**Acknowledgement**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Prof. N.R.Merchant and our H.O.D Prof. A.R.Chandegra for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & Faculties of Government Polytechnic, Ahmedabad for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to senior friends in developing the project and people who have willingly helped me out with their abilities. And special thanks to our respected principal Shree P.M. Patel sir for inspiring us for developing some innovative. without all this persons this project isn’t possible.

**Index**

Contents

Introduction6

Literature Survey7

Hardware Development8

(i) Block Diagram8

(ii) Circuit Diagram11

(iii) Components List & Specifications15

Software Development18

(i) Server18

(ii) Flow Chart25

**Project Outcomes & Applications26**

Future Expansions29

References30

**Chapter 1:**

**Introduction**

In this era of modernization, technologies are advancing rapidly. Every day we realize some new technology coming in market to simplify our lives more than ever. Back in time checking the pollution in a particular area was a very tedious task which was not very efficient also.

With the increasing pollution and advancing technology various new methods were introduced to keep an eye on the rapid increase in pollution more efficiently. Internet of things is one of the latest works that has been done in this path. The increment in use of internet and the interaction of human with machine gave rise to IOT.

It allows exchange of information among various devices like fridge, washing machine, automobiles, watches etc. This exchange of information takes place with the help numerous sensors. The account for the success of IOT is its efficiency and makes it a feasible technology at low cost. Air pollution Are two main constituents that have the most adverse effect on humans as well as the entire earth. Therefore it is very important to check and control it. Traditional methods involves manual work in which data loggers used to visit the site to collect the data, analyse it and perform comparisons to provide the output which was very lengthy and time consuming besides being inefficient.

The pollution monitoring system involves use of sensors (MQ135) which measures the Air pollution concentration and level of harmful gases like benzene, alcohol, smoke, steam which mainly pollutes the air. Comparisons are done automatically using previously stored data in database and output is stored on cloud to make it accessible from remote areas. This paper involves description of the system that presents its output to the server thingspeak.com which the user can access it whenever they want. It can be used for notifying the authorities.

**Chapter 2:**

**Literature Survey**

The motive of making a smart city can be fulfilled by using technology, thus making the life better and also enhancing the quality of services, therefore meeting every individual's needs.

With modern technology in fields of information and communication, it has become easy to interact with the authorized people of city to tell them where about of the area or city, how well the city is developing and how to make it possible to achieve a better life quality.

In this system, an application was created to make one more step in the fulfilment of the goal. An area is analysed for evaluating how much pollution is affecting the area. The components of gases and their amounts are calculated and checked.

If the amount is higher than normal then the officials are reported about it. After that the people are made to clear the area and taken to a safe place. The combined network architecture and the interconnecting mechanisms for the accurate estimation of parameters by sensors are being explained and delivery of data through internet is presented.

Some of the research work made for monitoring the pollution parameters in a particular location in order to make the environment safe and that area smart. Different methods were used in the past and are described in this section.

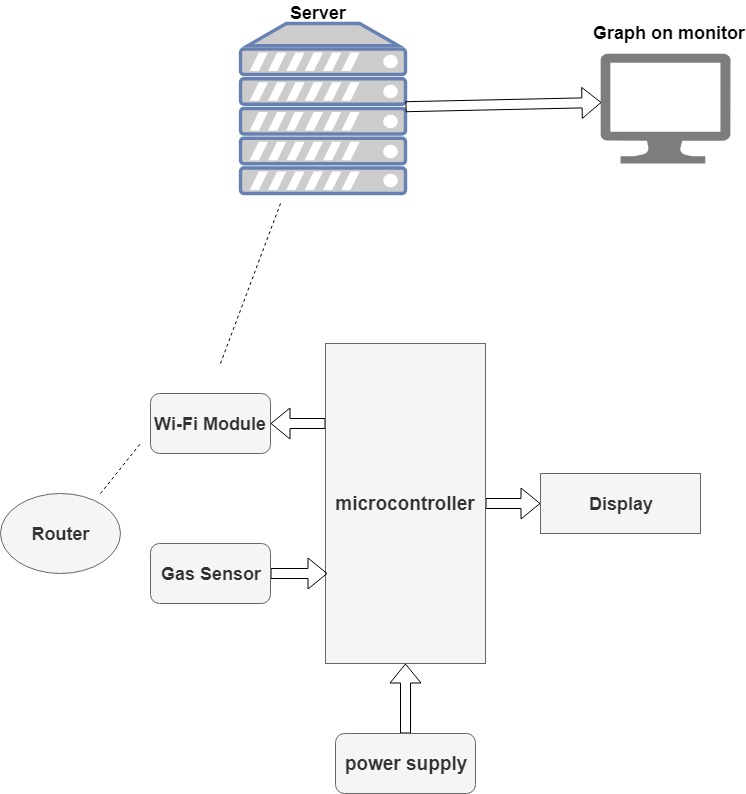
First is Smart Environment Monitoring using Wireless sensor networks in which the main focus was on the developing an environment free of pollution by making it smart. Wireless sensors are fitted all over the city and in public transports.

