Journal of Computer and Communications, 2025, 13(3), 1-20
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

LoRa-Based Smart Agriculture Monitoring and Automatic Irrigation System

Kayembe Sagali Enock[1], Mungusa Jonas Sagali[2], Umba Ilunga Jeannick[3], Daru Chen[4]

[1]School of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China
[2]School of Mechanical Engineering and Automation, Wuhan Textile University, Wuhan, China
[3]School of Mathematics and Computer Science Technology, Zhejiang Normal University, Jinhua, China
[4]Hangzhou Institute of Advanced Studies, Zhejiang Normal University, Hangzhou, China
Email: enocksagali@zjnu.edu.cn, jonas1sagali@gmail.com, jeannickumba94@gmail.com, daru@zjnu.cn

How to cite this paper: Enock, K.S., Sagali, M.J., Jeannick, U.I. and Chen, D. (2025) LoRa-Based Smart Agriculture Monitoring and Automatic Irrigation System. *Journal of Computer and Communications*, **13**, 1-20.
https://doi.org/10.4236/jcc.2025.133001

Received: January 15, 2025
Accepted: March 2, 2025
Published: March 5, 2025

Abstract

Agriculture is a sector that plays a crucial role in ensuring food security and sustainable development. However, traditional agriculture practices face challenges such as inefficient irrigation methods and lack of real-time monitoring, leading to water waste and reduced crop yield. Several systems that attempt to address these challenges exist, such as those based on Wi-Fi, Bluetooth, and 3G/4G cellular technology; but also encounter difficulties such as low transmission range, high power consumption, etc. To address all these issues, this paper proposes a smart agriculture monitoring and automatic irrigation system based on LoRa. The system utilizes LoRa technology for long-range wireless communication, Blynk platform for real-time data visualization and control, and ThingSpeak platform for data storage, visualization, and further analysis. The system incorporates multiple components, including a sensor node for data collection, a gateway for data transmission, and an actuator node for irrigation control. Experimental results show that the proposed system effectively monitors collected data such as soil moisture levels, visualizes data in real time, and automatically controls irrigation based on sensor data and user commands. The system proposed in this study provides a cost-effective and efficient solution for sustainable agriculture practices.

Keywords

Smart Agriculture, Internet of Things, LoRa, Power Consumption, Real-Time Monitoring

1. Introduction

Agriculture is currently facing many difficulties. In addition to the rising demand for sustenance as the world's population grows, less land is available for cultivation. Furthermore, complying with tighter sustainability rules has sometimes led to increased use of fertilizers and plant protection products. Increased levels of automation are required due to decreased agricultural product margins, a lack of qualified employees, and rising costs. Climate-related extreme weather events like droughts and torrential rains further complicate cultivation.

Using conventional technology for fertilizing and harvesting fields cannot solve these challenges. Smart Agriculture, or Smart Farming, is necessary. It uses advanced information and communication technologies such as the Internet of Things (IoT), sensors, location systems, robots, and artificial intelligence to enhance product quality and quantity while reducing human labor. The goal is to turn sensor data into useful information, simplify agricultural work, and facilitate precise and predictive analysis of weather and economic conditions. Smart agriculture also supports efficient resource management and enables information exchange between farms, creating a network accessible from smartphones or computers.

The use of IoT and the need for a fully automated system in agriculture are increasingly important, especially in countries like China and the U.S., where farming is done over large areas requiring constant crop monitoring. Without such a system, manual labor is expensive and often suboptimal. This automated system facilitates crop monitoring using sensors like humidity, temperature, and soil moisture, transmitting data remotely to a cloud platform for analysis. Short-range radio technologies like Wi-Fi or Bluetooth are typically used due to their popularity and ease of use. However, these are unsuitable for large-scale scenarios needing high-range data transmission [1].

Cellular technology solutions such as 3G/4G provide the necessary range but consume too much power [2]. Low Power Wide Area (LPWA) technologies including NB-IoT, Sigfox, and LoRa have emerged as the most suitable solution for smart agriculture due to their long-range transmission, low-cost implementation, and low power consumption [3]. Among these three LPWA technologies, LoRa appears to be the best choice for smart agriculture due to its several advantages such as its capacity to support a large number of devices simultaneously, its flexibility to use a variety of applications in smart agriculture such as monitoring soil moisture, tracking livestock, and controlling irrigation systems. It is also the only one with no subscription fee and is chosen as the main communication protocol for this proposed system.

2. Related Works

The development of the Internet of Things (IoT) has initiated a technological revolution in agriculture. Connected objects are essential tools for modern and future farming, addressing the challenges of feeding a projected global population of over 9.8 billion by 2050, amid issues such as agricultural labor shortages, drought, and global warming. IoT empowers farmers to enhance their operational efficiency by automating and optimizing production processes, while offering significant benefits for farm management. These advances in IoT technology have inspired researchers to develop innovative solutions aimed at improving agricultural practices and tackling the complex issues faced by the sector. Although IoT has made remarkable advancements in agriculture and irrigation, much of the established research remains theoretical. However, some recent works have implemented smart agriculture and irrigation systems with tangible outcomes. Below, we review studies that are closely related to our proposed system.

2.1. IoT-Based Systems Using Wi-Fi Communication

In the work of Seenu *et al.* [4], a smart agriculture and automated irrigation system was developed to optimize water usage for farming purposes. The system uses Wi-Fi modules to establish communication between an Arduino microcontroller and a mobile application, enabling users to monitor sensor data and control water pumps through a relay module. Soil moisture thresholds are pre-programmed into the microcontroller to determine the required irrigation levels for specific soil types. While users retain manual control over irrigation timing, the system sends notifications based on temperature, soil moisture percentage, and humidity data, analyzed by the embedded algorithm, to recommend irrigation when needed. The mobile application serves as the user interface, displaying real-time data collected from the microcontroller via the Wi-Fi connection and allowing users to send commands to the Arduino for irrigation control. Users can also override the system and manually deactivate water irrigation at any time. This system is cost-effective compared to other alternatives, utilizing widely available Wi-Fi networks for communication between the microcontroller and the user's device. Its accessibility is enhanced by the prevalence of home routers, making it a practical option for small to medium-sized agricultural applications. However, the system's reliance on a one-to-one communication model restricts scalability, as it cannot accommodate multiple microcontrollers within a single network. Additionally, the use of Wi-Fi limits the operational range of sensors without the integration of a mesh network, rendering the system less practical for large-scale agricultural projects. Despite its security and convenience for short-range communication, the lack of long-range connectivity hinders its applicability in expansive farming operations.

