

**DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY**

**PROJECT DOCUMENTATION FOR FINAL YEAR STUDY IN BACHELORS IN INFORMATION TECHNOLOGY**

**BY**

**MBUGUA DOUGLAS NG’ANG’A**

**REGISTRATION NUMBER: C025-01-1199/2019**

**PROJECT TITLE: iFARM AUTOMATED IRRIGATION SYSTEM USING AN OPEN SOURCE MICROCONTROLLER**

**PROJECT SUPERVISOR: Dr. PATRICK GIKUNDA**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF INFORMAION TECHNOLOGY IN THE SCHOOL OF COMPUTER SCIENCE AN INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT OF THE AWARD OF THE DEGREE OF BSc. INFORMATION TECHNOLOGY AT DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY**

# DECLARATION

I declare that this is my original work and has not been presented in any University for a degree or for any consideration of any certification.

NAME: Mbugua Douglas Ng’ang’a

REG. NUMBER: C025-01-1199/2019

SIGNATURE: ………………………..

DATE: …………………………

This proposal document has been submitted with my approval as the university supervisor.

NAME: Dr. Patrick Gikunda

SIGNATURE: ……………………………

DATE: ……………………………

Department of Information Technology

# ABSTRACT

This paper proposes an automated irrigation system using an ESP8266 microcontroller, that is quite cost-effective and can be employed in the farm or in an average home garden. The proposed system will automatically allow for watering of tomato plants from a water storage reserve through a pumping system whenever the soil moisture sensor detects that the soil water level is below the set threshold. The IoT system developed is a fully functional prototype consisting of a soil moisture sensor, a relay module used to control the on and off switch of the water pump, a water pump and a temperature sensor to predict when to expect yields. The temperature readings are transmitted over a wireless network to the system backend, where the data collected is analyzed and used for the yield prediction model. These will be represented to the end user, who is the everyday farmer, in a cross-platform application built using the kivy language framework. It allows them to make appropriate modifications to their respective farming calendars. The system aims at improving irrigation efficiency. Hardware testing is done to ensure that the system is fully operational.

KEY WORDS: microcontroller, threshold, sensor, prototype, wireless networks, model

# TABLE OF CONTENTS

Contents

[DECLARATION 2](#_Toc147229080)

[ABSTRACT 3](#_Toc147229081)

[TABLE OF CONTENTS 4](#_Toc147229082)

[CHAPTER ONE: INTRODUCTION 6](#_Toc147229083)

[1.1 BACKGROUND 6](#_Toc147229084)

[1.2 STATEMENT OF PROBLEM 6](#_Toc147229085)

[1.3 OBJECTIVES 7](#_Toc147229086)

[1.3.1 MAIN OBJECTIVE 7](#_Toc147229087)

[1.3.2 GENERAL OBJECTIVES 7](#_Toc147229088)

[1.3.3 RESEARCH QUESTIONS 7](#_Toc147229089)

[1.4 JUSTIFICATION 8](#_Toc147229090)

[1.5 SCOPE 8](#_Toc147229091)

[1.6 LIMITATIONS 8](#_Toc147229092)

[CHAPTER TWO: LITERATURE REVIEW 8](#_Toc147229093)

[2.1 INTRODUCTION 8](#_Toc147229094)

[2.2 CASE STUDIES 9](#_Toc147229095)

[2.2.1 CASE STUDY 1: e-KILIMO ACCELERATOR 9](#_Toc147229096)

[2.2.2 CASE STUDY 2: IoT DRIVEN IRRIGATION IN SOUTH AFRICA 9](#_Toc147229097)

[2.2.3 CASE STUDY 3: IoTree MONITORING COCOA FARMS IN GHANA 9](#_Toc147229098)

[2.2.4 CASE STUDY 4: PRECISION AGRICULTURE IN INDIA 10](#_Toc147229099)

[2.3 SUMMARY 11](#_Toc147229100)

[2.5 PROPOSED METHODOLOGY 11](#_Toc147229101)

[CHAPTER THREE: METHODOLOGY 11](#_Toc147229102)

[3.1 INTRODUCTION 11](#_Toc147229103)

[3.2 FACT-FINDING TECHNIQUES 11](#_Toc147229104)

[3.2.1 INTERVIEWS 11](#_Toc147229105)

[3.2.3 KAGGLE 11](#_Toc147229106)

[3.3 SOFTWARE DESIGN AND DEVELOPMENT PROCEDURES 11](#_Toc147229107)

[3.4 PRELIMINARY DATA PROCESSING AND ANALYSIS 14](#_Toc147229108)

[REFERENCES 15](#_Toc147229109)

[APPENDICES 16](#_Toc147229110)

# CHAPTER ONE: INTRODUCTION

## BACKGROUND

Agriculture is ranked among the top economic activities that contribute approximately 33 percent to our country’s gross domestic product (GDP) (Muraya Beth, 2017). One could argue that a major reason for this is because we are situated alongside the equator, and this affords us the luxury of having an all-year-round planting season. However, recently we have been faced with climatic changes due environmental degradation which has transformed the country into a mostly arid landscape with unpredictable rainfall periods(William Nderitu, 2012) The climatic changes have led to a change in our traditional weather patterns, altering planting and harvesting seasons and this has brought the need for better agricultural practices, such as conservation agriculture, using drought-resistant crops, precision agriculture, crop diversification and agroforestry (Ioannis Mylonas et al, 2020). In Kenya particularly, where most adversity in the industry is brought about by lack of sufficient water, incorporating irrigation has allowed the farmer to have a continuous agricultural calendar throughout the year. It also allows for reclaiming of land that has been degraded by harsh environmental conditions and turned arid. However, the practice itself does come with its own set of challenges. Water is an essential element for life itself and there is a need to conserve it. Traditional irrigation practices do not put this into consideration, as they only have the plants well-being in mind. Occurrences of over-irrigation, and the vice versa under-irrigation are common as farmers operate on an observational basis, and this in itself has been noted to do more harm than good. My system aims to eradicate this problem, by integrating an IoT system into the farm that allows for monitoring and regulation of the soil moisture level so as to achieve maximum efficiency and minimal wastefulness of this vital mineral.

