MINISTRY OF INDUSTRY AND TRADE

HANOI UNIVERSITY OF INDUSTRY

ALBERTO ERNESTO COBOI

DESIGN OF AN INDUSTRIAL DATA ACQUISITION SYSTEM USING ARM MICROCONTROLLER AND WIRELESS SENSORS

MASTER'S THESIS THESIS IN ELECTRICAL ENGINEERING

Hà Nội – 2025

MINISTRY OF INDUSTRY AND TRADE
HANOI UNIVERSITY OF INDUSTRY

ALBERTO ERNESTO COBOI

DESIGN OF AN INDUSTRIAL DATA ACQUISITION SYSTEM USING ARM MICROCONTROLLER AND WIRELESS SENSORS

Program: Master in Electrical Engineering
Course code: 8520201

MASTER'S THESIS THESIS IN ELECTRICAL ENGINEERING

SUPERVISORS:
1. DR. PHẠM VĂN NAM
2. DR. HÀ VĂN PHƯƠNG

Hà Nội – 2025

DECLARATION OF AUTHORSHIP

I am Alberto Ernesto Coboi, a graduate student of the Master's program in Electrical Engineering, cohort CH KTD K13.2. I hereby declare that the Master's thesis entitled:

"Design of an Industrial Data Acquisition System Using ARM Microcontroller and Wireless Sensors" presented in this document is the result of my independent research conducted under the academic supervision of Dr. Hà Văn Phương and Dr. Phạm Văn Nam.

All the research content, including analyses, experimental results, and evaluations presented in this thesis, is truthful and accurately reflects the process of work and effort I have carried out. To the best of my knowledge, the findings and results presented here have not been previously published in any other scientific work or used for any academic qualification elsewhere.

All figures and data used in the thesis were collected from reliable sources and are fully cited in the references section of the thesis. I affirm that I have strictly followed ethical research principles throughout the research and writing process. The referenced materials have been appropriately used and cited in accordance with academic standards, showing full respect for prior authors and ensuring transparency in my research.

I understand and accept that if any misconduct, plagiarism, or misrepresentation is found in this thesis, I will take full responsibility in accordance with the laws and the regulations of the University.

I hereby certify the accuracy and integrity of the entire content presented in this thesis.

Hanoi, Vietnam – 2025

_____

Alberto Ernesto Coboi

Contents

Table of Figures

List of Tables

ACKNOWLEDGEMENTS

First and foremost, I would like to express my heartfelt gratitude to God for the precious gift of life and the continuous protection bestowed upon me. Additionally, I extend my deepest appreciation to my friends and the entire university community at Hanoi University of Industry for their untiring support throughout my studies. Their unconditional assistance and acts of kindness are rare treasures that have greatly enriched my journey.

I would also like to thank my advisors, Dr. Hà Văn Phương and Dr. Phạm Văn Nam, for their invaluable guidance, encouragement, and expert feedback throughout this research. Your mentorship has been instrumental in shaping both the quality of this work and my growth as a researcher.

To my parents, thank you for your endless love, sacrifices, and confidence in my abilities. Your faith in me has been my strongest foundation.

A special thank you goes to my girlfriend, Juliet, for her unwavering support, patience, and love during this challenging process. Your constant belief in me kept me motivated every step of the way.

Finally, I am deeply indebted to Ms. Dung for her unconditional care and support during my time in Vietnam. You have been like a second mother to me, and your guidance during both challenges and triumphs has meant everything.

Thank you all for being part of this journey.

Abstraction

One of the major industries in Vietnam that makes a substantial contribution to the GDP is the coal mining sector. However, this industry faces obstacles that compromise operational safety and efficiency, such as hazardous subterranean conditions and the inertia of current monitoring systems. We present in this paper, a case study on creating and implementing a reliable, low-power, scalable environmental monitoring system for industrial environments. The suggested system collects data on temperature, humidity, gas concentration, and light intensity at two nodes (slave and master) across the monitored area using LoRa-based wireless sensor networks. To satisfy dependability requirements in areas with connectivity problems, a hybrid architecture has been created that allows data to be sent to a cloud platform via an ESP8266 Wi-Fi module while simultaneously displaying locally on-site and offline storage of information. With broad applicability potential across comparable industrial environments, the suggested solution offers a versatile, energy-efficient model for improved safety and operational oversight within coal and mineral plants in Vietnam.

Key words: Internet of Things, coal mining, environmental monitoring, Vietnam, real-time data monitoring, low power systems, ARM-based microcontrollers, STM32, LoRa Communications, Wireless Sensor Networks.

List of Abreviations

Abbreviation	Full Form	Context
ADC	Analog-to-Digital Converter	Converts analog sensor signals to digital for STM32
AM/PM	Ante Meridiem / Post Meridiem	Time format for data logging intervals
AP	Access Point	Wi-Fi mode of ESP8266 module
ARM	Advanced RISC Machine	Microcontroller architecture (e.g., ARM Cortex-M3)
BLE	Bluetooth Low Energy	Mentioned as an alternative IoT wireless technology
CO	Carbon Monoxide	One of the environmental gases measured
CTR	Counter Mode	Encryption mode mentioned in network security context
DHT11	Digital Humidity and Temperature Sensor	Sensor for temperature and humidity measurement
ESP8266	Espressif Systems Wi-Fi Module	Module for wireless communication
GPIO	General-Purpose Input/Output	Pins on STM32 used for interfacing sensors
I2C	Inter-Integrated Circuit	Communication protocol used in sensor connections
IDE	Integrated Development Environment	Used for programming Arduino and MIT App Inventor
IoT	Internet of Things	Core system concept throughout the thesis
JSON	JavaScript Object Notation	Data format used by Firebase

LCD	Liquid Crystal Display	Used to show sensor values locally
LoRa	Long Range	Wireless communication method used in the system
LoRaWAN	Long Range Wide Area Network	Protocol stack for LoRa communication
LPWAN	Low Power Wide Area Network	Communication category including LoRa
MAC	Media Access Control	Layer in LoRaWAN network protocol
MIT	Massachusetts Institute of Technology	Institution behind MIT App Inventor
MQ2	Metal Oxide Gas Sensor	Sensor for detecting CO and other gases
NB-IoT	Narrowband Internet of Things	Compared in LPWAN alternatives
NTC	Negative Temperature Coefficient	Thermistor type in DHT11 sensor
RH	Relative Humidity	Measured by DHT11 sensor
SF	Spreading Factor	LoRa parameter controlling data rate vs range
SPI	Serial Peripheral Interface	Communication between STM32 and sensors
STA	Station Mode	Wi-Fi mode for connecting ESP8266 to access points
STM32	STMicroelectronics 32-bit Microcontroller	ARM-based microcontroller used as core of the system
UART	Universal Asynchronous Receiver-Transmitter	Serial communication interface used between devices

Wi-Fi	Wireless Fidelity	Used for cloud data transfer (via ESP8266)
Zigbee	Zigbee Protocol	Mentioned in comparison with LoRa and BLE
°C	Degrees Celsius	Temperature unit used in experimental results
ppm	Parts Per Million	Unit for gas concentration
lux	Lux	Unit for light intensity measurement

CHAPTER 1. INTRODUCTION

1.1. RESEARCH RATIONALE AND URGENCY

In mineral coal plants, data collection and management play a vital role in ensuring production efficiency and labor safety. However, traditional data logging systems often face significant challenges in harsh operating environments characterized by dust, high humidity, and fluctuating temperatures, which compromise the performance and reliability of conventional equipment.

To address these issues, the application of ARM IC technology and wireless sensor networks has emerged as an advanced and effective solution. ARM microcontrollers offer high performance, low power consumption, and superior integrated features. When combined with wireless sensor networks, this technology enables the development of modern datalogger systems that automate data collection from multiple monitoring points throughout the plant.

An ARM-based wireless datalogger system not only reduces human error but also provides real-time monitoring and data analysis capabilities. This is particularly crucial for early detection of potential issues, accident prevention, and supporting timely managerial decisions. Furthermore, remote wireless data access optimizes production processes and enhances labor productivity.

Implementing this system in Vietnam's mineral coal plants will improve operational efficiency, ensure worker health and safety, reduce operating costs, and enhance corporate competitiveness. Given these practical benefits, developing such a datalogger system represents a significant step in modernizing Vietnam's coal mining and processing industry.

1.1.1. Research Background

Globally, advancements in datalogger systems and wireless sensor networks have been driven by the need for efficient monitoring and automation. Applications span industrial production, environmental monitoring, healthcare, and smart city development, with research primarily addressing:

- **Connectivity and Communication Protocols**: Challenges like signal interference, data loss, and network reliability are commonly addressed through advanced encryption and protocols (Zigbee, LoRa, BLE) to enhance stability and energy efficiency.
- **Sensor Durability**: Harsh conditions (high temperatures, dust, humidity) affect sensor reliability. Improvements in sensor materials and protective casings have extended lifespan and ensured data accuracy.
- **IoT and AI Integration**: Combining IoT with AI enables predictive analytics, real-time monitoring, and automated decision-making to enhance system effectiveness.

In Vietnam, these technologies are increasingly applied in environmental monitoring and industrial automation. Domestic efforts focus on adapting global innovations to local contexts, addressing cost constraints, and optimizing systems for industrial requirements.

This research bridges existing gaps by developing a cost-effective, durable, and high-precision datalogger system tailored for Vietnam's mineral coal plants. The system will incorporate AI for predictive maintenance, real-time decision support, and operational optimization, providing practical solutions to industry-specific challenges.

1.1.2. Research Methodology

To achieve the research objectives, the following methods will be employed:

1. **Theoretical Research**: Review and analyze literature on datalogger systems, ARM ICs, wireless sensor networks, and data analysis algorithms to establish a theoretical foundation and technical criteria.
2. **Modeling**: Develop system models to simulate sensor components, communication protocols, and data processing units, evaluating factors affecting performance and reliability in real-world conditions.
3. **Simulation**: Use specialized tools (MATLAB, Proteus, wireless network simulators) to test system feasibility and optimize design parameters (energy efficiency, data transmission speed, network connectivity).
4. **Experimental Deployment**: Implement a pilot system in a mineral coal plant to validate simulation results and assess performance in actual environmental conditions.
5. **Analysis and Evaluation**: Analyze collected data to evaluate accuracy, reliability, and operational efficiency, comparing results with traditional systems to identify strengths and limitations.

1.1.3. Research Objectives

- Design and develop an ARM microcontroller-based datalogger system for data collection and processing from plant-wide sensors.
- Propose a wireless sensor network deployment method to monitor critical parameters (temperature, humidity, pressure, emissions).
- Apply AI to analyze sensor data for decision support and production optimization.
- Develop algorithms for early fault detection and maintenance planning to minimize safety risks and enhance operational efficiency.

- Build and validate an experimental model to verify system accuracy and effectiveness in real-world conditions.

1.1.4. Research Tasks

- Analyze technical requirements for a mineral coal plant datalogger system.
- Develop and test an experimental model in an operational plant to verify research outcomes.
- Assess the system's economic feasibility and effectiveness in plant monitoring and optimization.
- Propose improvements based on experimental results and practical feedback.

1.1.5. Scientific Significance

- Identify key production parameters in mineral coal plants.
- Demonstrate IoT applications for wireless sensor data analysis.
- Establish an automated monitoring system to improve production management.
- Validate system performance through real-world experimentation.

1.1.6. Research Scope and Institutional Context

Thesis Title: "Application of ARM IC and Wireless Sensor Networks to Build a Datalogger System for Vietnam's Mineral Coal Plants".

1.1.7. Focus:

- Monitor and collect sensor data from production processes.
- Utilize IoT for data analysis, process optimization, and early fault detection.
- Address data collection, processing, and system deployment challenges.

Host Institution: Hanoi University of Industry (HaUI)

- Recognized as a leading technical training institution in Vietnam with modern facilities and outstanding faculty.
- Strong research capabilities in smart university models and internationally benchmarked fields.
- Extensive collaborations with educational, research, and industrial partners, providing access to data and resources for thesis completion and scientific publication.

1.2. STRUCTURE OF THE THESIS

This thesis is structured into five main chapters, each serving a specific purpose in addressing the research problem:

- Chapter 1: Introduction;
- Chapter 2: Vietnam's Coal Industry: Operational Challenges and IoT-Driven Safety Solutions;
- Chapter 3: Sensor Calibration Methodologies for Industrial IoT Systems;
- Chapter 4: Results and Discussion;
- Chapter 5: Conclusions and Future Directions.

1.3. CONCLUSION OF CHAPTER 1

Chapter 1 provided a comprehensive foundation for the research by highlighting the urgency and relevance of deploying intelligent monitoring solutions in Vietnam's mineral coal industry. It established the motivation for developing a low-power, scalable, and robust datalogger system using ARM microcontrollers and wireless sensor networks. The chapter outlined the current challenges in environmental monitoring within coal plants and proposed a research framework that integrates theoretical studies, simulation, system modeling, and real-world implementation. By clearly defining the research objectives, tasks, and significance, this chapter sets the stage for developing a

cost-effective and reliable solution tailored to the specific operational demands of Vietnam’s industrial environment.

CHAPTER 2. VIETNAM'S COAL INDUSTRY: OPERATIONAL CHALLENGES AND IOT-DRIVEN SAFETY SOLUTIONS

2.1. INTRODUCTION

The amount of coal produced by Vietnam's underground coal mines has grown significantly over the past 20 years, from 4.3 million tonnes in 2000 to 22.1 million tonnes in 2018 (a five-fold increase). Total coal production is expected to reach 41 million tons in 2019, 47.8 million tons in 2020, and 49.3 million tons in 2025, according to the Master Plan of Coal Industry Development in Vietnam by 2020 with perspective to 2030, which was modified in 2016 and the VINACOMIN mining plan [1]. Given the rising demand for coal for both domestic and export markets, the coal mining sector currently contributes significantly to Vietnam's economic growth. Nonetheless, a number of obstacles confront Vietnam's coal mining sector, such as growing coal output, prospecting and exploration, mining technologies, environmental issues, and sustainable development [2].

A lot of coal miners worry about their safety at work. Poor ventilation in subsurface mines exposes workers to heat, dust, and toxic gases, which can cause illness, injury, and even death [3]. The Industrial Internet of Things (IIoT) was created as a result of the Internet of Things' (IoT) use in industrial systems as well as other recent technological developments. IIoT can assist in overcoming the shortcomings of traditional monitoring and control systems, allowing businesses to establish a single monitoring system to automate processes, offer a secure workplace, efficiently enforce compliance, and manage environmental concerns. It makes sense that so many mining companies have increased their investment after implementing IoT-enabled

solutions in their organizations, given the benefits that IoT brings to the forefront. to improve mining operations' environmental sustainability, productivity, and safety [3,4].

This paper suggests an IoT-based datalogger system that makes use of the ARM Cortex family, which exhibits notable characteristics and is appropriate for embedded development, in order to overcome these constraints. Additionally, the wireless sensor node makes the embedded system design a reality. The nodes, which use an ARM Cortex microcontroller, are made to keep an eye on the coal mining sector. Consequently, the development of wireless sensor networks requires precise and low-power sensor nodes. In actuality, microcontrollers with promising features can be used to design the nodes. The hardware and software design issues affect the sensor nodes' features [5].

The physical layer, LoRa, allows the long-distance link, while the communication protocol and network system architecture are defined by the LoRaWAN architecture, which is referred to as a "star of stars." A node's battery life, network capacity, QoS, security, and the number of applications the network serves are all influenced by the protocol. The suggested system seeks to offer a dependable, scalable, and reasonably priced data logging solution that is suited to Vietnam's coal and mineral plants by combining these technologies [6].

By taking advantage of the increasing prevalence of wireless, mobile, and sensor devices as well as radio-frequency identification (RFID), the Internet of Things (IoT) has offered a promising chance to develop robust industrial systems and applications. In recent years, a vast array of industrial IoT applications have been created and implemented. By taking advantage of the increasing prevalence of wireless, mobile, and sensor devices as well as

radio-frequency identification (RFID), the Internet of Things (IoT) has offered a promising chance to develop robust industrial systems and applications. In recent years, numerous industrial IoT applications have been created and implemented [7].

LoRa, Sigfox, and NB-IoT are examples of LPWAN technologies that are crucial to this advancement. Long-distance communication in low-power devices with low operating costs is made possible by these technologies, which are appropriate wireless communication protocols for battery-powered Internet of Things objects. A number of review papers have examined LPWAN technology in the Internet of Things from various angles. Nonetheless, it is evident that there is a dearth of thorough research describing the function of these technologies in various IoT applications [8].

Data transfer relies on both wired and wireless communication technologies, each of which has advantages and disadvantages of its own. Because of their high speed, dependability, and security, wired systems such as Ethernet and fiber optics are perfect for stable settings like data centers and businesses. They do, however, have mobility and scalability issues. Although wireless technologies (like Wi-Fi and 5G) support mobile devices and the Internet of Things and offer flexibility and ease of deployment, they suffer from interference, security threats, and bandwidth limitations. In today's digital world, the decision between them is influenced by elements such as cost, mobility, and performance requirements, which shape user experience and efficiency [9].

These drawbacks are particularly noticeable when considering Vietnam's coal and mineral plants because of the country's rough terrain, deep mining activities, and possible exposure to explosive gases (Vinacomin, 2019).

Because of this, operators need a reliable and adaptable solution that can change with the site without becoming unaffordable.

Vietnam's coal industry has major gaps in data recording systems and energy efficiency, despite worldwide advancements in wireless technologies. Existing solutions like Wi-Fi, LoRa, and Zigbee provide low-energy, limited-range options, but they might not fully meet the needs of the industry. Vietnam should give thorough research into cellular networks top priority in order to close this gap. These networks could offer scalable, long-range, and energy-efficient alternatives that are suited to industrial needs.

Vinacomin is aware of the importance of occupational safety and health and risk management in addressing unknown risks and hazards [10]. Using artisanal labor is inevitable in the mining industry of Vietnam because there is less mechanized equipment than in other nations. Because of this, risk management in Vietnam ought to be given careful consideration through a variety of educational and training programs, financial incentives, and knowledge and technological transference [10].

The need for an effective and user-friendly system that incorporates cutting-edge technologies like ARM and LoRaWAM and combines low energy consumption with high performance is made clear by the difficulties this scientific study has revealed. In order to achieve this, we suggest a microcontroller-based long-range communication system that makes use of LoRa (Long Range) technology, which is well-known for its effectiveness in low-power transmissions and broad coverage.

This system will be a scalable and economical solution for real-time monitoring of various industrial environments in Vietnam, in addition to offering dependable communication. In addition to thoroughly examining these

issues, our research will create a better system design that takes into account the unique challenges encountered in the Vietnamese industrial setting.

This research is compartmentalized in sections to give an elaborate explanation of the presented system. The Introduction provides a background and overview of Vietnam's coal industry and the necessity of developing a monitoring protocol. In the next section, the problem statement is defined and the key limitations of current monitoring technologies in industrial operations in harsh environments are identified. Next, the solution is presented, and the architecture is explained, covering the adoption of ARM-based STM32 microcontrollers and LoRa wireless communication. The next section discusses both hardware and software architectural design choices made in terms of sensor integration, data flow, and cloud connection through the ESP8266 Wi-Fi module. The study then describes the testing methodology and presents experimental results using real-time data collected over several days. Finally, the paper bunks up with summary of findings, effects on industries, and suggestions for future improvisations.

2.2. SYSTEM DESIGN

The two node parts of the ARM-Based architecture of the system suggested in this paper, the main gateway and a slave node are intended for real-time monitoring and control in challenging industrial settings, such as Vietnam's coal mines. The STM32F4 microcontroller, which powers this remote system, gathers all of the data from the sensors and I/O modules. On the other hand, the ESP8266 microcontroller improves data processing and connectivity. After that, all of the data is transmitted to a Bus Station server via Lora Technology, which is a low-power communication method. Finally, the entire system makes use of the Internet of Things to send data to cloud servers and read it in an app designed to store and visualize it, guaranteeing

dependability under extremely difficult circumstances such as underground mines, as shown in figure 2-1.

