Smart Pollution Monitoring System

Stanly Wilson, Tony Manuel, Peter Augustin D

Abstract*: *The world has travelled a long way through the industrial revolution. One of the consequences that the industries and its different forms gave to humanity is pollution. The environment that we live is being polluted in different ways. Different parts of the world are already experiencing air pollution as a matter of concern. The increasing amount of industries and the emission of gas by the vehicles cause much damage to the air. We are in a situation where we need to monitor the amount of pollution in our areas of living and working. In order to monitor pollution, the paper proposes an efficient and low-cost method with the help of the internet of things (IoT). The system is designed to monitor the levels of CO, CO_2, smoke, alcohol, NH_3, temperature and humidity. The various alarms and notification are arranged in such a way that the information is given when there is any sign of threat. The remote monitoring is made possible with dedicated website and mobile app.

Index Terms: Air Quality Monitor, Cloud Server, Pollution, Prototype, Sensors, Toxic Gases.

I. INTRODUCTION

From the time immemorial, human beings were fascinated by the weather and climate changes. The rise and fall of the temperature made different seasons. The same climate still fascinates them by being affected by human interventions. Nowadays people are very cautious of the pollution in their living areas. Many cities in the world are already experiencing the impact of air pollution. Air pollution could cause various diseases. The exposure to air pollution, whether for short or long duration could increase the risk of cardiovascular diseases [1]. Some of the diseases caused by air pollution may be chronic in nature while others are not. They could range from respiratory irritation and infection, lung cancer and lung diseases, bronchitis, asthma and other diseases that are connected with the respiratory system. Air pollution could both cause and aggregate the above conditions that may lead to premature mortality or shortens life cycle [2]. The pollution rate is already in the danger level affecting the ordinary lifestyle of the people. The world is worried these days at the drastic changes that happen in the climate. The first move to control air pollution is to monitor it well [3]. There were different methods that are employed to understand and predict the weather. They ranged from simple to complex methods as well as from inexpensive to expensive. There are different sources from where the weather information could be obtained. The government and non-government organizations provide the weather statistics that gives the general information. This information could be very useful in the areas of agriculture, construction, various modes of transportation etc. This information is general in nature, means, it gives the information of a particular geographical area. What if someone wants to monitor the quality of air in their company or living place? With an increasing amount of air pollution, it becomes a necessity to have products that are not complex, easily available, less costly and that could be monitored remotely.

The paper focuses on air pollution in particular with the added advantage of temperature and humidity readings. One of the important aspects of weather forecasting is to make sense of the data that the sensors provide. These readings need to be made useful information for the people. For that matter, it must be presented well and accessible from anywhere. The data obtained, for that matter, must be processed and communicated to some server with the help of modules like GSM, Bluetooth, wired or wireless transmission etc. Another important aspect of the weather forecasting is to make the data reliable. For that, the sensors used must meet the standards that are set.

This paper brings a prototype system that is developed for particular situations like the home, office or some small-scale industry. Using relatively cheaper sensors, it tries to provide reliable and accurate readings of temperature, humidity, CO, CO_2 and other harmful gases that may be present in the air. The paper has the following sections. The literature review sheds light into the existing works and papers that are available in the area of air pollution detection and managing. Air quality parameters help to understand various toxic gases that can be detected with the help of this proposed model. The proposed methodology elaborates the structure of the prototype. The component description will discuss various tools and sensors that are used to create the prototype. The implementation part will look into coding in a brief manner, how the system works and the obtained data be analysed. Finally, the conclusion brings the benefits and possibility for the future enhancement of the model.

II. LITERATURE REVIEW

The paper [4] deals with climate monitoring using Raspberry Pi with the help of a specific protocol namely, message queueing telemetry transport (MQTT). MQTT uses a publish-subscribe architecture where the clients publish the data to the MQTT broker, and those clients who are interested can subscribe to the broker to receive the data.

