IOT BASED WEATHER REPORTING SYSTEM

Submitted by

AJITHA M **927622BEC006**

APARNA A **927622BEC013**

DHARANI S **927622BEC033**

BACHELOR OF ENGINEERING

in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639 113

MAY 2025

M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR

BONAFIDE CERTIFICATE

Certified that this**18ECP106L - Minor Project IV** report **“IOT based weather reporting system”** is the Bonafide work of “Ajitha M(927622BEC006), Aparna A(927622BEC013), Dharani S(927622BEC033)” who carried out the project work under my supervision in the academic year 2024 - 2025 **EVEN**.

SIGNATURE
Dr.A.KAVITHA, B.E., M.E., Ph.D.,
HEAD OF THE DEPARTMENT,
Professor,
Department of Electronics and
Communication Engineering,
M.Kumarasamy College of Engineering,
Thalavapalayam,
Karur-639113.

SIGNATURE
Mr.L.RAMESH, M.E.,(Ph.D.)
SUPERVISOR,
Assistant Professor,
Department of Electronics and
Communication Engineering,
M.Kumarasamy College of Engineering,
Thalavapalayam,
Karur-639113.

This report has been submitted for the **18ECP106L – Minor Project IV** final review held at M. Kumarasamy College of Engineering, Karur on _______________________.

PROJECT COORDINATOR

INSTITUTION VISION AND MISSION

Vision

To emerge as a leader among the top institutions in the field of technical education.

Mission

M1: Produce smart technocrats with empirical knowledge who can surmount the global challenges.

M2: Create a diverse, fully -engaged, learner -centric campus environment to provide quality education to the students.

M3: Maintain mutually beneficial partnerships with our alumni, industry and professional associations

DEPARTMENT VISION, MISSION, PEO, PO AND PSO

Vision

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

Mission

M1: Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.

M2: Inculcate the students in problem solving and lifelong learning ability.

M3: Provide entrepreneurial skills and leadership qualities.

M4: Render the technical knowledge and skills of faculty members.

Program Educational Objectives

PEO1: **Core Competence:** Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering

PEO2: **Professionalism:** Graduates will provide feasible solutions for the challenging problems through comprehensive research and innovation in the allied areas of Electronics and Communication Engineering.

PEO3: **Lifelong Learning:** Graduates will contribute to the social needs through lifelong learning, practicing professional ethics and leadership quality

Program Outcomes

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1: Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.

PSO2: Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfil the industrial expectations.

Abstract	**Matching with POs,PSOs**
Humidity, Weather Monitoring, Temperature	**PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12, PSO1, PSO2**

ACKNOWLEDGEMENT

We gratefully remember our beloved **Founder Chairman, (Late) Thiru. M. Kumarasamy**, whose vision and legacy laid the foundation for our education and inspired us to successfully complete this project.

We extend our sincere thanks to **Dr. K. Ramakrishnan, Chairman**, and **Mr. K. R. Charun Kumar, Joint Secretary**, for providing excellent infrastructure and continuous support throughout our academic journey.

We are privileged to extend our heartfelt thanks to our respected Principal, **Dr. B. S. Murugan, B.Tech., M.Tech., Ph.D.,** for providing us with a conducive environment and constant encouragement to pursue this project work.

We sincerely thank **Dr. A. Kavitha, B.E., M.E., Ph.D.,** Professor and **Head, Department of Electronics and Communication Engineering**, for her continuous support, valuable guidance, and motivation throughout the course of this project.

Our special thanks and deep sense of appreciation go to our **Project Supervisor, Mr.L.Ramesh, B.E., M.E.,(P.hD)** Assistant Professor, **Department of Electronics and Communication Engineering,** for him exceptional guidance, continuous supervision, constructive suggestions, and unwavering support, all of which have been instrumental in the successful execution of this project.

We would also like to acknowledge **Mrs. D. Pushpalatha, B.E., M.E.,** Assistant Professor, **our Class Advisor,** and **Mrs. L. Kavitha, B.E., M.E.,** Assistant Professor, **our Project Coordinator**, for their constant encouragement and coordination that contributed to the smooth progress and completion of our project work.

We gratefully thank all the **faculty members of the Department of Electronics and Communication Engineering** for their timely assistance, valuable insights, and constant support during various phases of the project.

Finally, we extend our profound gratitude to our **parents and friends** for their encouragement, moral support, and motivation, without which the successful completion of this project would not have been possible.

ABSTRACT

The increasing impact of climate change and unpredictable weather patterns has made accurate and real-time weather monitoring more essential than ever. Traditional weather stations, while effective, often come with high installation and maintenance costs and may not offer real-time, localized data. To address these limitations, this project proposes an Internet of Things (IoT)-based Weather Reporting System that provides a cost-effective, scalable, and real-time solution for monitoring environmental conditions. This system integrates various sensors to measure key weather parameters such as temperature, humidity, atmospheric pressure, and rainfall. These sensors are connected to a microcontroller unit, such as an Arduino which processes the data locally and transmits it to a cloud server using communication modules like Wi-Fi or GSM. The data is then stored, visualized, and made accessible via a user-friendly web dashboard or mobile application. Users can view real-time and historical data trends, enabling better decision-making in various domains such as agriculture, urban planning, disaster management, and personal weather monitoring. The core functionality of the system lies in its ability to operate autonomously and continuously, reducing the need for manual intervention. The use of cloud technology ensures scalability and remote accessibility, allowing users from different locations to monitor weather conditions at a specific site. Data can also be analysed for patterns or anomalies, offering potential integration with AI-based predictive analytics in future developments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM		ABBREVIATION
LCAWS	-	low-cost automatic weather stations
LCD	-	Liquid Crystal diode

CHAPTER 1

INTRODUCTION

Weather plays a vital role in our daily lives, influencing agriculture, transportation, disaster preparedness, and overall human activity. Accurate and timely weather data is essential for planning and decision-making, yet many traditional weather monitoring systems are expensive, limited in coverage, and rely on manual data collection. With the rise of the Internet of Things (IoT), it is now possible to design more efficient, real-time, and cost-effective solutions. This project proposes an **IoT-based Weather Reporting System** that uses interconnected sensors and microcontrollers to monitor environmental parameters such as temperature, humidity, atmospheric pressure, and rainfall.

