

# **IOT ARDUINO BASED REAL-TIME CATTLE MONITORING**

## **INTERDISCIPLINARY PROJECT**

Submitted in partial fulfilment of the requirements for the award of  
Bachelor of Technology in Information Technology

by

**RAMYASRI A (Reg. No - 41120005)**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
SCHOOL OF COMPUTING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**CATEGORY-1 UNIVERSITY BY UGC**

**Accredited "A++" by NAAC**

**12B Status by UGC |Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**MAY 2024**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

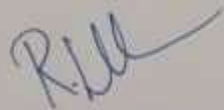
## DEPARTMENT OF INFORMATION TECHNOLOGY

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAMYASRI A (41120005)** who carried out the project entitled "**IOT BASED REAL-TIME CATTLE MONITORING SYSTEM**" under my supervision from January 2024 to May 2024.

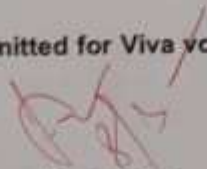
  
Internal Guide

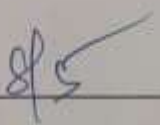
**Ms. MERLIN MARY JENITHA ,M.Tech.,**

  
Head of the Department

**Dr. R. SUBHASHINI, M.E., Ph.D.,**

Submitted for Viva voce Examination held on \_\_\_\_\_

  
Internal Examiner

  
External Examiner

## DECLARATION

I, RAMYASRI A (41120005) hereby declare that the Project Report entitled IOT BASED ARDUINO REAL-TIME CATTLE MONITORING done by me under the guidance of Ms.MERLIN MARY JENITHA, M.Tech., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 8/05/24

PLACE: CHENNAI

A. Ramyasri  
SIGNATURE OF THE CANDIDATE

## ACKNOWLEDGEMENT

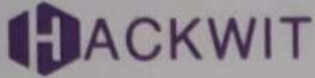
I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T. Sasikala M.E., Ph. D**, Dean, School of Computing **Dr. R. Subhashini M.E., Ph.D.**, Head of the Department of Information technology for providing me necessary support and details at the right time during the progressive

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Merlin Mary Jenitha M.Tech.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

## CERTIFICATE OF COMPLETION:



HACKWIT TECHNOLOGIES PVT LTD  
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,  
Ramapuram, Chennai - 600089.  
admin@hackwittechnologies.com  
www.hackwittechnologies.com

10.02.2024

### CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Addanki Ramyasri (Reg No: 41120005)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22<sup>nd</sup> January to 30<sup>th</sup> January 2024.

During the internship, **Addanki Ramyasri** has closely worked as a part of the **Internet of Things** by carrying out the project entitled "**IOT-Based Arduino-Based Real-Time Cattle Monitoring**".

She demonstrated good developing and learning skill with self-motivated attitude to learn new things. Her contribution to implement the Internet of Things was good. We wish her all the best for her future endeavors.

Best Regards,

Ravikumar S  
CEO & Managing Director  
Hackwit Technologies Pvt Ltd



## CERTIFICATE OF COMPLETION :



## **ABSTRACT**

The livestock industry stands to benefit significantly from technological advancements, particularly through the integration of Internet of Things (IoT) devices. This abstract presents an IoT-based Arduino real-time cattle monitoring system designed to enhance cattle management practices. The proposed system employs Arduino microcontrollers and various sensors to collect real-time data on key parameters such as body temperature, heart rate, location, and activity levels of cattle. This data is transmitted wirelessly to a centralized server using communication protocols such as Wi-Fi or LoRaWAN through a user-friendly interface, farmers and ranchers can access this data remotely via web or mobile applications. Advanced analytics algorithms process the collected data to provide insights into the health, behavior, and well-being of individual animals and the herd as a whole. Overall, the IoT-based Arduino real-time cattle monitoring system offers a cost-effective, scalable, and efficient solution for improving livestock management. It also offers several benefits, including early detection of health issues, timely intervention, and optimized resource allocation. Moreover, it provides valuable insights into cattle behavior and environmental conditions, aiding in decision-making processes for farmers and ranchers. enhancing animal welfare, and increasing productivity in the agricultural sector.

## TABLE OF CONTENTS

| CHAPTER NO | TITLE                             | PAGE NO  |
|------------|-----------------------------------|----------|
|            | ABSTRACT                          | vii      |
|            | LIST OF FIGURES                   | x        |
|            | LIST OF ABBREVIATIONS             | xi       |
| <b>1</b>   | <b>INTRODUCTION</b>               | <b>1</b> |
|            | 1.1 IOT                           | 2        |
|            | 1.2 PROBLEM STATEMENT             | 2        |
|            | 1.3 OBJECTIVES                    | 3        |
| <b>2</b>   | <b>METHODOLOGY</b>                | <b>4</b> |
|            | 2.1 AIM OF THE PROJECT            | 4        |
|            | 2.2 SCOPE OF THE PROJECT          | 4        |
|            | 2.3 HARDWARE/SOFTWARE REQUIREMENT | 5        |
|            | 2.3.1 WINDOWS/MAC/LINUX SYSTEM    | 6        |
|            | 2.3.2 ARDUINO IDE                 | 6        |
|            | 2.3.3 ARDUINO UNO                 | 6        |
|            | 2.3.4 PULSE SENSOR                | 7        |
|            | 2.3.5 TEMPERATURE SENSOR          | 7        |
|            | 2.3.6 BREADBOARD                  | 7        |
|            | 2.3.7 RESISTORS                   | 8        |
|            | 2.3.8 LCD                         | 8        |
|            | 2.3.9 LED                         | 9        |
|            | 2.3.10 ESP8266                    | 9        |
|            | 2.3.11 JUMPER WIRES               | 9        |
|            | 2.3.12 BUZZER                     | 10       |
|            | 2.3.13 I2C                        | 10       |
|            | 2.4 LIBRARIES USED                | 11       |

|          |     |                                   |           |
|----------|-----|-----------------------------------|-----------|
|          | 2.5 | MODULES                           | 12        |
|          | 2.6 | IMPLEMENTATION                    | 14        |
| <b>3</b> |     | <b>RESULTS AND DISCUSSION</b>     | <b>16</b> |
|          | 3.1 | TESTS                             | 16        |
|          | 3.2 | RESULTS                           | 18        |
| <b>4</b> |     | <b>CONCLUSION AND FUTURE WORK</b> | <b>20</b> |
|          | 4.1 | CONCLUSIONS                       | 20        |
|          | 4.2 | FUTURE WORK                       | 20        |
|          |     | <b>REFERENCES</b>                 | <b>21</b> |
|          |     | <b>APPENDICES</b>                 | <b>24</b> |
|          | A.  | SOURCE CODE                       | 24        |
|          | B.  | SCREENSHOTS                       | 27        |