By monitoring all the sensor networks, all the environmental happenings can be gathered as a streaming database to analyse the environmental position.

**Chapter 3:**

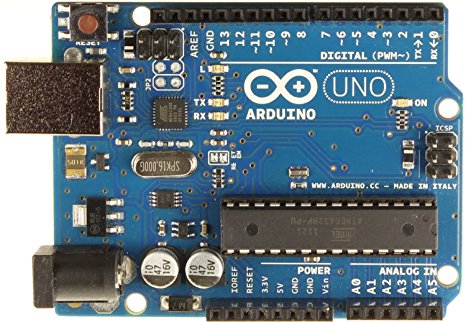
**Hardware Development**

**(i) Block diagram:**

****

**Explanation of blocks:**

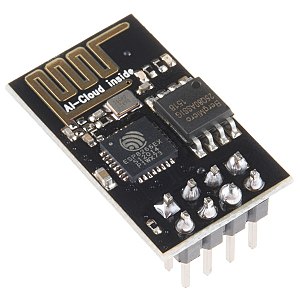
1) Arduino UNO As microcontroller: Arduino Uno is a microcontroller board based on the ATmega328P It has 14 digital input/output pins 6 analog inputs, a 16 MHz quartz crystal, a USB Connection, power jack, an ICSP header and a reset button as shown in figure.



2) MQ135 Sensor As gas sensor: The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases. It gives the output in form of voltage levels. Figure shows the sensor MQ135.

