On

Smart Indoor Farming: Realtime Monitoring and control with STM32 and FreeRTOS

Submitted in the partial fulfillment of the requirements for the PG Diploma In

PG-Diploma in Embedded Systems and Design (PG-DESD)

Guided by: Mr. Ankush Tembhurnikar

Submitted by:

240844230039 – Shashikanth Kuruvabathini

240844230059 – Harish Pittala

240844230049 – Pallavi Deshmukh

240844230071 – Sai Hajare



Sunbeam Institute of Information Technology,

Pune and Karad

Year: 2024-25

This is to acknowledge our indebtedness to our Project Guide, Mr. Ankush Tembhurnikar Sunbeam, Pune for his constant guidance and helpful suggestions for preparing this project, Smart Indoor Farming: Realtime Monitoring and Control with STM32 and FreeRTOS. We express our deep gratitude towards inspiration, personal involvement, and constructive criticism that he provided us with technical guidance during this project.

We take this opportunity to thank the Head of the department for providing us with such a great infrastructure and environment for our overall development. We express sincere thanks to Process Owner for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards (Course Coordinators, PG-DESD) valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to C-DAC ACTS Pune, which provided us with this opportunity to carry out this prestigious Project and enhance our learning in various technical fields.

240844230039 – Shashikanth Kuruvabathini

240844230059 – Harish Pittala

240844230049 – Pallavi Deshmukh

240844230071 – Sai Hajare

The Smart Indoor Farming project aims to develop an intelligent and automated agricultural system that enhances crop growth by continuously monitoring and controlling environmental conditions. Using an STM32 microcontroller and FreeRTOS, the system efficiently manages critical parameters such as temperature, humidity, soil moisture, and light intensity. Real-time data from sensors enables precise control of irrigation, lighting, and ventilation, ensuring optimal growing conditions for plants. The integration of FreeRTOS allows for multitasking, enabling smooth execution of various functions, including data acquisition, processing, and automated decision-making.

Additionally, the system incorporates wireless connectivity for remote monitoring and control, providing farmers with real-time access to environmental data and system status via a user-friendly interface. This smart farming solution reduces manual intervention, optimizes resource usage, and improves overall agricultural efficiency. By combining embedded systems, automation, and IoT-based remote access, the project contributes to sustainable indoor farming, making it more adaptable, scalable, and environmentally friendly.

Furthermore, the system is designed to be scalable and customizable, allowing integration with additional sensors and actuators to meet specific crop requirements. Advanced data analytics and machine learning algorithms can be incorporated to predict trends and optimize farming strategies over time. The use of energy-efficient components ensures sustainability, making it suitable for urban farming, greenhouses, and research applications. By leveraging automation and real-time monitoring, this project not only enhances crop productivity but also contributes to the broader goal of sustainable and technology-drivenagriculture, reducing water and energy consumption while maximizing yields

Front Page I

Acknowledgement II

Abstract III

Contents IV

1. [Introduction](#_TOC_250017)…………………………………………………………………………….5
   1. [History](#_TOC_250016) 6
   2. [Problem Statement](#_TOC_250015) 6
   3. [Objective and Specification](#_TOC_250014) 6
2. [Literature Review](#_TOC_250013) 7
3. [Methodology](#_TOC_250011) 8
   1. [Block Diagram](#_TOC_250010) 9
4. [Proposed System](#_TOC_250009) ….10
   1. [Circuit Diagram](#_TOC_250008) 11
   2. Beagle Black Bone Pin Configuration 12
   3. Connection Configuration 12
   4. Hardware and Components………………………………………………………15
5. [Software 2](#_TOC_250007)2
   1. Cloud Platform 22
   2. [STMCubeIDE 2](#_TOC_250006)3
   3. [Arduino IDE 2](#_TOC_250005)4
6. [Communication Protocols 2](#_TOC_250004)5
   1. [HTTP Protocol……………………………………………………………………2](#_TOC_250003)5
   2. UART Protocol……………………………………………………………………25
7. Output 26
8. [Conclusion 2](#_TOC_250001)9
9. [Future Scope](#_TOC_250000) 30

