VEHICLE POLLUTION MONITORING SYSTEM THROUGH CLOUD

S.CHANDRASEKHAR

Assistant professor of Electronics and Communication Engineering,GITAM(Deemed to be university),Rudaram,Patancheru mandal,Hydrabad,Telangana,INDIA.

G.Nithin Kumar, S.Prashanth Reddy, Sanivarapu Shilpitha Reddy

Student, Student, Student

Electronics and Communication Engineering,

Gandhi Institute of Technology and Management, Hyderabad, India.

Abstract: **Pollution has a major role in depleting air system. The health risks of air pollution is extremely serious. The Internet of Things (IOT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. The IOT concerns the connection and interaction of physical devices like cars, thermostats, smart phones, home lighting, tide sensors, smart meters, etc. to the Internet. Previously, to integrate sensors with cloud computing and IOT, some have opted for Zig Bee hybrid network connected to a Wi-Fi or Ethernet gateway. This methodology of connection is unreliable. The main objective of this paper is to implement IOT to measure the pollution of public vehicles using MQ7 sensor which is sensitive for Carbon Monoxide is implemented along with ESP8266 Microcontroller which would detect the pollution levels of the transport vehicle. The amount of Carbon Monoxide emitted is sensed in real time and also the ID of vehicle is used for distinguishing the vehicle from others. These are then integrated to the Amazon Cloud IOT which is more secure and reliable and many services of AWS can be used along with it. This would enable a Simple Notification Service (SNS) to send a push notification to the vehicle owner whenever the vehicle is causing higher level of pollutants.**

Index Terms – **AWS IOT(Amazon Web Service Internet Of Things),MQTT(Meassage Queuing Telemetry Transport), ICMP(Internet Control Message Protocol),SPIFFS(Serial Peripheral Interface Flash File System).**

1. INTRODUCTION

In the modern day, the pollution levels in the urban areas has grown so much such that the atmosphere remains toxic almost all throughout the day and night. Highly populated cities like Delhi, Mumbai and Hyderabad the pollution has exceeded the average non-harmful pollution levels, in other terms the pollution levels in these areas is harmful to breathe for a longer duration. People living in these cities who commute for more than 4 to 5 hours during the day time have shown some signs of nausea and breathing problems, and even unconsciousness for some road-side workers. There is a vast research available that supports the deadly effects of the atmospheric pollution to a person. It is also shown that the major contributors to the atmospheric pollution in the urban areas are the vehicles, they produce almost 80% of the entire pollution in the cities. The main pollutants or the particulate matter which is produced from the vehicles is Carbon-Monoxide (CO), 90% of the gas produced after the combustion of petroleum or diesel is Carbon-Monoxide and the rest is a mixture of methane, butane and other compounds. The production of CO gas can be greatly reduced by improving the engine efficiency and getting the vehicle serviced once in every 2 months. Most of the vehicle owners who own and have been using their vehicle for more than 3 years don't usually go for vehicle checkup or a service, thus making their vehicles engines produce more pollution than usual. To control and manage the pollution of a large number of vehicles an IoT device can be used to measure the pollution levels produced by a vehicle and monitor them remotely on a cloud-based database. The pollution data from each vehicle can be stored in separate tables and can be monitored real-time. The data can be used to alert the user if they exceed certain pollution level, also the data can be sent to the RTO office where the vehicle has been registered and challans or fines can be issued on the vehicle owner.

2. ESP 8266 NODE MCU

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WIFI network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

