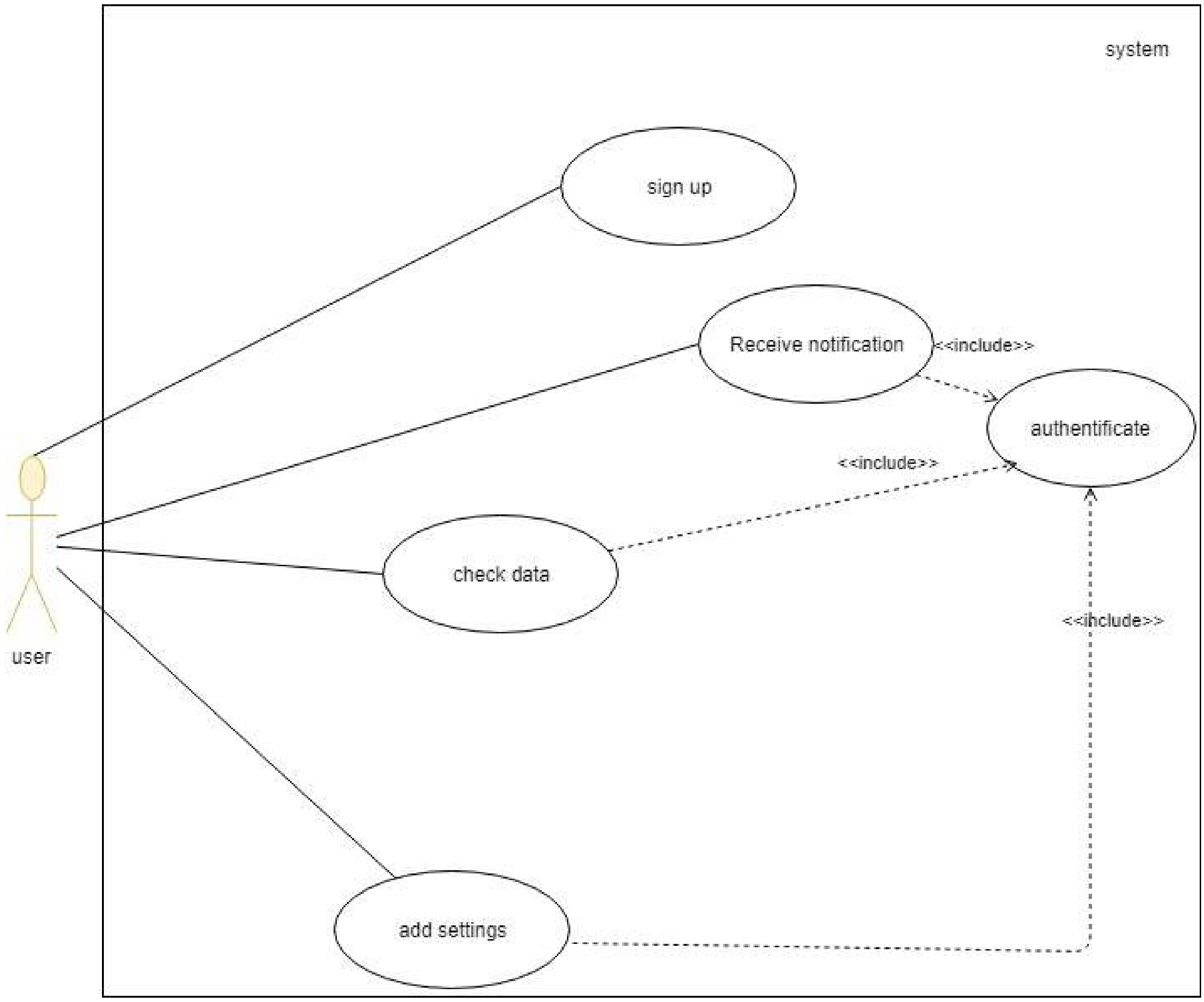


Figure 2.2: General Use Case Diagram

2.3.2 Sequence diagram

Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

The result of the General Sequence Diagram in figure 2.3 :

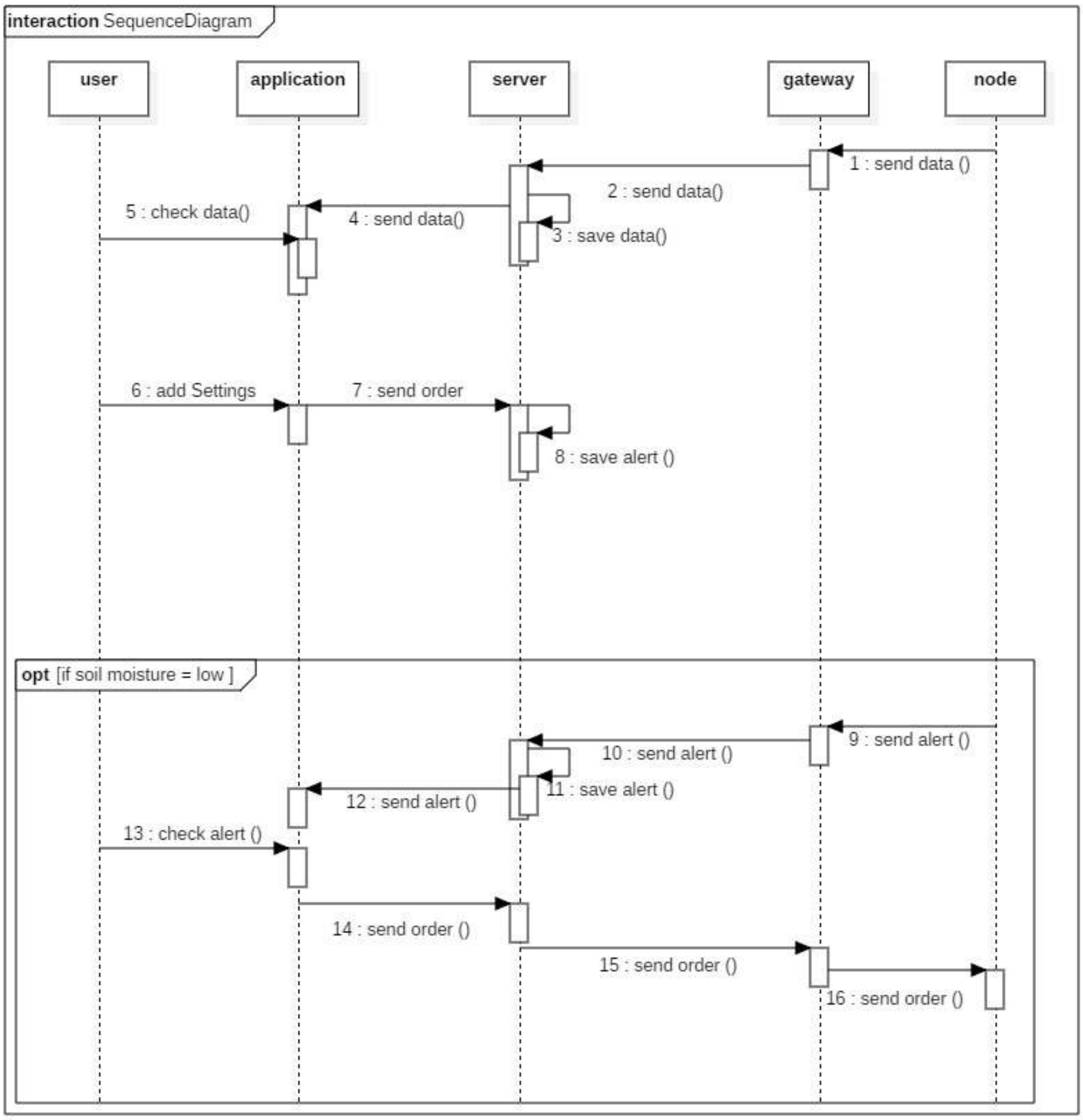


Figure 2.3: General Sequence Diagram

Conclusion

This chapter summarizes a step by step progress leading to the UML conception which explains the system interactions while attached to the proposed architecture composed of four main parts:five nodes,a gateway, a server and a mobile application that meets the expectations of the functional and non-functional requirements. In the next chapter we will explain our work environment.

Chapter 3

Work Environment

Plan

Introduction 17

1 Embedded IOT Protocols Environment 17

2 Hardware Environment 19

3 Software Environment 32

Conclusion 34

Introduction

After explaining the system interactions that meets the expectations of the functional and non-functional requirements. In This chapter, embedded IOT protocols, hardware and software are the portal to our work environment.

3.1 Embedded IOT Protocols Environment

In this section, we will define some of the major embedded IOT protocols in the world and compare them to figure out the most suitable for our solution.

3.1.1 Sigfox

Sigfox is a Low Power Wide Area Network (LPWAN) that works in 868 and 915 MHZ, it can send a small amount of data with 60 bit/s data rate per day (140 messages per day , 1 message every 10 minutes) and the payload of the message being transfer can't depass 12 bytes. It is bidirectional,for the downlink side sigfox node can only receive 4 messages per day [10].

3.1.2 NB-IoT

NarrowBand-Internet of Things (NB-IoT) is a standards-based low power wide area (LPWAN) technology developed to enable a wide range of new IoT devices and services. New physical layer signals and channels are designed to meet the demanding requirement of extended coverage rural and deep indoors and ultra-low device complexity. Initial cost of the NB-IoT modules is expected to be comparable to GSM/GPRS. The underlying technology is however much simpler than today's GSM/GPRS and its cost is expected to decrease rapidly as demand increases20. [11].

3.1.3 Lora and Lorawan

Long Range Wide-area network(LoRaWAN) is a Low Power Wide Area Network (LPWAN) with specification intended for wireless battery operated devices in regional, national or global network. It is Based on Long Range (LoRa) technology. This technology defines physical and partially data link layers. This open standard based technology uses sub-GHz bands which means that range and obstacle penetration is much better. In other words, LoRaWAN defines the communication protocol and system architecture for the network, while the LoRa physical layer enables the long-range communication link. LoRaWAN supports low-cost, mobile, and secure bi-directional communication for Internet of Things (IoT), machine-to-machine (M2M) and smart city industrial applications [12].

3.1.4 Comparison

In the following table, there is a comparison of the aspects that interest our solution in the mentioned embedded IOT protocols [13].

Table 3.1: Embedded IOT Protocols Review

Name	LoraWan	Sigfox	NB-IoT
Typical Range	14 km	17 km	22 km
Regulation	License free ISM band	License free ISM band	expensive dedicated regional frequency/channel
Signal Bandwidth	125 KHz	0.1 KHz	180 KHz
Data Rate	10 Kbps	100 bps	200 Kbps
Max Output POWER	0.025 W	0.025 W	0.2 W (max)
Battery life	10 years	10 years	10 years
Security	AES	Every device does have a unique static ID	LTE's authentication and encryption

After comparing, we still unable to decide the most suitable embedded IOT protocol so we will be considering the benefits and drawbacks of each :

3.1.4.1 Sigfox benefits and drawbacks

In Sigfox's favor, it works well for simple devices that transmit infrequently, because it sends very small amounts of data very slowly and it supports a wide coverage area in the areas where it is located. Sadly, Sigfox is ruled off the list because it is not deployed everywhere specially in Tunisia were it has 0% coverage.

3.1.4.2 NB-IoT benefits and drawbacks

NB-IoT devices rely on 4G coverage, so they would work well indoors and in dense urban areas. It has faster response times than LoRa and can guarantee a better quality of service. Again sadly this new technology is not available in Tunisia.

3.1.4.3 LoraWan benefits and drawbacks

Needless to state benefits and drawbacks, LoraWan is our choice per obligation yet it has its unique qualities such as:

– set up and manage personal network.

– LoRa is a good option if user needs bidirectionality.

