

Smart Industrial Equipment Monitoring System Using Lora-Based Long Range Wireless Communication

VIGNESH V K

714522106048

GOPIANANTH T

714522106011

HARIHARAN M

714522106013

AGASING RAJ P

714522106001

ELECTRONICS AND COMMUNICATION ENGINEERING

UNITED INSTITUTE OF TECHNOLOGY

MAY 2026

ABSTRACT

This project presents a Smart Industrial Equipment Monitoring System using LoRa-based wireless communication for real-time monitoring and fault detection. The system is designed to monitor key machine parameters such as temperature, vibration, and current to ensure efficient and safe operation of industrial equipment. The transmitter unit, built using an Arduino Nano, collects data from a DS18B20 temperature sensor, a vibration sensor, and an ACS712 current sensor. The acquired data is processed and transmitted over long distances using a LoRa module.

At the receiver side, an ESP8266 NodeMCU receives the data through LoRa communication and displays it on an I2C LCD. The system also uploads the data to the Ubidots cloud platform using the MQTT protocol for remote monitoring and analysis. Fault conditions such as overload, vibration fault, and power failure are detected based on sensor readings. LED strips controlled through a relay are used to simulate industrial load conditions.

The proposed system provides a cost-effective, low-power, and longrange solution for industrial monitoring. It supports predictive maintenance, reduces downtime, and enhances system reliability, making it suitable for Industry 4.0 applications.

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

In modern industrial environments, machinery plays a vital role in ensuring continuous production and operational efficiency. Industries rely heavily on equipment such as motors, pumps, compressors, and conveyor systems, which operate for extended durations under varying load conditions. Any unexpected failure in such equipment can lead to production downtime, financial losses, and safety risks. Therefore, continuous monitoring of machine health parameters has become essential in industrial applications.

Traditionally, industries have depended on manual inspection and periodic maintenance to assess machine conditions. However, these methods are time-consuming, less reliable, and incapable of detecting real-time faults. With the rapid advancement of automation and Industry 4.0 technologies, there is a growing need for intelligent monitoring systems that can continuously track machine parameters and provide timely alerts in case of abnormal conditions.

Wireless communication technologies have significantly improved the way industrial monitoring systems operate. Among these technologies, LoRa (Long Range) communication has gained considerable attention due to its ability to

transmit data over long distances with low power consumption. LoRa is particularly suitable for industrial environments where machines are distributed over large areas, and wired connections are not feasible or cost-effective.

This project focuses on the design and implementation of a Smart Industrial Equipment Monitoring System using LoRa-based communication.

The system is capable of monitoring key machine parameters such as temperature, vibration, and electrical current. These parameters are critical indicators of machine health and help in detecting faults such as overheating, mechanical imbalance, and overload conditions.

The transmitter unit of the system is built using an Arduino Nano microcontroller interfaced with sensors including the DS18B20 temperature sensor, vibration sensor, and ACS712 current sensor. The collected data is transmitted wirelessly using a LoRa module. On the receiver side, an ESP8266 NodeMCU receives the data, displays it on an I2C LCD, and sends it to the Ubidots cloud platform using MQTT protocol for remote monitoring.

To simulate real industrial conditions, LED strips are used as electrical loads, and a relay is employed to control the load. The system detects different fault conditions such as power failure, overload, and vibration abnormalities based on sensor readings.

Overall, the proposed system provides a reliable, low-cost, and efficient solution for industrial equipment monitoring. It supports real-time monitoring, reduces maintenance costs, and enhances operational safety, making it suitable for modern industrial applications.

1.2 PROBLEM STATEMENT

Industrial systems often suffer from unexpected failures due to lack of continuous monitoring and delayed fault detection. Traditional monitoring techniques rely on manual inspection and wired systems, which are inefficient and costly. There is a need for a wireless, real-time monitoring system that can detect faults early and improve system reliability.

The key problems addressed in this project are:

- Lack of real-time monitoring of industrial equipment
- High cost and complexity of wired monitoring systems
- Difficulty in monitoring machines located in remote or large industrial areas
- Delayed detection of faults such as overload, vibration, and overheating
- Absence of remote monitoring and cloud-based data analysis

This project aims to overcome these challenges by developing a LoRa-based wireless monitoring system that provides real-time data transmission and fault detection.

1.3 OBJECTIVES OF THE STUDY

The main objective of this project is to design and implement a smart industrial monitoring system using LoRa communication. The specific objectives are:

- To monitor temperature, vibration, and current parameters of industrial equipment
- To develop a wireless communication system using LoRa for longrange data transmission
- To detect fault conditions such as overload, vibration fault, and power failure
- To display real-time data on an LCD for local monitoring
- To upload sensor data to the Ubidots cloud platform using MQTT protocol
- To create a low-cost and efficient system suitable for industrial applications
- To improve machine reliability and reduce maintenance costs

1.4 SCOPE OF THE PROJECT

The scope of this project focuses on developing a prototype system for monitoring industrial equipment using wireless communication. The system is designed to monitor essential parameters such as temperature, vibration, and current.

The project includes the following scope:

- Design of transmitter unit using Arduino Nano and sensors
- Implementation of LoRa communication between transmitter and receiver
- Development of receiver unit using ESP8266
- Display of real-time data using I2C LCD
- Cloud integration using Ubidots for remote monitoring
- Simulation of industrial load using LED strips and relay
- Detection of basic fault conditions such as overload and vibration However, the project has certain limitations:
 - The system is implemented as a prototype and not tested in largescale industrial environments
 - Only basic fault detection is implemented
 - Advanced predictive maintenance techniques using AI are not included
 - The system monitors limited parameters

Despite these limitations, the project provides a strong foundation for future development and real-world implementation.

1.5 SIGNIFICANCE OF THE STUDY

The significance of this project lies in its ability to improve industrial monitoring and maintenance practices. By integrating LoRa communication with IoT technology, the system provides a modern solution for real-time monitoring and fault detection.

Key benefits of this study include:

Enables real-time monitoring of machine health parameters Reduces downtime by detecting faults early Eliminates the need for complex wiring systems Provides long-range communication suitable for large industrial areas Supports remote monitoring through cloud platforms Reduces maintenance costs and improves operational efficiency Enhances safety by identifying abnormal conditions

This project also contributes to the advancement of Industry 4.0 by integrating wireless communication and IoT-based monitoring. It demonstrates how low-cost hardware and cloud platforms can be used to develop efficient industrial monitoring systems.

The proposed system can be further extended to include advanced features such as predictive analytics, mobile notifications, and multi-machine monitoring, making it highly relevant for future industrial applications.

CHAPTER 2 LITERATURE SURVEY

2.1 INTRODUCTION TO LITERATURE SURVEY

Industrial equipment monitoring has become a crucial aspect of modern manufacturing systems due to the increasing demand for efficiency, reliability, and safety. With the evolution of Industry 4.0, there has been a significant shift from traditional monitoring methods to smart, automated, and wireless monitoring systems. The literature survey focuses on existing research works and technologies related to industrial monitoring, wireless communication, and IoT-based systems.

