IoT-BASED AUTOMATED SURFACE-DRIP IRRIGATION MONITORING SYSTEM PROTOTYPE: A CASE OF WORLD VEGETABLE CENTER IN ARUSHA, TANZANIA

Joseph Ngigi Wangere

A Project Report submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

August, 2024

ABSTRACT

Irrigation is pivotal for supplementing water, amidst climate change globally and surface-drip irrigation provides an efficient plant-watering method with minimal mineral depletion. In Tanzania, Africa, manual surface-drip irrigation persists despite attempts at SMS-based automation, leading to time-consuming routine irrigation tasks. Existing studies focus on XBee modules in their transparent operation mode, limiting scalability due to lack of information such as source addresses as well as lack of support for multiple sensor nodes. This project, carried out in Arusha, Tanzania, employed a wireless sensor network using XBee modules configured in API mode for soil-data collection and transmission, utilizing Node-RED for remote actuator control and soil data visualization. By employing JavaScript functions, data bytes were extracted, including transmitter addresses, from each sensor node, enhancing scalability. Sensors measured soil moisture, temperature, humidity, air temperature, and ultraviolet radiation to determine irrigation requirements. The outcome, an IoT-based Automated Surface-drip Irrigation Monitoring System Prototype was tested against user and system requirements. Remarkably, the system reduced irrigation control time by 96%, compared to manual control, by automatically triggering on the pump and valves when minimum soil moisture threshold is reached and turning them off once the maximum moisture threshold is reached. It also supported over-the-air firmware updates, to reduce maintenance costs, enabling the integration of technologies like digital twins. The system can integrate with existing manual irrigation systems, offering a cost-effective automation solution. Future improvements could involve incorporating machine learning for crop water requirement forecasting, expanding the sensor node network and extensive testing under different farm environments.

DECLARATION

I, Joseph Ngigi Wangere, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this project report is my original work and that it has neither been submitted nor being concurrently submitted for degree award in any other institution.

Joseph Ngigi Wangere		
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Ramadhani Sinde		
Name of Supervisor 1	**Signature**	**Date**

Dr. Mussa Ally Dida		
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the Nelson Mandela African Institution of Science and Technology, a dissertation titled **“IoT-based Automated Surface-drip Irrigation Monitoring System Prototype: A Case of World Vegetable Center in Arusha, Tanzania”** in partial fulfillment of the requirements for the award of the Degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology.

Dr. Ramadhani Sinde

Name of Supervisor 1	**Signature**	**Date**

Dr. Mussa Ally Dida

Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGEMENTS

I would like to express my gratitude to God for enabling me to carry out this work and for the guidance in all phases of the project.

My sincere appreciation goes to the Centre of Excellence for ICT in East Africa (CENIT@EA) for the scholarship to pursue the Master of Science in Embedded and Mobile Systems (EMoS) as well as the financial support towards achieving the course and project objectives.

Additionally, I would like to recognize and appreciate the invaluable technical and academic advice, coupled with positive criticism offered by my academic supervisor Dr. Ramadhani Sinde. His feedback, guidance, encouragement and expertise motivated me to shape the direction and quality of this work. Many thanks to the co-supervisor Dr. Mussa Ally for his support as well.

I would also like to thank the World Vegetable Center in Arusha and its staff for the opportunity to carry out my research at the center and for their invaluable contribution to the study during requirements gathering, development, and testing. Many thanks to my industrial supervisor Mr. Omary Mbwambo for all the guidance and support.

I am also grateful to my family for their encouraging prayers and support throughout my study phase. I would also like to pass my regards to the EMoS third cohort class for the experiences shared, the new skills learned together, their moral support and their collaboration spirit.

Finally, sincere regards to the Nelson Mandela African Institution of Science and Technology (NM-AIST) for availing research facilities and providing a conducive environment for research and innovation.

In conclusion, this project would not have been possible without the support and collaboration of all the aforementioned. I am eternally grateful for their immense contributions and their impact on my academic journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

4IR	Fourth Industrial Revolution
5IR	Fifth Industrial Revolution
AC	Alternative Current
AI	Artificial Intelligence
API	Application Programmable Interface
AVRDC	Asian Vegetable Research and Development Center
BLE	Bluetooth Low Energy
BOM	Bill of Materials
CAD	Computer-Aided Design
CAN	Controller Area Network
CENIT@EA	Centre of Excellence for ICT in East Africa
DC	Direct Current
DHT	Digital Humidity and Temperature Sensor
EDA	Electronic Design Automation
ET	Evapotranspiration
F/W	Firmware
FAO	Food and Agriculture Organization
GPH	Gallons Per Hour
GPIO	General Purpose Input Output
GPRS	General Packet Radio Service
H/W	Hardware
HTTP	Hyper Text Transfer Protocol
I2C	Inter-Integrated Circuit
ICT	Information and Communication Technology
IDE	Integrated Development Environment

IEEE	Institute of Electrical and Electronics Engineers
IITA	International Institute of Tropical Agriculture
IoT	Internet of Things
JSON	Java Script Object Notation
JTAG	Joint Test Action Group
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LoRA	Long Range
LoRA WAN	Long Range Wide Area Network
LTE	Long Term Evolution
MATLAB	Matrix Laboratory
MQTT	Message Queuing Telemetry Transport
NM-AIST	Nelson Mandela African Institution of Science and Technology
NM-IoT	Narrow Band Internet of Things
OTA	Over-The-Air
PCB	Printed Circuit Board
RF	Radio Frequency
RISING	Research in Sustainable Intensification for the Next Generation
S/W	Software
SAROS	Smart Agricultural Research Optimization System
SCADA	Supervisory Control and Data Acquisition
SMS	Short Message Service
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
TCAA	Tanzania Civil Aviation Authority
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicles

UML	Unified Modelling Language
USAID	United States Agency for International Development
WSN	Wireless Sensor Network
WSN	Wireless Sensor Network
WUCOLS	Water Use Classification of Landscape Species

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

Agriculture plays a crucial role in supporting livelihoods since it is the backbone of food security. This is particularly dominant in third-world countries where issues such as overpopulation and climate change have continually destabilized efforts to alleviate poverty and food insecurity (Pawlak & Kołodziejczak, 2020). The global population has been on an upward trajectory in recent years, extending further the challenge of food insecurity (FAO, 2021). This phenomenon can be seen in Fig. 1 which illustrates statistics of the rise in hunger between the years 2014 and 2019. In the year 2019 alone, 60 million more people were exposed to hunger compared to the year 2014.

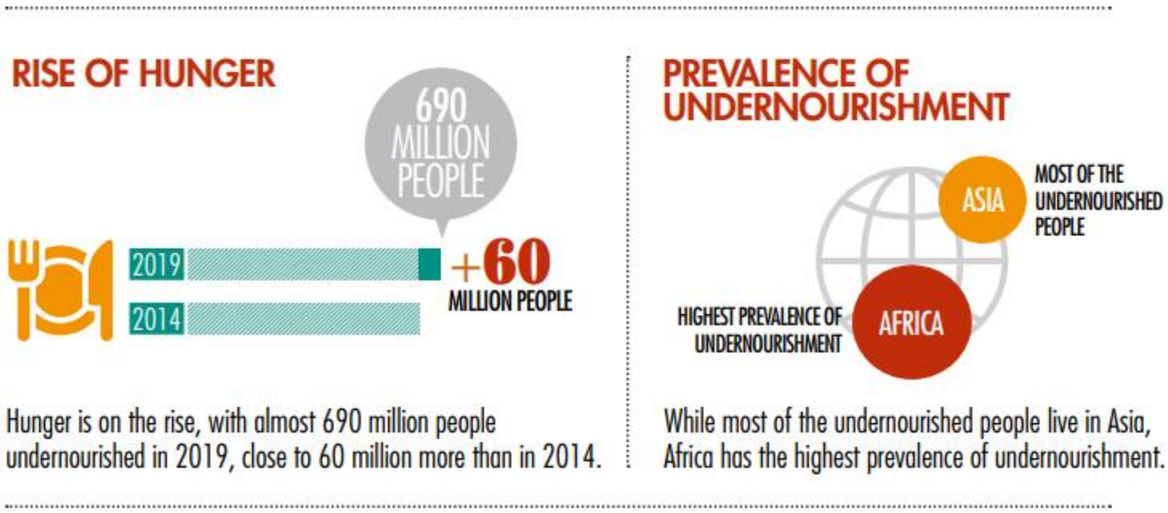


Figure 1: Global hunger prevalence

There have been global efforts to combat hunger and ensure minimum nutrition requirements for human beings are met. One such initiative is the Zero Hunger Challenge initiated by the United Nations Secretary-General in 2012. The main pillars of the Zero Hunger Challenge initiative are to combat hunger, eliminate poverty in rural regions by increasing the income as well as productivity of small-scale farmers, eliminate food wastage, increase access to healthy diets as well as adequate food all year round for everyone, get rid of undernutrition and develop food systems that are sustainable as well as inclusive (United Nations, 2012).

Figure 2: Elements of the zero-hunger challenge vision

Despite ongoing efforts globally to combat hunger, research continues to prove that there’s still much to be done if the United Nations' second sustainable goal is to be achieved. This situation was worsened by the COVID-19 global pandemic and as the world strives to recover, there is a high likelihood that by the year 2030, the world will not have achieved the zero hunger goal (Cooper *et al.*, 2021).

Due to the change in climate as a result of factors such as global warming, the world has been experiencing inconsistent rainfall patterns that cannot be solely relied upon for sustainable agriculture. Studies continue to reveal that the impact of the drought will be even more hazardous in the coming days (Wu *et al.*, 2022).

As the world continues to witness inconsistencies in rainfall patterns and general climatic conditions, there is a need for climate-smart agricultural methods to ensure continuous farm yield amidst harsh climatic factors. Some authors have expressed fears over what they term as little attention to issues such as climate change impacts on agriculture among smallholder farmers, biodiversity versus agricultural conservation goals as well as a reluctance to invest in research and development by policymakers (Dorward & Giller, 2022). Among the adopted technologies to counter climate changes include the development of more resistant crop species as well as resource management tools to ensure proper utilization of water, land, and energy.

Large-scale, as well as smallholder farmers, have adopted irrigation to supplement the insufficient rainfall and ensure there is crop yield throughout the year regardless of the season. However, most irrigation systems used on a large-scale basis are manual. Off-the-shelf automatic irrigation systems do exist but the cost of replacing the entire manual irrigation system with an automatic one is very high. Additionally, such off-the-shelf automatic systems attract high subscription and maintenance fees annually.

Irrigation is the artificial application of water to plants using a network of pipes, valves, pumps, and any other relevant technology. Manual irrigation systems have proven to be less efficient due to their lack of a mechanism for water preservation despite their affordability initially. Previous research has shown that Manual Irrigation systems consume more water per square meter compared to automatic irrigation systems (Ançay *et al.,* 2017).

There has been ongoing research in recent years on how to automate irrigation processes. The Application of the Internet of Things (IoT) has become a hot topic in this subject globally. IoT describes a blend of sensing devices, connectivity mediums, data storage, and visualization platforms (Abbassi & Benlahmer, 2021).

In this epoch of the Fourth Industrial Revolution (4IR), wireless sensor networks have gained global popularity owing to the development of infinitesimal, low-cost sensors that can detect and transmit desired parameter data while occupying minimum space on circuit boards, with minimum power requirements achieved from the sleep mode capabilities. Furthermore, advanced algorithms have been proven to minimize the energy consumption of these wireless sensor networks while enhancing their lifetime with minimal interruptions (Sinde *et al.*, 2020).

The World Vegetable Center, initially known as the Asian Vegetable Research and Development Center (AVRDC) is one among many centers that are contributing to the Zero Hunger Challenge. The center, whose headquarters are in Taiwan but with a global presence, was established in the year 1971, with the main objective of poverty and malnutrition alleviation by increased production as well as consumption of nutritious vegetables. The center achieves its mission by providing farmers with informed agricultural practices, post-harvest value addition as well as marketing for a variety of vegetables. In addition to contributing to employment creation and income generation for small-scale farmers, the center promotes diversity in crop farming and advocates for better-balanced diets. Their efforts are geared towards combating lifestyle diseases caused by a lack of vegetables in the diet (World Vegetable Center, 2013).

The Center, based in Arusha, Tanzania utilizes an integrated manual irrigation system consisting of a drip network as well as open channel irrigation to ensure continuous vegetable production all year round for different varieties. Furthermore, there is an existing hydroponic section, where soilless irrigation is carried out. Hydroponic technology involves cultivating plants without using the soil (Ezzahoui *et al.,* 2021). The hydroponic section at World Vegetable Center is semi-automated in that the pump is switched on and off remotely using SMS commands on a mobile phone.

In this study, the existing irrigation system used by the World vegetable Center, in Arusha located in Tanzania was examined. Water was found to be obtained from a natural river source using gravity and preserved in an open reservoir. Some of the water was observed to be channeled directly to the farm through irrigation canals. From the reservoir, water was found to be pumped to the irrigation fields using an electrical pump to a filtration chamber, to eliminate dirt and then to the different irrigation zones via drip pipes. Field staff had to manually switch on the water pump, then proceed to turn on the water valves for each irrigation farm sector.

Additionally, the farm's staff were found to rely on intuition and mere observation to determine when to irrigate the different vegetable field sectors and for how long. Other times, the farm zones were observed to be irrigated based on a fixed schedule, but there was no existing framework for using reliable data to determine when and how much irrigation is required at any given time. The illustration in Fig. 3 depicts the present situation at World Vegetable Center.

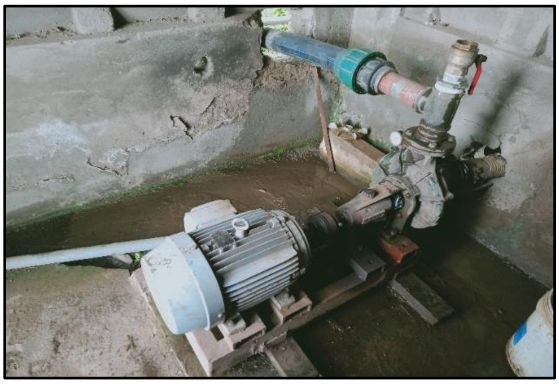


Figure 3: Manual water pump (left) and manual switch-box (right)

To activate the water pump illustrated on the left of Fig. 3, the contactors of the manual switches illustrated on the right of Fig. 3 have to be pressed manually. Besides this process being inefficient, it poses a safety threat due to the risk of electric shock if a user presses the switch with wet hands. After switching on the water pump, a field staff proceeds to turn on a series of gate valves in the irrigation zones manually. These gate valves as presented in Fig. 4 are approximately 15 in total, and they are distant from each other, hence it takes approximately one hour to go around manually switching all of them.

Figure 4: Manual gate valves

In addition to the manual operation of the aforementioned electrical and mechanical parts, the center has an array of electronic hand-held devices used to collect data regarding weather parameters. They include temperature, humidity, wind speed, ultraviolet radiation as well as soil-pH data loggers as illustrated in Fig. 5. The main drawback of such devices is that they are not backed by a software interface that users can access remotely to retrieve the data. The implication is that each device has to be individually connected to a computer via a USB interface to retrieve data. This process is inefficient in terms of time consumption and it is tedious to keep track of data especially when a user is responsible for organizing the folders in a computer, like in the case of the World Vegetable Center.

Figure 5: Stand-alone weather data loggers

The outlined manual methods are inefficient, unreliable, and prone to error. The water consumption is also high, and it can only be estimated as there are no monitoring mechanisms. Due to the possibility of pumping water unnecessarily, the operational cost is high resulting from unwarranted electricity bills. Furthermore, there is no record of how much water is used at regular intervals. It is nearly impossible to quantify the loss incurred. Unfortunately, there must be a farm staff always on-site to go around monitoring the vegetable fields to determine whether or not to start irrigating. The whole process is outdated, more so in this era of 4IR, where automation is at everyone’s disposal. Without proper data-driven techniques of applying irrigation, the organization is prone to make losses in terms of electricity bills by running the water pump unnecessarily as well as water wastage, but also extra labor costs, and misuse of time.

In this study, therefore, an IoT-based automated surface-drip irrigation monitoring system was developed. A wireless sensor network based on ZigBee was integrated with the system. The wireless sensor network consists of soil moisture sensors, soil temperature sensors, ambient humidity plus temperature sensors, rainfall sensors, water-flow sensors, as well as ultraviolet radiation sensors. Data from the mentioned sensors are processed and analyzed from a sink node to provide logic for automatically activating the switching on and off of the water pump as well as the opening and closure of solenoid valves to regulate the flow of water in different irrigation fields. A wireless sensor network describes an interconnection of Sensor devices that monitor different parameters and relay the sensor data to a central information processor for further control and decision-making.

1.2 Statement of the Problem

The current global population increase has raised the depletion of resources such as land and water. Therefore, the concept of urban agriculture has been developed to ensure sustainability and food security (Langemeyer *et al.,* 2021). This has seen cities engage in food production, distribution, and marketing to combat malnutrition.

As cities globally continue to evolve, in the 4IR, a need has been identified for automation to facilitate urban agriculture as well as other sectors. Previous studies have been conducted on-farm automation to improve efficiency and optimize natural resources such as water.

However, the world is entering industry 5.0 which will be mainly characterized by the interception between human expert innovation and their interaction with machines to improve production efficiency (Maddikunta *et al.,* 2022). Virtual reality as well as digital twin technologies are among the key components of the fifth industrial revolution (5IR). The digital twin is a recent paradigm that relates to the virtual representation of a physical phenomenon digitally to simulate how a physical system would behave (Nasirahmadi & Hensel, 2022).

Therefore, manual systems will be defunct over time. In the agriculture domain, manual irrigation systems are labor intensive, consume time, and lead to water wastage as well as energy. This situation is worsened by a lack of monitoring mechanisms for determining water consumption per irrigation field, as well as soil and climatic data.

This project addresses these challenges through an IoT-based automated surface-drip irrigation monitoring system. A low-power XBee-based wireless sensor network in API mode was deployed to collect soil sensor data from different nodes and transmit it to a central node. A remote dashboard was provided to ensure real-time data transmission as well as control of actuators like pumps and valves, even remotely.

1.3 Rationale of the Study

According to Casadei *et al.* (2021), irrigation can be classified as an efficient way of conserving water resources. The current growth in the global population has caused water to become a scarce resource, therefore creating the need for managing it to ensure proper conservation and optimal use. Furthermore, manual irrigation systems do not provide the required efficiency and precision in terms of water conservation. Monitoring water consumption for manual irrigation systems is tedious as there is no digital data collection method. Reading old manual water meters is tiresome and time-consuming. Such manual systems can irrigate the farm erroneously due to a lack of an actuation mechanism that is activated by reliable soil parameter data as collected by sensors. Additionally, the inefficiency of manual irrigation systems is worsened by their reliability on human intervention.

Therefore, this study entailed the development of an IoT-based automated surface-drip irrigation monitoring system prototype, using an XBee-based wireless sensor network, to measure soil parameters and control the actuation of water pumps and solenoid valves, with remote operation capability through the Node-red platform.

1.4 Objectives of the Study

1.4.1 Main Objective

To develop an IoT-based automated surface-drip irrigation monitoring system prototype for World Vegetable Center in Arusha Tanzania.

1.4.2 Specific Objectives

The study aimed to achieve the following specific objectives:

(i) To identify the functional and non-functional requirements for the development of an IoT-based automated surface-drip irrigation monitoring system prototype.

(ii) To develop an IoT-based automated surface-drip irrigation monitoring system prototype.

(iii) To validate the developed IoT-based automated surface-drip irrigation monitoring system prototype.

1.5 Research Questions

The study intended to answer the following questions:

(i) What are the functional and non-functional requirements for the development of an IoT-based Automated Surface-Drip Irrigation Monitoring System prototype?

(ii) How can the IoT-based Automated Surface-Drip Irrigation Monitoring System prototype be developed?

(iii) How would the developed IoT-based Automated Surface-Drip Irrigation Monitoring System prototype be validated?

1.6 Significance of the Study

The output of this system is a promising contribution towards natural resource conservation efforts by making proper use of available water. There is potential to reduce recurring bills such as electricity and labor for farmers.

The use of automation and remote-control capabilities suggest pathways to improve efficiency by saving time as well as labor that is spent moving from point to point to manually control valves and pumps (Abioye *et al.,* 2021).

In the long run, employing a data-driven approach to irrigation planning, monitoring, and controlling can improve accuracy in precision irrigation by providing the required amount of water for different irrigation fields. An analysis of overall water consumption can be provided at regular intervals depending on user requirements.

1.7 Delineation of the Study

Irrigation is a broad term that is an umbrella of many artificial techniques of supplying water to plants at root zones. In this study, a surface-drip irrigation system has been considered. Drip irrigation stands out as the most preferred water delivery mechanism due to its ability to deliver water and nutrients directly to the plant's root zone. Other categories of irrigation such as sprinklers or canals have not been considered in this work, due to their dependence on high amounts of water and their complexity in automation. Two remote nodes and a central coordinator node were considered, but with the possibility of replicating and increasing the nodes if it is desired. Compared to existing irrigation kits, the cost of this prototype is in the range of EURO 1000-1500. This cost could be further reduced after extensive testing and coupling of all components into a single processor board for the sensor nodes. Existing irrigation kits cost in excess of EURO 2500 exclusive of shipping to Arusha, Tanzania.

CHAPTER TWO

LITERATURE REVIEW

2.1 Search Strategy

To conduct the literature review, existing related reviews were first identified. This was done by first searching through well-structured databases of published works such as ScienceDirect, Web of Science, SpringerLink, and Research4Life. Later, the search was extended to less structured databases and other opensource resources. Grey literature was included for items such as reports, thesis, and dissertations of previous researchers from academic institutions. The search was finally spread to search engines including Google Scholar and Microsoft Academic. Due to the agile nature of the project, the literature review was done throughout the entire lifecycle of the project.

The language considered for this search was English. The literature was limited to ten years backwards from the time of carrying out this project for Journal articles, to obtain the latest existing works related to automated irrigation systems. However, related works from books sections were also used outside this specified ten-year time-frame. A sample search string as used to retrieve literature from ScienceDirect was formed using controlled vocabulary as follows: irrigation[mesh] OR "internet of things" OR "soil moisture" OR "irrigation systems" OR crops OR "moisture control" OR "agricultural robots" OR "automation" OR "IoT" OR "automatic irrigation."

While the search strings are not case sensitive, the operators 'OR', 'AND', 'NOT' are case-sensitive. The OR operator is used to merge alternative terms and to increase the results base. AND on the other hand, is applied when combining concepts while NOT is used to exclude specified search terms. All the literature was then filtered by reviewing the Titles and abstracts to determine relevance to this study.

2.2 Theoretical Literature Review

Humanity is dependent on nutrients to thrive and lead a healthy life. Nutrients are obtained from food substances. Both Human beings and livestock rely on plant foods to obtain these necessary nutrients. According to data from World Bank, the world population as of the year 2020 was approximately 7.762 billion (World Bank, 2020). If such statistics are anything to go by, the implication is a rapid increase in population globally. Consequently, there is an expected depletion in natural resources such as water and land. With the scarcity of vital resources like water, due to issues related to global warming and population growth, methods for water conservation are inevitable. One way of ensuring continuous Agricultural productivity amidst scantiness of water is employing irrigation techniques.

Different factors come into play when determining plant water requirements and the irrigation method to use. These factors include but are not limited to the plant type, soil type, depth of water table, plant growth stage, climatic conditions, and availability of water (Ali, 2010).

Elshaikh *et al.* (2018) carried out research work to review theories used in the evaluation of irrigation performance. One theory was based on direct measurement of indicators, which they found to be uncertain in terms of the collection of reliable data and was equally subjective. According to this research, remote sensing was also studied and found to lack much usefulness in the presence of clouds, in addition to requiring technical advanced skills. Feasibility of drone use in Africa, specifically Tanzania is still facing slow uptake due to the bureaucratic processes of obtaining legal permits as well as the cost implications of insurance, operational training, and license fees according to the Tanzania Civil Aviation Authority (TCAA, 2021).

Nechifor and Winning (2019) studied the impact of climate change and the increase in carbon (IV) oxide in the atmosphere on the performance of crops and their water requirements for irrigation. The study revealed that pronounced changes in climate result in diminishing crop output at the same time depressing water demand globally for crops that depend on irrigation, despite there being an increase in the intensities of irrigation water.

From a theoretical point of view, different theorists as reviewed in this section generally propose the use of irrigation to supplement the water requirements of plants and ensure sustainable farming throughout the different seasons of the year. There are different categories of irrigation methods ranging from drip, sprinkler, and furrow.

2.2.1 Drip Irrigation

Drip irrigation, as presented in Fig. 6, entails the application of water and sometimes nutrients directly to the soil along plant rows, close to the plant roots using narrow flexible tubes made of plastic that are laid on the soil surface. Surface drip irrigation is common, where the plastic tubes are placed on the soil surface. Subsurface irrigation on the other hand involves placing the tubes immediately underneath the soil surface to minimize lateral movement. The tubes have even holes that allow water to trickle to the soil vertically. Drip irrigation is preferred due to its ability to minimize water consumption, high efficiency allowing for easy fertigation thus minimizing fertilizer costs, flexibility and ease of automation to reduce labor, low risk of soil erosion, and minimal interruption by other farm activities such as harvesting as well as minimized weed invasion. Drawbacks of drip irrigation include capital investment, damage by rodents, the high maintenance cost of replacing tubes every season as well as poor frost protection (Hochmuth, 2017).

