ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2579

Lora Based Industrial Environment Monitoring System

Gaddala Nagesh[1]; Konda Karthik[2]; Nagula Ritish Kumar[3]; Dr. A.N. Satyanarayana[4]
[1,2,3]Students of B.tech, ECE Department, Sreenidhi Institute of Science and Technology, Hyderabad.
[4]Professor of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology

Abstract:- With the advent of automation, the concept of protecting the business environment has become popular and recognized in the business world. As the factory scale grows and the complexity of the process increases, advanced technology must be used in the device to measure and control many business processes simultaneously. Today, the trend is towards miniaturization of instruments and greater precision and accuracy. Reliable equipment and controls are crucial to the safety and efficiency of any process. The goal of this project is to gain knowledge about the most important aspects of the job to improve safety.

The system includes the use of LORA to monitor the business environment. In this project, we use Lora communication to send data, and the receiving end of the monitoring station receives the data from the sending end. In an emergency, when you press the emergency button on the monitoring station, a message is sent to the work unit by LORA. Since LORA is a transceiver, two-way communication is possible in the project. Temperature sensors, gas sensors and fire sensors are used to monitor the environment of business units. Temperature sensors are used in industry to calculate temperature and provide information to microcontrollers. Gas sensors are used in industry to detect gas leaks. Fire sensors are used to detect fire and send the information to the monitoring station. This product is delivered by LORA.

Keywords:- *LoRa Communication, Arduino, Monitoring.*

I. INTRODUCTION

The project aims to create a business automation central factory monitoring system focused on data collection, storage and transmission. The system uses an Arduino microcontroller and LoRa wireless communications to monitor and control multiple factory parameters.

Collecting information from the manufacturer in the automation industry is important for optimization and security. This project only solves three things: data collection, storage/display, and wireless communication. The microcontroller used by the Arduino ATMEGA 328 chip does the main work by taking data from the factory parameters and displaying them on the LCD panel.

Industrial automation relies on microcontrollers, which are widely used in embedded systems. From home appliances to car remotes, microcontrollers play an important role in modern technology. Precise measurement is essential for automatic control and protection in a work environment. Microcontrollers facilitate the encoding and decoding techniques required for digital communications.

Wireless communications, specifically LoRa technology, can transfer data from the production site to a central monitoring station. The system reads the meter and measurement data and transmits the data through the LoRa transceiver. This allows two-way communication, making it easy to quickly monitor and control.

Radio communication uses electromagnetic waves as the basis of wireless communication. In this project, digital communication with efficiency and reliability advantages was adopted. The system transmits high-quality digital data by changing the radio frequency.

The design operating frequency of this communication is approximately 433MHz. Digital data is superimposed on the carrier wave and transmitted as a modulated signal. Finally, when data is received, the data is demodulated, decoded and compared to the database. Any suspicious activity, such as making noise or displaying unusual information, will trigger the alarm.

Overall, this project integrates embedded systems and wireless communication technologies to create a powerful centralized factory monitoring system. It increases efficiency and security in the business environment by automating data collection and transmission. The detailed description in the report shows the description of physical products, functions and practical applications in the automation industry. configuration.

II. METHODOLOGY

The majority of actual physical quantities, including voltage, current, and temperature, are available in analog form. Even though an analog signal accurately represents a true physical characteristic, noise superimposition, as in the case of amplitude modulation, makes it impossible to process, store, or transmit the analog signal without significantly increasing inaccuracy. Therefore, it is frequently convenient to express this variable in digital form for processing,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2579

transmission, and storage needs. It improves the process automation's precision.

At a certain frequency rate, the transducer's analog signal is sampled. The sampled signal needs to be kept steady while the ADC converts it. This means that a sample and hold circuit integrated into the same device must come before the ADC. A binary digit sequence is the ADC's output. The controller employed in this project work is used to carry out the desired control algorithm's numerical computations. Since the Arduino controller has an integrated ADC, the sensor output is immediately sent into the controller's ADC pins.