Several researchers have proposed different methods to monitor industrial parameters such as temperature, vibration, current, and pressure. The integration of sensors, microcontrollers, and wireless communication technologies has enabled the development of intelligent monitoring systems. Among the various communication technologies available, LoRa (Long Range) communication has gained attention due to its long-range capability and low power consumption.

This chapter reviews relevant research works related to LoRa-based monitoring systems, IoT integration, and sensor-based fault detection techniques.

2.1.1 TRADITIONAL INDUSTRIAL MONITORING SYSTEMS

Traditional industrial monitoring systems are mainly based on wired communication and manual inspection. In such systems, sensors are connected directly to control units using cables. Although these systems provide accurate data, they suffer from several limitations: High installation and maintenance cost
Difficulty in expanding the system
Limited flexibility in large industrial areas
Increased complexity due to wiring

Manual inspection methods are still used in many industries, where technicians periodically check machine conditions. However, this approach is inefficient and cannot provide real-time fault detection. As a result, unexpected failures may occur, leading to downtime and financial losses.

These limitations have motivated researchers to develop wireless monitoring systems that can overcome the drawbacks of traditional methods.

2.1.2 WIRELESS COMMUNICATION TECHNOLOGIES IN INDUSTRIAL MONITORING

Various wireless communication technologies have been used in industrial monitoring systems, including Wi-Fi, Bluetooth, Zigbee, and GSM. Each technology has its advantages and limitations.

Wi-Fi provides high data rates but consumes more power and has limited coverage. Bluetooth is suitable for short-range communication but cannot be used in large industrial environments. Zigbee offers low power consumption but has a limited range and network complexity.

LoRa (Long Range) communication has emerged as a suitable solution for industrial monitoring due to its unique features:

- Long-range communication (up to several kilometers)
- Low power consumption
- High reliability in harsh environments
- Ability to support multiple devices

Researchers have demonstrated that LoRa technology is highly effective for IoT-based industrial applications, especially in scenarios where devices are distributed over large areas.

2.1.3 LORA-BASED INDUSTRIAL MONITORING SYSTEMS

Recent studies have focused on implementing LoRa-based monitoring systems for industrial applications.

In these systems, sensor nodes collect data and transmit it to a central receiver using LoRa communication.

A LoRa-based industrial monitoring system was developed to monitor parameters such as temperature and humidity.

The system successfully transmitted data over long distances with minimal power consumption. It also provided real-time monitoring through cloud platforms.

Another research work proposed a LoRa-based system for machine health monitoring. The system used sensors to detect vibration and temperature changes and transmit the data to a remote monitoring station.

The results showed that LoRa communication is reliable and suitable for industrial environments.

These studies highlight the effectiveness of LoRa in providing longrange, low-power communication for industrial monitoring applications.

2.1.4 SENSOR-BASED FAULT DETECTION TECHNIQUES

Sensors play a critical role in detecting faults in industrial equipment.

Different types of sensors are used to monitor various parameters:

Temperature Sensors

Temperature sensors such as DS18B20 are widely used to detect overheating in machines. Overheating can indicate problems such as friction, overload, or electrical faults.

Vibration Sensors

Vibration sensors are used to detect mechanical issues such as imbalance, misalignment, and bearing failures. Abnormal vibration patterns often indicate potential machine faults.

Current Sensors

Current sensors such as ACS712 are used to monitor electrical load conditions. By measuring current, it is possible to detect overload, underload, and power failure conditions.

Researchers have shown that combining multiple sensors provides better fault detection accuracy compared to using a single sensor.

2.1.5 IoT AND CLOUD-BASED MONITORING SYSTEMS

The integration of Internet of Things (IoT) technology has significantly improved industrial monitoring systems. IoT enables devices to collect, process, and transmit data to cloud platforms for remote monitoring.

Cloud platforms such as Ubidots provide features such as:

- Real-time data visualization
- Data storage and analysis
- Remote monitoring through dashboards
- Alert generation

Several studies have implemented IoT-based monitoring systems where sensor data is uploaded to the cloud using

protocols such as MQTT and HTTP. These systems allow users to monitor machine conditions from anywhere, improving efficiency and decision-making.

2.2 LIMITATIONS OF EXISTING SYSTEMS

Although significant progress has been made in industrial monitoring systems, existing systems still have certain limitations: High implementation cost in some systems Limited scalability in traditional wired systems Short communication range in technologies like Bluetooth and Zigbee Lack of integration between wireless communication and cloud monitoring Limited fault detection capabilities. These limitations indicate the need for a system that combines long-range communication, low power consumption, and cloud integration.

2.3 SUMMARY

From the literature survey, it is evident that wireless communication and IoT technologies play a vital role in modern industrial monitoring systems. LoRa communication provides a reliable solution for long-range data transmission, while IoT platforms enable remote monitoring and data analysis.

The proposed system builds upon these concepts by integrating LoRa communication with sensor-based monitoring and cloud connectivity. It aims to provide a cost-effective, scalable, and efficient solution for industrial equipment monitoring.

CHAPTER 3 SYSTEM DESIGN

3.1 EXISTING SYSTEM

In traditional industrial environments, equipment monitoring is typically performed using wired systems or manual inspection methods. Sensors are directly connected to control panels using cables, and data is monitored locally without remote access. In many cases, technicians periodically inspect machines to identify issues such as overheating, abnormal vibration, or electrical faults.

Some industries also use basic automation systems where sensors are connected to Programmable Logic Controllers (PLCs). These systems provide limited monitoring capabilities and are usually confined to a specific location. Wireless technologies such as Bluetooth and Zigbee have also been used, but they are limited in range and scalability.

Existing systems mainly focus on individual parameter monitoring rather than a comprehensive multi-parameter approach. Additionally, most systems do not support cloud integration, making it difficult to monitor equipment remotely.

3.2 DRAWBACKS OF EXISTING SYSTEM

Despite their usage, traditional monitoring systems have several limitations:

- **High Installation Cost:** Wired systems require extensive cabling, increasing cost and complexity.
- **Limited Scalability:** Adding new machines or sensors requires additional wiring and modifications.
- **Lack of Real-Time Monitoring:** Manual inspection cannot detect faults instantly, leading to delayed response.
- **Short Communication Range:** Wireless technologies like Bluetooth and Zigbee are not suitable for large industrial areas.
- **No Remote Access:** Data is usually available only at the local control unit.
- **Increased Maintenance Effort:** Regular inspection requires manpower and time.

- **Limited Fault Detection:** Many systems monitor only one parameter, reducing accuracy in fault detection.

These drawbacks highlight the need for a modern monitoring system that is wireless, scalable, and capable of real-time data analysis.

