TUNISIAN REPUBLIC

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF TUNIS EL MANAR

HIGHER INSTITUE OF COMPUTER SCIENCE

FINAL STUDIES PROJECT REPORT

Presented in order to obtain

National Diploma of Applied License in Science and Technology

Mention : Industrial Data Processing

Specialty: Embedded Systems

By

Abir Ben Soltane & Riadh Bougaalech

Design And Implementation Of A Wireless Sensor Network For The Remote Monitoring Of Buildings

Professional Supervisor:	**Mr.Aymen Boussetta**	Engineer R&D
Academic supervisor:	**Mrs.Zouhour Ben Azouz**	Assistant Professor
Academic supervisor:	**Mrs.Mariem Feki Fakhfakh**	Assistant Professor

Project proposed and elaborated within Intercoom Technologies

University Year: 2017 - 2018

TUNISIAN REPUBLIC

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF TUNIS EL MANAR

HIGHER INSTITUE OF COMPUTER SCIENCE

FINAL STUDIES PROJECT REPORT

Presented in order to obtain

National Diploma of Applied License in Science and Technology

Mention : Industrial Data Processing

Specialty: Embedded Systems

By

Abir Ben Soltane & Riadh Bougaalech

Design And Implementation Of A Wireless Sensor Network For The Remote Monitoring Of Buildings

Professional Supervisor:	**Mr.Aymen Boussetta**	Engineer R&D
Academic supervisor:	**Mrs.Zouhour Ben Azouz**	Assistant Professor
Academic supervisor:	**Mrs.Mariem Feki Fakhfakh**	Assistant Professor

Project proposed and elaborated within Intercoom Technologies

University Year: 2017 - 2018

Professional Supervisor

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

signature and stamp

Academic supervisor

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

signature

Academic supervisor

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

signature

Dedications

It is with deep gratitude and deep respect that I dedicate this modest work to my beloved parents, for their love, care and dedication. They have always believed in me, supported and guided me.

I also want to dedicate this project to my grandfather's soul, for all his advice, affection, encouragements and all the beautiful memories we shared together. To my sisters, my brother and all my friends who always gave me a hand whenever I needed it and to all my teachers during my educational journey.

To anyone who has engraved my life by a word that has oriented me to the good way.

Abir Ben Soltane

To my Dear Parents: The pillar of all joy and achievement in our lives. By this modest work, we hope to have honored: your non refundable sacrifices; your longing for my success your heartfelt prayers.To my Sister & my Brother & my grand-parents for their love, support and patience through the years.

To my Family & Friends thanks for your support

Riadh Bougaalech

ACKNOWLEDGEMENTS

First of all ,we would like to express our immense gratitude to our Professors Mrs.Zouhour BEN AZOUZ & Mrs.Mariem FEKI. It was a real privilege and honor for us to share their exceptional knowledge but also of their extraordinary human qualities . They have taught us more than we could ever give them credit for here .

We also would like to thank our supervisor at INTERCOM Technologies; Mr. Aymen BOUSSETTA for all the support and guidance he gave us. A special thanks goes to Mr. Aymen REZGUI for his help and constructive suggestions. Our recognition also goes to Mr. Tarek BOUSSETTA for giving us the opportunity to develop our skills under the roof of INTERCOM Technologies. We would like to express our special appreciation to the entire staff of INTERCOM Technologies for their helpful and friendly nature.

Table of contents

List of Figures

List of Tables

Abreviations list

- **ADR** = **A**daptive **D**ata **R**ate
- **ADR** = **A**daptive **D**ata **R**ate
- **AES** = **A**dvanced **E**ncryption **S**tandard
- **COAP** = **CO**nstrained **A**pplication **P**rotocol
- **CSS** = **C**ascading **S**tyle **S**heet
- **GPRS** = **G**lobal **P**acket for **R**adio **S**ervice
- **GSM** = **G**lobal **S**ystem for **M**obile
- **HTML** = **H**yper **T**ext **M**arkup Language
- **HTTP** = **H**yper **T**ext **T**ransfer Protocol
- **HTTPS** = **H**yper **T**ext **T**ransfer Protocol Secure
- **IDE** = **I**ntegrated **D**evelopment **E**nvironment
- **IEEE** = **I**nstitute of **E**lectrical **E**lectronics **E**nginee
- **IOT** = **I**nternet **O**f **T**hings
- **J2EE** = **J**ava 2 Platform **E**nterprise **E**dition
- **JSON** = **J**avaScript **S**cript **O**bject Notation
- **LPWAN** = **L**ow **P**ower **W**ide **A**rea **N**etwork
- **LoRa** = **L**ong **R**ang
- **MQTT** = **M**essage **Q**ueuing **T**elemetry **T**ransport
- **PHP** = **P**rocessor **H**yperext **P**rocessor
- **SQL** = **S**tructured **Q**uery **L**anguage
- **SSL** = **S**ocket **S**ecure **L**ayer
- **TLS** = **T**ransport **L**ayer **P**rotocol

- **UDP** = **U**ser **D**atagram **P**rotocol
- **UMTS** = **U**niversal **M**obile **T**elecommunications **S**ystem

Introduction

With the increasing popularity of electronic devices , it will be unavoidable to connect these objects to the net in order to enable them to communicate and exchange data. Therefore the internet of things was created to handle this issue by sensing or controlling devices remotely across existing network infrastructures .

Statistics [1] showed that connected devices have increased from 15 billions objects in 2016 to 23 billion in 2017 and they are expected to reach 75 billion objects by 2025. According to "Computers in Human Behavior" [2], the global market value of IOT for 2020 is estimated to $7.1 trillions. This is explained by the usefulness of IOT in different fields such as healthcare , agriculture ,energetic, transportation and building automation.

For instance, people need to control their buildings from anywhere and reduce their energy consumption. In this context, our project aims to design and implement a wireless connected sensor network for the remote monitoring of buildings. It is mainly based on a set of sensors and actuators nodes that transmit information to a server through a gateway. The server communicates to the user the transmitted information through a mobile application.

This report encloses five chapters. The first one presents the context and the specification of our project. It also reviews the main communication protocols used in IOT. In the second chapter, we globally describe our solution and model its behaviour using UML. The third chapter details the sensor nodes and the gateway design of our network. As for the fourth chapter, it describes the server and the mobile communication as well as the communication with the gateway. The last chapter resumes the project implementation. This report will be ended by a conclusion that summarizes our work and proposes possible improvements to our application .

Chapter 1

Project context

Plan

Introduction

This chapter defines the general context of our project. During the first step, we present the host organization. In the next one, we state the problematic and review existing solutions. We then specify the functional and non-functional requirements of our system and model its behaviour using UML.

1.1 Host organization

This project has been carried out within Intercom Technologies, which is an international company created in 2005 in France. It has launched its first agency in Tunisia in 2010 (in the technological center EL GHAZALA). Intercom Technologies is specialized in the expertise and management of technological projects. It has the vocation of supporting its customers in various sectors such as:

- mobile telephony:GSM,GPRS,UMTS,EDGE,WIFI,WIMAX
- Network :ADSL,VOIP, ATM,SDH
- Functional domains : supervision , synchronisation.
- RIT domains and network: Java,JEE, Databases SQL/Oracle
- Embedded systems : embedded electronics and internet of things

The figure 1.1 presents Intecom's logo.

Figure 1.1: Intercom's Logo

1.2 Problematic

Nowadays, modern buildings are equipped with different technologies and electronic devices. Gadgets and machines are powered by electric power or fuel. Thus many accidents may happen such us gas leak, water leak, fire...

Those potential risks make owners in constant fear. Moreover, planning a trip and leaving the building is a troublesome idea. As burglary and robbery is a potential threat. Hence, remote building monitoring and control is a necessity. those systems can prevent accident and save lives by detecting and controlling potential risks. In fact,checking and commanding buildings from anywhere in the world makes the owners live easier, secured and more comfortable.

1.3 Existing Solutions

Existing Solutions are based on different Wireless protocols.They have different Topologies.

1.3.1 Wireless Network Topologies

Point to point[3]

This topology provides dedicated link between two nodes. Connection can be permanent or switched, which means that direct link is established on demand.This topology is not useful in systems with high number of nodes.

Star

Nodes in this topology are divided into leaf nodes and central node.Each leaf node is connected to the central node, which is switching messages between them.

Mesh

In this topology, each node is able to act as a router and relay data and therefore all nodes together cooperate in the distribution of data, which can be distributed by flooding or routing techniques.

Tree

Tree topology contains leaf nodes and router nodes. No nodes can be connected to a leaf node, but on the other hand, both types of nodes can be connected to a router node.

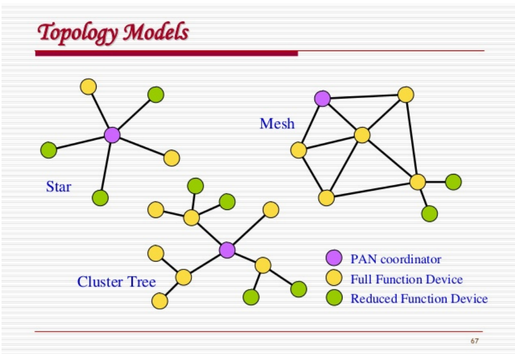

Figure 1.2: Topology types

To use a wireless sensor network , we need to distinguish between IOT protocols the most suitable protocol for our application :

1.3.2 IOT Protocols

IOT protocols is divided into web protocols and embedded protocols:

1.3.2.1 Web iot protocols

When we talk about IOT protocols, we talk about two large architectures : request/response architecture and publish/subscribe architecture . In the remainder of this section, we present the most used protocols.

COAP(Constrained Application Protocol):

COAP is an optimized transfer protocol based on request/response architecture running on the User Datagram Protocol(UDP) protocol. Its concept consists of generating an URL from a server resources in order to give the client access to them using REST methods such as GET,POST,PUT and DELETE.
UDP (User Datagram Protocol): UDP is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet . [4]

MQTT

MQTT is a messaging protocol based on Publish/Subscribe architecture. It is characterized by its simplicity and lightness. MQTT is specially designed for remote locations where a small code is required or the network bandwidth is limited. It is client/server model where the gateway and the servers are clients connected to a server named Broker through TCP protocol . Both mqtt and COAP are the most leading protocols for the internet of things but MQTT is more mature and stable than coap.

1.3.2.2 Embedded IOT protocols :

we have to distinguish between wireless sensors network the suitable embedded iot protocol for our project:

Bluetooth :

Bluetooth is a standardized protocol for sending and receiving data via a 2.4GHz wireless link. It's a secure protocol, and it's perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices [5].

Zigbee:

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection [6].

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 transfert can't depass 12 bytes. It is bidirectional ,for the downlink side sigfox node can only receive 4 messages per day .

Lorawan and Lora :

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[3]. 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), smart city industrial applications[7] .

1.3.3 Comparison

. The main WPAN technology is Bluetooth and Zigbee. Bluetooth has the lowest range (maximum 30m) and zigbee has a maximum range of 100m . Zigbee seems to be a suitable protocol for our application but later we have noticed that sigfox and lorawan , Low-Power Wide-Area Network(LPWAN) are better than zigbee because they consume less-power(the battery may last 10 years) and they have a higher range. Despite transfering low speed data, they remain the best protocols to create a private wireless sensor network, because the amount of sensor transfer data is extremely small. After this study the choice of our protocol has been delimited between lorawan and sigfox . Sigfox has a better range than lorawan (10 km in urban and 40 km in rural), while lorawan has 5 km in urban and 20 km in rural. However, lorawan has a high data rate 50 kb/s, wheras sigfox has a rate of 60 bp/s. Another advantage of lorawan

,it has a maximum payload length of 243 bytes which is much higher than sigfox 12 bytes. Moreover, sigfox has a limited number of allowed messages per day while lorawan is unlimited [8].

This comparative study has shown that LoraWan is the suitable embedded protocol for our application.

1.4 Lorawan Architecture

LoRaWAN network architecture is typically laid out in a star-of-stars topology in which gateways is a transparent bridge relaying messages between end devices and a central network server in the back-end. Gateways are connected to the network server via standard IP connections while end devices use single-hop wireless communication to one or many gateways [9].

End devices:

They are wireless sensors, actuators or embedded applications that sends or receive information in order to process it.

Gateway :

Objects are associated with a specific gateway. Any data transmitted by the object is sent to all gateways and each gateway transmits the signal received to a cloud server. Typically, gateways and servers are connected via a Backhaul (intermediate links between the core network and the small subnetworks at the "edge" such us cellular, Wi-Fi, Ethernet or satellite).

Network Server:

connects to multiple gateways over a secure TCP / IP connection,then eliminates duplicate messages,checks security, designates the gateway that must respond to the end-node message and manages the data rates of the end-node through an Adaptive Data Rate (ADR) schema to optimize network capacity,extend the battery life of the node.

LoraWAN classes:[10]

Lora objects classes is divided into 3 classes ,each class has its characteristics :

Class A :

Class A is characterized by :

- The frame in general is divided into uplink transmission and downlink transmission. Uplink is consists of 1 slot followed by 2 downlink slots

- Uplink slot is scheduled by End device itself based on its need. It is decided on random basis similar to ALOHA protocol
- It is the lowest power LoRa end device.

Class B :

Class B is characterized by :

- This class of end devices use extra receive windows during downlink period in addition to two time slots specified in class-A
- Class B devices will get extra receive windows at specified duration.
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- The duration is specified by the gateway using beacon frame
- Hence this way LoRa system indicates to the server when end device can listen

Class C:

Class C is characterized by :

- This class of end devices can listen all the time except in transmit mode. Hence it is ideal for applications requiring more downlink transmissions.
- It has lowest latency among all the LoRa class end devices for data communication between server and end device

But given the unavailability of LoRaWAN network in Tunisia, we are going to develop a secure protocol based on the LoRa link layer protocol which fulfills our application requirements. This protocol will be described later in the next chapters.

1.5 Project Specification

This project aims to design and implement a system which allows the user to monitor remotely his building. The system collects information such as temperature and humidity. It notifies the user if a gas leak, a fire or an intruder are detected. The user is also notified if a window is opened outside working hours.

The user can send control commands via a mobile device. The requested tasks are :

- Implementation of sensor nodes

- Implementation of a Lora Gateway
- Implementation of a Lora Server
- Implementation of a Mobile application

1.6 Conclusion

In this chapter, we defined the context of our project and reviewed existing solutions. We conclude that LoraWAN is the most suitable protocol for our application.Since it is unavailable in Tunisia, we will develop a secure protocol based on lora technology. In the next chapter we will detail the requirements of our application and model its behaviour using UML.

Chapter 2

Architecture and UML Modeling

Plan

Introduction

Throughout this chapter, we study the functional and non-functional requirements of our application. We then present a global description of our solution. In The remainder of the chapter, the UML is used to model the behaviour of our system.

