IOT Based Air Quality Monitoring and LPG Gas Leakage Detection System

Auhona Ghosh Thushara Sanjeev Daya Shankar Singh Kartikey Mishra

Abstract — **An IOT Based Air Quality Monitoring and LPG Gas Detection System is a technology which can be used to monitor the Air Quality and detect gas leakage over a web server using internet and will trigger an alarm when the air quality exceeds beyond a certain limit which means that there are sufficient amount of harmful gases present in the air like CO2, smoke, alcohol, benzene and NH3 which are fatal to our health and lifestyle. It will show the air quality in PPM on the LCD and also as on webpage in order that we will monitor it very easily. We have used MQ135 sensor which is that the most suitable option for monitoring Air Quality because it can detect most harmful gases and may measure their amount accurately. In this IOT project, you'll monitor the pollution level from anywhere using your computer or mobile. We can install this technique anywhere and may also trigger some device when pollution goes beyond some level, like we will turn on the fan or can send alert SMS/mail to the user**

Keywords — **Air Quality Monitoring, LPG Detection System, MQ135 sensor**

I. INTRODUCTION

Air pollution has become a common phenomenon everywhere. Specially within the urban areas, pollution may be a real-life problem. A lot of individuals get sick only thanks to pollution. In the urban areas, the increased number of petrol and diesel vehicles and the presence of industrial areas at the outskirts of the major cities are the main causes of air pollution. The problem is seriously intensified in the metropolitan cities. Also, the climate change is now apparent. The governments all around the world are taking every measure in their capacity. Many European countries have aimed to replace petrol and diesel vehicles with the electric vehicles by 2030. Even India has aimed to do so by 2025.

Air quality in the Delhi area has dipped slightly the past month and climbing levels of dangerous particulate matter are being reported across the region. Data from the Delhi Pollution Control Committee and the Central Pollution Control Board show that levels of PM 2.5 -- particulate matter less than 2.5 microns in diameter that lodge deep in the lungs -- have climbed to very unhealthy levels in most parts of the Delhi and therefore the NCR. For example, based on the most recent updates as of 9.34 am, the PM 2.5 levels in Sector 16A Faridabad and Sanjay Nagar in Ghaziabad were 278 and 275, respectively. Air pollution in Delhi, one of the world's most polluted cities, has been the subject of intense public debate as it usually during the onset of winter and has received international attention. Continued exposure to environments with poor air quality is a major public health concern in developed and developing countries. It is estimated that the pollutants liable for poor air quality cause nearly 2.5 million premature deaths per annum world-wide. Significantly, around 1.5 million of these deaths are due to polluted indoor air, and it is suggested that poor indoor air quality may pose a significant health risk to more than half of the world’s population. Due to its link with industrialization, societal health problems related to poor air quality disproportionately affects developed and developing nations – it's estimated that pollution is responsible for the premature deaths. Remedial action to enhance air quality is usually easy to implement once air borne pollutants are detected

According to the reports, over 1500 LPG accidents happen in India a day. This is like the death of 1500 people including the youngsters. Even the neighborhood is affected by a single accident. Hence, there comes the need to bring in technology to prevent accidents. IoT may be a fast- growing technology in Industries, Cars. IoT is basis for Industry 4.0 development.

II. LITERATURE REVIEW

The level of pollution has increased with times by lot of things just like the increase in population, increased vehicle use, industrialization and urbanization which ends up in harmful effects on human wellbeing by directly affecting health of population exposed thereto. It is necessary to watch air quality and keep it in check for a far better future and healthy living for all. Due to flexibility and low-cost Internet of things (IoT) is getting popular day by day. With the urbanization and with the increase in the vehicles on road the atmospheric conditions have considerably affected. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. Monitoring gives measurements of air pollutant and sound pollution concentrations, which can then be analyzed interpreted and presented.