## STATEMENT OF PROBLEM

There is an ever growing need to adopt modern technologies that will help increase agricultural productivity while simultaneously reducing resource wastefulness. Current agricultural practices are outdated and often lead to issues such as soil degradation and resource wastefulness **(\*\*insert reference\*\*)**. It is also a common occurrence for cases of over-watering or under-watering in the farm which are known to lead to reduced farm yields since crops need optimum water conditions. Monitoring of the irrigation systems can also be quite costly, as they usually require on-site maintenance and regulation. The altered farming calendar might also bring about uncertainties, as farmers fail to plan ahead of schedule. Through this system all the challenges faced by farmers will hopefully be addressed and resolved.

## OBJECTIVES

## MAIN OBJECTIVE

The primary objective of my project is to develop an automated IoT based irrigation system that allows for monitoring and determination of the soil moisture content and regulation of irrigation pumps in-real-time from a remote location.

## GENERAL OBJECTIVES

Other secondary objectives that I will achieve as I execute my project include:

1. Calibrate a soil moisture sensor to determine the soil moisture content in a fixed land portion.
2. Calibrate a temperature sensor to calculate the atmospheric temperature.
3. Train a ML model to predict when to expect yields based on daily temperature readings.
4. Build an easy-to-use dashboard, that allows for visualization and interpretation of the data collected from the sensors.
5. Extend dashboard functionality to include mobile alerts and notifications for end users.

My ‘iFarm’ project is guided by the following research questions, that provide a framework for investigating and addressing the problem:

1. How do you calibrate and validate a soil moisture sensor to determine the soil moisture of a given soil sample?
2. How do you calibrate and validate an ambient temperature sensor to determine the prevalent temperature conditions at a given time?
3. What is the best ML model to predict when to expect crop yields?
4. How can I best visualize data collected on the field for easy interpretation by the modern-day farmer?

## JUSTIFICATION

IoT systems are able to connect a range of devices, allowing for both machine to machine communication and machine to human communication, across great geographical distances at little to no cost. The world is witnessing an increased integration of IoT devices into our everyday routines, with **(\*\*amazon estimating a staggering 25billion connected devices by the year 2023(ref)\*\*)**. In Kenya, agriculture is known to be a major commercial activity, most especially in our rural areas. It is also slowly but surely creeping in to our urban cities, via the use of green-houses and kitchen gardens (Gichuki Alice, 2012). Recently we have witnessed massive technological advances worldwide. By employing precision agriculture practices, such as automated harvesting or even using drones to conduct spaying of fertilizers and pesticides on the farm, technology has been utilized to make the outdated farming practices more efficient **(\*\*insert ref of 4th industrial revolution in tech\*\*)**. Nevertheless, it is still quite cost inefficient for the everyday small-scale farmer to incorporate such technology into the daily operations of the farm. My project hopes to bridge the gap between advances in technology and the small-scale farmer by building a cost-effective system that does the same work efficiently.

## SCOPE

While irrigation and precision agriculture practices are advancing by the day with new technologies being developed in advanced countries such as China. My project will be focused mainly in Nyeri County in Kenya which is found in the African continent. Testing of the project will be conducted within my house and the school’s engineering lab as I am only building a prototype. The main crop that I aim to focus on is tomatoes since this a very popular horticultural crop in the country, commonly planted both in green-houses and the field. I have selected tomatoes of the **(\*\*which fast growing species\*\*)**, since it a quick growing variety with a maturity period of **(\*\*maturity period\*\*)**.

## LIMITATIONS

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 INTRODUCTION

This section will critically examine existing research, studies, and resources related to precise irrigation activities, carbon footprint measurement, tracking, and sustainable living. This section provides a foundation for understanding the current state of knowledge, highlighting key insights, methodologies, and gaps in the field.

## 2.2 CASE STUDIES

## 2.2.1 CASE STUDY 1: e-KILIMO ACCELERATOR

This case study focuses on the e-Kilimo platform in Kenya **[\*\*ref\*\*]**, which utilizes IoT sensors to provide smallholder farmers with real-time weather information, crop management advice, and market prices via mobile phones. This technology has helped farmers make informed decisions, optimize planting and harvesting times, and improve crop yields

However, shortcoming of the project include:

* Limited Internet Access: In some rural areas, farmers may have limited access to the internet or smartphones, which can restrict their ability to fully utilize the e-Kilimo platform**(\*\*insert ref\*\*)**.
* Maintenance and Technical Support: Ensuring that IoT devices and the platform remain functional and up-to-date can be challenging, especially in remote regions.

## 2.2.2 CASE STUDY 2: IoT DRIVEN IRRIGATION IN SOUTH AFRICA

This case study explores an irrigation project in South Africa that uses IoT sensors and weather data to optimize water usage for crop irrigation. By providing real-time information to farmers, the project has led to significant water savings, increased crop yields, and improved food security.