– LoRa devices work well when they are in motion.

Despite that Lora has a longer latency time than NB-IoT and requires a gateway to work, in many cases that could be an advantage.

3.2 Hardware Environment

In this section we'll be focusing on the hardware side of the system. Essentially, it will be composed of :

– A Raspberry Pi 2 Model B.

– An Arduino Mega 2560 .

– Two Arduino Nano.

– A DHT11.

– A soil moisture sensor.

– A Relay.

– A DC motor.

– Three RFM95 Lora module.

– A dragino Lora/GPS hat.

3.2.1 Raspberry Pi 2 Model B

The Raspberry Pi 2 model B series (figure3.1), based on the ARM Cortex-A7 quad-core processor, is the second generation of the RaspberryPi. This last generation of a copy gap enables the execution of several variants of the GNU / Linux operating system [14].

Our choice for this specific card is validated by the solution necessity for a gateway with an easy connection to the internet, a choice for the programming language and its best feature is the compatibility with the Dragino Lora/GPS hat. We would like to compare with some of the existing alternatives like the arduino Uno and the NetBurner MOD54415 in the table 3.2.

Table 3.2: Cards review

Product	Arduino Uno	Raspberry Pi 2	NetBurner MOD54415
Operating system	None	Real Time OS	Linux
Programming Language	Arduino	C/C++	Multiple
Low Power	Yes	No	No
Native Networking	No	Yes	Yes
Multitask apps and server usage	No	Yes	Yes
RAM	2KB	64MB	1GB
Memory	32 KB	32MB	None (SD card)
Processor Clock Speed	16 MHz	250 MHz	1.4 GHz
External Storage	None	USB/SD	USB/SD
Price	22.95$	99,00$	35$

Figure 3.1: Raspberry Pi 2

Specification:

- SoC: Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
- CPU: 900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
- GPU: Broadcom VideoCore IV @ 250 MHz
- Memory: 1 GB (shared with GPU)
- USB ports: 4
- Video input: 15-pin MIPI camera interface (CSI) connector
- Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
- Audio input: IS
- Audio outputs: Analog via 3.5 mm jack; digital via HDMI and IS
- Storage: MicroSD
- Network: 10/100Mbps Ethernet
- Peripherals: 17 GPIO plus specific functions, and HAT ID bus
- Power rating: 800 mA (4.0 W)
- Power source: 5 V via MicroUSB or GPIO header
- Size: 85.60mm 56.5mm
- Weight: 45g (1.6 oz)

3.2.2 Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. All Arduino boards have one thing in common: they are programmed through the Arduino IDE. Beyond that, there can be a lot of differences. The number of inputs and outputs, speed, operating voltage, and form factor are just a few of the variables. Illustrated in figure 3.2 is a variation of the arduino boards.

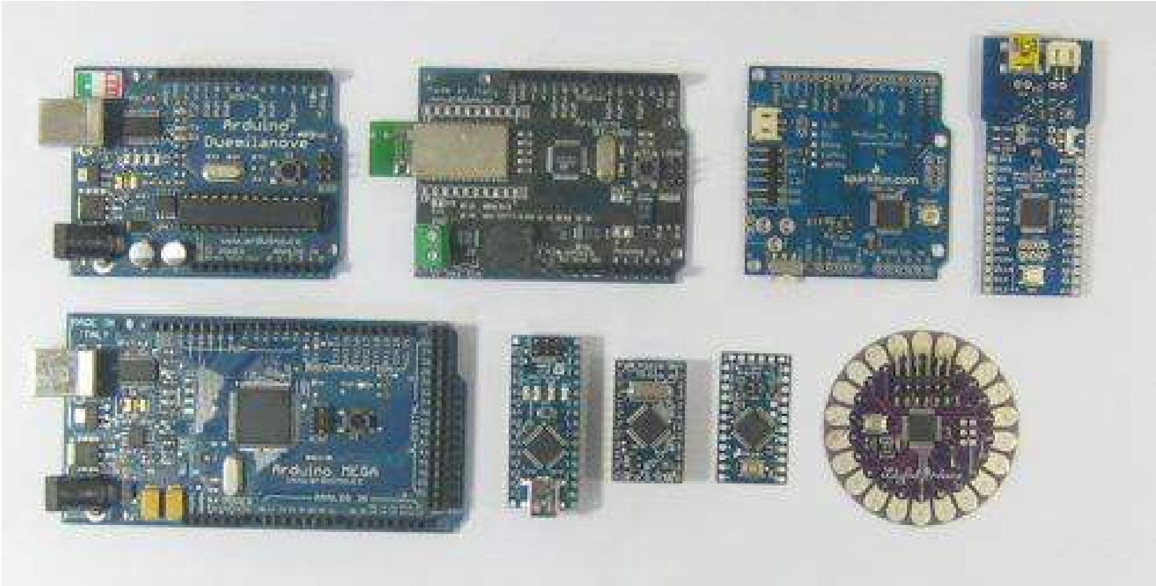


Figure 3.2: Arduino boards

In table 3.3 we will explain the difference between some of the arduino boards.

Table 3.3: Arduino Boards review

Name	Uno	Mega 2560	Nano	Mini
Processor	ATmega328P	ATmega2560	ATmega168	ATmega328P
Operating/Input Voltage	5 V / 7-12 V	5 V / 7-12 V	5 V / 7-9 V	5 V / 7-9 V
CPU Speed	16 MHz	16 MHz	16 MHz	16 MHz
Analog In/Out	6/0	16/0	8/0	8/0
Digital IO/PWM	14/6	54/15	14/6	14/6
EEPROM	1KB	4KB	0.512KB	1KB
SRAM	2KB	8KB	1KB	2KB
Flash	32KB	256KB	16KB	32KB
Price	22.95$	29.85$	2.03$	9.95$

After the comparison between the different adruino boards we concluded that the best choice for our senso/actuator is the arduino mega (figure 3.3) due to our need to high number of inputs and outputs and the necessary speed. Also our choice for the sensors node will be the arduino nano (figure3.4) since it is a perfect match for our sensor needs despite it is slightly less-performant than the arduino mini but much cheaper [15] [16].

Figure 3.3: The Arduino Mega 2560 Rev3

Specification of Arduino Mega2560 Rev3:

– ATmega2560 Processor

– 54 digital input/output pins (of which 15 can be used as PWM outputs)

– 16 analog inputs

– 4 UARTs (hardware serial ports)

– MHz crystal oscillator

– USB connection

– power jack

– ICSP header, and a reset button

– 256 KB Flash memory (of which 8 KB used by bootloader)

– 8 KB Sram

– 4 KB EEPROM

– Operating/Input Voltage 5 V / 7-12 V

– 20 mA DC Current per I/O Pins

– 50 mA DC Current for 3.3V Pin

– 16 MHz CPU Speed

– 13 LED BUILTIN

– 101.52 x 53.3 mm PCB Size

– 37g Weight

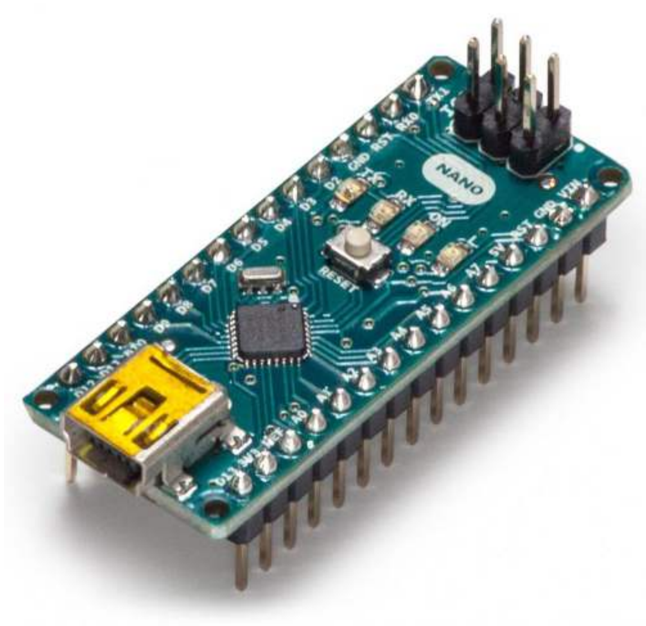


Figure 3.4: The Arduino Nano

Specification of Arduino Nano:

– ATmega2560 Processor

– 22digital input/output pins (of which 6 can be used as PWM outputs)

– 32 KB flash memory (of which 2 KB used by bootloader)

– 2 KB Sram

– 16 MHz Clock Speed

– USB connection

– 8 analog inputs

– 1 KB EEPROM

– 40 mA DC Current per I/O Pins

– Operating/Input Voltage 5 V / 7-12 V

– 19 mA Power Consumption

– 18 x 45 mm PCB Size

– 7g Weight

3.2.3 DHT11

The DHT11 is a basic, low-cost digital temperature and humidity sensor [17]. It uses a capacitive humidity sensor, a thermistor to measure the surrounding air and IC on the back side of the sensor. Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

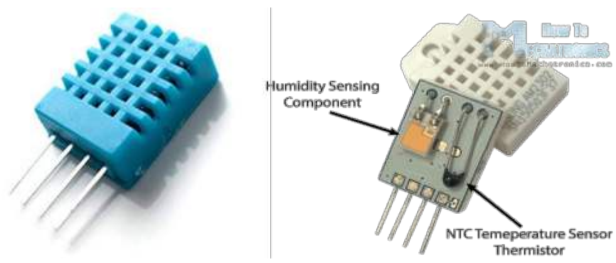


Figure 3.5: DHT11

Operating mode To measure humidity it uses capacitive humidity sensor which has two electrodes with moisture holding substrate between them. As the humidity level changes, the resistance between both of the electrodes or conductivity also changes correspondingly. This change is measured and processed by the IC which makes it ready to be read by a microcontroller. Figure 3.6 depicts the capacitive sensor of the DHT11.

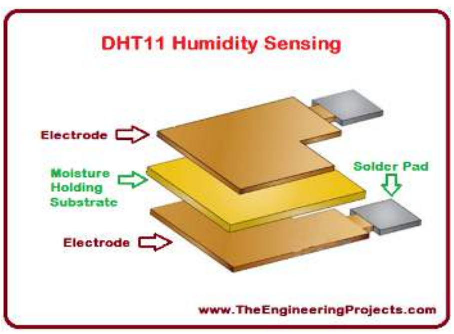


Figure 3.6: Capacitive sensor

To measure temperature, the thermistor (a variable resistor), changes its resistance with respect to the temperature changes. If the temperature increases, the resistance decreases and vice versa. That is why this component is named NTC (Negative Temperature Coefficient). The Figure 3.7 illustrates The thermistor(NTC) operating mode of the DHT11.

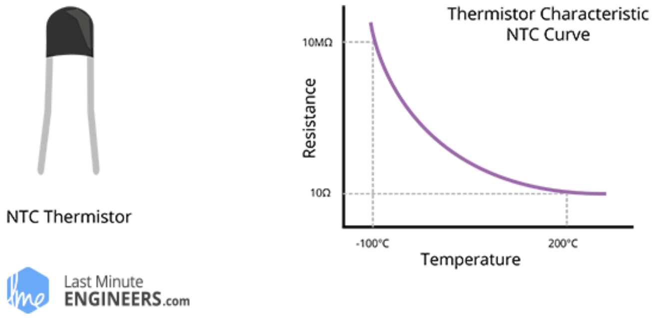


Figure 3.7: The thermistor(NTC) operating mode

Characteristics:

– Operating Voltage: 3 to 5V

– Max Operating Current : 2.5mA max

– Humidity Range : 20-80% / 5 %

– Temperature Range :0-50C / 2C %

– Sampling Rate : 1 Hz

– 4 pins with 0.1 spacing

– Body size : 15.5mm x 12mm x 5.5mm

– Ultra low cost

3.2.4 Soil moisture sensor

The soil moisture sensor consists of two probes which are used to measure the volumetric content of water [18]. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value. When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower. It has been widely used in agriculture, land irrigation and botanical gardening (figure 3.8).

