SMART FOOD MONITORING SYSTEM

MINI PROJECT REPORT

Submitted by

VASANTHAKUMAR M [RA2211017010055]
ROMANUS IRUDAYA RAJ R [RA2211017010070]
SURENDRAN T [RA2211017010071]

in partial fulfilment for the

Subject: 21EIC306J - Instrumentation system design

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & INSTRUMENTATION ENGINEERING

of

COLLEGE OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District
MAY 2025 -26

LAB PROJECT REPORT COVERSHEET

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

COLEGE OF ENGINEERIG AND TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

SIXTH SEMESTER, 2024-2025 (EVEN SEMESTER)

PROJECT TEAM MEMBERS	**:**	VASANTHKUMAR M, ROMANUS IRUDAYA RAJ R, SURENDHRAN T.
REGISTER NO.	**:**	RA2211017010055, RA2211017010070, RA2211017010071.
TITLE OF THE PROJECT	**:**	SMART FOOD MONITORING SYSTEM.
LAB SUPERVISOR	**:**	**DR. K. VIBHA**

REG. NO.		
MARKSPLITUP	**MAXIMUMMARKS**	**MARKSOBTAINED**
Novelty in the project/abstract	**5**	
Level of understanding the design/configuration	10	
Individual contribution to The project	5	
Report writing	5	
Total	25	

REPORT VERIFICATION:

Lab supervisor Signature with date:

TABLE OF CONTENTS

1. ABSTRACT

The increasing global focus on food safety and sustainability has created a demand for efficient, accurate, and affordable monitoring systems that can track food quality throughout its shelf life. This paper presents a smart food monitoring system using Arduino UNO, designed to assess key environmental parameters such as temperature, humidity, and gas concentrations that affect food freshness. The system incorporates multiple sensors, including DHT11 for temperature and humidity, and MQ-series gas sensors to detect spoilage-indicating gases such as ammonia and hydrogen sulfide. The collected data is processed by the Arduino UNO and displayed on an LCD screen, providing real-time monitoring of food storage conditions.

To enhance accessibility, the system is equipped with a Bluetooth module that enables data transmission to a smartphone application, allowing users to monitor food quality remotely. Alerts are programmed to notify users if critical thresholds are exceeded, signaling potential food spoilage. This research demonstrates the potential of using Arduino-based systems to provide a cost-effective, reliable, and scalable solution for real-time food quality monitoring, contributing to reduced food waste and improved consumer health.

2. INTRODUCTION

Food safety and preservation have become increasingly important in recent years, with growing concerns about food spoilage, waste, and contamination. As food products are stored and transported, various environmental factors such as temperature, humidity, and gas levels play a critical role in determining their freshness and quality. Monitoring these factors in real-time can help prevent food spoilage, reduce waste, and ultimately contribute to better food security and safety. In response to these needs, the development of affordable and effective monitoring solutions has gained significant attention, particularly for smaller storage environments and household applications.

This paper presents a smart food monitoring system that uses the Arduino UNO microcontroller, a DHT11 sensor, and an MQ4 gas sensor to monitor environmental conditions that affect food quality. The Arduino UNO serves as the central processing unit for this system, while the DHT11 sensor measures temperature and humidity levels two key indicators of an environment suitable for food storage. The MQ4 sensor detects methane gas, which can accumulate as a byproduct of food spoilage, particularly for perishable items. These three components work together to provide an efficient, low-cost system for monitoring conditions in small-scale food storage areas, like household pantries or compact food storage units.

The system is designed to be user-friendly and economical, focusing on essential parameters that signal early signs of spoilage. If temperature, humidity, or gas levels exceed preset thresholds, the system can trigger alerts to inform users of potential spoilage risks. This proactive monitoring allows users to make informed decisions about food usage, helping to reduce waste and maintain food quality.

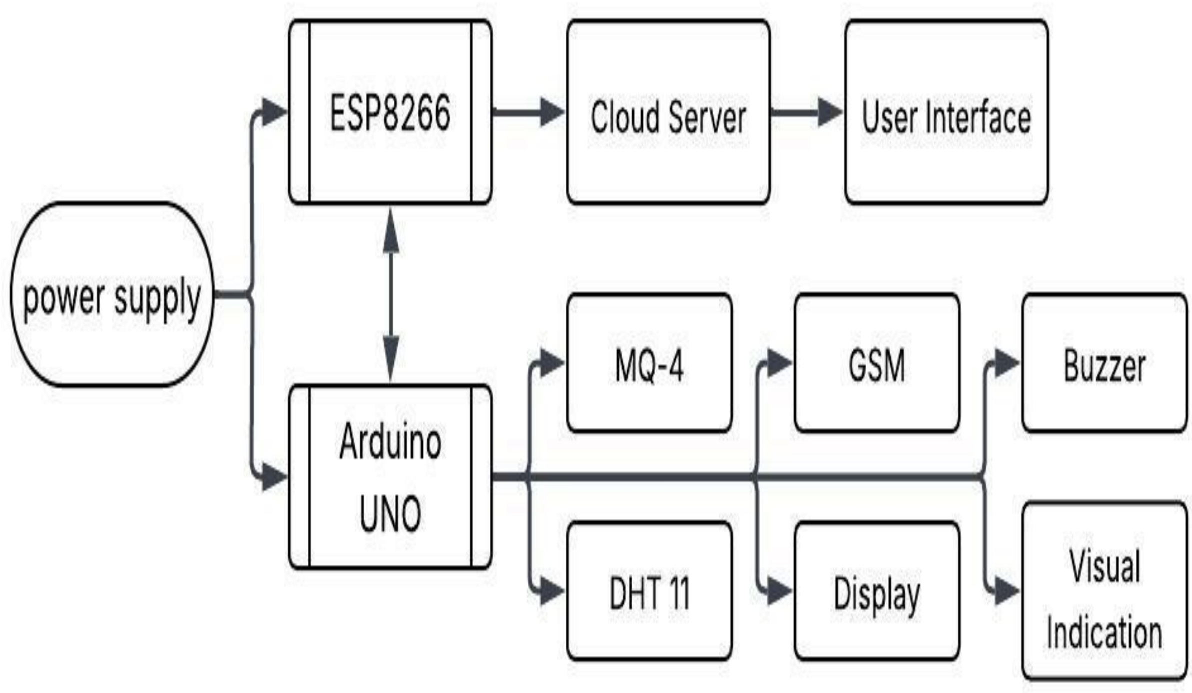
The simplicity and cost-effectiveness of this system make it accessible for widespread adoption in households or small businesses. By leveraging the basic components of an Arduino UNO, DHT11 sensor, and MQ4 gas sensor, this smart food monitoring system offers an innovative yet straightforward approach to real-time food quality monitoring.

