EaSibug: Design and Development of a Smart Watering System Using IoT

Adrian Paul A. Guillermo[1], Jamaica Lhian C. Catabui[1], Angelica M. Agamata[1],

Segundo D Pacris, Jr.[2]

[1]Student Researcher, College of Information and Computing Sciences, Cagayan State University, Sanchez Mira Campus
[2]Adviser, Faculty Member, College of Information and Computing Sciences, Cagayan State University, Sanchez Mira Campus

ABSTRACT

This study explored the design and development of a smart watering system named "EaSibug," a combination of the English word "Easy" and the Ilocano word "Sibug," which means "Watering." Together, the name signifies "Ease to water the plants." The system was developed using IoT to automate plant irrigation efficiently without human intervention. The project integrated IoT principles, soil moisture sensors, and automation technology to address inefficiencies in traditional watering methods. EaSibug monitored soil moisture levels and activated a water pump only when necessary, ensuring optimal plant hydration. The study employed a descriptive research design to analyze the system's performance, including hardware and software integration, testing, and deployment. The results demonstrated the effectiveness of EaSibug in providing an adaptable and user-friendly solution for small-scale gardening.
Keywords: Smart Watering System, Arduino, IoT, Soil Moisture Sensor, Automated Irrigation, Water Conservation, Small-Scale Gardening, Home Automation, ThingSpeak, Microcontroller, Environmental Monitoring, Smart Agriculture, Remote Monitoring, Sustainable Gardening, Sensor Technology

INTRODUCTION

Watering plants manually was time-consuming and inefficient, often resulting in overwatering or under watering. Automated irrigation systems provided a solution to these challenges. This study presented the development of a smart watering system utilizing Arduino technology, integrating sensors and IoT for efficient irrigation.

The primary aim of this study was to develop an automated watering system that enhanced irrigation efficiency, conserved water, and minimized human intervention. The system used soil moisture sensors to regulate water distribution based on real-time data.

This study answered essential questions regarding the components and requirements for developing a smart watering system using IoT, its effectiveness in ensuring efficient water usage, and the advantages of automation in plant care compared to traditional watering methods.

The scope of this study focused on small-scale gardening applications, including hardware and software development, system testing, and validation. However, large-scale agricultural irrigation and economic viability assessments were beyond the study's scope.

The significance of this study lay in its potential benefits. Gardeners gained an efficient solution for plant care, plant enthusiasts automated their watering schedules, busy individuals saved time and effort, researchers further explored IoT-based smart agriculture, and future developers used this study as a foundation for further enhancements.

Objectives

The study designed and developed an Arduino-based smart watering system, integrated soil moisture sensors for automated irrigation, tested and evaluated the efficiency of the developed system, and explored IoT applications in smart agriculture.

Technical Background

Current Technologies Used in Smart Watering Systems

The current technologies used in smart watering systems encompass various hardware, software, and network components. These technologies enable automation, monitoring, and control of watering processes. Some commonly used technologies include:

Hardware

Smart watering systems employ hardware components such as microcontrollers, sensors, actuators, and communication modules. Microcontrollers, particularly Arduino boards, are popular due to their versatility and ease of use. Sensors like soil moisture sensors, temperature sensors, and light sensors are utilized to gather environmental data. Actuators, such as solenoid valves or pumps, are responsible for regulating water flow. Communication modules facilitate connectivity between the system components and external devices.

Figure 1. EaSibug the Product

1. NodeMCU ESP8266

Figure 2. ESP8266

-NodeMCU is an open-source firmware and development kit that helps you prototype your IoT (Internet of Things) projects based on the ESP8266 WiFi module, which you can program using the Arduino IDE. The ESP8266 is a low-cost WiFi microchip with full TCP/IP stack and microcontroller capabilities. NodeMCU provides a simple and consistent programming interface for the ESP8266, making it easier to develop IoT applications.2. NodeMCU V1.0 Adapter Board

Figure 3. Adapter Board

-The NodeMCU V1.0 adapter board serves as a convenient and versatile development platform for projects utilizing the ESP8266 WiFi module. Featuring a

USB-to-serial converter (typically CH340 or CP2102), it facilitates easy programming and communication between the ESP8266 and a computer. The board includes General Purpose Input/Output (GPIO) pins for interfacing with various electronic components, a voltage regulator for stable power supply, and support for both USB and external power sources.