The integrator type and sequential approximation are the most often utilized ADCs. In applications where conversion speed is crucial, including data loggers and instrumentation, successive approximation ADCs are employed. Compared to integrating type converters, the successive approximation and comparator type are often quicker but less precise. The flash (comparator) kind costs more for a higher level of precision. Applications where conversion precision is crucial, such digital meters, panel meters, and monitoring systems, employ the integrated type converter. In this instance, the ADC and microcontroller are interfaced with the goal of reading the data acquired from the ADC and executing the control action in accordance with it. The following is a basic overview of microcontrollers and control theory as it relates to ADCs.

As described in the abstract and according to the program prepared, here the microcontroller controls only three channels that is temperature sensing channel, gas monitoring channel and fire sensing channel. The following is the description of individual sensing circuits.

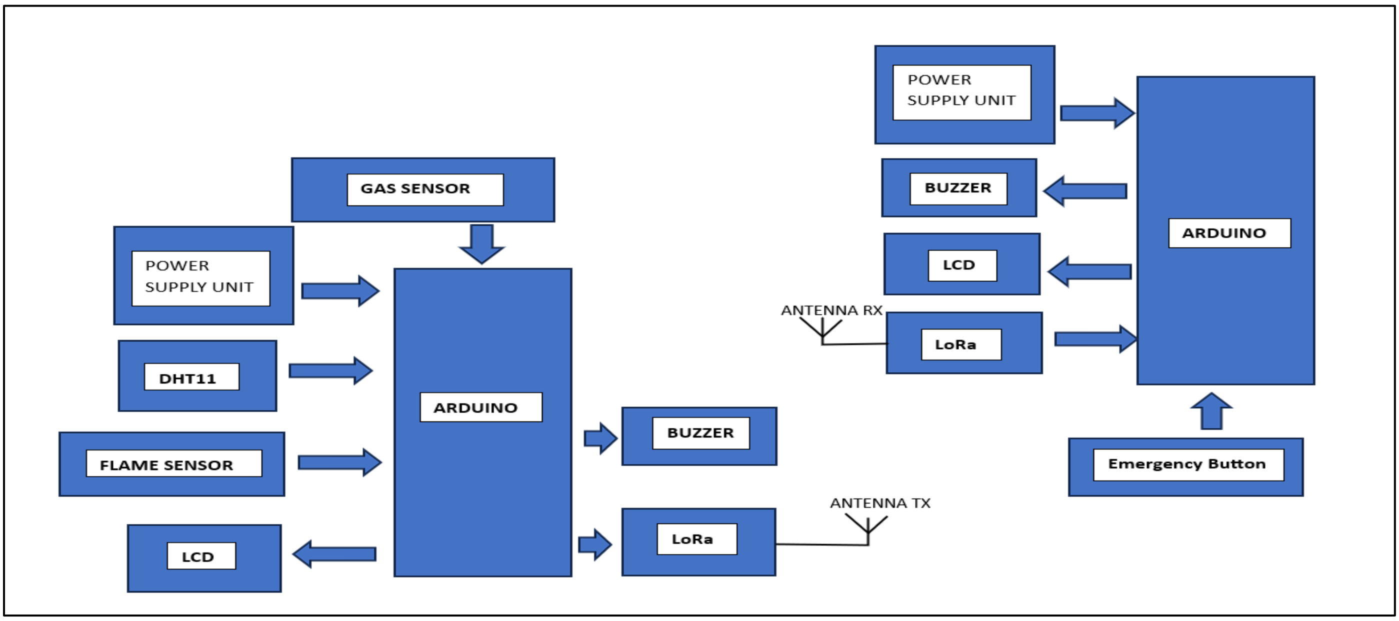

Fig. 1 Block Diagram

III. COMPONENTS

➢ *Temperature Sensor*

There are two types of temperature measurement techniques: contact and non-contact. Through the use of thermocouples, contact techniques entail direct heat exchange between the sensor and the item being measured. Non-contact techniques pick up heat energy sent into space. Because of its precision and dependability, an RTD (Resistance Temperature Detector) was selected for this project.

Various temperature sensing techniques exist, including RTDs, thermocouples, thermistors, and sensor ICs. The selection depends on factors like temperature range, accuracy, and cost. Due to affordability concerns, a potentiometer is used in this project as a prototype.

