VOL. 13, NO. 6, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

POLLUTION MONITORING SYSTEM USING IOT

Gaurav Rout, Sairam Karuturi and Padmini T. N.
Department of Embedded Technology, School of Electronics Engineering, Vellore Institute of Technology University, Vellore, Tamil Nadu, India
E-Mail: gaurav.rout@rediffmail.com

ABSTRACT

Humanity, moving to an era focused upon development has forgotten the importance of sustainability and has been the major culprit behind the rising pollution levels in the earth's atmosphere among all other living organisms. The pollution levels at some places have reached to such high extents that they have started harming our own health. Hence, it is a significant indication to keep track of pollution levels in our surroundings in order to ensure that we do not get affected by the ill effects of pollution and at the same time improvise our actions in order to maintain pollution free environmental conditions. So, this paper covers the design and development of a device prototype that monitors air and noise pollution in real time and sends alerts to desired authorities whenever the pollution levels cross the threshold via push notification service on mobile phones using a cloud service provider, hence setting an example for an efficient IOT (Internet of Things) device.

Keywords: internet of things, IOT, raspberry Pi, pollution, monitoring, air, noise.

1. INTRODUCTION

Anything that causes adverse change in the environment and in its abiotic (lithosphere, hydrosphere and atmosphere) and biotic (flora and fauna) constituents is termed as pollution [1]. Broadly, pollution can be categorized into air, water and noise pollution. So, these three types of pollutions are a common problem for the healthy sustainability of the human society and are often caused by the human societies itself such as combustion, construction, mining, agriculture, warfare and motor vehicle emissions are a significant cause for air pollution, release of chemicals, oil spills and hot water from some industries lead to water pollution and loud sounds of machinery, honking of cars and loud music lead to noise pollution [2]. Likewise, there are numerous other activities which contribute to pollution in the environment apart from those mentioned above.

Considering the ill effects of pollution on humans, in 2012, one in eight of total global deaths were caused by air pollution which was 7 million premature deaths globally [3]. These deaths were a result of numerous diseases such as ischemic heart disease, chronic obstructive pulmonary disease, stroke, lung cancer and acute lower respiratory infections in children [3]. The causes for all those diseases were associated with outdoor and indoor air pollution combined. Now, if one talks about water pollution, consuming contaminated water can lead to serious health issues in human beings and one might get affected by life threatening waterborne diseases caused by protozoans, viruses and bacteria's such as amoebiasis, hepatitis A, E coli and dysentery. As per the WHO (world health organization) these diseases have a share of around 3.6% in the total daily global burden of diseases [4], and cause about 1.5 million human deaths annually. Similarly, noise pollution is also as harmful as the other two kinds of pollution as it may lead to hearing impairment, hypertension, ischemic heart disease, annoyance, and sleep disturbance [5]. Many people especially in big cities are affected by the adverse effects of noise pollution and suffer from various health issues related to the latter.

So, by going through all these statistics mentioned above one may conclude that it is very important to monitor pollution at the minutest levels, starting from an individual's home to the entire country as a whole. Hence, this paper illustrates the design and implementation of an idea to build a device that is capable of monitoring air and noise pollution in real time and alert the authorities in case of an increase in the pollution levels above the set thresholds.

2. RELATED WORK

Considering pollution an alarming topic these days, a lot of work in the field of pollution monitoring has been done previously in order to curb the pollution levels, keep it under control and take necessary steps to avoid its ill effects.

One such step by C. Balasubramaniyan *et al*. [6] suggests the deployment of various sensor nodes at different locations to check for air quality. It gathers the sensor data through a Raspberry Pi and sends it for computation on ThingSpeak web interface which in turn plots the graphs of all the data from various nodes using MATLAB as its basic coding language. It also posts tweets of the data associated to noise pollution on twitter.

Anjaiah Guthi [7] has proposed a 4 tier model of an air and noise monitoring system which uses a noise sensor and a carbon monoxide (CO) sensor in order to check for pollution in the environment. It aims to make the environment interactive with the objects through wireless communication. Arduino Uno has been used as to compute the data and then all the data is stored in Google spread sheets so that it is accessible from anywhere in the world.