Revised Manuscript Received on 30 March 2019.
* Correspondence Author
Stanly Wilson*, Computer Science, Christ (Deemed to be University), Bangalore, India.
Tony Manuel, Computer Science, Christ (Deemed to be University), Bangalore, India.
Peter Augustin D, Computer Science, Christ (Deemed to be University), Bangalore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

The proposed prototype contains sensors that provide the values for the parameters like humidity, temperature, toxic gases and pressure. The cloud server used in this paper is Adafruit.io that not only provides the storage of the data but also has a well-defined visualization of the obtained data.

The paper [5] speaks about air quality monitoring made with Raspberry Pi. This too uses the MQTT protocol. The parameters that are taken into consideration here are atmospheric pressure, humidity, temperature and other common toxic gases. The transfer of data to the cloud server is done through the Raspberry Pi which has access to a router. For the storing and analysis of data, the IBM Bluemix platform is used.

The paper [6], in its process of developing a prototype, uses Microchip PIC18F87K22 microcontroller to monitor the quality of air. The interval for the data collection and transfer is 10 minutes. This prototype uses various gas sensors that could detect the presence of toxic gases, a GPS module to provide the geographical location of the modules, and a Wi-Fi module to communicate the data to the cloud server. It gives an optional possibility to have GPRS module to those places where there is no possibility to have a Wi-Fi router.

The paper [7] proposes a weather monitor system with the help of the readings such as temperature, humidity, dew point, heat index and light intensity. The microcontroller used in this model is Arduino Uno. It uses two sensors to measure temperature, DHT11 and LM35. It has an LCD screen to display the readings. The analysis of the data is only possible if it is connected to a computer where a text file of the readings is saved. This model does not have a communication or transmission module to send the data to some server. This means that remote monitoring is not possible in that model.

The paper [8] combines sound pollution detection along with air pollution monitor to form a prototype. The model uses Arduino Uno to which the other components are connected. It uses the MQ135 sensor to detect toxic gases. For the sake of data transmission, it uses the ESP8266 Wi-Fi module which transfers the data from the system to the cloud server. The LCD screen is used to display the values locally and a buzzer is used as an alarm in to notify in cases of emergency.

The paper [9] brings out a model for the weather forecasting using Raspberry Pi. The prototype is specifically developed for the use of farmers and agricultural purposes. This can be inferred easily by looking into the sensors employed in the model like moisture and rain sensor. The prototype is a kind of model that can be converted for different purposes by adding or removing sensors. The Wi-Fi adapter added to the model transmits the data to the cloud server through a Wi-Fi router. The closeness of the Wi-Fi adapter to the router is important for the data transfer.

The paper [10] tries to focus on the aspect of a wireless sensor network (WSN) to achieve environmental monitoring. It uses multiple data transmission modules to communicate with the server. The prototype uses an Arduino Mega board to which the sensors are connected. It uses multiple ZigBee modules for data transfer. The microcontroller sends the data through one ZigBee which will be received by another ZigBee module. Finally, the data reaches to the Raspberry Pi which sends the data to the cloud server.

The paper [3] offers an air pollution monitor system specifically for toxic gases like Carbon monoxide (CO) and Methane (CH_4). Two sensors are used to collect the air quality details and Raspberry Pi is used to process the same and send to the cloud server. An online python based analytical tool named 'Plotly' is used for the visualization and analysis of the obtained data. The paper also gives a clear understanding of the internal functioning of MQ7 and MQ4 sensors.