## LIST OF FIGURES

| FIGURE NO | NAME                 | PAGE NO |
|-----------|----------------------|---------|
| 2.1       | ARDUINO UNO          | 6       |
| 2.2       | TEMPERATURE SENSOR   | 7       |
| 2.3       | BREADBOARD           | 7       |
| 2.4       | RESISTOR             | 8       |
| 2.5       | LCD                  | 8       |
| 2.6       | LED                  | 8       |
| 2.7       | ESP8266              | 9       |
| 2.8       | JUMPER WIRES         | 9       |
| 2.9       | BUZZER               | 10      |
| 2.10      | ARCHITECTURE DIAGRAM | 13      |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                          |
|--------------|------------------------------------|
| LCD          | Liquid Crystal Display             |
| IDE          | Integrated Development Environment |
| ESP          | Event Stream Processing            |
| MQTT         | Message Query Telemetry Transport  |
| IOT          | Internet Of Things                 |
| TLS          | Transport Layer Security           |
| EMR          | Electronic Medical Records         |
| API          | Application Programming Interface  |
| SSL          | Secure Socket Layer                |
| LED          | Light Emitting Diode               |
| I2C          | Inter Integrated Circuit           |

# **CHAPTER 1**

## **INTRODUCTION**

We can say that the flow of animal lovers has gradually increased by the number of animal hospitals which proliferate every year. There are so many benefits of having a pet including companionship guarding the house friendship as well as a part of a family member. Pet owners now a days are very close to their pets and treat them as family members. That means health care or medical treatment is very crucial.

One of the problems is caused by veterinarians or staff in health care industry is not being able to take care of the animals 24/7 (24 hours a day, 7 days a week). In general, hospitals do not have any data monitor and alert systems which tracks how bad of each animal's case is when there are at the hospital. This is a major cause for sick animals to hinder recuperation with many likely to lose their lives. In general, several information is required to be able to identify the health of an animal. This information includes vital signs, pulse, breathing rate, temperature, blood pressure, etc. This is used in a health surveillance system, as such data is a data indication of life.

If data abnormalities occur, that may show that the vital signs of the animal may be dangerous. Therefore, a Web application that can monitor and record data from sick animals will play an important role. This application will focus on dogs and cats to start. Therefore, the researchers propose developing a web application to keep track of and monitor the heart rate and temperature of sick animals. It is primarily focused on recording heart rates and temperature. If the heart rate is abnormal, it can be assumed that the animal is at risk [2]. This system can control and monitor sick animal's symptoms from diagrams that show recorded data within the system. They also can report data to veterinarians or hospital staff. When abnormal data is reported, the animals can be assisted in time which might be able to save their life in time [6]. This research focuses on tracking and monitoring the heart rate and temperatures of sick animals in crisis or animal patients requiring recovery after preliminary treatment.

Dairy cattle are homoeothermic and necessary to maintain continuous body temperature, respiration, humidity, heart beat and rumination. The regular temperature of cow is 38.5-39.5°C. When the temperature is below 38.5-39.5°C the diseases arise are indigestion, milk infection, etc. distances. GSM and ZigBee are widely used wireless technologies as well. and when the temperature is above 41°C the diseases arises are influenza and anthrax. When the temperature of the animal is very high on that time it may die. Humidity can reduce heat exchange and have enervating impact on the cattle. When the stress will be more on that time milk quality will reduce. So using this technique we can help dairy ranchers to improve milk profit, quality and it will reduce the infection stress on the dairy herd and provide great level of animal security.

A wireless sensor network (WSN) is a system obtained by a huge amount of sensor nodes where every node is armed with a sensor to identify physical sensations such as temperature, stress, light etc. The sensor node is key part of a WSN. The sensor node contains hardware part which includes four sections: power and power administration unit, a microcontroller, a sensor and wireless transceiver. The sensor is the link of a wireless sensor network node which will give the atmosphere and tool status. The sensors are used to collect and transmit the signs, such as sensations, light and natural signs and then transfer it to the microcontroller. In this paper Arduino UNO receives the content from the sensor and development the content accordingly. The ESP 8266 Wi-Fi module will transfer the content, so that the physical accomplishment of interaction can be achieved..

## **1.1 IOT**

It refers to the network of interconnected devices that can communicate and exchange data with each other over the internet without requiring human intervention. These devices can range from everyday objects like household appliances, wearable devices, and industrial machines to vehicles and infrastructure components. IoT enables these devices to collect, analyze, and transmit data, leading to improved efficiency, automation, and decision-making in various domains such as healthcare, transportation, agriculture, and smart cities.

## **1.2 Problem Statement**

Despite advancements in livestock management, traditional methods of monitoring cattle often fall short in providing real-time insights into their health, behavior, and location. This lack of real-time monitoring can lead to inefficiencies in livestock management, decreased productivity, and compromised animal welfare. The challenge lies in developing an IoT-based solution that enables continuous and real-time monitoring of cattle, allowing farmers to remotely track their livestock's vital signs, activity levels, and location. Such a system should be capable of collecting and analyzing data in real-time, providing actionable insights to farmers for timely intervention and decision-making.

## **1.3 Objectives**

To Enable continuous monitoring of vital signs and health parameters such as heart rate, blood pressure, temperature, etc., without the need for frequent hospital visits .To Detect early signs of health deterioration or potential medical emergencies through real-time data analysis, allowing for timely intervention and treatment .To Facilitate remote patient monitoring, especially for individuals with chronic illnesses or those who require regular medical attention, enabling healthcare providers to remotely assess their condition and adjust treatment plans as needed. To Provide personalized healthcare insights and recommendations based on individual health data collected over time, tailored to each person's unique health profile and needs.To Reduce healthcare costs associated with hospital visits, emergency room admissions, and unnecessary medical procedures by shifting towards preventive and remote healthcare monitoring models.

## CHAPTER 2

### METHODOLOGY

#### 2.1 AIM OF THE PROJECT

The aim of the IoT real-time cattle monitoring project is to develop a comprehensive system that enables continuous and remote monitoring of cattle, leveraging IoT technologies to collect, analyze, and visualize data in real-time. This system aims to provide farmers with actionable insights into the health, behavior, and location of their livestock, ultimately improving livestock management practices and enhancing productivity, efficiency, and animal welfare.