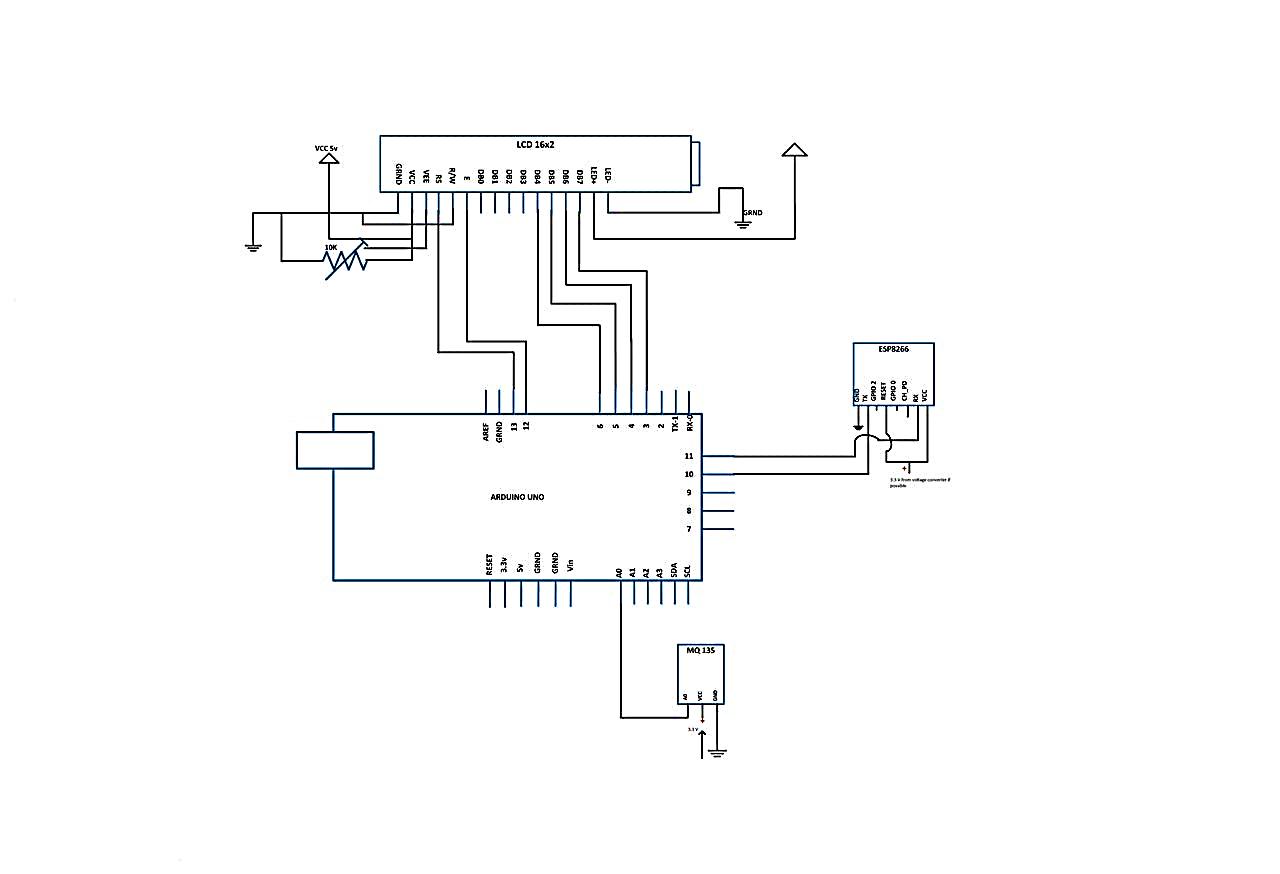


3) ESP8266 WIFI Module As Wi-Fi module: The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and MCU (microcontroller unit) capability. It runs on 3.3V and gives our system access to Wi-Fi or internet. Figure shows Wi-Fi Module (ESP8266).



4) 16x2 LCD As Display: This is a basic (16x2) 16 character by 2 line display. Black text on green background. It is used to indicate the Air and Humidity in PPM. Fig. 6 shows LCD (16x2).



**(ii) Circuit diagram:**

# Flow of Circuit:

First applying 5V DC to the Arduino board. Board will start executing program uploaded on chip. As per program first it will be try to connect to the router through Wi-Fi module it will search for SSID and password match entered in a program if SSID and password will match that will be connect else it will show Wi-Fi connect error.

Microcontrollers LCD display will be show the welcome note and all that written in code. Controller will read values coming from gas sensor and display on LCD, values coming from analog pin will display as fresh air or poor air. The range of fresh air is between 0 to 300 PPM above this value air will be considered as poor air.

Microcontroller will send the values on 16 seconds delay of each value to the WI-Fi module which is connected to the internet through the router.

Now, Wi-Fi module will send data to the thingspeak server through a special key called API key which specifies our channel and field for plotting graph of our sensor values.

For that first Wi-Fi module will request to server for providing service. This will establish TCP connection on “api.thingspeak.com” at port no. 80 Through AT commands.

After establishing TCP connection this start sending data through a data frame this frame contains start bit, stop bit, guard bit and tail bit. Trough AT commands data will be sending to the server on desired API key. After every delay values will be updated through microcontroller.

Thingspeak is free server for storing data it provides collect data analyze data and act on data. Thingspeak will collect data on 16 seconds delay of each value which is default for this server for analyze the data and plotting graph.

This will collect our sensor data and analyze it for visualization on graph. After analyze data It will show the graph on your monitor.

# Connections Description:

**1) LCD to Arduino:**

|  |  |
| --- | --- |
| GND | GND |
| VCC | 5V |
| VEE | 10k Pot (output) |
| RS | 13 |
| R/W | GND |
| Enable | 12 |
| DB4 | 6 |
| DB5 | 5 |
| DB6 | 4 |
| DB7 | 3 |
| LED+ | 5V |
| LED- | GND |

**2) ESP8266 to Arduino:**

|  |  |
| --- | --- |
| GND | GND |
| Tx | 10 |
| Reset | 3.3V |
| Rx | 11 |
| VCC | 3.3V |

**3) MQ135 to Arduino:**

|  |  |
| --- | --- |
| VCC | 5V |
| GND | GND |
| AO | A0 |

**4) Pot connections:**

Output pin of Pot must connect to 3rd pin (VEE) of LCD.

