1. Working of proposed project

1.1 Working

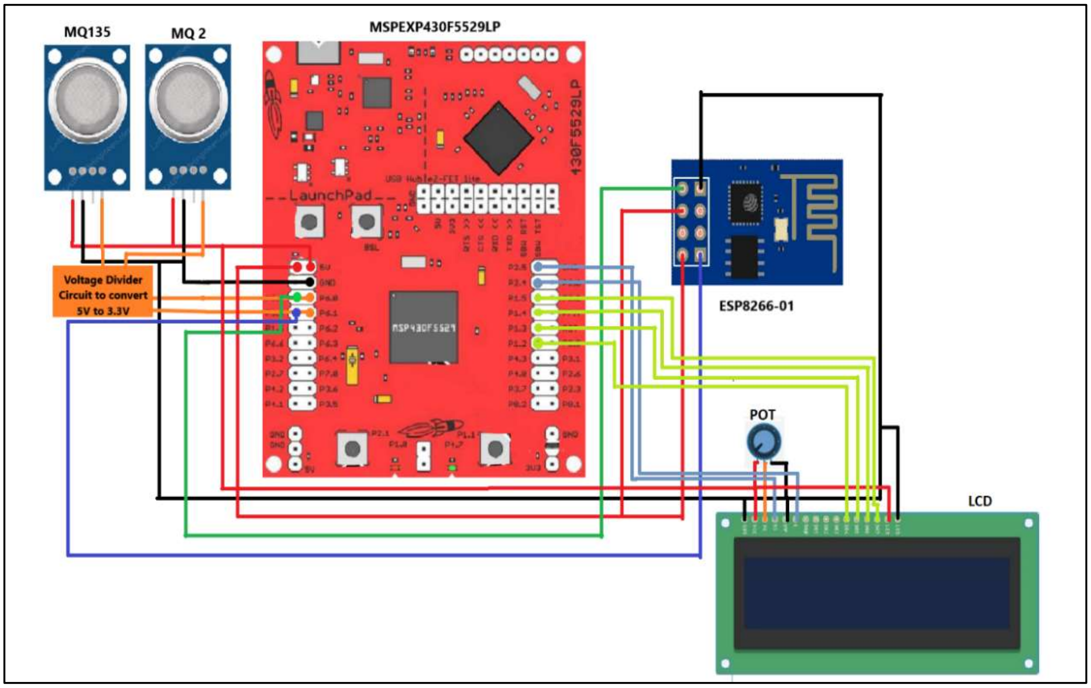
As described in the block diagram of this project. MSP430 is used as the primary microcontroller and other components like ESP8266-01, MQ2, MQ135, power supply, LCD display are interfaced with MSP430. After connecting all the modules to MSP430, the code will be uploaded to MSP430 via Energia IDE which includes libraries of sensors along with calibration calculations & Wi-fi credentials with Thingspeak's API key to connect ESP8266 -01 to the wi-fi and then send the data to the Thingspeak server.

As code will be uploaded the readings taken from the sensor will be first calibrated to give correct reading and then will start to appear on the LCD display, serial monitor of Energia & Channel created in Thingspeak. Channel in the Thingspeak server is easy to create as it is user friendly. Thingspeak is a free cloud hosting service provided by MATLAB.

The gas sensors MQ135, MQ2 are used to detect gases CO2, CO, LPG, and smoke for this project. MQ2 measure the CO, LPG whereas MQ135 will measure CO2 and smoke. When sensors are connected to MSP430, they will start detecting gases present in the environment around the sensors, then these readings will be calibrated to give correct PPM values. These correct values will appear on LCD & the web server. The reading displayed on the LCD will be temporary but the reading sent to the server will be stored and can be viewed later as well.

[7]MQ sensors are electrochemical sensors i.e., the voltage level of these sensors varies with the change in the concentration of gases present in the air. Higher the gas concentration present in the environment, higher is the voltage given as an output from the sensor. This phenomenon occurs due to chemical reactions between electrons from SnO_2 layer present in the MQ sensor and the environmental gases.

Below is the circuit diagram. The sensors are connected to the MSP430. The sensors will sense some gas and give voltage readings which would be further converted to resistance value and this would in turn be used to calculate value of gases in ppm. Since the sensor's operating voltage is 5V and that of MSP430 and ESP8266 wi-fi module are 3.3V, the o/p voltage from the sensors is turned into a smaller one by using the voltage divider circuit. ESP8266 - 01 is also directly connected to MSP430 since their operating voltage matches.

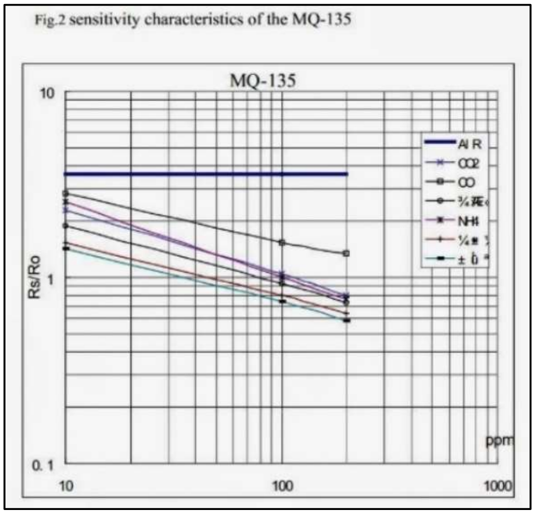


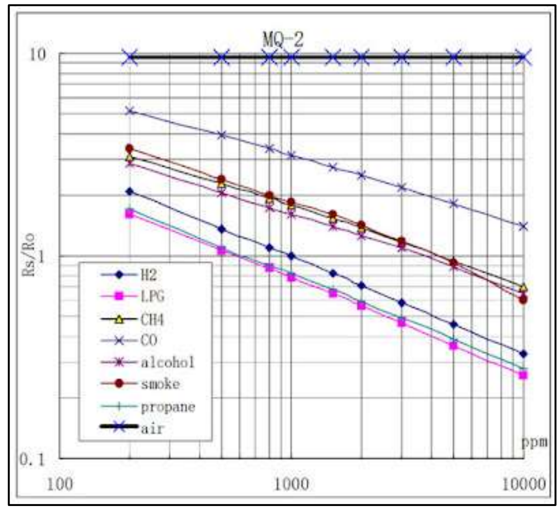
Circuit Diagram of IoT based Air pollution Monitoring System