2.1 Requirements analysis

In this section, we present the functional and non-functional requirement of the system.

2.1.1 Functional Requirements

Functional requirements are the features and actions that the system must perform.They are extracted from the specifications of the project . Our system should offer to the user the following operating mode:

- Checking temperature level
- Checking humidity level
- Checking air quality
- Receiving gas leak notification
- Receiving a notification if an intruder is detected
- Receiving a notification if a door is opened outside working hours.
- Receiving a notification if a light is on outside working hours.
- Getting building access notification: if a foreigner is in front of the door, the system takes his photo and send it to the user.

2.1.2 Non-Functional requirements

A non-functional requirement is one that specifies the system constraints, such as performance, ease of maintenance, scalability, and reliability requirements. Our system should meet the following non-functional requirements:

- Energy consumption:the system power consumption is low
- Performance: the system execution time should be low.

- Real-time operation:our system must guarantee a real-time response especially when transmitting alerts.
- Availability:the different data collected must be available at the request of the user.
- Reliability
- Security : All data exchanges should be secured by an encryption algorithm.

2.2 System architecture

We design a system that meets the previously mentioned requirements. Our system encloses four main components: nodes, gateway, the server and the mobile application. A node can be either a wireless sensor or a wireless actuator. In our system we need nine nodes: air quality node, humidity and temperature node, light node, gas node, motion node, a node to detect opened doors or windows, a camera node, a motor node and a light node. The figure 2.1 depicts the required system architecture.

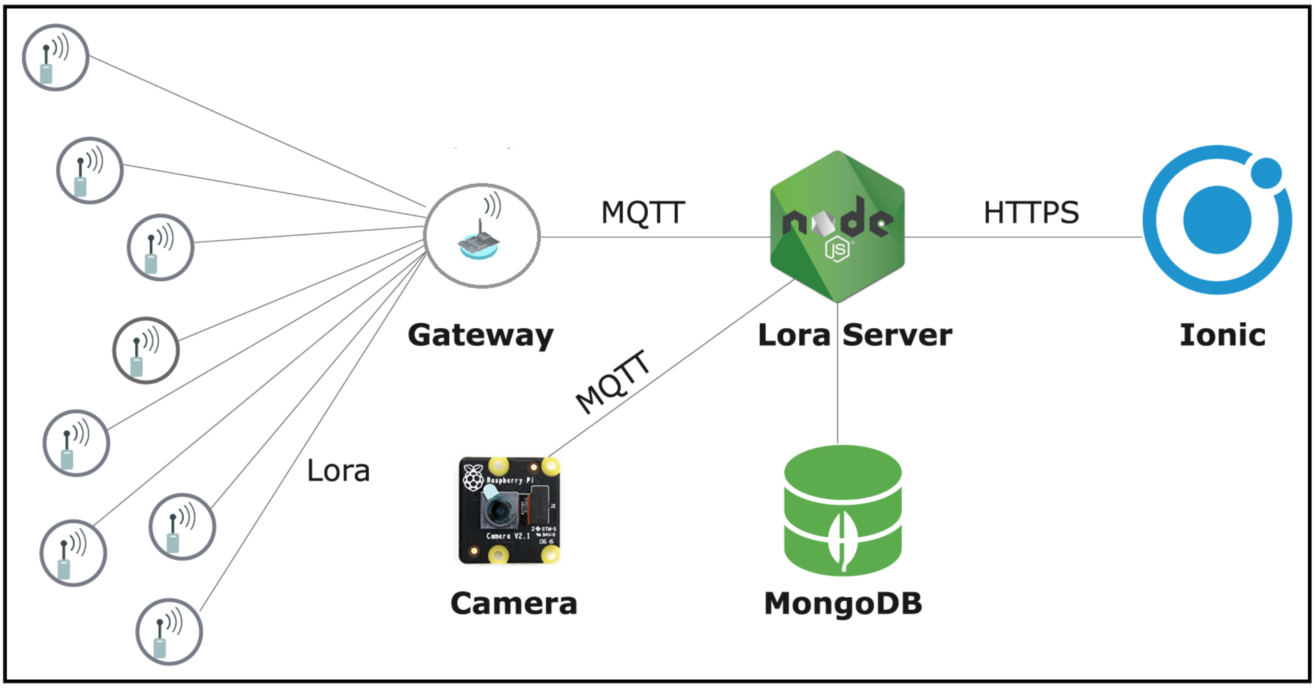

Figure 2.1: The system architecture

Since we were constrained to use the available hardware, we limited our system to 3 nodes (2 sensors and 1 actuator)as shown in figure 2.2.The first sensor node, which represents the camera node, communicates with the gateway through the WIFI protocol.The second sensor node englobes the rest of sensors. It communicates with the gateway using a lora technology.The communication between the gateway and the server is established through Ethernet.

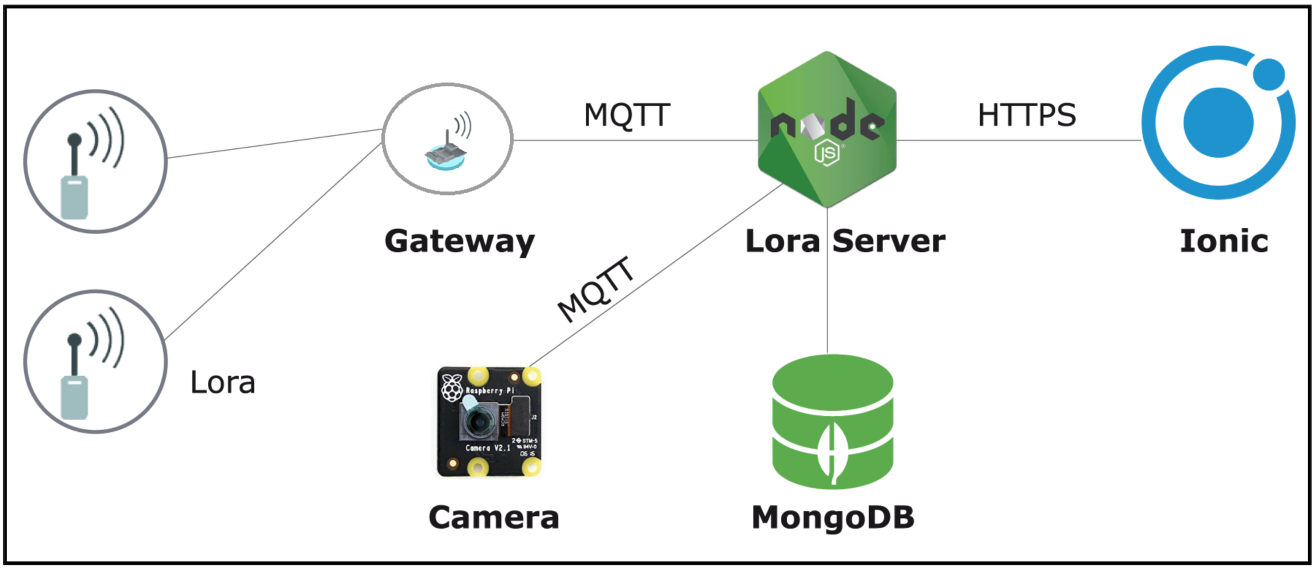

Figure 2.2: The system architecture

2.3 System Modeling

We use the UML language to model our system. We use WhiteStarUml as software for UML modeling.

2.3.1 Use Case diagram

In this paragraph we model the previously specified requirements using the Use Case diagrams. It summarize the different interactions between the user and the system. Figure 2.3 depicts the sequence diagram of our system.

2.3.2 Sequence diagram

the global sequence diagram of the system

Figure 2.4 depicts the sequence diagram of the system

Conclusion

In this chapter, we presented the functional and non-functional requirements.We proposed an architecture composed of four main parts:three nodes,a gateway, a server and a mobile application.We defined the protocols used between each part.And we finished the chapter by the UML conception which explains the system interactions.The next chapter, we will detail the embedded part of the system.

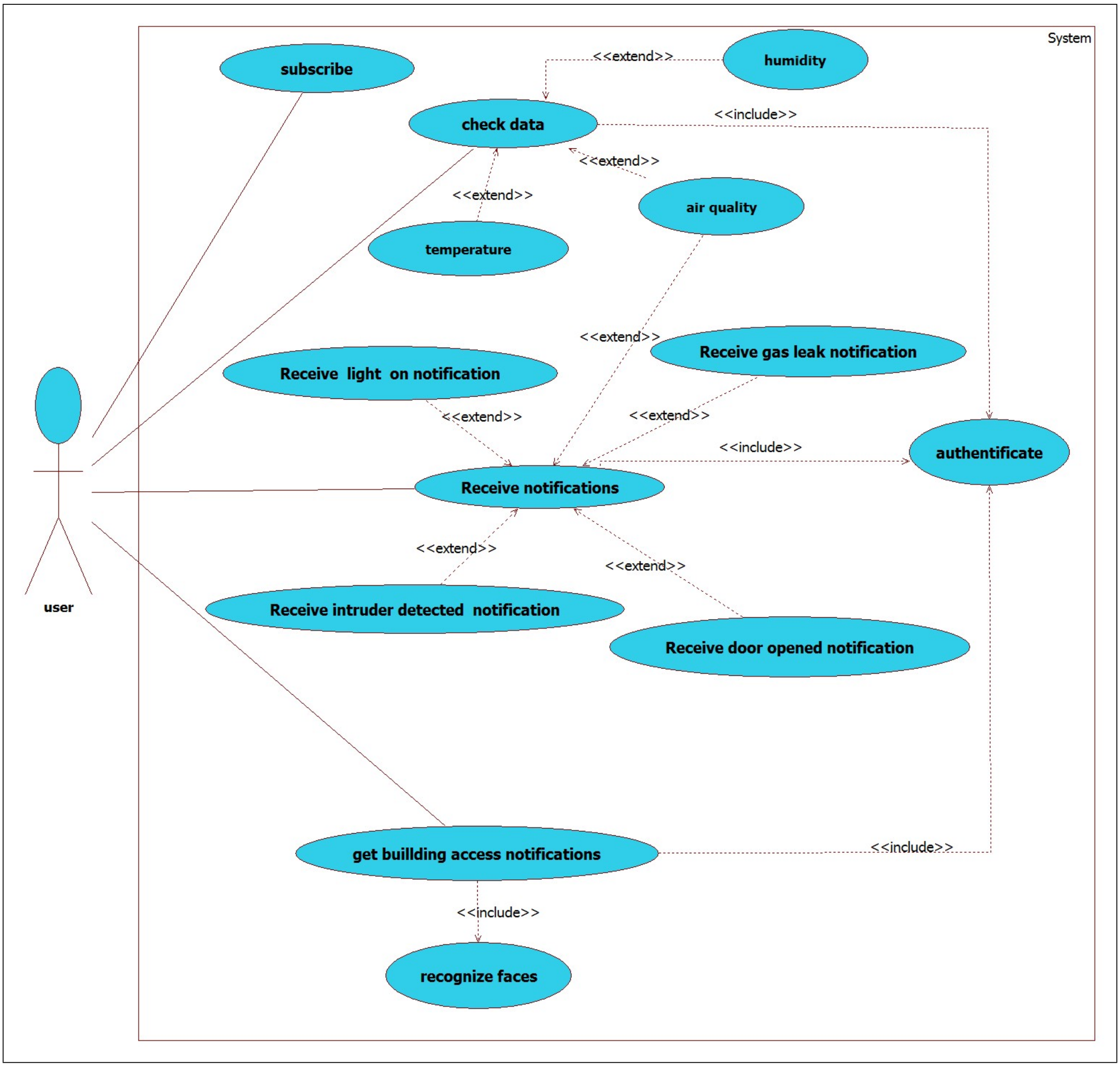

Figure 2.3: General Use Case Diagram

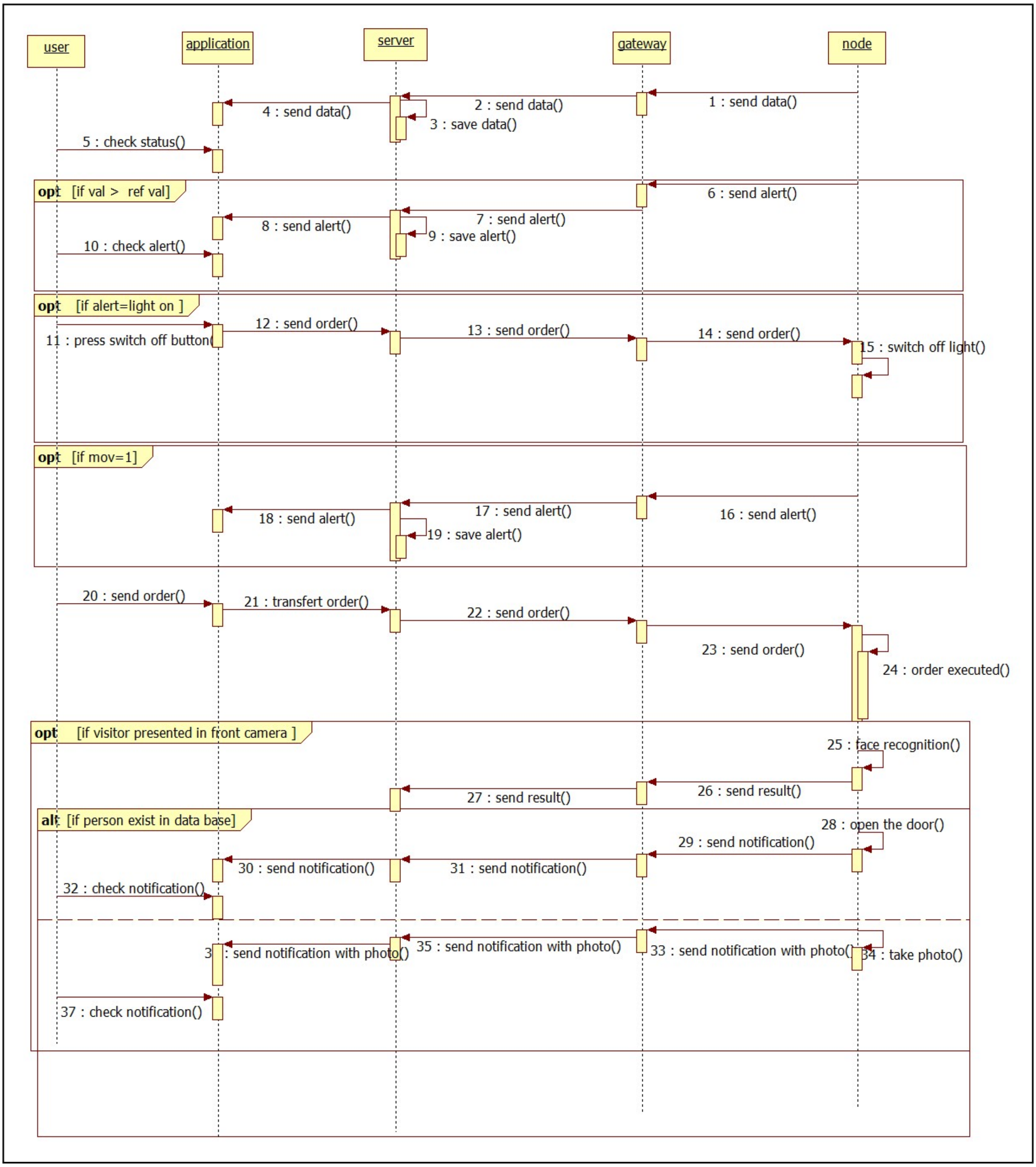

Figure 2.4: General Sequence Diagram

Chapter 3

Nodes and Gateway Design

Plan

Introduction

Across this chapter we detail the embedded part of the system which consists of two main parts: nodes and the gateway.

