IoT Based Indoor Air Quality Monitoring System

Department of
ELECTRONICS & TELECOMMUNICATION ENGINEERING

by

Hiten Bafna and Mitash Shah

Under the guidance of
Prof. Sneha Weakey
Prof. Pallavi Malame

Department of Electronics & Telecommunication Engineering Sardar
Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058

CERTIFICATE

This is to certify that the Mini Project entitled “IoT based Indoor Air Quality Monitoring System" has been completed successfully by Hiten S Bafna & Mitash A Shah under the guidance of Prof. Sneha Weakey for the second year term of the department Electronics & Telecommunication Engineering

Certified by

Prof. Sneha Weakey
Project Guide

Prof. Pallavi Malame
Examiner

Department of Electronics & Telecommunication Engineering
Sardar Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058
2020-2021

Abstract

Air pollution is now considered to be the world's largest environmental health threat, accounting for 7 million deaths around the world every year. Air pollution causes and exacerbates a number of diseases, ranging from asthma to cancer, pulmonary illnesses and heart disease. The MQ135 sensor is used to measure Indoor Air Quality, and the MQ7 sensor is used to measure Carbon Monoxide, CO. We're trying to put together a system that uses IoT platforms like Thingspeak to raise awareness about the environmental damage we're doing. New Delhi has already been dubbed the world's most polluted city, with air quality levels exceeding 300 parts per million. We have attempted to make a light & cheap model more suitable for indoor air quality analysis.

Acknowledgement

We have great pleasure in presenting the report on Iot Based Indoor Air Quality Monitoring Sytem . We take this opportunity to express my sincere thanks towards our guide, Assistant Professors & Head of Department of Electronics & Telecommunication Engineering, S.P.I.T., Mumbai, for providing the technical guidelines and the suggestions regarding line of this work. We would like to express our gratitude towards their constant encouragement, support and guidance throughout the development of the project.

We also thank all the staff of S.P.I.T., Mumbai for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues at S.P.I.T., Mumbai for their encouragement.

Hiten Santosh Bafna
Mitash Ashish Shah

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Air pollution has become a common phenomenon everywhere. Air pollution is a major issue, especially in urban areas. The rising usage of diesel and petrol automobiles in metropolitan areas, as well as the presence of industrial zones on the outskirts of major cities, are the primary sources of air pollution. In metropolitan areas, the problem has become much worse. This necessitates the measurement and analysis of real-time air quality monitoring in order to make proper decisions in a timely manner. This paper describes a stand-alone air quality monitor that may be used in real time.

The Internet of Things (IoT) is now widely used in almost every industry, and it plays an important role in our air quality monitoring system as well. The setup will display the air quality in PPM on a web page, allowing us to conveniently monitor it. You can monitor the pollution level using your computer or mobile device with this IoT project.

Chapter 2

Literature Review

Air pollution in major cities has a huge effect on people and the environment. India's environmental problems are rapidly worsening. Vehicles and industries are the primary sources of air pollution, which contribute to respiratory disorders such as asthma and sinusitis. Due to a huge amount of carbon dioxide, the air quality in metropolitan cities such as Kolkata, Delhi, and Mumbai is poor A significant number of projects that use lowcost air pollution detecting devices that can be carried by individuals have been detailed in the literature. This paper is inspired by the idea mentioned at [1] with less cost i.e. while pushing the data to the cloud server, there is no need to display the data on lcd, it increases the cost of the project [2] .Since this is an IoT project, our idea was to present all the data obver internet and hence compare & analyse it over the IoT platform, ThingSpeak .

Table no. 1
AIR QUALITY INDEX

Range(PPM)	Status
0-50	Good
51-100	Moderate
100-150	Unhealthy for sensitive groups
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

As evident from the Table 1 cited at [3], it explains about the Air Quality Index ranges, as seen in Table 1 mentioned at [3]. To begin with, a concentration of 0-50 PPM is considered completely safe. Moderate is defined as 51-100 PPM, which is typically found in high-traffic locations .For sensitive individuals concentrations of 100-150 PPM is unhealthy. Above 151 PPM [3] is completely unsafe or unhealthy where India's capital New Delhi falls in this range. Its very rare to record 300 PPM and above which can be considered as Hazardous, possibly due to Coal gas in mines.
This paper used ESP8266 WiFi to push the data onto the cloud rather using GSM or GPRS module [4]. Another paper proposes an open platform of a WiFi-enabled indoor air quality monitoring and control system, which could be incorporated into such a ‘smart building’ structure. The proposed system operates over an existing WiFi wireless network utilizing the MQTT protocol [5]

Another paper describes the development of a wireless monitoring system which can be deployed in a building. The system measures carbon dioxide, carbon monoxide and temperature. The system developed in this paper can serve as the monitoring component of a HVAC control system and function as an indoor air quality monitor independently.[6]. This [7] paper presents Polluino, a system for monitoring the air pollution via Arduino. Moreover, a Cloud-based platform that manages data coming from air quality sensors is developed.

Objective of this [8] paper is to present a system model which can facilitate the assessment of health impacts caused due to indoor air pollutant as well as outdoor and can intimate the human prior about the risk he/she going to have, here a sensing network based microcontroller equipped with gas sensors, optical dust particle sensor, humidity and temperature sensor has been used for air quality monitoring. IoT [9][10] encompasses regular objects ("Things") that have network availability, permitting them to send and get information. Things include people, information, software agents, or any other virtual participating actors. There are four kinds of Things utilized in this paper:

1) ArduinoUno [11] (Microcontroller)
2) ESP8266 (Wi-Fi Module)
3) cloud service(ThingSpeak)
4)Gas Sensors

These Things are coordinated to make a framework such that each Thing can work individually and can gather, store, and recover information to address the problem.