Other two remaining pins connect to 5v and Ground.

**(iii) Components list and specifications:**

1) Arduino UNO

2) ESP8266 Wi-Fi Module

3) MQ135 Gas Sensor

4) 16x2 LCD

5) 10k Potentiometer

**1) Arduino UNO Specifications:**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 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 Boot loader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**2) ESP8266 Wi-Fi Module Specifications:**

* 802.11 b/g/n
* Wi-Fi Direct (P2P), soft-AP
* Integrated TCP/IP protocol stack
* Integrated TR switch, balun, LNA, power amplifier and matching network
* Integrated PLLs, regulators, DCXO and power management units
* +19.5dBm output power in 802.11b mode
* Power down leakage current of <10uA
* 1MB Flash Memory
* Integrated low power 32-bit CPU could be used as application processor
* SDIO 1.1 / 2.0, SPI, UART
* STBC, 1×1 MIMO, 2×1 MIMO
* A-MPDU & A-MSDU aggregation & 0.4ms guard interval
* Wake up and transmit packets in < 2ms
* Standby power consumption of < 1.0mW (DTIM3)

**3) MQ135 Gas Sensor Specifications:**

* Wide detecting scope
* Fast response and High sensitivity
* Stable and long life
* Operating Voltage is +5V
* Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
* Analog output voltage: 0V to 5V
* Digital output voltage: 0V or 5V (TTL Logic)
* Preheat duration 20 seconds
* Can be used as a Digital or analog sensor
* The Sensitivity of Digital pin can be varied using the potentiometer

**4) 16x2 LCD Specifications:**

* Operating Voltage is 4.7V to 5.3V
* Current consumption is 1mA without backlight
* Alphanumeric LCD display module, meaning can display alphabets and numbers
* Consists of two rows and each row can print 16 characters.
* Each character is build by a 5×8 pixel box
* Can work on both 8-bit and 4-bit mode
* It can also display any custom generated characters
* Available in Green and Blue Backlight

**Chapter 4:**

**Software Development**

Q. How data of sensor will go to the server?

* Output of sensor mq135 will be in form of different voltage levels. This is sensation of alcohol, smoke and benzene gases. Different voltage levels will convert further in digital from 0 to 1024 in microcontroller. Controller will calculate the average value of gases. And it will represent it in form of PPM. This data in PPM is only the average value of alcohol, smoke and benzene. Now that data in PPM will go to the server through one desired special API key to that exact channel of us and from values of sensor graph will be automatically plotted by thingspeak.

**(i) Server:**

Here, for sending data of sensor on internet, we are using Thingspeak Server which can collect data analyses data and can also act as per given conditions of different data values. Thingspeak shows graph on their Sever of our channel, for that we have to sign up on thingspeak.com and have to create our channel. Process of this as shown in fig.

