EN2560 — Internet of Things Design and Competition: Project Report

Group Number: 13 **Submitted Date:** July 08, 2021
Project Members: P.M.N.S. Bandara – 180066F, S.S. Hettiarachchi – 180237G, A.U.P.H. Athukorala – 180051F

AirSPEC: An IoT-empowered air quality monitoring system integrated with a machine learning framework to detect and predict defined air quality parameters

Problem statement

The air that surrounds us is the main source of our respiration.So, it is undoubtedly vital to state that the balanced air quality is utmost important to the environmental existence including the survival of humankind as well as other animals and plants.The air around us contains various harmful substances leading to a gradual deterioration of global air homeostasis. Particulate matter is identified to be a great threat to that homeostasis which includes a mixture of solid particles and liquid droplets found in the air. Some of these particles, such as dust, dirt, soot, or smoke, are large or dark enough to be seen with the naked eye. Others are so small they can only be detected using an electron microscope. The deleterious impacts through these particles such as enviornmental and health threats urge the need for efficient and real-time novel approaches to motivate greater levels of engagement in the air quality debate by empowering citizens around the world to demand cleaner air and better policy formulations and to effectively use the air quality information by collecting and processing impartial data.

Methodology

System Overview

The IoT based system is comprised of three crucial blocks: the node-red framework (including the dashboard interface), the communication between the node-red framework and node-mcu (through MQTT) and the mobile client (which is accessible via a web portal and a mobile application).

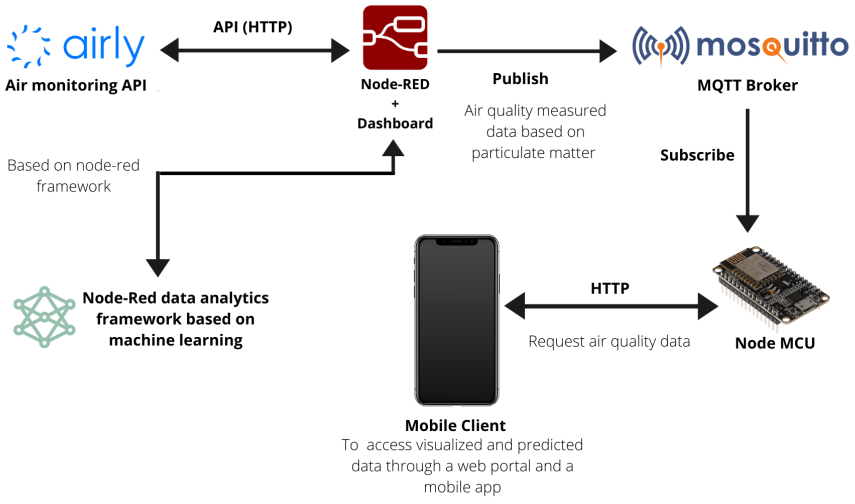

Figure 1: The implemented high-level block diagram of the system

A list of system functions and features

- "airly" air monitoring API
  - To obtain the real-time raw data of defined air quality parameters: Particulate Matter with different sizes
- Node-red framework
  - Initial data processing and virtualization obtained form airly API
  - Statistical machine learning model for time-series prediction using in-built modalities
- Node-red dashboard representation

- Location and time-based raw data virtualization for defined air quality parameters: particulate matter, air quality index, temperature, pressure and humidity
- Visualization of one-hour forested data and past data
- Simplified decision making options: location-based systematic representation
- File management system: Visualize and download the current and past data files

- Mosquitto server and protocols
  - To establish the publish-subscribe network application layer protocol between node-red and node-MCU
- To the client through mobile
  - Via Node-MCU:To obtain the processed data form node-red via the subscription through mosquitto to communicate with mobile (via http), Location and time-based data requesting and representation, Decisive notifications
- Mobile client: Web portal and the mobile application
  - To access the visualized-predicted data and notifications by the client
  - Predictions through nod-red based inbuilt framework for obtaining prediction through a conventional machine learning model

System Implementation

Node-red framework

Node-red is utilized as the base framework for obtaining the data from airly.org API and for visualizing the data. Further, it is the base for constructing the conventional machine learning model of *decision tree prediction* which is implemented in order to predict the time-series forecast values of the air parameters. Further, it communicates with the mode-mcu via assigned mqtt broker and the thus, the mobile client in order to process the interests of the users within the application arena.

