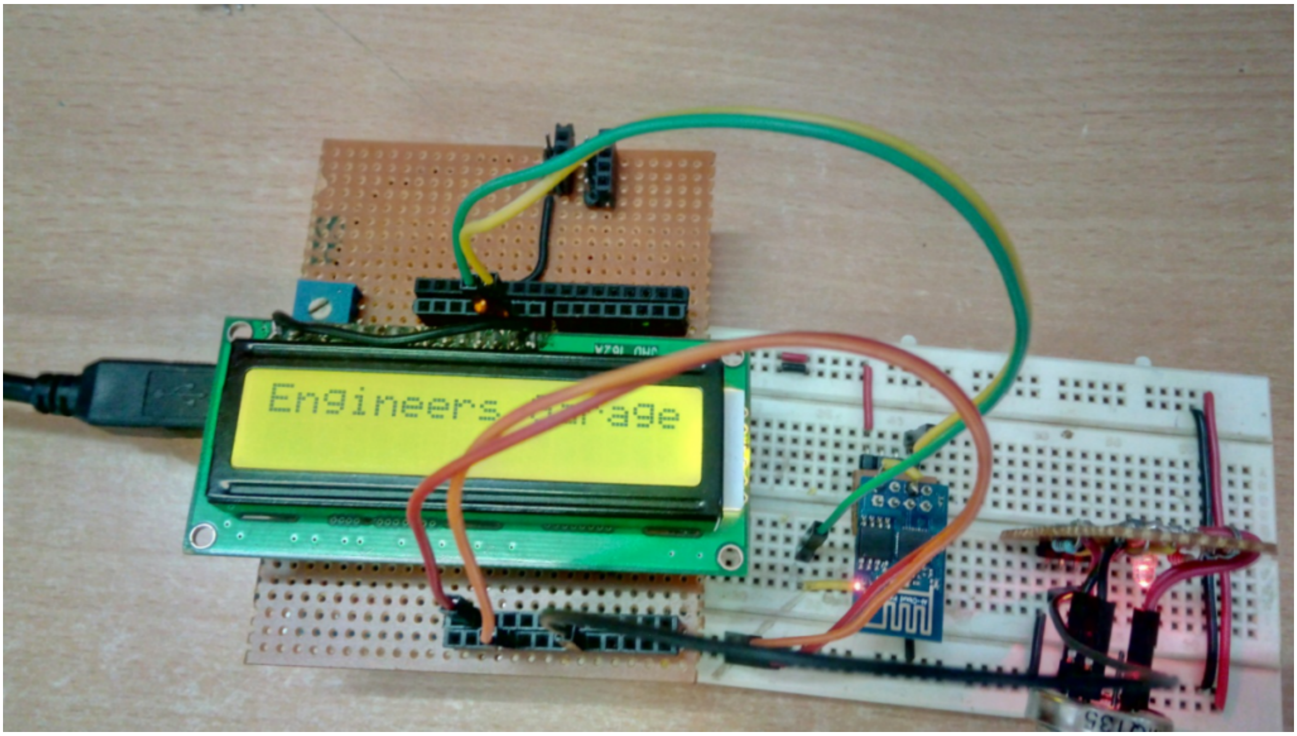
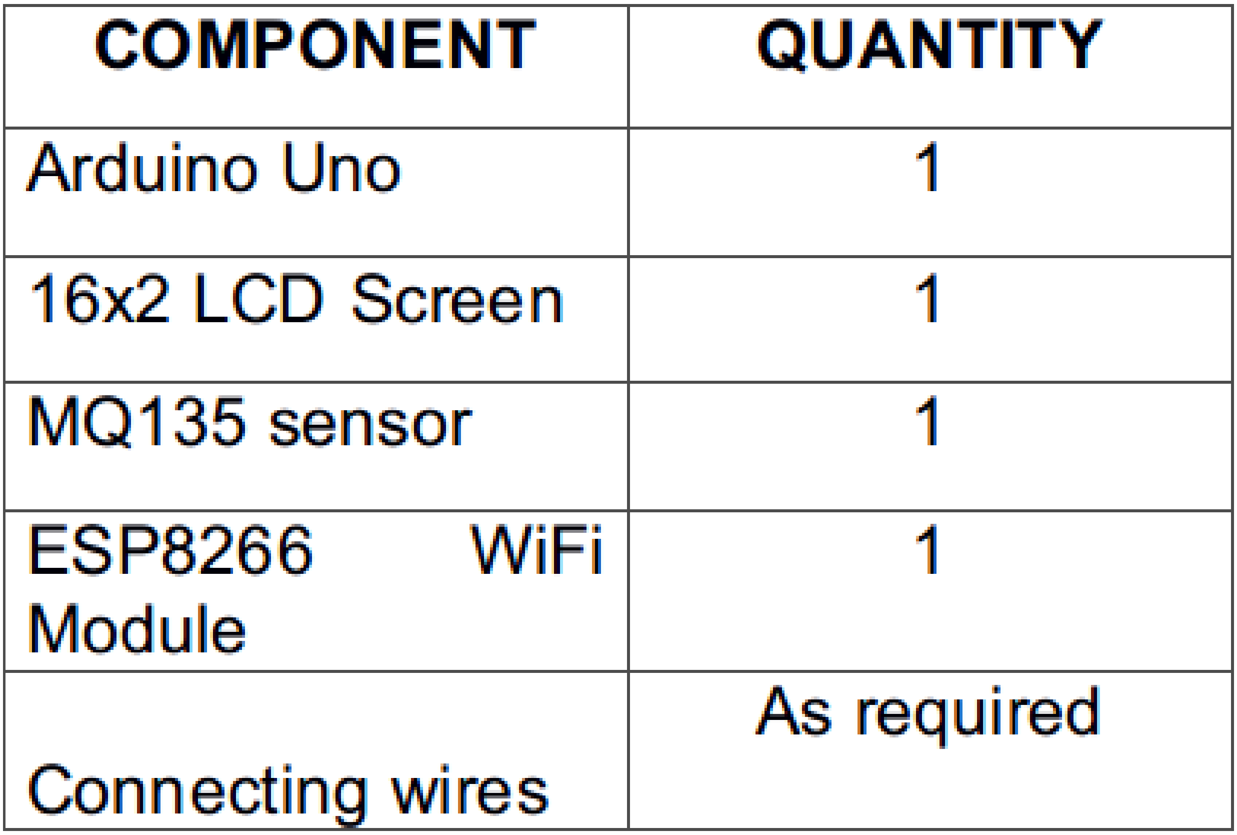
Arduino Based Air Quality Monitoring IOT Project

