Internet of Things-Based Smart Gardening System

Z. Neofytou, *Member, IEEE*

Abstract — This article outlines the development of a smart gardening system that utilises the Internet of Things (IoT) using the ESP32 Dev Kit. The system streamlines gardening labour by automating the monitoring of environmental factors such as temperature, humidity, and soil moisture, as well as handling critical tasks like watering and temperature control. The ESP32 microcontroller is equipped with hardware components including actuators like servo motors and relays, as well as sensors such the DHT11 Temperature and Humidity sensor, the HC-SR04 Ultrasonic sensor, and FC-28 Soil Moisture sensor. The Arduino IDE is used for software development, while integration with the Arduino Cloud enables remote monitoring functionalities. This paper encompasses system architecture, communication protocols, and hardware and software setup. The analysis of sensor data and the assessment of system performance are two of the resulting outputs. The study proposes more investigation and demonstrates how the Internet of Things (IoT) might improve agricultural activities.

Index Terms— Internet of Things, ESP32, smart gardening, environmental monitoring, Arduino Cloud, sensor integration, precision agriculture, remote monitoring, automation, sustainable agriculture, and data analytics.

1. INTRODUCTION

The custom of Growing Gardens has endured throughout time and adapted to advancements in technology. It is a fundamental component of human civilization. As the global population expands and concerns about environmental sustainability rise, there is a growing need to enhance traditional gardening methods. Consequently, we are embarking on a journey to explore the integration of Internet of Things (IoT) technology into gardening methods, which might provide a promising avenue for innovation. This introduction marks the start of our inquiry into the development of a smart gardening system based on the Internet of Things. Let us analyse the contents of this introduction more thoroughly:

1.1. Background

Upon examining the project's past, the challenges linked to traditional gardening techniques were discuss. Insufficient upkeep and inappropriate allocation of resources are prevalent issues in gardens worldwide, leading to decreased plant well-being and production. Now, let us analyse the primary issues:

1.1.1. Struggles in Garden Care

Traditional gardening is significantly hindered by neglect and inadequate care practices. Insufficient crop yields and impaired vegetation might arise from issues pertaining to insect control and irregular watering.

1.1.2. Challenges for Small-scale Gardeners

Small-scale gardeners have difficulties due to their restricted access to modern resources, particularly in rural areas. Coping with agricultural failures and financial crises may be arduous owing to resource scarcity.

1.1.3. Environmental Impact:

Conventional gardening methods may have a detrimental impact on the environment by introducing chemicals that pollute the soil and water. This leads to risks that might negatively impact biodiversity and the long-term sustainability of ecosystems.

1.2. Motivation

This objective is motivated by a strong desire to enhance the efficacy, sustainability, and variety of gardening activities. The goal is to empower individuals and communities to develop flourishing gardens while minimising their ecological footprint via the use of IoT technologies in gardening.

1.3. Objectives

The primary objective is to build and implement an Internet of Things (IoT) smart gardening system [2] that enhances gardening experiences, promotes robust plant growth, optimises resource utilisation, and automates environmental monitoring.

1.4. Scope

The specifications of the project are established, including the hardware and software elements used, as well as the characteristics of our intelligent gardening system. The presence of potential obstacles, such as financial and technical limitations is recognised, and possible solutions were discussed. Importance: This methodology has immense promise in addressing the limitations of traditional gardening approaches. The objective is to have a substantial impact on the responsible management of the environment and the assurance of food availability by advocating for sustainability, bolstering agricultural productivity, and conserving resources.

1.5. Overview of the Report

Finally, a glimpse of the next portions of this report including the methodology, system design, implementation particulars, and assessment findings.
The aim is to provide readers with a clear outline of this inquiry, allowing them to understand the extent and thoroughness of the research.

