# Cloud-based Internet of Things Approach for Smart Irrigation System: Design and Implementation

Thaer Thaher
*Information Technology Engineering*
*Al-Quds University*
*Jerusalem, Palestine*
thaer.thaher@gmail.com

Isam Ishaq
*Computer Engineering Department*
*Al-Quds University*
*Jerusalem, Palestine*
isam@alquds.edu

***Abstract*—Water plays a significant role in the economic development of countries. The agriculture sector is the most water-consuming; this sector consumes 69% of the freshwater. However, farmers often use traditional irrigation systems to water their crops. These systems are ineffective and consume a lot of time and effort, especially when there are several fields distributed in different geographical regions. Therefore, employing smart irrigation techniques will significantly overcome these problems. In this paper, we propose an intelligent irrigation framework based on Wireless Sensor Network (WSN) and Internet of Things (IoT) cloud services. The proposed system consists of three main components; the WSN, the control unit, and cloud services. Arduino Uno and XBee ZigBee modules are combined to gather sensors data and send them wirelessly to the control unit. YL-69 sensor is used to monitor the soil moisture. Raspberry Pi is utilized to gather data, process them, provide the proper decision, and transfer them to ThingSpeak IoT cloud. In the cloud, the data collected from the system is stored to create instance visualization of live data and send alerts. This allows farmers to monitor the status of crops and make the required decisions. After inspecting the prototype, many challenges are posed for future work.**

***Index Terms*—IoT, Smart irrigation, Cloud computing, ThingSpeak, Arduino, Raspberry Pi, XBee zigbee.**

## I. INTRODUCTION

Water plays a significant role in the economic development of countries. Specifically, the availability and accessibility of freshwater creates enormous opportunities for developing countries and is an effective strategy for sustainable economic growth, social, and environmental benefits [1]. Thus, investing in water management and improving its services is essential to eradicate poverty and make a difference in the lives of billions of people. For example, in Palestine, water scarcity is considered one of the most critical problems for environmental and political reasons, as water resources are limited, and the Israeli occupation controls most of them. Consequently, the optimal utilization of the available resources, as well as implementing effective strategies, has become an important requirement to handle this problem [2].

According to AQUASTAT (the Food and Agricultural Organization (FOA) global information system), 69% of the freshwater is consumed by the agriculture sector at the worldwide level [2], [3]. Water is essential for agricultural production and plays an important role in food security. Irrigated agriculture contributes to about 40% of the total food produced worldwide [4]. However, farmers often use traditional irrigation systems to water their crops. These systems are ineffective and consume a lot of time and effort, especially when there are several fields distributed in different geographical regions. For example, in developing countries, in particular, farmers are obliged to visit their agricultural fields regularly and remain present during the irrigation process to direct and organize the irrigation for the areas concerned. Moreover, traditional irrigation often leads to the problem of over-irrigation and hence consuming large amounts of water. It also may cause an under-irrigation problem that negatively impacts the quality of crops [5].

Adapting modern technologies to build smart automatic irrigation systems has attracted considerable interest in the agricultural domain [6]. The proper design and installation of an automated irrigation system have many benefits compared to the traditional one. That is to say, it will schedule the irrigation process in an optimal way and thus contribute to reducing time, effort, and cost. It also saves significant water while improving plant quality. The availability of innovative technologies such as the Internet of Things (IoT), sensors, wireless sensor networks (WSNs), and embedded processing significantly contributed to building effective irrigation systems. The availability of such technologies helps in controlling and monitoring the levels of water available for crops without or with minimal human intervention [7]. Moreover, they allow generating a vast amount of data that can be stored, visualized, and analyzed to extract valuable knowledge that will improve the overall planning strategies and decision-making process.

In this paper, a smart irrigation system was developed based on Raspberry Pi, Arduino Uno, and wireless sensor network technologies to control and monitor the irrigation process. The main objectives of this project are summarized as follows:

- Transferring the measured quantities by moisture sensors wirelessly to a central processing unit (Raspberry Pi). A WSN using XBee modules is utilized to achieve this task.
- The gathered data is transferred to the cloud for storing, visualizing, and monitoring the agricultural parameters.
- The system provides efficient automated irrigation that ensures high quality of crops, saving water, and reducing human involvement.