III. AIR QUALITY PARAMETERS

Before entering to discuss on the sensors and its working, it would be better to have some idea on what makes the air pollutant. First among them is Carbon dioxide (CO_2). It is an odourless, non-combustible and colourless gas. It is capable of interfering the availability of Oxygen for tissues. It is the gas that is emitted when the living beings exhale and a necessary one for the process of photosynthesis. It is a process where the plants absorb the CO_2 and emit the Oxygen which is necessary for us to breath. The concentration of CO_2 in the environment is increasing in a dangerous manner due to the burning of fuels by vehicles and industries. As the amount of CO_2 increases, the level of Oxygen decreases. The second one is Sulphur dioxide (SO_2). It is colourless but has distinct odour and taste. It is an outcome of the fuel engine emissions and by the industries. Its presence causes breathing difficulties. The third is Carbon monoxide (CO) which is odourless, colourless and tasteless. It can cause the reduction of Oxygen to the organs and tissues in the body if inhaled by the living. The fourth is Nitrogen dioxide (NO_2). It is a brownish gas with a specific odour. The Nitrogen monoxide is emitted by the burning of fuels and when it reaches the atmosphere, it undergoes some chemical changes and forms Nitrogen dioxide. Along with these two other parameters that are considered in this paper are temperature and humidity. Increasing temperature at times indicates the presence of fire. Hence it is also good to take into consideration these parameters for better safety and care [1][8].

IV. PROPOSED METHODOLOGY

The prototype developed in this paper is as follows. It is divided into 4 different modules. Figure 1 presents the flow of the data. The first module is the data collection. It has two gas sensors namely MQ135 and MQ7, as well as DHT11 which is a dedicated temperature-humidity sensor. The detailed description of the sensors will be dealt elaborately in the following sections. The readings provided by these sensors are collected by Arduino Uno.

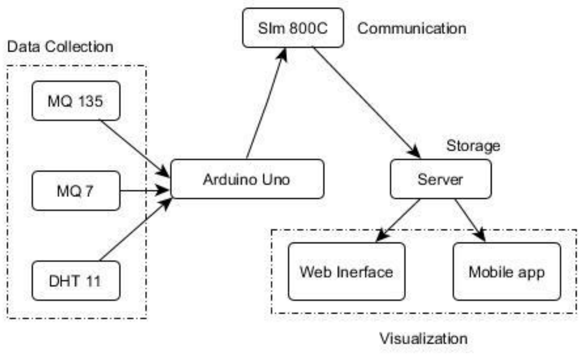

Fig. 1. Data flow diagram

The second module is the communication module. The collected data should reach to the storage location. For that purpose, the communication medium used here is SIM 800C. The third module is the storage. The values that are sent through the communication medium, stored in a reliable and efficient manner. The data is stored in a cloud server so that it could be accessed remotely and analysed well. The fourth is a visualization module. There are two ways to access the readings. One is with the help of a web browser and the other is with the android app dedicated for the prototype..

V. COMPONENT DESCRIPTION

The components that are used here are Arduino Uno, SIM 800C, MQ135, MQ7 and DHT11. There are different programming languages are employed and they are simplified C++ for Arduino, and Android for the mobile app. An array of web development languages like PHP, HTML, MySQL, Ajax and AngularJS are used for the website.

A. Arduino Uno

As an open-source platform, Arduino is used for coding and constructing procedures that can send and receive data from the devices that are locally or remotely located. The communication with these devices could be wired or wireless, with or without the use of the internet. The chip in Arduino can be programmed to perform numerous actions. In this paper, it collects the data from the sensors, and then push the same to the cloud server with the help of the SIM module. Hence both receiving and sending of the data are performed. It has a storage of 32KB while for the usage of the global parameters it has only 2KB. So, the program to be flashed into the Arduino Uno must be coded precisely and should use the space very wisely. It has two power options, with USB and external power supply. There are 6 analogue input pins and 16 digital pins that could be used for either input or for output. For the transmission of data, there are two pins named TX and RX, transmission and receiving respectively [11].

B. SIM 800C

The SIM module that is employed to transmit the data to the cloud server is SIM800C. It is a module that supports 4 different bands (quad band) for the data transfer like 850, 900, 1800 and 1900 MHz. It is a GSM/GPRS compactable module that used for both oral communications with the help of a mic, and for text communication (SMS), other than the communication over the internet. It permits the user to insert the 2G SIM to the slot through which the data or communication flows. It supports the TCP/IP protocol and the extended AT commands that are responsible for the transmission of data from Arduino. The low power consumption makes it a better choice for communication purposes [12].