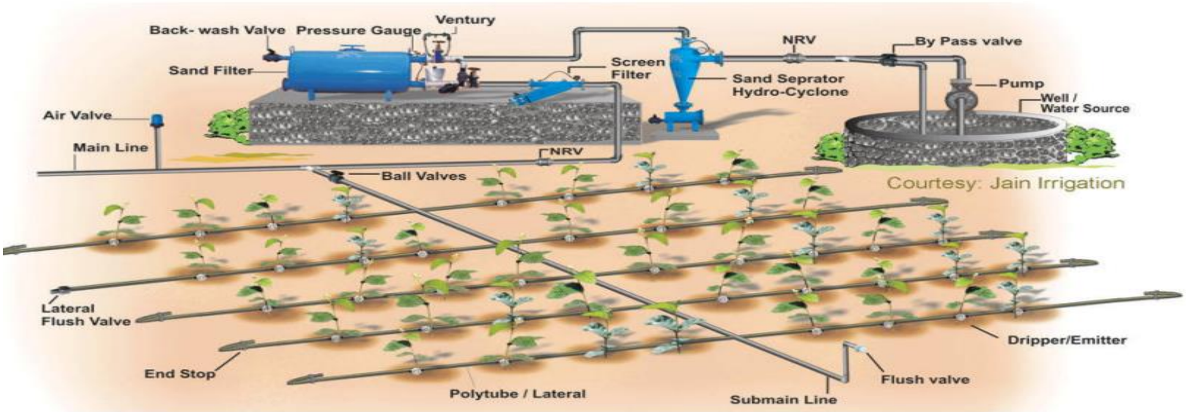


Figure 6: Drip irrigation layout overview (Jarwar *et al.*, 2019)

2.2.2 State of Art in Irrigation Automation

Traditional irrigation methods have received criticism over inefficiencies in terms of time as well as water wastage. Following recent rapid technological evolution, many researchers are formulating mechanisms for automating irrigation.

There are standard existing kits already in the market, many of which utilize Wi-Fi to relay field sensor data to a central controller that actuates pumps and valves (Hunter Industries, 2018). Wi-Fi boasts of high bandwidth, long transmission range, transmission speeds higher than other technologies such as ZigBee as well as non-line of sight capabilities of transmission (Li *et al.*, 2011). However, such performance hinders the effectiveness of wireless sensor networks in applications where energy efficiency is a determining factor, such as battery-powered devices.

Automatic drip irrigation has exhibited the potential for minimizing water consumption in irrigated fields by up to 15%. Soil-moisture sensors have been pivotal in the process of automating irrigation, by measuring the real-time water content in the soil and relaying the data to a central controller that initiates the process of applying water (Panigrahi *et al.*, 2019).

Different research and proof of concept systems have been developed to automate irrigation for different crops. To achieve optimum results, different technologies have been proposed. These range from Wireless Sensor Networks (WSN), Internet of Things (IoT), Unmanned Aerial Vehicles (UAV), Artificial Intelligence (AI) as well as cloud computing (Abdollahi *et al.*, 2021).

The WSNs leverage wireless communication technologies to relay data collected from sensors to a central hub or the cloud. They are autonomous and self-monitoring in nature; hence they are optimized to run for a long duration.

Emerging low-power wireless communication technologies for WSNs range from ZigBee, long-range wide area network (LoRa WAN), Narrow-band IoT (NB-IoT), Sigfox, and Long-Range (LoRa), Bluetooth Low Energy (BLE) as well as radio frequency communication. The choice of each depends on factors such as bandwidth, data rate, coverage, frequency-band as well as battery life. Sigfox operates on the unlicensed frequency and supports packets of small sizes of around 12 bytes. Despite being an extremely low-power technology, it has limitations in terms of the number of messages for both uplink and downlink, hence the low data rate. Sigfox is also associated with high implementation costs. Likewise, NB-IoT is a low-power technology characterized by high data rates. However, it relies on LTE coverage and has high implementation costs. ZigBee on the other hand is a low-power personal area network technology. The XBee modules have mesh network capabilities, despite high implementation costs. It has a high data rate, a maximum of 250 Kbps but is limited in coverage and performs best on flat surfaces (Klaina *et al.*, 2022).

2.3 Empirical Literature Review

In the following section, a review has been carried out of the various irrigation automation technologies proposed by researchers previously, critically analyzing their methods, overall strengths, and limitations.

Irrigation describes a process that involves applying water to the soil artificially. An irrigation system on the other hand integrates the various components involved in artificially applying water to the soil. Such components include but are not limited to pumps, valves, tubes, Power, sprinklers as well as water reservoirs (Bjorneberg, 2013). In recent times, technology has advanced in the development of irrigation systems to include an array of sensors and mechanisms for actuating valves and pumps automatically.

Selmani *et al.* (2018) designed an irrigation system that utilized multithreading, powered by Solar. Their design utilized concurrent programming to provide accurate and real-time management of Greenhouse irrigation activities. The design employed a Raspberry Pi microcomputer, DHT22 temperature humidity sensor, a water level sensor for monitoring water levels in the tank, and Soil moisture sensor. MySQL was used as the database server. A web server was utilized for remote access to the sensor data. Vert X tool was used for multithreading features, combining the pump control process with remote monitoring of pump activities. However, only one environment was monitored and controlled, which is an experimental Greenhouse. There is no mention of how to monitor different areas that have different irrigation requirements.

Mohammed *et al.* (2021) proposed a Smart Irrigation System focusing on subsurface irrigation for the enhancement of date palm irrigation management. Their automated system was based on IoT for control. Wireless sensor networks were deployed to collect and disseminate data regarding soil moisture content and weather. ThingSpeak open-source cloud data aggregation and analytics platform were used for visualization, scheduling, and real-time notifications. The framework of this solution consisted of a water pump and water source, an interconnection of pipes, disc filters, and valves as well as a power source (solar and charging unit) and control unit. The sensors used were a Flow meter, a DHT11 temperature humidity sensor, and anemometer for monitoring wind speed as well as a solar cell. The control unit consisted of a NodeMCU microcontroller, relay, and contactor. The MATLAB analysis was performed using ThingSpeak cloud and data was relayed to a user via a React-based Mobile Application. TalckBack App was used to issue commands to the NodeMCU microcontroller to switch the solenoid valve as well as the pump on and off. Despite the system conserving water consumption and increasing the date palm yield, there was one major drawback, related to power consumption. All the sensor nodes in this system were configured to use NodeMCU which sent data to ThingSpeak using Wireless Fidelity (Wi-Fi) technology. The Wi-Fi Technology which is based on IEEE 802.11 protocol, consumes more power than other wireless sensor technologies such as Bluetooth Low Energy as well as ZigBee which is based on IEEE 802.15.4 protocol, due to large bandwidth.

Gutierrez *et al.* (2014) used Wireless Sensor Network integrated with GPRS to design an automated irrigation system. The system used sensors for detecting temperature and moisture in the soil at the plant root zone. This system was powered using photovoltaic panels. A central controller unit was configured to process sensor data, control actuators such as pumps, and transmit sensor data to a web application. Scheduling of the irrigation could be controlled from a web page. The system saved water up to 90% in comparison to other systems. PIC24FJ64GB004 was used. Wireless sensor networks were configured to transmit sensor data to the central unit using an XBee Pro S2C module that operates on ZigBee IEEE 802.15.4 protocol. The central unit then was configured to transmit data to the cloud using GPRS cellular technology. The potential drawback of this system was the additional cost of airtime required for the GPRS module SIM card. GPRS is also unsuitable for real-time control and monitoring due to low data transmission rates practically as well as transmission delays compared to other wireless standards such as Long-Term Evolution (LTE).

Vera *et al.* (2021) conducted some research with the aim of irrigation automation using dielectric soil sensors. This study explored the use of electromagnetic sensors to detect volumetric soil water content and regulate solenoid valves for irrigation triggering. The study depicted savings in terms of labor, energy, and water using the proposed system. Potential drawbacks of the study included the limited soil volume that the electromagnetic sensors could capture. The high cost of acquiring the sensors and additional Supervisory Control and Data Acquisition (SCADA) machinery and software was highlighted.

Vimal *et al.* (2021) proposed an Agricultural Smart System for Irrigation using the drip method. The primary mechanism involved detecting the level of moisture in the soil and activating the water supply required as per arising needs. The technology used included an ATmega328 microcontroller, GSM module, Humidity sensor as well as soil moisture sensor. The limitation of this study was the lack of remote data processing as it is purely SMS-based. There was no provision for accessing historical sensor data.

Masseroni *et al.* (2020) reviewed how varied technologies of smart irrigation are applied in different research areas such as precision irrigation control models, natural resource prices, and Smart Irrigation using Information Communication Technologies (ICTs). Among their key findings was the integration of flow meters for controlling water distribution in real-time within the irrigation canals. Their key findings on the latest technologies perceived by farmers as utilities for improving irrigation efficiencies included automated notifications, control using mobile devices, integrating irrigation systems with weather forecasting and alerts, use of soil moisture sensors as well as model tools for gathering crop water requirements.

Tiglao *et al.* (2020) proposed the application of a meshed wireless sensor network in farms, responsible for collecting soil data in addition to the actuation of the water pump as well as valves. However, the study made use of nRF24L01 wireless radio modules, whose power consumption is higher compared to other wireless technologies such as ZigBee and LoRa due to lack of support for sleep mode, hence unsuitable for meshed wireless sensor networks.

Another study carried out by Canales-Ide *et al.* (2019) focused on the assessment of Subsurface Drip Irrigation as a case study of smart irrigation in an urban setting in Valdebabas, located in Madrid, Spain. The efficiency of water use was compared by using two methods. The first method involved scheduling of irrigation by the park manager, using plant water requirements that are derived from the evapotranspiration (ET) reference from a weather station nearby and a plant coefficient (K_c). The second method involved estimating the water requirements of plants using the Water Use Classification of Landscape Species (WUCOLS) III, which bases the water requirements of plants on the plant species composition as opposed to a single species. The study revealed better irrigation management using Park Manager's estimations compared to the second method. The study went further to show that automated irrigation has the potential to save water consumption extensively. The limitation of this study, however, was that it was only based on climatic data and did not make any considerations on other variables, including the water content of the soil. Monitoring of the on-site water content of the soil using soil moisture sensors could further aid in precision agriculture. Data regarding soil water content can be transmitted to the irrigation control unit station and be utilized in determining the amount and frequency of irrigation.

Domínguez-Niño *et al.* (2020) in their paper carried out research intending to test how suitable the FAO's method (Water balance) can be when integrated locally with sensors to trigger the irrigation scheduling algorithm. Their paper also tested how suitable capacitive soil moisture sensors are and how they can be integrated with automatic interpretation for effecting irrigation scheduling algorithms. The performance of all the mentioned approaches was also tested. Results from the automated scheduling were compared with manual scheduling as illustrated in Fig. 7, using evapotranspiration which was the actual value determined using a weighing lysimeter within the same locality. The overall observation was that the water-balance method integrated with capacitive soil moisture sensors has the potential of accurately scheduling automatic irrigation systems in orchards.

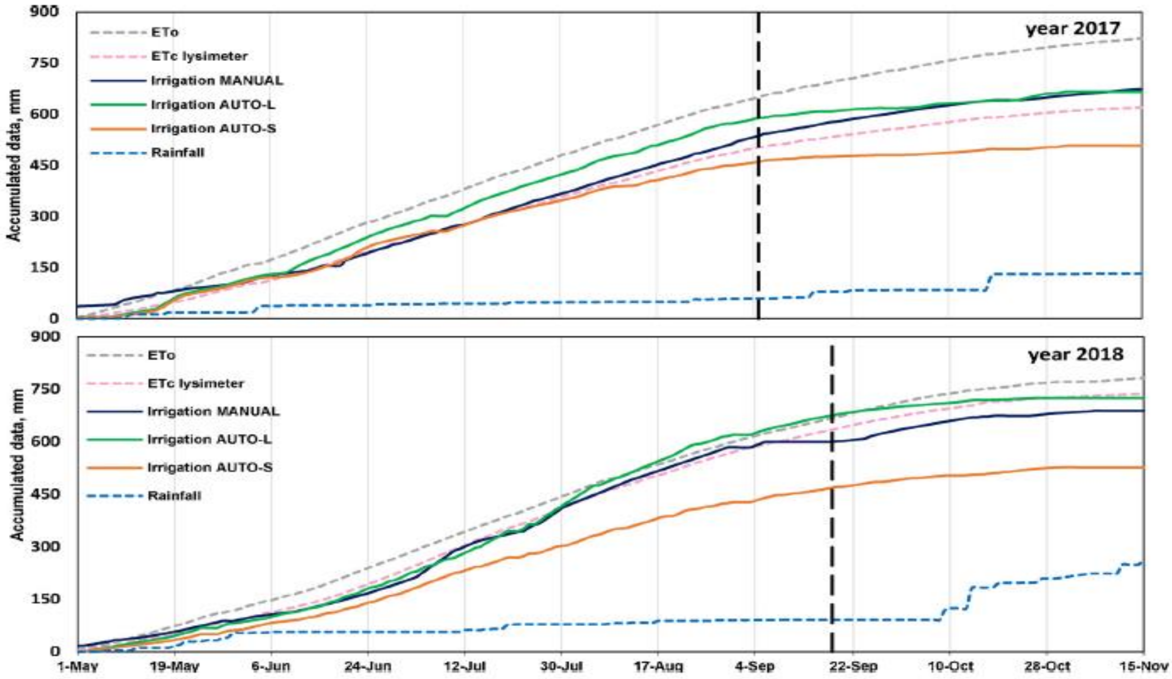


Figure 7: Comparing response of different irrigation systems (Domínguez-Niño *et al.*, 2020)

According to Domínguez-Niño *et al.* (2020), combining the water-balance method with sensors has several advantages. This integrated method has a smoother response. Due to automation, it is more predictable than manually switching valves on and off. Additionally, different irrigation sections can be controlled independently using sensors daily without interfering with the drainage distribution of the entire farm. However, the capacitive soil moisture sensors recommended would require to be calibrated when switching between soil types.

Vij *et al.* (2020) carried out a study to address the problem of over-irrigation, crop-specific irrigation as well as soil erosion using distributed networks of wireless sensors and algorithms based on machine learning for supporting irrigation pattern prediction. However, potential drawbacks of this proposed work range from the accuracy of the solution to relying on massive and region-specific data to train the model. This hinders the scalability of the solution. Additionally, in this work, Arduino microcontrollers were used in the sensor nodes, integrated with Wi-Fi modules. Wi-Fi transmission consumes a lot more power, hence replicating it on all sensor nodes could potentially result in more consumption of electrical power.

Koech and Langat (2018) in their work reviewed advancement efforts made to improve the efficiency of water use, specifically in Australia and other selected countries. Their study highlighted overall hindrances as well as opportunities that could be exploited. Among the emergent technologies in the domain of automatic irrigation scheduling as mentioned by Koech and Langat (2018) are remote sensing technologies such as satellite imagery, which is a key enabler due to its ability to cover larger geographical areas. Drawbacks of remote sensing include discrepancies in resolution temporally and spatially, thus degrading the quality of results. Remote sensing to determine crop water requirements using multispectral and thermal imagery via drones was reviewed. However, the uptake of drones is still very low and the cost of training, licensing, and purchasing is still high.

A model of a system for managing irrigation, automated with a network of wireless sensors was proposed by Haule and Michael (2014). Despite using a ZigBee-based wireless sensor network, there is no mention of any remote-control capabilities as well as integration of real-time remote access to sensor data. Additionally, there is no mention of water consumption monitoring. This system lacks important analysis required for monitoring and planning purposes.

A similar study was carried out by Njuu (2016), as a Master's dissertation from Nelson Mandela African Institution of Science and Technology. This proposed solution was monitoring irrigation water consumption in canals using XBee wireless sensor network and providing the analysis data using the cloud. This work has no mention of automatic control of irrigation pumps and valves. The focus was only on monitoring water levels in canals.

Koech and Langat (2018) also reviewed the use of Wireless Sensor Networks to collect soil parameters and relay them in real-time to a coordinator hub, from where the actuation of valves and pumps can be initiated automatically. Wireless Sensor Networks are gaining global attention due to the availability of low-cost sensors and their low energy requirements.

Another system was proposed by González-Perea *et al.* (2019). Their argument emphasized that farmers face difficulties acquiring the sophisticated technology required for real-time scheduling of irrigation. Their study implemented a middleware to avail a simpler tool to enable farmers to make informed decisions regarding the scheduling of irrigation using an irrigation module that was based on smart photovoltaic technology. Their solution integrated a user interface with data storage and connections to data sources remotely such as the soil sensors, irrigation control unit, and web platform. The drawbacks of their proposed solution include the full reliance of the water pumps on solar energy. Solar energy is not constant throughout the day and therefore, it could prove to be unreliable.

Subdividing a farm plot into different irrigation zones is an important technique when dealing with automatic irrigation systems. This is because different plants have different water requirements depending on their growth stage as well as their type. In this regard, some researchers have proposed solutions to take care of zoning. A similar system was developed using an Arduino Nano for the sensor nodes and a Raspberry Pi for the sink Node. A wireless sensor network was set up using RF technology and a human-machine interface was designed using Node-red technology. Water application was controlled by solenoid valves. While the system took care of the different plants, there was no mention of how to measure the water consumption. Furthermore, only three parameters were considered, one being soil moisture and the other air temperature as well as humidity (Benyezza *et al.,* 2021). The NRF24L01 radio modules used in this wireless sensor network have a far much higher energy consumption making them unsuitable for wireless sensor networks compared to similar technologies such as XBee (Saha, *et al.,* 2017).

2.4.1 Summary of Important Parameters Monitored

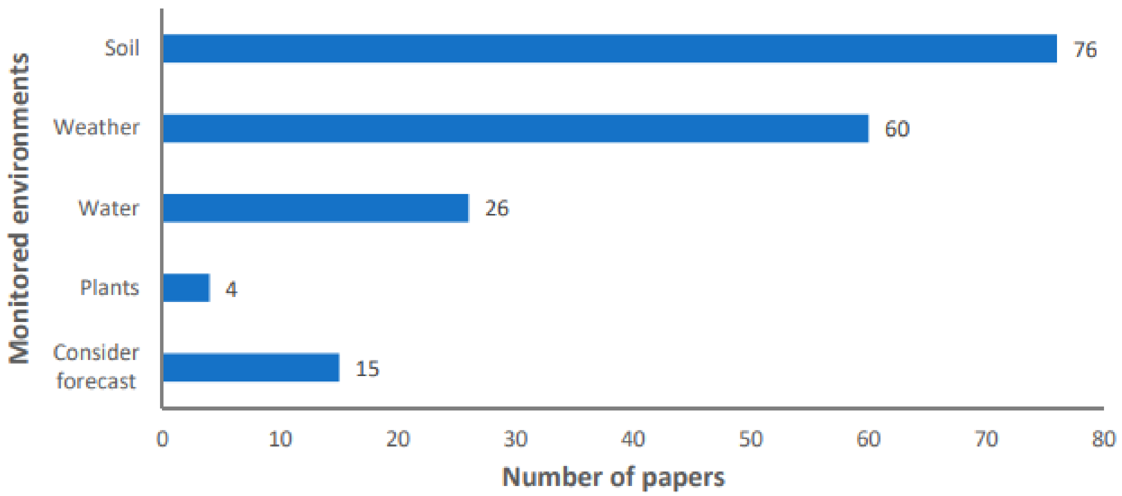


Figure 8: Summary of monitored parameters (García *et al.*, 2020)

A review carried out by García *et al.* (2020) analyzed the most commonly monitored parameters in irrigation systems as presented in published journal papers. Their findings, as illustrated in Fig. 8, revealed that soil moisture monitoring had the highest number of publications followed by weather parameters such as temperature, humidity, and rainfall. Other papers addressed water parameters, and plant water requirements and even considered forecasting.

With the advent of technological advancement, soilless farming is proving to offer immense capabilities in contributing to sustainable urban food security (Gumisiriza *et al.,* 2022). Hydroponic Agriculture has emerged in urban regions, offering flexibility for urban households who have no pieces of land to practice farming. Montesano *et al.* (2018) conducted an experiment that revealed Wireless Sensor Networks can be used to monitor the moisture content of substrates in real time to offer precise automation of irrigation in soilless greenhouses.

Benchmarking was carried out with a similar project that was previously carried out at Gallapo village located in Babati District, part of Northern Tanzania, under the Africa RISING (Research in Sustainable Intensification for the Next Generation) an initiative supported by the United States Agency for International Development (USAID) and led by International Institute of Tropical Agriculture (IITA) (Africa RISING, 2018). The project, known as Smart Agricultural Research Optimization System (SAROS) entailed a solar-powered smart kit for irrigation. Its functionality mainly consisted of automatic water-use monitoring using actual sensor data that is climate-smart.

The above system was studied and discovered to have stalled while still in its initial pilot stage. Among the key observations made that could have contributed to stalling was the use of an Arduino microcontroller in the system. Arduino is very popular in the electronics field for hobbyists and academic prototyping projects. However, there are existing debates that challenge its suitability for scaling industrial projects like automatic irrigation systems on a large scale. Specific constraints that hinder the suitability of Arduino microcontrollers for such applications include processor speed, memory, and the number of available GPIO ports (Kondaveeti *et al.,* 2021; El-Abd, 2017). The project's lack of integration with Wireless Sensor Networks further indicated the complexity of handling and transmitting field data as well as incorporating any analysis of the data.

2.3.1 Leakage Detection

Among the major contributing factors to water loss and wastage during irrigation are pipe leakages. Eventually, leakages could also contribute to water dearth for irrigated plants. An IoT-based system for detecting water pipeline leakage using soil moisture sensors was proposed by Elleuchi *et al.* (2019). The limitation of their proposed system is the reliance on soil moisture percentage to detect a leak. It would detect leakage in other water distribution situations but in a real drip irrigation section, this system would not be realistic as it would take a lot of time to detect the presence of leakage by analyzing soil moisture only.

2.4 Proposed System

This work proposes the development of an IoT-based automated surface-drip irrigation monitoring system prototype. A review of previous works reveals technical gaps in implementation of digital twins' framework for the automation and monitoring of irrigation systems. This system utilizes an XBee-based wireless sensor network for collecting and disseminating soil and weather data to a central coordinator node that also acts as the main irrigation controller. The central coordinator hosts a digital twin that provides virtual interaction between users and the actual physical farm equipment including pump and water valves for remote control. Data visualization is provided via a Node-red web dashboard that is updated in real time. The proposed system conceptual diagram illustrated in Fig. 9 was used.

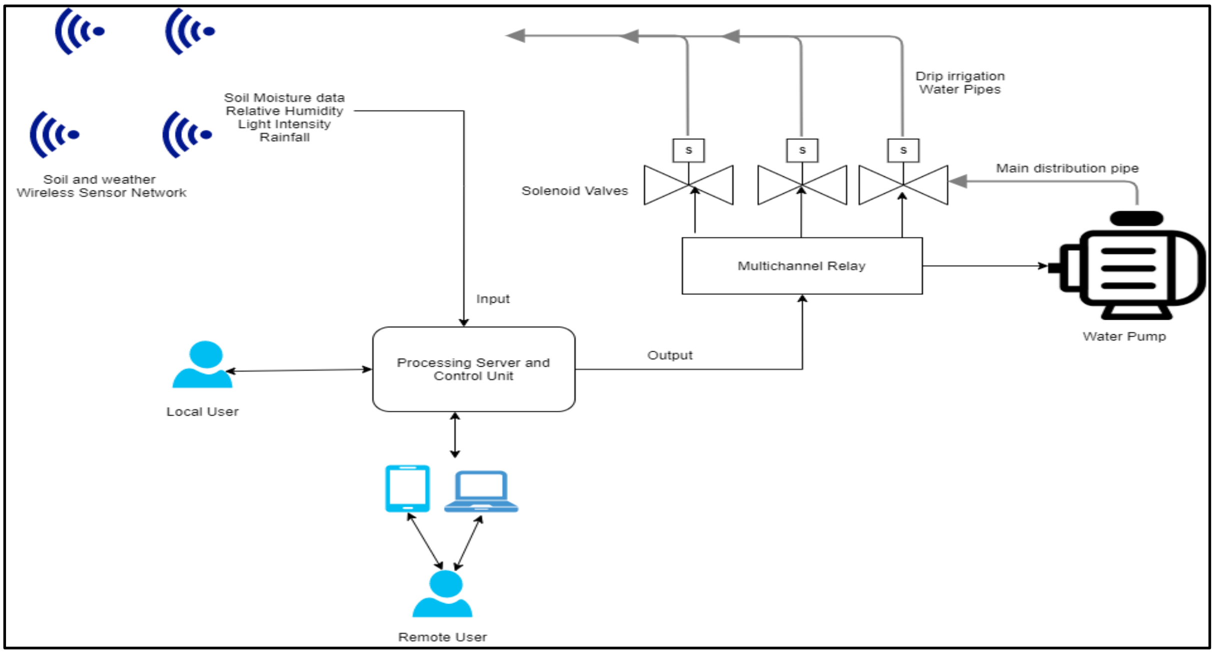


Figure 9: Project conceptual framework

As depicted in Fig. 9, wireless field sensors collect real-time weather and soil parameters like air temperature, humidity, soil temperature, ultraviolet radiation, soil-moisture as well as detecting presence of rainfall. The data is transmitted in packets using XBee radio modules to a central node consisting of a Raspberry Pi integrated with an XBee coordinator node. An analysis is carried out to determine if thresholds have been exceeded. If the real-time data is beyond the predetermined threshold, the pump and valves are automatically actuated while being monitored to ensure the water supplied does not exceed the required amount. Once the maximum threshold soil moisture content is reached, the pump is turned off followed by the individual valves.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Project Case Study

The project was carried out at The World Vegetable Center, located in the Arusha region in Tanzania. The scope entailed the development of a an IoT-based automated Surface-drip Irrigation Monitoring System, integrated with wireless sensor networks for soil data collection and actuator control. An initial online questionnaire was issued targeting 30 staff and subject matter experts from the World Vegetable Center in Arusha. The main objective of the questionnaire was to gather insight regarding user perception of automated irrigation monitoring systems and to collect user requirements.