The proposed solution, if it is generalized, especially in developing countries, will contribute to achieving two of the sustainable development goals (SDGs) [8]. First, it will ensure $Goal\#1$ (No poverty). Second, it will partially ensure $Goal\#6$ (Clean Water and Sanitation). These facts motivate us to provide this solution at the global level and also encourage us to provide a solution that contributes to the exploitation of our local water resources in Palestine, which suffers from the water scarcity problem.

The rest of the paper is structured as follows: The related approaches for developing smart irrigation systems are reviewed in Section II. Section III presents the details of the proposed design and architecture. The implementation and experimental results are discussed in Section IV. Finally, the conclusion and future directions are drawn in Section V.

## II. REVIEW OF RELATED WORKS

Several IoT related approaches have been employed in different manners for developing automated irrigation solutions. The proposed solutions vary in terms of their complexity, and the modules used in them, such as Arduino, Raspberry Pi, types of sensors based on the targeted problem nature and the main objectives. In this section, we review the recent works that are directly related to our proposed system.

Anbarasi et. al [9] proposed an IoT-based smart irrigation framework with the aim of increasing the crop yield. In this system, a real-time decision was made based on the collected data from the land. A moisture sensor was used to sense the level of soil moisture, and a pumping motor will turn OFF or ON based on the sensed data. To provide the farmer with information about moisture content and temperature of the soil at particular times, a web-based application was developed.

An automated irrigation system using the Arduino microcontroller was introduced by Hassan et al. [1]. In this system, a soil moisture sensor was utilized to check the soil humidity of the plant. Based on the sensed soil moisture level, the system will let the water pump to automatic irrigate the plant when it is too dry and turn off the water pump when the soil of the plant is wet. The proposed solution is composed of one sensing node. It is simple and did not handle the gathered data. It provides the monitoring function where users can check the soil moisture based on the reading on the Liquid Crystal Display (LCD).

In 2018, Al-Omary et al. [10] introduced a smart cloud-based IoT strategy for monitoring and controlling the garden irrigation process using Arduino Uno. Two sensors (soil moisture and light intensity) were used to monitor and maintain the garden soil moisture and the light intensity quantities. These quantities are continuously sent to ThingSpeak cloud-based framework in which the received data is visualized and analyzed. The proper decision is sent accordingly from the cloud to the automatic irrigation system to water the garden. However, the authors did not consider other irrigation parameters, such as air humidity and temperature. Besides, transferring all gathered data to the cloud would require a massive amount of data and increase the latency and privacy issues.

Fawzi and Jalal [11] utilized a WSN-based IoT strategy to design and implement a smart irrigation system. The WSN was employed for gathering, storing, and sharing sensors data. Two types of sensors, namely Telosb and VH400, were used. Telosb was employed for measuring light, temperature, humidity, and energy, while the VH400 sensor was used to measure soil moisture. Gathered data is sent to the cloud server for analysis and computation purposes. In this system, the concept of fog computing was not considered. In other words, all gathered data is sent to the cloud server for analysis and decision making.

In 2019, an automatic irrigation system based on Arduino was proposed by Okoye et al. [7]. The primary purpose of this study was to identify the irrigation time for three different types of soil in Nigeria. For this purpose, Arduino Uno integrated with soil moisture was used to control the irrigation process. The water level and other essential data status are displayed on an LCD. This system is simple and dedicated to studying how irrigation time depends on the type of soil. It did not consider the main concept of IoT because the collected data are not shared via the internet.

Mostly, it is clear from the reviewed studies that most of them used Arduino Uno. Few of them deployed WSNs and cloud-based approaches. Besides, those who considered the cloud services were transferring all gathered data to the cloud for the analysis and decision-making. Fog computing concepts mostly did not utilize. This strategy is not scalable, especially when a massive amount of data is generated and thus increases the latency and privacy issues. These facts motivated us to propose an efficient smart irrigation framework by utilizing the efficiency of WSN, fog computing, and cloud-based services.

