Real-Time Crop Monitoring and Alert System using CropBot

P. Kiranmayee
Electronics and communication Department
Nalla Narasimha Reddy Educations Society's Group of Institutions
Integrated campus
Hyderabad, Ghatkesar, India

H. Divyani
Electronics and communication Department
Nalla Narasimha Reddy Educations Society's Group of Institutions
Integrated campus
Hyderabad, Ghatkesar, India

G. Sai Nikhil
Electronics and communication Department
Nalla Narasimha Reddy Educations Society's Group of Institutions
Integrated campus
Hyderabad, Ghatkesar, India

I.Anjali
Electronics and communication Department
Nalla Narasimha Reddy Educations Society's Group of Institutions
Integrated campus
Hyderabad, Ghatkesar, India

***Abstract*—This paper explores a Real-Time Crop Monitoring and Alert System using Cropbot, designed to enhance agricultural productivity through continuous monitoring and timely alerts. The system integrates IoT-based platforms for data acquisition and analysis, employing sensors to measure soil moisture, temperature, and humidity in real time. After that, the gathered data is transferred to ThingSpeak for analysis and storage, and Blynk Cloud enables remote system control and monitoring. Additionally, Telegram is used to give farmers immediate warnings and alerts, guaranteeing proactive decision-making under unusual agricultural conditions. The prototype demonstrates reliability, scalability, making it a practical solution for precision agriculture and sustainable farming.**

Keywords: *Cropbot, IoT, precision agriculture, real- time monitoring, ThingSpeak, Blynk Cloud, Telegram alerts.*

I. INTRODUCTION

In countries like India, where the majority of the people makes their living from agriculture, agriculture plays a crucial role in the economic development of those countries. Nonetheless, the conventional agricultural approach still relies on manual monitoring of environmental factors including soil moisture, temperature, and humidity. As a result, water supplies are used inefficiently.

By offering a real-time monitoring system for agricultural fields, the Internet of Things (IoT) has created a new way to solve these problems in recent years. IoT-based smart farming integrates sensors, microcontrollers, and cloud services to analyze and transmit data efficiently.

Farmers can improve production and resource use by making well-informed decisions with real-time monitoring. With its camera and Wi-Fi capabilities, the ESP32 microcontroller offers an inexpensive and energy-efficient way to create intelligent farming systems. Real-time monitoring and control become easy and effective when paired with cloud services like Blynk and ThingSpeak.

Sensors, microcontrollers, and cloud platforms are used in IoT-based smart farming to effectively gather, process, and share data. Farmers can boost output by making educated decisions based on ongoing process monitoring. A cheap and energy-efficient option for smart agricultural systems is the ESP32 microcontroller, which features Wi-Fi and video capabilities. It is simple to visualize and control data in real time when paired with cloud platforms like Blynk and ThingSpeak.

II. LITRATURE SURVEY

Several researchers have explored IoT-based technologies to improve agricultural productivity through smart monitoring and automation.

This section presents a summary of existing works related to real-time crop health monitoring, IoT-based irrigation systems, and data-driven agriculture.

Meena Kumari et al. [1] developed an affordable, real- time crop health monitoring system aimed at promoting sustainable agriculture. Their work focused on using low-cost sensors to continuously assess crop conditions, highlighting the importance of scalability and farmer accessibility. However, the study did not integrate automated alert mechanisms for quick decision-making.

Rao and Krishna [*2*] used the ESP32 microcontroller with temperature, humidity, and soil moisture sensors to create an Internet of Things-based crop health monitoring system. An OLED display was also included in the system to provide real-time feedback. The system was effective at data sensing, but it lacked cloud-based data analysis and remote alerting features.

Bhutta and Ahmad [3] introduced a secure IoT framework for the identification and traceability of agricultural produce during transportation. Their work emphasized supply chain

transparency rather than in- field crop monitoring, which differentiates it from real-time agricultural IoT systems.

