ENSURING FOOD QUALITY BASED ON IoT

S.Syedakbar[1],S.SankaraNarayanan[2],B.Seralathan[2],F.Allwin Raj[2]
[1]Assistant Professor, Department of Electronics and Communication Engineering,
K Ramakrishnan College of Technology, Trichy-621112
Mail I'd: syedakbar.s@krct.ac.in, sankara7134@gmail.com, seramaz7@gmail.com, allwinraj10071007@gmail.com

Abstract
The objective of this project is to create a food quality sensor system utilizing the Arduino platform and a DHT-11 temperature and humidity sensor and the MQ-3 alcohol vapour detection sensor . The system provides a cost-effective solution for monitoring the freshness and safety of perishable food items. The DHT-11 sensor is employed to measure the temperature and humidity levels inside a food storage container. By monitoring these parameters, the system can determine whether the storage conditions are suitable for maintaining the quality of the food. If the temperature or humidity deviates from the desired range, the system can trigger an alert to notify the user. The MQ-3 sensor is utilized to detect alcohol vapour, which can be an indicator of food spoilage or contamination. When certain foods undergo decomposition, they release volatile compounds such as ethanol. The system can now detect the presence of alcohol vapours and generate an alert if detected beyond a predetermined threshold thanks to the integration of the MQ-3 sensor. The central processing unit for gathering data from the sensors and carrying out the required algorithms is the Arduino microcontroller. It interfaces with the DHT-11 and MQ-3 sensors to retrieve sensor readings. These readings are then processed to determine the food quality status. The Arduino can be programmed to send notifications via SMS, email, or other communication methods to inform the user about any potential issues. Overall, this food quality sensor system provides a cost-effective and easily implementable solution for monitoring the freshness and safety of food items. It can be utilized in various settings,

including homes, restaurants, or food storage facilities, to ensure that the stored food remains within optimal conditions and to prevent the consumption of spoiled or contaminated items

Keywords: DTH-11, MQ-3, ALCOHOL VAPOUR

INTRODUCTION

Food quality is a crucial aspect of our daily lives, as the consumption of contaminated or spoiled food can lead to health issues. To ensure food safety, the use of food quality sensors has become increasingly important. In this context, utilizing Arduino, DHT11, and MQ-3 sensors can provide an effective solution for monitoring and maintaining the quality of food. Arduino is a popular open-source electronics platform for interactive projects and prototyping. It is made up of a board for a microcontroller and a programming environment that enables users to create and upload code to the board. Arduino boards are versatile and can be easily integrated with various sensors to perform specific tasks. A low-cost digital temperature and humidity sensor with precise measurements is the DHT11 sensor. It has a 2 degree accuracy range for measuring ambient temperature, and a 5% accuracy range for measuring humidity, which ranges from 20% to 90%. These features make a DHT11 sensor suitable for monitoring the environmental conditions affecting food quality. The MQ-3 sensor, on the other hand, is a gas sensor that is primarily used for detecting alcohol vapor. It can detect alcohol concentrations in the range of 0.05 to 10 mg/L. Since the presence of alcohol vapors can indicate food spoilage or contamination, the MQ-3 sensor can be employed to determine the freshness and quality of certain food products.

Fig 1 : Food with humidity and temperature

We can create a food quality monitoring system by integrating the Arduino board, DHT11 sensor, and MQ-3 sensor. The DHT11 sensor detects alcohol vapours while the MQ-3 sensor monitors temperature and humidity. These sensor readings can be processed by the Arduino board, which can then trigger appropriate actions or provide feedback based on predefined thresholds. For example, if the temperature exceeds a certain limit or if alcohol vapours are detected above acceptable levels, the system can generate an alert or activate a cooling mechanism to preserve food. Additionally, patterns can be found, storage conditions can be improved, and possible problems with food quality can be predicted using the sensor data that has been gathered. In conclusion, using an Arduino board, DHT11 sensor, and MQ-3 sensor helps speed up the creation of a system for measuring the quality of food.To preserve the freshness and safety of food goods, this system can monitor temperature, humidity, and alcohol vapour levels. We may take proactive efforts to avoid food contamination and maintain high-quality standards by employing these sensors. The proposed effort entails creating a food quality sensor by combining the DTH11 and MQ-3 sensors with an Arduino microcontroller.

