**AIR QUALITY MONITORING SYSTEM**

**OBJECTIVE:**

The objective of this circuit is to construct an Indoor air quality monitoring system for detecting toxic gases

**ABSTRACT:**

IoT device has been programmed to collect and transmit data at an interval of one minute over WiFi connection to a gateway node that in turn communicates with the processing node. The sensor was calibrated using the standard calibration methods. As an additional capability, the proposed air pollution monitoring system can generate warnings when the pollution level exceeds beyond a predetermined threshold value.

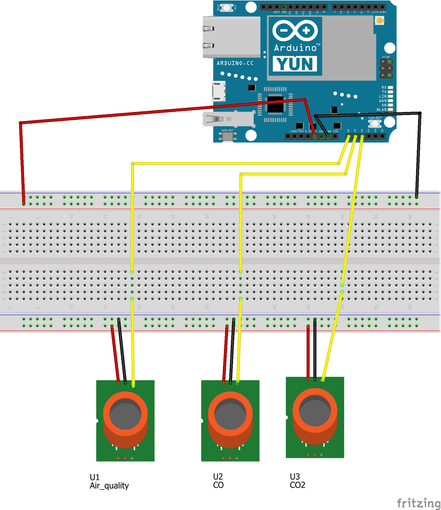
**INTRODUCTION**:

In recent times indoor air quality has attracted the attention of policy makers and researchers as an important similar to that of external air pollution. In certain sense, indoor air quality must be paid more attention than outdoor air quality as people spend more time indoors than outdoors. The indoor environments are confined and closed compared to external environments providing less opportunity for the pollutants to dilute. With the advancement of technology, working places have become more automated using machines to carry out the tasks that were hitherto done manually. These devices emit various solids and gases into the environment during their operation. These emissions contain many substances that are harmful to human health, when exposed to them for a prolonged period of time or more than certain levels of concentration.

**HARDWARE REQUIREMENT/DESCRIPTION:**

* NodeMCU
* Gas sensors (MQ-2)
* Android device
* Jumper wires
* Arduino IDE/Arduino Web IDE
* Blynk App

**CIRCUIT DIAGRAM:**



**CODE**:

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <MQ2Lib.h>

char auth[] = "81fdbab571fc4e3ea6a10d8efe9b9cde";

char ssid[] = "Nityaa";

char pass[] = "nexttech";

int pin = A0;

float lpg = 0, co = 0, smoke = 0;

BlynkTimer timer;

MQ2 mq2(pin, true);

void sendSensor()

{

  Blynk.virtualWrite(V1, lpg);

  delay (200);

  Blynk.virtualWrite(V2, co);

  delay (200);

  Blynk.virtualWrite(V3, smoke);

 }

void setup()

{

  // Debug console

  Serial.begin(9600);

  mq2.begin();

  Blynk.begin(auth, ssid, pass);

  // You can also specify server:

  //Blynk.begin(auth, ssid, pass, "[blynk-cloud.com](http://blynk-cloud.com)", 80);

  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  timer.setInterval(1000L, sendSensor);

}

void loop()

{

  Blynk.run();

  timer.run();

  float\* values= mq2.read(true); //set it false if you don't want to print the values in the Serial

  //Reading specific values:

  //lpg = values[0];

  lpg = mq2.readLPG();

  //co = values[1];

  co = mq2.readCO();

  //smoke = values[2];

  smoke = mq2.readSmoke();

  Serial.print ("CO: "); // output to the serial port

  Serial.println (co);

  Serial.print ("LPG: "); // output to the serial port

  Serial.println (lpg);

  Serial.print ("Smoke: "); // output to the serial port

  Serial.println (smoke);

  delay(100);

}

**APPROACH/WORKING OF PROJECT:**

The MQ-135 alcohol sensor consists of a tin dioxide (SnO2), a perspective layer inside Aluminium Oxide micro tubes (measuring electrodes) and a heating element inside a tubular casing. The end face of the sensor is enclosed by a stainless steel net and the back side holds the connection terminals. Ethyl alcohol present in the breath is oxidized into acetic acid passing through the heat element. With the ethyl alcohol cascade on the tin dioxide sensing layer, the resistance decreases. By using the external load resistance the resistance variation is converted into a suitable voltage variation.

It has lower conductivity compare to clean air and due to air pollution the conductivity is increases. The air quality sensor detects ammonia, nitrogen oxide, smoke, CO2 and other harmful gases. The air quality sensor has a small potentiometer that permits the adjustment of the load resistance of the sensor circuit.

The air quality sensor is a signal output indicator instruction. It has two outputs: analog output and TTL output. The TTL output is low signal light which can be accessed through the IO ports on the microcontroller. The analog output is an concentration, i.e. increasing voltage is directly proportional to increasing concentration. The resistance of the sensor decreases as the concentration of the target gas is increased in PPM while for clean air its resistance remains constant.

The VCC and Ground terminals of the sensor are connected to the common VCC and Ground. The Analog Output pin of the sensor is connected to the A0 pin of the Arduino. The analog output voltage from the sensor can be assumed directly proportional to the concentration of CO2 gas in PPM under standard conditions. The analog voltage is sensed from the sensor and converted to a digital value in range from 0 to 1023 by the inbuilt ADC channel of the controller. The digitized value is hence equal to the gas concentration in PPM.

**PROCEDURE:**

**Step 1**: We will start with getting all the component required for the project as mentioned above. To start with an iot project we need to get both hardware and software needed for the project. So the initial step is to collect all the components required.

**Step 2**: We will start with the assembling the hardware components which involves connecting the Gas Senser(mq-2) to the NodeMCU with jumper wires. Connect the MQ-2 with A0 pin of NodeMCU, then connect the GND(ground) of MQ-2 to the GND of NodeMCU and then connect the VCC pin of MQ-2 to the 3V pin of NodeMCU, this will provide 3 volts of power to the MQ-2. After establishing these connections connect the NodeMCU to your PC via USB cable.

**Step3**: Once you connect the NodeMCU to your PC open Arduino IDE , we can use other IDE as well but Ardunio is compatible with NodeMCU and provides in-built library to for the same. After opening the IDE write the code provided in this report. Remember to select the Board as “NodeMCU 1.0” under the tools dialog box and provide your network settings in the sixth and seventh line of the code provided.

**Step 4**: Download the app “BLYNK” in your smart phone. After logging in this app create a new project and add different gauges to give the LPG, CO,SMOKE reading . The aut token provided by the app to your mail should be entered in the fifth line of the code provided.

**Step 5**: After creating the project on the app, upload the code from the IDE to the NodeMCU, you can note the readings of the various gases on the Blynk app after the code is successfully uploaded.

**CONCLUSIONS:**

Thus, theIndoor Air Quality Monitoring Systemwas constructed and operated.

**REFERENCES:**

https://www.hackster.io/east-west-university/indoor-air-quality-monitoring-system-5b5244?use\_route=project

**APPENDIX:**

**MQ GAS SENSORS**

The MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature.

They can be calibrated more or less (see the section about "Load-resistor" and "Burn-in") but a know concentration of the measured gas or gasses is needed for that.

The output is an analog signal and can be read with an analog input of the Arduino.

**AIR QUALITY SENSOR MQ135**

Air quality sensor for detecting a wide range of gases, including NH3, NOx, alcohol, benzene, smoke and CO2. Ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and particularly suitable for Air quality monitoring application.

**NodeMCU**

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from EspressifSystems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.