#### 2.2 SCOPE OF THE PROJECT

Scope of IoT-Based real-time cattle Monitoring System:

- **Hardware Development:** Designing and selecting appropriate sensors and IoT devices for collecting data on vital signs (e.g., temperature, heart rate), activity levels, and location of cattle. This may include wearable devices, GPS trackers, and environmental sensors.
- **Software Development:** Developing the software infrastructure for data acquisition, transmission, storage, and analysis. This involves designing algorithms for real-time data processing, anomaly detection, and generating actionable insights for farmers.
- **Integration of IoT Technologies:** Integrating various IoT technologies such as wireless communication protocols (e.g., Bluetooth, LoRa, NB-IoT) for data transmission, cloud computing platforms for data storage and analysis, and mobile/web applications for user interface and visualization.
- **User Interface Design:** Creating user-friendly interfaces for farmers to access real-time data, receive alerts, and monitor the health, behavior, and location of individual cattle or the entire herd. The interface should provide visualization tools for data analysis and decision support.
- **Security and Privacy:** Implementing robust security measures to protect data integrity, confidentiality, and privacy. This includes encryption techniques, access control mechanisms, and secure authentication protocols to prevent unauthorized access to sensitive information.

- **Field Testing and Validation:** Conducting field trials and validation studies to assess the performance, reliability, and accuracy of the system in real-world farming environments. This involves collaborating with farmers and livestock experts to gather feedback and iterate on system improvements.
- **Scalability and Adaptability:** Designing the system to be scalable and adaptable to accommodate varying herd sizes, environmental conditions, and geographic locations. This may involve modular design principles and the flexibility to add or remove sensors as needed.
- **Regulatory Compliance:** Ensuring compliance with relevant regulatory standards and guidelines for livestock monitoring and data privacy (e.g., GDPR, HIPAA). This includes obtaining necessary approvals and certifications for deploying the system in agricultural settings.

By addressing these aspects within the scope of the project, the aim is to develop a comprehensive IoT-based solution for real-time cattle monitoring that meets the needs of farmers, improves livestock management practices, and enhances overall productivity and animal welfare

## 2.3 HARDWARE/SOFTWARE REQUIREMENT

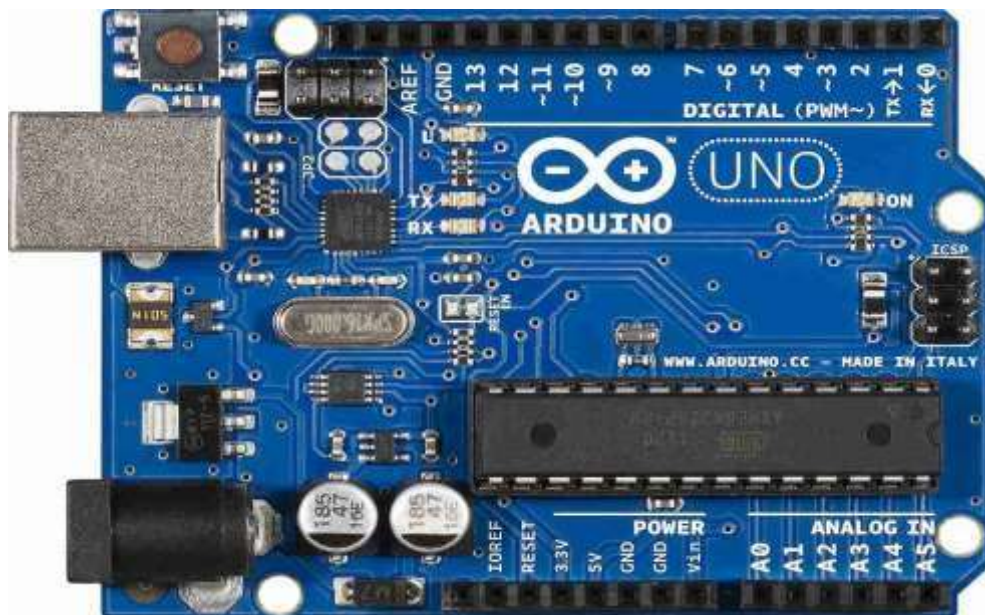
- Windows/Mac/Linux Based System
- Arduino IDE
- Arduino UNO
- Temperature Sensor
- Breadboard
- LED
- ESP8266
- Jumper Wires
- Ultrasonic Sensor

### **2.3.1 Windows/Mac/Linux System**

Windows, Mac or Linux based operating system for personal computers or any other internet of things devices. The system requirements for the Arduino IDE are relatively minimal, making it compatible with various operating systems.

### **2.3.2 Arduino IDE**

The Arduino software, also known as the Arduino IDE (Integrated Development Environment), is a platform used for writing, compiling, and uploading code to Arduino microcontroller-based boards. The Arduino software provides a user-friendly and accessible platform for programming Arduino boards, making it suitable for beginners and experienced users alike to create a wide range of embedded projects.



**Fig: 2.1 ARDUINO UNO**

### **2.3.3 Ultrasonic Sensor**

Ultrasonic sensors can be placed in cattle housing facilities or feeding areas to monitor the movement and activity levels of individual cattle or the herd as a whole. Changes in activity patterns can indicate health issues, such as lameness or discomfort. By installing ultrasonic sensors in feed troughs or waterers, the system can track the frequency and duration of feed and water consumption by each animal. This data can help farmers ensure that all cattle are receiving adequate nutrition and hydration. It can be used to measure the respiration rate of cattle, providing insights into their respiratory health. An abnormal respiratory rate may indicate respiratory infections or other health issues that require attention.



**Fig: 2.2 ULTRASONIC SENSOR**

### **2.3.4 Temperature Sensor**

When selecting a temperature sensor for real time cattle monitoring, factors such as accuracy, response time, comfort, and hygiene should be considered. Additionally, the sensor's compatibility with the monitoring system and its ease of integration should also be taken into account. Ultimately, the choice of temperature sensor will depend on the specific requirements of the monitoring application and the preferences of healthcare professionals.



**Fig: 2.3 TEMPERATURE SENSOR**

### **2.3.5 Breadboard**

A breadboard is a versatile and widely used tool in electronics prototyping and circuit design. It allows electronic components to be easily connected and tested without the need for soldering. Breadboards are essential tools for electronics hobbyists, students, and professionals alike, providing a convenient platform for experimenting with circuit designs and quickly bringing ideas to life.