3. HARDWARE AND DISCRIPTION

3.1. Block Diagram:

Figure 1 represents the general block diagram of the project. The primary objective of a smart food monitoring system is to ensure the safe storage and quality maintenance of food by continuously monitoring environmental parameters like temperature, humidity, and gas emissions. It aims to prevent food spoilage and contamination by using IoT-based sensors to detect unfavorable conditions in real-time. The system provides automated alerts and can adjust conditions as needed, helping reduce food waste, maintain hygiene, and extend shelf life. Additionally, it supports data collection for analysis and future improvements in food storage practices.

Figure 1: Block Diagram



Objective:

- **Preserve Food Quality and Extend Shelf Life:** By continuously monitoring critical parameters like temperature, humidity, and gas levels, smart food monitoring systems help maintain optimal storage conditions, which can extend the shelf life of perishable goods

- **Reduce Food Wastage:** With real-time alerts and automated adjustments, these systems help prevent food spoilage by responding immediately to changes in storage conditions, reducing food waste and loss during transportation or storage

- **Enhance Food Safety:** By detecting spoilage indicators, such as gas emissions from decomposing produce, these systems ensure food safety for consumers by identifying and isolating contaminated items before they reach the market.

3.2. Hardware Components:

As per the block diagram, the main hardware components used in this project are listed below:

- **Arduino UNO & ESP8266**
- **DHT-11 Sensor**
- **MQ-4 Methane sensor**
- **LCD Display**
- **Buzzer & LED**
- **PCB Board**
- **Connecting Wires**

ARDUINO UNO:

The Arduino UNO is one of the most popular microcontroller boards widely used in electronics and embedded systems. Designed for both beginners and experts, it allows users to create a variety of projects, from simple LED blink programs to complex robotics. Below is a detailed look at its features, specifications, and applications, providing a comprehensive understanding of what makes the Arduino UNO so versatile. The Arduino UNO was introduced as an open- source microcontroller development board based on the ATmega328P microchip.

Key Specifications and Features:

- **Microcontroller:** The UNO is based on the ATmega328P microcontroller, which operates at 16 MHz and includes 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM. These specifications are sufficient for handling basic to intermediate tasks.

- **Digital I/O Pins:** It has 14 digital I/O pins, 6 of which can provide PWM (Pulse Width Modulation) output. These pins can be used to interface with various digital devices, such as LEDs, buttons, and other digital components.

- **Analog Input Pins:** The board has 6 analog input pins, labeled A0 through A5, which can be used to measure varying voltage levels, typically from sensors that output analog signals.

- **Power Supply:** The board can be powered via USB (typically from a computer) or through an external 9-12V power source. The onboard voltage regulator ensures that components receive a stable 5V or 3.3V.

- **Programming:** The Arduino UNO can be programmed using the Arduino IDE (Integrated Development Environment), which supports a simplified version of C++. It can be programmed through USB, and no external programmer is required.

Hardware Design and Pin Layout:

The UNO has a standard design that includes a USB port, a power jack, a reset button, and a series of headers for connecting components. The board also includes an ICSP (In-Circuit Serial Programming) header, which allows for programming the ATmega328P directly.

- **Digital and PWM Pins:** The 14 digital pins (D0-D13) can act as input or output pins, controlled by code. Of these, pins D3, D5, D6, D9, D10, and D11 can produce PWM signals, which are useful for dimming LEDs or controlling motors with variable speed.

- **Analog Pins:** The analog pins, A0-A5, can read voltages in the range of 0-5V and convert them into digital value between 0 and 1023. This is useful for sensors like potentiometers or temperature sensors, which provide variable voltage outputs based on physical changes.

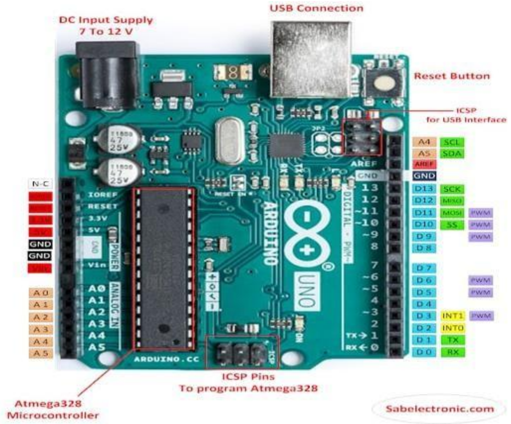
Advantages of Using Arduino UNO:

- **Ease of Use:** With the IDE's user-friendly interface, programming and debugging are accessible even to non-technical users.
- **Open-Source:** As an open-source hardware and software platform, all documentation, schematics, and codes are freely available, encouraging learning and modification.
- **Vast Community and Resources:** The Arduino UNO has one of the largest online communities, which provides tutorials, forums, and troubleshooting guides, making it easier to find support or inspiration for new projects.
- **Affordable:** Compared to other microcontrollers, the Arduino UNO is very affordable, which lowers the entry barrier for beginners and hobbyists.
-

Limitations:

From the figure 2, Despite its versatility, the Arduino UNO does have limitations. It has limited memory and processing power compared to more powerful boards, making it unsuitable for high-performance or memory-intensive applications. Additionally, it lacks wireless connectivity features like Wi-Fi. In summary, the Arduino UNO is an essential tool in the maker movement, offering a robust and accessible platform for prototyping and learning about embedded systems. Its extensive support and ease of use have made it a staple in classrooms, workshops, and labs worldwide.

