# AIR QUALITY MONITIORING SYSTEM

# FLOW DIAGRAM

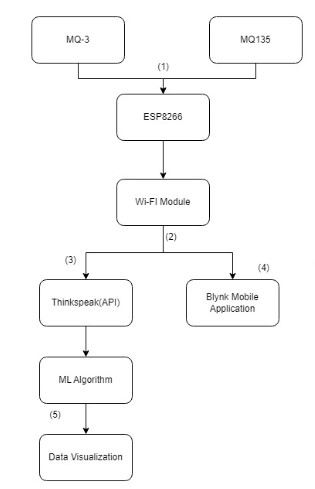


Figure 1: Flow diagram.

1. Input from MQ3, MQ135 is taken in the format of analog and the send to the ESP8266.
2. The Wi-Fi module conneccted to the nearby Wi-Fi sends the data to ThingSpeak and Blynk IoT platform.
3. The raw data from the sensors is sent to the ThingSpeak.
4. The calculated ppm values are sent to the Blynk IoT mobile application.
5. The data collected on ThingSpeak is exported as csv file and processed through the ML algorithm for

data visualization.

# CIRCUIT DESIGN

The sensors selected for the system were the MQ 135 gas sensor for volatile organic compounds (VOCs) and the MQ 3 gas sensor for alcohol. The sensors were calibrated by exposing them to known levels of pollutants and adjusting the readings to match the expected values. The hardware design also consisted of an ESP8266, Wi-Fi module, MQ 135, MQ 3 gas sensors, an Arduino microcontroller, and a power source.

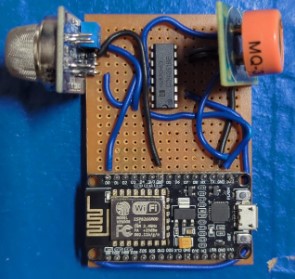


Figure 2: Circuit Design

The sensors were connected to the microcontroller, and the ESP8266 was used to establish a Wi-Fi connection for data transmission. The software development involved programming the microcontroller using Arduino IDE.

The code was designed to read data from the sensors, process it, and transmit it wirelessly to a cloud-based server. The system collects and stores data from the sensors at regular intervals. The collected data were analyzed using machine learning algorithms to predict future pollution levels.

# WORING ALGORITHM

The below algorithm is followed to collect data from the sensors.

1. Define the Blynk credentials, WiFi credentials, and other variables requiredfor the code.
2. Setup the serial communication and the Blynk connection using Blynk.begin(). 3. Set up the timer to run a function to send data to ThingSpeak every second.
3. Connect to the WiFi network using WiFi.begin() and wait until the connectionis established.
4. Define the changeMUX function and set the MUX\_A pin as output.
5. In the loop, run the Blynk and timer functions, and read the sensor datafrom the analog pin A0.
6. Calculate the sensor value 1 (ppm (parts per million)) value for the sensordata using a formula.
7. Read the sensor data from A0 for a total of six times, and take the average ofthese readings to get the sensor value 0.
8. Change the MUX\_A pin to HIGH, and read the sensor data from A0 foranother six times, and take the average of these readings to get the sensor value

1.

1. Connect to ThingSpeak using the WiFiClient object.
2. Build the request string with the ThingSpeak API key and field values (sensorValue0 and sensorValue1) and send the GET request using the HTTPClient object.
3. Delay for a second before running the loop again.
4. Define the function to be called by the timer to send data to ThinkSpeak.
5. Change the MUX\_A pin to LOW and read the sensor data from A0.
6. Calculate the ppm value for the sensor data using a formula.
7. Change the MUX\_A pin to HIGH and read the sensor data from A0 for atotal of six times, and take the average of these readings to get the sensor value

2.

1. Write the sensor value 1 and sensor value 2 to virtual pins V 1 and V 2 respectively using Blynk.virtualWrite().