INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Intelligent rainwater reuse system for irrigation

Maria Inês Soares de Matos dos Santos Pires

Master in, Telecommunications and Computer Engineering

Supervisor:
Prof. Dr. Pedro Joaquim Amaro Sebastião, Assistant Professor,
ISCTE-IUL

Co-Supervisor:
Prof. André Filipe Xavier da Glória, Invited Assistant,
ISCTE-IUL

November, 2020

Resumo

Os avanços tecnológicos na área da Internet das Coisas têm vindo a criar cada vez mais soluções na área da agricultura. Estas soluções são bastante importantes para a vida, uma vez que conduzem à poupança do recurso mais precioso, a água, sendo esta necessidade de poupar água uma preocupação mundial.

A dissertação propõe a criação de um sistema Internet das Coisas, baseado numa rede de sensores e actuadores interligados que monitorizam automaticamente a qualidade da água da chuva que é armazenada dentro de um tanque, a fim de ser utilizada para irrigação.

O principal objectivo é promover a sustentabilidade, reutilizando a água da chuva para sistemas de irrigação, em vez da água que normalmente está disponível para outras funções, tais como outras produções ou mesmo tarefas domésticas.

Foi desenvolvida uma aplicação móvel, para o Android, para que o utilizador possa controlar e monitorizar o seu sistema, em tempo real. Na aplicação é possível visualizar os dados que traduzem a qualidade da água inserida no tanque, bem como realizar algumas acções sobre os actuadores implementados, tais como iniciar/parar o sistema de rega e despejar a água em caso de má qualidade da água.

O sistema implementado traduz uma solução simples com um elevado nível de eficiência e testes e resultados obtidos, dentro do ambiente possível.

Palavras Chave: Internet das coisas, Sistema de rega, Rede sem fios de sensores e atuadores , ESP32, Sustentabilidade, Reutilização da água, Eficiência no uso da água.

Abstract

The technological advances in the area of Internet of Things have been creating more and more solutions in the area of agriculture. These solutions are quite important for life, as they lead to the saving of the most precious resource, water, being this need to save water a concern worldwide.

The dissertation proposes the creation of an Internet of Things system, based on a network of sensors and interconnected actuators that automatically monitors the quality of the rainwater that is stored inside a tank, in order to be used for irrigation.

The main objective is to promote sustainability by reusing rainwater for irrigation systems, instead of water that is usually available for other functions, such as other productions or even domestic tasks.

A mobile application was developed, for Android, so that the user can control and monitor his system, in real time. In the application it is possible to visualize the data that translate the quality of the water inserted in the tank, as well as perform some actions on the implemented actuators, such as start/stop the irrigation system and pour the water in case of poor water quality.

The implemented system translates a simple solution with a high level of efficiency and tests and results obtained, within the possible environment.

Keywords: Internet of Things, Irrigation System, Wireless Sensor and Actuator Network, ESP32, Sustainability, Water Reuse, Water Efficiency.

Contents

List of Figures

List of Tables

List of Acronyms

ADC	Analog-to-digital converter
AP	Access Point
BLE	Bluetooth Low Energy
BT	Bluetooth
CCTV	Closed-Circuit Television
CSS	Chirp Spread Spectrum
DAC	Digital-to-Analogic Converter
DC	Digital Converter
EEPROM	Electrically Erasable Programmable Read-Only Memory
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IoT	Internet of Things
I2C	Inter-Integrated Circuit
LAN	Local Area Network
LoRa	Long Range
LR-WPAN	Low-Rate Wireless Personal Area Network
MCU	Microcontroller Unit
MQTT	Message Queue Telemetry Transport
NB-IoT	Narrowband IoT
TSS	Total Suspended Solids
PWM	Pulse Width Modulation
P2P	Peer to Peer
RAM	Random Access Memory
ROM	Read-Only Memory

SDIO	Secure Digital Input Output
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SWAMP	Surface Water Ambient Monitoring Program
UART	Universal Asynchronous Receiver-Transmitter
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network

CHAPTER 1

Introduction

1.1. Motivation and Framework

The high world population growth forces the increasing production of food with the solution to humankind's food problems depending mainly on the use of agriculture and the correct amount of water for irrigation. The management of irrigation water should focus on increasing its productivity and efficiency of its use, defining a need to create better water management solutions. Water management not only needs to focus on efficiency, but also on helping prevent situations such as droughts and water scarcity, mainly in certain regions of the globe. Drought and water scarcity are distinct phenomena that affect the use of water in a very similar way, as an essential resource for life and as the basis for countless human activities [**1**].

The increased accessibility and adaptability of information and communication technologies has resulted in their wider use, even within agricultural dependent farms. There is an increased interest in the potential of internet technologies to support poverty reduction and the improvement of the living standards of people in rural areas.

Internet of Things (IoT) technologies can support precision agriculture, a form of agriculture whose goal is to maximise the return on investment in agriculture. Irrigation/ water detection/ soil detection sensors provide alerts to help protect a farmer's crop and transmit the information wirelessly to water reserve points about when to irrigate. This can be achieved by using data from various sensors combined with intelligent systems, that control not only where the water is released but also how much is needed [**2**].

IoT can provide real-time data collection on the availability and use of energy and water resources, increasing the management of resources, and can act upon the data collection to improve efficiency and sustainability of processes, decreasing waste and CO2 emissions [**3**].

1.2. Objectives

The objective of this thesis is the development of a smart system to collect rainwater in agricultural environments, that is able to control and monitor the water stored in that

container, or other wells or reserves. This system is responsible for collecting significant data from the maintenance point of view, through a network of low-cost wireless sensors, in order to assess whether the quality of rainwater is within the desired parameters, so that it can be reused for irrigation.

This work proposes a new system capable of monitoring and controlling water properties (pH, water level, temperature, turbidity) and if it is raining, as well as control the mechanics to gather the water when is raining and feeding this water into the irrigation supply.

The data obtained from the sensors will be stored and processed in order to keep the user informed about the parameters considered relevant for water quality, allowing not only to check the conditions in real-time, through a mobile app, but also remotely control the entire system.

1.3. Structure of the Dissertation

The dissertation is composed of six chapters. After this introductory chapter, chapter 2 considers the state of the art, introducing projects related to the proposed theme as well as the explanation of some choices in terms of protocols, hardware and even software. Chapter 3 presents in more detail the system architecture, specifically, the sensors used and their connections, how and what data is collected from them, what implementations have been made, as well as the application developed and the functionalities of the system. Chapter 4 presents the tests and results resulting from the system application. The fifth chapter, a demonstration of the operation of the prototype in a test environment, where the results are discussed, is made. And, finally in chapter 5 are presented the main conclusions of the dissertation presented, with aspects to improve and future work.

At the end two annexes are presented, a technical manual to help the user understand the implementation performed and a list where the characteristics of the components used in the prototype are specified.

1.4. Main Contributions

This dissertation includes the following contributions:

- The design and implementation of a network of wireless sensors and actuators;
- Development of a mobile application for analyzing and controlling an irrigation system.

CHAPTER 2

State of Art

This chapter consists of a review of possible approaches for the development of the proposed system, such as the current IoT relevance and its application in the area of agriculture, how wireless sensor networks help with the proliferation of IoT and how they are composed, including the most common communication protocols used, and the description of hardware and software necessary for the implementation of an intelligent system. Alongside each topic, some related work is presented, to understand what is the research status on those topics and how our solution will differ.

2.1. Internet of Things

The technology has been improving in the last decades, particularly in the interconnections of computers, devices, people and objects.

The IoT is the evolution of the current Internet in a ubiquitous network, composed of interconnected devices that are capable of transmitting data [**4**]. Devices can be varied, such as cars, Closed-Circuit Television (CCTV) cameras, appliances, etc., and the information collected from these IoT devices can be accessed in any place with ease, as Internet connection become widely available anywhere and with almost any device.

This technology is capable of collecting and controlling data from the physical world with the support of various devices deployed. These devices have the ability to perceive, compute, execute and communicate between the user and things.

This concept consists of data acquisition, processing, transmission and storage, which provides a more robust and efficient communication system and it has been growing and evolving. To continue to do so, it is necessary to strengthen the technologies that already exist, as well as the creation of new ones [**5**]. This, together with the cloud, large volume of data, analysis and machine learning is able to assist in creating opportunities that have an influence on our daily lives.

IoT technologies can have multiple applications as they have components capable of providing important information for the operation, conditions and performance of any environment that can be remotely controlled. So far several sectors have adopted these technologies, for example, health, transport, retail, buildings and agriculture [**6**].

Agriculture is one of the major markets in IoT, not only to keep track of every detail in the fields, that helps with the proliferation of precision agriculture, but can also take an important role in the resource management, mainly in water, as irrigation system in agriculture account for 70% of the world's fresh water is used in these systems, from agricultural fields to gardens, and the problem is that around 30% of it is potentially being wasted due to lack of monitoring or environmental issues [**7**].

2.1.1. IoT for Agriculture Irrigation Systems

Climate change requires special attention because it is one of the most important and worrying issues in the world today. It is contributing to the unavailability of food since it impairs its quality, due to the decreased availability of water resources, changes in rainfall patterns and extreme weather events, all leading to a decrease in agricultural production. Since agriculture is also the area that consumes the largest percentage of water resources, there is a greater concern to make it sustainable.

The practice of agriculture has both positive and negative consequences. Positive on the environment, since it is able to capture the greenhouse gases that are inserted into the land of crops, or even reduce the risk of flooding, given the use of specific agricultural practices for this purpose. The negative impacts are related to pollution and degradation of air, water and soil quality.

The concept of IoT has been increasingly relevant and there are more and more solutions that fit into this area, leading to greater control over the environment in which we live. IoT devices have the ability to collect and distribute data in real time, such as the measurement of a river's water level or the value of moisture measured on agricultural land. These types of measurements help to develop greater control and management of the resources of irrigation systems. There are also airborne drone solutions that play a role in monitoring agricultural land, providing valuable information about it. Technological solutions are increasing day by day to be integrated into your smart agricultural products in order to lower the overall cost by increasing the quality and quantity of harvest systems [**8**].

Research in this area reveal that a high percentage of the water used in irrigation systems is wasted due to little supervision and real-time control.

In [**7**] presents a new way of managing water in irrigation systems that plays the same role as human intervention, using wireless network sensors, with the objective of improving the efficiency of the irrigation process in agriculture, allowing monetary and water resource

savings, is presented. This is done by collecting data in real time, allowing the system to determine the precise water needed for irrigation. This system used a Wireless Sensor Network (WSN) based architecture, where the implemented sensors communicated using LoRa with each other and, Message Queue Telemetry Transport (MQTT), via Wireless Fidelity (Wi-Fi), with the server. The Hardware was based on an ESP32 board, an RFM95W transceiver Long Range (LoRa) radio module and 2 sensors, one that measured the temperature and humidity of the air and one that measured the amount of water contained in the field. This solution was implemented in a garden for 2 months and its tests resulted in a 34% improvement in irrigation system performance. The presented system creates a good solution and is a good source of information for the proposal presented, and will be a good complement to the solution that this thesis will develop.

In [**9**] an analysis of an architecture, platform and Surface Water Ambient Monitoring Program (SWAMP) system, based on IoT for water management, is presented with the objective of creating a precision irrigation in agriculture and it includes a FIWARE[**10**] components performance analysis. The results presented on paper show that the platform can provide adequate performance, but it lacks some specific configurations that, using less computational resources, have the ability to scale the system.

In [**11**] a proposal for a system implemented with sensors which data is analyzed to calculate the amount of water needed for irrigation, is presented. The hardware used includes a Raspberry Model B Pi 3, a microchip IC3208, a sensor that measures soil moisture, a sensor that measures temperature, a LM358, a relay and a bell. Through the measurement obtained by the temperature sensor and taking into account the calibration value, the exact drying level of the soil is measured and this way, an automatic message is sent for the engine to be switched off immediately. The hardware was fully integrated and experimental tests we made, proving that the proposal is a solution to irrigation problems, helping to improve the land and production on it. The real-time information provided will help the owners of the land to make the appropriate decisions.

Our solution works as a complement to these applications, as our proposal presents an implementation of a mobile application, so that the owner of the agricultural system can control it, offers visualization of the sensor data as well as some interactive part, where he can control the beginning and end of the irrigation system as well as the dumping of the tank, according to the water quality that he will visualize and analyze.

As for intelligent irrigation software, not only academic research exists, but there are also already some solutions on the market that allow people to be more involved in irrigation activities and to optimize water use. Depending on several factors such as weather forecasts and soil moisture levels it is possible for these programs to communicate with irrigation controls, and give information to the system, so it knows when the crops needs more water. These programs make it possible to irrigate throughout the entire life cycle of a crop, from seeding to harvesting season [**12**].

Applications such as GreenIQ, an IoT platform for efficient irrigation, able to control irrigation scheduling based on the current and expected time, saving up to 50% on garden water consumption. This system is connected to the Internet, via Wi-Fi or 3G, and can be controlled remotely. The control is done through the data provided by the soil sensors that control the moisture balance [**13**].

The Rachio mobile application is compared to a personal assistant who takes care of and waters the grass. In order to avoid wasting water, customized watering schedules are created with the support of weather monitoring [**14**].

For the objectives that the proposal presents, there are already some hardware and software solutions, as described above. Of these, none had as base what the system proposes, the implementation of a network of sensors and actuators that control the irrigation system in an intelligent way, making real-time analysis of the information collected by the implemented sensors and, allowing the user to analyze this data and also perform actions on their system. All through the use of a WSN architecture, with the use of communication protocol that allows communication between the entire system and user, with low consumption and energy power.

2.1.2. IoT for water reusage

The agricultural sector is the largest consumer of water in the world, which leads to greater concern about the existing water crisis. These concerns increases the studies and practices of water reuse which, with the help of technological advances, makes it possible to obtain water with a high level of quality [**15**].

In [**16**] is presented a short summary about the reuse of water in Portugal as a solution to the existing limitations in risk assessment. It describes the importance of the water reuse and also the risks, both to the environment and health, because of disinfection products and pathogenic microorganisms. This concern has led several countries to impose standards for water reuse, introducing quality standards. Some projects in the area of

water reuse have been developed such as in Algarve, Portugal, where one of the best examples is the plant treatment system that irrigates a golf course and at the same time maintains an ecosystem, where approximately 14,500 m^3 of tertiary effluents are used every day to irrigate the course and the rest is used to maintain a lagoon, which is an important nesting area for protected bird species. . There are also other projects, but on a smaller scale, such as water from the drainage of berry production is reused in the irrigation of other crops, where it is possible to reduce about 15% of total irrigation needs during the dry season.

