LoRa-Enabled Greenhouse Monitoring System

A Mini Project report submitted in partial fulfilment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND TELEMATICS ENGINEERING

BY

A. ANKITHA	**21251A1733**
K. SYARVANI	**21251A1745**
L. JHANSI AAKANKSHA	**21251A1747**
S. MITHILA	**21251A1759**

Under guidance of **N. Rama Krishna**, **Assistant Professor,** ETE department

Department of Electronics & Telematics Engineering

G. Narayanamma Institute of Technology &Science (For Women)

Autonomous

(NAAC & NBA Accredited Institution)

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana)

Shaikpet, Hyderabad – 500104, Telangana, India

May 2024

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE
(AUTONOMOUS)
(For Women)
(Sponsored by G. Pulla Reddy Charities Trust, Hyderabad)
Accredited by NBA (UG : ECE, ETM, EEE, CSE & IT ; PG : CSE, CNIS, PEED, DECE) & NAAC A+
Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad

CERTIFICATE

Date: 2nd September, 2024

This is to certify that the Mini Project report on "LoRa-Enabled Greenhouse Monitoring System" is a bonafide work carried out by A. Ankitha (21251A1733), K.Syarvani (21251A1745), L. Jhansi Aakanksha (2121A1747), S. Mithila (21251A1759) in the partial fulfillment for the award of B.Tech degree in Electronics and Telematics Engineering, G. Narayanamma Institute of Technology & Science, Shaikpet, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the Mini Project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide	Head of Department
N. Rama Krishna	Dr. Rajkumar L Biradar
Assistant Professor	Professor and HOD
Department of ETM	Department of ETM

EXTERNAL EXAMINER

8-1-297/2/I, Shaikpet, Hyderabad - 500 104, Telangana
: 040-2956 5856 / 2956 7756 E-mail : principal@gnits.ac.in Website : www.gnits.ac.in

ACKNOWLEDGEMENTS

I sincerely express my thanks to my internal guide, **N. Rama Krishna** , Assistant Professor of Electronics and Telematics Engineering department. H**is** knowledge and his logical way of thinking have been of great value for us. His suggestions and personal guidance have provided a good basis for the project work.

We would like to express our thankfulness to **Dr. Rajkumar L Biradar**, Head of the Department of Electronics and Telematics Engineering, who gave us valuable suggestions, guidance and motivation to complete this project work.

We wish to express our thanks to our Principal **Dr. K. Ramesh Reddy**, who has been observed posing valiance in abundance forwards our individual to acknowledgement our project work, tendentiously.

We would like to thank our faculty and staff all those for helping us in the project work in one way or the other.

We would like to express my thanks to my friends and family members for their help and wishes for the successful completion of this project.

ABSTRACT

In Green Houses, precise environmental monitoring is crucial for optimizing crop yield and quality. This project introduces a Lora-enabled greenhouse monitoring system designed to enhance agricultural efficiency through wireless technology. The system integrates sensors for UV light intensity, gas detection, temperature and humidity (DHT11), soil moisture, and a motor for automated plant watering. The UV light sensor ensures plants receive optimal light exposure, while the gas sensor monitors environmental conditions for plant health. The DHT11 sensor tracks temperature and humidity levels critical for plant growth. Soil moisture sensors provide real-time data on soil conditions, enabling automated watering through a motorized system to maintain optimal moisture levels.

Utilizing Lora technology, the system wirelessly transmits sensor data to a centralized monitoring station. This enables real-time monitoring and remote management of greenhouse parameters, empowering farmers to make informed decisions and optimize resource allocation. By promoting sustainable agricultural practices, this project aims to increase productivity and resilience in greenhouse cultivation environments.

S/W Requirements: Arduino IDE

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

S. NO	ABBREVIATIONS		
1	AP	-	Access Point
2	AWGN	-	Additive White Gaussian Noise
3	BER	-	Bit Error Rate
4	BLAST	-	Bell Labs Layered Space Time Architecture
5	BLLST	-	Bell laboratories Layered space time
6	BPSK	-	Binary Phase Shift Keying
7	CC	-	Complete Complementary
8	CDD	-	Cyclic Delay diversity
9	CDMA	-	Code Division Multiple Access
10	CSI	-	Channel State Information

INDEX

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction:

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. LoRa is a long range, low power wireless platform that has become the wireless platform of Internet of Things (IoT). LoRa devices and networks such as the LoRaWAN enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, and disaster prevention. LoRa devices have amassed several hundred known uses cases for smart cities, homes and buildings, communities, metering, supply chain and logistics, agriculture, and more.
One such application of lora is LoRa enabled Greenhouse monitoring system.

A greenhouse is a safe haven for plants. It reduces the accessibility of insects and animals that have the potential to damage or destroy your plants. Greenhouse environment reduces exposure to extreme weather conditions such as torrential rain and droughts. Inadequate food supply due to high population growth is a global threat. Due to major issues like climate change soil erosion and biodiversity loss etc. Food production by conventional means is not sufficient. Therefore, agricultural production in a greenhouse with highly controlled environmental conditions is expected that the production contributes to a stable supply of agricultural commodities regardless of the weather.

In particular, the introduction of the Internet of Things (IoT) applications, every single step of agriculture or greenhouse production can be improved: from soil management to minimizing water consumption, from plant protection to animal health and farm automation. Smart devices located in fields are able to collect information and control processes at various production stages. Besides, the miniaturization of electronic components allows to implement IoT sensors with reduced form factor and energy consumption, monitoring many variables of interest, such as temperature, humidity, wind speed and direction, soil conditions, chemical concentrations, crop growth, and solar exposure. Among the available technologies in terms of power consumption and coverage range, one of the most adopted LPWAN technology is LoRaWAN, which offers wide network coverage, built-in security, low cost, and limited power consumption during operation. Lora can be used when there may be areas where no signal/coverage of cellular services available. Now suppose, a farmer who has big cropland spread in multi-hectares. A big greenhouse is built in it. In such case use/installation of above wired/wireless technology infrastructure will not cost efficient. In this case use of LoRa wireless technology can play important role in communication.

1.2 Evolution:

Electronic Sensors: Introduction of electronic sensors for temperature, humidity, and soil moisture.

Wireless Technology: Introduction of wireless sensors using Wi-Fi and ZigBee; centralized monitoring systems.

Early IoT Integration: Initial IoT platforms for remote monitoring; smartphone apps for real time alerts.

LoRa and LPWAN: Introduction of LoRa for log range, low power communication and scalable networks.

AI and Machine Learning: AI for predictive analytics and automation; machine learning for plant health predictions

1.3 Objectives:

Automated Systems: Implement automated climate control, irrigation to reduce the need for manual intervention and improve consistency

Remote Monitoring: Enable farmers to monitor and control greenhouse conditions remotely using smartphones or computers, providing flexibility and convenience

Technological Advancement: To push the boundaries of medical robotics and AI integration for better patient outcomes and advancements in medical technology.

1.4 Significance:

Optimal Growing Conditions: By continuously monitoring environmental parameters, LoRa-based systems ensure plants are always in optimal conditions, leading to higher yields and better-quality produce.

Adaptability: The flexibility of LoRa technology allows for integration with various sensors and devices, providing a customizable solution tailored to specific greenhouse needs.

Global Access: Remote monitoring enables farmers to manage their greenhouses from anywhere in the world, enhancing operational flexibility.

1.5 Applications:

1. **Environmental Monitoring**:

- **Temperature and Humidity**: Continuous monitoring of temperature and humidity to maintain optimal growing conditions.
- **CO2 Levels**: Tracking CO2 concentrations to ensure adequate levels for plant growth.

2. **Irrigation Management**:

- **Precision Irrigation**: Automated irrigation systems that provide the right amount of water based on real-time soil moisture data.
- **Water Usage Tracking**: Monitoring and analysing water usage to enhance water conservation efforts.

3. **Crop Monitoring**:

- **Growth Tracking**: Monitoring plant growth parameters such as height, leaf area, and fruit size to assess health and development.
- **Yield Prediction**: Using data analytics to predict crop yields and optimize harvesting schedules.

4. **Remote Monitoring and Control**:

- **Real-Time Alerts**: Providing real-time alerts for critical conditions such as extreme temperatures or equipment failures.
- **Mobile and Web Access**: Allowing farmers to monitor and control greenhouse conditions remotely using smartphones or computer.

5. **Integration with Other Systems**:

- **Weather Stations**: Integrating with weather stations to adjust greenhouse conditions based on external weather data.
- **Smart Farming Platforms**: Connecting with broader smart farming platforms to integrate greenhouse data with overall farm management.

6. **Research and Development**:

- **Experimental Trials**: Conducting experimental trials with different crops or growing conditions and analysing the results.
- **Innovation**: Using collected data to innovate and improve greenhouse management practices.

CHAPTER 2: LITERATURE REVIEW

1.A Survey on LoRa for Smart Agriculture: Current Trends and Future

Authors: Antonino Pagano, Daniele Croce, Ilenia Tinnirello, and Gianpaolo Vitale

Abstract:

This paper explores the use of LoRa technology in modern agriculture. We examine state-of-the-art smart farming solutions and assess LoRa's potential for various field applications. Specifically, we focus on irrigation systems, crop monitoring, tree care, and livestock management, each with unique requirements for network speed, sensor complexity, energy efficiency, and decision-making latency. We discuss LoRa's suitability for these scenarios, considering scalability, interoperability, network architecture, and energy efficiency. Finally, we outline future research directions and identify potential challenges that may shape future trends.

Methodology:

The agricultural sector faces significant challenges due to resource scarcity, extreme weather, population growth, and shrinking arable land. According to the FAO, by 2050, the world's population may reach 10 billion, demanding a 70% increase in food production. Smart agriculture, utilizing information and communication technologies (ICTs), offers a practical solution. IoT applications can enhance every aspect of agricultural production, from soil management to water conservation, plant protection, and livestock health.

Smart devices deployed in fields collect the data and monitor various production processes. Miniaturized sensors, with reduced size and energy consumption, track temperature, humidity, wind conditions, soil quality, chemical levels, crop growth, and solar exposure, along with potential damage from drought, hail, or flooding. IoT systems integrate this data and provide valuable insights, such as predictive analytics for efficient resource allocation. Despite these benefits, the adoption of smart agriculture systems is still in its early stages. Internet connectivity limitations in many rural areas hinder digitization.

Several communication protocols have been proposed, each with distinct characteristics in terms of cost, coverage, power consumption, and reliability. Low-power wide-area networks (LPWANs) offer a promising solution for smart agriculture. LoRaWAN, a popular LPWAN technology, provides extensive coverage, built-in security, low cost, and minimal power consumption. Its robust modulation enables efficient communication in rural areas using simple devices. LoRa has been widely adopted and tested in agriculture, monitoring environmental conditions, controlling actuators (like irrigation valves), and supporting applications like tractor communication, livestock monitoring, and location tracking.

Unlike existing surveys that either combine LoRa with other IoT solutions or delve into specific technical aspects, this paper offers a comprehensive and focused analysis of LoRa/LoRaWAN applications in smart agriculture. We provide a detailed overview of advancements and in-field applications of this IoT technology.

