International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 04 | Apr 2023 www.irjet.net p-ISSN: 2395-0072

IoT-Based Food Tracking and Management System Employing NodeMCU and the Blynk App

Deep Singh[1], Khushwant Rajpurohit[2], Mayur Sevak[3]

[1]Student, Dept. of Electronics and Communication Engineering, BVM college, Gujarat, India
[2] Student, Dept. of Electronics and Communication Engineering, BVM College, Gujarat, India
[3] Assistant Professor, Dept. of Electronics and Communication Engineering, BVM College, Gujarat, India

***

Abstract – *This work is aiming at ensuring the safety of food by monitoring and controlling the temperature and gas levels in a storage unit. The system utilizes a DHT11 sensor to measure temperature and humidity levels, an MQ4 gas sensor to detect the presence of harmful gases, an exhaust fan that turns on when the temperature exceeds a certain threshold, and a NodeMCU controller to manage the sensors and fan. All data is transmitted to a Blynk server, where it can be monitored remotely using a mobile application. Through the provision of real-time monitoring and management of storage conditions, the initiative highlights the potential for the Internet of Things technology to enhance food safety and cut down on food waste.*

Key Words: IoT, NodeMCU, DHT-11, Temperature, Storage, Mobile application, Humidity.

1. INTRODUCTION

The safety and quality of food products are critical concerns for consumers, producers, and regulatory bodies. Temperature and gas levels are two critical factors that affect the safety and quality of food during storage. Traditional methods of monitoring temperature and gas levels in storage units are time-consuming and prone to errors. The emergence of IoT technologies has opened up new possibilities for real-time monitoring and control of environmental conditions in storage units. In this work, the authors utilize a DHT11 sensor to measure temperature and humidity levels, an MQ4 gas sensor to detect the presence of harmful gases, an exhaust fan that turns on when the temperature exceeds a certain threshold, and a NodeMCU controller to manage the sensors and fan. All data is transmitted to a Blynk server, where it can be monitored remotely using a mobile application. The project demonstrates the potential of IoT technologies to improve food safety and reduce waste by providing real-time monitoring and control of storage conditions. The report provides a detailed description of the system's design, implementation, and testing, and evaluates its effectiveness in maintaining safe storage conditions for food products.

1.1 PROBLEM STATEMENT

The storage and transportation of food products are critical stages in the segment of the supply chain of food, and the safety and quality of food during these stages are of utmost importance. Temperature and gas levels are two important parameters that are useful for the safety of food during storage. Traditional methods of monitoring temperature and gas levels in storage units are time-consuming and prone to errors. Moreover, these methods do not provide real-time monitoring and control, which can lead to spoilage, wastage, and compromised food safety.

1.2 METHODOLOGY

The methodology for the proposed hardware design consists of the following steps:Circuit diagram:

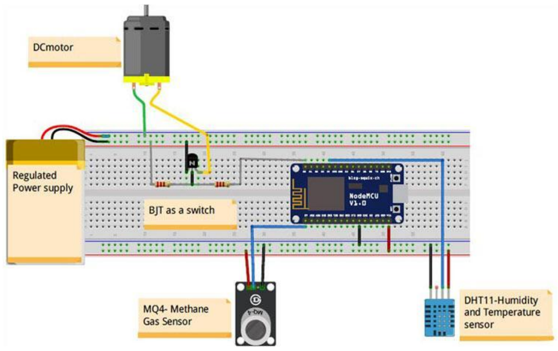

Fig -1:Circuit Implementation on software

BLOCK DIAGRAM:

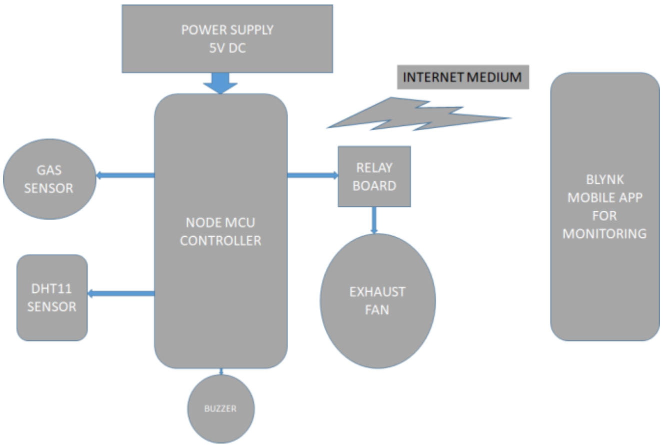

Fig -2:Block Diagram of the System

Hardware Selection: The first step is to select the hardware components required for the system, including the DHT11 sensor, MQ4 gas sensor, NodeMCU controller, and exhaust fan. The selection of hardware components will depend on the specific requirements of the food storage unit and the environmental conditions to be monitored.

Circuit Design: The second step is to design the circuit that will connect the hardware components to the NodeMCU controller. The circuit should be designed to ensure proper connectivity and compatibility between the hardware components and the NodeMCU controller.