## III. THE PROPOSED SYSTEM AND ARCHITECTURE

This study aims to propose an efficient smart irrigation framework. The system can be divided into three parts (or subsystems) that cooperate to achieve the purpose: A Wireless Sensor Network, a central processing unit, and cloud computing services. The complete smart watering architecture is depicted in Figure 1. The following subsections introduce the theoretical background as well as the main components of each part.

### *A. Wireless sensor network*

The sensor is a device that measures a property of the physical environment, such as moisture, and converts it to data. The WSN consists of interconnected sensor nodes that communicate wirelessly to collect data about the surrounding environment [12]. In this project, the main parts of the WSN are sensor nodes, and the coordinator (gateway).

*1) Sensor node:* In general, the sensor node is employed to gather data from different types of sensors. It can accomplish limited processing tasks based on available resources. The main feature is the presence of the transceiver that allows the exchange of information with other nodes. The main four units that describe any sensor node are sensing unit, processing unit, transceiver unit, and power source unit [13]. In this project,

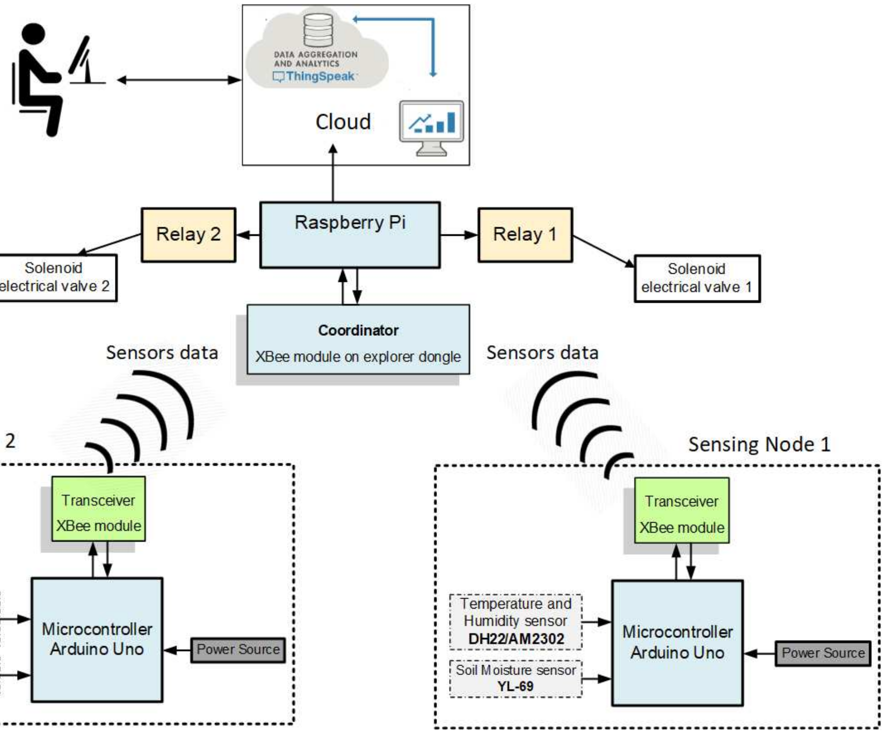
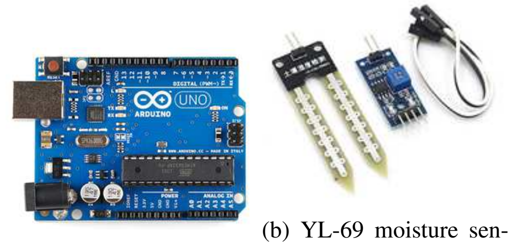


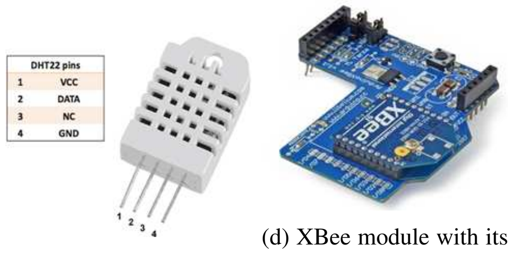
Fig. 1: The conceptual diagram for the proposed smart irrigation framework

we employed the following four components as a sensor node (see Figure 2):