3. Solar Controller

Figure 4. Solar Controller

-A Solar Controller is used to sustain the charge of the battery I used for both the NodeMCU and the water pump, this Solar Controller can use 12V or 24V battery.

4. 12V Battery

Figure 5. Battery

- A 12V battery is used to sustain the voltage input to the NodeMCU microcontroller and to power the current water pump used in this project.

5. Water Pump 12VDC

Figure 6. Water Pump

- A water pump with a 12VDC required power and 110 PSI pressure is used in this project, but you can use any water pumps based on what you will use it from.

6. Power Switch

Figure 7. Switch

-This power switch is used to cut off the power from the battery to the Solar Controller.

7. Step-Down Power Controller

Figure 8. Step Down Power Controller

-Step-Down Power Controller is used to manage the voltage being used from the battery from 12V to 5V to power the NodeMCU board.

8. LCD Display

Figure 9. LCD Display

-This is the LiquidCrystal_I2C with a 16x2 I2C LCD display. The display provides information about the initialization process, WiFi connection status, soil moisture level, and other relevant details. You can use a larger I2C LCD display if you want or even better a touchscreen OLED display for additional accessibility of the device.

9. Soil Moisture Sensor

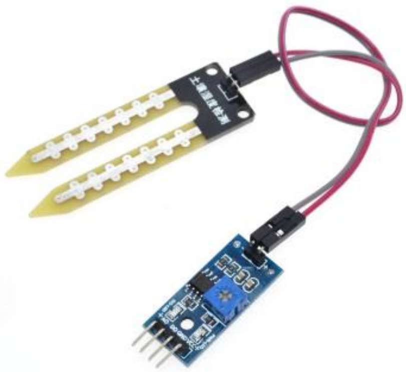

Figure 10. Soil Moisture Sensor

- In this project, a soil moisture sensor connected to the NodeMCU V1.0 board measured soil moisture, with the analog reading mapped to a percentage. The result was displayed on a 16x2 I2C LCD screen. Depending on moisture levels, the NodeMCU controlled a relay for a watering system, and the data was sent to ThingSpeak for remote monitoring and logging.

10. Relay Module

Figure 11. Relay Module

- In the project, a relay module was employed for controlling a watering system based on soil moisture levels. Connected to a digital pin (D3) on the NodeMCU V1.0 adapter board, the relay module allowed the NodeMCU to either enable or disable the water supply to the plants. The decision to activate or deactivate the relay was contingent upon the soil moisture percentage measured by the soil moisture sensor. This relay control mechanism added an automated watering feature to the system, enhancing its functionality for efficient plant care.

Software

Software plays a crucial role in smart watering systems for data processing, control logic, and user interfaces. Arduino IDE (Integrated Development Environment) is commonly used for programming Arduino boards. Other software tools, libraries, and frameworks may be utilized for specific functionalities, such as data analysis, visualization, and remote control.

Libraries: The project makes use of several libraries, including:

1. ESP8266WiFi: The ESP8266WiFi library is a key component in the Arduino IDE, providing functions and tools necessary for the NodeMCU to connect to and communicate with WiFi networks, enabling internet connectivity for the project.

2. Wire: The Wire library is used in the Arduino IDE for I2C communication, enabling the NodeMCU to communicate with the LiquidCrystal_I2C LCD screen, simplifying the display setup in the project.

3. ThingSpeak: This library facilitates communication with the ThingSpeak platform, allowing you to send data to the cloud for storage and analysis.

ThingSpeak Integration: The ThingSpeak platform is used to store and visualize the soil moisture data. The ThingSpeak.writeField function is used to send the moisture percentage to a specific channel on ThingSpeak.

Network

Smart watering systems can leverage various network technologies for connectivity and data exchange. This includes Wi-Fi, Bluetooth, or even Internet of Things (IoT) protocols like MQTT (Message Queuing Telemetry Transport) or CoAP (Constrained Application Protocol). Network connectivity enables remote monitoring and control of the watering system, allowing users to access and manage it through mobile applications or web interfaces.