III. PROBLEM STATEMENT AND APPLICATION AREAS

It is now important to watch pollution in real time in most of the urban areas. This project is aimed toward developing an IOT device which may monitor pollution in real time and log data to a foreign server. Remote monitoring was facilitated using classical motes within the past, which has some pitfalls like limited memory, processing speed and sophisticated programming strategies. By using Internet of Things and recording sensor data to a foreign server, the restrictions of memory within the monitoring devices and manual collection of knowledge from the installed devices can be overcome. The IOT also helps monitoring the info in real time.

The advantage of this automated detection and alerting system over the manual method is that it offers quick reaction time and accurate detection of an emergency and successively leading faster diffusion of the critical situation. The Arduino plays a major role in this project. In this project gas leakage is identified by using sensors which works only in closed environment. In present situation there are many cases related to gas leakage which causes innocent people lives and property damage. Implementing this application are often useful for companies, houses, which may save lives of individuals.

Real-Time monitoring and analysis of air quality which can viewed on a webpage and notification system to alert a user when the air quality decreases in their area. The user can also view graphs with Air Quality in Y-Axis vs Time in X-Axis that can be used to track and monitor air quality and gas leakage at any specific time and can also be used to predict and analyze the air quality in the future. Predictive analysis is the next step. We can further use this technology to place air quality sensors around the city and map air quality in the city and create a website so users can track the pollution in their respective regions

IV. PROPOSED METHOD

A. *ARCHITECTURE DESIGN*

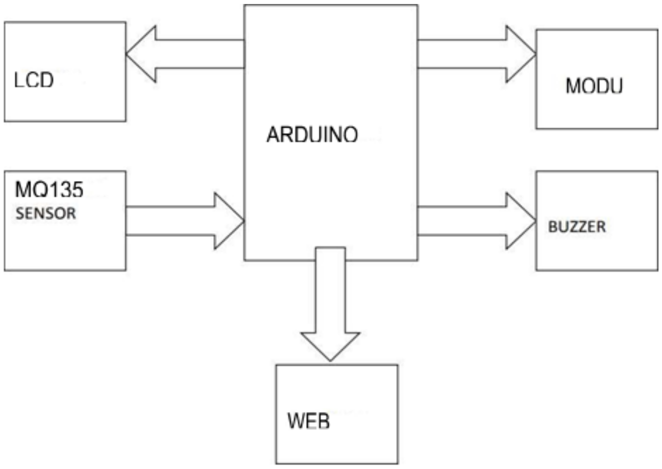


Fig. 1

The air pollution monitoring device developed in this project is based on Arduino Mega. The Arduino board connects with Thing Speak platform using Virtual Terminal port. As the cities usually have Wi-Fi hotspots at the most of the places, therefore the device are often easily installed near any hotspot for its operation. The Thing Speak may be a popular IOT platform which is straightforward to use and program. The sensors used for monitoring the air pollution is MQ-135 and MQ6 gas sensor. The sensor data is additionally displayed on a personality LCD interfaced within the monitoring IOT device.

The sending of data and sending it to the Thing Speak server using virtual terminal is managed by the Arduino Sketch. The Arduino sketch is written, compiled and loaded to the Arduino board using Arduino IDE. Things HTTP and REACT are 2 applications in Thing Speak which can be used to send triggers to a mobile device with the help of If This Then That (IFTTT) Application. The IFTTT is an easy to use application that can be installed on mobiles to send custom notifications by SMS, E-mail or Call. On mapping Thing Speak with IFTTT, we are able to get a notification every time the air quality level is high, or gas leakage is detected. Thing Speak is bundled along with MATLAB software to extend applications. Data is represented by a graph with Air Quality in Y-Axis vs Time in X-Axis. This can be used to track and monitor air quality and gas leakage at any specific time and can also be used to predict and analyze the air quality in the future. Predictive analysis is the next step. Things View is a mobile application which can be used to track the air quality by a single touch.

B. *REQUIREMENTS*

Hardware Requirement

- MQ135 Gas sensor
- MQ6 LPG sensor
- Arduino Mega
- Virtual Terminal Port
- 16X2 LCD
- Breadboard
- DHT sensor
- LED.

