START BUILDING THE IOT AIR QUALITY MONITORING SYSTEM

Ganesh college of engineering

SUBMITTED BY : C. NANDHINI
T. SWATHI
S.SEVVANTHI
M.ARIVUMATHI

1.Introduction

Nowadays, the existing of air quality detector is already been used but still lack of raising awareness of health risk caused by poor air quality. As an example, the tragedy occurred at Sungai Kim-Kim, Pasir Gudang, due to the chemical dump into the river has affected the health of more than 4,000 people, including students, and forcing the government to close at 111 schools in the district.

The implementation of this device helps in solving this kind of problem by showing the results of air quality level not just on LCD screen but also on a web server. Thus, an air quality monitoring system using IoT has been developed to share the air quality level to the public continuously in order to be aware of the unhealthy environment.

With such system implemented at various locations specifically at industrial area or high population cities, the data can be simultaneously monitored online through the real time information that is sent through the system. Thus, people can take further action to prevent themselves from air pollution that caused by industrial activities as the pollutants are released as by-products of these processes by using any alternative road and not easily expose to the unhealthy environment.

2. Air Quality Monitoring

The primary purpose of an air quality monitoring based on the equipment and systems created is to monitor the air pollution and to distinguish between areas where pollutant levels violate an ambient air quality standard and areas where they do not. As health-based ambient air quality standards are set at level of pollutant concentrations that result in adverse impacts on human health, evidence of levels exceeding an ambient air quality standard in an area requires a public air quality agency to mitigate the corresponding pollutant. This safety equipment become a compulsory in every building especially in industrial area in order to be a aware of the air quality level and avoid the hazardous area.

The existing method, some of the existing instruments which is employ analytical techniques to identify gases for air pollution monitoring are Fourier transform infrared (FTIR) instruments, gas chromatographs and mass spectrometers. These instruments provide fairly accurate and selective gas readings. A gas sensor that is compact, robust with versatile applications and low cost could be an equally effective alternative. Some of the gases monitoring technologies are electrochemical, infrared, catalytic bead, photo ionization and solid-state [3]. One of the large scale sensor networks for monitoring and forecasting is Environment Observation and Forecasting System (EOFS). Air pollution monitoring system based on geo sensor network with control action and adaptive sampling rates proposed in also cannot be vast deployment due to high cost [4].

3. Project Design

The development of this system via XAMPP platform allows the air quality level in parts per million (ppm) data to be stored in an online database, thus allowing the public to continuously monitor the air quality level and avoid themselves to be exposed rapidly to these harmful gases. The developed hardware system consists of the MQ135 gas sensor. The gas sensor is able to sense the present of gases through the chemical reaction when the gases flows close to the sensor. The reading of air quality level appears on an I2C LCD. Figure 1 shows the block diagram of project design. There are two mains part of the design which are hardware design and software design. Figure 2 shows the flowchart of the hardware design and software design.

