AIR QUALITY MONITORING

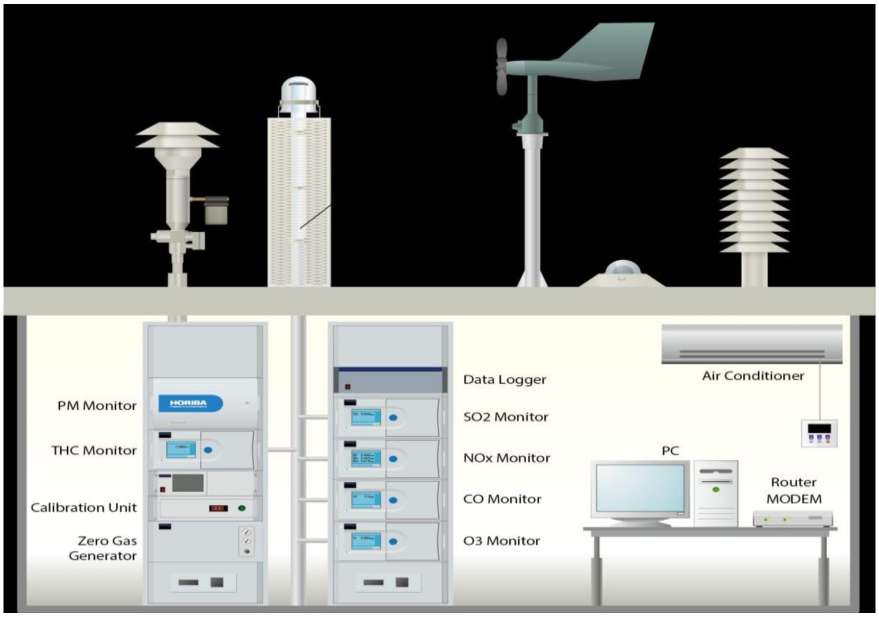
USING IOT

ABISHEK.T
au812921106002
abishekmusiri203@gmail.com

INTRODUCTION:

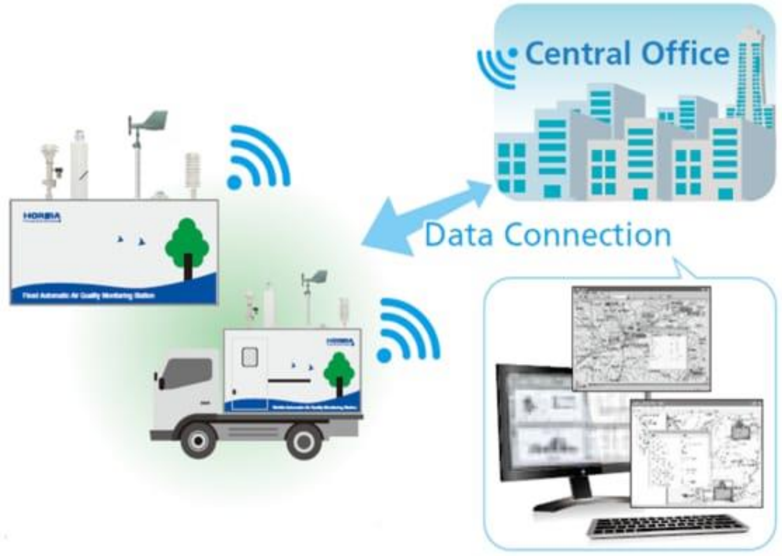
●Air Quality Monitoring Networks allow the measurement, operation and predictive analysis of the evolution of air pollution in different areas (urban areas, industrial areas, special nature conservation areas, etc.) Some stations are equipped with meteorological sensors and/or noise level meters to measure noise levels.

BLOCK DIAGRAM:



AQMS, the doctor for "Human Health"

●An Air Quality Monitoring Station (AQMS) is a system that measures metrological parameters such as wind speed, wind direction, rainfall, radiation, temperature, barometric pressure and ambient parameters. The AQMS also integrates a series of ambient analyzers to monitor the concentration of air pollutants (such as SO2, NOx, CO, O3, THC, PM, etc.), continuously. HORIBA also provides mobile monitoring stations that can be used to monitor ambient conditions at multiple sites.



●HORIBA has more than 50 years experience providing ambient monitoring solutions, recognized around the world. HORIBA has supplied over 15,000 units with the major share in many regions. The monitoring station is tailor-made according to the customer's request. HORIBA can provide several types of stations, calibration equipment and more to meet your challenging monitoring requirements.

