(urban areas, industrial areas, special nature conservaon areas, etc.) Some

operaon and predicve analysis of the evoluon of air polluon in dierent areas

Air Quality Monitoring Networks allow the measurement,

⚫

Phase 3

INTRODUCTION:

**PROJECT TITLE:**

**AIR QUALITY**



**MONITORING**

**USING IOT**

**COLLEGE NAME: PERI INSTITUTE OF  
TECHNOLOGY**

**DOMAIN: INTERNET OF THINGS**

**PHASE 3**

**SUBMITTED BY:**

**MALIN.S - 411521106029**

condions at mulple sites.

provides mobile monitoring staons that can be used to monitor ambient

pollutants (such as SO2, NOx, CO, O3, THC, PM, etc.), connuously. HORIBA also

integrates a series of ambient analyzers to monitor the concentraon of air

temperature, barometric pressure and ambient parameters. The AQMS also

metrological parameters such as wind speed, wind direcon, rainfall, radiaon,

An Air Quality Monitoring Staon (AQMS) is a system that measures

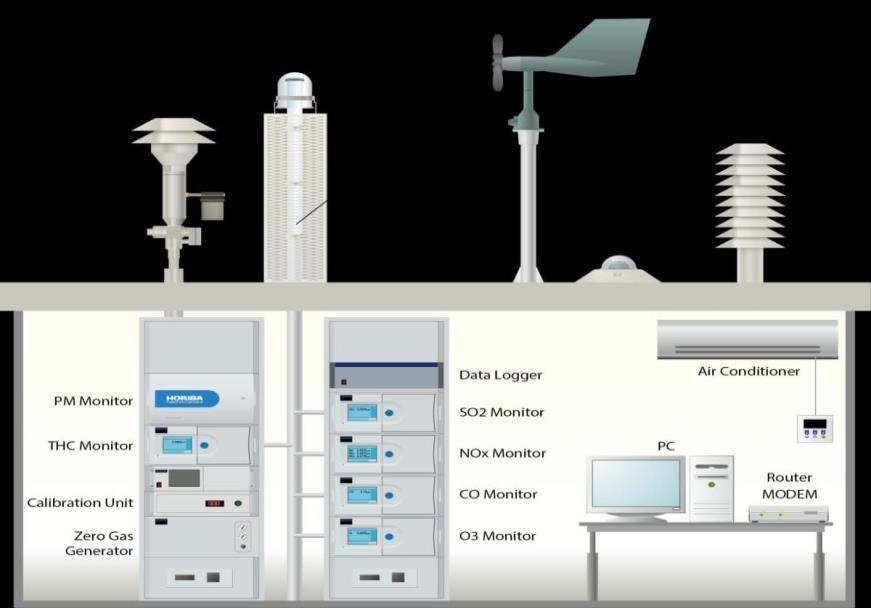
⚫

AQMS, the doctor for "Human Health"

BLOCK DIAGRAM:

measure noise levels.

staons are equipped with meteorological sensors and/or noise level meters to



# link for extract html data

from bs4 import BeaufulSoup

import requests

from tkinter import \*

PROGRAM:

polluted days.

the public can prevent outdoor acvies and reduce health impacts during heavy

for easy public access to raise awareness on current air polluon levels. This way,

formats to the local central authories. The data can be published via the Internet

The measured data can be remotely monitored and exported in various

⚫

monitoring requirements.

types of staons, calibraon equipment and more to meet your challenging

tailor-made according to the customer's request. HORIBA can provide several

15,000 units with the major share in many regions. The monitoring staon is

monitoring soluons, recognized around the world. HORIBA has supplied over

HORIBA has more than 50 years experience providing ambient

⚫



so2.set(air\_data[2])

no2.set(air\_data[1])

o3.set(air\_data[0])

ar.set(res\_data)

air\_data=[data.text for data in air\_data]

pollutantDialText--3Y7DJ")

air\_data = soup.nd\_all(class\_="DonutChart--innerValue--2rO41 AirQuality--

extendedDialText--2AsJa").text

res\_data = soup.nd(class\_="DonutChart--innerValue--2rO41 AirQuality--

soup = BeaufulSoup(htmldata, 'html.parser')

def airinfo():

return r.text

r = requests.get(url)

def getdata(url):

elif res <= 400 and res >= 201:

diseases"

impact = "Breathing discomfort to the people with lungs, asthma and heart

remark = "Moderate"

elif res <= 200 and res >= 101:

impact = "Minor breathing discomfort to sensive people"

remark = "Sasfactory"

elif res <= 100 and res > 51:

impact = "Minimal impact"

remark = "Good"

if res <= 50:

res = int(res\_data)

co.set(air\_data[5])

pml.set(air\_data[4])

pm.set(air\_data[3])

air\_data = StringVar()

# Variable Classes in tkinter

master.congure(bg='light grey')

master = Tk()

# and background set to grey

# object of tkinter

res\_imp.set(impact)

res\_remark.set(remark)

diseases"

impact = "Aects healthy people and seriously impacts those with exisng

remark = "Severe"

elif res <= 500 and res >= 401:

impact = "Breathing discomfort to most people on prolonged exposure"

remark = "Very Poor"

bg="light grey").grid(row=1, scky=W)

Label(master, text="O3 (μg/m3) :",

bg="light grey").grid(row=0, scky=W)

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

# name using widget Label

# Creang label for each informaon

res\_imp = StringVar()

res\_remark = StringVar()

co = StringVar()

pml = StringVar()

pm = StringVar()

so2 = StringVar()

no2 = StringVar()

o3 = StringVar()

ar = StringVar()

bg="light grey").grid(row=8, scky=W)

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

bg="light grey").grid(row=7, scky=W)

bg="light grey").grid(row=6, scky=W) Label(master, text="Remark :",

Label(master, text="CO (μg/m3) :",

bg="light grey").grid(row=5, scky=W)

Label(master, text="PM10 (μg/m3) :",

bg="light grey").grid(row=4, scky=W)

Label(master, text="PM2.5 (μg/m3) :",

bg="light grey").grid(row=3, scky=W)

Label(master, text="SO2 (μg/m3) :",

bg="light grey").grid(row=2, scky=W)

Label(master, text="NO2 (μg/m3) :",

bg="light grey").grid(

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

row=3, column=1, scky=W)

bg="light grey").grid(

Label(master, text="", textvariable=so2,

row=2, column=1, scky=W)

bg="light grey").grid(

row=1, column=1, scky=W) Label(master, text="", textvariable=no2,

bg="light grey").grid(

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

row=0, column=1, scky=W)

bg="light grey").grid(

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

# name using widget Entry

# Creang label for class variable

command=airinfo, bg="Blue")

b = Buon(master, text="Check",

# creang a buon using the widget

bg="light grey").grid(row=8, column=1, scky=W)

Label(master, text="", textvariable=res\_imp,

bg="light grey").grid(row=7, column=1, scky=W)

Label(master, text="", textvariable=res\_remark,

row=6, column=1, scky=W)

bg="light grey").grid(

row=5, column=1, scky=W) Label(master, text="", textvariable=co,

bg="light grey").grid(

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

row=4, column=1, scky=W)

mainloop()

rowspan=2, padx=5, pady=5,)

b.grid(row=0, column=2, columnspan=2,