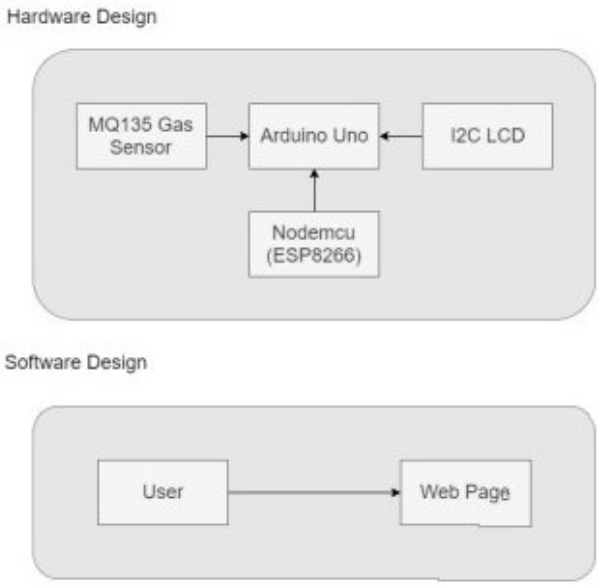

Figure 1: Block diagram of the project design

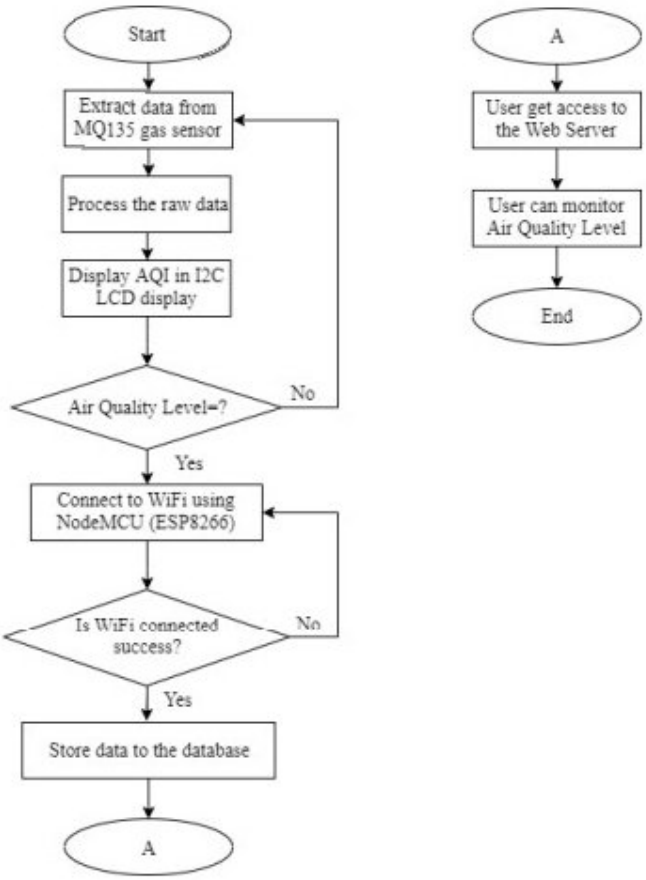

Figure 2: Flowchart of air quality monitoring system

3.1 Hardware Development

A few of suitable components with high performance were used for hardware development. All of the components were at a reasonable cost to integrate to the web server platform of air quality monitoring system. Figure 3 shows the connection of the components for hardware development.

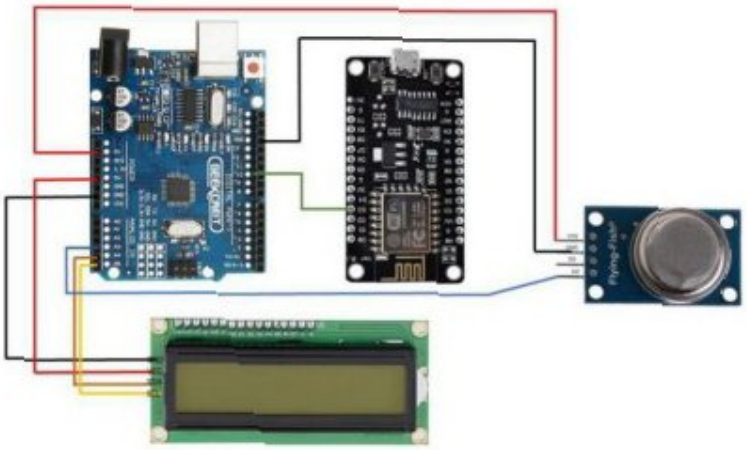

Figure 3: The connection of hardware component

quality level. The GND and VCC pin for Arduino Uno and I2C LCD component are connected to each other. WiFi module, NodeMCU (ESP8266) is connected to the Arduino and act as the bridge between hardware part to the web database.

Arduino Uno has a two ways to supply power in board Arduino Uno through an adapter and USB cable. The adapter has a 12 V that flows in the diode as an output Vin directly to the 5 V regulator. The regulator used to decrease Vin output to the fixed voltage 5 V and the current up to 1 A. The microcontroller Atmega328 are supplied by the voltage 5V and other component such as regulator 3.3 V. Besides, the USB cable has a 5V that flow in the PTC as a dropping currents up to 500mA and other 5V component such as microcontroller and regulator 3.3 V. In addition, it is having an IC comparator to compare the voltage in adapter and USB cable and the priority give to the adapter. However, the USB cable also used to program the Arduino Uno in the Atmega328 that consists of bootloader using a serial communication. The Atmega16U2 in board Arduino Uno must be ensure that the Atmega328 in a bootloader mod before the uploading program using an auto reset circuit. The data from USB cable been change in serial to program the Arduino Uno [5].