Figure 2: ARDUINO UNO



ESP8266 Wi-Fi Module:

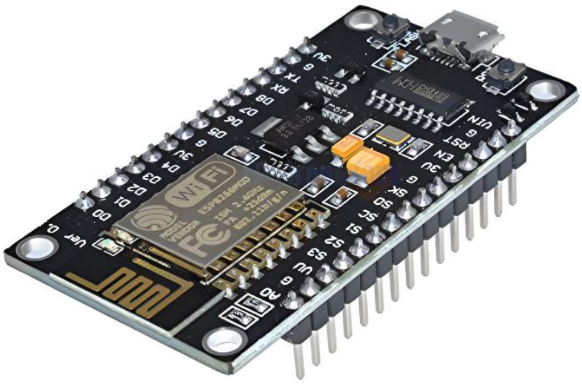
The ESP8266 is a highly integrated, cost-effective Wi-Fi module developed by Esp ressif Systems that provides full internet connectivity to embedded devices. It is widely used in IoT (Internet of Things) projects due to its low power consumption, compact size, and built-in networking capabilities.

Key Features:

- **Built-in Wi-Fi (802.11 b/g/n):** Enables wireless connectivity to local networks or the internet.
- **Integrated TCP/IP Protocol Stack:** Allows easy internet communication using standard network protocols.
- **Microcontroller Capability:** It features a 32-bit RISC CPU (Tensilica L106) that can operate at up to 160 MHz
- **Flash Memory:** Typically, available with 512 KB to 4 MB flash memory, depending on the model.
- **GPIO Pins:** Multiple general-purpose input/output pins for connecting sensors, LEDs, relays, etc.
- **Serial Communication:** Communicates via UART (RX/TX), making it easy

to interface with Arduino and other controllers.

- **Low Power Consumption:** Supports deep sleep modes, making it ideal for battery-powered IoT applications.
- **Operating Voltage:** 3.3V logic level.



DHT11 Sensor:

The DHT11 is a low-cost, digital temperature and humidity sensor widely used in embedded systems and electronics projects. This sensor provides reliable and reasonably accurate temperature and humidity readings, making it ideal for various applications, from weather monitoring stations to home automation. Below is a detailed overview of the DHT11 sensor, including its specifications, features, and applications, as well as how to interface it with microcontrollers like Arduino.

Overview of the DHT11 Sensor:

The DHT11 sensor is a part of the DHT series of sensors, which are specifically designed for measuring ambient temperature and humidity levels. The DHT11 integrates a thermistor (for temperature measurement) and a capacitive humidity sensor onto a single chip, producing a digital signal output that is easy to read with a microcontroller. Its low cost, simplicity, and ease of use have made it a popular choice among beginners and professionals alike.

Key Specifications and Features:

- **Temperature Range:** DHT11 can measure temperatures from 0°C to 50°C with an accuracy of ±2°C. While this range is suitable for many basic applications, it may not be ideal for extreme temperatures.
- **Humidity Range:** It can measure humidity levels from 20% to 90% relative humidity (RH) with an accuracy of ±5% RH, which makes it sufficient for typical indoor and outdoor environments.
- **Sampling Rate:** DHT11 has a sampling rate of 1 reading per second (1 Hz), which means it updates its measurements every second. This rate is adequate for most applications but may not be suitable for rapid real-time monitoring.
- **Output:** The DHT11 outputs data as a digital signal in a serialized form, allowing it to communicate easily with microcontrollers. This eliminates the need for complex signal conditioning circuits.
- **Power Requirements:** It operates at a voltage of 3.3V to 5V and consumes very low power, making it suitable for battery-powered applications.

The sensor has a compact form factor and typically comes with three pins: VCC, GND, and Data. Some modules also have a fourth pin (NC or not connected) for additional stability. The Data pin is a single-wire communication interface, which simplifies the wiring and reduces pin usage on the microcontroller.

Internet Working and Communication Protocol:

The DHT11 uses a resistive humidity sensor and a thermistor for temperature measurement. The sensor uses a single-wire proprietary protocol for communication. Here’s a brief outline of how data transmission works:

- **Start Signal:** The microcontroller sends a start signal by pulling the data line low for a certain period, alerting the DHT11 to begin transmission.

- **Response:** After receiving the start signal, the DHT11 pulls the data line low for 80 microseconds and then high for another 80 microseconds.

- **Data Transfer:** The sensor sends temperature and humidity data in a 40-bit sequence. The data is transmitted as binary and consists of integer values representing humidity and temperature, followed by a checksum for data integrity verification.

- **Checksum:** The last 8 bits of the transmission are a checksum, which helps ensure the transmitted data is accurate. The microcontroller checks the checksum against the received data to confirm the integrity of the data received.

Application of DHT11:

The DHT11's reliability, low power consumption, and low cost make it suitable for various applications, including:

- **Weather Stations:** The DHT11 is commonly used in DIY weather stations for measuring temperature and humidity. It can provide real-time data for weather prediction models or simply to track indoor/outdoor climate changes.

- **HVAC Systems:** HVAC (Heating, Ventilation, and Air Conditioning) systems often use sensors like the DHT11 to monitor room temperature and humidity and adjust heating or cooling levels accordingly.

- **Home Automation:** In smart homes, the DHT11 is used to monitor room climate, helping systems control the thermostat or dehumidifier.

- **Agriculture:** Farmers use temperature and humidity sensors to monitor greenhouse conditions, ensuring optimal growth conditions for plants.

- **Environmental Monitoring:** The DHT11 can be used in environmental monitoring systems to track climate conditions in various habitats or regions, from urban areas to rainforests.

The microcontroller checks the checksum against the received data to confirm the integrity of the data received. The DHT11 temperature and humidity sensor is a practical and economical choice for a wide range of applications projects.

Advantages and Limitations of DHT11:

Advantages:

- Easy to Interface: With a single-wire digital output, it is easy to interface with microcontrollers without needing ADC (Analog-to-Digital Conversion) or complex circuitry.

- Cost-Effective: DHT11 is one of the most affordable temperatures and a humidity sensor available, making it suitable for large-scale projects or educational use.

- Low Power Consumption: Its low operating current makes it ideal for battery powered applications or low-power devices.

Limitations:

1. **Limited Accuracy:** The ±2°C temperature accuracy and ±5% humidity accuracy may not be adequate for applications requiring high precision.

2. **Narrow Range:** With a limited temperature range (0-50°C) and humidity range (20-90%), the DHT11 is unsuitable for extreme conditions.