However, shortcomings of the project are that:

* Initial Investment Costs: Implementing IoT-driven irrigation systems can be expensive, which may pose a barrier to adoption for small-scale farmers.
* Technical Skills: Farmers need to acquire the technical skills to operate and maintain the IoT systems, which may be a challenge for some.

## 2.2.3 CASE STUDY 3: IoTree MONITORING COCOA FARMS IN GHANA

IoTree is an IoT-based project in Ghana that focuses on monitoring cocoa farms. The case study describes how IoT sensors are used to track soil conditions, weather patterns, and pest infestations. This technology has helped cocoa farmers better manage their crops and increase yields. However, shortcomings of the project are that:

* Accessibility of IoT Devices: Not all cocoa farmers may have access to IoT devices, limiting the reach of such initiatives.
* Data Privacy Concerns: Collecting data from individual farmers may raise concerns about data privacy and security.

## 2.2.4 CASE STUDY 4: PRECISION AGRICULTURE IN INDIA

Precision agriculture, also known as "smart farming," is gaining momentum in India as a means to boost crop yields and resource efficiency. The "SmartFarm India" project, implemented in Punjab, aimed to leverage IoT technology and data analytics for precision agriculture.

However, shortcomings of the project are that:

* High Initial Costs: The installation of IoT sensors and the development of the data analytics infrastructure required a significant upfront investment. This cost may be prohibitive for many small-scale farmers in India.
* Limited Connectivity: In rural areas of India, access to reliable internet connectivity can be sporadic. This limitation affects the real-time transmission of data from the IoT sensors to the analytics platform.
* Technical Expertise: Effective utilization of precision agriculture technologies demands a certain level of technical knowledge and skills. Many farmers in rural India may lack the necessary training to operate and maintain these systems effectively.
* Data Privacy Concerns: Farmers expressed concerns about the collection and management of their data. There were worries about data security, ownership, and how the information would be used.
* Resource Availability: Implementing precision agriculture often requires specific resources such as high-quality fertilizers, pesticides, and irrigation equipment. Some farmers may struggle to afford these resources.
* Customization for Crop Variability: Precision agriculture systems may need to be tailored to specific crops and soil types. A one-size-fits-all approach may not be suitable for the diversity of crops grown in India.

## 2.3 SUMMARY

According to the case studies described above there are already existing solutions to help farmers automate irrigation and monitoring of their farm. However, it is evident that each comes with its own share or challenges associated with it. Some solutions are expensive to local farmers, in terms of both installation and maintenance, while other solutions are not available in Kenya. Some solutions are only available for one plant only, while others seem to require some technical expertise to operate efficiently. Most of the solutions mentioned also seem to face data privacy concerns.

## 2.4 RESEARCH GAP

This project aims to provide an automated irrigation system that is inexpensive, efficient and affordable to our local farmers. Operating and maintaining the system will require little to no technical expertise. The system also integrates a machine learning model that predicts when one should expect crop yields. Data collected is transmitted over the internet via wi-fi, to an application on the farmer’s phone, hence alleviating all privacy concerns.

## 2.5 PROPOSED METHODOLOGY

Based on the insights gained from the literature review, the ‘iFarm’ project proposes a methodology that aligns with the best practices and successful approaches identified in existing research. This methodology will guide development, testing and evaluation of the system ensuring its effectiveness in promoting precision agriculture practices in farming.

# CHAPTER THREE: METHODOLOGY

## 3.1 INTRODUCTION

This section will explain the methods that I will employ to achieve the objectives of my project and also elaborates why I have chosen them in preference to others. The system development methodology that will be used to develop the system will be described here. The techniques will enable in collection of facts and data that will be used as inputs in understanding what the system will do.

## 3.2 DATA ACQUISITION

Data acquisition is the process of collection raw data and information based on a number of given techniques. These could either be primary or secondary data sources. Primary data sources provide information that did not exist before, and include techniques such as observation, questionnaires and interviews. Secondary data sources provide data that has already been collected and analyzed by somebody else, and include techniques such as sampling of existing documents, research and browsing the internet. The methods that I utilized have been discussed below:

## 3.2.1 INTERVIEW

To help support the problem statement for this research, the researcher conducted an interview with the head of Horticultural management in Dedan Kimathi University of Science and Technology, Mr. Isaac Muiga. The interview session helped me to identify the most popular plants grown in Nyeri county and the one best suited to grow to maturity within my allocated time. We concluded to use a tomato fruit of the species **(\*\*insert species\*\*)** as it has a fast maturity period. Although we had planned to grow the plant and test the system on it out on the field, under the guidance of my panel, we opted to instead plant it as a bucket experiment for the purpose of presentation. The interview also help me to understand the impact of poor farming practices in general, to both the soil and water sources.

## 3.2.3 KAGGLE

It is an open-source marketplace for datasets with free access **(\*\*insert ref\*\*).** Datasets range from health, agriculture, technology and economy. The researcher aims to train and test the model from this source. I found a dataset of daily temperature readings **(\*\*insert ref\*\*)**, and used it to build my model.