3.3 PROPOSED SYSTEM

The proposed system is a Smart Industrial Equipment Monitoring System based on LoRa wireless communication. It is designed to monitor critical machine parameters such as temperature, vibration, and current in real time.

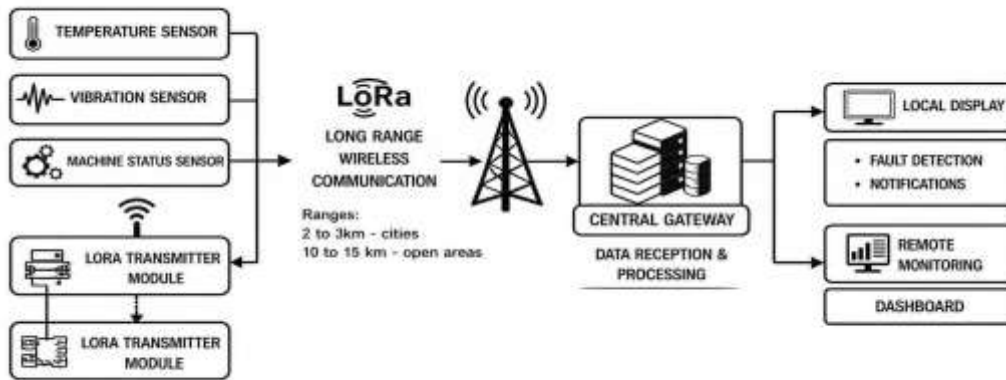


Fig 1 : Block Diagram of the Proposed System

Transmitter Unit

The transmitter unit is built using Arduino Nano and connected sensors:

- DS18B20 temperature sensor
- Vibration sensor
- ACS712 current sensor

These sensors continuously collect machine data. The Arduino processes the data and transmits it using a LoRa module.

Receiver Unit

The receiver unit is based on ESP8266 NodeMCU. It receives data via LoRa communication and performs the following functions:

- Displays data on an I2C LCD
- Sends data to Ubidots cloud using MQTT

Relay and LED strips are used to simulate industrial load conditions. The current flowing through the load is measured using ACS712 to detect overload conditions.

3.3.1 WORKING PRINCIPLE

The working of the proposed system is based on continuous monitoring, data processing, and wireless transmission.

At the transmitter side, sensors collect data from the equipment. The DS18B20 measures temperature, the vibration sensor detects abnormal motion, and the ACS712 measures current flow through the load. The Arduino Nano reads these values and processes them.

The system uses predefined threshold values to determine the condition of the equipment. If the current value is within a normal range, the system indicates normal operation. When additional load (LED strips) is connected, the current increases, and the system detects an overload condition. If no load is present, the current value indicates a power failure. If the vibration sensor is triggered, the system detects a vibration fault.

The processed data is transmitted wirelessly using the LoRa module. LoRa enables long-range communication with low power consumption, making it suitable for industrial applications.

At the receiver side, the ESP8266 receives the data through LoRa. The data is displayed on an I2C LCD for real-time monitoring. Additionally, the ESP8266 connects to Wi-Fi and sends the data to the Ubidots cloud using MQTT protocol. This allows remote monitoring of equipment parameters through a web dashboard.

The system operates continuously, providing real-time monitoring and fault detection.

3.3.2 FEATURES OF THE PROPOSED SYSTEM

The proposed system offers several advantages and features:

- **Wireless Communication:** Uses LoRa technology for long-range data transmission
- **Real-Time Monitoring:** Continuously monitors temperature, vibration, and current
- **Fault Detection:** Detects overload, power failure, and vibration faults
- **Cloud Integration:** Sends data to Ubidots for remote monitoring
- **Local Display:** Displays real-time data on I2C LCD
- **Low Power Consumption:** Efficient operation using low-power components
- **Scalable System:** Easy to expand for multiple machines
- **Cost-Effective:** Uses low-cost components like Arduino and ESP8266
- **Reduced Wiring:** Eliminates complex wired connections
- **User-Friendly:** Simple interface for monitoring and analysis

3.4 SYSTEM ARCHITECTURE

The system architecture of the proposed industrial monitoring system is divided into two main sections: the transmitter unit and the receiver unit. The architecture is designed to ensure efficient data acquisition, processing, wireless transmission, and remote monitoring.

The transmitter unit is responsible for collecting data from sensors and transmitting it using LoRa communication. The receiver unit processes the received data, displays it locally, and uploads it to the cloud for remote monitoring.

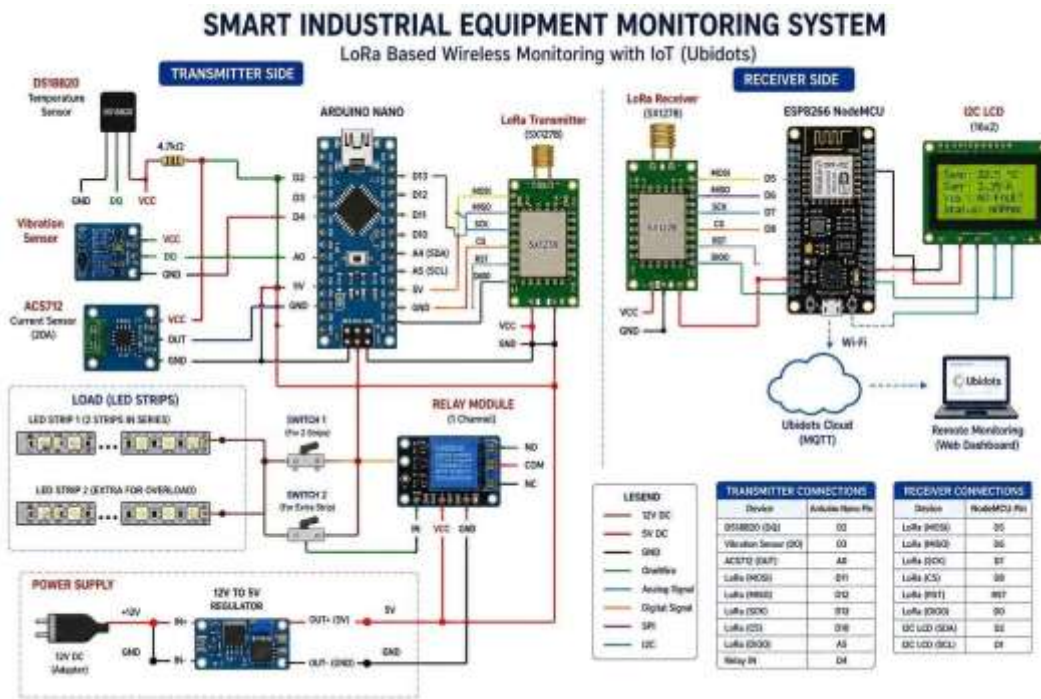


Fig 2 : System Architecture Diagram

3.4.1 TRANSMITTER UNIT

The transmitter unit is built using an Arduino Nano microcontroller and multiple sensors to monitor industrial parameters.