2. LITERATURE REVIEW:

In this section, the literature, research, and statistics pertaining to Internet of Things-based smart gardening systems, were thoroughly analysed. It is needed to comprehend prior research in this field in order to efficiently guide this effort. The topics of discussion will include:

2.1. Overview of IoT Applications in Gardening:

The Internet of Things technology is revolutionising gardening by enabling the measurement of plant health, monitoring of environmental conditions, and optimisation of resource utilisation. This foundational information enhances our understanding of the potential benefits and challenges associated with using IoT in gardening methodologies.

2.2. Smart Gardening Systems:

Historically, studies have investigated intelligent gardening systems that optimise irrigation, plant maintenance, and pest management via the use of automation, actuators, and sensors. By studying these projects, we may get information about effective design strategies and identify possible issues to avoid in our intelligent gardening system.

2.3. Environmental Monitoring in Gardening:

To ensure optimal growth in gardening, it is necessary to meticulously monitor and control temperature, humidity, and soil moisture levels. Accurate environmental data enables educated decision-making and promotes better plant development, leading to a more productive garden.

2.4. Automated Watering Systems:

Utilising automated systems to water plants is a very efficient method to provide enough moisture and reduce the risk of both excessive or insufficient watering. In this analysis, we will examine many irrigation methods and evaluate their effectiveness in conserving water while promoting optimal plant growth and health within a garden setting.

2.5. Integration of Actuators in Smart Gardening:

Actuators like pumps and valves are crucial in smart gardening systems to automate tasks including as humidification, shade control, and watering. Understanding the roles and purposes of these functions enables us to design a reliable and efficient system that is well-suited for a garden environment.

2.6. Remote Monitoring and Control:

Remote monitoring and control tools, including as internet interfaces and mobile applications, provide convenient access to garden data and management functionalities. With the advancements in technology, gardeners now have the ability to remotely monitor and adjust conditions in their gardens, providing them with more flexibility and convenience in caring for their plants.

2.7. Summary and Synthesis:

A concise overview of the key discoveries derived from our comprehensive examination of existing literature will be provided. This will include highlighting common patterns and identifying prospective avenues for further investigation within the realm of gardening. The future phases of this project are directed by this synthesis, ensuring conformity with existing knowledge and addressing any deficiencies in understanding of garden management.

3. SYSTEM ARCHITECTURE

3.1. Overview

The objective of the system design for the Internet of Things-based Smart Gardening System is to provide a fully automated and intelligent environment for cultivating plants. The primary idea is to use the ESP32 Dev Kit to establish a connection with various sensors and actuators in order to oversee and control crucial garden variables such as soil moisture, water level, temperature, and humidity. Subsequently, the collected data is sent to the Arduino Cloud for remote scrutiny and analysis. The system's intelligence stems from its capacity to make real-time judgements and provide optimal growth circumstances for plants.

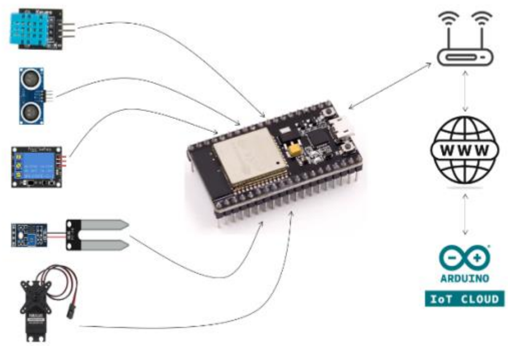

Fig. 1. An Illustration of System Architecture

3.2. Hardware Components

The hardware components serve as the foundation of the Smart Gardening System, facilitating the gathering and control of environmental factors. The essential hardware components comprise:

3.2.1. ESP32 Dev Kit

ESP32 [3], the central processing unit is responsible for orchestrating sensor readings, actuator control, and communication with the Arduino Cloud with the help of Wi-Fi connectivity for remote monitoring and control.

Fig. 2. ESP32 Dev Kit V4

3.2.2. DHT11 Temperature and Humidity Sensor

The DHT11 sensor allows the system to regulate conditions depending on predefined thresholds by monitoring the garden's ambient temperature and humidity levels.

