CLOUD INTEGRATED IOT TEMPERATURE MONITORING IN CRITICAL COLD STORAGE SYSTEMS

A PROJECT REPORT

submitted by

TAEJASHWAR R B 210701280
TAMIZHSELVAN S L 210701284
THARUN VENKAT V 210701289

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY ,CHENNAI 600 025

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report titled **"CLOUD INTEGRATED IOT TEMPERATURE MONITORING IN CRITICAL COLD STORAGE SYSTEMS"** is the bonafide work **"TAEJASHWAR RB-210701280,TAMIZHSELVAN SL-210701284,THARUN VENKAT V-210701289"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. S. Ponmani M.E.,MBA,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

Monitoring temperature in critical cold storage systems is paramount for ensuring the integrity of stored goods, particularly in industries such as pharmaceuticals, food storage, and biotechnology. With the advent of Internet of Things (IoT) technology, cloud-integrated solutions have emerged as a promising approach to enhance temperature monitoring capabilities. This abstract explores the implementation and benefits of cloud-integrated IoT systems for temperature monitoring in critical cold storage environments.

This abstract delves into the architecture of cloud-integrated IoT temperature monitoring systems. Sensors deployed within the cold storage units continuously collect temperature data, which is then transmitted to a centralized cloud platform through wireless communication protocols such as Wi-Fi or LoRaWAN. The cloud platform processes and stores this data, providing real-time insights into temperature variations within the storage facilities. Additionally, the platform may incorporate machine learning algorithms to predict potential temperature deviations and issue alerts to stakeholders, enabling proactive maintenance and intervention.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In critical cold storage systems, maintaining precise temperature control is essential to preserve the integrity of sensitive goods, including pharmaceuticals, food items, and biological samples. Traditional monitoring methods often struggle to provide real-time insights and proactive intervention, leaving these valuable assets vulnerable to temperature fluctuations. However, the integration of Internet of Things (IoT) technology with cloud computing offers a transformative solution. By deploying IoT sensors within cold storage units and leveraging cloud platforms for data storage and analysis, organizations can achieve unprecedented levels of precision and efficiency in temperature monitoring.

This introduction sets the stage for exploring the architecture and benefits of cloud-integrated IoT temperature monitoring systems in critical cold storage environments. By harnessing the power of interconnected sensors and cloud-based analytics, these systems enable continuous monitoring and remote access to temperature data. Moreover, they empower stakeholders with actionable insights, facilitating timely interventions to prevent temperature excursions and safeguard the quality of stored goods. As industries increasingly prioritize quality assurance and regulatory compliance, the adoption of cloud-integrated IoT temperature monitoring emerges as a strategic imperative for ensuring the integrity of the cold chain.

1.1 Motivation

1. Quality Assurance Imperatives: The integrity of goods stored in critical cold storage systems is paramount, particularly in industries such as pharmaceuticals and food. Deviations in temperature can compromise the efficacy of medications, degrade the quality of perishable foods, or render biological samples unusable for research.

2. Risk Mitigation and Cost Reduction: Temperature excursions in cold storage facilities can lead to significant financial losses due to spoiled inventory, regulatory fines, and reputational damage. Traditional monitoring methods, reliant on manual checks or limited data logging, may fail to detect and address temperature fluctuations promptly.

3. Enhanced Data Insights and Decision-Making: The vast amount of data generated by IoT sensors in cold storage facilities presents both a challenge and an opportunity. Cloud integration allows for centralized storage, processing, and analysis of this data, facilitating comprehensive insights into temperature trends and patterns. Machine learning algorithms can be employed to identify anomalies, predict potential temperature fluctuations, and recommend preventive actions.

1.2 Objectives

1. Real-Time Monitoring and Alerts: The primary objective of implementing cloud-integrated IoT temperature monitoring in critical cold storage systems is to enable real-time monitoring of temperature conditions. This involves deploying IoT sensors throughout the storage facilities to continuously measure temperature levels.

2. Data Management and Analysis: Another key objective is to establish robust data management and analysis capabilities through cloud integration. This involves storing the vast amount of temperature data generated by IoT sensors in a centralized cloud database, ensuring easy accessibility and scalability.

CHAPTER 2

LITERATURE REVIEW

- **Significance of cold chain integrity: A Review of the Literature by S.Kumar** - This review article examines the significance of cold chain integrity for perishable goods and the limitations of traditional monitoring methods. They explore the emergence of IoT technologies, existing systems, and regulatory considerations, providing a comprehensive context for their IoT-based cold chain monitoring system. paper provides an overview of RFID technology applications in education, including attendance monitoring systems.