In [**17**] a software application to monitor and control the amount of water, from the municipality is presented. This system was developed with the objective to guarantee a minimum waste of water, in an autonomous way. In this, the rain water that reaches the top of the buildings, is stored in the facilities of the same, where is then funneled to a sand filter, where the dust, among others, is filtered, leaving the filtered water in a tank that when is full, is poured into a reservoir. It also presents an application software with the purpose of controlling the water supply for each house of the municipality with the objective of, when a user looks for water supply, it refers in the application the number of people to supply and the municipality will supply according to that number, for that specific house. If they need more water than the amount calculated, then they will have to pay. The findings show that the solution is beneficial for families, where water is a limited resource. And the software shows that it controls the demand for water, decreasing its waste in a smart way.

The proposal presented is a complement to these two works, to the extent that, proposes to reuse rainwater in irrigation, so as not to waste the water from the reserve tank that is used for other consummations by the user. This is achieved thanks to the data collected by the implemented sensors that, when it rains, open the lid of the tank so that the water is stored. This water is analyzed so that its quality is maintained in order to water the land with good water. The advantage of this system is also that the user has access, through the mobile application, to the data that the application collects and, in this way, is able to view each specific data that measures the quality of the water and start or stop, if necessary, the system. This way the user can control the system and reuse the rainwater for it in a simple and automatic way.

2.2. Wireless Sensor Network

A WSN can be defined as a network of devices that can communicate, through wireless links, the data that those devices collect. The data is routed through multiple nodes, all leading to a gateway, that is connected to other networks such as the Internet. This network type, along with wireless communication, provides a huge variety of possible applications. These networks are designed to save energy because this is the most difficult resource to find where it is installed, so they must be reliable, scalable and robust.

These networks have the ability to implement a large set of small nodes, that jointly, allow the assembly and configuration of a single goal. This type of networks have several applications, such as: surveillance of a specific areas, environmental control, accurate and intelligent agriculture, health care, traffic control, and object tracking [**18**]. WSN have the ability to adapt and actively respond to topological changes in their environment. The networks are usually composed of four modules: the Sensor module, processor, power supply and transceiver. Typically, in WSN, the nodes that constitute it are capable of organizing themselves without control, translating their ad hoc configuration [**18**].

In short, this network is described as a network of nodes that, when they work together, enable the control of the environment, enabling the interaction between the environment and people, and that is why WSN is so important in IoT systems.

For WSN to work properly, they can be divided into 3 large parts: Sensing, Communications and Cloud Based Technologies.

2.2.1. Sensing

A sensor is a component whose main function is to collect information from the environment in which it is inserted. They connect directly or indirectly to IoT systems, as they can be act upon them or processes for future actions. In WSN, the information that the sensor collects is forwarded by the nodes that compose it, until it is received by other networks, applications or services.

The way that nodes have to gather and transmit information, can vary depending on the application or main goal, as it needs to adapted to each scenario.

Although the node constitution can be modified for each specific purpose, the core of a node has specific needs. One of the most common type of nodes, is the sensor node, which is integrated at the lowest level of the network, as it has only one or more sensor and a communication mechanism, attached to a microcontroller, allowing it to pass the data to another node in the network.

Another common node is the gateway node, responsible for receiving the date from the sensor nodes and to send them to the server, being the only node with Internet connectivity. Other nodes can include more hardware components, for example to control actuators or do more complex computational activities, such as data analysis.

As such, a big part of the node constitution is the controlling platform used, or the microcontroller, with several being available for the construction of embedded systems and electronic projects.

2.2.1.1. ***Controlling Platforms.*** To automate the proposed system, the use of a control platform is required. This type of platform is able to develop automation methods and integrate them in several areas, incorporated in small and low cost components.

In this section the Arduino, Figure 2.1, and ESP32, Figure 2.2, control platforms will be analyzed, two platforms that aim to allow programmers to create their electronic projects easily and with little monetary cost.

Arduino

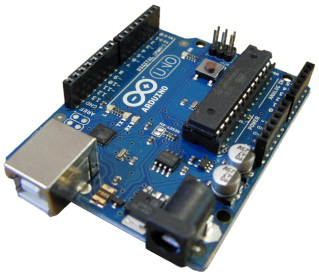


FIGURE 2.1. Arduino platform

Arduino, Figure 2.1, is an open-source, user-friendly electronic hardware platform that is capable of signal acquisition, from analog sensors or other digital I/O. This board is designed with an Atmel AVR microcontroller with built-in input/output support and uses the Arduino programming language, which is essentially C/C++ and the Arduino Software (IDE), based on Processing [**19**].

ESP32

The ESP32-WROOM-32, Figure 2.2, is a Wi-Fi+BT+BLE MCU module designed for a variety of applications, from low power sensor networks to the most demanding tasks

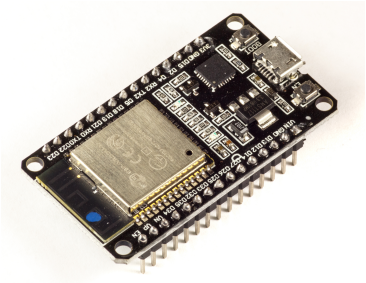


FIGURE 2.2. ESP32-WROOM-32 platform

such as voice encoding, MP3 decoding and music streaming. The design of this module is open-source and as ESP32 has Wi-Fi and Bluetooth capabilities, it makes this chip flexible for IoT project development [**20**]. This is considered an improvement over the popular ESP8266 that is usually used in IoT projects, and for that reason the ESP8266 is not in this study.

2.2.1.2. ***Remarks.*** Table 2.1 presents the key characteristics of the two analysed boards and on close comparison it is possible to verify that ESP32 is more robust regarding hardware, as well as includes other features that can help achieve the proposed solution for this thesis [**21**]. ESP32 is a simpler device and it is more robust, making it easier to set up, and as such it will be the used controlling platform.

2.2.2. Communications

The objective of these wireless nodes is to collect information from a field or area and send that data to other nodes within the network, and this is why communication in WSN is a very important component.

As mentioned in the Section 2.1, one of the main components of an IoT system is communication and, for that, there is a wide variety of protocols. Since the proposed system is a wireless sensor network, a protocol of this type will be used. The most common protocols for this type of communication will be presented below.

2.2.2.1. ***Intra-network communications.*** A network that is used for sharing data and where access is restricted to people from abroad, is called an intranet network. As the name implies, "intra" means "internal", and therefore this is a private network, where those used to access the content it shares need to have authorization.

The following sections will summarize some of the protocols used in IoT for this type of network.

TABLE 2.1. Arduino and ESP32 specifications

Description	Arduino UNO	ESP32
Power source	5V	2,2V ~ 3,3V DC
Regulated Input (Vin)	7 ~ 12V	5 ~ 9V
Current Consumption	media of 15mA	media of 80mA
Operating Temperature	-40°C ~ +85°C	-40°C ~ +85°C
Processor	AVR 8-bit RISC	Xtensa Dual-Core 32-bit LX6
Operating Frequency	0 ~ 16 MHz	80MHz ~ 240MHz
FLASH Memory	32kB	4MB
RAM/SRAM Memory	2kB	520kB
ROM/EEPROM Memory	4kB	448kB
I/O Pins	23 pins with 6 PWM	34 pins with 16 PWM
ADC converters	6 ADC with 10-bit of resolution (1024 bits)	18 ADC with 12-bit of resolution (4096 bits)
DAC converters	None	2 DAC with 8-bit of resolution (256 bits)
Wi-Fi	Only with Shield	2,4 GHz, 802.11 b/g/n/e/i (802.11n to 150 Mbps)
Bluetooth	Doesn't have	Bluetooth Low Energy v4.2 (BLE)
Timers	3 Timers, one with 16-bit and two with 8bit	4 Timers with 64-bit
Module Interfaces	I2C, SPI, UART and LED PWM	SPI, SDIO, LED PWM, Motor PWM, I2S and IR
Embedded Sensors	Temperature and Capacitive Touch	Temperature (some versions), Hall effect and Capacitive Touch

LoRa

LoRa is a Low-power wide-area network protocol that offers an adaptive data rate modulation technology, which allows long range, adaptive communication with low power consumption and design [**22**].

LoRa works in the ISM 433-, 868- or 915-MHz bands, and its transmission payload can vary between 2-255 octets, and the data rate can reach 50 kbps when channel aggregation is used. A network of this kind usually translates a star topology [**23**]. Advantages and disadvantages of LoRa:

- Advantages - Long communication range and low power consumption;
- Disadvantages - Max bit rate is 50kpbs and very limited number of nodes.

Bluetooth Low Energy

BLE is a short-range wireless communication technology that provides low cost and power,

and is an ultra-low power version of classic Bluetooth. The classic Bluetooth has about 79 channels, each with a width of 1 MHz, BLE was developed to work in the 2.4 GHz ISM band and can use about 40 channels, with a width of 2 MHz each. These features make BLE more beneficial when used in short-range communication technologies. This technology is usually used for sensors and low power components, that require long battery life [**24**]. Advantages and disadvantages of BLE:

- Advantages - Master is able to control many slaves, is widely supported and has low power consumption;
- Disadvantages - Supports only star topology, has very limited number of nodes and low communication range.

ESP-NOW

ESP-NOW is a protocol that allows direct communication between several devices without a Wi-Fi connection. It is similar to the protocol that is commonly used in wireless mice and keyboards, which translates into low-power, 2.4GHz wireless connections [**25**].

These devices need to pair first to automatically initiate communication when re-booted, making the connection secure and peer-to-peer, requiring no handshake [**26**]. Advantages and disadvantages of ESP-NOW:

- Advantages - Faster data transmission since it does not need to connect to a Wi-Fi Access Point, supports a maximum of 20 nodes and does not have overhead;
- Disadvantages - Limited data transfer rate, cannot use Wi-Fi while using this protocol and limit of 250 byted of data package.

ZigBee

ZigBee, IEEE 802.15.4, defines specifications for Low-Rate Wireless Personal Area Network (LR-WPAN) to support simple devices that consume the minimum amount of energy and work within 10m. ZigBee provides a self-organized, multi-function, reliable network with long battery life[**27**]. Advantages and disadvantages of ZigBee:

- Advantages - Low power consumption, supports star, tree and P2P topologies and with one coordinator can control many slaves;
- Disadvantages - Additional equipment is required and is not compatible with other communication protocols.

2.2.2.2. *Network to server communications.* To send information to the cloud, the so-called internet network is used. Its main function is to connect different computer networks simultaneously and it is a public network, which means it has no restrictions and all users have access to it.

The following will be highlighted some technologies of this type and some summary information about them.

Wi-Fi

Wireless Fidelity (Wi-Fi), known by the IEEE 802.1 la/b/g standards for Wireless Local Area Network (WLAN), allows users to surf the Internet in broadband when connected to an Access Point (AP) or in ad hoc mode. The architecture of the standard consists of the interaction of several components in order to provide a wireless Local Area Network (LAN) that supports the mobility of stations transparently to the top layers [**27**].

The transmission of this type of network is done through radio frequency signals, which propagate through the air and can cover areas in the hundreds of meters. Advantages and disadvantages of Wi-Fi:

- Advantages - Able to penetrate walls and other type of obstacles on the way and has a simple process of adding and removing devices from network;
- Disadvantages - High energy consumption and radio waves may interfere with other equipment.

NB-IoT

NarrowBand-Internet of Things (NB-IoT) is a technology based on wide area low power (LPWA) standards, developed by 3GPP. This technology allows a wide range of services and cellulating devices.

This is a wireless communication technology that allows the connection of multiple devices, low cost, long battery life and high connection density, improving the power consumption of user devices, particularly in deep coverage. NB-IoT uses a subset of the LTE standard, an OFDM modification for downlink communications and SC-FDMA for uplink communications and has a single bandwidth of 200kHz [**28**].

This technology is quite advanced for IoT systems, given the need for these to maintain constant communication and an NB-IoT has no duty cycle limitation operating in the licensed spectrum [**28**].

Advantages and disadvantages NB-IoT:

- Advantages - Excellent indoor coverage, supports a large number of connections and low power consumption;
- Disadvantages - Low data range.

SigFox

SigFox is a Low-power wide-area network that is composed by a several devices connected to the Internet. This network has a function to send data through the SigFox network to a SigFox base gateway. This gateway uses three channels to detect, demodulate and report messages to the SigFox cloud every 10 minutes [**29**].

This technology contributes to a long range communications network, with low power and output and high protection against environmental interference, making data transmission effective [**30**]. Advantages and disadvantages SigFox:

- Advantages - Long data range and low power consumption;
- Disadvantages - Supports one way communication without acknowledgement and cannot be used for high rate applications.

LoRaWAN

LoRaWAN defines the communication protocol for the network, as well as the system architecture, while the LoRa physical layer is used to create a long-range communication link [**31**].

This long range link is achieved thanks to the use of Chirp Spread Spectrum (CSS) modulation, by LoRa. It uses chirps of frequency with linear variation to code the information over time. Advantages and disadvantages of LoraWAN:

- Advantages - Low power consumption, long communication range, low complexity and secure connection provided by AES encryption;
- Disadvantages - A packet can only have a maxi-mum size of 55 bytes.

2.2.2.3. ***Remarks.*** Table 2.2 shows the key characteristics for all the presented protocols.

Since the proposed system can be composed of multiple nodes, that need to communicate among them but also with the server, both types of communication can be used in the final system.

For intra-network communication LoRa appears as a clear best solution, with its long range and low-power consumption, it fits the needs of our system.

The communication protocol chosen for network to server communication, was Wi-Fi since it is already integrated in ESP32. In an environment with lack of Wi-Fi coverage, such as deep agricultural sites, LoRa or NB-IoT network could be used.

TABLE 2.2. Communications remarks

Feature	Wi-Fi	BLE	ZigBee	LoRa	SigFox	NB-IoT
Based Data Rate [kbps]	$11x10^3$	$1x10^3$	250	110	$1x10^{-3}$	250
Frequency [GHz]	2.4	2.4	2.4	0.868	0.868	1.8
Range [m]	100	100	100	5000	10000	1000
Nodes	32	7	65540	15000	-	-
Power Consumption [mA]	100-350	1-35	1-10	1-10	1-10	10-100
Price per message [€]	-	-	-	-	16.13€/ year[1]	0.10€/ Mb[2]
Availability	High	Low	Low	Low	Medium	Medium
Complexity	High	Medium	Medium	Low	Low	Medium

2.3. Cloud Based Technologies

The cloud can be translated as a large network of remote servers, dispersed around the world, which are connected to each other and operating as a single computer. This way, from a remote location, it is possible to access a service or software platform anywhere, with only a internet connection.