The soil Moisture sensor FC-28 has four pins.

- VCC: For power
- A0: Analog output
- D0: Digital output
- GND: Ground

The Module also contains a potentiometer which will set the threshold value and then this threshold value will be compared by the LM393 comparator. The output LED will light up and down according to this threshold value.

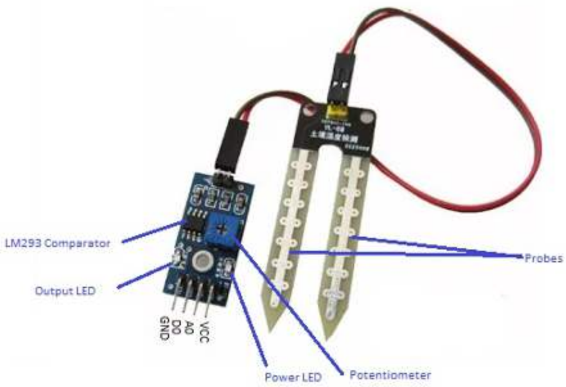


Figure 3.8: The Soil moisture sensor

Characteristics:

- Input Voltaget: 3.3 – 5V
- Output Voltage: 0 – 4.2V
- Input Current: 35mA
- Output Signal : Both Analog and Digital
- Working Temperature: 10C 30C
- 4 pins

3.2.5 DC motor

775 motor (figure 3.9) is a powerful device used in small applications like water pumps [19].

Specifications:

- DC Motor
- Nominal Voltage: 12V
- No Load RPM: 5600
- No Load Current: 0.5A
- Rated RPM: 4740
- Stall Current: 15A
- Stall Torque: 48 oz-in
- Shaft Type: Round

Dimensions:

- Size: 44.5D x 60.5L mm
- Weight: 295 g
- Shaft Diameter: 5 mm
- Shaft Length: 8.25 mm
- Rear shaft length: 8.5mm

Figure 3.9: The 755 motor

3.2.6 Relay

The relay module is an electrically operated switch that allows you to turn on or off a circuit using voltage and/or current much higher than a microcontroller could handle. There is no connection between the low voltage circuit operated by the microcontroller and the high power circuit. The relay protects each circuit from each other (figure 3.10). Each channel in the module has three connections named NC, COM, and NO. Depending on the input signal trigger mode, the jumper cap can be placed at high level effective mode which 'closes' the normally open (NO) switch at high level input and at low level effective mode which operates the same but at low level input [20].

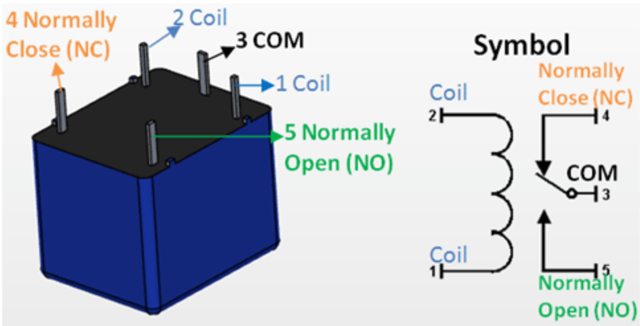


Figure 3.10: The Relay

Specifications

- Trigger Voltage (Voltage across coil) : 5V DC
- Trigger Current (Nominal current) : 70mA
- Maximum AC load current: 10A @ 250/125V AC
- Maximum DC load current: 10A @ 30/28V DC
- Compact 5-pin configuration with plastic mouldinRear shaft length: 8.5mm
- Operating time: 10msec Release time: 5msec
- Maximum switching: 300 operating/minute (mechanically)

3.2.7 RFM95 Lora module

The RFM95 transceivers (figure 3.11) feature the LoRa™ long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption. Using the patented LoRa™ modulation technique RFM95 can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness [21].

Figure 3.11: The RFM95

Specification:

– LoRa^{TM} Modem.

– 168 dB maximum link budget.

– +20 dBm - 100 mW constant RF output vs. V supply.

– +14 dBm high efficiency PA.

– Programmable bit rate up to 300 kbps.

– High sensitivity: down to -148 dBm.

– Bullet-proof front end: IIP3 = -12.5 dBm.

– Excellent blocking immunity.

– Low RX current of 10.3 mA, 200 mA register retention.

– Fully integrated synthesizer with a resolution of 61 Hz.

– FSK, GFSK, MSK, GMSK, LoRa^{TM} and OOK modulation.

– Built-in bit synchronizer for clock recovery.

– Preamble detection.

– 127 dB Dynamic Range RSSI.

– Automatic RF Sense and CAD with ultra-fast AFC.

– Packet engine up to 256 bytes with CRC.

– Built-in temperature sensor and low battery indicator.

– Module Size1616mm

3.2.8 Dragino Lora/GPS hat

The Dragino LoRa Shield (figure 3.12) is a long range transceiver on a Microcontroller shield form factor and based on Open source library.The LoRa Shield allows the user to send data and reach extremely long ranges at low data-rates.It provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption. The LoRa Shield based on SX1276/SX1278 targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and so on [22].

Figure 3.12: The Dragino Lora Shield

Specification:

– 168 dB maximum link budget.

– +20 dBm - 100 mW constant RF output vs.

– +14 dBm high efficiency PA.

- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

3.3 Software Environment

In this section we will be focusing on the Software side of the solution.

3.3.1 Visual studio

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio includes a code editor and supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. The most basic edition of Visual Studio, the Community edition, is available free of charge. We are specifically interested in the xamarin and Microsoft azure utilities.

3.3.1.1 Xamarin

Xamarin is a Microsoft-owned software company and an open source tool for development of mobile applications.xamarin tools use C code to write native apps(Android, IOS, and Windows) with native user interfaces and share code across multiple platforms [23].

Advantages of Xamarin

- The ability to create a native cross-platform application.
- 90 % of the code is shared.
- No need to know the proper development languages of each platform.
- Links with the Visual Studio IDE.

3.3.1.2 Microsoft Azure

Microsoft azure is a cloud computing service created by Microsoft for building, testing, deploying and managing applications and services through Microsoft-managed data centers. Here we state four key advantages of Microsoft's azure cloud [24] :

Security

Microsoft provides some of the most advanced security technology in the azure-cloud. Your Azure environment is safeguarded with tools like Threat Intelligence, Advanced Threat Analytics, Azure Information Protection, and Multi-Factor Authorization.

Privacy

With Azure, you own and control the collection, use, and distribution of your customer data. Microsoft provides in depth information on how they will handle your data.

High Availability and Scalability

Azure provides high availability and redundancy because of Microsoft's vast global footprint. With data centers located in all parts of the world, Azure can offer service level agreements guaranteeing 99.95 percent up time.

Cost Effectiveness

Azure offers a pay-as-you go payment plan that allows businesses to have better control over their IT budgets since they purchase only what they need.

xamarin and azure descend from the same family, with xamarin cross-platform features and azure cloud utilities plus the azure for student plan that gives those with verifiable school address full access to more than 25 Azure services for free plus $100 in credit.

3.3.2 SQL Server Management Studio 2017

SQL Server Management Studio is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor and administer instances of SQL Server and databases [25].

3.3.3 Star UML 2

StarUML is an open source software modeling tool characterized by it flexibility and extensibility that supports UML [26].

3.3.4 Arduino IDE

Arduino IDE is a lightweight, cross-platform application that introduces programming to novices. It has both an online editor and an on-premise application, for users to have the option whether they want to save their sketches on the cloud or locally on their own computers.With it , users can easily access contributed libraries and receive up-to-date support for the latest Arduino boards, so they can create sketches that are backed by the newest version of the IDE [27].

3.3.5 Device Explorer

The Device Explorer tool is a Windows-only graphical tool for testing your IoT hub, including adding a device, monitoring the device message and sending a message to the device [28].

Conclusion

This chapter summarizes our work environment choices after comparing the different available options. Next we will guide you through the implementation of our solution.

Chapter 4

Solution Implementation

Plan

Introduction 36

1 Gateway implementation 36

2 Nodes Implementation 38

3 Microsoft Azure cloud implementation 44

4 Mobile Application implementation 47

Conclusion 53

Introduction

After stating our choices in the previous chapter, this one mainly states the result of our choices after implementing our work.

4.1 Gateway implementation

4.1.1 Gateway Configuration

This section is a step by step guide to preparing the gateway configuration.

1- Connecting the Raspberry pi 2 to internet through Ethernet cable.

2- Installing GPIO access library : wiring pi.

3- Enabling SPI in the Raspberry pi 2 configurations.

4- Installing and building the LoRaWAN Gateway on our raspbian interface providing information (location, Ip address of the TTN).

5- Edit the 'main.cpp' file using the 'nano' command (figure 4.1)

- Set the spreading factor SF7.
- Define the TTN server IP .
- set the frequency to 868.1 MHZ (the iot frequency band used in Tunisia).
- set the Gateway location (latitude, longitude,altitude)

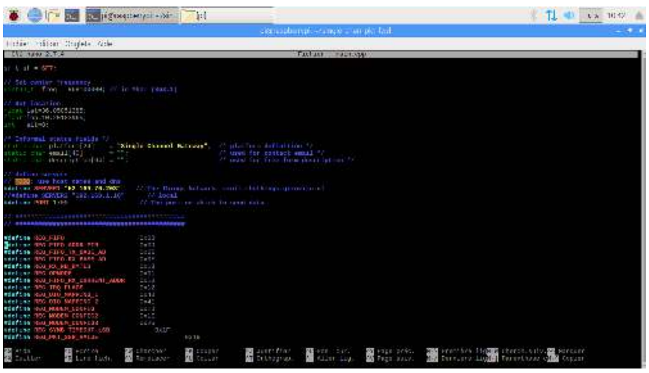


Figure 4.1: "main.cpp" Configuration

6- Fixing the Dragino Lora Shield with our Raspberry Pi 2 (figure 4.2).

7- Lunching the packet forward the executable of the LoRaWAN Gateway in order to start the gateway and get the unique ID (figure 4.3).

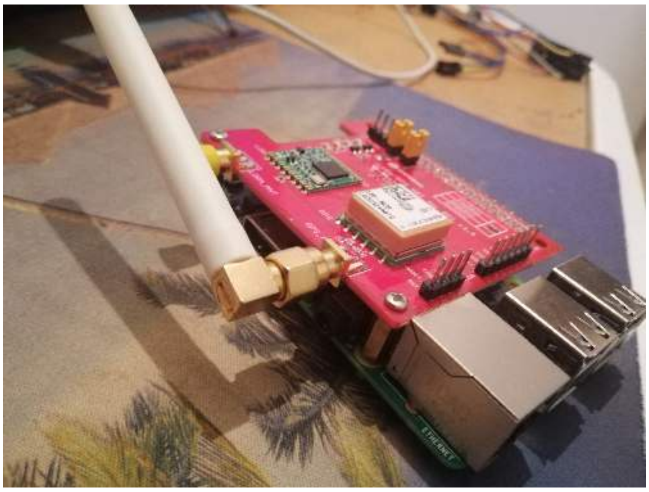


Figure 4.2: Dragino Lora Shield with Raspberry Pi 2

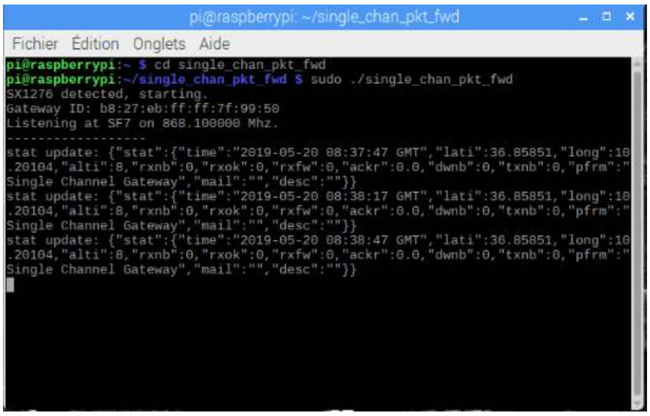


Figure 4.3: gateway Lunching

4.1.2 Gateway implementation to The Things Network

The Things Network is a global, crowd sourced, open, free and decentralized internet of things network. The network allows for things to connect to the internet using little power and little data. Opening use cases never possible before. It makes use of the LoRaWAN technology [29]. Now we are going to explain how to add our gateway into the TTN server by following these steps.

