ISSN: 3068-014X

Research Article

Archivos de Ciencia e Investigación

Conceptual Design of an IoT-Based Livestock Environmental Condition Monitoring System

Tunji John ERINLE[1,2*], Joshua Boluwatife AJEWOLE[3] and Ayodeji Israel FATOLA[4]

[1]*Department of Mechanical Engineering, School of Engineering, Federal Polytechnic, Nigeria*

[2]*Department of Mechatronics Engineering, School of Engineering, Federal Polytechnic, Nigeria*

[3]*Department of Mechanical Engineering & Mechatronics Engineering, College of Engineering, Landmark University, Nigeria*

[4]*Department of Mechanical Engineering, Faculty of Engineering, Kwara State University, Nigeria*

[*]Corresponding Author
Tunji John ERINLE, Department of Mechanical Engineering, School of Engineering, Federal Polytechnic, Nigeria; Department of Mechanical Engineering, School of Engineering, Federal Polytechnic, Nigeria.

Submitted: 2026, Jan 16; **Accepted:** 2026, Feb 20; **Published:** 2026, Feb 24

Citation: ERINLE, T. J., AJEWOLE, J, B., FATOLA, A. I. (2026). Conceptual Design of an IoT-Based Livestock Environmental Condition Monitoring System. *Arch Cienc Investig, 2*(1), 01-10.

Abstract

Livestock farming plays a vital role in global food production, but its efficiency and sustainability are often challenged by poor environmental conditions that negatively impact animal health, welfare, and productivity. This study presents the design of a smart, Internet of Things (IoT)-based system tailored to monitor and manage key environmental factors within livestock housing environments. The system integrates multiple sensors, including temperature, humidity, and air quality sensors, with a microcontroller, which processes and transmits data in real time to a cloud platform using Wi-Fi and GSM technology. A mobile and web-based interface allows farmers to visualize environmental trends, receive instant alerts when conditions exceed preset thresholds, and make timely, informed decisions to protect animal welfare. The design prioritizes affordability, modularity, and ease of use, making it suitable for both smallholder and commercial farms. Solar-powered support ensures operation in off-grid rural areas, while built-in alert mechanisms, such as a buzzer and LED, provide immediate on-site feedback. Calibration and field testing would demonstrate the system's accuracy and reliability under varying farm conditions. By enabling real-time monitoring, predictive insight, and proactive intervention, the proposed system supports smarter livestock management, reduces animal stress, minimizes disease risks, and enhances productivity. Beyond the technological innovation, the study advocates for a shift toward data-driven, ethical farming practices aligned with global sustainability goals. It would bridge the digital divide in agriculture and equip farmers with a practical tool to better understand and respond to the environmental needs of their animals. This research contributes not only to the body of knowledge in precision livestock farming but also to the broader mission of ensuring food security and sustainable agriculture in an increasingly climate-sensitive world.

Keywords: Agricultural Innovation, IoT, Livestock Monitoring, Precision Agriculture, Sustainability

1. Introduction

The integration of IoT technology in agriculture has transformed cattle production, notably in terms of environmental monitoring. Livestock animals are sensitive to environmental changes, which affect their health, productivity, and well-being. As a result, establishing an IoT-based livestock environmental condition monitoring system is critical for successful animal farming operations [1]. Traditional techniques of monitoring livestock surroundings, such as manual data collection and human observation, are labour-intensive and prone to human error, limiting precision livestock husbandry. IoT technology enables real-time data collection, remote monitoring, and sophisticated analytics to improve decision-making [2].

IoT-based livestock monitoring systems collect real-time environmental data through interconnected sensors, microcontrollers, wireless communication modules, and cloud-based data platforms. These data are transferred to a centralised server or mobile application over a Wi-Fi network, allowing for real-time monitoring and abnormal condition notifications [3]. Environmental stresses such as heat and poor air quality can affect animals' immune systems, resulting in decreased production, increased disease susceptibility, and death. Heat stress is a problem for poultry, and excessive levels of ammonia and carbon dioxide in animal housing are bad for respiratory health. Continuous monitoring reduces dangers and improves the well-being of animals [4]. Islam et al. developed a smart cattle monitoring system that employs sensors to detect temperature and humidity in animal shelters and sends data to a cloud server over WiFi. However, the system has difficulties in rural connection and scalability, necessitating further study.

The integration of IoT with data analytics and machine learning algorithms allows for predictive maintenance and intelligent alerts, predicting environmental issues and suggesting corrective actions before they escalate. This proactive approach reduces losses, improves animal comfort, and optimises farm operations efficiency [5]. Such a method has a substantial economic impact since it eliminates disease outbreaks, lowers death rates, and increases production, resulting in higher farmer profitability. Data collected may be used to inform policymaking, animal welfare certification, and sustainable agricultural methods, all of which are aligned with the worldwide move towards smart agriculture and the United Nations SDGs. Particularly, Goal 2 is to alleviate hunger, while Goal 12 is to promote responsible consumption and production (Food and Agriculture Organization [6].