This is a popular and sought after option by people and companies as it allows for cost savings, increased productivity, efficiency, performance and security.

These resources, able to offer, through the Internet, various services, have understood tools such as servers, databases, networks and software. This is possible because the files are not stored locally, but in a cloud-based storage, with the opportunity to store them in a remote database. And, this way, not only the device has access to the web, but also access to data and software programs [**32**]. The cloud database, which does the functions of a traditional database, but includes the benefits and agility that the cloud offers, where users are able to own databases in their companies, for example, without the need for specific hardware.

This type of services is usually accessed through a web browser, such as Google Chrome or a mobile application as cloud applications can be compatible with Android and iOS, making them more attractive.

[1]Price per device (140 messages maximum per day)
[2]Using the Things Mobile network, IoT system provider for connectivity around the world

Given the integration of IoT devices in all areas and the valuable data they collect, there is a need for a control dashboard. A IoT dashboard is web-based software that gives the user control over their ecosystem, allowing them to review the data collected by IoT devices in real time.

Today there are several technology solutions for creating complete IoT applications, where these dashboards are inserted, such as IBM Cloud plataforms, Microsoft Azure, Google Cloud, etc.

This type of software allows the combination of data sets and programming of actions based on rules, allowing the control and monitoring of IoT systems in an efficient way. As, for example, in the healthcare area, doctors are able to monitor a patient's vital data, without the need to visit, in real time, thanks to these platforms [**33**].

Despite these technological advances, it is still common to create applications for specific projects, mainly for privacy and security reasons, given the value of the data that each system possesses and so as not to compromise the users' data. These proprietary applications are mostly developed on iOS or Android systems.

These two systems, Android is the most used given that the largest number of devices in the world integrate this operating system, being a total of 74.43% compared to 24.99% of iOS [**34**]. Other advantages of this operating system are the lower cost in development and the greater ease of learning, since the programming language is java.

CHAPTER 3

System Architecture

This chapter describes the architecture of the system and its components, as represented in Figure 3.1.

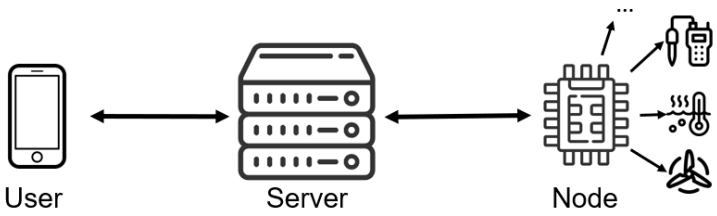


FIGURE 3.1. System architecture

This work intends to develop a system of analysis of the quality of water stored in a tank, resulting from rain, for its reuse in irrigation. This system uses a set of specific sensors that, when analyzed together, reveal good or bad water quality.

These sensors are placed inside the tank and the user, having access to the resulting data, can check the quality and perform actions that he finds necessary, such as dumping the tank, if the quality is not the one he idealizes.

Figure 3.2 presents an illustration of the system's implementation in real environment:

To better describe the system architecture, this section was divided into three subsections:

- Data collection - where the hardware, connections, actuators and sensors used are described, as well as their functions;
- Data analyses - where the scripts that control the entire system are explained;
- Data visualization - where the android mobile application developed for the user to visualize and control the system is described;

3.1. Data Collection and System Control

The proposed work is based on a network of sensors and actuators connected wirelessly, which form an architecture similar to a Wireless Sensor and Actuator Network (WSAN),

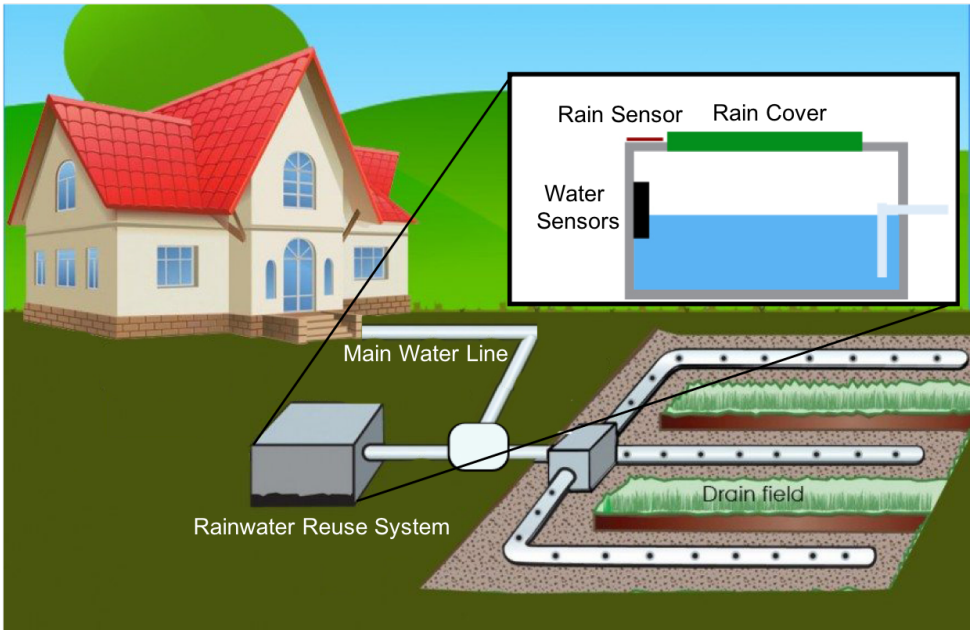


FIGURE 3.2. Illustration of implementation in real environment

able to collect the information measured by the sensors, perform actions, and connecting to a cloud server.

In this system, there is only one node, the controller node, that combines the functionalities of a sensor, gateway and actuator node. Since all the information is gathered inside the tank, and the system actuators were also in there, there was no need to split the hardware into multiple nodes, creating a lower cost, more reliable and robust system.

In order to make the implemented system as efficient as possible, with low cost and power consumption, the best microcontrollers and communication modules, as studied and described in Chapter 2, were used.

3.1.1. Controller Node

The controller node englobes all the nodes functions. This node is responsible for keeping the network connected and in communication with the server, gathering sensor data and forwarding it to the server, as well as receiving the actions that the server sends and perform them in the respective actuators. All of these transactions take place via MQTT messages taking advantage of the ESP32 integrated WiFi module.

This Wi-Fi connection can be replaced with a NB-IoT ceular connection in areas where the tank is more remote. For that, the SIM7000E module can be attached to the ESP32, to provide 2G and NB-IoT connectivity to the controller node.

Figure 3.3 how the node is divided by the type of components:

- sensors - temperature, water level, turbidity, pH and rain;

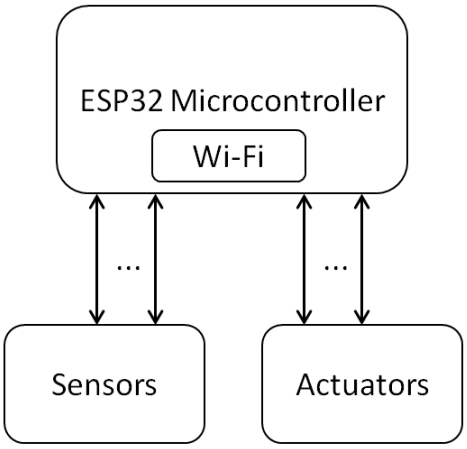


FIGURE 3.3. Controller node

- actuators - motor servo and water pump;

3.1.2. Sensors

To detect, measure and record data from the proposed water reuse system, a set of sensors was connected to the controller node, in order to evaluate, in real time, not only the conditions of the water stored, but also to control the collection of rain water, when needed.

To create a trully IoT system all these data is then forward from the network to the Internet, for user visualization and a data processing.

Next, all the sensors used are described in detail, with technical specifications available in Appendix A.

3.1.2.1. ***Analog pH Sensor.*** To monitor the water quality, it is important to measure its pH, by measuring the activity of the hydrogen ions in the solutions. The measured value translates the acidity or alkalinity of a solution on a scale of 0 to 14. pH value 7 is considered neutral, being acidic when lower value and higher values being caustic or alkaline.

An analog pH meter will be used, Figure 3.4, specially designed for Arduino controllers and has a simple, convenient and practical connection and features. The pH accepted in irrigation water systems is generally between 5.5 and 7.5 [**35**].

3.1.2.2. ***Capacitive Sensor.*** It's a simple breakout, composed by multiple pads that work like probes, that uses the conductivity between them for measuring water content, such as rain or water level, Figure 3.5. This Rain Water Level Sensor is a very simple, cost-effective and high level/capacity sensor, achieved through a series of parallel wires

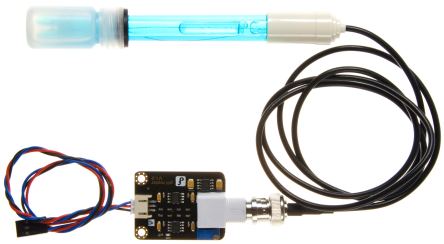


FIGURE 3.4. pH sensor

that present samples of measured droplets/volume of water, in order to assess the water level.

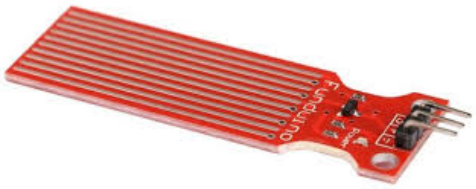


FIGURE 3.5. Capacitive sensor

3.1.2.3. ***Analog Turbidity Sensor.*** To detect the quality of the water reserved, a turbidity sensor, Figure 3.6, will be used to measure turbidity levels using light to detect suspended particles in the water by measuring the transmittance of light and the dispersion rate, which varies with the amount of TSS in the water. TSS translates the portion of fine particles suspended in water. Similar to turbidity, it provides the real value of the weight of the particles, usually in mg/l, for a given sample volume and, as it increases, the level of liquid turbidity increases. This liquid sensor provides both analog and digital signal output modes. The threshold is adjustable when in digital signal mode.

3.1.2.4. ***Waterproof DS18B20 digital temperature sensor.*** This digital sealed temperature component, Figure 3.7, has the ability to measure the value of precise temperatures, in humid environments, with a simple communication one-wire bus protocol that, to give information to the microprocessor, uses a single line of data. It offers temperature readings from 9 to 12 bits.

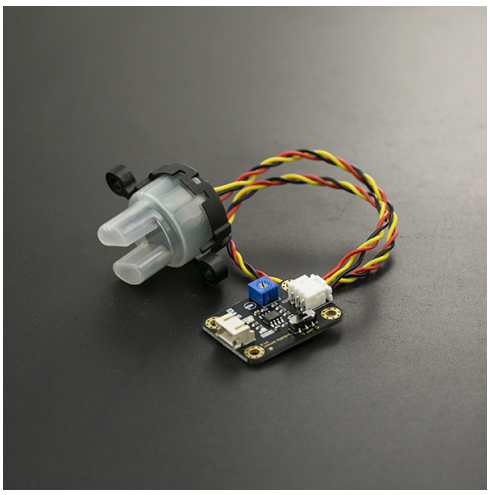


FIGURE 3.6. Turbidity sensor

In this system, this sensor will measure the water tank temperature. The measurement of water temperature is important given its chemical affluence, which increases according to the increase in temperature, and decreases the oxygen level in its composition, becoming inappropriate for, for example, the survival of aquatic species.

Irrigation water temperature for a good quality must be in the range [13ºC,30ºC].[**36**].

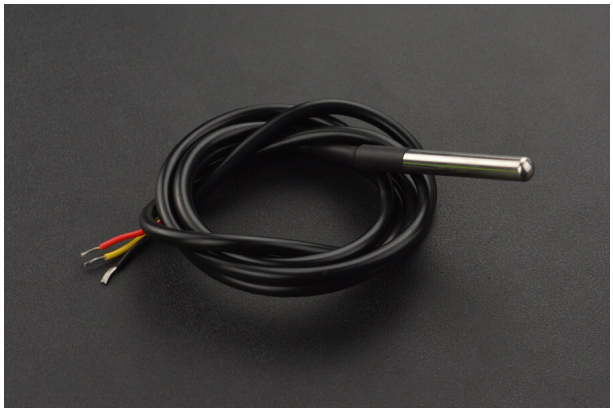


FIGURE 3.7. Digital temperature sensor

3.1.2.5. ***Photoelectric liquid level sensor.*** To measure the water level of the tank and check if it is sufficient to be reused for irrigation, a photoelectric liquid level sensor, Figure 3.8, will be used which works according to traditional optical principles. The advantages are high sensitivity and no mechanical parts are required. The corrosion resistant probe is easily mounted and can withstand high temperatures and high pressures.

3.1.3. Actuators

An actuator is a component that generates and controls the movement of a system or mechanism, converting hydraulic, electrical or pneumatic energy into mechanical energy.

FIGURE 3.8. Liquid level sensor

They need an energy source and a control signal in order to transform the incoming energy into output movements.

In order to control the tank features, in real time, actuators were attached to the controller node, being these act upon user interaction or automatic system control, due to specific sensor data.

Next, all the actuators used are described in detail, with technical specifications available in Appendix A

3.1.3.1. ***Servo motor.*** The servo motor, Figure 3.9, is a machine that presents movement proportional to a command, that is, it receives a control signal that checks the current position to control its movement to the final position. These motors can rotate about 180^{o} or 360^{o} in some models and are accurate as to their position. The size, material and speed of the motor are key features for servo motor specification and in this system, this component will integrate the tank's cover, and will open and close, according to the result of the rest of the measurements, to fill or not the rain tank.

3.1.3.2. ***Water Pump.*** The pumps take advantage of direct current from the motor, battery, or solar energy to move the fluid in various ways. This pump is immersible and can be used to water plants, or make a fountain or cascade. It works quietly with the sound level below 30db. The pump, Figure 3.10, has a filter inside, as well as a suction cup that can help glue it to smooth, tight surfaces.

For the use of the water pump it was necessary to implement an electric relay switch. This component can be switched on or off, and lets the current through or not, depending

FIGURE 3.9. Motor servo

on that state and was used to make the motor run, or stop, when it receives a signal from ESP32.

FIGURE 3.10. Immersible pump water tube

3.1.4. Message Exchange

As described in Section 2.2, and since the system is composed with only node node and intra-node communication is needed, the wireless communication protocol chosen to connect the network to the server was Wi-Fi, since the ESP32 module used incorporates it, and that is how the WSN controller node and server communicate.