Chapter 3

Project Objectives

The main aim of this project is

- to develop a system which can monitor PPM in air in real time,
- tell the quality of air and
- log data to a remote server.
- to hence combine advanced detection technologies to produce an air quality monitoring system with advanced capabilities to provide low cost comprehensive monitoring

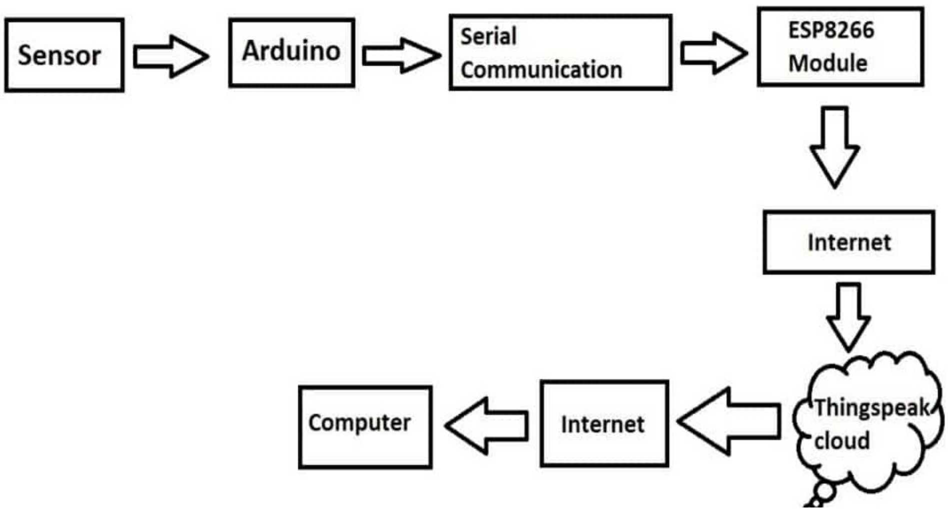

Fig. 1. Block Diagram of the Model

Chapter 4

Methodology

Calculations:

The most crucial step is to calibrate the sensor in fresh air and then create an equation that transforms the sensor output voltage value into our usable units of PPM (parts per million). Here are the results of the mathematical computations that are cited [4] .

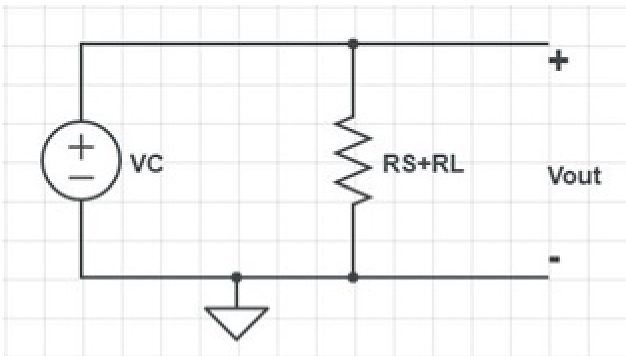

Fig. 2. Internal circuit diagram of MQ135

Fig 2 is Internal circuit diagram of MQ135 sensor Rs and Rl combined From Ohm's Law, at constant temperature, we can derive I as follows:

From Ohm's Law, we can derive I as follows:

$$I = \frac{V}{R} \tag{1}$$

From fig 2, equation 1 is equivalent to

$$I = \frac{V_c}{R_s + R_l} \tag{2}$$

We can calculate the output voltage at the load resistor using the value obtained for I and Ohm's Law at constant temperature. V = I*R

$$V_{R_l} = \left[\frac{V_c}{R_s + R_l} \right] * R_L \tag{3}$$

$$V_{R_l} = \left[\frac{V_c * R_l}{R_s + R_l} \right] \tag{4}$$

So now we solve for R_S:

$$V_{R_l} * (R_s + R_l) = V_c * R_L \tag{5}$$

$$(V_{R_l} * R_s) + (V_{R_l} * R_l) = V_c * R_L \tag{6}$$

$$V_{R_l} * R_s = (V_c * R_l) - (V_{R_l} * R_l) \tag{7}$$

$$R_s = \frac{(V_c * R_l) - (V_{R_l} * R_l)}{V_{R_l}} \tag{8}$$

$$R_s = \frac{(V_c * R_l)}{V_{R_l}} - R_l \tag{9}$$

Equation 9 help us to find the internal sensor resistance for fresh air [13].

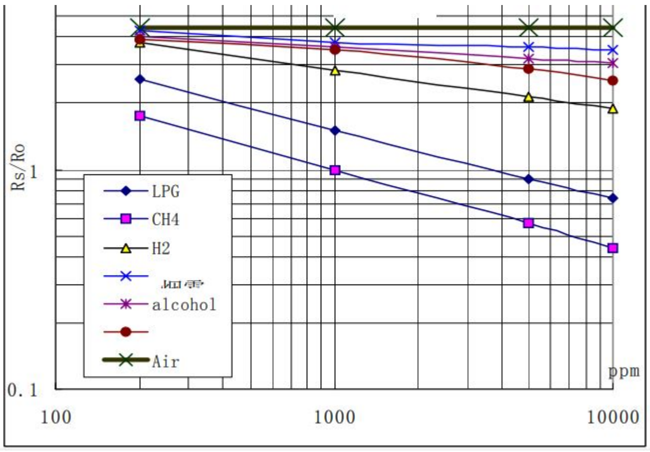

Fig. 3. MQ135 Data sheet- Change in Resistance vs change in PPM

From the graph shown in fig 3, we can see that the resistance ratio in fresh air is a constant [13]:

$$Rs/R0 = 3.6 \tag{10}$$

Value 3.6 which is mentioned at Equation 10 is obtained from the datasheet shown in Fig 3. We'll need to determine the value of the Rs in fresh air to calculate R0. This will be accomplished by converting the analog average readings from the sensor to voltage [13]. Then we will apply the Rs formula to calculate R0. First and foremost, the lines will be treated as if they were linear. This way, we may use a single formula to relate the ratio and the concentration linearly [14]. By doing so, we can find the concentration of a gas at any ratio value even outside of the graph's boundaries. The formula we will be using is the equation for a line, but for a log-log scale. The formula for a line is: From above Figure 3, we try to derive the following calculations.

$$y = mx + b \tag{11}$$

For a log-log scale, the formula looks like this:

$$log_{10}y = m * log_{10}x + b \tag{12}$$

Let's find the slope. In order to accomplish so, we must select two points from the graph. In our situation, the points (200,2.6) and (10000,0.75) on the LPG (Liquid Petroleum Gas) line from fig 3 were chosen. The LPG line is the outcome of a sensor that was being tested with different levels of LPG as input. The following is the formula for calculating slope m(here) [13]:

$$m = \frac{logy - log(y_0)}{logx - log(x_0)} \tag{13}$$

If we apply the logarithmic quotient rule we get the following:

$$m = \frac{log(y/y_0)}{log(x/(x_0)} \tag{14}$$

Now we substitute the values for x, x_0, y, and y_0:

$$m = \frac{log(0.75/2.6)}{log(10000/200)} \tag{15}$$

$$m = -0.318 \tag{16}$$

Now that we have m, we can calculate the y intercept. To do so, we need to choose one point from the graph (once again from the CO2 line). In our case, we chose (5000,0.9) [13]

$$log(y) = m * log(x) + b \tag{17}$$

$$b = log(0.9) - (-0.318) * log(5000) \tag{18}$$

$$b = 1.13 \tag{19}$$

Now that we have m and b, we can find the gas concentra-tion for any ratio with the following formula:

$$log(x) = \frac{log(y) - b}{m} \tag{20}$$

However, we must find the inverse log of x in order to obtain the real value of the gas concentration based on the log-log plot:

$$x = 10^{\frac{log(y)-b}{m}} \tag{21}$$

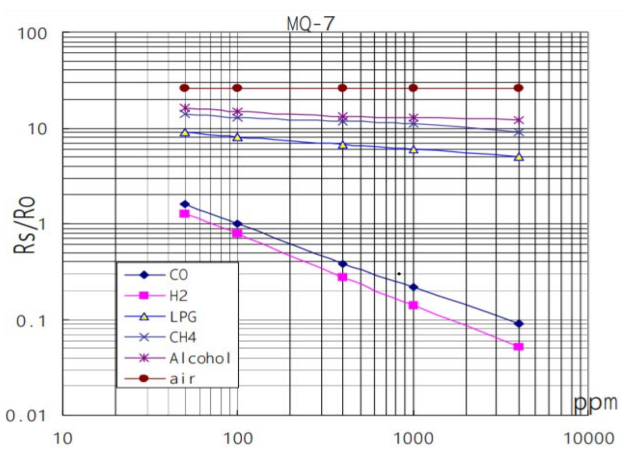

Fig. 4. MQ7 Data sheet- Change in Resistance vs change in PPM

After calibration of the sensors, we get the R0 values for both the sensors which we use in the code to sense the quality of air in real time and log the collected data to a remote server by sending AT commands from ESP8266 module. The module first test the AT startup by sending the following command - AT.

The command is passed by the controller to the Wi-Fi module using software serial function. In response to the command 'AT', the platform must respond with 'OK' if the cloud service is running. Then, the AT command to view the version information is passed as follow - AT + GMR

In response to this command, the IOT platform must respond by sending back the version information, sdk version and the time bin is compiled. Next, the AT command to set the connection to Wi-Fi mode is sent as follow - AT + CWMODE = 1

By setting the parameter in CWMODE to 3, the Wi-Fi connection is configured to SoftAP as well as station mode. This AT command can in fact take three parameters as follow -

- set Wi-Fi connection to station mode
- set Wi-Fi connection to SoftAP mode
- set Wi-Fi connection to SoftAP + station mode

In response to this command, the IOT platform must send back the string indication the Wi-Fi connection mode set. Now the AT command to reset the module is sent as follow - AT + RST

In response to this command, the Wi-Fi module must restart and send back a response of 'OK'. After resetting the module, AT command to setup multiple connections is enabled by sending the following command - AT + CIPMUX=1

This AT command can take two parameters - 0 for setting single connection and 1 for setting multiple connections. Next, the command to connect with the Access Point (AP) is passed which takes two parameters where first parameter is the SSID of the registered cloud service on ThingSpeak and the other parameter is the password to login the cloud service.