Gupta and Soni[4] proposed an IoT-based smart irrigation system using ESP32, DHT11, and soil sensors with the Blynk platform for monitoring. The proposed system was very effective in automating the irrigation process but lacked advanced data analytics and prediction capabilities.

Karthik and Singh [5] also implemented an IoT-based irrigation system using the Blynk platform to control water flow automatically. Their system demonstrated the benefits of real-time monitoring; however, it primarily focused on irrigation rather than complete crop health management.

Patel and Bhatt [6] reviewed IoT and smart agriculture technologies, emphasizing the potential of sensor networks and cloud computing in optimizing farm operations. Their work provided a theoretical foundation but lacked a practical, implemented system.

Jangam and Deshmukh [7] proposed an IoT-based smart farming system for real-time environmental monitoring using multiple sensors. Although the system was able to provide accurate readings, it lacked scalability and alert communication mechanisms.

Praveen and Venkatesh [8] proposed an IoT and cloud computing-based smart agriculture framework that enhanced data accessibility through cloud integration. However, the model's dependence on constant internet connectivity posed reliability issues in rural areas.

Sharma and Patel [9] has created an ESP32 and Blynk-based smart agriculture monitoring system. The ESP32 processes the data before sending it to cloud platforms like ThingSpeak for analysis and data logging. Simultaneously, Blynk Cloud enables real- time visualization of sensor data through a mobile app. If any parameter falls outside the desired range, platform, focusing on remote data visualization. Despite achieving real-time monitoring, their system lacked integration with a data analytics platform and an intelligent alert feature.

Qureshi and Shaikh [10] implemented an Arduino- based smart irrigation and crop monitoring system that measured soil moisture and temperature in real-time. Their approach, while simple and low- cost, had limited scalability and network capabilities compared to more advanced IoT microcontrollers..

III. DESIGN METHODOGY

A. System Components

The central control unit of the suggested system is an ESP32 microcontroller, which is linked to DHT11, soil moisture, and LDR sensors to measure temperature, humidity, light intensity, and soil moisture. Depending on the state of the soil, a DC motor that functions as a water pump is controlled by the relay module to enable automatic watering. To guarantee that the sensors and actuators operate as intended, the system is powered by regulated 5V and 12V voltage supply.

B. Block Diagram

CropBot Serves as the main Internet of the Things platform in this design, interacting with a variety of sensors to provide real-time agricultural data including temperature, humidity, and soil moisture. Real-time data analysis is made possible by processing the sensor values and sending them to the Blynk Cloud and ThingSpeak platforms via the ESP32 module. When there are variations in the values, the Telegram warning is automatically sent out.Effective communication between the gadget and the user, automated monitoring, and efficient data collection are all made possible by this architecture.

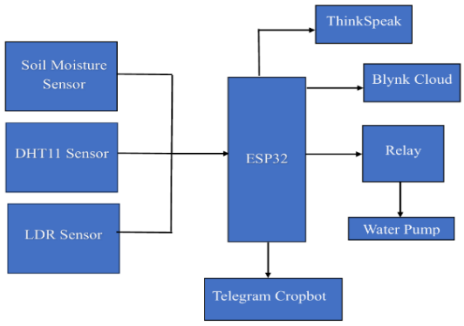

Figure 1. Block Diagram

The Real-Time Crop Monitoring and Alert System Using Cropbot block diagram, "Fig 1", illustrates how various sensors, including soil moisture, temperature, and humidity, are combined with the ESP32 microprocessor. The agricultural field's current environmental conditions are measured by the sensors.

The farmer receives an instant alert via Telegram Bot. In the end, this technology supports intelligent and sustainable agriculture by offering efficient monitoring, transparency, and quick communication.

IV. IMPLEMENTATION

A Real-time Crop Monitoring and Control System built using the ESP32 microcontroller is depicted in the circuit schematic "Fig 2" above. The system's central processing unit, the ESP32 microcontroller, reads data from the sensors and uses that data to drive the output device. The ESP32 microcontroller's GPIO 34 is linked to the soil moisture sensor.

