IT-478

Internet Of Things

IoT based Traffic Pollution Monitoring

Team-Members

Krishanu Konar-201401127

Kunal Suthar-201401131

Virendra Pratap-201401404

Abstract:

Vehicular pollution has a major role in degrading the environmental system. The level of traffic pollution has increased with time due to a lot of factors like the increase in population, increased vehicle use etc. which results in harmful effects on human beings by directly affecting health of population. In this project, we focus on traffic pollution monitoring. Every vehicle has a standard of emission of gases, but the difficulty occurs when the emission is beyond the standardized values. This emission from vehicles cannot be completely avoided, but it can be definitely controlled. The aim of our project will be to monitor the pollutants at traffic signals using the pollution detection circuit. This pollution detection circuit will consist of various sensors like gas sensor(CO), temperature sensor, humidity sensor etc. and are connected to a Microcontroller(Arduino) which gives us various environmental data. The data collected by the detection circuit is sent via a wifi module to a web-server, from where it is used for various analytical purposes which gives us qualitative data about pollution.

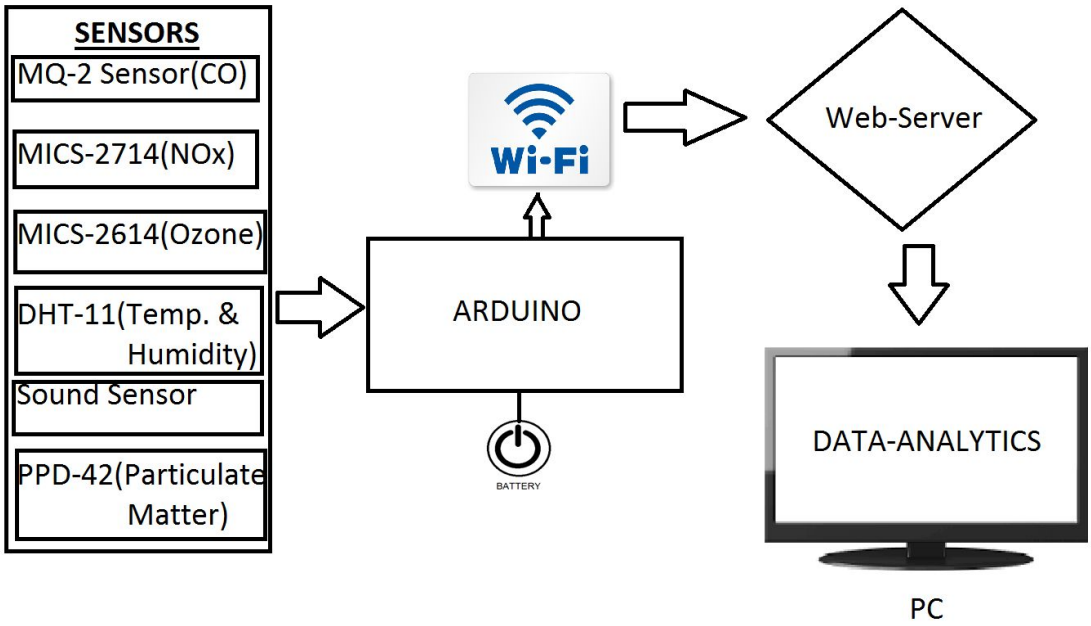
Requested components with its purpose in project:

- MQ7- mainly for Carbon Monoxide, also benzene & other aromatic compounds
- DHT-11: Temperature and Humidity Sensor
- NodeMCU ESP 8266 WiFi module for arduino
- 1 Arduino Mega
- Breadboard

Proposed solution:

We plan to install our pollution monitoring node at traffic lights/intersection points, parks, highways, society vicinity, as the traffic here tends to be high, because of the stopping vehicles(static vehicles emit more pollutants and particulate matter than the running ones). This system will routinely sense pollution values from time to time and upload it to the web server, from where the data can be accessed by a remote computer, which can run various analytics on the data, to produce meaningful insights like traffic peak time, heatmap of various gases and pollution level, etc. Having collected the pollution data, we would be aggregating the data, to predict the levels of pollution under specific conditions. This could be further used for prediction of pollution levels using Regression analysis.