Air pollution has become a common phenomenon everywhere. Specially in the urban areas, air pollution is a real-life problem. A lot of people get sick only due to air pollution. In the urban areas, the increased number of petrol and diesel vehicles and the presence of industrial areas at the outskirts of the major cities are the main causes of air pollution.The problem is seriously intensified in the metropolitan cities. Also, the climate change is now apparent. The governments all around the world are taking every measure in their capacity. Many European countries have aimed to replace petrol and diesel vehicles with the electric vehicles by 2030. Even India has aimed to do so by 2025. The use of coal for electricity generation is now going to be a thing of past. The nations are now focusing to generate energy from nuclear reactors and the renewable resources like solar energy, wind energy and hydroelectric power.



Components Required :



COMPONENT	QUANTITY
Arduino Uno	1
16x2 LCD Screen	1
MQ135 sensor	1
ESP8266 WiFi Module	1
Connecting wires	As required

Block Diagram –

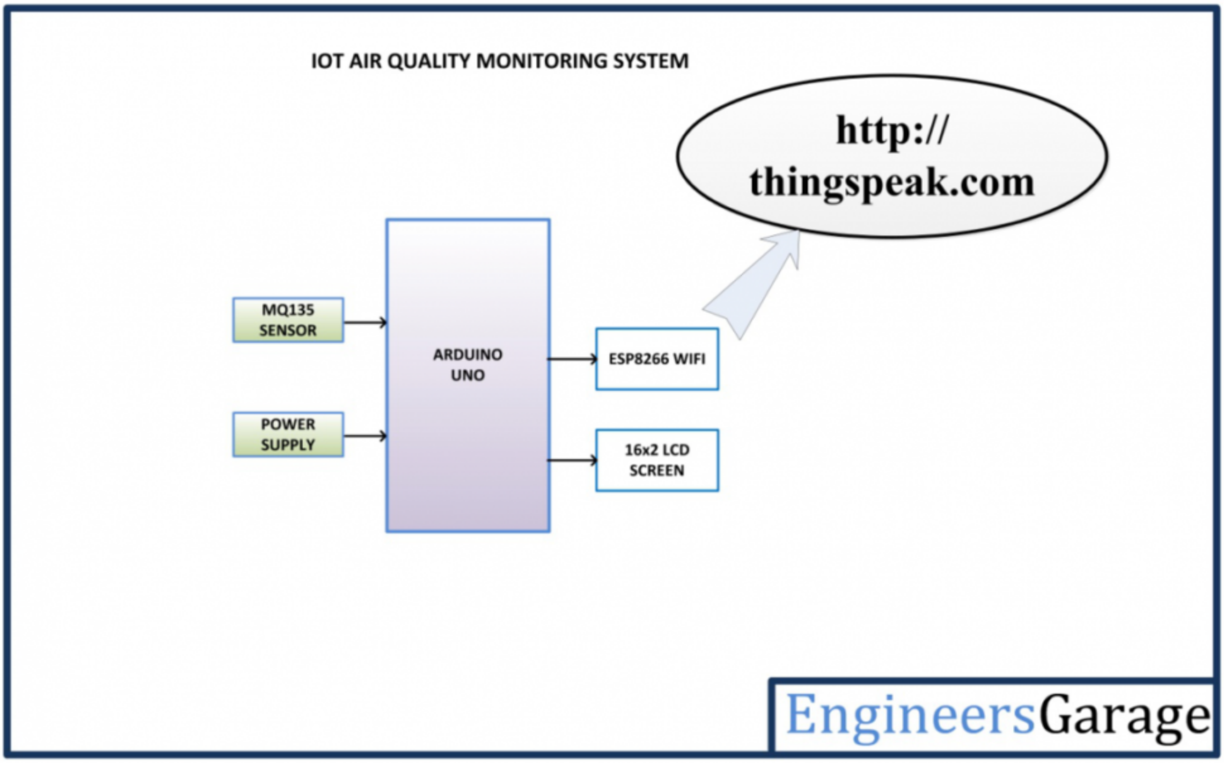


Fig. 3: Block Diagram of Arduino based IOT device for Real Time Air Pollution Monitoring

Circuit Connections –

The air pollution monitoring device is built by assembling the following components

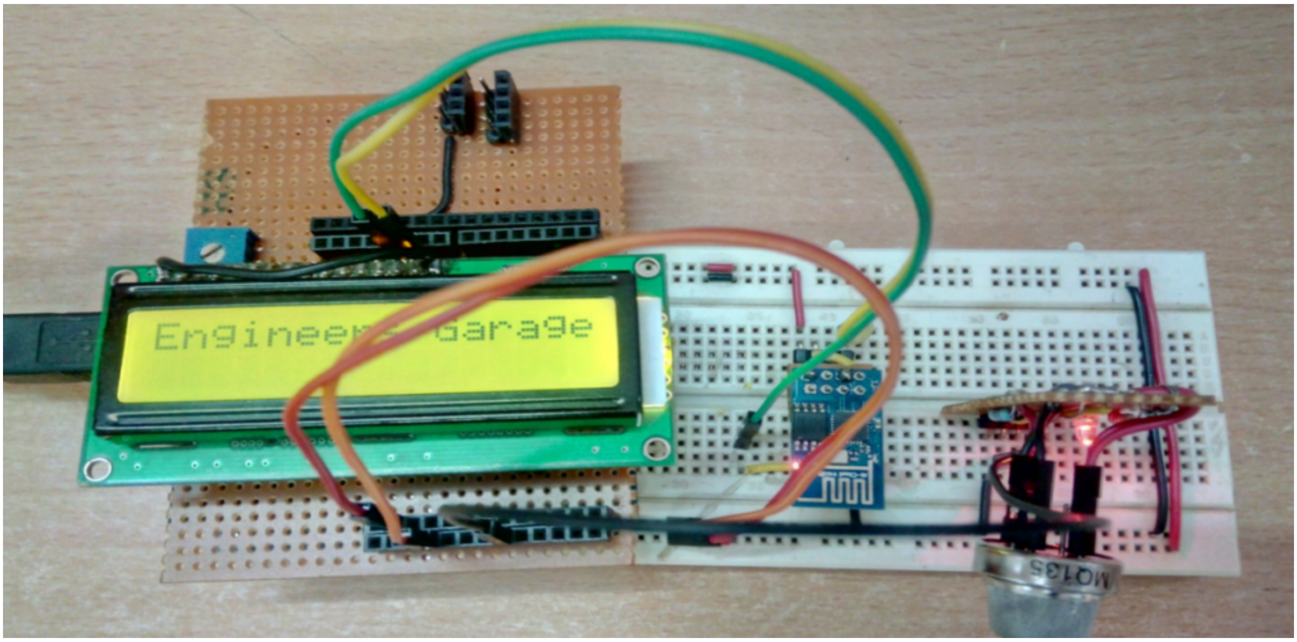
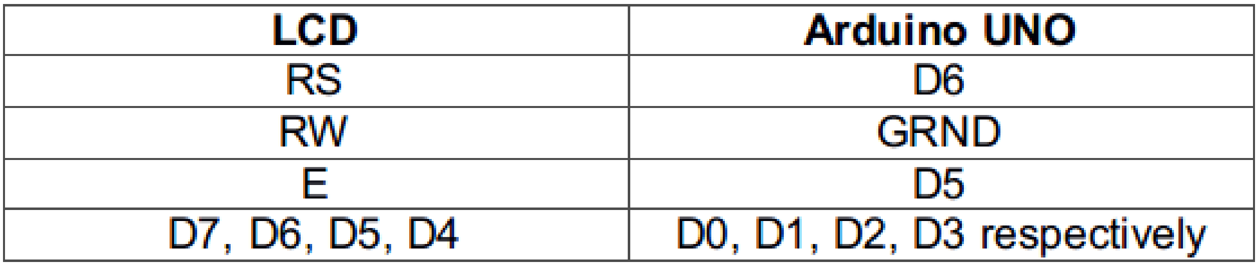