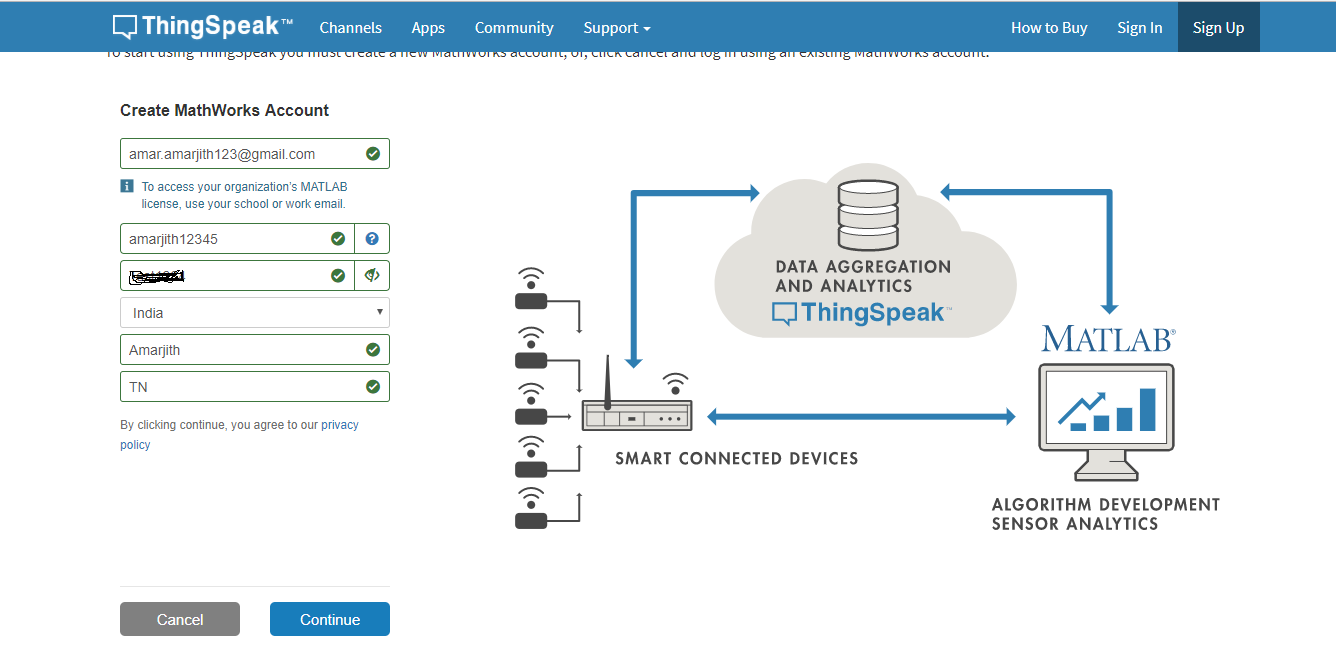
**Create thingspeak account**

**Step: 1** sign up things speak

https://thingspeak.com/users/sign\_up

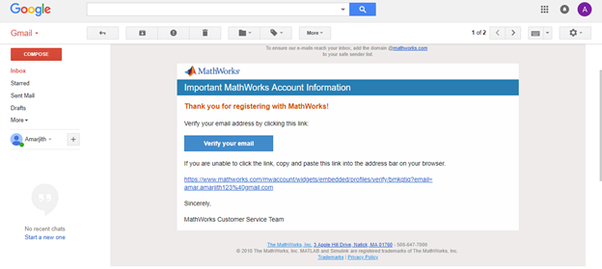
**Step: 2**

Fill out all fields



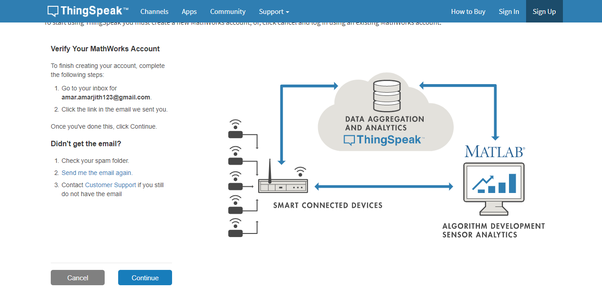
**Step: 3**

Verify email account



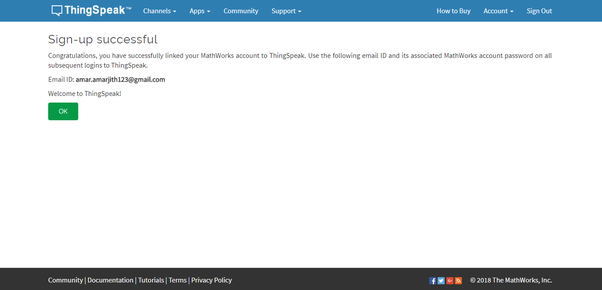
**Step:4**

After verification, click on continue



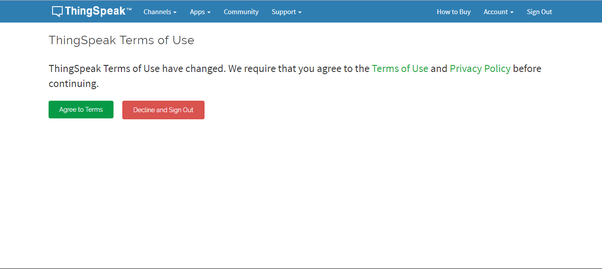
**Step:5**

OK



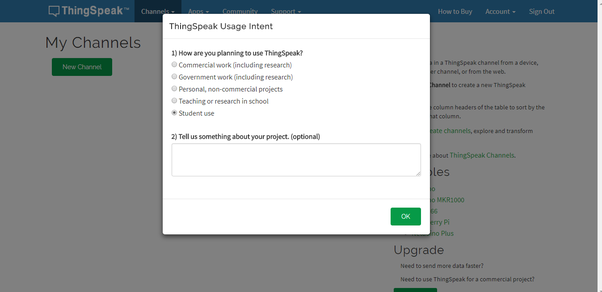