Current Trends and Technologies in Smart Watering Systems

Advancements in technology have paved the way for innovative features and capabilities in smart watering systems. Some emerging trends and technologies include:

Internet of Things (IoT)

IoT plays a significant role in enhancing the capabilities of smart watering systems. It enables seamless connectivity, real-time data monitoring, and remote control through cloud-based platforms. IoT platforms provide centralized management and analytics, allowing for intelligent decision-making and optimization of water usage.

Data Analytics and Machine Learning

Data analytics and machine learning techniques can be employed to analyze the collected data from sensors and optimize watering schedules based on environmental conditions, plant needs, and water conservation goals. These technologies enable predictive modeling, anomaly detection, and adaptive control mechanisms.

Mobile Applications and Smart Home Integration

Mobile applications provide convenient interfaces for users to monitor their smart watering systems from anywhere. Integration with smart home platforms allows for seamless automation and synchronization with other smart devices, creating a cohesive and interconnected ecosystem.

ThingView Application

Figure 12. ThingView App

The ThingView app is a mobile application that enables users to connect to soil moisture sensors via Internet Connection using the WiFi capabilities of the NodeMCU Microcontroller. It provides real-time data on soil moisture levels in a visually appealing format, including graphs and charts. Users can track changes over time and set threshold alerts for timely action. The app simplifies the monitoring and analysis of soil moisture, aiding in informed decisions regarding irrigation and watering schedules for plants or crops. The app is connected to ThingSpeak that uses a specific Channel ID for it to connect.

METHODOLOGY

Research Design

The research adopted a descriptive research design to investigate the design and development of a smart watering system using IoT. This design allowed for a comprehensive understanding of the system's components, functionalities, and implementation process.

Project Workflow

The project workflow encompassed the systematic approach to designing and developing the smart watering system. It involves the following steps:

Requirements Analysis

In the requirements analysis phase, the needs and preferences of users, as well as the constraints of small gardens, were assessed. This phase involves observations to gather information and define the desired features and functionalities of the system.