Challenges:

As researchers dive into the world of smart agriculture, they recognize that digitizing farming processes is no small feat. Here's a closer look at the main challenges:

1) **Communication Issues:** For smart farming technologies to really take off, we need reliable and secure communication networks. The ability to share data effectively is crucial, and that's where technologies like LoRa/LoRaWAN come in to help bridge the gap.

2) **Energy Management**: Powering smart devices is a big hurdle. We need to find smart ways to keep sensors running efficiently. With sensors using a lot of energy, especially when performing edge computing tasks, optimizing power consumption and ensuring effective energy storage becomes essential.

3) **Data/Device Diversity**: Farms use a wide range of sensors—from those measuring soil conditions to weather and plant health. These devices often operate on different networks and protocols, making it tough to integrate them. This variety can be overwhelming, especially for farmers who may not be educated.

4) **Physical Deployments:** Setting up devices across large farming areas poses a significant challenge. Each part of the farm—whether it's soil, crops, or livestock—needs careful monitoring, which requires strategic placement of these technologies.

5) **Data Management:** Interpreting the sheer volume of data from multiple sensors can be daunting. Big data analysis might be necessary to navigate the unpredictability of agricultural ecosystems, but it can also feel overwhelming for farmers.

6) **User-Friendly Platforms**: For smart agriculture to truly take hold, we need intuitive software that farmers can easily use. The challenge lies in creating a flexible platform that can adapt to various needs, whether monitoring crops or livestock.

Alongside these challenges, there are additional hurdles like the costs of new technology, the complexity of these systems, gaps in farmers' training, and concerns about data ownership, privacy, and security. These factors have sparked innovation in developing new technologies and platforms tailored for smart agriculture.

CHAPTER 3: FUNCTIONAL DESCRIPTION OF THE PROJECT

The main goal of this project is to create a greenhouse monitoring system using LoRa technology, which operates autonomously and communicates with a receiver and sensor values are monitored on a web page. This system will continuously track important environmental factors like temperature, humidity, soil moisture, and UV light intensity. To ensure accurate results, we'll take a careful, step-by-step approach. The monitoring system will be made up of a network of sensors working together, a microcontroller, and communication modules working together to provide real-time data.

The sensors collect critical information about the greenhouse environment, which is processed by the microcontroller to control automated systems, such as irrigation and climate control. The data is transmitted over LoRa will be used to enable long-range communication while keeping power consumption low. The system is designed using a microcontroller (such as ESP 8266), various environmental sensors, and a LoRa module for effective communication. By leveraging this technology, the greenhouse monitoring system can optimize resource usage, enhance crop yields, and provide valuable insights for informed decision-making in agricultural practices.

3.1 BLOCK DIAGRAM:

Figure 3.1 (a) and Figure 3.1 (b) presents the block diagram of the project, which features an ESP 8266 which is a microcontroller acts as the central processing unit for the entire system. The microcontroller controls all the sensors, including a DHT11 sensor which tracks the greenhouse's temperature and humidity, a MQ135 sensor that monitors the air quality, GUVA S12SD to measure UV light. To transmit sensor values a communication module LoRa SX1278 is interfaced with ESP8266. At the receiver end another microcontroller ESP 8266 is connected to a LoRa module to collect the sensor data.

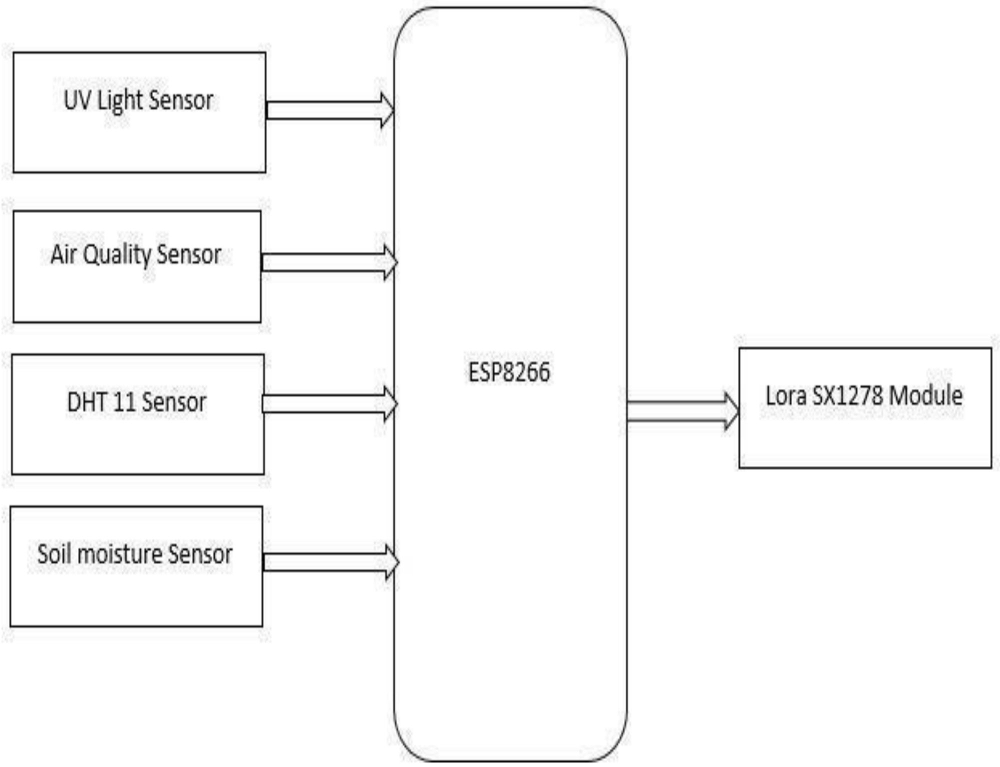


Fig3.1(a)- Transmitter Block Diagram

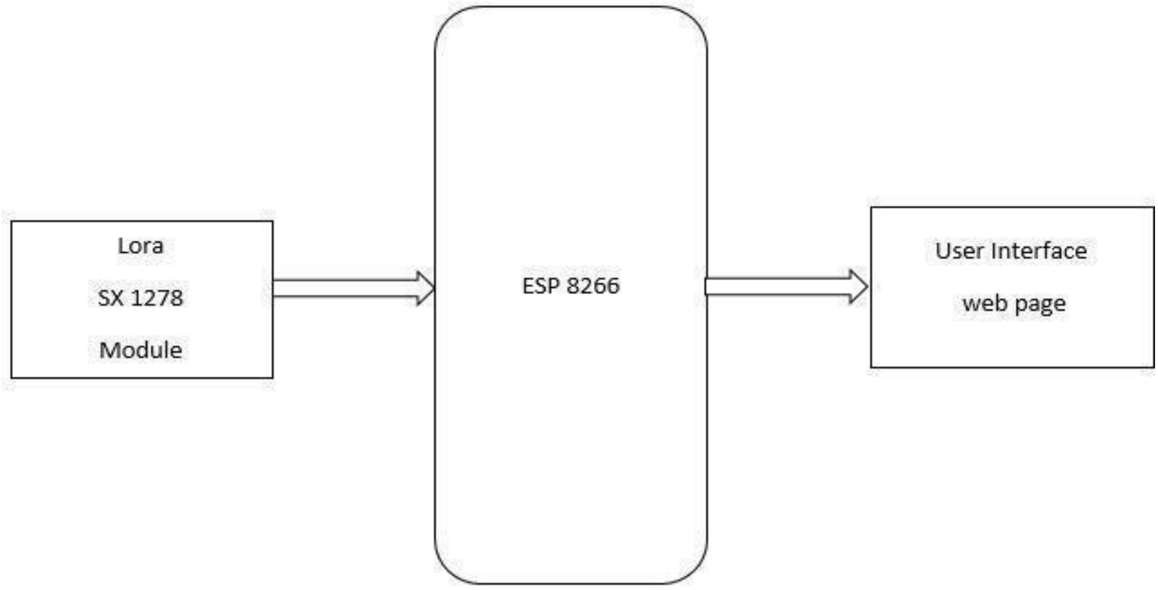


Fig3.1(b): Receiver block diagram

The LoRa receiver is a crucial component in a Long Range (LoRa) communication setup, designed to capture and decode signals sent over long distances while using minimal power. This receiver uses advanced modulation techniques to achieve that to high sensitivity, enabling it to detect faint signals even in challenging environments.

3.2 FLOWCHART

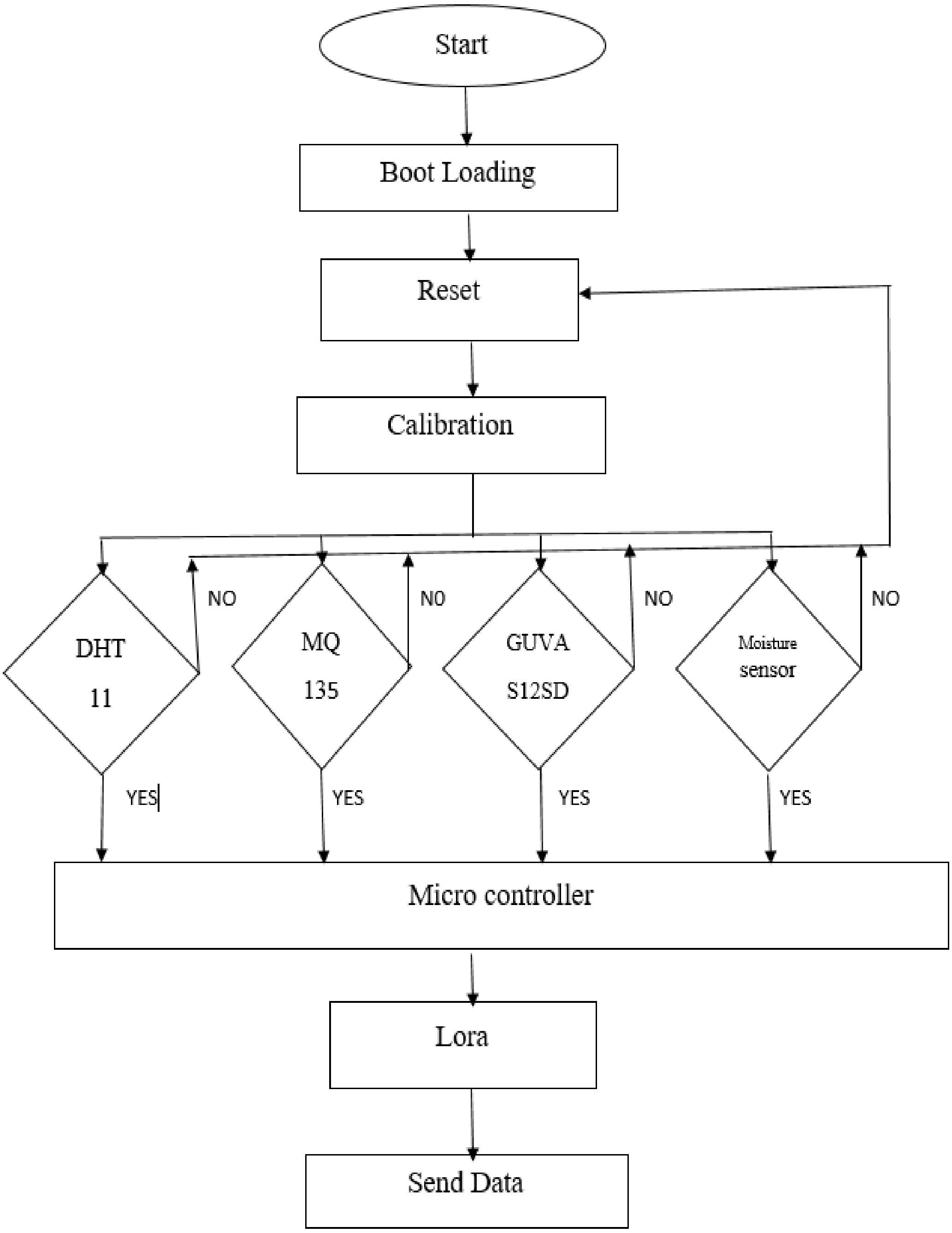


Fig3.2(a)-transmitter

A flowchart transmitter is a pivotal component in a communication system designed to encode and transmit data efficiently through sequential steps. It employs a visual representation of processes and decision points, facilitating the clear and systematic flow of information from input to output.