IoT-based monitoring of livestock farms improves transparency, customer confidence, and supply chain integrity. It resolves concerns about animal welfare and environmental effects, while also giving verifiable evidence for environmentally friendly agricultural techniques, which boosts market competitiveness [7].

Implementing IoT in livestock monitoring presents problems such as high initial infrastructure costs, a lack of technical competence, data privacy concerns, and connection limitations in remote locations [8]. However, advances in low-power wide-area networks and low-cost sensor technologies make adoption more likely [9]. To improve animal welfare and farm production, an IoT-based livestock environmental monitoring system must be designed to be tough, waterproof, dustproof, and able to function under different circumstances, as well as prioritise energy efficiency, particularly in off-grid rural areas [10,11]. Mekonnen et al. constructed an energy-efficient IoT-based livestock monitoring system with solar-powered nodes and edge computing. The system reduces latency and provides real-time responsiveness, reflecting the growing trend of decentralised IoT systems.

An IoT-based livestock environmental monitoring system transforms current animal husbandry by supporting data-driven management, animal health, operational efficiency, economic growth, and sustainable practices. Shahinfar et al. emphasised the importance of collaborative frameworks combining governments, NGOs, and agritech businesses to promote IoT adoption and rural farmer training, therefore, realising the full potential of smart livestock monitoring. Ravi et al. highlighted the need for simple dashboards that present environmental data in local languages and with simplified images, allowing farmers without sophisticated technical understanding to benefit.

1.1. Benefits and Potential Applications

1.1.1. *Benefits*

i. Monitoring environmental conditions can help prevent heat stress, respiratory problems, and other health issues.
ii. Optimizing environmental conditions can improve animal growth, reproduction, and overall productivity.
iii. Early detection of environmental issues can help reduce animal mortality.
iv. Enhanced decision-making: Data-driven insights can inform decision-making and improve farm management.

1.1.2. *Potential Applications*

i. Monitoring environmental conditions in poultry farms can help improve bird health and productivity.
ii. Monitoring environmental conditions in dairy farms can help improve cow health and milk production.
iii. Monitoring environmental conditions in pig farms can help improve pig health and growth.

2. Materials and Methods

2.1. Materials

A real-time environmental monitoring IoT system generally contains sensors that monitor temperature, humidity, and air quality levels, as well as a microcontroller to process data and a communication module to send the data to a remote platform. The design also includes power management, data storage, and a possible user interface for visualisation and control.

2.1.1. Sensor Selection and Placement

Sensors: Air quality, temperature, and humidity are monitored using a semiconductor gas sensor (MQ-135) as well as a humidity and temperature sensor. These sensors can detect and monitor a wide range of chemicals in the air, including ammonia, nitrogen oxides, alcohol, benzene, smoke, and carbon dioxide, as well as environmental conditions.

Placement: Sensors should be fitted to the casing to collect representative data.

2.1.2. Microcontroller and Data Acquisition

Microcontroller: An Arduino Uno with an ESP8266 microcontroller is used to read, analyse, and send sensor data, as well as operate other components.

Data Acquisition: The microcontroller collects data from the sensors and could perform initial processing, such as averaging or filtering.

2.1.3. Communication Module

Connectivity: WiFi or cellular modules would be used to send data to a remote server or cloud platform.

Data Transmission: The microcontroller sends the processed data to the chosen platform for storage, analysis, and visualization.

2.1.4. Power Management

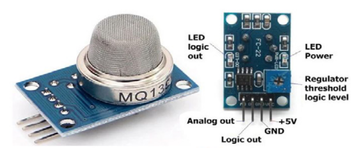
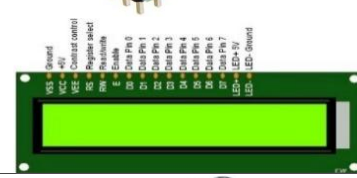
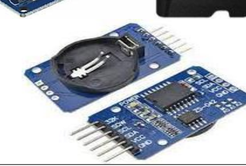
Power Source: The device would be powered via solar panels and batteries.

Power Optimization: Low-power modes and efficient data transfer processes assist in extending battery life.

2.1.5. Data Logging and Visualization

Data Storage: Data will be stored on the Micro SD card and cloud platform storage media for additional evaluation. User Interface: A liquid crystal display (LCD) and mobile application would enable users to examine real-time data and set alerts. IoT devices could be placed in the farm environment to collect data from the sensors. The cloud stores and processes the collected data, providing insights and alerts to farmers.

Table 1 provides a thorough list of electronic and electrical components, as well as additional accessories, necessary for the design of an IoT-based environmental monitoring system for real-time monitoring in livestock farming.