3.1 Nodes Network

As we mentioned in the second chapter, we have three nodes: an actuator node, a camera node and a node enclosing the rest of sensors. In the remainder of this chapter the third node will be named the sensors node.

3.1.1 Sensors Nodes

This node contains:

- Arduino Mega2560 Rev3
- Motion sensor HC-SR501
- DHT-22
- MQ-5 gas sensor
- MQ-135 gas sensor
- Hall effect sensor Module
- Photoresistor Module
- LoRaWAN module RN2483

We describe in this section, the operating mode of each sensor ,the LoraWAN module and the Arduino.

3.1.1.1 Motion sensor HC-SR501

HC-SR501 [11] is a passive infrared (PIR) sensor module, that detects physical movement. That's why it doesn't generate or radiate any energy for detection purposes. It detects the infrared radiation emitted or reflected from an object but it doesn't detect or measure temprerature.

The figure 3.1 illustrates the components of the HC-SR501 :

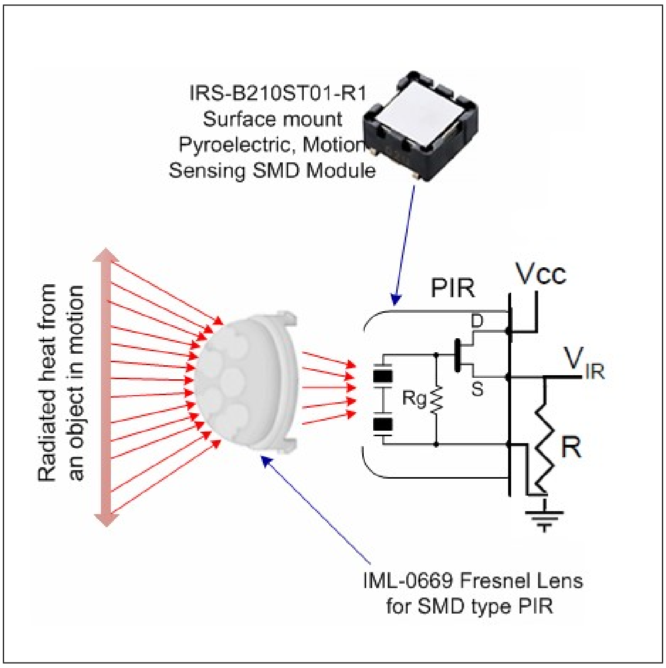

Figure 3.1: Components of the HC-SR501

Operating mode

The motion sensor has two slots. Each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or an animal passes by, it is intercepted by the first slot of the sensor(figure 3.2). The radiation emitted enters through the front of the sensor, known as the 'sensor face'. It strikes a solid state sensor or set of sensors made from pyroelectric materials[12] (naturally electrically polarized crystals generate energy when exposed to heat), which generates an electrical charge. As a result, a positive difference of potential is generated between the two slots. When the warm body leaves the sensing area, the reverse happens. Hence, the sensor generates a negative difference of potential.

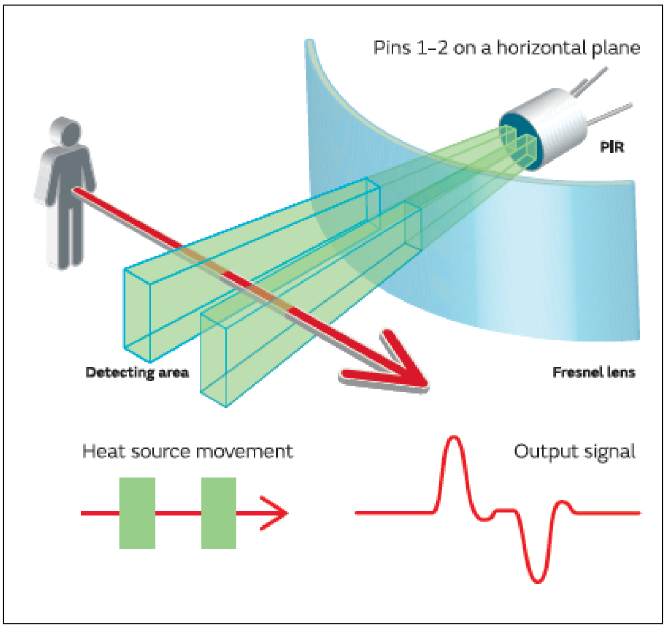

Figure 3.2: radiation detection by the motion sensor

Characteristics

- Voltage:5V – 20V
- Power Consumption: 65mA
- Sensing range: less than 120 degree, within 7 meters

3.1.1.2 DHT-22

The DHT-22 (figure 3.3)is a digital humidity and temperature sensor [13]. It uses a capacitive humidity sensor, an NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. * Operating mode To measure humidity it uses capacitive humidity sensor which has

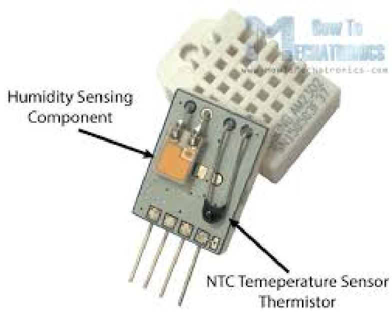

Figure 3.3: DHT22 sensor

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 3.4 depicts the capacitive sensor of the DHT22:

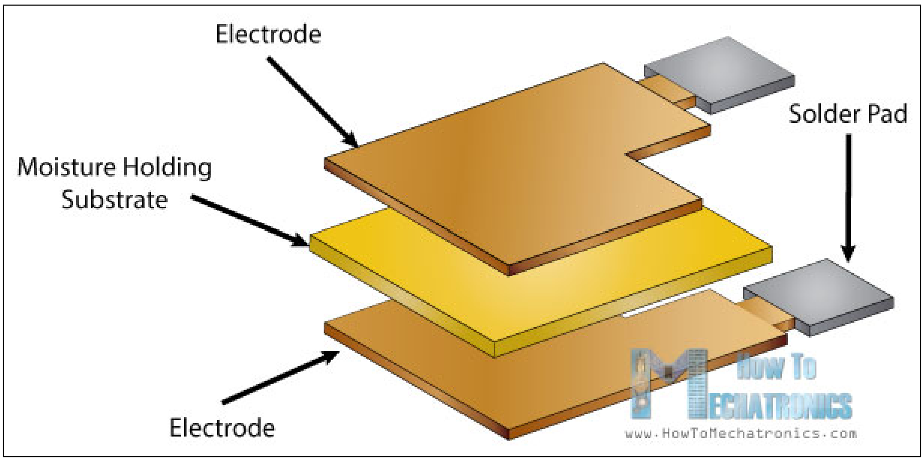

Figure 3.4: Capacitive sensor

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 3.5 illustrates The thermistor(NTC) operating mode of the DHT22:

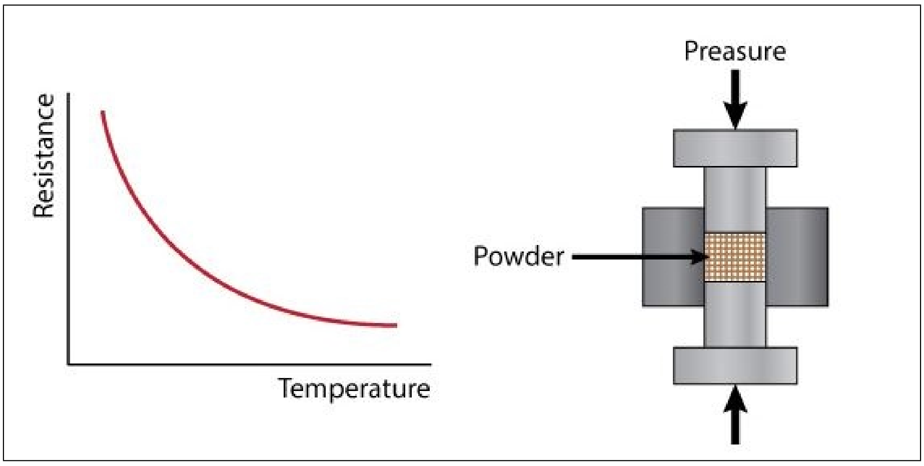

Figure 3.5: The thermistor(NTC) operating mode

Characteristics

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for -40 to 80C temperature readings 0.5C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)

- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing

3.1.1.3 MQ-5 gas sensor

MQ-5 gas sensor (figure 3.6) is used to detect gas leakage [14].

Operating mode

The heating coil H is in contact with the SnO2 filament This sensor contains an SnO2 filament sensitive to combustible gas connected to a heating coil H.In clean air the resistance across the heating coil does not vary so the filament have low electrical conductivity.When a combustible gas such as Methane is introduced ,the resistance of the filament drops.As a result the Output Voltage (Vout) rises .This voltage indicates the equivalent gas concentration.The figure 3.7 illustrates the structure of MQ5

characteristicss

- high sensitivity to LPG,Methane,coal gas.
- Working Voltage: 4.9/5.1 v
- concentration 300-10000ppm

Figure 3.6: MQ5

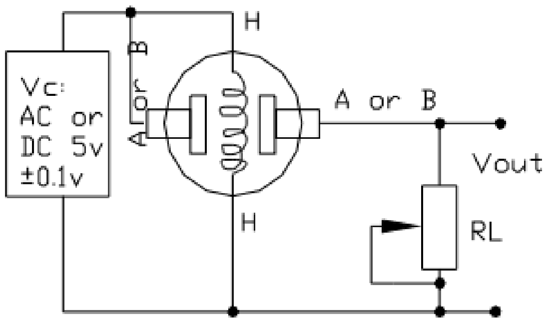

Figure 3.7: structure of MQ5

3.1.1.4 MQ-135 gas sensor

MQ-135 gas sensor[15],shown in the figure 3.8, is used in air quality control and air pollution detection.

Figure 3.8: MQ135 sensor

Operating mode

This sensor contains an SnO2 filament sensitive to combustible gas.In clean air the filament have low electrical conductivity.When a combustible gas such as ammonia is introduced , the filament electrical conductivity rises.As a result its resistance and conductance changes which indicates the equivalent gas concentration. The figure 3.9 illustrates the structure of MQ135

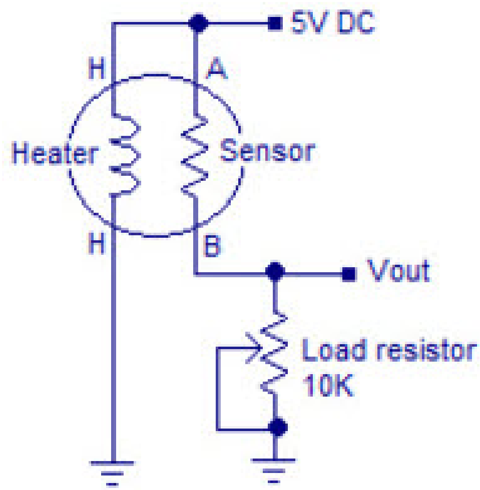

Figure 3.9: structure of MQ135

Characteristics

- Possesses high sensitivity to ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulfide and smoke.
- Working Voltage: 2.5/5.0 v
- Preheat duration 20 seconds

3.1.1.5 Photoresistor Module

Photoresistor (LDR) [16],shown in figure 3.10,is light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity.

Operating mode

In the darkness, their resistance is very high, sometimes up to 1M, and when exposed to light, the resistance drops , even down to a few ohms, depending on the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices.

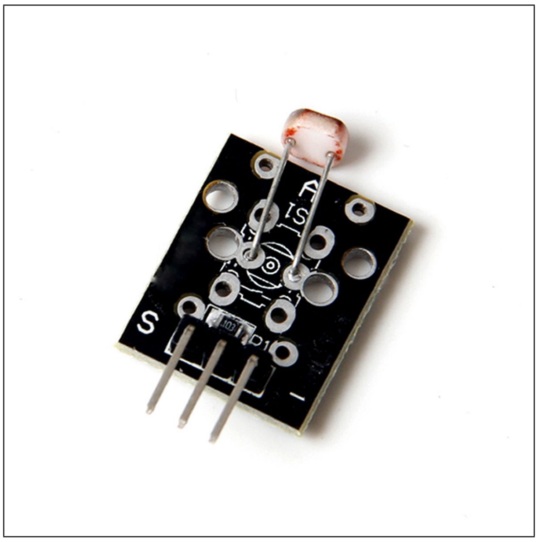

Figure 3.10: Photoresistor sensor

3.1.1.6 Hall effect sensor Module

A Hall effect sensor [17],represented in figure 3.11, is a transducer that varies its output voltage in response to a magnetic field.It is used to detect object position and to measure Motor speed.

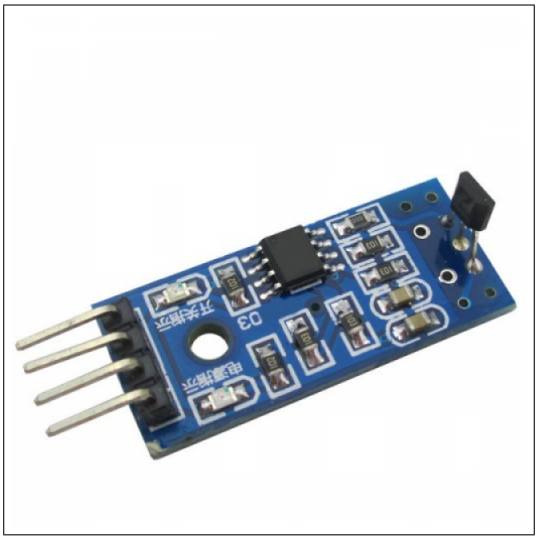

Figure 3.11: Hall effect sensor Module

Operating mode

If there is an magnetic field present, it generates a high level signal.

characteristicss

- Volt: 2.3V 5.3V

- Dimension: 27.0mm * 15.5mm
- Mounting holes size: 2.0mm

3.1.1.7 Arduino

Arduino[18] is an open source programmable micro-controller board based on easy-to-use hardware and software.The Arduino has a special language(set of instructions) and a software IDE.There is multiple Arduino boards,illustrated in figure 3.12 ,and each one has it's special features.For our project , we choose to use Arduino Mega 2560 Rev3,represented by the figure 3.13 , for the transmitting node because this node contains all the sensors that's why we need more input/output pins and the board has 70 input/output pin.