Node MCU web server implementation

ESP8266 is used as the webserver for the data transmission between Node-RED dashboard and mobile. Data communication between ESP8266 and Node-RED is occurring through the MQTT communication protocol. The selected broker service is Eclipse Mosquitto. Node MCU and Node-RED both working as publisher and subscriber. Data communication between ESP8266 and mobile is occurring through the HTTP protocol. Here we developed a mobile app and web pages to communicate with the client and view requested data.As the ESP8266 is a local server, mobile phone and ESP8266 should be connected with the same WIFI network, unless it does not work as expected.

Mobile Application

In the system, a mobile application is used to visualize data to the user. This mobile application is designed using Android Studio 4.2.2, in order to facilitate the user to obtain air quality data in the present location.As an additional feature, the user can generate his own location coordinates and have access to his location or directions to a certain place via accessing the Google maps from the mobile application.

- "Obtaining current latitude and longitude in Android
  - After Adding location permissions for the application for receiving the location update wa LocationManager instance has been created as a reference to the location service.Then requesting location from LocationManager and receiving location update from LocationListener on change of location.
- "Access google map
  - The mobile application display a map, using the Maps SDK for Android. The Added fragment element to the map activity's layout file, defines a SupportMapFragment to act as a container for the map and to provide access to the GoogleMap object
- "Obtaining Air Quality Data
  - The user sends a HTTP request to the Node MCU server carrying the air quality parameter values and then obtain data from the web server through the mobile application . The local node MCU server carrying data in JSON format. So,the mobile application has been designed to access and fetch JSON type data.

Web Portal
In the system, a web portal has been created for user to input location coordinates from one web page and obtain air quality data from another web page.

Resource selection and allocation

Node-red framework
Node-red framework is supported by the following special libraries which serve the different functions of the integrated system.

- node-red-contrib-machine-learning-v2: for building machine learning model
- node-red-contrib-credentials: for including the credentials
- node-red-node-email: for sending emails where threats are upcoming
- node-red-contrib-fs: for organizing the files

Node MCU web server implementation
For the Arduino code implementation following libraries are used.

- "ESP8266WiFi.h Arduino library
  - For connecting with the router and communicating
- "PubSubClient.h Arduino Library
  - For MQTT implementation
- "ESP8266WebServer.h
  - For creating a local webserver and communicating

Mobile Application

- Android Studio
  - Android Studio 4.2.2 version has been used as the integrated development environment (IDE) for the mo-bile application development. This mobile application is based on a project in Android Studio which has modalities with source codes and resource files. The modalities include Android app modules, Library modules, and Google App Engine modules. The programming languages used for the development of mobile application is Java.
- Volley HTTP Library
  - Volley, the HTTP library which is capable of building a networking for the mobile application has been used here. It also provides the facility of Automatic scheduling of network requests. ,Transparent disk and memory response caching with standard HTTP cache coherence
- Postman API Client
  - The Postman API client allows users to create and save simple and complex HTTP/s requests, as well as read their responses.This API client has been used for the project to test the APIs(Application Programming Interface).
- Google Maps
  - Google map has been used to display the current location of the user, navigate location direction and search location

Web Portal

- HTML
  - Has been used to design the content to be displayed in the web page
- CSS- Cascaded Style Sheets
  - This has been used for describing the presentation of the document written HTML.

System functionality achieved

Node-red framework

The admin user can enter the coordinates of the location where he prefers to obtain the air quality. The temporal variations and the raw data are visualized through the dashboard of the node-red. Further, the user can request and visualize the immediate past data and obtain the forecasted data through the API as well as the built machine learning model.

In addition to the basic capabilities, the user can download raw data files and data set which is used to develop the machine learning model. Furthermore, the framework is integrated to work with the requests which are adhered through the mqtt requests through the node-mcu and web and mobile portals of the users.

Figure 2: Dashboard preview 01

Figure 3: Dashboard preview 02

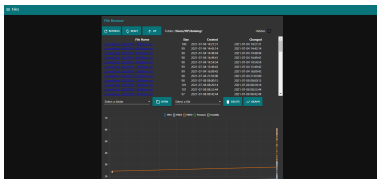

Figure 4: Dashboard preview 03

Mobile Application

Obtaining current latitude and longitude in Android

The user can click the obtain location coordinates button on the mobile application interface and obtain the current latitude and longitude of his location.

Google Map

The user can choose the option of see the current location and go to the map view in mobile application then select google map option he need to access.

Obtaining Air Quality Data

The user friendly interface of the mobile application visualize the air quality parameters with their values at the given time, the forecast air quality data and also air quality data for the last 24 hours.