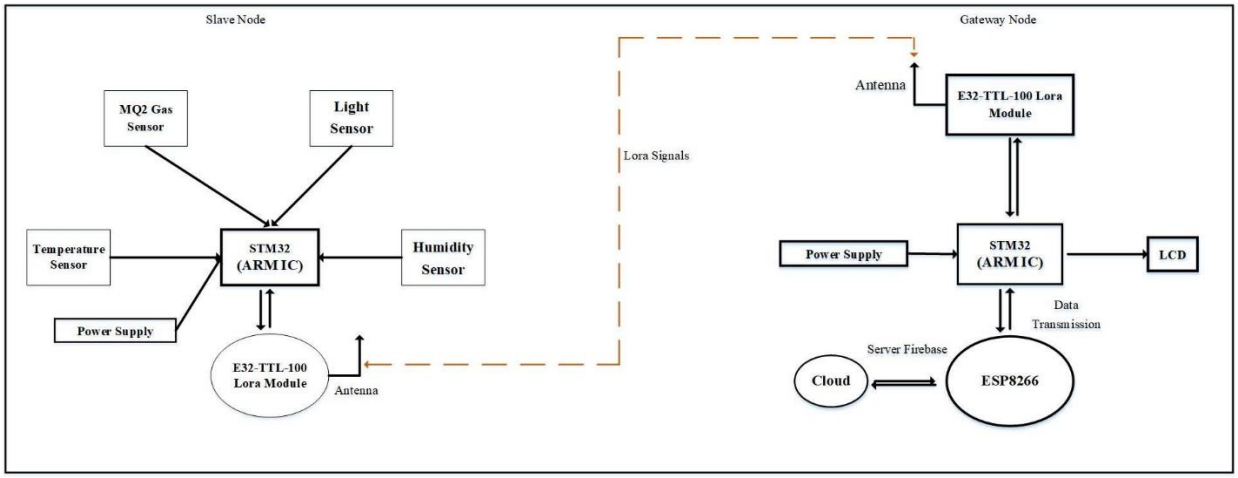


Figure 2-1: System Design For Our Model.

A Master Node (gateway) and Slave Nodes (sensor nodes) make up the hierarchical communication architecture used in the suggested system. Every Slave Node incorporates environmental sensors (temperature, humidity, MQ2 gas, and light) for real-time monitoring in addition to an STM32 microcontroller to control data collection and transmission. Low-power, long-range data broadcasting to the Master Node is made possible by a LoRa module. These nodes are made to work effectively in industrial environments, using the least amount of energy possible while still collecting data with accuracy.

The Master Node acts as a conduit between the cloud and Slave Nodes. It has an STM32 microcontroller that processes incoming data and uses an LCD screen to display it locally. The verified data is then sent to a cloud-based IoT platform by the ESP8266 Wi-Fi module for remote storage and real-time monitoring. This system provides a scalable, energy-efficient, and extremely dependable solution for industrial applications in Vietnam by utilizing LoRa

for long-range communication and Wi-Fi for cloud connectivity, meeting crucial demands in data management and remote monitoring.

2.2.1. Problem Formulation

An essential part of the Vietnamese economy, the coal and mineral sector employs thousands of people and makes a substantial GDP contribution. The industry does, however, face many operational difficulties, especially when it comes to maintaining equipment dependability and worker safety in hostile environments that are marked by dust, high humidity, mechanical vibrations, and explosive gases. Conventional monitoring systems, which depend on wired or short-range wireless technologies, are frequently expensive, rigid, and prone to malfunctioning under these circumstances. The inability of these systems to offer remote access and real-time data makes it challenging to react quickly to emergencies like gas leaks or equipment failures.

Current wireless solutions, like Wi-Fi and Zigbee, are not appropriate for large-scale industrial deployments because they either have a short range or use excessive amounts of power. Although cellular networks have a large coverage area, they are expensive to operate and might not offer dependable connectivity in isolated or subterranean areas. Additionally, a lot of the systems in use today are unable to store data locally and instead rely entirely on cloud-based solutions, which can malfunction in places with inadequate network coverage. In Vietnam's coal and mineral plants, this technological divide has resulted in ineffective operations, elevated safety hazards, and increased maintenance expenses.

2.2.2. Problem Statement

A vital component of the Vietnamese economy, the coal and mineral extraction sector generates thousands of jobs and contributes to GDP growth. But given the harsh and uncertain conditions of mining environments, the industry faces significant operational challenges that endanger worker safety, equipment dependability, and overall productivity.

2.2.3. Key Challenges

Given that it depends on antiquated monitoring equipment, Vietnam's coal mining sector faces serious operational and safety challenges. Employees work in dangerous environments where they are frequently exposed to potentially explosive gases, high humidity, dense dust clouds, and mechanical vibrations. These environmental risks are exacerbated by antiquated monitoring systems that are unable to identify crises instantly, resulting in hazardous delays in reacting to equipment malfunctions or gas leaks. In addition to putting workers in danger, delayed hazard detection causes expensive operational disruptions and equipment damage.

Another significant drawback of the sector is its reliance on wired monitoring systems. These systems offer little flexibility in dynamic mining environments and necessitate a large physical infrastructure, which raises installation costs. Supervisors are also unable to continuously monitor equipment status or environmental conditions across extensive mining operations due to the lack of remote monitoring capabilities. Wireless alternatives like Wi-Fi and Zigbee have been tried by many operations, but in large-scale industrial settings, these technologies have reliability problems and a limited range. Cellular networks offer greater coverage, but they are not practical due to their high operating costs and erratic performance in isolated

or subterranean locations. Furthermore, the majority of current systems do not have local data storage, which makes them ineffective in areas with inadequate internet connectivity, which is a common problem in Vietnam's mining regions.

2.2.4. Proposed Solution:

We suggest creating an intelligent industrial monitoring system that combines reliable sensor technology with an adaptive communication architecture in order to address these urgent issues. To provide thorough real-time monitoring of mining conditions, the system will incorporate four essential environmental sensors: those that measure temperature, humidity, flammable gases, and light intensity. A dispersed network of ARM microcontroller-based nodes, selected for their ideal ratio of processing power to energy efficiency, will link these sensors. The system will make use of LoRa technology, which provides remarkable penetration through obstacles while maintaining low power consumption, for dependable long-range communication in difficult environments.

The dual-layer architecture of the solution guarantees uninterrupted operation under all connectivity circumstances. Sensor data will be gathered and processed by a local gateway node, which will simultaneously send it to cloud storage via Wi-Fi when it is available and display important information locally via an LCD interface. This hybrid strategy stores data locally until connectivity is restored, preserving functionality during internet outages. Since the system was created with scalability in mind, it can be used profitably in operations of all sizes, from massive coal mines to tiny mineral extraction sites. Vietnamese mining operations can attain previously unheard-of levels of safety oversight, operational effectiveness, and cost-effectiveness by putting this cutting-edge monitoring network into place. This will turn one of the country's

most important industries into a prototype for contemporary industrial monitoring techniques.

2.3. TECHNOLOGICAL FOUNDATIONS FOR INDUSTRIAL IOT MONITORING SYSTEMS

2.3.1. ARM Microcontrollers

Because of their strong architecture, scalability, and extensive peripheral integration, STM32 microcontrollers are unique in industrial settings. They are perfect for a variety of industrial applications because they are designed to withstand harsh conditions and offer excellent performance, dependability, and efficiency. From entry-level to high-performance models, their scalable range enables developers to choose cost-effective solutions that are suited to particular requirements. STM32 microcontrollers also have a wide range of peripherals (GPIO, UART, SPI, ADC, etc.) that allow for easy integration with sensors, actuators, and communication modules, increasing their industry-wide adaptability [11, 12].

STM32 microcontrollers' real-time capabilities, energy efficiency, and extensive development ecosystem are among their main advantages. They guarantee accurate timing for crucial tasks like automation and motor control with ARM Cortex-M cores. They are appropriate for battery-operated systems because of their sophisticated power management, which lowers energy consumption. In addition, STMicroelectronics expedites deployment while protecting industrial systems from cyberattacks by providing developers with long-term availability, strong security features (secure boot, cryptographic acceleration), and an extensively documented IDE ecosystem. STM32 microcontrollers are a flexible, future-proof option for industrial innovation because of these qualities combined [11, 12].

2.3.2. Wireless Sensor Networks WSNs

A collection of nodes makes up a wireless sensor network (WSN). A microcontroller, data storage, sensor, analogue-to-digital (ADC) converters, a data transceiver, and an energy source are all common components of a node. Various architectures are used to connect the nodes to one another, depending on the environment and applications. The restricted physical size of the sensor nodes limits their ability to use certain resources. As a result, sensor nodes have limited computational power, limited memories, limited transmission range, and limited battery energy supply. Nonetheless, there are numerous benefits that allow WSNs to be widely used for civil and surveillance purposes. A few of these characteristics can be summed up as follows: densely dispersed nodes in the working field that are capable of self-organization and random location; locally processing data gathered from the natural environment; and adaptive cooperation between nodes for data transmission and routing [13].

2.3.3. LoRa Fundamentals

Recently, a number of long-range wireless communication technologies have been put forth. Since LoRa was among the first of these technologies to be made commercially available, a lot of scholarly research has been done to examine and evaluate its functionality. LoRaWAN is a networking design that allows telecom operators to offer subscription services based on LoRa PHY, which is a long-range wireless transmission technique. Utilizing the sub1 GHz wireless frequency band, LoRa PHY is a proprietary chirp spread spectrum scheme [14,15].

Since the frequency of the chirp signal changes linearly over time, LoRa technology is immune to interference thanks to CSS modulation, which is used by major LPWAN technologies in the sub-GHz band.

Compared to other LPWAN technologies, the chirp signals instantly use the available bandwidth while using less power, table 2:1 compares some frequencies in different regions. Compared to other technologies, a nominal coverage of 5–15 km is achieved with a higher payload (up to 250 bytes). Compared to Sigfox and Ingenu, LoRa networks have superior downlink capabilities. During activation, LoRa networking offers lightweight authentication and encryption methods that can be set up. Configuration and firmware updates can be transmitted over the air, which is another significant benefit of LoRa networks [14,15].

Table 2:1: Regional Frequency Allocations.

Region	**Frequency (MHz)**
Asia	433
Europe, Russia, India, Africa (parts)	863–870
US	902–928
Australia	915–928
Canada	779–787
China	779–787, 470–510

The interplay of spreading factors, bandwidth, and coding rate has a substantial impact on the maximum payload length and associated data rate in LoRaWAN. Generally speaking, selecting lower bandwidths and spreading factors results in longer transmission ranges but lower data rates; conversely, selecting lower bandwidths and spreading factors results in shorter transmission ranges but higher data rates [16].

Although there is no set range for LoRa data rates, they can be anywhere between 0.162 kb/s and 50 kb/s. LoRaWAN-based systems that use LoRa in the physical layer (PHY) can reach data speeds of 5.47 kb/s, 11 kb/s, or 21.9 kb/s, depending on the deployment location. The LoRaWAN can reach a maximum data rate of 50 kb/s thanks to the frequency shift keying (FSK) modulation technique [16].

In order to overcome the communication distance limitation of distributed energy harvesting systems and to reduce the cost of network construction, LoRa wireless sensing modules can be deployed farther, achieving the unification of low power consumption and long distance. This allows wireless radio frequency communication to be extended by three to five times the traditional one under the same power consumption [17].

Spreading factor, channel bandwidth, and coding rate are all modifiable system parameters that affect the bit rate of LoRa radios. A higher spreading factor improves transmission reliability at a lower bit rate by requiring more chirps to modulate a single bit. When two or more incoming packets overlap in frequency and time, LoRa communication is vulnerable to collisions in the absence of a MAC protocol. The frequency band and the configured channel bandwidth determine how many LoRa channels are available [18].

Scaling to a hundred or more sensor nodes is possible with the LoRa multi-hop mesh network solution. where nodes use coordinated duty cycles to report data to a data sink. Between duty cycles, nodes hibernate in a low-power state to save energy [18]. Adaptive Data Rate (ADR) is a key component of LoRaWAN. By modifying the data rate based on the link budget for each end node in a LoRaWAN, this scheme seeks to minimize energy consumption and maximize throughput. Bandwidth (BW), Spreading Factor (SF), Transmission Power (TP), and Coding Rate (CR) are the transmission parameters that are

controlled by ADR. Because data packets transmitted with different SFs are orthogonal and can thus be received concurrently, optimizing the ADR reduces airtime and increases network capacity. Depending on the link budget, ADR regulates the uplink (UL) transmission parameters of LoRa devices [19].

2.3.3.1. LoRa Modulation Techniques

LoRa employs Chirp Spread Spectrum (CSS), a modulation technique that transforms digital data into analog signals suitable for wireless transmission. A unique method known as chirp spread spectrum (CSS) is employed in LoRa modulation, in which the transmitted signal's frequency varies over time in a distinct pattern known as a chirp. The signal can withstand interference and multipath effects because of its extended duration, which is made possible by this chirping effect. The frequency modulation of the chirp signal is used to encode data in CSS modulation. Long symbols and enhanced interference resistance are the outcomes of LoRa's CSS modulation, which employs a wide bandwidth and a slow chirp rate. LoRa devices can communicate over long distances thanks to this modulation scheme [20].

CSS is renowned for its exceptional adaptability in offering a trade-off between data rate and reception sensitivity. The most crucial CSS modulation parameter is the spreading factor (SF). Although increasing SF results in a lower transmission rate, it can greatly increase the communication range. Another LoRaWAN parameter that can be changed is bandwidth. Using a wider bandwidth, as anticipated, improves the transmission rate while also offering greater protection against narrowband noise and intrusion. [20] provides an assessment of LoRaWAN's system-level performance as well as link performance.

2.3.4. WSN Applications Utilizing LoRa Technology

LoRa technology is revolutionizing wireless sensor networks (WSNs) in a number of industries thanks to its long-range, low-power capabilities. It is tailored for Internet of Things applications and runs on the LoRaWAN protocol, which allows for dependable communication over long distances while using little energy. Key uses of LoRa in WSNs are listed below:

1. Public Transportation

The sensor nodes on public transportation vehicles are mounted on the vehicles themselves; they send position, date, and time data to the gateway along with data from weather and air quality sensors. The information is received by gateways located throughout the city and sent to the supergateway, a central node that handles the processing of the data. The supergateway may report any potential malfunction, display the data, and monitor the proper operation of the network's end nodes and gateways [21].

2. Smart Agriculture

LoRa-based WSNs are used in smart agriculture to decrease human intervention and increase crop yield. Based on crop requirements and soil moisture, systems can regulate water flow while sending data to farmers via mobile devices over great distances. This guarantees timely data delivery and effective water use, both of which are essential for improving agricultural practices [22]. Furthermore, adaptive data rate algorithms that increase data extraction rates and decrease collision rates have been developed to improve the quality of service in agricultural settings [23].

3. LoRa technology in smart home.

Because of its long range and low power consumption, LoRa technology is becoming more and more popular in smart home automation. This makes it

perfect for large properties or rural homes. It makes it possible to easily control and keep an eye on household appliances like door locks, HVAC systems, lighting, and security cameras. Through smartphone applications, it can be integrated with IoT systems to allow remote monitoring and control of household appliances, improving system efficiency and user convenience [24, 25].

4. Smart Energy Management

A developing field that uses the Internet of Things (IoT) to monitor electricity, gas, and water consumption in real time is smart energy management with LoRa technology. This method's affordability, long-range connectivity, and low power consumption make it especially advantageous in places with inadequate infrastructure, like suburban or rural areas. Real-time monitoring, cost-effectiveness and coverage, data transmission and analysis, integration with renewable energy and edge computing, and energy efficiency are some of the main advantages of LoRa in smart energy management [26].

5. Healthcare

These studies indicate that LoRa technology improves patient care through long-range, low-power communication and enables prompt medical interventions in remote and difficult environments by facilitating efficient and dependable remote monitoring of vital signs like heart rate, body temperature, blood pressure, and oxygen saturation. LoRa technology is being used more and more in the medical field to continuously monitor vital signs like blood pressure, heart rate, and glucose levels remotely. Even in remote areas, its long-range connectivity guarantees dependable data transfer, enhancing patient care and facilitating the early identification of health problems. [27].

6. Industrial Automation

In industrial settings, LoRa is very useful, especially for predictive maintenance and machinery monitoring. Large factories or remote locations benefit from its long-range capabilities, which lower operating costs and downtime. In order to facilitate real-time data flows, which are essential for industrial Internet of things applications, several studies have suggested improvements to LoRa's medium access strategies. Reliable, low-latency communication is made possible by these advancements, which is crucial for Industry 4.0 settings [28].

7. Telecommunications

LoRa is perfect for scalable networks in smart city applications because it allows low-data-rate communication for Internet of Things devices. It connects thousands of devices in a single network by effectively sending tiny data packets over great distances. Applications needing wide coverage and low power consumption will find this capability especially helpful [29].

8. Military and Defense

By facilitating real-time surveillance and reconnaissance, LoRa improves military operations. By linking tactical sensor networks for troop monitoring and intrusion detection, it enables secure communication in difficult terrain. In remote or hostile environments, the technology's ability to transmit data without the need for a SIM card or internet connection is especially beneficial [30].

2.3.5. Architectures of LoRa Protocol

In order to improve IoT applications' scalability, energy efficiency, and reliability while tackling issues like latency, collision avoidance, and compatibility, LoRa protocol incorporates a number of multi-hop and real-time

protocols, adaptive data rate mechanisms, and routing strategies. Lora protocol is illustrated in figure 2-2.

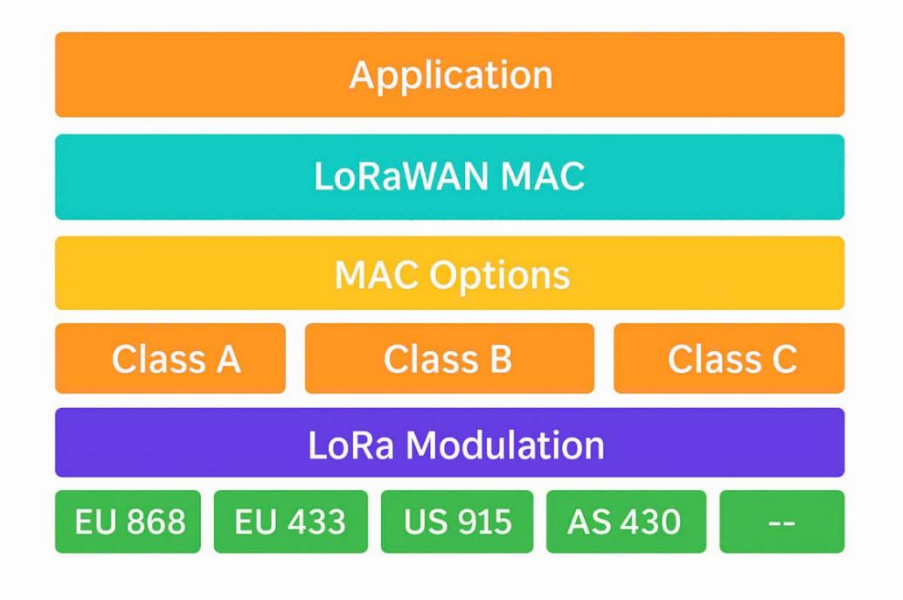


Figure 2-2: Architectures of LoRa Protocol.

I. Physical layer

Noncoherent detection is used at the receiver, and CSS is used on top of a variation of frequency shift keying in the LoRa physical layer (PHY). Even though it provides a good trade-off between coverage, data rate, and device simplicity, some applications are still constrained by its maximum achievable data rate. Furthermore, the CSS generic case that is, when data to be transmitted is encoded in different waveform parameters is not fully utilized by the current LoRa standard. It is crucial to remember that the receiver uses non-coherent detection in this instance, and the discrete chirp rate is set to unity. Furthermore, according to LoRa PHY, the SF is the number of bits—which can vary from 6 to 12 bits that a single symbol can carry [31].

II. Mac layer (media access control)

End nodes, gateways, network, and application servers make up the LoRa networking system. Sensory data produced and sent by dispersed LoRa

end nodes via wireless channels is relayed by gateways before arriving at network and application servers. Multiple gateways can act as forwarders to the network servers for a LoRa packet at the same time. These servers suppress duplicate receptions, schedule acknowledgments, conduct security checks, and modify the network configurations on end nodes and gateways as necessary. The received data is eventually sent to the appropriate application servers for additional processing [33].

III. Network layer

End nodes, gateways, network, and application servers make up the LoRa networking system. Sensory data produced and sent by dispersed LoRa end nodes via wireless channels is relayed by gateways before arriving at network and application servers. Multiple gateways can act as forwarders to the network servers for a LoRa packet at the same time. These servers suppress duplicate receptions, schedule acknowledgments, conduct security checks, and modify the network configurations on end nodes and gateways as necessary. The received data is eventually sent to the appropriate application servers for additional processing [33].