Software Requirement

- ARDUINO 1.6.13 software
- Embedded C language, Python and React native
- ThingSpeak
- If This Then That (IFTTT) Application

MODULE’ DESCRIPTION

MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and a few other gases, so it's ideal gas sensor for our Air Quality Monitoring System. The MQ6 sensor for Gas Leakage system which prints the units on the LCD display. On connecting these sensors to Arduino, the air quality and LPG gas are detected and get the Pollution level in PPM (parts per million). The data from the MQ135 gas sensor is sent to ThingSpeak via the virtual terminal port. Once this data is mapped on ThingSpeak, we can move on to the next part which is notification. ThingsHTTP and REACT are two applications supported by ThingSpeak to trigger a notification every time the level exceeds. IFFFT is used to send a notification to your mobile device. When sensor value exceeds the limit of 1000 PPM, then it starts causing headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases. Whenever the value of MQ6 sensor exceeds 50 units, then the buzzer will start beeping, LED turns on and the LCD will display the LPG units every second.

C. MODULE INTEGRATION

Integration of the LCD and Arduino:

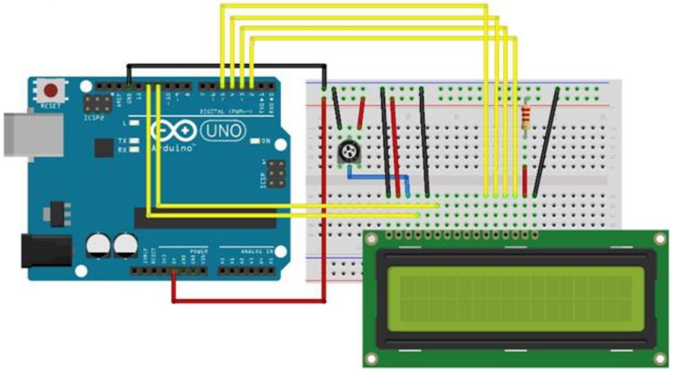
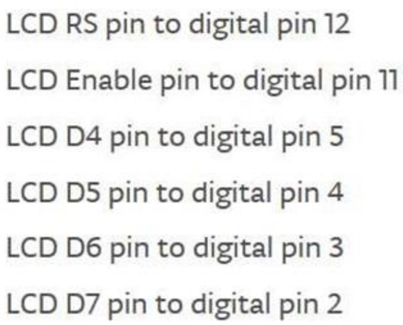


Fig. 2



LCD RS pin to digital pin 12

LCD Enable pin to digital pin 11

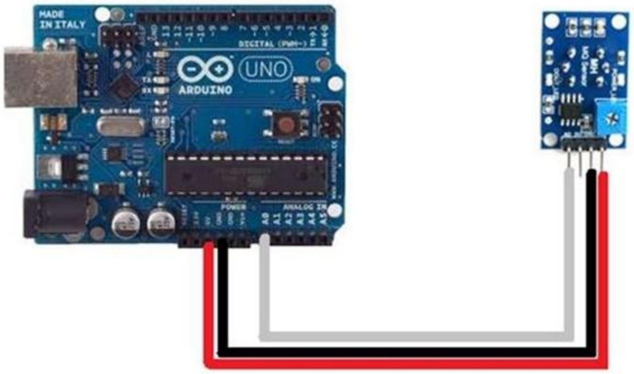
LCD D4 pin to digital pin 5

LCD D5 pin to digital pin 4

LCD D6 pin to digital pin 3

LCD D7 pin to digital pin 2

D. Integration of LCD and MQ135 Gas Sensor:



Sensor A0 pin to Analog pin 0
Sensor D0 pin to Digital pin 0
Buzzer Vin pin to Digital pin 9