3.2 Research Methods

Both qualitative and quantitative methods were employed with the Case-Study approach being adopted. These assisted in collection of data regarding Irrigation automation requirements and the challenges of manual irrigation systems. This provided an overview of the vital areas of concern and gathering user requirements.

Primary data for user requirements were collected using face-to-face key person structured interviews, observation, questionnaires, and round table discussions with the Farm Manager, Farm Technicians, and Specialists to gather requirements for developing the proposed system.

A structured questionnaire was used to gather quantitative data from staff at the World Vegetable Center in Arusha. The sample questionnaire is presented in Appendix 1. The key person interview guide on the other hand has been presented in Appendix 2.

A literature survey of existing automatic irrigation systems was carried out to come up with the most suitable approach for the development of the proposed system. Secondary data was obtained from text sources including journal articles from reputable science journals retrieved from ScienceDirect, Google Scholar, reports, websites, and previous dissertations.

3.3 Target Population

The target population in this project were small scale farmers, agricultural experts and users of irrigation systems at World Vegetable Center in Arusha, Tanzania.

3.4 Study Area and Scope

The project study was carried out at the World Vegetable Center, Eastern and Southern Africa, based in Arusha, Tanzania. The scope entailed the development of an IoT-based automated Surface-drip Irrigation Monitoring System.

Benchmarking was carried out with a similar project that was previously carried out at Gallapo village located in Babati District, part of Northern Tanzania, under the Africa RISING an initiative supported by USAID and led by IITA (Africa RISING, 2018). The project, known as SAROS entailed a solar-powered smart kit for irrigation. Its functionality mainly consisted of automatic water-use monitoring using actual sensor data that is climate-smart.

The above system was studied and discovered to have stalled while still in its initial pilot stage. Among the key observations made that could have contributed to stalling was the use of an Arduino microcontroller in the system. Arduino is very popular in the electronics field for hobbyists and academic prototyping projects. However, there are existing debates that challenge its suitability for scaling industrial projects like automatic irrigation systems on a large scale. Specific constraints that hinder the suitability of Arduino microcontrollers for such applications include processor speed, memory, and the number of available GPIO ports (Kondaveeti *et al.,* 2021; El-Abd, 2017). The project's lack of integration with Wireless Sensor Networks further indicated the complexity of handling and transmitting field data as well as incorporating any analysis of the data.

3.4.1 Sampling Technique and Sample Size

The convenience non-probability sampling technique was used when issuing out the online questionnaire to gather user requirements. The selection of this sampling method was influenced by the research constraints in terms of time, budget, sample size and willingness to participate (Lo, 2009). The sample size used was 30 participants, comprising of farmers and employees of World Vegetable Center Arusha.

3.4.2 Requirements Elicitation

This section highlights the methods used to identify requirements for the development of an IoT-based automated surface-drip irrigation monitoring system prototype for the World Vegetable Center in Arusha, Tanzania.

Requirements were gathered by carrying out key person interviews and discussions with four subject matter experts, two representatives from the management team at World Vegetable Center, Group discussions, and issuing a questionnaire targeting 30 staff at the world vegetable center. Observation of daily operations at the World Vegetable Center for two months between March and April also helped in the identification of gaps and challenges to identify requirements for the proposed system.

3.5 Materials Framework

Software and Hardware development tools were adopted in the development of the IoT-based automated surface-drip irrigation monitoring system prototype. This section outlines the materials that were used in this study. They have been described and categorized into software as well as hardware tools.

3.5.1 Software Tools

In this section, the software tools that were used in the study have been described. The function and justification of each tool have been explained.

(i) Node-red

Node-red is an open-source software platform that enables software developers to create meaningful graphical user interfaces for embedded and IoT applications using workflows that incorporate minimal code used by programming visually (O'Leary, 2022). It was selected for this project due to its open-source nature and wide community support globally. Node-red is also light-weight and does not have huge memory requirements hence suitable for deployment in this project's main control unit consisting of a Raspberry Pi 4. This is in contrast to other alternatives in the market such as Qt software which is built on C++ language, requires heavy customization, and is available on a subscription basis, especially the development framework. Version v2.2.2 was specifically used in this project, which is the latest version as of the implementation time.

(ii) The STM32Cube Programmer

The STM32Cube programmer was selected for flashing the bootloader and programming the STM32F103C8T6 (*Blue-pill*) microcontroller board that was deployed on the sensor nodes. This programmer is a powerful tool from ST Microelectronics that is versatile due to its support for multiple communication protocols such as Universal Asynchronous Receiver/Transmitter (UART), Joint Test Action Group (JTAG), Serial Peripheral Interface (SPI), Controller Area Network (CAN), and Serial Wire Debug (SWD). Moreover, it is multiplatform and hence is supported by all operating systems platforms (STMicroelectronics, 2022b). A bootloader is a low-level software application that is responsible for managing the process of updating the flash memory of a microcontroller with new application data or code (STMicroelectronics, 2022a).

(iii) Raspberry Pi OS

Raspberry Pi OS is an open-source operating system designed for the Raspberry Pi mini-computer, enabling users to program and control peripherals on the Raspberry Pi. The specific version of the OS used for this project was the buster version as it is the version customized for running Node-red applications. It also features an updated graphics library (Long, 2019). At the time of conducting this study, the buster version of the Raspberry Pi OS was the only version customized to support Node-Red. In all the other versions, Node-red would crash at run-time.

(iv) Python Programming Language

Python was selected for programming the sink node which consists of a Raspberry Pi that receives sensor data and activates the actuation of the water pump and valves. Python is easy to use and is pre-installed in all Raspberry pi operating systems.

(v) KiCad

Circuit simulations have become inevitable with the increase in the complexity of chips and electronic design capabilities. The KiCad is a recent free electronic design automation software. It is feature-rich with circuit schematic design capabilities, printed circuit board layout as well as 3-Dimensional visualization and simulation. Having been built on an open-source framework, KiCad was the best candidate for this project due to large community support, a no-cost model, and an easy learning curve (Evans, 2021).

A comparison of KiCad and other Computer-aided design tools (CAD) has been tabulated in Table 1. As demonstrated, KiCad outdoes the other tools in terms of a very huge open-source community making learning easy and free.

Table 1: Comparison between KiCad and other PCB design tools

	KiCad	Proteus	Altium	EasyEDA	Fusion360
Cost	Free and open-source	Paid	Paid	Both Free and Subscription plans	Free for 3 years for qualifying non-commercial use, otherwise paid
OS Platform	Windows, Linux, macOS	Windows	Windows	Windows, Linux, macOS	Windows, macOS
Support	Huge community support	Minimal support	Minimal support (requires paid training)	Has community support	Limited community support
3-D features	Full 3-D enabled	Full 3-D enabled	Full 3-D enabled	Only allows mechanical 3-D of the board, not components	Full 3-D features enabled

(vi) Fritzing

Fritzing is a first-level circuit simulation software tool. It was selected for the initial prototyping of the sensor node and sink node circuits as would be visualized on a breadboard before embarking on the final printed circuit board design. It facilitates the process of sharing rapid prototypes of initial designs of electronic circuits, and minimizes design faults before final production thus eliminating the risks of incurring unnecessary post-production costs.

(vii) The XCTU Software

Digi is the company responsible for the roll-out of XBee modules. XCTU software is a tool that they provide for configuring the XBee radio modules (Digi International Inc., 2020).

(viii) Pi-Tunnel

The Pi-Tunnel is a cloud-based solution that provides hosting services for applications that run on edge devices, specifically Raspberry Pi. Through this platform, IoT applications running on the raspberry pi can be accessed remotely via custom web domains, at the same time allowing for remote monitoring of the raspberry pi as well as remote firmware upgrades when required.

3.5.2 Hardware Tools

In this section, the specific hardware tools deployed in this study have been outlined.

(i) The STM32F103C8T6 Microcontroller Board

The STM32F103C8T6 is a low-cost 32-bit microcontroller based on an ARM Cortex-M3 processor. It was selected for the sensor nodes in this project due to its high processor speed (72 MHz), and the adequate number of usable GPIO pins, as presented in Fig. 10, which extends to 37 pins, 20 KB static RAM as well as 128 Kb or 64 KB of flash memory. In terms of communication interfaces, this board has 3 UARTs, 2 I2Cs, 2 SPIs, and a micro-USB port (STMicroelectronics, 2021).

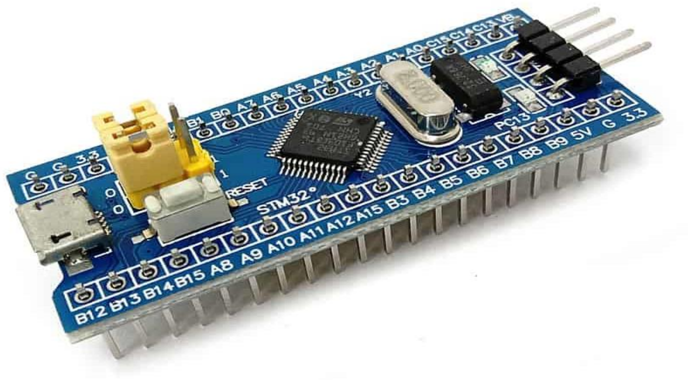


Figure 10: The STM32F103C8T6 Microcontroller Board

The board specifications are detailed in Table 2.

Table 2: The STM32F103C8T6 Technical Specifications

S/N	Feature	Specification
1.	CPU	ARM 32-bit Cortex M3
2.	Max CPU Frequency	72 MHz
3.	Memory	64/128 Kb flash mem, 20 Kb SRAM
4.	Power Supply Management	2.0-3.6 V
5.	DMA	7-channel DMA controller
6.	I/O Ports	Up to 80
7.	Debug Mode	Serial Wire Debug (SWD) and JTAG interfaces
8.	Timers	7
9.	Communication interfaces	2 -I2C, 3 USARTs, 2 SPIs, CAN, USB 2.0

(ii) Raspberry Pi 4

Raspberry Pi 4 Model B single board computer presented in Fig. 11 was selected for the coordinator node, mainly due to its unique features as highlighted in Table 3.

The raspberry pi is an open-source single board computer that is popular in many edge-computing applications. Ongoing research has pointed to its ability to handle real-time data more conveniently offering faster response time. It gets rid of needless data brokers, ensuring the data processing is as close as possible to the source and to the user (Debauche *et al.*, 2022).

Table 3: Raspberry pi 4 model B technical specifications

Feature	Specification
RAM	4 GB
Processor	64-Bit quad-core ARM Cortex-A72
Communication	Dual-band Wi-Fi, Bluetooth, Ethernet, UART, SPI, I2C, USB
Display	Dual 4K HDMI output interfaces, 2-lane MIPI display interface
Other peripherals	Camera CSI interface
Power	5 V DC via USB type-c and GPIO pins (Minimum current 3A)
Processor Speed	1.5 GHz
GPIO	Standard header 40-pin with backward compatibility

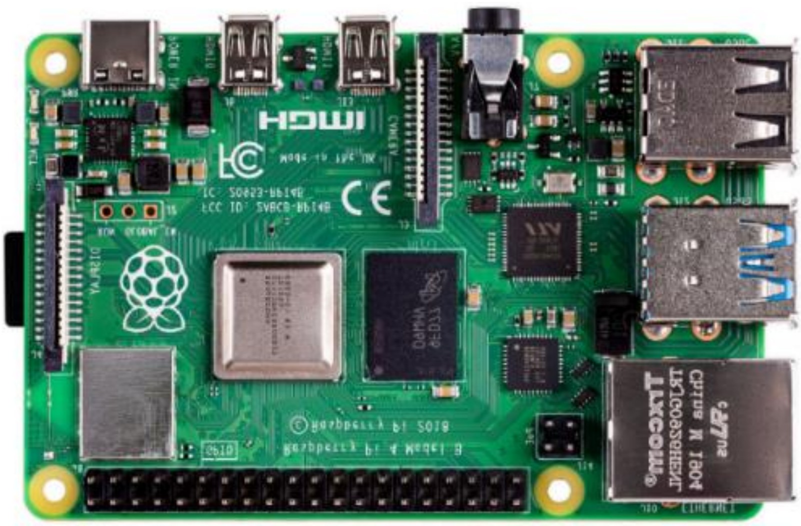


Figure 11: Raspberry Pi 4 model B

(iii) The ICStation Resistive Soil moisture sensor

This soil moisture sensor presented in Fig. 12 was selected because of its ease of reading, ability to adjust threshold, power, and signal indicator LEDs for easier operation, and durable probes that are resistant to corrosion. It can utilize input voltage between 3 V and 12 V, making it ideal for this sort of application (ICStation, 2020).

Figure 12: The ICStation Soil Moisture sensor

(iv) Ambient DHT22 wired temperature and humidity sensor (AM2303)

A wired version of the DHT22 digital temperature and humidity sensor presented in Fig. 13 was selected due to its ability to withstand hostile environments since it has a last plastic enclosure for protection and the fact that it is wired which makes its deployment easier, without having to worry about space, as only the signal wires are required on the circuit board. Its technical details include low power between 3 and 5 V, a sensor range of between 0 and 100% for humidity with an accuracy ranging between 2 and 5%, while the temperature sensing range is between -40 and 80℃ with an accuracy of + or – 0.5℃. It has a sampling rate of 0.5 Hz and has 3 connectors for Power, ground, and signal. It is also relatively simple to program this sensor (Lady, 2021).

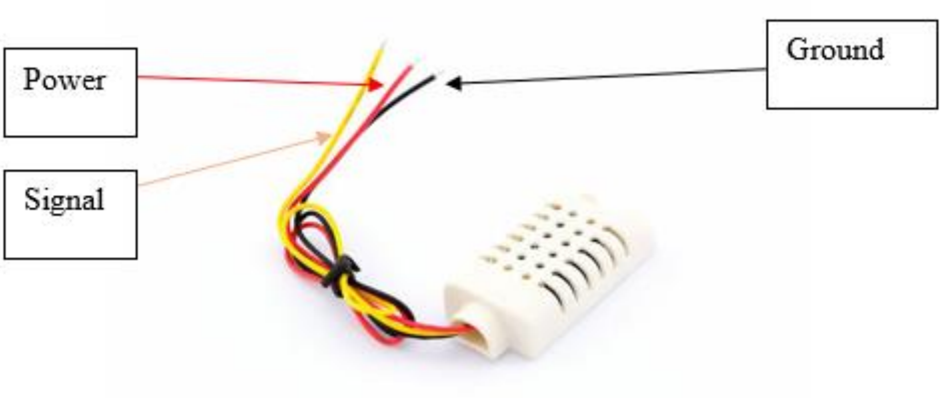


Figure 13: The AM2302 wired digital temperature humidity sensor

(v) The XBee Pro S2C Radios

The XBee Pro S2C radios presented in Fig. 14 communicate using the IEEE 802.15.4 ZigBee protocol. This is a low-power wireless communication protocol that is preferred for meshed IoT network applications since it supports sleep mode operation hence saving power and also due to the ability to interconnect different devices and networks to create a mesh. These radios have a range of up to 2 km outdoors, due to the powerful antenna with 63 mW with a maximum data rate of 250 kbps (Digi International Inc., 2020).

Figure 14: The XBee S2C module (left) and Funduino XBee adapter (right)

(vi) Water-Pump

A 12 Volt (V) DC submersible electric bilge pump with a flow rate of 750 Gallons per hour (GPH) was used in this project. It was selected due to its low cost (less than USD 20), high-performance motors, and safe operational procedures (SEAFLO, 2021). The submersible bilge water pump, as presented in Fig. 15 is manufactured by SEAFLO. The pump has two power terminals, positive and negative, and incorporates a float switch as well as a normal switch.

Hence, its operation mode is easy to set up and configure to control using any DC-controlled relay module.

Figure 15: Submersible bilge pump

(vii) Industrial-Grade Solid-State Relay

The SSR-40 DA which is a solid-state relay presented in Fig. 16 was selected for switching on the water pump. Among its key features are the fact that it is Single-phase but also has an in-built opto-couple, which means it can be interfaced directly to a microcontroller without needing any additional connection circuit. It is more durable due to the enclosure as well as the high resistance of the insulation, which exceeds 50 Mega-ohms. It enables current is also minimal, below 7.5 mA. It has a high Load current (40 A) and Load voltage (28-380 VAC). The control voltage is also low ranging from 3-32 VDC, making it compatible with Raspberry Pi GPIOs whose voltage level is normally 3.3 V. It can be switched at a faster speed compared to other mechanical relays as it has no mechanical movable contacts (Fotek Controls LTD, 2022).

Figure 16: Industrial SSR-40 DA solid-state relay

(viii) Industrial Motorized Ball Valve

The motorized ball brass valve illustrated in Fig. 17 was selected for this project, with a half-inch valve size DN15. Its key performance features include the ability to open and close in under 8 seconds. In addition, it can remain in whichever state without further consuming energy as opposed to most solenoid valves which require to be continuously energized while in a changed position. It is powered by a DC voltage of 12 V and its working current is less than 80 mA, hence power consumption is less than 2 W. It has five wires, two to supply power for rotating opposite directions, one for ground and two for feedback to indicate fully open or fully closed positions (U.S. Solid, 2022).

Figure 17: Motorized ball valve

(ix) Liquid-Flow sensor

Liquid Flow Sensor model YF-S201 ½” presented in Fig. 18 was used. The main function of the water flow sensor is to allow computation of the total water consumption via irrigation. The YF-S201 has a minimum operational voltage of 3.5 V DC and a maximum voltage of 24 VDC. The maximum operating current is 15 mA and the maximum liquid pressure allowed Is 1.75 Mpa.

Figure 18: The YF-S201 Water Flow Sensor

3.6 System Development Methodology

As per the second specific objective, to develop an IoT-based automated surface-drip irrigation monitoring system prototype, a hybrid approach was selected for system development. The System prototype was developed using agile system development methodology due to its flexibility in accommodating changes, its emphasis on high-quality software, minimal risk profile, concurrency of development with testing, and its suitability in situations where requirements are dynamic, and not fully mature. Agile also emphasizes delivering both value and quality for the customer as well as project completion within the provided constraints such as schedule (McLellan *et al.,* 2021). Furthermore, Agile enhances the satisfaction of the customers as well as stakeholders (Douglass, 2013). Changes initiated by the customer are well embraced (Kaisti *et al.,* 2014). The specific agile framework adopted was Extreme Programming. However, waterfall model concepts were also incorporated during the Hardware Subsystem development to minimize the likelihood of running over the budget.

3.7 System Design

The system was designed by interpreting the gathered requirements into a textual format and further into a diagrammatic system format. A modeling language, Unified Modeling Language (UML), and SysML were used to model the requirements to come up with a Use Case diagram, Flow charts, activity diagram as well as a state diagram. The software tools used for the modeling included Draw.io.

3.7.1 Use Case Diagram

A use case diagram presented in Fig. 19 was designed to model the interaction between the users and the system. The system's main actors were identified as the Admin, Irrigation technician as well as the control unit of the system.

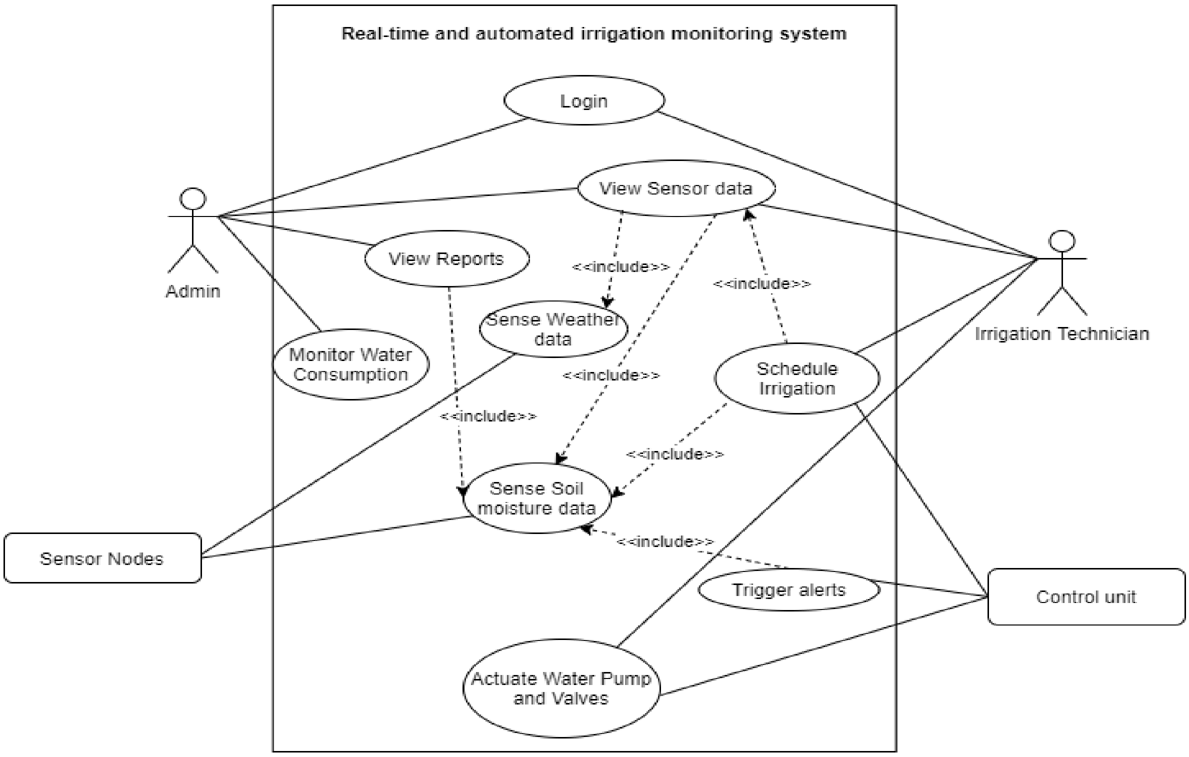


Figure 19: System use case diagram

The sensor node sub-system senses soil moisture data, senses weather data and relays to the control unit which is the coordinator node. The control unit triggers alerts, notifications and events which include actuation of water pump and valves. An irrigation technician can view sensor data, schedule irrigation, and trigger the actuation of pumps and valves.

3.7.2 System Block Diagram

A block diagram of the entire system was modeled as represented in Fig. 20 for the proposed IoT-based automated irrigation monitoring system. The System entails a processing and control unit for receiving sensor data from the sensing units and controlling the actuation unit. The Actuation unit is responsible for the conversion of the 3.3 V from the control unit to the equivalent AC and DC voltage responsible for driving the switching on and off the water pump and the solenoid valves respectively.