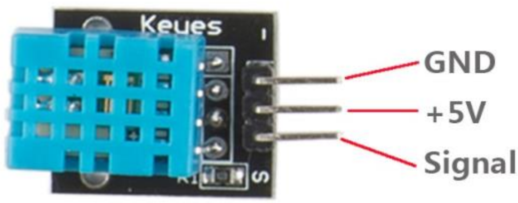

Fig. 3. DHT11 Sensor

3.2.3. HC-SR04 Ultrasonic Sensor

By preventing overwatering or water scarcity, the HC-SR04 monitors the water level in the tank and guarantees effective water management.

Fig. 4. HC-SR04 Sensor

3.2.4. Servo Motor

Controls the garden door/windows, allowing for automated opening and closing. Facilitates ventilation and protection of plants from external factors.

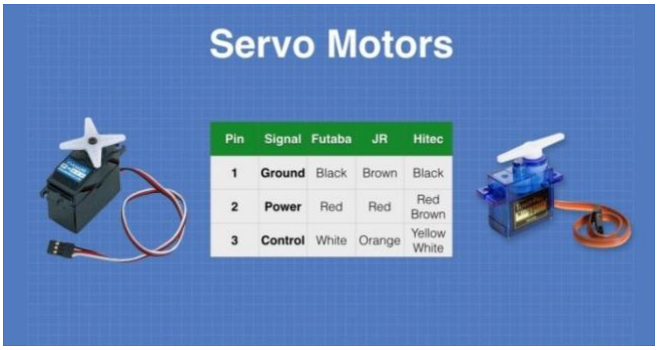

Fig. 5. Servo Motor for ESP32

3.2.5. Relay Modules

Motor Pump Relay: Controls the water pump for automated watering based on soil moisture levels.
Humidifier Relay: Activates the humidifier when humidity falls below the desired level.
Heater Relay: Controls the heater to maintain optimal temperature conditions.

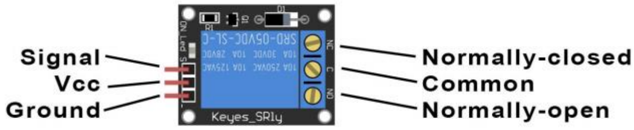

Fig. 6. Relay

3.2.6. FC-28 Soil Moisture Sensor

Monitors the moisture level in the soil. Guides the system in determining when to initiate watering cycles.

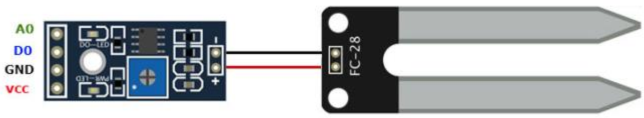

Fig. 7. FC-28 Soil Moisture Sensor

3.3. Software Components

The software parts oversee controlling data processing, enabling connectivity with the Arduino Cloud, and programming the ESP32 Dev Kit. The primary components of the software are as follows:

3.3.1. Arduino IDE Programming

Arduino IDE [4] is used for programming code for the ESP32 to manage communication protocols, actuator control, and sensor readings. Code uses algorithms for generating decisions to provide automatic reactions based on sensor data.