Block diagram



Working:

- We start with creating a Pollution monitor circuit, which acts as a sensor node in our system. For measuring the Carbon Monoxide content in the atmosphere, we have used the MQ-7 sensor. It has 6 pins. It is divided into 2 groups, A and B(having 3 pins each). The middle pin of group A goes to the Vcc(5 volts). The other middle one of group B goes to the ground. The remaining 2 pins in each group are shorted with each other. The two shorted A wires go the positive end of the breadboard, and the other two shorted wires of B are used as the analog input for the arduino. The CO level is then extracted from the sensor using the analog pins of the arduino, and retrieved in the code using analog read. It returns values from 0 to 1023. Higher value suggests high levels of CO in the air.

- Now for transmitting the sensor value. We have used DHT-11 sensor for measuring the temperature and humidity levels. We imported the arduino library which would convert the analog input from the sensor to the actual temperature and humidity level. It has three pins Vin(for the voltage), Ground pin and the data pin from which the analog data is transferred to the analog pin of the arduino. The analog read is then performed to retrieve the data.

- From the values to the web server, we take the average value sampled at one second for 10 seconds, so as to not get any outliers in the temperature, humidity and the CO-level values. We transmit the aggregated averaged value of all the parameters from the arduino to the Wifi module via serial transmission, which further sends the data to the server.

- For transmitting the values to the server, we first connect the WiFi Module to a WiFi Network.

- ***Old Approach:*** *After connecting the module, we send the data in the form of packets, by establishing a TCP connection and then using the HTTP to send the data at the application level. The data is transmitted to the server. The data from the server is retrieved by another host(a laptop) from where onwards, the data analytics is performed. For setting up the ESP8266 WiFi Module, we power it*

through a microUSB cable(other option being giving it Vcc and ground through the pins). Before testing the Wifi module, we first need to install the latest AI-thinker firmware. We installed the firmware, using esp8266_flasher software. After installing the firmware, we test the module via the PuTTY software on the Serial Ports at a 9600 baud rate. We use the AT command for checking whether the module is working or not. If it is working, it returns OK on the serial port. We change the mode of the WiFi module using the command AT+CWMODE=1, which enables the module to connect to a WiFi network(WPA secured). Now the module is ready to transmit the data.

- **Current Approach:** In the previous approach, we were trying to interface the WiFi module with the arduino, and all the code and work was done on the arduino itself. However, we ran into several problems related to synchronization and sending the values to the server. In the current approach, we have divided our task in 2 parts: The data collection part is handled by arduino and the transmission part is handled by ***programming***[2] NodeMCU module using Arduino's IDE and its existing WiFi libraries. The connection and GET request is coded in the NodeMCU module.

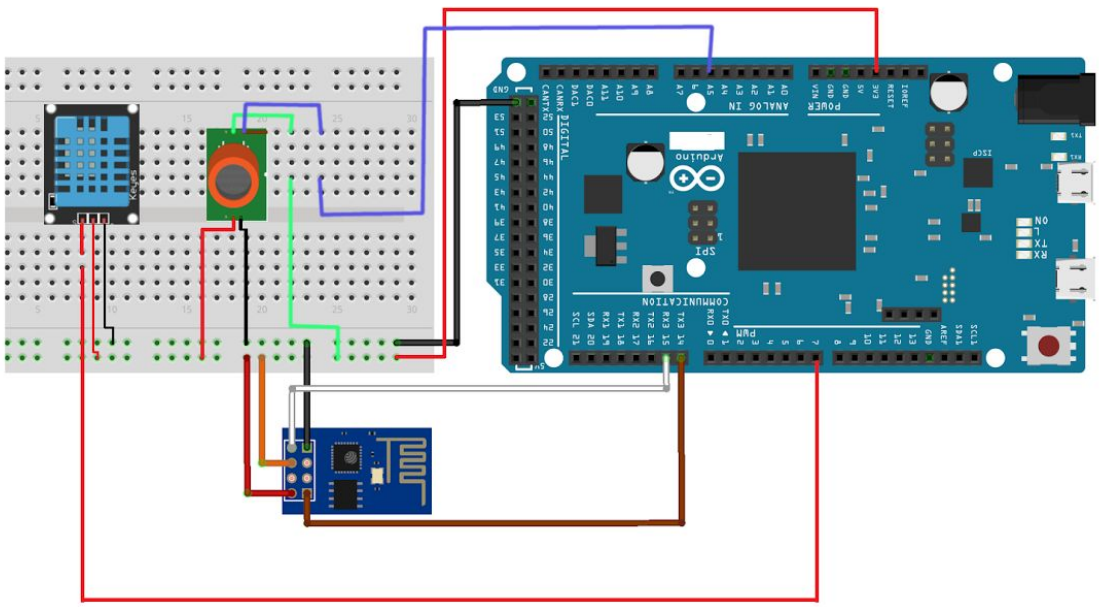
- Since NodeMCU doesn't have a very high processing capability, we perform most of our tasks on Arduino itself. We are taking the average of ten values for each sensor, and then the data is aggregated in the form of the concatenated strings. This string includes pollution (CO) levels, temperature and humidity. This data is transmitted using Serial Port Communication(Rx, Tx) of arduino and is then transmitted to the ESP8266 wifi module with the NodeMCU board and then to the google server by issuing a GET request using in-built http post libraries[3].