The goal is to create a device that can assess the freshness and safety of food products based on various parameters. Hardware setup is Connect the DTH11 sensor and MQ-3 sensor to the Arduino R3 board appropriately. The DTH11 sensor will monitor temperature and humidity, while the MQ-3 sensor will detect alcohol vapor, which might indicate food rotting. Data collection and calibration: To achieve accurate measurements, calibrate the sensors. Create a program that reads data from the sensors at regular intervals using the Arduino programming language. This software should cater for the unique needs of each sensor and guarantee effective data collecting. Data processing and analysis is Use the acquired sensor data to assess the quality of the food. For example, the temperature and humidity readings from the DTH-11 sensor can be compared against predefined thresholds to determine if the food is stored under suitable conditions. The MQ-3 sensor data can be analyzed to identify the presence of alcohol vapours, indicating potential spoilage. Alert system implements a mechanism to alert the user if the food quality falls below acceptable levels. This can be achieved through visual indicators (such as LEDs), audible alarms, or even notifications sent to a connected device. UI is Design and develop a user interface to display the sensor readings and status. This can be a simple LCD display connected to the Arduino or a more advanced graphical interface using additional components like an OLED screen or a computer interface. Prototyping and testing is Build a functional prototype of the food quality sensor system. Test the device with various food samples under different storage conditions to evaluate its effectiveness and reliability.

Fig2: VAPOUR OF FOOD

Make necessary adjustments to the hardware or software as required. Documentation and reporting is Document the entire development process, including the circuit diagram, wiring details, and program code. Prepare a report summarizing the findings, limitations, and potential improvements of the food quality sensor system. It's important to keep in mind that the project's scope can be increased depending on the demands. For a more thorough investigation of food quality, more sensors might be added to assess variables like pH or certain gas concentrations. You may use the MQ3 sensor and DHT 11 sensor with an Arduino Uno R3 and the following methods to construct a food quality sensor. Designing systems is Define your food quality sensor's capabilities and general system requirements. Determine what parameters you want to measure, such as alcohol concentration (MQ3 sensor) and temperature/humidity (DHT 11 sensor). Arduino Programming is Write the code for your Arduino board to read data from the sensors and process it. This may be accomplished with the Arduino IDE or any other programming environment that is compatible. UI is a simple example to get you started. You may develop a User Interface to show sensor readings or offer an output to the user if needed. This can be accomplished with the use of an LCD display, LEDs, or even a graphical interface on a computer or mobile device. Testing and optimization are essential. To guarantee accuracy and dependability, test your food quality sensor system with a variety of food samples and scenarios. Make any required changes to the hardware or software to improve performance, For additional details on the sensors and their use with Arduino, go to the sensor datasheets, Arduino documentation, and appropriate references. In addition, feel free to modify and build on this process to meet the needs of your specific project.