1- Creating an account on The Things Network.

2- Adding a gateway using the ID obtained in previous steps.

3- Providing our gateway info (name, latitude, longitude, altitude and a human description).

Finally, our gateway is up and running and ready to receive data from sensor nodes (figure 4.4).

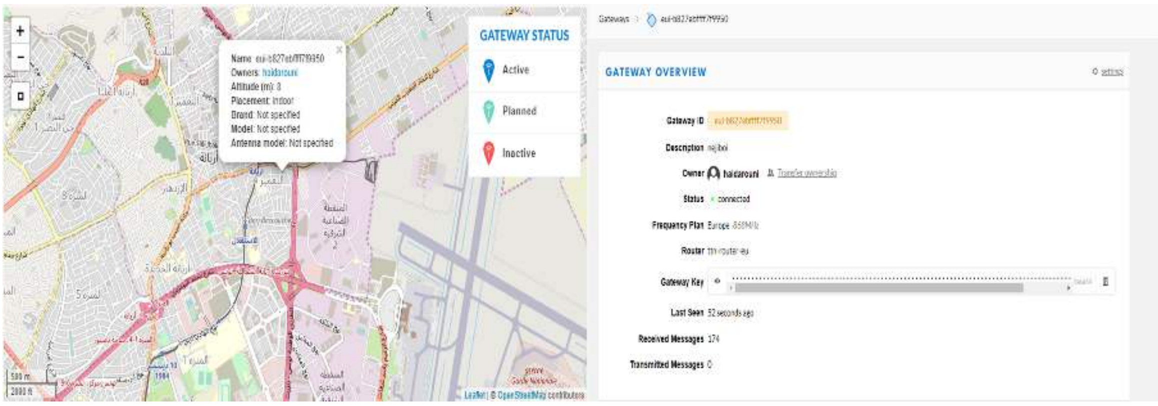


Figure 4.4: TTN Status

4.2 Nodes Implementation

We have 3 nodes: two sensors nodes and one sensor/actuator node. The sensors nodes are similar so we are gonna walk you step by step through their implementation

4.2.1 Sensor node implementation

The sensor node is the device responsible of calculating soil moisture and sending the data. For this process we need :

- The arduino nano.
- The RFM95 module
- The soil moisture sensor.
- A 4.5V battery.

The first step is the wiring as shown in the figures 4.5 and 4.6.

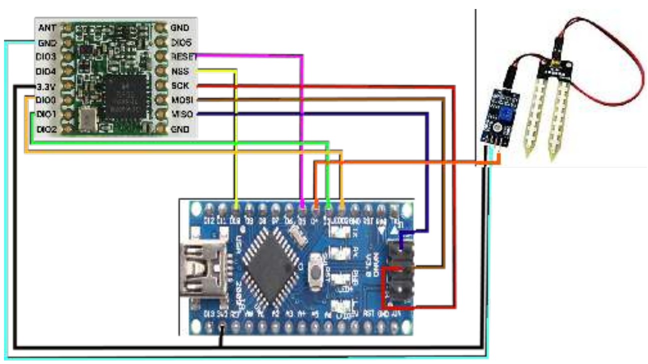


Figure 4.5: Sensor Node

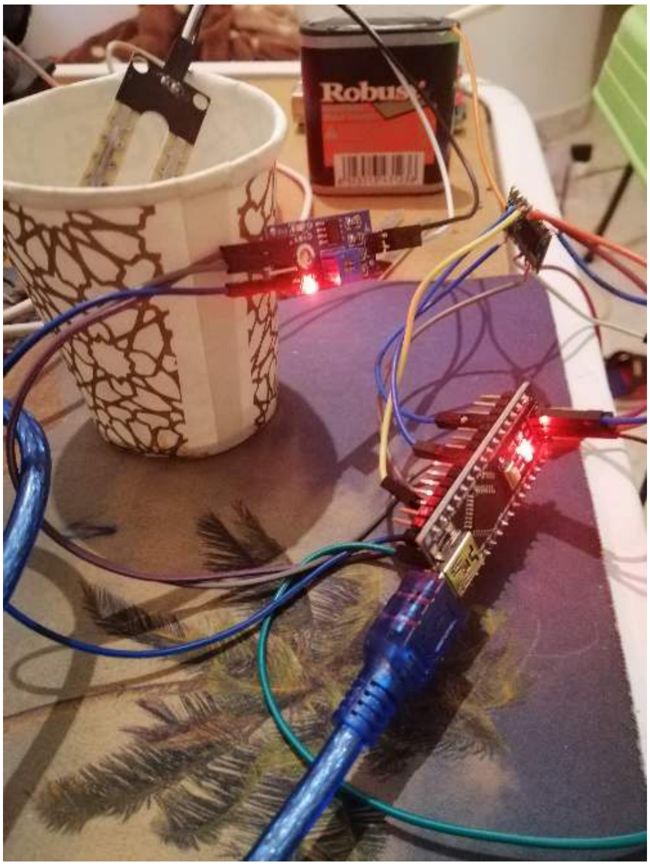


Figure 4.6: Soil Moistre Node

The second step is creating the application device in the things network platform, in order to do that you must first sign in. After getting to your personal work space where you can either register a gateway like we did previously or also you can register a device (figure 4.7). After that we move to the application interface where you can find all your registered applications, we will be adding a new one with a unique name and a human readable description (figure 4.8). After adding the device we get to the new application interface where we will be interested in the device address, network session key and the app session key to be used in an advanced step (figure 4.9).

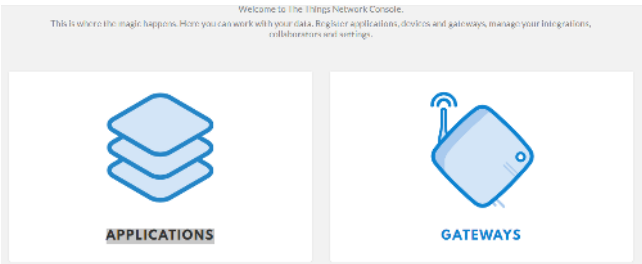


Figure 4.7: Personal interface

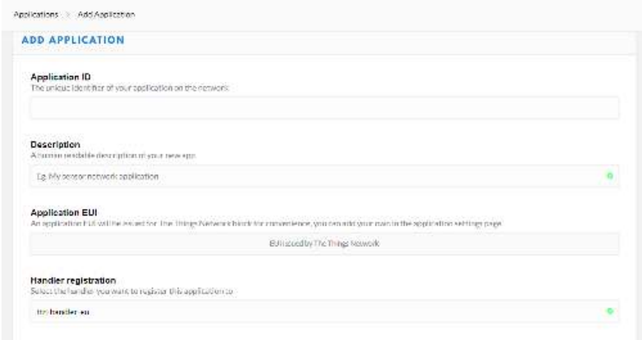


Figure 4.8: Device registration

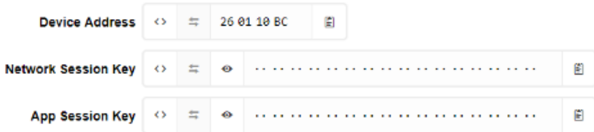
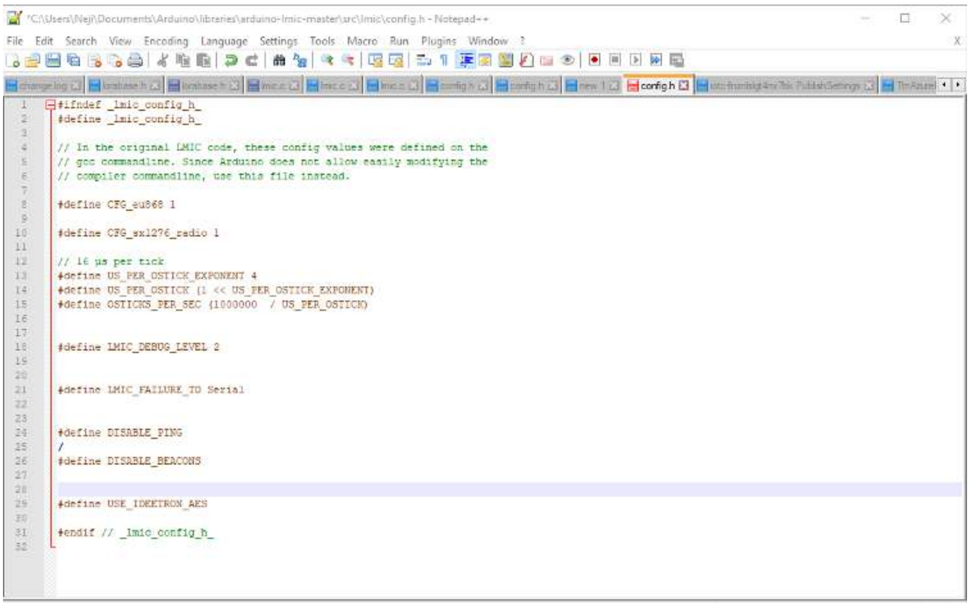


Figure 4.9: Access keys

The final step is the arduino IDE where we develop our code. Initializing the code by adding arduino-lmic library (LoraMAC-in-C library), it's slightly modified to run in the Arduino environment, allowing use of the SX1272, SX1276 transceivers and compatible modules (such as some HopeRF RFM9x modules) and provides a fairly complete LoRaWAN Class A and Class B implementation, supporting the EU-868 and US-915 bands. Then we are going to edit the config.h library file and set the frequency to 868.1 MHz because 868MHz is the iot frequency band used in Tunisia. The command used for that is "define CFG_eu868 1" and we set the transceivers to SX1276 because The HopeRF RFM95w module implement the SX1276 chip. The SX1276 chip proposes a +14dB transmission for CE area and +20dB for FCC area. The command used for that is "define CFG_sx1276_radio 1 " (figure4.10).