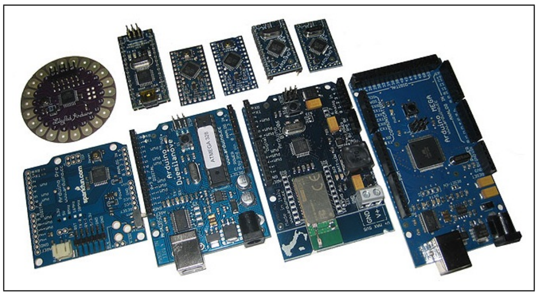

Figure 3.12: Arduino boards

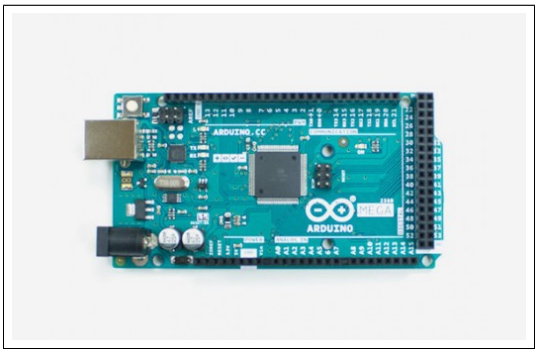

Figure 3.13: Arduino Mega2560 Rev3

Characteristicss of Arduino Mega2560 Rev3[19]

- 54 digital input/output pins (of which 15 can be used as PWM outputs)
- 16 analog inputs
- 4 UARTs (hardware serial ports)
- 16 MHz crystal oscillator

- USB connection
- power jack
- ICSP header, and a reset button
- 256 KB Flash memory
- Operating/Input Voltage 5 V / 7-12 V
- 16 MHz CPU Speed
- ATmega2560 Processor

3.1.1.8 LoRaWAN Module RN2483

The LoRaWAN module RN2483 [20]is a long-range transceiver module based on low power wireless LoRa technology.The RN2483 uses the 433MHz / 868MHz frequency band.The RN2483 Module uses a unique spread spectrum modulation within the Sub-GHz band to enable long range, low power, and high network capacity.The module's embedded LoRaWAN^{TM} Class A protocol enables seamless connectivity to any LoRaWAN compliant network infrastructure, whether public or privately deployed.It has two operating modes:

- point to point mode:communication between nodes.In this mode it works as a LoRa module.
- Hybrid Mode.It is the LoRaWAN mode (P2P + gateway to LoRaWAN Network)

It incorporates a base band controller, RF, Application Programming Interface (API) processor, making it a complete low power Bluetooth solution. The RN2483 is suitable for single-wide, long-range sensor applications with an external host MCU.In the rest of the chapters the RN2483 will be named RN.

Characteristics[20]

- ASCII command interface over UART
- Frequency: 868 MHz and 433MHz ISM frequency band
- Radio Bit Rate: from 250 to 5470 bps
- Operating Temperature Range: -40°C to 85°C
- Range: >15 km at suburban and >5 km at urban area

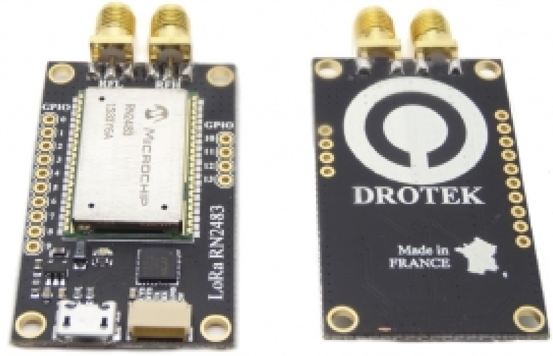

Figure 3.14: LoRaWAN RN2483 Module

- Chipset consumption: 38.9 mA
- 14 GPIO for control, status, and ADC
- TX Current Consumption: 40mA (14dBm, 868MHz)
- RX Current Consumption: 14.2mA

3.1.2 Camera Node

This node is composed of a Raspberry Pi3 and a PiNoir camera.

3.1.2.1 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B(Figure 3.15 illustrates this development board) is the third-generation of the Raspberry Pi series of small single-board computers developed by the Raspberry Pi Foundation.This chip, with a quad-core ARM Cortex-A53 processor, is able to support different operating systems[21][22]:

- Several variants of the GNU / Linux free operating systems (Debian, Fedora, Kali Linux, Raspbian, RISC OS, Ubuntu ...)

- Windows 10 IoT Core
- Android Things

Figure 3.15: Raspberry Pi 3 Model B

The technical characteristics of the Raspberry Pi 3 Model B are [23]:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

After a comparison between existing boards in Tunisia markets, we choose the Raspberry Pi 3 chip because it offers a high performance, has a large memory capacity and a camera module that can be added easily. Moreover it integrates a WIFI module that is usefull to establish a connection between the server and the camera node . Unlike the Raspberry pi board, Arduino is less powerful with an 8-bit processor with a frequency in the range of 16 MHz and 84 MHz. nd 2KB of RAM memory[19][24].To use our Raspberry Pi, we need the following accessories:

- MicroSD memory card: this card will contain the operating system of Raspberry Pi.The capacity of the memory card, must be at least 16 GB . It contains Raspbian operating system and data.
- Micro USB power supply (2.1 A)
- An RJ45 Ethernet cable: we use it to access the pi with a static ip through ssh

3.1.2.2 The camera module

The used camera module v2 is a Pi NoIR night vision sensor having 8 megapixels and without infrared filter (Figure 3.16.It is used basically in dark as it integrates infrared lighting to take videos as well as still photos.This module is connected to the Raspberry Pi through a 15cm ribbon cable to the CSI port of the pi and it is compatible with all versions of Raspberry Pi. The camera supports 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 videos [11].

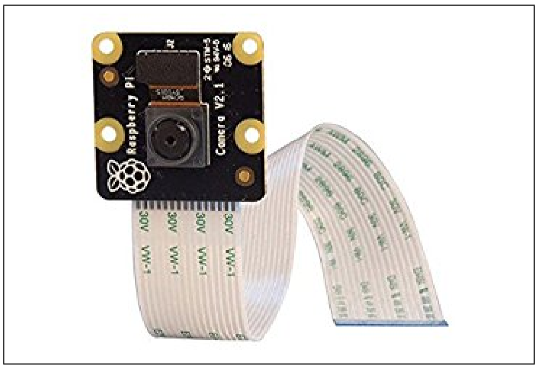

Figure 3.16: The Raspberry Pi camera module V2 night vision

The main task of this node is to control the access to the building by facial recognition. We worked with a python library based on the neural network.The technique of facial recognition consists mainly of three steps which are:learning, detection and recognition. The first step is detection, which consists of detecting the main characteristics of the face from the captured image by the camera. The second phase uses the faces data base and extracts a mathematical representation containing the characteristics of each face. The last phase used to compare the facial image to be recognized with the coordinates obtained from the data base according to the characteristics of the faces.

3.1.3 The Actuator Node

This node contains the actuators which are : stepper motor and a led. These actuators are connected to an Arduino UNO and a LoraWAN RN module.

3.1.3.1 Arduino UNO

Arduino UNO Rev3 ,represented in figure 3.17, is one of Arduino borad distributions, and based on the ATmega328P processor. We choose this board for the actuator node because we need few pins and the Arduino fits our needs.

Figure 3.17: Arduino UNO Rev3

characteristicss of Arduino UNO Rev3[19]

- 20 digital input/output pins
- 6 analog inputs
- 1 UARTs (hardware serial ports)
- USB connection
- power jack
- ICSP header, and a reset button
- 32 KB Flash memory
- Operating/Input Voltage 5 V / 7-12 V
- 16 MHz CPU Speed
- ATmega328P Processor

Figure 3.18: servomotor

3.1.3.2 servomotor

A servo motor [25]is self-contained electric device that rotate or push parts of a machine with great precision. .

This node receives orders from the mobile application that have been transmitted through the gateway.It controls the door.

3.2 Gateway

The gateway is composed of a raspberry Pi 3 B and a LoraWAN module. As we previously mentioned, the raspberry Pi has features that meet our requirements such as wifi communication and processor performance. The gateway is the communication interface between the Loraserver and nodes. For the uplink mode, it receives messages sent form an end-device and transfer them to the server. As for the downlink mode, the gateway receives messages from the server and transfer them to the nodes.

3.2.1 Uplink Mode

In the uplink mode, we need to setup a lora receiver module using the same configuration as a lora end-device. Once data is received, it checks the gateway identifier from the packet received in order to verify if it is a valid packet. If this condition is proved, then the gateway establish a connection to the mosquitto broker in order to send packet to the the LoraServer.

3.2.2 Downlink Mode

In this mode, the gateway establish a connection to the Mosquitto broker in order to receive a packet from the loraserver. Once the packet is received, it checks the gateway identifier from the packet received. If the identifier is approved, the gateway setups a lora sender module using the same configuration as the lora end-device.

3.3 Security

As we mentioned in section 1.3.3, we aim to develop a communication protocol ensuring a secure bidirectional communication. Following the LoraWAN protocol, we propose to use the AES128 encryption algorithm.

3.3.1 AES128 Algorithm

According to NASA researches , the design and strength of all key lengths of the Advanced Encryption Standard (AES) algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the secret level.
Top secret information will require use of either the 192 or 256 key lengths[26].
Therefore AES128 encryption algorithm will be crucial for our system, even though running this algorithm on Arduino is time consuming. According to Arduino Cryptography Library [27], the execution time of the AES128 on Arduino Uno (16 MHz) per byte is given as follows:

- The encryption takes 33.28us.
- The decryption takes 63.18us.
- The key setup takes 158.68us.

3.3.2 The Proposed Protocol

Security is crucial in radio protocols because anyone setting in the middle can capture messages and stores them. With AES128, encryption Algorithm, no one can decrypt messages being transferred without the application Key(AppKey). In addition, if a malicious third party is tampering with the messages, the integrity of the message check will fail. Therefore it is not possible to tamper with the transfered packets without having the Network Key(NewKey). It is also possible to re-transmit the same messages. This kind of attacks, called replay attacks, can be detected and ignored via the frame counters. When a device is activated, these frame counters (FCntUp and FCntDown) are both set to 0. Each time the device transmits an uplink message, the FCntUp is incremented and each time the network sends a downlink message, the FCntDown is incremented. If either the device or the network receives a message with a frame counter that is lower than the last one, the message is ignored. The frame counters reset to zero every time the device restarts (when the firmware is flashed or unpluged). As a result,

our system will block all messages from the device until the FCntUp becomes higher than the previous FCntUp.Therefore, we should reset the FCntup in the backend every time we reset tehe end device.This solution has been inspired from the lorawan network security layer[28].
Security will be checked in the uplink and the downlink side :

3.3.2.1 Uplink Data

The packet is generated using the following format :

Tableau 3.1: Textual description of uplink packet

Header	**Encrypted uplink payload**	**MIC**

The header contains the node identifier, the gateway identifier and the FCntUp frame counter. The payload contains critical data that need to be encrypted with AES128 AppKey in order to create an encrypted payload field. The MIC is the result of the encryption of the Header and the encrypted payload with AES128 NewKey in order to check the integrity of the message .

3.3.2.2 Downlink Data

The format packet that will be received from gateway is as follows:

Tableau 3.2: Textual description of downlink packet

Header	**Encrypted downlink packet**	**MIC**

The header contains the node identifier, the gateway identifier and the FCntDown frame counter. The encrypted packet downlink contains critical action message that need to be encrypted with a AES128 NewKey and compared to the received MIC to check the integrity of the message being received.
If there is a match , the encrypted packet payload will be decrypted with the AES128 AppKey.

Conclusion

In this chapter, we precised the nodes components and its operating mode, and we described the protocol that we set between the nodes and the gateway.We explained the security of the protocol that we have used for Lora communication.In the next chapter, we will present the Lora Server and the mobile application.

Chapter 4

Lora Server and Mobile Application

Plan

Introduction

During this chapter we talk about the Server and the mobile application

4.1 Lora Server

As we mentioned previously, LoraWAN demands the integration of a network server that we have described its functionalities in section 1.4. To mimic the security layer from it ,we have to develop a server that integrates LoraWAN security functionality.We name this server Lora Server.

4.1.1 comparison between Server languages

4.1.1.1 J2EE

Java 2 Platform, Enterprise Edition (J2EE) is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multi-tiered, Web-based applications [29].

4.1.1.2 Python

Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development[30].

4.1.1.3 PHP

PHP is a server side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Preprocessor, that earlier stood for Personal Home Pages[31].

4.1.1.4 Node.js

Node.js is a JavaScript runtime built on chrome v8 engine created by Ryan Lienhart Dahl in 2010 . Node.js uses an event-driven non-blocking I/O and Node Package Manager (Npm) as a package ecosystem [32]. Npm is the package manager for JavaScript and the world's largest software registry[33].

4.1.2 operating models of servers

4.1.2.1 Blocking model

Blocking model means that an operation block further processing until the end of its execution. Synchronous blocking model is how some web servers like java and PHP handle the I/O requests. For instance, when a program is reading from a file or a database it blocks everything until it finishes reading.
In this mode, the machine is holding onto memory and processing time for a thread that isn't doing anything.

4.1.2.2 Non-Blocking model

This mode refers to that an operation does not block further processing.
Asynchronous and non-blocking servers, like Node.js, uses only one-thread to process all the requests. Otherwise, when requests arrive to the server ,they are served once at a time. Let's go back to the example of the code that needs to read from a file or a database. In this case, the main thread does not wait until the operation finishes reading. It will continue running and when the DB operations completes it will be sent as a callback. Figure 4.1 illustrates the difference between the blocking model and non-blocking model.