**Step:6**

Agree to terms



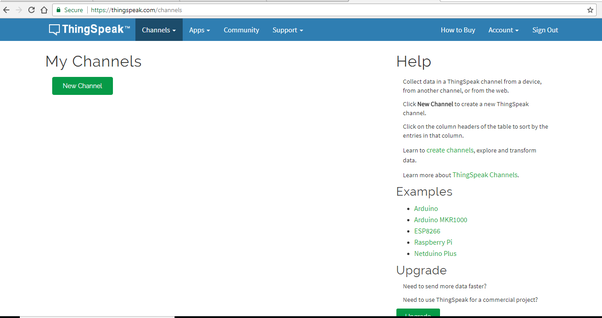
**Step: 7**

Select Student Use



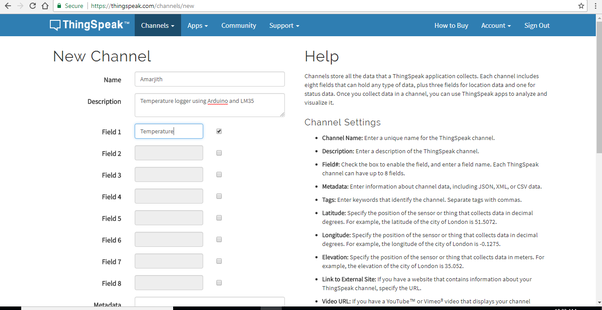
**Step : 8**

Wait for this window



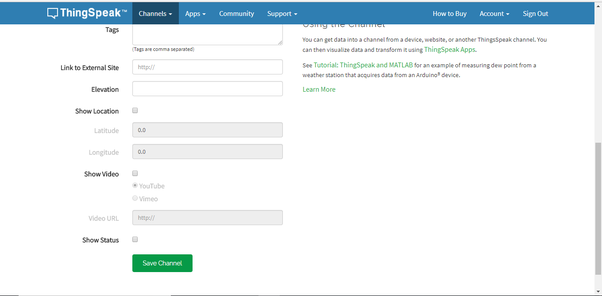
**Step:9**

Create new channel



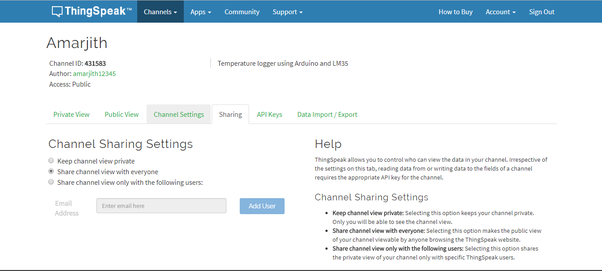
**Step:10**

Save channel



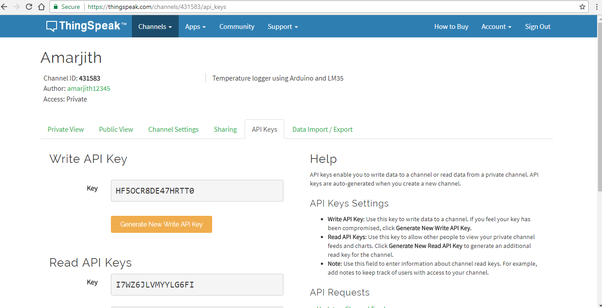
**Step:11**

Make channel public



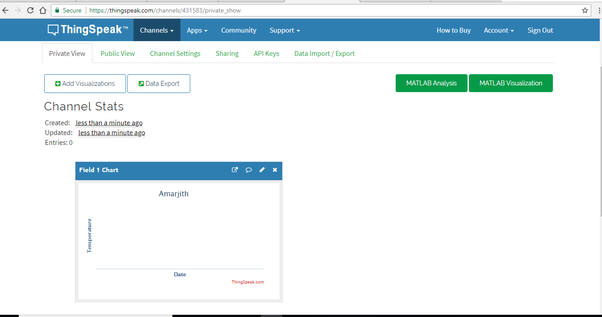
**Step:12**

Copy “write apikey” and then paste it in your program (later)