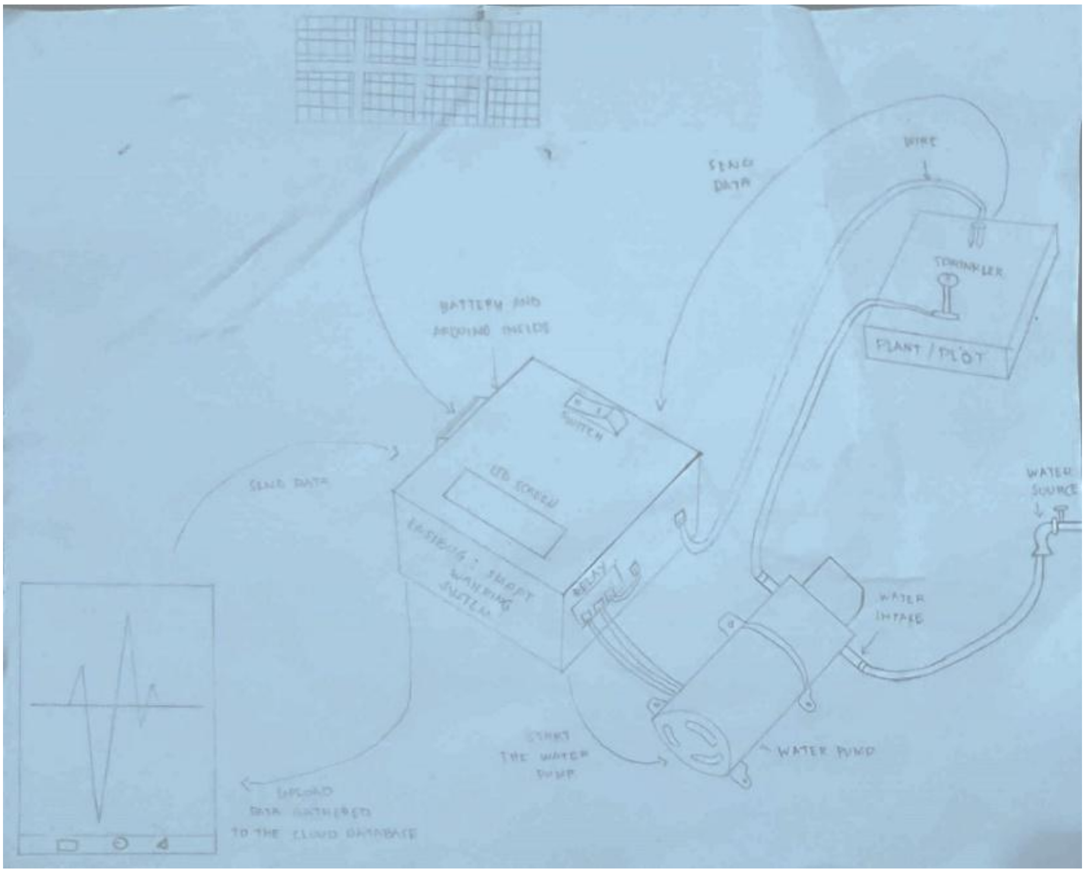

Figure 13. Drawing Plan

Hierarchical Diagram

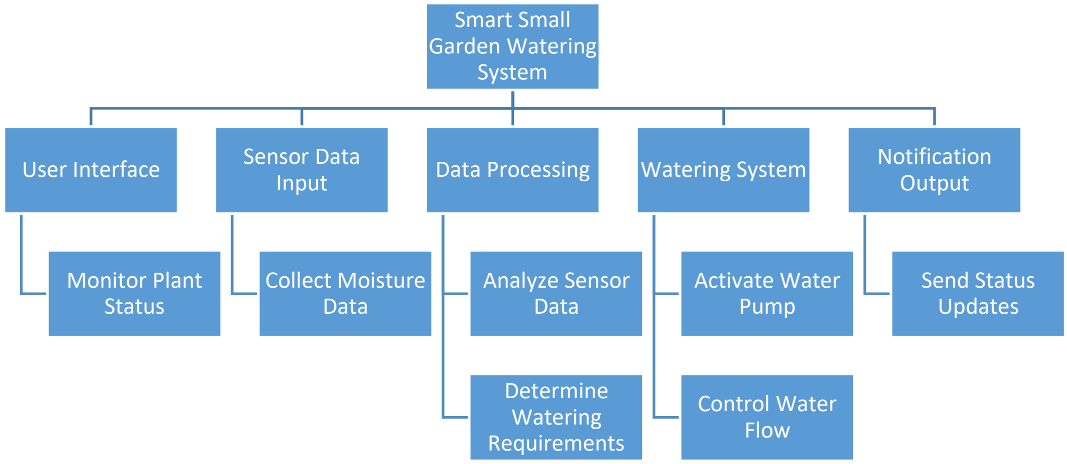

Figure 14. Hierarchical Diagram

Use Case Diagram

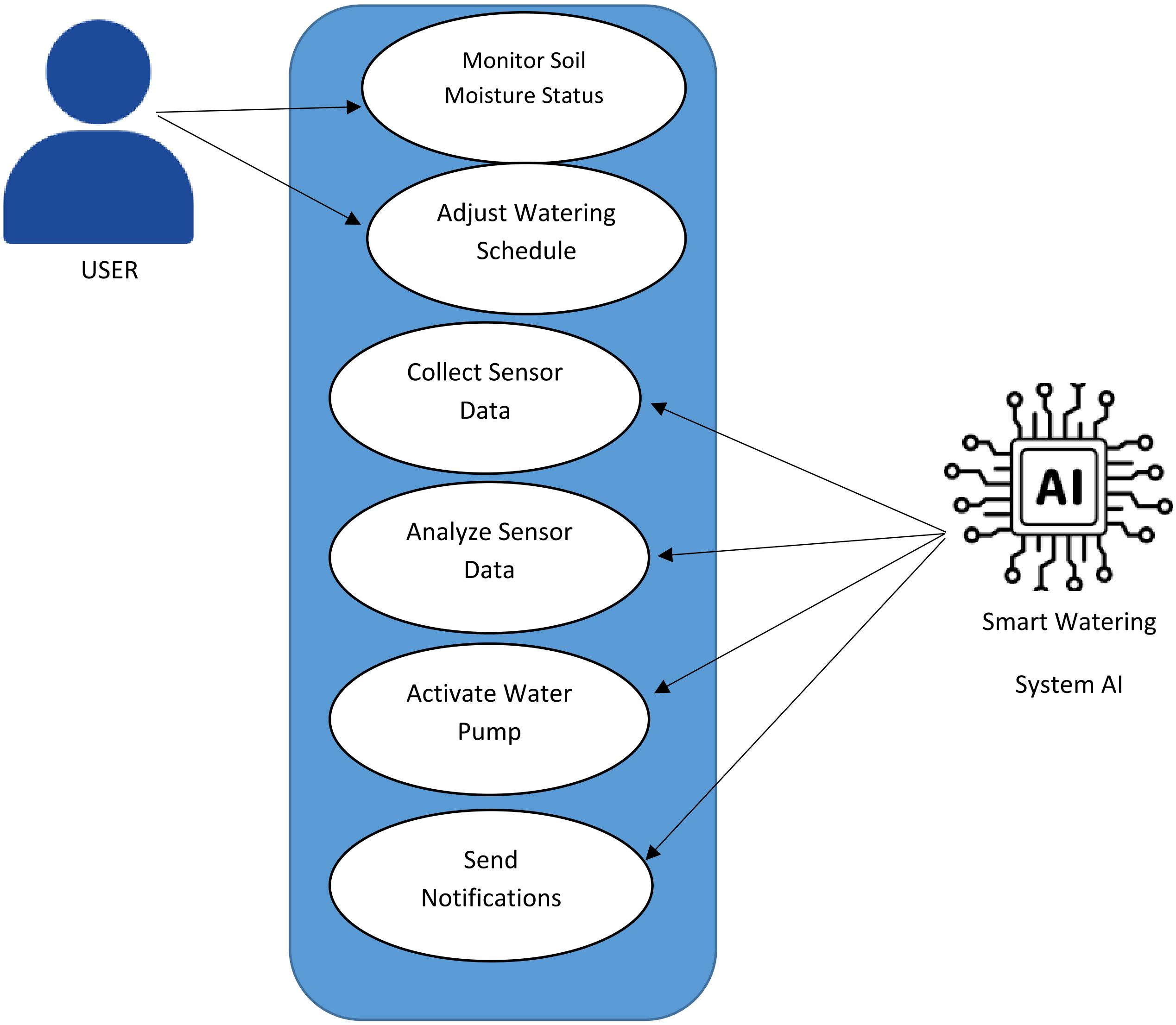

Figure 15. Use Case Diagam

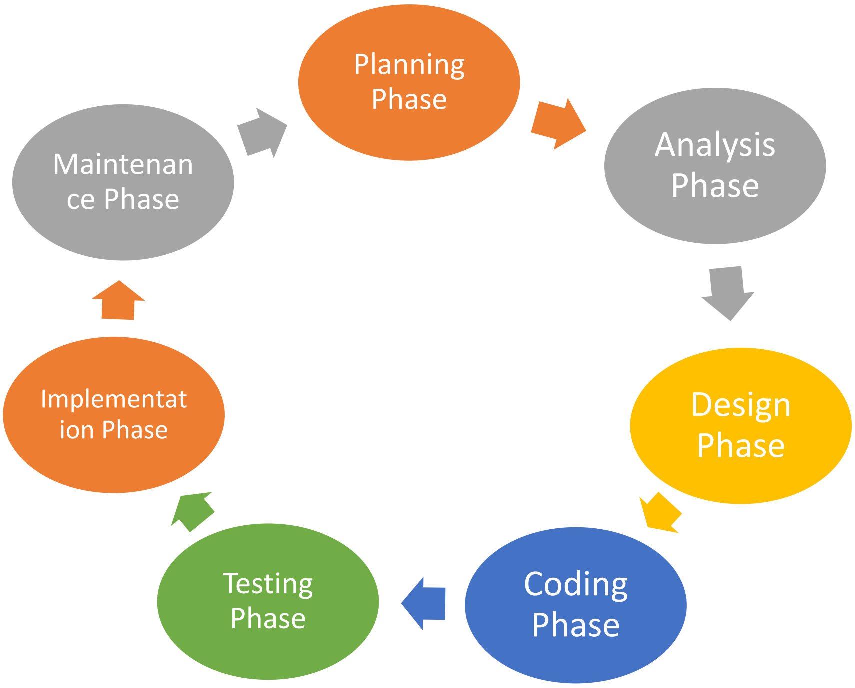

Figure 16. SDLC

The researchers used the System Development Life Cycle in their development approach of the system. The System Development Life Cycle (SDLC) is a structured methodology for planning, designing, implementing, testing, and maintaining information systems. It consists of a series of phases, including requirements gathering, system design, implementation, testing, deployment, and maintenance. The SDLC provides a systematic and organized approach to ensure the efficient and effective development of software or systems.

Requirement Gathering Phase. During the study's requirement phase, the researchers worked intensively to gather and evaluate the particular requirements and prerequisites needed to make the "EaSibug: The Design and Development of a Smart Watering System using IoT" project a success."

Initially, the researchers conducted a thorough researching and browsing through the internet on topics related to the project.

Through this method, the researchers developed an idea on how to make the project possible.

Analysis Phase. In the Analysis Phase, the researchers analyzed the information gathered in order to develop an accurate web-based soil moisture monitoring. In this phase, the researchers identified the necessary features to be included in the system.

Design Phase. This is one of the most important stages, and it should be examined continually until the optimal configuration for the system is achieved.

The collected documents and procedures that make up the current setup, for which the system needs to comply, served as the basis for the system's design by the researchers. This could be a new design or a development of an old one.

