Navigation System in Hazardous Zone Using IIoT

[1]Muthukumar P, [2] Monika S, [3]Mugesh N, [4]Naresh K, [5]Yogitha S
[1]Student, [2] Student, [3] Student, [4]Student, [5]Assistant professor
[1]Electronics and Communication Engineering,
[1]Sri Shakthi Institute of Engineering and Technology, Coimbatore, India

Abstract: — The effect of air pollutants has long been adversely affecting the environment not only climate but also entire ecosystems, human property as well as agricultural crops. Particular pollutants such Carbon Dioxide, Carbon Monoxide and Nitrogen Oxides are common emissions and without proper monitoring method will continue to greatly affect the environment. Innovative technologies to identify, quantify and assess fluxes exchange of the pollutant gases in a locality are required. This paper proposes the development of a gas sensor system for a small Unmanned Vehicle to monitor pollutant gases in the desired locality, collect data and display the details. The prototype has a light portable gas sensor, ESP8266-12E, Raspberry Pi and an unmanned vehicle as principal systems. The prototype will be suitable to operate in the remote areas. One of the limitations is only limited power backup is available therefore a small and low power consumption payload is designed and built for this research. The specific gases targeted in this research are NH_3, NO_x, Benzene, and Smoke, CO_2 mostly produced by traffic and NH_3 from farming, with concentrations above 0.05 ppm and 35 ppm respectively which are harmful to human health. The developed prototype will be a useful tool for scientists to analyse the behaviour and tendencies of pollutant gases producing more realistic models of them.

Keywords — Environmental monitoring, Gas sensing, Pollutants, Unmanned Vehicle.

I. INTRODUCTION

Identifying, quantifying and assessing fluxes exchange of pollutant gases in atmosphere is important to have a better understanding of the problems they are causing and future tendencies [1]. Pollutant gases such as CO2, C6H6, NH3, NO_x, gases are mainly responsible for climate change [2, 3], global warming and plant diseases [4, 5]. Currently, there are different technologies available in the market to measure these gases effectively at the ground level; balloons, satellites and manned airplanes are the options available at the atmospheric level. The use of satellites and manned airplanes can be complex, and expensive to run. Manned vehicles cannot be sent to all remote and hazardous regions and there will be size complexity in such vehicles. Consequently, it is necessary to develop a cost-effective system capable of detecting and monitoring pollutant gases to allow further analysis of the behavior, tendencies and harm of pollutants. The system proposed in this research has MQ135 Sensor, ESP8266-12E, Raspberry Pi and an Unmanned Vehicle as its main components. MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed so as to be easy to implement. MQTT works on top of TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The idea of installing a gas sensor system in an unmanned vehicle for environmental purposes is a growing field and several pioneering works have been developed in this area. An Environmental Air Pollution Monitoring System (EAPMS) for monitoring the concentrations of major air pollutant gases has been developed [6], complying with the IEEE 1451.2 standard. This system measures concentrations of gases such as CO, NO2, SO2, and O3 using semiconductor sensors. The smart transducer interface module (STIM) was implemented using the analog devices' ADuC812 microcontroller unit. Network Capable Application Processor (NCAP) was developed using a personal computer and connected to the STIM via the transducer independent interface. Concentration levels and information regarding the STIM can be seen on the graphical user interface of the NCAP. Further, the EAPMS is capable of warning when the pollutant levels exceed predetermined maxima and the system can be developed into a low cost version for developing countries. IoT based gas emission monitoring system [7] is to introduce vehicle emission monitoring system using Internet of Things (IoT) which is a green thumb for tracking down vehicle causing taint on the city roads and measures multifarious genres of toxic wastes, and its level in air. The measured data is shared to vehicle proprietor via text message, and agencies of national environment. This assay shows

