Implementing Humidity Sensors in the Food Industry

Submitted in Partial Fulfillment of the Requirements for the

Naan Mudhalvan Mandatory Courses

Industrial IoT and Industry 4.0

Submitted by

DEEPAK RAJ S	513121106301
MANIMARAN V	513121106302
PREM KUMAR S	513121106306

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY.

VELLORE – 632 002

APRIL-2024.

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY

VELLORE-632 002.

BONAFIDE CERTIFICATE

Certified that this project report "Implementing Humidity Sensors in the Food Industry" is bonafide work of **DEEPAK RAJ S(513121106301), MANIMARAN V (513121106302), PREM KUMAR S(513121106306)** who carried out the work under my supervision.

SIGNATURE

Dr.S.LETITIA,M.E.,Ph.D.,
PROFESSOR,
HEAD OF THE DEPARTMENT
Department of ECE,
THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY-VELLORE.

SIGNATURE

Prof.S.SABARI RAJAN,M.E.,
ASSISTANT PROFESSOR ,
Department of ECE,
THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY-VELLORE.

Project work Viva-Voce Examination held on _______________

INTERNAL EXAMINER

EXTERNAL EXAMINER

OBJECTIVE:

The objective of the provided Arduino program and related content is to implement a basic system for monitoring humidity levels using an Arduino microcontroller and a humidity sensor. The program aims to achieve the following objectives:

1. **Humidity Monitoring:**

   Continuously read humidity data from the sensor to monitor environmental humidity levels.

2. **Threshold Alerting:**

   Set a predefined threshold for humidity levels, triggering an alert if the measured humidity exceeds this threshold. This feature is crucial for maintaining optimal humidity conditions in food processing environments to ensure product quality and safety.

3. **Data Display:**

   Display real-time humidity readings on the Arduino Serial Monitor for immediate observation and analysis.

4. **Sensor Integration:**

   Demonstrate the integration of a humidity sensor with Arduino microcontroller hardware, illustrating the practical implementation of sensor technology in electronic systems.

5. **Baseline Implementation:**

   Serve as a foundational example for more advanced applications, such as real-time monitoring systems, automated control mechanisms, and integration with broader IoT (Internet of Things) networks in food processing facilities.

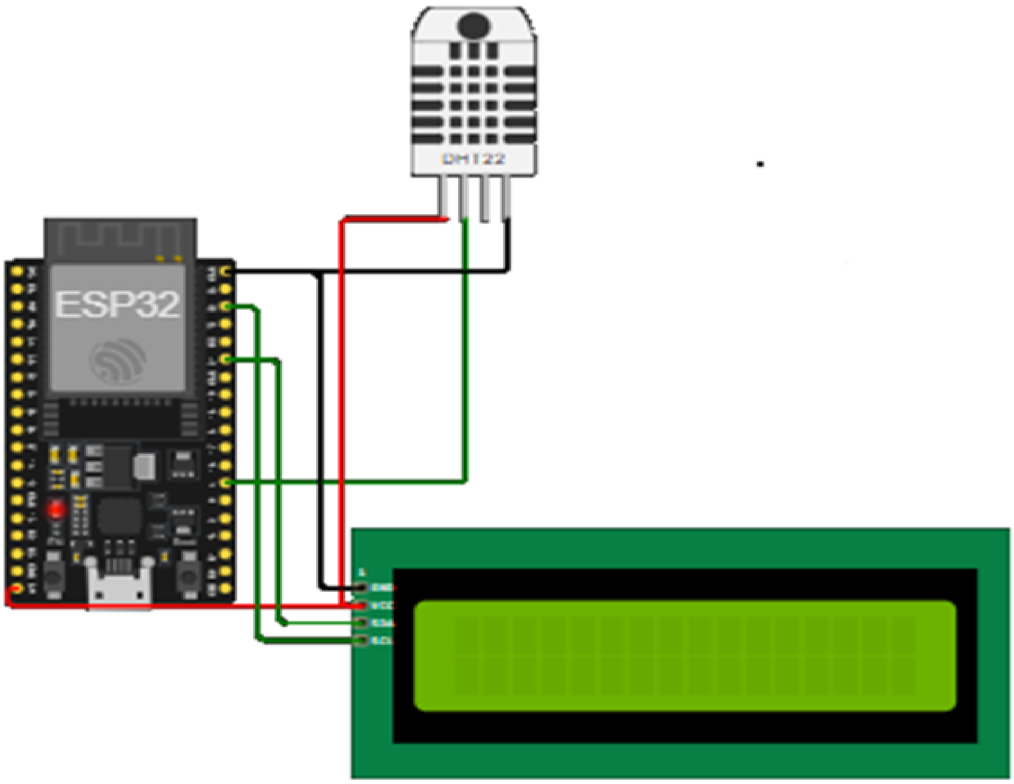
INTRODUCTION:

Humidity control is paramount in the food industry to ensure product quality, safety, and compliance with industry regulations. Maintaining optimal humidity levels throughout food processing operations such as baking, drying, and fermentation is critical for preventing microbial growth, preserving shelf life, and enhancing the overall quality of food products. To achieve this, integrating humidity sensors into food processing equipment and implementing real-time monitoring systems are essential steps.

