Mobile Air Quality Monitor Device

Ahmed M.Al-Khalidi

Department of Electrical and Electronic Engineering

University of Bahrain

Email: ammk6000@gmail.com

Abstract- This project seeks to solve the problem of monitoring public indoor places owned by individuals who don't follow air quality rules and regulations. The solution device will detect in the current phase the concentration of CO and NO2 and these two components are toxic and can cause huge damage to the people who consume the air in these places.

***Keywords*-Air quality, CO detection, NO2 detection, gas sensors, IoT cloud**

I. INTRODUCTION

In countries where the temperature and humidity are too high like the GCC countries, most people spend their time in indoor places like shopping malls, schools, hospitals, and residential buildings. These places are closed and need to have a good ventilation system to supply the buildings with fresh air. If the air quality inside these places is not following the health organization standards this can lead to serious health issues. The aim of this project is to detect the level of the most dangerous chemicals in the air according to the United States environmental protection agency there are some chemicals whose concentration in the air should not exceed the acceptable limit or it will cause serious damage to the people. The chemicals that this device is supposed to detect are Carbon Monoxide and Nitrogen Dioxide [1][2].

II. LITERATURE REVIEW

There are multiple approaches to solving the air quality issue. The following are some of the interesting research that has been published recently.

In the first paper, the researcher uses raspberry pi and gas sensors to measure toxic particles like CO and CO2 indoors and outdoors. The researcher uses the IBM Watson IoT platform for the IoT application [3].

Another interesting research is about Sewage Gas Monitoring we all know how dangerous the gases in the sewage system but that doesn't mean we should not monitor them because sometimes these gases can reach a limit where the worker in this system should go out or wear a mask the idea is to alert the workers. This project uses GSM communication, Arduino, and a gas sensor for the IoT they use to think to speak platform [4].

The next paper discusses another project relating to air quality, but this one is designed to detect industrial gas leakages. The project is designed to work in real time and at a low cost. The device uses the famous gas sensors; MQ2, MQ7, and MQ135 to detect leakages and use the IoT cloud to analyze the data and then send SMS messages as alerts to those who are in the area [5].

The next research is providing rich information about the air quality system design it gives in details information about the common toxic particles and the famous sensors used to detect them, and it gives a comparison of the microcontrollers and single-board computers that can be used to build the air monitoring system [6].

The last paper is discussing a topic relating to my paper, but this research is about monitoring the ozone gas near the photocopy machines in indoor places. The research used a design that depends on Bluetooth and ethernet. The sensor will send the data through Bluetooth to the gateway and the gateway will connect the device to the cloud.

My project is designed to be mobile, and it can detect two toxic gases, CO and NO2. The device works in real time and provides the GPS location for the measurements.

III. HARDWARE AND DESIGN

The components used in this project are the following the Data processing unit, sensors, and GPS unit.

The first part is the data processing unit and I used ESP8266. It is a 32-bit system-on-chip unit it has Bluetooth and Wi-Fi integrated with the chip. The chip is cheap compared with the other solutions available, around 6 dollars. The cheap price allows us to use it in mass production.

Figure1: EESP8266

The microcontroller is the brain of the system. It analyses the signals that come to form the sensors and processes them depending on the program installed in their flash memory. That is why choosing esp8266 is important for this project because it is 32 bit which means it is faster than the other MCU available also the integrated circuits like Wi-Fi and Bluetooth help to reduce the size that this unit takes and this is an important feature because it helps to reduce the size of the whole device which mean we can make small sizes that can be put inside our pocket.

The sensor used in this project is Grove - Multichannel Gas Sensor v2. The sensor has four sensors integrated with it and it uses the I2C protocol to connect with the processing unit. It can detect four different particles in the air which are Carbon monoxide (CO), Nitrogen dioxide (NO2), Ethyl alcohol(C2H5CH), and Volatile Organic Compounds (VOC).