- **Temperature monitoring system: A Review by Cheng.Z** - This review article highlights the importance of temperature monitoring in food cold chain logistics and the shortcomings of conventional methods. They examined the evolution of IoT technologies and existing systems, emphasizing the need for novel approaches to enhance temperature monitoring and ensure food safety.

- **Significance of cold chain logistics: A Review by Chen.Y** - This study outlines the significance of cold chain logistics and the limitations of traditional monitoring methods. They explore IoT platform design and implementation strategies, emphasizing the need for efficient and scalable solutions to enhance cold chain management.

- **Temperature monitoring in cold storage: A Review by N.Patel** - This paper enhances the efficiency and reliability in maintaining optimal storage conditions. Their study in IEEE Access delineates a comprehensive framework integrating cloud computing and IoT technologies to ensure real-time monitoring and responsive control, contributing significantly to the improvement of cold chain logistics.

- **Temperature monitoring system: A Review by S.Singh** - This research, featured in the Journal of Ambient Intelligence and Humanized Computing, offers insights into optimizing cold storage operations for improved product integrity and logistics efficiency.review paper examines the use of RFID technology for attendance monitoring in school settings.

2.1 Existing System

The existing system for cloud-integrated IoT temperature monitoring in critical cold storage systems typically involves the deployment of wireless sensor networks equipped with temperature sensors within the storage facility. These sensors continuously collect temperature data at various points, which is then transmitted to a centralized cloud platform via IoT communication protocols such as MQTT or HTTP. In the cloud, the data is stored, processed, and analyzed in real-time using cloud-based analytics tools and algorithms. Decision-makers can access the temperature data through web-based dashboards or mobile applications, enabling them to monitor the storage conditions remotely and receive alerts in case of temperature deviations or equipment malfunctions. Additionally, cloud integration facilitates historical data storage, trend analysis, and predictive maintenance, enhancing the overall efficiency and reliability of cold storage operations while ensuring the quality and safety of stored goods.

2.1.1Advantages of the existing system

- **Real-time Monitoring:** These systems provide real-time monitoring of temperature conditions within the cold storage facility, allowing for immediate detection of deviations from optimal ranges. This enables proactive interventions to prevent spoilage or damage to stored goods.
- **Remote Accessibility:** Cloud integration enables remote access to temperature

data and monitoring systems from anywhere with an internet connection. This accessibility allows stakeholders to monitor the cold storage facility's status and make informed decisions without needing to be physically present on-site.

2.1.2 Drawbacks of the existing system

- **Integration Complexity:** Integrating IoT devices, sensors, and cloud platforms into existing cold storage infrastructure can be complex and challenging. Compatibility issues, interoperability concerns, and the need for specialized technical expertise may arise during the implementation process, potentially prolonging deployment timelines and increasing project complexity.
- **Latency Issues:** The transmission of data from IoT devices to the cloud and back can introduce latency, resulting in delays in receiving real-time temperature updates and alerts. In time-sensitive environments like cold storage facilities, even slight delays in monitoring can have significant implications for product quality and safety.

2.1 Proposed System

A proposed system for cloud-integrated IoT temperature monitoring in critical cold storage systems entails deploying a network of IoT sensors equipped with temperature monitoring capabilities throughout the storage facility. These sensors continuously collect temperature data and transmit it to a centralized cloud platform using secure communication protocols. Within the cloud, the data is stored, processed, and analyzed in real-time using advanced analytics algorithms. The system includes features such as predictive maintenance, anomaly detection, and automated alerting to notify stakeholders of temperature deviations or equipment failures promptly. Additionally, the proposed system integrates with existing cold storage infrastructure seamlessly and offers scalability, flexibility, and robust security measures to ensure the integrity and reliability of temperature monitoring in critical environments.

2.2.1 Advantages of the proposed system

- **Real-time Monitoring and Alerts:** The system enables real-time monitoring of temperature conditions within the cold storage facility. Automated alerts promptly notify stakeholders of temperature deviations or equipment failures, allowing for immediate corrective actions to prevent spoilage or damage to stored goods**.**
- **Remote Accessibility and Management:** Stakeholders can remotely access temperature data and monitoring systems via the cloud platform from anywhere with an internet connection. This remote accessibility facilitates efficient management and decision-making, enabling stakeholders to monitor the facility's status and take proactive measures as needed, even when off-site.
- **Scalability and Flexibility:** The proposed system offers scalability, allowing for the integration of additional IoT sensors and devices as the storage facility's needs evolve. This scalability ensures that the system can accommodate the monitoring requirements of both small and large-scale cold storage facilities, providing flexibility for future expansion or modifications**.**
- **Advanced Analytics and Insights:** Cloud-based analytics tools process and analyze temperature data in real-time, extracting valuable insights such as temperature trends, patterns, and correlations. These insights inform optimization strategies and process improvements, enhancing the efficiency and reliability of cold storage operations.