In [8], a general framework to monitor environmental conditions such as temperature, light intensity, current and voltage is proposed. It's a 3 sensor network in which ZigBee wireless sensor network is used for wireless

VOL. 13, NO. 6, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

communication and all the data is displayed on a website in time series graphs.

3. PROPOSED MODEL

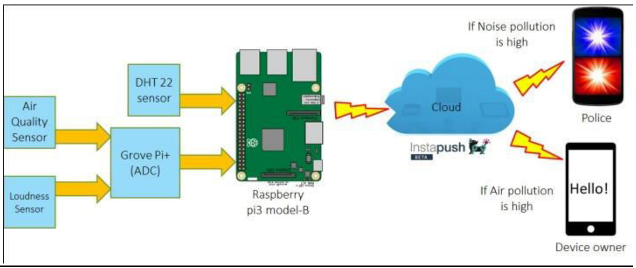


Figure-1. Block diagram of device prototype.

Figure-1 shows the block diagram of the proposed device prototype. This device monitors air and noise pollution using an air quality sensor and a loudness sensor respectively. It is also capable of sensing temperature and humidity using a digital temperature and humidity sensor, which it uses to find the dew point, which in turn helps in predicting rough visibility. All the computation associated with the sensor data is done in a Raspberry Pi (system on Chip).

In case of high air pollution, all this data is sent to the device owner on his/her mobile phone as an alert. And in case of high noise pollution, this device sends an alert to the police along with its unique token number which can either be traced using a database of token numbers for all the installed devices in the locality, or its location can be viewed using the "location magic" website whose link is also sent with the alert. Also, it takes care that at what time of the day the noise pollution occurred and based on that, the threshold is varied. All this data is sent via push notification service provided by a third party application, Instapush. This application must be installed in the device owners mobile as well as in the mobile phone of the police. Hence, helping the police to pin point the location where the noise pollution is high.

4. TECHNICAL SPECIFICATIONS

The following components have been used in designing this device prototype-

A. Raspberry pi 3 model B (system on chip)

Figure-2. Raspberry Pi 3 model B.

Raspberry Pi3 Model B is a small sized SOC (system on chip) which is capable of doing all the tasks that an average desktop computer can do like spread sheets, word processing, internet browsing, programming, gaming etc. Raspberry Pi3 Model B, built on the latest 1.2 GHz Quad Core Broadcom BCM 2837 ARMv8 64bit processor, is the 3rd generation of Raspberry Pi and is faster and more powerful than its predecessors. With built-in wireless and Bluetooth modules, it is an ideal IOT (internet of things) solution. It also consists of 1GB RAM, 4x USB 2 ports, 40pin extended GPIO, HDMI and RCA video output [9]. Hence making it ideal for computing the sensor data in our prototype.

B. Grove Pi +

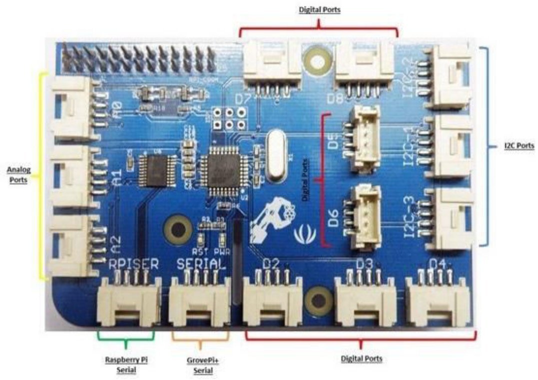


Figure-3. Grove Pi + board.

Figure-3 shows the Grove Pi+ board which is the bridge between the Grove sensors and the Raspberry Pi. It consists of ATMEL Atmega 328P processor on it. It is mounted on top of the Raspberry Pi covering 26 pins including power, ground and GPIO. The Grove sensors are connected to this board and accessed through the Raspberry Pi. It has 5 digital ports, 3 I2C ports, 3 analog ports and 2 serial ports, making it ideal to stack up multiple sensors.

C. Grove air quality sensor

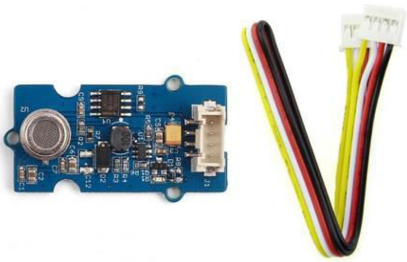


Figure-4. Grove air quality sensor with connector.