To ease the message exchange between the node and the server, via Wi-Fi connection, the MQTT protocol was used, since it is an easy protocol to implement in software and fast in data transmission and it is more suitable when it is referred to IoT development.

The protocol is based on asynchronous messages system that follow the publish-subscribe model. MQTT request triggers an action for the receiver and the response contains the result of this action [**37**].

To transfer data to the server, MQTT protocol was adapted, and so were the topics to communicate to each direction, for the MQTT broker:

- "teste/maria/out" - topic that affects the messages going from the controller node to the server and/or mobile application;
- "teste/maria/in" - messages that go in the opposite way, that is, on the way to the node

The first topic, "teste/maria/out", is subscribed by the node, and is regularly listening to new messages, to control the actuators that, when the user decides to select a specific actuator action, upon message received, performs the action in real time. In this case, the implemented code extracts from the message the value and the indicated actuator and, according to the value, acts in a specific way:

- Pour tank - when this message arrives, the value with it is always 1, since it indicates that the tank must be poured. Upon this message, the controller node will activate the water pump, and the water will be pour out of the tank, until it is empty.
- Irrigation system - this message is combined with a value that indicates either the irrigation system must start or stop. And, accordingly, the controller node will indicate that to the pump, that will start or stop pumping.

3.2. Data Analysis

The user is able to monitorize and control his system through the real time data available in the mobile application and for this purpose, several scripts were implemented. The main function of this scripts is to receive data from the controller node and forward it to the application.

This section is divided in three parts: the description of the flux that starts in the controller node, where the data from the sensors is collected, and sent to the server to give specific content to the mobile application, the description of the flux that starts in the mobile application, where the user selects specific actions for the actuators, and the MySQL Server function in both fluxes.

3.2.1. Controller Node

The systems starts in the controller node, where the data is collected, as explained in the Section 3.1.1.

Figure 3.11 translates what the system does in terms of data analysis, from the controller node point of view. It starts by collecting the data from the sensors and analyze

them to, if necessary, performed actions, followed by passing the information to the user, that is, to the mobile application, where it will be visualize.

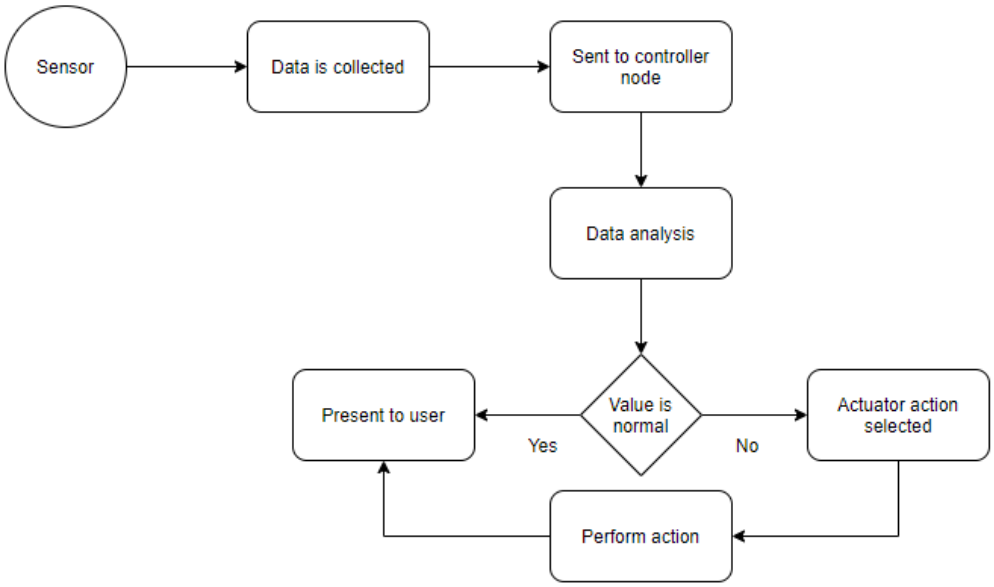


FIGURE 3.11. Communication from controller node

It was created a script named mqtt.ino where the information collected by the ESP32 from the sensors is first analysed for internal and automatic system control, such as open the tank cover or sump water, with the following described methods:

- checkWaterQuality() - checks if the water is in good or bad condition and, according to the result, discharges the tank contents. This result is obtained by analyzing the data resulting from the sensors, that is, if the turbidity value is 1, the pH value is not close to 7 (5.5 and 7.5 [**35**]), and if the temperature is not between 13^{o}C and 30^{o}C, then the water is considered of bad quality.
- checkWeather() - this method checks if there is a sufficient humidity/rainfall value to be stored in the tank. If there is and, if the tank is not full, it is opened for the water to be stored. If there is no rain, the tank is closed so that the water does not evaporate.

Next, the data is sent to the server and, in order for it to reach the mobile application, to be shown to the user, it is sent via MQTT, as already described, using the PubSubClient MQTT library to allow a connection with the broker client was made.

When the connection is successfully made, a message is published to the client, that consists on the name of the sensor and its value and is made as follows: "client.publish(output_topic, message);". The client being the broker to which the link is created, the output_topic is

referring to "teste/maria/out", used by the broker to communicate with the client and share the published message, that indicated the sensor and its value, for example "Temperature,25".

3.2.2. Storage

The server is responsible for storing the information collected by the sensors sent by the ESP32, allowing for future analysis of this data, as well as the information about the registered users, that will have access to the application. For that it was created a remote server, on a MySQL database platform, creating a connection between the user and the hardware elements.

Two tables were created to store all the necessary information's, as can be seen in Figure 3.12. These two tables are connected as the Figure 3.12 presents, including details of each table column.

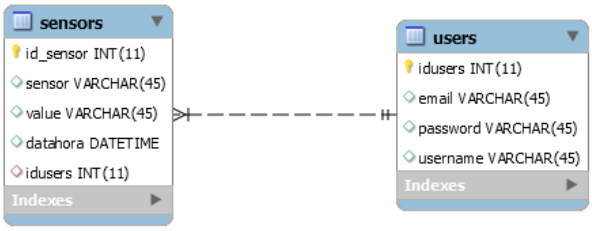


FIGURE 3.12. Users and Sensors database tables connection

The data stored in the *sensor* table is the most relevant for the user and the system to successfully monitor and control it through the mobile application. These are them:

- id - the sensor id to identify each entry in the database;
- sensor and value - the name and value of the sensor indicating the type and the data read by it at that moment;
- datahora - the date that indicates the instant when the reading was made;
- idusers - represents the foreign key that connects to the users table, giving access to the data to the users, in the mobile application.

As for the users, the *users* table was created on the server with only 4 entries:

- id - the user id to identify each entry in the database;
- username - the username that the person will choose, that will be used to login afterwards;
- email - user's email address, that will be referred to the user and may be used in the future to communicate a problem or so;

- password - this is the authentication password that the person will need to gain access to the rest of the application.

To connect the network to the database, a script named mqtt.py was developed capable of receiving the MQTT messages sent by the node containing sensor data, process those messages, to understand to which sensor it regards, and finally store the information in the database.

For that, the Python script created imports the Paho MQTT Python library, and subscribes to the corresponding topics for input messages, being always waiting for new messages. When one arrives, the script splits the message to get the sensor name and corresponding value, and stores the those data on the server. For this purpose, the script starts by creating a connection to the database created on the MySQL server using the MySQLdb library and performs a specific query for the intended goal.

3.2.3. Application Programming Interface

In order to make the server stored information available in the application, it was created another Python script named sensors_data.py that, using flask, was able to receive the specific information. Flask is a Python API that allows building web applications.

The mobile application needs to get specific data from the server and for that cause, it was implemented a HTTP connection in Java Android scripts to retrieve the information from the url that the flask script shows when running. This script imports various libraries and uses it in this order:

- Flask - generates an web application;
- MySQLdb - creates a connection with the remote database;
- request - routes a get request to the specific paths:
  - '/distinctsensors' - following the execution of a request made to this url, the content returned to the template will be the most recent data of each sensor that the database has stored. This is given thanks to the execution of a query that the mysql library provides;
  - '/allvalues' - when flask receives a request for this url, a query is executed in the database, in order to obtain and return the last 500 inputs of each sensor;
  - '/login' - the moment flask receives a request from this path, the method will be a post constituted by 2 parameters, the username and the password,

which will be used to execute a search in the users table. If the table contains an entry with these 2 fields, the entry result will be printed to the .html template, otherwise the value "None" is returned and the application will present a message according to the user who is trying to perform authentication;

  - '/signup' - flask will perform the insertion of the received data into the users database table. If this transaction is successful, the printout will be "true", otherwise it will result in another value printed on the page and sent to the application.

- json - this library is used to convert the format that the data retrieved from the server database table, to json gives, so it is easier to manipulate in the java application;
- render_template: is used to generate output from a template. Each request will have an .html template associated that this render will populate with the values returned from the server.

Once the data is received in the application, is manipulated and shown in the specific screens, as the Section 3.3 will demonstrates.

3.3. Data Visualization

In order for the user to be able to visualize the data obtained from the proposed water reuse system, it was necessary to develop an application. Among the mobile and web applications, the mobile application was chosen given some important advantages for this system, such as the ease of sending notifications and the possibility of being used offline.

The operating system chosen was Android, which was developed by Google, based on Linux, opensource, designed for mobile devices such as smartphones and tablets. Android is the most widely used operating system and it was the chosen one because Android Application Development is easy and quick to learn [**38**].

As such, an Android mobile application was developed from scratch, which logo is demonstrated at Figure 3.13, with both features and Android activities layouts and functionalities being presented in the following sections.

3.3.1. Application Features

The mobile application has functionalities that only the user is able to use, that are related with both sensors and actuators. This features need to allow the user to monitor

FIGURE 3.13. Application logo

and control, from the application, the controller node, and the following diagram translates the operation of the system into this actions.

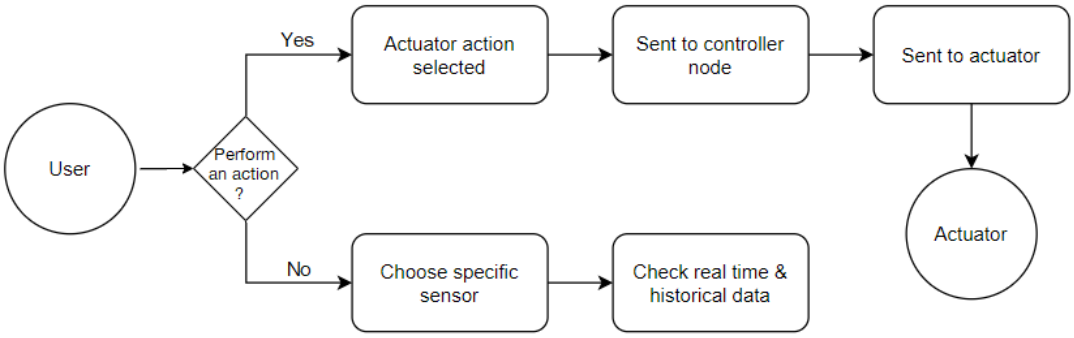


FIGURE 3.14. Data analysis towards controller node

Figure 3.14 functions performed by the system towards controller node, described more succinctly in the next paragraphs.

For the communication with and from the network, the Paho Java MQTT client library was used in the application's Java implementation, for the Android platform. With this library, it was able to create a connection to the client that MQTT broker has a link with and, just like it was made in the controller node, were was created specific messages and topics to publish and receive information.

In terms of sensors, the application allows the user to check all the collected and stores data by the network, with a dynamic dashboard showing the most recent values, or specific pages with historical data for each one. Besides that, the user can check values in real time, as they arrive from the network, as the application subscribes to the MQTT topic to where messages are published, as described in Section 3.1.4.

Regarding the actuator features, the mobile application allows for total control over the network. It starts from the user who, through the mobile application developed, selects the desired action. The action is sent to the node controller, which will perform the action on the actuator.

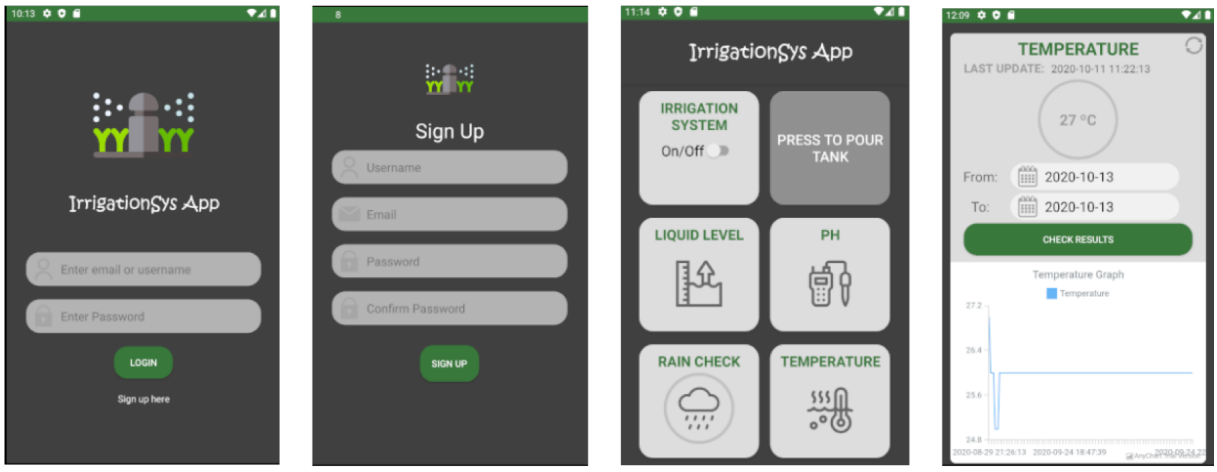
The application allow the user to control the actuators that were described in Section 3.1.3, and, when the user selects a specific actuator, it is created a connection with the broker client and, when successfully made, a message is published to that client, using the same method "client.publish(input_topic, message);". The message contains the value and the actuator that is being called, for example "1,pour_tank" and the topic differs since the flux is going in the opposite direction, and as already described, "teste/maria/in" topic is used.

With this the user can dump the tank, when he thinks is needed, sending the message "pour_tank", or irrigate the field with the tank water, sending the message "irrigation_system". The way these messages affect the corresponding actuators was already described in Section 3.1.4.

3.3.2. Android Activities

To allow the user to perform all the features previously described, the Android application is composed by a set of screens for each of the proposed activities. These screens provide a visual and interactive way to help the user to have a better experience.

Figure 3.15 shows all the screens developed, and the following sections will detail each of them.



a) b) c) d)

FIGURE 3.15. Application screen

3.3.2.1. *Login.* When opening the application, the Login screen, Figure 3.15 a), is displayed for the user to fill in the authentication data.