(a) Arduino Uno

(b) YL-69 moisture sensor



(c) DH22 sensor

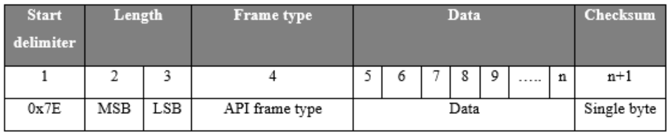
(d) XBee module with its shield

Fig. 2: : Sensing node components

- **Arduino Uno**: Arduino is an open-source platform utilized for developing electronic projects. We selected Arduino Uno due to its flexibility and ease of integration with the XBee module. The primary purpose of Arduino in this project is to read the input values of the connected sensors and send them serially to the connected Xbee module.
- **YL-69 moisture sensor**: This sensor is usually used to measure the humidity of the soil (see Figure 2 (b)). In this project, we used the analog signal pin, and thus the output voltage value will be between 0 and 1023. The voltage value decreases as the soil becomes more moisture while it increases as the soil becomes dry.
- **The humidity and temperature sensor (DH22)**: The DH22 is a digital-output sensor utilized to measure relative humidity and temperature. This sensor can get new readings every 2 second, so timing should be considered carefully. For more details, refer to the sensor datasheet [14].
- **XBee wireless communication module**: It is a wireless communication module useful for applications that require low-power consumption. It is easy to deploy and can be configured using the XBee Configuration and Test Utility (XCTU) platform [15]. XBee can also ensure the safe delivery of data to the targeted node using addressing, acknowledgments, and retries. The checksum approach is used to ensure error-free communication. XBee module can be distinguished by two addresses: a fixed 64-bit address that is not configurable (MAC address) and a 16-bit configurable address. Two types of communication modes, called transparent mode and Application Programming Interface (API) mode, can be utilized to describe the way the host device communicates with an XBee module through the serial interface. In the

API mode, a protocol is used to identify the way information is exchanged. In this mode, data is encapsulated in a structured packets called frames, the structure of the frame is defined by four parts: start delimiter, length, Frame Data, and checksum as shown in Figure 3. This mode is better for large WSN when data collected from multiple remote locations. Therefore, we used it in this work.

- **Power source unit and power management**: The moisture sensor will be planted in the soil, so the nodes, as well as the attached sensors, are far from the power source. In this prototype, we used a 9v Alkaline battery to provide the sensor node with the required power. We can also utilize solar panels to recharge the batteries.
  Minimizing the energy used by the device or increasing the battery capacity will increase the lifetime of the battery and reduce the total cost of ownership of the product. XBee transceiver is the most critical part of this project (the most significant energy-consumption part). Therefore, the XBee should be appropriately configured to allow it to work in the sleep mode most of the time. In this project, the remote XBee remains in the sleeping mode for a while and wakes up when it receives data from the microcontroller. Moreover, the ZigBee communication protocol used in our solution is distinguished as a low power consumption protocol compared to other wireless protocols such as Wi-Fi.



| Start delimiter | Length | | Frame type | Data | | | | | | | Checksum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ….. | n | n+1 |
| 0x7E | MSB | LSB | API frame type | Data | | | | | | | Single byte |

Fig. 3: API mode frame structure

### B. Zigbee wireless communication protocol

ZigBee is a primary type of wireless communication protocol that is used for communication in Wireless Personal Area Network (WPAN). The low power consumption, low data rate, and low cost are the main features of ZigBee. Therefore, it is practical for low power and low data rate application domains. A Zigbee device in the WSN can be configured to be an end device, router, and coordinator [16]. In this work, we implemented a WSN that contains two end nodes and one coordinator. The XBee modules attached with Arduino are configured to be the end devices, while the XBee module connected to Raspberry Pi is configured to be the coordinator.

### C. The processing node

This is the second subsystem of the proposed framework. This subsystem consists of a Raspberry Pi, a coordinator XBee module, and a solenoid water electrical valve. The primary purpose of this node is to collect data from the remote devices, analyze them, and control the irrigation process.