# **Introduction**

With the increasing demand for sustainable and efficient farming solutions, Smart Indoor Farming has emerged as a promising approach to enhance agricultural productivity while minimizing resource wastage. Traditional farming methods often face challenges such as unpredictable weather conditions, inefficient resource utilization, and high labour dependency. Indoor farming, combined with automation and real-time monitoring, provides a controlled environment where crops can grow optimally regardless of external factors. This project focuses on developing a smart indoor farming system that leverages embedded systems and real-time operating technology to monitor and control critical environmental parameters such as temperature, humidity, soil moisture, and light intensity.

To achieve this, the system is built using an STM32 microcontroller and FreeRTOS, enabling efficient multitasking for sensor data acquisition, processing, and automated decision-making. The integration of wirelessconnectivity allows farmers to remotely monitor and control the system, reducing the need for manual intervention while improving operational efficiency. By automating irrigation, lighting, and ventilation based on real-time sensor data, this smart farming solution optimizes resource usage and enhances plant growth conditions. Additionally, its scalability and potential for integration with advanced analytics and machine learning make it a future-ready solution for modern agriculture. This project not only aims to improve farming efficiency but also contributes to the global goal of sustainable and technology-driven agriculture, making indoor farming more accessible and productive.

**Here are some of the benefits of using a smart indoor farming:**

* Smart indoor farming ensures efficient use of water, energy, and nutrients by automating irrigation, lighting, and climate control. This reduces waste and lowers operational costs.
* Unlike traditional farming, indoor farming allows cultivation throughout the year, independent of seasonal changes and external weather conditions, leading to higher and more consistent yields.
* Indoor farming can be implemented in urban areas, greenhouses, and vertical farms, making it a viable solution for food production in space-constrained environments.

**1.1 History**

The concept of indoor farming has evolved significantly over the years, driven by the need for sustainable food production and technological advancements. While traditional greenhouses have existed for centuries, modern smart indoorfarming integrates automation, sensor technology, and data analytics to optimize plant growth.

**1.2 Problem Statement**

Traditional farming methods face significant challenges due to climate change, unpredictable weather patterns, inefficient resource utilization, and high labour dependency. Outdoor farming is vulnerable to environmental fluctuations, leading to crop loss, inconsistent yields, andexcessive water and energy consumption.

Additionally, the increasing demand for food production in urban areas and space-limited environments requires innovative solutions to ensure sustainable agriculture.

While indoor farming provides a controlled environment for plant growth, existing systems often lack automation, real-time monitoring, and intelligent decision-making capabilities. Many conventional indoor farms rely on manual interventions, making them inefficient and resource-intensive. Furthermore, the absence of a scalable and remotely accessible solution limits farmers’ ability to optimize growing conditions effectively.

**1.3 Objective and Specification**

The primary goal of this project is to develop a Smart Indoor Farming System that enhances crop growth through real-time monitoring andautomatedcontrol using an STM32 microcontroller and FreeRTOS. The key objectives include:

* STM32 series MCU for efficient real-time processing and multitasking capabilities.
* FreeRTOS for task scheduling, sensor data acquisition, and efficient resource management.
* Continuously measure critical farming parameters such as temperature, humidity, soil moisture, and light intensity using sensors.
* Optimize water and energy consumption by automating processes and preventing resource wastage.

# 

# **2.Literature Review**

The concept of smart indoor farming has gained significant attention in recent years, driven by advancements in embedded systems, IoT, automation, andreal-time monitoring. Several research studies and technological developments have explored methods to enhance agricultural productivity through automatedenvironmental control systems. This literature review highlights key findings from previous studies related to indoor farming, sensor-based monitoring, embedded system applications, and FreeRTOS-based automation.