Component	Description	Images
Microcontroller	Arduino Uno It serves as the central processor, handling sensor inputs and decision logic.	
Gas Sensor	MQ-135 (Semiconductor Type) It is used to detect, measure, and monitor a wide range of gases present in the air, and it transmits data to the microcontroller.	
Temperature & Humidity Sensors	DHT11 It measures the temperature and relative humidity of the farm environment.	
Data Communication Modules	Wi-Fi Module (ESP8266) It transmits sensor data to a mobile app in real-time.	
Display Unit	LCD Display (16 x 2) It provides visual feedback for local farmers to view sensor readings.	
Power Supply Unit (Solar Panel and Battery)	12V Power Supply Module It powers the system to support off-grid operation.	
Storage Device	Micro SD Card Module with Card	
Real-Time Clock (RTC)	RTC Module	

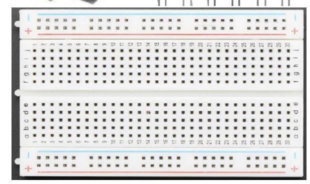
Connectors and Cables	Jumper Cables for Connection	
Buzzer/LEDs	Local alerts for critical environmental conditions	
Breadboard	PVC Breadboard	

Table 1: Materials for the Design of the System with Images

2.2. Methods

The sensors are embedded in various areas of the field and linked to an Arduino Uno. The microprocessor receives analogue and digital input from the gas, temperature, and humidity sensors. The ESP8266 Wi-Fi module connects the Arduino to a remote mobile app, allowing for real-time data visualisation. A smartphone application was created to notify users of threshold breaches. A gas monitor is a device that detects gases in the air by placing it in the surroundings. The circuit diagram describes how the circuit elements connect one another, as shown in Figures 1–3. The gas sensor was attached to the Arduino Uno's digital pin 9. The liquid crystal display (LCD) for data visualisation was connected to analogue pins 4, serial data (SDA), and 5, serial clock (SCL), respectively. Figure 4 depicts a block diagram of the designed system.