Fig. 3

E. Circuit on Proteus:

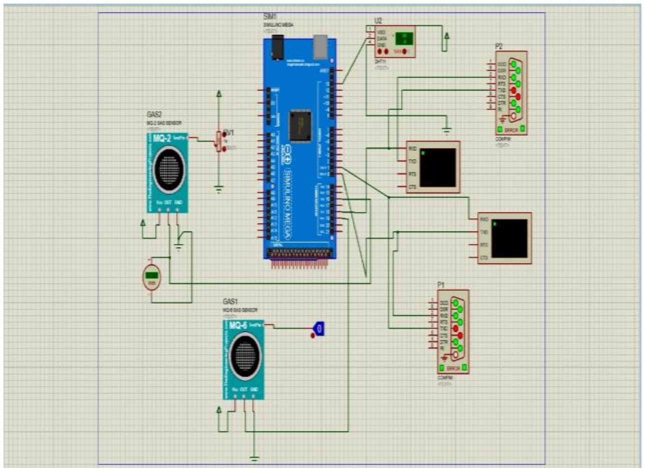
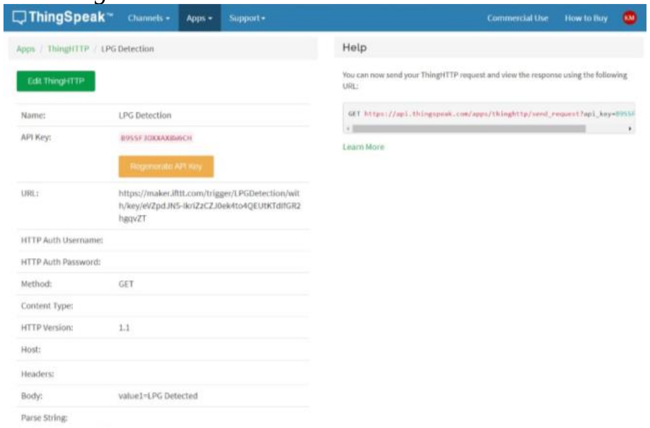


Fig. 4

F. ThingsHTTP



Apps / ThingHTTP / LPG Detection

Name:	LPG Detection
API Key:	B9SSF3OXXAX8W6CH
URL:	https://maker.ifttt.com/trigger/LPGDetection/with/key/eVZpdJN5-ikriZzCZJ0ek4to4QEUtKTdIfGR2hgqvZT
HTTP Auth Username:	
HTTP Auth Password:	
Method:	GET
Content Type:	
HTTP Version:	1.1
Host:	
Headers:	
Body:	value1=LPG Detected
Parse String:	