**Relevant Research Papers:**

* **“Smart Farming and IoT-Based Agricultural Systems”:** discusses how IoT-enabled systems can efficiently monitor environmental parameters like temperature, humidity, and soil moisture, allowing automated decision-making for irrigation and climate control.
* **“Embedded Systems in Indoor Farming”:** explores how STM32 microcontrollers provide efficient processing power and low energy consumption, making them ideal for continuous monitoring applications. Other research highlights the advantage of real-time task scheduling in FreeRTOS, which enables simultaneous execution of multiple farming operations such as sensor readings, data logging**.**
* **“Sensor-Based Environmental Monitoring in Agriculture”:** focus on the importance of multi-sensor networks for agricultural automation. Sensors such as DHT22 (temperature & humidity), capacitive soil moisture sensors, and BH1750 (light intensity sensors) have been widely used in research for indoor farming. Findings suggest that precise and continuous monitoring of these parameters leads to higher crop yield, reduced water consumption, and better resource management.
* **“FreeRTOS and Its Application in Smart Farming”:** describes how FreeRTOS allows efficient multitasking, priority-based scheduling, and low-power operation, making it suitable for applications where sensor readings, data processing, and actuator control must occur simultaneously. These findings validate the selection of FreeRTOS for smart indoor farming to ensure smooth, reliable, and real-time execution of farming tasks.

# **3.Methodology**

The **Smart Indoor Farming system** circuit diagram consists of the following key components:

**STM32F407 Board:**

* + - The STM32 collects real-time data from environmental sensors, including temperature, humidity, soil moisture, and light intensity.
    - It processes this information to determine optimal farming conditions.

**Ultrasonic Sensor HC-SR04**:

* Primarily used in water level monitoring.
* The sensor can also be used to measure the height of plants over time, providing useful data on plant growth rates.

**NodeMCU ESP8266:**

* NodeMCU provides Wi-Fi capability, allowing real-time data transmission from the STM32 microcontroller to a remote server.
* Provides real-time monitoring and Wi-Fi connectivity.

**HTTP-Based Communication Module (ESP8266 Module)**:

* HTTP protocol acts as the communication backbone for data exchange between the smart farming system, cloud services, and user interfaces.
* It enables real-time monitoring, remote control, cloud integration, and web-based interaction, making the system more accessible, automated, and IoT-enabled.

**Power Supply**:

* 5V and GND connections provide power to all components.
* Proper voltage regulation ensures smooth operation.

**Working Principle:**

* The STM32 microcontroller reads real-time environmental data from sensors (temperature, humidity, soil moisture, light intensity, and water level).
* FreeRTOS manages multitasking, processing sensor data
* The NodeMCU module transmits sensor data to a cloud server via HTTP
* Users can monitor data, send control commands

## Block Diagram

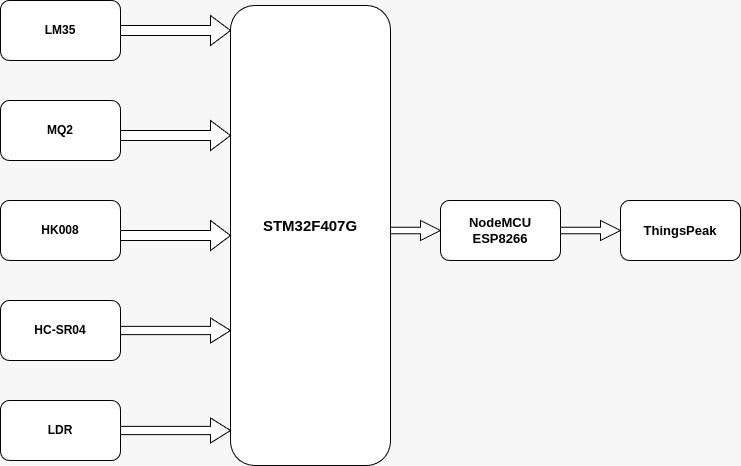


Figure: Block Diagram for proposed System.