Serial communication over a single wire is used in this project. A DHT module receives a start signal from an Arduino and returns temperature and humidity information. A 16x2 LCD is used to show the temperature data after it has been extracted. The DHT11 sensor module produces a calibrated digital output signal together with accurate humidity and temperature data.

The DHT11 module operates on serial communication, transmitting data in a specific pulse train format. Arduino initializes the module with a delay and receives response signals to extract temperature data. The data format consists of integral and decimal parts for relative humidity (RH) and temperature (T), along with a checksum.

It is advised that DHT sensors situated within 20 meters use a 5k pull-up resistor. The LCD and Arduino are directly connected to display temperature and humidity. Certain digital pins on the Arduino board are connected to pins on the LCD and DHT11 sensor module.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2579

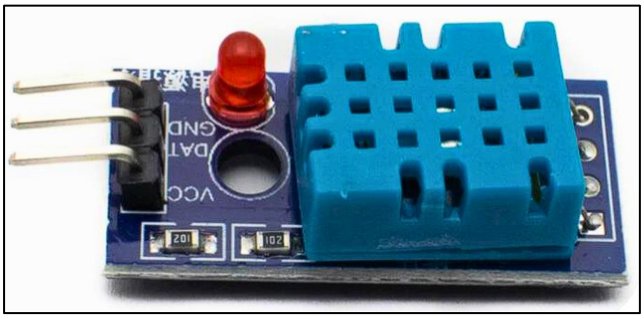

Fig 2 DHT11 Sensor

➢ *Flame Sensor*

The project includes an IR diode module for fire detection, using IR sensor technology to detect the unique spectral patterns emitted by hot fire-related gases in the IR region. Weighing only 5 grams, this compact module is easy to install and responsive.

When a fire is detected, the module generates a high output signal that can be used to trigger appropriate actions, such as wireless transmission of information to a monitoring unit using LoRa technology. In addition, the built-in LED provides a visual indication of fire detection, increasing user awareness.

With a fast response time of only 1 second, the IR diode module ensures early detection of fires and enables quick and effective responses to potential fire hazards.

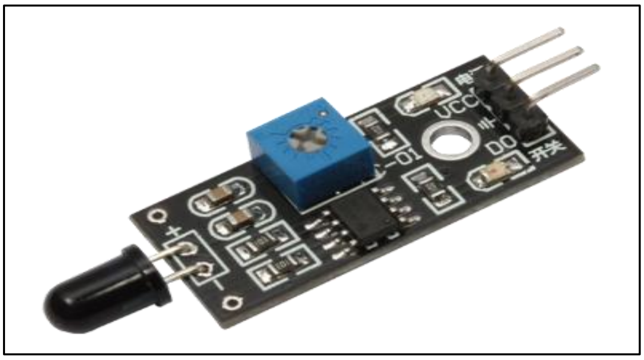
Fig 3 Flame Sensor

➢ *Gas Sensor*

MQ2 is a versatile sensor that exhibits strong sensitivity for a wide range of gases. This device requires a consistent 5V power supply to function; the voltage can vary according on the model. The MQ2 is ideal for residential gas or smoke detectors because it can detect methane, propane, and butane.

The stable working time of MQ2 is very short, and its properties after long working time are stable and excellent. The MQ2 is sensitive to "noise gases", greatly reducing the problem of false alarms. The most effective use of the MQ2 is in power generation, where it controls the voltage at a constant value of 5V. Due to its wide range of applications, this power factor is useful when determining the design. This makes the use of MQ2 a unique industry opportunity to produce low-energy, reliable fuel cells.

MQ2 is particularly sensitive to methane, propane and butane, making it very suitable for urban maintenance and LPG. With additional features such as short-term stability and reliable switching features, MQ2 represents Figaro's new generation of fuel sensors. These sensors are molded in resin capabilities.