1. Importance of Weather Monitoring

Weather monitoring plays a crucial role in our daily lives and in the broader context of environmental sustainability, economic stability, and public safety. Accurate and timely information about weather conditions allows individuals, communities, and governments to make informed decisions that impact sectors such as agriculture, transportation, construction, disaster management, and energy production. One of the most significant areas where weather monitoring is essential is **agriculture**. Farmers rely heavily on weather data to determine the best times for planting, irrigation, fertilization, and harvesting. Knowledge of rainfall patterns, humidity levels, temperature fluctuations, and wind speed can significantly enhance crop yield and reduce the risk of crop failure. With climate change causing more erratic weather patterns, real-time monitoring is more critical than ever to adapt farming practices and safeguard food security. Another key area is disaster management and preparedness. Natural disasters such as floods, cyclones, hurricanes, and heatwaves are often preceded by specific weather patterns. Early detection and timely reporting of such conditions can save lives and reduce property damage. Weather monitoring systems can help governments and emergency services issue early warnings, organize evacuations, and plan responses effectively, thereby minimizing the impact of such events on communities.

Transportation systems—including aviation, shipping, and road transport—also depend heavily on weather monitoring. Pilots, sailors, and drivers must be informed about adverse conditions like storms, fog, or icy roads to ensure safety and prevent accidents. For instance, airports use real-time weather data to schedule flights and manage delays, while marine operations rely on accurate forecasts to avoid dangerous sea conditions. In the era of smart cities and sustainable development, weather monitoring also supports urban planning and environmental management. Monitoring air quality, rainfall, and temperature helps in designing resilient infrastructure, managing water resources, and controlling pollution. Furthermore, it contributes to the efficient operation of renewable energy systems such as solar and wind power, which are sensitive to atmospheric conditions. Additionally, weather monitoring plays a critical role in public health. High temperatures, humidity, and air quality can exacerbate conditions such as asthma, heatstroke, and cardiovascular diseases. Real-time weather alerts can help individuals take precautions and enable health authorities to plan interventions accordingly.

2. Limitations of Traditional Weather Monitoring Systems

Ttraditional weather monitoring systems have been essential tools for tracking atmospheric conditions for decades. These systems typically include large weather stations equipped with various sensors for measuring temperature, humidity, wind speed, rainfall, and atmospheric pressure. While effective in many contexts, traditional weather systems have several limitations that hinder their efficiency, accessibility, and adaptability in today's fast-paced, data-driven world. One of the primary limitations is high cost. Setting up a traditional weather station requires expensive equipment, including precision-grade sensors, towers, enclosures, and data loggers. In addition, these systems often need dedicated infrastructure such as satellite links, landlines, or wired communication systems to transmit data. This makes them impractical for deployment in remote, underdeveloped, or resource-limited areas, where accurate and localized weather data is often most needed. Another issue is the limited spatial coverage of traditional systems. National meteorological departments typically install weather stations in select urban or strategically important locations. As a result, vast rural areas, agricultural zones, and mountainous or coastal regions may not receive real-time, location-specific weather updates. This limitation reduces the accuracy of weather forecasts in regions that do not have nearby stations, impacting sectors like farming, disaster management, and local planning. Traditional systems also tend to suffer from low data frequency. Data is often collected at fixed intervals—such as every 30 minutes or hour—and sent to a central processing system for analysis. This delay can make it difficult to detect and respond to rapidly changing weather events in real-time, such as flash floods, storms, or heatwaves.

CHAPTER – 2

LITERATURE SURVEY

1. **Low-cost IoT based weather monitoring system for smart community**

   Authors: K. S. Nikhilesh, Y. H. Raghavendra, P. J. Madhu Soothanan and R. Resmi

Description

The Low-Cost IoT-Based Weather Monitoring System for Smart Communities is designed to provide affordable and real-time weather data collection using Internet of Things (IoT) technology. It integrates low-cost sensors with microcontrollers such as Arduino or ESP8266 to measure environmental parameters like temperature, humidity, atmospheric pressure, and rainfall. The collected data is transmitted wirelessly to a cloud platform, where it can be accessed via mobile or web applications. This system is particularly beneficial for agriculture, urban planning, and disaster management, as it provides localized, up-to-date weather information to support informed decision-making. Its affordability and ease of deployment make it ideal for rural and remote areas that lack access to traditional weather stations. Additionally, it can be scaled to serve larger areas or expanded to monitor more variables, supporting the goals of smart and sustainable community development. Despite its advantages, the system also has several limitations. The use of low-cost sensors often leads to reduced accuracy and reliability compared to professional-grade equipment. These sensors may require frequent calibration and maintenance, increasing long-term costs. The system's dependence on stable internet connectivity can be problematic in remote areas, limiting real-time functionality. Power supply issues also arise, particularly in off-grid areas, where battery replacement or solar integration may be necessary. Furthermore, data security and privacy concerns must be addressed, especially when weather information is stored or transmitted over the internet. While the system offers a promising low-cost solution, improvements in reliability, security, and scalability are essential for widespread and long-term deployment in smart communities.