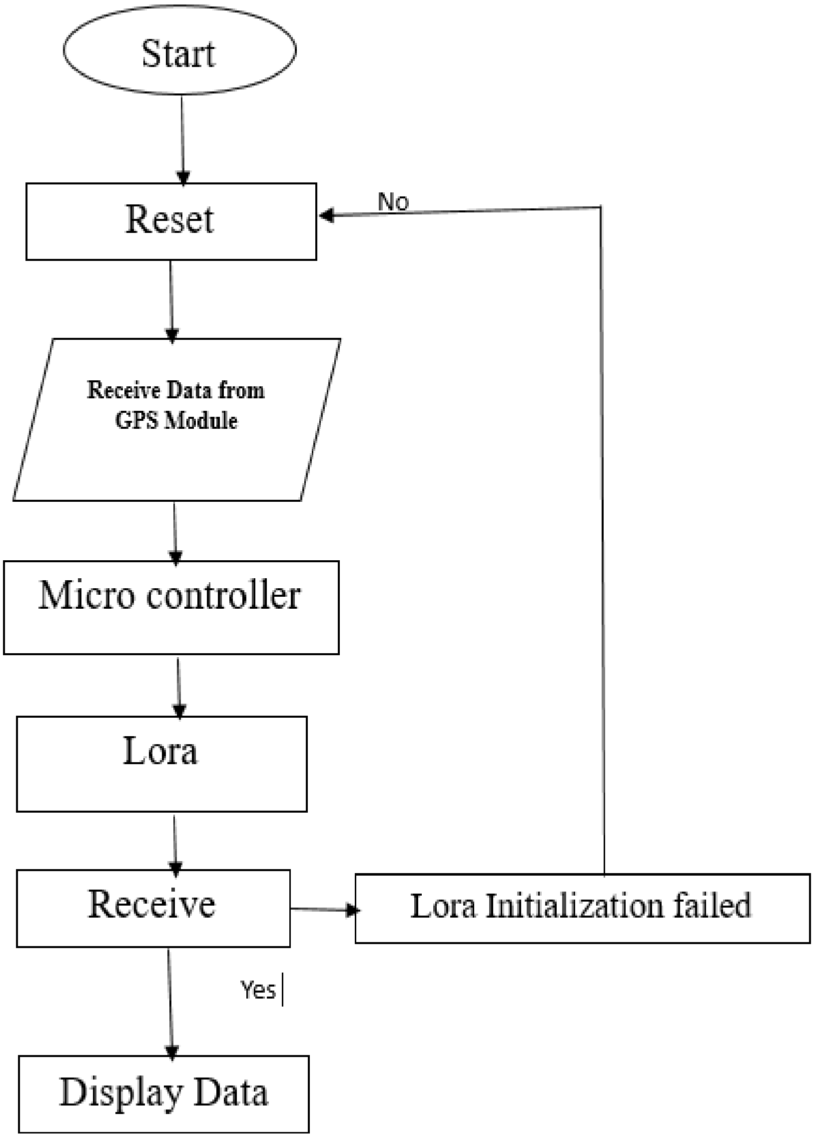


Fig 3.2(b)-receiver

a critical component in a communication system designed to receive, decode, and process incoming data in a structured and systematic manner. It visually represents the sequence of steps involved in receiving and interpreting signals, ensuring efficient and reliable data retrieval from the communication channel.

CHAPTER 4: HARDWARE DETAILS

4.1 ESP 8266 BOARD

In this project, we had used ESP 8266 to expand its capabilities by integrating various sensors and Lora module, allowing us to create a responsive and interactive system.

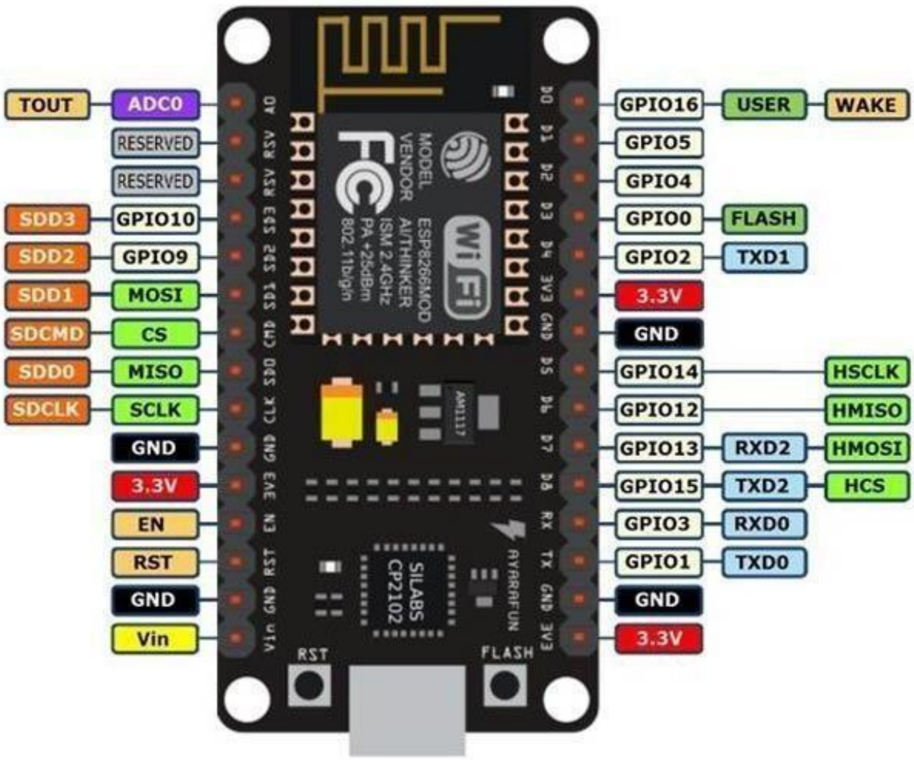


Fig4.1-ESP 8266 Board

Key components of an ESP 8266 board.

- **Microcontroller**: ESP8266
- **USB port**: Used for connecting the ESP8266 to a computer for programming and power.
- **USB to Serial chip**: Integrated into the ESP8266 module to facilitate communication with a computer via USB.
- **Digital pins**: The ESP8266 has GPIO (General Purpose Input/Output) pins that can be set up to work as either digital inputs or outputs.

- **Analog pins**: Some GPIO pins on the ESP8266 can read analog values using the ADC (Analog-to-Digital Converter) with varying resolution.
- **3.3V pin**: It Provides 3.3V of output to power external components.
- **GND:** Ground pin for completing electrical circuits.
- **VIN**: Input voltage pin, typically used to connect an external power supply (e.g., 5V) to power the ESP8266.

4.2 LoRa SX1278

The LoRa SX1278 is a wireless communication module that uses Long Range (LoRa) spread spectrum modulation technology, which utilizes the SX1278 chip from Semtech, that enables long-range communication with keeping power consumption low. The module works in the ISM (Industrial, Scientific, and Medical) bands, usually around 433 MHz or 868/915 MHz. depending on the region, offering robust communication over long distances, even in challenging environments. It is commonly used in IoT (Internet of Things) applications for transmitting data wirelessly over several kilometers without requiring a complex network infrastructure, making it perfect for applications such as remote sensor monitoring, agriculture, smart cities, and more. The SX1278 module supports various modulation schemes and features like CRC error checking, which ensures reliable data transmission over extended ranges.

fig4.2- LoRa SX1278

4.3 DHT11 SENSOR

The DHT11 is an affordable digital sensor that measures temperature and humidity. It can be easily connected to any microcontroller, like an Arduino, to provide real-time readings.

You can find the DHT11 as both a standalone sensor and as a module. The main difference is that the module includes a pull-up resistor and a power-on LED. As a relative humidity sensor, the DHT11 uses a thermistor along with a capacitive humidity sensor to gauge the moisture in the air.

The DHT11 features four pins: VCC, GND, a Data Pin, and one pin that's not connected. For communication between the sensor and the microcontroller, a pull-up resistor of 5k to 10k ohms is include.

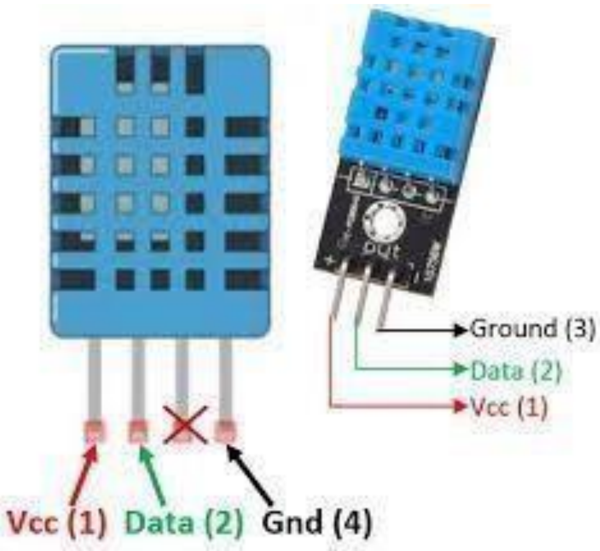


fig4.3-DHT11 sensor

4.4 MQ 135 sensor

The MQ-135 sensor is a versatile gas sensor designed to monitor air quality. It’s commonly used to measure the concentration of different gases in the atmosphere. Here’s a quick overview of the MQ-135 sensor:

The MQ-135 sensor uses the metal oxide semiconductor (MOS) principle and is sensitive to a wide range of gases, including ammonia (NH3), sulfide (S), benzene (C6H6), carbon monoxide (CO), and more. It works by heating a sensitive layer made of tin dioxide (SnO2), which alters its electrical resistance based on the concentration of gases in the air. This change in resistance is then measured and converted into an analog signal that can be easily read by a microcontroller or other electronic devices.

Applications of the MQ-135 sensor include air quality monitoring in indoor environments, detection of harmful gases in industrial settings, and integration into IoT devices for smart home applications. It provides a low cost solution for monitoring air pollution and ensuring safety in various environments where gas detection is crucial.