C. MQ135

This gas sensor module is a stable one and SNO_2 is used as the sensitive material. SNO_2 has lower conductivity than that of the air. This property of SNO_2 makes the conductivity to increase if the volume of the polluted gas increases and detects the pollution. MQ135 can detect unhealthy gases like NH_3, Sulphides, CO_2, alcohol, smoke and other toxic gases. The module has a digital pin that enables it to work even when the controller is not present and this can be handy when one tries to detect a particular gas. If the presence of the gas to be measured in terms of PPM (parts per Million), then the analogue pin comes to use with the controller. Overall it is a very good choice to calculate the air quality in general [13].

D. DHT11

It is a digital sensor that provides temperature and humidity readings. The capacitive humidity sensor and the thermistor measure the air and give the values. Three main components constitute DHT11. The first is a resistive type humidity sensor which gives the humidity reading. The second is a negative temperature coefficient (NTC) thermistor whose function is to measure the temperature of the desired area. The third is an 8-bit microcontroller that converts and sends the obtained analogue readings of temperature and humidity into a single digital signal. Humidity is the amount of moisture in the air. There may be situations where the temperature may be low, but people feel the heat and it due to the high humidity. So, humidity readings make sense at times when those situations arise. DHT 11 is low cost, small and high-performance sensor [7].

E. MQ7

This sensor is used mainly to detect the amount of Carbon Monoxide (CO). It consists of a micro AL203 ceramic tube, a sensitive layer made out of Tin Oxide (SNO_2), a heater and a measuring electrode. It is a multipurpose and low-cost sensor to monitor the level of CO [14].

VI. IMPLEMENTATION

In this prototype, the sensors are connected to the Arduino Uno and kept in a plastic box with the sensors projected outward. It uses an external power supply using the power adapter. For the data collection, the system was placed outdoors in a town located in Bangalore, Karnataka, India. The intention of keeping the sensors projected outward at the bottom of the box is to get the readings well while other parts are not affected by the sunlight or the rain. Since the outward projection is below the box, no direct sunlight or rainwater will affect the readings. Figure 2 shows the setup of the system.

Fig. 2. The prototype setup

Figure 3 shows the circuit diagram and how the modules are connected to the Arduino Uno. Other than the components mentioned earlier, two more could be seen in the figure and they are a buzzer and RTC. There may be emergency situations that need to be alerted as soon as possible. Even though the alerts and warnings will be notified on the website and mobile app, it must be available locally also. At times when emergency situations arise, and data transmission may not be possible, buzzer provides an added advantage. It is programmatically coded to alarm when the readings go to the danger level.

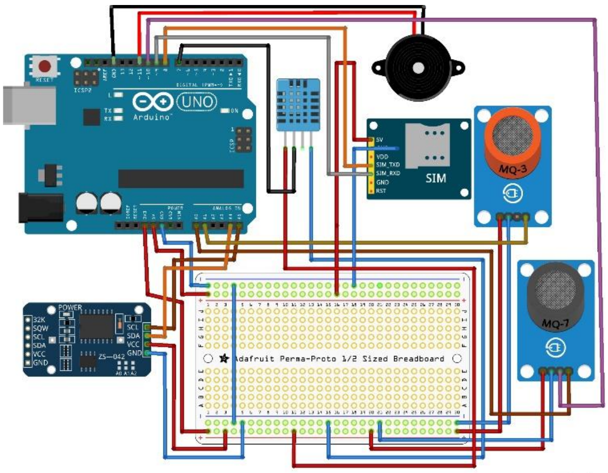

Fig. 3. Circuit Diagram

The second component is an RTC (real time clock). The module used here is the DS3231. This module is employed to give the current time in which the data is collected and sent from the device to the cloud server. The data may be collected at one time while it may reach the server another time. This is usually due to transmission time. The time generated by the RTC will be attached to the data and sent to the server. It is observed that there is a delay of more than two seconds between the data that was sent and the data arrived in the server.