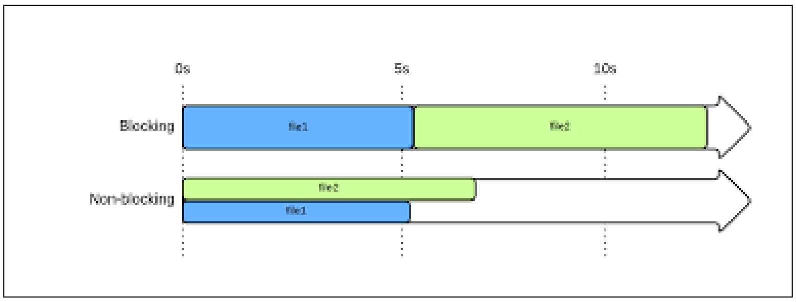

Figure 4.1: non-blocking model vs blocking model

Non-blocking model is very recommended in IOT applications to guarantee real time data exchange. Therefore, we choose Node.js for the development of our Lora Server. After choosing the server language,we have to define its framework :

Express

Express is a minimalist and flexible Node.js application framework that facilitates web and mobile applications development. It enables us to quickly develop node web based application by allowing middlewares set up in order to respond to http requests. It also provides a mechanism that is used to perform different actions based on HTTP method and Url called as routing table and allows to dynamically render HTML pages based on passing arguments to templates.

4.1.3 Database

Data base is a basic part to stock data and in this section we compare between the DataBase technologies to choose the suitable one for our application:

4.1.3.1 SQL:

SQL uses structured query language for defining and manipulating data. For exemple :MySQL

4.1.3.2 No-SQL:

No-SQL has a dynamic schema for unstructured data which can be stored in several ways such as document-oriented or graph based oriented. MongoDB is a free and open source platform classified as a No-SQL database. Due to the fact that our application will handle a significant number of data storage, it is essential to use a No-SQL database and more specifically MongoDB. MongoDB has no structure imposed into the document which means any document can be stored into any collection . It is the job of the developer to maintain the structure of the data to be stored, all fields of the document and their types. To handle this task, Mongoose is used to define the schema and the structure of our documents.

4.1.4 Security

In this chapter we talk about the Lora Server and the mobile application .Therefore, in this section we focus on the security between the Lora Server and the Mobile Application communication as well as the gateway and the Lora Server communication.

4.1.4.1 Security Of The communication Between the Lora Server And The Mobile Application

There is several ways to establish a security communication , in this section we mention the most used ways .And we compare between them in order to choose the suitable one for our application.

Authentication

- Cookies:A cookie is a small piece of data sent from a web server and stored on the client side. Each subsequent request from the client should include the cookie in the request header.

- Sessions:A session is a combination of a cookie with a session identifier. The server tracks the information associated with that session identifier. By default session cookies are stored in the memory side.

- Token:A token is a piece of data created by the server containing information to uniquely identify the user. A new token is created for every token request. Therefore there could be multiple tokens for the same user. Token has a lifetime. It is valid for 1800 seconds from its creation(configurable). Expired tokens are not valid and will be discarded. [34]

- Passport:Passport is an authentication middleware that supports various strategies for authentication. It includes the user local strategy using username and password. It also includes third party authentication using oauth and oauth2 which means getting access to his platform account using social networks like facebook, twitter and linkedin.

Cookies are very useful but they have limitations. They have a fixed size so they cannot encode a lot of information. A lot more information to be tracked about a client on the server side then session enables us to do that. Sessions seem to be the best authentication method for our project but later we have noticed that the Mobile application takes long time to handle cookies and sessions. Therefore token based authentication is more useful. To set up an authentication we use both passport with local strategy and token .

Https: To establish a communication between the Lora Server and the mobile application, we must think about doing that over a secure channel. Hence, if there is some one in the middle, he can't access to the messages being exchanged between the sender and the receiver. To do that we should not use HTTP. Instead we should use a secure version of http protocol called HTTPS. So we use SSL(socket secure layer) and TLS (transport layer protocol). These two cryptography protocols enables a secure communication between the Lora Server and the mobile application. These two components will communicate over the internet using encrypted messages. SSL and TLS maintain both integrity and privacy. Integrity means that the receiver will be assured that the message has not being tampered with.

4.1.4.2 Security Of The communication Between the Lora Server And the Gateway

Data Uplink :

In the Uplink mode, a packet received from the gateway over the mqtt protocol needs to be encrypted with a secret AES128 NewKey. Then it is compared with the mic value which is retrieved from the packet in order to check its integrity. If there is a match, the packet payload will be decrypted with a secret AES128 AppKey. The server will retrieve the FcntUp field from the payload in order to ignore the redundancy of packets being received.

Data Downlink :
In the Downlink mode, the server will create a packet with the message received from the mobile app. It encrypts the payload with a AES128 AppKey and encrypst the packet with a AES128 NewKey to calculate the Mic. Then, it generates a final packet that will be send to the gateway. Each subsequent command received from the mobile application will automatically increment the FCntDownk field in the packet.

4.1.5 Communication between system components

The communication between the Lora Server and the mobile application is established via RESTFUL API and websockets (socket.io) in order to send JSON data.
REST: is an architecture style based on HTTP protocol. The access to the resource URL is established via Rest methods(GET,PUT,POST,Delete)

Socket.io : It is a javascript library that uses websocket protocol designed to enable real-time bidirectional communication between web clients and servers. The communication between the LoraServer and the gateway is established via MQTT protocol using mosquitto as a broker.

Mosquitto : Mosquitto is an open source message broker that implements the MQTT protocol versions 3.1 and 3.1.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers[35].Figure 4.2 illustrates operating mode of the MQTT.

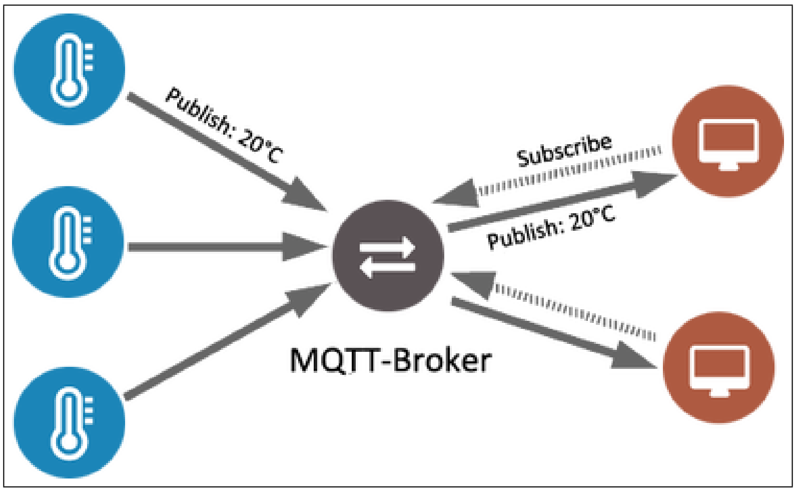

Figure 4.2: operating mode of MQTT

4.2 Mobile application

Mobile applications have different languages and plate-forms.

4.2.1 Mobile application languages

4.2.2 Android

Android is an open source and Linux-based Operating System for mobile devices such as smart-phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android[36].

4.2.3 Ionic

Ionic is an open source, cross-platform framework used to develop hybrid mobile applications. Ionic is a mobile app development framework based on the HTML5 programming language. The HTML5 framework always works with the help of Cordova or PhoneGap. Hence, Ionic needs both of them.
Cordova:Cordova is an open-source mobile development framework. It allows you to use standard web technologies such as HTML5, CSS3, and JavaScript for cross-platform development[37].
PhoneGap:PhoneGap Build is a cloud service for compiling PhoneGap applications. Adobe PhoneGap provides a way for users to create mobile applications using technologies such as HTML, CSS, and Javascript[5].

4.2.3.1 Advantages of the Ionic Framework

- It is platform independent whether it has to work with HTML, CSS, JS, or AngularJS.
- It helps in creating default mobile app UI functionalities easily and efficiently.
- It is wrapped by Cordova and PhoneGap.
- It works on iOS's UIWebView or Android's WebView[38].

For the development of the mobile application we are going to use Ionic as it is cross-platform.

Conclusion

In this chapter, we compare between different technologies used to develop Lora server which is an application server with a security layer. We choose the Node.Js technology because it's a non-blocking model.In addition, we choose the MongoDB as DataBase .The mobile application is implemented with Ionic because it provides tools and services for developing cross-platform mobile applications. The security is established via AES128 algorithm.

Chapter 5

IMPLEMENTATION

Plan

Introduction

In this last chapter, we finish our report by detailing the implementation and the test of our project. We also discuss the encountered problems.

5.1 Node Implementation

As we have Arduino in the nodes, we worked with Arduino IDE [39]. The communication between the Arduino and the RN moduleis is UART. fisrt, we have to setup lora module with the list of available RN commands. There are 3 main types of commands[40]:

- sys for system command
- mac for LoRaWan protocol related command
- radio for Lora radio transmission

Since the gateway is a local gateway designed for our system, the communication between the node and the gateway is peer to peer. Otherwise, we have to make the RN module work using the lora technology. Therefore we have to pause the mac level that manage the LoRaWAN protocol.For that, we use the "mac pause" command after setting up the baud rate of the serial port to 5600bps. we use radio commands to set Lora mode by the command "radio set mod lora".Then we set the frequency to 868.1 MHz because 868MHz is the iot frequency band used in Tunisia. The command used for that is : "radio set freq 868100000"
After that, we have to set the spreading factor 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. We have to set a parameter of error correction, the CR (Coding rate). 4/5 is the best correction mechanism especially when the signal is low. The used bandwidth specifies time on air and sensitivity.The RN bandwidth is from 125KHz to 500KHz. We use 125 KHz. Moreover,Power level gives the power in dB of the transmitter ; values are from -3db to 15dB.Figure 5.1, illustrates the RN configuration in the node.
And table 5.1 illustrates the different parameters and the attributed values.

Tableau 5.1: Parameters Table

Parameter Name	Signification	attributed value
freq	frequency	868.1MHz
SF	spreading factor:chirp spread spectrum modulation	7
CR	Coding rate	4/5
bw	bandwidth	125 KHz
pw	Power level gives the power in dB of the transmitter	13

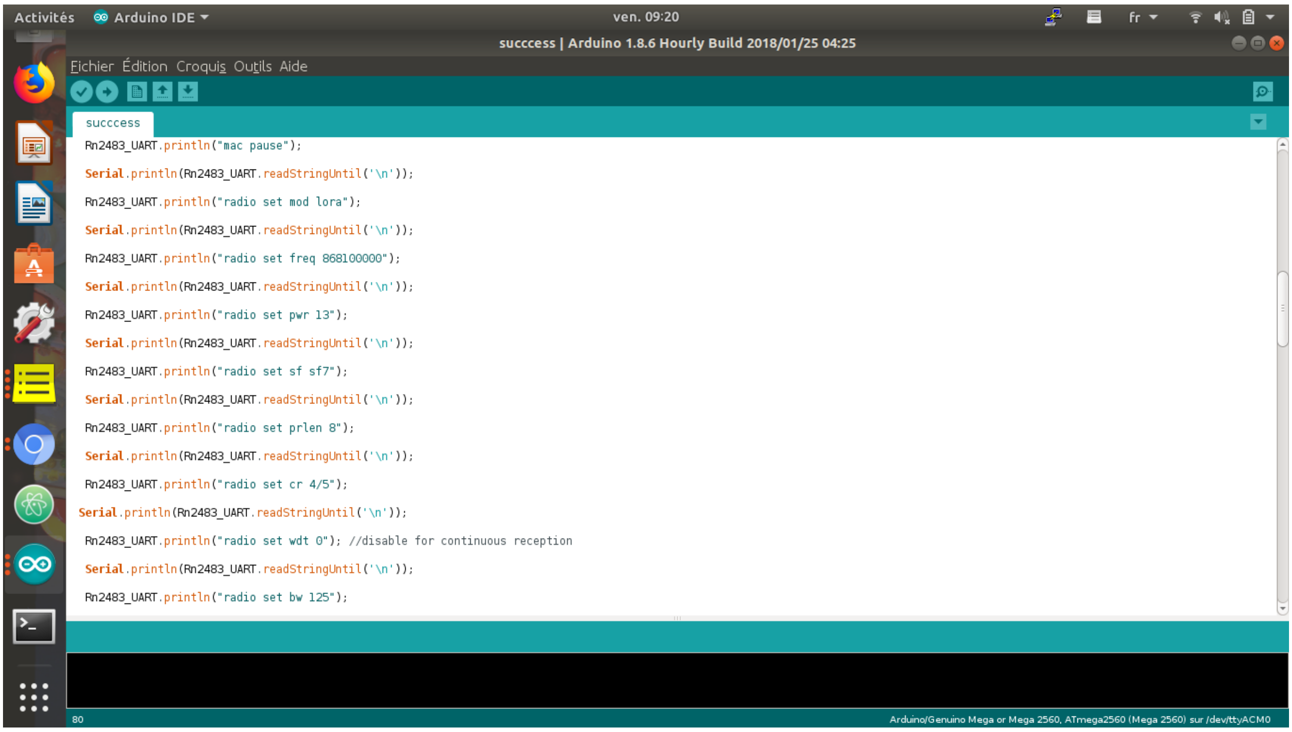

Figure 5.1: RN node configuration

The Sensors node (figure 5.2)sends data as an Hexadecimal string and the message should begin with "radio tx". The camera node works well. The last node, the actuator node, we still working on it. The order can be executed by the mobile application and can be transmitted to the server but the MQTT broker is still not transmitting the message to the gateway. We are going to fix this problem .

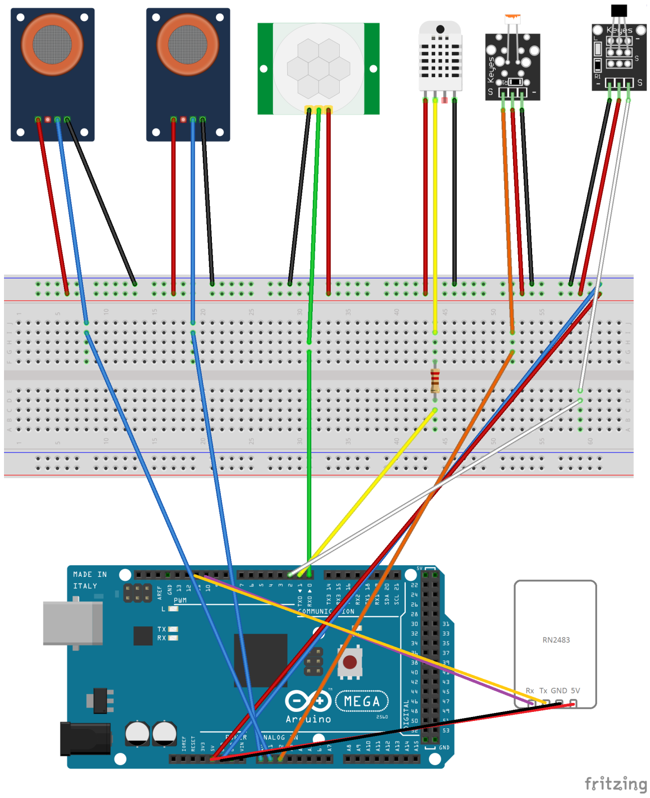

Figure 5.2: Sensors Node

5.2 Gateway Implementation

The same configuration done in the Sensors node should be performed in the gateway (illustrated in figure 5.4). We use python [41] to develop the gateway script. Python is commonly used by raspberry pi developers. It supports packages and modules ready to use for the raspberry Pi. Figure 5.3 illustrates the RN configuration in the gateway. We configured the LoraWAN RN2483 in a single channel mode. Hence, we cannot use one module for both uplink and downlink modes. Therefore we must install two modules on the raspberry pi one for the uplink and another for the downlink . The data received from the node is transmitted to the MQTT Broker. Figure 5.5 depicts the data received in the gateway.

