LORA BASED EFFICIENT CROP MONITORING SYSTEM

A PROJECT REPORT

Submitted by

SHANTOSH N	**(822720106313)**
MOHAMMED THAJUDEEN R	**(822720106025)**
KAVICHOZHAN K	**(822720106016)**
VASANTH S	**(822720106046)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING, THANJAVUR

ANNA UNIVERSITY : CHENNAI - 600 025

MAY 2024

ANNA UNIVERSITY : CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report **" LORA BASED EFFICIENT CROP MONITORING SYSTEM "** is the bonafide work of **"SHANTOSH N (822720106313), MOHAMMED THAJUDEEN R (822720106025), KAVICHOZHAN K (822720106016), VASANTH S (822720106046)"** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Mr. R. RAMJI, M.E.,(Ph.D).	**Mr. K. MANIKANDAN, M.E.**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
Department of ECE	Department of ECE
Government College of Engineering	Government College of Engineering
Thanjavur-613 402	Thanjavur-613 402

Submitted for the project and viva-voce examination held on _________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

This project work is dedicated to Almighty God blessing with inspirational parents, teachers and good friends.

We would like to express our deep gratitude to our beloved Principal **Dr. S. JAYABAL., M.E., Ph.D.** for his moral support and words of enthusiasmwhich helped us a lot in completing this project.

We would like to thank **Mr. R. RAMJI M.E., (Ph.D).** Head of Electronics and Communication Engineering department for his invaluable guidance and encouragement in completing this project work.

We would like to express our deep gratitude and thanks to our well- wisher and internal guide **Mr. K. MANIKANDAN,M.E.** for his invaluable guidance, advice, moral support and encouragement without which this project would not be successful.

We would like to extend our sincere thanks to our project coordinators for their constant efforts to bring this project successful.

We also like to thank all the respected staff members of Electronics and Communication department and various department of our college for their directand indirect involvement in successful completion of this project.

We like to express our sincere thanks and gratitude to our parents and friends for their continuous encouragement and support.

CHAPTER 1

INTRODUCTION

In an era marked by rapid urbanization and climate change, the demand for sustainable and efficient agricultural practices has never been greater. Traditional irrigation methods often lead to water wastage, decreased crop yields, and increased operational costs. However, with advancements in Internet of Things (IoT) technology, a new era of smart agriculture is dawning. Our project introduces a LoRa-based IoT Smart Irrigation System, leveraging the power of LoRa (Long Range) communication, ESP8266 microcontroller. This innovative system not only addresses the challenges faced by conventional irrigation methods but also offers a scalable and cost-effective solution for farmers and agricultural enthusiasts alike. LoRa enables long-range communication, allowing our irrigation system to cover vast agricultural areas with minimal infrastructure requirements. LoRa's low power consumption and robustness make it ideal for remote and off-grid applications. The ESP8266 serves as the brain of our system, providing Wi-Fi connectivity and interfacing with various sensors and actuators. Its affordability, versatility, and compatibility with the Arduino IDE make it a popular choice for IoT projects. Our Smart Irrigation System consists of sensor nodes deployed across the agricultural field, each equipped with soil moisture sensors, temperature sensors, and LoRa transceivers. These nodes continuously monitor environmental conditions and transmit data wirelessly to a central gateway.

The gateway, based on an ESP8266 module, receives sensor data from multiple nodes. Users can access web dashboard to view sensor readings, adjust irrigation settings, and analyze historical data trends. By embracing IoT technology, our Smart Irrigation System offers several significant benefits: Precise monitoring of soil moisture levels allows for targeted irrigation, reducing water wastage and promoting sustainable water usage practices. Optimal irrigation scheduling based on real-time environmental data ensures that crops receive the right amount of water at the right time, leading to healthier plants and higher yields. By minimizing water usage and optimizing resource allocation, farmers can reduce operational costs associated with irrigation, fertilization, and labor. The modular design of our system allows for easy scalability to accommodate varying farm sizes and crop types. Additionally, its adaptability to different environmental conditions makes it suitable for diverse agricultural settings worldwide. Enable users to remotely monitor and control the irrigation system from anywhere , providing convenience and flexibility. Implement LoRa communication between the irrigation system and a central gateway, allowing for long-range communication even in remote areas with limited connectivity. Integrate ESP8266 microcontrollers into the system to control actuators such as water pumps, valves, and sprinklers based on sensor data and user commands. Implement data logging functionality to record sensor readings and irrigation events over time, enabling users to analyze trends, optimize watering strategies, and identify areas for improvement. Optimize power consumption of the ESP8266 nodes to prolong battery life in battery-operated setups, ensuring continuous operation without frequent battery replacements. Incorporate features for fault detection and diagnostics to identify and address issues such as sensor malfunctions, communication errors, or water leaks promptly.

CHAPTER 2

LITERATURE SURVEY

2.1. IOT-EQUIPPED AND AI-ENABLED NEXT GENERATION SMART AGRICULTURE: A CRITICAL REVIEW, CURRENT CHALLENGES AND FUTURE TRENDS

BRIEF INTRODUCTION

A detailed tutorial on the available advancements in the field of smart agriculture systems through IoT technologies and AI techniques; A critical review of these two available technologies and challenges in their widespread deployment; and An in-depth discussion about the future trends including both technological and social, when smart agriculture systems will be widely adopted by the farmers globally.

TECHNIQUES USED

IoT sensors are deployed in agricultural fields to collect real-time data on various parameters such as soil moisture, temperature, humidity, light intensity, and crop health. These sensors are crucial for monitoring the conditions of the farm remotely. This involves processing, analyzing, and interpreting the data to derive meaningful insights. Machine learning algorithms are often used for predictive analytics, anomaly detection, and pattern recognition. ML and AI techniques play a vital role in smart agriculture by enabling predictive modeling, decision-making, and automation. These techniques are used for tasks such as crop disease detection, yield prediction, pest detection, and precision farming. These technologies provide valuable information for crop monitoring, precision agriculture, and environmental management.

2.2. AN AUTOMATED IRRIGATION SYSTEM FOR AGRICULTURE USING IOT

BRIEF INTRODUCTION

Smart irrigation is a cutting-edge method of water management that makes use of cutting-edge technology to enhance irrigation procedures. To choose the ideal quantity and time of water application for crops or lawns, this system combines sensors, weather information, and computer algorithms. Smart irrigation systems help to preserve water resources, cut down on water waste, and boost crop yields by only giving plants the water they require. Additionally, smart irrigation has a number of benefits over conventional irrigation techniques, including simplicity in installation, enhanced control and monitoring, and increased effectiveness.

TECHNIQUES USED

Deploy IoT sensors throughout the agricultural field to monitor soil moisture levels, temperature, humidity, and other relevant environmental parameters. These sensors collect real-time data, which is crucial for determining when and how much to irrigate. Develop a data acquisition system to collect and process data from the IoT sensors.. Enable automation features to automate irrigation scheduling based on predefined criteria or environmental conditions. Optimization algorithms can adjust irrigation parameters dynamically to maximize water efficiency and crop yield while minimizing resource usage.

2.3. ENHANCING IRRIGATION EFFICIENCY WITH AI-BASED INSTINCTIVE IRRIGATION SYSTEM (IIS) IN WIRELESS SENSOR NETWORKS

BRIEF INTRODUCTION

An AI-assisted intuitive irrigation system built on Arduino and Zigbee that operates within a wireless sensor network is an example of an innovative application of technology in the field of agriculture. This technology enhances the process of watering in real time by utilizing sensors, microcontrollers, a wireless connection, and algorithmic programs that are powered by artificial intelligence. An open-source microcontroller platform serves as the “brains” of the system, receiving data from a wide variety of sensors and using that information to direct the operation of the irrigation system.

TECHNIQUES USED

Deploy WSNs consisting of IoT sensors throughout the agricultural field to monitor various environmental parameters such as soil moisture, temperature, humidity, and light intensity. These sensors communicate wirelessly with a central control unit. Utilize AI algorithms, such as machine learning and deep learning models, to analyze the data collected by the WSNs and make intelligent decisions regarding irrigation scheduling. The AI system learns from historical data and environmental conditions to predict optimal irrigation timing and volume. Incorporate adaptive learning capabilities into the AI algorithms to continuously improve irrigation strategies over time. The system adapts to changing environmental conditions, crop growth stages, and farmer preferences, optimizing irrigation efficiency and resource utilization.

2.4. AN AUTOMATED IRRIGATION SYSTEM: AN IOT APPLICATION

BRIEF INTRODUCTION

Agriculture plays an important role in the development of the country and our country India is a global agricultural powerhouse. Agriculture and related activities provides livelihood to major part of ruler India and requires labour at major scale. It is an Integration of IoT with Agriculture. The Irrigation of farms can be done automatically, by considering two important factors i.e. soil moisture level and weather forecast .our aims is to get the moisture level from soil using moisture level sensor and weather forecast using API's. After doing required calculation decision of field irrigation take place, in an automated manner also with user access to alter it. This project can reduce one of the biggest workloads in Agriculture and drive it towards the automation.

TECHNIQUES USED

Deploy IoT sensors throughout the agricultural field to monitor soil moisture levels, temperature, humidity, and other relevant environmental parameters. These sensors collect real-time data, which is crucial for determining when and how much to irrigate. Develop a data acquisition system to collect and process data from the IoT sensors. This system may include microcontrollers or single-board computers such as Arduino or Raspberry Pi, capable of interfacing with sensors and transmitting data to the cloud. Utilize a cloud platform to store, manage, and analyze the data collected from the field. Cloud platforms offer scalability, reliability, and accessibility, allowing users to access irrigation data from anywhere with an internet connection.

2.5. DESIGN AND TESTING OF AN AUTOMATIC IRRIGATION CONTROLLER FOR FRUIT TREE ORCHARDS BASED ON SAP FLOW MEASUREMENTS

BRIEF INTRODUCTION

The device calculates the irrigation dose (ID) from sap flow readings in the trunk of trees irrigated to replenish the crop water needs, relative to similar measurements made in over-irrigated trees. It then acts on the pump and electrovalve to supply an ID sufficient to keep the soil close to its field capacity during the irrigation period. Remote control of the system is possible from any computer or Smartphone connected to the Internet. We tested the CRP in an olive orchard in southern Spain. The device was robust and able to filter and amplify the output voltages of the heat-pulse velocity probes and to calculate reliable sap flow data.