2.2. GSM-Based Automated Irrigation Systems

Karan Kansara *et al.* [5] proposed an automated irrigation system that leverages the Global System for Mobile Communications (GSM). The system incorporates multiple sensor nodes that monitor environmental parameters and control water valves for irrigation. These nodes are organized in a star-shaped mesh network, with all nodes connected to a central node. The central node serves as a hub, transferring collected data to a computer via a serial connection. One of the nodes, designated as the actuator node, receives commands from the central node to activate valves in areas requiring irrigation based on sensor data. The system measures critical parameters such as soil moisture, air and soil temperature, humidity, solar radiation, and wind speed to determine optimal water levels. When the detected levels fall below the calculated threshold, the sensor nodes communicate this information to the central node, which processes the data and instructs the actuator node to activate the irrigation valves in the necessary areas. The system's ability to connect multiple sensor nodes allows it to cover larger agricultural areas, making it more versatile than alternatives. Additionally, GSM provides reliable and secure data transmission compared to Wi-Fi. However, the system's reliance on GSM technology introduces challenges. It requires high power consumption and a self-sustaining electricity source for continuous operation, which may limit its feasibility in rural regions lacking GSM tower coverage or stable energy infrastructure. Despite these limitations, the system represents a significant step forward in scalable and automated irrigation management.

2.3. Systems Utilizing XBEE Protocols and Cloud Platforms

Valecce Giovanni *et al.* [6] designed a smart irrigation system that uses XBEE modulation for communication between sensor nodes and a central node. The system connects to the cloud via an MQTT broker and utilizes XBEE's low-interference frequency band, enabling reliable long-distance communication (up to 1.6 km). Additionally, the system incorporates a multi-hop mesh network to extend its coverage area and reduce power consumption through solar-powered sensor nodes. A Raspberry Pi serves as the MQTT broker, connecting to Amazon Web Services (AWS) for cloud storage and data visualization. Despite its benefits, the cost of XBEE modules and the system's complexity may pose challenges for widespread adoption.

2.4. Long-Range Low-Power Systems Using LoRaWAN Technology

In recent years, the adoption of low-power wide-area network (LPWAN) technologies such as LoRaWAN has revolutionized smart agriculture by addressing critical limitations of earlier systems, including high-energy consumption, short communication ranges, and limited scalability. LoRaWAN is particularly well suited for agricultural applications requiring long-distance communication with minimal energy usage, offering a cost-effective and efficient solution for large-scale monitoring and automation. Xiaoming Ding *et al.* [7] proposed an agricultural data monitoring system using LoRa technology. The system leverages LoRa wireless modules for data transmission between field sensor nodes and sink nodes, with further data relayed to servers via 4G network. This setup achieves wireless data acquisition, transmission, and remote monitoring, significantly extending communication ranges while maintaining low energy consumption. By employing LoRa for long-range communication, the system effectively addresses challenges associated with conventional wireless networks, such as their inability to combine extended range with low power usage. Similarly, Wenju Zhao *et al.* [8]

developed a smart irrigation system integrating LoRaWAN technology for cloud-based monitoring and control. This system enables irrigation devices to communicate with the cloud platform through a gateway, providing real-time control capabilities via application programming interfaces (APIs). Additionally, the solenoid valve in the irrigation node is powered by a hydroelectric generator, demonstrating a novel approach to achieving energy efficiency and sustainability. The study highlights the advantages of using LoRaWAN for remote agricultural management, including reduced deployment complexity and enhanced scalability. Recent research also emphasizes the energy efficiency and coverage capabilities of LoRa and LoRaWAN. Muhammad Shoaib Farooq *et al.* [9] evaluated these technologies by proposing an energy consumption model to estimate the power requirements of individual sensor node components. Their analysis provides valuable insights into optimizing energy usage, making LoRaWAN an even more practical choice for IoT-based agricultural systems. Additional studies confirm LoRa's performance under diverse conditions. Achmad Fauzi Rachmani *et al.* [10] demonstrated successful data transmission over 2.5 km in rural environments using LoRa devices equipped with 2.2 dB omnidirectional antennas, achieving a packet error rate below 20%. Taoufik Bouguera *et al.* [11] tested LoRa for starfruit cultivation, achieving a maximum coverage of 700 meters with an acceptable packet delivery rate of over 50%. Moreover, in [12], Kamal Elhattab *et al.* presented an autonomous, energy-efficient agricultural system that combines LoRa communication, ESP32 microcontrollers, and solar power. This system uses sensors for real-time monitoring of parameters like soil moisture, temperature, and pH, enabling precise irrigation control. The integration of Blynk mobile application provides users with real-time data visualization and remote control of irrigation system. The use of solar energy ensures its operability in off-grid locations, while LoRa technology allows long-range communication across vast agricultural fields, demonstrating significant improvements in resource management and sustainability. In [13], Artetxe *et al.* proposed a cutting-edge system combining fuzzy logic, LoRa communication, and cloud computing for precision irrigation. The study highlights how the integration of precipitation forecasts via cloud APIs enhances water management efficiency, leading to a 23.1% reduction in water usage compared to conventional methods. Furthermore, the system's design emphasizes real-time control and predictive analytics, illustrating the transformative potential of combining IoT, artificial intelligence, and LPWAN technologies for sustainable agriculture.

These advancements illustrate the growing preference for LoRa-based systems in smart agriculture. Unlike traditional approaches such as GSM, Wi-Fi, and XBEE, which are often limited by range, energy demands, or complexity, LoRaWAN offers a robust, scalable solution. Its benefits include long-range communication, low power consumption, and cost-effectiveness, making it an ideal choice for large-scale remote crop monitoring and irrigation systems. Our proposed system builds on these developments, integrating LoRa technology for long-range wireless communication, the combination of two cloud platforms, Blynk and ThingSpeak. Blynk is used to visualize real-time data and control systems remotely, while ThingSpeak is used to store data and perform predictive analytics to optimize irrigation. By combining these technologies, our solution contributes to more efficient and sustainable agriculture.

3. Methodology

3.1. Research Design and Approach

As mentioned previously, this proposed system aims to address challenges encountered in conventional agricultural practices, such as inefficient irrigation methods and lack of real-time monitoring. Another goal is to overcome the limitations of existing systems such as Wi-Fi or Bluetooth-based systems (limited range) and 3G/4G-based systems (high-energy consumption). For this reason, the proposed system uses LoRa technology as the main means of communication. LoRa is a wireless radio communication technique mainly used in the Internet of Things. It uses Chirp Spread Spectrum modulation technology, a technology initially used in the space and military fields since it is very resistant to interference, offering long-range wireless communication and low energy consumption. Unlike short-range Wi-Fi/Bluetooth networks or standard 3G/4G cellular networks that allow large amounts of data to be sent, LoRa sends small-sized data, with a flexible rate of 0.3 to 50 KB per second. In addition, these data can be transmitted over longer distances, between 1 and 5 km in urban areas, and up to 15 km in rural areas.

In smart agriculture monitoring, short-range Wi-Fi/Bluetooth and 3G/4G networks offer reliable connections but are limited by range and high power consumption, requiring sensor nodes to be near power sources. LoRa connects sensing nodes to the gateway, creating a network capable of long-distance operation with minimal interference. LoRa modules consume low power, extending battery life and allowing longer network connectivity. The gateway connects to the Internet via 4G, Ethernet, or Wi-Fi, enabling easy data transmission from sensor nodes. A comparison of short-range, cellular, and low-power wide-area networks is shown in Figure 1.