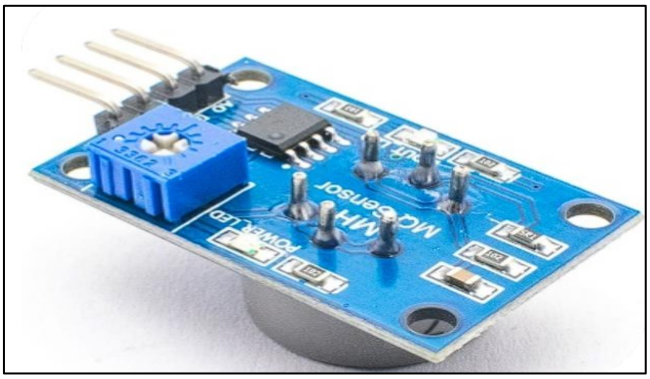
Fig 4 MQ2 GAS Sensor

➢ *LCD*

A serial computer bus, as seen in Figure 5, is a small interface that links processors and microcontrollers to peripheral integrated circuits (ICs) operating at lower speeds. This bus is essential to our project since it makes it possible for the LCD display and Arduino microcontroller to connect.

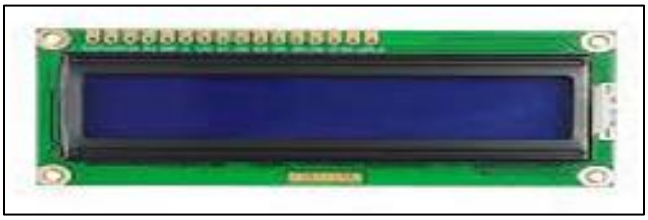
Fig 5 LCD Connected to ESP32

➢ *LoRa*

LoRa, or long-range wireless technology, is intended for low-power, long-distance communication between gateways and sensors in Internet of Things (IoT) and machine-to-machine (M2M) applications. It uses spread-spectrum technology with frequency-modulated chirp for enhanced receiver sensitivity and works in unlicensed radio spectrum channels.

LoRaWAN, built upon LoRa technology by the LoRa Alliance, is an open-source LPWAN (Low Power Wide Area Network) protocol specification. It enables the creation of IoT networks with bi-directional secure communication, interoperability, and mobility, supporting accurate localization.

Introduced by the French start-up Cycle in 2010 and later acquired by SEMTECH, LoRa offers exceptional sensitivity and coverage, reaching up to -148 dBm with a diffusion factor of 12. It typically provides a range of 3 km in urban areas and 14 km in rural areas, with battery life extending up to 10 years.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2579

LoRa modules offer different classes (A, B, and C) based on power consumption and delay capabilities, catering to various IoT device requirements. These modules can connect to millions of end devices, supporting applications such as smart metering, inventory tracking, and automotive industry needs.

Ultra-long-range spread spectrum communication, strong interference immunity, and low current consumption are features of Semtech's LoRa transceivers. Sensitivities of -137dBm and -148dBm are possible because to the proprietary modulation approach, which makes them perfect for IoT applications that need resilience and range.

The project utilizes the SX1278 Ra-02 LoRa module operating at 433MHz. It's essential to ensure compliance with local regulations regarding frequency usage, as LoRa modules are available in different frequency ranges (433MHz, 915MHz, 868MHz). Additionally, LoRa modules can be purchased as standalone units or integrated chips.

➢ *Arduino UNO*

Using the ATmega328 as its foundation, the Arduino Uno is a microcontroller board. A 16 MHz ceramic resonator, 6 analog inputs, a USB port, a power jack, an ICSP header, a reset button, and 14 digital input/output pins are all included on it. This microcontroller comes with all the necessary components to operate it; all you need to do is use a USB cable to connect it to a computer or an AC-to-DC adapter or battery to power it. With regard to using the FTDI USB-to-serial driver chip, the Uno is different from all previous boards. Rather, it has the Atmega16U2 (or Atmega8U2 up to version R2) configured as a serial-to-USB converter using programming. To facilitate DFU mode booting, the Uno board revision 2 has a resistor that pulls the 8U2 HWB line to ground.

The following new features are included in the board's revision 3: ϖ 1.0 Pinout: two more new pins, the IOREF, are positioned next to the RESET pin and the SDA and SCL pins that were introduced to allow the shields to adjust to the voltage supplied by the board. Future shields will work with the Arduino Due, which runs on 3.3V, as well as boards that use the AVR, which runs on 5V. A disconnected pin designated for future use is the second one. A more robust RESET circuit. An Atmega 16U2 in instead of an 8U2.