3. **Slow Sampling Rate:** The 1 Hz sampling rate may be too slow for real-time applications that require rapid updates.

Figure 3: Dht-11 sensor.

MQ-4 METHANE GAS SENSOR:

The MQ-4 methane gas sensor is a popular choice for detecting natural gases like methane (CH_4) in various environments. It is widely used in safety monitoring applications due to its ability to detect gas leaks, which can prevent potential fire hazards or explosions. This sensor offers a simple, cost-effective way to add gas detection capabilities to projects and is widely used in industrial safety, home automation, and environmental monitoring.

Overview of the MQ-4 Methane Gas Sensor:

The MQ-4 sensor is part of the MQ series of gas sensors, which are specifically designed to detect various types of gases. The MQ-4 focuses on detecting methane (CH_4), making it suitable for applications in environments where natural gas is used, such as homes and industrial facilities. Methane is a highly flammable gas that can pose serious safety risks, making early detection critical.

Key Specifications and Features:

- **Detection Range:** The MQ-4 can detect methane gas concentrations in the range of 200 ppm to 10,000 ppm (parts per million), which makes it suitable for both low and high methane concentrations.

- **Sensitivity:** The sensor's sensitivity to methane allows it to detect even low concentrations in the air, providing reliable results for leak detection.

- **Operating Voltage:** The MQ-4 sensor requires a 5V power supply for the heating element, while the output pin can be connected to a microcontroller's analog or digital input.

- **Response Time:** The MQ-4 has a relatively fast response time, typically under 10 seconds, allowing it to react quickly to changes in methane levels.

- **Heating Element:** The sensor incorporates a small heating element that helps to stabilize its sensitivity. This element requires a warm-up time of about 20-30 seconds before taking accurate readings

Internal Working and Sensitivity Adjustment:

The MQ-4 sensor operates based on a metal-oxide semiconductor principle. Inside the sensor, the SnO_2 material interacts with gases in the air, changing its electrical resistance. Here’s a breakdown of how the sensor works:

- **Heating Element:** When the sensor is powered, the heating element heats up the SnO_2 layer. This process creates a conductive layer of oxygen ions on the SnO_2 surface.

- **Gas Interaction:** When methane gas molecules come into contact with the SnO_2 surface, they react with the oxygen ions. This reaction changes the concentration of free electrons, altering the material's resistance.

- **Resistance Variation:** The resistance of the SnO_2 layer decreases in the presence of methane, allowing more current to flow through the sensor.

- **Signal Conversion:** This change in resistance is translated into a varying voltage output by the sensor, which can then be read by a microcontroller as an analog signal.

To adjust the sensor's sensitivity, the module includes a potentiometer that controls the voltage comparator threshold.

Applications of the MQ-4 Methane Gas Sensor:

The MQ-4 sensor’s sensitivity to methane and ease of interfacing makes it ideal for various applications, particularly those related to safety and environmental monitoring:

- **Gas Leak Detection:** The MQ-4 is commonly used in systems that detect methane leaks, such as in gas pipelines, kitchens, and industrial sites. This helps to ensure timely detection and prevent potential hazards.

- **Home Automation:** In smart homes, the MQ-4 can be integrated into alarm systems to detect natural gas leaks from stoves or heating systems. This enhances safety by alerting users to gas leaks before they reach dangerous levels.

- **Environmental Monitoring:** The MQ-4 can be used in environmental monitoring stations to detect methane emissions in the atmosphere. Methane is a potent greenhouse gas, and tracking its levels is important for understanding environmental impact.

Calibration and Sensitivity Considerations:

The MQ-4 sensor’s accuracy can vary due to factors like temperature, humidity, and pressure. Calibration involves measuring the output voltage for known gas concentrations and creating a response curve, which allows for more precise readings in real-world conditions. The sensor may also require periodic recalibration to maintain its sensitivity and accuracy.

Advantages and Limitations of MQ-4:

Advantages:

- **Low Cost:** The MQ-4 is a cost-effective gas detection solution, making it accessible for hobbyists, researchers, and industry professionals.

- **Sensitivity:** It has a high sensitivity to methane, making it effective in detecting gas leaks and ensuring safety in areas with methane exposure.

- **Wide Detection Range:** With a range of 200 ppm to 10,000 ppm, the MQ-4 can detect methane concentrations at both low and high levels.

- Analog and Digital Output: The sensor module offers both analog and digital outputs, making it versatile and easy to integrate into a variety of projects.

Limitations:

- **Calibration Requirement**: The MQ-4 needs to be calibrated for accurate readings, and it may require periodic recalibration to maintain performance over time.

- **Temperature and Humidity Sensitivity:** The sensor’s performance can be affected by environmental factors like temperature and humidity, which may impact its accuracy.

- **Warm-Up Time:** The sensor requires a warm-up time of 20-30 seconds for optimal sensitivity, which may delay initial readings.

- **Limited Selectivity:** While the MQ-4 is designed for methane detection, it may also respond to other hydrocarbons, which could lead to false positives in certain environments.

- As methane monitoring becomes increasingly important due to environmental and safety concerns, the demand for reliable, sensitive gas sensors continues to grow. While the MQ-4 is a popular choice, there are more advanced alternatives like the MQ-6 (for LPG detection) and digital sensors that offer higher accuracy and better selectivity for specific gases.

- MQ-4 methane gas sensor is a practical and economical tool for detecting methane gas. Its wide detection range, ease of interfacing, and sensitivity make it a valuable component in gas leak detection systems, home automation, and environmental monitoring. While it has limitations in terms of calibration and environmental sensitivity, the MQ-4 provides a robust solution for basic gas detection needs and can be effectively integrated into projects with microcontrollers like Arduino. As methane detection technology continues to evolve, sensors like the MQ-4 will remain vital tools in both safety and environmental applications.

Figure 3: mQ-4 sensor

LCD DISPLAY:

A 16×2 LCD display is a widely used electronic display module that can display 16 characters per row on two lines. These displays are affordable, low-power, and easy to interface with microcontrollers, making them popular in embedded systems and electronics projects.