- On the server side, the running google script receives the data. The script appends a timestamp to this particular set of values, extracts the values of temperature and humidity and alters the CO pollution value considering the effect of temperature and humidity[4]. The processed temperature, humidity and pollution values is then written to our Data Spreadsheet. The GET request is issued at regular intervals of 10 sec.

- The data present on Google Drive is downloaded on the remote computer using Google Drive's API and data is extracted from the spreadsheet to form a pandas

dataframe(python library), which stores all the data in a matrix form and then this data can be accessed easily. The extracted data can now be used in the monitoring of the pollution levels of different locations and used in different analytical purposes. For eg., (as in our demonstration), we can plot a *time-CO levels* graph from data from a particular location to observe the peak-times of pollution. Once we have enough sensors on different locations, we can actually use all the data to obtain a heatmap of pollution values at any given time.

Circuit diagram:



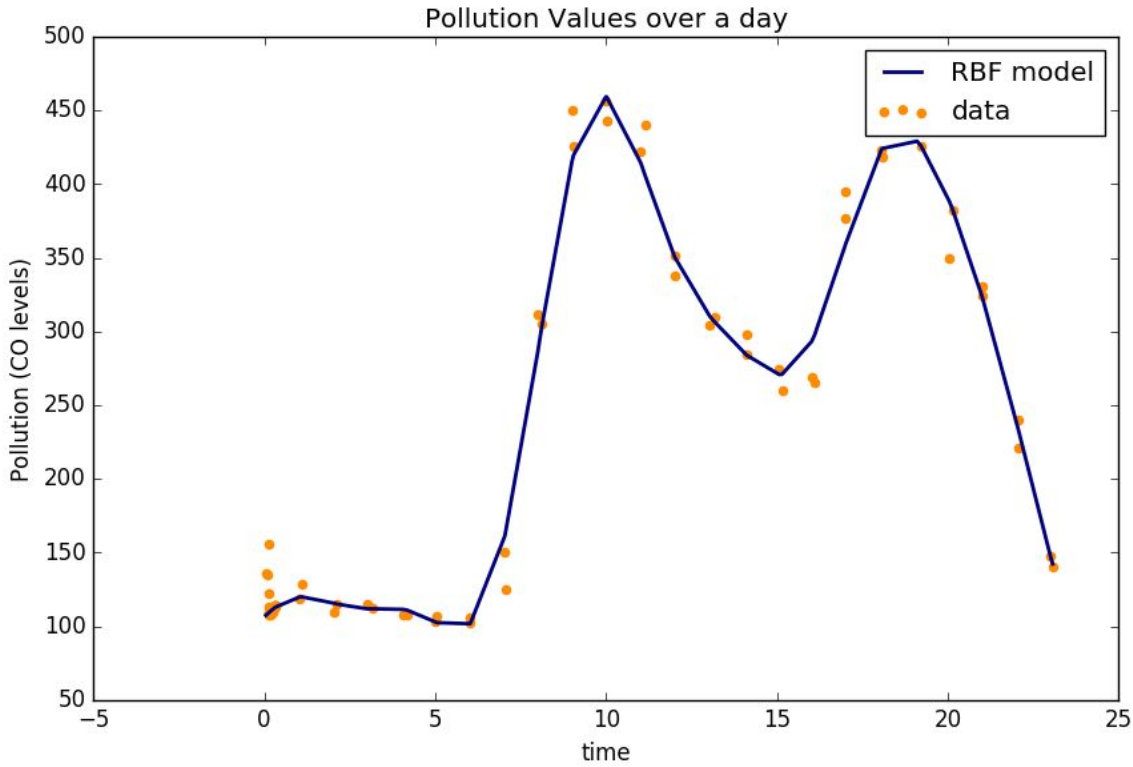
Limitation/Problems faced

The biggest challenge in this project is the modelling of environmental data. It is a non trivial and difficult task. A lot of data is required to come up with a realistic model that conforms to the real world scenario. We also faced difficulty in setting up the wifi module with arduino as the existing firmware wasn't supported with arduino. Interfacing the WiFi module with arduino also ran into many errors.