Drawbacks

The Low-Cost IoT-Based Weather Monitoring System for Smart Communities faces several drawbacks. Firstly, sensor accuracy may be compromised due to the use of low-cost components, requiring frequent calibration. Additionally, the system relies on stable internet connectivity, which can be a challenge in remote areas. Energy consumption remains an issue, especially in off-grid locations, necessitating the use of batteries or solar power. Security concerns, including data privacy and unauthorized access, must be addressed. Moreover, maintenance requirements for sensor cleaning and replacement may increase operational costs, limiting the system's long-term viability in large-scale deployments.

2. Design and Implementation of IoT-Based Real-Time Weather Monitoring System

Author: Amit Kumar, Priya Sharma, and Dr. R. K. Verma

Description

The paper titled Design and Implementation of IoT-Based Real-Time Weather Monitoring System explores the development of a cost-effective, portable, and efficient weather reporting system using Internet of Things (IoT) technology. The authors aim to automate weather monitoring through the integration of environmental sensors with microcontrollers and cloud platforms. The system uses sensors like DHT11 (for temperature and humidity), BMP180 (for pressure), and an MQ135 gas sensor (for air quality), connected to a Node MCU microcontroller for data collection. The collected data is transmitted in real-time to cloud platforms such as Thing Speak or Blynk via Wi-Fi. This allows for real-time visualization, analysis, and storage of weather parameters. Users can access live data remotely through web or mobile applications. The paper emphasizes the advantages of the system in terms of scalability, low maintenance, and adaptability for rural and urban use.

The authors also address the importance of accurate environmental monitoring in areas like agriculture, smart cities, and disaster management. Compared to traditional weather stations, the proposed system is more accessible and customizable for specific regional requirements. Experimental results validate the accuracy of the system and confirm its practical applicability.

This study demonstrates how IoT technology can significantly improve the efficiency, cost, and reach of weather data collection and reporting, making it a viable solution for both personal and community-level environmental monitoring.

Drawbacks

While the system is effective for basic weather monitoring, it has several limitations. The use of low-cost sensors can affect data accuracy and reliability, especially under extreme weather conditions. It also lacks advanced predictive capabilities such as machine learning for forecasting. Internet connectivity is crucial for real-time updates; in remote areas with poor network coverage, this can lead to data transmission delays or losses. Additionally, the system does not support integration with broader meteorological networks, limiting its scalability and potential for large-scale data sharing and analysis.

3. High-Resolution and Secure IoT-Based Weather Station Design

Authors: Farah Nazar Ibraheem, Sura Nawfal Abdulrazzaq, Inaam Fathi, Qutaiba Ali

Description

This study presents the design of a high-resolution and secure IoT-based weather station aimed at providing detailed and accurate weather data while ensuring data security. The system incorporates advanced sensors to measure various atmospheric parameters, including temperature, humidity, pressure, and rainfall. The collected data is transmitted to a cloud platform for real-time monitoring and analysis. To enhance data accuracy and resolution, the system employs high-quality sensors and implements advanced data processing techniques. The integration of security measures ensures the protection of data from unauthorized access and tampering, addressing concerns related to data integrity and privacy.

The weather station's design focuses on scalability and flexibility, allowing for deployment in diverse environments and adaptability to different monitoring requirements. The system's architecture supports the addition of more sensors and the integration of emerging technologies, such as artificial intelligence and machine learning, to further improve forecasting capabilities. The study underscores the importance of combining high-resolution data collection with robust security protocols to create reliable and trustworthy weather monitoring systems.

Drawbacks

Despite its advanced features, the proposed weather station system faces certain limitations. The high-resolution sensors and complex data processing algorithms may increase the system's cost and power consumption, potentially limiting its affordability and suitability for remote areas.

4. **Prototyping Low-Cost Automatic Weather Stations for Natural Disaster Monitoring**

Authors: Gabriel Francisco Lorençon Ribeiro Bernardes, Rogério Ishibashi, André Aparecido de Souza Ivo, Valério Rosset, Bruno Yuji Lino Kimura

Description

This paper discusses the development of low-cost automatic weather stations (LCAWS) designed for natural disaster monitoring. The authors address the high costs associated with professional weather instruments, which often hinder large-scale deployment in disaster-prone areas. To overcome this challenge, the team utilized commercial-off-the-shelf components and open-source IoT technologies to create affordable and reliable weather stations. The LCAWS prototypes were equipped with sensors to measure parameters such as temperature, humidity, pressure, and rainfall. Data collected by these sensors were transmitted to a central system for analysis and monitoring. The study emphasizes the importance of sensor calibration to ensure data accuracy, proposing an intelligent calibration method to align the LCAWS measurements with those of professional weather stations. Experimental results demonstrated that the calibrated LCAWS provided data with no statistically significant differences compared to professional weather stations, validating their effectiveness for natural disaster monitoring. The paper highlights the potential of these low-cost systems to enhance early warning capabilities and disaster preparedness in vulnerable regions.