Fig. 4: Image showing circuit connections of Arduino based IOT device for Real Time Air Pollution Monitorin

Arduino UNO – Arduino UNO is one of the most popular prototyping boards. It is small in size and packed with rich features. The board comes with built-in Arduino boot loader

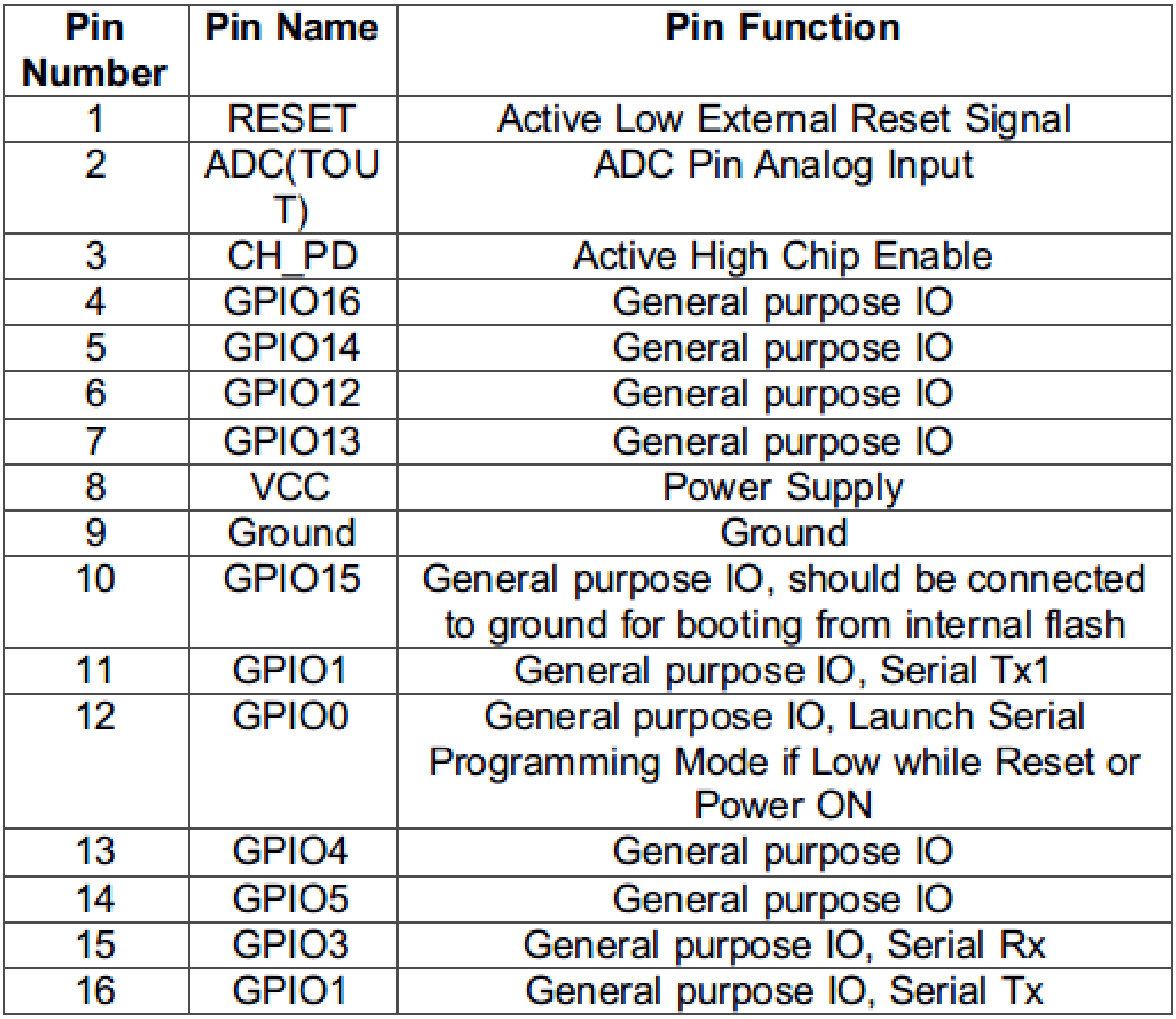
.**16X2 Character LCD –** The 16X2 LCD display is used to monitor the sensor values read by the Arduino board from MQ-135. It is interfaced with the Arduino UNO by connecting its data pins D4 to D7 with pins 6 down to 3 of the controller respectively.