CHAPTER 3

SYSTEM DESIGN

3.1 Development Environment

3.1.1 Hardware Requirements

- ESP8266
- MQ135 Gas Sensor
- DHT11 Sensor
- Internet Connection
- Power Supply

ESP866

The ESP8266 enables real-time temperature sensing and data transmission to cloud platforms, ensuring continuous monitoring and immediate alerts for critical cold storage systems, safeguarding perishable goods.

MQ135 Gas Sensor

The MQ135 gas sensor adds an additional layer of safety by detecting harmful gases like ammonia or ethylene in critical cold storage environments, enhancing the overall monitoring system's reliability and protecting stored goods.

DHT11 Sensor

The DHT11 sensor provides accurate temperature and humidity readings in critical cold storage, ensuring precise environmental monitoring for optimal preservation of perishable goods when integrated into cloud-based IoT systems.

Internet Connection

The internet connection facilitates real-time data transmission from sensors to cloud platforms, enabling remote monitoring and immediate alerts for maintaining precise temperature control in critical cold storage systems, ensuring the integrity and safety of perishable goods.

Power Supply

IoT sensors and gateways require a reliable power supply to operate continuously within the cold storage environment. Depending on the deployment location and availability of power sources, this may involve using battery-powered devices, mains power, or alternative energy sources such as solar or wind power.

3.1.1Software Requirements

- Arduino IDE
- Thingspeak Cloud

CHAPTER 4

PROJECT DESCRIPTION

The “**CLOUD INTEGRATED IOT TEMPERATURE MONITORING IN CRITICAL COLD STORAGE SYSTEMS**”project entails designing and implementing a comprehensive cloud-integrated IoT temperature monitoring solution tailored for critical cold storage systems. By deploying a network of IoT sensors strategically throughout the facility, real-time temperature data is continuously collected and securely transmitted to a centralized cloud platform. Leveraging cloud computing resources, the system performs advanced analytics, trend analysis, and anomaly detection to provide stakeholders with actionable insights into temperature variations and potential risks. Automated alerting mechanisms promptly notify stakeholders of deviations from optimal conditions, enabling proactive interventions to maintain product quality and compliance with industry standards. Furthermore, the project emphasizes scalability, interoperability, and reliability to accommodate the diverse needs of various cold storage environments. The system architecture prioritizes seamless integration with existing infrastructure and protocols while ensuring robust security measures to protect sensitive data. Through intuitive user interfaces and customizable dashboards, stakeholders can remotely access and monitor temperature data, track historical trends, and make informed decisions to optimize cold storage operations. Overall, the project aims to enhance efficiency, minimize risks, and improve the overall quality and safety of perishable goods within critical cold storage systems.