In order for the user to successfully authenticate he needs to fill in the username and the corresponding password, which were set during the registration process. When the

login task is completed successfully, the user is given access to the Dashboard screen and the user will be able to access sensor data and actuator actions. The user can decide to make a new registration on this screen by pressing the button accordingly.

3.3.2.2. ***Sign up.*** The sign up screen, Figure 3.15 b), is displayed to the user when, in the Login screen, the "Sign up here" button is pressed.

To successfully complete the sign up process, the user must fill in the username, email and password fields and then press the "sign up" button, redirecting the user to the Login homepage.

3.3.2.3. ***Dashboard.*** After the user successfully logs in, the main screen of the application is displayed, Figure 3.15 c). This shows all the features that the application provides to the user, in order to control and monitor their water reuse system.

The grid presents various models that differ as follows:

- Sensor model - shows the title at the top followed by the icon that represents its type. But there is a difference in this presentation that depends on the sensor:
  - rain sensor - to present the value that the rain sensor is collecting, the model shows a circle around the sensor icon. Depending on the value received for this sensor, the color of the circle changes, being colored green if it is raining and gray otherwise. If the user presses this model, a message will appear according to: "It's raining", Figure 3.17, or "It's not raining", Figure 3.16,;

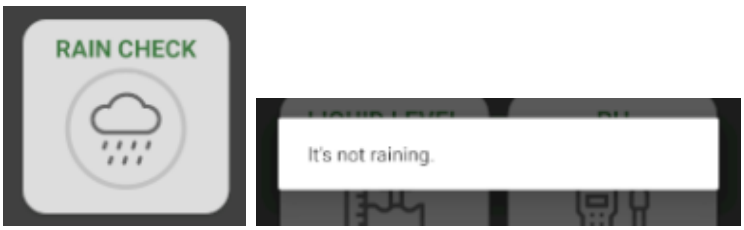


FIGURE 3.16. Not raining mode

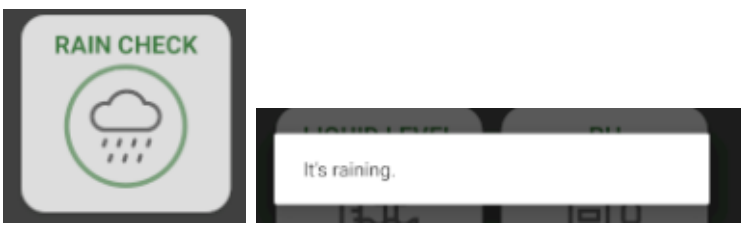


FIGURE 3.17. Raining mode

– other sensors - in order to display the rest of the sensors, the model always has the same format, Figure 3.18, and if the user selects a model of this type, is redirected to a screen that details the sensor, where he can see the value it measures, in real time;

FIGURE 3.18. Model for single pH sensor

- Actuators model - depends on its type:
  – water pump model - for the pump the title is displayed at the top, with the action it will perform when selecting this model, followed by a switch (On/Off) that represents the current state, Figure 3.19,. When selecting this model, the state is changed and this information is forwarded to the hardware.

FIGURE 3.19. Pump actuator model

  – servo motor model - for the servo motor it shows the action that will be performed if the user presses it, Figure 3.20,. This model works like a button and when pressed, this information is forwarded to the hardware.

FIGURE 3.20. Servo actuator model

3.3.2.4. *Sensor detail.* The last screen of the application, Figure 3.15 d), to be presented to the user is obtained when the user selects one of the sensors on the Dashboard screen.

The screen has, at the top, the name of the sensor, followed by the last data it was collected, and the date of collection. Below is a graph with all the most recent stored values, more specifically, the last 500 data collected. This period can be changed by changing the first and last day of data collected by selecting them in the "From:" and "To:" fields.

CHAPTER 4

System Implementation

The evolution of technology evidently leads to a greater search for solutions in this area that facilitate people's daily lives, especially through mobile devices, which are almost indispensable to anyone.

The system developed allows this with a network of sensors and actuators and a mobile application that translates to the user the status of water reuse system, presenting information about the components installed in the system, all within reach of his mobile phone.

In the second chapter the best solution for the system was studied and in the third chapter its architecture was described. Taking these two chapters into consideration, this fourth chapter is divided in two parts that portrays how the one previously described was implemented and tested in an experimental environment.

4.1. Assembly

This chapter aims to describe the implementation of the system, which represents a prototype model of the actual system.

4.1.1. Controller Node

The system consists of only one node, the controller. This has all the functions of the various nodes that are usually common in WSN, as described in Section 3.1.1.

4.1.2. Communication

One of controller's node functions is the connection between the network, the mobile application and the server. It routes messages through the MQTT to their destination, based on the topics, and receives messages destined to the actuators, with specified actions, sent from the mobile application, selected by the user.

To simulate the sending of these messages and test their success in sending and receiving, the HiveMQ platform was used. We started by entering the necessary information to establish the connection with the server and indicated the topic "test/maria/out" subscribed, Figure 4.1, which will receive messages from the server.

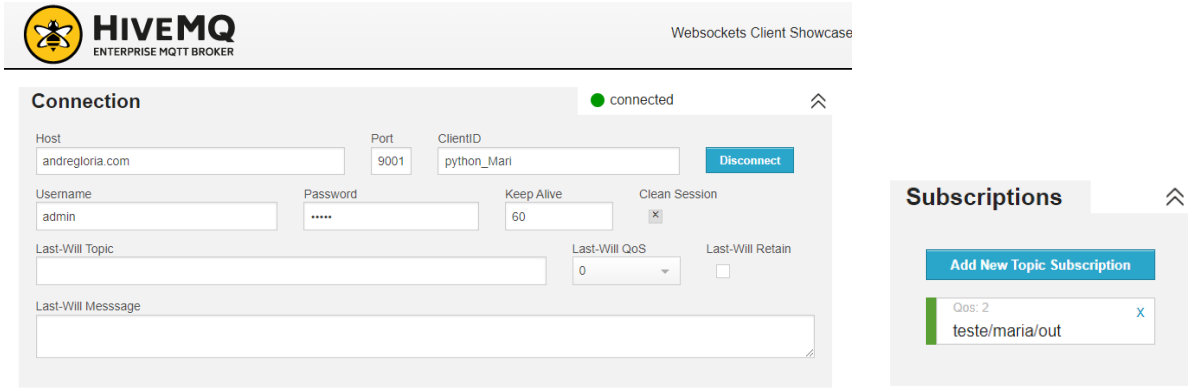


FIGURE 4.1. HiveMQTT connection creation and subscription of topic

The controller node received the reading made by the sensors and actuators and then sent this information to the server, through messages via MQTT. As explained in Section 3.2.2, the Python script is running and has a subscribe method that is always waiting for messages from the topic "teste/maria/out" and, when receiving a message from it, stores the message content in the database. With the connection established at the HiveMQTT, and the topic subscribed, the messages that go to the server were catch in the platform, Figure 4.2.

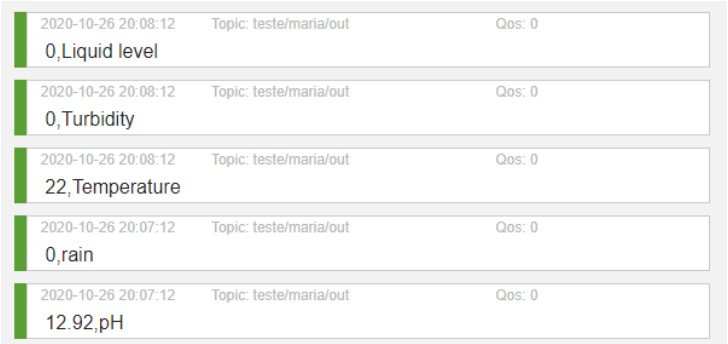


FIGURE 4.2. HiveMQTT messages from the subscribed topic

The topic that makes the communication in the opposite direction is "teste/maria/in", that is set in the arduino sketch to always be awaiting for a message from the topic. This messages in the system are sent from the mobile application, where the user chooses an actuators actions. To test this feature, the three kinds of messages that the mobile application is able to send were simulated and were successfully received in the controller node, Figure 4.3 and 4.4.

4.1.3. Sensors

The set of sensors that constitute the controller node and the I/O pins used in the ESP32 are represented in Table 4.1.

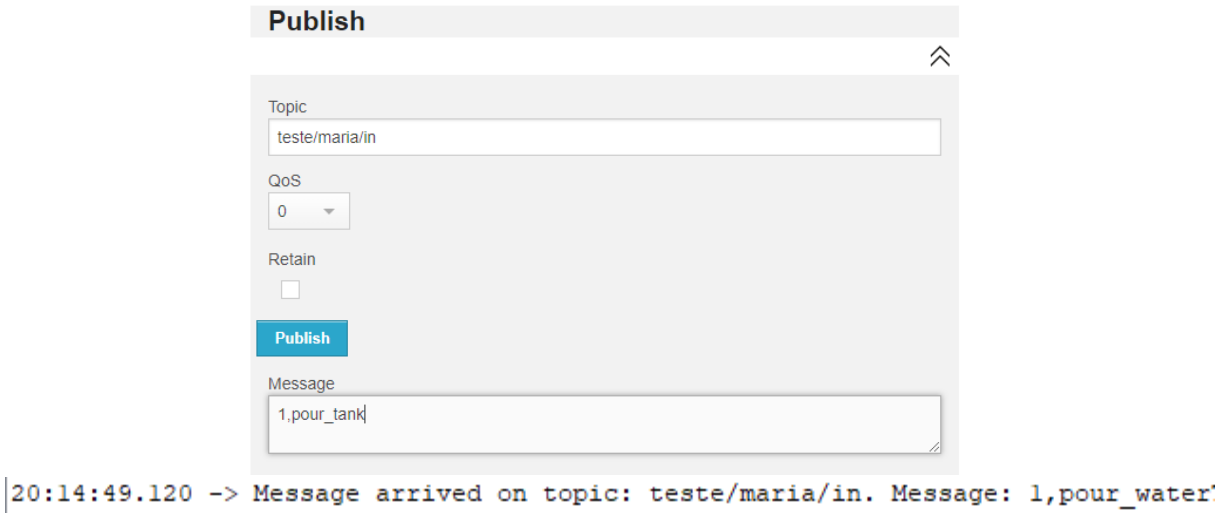


FIGURE 4.3. HiveMQTT test input topic for pouring the tank

FIGURE 4.4. HiveMQTT test input topic for start/stop irrigation system

The sensors are read in the script created for the arduino platform that is implemented on ESP32, as described in Section 3.2.1. To successfully read the temperature sensor, it was necessary to import two libraries:

TABLE 4.1. Sensor's details

Sensor	Pin	Range values
pH	33	[0, 14]
Temperature	4	[-55, 125^{o}C]
Rain	34	[0, 1]
Turbidity	2	[0, 1]
Liquid level	5	[0, 1]

- OneWire - gives access to one wire temperature sensors, memory and other components;
- DallasTemperature - This library allows you to use the sensor and other similar 1-wire temperature sensors. For this, it is necessary to first define a 1-wire bus to which dallas sensors are connected, in this case DallasTemperature sensors(&oneWire).

For the water pH value, the sensor translates the amount of H+ protons that are integrated, which indicates whether the solution is acid, neutral or alkaline. Thus, the pH meter of the water must be calibrated, since the H+ varies over time, making the readings more accurate. For this, the calibration was done with three substances, being the experimental test illustrated in Figure 4.5. The substances were vinegar, leach and water, and the results of these calibration tests are represented in Figure 4.6.

FIGURE 4.5. pH sensor tests experiment

The three figures translates the reading made from the pH sensor to the three substances and its variation, in the period of ten minutes.

The first one represents the solution with water, to analyse the difference in ten minutes of the good quality water pH.

```
11:34:50.081 -> Value read: 7.12 | 10:13:32.146 -> Value read: 2.19
11:35:50.103 -> Value read: 7.24 | 10:14:32.464 -> Value read: 2.07
11:36:50.099 -> Value read: 7.19 | 10:15:32.738 -> Value read: 2.11
11:37:50.090 -> Value read: 7.19 | 10:16:33.061 -> Value read: 2.11
11:38:50.083 -> Value read: 7.26 | 10:17:33.372 -> Value read: 2.19
11:39:50.075 -> Value read: 7.27 | 10:18:33.651 -> Value read: 2.21
11:40:50.111 -> Value read: 7.15 | 10:19:33.937 -> Value read: 2.19
11:41:50.076 -> Value read: 7.13 | 10:20:34.233 -> Value read: 2.08
11:42:50.080 -> Value read: 7.21 | 10:21:29.723 -> Value read: 2.19
11:43:50.072 -> Value read: 7.27 | 10:22:30.050 -> Value read: 2.17
11:44:50.094 -> Value read: 7.27 | 10:23:30.347 -> Value read: 2.19
                10:26:31.238 -> Value read: 8.39
                10:27:31.556 -> Value read: 10.43
                10:28:31.851 -> Value read: 10.36
                10:29:32.141 -> Value read: 10.50
                10:30:32.427 -> Value read: 10.51
                10:31:32.742 -> Value read: 10.40
                10:32:33.029 -> Value read: 10.51
                10:33:33.318 -> Value read: 10.40
                10:34:33.653 -> Value read: 10.50
                10:35:33.942 -> Value read: 10.44
                10:36:34.253 -> Value read: 10.50
```

FIGURE 4.6. pH sensor tests results of water, vinegar and leachate

Second figure, represents vinegar added to 7 pH water, so that the pH value would decrease to an acid solution pH value and it happened as expected, making the test successful.

For the third one, where leached was added to 7 pH water, also happened as predicted, in which the solution pH increased to a basic pH values solution.

The results presented reflect a successful calibration, since the pH value increased and decreased as expected.

4.1.4. Actuators

The set of actuators that constitute the controller node and the I/O pins used in the ESP32 are represented in Table 4.2.

TABLE 4.2. Actuator's details

Actuator	Pin	Range values
Servo motor	23	[0^{o}, 180^{o}]
Water bomb	19	[HIGH, LOW]

To read and write the values of the servo motor actuator, it was necessary to import, in the arduino platform script, a Servo library, to obtain its successful operation.

4.2. Laboratory Implementation

To verify that all components work correctly together, an experimental test was performed, using an equivalent but proportionally smaller scheme. This way, it was possible

to confirm that the system and all the surrounding components communicated successfully in the intended way.