The micro controller, NodeMCU (ESP8266) which is used for the project is manufactured by Espressif Systems. The micro controller contains WiFi solution that act as a bridge from Arduino Uno board to the assigned WiFi network. NodeMCU (ESP8266) includes USB connector and a rich assortment of pin-outs.

The LCD screen is an electronic display module that are very commonly used in circuits and devices due to its economical nature and ease of programming as well as its ability in displaying all types of special, custom characters and even animations [6]. The 16x2 refers to its display feature whereby 16 characters can be displayed per line over the 2 lines width of the screen, thus making this LCD a 32-character array. Each character is displayed in a 5x7 pixel matrix, with two working registers incorporated within the LCD, namely Command and Data. The command register stores the instructions given to the LCD to perform a predefined task, whereas the data register stores the ASCII value of the characters to be displayed on the LCD.

The MQ135 sensor has capability to detect some typical dangerous gases. The typical dangerous gases are Ammonia, Alcohol, Benzene, smoke and Carbon dioxide. The MQ135 is a suitable gas sensor to be used in this project. It is attached to Arduino Uno board for capturing the air quality level in parts per million (ppm). The gas sensor translates the gas concentration in the form of voltage level before converting the value into parts per million (ppm). The output conversion from voltage to PPM unit, a library was used for MQ135 sensor by using Arduino microcontroller. In order to calibrate the gas sensor MQ135, a recognized concentration of measured gas is required [7].

3.2 Software Development

The software implemented and programmed for this project include the Arduino Software IDE which allows the Arduino program to be uploaded to the Arduino board, as well as the XAMPP online platform which allows users to monitor data on a web page interface.

The Arduino Software IDE is used for writing of the Arduino programming language or code whereby the program can be uploaded to the Arduino board via the Atmega8U2 chip and USB connection to the computer. This allows the board to perform the functions specified in the program, by which the output data or results of this particular program were displayed in the Serial Monitor feature that is incorporated in the Arduino software. Arduino is an easy tool for fast prototyping which allows the development of projects that are fully-functional at a low cost. Arduino board can apply and adapt to various needs and challenges differentiating its offer from simple 8-bit boards to products for

IoT applications, wearable, 3D printing and embedded environments. Arduino simplifies the process of working with microcontrollers and offers a lot of advantages which include inexpensive, available in cross-platform, allow simple, clear programming environment, open source and extensible software as well as hardware [8].

XAMPP is a free and open-source cross-platform (X) web server software stack package developed by Apache Friends, whereby it consists of the Apache HTTP Server (A), MariaDB database (M), and interpreters that incorporate Hypertext Preprocessor or PHP (P) and Perl Programming Language scripts (P) [9] [10]. The cross-platform feature allows it to run on any computer without the need of an operating system, whereas the MariaDB database is a server developed by the MYSQL team. The PHP server-side scripting language that allows web development to be carried out, and the Perl Programming Language is used for the development of a web application [11]. One of the most attractive features of the XAMPP development tool is ability to allow website interfaces and programming languages to be tested on a local computer without requiring access to the internet [12]. Hence, it allows this website to be initially tested before they are uploaded to a remote web server or computer [11]. An additional tool to this software is a password feature that protects the most crucial parts of the package [13].

The XAMPP software is initially downloaded and installed according to the current Windows specification used by the computer. Once this is done, the Command Prompt (CMD) window can be opened and the internet protocol configuration (ipconfig) command can be entered to determine the Internet Protocol (IP) address used to run XAMPP on the computer [12]. This is illustrated in Figure 4, where the IPv4 address is displayed as 192.168.43.158 in the CMD window, which is used as the initials of the link to the online XAMPP database. At the same time, XAMPP control panel is opened and the two essential ports, namely Apache and MySQL are started up, whilst the 'admin' (administrator) option tool for MySQL is also selected to serve as a workbench that provides a unified visual tool for the database development [12]. This process is depicted in Figure 5.