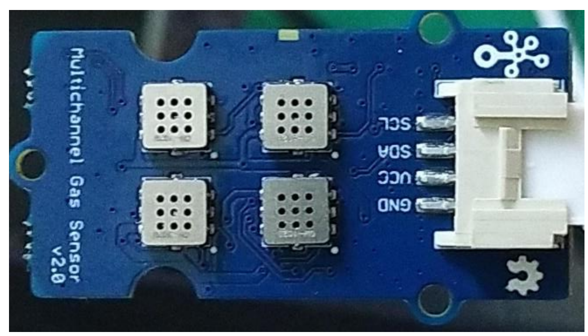

Figure2: Multichannel gas sensor

The sensor is the most important part because it will give us the measurement of the toxic chemicals in the air so accuracy is very important in this part. The level of the gases should not exceed the limit below a table showing the level of CO and what health issues can cause at each level.

Table 1: The level of CO and its effect [8]

CO (Carbon Monoxide) Level	Effects
0 PPM	Normal oxygen level.
9 PPM	Maximum allowable short-term exposure.
10-24 PPM	Possible health effects with long-term exposure.
25 PPM	Maximum allowable for indoor car parks.
50 PPM	Maximum permissible exposure in workplace.
200 PPM	Fatigue and headache after 2-3 hours of exposure.
400 PPM	Headaches within 1-2 hours. Life threatening after 3 hours.
800 PPM	Headache, nausea and dizziness after 45 minutes. Unconsciousness after 1 hour of exposure. Death within 2-3 hours.
1000 PPM	Loss of consciousness after 1 hour of exposure.
1600 PPM	Headache, nausea and dizziness after 20 minutes of exposure. Death within 1-2 hours.
3200 PPM	Headache, nausea and dizziness after 5-10 minutes. Collapse and unconsciousness after 30 minutes of exposure and death within 1 hour.
6400 PPM	Death within 30 minutes.
12,800 PPM	Death within 1-3 minutes of exposure.

PPM = Parts Per Million www.abgo.co.uk

The GPS device used in this project is SKGPS-53 it has the following features Ultra high sensitivity: -165dBm 22 tracks/66 acquisition-channel receivers, WAAS/EGNOS/MSAS/GAGAN support, NMEA protocols, Default Baud Rate: 9600, One serial port, Operating temperature range: -40 to 85℃.

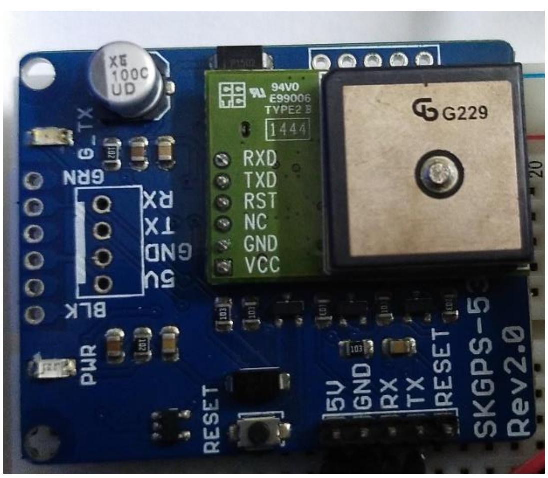

Figure3: GPS unit SKGPS-53

Adding this GPS to the system can bring a lot of advantages one of these advantages is to give strong approval to your measurement that it is accurate and not fake them. Also, it gave the device a security feature where if anyone tries to steal the device, they can find it very easily. Another feature is to help to create a map that includes all the areas that include the most indoor places visited by people the map will show the measurement above each location this map can be shared with the society to make them aware of the levels of the most common toxic chemicals in the places that they spend most of their time there.

The design of this device is as follows the multichannel sensor is connected to the ESP8266 using the I2C protocol, and The GPS unit is connected to the ESP8266 using the UART protocol or the serial protocol. The device is connected to the internet using Wi-Fi. There is an external gateway that connects the device to the internet.

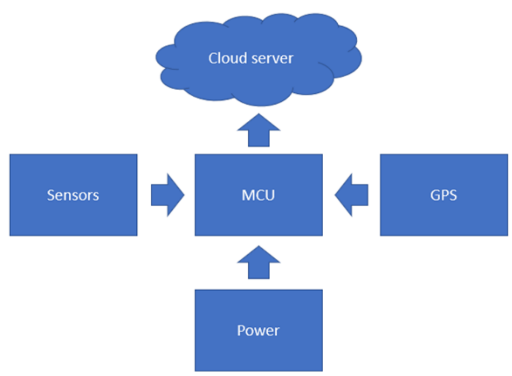

Figure 4: This diagram shows the design.