AT+CWJAP="SSID","PASS\”

Now, the AT command to get local IP address is passed as follow - AT + CIFSR

In response to this command, the local IP address of the Wi-Fi connection is sent back by the module. Now, the module is ready to establish TCP IP connection with the ThingSpeak server. The controller reads the sensor data and store it in a string variable. The TCP IP connection is established by sending the following AT command - AT + CIPSTART = 4, "TCP", "184.106.153.149", 80

The AT + CIPSTART command can be used to establish a TCP connection, register an UDP port or establish an SSL connection. In the above command, it is used to establish a TCP IP connection. For establishing a TCP-IP connection, the command takes four parameters where first parameter is link ID which can be a number between 0 to 4, second parameter is connection type which can be TCP or UDP, third parameter is remote IP address or IP address of the cloud service to connect with and last parameter is detection time interval for checking if the connection is live. If the last parameter is set to 0, the TCP keep-alive feature is disabled otherwise a time interval in seconds range from 1 to 7200 can be passed as parameter. In response to this command, the server must respond with 'OK' if connection is successfully established otherwise it should respond with message 'ERROR'.

Now when the connection with the server is successfully established and the controller has read the sensor value, it can send the data to the cloud using the following command - AT + CIPSEND = 4

SENSING AND MONITORING OPERATION FLOW

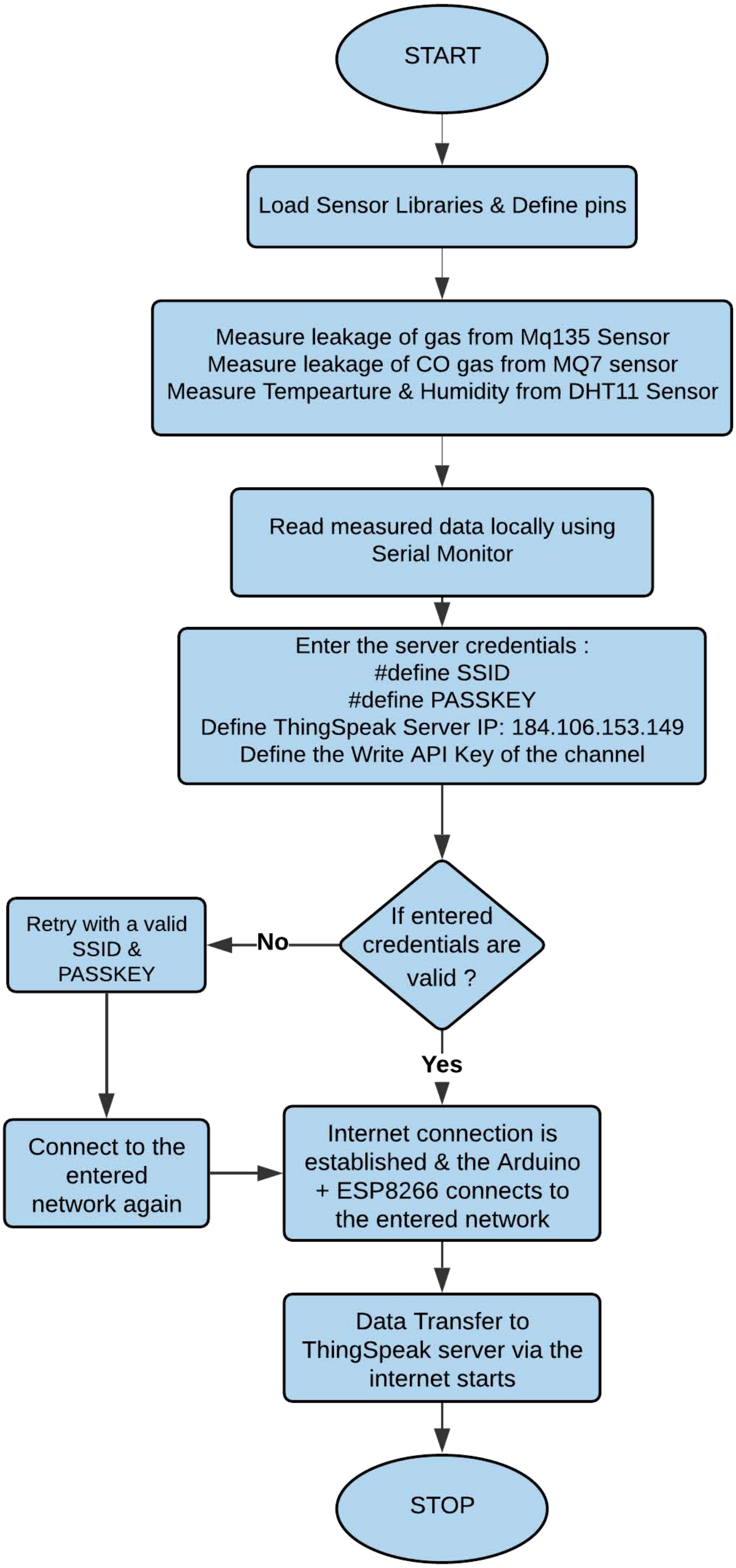

Fig. 5. Flowchart of Sensing & Monitoring Operation

Chapter 5

System Design & Architecture

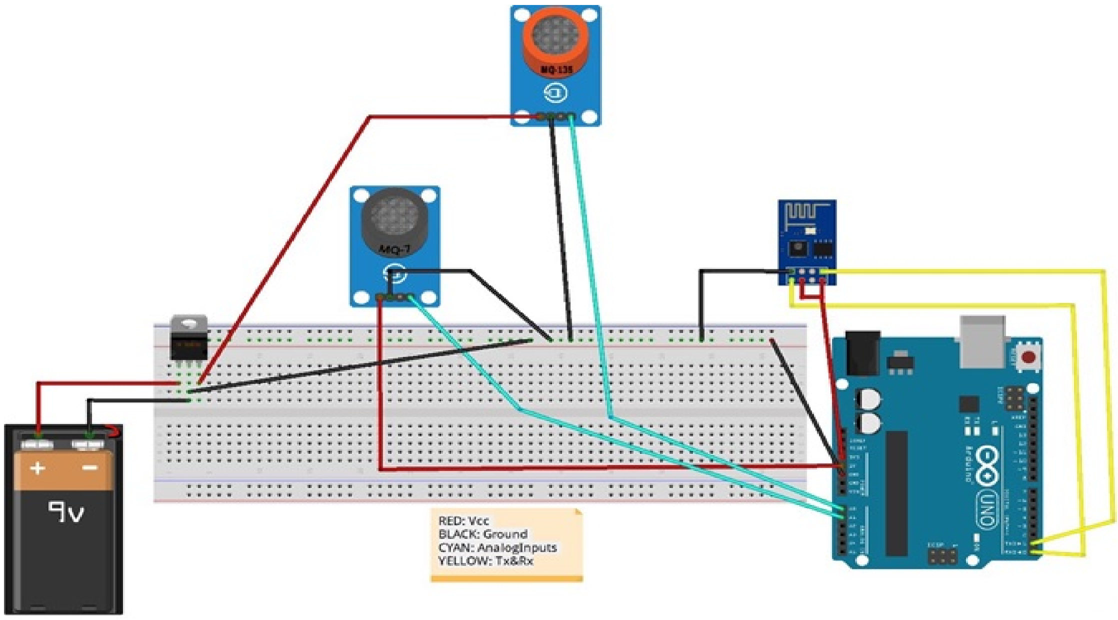

Fig. 6. Connections Sketch

The Arduino Uno Development Kit, which includes an ATMega328P microprocessor, was used. We have used a low-cost ESP8266(ESP-01) WiFi module to give WiFi support for it, which allows us to connect to the ThingSpeak Platform. The connections between the components used, such as the Arduino Uno, MQ135, MQ7, ESP-01 Wifi Module, 9Volts Battery, and LM7805 Regulator, are shown in Figure above.The ESP8266 is connected to the Arduino Uno's 3.3V pin. MQ135 is wired to the Arduino Uno's 5V pin. As there is not enough power to drive another sensor, MQ7 is connected to a 9V battery via a 5V LM7805 Regulator. By providing the relevant SSID and Password, the ESP-01 is connected to the Local Hotspot. The reason for using LM7805 Regulator is that 9Volts supply should not be directly given to MQ7 sensor where it needs only 5Volts input at maximum, so regulator does the job of stepping 9Volts to 5Volts.