Key Features:

- **Characters:** Displays up to 32 characters (16 in each row).
- **Display Technology:** Typically uses Liquid Crystal Display(LCD) technology.
- **Interface:** Can communicate through 4-bit or 8-bit parallel interface.
- **Power** Low power consumption (typically 5V).
- **Backlight:** Often includes an LED backlight, making it readable in low light.
- **Customizable Characters**: Can create custom characters by programming 5x8 dot matrix pixels.

Pin Configuration:

- **VSS:** Ground.
- **VCC:** Power supply (usually 5V).
- **VO:** Controls contrast (typically connected to a potentiometer).
- **RS (Register Select):** Selects the command or data register.
- **RW (Read/Write):** Reads or writes data to/from the LCD.
- **EN (Enable):** Starts data read/write processes.
- **D0 to D7:** Data pins for sending data in 8-bit mode (or D4 to D7 in 4-bit mode).
- **LED+ and LED:** Pins for the backlight LED (optional).

Working Principle:

Text or characters are displayed by lighting up specific dots on the 5x8 dot matrix for each character space.The display is controlled through commands sent to its controller, typically the HD44780 or similar.

Interfacing with Microcontrollers:

- **4-bit Mode:** Uses only four data pins (D4–D7) plus control pins (RS, RW, and EN). Requires two data transactions to send one 8-bit character, but saves pins.

- **8-bit Mode:** Uses all eight data pins (D0–D7) for faster communication. Suitable when there are enough pins available.

LCD Commands:

- **Clear Display:** 0x01 - Clears all text.
- **Return Home:** 0x02 - Resets cursor to the first row, first column.
- **Entry Mode Set:** 0x06 - Configures cursor movement.
- **Display ON/OFF:** 0x0C - Turns the display on without cursor or blinking.
- **Cursor Display Shift:** 0x10 - Moves cursor and display.
- **Set CGRAM Address:** 0x40 - Sets address for custom characters.

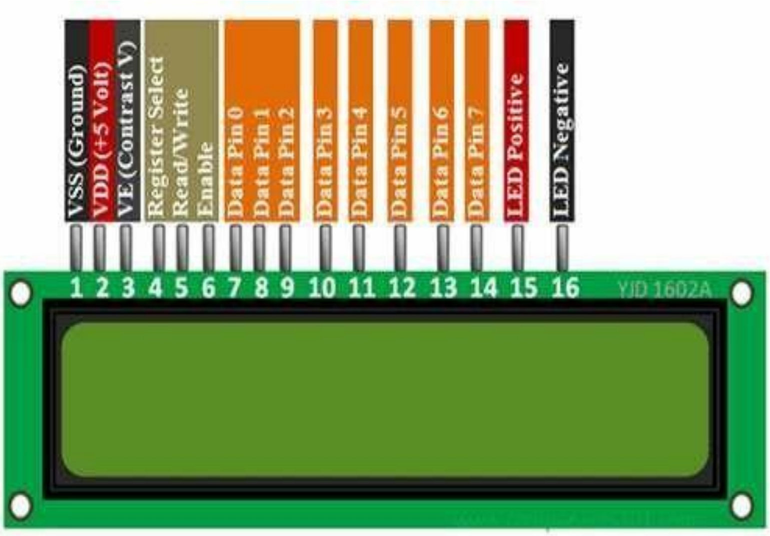
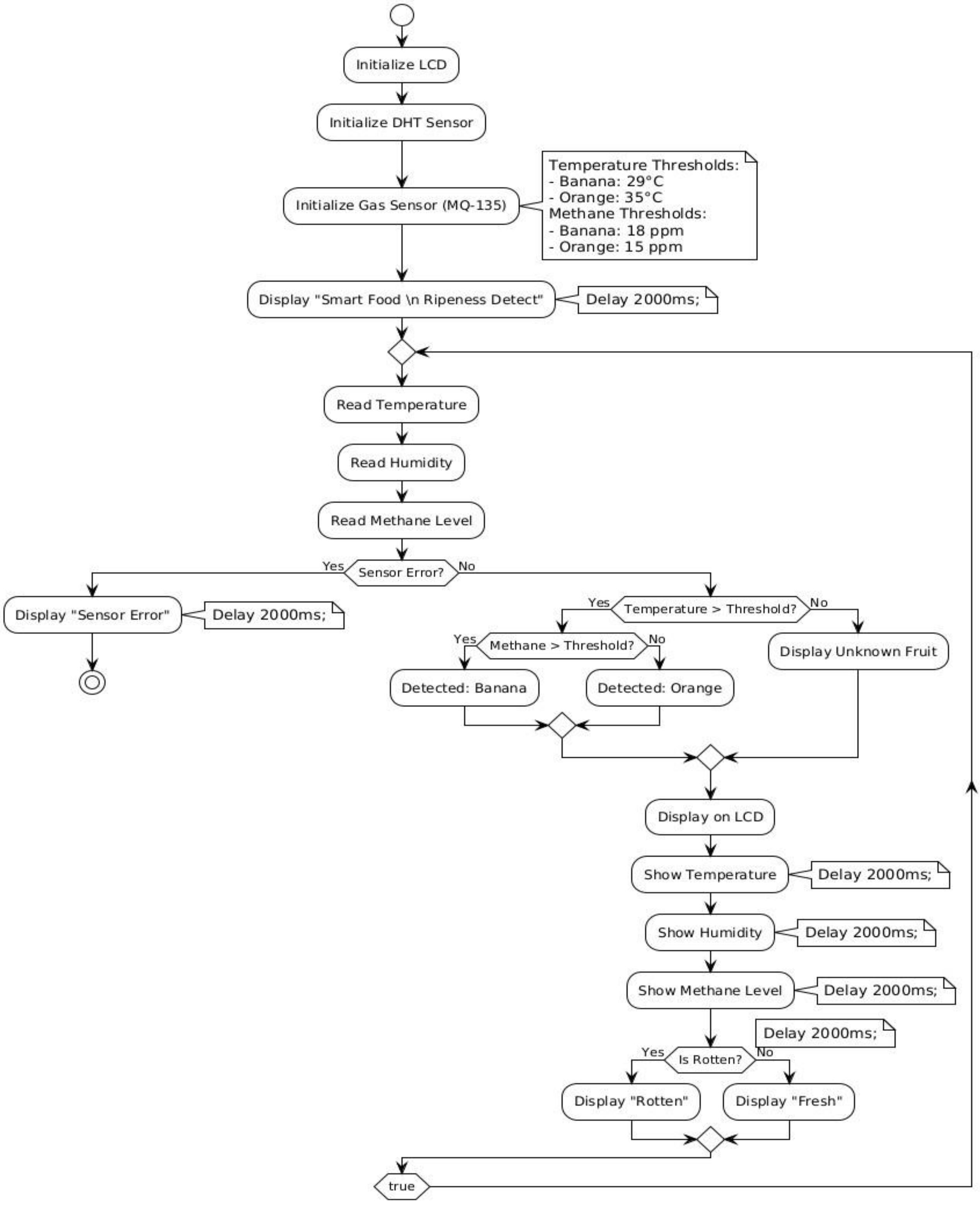


Figure4: lcd display

Applications:

Widely used in DIY electronics, Arduino projects, embedded systems, industrial machines, and information displays.