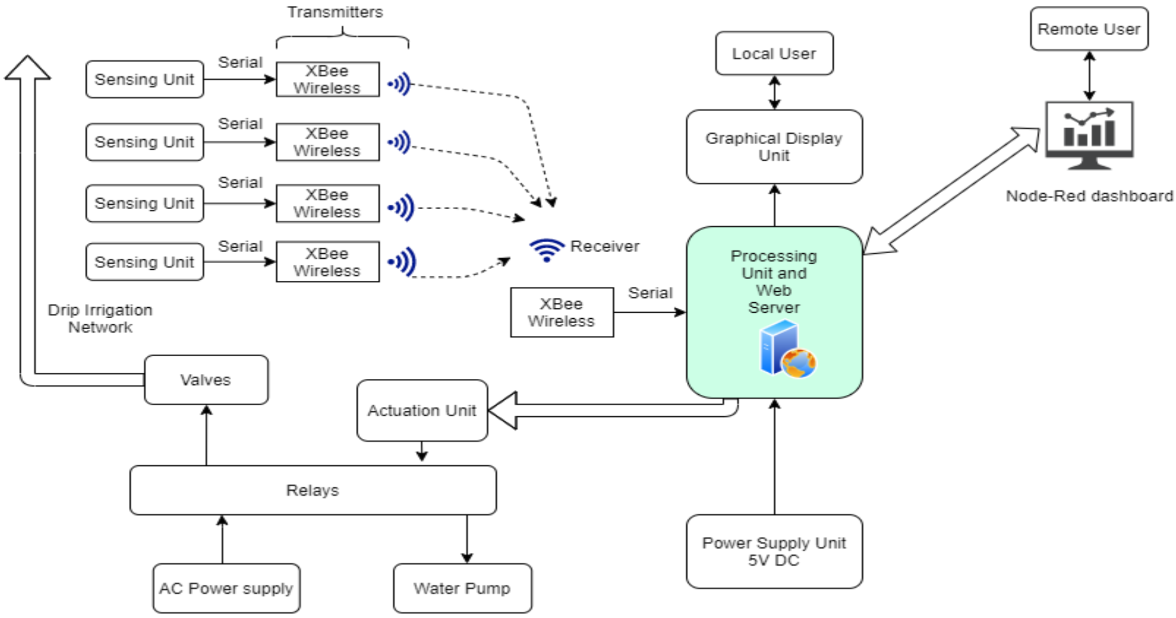


Figure 20: Block diagram of the system prototype

Additionally, a block diagram of the sensor node subsystem presented in Fig. 21 was modeled. The main components of the sensor node are the STM32F103C8T6 microcontroller, air temperature-humidity sensor, UV-light intensity sensor, soil-temperature sensor, XBee S2C radio module, LCD, solar module, solar charge controller, and Lithium-Polymer battery.

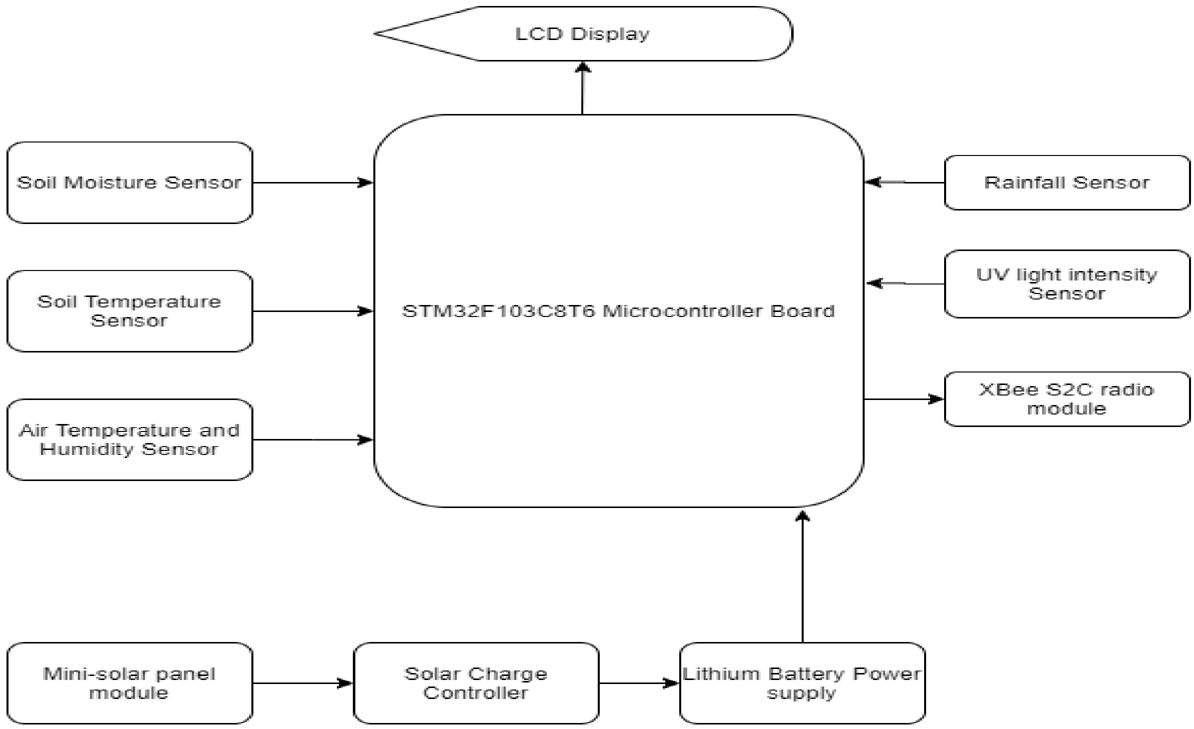


Figure 21: Block diagram of the sensor node hardware subsystem

Each sensor node was interfaced with a liquid crystal display, soil moisture sensor, soil temperature sensor, ambient temperature, and humidity sensor, XBee S2C radio module, UV light intensity sensor, rainfall sensor, 5 V mini-solar panel as well as lithium polymer batteries.

On the other hand, the main control unit (sink node) as presented in Fig. 22 consisted of a Raspberry Pi 4 micro-computer interfaced with a 7" Touch-screen display, multi-channel relay unit, XBee S2C coordinator module, GSM module, water flow sensor, and a switch mode power supply unit.

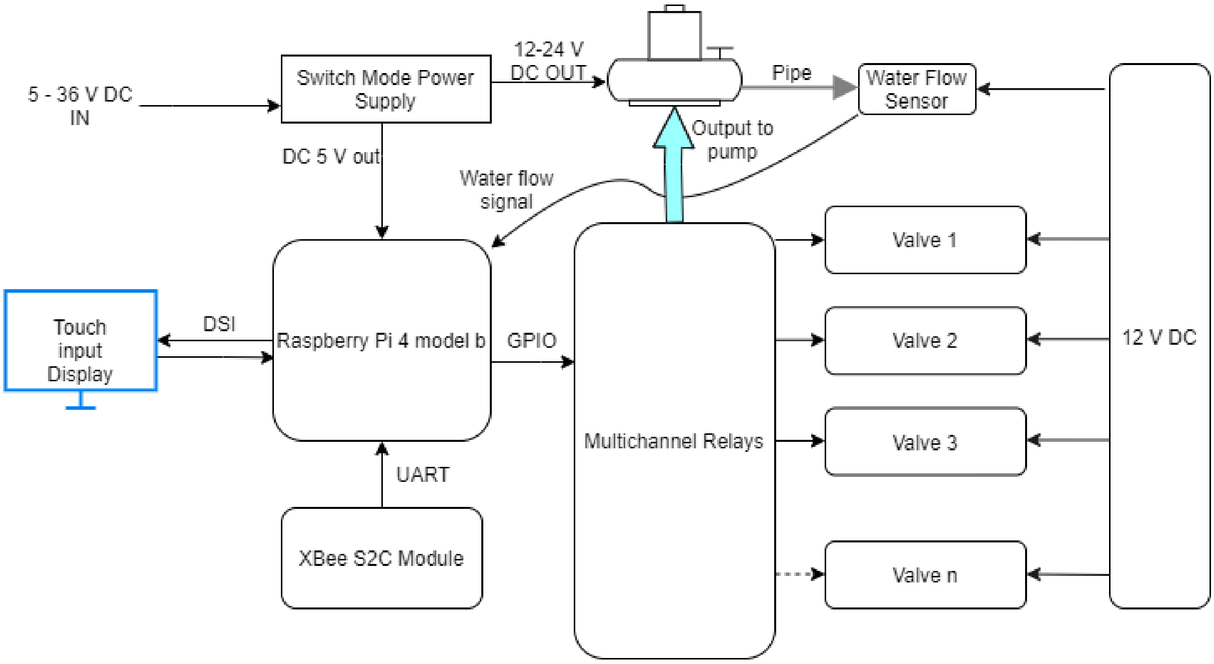


Figure 22: Coordinator control unit block diagram

3.7.3 System Architecture Diagram

To visualize the interconnection between the different subsystems, a system architecture diagram presented in Fig. 23 was modeled. A central coordinator node consisting of a Raspberry Pi 4 integrated with an XBee module configured as a coordinator receives real-time data from soil and environmental sensors. Based on the received soil moisture levels, the coordinator node can control the pump automatically and control the opening and closure of the motorized ball valves, through the activation of relays. The sensor data from all nodes are transmitted to Node-red for further analysis or future retrieval. Additionally, Node-Red was selected for creating a graphical user interface (O'Leary, 2022) that users can interact with to access sensor data and control irrigation remotely if need be.

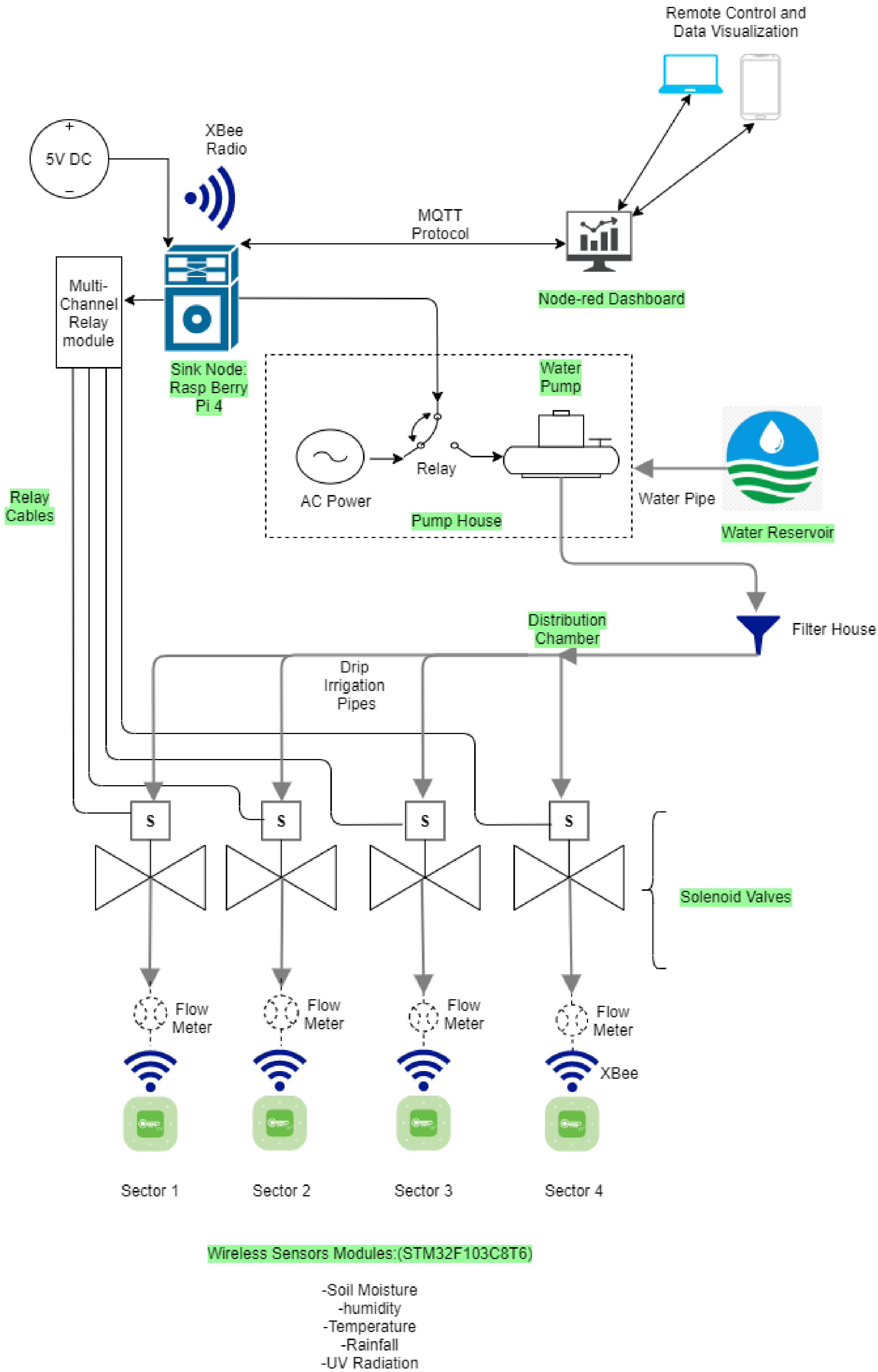


Figure 23: Architecture design of the proposed system

3.7.4 Sensor Node Circuit Design

The sensor node was first designed using fritzing as shown in Fig. 24 to visualize the GPIO pin selection on the STM32F103C8T6 microcontroller board as well as the component selection and how the different sensors would be interfaced.

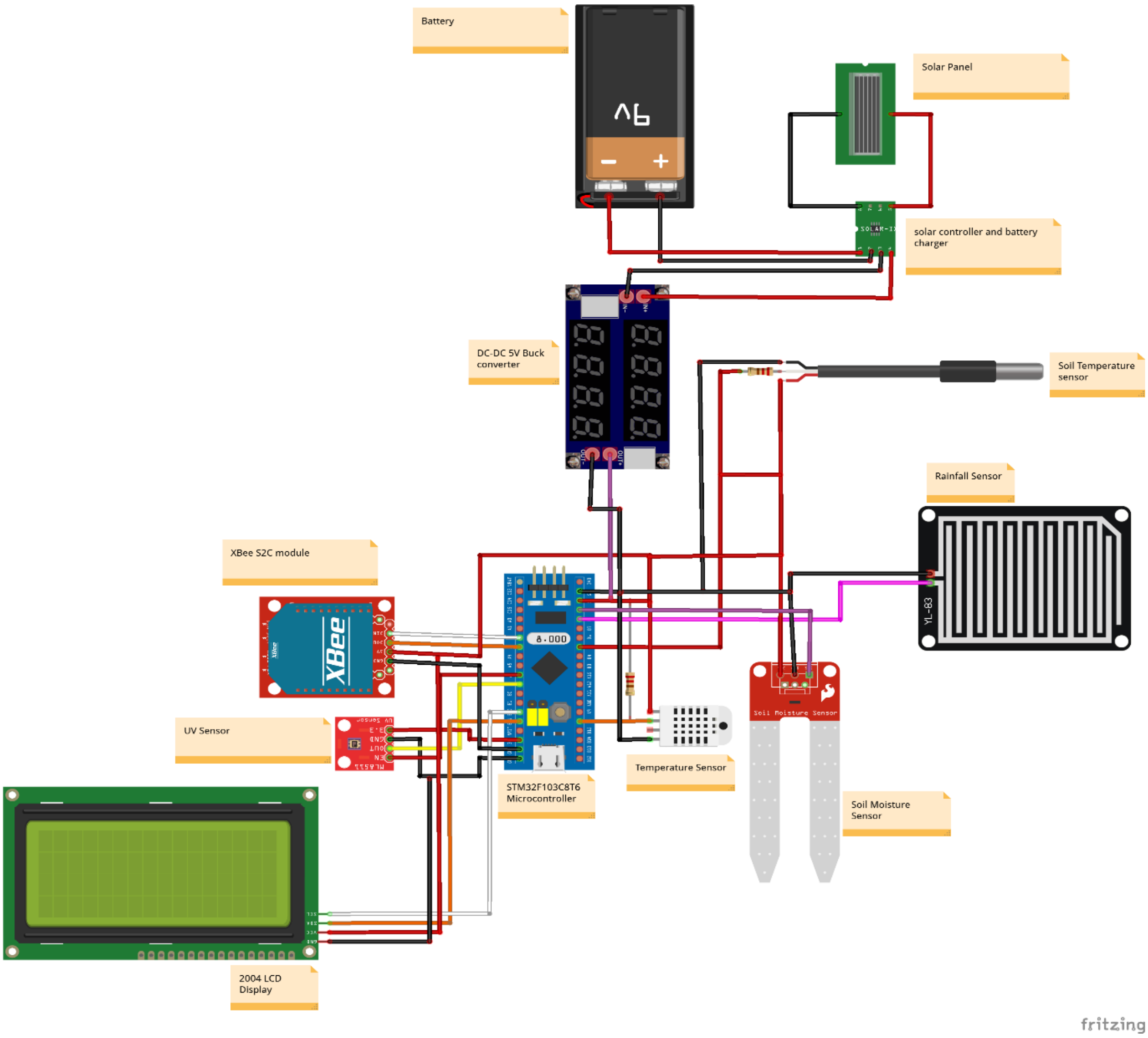


Figure 24: Sensor node circuit design

3.7.5 Coordinator Node Circuit Design

The coordinator node was first designed from fritzing as shown in Fig. 25 to assign GPIO pin distribution and assess layout requirements. XBee coordinator module was connected to the Raspberry Pi 4 via the UART Serial interface. The multichannel relay was powered via the 5 V power pin to control the pump and two valves. Waterflow sensors were connected to the GPIOs to monitor the flow rate via each valve-controlled channel.

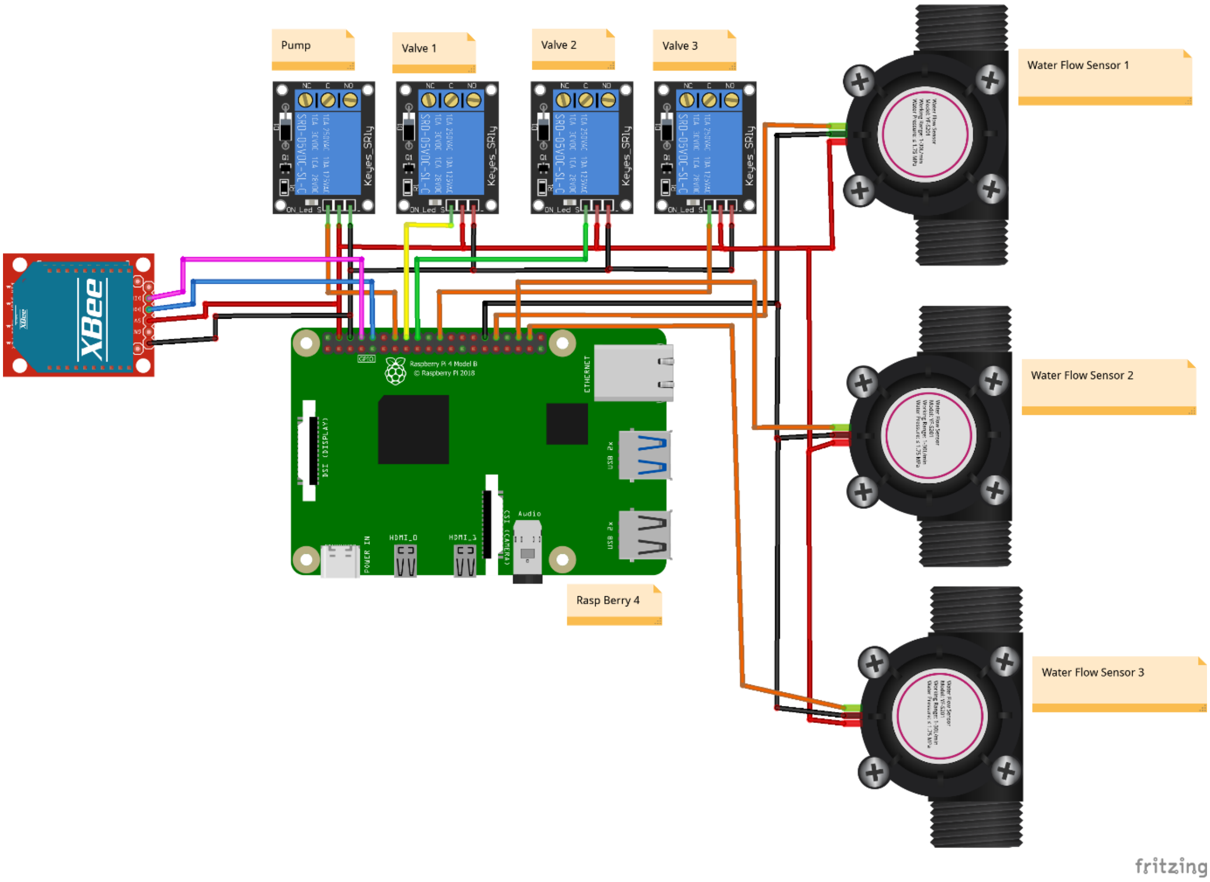


Figure 25: Coordinator node design

3.7.6 Printed Circuit Board Diagrams

The final circuit design was carried out using KiCad. The KiCad is a free open-source software solution for computer-aided design. It stands out as a favorite tool for hardware engineers among the existing tools because despite being free, it is simple to use even for the most complex circuits (Antonella Del, 2015).

As an Electronic Design Automation (EDA) software suite, KiCad has become primarily the most sought-after design tool within the open-source community. It combines schematic capture, PCB layout, creation of bill of materials (BOM) Gerber files for final fabrication as well as PCB 3D views. All these features streamline the whole process of creation and verification of designs in a digital format before fabrication (Abhimayu, 2018). The illustration in Fig. 26 represents the general highlight of the schematic's mode on KiCad.

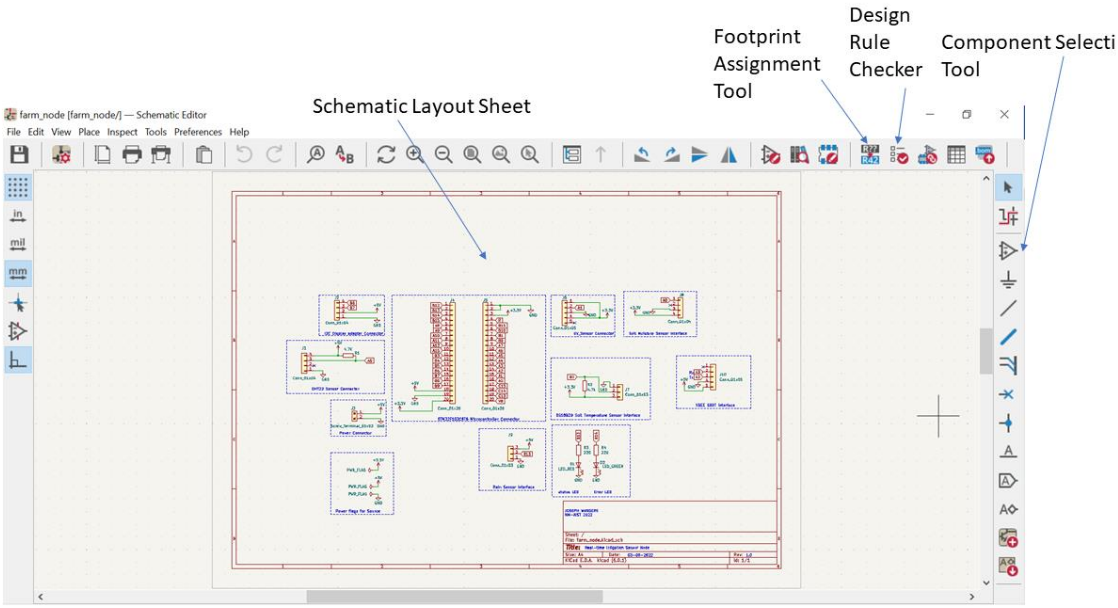


Figure 26: Schematic view on KiCad

(i) Sensor Node Schematic Diagram

The sensor node schematics as captured in KiCad presented in Fig. 27 consists of 10 mini-sections. These are the 5 V power input, Air-temperature sensor interface, rain-sensor interface, XBee UART connector interface, I2C-Display connector, Soil-moisture sensor connector, Soil temperature sensor connector, Microcontroller connectors as well as UV-sensor connector.

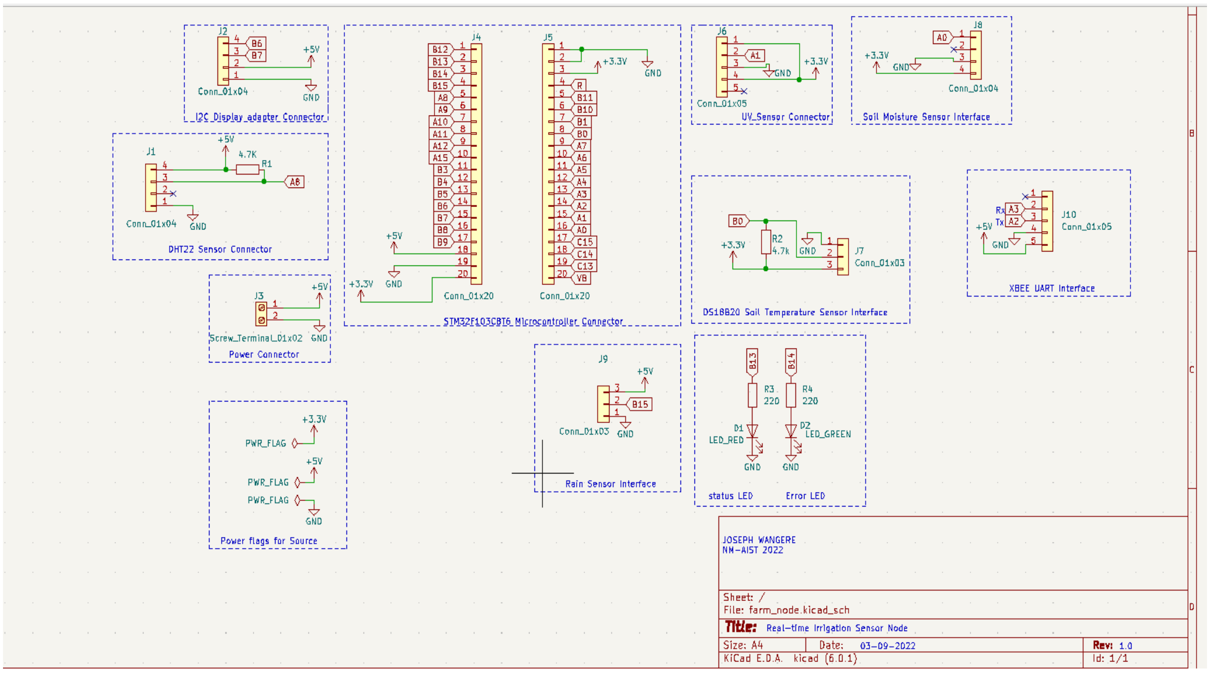


Figure 27: Sensor node circuit schematic diagram

(ii) Sensor Node PCB Layout

After the schematics capture process, the circuit components were associated with their respective footprints and a netlist was generated. The netlist was further imported into the into the PCB mode within KiCad, as shown in Fig. 28 in order to place components on the board with their footprint models and to draw the required signal as well as power traces.