METHODOLOGY

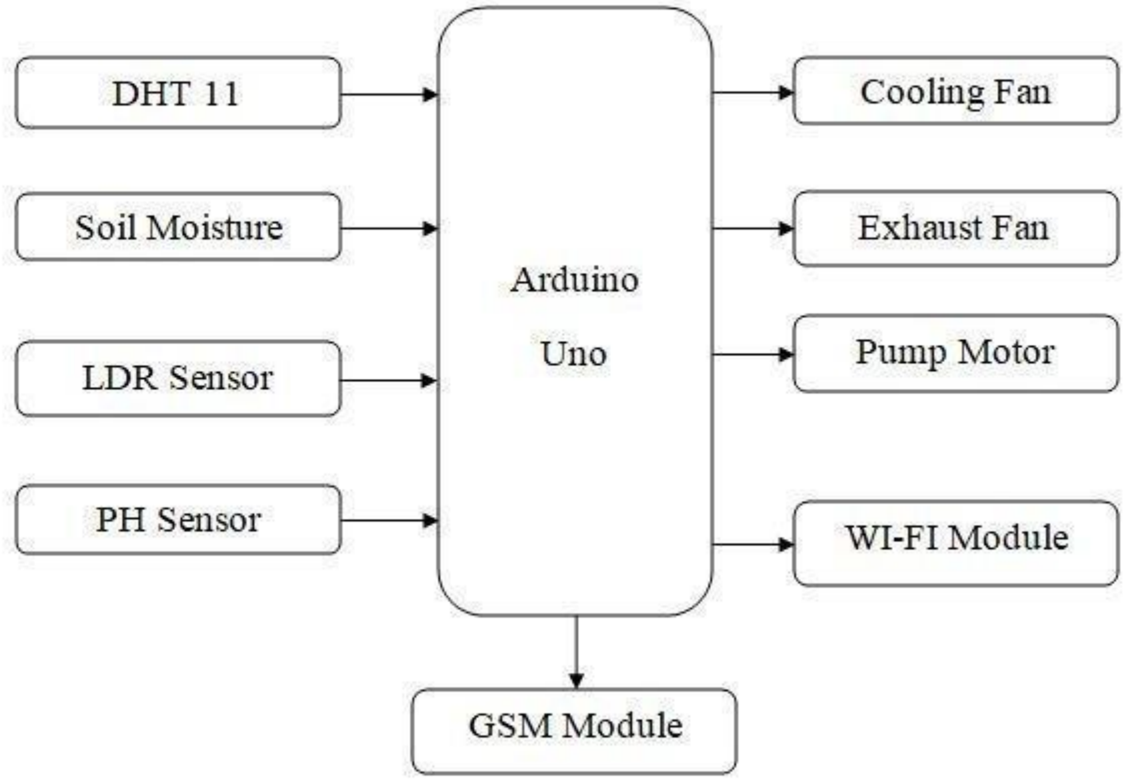

Fig 3: Block diagram of food quality sensor using arduino

The Arduino UNO is a microcontroller board that is based on the ATmega328. It is one of the most extensively used prototype boards. The board already includes an Arduino boot loader. It has 6 PWM pins, 6 Analog inputs, 14 GPIO pins, an on-board resonator, a reset button, pin header mounting holes, and on-board UART, SPI, and TWI interfaces. While being programmed, the board can be connected to a PC through a USB port and powered via USB. The Arduino UNO's 32 KB Flash, 1 KB EEPROM, and 2 KB SRAM are all scalable. The board can be connected to multiple Arduino Shields for networking through Ethernet, Bluetooth, Wi-Fi, Zigbee, or Cellular networks. It is also compatible with a great deal of IoT systems.

Calibration: Calibrate the sensors to obtain a baseline for typical conditions. This requires exposing the sensors to well-known circumstances. The NTC temperature sensor (or Thermistor) and the humidity measuring element are the two major components of the DHT11 sensor. Thermistors operate as variable resistors, changing resistance in response to temperature variations. They both measure the temperature and humidity of the surrounding air before passing their findings to the IC, which is located on the sensor's rear side. The four pins on the sensor are VCC, Ground, data Out, and NC. The VCC and Ground pins are connected to the common VCC and Grounds, respectively.

The Arduino board's PB0 pin is linked to the sensor's Data Out pin. Because SnO2 has a reduced conductivity in clean air, making it a sensitive material, the MQ3 sensor is used to detect the presence of ethanol. As ethanol gas concentrations grow, so does the conductivity of the atmosphere. The MQ-3 sensor is both digital and analog. Food contaminated with ethanol vapours has deteriorated. As a result, the MQ3 sensor detects the beginning of food rot. The sensor's four pins are Ground, VCC, Analog Out, and Digital Output. The VCC is linked to the common VCC as well as Ground. The digital output pin is left unplugged because it is not in use. The analog output pin, which is coupled to the sensor, provides the output. LDR sensor: The LDR measures the light's intensity. The sensor is attached to the Arduino board's output-connected A1 pin. The sensor is linked to a circuit with a potential divider. The analog voltage generated by the LDR is converted into a digital reading by the built-in ADC.16X2 LCD - Join the 16X2 LCD display's data pins to pins 2 through 5 on the Arduino board. The RS and E pins of the LCD are linked to the Arduino board's pins 10 and 9, respectively. The RW pin on the LCD is grounded. The ESP8266 Wi-Fi Module is a self-contained SOC that can connect to wireless networks and contains an integrated TCP/IP protocol stack. The ESP8266 is capable of hosting an application or delegating all Wi-Fi networking duties to another application processor. Each ESP8266 module comes pre-programmed with an AT command set software. The module is offered in two models: ESP-01 and ESP-12. While ESP-12 has 16 interface pins available, ESP-01 only has 8 useable pins.