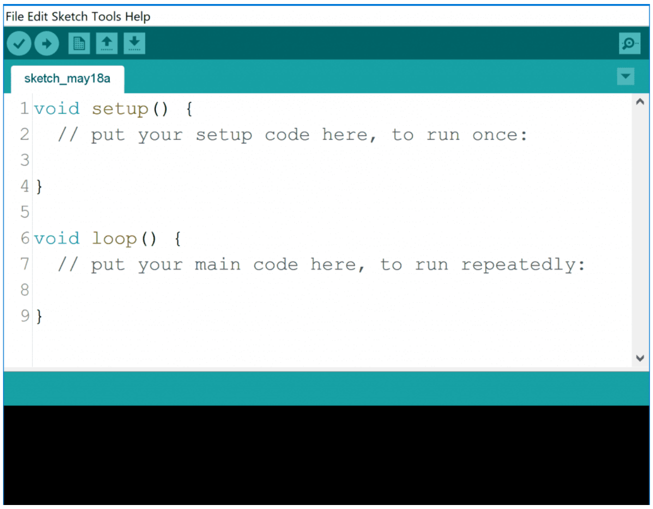

Fig. 8. Arduino IDE Default Screen

3.3.2. Cloud Integration (Arduino Cloud)

- Establishment of a secure connection between the Smart Gardening System and the Arduino Cloud [5].
- Enables remote monitoring and control through

a user-friendly interface.

- Facilitates data logging and analysis for performance evaluation.

Fig. 8. Arduino IoT Cloud

The Smart Gardening System's sturdy architecture is produced by the thorough integration of hardware and software components, which enables accurate environmental parameter monitoring and control to maximize plant growth circumstances.

4. Methodology

The methodology section provides a comprehensive description of the approach used to strategize, execute, and assess the Internet of Things-based Smart Gardening System. This encompasses the hardware and software implementation, system design, and overall technique used to provide a reliable and efficient smart gardening solution.

4.1. System Design

Defining the structure and capabilities of the Smart Gardening System is an essential undertaking during the system design process. This involves dismantling the overall system into smaller, more manageable components and understanding the interconnections between these components.

4.1.1. Sensor Integration

Selecting and incorporating sensors such as the soil moisture sensor FC-28, the water level sensor HC-SR04, and the temperature and humidity sensor DHT11. And establishing protocols for communication between the ESP32 Dev Kit and the sensors.

4.1.2. Actuator Integration

Assembling and setting up actuators, such as the garden door servo motor and the water pump, humidifier, and heater relay modules. Create logic that will allow actuators to be controlled by sensor readings.

4.1.3. Cloud Integration

Establishing the ESP32 Dev Kit and Arduino Cloud's connectivity. Arduino Cloud provides an easy-to-use interface for controlling and monitoring remotely.

4.1.4. Arduino IDE Programming

The programming phase involves writing code for the ESP32 Dev Kit to implement the designed system.

This includes:

- Sensor Readings: Code to collect data from sensors, ensuring accurate and timely readings.
- Actuator Control: Logic to control actuators based on sensor data, such as activating the water pump when soil moisture is low.
- Cloud Communication: Code to establish a secure connection to the Arduino Cloud, enabling real-time data transmission.
- Sensor Calibration: Calibration is essential to ensure accurate readings from the sensors. This involves adjusting sensor values to match real-world conditions and improve the overall reliability of the system.

4.2. System Integration

The ESP32 Dev Kit, DHT11 for temperature and humidity, HC-SR04 for water level sensing, FC-28 soil moisture sensor, servo motor for door control, and relays for pump, humidifier, and heater control are examples of components that work cohesively together to form a system. Through the establishment of strong communication protocols, interface configuration, and data flow optimization, this complex procedure guarantees the smooth operation of hardware and software components. These elements function together with effective integration to enable data-driven insights, precise control, and real-time monitoring [6] [7]. This leads to an intelligently linked gardening solution that maximizes the potential of every part for a more effective and enjoyable gardening experience.

4.2.1. Testing and Debugging

To ensure correct functionality, this phase entails extensive testing of hardware components such as the ESP32 Dev Kit, sensors, and actuators. The Code is tested for correct data collecting and connectivity with Arduino Cloud at the same time. Robust debugging is utilized to tackle any detected problems, guaranteeing a seamless and dependable functioning of the intelligent gardening system.

The IoT-based Smart Gardening System can be developed methodically with the help of the above-described methodology, which guarantees that every part is properly designed, put into practice, and tested for peak performance.