Figure-4 shows the grove air quality sensor which is designed to comprehensively monitor indoor air condition. It is responsive to a wide scope of harmful gases like carbon monoxide, alcohol, acetone, thinner, formaldehyde etc. Due to the measuring mechanism, one

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

cannot get a quantitative output to describe the concentration of target gases from this sensor. But it is still competent enough to be used in applications that require only qualitative results, like here [10].

D. Grove DHT 22 sensor

Figure-5. Grove temperature and humidity sensor pro (DHT 22).

Figure-5 shows the DHT 22 sensor which measures temperature and humidity. It is the advanced version of DHT 11 sensor and is more accurate than the latter. The detection range of this sensor is 5% RH - 99% RH, and -40 °C – 80 °C. And its accuracy reaches up to 2% RH and 0.5 °C where RH is the relative humidity [10]. This sensor is used in this device to detect temperature and humidity.

E. Grove loudness sensor

Figure-6. Grove loudness sensor.

Figure-6 shows the shows the grove loudness sensor which is designed to detect the sound in its environment. It is based on LM2904 amplifier and has a built-in microphone. It amplifies and filters the high frequency signals that are received from the microphone, and gives a positive envelope as output which is used for Raspberry Pi's signal acquisition. The output value depends on the level of sound input. In order to avoid unnecessary signal disturbances, input signal is filtered twice inside the module. Also, there is a screw potentiometer that enables manual adjustments of the output gain.

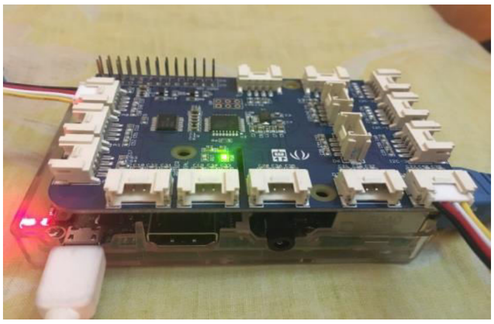
Its input voltage may vary from 3.5V ~ 10V and its working frequency is 50 ~ 2000 Hz with a sensitivity of -48 ~ 66 db. Output signal is in the range of 0 ~ 1023, analog [10].

5. IMPLEMENTAION

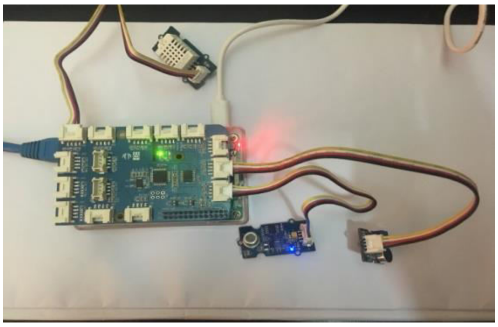
The flow of the device implementation can be categorized into five parts which would start from initial hardware and software setup. Then it would jump to essential calculations and computing followed by calibration of air and noise data. And at the end it would send the alerts.

Detailed explanation of each step is given below:

A. Initial setup



(a)



(b)

Figure-7. (a) Grove Pi+ stacked up on raspberry (b) Total device setup with sensors.

First of all, the Raspberry Pi is loaded with a customized operating system known as "Raspbian for Robots" developed by Dexter Industries. This operating system consists of pre-loaded libraries needed to access the Grove sensors. Then the Grove pi is stacked up on the Raspberry Pi and the sensors are connected as shown in Figure-7.

To make use of this device, the Instapush application must be present in the mobile phones of the police and the device owner, which can be downloaded from the application store of the mobile. For every device, two unique IDs corresponding to police and device owner are created in the Instapush website and also an API (application programming interface) is created in those IDs in order to generate unique token ID and password [11]. The events associated with the police ID is "noise pollution" and the event associated with the device owner ID is "air pollution". After this the respective IDs are accessed by signing in from the mobile applications of the police and the device owner.

Also to make use of location tracking services, an ID is created on "location magic" website to get another unique token number associated with the particular device and a script is run in the raspberry pi terminal of the device which fixes the token ID for that device, and sends location based on IP address to the location magic website on an hourly basis when connected to the internet.

Furthermore, all the data computing in the device has been done by using python 2 IDLE.