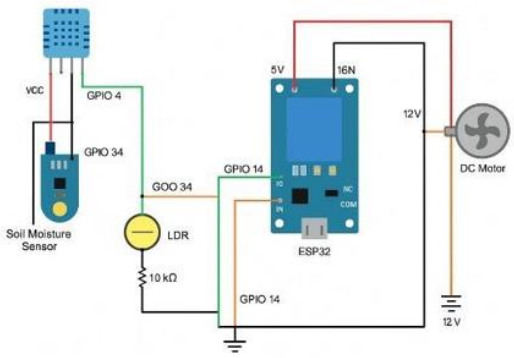

Figure 2. Circuit Diagram

The ESP32 reads the analog output from this sensor to determine if the soil is sufficiently wet or dry. In order to help the system distinguish between day and night, an LDR (Light Dependent Resistor) with a 10 kΩ resistor is also attached to GPIO 34. A relay module attached to GPIO 14 is controlled by the ESP32 based on the output received from these sensors. The relay module serves as a switch for a water pump and a 12V DC motor. The ESP32 activates the relay and turns on the DC motor to irrigate the soil when the soil dries below a predetermined threshold. The motor and relay are shut off when the soil is suitably moist. Because of its higher power needs, the motor is powered by a 12V supply, although the ESP32 and relay module are powered by a 5V source. This technique is ideal for smart agriculture since it enables automatic irrigation based on current environmental conditions.

V. EXPERIMENTAL SETUP

shows that the ESP32 microcontroller, which serves as the system's central control unit, connects the hardware components to one another. Processing data from various sensors and managing output devices like the relay and DC motor (water pump) are the responsibilities of the ESP32 microcontroller. The Soil Moisture Sensor is powered by either a 3.3V or 5V source, connected to the analog input pin GPIO 34, and properly grounded.

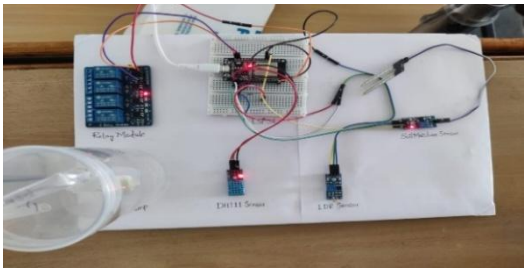

Figure 3. Hardware Connection

In “Fig 3” It allows the ESP32 to identify when irrigation is necessary by detecting the amount of water in the soil. The VCC and GND pins of the DHT11 Sensor, which detects temperature and humidity, are connected to the power and ground lines via GPIO 4. The soil moisture sensor and the LDR sensor share a time multiplexing approach to measure light intensity.

The LDR sensor is constructed as a voltage divider with a 10kΩ resistor. The ESP32's GPIO 14 is linked to the relay module, an electrically actuated switch that is powered by a 5V supply. It regulates the 12V DC motor (water pump) that is connected to its COM (Common) and NO (Normally Open) terminals, allowing the pump to be turned on and off in response to the moisture content of the soil.

To guarantee sufficient operating current and voltage, the motor is externally powered by a 12V power supply. Remote monitoring and control are made possible by the ESP32 module's Wi-Fi and Bluetooth capabilities, which allow real-time data transfer to platforms like Blynk and ThingSpeak. Together, the hardware components provide a fully automated smart irrigation system that efficiently performs irrigation operations while continuously monitoring and processing environmental factors. To guarantee the crop system's seamless operation and real-time monitoring, a number of software components were used.

The ESP32 microcontroller was programmed and coded using the Arduino IDE software. Additionally, it made it possible to leverage essential libraries like DHT, Blynk, and ThingSpeak APIs, which made it easier for cloud platforms and hardware to communicate effectively. The Blynk Cloud platform offered a mobile interface with dashboards that could be customized to display all sensor data.