Components Used:

- Arduino Nano
- DS18B20 Temperature Sensor
- Vibration Sensor
- ACS712 Current Sensor
- LoRa SX1278 Module
- Relay Module
- LED Strip Load

Working of Transmitter Unit:

The DS18B20 sensor measures the temperature of the equipment continuously and sends digital data to the Arduino Nano. The vibration sensor detects abnormal vibrations and provides a digital signal indicating machine instability. The ACS712 current sensor measures the current flowing through the load (LED strips), which helps in identifying load variations.

The Arduino Nano collects all sensor data and processes it. Based on predefined threshold values, it determines the machine condition such as normal operation, overload, power failure, or vibration fault.

The processed data is then transmitted using the LoRa module. The LoRa transmitter converts the data into radio signals and sends it wirelessly over a long distance.

The relay module is used to control the LED strip load, which simulates industrial equipment load. When additional LED strips are connected, the current increases and helps in demonstrating overload conditions.

3.4.2 RECEIVER UNIT

The receiver unit is responsible for receiving data from the transmitter and displaying and uploading it for monitoring.

Components Used:

- ESP8266 NodeMCU
- LoRa SX1278 Module
- I2C 16x2 LCD Display
- Wi-Fi Connectivity
- Ubidots Cloud Platform

Working of Receiver Unit:

The LoRa receiver module receives the transmitted data from the transmitter unit. The ESP8266 NodeMCU processes the received data and extracts parameters such as temperature, current, vibration, and fault status.

The processed data is displayed on the I2C LCD in real time, allowing local monitoring of machine conditions.

The ESP8266 also connects to a Wi-Fi network and sends the data to the Ubidots cloud platform using MQTT protocol. This enables remote monitoring through dashboards and provides data visualization.

The receiver continuously updates the display and cloud data, ensuring real-time monitoring of industrial equipment.

The LoRa receiver module receives the transmitted data from the transmitter unit. The ESP8266 NodeMCU processes the received data and extracts parameters such as temperature, current, vibration, and fault status.

The processed data is displayed on the I2C LCD in real time, allowing local monitoring of machine conditions.

The ESP8266 also connects to a Wi-Fi network and sends the data to the Ubidots cloud platform using MQTT protocol. This enables remote monitoring through dashboards and provides data visualization. The receiver continuously updates the display and cloud data, ensuring real-time monitoring of industrial equipment.

3.5 ADVANTAGES OF THE PROPOSED SYSTEM

- **Long-Range Communication:** LoRa enables communication over large distances, suitable for industrial environments
- **Low Power Consumption:** Efficient operation with minimal energy usage
- **Real-Time Monitoring:** Continuous tracking of machine parameters
- **Wireless System:** Eliminates complex wiring and reduces installation cost

- **Fault Detection:** Identifies overload, vibration fault, and power failure
- **Cloud Integration:** Enables remote monitoring through Ubidots
- **Scalability:** Can be expanded to monitor multiple machines
- **Cost-Effective:** Uses low-cost components like Arduino and ESP8266
- **User-Friendly Interface:** LCD display and cloud dashboard for easy monitoring
- **Improved Reliability:** Early fault detection reduces downtime

3.6 COMPARISON WITH THE EXISTING SYSTEM

Feature	Existing System	Proposed System
Communication	Wired / Short-range wireless	Long-range LoRa
Monitoring	Manual / Limited	Real-time monitoring
Installation Cost	High	Low
Scalability	Difficult	Easy
Fault Detection	Limited	Multiple fault detection
Remote Monitoring	Not available	Available via cloud
Power Consumption	High	Low
Flexibility	Low	High
Data Access	Local only	Local + Cloud

CHAPTER 4 SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

The proposed system requires a set of hardware components to monitor industrial equipment parameters and perform wireless communication. The hardware components are selected based on reliability, cost-effectiveness, and compatibility.

List of Hardware Components:

- Arduino Nano
- ESP8266 NodeMCU
- LoRa SX1278 Module (Transmitter & Receiver)

- DS18B20 Temperature Sensor
- Vibration Sensor (SW420)
- ACS712 Current Sensor (5A)
- Relay Module
- LED Strip (Load – 3 units)
- I2C 16x2 LCD Display
- Power Supply (5V & 12V)
- Connecting wires and breadboard

These components work together to measure parameters, process data, transmit wirelessly, and display results.

4.2 SOFTWARE REQUIREMENTS

The system requires software tools for programming, data transmission, and cloud integration.

Software Tools:

- Arduino IDE (for coding and uploading programs)
- ESP8266 Board Package (for NodeMCU programming)
- Ubidots Cloud Platform (for data visualization)

Libraries Used:

- SPI Library (for LoRa communication)
- LoRa Library (SX1278 communication)
- OneWire Library (for DS18B20 sensor)
- DallasTemperature Library (temperature readings)
- Ubidots MQTT Library (cloud communication)
- Wire Library (for I2C LCD)
- LiquidCrystal_I2C Library (LCD display control)

These tools and libraries enable efficient system development and operation.

4.3 SYSTEM DESCRIPTION

The system is designed to monitor industrial equipment parameters and detect faults in real time. It consists of two main sections: a transmitter and a receiver.

The transmitter unit collects data from sensors such as temperature, vibration, and current. The Arduino Nano processes this data and sends it using LoRa communication.

The receiver unit receives the data using LoRa and processes it using ESP8266. The data is displayed locally on an LCD and also uploaded to the Ubidots cloud platform using MQTT protocol.

The system detects:

- Normal operation
- Overload condition
- Power failure
- Vibration fault

The LED strips act as a load to simulate industrial equipment, and the relay controls the load operation.