The program flashed to the Arduino performs certain actions. It collects the data from the sensors and assigns into different variables. These details are then sent to the cloud server through the SIM module. The Arduino is given the direct power supply. At times due to the power failure, the SIM module does not start by itself and may need to reset manually. To overcome this difficulty, the reset function is coded in the program itself. The board resets by itself in every 15 minutes and by doing this there won't be any manual reset will be required. This functionality is achieved using the inbuilt function *millis*. The model is much dependent on the availability of the mobile network. It is assumed that in every area at least one mobile network provider has sufficient network. It is important to choose the best network provider and then needs to give that specific APN name in the program.

For the storage, the database was purchased and manually coded both the frontend and backend. The program in Arduino is made in such a way that the data would be sent to the cloud server in every twenty seconds. That means 3 records per minute and 180 records per hour. Each record contains nine fields out of which three are auto-generated by the database while six of them are received from the prototype. Figure 4 shows the database.

id 1	rtctime	rtcdate	time	date	co	air	temp	humidity
258243	13:17:27	24.09.2018	01:17:33 pm	24/09/2018	51	85	28	84
258242	13:17:06	24.09.2018	01:17:15 pm	24/09/2018	51	86	28	84
258241	13:16:05	24.09.2018	01:16:54 pm	24/09/2018	52	86	28	84
258240	13:15:45	24.09.2018	01:16:05 pm	24/09/2018	51	85	28	85
258239	13:15:25	24.09.2018	01:15:42 pm	24/09/2018	51	86	28	86
258238	13:15:04	24.09.2018	01:15:12 pm	24/09/2018	51	86	28	86
258237	13:14:32	24.09.2018	01:14:39 pm	24/09/2018	51	86	28	86
258236	13:14:11	24.09.2018	01:14:22 pm	24/09/2018	51	86	28	86

Fig. 4. Database View

The id, time and date are auto-generated by the server. Those values that come from the prototype are rtctime and rtcdate, air, co, temperature and humidity. The database has two tables. The first table has all the records that come from the prototype. The second table has the average readings. That means it takes the average of 180 values that come for that particular hour and then saves in the second table. It is from the second tale the graphs and analysis are performed.

VII. Results and Discussions

The data need to be interpreted well to make any conclusions. For the parameters of air quality, till 1000 PPM (parts per million) it is considered to be fresh. Once it goes beyond the limit then it could cause situations like headache, breathing trouble, chest congestion etc. These are the symptoms that were discussed in the air quality parameters. The range of CO is till 100 PPM. Once it goes above the mark, then starts the trouble. If not taken care well it may lead to irrevocable effects in the human body as these gases cause the reduction of Oxygen for the tissues [8]. The temperature readings are also needed to be under check. If the temperature goes above the normal level or a sudden increase in it would be an indication some fire outbreak that needs to be alarmed. The rise of these parameters will be alerted locally with the buzzer and remotely using the website and mobile app.

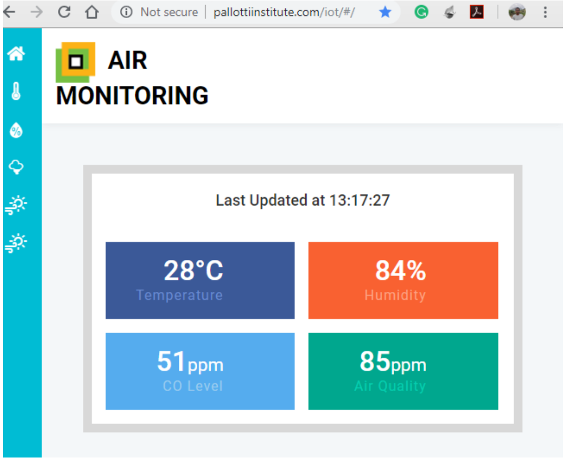

Fig. 5. Current Readings with a Timestamp

For better analysis and visualization, various graphs are plotted which gives a clear idea of the situation. Figure 5 shows the current readings from the sensor with the time. Though there are many graphs only three types of graphs will be discussed here. First is the daily readings graph which plots the readings of the particular parameter for the last 12 hours. Figure 6 shows the readings of a particular day. It can select the last 7 days readings from the drop-down menu.