***Fig: 2.4 BREADBOARD***

### **2.3.6 Resistors**

Resistors are passive electronic components that impede the flow of electric current. They are widely used in electronic circuits to control the amount of current flowing through various parts of the circuit, adjust signal levels, and divide voltages.



***Fig: 2.5 RESISTOR***

### **2.3.7 LED**

LED lights are a type of lighting that has gained immense popularity in recent years due to their energy efficiency, longevity, and versatility. Unlike traditional incandescent bulbs, which use a filament to produce light, LEDs produce light through the movement of electrons in a semiconductor material. In scenarios where monitoring takes place during nighttime or low-light conditions, LEDs can be used to provide illumination for cameras or other sensors. This ensures that monitoring can continue effectively even in the absence of natural light.



**Fig: 2.6 LED**

### **2.3.8 ESP8266**

The ESP8266 is a highly popular and versatile Wi-Fi module developed by Espressif Systems. It is widely used in IoT (Internet of Things) and embedded systems projects due to its low cost, low power consumption, and built-in Wi-Fi capabilities.



**Fig: 2.7 ESP8266**

### **2.3.9 Jumper Wires**

Jumper wires are electrical wires with connectors at each end, typically used to create temporary connections between electronic components on a breadboard or between a breadboard and other components. Jumper wires are essential tools for electronics prototyping and experimentation, providing a flexible and convenient way to create temporary connections between components in breadboard-based circuits.



**Fig 2.8 JUMPER WIRES**

### **2.3.10 Buzzer**

A buzzer is an electromechanical device that produces a buzzing or buzzing sound when an electric current passes through it. It typically consists of a vibrating reed or membrane, an electromagnetic coil, and a housing to amplify and project the sound. Buzzers play a crucial role in alerting users to important events and conditions in electronic devices and systems, contributing to their usability and safety.



**Fig 2.9 BUZZER**

## 2.4 LIBRARIES USED

For an IOT-based real-time cattle monitoring system, several libraries can be useful depending on the specific requirements of the project. Here are some common libraries used in developing IoT-based health monitoring systems:

- **MQTT Libraries:** MQTT is a lightweight messaging protocol commonly used in IoT applications for communication between devices and servers. Libraries such as PubSubClient for Arduino or Paho MQTT for Python provide MQTT client implementations that enable devices to publish sensor data and subscribe to commands or updates from the server.
- **WiFi Libraries:** WiFi libraries are essential for connecting IoT devices to a wireless network. For Arduino-based projects, libraries such as ESP8266WiFi or WiFi101 are commonly used to establish WiFi connections and configure network settings.
- **Web Server Libraries:** Web server libraries enable IoT devices to host web pages or APIs for remote monitoring and control. Libraries such as ESP8266WebServer or Flask for Python facilitate the creation of web interfaces where users can view health data, configure settings, or receive alerts.
- **Sensor Libraries:** Depending on the sensors used for health monitoring (e.g., heart rate sensors, temperature sensors), specific sensor libraries may be required to interface with the sensors and collect data. Many sensor manufacturers provide Arduino or Python libraries for their products, simplifying the integration process.
- **Data Visualization Libraries:** Libraries such as Chart.js, Plotly, or Matplotlib can be used to visualize health data in real-time or generate historical charts and graphs for analysis. These libraries enable users to gain insights from the collected data and track trends over time.

- **Database Libraries:** If data logging or storage is required, database libraries such as SQLite (for local storage) or Firebase Realtime Database (for cloud storage) can be used to store and retrieve health data. These libraries provide functionalities for data management, querying, and synchronization between devices.
- **Security Libraries:** Security libraries such as Crypto Arduino or Python's hash lib module can be used to implement encryption and hashing algorithms for securing sensitive health data during transmission or storage. Additionally, libraries like TLS for Arduino or Python's SSL module provide secure communication over HTTPS.
- **Notification Libraries:** Libraries such as Twilio, Push bullet, or Telegram Bot can be integrated to send notifications or alerts to users' smartphones or other devices based on predefined conditions (e.g., abnormal health readings, medication reminders).

## 2.5 MODULES

A microcontroller or microprocessor to process the data, and a communication module to transmit the data to a remote server or cloud platform for further analysis and storage. Here's a simplified circuit diagram explanation for such a system:

**Arduino:** The Arduino Server as the central unit in this setup. It's a microcontroller board that can be programmed to interact with various sensors and devices. It processes from sensors and control output to other components based on the programmed logic.

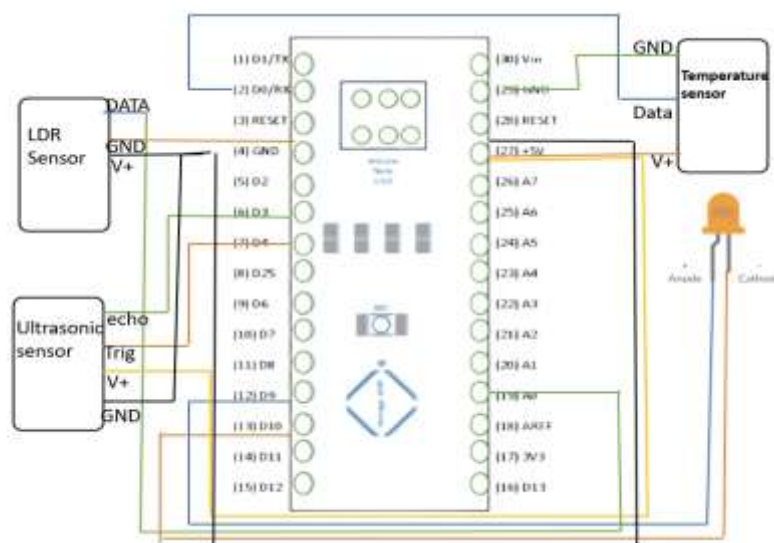
**LDR (Light dependent resistor):** The LDR is a sensor that detects light intensity. It changes its resistance based on the amount of the light falling on it. In the architecture diagram, the LDR is connected to one of the analog input pins of,

**Arduino Buzzer:** A buzzer is an electronic device that produces sound when an electric current passes through it. In your project, it can be used to create audible alerts or alarms.