## Connection Configuration

## **Hardware Setup:**

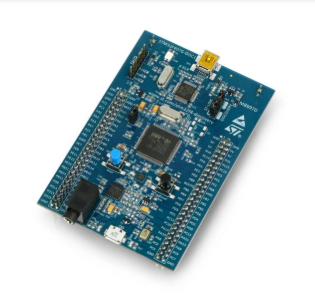
## 

## Output :

## 

## Hardware and Components

**STM32F407 Board**

****

The **STM32F407** is a high-performance microcontroller board based on the **STM32F407VGT6** chip from STMicroelectronics. It is widely used in embedded systems, automation, robotics, and real-time applications due to its **powerful ARM Cortex-M4 processor, high-speed peripherals, and extensive connectivity options**. **Key Features:**

* **Processor:** ARM **Cortex-M4** (32-bit RISC core) with **floating-point unit (FPU)** for high-performance computations.
* **Clock Speed:** Up to **168 MHz**, enabling fast execution of instructions.
* **Memory:**
  + **Flash Memory**: 1 MB for storing firmware.
  + **SRAM**: 192 KB for fast data storage.
* **Timers:** 17 timers (including general-purpose, PWM, and advanced timers) for precise task scheduling.
* **Communication Interfaces:**
* **USB 2.0 OTG (On-The-Go)** – Can act as both host and device.
* **Ethernet MAC** – Supports network connectivity.
* **UART (USARTs)** – 3× Universal Synchronous/Asynchronous Receivers/Transmitters for serial communication.
* **SPI/I2C** – Supports multiple SPI and I2C interfaces for sensor integration.
* **CAN (Controller Area Network)** – Useful for automotive and industrial applications.
* **SDIO (Secure Digital Input/Output)** – Supports SD card interfaces.
* **ADC (Analog-to-Digital Converters)**: 3× 12-bit ADCs (with up to 16 input channels) for high-speed sensor readings.
* **DAC (Digital-to-Analog Converter)**: 2× 12-bit DACs for applications requiring analog signal generation.
* **82 GPIO pins**, configurable as digital I/O, PWM outputs, or peripheral interfaces.
* **Operating Voltage**: **1.8V to 3.6V** (typically 3.3V for most applications).
* **Low Power Modes**: Multiple power-saving modes to optimize energy consumption.

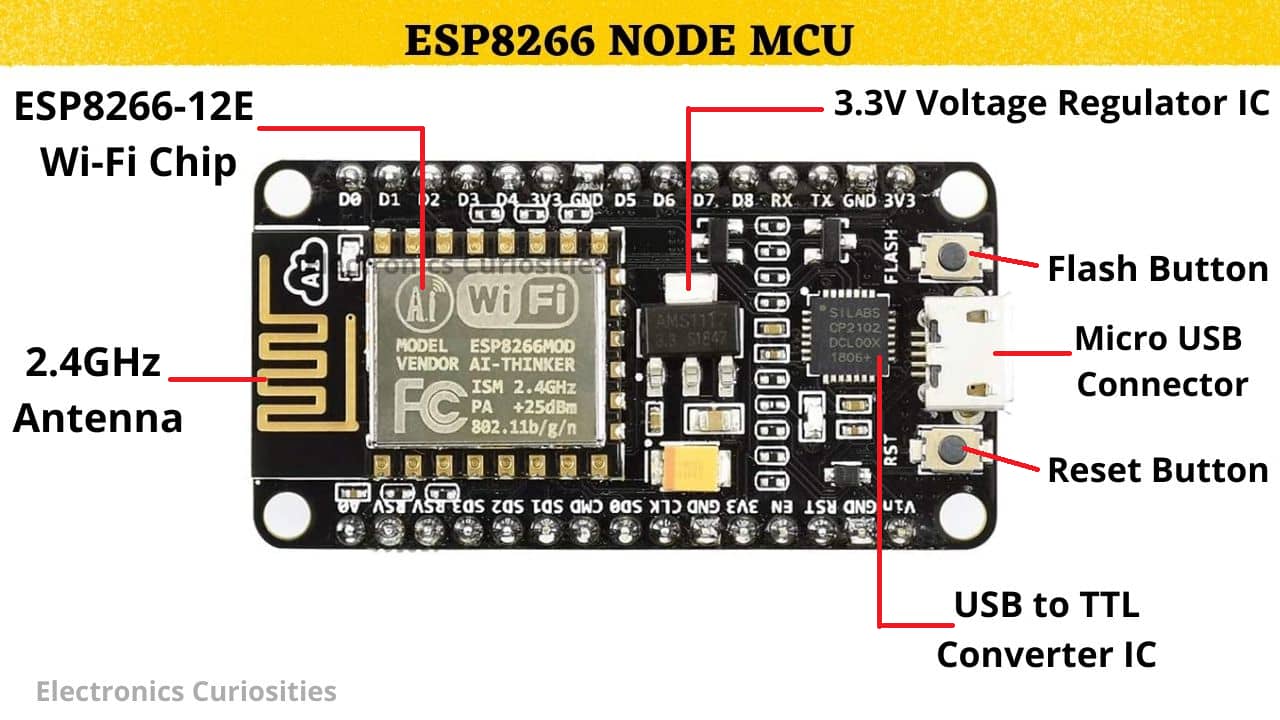
**Advantages of the STM32 Microcontroller:**