Additionally, it provided capabilities like the ability to manually control the irrigation pump and send alarms as necessary. Through the usage of ThingSpeak for data logging and graphical representation, the system was able to upload sensor data on a regular basis and show it in a graphical style for easier comprehension. Furthermore, a Telegram Cropbot was created to deliver real-time notifications like "Soil Moisture Low – Irrigation Started," allowing farmers to take the appropriate action even when they are not physically in their fields.

Creating a mobile dashboard for remote control and real-time data visualization was part of the Blynk Cloud setup. A new project was initiated using an ESP32 as the device and Wi-Fi as the connection method after downloading the Blynk app from the Play Store or App Store. The dashboard was expanded with several widgets, including an LED or switch for pump status, a gauge for soil moisture, and value displays for temperature and humidity.

The Arduino IDE code was modified to include a duplicate of the Blynk-specific Auth Token. The code uses the function Blynk begin(auth, ssid, pass); to connect, and Blynk.virtualWrite(Vx, sensor value); to update the sensor values in the application. The ESP32 and sensors are the only hardware needed for the system, which also offers a user-friendly graphical interface for farmers and allows real-time monitoring.

MATLAB-based analytics were used to store and analyse the sensor data over time utilizing the ThingSpeak Cloud setup. Making an account on ThinkSpeak and then configuring a new channel with the fields for moisture, temperature, humidity, and light marked the first step in the configuration process. To make communication easier, the Write API Key was transferred into the Arduino IDE code. Data was sent to the ThingSpeak server using HTTP POST requests using the WIFI Client and HTTP Client libraries. For example, live readings are transmitted to the cloud using the codesnippetStringurl="https://api.thingspeak.com/update?api_key=YOUR_API_KEY&field1=" + String(moisture)+ "&field2=" + String(temp); http. Begin(URL); http. GET();.

For instant notification, a Telegram Bot was set up using the Telegram Bot API to directly send real-time notifications to the user’s mobile phone. The process involved opening the Telegram app and searching for “Bot Father” to create a new bot by using the /newbot command to get the Bot Token. Then, the Chat ID of the user or group was obtained by using the @userinfobot command or a simple bot script. In the

ArduinoIDE WiFi Client secure.h and Universal Telegram BoT libraries were employed for secure data transfer.
The bot was secured using
#define BOT_TOKEN "YOUR_BOT_TOKEN"
#defineCHAT_ID "YOUR_CHAT_ID" and notifications were sent using bot. Send Message(CHAT_ID,"ı.Soil is dry. Irrigation started.", "");This system allows farmers to instantly

receive notifications about soil moisture, temperature, or irrigation without having to constantly monitor the apps. Additionally, multiple farmers can be included in a single group to facilitate joint monitoring and rapid responses to changes in the field conditions.

VI. FLOWCHART

Based on the field's current circumstances, the alert system is designed to provide the farmer with automated, instantaneous, and actionable replies. When the temperature or humidity rises above the allowed range, the soil moisture level falls below the predetermined threshold, or the light intensity abnormally drops, an alert is triggered. The Telegram Bot, which sends out messages like "Alert! Low Soil Moisture detected," is how the system delivers immediate alerts. The irrigation process began. and through the Blynk App, which provides pop-up alerts and live widget updates for real-time monitoring of pump status and sensor values. The relay circuit is automatically controlled based on soil moisture levels turning the DC water pump ON when the soil is too dry and OFF when the required moisture level is achieved ensuring efficient irrigation and water conservation. Example the alerts are:

1.“Alert! Soil too dry” irrigation started automatically.

2.“Alert! Soil moisture optimal” irrigation stopped automatically.

The flowchart in “Fig 4” depicts how an Internet of Things-based real-time crop monitoring and alert system operates. The start node initiates the system and sets up the entire procedure. The next stage involves gathering data from the agricultural field using Internet of Things (IoT) sensors, such as temperature, humidity, soil moisture, and light sensors.