B. Dew point calculation and visibility

Dew point is the temperature to which air must be cooled to become saturated with water vapor [12]. For every 1 °C difference in the dew point and atmospheric temperature, the relative humidity decreases by 5%, starting with RH = 100% when the dew point equals the atmospheric temperature [13]. This is an approximation accurate to within about ±1 °C as long as the relative humidity is above 50%. Hence giving rise to the formula:

$$Tdp = T - (100 - RH)/5 \tag{1}$$

In (1), Tdp is the dew point temperature, T is the atmospheric temperature and RH is the relative humidity.

Now when the difference between the temperature and the dew point is less than 2.5 °C [14], the condition would lead to fog and hence giving us a rough estimate of visibility to be less than 1 km by the standard definition of fog [15]. Similarly the threshold of mist can be set around the temperature difference value of 5 °C and an estimated visibility of 2-5 km. In other conditions the visibility can be predicted as 10 - 296km or clear. All this is calculated by the device using DHT22 sensor.

C. Air pollution calibration

The air quality sensor gives the data in ppm (parts per million) in the range of 10 - 1000 ppm. In order to differentiate between different intensity of pollution, the pollution levels are divided into three parts, namely- fresh air, low pollution and high pollution. The threshold for low pollution has been set as greater than 300 ppm and the threshold for high pollution has been set as greater than 700 ppm [10]. Anything under 300 ppm is considered as fresh air. The sensors output is based on the presence of any one of the gasses to which the sensor is susceptible such as alcohol vapor, carbon monoxide, acetylene etc.

The thresholds of the sensor can be calibrated with reference to fresh air when the final device is being installed at any particular location. Also these thresholds would vary based on the country or state rules related to pollution levels.

D. Noise pollution calibration

Sound levels which can be considered as noise varies based on the location and time. The tolerable noise levels at residential areas during the day time is 55 dB (A) Leq and at night time is 45 dB (A) Leq [16]. Similarly it varies for industrial areas, commercial areas and silence zones. That's the reason; the threshold of this device for loudness sensor is customizable, based on the location where the device is being installed. Also the time is monitored simultaneously and based on the time the threshold is decided.

In order to avoid false triggering of an alert in case of a short burst of sound, the duration of sound that is to be considered as noise pollution can also be set as the device recognizes continuous sounds as noise pollution instead of a single high peak.

The output of the device is in between 0 – 1023 and the respective threshold can be found out using initial calibration of the device in the presence of a monotonous sound in a noise free environment.

E. Sending push notification

In order to make the device an efficient IOT (internet of things) device it is supposed to be connected to the internet where it sends its data to be monitored from anywhere in the world. So, when all the calculations have been done by the device, if there is an increase in the level of air pollution and if it exceeds the threshold specified for the device in that country then a notification is sent to the device owner's mobile which consists of the air pollution level in ppm with a statement that the pollution is high or moderate along with a predicted visibility. It also gives the information about the temperature, humidity and dew point of that area at that point. The device owner gets the alert via the Instapush application installed in his mobile.

Now, if the sound level increases in that particular location where the device is kept, and the sound is consistent for a duration that it's to be considered as noise pollution (set according to country norms) then the local police is informed about it via the same Instapush application. But along with the alert, the unique token ID of the device is also sent so that the device can be located either by using a separate data base or by going to the link of location magic and typing the token id to sign in and locate the device as the link is also forwarded with the alert.

6. RESULTS

A. Air pollution results

Figure-8 shows the raspberry pi terminal when it is monitoring air and noise pollution levels. When the air sensor value (marked in red, 502 ppm) exceeds the threshold (300 ppm), one can see that the notification is sent and acknowledgement is received by the raspberry pi in the very next line. Also, it can be noticed that when the air sensor value (marked in blue) was below the threshold, no notification was sent.

