COLD STORAGE MONITORING

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING (Internet of Things)

by

I. Venkata Hari Paniindra 2111CS050106

P. Naveen 2111CS050090

S. Manideep 2111CS050117

Anand Charan 2111CS050072

Under the esteemed guidance of

Mrs. G. Sailaja

Assistant Professor

Department of Computer Science & Engineering (IOT)

School of Engineering

MALLA REDDY UNIVERSITY

Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2024

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

Department of Computer Science and Engineering (Internet of Things)

CERTIFICATE

This is to certify that the project report entitled **“COLD STORAGE MONITORING”,** submitted by **I Venkata Hari Paniindra (2111CS050106), P Naveen (2111CS050090), S Manideep (2111CS050117),A Anand Charan (2111CS050072)** towards the partial fulfilment for the award of bachelor’s degree in Computer Science and Engineering from the department of Internet of Things, Mall Reddy University, Hyderabad, is a record of bonafide work done by him/ her. The results embodied in the work are not submitted to any other university or institute for award of any degree or diploma.

Internal Guide	**Head of the department**
Mrs. G . Sailaja	**Dr. G. Anand Kumar**
Assistant Professor	**CSE(IOT)**

External Examiner

DECLARATION

We hereby declare that the project report entitled "**COLD STORAGE MONITORING**"has been carried out by us and this work has been submitted to the **Department of Computer Science and Engineering (Internet of Things), Malla Reddy University**, Hyderabad in partialfulfillment of the requirements for the award of degree of Bachelor of Technology. We furtherdeclare that this project work has not been submitted in full or part for the award of any otherdegree in any other educational institutions.

Place:

Date:

I VENKATA HARI PANIINDRA	2111CS050106
P NAVEEN	2111CS050090
S MANIDEEP	2111CS050117
ANAND CHARAN	2111CS050072

ACKNOWLEDGEMENT

We extend our sincere gratitude to all those who have contributed to the completion of this project report. Firstly, we would like to extend our gratitude to **Dr. V. S. K Reddy**, Vice Chancellor, for his visionary leadership and unwavering commitment to academic excellence.

We would also like to express my deepest appreciation to our project guide and our class PRC project Coordinator **Mrs. G. Sailaja, Assistant Professor,** whose invaluable guidance, insightful feedback, and unwavering support have been instrumental throughout the course of this project for successfuloutcomes.

We extend our gratitude to our **PRC-convenor, Dr. G. Latha,** for giving valuable inputs andtimely quality of our project through a critical review process.

We are also grateful to **Dr. G. Anand Kumar, Head of the Department of Internet of Things**, for providing with the necessary resources and facilities to carry out this project.

We would like to thank **Dr. Kasa Ravindra, Dean, School of Engineering**, for his encouragement and support throughout my academic pursuit.

My heartfelt thanks also go to **Dr. Harikrishna Kamatham, Associate Dean School of Engineering** for his guidance and encouragement. We are deeply indebted to all of them for their support, encouragement, and guidance, without which this project would not have been possible.

I VENKATA HARI PANIINDRA	2111CS050106
P NAVEEN	2111CS050090
S MANIDEEP	2111CS050117
A ANAND CHARAN	2111CS050072

ABSTRACT

IoT technology has enabled the development of useful application to improve the quality of life. We can also the need to develop other application to make an effective use of the devices and the internet connectivity. Our project try's to solve the problems occurring due to lack of monitoring of our goods. In this regard we have taken the first to solve the problem by proposing a project which can Monitor the temperature of goods present in an container and notify the person in-charge according to the constraints he/she have provided . For this project we will be using ESP8266 as the processing unit due to wireless wi-fi connectivity and upload HTML code to monitor temperature and daily usage of our model through the local network connected to the device.

INDEX

Contents	Page No.

CHAPTER-1
INTRODUCTION

1.1 Problem Definition & Description

Despite advancements in cold storage technology, maintaining consistent temperature and humidity levels within optimal ranges remains a challenge in many facilities. This inconsistency poses a significant risk to the quality and safety of perishable goods, leading to potential spoilage, degradation, and financial losses. Thus, there is a pressing need for an efficient and reliable cold storage monitoring system that can accurately monitor environmental conditions, promptly detect deviations from the desired parameters, and alert personnel to take timely corrective actions, ensuring the integrity and longevity of stored products

The purpose of a cold storage monitoring system is to ensure the proper storage conditions for perishable goods, such as food, pharmaceuticals, and biological samples, that require specific temperature and humidity levels to maintain quality and safety. These systems typically involve sensors placed within the storage facility to continuously monitor environmental conditions. If the temperature or humidity deviates from the specified range, the system can alert personnel, allowing them to take corrective action to prevent spoilage, degradation, or loss of valuable products. Additionally, these systems often provide data logging capabilities for regulatory compliance, quality control, and process optimization. Overall, cold storage monitoring systems play a critical role in preserving the integrity of sensitive goods throughout their storage lifecycle.

Problem Statement

Despite advancements in cold storage technology, maintaining consistent temperature and humidity levels within optimal ranges remains a challenge in many facilities. This inconsistency poses a significant risk to the quality and safety of perishable goods, leading to potential spoilage, degradation, and financial losses. Thus, there is a pressing need for an efficient and reliable cold storage monitoring system that can accurately monitor environmental conditions, promptly detect deviations from the desired parameters, and alert personnel to take timely corrective actions, ensuring the integrity and longevity of stored products.

1.2 Objective of the Project

The objectives of cold storage monitoring system include:

Temperature Monitoring: Ensuring that the temperature within the cold storage remains within specified ranges to preserve the quality and safety of the stored goods, such as food, pharmaceuticals, or other temperature-sensitive items.

Real-time Alerts: Providing real-time alerts in case of temperature fluctuations or deviations from the optimal range, allowing prompt action to be taken to prevent spoilage or damage to the stored goods.

Data Logging: Logging temperature data over time to track temperature trends, identify potential issues, and comply with regulatory requirements.

Remote Monitoring: Enabling remote monitoring of the cold storage facility, allowing stakeholders to monitor temperature conditions from anywhere, at any time, via web or mobile interfaces.

Energy Efficiency: Optimizing energy usage by monitoring temperature conditions and adjusting cooling systems as needed to maintain optimal conditions while minimizing energy consumption.

Compliance and Reporting: Facilitating compliance with regulatory standards and requirements by providing accurate temperature records and reports.

Cost Reduction: Minimizing losses due to spoilage or damage by ensuring optimal storage conditions, thereby reducing operational costs and maximizing efficiency.

Predictive Maintenance: Utilizing collected data to implement predictive maintenance strategies, helping to identify potential equipment issues before they lead to costly failures or downtime.