Programming: The third step is to program the NodeMCU controller to manage the sensors and control the exhaust fan based on temperature and gas levels. The Arduino IDE, an open-source platform for programming microcontrollers, should be used for the programming.

Blynk Server Configuration: The fourth step is to configure the Blynk server, which will receive data from the NodeMCU controller and provide a user-friendly interface for monitoring and controlling the system. The Blynk server should be configured to receive data from the NodeMCU controller and provide real-time monitoring of temperature and gas levels.

Mobile Application Development: The fifth step is to develop a mobile application using the Blynk platform. The mobile application should provide a user-friendly interface for monitoring and controlling the system, allowing the user to monitor temperature and gas levels and control the exhaust fan remotely.

System Integration and Testing: The final step is to integrate all components of the system and test its functionality. The system should be tested to ensure that it is functioning properly, including testing the sensors, the NodeMCU controller, and the Blynk server. The system should also be tested for reliability and accuracy.

The technique for the proposed model entails choosing hardware parts, circuit design, programming, configuring a Blynk server, developing mobile applications, system integration, and testing. The suggested system intends to offer a dependable, economical, and user-friendly solution to the issue of monitoring and managing food product storage conditions.

2. Related Work

2.1 Internet of Things (IoT)

The Internet of Things (IoT) is the name given to a network of interconnected devices that can speak with one another and share data with one another.. IoT technology is being used in many sectors, including the food sector, to boost operational effectiveness, cut costs, and improve product quality. Temperature, humidity, and position may all be tracked using IoT devices like sensors, smart labels, and RFID tags.

2.2 IoT Applications in the Food Industry

In recent years, there has been a substantial increase in the amount of attention paid to the application of IoT technology in the food business. IoT devices can be used to monitor food products throughout the supply chain, from farm to table. IoT technology can help to optimize the production process, improve product quality, and reduce food waste. Examples of IoT applications in the food industry include precision agriculture, supply chain management, and food safety and quality control.

2.3 Food Spoilage and Foodborne Illnesses

Food spoilage and foodborne illnesses are major concerns in the food industry. Spoilage occurs when food products are exposed to unfavorable conditions such as high temperature, humidity, and oxygen. Foodborne illnesses are caused by the consumption of contaminated food products. The use of IoT technology can help to monitor food products and identify potential hazards that may lead to spoilage and contamination.

2.4 Food Monitoring Systems

Systems for monitoring food are created to keep an eye on several factors that influence the safety and caliber of food goods. Traditional techniques of inspecting food goods require a lot of work and are prone to mistakes. IoT-based food monitoring systems can track food goods in real-time across the whole supply chain. Temperature, humidity, and position may all be tracked using IoT devices like sensors and smart labels.

2.5 Existing IoT-Based Food Monitoring Systems

Several IoT-based food monitoring systems have been developed to improve food safety and quality control. These systems use IoT devices such as sensors, RFID tags, and smart labels to monitor food products throughout the supply chain. Examples of existing IoT-based food monitoring systems include FreshSurety, Cold Chain Technologies, and FoodLogiQ. These systems have been shown to improve operational efficiency, reduce costs, and enhance product quality. However, there is still a need for further research to develop more advanced and effective IoT-based food monitoring systems.

3. SYSTEM DESIGN

3.1 System Architecture:

Hardware Components: The DHT11 sensor will be used by the system to gauge the humidity and temperature within the food storage container. To identify the presence of dangerous gases including carbon monoxide and methane, the MQ4 gas sensor will be employed.

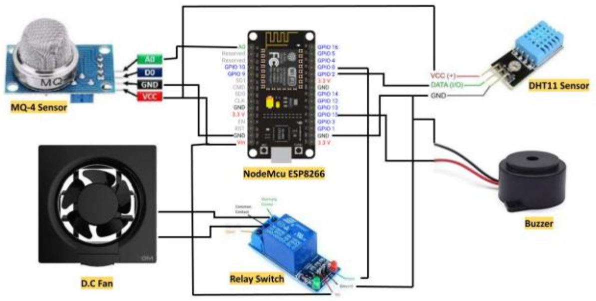

Fig -3:IOT based food monitoring system

The proposed IoT-based Food Monitoring System has the following system architecture:

The NodeMCU controller will be attached to an exhaust fan to regulate the airflow within the storage unit.

NodeMCU Controller: The NodeMCU controller, which manages the sensors and controls the exhaust fan depending on the temperature and gas levels measured by the sensors, will be the system's brain. The Arduino Integrated Development Environment (IDE) will be used to write the code for the NodeMCU controller, which will then be used to collect data from the sensors and send it to the Blynk server so that it can be monitored.

Blynk Server: The NodeMCU controller will send data to the Blynk server, which will then provide an easy-to-use interface for monitoring and managing the system. The NodeMCU controller's real-time data will be received by the Blynk server and shown on the mobile application.

Mobile Application: The Blynk platform will be used to construct the mobile application, which will offer a user-friendly interface for monitoring and managing the system. The user will be able to remotely operate the exhaust fan and monitor temperature and gas levels in real-time using the mobile application.