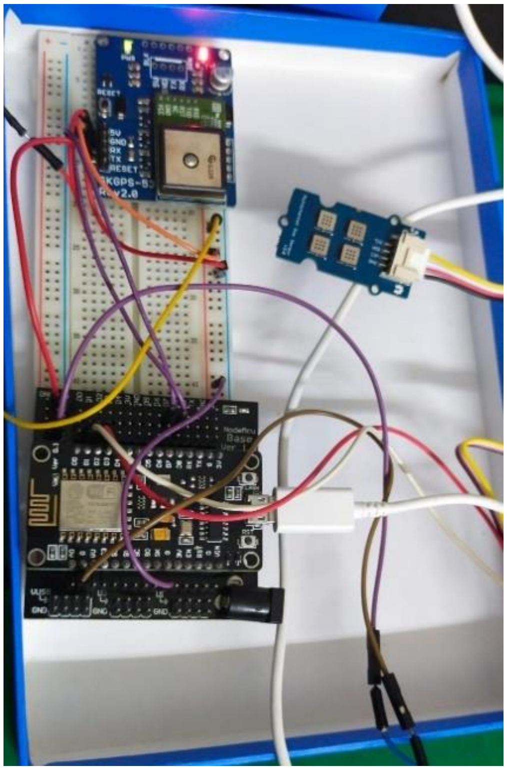

Figure5: The prototype device

IV. IOT SERVICE

The IoT part is done using the Arduino cloud platform. It has many advantages it is easy to use, it has a simple GUI, also you can program your board online without the need to install the ide and all the famous libraries are available on the platforms.

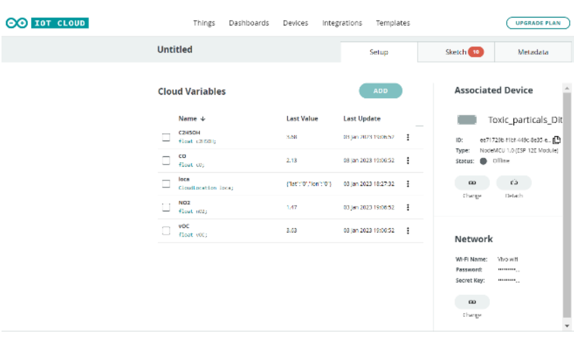

Figure 6: Arduino cloud variables received from the ESP8266.

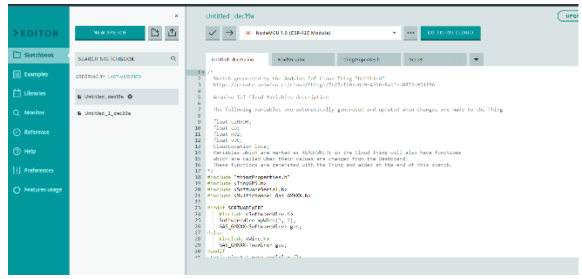

Figure 7: Arduino cloud editor where we can program the board online.

V. RESULTS AND DATA ANALYSIS

The data collected by the sensors are sent to the cloud and in the cloud, we will write the algorithms that analyze these data. Arduino cloud provides good widgets to view the data in a way we can understand.

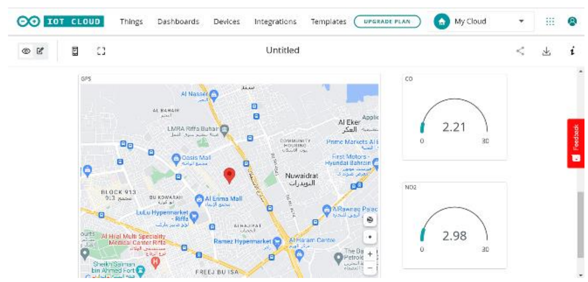

Figure8: Arduino cloud widgets

cloud gives the option to install all the collected data as an excel file so you can use the data in any data analysis application.