TECHNIQUES USED

The remote control facility proved to be useful for getting real-time information both on the CRP behaviour and the applied IDs, and for changing parameters of the irrigation protocol. For our conditions, olive trees with big root systems growing in a soil with a remarkable water-holding capacity, the approach mentioned above for calculating ID had not enough resolution to replace the daily crop water consumption. The device, however, was able to react when the soil water content fell below the threshold for soil water deficit. The threshold value was identified with simultaneous measurements of stem water potential in the instrumented trees. Our results suggest a change in the irrigation protocol that will allow the CRP to apply a recovery irrigation whenever that threshold is reached, making the device suitable for applying a deficit irrigation strategy in the orchard.

CHAPTER 3
EXISTING SYSTEM

3.1 EXISTING SYSTEM

Leveraging IoT Technology is imperative for advancing agricultural productivity, as it facilitates the precise management of critical concerns like theft and destruction in crop fields. By deploying IoT sensors and monitoring systems, farmers can establish real-time surveillance and data-driven insights into their fields, enabling proactive measures against theft and minimizing destruction. Micro and small scale enterprises in agriculture encounter various challenges that hinder their productivity and profitability. These challenges include limited access to resources, lack of technical knowledge, unpredictable weather conditions, and inefficient resource management practices. Farmers may struggle to access real-time data on soil moisture levels, weather forecasts, and crop water requirements, making it difficult to make informed decisions about irrigation scheduling. Many irrigation systems require manual operation, which can be time-consuming and inefficient, especially for farmers managing large or multiple fields from a distance. In remote areas, farmers may face challenges with unreliable or limited access to electricity, which is essential for powering irrigation pumps and automation systems.

3.2 DISADVANTAGES

- In areas with water scarcity or competing demands for water resources, farmers may face challenges in accessing sufficient water for irrigation, especially if they are managing their operations from a distance and unable to monitor water availability closely.
- Farmers may need to comply with regulations related to water usage and conservation, which can be challenging to monitor and enforce remotely.

CHAPTER 4
PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

Leveraging IoT Technology is imperative for advancing agricultural productivity, as it facilitates the precise management of critical concerns like theft and destruction in crop fields. By deploying IoT sensors and monitoring systems, farmers can establish real-time surveillance and data-driven insights into their fields, enabling proactive measures against theft and minimizing destruction.

4.2 ADVANTAGES

- Real-time alerts for farmers to take Immediate action to protect the crops.
- LoRa device can transmit data over 10 kilometers without an internet.
- Flexible to monitor animals, humans and theft activity in crop field.
- Solutions helps Customizable settings, user-friendly interface.
- Agricultural motors can be remotely operated using LoRa modules, and precise control of solenoid valves.
- This data can be accessed in the cloud at any time.

4.3 BLOCK DIAGRAM

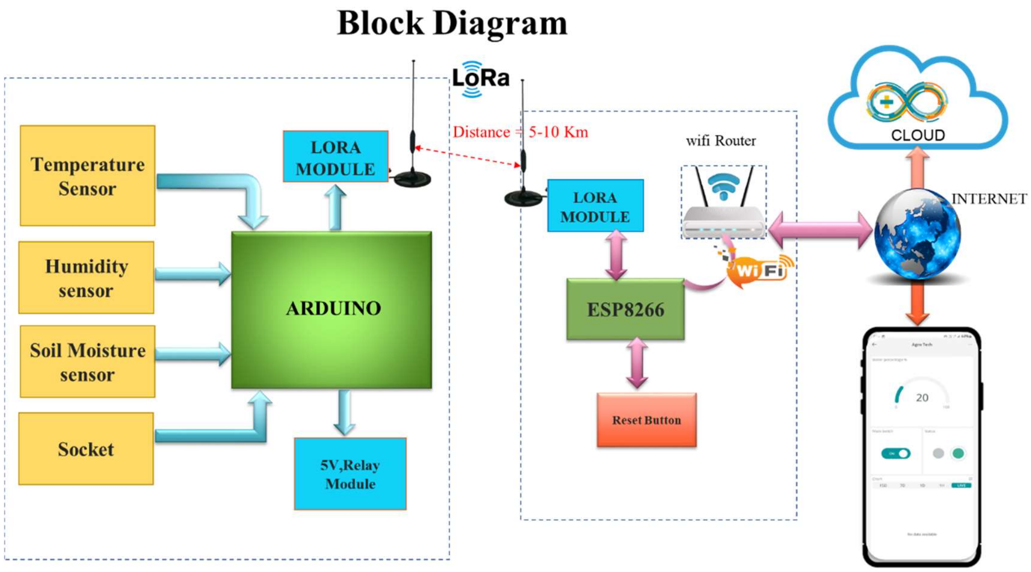

FIGURE 4.1 Overview Of Proposed System

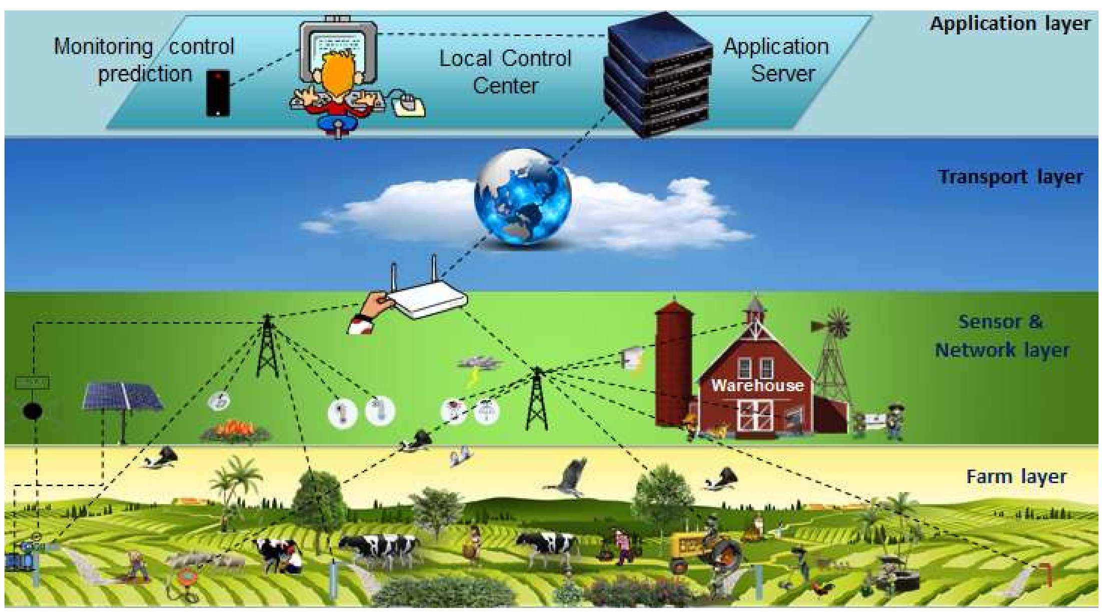

FIGURE 4.2 Workflow Of The Proposed System

4.4 COMPONENTS DESCRIPTION

For LoRa based two way wireless communication system, we required the following components.

- Arduino Nano
- LoRa
- NodeMCU ESP8266-12E Board
- Capacitive Soil Moisture Sensor
- DHT11 Temperature & Humidity Sensor
- 5V Single-Channel Relay Module
- Few jumpers wires
- 5V DC Water Pump

4.4.1 ARDUINO NANO

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on the use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -duino.

An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left. Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8,[28] ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

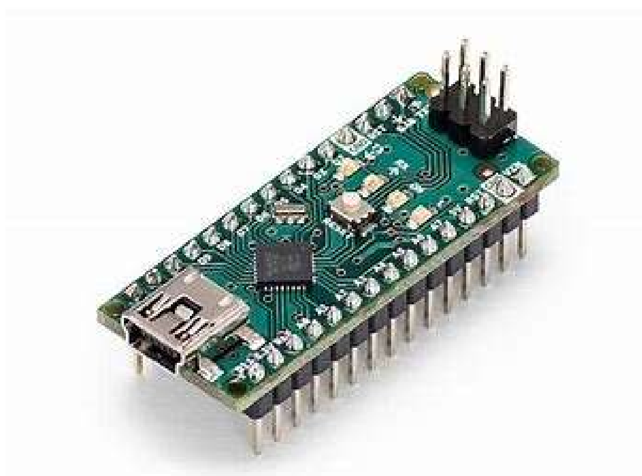

FIGURE 4.3 Arduino Nano

Arduino microcontrollers are pre-programmed with a bootloader that simplifies the uploading of programs to the on-chip flash memory. The default bootloader of the Arduino Uno is the Optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL serial) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila,[a] Duemilanove,[b] and current Uno[c] provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins.

4.4.2 LORA MODULE

LoRa (Long Range) hardware refers to the physical devices and components used in LoRaWAN (Long Range Wide Area Network) wireless communication systems. LoRa is a low-power, long-range wireless technology that enables devices to communicate over long distances with low data rates while consuming minimal power.

LoRa (from "long range") is a physical proprietary radio communication technique. It is based on spread spectrum modulation techniques derived from chirp spread spectrum (CSS) technology. It was developed by Cycleo, a company of Grenoble, France, and patented in 2014. Cycleo was later acquired by Semtech. LoRaWAN (wide area network) defines the communication protocol and system architecture. LoRaWAN is an official standard of the International Telecommunication Union (ITU), ITU-T Y.4480. The continued development of the LoRaWAN protocol is managed by the open, non-profit LoRa Alliance, of which Semtech is a founding member.

FIGURE 4.4 Lora Module

Together, LoRa and LoRaWAN define a low-power, wide-area (LPWA) networking protocol designed to wirelessly connect battery operated devices to the Internet in regional, national or global networks, and targets key Internet of things (IoT) requirements, such as bi-directional communication, end-to-end security, mobility and localization services. The low power, low bit rate, and IoT use distinguish this type of network from a wireless WAN that is designed to connect users or businesses, and carry more data, using more power. The LoRaWAN data rate ranges from 0.3 kbit/s to 50 kbit/s per channel.