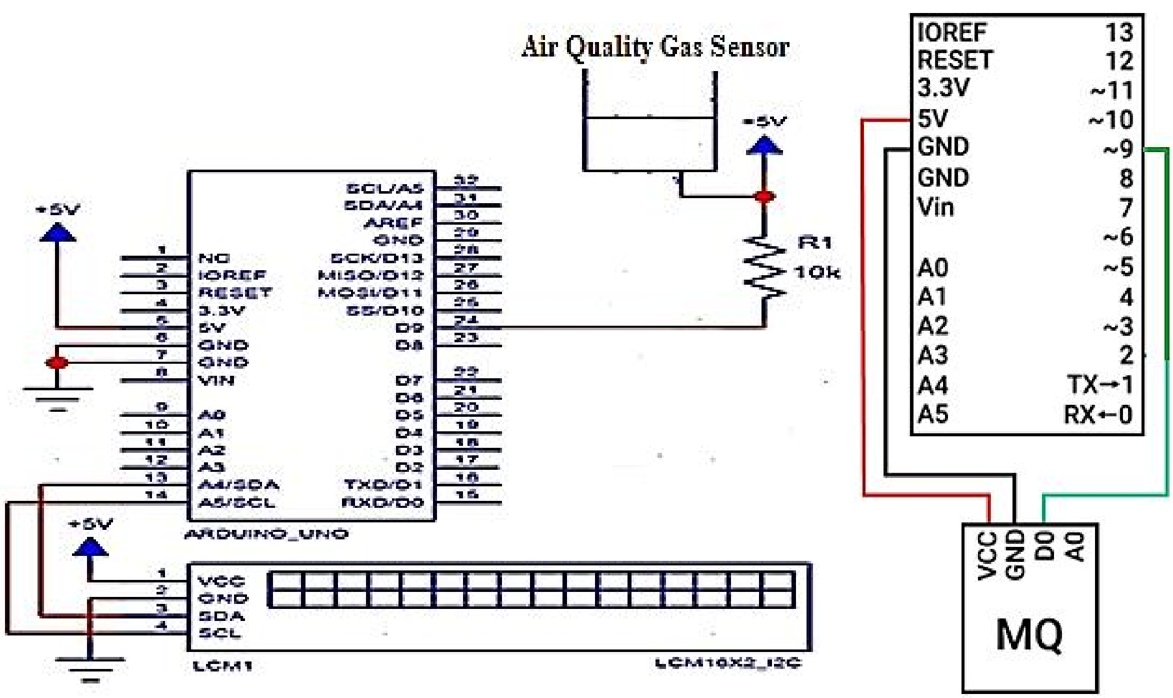


Figure 1: Circuit Diagram for Air Quality Gas Sensor Connection

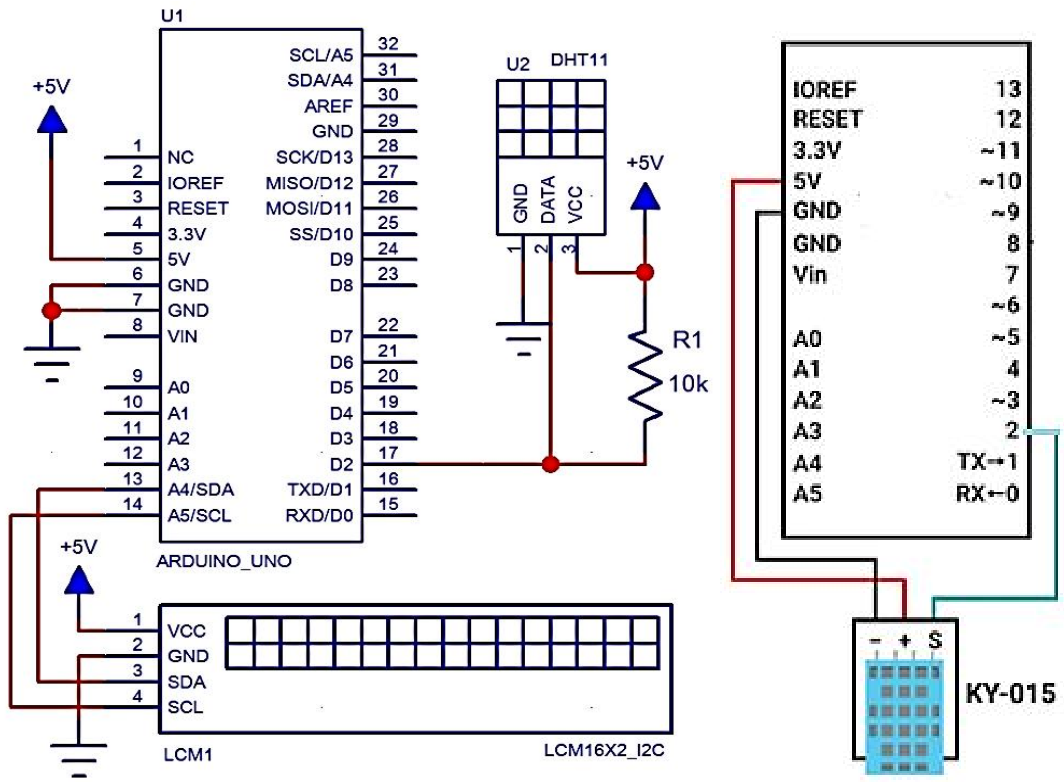


Figure 2: Circuit Diagram for Temperature and Humidity Sensor Connectio

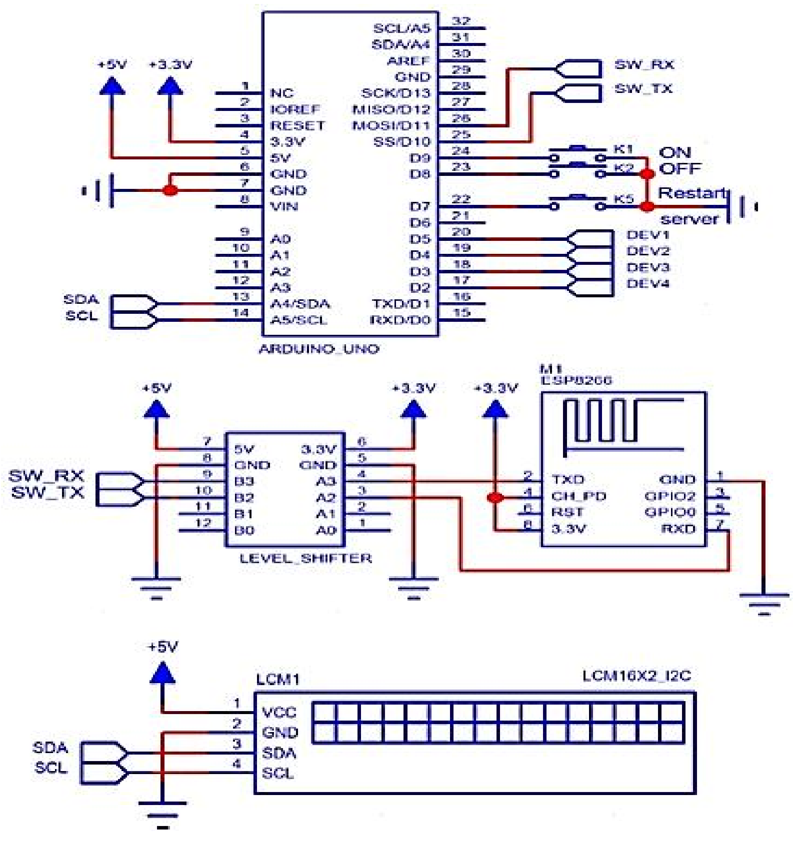


Figure 3: Circuit Diagram for ESP8266 Wi-Fi Module Connection

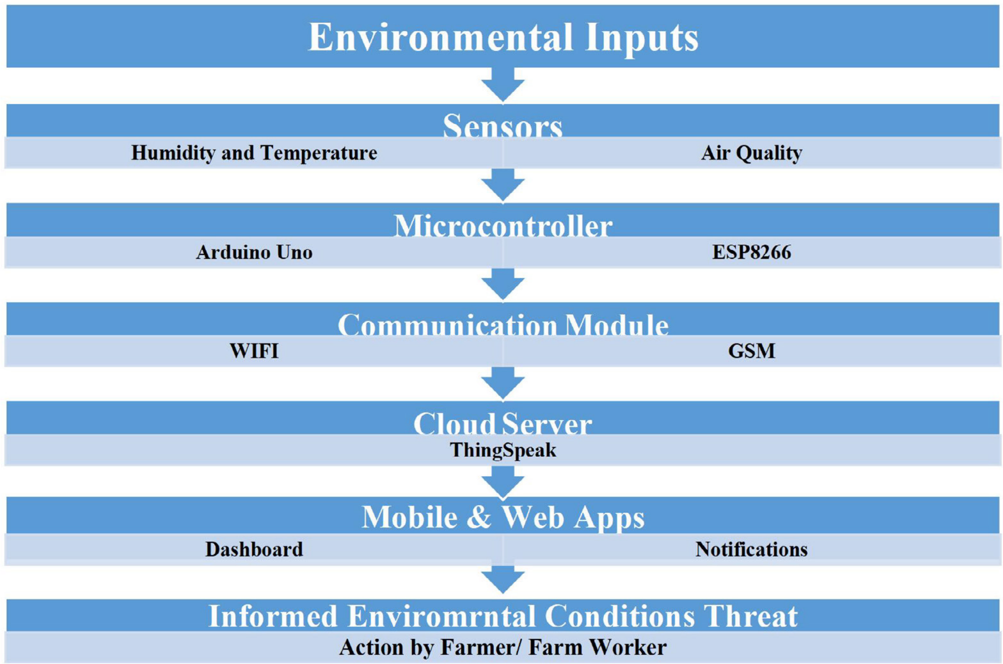


Figure 4: Block Diagram for the Designed System

3. Results and Discussion

3.1. Programming of the Monitoring of the Soil Moisture and Environmental Conditions Parameter

Programming a parameter monitoring system includes setting hardware, writing code to communicate with sensors, and creating software for data collecting, processing, and perhaps control. Figure 5 shows the Arduino Integrated Development Environment (IDE), which is used to program the microcontroller in C++.

Figure 5: Programming of Monitoring of the Parameters

The air quality gas sensor is used to monitor the gas in the environment, as shown in Figure 6. The buzzer +ve pin was connected to the digital 2 pin of the Arduino Uno, while the gas sensor was connected to the digital pin 9. When the gas content falls below a certain value, the buzzer will start to beep. The programming code for the microcontroller controls the gas sensor to produce the desired output response.