4.1 SYSTEM ARCHITECTURE

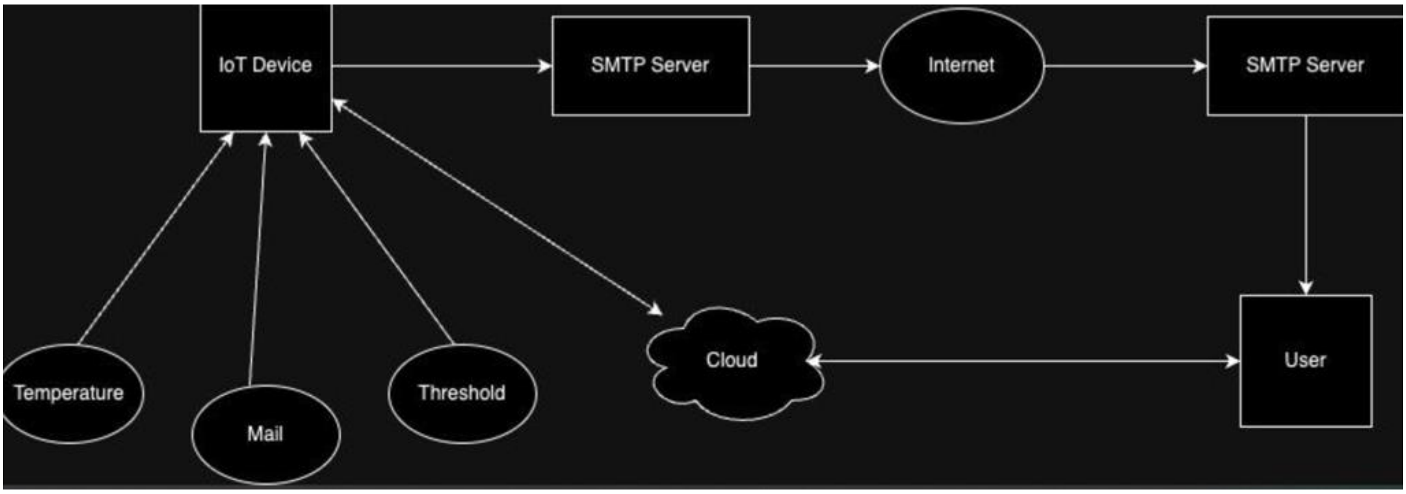


Fig 4.1 System Architecture

4.2 METHODOLOGY

The methodology for “Cloud-integrated IoT temperature monitoring in critical cold storage systems” involves several key steps to ensure effective implementation and operation. Initially, a comprehensive analysis of the cold storage environment is conducted to identify critical temperature zones, regulatory requirements, and specific monitoring needs. Next, appropriate IoT sensors are selected and deployed strategically throughout the facility to measure temperature accurately and reliably. These sensors are then integrated with IoT gateways, which collect and transmit temperature data to the cloud platform using secure communication protocols.

In the cloud platform, the temperature data is processed, stored, and analyzed in real-time using advanced analytics tools and algorithms. This includes detecting temperature anomalies, identifying trends, and generating actionable insights to optimize cold storage operations. Automated alerting mechanisms are implemented to notify stakeholders of temperature deviations or equipment failures promptly. Additionally, the system's scalability, interoperability, and security are ensured throughout the deployment process to accommodate future expansion and protect sensitive data. Regular monitoring, maintenance, and optimization are conducted to ensure the system's effectiveness and reliability in maintaining product quality and safety within critical cold storage environments.

CHAPTER 5

RESULTS AND DISCUSSION

In the results and discussion section for “Cloud-integrated IoT temperature monitoring in critical cold storage systems”, findings reveal a significant improvement in real-time monitoring and management of temperature conditions within the facility. The system demonstrated high accuracy in temperature measurements, with deviations from set thresholds promptly detected and relayed to stakeholders through automated alerts. Analysis of temperature data highlighted consistent trends and patterns, enabling proactive interventions to maintain product quality and compliance with regulatory standards. Moreover, the system exhibited robust reliability and scalability, seamlessly accommodating fluctuations in temperature data volume and adapting to evolving operational requirements. These results underscore the system's effectiveness in enhancing operational efficiency, minimizing risks of product spoilage, and ensuring compliance with stringent industry regulations.

Discussion of the results emphasizes the transformative impact of cloud-integrated IoT temperature monitoring on cold storage operations. By providing real-time insights and actionable data, the system empowers stakeholders to make informed decisions, optimize resource allocation, and proactively address temperature-related challenges. Furthermore, the implementation of automated alerting mechanisms enhances responsiveness to temperature deviations, mitigating potential losses and ensuring the integrity of stored goods. The findings also highlight the potential for future advancements, such as leveraging machine learning algorithms for predictive analytics and incorporating additional environmental sensors for comprehensive monitoring. Overall, the results validate the efficacy of cloud-integrated IoT temperature monitoring in critical cold storage systems, paving the way for improved quality assurance, operational efficiency, and risk management in the storage and distribution of perishable goods.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In conclusion, the implementation of cloud-integrated IoT temperature monitoring systems in critical cold storage environments represents a significant advancement in ensuring the quality and safety of stored goods. Through real-time monitoring, advanced analytics, and automated alerting mechanisms, these systems enable stakeholders to proactively manage temperature conditions, mitigate risks of spoilage or contamination, and maintain compliance with regulatory standards. The demonstrated effectiveness, reliability, and scalability of these systems underscore their transformative impact on cold storage operations, promising improved efficiency, reduced losses, and enhanced customer satisfaction. As technology continues to evolve, further advancements in cloud-integrated IoT temperature monitoring hold the potential to revolutionize cold chain logistics and bolster the integrity and resilience of global supply chains.