LoRa uses license-free sub-gigahertz radio frequency bands EU868 (863–870/873 MHz) in Europe; AU915/AS923-1 (915–928 MHz) in South America; US915 (902–928 MHz) in North America; IN865 (865–867 MHz) in India; and AS923 (915–928 MHz) in Asia; LoRa enables long-range transmissions with low power consumption. The technology covers the physical layer, while other technologies and protocols such as LoRaWAN (long range wide area network).

LoRa Module SX1278

The SX1278 Ra-02 Chip is manufactured by Semtech. The SX1278 RF module is mainly used for long-range spread spectrum communication. It can resist Minimizing current consumption & has a high sensitivity of -148 dBm with a power output of +20 dBm. A long transmission distance of almost 5km can be achieved with high reliability. Thus SX1278 is great for IoT applications to send sensor data to the cloud over long ranges.

SX1278-Module

This module uses SX1278 IC from SEMTECH and works on a 433MHz frequency. Frequency hopping—which gives you that sweet balance of quality signal transmission will cover a range of 420-450 MHz. This long-range wireless capability is packed into a small (17 x 16mm) package and delivered without an antenna.

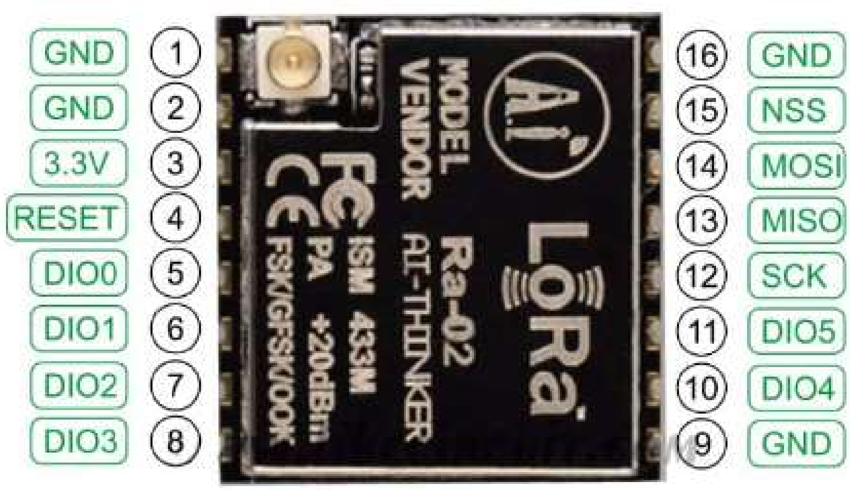

FIGURE 4.5 SX1278 Pinout

PIN	DESCRIPTION
ANT	This pin is to attach to the antenna.
Pin 2, 9, 16: GND	Ground pin of common ground with power supply and controllers.
Pin 3: 3.3V	To power up the device pin 3 will help for power input.
RESET	Pin 4 is to reset the module through an external signal.
Pin 5, 6, 7, 8, 10, 11: DIO0, DIO1, DIO2, DIO3, DIO4, DIO5	To perform the general, I/O function through module the DIO pins will help. These pins is customizable as an interrupt pin.
Pin 12: SCK	SCK pin is for the clock pulse during the SPI communication.
Pin 13: MISO	MISO means Master in and Slave out that transfers data transfer from the Module to the Controller. The Master is a controller & SX1278 is Slave.
Pin 14: MOSI	MOSI means Master out Slave In. So, this pin will receive the data from Controller.
Pin 15: NSS	NSS is a chip select/enable pin will help to activate the slave

TABLE 4.1 SX1278 PINOUT CONFIGUREURATION

4.4.3 NodeMCU ESP8266-12E Board

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

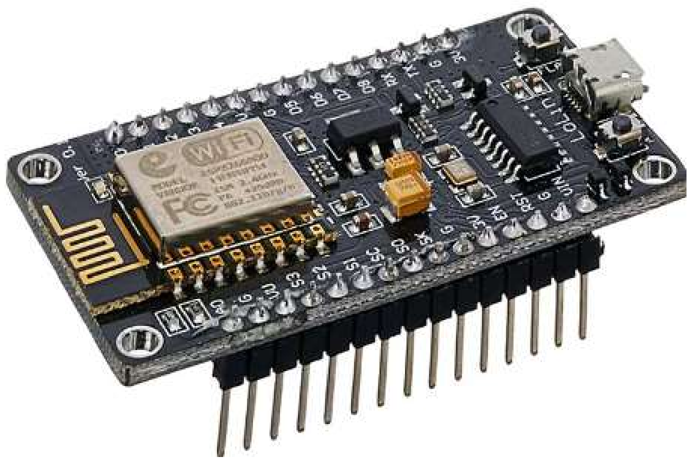

FIGURE 4.6 Esp8266 Node Mcu

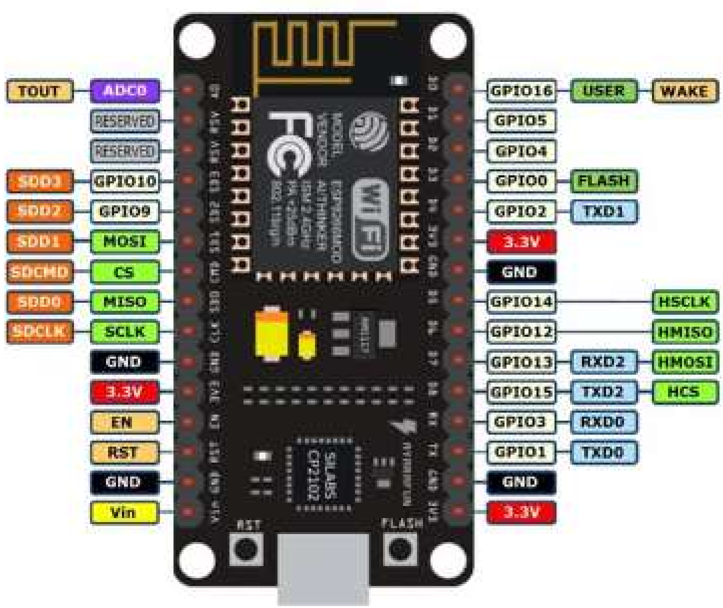

FIGURE 4.7 Esp8266 Node Mcu Pinout

PIN CATEGORY	NAME	DESCRIPTION
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

TABLE 4.2: NodeMCU PINOUT CONFIGUREURATION

4.4.4 Capacitive Soil Moisture Sensor

Traditional soil moisture sensor uses resistance variations to detect the amount of moisture present in the soil. These sensors were very much prone to degradation due to electrode corrosion. It means they need a lot of maintenance and lose accuracy soon. The capacitive soil moisture sensor can detect the moisture without coming in direct contact with the water in the soil. Read the analog value, which is proportional to the moisture in the ground. The moisture in the soil changes the dielectric value, thereby changing the read capacitance. The sensor converts its capacitance into an analog value, which you can read using Arduino's analog inputs. Most of the sensors come in a wide range of supply voltage. You can connect the sensors to 3.3 V and 5 V Arduino boards.

The benefits of using the capacitive soil moisture sensor are

- No need to calibrate the sensor frequently
- No loss of accuracy over time
- No corrosion of the sensor