The transmitters and receiver pins of ESP8266 – 01 module are connected to MSP430 for the data transfer via UART. This data would be visualized on channel created on Thingspeak server.

1.2 Calibration for different gases

As mentioned earlier, analog sensor will give raw analog reading which has to be converted to ppm values of mentioned gas. In this topic we will be discussing about converting raw analog values into correct (very close to correct) values.





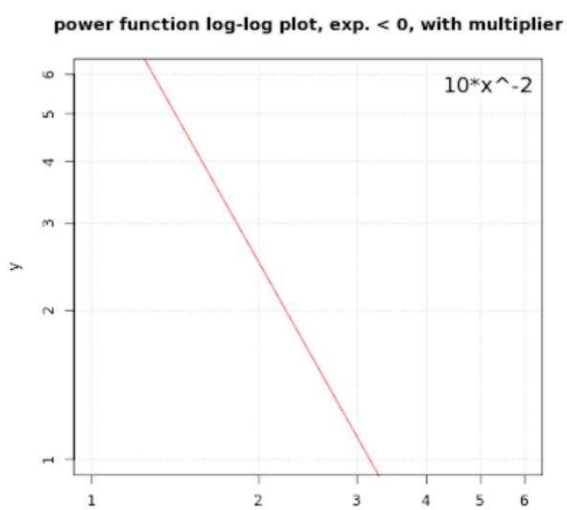
Sensitivity curve of MQ-135

Sensitivity curve of MQ-2

It is worth noting that the converted values will consider already present environmental gases as well as sense externally generated gas if present now. (For example, burning a match stick, paper, etc.)

To calculate correct reading, we take help of the graphs (in log-log scale) provided by the manufacturer of the sensor via datasheets

For simplicity and easy calculation, we are exchanging the x-axis and y-axis. Therefore, x-axis will have Rs/Ro ratio and y-axis will have ppm values of different gases where Rs denoted resistance of sensor in presence of specific gas and Ro denoted resistance of sensor in fresh air.



2.5 Graphical Representation of line having equation y = a*x^-b

When we invert the axis of graphs and analyze. We can see that the line equation can be expressed as the function of power with a negative exponent. Therefore, the line equation will be:

$$y = a*x^b \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (1) \text{ where } b < 0.$$

$$ppm = a*(Rs/Ro)\ ^b \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (2) \text{ where } b < 0.$$

The co-efficient a and b are obtained through curve fitting. There is a web plot digitalizer from which we can find out x and y point of our needed gas from the given graph in the datasheet. After obtaining various x and y points. We do curve fitting to find appropriate value of a and b using R script by substituting values obtained previously (x and y).

So rearranging equation no. (2)

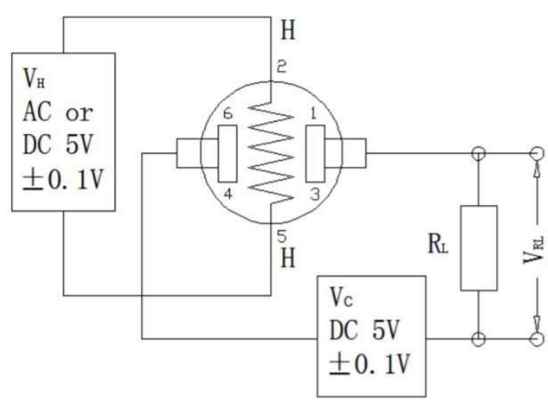
$$Ro = Rs*\ powf\ (a/ppm,\ 1/b) = Rs*(a/ppm)\ ^(1/b)$$

$$Ro = Rs*(a/ppm)\ ^(1/b) \ \ldots\ldots\ldots \ (3)$$

To calculate Ro (Sensor resistance in fresh air), we substitute values of a, b, Rs and average/current ppm amount of the gas to be measured where Rs (Sensor Resistance in presence of gas) is calculated from the analog reading taken from the sensor.

$$Rs = ((1023 / Raw\ Analog\ Value) - 1.) * RLoad \ \ldots\ldots\ldots \ (4)$$

Calculation of Rs



Test circuit of MQ-135

From the above figure sensor resistance Rs and RLoad form a voltage divider, which can be described with the following formula:

$$VRload = [Rload / (Rload + Rs)] * Vin$$

Therefore,

$$Rs = [(Vin / VRload) - 1] * Rload$$

We can replace Vin / VRload relation with ADCmaxvalue / ADCvalue relation (where ADCmaxvalue is usually 4096 for MSP430 and always represents ADC value when Vadc = Vref, ADCvalue is proportional to Vadc), so we get:

$$Rs = (ADCmaxvalue / ADCvalue - 1) * Rload$$

or

$$Rs = (4096 / ADCvalue - 1) * Rload$$

All the calculated values from equation no. *(2), (3)* and *(4)* are then substituted in equation no. *(1)* to calculate correct (approximate) ppm values from raw analog values. These calculated values are then displayed on the created channel as well as on the LCD display.