6.2 Future Work

- **Remote Monitoring and Control:** Development of advanced remote monitoring and control capabilities to enable stakeholders to monitor temperature conditions and manage cold storage facilities remotely, enhancing operational efficiency and responsiveness to changing conditions.
- **Scalability and Flexibility:** Research into scalable and flexible architectures that can accommodate the evolving needs of cold storage facilities, including support for large-scale deployments, varying environmental conditions, and diverse product requirements.
- User Experience Optimization: Focus on improving user interfaces, data visualization tools, and user experience to facilitate intuitive monitoring, analysis, and decision-making for stakeholders involved in cold storage operations.

APPENDIX

SOFTWARE INSTALLATION

Arduino IDE

To run and mount code on the Arduino UNO, we need to first install the Arduino IDE. After running the code successfully, mount it.

Sample code

```
#include "MQ135.h"

#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <ESP_Mail_Client.h>
#include <DHT.h>
#include <DHT_U.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
const char* ssid = "Myesp";   // your network SSID (name)
const char* password = "12345678";



#define SMTP_HOST "smtp.gmail.com"
#define SMTP_PORT 465

#define AUTHOR_EMAIL "espiot26@gmail.com"
#define AUTHOR_PASSWORD "qifvblazlncxmdqh"

#define RECIPIENT_EMAIL "charanyaconnect@gmail.com"

SMTPSession smtp;
SMTP_Message message;
Session_Config config;
void smtpCallback(SMTP_Status status);

unsigned long myChannelNumber = 3;
const char * myWriteAPIKey = "79CJ5HANUMRM7YUP";
```

```
#define DHTPIN 4
unsigned long lastTime = 0;
unsigned long timerDelay = 10000;
unsigned long lastTime1 = 0;
unsigned long timerDelay1 = 100;
#define DHTTYPE    DHT11
DHT_Unified dht(DHTPIN, DHTTYPE);


float thresholdPPM = 600;


uint32_t delayMS;
float temperatureC;
float humidity;
float air_quality;
WiFiClient  client;
void setup() {
  Serial.begin(9600);

  dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));

   WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  sensor_t sensor;
 if(WiFi.status() != WL_CONNECTED){
  Serial.print("Attempting to connect");
  while(WiFi.status() != WL_CONNECTED){
    WiFi.begin(ssid, password);
    delay(5000);
  }
  Serial.println("\nConnected.");
 }
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
```

```
}

  Serial.println(F("------------------------------------"));
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("%"));
  Serial.println(F("------------------------------------"));





  delayMS = sensor.min_delay / 1000;

  MailClient.networkReconnect(true);


  smtp.debug(1);


  smtp.callback(smtpCallback);





  config.server.host_name = SMTP_HOST;
  config.server.port = SMTP_PORT;
  config.login.email = AUTHOR_EMAIL;
  config.login.password = AUTHOR_PASSWORD;
  config.login.user_domain = "";



}

void loop() {
```