1.3 Scope of the project

The main aim of the project is to be aware of damage being caused to products due to temperature and change and also the affects of monitoring data can be useful to create predictive models which can send data and make changes on their own without human intervention.

CHAPTER - 2
SYSTEM ANALYSIS

2.1 Existing System

2.1.1 Background & Literature Survey

Cold storage systems are vital infrastructures in various industries, including food and pharmaceuticals, where perishable goods need to be preserved at low temperatures to maintain quality and safety. The design, operation, and optimization of cold storage systems are essential to ensure efficient cooling, energy conservation, and product integrity. This review provides an overview of the background and existing literature on cold storage systems, focusing on key aspects such as design principles, energy efficiency, monitoring technologies, and recent advancements.

2.1.2 Limitations of Existing System

The limitation of cold storage devices is their inability to be notified when required or having an immediate response due to delay in transfer of message this delay can cause lot of damage in the longer run due to being ignored this have an effect in long term where the unawareness leads to huge problems and cannot be optimized in a useful manner.

2.2 Proposed System

The proposal is to use Oled and SMTP protocols to send information when required to have a connection with the internet services to send data to nearby networks to ensure successful elimination of spoilage or harm due to ignorance of the change in temperature of storage container

The SMTP plays an important role in sending the information through g-mail and is responsible to transfer information up to files like image to ensure more credibility of the information being transferred the information is used to transfer the way we can react to change of temperature which can be harmful and can be avoided with the help of other innovations from IoT.

2.3 Software & Hardware Requirements

2.3.1 Software Requirements

- Arduino IDE
- Installed Libraries(SMTP, Adafruits,etc)

2.3.2 Hardware Requirements

- ESP8266
- DHT11
- OLED
- BUZZER
- LED’s
- Jumper wire

2.4 Feasibility Study

2.4.1 Technical Feasibility

Cold storage monitoring systems are technically feasible due to advancements in sensor technology, data communication, and data analytics. This section outlines the technical aspects that contribute to the feasibility of implementing such systems. The feasibility study of the Cold storage monitoring system derives the nature of the project and to the extent of information.

2.4.2 Robustness & Reliability

Robustness and reliability are essential characteristics of cold storage monitoring systems to ensure continuous and accurate monitoring of environmental conditions within storage facilities. This section explores how these systems achieve robustness and reliability. Hardware Redundancy is to mitigate the risk of component failures. This includes redundant sensors, communication modules, and power supplies. If a sensor or communication link fails, redundant components can take over, ensuring uninterrupted monitoring. Redundancy enhances the robustness of the system and reduces the likelihood of data loss or system downtime. To maintain reliability and accuracy, cold storage monitoring systems require regular maintenance and calibration. Sensors should be calibrated periodically to ensure accuracy and consistency of measurements. Routine inspections and preventive maintenance help identify and address potential issues before they escalate into major problems. By keeping the system well-maintained, operators can ensure its long-term reliability and performance.

2.4.3 Economic Feasibility

Cold storage monitoring systems offer various economic benefits by optimizing energy usage, reducing product losses, enhancing operational efficiency, and minimizing maintenance costs. This section examines the economic feasibility of implementing such systems:

Cost Savings through Energy Efficiency:

One of the primary economic benefits of cold storage monitoring systems is the potential for energy savings. By accurately monitoring and controlling temperature and humidity levels, these systems optimize the operation of refrigeration equipment, reducing energy consumption. The cost savings from improved energy efficiency can be substantial, particularly in large-scale cold storage facilities where refrigeration accounts for a significant portion of operating expenses.

Reduction in Product Losses:

Cold storage monitoring systems help prevent product losses by maintaining optimal storage conditions and detecting deviations that could lead to spoilage or degradation of perishable goods. By minimizing product losses due to temperature fluctuations, improper humidity levels, or equipment failures, these systems help businesses protect their investments and maximize profits. The economic value of reduced product losses can be significant, especially for high-value or sensitive products.

CHAPTER – 3
ARCHITECTURAL DESIGN

3.1 Module Design

3.1.1 ESP8266

ESP8266 is a microcontroller board which is capable of with holding a capacity 31% from total size of the ram.

The esp8266 contains an inbuilt Wi-Fi module so that it can host web page and connect with SMTP protocols.

3.1.2 OLED Display

An OLED display size starts from around 0.96 inches it is used to display the contents of the matter and is always used in project where serial monitor is not visible and a larger display cannot be fitted The OLED makes use of TX and RX pins to connect with the ESP8266 to send data and receive information of its position

3.1.3 DHT11 Sensor

The DHT sensor is capable of measuring temperature from -20 degrees to 50 degress and is commonly used in projects with less temperature variance the usage of DHT is also done to find out the humidity the value is always between 0-100

The data transfer is done once every five seconds or so to update the values in the controller to enable the usage of the data in effective manner.

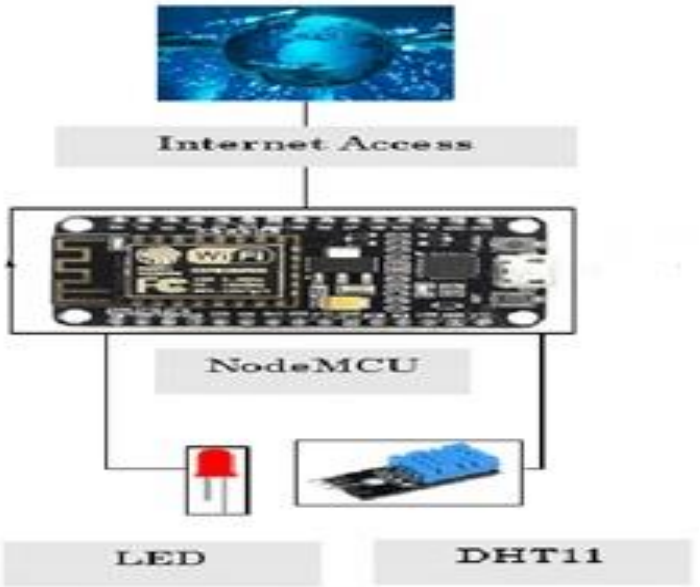
3.2 Method & Algorithm design

3.2.1 Method-1: Working

The user enters his Wi-fi credentials into the source code along with the receiver mail. Then the connections are made and the code is loaded to the esp8266 when the temperature data is sent through the controller if the temperature is greater than the threshold value the led and the buzzer start blinking/buzzing every 2 secs this is done to indicate the temperature is above the threshold value. When the temperature is above the threshold value the SMTP sends an email to the receiver that the temperature is above the threshold after sending the email the controller waits for 5 minutes to resend the message about the temperature being above the threshold during this time if the temperature falls below the threshold value then led and buzzer go to low state and the smtp sends an email stating that the temperature has now reached below the threshold value.