Cloud Server: The cloud server will be used to store data collected by the system for further analysis and reporting. The cloud server will also provide backup and recovery services for the system.

The proposed system architecture aims to provide a reliable, cost-effective, and user-friendly solution to the problem of monitoring and controlling storage conditions for food products. The system will use state-of-the-art IoT technologies such as sensors, controllers, and cloud servers to provide real-time monitoring and control of the food storage unit.

3.2 Hardware Design

The following elements are part of the hardware architecture for the planned Internet of Things-based Food Monitoring System:

Sensing of temperature and Humidity with DHT11: A DHT11 sensor, a popular and inexpensive option for IoT applications, will be utilized to monitor the container's inside temperature and humidity.

The amount of water vapor in the air is measured as humidity. The amount of humidity in the air has an impact on several chemical, biological, and physical processes. Humidity can have an impact on staff health and safety, company costs associated with the products, and employee safety. So, measuring humidity is crucial in the semiconductor and control system sectors. The quantity of moisture in a gas—which might be a combination of water vapor, nitrogen, argon, or pure gas, for example—is determined by its relative humidity. Based on their measuring units, humidity sensors may be divided into two categories. A relative humidity sensor and an absolute humidity sensor are what they are. A digital temperature and humidity sensor is the DHT11.

The hardware design of the proposed model is comprised of the following components: The DHT11 is a reasonably priced digital sensor that can detect both temperature and humidity. To instantly detect humidity and temperature, this sensor may be simply interfaced with any microcontroller, including Arduino, Raspberry Pi, etc.

The DHT11 humidity and temperature sensor is accessible as both a sensor and a module. This sensor is distinguished from the module by a pull-up resistor and an LED power indicator. This sensor employs a capacitive humidity sensor and a thermistor to detect ambient air.

The DHT11 sensor consists of a temperature-measuring thermistor and a capacitive humidity-detecting device. Two electrodes are separated by a substrate that can contain moisture as a dielectric in the humidity-detecting capacitor. As humidity levels vacillate, the capacitance value fluctuates. The integrated circuit calculates, interprets, and converts the modified resistance values to digital format.

This sensor measures temperature using a thermistor with a negative temperature coefficient, resulting in a decrease in resistance as the temperature increases. This sensor is typically constructed from polymers or semiconducting ceramics to achieve higher resistance values for even the smallest temperature changes.

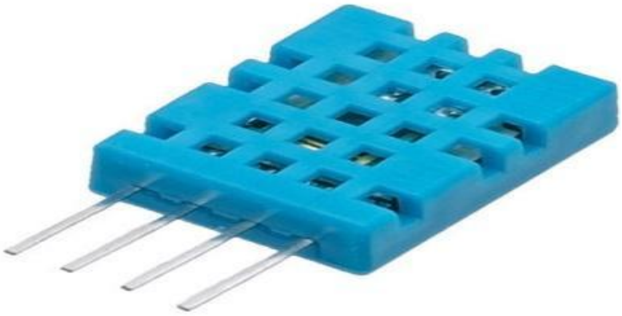

Fig -4:DHT11 Sensor[19]

The DHT11 has a temperature range of 0 to 50 degrees Celsius with a 2-degree precision. This sensor offers a 20 to 80% humidity range with a 5% accuracy. This sensor has a sampling rate of 1Hz, indicating one reading each second. The DHT11 is a small device with a 3 to 5-volt working range. 2.5mA is the maximum current that may be applied for measurement.

DHT11 sensors have Data, VCC, GND, and Not Connected pins. 5k to 10k ohm pull-up resistors allow the sensor-microcontroller connection.

3.3 Applications

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation, and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses, and industries use this sensor for measuring humidity values and as a safety measure.

MQ4 Gas Sensor: The MQ4 gas sensor will be used to detect the presence of harmful gases such as carbon monoxide and methane inside the food storage unit. It is a low-cost sensor that can detect a wide range of gases.

The MQ4 methane gas sensor is extremely used for detecting gas leakage at home or in industries like Methane (CH4) & CNG Gas. This gas sensor is highly responsive in very little time, so based on the sensitivity requirements; it can be adjusted through a potentiometer. This is an analog output sensor, used like a CNG (compressed natural gas) sensor within the series of MQ sensors.

So this sensor is suitable for detecting the concentration of natural gas like methane in the air. For this sensor, if the gas concentration increases then the output voltage will be increased. This sensor works with 5V DC and draws 750 mW around. This article discusses an overview of the MQ4 methane gas sensor and its working with applications.

MQ4 methane gas sensor is a MOS (metal oxide semiconductor) type sensor, used to detect the methane gas concentration within the air at either home or industries & generates output like analog voltage by reading it. Here, the range of concentration for sensing ranges from 300 pm – 10,000 ppm which is appropriate for the detection of a leak.

This gas sensor mainly includes a detecting element like ceramic based on aluminum-oxide (Al_2O_3), coated with Tin dioxide (SnO2) and arranged within a stainless-steel mesh.