that the system runs abiding, an economical and can be controlled tractably, it can smell out the vehicle exhaust in real-time, and can improve the detecting level and accuracy of the exhaust monitoring system. In this system sensors such as CO, hydrogen and gas sensors are connected to the input pins of arduino and the values are sensed. If the sensed value goes beyond threshold value set in the program then automatically an alert message will be sent to the vehicle owner by using ESP8266. Implementation of a UAV-based platform for air pollution monitoring [8]. The measurement methods using pressurized balloons, satellite imagery, or earth stations result in considerable investment, as well as providing low space-time resolution. In Monterrey, Mexico, the Integrated Environmental Monitoring System (SIMA) consisting of 10 terrestrial monitoring stations, provide online information of pollution levels through its website however the spatial resolution is low. This research work proposes the design and development of a system for measuring atmospheric pollutants and tracking contamination sources with the use of an Unmanned Aerial Vehicle and a real time processing by using a metaheuristic algorithm. This algorithm is based on the simulated annealing methodology to generate the UAV movement directions, in the pollution source search, by considering that it is a stochastic system. Emissions of gases and particles from seafaring ships have been shown to impact on the atmospheric chemistry and climate [9]. To efficiently monitor and report these emissions found from a ship's plume, the concept of using a multi-rotor or UAV to hover inside or near the exhaust of the ship to actively record the data in real time is being developed. However, for the required sensors obtain the data; their sensors must face into the airflow of the ships plume. Airborne photographs can be expediently used in environmental monitoring [10]; A new and cost-effective platform of airborne remote sensing is the UAV (unmanned aerial vehicle) or drone. In this experimental work, aerial photos were made in Bakony Mountains using three UAVs equipped with small HD (high definition) cameras; resolution: 1,280 × 720 pixels. Within the framework of this work, a small lake was photographed, where the beginning of eutrophication was detected. This hardly can be observed from ground, however, it is visible on the aerial photos. This device senses and detects the gases in a locality using MQ135 gas sensor and sends the data to broker which is Raspberry Pi using MQTT protocol in a specified topic. Devices which are all connected to that topic will get the collected values displayed in a console. The motion and direction of the vehicle can be controlled using commands published through console to the broker and when the devices subscribe it they will receive the command through which the vehicle motion and direction can be controlled. In MQTT the number of subscribers and their information is not visible to the other subscribers. At the moment it is not possible to speak pure MQTT within a browser, because there is no possibility to open a raw API, but at the moment there are only a few browsers implementing this API [11]. The rest of this paper is organized as follows: (i) System design and construction describes the selection of the components, calibration and lab testing. (ii) System Assembly define the interactions of the MQTT-Gas sensor system and depict the principal components; (iii) Conclusion summaries the results and define the research direction of the project.

II. SYSTEM DESIGN AND CONSTRUCTION

A. ESP8266

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 WiFi SoC from Espressif systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. It is based on the Lua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua, json, and spiffs. ESP8266 ARDUINO CORE: As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled down to these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC that is available at the GitHub ESP8266 Core webpage. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCU.

Table 2.1: Pins of NodeMCU

IO INDEX	NODE MCU	IO INDEX	NODE MCU
0	GPIO16	7	GPIO 13
1	GPIO 5	8	GPIO 15
2	GPIO 4	9	GPIO 3
3	GPIO 0	10	GPIO 1
4	GPIO 2	11	GPIO 9
5	GPIO 14	12	GPIO 10
6	GPIO 12		

B. H-Bridge

An H bridge is an electronic circuit that enables a voltage to be applied across a load in opposite direction. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards. The H-bridge arrangement is generally used to reverse the polarity/direction of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted, or to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit. The following table summaries operation, with IN1, IN2, IN3 and IN4.

Table 2.2: H-bridge logic

IN1	IN2	IN3	IN4	RESULT
0	0	0	0	STOP
0	1	0	0	FORWARD
1	0	0	0	BACKWARD
0	0	1	0	RIGHT
0	0	0	1	LEFT

C. Gas sensor

The development of the gas sensors is a collaborative research between Brescia University (Italy) and QUT (Australia).The sensor response was calculated as $(R_{gas}R_{air})/R_{air}$, shorted as ΔR/R. This research work made use of MQ135 gas sensor. The material of MQ135 is SnO2, when exposed to the clean air, it is hardly being conducted, however when put it in an environment with combustible gas, it has a pretty performance of conductivity. Just make a simple electronic circuit, convert the change of conductivity to a corresponding output signal. MQ135 gas sensor is sensitive to Ammonia, Benzene steam, smoke, NOX and other harmful gases. Used for noxious gas detection, apply to ammonia, benzene vapor and other harmful gases/smoke having gas detection tested concentration range of 10 to 1000ppm. In normal environment which don't have detected gas, set sensor's output voltage as reference voltage, analog output voltage will be about 1V, when sensor detect gas, harmful gas's concentration increases 20ppm per voltage increase 0.1V.

D. OTA

OTA (Over the Air) update is the process of loading the firmware to ESP module using Wi-Fi connection rather that a serial port. Such functionality became extremely useful in case of limited or no physical access to the module. OTA may be done using: Arduino IDE, Web Browser, HTTP Server. This research work has been done with Web Browser.