(a) Raspberry Pi (b) XBee explorer dongle

Fig. 4: The processing node components

*1) Raspberry Pi:* We utilized a raspberry Pi 3 [17] to control the smart irrigation process. The main tasks are summarized as follows:

- Read data from the connected coordinator XBee module, which is collected from the remote wireless nodes.
- Python code is implemented to analyze the gathered sensors quantities.
- Based on the analyzed data, the general purpose input/output (GPIO) pins are utilized to control the electrical water valve. In other words, it will decide whether to irrigate or not.
- The collected data are transferred to cloud using ThingSpeak API.

It is worth mentioning that the Raspberry Pi in our solution is considered as the fog computing node. As mentioned earlier it has the ability to analyze the data and provide the proper decision. This will handle the problems of cloud-based computing, such as high latency and security issues. Due to the limited resources available in the Raspberry Pi, we transfer the data to the cloud for storing, visualizing, and analyzing the vast amount of generated data.

*2) The coordinator XBee and explorer dongle:* We used one XBee module as a coordinator. The main purpose is to collect data sent from the remote end devices. It is connected directly to the raspberry pi by using the XBee explorer dongle, as shown in Figure 4(b).

*3) Solenoid electrical valve:* In our prototype, we used a 12v solenoid valve to control the flow of water. However, the Raspberry Pi can provide a maximum of 5 volts. To handle this issue, we used a 2-channel relay module.

### D. The Cloud computing unit (ThingSpeak IoT cloud)

Since there is a massive amount of generated data, analyzing and storing it is a significant challenge. The IoT makes the ability of providing and exchanging different physical data efficiently possible. This is achieved by connecting electronic sensors used in physical devices to the internet [10].

To improve the effectiveness of the proposed system and to benefit from the cloud services, we used *ThingSpeak* IoT cloud. *ThingSpeak* is an IoT analytical platform cloud service that can be utilized to aggregate, visualize, and analyze live data streams in the cloud. The REST API calls such as GET, POST, PUT, and DELETE, can be used to create and update ThingSpeak channels and charts (update an existing channel, clear a channel feed, and delete a channel) or retrieve data from channels. In this project, since the Raspberry Pi is connected directly to a stable power supply, and we have enough resources, we used the REST API to send data. In summary, we utilize ThingSpeak cloud service for three main purposes:

- Store and visualize the received data from the Raspberry Pi using different types of charts.
- Provide farmers and interest persons with a live streaming platform to monitor the status of the plants, create instance visualization, and sending alerts.
- The gathered data can be utilized to extract hidden and valuable knowledge by employing Data Mining and Machine Learning techniques. This helps farmers and others for planning and decision-making. It is also helpful for early prediction purposes.

### *E. Addressing the major security issues*

Protecting the system from intruders is becoming very important challenge that should be considered when designing WSN and IoT solutions [13]. In WSNs, various threats may pose due to several security issues. These issues are at the hardware and software levels. The deployment nature of sensor networks makes them more vulnerable to various attacks [18]. For instance, in our proposed WSN, the moisture sensor employed in the sensor node should be placed in the soil that makes it highly prone to capture and physical vandalism. To handle the majority of security issues, our proposed solution have already some efficient techniques, and we will suggest other solutions for future application and to make the system safer. Regarding the physical/hardware security, we suggest employing a surveillance system based on cameras and motion detection techniques to protect and observe the whole farm. At the software level, we used the XBee ZigBee module to provide efficient strategies that will prevent most of the attacks. It can ensure the safe delivery of data to the targeted node using addressing, acknowledgments, and retries. The device does not process frames sent through the serial interface with incorrect checksums and ignores their data. Moreover, ZigBee offers encryption services, authentication, and application services such that each node belongs to a predefined cluster and can take a predefined number of actions. These techniques are useful against passive information gathering, message corruption, and traffic analysis attacks.

## IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

### *A. Design and implementation*

The proposed prototype was designed and implemented using different hardware and software components as follows:

- **Hardware components**: 1x Raspberry Pi 3, 2x Arduino Uno, 2x Moisture sensor YL-69, 2x Humidity and temperature sensor DH22/AM2302, 3x XBee RF series 1 (2 end devices and 1 coordinator), 1x XBee explorer dongle, 2x 9v and 1x 12 v batteries, 2-channel relay module, 2x solenoid electrical valve 12v.
- **Software and protocols**: XCTU, Raspbian Operating System, Arduino IDE, Python 3, ZigBee communication protocol, ThingSpeak IoT platform, XBee libraries

The Arduino is programmed using IDE and C++ programming language, while Raspberry Pi was programmed using Python programming language. XCTU platform was used to setup the configuration for XBee modules such as PAN, source and destination 16-bit addresses, names, and operation modes. The main configurations for the employed XBee modules are reported in Table I. The performed steps by the sensing node and control unit are presented in Figures 5 and 6, respectively.

TABLE I: XBee modules configurations

| parameter | coordinator | Node 1 | Node 2 |
|---|---|---|---|
| CH channel | C | C | C |
| PAN ID | 1001 | 1001 | 1001 |
| CE coordinator enable | Coordinator | End point | End point |
| My 16-bit source address | 4 | 3 | 1 |
| Baud rate | 9600 | 9600 | 9600 |
| Destination address | FFFF | 4 | 4 |
| Mode | API | API | API |

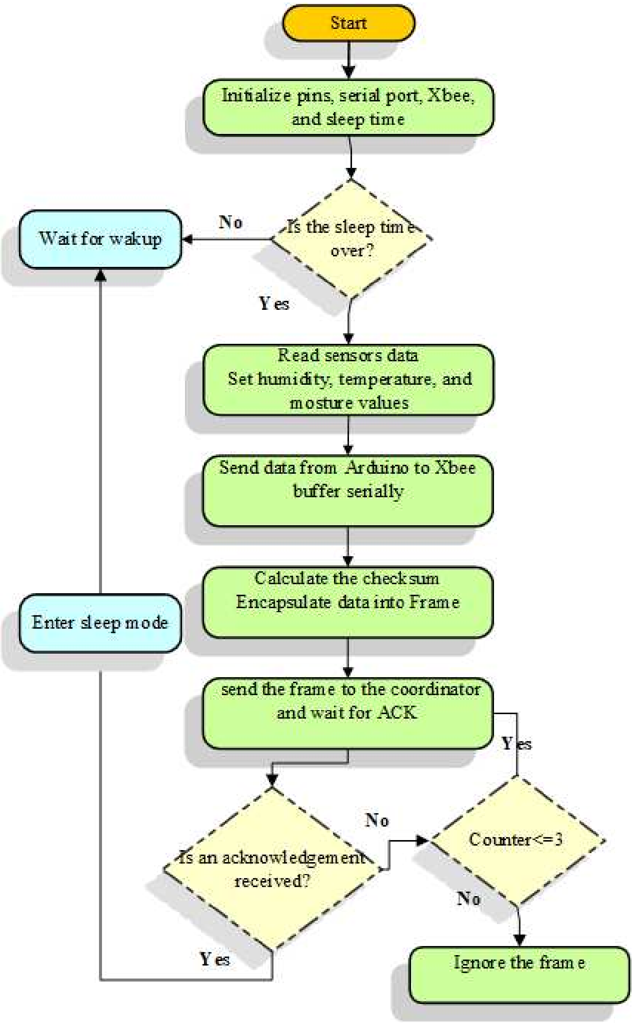


Fig. 5: Sensor node process

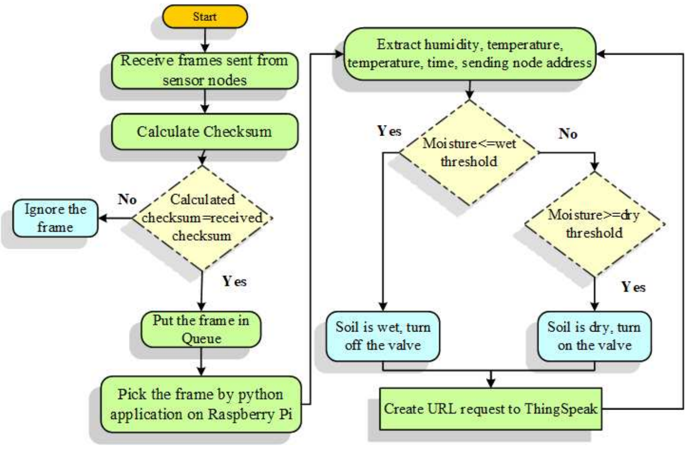


Fig. 6: control unit process

### B. Experimental results

At this phase of the project, we were satisfied with verifying that the proposed model is working properly in all its parts. The investigated prototype is shown in Figure 7. Figure 8 shows a sample of the live data streams visualization obtained from *ThingSpeak* platform over a period of time.