* **ARM Cortex-M4 Processor –** Runs at 168 MHz with a Floating-Point Unit (FPU), enabling fast and efficient computations for real-time applications.
* **Multiple Power Modes –** Includes sleep, stop, and standby modes to reduce energy consumption, making it suitable for battery-powered applications.
* **FreeRTOS Compatibility –** Enables efficient task scheduling, real-time sensor monitoring, and actuator control without performance bottlenecks.

**Use Cases:**

* **Real-time Sensor Monitoring –** Reads data from temperature, humidity, soil moisture, and light sensors to optimize crop growth.
* **Motor Control –** Uses PWM and high-speed timers to control DC motors, servos, and stepper motors in industrial machinery.
* **Smart Energy Management –** Monitors and controls energy consumption in homes and industries.
* **Industrial Automation:** Data acquisition, process control, and monitoring.

**NodeMCU**



**NodeMCU (ESP8266 Module):**

The NodeMCU ESP8266 is a low-cost, Wi-Fi-enabled microcontroller development board based on the ESP8266 SoC (System on Chip) from Espressif Systems. It is widely used in IoT (Internet of Things) applications, smart automation, and wireless sensor networks due to its built-in Wi-Fi, low power consumption, and ease of programming.3-Axis Acceleration Sensing: Measures acceleration along the X, Y, and Z axes, providing comprehensive motion information. Key Features:

* **Chipset:** ESP8266EX
* **Processor:** 32-bit Xtensa L106 running at 80 MHz (can be overclocked to 160 MHz).
* **RAM:** 64 KB for instruction storage, 96 KB for data storage.
* **Flash Memory:** Typically, 4 MB (SPI Flash), used for firmware and data storage.

**Sensors:**

1. **LM35 [Temperature Sensor]**

The LM35 is an analog temperature sensor that provides a linear output voltage proportional to temperature. It is widely used for temperature monitoring and control applications in agriculture, industrial automation, and environmental monitoring.

**Specifications:**

* Operating Voltage: 4V – 30V (Typically 5V or 3.3V).
* Temperature Range: -55°C to 150°C.
* Accuracy: ±0.5°C.
* Output: Analog voltage (10mV per °C).

**Working Principle**

* The LM35 generates an output voltage (Vout) that is directly proportional to the temperature in Celsius.
* The output voltage follows the formula:



* The microcontroller (STM32F407) reads this analog voltage through its ADC (Analog-to-Digital Converter) and converts it into a temperature value.

**Integration in the System**

* Connected to STM32F407 via an analog input pin.
* STM32 reads the temperature data, processes it, and determines whether cooling or heating systems should be activated.
* Data is transmitted to NodeMCU (ESP8266) for remote monitoring via a web dashboard.