Drawbacks

While the LCAWS prototypes offer a cost-effective solution for weather monitoring, they are not without limitations. The reliance on commercial-off-the-shelf components may result in variability in sensor performance and durability, potentially affecting long-term reliability. The calibration process, while improving accuracy, may not fully account for all environmental factors influencing sensor readings. Additionally, the deployment of these systems in remote or harsh environments could pose logistical challenges, including maintenance and data transmission issues. The scalability of the system may also be constrained by the availability of resources and infrastructure necessary for widespread implementation

CHAPTER – 3

EXISTING SYSTEM

In the current landscape, several existing weather reporting systems utilize either traditional meteorological setups or early-stage Internet of Things (IoT) technologies. Traditional systems rely heavily on large, stationary weather stations equipped with various sensors and instruments to record data like temperature, humidity, rainfall, wind speed, and atmospheric pressure. These systems are usually operated and maintained by national meteorological departments and require manual data retrieval or semi-automated processes. Data collected is often made available to the public through official websites or media outlets but can lack real-time updates and hyperlocal accuracy. While accurate, these conventional systems are typically expensive, complex to maintain, and geographically limited. They often require substantial infrastructure, power supply, and regular maintenance. Furthermore, the frequency of data updates may not always meet the real-time needs of users in fast-changing weather conditions, especially in remote or rural areas.

To overcome these challenges, newer IoT-based systems have begun to emerge. These systems use low-cost sensors and microcontrollers, such as Arduino and ESP8266 or ESP32 boards, to create compact and more affordable weather monitoring stations. These setups are capable of collecting real-time data and transmitting it wirelessly to cloud platforms like Thing Speak, Blynk, Firebase, or AWS IoT. Users can then view and analyze the data through dashboards accessible via web or mobile applications. Most existing IoT weather systems can monitor basic weather parameters such as temperature, humidity, and light intensity. Some advanced systems also incorporate air quality sensors (e.g., MQ135), barometric pressure sensors (e.g., BMP280), and rain detection modules. These systems are often deployed by research institutes, educational institutions, weather enthusiasts, or smart farming projects. They are favored for their simplicity, affordability, and ease of deployment.

However, the existing IoT-based weather reporting systems still have limitations. Many lack accuracy and calibration when compared to professional-grade meteorological equipment. Low-cost sensors can degrade over time or become inaccurate without proper maintenance. Connectivity issues are also common in remote areas where reliable Wi-Fi or mobile data may not be available. Additionally, not all existing systems support advanced features like predictive analytics, alert generation, or full data logging and visualization. Furthermore, many existing systems operate independently without integration into larger networks, meaning they often serve local or experimental purposes rather than contributing to broader environmental monitoring systems. The lack of standardized protocols and interoperability also limits data sharing between different weather stations or government meteorological networks.

CHAPTER – 4

PROPOSED SYSTEM

The Internet of Things (IoT) has brought innovative changes in the field of environmental monitoring. An IoT-based weather reporting system uses smart sensors to measure various climatic parameters such as temperature, humidity, and atmospheric pressure. These readings are automatically transmitted to a cloud platform, eliminating manual data collection. The system provides real-time, accurate weather updates that are useful for agriculture, disaster management, and smart city applications. It enhances efficiency, reduces human effort, and ensures data accuracy. This intelligent automation supports timely decision-making and improves overall environmental awareness through continuous monitoring and data-driven insights.

1. Environmental Sensing Module

The Environmental Sensing Module is the core component of the IoT-based weather reporting system. It is responsible for collecting real-time environmental data from the surroundings using various sensors. These sensors are chosen based on the weather parameters that need to be monitored, such as temperature, humidity, and atmospheric pressure. The most commonly used sensor for measuring both temperature and humidity is the DHT11 or DHT22. These sensors are compact, low-cost, and provide reasonably accurate readings for basic applications. For measuring atmospheric pressure, the BMP180 or BMP280 is widely used. These sensors not only measure pressure but also provide altitude data, which is useful in advanced weather predictions and geographic analysis. The BMP180 offers digital output via I2C communication, making it easy to integrate with microcontrollers. These sensors are interfaced with a microcontroller through digital or analog pins. The module collects data at regular intervals, which is then processed and transmitted for storage and analysis. All sensors used are energy-efficient, allowing the system to run on low power for long periods—ideal for remote or outdoor installations.

2. Microcontroller and Data Processing

The microcontroller serves as the central processing unit of the IoT-based weather reporting system. It connects to the environmental sensors and handles data collection, processing, and communication. Popular microcontrollers used in such systems include Arduino Uno, Arduino Nano, and Node MCU (ESP8266). These boards are chosen for their compact size, ease of programming, and compatibility with various sensors and communication modules. Once the sensors collect raw environmental data (such as temperature, humidity, and pressure), the microcontroller reads these inputs and converts the analog or digital signals into a usable format. In some cases, it performs basic filtering to remove noise or errors in the data, ensuring that only accurate information is transmitted. The microcontroller can also be programmed to operate at specific time intervals to read sensor data, process it, and send it to the cloud platform. This periodic data collection helps conserve energy, especially in battery-operated systems. In addition to data handling, the microcontroller ensures system stability by checking the status of sensors and network connectivity. If any issue arises, it can reset modules or generate error logs for troubleshooting. Overall, the microcontroller plays a crucial role in ensuring smooth operation, accurate data processing, and seamless communication in the weather reporting system.