**Temperature sensor:** This sensor measures the ambient temperature of its surroundings. There are various temperature sensors available for Arduino projects, such as the DHT11 or DHT22. These sensors can measure temperature and humidity.

**Breadboard:** This provides a platform for prototyping circuits without soldering. It allows you to easily connect and disconnect and connecting multiple components without soldering.

**Ultrasonic sensor:** It emits ultrasonic waves and measures the time taken for the waves to bounce back. This helps in determine the distance of an object connect the ultrasonic sensor's trigger and echo pins to digital pins on the Arduino. Connect the LDR to an analog pin. connect the buzzer to a digital pin connect the temperature sensor to the appropriate pins .The Arduino board will then read data from the sensors, process it according to the programmed logic, and activate the buzzer or perform other actions.



**Fig: 2.10 Circuit Diagram**

## 2.6 IMPLEMENTATION

Implementing a health monitoring system using IoT involves several steps, including hardware setup, data collection, transmission, storage, and visualization. Here's a general overview of how you can implement such a system:

### 1. *Define Requirements:*

Clearly define the requirements of your project, including the parameters you want to monitor (e.g., temperature, humidity, activity levels), the hardware components you'll need, and the desired features of the monitoring system

### 2. *Select Hardware Components:*

Choose the appropriate hardware components for your project, such as microcontrollers (e.g., Arduino, Raspberry Pi), sensors (e.g., temperature sensors, motion sensors), communication modules (e.g., Wi-Fi, Bluetooth), and power sources (e.g., batteries, solar panels).

### 3. *Hardware Setup:*

Assemble the hardware components according to your project requirements. Connect sensors to the microcontroller or single-board computer, ensure proper power supply, and configure communication modules for data transmission

### 4. *Software Development:*

**Firmware Development:** Write firmware code to interface with sensors, collect data, and transmit it to the cloud or a local server. Use appropriate programming languages (e.g., C/C++, Python) and libraries to interact with hardware components.

**Data Processing:** Develop algorithms for real-time data processing, analysis, and anomaly detection. Implement filtering techniques, statistical methods, or machine learning algorithms to extract useful insights from sensor data.

**Communication Protocols:** Implement communication protocols such as MQTT, HTTP, or CoAP to establish a connection between IoT devices and the cloud or server for data transmission.

### **5. *Cloud Integration:***

Set up a cloud platform (e.g., AWS IoT, Azure IoT Hub, Google Cloud IoT Core) to receive and store sensor data, perform further processing, and enable remote monitoring and control. Configure IoT devices to connect to the cloud platform securely using appropriate authentication mechanisms.

### **6. *Data Visualization and User Interface:***

Develop a user-friendly dashboard or web interface for visualizing real-time sensor data, receiving alerts, and monitoring cattle health and behavior. Use libraries such as Matplotlib, Plotly, or D3.js for data visualization, and integrate with mapping APIs for geolocation functionality if needed.

### **7. *Testing and Validation:***

Conduct thorough testing of the monitoring system to ensure hardware reliability, data accuracy, and software functionality. Test different scenarios and edge cases to validate the system's performance under various conditions. Gather feedback from stakeholders and make necessary adjustments or improvements.

### **8. *Deployment:***

Deploy the monitoring system in the target environment (e.g., cattle farm) and ensure proper installation, calibration, and configuration of hardware components. Provide training to end-users on how to use the system effectively and troubleshoot common issues.

### **9. *Maintenance and Support:***

Establish a maintenance plan to regularly monitor system performance, address any hardware or software issues, and apply updates or patches as needed. Provide ongoing support to end-users and continuously evaluate the system's effectiveness in meeting their needs.

## CHAPTER 3

### RESULTS AND DISCUSSION

#### 3.1 TESTS

Testing a real-time cattle monitoring system using IoT involves verifying various aspects of the system, including hardware functionality, sensor accuracy, data transmission reliability, and user interface usability. Here's a suggested approach for testing such a system:

##### 1. Hardware Testing:

- Test the integration of sensors, microcontrollers (e.g., Arduino, Raspberry Pi), and communication modules to ensure proper connectivity and data exchange between components.
- Test each sensor individually to verify its accuracy and reliability in measuring vital signs
- Verify that the hardware setup is properly configured and wired according to the system design specifications.

##### 2. Communication Testing:

- Test the communication between the IoT device and the cloud server or gateway to ensure reliable data transmission.
- Verify that data packets are sent and received correctly without loss or corruption.
- Test the system's resilience to network disruptions, packet loss, and latency by simulating various network conditions.

##### 3. Data Validation:

- Validate the accuracy of the sensor data by comparing it against reference measurements obtained from certified medical devices or clinical standards.
- Perform data integrity checks to ensure that sensor readings are within acceptable ranges and free from errors or anomalies

#### **4. Functional Testing:**

- Test the functionality of the system features, such as real-time monitoring, data visualization, alerting, and user interaction.
- Verify that users can access the system interface, view health data, set thresholds, and receive alerts as intended.
- Test edge cases and error scenarios to ensure that the system behaves predictably and gracefully handles exceptions.

#### **5. Security Testing:**

- Perform security testing to identify and mitigate vulnerabilities in the system, including unauthorized access, data breaches, and privacy risks.
- Test the strength of encryption mechanisms used to secure data transmission and storage.
- Verify that user authentication, access control, and encryption protocols are implemented correctly and effectively.

#### **6. Usability Testing:**

- Conduct usability testing with end-users to evaluate the user experience and interface design.
- Gather feedback on the ease of use, clarity of information, responsiveness, and overall satisfaction with the system.
- Use surveys, interviews, or user observation sessions to identify usability issues and areas for improvement.

#### **7. Performance Testing:**

- Measure the performance of the system in terms of data throughput, response time, and resource utilization.
- Test the scalability of the system by simulating a large number of concurrent users or devices.
- Identify any bottlenecks or performance limitations and optimize the system architecture and algorithms accordingly.

#### **8. Integration Testing:**

- Test the integration of the health monitoring system with external

services, such as healthcare providers, electronic medical records (EMR) systems, or third-party applications.

- Verify that data exchange, interoperability, and compatibility with external systems are seamless and reliable.

#### **9. Regression Testing:**

- Perform regression testing to ensure that new updates or changes to the system do not introduce regressions or unintended side effects.
- Re-run previous test cases and verify that existing functionality remains intact after modifications to the system.