3.3 Project Architecture

3.3.1. Architecture Diagram



Architecture of NodeMCU working

With in the system the DHT11 is used to gather the temperature data. The Oled is used to display information on the actuator side, the SMTP uses the internet access to send data through email.

3.3.2 Data Flow Diagram

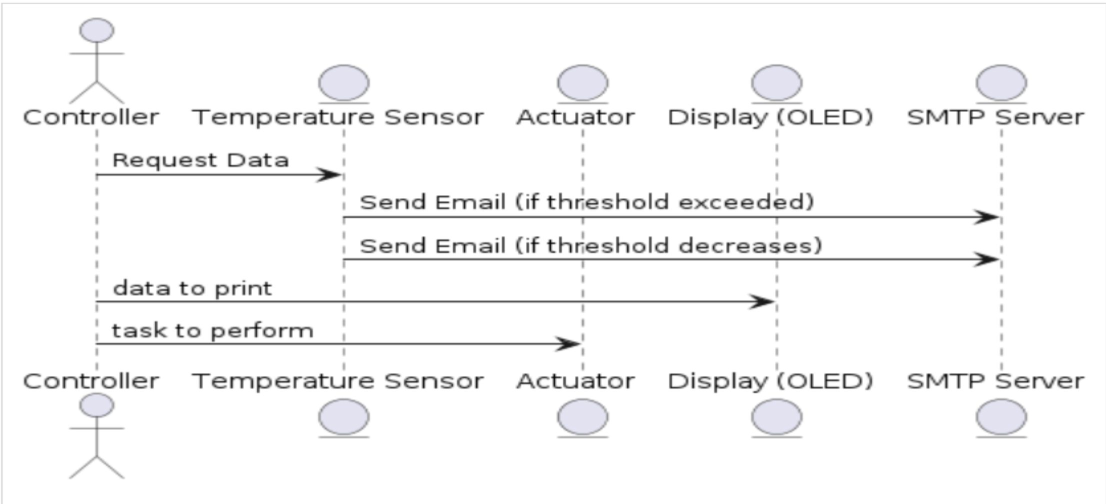


Diagram 1: Data flow diagram

User: The User, represented as an external entity, initiates the data flow by requesting information from the system.

Temperature Sensor: The Temperature Sensor, another external entity, senses the temperature with in the cold storage environment and sends the temperature data to the system for processing.

Database: The Database, depicted as a central data repository, receives the temperature data from the Sensor. It stores and manages the data for further analysis and retrieval.

Actuator: The Actuator, an internal component of the system, receives instructions or commands from the Database based on the analyzed temperature data. It may trigger actions such as adjusting cooling systems or activating alarms.

Display (OLED): The Display, represented as an OLED screen, receives data from the Database and visually presents information such as current temperature readings or system status to users or operators.

SMTP Server: The SMTP Server, another external entity, is responsible for sending email notifications if the temperature within the cold storage exceeds a predefined threshold. It receives alerts from the Database and forwards them to designated recipients for timely action.

3.3.3 Class Diagram

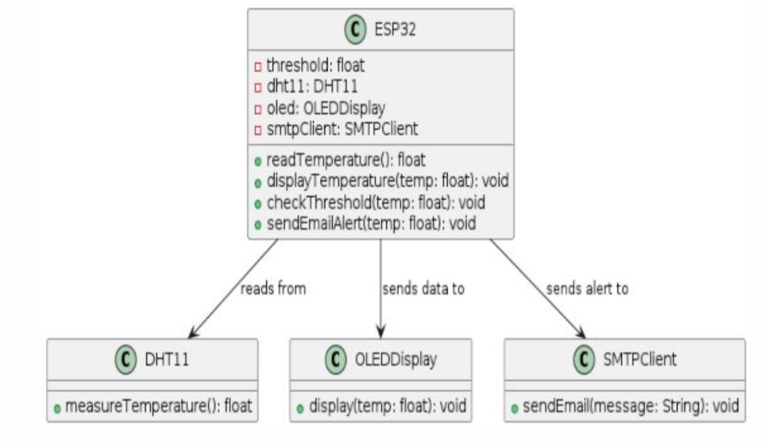


Diagram 2: Class Diagram

3.3.4 Use Case Diagram

The temperature is monitored by the esp8266 module and when the temperature reaches above the threshold value then the led and buzzer start blinking and buzzing begin and smtp sends an email and it continues the process of monitoring the temperature.