After saving the modifications of the config.h library file we are going to start writing the actual code by defining the lmic library adding the keys from the TTN application that I did before, and then we have to set the spreading factors as we know that Lora uses spread spectrum modulation. It is a variant of the chirp spread spectrum modulation. The spreading factor defines the bandwidth and the bit rate so that it is possible to send faster or further. The lower the spreading factor (SF), the faster the communication becomes. The higher the SF, the further it sends. The SF varies from 7 to 12. We set it at 7 except the data rate used for down link packets RX2 window we set it at 9 because TTN uses SF9 for its RX2 window (figure4.11).



```c
#ifndef _lmic_config_h_
#define _lmic_config_h_

// In the original LMIC code, these config values were defined on the
// gcc commandline. Since Arduino does not allow easily modifying the
// compiler commandline, use this file instead.

#define CFG_eu868 1

#define CFG_sx1276_radio 1

// 16 µs per tick
#define US_PER_OSTICK_EXPONENT 4
#define US_PER_OSTICK (1 << US_PER_OSTICK_EXPONENT)
#define OSTICKS_PER_SEC (1000000 / US_PER_OSTICK)


#define LMIC_DEBUG_LEVEL 2


#define LMIC_FAILURE_TO Serial


#define DISABLE_PING
/
#define DISABLE_BEACONS


#define USE_IDEETRON_AES

#endif // _lmic_config_h_
```

Figure 4.10: Config.h

```c
#if defined(CFG_eu868)

LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);      // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK,  DR_FSK),  BAND_MILLI);      // g2-band
/
LMIC_selectSubBand(1);
#endif


LMIC_setLinkCheckMode(0);

// TTN uses SF9 for its RX2 window.
LMIC.dn2Dr = DR_SF9;
```

Figure 4.11: Arduino Code

And finally we are going to add the soil moisture code to our code by defining two variables, one for the soil moisture sensor pin and the other for storing the output of the sensor and then adding it using the analogRead command to get the values from the output variable, next is mapping the output values to 0-100, because the moisture is measured in percentage. When we first took the readings from soil, the value was varying from 10 to 550. So, we needed to map these values to get the moisture in the interval [0-100] and into integer format, after that we convert these values to hexadecimal string format to be ready to send one data at a time in the shape of an array.

Now the data is well received in our TTN application interface, but in hexadecimal string format it is one MQTT protocol qualities (figure 4.12). TTN provides us with a payload format to decode and convert the hexadecimal string received to the original form (figure 4.13). And there is our data well received (figure 4.14).

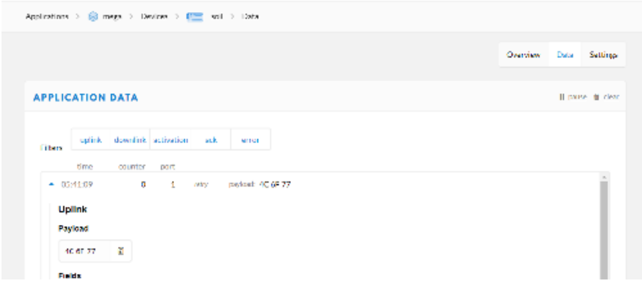


Figure 4.12: Exadecimal String Format

Figure 4.13: Payload Format

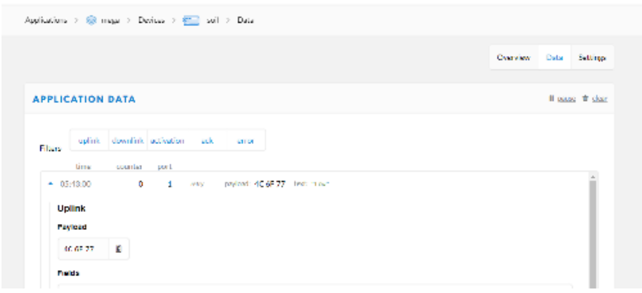


Figure 4.14: Original Format

4.2.2 Sensor/Actuator Node Implementation

The sensor/actuator node is actually a node that sends air humidity and temperature with a triggered actions, for the actuator we need :

- The arduino mega2560.
- The RFM95 module
- The dht11.
- The 755 motor.

– 5V relay.

The first step is the wiring as shown in the figures 4.15 and 4.16.

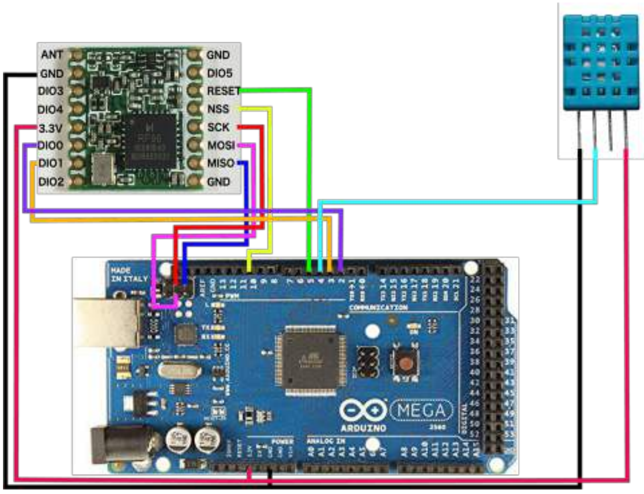


Figure 4.15: Actuator Node

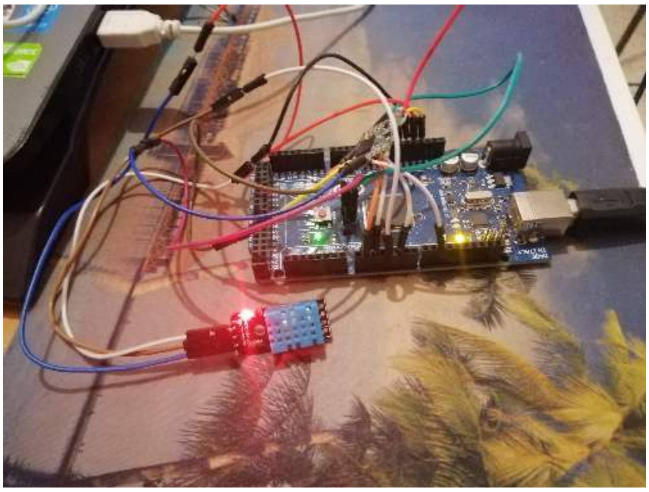


Figure 4.16: DHT11 Node

The next steps are similar to the sensor node except for the arduino IDE stage where we are going to make changes in the previous code. Beginning with deleting the soil moisture code to adding the DHT11 library that enables the reading of the temperature and humidity in a float format which as noted earlier must be converted to hexadecimal string format. In order to do that we made some modifications to enable "DHT11.read()" command to returns the values of the temperature and humidity in integer format. The water pump is ready and waiting for orders.

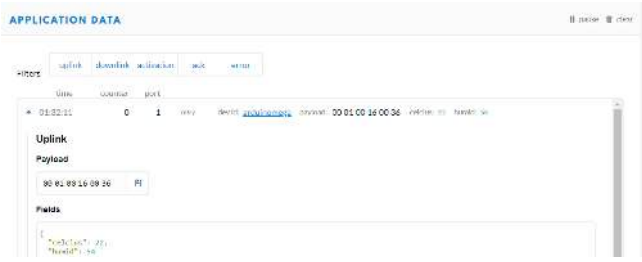


Figure 4.17: DHT11 Data

4.3 Microsoft Azure cloud implementation

This section will be divided into two subsections, the connection from TTN to the azure IoT hub then the deployment of the mobile app to the azure cloud.

4.3.1 TTN to Azure Iot Hub

Further processing of the telemetry on the The Things Network platform is not possible. Processing telemetry has to be done on your own IoT platform of your choice. In this case we choose the Azure IoT platform but first we need the secrets from the TTN platform to be able to create a secure connection between the TTN and your own platform. A secure connection between platforms is called a bridge. We will configure and deploy a bridge with the Azure IoT Hub. Starting from our application interface where we collect the application ID and Access Key. Next, we are going to create our Azure Iot Hub by following these steps, first we sign into our azure account, as noted in an earlier chapter the azure for student plan that gives those with verifiable school-address full access to more than 25 Azure services for free plus $100 in credit, then create our new Iot Hub selecting West Europe as its region. The bridge requires an Azure IoT Hub Shared access policy key name with Registry read, write and Device connect permissions. We use the iothubowner policy which has these permissions enabled by default. Always in the iothubowner interface where we will be interested in the Primary Key and the Connection String-Primary Key. Finally we edit the 'TtnAzureBridge.exe.config' file using all the secrets. Lunching our bridge (figure4.18) and verifying the incoming messages in the Azure IoT Hub using the Device Explorer software (figure4.19). We finally notice the device added automatically in our Iot Hub (figure4.20).

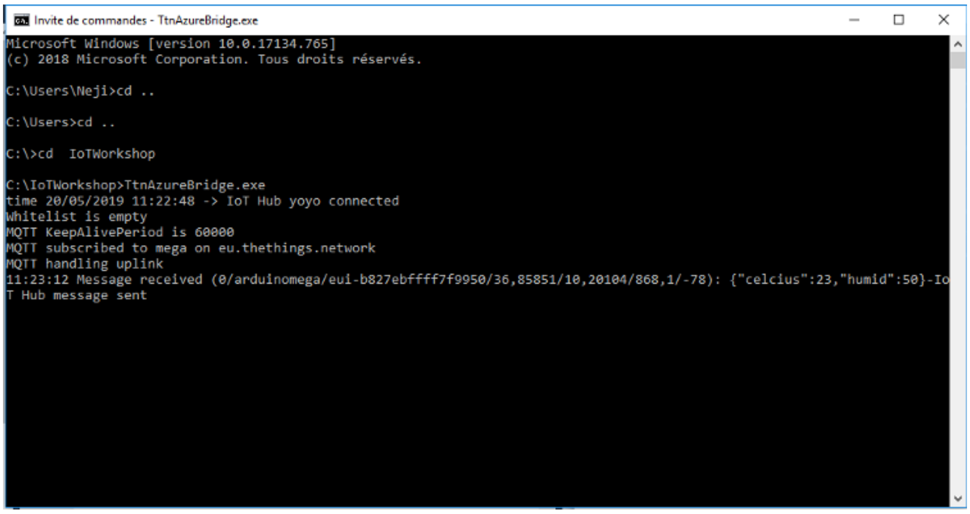


Figure 4.18: TTN Azure Bridge

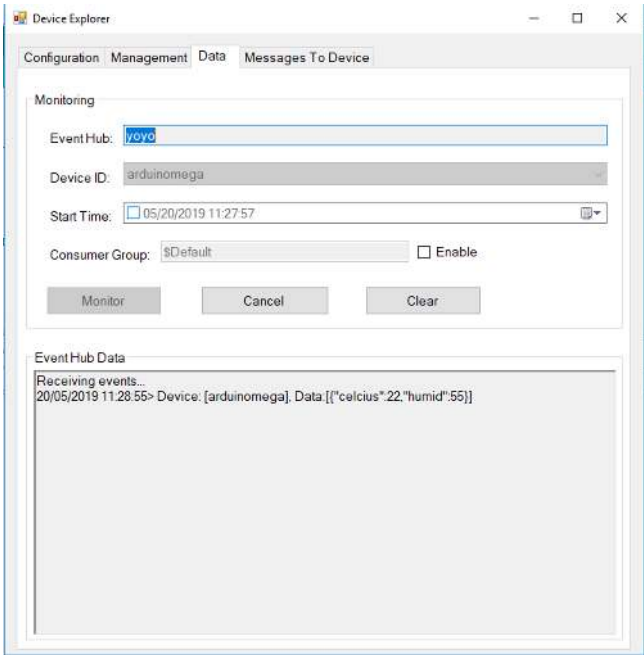


Figure 4.19: Device Explorer Monitor

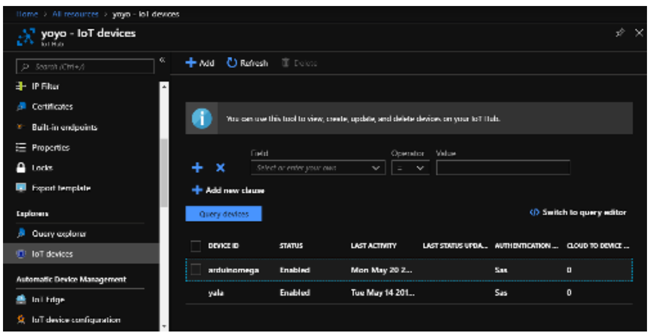


Figure 4.20: Azure Iot Hub Devices

4.3.2 Mobile application to Azure Cloud

In order to store an object we need a storage space, in IT world a storage room has many names and a database is the most commune, a database has two types SQL and NoSQL. The difference is that SQL is a relational database meaning it is a collection of tables, each with a schema that represents the fixed attributes and data types that the items in the table will have. Also provide functionality for reading, creating, updating and deleting data. Typically by means of Structured Query Language statements. On the other side NoSQL databases are Non-relational and can take a variety of forms. However, the critical difference between NoSQL and SQL databases is that RDBMS schemas rigidly define how all data inserted into the database must be typed and composed, whereas NoSQL databases can be schema agnostic, allowing unstructured and semi-structured data to be stored and manipulated. Back to azure who provides a variation of services related to database connections types.