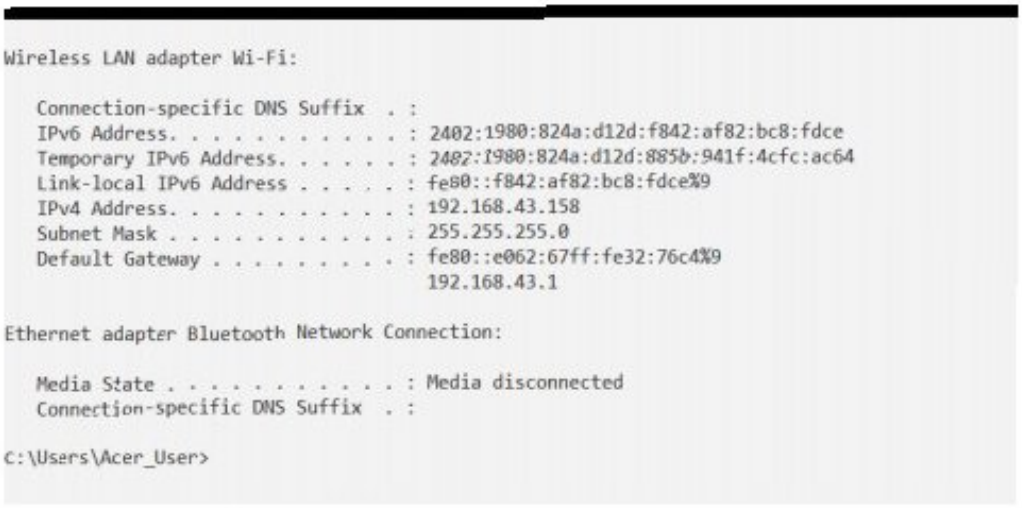

Figure 4: Command Prompt (CMD) windows displaying the IP address

Figure 5: XAMPP control panel windows

Figure 6 shows the main web page of the database displayed online. This database is later used as a platform to store and organize the information of air quality level and time into columns that correlate to the functions of the air quality monitoring system. the database site can be visited by entering the IPv4 address followed by SQLyog, which is the administration tool written in PHP language that is used for the production of these open-sources databases [12].

Figure 6: SQLyog database system

The web page interface is separately developed in the Notepad++ source code editor, which includes the most important home page. Upon completion of both the database and web page, the system is initially tested by adding particular data into the database and subsequently checking if the same information is updated in the web page. If the data transferred is successful, then the system is verified for its functionality. Conversely, an in-depth troubleshooting has to be carried out if the system does not successfully produce the desired response. The entire process of the XAMPP and web page development is depicted in the flowchart provided in Figure 7.

4. Results

The web page interfaces for this system were developed in the Notepad ++ source code editor, with the coding written in Hypertext Preprocessor (PHP) programming language and linked directly to the SQLyog database.

The page allows the user to log into the system in order to monitor air quality level in a real time since the data were sent into the web page for every ten second through the database system as shown in Figure 8.