IV. Application layer

To accomplish application-specific and secure end-to-end data delivery, the application layer uses the under-layer transmission primitives. Due to its long-range communication capabilities, LoRa is unavoidably vulnerable to wireless attacks coming from hidden and distant locations. A few security keys that are either generated during the over-the-air activation (OTAA) registration process or pre-installed on end nodes are used to accomplish authentication, integrity, and encryption for the App layer [33].

V. Security architecture

For network security, LoRaWAN is developing integrity, encryption, and authentication. Mutual authentication, confidentiality, and integrity are made possible by the LoRaWAN security policy, which complies with cutting-edge concepts like the application of a standard security algorithm and end-to-end secure communication protocols. As a result, the LoRaWAN security mechanism uses the standardized AES cryptographic algorithm that has been approved by the National Institute of Standards and Technology (NIST). It combines multiple modes of operation, such as a Counter Mode (CTR) and a Cipher-based Message Authentication Code (CMAC), with the original AES encryption/decryption algorithm.

While the latter is used for data encryption, the former is used to safeguard the integrity of messages. When a new device joins, the application session key (AppSKey) and network session keys (FNwkSIntKey, SNwkSIntKey, and NwKSEncKey for R1.1) are generated using a globally unique identifier DevEUI and a unique 128-bit AppKey [34].

2.3.6. Comparison of LoRa with Other LPWAN Technologies

The needs of every IoT application might not be satisfied by a single technology. As a result, some IoT applications are better suited for one technology than another. Making the correct choice enables IoT applications to save money, time, and increase productivity. Along with the suitability assessment of the aforementioned technologies based on the literature, some real-world tests are conducted using off-the-shelf commercial hardware. Both indoor and outdoor LoRa and Sigfox user devices are taken into account in these measurements. The measurements yielded two key performance indicators: the received signal strength indicator (RSSI) and the signal-to-noise

ratio (SNR). Furthermore, penetration loss is calculated from outdoor to indoor environments [35].

LoRa is frequently contrasted with short-range technologies like Wi-Fi, Bluetooth, and Zigbee, as well as other LPWAN technologies like Sigfox and NB-IoT (Narrowband IoT), table 2:2 contains detailed comparisons of Lora and other technologies. Because each technology has unique advantages and disadvantages, it can be used in a variety of ways. We present a thorough comparison of these technologies below, taking into account important parameters like cost, power consumption, data rate, range, and application suitability.

Table 2:2: Comparison of LoRa, NB-IoT, Sigfox, Zigbee, Bluetooth, and Wi-Fi.

Technology	**Range**	**Data Rate**	**Power Consumption**	**Cost**	**Work**
LoRa	10-15 km (rural)	0.3-50 kbps	Very Low	Low	[35]
NB-IoT	1-10 km	Up to 200 kbps	Low	Medium	[36]
Sigfox	10-50 km	100 bps	Very Low	Low	[36]
Zigbee	10-100 m	Up to 250 kbps	Low	Low	[37]
Bluetooth	10-50 m	Up to 2 Mbps (BLE)	Low (BLE)	Low	[38]
Wi-Fi	50-150 m	Up to Gbps	High	Medium	[38]

2.4. TOPOLOGIES OF LORA (LONG RANGE)

Different topologies that balance scalability, power consumption, and range are supported by LoRa networks, as highlighted in figure 2-3, each of which is designed for a particular use case.

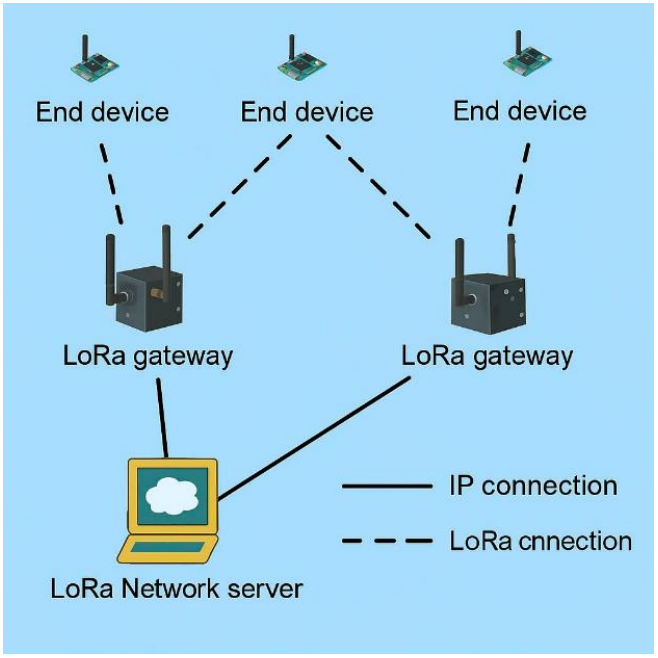


Figure 2-3: Topologies of LoRa (Long Range).

2.4.1. Star Topology

Beyond Topology of Stars Conventional LoRa networks use start topology to link end devices that are positioned around gateways, which limits their coverage area and ability to cover additional end devices. To address this problem, multihop communication based on LoRa features can be used to manage direct device-to-device links. For instance, the spreading factors give the multi-access channel an additional dimension in addition to flexibility with regard to the data rate and sensitivity. The authors of [39] used this new dimension to adopt multi-hop communication, which offloads data traffic into multiple subnetworks, in order to increase the capacity of the LoRa network.

By grouping End Devices according to the SFs, this scheme achieves paralleled transmission (TX); for example, each subnet uses distinct Spreading Factors to communicate its data. Subnet connectivity is taken into consideration when clustering, a balanced traffic load is maintained because of the Spreading Factor asymmetry, and the hop count is minimized to reduce Transmission (TX) airtime [40].

2.4.2. Star-of-Stars Topology

The coverage area of the network is defined by gateways in LoRaWAN deployments with a star of stars topology. As a result, the applications built upon are subject to an unbalanced uplink data path (nodes → gateway → cloud). More adaptable network topologies would be advantageous for use cases that don't entirely fit into this architecture. Adopting network models that permit multi-hop packet transmission between nodes could enable the deployment of certain applications with reduced ownership costs. Infrastructure-less and decentralized IoT systems that must disperse data among nodes dispersed over wide regions and linked by a low-power mesh network in order to carry out computations at the edge may eventually take advantage of this [41].

2.4.3. Mesh Topology

The geographical distribution and accessibility of LoRaWAN's gateways are key factors in determining the network's coverage. LoRa technology facilitates long-distance communication, but it loses effectiveness in large areas where it is not practical or cost-effective to install more gateways. Multi-hop communication, which allows data to pass through several intermediary devices before arriving at its destination, has been investigated by researchers as a solution to this constraint.

Using specialized relay devices, integrating multi-hop functionality at the gateway level, or implementing it within end nodes are the three main multi-hop strategies that have been identified. According to [42], Tian et al. presented LoRaHop, a protocol that uses network coding and Concurrent Transmissions (CTs) to enable effective message relaying without the use of routing tables. In order to minimize interference and lower power consumption, this method strategically takes advantage of idle transmission slots. Similarly, by organizing network nodes into distance-based rings, Islam et al. quoted in [42] expanded the Distance Ring Exponential Stations Generator (DRESG) framework. In order to facilitate an adaptive transmission scheme and improve network scalability and efficiency, their methodology makes use of intermediate gateways.

Lundell et al., referenced in [42], have suggested additional developments by creating a mesh networking protocol for inter-gateway communication. By modifying Ad-Hoc On-Demand Distance Vector (AODV) and Hybrid Wireless Mesh Protocol (HWMP) routing to account for LoRa-specific limitations, their method enables gateways without direct Internet access to relay data through those with backhaul connectivity. Furthermore, a synchronous LoRa mesh network was created by Ebi et al. for the purpose of monitoring subterranean infrastructure, as stated in [42]. In order to bridge the synchronous mesh segment with traditional LoRaWAN gateways, their framework adds repeater nodes, which greatly increases the reliability of packet delivery in environments with limited range. To preserve network coherence and operational efficiency, this approach requires exact time synchronization, such as via GPS or DCF77 time signaling [42].

2.4.4. Point-to-Point Topology

An easy deployment is made possible by a LoRa point-to-point system, as shown in figure 2-4. Since no license or large infrastructure investment is needed to establish point-to-point links, the entry barrier is especially low. It makes it possible to create "Ad-hoc-Networks-of-Things" with extensive coverage that can be kept private if preferred. Large farms in isolated areas, ports, and nature reserves are examples of possible application domains. Furthermore, moving these networks is simple [43].

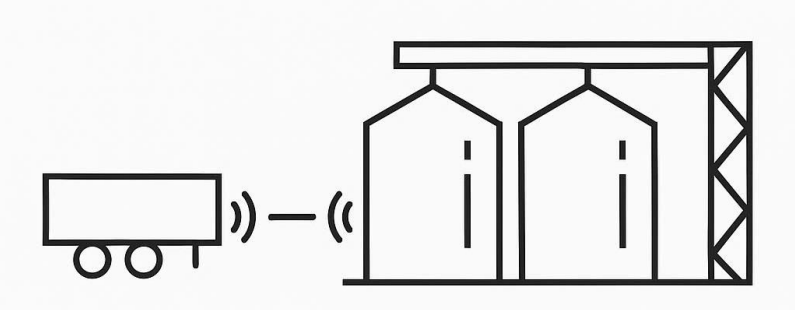


Figure 2-4: Point-to-Point Topology.

Direct communication between two devices without the use of a gateway characterizes the most basic type, known as point-to-point topology. Although it lacks scalability and range, it is dependable for small-scale applications such as remote asset monitoring. For specialized, one-to-one communication requirements, this topology works well.

2.4.5. Hybrid Topology

It is well known that in certain regions of LoRaWAN networks, obstacles or changes in topography prevent end-device signals from reaching the gateway. These regions define shadow areas in the LoRaWAN coverage and prevent end-device nodes from sending data [44]. Figure 2-5 illustrates how is the eorking principle of the hybrid topology.

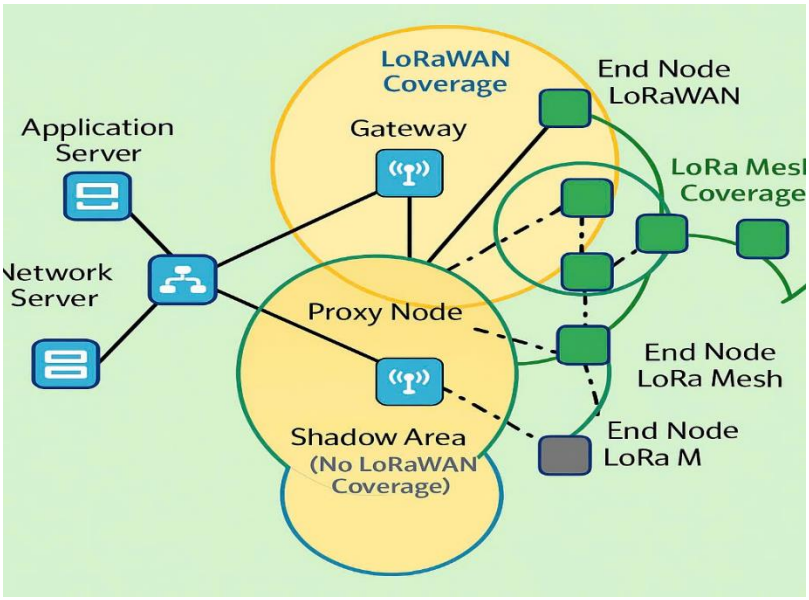


Figure 2-5: Lora hybrid topology.

Several topologies are combined in hybrid topology to capitalize on their individual advantages. It provides flexibility while maximizing fault tolerance, power consumption, and coverage. This method works well in complex settings where devices have different communication needs, such as smart cities or industrial sites.

2.5. LORAWAN NETWORK ARCHITECTURE

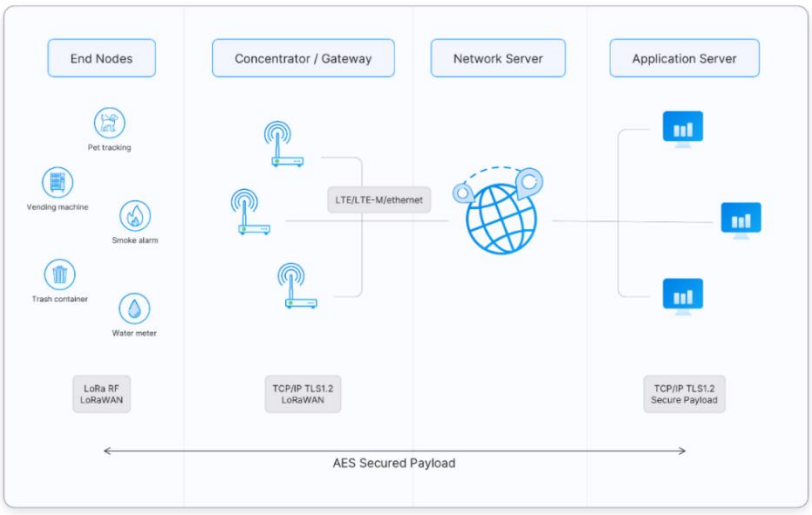


Figure 2-6: A typical LoRaWAN network architecture.

LoRaWAN uses a star-of-stars topology, in which gateways forward data to a central network server after receiving direct communication from end devices. Because it does not require complicated device-level routing mechanisms, this architecture makes routing easier for devices with limited energy, and this is illustrated in figure 2-6. This method has drawbacks, though, especially when devices are outside of the gateway's range or when obstructions prevent line-of-sight communication. Researchers are investigating extensions like mesh and multi-hop networking, which improve coverage and dependability in expansive or difficult environments, to address these issues.

2.5.1. Devices for LoRa Networks

Long-range, low-power Internet of Things communication is made possible by LoRa networks, which are made up of a number of essential devices. Here is a summary of the main elements:

a) Nodes, or end devices

Sensor nodes or Internet of Things devices that gather and send data to gateways are known as end devices. Usually, they are battery-powered, low-power devices with sensors for motion, humidity, and temperature. Because the end device in LoRaWAN doesn't synchronize, it can send data at any time, through any channel, and at any speed. Using various spectrum spreading factors, communication between end devices and the gateway is based on frequency changes that occur during radio transmission, a technique known as frequency shift keying (FSK).

b) Gateways

Gateways serve as a link between the network server and endpoints. They use Ethernet, Wi-Fi, or cellular connections to send the data they receive from various end devices to the network server. If multiple LoRa signals are

transmitted on different channels or with different Spreading Factors (SF), LoRaWAN gateways can demodulate them all at once. The gateways use an SX1301 chip, which has eight demodulator chips, to accomplish this. The great majority of LoRaWAN gateways use the SX1301 component. This chip continuously scans all channels and SFs for preambles. The SX1301's packet arbiter determines whether to configure one of the eight demodulators to demodulate the corresponding frame when it detects a preamble [46].

c) Network Servers for LoRaWAN

The LoRa network server is in charge of protocol analysis and packet processing after receiving packets that have been forwarded from the LoRa gateway. On the LoRa network server, a vast number of functions must be implemented. Examples include scheduling acknowledgements, filtering duplicate uplink packets, determining whether end devices are legal, forwarding application layer data to the application server, and carrying out network management procedures. For effective processing and easy management, a LoRa network server's architecture must be well-designed. In particular, the scalability can accommodate the requirements of supporting large numbers of LoRa Nodes, and the flexibility enables users to develop a variety of applications on already-existing LoRa hardware devices. To assist users in registering, managing, and keeping an eye on their LoRa devices including nodes and gateways a management framework is also required. with the intention of shedding light on how to enhance LoRa network performance [47].

d) Servers for applications

Application layer payload processing, encryption, and decryption are handled by the LoRa Application Server. In order to guarantee data security and boost transmission efficiency, it can support a variety of applications using

different encryption or encoding techniques, like Protocol Buffer. Through Application Program Interfaces (APIs), the Internet of Things cloud offers necessary services, after which users can access applications. Users can control LoRa Nodes and access LoRa network applications from any location using web browsers or smartphones thanks to the LoRa Application Server, which serves as a bridge between the LoRa network server and the IoT cloud [47].

e) IoT Sensors and Actuators

IoT sensors gather information (temperature, humidity, etc.), and actuators carry out commands to open valves or turn on lights, among other things. Rural businesses can be remotely monitored and controlled by LoRa IoT sensors and actuators, which enhances productivity, resource optimization, and operational efficiency.

2.6. PROTOCOLS FOR ROUTING IN LORA

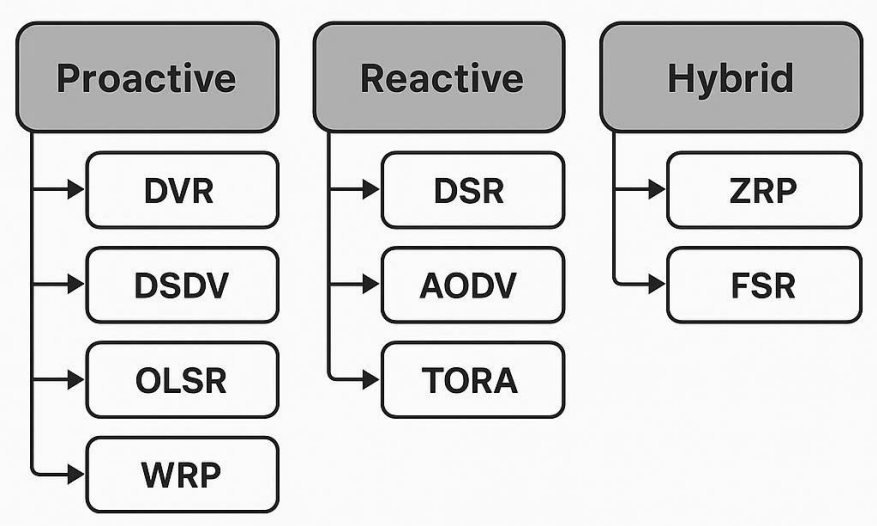
The process of analyzing potential routes from a source node to a destination node and choosing the best one is known as routing. Routing is also in charge of sending data packets across the network. The number of hops, link status, battery charge level, and geographic location are the criteria used to determine the optimal routes. The characteristics of the routing protocols for ad hoc wireless networks, such as the information shared, the manner in which data is shared, or the timing of the sharing, can be used to categorize them. Link state protocols, vector-distance based protocols, energy-aware, multipath, and so forth are a few examples of classifications [48].

Only when required does a reactive protocol, like AODV or DSR, calculate a path to a destination.

Consequently, there may be a delay in path setup before data transmission begins at the source. Topology information is periodically distributed by a proactive protocol, like OLSR or TBRPF. As a result, there are

always routes to every location, but doing so comes at the expense of the overhead required to maintain new routes [51].

The term "hybrid routing protocols" describes the features of proactive and reactive protocols combined to allow for the easy application of both approaches' benefits [48]. Figure 2-7, shows different protocols in Lora.



Figures 2-7: Classification of routing protocols for ad hoc wireless networks.

2.6.1. Protocol for Ad Hoc On-Demand Distance Vectors (AODV)

One example of a reactive protocol is Ad Hoc On-Demand Distance Vector (AODV). This protocol performs better when exposed to frequent topology changes (favoring dynamic routing) because it only finds routes when there is a demand for transmission and the path to the destination is unknown or unavailable. The development of a route discovery and maintenance mechanism based on a request and response model is made possible by such a protocol. Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) are the three primary message types that AODV uses [48].

2.6.2. Routing Protocols Based on DVRs

With the help of the dynamic, distributed, asynchronous, and iterative Distance Vector Routing Algorithm (DVRA) protocol, routers update their routing tables on a regular basis using data shared with nearby routers. Every router builds a Distance Vector Table (DVT), shares it with its neighbors, and computes the shortest path to every other router in the network on a regular basis. This procedure keeps going until a stable Distance Vector Routing Table (DVRT) is obtained, at which point packets are forwarded [49].

Direct point-to-point connections or broadcast subnets like Ethernet or Token Ring are used to connect neighboring routers. The path with the least amount of delay is usually referred to as the "shortest path." Asynchronously sending time-stamped ECHO REQUEST packets and receiving corresponding ECHO RESPONSE packets from neighbors allows routers to estimate distances. The Distributed Asynchronous Bellman-Ford Algorithm is then used to update the DVRT. A fresh DVT is created and distributed to every neighbor if any changes are found. For precise and effective routing, this cycle makes sure that every router keeps an updated DVRT [49].