3.2. System Architecture

The proposed system architecture, as shown in Figure 2, consists of three main components: devices, cloud, and application. The general working principle of the system is as follows: environmental information from sensors such as temperature, humidity, soil moisture, rain, gas, and light is collected by the node and transmitted to the gateway using LoRa wireless communication technology. The gateway then transfers and stores the received information in the Blynk and ThingSpeak clouds. End users can access this data via the Web or Android/iOS. Additionally, the users can control the actuator node for irrigation purposes by sending commands through the application.

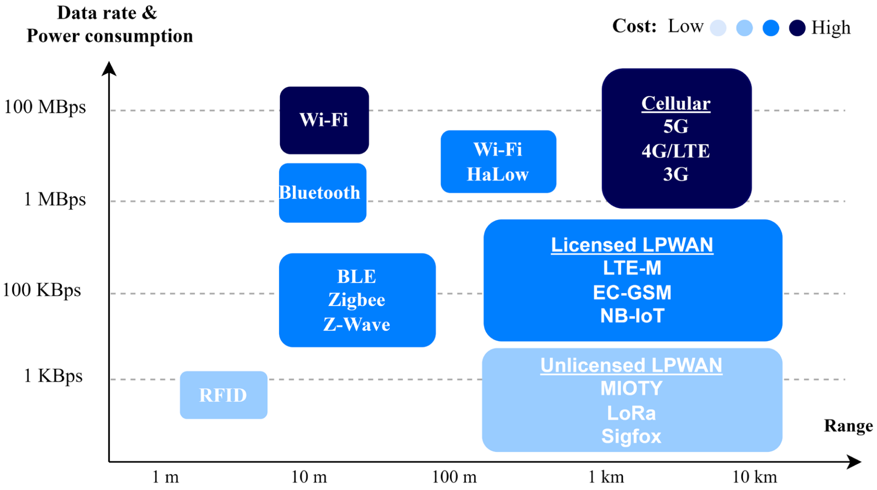


Figure 1. Comparison between LoRa and other wireless technologies [14].

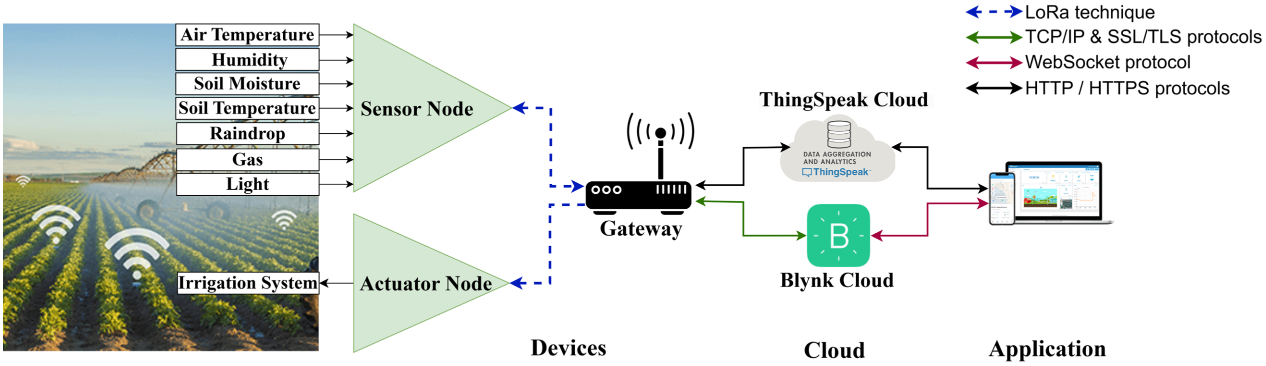


Figure 2. Illustration of system architecture.

3.2.1. Devices

There are three kinds of devices in this proposed system: sensor node, actuator node, and gateway. The sensor node is responsible for collecting data from sensors deployed in the agricultural field and transmitting it wirelessly through LoRa to a gateway and then to the clouds for processing and analysis. The sensor node monitors various parameters such as temperature, humidity, soil moisture, rain, gas, light intensity, and other environmental factors affecting plant growth and development. The actuator node is an important device for automating irrigation and other processes based on sensor readings. Its main function in this proposed system is to control the physical environment of the agricultural field by automatically turning on or off devices such as water pumps/sprinklers, valves, or other equipment based on sensor readings. By controlling physical devices based on sensor readings, the actuator node ensures that crops receive the right amount of water at the right time, which can improve crop yields and reduce water waste. The gateway as a relay node serves as a central communication hub that connects sensor/actuator nodes to the internet or cloud server. The main function of the gateway in this proposed system is first to receive data from the sensor node and transmit it to the cloud/servers for further processing and analysis, then send control commands to the actuator node for irrigation purposes.

3.2.2. Cloud

The received data must be brought into the internet environment to ensure that the end user can access the information received by the gateway or send the control commands to the actuator node. This is accomplished by using simultaneously two IoT development platforms, Blynk and ThingSpeak. They provide cloud-based services for communication between web/mobile apps and connected devices, providing functions such as data storage, data synchronization, remote device control, user management, app configuration, etc.

3.2.3. Application

By using the web/mobile applications provided by Blynk and ThingSpeak IoT platforms, the end user can create a custom Graphical User Interface (GUI) to obtain and monitor the status of sensor and actuator nodes in the field. The user can add various widgets such as graphs, gauges, sliders, and charts to display real-time data on the parameters being monitored. The provided apps also allow the user to send control commands to the actuator node to turn on and off the water pumps for irrigation purposes.

4. System Design and Implementation

The principle block diagram of the LoRa-based smart agriculture monitoring and automatic irrigation system is shown in Figure 3. The diagram shows how hardware and software components interact together and also how information is exchanged between them.