Arduino Uno:-

Product Description:

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board micro controllers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.

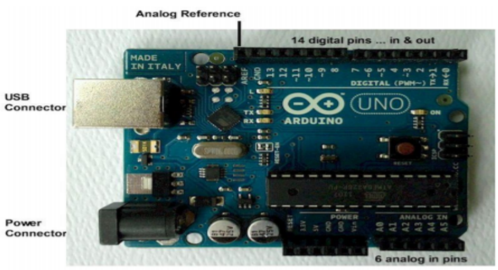

Fig. 7. Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P It has 14 digital input/ output pins 6 analog inputs, a 16 MHz quartz crystal, a USB Connection, power jack, an ICSP header and a reset button. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

Pin Description:

Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Table 2: Pin Description of Arduino Uno

Technical Specification:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 3: Technical Specification of Arduino Uno

MQ7 CO Sensor:-

Product Description:

MQ7 is a Carbon Monoxide (CO) sensor, suitable for sensing Carbon Monoxide concentrations(PPM) in the air. This sensor has a high sensitivity and fast response time. The sensor's output is an analog resistance.

Fig. 8. MQ7 Sensor

Air Quality Sensor (MQ135):-

Product Description:

Ammonia (NH3), nitrogen oxides (NOx), benzene, smoking, CO2, and other toxic or dangerous gases that affect air quality can all be detected with an air quality sensor. Tin dioxide (SnO2), an inorganic substance with lower conductivity in clean air than when polluting gases are present, is used in the MQ-135 sensor unit.

Fig. 9. MQ135 Sensor

Pin Description of both the sensors:

- 1: the VCC power supply 5V DC
- 2: GND , used to connect the module to system ground
- 3: DIGITAL OUT, You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
- 4: ANALOG OUT, This pin outputs 0-5V analog voltage based on the intensity of the gas.

ESP8266 (ESP-01) WiFi Module:-

Product Description:

The ESP-01 ESP8266 Serial WIFI Wireless Transceiver Module is a self-contained SOC with an integrated TCP/IP protocol stack that can provide access to your WiFi network to any microcontroller. The ESP8266 may either host an application or offload all Wi-Fi networking functionality to a separate application processor. The Wi-Fi module can be connected with the any local server by sending AT commands from the module.

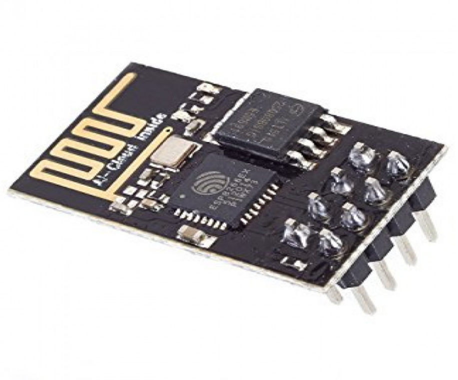

Fig. 10. ESP8266 WiFi Module

Pin Configuration:

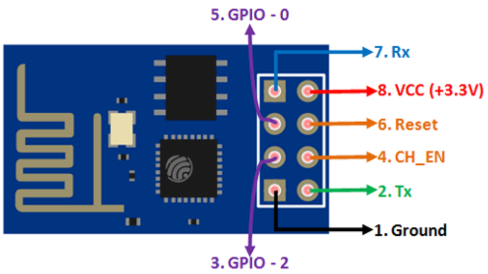

Fig. 11. ESP8266 Pin out

LM7805 Voltage Regulator:-

Product Description:

The 7805 IC is a well-known regulator IC that is used in a wide range of projects. The designation 7805 has two meanings: "78" indicates that it is a positive voltage regulator, and "05" indicates that it outputs 5V. As a result, our 7805 will output a value of +5V. This IC has a maximum output current of 1.5A.