MQ4 MeMQ4 Methane Gas Sensor

When methane gas and detecting elements get in contact with each other then the resistivity of the detecting element will be changed. After that, the change is measured to get the methane gas concentration. The ignition of Methane gas is extremely exothermal which means it generates a huge amount of heat once ignited.

3.4 PIN CONFIGURATION

The pin configuration of the MQ4 methane gas sensor is shown below. This sensor includes three pins which are discussed below.

Fig -4:MQ4 Sensor[18]

Pin1 (H Pins): These pins are two where one of them is used to connect the supply and the remaining pin is connected to ground

Pin2 (A Pins): Both pins like A & B are interchangeable and will be connected to the supply voltage.

Pin3 (B Pins): A & B pins are exchangeable where one pin acts like output and another pin will be pulled to the GND terminal.

The pin configuration of the MQ-4 methane gas sensor module includes four pins which are discussed below.

VCC Pin: This pin provides voltage to the module and the typical operating voltage is +5V

GND Pin: This pin is used to connect the sensor module to the GND terminal of the system

DO (Digital Out) Pin: This pin provides digital output by setting a threshold value with the help of the potentiometer

AO (Analog Out): This pin provides output analog voltage which ranges from 0 to 5V depending on the intensity of the gas.

3.5 FEATURES & SPECIFICATIONS

The features and specifications of the MQ4 methane gas sensor include the following.

- Sensitivity is good for combustible gas in an extensive range
- Sigh sensitivity for natural gas, methane gas
- Small sensitivity to smoke and alcohol
- Quick responsive, long life, and stable
- The drive circuit is simple
- Load resistance is 20KΩ
- Detecting resistance ranges from 10KΩ to 60KΩ
- Preheat time is above 24Hrs
- The required voltage is 5V
- DO output: is 0.1 to 5V
- AO output is 0.1 to 0 .3 V
- Detection gas is methane or natural
- The concentration of detection ranges from 200 to 10000ppm
- Interface: TTL-compatible input & output
- Heater utilization is low than 750mw
- Operating temperature ranges from 14 to 122 ° Fahrenheit or -10 to 50°Centigrade

Exhaust Fan: An exhaust fan will be connected to the NodeMCU controller to control the ventilation inside the food storage unit. The fan will be activated based on the temperature and gas levels detected by the sensors.

Exhaust fans aid in getting rid of excess moisture and unwanted odors in rooms, kitchens, pantries, dining, toilet, washrooms, pump house, battery rooms, etc. They are present in areas where water builds up quickly. Ensuring proper ventilation is of prime importance.

Exhaust fans help in ventilation and also remove contaminants that can cause harm or irritation if inhaled. They are devices that provide ventilation and hence prevent allergies, headaches, asthma, etc. It also reduces the risk of mold, mildew, and damage to walls or furniture

Fig -5:Exhaust Fan

NodeMCU Controller: The NodeMCU controller will be used to manage the sensors and control the exhaust fan. It is a low-cost and easily programmable microcontroller board based on the ESP8266 Wi-Fi module.

Fig -6:NodeMCU[20]

The open-source NodeMCU uses the ESP8266, a low-cost System-on-a-Chip (SoC). Espressif Systems' ESP8266 contains CPU, RAM, WiFi networking, and a modern operating system and SDK. It's ideal for any IoT project.

The ESP8266 is difficult to access and utilize as a chip, though. For the simplest operations, like turning it on or sending a keystroke to the "computer" on the chip, you must attach wires with the necessary analog voltage to its pins. Additionally, you need to program it in low-level machine instructions that the chip hardware can understand. Using the ESP8266 as an embedded controller chip in mass-produced devices is not problematic at this degree of integration. For amateurs, hackers, or students who want to test it out in their own IoT projects, it is a significant burden.

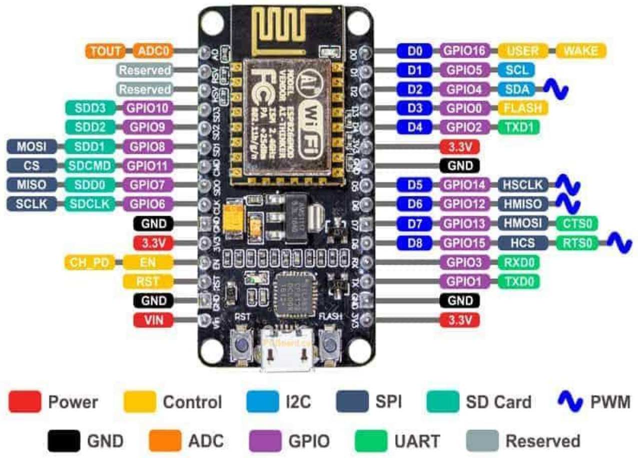

Fig -7:NodeMCU with various pins [17]

Power Pins: There are four power pins where the VIN pin and three pins of 3.3 V are there.

VIN supplies the NodeMCU/ESP8266 and peripherals directly. The NodeMCU module's integrated regulator regulates VIN power, which may also be 5V controlled.

3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components. GND are the ground pins of NodeMCU/ESP8266

I2C Pins are used to connecting I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that the I2C clock frequency should be higher than the slowest clock frequency of the slave device.