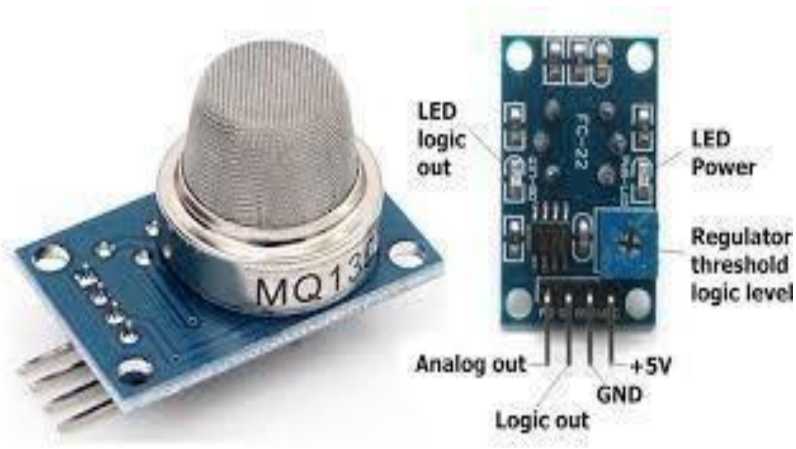


Fig 4.4-MQ135 sensor

4.5 GUVA S12SD

The GUVA-S12SD sensor is a compact ultraviolet (UV) sensor module that detects UV light in the 200-400nm wavelength range. Here's a brief overview of how the GUVA-S12SD sensor works:

UV Detection: The sensor contains a photodiode that is sensitive to UV light. When UV radiation falls on the photodiode, it generates a small current related to the intensity of the UV light.

Amplification and Filtering: The sensor module typically includes circuitry to amplify the small current generated by the photodiode. This amplification stage helps to enhance the signal-to-noise ratio and enhances the sensor's sensitivity to UV light.

Output Signal: The amplified signal is then processed and output as an analog voltage signal. The output voltage is the signal produced related to the UV light intensity detected by the sensor.

Calibration: To obtain accurate UV intensity measurements, the sensor may require calibration against known UV light sources or standard reference values. This calibration ensures that the output voltage corresponds correctly to the UV light intensity in the environment.

Fig 4.4 – GUVA SA12SD

4.6 Soil Moisture Sensor

A soil moisture sensor is made to measure how much water is present in the soil. Here’s a quick overview of how a typical soil moisture sensor operates:

Probe Design: The sensor contains of two or more electrodes that are placed into the soil. These electrodes are usually made of corrosion-resistant materials like stainless steel or coated with materials that prevent oxidation.

Measurement Principle: The sensor checks the electrical conductivity or dielectric permittivity of the soil, both of which are influenced by the amount of water present.

Capacitance Measurement (Dielectric Sensors): Many modern soil moisture sensors operate on the principle of capacitance. The sensor sends a high-frequency electrical signal between the electrodes. The presence of water in the soil changes the capacitance between the electrodes because water has a higher dielectric constant than dry soil.

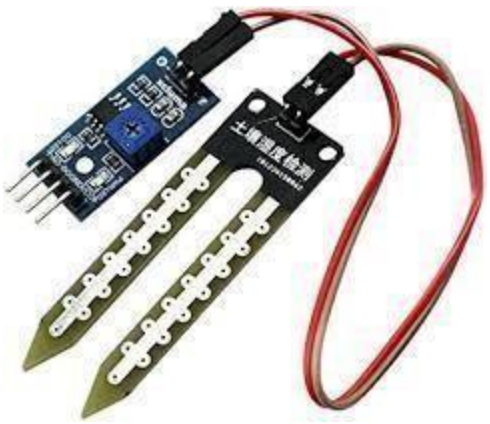


Fig 4.6-soil moisture sensor

CHAPTER 5: SOFTWARE OVERVIEW

5.1 ARDUINO IDE

The Arduino IDE (Integrated Development Environment) is an essential tool for programming ESP boards like the ESP8266. Here's a look at how the Arduino IDE functions and the basic steps involved in using it:

1. **Installation and Setup**:
   - Download and install the Arduino IDE from the Arduino website (https://www.arduino.cc/en/software).
   - After installation, connect your Arduino UNO board to your computer using a USB cable.

2. **Launching the IDE:**
   - Open the Arduino IDE software. It provides a user-friendly interface with various menus, buttons, and a text editor where you can write your code.

3. **Writing Code:**

- In the Arduino IDE, write program in a language similar to C/C++. The IDE simplifies the programming process by offering a set of pre-written libraries and functions specific to Arduino boards.

- code consists of a setup functions, which run once when the board is on power up or reset, and a loop function, which runs continuously thereafter.

4. **Verifying and Compiling:**

- Before uploading code to the ESP 8266, verify and compile it by clicking the "Verify" button (checkmark icon) or pressing `Ctrl + R`. This step checks for syntax errors and compiles code into a format that the Arduino can understand.

5. **Uploading Code:**

- After code compiles successfully, upload it to the Arduino UNO board by clicking the "Upload" (right arrow icon) or by pressing `Ctrl + U`.

- Arduino IDE communicates with the Arduino board via the USB connection and transfers the compiled binary code (hex file) to the board's microcontroller.

6. **Execution on ESP 8266:**

- After uploading, the ESP 8266 starts executing program immediately. the expected behavior based on the code wrote is shown.

- Check the board's serial output (if code includes `Serial. Print () ` statements) using the Serial Monitor in the Arduino IDE (`Ctrl + Shift + M`).

7. Debugging and Iteration:

- If there are issues or bugs in the program, the Arduino IDE provides tools for debugging. use `Serial. Print()` statements to output debugging information to the Serial Monitor.

Make changes to code, re-compile, and re-upload as needed until project functions as intended.

8. Libraries and Examples:

- The Arduino IDE includes a library manager where one can add and manage additional libraries that provide extra functionality (e.g., for specific sensors or communication protocols).
- It also offers a wide range of example sketches (code templates) that demonstrate how to use various Arduino functions and libraries.

9. Saving Projects:

- Save Arduino sketches (projects) locally on your computer

10. Additional Tools:

- The Arduino IDE supports advanced users with tools for directly accessing the microcontroller's registers (`Register` tab), configuring fuse bits (`Burn Bootloader`), and using external programmers (`Upload Using Programmer`).

In summary, the Arduino IDE makes it easy to write, compile, and upload code to Arduino boards. It provides an accessible environment for beginners while offering powerful tools for more advanced users to develop complex projects.

- PROGRAM –

Transmitter code: -

```
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>  // Include DHT library

#define SS    D8 // LoRa module pins
#define RST  D3
#define DI0  D2

#define BAND  433E6 // LoRa frequency band
```

```
#define ENCRYPT 0x78 // LoRa encryption sync word

#define DHTPIN D4 // DHT11 pin
#define DHTTYPE DHT11

#define MQ_PIN A0    // MQ135 analog pin
#define UV_PIN 10    // UV light sensor analog pin

#define TX_P  17    // LoRa TX power (example value, adjust as needed)

const int soilMoisturePin = D0; // Soil moisture sensor digital output connected to D0
const int ledPin = D1; // LED connected to D1
const int motorIn1 = D2; // Motor IN1 connected to D2
const int motorIn2 = D3; // Motor IN2 connected to D3

DHT dht(DHTPIN, DHTTYPE);  // Initialize DHT object

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Sender");

  // Initialize LoRa module
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setTxPower(TX_P); // Set LoRa TX power
  LoRa.setSyncWord(ENCRYPT);

  // Initialize DHT11 sensor
  dht.begin();
```

```
  // Setup for soil moisture sensor, LED, and motor
  pinMode(soilMoisturePin, INPUT);
  pinMode(ledPin,  OUTPUT);
  digitalWrite(ledPin, LOW);
  pinMode(motorIn1, OUTPUT);
  pinMode(motorIn2, OUTPUT);
  digitalWrite(motorIn1, LOW);
  digitalWrite(motorIn2, LOW);
}

void loop() {
  // Read DHT11 sensor data
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Read MQ135 sensor data
  int mqValue = analogRead(MQ_PIN);

  // Read soil moisture sensor data
  int soilMoistureValue = digitalRead(soilMoisturePin);

  // Read UV light sensor data
  int uvValue = analogRead(UV_PIN);

  // Prepare LoRa packet
  String data = "Humidity: " + String(humidity) + "%, Temperature: " +
String(temperature) + "C, MQ135: " + String(mqValue) + ", Soil Moisture: " +
String(soilMoistureValue) + ", UV Light: " + String(uvValue);

  // Send packet via LoRa
  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();
```

```
  Serial.println("Packet sent: " + data);

  // Control LED and motor based on soil moisture value
  if (soilMoistureValue == LOW) {
    digitalWrite(ledPin, HIGH); // Turn ON the LED
    digitalWrite(motorIn1, HIGH); // Turn ON the motor (clockwise direction)
    digitalWrite(motorIn2, LOW);
  } else {
    digitalWrite(ledPin, LOW); // Turn OFF the LED
    digitalWrite(motorIn1, LOW); // Turn OFF the motor
    digitalWrite(motorIn2, LOW);
  }

  delay(100); // Send data every 10 seconds
}
```

Receiver code :-

```
#include <SPI.h>
#include <LoRa.h>

#define SS    D8 // LoRa module pins
#define RST  D3
#define DI0  D2

#define BAND  433E6 // LoRa frequency band
#define ENCRYPT 0x78 // LoRa encryption sync word

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver");
```

```
  // Initialize LoRa module
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setSyncWord(ENCRYPT);

  Serial.println("LoRa Initializing OK!");
}

void loop() {
  // Try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // Received a packet
    Serial.print("Received packet with RSSI ");
    Serial.print(LoRa.packetRssi());
    Serial.print(": ");

    // Read the packet data
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
    }
    Serial.println();
  }

  delay(100);  // Delay to avoid continuous polling
}
```

CHAPTER 6: WORKING PRINCIPLES OF SENSORS

6.1 MQ 135 sensor

6.1.1 Working of MQ 135 sensor

The MQ-135 sensor operates by detecting different gases in the air through their interaction with a semiconductor material, typically tin dioxide (SnO_2). Here's a detailed explanation of its working principle:

Semiconductor Material: The MQ-135 sensor contains a sensitive layer of tin dioxide (SnO_2) as its main sensing element. Tin dioxide is a semiconductor material that exhibits changes in its electrical conductivity when exposed to different gases.

Heating Element: The sensor also includes a small heating element or coil embedded within the tin dioxide layer. This heating element raises the temperature of the tin dioxide to a level where it becomes more reactive to gases.

Interaction with Gases: : When the MQ-135 sensor is exposed to air containing certain gases (such as Ammonia (NH_3), Sulfide (S), Benzene (C_6H_6), Carbon monoxide (CO), etc.), these gases undergo a chemical reaction on the surface of the tin dioxide semiconductor.

Gas Adsorption: Gaseous molecules from the environment adsorb onto the surface of the tin dioxide. This adsorption changes the number of free electrons in the SnO_2 material, which in turn alters its electrical conductivity.