5. Implementation Details

5.1. Hardware Implementation

During the hardware implementation stage, the IoT-Powered Smart Gardening System is put together and tested to make sure it works as it should. Configuring the ESP32 Dev Kit involves attaching it to necessary peripherals and setting up a reliable power source. Actuators and sensors such as the servo motor for door control, HC-SR04 ultrasonic sensor for water level measurement, FC-28 soil moisture sensor, and DHT11 temperature and humidity sensor are incorporated according to the predetermined system design. To guarantee correct operation, every connection is carefully examined, and any possible hardware problems are quickly fixed.

Fig. 9. Arduino Cloud Dashboard

When it comes to managing appliances like the heater, humidifier, and water pump, relays are essential. In this stage, relay modules are set up and tested to ensure that they precisely control linked devices by system specifications.

Comprehensive testing and debugging are carried out to evaluate the hardware configuration as a whole. This entails checking actuator reactions, confirming sensor data, and closely examining each component's performance. To achieve smooth integration and dependable operation of the complete smart gardening system, anomalies found during testing are methodically rectified.

5.2. Software Implementation

The digital elements of the IoT-Based Smart Gardening System have been established and the control mechanisms have been incorporated.

The primary programming environment for the ESP32 Dev Kit is the Arduino IDE. The purpose of code is to facilitate connection with the Arduino Cloud, manipulate actuators, and collect data from sensors. Code optimisation methods are employed to ensure efficient performance and resource utilisation.

The firmware development include programming logic for actuator control, sensor data collection, and communication protocols to provide seamless interaction with the cloud-based monitoring system. Calibration procedures are used to enhance the accuracy of sensor readings and increase the overall precision of the system.

Testing techniques include evaluating the code's efficiency and verifying the precision of sensor data and actuator reaction. A meticulous process of debugging is conducted to identify and resolve any issues, ensuring seamless compatibility between the software and hardware components.

Configuring the system for remote control and monitoring is a crucial step in the integration process of Arduino Cloud. This involves establishing channels of communication with the cloud platform and defining the specific characteristics that need to be monitored.

5.3. Cloud Implementation

A key component of the smart gardening system is its interaction with Arduino Cloud. Configuration procedures entail preparing the system for remote control and monitoring. Establishing communication channels with the cloud platform and specifying the parameters to be monitored are part of this process. The system is tested to guarantee that it will connect with Arduino Cloud without a hitch, allowing for real-time data monitoring and a convenient remote control interface. One can also monitor the garden using a mobile app [8].

5.4. Circuit Diagram

Below is a schematic diagram of the system:

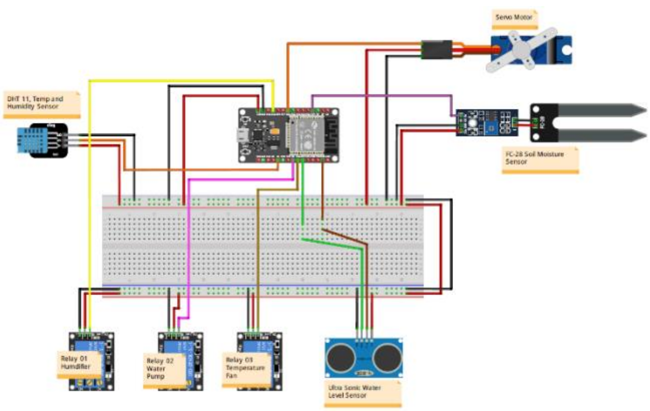

Fig. 10. Wiring schematics of system

6. Results and Discussion

The project underwent extensive testing and assessment after the hardware and software implementation was finished and the Arduino Cloud integration was completed. The results are emphasized in the parts that follow, along with details on the functioning and operation of the system.

6.1. Hardware Setup and Configuration

The successful physical assembly of the components was demonstrated by the ESP32 Dev Kit's smooth peripheral connection. The system design was followed in the integration of sensors, actuators, and relays, and Wi-Fi settings were set up for connection with the Arduino Cloud. During the initial testing, the system proved to be stable and the correct connections were confirmed.