5.3 Lora Server Implementation

The packet has been successfully received from the gateway , decrypted, sent to the mobile application and stored in the database. For the downlink phase, the control messages have been successfully received from the mobile application and transferred to the gateway.

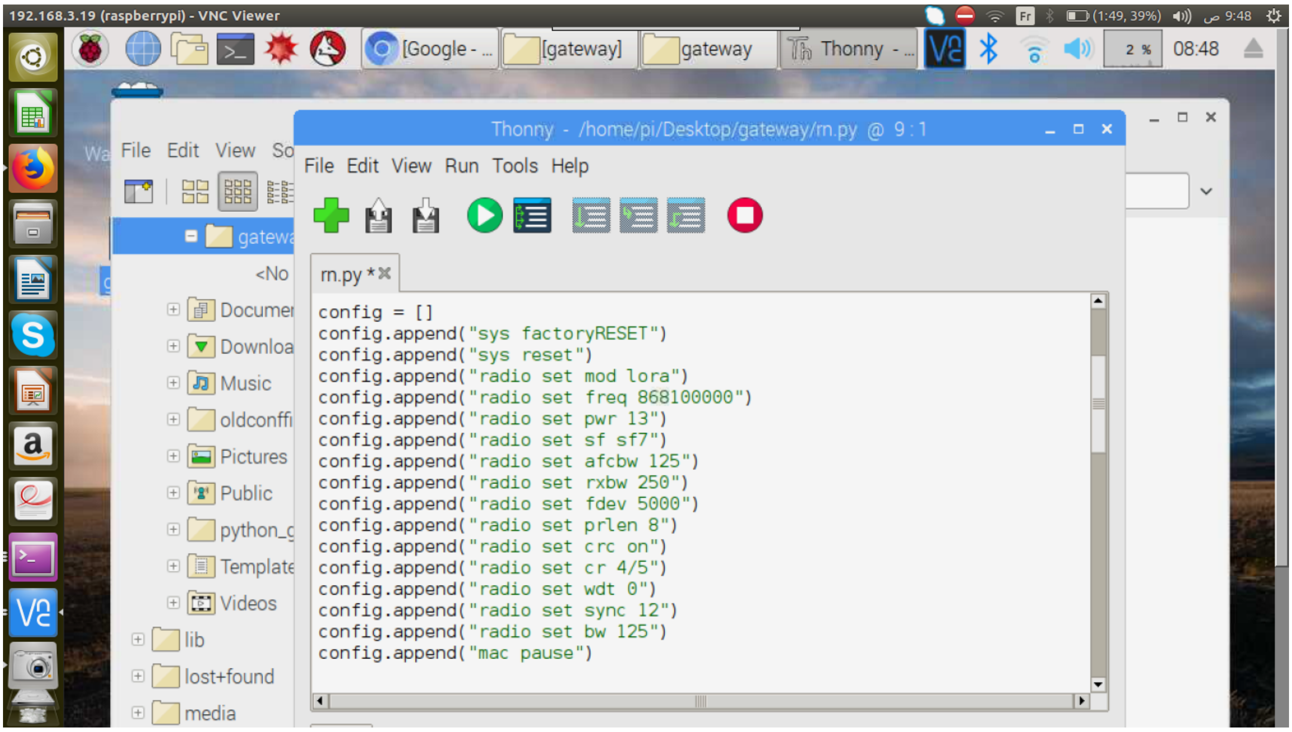

Figure 5.3: RN gateway configuration

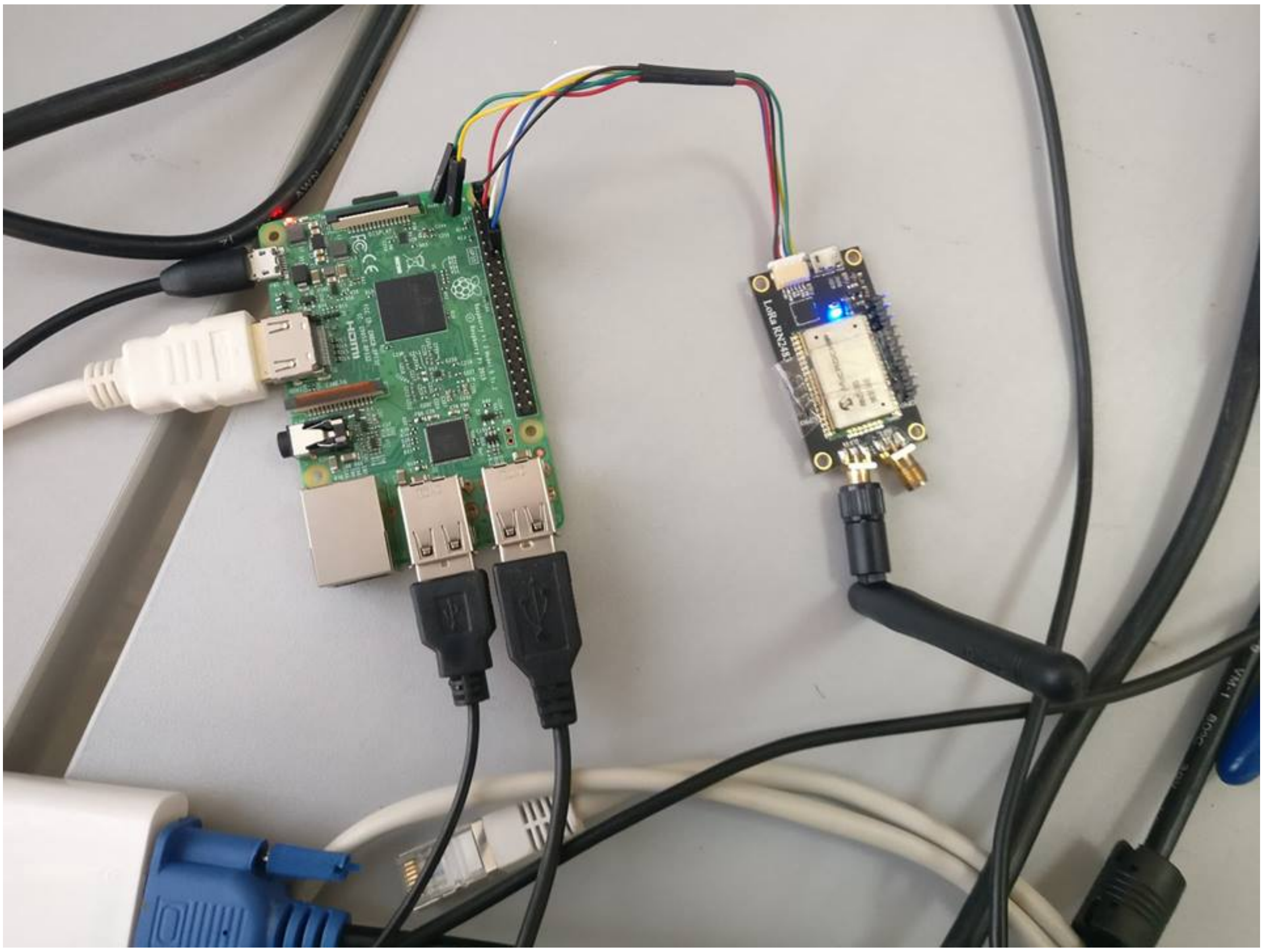

Figure 5.4: RN Gateway

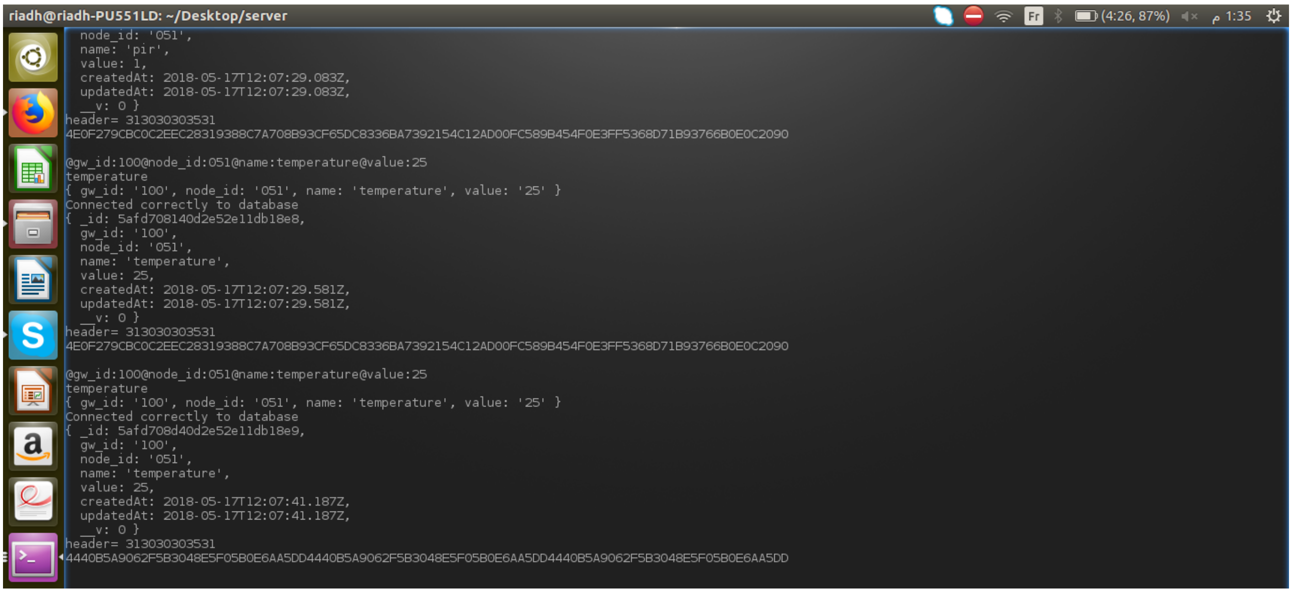

Figure 5.5: received data

5.4 Mobile Application implementaion

The main role of the mobile application is to visualize different information such as the temperature level, the humidity level and the air quality level. Through this application, the user can check the curves of temperature level during a period. In addition, the application notifies the user in case this cases:

- an intruder is detected
- a door is open beyond working hours
- a gas leak is detected
- light is on beyond working hours

The application contains four views(figure 5.6).

The first view is the login (figure 5.7).

The second view is the control view(figure 5.8): it is for controlling the light and the door.It visualize the notifications in the previously mentioned cases.

The third view is the curve view(figure 5.9):to visualize the curve of temperature. This view is not yet working as it should be. The problem is when the curve is loaded,the values extracted from the database are not token into consideration..We focused more in the embedded part as it is more important and now we are working to find a solution to this problem.

The fourth view is for subscription(figure 5.10): every user should have an account to manage his own building.