Data acquisition: Install an Internet of Things (IoT) gadget based on Arduino in a supermarket. It begins reading data from the interfaced sensors, including the MQ3 Sensor, LDR Sensor, and DHT-11 Temperature and Humidity Sensor, after correctly installed and switched on. The DHT-11 Temperature and Humidity Sensor is a digital sensor that may be used as a thermometer as well as a capacitive humidity sensor. Every two seconds, it offers a real-time readout of the temperature and humidity. The sensor can read temperatures ranging from 0°C to 50°C and relative humidity levels ranging from 20% to 95% when powered by a 3.5 to 5.5 V source. A union of 80 microseconds of LOW and 80 microseconds of HIGH occurred. After transmitting the start signal, the pin is set to digital. The start signal consists of a LOW for 18 milliseconds, a HIGH for 20 to 40 microseconds, a LOW for 80 microseconds, and a HIGH for 80 microseconds. The pin is set to digital output after receiving the start signal, and 40-bit data, including the temperature and humidity reading, is latch-out. In the first two bytes of the five-bit data, relative humidity is measured as an integer and a decimal number, temperature is measured as an integer and a decimal number, and a checksum byte is provided in the final byte.

The DHT-11 sensor already has an Arduino standard library. The read11() method of the DHT class may be used to quickly prepare sensor data. The LDR sensor transmits a voltage to the controller's analog input pin through a potential divider circuit link. Using the device's built-in ADC channel, the voltage is read and digitized.

Ethanol-related gas emissions are detected using the MQ3 sensor. Food and fruits create ethanol-like scents when they deteriorate. To detect such gases, the MQ3 sensor, which gives an analog signal proportional to the gas concentration, is utilized. The Arduino's built-in ADC receives the analog output and converts it to a digital value. The Arduino collects data from each sensor and turns the results into strings. The sensor data is sent to the character LCD and displayed in the relevant strings. The data is uploaded to ThingSpeak Server by the Arduino and the ESP8266 Wi-Fi module. A digital dashboard or a data broker is required to show and manage data that has been uploaded to the ThingSpeak server. A digital dashboard called Freeboard.io is utilized in this project to graphically track sensor data online. JASON files are used by Freeboard.io to show data from ThingSpeak. A dashboard can be built in three components.

1) Data sources - Data sources rely on outside sources for their information. These external sources could be data broker services, JavaScript scripts, or JSON files that include content from an HTTP server. The project's data source is a JSON file that receives data from the ThingSpeak server.

2) Widgets - The Widgets support the textual or graphical display of data. Freeboard.io offers a wide variety of widgets, including text, graphs, gauges, and more.

3) Panes - Widgets are arranged using this.

Data processing and analysis: The MQ3, DHT-11, and LDR sensors may all provide data, which the Arduino application can then stringify, display on a character LCD, and send to the ThingSpeak server. The common open-source Arduino library for interacting with DHT11 must first be loaded before importing the software Serial library for serial connection with the Wi-Fi module. The variables that represent the pin connections required to read sensor data undergo initialization. The serial communication's baud rate is set to 9600 when the setup() function is used to communicate with the Wi-Fi modem. The Wi-Fi mode and network connexion are established via AT instructions, but with significant delays. The delay should be displayed based on how long it takes to connect to the network. When the setup() method is used to interact with the Wi-Fi modem, the serial communication's baud rate is set to 9600. Through AT instructions, the Wi-Fi mode and network connection are established, although with noticeable delays. Based on how long it takes to connect to the network, the delay ought to be shown. The esp8266() method is given the AT commands to create a TCP connection, and then the ThingSpeak API key is given to send data to the specified channel. Every 16 seconds, the data on the ThingSpeak channel is refreshed.