4.3.1 HARDWARE SPECIFICATION

Arduino Nano

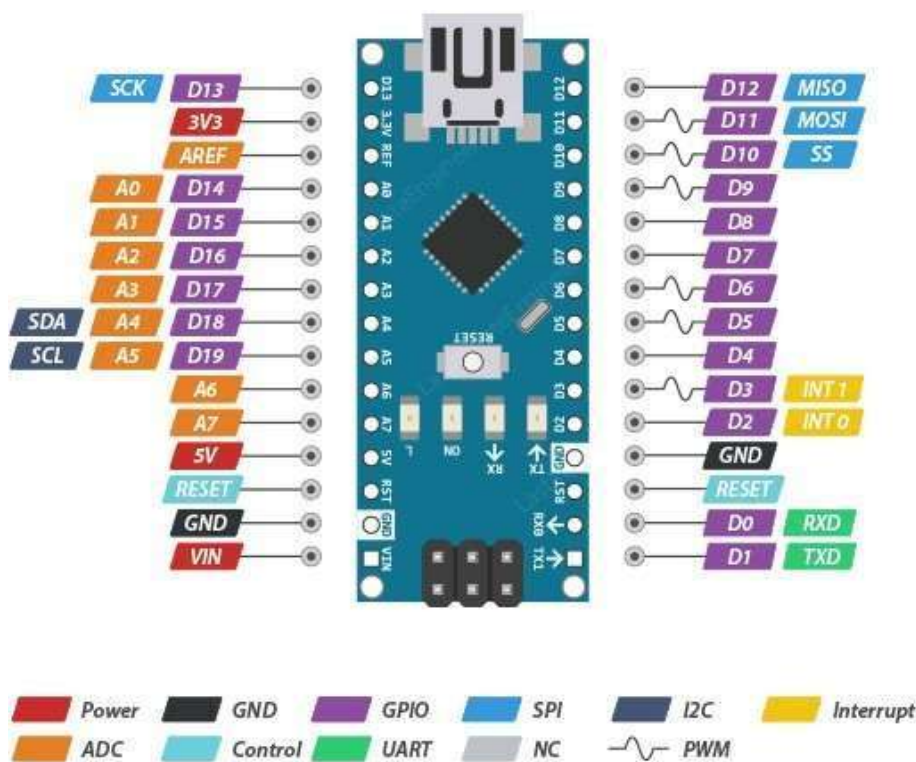


Fig 3 : Arduino Nano

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Digital I/O Pins: 14
- Analog Input Pins: 8
- Clock Speed: 16 MHz

ESP8266 NodeMCU

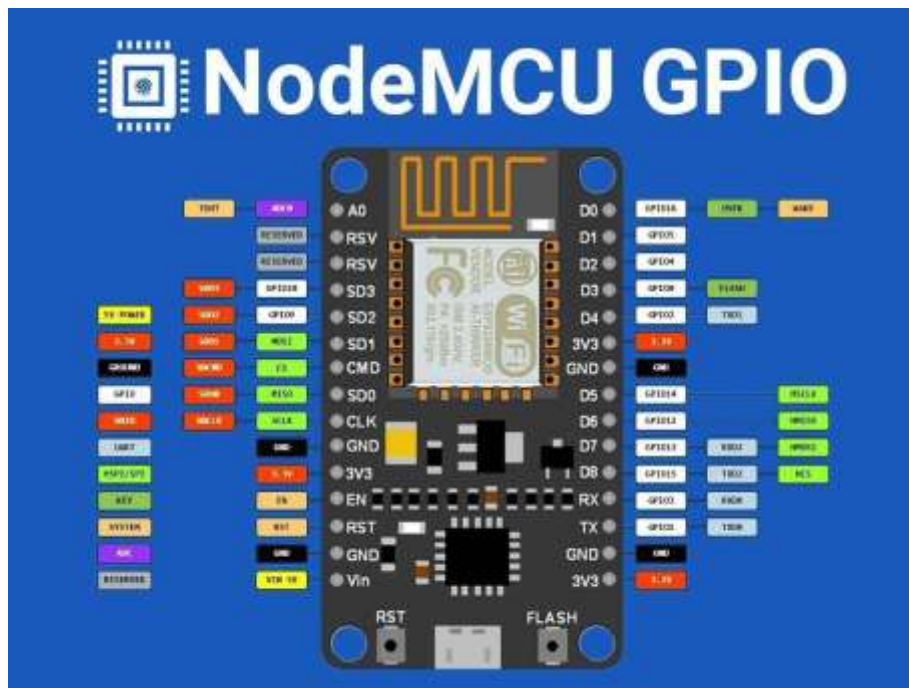


Fig 4 : ESP8266 NodeMCU

- Operating Voltage: 3.3V
- Wi-Fi Enabled
- GPIO Pins: 11
- Flash Memory: 4MB

LoRa SX1278 Module

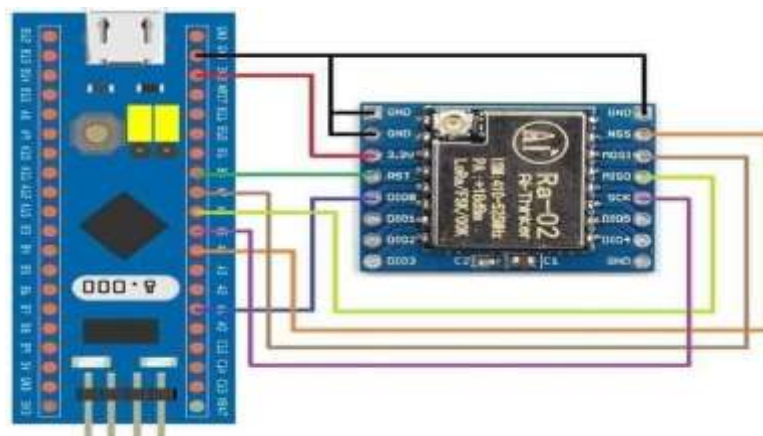


Fig 5 : LoRa SX1278 Module

- Frequency: 433 MHz
- Communication: SPI
- Range: Up to several kilometers

- Low power consumption

DS18B20 Temperature Sensor



Fig 6 : DS18B20 Temperature Sensor

- Digital output
- Temperature range: -55°C to $+125^{\circ}\text{C}$
- High accuracy

Vibration Sensor (SW420)

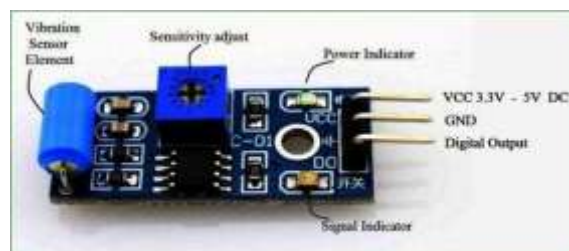


Fig 7 : Vibration Sensor (SW420)