1. **HC-SR04 [Ultrasonic Sensor]**

The HC-SR04 is an ultrasonic distance sensor used for measuring water levels in irrigation tanks and detecting obstacles in automation systems.

**Specifications**

* Operating Voltage: 5V.
* Detection Range: 2 cm to 400 cm.
* Accuracy: ±3 mm.
* Output: Digital (pulse width).

**Working Principle**

* The sensor sends ultrasonic sound waves through the TRIG pin.
* The sound waves bounce off an object (water surface) and return to the ECHO pin.
* The time taken by the waves to return is used to calculate the distance:

(Speed of sound ≈ 343 m/s).



**Integration in the System**

* Connected to STM32F407 via digital I/O pins (TRIG and ECHO).
* STM32 reads the distance data and calculates water levels in the irrigation tank.
* If the water level is low, the system activates the water pump automatically.
* Data is transmitted to NodeMCU (ESP8266) for remote water level monitoring.

1. **LDR [Light Dependent Resistor]**

An LDR (Light Dependent Resistor) is a light sensor used to measure the amount of light in the greenhouse and control artificial lighting.

**Specifications**

* Operating Voltage: 3.3V – 5V.
* Output: Analog resistance value (varies with light intensity).
* Light Sensitivity: Resistance decreases with increasing light intensity.

**Working Principle**

* When light intensity increases, the resistance of the LDR decreases, and vice versa.
* The voltage drop across LDR changes, which is read by the ADC of STM32.
* Based on light levels, the system turns ON/OFF artificial grow lights.

**Integration in the System**

* Connected to STM32F407 ADC pin for real-time light monitoring.
* STM32 processes the data and decides whether LED grow lights need to be activated.
* The data is transmitted to NodeMCU (ESP8266) for remote access.

1. **MQ2 [Gas Sensor]**

The MQ2 gas sensor detects harmful gases like CO₂, methane, LPG, and smoke, which can affect plant growth and safety.

**Specifications:**

* Operating Voltage: 5V.
* Detection Range: 200 – 10000 ppm.
* Output: Analog voltage (proportional to gas concentration).

**Working Principle:**

* The sensor contains a heating element and a metal-oxide (SnO₂) sensing layer.
* When gas molecules interact with the sensing layer, its resistance changes, producing a varying analog output voltage.
* STM32 reads this voltage and determines gas concentration levels.

**Integration in the System**

* Connected to STM32F407 ADC pin for gas level monitoring.
* If gas levels exceed a safe threshold, STM32 triggers an alarm or ventilation system.
* Data is sent to NodeMCU (ESP8266) for remote monitoring.

1. **Soil Moisture Sensor**

The soil moisture sensor measures water content in soil to optimize irrigation.

**Specifications:**

* Operating Voltage: 3.3V – 5V.
* Output: Analog voltage (proportional to soil moisture).
* Moisture Range: 0% (dry) to 100% (wet).

**Working Principle**

* The sensor measures soil resistance:
  + Wet soil – Low resistance (high conductivity).
  + Dry soil – High resistance (low conductivity).
* The analog voltage is read by STM32's ADC to determine moisture levels.

**Integration in the System**

* Connected to STM32F407 ADC pin for real-time soil monitoring.
* If soil moisture is too low, STM32 activates the irrigation system.
* Data is sent to NodeMCU (ESP8266) for remote tracking and automation.

**Cloud Platform:**

**Overview of ThingSpeak Cloud Platform**

ThingSpeak is an IoT cloud platform that enables real-time data collection, visualization, and analysis from connected devices. It supports HTTP, MQTT, and REST API communication, making it ideal for remote monitoring and automation. ThingSpeak allows users to store sensor data in the cloud, process it with MATLAB analytics, and trigger actions based on predefined conditions.