## 3.3 SYSTEM DESIGN

## 3.3.1 HARDWARE REQUIREMENTS

### 1. ARDUINO UNO

The Arduino Uno microcontroller is a widely embraced and user-friendly platform renowned for its accessibility and versatility in DIY electronics projects. Powered by an Atmel ATMega328P microcontroller running at a clock speed of 16 MHz, the Uno features 2 KB of SRAM, making it well-suited for a broad range of applications. With 14 digital input/output pins and 6 analog inputs, it provides ample connectivity for sensors, actuators, and peripherals. Programmed in a simplified version of C/C++ via the Arduino IDE, the Uno is particularly popular among beginners due to its ease of use, extensive online resources, and an active community. Its widespread adoption can be attributed to its open-source nature, allowing users to modify and share designs freely. Whether used for educational purposes, prototyping, or hobbyist projects, the Arduino Uno's combination of accessible programming, diverse connectivity options, and community support cements its status as a go-to microcontroller for countless electronic enthusiasts.

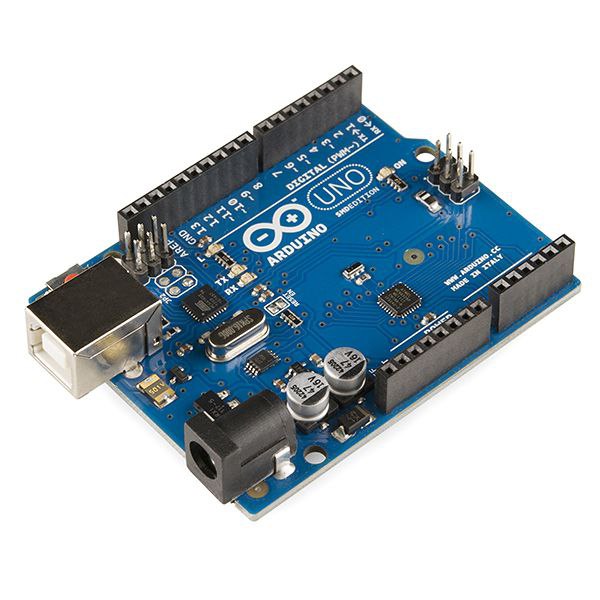


Figure 1Arduino Uno Microcontroller

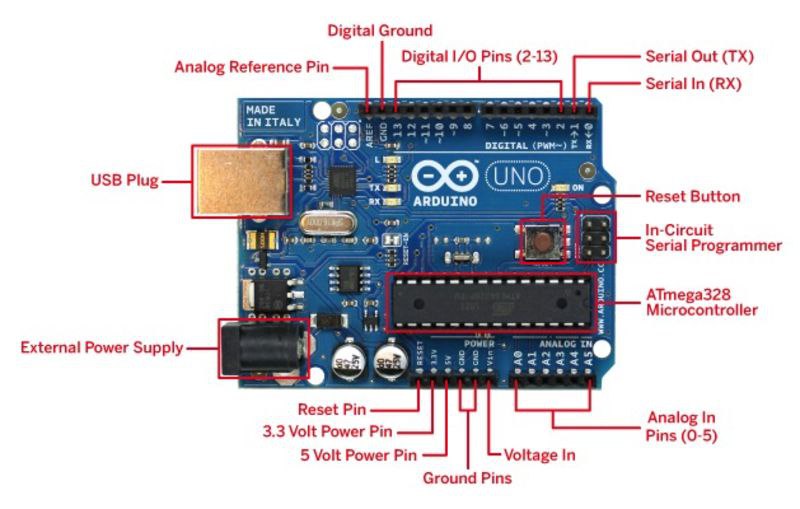


Figure 2Labelled Components of an Arduino Uno microcontroller

### 2. ESP8266

The ESP8266 microcontroller, developed by Espressif Systems, is a powerful and compact device known for its embedded Wi-Fi capabilities. Running on a Tensilica L106 32-bit microcontroller with a clock speed of 80 MHz, the ESP8266 offers impressive performance. It typically features 80 KB of RAM, providing sufficient memory for a variety of IoT and connectivity applications. With a focus on wireless communication, the ESP8266 includes integrated Wi-Fi functionality, making it suitable for projects requiring internet connectivity. It boasts a variable number of general-purpose input/output (GPIO) pins, often around 9 to 17 pins, depending on the specific module or development board.

Programming for the ESP8266 is commonly done using the Arduino IDE, though it also supports the Espressif IoT Development Framework (ESP-IDF) for more advanced users. Its popularity stems from its affordability, compact size, and the ease with which it enables IoT projects. The ESP8266 has become a go-to choice for home automation, sensor networks, and other applications where seamless wireless communication is essential. The strong community support, rich documentation, and the availability of numerous third-party libraries contribute to the widespread adoption of the ESP8266 in the maker and IoT communities.