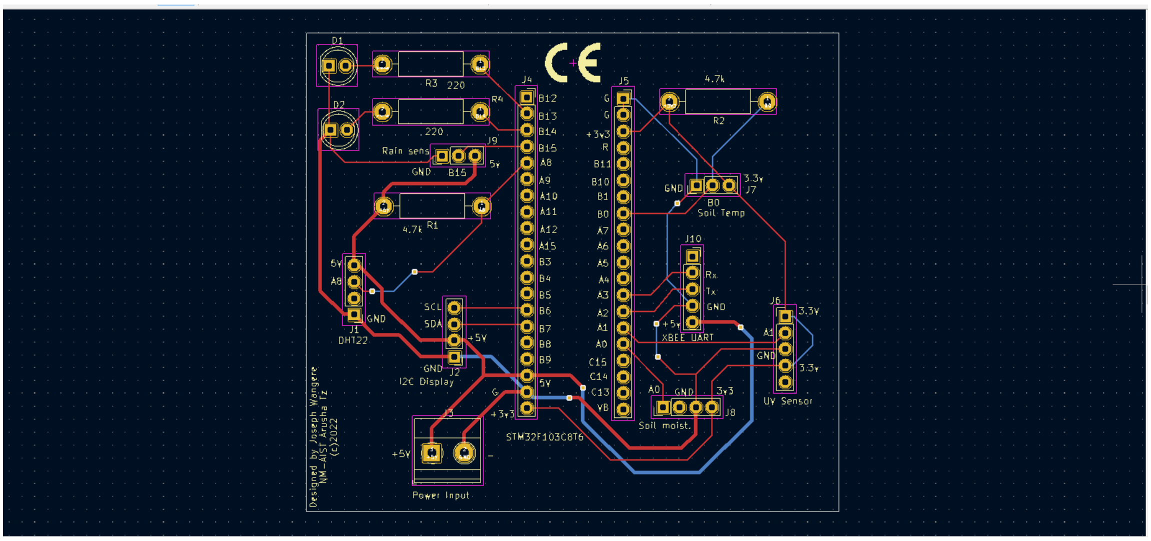


Figure 28: Sensor node PCB layout

(iii) Sensor Node 3-D Visualization

To visualize the actual layout of the circuit board post-production, the designed model was simulated as shown in Fig. 29 using KiCad's powerful inbuilt 3-D simulator. However, customization had to be made to accommodate the STM32F103C8T6 microcontroller board on the final 3-D simulation.

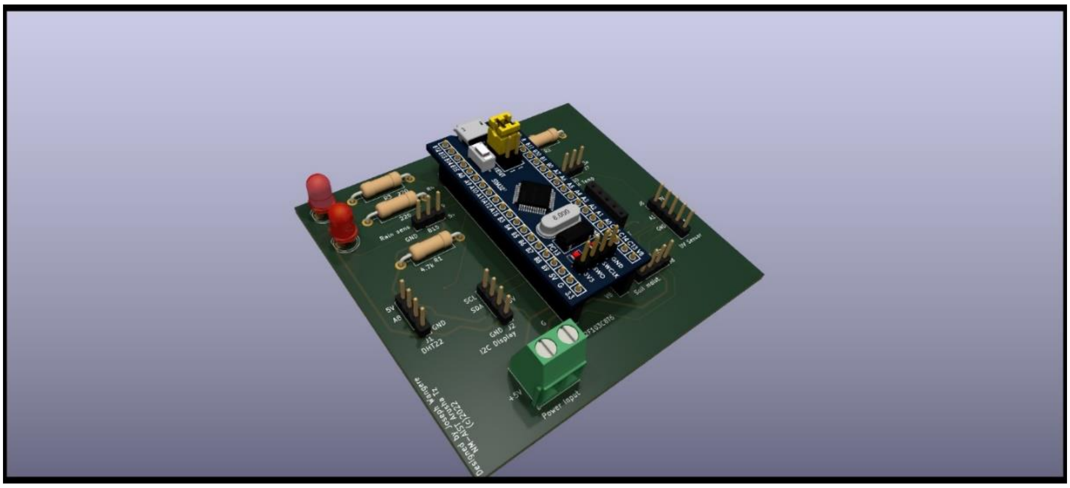


Figure 29: Sensor node PCB 3-D visualization

3.8 System Validation

Validation was carried out by running test cases to align the developed system prototype with the defined user requirements. Testing was carried out at each development phase while incorporating required changes arising from the tests.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results from Questionnaire

The first specific objective of this study was to identify the functional and non-functional requirements for the development of an IoT-based automated surface-drip irrigation monitoring system prototype. In order to identify and gather the requirements, qualitative and quantitative methods were applied. For qualitative methods, case studies were reviewed, focus groups with the different farm attendants as well as key person interviews. An online questionnaire was issued to stakeholders, targeting the 30 members of staff at the World Vegetable Center. Out of the targeted 30 members of staff, 9 responded to the online questionnaire as presented in the following section.

(i) Staff whose work is directly or indirectly affected by the irrigation system

As shown in Fig. 30, 77.8% of respondents felt that their work was directly affected by irrigation system.

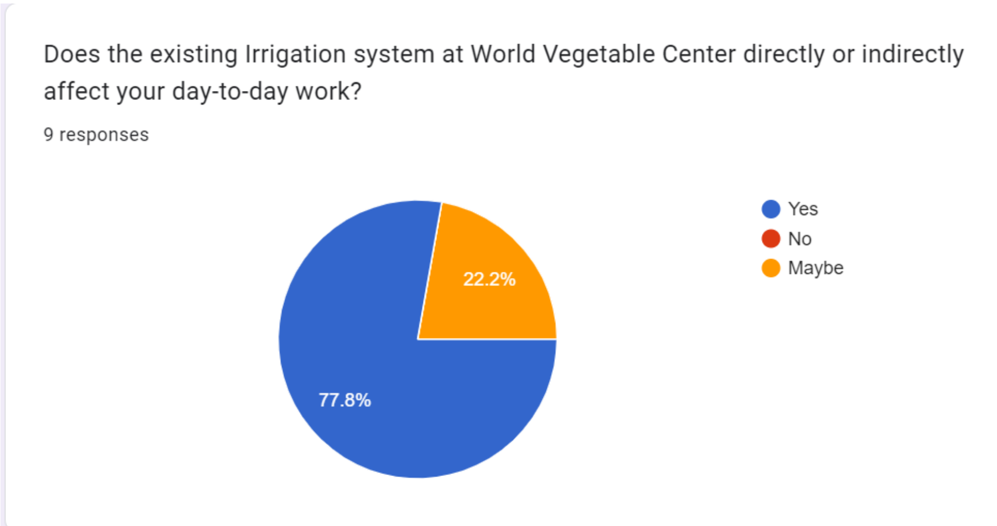


Figure 30: Staff that are affected by the irrigation system

(ii) User awareness regarding smart irrigation systems

Out of the total number of respondents, 88.9% expressed being aware of smart irrigation systems as presented in Fig. 31 while the rest 11.1% did not have any prior knowledge regarding smart irrigation systems.

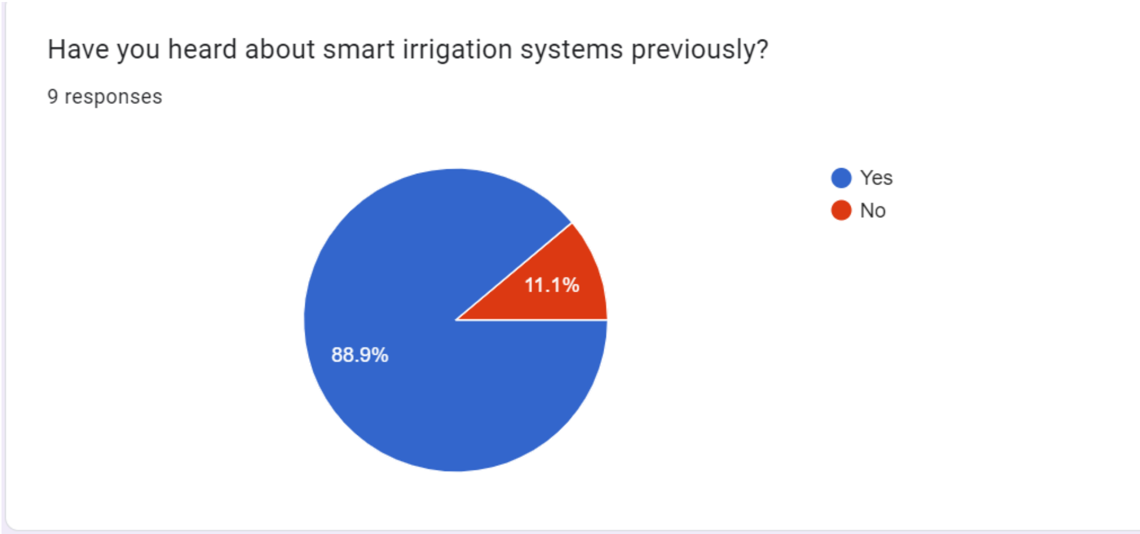


Figure 31: User awareness of smart irrigation systems

(iii) General user satisfaction with current manual system

As presented in Fig. 32 he general level of satisfaction with the existing manual system was tested.

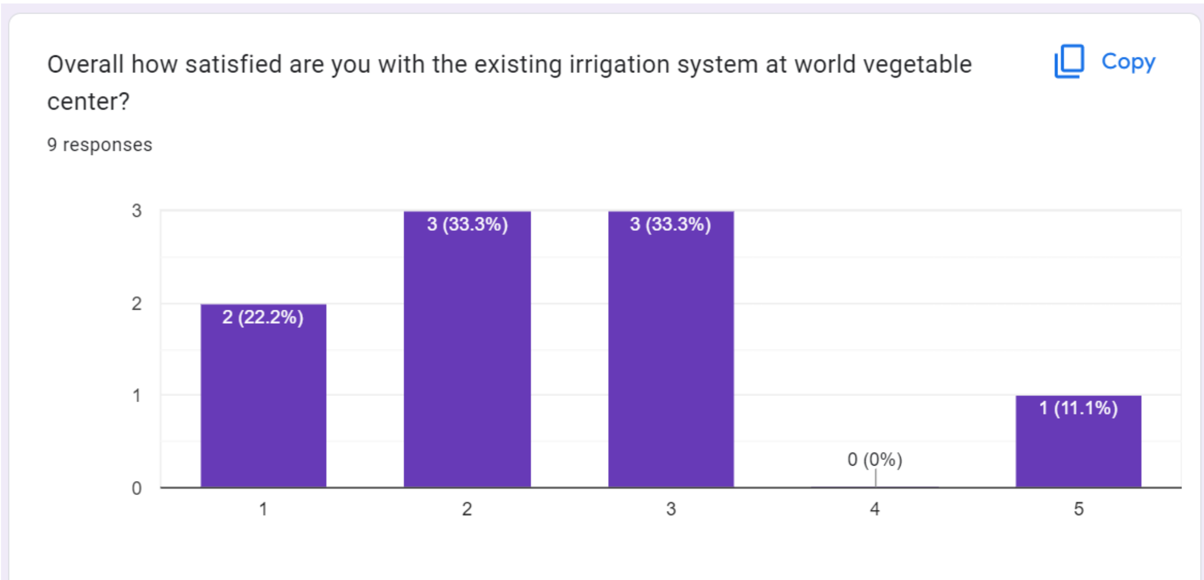


Figure 32: General level of satisfaction with existing manual irrigation system

A scale of 1 to 5 was used, where 1 presented least satisfied while 5 presented very satisfied. More than half of the total respondents expressed dissatisfaction in the existing manual irrigation system. This accounted for 55.5%. Another 33.3% were neither satisfied nor dissatisfied while the last category expressed being very satisfied.

(iv) Reasons behind dissatisfaction with the existing manual system

Participants were asked to provide reasons as to why they were dissatisfied in the existing manual irrigation system. Respondent 1 had this to say: "*The Current irrigation system is wasteful.*" They went on to add, *"Irrigation is not done based on crop needs (for example, based on the moisture content of the soil or water requirement of the crop) and the Irrigation system is manual requiring human presence for nearly every operation."*

Another respondent had this to say, "*A lot of water is lost, and water shortage in reaching some plots hinders crop growth and performance."*

Another respondent expressed similar concerns regarding the system requiring human intervention. They had this to say, *"… the existing system depends on human judgment on when and how long to irrigate which sometimes can be incorrect."*

(v) Need for an automated irrigation system

The respondents expressed a need to have an automated irrigation system to deal with their expressed concerns regarding a manual system. As presented in Fig. 33, all respondents agreed that an automated irrigation monitoring system is necessary.

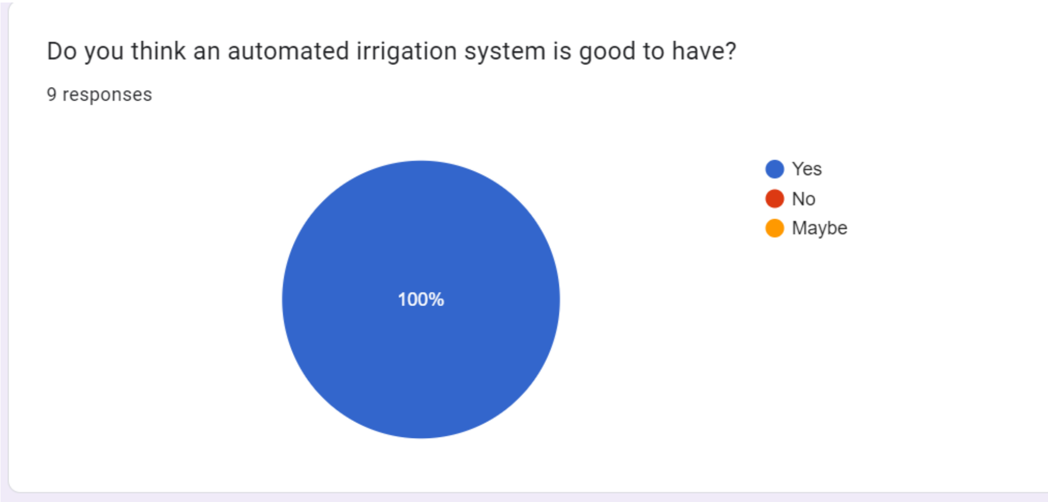


Figure 33: Need for an automated irrigation monitoring system

When asked why an automated irrigation monitoring system was necessary, the respondents had the following to say:

Respondent 1 – "*This will help to be precise on the time and amount of water to be irrigated and will reduce the cost of pumping unnecessary water to the field.*"

Respondent 2 – "*It will help us economize the use of water and it will help reduce our operational cost*"

(vi) Key specific features of the system

The final part of the questionnaire was centered around features that the participants felt were key for the system to have. The most common feature among the respondents was: *"PC or mobile interface, access to adjustments based on our needs for specific activities involving irrigation of our crops."*

4.1.1 Demographic Characteristics

The demographic composition of the respondents has been presented in this section.

(i) Gender

As shown in Fig. 34, out of all the respondents, 55.6% were male while 44.4% were female.

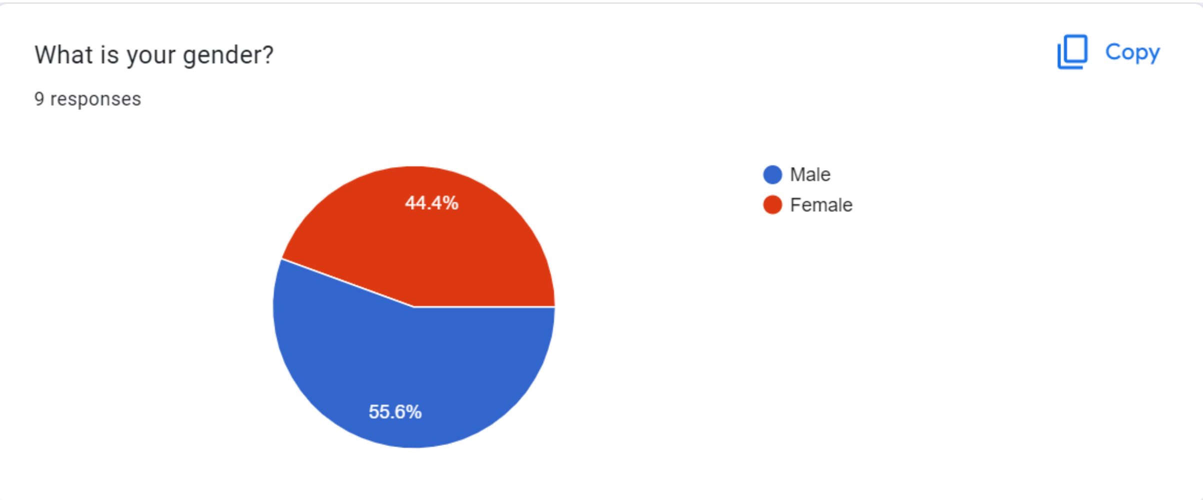


Figure 34: Gender composition of respondents

(ii) Age composition

Figure 35 presents the age composition of the respondents. The highest percentage which was 56.6% was composed of youth between 18 and 28 years.

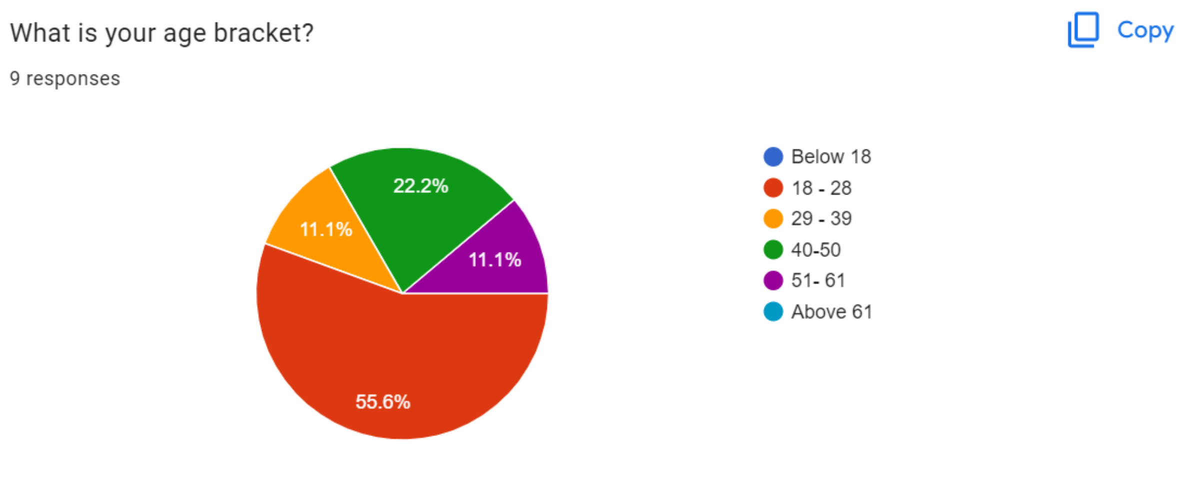


Figure 35: Age composition of the respondents

(iii) Specialization

The respondents were drawn from different specializations within the agricultural domain. These were composed of general administration, crop protection experts, seed science and conservation, crop production, agricultural economist, farm management, breeding, post-harvest as well as field-trial managers.

4.2 Identified Requirements

This section highlights the identified requirements as gathered from the users. The requirements were categorized into functional and non-functional requirements. The functional requirements as captured from the data collected from participants have been tabulated in Table 4. The requirements are a compilation of the results from the questionnaire, key-person, interviews, field observation as well as focus group discussions.

Table 4: Functional requirements

S/N	Requirement
1.	The system shall enable remote control of the pump and valves from a PC or mobile device by the irrigation technician.
2.	The system shall factor in time
3.	The system shall monitor soil moisture and soil temperature
4.	The system shall monitor weather parameters mainly: Air temperature and humidity, ultraviolet radiation
5.	The system shall automatically switch the pump and valves on and off based on soil moisture content
6.	The system shall present real-time data to users, both admin and irrigation technician.
7.	The system shall send notification alerts to both the admin and irrigation technicians.
8.	The system shall have a user interface to monitor and control the system

In addition to the functional requirements, non-functional requirements were captured as presented in Table 5.

Table 5: Non-functional requirements

S/N	Requirement
1.	The system shall be accessible always
2.	The system shall be reliable
3.	The system shall be safe to operate The system shall be secure
4.	The system shall be maintainable
5.	The system shall be user-friendly

The functional and non-functional requirements are translated into specific hardware and software tools to address them. A summary of the hardware and software tools has been outlined in Appendix 3 and Appendix 4 respectively.

4.3 System Development Results

This section presents results from the development of the IoT-based automated surface drip irrigation monitoring system. It has been categorized into the hardware submodule and software as well as firmware submodule.

4.3.1 Software and Firmware Subsystem

The developed software subsystem consists of a web dashboard which was hosted on a Raspberry Pi 4 microcomputer hardware platform. The Raspberry Pi 4 was used as the web server receiving HTTP requests locally and remotely. In order to access the web application locally, a user requires to be connected to the same network as the Raspberry Pi 4. However, for remote access to the web application, a tool called PiTunnel was used. PiTunnel allows remote access to Raspberry Pi projects from any network as well as remote access to network services running in the Raspberry Pi from any network (PiTunnel, 2021).

Upon loading the URL, a login page is displayed as shown in Fig. 36 based on the privilege level. There is an admin level and normal user level.

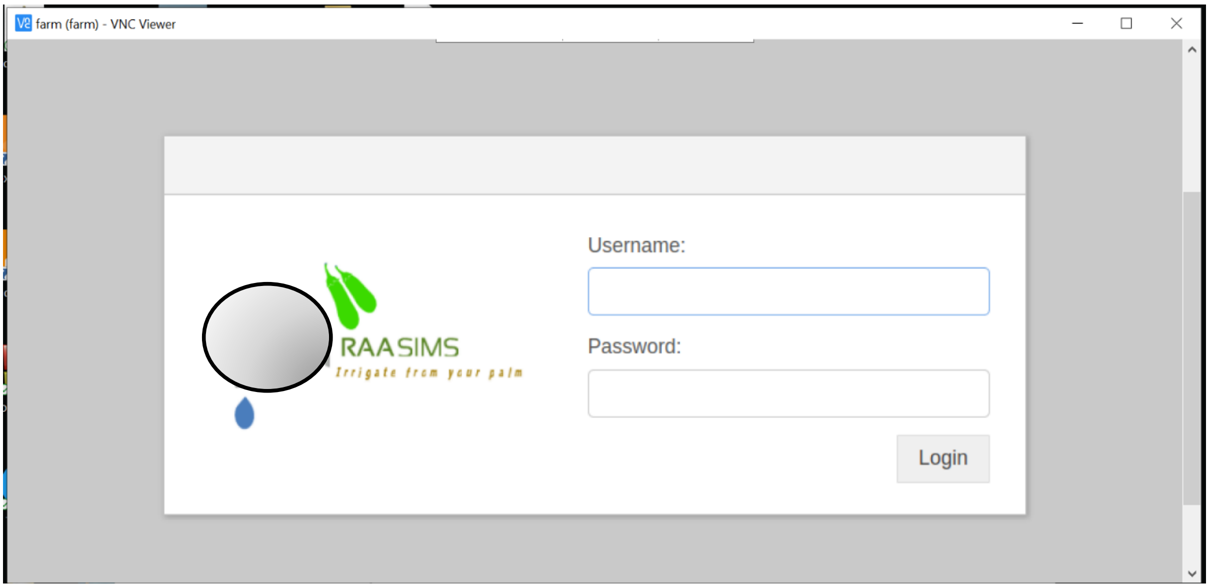


Figure 36: Log-in page

For a normal user, after successful log-in, a page is displayed that has a sub-menu on the left as shown in Fig. 37, where a user can either select to view the irrigation zones or the pump and valves zone.