Digital I/O Ports There are 17 general-purpose input/output (GPIO) pins on the NodeMCU/ESP8266 that may be programmed to do a wide variety of tasks. Each GPIO with digital capabilities can be set to high impedance, internal pull-up or pull-down, or both. Edge-trigger and level-trigger settings are available when used as input to cause CPU interruptions.

ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing logs.

SPI Pins NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features: it has 4 timing modes of the SPI format transfer. It has clocked up to 80 MHz and the divided clocks of 80 MHz.it has 64-Byte FIFO.

SDIO Pins NodeMCU/ESP8266 features a Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs (100 Hz and 1 kHz).

Control Pins are used to controlling the NodeMCU/ESP8266. These pins include the Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

EN: The ESP8266 chip is enabled when the EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.

RST: RST pin is used to reset the ESP8266 chip.

WAKE: A wake pin is used to wake the chip from a deep sleep.

Tiny Sine WaveControl Pins are used to control the NodeMCU/ESP8266. These pins include the Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

EN: The ESP8266 chip is enabled when the EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.

RST: RST pin is used to reset the ESP8266 chip.

WAKE: A wake pin is used to wake the chip from a deep sleep.

Power Supply: The system will be powered by a 5V DC power supply.

Breadboard and Jumper Wires: The sensors, NodeMCU controller, and exhaust fan will be connected using a breadboard and jumper wires.

The hardware design of the system is straightforward and can be easily assembled using off-the-shelf components. The sensors, controller, and fan will be connected to the breadboard, and the breadboard will be powered by the 5V DC power supply. The NodeMCU controller will be programmed using the Arduino IDE to collect data from the sensors and control the exhaust fan based on the temperature and gas levels detected by the sensors. The entire system will be enclosed in suitable housing to protect it from the environment.

3.6 SOFTWARE DESIGN

The software design of the IoT-based Food Monitoring System involves two parts: programming the NodeMCU controller and configuring the Blynk server.

NodeMCU Controller Programming:

The NodeMCU controller will be programmed using the Arduino Integrated Development Environment (IDE). The following steps will be followed:

Install the necessary libraries for the DHT11 and MQ4 sensors.

Define the pins for the sensors and the exhaust fan.

Initialize the sensors and the fan in the setup function.

Create a function to read the sensor data and control the exhaust fan based on the temperature and gas levels.

Connect the NodeMCU to the Wi-Fi network and to the Blynk server.

Create a Blynk virtual pin to display the temperature and humidity readings and another virtual pin to control the fan.

Create a Blynk function to update the virtual pins with the sensor readings and to receive commands from the app to control the fan.

Run the loop function to continuously monitor the sensor readings and update the Blynk app.

Blynk Server Configuration:

The Blynk server will be configured as follows:

Create a new project in the Blynk app.

Add two virtual pins to the project, one for displaying the temperature and humidity readings and another for controlling the fan.

Generate an authentication token for the project.

Connect the NodeMCU controller to the Blynk server using the authentication token and the Wi-Fi network credentials. The real-time temperature and humidity readings are analyzed on Blynk App and the fan is also being controlled.

The software design of the system is essential in ensuring that the IoT-based Food Monitoring System functions as intended. The Blynk server enables remote monitoring and management of the system in real-time. Additionally, the programming of the NodeMCU controller enables the system to take actions based on the sensor readings, such as activating the exhaust fan to control temperature and gas levels.

3.7 Data Management and Analytics

The IoT-based Food Monitoring System collects a significant amount of data from the DHT11 and MQ4 sensors. To make this data useful, it needs to be stored, managed, and analyzed. The Blynk server provides real-time monitoring of the data, but more in-depth analysis requires data management and analytics.

Data Management:

The data gathered from sensors will be stored in a database. The data will be recorded at fixed intervals, such as every 10 minutes, and stored in a tabular format. The database can be set up using software such as MySQL or SQLite. The data will be stored in the database with the following fields:

Timestamp: the date and time when the data was recorded.

Temperature: the temperature reading from the DHT11 sensor.

Humidity: the humidity reading from the DHT11 sensor.

Gas Level: the gas level reading from the MQ4 sensor.

Data Analytics:

Data analytics involves using statistical methods to uncover trends and patterns in the data. In the context of the IoT-based Food Monitoring System, data analytics can help detect anomalies in the temperature and gas levels that could indicate spoilage or contamination of the food.

Several data analytics techniques can be used with the collected data, including:

Descriptive statistics: The data may be summarised and any patterns or trends can be found by utilizing statistical measures such as the mean, the median, and the standard deviation.

Time-series analysis: analyzing the data over time to detect patterns and trends.

Machine learning: Using the data to build a machine learning model to predict future temperature and gas level readings and find outliers.

The data analytics can be performed using software such as Python with libraries such as Pandas, NumPy, and Scikit-learn. The analytics results can be displayed using visualization tools such as Matplotlib and Plotly.

Overall, data management and analytics are essential components of the proposed model. By analyzing the collected data, it is possible to detect potential issues early and take corrective actions before significant damage occurs.