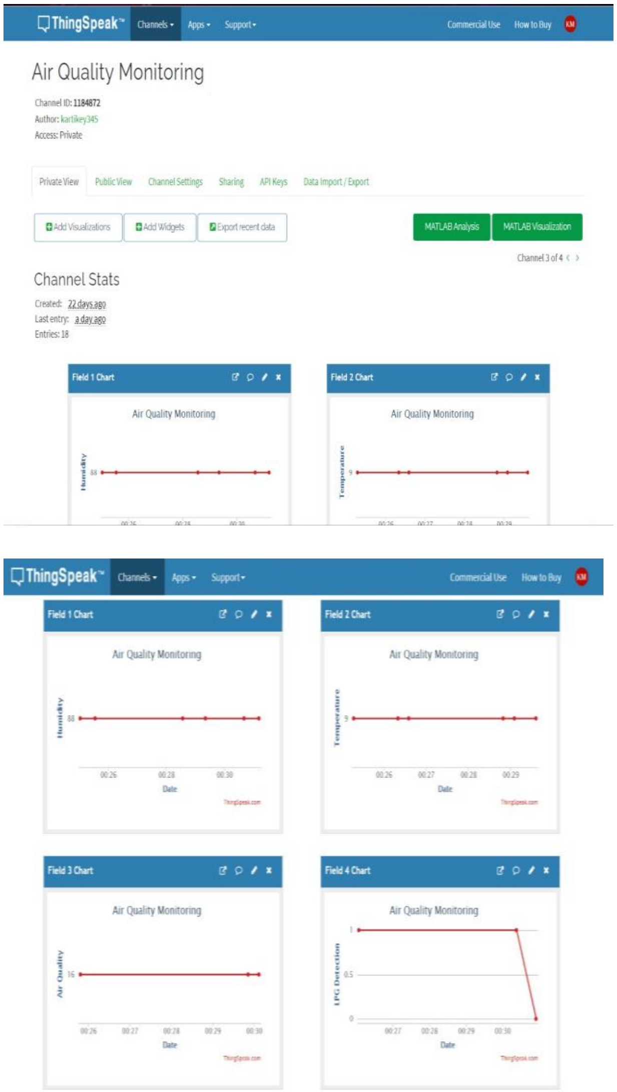
Help

You can now send your ThingHTTP request and view the response using the following URL:

```
GET https://api.thingspeak.com/apps/thinghttp/send_request?api_key=B9SSF
```

Fig. 5

G. Thingspeak



Air Quality Monitoring

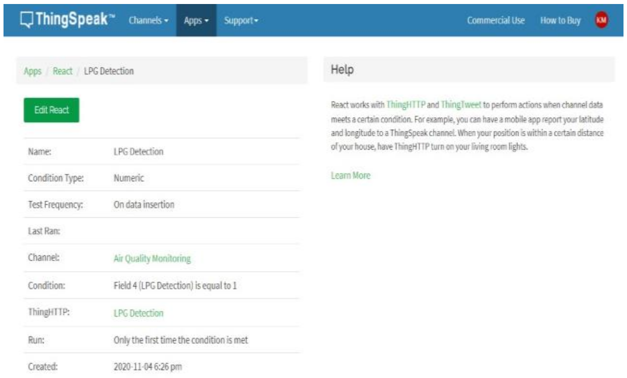
Channel ID: 1184872
Author: kartikey345
Access: Private

Channel Stats

Created: 22 days ago
Last entry: a day ago
Entries: 18

Fig 6

H. React use



Apps / React / LPG Detection

Name:	LPG Detection
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	
Channel:	Air Quality Monitoring
Condition:	Field 4 (LPG Detection) is equal to 1
ThingHTTP:	LPG Detection
Run:	Only the first time the condition is met
Created:	2020-11-04 6:26 pm

Help

React works with ThingHTTP and ThingTweet to perform actions when channel data meets a certain condition. For example, you can have a mobile app report your latitude and longitude to a ThingSpeak channel. When your position is within a certain distance of your house, have ThingHTTP turn on your living room lights.

Fig. 7

V. IMPLEMENTATION

A. Arduino Mega:

Arduino Mega is a microcontroller board. It has digital and analog I/O pins. It has 54 digital I/O pins, 16 analog I/O pins. We program it with Arduino IDE via a type B USB cable. We use Arduino in our project to load programs. We use it to mount sensors on it. Those sensors then fetch data and processing is done. So, it won't be incorrect if we say that Arduino Uno is the brain of our application as it processes the data from the sensors. We have used C language to code.

B. Virtual Terminal

Gives access to Wi-Fi networks. It is used to establish Wi-Fi connection between Arduino and platform. It makes simple TCP/IP connections using Hayes-style commands.

C. 16X2 LCD

Liquid Crystal Display is a screen used to display. It means it can display 16 characters per line and there are 2 such lines. Each character is displayed using a 5x7 pixel matrix. Command register is used to store commands given to display and data register stores the data that has to be displayed.

D. MQ135 Gas Sensor

Detects wide range of gases such as NH3, NOx, Alcohol, Benzene, Smoke, CO2. This module is powered with a voltage source and we can see the LED to glow. The LED remains turned off if no gas detection is done. This sensor also measures PPM.

E. MQ6 Gas Sensor

This sensor is particularly used to detect LPG concentration in air. It can detect concentration in the range 200-10000 ppm.

F. *DHT Sensor*

This sensor particularly come with humidity sensor with thermistor which can measure the temperature.

G. *Web Page*

The web page is used to display the concluded result. It displays the channel data obtained from sensor to ThingSpeak. IFTTT applets are created to send email and notifications.