FIGURE 4.8 Capacitive Soil Moisture Sensor

4.4.5 TRANSMITTER

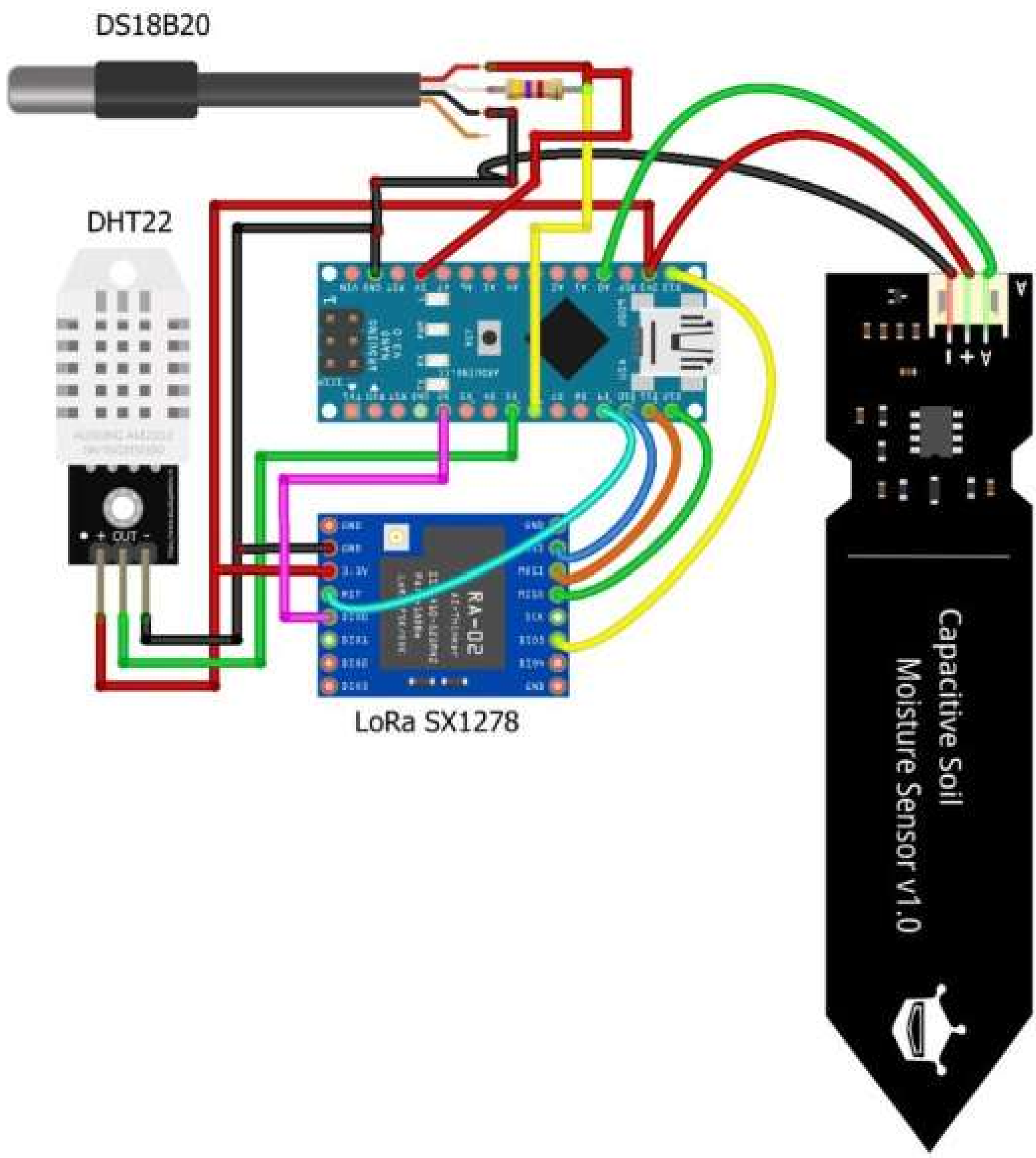

FIGURE 4.9 Transmitter Circuit Diagram

LoRa SX1278	**Arduino Nano**
GND	GND
3.3V	3.3V
NSS	D10
MOSI	D11
MISO	D12
SCK	D13
RST	D9
DIO0	D2
Capacitive Soil Moisture	Arduino Nano
VCC	3.3V
GND	GND
DHT11/22	Arduino Nano
VCC	3.3V
GND	GND
Signal	D5
DS18B20	Arduino Nano
VCC	5V
GND	GND
Data	D6
4.7k ohm resistor	VCC & Data pin
5V Relay Module	Arduino Nano
VCC	5V
GND	GND
IN1	D7

TABLE 4.3 TRANSMITTER CIRCUIT CONNECTION

4.4.6 RECEIVER

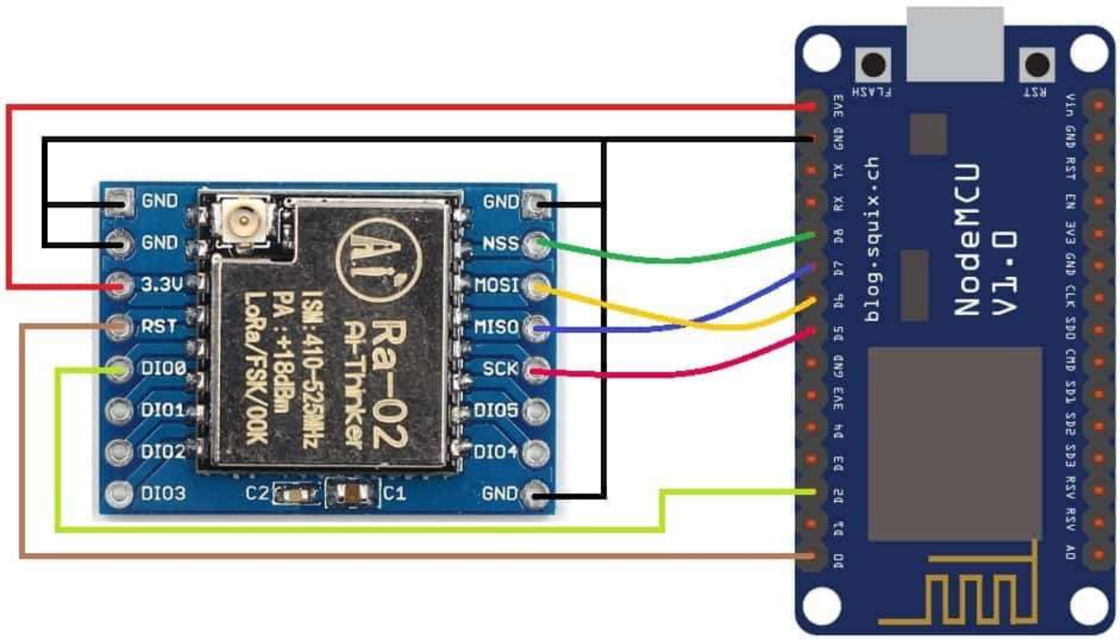

FIGURE 4.10 Receiver Circuit Diagram

SX1278 LoRa	ESP8266 NodeMCU
GND	GND
3.3V	3.3V
NSS	D8
MOSI	D7
MISO	D6
SCK	D5
RST	D0
OLED Display	NodeMCU ESP8266
GND	GND
VDD	3.3V
SCK	D1
SDA	D2

TABLE 4.4 Receiver Circuit Connection

4.4.7 DHT11 Temperature & Humidity Sensor

Adding a DHT11 temperature and humidity sensor to your LoRa-based IoT Smart Irrigation System with ESP8266 can further enhance its capabilities.

Environmental Monitoring: The DHT11 sensor provides accurate measurements of temperature and humidity, allowing your irrigation system to monitor environmental conditions more comprehensively. This data can be crucial for determining optimal irrigation schedules based on plant requirements and current weather conditions.

Water Conservation: By incorporating temperature and humidity data into your irrigation decision-making process, you can avoid unnecessary watering during periods of high humidity or low temperatures, reducing water waste and promoting more efficient water usage.

Plant Health Management: Temperature and humidity levels directly impact plant health and growth. Integrating these measurements into your irrigation system enables you to create customized irrigation schedules tailored to specific plant needs, optimizing plant health and yield.

Preventing Water Stress: High temperatures coupled with low humidity can lead to water stress in plants. By monitoring these conditions in real-time, your irrigation system can respond promptly by adjusting watering frequency or duration to prevent water stress and maintain optimal growing conditions.

Data-driven Decision Making: The addition of temperature and humidity data enables data-driven decision-making processes for irrigation management. Analyzing historical data trends can help identify patterns and correlations between environmental factors and plant responses, guiding future irrigation strategies for improved outcomes.

Integration with Existing Infrastructure: The DHT11 sensor can be seamlessly integrated into your existing ESP8266-based system, allowing you to leverage the same communication protocols and data processing mechanisms already in place for other sensors. This simplifies the integration process and ensures compatibility with your current setup.

Sensor Placement Considerations: When deploying DHT11 sensors in the field, consider factors such as sensor placement and protection from environmental elements to ensure accurate and reliable measurements. Shielding the sensors from direct sunlight and moisture can help prevent interference and prolong sensor lifespan.

Power Consumption Optimization: Implement power-saving techniques to minimize the energy consumption of the DHT11 sensor and prolong battery life in battery-operated setups. This may involve adjusting sampling intervals or using sleep modes to conserve power when sensor data is not required.

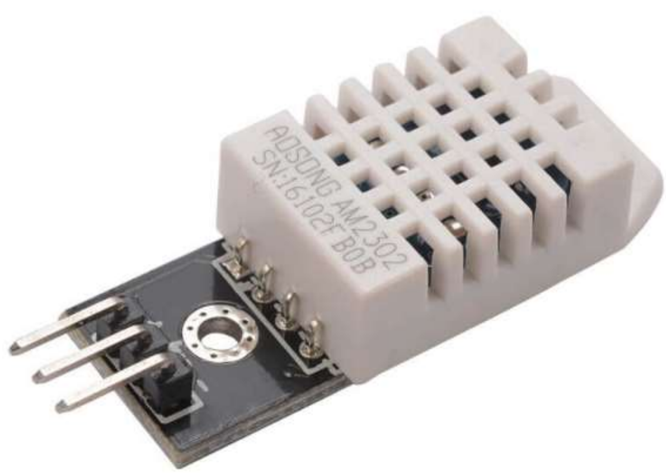

FIGURE 4.11 Dht11 Temperature & Humidity Sensor

4.4.8 DS18B20 waterproof Temperature Sensor

The DS18B20 sensor is suitable for measuring soil temperature, providing valuable insights into soil conditions crucial for plant growth. Monitoring soil temperature helps optimize irrigation schedules, seed germination, and plant development. Soil temperature data from the DS18B20 sensor can help detect and prevent frost damage to plants. By monitoring soil temperature thresholds and triggering alerts or irrigation actions when temperatures approach freezing levels, you can protect sensitive plants from frost-related harm. Soil temperature influences water uptake and nutrient absorption by plants. Integrating soil temperature measurements into your irrigation system allows for more precise timing of watering events based on soil temperature thresholds, ensuring optimal growing conditions and water efficiency.

Soil temperature varies with seasonal changes, affecting plant growth and nutrient availability. By continuously monitoring soil temperature throughout the year, your irrigation system can adapt irrigation schedules and strategies to seasonal variations, promoting healthier plants and maximizing yield. Soil temperature data can help prevent overwatering by adjusting irrigation schedules based on soil moisture and temperature conditions.

Figure 4.12 Ds18b20 Waterproof Temperature Sensor

4.4.9 5V Single-Channel Relay Module

The relay module can be used to control irrigation valves remotely, allowing you to open or close valves based on sensor readings, user commands, or predefined schedules. This enables precise control over water flow to individual zones or plants, optimizing irrigation efficiency and water usage. Integrate the relay module with your irrigation system to automate watering processes based on environmental conditions, such as soil moisture levels, temperature, and humidity. The ESP8266 can trigger the relay to activate irrigation valves when specific thresholds are met, ensuring plants receive water when needed. Provide a manual override option through the relay module, allowing users to manually activate or deactivate irrigation valves as needed. This gives users flexibility and control over irrigation operations, especially in situations where immediate adjustments are required.