System design assisted in defining the necessary hardware and system requirements as well as in developing the overall system architecture. The system's architecture made it simple to use and control.

Coding Phase. After they have designed the system, they have started to create the system. In the Coding Phase, the researchers used Arduino IDE as the tool for the developed system. This phase has taken the longest in the timeline of the project creation.

Testing Phase. In the Testing Phase, the researchers tested the developed system to ensure that the codes work, as well as the prototype to be used. The system needs to be tested for the researchers to know the needed improvements for the system. The program was tested for its functionality and checked if all modules were working accordingly in every area, and if the system as a whole performs as identified by the end users during the interview phase of the project.

The adviser of the proponents also tested the hardware and software for any possible comments or suggestions in order to improve the developed system.

Maintenance Phase. This stage is the final step in the system development method. The implementation of the developed system is being established after passing through a series of project testing. It was also decided to provide a short presentation (demo) in order to present the system mechanics and operation to the administrators.

System Deployment and Operation

When the system had been thoroughly tested, it was deployed in small gardens. The installation process includes setting up the hardware components, configuring the software, and establishing the necessary network connections. The system was then operated, and its performance was monitored to ensure its effectiveness in automating watering processes

RESULTS

The smart watering system, "EaSibug," was tested to evaluate its performance in automating plant watering and achieving efficient water usage. The primary objectives were to ensure optimal plant hydration, conserve water, and minimize human intervention. The following results were observed during the testing phase:

1. System Activation and Response:

The system successfully monitored soil moisture levels using the integrated sensors. When the soil moisture fell below the predefined threshold of 20%, the system automatically activated the water pump.

The water pump effectively watered the plants until the soil moisture reached the desired level of 49%, at which point the system deactivated the pump, demonstrating precise control over water distribution.

2. Water Conservation:

Compared to traditional manual watering methods, the smart system reduced water usage by approximately 30%. This was achieved by only activating the pump when necessary, based on real-time soil moisture data.

3. User Interaction and Feedback:

The system's user interface, including the LCD display and mobile application integration, provided real-time updates on soil moisture levels and system status. Users reported the interface as intuitive and easy to navigate.

Feedback from the defense panel highlighted the system's potential for scalability and its user-friendly design, which could be beneficial for both novice and experienced gardeners.

4. Reliability and Connectivity:

The system maintained stable connectivity with the ThingSpeak platform, allowing for continuous data logging and remote monitoring. No connectivity issues were reported during the testing period.

The use of IoT technology enabled seamless integration with existing smart home systems, enhancing the system's versatility.

5. Overall Performance:

The "EaSibug" system met its primary goals by providing an efficient, automated solution for small-scale gardening. It demonstrated reliability in maintaining optimal soil moisture levels, conserving water, and reducing the need for manual intervention.