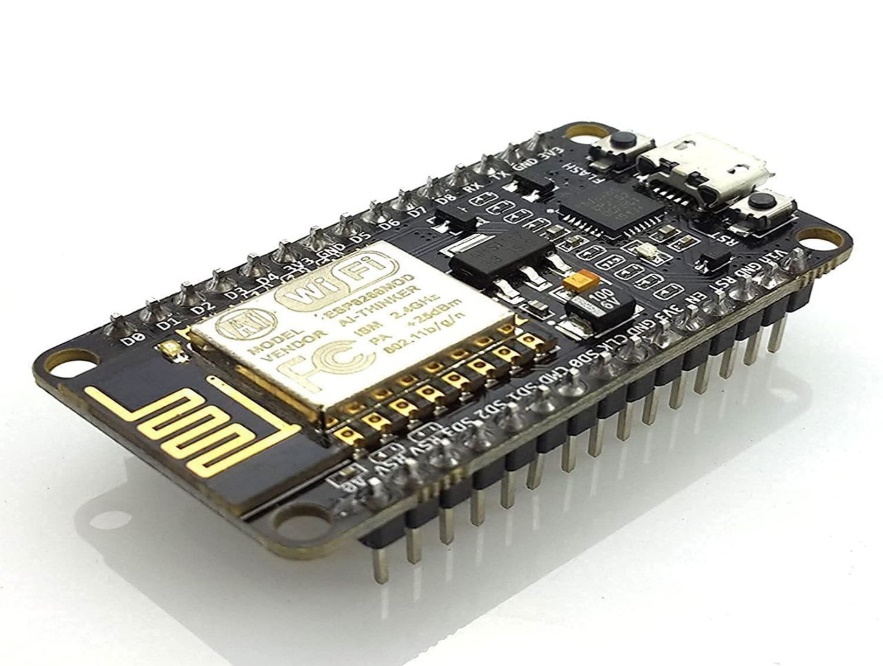


Figure 3 ESP8266 microcontroller

### 3. DHT11 SENSOR

The DHT11 temperature sensor is a widely-used and cost-effective sensor for measuring temperature and humidity in electronic projects. Featuring a simple design, it incorporates a humidity and temperature sensor, as well as a signal conditioning and analog-to-digital conversion circuit into a single package. The DHT11 is known for its ease of use and compatibility with various microcontrollers.

The sensor operates within a temperature range of 0 to 50 degrees Celsius with a humidity range of 20% to 90%. It communicates over a one-wire digital interface, simplifying the connection to microcontrollers. Programmatically, the DHT11 sensor can be interfaced with microcontrollers like Arduino through dedicated libraries. Code snippets in languages like C or Python allow users to read temperature and humidity values from the sensor easily. The DHT11's simplicity, low cost, and compatibility contribute to its popularity in various applications.

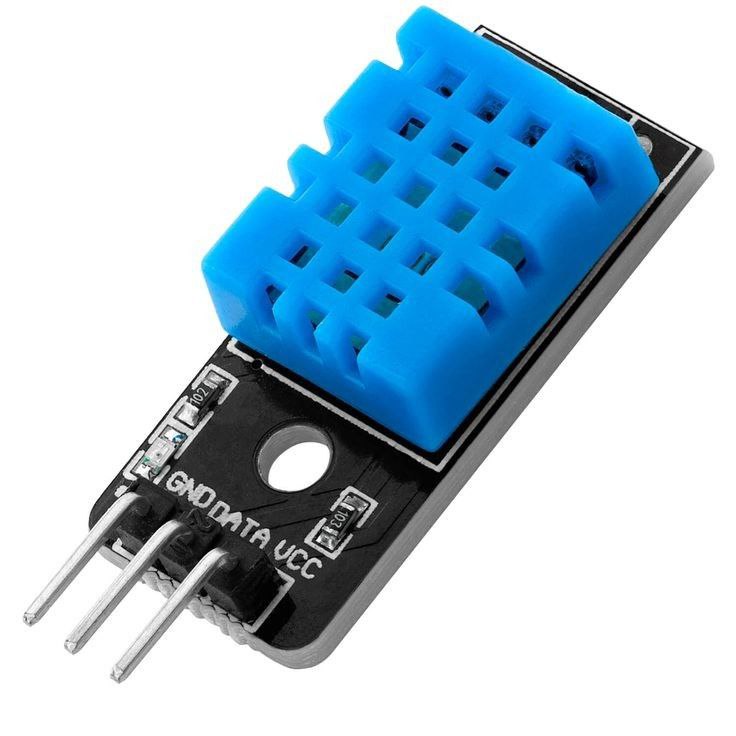


Figure 4 DHT11 Sensor

### 4. SOIL MOISTURE SENSOR

The resistive soil moisture sensor, exemplified by the FC-28, serves as a pivotal tool in agriculture and environmental monitoring by gauging the moisture content of soil. Featuring a pair of exposed electrodes, this sensor measures soil resistance changes, providing an analog or digital output indicative of soil moisture levels. Compatible with popular microcontrollers like Arduino or Raspberry Pi, it enables seamless integration into various projects. Programmatic interfacing involves utilizing analog or digital pins, with libraries readily available for simplified coding. Widely embraced for its affordability, user-friendly design, and adaptability, the resistive soil moisture sensor finds applications in precision agriculture, enabling efficient irrigation management and fostering optimal plant growth conditions in both professional and DIY contexts.