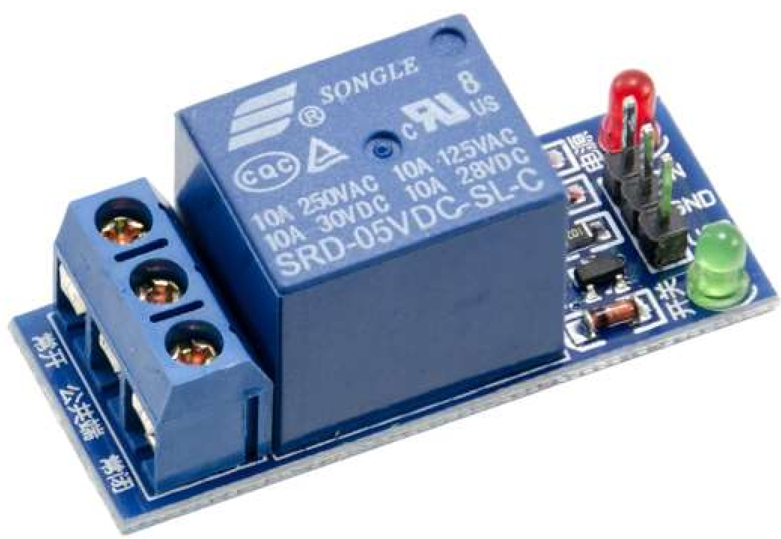

FIGURE 4.13 5v Single-Channel Relay Module

4.4.10 Jumping wires

In breadboard, jumping wires are used to create temporary connections between various components (such as resistors, LEDs, sensors, etc.) and the pins of an integrated circuit (IC) or microcontroller placed on the breadboard. This allows for quick prototyping and testing of electronic circuits without the need for soldering. Jumping wires are useful for troubleshooting and debugging circuits. They can be used to bypass or replace faulty connections, verify signal paths, and isolate specific components for testing or replacement. Jumping wires facilitate the connection of different components within a circuit. For example, they can be used to connect the output of one component (e.g., a sensor) to the input of another component (e.g., a microcontroller), enabling data transmission or control signals.

FIGURE 4.14 Jumping Wires

4.4.11 5V DC Water Pump

Use the 5V DC water pump to automate the irrigation process by delivering water directly to plants or irrigation zones based on predetermined schedules or sensor data. The ESP8266 can control the water pump's operation, activating it when soil moisture levels are below a certain threshold or at specific times of the day.

The water pump enables precise control over water delivery, allowing you to deliver the right amount of water to each plant or zone. This prevents overwatering and ensures that plants receive adequate hydration without wastage. Divide your garden or agricultural area into different irrigation zones, each served by its own water pump.

FIGURE 4.15 5v Dc Water Pump

This allows for customized watering schedules and irrigation strategies tailored to the specific needs of different plants or areas. By automating irrigation with the water pump, you can optimize water usage and minimize waste. The system can water plants only when needed, based on soil moisture levels or environmental conditions, reducing water consumption and promoting conservation.

Remote Control and Monitoring: Integrate the water pump with your LoRa-based IoT system to enable remote control and monitoring. You can remotely activate or deactivate the water pump, adjust watering schedules, and monitor irrigation status from anywhere with an internet connection, providing greater flexibility and convenience.

Emergency Watering: Use the water pump for emergency watering during periods of drought or extreme heat. The system can be programmed to provide additional water to plants during critical periods to prevent wilting or dehydration, ensuring plant health and survival.

Versatile Applications: The 5V DC water pump can be used for various applications beyond irrigation, such as hydroponics, aquariums, fountain displays, and water circulation systems. Its compact size and low voltage make it suitable for a wide range of projects requiring water pumping capabilities.

Integration with Sensor Data: Integrate the water pump with sensor data, such as soil moisture levels or weather forecasts, to implement intelligent irrigation strategies. The ESP8266 can trigger the water pump based on real-time sensor readings or predictive analytics, optimizing watering decisions for maximum efficiency and effectiveness.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 ARDUINO SOFTWARE

Arduino is an Italian open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. It's hardware products are licensed under a CC BY-SA license, while the software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages (Embedded C), using a standard API which is also known as the Arduino Programming Language, inspired by the Processing language and used with a modified version of the Processing IDE. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

FEARURES OF ARDUINO

Arduino is an open-source hardware and software platform that consists of microcontrollers and development boards that can be programmed to perform a wide variety of tasks. Here are some of the features of Arduino:

1. Microcontroller:Arduino boards are built around a microcontroller that can be programmed tocontrol sensors, motors, and other electronic devices.
2. Input/output pins:Arduino boards have a variety of input and output pins that can be used to interface with sensors, switches, and other electronic devices.
3. Analog-to-digital converter:Arduino boards have an onboard analog to-digital converter(ADC) that allows them to read analog signals from sensors and other devices.
4. Integrated development environment (IDE):Arduino has a user friendly IDE that makes it easyto write, compile, and upload code to the board.
5. Shields:Arduino boards can be expanded with a variety of shields that provide additionalfunctionality, such as WiFi, Bluetooth, and GPS.
6. Libraries:Arduino has a large collection of libraries that make it easy to interface withcommon electronic devices and sensors.
7. Open source:Arduino is an open-source platform, which means that the hardware designs, software code, and documentation are freely available to the public.

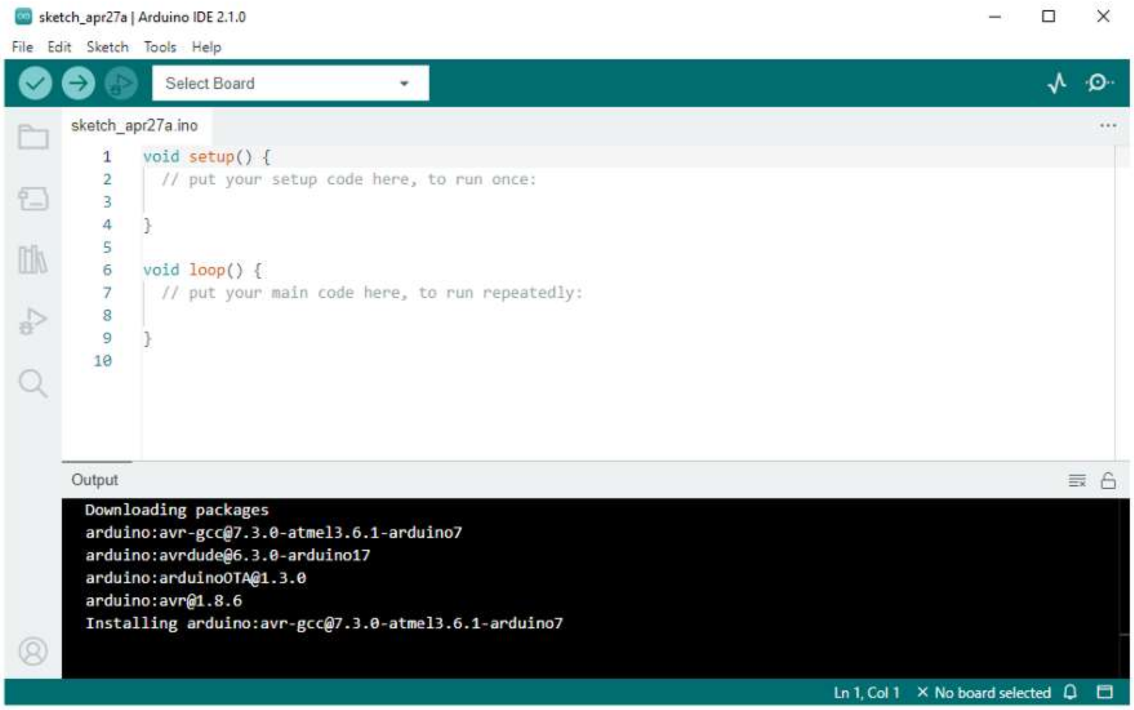

FIGURE 5.1 ARDUINO IDE

An initial alpha preview of a new Arduino IDE was released on October 18, 2019, as the Arduino Pro IDE. The beta preview was released on March 1, 2021, renamed IDE 2.0. On September 14, 2022, the Arduino IDE 2.0 was officially released as stable. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment and autocompletion support. The application frontend is based on the Eclipse Theia Open Source IDE. Its main new features are:

- Modern, fully featured development environment
- New Board Manager
- New Library Manager
- Board List
- Basic Auto-Completion
- Serial Monitor