The irrigation zones provide a dashboard where all sensor data is presented to the user for monitoring. On the other hand, the pump and valve section presented in Fig. 37 is for overriding the automatic operation of the system to allow a user to remotely turn on/off the pump and the valves using the provided dashboard buttons.

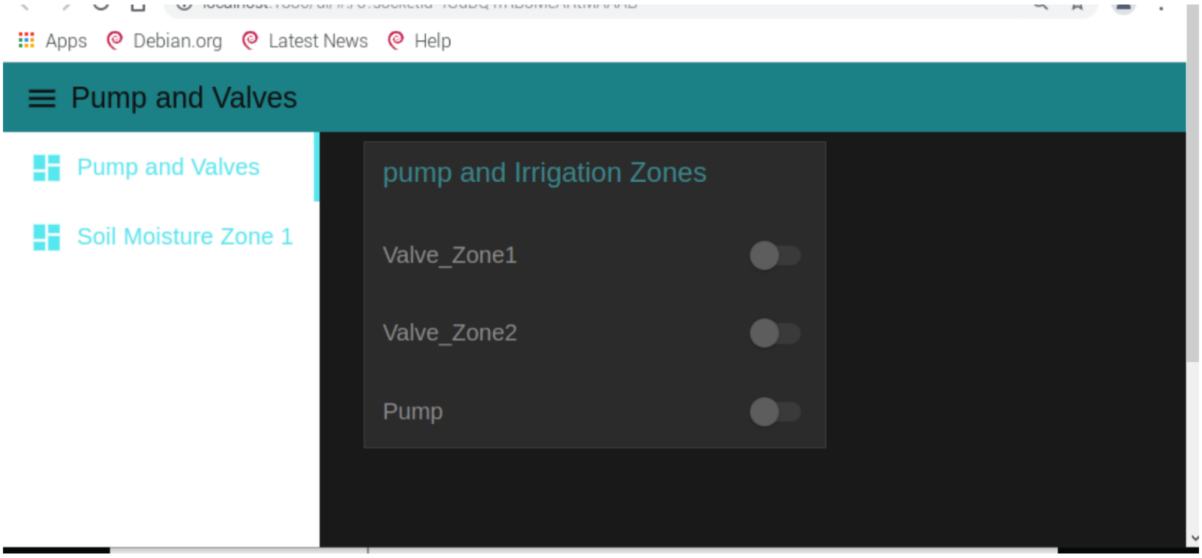


Figure 37: Irrigation zones, pump and valves sub-menu

The irrigation zone sub-menu presented in Fig. 38 allows the users to access real-time dashboard that presents the soil data from the wireless sensors. Node-red gauges were used to present the different soil parameters being monitored. The color on the gauge's changes based on the readings and whether or not they have exceeded the threshold. If the threshold is exceeded or not met, the gauge color changes to red. As shown in Fig. 38 the numerical values of the sensor data are displayed below the gauges, for clarity to the user.

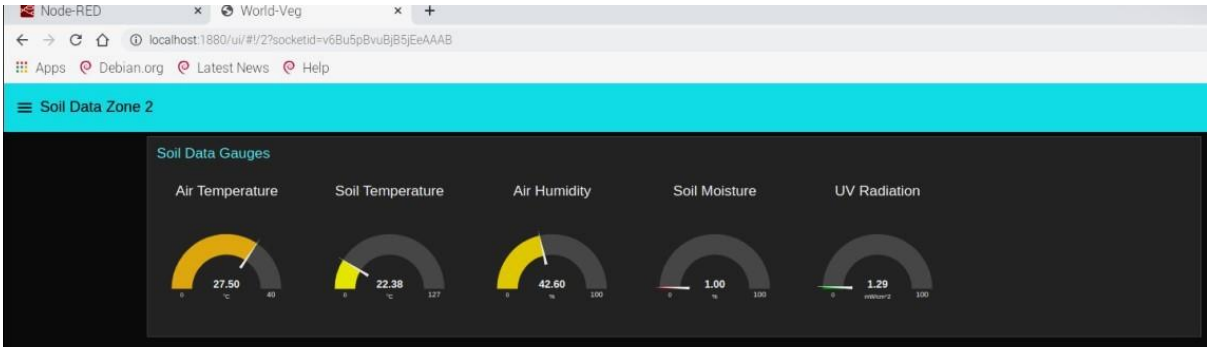


Figure 38: Irrigation zone sub-menu and the dashboard for dry soil sample

Figure 39 presents the pump and valves section for remote control of the pump and valves.

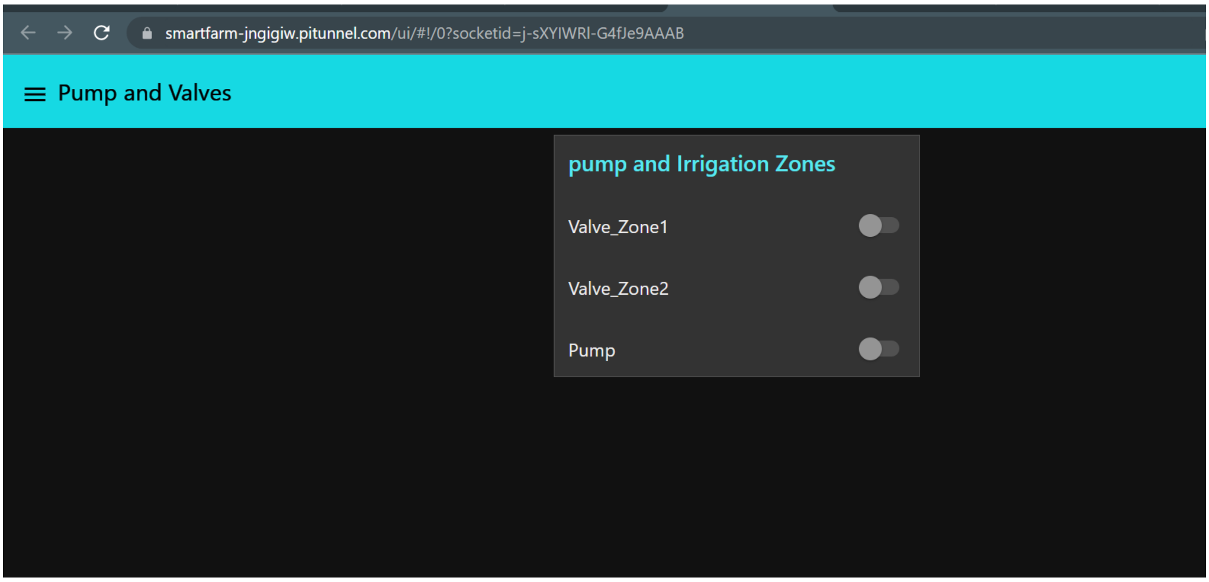


Figure 39: Pump and valve control sub-menu

4.3.2 Firmware Sub-section

The C++ programming language and embedded C concepts were used to customize the firmware running on the microcontrollers on the wireless sensor nodes. The modular code for the wireless sensor nodes, which is generic for both, will be described in this sub-section.

The modules were categorized into functions. For the temperature and humidity function, float variables store the two parameters after reading the DHT sensor. Figure 40 presents the function that prints the values to the 2004-character LCD.

```
void getDhtData()
{
  //delay(2000);   //first wait for two seconds

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  tem = String(temperature);
  hum = String(humidity);
  Serial.println("temp:");
  Serial.println(tem);

  Serial.println("humidity:");
  Serial.println(hum);
  lcd.setCursor(3,0);
  lcd.print(temperature,1);
  lcd.print((char)223);
  lcd.print("C");
  lcd.setCursor(13,0);
  lcd.print(humidity,1);
  lcd.print("%");
}
```

Figure 40: Function for reading ambient temperature and humidity

Similarly, a function to compute the soil moisture as shown in Fig. 41 was developed.

```
float getMoisture()
{
  for (int i = 0; i <= 100; i++)
    {
      soilSensorValue = soilSensorValue + analogRead(sensorPin);
      delay(1);
    }
    soilSensorValue = soilSensorValue/100.0;
    soilSensorValue = map(soilSensorValue,4172, 2243, 0, 100);    /*Convert the analog sensor values to percentage
    //moist = String(soilSensorValue);                                In plain water (max), analog value = 2243.0 but
    Serial.println("Moisture");                                       in dry in dry soil (min), analog val = 4172   */
    Serial.println(soilSensorValue);
    Serial.println("\n");
    lcd.setCursor(12,2);
    lcd.print(soilSensorValue);
    lcd.setCursor(18,2);
    lcd.print("%");
    return soilSensorValue;

}
```

Figure 41: Function to compute soil moisture content

The soil temperature was determined using a ds18b20 sensor. The function was developed as shown in Fig. 42 to send a command to the ds18b20 I2C interface in order to retrieve the temperature in degree Celsius.

```
void getSoilTemp()
{
  //issue command to get temperature data
  sensors.requestTemperatures();
  Serial.println("Soil Temperature: ");
  Serial.println(sensors.getTempCByIndex(0));
  lcd.setCursor(12,1);
  lcd.print(sensors.getTempCByIndex(0));
  lcd.print((char)223);
  lcd.print("C");

  //update after one second
  //delay(1000);
}
```

Figure 42: Function to display soil temperature

Transmission of the sensor data packets from the wireless sensor nodes to the coordinator node was achieved using the ZigBee protocol. The specific radios used were the XBee S2C modules from DIGI International due to their well-documented API transmission mode that allows multiple radios (routers) to transmit data to a coordinator node securely, while exposing their address of origin. Figure 43 shows the function to transmit the data packets for the different sensors.

```
void sendPacket() {
    // Prepare the Zigbee Transmit Request API packet
    ZBTxRequest txRequest;
    txRequest.setAddress64(0x0000000000000000);

    // Allocate 21 payload bytes: 1 type byte plus five floats of 4 bytes each
    AllocBuffer<21> packet;

    // Packet type, temperature, humidity, soil moisture, soil temp, uv intensity
    packet.append<uint8_t>(1);
    packet.append<float>(dht.readTemperature());
    packet.append<float>(dht.readHumidity());
    packet.append<float>(sensors.getTempCByIndex(0));
    packet.append<float>(soilSensorValue);
    packet.append<float>(getUVintensity());
    txRequest.setPayload(packet.head, packet.len());

    // And send it
    xbee.send(txRequest);
}

unsigned long last_tx_time = 0;
```

Figure 43: Function to transmit data packets using XBee

4.3.3 Hardware Submodule

The hardware sub-module is a combination of the wireless sensor nodes, the coordinator as well as the actuators, that is, pump and valves.

To improve the versatility of the wireless sensor nodes while out in the farm, mini-solar panels were utilized, to charge the lithium batteries. This is to ensure they are continuously powered while in the field.

In this sub-section, the laboratory experimental set-up of the hardware sub-module will be presented. The illustration in Fig. 44 presents the wireless sensor nodes.

Figure 44: Wireless sensor nodes prototype

After the assembly of the sensors and the integration of the circuits, the power module was tested and calibration of the soil moisture sensor carried out using the test set-up presented in Fig. 45 in a controlled environment.

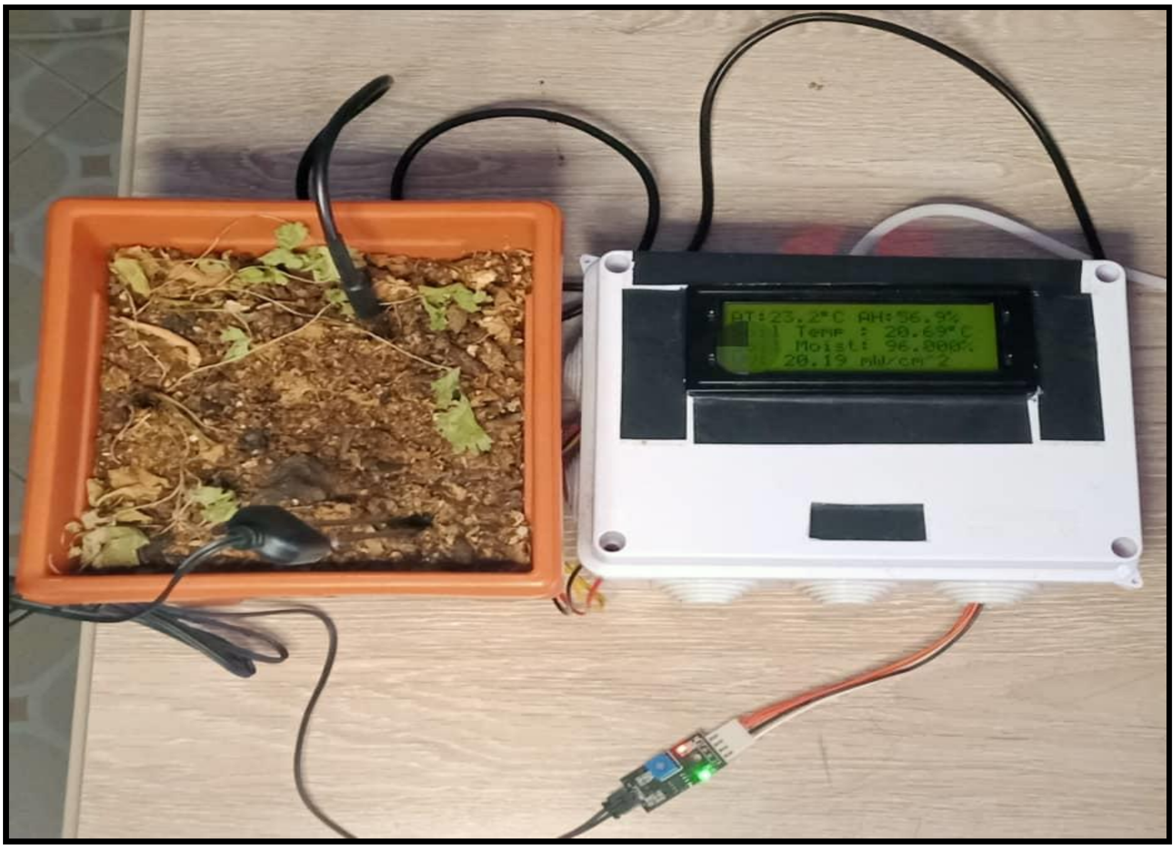


Figure 45: Wireless sensor node calibration set-up

Table 6: Test results

S/N	Requirement	Result
1.	The system shall enable remote control of the pump and valves from a PC or mobile device by the irrigation technician.	PASS
2.	The system shall factor in time	PASS
3.	The system shall monitor soil moisture and soil temperature	PASS
4.	The system shall monitor weather parameters mainly: Air temperature and humidity, ultraviolet radiation	PASS
5.	The system shall automatically switch the pump and valves on and off based on soil moisture content	PASS
6.	The system shall present real-time data to users, both admin and irrigation technician.	PASS
7.	The system shall send notification alerts to both admin and irrigation technician.	PASS
8.	The system shall have a user interface to monitor and control the system	PASS

The system was tested against the user requirements and the results are documented in Table 6. The detailed test execution plan is presented in Appendix 5. Remote control functionality was tested from both a mobile device and a computer. Using Pi-tunnel, a custom domain name was provisioned, which when accessed over the browser, linked to the Node-red web application hosted on the Raspberry Pi.

As for time, the duration for which the valves and pump were kept and was implemented in the control algorithm and measured for validation during the test.

All sensors were calibrated and benchmarked with industrial counterparts to check for deviations. The automatic switching on and off for the valves and pump was tested in different soil conditions and the results were tabulated.

The data from the soil sensors was observed in real-time and the variations were monitored to ensure any changes in soil or weather data were reflected in the real-time gauges.

Notification alerts were sent using a Twilio subscription for predefined events, including when the soil was dry.

4.4 System Validation Results

This section highlights a section of the user acceptance survey and the sample responses. In order to gather insights regarding the usability of the system, a system demo was run for intended users, and a survey was issued afterward to collect user views. The survey was carried out with three farmers, one supervisor, and one irrigation technician. The participants were selected based on convenience sampling due to constraints in terms of schedules. Sample responses from users have been presented in Figs. 46, 47 and 48.

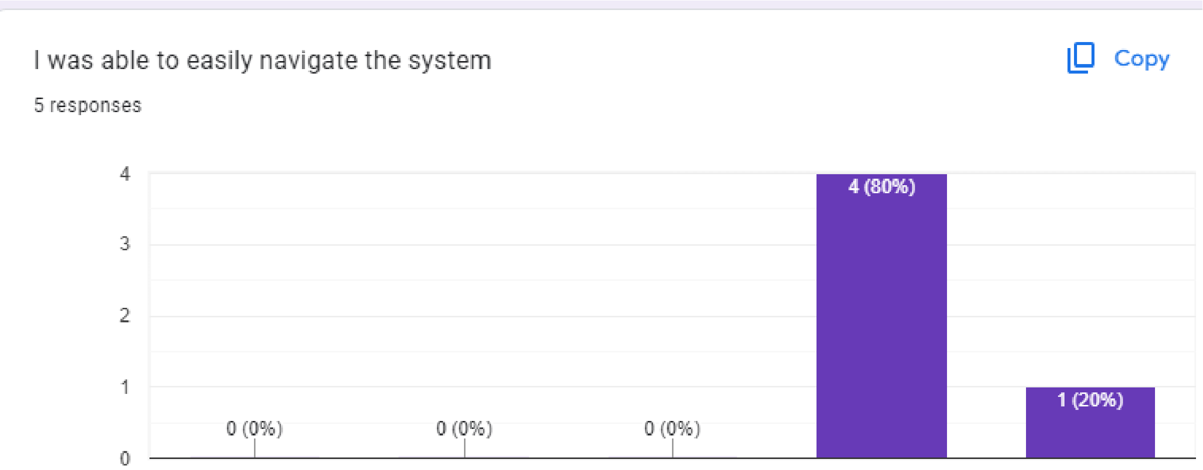


Figure 46: System ease of use

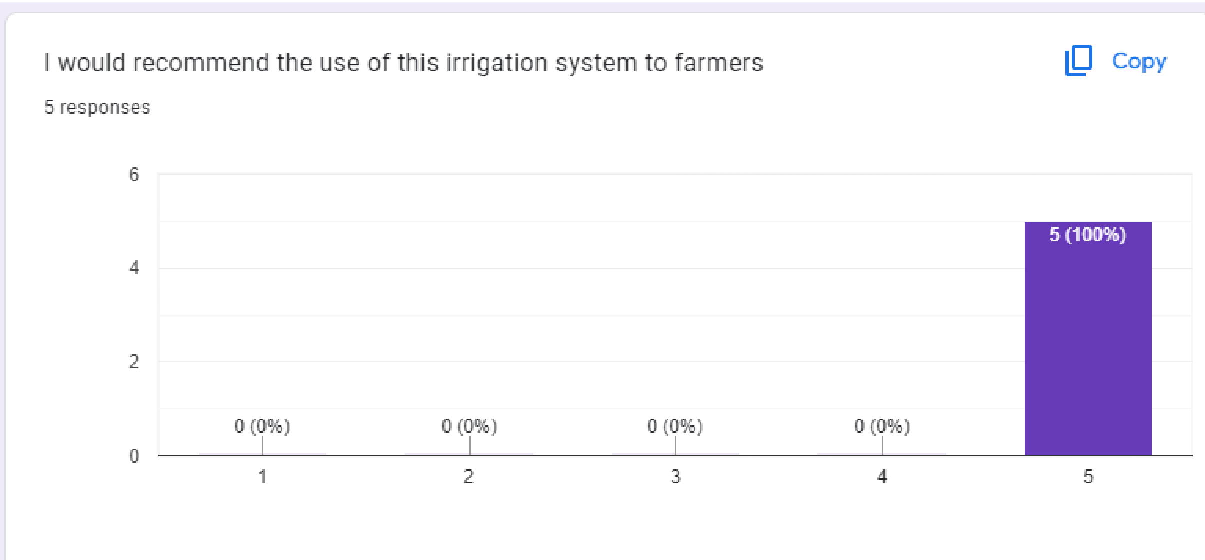
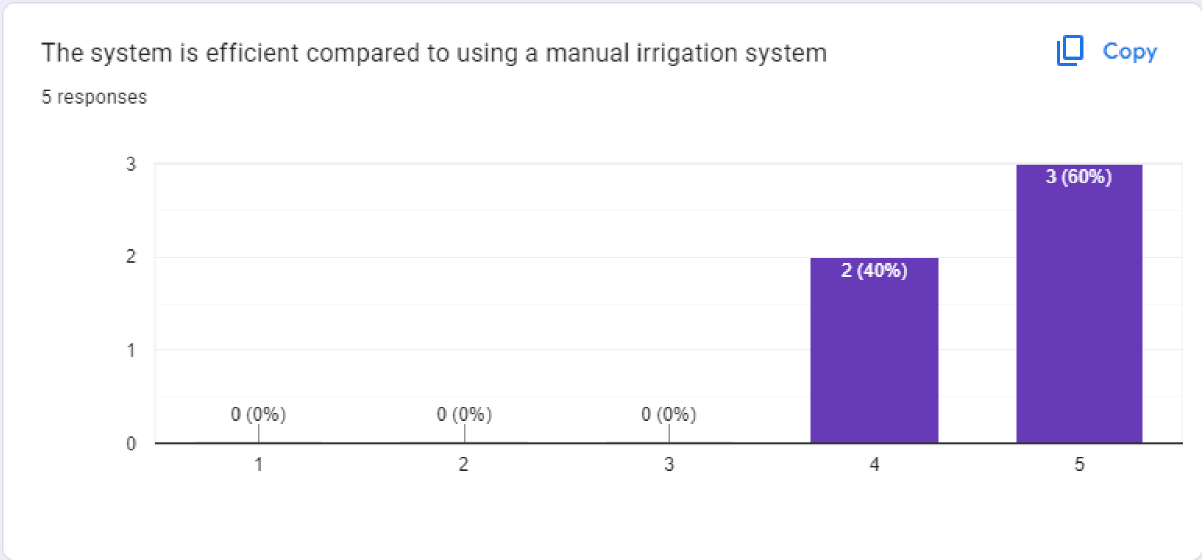


Figure 47: Users recommending the system



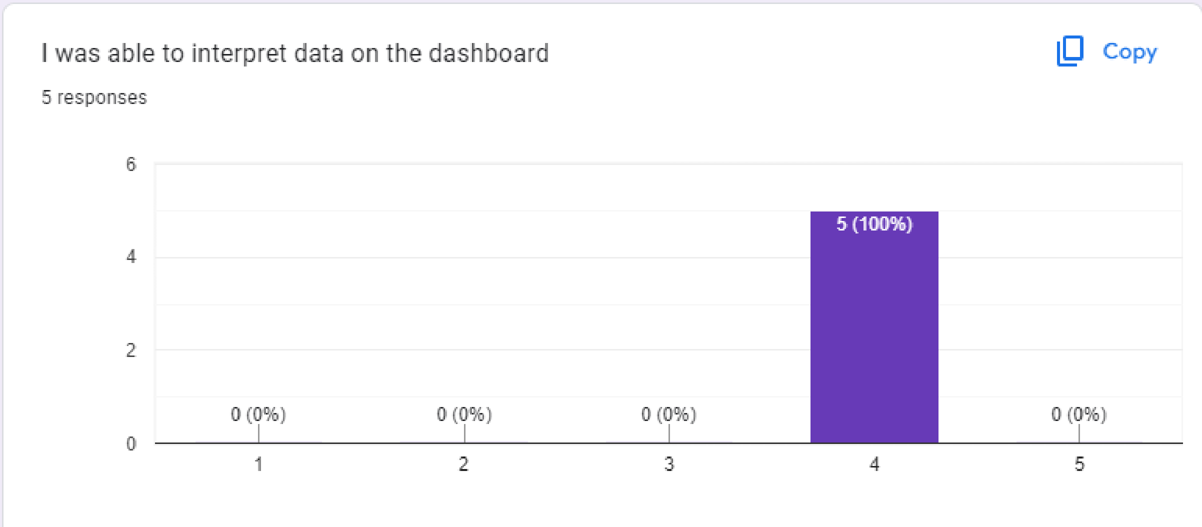


Figure 48: System efficiency and data presentation

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In the confines of this study, an IoT-based Automated Surface-drip Irrigation Monitoring System was developed. This system addresses key challenges related to manual irrigation monitoring systems by incorporating low-power wireless sensor networks using ZigBee protocol to collect and transmit soil parameter data such as soil moisture, soil temperature, air humidity, air temperature as well as UV-radiation. The real-time transmission of this data to a central coordinator node allows for rapid processing and visualization, enabling users to access the data and control valves and pumps remotely. The XBee wireless modules were programmed in API mode to ensure scalability by allowing the addition of multiple sensor nodes in the future as per requirements. Furthermore, programming the wireless sensor nodes in API mode allows addressing thus enabling easy identification of the source of data using the 64-bit MAC address. The system was validated by carrying out usability tests with five farm field staff to gather insights regarding the suitability of the prototype for surface-drip irrigation kits in a select vegetable farm.