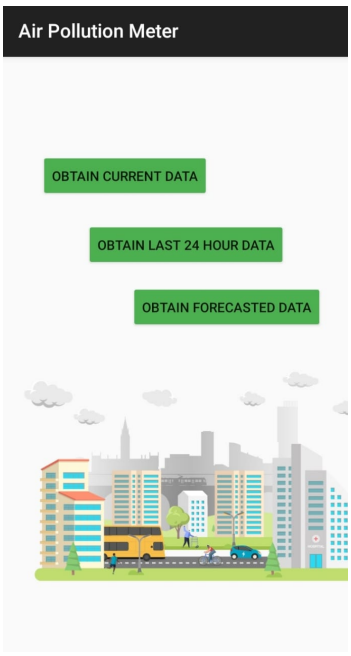

Figure 5: Air quality data

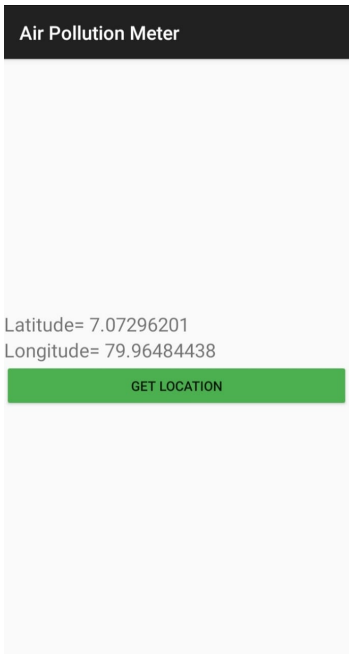

Figure 6: Obtaining coordinates

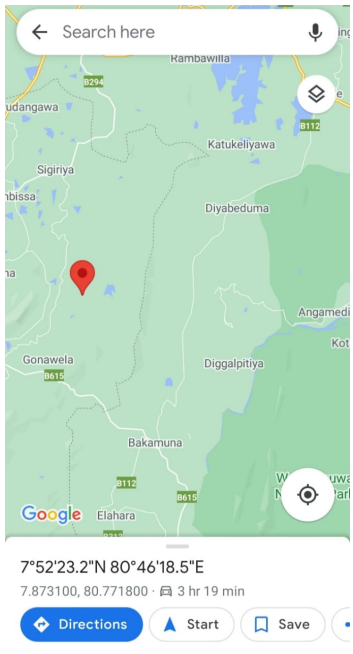

Figure 7: Access google map

Web Portal

Input location coordinates

The user visit the web portal and input the current location coordinates and enter.

Obtaining Air quality data

The user receives air quality data of the location he entered.

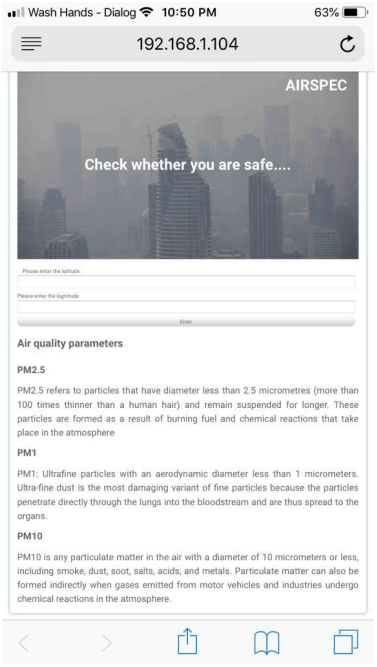

Figure 8: Coordinates entering

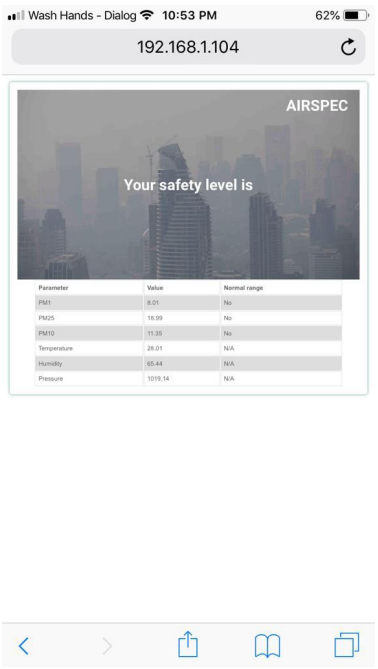

Figure 9: Recieving data

Figure 10: Node MCU web server