Specifically the sensors, that might need some kind of calibration or conversion of the information that is collected from them, were all tested, since they are crucial for the system and it is necessary to guarantee that the collection of correct data id done, in order to further store and analyze them in the final system.

The actuators, on the other hand, need another type of evaluation because they are used to perform actions that the user wants and that are available in the mobile application developed.

4.2.1. Sensor Calibration and Tests

All of the system's sensors were tested. Since the sensors that measure water temperature and pH have a high variation, these were tested for a period of 7 days in a experimental period, every 5 minutes, resulting in a total of 2016 samples per sensor that are represented in Figure 4.7 and 4.8.

The graphics presented were generated in Excel, taking information from the database, according to the sensor and the specific period.

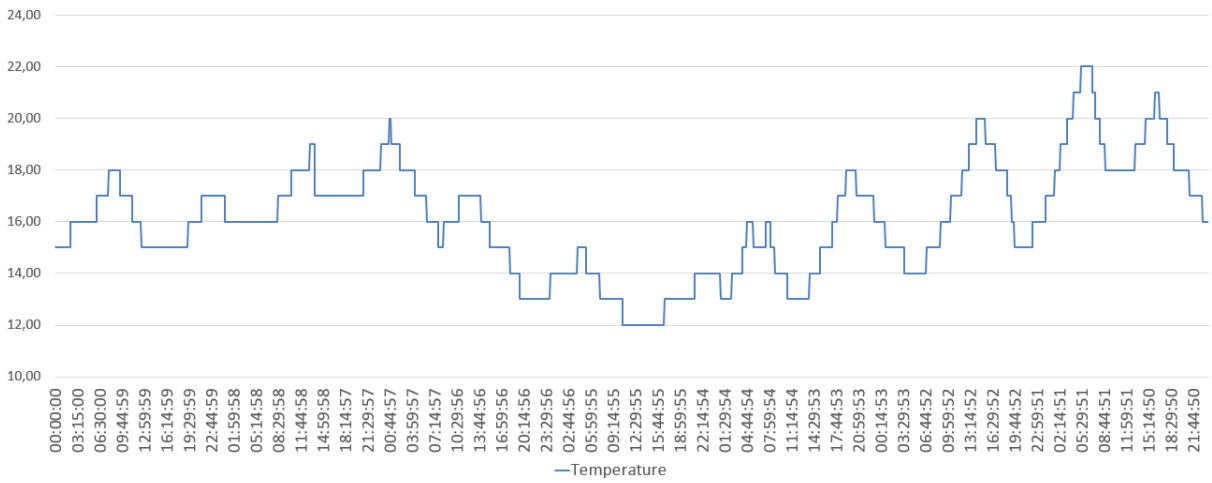


FIGURE 4.7. Temperature sensor measurements in 7 days period

Analyzing the results presented in the pH chart, it is possible to verify the oscillation of its value between 6 and 7.5/8. This variation was predictable, given that the stipulated pH value for water is, on average, 7, hence the fluctuation between more or less this value. For the temperature, it also presents an expected value, since the water is in a closed tank and, in this experimental test, it was in a closed environment, not having the heating that could exist in a place with light and solar exposure, for example.

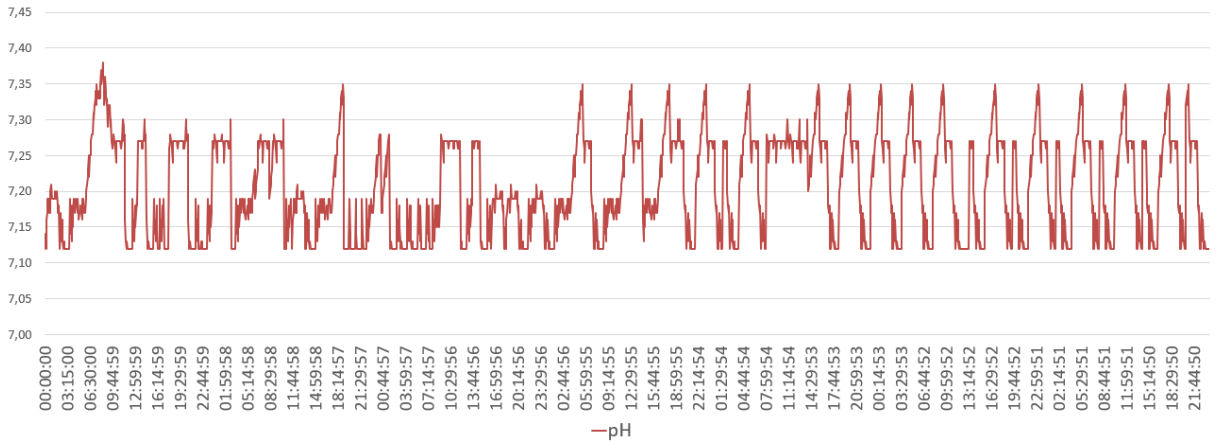


FIGURE 4.8. pH sensor measurements

As for the sensors that measure the turbidity and water level and the capacitive sensor that checks if there is rain, return values between 0 and 1, as stated by Table 4.1 and, therefore, it does not make sense to perform experimental analysis over a period of days, as was done for the sensor that measures the temperature and pH of the water. Only one test was then done, to check if the sensor detects the variation of values, and the success of the reading of these three was verified.

4.2.2. Actuators Tests

To test the actuators acting in the system, individual tests were made to verify their successful operation.

The servo motor was tested performing the complete rotation of the tank and its closing, going from 0^{o} to 90^{o} and the opposite direction. While this movement was being performed, the values of the actuator were read, Figure 4.9, which are presented in the following figure and which translate the predictable values of the opening and closing angles, Figure 4.10.

```
14:11:27.968 -> Opening servo
14:11:27.968 -> Value read before write: 0
14:11:28.782 -> Value read after write : 90
14:11:31.767 -> Closing servo
14:11:31.800 -> Value read before write: 90
14:11:32.309 -> Value read after write : 0
```

FIGURE 4.9. Servo test

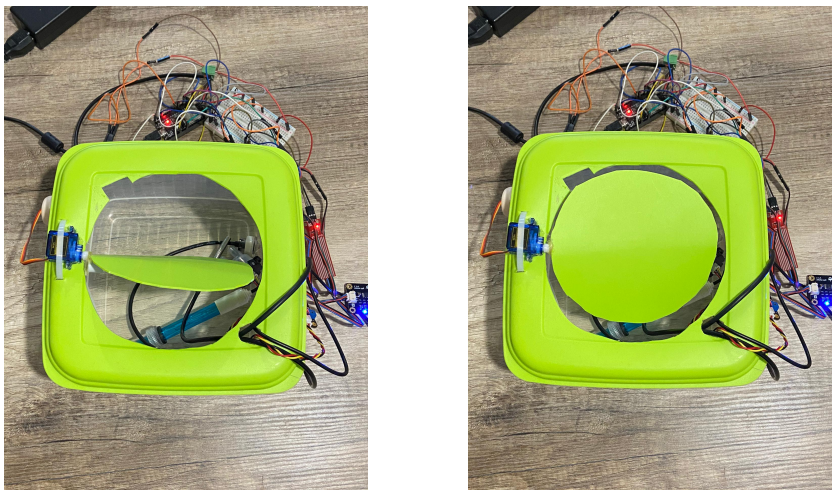


FIGURE 4.10. Servo with 90^{o} and 0^{o}

4.3. Experimental Implementation

The system's components were tested for the different possible scenarios in order to ensure the correct reading and processing of the data that is presented to the user, and to verify that the network works correctly when it comes to exchanging messages.

This step is very important to verify some anomaly of the system, before using the prototype in a real environment, and to assess some future improvement that can be performed.

Due to current conditions, it was not possible to build the tank and apply it in a real environment, so all functionalities were tested in laboratory environment.

The tank and the sensors were implemented at home, with normal conditions and environmental temperature, simulating an underground tank, that in a real environmental would be distant from the irrigation terrain.

The only significant difference is not having the component of rain that will exist in the real environment. To simulate the existence of precipitation, water was splashed on the sensor, from time to time. This action is quite important, since it is what makes the opening and closing of the lid of the tank in order to retrieve the rain water that is going to be analyzed and reused.

The scenarios for the test were then performed, starting with the implementation of the controller node, where all the sensors and actuators components are connected to the ESP32. To simulate the tank, a container with a lid was used, as Figure 4.11 shows.

The prototype, as shown, is composed by:

- 1 - The tank where are contained the components that evaluate the water quality;

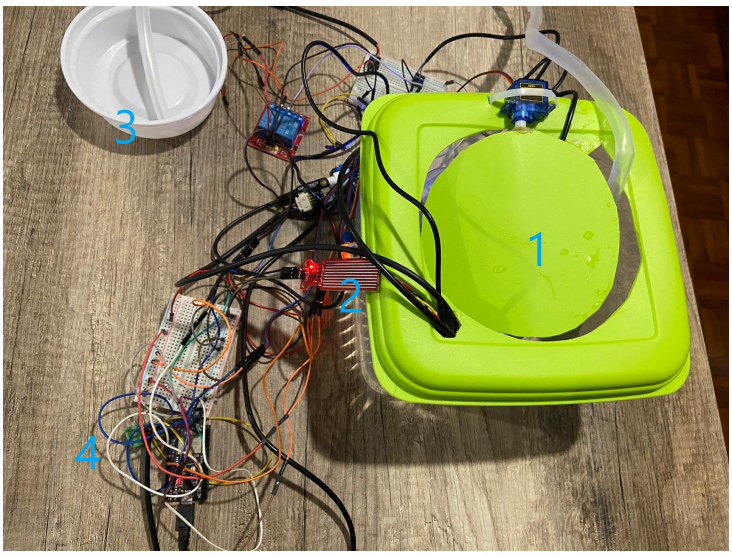


FIGURE 4.11. System prototype outside of "tank"

- 2 - Capacitive component that checks if it is raining or not, so the servo opens or closes;
- 3 - Represents the field, where the pump will push the water to, when irrigation starts;
- 4 - Where all the components of the system are connected, to the esp32 that communicates to the network the component's values.

Inside the tank, where the water is stored, the necessary sensors and actuators, temperature sensor, pH, liquid level, turbidity and the water pump actuator, were placed. To start the the test scenarios, some water was poured inside the tank, Figure 4.12,to allow the sensors to measure data.

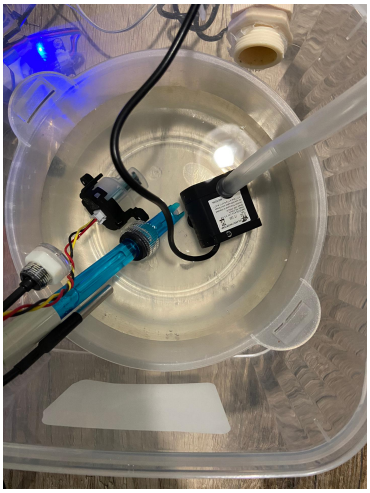


FIGURE 4.12. Real system prototype inside the fake tank

The various possible scenarios to be tested in this environment will be described below, following by the results and discussion.

4.3.1. Scenarios

The tests started with the scenario where the tank is empty, which means that it does not contain enough water inside and, if when selecting the option to start watering, in the mobile application, it is done with the water contained in the reserve tank.

In order to fill the empty tank, it is necessary to rain. For this to happen, the rain was simulated, splashing the capacitive sensor that measures it. This way, the controller checks if there is enough rain to justify the opening of the lid, to allow water to fill the tank.

The rain water is stored until it reaches the maximum capacity level of the tank, indicated by the liquid level sensor, or when it stops raining, ending this cycle with the closing of the tank lid.

This water is kept in the tank for the necessary period, until the user decides to start watering again or to dump the water from the tank, through the mobile application. The dump can also be done without the user intervention, since the water quality is under constant analysis and one of the tests is to check the dump of the tank, when this quality does not exist.

4.3.2. Results & Discussion

After implementation, the previously described scenarios were performed at several times per day, during the 24-hour period.

The actual test would imply a longer period, since irrigation is done at specific times of the day, depending on the clear terrain, but given the circumstances that led to the test being done in the experimental environment, the decision was to test the various scenarios in this period.

The results obtained from this experimental test were put into graphs. Figure 4.13 represents the variation of data over time for: pH, temperature, turbidity, liquid level max and min and rain, resulting from the tests.

Starting with the evaluation of the temperature variation, it is almost always constant with the ambient temperature, since the environment was the interior of a house, so it did not catch extreme temperatures that can happen when implementation in a real environment. Hot water was put in to test a water that was not good for irrigation,

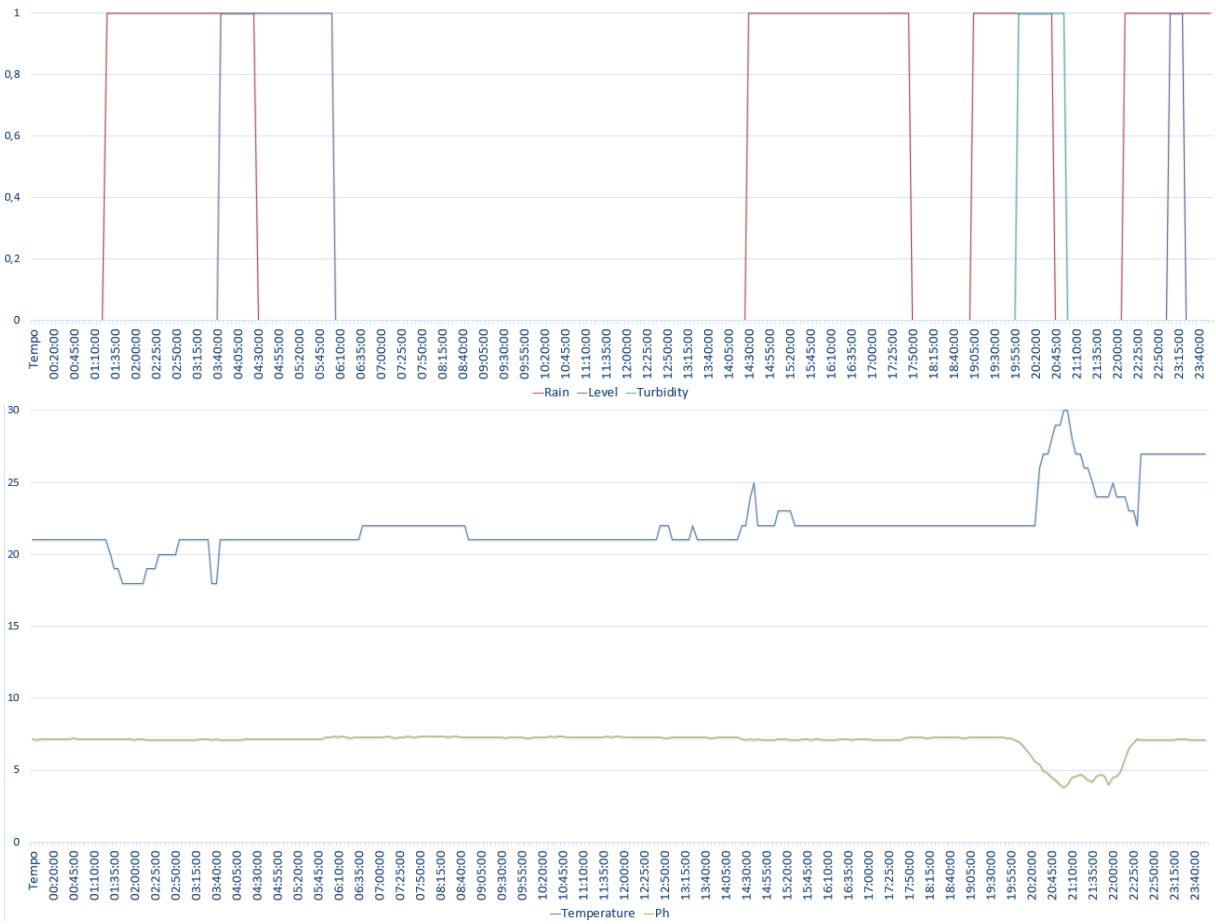


FIGURE 4.13. Results of prototype implementation tests

together with other relevant values, and it is possible to check when this happened, where the graph started increasing at 20:25, reaching 30ºC maximum.