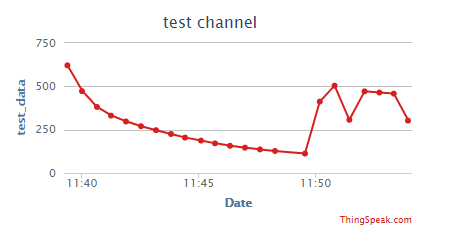


**Step:13**

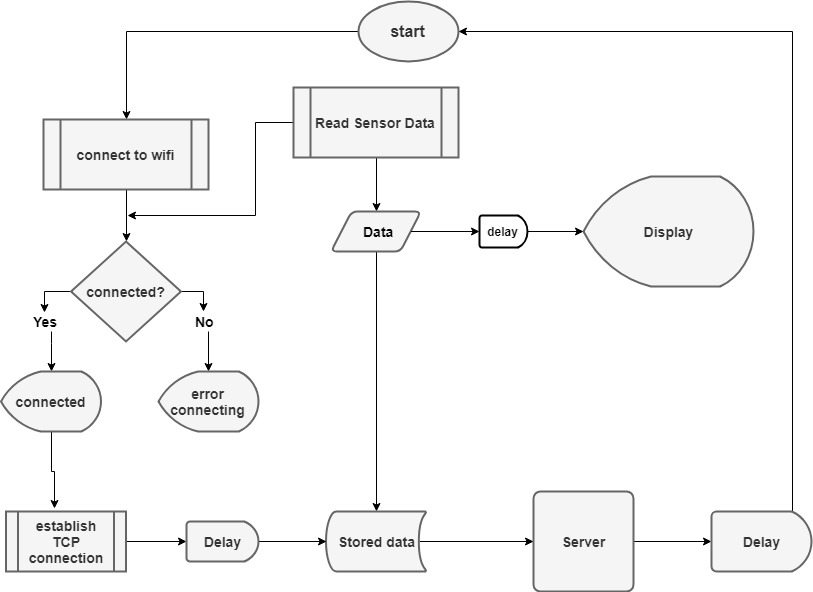
Open private view



Here in a Private view you can see the output graph like this shown in fig.