**How We Used ThingSpeak in Our Project**

1. Real-Time Data Monitoring – Sensor data (temperature, humidity, soil moisture, light levels, and water levels) is uploaded to ThingSpeak via NodeMCU (ESP8266) for remote access.
2. Data Visualization – ThingSpeak's built-in graphs and charts display sensor readings for easy analysis.
3. Alerts and Automation – Custom thresholds trigger alerts or control actions, like turning on irrigation when soil moisture is low.
4. Remote Access – Users can monitor farm conditions from anywhere via the ThingSpeak dashboard.
5. Data Logging – Historical data is stored and analyzed to optimize farming operations.

By integrating ThingSpeak with STM32F407 and NodeMCU, we achieve efficient, remote-controlled smart farming with real-time decision-making.

**Steps to Set Up a ThingSpeak Account**

Follow these steps to create and configure a ThingSpeak account for real-time data monitoring in our Smart Indoor Farming Project:

**Step 1: Create a ThingSpeak Account**

1. Go to [ThingSpeak Website](https://thingspeak.com/).
2. Click on "Sign Up" and create a MathWorks account (if you don’t have one).
3. Verify your email and log in to your ThingSpeak dashboard.

**Step 2: Create a New Channel**

1. Click on "Channels" > "My Channels" > "New Channel".
2. Enter a Channel Name (e.g., “Smart Farming Data”).
3. Enable the Fields (e.g., Field 1: Temperature, Field 2: Soil Moisture, etc.) according to the sensors used.
4. **Click "Save Channel".**

Step 3: Get API Keys for Data Communication

1. Go to the "API Keys" tab in your channel.
2. Copy the "Write API Key" (used to send data) and "Read API Key" (used to read data).
3. These API keys will be used in the NodeMCU (ESP8266) code to send sensor data to ThingSpeak.

**Step 4: Configure Data Visualization**

1. Go to the "Private View" tab and click "Add Widget".
2. Select "Line Chart", "Gauge", or other visualization tools.
3. Choose the corresponding sensor field (e.g., Temperature for Field 1).
4. Click "Save" to display real-time sensor data.

**Step 5: Connect ThingSpeak with NodeMCU (ESP8266)**

1. Install the ThingSpeak and ESP8266WiFi libraries in the Arduino IDE.
2. Modify the ESP8266 code to include the Wi-Fi credentials and API key.
3. Upload the code to NodeMCU and start sending data to ThingSpeak.

**Software Requirement:**

**1. STM32CubeIDE**

**Overview**

STM32CubeIDE is an official development environment from STMicroelectronics used for programming and debugging STM32 microcontrollers. It is an integrated development environment (IDE) that combines STM32CubeMX for peripheral configuration and Eclipse-based code editing with GCC compiler support.

**Features**

1. Supports STM32 family (including STM32F407).
2. Integrated STM32CubeMX for peripheral setup and initialization.
3. Built-in debugger with ST-Link support.
4. FreeRTOS support for real-time task management.
5. Code auto-generation for efficient development.

**Usage in Our Project**

1. Used to program the STM32F407 board to read sensor data.
2. Configures ADC (for LM35, soil moisture, and LDR), GPIO (for HC-SR04 and MQ2), and UART (for ESP8266 communication).
3. Implements FreeRTOS for managing multiple tasks like sensor reading and data transmission.
4. Debugs and tests the sensor integration and control mechanisms.

**2. Arduino IDE**

**Overview**

Arduino IDE is an open-source software used for programming microcontrollers like NodeMCU ESP8266. It provides a simple interface for writing, compiling, and uploading code using the C++-based Arduino programming language.

**Features**

1. Compatible with ESP8266 and other microcontrollers.
2. Supports libraries for Wi-Fi (ESP8266WiFi), HTTP communication, and ThingSpeak API.
3. Easy-to-use interface with serial monitor support.
4. Open-source with a large community and library support.

**Usage in Our Project**

1. Used to program NodeMCU ESP8266 for Wi-Fi communication.
2. Implements HTTP protocol to send sensor data to ThingSpeak cloud.
3. Reads data from STM32 via UART and uploads it to the cloud.
4. Provides real-time monitoring of farm conditions via a web dashboard.