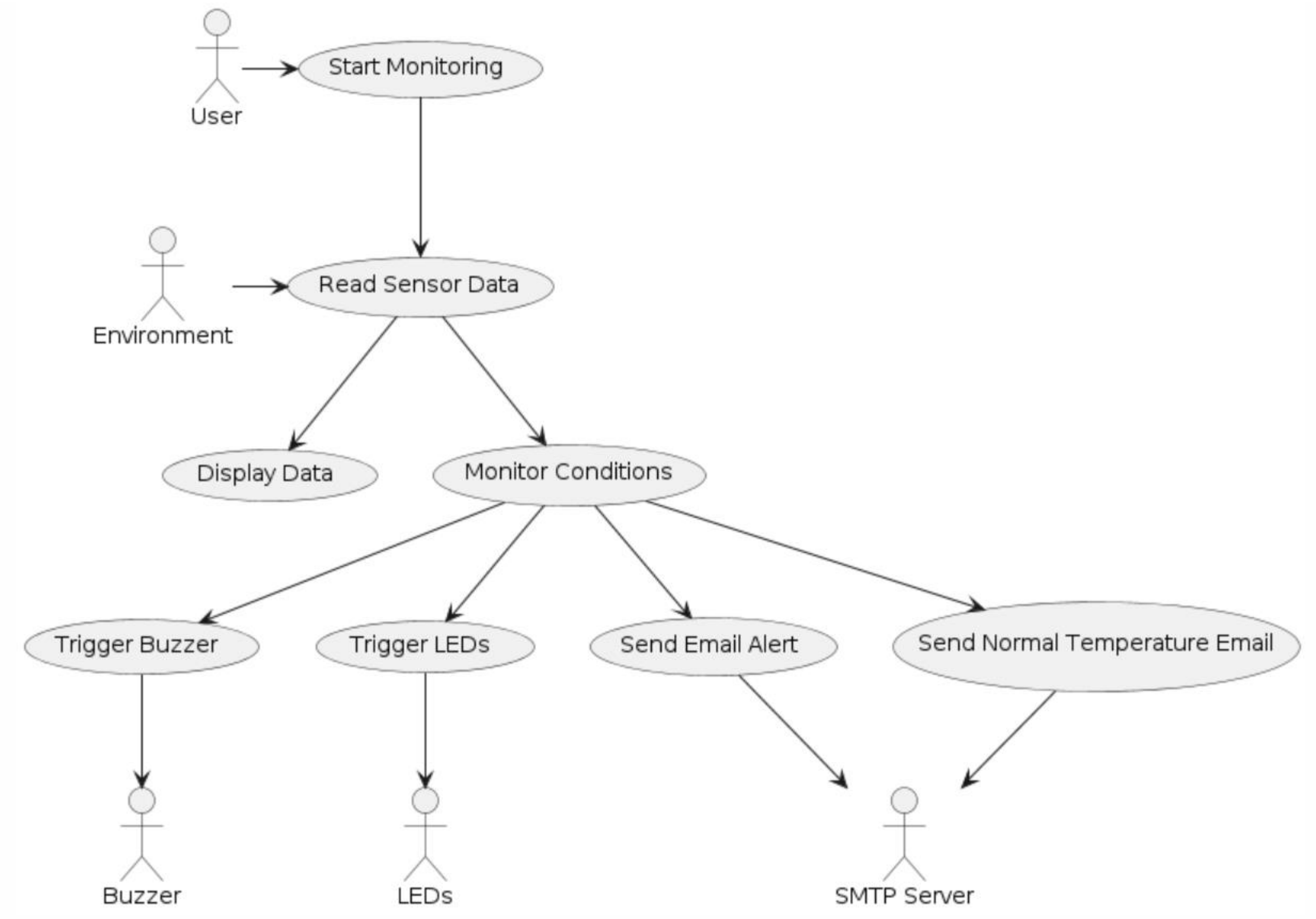


Diagram 3: Use case Diagram

- **Start Monitoring:** User starts the monitoring process.
- **Read Sensor Data:** System reads data from the DHT11 sensor.
- **Display Data:** System displays the data on the OLED display.
- **Monitor Conditions:** System checks if the temperature is above or below the threshold.
- **Trigger Buzzer:** System activates the buzzer if the temperature exceeds the threshold.
- **Trigger LEDs:** System activates the LEDs if the temperature exceeds the threshold.
- **Send Email Alert:** System sends an email alert if the temperature exceeds the threshold.
- **Send Normal Temperature Email:** System sends an email when the temperature returns to normal.

3.3.5 Sequence Diagram

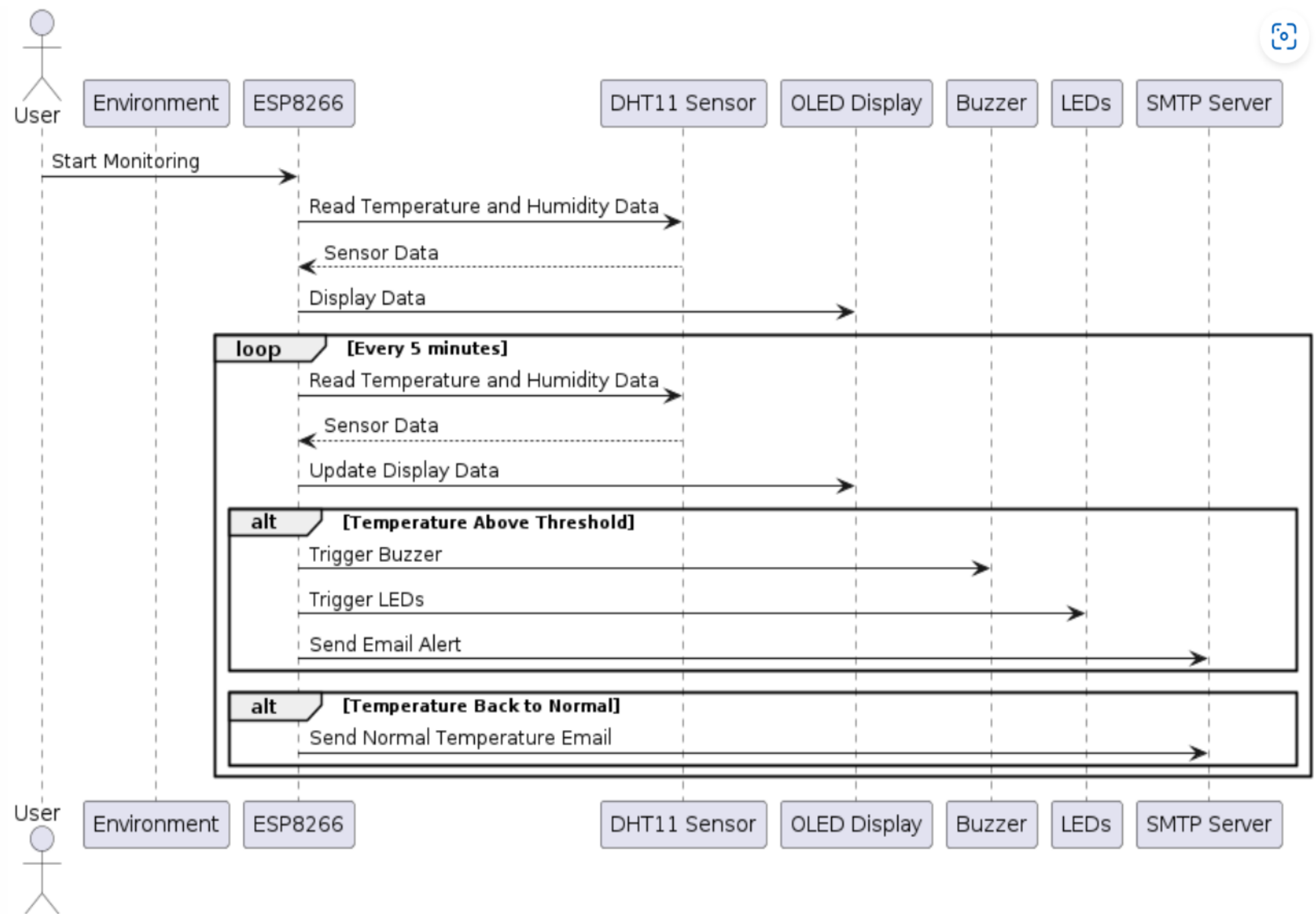


Diagram 4: Sequence diagram

3.3.6 Activity Diagram

- ESP initialization: - Power supply to the board
- Connect to the Wi-fi
- Receive data from DHT11
- Check for Threshold values
- Send corresponding actions to Actuators and SMTP
- Wait for Interval of 5 mins before sending information back through SMTP