2.6.3. Protocol for Dynamic Source Routing (DSR)

The process of finding the most effective path between a transmitter and a receiver that is improved by accounting for the energy of the node is known as dynamic source routing protocol. This work presents the Dynamic Source Routing (DSR) Protocol, which uses node energy, node centrality, node degree, and link cost to select relay nodes in order to improve quality-of-service (QoS) by calculating path reliability and link sustainability criteria. Transmission from the source node to the cluster head (CH) is accomplished via qualified relaying nodes. Data packets are sent to their destination via a single node, and the relay nodes are selected as efficiently as possible. Regarding delay, energy usage, and packet loss ratio [50].

2.7. FOUNDATIONS OF THE INTERNET OF THINGS (IOT)

The Internet of Things (IoT) is a revolutionary technological paradigm that uses a vast network of smart devices to link the digital and physical worlds. Fundamentally, the Internet of Things is a huge ecosystem of physically connected objects that are embedded with sensors, software, and network connectivity to allow them to gather, share, and act upon data, table 2:3 describes the different layers in Internet of Things. This network encompasses a wide range of commonplace items, from wearable technology to urban infrastructure, from industrial machinery to home appliances.

Table 2:3: IoT Architecture Layers and Components.

Layer	**Sub-layer**	**Key Features**	**Key Technologies**
Application Layer	IoT Applications	Handheld devices, user interfaces, service delivery	Cloud computing, M2M, middleware, service platforms
	Application Support Layer	Data processing, interoperability	APIs, data analytics frameworks
Transmission Layer	Local and Wide Area Network	Connectivity establishment, data transmission	Wi-Fi, GPRS, 5G, Ad hoc networks
	Core Network	Backbone infrastructure, routing	IP networks, SDN

	Access Network	Last-mile connectivity	LTE, LoRaWAN, Zigbee
Perception Layer	Perception Network	Environment sensing, device communication	RFID, WSN, Bluetooth, NFC
	Perception Nodes	Data collection, actuation	Sensors, actuators, embedded systems
Network Management	Physical & Security Management	Device authentication, trust management	Encryption, PKI, security protocols
	Performance Management	QoS monitoring, network optimization	SNMP, network analytics tools

2.7.1. Layer of Perception

The lowest layer in the Internet of Things architecture is called the Perception Layer, sometimes known as the Device Layer, Sensory Layer, or Recognition Layer. It is essential for connecting the digital and real worlds. This layer includes the technologies that sense the environment, identify objects using unique identifiers, trigger mechanical reactions based on data gathered, and enable communication between diverse smart devices with little assistance from humans. Its main purpose is to digitize and capture information from the real world. According to its functions, the Perception Layer is usually separated into two sub-layers: the Perception Network, which facilitates communication between the sensory elements and higher layers of the Internet

of Things, and the Perception Nodes, also known as Sensory Nodes, which manage data collection and object recognition.

2.7.2. Nodes of Perception

Sensors, actuators, and controllers are examples of physical devices known as perception nodes that engage with their surroundings in order to gather and process data. These nodes support scalable and adaptable deployment and can function in mesh, ad hoc, or multi-hop networks. RFID readers, QR/barcode scanners, Bluetooth, GPS modules, and a variety of environmental sensors (such as those for temperature, humidity, light, and pollution) are among the technologies utilized [52].

Sensing environmental information, object identification, data management, and device behavior control are some of their primary duties. For instance, smart sensors can alternate between low-power and active modes to save energy, and RFID readers use embedded tags to identify objects. In sophisticated applications, intelligent sensing tasks are carried out by microchips embedded in objects using nanotechnology, particularly in situations where direct perception is impractical [52].

2.8. CONCLUSION OF CHAPTER 2

Chapter 2 analyzed the critical issues facing Vietnam's coal mining sector, particularly the limitations of traditional monitoring technologies in hazardous and dynamic environments. It introduced the potential of Industrial IoT, specifically LoRa-based wireless sensor networks, as a strategic solution for real-time environmental monitoring. The chapter explored the technological foundation of the proposed system, detailing the benefits and limitations of ARM microcontrollers, WSNs, and LoRa communication protocols. It also examined system topologies and network architectures suitable for harsh

industrial conditions. This chapter provided a strong technical and contextual background, justifying the need for a hybrid, intelligent monitoring system designed to address Vietnam’s unique mining challenges.

CHAPTER 3. SENSOR CALIBRATION METHODOLOGIES FOR INDUSTRIAL IOT SYSTEMS

3.1. METHODS FOR CALIBRATION OF SENSORS

A thorough analysis of methods for enhancing the precision and functionality of gas concentration sensors, with a focus on semiconductor-based gas sensors, which are renowned for their affordability but are vulnerable to bias, environmental changes, and aging [53].

3.1.1. Linear function calibration

The process of fixing systematic errors (biases) in sensor readings is commonly referred to as calibration. Additionally, the process of mapping sensor raw outputs to standardized units has frequently been referred to by this term. To establish a direct mapping between sensor outputs and expected values, traditional single-sensor calibration frequently depends on presenting a particular stimulus with a known outcome. As a result, a sensor's calibration is frequently subject to certain ranges and operating condition limitations, which are detailed in the sensor's manufacturer specifications [54].

When manual calibration is impractical, the authors of [55] offer a novel method for calibrating dense sensor networks. Using temporal correlation of synchronized measurements, their two-phase method first creates pairwise relationships between nearby sensors.

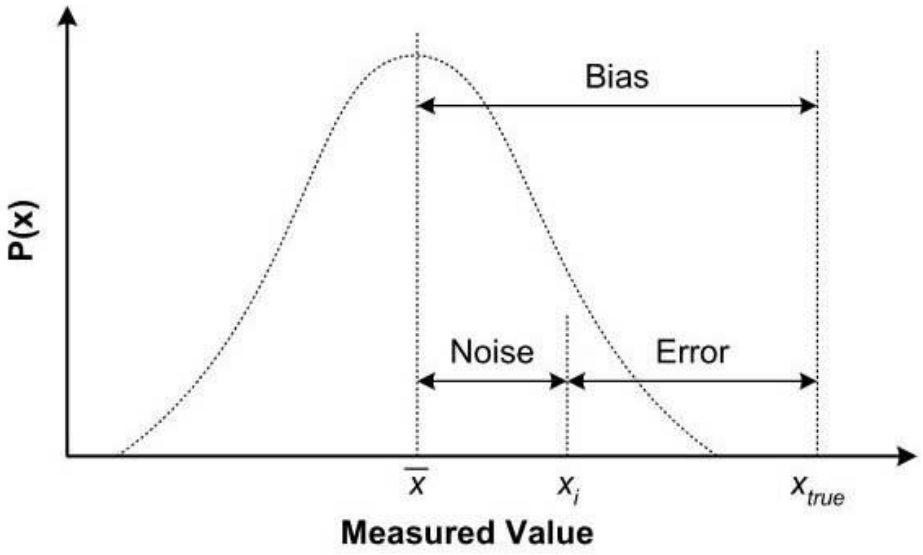


Figure 3-1: Sensor measurement error terminology.

For targeted calibration, figure 3-1 makes a crucial distinction between random noise and systematic bias.

Bias and Systematic Errors: The bias is the difference between the true value xtrue and the mean amplitude of sensor readings x. The bias could be influenced by the environment, the sensed phenomena, time, or other variables [55].

Random Errors (Noise): This error's random component could be the result of hardware noise, other unpredictable transient events, or outside events that affect sensor readings. In certain situations, a particular distribution may be used to model the noise in measurements [55]. The two components of sensor noise a high frequency component and a low frequency component are the main source of inertial sensor random errors. Correlated noise characterizes the low frequency component, whereas white noise characterizes the high frequency component [56].

When detecting the same environmental changes, like changes in temperature or variations in gas concentration, the first step is for nearby sensors to compare their measurements. Figure 3-1 illustrates how bias and

noise are the two main forms of inaccuracy that sensors commonly display [57]. Using straightforward linear relationships, the system determines adjustment factors between sensor pairs. When measuring the same conditions, Sensor A applies a correction formula such as this if it consistently reads 5% higher than Sensor B [55]:

$$F_{\{i,j\}(x)} = a_{\{i,j\}} * x + b_{\{i,j\}}$$

Traversing a path from nodes s1 to s_2 to s_3, then we have:

$$F_{\{2,3\}}F_{1,2}(x)) = a_{\{2,3\}} \cdot \left(a_{\{1,2\}} \cdot x + b_{\{1,2\}}\right) + b_{\{2,3\}}$$

$$= a_{\{2,3\}} \cdot a_{\{1,2\}} \cdot x + a_{\{2,3\}} \cdot b_{\{1,2\}} + b_{\{2,3\}}$$

Two significant conclusions can be drawn from this, as stated in [55]:

1. Prioritizing consistency at the local scale is essential because calibration functions are inevitably built from local measurements. This means that shorter paths should be preferred within the calibration graph (CG), as shown in figure 9, as this reduces the impact of cumulative errors and maintains the reliability of local relationships.
2. It is reasonable to anticipate greater consistency from paths dominated by CFs with higher confidence because each path in the CG is made up of a series of calibration functions (CFs) with different levels of confidence. This implies that in real-world applications, traversals with greater confidence are probably going to produce calibration results that are more reliable and stable.

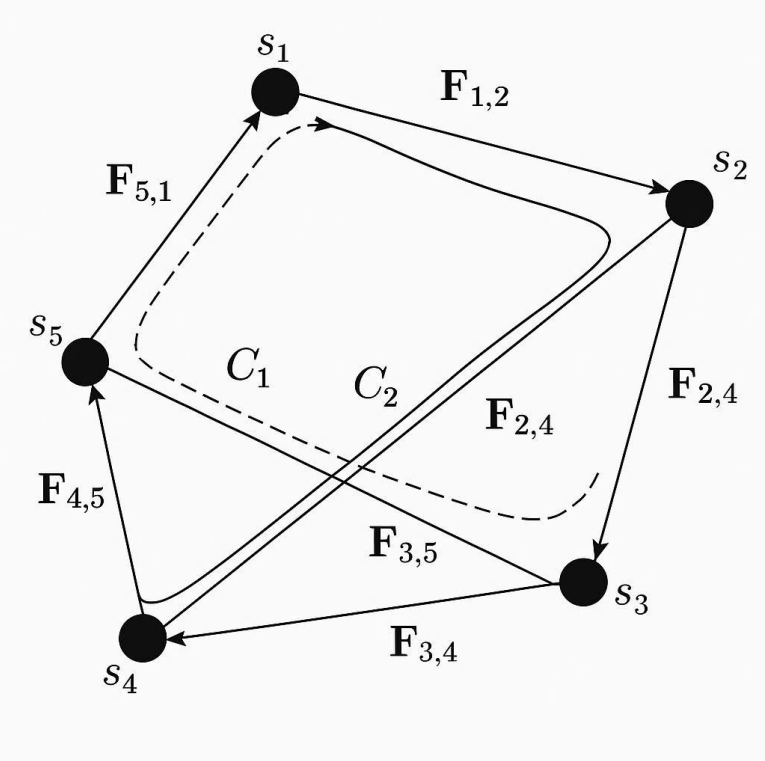


Figure 3-2: Calibration graph.

However, the system includes a second optimization phase because different sensor pairs might recommend slightly different adjustments. This procedure assesses all inter-sensor relationships throughout the network, as shown in Figure 3-2, much like several experts might resolve their divergent viewpoints to come to an agreement. The system determines the most dependable calibration adjustments for the entire network by examining these connections. According to field tests, this method successfully brought 80% of sensors within 5°C of reference measurements, proving its usefulness for real-world applications where practical accuracy is sufficient but absolute precision is difficult [58, 59].

3.1.2. Self-Calibration by Combining Multiple Sensors

Cross-sensitivity compensation is a concept that allows both the interference signal C and the signal to be measured X to alter sensor 1's output

signal. The interference signal C is detected by sensor 2, which compensates for the cross-sensitivity in order to remove the influence of the signal. Reproducibility of sensor 1's cross-sensitivity to the interfering signal C is necessary for this method to work. There may be limited improvements possible if this cross-sensitivity varies significantly over time. The performance can be greatly enhanced by adding sensors where they have a defined cross-sensitivity [59, 60], refer to figure 3-3 for better understanding.

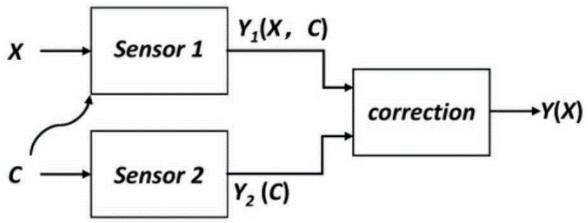


Figure 3-3: Compensation for cross-sensitivity.

From the figure 3-3, we have the calibrated compensation:

$$Y_{corrected} = Y^1 - \left(\frac{\beta}{\gamma}\right) Y^2 \approx \alpha X + \left(\varepsilon - \left(\frac{\beta}{\gamma}\right)\delta\right)$$

Where:

- ✓ The calibrated compensation ratio is denoted by the term (β/γ).
- ✓ Perfect compensation happens when ε,δ → 0 and β/γ is stable.

Using two identical sensors to measure two signals of the same magnitude and opposite phase is known as differential compensation (Figure 3-4 A). The ensuing errors are eliminated in the following computation since the interfering signal C acts on both sensors at the same time. The Wheatstone full-bridge, a common differential compensation circuit, is depicted in Figure 3-4 B. R1, R2, R3, and R4 are all depicted as a collection of strain gauges positioned perpendicular to one another [59, 60]. Temperature-induced changes on the four strain gauges should be identical, as should the absolute values of the resistance changes caused by the object's deformation. A straightforward

calculation demonstrates that the absolute value of the resistance change is proportional to the final output voltage, Vout. The supply voltage Vbias is numerically equal to the sensitivity, and there is no non-linear error [60].

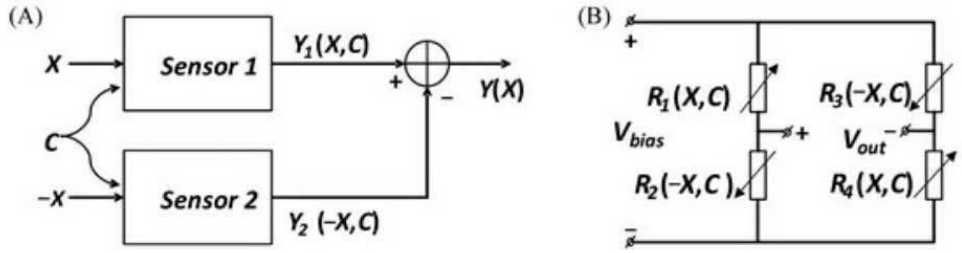


Figure 3-4: Application and compensation that are different. (A) Compensation that is different. Wheatstone Bridge (B).

For Figure 3-4 (A):

$$Y(X) = Y_{1(X,C)} - Y_{2(-X,C)}$$

For figure 3-4 (B), The resistor ratios determine the voltage differential Vout across the bridge:

$$V_{out} = V_{bias}\left(\frac{R_4}{R_2 + R_4} - \frac{R_3}{R_2 + R_3}\right)$$

3.1.3. Frequency-based filtering techniques

Wavelet filtering is regarded as being better than conventional static filters (like low-pass, high-pass, or band-stop filters) among frequency-based filtering techniques [61, 62]. Wavelet analysis is unique in that it can target precise time windows in addition to choosing particular frequency components for filtering. Because of this, it works particularly well with pulsed sensors, in which airflow passes over the sensor surface on a regular basis.

As mentioned in [63], some sensor measurement circuits that use microcontrollers have used correction functions in addition to thermal compensation components to more precisely adjust sensor output.

This formula adjusts the ratio of the sensor's resistance R_s to its baseline resistance R_0, using a linear correction model:

$$\left(\frac{R_s}{R_0}\right)_{\{corr\}} = \left(\frac{R_s}{R_0}\right)(a + b * T)$$

$$\left(\frac{R_s}{R_0}\right)_{\{corr\}} = \left(\frac{R_s}{R_0}\right)(a + b * T + c * RH)$$

- ✓ *RH* is relative humidity;
- ✓ The ***c*** is a constant.

Using the corrected resistance ratio, we can estimate the gas concentration in parts per million (ppm) via a linear relation:

$$ppm = \alpha + \beta \cdot \left(\frac{R_s}{R_0}\right)_{\{corr\}}$$

These models are used to refine the sensor's output to better reflect actual gas concentrations.

3.1.4. Linear and Nonlinear Function Models:

1. Linear 1st-order model:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3$$

2. Quadratic with interaction terms:

$$y = y_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{\{12\}x_1x_2} + b_{\{13\}x_1x_3} + b_{\{23\}x_2x_3}$$

3. Polynomial with squared terms:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{\{11\}x_1^2} + b_{\{22\}x_2^2} + b_{\{33\}x_3^2}$$

4. Comprehensive nonlinear model (including squares and interactions):

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{\{11\}x_1^2} + b_{\{22\}x_2^2} + b_{\{33\}x_3^2} + b_{\{12\}x_1x_2} + b_{\{13\}x_1x_3} + b_{\{23\}x_2x_3}$$

Where:

- ✓ x_1: voltage across the sensor,
- ✓ x_2:load resistance,
- ✓ x_3: temperature.

3.1.5. Piecewise linear approximation

The PLA family of linear approximation techniques uses a series of line segments to represent data samples while maintaining the original samples within a specified approximation tolerance. Actually, the goal of PLA methods is to minimize the energy consumption during data transmission by approximating the time series using a line sequence (each line having only two end points), figure 3-5 validates this hypothesis. PLA results in an efficient representation of time series in terms of memory and transmission requirements because a line segment can be identified by just two end points [64].

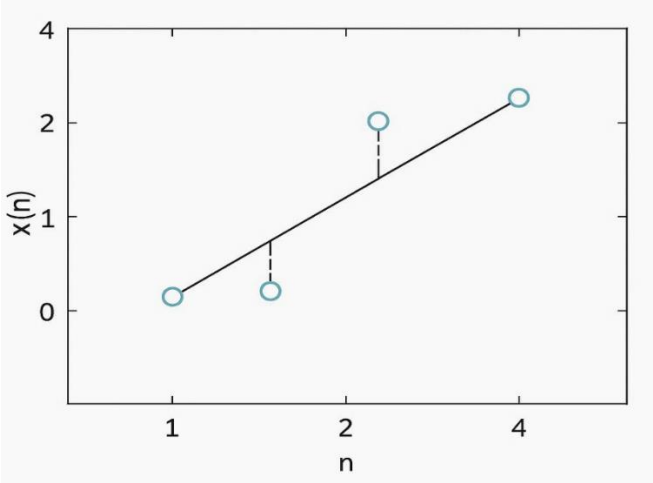


Figure 3-5: Approximation of a time series x(n) by a segment.

a) Receiver-Side Signal Approximation

At the receiver end, the original signal samples are approximated by vertically projecting them onto their corresponding line segments, as illustrated in the figure. The resulting value, denoted as $\hat{x}(n)$, represents the estimated signal. The difference between the original sample $x(n)$ and the projected approximation $\hat{x}(n)$ is referred to as the approximation error, and it is calculated using the following expression [64]:

$$Error = |x(n) - \hat{x}(n)|$$

This method helps simplify the received data while still maintaining a close representation of the original signal.

3.2. IMPLEMENTATION

3.2.1. System design of Monitoring System Using ARM IC and LoRa Module

The STM32 (ARM IC) of multiple sensors, including a MQ2 gas sensor, light sensor, temperature sensor, and humidity sensor, is the basis for this device's slave application. The STM32 processes all of the incoming data while the sensors continuously monitor the surrounding conditions. The entire system is powered by an external source to ensure dependable performance. The E32-TTL-100 LoRa module is used in conjunction with the STM32 to transmit wireless data over long distances. This module enables communication over long distances by transmitting the sensor data to a remote gateway via an antenna, as it is illustrated in figure 3-6.

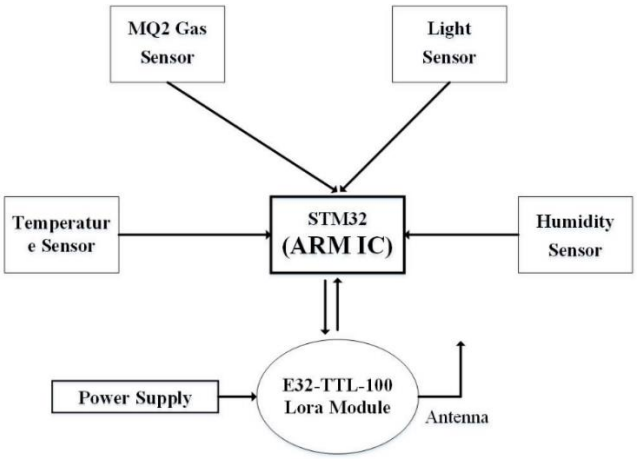


Figure 3-6: This is STM32 (ARM IC)-based slave that collects data from multiple sensors.