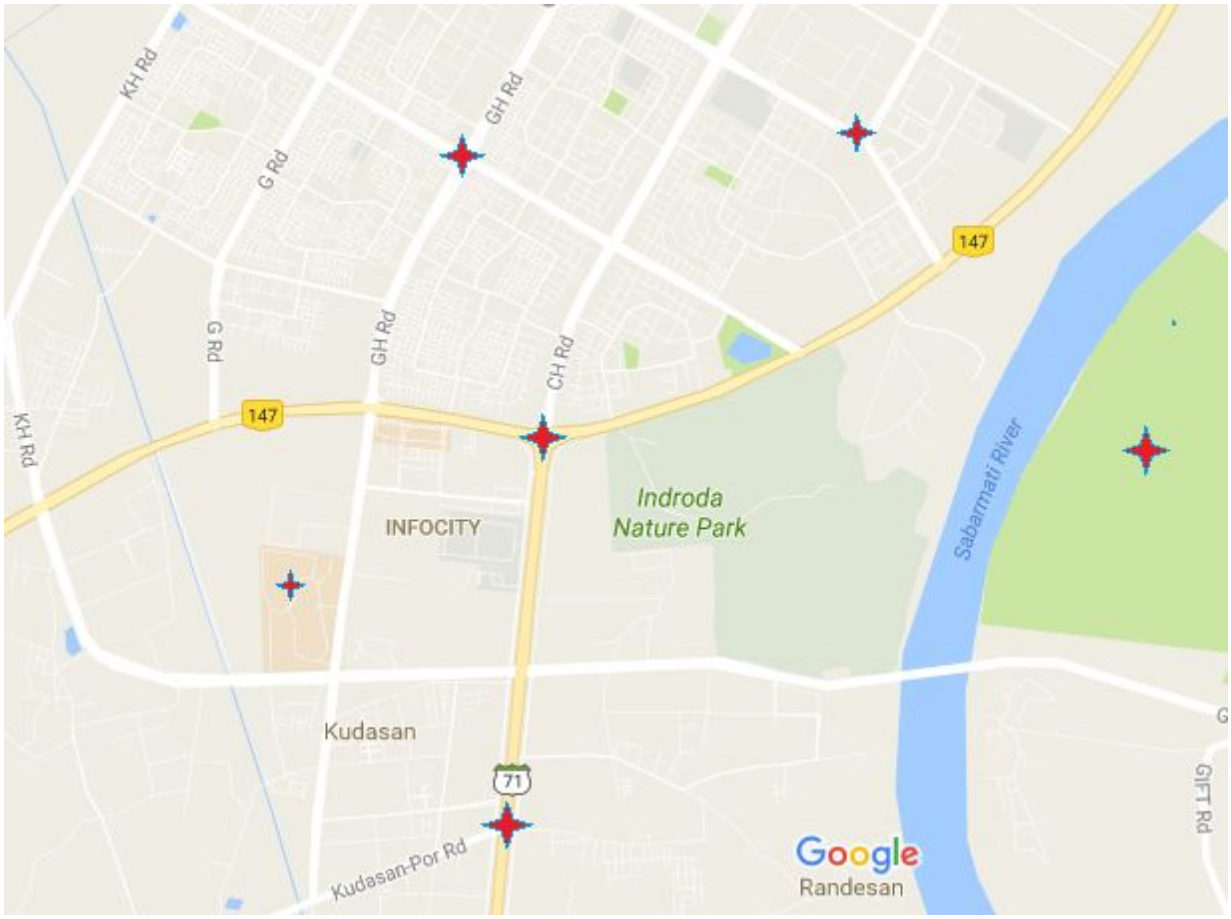
Results and Future Work

We are successfully transmitting the sensor node values (pollution levesl, temperature, humidity) to the Google SpreadSheet. However, because of shortage of time, we could not collect enough data, to perform any realistic analysis. So, we have used a few dummy data points to show our analysis approach.