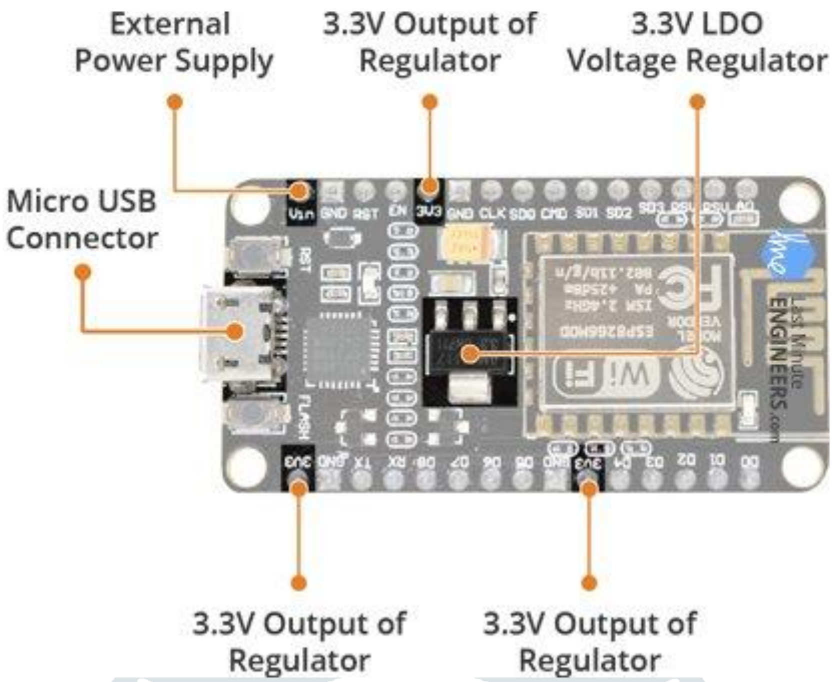

Figure 1 – ESP8266 Node MCU

3.MQ 7 SENSOR

The MQ7 is a simple-to-use Carbon Monoxide (CO) sensor suitable for sensing CO concentrations in the air. It can detect CO-gas concentrations anywhere from 20 to 2000ppm. MQ7 is a high sensitivity to carbon monoxide and stable and long-life span.Sensitive for carbon monoxide output voltage boosts along with the concentration of the measured gases increases. Fast response and recovery to changes in the environment. It has adjustable sensitivity and signal output indicator.

- Power: 2.5V ~ 5.0V
- Dimension: 40.0mm * 21.0mm
- Mounting holes size: 2.0mm

In the case of working with a MCU:
VCC ↔ 2.5V ~ 5.0V,GND ↔ power supply ground,AOUT ↔ MCU.IO (analog output) ,DOUT ↔ MCU.IO (digital output)

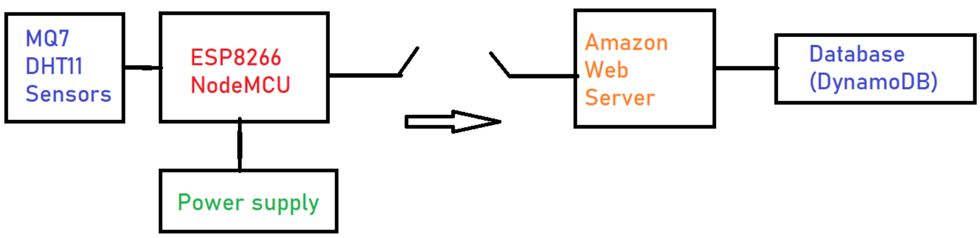

Figure 3 – Process through web server

3. DHT11 SENSOR

The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.