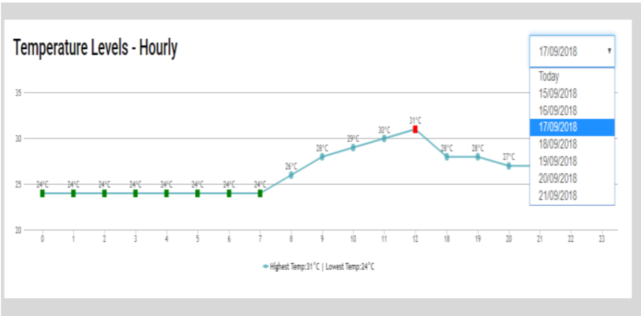

Fig. 6. Current Day Graph

The next is the date wise graph. This plots the lowest and highest readings of a particular parameter for the specific day. This graph will tell what was the minimum and maximum readings recorded on a particular day. Figure 7 shows the date wise graph.

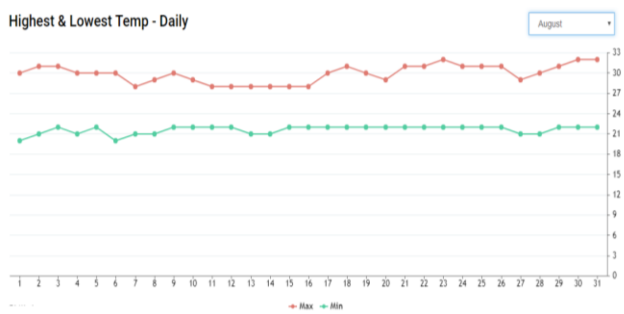

Fig. 7. Date-wise Graph

The third is plotted with the specific months. The lowest and highest reading for the particular parameter will be taken into consideration to plot the graph. As per the necessity, one can choose the specific month from the drop-down list. Figure 8 shows the monthly graph.

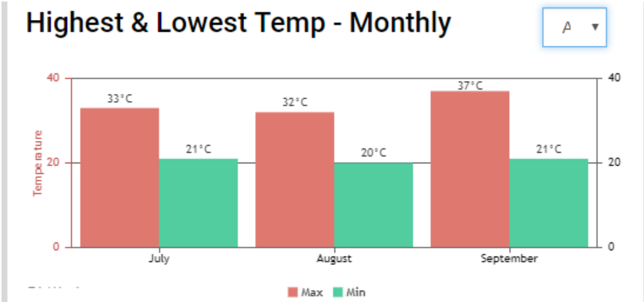

Fig 8. Monthly Graph

VIII. CONCLUSION

The paper focused itself on the development of a product that can be placed anywhere, even those places where there is no possibility of Wi-Fi. The mobile network in the country has grown such that most of the places have a mobile network. The basic GPRS provided by the SIM module is enough for the data transfer. Hence the use of the SIM module gave the freedom of placing the model anywhere. Only one thing that must be taken into consideration is that the SIM placed in the module must be enabled for the use of GPRS periodically. The readings obtained from the sensors are not exactly equal to the values obtained using the high-end devices, but they are still very much closer. The proposed model can be used well in both indoor as well as outdoors because of its compact structure and efficiency. The paper achieves its purpose of low-cost product with higher efficiency in some manner in this model.