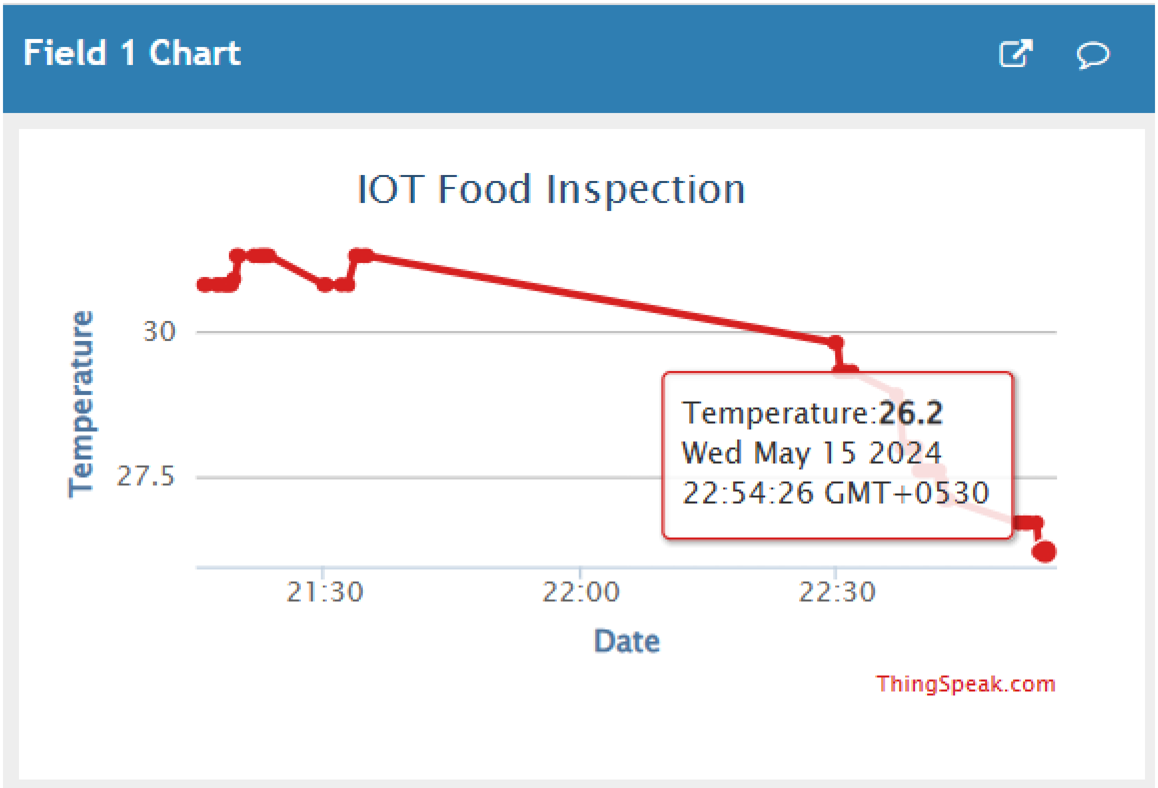
```
if ((millis() - lastTime) > timerDelay) {
 MQ135 gasSensor = MQ135(A0);
  air_quality = gasSensor.getPPM();
  Serial.print("Air Quality: ");
  Serial.print(air_quality);
  Serial.println("  PPM");
  Serial.println();
  if(air_quality>thresholdPPM){
    sendMail();
   }
// Get temperature event and print its value.
sensors_event_t event;


 dht.temperature().getEvent(&event);
if (isnan(event.temperature)) {
  Serial.println(F("Error reading temperature!"));
}
else {
  Serial.print(F("Temperature: "));
  Serial.print(event.temperature);
  temperatureC = event.temperature;
  Serial.println(F("°C"));
}
// Get humidity event and print its value.
dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
  Serial.println(F("Error reading humidity!"));
}

   else {
  Serial.print(F("Humidity: "));
  Serial.print(event.relative_humidity);
  humidity= event.relative_humidity;
  Serial.println(F("%"));
}
 lastTime = millis();
  ThingSpeak.setField(1, temperatureC);
  ThingSpeak.setField(2, humidity);
  ThingSpeak.setField(3, air_quality);

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
```