Resistance Change: The presence of different gases causes a change in the resistance of the tin dioxide layer. Each type of gas affects the resistance differently, resulting in a unique electrical response pattern for each gas.

Output Signal**:** The change in resistance of the tin dioxide sensor is converted into a corresponding change in voltage across the sensor terminals. This voltage change is then amplified and processed by the sensor circuitry.

6.1.2 Circuit of ESP 8266 MQ135 Sensor

The above image shows the circuit diagram of the ESP 8266 interfaced with MQ135. The sensor has a clip to inserting your finger and features three pins for connecting VCC, GND and the Data.

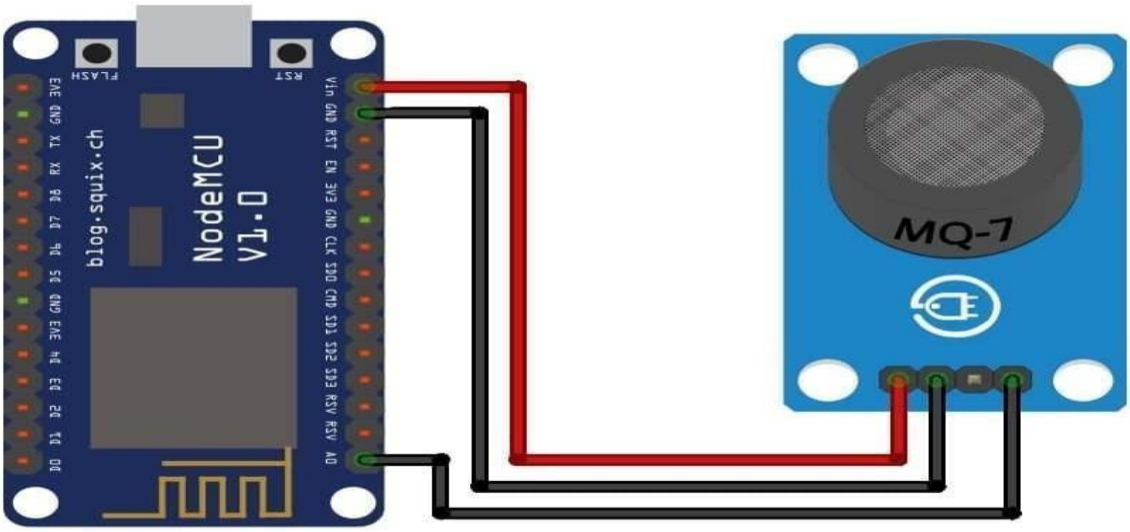


Fig 6.3- Circuit of ESP 8266 with MQ 135

Components required :

1. ESP8266 module (e.g., NodeMCU, ESP-01, etc.)
2. MQ-135 gas sensor module
3. Breadboard and jumper wires
4. Power source (e.g., USB cable for NodeMCU)

Procedure:

Power Connections**:**

ESP8266: Connect the 3.3V pin on the ESP8266 to the VCC pin on theMQ-135 sensor module.

MQ-135: Connect the GND pin on the MQ-135 sensor module to the GND pin on the ESP8266.

Ensure the power source can supply sufficient current for both the ESP8266 and the MQ-135 sensor.

Analog Output Connection:

MQ-135: Connect the analog output (usually labeled AO or AOUT) from the MQ-135 sensor module to one of the analog input pins (like A0) on the ESP8266..

This analog output provides a voltage signal proportional to the gas concentration detected by the MQ-135 sensor.

Digital Output Connection (Optional):

Some MQ-135 sensor modules also provide a digital output (DO) that indicates a certain threshold of gas concentration has been exceeded. If needed, connect the DO pin of the MQ-135 to a digital input pin on the ESP8266.

Additional Considerations:

Ensure the ESP8266 is properly programmed to read analog input values from the MQ-135 sensor. Use the appropriate ADC (Analog-to-Digital Converter) functions in your programming environment (Arduino IDE, etc.) to read the analog voltage from the sensor.

Calibrate the MQ-135 sensor if necessary, based on the gas concentrations you expect to measure. Calibration involves exposing the sensor to known concentrations of gases and adjusting your software to interpret the sensor readings correctly.

Programming:

Write a program for the ESP8266 to read the analog voltage from the MQ-135 sensor. Convert this analog voltage to a gas concentration using calibration factors provided in the MQ-135 sensor datasheet or calibration procedure.

Depending on your application, you may also handle digital output signals from the MQ-135 sensor to trigger events or alarms based on gas concentration thresholds.

6.2 DHT11 SENSOR

6.2.1 Principle of the DHT Sensor

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form. For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

The DHT11 sensor has four pins: VCC, GND, a Data Pin, and one that isn't connected. It also includes a pull-up resistor of 5k to 10k ohms to facilitate communication between the sensor and the microcontroller.

6.2.2 Working of DHT11 Sensor

DHT11 Humidity Sensor consists of 4 pins: VCC, Data Out, Not Connected (NC) and GND. The range of voltage for VCC pin is 3.5V to 5.5V. A 5V supply would do fine. The data from the Data Out pin is a serial digital data.

The following image shows a typical application circuit for DHT11 Humidity and

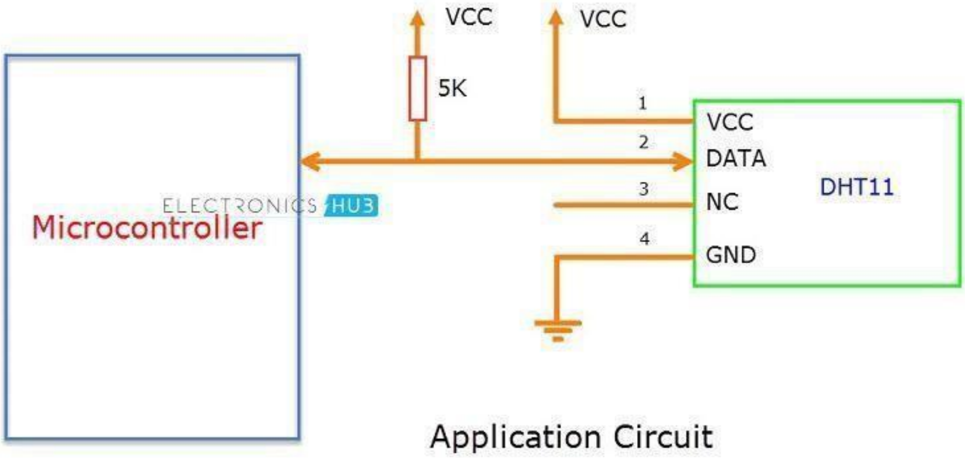


Fig 6.4- working of DHT sensor

Temperature Sensor: DHT11 Sensor can measure a humidity value in the range of 20 – 90% of Relative Humidity (RH) and a temperature in the range of 0 – 50^{o}C. The sampling period of the sensor is 1 second i.e.

https://www.electronicshub.org/wp-content/uploads/2017/06/DHT11-Application-Circuit.jpg

All DHT11 sensors are accurately calibrated in the lab, and their results are stored in memory. You can establish single-wire communication between any microcontroller, like an Arduino, and the DHT11 sensor. Plus, the cable length can be up to 20 meters.
The data from the sensor includes both integral and decimal parts for Relative Humidity (RH) and temperature. Specifically, the information consists of 40 bits formatted as follows: 8 bits for the integral RH value, 8 bits for the decimal RH value, 8 bits for the integral temperature value, another 8 bits for the decimal temperature value, and 8 bits for a checksum.

Example:

Consider the data received from the DHT11 Sensor is

00100101 00000000 00011001 00000000 00111110.

This data can be separated based on the above-mentioned structure as follows

00100101	00000000	00011001	00000000	00111110
High Humidity	Low Humidity	High Temperature	Low Temperature	Checksum (Parity)

In order to check whether the received data is correct or not, we need to perform a small calculation. Add all the integral and decimals values of RH and Temperature and check whether the sum is equal to the checksum value i.e. the last 8 – bit data.

00100101 + 00000000 + 00011001 + 00000000 = 00111110

This value is same as checksum and hence the received data is valid. Now to get the RH and Temperature values, just convert the binary data to decimal data.

RH = Decimal of 00100101 = 37%

Temperature = Decimal of 00011001 = 25^{0}C

6.2.3 Circuit of ESP 8266 Based DHT11 Sensor

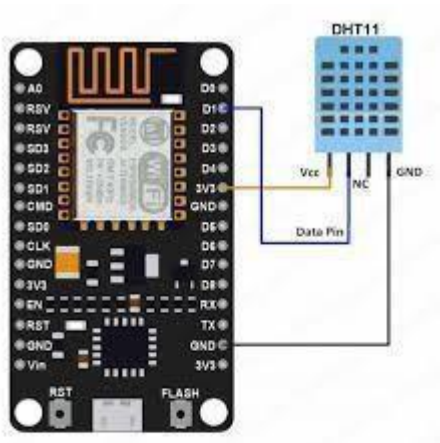


fig 6.5- Circuit of ESP 8266 Based DHT11 Sensor

Components Required:

1. ESP 8266
2. DHT11 Temperature Sensor
3. Breadboard
4. Power supply
5. Connecting wires

6.3 GUVA S12SD

6.3.1 working of GUVA S12SD

The GUVA-S12SD sensor is an ultraviolet (UV) sensor module that detects UV light in the 200-400nm wavelength range. Here's an overview of how the GUVA-S12SD sensor typically works:

Photodiode: The sensor contains a photodiode that is sensitive to UV light. A photodiode is a semiconductor device that generates a current when exposed to light.

UV Detection: When UV radiation from the sun or other UV sources falls on the photodiode of the GUVA-S12SD sensor, it generates a photocurrent. The magnitude of this photocurrent is directly relational to the intensity of the UV light hitting the photodiode.

Amplification and Filtering: The small photocurrent generated by the photodiode is typically very weak. Therefore, the sensor circuitry includes an amplifier stage to amplify this current to a measurable level. Additionally, there may be filters in place to ensure that only UV light in the desired wavelength range (200-400nm) is detected.

Output Signal: The amplified photocurrent is converted into an output voltage or current signal. This signal represents the intensity of the UV light detected by the sensor.

Calibration: To provide accurate UV intensity measurements, the GUVA-S12SD sensor may require calibration against known UV light sources or standard reference values. This calibration ensures that the output signal corresponds correctly to the UV light intensity in the environment.

6.4 Soil Moisture sensor

6.4.1 working of soil moisture sensor

A soil moisture sensor measures how much water is present in the soil. There are different types of soil moisture sensors, but one thing they all have in common works based on the principle of electrical conductivity or capacitance. Here’s how they typically work:

Sensor Design:

Soil moisture sensors typically have two or more electrodes or probes that are placed into the soil. These electrodes are often made of conductive materials like stainless steel or coated to prevent corrosion.

Capacitance Measurement Principle:

Many modern soil moisture sensors operate on the principle of capacitance. These sensors send high-frequency electromagnetic signals between the electrodes. The presence of water molecules in the soil affects the dielectric constant between the electrodes.

When the soil is dry, the capacitance between the electrodes is lower because dry soil has fewer water molecules, which have a lower dielectric constant.

When the soil is moist or wet, the capacitance increases because there are more water molecules present, which increases the dielectric constant between the electrodes.