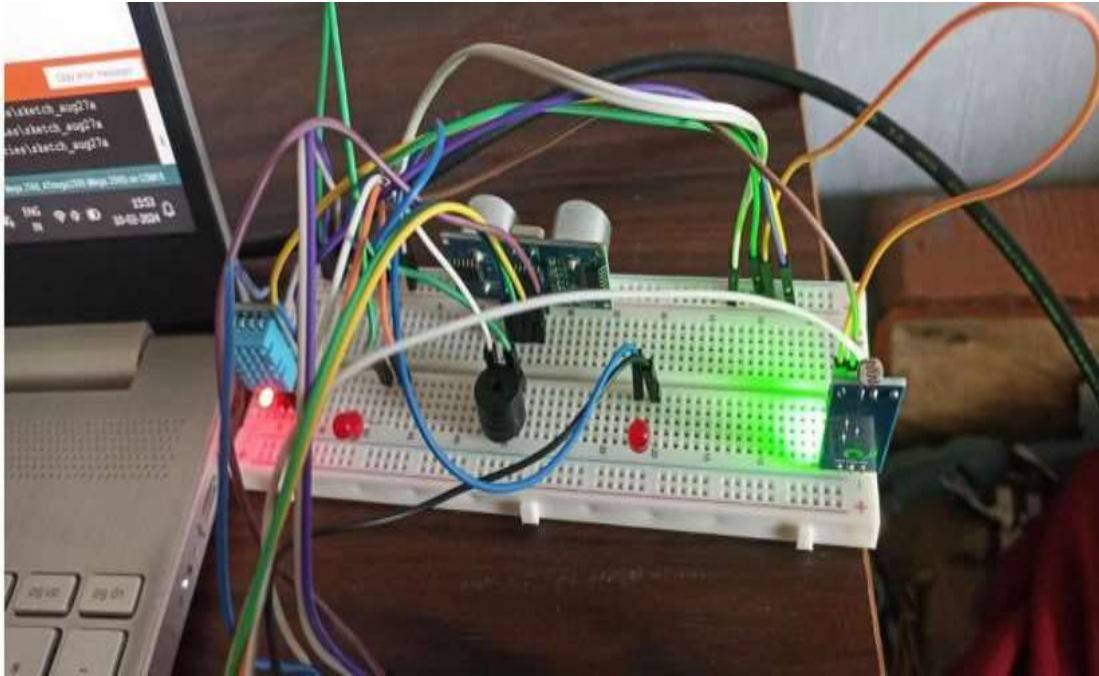
#### **10. Documentation and Reporting:**

- Document test cases, test results, and any issues or defects discovered during testing.
- Prepare a comprehensive test report summarizing the testing process, outcomes, and recommendations for improvement.

### **3.2 Results**

The results of a real-time cattle monitoring system using IoT can vary depending on various factors, including the design of the system, the accuracy of the sensors, the reliability of the communication network, and the effectiveness of data processing and analysis algorithms. Here are some potential results and outcomes of implementing such a system:

**Early Disease Detection.** Utilize real time monitoring of vital signs and behavior patterns to detect early signs of illness or disease in individual cattle. Abnormalities such as changes in temperature, heart rate, activity levels, or feeding behavior can indicate potential health issues. Implement machine learning algorithms to analyze sensor data and identify patterns associated with specific diseases or health conditions. Train the algorithms using historical health data and expert knowledge to improve accuracy in disease detection



***FIG.3.1 OUTPUT OF THE MODEL***

This report outlines the findings of a trial conducted in a residential community to assess the effectiveness of an IoT-based real-time cattle monitoring system. By employing advanced sensor technologies and IoT communication systems, the project aimed to enhance livestock management practices. The integration of various sensors, including those for temperature changes and ultrasonic sensor to emit the ultrasonic waves with the Arduino Uno microcontroller or the microchip which has enabled accurate monitoring of cattle health and behavior. Real-time data transmission via GSM or Wi-Fi communication modules facilitated remote monitoring and timely alerts for stakeholders. The trial demonstrated the system's potential to improve animal welfare and operational efficiency in cattle farming and their managing thereby mitigating potential health issues by detecting changes in the temperatures around them and optimizing the resource allocation.

## CHAPTER 4

### CONCLUSION AND FUTURE WORK

#### 4.1 CONCLUSION

The implementation of IoT-based Arduino real-time cattle monitoring system presents a promising solution for modernizing livestock management practices. Through the integration of sensors, data analytics, and real-time communication, this system offers farmers unprecedented insights into the health, behavior, and location of their cattle. By harnessing the power of IoT, farmers can enhance their decision-making processes, optimize resource utilization, and improve overall herd management efficiency. The real-time monitoring capabilities enable timely interventions, leading to better health outcomes and increased productivity.

#### 4.2 FUTURE WORK

**Predictive Analytics:** Develop predictive analytics models that leverage historical data and real-time sensor readings to forecast potential health issues or predict optimal breeding and feeding times. These predictive capabilities could empower farmers to make data-driven decisions and optimize management practices for improved cattle productivity and welfare.

**Mobile Application Development:** Create a user-friendly mobile application that enables farmers and ranchers to remotely monitor their cattle's health and location, receive alerts for critical events, and access historical data and analytics insights. Incorporating features such as geofencing and push notifications could enhance the usability and accessibility of the monitoring system.

**Integration with Agricultural Systems:** Explore opportunities to integrate the cattle monitoring system with existing agricultural management systems, such as herd management software or farm automation platforms. Seamless integration would enable holistic farm management and streamline data sharing and analysis across different aspects of agricultural operations.

**Scalability and Deployment Optimization:** Investigate strategies to optimize the scalability and deployment of the monitoring system, particularly for large-scale cattle operations or geographically dispersed ranches. This may involve exploring low-power communication protocols, modular hardware designs,