3. System Customization and Scalability

System customization and scalability are vital aspects of an IoT-based weather reporting system, enabling it to adapt to varying user needs and environmental conditions. Customization allows the system to be tailored for specific applications such as agriculture, urban weather monitoring, school or campus-level use, or industrial environments. Depending on the use case, users can choose different sensors or modules. For example, in areas prone to flooding, rain sensors can be added. In colder regions, snow or frost detection modules can be integrated. On the software side, customization includes user-defined data logging intervals, alert settings, and display preferences. Dashboards can be configured to show only relevant parameters, and notifications can be sent via email or SMS based on specific thresholds. Scalability refers to the system's ability to expand its capacity and functionality without major changes to the core architecture. Additional sensor nodes can be easily added to monitor larger areas or more parameters. The cloud infrastructure, using platforms like Thing speak or Firebase, supports the handling of increased data volume with minimal effort. Moreover, scalable systems can integrate with other smart systems, such as irrigation controllers or emergency alert systems. This makes the weather reporting system more powerful and versatile .By incorporating customization and scalability, the system becomes future-ready, allowing users to start small and upgrade over time as needs grow. This flexibility makes the IoT-based weather system a long-term, cost-effective solution for accurate environmental monitoring.

CHAPTER – 5

BLOCK DIAGRAM

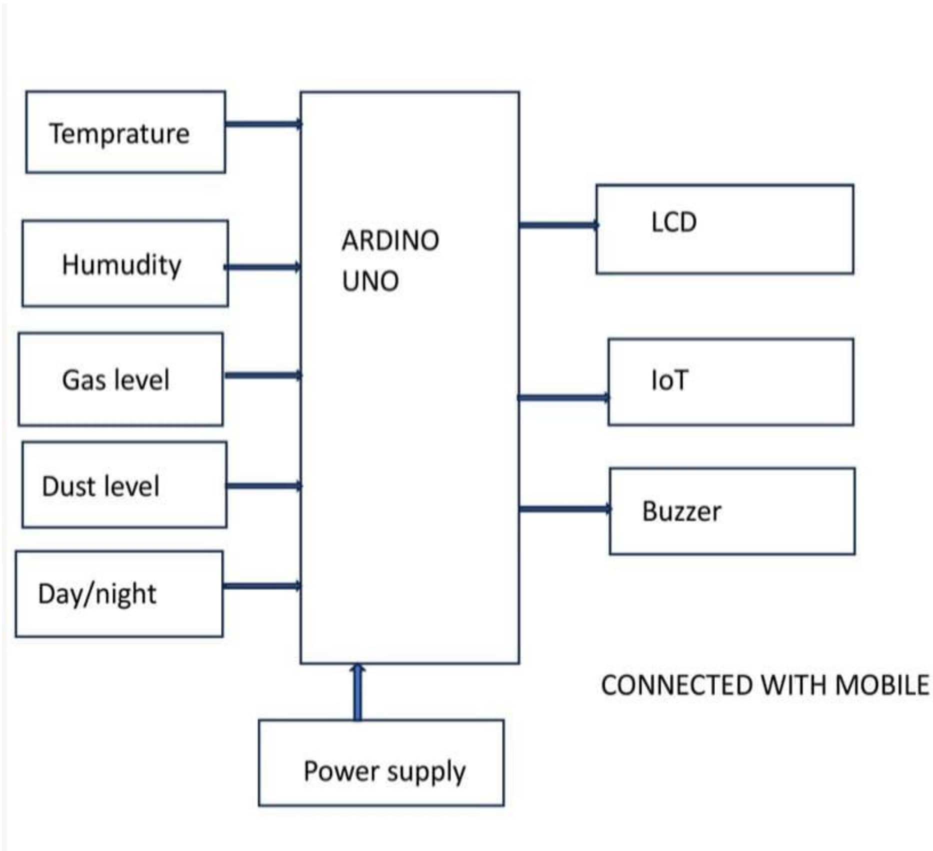

Fig:5.1(Block Diagram)

1. Temperature Sensor:

A temperature sensor, such as the DHT11 or DHT22, is essential for measuring ambient temperature in a variety of applications, including weather stations, home automation, and industrial monitoring. These sensors are connected to an Arduino UNO, providing real-time temperature data that can be processed and displayed on a screen or logged for further analysis. When interfacing with an Arduino UNO, these sensors communicate via a digital signal, where the Arduino reads the temperature data from the sensor and processes it accordingly. The temperature readings can be used in automated systems to control heating or cooling systems, trigger fans, or be displayed in real-time on a user interface, contributing to smart home or weather-based applications. The sensor's integration with Arduino opens up numerous possibilities for environmental monitoring and control.

2. Humidity Sensor:

The humidity sensor, often integrated into a single unit with temperature sensors like the DHT11 or DHT22, is used to measure the amount of moisture in the air. This sensor plays a vital role in environmental control systems, agricultural monitoring, and weather forecasting. Humidity is a key factor influencing comfort levels, air quality, and crop growth. These sensors typically output data in relative humidity percentages, where a value of 100% represents complete moisture saturation. Humidity levels above or below optimal ranges can cause discomfort, affect health, or damage equipment. In homes, for instance, a humidity sensor can trigger a dehumidifier or humidifier to maintain a comfortable living environment. In agricultural applications, maintaining proper humidity levels is crucial for the health of plants. A high level of humidity can lead to mold growth and plant diseases, while too little humidity can cause dehydration and hinder plant growth.