Measurement Process:

The sensor measures the capacitance between the electrodes, which varies with the moisture content in the soil. This change in capacitance is then converted into a corresponding voltage or current signal by the sensor's circuitry.

Output Signal:

The output signal (voltage or current) is typically analog and varies linearly with the soil moisture content. Higher output values indicate higher moisture content in soil.

Calibration:

Soil moisture sensors may require calibration to account for differences in soil types and environmental conditions. Calibration involves comparing sensor readings with actual soil moisture measurements obtained by other methods (e.g., gravimetric method).

6.4.2 Circuit diagram of soil moisture sensor with ESP 8266

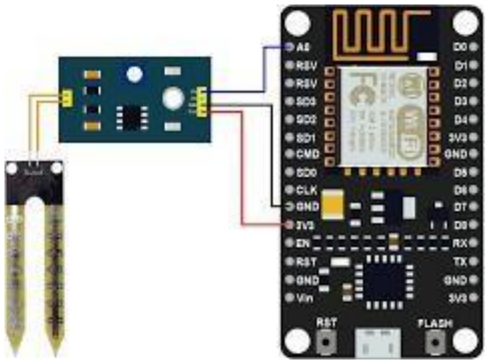


Fig 6.6 – circuit of ESP 8266 with soil moisture sensor

Materials Needed:

ESP8266 module (e.g., Node MCU, ESP-01, etc.)

Soil moisture sensor module (typically includes both analog and digital outputs)

Breadboard and jumper wires

Power source (e.g., USB cable for NodeMCU)

Procedure:

Power Connections:

ESP8266: Connect the 3.3V pin on the ESP8266 to the VCC pin on the soil moisture sensor module.

Soil Moisture Sensor: Connect the GND pin on the soil moisture sensor module to the GND pin on the ESP8266.

Ensure the power source can supply sufficient current for both the ESP8266 and the soil moisture sensor.

Analog Output Connection:

Soil Moisture Sensor: Connect analog output pin (typically labeled AO or AOUT) of the soil moisture sensor module to one of the analog input pins (e.g., A0) on the ESP8266.

This analog output provides a voltage signal proportional to the soil moisture level.

6.5 LoRa SX 1278

6.5.1 working of LoRa sx1278

The SX1278 is a popular LoRa transceiver module recognized for its long-range communication features in low-power applications. Here's detailed overview of how the SX1278 LoRa module typically works:

Frequency Bands and Regulations:

The SX1278 supports multiple frequency bands, typically 433MHz, 868MHz, or 915MHz, depending on the region and regulatory requirements.

It complies with various international standards and regulations for wireless communication.

Modulation Technique:

The SX1278 uses spread spectrum modulation techniques, specifically LoRa (Long Range), developed by Semtech. LoRa modulation allows for robust communication over long distances while using very little power.

LoRa modulation is characterized by chirp spread spectrum (CSS) technology, which provides significant advantages in terms of sensitivity and interference resilience compared to traditional modulation schemes.

Transceiver Architecture:

The SX1278 integrates both transmitter (Tx) and receiver (Rx) functionalities in a single chip.

Transmitter (Tx): Converts digital data into radio signals for transmission. It modulates the data using LoRa modulation and amplifies it to the desired output power level.

Receiver (Rx): Detects incoming radio signals, demodulates them, and decodes the digital data. It utilizes LoRa's unique demodulation algorithms to recover data from weak signals, making it suitable for long-range communication.

SPI Interface:

The SX1278 communicates with microcontrollers or other devices using a Serial Peripheral Interface (SPI).

Commands and configuration settings are exchanged between the microcontroller and the SX1278 module via SPI, allowing control over modulation settings, frequency selection, power management, and data transmission parameters.

Antenna and Range:

An external antenna is typically required for optimal performance. The antenna choice affects the module’s communication range and reliability.

The SX1278 can achieve communication ranges of several kilometers in open environments with line-of-sight conditions, depending on antenna configuration and environmental factor

6.5.2 Circuit diagram of ESP 8266 interfaced with lora SX1278

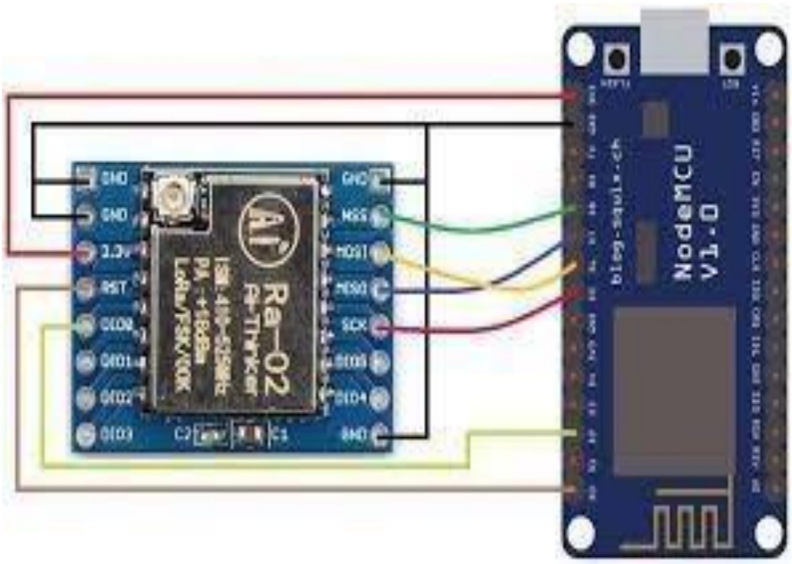


Fig 6.7 : circuit of ESP 8266 with LoRa sx 1278

To connect an SX1278 LoRa module with an ESP8266 module for communication, you need to establish both hardware and software connections. Here’s a step-by-step guide on how to do it:

Materials Needed:

1. ESP8266 module (e.g., NodeMCU, ESP-01, etc.)
2. SX1278 LoRa module (make sure it's compatible with your ESP8266's operating frequency, e.g., 433MHz, 868MHz, or 915MHz)
3. Antennas suitable for both modules (usually a wire or an external antenna for SX1278 and onboard antenna for ESP8266)
4. Breadboard and jumper wires
5. Power source (e.g., USB cable for NodeMCU)

Hardware Connections:

1. **Power Connections**:

   ESP8266: Connect the 3.3V pin on the ESP8266 to VCC pin on the SX1278 module.

   SX1278: Connect GND pin on SX1278 module to the GND pin on the ESP8266.Ensure both modules share a common ground.

2. **SPI Connections**:

   SX1278: Connect the above pins from the SX1278 module to the corresponding Pins on

   the ESP 8266:

   SX1278 VCC to **ESP8266 3.3**

   SX1278 GND to **ESP8266 GND**

   SX1278 MOSI to **ESP8266 GPIO12 (MOSI)**

   SX1278 MISO to **ESP8266 GPIO13 (MISO)**

   SX1278 SCK to **ESP8266 GPIO14 (SCK)**

   SX1278 NSS/SEL to **ESP8266 GPIO15 (CS)**

3. **Antenna Connections**:

Attach suitable antennas to both the SX1278 and ESP8266 modules to ensure optimal communication range. The SX1278 typically requires an external wire antenna.

Software Configuration:

1. **Arduino IDE Setup**: Ensure you have the ESP8266 board package installed in the Arduino IDE. Install the necessary libraries for ESP8266 and SX1278 LoRa communication. Popular libraries include: **ESP8266WiFi** library for ESP8266 Wi- Fi communication. **SPI** a library for SPI communication and a LoRa library specifically for SX1278 LoRa communication.

2. **Programming**:

   Write a program in Arduino IDE that initializes the SX1278 LoRa module and establishes LoRa communication between ESP8266 modules.

   Use the LoRa library to configure the SX1278 module, set frequency, bandwidth, spreading factor, and coding rate according to your application requirements.

   Implement LoRa transmission and reception functions in the program to send and receive data between ESP8266 modules using LoRa communication.

3. **Testing and Integration**:

   Upload the compiled code to both ESP8266 modules. Open the serial monitor in Arduino IDE to debug and monitor the communication between ESP8266 and SX1278 LoRa modules. Verify that data transmission and reception are working as expected, ensuring proper connectivity and signal strength.

6.6 Energy Management:

Solar panels: Solar panels offer a renewable and sustainable energy source, enabling the greenhouse monitoring system to function independently from the grid. This is especially beneficial in remote or rural areas where electricity access might be limited or unreliable.

Fig 6.8: solar panels

HX 2S D 2O BMS:

The primary function of the HX 2S D 2O BMS is to protect the batteries from overcharging, over-discharging, and short circuits. In a greenhouse monitoring system powered by lithium-ion batteries, this ensures the batteries operate within safe voltage limits, prolonging their lifespan.

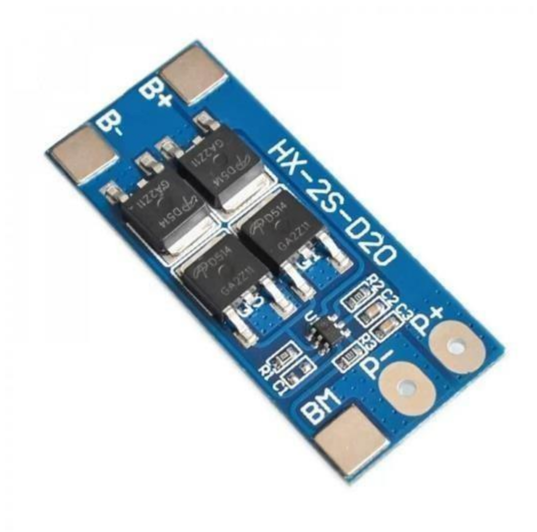
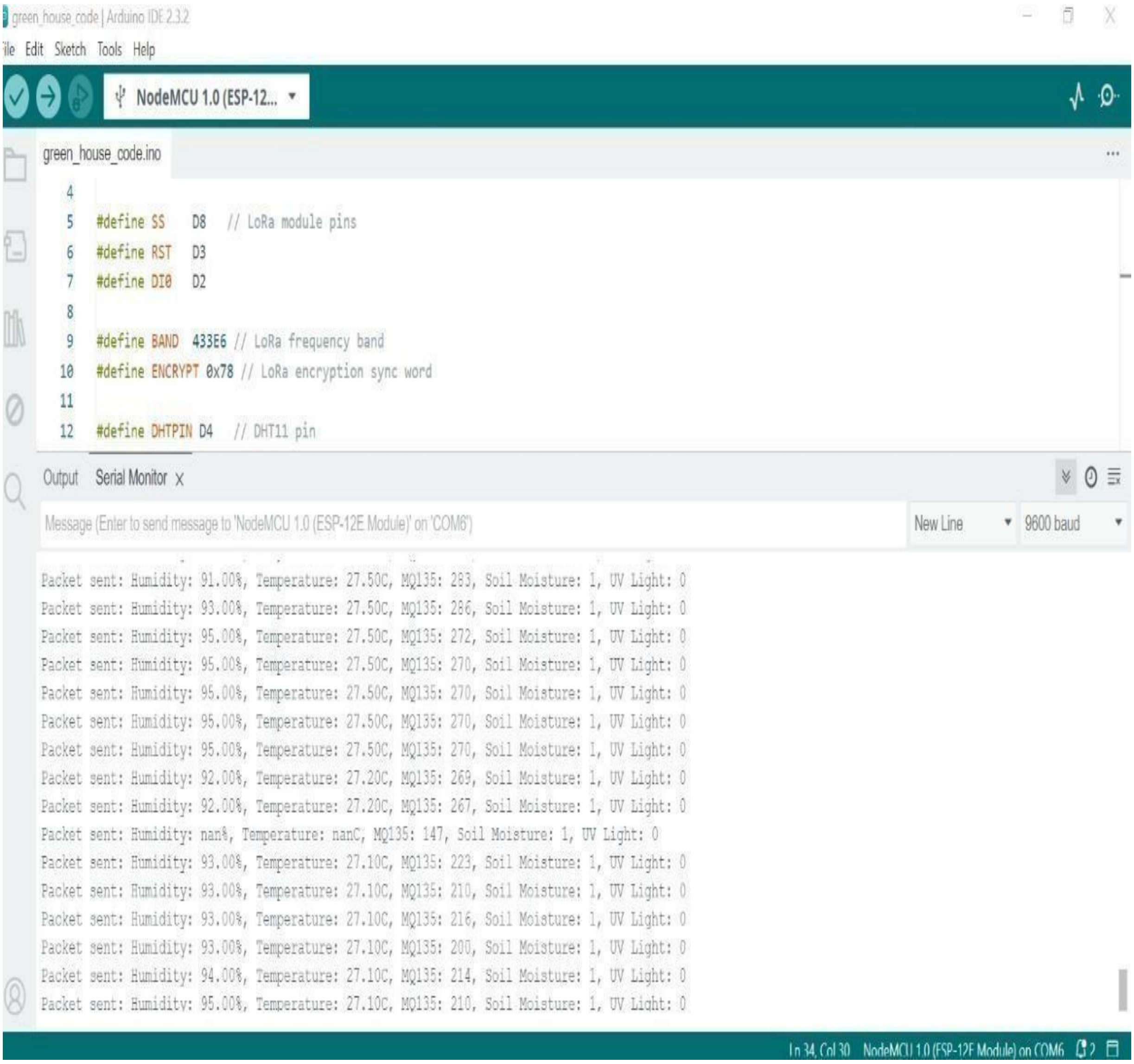