6.2. Software Implementation and Coding Practices

The Arduino IDE was used to program the software, which correctly carried out instructions to enable actuator control, sensor data collecting, and communication with the Arduino Cloud. The efficient coding techniques were evident in the system's seamless operation. The process was made simpler by the software's clear instructions, which improved the smart gardening system's overall responsiveness.

6.3. Arduino Cloud Results

The successful integration with the Arduino Cloud made it possible to monitor and control things remotely. Real-time data monitoring was made possible by the system's design, which also gave users an easy-to-use interface for remotely monitoring and controlling the garden's characteristics. Preliminary experiments validated the smooth communication between the Arduino Cloud platform and the smart gardening system.

6.4. Discussion and Future Considerations

Strong groundwork is done for a dependable and functional smart gardening system by successfully integrating hardware and software components and integrating them with the Arduino Cloud. In the future, it might be necessary to explore new features, improve the system's usability through refinement, and further optimize its performance in response to user input and changing requirements. The goal of the conversation is to assess the attained outcomes, pinpoint areas in need of improvement, and provide the groundwork for the system's future evolution.

7. Challenges Faced and Solutions

7.1. Sensor Calibration Challenges:

When the FC-28 soil moisture sensor proved challenging to calibrate for precise readings, incremental calibration procedures and modifications were put in place, which increased data collecting precision.

7.2. Software Code Optimization:

The problem of writing efficient code was addressed by reworking existing code and implementing resource-efficient methods, which greatly improved system performance and prevented latency.

7.3. Cloud Integration Complexity:

To successfully and seamlessly communicate between the smart gardening system and the cloud platform, thorough documentation and methodical testing were used to address the complexities of connecting the system with Arduino Cloud.

7.4. User Interface Enhancement:

To improve user experience, the task of creating an interface that is easy to use was addressed by adopting a straightforward and intuitive design and continuously refining it based on usability testing.

8. Conclusion

The Internet of Things-Based Smart Gardening System, using the ESP32 Dev Kit together with a range of sensors and actuators, signifies a pioneering progression in precision agriculture. The integration of physical components, such as meticulous networking settings, together with sophisticated software, has yielded a resilient and efficient system.

The accurate understanding of the garden's climatic conditions has been facilitated by the effective calibration of sensors, particularly the FC-28 soil moisture sensor, resulting in enhanced data accuracy. The project team's adept problem-solving was shown by the systematic manner in which challenges with hardware setup, sensor calibration, and cloud integration were overcome.

The code was developed using the Arduino IDE and demonstrated effective resource allocation, as well as proficiency in coding methodologies. The smart gardening system's overall responsiveness and reliability have been improved by efficient algorithms and precise directions.

The integration of the system with Arduino Cloud enables real-time data access and facilitates the remote monitoring and administration of the garden's components. This capability enables the implementation of intelligent gardening in many settings, ranging from individual homes to big agricultural operations.

All the challenges that occurred throughout the project, such as issues with hardware connection, fine-tuning sensor calibration, and the complexity of integrating with the cloud, were systematically overcome. These achievements demonstrate that the system is a versatile tool that may be used in smart home gardening and precision agriculture.

The IoT-Based Smart Gardening System shown is a sophisticated blend of cloud, software, and hardware technologies. The system's scalability and problem-solving capabilities make it a practical choice for many agricultural applications. The project is a promising undertaking in the realm of smart gardening systems. It establishes the groundwork for future enhancements, optimisations, and potential integration with advanced technology.

9. Future Work

Several directions for future research and development present themselves as the IoT-Based Smart Gardening System approaches a major developmental milestone. These pathways present chances for improvements, extensions, and additional investigation. The following areas indicate potential future growth directions:

9.1. Enhanced Sensor Integration:

- Investigating adding more sensors to increase the scope of metrics being tracked, like light intensity, insect control, or other atmospheric conditions.
- Looking into cutting-edge sensor technology to increase data-collecting accuracy and dependability.