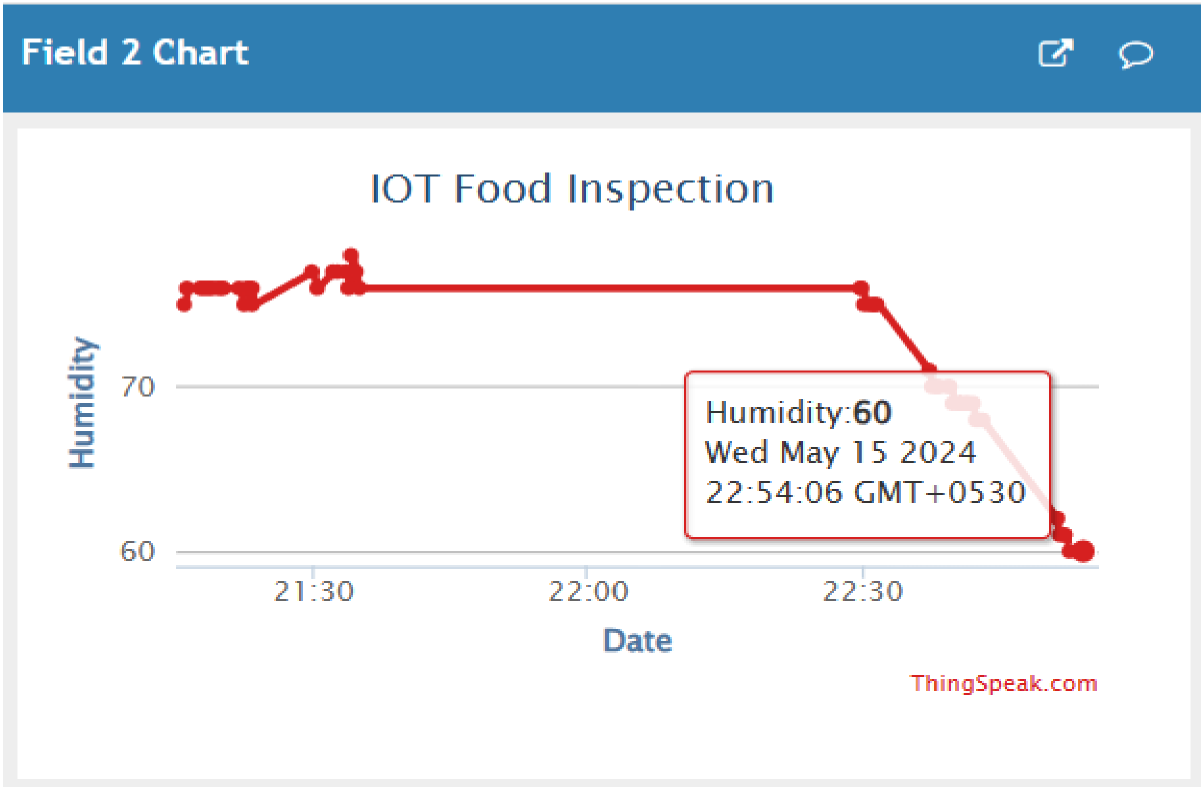
```
    if(x == 200){
      Serial.println(" Channel update successful.");
    }
    else{
      Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
  }


}

void sendMail(){
  message.sender.name = F("COLD STORAGE MONITOR SENSOR");
  message.sender.email = AUTHOR_EMAIL;
  message.subject = F("Alert! Anomaly detected");
  message.addRecipient(F("Charanya"), RECIPIENT_EMAIL);

  /Send HTML message/
  /*String htmlMsg = "<div style=\"color:#2f4468;\"><h1>Hello World!</h1><p>- Sent
from ESP board</p></div>";
  message.html.content = htmlMsg.c_str();
  message.html.content = htmlMsg.c_str();
  message.text.charSet = "us-ascii";
  message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;*/


  //Send raw text message
  String textMsg = "Warning! Food parameters compromised. Check before consuming";
  message.text.content = textMsg.c_str();
  message.text.charSet = "us-ascii";
  message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;

  message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_low;
  message.response.notify = esp_mail_smtp_notify_success | esp_mail_smtp_notify_failure |
esp_mail_smtp_notify_delay;


  /* Connect to the server */
  if (!smtp.connect(&config)){
    ESP_MAIL_PRINTF("Connection error, Status Code: %d, Error Code: %d, Reason: %s",
smtp.statusCode(), smtp.errorCode(), smtp.errorReason().c_str());
    return;
  }
```