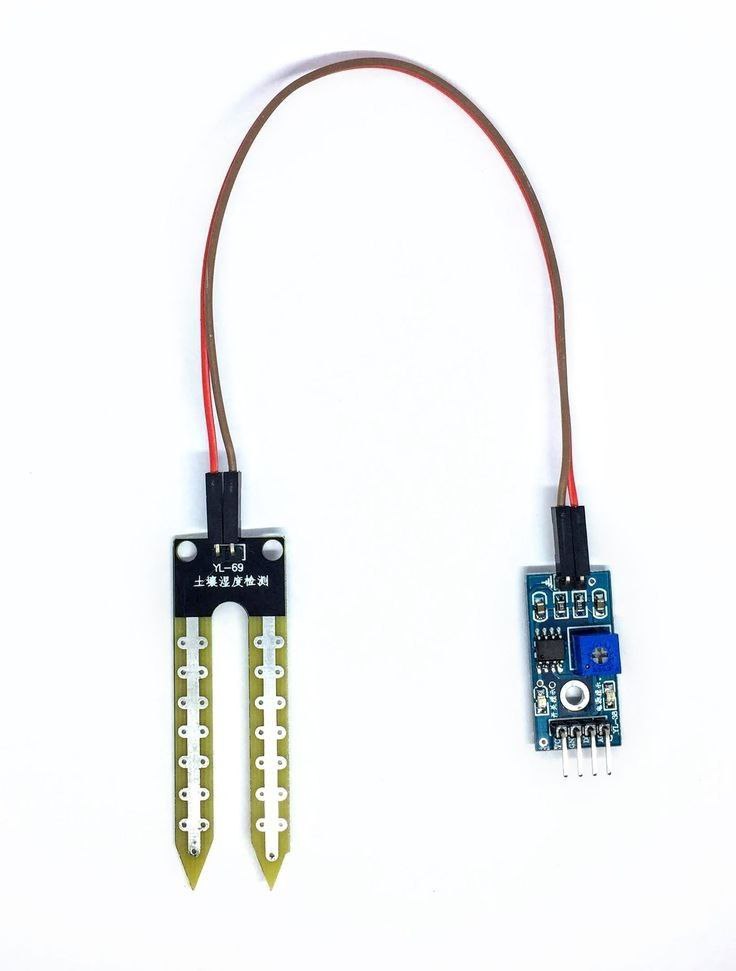


Figure 5 Resistive Soil Moisture Sensor

### 5. GROVE RELAY MODULE

The 2-Way Grove Relay is a highly popular electronic component known for its versatile applications in automation and control systems. Equipped with two channels and typically requiring two GPIO pins for interfacing, it allows users to control two separate electrical circuits independently. This relay module is widely embraced for its simplicity and ease of integration into projects, making it accessible for both beginners and experienced enthusiasts. Its popularity can be attributed to the convenience it offers in managing diverse devices, such as lights, motors, or other electrical appliances, through a microcontroller like Arduino. The 2-Way Grove Relay's ability to facilitate bidirectional communication and control. It is a preferred choice for a broad spectrum of electronic projects requiring reliable and efficient relay functionality.

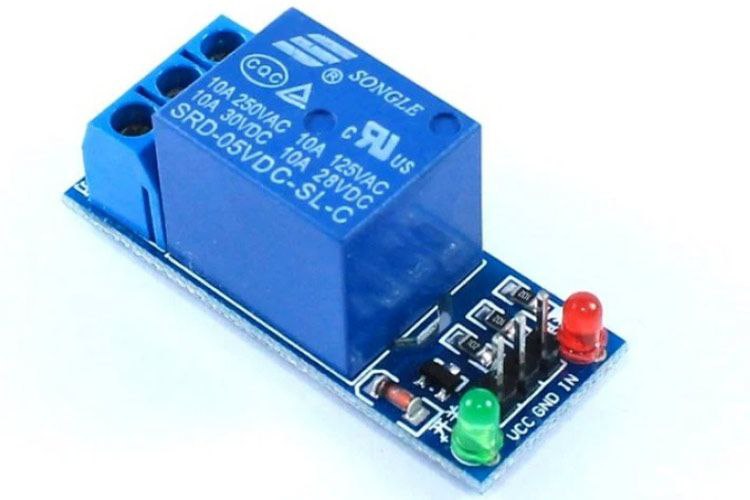


Figure 6 Grove Relay

### 6. WATER PUMP

A 5V water pump is a compact and efficient device designed for pumping water in various electronic projects. Operating at a low voltage of 5 volts, it is often powered by a USB connection or a dedicated power source. These pumps are commonly used in small-scale applications, such as desktop fountains, aquariums, or irrigation systems for potted plants. With a compact form factor and straightforward design, the 5V water pump is easy to integrate into DIY projects, providing a reliable solution for circulating or transporting water. Its low voltage requirement makes it compatible with microcontrollers like Arduino, adding to its versatility and popularity in electronics and hobbyist communities.



Figure 75V Water Pump

Table 1: How I interfaced my components

|  |  |
| --- | --- |
| **COMPONENTS** | **CONNECTION TO THE ESP8266 OR ARDUINO UNO** |
| DHT11 sensor | Power(4-20v) - 5v pin on the Arduino Uno  Out - Analog Pin 1(A1) on the Arduino Uno  Ground - GND pin on the Arduino Uno  **(\*\*confirm on buying\*\*)** |
| Soil moisture sensor | VCC - 5V pin of Arduino board  GND - GND pin on Arduino board  DO -  AO - |
| Relay module |  |
| Water pump |  |

## 3.3.2 SOFTWARE REQUIREMENTS

### 1. PYTHON PROGRAMMING SKILLS

Python is a very popular programming language, mostly especially due to its very large community and its rather easy syntax. It was created by Guido Van Rossum in 1991 and today its used in a multitude of different application areas, ranging from web development, game development, machine learning, artificial intelligence, data science and data visualization, web scrapping among others. In this project, I employed python to build the GUI(graphical use interface) for the dashboard using the kivy language framework.

### 2. SUBLIME TEXT

Sublime Text is a sophisticated and widely-used text editor appreciated for its speed, versatility, and minimalist design. Available on multiple platforms, it offers a distraction-free coding environment with a highly responsive interface. Sublime Text supports various programming languages, providing syntax highlighting, auto-completion, and an extensive range of plugins that enhance functionality. It has become a preferred choice for developers seeking a powerful and efficient text editor for coding and scripting tasks.