Updates are done with a web browser that can be useful in the following typical scenarios: after application deployment if loading directly from Arduino IDE is inconvenient or not possible, after deployment if user is unable to expose module for OTA from external update server, to provide updates after deployment to small quantity of modules when setting an update server is not practicable It is necessary that the ESP and the computer must be connected to the same network.

E. MQTT

MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium. MQTT works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

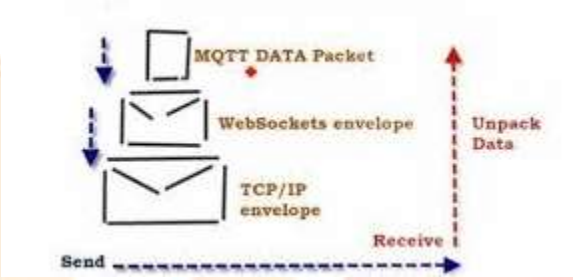

Figure 2.1: MQTT over Web sockets Illustrations

MQTT specification does a good job in describing what MQTT is all about. It is very light weight and binary protocol, which excels when transferring data over the wire in comparison to protocols like HTTP, because it has only a minimal packet overhead. Another important aspect is that MQTT is extremely easy to implement on the client side. This fits perfectly for constrained devices with limited resources. Actually this was one of the goals when MQTT was invented in the first place. The MQTT connection itself is always between one client and the broker, no client is connected to another client directly. The connection is initiated through a client sending a CONNECT message to the broker. The broker responds with a CONNACK and a status code. Once the connection is established, the broker will keep it open as long as the client doesn't send a disconnect command or it loses the connection

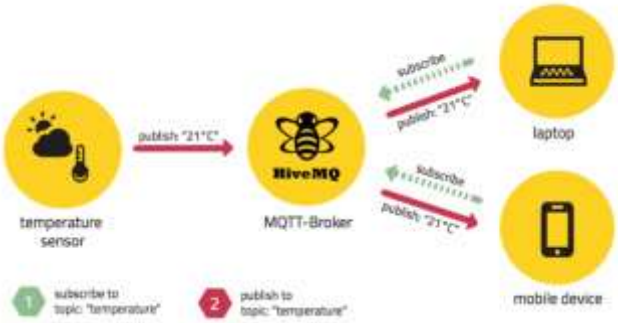

Figure 2.1: Pub/Sub

MQTT CONNECTION: The MQTT protocol is based on top of TCP/IP and both client and broker need to have a TCP/IP stack. The publish/subscribe pattern (pub/sub) is an alternative to the traditional client-server model, where a client communicates directly with an endpoint. However, Pub/Sub [10] decouples a client, who is sending a particular

message (called publisher) from another client (or more clients), who is receiving the message (called subscriber). This means that the publisher and subscriber don't know about the existence of one another. There is a third component called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly.

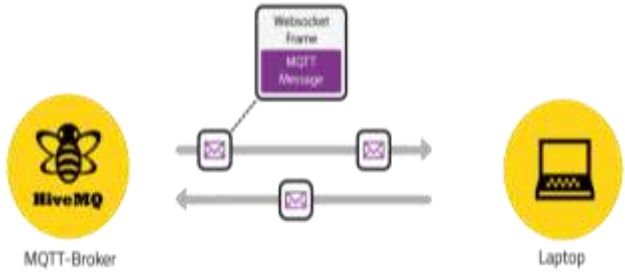

Figure 2.1: MQTT packet flow

BROKER: The Raspberry Pi is a series of small single board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache and equipped with 2.4 GHz WiFi 802.11n (150Mbit/s) and Bluetooth 4.1(24 Mbit/s) based on Broadcom BCM43438 FullMAC[12] .

QoS: A Quality of Service Level (QoS) for this message. The level (0, 1or 2) determines the guarantee of a message reaching the other end (client or broker)[13]. At most once (0), At least once (1), Exactly once (2).

III. SYSTEM ASSEMBLY

The controller retrieves the data from the Gas sensor and process the data and publishes the processed data in JSON.

(JavaScript Object Notation) format to the MQTT broker i.e. RPI for every 100ms. The messages published in the console are retrieved by the controller which has specific topic in it. Now the controller subscribes the input received and acts according to the data flow.