Data is sent to the STM32 microcontroller via the E32-TTL-100 LoRa module, which is in communication with the slave nodes. After that, it parses the data it has received and shows it locally on an LCD screen for observation. After that, the STM32 transmits the gathered data to an ESP8266 Wi-Fi module, which forwards it to a Firebase cloud server for cloud-based remote access. This makes it possible for users to monitor sensor data remotely via the internet. It can operate continuously thanks to an uninterruptible power source, figure 3-7.

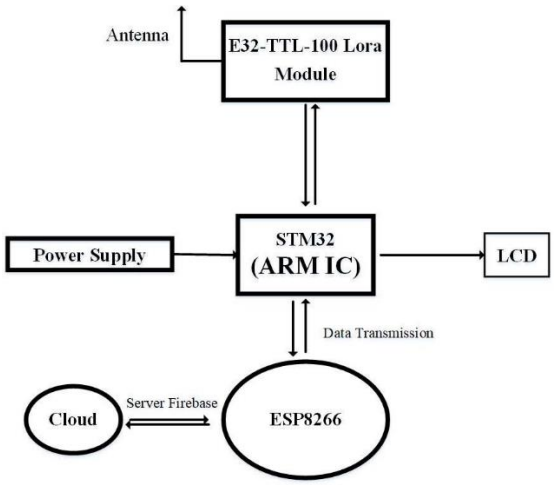


Figure 3-7: This is STM32 (ARM IC)-based Gateway.

3.2.2. Software Design

The gateway flowchart, figure 3-8, that follows illustrates the operation of the STM32-based master node that receives and transmits sensor data. It first initializes the sensor node, then configures the I/O pins, and lastly connects to the Wi-Fi network, which is typically accomplished by the ESP8266 module. For remote monitoring, the final data is uploaded to the cloud (Firebase, etc.). After a successful data transfer, the flow is finished and ready for the following cycle.

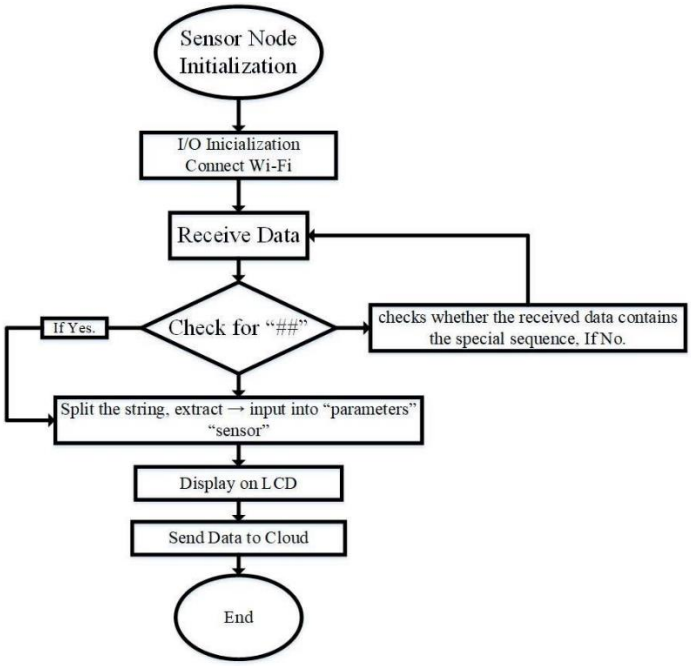


Figure 3-8: Gateway flowchart illustrates the working logic of the STM32-based master node.

The slave flowchart in figure 3-9, outlines how the sensor node should function, gather environmental data, and interact with the gateway. Upon startup, it begins to read the values from the gas, temperature, humidity, and light sensors after initializing peripherals like ADC channels and sensor pins.

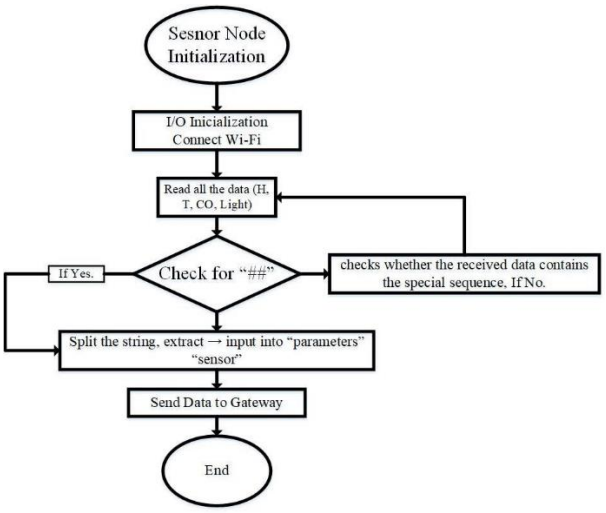


Figure 3-10: Flowchart outlines the operational flow of the sensor node, which gathers environmental data and communicates with the gateway.

3.2.3. Hardware design of the system

a) STM32F103C8T6 (Blue Pill or ARM Cortex-M3)

Additionally, the STM32F103C8T6 (also known as the Blue Pill, figure 3-10) is a low-cost ARM Cortex-M3 microcontroller that is widely used in industrial automation, embedded systems, and the Internet of Things. This guide offers thorough instructions for working with microcontrollers, covering everything from hardware configuration to firmware development, including peripheral interface, communication protocols, optimization methods, etc.

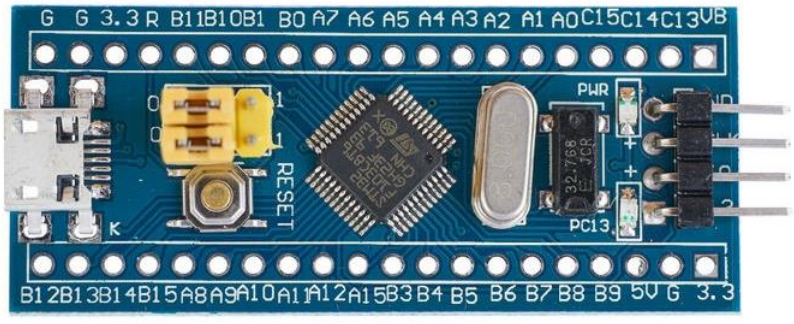


Figure 3-11: STM32F103C8T6 (Blue Pill or ARM Cortex-M3).

Table 3:1: Key Specifications of STM32F103C8T6 (Blue Pill or ARM Cortex-M3).

Feature	**Details**
Core	ARM Cortex-M3 @ 72 MHz
Flash Memory	64 KB
SRAM	20 KB
GPIOs	37 (5V-tolerant inputs)
ADC	12-bit, 10 channels (0–3.3V)
Timers	4x 16-bit, 2x PWM
Communication	2x I2C, 3x USART, 2x SPI, USB 2.0
Operating Voltage	2.0–3.6V (3.3V recommended)

STMF103C8T6 Minimal circuit For stable operation, the STM32F103C8T6 needs a minimal hardware configuration. With decoupling capacitors (0.1 µF for high-frequency noise and 10 10 µF for bulk filtering) near the power pins, a 3.3V regulator (AMS1117-3.3) is essential for powering the MCU. The boot mode is set up with BOOT0 (which is connected to ground for Flash mode and to VCC for System Memory) and BOOT1 (which is typically tied to ground). The reset circuit is made up of a 10K pull-down resistor connected to NRST to reduce noisy power-up conditions. These characteristics make it possible for applications like industrial automation (Modbus, CAN), motor control (PWM, encoders), and Internet of Things sensor nodes (LoRa, Wi-Fi, BLE) to function consistently, as shown in table 3:1.

The STM32F103C8T6 uses USB to work with a single chip, I2C and SPI to interface with peripherals like displays (LCD/OLED), sensors, and serial interfaces (USART) for debugging or communication with LoRa modules. Power efficiency is made possible by low-power modes (Sleep, Stop, and Standby), which reduce current consumption to 2µA in Standby mode (helpful in battery-operated systems). Because of its adaptability, it is perfect for applications where system energy performance and energy efficiency must be balanced, such as wireless sensor networks, real-time control systems, and high performance embedded displays.

3.2.4. E32-TTL-100 LoRa Module

The E32-TTL-100, seen in figure 3-11, is a LoRa wireless transceiving module from EBYTE that primarily operates at 433MHz, though versions that operate at 868MHz and 915MHz are also available. It is designed for long-range Internet of Things applications and connects to well-known platforms (such as Arduino, ESP32, and Raspberry Pi) via UART (TTL). With 3–8 km line-of-sight transmission and multiple operating modes and adjustable parameters, the device offers incredibly low power consumption. Figure 3-12 are the dimmentions of the Lora device.

Figure 3-12: The E32-TTL-100 SX1278 LoRa Module.

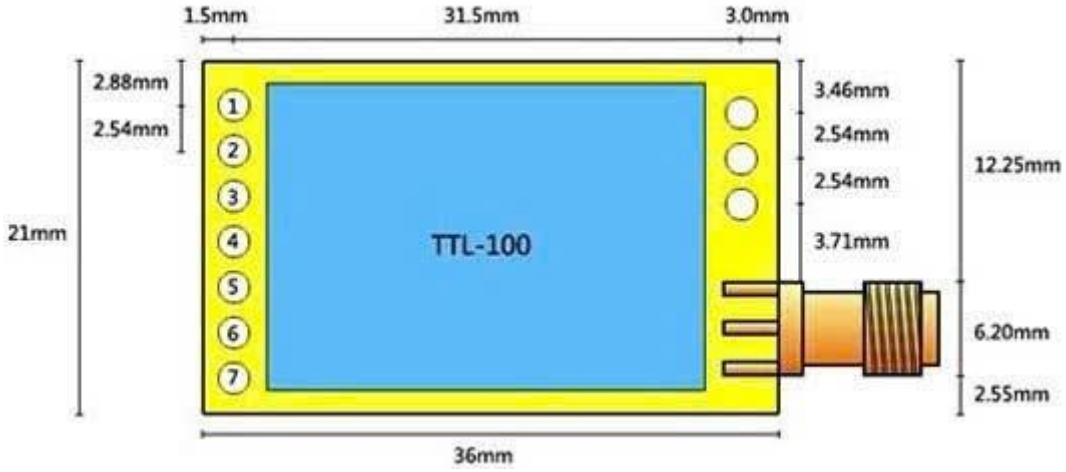


Figure 3-13: Dimensional measurements (in millimeters) for the E32-TTL-100 LoRa module.

a) Data Transmission Modes

The module offers three distinct communication channels:

i. Broadcast Mode is appropriate for weather and other public announcements since it transmits transparently to all devices using the same frequency and network ID.

ii. Transmission using 16-bit addresses with byte-wise destination for addressed communication.

iii. By guaranteeing that receivers sleep until a preamble signal is detected and that transmitters must only be in wake-up mode (M0=0, M1=1) to activate them, the Wake-on-Radio (WOR) mode set in receivers uses the least amount of power possible.

b) Configuration and Operation

By default, the DS021 has a 433MHz, 0x0000, 2.4kbps air data rate, and 9600 UART baud rate; AT commands can alter these settings. Depending on the particular use case, there are four operational modes: Normal Mode for the first real-time bidirectional communication, Wake-up Mode for periodic downlinks that save battery life, Power-saving Mode for remote sensors, and Sleep Mode for times when only configuration needs to be done. The primary

specifications support 3.3V-5V supply voltages, 0.3-19.2 kbps configurable data rates, and adjustable transmission power (up to 20dBm/100mW). The table 3:2, illustrates all the technical references in Lora module.

Table 3:2: E32-TTL-100 LoRa Module Technical Reference.

Category	**Key Parameters**	**Details**
Specifications	Frequency	433MHz (868/915MHz variants)
	Max Range	3–8 km (line-of-sight)
	Transmit Power	20 dBm (100mW), up to 30 dBm
	Interface	UART (TTL, 3.3V–5V)
Pinout	VCC → MCU 3.3V/5V	Do not exceed 5.5V
	TX → MCU RX, RX → MCU TX	Cross-wire connections
	M0/M1 → GPIO	Mode control (see below)
Operating Modes	Normal (M0=0, M1=0)	Default bidirectional mode
	Power-saving (M0=0, M1=1)	Sleeps, wakes to receive (WOR)
	Sleep (M0=1, M1=1)	Lowest power (0.1μA)
Optimization	Antenna	SMA-K (50Ω), high-gain for longer range
	Data Rate	0.3kbps (max range) to 19.2kbps (max speed)

c) Applications

It excels in a variety of Internet of Things applications, ranging from industrial automation (machine monitoring, asset tracking) and smart agriculture (soil moisture sensors, livestock tracking) to smart city infrastructure (street lighting, waste management, and vital communication for disaster notice in remote areas. Networked sensors in battery-powered devices greatly benefit from this technology's long-range capability, low power consumption, and adaptable configuration.

3.2.5. ESP8266 Wi-Fi Module

The Chinese company Espressif Systems developed the ESP8266, an incredibly inexpensive Wi-Fi microcontroller. With a 32-bit Tensilica L106 processor that can operate at 80–160 MHz and embedded 802.11 b/g/n Wi-Fi (2.4 GHz), the chips are ideal for any Internet of Things, home automation, or wireless application. The module's small size, low power consumption, and capacity to function both independently and as a Wi-Fi adapter for other microcontrollers make it a popular choice.

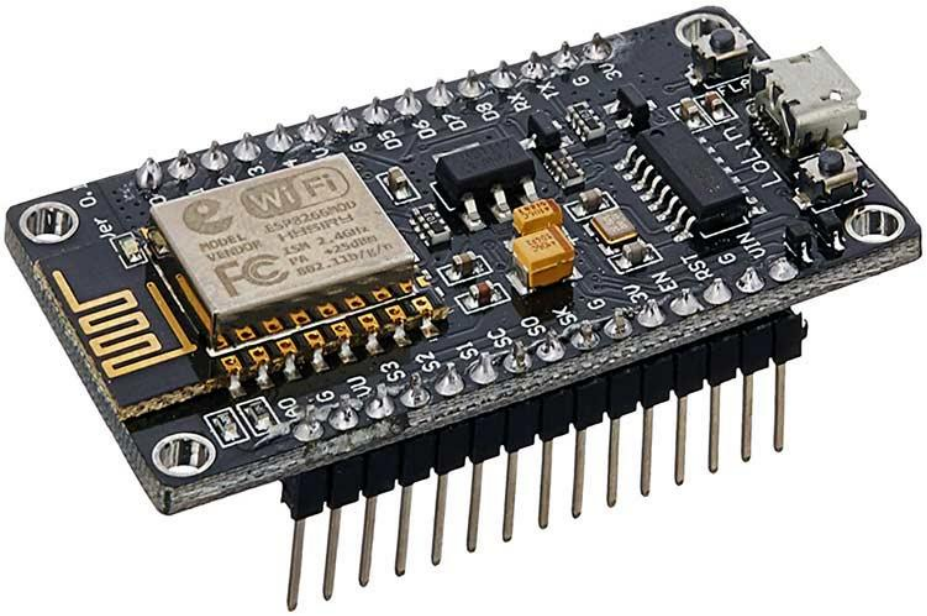


Figure 3-14: Node MCU ESP8266.

The ESP8266, shown in figure 3-13, is a multifunctional Wi-Fi microcontroller with several key features that make it ideal for Internet of Things development. It features a 10-bit ADC (0–1V range) for basic analog sensing, multiple GPIOs (11–17 depending on the module variant) to interface with sensors, LEDs, and relays, and an onboard TCP/IP stack that enables direct internet access. Software I2C, SPI, and UART facilitate peripheral communication, and flash memory (0.5–16 MB) offers sufficient firmware storage. Three Wi-Fi modes are supported by the module: Hybrid (STA + AP), Access Point (AP), and Station (STA).

a) DHT11 Temperature and Humidity Sensor

Hardware: The DHT11 sensor has a thermistor for temperature measurement and a capacitive humidity sensor. Two electrodes and a moisture-adsorbing substrate acting as the dielectric between them make up an example of a capacitive humidity sensor. The capacitance value is impacted by changes in humidity. These fluctuating resistance values are measured and processed by an IC to create a digital signal.

The resistance decreases as the temperature rises because the DHT11 senses temperature using an NTC (Negative Temperature Coefficient) thermistor. The common thermistor is made of semiconductor polymer or ceramic, refer to figure 3-14, to make it more sensitive to even the smallest temperature changes.

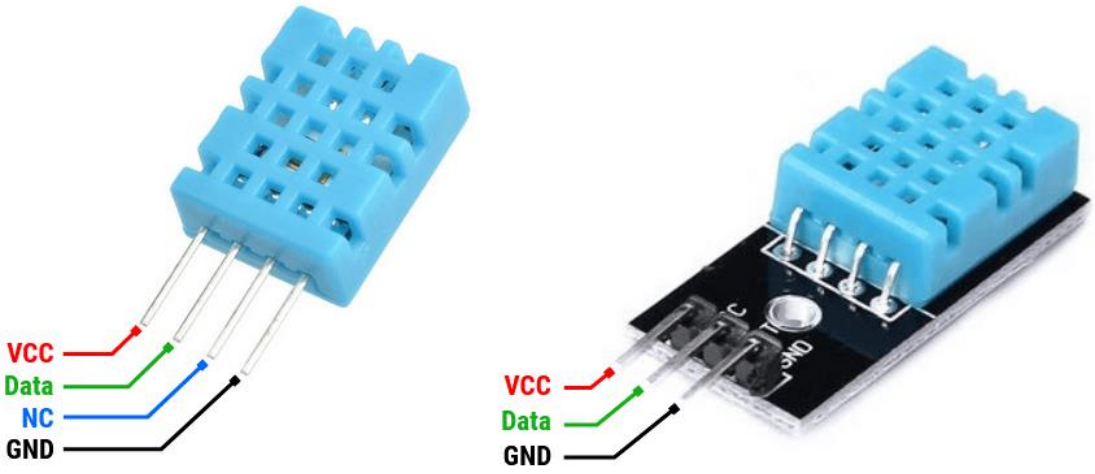


Figure 3-15: Pinout of DHT11.

The majority of asphalt testing manufacturers use NTC (negative temperature co-efficient) thermistors. The resistance fluctuates, as shown in the graph below, indicating that this resistor's operating range is between -100°C and 200°C, meaning that when used, it would be environmentally acceptable. A typical non-linear response curve of an NTC thermistor, figure 3-15, which is used in many temperature sensing, compensation, and control applications due to its high sensitivity and responsiveness to thermal changes, is also described in the application. It is employed to estimate the thermistor's performance in various scenarios.

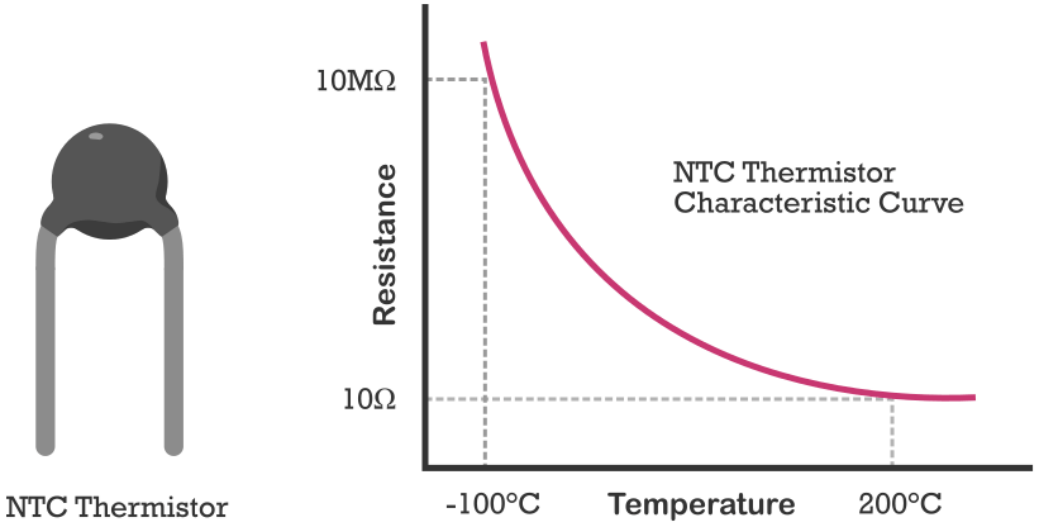


Figure 3-16: An NTC (Negative Temperature Coefficient Thermistor) thermistor's resistance-temperature characteristic curve.

b) DHT11 Key Specifications:

- ✓ Temperature Limit: 0 ~ 50 °C (± 2 °C precision).
- ✓ Humidity Range: 20 to 80% RH (± 5% precision).
- ✓ Sampling Rate: 1Hz (one reading per second)
- ✓ VCC (Supply Voltage): 3V to 5V DC.
- ✓ Power Draw: 2.5mA (Maximum During Measurement).

c) Features of the DHT11 Module:

- ✓ Built-in 5.1kΩ Pull-up resistor–eleminates the complexity of the wiring (unlike the raw DHT11 sensor)
- ✓ Simple interfacing to using 1-Wire communication protocol with microcontroller (comatible microcontroller are Arduino, ESP8266, etc)
- ✓ Onboard signal processing provides calibrated digital output (no external calculations required).
- ✓ Small footprint and 5V DC compatibility is suitable for embedded systems.