- Digital output
- Detects vibration and shock

ACS712 Current Sensor

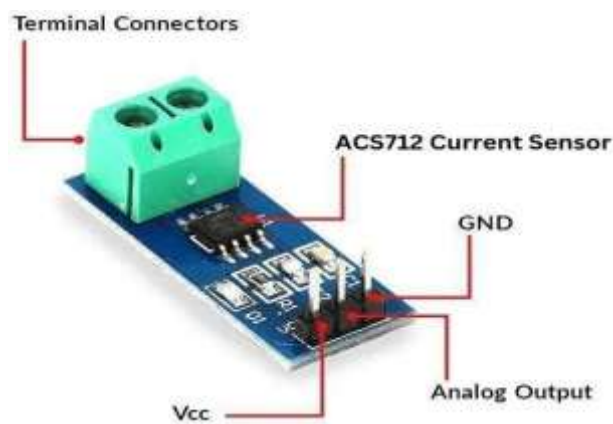


Fig 8 : ACS712 Current Sensor

- Measures AC/DC current
- Analog output
- 5A range

Relay Module



Fig 9 : Relay Module

- Used to control load
- Operates at 5V

LED Strip

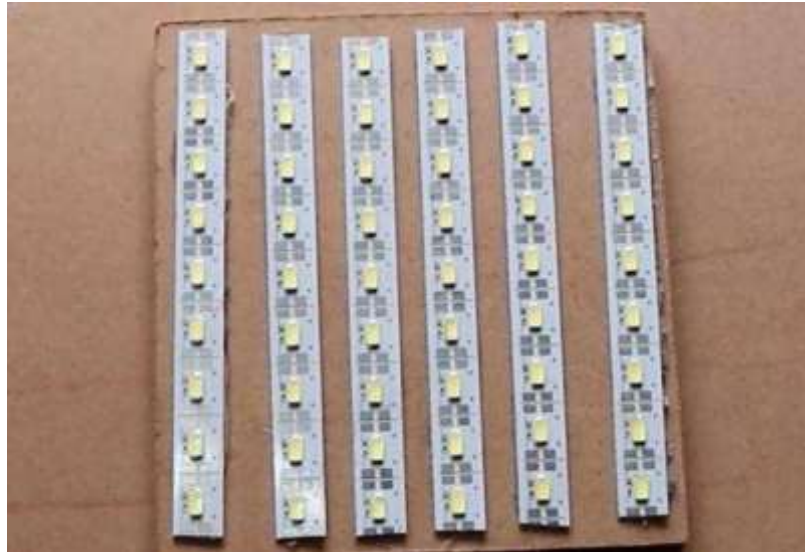


Fig 10 : LED Strip

- Used as load for testing
- Operates at 12V

I2C LCD Display



Fig 11 : I2C LCD Display

- 16x2 display
- Uses SDA and SCL pins

4.3.2 SOFTWARE SPECIFICATION

Arduino IDE

The Arduino IDE is used to write, compile, and upload code to both Arduino Nano and ESP8266. It provides an easy interface for programming and debugging.

Embedded C Programming

The system is programmed using Embedded C language. It includes sensor reading, data processing, fault detection, and communication logic.

LoRa Communication Protocol

The LoRa library is used to enable wireless communication between transmitter and receiver. It handles data transmission and reception.

MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is used for sending data to the Ubidots cloud. It is lightweight and suitable for IoT applications.

Ubidots Cloud Platform

Data visualization, Remote monitoring, Dashboard creation, Real-time updates.

LCD Display Programming

The LiquidCrystal_I2C library is used to display sensor data and system status on the LCD.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 SYSTEM IMPLEMENTATION PROCESS

The implementation of the proposed system involves the integration of hardware components, development of embedded software, and testing of communication between transmitter and receiver units.

Initially, all hardware components such as Arduino Nano, ESP8266, LoRa modules, sensors, relay, and LED strips were assembled and tested individually. Each sensor was verified for proper operation. The DS18B20 sensor was tested for accurate temperature readings, the vibration sensor was checked for digital output response, and the ACS712 current sensor was calibrated to obtain stable readings.

After verifying individual components, the transmitter unit was developed by interfacing all sensors with the Arduino Nano. The LoRa transmitter module was connected using SPI communication pins, and the program was written to read sensor values, process them, and transmit the data wirelessly.

The receiver unit was then implemented using ESP8266 NodeMCU. The LoRa receiver module was connected, and the code was developed to receive data, extract parameters, and display them on an I2C LCD. Wi-Fi connectivity was established, and the MQTT protocol was used to send data to the Ubidots cloud platform.

Finally, the complete system was tested under different operating conditions by varying the load using LED strips. Fault detection logic was verified, and the system's performance was evaluated.

5.2 MODULE DESCRIPTION

The system is divided into four main modules to simplify design and implementation:

5.2.1 Module 1: Transmitter Side

The transmitter module is responsible for sensing and transmitting data.

Components:

- Arduino Nano
- DS18B20 Temperature Sensor
- Vibration Sensor
- ACS712 Current Sensor
- LoRa SX1278 (Transmitter)

Functionality:

The DS18B20 sensor measures the temperature of the system. The vibration sensor detects abnormal vibrations, and the ACS712 measures the current flowing through the load. The Arduino Nano processes all sensor readings and determines the system's condition.

Based on the readings, the Arduino identifies:

- Normal condition
- Overload condition
- Power failure
- Vibration fault

5.2.2 Module 2: Receiver Side

The receiver module is responsible for receiving, displaying, and uploading data.

Components:

- ESP8266 NodeMCU
- LoRa SX1278 (Receiver)
- I2C LCD Display
- Wi-Fi Connection
- Ubidots Cloud

Functionality:

The LoRa receiver module receives the transmitted data from the transmitter unit. The ESP8266 processes the received data and extracts the parameters.

The data is displayed on the I2C LCD in real time, showing temperature, current, and system status. The ESP8266 also connects to Wi-Fi and sends the data to the Ubidots cloud platform using the MQTT protocol.

This enables both local and remote monitoring of industrial equipment.

5.2.3 Module 3: Closed-Loop Control Mechanism This module demonstrates load control and fault simulation.**Components:**

- Relay Module

- LED Strip Load • ACS712 Current Sensor

Functionality:

The relay is used to control the LED strip load. When the load is connected, current flows through the ACS712 sensor. The system monitors the current continuously.

By increasing the number of LED strips, the current increases, simulating an overload condition. The Arduino detects this condition and updates the system status.

Although this is not a fully automated feedback control system, it acts as a closed-loop demonstration where the system responds to changes in load conditions.

5.2.4 Module 4: Multi-Channel Notification Strategy

This module ensures that the system provides information through multiple channels.

Notification Channels:

- LCD Display (Local Monitoring)
- Serial Monitor (Debugging)
- Ubidots Cloud Dashboard (Remote Monitoring)

Functionality:

The system displays real-time data on the LCD, allowing users to monitor machine conditions locally. The serial monitor is used during development for debugging and verifying data.

The most important feature is cloud integration. The ESP8266 sends data to Ubidots using the MQTT protocol, where users can view data remotely through dashboards. This enables real-time monitoring from any location.

Future enhancements can include:

- SMS alerts
- Mobile app notifications
- Email alerts

5.3 TECHNIQUES USED

The proposed system integrates multiple techniques to achieve efficient monitoring, communication, and fault detection. These techniques ensure accurate data acquisition, reliable transmission, and real-time monitoring.

1. Sensor-Based Data Acquisition

The system uses multiple sensors to collect machine parameters:

- DS18B20 for temperature measurement
- Vibration sensor for detecting mechanical disturbances
- ACS712 for measuring current

These sensors provide continuous real-time data, which forms the basis for fault detection.

2. Analog-to-Digital Conversion (ADC)

The ACS712 current sensor produces an analog output proportional to current. The Arduino Nano converts this analog signal into digital values using its built-in ADC. Multiple readings are averaged to improve accuracy and reduce noise.