For this part we will be using the SQL database to store the users contact data and the settings related. The first step is to create the database in the azure portal, it is obligatory to provide a server to make the connection possible to the database. We will create the server as well in the portal providing it with an admin login and an admin password to grant access with, one last step to be able to test the server with the use of the SQL management studio 2017 is to make an exception for your network IP in the server's firewall, figure 4.21 shows the server's test.

Next step is the API application. Which is a software intermediary that allows two applications to talk to each other. Since azure is a platform we need this API app to communicate between azure and the mobile application. Always using the portal we will create our own API then we head to configuration phase. This phase requires a connection string between the API app and the database, the database strings must contain the server's admin login and password. After deploying the strings in the database we move to phase two, the API back end, there is a variation of languages like javascript, nodejs and the C, and since we're already familiar with the C it is our choice for this phase, we are able to download the back end from the portal and we will be making our own modifications from table creating to our personal managing data methods using visual studio. For the final phase we need to publish the API back end to the azure portal with visual studio then we retrieve the API app URL to be used in the mobile application.

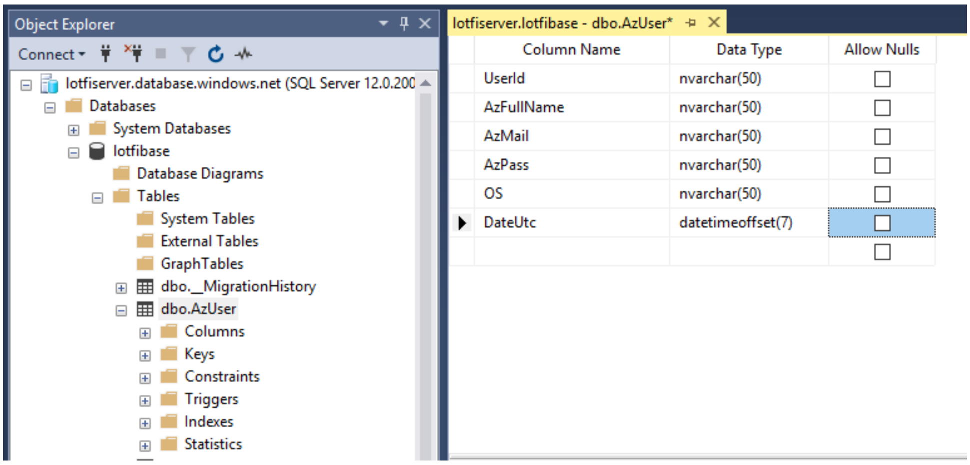


Figure 4.21: SQL management studio Test

4.4 Mobile Application implementation

The main role of the mobile application is to visualize different information such as the soil moisture level, the air humidity and temperature level. Through this application, the user is able to set the suitable irrigation options manually. In addition, the application notifies the user in case:

- Soil moisture level drops.
- Pump is activated.

4.4.1 Mobile Application frontend

The application consists of five view including a sign in page,sign up page and a tabbed page that contains the home page, settings page and a description page that goes with the name about page.

The sign in page is the first view (figure4.22).

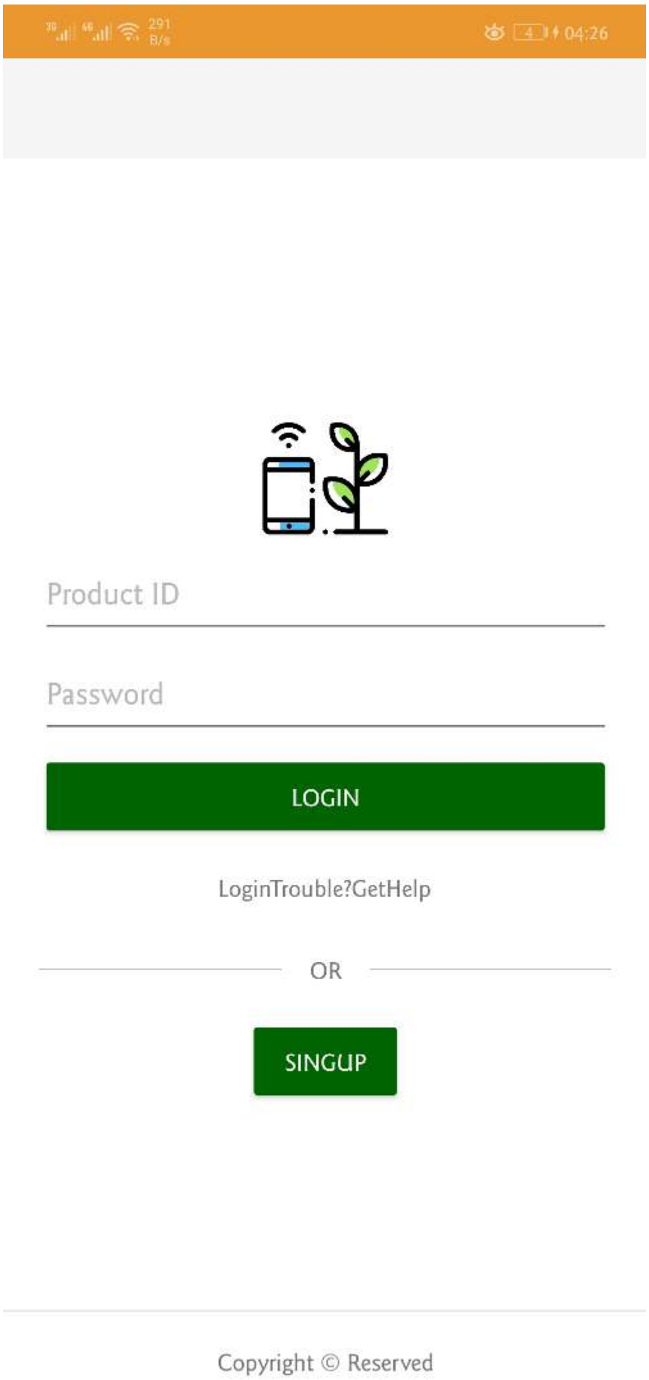


Figure 4.22: sign in View

The sign up page is the second view (figure4.23).

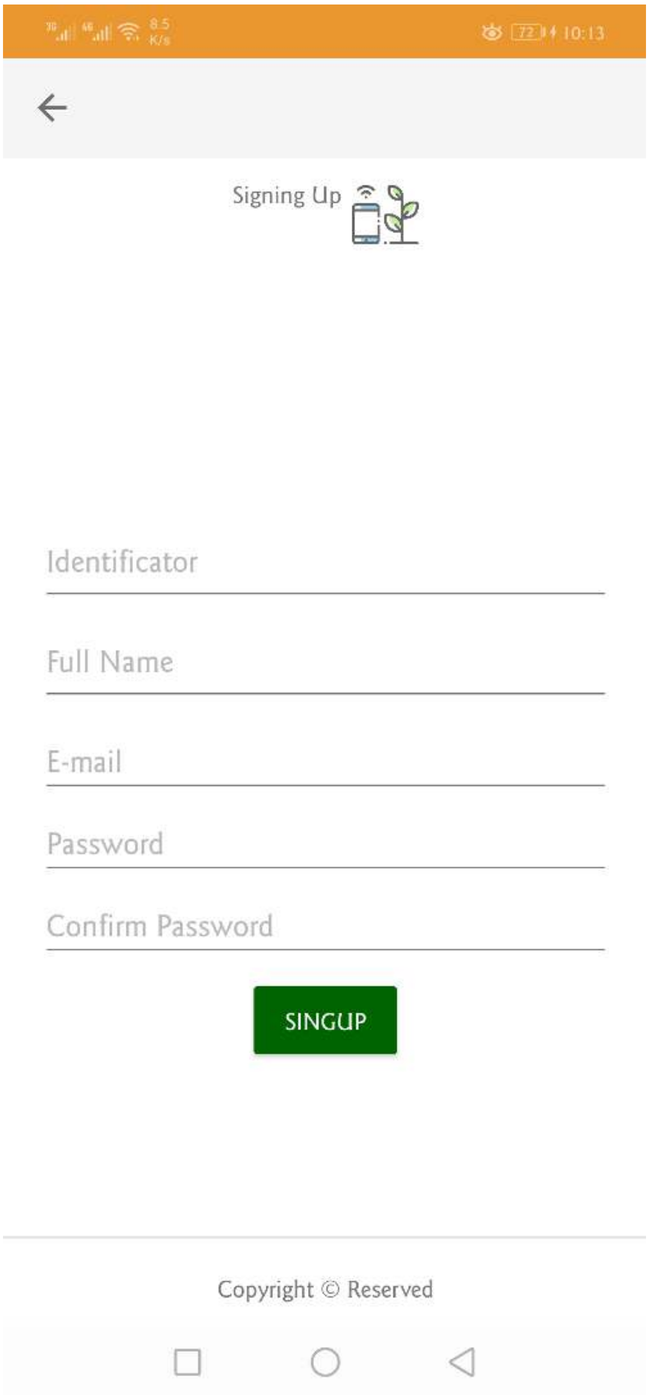


Figure 4.23: Sign Up View

The tabbed page is the third view (figure4.24). And it already contains the home page with the data as shown.

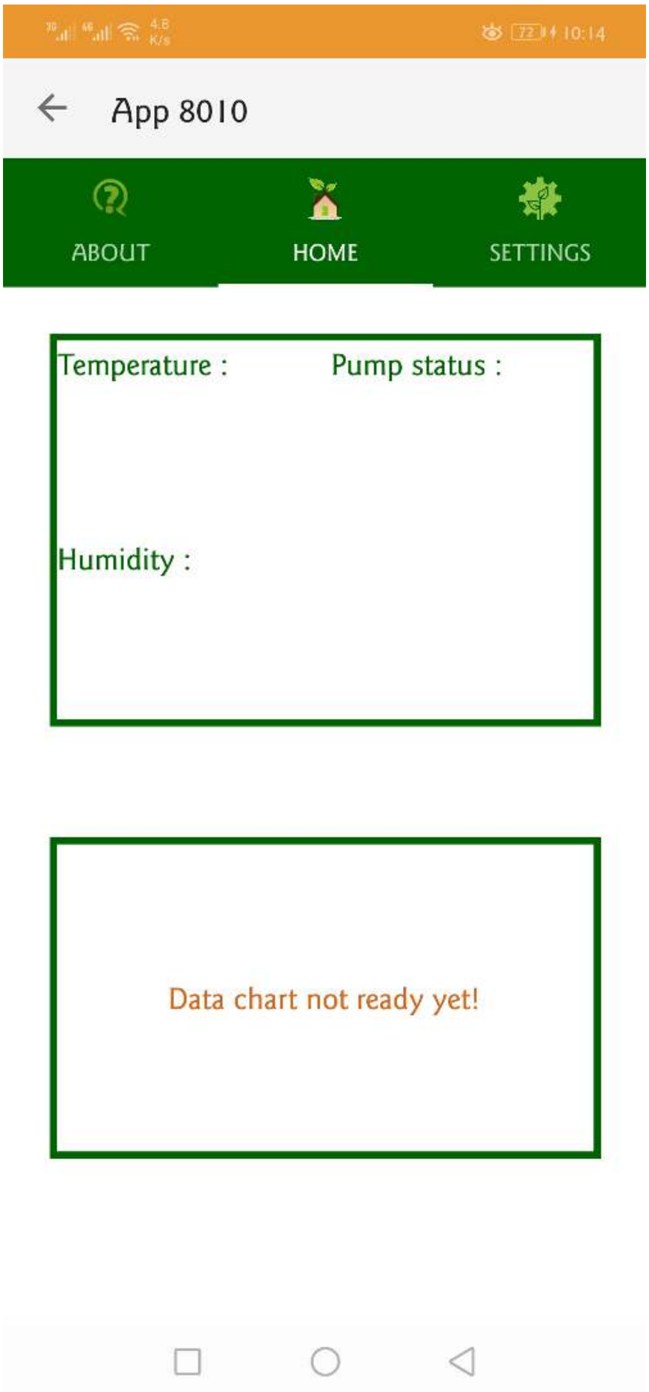


Figure 4.24: Home View

The settings page is the fourth view (figure4.25).