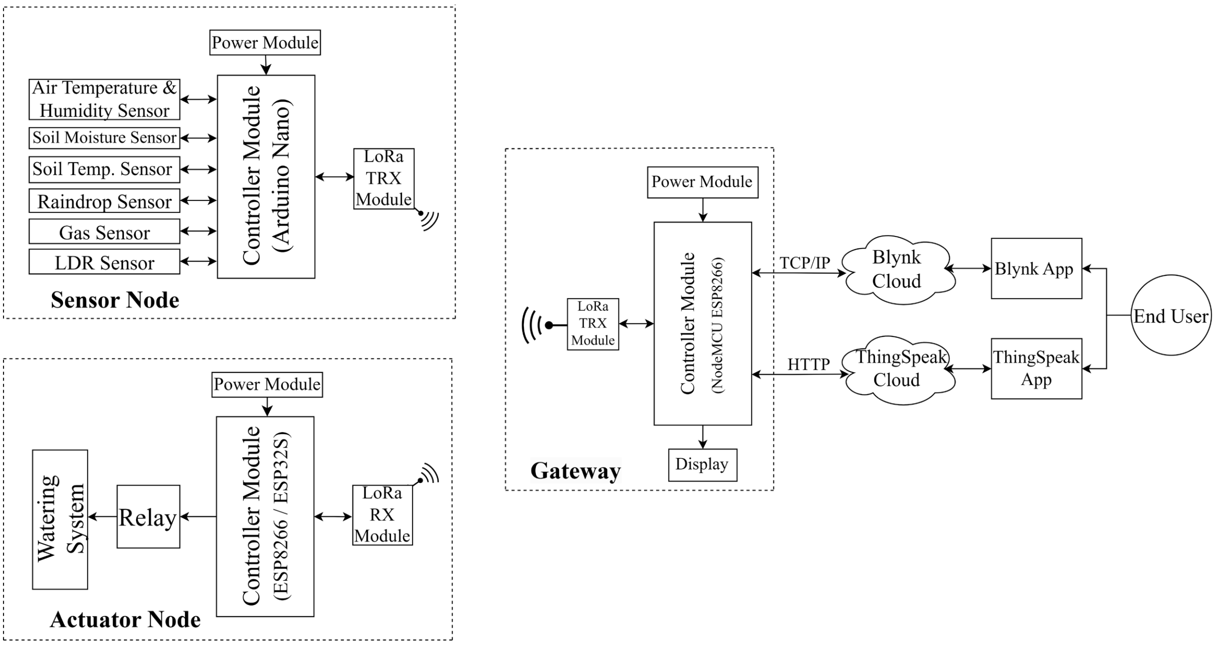


Figure 3. Block diagram of the whole system.

4.1. Hardware System

The system's hardware components consist of three parts: sensor node, actuator node, and gateway.

4.1.1. Sensor Node

As explained in Section 3, the sensor node, as a whole, is responsible for collecting data from the environmental sensors and transmitting it wirelessly via LoRa to the gateway. Following **Figure 3**, the sensor node contains four modules: sensor connectors, controller module, transceiver module, and power module. The prototype of the sensor node is shown in **Figure 4**.

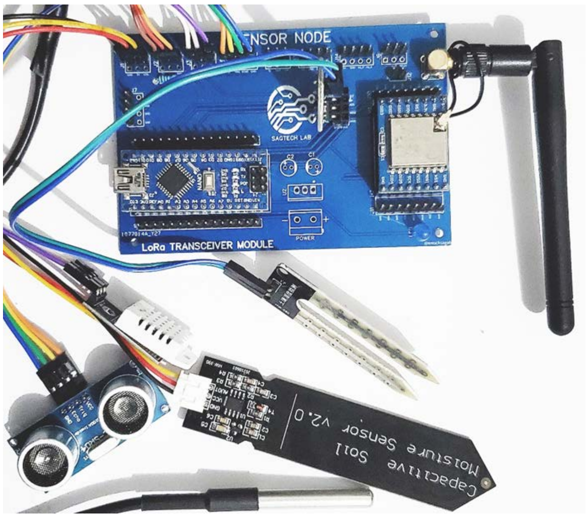


Figure 4. Prototype of sensor node.

- Sensor connectors: the DHT22 sensor, capacitive soil moisture sensor, DS18B20 sensor, MH-RD sensor, MQ-6 sensor, and GL5528 sensor, are connected to the node and are respectively used to measure the air temperature/humidity, the water content of the soil, the soil temperature, the raindrops, the gas level/air quality, and the light intensity. The measured data is then sent to the controller module for processing.
- Controller module: based on the ATmega328P microcontroller unit, the Arduino Nano is used as the controller module to realize all control functions in the sensor node. The module acts as the brain of the sensor node, orchestrating all the operations and facilitating communication with the gateway. It includes various components and connectors that serve different functions. The ATmega328 microcontroller runs the Arduino firmware, executes the code, and controls the sensors connected to the digital/analog input/output (I/O) Pins. The Arduino Nano controller also includes a reset button, which can be pressed to reset the microcontroller and restart the execution of the code.
- Transceiver module: the Ra-02 LoRa module based on the Semtech SX1278 chip is used as the receiver and transmitter module to receive request signals from the gateway and send the data processed by the controller module. This

transceiver module operates in the unlicensed 433MHz ISM (Industrial-Scientific-Medical) frequency band. The serial peripheral interface (SCK, MISO, and MOSI), the chip select (NSS), and the RESET pins of the module are connected to the corresponding pins on the Arduino Nano to establish communication between the two modules.

- Power module: the sensor node is powered by a lithium-ion battery. The module includes charging circuitry to recharge the battery from an external source like an AC adapter or a solar panel. Note that the sensor node can also be powered by connecting a USB power source to the Arduino Nano controller.

4.1.2. Actuator Node

The actuator node contains five modules: the receiver module, controller module, relay module, irrigation system, and power module. The Ra-02 LoRa module is used as the receiver module to receive control commands from the gateway. The ESP8266 and ESP32S are used as the controller modules to process data from the LoRa module and perform all control functions, including turning on and off the relay and irrigation system based on the received data. The actuator node can be powered by a lithium-ion battery or by connecting a USB power module source to the ESP8266 or ESP32S board. The prototype of the actuator node is shown in **Figure 5**.

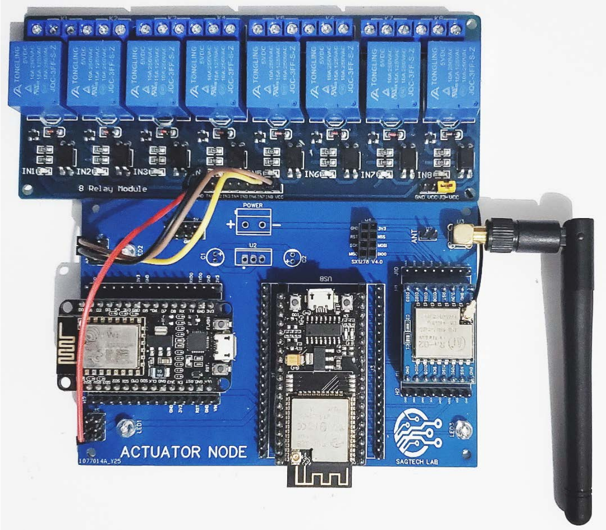


Figure 5. Prototype of actuator node.

4.1.3. Gateway

The gateway is used for receiving and sending LoRa packets to the sensor/actuator nodes, and for communicating with the clouds. The gateway consists of four modules: the Ra-02 LoRa transceiver module, the controller module, and the power module, along with a display. The Ra-02 module is responsible for receiving and sending LoRa packets. The NodeMCU ESP8266 is used as the controller module to perform all the control functions, and it has built-in Wi-Fi capability that allows sending data to the cloud and receiving commands from the cloud. The gateway is designed and programmed to establish communication with the sensor node within the first 5 seconds of a 10-second timeframe, by sending multiple requests and receiving data packets as responses. Subsequently, the gateway transmits the received data from both the sensor node and the cloud to communicate with the actuator node for the remaining 5 seconds. The gateway repeats the same process after this time interval. The prototype of the gateway is shown in **Figure 6**.