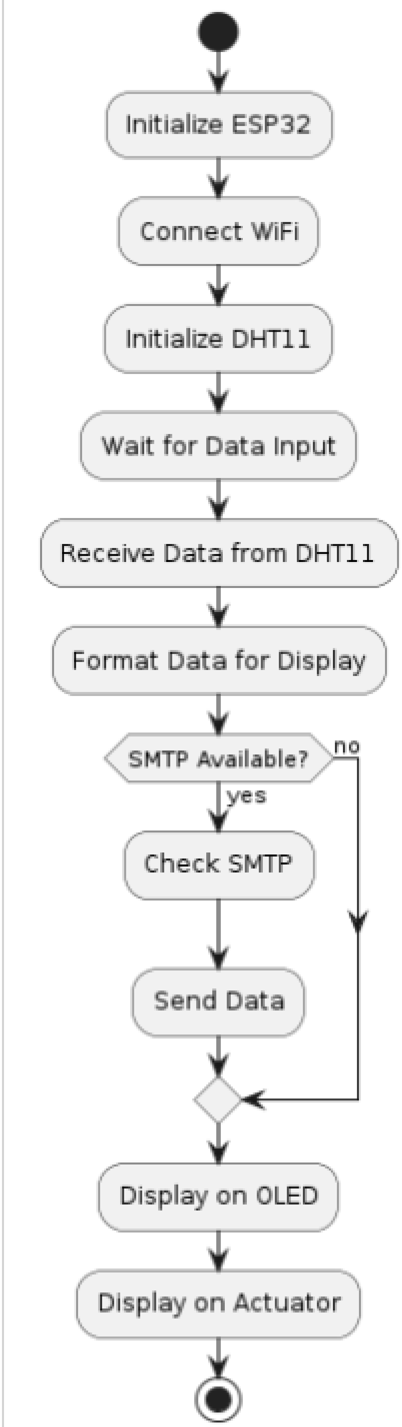


Diagram 5: Activity Diagram

Chapter 4

Implementation & Testing

4.1 Code Block

```
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP8266WiFi.h>
#include <base64.h>
#include <WiFiClientSecure.h>
#include <WebSocketsServer.h> // Include the WebSocket library

//#WebSocketsServer webSocket = WebSocketsServer(81);
#define DHTPIN D5       // Pin connected to the DHT sensor
#define DHTTYPE DHT11    // Type of DHT sensor (DHT11)

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

#define LED_PIN D4       // Pin connected to the LED
#define LED_PIN2 D8      // Pin connected to the second LED
#define BUZZER_PIN D6    // Pin connected to the Buzzer

// Temperature threshold in Celsius
DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
const char* ssid = "ACTFIBERNET";
const char* password = "47909456";
const char* smtpServer = "smtp.gmail.com";
const int smtpPort = 465;  // SMTP port (usually 465 for SSL/TLS)
```

```
const char* emailFrom = "t35065834@gmail.com";
const char* emailPassword = "bymc wrqk pzua qmik";
const char* emailTo = "2111cs050106@mallareddyuniversity.ac.in";
const char* subject = "Temperature Alert";

unsigned long lastEmailTime = 0;
unsigned long highTempStartTime = 0;
bool isTempHigh = false;

WebSocketsServer webSocket = WebSocketsServer(81); // WebSocket server on port 81
float TEMP_THRESHOLD = 25.0;

void webSocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t length) {
  switch(type) {
    case WStype_TEXT:
      Serial.printf("Received text: %s\n", payload);
      // Parse the incoming message and update temperature threshold value accordingly
      TEMP_THRESHOLD = atof((const char*)payload);
      Serial.printf("New temperature threshold: %.2f\n", TEMP_THRESHOLD);
      break;
    default:
      break;
  }
}void setup() {
  Serial.begin(115200);
  delay(2000); // Give time for the serial monitor to initialize
  pinMode(LED_PIN, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(LED_PIN, HIGH); // Keep LED on during normal temperature readings
  dht.begin();

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
```

```
  display.display();
  delay(2000);
  display.clearDisplay();

  // Connect to WiFi
  connectToWiFi();
  webSocket.begin();
  webSocket.onEvent(webSocketEvent);
}

void loop() {
  delay(2000);  // Wait for 2 seconds between measurements
  webSocket.loop();

  float temperature = dht.readTemperature();
  if (isnan(temperature)) {

    Serial.println("Failed to read temperature from DHT sensor!");
    return;
  }

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

  // Display temperature on OLED
  display.clearDisplay();
  display.setTextSize(3);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,0);
  display.print("Temp: ");
  display.print(temperature);
  display.print(" C");
  display.display();
```

```
if (temperature > TEMP_THRESHOLD) {
  // Temperature is above threshold
  if (!isTempHigh) {
    // Temperature just crossed the threshold
    highTempStartTime = millis();
    isTempHigh = true;
    // Send an immediate email alert
    sendTemperatureAlert(temperature, "Temperature is above the threshold");
  }

  // Blink LED if temperature is above the threshold
  digitalWrite(LED_PIN, LOW);
  delay(1000);
  digitalWrite(LED_PIN, HIGH);
  delay(1000);

  // Turn on the second LED
  digitalWrite(LED_PIN2, HIGH);

  // Send periodic email alerts every 5 minutes
  if (millis() - lastEmailTime > 300000) { // 5 minutes = 300000 milliseconds
    unsigned long duration = (millis() - highTempStartTime) / 60000; // duration in minutes
    String body = "The temperature has exceeded the threshold for " + String(duration) + " minutes.";
    sendTemperatureAlert(temperature, body.c_str());
    lastEmailTime = millis();
  }

} else {
  // Temperature is below threshold
  if (isTempHigh) {
    // Temperature just dropped below the threshold
    isTempHigh = false;
    sendTemperatureAlert(temperature, "Temperature is back to normal");
  }
  // Keep LED on during normal temperature readings
  digitalWrite(LED_PIN, HIGH);
```

```
    digitalWrite(LED_PIN2, LOW);

    // Activate the buzzer for 5 seconds when the temperature falls below the threshold
    digitalWrite(BUZZER_PIN, HIGH);
    delay(5000);
    digitalWrite(BUZZER_PIN, LOW);
  }
}

void connectToWiFi() {
  WiFi.begin(ssid, password);
  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    delay(500);
    Serial.print(".");
    attempts++;
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi connected");
  } else {
    Serial.println("Failed to connect to WiFi");
  }
  Serial.println("Connected to WiFi");

  Serial.print("Local IP address: ");
  Serial.println(WiFi.localIP());
}


void sendTemperatureAlert(float temperature, const char* body) {
  // Configure secure client
  WiFiClientSecure client;
  client.setInsecure();

  // Connect to SMTP server
  if (!client.connect(smtpServer, smtpPort)) {
```

```
    Serial.println("Connection to SMTP server failed");
    return;
  }
  String fullBody = String(body) + "\nCurrent temperature: " + String(temperature) + " °C";
  if (!sendEmail(client, emailFrom, emailTo, subject, fullBody.c_str())) {
    Serial.println("Failed to send email");
  } else {
    Serial.println("Email sent successfully");
  }
  client.stop();
}

bool sendEmail(WiFiClientSecure &client, const char* from, const char* to, const char* subject, const char*
body) {
  if (!expect(client, "220")) return false;

  // Send HELO command
  client.println("HELO example.com");
  if (!expect(client, "250")) return false;

  // Send login
  client.println("AUTH LOGIN");
  if (!expect(client, "334")) return false;

  // Send username
  client.println(base64::encode(emailFrom));
  if (!expect(client, "334")) return false;

  // Send password
  client.println(base64::encode(emailPassword));

  if (!expect(client, "235")) return false;

  // Send mail from
  client.println("MAIL FROM:<" + String(from) + ">");
  if (!expect(client, "250")) return false;
```

```
  // Send rcpt to
  client.println("RCPT TO:<" + String(to) + ">");
  if (!expect(client, "250")) return false;

  // Send data
  client.println("DATA");
  if (!expect(client, "354")) return false;
  // Send email headers
  client.println("From: " + String(from));
  client.println("To: " + String(to));
  client.println("Subject: " + String(subject));
  client.println();
  // Send email body
  client.println(String(body));

  // End email data
  client.println(".");
  if (!expect(client, "250")) return false;
  // Send QUIT
  client.println("QUIT");
  if (!expect(client, "221")) return false;
  return true;
}
bool expect(WiFiClientSecure &client, const char* str) {
  if (!client.connected()) {
    Serial.println("Error: Not connected to SMTP server");
    return false;
  }
  while (!client.available()) {
    delay(1);
  }
  String rsp = client.readStringUntil('\n');
  Serial.println(rsp);
  return rsp.startsWith(str);
}
```