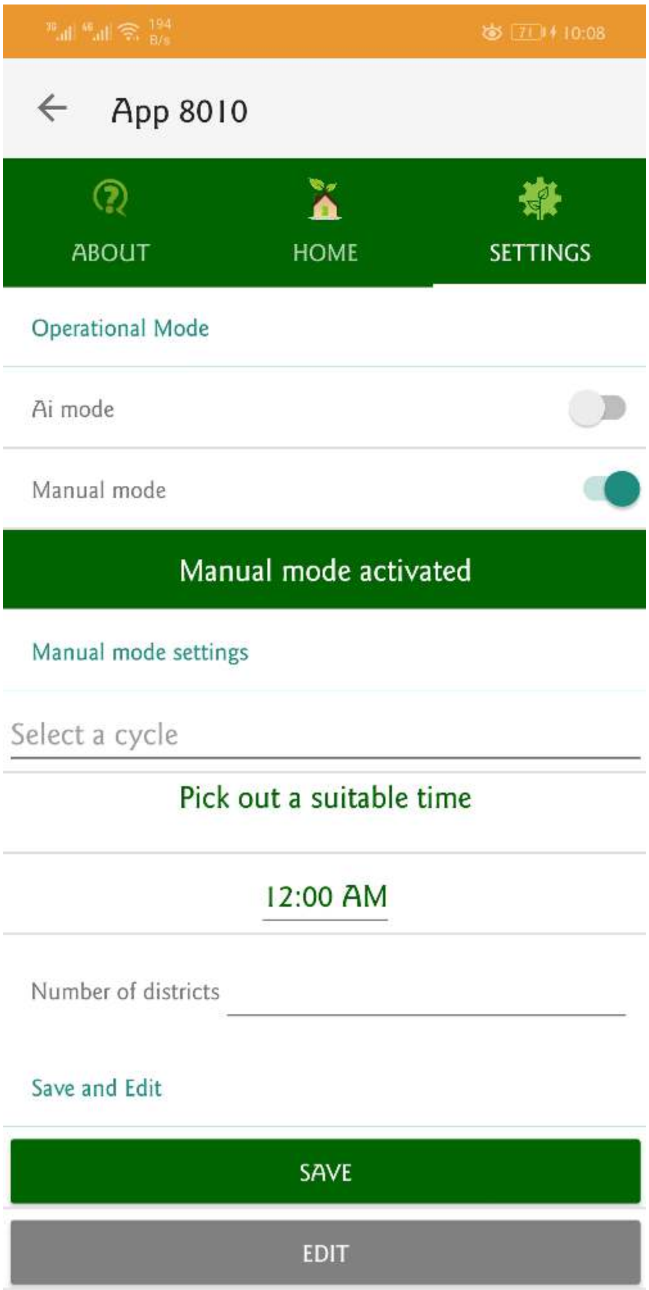


Figure 4.25: Settings View

Last but not least **The about page** is the fifth view (figure4.26).

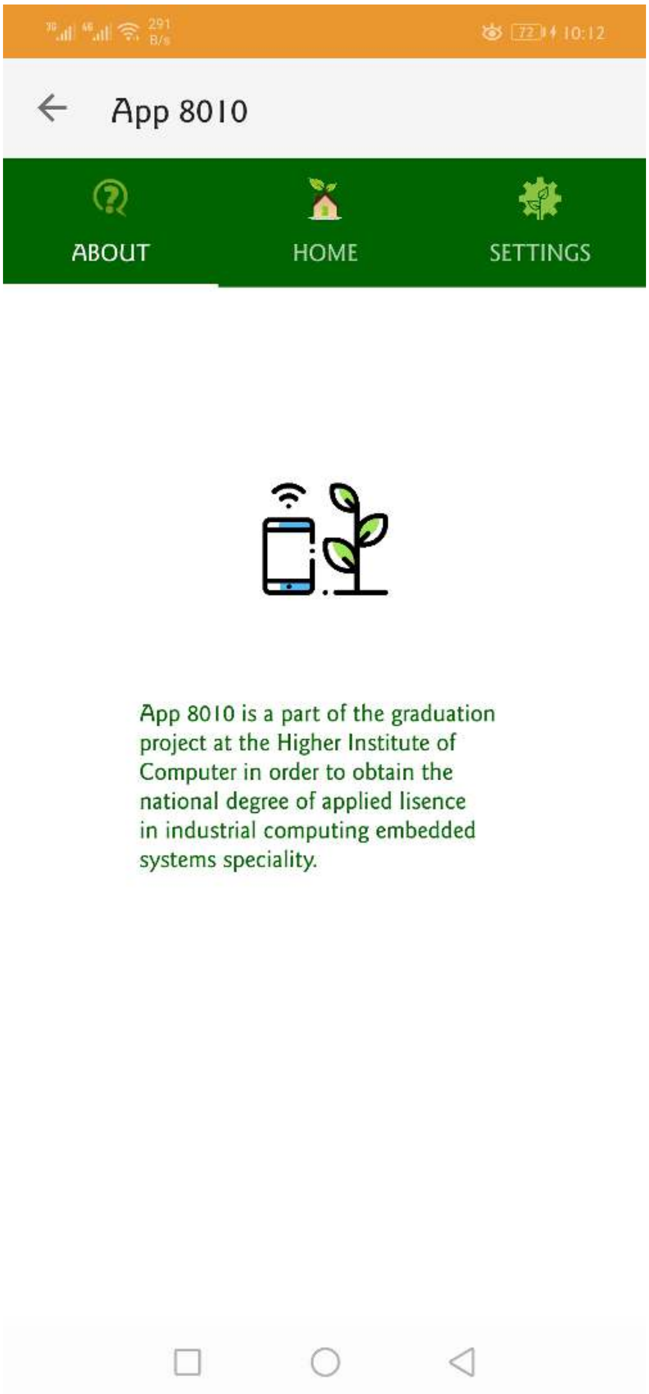


Figure 4.26: About View

4.4.2 Mobile Application backend

The backend of the application is a work in progress due to the complexity of the solution. For the meantime :

- Settings saved and retrieved from cloud .
- Data retrieved from cloud.

- Online/Offline synchronization.
- Authentication in progress.
- Notifications in progress.

Conclusion

This chapter guided you through the general aspects and results after implementing our solution presenting its advantages and encountered problems.

Conclusion and Perspectives

Our project is elaborated within the higher institute of computer science. Our goal is to make a wireless sensor network in order to achieve a precise irrigation. Our solutions objectives is to avoid water wastage, minimize runoffs, neglect the human errors and save energy, time, and valuable resources.

Briefly, in this report we walked you through the different aspects of the project from the objectives to the architecture exploring the work environment and right into the implementation.

The project has been beneficial for us as we have discovered some of the most complicated IoT principles learning the Lora, also we improved our embedded system awareness and learned new cross platform development languages and finally discovered the sea of clouds.

About the difficulties that we encountered, we had to wait for a long time for some of the materials arrival since we could not found it anywhere in Tunisia, also we made some bad choices regarding the mobile app at first, the complexity of solution made it hard to implement.

Did we fulfill our duties ? it is a question that can only be answered by you because your evaluation for our work is what distinguishes the success or failure of this solution.

We need, at this point, to go back to the idea of the project, we were highly motivated with high hopes of making an impact following the model of major talented IT leaders, sadly the impact is not made at the moment, however what is most important is that the hopes are still up, we will not stop giving.

In the future, this solution reveals also some perspectives like artificial intelligence and web mapping for more precise and developed features.

Bibliography

[1] Available on https://www.iotforall.com/what-is-iot-simple-explanation/ [Accessed 5-Mars-2019].

[2] Available on https://medium.com/datadriveninvestor/4-stages-of-iot-architecture-explained-in-simple-words-b2ea8b4f777f [Accessed 5-Mars-2019].

[3] Available on https://medium.com/sciforce/smart-farming-or-the-future-of-agriculture-359f0089df69 [Accessed 5-Mars-2019].

[4] Available on http://www.fao.org/nr/water/aquastat/didyouknow/index3.stm [Accessed 7-Mars-2019].

[5] Available on https://plantix.net [Accessed 10-Mars-2019].

[6] Available on http://www.phyt-eau.com/ [Accessed 10-Mars-2019].

[7] Available on http://www.chercheinfo.com/view-listing.php?id=4892 [Accessed 10-Mars-2019].

[8] Available on https://www.rachio.com/rachio-3/ [Accessed 10-Mars-2019].

[9] Available on https://bhyve.orbitonline.com/ [Accessed 7-Mars-2019].

[10] Available on https://www.sigfox.com/en [Accessed 25-April-2019].

[11] Available on https://blog.integral-system.fr/quest-ce-que-nb-iot/ [Accessed 25-April-2019].

[12] Available on https://lora-alliance.org/about-lorawan [Accessed 25-April-2019].

[13] Available on https://www.polymorph.co.za/iot-connectivity-comparison-gsm-vs-lora-vs-sigfox-vs-nb-iot/?fbclid=IwAR2zUZZC85I-vCX8uMe6gh3R2MCE9bEZf0AmL66Kzmn-PSSEQvVwMzDX6aE [Accessed 19-May-2019].

[14] Available on https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ [Accessed 12-May-2019].

[15] Available on https://store.arduino.cc/mega-2560-r3 [Accessed 12-May-2019].

[16] Available on https://store.arduino.cc/arduino-nano [Accessed 12-May-2019].

[17] Available on https://www.robot-r-us.com/vmchk/sensor-temp/humid/dht11-temperature-and-humidity-sensor.html [Accessed 12-May-2019].

[18] Available on http://www.circuitstoday.com/arduino-soil-moisture-sensor [Accessed 12-May-2019].

[19] Available on https://www.robotshop.com/en/rs-755-12v-4791oz-in-5600rpm-brushed-dc-motor.html?fbclid=IwAR05quNp5eYaVz6AzgLedyqU4p95YLNKZ0q8SxCDIUorCQUHRmXvCgX8g [Accessed 12-May-2019].

[20] Available on https://components101.com/5v-relay-pinout-working-datasheet [Accessed 19-May-2019].

[21] Available on https://www.seeedstudio.com/RFM95-Ultra-long-Range-Transceiver-Module-LoRa-Module-support-868M-frequency-p-2807.html [Accessed 12-May-2019].

[22] Available on http://www.dragino.com/products/module/item/102-lora-shield.html [Accessed 12-May-2019].

[23] Available on https://visualstudio.microsoft.com/fr/xamarin/ [Accessed 19-May-2019].

[24] Available on https://azure.microsoft.com/fr-fr/overview/what-is-azure/ [Accessed 19-May-2019].

