

## **Code Explanation :**

Using MQ135 library we can directly get the PPM values, by just using the below two lines:

```
MQ135 gasSensor = MQ135(A0);
```

```
float air_quality = gasSensor.get  
PPM ();
```

to calibrate the MQ135 sensor, for calibrating the sensor upload the below given code and let it run for 12 to 24 hours and then get the RZERO value.

```
#include "MQ135.h"
```

```
void setup()
```

```
Serial.begin (9600);
```

```
}
```

```
void loop() {
```

```
MQ135 gasSensor = MQ135(A0); //
```

```
Attach sensor to pin A0
```

```
float rzero = gasSensor.getRZero();
```

```
Serial.println (rzero);
```

```
delay(1000);
```

```
}
```

we have included the library for the LCD

and have defined the pins for the same.

We have also defined two more variables:  
one for the sensor analog pin and other  
for storing air\_quality value.

```
#include <SoftwareSerial.h>
```

```
#define DEBUG true
```

```
SoftwareSerial esp8266(9,10);
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12,11, 5, 4, 3, 2);
```

```
const int sensorPin= 0;
```

**int air\_quality;**

declare the pin 8 as the output pin where we have connected the buzzer.

lcd.begin(16,2) command will start the LCD to receive data

**pinMode(8, OUTPUT);**

**lcd.begin(16,2);**

**lcd.setCursor (0,0);**

**lcd.print ("IOT ");**

**lcd.setCursor (0,1);**

**lcd.print ("Sensor Warming ");**

**delay(1000);**

send the commands to set the ESP to communicate with the Arduino and show the IP address on the serial monitor.

**Serial.begin(115200);**

**esp8266.begin(115200);**

```
sendData("AT+RST\r\n",2000,DEBUG);
```

using HTML programming. So, we have created a string named webpage and stored the output in it. We are subtracting 48 from the output because the read() function returns the ASCII decimal value and the first decimal number which is 0 starts at 48.

```
if(esp8266.available())
```

```
{
```

```
if(esp8266.find("+IPD,"))
```

```
{
```

```
delay(1000);
```

```
int connectionId = esp8266.read()-48;
```

```
String webpage = "<h1>IOT Air Pollution  
Monitoring System</h1>";
```

```
webpage += "<p><h2>";
```

```
webpage+= " Air Quality is ";
```

```
webpage+= air_quality;
```

```
webpage+=" PPM";
```

```
webpage += "<p>";
```

The following code will call a function named sendData and will send the data & message strings to the webpage to show.

```
sendData(cipSend,1000,DEBUG);
```

```
sendData(webpage,1000,DEBUG);
```

```
cipSend = "AT+CIPSEND=";
```

```
cipSend += connectionId;
```

```
cipSend += ",";
```

```
cipSend +=webpage.length();
```

```
cipSend += "\r\n";
```

code will print the data on the LCD. We have applied various

conditions for checking air quality, and LCD will print the messages according to conditions and buzzer will also beep if the pollution goes beyond 1000 PPM.

```
lcd.setCursor (0, 0);
```

```
lcd.print ("Air Quality is ");
```

```
lcd.print (air_quality);
```

```
lcd.print (" PPM ");
```

```
lcd.setCursor (0,1);
```

```
if (air_quality<=1000)
```

```
{
```

```
lcd.print("Fresh Air");
```

```
digitalWrite(8, LOW);
```

Finally the below function will send and show the data on the webpage. The

data we stored in string named 'webpage' will be saved in string

named 'command'. The ESP will then read the character one by one from the 'command' and will print it on the webpage.

```
String sendData(String command, const  
int timeout, boolean debug)
```

```
{
```

```
String response = "";
```

```
esp8266.print(command); // send the  
read character to the esp8266
```

```
long int time = millis();
```

```
while( (time+timeout) > millis())
```

```
{
```

```
while(esp8266.available())
```

```
{
```

```
// The esp has data so display its output  
to the serial window
```

**char c = esp8266.read(); // read the next character.**

**response+=c;**

**}**

**}**

**if(debug)**

**{**

**Serial.print(response);**

**}**

**return response;**

**}**



# Output:



```
OK
bBÖt@RcâUR%#`BiyÊÉVÜiyÂID
[System Ready, Vendor:www.ai-thinker.com]
AT+CWMODE=2
no change
AT+CIFSR
192.168.4.1
OK
AT+CIPMUX=1
OK
AT+CIPSERVER=1,80
OK
```



## IOT Air Pollution Monitoring System

Air Quality is 977 PPM

Good Air

# Using Blynk App