●The measured data can be remotely monitored and exported in various formats to the local central authorities. The data can be published via the Internet for easy public access to raise awareness on current air pollution levels. This way, the public can prevent outdoor activities and reduce health impacts during heavy polluted days.

PROGRAM:

```
from tkinter import *
```

```
import requests

from bs4 import BeautifulSoup


# link for extract html data


def getdata(url):

    r = requests.get(url)

    return r.text



def airinfo():

    soup = BeautifulSoup(htmldata, 'html.parser')

    res_data = soup.find(class_="DonutChart--innerValue--2rO41 AirQuality--extendedDialText--2AsJa").text
    air_data = soup.find_all(class_="DonutChart--innerValue--2rO41 AirQuality--pollutantDialText--3Y7DJ")
    air_data=[data.text for data in air_data]
    ar.set(res_data)
    o3.set(air_data[0])
    no2.set(air_data[1])
    so2.set(air_data[2])
    pm.set(air_data[3])
    pml.set(air_data[4])
    co.set(air_data[5])
    res = int(res_data)
    if res <= 50:
        remark = "Good"
        impact = "Minimal impact"
    elif res <= 100 and res > 51:
        remark = "Satisfactory"
        impact = "Minor breathing discomfort to sensitive people"
    elif res <= 200 and res >= 101:
        remark = "Moderate"
        impact = "Breathing discomfort to the people with lungs, asthma and heart diseases"
    elif res <= 400 and res >= 201:
        remark = "Very Poor"
        impact = "Breathing discomfort to most people on prolonged exposure"
    elif res <= 500 and res >= 401:
        remark = "Severe"

        impact = "Affects healthy people and seriously impacts those with existing diseases"

    res_remark.set(remark)

    res_imp.set(impact)


# object of tkinter
# and background set to grey

master = Tk()

master.configure(bg='light grey')

# Variable Classes in tkinter

air_data = StringVar()

ar = StringVar()

o3 = StringVar()
```

```
no2 = StringVar()

so2 = StringVar()

pm = StringVar()

pml = StringVar()

co = StringVar()

res_remark = StringVar()

res_imp = StringVar()


# Creating label for each information
# name using widget Label

Label(master, text="Air Quality : ",

        bg="light grey").grid(row=0, sticky=W)

Label(master, text="O3 (µg/m3) :",
        bg="light grey").grid(row=1, sticky=W)
Label(master, text="NO2 (µg/m3) :",
        bg="light grey").grid(row=2, sticky=W)
Label(master, text="SO2 (µg/m3) :",
        bg="light grey").grid(row=3, sticky=W)
Label(master, text="PM2.5 (µg/m3) :",
        bg="light grey").grid(row=4, sticky=W)
Label(master, text="PM10 (µg/m3) :",
        bg="light grey").grid(row=5, sticky=W)
Label(master, text="CO (µg/m3) :",
        bg="light grey").grid(row=6, sticky=W)
```

```
Label(master, text="Remark :",

            bg="light grey").grid(row=7, sticky=W)

Label(master, text="Possible Health Impacts :",

            bg="light grey").grid(row=8, sticky=W)



# Creating label for class variable
# name using widget Entry

Label(master, text="", textvariable=ar,

            bg="light grey").grid(

        row=0, column=1, sticky=W)

Label(master, text="", textvariable=o3,

            bg="light grey").grid(

        row=1, column=1, sticky=W)
```

```
Label(master, text="", textvariable=no2,
      bg="light grey").grid(
    row=2, column=1, sticky=W)
Label(master, text="", textvariable=so2,
      bg="light grey").grid(
    row=3, column=1, sticky=W)
Label(master, text="", textvariable=pm,
      bg="light grey").grid(
    row=4, column=1, sticky=W)
Label(master, text="", textvariable=pml,
      bg="light grey").grid(
    row=5, column=1, sticky=W)
```

```
Label(master, text="", textvariable=co,
            bg="light grey").grid(
        row=6, column=1, sticky=W)
Label(master, text="", textvariable=res_remark,
            bg="light grey").grid(row=7, column=1, sticky=W)
Label(master, text="", textvariable=res_imp,
            bg="light grey").grid(row=8, column=1, sticky=W)

# creating a button using the widget
b = Button(master, text="Check",
                      command=airinfo, bg="Blue")
b.grid(row=0, column=2, columnspan=2,
              rowspan=2, padx=5, pady=5,)
```

```
mainloop()
```