LCD	Arduino UNO
RS	D6
RW	GRND
E	D5
D7, D6, D5, D4	D0, D1, D2, D3 respectively

Fig. 5: Table listing circuit connections between Character LCD and Arduino Uno

ESP8266 Wi-Fi Module – The ESP8266 Wi-Fi Module is used to connect with any available internet hotspot and transfer sensor data to ThingSpeak Platform via Wi-Fi. The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to a Wi-Fi network.



Pin Number	Pin Name	Pin Function
1	RESET	Active Low External Reset Signal
2	ADC(TOUT)	ADC Pin Analog Input
3	CH_PD	Active High Chip Enable
4	GPIO16	General purpose IO
5	GPIO14	General purpose IO
6	GPIO12	General purpose IO
7	GPIO13	General purpose IO
8	VCC	Power Supply
9	Ground	Ground
10	GPIO15	General purpose IO, should be connected to ground for booting from internal flash
11	GPIO1	General purpose IO, Serial Tx1
12	GPIO0	General purpose IO, Launch Serial Programming Mode if Low while Reset or Power ON
13	GPIO4	General purpose IO
14	GPIO5	General purpose IO
15	GPIO3	General purpose IO, Serial Rx
16	GPIO1	General purpose IO, Serial Tx

Fig. 6: Table listing pin configuration og ESP8266 ESP-12 Wi-Fi Modem

MQ-135 Sensor – MQ-135 is a gas sensor which is used to measure the concentration of combustible gases. It has lower conductivity in clean air while its conductivity increases with the presence of the combustible gases in the air.