We connect MQ135 and MQ6 sensors to Arduino. The sensor sends the data to ThingSpeak via ESP Wi-Fi module. ThingsHTTP and REACT are two applications supported by ThingSpeak to trigger a notification every time the level exceeds. ThingSpeak is bundled with software. IFFFT sends notifications to user's mobile. It informs what was the problem diagnosed, when was it diagnosed, average air quality for the past day, current air quality and the quality level. The buzzer beeps every time the value read by MQ6 exceeds 50 units. This turns on the LED and units of LPG are displayed every second

VI. CODE

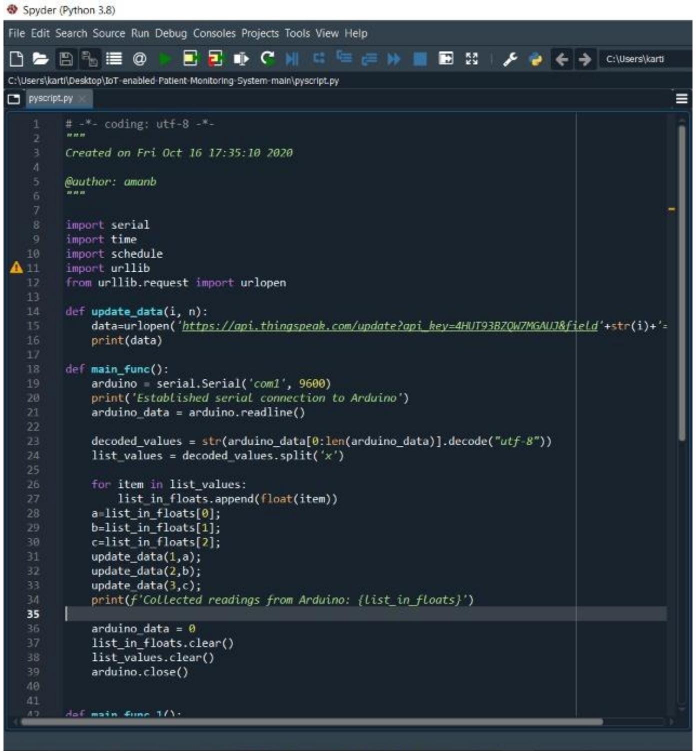
A. *ARDUINO*

```
#include <LiquidCrystal.h> // includes the LiquidCrystal
Library LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an
LC object. Parameters: (rs, enable, d4, d5, d6, d7)

int gas_in = A1; int buzz = 8;
int val;
void setup() {

// put your setup code here, to run once: lcd.begin(16,
2);
pinMode(buzz,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly: val
= analog Read(gas_in); lcd.print(val);

if     (val  >  15) tone(buzz,0.0000001);

//digitalWrite(buzz, HIGH); delay(2000);
lcd.clear(); digitalWrite(buzz, LOW);
}
```

Arduino Code (ESP8266)

```
#include <ESP8266WiFi.h>;

#include <WiFiClient.h>;

#include <ThingSpeak.h>;

const char* ssid = "Hotspot ABCD"; //Your Network
SSID

const char* password = "aaaaaaaa"; //Your Network
Password

int val;

int Gas_in = A0; //LDR Pin Connected at A0 Pin

WiFiClient client;

unsigned long myChannelNumber = 894553; //Your
Channel Number (Without Brackets)


const      char      *      myWriteAPIKey      =
"6G9KUN05KGLUEGBF"; //Your Write API Key


void setup()

{

Serial.begin(9600);

delay(10);

// Connect to WiFi network

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED)

{

delay(500); Serial.print("."
);

}
ThingSpeak.begin(client);

}

void loop()

{

val = analogRead(Gas_in); //Read Analog values and
Store in val variable
//int val1=80;

//Serial.print(val1); //Print on Serial Monitor

delay(1000);

ThingSpeak.writeField(myChannelNumber,1,val,
myWriteAPIKey);
ThingSpeak.writeField(895806,1,val,"I2RAGQTTV854H
CQZ");
//Update in ThingSpeak

delay(100);

}
```