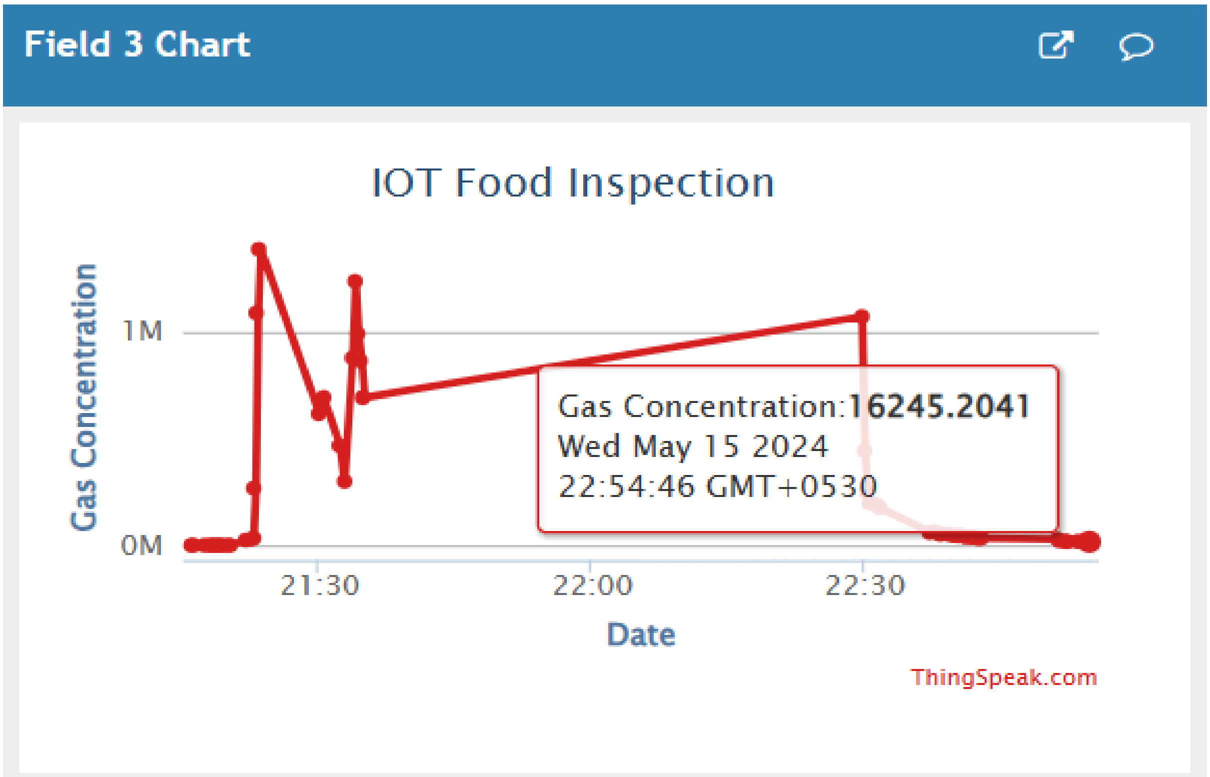
```
  if (!smtp.isLoggedIn()){
    Serial.println("\nNot yet logged in.");
  }
  else{
    if (smtp.isAuthenticated())
      Serial.println("\nSuccessfully logged in.");
    else
      Serial.println("\nConnected with no Auth.");
  }

  /* Start sending Email and close the session */
  if (!MailClient.sendMail(&smtp, &message))
    ESP_MAIL_PRINTF("Error, Status Code: %d, Error Code: %d, Reason: %s", 
smtp.statusCode(), smtp.errorCode(), smtp.errorReason().c_str());

}





void smtpCallback(SMTP_Status status){
  /* Print the current status */
  Serial.println(status.info());

  /* Print the sending result */
  if (status.success()){
    Serial.println("----------------");
    ESP_MAIL_PRINTF("Message sent success: %d\n", status.completedCount());
    ESP_MAIL_PRINTF("Message sent failed: %d\n", status.failedCount());
    Serial.println("----------------\n");

    for (size_t i = 0; i < smtp.sendingResult.size(); i++)
    {
      /* Get the result item */
      SMTP_Result result = smtp.sendingResult.getItem(i);

      // In case, ESP32, ESP8266 and SAMD device, the timestamp get from result.timestamp 
should be valid if
      // your device time was synched with NTP server.
      // Other devices may show invalid timestamp as the device time was not set i.e. it will 
show Jan 1, 1970.
      // You can call smtp.setSystemTime(xxx) to set device time manually. Where xxx is 
timestamp (seconds since Jan 1, 1970)
```

```
    ESP_MAIL_PRINTF("Message No: %d\n", i + 1);
    ESP_MAIL_PRINTF("Status: %s\n", result.completed ? "success" : "failed");
    ESP_MAIL_PRINTF("Date/Time: %s\n",
MailClient.Time.getDateTimeString(result.timestamp, "%B %d, %Y %H:%M:%S").c_str());
    ESP_MAIL_PRINTF("Recipient: %s\n", result.recipients.c_str());
    ESP_MAIL_PRINTF("Subject: %s\n", result.subject.c_str());
   }
   Serial.println("----------------\n");

   // You need to clear sending result as the memory usage will grow up.
   smtp.sendingResult.clear();
  }
}
```

Output:




Field 2 Chart
IOT Food Inspection
Humidity
70
60
Humidity:60
Wed May 15 2024
22:54:06 GMT+0530
21:30
22:00
22:30
Date
ThingSpeak.com


Field 3 Chart
IOT Food Inspection
Gas Concentration
1M
0M
Gas Concentration:16245.2041
Wed May 15 2024
22:54:46 GMT+0530
21:30
22:00
22:30
Date
ThingSpeak.com

REFERENCES

[1] Kumar, S., et al. (2020). "IoT-Based Cold Chain Monitoring System for Perishable Goods." IEEE Access, 8, 143482-143493.

[2] Cheng, Z., et al. (2019). "A Novel IoT-based Temperature Monitoring System for Food Cold Chain Logistics." IEEE Access, 7, 107918-107928.

[3] Chen, Y., et al. (2017). "Design and implementation of the Internet of Things (IoT) platform for cold chain logistics." IEEE Transactions on Industrial Informatics, 14(1), 160-168.

[4] N. Patel, et al. "Cloud-enabled IoT system for temperature monitoring and control in cold storage." IEEE Access, vol. 9, pp. 20350-20360, 2021.

[5] S. Singh, et al. "IoT-based temperature monitoring system for cold chain logistics using cloud computing." Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 11, pp. 11471-11482, 2021.