### 3. ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a user-friendly software application designed for programming and uploading code to Arduino microcontrollers. It simplifies the process of creating and managing code for Arduino-based projects, catering to users of all skill levels. The IDE supports the Arduino programming language, which is a simplified version of C/C++. It features a straightforward interface with a text editor for writing code, a toolbar for uploading sketches to Arduino boards, and a serial monitor for debugging and communication. The Arduino IDE includes a vast library of pre-built functions and examples, making it accessible for beginners while offering advanced features for experienced developers. It has been widely adopted for its ease of use and robust functionality.

### 4. KIVY FRAMEWORK

Kivy is an open-source Python framework for developing multi-touch applications. It is specifically designed to facilitate the creation of cross-platform applications with a natural user interface, making it suitable for desktop, mobile, and other touch-enabled devices. One distinctive feature of Kivy is its own declarative language, also known as the Kivy Language (KV). KV language allows developers to design user interfaces by describing the layout and behavior of UI elements in a concise and readable manner, separate from the application's Python code. This approach enhances code readability, promotes efficient collaboration between designers and developers, and facilitates rapid prototyping.

### 5. WIRELESS COMMUNICATION PROTOCOLS

## 3.3.3 SYSTEM ARCHITECTURE

**(\*\*show diagram of how everything is interconnected\*Uliza jere venye utachora hii diagram\*)**

## 3.4 JUSTIFICATION FOR METHODOLOGY

Evolutionary prototyping is a development approach that involves building an initial, basic version of a system and gradually refining it through iterations based on feedback and insights gained during each phase. This method is particularly justified in scenarios where project requirements are not well-defined or are subject to frequent changes. By allowing for continuous refinement and adaptation, evolutionary prototyping accommodates evolving project needs and requirements, reducing the risk of developing a solution that might become outdated or incompatible with user expectations. It promotes a flexible and responsive development process, encourages stakeholder involvement, and enables the incorporation of valuable insights gained during the prototyping stages. This iterative approach is well-suited for projects where the end goals are not fully clear initially, providing the flexibility needed to refine and enhance the system incrementally in response to user feedback and changing requirements.

## 3.3 SOFTWARE DESIGN AND DEVELOPMENT PROCEDURES

The system development methodology for the project is discussed here. It walks you through the rationale of the adopted system. If a proper technique is adopted, a good system will be built. I chose evolutionary prototyping and agile method of software development.

1. Project Planning and Preparation:

Define Project Objectives: Clearly articulate the goals and objectives of the IoT-based agricultural project, including the specific outcomes to be achieved.

Stakeholder Engagement: Identify key stakeholders, including farmers, agricultural experts, and technology partners. Establish communication channels for collaboration and feedback.

1. Needs Assessment:

Farm Surveys: Conduct surveys and interviews with local farmers to understand their needs, challenges, and priorities.

Environmental Analysis: Assess local climate conditions, soil types, and crop varieties to tailor IoT solutions to specific agricultural contexts.

1. Technology Selection and Design:

IoT Component Selection: Choose appropriate IoT sensors, actuators, and data communication technologies based on the project's objectives and the local environment.

Hardware Prototyping: Develop prototypes of IoT devices and systems for testing and validation.

4. Sensor Deployment and Data Collection:

Installation: Deploy IoT sensors for monitoring soil conditions (moisture, NPK), weather data (temperature).

Data Collection: Collect real-time data from deployed sensors and ensure reliable data transmission to a central database.

5. Data Analysis and Machine Learning:

Data Processing: Clean and preprocess the collected data to ensure accuracy and consistency.

Machine Learning Model Development: Train machine learning models using historical data for predicting crop yields based on factors like temperature, soil conditions, and irrigation patterns.

6. Automated Irrigation System:

Design and Implementation: Develop an automated irrigation system that uses IoT sensor data to determine optimal irrigation schedules.

Testing: Validate the system's performance by monitoring crop growth, soil moisture, and water consumption.

7. Soil Fertility Monitoring:

IoT Sensor Integration: Implement soil sensors to monitor soil moisture levels and NPK.

Fertilization Recommendations: Develop algorithms to provide farmers with precise recommendations for fertilization based on real-time soil data.

8. User Interface Development:

Dashboard Creation: Design a user-friendly dashboard accessible via web or mobile app for farmers to monitor and control the IoT-based system.

Training and Education: Provide training to farmers on how to use the system effectively.

9. Field Testing and Evaluation:

- Real-world Deployment: Install the complete IoT-based agricultural system on selected farms.

- Performance Monitoring: Continuously monitor the system's performance, gather feedback from farmers, and make necessary adjustments.

10. Data Privacy and Security:

- Implement robust data privacy and security measures to protect farmer data and ensure compliance with relevant regulations.

11. Documentation and Reporting:

- Document the project's progress, outcomes, and challenges at each stage.

- Prepare regular reports and share findings with stakeholders.

12. Scale-up and Outreach:

- Based on the project's success, consider scaling up the IoT-based agricultural system to benefit a larger farming community.

- Conduct awareness campaigns and workshops to encourage adoption among local farmers.

13. Project Evaluation and Future Enhancements:

- Evaluate the project's impact on agricultural productivity, resource efficiency, and farmer livelihoods.

- Identify areas for improvement and potential enhancements for future phases.