4.2 Execution Flow

1. User Initiation

- **User Action**: The user provides input for his wifi and email receiver.
- **Uploading** : The code shall be uploaded by the user

2. Authentication

- **Credentials Verification:-** The micro controller try connecting to Wifi using credentials provided.
- **Email verification :-** The micro controller checks for the email and works wether the email can be used to send emails or not.

3. Data Collection

- **DHT :-** The data is generated by DHT11 which is stored and verified by controller
- **Thingspeak** :- The data is collected to the cloud and can be uploaded to create graphs

4. Command Transmission

- **Controller command:-** The controller decides to send email to the cloud and is capable of tasks like these.

5. Actuators Action

- **Devices**: The actuators follow the direction of the controllers and work on the commands they are issued with.

4.3 Testing

4.3.1 Test Cases-1:

- Writing the code in the Arduino Ide and Compiling the Arduino Code if were no error in the code we can directly upload the code Arduino Board.
- If we encounter any error in the code we have to debug it and then we have to compile and upload.

4.3.2 Test Case-2:

- When the DHT11 sensor is unable to read data due to error or wrong pin connection
- When OLED is not properly connected then it does not show display on to the code.

4.3.3. Test Case-3:

- When the code is successfully executed without errors from hardware or debug issue then code is said to be able to perform its remote task successfully.
- The serial monitor observers the reading of Temperature on the serial monitor and if temperature is above the threshold then an email is sent and can be used.

CHAPTER 5

RESULTS

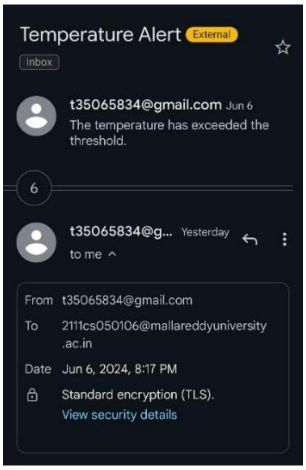
5.1 Resulting Screens:

5.1.1 Screen Shot 1

```
250 2.1.0 OK d9443c01a7336-1f6eccc0bd0sm463565ad.105 - gsmtp
250 2.1.5 OK d9443c01a7336-1f6eccc0bd0sm463565ad.105 - gsmtp
354  Go ahead d9443c01a7336-1f6eccc0bd0sm463565ad.105 - gsmtp
250 2.0.0 OK  1717766457 d9443c01a7336-1f6eccc0bd0sm463565ad.105 - gsmtp
221 2.0.0 closing connection d9443c01a7336-1f6eccc0bd0sm463565ad.105 - gsmtp
Email sent successfully
Temperature: 28.30 °C
```

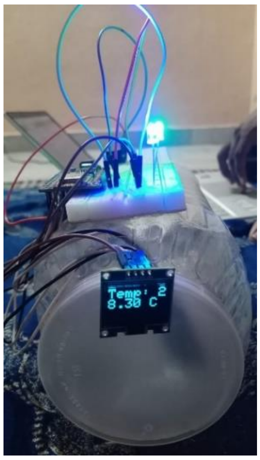
Code Executed

5.1.2 Screen Shot 2



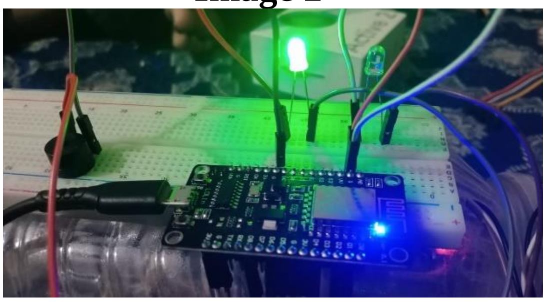
Email Received

Image 1



Temperature is displayed

Image 2



LED Blinking and Buzzer

5.2 Result Summary

The code when successfully executed can send emails when the temperature is above certain temperature and is capable of sending that information through email.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 Conclusion

Using Temperature Monitoring Systems to record the temperature of the freezer, capture data, then sending it to the cloud database using Communication Data and also initiate a warning by using an alarm can help an operational Freezer healthy, so it can keep food fresh dan safe for consuming. We can also view the operation of the Freezer in term of temperature 24 hours using a graphic, so the user can predict problems that occurs based on the history of temperature capture. The user can also start a maintenance process before the freezer is totally damaged and stops functioning.

6.2 Future Scope

We can also view the operation of the Freezer in term of temperature 24 hours using a graphic, so the user can predict problems that occurs based on the history of temperature capture. The user can also start a maintenance process before the freezer is totally damaged and stops functioning. To enhance this Monitoring system, the next step that can be taken can be something like sending a notification using other messaging systems like SMS, Email or Whatsapp.

6.3 References

[1]Motoki, K., Saito, T., Nouchi, R., Kawashima, R., & Sugiura, M. (2018). The paradox of warmth: Ambient warm temperature decreases preference for savory foods. Food quality and preference, 69, 1-9.