5.2 Recommendations

Firstly, to the policy makers, this IoT-based Automated Surface-drip Irrigation Monitoring System Prototype can assist farmers and agricultural production centers to maximize on technology in order to automate daily recurring irrigation activities, which will help conserve energy and save on time by improving efficiency. The Ministry of Agriculture should come up with initiatives to empower youth and small-scale farmers by enabling them access credit facilities to help them acquire the drip irrigation kits fitted with the monitoring system which in turn will motivate the population to venture into agriculture and eventually attain food security at a national and global level.

Secondly, to the farmers, in order to attain the full benefits of the system, a reliable internet connectivity is required in order for the system to operate seamlessly, specifically for remote control and access of sensor data. For local access, internet connectivity is not mandatory as long as the client device is in the same local network as the Raspberry Pi server which acts as the coordinator node.

In future, this work can be improved by leveraging on more sensors and virtual reality techniques to provide more visualizations and interaction. High-definition cameras can be used to acquire real-time images of the farm and capture more spatial data to aid in decision-making. A mobile application can also be developed and integrated with the existing web application for easy monitoring for users who rely on mobile hand-held devices. Furthermore, machine learning algorithms can be developed to enable the prediction of crop water requirements and more autonomous irrigation monitoring.

For practicality, this system can be deployed in various farm settings such as mid-sized farms and large-sized farms. The sensor nodes can be scaled in the order of hundreds or thousands and monitored extensively to determine performance metrics that could not be determined with this prototype that only used two sensor nodes and one coordinator node.

REFERENCES

Abbassi, Y., & Benlahmer, H. (2021). The Internet of Things at the service of tomorrow's agriculture. *Procedia Computer Science, 191*, 475–480.

Abdollahi, A., Rejeb, K., Rejeb, A., Mostafa, M. M., & Zailani, S. (2021). Wireless Sensor Networks in Agriculture: Insights from Bibliometric Analysis. *Sustainability*, *13*(21), 12011. https://doi.org/10.3390/su132112011.

Abhimayu, R. (2018). Electronics For You [Corporate]. *KiCad: An Electronics Schematic Design and Layout Creation Suite*. https://www.electronicsforu.com/buyers-guides/software-buyers-guide/kicad-electronics-schematic-design.

Abioye, E. A., Abidin, M. S. Z., Mahmud, M. S. A., Buyamin, S., AbdRahman, M. K. I., Otuoze, A. O., Ramli, M. S. A., & Ijike, O. D. (2021). IoT-based monitoring and data-driven modeling of drip irrigation system for mustard leaf cultivation experiment. *Information Processing in Agriculture*, *8*(2), 270–283.

Africa RISING. (2018). Africa RISING [Corporate]. *About Us*. https://africa-rising.net/about-us.

Ali, H. (2010). *Fundamentals of irrigation and on-farm water management: Volume 1*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4419-6335-2.

Ançay, A., Michel, V., & Baroffio, C. A. (2017). Comparison of manual and automatic irrigation systems in strawberries. *Acta Horticulturae, 1156*, 263–268. https://doi.org/10.17660/ActaHortic.2017.1156.40.

Antonella Del, R. (2015). CERN Bulletin [Corporate]. *KICAD Challenges the Big Ones*. https://cds.cern.ch/journal/CERNBulletin/2015/09/News%20Articles/1988318.

Benyezza, H., Bouhedda, M., & Rebouh, S. (2021). Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving. *Journal of Cleaner Production, 302,* 127001. https://doi.org/10.1016/j.jclepro.2021.127001.

Bjorneberg, D. L. (2013). Irrigation | Methods. In *Reference Module in Earth Systems and Environmental Sciences*, B9780124095489053000. https://doi.org/10.1016/B978-0-12-409548-9.05195-2.

Canales-Ide, F., Zubelzu, S., & Rodríguez-Sinobas, L. (2019). Irrigation systems in smart cities coping with water scarcity: The case of Valdebebas, Madrid (Spain). *Journal of Environmental Management*, *247*, 187–195.

Casadei, S., Peppoloni, F., Ventura, F., Teodorescu, R., Dunea, D., & Petrescu, N. (2021). Application of smart irrigation systems for water conservation in Italian farms. *Environmental Science and Pollution Research*, *28*(21), 26488–26499. https://doi.org/10.1007/s11356-021-12524-6.

Cooper, M., Müller, B., Cafiero, C., Laso Bayas, J. C., Crespo Cuaresma, J., & Kharas, H. (2021). Monitoring and projecting global hunger: Are we on track? *Global Food Security*, *30*, 100568. https://doi.org/10.1016/j.gfs.2021.100568.

Debauche, O., Mahmoudi, S., & Guttadauria, A. (2022). A New Edge Computing Architecture for IoT and Multimedia Data Management. *Information*, *13*(2), 89. https://doi.org/10.3390/info13020089.

Digi International Inc. (2020). *XBee/XBee-PRO S2C ZigBee User Guide*. Digi International Inc. https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf.

Domínguez-Niño, J. M., Oliver-Manera, J., Girona, J., & Casadesús, J. (2020). Differential irrigation scheduling by an automated algorithm of water balance tuned by capacitance-type soil moisture sensors. *Agricultural Water Management*, *228*, 105880. https://doi.org/10.1016/j.agwat.2019.105880.

Dorward, A., & Giller, K. E. (2022). Change in the climate and other factors affecting agriculture, food or poverty: An opportunity, a threat or both? A personal perspective. *Global Food Security*, *33*, 100623. https://doi.org/10.1016/j.gfs.2022.100623.

Douglass, B. (2013). Agile Development for Embedded Systems. *Software Engineering for Embedded Systems*, 731–766. Elsevier. https://doi.org/10.1016/B978-0-12-415917-4.00021-9.

El-Abd, M. (2017). A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions. *International Journal of Engineering Pedagogy*, *7*(2), 79. https://doi.org/10.3991/ijep.v7i2.6845.

Elleuchi, M., Khelif, R., Kharrat, M., Aseeri, M., Obeid, A., & Abid, M. (2019). Water Pipeline Monitoring and Leak Detection using soil moisture Sensors: IoT based solution. *2019 16th International Multi-Conference on Systems, Signals & Devices*, 772–775. https://doi.org/10.1109/SSD.2019.8893200.

Elshaikh, A. E., Jiao, X., & Yang, S. (2018). Performance evaluation of irrigation projects: Theories, methods, and techniques. *Agricultural Water Management*, *203*, 87–96. https://doi.org/10.1016/j.agwat.2018.02.034.

Evans. (2021). *Introduction*. The KiCad Team.

Ezzahoui, I., Abdelouahid, R. A., Taji, K., & Marzak, A. (2021). Hydroponic and Aquaponic Farming: Comparative Study Based on Internet of Things IoT technologies. *Procedia Computer Science*, *191*, 499–504. https://doi.org/10.1016/j.procs.2021.07.064.

FAO. (2021). *World Food and Agriculture - Statistical Yearbook 2021*. Food & Agriculture Org.

Fotek Controls LTD. (2022). Fotek Controls Limited [Corporate]. *Solid State Module*. https://www.fotek.com.tw/en-gb/product/802.

García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors*, *20*(4), 1042. https://doi.org/10.3390/s20041042.

González-Perea, R., Mérida-García, A., Fernández-García, I., Poyato, C. E., Montesinos, P., & Rodríguez-Díaz, J. A. (2019). Middleware to Operate Smart Photovoltaic Irrigation Systems in Real Time. *Water*, *11*(7), 1508.

Gumisiriza, M. S., Kabirizi, J. M. L., Mugerwa, M., Ndakidemi, Patrick. A., & Mbega, E. R. (2022). Can soilless farming feed urban East Africa? An assessment of the benefits and challenges of hydroponics in Uganda and Tanzania. *Environmental Challenges*, *6*, 100413. https://doi.org/10.1016/j.envc.2021.100413.

Gutierrez, J., Villa-Medina, J. F., Nieto-Garibay, A., & Porta-Gandara, M. A. (2014). Automated Irrigation System Using a Wireless Sensor Network and GPRS Module.

IEEE Transactions on Instrumentation and Measurement, *63*(1), 166–176. https://doi.org/10.1109/TIM.2013.2276487.

Haule, J., & Michael, K. (2014). Designing and Simulation of an Automated Irrigation Management System Deployed by using Wireless Sensor Networks (WSN). *Journal of Electronics and Communication Engineering*, *9*(5), 67–73.

Hochmuth, G. (2017). Drip Irrigation. *A Guide to the Manufacture, Performance, and Potential of Plastics in Agriculture*, 79–105. Elsevier. https://doi.org/10.1016/B978-0-08-102170-5.00005-1.

Hunter Industries. (2018, May 6). Hydrawise [Corporate]. *Hydrawise HCC*. https://www.hunterindustries.com/irrigation-product/hydrawise/hcc.

ICStation. (2020). *Soil Hygrometer Detection Humidity Sensor Module Corrosion Resistance Probe DC 3.3-12V for Arduino* [Corporate]. https://www.icstation.com/soil-moisture-hygrometer-detection-humidity-sensor-module-corrosion-resistance-probe-arduino-p-12322.html.

Kaisti, M., Mujunen, T., Mäkilä, T., Rantala, V., & Lehtonen, T. (2014). Agile Principles in the Embedded System Development. In G. Cantone & M. Marchesi (Eds.), *Agile Processes in Software Engineering and Extreme Programming*, *179*, 16–31. Springer International Publishing. https://doi.org/10.1007/978-3-319-06862-6_2.

Klaina, H., Guembe, I. P., Lopez-Iturri, P., Campo-Bescós, M. Á., Azpilicueta, L., Aghzout, O., Alejos, A. V., & Falcone, F. (2022). Analysis of low power wide area network wireless technologies in smart agriculture for large-scale farm monitoring and tractor communications. *Measurement*, *187*, 110231.

Koech, R., & Langat, P. (2018). Improving Irrigation Water Use Efficiency: A Review of Advances, Challenges and Opportunities in the Australian Context. *Water*, *10*(12), 1771. https://doi.org/10.3390/w10121771.

Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. *Computer Science Review*, *40*, 100364. https://doi.org/10.1016/j.cosrev.2021.100364.

Lady, A. (2021). *DHT11, DHT22 and AM2302 Sensors*. Adafruit Industries. https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf.

Langemeyer, J., Madrid-Lopez, C., Mendoza Beltran, A., & Villalba Mendez, G. (2021). Urban agriculture. A necessary pathway towards urban resilience and global sustainability? *Landscape and Urban Planning*, *210*, 104055.

Last Minute Engineers. (2019). How Rain Sensor Works and Interface it with Arduino [Corporate]. *Last Minute Engineers*. https://lastminuteengineers.com/rain-sensor-arduino-tutorial.

Li, L., Xiaoguang, H., Ke, C., & Ketai, H. (2011). The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid. *6th IEEE Conference on Industrial Electronics and Applications*, 789–793.

Lo, L. (2009). Sampling. In *International Encyclopedia of Human Geography*, 67–77. Elsevier. https://doi.org/10.1016/B978-0-08-102295-5.10444-5.

Long, S. (2019). Raspberry Pi News [Corporate]. *Buster: The New Version of the Raspbian*. https://www.raspberrypi.com/news/buster-the-new-version-of-raspbian.

Maddikunta, P. K. R., Pham, Q. V., B, P., Deepa, N., Dev, K., Gadekallu, T. R., Ruby, R., & Liyanage, M. (2022). Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration*, *26*, 100257. https://doi.org/10.1016/j.jii.2021.100257.

Masseroni, D., Arbat, G., & De-Lima, I. P. (2020). Editorial. Managing and Planning Water Resources for Irrigation: Smart-Irrigation Systems for Providing Sustainable Agriculture and Maintaining Ecosystem Services. *Water*, *12*(1), 263.

McLellan, J., Young, W. A., Levin, E. C., & Johnson, L. W. (2021). Developing Innovative Integrated Business Solutions Using a Scrum Project Management Methodology. *Businesses*, *1*(2), 91–101. https://doi.org/10.3390/businesses1020007.

Mohammed, M., Riad, K., & Alqahtani, N. (2021). Efficient IoT-Based Control for a Smart Subsurface Irrigation System to Enhance Irrigation Management of Date Palm. *Sensors*, *21*(12), 3942. https://doi.org/10.3390/s21123942.

Montesano, F. F., van Iersel, M. W., Boari, F., Cantore, V., D'Amato, G., & Parente, A. (2018). Sensor-based irrigation management of soilless basil using a new smart irrigation system: Effects of set-point on plant physiological responses and crop performance. *Agricultural Water Management*, *203*, 20–29.

Nasirahmadi, A., & Hensel, O. (2022). Toward the Next Generation of Digitalization in Agriculture Based on Digital Twin Paradigm. *Sensors*, *22*(2), 498. https://doi.org/10.3390/s22020498.

Nechifor, V., & Winning, M. (2019). Global crop output and irrigation water requirements under a changing climate. *Heliyon*, *5*(3), e01266.

Njuu, K. (2016). *Sensor based method for water monitoring in smallholder irrigated agriculture in Tanzania* (Masters Dissertation, Nelson Mandela African Institution of Science and Technology, Tanzania).

O'Leary, N. (2022). Node-RED. *Version 2.2 Released*.

Panigrahi, P., Raychaudhuri, S., Thakur, A. K., Nayak, A. K., Sahu, P., & Ambast, S. K. (2019). Automatic drip irrigation scheduling effects on yield and water productivity of banana. *Scientia Horticulturae*, *257*, 108677. https://doi.org/10.1016/j.scienta.2019.108677.

Pawlak, K., & Kołodziejczak, M. (2020). The Role of Agriculture in Ensuring Food Security in Developing Countries: Considerations in the Context of the Problem of Sustainable Food Production. *Sustainability*, *12*(13), 5488. https://doi.org/10.3390/su12135488.

PiTunnel. (2021). [Corporate]. *PiTunnel: Access Your Raspberry Pi Projects from Anywhere*. https://www.pitunnel.com.

Raspberry Pi. (2021). *Raspberry Pi 4 Computer Model B*. Raspberry Pi Trading Ltd. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

Saha, H., Mandal, S., Mitra, S., Banerjee, S., & Saha, U. (2017). Comparative performance analysis between nRF24L01+ and XBEE ZB module-based wireless Ad-Hoc networks. *International Journal of Computer Network and Information Security*, *9*(7), 36-44. https://doi.org/10.5815/ijcnis.2017.07.05.

SEAFLO. (2021). SEAFLO electric powered bilge pump. SEAFLO. Retrieved from http://www.seaflo.com/en-us/product/detail/618.html.

Selmani, A., Outanoute, M., Alaoui, M. A., El Khayat, M., Guerbaoui, M., Ed-dahhak, A., Lachhab, A., & Bouchikhi, B. (2018). Multithreading design for an embedded irrigation system running on solar power. *4th International Conference on Optimization and Applications*, 1–5. https://doi.org/10.1109/ICOA.2018.8370519.

Sinde, R., Begum, F., Njau, K., & Kaijage, S. (2020). Refining Network Lifetime of Wireless Sensor Network Using Energy-Efficient Clustering and DRL-Based Sleep Scheduling. *Sensors*, *20*(5), 1540. https://doi.org/10.3390/s20051540.

STMicroelectronics. (2021). *RM0008 reference manual*. STMicroelectronics. Retrieved from https://www.st.com/resource/en/reference_manual/cd00171190-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf.

STMicroelectronics. (2022a). *AN2606 application note*. STMicroelectronics. Retrieved from https://www.st.com/resource/en/application_note/cd00167594-stm32-microcontroller-system-memory-boot-mode-stmicroelectronics.pdf.

STMicroelectronics. (2022b). *UM2237 user manual*. STMicroelectronics. Retrieved from https://www.st.com/resource/en/user_manual/dm00403500-stm32cubeprogrammer-software-description-stmicroelectronics.pdf.

TCAA. (2021). *TCAA Advisory Circular: Remotely Piloted Aircraft Systems, Aerial Work Fees and Charges*. https://tcaa.go.tz/web/ctrback/docs/bqqP0yuF3I8e20220425103606.pdf.

The World Bank. (2020). *The World Bank population, total*. The World Bank. Retrieved from https://data.worldbank.org/indicator/SP.POP.TOTL.

Tiglao, N. M., Alipio, M., Balanay, J. V., Saldivar, E., & Tiston, J. L. (2020). Agrinex: A low-cost wireless mesh-based smart irrigation system. *Measurement*, *161*, 107874. https://doi.org/10.1016/j.measurement.2020.107874.

United Nations. (2012). *Zero Hunger Challenge: Pathways to Zero Hunger*. Retrieved from https://www.un.org/zerohunger/content/pathways-zero-hunger#:~

=The%20Zero%20Hunger%20Challenge%20was,inclusive%20and%20sustainable%20food %20systems.

U.S. Solid. (2022). U.S. SOLID [Corporate]. U.S Solid Motorized Ball Valve 34 Brass Electrical Ball Valve with Standard Port 9, 24 VDC, 5 Wire Setup. https://ussolid.com/u-s-solid-motorized-ball-valve-3-4-brass-electrical-ball-valve-with-standard-port-9-24-v-dc-5-wire-setup.html.

Vera, J., Conejero, W., Mira-García, A. B., Conesa, M. R., & Ruiz-Sánchez, M. C. (2021). Towards irrigation automation based on dielectric soil sensors. *The Journal of Horticultural Science and Biotechnology*, *96*(6), 696–707.

Vij, A., Vijendra, S., Jain, A., Bajaj, S., Bassi, A., & Sharma, A. (2020). IoT and Machine Learning Approaches for Automation of Farm Irrigation System. *Procedia Computer Science*, *167*, 1250–1257. https://doi.org/10.1016/j.procs.2020.03.440.

Vimal, S. P., Kumar, S. N., Kasiselvanathan, M., & Gurumoorthy, K. B. (2021). Smart Irrigation System in Agriculture. *Journal of Physics: Conference Series*, *1917*(1), 012028. https://doi.org/10.1088/1742-6596/1917/1/012028.

World Vegetable Center. (2013). *History*. World Vegetable Center. Retrieved from https://avrdc.org/about-avrdc/history.

Wu, G., Chen, J., Kim, J. S., Gu, L., Lee, J. H., & Zhang, L. (2022). Impacts of climate change on global meteorological multi-year droughts using the last millennium simulation as a baseline. *Journal of Hydrology*, *610*, 127937.

APPENDICES

Appendix 1: Sample Questionnaire

What is your gender?

- ○ Male
- ○ Female

What is your age bracket?

- ○ Below 18
- ○ 18 - 28
- ○ 29 - 39
- ○ 40-50
- ○ 51- 61
- ○ Above 61

What is your area of specialization at World Vegetable Center? e.g gene mapping

Short answer text

Does the existing Irrigation system at World Vegetable Center directly or indirectly affect your day-to-day work?

- ○ Yes
- ○ No
- ○ Maybe

Have you heard about smart irrigation systems previously?

- Yes
- No

Overall how satisfied are you with the existing irrigation system at world vegetable center?

	1	2	3	4	5	
Not satisfied	○	○	○	○	○	Very satisfied

What reasons do you have for your answer above?

Long answer text

Do you think an automated irrigation system is good to have?

- Yes
- No
- Maybe

Why?

Long answer text

What would you say annoys you the most about a manual irrigation system?

Long answer text

Do you have any specific features you would recommend or would like to see implemented on the automated irrigation system?

Long answer text

Appendix 2: Key Person Interview Guide

THE NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY

KEY PERSON INTERVIEW GUIDE

PROJECT TITLE: IoT Based Automated Surface-drip Irrigation Monitoring System Prototype: A Case of World Vegetable Center in Arusha Tanzania

SECTION I: INTRODUCTION

1. Please explain what the World Vegetable Center deals with.
2. How many departments exist at the center and what are their primary roles?

SECTION II: FARMERS/FARM STAFF

1. Which type of crops do you grow?
2. Are they seasonal or permanent?
3. Which type of irrigation do you use? Why?
4. How do you determine plant water requirement and irrigation schedule?
5. Is soil type an important factor in determining your irrigation requirements?
6. How do you analyze your soil?
7. How does the irrigation method determine the seedbed structure?
8. How often do you irrigate your plots?
9. What main challenges do you face currently in terms of irrigation for the different plots?

SECTION III: FARM MANAGER

1. Which type of crops are grown?
2. Are weather parameters important for plant health monitoring and irrigation planning?
3. What weather data do you monitor?
4. How is the weather data monitored currently?
5. At what frequency are the weather parameters recorded?
6. Are there any challenges with this method of collecting weather parameters?
7. What irrigation methods are used in the farm? Why?

8. What is the main source of the water used for irrigation?
9. How is the water obtained from the source to the reservoir and how does it reach the specific irrigation plots?
10. What is the main equipment used in distributing water to the plots?
11. How is the irrigation pattern predetermined?
12. Are you satisfied with the current operations of the irrigation system? Why?
13. How do you monitor the plots to determine water requirements for the crops?
14. During holidays and breaks, how are the plots monitored and irrigated? Is there a cost implication?

Appendix 3: Summary of Hardware Tools

Component	Model
Soil Moisture Sensor	ICStation
Soil Temperature Sensor	DS18B20
Ambient Temperature Humidity Sensor	AM2303
UV Sensor	GY-ML8511
Ball valve	½" DN15 Brass Motorized Ball Valve
Water pump	Seaflo 750GPH
Mini-solar panels	Generic 6 watts
Rechargeable battery	Li-Ion (1000mAh)
Water flow sensor	YFS-201
ZigBee radio Modules	XBee S2C Pro
Raspberry pi	4 model B
Sensor node MCU	STM32F103C8T6
LCD Display	2004 Liquid Crystal Display
I2C Display Adapter	PCF8574
Solar Controller and battery charging module	SD XMDZ

Appendix 4: Software Tools

Tool	Version	Use
Node-red	V2.2.2	Low code platform for creating graphical user interfaces for embedded and IoT applications using (O'Leary, 2022).
STM32Cube Programmer	V2.5.7	Flashing the bootloader and programming the STM32F103C8T6 (*Blue-pill*) microcontroller board that was deployed on the sensor nodes.
Raspberry Pi OS	Buster	Open-source operating system designed for the Raspberry Pi mini-computer, enabling users to program and control peripherals on the Raspberry Pi.
Python	3.10	Programming automation scripts for the sink node.
KiCad	6.0	Open-source electronic PCB design automation software.
Fritzing	0.9.3	First-level circuit simulation software tool, selected for the initial prototyping of the sensor node and sink node circuits.
XCTU	6.5.12	XCTU software is a tool from DIGI for configuring the XBee radio modules (Digi International Inc., 2020)
Pi-Tunnel	2023	cloud-based solution that provides hosting services for applications that run on edge devices, specifically raspberry pi.

Appendix 5: Test Execution Plan

Test Area	Test Objective	Test Steps	Pass Criteria
1. Remote control	Verify remote control of pump and valves from PC/Mobile devices	1.Type the Ip address of the coordinator node a browser. 2.Attempt to log in with correct credentials 3.Attempt remote control from PC/Mobile.	System allows successful remote control.
2. Time-based operations	Validate scheduling and execution of pump and valve operations based on time.	1. Set different time intervals for operations. 2. Confirm pump and valve actions at scheduled times.	System executes operations according to specified time intervals.
3. Environmental factors	3.1 Soil moisture	1. Validate accurate soil moisture measurement. 2. Trigger actions based on moisture thresholds.	System accurately measures soil moisture and triggers actions accordingly.
	3.2 Humidity	1. Verify accuracy of humidity measurements. 2. Confirm system response to humidity changes.	System accurately measures humidity and responds appropriately.

	3.3 Soil and ambient temperature	1. Test accuracy of temperature measurements. 2. Ensure system considers temperature in decision making.	System accurately measures temperatures and incorporates it into decisions.
4. Automatic Control	Validate autonomous switching of pump and valves based on soil moisture.	1. Simulate scenarios with different moisture levels. 2. Confirm system triggers actions to switch pump and valves on/off.	System autonomously controls pump and valves based on soil moisture content.
5. Data presentation	Confirm consistent presentation of farm soil data to users.	1. Verify real-time updates of soil data.	System consistently presents rea-time farm soil data to users.
6. Notification alerts	Ensure timely and accurate notification alerts to users.	1. Simulate scenarios triggering alerts (low soil moisture). 2. Verify SMS alert is sent to a mobile phone promptly.	System sends timely and accurate notification alerts to users.