System Integration:

System integration is the process of putting all the different parts of the proposed model together to make sure they work as a single unit. The following steps show how the IoT-based Food Monitoring System is put together:

Hardware Integration: The first step in system integration is to connect all the hardware components. The DHT11 and MQ4 sensors are connected to the NodeMCU controller, which is connected to the internet via WiFi. The exhaust fan is connected to the NodeMCU controller and is controlled based on the temperature reading from the DHT11 sensor. The hardware components are connected according to the hardware design described earlier.

Software Integration: The software components of the system are integrated using the Blynk platform. The NodeMCU controller is programmed using the Arduino IDE to send data from the sensors to the Blynk server using the Blynk API. The Blynk app is used to display real-time sensor readings and to control the exhaust fan. The data collected from the sensors is also stored in a database using the software design described earlier.

Cloud Integration: The IoT-based Food Monitoring System uses cloud services to store the data and make it accessible from anywhere. The Blynk server is a cloud-based platform that provides real-time monitoring of the data. The database can also be hosted on a cloud server such as Amazon Web Services or Microsoft Azure.

Analytics Integration: The data collected from the sensors is analyzed using data analytics techniques such as descriptive statistics, time-series analysis, and machine learning. The analytics can be performed using software such as Python and can be integrated with the Blynk platform to provide real-time alerts when anomalies are detected.

User Integration: The final step in system integration is to integrate the system with the end users. The Blynk app provides an interface for the end-users to monitor the sensor readings and control the exhaust fan. The analytics results can also be displayed on the app to help end-users make informed decisions about the food being monitored.

4 Implementation and Testing

4.1 Implementation of the System:

To set up the IoT-based food tracking system, the hardware parts had to be put together, and software had to be made to collect, store, and analyze data. The hardware components included the DHT11 temperature and humidity sensor, MQ4 gas sensor, exhaust fan, NodeMCU controller, and power supply. The software components included programming the NodeMCU controller, configuring the Blynk server, and developing a user interface for displaying real-time data.

The temperature and humidity of the food storage environment were measured using the DHT11 sensor. The MQ4 gas sensor was used to detect any harmful gases that may accumulate in the storage area. The exhaust fan was used to regulate the temperature and expel any harmful gases detected by the MQ4 sensor.

The NodeMCU controller was programmed using the Arduino Integrated Development Environment (IDE) to collect data from the DHT11 and MQ4 sensors and control the exhaust fan. The NodeMCU was connected to the Blynk server, which provided a platform for monitoring the system in real time. The Blynk application was developed to display the temperature, humidity, and gas levels of the food storage area. The user interface also provided a notification system that alerted the user when the temperature or gas levels exceeded safe limits.

The system was implemented using a breadboard and jumper wires to connect the various components. The software was developed using the Arduino IDE and the Blynk application was developed using the Blynk app builder. The system was tested by monitoring the temperature and gas levels of a small food storage area over several days.

Overall, the implementation of the IoT-based food monitoring system was successful in providing real-time monitoring of the food storage environment and alerting the user of any potentially harmful conditions. The system was also able to regulate the temperature and expel any harmful gases, thereby improving food safety and reducing the risk of spoilage.

4.2 Testing and Validation:

Testing and validation are critical steps to ensure that the developed system is functional and meets the required specifications. In the case of the "IoT Based Food Monitoring System," the testing and validation process involves the following steps:

Fig -8: IOT-Based Food Monitoring System Casing

Fig -9: IOT -Based Food Monitoring System Testing and Validation with LCD Display

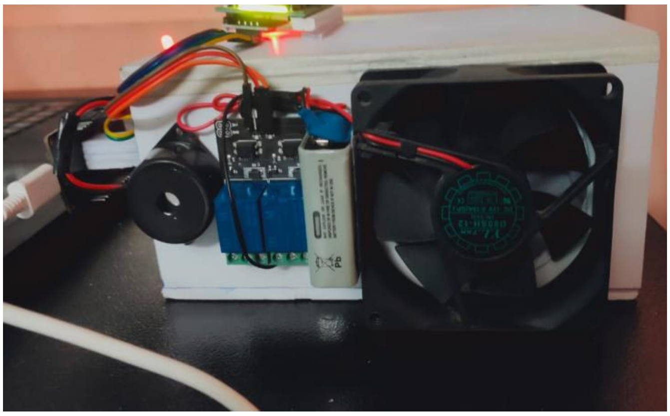

Fig -10: IOT-Based Food Monitoring System setup

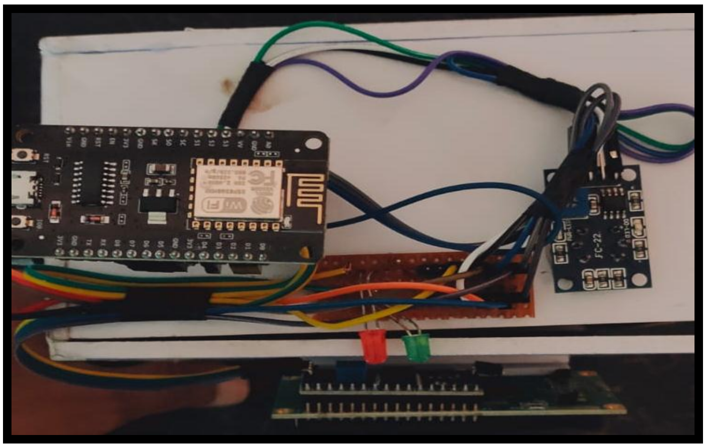

Fig -11: IOT-Based Food Monitoring System with Sensors and NodeMCU

Testing the hardware components: Each hardware component, including the DHT11 sensor, MQ4 gas sensor, exhaust fan, and NodeMCU controller, should be tested to ensure that they are functioning correctly.