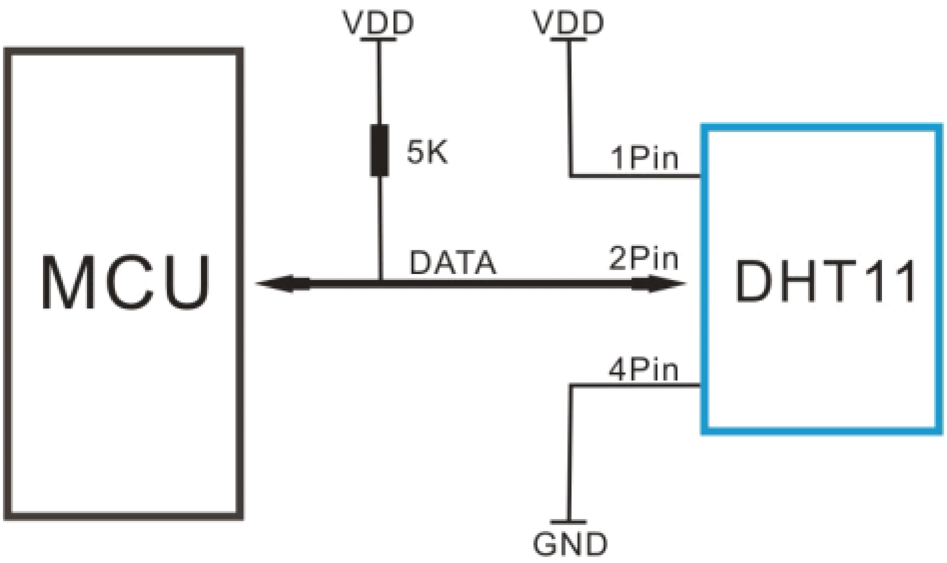

Fig 4 –Connection between MCU and DHT11

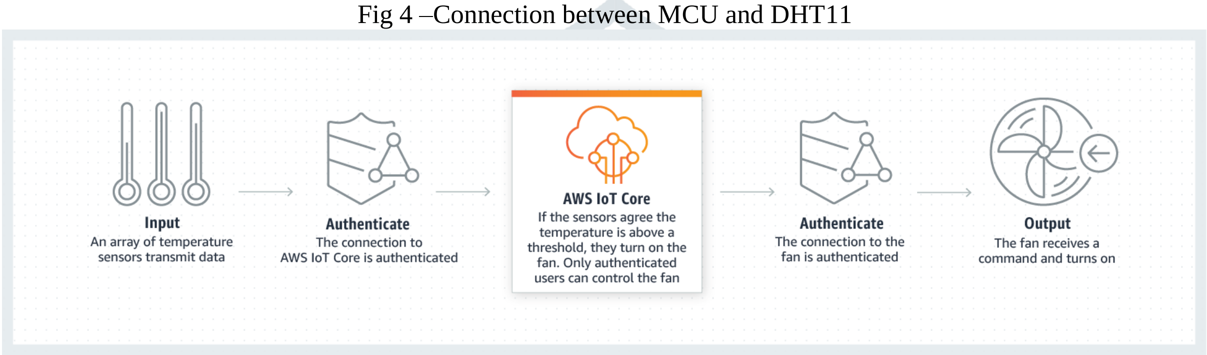

Fig 5 – AWS IoT Core

4. AWS IOT CORE

AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT Core can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely. With AWS IoT Core, your applications can keep track of and communicate with all your devices, all the time, even when they aren't connected.

AWS IoT Core also makes it easy to use AWS and Amazon services like AWS Lambda, Amazon Kinesis, Amazon S3, Amazon SageMaker, Amazon DynamoDB, Amazon CloudWatch, AWS CloudTrail, Amazon QuickSight, and Alexa Voice Service to build IoT applications that gather, process, analyze and act on data generated by connected devices, without having to manage any infrastructure.

AWS IoT Core working

- Connect and manage devices
- Secure device connections and data
- Process and act upon device data
- Read and set device state at any time
- Cost-effectively scale to hundreds of millions of Alexa Built-in devices
- AWS IoT Core allows you to easily connect any number of devices to the cloud and to other devices. AWS IoT Core supports HTTP, WebSockets, and MQTT, a lightweight communication protocol specifically designed to tolerate intermittent connections, minimize the code footprint on devices, and reduce network bandwidth requirements. AWS IoT Core also supports other industry-standard and custom protocols, and devices can communicate with each other even if they are using different protocols.