The pH of the water is always in values that are close to what translates into a neutral pH. To test the water with poor quality, this pH was manipulated, so that the test of dumping of the tank was carried out, Figure 4.14, and it is possible to verify that it was done at 20:25. The water level remained at 0 indicating that there was no water in the

FIGURE 4.14. Scenario where the water pump starts pumping to another recipient, simulating the pouring of the water or the start of irrigation

tank, or that the tank still had room for more water. When it became full, figure 4.15,

its value changed to 1 at 03:40, 16:40 and at 22:55. This forces the servo to close, which he successfully accomplished. The rain simulation, Figure 4.16, was carried out at 01:10,

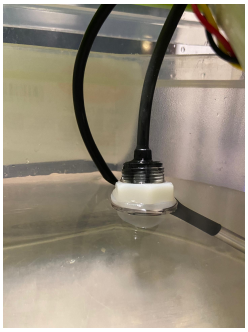


FIGURE 4.15. Scenario where the liquid level sensor value is 1, as the water reaches the limit of the tank

14:25, 20:25 and 22:10, and it can be confirmed given the value of this sensor translate 1 at this time. This caused the tank to open, by the servo, Figure 4.17, and the water

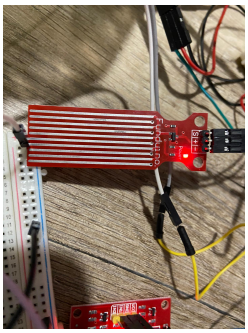


FIGURE 4.16. Scenario where the capacitive sensor read 1, since it had water on the top of it, simulating rain

enters the tank and when it stops raining, its value changes to 0 and the tank is closed by the servo too, Figure 4.18. If the tank is full, the the servo will keep the tank closed. The value of the turbidity also changed in the period of 19:55 to 21:10, reflecting a poor water quality.

The joint analysis concludes that at 01:10 the tank for the entry of the water was opened, as it started to rain.

The water was kept until it reached the limit of the tank and it closed, at 03:40, even if the rain was still going on.

The water was then poured through the existing option in the mobile application, translated into the liquid level value which changes to 0 at 06:10.

Then, at 14:25, it is possible to verify that it rains again and the process is done again, the servo is opened to enter the rain and, this time, when the rain stops, the servo is closed, at 17:40. This water was poured out by an action sent by mobile application.

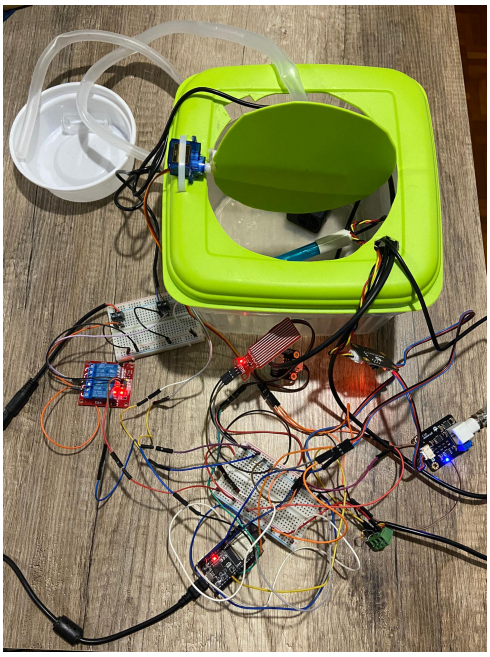


FIGURE 4.17. Scenario where the tank is opened because the simulation of rain started and the tank was not full yet

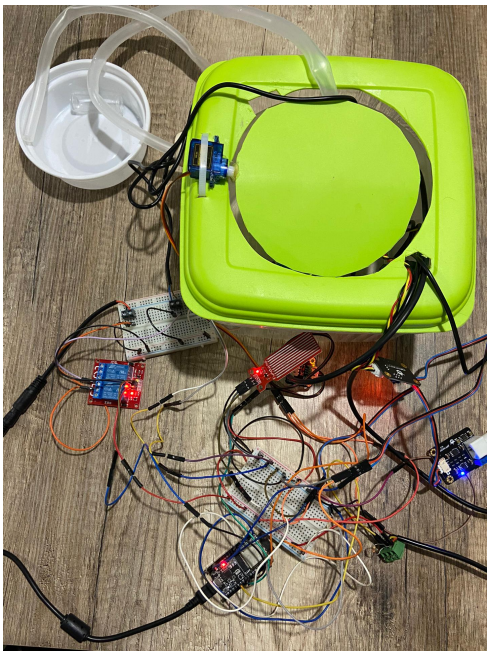


FIGURE 4.18. Scenario where the tank is closed because the simulation of rain stopped or because the tank has fully filled

At 18:55 started to rain again and stopped at 20:40. It can be verified in the graph that this new water is composed of different values from the previous ones, reflecting a poor quality. The water quality is calculated by the values of temperature, pH and turbidity, which leads to the emptying of the tank, at 21:10, where temperature, pH and turbidity stopped measuring not recommended values.

There was rain again at 22:10 and this water, since it was good, was used for irrigation, a request made by the mobile application at 23:25 and carried out. As this action was made, the water level reached it maximum but, with the irrigation starting, it was again decreased.

Some actions are not possible to check on the graph, but an attempt to start irrigation by the application when the tank was empty was also tested. This test was successful since, as the tank was empty, irrigation was impossible.

CHAPTER 5

Conclusions

5.1. Conclusions

The work presented describes a solution based on IoT technology for the creation of a monitor and controlling system for intelligent irrigation systems, with the support of a mobile application. This solution has as main objective the saving of the water resource and is based on a hardware system and protocols chosen specifically for the intended environment, after a study in which the various options were compared.

The system consists of only one node, the controller node, which communicates over Wi-Fi, since it is based on an ESP32 controller, which incorporates this technology. The system is composed of sensor and actuator components creating a WSAN.

The communication between the controller node and the server was implemented using the MQTT protocol, given its simplicity, and for this purpose the topics in and out were specified, which translate the two communication directions. Besides the node-server communication, this protocol is used to enter the values read from the sensors in the database server.

In the experimental testing phase, the communication was first tested, in which the messages exchanged on the network were tested in both directions, with the two topics, through the HiveMQTT platform. Then, individual tests were also performed on the hardware components of the system, both sensors and actuators. Before these individual tests, which were more detailed for the pH and temperature sensor, given its range of values and for the servo actuator, for the same reasons, the calibration of the components was performed, which was described only the most relevant, which was the pH, given the importance of this value for the proposed system. These experimental tests were all successful, translating into a confidence in the components that were to be used for the implementation.

Once the laboratory tests phase was finished, the hardware and software were implemented in an experimental environment, since it was not possible to implement them in a real one, as a consequence of the current conditions. The period in which the test was

performed was less than what would have been expected in a real environment, since the same environmental conditions do not exist.

Despite the different characteristics of the environment, compared with what one would expect, the data reading translated in a good performance of the system in the experimental environment, since it was possible to test all possible scenarios, which would occur in a real environment, extracting the data from the sensors, sending them to the network, where they are stored, and presenting them, through the mobile application, to the end user.

In short, the system presented was able to achieve the main objectives proposed for the dissertation, which reflected the contribution to water saving and, consequently, to sustainability, through the control and monitoring of water in the irrigation system. In this way, water that comes from rain, which would be wasted, is saved for future irrigation, saving water that is used for other actions, such as irrigation of other crops or even water for housing.

5.2. Future work

Despite the benefits that the implemented system presents, there are improvements that can be made:

- testing in a real environment - given the circumstances, it was not possible to test in a real environment, and these tests may indicate other possible improvements that have not been verified here;
- iOS application - the replication of the application to a new operating system, will increase the number of users of the application;
- increase the number of sensors - at this point there is only one sensor that measures the water level when it reaches the top, a sensor could be added at the bottom, checking that the tank is empty, or another sensor that would increase the accuracy of the water quality value.

References

[1] E. Vivas and R. Maia, "AVALIAÇÃO DE SITUAÇÕES DE SECA E ESCASSEZ DER ÁGUA EM PORTUGAL CONTINENTAL. Utilização de indicadores," *9º Congresso da Água*, p. 15, 2008.

[2] N. Dlodlo and J. Kalezhi, "The internet of things in agriculture for sustainable rural development," *Proceedings of 2015 International Conference on Emerging Trends in Networks and Computer Communications, ETNCC 2015*, pp. 13–18, 2015.

[3] B. M. Nonnecke, M. Bruch, and C. Crittenden, "IoT and Sustainability: Practice, Policy and Promise Public Symposium," 2016. [Online]. Available: https://escholarship.org/uc/item/7dp1t4p8

[4] F. Nicolalde, F. Silva, B. Herrera, and A. Pereira, "Big Data analysis tools in IoT and their challenges in open researches — Herramientas de Análisis de Big Data en IoT y sus Desafíos en Investigaciones Abiertas," *Iberian Conference on Information Systems and Technologies, CISTI*, vol. 2018-June, pp. 1–6, 2018.

[5] P. Matta and B. Pant, "Internet-of-things: Genesis, challenges and applications," *Journal of Engineering Science and Technology*, vol. 14, no. 3, pp. 1717–1750, 2019.

[6] "IoT application areas," (visited 15/09/2020). [Online]. Available: https://www.fracttal.com/en/blog/the-9-most-important-applications-of-the-internet-of-things

[7] A. Glória, P. Sebastião, C. Dionísio, G. Simões, and J. Cardoso, "Water management for sustainable irrigation systems using internet-of-things," *Sensors (Switzerland)*, vol. 20, no. 5, pp. 1–14, 2020.

[8] S. Pooja, D. V. Uday, U. B. Nagesh, and S. G. Talekar, "Application of MQTT protocol for real time weather monitoring and precision farming," *International Conference on Electrical, Electronics, Communication Computer Technologies and Optimization Techniques, ICEECCOT 2017*, vol. 2018-Janua, pp. 814–819, 2018.

[9] C. Kamienski, J. P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. S. Cinotti, R. F. Maia, and A. T. Neto, "Smart water management platform: IoT-based precision irrigation for agriculture," *Sensors (Switzerland)*, vol. 19, no. 2, 2019.

[10] "FIWARE," (visited 14/09/2020). [Online]. Available: https://www.fiware.org/

[11] R. Nageswara Rao and B. Sridhar, "IoT based smart crop-field monitoring and automation irrigation system," *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, no. Icisc, pp. 478–483, 2018.

[12] M. D. Dukes, "Water conservation potential of landscape irrigation smart controllers," *Transactions of the ASABE*, 2012.

[13] "GreenIQ application," (visited 10/09/2020). [Online]. Available: https://greeniq.com/

[14] "Racho application," (visited 13/09/2020). [Online]. Available: https://rachio.com/

[15] F. Shoushtarian and M. Negahban-Azar, "World wide regulations and guidelines for agriculturalwater reuse: A critical review," *Water (Switzerland)*, vol. 12, no. 4, 2020.

[16] A. Rebelo, M. Quadrado, A. Franco, N. Lacasta, and P. Machado, "Water reuse in Portugal: New legislation trends to support the definition of water quality standards based on risk characterization," *Water Cycle*, vol. 1, no. May, pp. 41–53, 2020. [Online]. Available: https://doi.org/10.1016/j.watcyc.2020.05.006

[17] A. K. Charan, "IoT based Domestic Water Recharge System," no. Icssit, pp. 901–906, 2020.

[18] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, "Communication protocols for wireless sensor networks: A survey and comparison," 2019.

[19] "Android," (visited 13/09/2020). [Online]. Available: https://www.arduino.cc/en/guide/introduction

[20] E. Systems, "Datasheet," vol. 32.

[21] "Comparison between Android and ESP32," (visited 10/10/2020). [Online]. Available: https://xprojetos.net/arduino-esp32-e-esp8266-comparacao/

[22] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, 2017.

[23] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of Lora: Long range & low power networks for the internet of things," *Sensors (Switzerland)*, 2016.

[24] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Jihoon, and K. Han, "Analysis of latency performance of bluetooth low energy (BLE) networks," *Sensors (Switzerland)*, 2015.

[25] "ESP-NOW," (visited 13/09/2020). [Online]. Available: https://www.espressif.com/en/products/software/esp-now/overview

[26] "ESP NOW Wi-Fi," (visited 13/09/2020). [Online]. Available: https://learn.circuit.rocks/esp-now-the-fastest-esp8266-protocoladvantages

[27] J. Cai and O. C. Ugweje, "A comparative study of wireless ATM MAC protocols," *2002 International Conference on Communications, Circuits and Systems and West Sino Exposition, ICCCAS 2002 - Proceedings*, pp. 516–520, 2002.