In this context, this project aims to address the multifaceted challenges of humidity management in the food industry using Arduino microcontroller technology and humidity sensors. The project encompasses tasks ranging from sensor selection and calibration to integration with food processing equipment and real-time monitoring implementation. By leveraging Arduino's versatility and simplicity, along with appropriate humidity sensors, the project endeavors to provide a robust solution for humidity control in food processing environments.

Through the following sections, we will delve into the detailed tasks involved in this project, outlining strategies for sensor selection, calibration planning, integration with processing equipment, real-time monitoring system development, and maintenance considerations. By systematically addressing these tasks, we aim to offer a comprehensive approach to humidity management tailored to the specific requirements and challenges of the food industry. Ultimately, this project aims to contribute to enhanced food safety, quality assurance, and regulatory compliance in food processing operations.

CIRCUIT DAIGRAM:



COMPONENTS REQUIRED:

1)ESP8266

2)BREAD BOARD

3)LCD DISPLAY (I2C)

5)DHT(HUMIDITY SENSOR)

6)JUMPER WIRES

SOURCE CODE:

```
#include <WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include<BlynkSimpleEsp32.h>
BlynkTimer timer;
char auth[]=’”;
const char* ssid = "wokwi-GUST";
const char* password = "";
const char* host = "YOUR_SERVER_IP_OR_DOMAIN";
const int port = 80;
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  Serial.begin(115200);
  delay(100);
  dht.begin();
  lcd.init();
  lcd.backlight();
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  delay(1000);
}
void loop() {
  float humidity = dht.readHumidity();
  if (isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println("%");
  displayHumidity(humidity);
  sendData(humidity);

  delay(2000);
}
void displayHumidity(float humidity) {
  lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print("%");
}
void sendData(float humidity) {
```

```
 WiFiClient client;
 if (!client.connect(host, port)) {
   Serial.println("Connection failed");
   return;
 }
 String url = "/update?humidity=" + String(humidity);
 Serial.print("Requesting URL: ");
 Serial.println(url);
 client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
 delay(10);
 while (client.available()) {
   String line = client.readStringUntil('\r');
   Serial.print(line);
 }
 Serial.println();
 Serial.println("Request complete");
}
```

HARDWARE COMPONENTS DESCRIPTION:

1. **Microcontroller:**

The Arduino board serves as the brain of the project, providing the necessary processing power, input/output pins, and communication interfaces. It executes the program logic for reading humidity data from the sensor, processing it, and triggering alerts if necessary. Arduino boards come in various models, such as Arduino Uno, Arduino Mega, or Arduino Nano, offering different capabilities and form factors to suit the project requirements.

2. **Humidity Sensor:**

The humidity sensor is a crucial component responsible for measuring the relative humidity in the environment. It detects moisture levels and converts them into electrical signals that can be interpreted by the Arduino board. Common types of humidity sensors include capacitive humidity sensors, resistive humidity sensors, and integrated digital sensors like the DHT series (e.g., DHT11, DHT22). Select a sensor that meets the requirements for accuracy, reliability, and compatibility with food processing environments.

3. **Breadboard and Jumper Wires:**

The breadboard provides a convenient platform for prototyping and connecting various electronic components, including the Arduino board and sensors. Jumper wires are used to establish electrical connections between different components on the breadboard, allowing for easy experimentation and troubleshooting without soldering.

4. **LCD DISPLAY(I2C):**

LCD displays with I2C communication typically have only four lines: VCC, GND, SDA, and SCL. VCC is for power, GND is ground, SDA is the data line, and SCL is the clock line. These displays are widely used in embedded systems where minimizing the number of pins used for communication is important.

5. **Power Supply:**

The power supply provides the necessary electrical power to operate the Esp board and connected components. Depending on the Esp model and power requirements of the project, power can be supplied via a USB connection from a computer or an external power adapter connected to the Esp's power jack or Vin pin.

6. **Computer with Arduino IDE:**

The Arduino Integrated Development Environment (IDE) is software used for writing, compiling, and uploading program code to the Arduino board. It provides a user-friendly interface for programming Arduino projects and debugging code. Ensure access to a computer with the Arduino IDE installed for programming and testing the project.

WORKING :

1. **Initialization:**

The Arduino program begins by initializing the necessary components, including the humidity sensor, serial communication, and any optional components such as LEDs.

2. **Read Humidity Data:**

The Arduino board reads humidity data from the connected humidity sensor at regular intervals. The specific method for reading data depends on the type of sensor used and may involve calling appropriate functions provided by sensor libraries.