REFERENCES

1. Brook, Robert D., Barry Franklin, Wayne Cascio, Yuling Hong, George Howard, Michael Lipsett, Russell Luepker et al., "Air Pollution and Cardiovascular Disease: A Statement for Healthcare Professionals from the Expert Panel on Population and Prevention Science of the American Heart Association", *Circulation*, vol. 109, no. 21, 2004, pp. 2655-2671.
2. Kampa, Marilena, and Elias Castanas. "Human Health Effects of Air Pollution", *Environmental pollution*, vol. 151, no. 2, 2008, pp. 362-367.
3. Karamchandani, Sunil, Aaklin Gonsalves, and Deven Gupta. "Pervasive Monitoring of Carbon Monoxide and Methane using Air Quality Prediction", *3rd International Conference on Computing for Sustainable Global Development*, 2016, pp. 2498-2502, IEEE.
4. Shete, Rohini, and Sushma Agrawal. "IoT Based Urban Climate Monitoring using Raspberry Pi", *International Conference on Communication and Signal Processing*, 2016, pp. 2008-2012. IEEE.
5. Kumar, Somansh, and Ashish Jasuja. "Air Quality Monitoring System Based on IoT using Raspberry Pi", *International Conference on Computing, Communication and Automation*, 2017, pp. 1341-1346, IEEE.
6. Marinov, Marin B., Ivan Topalov, Elitsa Gieva and Georgi Nikolov, "Air Quality Monitoring in Urban Environments", *39th International Spring Seminar on Electronics Technology*, 2016, pp. 443-448, IEEE.
7. Krishnamurthi, Karthik, Suraj Thapa, Lokesh Kothari, and Arun Prakash. "Arduino Based Weather Monitoring System", *International Journal of Engineering Research and General Science*, vol. 3, no. 2, 2015, pp. 452-458.
8. Pal, Poonam, Ritik Gupta, Sanjana Tiwari, and Ashutosh Sharma. "IoT Based Air Pollution Monitoring System using Arduino", *International Research Journal of Engineering and Technology*, 2017, pp. 1137-1140.

9. VivekBabu, K., K. Anudeep Reddy, C. M. Vidhyapathi, and B. Karthikeyan. "Weather Forecasting Using Raspberry Pi with Internet of Things (IoT)", *ARPN Journal of Engineering and Applied Science*, vol. 12, no. 17, 2017, pp. 5129-5134.
10. Jaladi, Aarti Rao, Karishma Khithani, Pankaja Pawar, Kiran Malvi, and Gauri Sahoo. "Environmental Monitoring using Wireless Sensor Networks (WSN) Based on IOT", *International Research Journal of Engineering and Technology*, vol. 4, no. 1, 2017, pp. 1371-1378.
11. Badamasi, Yusuf Abdullahi. "The Working Principle of an Arduino", *11th International Conference on Electronics, Computer and Computation*, 2014, pp. 1-4, IEEE.
12. Balaji, G. Naveen, V. Nandhini, S. Mithra, N. Priya, and R. Naveena. "IOT Based Smart Crop Monitoring in Farm Land", *Imperial Journal of Interdisciplinary Research*, vol. 4, no. 1, 2018, pp. 88-92.
13. Vandana, K., Chaitanya Baweja, Simmarpreet, and S. Chopra, "Influence of Temperature and Humidity on the Output Resistance Ratio of the MQ-135 Sensor", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 4, April 2016, pp. 423-429.
14. Deepa, T. "FPGA Based Pollution Control System for Vehicles using Special Sensors", *International Journal of Engineering Sciences & Research Technology*, vol. 4, no. 3, 2014.

AUTHORS PROFILE

Stanly Wilson Master of Computer Applications, Christ (Deemed to be University) Bangalore, Karnataka, India.

Tony Manuel, Master of Computer Applications, Christ (Deemed to be University) Bangalore, Karnataka, India.

Dr. Peter Augustin D, Associate Professor, Christ (Deemed to be University) Bangalore, Karnataka, India.