3. Threshold-Based Fault Detection

The system uses predefined threshold values to detect faults:

- High current → Overload condition
- No current → Power failure
- Vibration detected → Mechanical fault

This method is simple, efficient, and suitable for real-time applications.

4. LoRa Wireless Communication

LoRa technology is used for long-range communication between transmitter and receiver. It operates at 433 MHz and provides:

- Low power consumption
- High communication range
- Reliable data transmission

5. MQTT-Based Cloud Communication

The ESP8266 uses MQTT protocol to send data to the Ubidots cloud platform. MQTT is a lightweight protocol designed for IoT applications, enabling efficient data transfer with minimal bandwidth usage.

6. I2C Communication

The I2C protocol is used to interface the LCD display with ESP8266. It reduces wiring complexity by using only two pins (SDA and SCL).

7. Embedded Programming

- Sensor reading
- Data processing • Fault detection
- Communication

8. Load Simulation Technique

LED strips are used as electrical loads to simulate industrial conditions. By varying the number of LED strips, different load conditions such as normal and overload are created.

5.4 PERFORMANCE METRICS / PERFORMANCE EVALUATION

The performance of the system is evaluated based on accuracy, efficiency, reliability, and response time. The system was tested under different load conditions to verify its performance.

5.4.1 ACCURACY

Accuracy refers to how closely the measured values match the actual values.

Temperature Accuracy

The DS18B20 sensor provides accurate temperature readings with minimal deviation. The readings were consistent and reliable during testing.

Current Measurement Accuracy

The ACS712 sensor provided stable readings after averaging multiple samples. The system successfully differentiated between:

- No load
- Normal load
- Overload

The variation in ADC values was sufficient to detect different conditions accurately.

Vibration Detection Accuracy

The vibration sensor provided a clear digital signal indicating vibration. The system was able to detect vibration faults effectively when triggered.

Overall Accuracy

The system demonstrated good accuracy in detecting machine conditions. Minor variations due to noise were reduced using averaging techniques.

5.4.2 OVERALL SYSTEM EFFICIENCY

System efficiency is evaluated based on communication speed, reliability, and power consumption.

Communication Efficiency

LoRa communication provides stable and reliable data transmission over long distances. No significant data loss was observed during testing.

Processing Efficiency

The Arduino Nano processed sensor data quickly and transmitted it without delay. The ESP8266 efficiently handled data reception and cloud communication.

Power Efficiency

The system uses low-power components such as Arduino Nano, ESP8266, and LoRa modules, making it energy-efficient and suitable for continuous operation.

Response Time

The system responded quickly to changes in load and vibration conditions. The fault detection was almost instantaneous.

Cloud Performance

The Ubidots platform successfully displayed real-time data with minimal delay. Data updates were consistent and reliable.

5.5 SUMMARY

This chapter discussed the implementation of the proposed industrial monitoring system. Various techniques such as sensor-based data acquisition, LoRa communication, MQTT protocol, and threshold-based fault detection were used to build the system.

The system was tested under different conditions, and its performance was evaluated based on accuracy and efficiency. The results showed that the system is capable of accurately detecting machine conditions and transmitting data reliably.

The integration of wireless communication and cloud monitoring makes the system efficient, scalable, and suitable for real-world industrial applications.

CHAPTER 6 RESULTS AND DISCUSSION

The proposed Smart Industrial Equipment Monitoring System was successfully implemented and tested under various operating conditions. The system was evaluated based on its ability to monitor temperature, vibration, and current parameters and detect fault conditions accurately.

During testing, different load conditions were created using LED strips to simulate industrial equipment behavior. The ACS712 current sensor measured the current variations effectively, and the system was able to classify operating conditions as normal, overload, or power failure.

When no load was connected, the system displayed a power failure condition. When a single LED strip was connected, the system operated under normal conditions. When multiple LED strips were connected, the current increased, and the system detected an overload condition. This confirmed the effectiveness of fault detection logic.

The vibration sensor successfully detected abnormal vibration conditions. When the sensor was triggered, the system displayed a vibration fault, indicating proper functioning of the vibration detection mechanism.

The DS18B20 temperature sensor provides stable and accurate temperature readings. The system continuously monitors temperature, which is essential for detecting overheating in industrial equipment.

LoRa communication between the transmitter and receiver was tested and found to be reliable. Data was transmitted successfully without significant delay or loss. This demonstrates the suitability of LoRa technology for longrange industrial communication.

At the receiver side, the ESP8266 processed the received data and displayed it on the I2C LCD. The LCD provided real-time information about temperature, current, and system status. Additionally, the ESP8266 successfully transmitted data to the Ubidots cloud platform using the MQTT protocol. The cloud dashboard displayed real-time updates, enabling remote monitoring.

Overall, the system performed efficiently in terms of accuracy, communication reliability, and response time. The

integration of LoRa and IoT technologies proved to be effective for industrial monitoring applications.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The Smart Industrial Equipment Monitoring System using LoRa communication has been successfully designed and implemented. The system effectively monitors important machine parameters such as temperature, vibration, and current, and detects fault conditions in real time.

The use of LoRa communication enables long-range wireless data transmission with low power consumption, making the system suitable for large industrial environments. The integration of ESP8266 with IoT cloud platforms allows remote monitoring and data visualization, enhancing system usability.

The system demonstrates the ability to detect overload, vibration faults, and power failure conditions accurately. The use of LED strips as load provides a practical way to simulate industrial conditions and validate system performance.

Overall, the proposed system offers a cost-effective, scalable, and efficient solution for industrial monitoring. It improves reliability, reduces downtime, and supports predictive maintenance, making it highly relevant for modern industrial applications.

7.2 FUTURE WORK

Although the system performs effectively, it can be further enhanced with additional features and improvements:

- Integration of mobile application for real-time alerts and notifications
- Implementation of SMS or email alert system for fault conditions
- Expansion of the system to monitor multiple machines simultaneously
- Integration of additional sensors such as humidity, pressure, and gas sensors
- Development of AI-based predictive maintenance using machine learning algorithms
- Use of battery-powered sensor nodes for remote deployment
- Implementation of advanced data analytics and visualization techniques
- Improvement in fault detection accuracy using advanced algorithms

These enhancements will further improve the system's functionality and make it more suitable for large-scale industrial applications.

REFERENCES

1. Semtech Corporation, "LoRa Modulation Basics," Application Note AN1200.22, 2015.
2. S. Raza, P. Misra, Z. He, and T. Voigt, "Building the Internet of Things with LoRa," IEEE Communications Magazine, vol. 55, no. 9, pp. 44–51, Sept. 2017.
3. A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," Sensors, vol. 16, no. 9, pp. 1–18, 2016.
4. R. Want, "An Introduction to IoT and MQTT Protocol," IEEE Internet of Things Journal, vol. 3, no. 1, pp. 1–9, Feb. 2016.
5. P. Ferrari, A. Flammini, E. Sisinni, and D. Rinaldi, "Evaluation of LoRa for Industrial IoT Applications,"

IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 9, pp. 1–10, 2019.