5.2 OUTPUT

FIGURE 5.2 Transmitter Hardware

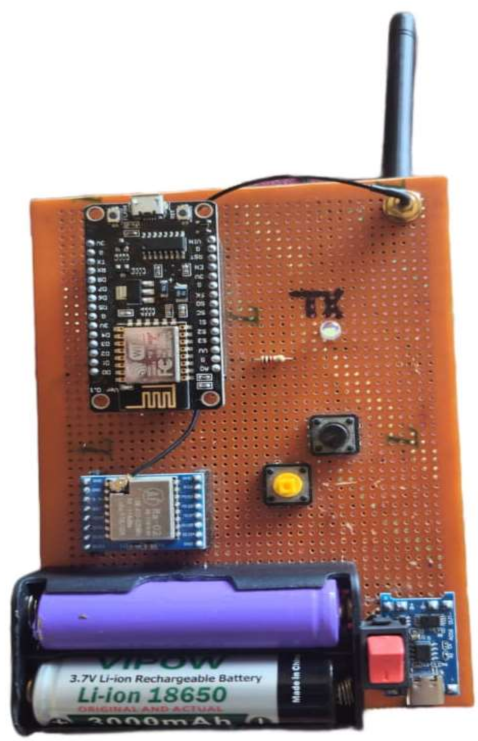

FIGURE 5.3 Receiver Hardware

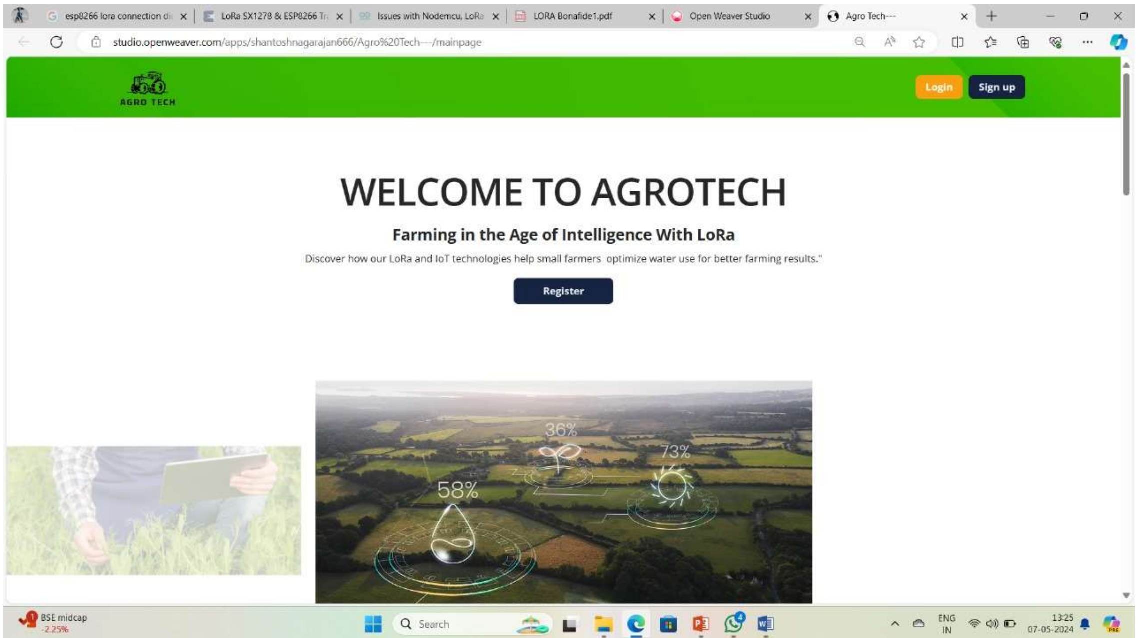

FIGURE 5.4 Home Page

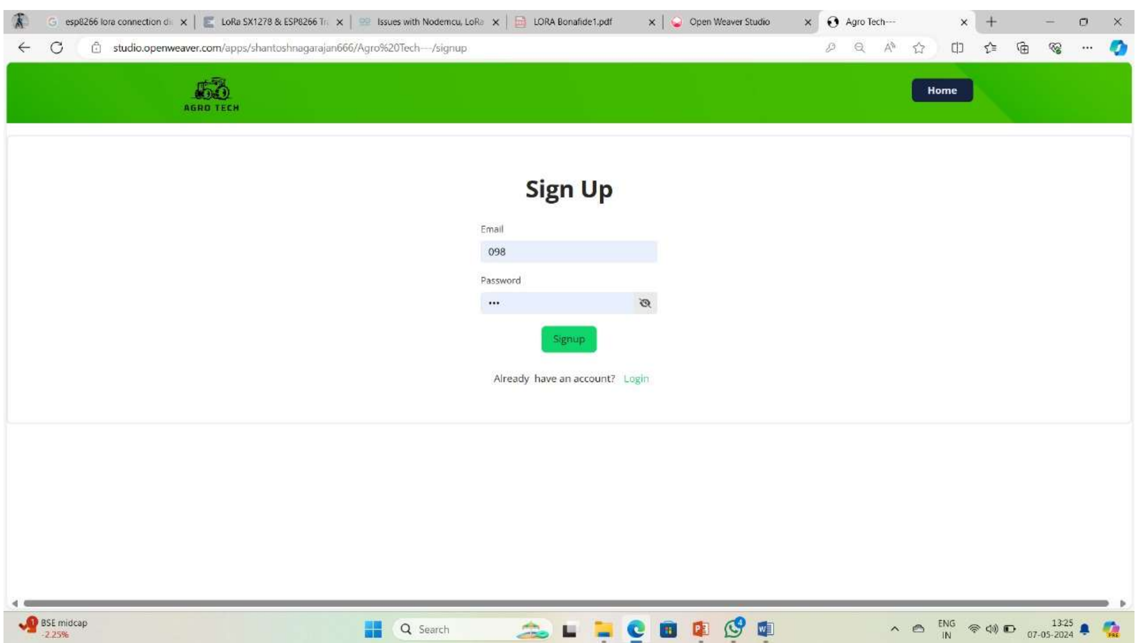

FIGURE 5.5 Signup Page

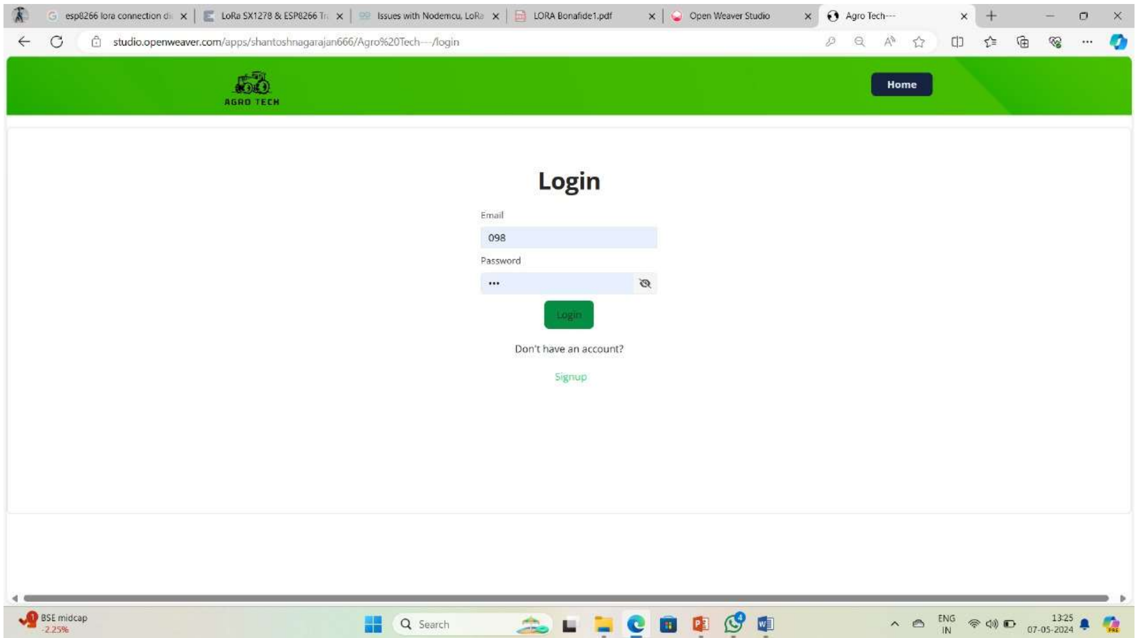

FIGURE 5.6 Login Page

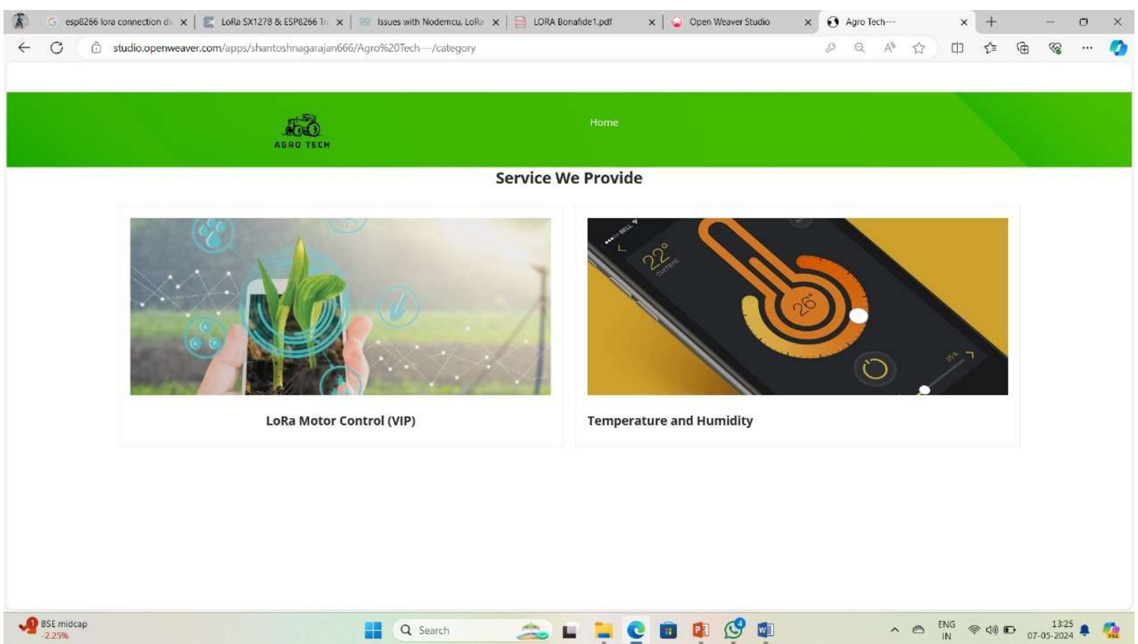

FIGURE 5.7 Services Provided

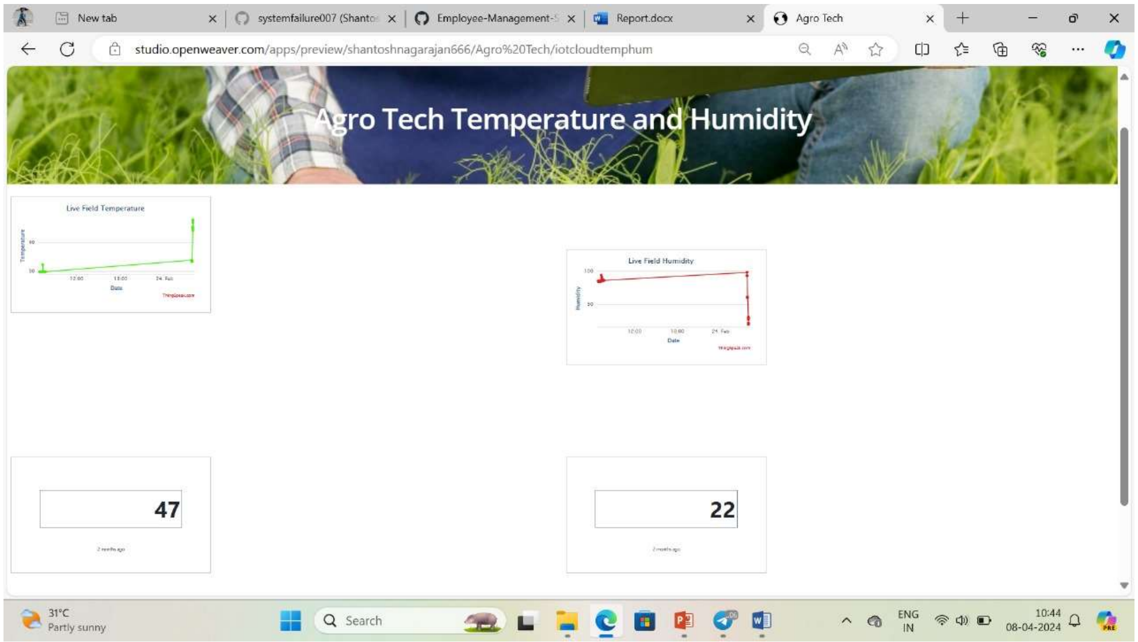

FIGURE 5.8 Temperature And Humidity

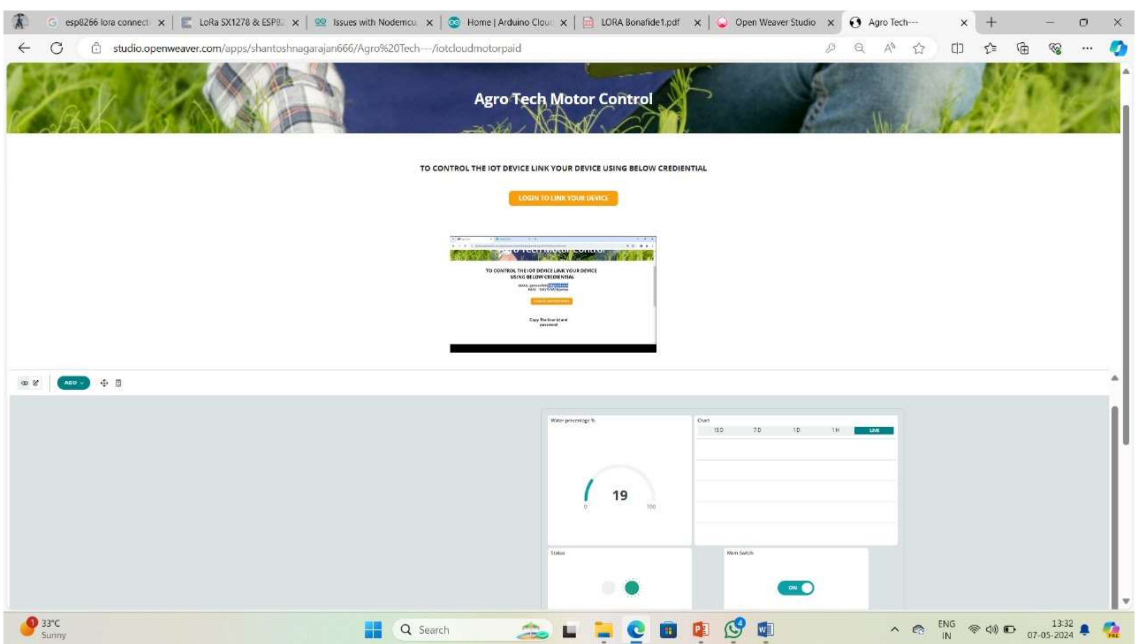

FIGURE 5.9 Linking With Device

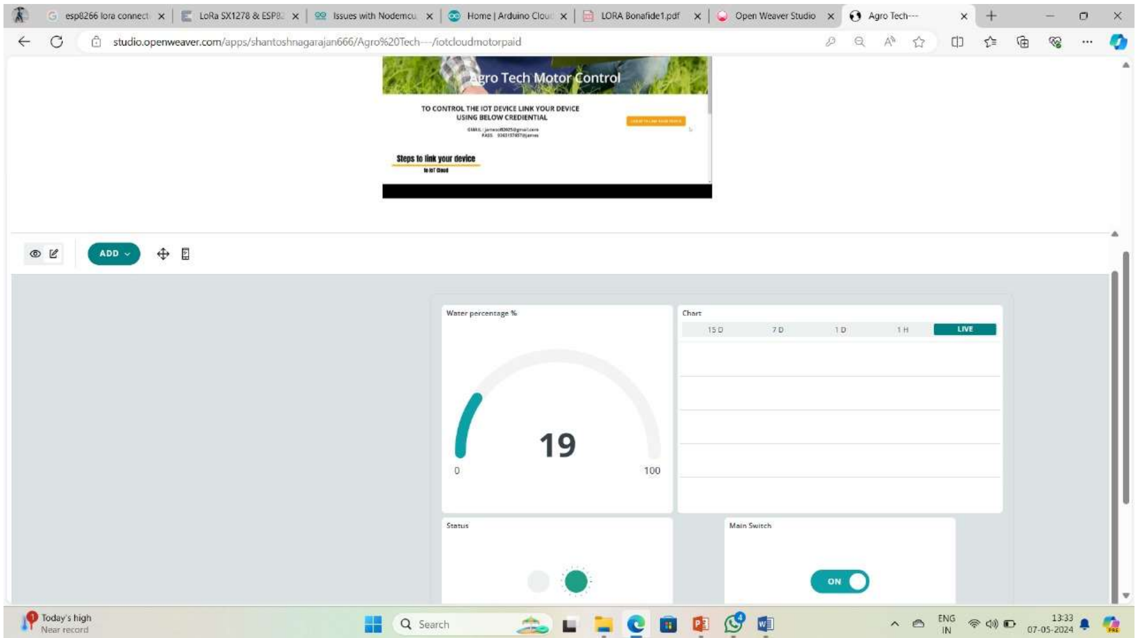

FIGURE 5.10 Motor Control

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

Small-scale farmers with limited resources and technical expertise. Micro-enterprises in agriculture seeking to improve efficiency and productivity, Large-scale Agricultural Enterprises, Agribusiness Companies, Government Agricultural Departments , Agricultural Research Institutions, Technology Enthusiasts and Innovators. AI technologies enable farmers to monitor crops, soil, and weather conditions with unprecedented precision. This leads to optimized resource utilization such as water, fertilizers, and pesticides, reducing waste and environmental impact. AI algorithms can analyze vast amounts of data including historical weather patterns, soil quality, and crop performance to predict outcomes..

AI-powered drones and satellite imagery can provide real-time monitoring of crop health, detecting issues like pests, diseases, or nutrient deficiencies early on. This allows for targeted interventions, minimizing crop loss and increasing yields. AI-driven automation can streamline various tasks in agriculture, reducing the need for manual labor. AI algorithms can analyze patterns and identify signs of pests or diseases in crops much faster than human observation alone. Early detection allows for prompt intervention, potentially preventing widespread infestations and reducing the need for chemical treatments.

6.2 FUTURE SCOPE

Enhance the system by integrating weather forecast data. This can help optimize irrigation schedules based on predicted rainfall, humidity, and temperature, thus conserving water and improving plant health. This ensures more accurate irrigation decisions and prevents overwatering or under watering. Implement remote monitoring and control capabilities via a web or mobile application. This allows users to monitor soil moisture levels, adjust irrigation schedules, and receive alerts from anywhere, enhancing convenience and efficiency.

Incorporate data analytics capabilities to analyze historical sensor data and identify patterns or trends. This can provide valuable insights for optimizing irrigation strategies, detecting anomalies, and predicting future irrigation needs. Expand the system's functionality by integrating additional sensors, such as temperature, humidity, light intensity, or nutrient levels. This provides a more comprehensive view of environmental conditions and enables more precise irrigation management. Develop a system for automatic plant identification