Alert system: Sensors used in food quality are tailored for their specific applications, such as DTH11 for food humidity and temperature detection. The MQ3 Sensor is used to detect gas particles and vapour in closed container meals.These figures will be shown on the LCD panel. It is critical to note that the particular implementation details may change depending on the demands and constraints of your project. Consider adding more sensors or features as needed to improve the food quality monitoring capabilities.

Here the proposed method

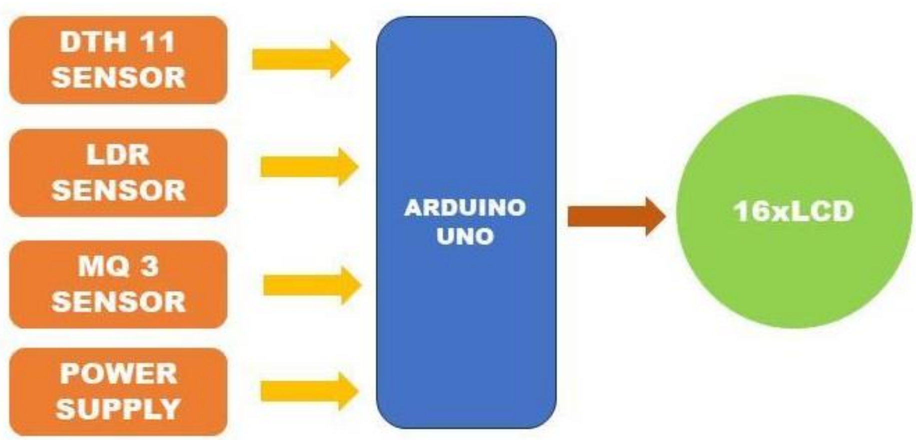

Fig 4 : Proposed Method

Creating a food quality sensor using the MQ-3 alcohol gas sensor and the DHT11 temperature and humidity sensor without using a NodeMCU involves interfacing these sensors with a microcontroller or microprocessor and implementing a suitable algorithm to analyze the data. Below is a general methodology to help you get started:

Methodology:

The MQ-3 alcohol gas sensor, the DHT11 temperature and humidity sensor, and a NodeMCU can be used to create a food quality sensor by connecting these sensors to a CPU and creating an appropriate algorithm to interpret the data. Listed below is a generic process to get you started:

Components required are:

1. Alcohol Gas Sensor MQ-3
2. DHT11 Temperature and Humidity Sensor
3. Microcontroller
4. Jumper wires and a breadboard
5. Power supply

Methodology:

Sensor Connections:

Connect the MQ-3 sensor to the appropriate microcontroller pins. Because the MQ-3 sensor generally produces an analog output, it is utilized to detect the gas volume of food in a closed container and connect its analog pin to one of the microcontroller's analog input pins. Connect the DHT11 sensor to a digital device that detects humidity and temperature in a closed container of food. LDR sensors are components that are used to sense data in light or open sunlight in order to determine the worth of food.

Power Supply:

Power up the MQ-3 and DHT11 sensors simultaneously. Be sure to provide consistent power and adhere to each sensor's voltage specifications.

MQ-3 calibration:

For the MQ-3 sensor to convert the analog output to alcohol concentration, calibration is necessary. The sensor must be exposed to a known alcohol concentration, and the analog output must be recorded. Create a calibration curve that connects sensor output to alcohol concentration using the data from this step.

Data Collection:

Read analog output from the MQ-3 sensor using the microcontroller's analog-to-digital converter (ADC). Convert this analog number to an alcohol concentration using the calibration curve.To read the temperature and humidity readings from the DHT11 sensor, use digital communication protocols.

Data Processing:

Assess the quality of the food using the collected data. For instance, temperature and alcohol concentration are crucial aspects to take into account. Given that some foods are sensitive to moisture, you might also wish to take humidity levels into consideration.

Alerts and Thresholds:

Set parameters that indicate what constitutes acceptable food quality, such as alcohol content, temperature, and humidity. Alerts should be set off if any of these numbers fall outside of the allowed range.

Alert System:

Set an alert system, such as an LED that blinks, a buzzer that sounds, or email or text message notifications. (Optional) The user interface You can optionally connect a display (such as an LCD) to the board's microcontroller to display readings and alarms in real time.