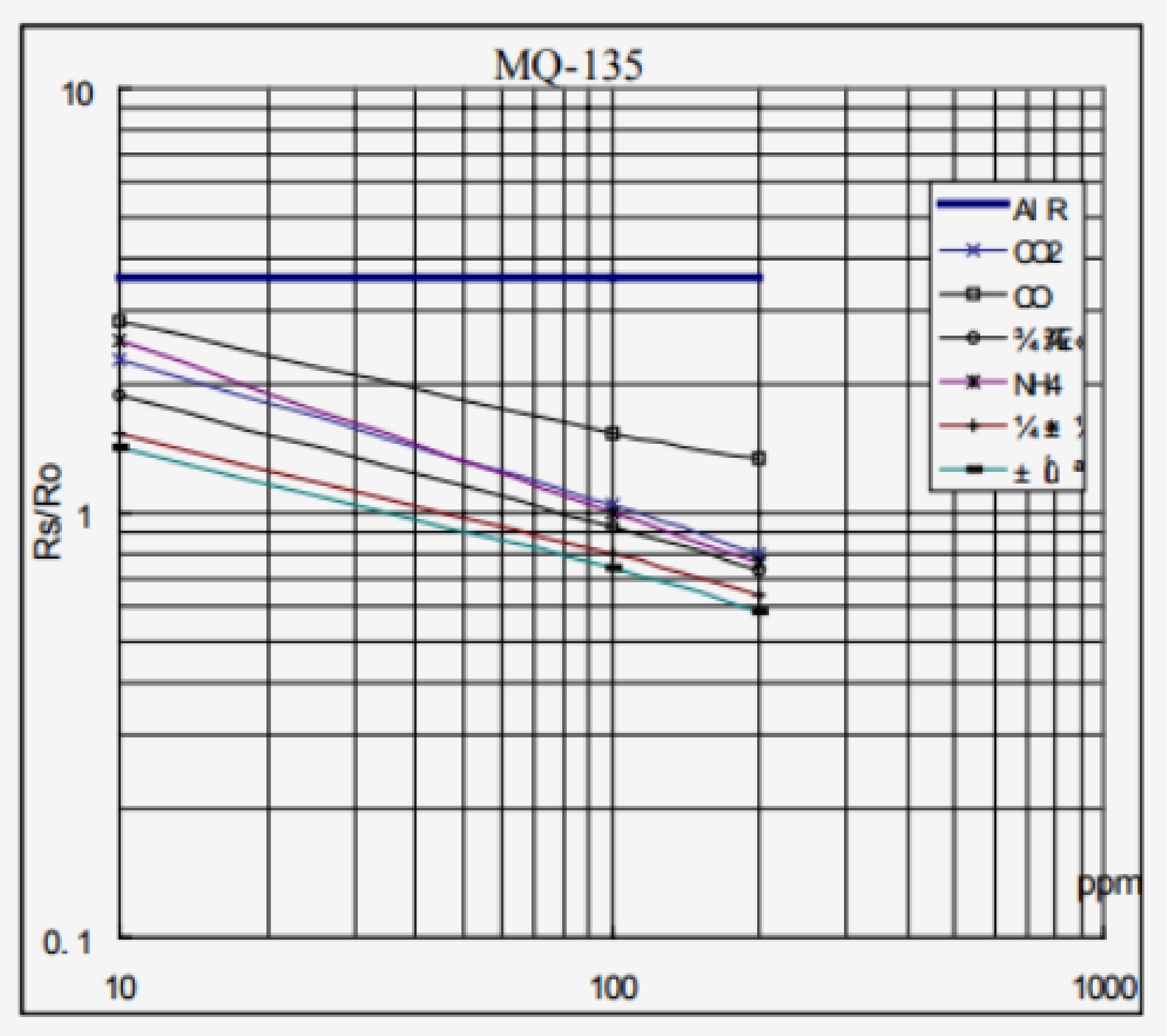


Fig. 7: Graph showing Sensitivity Curve of MQ-135 Sensor

Power Supply – The Arduino board and the Wi-Fi module require 3.3 V while LCD and MQ-135 sensor need 5V DC for their operation. The Arduino can be powered by connecting it to a USB connection.

How the circuit works – The device developed in this project can be installed near any Wi-Fi hotspot in a populated urban area. As the device is powered, the Arduino board loads the required libraries, flashes some initial messages on the LCD screen and start sensing data from the MQ-135 sensor. The sensitivity curve of the sensor for different combustible gases is already mentioned above. The sensor can be calibrated so that its analog output voltage is proportional to the concentration of polluting gases in PPM. The analog voltage sensed at the pin A0 of the Arduino is converted to a digital value by using the in-built ADC channel of the Arduino. The Arduino board has 10-bit ADC channels, so the digitized value ranges from 0 to 1023. The digitized value can be assumed proportional to the concentration of gases in PPM. The read value is first displayed on LCD screen and passed to the ESP8266 module wrapped in proper string through virtual serial function. The Wi-Fi module is configured to connect with the ThingSpeak IOT platform. ThingSpeak is an IOT analytics platform service that allows to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by the IOT devices to ThingSpeak server. With the ability to execute MATLAB code in ThingSpeak one can perform online analysis and processing of the data as it comes in.

The Wi-Fi module can be connected with the ThingSpeak server by sending AT commands from the 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 –