6. J. Granjal, E. Monteiro, and J. S. Silva, “*Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues*,” IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1294–1312, 2015.

7. K. Ashton, “*That ‘Internet of Things’ Thing*,” RFID Journal, 2009.

APPENDIX

Appendix A: Transmitter Code

```
#include <SPI.h> #include <LoRa.h> #include <OneWire.h>
#include <DallasTemperature.h>
```

```
#define ss 10
```

```
#define rst 9
```

```
#define dio0 2
```

```
#define ONE_WIRE_BUS 6
```

```
#define vibrationPin 4 #define currentPin A0
```

```
OneWire oneWire(ONE_WIRE_BUS); DallasTemperature sensors(&oneWire);
```

```
float temperature; int currentValue; int vibration;
String faultStatus void setup() {
  Serial.begin(9600);
```

```
  pinMode(vibrationPin, INPUT);
```

```
  sensors.begin();
```

```
  LoRa.setPins(ss, rst, dio0);
```

```
  if (!LoRa.begin(433E6))
```

```
  {
```

```
    Serial.println("LoRa Failed!"); while (1);
```

```
  }
```

```
  Serial.println("Transmitter Ready");
```

```
}
void loop()

{

// Temperature reading sensors.requestTemperatures(); temperature = sensors.getTempCByIndex(0);

// Averaged Current reading long total = 0; for(int i = 0; i < 20; i++)
{

total += analogRead(currentPin); delay(2);
}    currentValue = total / 20;

// Vibration reading          vibration = digitalRead(vibrationPin);

//      Fault Detection      Logic if(currentValue < 635)
{

faultStatus = "OVERLOAD FAULT";

}

else if(currentValue > 641)

{

faultStatus = "POWER FAILURE";
}    else if(vibration == HIGH)
{

faultStatus = "VIBRATION FAULT";

}

else

{

faultStatus = "NORMAL";

}

// Serial Monitor Output Serial.print("Temp:"); Serial.print(temperature); Serial.print("C Current:");
Serial.print(currentValue); Serial.print(" Vibration:"); Serial.print(vibration); Serial.print(" Status:");
Serial.println(faultStatus);

// Send via LoRa
```

```
LoRa.beginPacket(); LoRa.print("Temp:");
LoRa.print(temperature); LoRa.print(",Current:"); LoRa.print(currentValue); LoRa.print(",Vibration:");
LoRa.print(vibration); LoRa.print(",Status:"); LoRa.print(faultStatus);
```

```
LoRa.endPacket();
```

```
delay(2000);
```

```
}
```

Appendix B: Receiver Code

```
#include <SPI.h> #include <LoRa.h> #include <Wire.h> #include
```

```
<LiquidCrystal_I2C.h> #include "UbidotsESPMQTT.h" #define ss D8 #define rst D0 #define dio0 D1
#define TOKEN "BBUS-6UDEBaQxhhLuxlBgFggc20gqRzTGKG" #define WIFINAME "Lora" #define WIFIPASS
"12345678"
```

```
Ubidots client(TOKEN); LiquidCrystal_I2C lcd(0x27,16,2); float temperature; int current; int vibration; String status;
void callback(char* topic, byte* payload, unsigned int length) { Serial.print("Message
arrived
```

```
[");
```

```
Serial.print(topic); Serial.print("]"); Serial.println(); } void setup() { Serial.begin(9600);
```

```
Wire.begin(D4,D3); lcd.begin(); lcd.backlight(); client.wifiConnection(WIFINAME,WIFIPASS);
client.begin(callback); LoRa.setPins(ss,rst,dio0); if(!LoRa.begin(433E6)) { Serial.println(" LoRa Failed"); while(1); }
lcd.print(" Receiver Ready"); } void loop() { if(!client.connected()) { client.reconnect(); } int packetSize =
LoRa.parsePacket(); if(packetSize) { String received="";
while(LoRa.available())
```

```
{
```

```
received+=(char)LoRa.read();
```

```
}
```

```
Serial.println(received);
```

```
int tIndex = received.indexOf("Temp:"); int
```

```
cIndex = received.indexOf(",Current:");
```

```
int
```

```
vIndex = received.indexOf(",Vibration:");
```

```
sIndex = received.indexOf(",Status:");
```

```
temperature = received.substring(tIndex+5,cIndex).toFloat();
```

```
current = received.substring(cIndex+9,vIndex).toInt();
```

```
vibration = received.substring(vIndex+11,sIndex).toInt();
```

```

status = received.substring(sIndex+8);

lcd.clear(); lcd.setCursor(0,0); lcd.print("T:"); lcd.print(temperature);

lcd.setCursor(8,0); lcd.print("V:"); lcd.print(vibration);

lcd.setCursor(12,0); lcd.print(current);

lcd.setCursor(1,1); lcd.print(status);

client.add("temperature", temperature); client.add("current",current); client.add("vibration",
vibration);

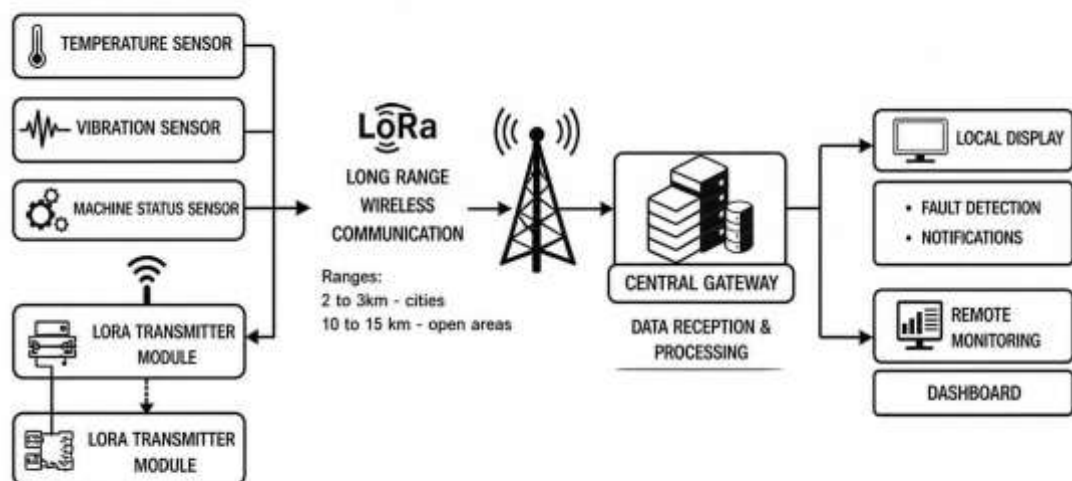
client.ubidotsPublish("machine_monitor");
} client.loop();}

```

Appendix C:

Circuit Diagram

Appendix D: Block Diagram



Appendix E: Output Screenshots