[28] IoT, "Nb-IoT." [Online]. Available: https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/

[29] "Sig Fox," (visited 13/09/2020). [Online]. Available: https://www.sigfox.com/en/what-sigfox/technology

[30] "Explaning Sig Fox," (visited 23/09/2020). [Online]. Available: https://dzone.com/articles/explaining-sigfox

[31] T. M. Workgroup, "A technical overview of LoRa ® and LoRaWAN ™ What is it?" no. November, 2015. [Online]. Available: https://lora-alliance.org/resource-hub/what-lorawantm

[32] "Cloud technologies," (visited 09/10/2020). [Online]. Available: https://www.investopedia.com/terms/c/cloud-computing.asp

[33] "IoT Dashboard," (visited 15/09/2020). [Online]. Available: https://www.mobindustry.net/how-to-create-web-dashboards-for-iot-devices/

[34] "Statistics Android vs iOS," (visited 03/10/2020). [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[35] V. Brunton, "Irrigation water quality," no. November, p. 2, 2011.

[36] R.J. Worrall, "THE EFFECT OF IRRIGATION WATER TEMPERATURE ON THE GERMINATION AND GROWTH OF PLANTS," (visited 13/09/2020). [Online]. Available: https://www.actahort.org/books/79/79$_1$6.*htm*

[37] "MQTT Protocol," (visited 20/09/2020). [Online]. Available: https://www.hivemq.com/blog/mqtt5-essentials-part9-request-response-pattern/

[38] "Android and iOS applications," (visited 13/09/2020). [Online]. Available: https://theappsolutions.com/blog/development/ios-vs-android/

Appendices

APPENDIX A

Sensor's and Actuator's Specification

University Institute of Lisbon

Department of Information Science and Technology

Sensor's and Actuator's Specifications

Maria Inês dos Santos Pires

November, 2020

Contents

Chapter 1

Sensor's and Actuator's Specifications

1.1 Sensors

1.1.1 Analog Ph sensor

- Module Power - 5.00V
- Module Size - 43mm×32mm
- Measuring Range-0-14PH
- Measuring Temperature -0-60 ℃
- Accuracy - ± 0.1pH (25 ℃)
- Response Time - <= 1min
- pH Sensor with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED
- Cable Length from sensor to BNC connector-660mm

1

1.1.2 Capacitive sensor

- Operating voltage- DC 3-5V
- Operating current- less than 20mA
- Sensor Type- Analog
- Detection Area- 40mm x 16mm
- Production process- FR4 double-sided HASL
- Operating temperature- 10^{o}C-30^{o}C
- Humidity- 10%-90% non-condensing
- Dimension- 60 x 19.5 x 2mm
- Weight- 3.5 g

1.1.3 Analog turbidity sensor

- Operating Current- 40mA (MAX)
- Response Time - <500ms
- Insulation Resistance- 100M (Min)
- Output Method:
  - Analog output- 0-4.5V
  - Digital Output- High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature- 5℃ 90℃
- Storage Temperature- -10℃ 90℃
- Weight- 30g
- Adapter Dimensions- 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

1.1.4 Waterproof DS18B20 digital temperature sensor

- Usable with 3.0V to 5.5V power/data ±0.5℃
- Accuracy from -10℃ to +85℃
- Usable temperature range - -55 to 125^{o}C (-67^{o}F to +257^{o}F)
- 9 to 12 bit selectable resolution
- Uses 1-Wire interface- requires only one digital pin for communication
- Unique 64 bit ID burned into chip
- Multiple sensors can share one pin
- Temperature-limit alarm system
- Query time is less than 750ms
- 3 wires interface:
  - Red wire - VCC
  - Black wire – GND
  - Yellow wire - DATA
- Stainless steel tube 6mm diameter by 35mm long
- Cable diameter- 4mm
- Length- 90cm

1.1.5 Photoelectric liquid level sensor

- Model- FS-IR02
- Type- Photoelectric Liquid Level Sensor
- Operating Voltage- 5V DC Output
- Current- 12mA
- Operating Temperature- - 25 105 ℃

- Low Level Output- < 0.1 V
- High Level Output- > 4.6 V
- Liquid Level Detection Accuracy- ± 0.5 mm
- Material- Polycarbonate
- Measuring Range- No limit
- Life- 50,000 hours

1.2 Actuators

1.2.1 Servo motor

These machines have three components:

- a system which is composed of a DC or AC electric motor;
- the sensor that indicates the angular position of the axis through its electrical resistance;
- the control circuit that is composed of an oscillator and a PID controller (integrative and derivative proportional control) that receives signals from the sensor (axis position) and the control and moves the motor in the desired direction and angle.

1.2.2 Water tube

- Power supply - 3.5 12V DC,65mA-500mA
- Pumping head - 40-220cm
- Capacity - 100-350L/H
- Power range - 0.5W-5W
- Dimensions - 38x38x29mm

- Weight - 125g
- Cable length - 1m

5

APPENDIX B

User & Technical Manual

University Institute of Lisbon

Department of Information Science and Technology

User & Technical Manual

Maria Inês dos Santos Pires

November, 2020

Contents

List of Figures

Chapter 1

User Technical Manual

For the user, owner of the irrigation system, to be able to control and monitor that system, in a practical way, a mobile application was developed, in the Android operating system. Through the application, the user can view the data that the implemented sensors read, in real time.

This user manual describes how to use the system and, step by step, how to assemble and install it.

1.1 Mobile Application

To describe the use of the system, the various screens of the application, its meaning, the way to reach it and the actions that the user can do in it, if applicable, will be presented.

1.1.1 Login

Login screen is the first screen displayed, when opening the application, as figure 1.1 illustrates.

1

User Technical Manual*User Technical Manual*

Here, the user must fill in the authentication data and press the "Login" button to proceed to the next screen of the application.

In order for the user to successfully authenticate, he needs to fill in the username and the corresponding password, which were set during the registration process, 1.1.2. Once this task is completed successfully, the user is given access to the Dashboard page, figure 1.8, and will be able to access sensors' information and actuators' actions.

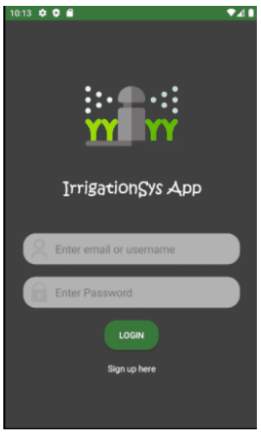


FIGURE 1.1: Login screen

The user can decide to make a new registration on this screen by pressing the button accordingly.

1.1.2 Sign up

In case the user hasn't ever sign up in the application, he has to select, in the Login screen, 1.1, the "Sign up here" button, and the sign up screen, 1.2, is displayed.

To successfully complete the sign up process, the user must fill in the username, email and password fields and then press the "sign up" button, redirecting to the

User Technical Manual*User Technical Manual*

Login homepage again, where he must fill the correct data to access the mobile application content.

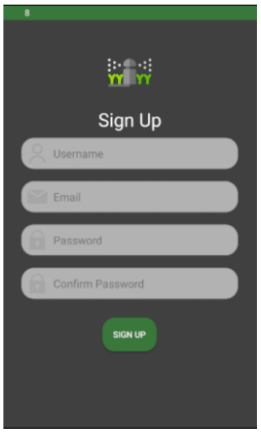


FIGURE 1.2: Sign up screen

1.1.3 Dashboard

After the user successfully finishes the log in task, the main screen of the application is displayed, 1.8. Here are presented all the features that the application provides to the user, in order to control and monitor his irrigation system.

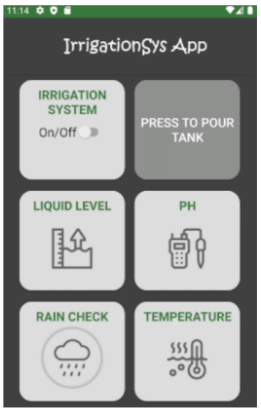


FIGURE 1.3: Dashboard screen

User Technical Manual*User Technical Manual*

As the figure 1.8 presents, the screen will contain different topics on the grid, and each has a different propose that is described next.

1.1.3.1 Pour tank icon

This icon represents a button that, once the user presses, the system's tank will be poured. This action can be made, for example, if the user sees the values of the water in the tank are not granting the quality the system needs.

FIGURE 1.4: Pour tank icon

1.1.3.2 Start/Stop irrigation system icon

When the user pretends to start or stop the irrigation system, he can do that action from the application simply by pressing the switch, as the figure 1.5 illustrates.

FIGURE 1.5: Pour tank icon

1.1.3.3 Rain check icon

This icon traduces if it is raining or not. The circle around the icon will turn green if it is raining and gray if it is not. Also, the user can press the icon and a message will describe it.

User Technical Manual*User Technical Manual*

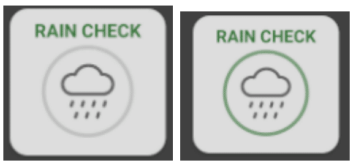


FIGURE 1.6: Rain check icon

FIGURE 1.7: Rain check messages

1.1.3.4 Basic sensor icon

This type of icon represents a sensor. It will have on the top of the icon the type of the sensor and the image centered, describes that type.

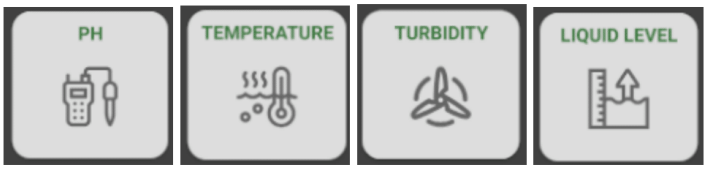


FIGURE 1.8: Different sensors' screens

Once this type of icon is pressed, the final mobile application screen is presented, 1.1.4.

1.1.4 Sensor detail

The last screen of the mobile application will reveal the detail information about the sensors. It can be accessed by pressing one of the basic sensor icons, 1.1.3.4.

Read sensor value will be presented, next to the time that the value was measured, as figure 1.9 shows. Also, at the bottom of the screen, is presented a graph with the last 500 values that the specific sensor read. This values can be

5

User Technical Manual *User Technical Manual*

rearranged using the "To" and "From" field. Next image describes a sensor detail screen of temperature icon.

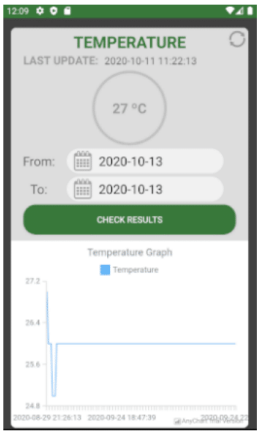


FIGURE 1.9: Sensor detail screen

1.2 Server

To transport the data from the components implemented in the system to the developed android application, it was necessary to use a server, a database and python scripts.

The server has the function of saving the data collected from the sensors, as well as the user authentication data.

The information from the sensors reaches the mobile application thanks to the communication with the server, which is done by sending and receiving MQTT messages. This exchange of messages is possible because the python scripts are running in the background and create this same connection.

1.2.1 Database

Database e constituted by two tables:

User Technical Manual*User Technical Manual*

- Users - contains the information that the user gives on sign up, 1.2. The idusers field represents the key that connects to the other table, that each user will have access;
- Sensors - when the sensors make a reading, it is published to the network and the server saves that data in the table, storing the value and type of the sensor, and also the date that the measurement was made.

Figure 1.10 represents the UML diagram of the tables created.

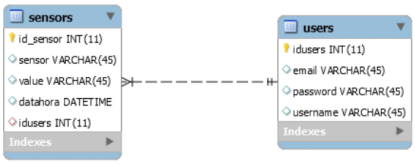


FIGURE 1.10: Database Relational Model

1.2.2 Scripts

To make the database information available in the mobile application, a python script is retrieving the data and presenting it in a html page, 1.11, where the android application will access, when in need. This script name is sensors_data.py.

FIGURE 1.11: Html pages

The script stores different information in four different html files, 1.11:

- allvalues - gives the information of the last 500 entries of each sensor, so the application can reproduce the graphs, 1.9;

User Technical Manual*User Technical Manual*

- distinctsensors - offers the most recent entry of each sensor;
- login - is used when the user is logging in the application, and the user and password are sent to the page, to make a connection to the database, checking if there is or not an entry with the values, and prints to the html accordingly.
- signup - when the user is making the register, in the mobile application, the information is sent to this page, making a succesfull insert in the database with the filled values in screen 1.1.2.

When sensors make measurements and sent it to the network, there is another script python running, mqtt.py, that is always waiting for a message from there. When receives a message, the script is ready to store that information on the database.

1.3 Arduino IDE

The hardware used to implement this system is compatible with Arduino IDE, that can be downloaded at https://www.arduino.cc/en/Main/OldSoftwareReleases.

There was a need to include some specific libraries, for the successful use of some of the system components, such as:

- DallasTemperature;
- Servo;
- OneWire;
- Wi-Fi;
- PubSubClient.

There was only one script .ino developed that was used to forward messages between the network and the node, obtain the data from the sensors and turn on or off the actuators, actions that were received from broker.

9

APPENDIX C

Scientific Contributions

Intelligent rainwater reuse system for irrigation

Maria M. S. Pires, André F. X. Glória, Pedro J. A. Sebastião

***Abstract*—**The technological advances in the area of Internet of Things have been creating more and more solutions in the area of agriculture. These solutions are quite important for life, as they lead to the saving of the most precious resource, water, being this need to save water a concern worldwide. The paper proposes the creation of an Internet of Things system, based on a network of sensors and interconnected actuators that automatically monitors the quality of the rainwater that is stored inside a tank, in order to be used for irrigation. The main objective is to promote sustainability by reusing rainwater for irrigation systems, instead of water that is usually available for other functions, such as other productions or even domestic tasks. A mobile application was developed, for Android, so that the user can control and monitor his system, in real time. In the application it is possible to visualize the data that translate the quality of the water inserted in the tank, as well as perform some actions on the implemented actuators, such as start/stop the irrigation system and pour the water in case of poor water quality. The implemented system translates a simple solution with a high level of efficiency and tests and results obtained, within the possible environment.

***Keywords*—**Internet of Things, Irrigation System, Wireless Sensor and Actuator Network, ESP32, Sustainability, Water Reuse, Water Efficiency.

Maria M. S. Pires is with the Department of Information Science and Technology, ISCTE-Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal (e-mail: misms@iscte-iul.pt).

André F. X. Gloria is with the Department of Information Science and Technology, ISCTE-Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal and Instituto de Telecomunicações, 1049-001 Lisbon, Portugal (e-mail: andre_gloria@iscte-iul.pt).

Pedro J. A. Sebastião is with the Department of Information Science and Technology, ISCTE-Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal and Instituto de Telecomunicações, 1049-001 Lisbon, Portugal (e-mail: pedro.sebastiao@iscte-iul.pt).