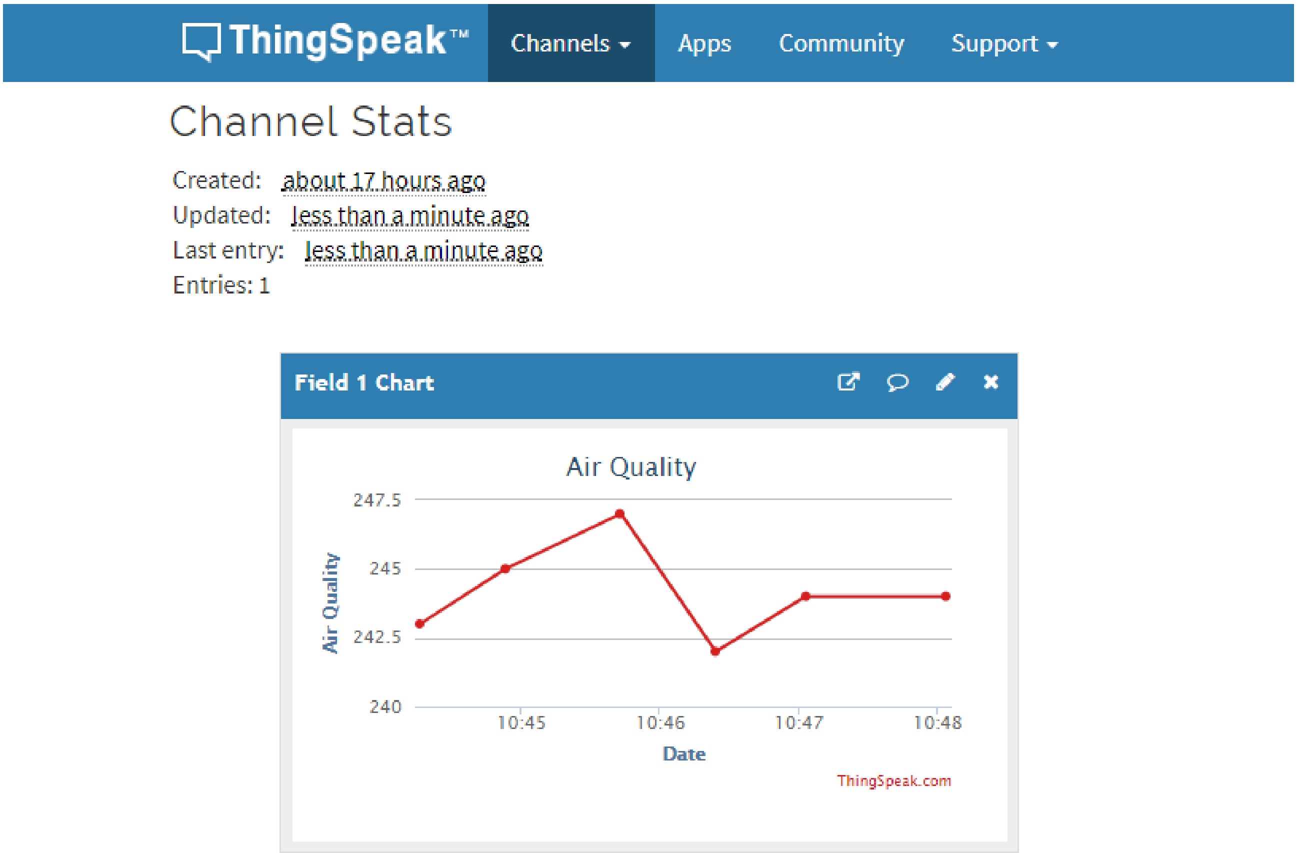
AT + CWMODE = 3

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 –

1 – set Wi-Fi connection to station mode

2 – set Wi-Fi connection to SoftAP mode

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



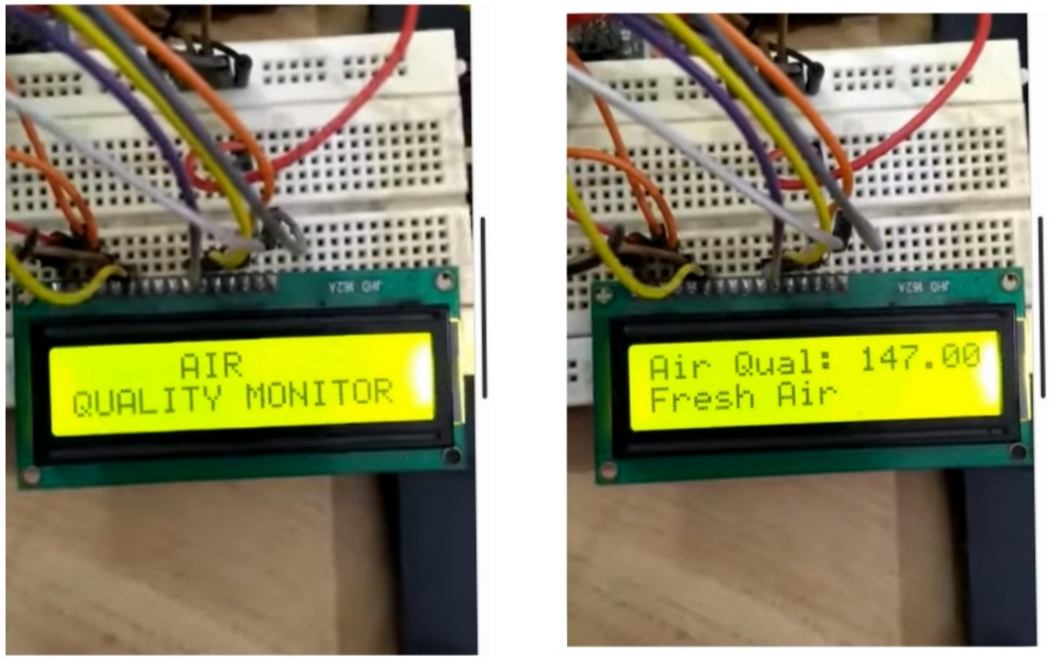
Channel Stats

Created: about 17 hours ago
Updated: less than a minute ago
Last entry: less than a minute ago
Entries: 1