Evaluation and Improvement:

To ensure accurate readings and pertinent notifications, test the sensor system with a variety of foods and environmental factors while checking food quality by dth 11 and mq3 sensors inorder to prevent from any disease causes.

Power Control (Battery-Powered Configuration):

Use power-saving techniques to increase battery life if you're using batteries. This can entail sleeping the microcontroller and waking it up to take readings at set intervals.

Documentation:

Include all aspects of the setup, such as sensor connections, the calibration process, data processing methods, and alert systems, in your documentation. This documentation will be useful for future reference and troubleshooting. Keep in mind that your food quality sensor's accuracy is based on correct calibration, sensor quality, and the algorithm you choose to process the data. Additionally, it's critical to take into account the particular needs of the foods you're observing and modify your methods accordingly.

RESULTS AND DISCUSSION

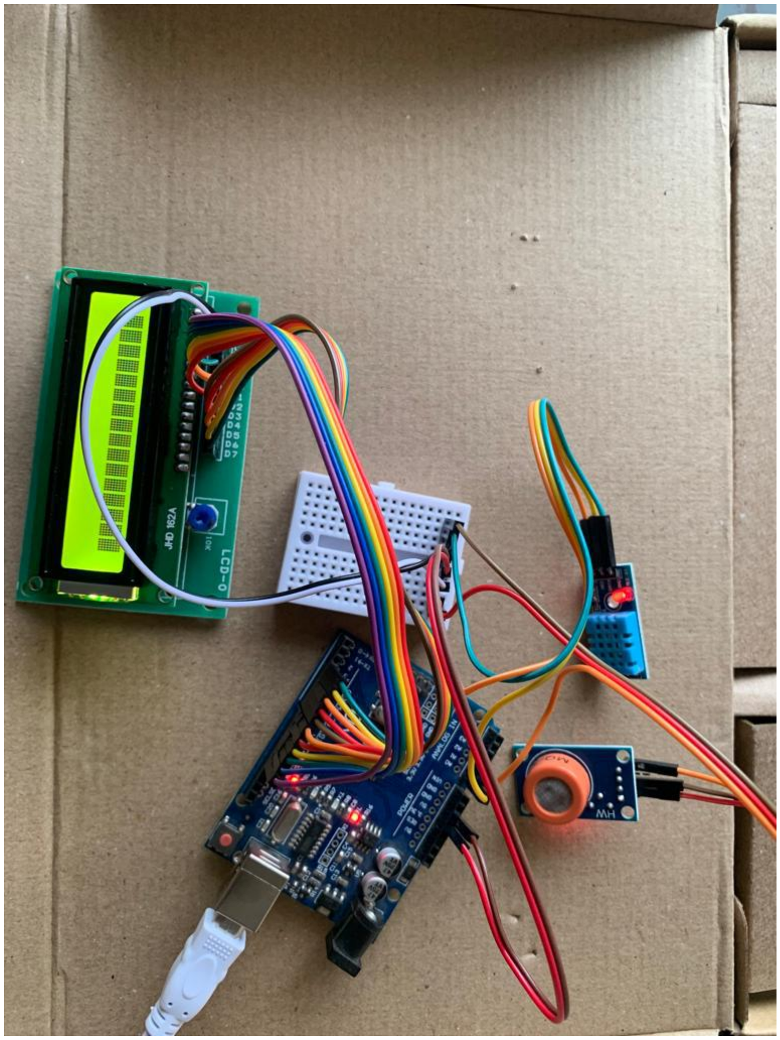

Fig 5: Hardware Module of Proposed Methodology

We, the undersigned, hereby declare the development of a Food Quality Sensor utilizing the DHT11 and MQ-3 sensors. This sensor aims to enhance the assessment and monitoring of food quality parameters, including temperature, humidity, and alcohol content, in various food products. The purpose of this sensor is to provide a reliable and cost-effective solution for individuals and businesses involved in the food industry to ensure the safety and freshness of food products. By accurately measuring and monitoring critical parameters, the sensor will assist in preventing spoilage, detecting contamination, and promoting food safety.Additionally, the MQ-3 sensor, based on gas-sensing technology, is integrated into the system to measure alcohol content. This feature is particularly useful for assessing the freshness and quality of fermented food and beverages. The MQ-3 sensor can detect the presence of alcohol vapor in the surrounding environment, allowing for real-time monitoring and ensuring compliance with quality standards.The Food Quality Sensor system is designed with the following key features as follows Accurate measurement of temperature, humidity, and alcohol content, Real-time monitoring and display of sensor readings, Alarm notifications for critical threshold breaches, Data logging capability for analysis and record-keeping, User-friendly interface for easy operation and configuration,Compact and portable design for versatile application.The result of the food quality sensing system using Arduino Uno R3, DHT11 sensor, MQ-3 sensor, and an LCD display will depend on the specific implementation and the thresholds set for determining the food quality.