Fig. 7: The proposed irrigation system prototype

(a) Humidity

(b) Moisture

Fig. 8: Charts obtained from ThingSpeek cloud platform

## V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, an efficient cloud-based IoT framework for controlling and monitoring the irrigation process was designed and implemented. The main objective is to save water and improve the quality of crops. The proposed system was implemented based on WSN, Raspberry Pi, and ThingSpak Iot cloud platform. It was obvious from the implemented prototype that the WSN integrated with ThingSpeak platform provides more flexibility in monitoring and managing the system.

For future work, extensive experiments will be conducted on real crops and analyze the resulting data using machine learning to obtain useful recommendations. A comparison can be performed with the traditional irrigation systems in terms of water consumption and the quality of crops to ensure the effectiveness of the proposed model. Moreover, the system can be adapted to consider more environmental data by adding different types of sensors.

## REFERENCES

[1] A. Hassan, S. Sheng, W. Md Shah, and N. Bahaman, *An Automated Irrigation System Using Arduino Microcontroller*, 12 2018, pp. 3–13.

[2] K. Taneja and S. Bhatia, "Automatic irrigation system using arduino uno," in *International Conference on Intelligent Computing and Control Systems (ICICCS)*, 06 2017, pp. 132–135.

[3] AQUASTAT. Fao's global information system on water and agriculture. [Online]. Available: http://www.fao.org/aquastat/en/overview/methodology/water-use

[4] worldbank. water agriculture. [Online]. Available: https://www.worldbank.org/en/topic/water-in-agriculture

[5] L. Seyfi, E. Akman, and T. Topak, "Smart irrigation system," in *XIII International Conference on Electrical, Computer, Electronics and Communication engineering*, 01 2015.

[6] S. Bitla, S. Santhan, S. Bhagat, A. Pandey, and V. Nath, "Smart irrigation system: A review," in *Nano electronics, Circuits and Communication Systems*, V. Nath and J. K. Mandal, Eds. Singapore: Springer Singapore, 2020, pp. 569–578.

[7] F. Okoye, E. Orji, and G. Ozor, "Using arduino based automatic irrigation system to determine irrigation time for different soil types in nigeria," *international journal of advanced research in computer and communication engineering*, vol. 07, pp. 42–47, 07 2018.

[8] U. Nations, *The Sustainable Development Goals Report 2019*, 2019. [Online]. Available: https://www.un-ilibrary.org/content/publication/55eb9109-en

[9] M. Anbarasi, T. Karthikeyan, L. Ramanathan, S. Ramani, and N. Nalini, "Smart multi-crop irrigation system using iot," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 7, 2019.

[10] 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*, 2018, pp. 1–6.

[11] N. Fawzi and A. Jalal, "Design and implementation of smart irrigation," *International Journal of Scientific & Engineering Research*, vol. 8, no. 4, 2017.

[12] H. K. Patil and T. M. Chen, *Wireless Sensor Network Security*, 12 2017.

[13] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, 07 2012.

[14] Am2303. [Online]. Available: https://datasheetspdf.com/pdf/845488/Aosong/AM2303/1

[15] DiGi. Digi xbee rf modules. [Online]. Available: https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules

[16] P. Dhillon and H. Sadawarti, "A review paper on zigbee (ieee 802.15.4) standard," *International journal of engineering research and technology*, vol. 3, 2014.

[17] J. Newmarch, *Linux Sound Programming*, 01 2017.

[18] A. Alharbi, "Security issues in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 10, pp. 1–5, 02 2017.