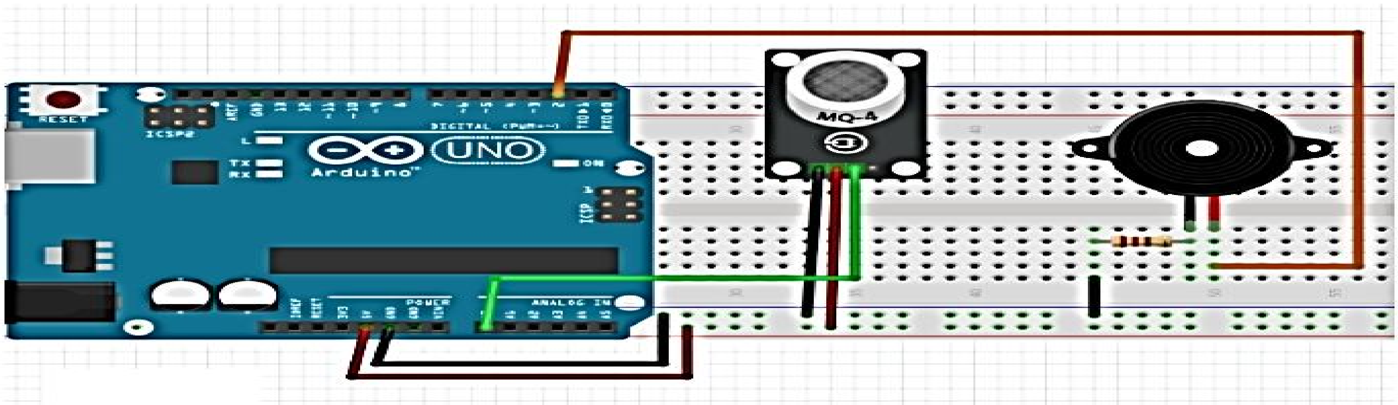


Figure 6: Air Quality Gas Sensor with Buzzer Connection to Microcontroller

```
// Define digital pin where D9 is connected
const int sensorPin = 2;
const int LEDPin = 13; // Define pin for the onboard LED
void setup() {
 pinMode(sensorPin, INPUT);
 pinMode(LEDPin, OUTPUT);
 Serial.begin(9600);
}
void loop() {
 // Read digital signal from MQ-135
 int sensorState = digitalRead(sensorPin);
 if(sensorState == HIGH) {
 // gas concentration is above the threshold
 digitalWrite(LEDPin, HIGH); // Turn on the LED
 Serial.println("Gas detected!");
 } else {
 digitalWrite(LEDPin, LOW); // Turn off the LED
 Serial.println("No gas detected.");
 }

 delay(1000);
}
```

The temperature and humidity sensor module provides a digital serial interface for monitoring humidity and temperature, as shown in Figure 7. The programming code for the microcontroller controls the temperature and humidity sensor to produce the desired output response of the environmental conditions.

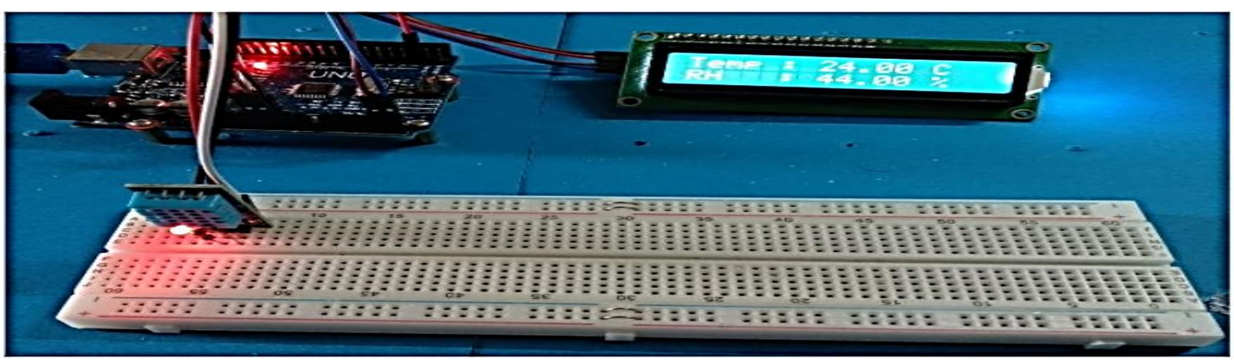


Figure 7: Temperature and Humidity Sensor Connection to Microcontroller

```
const int sensorPin = 2;
// buffer for data
byte dat[5];
byte readData () {
 int result = 0;
 for (int i = 0; i < 8; i ++) {
 while (digitalRead(sensorPin) == LOW);
 delayMicroseconds (30);

 if (digitalRead(sensorPin) == HIGH) {
 result | = (1 << (8 - i));
 }
 while (digitalRead(sensorPin) == HIGH);
 }
 return result;
}
void startTest () {
 // send start signal
 digitalWrite(sensorPin, LOW);
 // delay 30 ms for DHT11 to read start signal
 delay(30);
 digitalWrite(sensorPin, HIGH);
```

```
// waiting for a response from DHT11
delayMicroseconds (40);
pinMode(sensorPin, INPUT);
while (digitalRead(sensorPin) == HIGH);
// waiting for DHT11 data to be sent
delayMicroseconds (80);
if (digitalRead(sensorPin) == LOW) {
delayMicroseconds (80);
}
for (int i = 0; i < 5; i ++) {
dat [i] = readData ();
}
pinMode(sensorPin, OUTPUT);
digitalWrite(sensorPin, HIGH);
}
void setup() {
Serial.begin(9600);
pinMode(sensorPin, OUTPUT);
}
void loop() {
startTest ();
Serial.print("Humidity = ");
// Output the whole part of the moisture
Serial.print(dat [0], DEC);
Serial.print('.');
// Display the decimal part of moisture
Serial.print(dat [1], DEC);
Serial.println('%');
Serial.print("Temperature = ");
// Output the integer part of the temperature
Serial.print(dat [2], DEC);
Serial.print('.');
// Display the decimal part of the temperature
Serial.print(dat [3], DEC);
Serial.println('C');
// checksum check
byte checksum = dat [0] + dat [1] + dat [2] + dat [3];
if (dat [4]! = checksum)
Serial.println("- Checksum Error!");
else
Serial.println("- OK");
delay(1000);
}
```

The ESP8266 Wi-Fi Module acts as a web server. An Android App accesses the web server to control the real-time monitoring of gas, temperature, and humidity for data visualisation, as shown in Figure 8. Figure 9 shows the block diagram for the ESP8266 Wi-Fi connection to the Mobile App. The programming code for the microcontroller controls the ESP8266 Wi-Fi to transmit the information from the sensors through the Mobile App.