ARPN Journal of Engineering and Applied Sciences

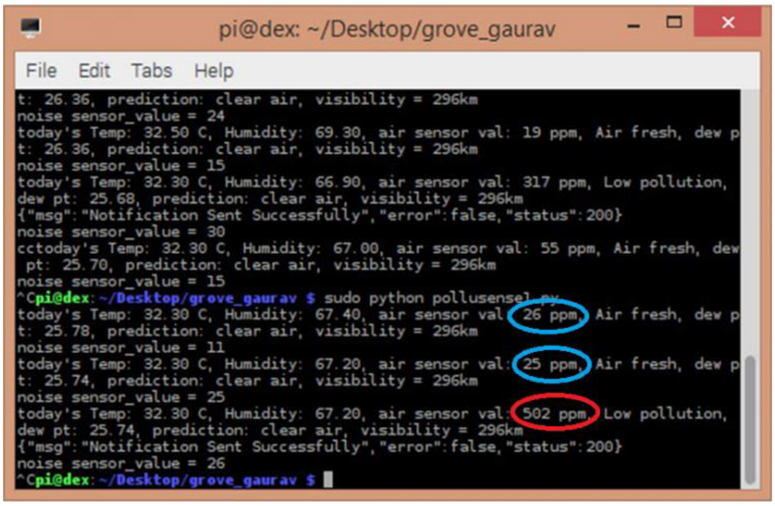
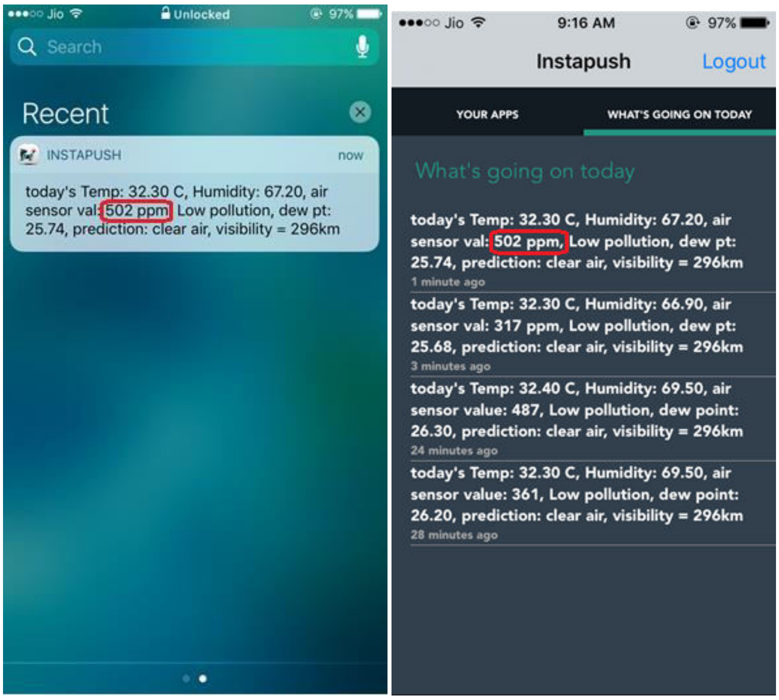


Figure-8. Raspberry Pi terminal to show presence of air pollution with 300 ppm as threshold.



(a) (b)

Figure-9. (a) Notification on the device owner's mobile phone (b) notification as seen from device owner's Instapush application.

Figure-9 shows the device owner's mobile screen. Notice that when the threshold was exceeded in Figure-8, in the Raspberry Pi terminal with a value of 502 ppm, the same value can be seen on the mobile screen with all the other data which was a part of the notification and was a result of all the computing of the sensor values in the device. So, by this the device owner can come to know about the air quality at his home or in his surroundings and can take the necessary measures to prevent himself from the ill effects of pollution.

