**Remote Gas Pipeline Tunnel Temperature Monitoring System**

**Submitted by:**

Ruchika Dubey

Bhumika Mundada

Manjushri Jiwane

**Objective of the project:**

**The main goal of the project is:**

To build remote gas pipeline tunnel temperature monitoring system to monitor the temperature and leakage of the gas inside the pipeline tunnels .

**Abstract:**

Due to poor ventilation conditions in the tunnel, if gas pipeline leaks, the consequence of the accident will be more serious. In the confined space of a tunnel, temperature rises easily, and the overheating could cause pipeline fractures that could lead to gas leaks or even explosions. This may cause loss of property, explosion and human loss, so in order to overcome that a remote monitoring system is designed.

|  |  |
| --- | --- |
| Sr no | Components used |
| 1 | Node MCU (ESP8266) |
| 2 | Gas sensor |
| 3 | Temperature sensor |
| 4 | Wires |
| 5 | Arduino IDE |
| 6 | Node RED |
| 7 | MIT app inventor |
| 8 | IBM cloud platform |

**Working Of the Project:**

To monitor the temperature and leakage parameters at various places routers can be placed which consists of the temperature ssssensor (LM35) to monitor temperature and gas sensor ( MQ 135 sensor) to detect any gas leakage. This is connected to the ESP8266 (ESP12E). The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. Then the data is send to the IBM cloud. IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS). With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources -- such as compute power, storage and networking over the internet. By using the Node-RED which is one of the application on the IBM cloud platform, we make flows using wide range of nodes in the palette that can be deployed to its runtime in a single-click .Which is a programming tool that wires together hardware devices, APIs and online services in new and interesting ways. The data is then visualized on the web app (MIT APP INVENTOR) from where the desired persons can monitor and take necessary actions when needed.

**Description about the requirements:**

**Hardware requirements :**

1. Node MCU: Its a self contained SOC with integrated TCP/IP protocol stack that can give any micro controller access to your WiFi network.
2. Temperature sensor: An example for a temperature sensor is LM35.The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius temperature. The LM35 is operates at -55˚ to +120˚C.
3. Gas sensor: The sensor used in this project is MQ135. Its a air quality monitor , sensitive towards benzene ,NH3,NOx, alcohol, smoke,CO2. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide and Benze steam, also sensitive to smoke and other harmful gases.

**Software Requirements:**

1. Ardino IDE :  IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board .
2. IBM cloud : IBM Cloud is a platform that helps developers build and run modern apps and services. . With services across mobile, IoT, IBM Watson and more, IBM Cloud is an ideal platform to power the next wave of apps that thrive on data. We will be using two if the IBM services NodeRed and IBM IOT Platform.
3. MIT app inventor: **App Inventor** for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (**MIT**), which allows newcomers to computer programming to create software applications for the Android operating system (OS).

**Flow diagram:**

Gas sensor

IIBM Cloud Platform

Node MCU

Temperature sensor

Mobile App

**PROJECT CODE :**

//Remote Gas Pipeline Tunnel Temperature Monitoring System

//gas sensor D2 and temp sensor A0

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "YOUJNTU1S12";

const char\* password = "YOUJNTU1S12";

#define ORG "dno83a"

#define DEVICE\_TYPE "ruchika"

#define DEVICE\_ID "1234"

#define TOKEN "12345678"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic1[] = "iot-2/evt/gas/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883,wifiClient);

int gas = D6;

const int LM\_35 = A0;

int input\_val = 0;

float temp = 0;

void setup()

{

Serial.begin(115200);

Serial.println();

Serial.print("Connecting to ");

Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

pinMode(gas, INPUT);

pinMode(LM\_35,INPUT);

}

void loop()

{

int gassensor = digitalRead(gas);

input\_val = analogRead(LM\_35);

temp = (input\_val \* 4.88);

temp = temp/10;

Serial.print("Temperature is : " );

Serial.println(temp);

Serial.print("Pin gas sensor: ");

Serial.println(gassensor);

delay(1000);

if (isnan(temp) || isnan(gassensor))

{

Serial.println("Failed to read from sensor!");

delay(1000);

return;

}

PublishData(temp,gassensor);

delay(1000);

}

void PublishData(float temp, int gassensor){

if (!!!client.connected()) {

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"temperature\":";

payload += temp;

payload+=",""\"gassensor\":";

payload += gassensor;

payload += "}}";

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(topic1, (char\*) payload.c\_str())) {

Serial.println("Publish ok");

} else {

Serial.println("Publish failed");

}

}

**Advantages of the Project:**

1. Wastage due to leak in pipelines can be rectified as we can detect the leakage and take actions accordingly.
2. Maintenance of pipelines can be easily done as we can detect where it should be repaired.
3. We can know where the temperature is exceeding or where there is a leakage and accident can be prevented.
4. Pipeline operators frequently face false alarms whereby the sensors are hypersensitive and to overcome this challenge, the use of wireless and remote technologies is becoming more prevalent.

**Future scope:**

1. The length of total Natural Gas Pipelines was 10246 Kms in India as on 31st March 2010 which increased by 17680.397 Kms as on 31st March 2016, so in the future there are a lot of projects to implement more pipeline transport in India so by the implementation of the project we can prevent leakage as well as accidents in future.
2. gas on the pipeline is actually turning out to be 30% cheaper than an LPG cylinder so in future more people will go for gas pipeline than LPG