[25] Available on https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017 [Accessed 21-May-2019].

[26] Available on http://staruml.io/ [Accessed 19-May-2019].

[27] Available on https://www.arduino.cc/en/Main/Software [Accessed 19-May-2019].

[28] Available on https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer [Accessed 21-May-2019].

[29] Available on https://www.thethingsnetwork.org/docs/gateways/ [Accessed 19-May-2019].

Annexes

Annexe1

Arduino Mega 2560 Pin Diagram.

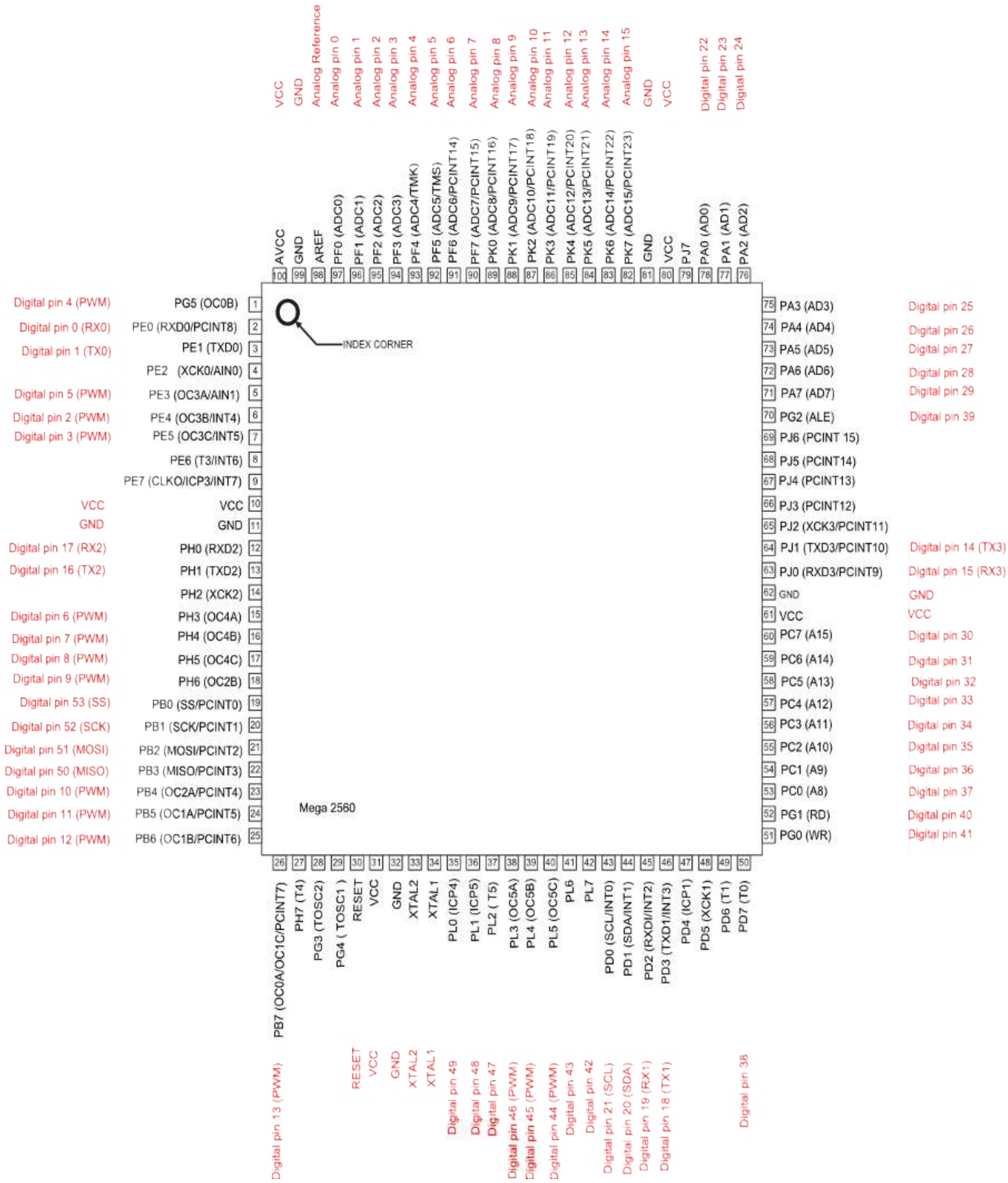


Figure 4.27: Arduino Mega 2560 Pin Mapping

Annexe2

Arduino Mega Nano Pin Diagram.

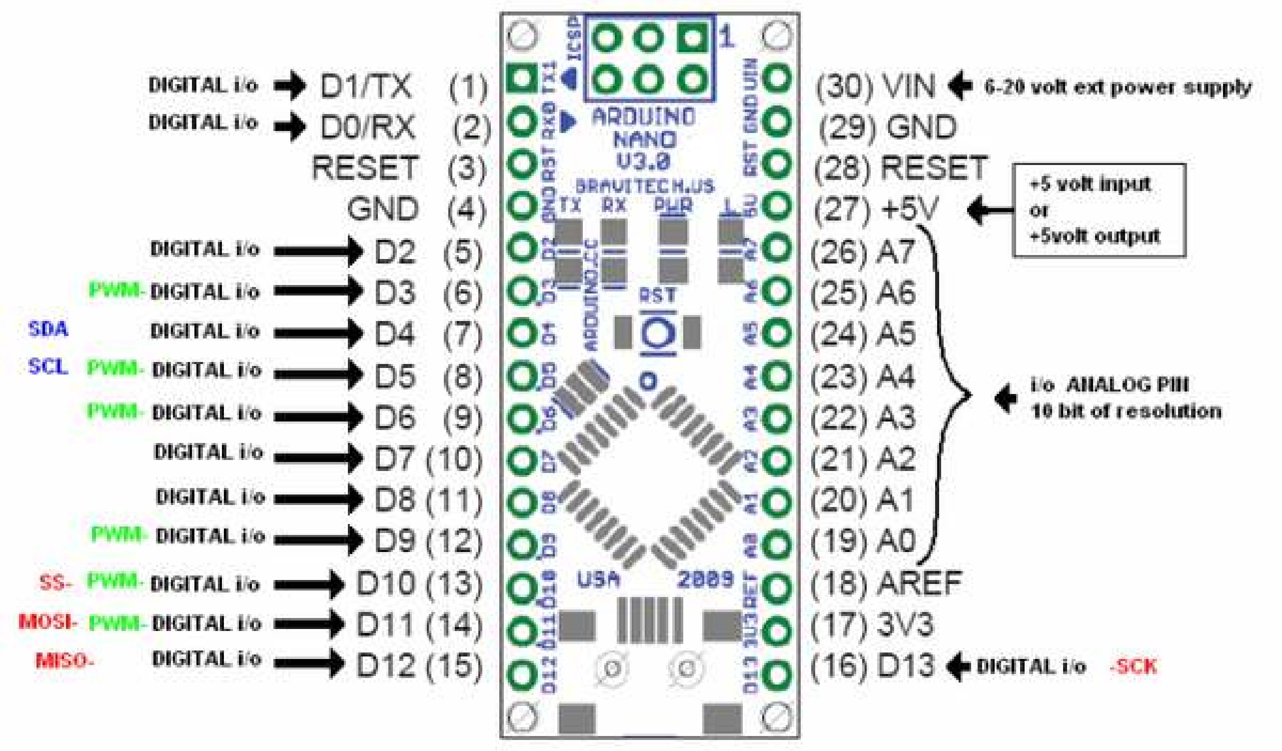


Figure 4.28: Arduino Nano Pin Mapping

Annexe3

RFM 95 Pin Diagram.

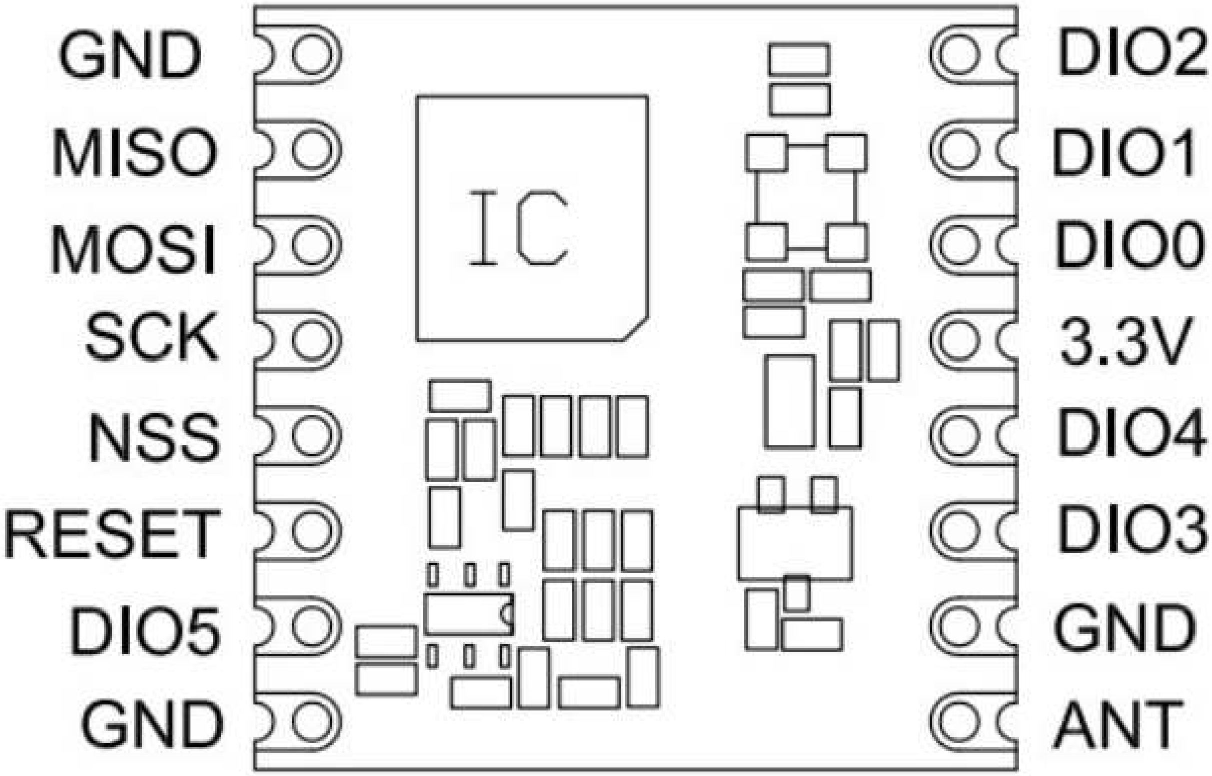


Figure 4.29: RFM 95 Pin Mapping

Résumé

Dans le cadre du projet de fin d'études à l'Institut Supérieur d'informatique et afin d'obtenir le diplôme national de licence appliquée en informatique industrielle spécialité systèmes embarqués, nous avons réalisé un systeme intelligent pour une irrigation précise et automatique à base de la technologie LoRa.

Mots clés : Iot,Smart farm, LoRa, Precise irrigation, Azure

Abstract

As part of the graduation project at the Higher Institute of Computer and in order to obtain the national degree of applied license in industrial computing specialty embedded systems, we realized a smart system for precise and automatic irrigation based on Lora technology.

Keywords : Iot, Smart farm, LoRa, Precise irrigation, Azure

2 نهج أبو الريحان الباروني 2080 أريانة الهاتف: 71 706 164 الفاكس: 71 706 698 البريد الالكتروني: isi@isim.rnu.tn
2 Abou Raihane Bayrouni 2080 l'Ariana **Tél**: 71 706 164 **Fax**: 71 706 698 **Email**: isi@isim.rnu.tn