**(ii) Flow chart:**

****

**Chapter 5:**

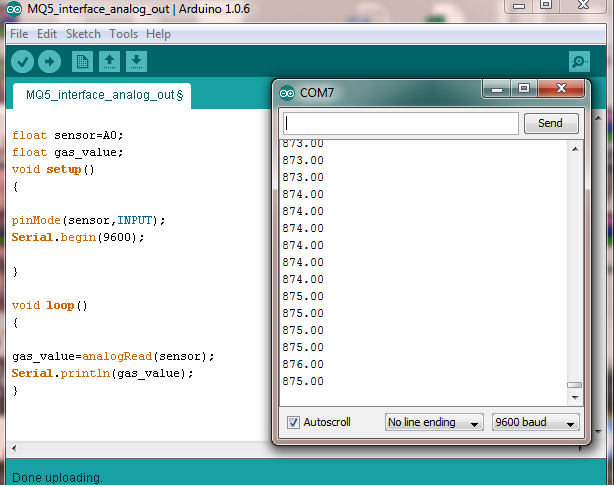
**Project Outcomes and Applications**

**Outcomes:**

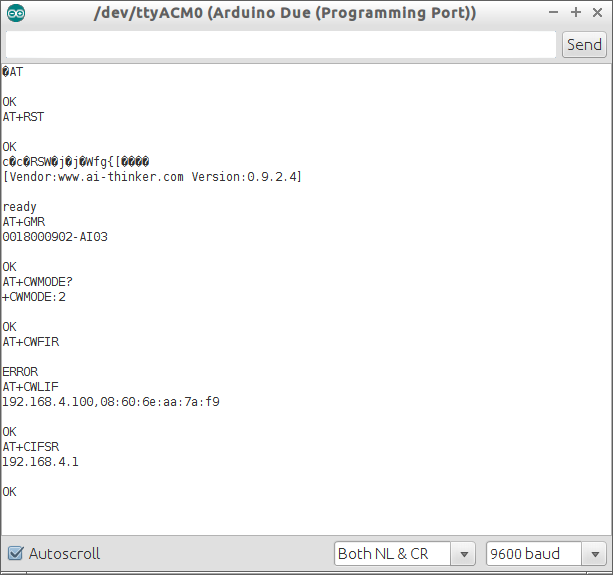
1) Output of Arduino to LCD test interfacing



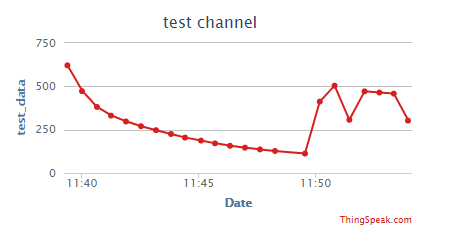
2) Output of Arduino to Gas sensor interfacing

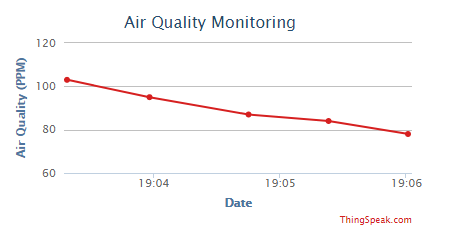


3) Testing AT commands on ESP8266

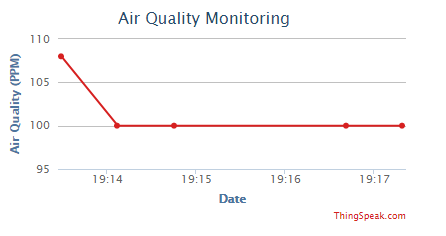


4) Output graph on thingspeak



****

**[Location: EC Dept. GP Ahmedabad]**

****

**[Location: Garden GP Ahmedabad]**

**Applications:**

* In your factory
* In your office
* In your home
* In hospitals and pollution affected areas
* Air pollution control for coating industry
* Measurement of methane and non-methane hydrocarbons for environmental air quality
* Ambient air quality monitoring for governments.
* Continues emissions monitoring for CO2 capture.
* For tunnel monitoring

**Chapter 6:**

**Future Expansions**

This system is IOT based means this system can access internet. You can monitor your home, office and factory pollution continuouslyIn future for making this system more advanced we can interface an air purifier or air ionizer through relay in a system.

That means we can switch on or off our purifier on some conditions like, if pollution increases from some PPM then on the relay that enables purifier and in else off the relay.

In next step we can use node MCU instead of Arduino it decreases cost, size and power also because node MCU takes only 3.3V dc for running it.

In more advanced we can put Raspberry pi and we can control system through voice commands from anywhere on earth. We can interface more number of sensors in system that causes different pollution levels measurement on different department of my office or factory.

This system can be used after making it a powerful and reliable in corporate offices and manufacturing.

**Chapter 7:**

**References**

[1] circuitdigest.com/microcontroller-projects/iot-air-pollution-monitoring-using-arduino

[2] www.engineersgarage.com/contribution/arduino-based-air-quality-monitoring-iot-project

[3] irjet.net/archives/V4/i10/IRJET-V4I10207.pdf

# [4] Exploring Arduino: Tools and Techniques for Engineering Wizardry 1st Edition by Jeremy Blum

[5] www.arduino.cc

[6] www.quora.com/How-do-I-send-data-on-ThingSpeak-for-an-MQ135-gas-sensor

[7] create.arduino.cc/projecthub/ruchir1674/air-quality-monitoring-7c5bae?ref=search&ref\_id=air%20quality&offset=3

[8] www.instructables.com/id/Air-Qualiy-Monitoring/

[9] www.hackster.io/ruchir1674/air-quality-monitoring-7c5bae

[10] www.components101.com/sensors/mq135-gas-sensor-for-air-quality