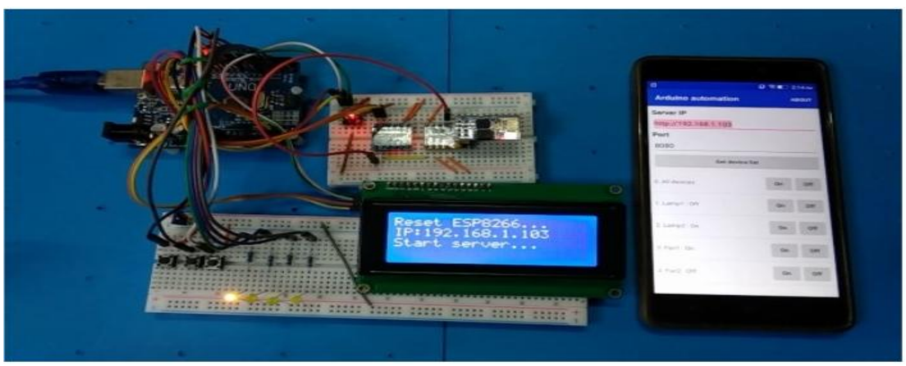


Figure 8: ESP8266 Wi-Fi Module Connection to Microcontroller with Mobile App

```
#include <ESP8266WiFi.h>
#include "DHT.h"
// WiFi credentials
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// Server endpoint (replace with your server or local IP)
const char* host = "thingspeak.com"; // 192.168.1.100/the cloud API
const int httpPort = 80;
// DHT Settings
#define DHTPIN D4 // GPIO2
#define DHTTYPE DHT11 // Use DHT22 if needed
DHT dht(DHTPIN, DHTTYPE);
void setup() {
Serial.begin(115200);
delay(10);
dht.begin();
Serial.println("Connecting to Wi-Fi...");
WiFi.begin(ssid, password);
// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.print(".");
}
Serial.println("\nWiFi connected. IP address: ");
Serial.println(WiFi.localIP());
}
void loop() {
// Read temperature and humidity
```

```
float h = dht.readHumidity();
float t = dht.readTemperature(); // Celsius
// Read air quality from analog pin A0
int airQualityValue = analogRead(A0);
// Check for errors
if (isnan(h) || isnan(t)) {
Serial.println("Failed to read from DHT sensor!");
delay(2000);
return;
}
// Print data to serial monitor
Serial.print("Temperature: "); Serial.print(t);
Serial.print(" °C, Humidity: "); Serial.print(h);
Serial.print("ppm, Air Quality: "); Serial.println(airQualityValue);
// Prepare GET request URL
String url = "/upload?temp=" + String(t) +
"&hum=" + String(h) +
"&soil=" + String(airQualityValue)
WiFiClient client;
if (!client.connect(host, httpPort)) {
Serial.println("Connection to server failed!");
delay(5000);
return;
}
// Send HTTP GET request
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
delay(10);
// Print response from server
while (client.available()) {
String line = client.readStringUntil('\r');
Serial.print(line);
}
Serial.println();
Serial.println("Data sent successfully.");
delay(15000); // Delay 15 seconds before next reading
}
```

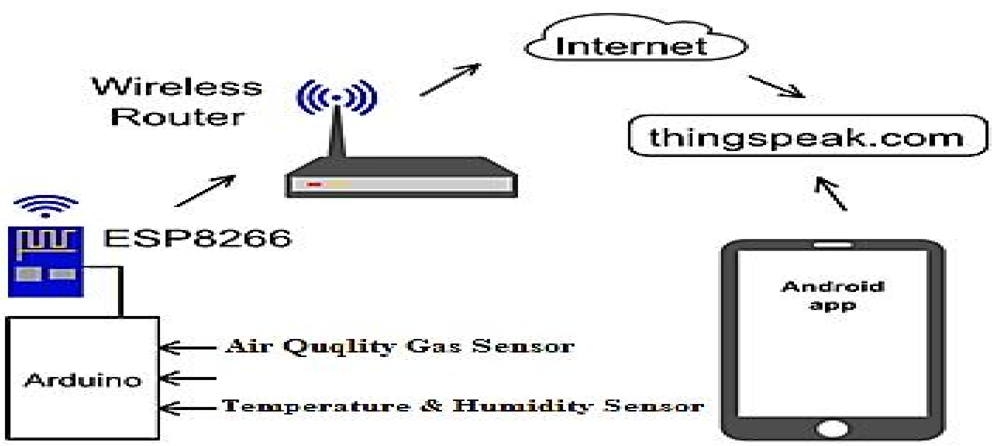


Figure 9: Block Diagram for ESP8266 Wi-Fi Connection to Mobile App/Cloud Server

3.2. Calibration and Threshold Setting

The gas sensor can be calibrated by observing its response to clean atmospheric air under stable conditions. Threshold values for gas concentration, temperature, and humidity can be established based on standard guidelines for livestock health and comfort. Figure 10 presents the design model of the IoT-based livestock monitoring system for gas and environmental conditions in livestock housing.