## REFERENCES

1. Alzubaidi, L., & Balasundram, S. K. (2020). Internet of Things (IoT) Applications In Agriculture: A Systematic Literature Review. *Computers and Electronics in Agriculture*, 177, 105693. doi:10.1016/j.compag.2020.105693
2. Al-Saffar, A. A., & Hasan, M. K. (2018). IoT Based Smart Agriculture: A Review. In 2018 4th International Conference on Computer and Technology Applications (ICCTA) (pp. 178–183). IEEE. doi:10.1109/iccta.2018.8397011
3. Yaqoob, I., Hashem, I. A. T., Ahmed, E., Kazmi, S. A., Guizani, N., & Hong, C. S. (2019). Internet of Things Forensics: Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Access*, 7, 160541–160561. doi:10.1109/access.2019.2940635
4. Zulkefli, N. A., Abdullah, N. N., Razak, M. F. A., & Arshad, M. R. (2018). Development of a Cattle Monitoring System Based on Internet of Things (IoT). In 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech) (pp. 1–4). IEEE. doi:10.23919/splitech.2018.8441169
5. Vlăduțu, A., & Drăgoicea, M. (2020). IoT-Based Animal Health Monitoring System. In 2020 43rd International Spring Seminar on Electronics Technology (ISSE) (pp. 1–6). IEEE. doi:10.1109/isse50213.2020.9122126
6. Han, J., Lee, S., Kim, H., & Kim, S. (2019). IoT-Based Cattle Health Monitoring System. In Proceedings of the 2019 5th International Conference on Information Management (ICIM) (pp. 180–184). ACM. doi:10.1145/3358760.3358789
7. Gebremedhin, K., Gebremedhin, A., Gebrelibanos, M., & Teweldemedhin, M. (2018). Design and Implementation of IoT-Based Livestock Health Monitoring System. In 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD) (pp. 1–6). IEEE. doi:10.1109/icabcd.2018.8629688

8. Kumar, M., Manhas, M., & Kaur, A. (2018). IoT Based Smart Healthcare Monitoring System. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–6). IEEE. doi:10.1109/icccnt.2018.8494179
  
9. Dar, M. A., & Khursheed, M. (2019). Cattle Health Monitoring System using Internet of Things. In 2019 2nd International Conference on Advanced Computational and Communication Paradigms (ICACCP) (pp. 1–5). IEEE. doi:10.1109/icaccp.2019.8885721
  
10. Wu, Y., Cheng, Y., Zhang, Y., & Liu, J. (2017). Design of a Low-Cost IoT System for Cattle Health Monitoring. In 2017 29th Chinese Control And Decision Conference (CCDC) (pp. 3894–3898). IEEE. doi:10.1109/ccdc.2017.7978873
  
11. Jiang, S., Liu, W., Li, H., & Chen, G. (2019). An IoT-Based Smart Cattle Farm System. In 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 233–236). IEEE. doi:10.1109/icccbd.2019.00051
  
12. Yan, Z., Jin, D., & Vasilakos, A. V. (2014). A Survey on Trust Management for Internet of Things. *Journal of Network and Computer Applications*, 42, 120–134. doi:10.1016/j.jnca.2014.01.015
  
13. [13] Atefi, A., Jalili, M., & Lotfi, S. (2017). A Real-Time IoT-Based Livestock Monitoring System Using Machine Learning Algorithms. In 2017 4th International Conference on Computer and Robotics (ICCR) (pp. 8–13). IEEE. doi:10.1109/iccr.2017.35
  
14. [14] Akmal, A., & Ahmed, S. (2018). An IoT-Based Smart Cattle Monitoring System. In 2018 2nd International Conference on Computing and Communication Systems (I3CS) (pp. 84–89). IEEE. doi:10.1109/i3cs.2018.8522280

15. Mahmoud, A., & Abu-Ein, H. (2019). IoT-Based Smart Agriculture: Greenhouse Monitoring and Controlling System. In 2019 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1–6). IEEE. doi:10.1109/hnicem.2019.8672144
16. Patil, P. S., & Doshi, K. M. (2017). IoT Based Smart Agriculture Monitoring System. In 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 784–787). IEEE. doi:10.1109/icicct1.2017.7975230
17. Patel, P., & Dwivedi, V. (2017). IoT Based Smart Agriculture Monitoring and Controlling System. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1–5). IEEE. doi:10.1109/icaccs.2017.8014594
18. Shweta, D., & Patil, D. (2019). IoT Based Smart Agriculture Monitoring System. In 2019 IEEE 9th International Conference on Advanced Computing (IACC) (pp. 92–97). IEEE. doi:10.1109/iacc48016.2019.8970915
19. Singh, M., & Lohani, A. K. (2018). IoT Based Smart Agriculture Monitoring System. In 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1–6). IEEE. doi:10.1109/iot-siu.2018.8588435
20. Suryawanshi, S., & Deshmukh, V. (2019). IoT Based Smart Agriculture Monitoring and Controlling System. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1–5). IEEE. doi:10.1109/icecct.2019.8869475

## APPENDICES

### A.SAMPLE CODE

```
#include <TinyDHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11

const int trigPin = 3;
const int echoPin = 4;
const int ldrPin = A0;
const int ledPin1 = 13;
const int ledPin2 = 12;
const int buzzerPin = 9;

DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ldrPin, INPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop()
{
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration / 2) / 29.1;

float temperature =dht.readTemperature();
float humidity = dht.readHumidity();

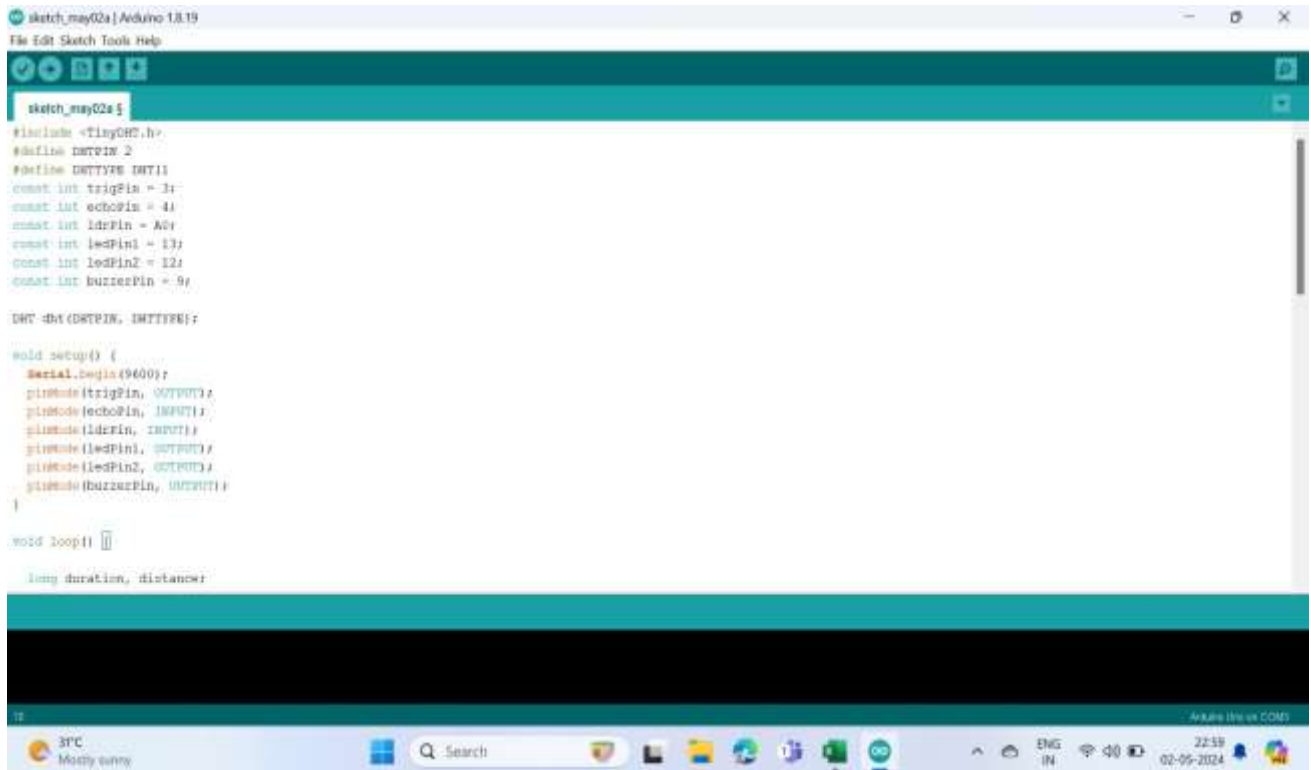
int lightIntensity = analogRead(ldrPin);

Serial.print("Distance: ");
Serial.print(distance);
Serial.print(" cm, Temperature: ");
Serial.print(temperature);
Serial.print(" °C, Humidity: ");
Serial.print(humidity);
Serial.print(" %, Light Intensity: ");
Serial.println(lightIntensity);
if (distance < 20)
{
digitalWrite(ledPin1, HIGH);
digitalWrite(ledPin2, LOW);
digitalWrite(buzzerPin, HIGH);
}