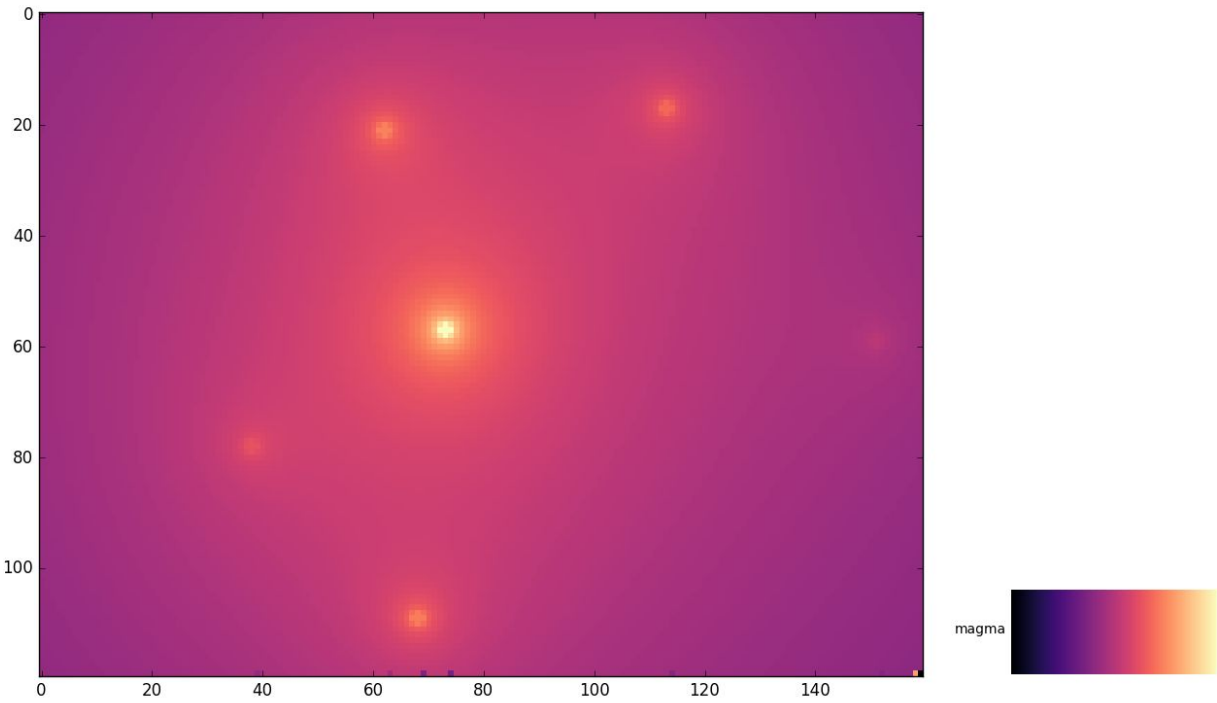
For predicting pollution levels, we can use regression analysis, for which the data-sets would be collected throughout the day at different times. For a given day, we plot the pollution values with respect to time. Then, we can run a support vector regression model with a non linear kernel to predict pollution values on that time, at any other day.