The DHT11 is frequently utilized in weather stations, HVAC systems, and Internet of Things projects because of its affordable cost, user-friendliness,

and dependable performance. DHT22 (temperature -40°C to 80°C and ±0.5°C accuracy) is a better option if you require fine grading, as DHT11 is only good for basic temperature sensing. The table 3:3 brings comparisons between the DHT11 sensor with other sensors.

Table 3:3: Comparison with Other Sensors.

Sensor	**Temperature Range**	**Humidity Range**	**Accuracy**	**Communication**
DHT11	0°C–50°C	20–80% RH	±2°C, ±5% RH	1-Wire
DHT22	-40°C–80°C	0–100% RH	±0.5°C, ±2% RH	1-Wire
AM2302	-40°C–80°C	0–100% RH	±0.5°C, ±2% RH	Digital (similar to DHT22)

While the DHT22/AM2302 offers greater accuracy and a wider range for more complex projects, the DHT11 is appropriate for simpler applications.

3.2.6. MQ-2 Gas Sensor

A semiconductor device called the MQ-2 Gas Sensor, figure 3-16, is used to detect smoke and flammable gases like LPG, propane, methane, hydrogen, and alcohol. It is widely used for home automation, industrial safety systems, and do-it-yourself electronics due to its low cost, ease of integration, and broad detection range. It operates on 5V DC and provides both analog and digital data output, allowing for seamless integration with microcontrollers such as Arduino or Raspberry Pi. When exposed to gas, the tin dioxide (SnO_2) sensing layer reacts, changing its resistance and producing detectable voltage changes. It also features an integrated heating coil for improved performance.

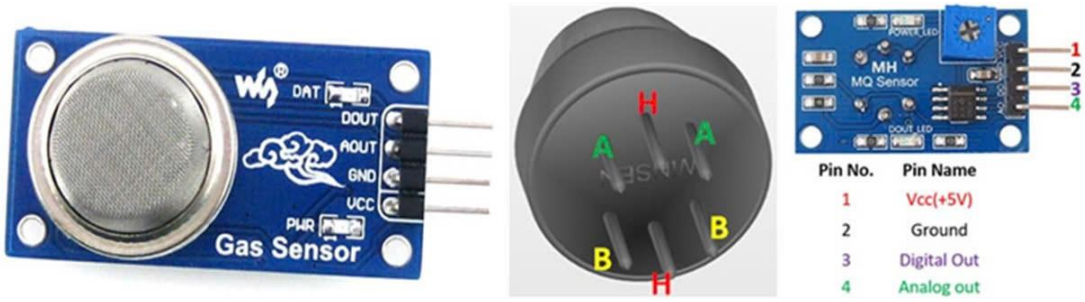


Figure 3-17: MQ2 gas sensor.

The sensor's 5V (VCC) and ground (GND) power sockets are connected in an intermediate manner to the microcontroller's ADC pin. Note: For conditioning analog signals, a 10KΩ load resistor is frequently used. After preheating for 24 to 48 hours to allow the readings to stabilize, an onboard e-pot is used to set the thresholds for calibration. Applications include industrial safety monitoring, fire alarms, and gas leak detection in kitchens; however, the sensor requires adjustment for temperature and humidity changes.

3.2.7. Light Dependent Resistors

The idea of photoconductivity, which states that materials like semiconductors alter their conductivity in response to varying light levels, is where LDRs fall short. A thin strip of photoconductive material on a ceramic base, shielded by a transparent covering, is the standard configuration for their straightforward design. This eliminates the need to power anything or remove any complex circuits, making it easy to measure with an ohmmeter. For the majority of slow-changing and static light experiments, LDRs are a more practical choice because, despite their ability to measure light directly, photodiodes and phototransistors depend on external circuits.

According to the theory of photoconductivity, which states that materials like semiconductors alter their conductivity in response to varying light levels, LDRs perform poorly. They usually have a thin strip of photoconductive material on a ceramic base that is covered in a transparent layer. Their design

is straightforward. This eliminates the need for complicated circuits or powering anything, making it easy to measure with an ohmmeter. Since photodiodes and phototransistors require external circuits to measure light, LDRs are a more practical choice for the majority of static and slowly changing light experiments, see figure 3-17.

Figure 3-18: Light Sensor.

When employing an LDR, the microcontroller's ADC pin is read to convert the small change in resistance into a sufficient analog voltage via a voltage divider circuit. On the one hand, you never have to worry about calibration because the BH1750 offers more advanced signal processing than any multi-sensor solution could, and its results are reliable under a wide range of circumstances. LDRs are excellent for low-budget projects, but situations requiring high precision are better suited for the BH1750.

3.3. HARDWARE DESIGN: MASTER AND SLAVE NODE ARCHITECTURE

Table 3:4: Master Node Pin Configuration.

STM32 Pin	ESP8266	BUZZER	LCD + I2C	LoRa E32
PA9	RX			
PA10	TX			
PB6			SCL	
PB7			SDA	
PA0		+		
	D2			TX
	D3			RX
VCC	VCC	VCC	VCC	VCC
GND	GND	-	GND	GND

Table 3:5: Master Node Pin Configuration.

STM32 Pin	Connected To	Function
PA9	ESP8266 RX	UART Communication
PB6	LCD SCL	I2C Clock Line
VCC	All Modules	Power Supply (3.3V/5V)

The master node's hardware architecture, which uses an STM32 microcontroller as the CPU It includes an LCD with an I2C interface for local data display, a buzzer for alerts, an ESP8266 Wi-Fi module for cloud connectivity, and a LoRa E32 module for long-distance data transmission from slave nodes. Annotations for power distribution (VCC/GND) and pin

connections (PB6-PB7 for I2C with LCD, PA9-PA10 for UART with ESP8266), as refers the table 3:4 and 3:5.

Table 3:6: Slave Node Pin Configuration.

STM32 Pin	**DHT11 (Temp & Humid.)**	**MQ2 (Gas)**	**LDR(Light)**	**LoRa E32**
PB0				
PA0	DATA	A0		
PA1			OUT	
PA9				RX
PA10				TX
VCC	VCC	VCC	VCC	VCC
GND	GND	GND	GND	GND

Table 3:7: Slave Node Pin Configuration

STM32 Pin	**Connected To**	**Function**
PB0	DHT11 DATA	Digital Sensor Input
PA0	MQ2 A0	Analog Gas Detection
PA9	LoRa E32 RX	LoRa UART Receive

In table 3:6 and table 3:7, an STM32 microcontroller and environmental sensors (DHT11, MQ2, and LDR) were used in the hardware design of the slave node schematic. These sensors interfaced with the master node via an uplink LoRa E32 module and measured temperature/humidity, gas detection, and light intensity, respectively. Power rails (VCC/GND) and pin mappings (PB0 - DHT11 data; PA0 - MQ2 analog output) are clearly marked.

3.3.1. Arduino IDE and Basic Programming Structure

To program Arduino boards effectively, one of the first steps is to install the Arduino IDE (Integrated Development Environment, figure 3-18) a user-friendly platform designed to write, compile, and upload code to Arduino-compatible boards. The interface is clean and intuitive, making it accessible even to beginners, while still offering powerful features for experienced developers.

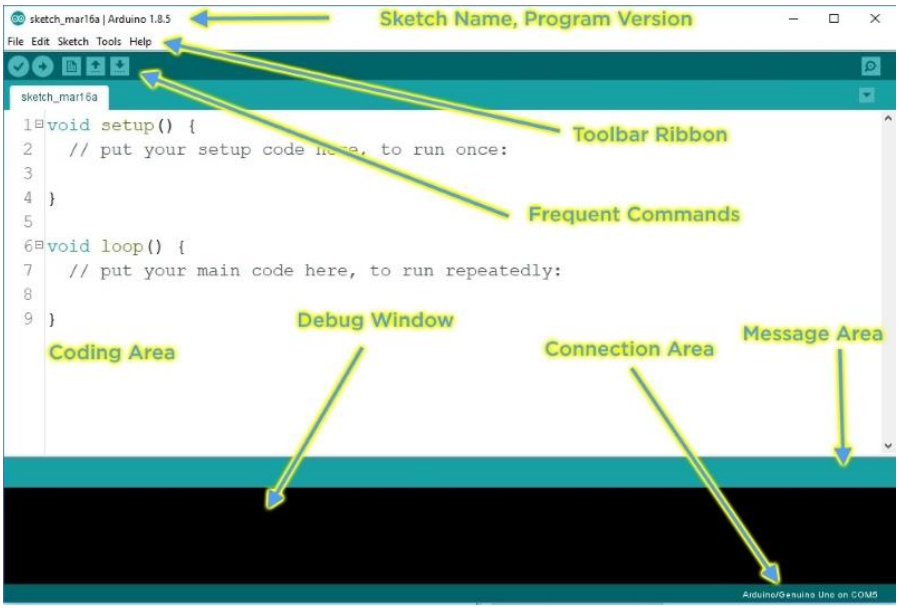


Figure 3-19: Arduino IDE interface and key functional areas.

The main parts of the Arduino IDE interface are illustrated in Figure 27. Key components include:

- **Verify button**: Checks your code for errors without uploading it.
- **Upload button**: Compiles your code and sends it to the connected Arduino board.
- **Serial Monitor**: A tool for displaying data sent from the Arduino to the computer using the Serial.print() or Serial.println() functions.
- **Code editor**: Where you write your programs (also called “sketches”).
- **Message area**: Displays status messages, compile errors, or upload confirmations.

Another important feature is the “Examples” menu, figure 3-19, located under File > Examples, which provides ready-to-run sample programs. For instance, the popular “Blink” example is often used to test new boards by blinking an LED.

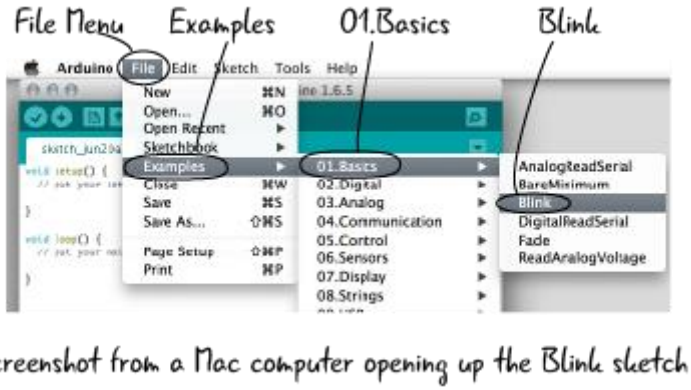


Figure 3-20: Selecting board type and COM port in Arduino IDE.

When connecting an Arduino board for the first time, you must select the correct Board Type and Port via the Tools menu. This ensures the IDE communicates with the correct hardware. If you're using an Arduino Uno, the IDE usually selects it by default, but for other boards, you’ll need to adjust the settings manually. Once the board is connected, selecting the correct COM port is crucial, see Figure 3-19 for an example of selecting a COM port.

3.3.2. Program Structure Basics

Every Arduino sketch follows this template, figure 3-20:

```
// SECTION 1: VARIABLE DECLARATIONS
#define LED_PIN 13   // Assigns pin 13 to "LED_PIN"
int sensorValue = 0; // Creates integer variable

// SECTION 2: SETUP (RUNS ONCE)
void setup() {
  pinMode(LED_PIN, OUTPUT); // Configures pin as output
  Serial.begin(9600); // Starts serial communication
}

// SECTION 3: MAIN LOOP (REPEATS)
void loop() {
  digitalWrite(LED_PIN, HIGH); // Turns LED on
  delay(1000);                 // Waits 1 second
  digitalWrite(LED_PIN, LOW);  // Turns LED off
  delay(1000);                 // Waits 1 second
}
```

Figure 3-21: Program Structure Basics.

a) Key Commands (Table 3:8):

Table 3:8: Fundamental Arduino Commands.

Command	**Description**	**Example**
pinMode()	Sets pin as INPUT/OUTPUT	pinMode(2, INPUT);
digitalWrite()	Writes HIGH/LOW to pin	digitalWrite(13, HIGH);
analogRead()	Reads 0-1023 from analog pin	sensorValue = analogRead(A0);
Serial.begin()	Starts serial communication	Serial.begin(9600);
if-else	Conditional logic	if (temp > 30) { fanOn(); }

3.3.3. MIT App Inventor and Its Interface

MIT App Inventor is a free, open-source web application originally developed by Google and now maintained by the Massachusetts Institute of Technology (MIT, figure 3-21). Designed especially for beginners, it allows users to create Android apps using a visual, block-based programming interface, no prior coding experience required. Users simply drag and drop code blocks and components to build functional mobile applications that can run on Android devices.

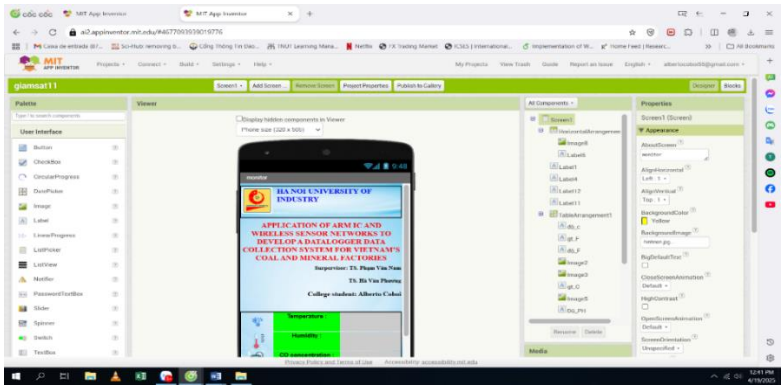


Figure 3-22: Programming interface on App Inventor.

The platform's primary goal is to empower students, educators, and anyone new to programming to create meaningful and useful mobile apps. Since 2017, developers have also begun testing a version for iOS, making it even more accessible across platforms.

MIT App Inventor is especially helpful for educational projects, games, and basic utilities. Some users start with simple ideas like a “Whack-a-Mole” style game or a drawing tool. Others build more advanced applications that make use of the phone's sensors, like tilting the device to move a ball through a maze. Thanks to Android's built-in text-to-speech capabilities, it's even possible to make study apps that can read questions aloud, figure 3-22 ilustrates the building of the logic function of our app.

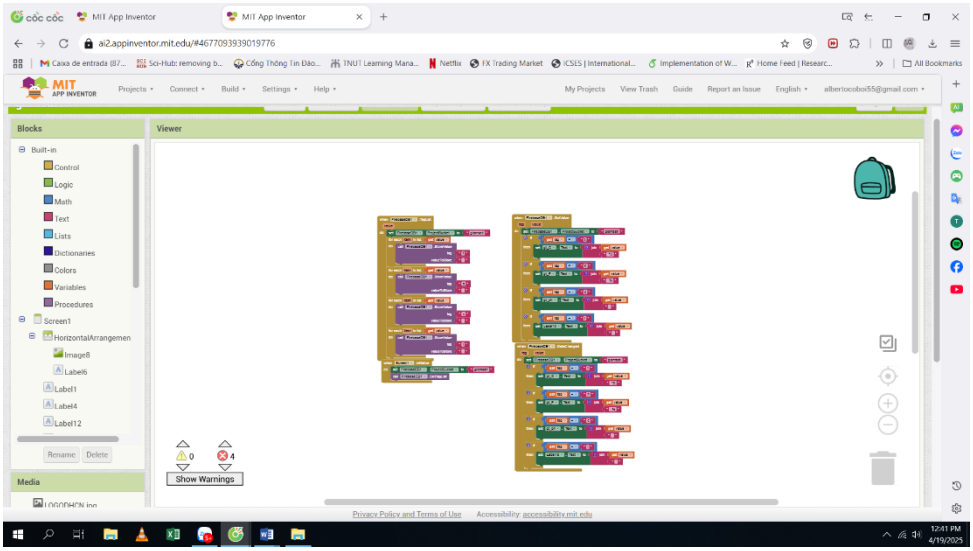


Figure 3-23: Blocks View for building the logic and functionality of the app.

a) Using the MIT App Inventor Web Interface

To start building apps, users can visit http://ai2.appinventor.mit.edu, log in with a Google account, and create a new project. The platform has two main modes:

- **Designer View** – for designing the app’s layout and user interface
- **Blocks View** – for building the logic and functionality of the app

At the top of the interface (highlighted in green), users will see controls for managing screens, switching between Designer and Blocks views, and naming their project, table 3:9, the interface columns.

b) Designer View (Interface Design)

Table 3:9: The Designer interface is split into five main columns.

Panel	**Function**
Palette	Contains common UI components such as Buttons, Labels, and Textboxes.
Viewer	The drag-and-drop workspace where components are placed visually
Components	Lists the components used in the project and their arrangement
Media	Lets users upload custom media files (e.g., images, sounds)
Properties	Allows editing the appearance and behavior of selected components

To test applications built with MIT App Inventor, users can choose from four connection methods, depending on the devices and network setup available. The most recommended method is using a wireless connection through the MIT AI2 Companion App, which is available for free on the Google Play Store. This allows real-time testing by scanning a QR code within the App Inventor interface, requiring no software installation on the computer.

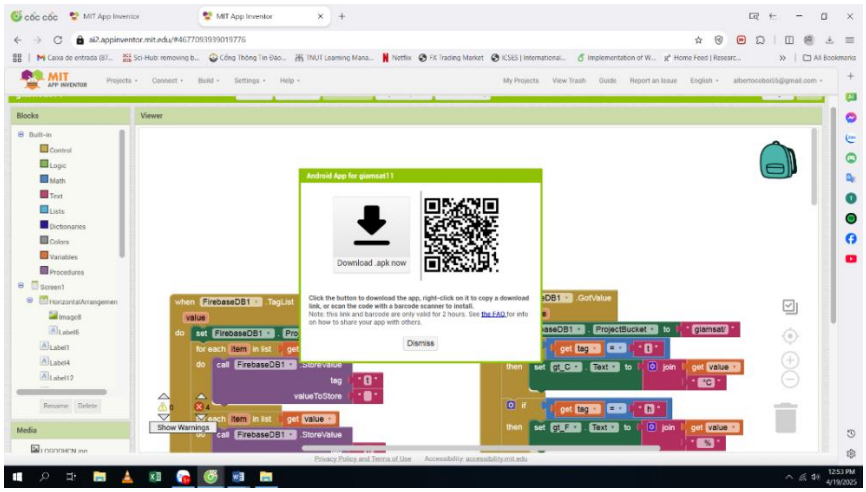


Figure 3-24: Simulation on phone screen.

Alternatively, users working on a Chromebook can also use the Companion App for testing, provided the device supports Android apps, figure 3-23. For those without access to an Android phone, MIT App Inventor offers an emulator that runs on the computer, allowing users to simulate their app's behavior without any external devices. Though slower, it is a useful substitute for hardware.

If Wi-Fi is not available, the final option is to connect an Android device to the computer via USB. This method is more complex, often requiring additional drivers and setup steps, especially on Windows systems. As such, USB connection is considered a last resort and is generally used only when other methods are not viable.

3.3.4. Firebase Integration

i) Introduction to Firebase

Firebase is a cloud-based platform developed by Google that provides powerful backend services for mobile and web applications. At its core, Firebase offers real-time database management, making it easier for developers to handle data operations through simplified APIs. Designed to support both Android and iOS platforms, Firebase has become a go-to choice for millions of

developers worldwide thanks to its flexibility, scalability, and strong security features.