using image processing techniques. This allows the system to tailor irrigation schedules and requirements to specific plant species, optimizing water usage and promoting plant health. Integrate water quality sensors to monitor the quality of irrigation water. This ensures that plants receive clean water, free from contaminants or pollutants, which can impact plant growth and soil health. Enable integration with smart home systems (e.g., Google Home, Amazon Alexa) for voice-controlled operation and seamless integration with other smart devices in the home environment. Optimize the system for energy efficiency by implementing power-saving techniques for the ESP8266 module and other components.

APPENDIX

LoRA TRANSMITTER PROGRAM CODE :

```
#include <Wire.h>
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define DHTPIN 5          //pin where the DHT11 is connected
DHT dht(DHTPIN, DHT11);
#define relay 7
#define ss 10
#define rst 9
#define dio0 2
#define ONE_WIRE_BUS 6
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

String LoRaMessage = "";
int counter = 0;

const int AirValue = 590;   //you need to replace this value with Value_1
const int WaterValue = 300;  //you need to replace this value with Value_2
const int SensorPin = A0;
int soilMoistureValue = 0;
int soilmoisturepercent = 0;


void setup()
{
  Serial.begin(115200);
  dht.begin();
  sensors.begin(); // Dallas temperature
```

```
  pinMode(relay, OUTPUT);

  while (!Serial);
  Serial.println("LoRa Sender");
  LoRa.setPins(ss, rst, dio0);
    if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    delay(100);
    while (1);
  }
}

void loop()
{
  soilMoistureValue = analogRead(SensorPin);  //put Sensor insert into soil
  soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  sensors.requestTemperatures();
  float temp = sensors.getTempCByIndex(0);

  if (isnan(h) || isnan(t))
  {
  Serial.println("Failed to read from DHT sensor!");
  return;
  }
  Serial.print("Soil Moisture Value: ");
  Serial.println(soilMoistureValue);

  Serial.print("Soil Moisture: ");
  Serial.print(soilmoisturepercent);
  Serial.println("%");

  Serial.print("Soil Temperature: ");
  Serial.print(temp);
  Serial.println("°C");
```

```
Serial.print("Temperature: ");
Serial.print(t);
Serial.println("°C");

Serial.print("Humidity: ");
Serial.print(h);
Serial.println("%");
Serial.println("");

Serial.print("Sending packet: ");
Serial.println(counter);
  if (soilmoisturepercent >= 0 && soilmoisturepercent <= 40)
{
  Serial.println("Plants need water..., notification sent");
  digitalWrite(relay, LOW);
  Serial.println("Motor is ON");
  //WidgetLED PumpLed(V5);
  //PumpLed.on();
   }
else if (soilmoisturepercent > 30 && soilmoisturepercent <= 100)
{
  Serial.println("Soil Moisture level looks good...");
  digitalWrite(relay, HIGH);
  Serial.println("Motor is OFF");
  //WidgetLED PumpLed(V5);
  //PumpLed.off();
}
LoRaMessage = String(counter) +"/" + String(soilMoistureValue) + "&" +
 // send packet
LoRa.beginPacket();
LoRa.print(LoRaMessage);
LoRa.endPacket();
 counter++;

delay(1500);
```