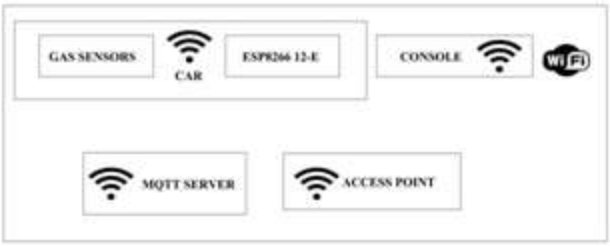

Figure 3.1: Block Diagram

The H-Bridge enables a voltage to be applied across a load in opposite direction and makes the DC motors to run in the desired direction.

Figure 3.2: Prototype

A small html page is made to support the console and the same gets subscribed to that specified topic. The real time values detected by the gas sensor can be viewed in the console and similarly the navigation commands given through the console can be retrieved in the controller.

IV. CONCLUSION

This paper reports on the gas sensor system designed and developed for a small Unmanned Vehicle to track various gases in remote areas and hazardous zones. The prototype reveals the feasibility of a low power, portable small gas sensor system built to track pollutant gases in the areas where humans unable to reach. The system is a necessary tool for scientist to better understand and model pollutant gas behavior and to provide information about gas emission and leakages. Ongoing activities focus on testing the system in a remote area with gases at different concentration and monitoring them. Further work can be made as the unmanned vehicle to carry more number of gas sensors and has the capability of making the vehicle to climb the wall. The vision of the project is to develop an uninterrupted platform to monitor the gas concentration and transmitting real time data to the GUI.

The vision of the project is to develop an uninterrupted platform to monitor the gas concentration and transmitting real time data to the GUI.

REFERENCES

[1] Ph. Ciais, B.M., W. Steffen, M. Hood, S. Quegan, J. Cihlar, M. Raupach, J. Tschirley, G. Inoue, S. Doney, C. Heinze, C. SabineK. Hibbard, D. Schulze, M. Heimann, A. Chédin, P. Monfray,A. Watson, C. Basu, S. 1997. The Investment Performance of Common Stocks in Relation to their Price to Earnings Ratio: A Test of the Efficient Markets Hypothesis. Journal of Finance, 33(3): 663-682. LeQuéré, P. Tans, H. Dolman, R. Valentini, O. Arino,J. Townshend, G. Seufert, C. Field, T. Igarashi, C. Goodale, A. Nobre,D. Crisp, D. Baldocchi, S. Denning, I. Rasool, W. Cramer, R. Francey,D. Wickland, A Strategy to Realise a Coordinated System of Integrated Global Carbon Cycle Observations, 2004, Integrated Global Carbon Observation Theme.

[2] GLOBALVIEW-CO2: Cooperative Atmospheric Data Integration Project - Carbon Dioxide, 2001.

[3] S. Montzka, E.J.D., J. H. Butler, Non-CO2 greenhouse gases and climate change. Nature, 2011. 476(7358): p. 43-50.

[4] R. Ghini, E.H., W. Bettiol., Climate Change and Plant Diseases. Sci. Agric. (Piracicaba, Braz.), 2008. 65: p. 98-107.

[5] L. Frati, E.C., S. Santoni, C. Gaggi, A. Guttova, S. Gaudino, A. Pati, S. Rosamilia, S. Pirintsos, S. Loppi., Effects of NO2 and NH3 from road traffic on epiphytic lichens. Environmental Pollution, 2006. 142(1): p. 58-64.

[6] An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements, Nihal Kularatna, Senior Member, IEEE, and B. H. Sudantha, Member, IEEE

[7] IoT Based Vehicle Emission Monitoring System, Abinayaa Balasundaram, Aiswarya Udayakumar, IJIRST National Conference on Networks, Intelligence and Computing Systems March 2017.

[8] Noe M. Yungaicela-Naula, Luis E. Garza-Castanon, Design and Implementation of an UAV-based Platform for Air Pollution Monitoring and Source Identification.

[9] E. Berman, M.F., J. Liem, R. Kolyer, M. Gupta., Greenhouse Gas Analyzer for Measurements of Carbon Dioxide, Methane, and Water Vapor Aboard an Unmanned Aerial Vehicle. Sensors and Actuators B: Chemical, 2012(0).

[10] https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe

[11] https://www.hivemq.com/blog/mqtt-essentials-special-mqtt-overwebsockets

[12]https://www.hivemq.com/blog/mqtt-essentials-part-3-client-brokerconnection-establishment

[13] https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-ofservice-levels