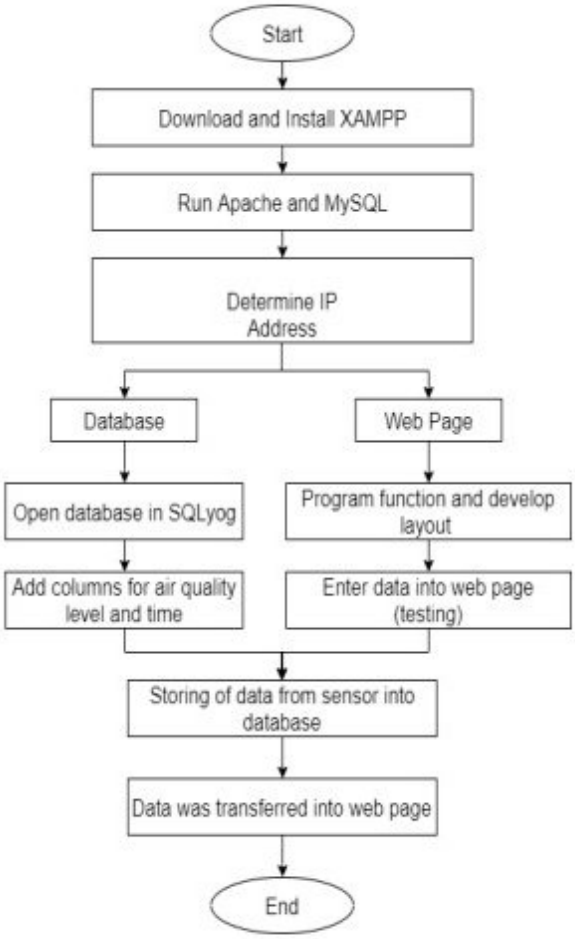

Figure 7: The entire process of XAMPP and web page development

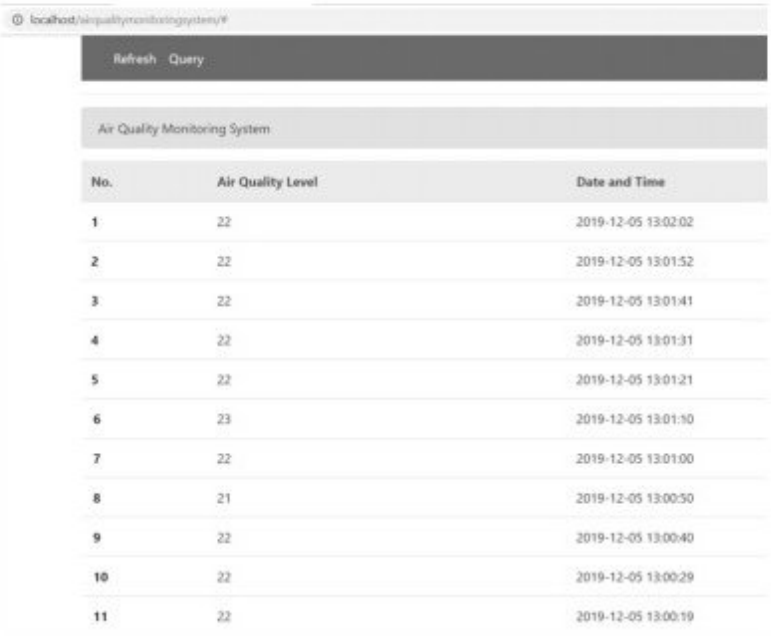

Figure 8: The web page for air quality monitoring system

```
//Program to
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 6, 5, 4, 3);

float t=0;

char data = 0;

// replace with your channel's thingspeak API key
String apiKey = "8NBNB4VQ9F2EEWQM";

// connect 10 to TX of Serial USB
// connect 11 to RX of serial USB
SoftwareSerial ser(10,11); // RX, TX

// this runs once
void setup()
{
  // enable debug serial
  //Serial.begin(9600);
  // enable software serial
  ser.begin(9600);
  lcd.begin(16, 2);

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

```
  lcd.setCursor(0,0);
  lcd.print("Engineers Garage");
  lcd.setCursor(0,1);
  lcd.print("                ");
  delay(3000);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("    IOT AIR");
  lcd.setCursor(0,1);
  lcd.print("QUALITY MONITOR");
  delay(3000);

 // pinMode(12, INPUT);


  // reset ESP8266 WiFi connection AT+CIPMUX=1
AT+CWJAP
  ser.println("AT");
  delay(1000);
  ser.println("AT+GMR");
  delay(1000);
  ser.println("AT+CWMODE=3");
  delay(1000);
  ser.println("AT+RST");
  delay(5000);
  ser.println("AT+CIPMUX=1");
  delay(1000);
```

```
  String
cmd="AT+CWJAP=\"EngineersGarage\",\"egP@$$w0rd?\"";
  ser.println(cmd);
  delay(1000);
  ser.println("AT+CIFSR");
  delay(1000);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("      WIFI");
  lcd.setCursor(0,1);
  lcd.print("   CONNECTED");
 }


// the loop
void loop()
{
  delay(1000);
  t = analogRead(A0);
  Serial.print("Airquality = ");
  Serial.println(t);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  SENDING DATA");
  lcd.setCursor(0,1);
  lcd.print("    TO CLOUD");
  esp_8266();
```

```
}


void esp_8266()