"Uno" is an Italian word for one, and it was chosen to commemorate the impending introduction of Arduino 1.0. Going future, the Arduino reference versions will be the Uno and version 1.0. The Uno is the most recent in a line of USB Arduino boards and the platform's reference model; view the index of Arduino boards for a comparison with earlier iterations.

➢ *Overview:*

- ATmega328 microcontroller;
- 5V operating voltage;
- 7–12V suggested input voltage;
- 6–20V limited input voltage;
- 14 digital I/O pins (six of which give PWM output)
- Pins 6 for analog input;
- 40 mA for DC current per I/O pin;
- 50 mA for DC current per 3.3V pin;
- 32 KB of flash memory (ATmega328), of which 0.5 KB is used by the bootloader;
- 2 KB for SRAM; and 1 KB for EEPROM
- Clock Speed 16 MHz

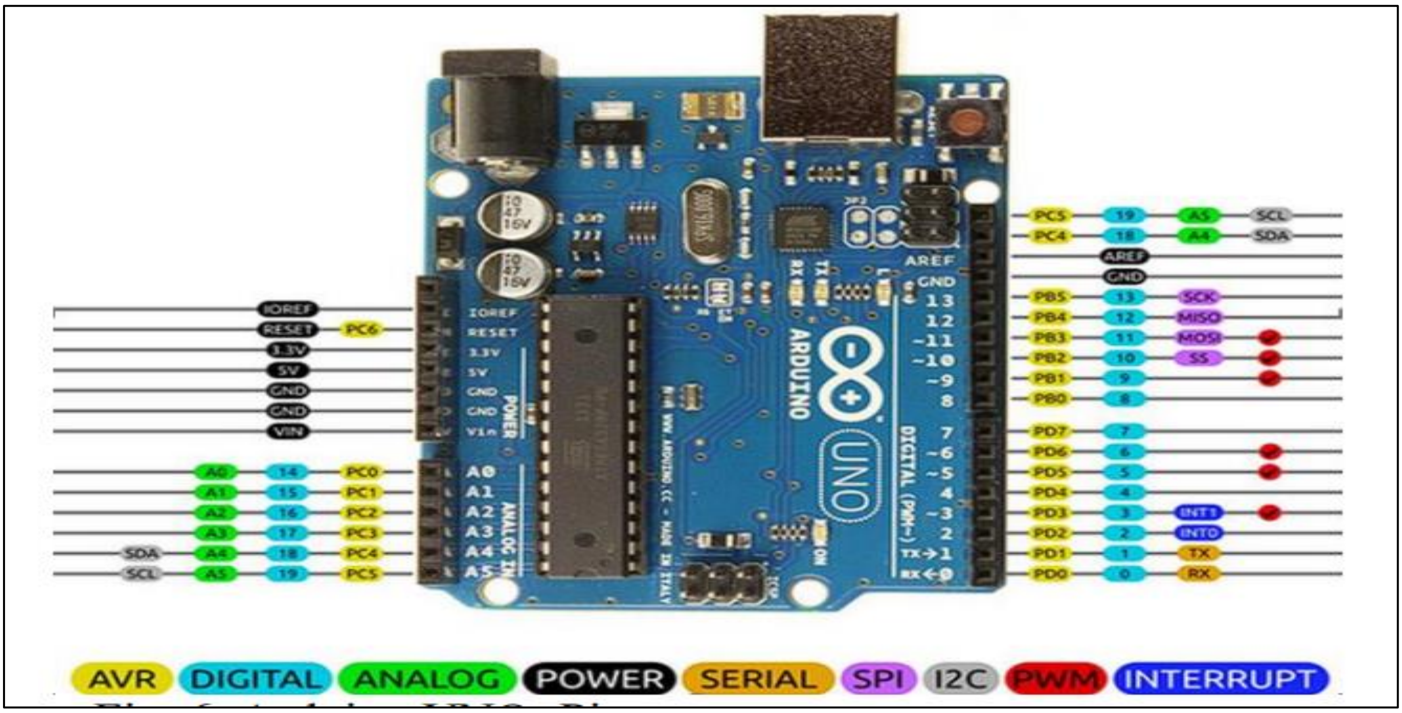

Fig 6 Arduino UNO: Pinout

IV. IMPLEMENTATION

The transmitter phase and receiver phase are the two components of the Lora protocol, a real-time wireless monitoring framework, that we apply in our research.The power supply (12 volts), voltage sensor (5 volts), current sensor (5 volts), and temperature sensor make up the receiver phase.

Another Arduino microcontroller with an LCD that shows the received data makes up the transmitter phase.

The value of the current transformer analogy aids in figuring out the voltage and current input values that are supplied to the Arduino microcontroller via that sensor. The temperature sensor's value is also transmitted to the microcontroller.