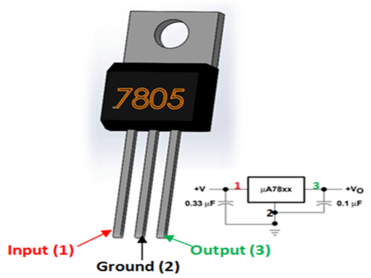

Fig. 12. LM7805 Voltage Regulator

Temperature and humidity sensor (DHT11):-

Product Description:

The DHT11 is a temperature and humidity sensor that is widely used. The sensor includes a dedicated NTC for temperature measurement and an 8-bit microprocessor for serial data output of temperature and humidity values. The sensor is factory calibrated, making it simple to connect to other microcontrollers. The sensor can detect temperatures ranging from 0°C to 50°C, as well as humidity levels ranging from 20% to 90% with an accuracy of ±1°C and ±1%.

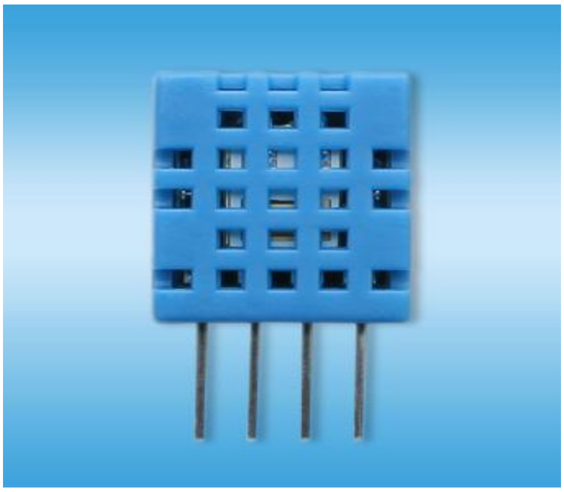

Fig.13. DHT11 Sensor

Pin Description:

- 1, the VDD power supply 3.5～5.5V DC
- 2 DATA serial data, a single bus
- 3, NC, empty pin
- 4, GND, used to connect the module to system ground

Software Used:

- Arduino IDE
- thingspeak.com

Chapter 6

Simulation & Experimental Results

Calibrating the sensors by finding R0 :
(**Note :** Sensors were calibrated in fresh air)