3. Gas Level Sensor:

Gas sensors, such as the MQ series (MQ-135, MQ-2, etc.), are designed to detect various harmful gases, including carbon monoxide (CO), carbon dioxide (CO_2), methane (CH_4), and liquefied petroleum gas (LPG). These sensors are commonly used in air quality monitoring, pollution detection, and safety systems, particularly in homes and industrial settings. The MQ series sensors use a semiconductor material that changes its resistance when exposed to different gases. This change in resistance is detected by the Arduino UNO and converted into a measurable signal. The sensor is particularly valuable in preventing accidents, such as gas leaks or carbon monoxide poisoning, which are often undetectable without specialized equipment. These sensors are often used in combination with alarms or automated systems to provide real-time alerts

4. Dust Level Sensor:

Dust level sensors, which detect particulate matter (PM) like PM2.5 and PM10, are vital tools for monitoring air quality. These sensors are used to assess pollution levels, which have direct implications for environmental health and public safety. In the context of an Arduino-based system, a dust sensor, such as the GP2Y1010AU0F or the PMS5003, can detect the concentration of these fine particles in the air. The sensor typically works by drawing air through a sensing chamber where a light is scattered by particles, and a photodiode detects the scattered light. Dust level sensors are important in urban air quality monitoring and can help trigger environmental control systems. For instance, when elevated dust levels are detected, a fan or air purifier can be activated. In industrial applications, dust sensors can ensure compliance with safety standards, alerting workers to harmful air conditions.

5. Day/Night Sensor:

The day/night sensor, often implemented using a Light Dependent Resistor (LDR), is used to detect ambient light levels and determine whether it’s day or night. This sensor is valuable in applications where lighting conditions influence system behaviour, such as outdoor lighting systems, solar power systems, and smart home automation. An LDR changes its resistance based on the amount of light it receives. In bright light conditions (daytime), the resistance is low, allowing current to pass through easily. At night, when light levels drop, the resistance increases, causing the current to decrease. In automated lighting systems, this sensor can control outdoor lights, turning them on when it gets dark and off when it becomes light. Similarly, in solar-powered systems, day/night sensors help switch between charging mode during the day and power-saving mode at night.

LCD Display:

The LCD Display is connected to the Arduino through the I2C or parallel communication pins (typically SDA, SCL for I2C or RS, RW, E for parallel). It serves to visually communicate the system's status to the user, showing whether a fall has been detected and if an alert is in progress. The LCD is powered by the 5V supply, and the data is transmitted from the Arduino to the LCD display to update the status messages.

Buzzer (Alert Mechanism):

The Buzzer is connected to one of the digital output pins of the Arduino. The Arduino controls the buzzer by sending a signal that causes the buzzer to sound when a fall is detected. The buzzer typically works on a 5V supply, which is drawn from the Arduino’s power output. A transistor may be used to ensure the proper current handling for the buzzer, ensuring it operates without overloading the microcontroller.

Arduino Uno Microcontroller:

The Arduino Uno microcontroller forms the heart of the circuit. It is powered by a 5V supply from either a DC power supply or a rechargeable battery. The Arduino is connected to the sensors and output devices. The analog or digital input pins of the Arduino are used to receive data from the Accelerometer and Gyroscope. The Arduino uses this data to calculate movement patterns and determine if a fall has occurred.

CHAPTER – 6

HARDWARE REQUIREMENT

ARDUINO UNO

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter

Fig : 6.1(Arduino UNO)

The Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $100, a considerable expense for many students. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino. Early Arduino boards used the FTDI USB-to-serial driver chip and an ATmega168. The Uno differed from all preceding boards by featuring the ATmega328P microcontroller and an ATmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

SPECIFICATION

1. Microcontroller: Microchip ATmega328P
2. Operating Voltage: 5 Volt
3. Input Voltage: 7 to 20 Volts
4. Digital I/O Pins: 14 (of which 6 provide PWM output)
5. Analog Input Pins: 6

6. DC Current per I/O Pin: 20 mA
7. DC Current for 3.3V Pin: 50 mA
8. Flash Memory: 32 KB of which 0.5 KB used by boot loader
9. SRAM: 2 KB
10. EEPROM: 1 KB
11. Clock Speed: 16 MHz
12. Length: 68.6 mm
13. Width: 53.4 mm
14. Weight: 25 g

COMMUNICATION

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins

PINS General Pin functions

- LED: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- VIN: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or

other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- 3V3: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND: Ground pins.
- IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- Reset: Typically used to add a reset button to shields which block the one on the board.

Special Pin Functions

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function.

In addition, some pins have specialized functions:

- Serial / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM (Pulse Width Modulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analog Write () function.
- SPI (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- TWI (Two Wire Interface) / I²C: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- AREF (Analog Reference): Reference voltage for the analog inputs

LIQUID CRYSTAL DISPLAY

A liquid crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. An LCD is a small low cost display. It is easy to interface with a micro-controller because of an embedded controller (the black blob on the back of the board). This controller is standard across many displays (HD 44780) which means many micro-controllers (including the Arduino) have libraries that make displaying messages as easy as a single line of code.

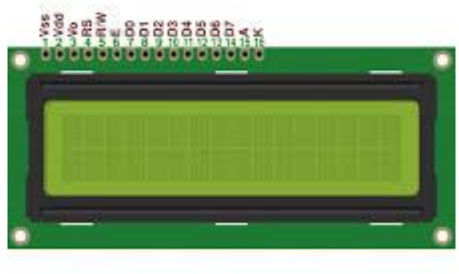

Fig : 6.2(LCD display unit)

LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and signage. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones, and have replaced cathode ray tube (CRT) displays in most applications. They are available in a wider range of screen sizes than CRT and plasma displays, and since they do not use phosphors, they do not suffer image burn-in. LCDs are, however, susceptible to image persistence.