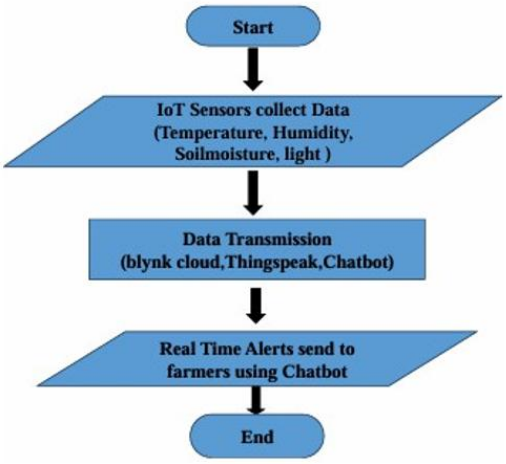

Figure 4. Flowchart

Following its acquisition, the data is transmitted to internet platforms such as ThingSpeak, Blynk Cloud, and a chatbot interface. Data exchange, visualization, and cloud storage are made possible via online platforms. In the last phase, the system employs a chatbot (like the Telegram bot) to notify farmers in real time of any significant changes that have taken place, such high temperatures or low soil moisture. The end node, which indicates that the data cycle is over until the subsequent update, marks the end of the procedure.

VII. RESULT

A. *Blynk Cloud Output*

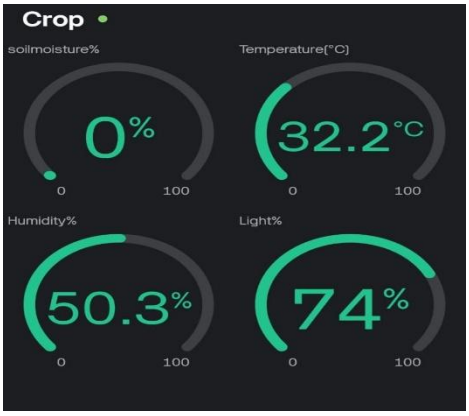

Figure 5. Blynk Cloud Output

The dashboard that indicates the current crop conditions is depicted in “Fig. 5” above. At zero, the soil moisture content is extremely low, which encourages irrigation. The light intensity is strong at 74, the temperature is high at 32.2 °C, and the moisture content is moderate at 50.3.

B. *ThingSpeak Graph Visualization*

The environmental elements' real-time graphs are displayed on the ThingSpeak dashboard. The soil moisture is critically low at about 0%, the temperature is steady at 32°C, the humidity slightly drops to 48.5%, and the light intensity is high at 80% in “Fig 6.” Irrigation automation and crop status monitoring can benefit from the values.