It's possible that the LCD attached to the Arduino microcontroller is showing all of these figures. Typically, the relay module that is attached to the load uses low power input to control the high-power output.

The Lora protocol, which enables long-distance, low-power communication, is employed throughout the entire transmission process. Imagine that in the event of an overvoltage in the equipment, the Arduino microcontroller uses the Lora protocol to activate the buzzer on the transmitter side. Manually turning off the machinery is possible with the keypad.

V. RESULTS

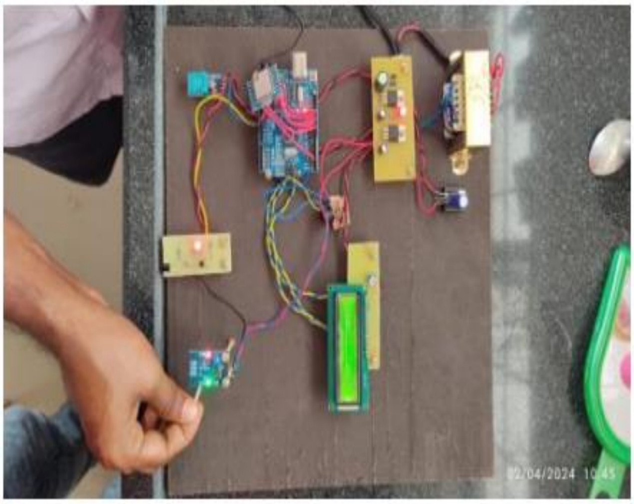

Fig 7 Harmful Gas Detection at Transmitter

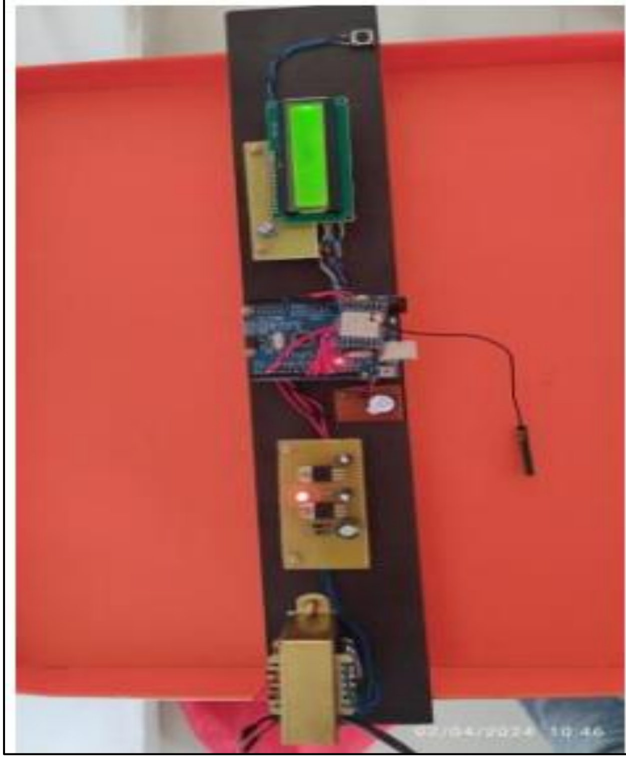

Fig 8 Receiver Showing Alert Message

VI. CONCLUSION

The "Industrial Environment Monitoring using LoRa" project work has been successfully conceived and developed. A prototype module is built for demonstration purposes, and the findings are deemed adequate. Only two parameters are taken into account because it is a prototype module, but with minimal modifications, other parameters like voltage, current, pressure, humidity, light, etc. can also be measured in the same system. Since this device has numerous I/O lines, the system is now designed with a controller. However, the system can be expanded to include more facilities and be controlled.