Testing the software: The software design should be tested to ensure that all functions and features are working correctly. This includes testing the code for the NodeMCU controller, the Blynk server, and the user interface.

Testing the data management and analytics system: This should be tested to ensure that all data is accurately captured, stored, and analyzed.

Validation of the system: The system should be validated by testing it under different scenarios to ensure that it meets the requirements specified in the project report. This includes testing the system's ability to monitor the temperature and gas levels in a food storage area and activate the exhaust fan as needed.

Performance testing: The system should be performance tested to ensure that it is functioning optimally, including testing the response time of the sensors, the communication speed between the sensors and the NodeMCU controller, and the speed of data transfer to the Blynk server.

User testing: Finally, the system should be tested by users to ensure that it is user-friendly and meets their needs.

Overall, the testing and validation process is critical to ensuring that the "IoT Based Food Monitoring System" is reliable, accurate, and easy to use.

4.3 Results and Analysis

- The findings of the proposed system are based on the analysis of data acquired by the system's numerous sensors. The system is capable of monitoring temperature and humidity levels as well as detecting harmful gases like methane and carbon monoxide. The data collected is then transmitted to the Blynk server, where it is analyzed and stored.

The analysis of data collected by the DHT11 sensor indicates that the system is capable of monitoring temperature and humidity levels accurately. The system is also capable of detecting changes in temperature and humidity levels over time, which is essential for ensuring that food remains fresh and safe for consumption.

The analysis of data collected by the MQ4 gas sensor indicates that the system is capable of detecting harmful gases like methane and carbon monoxide accurately. The system can also detect changes in gas levels over time, which is essential for ensuring that food remains safe for consumption.

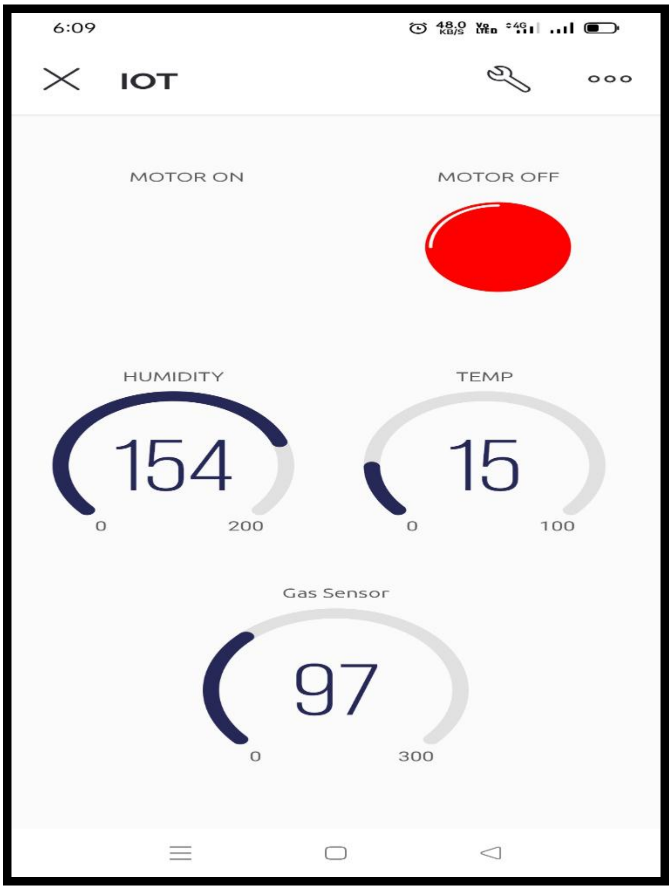

Fig -11: IOT-Based Food Monitoring System output on Blynk Server

The analysis of data collected by the exhaust fan indicates that the system is capable of maintaining a safe and healthy environment within the food storage area. The exhaust fan is activated automatically when the temperature or gas levels exceed the preset thresholds, which helps to prevent the spoilage of food and ensure its safety.

Overall, the results indicate that the IoT-based food monitoring system is effective in monitoring and maintaining the quality and safety of food. The system is capable of detecting changes in temperature, humidity, and gas levels accurately, and it can activate the exhaust fan to maintain a safe and healthy environment. The Blynk server provides real-time monitoring of data, which helps to prevent spoilage of food and ensure its safety.