[2] J. A. EVANS, A. M. FOSTER, & T. BROWN. (2014). 3rd IIR International Cold Chain Conference. TEMPERATURE CONTROL IN DOMESTIC FREEZERS AND FREEZERS

[3] Zainuddin, Jufrizal, Eswanto, 2016. The Heat Exchanger Performance of Shell and Multi Tube Helical Coil as a Heater through h the Utilization of a Diesel Machine's Exhaust Gas. Aceh International Journal and Technology, vol 5, No.1, 21 – 28.

[4] Stringer, S. C., & Metris, A. (2018). Predicting bacterial behaviour in sous vide food. International Journal of Gastronomy and Food Science, 13, 117-128.

[5] Babiuch, M., Foltýnek, P., & Smutný, P. (2019, May). Using the ESP32 Microcontroller for Data Processing. In 2019 20th International Carpathian Control Conference (ICCC) (pp. 1-6). IEEE. Ding, H. (2017).

[6] Ding, H. (2017). Application and Design of Patient Temperature. Acquisition System Based on Wireless Sensor Network.

Bibliography:

[1] Tavares, J., Velez, F.J., Ferro, J.M., “Application of wireless sensor networks to automobiles,” Measurement Science Review, Vol. 8, Secion 3, pp: 65-70, 2008.

[2] “ZigBee Home Automation Public Application Profile”, ZigBee Alliance, Rev 25, Ver 1.0, 2007.

[3] Yu, Y., Prasanna, V.K., Krishnamachari, B., “Information Processing and Routing in Wireless Sensor Networks,” World Scientific Publishing Co., 2006.

[4] “Zigbee Sensor Tag Data Sheet”, Pultronics Inc.

[5] George W. Irwin, Jeremy Colandairaj and William G. Scanlon, “An Overview of Wireless Networks in Control and Monitoring,” International Conference on Intelligent Computing (CHINE 2006), Vol. 4114, pp. 1061-1072, 2006.

[6] Cai Bin, Jin Xinchao, Yan Shaomin; Yang Jianxu; Zhao Xibin; Zou Guowei; “ Application research on temperature WSN nodes in switchgear assemblies based on TinyOS and ZigBee,” 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 201, pp: 535 – 538, 2011

[7] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, “An Advanced Wireless Sensor Network for Health Monitoring,” Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), April 2-4, 2006.

[8] Zhang Zuojing; Zhang Haihui; "Design of Wireless Monitoring and Warning System for Protected Agriculture Environment," 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), pg: 1 – 5, 2010.

[9] Emil Jovanov, Aleksandar Milenkovic, Chris Otto and Piet C de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," Journal of NeuroEngineering and Rehabilitation, pp: 2:6, 2005.

[10] Namjun Cho; Seong-Jun Song; Sunyoung Kim; Shiho Kim; Hoi-Jun Yoo, "A 5.1-μW UHF RFID tag chip integrated with sensors for wireless environmental monitoring," Proceedings of the 31st European Solid-State Circuits Conference (ESSCIRC), pp: 279 – 282, 2005.

[11] "P89V51RB2/RC2/RD2: 8-bit 80C51 5 V low power 16/32/64 kB Flash microcontroller with 1 kB RAM," Datasheet from Philips Electronics, Rev. 03, 02 December 2004.

[12] "LM35 Precision Centigrade Temperature Sensors," National Semiconductor Corporation, November 2000.

COLD STORAGE MONITORING

G.Sailaja [1],Ponnala Naveen[2], Anand Charan[3], Surasani Manideep[4],

Inala Venkata Hari Paniindra[5]

[1-4]Department Of Internet Of Things, Malla reddy University, Hyderabad, Telangana, India.
[5]Assistant Professor, Department of Internet of Things, Malla reddy University, Hyderabad, Telangana, India.

g_sailaja@mallareddyuniversity.ac.in [1],2111cs050090@mallareddyuniversity.ac.in[2]

2111cs040072@mallareddyuniversity.ac.in[3] ,2111cs050117@mallareddyuniversity.ac.in[4], 2111cs050106@mallareddyuniversity.ac.in[5]

ABSTRACT

IoT technology has enabled the development of useful application to improve the quality of life. We can also the need to develop other application to make an effective use of the devices and the internet connectivity. Our project try's to solve the problems occuring due to lack of monitoring of our goods. In this regard we have taken the first to solve the problem by proposing a project which can Monitor the temperature of goods present in a container and notify the person in-charge according to the constraints he/she have provided. For this project we will be using ESP32 as the processing unit due to wireless wi-fi connectivity and upload HTML code to monitor temperature and daily usage of our model through the local network connected to the device.

Keywords: Monitor ,ESP32,Alerts,Cold Storage,NodeMCU,DHT11 Sensor

I. INTRODUCTION

The Internet of Things (IoT) is a new, but at the same time is an old term. It is already mentioned by Kevin Ashton in 1999. Since then, the use of this term has blossomed and major companies have predicted an increase in IoT. One prediction is that the number of connected things in the world will have thirtyfold increases between 2009and 2020, thus by 2020 there will be 26 billion components are connected in IoT devices.

The components we are using in this equipment are ESP32 wroom dev kit, temperature sensor bsm280, led's, Bread board are the things which are used to build a working prototype of the device.

The main justification of ESP32 module is easy to usage in connecting to wireless networks and being capable of connecting to commonly available communication devices like the Wi-fi and Bluetooth which are proven to be useful and do not need an external connections or modules to establish a connection. The bsm280 is an temperature and humidity sensor capable of calculating humidity and temperature and can also calculate altitude if required directly to.

II. METHODOLOGY

We connect our esp32 module with temperature sensor(BSM280) and use webserver connection to integrate these two by using Software libraries like JSON, The web development is done to display to the temperature readings based on the constraints if the temperature is under the specified constraints the green led is seen glowing. When the temperature increases above the limit a led will glow notify the in-charge about temperature.

III. MODULE DESCRIPTION

Now-a-days the control of temperature has become an important task in many operations. The method that we used to build, can be operated manually and automatically. So, the method we used to control the temperature is to initialize the serial monitor, temperature sensor test, and Webserver connection. Then we measure the temperature threshold value or the set pointvalue i.e., we will be using the setpoint to compare the current temperature that to be measured and it switch on the led and email notification according to the given condition

A. System Architecture & Components

The system architecture and components of IoT based Temperature Monitoring System Using NodeMCU & Blynk App. The system consists of four main components: the NodeMCU 32s module, DHT11 Sensor, 5V 2-Channel Relay Module, DS18B20 Temperature Sensor as shown in the figure 2.