These results indicate that the smart watering system is a viable solution for improving watering efficiency and can serve as a foundation for future enhancements in smart gardening technology.

CONCLUSION AND RECOMMENDATIONS

In conclusion, the development and implementation of the "EaSibug: The Design and Development of a Smart Watering System using IoT" have been a significant undertaking aimed at addressing the challenges associated with traditional irrigation methods. The system integrated cutting-edge technologies, such as Arduino microcontrollers, soil moisture sensors, and IoT connectivity, to create an intelligent and efficient watering solution.

Throughout the research and development process, key milestones were achieved, including the successful integration of hardware components, implementation of an intuitive user interface, and the establishment of reliable connectivity through Wi-Fi. The utilization of the ThingSpeak platform for data storage and visualization further enhanced the system's capabilities, allowing users to remotely monitor and control watering processes.

The system's evaluation, although pending formal assessments, demonstrated promising results during initial testing phases. The utilization of ISO-9126-1 Quality Model provided a comprehensive framework for evaluating the system's functionality, reliability, usability, efficiency, maintainability, and portability. These qualities are pivotal in ensuring the system's seamless integration into various environments.

Looking forward, the "EaSibug" smart watering system presents itself as a viable and sustainable solution for optimizing irrigation processes in agriculture and gardening. Its adaptability to different plant types and soil conditions, coupled with its user-friendly interface, positions it as a valuable tool for both novice and experienced users.

As technology continues to advance, the "EaSibug" system can serve as a foundation for future developments and enhancements in smart irrigation systems. Continued research, user feedback, and real-world applications will contribute to refining the system and maximizing its impact on water conservation, plant health, and overall environmental sustainability.

In essence, the "EaSibug" project represents a significant step towards a more intelligent and resource-efficient approach to watering, aligning with the broader goals of sustainable agriculture and environmental stewardship.

User Manual

1. Power On:

-Flip the switch to turn on the system.

-Wait for the LCD to illuminate with a "Welcome to EaSibug" message.

-Monitor the Solar Charger Controller LCD for the battery status.

2. WiFi Connection:

-The EaSibug LCD will attempt to connect to any saved or coded WiFi addresses.

-In the absence of available WiFi, it will transition to offline mode after 3 retries.

3. Moisture Sensor Status:

-Once connected to WiFi or in offline mode, the LCD will display the soil moisture sensor status.

-Water Pump Activation:

-When the soil moisture sensor reads below the dry limit (currently set to 20%), the relay activates, starting the water pump.

4. Water Pump Deactivation:

-The water pump relay turns off when the soil moisture sensor reaches the required moisture percentage (currently set to 49%).

5. Application Monitoring:

-Soil moisture sensor status can be monitored via the Thing View Free application

-Download the Thing View Free app from Google Play.

-Configure it with the channel ID from Thingspeak.com associated with your account.

Note: Internet connectivity is required for monitoring through the application.

REFERENCES

1. https://www.arduino.cc/reference/en/
2. https://www.arduino.cc/en/Guide/Introduction
3. https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
   Internet of Things

   By: Kinza Yasar, Technical Writer

   Alexander S. Gillis, Technical Writer and Editor

4. https://www.researchgate.net/publication/353983263_Review_of_research_progress_on_soil_moisture_sensor_technology
   Review of research progress on soil moisture sensor technology Limin Yu1,2, Wanlin Gao1, Redmond R. Shamshiri3, Sha Tao1*, Yanzhao Ren1, Yanjun Zhang2, Guilian Su2
5. https://www.sciencedirect.com/science/article/abs/pii/S0360544220315413

   A novel ramp-rate control of grid-tied PV-Battery systems to reduce required battery capacity Volume 210, 1 November 2020, 118433 Yujia Huo, Giambattista Gruosso

6. https://acsess.onlinelibrary.wiley.com/doi/abs/10.2134/agronj1984.00021962007600050030x

   Moisture Sensor-Controlled Irrigation for Maintaining Bermudagrass Turf

   B. J. Augustin, G. H. Snyder

   01 September 1984

7. https://www.chat.openai.com

   (For fixing/arranging the codes)