16X2 LCD SPECIFICATIONS

- Display Format: 16 characters per line, 2 lines total.
- Character Size: 5x8 pixels for standard characters.
- Dimensions: Approximately 80mm x 36mm x 13mm.
- Interface: Parallel (4-bit or 8-bit mode).
- Supply Voltage: Typically, 5V DC.
- Current Consumption: Around 1.5 mA at 5V.
- Backlight: LED backlight (3.3V to 5V).
- Temperature Range: 0°C to 70°C operating, -20°C to 80°C storage.
- Response Time: Under 10 ms.
- Mounting: PCB or breadboard compatible.
- Character Set: Standard ASCII with custom character support.

BUZZZER:

A "buzzer sensor" typically refers to an electronic component that includes a buzzer integrated with additional circuitry to enable it to be controlled or triggered by an external signal.

Fig : 6.3(Buzzer)

Pin Configuration:

Magnetic Buzzer (Active Buzzer):

- Connect the positive leg of the buzzer to a digital output pin (e.g., pin 9) on the Arduino.
- Connect the negative leg of the buzzer to the GND pin on the Arduino.

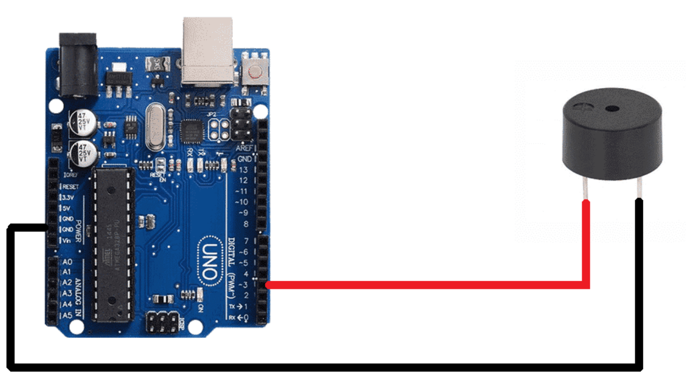

Fig : 6.4(Buzzer pin configuration)

Applications

- **Indication and Alerts:** Used in electronic devices and systems to provide audible feedback, alerts, or warnings. For example, in alarm systems, timers, and notification devices.
- **User Interaction:** Incorporated into user interfaces to provide feedback for button presses, actions, or system status.
- **Testing and Troubleshooting:** Used in testing and troubleshooting scenarios to provide audible feedback for diagnostics and operational status.
- **Games and Entertainment:** Integrated into games, toys, and entertainment devices to provide sound effects and enhance user experience.

DHT11(Temperature & Humidity sensor)

The DHT-11 Digital Temperature and Humidity Sensor is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

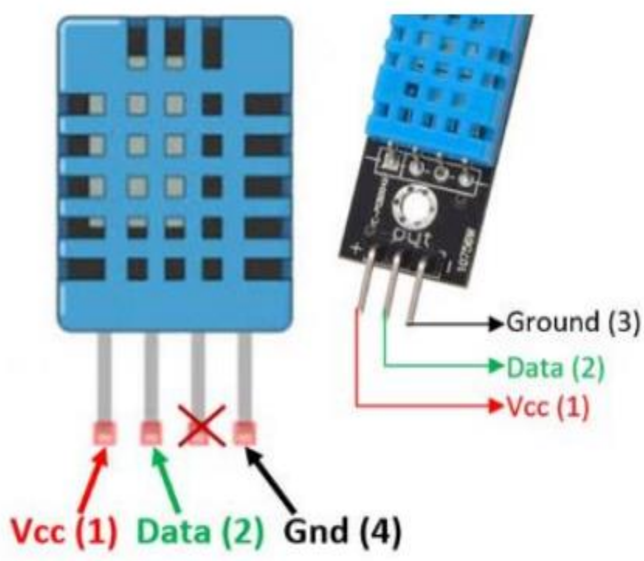

Fig: 6.5(DHT11)

1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both temperature and humidity through serial data
3	NC	No Connection and hence not used
4	ground	Connected to the ground of the Circuit

Table no: 6.1(Pin Configuration of DHT11)

DHT11 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

CHAPTER – 7

RESULT AND DESCRIPTION

An IoT-Based Weather Reporting System is a modern technological solution that leverages the Internet of Things (IoT) to monitor, analyze, and report real-time weather conditions efficiently. These systems utilize interconnected sensors, cloud computing, and artificial intelligence (AI) to deliver highly accurate and timely weather information for various applications, including agriculture, disaster management, transportation, and urban planning. The results of an IoT-based weather reporting system are impressive in terms of precision, automation, and accessibility. With real-time data collection, these systems can instantly monitor temperature, humidity, air pressure, wind speed, and precipitation levels. IoT-enabled weather stations ensure uninterrupted and automated data transmission to a centralized cloud platform, where AI algorithms process and generate forecasts. This allows for more informed decision-making across industries that rely on weather-sensitive operations. For example, farmers can use IoT-driven weather insights to optimize irrigation schedules, reducing water wastage and improving crop yields. Similarly, disaster management authorities can receive early warnings of cyclones, floods, or extreme weather events, enabling better preparedness and response. The integration of IoT technology with weather monitoring ensures not only accuracy but also efficiency in predicting and responding to environmental changes.