9.2. Machine Learning Algorithms:

- Using machine learning algorithms to evaluate the gathered information and offer forecasts.
- Creating models for automated decision-making that maximize resource use in gardening by drawing on patterns in past data.

9.3. Automation and Robotics:

- Investigating the application of artificial intelligence (AI) to enable the system to adapt and respond dynamically to changing environmental conditions.
- Integrating robotic systems for automated planting, harvesting, and maintenance operations.

9.4. Energy Harvesting Solutions:

- Implementing energy harvesting technology, including solar or wind energy, to improve the system's sustainability.
- Put energy-saving devices into place and investigate the viability of using different power sources.

9.5. User Interface Refinement:

- Use usability testing and user feedback to continuously improve the user interface.
- Use user-customizable alerts and dashboards to provide users with a more engaging and tailored experience.

9.6. Community and Social Integration:

- Examining how social and community aspects might be integrated to enable users to exchange gardening views, advice, and experiences.
- Creating collaborative tools to encourage knowledgeable gardeners to share their knowledge.

9.7. Security and Privacy Measures:

- Putting strong security measures in place to guard user information and guarantee system integrity.
- Looking into privacy-preserving strategies that give users authority over how their gardening data is shared and stored.

9.8. Commercial Deployment and Scaling:

- Carrying out feasibility studies with an eye on cost-effectiveness and scalability for commercial deployment.
- For large-scale implementations, look into joint ventures with enterprises or agricultural organizations [9].

The IoT-Based Smart Gardening System can advance into a more complex and adaptable solution by concentrating on these future research areas, which will increase its influence on smart gardening and precision agriculture techniques. Every path offers different chances and difficulties, which support the industry's ongoing innovation and progress.

10. References

[1] A. Shaik Masthan Babu, A. Jaya Lakshmi, and B. Thirumala Rao, "A study on cloud-based Internet of Things: CloudIoT," in *2015 Global Conference on Communication Technologies (GCCT)*, pp. 1-5, Thuckalay, India, IEEE, DOI: 10.1109/GCCT.2015.7342624.

[2] N. Sukhdev, N. Nahata, S. Sridhara, and G. Swamy, "IoT Enabled Smart Gardening," in *2018 Fourteenth International Conference on Information Processing (ICINPRO)*, pp. 1-2, Bangalore, India, IEEE, DOI: 10.1109/ICINPRO43533.2018.9096672.

[3] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of things," in *2017 Internet Technologies and Applications (ITA)*, pp. 1-6, Wrexham, UK, IEEE, DOI: 10.1109/ITECHA.2017.8101926.

[4] C. Wootton, "Using Arduino IDE," in *Beginning Samsung ARTIK*, pp. 195-218, DOI: 10.1007/978-1-4842-1952-2_13.

[5] A. Kurniawan, "Arduino IoT Cloud," DOI: 10.1007/978-1-4842-6446-1_5.

[6] M. A. Muhtasim, S. Ramisa Fariha, and A. M. Ornab, "Smart Garden Automated and Real Time Plant Watering and Lighting System with Security Features," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 676-679, Greater Noida, India, IEEE, DOI: 10.1109/GUCON.2018.8675077.

[7] A. Al-Omary, H. M. AlSabbagh, and H. Al-Rizzo, "Cloud based IoT for smart garden watering system using Arduino Uno," in *Smart Cities Symposium 2018*, pp. 1-6, Bahrain, IEEE, DOI: 10.1049/cp.2018.1401.

[8] M. Sheth and P. Rupani, "Smart Gardening Automation using IoT With BLYNK App," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 266-270, Tirunelveli, India, IEEE, DOI: 10.1109/ICOEI.2019.8862591.

[9] S. Salvi et al., "Cloud based data analysis and monitoring of smart multi-level irrigation system using IoT," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 752-757, Palladam, India, IEEE, DOI: 10.1109/I-SMAC.2017.8058279.