```

```
else
{
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, LOW);
digitalWrite(buzzerPin, LOW);
}
if (lightIntensity > 500)
{
digitalWrite(ledPin1, HIGH);
digitalWrite(ledPin2, HIGH);
//  tone(buzzerPin, 2000, 500);
}
delay(1000);
```

## B SCREEN SHOTS



```
sketch_may02a | Arduino 1.8.19
File Edit Sketch Tools Help

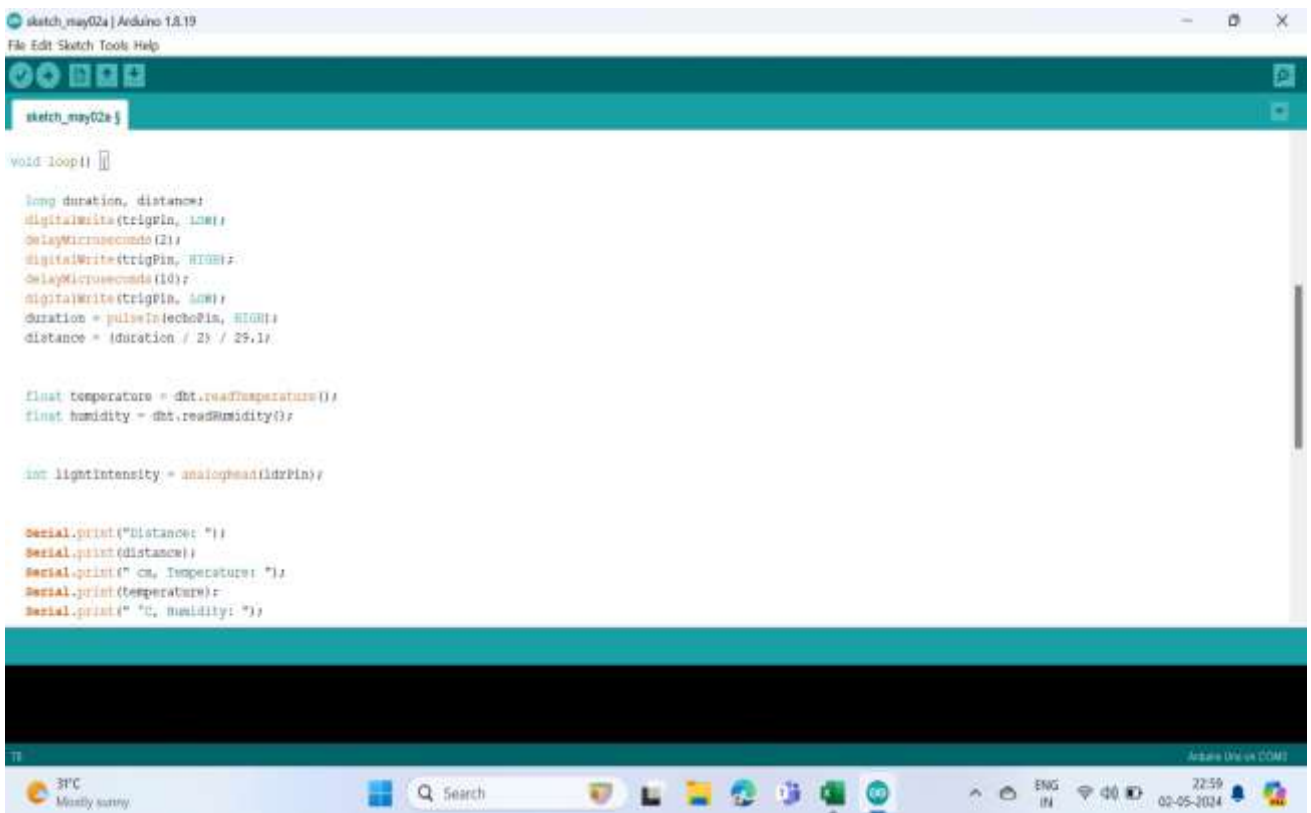
sketch_may02a |

#include <TinyDHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
const int trigPin = 2;
const int echoPin = 4;
const int ldrPin = A0;
const int ledPin1 = 13;
const int ledPin2 = 12;
const int buzzerPin = 9;

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ldrPin, INPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  long duration, distance;
```



```
sketch_may02a | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_may02a |

void loop() {

  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;

  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  int lightIntensity = analogRead(ldrPin);

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print(" cm, Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C, Humidity: ");
```