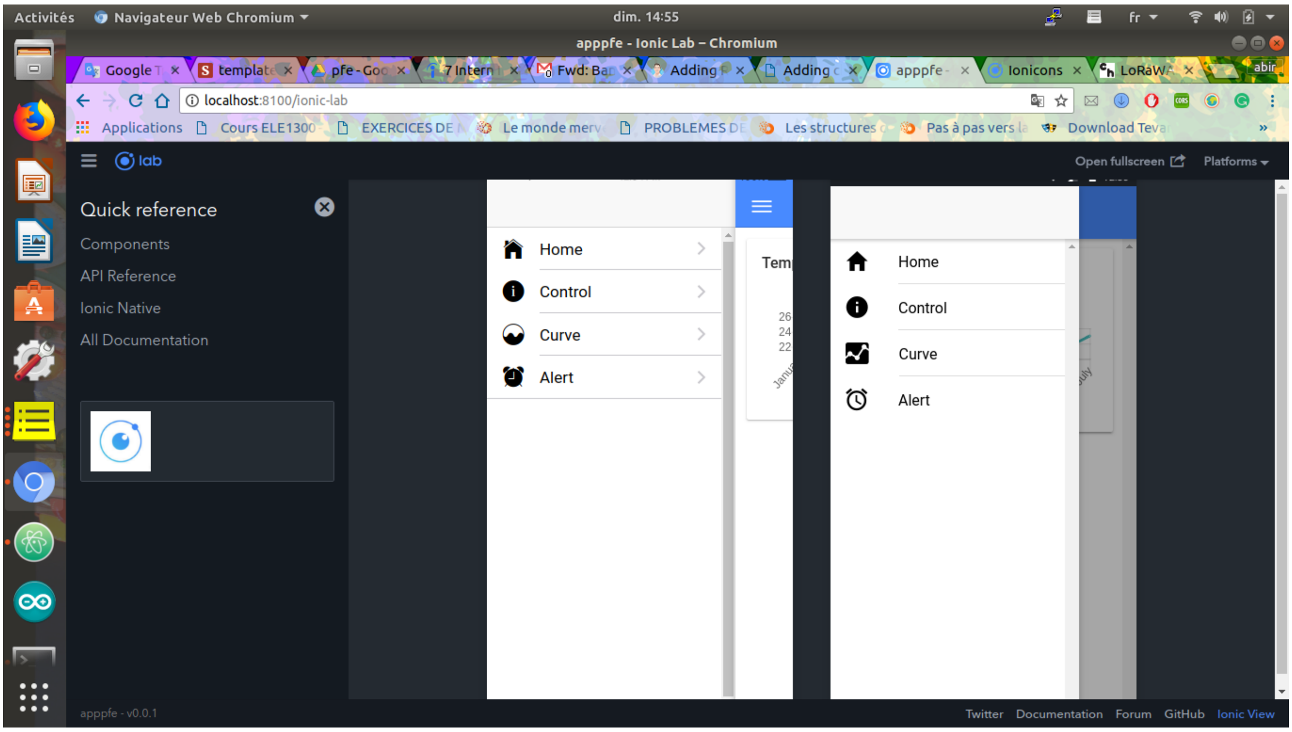

Figure 5.6: the application Views

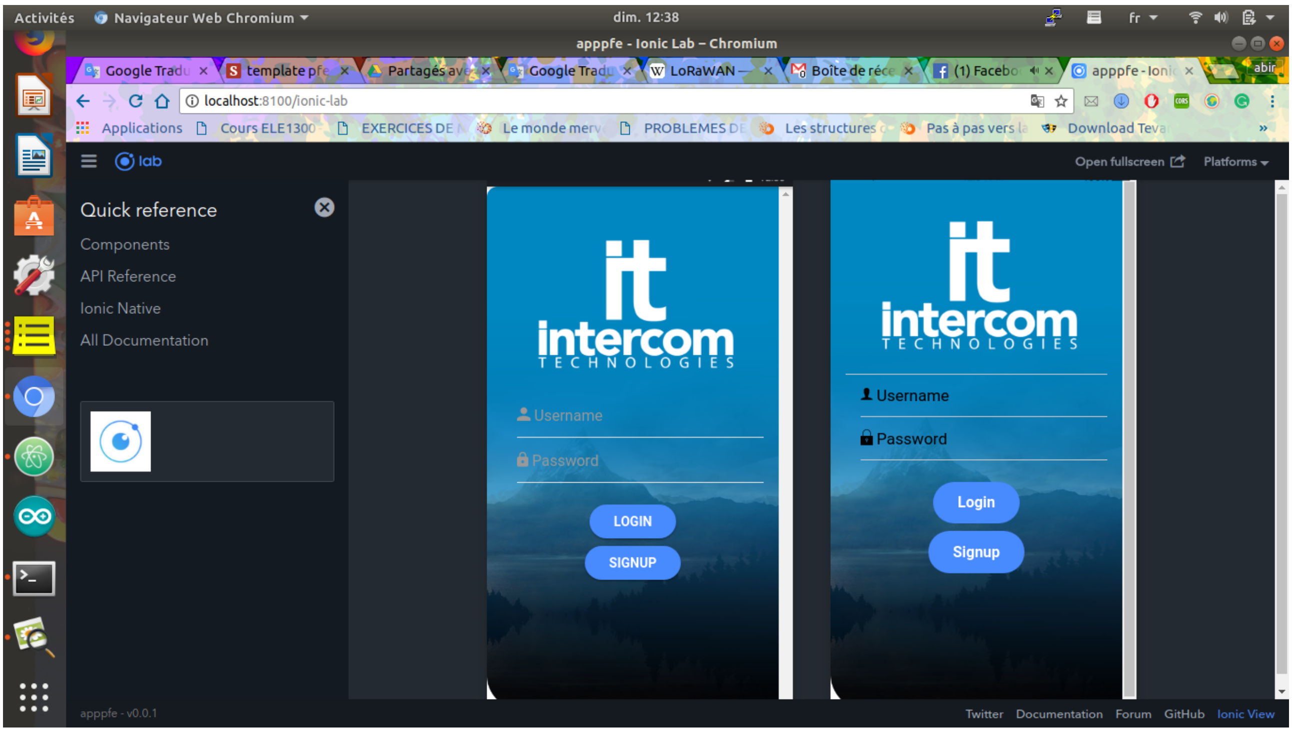

Figure 5.7: home View

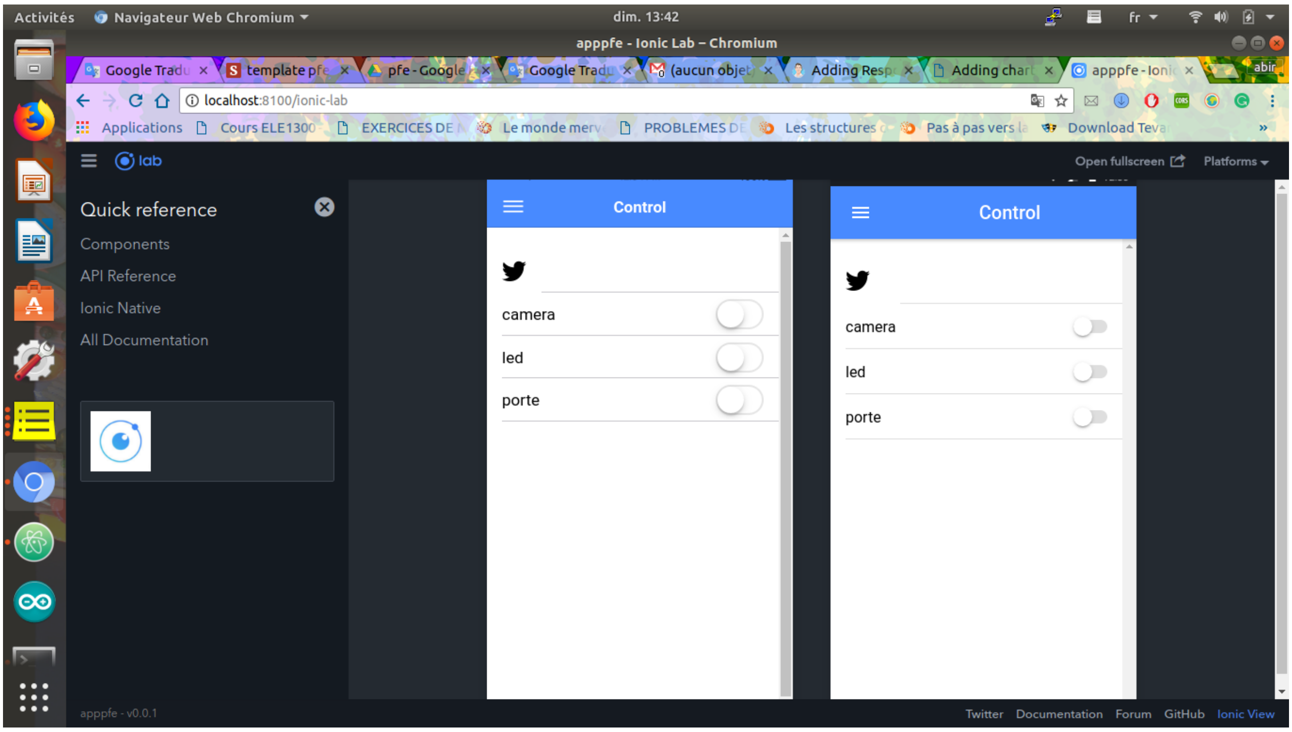

Figure 5.8: Control View

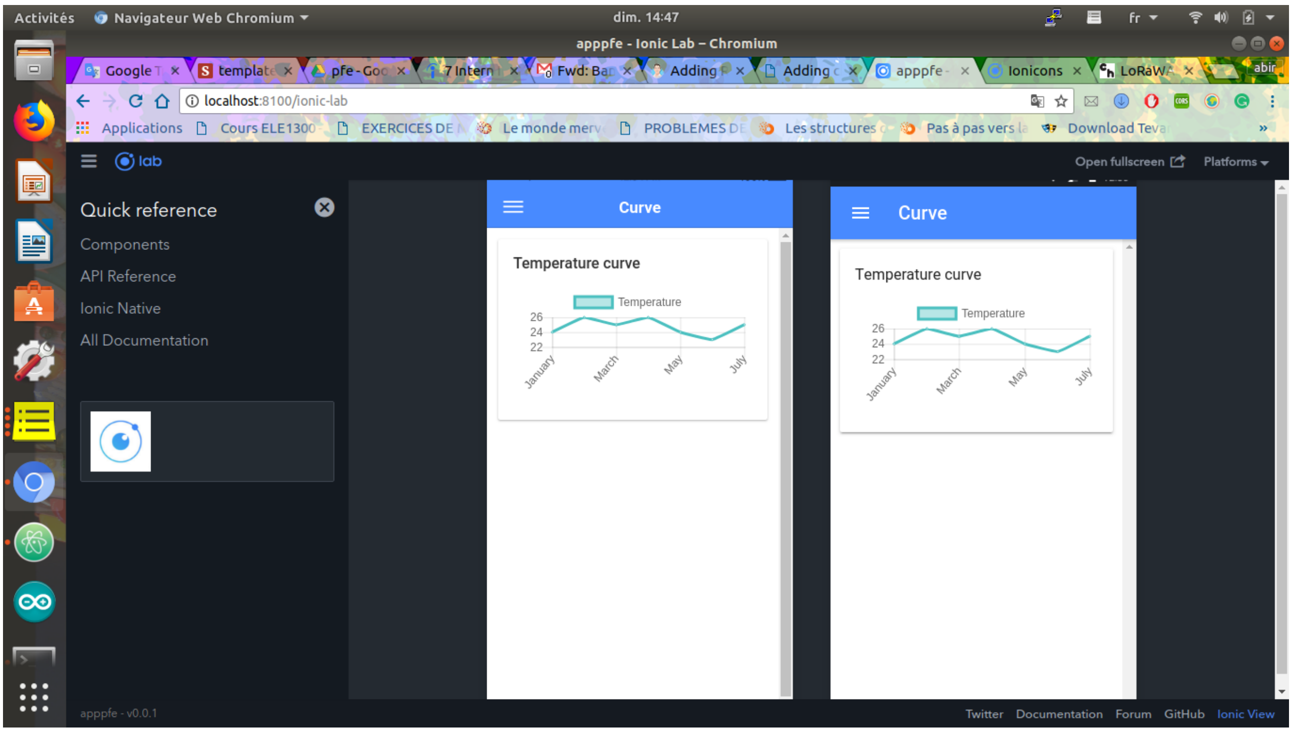

Figure 5.9: Curve View

The information sent from the server are successfully visualized(Figure 5.11).

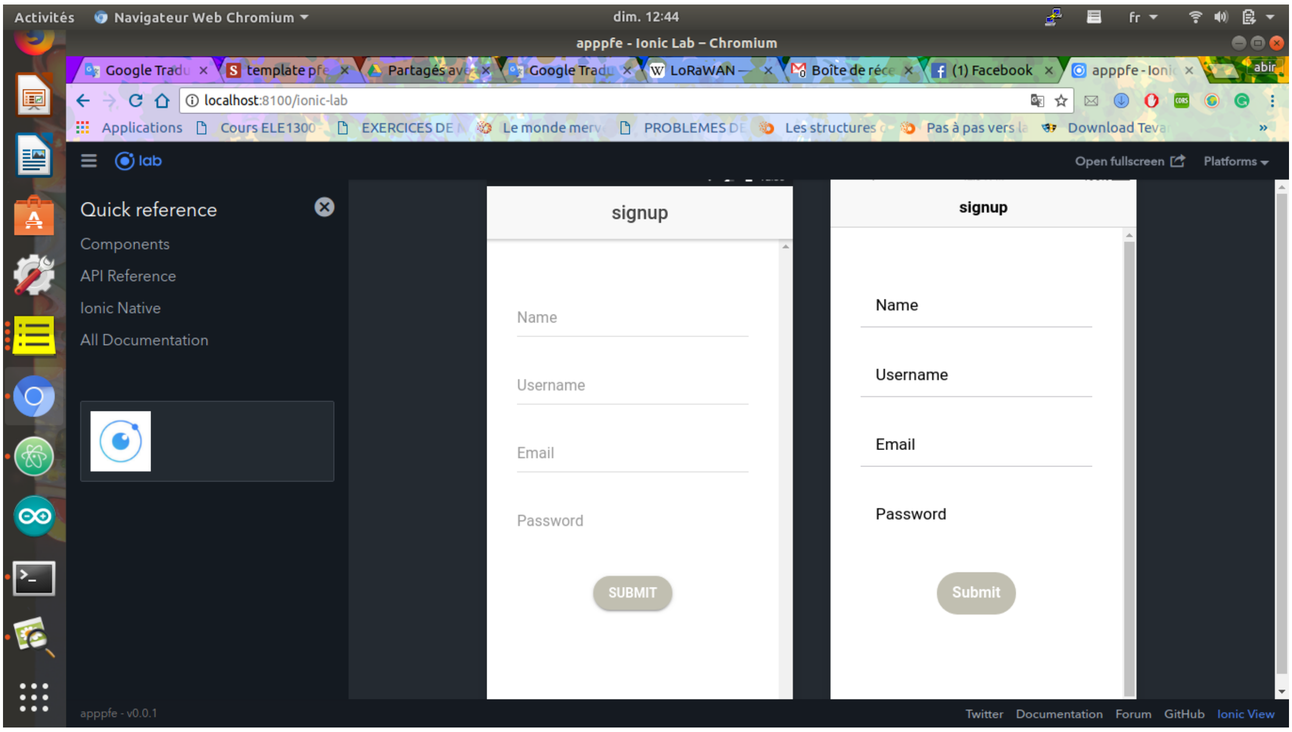

Figure 5.10: Subscribe View

5.5 Discussion

In this section we present the advantages , the encountered problems and the proposed solutions.

5.5.1 Advantages

Our system is perfectly working in real time and it also guarantees a perfect security no one can read transmitted messages without having the application key. The system is low power consuming, the batteries last for a long time(10 years at least).

5.5.2 Problems encountered and Proposed Solutions

5.5.2.1 Node tests

The RN module accepts only a Hexadecimal string. Since the temperature sensor DHT22 issues a number, a conversion has to be performed. This step always generates additional zeros and after a certain time, it stops giving right temperature values. The serial monitor shows inexplicable format of data. Each time we get this behavior, we have to reboot the board manually. In fact, the problem is the low capacity of the ram integrated into the Arduino

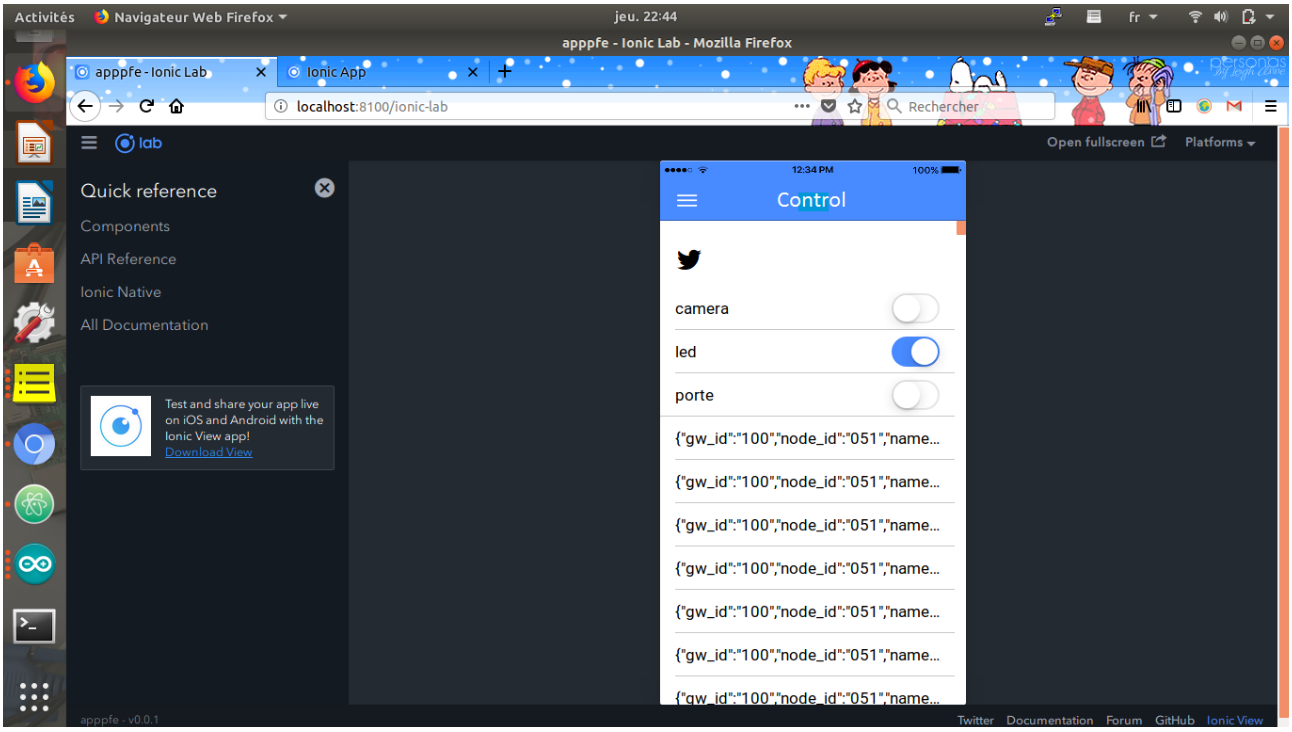

Figure 5.11: information visualization

Mega(8KB). As the code is a loop, it may cause memory fragmentation. To fix this problem, we implemented a function that contains the instruction: asm volatile (" jmp 0"). This instruction restarts the program. After resolving that problem, the first step of sending clear data is achieved in short and fast time. As we introduced AES encryption the Arduino becomes slow and it affects the transmission time. Therefore we encrypt only the data and we let the header without encryption.