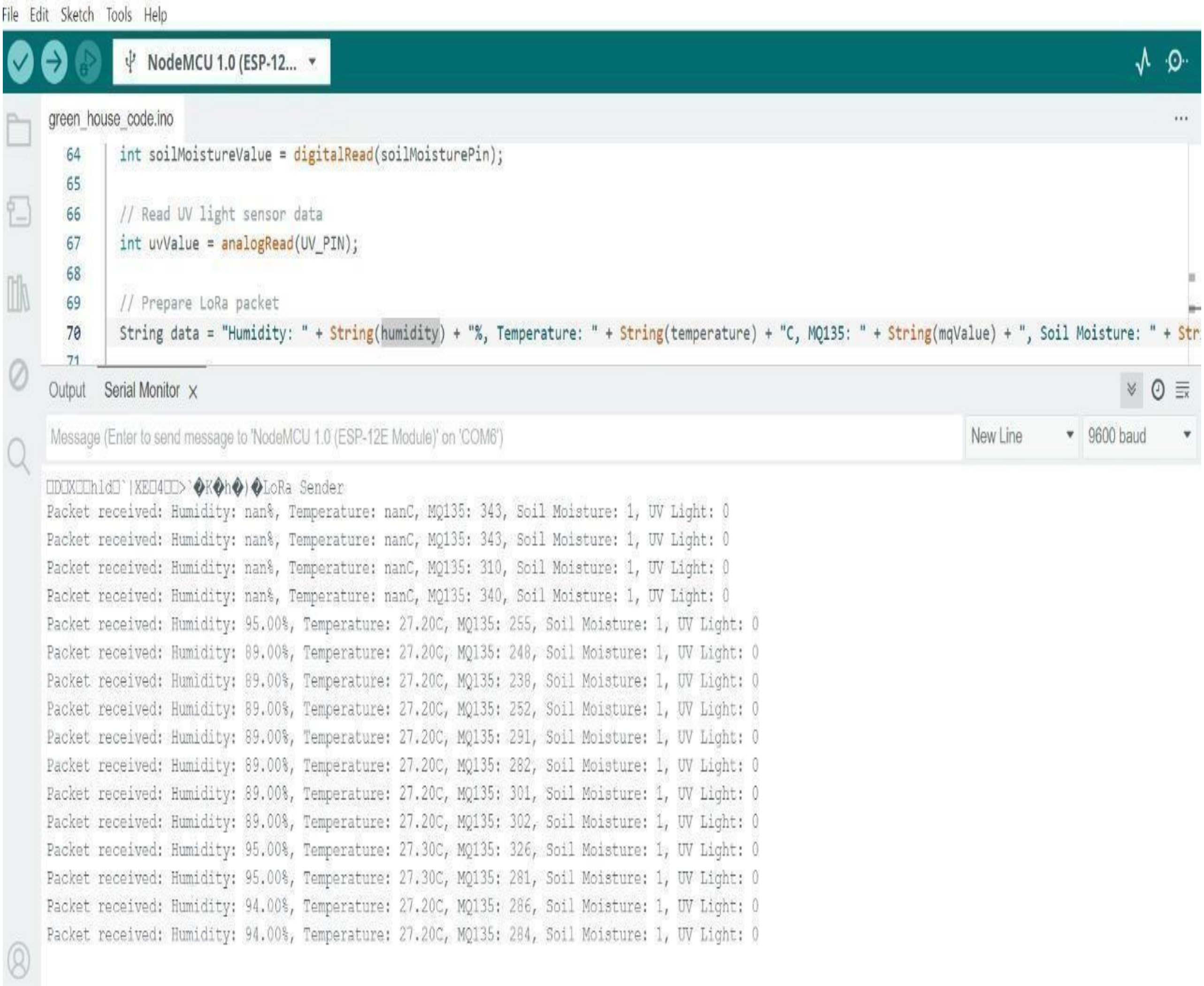
Fig 6.9 : HX 2S D 2O BMS:

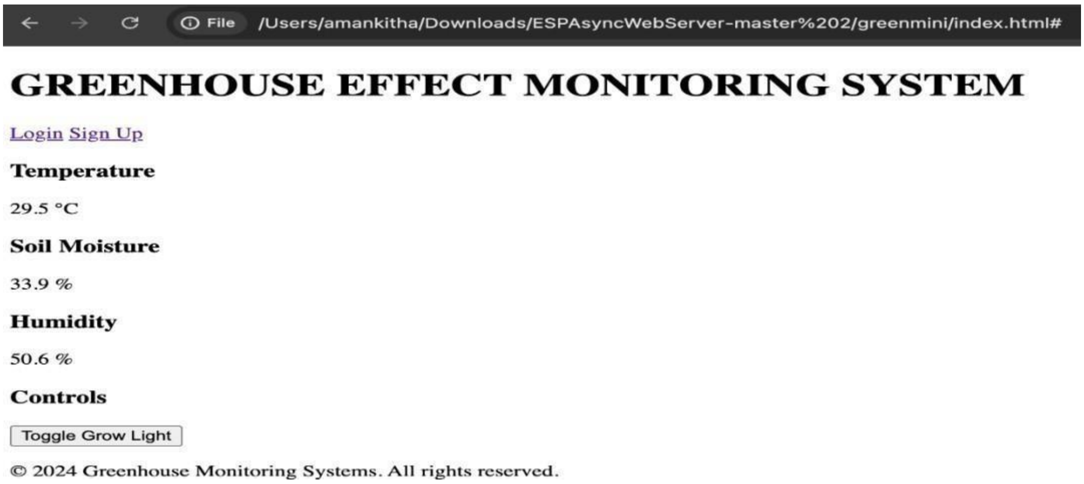
CHAPTER 7: RESULT ANANLYSIS

1. transmitter result: sensor values are sent in packets



2. Receiver result : packets are received

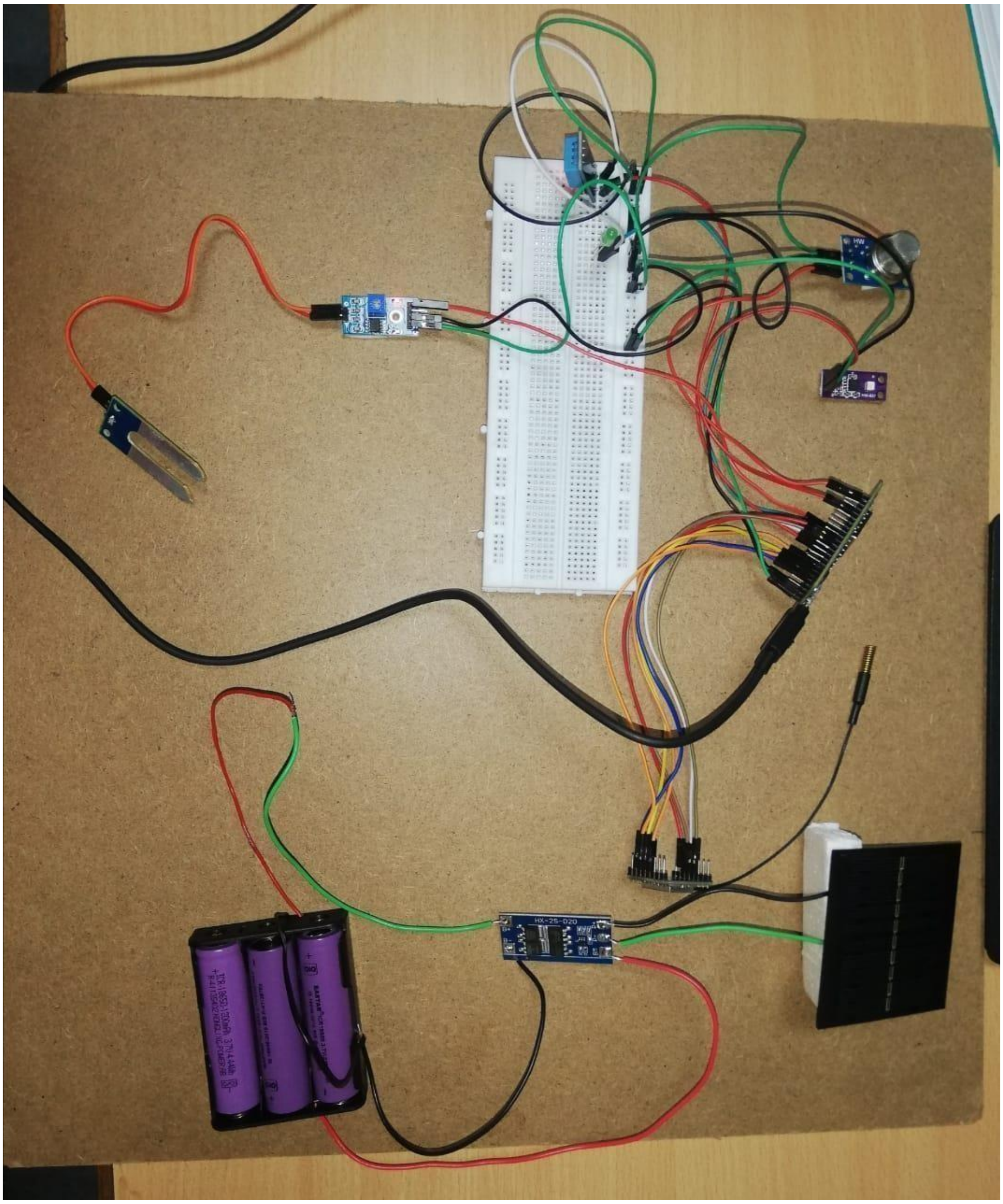




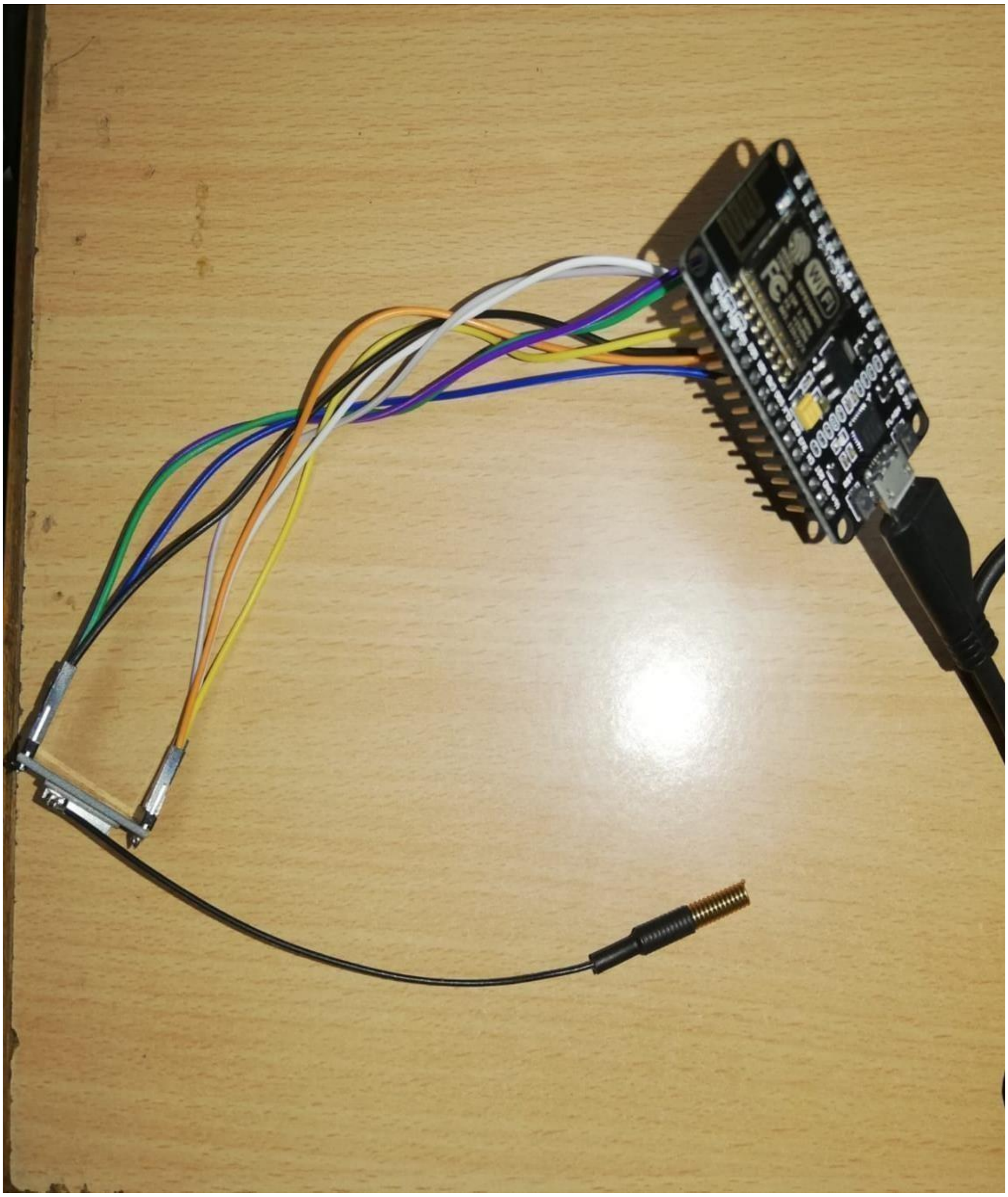
User Interface

Prototype model :

Transmitter :



Receiver :



CHAPTER 8: CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The development and implementation of the LoRa-based greenhouse monitoring system have been successfully completed, yielding satisfactory results in prototype demonstration. The project focused heavily on integrating advanced sensor technologies with robust communication capabilities to allow for real-time monitoring and management of greenhouse environmental factors.

Significant effort was dedicated to designing a reliable and scalable system architecture. Reference was made to current practices in agricultural IoT and LoRa technology to ensure the system's effectiveness in transmitting data over long distances while maintaining low power consumption.

Looking forward, the prototype serves as a foundation for future enhancements and scalability. Potential advancements include integrating artificial intelligence for predictive analytics, expanding sensor capabilities for comprehensive environmental monitoring, and leveraging cloud-based platforms for data aggregation and analysis.

In conclusion, the LoRa-based greenhouse monitoring system represents a significant advancement in smart agriculture technology. It not only enhances productivity through precise environmental control but also supports sustainable farming practices by optimizing resource utilization and reducing environmental impact. As the system evolves, it promises to revolutionize greenhouse management by leveraging cutting-edge IoT solutions tailored to agricultural needs.

8.2 FUTURE SCOPE

- **Precision Agriculture Techniques**:
  - **Sensor Integration**: Enhance the system with advanced sensors for precise monitoring of soil nutrients, pH levels, and plant health indicators (e.g., chlorophyll content).
  - **Data Fusion and Analytics**: Implement data fusion techniques to integrate sensor data with weather forecasts, soil analysis, and historical data for more accurate decision-making.
  - **Automated Control Systems**: Develop automated systems for irrigation, fertilization, and climate control based on real-time sensor data, optimizing resource usage and crop yield.

- **Remote Monitoring and Management**:
  - **Cloud Integration**: Integrate the monitoring system with cloud platforms for remote data storage, analysis, and access from anywhere.
  - **Mobile Applications**: Develop mobile apps for farmers to monitor greenhouse conditions, receive alerts, and remotely control operations using smartphones or tablets.

- **Environmental Sustainability**:

  - **Energy Optimization**: Adopt energy-efficient practices, like using renewable energy sources such as solar and wind, to power sensors and control systems.
  - **Water Management** Incorporate water monitoring sensors to optimize irrigation schedules based on soil moisture levels and weather forecasts consumption.

- **Predictive Analytics and AI**:

  - **Predictive Models**: Uses historical data and machine learning algorithms to develop predictive models for crop growth, pest management, and disease prevention.
  - **Smart Pest Management**: Integrate pest monitoring sensors and AI algorithms to detect pests early and implement targeted pest control measures.

- **Integration with IoT Ecosystem**:

  - **Smart Grid Integration**: Collaborate with utility providers to integrate greenhouse energy consumption data into smart grid systems for optimized energy distribution.
  - **Supply Chain Optimization**: Utilize IoT sensors for real-time monitoring of harvested produce during transportation and storage, ensuring quality and reducing waste.

- **Regulatory Compliance and Reporting**:

  - **Environmental Regulations**: Ensure the system complies with environmental regulations by monitoring and reporting greenhouse gas emissions and chemical usage.
  - **Certification Standards**: Implement features to track and maintain certifications standards (e.g., organic farming) through automated documentation and reporting.

PICTURE REFERENCES

1. Fig4.1:https://docs.arduino.cc/static/d0c28c5bd0894792476c6052dea5fa63/29114/board-anatomy.png

2. Fig4.2: Https://www.electronicshub.org/wp-content/uploads/2017/11/Heartbeat-Sensor-Image-7.jpg

3. Fig4.3: https://www.electronicshub.org/wp-content/uploads/2017/06/DHT11-Pinout.jpg

4. Fig5.1https://s.cafebazaar.ir/images/icons/de.kai_morich.serial_wifi_terminal-cf0900ed-200f-44ab-8201-ceca4a3ebefb_512x512.png?x-img=v1/format,type_webp,lossless_false/resize,h_256,w_256,lossless_false/optimize

5. Fig6.1: https://www.electronicshub.org/wp-content/uploads/2017/11/Heartbeat-Sensor-Image-9.jpg

6. Fig6.4: https://www.electronicshub.org/wp-content/uploads/2017/06/DHT11-Application-Circuit.jpg

7. Fig6.5: https://www.electronicshub.org/wp-content/uploads/2017/06/DHT11-Humidity-Sensor-on-Arduino-Circuit-Diagram.jpg

REFERENCES

[1]. ++Technology Entrepreneurship in Developing Countries: Role of Telepresence Robots in Healthcare. https://ieeexplore.ieee.org/document/9330532

[2]. Role of Robots in Healthcare Management. https://ieeexplore.ieee.org/document/8991181

[3]. Applications of Healthcare Robots in Combating the COVID-19 Pandemic. https://www.researchgate.net/publication/356485655_Applications_of_Healthcare_Robots_in_Combating_the_COVID-19_Pandemic

[4]. The role of robotics in medical science: Advancements, applications, and future directions. https://www.researchgate.net/publication/377328095_The_role_of_robotics_in_medical_science_Advancements_applications_and_future_directions.

[5]. Technological interventions for medication adherence in adult mental health and substance use disorders: a systematic scoping review (Preprint). https://www.researchgate.net/publication/329636296_Technological_interventions_for_medication_adherence_in_adult_mental_health_and_substance_use_disorders_a_systematic_scoping_review_Preprint.

[6]. On the Evaluation of DHT22 Temperature Sensor for IoT Application. https://ieeexplore.ieee.org/document/9467147.

[7]. DC Motor Drives. https://ieeexplore.ieee.org/document/7123302.

[8]. HOME AUTOMATION ON ESP8266. https://www.researchgate.net/publication/316926871_HOME_AUTOMATION_ON_ESP8266.

[9]. Analysis of Voltage-Regulator Operation. https://ieeexplore.ieee.org/document/1696687.

[10]. A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. https://www.sciencedirect.com/science/article/abs/pii/S1574013721000046

Appendix

We explore a system designed to monitor greenhouse conditions using LoRa (Long Range) technology. This setup includes various sensors placed throughout the greenhouse to track things like CO2 levels, temperature, and humidity. These sensors wirelessly send data to a central hub using LoRa nodes, which can transmit over long distances without draining much power. The system follows LoRaWAN standards for efficient communication. Data collected is securely sent to a cloud platform where it's analyzed in real-time and stored for historical reference. This technology helps farmers and researchers make informed decisions about managing their crops and understanding climate impacts. The appendix covers how we set up the system, manage data security, and our findings from using LoRa in this innovative way.