4. FLOWCHART:



4.1. Working:

A working of a smart food monitoring system helps in ensuring the freshness and quality of stored food by continuously tracking environmental conditions like temperature, humidity, and gas emissions. This system typically uses an Arduino UNO microcontroller, a DHT-11 sensor for temperature and humidity, an MQ-4 gas sensor for detecting gas emissions, and a 16×2 LCD display to show real-time readings. The purpose is to detect any signs of spoilage by observing changes in these conditions, providing alerts if necessary.

The DHT-11 sensor measures temperature and humidity, which are crucial for food preservation. Food items, especially perishables, are highly sensitive to temperature fluctuations and high humidity, as these can speed up microbial growth and spoilage. The sensor reads these values and sends the data to the Arduino, which then processes the information to determine if conditions are within safe levels for food storage.

Alongside temperature and humidity, the MQ-4 gas sensor plays a key role in spoilage detection. As food begins to spoil, it releases certain gases, including methane, which the MQ-4 sensor can detect. The sensor’s analogy output is read by the Arduino, and if the detected gas concentration exceeds a set threshold, the system interprets this as a sign of potential spoilage. This helps in identifying food items that may be going bad before it becomes visually noticeable.

All the data collected from the sensors are displayed on the LCD screen in real-time, providing easy access to temperature, humidity, and gas levels. The Arduino updates the display periodically, allowing the user to monitor food storage conditions at a glance. If gas levels surpass the predefined threshold, the system can also trigger a warning signal, like a buzzer or flashing LED, to alert the user to check the food items.

This smart food monitoring system is highly versatile and can be used in various settings, including home kitchens, grocery stores, and food storage facilities. By continuously monitoring environmental factors, it helps reduce food waste by ensuring optimal storage conditions and early detection of spoilage.

4.2. Software Discription:

Proramming:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD I2C address might vary

// DHT setup
#define DHTPIN 2    // DHT11 connected to pin D2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Gas sensor setup
#define gasSensorPin A0  // MQ-135 connected to A0

// Thresholds for ripeness detection for each fruit
float thresholdMethaneBanana = 18.0;     // Methane threshold for banana ripeness
float thresholdMethaneOrange = 15.0;     // Methane threshold for orange ripeness
float thresholdMethaneApple = 10.0;      // Methane threshold for apple ripeness
float thresholdMethaneMango = 20.0;      // Methane threshold for mango ripeness
float thresholdMethaneGrapes = 12.0;     // Methane threshold for grapes ripeness
float thresholdMethanePineapple = 22.0;  // Methane threshold for pineapple ripeness
float thresholdMethaneStrawberry = 8.0;  // Methane threshold for strawberry ripeness

float temperatureThresholdBanana = 29.0;    // Temperature threshold for banana ripeness (in °C)
float temperatureThresholdOrange = 35.0;    // Temperature threshold for orange ripeness (in °C)
float temperatureThresholdApple = 25.0;     // Temperature threshold for apple ripeness (in °C)
float temperatureThresholdMango = 30.0;     // Temperature threshold for mango ripeness (in °C)
float temperatureThresholdGrapes = 24.0;    // Temperature threshold for grapes ripeness (in °C)
float temperatureThresholdPineapple = 32.0; // Temperature threshold for pineapple ripeness (in
°C)
float temperatureThresholdStrawberry = 20.0;// Temperature threshold for strawberry ripeness (in
°C)

void setup() {
  lcd.init();
  lcd.backlight();
  dht.begin();
  Serial.begin(9600);
  lcd.setCursor(0, 0);
```

```
  lcd.setCursor(0, 1);


  lcd.print("Ripeness Detect");
  delay(2000);
  lcd.clear();
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  int gasValueRaw = analogRead(gasSensorPin);
  float methaneConcentration = calculateMethane(gasValueRaw);

  if (isnan(temperature) || isnan(humidity)) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sensor Error");
    delay(2000);
    return;
  }

  String fruitType = detectFruitType(temperature, humidity);
  bool isRotten = checkRipeness(fruitType, methaneConcentration, temperature);

  displayOnLCD(fruitType, temperature, humidity, methaneConcentration, isRotten);

  delay(5000);
}

// Function to calculate methane concentration
float calculateMethane(int gasValue) {
  return (gasValue / 1024.0) * 100;
}

// Function to detect fruit type based on temperature and humidity
String detectFruitType(float temp, float hum) {
  if (temp > 29 && hum > 65) {
    return "Banana";
```

```
 } else if (temp <= 32 && hum < 65) {
    return "Orange";
  } else if (temp < 25 && hum >= 50 && hum < 80) {
    return "Apple";
  } else if (temp > 28 && hum >= 60 && hum < 85) {
    return "Mango";
  } else if (temp < 24 && hum > 55 && hum < 80) {
    return "Grapes";

  } else if (temp > 30 && hum > 70 && hum < 85) {
    return "Pineapple";
  } else if (temp < 22 && hum > 65 && hum < 85) {
    return "Strawberry";
  }
  return "Unknown";
}