ARPN Journal of Engineering and Applied Sciences

B. Noise pollution results

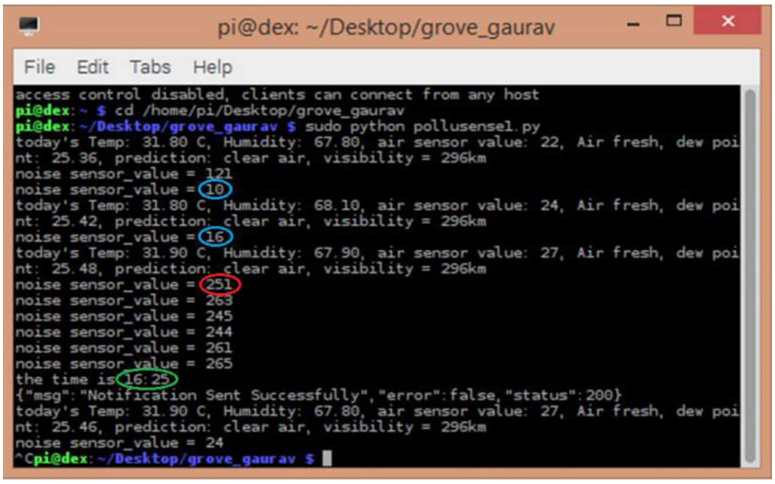
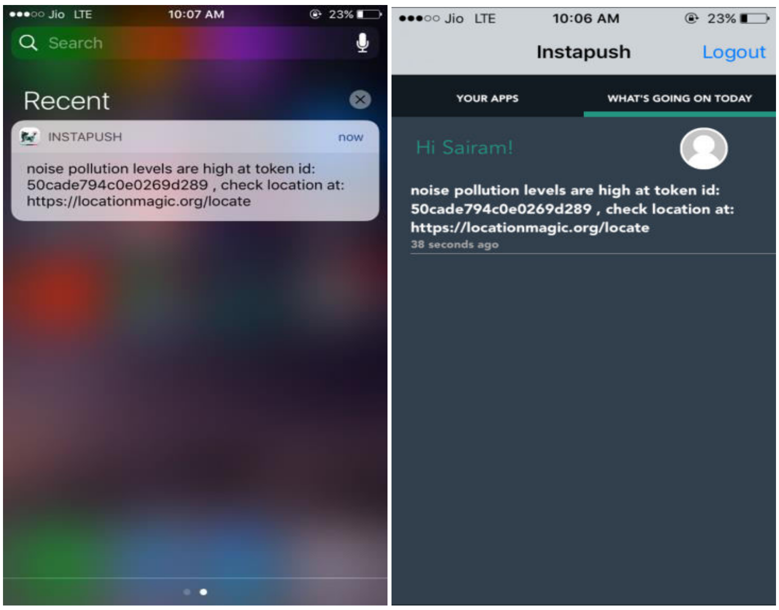


Figure-10. Raspberry Pi terminal to show noise pollution with noise sensor value 100 as threshold.

In Figure-10, when the noise sensor value (marked in blue) was below 100 (threshold for experiment) no notification was sent. But when it increased to 251 (marked in red), which is above threshold, continuous monitoring started to take place with a gap of 2 seconds (experimental duration) in between to check if the noise was just a short loud sound or a recurring sound to be termed as noise pollution. When it was confirmed that it continued for 10 long seconds by checking 5 extra values and ensuring all were above threshold, a notification was sent to the mobile of the police or for that matter "concerned authorities" and acknowledgement was received by the device.

But, one can notice that, before sending the notification the Raspberry Pi terminal flashes the time, stating it is 16:25 (clock time during experiment, marked in green). This "time" is important because all the above process happened as the "current time" during acquisition of loudness data exceeded the threshold time of 16:20, which was kept for validating the results. If it would have not exceeded the threshold of time and would have only exceeded the loudness threshold of 100, then no notification would have been sent and no continuous monitoring would have occurred. In real life scenario, in most of the Indian cities, loud noise is prohibited after 10 pm in residential areas, hence for that, the threshold time would be 22.00.

The notification sent, consists the unique token ID of the device along with location tracking website link.



(a) (b)

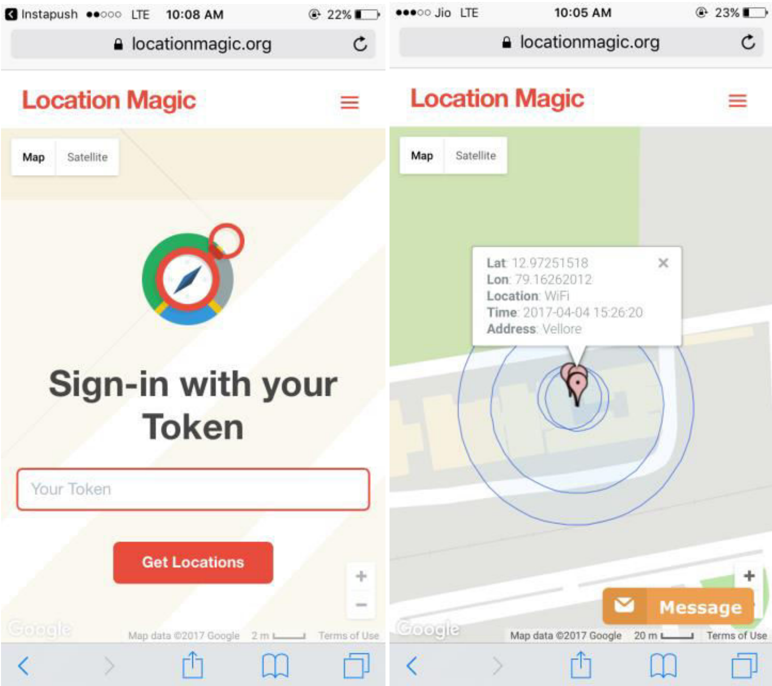
Figure-11. (a) Notification on the mobile phone of the police (b) Notification as seen from police's Instapush application.