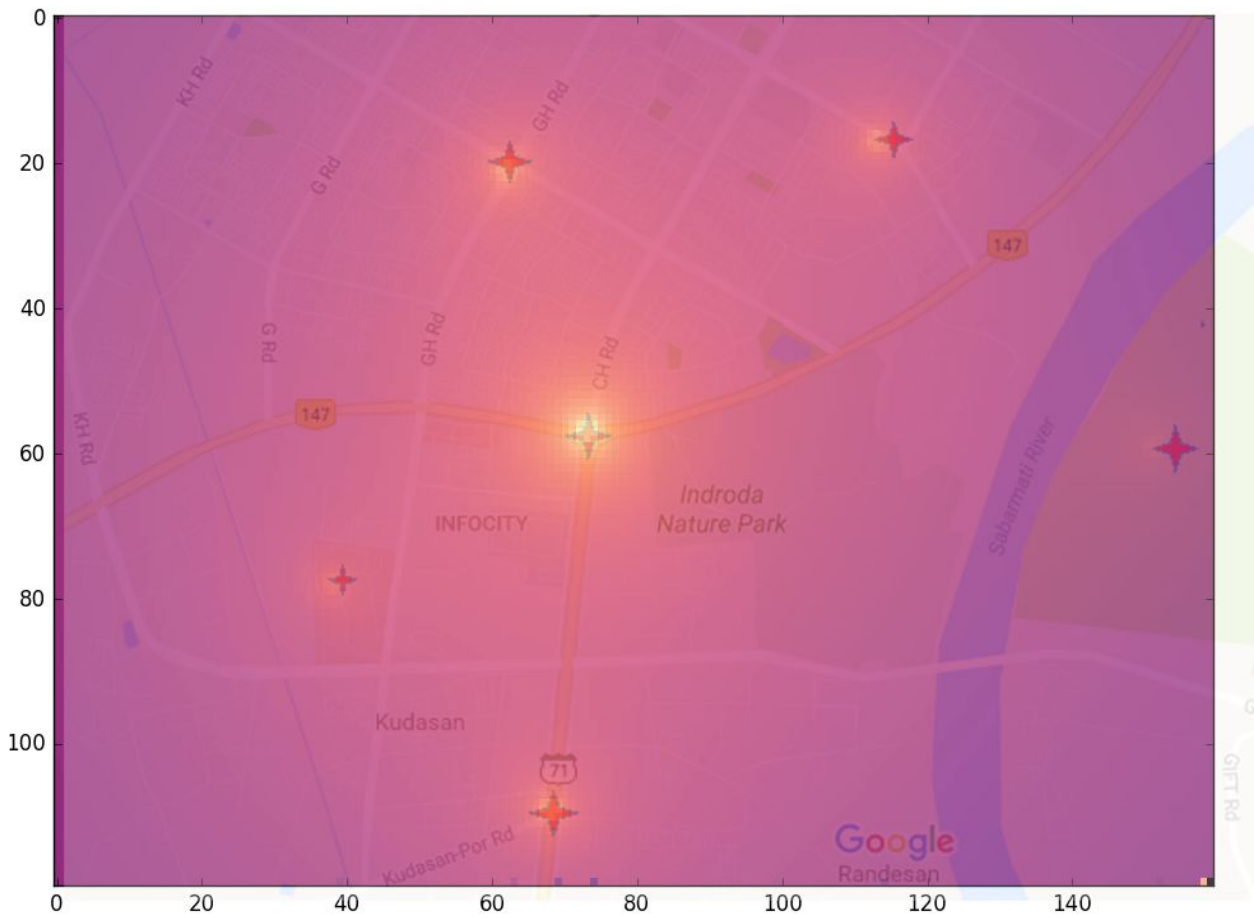
If we have enough sensors at different places, then we can use the data from all of these places to map it on a common space, and hence we can generate a heatmap[5] of that given space. We have used dummy values, to come up with a sample heat-map, which consisted of 6 sensor nodes at six different locations. We assumed the pollution levels according to the nature of the place(For e.g, area near roundabouts/ highways would have a pollution level which is relatively higher than the one near a park; only for illustration purposes, could vary in real data.). The sample heat map is as follows:



Star denotes the position of sensor values



Heat Map for the above locations



Heat Map superimposed with the actual map locations

For future extensions, we can add GPS module to the sensor nodes, to detect the position, making it easier to deploy at different places. We can also add more Gas sensors, so as to have a better bifurcation of different gases, some gases are more harmful than others. We can integrate Google maps, so we can use this data to come up with predictions that can be used by people for selecting their routes or by authorities to re-route traffic etc.

References:

1. Installing Firmware in ESP8266:
   https://roboindia.com/tutorials/nodemcu-amica-esp8266-board-installation

2. ESP8266 Programming Using Arduino IDE:
   - https://www.youtube.com/watch?v=G6CqvhXpBKM
   - http://learn.acrobotic.com/tutorials/post/esp8266-getting-started

3. How to post data to Google drive:
   http://embedded-lab.com/blog/post-data-google-sheets-using-esp8266/
4. MQ7 Datasheet (for effect of temp. and humidity on sensor values):
   https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7%20Ver1.3%20-%20Manual.pdf
5. Heatmap algorithm: http://www.geeksforgeeks.org/find-number-of-islands/