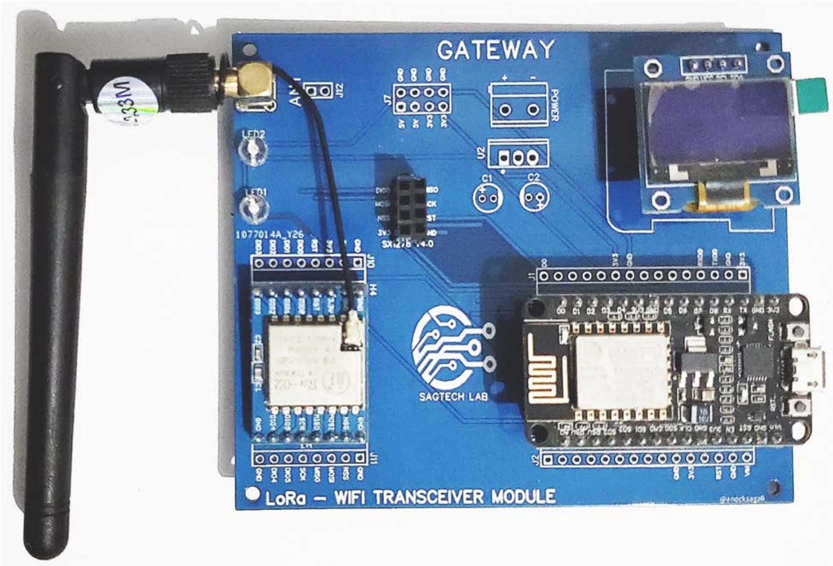


Figure 6. Prototype of gateway.

4.2. Software System

The software infrastructure of our proposed system consists of two main components: the Cloud and the Application.

4.2.1. Cloud

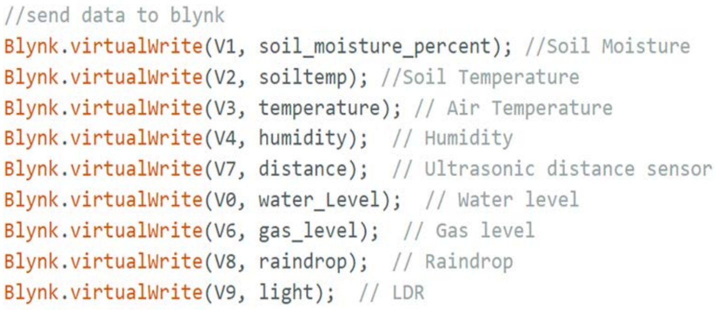
We simultaneously use two cloud platforms, ThingSpeak and Blynk, to enable remote access, control, and device management. The ThingSpeak cloud communicates with our gateway using a RESTful API, exchanging data in JSON or XML format. We start by signing up and creating an account on ThingSpeak. We then create a new channel, naming it and specifying the fields for data storage. After setting up the channel, ThingSpeak generates an API Key for authentication. Next, we download and install the ThingSpeak library for Arduino IDE, which includes the necessary functions to communicate with the ThingSpeak cloud. In the gateway's code, we include the library, configure the connection with the API Key, and set up network credentials (Wi-Fi SSID and password) to connect to the internet. The *ThingSpeak.begin(client)* function initializes the library, and data is sent to the ThingSpeak cloud in string format, as illustrated in **Figure 7**.

The Blynk cloud communicates with our gateway using standard internet protocols such as TCP/IP and SSL/TLS, ensuring secure communication. The gateway device integrates the Blynk library, allowing easy connection to the Blynk cloud server. Here is how we establish communication:

```
// Send data to ThingSpeak
if (millis() - lastConnectionTime > postingInterval) {
  lastConnectionTime = millis();
  if (client.connect("api.thingspeak.com", 80)) {
    String postData = "api_key=" + String(THINGSPEAK_API_KEY) +
                      "&field1=" + String(soil_moisture_percent) +
                      "&field2=" + String(soiltemp) +
                      "&field3=" + String(temperature) +
                      "&field4=" + String(humidity) +
                      "&field5=" + String(distance) +
                      "&field6=" + String(water_Level);
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postData.length());
    client.print("\n\n");
    client.print(postData);
    client.print("\n\n");
    //delay(100);
    client.stop();
    Serial.println("Data sent to ThingSpeak");
  } else {
    Serial.println("Failed to connect to ThingSpeak");
  }
}
```

Figure 7. Library function to send data to ThingSpeak Cloud.

First, we sign up and log in to the Blynk cloud via their website. We then create a new project by clicking "Create New Project" in our Blynk account dashboard, naming it, and selecting the appropriate hardware model (e.g., NodeMCU ESP8266). After creating the project, Blynk generates an "Auth Token" for authentication in the gateway device's code. The necessary Blynk library for Arduino IDE is downloaded and installed to provide the required functions for cloud communication. In the gateway's code, we include the Blynk library, configure the Blynk cloud connection using the Auth Token, and set up the same network credentials (Wi-Fi SSID and password) used for ThingSpeak to connect to the internet. We call the *Blynk.run()* function in the *loop()* function to handle communication. Functions like *Blynk.virtualWrite()* and *Blynk.virtualRead()* facilitate data exchange between the gateway sketch and Blynk cloud using virtual pins, as shown in **Figure 8**.



```
//send data to blynk
Blynk.virtualWrite(V1, soil_moisture_percent); //Soil Moisture
Blynk.virtualWrite(V2, soiltemp); //Soil Temperature
Blynk.virtualWrite(V3, temperature); // Air Temperature
Blynk.virtualWrite(V4, humidity);  // Humidity
Blynk.virtualWrite(V7, distance);  // Ultrasonic distance sensor
Blynk.virtualWrite(V0, water_Level);  // Water level
Blynk.virtualWrite(V6, gas_level);  // Gas level
Blynk.virtualWrite(V8, raindrop);  // Raindrop
Blynk.virtualWrite(V9, light);  // LDR
```

Figure 8. Library function to send data to Blynk Cloud.

4.2.2. Application

In addition to using cloud-based services from Blynk and ThingSpeak to manage, collect, and store data from connected devices, these two platforms provide mobile apps or web interfaces that allow remote monitoring and control of connected devices. Both the Blynk App and its Web Interface are used for this proposed system. Customizable dashboards are created to visualize and control devices. Data streams, which are like pipelines or channels through which data will be received or sent, are created. In this project, we receive ten pieces of data (soil moisture, soil temperature, air temperature, humidity, raindrop, gas level, light intensity, irrigation system status, water distance, and tank water level). This means that ten data streams are created. Widgets such as gauges, charts, switches, and indicators are added, and their properties are customized and linked to data streams to interact with connected devices.

The ThingSpeak web application is used for this proposed system to retrieve sensor data from the ThingSpeak channel using the ThingSpeak API, which provides RESTful endpoints for data access. Widgets such as charts and graphs are added to the account dashboard for visualizing the data sent from the gateway device. The application also offers additional features, such as data analysis, processing, and visualization using MATLAB, for further analysis of the data.