	A	B	C	D	E	F
1	time	value				
2	2023-01-01T15:07:51.763Z	{"lat": 0.0	"lon": 0.0}			
3	2023-01-01T15:09:49.223Z	{"lat": 26.:	"lon": 50.579891204833984}			
4	2023-01-01T15:09:50.498Z	{"lat": 26.:	"lon": 50.57990264892578}			
5	2023-01-01T15:09:58.243Z	{"lat": 26.:	"lon": 50.57989501953125}			
6	2023-01-01T15:10:08.823Z	{"lat": 26.:	"lon": 50.57989501953125}			
7	2023-01-01T15:10:17.243Z	{"lat": 26.:	"lon": 50.579864501953125}			
8	2023-01-03T07:01:58.139Z	{"lat": 0.0	"lon": 0.0}			
9	2023-01-03T07:10:13.123Z	{"lat": 26.:	"lon": 50.580039978027344}			
10						
11						

Figure9: GPS reading

	A	B	C	D	E
145	2023-01-03T0	2.22			
146	2023-01-03T0	2.24			
147	2023-01-03T0	2.19			
148	2023-01-03T0	2.2			
149	2023-01-03T0	2.21			
150	2023-01-03T0	2.22			
151	2023-01-03T0	2.21			
152	2023-01-03T0	2.2			
153	2023-01-03T0	2.19			
154	2023-01-03T0	2.21			
155	2023-01-03T0	2.22			
156	2023-01-03T0	2.23			
157	2023-01-03T0	2.23			
158	2023-01-03T0	2.26			
159	2023-01-03T0	2.37			
160	2023-01-03T0	2.35			
161	2023-01-03T0	2.29			
162	2023-01-03T0	2.27			
163	2023-01-03T0	2.29			
164					

Figure 10: CO sensor reading.

For the data analysis, the collected data will be used to determine whether the place is safe to stay in or not. This part is related to phase two which I will be doing to upgrade this project. The current project is collecting data and storing it in the cloud but for the data analysis part, I will use in the next upgradation Machine learning algorithms to do the decision and I will add more sensors like ozone sensors and CO2 sensors, and temperature sensors.

VI. CONCLUSION

In conclusion, this project is an attempt to design a device that can be used by health organizations or reporters, or any person who wants to investigate the air quality in indoor places like schools, factories, or any other place to check if the administration that runs the place are following the standard rules and regulation. This device has the ability to develop into a more complex device that can read more toxic chemicals and we can implement other communications technologies like Lora wan so this device is a platform that can be developed in a way that serves the consumer needs below a suggested design about how this device should look like. Of course, this is just a model and other designs can be used to reduce the size.

Figure11: The case design for the project

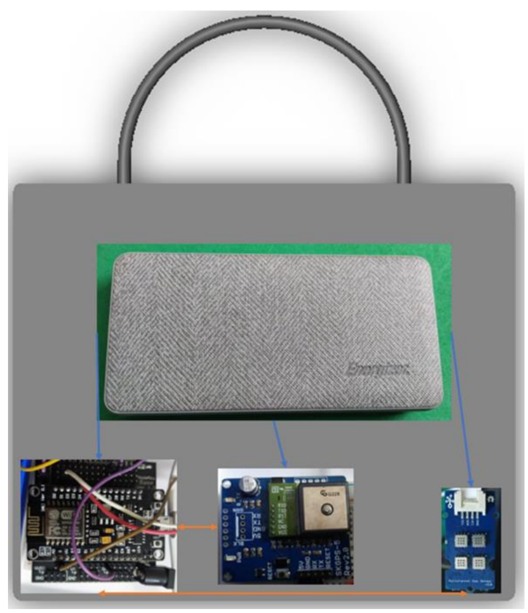

Figure12: The components of the device inside the case

REFERENCES

[1] *EPA*. [Online]. Available: https://www.epa.gov/indoor-air-quality-iaq/carbon-monoxides-impact-indoor-air-quality. [Accessed: 01-Jan-2023].

[2] *EPA*. [Online]. Available: https://www.epa.gov/indoor-air-quality-iaq/nitrogen-dioxides-impact-indoor-air-quality. [Accessed: 01-Jan-2023].

[3] S. Prabha, R. S. Raghav, C. Moulya, K. G. Preethi, and K. S. Sankaran, "Analysis and monitoring air quality system using Raspberry Pi," *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020.

[4] N. Asthana and R. Bahl, "IOT device for sewage gas monitoring and Alert System," *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019.