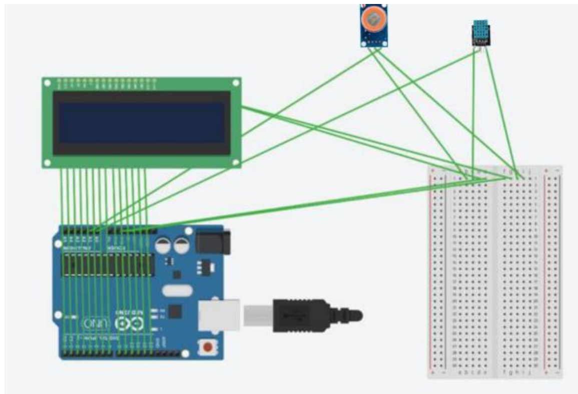

FIG 6: SCHEMATIC VISULATION

In general, the system can provide the following information.
Temperature and Humidity: The DHT11 sensor measures temperature and humidity. The system can display the current temperature and humidity values on the LCD display.

Gas Detection: The MQ-3 sensor measures alcohol and other gases. It can detect the presence of gases that may indicate food spoilage or contamination. The system can display the gas value on the LCD display.

Threshold Violations: The system can compare the temperature and gas values with predefined thresholds. If the temperature exceeds a certain threshold, it can display a warning message on the LCD display indicating high temperature. Similarly, if the gas value exceeds a specified threshold, it can display a warning message indicating the presence of high gas levels.

Quality Assessment: Based on the temperature and gas readings and the threshold violations, the system can provide an overall assessment of the food quality. For example, if the temperature and gas values are within acceptable ranges, it can display a message indicating that the food quality is "OK." If any threshold is violated, it can display a warning message indicating a potential issue with the food quality.

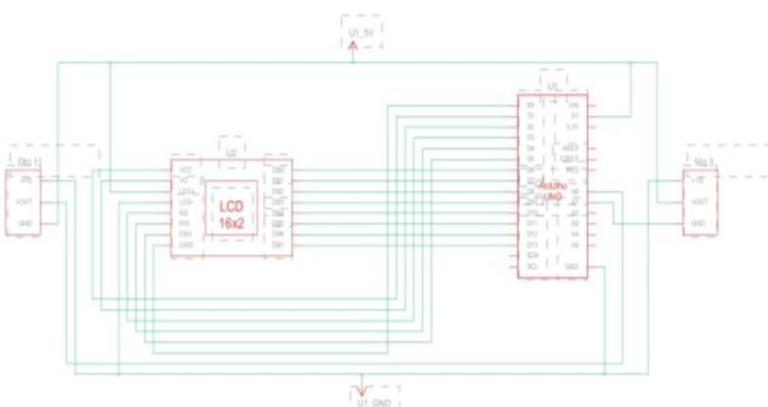

FIG 7: CIRCUIT DIAGRAM

Remember that the specific implementation and interpretation of the results may vary depending on your requirements and the algorithms you use to analyze the sensor data. The example code provided earlier in the conversation can serve as a starting point for developing your own food quality sensing system using the mentioned components.