// Function to check ripeness based on thresholds for each fruit
bool checkRipeness(String fruit, float methane, float temp) {
if (fruit == "Banana") {
    return (methane > thresholdMethaneBanana || temp > temperatureThresholdBanana);
  } else if (fruit == "Orange") {
    return (methane > thresholdMethaneOrange || temp > temperatureThresholdOrange);
  } else if (fruit == "Apple") {
    return (methane > thresholdMethaneApple || temp > temperatureThresholdApple);
  } else if (fruit == "Mango") {
    return (methane > thresholdMethaneMango || temp > temperatureThresholdMango);
  } else if (fruit == "Grapes") {
    return (methane > thresholdMethaneGrapes || temp > temperatureThresholdGrapes);
  } else if (fruit == "Pineapple") {
    return (methane > thresholdMethanePineapple || temp > temperatureThresholdPineapple);
  } else if (fruit == "Strawberry") {
    return (methane > thresholdMethaneStrawberry || temp > temperatureThresholdStrawberry);
  }
  return false;
}
```

```
// Function to display data on LCD with detailed information for each fruit
void displayOnLCD(String fruit, float temp, float hum, float methane, bool rotten) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(fruit);
  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.print(temp, 2);
  lcd.print("C");
  delay(2000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(fruit);
  lcd.setCursor(0, 1);
  lcd.print("Hum: ");
  lcd.print(hum, 2);
  lcd.print("%");
  delay(2000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(fruit);
  lcd.setCursor(0, 1);
  lcd.print("Methane: ");
  lcd.print(methane, 2);
  lcd.print(" ppm");
  delay(2000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(fruit);
  lcd.setCursor(0, 1);
  lcd.print(rotten ? "Rotten" : "Fresh");
  delay(2000);
```

4.3. APPLICATIONS:

A Smart Food Monitoring System, incorporating sensors like gas (methane), temperature, and humidity sensors alongside a microcontroller (e.g., Arduino), has diverse applications across multiple sectors. Here, I'll outline key applications, detailing how this technology can benefit industries and consumers by promoting freshness, reducing waste, and enhancing food safety.

Quality Control in Food Supply Chains:

A primary application of a smart food monitoring system is in food supply chains, where it can monitor the quality of perishable items during transport and storage. Fresh produce, meat, and dairy are sensitive to changes in environmental conditions, and continuous monitoring of factors like temperature, humidity, and gas levels (e.g., methane for ripeness or spoilage) allows supply chain managers to detect spoilage risks early.

Furthermore, this technology can be integrated with tracking systems, allowing real-time quality monitoring and automated logging for compliance purposes. This not only improves product quality at the point of sale but also increases consumer trust, as end customers receive food that has been consistently monitored from farm to fork.

Supermarkets and Grocery Stores:

Supermarkets and grocery stores can use smart food monitoring systems to improve inventory management and quality control in real-time. By monitoring fruit ripeness (for example, bananas, oranges, or mangoes), store managers can strategically place ripe items in promotional areas, sell items nearing spoilage at a discount, and prevent waste by pulling spoiled products off the shelves promptly.

The system can also help with forecasting and restocking strategies. By analyzing environmental data from the sensors, stores can predict when certain produce is likely to ripen or spoil, allowing for better inventory decisions, reducing spoilage costs, and ensuring that fresh produce is always available to customers.

Restaurants and Catering Services:

In the restaurant and catering sectors, food quality and safety are critical. A smart food monitoring system can be implemented to oversee the freshness and condition of ingredients used in kitchens, particularly for high-value or perishable items like seafood, meats, and exotic fruits. The system can alert kitchen staff to ingredients approaching spoilage, allowing them to prioritize their use and reduce the risk of serving spoiled food, which is essential for maintaining food safety standards and customer satisfaction.

Additionally, restaurants can use the data collected by these systems to track usage patterns and optimize their purchasing strategies. This proactive approach ensures that chefs have access to fresh ingredients daily and helps minimize food waste, thereby reducing costs and promoting sustainability.

Home Food Storage and Management:

With increasing awareness of food waste, a smart food monitoring system can be adapted for home use, allowing families to track the condition of food stored in their refrigerators or pantries. By integrating the system with a mobile app, users could receive notifications about the freshness or ripeness of stored produce and even recommendations on when to use certain ingredients to prevent spoilage.

This application is especially beneficial for busy households or individuals who buy in bulk. Not only does it help in managing food freshness, but it also aids in meal planning, reducing the likelihood of forgotten or spoiled ingredients and encouraging a sustainable, waste-free lifestyle.

Cold Storage and Warehousing:

In cold storage facilities and warehouses, where bulk quantities of food items are stored for extended periods, a smart monitoring system is invaluable. These facilities typically store perishable goods under controlled conditions, but any fluctuation in temperature, humidity, or gas levels can compromise product quality. By continuously monitoring these parameters, smart systems can quickly alert facility managers to potential issues, ensuring that corrective actions are taken to maintain the desired storage environment.

5. RESULT:

The implementation of a Smart Food Monitoring System these results demonstrates how effective monitoring can positively impact the freshness of produce, reduce waste, and ensure food safety in diverse environments.

First, in the food supply chain, real-time monitoring of environmental conditions has significantly reduced spoilage rates. By providing continuous feedback on temperature, humidity, and gas levels during transportation and storage, this system allows for quick corrective actions when conditions deviate from optimal ranges. For example, monitoring the ripeness of bananas, mangoes, and other sensitive produce helps prevent over-ripening, ensuring fresher products upon arrival and enhancing customer satisfaction.

In supermarkets and grocery stores, the system has improved inventory management and freshness of displayed produce. By identifying items nearing ripeness or spoilage, store managers can promote or discount those items strategically, minimizing waste. The technology has proven effective in helping supermarkets make informed decisions on inventory rotation, thereby ensuring that fresh items are always available for customers and reducing food loss significantly.

For restaurants and catering services, this system has enhanced food safety by preventing the use of ingredients that may have deteriorated. Freshness alerts help chefs prioritize perishable ingredients and prevent spoilage from entering food preparations.

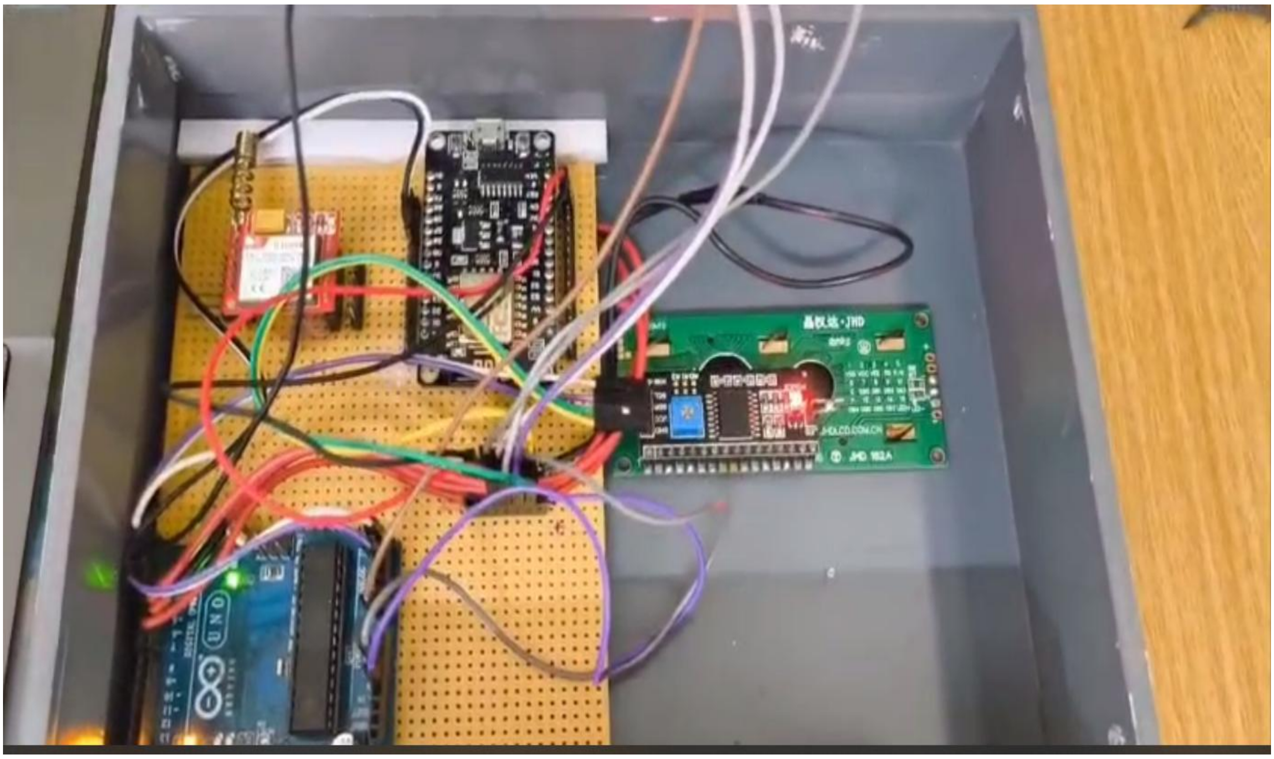
Overall, the Smart Food Monitoring System has shown excellent results in ensuring food quality and safety across various sectors. By enabling real-time monitoring and early detection of spoilage, the system has promoted efficient food management, reduced waste, and offered economic and environmental benefits.

Let’s take a look at some images of our **Smart Food Monitoring Device** to better understand its components and setup.

Or simply:Now, let’s see the images of our Smart Food Monitoring Device in action.


us: Fresh & Safe


Blynk.Console
My organization - 8207GE
Dashboards
Developer Zone
Devices
Automations
Users
Organizations
Locations
Demand Response
Fleet Management
In-App Messaging
Food Monitoring Device
Online
sukin
My organization - 8207GE
Edit
Live
1h
6h
1d
1w
1mo
3mo
6mo
1y
Temperature
34.2
0
100
Humidity
67
0
100
Methane Gas
31.2
0
100
Fresh & safe
Rotten
Food Monitoring Device
Temperature
Spoiled
Humidity
Fresh & Safe
methane



6. CONCLUSION:

In conclusion, a Smart Food Monitoring System presents a practical and impactful solution for enhancing food quality, safety, and sustainability across various sectors. By integrating sensors that monitor key environmental factors like temperature, humidity, and gas emissions, the system effectively tracks food freshness and ripeness in real time. This enables timely interventions to prevent spoilage and prolongs the shelf life of perishable items, benefiting consumers, businesses, and the entire food supply chain.

The applications of this system in supermarkets, supply chains, restaurants, and even households demonstrate their versatility and value. For businesses, it reduces waste and operational costs by helping managers make informed decisions on food inventory. In households, it supports more efficient food management, reducing unnecessary food waste and encouraging sustainable practices. These benefits contribute to reducing food loss on a larger scale, which is essential for addressing environmental and economic challenges

As technology continues to advance, smart monitoring systems for food are likely to become even more sophisticated and accessible. Integrating them with IoT platforms and data analytics can further optimize food management, offering insights that improve decision-making and streamline processes across the food industry. The Smart Food Monitoring System thus holds significant promise for creating a more efficient, safe, and sustainable food system that benefits everyone involved.

7. REFERENCES

1.Arjun Krishna et.al, 2019. Food Monitoring and Control System Using Internet of Things. International Journal for Research in AppliedScience & Engineering Technology. 7(4): 01- 05

2.Arun Kumar, 2020. An Arduino Sensor-Based Approach for Detecting the Food Spoilage.International Journal of Engineering Applied Science and Technology. 5(3): 596-599.

3.Prabha Sundaravadivel et.al, 2019. A Deep Learning-based Automated Nutrition Monitoring System in the IoT. Institute of Electricaland Electronics Engineering Xplore. 20(3): 1-9.

4.Ravi Chander et.al, 2020. Food Quality Monitoring System by Using Arduino. Journal of Engineering Science. 11(4): 625-630.

5..Shivani Bhandari et.al, 2020. IoT Food Monitoring System in Warehouses. International Research Journal of Engineering andTechnology. 7(4): 5957-5960.

6..Sujihelen et.al, 2020. Automated Nutrition Monitoring System in IoT. Journal of Critical Reviews.7(19): 5071-5075.

7.Venkatesh et.al, 2017. A Food Monitoring System Based on Bluetooth Low Energy and IoT.International Journal of Engineering Research and Application. 7(3): 30-34.

8.Yamine Bouzembrek et.al, 2019. IoT in Food Safety of Food Science and Technology. Elsevie