5. Results and Performance Evaluation

The LoRa-based smart agriculture monitoring and automatic irrigation system, which utilizes Blynk and ThingSpeak for data visualization, data storage, and further analysis, was evaluated through a series of experiments to assess its performance, reliability, and power consumption. The following are the key findings from the experimental results:

5.1. Real-Time Data Visualization

The sensor node of the system effectively collected environmental data, such as air temperature, humidity, soil moisture, light intensity, and gas levels. The collected data from the sensor node was transmitted wirelessly through LoRa to the gateway. Using Blynk and ThingSpeak apps/platforms as shown in **Figure 9** and **Figure 10**, we visualized and monitored the collected data in real-time, and stored it in a secure and organized manner allowing access to historical data for trends and further analysis. The system automatically controlled the irrigation process based on sensor data and user commands. The maximum threshold for air and soil temperature is set at 40˚C, and the minimum threshold for soil moisture is set at 30%, meaning that if the levels exceed or fall below these thresholds, the actuator node turns on the irrigation system to lower the temperature or increase the soil moisture. The actuator node also received control commands from the user through the Blynk app and executed irrigation actions accordingly.

Figure 9. Real-time data visualization on the Blynk app.

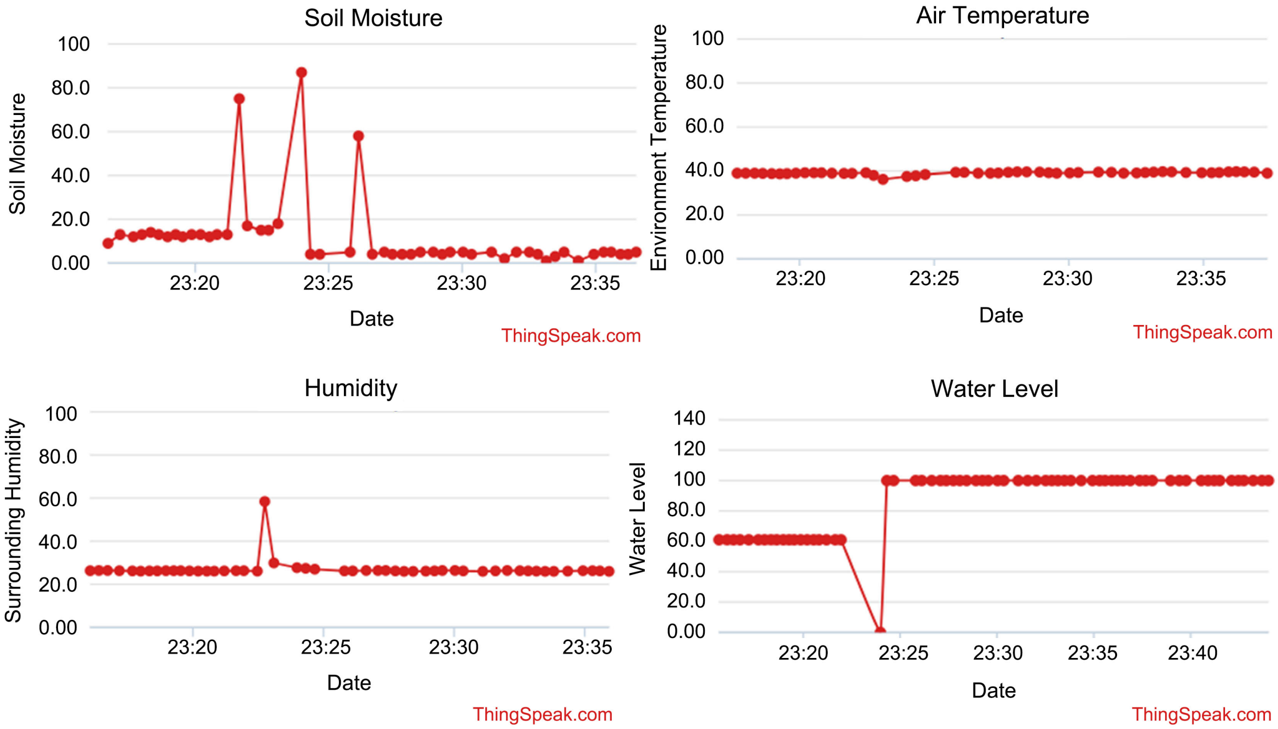


Figure 10. Data visualization on the ThingSpeak app.

5.2. Transmission Performance

The transmission range experiment was conducted in a closed area around Zhejiang Normal University in Jinhua City. The gateway was placed above a university building. The three system modules, *i.e.* the sensor node, the actuator node, and the gateway operate at the frequency band of 433 MHz. The main parameters set during the field tests are listed in **Table 1**.

Table 1. Node parameters setting.

Parameters	**Value**
Modulation Scheme	LoRa (Chirp Spread Spectrum)
Carrier Frequency	433 MHz
Transmit Power	20 dBm
Frequency Bandwidth	125 kHz
Spreading Factor	12
Coding Rate	4/5

Considering these parameters, the following factors are calculated to test the transmission performance of the system.

5.2.1. Data Rate

The rate at which data is transmitted between the sensor, actuator nodes, and the gateway is found using the following formula:

$$R_b = SF \times \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} \times 1000 \tag{1}$$

where R_b is the data rate in bits per second (bps). SF is the spreading factor, which determines the rate at which data is spread across the bandwidth. CR is the coding rate, which determines the amount of error correction coding applied to the data. BW is the frequency bandwidth of the LoRa signal in Hz.

The data rate with SF equals 12, CR 4/5, and BW 125 kHz is found and approximated to 3.4 kbps.

5.2.2. Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is a measure of the quality of the received signal, representing the ratio of the signal power to the noise power. A higher SNR generally indicates a better-quality signal with less noise, improving the receiver's ability to correctly decode the transmitted data.

The SNR is calculated by adopting the following formula:

$$\text{SNR}\left(\text{dB}\right) = 10\log_{10}\left(\frac{P_{Rx}}{P_N}\right) = P_{Rx} - P_N \tag{2}$$

where P_N is the noise power and is calculated using the following formula:

$$P_N\left(\text{dBm}\right)=10\log_{10}\frac{K\times B\times T\times F}{1mW} \tag{3}$$

where K is Boltzmann's constant (1.38e−23 J/K), T is the reference temperature in kelvin (usually taken as 290 K), B is the receiver bandwidth in Hz, and F is the noise figure of the receiver, expressed as a ratio.

P_{Rx} is the received power, which is calculated and approximated using the Friis transmission equation, which assumes ideal conditions, such as free space propagation with no obstacles, and neglects other factors such as interferences, and multi-path effects. The Friis transmission equation is given by the following formula:

$$P_{Rx}\left(\text{dBm}\right)=P_{Tx}+G_{Tx}+G_{Rx}+20\log_{10}\left(\frac{\lambda}{4\pi D_r}\right)+P_L \tag{4}$$

where P_{Rx} is the received power in decibel-milliwatts (dBm), P_{Tx} is the transmitted power in decibel-milliwatts (dBm), G_{Tx} is the transmitter antenna gain in decibel-isotropic (dBi), G_{Rx} is the receiver antenna gain in decibel-isotropic (dBi), λ is the wavelength of the transmitted signal (in meters), which can be calculated as the speed of light divided by the frequency (λ = c/f), c is the speed of light (approximately 3×10^8 m/s), f is the carrier frequency (in Hz), D_r is the distance between the transmitter and receiver (in meters), P_L is the path loss factor, expressed in decibels (dB).