# CHAPTER 4: SYSTEM ANALYSIS AND DESIGN

## 4.1 INTRODUCTION

System analysis involves the studying of a system by examining its component and their interactions while design refers to the process of defining the architecture, components, and data of a system to satisfy specified requirements (Tilley &Rosenblatt, 2016). System analysis and design is the process that software engineering companies use to evaluate some environmental variables either macro or micro in the business and design and develop

optimal ways to counter or solve the problems caused by the business environmental variables.

System analysis is a very crucial task performed during the development of a

particular system as the designers and developers have clear insights about the problem to be

solved and also gather more information about the proposed system. Proper and appropriate

fact-finding techniques are used in the data collection and analysis that would assist in the

development of a successful product.

The design is also a key element in the process. The designer must develop a design,

showing the key objectives to be done by the system. The design plans must be acceptable by

the stakeholders or users of the system. The design should include all the affordance

properties to assist the users of the system understand it better in terms of user interface and

functionality.

## 4.2 FEASIBILITY ANALYSIS

This is a detailed analysis of a project which supports processes of decision-making by scrutinous examination of its strengths, weaknesses, opportunities and threats, while also providing information about necessary resources and chances to succeed. It is not a methodology but rather a term that describes action to evaluate at the early stages of planning whether a project should proceed, be redesigned or abandoned. From the market research and general environmental analysis conducted, the system was rendered viable as the benefits outweighed the demerits.

### 4.2.1 ECONOMIC FEASIBILITY

Here we take a look at the ability of a project to generate enough revenue to cover its costs and provide a reasonable return on investment. One analyzes cost and benefits of a project, including cost of materials, labour and equipment, then compares it to the projected sales from the revenue to other sources of income. From the market resaerch and general environmental analysis conducted, the system was rendered viable due to a large return on investment.

### 4.2.3 TECHNICAL FEASIBILITY

This involves researching the technical requirements of a project, such as software and hardware needed. You will evaluate the prospective implementation and development of a proposed project while considering the available resources, technology and overall capabilities of the primary team. The technology used in the development was compatible with the smartphone of the farmers hence it was considered to be technically feasible.

### 4.2.3 OPERATIONAL FEASIBILITY

This is basically how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition. The system being built was in a position to optimally solve problems identified, hence considered operationally feasible.

## 4.3 REQUIREMENT ANALYSIS

This is the process of determining user expectations for a new or modified product. It is an essential part of software development that helps the team to undersatnd, finalize and document features and functionalities required of the end product. Requirement analysis is majorly categorized into either functional and non-functional requirements.

### 4.3.1 FUNCTIONAL REQUIREMENTS

These define specific behaviors or functions of a system. Basically they entail what a system is required to do so as to meet all of its objectives. They are also the requirements that the user wants to be in the system so as to facilitate its frequent and daily usage. In our system, they were:

1. Users can login and register into the system.
2. Users can see the soil moisture content of their farm.
3. Users can see the ambient temperature on their farm.
4. Users can regulate the pump status to be either on or off.
5. Users can see the growing stage of their crop.

### 4.3.2 NON-FUNCTIONAL REQUIREMENTS

These specify criteria that can be used to judge the operation of a system rather than specific behaviors. They do not tell what the system does, but instead how well it does it. In our system, they include:

1. Open source – source code is made freely available for possible modification or redistribution.
2. Transparency – it operates in such a way that it is easy for others to see what actions are being performed.
3. Efficiency – this is a systems capability to avoid making mistakes or wasting resources while performing a certain task.
4. Durability – this is the ability of a physical product to remain functional without requiring excessive maintenance or repair when faced with challenges.
5. Documentation – includes communicable materials used to describe, explain or instruct regarding some attributes of a system such as its assembly, installation or maintenance.
6. Data Integrity – The maintenance and the assurance of data accuracy and consistency over its entire life cycle.

## 4.4 DATA ANALYSIS

## 4.5 UML DIAGRAMS

### 4.5.1 USE-CASE DIAGRAM

### 4.5.2 ENTITY RELATONSHIP DIAGRAM

### 4.5.3 DATA FLOW DIAGRAM

### 4.5.4 ACTIVITY DIAGRAM (FLOW CHART)

# REFERENCES

1. Muraya, Beth W. *Determinants of agricultural productivity in Kenya*. Diss. 2017.
2. [www.statista.com](https://www.statista.com/statistics/451143/share-of-economic-sectors-in-the-gdp-in-kenya/" \l ":~:text=This statistic shows the share,sector contributed about 55.06 percent.)
3. Maina, William Nderitu, and Fr Mathenge Paul Maina. "Youth engagement in agriculture in Kenya: Challenges and prospects." *Update* 2 (2012).
4. Mylonas, Ioannis, et al. "Better farming practices to combat climate change." *Climate change and food security with emphasis on wheat*. Academic Press, 2020. 1-29.
5. Pierce, Francis J., and Peter Nowak. "Aspects of precision agriculture." *Advances in agronomy* 67 (1999): 1-85.
6. Gichuki, Alice W. *Factors influencing adoption of greenhouse technology: case of Gatitu sub-location in Nyeri Central district, Nyeri county*. Diss. University of Nairobi, Kenya, 2012.
7. [e-Kilimo Accelerator (saharaventures.com)](https://ekilimo.saharaventures.com/)

# APPENDICES

Schedule table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DURATION | October | November | December | January | February | March |
| TASK |
| FACT-FINDING |  |  |  |  |  |  |
| ANALYSIS |  |  |  |  |  |  |
| DESIGN |  |  |  |  |  |  |
| DEVELOPMENT |  |  |  |  |  |  |
| DEPLOYMENT |  |  |  |  |  |  |