LoRa RECEIVER PROGRAM CODE:

```
// Fill-in information from your Blynk Template here
#define BLYNK_TEMPLATE_ID "TMPLxxxx"
#define BLYNK_DEVICE_NAME "IoT Smart Agro"

#define BLYNK_FIRMWARE_VERSION        "0.1.0"

#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG

#define APP_DEBUG

// Uncomment your board, or conFIGUREure a custom board in Settings.h
//#define USE_SPARKFUN_BLYNK_BOARD
#define USE_NODE_MCU_BOARD
//#define USE_WITTY_CLOUD_BOARD
//#define USE_WEMOS_D1_MINI

#include <SPI.h>
#include <LoRa.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "BlynkEdgent.h"

#define ss 15
#define rst 16
#define dio0 4

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

```
String counter;
String soilMoistureValue;
String soilmoisturepercent;
String soiltemp;
String temperature;
String humidity;

void setup() {
  Serial.begin(115200);
  BlynkEdgent.begin();
  while (!Serial);

  Serial.println("LoRa Receiver");

  LoRa.setPins(ss, rst, dio0);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C
addr 0x3C (128x64)
  display.clearDisplay();
}

void loop() {
  BlynkEdgent.run();
  // try to parse packet
  int pos1, pos2, pos3, pos4, pos5;
  int packetSize = LoRa.parsePacket();
  if (packetSize)
  {
    // received a packet
    Serial.print("Received packet:  ");
    String LoRaData = LoRa.readString();
```

```
Serial.print(LoRaData);
// read packet
while (LoRa.available()) {
  Serial.print((char)LoRa.read());
}
// print RSSI of packet
Serial.print("' with RSSI ");
Serial.println(LoRa.packetRssi());

pos1 = LoRaData.indexOf('/');
pos2 = LoRaData.indexOf('&');
pos3 = LoRaData.indexOf('#');
pos4 = LoRaData.indexOf('@');
pos5 = LoRaData.indexOf('$');

counter = LoRaData.substring(0, pos1);
soilMoistureValue = LoRaData.substring(pos1 + 1, pos2);
soilmoisturepercent = LoRaData.substring(pos2 + 1, pos3);
soiltemp = LoRaData.substring(pos3 + 1, pos4);
temperature = LoRaData.substring(pos4 + 1, pos5);
humidity = LoRaData.substring(pos5 + 1, LoRaData.length());

//send data to blynk
Blynk.virtualWrite(V1, soilmoisturepercent); //Soil Moisture
Blynk.virtualWrite(V2, soiltemp); //Soil Temperature
Blynk.virtualWrite(V3, temperature);  // for Temperature
Blynk.virtualWrite(V4, humidity);  //for Humidity

Serial.print(F("Packet No = "));
Serial.println(counter);

Serial.print(F("Soil Moisture: "));
Serial.print(soilmoisturepercent);
Serial.println(F("%"));

Serial.print(F("Soil Temperature: "));
```

```
Serial.print(soiltemp);
Serial.println(F("°C"));

Serial.print(F("Temperature: "));
Serial.print(temperature);
Serial.println(F("°C"));

Serial.print(F("Humidity = "));
Serial.print(humidity);
Serial.println(F("%"));

Serial.print("Soil Moisture Value: ");
Serial.print(soilMoistureValue)
Serial.println();
if (soilmoisturepercent.toInt() > 100)
{
  display.clearDisplay();

  // display Soil Humidity
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 5);
  display.print("RH of Soil: ");
  display.print("100");
  display.print(" %");

  // display soil temperature
  display.setCursor(0, 20);
  display.print("Soil Temp: ");
  display.print(soiltemp);
  display.print(" ");
  display.cp437(true);
  display.write(167);
  display.print("C");

  display.display();
```

```
  delay(1500);
}
else if (soilmoisturepercent.toInt() < 0)
{
  display.clearDisplay();

  // display Soil Humidity
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 5);
  display.print("RH of Soil: ");
  display.print("0");
  display.print(" %");

  // display soil temperature
  display.setCursor(0, 20);
  display.print("Soil Temp: ");
  display.print(soiltemp);
  display.print(" ");
  display.cp437(true);
  display.write(167);
  display.print("C");

  // display air temperature
  display.setCursor(0, 35);
  display.print("Air Temp: ");
  display.print(temperature);
  display.print(" ");
  display.cp437(true);
  display.write(167);
  display.print("C");

  // display relative humidity of Air
  display.setCursor(0, 50);
  display.print("RH of Air: ");
  display.print(humidity);
```

```
    display.print(" %");

    display.display();
    delay(1500);
  }
  else if (soilmoisturepercent.toInt() >= 0 && soilmoisturepercent.toInt() <=
100)
  {
    display.clearDisplay();

    // display Soil humidity
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0, 5);
    display.print("RH of Soil: ");
    display.print(soilmoisturepercent);
    display.print(" %");

    // display soil temperature
    display.setCursor(0, 20);
    display.print("Soil Temp: ");
    display.print(soiltemp);
    display.print(" ");
    display.cp437(true);
    display.write(167);
    display.print("C");

    // display air temperature
    display.setCursor(0, 35);
    display.print("Air Temp: ");
    display.print(temperature);
    display.print(" ");
    display.cp437(true);
    display.write(167);
    display.print("C");
```

```
    // display relative humidity of Air
    display.setCursor(0, 50);
    display.print("RH of Air: ");
    display.print(humidity);
    display.print(" %");

    display.display();
    delay(1500);
   }
   if (soilmoisturepercent.toInt() >= 0 && soilmoisturepercent.toInt() <= 30)
   {
    Serial.println("needs water, send notification");
    //send notification
    Blynk.logEvent("water_your_plants", "Please Water your plants they are
about to die...") ;
    Serial.println("Motor is ON");
    Blynk.virtualWrite(V5, 255);
    delay(1000);
   }
   else if (soilmoisturepercent.toInt() > 30 && soilmoisturepercent.toInt() <=
100)
   {
    Serial.println("Soil Moisture level looks good...");
    Serial.println("Motor is OFF");
    Blynk.virtualWrite(V5, 0);
    delay(1000);
   }

  }
}
```

REFERENCES

[1] G. S. Nagaraja, A. B. Soppimath, T. Soumya and A. Abhinith, "IoT based smart agriculture management system", Proc. 4th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solution (CSITSS), pp. 1-5, Dec. 2019.

[2] E. Siddhartha and M. C. Lakkannavar, "Smart irrigation and crop health prediction", Proc. Int. Conf. Recent Trends Electron. Inf. Commun. Technol. (RTEICT), pp. 739-742, Aug. 2021.

[3] K. Parasuraman, U. Anandan and A. Anbarasan, "IoT based smart agriculture automation in artificial intelligence", Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV), pp. 420-427, Feb. 2021.

[4] K. G. Liakos, P. Busato, D. Moshou, S. Pearson and D. Bochtis, "Machine learning in agriculture: A review", Sensors, vol. 18, no. 8, pp. 2674, 2018.

[5] Tanha Talaviya, Dhara Shah, Nivedita Patel, Hiteshri Yagnik and Manan Shah, "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides", Artificial Intelligence in Agriculture, vol. 4, pp. 58-73, 2020.

[6] Y. Mekonnen, S. Namuduri, L. Burton, A. Sarwat and S. Bhansali, "Machine learning techniques in wireless sensor network based precision agriculture", Journal of the Electrochemical Society, vol. 167, no. 3, pp. 037522, 2019.

[7] K. Jha, A. Doshi, P. Patel and M. Shah, "A comprehensive review on automation in agriculture using artificial intelligence", Artificial Intelligence in Agriculture, vol. 2, pp. 1-12, 2019.

[8] S. S. L. Chukkapalli, S. Mittal, M. Gupta, M. Abdelsalam, A. Joshi, R. Sandhu, et al., "Ontologies and artificial intelligence systems for the cooperative smart farming ecosystem", Ieee Access, vol. 8, pp. 164045-164064, 2020.

[9] A. Sharma, A. Jain, P. Gupta and V. Chowdary, "Machine learning applications for precision agriculture: A comprehensive review", IEEE Access, vol. 9, pp. 4843-4873, 2020.

[10] K. Jha, A. Doshi, P. Patel and M. Shah, "A comprehensive review on automation in agriculture using artificial intelligence", Arti13ficial Intelligence in Agriculture, vol. 2, pp. 1-12, 2019.

[11] H. Zhou, X. Wang, W. Au, H. Kang and C. Chen, "Intelligent robots for fruit harvesting: Recent developments and future challenges", Precision Agriculture, pp. 1-52, 2022.
[12] Al Lawati Haydar, "Remittances of expatriate workers in the Gulf", Alroya Electronic Newspaper., 2020.
[13] "Internet of Things for smart agriculture: Technologies practices and future direction", Ray, vol. 9, no. 4, pp. 395-420, 2017.
[14] T. Ojha, S. Misra and N. S. Raghuwanshi, "Wireless sensor networks for agriculture:The state-of-the-art in practice and future challenges", Comput. Electron. Agricult, vol. 118, pp. 66-84, Oct. 2015.
[15] X. Zhang, J. Zhang, L. Li, Y. Zhang and G. Yang, "Monitoring citrus soil moisture and nutrients using an IoT based system", Sensors, vol. 17, no. 3, pp. 447, 2017.
[16] Al Badi Maryam, "65.4 thousand job seekers in the Sultanate.72 of them are youth", Alroya Electronic Newspaper, 2021.
[17] G. J. Rosline, P. Rani and D. Gnana Rajesh, "Comprehensive Analysis on Security Threats Prevalent in IoT-Based Smart Farming Systems", Ubiquitous Intelligent Systems, pp. 185-194, 2022.
[18] D. Sasikala and K. V. Sharma, "Future Intelligent Agriculture with Bootstrapped Meta-Learning and ϵba-greedy Q-learning", Journal of Artificial Intelligence and Capsule Networks, vol. 4, no. 3, pp. 149-159, 2022.