ESP32 Wroom dev kit

The ESP32-WROOM module features a powerful 32-bit dual-core microcontroller running at up to 240 MHz’s It's based on the Tensilica Xtensa LX6 architecture, which is highly efficient and offers great performance. One of the key features of the ESP32 is its built-in Wi-Fi and Bluetooth connectivity, which makes it ideal for IoT projects requiring wireless communication. It supports various Wi-Fi modes (station, access point, or both simultaneously) and Bluetooth Classic and BLE. The ESP32-WROOM module includes a wide range of peripheral interfaces, such as SPI, I2C, UART, ADC, DAC, and more, making it versatile and suitable for a variety of projects

Fig. 1. NodeMCU ESP32

DHT 11 Temperature and Humidity sensor

The **DHT11** is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for user.

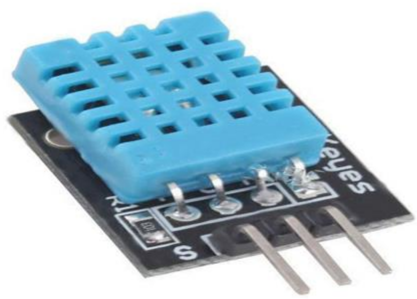


Fig. 2 DHT11 Temperature and Humidity Sensor

OLED

Organic Light Emitting Diode (OLED) technology is a display technology that uses organic compounds to emit light when an electric current is applied. Unlike traditional LCDs that require a backlight, OLEDs are self-emissive, meaning each pixel generates its own light. This characteristic allows for deeper blacks and higher contrast ratios, as individual pixels can be turned off completely

Fig. 3. OLED display

V. RESULTS & DISCUSSION

Process

1.Setting up Wi-fi modules:-

Connect to set up the Wi-Fi using credentials and the email credentials and if authentication is done and is sent.

2.Set the threshold values:-

By default, 25 degrees is set as threshold values you can do so. The values are compared with readings and are updated when constraints are checked.

3.Notification:-

The user is notified if the temperature is not in the constraint of temperature he/she has set using the objects through the notification of LED. And email

Results:

Fig 4 Working of Model

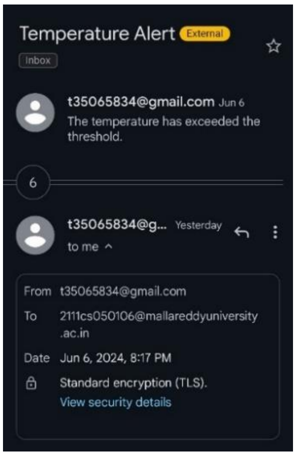


Fig 5 When temperature is not in limit email is sent

VI. CONCLUSION

The cold storage monitoring is an useful IOT project for checking the temperature of our closed storage so we can ensure by getting notified when we require to.

VII. FUTURE SCOPE

Integerating with automation devices and can connect with an RFID or GSP module to access the temperature .

COLD STORGE DEVICE

3rd year Department of IOT

MALLA REDDY UNIVERSITY

ABSTRACT

Iot technology has enabled the development of useful application to improve the quality of life. We can also the need to develop other application to make an effective use of the devices and the internet connectivity. Our project try's to solve the problems occuring due to lack of monitoring of our goods. In this regard we have taken the first to solve the problem by proposing a project which can Monitor the temperature of goods present in an container and notify the person in-charge according to the constraints he/she have provided .For this project we will be using ESP32 as the processing unit due to wireless wifi connectivity and upload HTML code to monitor temperature and daily usage of our model through the local network connected to the device.

OBJECTIVES

To establish monitoring system which is capable of alterting user if the containing components are well preserved or not by obersving temperature and humidity of the component in a closed container.

The use of a freezer for food is to preserve food ingredients for use in the future. This is because food storage also prevents the growth of bacteria, fungi, and others [4]. But not just preserving food, freezers also play a role in the taste and texture of stored food. Cooling food can make some foods taste better like chilled fruit and drinks

We connect our esp32 module with temperature sensor(BSM280) and use webserver connection to integrate these two by using Software .

MATERIALS & METHODS

Components:-

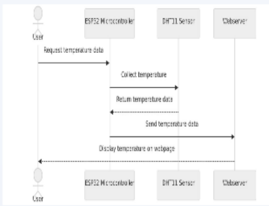
ESP32

DHT11

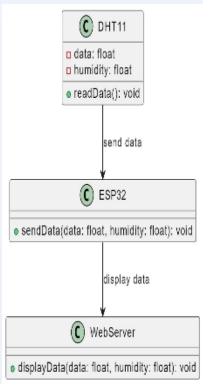
OLED

BUZZER

We use these sensors and actuator to create our smart object capable of sensing and acting on the constraints provided to it.



The primary function of the web server is is to determine the temperature on the web page and is also responsible for the constraints which can be used to check the working of the sensor if data is being collected and determine the quality of stored goods.



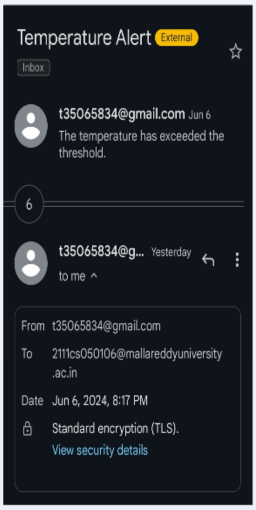
RESULTS

Through this project we can monitor the temperature reading through web page .

We can set the constraints for

1.Setting up SMTP module:-

Connect to set up the access point and search for the web page address "esp/local" where you will find the temperature monitoring web page. The is done through the help of DNS and WiFi.API



CONCLUSION

Ths project is used in places wherer we can be notified about the spoilage of our goods through the usage of OLED display and the use of BUZZER for notification. To ensure the environment is sustainable and can be monitored with the web hosted web page can make the difference between this project application and not using the application

Temperature Monitoring Systems to record the temperature of the freezer, capture data, then sending it to the cloud database using Communication Data and also initiate a warning by using an alarm can help an operational Freezer healthy, so it can keep food fresh dan safe for consuming. We can also view the operation of the Freezer in term of temperature 24 hours using a graphic, so the user can predict problems that occurs based on the history of temperature capture.

REFERNCES

1. Motoki, K., Saito, T., Nouchi, R., Kawashima, R., & Sugiura, M. (2018). The paradox of warmth: Ambient warm temperature decreases preference for savory foods. Food quality and preference, 69, 1-9.
2. J. A. EVANS, A. M. FOSTER, & T. BROWN. (2014). 3rd IIR International Cold Chain Conference. TEMPERATURE CONTROL IN DOMESTIC FREEZERS AND FREEZERS .

GUIDED BY

Mrs.G Sailaja
Assistant professor

PRESENTED BY

I. Venkata Hari Paniindra
P. Naveen
Anand Charan
S.Manideep