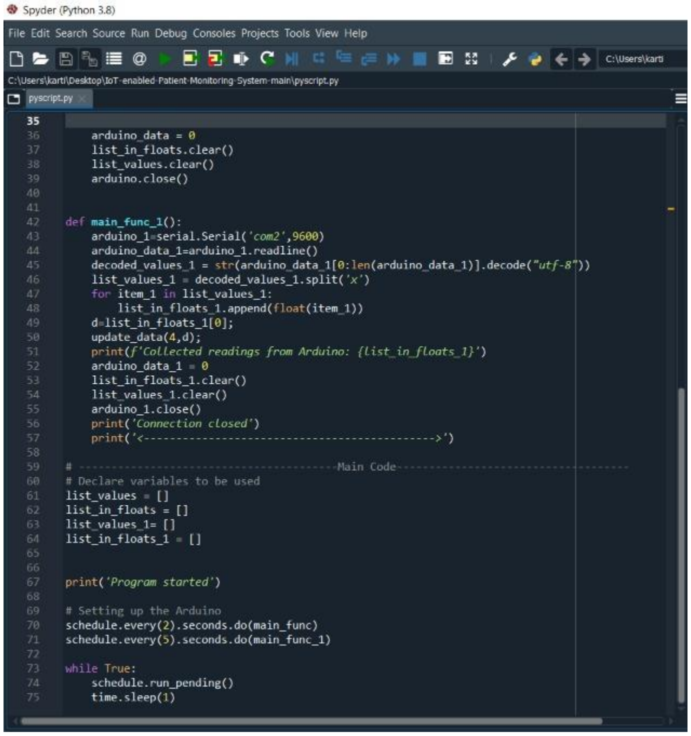
B. *Python script for connection Proteus to Thingspeak*



Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\karti
C:\Users\karti\Desktop\IoT-enabled-Patient-Monitoring-System-main\pyscript.py
pyscript.py

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 16 17:35:10 2020

@author: amanb
"""

import serial
import time
import schedule
import urllib
from urllib.request import urlopen

def update_data(i, n):
    data=urlopen('https://api.thingspeak.com/update?api_key=4HUT93BZQW7MGAUJ&field'+str(i)+'=
    print(data)

def main_func():
    arduino = serial.Serial('com1', 9600)
    print('Established serial connection to Arduino')
    arduino_data = arduino.readline()

    decoded_values = str(arduino_data[0:len(arduino_data)].decode("utf-8"))
    list_values = decoded_values.split('x')

    for item in list_values:
        list_in_floats.append(float(item))
    a=list_in_floats[0];
    b=list_in_floats[1];
    c=list_in_floats[2];
    update_data(1,a);
    update_data(2,b);
    update_data(3,c);
    print(f'Collected readings from Arduino: {list_in_floats}')

    arduino_data = 0
    list_in_floats.clear()
    list_values.clear()
    arduino.close()


def main_func_1():
```

Fig. 8

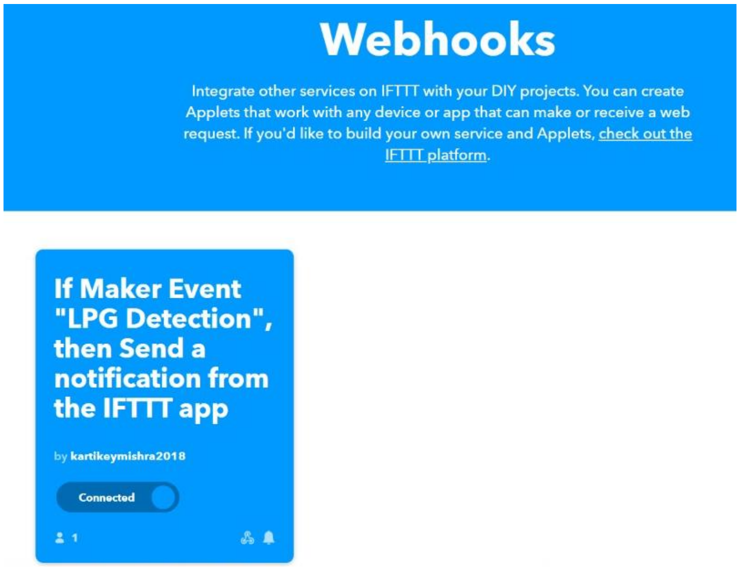


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\karti
C:\Users\karti\Desktop\IoT-enabled-Patient-Monitoring-System-main\pyscript.py
pyscript.py