CONCLUSIONS

One of the most important global concerns is food waste. Improper warehouse management is one of the leading causes of food loss, however given the level of technology, this problem is reasonably solvable. After reviewing several research and prospective solutions, we decided that the IoT industry can provide a highly effective solution to this problem. As a consequence, we discussed a food quality monitoring system based on the Internet of Things (IoT) that will govern many environmental characteristics such as light intensity, humidity, and temperature, all of which must be maintained at a particular level to prevent food spoilage. Additionally, it offers a user-friendly software that allows users to monitor the light intensity parameters and receive notifications when food is spoilt or a fire hazard arises. For reliable and effective food quality sensing, combine an LCD display, DHT11 sensor, MQ3 sensor, and Arduino R3. The central processing unit, the Arduino R3, receives information from the DHT11 sensor for monitoring temperature and humidity as well as information from the MQ3 sensor for detecting the presence of hazardous chemicals.The DHT11 sensor provides exact temperature and humidity measurements, enabling close observation of the food environment. Hazardous gases, such as carbon dioxide or ammonia, which may indicate spoilage or contamination, are crucially detected by the MQ3 sensor. Potential health risks can be detected early by continuously monitoring the air quality around the food, enabling prompt treatments to stop foodborne infections.A user-friendly interface, the LCD display provides real-time feedback on temperature, humidity, and gas levels. It makes it simple to understand sensor data, making it easier for both experts and consumers to evaluate the nutritional value and safety of food.Overall, the combination of these elements provides a thorough food quality sensing system, enabling people and organizations to take proactive steps to preserve food freshness, avoid spoilage, and guarantee customer safety. The Arduino R3, DHT11 sensor, MQ3 sensor, and LCD display together provide an effective solution for food quality monitoring in various settings, such as homes, restaurants, and food processing facilities.

REFERENCES

1. Ajay Doltade, Ankita Kadam, Sayali Honmore, Sanjeev Wagh, "Intelligent Grain Storage Management System based on IoT", International Journal of Science and Research (IJSR), March 2019,Vol. 8 Issue 3.
2. Amrita Srivastava, Ankita Gulati, "ITrack: IoT framework for Smart Food Monitoring System", International Journal of Computer Applications, August 2016, Vol. 148 - No.12.
3. Atkare Prajwal, Patil Vaishali, Zade payal, Dhapudkar Sumit, "Food Quality Detection and Monitoring System", IEEE International Student's 'Conference on Electrical, Electronics and Computer Science 2020
4. Bin Yu, Ping Zhan, Ming Lei, Fang Zhou & Peng Wang, "Food Quality Monitoring System based on Smart Contracts and Evaluation Models", IEEE, 13 Jan 2020, Vol. 8.
5. Fu Ying, Li Fengquan, "Application of internet of things to the Monitoring System for Food Quality safety", 2013 fourth international conference on digitalmanufacturing & automation, 16 September 2013.
6. Jun Liu, Yue Shen, Cheng'an Wang, Jiahui Li, "Design of Fruits Warehousing Monitoring and Control System Based on Wifi ", IEEE, 2015.
7. Dr. Keshavmurty, Megaha shree.M, Meghamala M, Mariyam Steffi J, "Automatized Food Quality Detection & System using Neural Network", 4th international conference on recent trends on electronics, information, communication & technology, 18 May
8. Maen Taruri, Abubakar, Noora Alnaqbi, Hessa Al Shehhi, Abdul Halim M Jallad & Amine Bermak , "AMachine Learning Approach Monitoring using a DoFP Sensor",IEEE, 17 Aug 2020, Vol. 8
9. Meo Vincent C.Caya,Febus Reidj G.Cruz,Carolle Marian N.Fernando,Wen yaw chung, "Monitoring and Detection of Fruits & Vegetable Spoilage in the Refrigerator using Electronic Nose Based on Principal Component Analysis", IEEE, 15 Jun 2020.
10. Naveed Shahzad, Usman Khalid, Atif Iqbal, Meezan-Ur- Rahman, "eFresh – a Device to Detect Food Freshness", International Journal of Soft Computing and Engineering (IJSCE), September 2018, Vol. 8 Issue 3.
11. Ravi Chander B, P.A. Lovina & G. Shiva Kumari, "Food Quality Monitoring System by using Arduino", Journal of engineering sciences, Apr 2020, Vol. 11 Issue 4.
12. Venkatesh.A, Saravanakumar.T, Vairamsrinivasan S, Vigneshwar A, Santhosh Kumar M, "A Food Monitoring System Based on Bluetooth Low Energy and Internet of Things", International Journal of Engineering Research and Application, March 2017,Vol. 7 Issue 3.