In Figure-11, it can be seen that the token ID is received in the notification and this token ID is further used to locate the device by clicking on the link in the notification. This link is redirected to location magic website which then uses the IP address of the device and the data from the Wi-Fi routers to pin point the device on a map.

It is very important that the device remains connected with the internet all the time in order to send the alerts on mobiles as well as to give the accurate location. If the device is moved from its position, it can also be traced as the location magic website stores the previous location and the script running in the device refreshes its location details hourly.

After clicking on the link in the notification, the login page of location magic website appears, as shown in Figure-12 (a) where the token ID would need to be filled in the space given.

On clicking the "get location" icon the police/authorities would be able to trace the device on the map as shown in Figure-12 (b).



(a) (b)

Figure-12. (a) Login page of location magic website (b) location of the device on map.

7. CONCLUSIONS

This device is the prototype of a self-generated idea and is a step to monitor noise and air pollution together, serving multipurpose roles. The device can be modified by using better sensors in order to predict much more accurate visibility and also its dependency on third party applications can be reduced by developing a device specific application. Although in some aspects the accuracy is a key issue but still this device can be used in the practical environment where exact values don't have significance, instead a rough estimate of range of high, low and no pollution are enough to hint a layman to take necessary steps to curb the pollution and to help the police to keep a check on noise pollution levels. Hence, this device is capable enough to fill the gap in the race to overcome environmental pollution.

REFERENCES

[1] https://www.merriam-webster.com/dictionary/pollution.

[2] P Aarne Vesilind, J. Jeffrey Peirce, Ruth F. Weiner. 2003. Environmental Engineering, 4e: 375-382.

[3] WHO, report 2014. http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/.

[4] 2014. Burden of disease and cost-effectiveness estimates. World Health Organization.

[5] Passchier-Vermeer W, Passchier WF. "Noise exposure and public health. Environ. Health Perspect, 2000, 108 (Suppl 1): 123-31.

[6] Balasubramaniyan, C. and D. Manivannan. 2016. IoT Enabled Air Quality Monitoring System (AQMS) using Raspberry Pi. Indian Journal of Science and Technology. 9.39.

[7] Anjaiah Guthi. 2016. Implementation of an Efficient Noise and Air Pollution Monitoring System Using Internet of Things (IoT). International Journal of Advanced Research in Computer and Communication Engineering. 5(7).

[8] Kelly, Sean Dieter Tebje, Nagender Kumar Suryadevara and Subhas Chandra Mukhopadhyay. 2013. Towards the implementation of IoT for environmental condition monitoring in homes. IEEE Sensors Journal. 13.10: 3846-3853.

[9] www.au.element14.com.

[10] www.seedstudio.com.

[11] www.instapush.com.

[12] National Air Quality index. 2014. Available from: http:// www.cpcb.nic.in/FINALREPORT_AQI_.pdf.

[13] Lawrence, Mark G. 2005. The Relationship between Relative Humidity and the Dewpoint, Temperature in Moist Air: A Simple Conversion and Applications. Bulletin of the American Meteorological Society. 86 (2): 225-233.

[14] Fog -AMS Glossary 2013.

[15] Federal Meteorological Handbook Number 1: Chapter 8 – Present Weather (PDF). Office of the Federal Coordinator for Meteorology. 1 September 2005. pp. 8-1, 8-2.

[16] Ministry of Environment, Forest and Climate Change, www.envfor.nic.in.