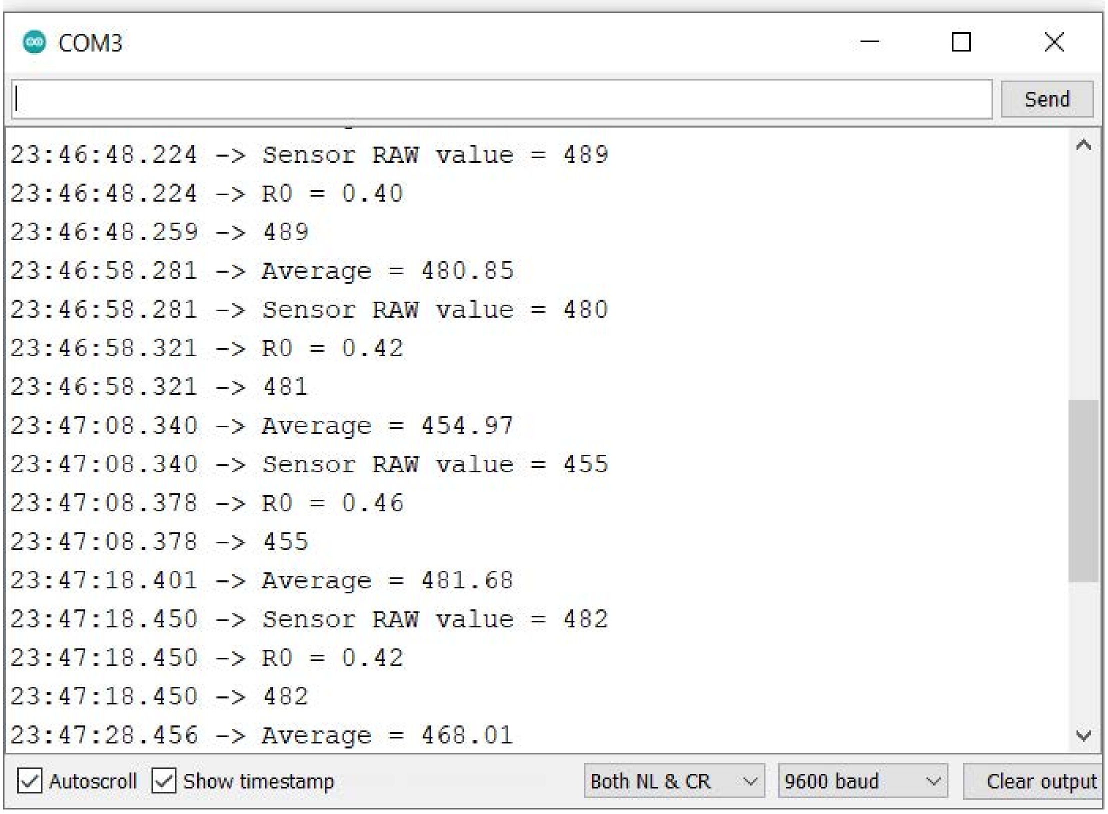

Fig. 14. R0 value of MQ7 sensor

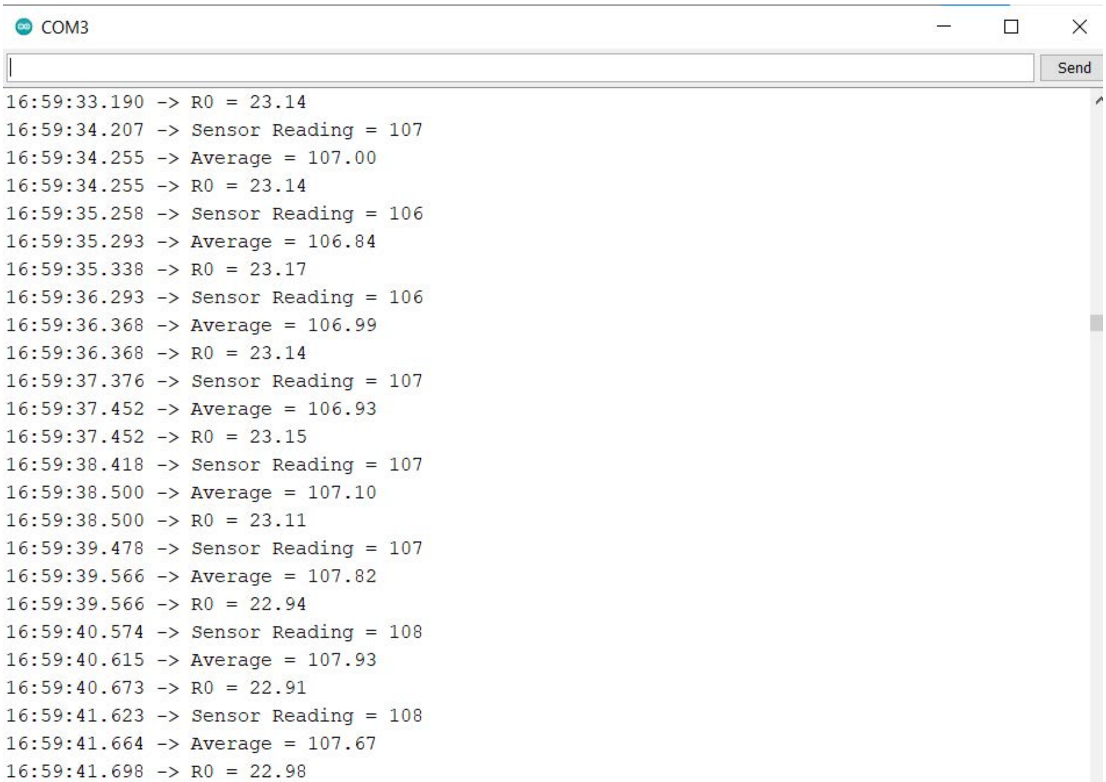

Fig. 15. R0 value of MQ135 sensor

The MQ135 Sensor is used to sense air quality, which is affected by pollutants such as carbon dioxide, sulphur dioxide, nitrogen oxide, and smoke. The fluctuation of CO2 vs time as measured by the MQ135 Sensor in a real-time environment is shown in Fig. The concentrated incense stick smoke in the air during the experiment caused the fluctuation. The fluctuation of carbon dioxide in two separate places is depicted in Fig.16 & 17. The level of carbon dioxide present in the environment of an incense stick's smoke in the air, may be seen in Fig. 16. The amount of carbon dioxide present in the society compound when evaluated in a less crowded location and also in a normalized manner is depicted in Fig. 17.
From the plot shown in Fig. 16, we can easily identify the range of amount of CO2 present in that particular environment. The values of air quality is between 40 to 60 PPM which is moderate but are very high levels with respect to the levels when tested in the society which is around 0.15 to 1.5 PPM. The values of CO ranges from 2-10 PPM which is a little high but people will not suffer from any adverse effects. On the other hand levels of CO in other environment are around 0.8-0.9 PPM. The temperature in the fresh environment is 30°C and humidity is around 85-86 % while in the smoky environment the temperature increase to around 33°C and humidity decreases to 77 %.
We can readily observe the data for each data point over time by looking at the plot. As a result, drawing conclusions from this data is simple. We can see that the Air Quality values were at their greatest at some times, but that they dropped down and changed a lot at first. This occurred in parallel with a drop in temperature, which is an important thing to note.

Fig. 16. Concentrated Incense Stick smoke Environment

Entries: 59

Fig. 17. Graph showing Air Quality in the society compound

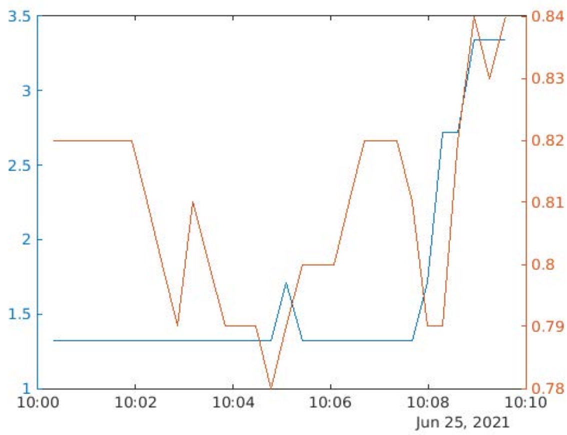

Fig. 18. 2-D line plots with y-axes for MQ135 on left & MQ7 on right in the society compound