5.5.2.2 Gateway tests

The RN module communicate with the Raspberry Pi through UART communication. Raspberry Pi 3 model B has only two UART interfaces: a hard(PL011) and a soft one (mini UART). The PL011 is connected to the Bluetooth module, while the mini UART is used for Linux console output. The UART doesn't work even after we enable it because we should update the repository and upgrade the installed packages and the most important thing is to disable the console.

Moreover, the RFM95W is a LoRa module that is used in real time application more than the RN module. We intended to use it in the gateway instead of the RN module(figure 5.12). Since the RFM module has a UART converter and the pins are very thin, the connection is not stable. Therefore, we desoldered the RFM from the groove converter and we used the SPI communication between the RFM and the Raspberry Pi 3, as it is shown in the annexe 5.13.

We used a C++ library to manipulate the RFM. The transmission between the gateway and the RFM is perfectly working in real time. But when we introduced the AES encryption, the information received from the node is not correct. After a thorough study, we concluded that the problem is due to incompatibility between the RMF and the RN modules. Since we have limit hardware resources, we choose to use the RN module as it handles encrypted data from the node and it has an acceptable time of transmission.

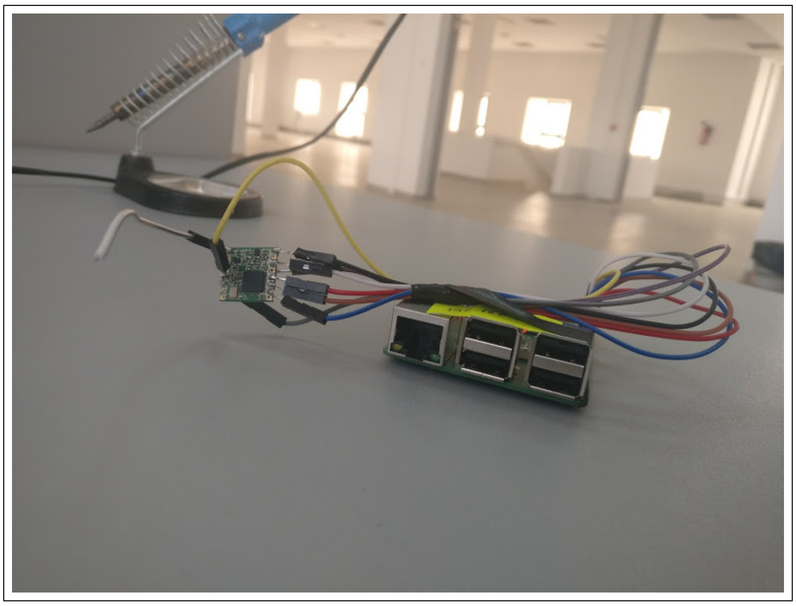

Figure 5.12: RFM gateway

Conclusion

In this chapter we have described the implementation of every part of our project.Moreover,we mentioned the systems advantages .We finish it by presenting the problems encountered in the implementations and we explained the proposed solutions.

General Conclusion

Our project is elaborated within Intercom Technologies. It aims to make a Wireless Sensor Network For the Remote Monitoring of Buildings.
In this report, we have presented the LoraWan architecture and we have concluded that it is the most suitable protocol for our application. Since we don't have a developed LoraWAN network in Tunisia, we have decided to mimic it, by using the Lora technology and we add to it a security layer. We have also presented our system architecture, UML design, the end-device, the Lora Server and the mobile application implementation. These four months of internship have been very beneficial for us as we have learned the IOT principles likewise we have deepened our knowledge in embedded systems, web, and mobile application development. As we mentioned previously, our system is working perfectly and the problem encountered has been solved successfully but it still needs some improvements: The use of an Arduino card is not really a perfect choice as we noticed that the encryption takes too long to be executed. Therefore it would be better to use a performing board of higher processor frequency and higher memory storage. And the use of an Rn module as a gateway has proved to be less efficient in term of real-time; compared to the RFM module.

Annexes

Annexe1

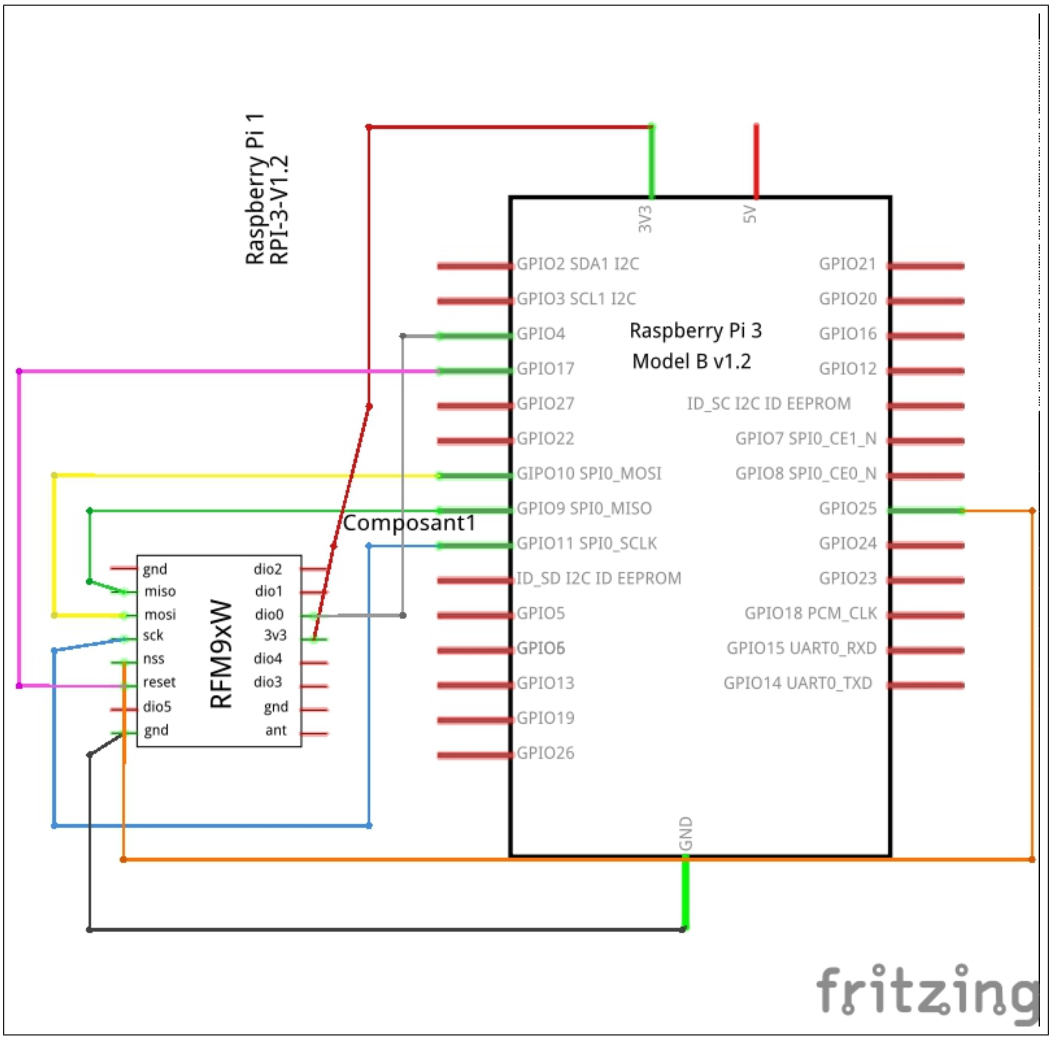

Figure 5.13: node cabling

Webography

[1] Available on: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ [Accessed 14-March-2018].

[2] Available on: https://www.sciencedirect.com/science/article/pii/S0747563216302990?via%3Dihub [Accessed 14-March-2018].

[3] Available on: https://dspace.cvut.cz/bitstream/handle/10467/64704/F3-BP-2016-Pikous-Tomas-BP-Tomas-Pikous-2016.pdf [Accessed 14-March-2018].

[4] Available on: https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol [Accessed 14-March-2018].

[5] Available on: http://docs.phonegap.com/phonegap-build/overview/ [Accessed 14-March-2018].

[6] Available on: https://web.archive.org/web/20130118103359/https://docs.zigbee.org/zigbee-docs/dcn/12/docs-12-0629-01-0mwg-zigbee-rf4ce-a-quiet-revolution-is-underway-webinar-slides.pdf [Accessed 14-March-2018].

[7] Available on: https://lora-alliance.org/lorawan-for-developers [Accessed 14-March-2018].

[8] Available on: https://www.cooking-hacks.com/documentation/tutorials/lorawan-for-arduino-raspberry-pi-waspmote-868-900-915-433-mhz/ [Accessed 14-March-2018].

[9] Available on: https://www.sciencedirect.com/science/article/pii/S2405959517302953 [Accessed 14-March-2018].

[10] Available on: http://www.rfwireless-world.com/Tutorials/LoRaWAN-classes.html [Accessed 14-March-2018].

[11] Available on: https://www.raspberrypi.org/products/pi-noir-camera-v2/ [Accessed 13-March-2018].

[12] Available on: https://www.britannica.com/science/pyroelectricity [Accessed 14-March-2018].

[13] Available on: https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/ [Accessed 14-March-2018].

[14] Available on: https://wisense.wordpress.com/2015/07/06/gas-leakage-module-using-the-mq-5-gas-sensor/ [Accessed 14-March-2018].

[15] Available on: https://www.elprocus.com/mq-135-alcohol-sensor-circuit-and-working/ [Accessed 13-March-2018].

[16] Available on: http://www.resistorguide.com/photoresistor/ [Accessed 14-March-2018].

[17] Available on: http://arduinolearning.com/code/arduino-hall-effect-sensor.php [Accessed 14-March-2018].

[18] Available on: https://www.arduino.cc/en/Guide/Introduction [Accessed 14-March-2018].

[19] Available on: https://www.arduino.cc/en/Products/Compare [Accessed 13-March-2018].

[20] Available on: http://www.microchip.com/wwwproducts/en/RN2483 [Accessed 14-March-2018].

[21] Available on: https://www.raspberrypi.org/downloads/ [Accessed 13-March-2018].

[22] Available on: https://raspberrypi.stackexchange.com/questions/534/definitive-list-of-operating-systems [Accessed 13-March-2018].

[23] Available on: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ [Accessed 12-March-2018].

[24] Available on: http://gdr-miv.fr/gdr/wp-content/uploads/import/MIFOBIO2016/ATELIERS2016/A59-2016.pdf [Accessed le 13-March-2018].

[25] Available on: https://www.omega.com/prodinfo/stepper$_motors.html[Accessed14 - March - 2018]$.

[26] Available on: https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development [Accessed 14-March-2018].

[27] Available on: https://rweather.github.io/arduinolibs/crypto.html [Accessed 14-March-2018].

[28] Available on: https://www.thethingsnetwork.org/docs/lorawan/security.html [Accessed 14-March-2018].

[29] Available on: https://www.webopedia.com/TERM/J/J2EE.html [Accessed 14-March-2018].

[30] Available on: http://www.pythonforbeginners.com/learn-python/what-is-python/ [Accessed 14-March-2018].

[31] Available on: https://www.guru99.com/what-is-php-first-php-program.html [Accessed 14-March-2018].

[32] Available on: https://nodejs.org/en/ [Accessed 14-March-2018].

[33] Available on: https://www.npmjs.com/ [Accessed 14-March-2018].

[34] Available on: https://www.wavemaker.com/learn/app-development/app-security/token-based-authentication/ [Accessed 14-March-2018].

[35] Available on: https://mosquitto.org/ [Accessed 14-March-2018].

[36] Available on: https://www.tutorialspoint.com/android/android$_{o}verview.htm[Accessed14-March-2018]$.

[37] Available on: https://cordova.apache.org/docs/en/3.0.0/guide/overview/index.html [Accessed 14-March-2018].

[38] Available on: https://dzone.com/articles/ionic-vs-react-native-which-framework-is-better-fo [Accessed 14-March-2018].

[39] Available on: https://www.arduino.cc/en/Guide/Environment [Accessed 14-March-2018].

[40] Available on: http://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf [Accessed 14-March-2018].

[41] Available on: https://www.python.org/doc/essays/blurb/ [Accessed 14-March-2018].

ملخّص

هذا المشروع يندرج ضمن مجال إنترنت الأشياء و يهدف إلى تأسيس نظام قائم على مراقبة المباني و التحكم بها عن بعد عن طريق .الهواتف الذكية بإستعمال لورا كتقنية لشبكة الاستشعار اللاسلكية التي تمتاز بالانخفاض في إستهلاك الطاقة

، انترنت الأشيا ء ،لورا، الطاقة

Résumé

Ce projet consiste à concevoir et réaliser un réseau de capteurs sans fil pour la surveillance à distance d'un bâtiment de faible consommation énergétique à base de la technologie LoRa.

Mots clés : IOT,LoRa,Energie,

Abstract

This project aims to design a wireless low-energy consuming sensor network for the remote monitoring of buildings based on Lora technology.

Keywords : IOT,LoRa,Energy,