The SNR experimental result summarized in Figure 11 shows that the SNR decreases as the distance between the sensor node (transmitter) and gateway (receiver) increases, indicating the impact of noise and distance on signal quality. It decreases by 12 dB in the first 500 meters and continues to slowly decrease in the last 6 km until the gateway loses the signal.

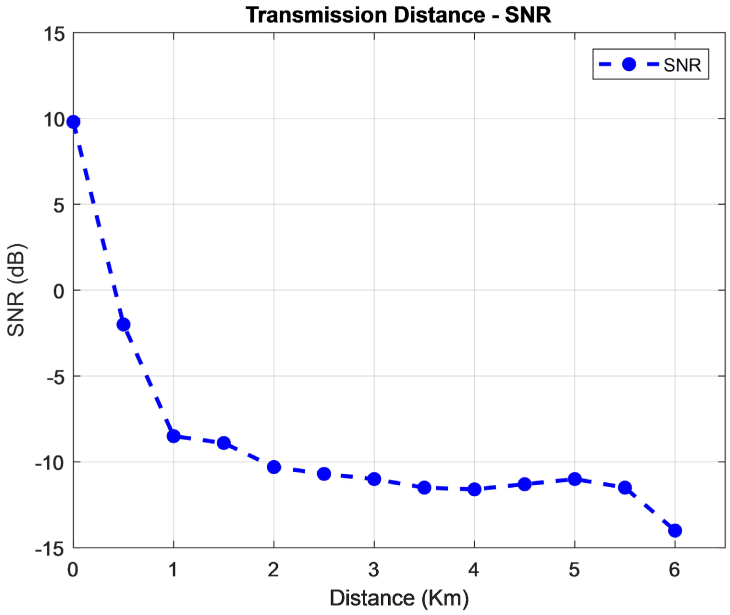


Figure 11. Transmission distance SNR.

5.2.3. Receiver Signal Strength Indicator

The Receiver Signal Strength Indicator (RSSI) is a measure of the strength of the received signal at the receiver. The RSSI is also represented in decibels (dB) or decibel-milliwatts (dBm) as it refers to power. In addition to the SNR, the RSSI is also used to estimate the quality of the received signal. The higher the RSSI value, the stronger the signal, while the lower RSSI value indicates a weaker signal.

The RSSI is calculated by adopting the following formula:

$$\text{RSSI}(\text{dBm}) = P_{Rx} + G_{Tx} + G_{Rx} - P_L \tag{5}$$

where P_{Rx} is the received power in decibel-milliwatts (dBm), which is calculated using the Friis transmission equation, G_{Tx} is the transmitter antenna gain in decibel-isotropic (dBi), G_{Rx} is the receiver antenna gain in decibel-isotropic (dBi), P_L is the path loss factor in decibels (dB), which is calculated by the following formula:

$$P_L(\text{dB}) = 20\log_{10}(d) + 20\log_{10}(f) + 32.45 \tag{6}$$

where f is the carrier frequency in MHz, d is the distance between the sensor node (transmitter) and the gateway (receiver) in kilometers.

The RSSI experimental result summarized in **Figure 12** shows that the RSSI decreases as the distance between the sensor node (transmitter) and gateway (receiver) increases, indicating the impact of distance on signal quality.

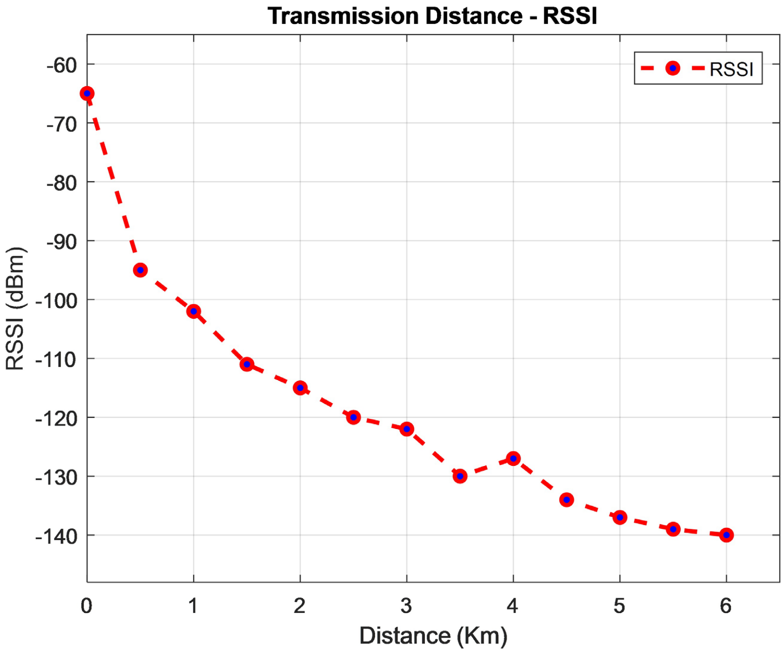


Figure 12. Transmission distance RSSI.

5.3. Power Consumption Performance

The power consumption performance of the system, in both active and deep sleep modes, was evaluated by comparing the power consumption of the three modules: sensor node, gateway, and actuator node. The experimental results in **Figure 13** show that deep sleep mode offers a significant reduction in power consumption for all three modules, with reductions of up to 86%, 88%, and 83%, respectively.

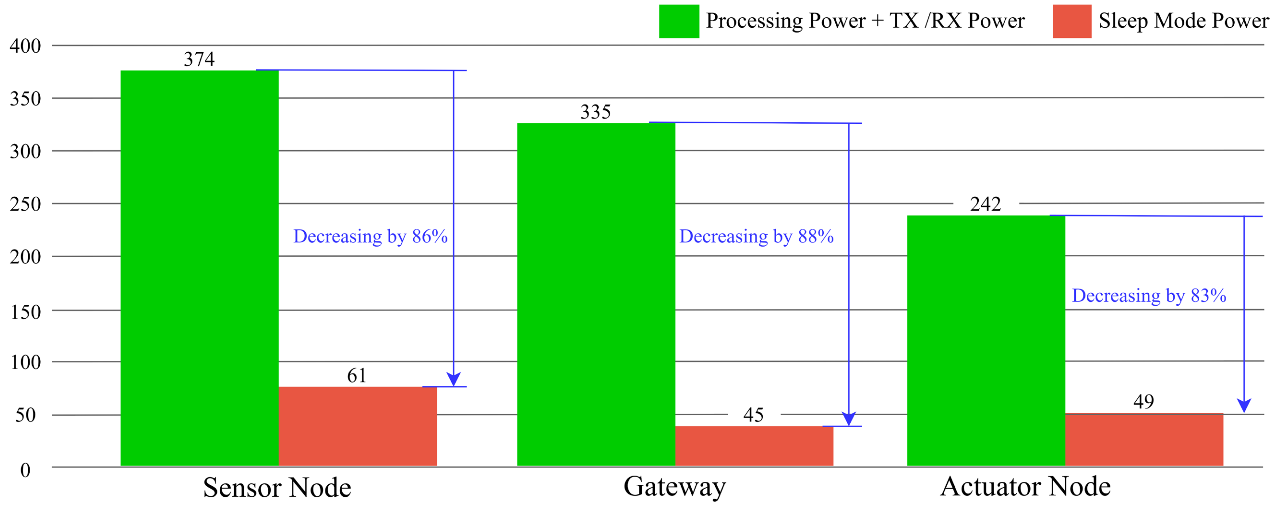


Figure 13. System power consumption performance (mW).