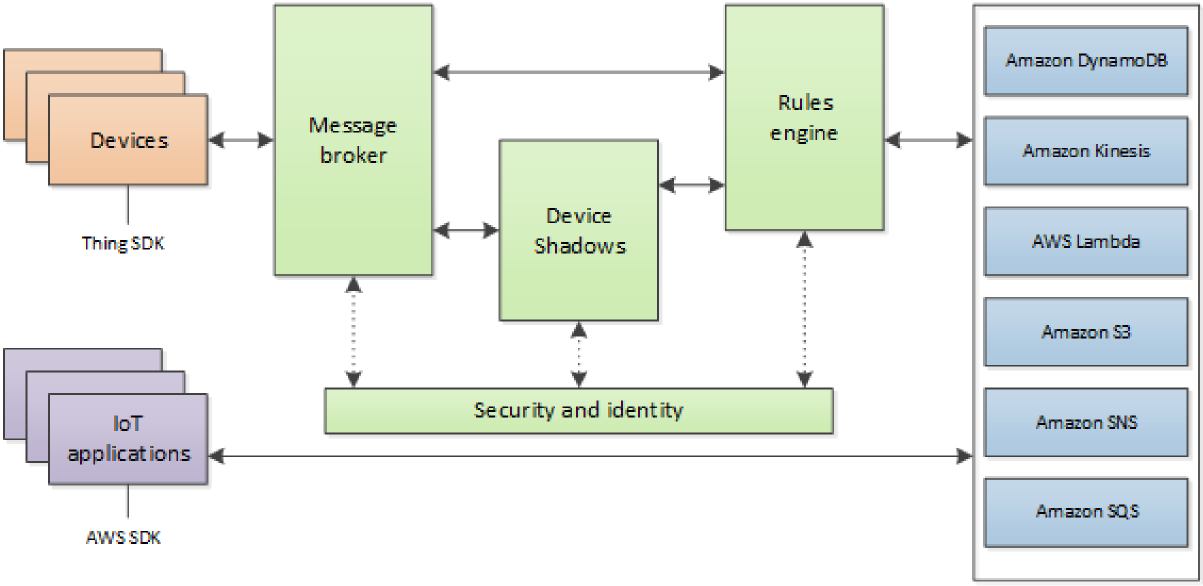

Figure 6 – AWS

Setting up the cloud

The following few steps will set up the cloud for receiving and storing the data in the DynamoDB database and also perform actions when a rule is triggered.

Steps involved in this set up:

1. Setting up Amazon DynamoDB
2. Creating a rule to fetch data from the MQTT topic
3. Send the data to DynamoDB

5. ALGORITHM

1.const char* ssid = "WIFI_SSID";
2.const char* password = "WIFI_PASSWORD";
3.const char* AWS_endpoint = "a3dia9y5nk43jy-ats.iot.us-east-1.amazonaws.com"; (MQTT broker ip)
4.set MQTT port number to 8883 as per //standard
5.connecting MQTT ,ESPthing,MQ7Topic.
6.initialize digital pin LED_BUILTIN as an output.
7.Load certificate file(uploaded file name)
8.WiFi.macAddress(mac);
9.mqttClient.publish("MQ7Topic", msg);
10.Turn the LED on (HIGH is the voltage level)
11.Turn the LED off by making the voltage LOW

6. MQTT

The latest standard is Stan MQTT v5.0 is an OASIS Standard. MQTT v3.1.1 is an older ISO and OASIS Standard.
MQTT-SN v1.2, formerly known as MQTT-S, is available. MQTT for Sensor Networks is aimed at embedded devices on non-TCP/IP networks, such as Zigbee. MQTT-SN is a publish/subscribe messaging protocol for wireless sensor networks (WSN), with the aim of extending the MQTT protocol beyond the reach of TCP/IP infrastructure for Sensor and Actuator solutions.

Usage of MQTT protocol

The basic usage of MQTT protocol is for sending and receiving messages between a client and a server, the messages are called payloads which are in the form of web packets which are in a JSON format.