{

   // TCP connection
AT+CIPSTART=4,"TCP","184.106.153.149",80

    String cmd = "AT+CIPSTART=4,\"TCP\",\"";

    cmd += "184.106.153.149"; // api.thingspeak.com

    cmd += "\",80";

    ser.println(cmd);

    Serial.println(cmd);

    if(ser.find("Error"))

    {

      Serial.println("AT+CIPSTART error");

      return;

    }



  // prepare GET string GET
https://api.thingspeak.com/update?
api_key=LHAG4NSIYJ5UWS6U&field1=0\r\n\r\n

  String getStr = "GET /update?api_key=";

  getStr += apiKey;

  //getStr +="&field1=";

  //getStr +=String(h);

  getStr +="&field1=";

  getStr +=String(t);

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

```
      Serial.println("AT+CIPSTART error");
      return;
    }

  // prepare GET string GET
https://api.thingspeak.com/update?
api_key=LHAG4NSIYJ5UWS6U&field1=0\r\n\r\n
  String getStr = "GET /update?api_key=";
  getStr += apiKey;
  //getStr +="&field1=";
  //getStr +=String(h);
  getStr +="&field1=";
  getStr +=String(t);
  getStr += "\r\n\r\n";
  // send data length
  cmd = "AT+CIPSEND=4,";
  cmd += String(getStr.length());
  ser.println(cmd);
  Serial.println(cmd);
  delay(1000);
  ser.print(getStr);
  Serial.println(getStr);

  // thingspeak needs 15 sec delay between updates
  delay(16000);
}
[/restrict]
```

In order for the web page to perform its desired functions, the air quality level file for this page has to be programmed to check whether the air quality level and date and time column has been filled. If the column is filled, the GET request method which accepts the details enclosed within the body of the message is used to determine if the air quality level and time matches to the information stored in the database. Figure 9 shows the line chart data from Google Chart which uses Vector Markup Language (VMP) and Scalable Vector Graphic (SVG). It is needed to create encode within the browser in order to show the level of air quality.

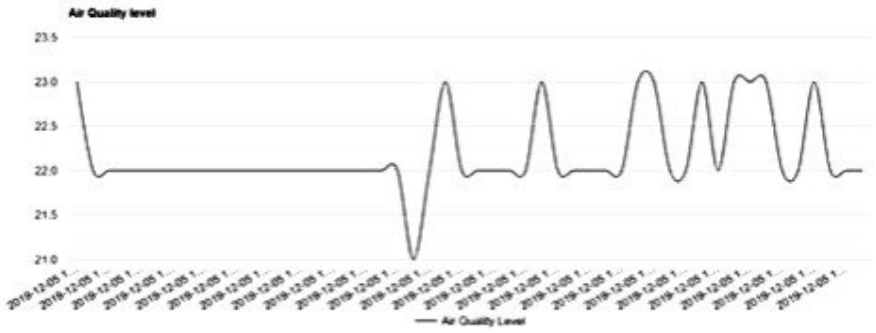

Figure 9: The line chart of air quality monitoring system in normal reading condition

Figure 10, it shows the air quality monitoring system is placed on the table where it should be four to six foot above the ground to record the air quality level in Universiti Tun Hussein Onn Malaysia, Parit Raja. If the air quality level much bigger than the threshold level which is 100, the I2C LCD displays "Unhealthy" on the screen but when the reading is under threshold level then it displays as "Normal".

Figure 10: End product of air quality monitoring system

The air quality monitoring system with IoT have been tested in Universiti Tun Hussein Onn Malaysia (UTHM) for outdoor location. The experiment was done in one day to determine the actual value for normal condition without any existence of harmful gases which means in a safe environment. The normal condition level usually around 15 to 50 ppm which depends on the MQ135 sensor sensitivity in clean air factor resistance. The value start from 50 to 80 which can be defined as lower air pollution presence. Figure 11 shows that the ppm value senses by the MQ135 sensor was in normal condition at UTHM area near Pusat Kesihatan Universiti (PKU). Therefore, the threshold level has been selected setup for 100 parts per million (ppm). In additional, the value of 100 ppm and above can cause to sore eyes, cough and hard breathing. The experiment was scheduled in the morning where the reading slightly increased when there have a few cars passed through the area. Figure 12 shows the reading for air quality index at night time. The value is slightly decreased to 18 ppm when the temperature was decrease due to the rainy weather. The record time for the reading is between 10 to 15 minutes.