The main drawback of using microcontroller is that it cannot store the information, presently the system is designed to display the present values with control action if necessary, but the previous data cannot be stored. So to achieve this if a computer is interfaced to this system, the data can be stored from time to time.

The main advantage of using computer is, if required the data of every individual parameter can be stored independently and failure causes information can also be stored. This information is quite useful for the user for the analysis of the various parameters of the industry.

FUTURE WORK

A system optimized for monitoring, controlling, and long-distance communication with the machinery. Additionally, it warns the user to manually stop the equipment in order to avert additional collapse. In keeping with this, we have intended to identify potential future harm to machinery in prior updates.

REFERENCES

[1]. The 8051 Micro-controller Architecture, programming & Applications By: Kenneth J. Ayala
[2]. The 8051 Microcontroller and Embedded Systems By: MD Ali Mazidi
[3]. Practical transistor circuit design and analysis By: GERALD E. WILLIAMS
[4]. Programming and Customizing the 8051 Micro-controller By: Myke Predko
[5]. The concepts and Features of Micro-controllers - By: Raj Kamal
[6]. Electronic Circuit guide book – Sensors – By JOSEPH J.CARR
[7]. Mechanical and Industrial Measurements – R. K. Jain
[8]. Yookesh, T. L., et al. "Efficiency of iterative filtering method for solving Volterra fuzzy integral equations with a delay and material investigation." Materials today: Proceedings 47 (2021): 6101-6104.Digital Electronics. By JOSEPH J.CARR
[9]. Electronics for you – Monthly Magazine
[10]. Y. Yi, H. Zhao, and Y. Wang, “LoRa Signal Monitoring System of Multi-Node Software Define Radio,” in 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2020, pp. 1–5.
[11]. Y. Ouo and Z. Liu, “Time-Delay-Estimation-Liked Detection Algorithm for LoRa Signals over Multipath Channels,” IEEE Wireless Communication. Lett., 2020.
[12]. Hammi, B.; Khatoun, R.; Zeadally, S.; Fayad, A.; Khoukhi, L. IoT technologies for smart cities. IET Netw. 2018,. 1–13.17.Z
[13]. Kavya.R, Gowripriya.K, J.Dhanasekar, “ IEEE 802.15.4: A Wireless Communication technology”, International Journal of Emerging Trends in Science and Technology, Vol 1, Issue 3, October 2015.
[14]. P. J. Ryan and R. B. Watson, “Research Challenges for the Internet of Things: What Role Can OR Play?,” Systems, vol. 5, no. 1, pp. 1–34,2017.
[15]. C. Venkataraman, Hema M, Kalaivani C T, A Manjunathan. K Sheikdavood, D.Sindhu, A Manikandan “A Hybrid MAC Protocol for Different Densities of Clusters in Wireless Sensor Networks”, Design Engineering Issue: 5, Pages: 347 – 355, pp. 55-66, 2021.

CODE

- *Arduino Code :*

```
TX:
//3.3V-3.3V,GND-GND,NSS-D10,DIO0-D2,SCK-D13,MISO-D12,MOSI-D11,RST-D9
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(A1,A0, 6, 5, 4, 3);
#define DHTPIN 7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE); //Defining the pin and the dhttype
const int GAS=8;
const int FIRE=A2;
const int BUZZ=A3;
float h=0;
float t=0;
int counter=0;
int cnt=0;
char b[200];
void setup()

{lcd.begin(16, 2);
lcd.setCursor(0,0);
lcd.print("FIRE: T:00C");
lcd.setCursor(0,1);
lcd.print("GAS:  ");
Serial.begin(9600);
Serial.println("HI WELCOME!!");
pinMode(GAS,INPUT);
pinMode(FIRE,INPUT);
pinMode(BUZZ,OUTPUT);
digitalWrite(BUZZ,LOW);
if (!LoRa.begin(433E6)) {
Serial.println("LoRa start failed!");
while (1);}
dht.begin();}
void loop()

{int packetSize = LoRa.parsePacket();
if (packetSize)

{Serial.println(“packet Recieved '");
digitalWrite(BUZZ,HIGH);
while (LoRa.available())

{b[cnt]=LoRa.read();
lcd.print((char)b[cnt]);
cnt=cnt+1;
if(cnt==16)
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2579