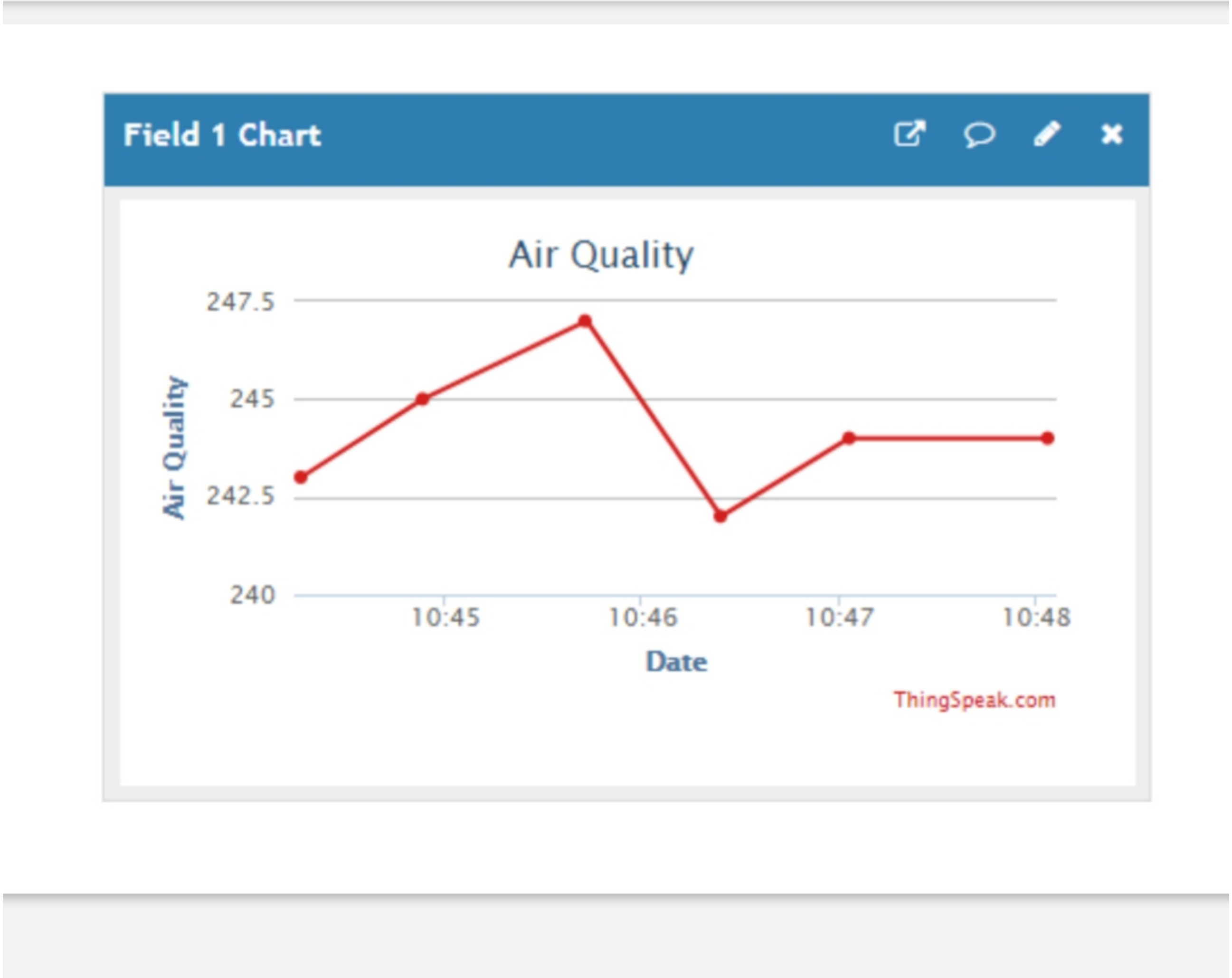
Fig. 8: Screenshot of User Interface of ThingSpeak Platform

The user needs to login the ThingSpeak platform from the registered account to view and monitor the sensor data. The Arduino sketch manages to read sensor data and send the AT commands for connecting with the IOT platform. Get an understanding of the Arduino code from the programming guide.

Hard ware Outputs;



Soft ware output


Field 1 Chart
Air Quality
247.5
245
242.5
240
Air Quality
10:45
10:46
10:47
10:48
Date
ThingSpeak.com