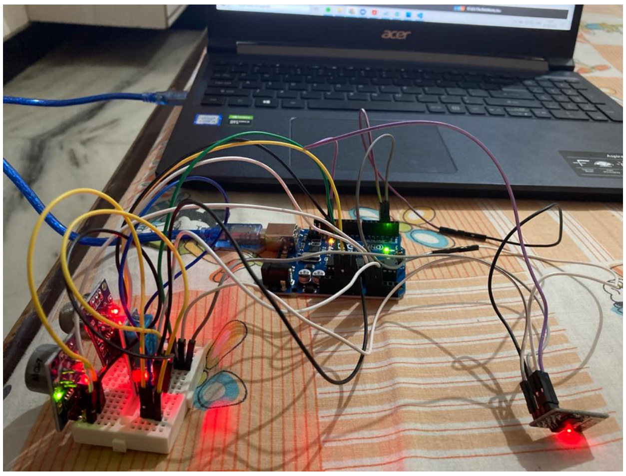

Fig. 19. Hardware Setup

Chapter 7

Conclusions

In this study, an IoT-based air quality monitoring system is used to study air pollution in various indoor environments. Furthermore, codes were written for various mathematical formulae that are applied to effortlessly monitor data on the ThingSpeak platform. Different sensors such as (i) MQ135 Sensor (ii) DHT11 as a temperature and humidity sensor, (iii) MQ-7 Sensor (for Carbon Monoxide) are used for the project. The Internet of Things concept enables full data assimilation, revealing important information about air quality, temperature and humidity levels, and the presence of various gases in various environments. The information relevant to the environment under study for air quality monitoring can be accessed through the cloud server that was developed. Even the smoke from an incense stick can deteriorate the quality of the air and create respiratory difficulties. Because high indoor humidity makes the air too dry, those with diseases like whizzing will be more impacted, and it will be difficult for normal people to stay in that environment. ThingSpeak platform is highly useful to analyze the Cloud data from different environments. As a result, we were able to determine Air Quality in PPM. The disadvantage of the MQ135 sensor is that it cannot indicate the quantity of Carbon Monoxide or Carbon Dioxide in the atmosphere, but the advantages are that it can detect smoke, CO, CO_2, and NH_4. So, we used a CO (Carbon Monoxide) MQ7 sensor solely to tell the level of specific gases. This project is suitable for both indoor and outdoor use. We can develop this kit as a compact device for inside use so that if every home uses it, we can monitor the indoor air quality of a certain targeted area. Due to rising air pollution, it is also necessary to monitor indoor air quality. However, one sensor is insufficient for monitoring outdoor air quality because one sensor has a sensitivity range of roughly a meter, hence a network of sensors must be placed. While calibrating the sensors, great care was taken.

Bibliography

1. Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G., ... & Gavras, A. (2014, March). Enabling reliable and secure IoT-based smart city applications. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS) (pp. 111-116). IEEE.
2. Informatics, 10(2), 1596-1605.Zheng, K., Zhao, S., Yang, Z., Xiong, X., & Xiang, W. (2016). Design and implementation of LPWA-based air quality monitoring system. IEEE Access, 4, 3238-3245.
3. Ahlgren, B., Hidell, M., & Ngai, E. C. H. (2016). Internet of things for smart cities: Interoperability and open data. IEEE Internet Computing, 20(6), 52-56
4. Shah, J., & Mishra, B. (2016, January). IoT enabled environmental monitoring system for smart cities. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 383-388). IEEE.
5. Xiaoke Yang, Lingyu Yang, Jing Zhang (School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China) A WiFi-enabled indoor air quality monitoring and control system. In Control & Automation (ICCA), 2017 13th IEEE International Conference
6. R du Plessis, A Kumar, GP Hancke (Department of Electrical, Electronic and Computer Engineering, University of Pretoria, SouthAfrica). A wireless system for indoor air quality monitoring. Industrial Electronics Society , IECON 2016 -42nd Annual Conference of the IEEE
7. Polluino: An efficient cloud-based management of IoT devices for air quality monitoring. Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum
8. Sneha Jangid, Sandeep Sharma (School of ICT, Gautam Buddha University, Greater Noida, India) An embedded system model for air quality monitoring. Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference
9. J. A. Stankovi´c, “Research directions for the Internet of Things,” IEEE Internet Things J., vol. 1, no. 1, pp. 3–9, Feb. 2014.
10. D. Uckelmann, M. Harrison, and F. Michahelles, “An architectural approach towards the future Internet of Things,” in Architecting the Internet of Things. Heidelberg, Germany: Springer, 2011, pp. 1 24.
11. M. Schmidt, Arduino: A Quick-Start Guide. Dallas, TX, USA: Pragmatic Bookshelf, 2015.
12. Marques, G., & Pitarma, R. (2016). An indoor monitoring system for ambient assisted living based on internet of things architecture. International journal of environmental research and public health, 13(11), 1152.
13. Systems, J. (2016, May 09). Understanding a Gas Sensor. Retrieved from https://jayconsystems.com/blog/understanding-a- gas-sensor
14. Kang, B., Park, S., Lee, T., & Park, S. (2015, January). IoT-based monitoring system using tri-level context making model for smart home services. In 2015 IEEE International Conference on Consumer Electronics (ICCE) (pp. 198-199). IEEE.
15. Components101.com
16. 2014 :: Datasheetspdf.com