```python

    arduino_data = 0
    list_in_floats.clear()
    list_values.clear()
    arduino.close()


def main_func_1():
    arduino_1=serial.Serial('com2',9600)
    arduino_data_1=arduino_1.readline()
    decoded_values_1 = str(arduino_data_1[0:len(arduino_data_1)].decode("utf-8"))
    list_values_1 = decoded_values_1.split('x')
    for item_1 in list_values_1:
        list_in_floats_1.append(float(item_1))
    d=list_in_floats_1[0];
    update_data(4,d);
    print(f'Collected readings from Arduino: {list_in_floats_1}')
    arduino_data_1 = 0
    list_in_floats_1.clear()
    list_values_1.clear()
    arduino_1.close()
    print('Connection closed')
    print('<----------------------------------------->')

# ----------------------------------------Main Code----------------------------------------
# Declare variables to be used
list_values = []
list_in_floats = []
list_values_1= []
list_in_floats_1 = []


print('Program started')

# Setting up the Arduino
schedule.every(2).seconds.do(main_func)
schedule.every(5).seconds.do(main_func_1)

while True:
    schedule.run_pending()
    time.sleep(1)
```

Fig. 9

C. *IFTTT*



Webhooks

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

If Maker Event "LPG Detection", then Send a notification from the IFTTT app

by kartikeymishra2018

Fig. 10

VII. CONCLUSION AND FUTURE WORK

This system to monitor the air pollution of environment using Arduino microcontroller, IOT Technology is proposed to improve quality of air. Gas Leakage detection system can be used to alert and notify the user on his mobile device thus saving lives. With the utilization of IOT technology enhances the method of monitoring various aspects of environment like air quality monitoring issue proposed during this paper.

Here, we use the MQ135 and MQ6gas sensor which gives the sense air quality and Arduino which is the heart of this project. Virtual Terminal connects the whole process to internet and LCD is used for the visual output. Things Speak is used to monitor, analyze and display data while IFTTT application is used to notify and alert the user.

The recent air pollution incidents have proved how harmful it is breathing bad air. Delhi can be used as an apt example of air pollution and its ill effects on human health and lifestyle. It is now too late to make any changes to the city which is also the capital of India. Life expectancy decreases when you breathe bad air. We can use Delhi as an example and learn from our mistakes by monitoring and tracking air quality and thus make effective changes.

We can further use this technology to place air quality sensors around the city and map air quality in the city and create a website so users can track the pollution in their respective regions.

REFERENCES

[1] Varma A., Prabhakar, S. and Jayavel K. Gas Leakage Detection and Smart Alerting and prediction using IoT.

[2] Keshamoni K, Hemanth S "Smart Gas Level Monitoring, Booking & Gas Leakage Detector over IoT"

[3] Ying-Tsung Lee, Wei-Hsuan Hsiao, Chin-Meng Huang and Seng-Cho T. Chou "An Integrated Cloud-Based Smart Home Management System with Community Hierarchy"

[4] Jetendra Joshi, Vishal Rajapriya, S R Rahul, Pranith Kumar, Siddhanth Polepally, Rohit Samineni and D G Kamal Tej "Performance Enhancement and IoT Based Monitoring for Smart Home"

[5] Asmi, A., Putra, J. C.P., &Rahman, I. B. A. "A Study of Indoor Air Quality of Public Toilet in University's Building"

[6] QasemAbuAl-Haija HasanAl-Qadeeb Abdulmohsen Al- Lwaimi "Monitoring of AIR Quality in King Faisal University Using a Microcontroller and WSN"