**Communication Protocol:**

**Overview of Communication Protocols Used**

In our Smart Indoor Farming Project, we use two main communication protocols: UART and HTTP. These protocols enable seamless data exchange between the STM32 microcontroller, NodeMCU ESP8266, and the ThingSpeak cloud for real-time monitoring and automation.

**1. UART (Universal Asynchronous Receiver-Transmitter) Protocol**

**Overview**

UART is a serial communication protocol used for device-to-device communication. It enables asynchronous data transfer between microcontrollers and peripheral devices without requiring a clock signal.

**Key Features**

1. Asynchronous communication (no clock signal needed).
2. Uses TX (Transmit) and RX (Receive) pins for serial data exchange.
3. Baud rate configurable (e.g., 9600, 115200 bps).
4. Supports full-duplex communication.

**Usage in Our Project**

1. STM32F407 communicates with NodeMCU ESP8266 via UART.
2. Sensor data from STM32 is transmitted to NodeMCU, which then uploads it to the cloud.
3. Ensures efficient data exchange between microcontrollers without requiring additional interfaces.

**2. HTTP (Hypertext Transfer Protocol)**

**Overview**

HTTP is a request-response protocol used for web-based communication. It allows microcontrollers to send sensor data to cloud platforms (ThingSpeak) and retrieve data from remote servers.

**Key Features**

1. Used for internet-based communication.
2. Uses GET and POST requests to exchange data.
3. Works over TCP/IP networks.
4. Supports IoT cloud integration.

**Usage in Our Project**

1. NodeMCU ESP8266 sends sensor data to ThingSpeak using HTTP POST requests.
2. Allows remote monitoring of farm conditions via a web dashboard.
3. Ensures real-time data visualization and analytics for better decision-making.

**Future Scope:**

1. The Smart Indoor Farming System has immense potential for future advancements, making it a scalable, efficient, and sustainable solution for modern agriculture. With further development, the system can be enhanced in the following ways:
2. Automation of Farming Processes – The system can be upgraded to include automated irrigation, lighting, and ventilation based on real-time sensor readings, reducing manual intervention.
3. Energy and Resource Optimization – By integrating AI-based predictive analytics, the system can optimize water and electricity usage, ensuring efficient resource management.
4. Advanced Data Analytics and AI Integration – Machine learning algorithms can be implemented to analyze historical sensor data, predict environmental conditions, and suggest optimal farming strategies.
5. Integration with Mobile Applications – A dedicated mobile app can be developed to allow users to monitor and control farm conditions in real-time from their smartphones.
6. Multi-Cloud Support – Expanding compatibility with other cloud platforms like AWS IoT, Google Firebase, or Microsoft Azure for more robust data storage and analytics.
7. Wireless Sensor Networks (WSN) – Using multiple wireless sensors in large-scale indoor farms can enhance data accuracy, reliability, and range of monitoring.

**Conclusion:**

The Smart Indoor Farming System effectively utilizes IoT and embedded systems to enable real-time monitoring of key environmental parameters such as temperature, humidity, soil moisture, light intensity, and water levels. By integrating STM32F407, NodeMCU ESP8266, and multiple sensors, the system ensures accurate data collection and seamless communication between hardware components.

Using the UART protocol, data is efficiently transmitted from STM32 to NodeMCU, while the HTTP protocol facilitates wireless data transfer to the ThingSpeak cloud. This allows users to remotely access and analyze farm conditions through a web-based dashboard, making it easier to track changes and respond accordingly. The system provides a scalable and flexible architecture, allowing for the addition of more sensors or automation modules to further enhance functionality.

By leveraging IoT technology and cloud-based monitoring, this project offers a modern, data-driven approach to precision farming. It empowers farmers with real-time insights, helping them optimize environmental conditions for better crop growth and improved productivity. The ability to remotely monitor farm conditions from anywhere makes this system a valuable tool for sustainable and efficient indoor farming.

# 