```
{lcd.setCursor(0,1);}}
lcd.setCursor(0,0);
lcd.print("danger ");
lcd.setCursor(0,1);
lcd.print("  ");
while(1);}

if(digitalRead(GAS)==LOW)

{lcd.setCursor(4,1);
lcd.print("YES");
Serial.println("HARMFUL GASES!");
for(counter=0;counter<=10;counter++)

{Serial.print("Sending packet:");
Serial.println(counter);
LoRa.beginPacket();
LoRa.print("!HARMFUL GASES Detected");
LoRa.endPacket();}}
else

{lcd.setCursor(4,1);
lcd.print("NO ");
Serial.println("NO HARMFUL GASES");}
if(digitalRead(FIRE)==HIGH)
{lcd.setCursor(5,0);
lcd.print("YES");
Serial.println("FIRE DETECTED");
for(counter=0;counter<=10;counter++)
{Serial.print("Sending packet: ");
Serial.println(counter);
LoRa.beginPacket();
LoRa.print("ALERT!FIRE DETECTED");
LoRa.endPacket();}}
else
{lcd.setCursor(5,0);
lcd.print("NO ");
Serial.println("NO FIRE");}
t = dht.readTemperature();

Serial.print("TEMPERATURE:");
Serial.print(t);
Serial.println('C');

lcd.setCursor(13,0);
lcd.print(t);
lcd.setCursor(15,0);
lcd.print("C ");
if(t>36)
{for(counter=0;counter<=10;counter++)
{Serial.print("Sending packet: ");
Serial.println(counter);
LoRa.beginPacket();
LoRa.print("T:");
LoRa.print(t);
LoRa.print("C,HIGH TEMPERATURE");
LoRa.endPacket();}}}
```

RX:

```
//3.3V-3.3V,GND-GND,NSS-D10,DIO0-D2,
SCK-D13,MISO-D12,MOSI-D11,RST-D9
#include <LiquidCrystal.h>
LiquidCrystal lcd(A1,A0,6,5,4,3);
#include <SPI.h>
#include <LoRa.h>
const int key=8;
const int BUZZ=7;
int cnt=0;
char b[200];
int i=0;
int counter=0;
void setup()
{lcd.begin(16, 2);
lcd.setCursor(0,0);
lcd.print("WELCOME ");
pinMode(key,INPUT);
pinMode(BUZZ,OUTPUT);
digitalWrite(BUZZ,LOW);
digitalWrite(key,HIGH);
Serial.begin(9600);
Serial.println("ready");
if (!LoRa.begin(433E6)) {
Serial.println("Starting LoRa failed!");
lcd.setCursor(5,1);
lcd.print("error");
while (1);}}
void loop()
{lcd.setCursor(0,0);
int packetSize = LoRa.parsePacket();
if (packetSize)
{Serial.println("Received packet '");
digitalWrite(BUZZ,HIGH);
while (LoRa.available())
{b[cnt]=LoRa.read();
lcd.print((char)b[cnt]);
cnt=cnt+1;
if(cnt==16)
{lcd.setCursor(0,1);}}
delay(5000);
digitalWrite(BUZZ,LOW);
cnt=0;
for(i=0;i<50;i++)

{b[i]=0;}
i=0;}

if(digitalRead(key)==LOW)
{for(counter=0;counter<=10;counter++)
{Serial.print("Sending packet: ");
Serial.println(counter);
LoRa.beginPacket();
LoRa.print("DANGER!");
LoRa.endPacket();}}}
```