JSON Policy Syntax

The policies are in the form of JSON format, the following identity-based policy allows the implied principal to list a single Amazon S3 bucket named example_bucket:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
```

7. RESULTS

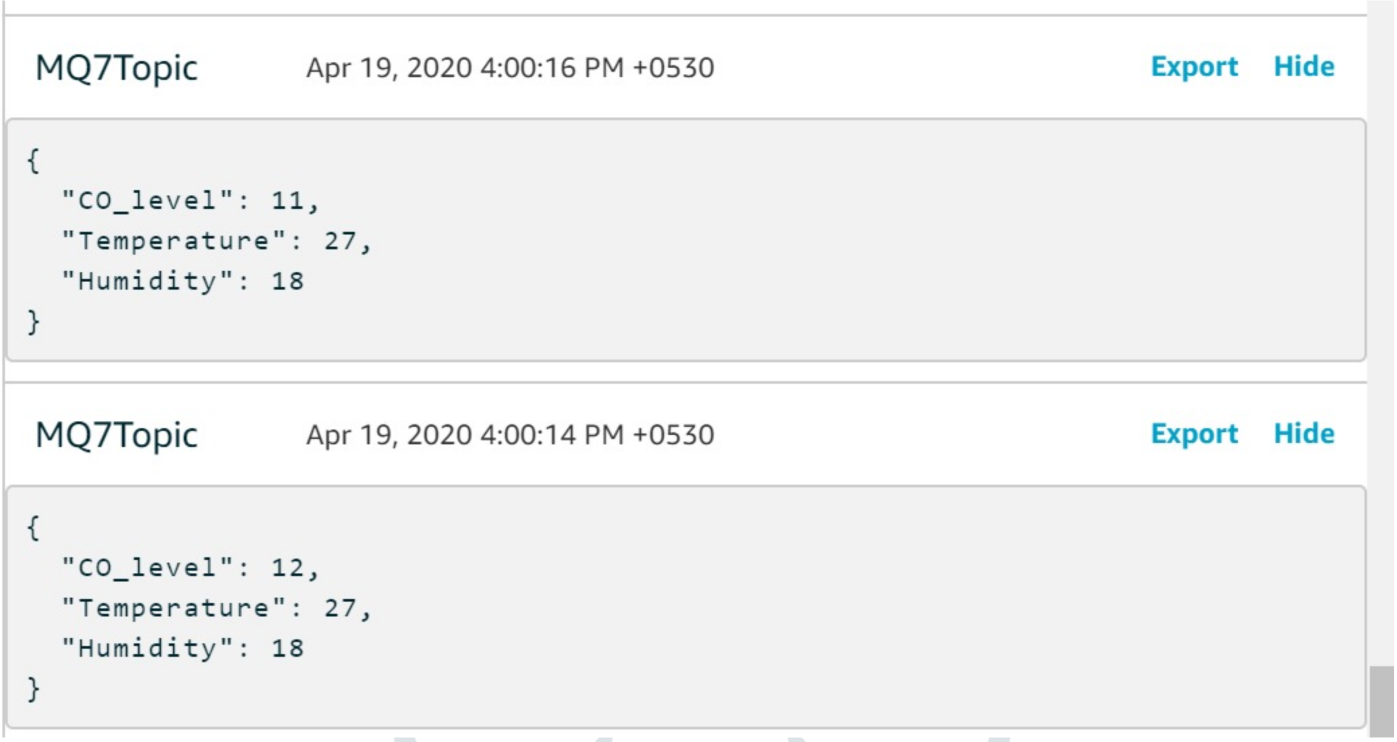

Figure 10- CO-Level

ACKNOWLEDGEMENT

We would like to thank our Pro Vice Chancellor- Dr.N Siva Prasad, Head of the Department- Dr.K Manjunathachari and our respected guide-Mr Chandra Shekar, who have helped us in making this project. Without their support, this project would not have been made possible.

I would also like to thank the different resources and papers which acted as a guide in helping us do the project.

REFERENCES

[1] International Conference on Pollution Control & Sustainable Environment, April 25-26, 2016 Dubai, UAE. http://www.omicsonline.org/conferences-list/effects-of-pollution.

[2] AWS IOT,docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html,© 2016, Amazon Web Services.

[3] Wan-Young Chung, Sung-Ju Oh, Remote monitoring system with wireless sensors module for room environment, Sensors and Actuators B: Chemical, Volume 113, Issue 1, 17 January2006, Pages 64-70, ISSN 0925-4005, 10.1016/j.snb.2005.02.023.

[4] Prakash Doraiswamy, Wayne T. Davis, Terry L. Miller, Joshua S. Fu and Yun-Fat Lam. 2005. Measuring Air Pollution Inside And Outside of Diesel Truck Cabs. Report prepared for US EPA by University of Tennessee, Knoxville TN.http://www.epa.gov/smartway/documents/publications/in cabairquality-110405.pdf.

[5] Tsow, F.; Forzani, E.; Rai, A.; Rui Wang; Tsui, R.; Mastroianni, S.; Knobbe, C.; Gandolfi, A.J.; Tao, N.J., "A Wearable and Wireless Sensor System for Real-Time Monitoring of Toxic Environmental Volatile Organic Compounds," Sensors Journal, IEEE , vol.9, no.12, pp.1734,1740, Dec. 2009. DOI= 10.1109/JSEN.2009.2030747.

[6] C. A. Trasviña-Moreno, R. Blasco, R.Casas, and A. Marcus.Autonomous WiFi Sensor for Heating Systems in the Internet of Things