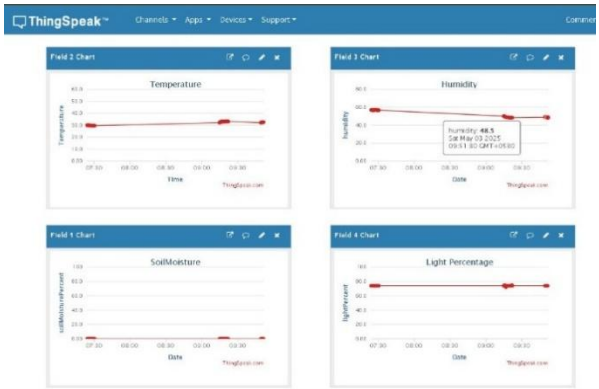

Figure 6. ThinkSpeak Graph Visualization

C. *Real-Time Alerts from Telegram*

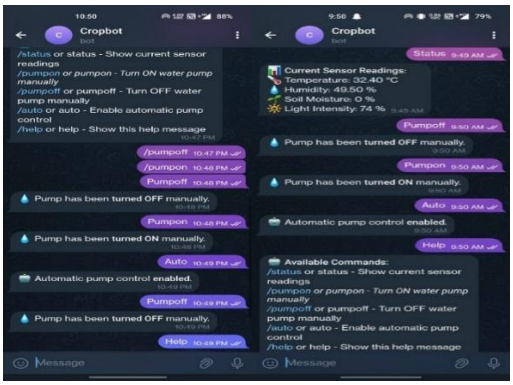

Figure 7. Real-Time Alerts from Telegram

A Telegram conversation between a user and a bot called Cropbot, which is used to operate a water pump system, is depicted in "Fig 7". The Telegram conversation provides instructions such as /auto to activate the automated mode, pump on to manually turn the pump on, and pumpoff to turn it off. The user can use these commands to manually switch the pump on and off and to activate the automatic mode, as demonstrated in the Telegram discussion.

VII. CONCLUSION

It has been determined that the Cropbot Real-Time Crop Monitoring and Alert System is a clever and effective way to address the contemporary issues facing agriculture. This system enables real-time, continuous monitoring of vital environmental parameters including soil moisture, temperature, humidity, and light intensity by integrating IoT sensors, an ESP32 microcontroller, and cloud services like Blynk Cloud, ThingSpeak, and Telegram.

In addition to assisting with precise data gathering and cloud-based visualization, this technology offers real-time warnings and communication with farmers via the Telegram Cropbot service. Water conservation and human intervention are further reduced by the irrigation automation process that uses a relay-controlled water pump based on soil moisture data.

XI. FUTURE SCOPE

The Real-Time Crop Monitoring and Alert System demonstrates a practical, low-cost, and efficient solution for addressing common agricultural challenges through IoT-based automation. However, as technology continues to evolve, there is significant potential for enhancement and scalability of this project to serve broader agricultural needs. The following future advancements are proposed:

1. GPS and GIS-Based Analytics
2. Cloud and Database Enhancements
3. Automation in Fertilizer and Pesticide Delivery

REFERENCES

[1] Meena Kumari, Dr. Naimur Rahman Kidwai, Dr. Shimi S.L. "Sustainable Agriculture through Affordable, Real-Time Crop Health Monitoring," in2024, Lucknow.

[2] G. V. Rao and P. V. Krishna, *"IoT-Based Crop Health Monitoring System Using ESP32"*, 2022. Method: ESP32 with temperature, humidity, and moisture sensors, OLED feedback.

[3] Muhammad Nasir Mumtaz Bhutta and Muneer Ahmad "*Secure Identification, Traceability and Real-Time Tracking of Agricultural Food Supply During Transportation Using Internet of Things,"* IEEE Access, 2021.

[4] S. Gupta and R. Soni, *"IoT-Based Real-Time Crop Monitoring and Smart Irrigation System"*, 2020. Method: IoT with ESP32, soil and DHT11 sensors, Blynk for monitoring.

[5] R. S. Karthik and A. R. Singh, "*Development of IoT-Based Smart Irrigation System Using Blynk Platform,"* International Journal of Science and Research (IJSR), Sept 2020.

[6] A. Patel and C. Bhatt, *"Internet of Things and Smart Agriculture Technologies", Boca* Raton, 2019.

[7] S. R. Jangam and A. A. Deshmukh, "IoT-Based Smart Farming System for Real-Time Monitoring," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 5, pp. 1030–1034, May 2019.

[8] T. R. Praveen and K. V. Venkatesh, "Smart Agriculture Using IoT and Cloud Computing," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, no. 6, pp. 120–125, 2019.

[9] M. Sharma and V. Patel, *"Smart Agriculture Monitoring System using ESP32 and Blynk," International Conference on Intelligent Computing and Control Systems (ICICCS)2019*.

[10] S. N. Qureshi and M. A. Shaikh, "IoT Based Real Time Crop Monitoring and Smart Irrigation System Using Arduino," *IEEE Xplore*, 2018.