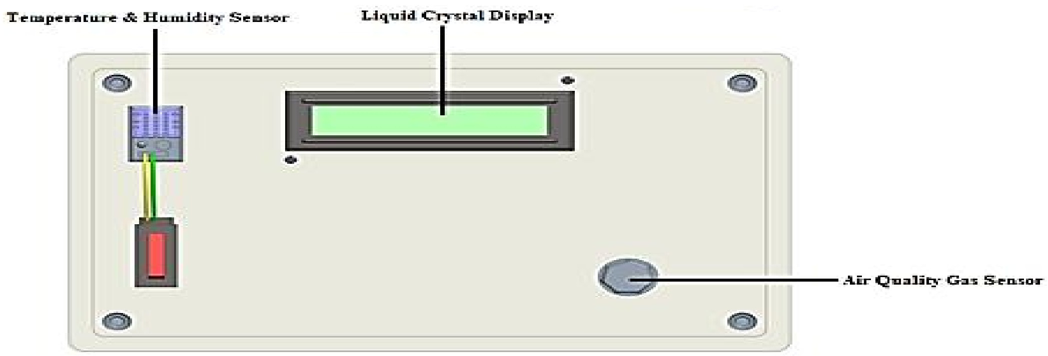


Figure 10: Design Model of the IoT-Based Livestock Environmental Condition Monitoring System

4. Conclusion

The design of an IoT-based environmental condition monitoring system represents a key leap in modern livestock farming, with an emphasis on precision, sustainability, and animal welfare. The study designed an IoT-based environmental condition monitoring system for livestock farming, addressing issues such as heat stress, poor air quality, and limited real-time monitoring capabilities in rural areas, demonstrating the importance of precision, sustainability, and animal welfare in modern livestock farming. The designed system combines low-cost sensors, microcontrollers, wireless communication modules, and renewable energy sources to provide continuous environmental monitoring. It offers farmers immediate information and proactive alarms, allowing them to take action before adverse conditions escalate. The user-friendly interface is available to all farmers. This strategy is on real-world adaptation, including solar-powered operation and on-site notifications to accommodate uncertain agricultural communities. Remote data access improves agricultural management, particularly in places with limited staff and experience. The research stimulates a paradigm shift in animal agriculture, with an emphasis on data-driven decision-making, risk prediction, and animal welfare. As the global agricultural sector confronts climate change, disease outbreaks, and ethical food demand, such systems will become increasingly important. In conclusion, this work provides a plan for a smart monitoring device for livestock farming that aims to improve agricultural efficiency, food security, and animal welfare through sustainable innovation and stakeholder participation, with the possibility of future testing and calibration.

References

1. Shahinfar, S., Hassani, M., & Samadi, S. (2020). IoT-based livestock monitoring and management: A review. *Computers and Electronics in Agriculture, 173*, 105441.
2. Ravi, V., Pushpa, S., & Rani, S. (2021). Precision livestock monitoring using the Internet of Things (IoT). *Materials Today: Proceedings, 46*(6), 6321–6325.
3. Islam, M. S., Hossain, M. I., & Sadi, M. S. (2020). Smart cattle monitoring system based on the Internet of Things (IoT). *International Journal of Computer Applications, 176*(29), 32–36.
4. Banhazi, T. M., Lehr, H., Black, J. L., Crabtree, H., Schofield, P., Tscharke, M., & Berckmans, D. (2019). Precision livestock farming: An international review of scientific and commercial aspects. *International Journal of Agricultural and Biological Engineering, 12*(1), 1–17.
5. Mekonnen, G. A., Hoekstra, A. Y., & Becht, R. (2022). Improving livestock productivity through smart environmental monitoring. *Livestock Science, 258*, 104842.
6. Food and Agriculture Organization (FAO). (2021). Smart Agriculture for Sustainability: Innovation and Technology for Livestock and Crops. Food and Agriculture Organization of the United Nations.
7. Zhang, Y., Yu, C., & Li, J. (2021). Enhancing consumer trust through smart livestock monitoring. *Journal of Animal Science and Technology, 63*(4), 789–797.
8. Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2020). An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal, 5* (5), 3758–3773.
9. Bhatt, S., Jani, R., & Patel, A. (2022). IoT-based Smart Monitoring System for Livestock Welfare. *Journal of Internet of Things and Information Technology, 4*(2), 45–52.
10. Patel, D. K., Rani, P., & Bairwa, R. K. (2023). Design and Implementation of IoT-based Livestock Monitoring System. *Journal of Agricultural Research and Development, 7*(1), 59–68.
11. Singh, R., & Gupta, M. (2023). A low-power wireless communication framework for smart agriculture using ZigBee and LoRa. *Journal of Network and Computer Applications, 215,* 103594.