3. **Data Processing:**

Once the humidity data is obtained, the Arduino board processes it as needed. This may include converting raw sensor readings into meaningful humidity values, performing calculations, or comparing the measured humidity against predefined thresholds.

4. **Threshold Che**cking:

If the project requires monitoring humidity levels and triggering alerts when certain thresholds are exceeded, the Arduino program compares the measured humidity against predefined threshold values. If the humidity exceeds a threshold, the program can trigger actions such as activating LEDs for visual alerts or sending notifications through external communication modules.

5. **Alert Mechanisms (Optional):**

In projects where alert mechanisms are implemented, the Arduino program activates visual or audible indicators to alert users or trigger actions when specific conditions are met. For example, if the humidity exceeds a predefined threshold, the program can turn on an LED to indicate high humidity levels.

6. **Continuous Operation:**

The Arduino program continues to loop through these steps, continuously monitoring humidity levels, processing data, and triggering alerts as necessary. This loop ensures that the system operates in real-time and responds promptly to changes in humidity conditions.

7. **Integration with Control Systems (Optional):**

In more advanced applications, the Arduino system may be integrated with control systems or actuators to automate processes based on humidity readings. For example, the Arduino board can control humidifiers or ventilation systems to maintain optimal humidity levels in a controlled environment.

8. **Maintenance and Monitoring:**

Throughout operation, it's essential to monitor the system for any issues or abnormalities. Regular maintenance, including calibration checks and sensor cleaning, helps ensure the system's accuracy and reliability over time.

APPLICATIONS:

1. **Food Processing Industry:**

Humidity monitoring and control play a crucial role in various food processing operations, including baking, drying, fermentation, and storage. Arduino-based humidity monitoring systems can help maintain optimal humidity levels to ensure product quality, shelf life, and compliance with food safety regulations.

2. **Greenhouses and Agriculture:**

Arduino-based humidity sensors can be used in greenhouses and agricultural settings to monitor and control humidity levels for optimal plant growth and crop yield. Maintaining proper humidity conditions helps prevent diseases, improve plant health, and increase agricultural productivity.

3. **HVAC Systems:**

In heating, ventilation, and air conditioning (HVAC) systems, humidity sensors integrated with Arduino boards enable precise control of indoor humidity levels. This ensures occupant comfort, prevents mold growth, and protects building materials from damage due to excess moisture.

4. **Home Automation:**

Arduino-based humidity sensors can be incorporated into home automation systems to monitor indoor humidity levels and adjust HVAC settings accordingly. This helps maintain a comfortable indoor environment while conserving energy and reducing utility costs.

5. **Museum and Archive Preservation:**

Humidity monitoring is critical for preserving artifacts, documents, and historical materials in museums and archives. Arduino-based humidity sensors can provide real-time monitoring of humidity levels in storage facilities to prevent deterioration and damage to valuable collections.

6. **Medical and Healthcare:**

In medical and healthcare facilities, Arduino-based humidity sensors can be used to monitor humidity levels in patient rooms, laboratories, and storage areas. Maintaining proper humidity conditions helps prevent the growth of bacteria and viruses, ensuring a safe and hygienic environment for patients and healthcare workers.

7. **Electronics Manufacturing:**

Humidity control is essential in electronics manufacturing processes to prevent moisture-related damage to components and assemblies. Arduino-based humidity sensors can be integrated into manufacturing equipment to monitor humidity levels in cleanrooms and production areas, ensuring product quality and reliability.

8. **Weather Monitoring Stations:**

Arduino-based weather monitoring stations can include humidity sensors to measure relative humidity levels in the atmosphere. This data, along with other weather parameters, can be used for weather forecasting, climate research, and environmental monitoring applications.

9. **Laboratories and Research Facilities:**

IoT-based humidity sensors are valuable tools in laboratories and research facilities for monitoring humidity levels in controlled environments, incubators, and chambers. Precise humidity control is essential for experiments, sample storage, and maintaining the integrity of research materials.

CONCLUSION:

The implementation of humidity sensors tailored for the food industry is crucial for ensuring product quality, safety, and regulatory compliance throughout food processing operations. By integrating humidity sensors with Arduino microcontrollers, we can create robust monitoring and control systems specifically designed to meet the unique requirements of food processing environments.

Through this project, we have explored the significance of humidity control in the food industry and outlined the key components, working principles, and applications of Arduino-based humidity monitoring systems. By selecting sensors that comply with industry regulations and standards, we can mitigate risks associated with microbial contamination, moisture-related spoilage, and product degradation.

IoT-based humidity monitoring systems offer real-time insights into humidity levels during various food processing stages such as baking, drying, and fermentation. By continuously monitoring humidity levels and implementing threshold alerts, these systems enable proactive interventions to maintain optimal conditions and prevent quality issues or safety hazards.