[5] M. Mahbub, M. A. Rouf, and M. M. Saym, "Industrial Plant Environment Surveillance and Safety Assurance System based on IOT," *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2020.

[6] H. Mokrani, R. Lounas, M. T. Bennai, D. E. Salhi, and R. Djerbi, "Air Quality Monitoring using IOT: A survey," *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019.

[7] M. F. M. Firdhous, B. H. Sudantha, and P. M. Karunaratne, "IOT enabled proactive indoor air quality monitoring system for Sustainable Health Management," *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, 2017.

[8] Gasappliancespares, "What to do if your carbon monoxide alarm sounds," *ABGO Blog*, 17-Apr-2019. [Online]. Available: https://abgoheatingspares.home.blog/2019/04/17/what-to-do-if-your-carbon-monoxide-alarm-sounds/. [Accessed: 17-Jan-2023].

Appendix:

Project code

```
/*

  Sketch generated by the Arduino IoT Cloud Thing "Untitled"

  https://create.arduino.cc/cloud/things/7b27cf60-e619-4266-8a1f-a8872a951f86


  Arduino IoT Cloud Variables description


  The following variables are automatically generated and updated when changes are made to the Thing


  float c2H5OH;

  float cO;

  float nO2;

  float vOC;

  CloudLocation loca;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions

  which are called when their values are changed from the Dashboard.

  These functions are generated with the Thing and added at the end of this sketch.

*/

#include "thingProperties.h"

#include <TinyGPS.h>

#include <SoftwareSerial.h>

#include <Multichannel_Gas_GMXXX.h>


#ifdef SOFTWAREWIRE

    #include <SoftwareWire.h>

    SoftwareWire myWire(3, 2);

    GAS_GMXXX<SoftwareWire> gas;

#else

    #include <Wire.h>

    GAS_GMXXX<TwoWire> gas;

#endif

static uint8_t recv_cmd[8] = {};


SoftwareSerial GPS(13,15);   //RX, TX

//I used softserial because pins 0, and 1 are for

//communicating with pc/laptop


float x1=0;

float x2=0;

float x3=0;

float x4=0;


TinyGPS gps;

void gpsdump(TinyGPS &gps);

bool feedgps();

void getGPS();

long lat, lon;

float LAT, LON;



void setup(){

GPS.begin(9600);      //GPS baud rate 9600 bps

gas.begin(Wire, 0x08); // use the hardware I2C

    //gas.begin(MyWire, 0x08); // use the software I2C

    //gas.setAddress(0x64); change thee I2C address

Serial.begin(115200); //Pc/Laptop to Arduino communication at 115200 bps


 // Defined in thingProperties.h

  initProperties();


  // Connect to Arduino IoT Cloud

  ArduinoCloud.begin(ArduinoIoTPreferredConnection);


  /*

    The following function allows you to obtain more information

    related to the state of network and IoT Cloud connection and errors

    the higher number the more granular information you’ll get.

    The default is 0 (only errors).

    Maximum is 4

  */

  setDebugMessageLevel(2);

  ArduinoCloud.printDebugInfo();
```

```
}

void loop(){
  ArduinoCloud.update();

getGPS();

x1= gas.measure_C2H5OH();

c2H5OH = x1/100;

x2=gas.measure_CO();

cO = x2/100;

x3=gas.measure_NO2();

nO2 = x3/100;

x4=gas.measure_VOC();

vOC = x4/100;

float xx= (LAT/1000000.000000);
float yy=(LON/1000000.000000);

loca = Location(xx,yy);



   delay(2000);



}


void getGPS(){
bool newdata = false;
unsigned long start = millis();
// Every 1 seconds we print an update
while (millis() - start < 1000)
{
  if (feedgps ()){
  newdata = true;
  }
}

if (newdata)
{
  gpsdump(gps);
}


}

bool feedgps(){
while (GPS.available())
{
  if (gps.encode(GPS.read()))
  return true;
}
return 0;
}

void gpsdump(TinyGPS &gps)
{
//byte month, day, hour, minute, second, hundredths;
gps.get_position(&lat, &lon);
LAT = lat;
LON = lon;
{
feedgps(); // If we don't feed the gps during this long routine,
        //we may drop characters and get checksum errors
}
}
```