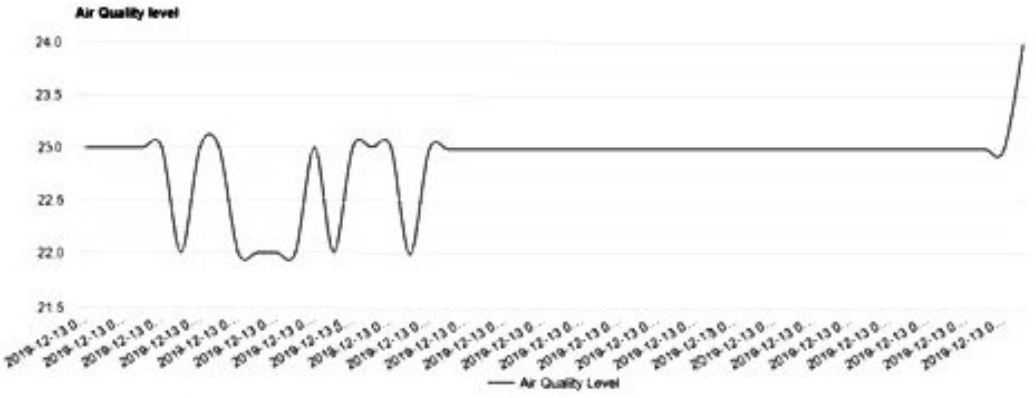

Figure 11: The air quality level in the morning on 13th of December 2019

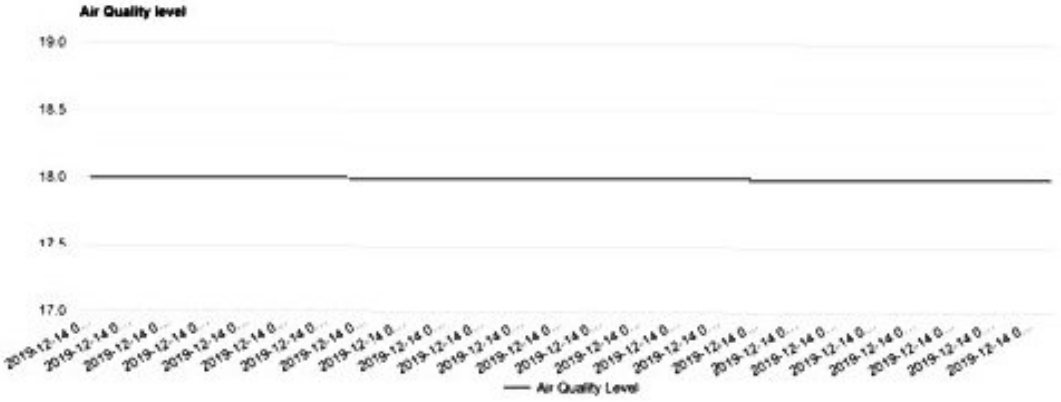

Figure 12: The air quality level in the night on 13th of December 2019

5. Conclusion

As a conclusion, the Air Quality Monitoring System with IoT has achieved the objective that able to detect air pollution or harmful gases by using MQ135 gas sensor. It is also capable of spreading awareness of how important to know the effects of air pollution towards healthy and environment. The following objective which is to develop the monitoring system of air pollution for environmental sensing application using Internet of Things (IoT), has also been fulfilled. The entire system includes the hardware components such as the Arduino Uno microcontroller, MQ135 gas sensor, I2C LCD and NodeMCU (ESP8266) as well as the software components comprising of the Arduino IDE, XAMPP platform and SQLyog for database system. These features are designed in order to work hand-in-hand in order to provide an ideal system for user. The last objective is to verify the function of the system in a different level of air quality level which also been accomplished. The experimental result with a different time and locations in order to get the data collection but for high reading of air quality level which is exceeding to the 100 ppm, couldn't be justified due to the no presence of any harmful gases or air pollution at the location selected. Thus, this ensures that the system is capable of performing the required safety and monitoring purposes.

THANK YOU