Appendix 6: Node-red Sample Codes for Dashboard


```
[
    {
        "id": "65ad405e69f509e9",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "ad4acd4b89937f65",
        "type": "ui_group",
        "name": "Soil data gauges",
        "tab": "d937a39357882ecb",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "2370562e0dc9aa44",
        "type": "ui_base",
        "theme": {
            "name": "theme-dark",
            "lightTheme": {
                "default": "#0094CE",
                "baseColor": "#0069cc",
                "baseFont": "Lucida Sans Unicode,Lucida Grande,sans-
serif",
                "edited": true,
                "reset": false
            },
            "darkTheme": {
                "default": "#097479",
                "baseColor": "#16d9e3",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                "edited": true,
                "reset": false
            },
            "customTheme": {
                "name": "Untitled Theme 1",
                "default": "#4B7930",
                "baseColor": "#4B7930",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
            },
            "themeState": {
                "base-color": {
                    "default": "#097479",
                    "value": "#16d9e3",
                    "edited": true
                },
```

```
                "page-titlebar-backgroundColor": {
                    "value": "#16d9e3",
                    "edited": false
                },
                "page-backgroundColor": {
                    "value": "#111111",
                    "edited": false
                },
                "page-sidebar-backgroundColor": {
                    "value": "#ffffff",
                    "edited": false
                },
                "group-textColor": {
                    "value": "#57e7ef",
                    "edited": false
                },
                "group-borderColor": {
                    "value": "#555555",
                    "edited": false
                },
                "group-backgroundColor": {
                    "value": "#333333",
                    "edited": false
                },
                "widget-textColor": {
                    "value": "#eeeeee",
                    "edited": false
                },
                "widget-backgroundColor": {
                    "value": "#16d9e3",
                    "edited": false
                },
                "widget-borderColor": {
                    "value": "#333333",
                    "edited": false
                },
                "base-font": {
                    "value": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
                }
            },
            "angularTheme": {
                "primary": "indigo",
                "accents": "blue",
                "warn": "red",
                "background": "grey",
                "palette": "light"
            }
        },
        "site": {
            "name": "World-Veg",
            "hideToolbar": "false",
            "allowSwipe": "false",
            "lockMenu": "false",
```

```
        "allowTempTheme": "false",
        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 0,
            "py": 0
        }
    }
},
{
    "id": "b5f6ad382ef6bfe5",
    "type": "serial-port",
    "serialport": "/dev/ttyS0",
    "serialbaud": "9600",
    "databits": "8",
    "parity": "none",
    "stopbits": "1",
    "waitfor": "",
    "dtr": "none",
    "rts": "none",
    "cts": "none",
    "dsr": "none",
    "newline": "\\n",
    "bin": "false",
    "out": "char",
    "addchar": "",
    "responsetimeout": "10000"
},
{
    "id": "d937a39357882ecb",
    "type": "ui_tab",
    "name": "Soil Data Zone 1",
    "icon": "dashboard",
    "order": 2,
    "disabled": false,
    "hidden": false
},
{
    "id": "575bd2482fa09f10",
    "type": "ui_tab",
    "name": "Pump and Valves",
    "icon": "dashboard",
    "order": 2,
    "disabled": false,
    "hidden": false
},
{
    "id": "0a085f4e93720aba",
    "type": "ui_group",
```

```
        "name": "pump and Irrigation Zones",
        "tab": "575bd2482fa09f10",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "2ae634475497e04f",
        "type": "xbeeapi-config",
        "z": "65ad405e69f509e9",
        "name": "coord",
        "api_mode": "1",
        "module": "802.15.4",
        "convert_adc": "true",
        "vref_adc": "1200"
    },
    {
        "id": "c9f27b7c6d523eac",
        "type": "xbee-config",
        "z": "65ad405e69f509e9",
        "apiMode": "2",
        "rawFrames": true,
        "convertAdc": true,
        "vrefAdc": "1200",
        "serialPort": "/dev/ttyS0",
        "lock": false,
        "baudRate": "9600",
        "dataBits": "8",
        "stopBits": "1",
        "parity": "none",
        "bufferSize": "65536",
        "rtscts": false,
        "xon": false,
        "xoff": false,
        "xany": false,
        "vmin": "1",
        "vtime": "0"
    },
    {
        "id": "52290351eefdc721",
        "type": "twilio-api",
        "name": "FARM_ALERTS",
        "sid": "AC81ef28026de61b817d6c0a6c820fa62b",
        "from": "+16203180730"
    },
    {
        "id": "bd1859f1b8b4e34e",
        "type": "rpi-gpio out",
        "z": "65ad405e69f509e9",
        "name": "",
        "pin": "23",
        "set": true,
```

```
        "level": "1",
        "freq": "",
        "out": "out",
        "bcm": true,
        "x": 860,
        "y": 80,
        "wires": []
    },
    {
        "id": "80a7b7b7037c3769",
        "type": "debug",
        "z": "65ad405e69f509e9",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 870,
        "y": 180,
        "wires": []
    },
    {
        "id": "ccce3e3723cc185c",
        "type": "rpi-gpio out",
        "z": "65ad405e69f509e9",
        "name": "",
        "pin": "24",
        "set": true,
        "level": "1",
        "freq": "",
        "out": "out",
        "bcm": true,
        "x": 860,
        "y": 260,
        "wires": []
    },
    {
        "id": "4e64c4dd12b29a24",
        "type": "ui_switch",
        "z": "65ad405e69f509e9",
        "name": "",
        "label": "Pump",
        "tooltip": "",
        "group": "0a085f4e93720aba",
        "order": 4,
        "width": 0,
        "height": 0,
        "passthru": true,
        "decouple": "false",
        "topic": "topic",
        "topicType": "msg",
```

```
        "style": "",
        "onvalue": "false",
        "onvalueType": "bool",
        "onicon": "",
        "oncolor": "",
        "offvalue": "true",
        "offvalueType": "bool",
        "officon": "",
        "offcolor": "",
        "animate": false,
        "className": "",
        "x": 450,
        "y": 80,
        "wires": [
            [
                "bd1859f1b8b4e34e",
                "80a7b7b7037c3769"
            ]
        ]
    },
    {
        "id": "f4c43f8d4a01812e",
        "type": "ui_switch",
        "z": "65ad405e69f509e9",
        "name": "",
        "label": "Valve_Zone1",
        "tooltip": "",
        "group": "0a085f4e93720aba",
        "order": 1,
        "width": 0,
        "height": 0,
        "passthru": true,
        "decouple": "false",
        "topic": "topic",
        "topicType": "msg",
        "style": "",
        "onvalue": "false",
        "onvalueType": "bool",
        "onicon": "",
        "oncolor": "",
        "offvalue": "true",
        "offvalueType": "bool",
        "officon": "",
        "offcolor": "",
        "animate": false,
        "className": "",
        "x": 470,
        "y": 180,
        "wires": [
            [
                "ccce3e3723cc185c",
                "80a7b7b7037c3769"
            ]
        ]
```

```
},
{
    "id": "c2d5c89a5d81d7de",
    "type": "ui_switch",
    "z": "65ad405e69f509e9",
    "name": "",
    "label": "Valve_Zone2",
    "tooltip": "",
    "group": "0a085f4e93720aba",
    "order": 2,
    "width": 0,
    "height": 0,
    "passthru": true,
    "decouple": "false",
    "topic": "topic",
    "topicType": "msg",
    "style": "",
    "onvalue": "false",
    "onvalueType": "bool",
    "onicon": "",
    "oncolor": "",
    "offvalue": "true",
    "offvalueType": "bool",
    "officon": "",
    "offcolor": "",
    "animate": false,
    "className": "",
    "x": 470,
    "y": 320,
    "wires": [
        [
            "15d4628b72cdc073",
            "80a7b7b7037c3769"
        ]
    ]
},
{
    "id": "15d4628b72cdc073",
    "type": "rpi-gpio out",
    "z": "65ad405e69f509e9",
    "name": "",
    "pin": "25",
    "set": true,
    "level": "1",
    "freq": "",
    "out": "out",
    "bcm": true,
    "x": 860,
    "y": 340,
    "wires": []
},
{
    "id": "d4ecd3a45cf3da68",
    "type": "xbee-rx",
```

```
    "z": "65ad405e69f509e9",
    "name": "coor_xbee_node",
    "xBee": "c9f27b7c6d523eac",
    "x": 80,
    "y": 600,
    "wires": [
        [
            "b7dd5df40038ac08"
        ]
    ]
},
{
    "id": "b7dd5df40038ac08",
    "type": "buffer-parser",
    "z": "65ad405e69f509e9",
    "name": "",
    "data": "payload",
    "dataType": "msg",
    "specification": "spec",
    "specificationType": "ui",
    "items": [
        {
            "type": "buffer",
            "name": "source address",
            "offset": 5,
            "length": 7,
            "offsetbit": 0,
            "scale": "1",
            "mask": ""
        },
        {
            "type": "buffer",
            "name": "Air Temp",
            "offset": 16,
            "length": 4,
            "offsetbit": 0,
            "scale": "1",
            "mask": ""
        },
        {
            "type": "buffer",
            "name": "Humidity",
            "offset": 20,
            "length": 4,
            "offsetbit": 0,
            "scale": "1",
            "mask": ""
        },
        {
            "type": "buffer",
            "name": "Soil Temperature",
            "offset": 24,
            "length": 4,
            "offsetbit": 0,
```

```
                "scale": "1",
                "mask": ""
            },
            {
                "type": "buffer",
                "name": "Soil Moisture",
                "offset": 28,
                "length": 4,
                "offsetbit": 0,
                "scale": "1",
                "mask": ""
            },
            {
                "type": "buffer",
                "name": "UV Radiation",
                "offset": 32,
                "length": 4,
                "offsetbit": 0,
                "scale": "1",
                "mask": ""
            }
        ],
        "swap1": "",
        "swap2": "",
        "swap3": "",
        "swap1Type": "swap",
        "swap2Type": "swap",
        "swap3Type": "swap",
        "msgProperty": "payload",
        "msgPropertyType": "str",
        "resultType": "buffer",
        "resultTypeType": "return",
        "multipleResult": false,
        "fanOutMultipleResult": false,
        "setTopic": true,
        "outputs": 1,
        "x": 290,
        "y": 600,
        "wires": [
            [
                "9f181b868ca5e8c0"
            ]
        ]
    },
    {
        "id": "327d21e1f716700b",
        "type": "debug",
        "z": "65ad405e69f509e9",
        "name": "",
        "active": false,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
```

```
        "targetType": "full",
        "statusVal": "",
        "statusType": "auto",
        "x": 1170,
        "y": 700,
        "wires": []
    },
    {
        "id": "9f181b868ca5e8c0",
        "type": "function",
        "z": "65ad405e69f509e9",
        "name": "extract_xbee_bytes",
        "func": "/*Function to extract XBee Bytes\nVersion 1.1\nAuthor:
Joseph Wangere\nDate: 28-04-2023\n*/\n\n//define a variable to store the
payload\nvar buf = msg.payload;\n\n// Define the start and end markers
for the XBee frame\nvar startMarker = 0x7E;\nvar endMarker = 0x7E;\n\n//
Check if the buffer starts with the start marker\nif (buf[0] ===
startMarker) {\n\n  // Extract the length of the XBee frame\n  var length
= (buf[1] << 8) + buf[2];\n\n  // Extract the XBee frame data\n  var
xbeeData = buf.slice(3, length + 3);\n\n  // Extract the checksum from
the XBee frame\n  var checksum = buf[length + 3];\n\n  // Calculate the
expected checksum\n  var expectedChecksum = 0;\n  for (var i = 0; i <
length; i++) {\n    expectedChecksum += xbeeData[i];\n  }\n
expectedChecksum = 0xFF - (expectedChecksum & 0xFF);\n\n  // Check if the
calculated checksum matches the expected checksum\n  if (checksum ===
expectedChecksum) {\n\n    // Extract the data and address bytes from the
XBee frame\n    var dataBytes = xbeeData.slice(12);\n    var addressBytes
= xbeeData.slice(2, 9);\n    \n    /*msg.payload = {\n
\"dataBytes\": dataBytes,\n      \"addressBytes\": addressBytes\n
};*/\n    \n    // Convert addressBytes to hex string\n    // Convert the
addressBytes to a hex string\n    var hexAddressBytes = '';\n    for (var
j = 0; j < addressBytes.length; j++) {\n        hexAddressBytes +=
addressBytes[j].toString(16).padStart(2, '0');\n    }\n    \n    var
address = hexAddressBytes;\n    var airTemp =
dataBytes[1].toString(16).padStart(2,'0') +
dataBytes[2].toString(16).padStart(2,'0') +
dataBytes[3].toString(16).padStart(2,'0') +
dataBytes[4].toString(16).padStart(2,'0');\n    var airHum =
dataBytes[5].toString(16).padStart(2,'0') +
dataBytes[6].toString(16).padStart(2,'0') +
dataBytes[7].toString(16).padStart(2,'0') +
dataBytes[8].toString(16).padStart(2,'0');\n    var soilTemp =
dataBytes[9].toString(16).padStart(2,'0') +
dataBytes[10].toString(16).padStart(2,'0') +
dataBytes[11].toString(16).padStart(2,'0') +
dataBytes[12].toString(16).padStart(2,'0');\n    var soilMoist =
dataBytes[13].toString(16).padStart(2,'0') +
dataBytes[14].toString(16).padStart(2,'0') +
dataBytes[15].toString(16).padStart(2,'0') +
dataBytes[16].toString(16).padStart(2,'0');\n    var uvRad =
dataBytes[17].toString(16).padStart(2,'0') +
dataBytes[18].toString(16).padStart(2,'0') +
dataBytes[19].toString(16).padStart(2,'0') +
dataBytes[20].toString(16).padStart(2,'0');\n    \n    \n    \n
```

```
msg.payload = {\n      address,      \n      airTemp,\n      airHum,\n
soilTemp,\n      soilMoist,\n      uvRad\n    };\n    \n    return msg;\n
}\n}",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 350,
        "y": 480,
        "wires": [
            [
                "3c195da1730a8f8a"
            ]
        ]
    },
    {
        "id": "3c195da1730a8f8a",
        "type": "function",
        "z": "65ad405e69f509e9",
        "name": "",
        "func": "var hexString = msg.field1;\n\n/*\n//check if hex string
has 8 character length\nif (hexString.length == 8) {\n    \n    //split
the hex string into 2 4-character hex values\n    var hexValues = [\n
hexString.slice(0, 4),\n        hexString.slice(4, 8)\n    ];\n
//convert each hex value into an int\n    var integers =
hexValues.map(function(hexValue) {\n        return parseInt(hexvalue,
16);\n    });\n    \n    //combine the integers into a single 32-bit
floating point number\n    var value = new Float32Array(new
uint32Array([\n        (integers[0] << 16) + integers[1]\n
]).buffer)[0];*/\n    console.log(hexString);\n    return msg;\n\n",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 560,
        "y": 460,
        "wires": [
            [
                "5d687115d4b10b3b"
            ]
        ]
    },
    {
        "id": "5d687115d4b10b3b",
        "type": "function",
        "z": "65ad405e69f509e9",
        "name": "convert_bytestring_to_float",
        "func": "// Get hex strings from payload fields\nvar hexStrings =
[msg.payload.airTemp, msg.payload.airHum, msg.payload.soilTemp,
msg.payload.soilMoist, msg.payload.uvRad];\n\n// Convert hex strings to
buffer and read float values\nvar floatValues = hexStrings.map(hexString
=> {\n    var buffer = Buffer.from(hexString, 'hex');\n    \n    //we
```

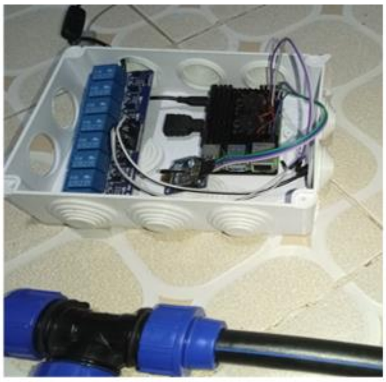
```
have to use readFloatLE buffer function to convert hex string to float\n
//since the byte order is in little endian.\n    return
buffer.readFloatLE();\n});\n\n// Update payload with float values except
for the address\nmsg.payload.address;\nmsg.payload.airTemp =
floatValues[0].toFixed(2);\nmsg.payload.airHum =
floatValues[1].toFixed(2);\nmsg.payload.soilTemp =
floatValues[2].toFixed(2);\nmsg.payload.soilMoist =
floatValues[3].toFixed(2);\nmsg.payload.uvRad =
floatValues[4].toFixed(2);\n\nreturn msg;\n",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 700,
        "y": 540,
        "wires": [
            [
                "d7ae22de0a560156",
                "327d21e1f716700b"
            ]
        ]
    },
    {
        "id": "d7ae22de0a560156",
        "type": "switch",
        "z": "65ad405e69f509e9",
        "name": "Node_1_data_switch",
        "property": "payload.address",
        "propertyType": "msg",
        "rules": [
            {
                "t": "eq",
                "v": "13a20041b8904e",
                "vt": "str"
            }
        ],
        "checkall": "true",
        "repair": false,
        "outputs": 1,
        "x": 740,
        "y": 640,
        "wires": [
            [
                "0f2bf51af3f19c2a",
                "a20d80a10d0882d5",
                "691961d5b5b0e72a"
            ]
        ]
    },
    {
        "id": "0f2bf51af3f19c2a",
        "type": "ui_gauge",
        "z": "65ad405e69f509e9",
```

```
        "name": "",
        "group": "ad4acd4b89937f65",
        "order": 0,
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "Node_1_Air_Temp",
        "label": "°C",
        "format": "{{msg.payload.airTemp}}",
        "min": 0,
        "max": "40",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "className": "",
        "x": 1090,
        "y": 780,
        "wires": []
    },
    {
        "id": "a20d80a10d0882d5",
        "type": "ui_gauge",
        "z": "65ad405e69f509e9",
        "name": "",
        "group": "ad4acd4b89937f65",
        "order": 1,
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "Node_1_air_humidity",
        "label": "%",
        "format": "{{msg.payload.airHum}}",
        "min": 0,
        "max": "100",
        "colors": [
            "#b30012",
            "#e6e600",
            "#3acb3e"
        ],
        "seg1": "",
        "seg2": "",
        "className": "",
        "x": 1100,
        "y": 840,
        "wires": []
    },
    {
        "id": "691961d5b5b0e72a",
        "type": "ui_gauge",
        "z": "65ad405e69f509e9",
```

```
        "name": "",
        "group": "ad4acd4b89937f65",
        "order": 2,
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "Node_1_soil_temperature",
        "label": "units",
        "format": "{{msg.payload.soilTemp}}",
        "min": 0,
        "max": "127",
        "colors": [
            "#b32400",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "19.1",
        "seg2": "24",
        "className": "",
        "x": 1110,
        "y": 900,
        "wires": []
    }
]
```

Appendix 7: T-Junction Pipe and Motorized Ball-valves

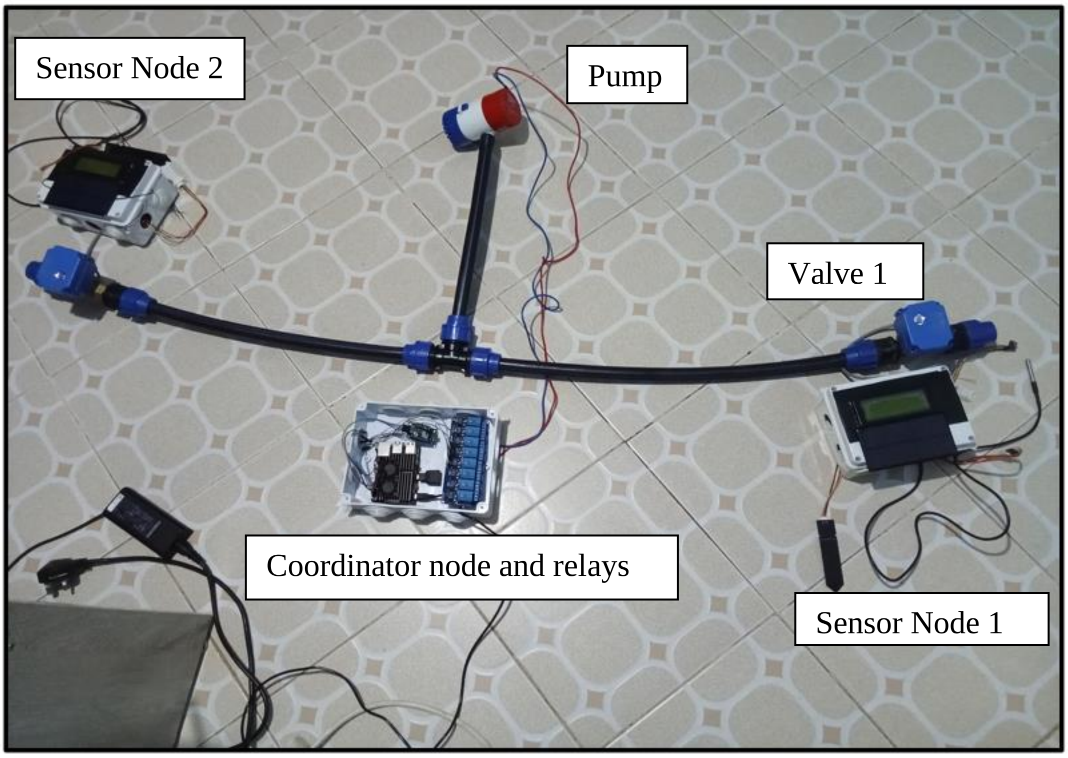
Water Pump and T-Junction pipe



Coordinator node and relays

Motorized Ball Valves

Appendix 8: Assembled Prototype



Appendix 9: User Acceptance Questionnaire

System User Acceptance Questionnaire

Please select the statement that best describes what you think about the demonstrated system prototype, on a scale of 1-5, where 1 is strongly disagree, 2 is disagree, 3 is neutral, 4 is agree and 5 is strongly agree.

I was able to easily navigate the system

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

The system is efficient compared to using a manual irrigation system

	1	2	3	4	5	
strongly disagree	○	○	○	○	○	strongly agree

I was able to interpret data on the dashboard

	1	2	3	4	5	
strongly disagree	○	○	○	○	○	strongly agree

I would recommend the use of this irrigation system to farmers

	1	2	3	4	5	
strongly disagree	○	○	○	○	○	strongly agree

RESEARCH OUTPUTS

(i) Research Paper

Wangere, J., Sinde, R., Ally, M., & Mbwambo, O. (2022). A Digital Twin Approach and Challenges for Real-Time Automated Surface-Drip Irrigation Monitoring: A Case of Arusha, Tanzania. *International Conference on Technological Advancement in Embedded and Mobile Systems*, 47-59.

(ii) Poster Presentation

Poster Presentation

An IoT-based Automated Surface-drip Irrigation Monitoring System Prototype: A Case of World Vegetable Center in Arusha Tanzania

Joseph Wangere, Ramadhani Sinde (PhD), Mussa Ally (PhD)

INTRODUCTION

This project was carried out in Arusha, Tanzania, by employing a wireless sensor network using XBee modules configured in API mode for soil-data collection and transmission, utilizing Node-RED for remote actuator control and soil data visualization.

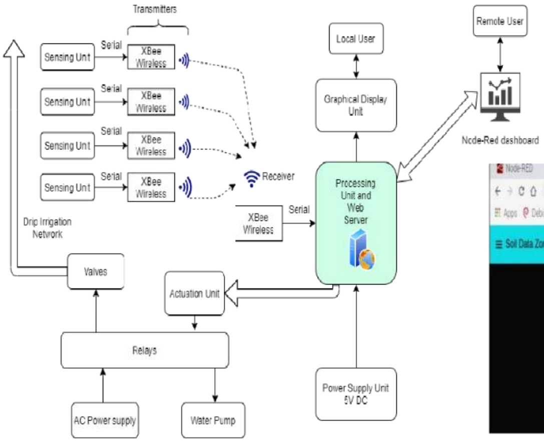
PROBLEM STATEMENT

Manual irrigation systems are labor intensive, consume time, and lead to water wastage as well as energy. This situation is worsened by a lack of monitoring mechanisms for determining water consumption per irrigation field, as well as soil and climatic data.

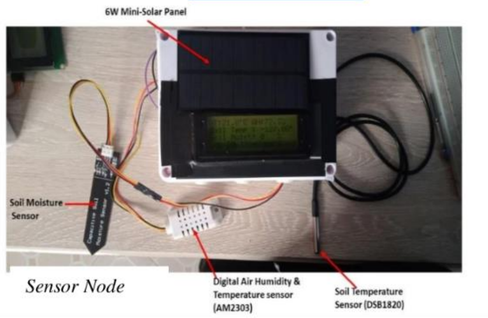
OBJECTIVE

To develop an IoT-based automated surface-drip irrigation monitoring system prototype for World Vegetable Center in Arusha Tanzania.

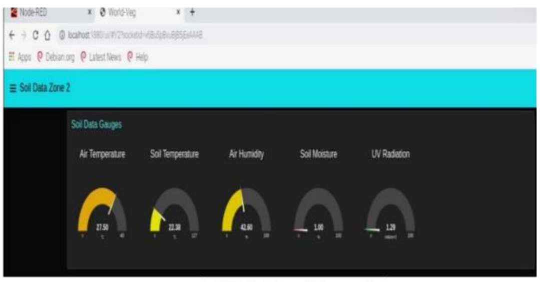
SYSTEM ARCHITECTURE



RESULTS



Sensor Node



Web Dashboard

CONCLUSION

This system automates surface-drip irrigation systems by incorporating low-power wireless sensor networks to collect and transmit soil parameter data such as soil moisture, soil temperature, air humidity, air temperature as well as UV-radiation.