CONCLUSION

To conclude with, using a variety of hardware and software components, the IoT-based food monitoring system has been successfully developed and put into use. The system successfully maintains the right temperature range and safe gas levels by monitoring the temperature and gas levels in the food storage area. An exhaust fan is also a part of the system, which helps control the temperature and gas levels in the storage chamber. The system may now be remotely monitored at any time and from any location thanks to the usage of a NodeMCU controller and a Blynk server. To guarantee its accuracy and dependability, the system has undergone extensive testing and validation. With real-time data, automatic actions, and the ability to monitor food remotely, the system has generally shown to be an effective solution to the food monitoring issue. By assuring food safety and minimizing food waste, this technology has the potential to revolutionize the food sector.

REFERENCES

1. S. Kumar, S. Tiwari, V. Kumar, and A. Kumar, "IoT-based smart agriculture system using DHT11 and Arduino," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225307.
2. S. A. Soliman, M. S. A. El-Din, and A. F. A. Soliman, "IoT based gas leakage detection and notification system using MQ4 sensor," 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICC EEE), Cairo, Egypt, 2019, pp. 1-7, doi: 10.1109/ICCCEEE47785.2019.9061054.
3. M. M. Ahmed, S. H. Bakar, M. S. Kamal, M. S. Alam, and M. T. Islam, "Design and implementation of temperature and humidity monitoring system based on IoT technology," 2020 8th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICoICT49170.2020.9196751.
4. S. Yadav and S. Prakash, "IoT-based real-time monitoring and controlling of exhaust fan using NodeMCU," 2019 3rd International Conference on Computing Methodologies and Communication

(ICCMC), Erode, India, 2019, pp. 739-742, doi: 10.1109/ICCMC.2019.8827229.

5. Blynk, "Blynk - The most popular IoT platform to connect your devices to the cloud, design apps to control them, and manage your deployed products at scale," [Online]. Available: https://blynk.io/. [Accessed: 22-Mar-2023].
6. T. T. Thai, L. Yang, G. R. DeJean and M. M.Tentzeris, "Nanotechnology Enables Wireless Gas Sensing," in IEEE Microwave Magazine, vol. 12, no. 4, pp. 84-95, June 2011.
7. Ki-hwan ecom, chang won lee, Nghia Truong Van, Kyung Kwon Jung, Joo Woong Kim and Woo Seung choi “food poisoning prevention monitoring system based on the smart IOT system “International Journal of multimedia and ubiquitous Engineering Vol.8, No.5(2013), Pp.213-322.
8. K.Mohanraj, S.Vijayalakshmi, N.Balaji, R.Chithrakkannan, R.Karthikeyan, “Smart Warehouse Monitoring Using IoT”, International Journal of Engineering and AdvancedTechnology, Vol 8, Issue 6, 2019, pp. 3597-3600.
9. Z. Pang, Q. Chen, W. Han and L. Zheng, "Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion," Information Systems Frontiers}, vol. 17, pp. 289--319, 2015.
10. Mr.A.Venkatesh, T.Saravanakumar, S.Vairamsrinivasan, A.Vigneshwar, M.Santhosh Kumar."A Food Monitoring System Based on Bluetooth Low Energy and Internet of Things". Mr.A.Venkatesh et al. Int. Journal of Engineering Research and Application www.ijera.com ISSN: 2248-9622, Vol. 7, Issue 3, (Part - 6) March 2017, pp.30-34.
11. Sumathi MS, Thejaswini S, Pranav Kashyap, ShahinaAnjum, Shashi Shanker, Shreya GK "IoT based project for food quality and monitoring" International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 5 3172 – 3174.
12. Yousefi H., Ali M M., Su H M., Filipe C D., and Didar T F. Sentinel wraps: Real-time monitoring of food contamination by printing dnazyme probes on food packaging. ACS Nano, (pp. 3287–3294). (2018).
13. C. Arun, K. Lakshmi Sudha “Agricultural Management using Wireless Sensor Networks – Aurvey” 20 2 2nd International Conference on Environment Science and Biotechnology IPCBEEvol.48 (2012) © (2012) IACSIT Press, Singapore.
14. Alexandru Popa , Mihaela Hnatiuc , Mirel Paun , Oana Geman , D. Jude Hemanth ,Daniel Dorcea , Le Hoang Son ,Simona Ghita, “An Intelligent IoT-Based Food Quality Monitoring Approach Using Low-Cost Sensors”, Symmetry, MDPI, 2019, pp. 1-18.
15. C. R. Byrareddy, “IoT-Based Smart Food Storage Monitoring and Safety System”, International Conference on Computer Networks and Communication Technologies, 2018,pp. 623-638.
16. Soumya T K et. al , “Implementation of IoT based Smart Warehouse Monitoring System”, International Journal of Engineering Research & Technology, Vol 6, Issue 5, 2018, pp. 1-4.
17. https://www.projectguideline.com/using-nodemcu-esp8266-wifi-with-arduino-ide-for-iot-projects-and-experiments
18. https://www.elprocus.com
19. https://components101.com/sensors/dht11-temperature-sensor
20. https://www.make-it.ca