Originally launched in 2012 as a Backend-as-a-Service platform (with roots in an earlier tool called Envolve), Firebase was acquired by Google in 2014. Since then, it has evolved into a full-featured development platform that supports real-time synchronization, authentication, hosting, analytics, machine learning, and more. Firebase, figure 3-24 simplifies the development of connected apps by enabling real-time communication between clients and servers, making it ideal for mobile, IoT, and cross-platform applications.

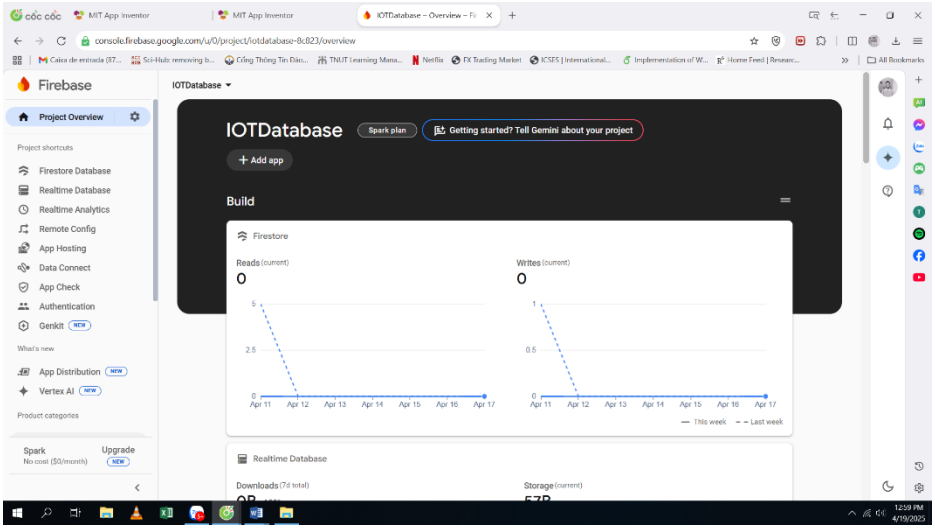


Figure 3-25: Firebase Integration for real time database.

3.3.5. Core Features of Firebase

One of Firebase's standout features is the Realtime Database, which stores data in JSON format and syncs it across all connected devices instantly. Any changes to the database are immediately reflected on all clients, even across different platforms. It also provides robust offline support, data is cached locally and synced once the device reconnects. All communication occurs over secure SSL connections, ensuring user safety.

Firebase also offers Firebase Authentication, a system that simplifies user login using various providers like Google, Facebook, Twitter, GitHub, email/password, or even anonymous sign-in. This helps app developers protect user data and maintain account security with minimal setup.

Another key service is Firebase Hosting, which allows developers to host web content and APIs quickly using secure SSL and CDN (Content Delivery Network) distribution, ensuring fast and secure delivery.

a) Advantages of Firebase include:

- Easy setup and integration
- Real-time data syncing
- No need to manage servers
- Multi-platform support (Android/iOS)
- Built-in authentication
- Hosting and storage options
- Machine learning capabilities
- Google-backed reliability and performance

b) Connecting Firebase to MIT App Inventor

Connecting MIT App Inventor to Firebase allows developers to create interactive mobile applications that can read from and write to the cloud in real time. For example, in an IoT system that measures environmental data (like temperature, humidity, and CO_2 levels), Firebase acts as the central hub that stores and transmits data between devices and the mobile app.

To get started, users must create a Firebase project at firebase.google.com, set up a Realtime Database, and configure read/write permissions (read: true, write: true). This allows components like ESP8266 microcontrollers to send and receive data. From the Firebase settings,

developers obtain a Web API Key used to connect MIT App Inventor to the Firebase database.

Within App Inventor, the app interface is built using visual elements like Labels to display temperature, humidity, and gas readings. Using the Blocks editor, control logic is created to retrieve data from Firebase and display it in real time. Developers must copy the Firebase Database URL into the FirebaseURL component within the app.

After building the app, the ".apk" file is generated and can be installed on Android devices by scanning the provided QR code. Once installed, the app reads live data from Firebase and reflects the values on-screen. To enable data transmission from hardware to Firebase, the ESP8266 module must be programmed using the Arduino IDE with the appropriate libraries (ArduinoJson and FirebaseArduino).

3.4. CONCLUSION OF CHAPTER 3

Chapter 3 focused on ensuring the accuracy and reliability of environmental sensor data through calibration techniques. It discussed various methodologies including linear calibration, cross-sensitivity compensation, and frequency-based filtering, as well as both linear and nonlinear modeling approaches. The chapter also introduced piecewise linear approximations and their relevance in energy-constrained systems. In addition, it covered the system's implementation using STM32 (ARM IC), LoRa modules, and sensor nodes, alongside software tools like the Arduino IDE, MIT App Inventor, and Firebase for real-time data display and storage. This chapter laid the groundwork for creating a precise, resilient, and efficient IoT-based monitoring platform suitable for industrial environments.

CHAPTER 4. RESULTS AND DISCUSSION

4.1. SYSTEM TESTING

The following figure 4-1 and figure 4-2 show one of these devices in its physical form. While the internal section handles the sensor and communication connections, the external section is responsible for data visualization.

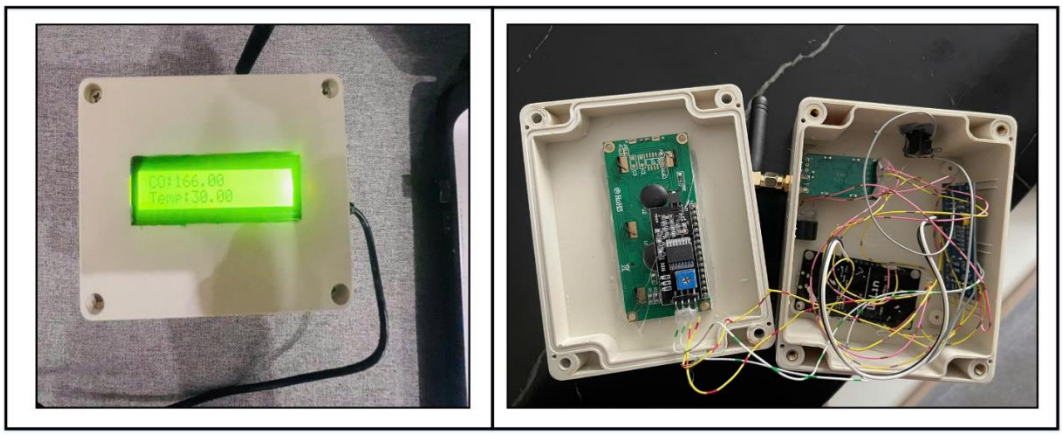


Figure 4-1: Model and location of system devices for the Master Node and showing the Measured value results.

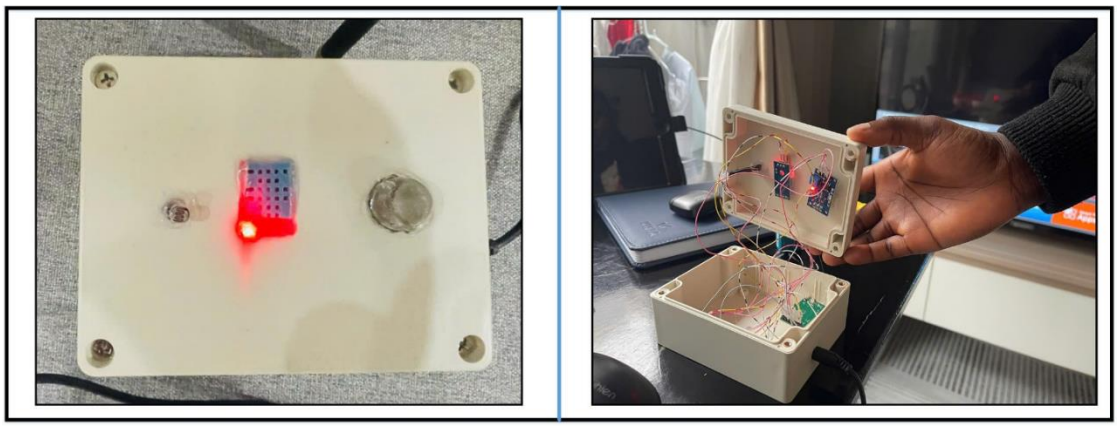


Figure 4-2: Model and location of system devices for the Slave Node.

Following the demonstration of the physical devices, we will turn our attention to the application interface results and their visual representation. When comparing the actual temperature readings with those recorded by the device, it is clear from the result tables shown for each of the three days that

the system always operates with a comfortable margin of error. Furthermore, Figure 4-3 illustrates how the collected data is organized and kept in the IoT Firebase cloud database.

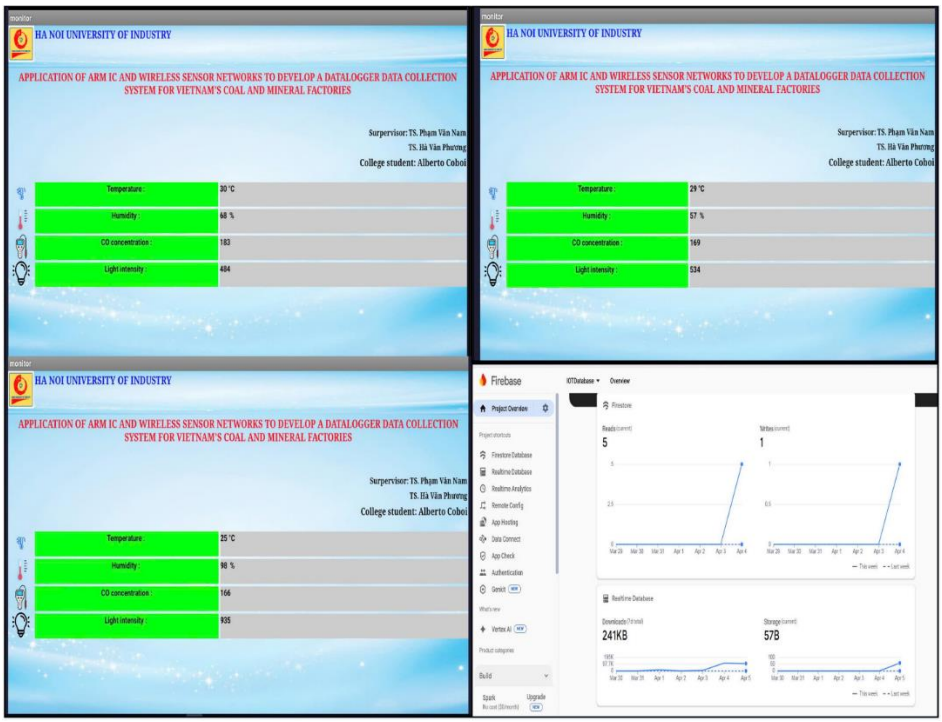


Figure 4-3: Using an app interface, temperature, humidity, CO2, and light intensity were monitored over the course of three distinct experiment days. A lines chart displaying the values of the data stored in Firebase is also included.

Tables 4:1, 4:2, and 4:3, provide a summary of the data that was gathered over the course of three days at various times of day. Temperature values generally vary with the time of day, as evidenced by the data collected. The measurement error margin fluctuates based on environmental factors and sensor sensitivity.

Table 4:1: Day 1 Results.

Time	Actual Temp (°C)	Measured Temp (°C)	Absolute Error (°C)	Relative Error (%)	Humidity (%)	CO (ppm)	Light (lux)
09:00 AM	27.0	27.3	0.3	1.11%	60	5	300
12:00 PM	30.5	31.0	0.5	1.64%	58	6	520
03:00 PM	32.0	32.5	0.5	1.56%	57	5	480
06:00 PM	28.0	28.1	0.1	0.36%	61	4	120

Table 4:2: Day 2 Results.

Time	Actual Temp (°C)	Measured Temp (°C)	Absolute Error (°C)	Relative Error (%)	Humidity (%)	CO (ppm)	Light (lux)
09:00 AM	26.5	26.8	0.3	1.13%	62	5	310

12:00 PM	31.0	31.4	0.4	1.29%	59	7	540
03:00 PM	33.0	33.6	0.6	1.82%	56	6	500
06:0 PM	28.5	28.3	0.2	0.70%	60	5	130

Table 4:3: Day 3 Results.

Time	**Actual T°C**	**Measured T°C**	**Absolute Error (°C)**	**Relative Error (%)**	**Humidity (%)**	**CO (ppm)**	**Light (lux)**
09:00 AM	27.5	27.9	0.4	1.45%	63	4	320
12:00 PM	32.0	32.7	0.7	2.19%	58	6	560
03:00 PM	34.0	34.5	0.5	1.47%	57	6	510
06:00 PM	29.0	28.7	0.3	1.03%	62	5	150

Visualizing trends in the specified environmental parameters temperature, CO concentration, light intensity, and humidity is the first step. Three consecutive days' worth of data were gathered at 4-hour intervals: 9:00

AM, 12:00 PM, 03:00 PM, and 6:00 PM. On Day 2, the CO levels rose to approximately 7 parts per million at around 12:00 PM. This is a sign of activities that take place during the day or changes in the surroundings brought on by the diurnal patterns of temperature and light levels. Such trend data actually makes it easier to identify patterns and provides a fairly good overview of environmental behavior over a longer timescale. Environmentally conscious decision-making systems may find this helpful.

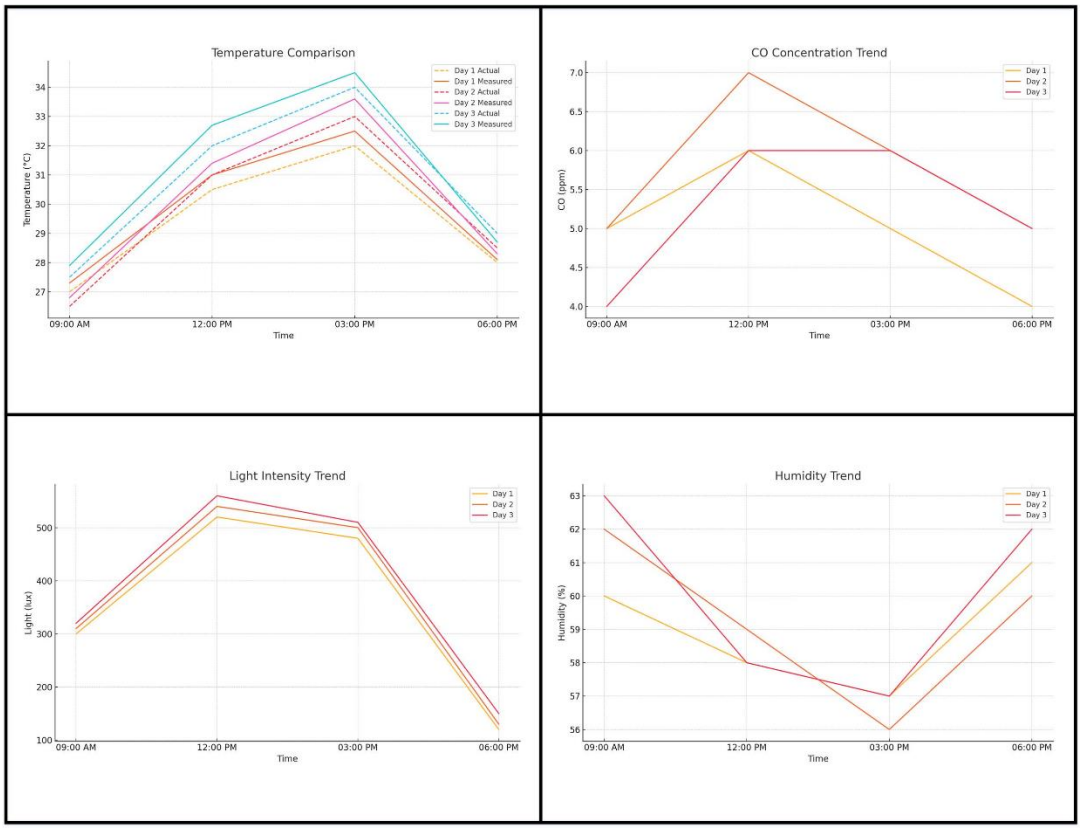


Figure 4-4: The for quadrants show the daily trend summaries of environmental variables.

CO concentration, humidity, and light intensity measured in real time by a sensor (with a 60-second interval), as pictured in the figure 4-4. These graphs display the sensing system's immediate response to its surroundings, emphasizing short-term fluctuations that are typically ignored in aggregated time-series trend analysis. Given the trend data, it is crucial to keep in mind that

these real-time readings serve to both validate the sensing system and demonstrate how the raw data adds to longer-term environmental insights. Figure 4-5 illustrates the continuous real-time sensor readings.

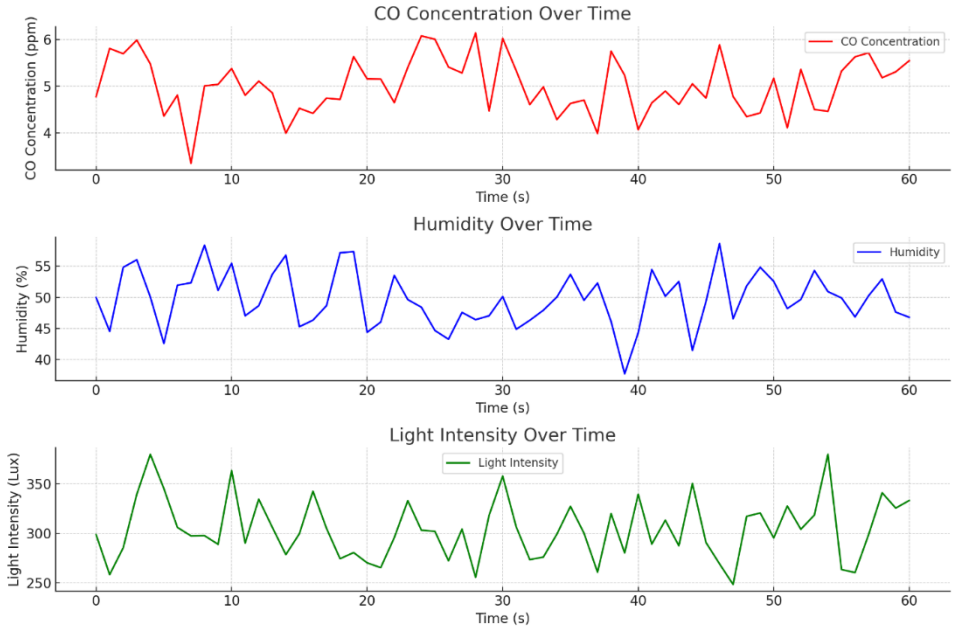


Figure 4-5: Continuous real-time sensor readings.

4.2. CONCLUSION OF CHAPTER 4

Chapter 4 presented the practical validation of the proposed datalogger system through experimental testing. The test results demonstrated the system's ability to monitor key environmental parameters temperature, humidity, CO gas concentration, and light intensity with reliable transmission over long distances using LoRa. Data visualization via LCD and remote access via Firebase confirmed the system's dual communication capability. The real-time monitoring functioned effectively over multiple days of testing, showing consistency and accuracy. The chapter concluded that the developed solution is both technically feasible and practically applicable for deployment in Vietnam's coal plants, offering enhanced safety, data availability, and operational efficiency.

CHAPTER 5. CONCLUSIONS AND FUTURE DEVELOPMENT

5.1. CONCLUSION

Our IoT-based environmental monitoring system, which was piloted for three days, integrated and demonstrated excellent performance at every stage, from cloud-based data visualization to sensor node slaves. Temperature (±0.1–0.7°C accuracy, 0.36–2.19% error), humidity (stable with expected diurnal variation), CO concentration (consistently safe 4–7 ppm range), and light intensity (peaking at noon and diminishing at night) were the four critical parameters that it was able to successfully capture, transmit, and log with full end-to-end reliability. The master node received the measurements from the slave devices and sent them to Firebase for cloud storage and mobile app representation. Importantly, CO levels refuted the existence of a clinically relevant environment in the environment under study, and temperature data demonstrated laboratory-level sensitivity (<2.2% error). The system's readiness for industrial applications is generally demonstrated by its adherence to reliable communication channels and time-stamped logging at all levels, particularly in Vietnam's coal and mineral industries where environmental monitoring is essential. Future iterations could use firmware-based calibration refinements to address some of the minor deviations that were noticed (humidity drift in afternoon hours).

5.2. FUTURE DEVELOPMENT DIRECTION

For future development, the system can be expanded to support additional types of sensors, such as toxic gas detectors and noise level monitors, tailored to the specific conditions of the mining industry. Integrating artificial intelligence will enable real-time data analysis, incident prediction, and operational optimization. Further improvements should also focus on

enhancing data security and managing user access more effectively in industrial environments. Finally, developing a more user-friendly interface for both web and mobile platforms will improve usability and accessibility for end users.