6. Conclusions

After conducting an in-depth study on the LoRa-based smart agriculture monitoring and automatic irrigation system, we can conclude that the proposed system effectively meets the objectives set in the introduction. In other words, the system addresses challenges found in conventional agricultural practices, such as inefficient irrigation methods and lack of real-time monitoring. Additionally, the system overcomes the limitations faced by existing systems, including those based on Wi-Fi or Bluetooth, which have limited range, as well as 3G/4G-based systems that have high power consumption. Experimental results demonstrate and validate the applicability of this system.

LoRa technology used as the main means of communication of this system provides reliable long-range wireless communication of more than 6 kilometers between the sensor/actuator nodes and the gateway, enabling remote crop monitoring and data collection. The use of the Blynk platform in this system provides an intuitive and user-friendly interface for real-time data visualization and control, while the ThingSpeak platform provides a robust data storage and analysis solution.

The integration of multiple components, including the sensor node, gateway, and actuator node, results in a highly efficient and automated irrigation system. The system automatically controls the irrigation process based on sensor data and user commands. The maximum threshold for air and soil temperature is set at 40˚C, and the minimum threshold for soil moisture is set at 30%, meaning that if the levels exceed or fall below these thresholds, the actuator node turns on the irrigation system to lower the temperature or increase the soil moisture. The actuator node also received control commands from the user through the Blynk app and executed irrigation actions accordingly. The experimental results also demonstrate that the system modules have low power consumption and a long lifetime. Overall, the proposed system presents a practical and efficient solution for modern agriculture, and it is expected to revolutionize the way farmers manage their crops in the future.

To improve the system in the future, a GSM module can be used instead of the Wi-Fi module, which will allow the gateway device to access the Internet more easily, thus avoiding the need for the gateway to be close to the Wi-Fi. By replacing or improving the LoRa communication module used on the gateway with a multi-channel module, several nodes can be connected simultaneously, which will allow network coverage over a very large crop area.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Chen Daru, for providing me with valuable guidance, support, and encouragement throughout my research. Without his expertise and insights, this work would not have been possible. Finally, I would like to express my gratitude to my parents, Marcel Sagali and Sidonie Mupanga, for their support and encouragement throughout my academic journey. Their love and encouragement have been a constant source of inspiration for me.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Song, Y., Lin, J., Tang, M. and Dong, S. (2017) An Internet of Energy Things Based on Wireless LPWAN. *Engineering*, **3**, 460-466. https://doi.org/10.1016/j.eng.2017.04.011

[2] Zou, L., Javed, A. and Muntean, G. (2017) Smart Mobile Device Power Consumption Measurement for Video Streaming in Wireless Environments: WiFi vs. LTE. 2017 *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting* (*BMSB*), Cagliari, 7-9 June 2017, 1-6. https://doi.org/10.1109/bmsb.2017.7986151

[3] Mekki, K., Bajic, E., Chaxel, F. and Meyer, F. (2019) A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment. *ICT Express*, **5**, 1-7. https://doi.org/10.1016/j.icte.2017.12.005

[4] Seenu, N., Mohan, M. and Jeevanath, V.S. (2018) Android Based Intelligent Irrigation System. *International Journal of Pure and Applied Mathematics*, **119**, 67-71. https://acadpubl.eu/jsi/2018-119-7/articles/7a/7.pdf

[5] Kansara, K., Zaveri, V., Shah, S., Delwadkar, S. and Jani, K. (2015) Sensor-Based Automated Irrigation System with IOT: A Technical Review. *International Journal of Computer Science and Information Technologies*, **6**, 5331-5333.

[6] Valecce, G., Strazzella, S., Radesca, A. and Grieco, L.A. (2019) Solarfertigation: Internet of Things Architecture for Smart Agriculture. 2019 *IEEE International Conference on Communications Workshops* (*ICC Workshops*), Shanghai, 20-24 May 2019, 1-6. https://doi.org/10.1109/iccw.2019.8756735

[7] Ding, X., Huang, J., Hu, S., Zhou, J. and Lu, C. (2022) Research on Distributed Agricultural Environment Monitoring System Based on Lora Technology. 2022 *IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference* (*IAEAC*), Beijing, 3-5 October 2022, 1064-1067. https://doi.org/10.1109/iaeac54830.2022.9930061

[8] Zhao, W., Lin, S., Han, J., Xu, R. and Hou, L. (2017) Design and Implementation of Smart Irrigation System Based on LoRa. 2017 *IEEE Globecom Workshops* (*GC Wkshps*), Singapore, 4-8 December 2017, 1-6. https://doi.org/10.1109/glocomw.2017.8269115

[9] Farooq, M.S., Riaz, S., Abid, A., Abid, K. and Naeem, M.A. (2019) A Survey on the Role of Iot in Agriculture for the Implementation of Smart Farming. *IEEE Access*, **7**, 156237-156271. https://doi.org/10.1109/access.2019.2949703

[10] Rachmani, A.F. and Zulkifli, F.Y. (2018) Design of IoT Monitoring System Based on Lora Technology for Starfruit Plantation. *TENCON* 2018: 2018 *IEEE Region* 10 *Conference*, Jeju, 28-31 October 2018, 1241-1245. https://doi.org/10.1109/tencon.2018.8650052

[11] Bouguera, T., Diouris, J., Chaillout, J., Jaouadi, R. and Andrieux, G. (2018) Energy Consumption Model for Sensor Nodes Based on LoRa and Lorawan. *Sensors*, **18**, Article 2104. https://doi.org/10.3390/s18072104

[12] Elhattab, K. and Abouelmehdi, K. (2024) Intelligent Agriculture System Using Low Energy and Based on the Use of the Internet of Things. *Bulletin of Electrical Engineering and Informatics*, **13**, 1286-1297. https://doi.org/10.11591/eei.v13i2.6346

[13] Artetxe, E., Barambones, O., Martín Toral, I., Uralde, J., Calvo, I. and del Rio, A. (2024) Smart IoT Irrigation System Based on Fuzzy Logic, Lora, and Cloud Integration. *Electronics*, **13**, Article 1949. https://doi.org/10.3390/electronics13101949

[14] (2021) Comparison Between LoRa and Other Wireless Technologies. https://www.mokolora.com/lora-and-wireless-technologies/