The description of an IoT-Based Weather Reporting System revolves around its key components, including smart sensors, communication networks, data processing units, and user-friendly interfaces. These sensors continuously measure atmospheric variables such as temperature, humidity, wind speed, and pressure, transmitting this data to an edge device or directly to cloud-based platforms. IoT connectivity enables seamless communication between various devices, ensuring that weather data is updated in real time without human intervention. The data is processed using AI-driven algorithms that analyze trends and patterns, leading to more refined and predictive forecasting models. In transportation and aviation, these systems help plan safer travel routes by assessing wind conditions and visibility. The system also enables personalized weather alerts, where individuals or organizations can set preferences for receiving weather warnings tailored to their specific needs. Unlike traditional weather reporting systems, IoT-based models minimize manual intervention and enhance precision, providing round-the-clock monitoring and remote accessibility. Furthermore, because IoT technology supports wireless communication, weather stations can be deployed in remote and hard-to-reach locations, improving data collection in underserved regions. By automating weather monitoring processes, IoT-driven solutions contribute to sustainable and smart city development while enabling better adaptability to climate variations. With the continuous expansion of IoT applications, these weather systems will become more sophisticated, precise, and accessible, shaping the future of meteorology and climate resilience.

CHAPTER – 8

CONCLUSION

The IoT-Based Weather Reporting System presented in this project highlights the transformative potential of the Internet of Things in environmental monitoring and smart data collection. Traditional weather monitoring methods, although accurate and well-established, often involve large infrastructure, high costs, and significant manpower. By contrast, IoT-based systems provide a compact, cost-effective, and real-time solution that can be implemented in diverse environments, from urban centers to remote rural locations. This project successfully designed and demonstrated a prototype that integrates various sensors—such as temperature, humidity, rain, and pressure sensors—along with a microcontroller and a wireless module to gather and transmit real-time weather data. The collected information is then uploaded to the cloud, where it can be accessed through a user-friendly web or mobile interface. This continuous flow of data ensures that users have up-to-date weather information at their fingertips, enabling better decision-making in fields like agriculture, transportation, disaster management, and everyday planning. One of the key achievements of the system is its ability to function autonomously, requiring minimal human intervention. The system not only collects data but also analyses it to detect patterns and abnormalities, which can be further expanded into predictive analysis in future enhancements. Moreover, the system is scalable and modular, allowing for the easy addition of new sensors or functionalities, such as wind speed monitoring, UV index tracking, or air quality analysis. In the broader context, this IoT-based weather system contributes to the growing trend of smart cities and connected environments. With climate change becoming an increasing concern, real-time weather data plays a crucial role in alerting citizens and authorities about extreme weather conditions. Integrating such systems into public infrastructure can significantly enhance emergency preparedness and response times. In conclusion, the IoT-Based Weather Reporting System developed in this project is a significant step towards creating smarter, more responsive weather monitoring solutions. It proves that even small-scale systems can make a meaningful impact when designed with efficiency and scalability in mind. With further development, this project could be transformed into a commercial or community-based application, especially beneficial for regions with limited access to professional meteorological data. Future improvements may include enhanced sensor accuracy, solar-powered modules for energy efficiency, machine learning integration for predictive weather modelling, and expanded connectivity using LoRa or 5G technologies. As IoT continues to evolve, its application in weather reporting will undoubtedly become more sophisticated, reliable, and vital in our day-to-day lives.

REFERENCE

[1] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer et al., “Building the internet of things using RFID: The RFID experience,” IEEE internet compute., vol. 13, no. 3, pp.48- 55, May Jun. 2009.

[2] Shifeng Fang; Li Da Xu; Yunqiang Zhu; Jiaerheng Ahati; Huan Pei; Jianwu Yan; Zhihui Liu., “An integrated system for regional environmental monitoring and management based on internet of things”, IEEE Transactions on Industrial Informatics, vol.10, no. 2, pp.1596-1605, May-Jun. 2014.

[3] J. A. Stankovic, “Research directions for the Internet of Things,” IEEE Internet Things., vol. 1, no. 1, pp. 3–9, Feb. 2014 [4] Shanzhi Chen; HuiXu; Dake Liu; Bo Hu; Hucheng Wang.

[5] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Compute. New., vol. 54, no. 15, pp. 2787–2805, 2010

[6] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini,“Convergence of MANET and WSN in IoT urban scenarios,”IEEE Sens. J., vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[7] BulipeSrinivasRao , Prof. Dr. K. SrinivasaRao , Mr. N. Ome, “Internet of Things (IOT) Based Weather Monitoring system”, IJARCCE Journal, Vol. 5, no. 9, sept. 2016.

[8] B. Vongsagon, J. Ekchamanonta, K.Bumrungkhet, and S.Kittipiyakul, “XBee wireless sensor networks for temperature monitoring”, Retrieved 7/11/15 World Wide Web http://citeseerx.i st.psu.edu/viewdoc/download?doi=10.1.1.4

[9] Nashwa El-Bendary, Mohamed Mostafa M. Fouad, Rabie A. Ramadan, Soumya Banerjee and Aboul Ella Hassanien, “Smart Environmental Monitoring Using Networks”,K15146_C025.indd, 2013 Wireless Sensor

[10] Grzegorz Lehmann, Andreas Rieger, Marco Blumendorf, SahinAlbayrakDAI, “A 3-Layer Architecture for Smart Environment Models”/A model-based approach/LaborTechnische University Berlin, Germany 978-1-4244-5328-3/10 © IEEE,2010.

[11] CharalamposDoukas, “Building Internet of Things with the ESP8266”, CreateSpace Publications, 2012

[12] Shifeng Fang et al., "An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things," , IEEE Transactions on Industrial Informatics , vol.10, no.2, pp.1596-1605, May 2014.