References

1. Nguyen, Q.N., Nguyen, V.H., Pham, T.P. and Chu, T.K.L., 2021. Current status of coal mining and some highlights in the 2030 development plan of coal industry in Vietnam. *Inżynieria Mineralna*.
2. Bui, Xuan-Nam & Lê, Quí. (2011). Coal Mining Industry in Vietnam.
3. Mahdevari, S., Shahriar, K. and Esfahanipour, A., 2014. Human health and safety risks management in underground coal mines using fuzzy TOPSIS. *Science of the Total Environment*, *488*, pp.85-99.
4. Zvarivadza, T., Onifade, M., Dayo-Olupona, O., Said, K.O., Githiria, J.M., Genc, B. and Celik, T., 2024. On the impact of Industrial Internet of Things (IIoT)-mining sector perspectives. *International Journal of Mining, Reclamation and Environment*, *38*(10), pp.771-809.
5. Chavan, S.V., Tilekar, S.K., Mane-Deshmukh, P.V. and Ladgaonkar, B.P., 2017. ARM Microcontroller Based Wireless Sensor Network to Monitor Environmental Parameters of Textile Industry. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(5).
6. Silva, J.D.C., Rodrigues, J.J., Alberti, A.M., Solic, P. and Aquino, A.L., 2017, August. LoRaWAN-A low power WAN protocol for Internet of Things: A review and opportunities. In *2nd International Multidisciplinary Conference on Computer and Energy Science, SpliTech 2017* (p. 8019271). Institute of Electrical and Electronics Engineers Inc.
7. Da Xu, L., He, W. and Li, S., 2014. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, *10*(4), pp.2233-2243.

8. Mousavi, S.M., Khademzadeh, A. and Rahmani, A.M., 2022. The role of low-power wide-area network technologies in Internet of Things: A systematic and comprehensive review. *International Journal of Communication Systems*, *35*(3), p.e5036.
9. Goh, J.Y. and Chua, D.W. (2024) 'A comparative analysis of wired vs. wireless data communication technologies', *International Journal of Electrical and Data Communication*, 5(2), pp. 8–12.
10. Nguyen, Q.K., 2024. How does financial flexibility strategy impact on risk management effectiveness?. *Sage Open*, *14*(2), p.21582440241240842.Brown, G., 2012. Discovering the STM32 microcontroller. *Cortex*, *3*(34), p.64.
11. Brown, G., 2012. Discovering the STM32 microcontroller. *Cortex*, *3*(34), p.64.
12. STM32F101xx, S. and STM32F103xx, S., 2010. STM32F107xx advanced ARM-based 32-bit MCUs. *Reference manual. STMicroelectronics*.
13. Abuagoub, A.M.A. (2016) 'An Overview of Industrial Wireless Sensor Networks', *International Journal of Computer Science and Information Technology Research*, 4(4), pp. 68–80.
14. LEE, Huang-Chen; KE, Kai-Hsiang. Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation. *IEEE Transactions on Instrumentation and Measurement*, 2018, 67.9: 2177-2187.
15. Sundaram, J.P.S., Du, W. and Zhao, Z., 2019. A survey on LoRa networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials*, *22*(1), pp.371-388.

16. Alkhayyal, M. and Mostafa, A., 2024. Recent developments in AI and ML for IoT: A systematic literature review on LoRaWAN energy efficiency and performance optimization. *Sensors*, *24*(14), p.4482.
17. Gao, R., Jiang, M. and Zhu, Z., 2023. Low-power wireless sensor design for LoRa-based distributed energy harvesting system. *Energy Reports*, *9*, pp.35-40.
18. Wu, D. and Liebeherr, J., 2023. A low-cost low-power LoRa mesh network for large-scale environmental sensing. *IEEE Internet of Things Journal*, *10*(19), pp.16700-16714.
19. Kufakunesu, R., Hancke, G.P. and Abu-Mahfouz, A.M., 2020. A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges. *Sensors*, *20*(18), p.5044.
20. Nguyen, T. T., & Nguyen, H. H. (2022). Design of noncoherent and coherent receivers for chirp spread spectrum systems. *IEEE Internet of Things Journal*, *9*(20), 19988-20002.
21. Bertoldo, S., Carosso, L., Marchetta, E., Paredes, M. and Allegretti, M., 2018. Feasibility analysis of a LoRa-based WSN using public transport. *Applied System Innovation*, *1*(4), p.49.
22. Mansingh, P.B., Jeevaprakash, R., Harish, G., Chandrasekar, R. and Dinesh, S., 2022. LoRa based smart irrigation system using WSN technology. *International Joumal of Research and Analytical Reviews*.
23. Wang, H., Zhang, X., Liao, J., Zhang, Y. and Li, H., 2024. An improved adaptive data rate algorithm of LoRaWAN for agricultural mobile sensor nodes. *Computers and Electronics in Agriculture*, *219*, p.108773.
24. Islam, R., Rahman, M.W., Rubaiat, R., Hasan, M.M., Reza, M.M. and Rahman, M.M., 2022. LoRa and server-based home automation using

the internet of things (IoT). *Journal of King Saud University-Computer and Information Sciences*, *34*(6), pp.3703-3712.

25. Ahsan, M., Based, M.A., Haider, J. and Rodrigues, E.M., 2021. Smart monitoring and controlling of appliances using LoRa based IoT system. *Designs*, *5*(1), p.17.
26. Sánchez-Sutil, F., Cano-Ortega, A. and Hernández, J.C., 2021. Design and implementation of a smart energy meter using a LoRa network in real time. *Electronics*, *10*(24), p.3152.
27. Waly, M., Smida, J., Bakouri, M., Alresheedi, B.A., Alqahtani, T.M., Alonzi, K.A. and Smida, A., 2024. Optimization of a Compact Wearable LoRa Patch Antenna for Vital Sign Monitoring in WBAN Medical Applications using Machine Learning. *IEEE Access*.
28. Leonardi, L., Bello, L.L. and Patti, G., 2022. LoRa support for long-range real-time inter-cluster communications over Bluetooth Low Energy industrial networks. *Computer Communications*, *192*, pp.57-65.
29. Fahmida, S., Modekurthy, V.P., Ismail, D., Jain, A. and Saifullah, A., 2022, May. Real-time communication over lora networks. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)* (pp. 14-27). IEEE.
30. Satyavathi, D.M., Mala, B.V., Vamsi, C.V., Chiranjeevi, C.C. and Neeraj, C.N., 2022. Real-Time Hidden Data Transmission Using Lora. *International Journal of Advanced Science Computing and Engineering*, *4*(2), pp.130-137.
31. De Almeida, I.B.F., Chafii, M., Nimr, A. and Fettweis, G., 2021. Alternative chirp spread spectrum techniques for LPWANs. *IEEE Transactions on Green Communications and Networking*, *5*(4), pp.1846-1855.

32. Maleki, A., Nguyen, H.H., Bedeer, E. and Barton, R., 2024. A Tutorial on Chirp Spread Spectrum Modulation for LoRaWAN: Basics and Key Advances. *IEEE Open Journal of the Communications Society*.
33. Li, C. and Cao, Z., 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. *ACM Computing Surveys (CSUR)*, *55*(3), pp.1-36.
34. HU, S.J. and PARK, H., Low-Power AES Data Encryption Architecture for a LoRaWAN.
35. Khan, I., 2019. Suitability of LoRa, SigFox and NB-IoT Different Internet-of-Things Applications.
36. Becoña, J.P., Grané, M., Miguez, M. and Arnaud, A., 2024. LoRa, Sigfox, and NB-IoT: An Empirical Comparison for IoT LPWAN Technologies in the Agribusiness. *IEEE Embedded Systems Letters*.
37. Coboi, Alberto E., V. Nguyen, Minh Nguyen, Nguyen DUY, and Thang TRAN. "An Analysis of ZigBee Technologies for Data Routing in Wireless Sensor Networks." *ICSES Trans. Comput. Netw. Commun.(ITCNC)* (2021).
38. Nguyen, Minh, et al. "Wireless Communication Technologies and Applications for Wireless Sensor Networks: A Survey." ICSES Transactions on Computer Networks and Communications 5.1 (2019): 1-15.
39. Zhu, G., Liao, C.H., Sakdejayont, T., Lai, I.W., Narusue, Y. and Morikawa, H., 2019. Improving the capacity of a mesh LoRa network by spreading-factor-based network clustering. *IEEE Access*, *7*, pp.21584-21596.
40. Jouhari, M., Saeed, N., Alouini, M.S. and Amhoud, E.M., 2023. A survey on scalable LoRaWAN for massive IoT: Recent advances,

potentials, and challenges. *IEEE Communications Surveys & Tutorials*, *25*(3), pp.1841-1876.

41. Centelles, R.P., Freitag, F., Meseguer, R. and Navarro, L., 2021. Beyond the star of stars: An introduction to multihop and mesh for LoRa and LoRaWAN. *IEEE Pervasive Computing*, *20*(2), pp.63-72.
42. Centelles, R.P., Meseguer, R., Freitag, F., Viñas, R.B. and Navarro, L., 2024. A minimalistic distance-vector routing protocol for LoRa mesh networks. *IEEE access*.
43. Callebaut, G. and Van der Perre, L., 2019. Characterization of LoRa point-to-point path loss: Measurement campaigns and modeling considering censored data. *IEEE Internet of Things Journal*, *7*(3), pp.1910-1918.
44. Almeida, N.C., Rolle, R.P., Godoy, E.P., Ferrari, P. and Sisinni, E., 2020, June. Proposal of a hybrid LoRa Mesh/LoRaWAN network. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT* (pp. 702-707). Ieee.
45. Mehic, M., Duliman, M., Selimovic, N. and Voznak, M., 2022. LoRaWAN end nodes: security and energy efficiency analysis. *Alexandria Engineering Journal*, *61*(11), pp.8997-9009.
46. Guitton, A. and Kaneko, M., 2022, December. Multi-gateway demodulation in LoRa. In *GLOBECOM 2022-2022 IEEE Global Communications Conference* (pp. 2008-2013). IEEE.
47. Zhou, Q., Zheng, K., Hou, L., Xing, J. and Xu, R., 2019. Design and implementation of open LoRa for IoT. *Ieee Access*, *7*, pp.100649-100657.
48. de Farias Medeiros, D., Villarim, M.R., de Carvalho, F.B.S. and de Souza, C.P., 2020, November. Implementation and analysis of routing

protocols for LoRa wireless mesh networks. In *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0020-0025). IEEE.

49. Debbarma, M.K., Debbarma, J., Sen, S.K. and Roy, S., DVR-BASED HYBRID ROUTING PROTOCOLS IN MOBILE AD-HOC NETWORK: APPLICATION AND CHALLENGES.
50. Kishore, C.N. and Kumar, H.V., 2023. Dynamic source routing protocol for robust path reliability and link sustainability aware routing in wireless communication. *Optik*, *282*, p.170036.
51. Shaukat, K. and Syrotiuk, V.R., 2010, August. Locally proactive routing protocols. In *International Conference on Ad-Hoc Networks and Wireless* (pp. 67-80). Berlin, Heidelberg: Springer Berlin Heidelberg.
52. Gupta, B.B. and Quamara, M., 2020. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience*, *32*(21), p.e4946.
53. Nikolic, M.V., Milovanovic, V., Vasiljevic, Z.Z. and Stamenkovic, Z., 2020. Semiconductor gas sensors: Materials, technology, design, and application. *Sensors*, *20*(22), p.6694.
54. Xiong, X., Cao, C. and Chander, G., 2010. An overview of sensor calibration inter-comparison and applications. *Frontiers of Earth Science in China*, *4*, pp.237-252.
55. Bychkovskiy, V., Megerian, S., Estrin, D. and Potkonjak, M., 2003, April. A collaborative approach to in-place sensor calibration. In *Information processing in sensor networks* (pp. 301-316). Berlin, Heidelberg: Springer Berlin Heidelberg.

56. El-Diasty, M. and Pagiatakis, S., 2010. Calibration and stochastic m odelling of i nertial na vigation s ensor errors. *Positioning (POS) Journal Information*, p.80.
57. Ren, B., Jia, T., Liu, H., Wang, Y. and Yan, J., 2024. Efficient estimation for sensor biases and target states in the presence of sensor position errors. *IEEE Sensors Journal*.
58. Delaine, F., Lebental, B. and Rivano, H., 2020. Framework for the simulation of sensor networks aimed at evaluating in situ calibration algorithms. *Sensors*, *20*(16), p.4577.
59. Barcelo-Ordinas, J.M., Doudou, M., Garcia-Vidal, J. and Badache, N., 2019. Self-calibration methods for uncontrolled environments in sensor networks: A reference survey. *Ad Hoc Networks*, *88*, pp.142-159.
60. Fu, J., Gao, Q. and Li, S., 2023. Application of intelligent medical sensing technology. *Biosensors*, *13*(8), p.812.
61. Nakamoto, T., Ikeda, T., Hirano, H. and Arimoto, T., 2009, October. Humidity compensation by neural network for bad-smell sensing system using gas detector tube and built-in camera. In *SENSORS, 2009 IEEE* (pp. 281-286). IEEE.
62. Zuppa, M., Distante, C., Persaud, K.C. and Siciliano, P., 2007. Recovery of drifting sensor responses by means of DWT analysis. *Sensors and Actuators B: Chemical*, *120*(2), pp.411-416.
63. Gardner, J.W. and Bartlett, P.N., 1992. Pattern recognition in odour sensing. In *Sensors and sensory systems for an electronic nose* (pp. 161-179). Dordrecht: Springer Netherlands.
64. Al Fallah, S., Arioua, M., El Oualkadi, A. and El Asri, J., 2018, April. On the performance of piecewise linear approximation techniques in

WSNs. In *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)* (pp. 1-6). IEEE.

Appendices

Code for the Master (Gateway)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  /// scl b6, sda b7
#define bom PA0
String A="A";
String B="B";
String C="C";
String D="D";
String E="E";
String inputString="";
bool stringComplete=false;
float vtemp,vhumi,vam;
void setup() {
  lcd.begin();
  lcd.backlight();
   Serial.begin(9600);
//pinMode(PC13,OUTPUT);
//digitalWrite(PC13,0);
delay(1000);
}

void loop() {
nhandata_lora();

//delay(200);
}

void nhandata_lora(){
   if(Serial.available()>0) {

   char inChar = (char)Serial.read();
   inputString+=inChar;
   if(inChar=='#')
   {
    stringComplete=true;
   }
   if(stringComplete)
```

```
    {
//     Serial.println(inputString) ;

     int TimA,TimB,TimC,TimD,TimE;
      TimA=inputString.indexOf("A");
      TimB=inputString.indexOf("B");
      TimC=inputString.indexOf("C");
      TimD=inputString.indexOf("D");
      TimE=inputString.indexOf("E");

if(TimA>=0&&TimB>=0&&TimC>=0&&TimD>=0&&TimE>=0)
      {
       String ChuoiA,ChuoiB,ChuoiC,ChuoiD="";

      ChuoiA=inputString.substring(TimA+1,TimB);
      ChuoiB=inputString.substring(TimB+1,TimC);
      ChuoiC=inputString.substring(TimC+1,TimD);
      ChuoiD=inputString.substring(TimD+1,TimE);

      vtemp=ChuoiB.toInt();
      vam=ChuoiC.toInt();

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("CO:");
      lcd.print(vam);

      lcd.setCursor(0, 1);
      lcd.print("Temp:");
      lcd.print(vtemp);
//      lcd.setCursor(0, 1);
//      lcd.print("CO:");

//      lcd.print(ChuoiC);
//      lcd.setCursor(8, 1);
//      lcd.print("L:");
//
//      lcd.print(ChuoiD);

     }
```

```
      inputString="";
     stringComplete=false;
  }
    }

}
```

Code for the slave node

```
////////////////////////////////////////// dht11
#include <DHT.h>
#define DHTPIN PB0
DHT dht(DHTPIN, DHT11);
float h;
float t;

#define co PA0       //
#define LIGH PA1
int dat_value;
int as_value;

String str="";
String A="A";
String B="B";
String C="C";
String D="D";
String E="E";
int conut = 0;
void setup() {
  dht.begin();
  Serial.begin(9600);
   pinMode(co,INPUT);
   pinMode(LIGH,INPUT);
   pinMode(PC13,OUTPUT);
  delay(1000);
}

void loop() {
  ////////////////////////////// dht11
    h = dht.readHumidity();
    t = dht.readTemperature();
    dat_value=analogRead(co);
    as_value=analogRead(LIGH);
  str=A + h + B + t + C + dat_value + D + as_value + E +'#';
  Serial.print(str);
  digitalWrite(PC13,1);
  delay(200);
digitalWrite(PC13,0);
```

```
delay(200);
}
```

Code for the ESP8266 Wi-Fi

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial serial_ESP(D2,D3);//D2 = RX -- D3 = TX
//SoftwareSerial serial_ESP1(D5,D6);//D2 = RX -- D3 = TX
const char *ssid = "17B6"; // replace with your wifi ssid and wpa2 key
const char *pass = "khongbietduoc";
//// cài đặt fribase
#include<FirebaseESP8266.h>
FirebaseData firebaseData;
FirebaseJson json;
#define FIREBASE_HOST "https://iotdatabase-8c823-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "vm4fyPMPa5tM0AxCC2gJnAvEaCEXDMx2HFauqnwy"
String dulieu,dulieu1;
String str,str1 ="";


String inputString="";
bool stringComplete=false;
String ChuoiGui="";
String A="A";
String B="B";
String C="C";
String D="D";
String E="E";

int vtemp,vhumi,vco,vligh;


void setup() {
  pinMode(D2,INPUT);
  pinMode(D3,OUTPUT);
  pinMode(D4,OUTPUT);
  serial_ESP.begin(9600);

  Serial.begin(9600);

Serial.println("Connecting to ");
```

```
        Serial.println(ssid);
        WiFi.begin(ssid, pass);
        while (WiFi.status() != WL_CONNECTED)
        {
        delay(500);
        Serial.print(".");
        }
        Serial.println("");
        Serial.println("WiFi connected");
          Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
        }

        void loop() {
        while(serial_ESP.available())
           {
           char inChar = (char)serial_ESP.read(); ////////// chuỗi nhận có dạng A
75 B 30 C 2,5 D 150 E P1 F #
           Serial.print(inChar);
           inputString += inChar;

           if(inChar=='#')
           {

            stringComplete=true;
           }

           if(stringComplete)
           {
             int TimA,TimB,TimC,TimD,TimE;
              TimA=inputString.indexOf("A");
              TimB=inputString.indexOf("B");
              TimC=inputString.indexOf("C");
              TimD=inputString.indexOf("D");
              TimE=inputString.indexOf("E");

if(TimA>=0&&TimB>=0&&TimC>=0&&TimD>=0&&TimE>=0)
              {
                String ChuoiA,ChuoiB,ChuoiC,ChuoiD="";

               ChuoiA=inputString.substring(TimA+1,TimB);
```

```
      ChuoiB=inputString.substring(TimB+1,TimC);
      ChuoiC=inputString.substring(TimC+1,TimD);
      ChuoiD=inputString.substring(TimD+1,TimE);
//          if(Firebase.getString(firebaseData,"/baochay/control"))  //
//          {
//           dulieu = firebaseData.stringData();
//           dulieu.remove(0,2);
//           dulieu.remove(dulieu.length() - 2, 2);
//           if(dulieu == "A2"){
//             if(vam >600){
//               digitalWrite(D4, 0);
//             }
//             else if(vam <600){
//               digitalWrite(D4, 1);
//             }
//           }
//           if(dulieu == "A0"){
//             digitalWrite(D4, 0);
//           }
//           if(dulieu == "A1"){
//             digitalWrite(D4, 1);
//           }
//
//          }

        vtemp =ChuoiB.toFloat();
      vhumi =ChuoiA.toFloat();
      vco = ChuoiC.toFloat();
       vligh = ChuoiD.toFloat();
      firebase1();
     }
      inputString="";
    stringComplete=false;
  }
  }
}

void firebase1(){
 Firebase.setDouble(firebaseData, "/giamsat/t", vtemp);///// nhiệt độ
 Firebase.setDouble(firebaseData, "/giamsat/h", vhumi);///// ẩm
```

```
  Firebase.setDouble(firebaseData, "/giamsat/c", vco);///// CO
  Firebase.setDouble(firebaseData, "/giamsat/l", vligh);///// As
}
```