International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

IoT Based Air Quality Monitoring System Using Arduino

Shazia Afroze [1], Md. Istiak Hossain Paran [2], Rakibul Hasan Roki [3]

[1]Assistant Professor, Dept. of Electrical & Electronic Engineering, Stamford University Bangladesh (SUB)
[2,3]Student, Dept. of Electrical & Electronic Engineering, Stamford University Bangladesh (SUB)

Abstract
In this work, Internet of Things (IoT) system has been devised to monitor in real time the basic pollutants like Carbon dioxide, Carbon monoxide, Particulate matter, Temperature, Humidity, Pressure. The Primary focus of this paper is to monitor air quality at indoor and outdoor level of residential and industrial sectors by using in total of five sensors, ESP8266 Wi-Fi module and Arduino Uno. The emission from vehicle modeling and vehicle's other emissions can be monitored easily through the internet. A real time air quality monitoring system is essential for Bangladesh especially in Dhaka City because, here the pure air is getting denser and polluted day by day. All sensors real-time data will be displayed on LCD and only the humidity and temperature will be shown on the server. With the help of this project, we have been able to take real-time data of the North and the South side of Dhaka city in February 2023 and plotted curves in Excel. After the analysis of output curves from the Excel plots, we found out our air is very polluted and getting thicker day by day.

Keywords: Air Quality, IoT, Arduino Uno, Embedded Systems, ESP8266, sensors.

Introduction

In the era of fast data transmission over internet, Internet of Things (IoT) has gained a lot of momentum. Air is invisible gaseous substance[1]. On average a person breathes in about approximately 14,000 liters of air. People's health can be affected by the presence of contaminated elements in the air. The Air Quality Indicator (AQI) is calculated and supported on air pollutants like CO, CO_2, NO_2 Compounds that consume opposing possessions happening the atmosphere and human health[1]-[3]. The air of Dhaka city is hazardous for human body as the volume of poisonous particles in air has reached far beyond safe level. Statistics shows that, gas leakage is a common phenomenon in Bangladesh. This project is an implementation of IoT Based Air Pollution Monitoring System Using Arduino. The air around us is getting polluted day by day and it is necessary to monitor air quality for a better future [11]-[12].

Literature Review

This chapter reviews some of the past works in processing and understanding of IoT based Air Quality Monitoring system. Air quality monitoring has gathered a lot of attention and many researchers have developed several microcontroller-based systems . People in Dhaka were able to inhale fresh air only 38 days in the last six years, revealed by the Center for Atmospheric Pollution Studies (CAPS) at Stamford

University Bangladesh. Analyzing six years of data on the air quality index in January CAPS found the average air pollution of level increased 9.8% in 2021 compared to 2020[5]. Temporal variation of CO and CO_2 concentration in indoor environment has been investigated in detail by Kiuriski et al[4]. It concludes the level of these two gases increases to a peak value at certain times in a day and proper ventilation could help in reducing the harmful effect of these gases within the confined environment. Our work takes into consideration this growing problem of air pollution and provides a solution by constantly monitoring air quality and vehicular flow if air is polluted beyond threshold level.

Proposed System and Method

A. Materials:

The primary materials we use in this system are as follows:

- Arduino Uno R3
- ESP8266 Wi-fi module
- MQ135 gas sensor
- MQ7 carbon monoxide sensor
- GP2Y1014AU0F dust sensor
- BMP280
- DHT11

In this study we used Arduino Uno as a microcontroller and ESP8266 as a wi-fi module. The Arduino Uno is a low-cost and low-power microcontroller series integrated with atmega328p microcontroller. The ESP8266 is a low-cost wi-fi microchip, with built in TCP/IP networking software, and microcontroller capability. The Arduino Uno and ESP8266 pin diagrams are given in Fig. 1 and Fig. 2 respectively.

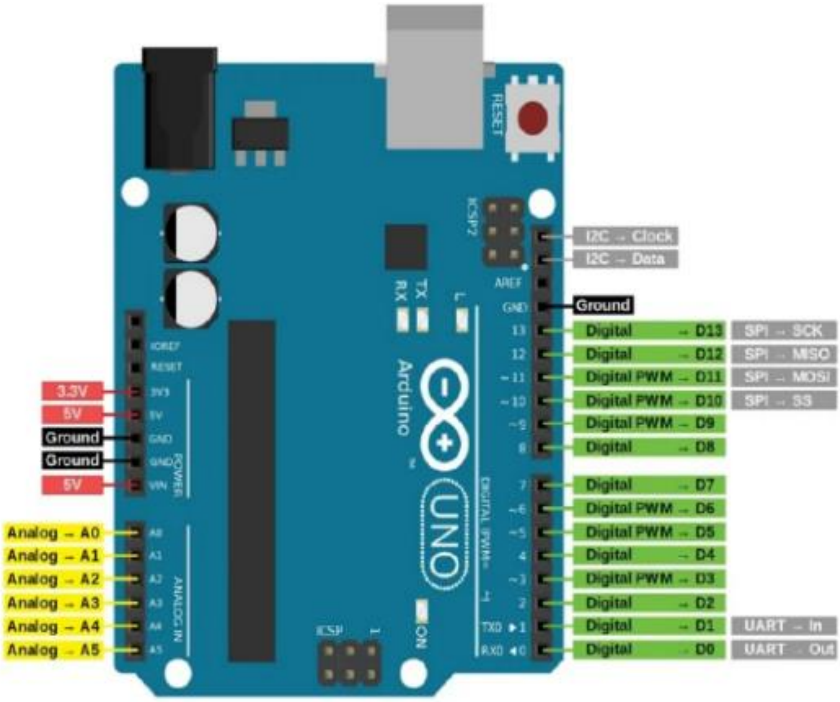
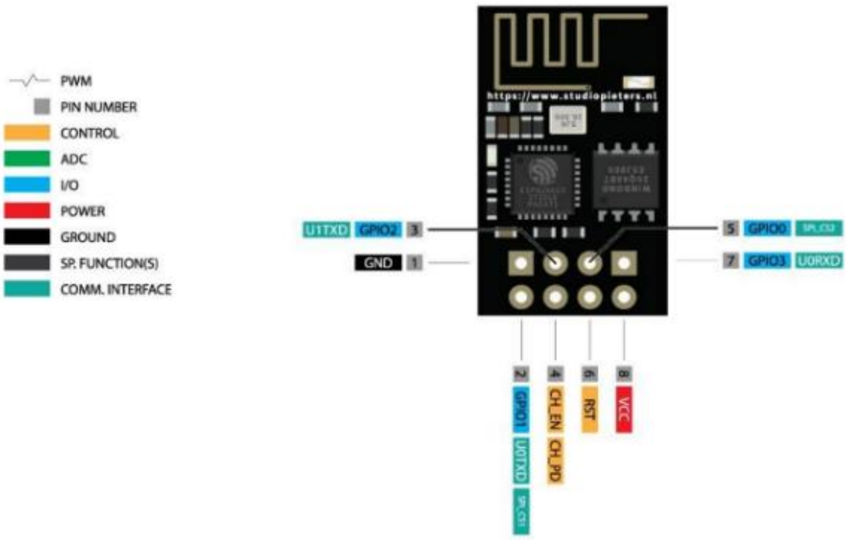


Figure 1: Pin diagram of Arduino Uno

Figure 2: Pin diagram of ESP8266



The gas sensor we used in this system is MQ135 and MQ7. This sensor has a pin for both digital and analog output. The onboard potentiometer can be used to adjust this threshold value. An analog voltage that is produced by the analog output pin can be used to approximatively determine the concentration of various gases in the atmosphere (MQ135). On the other hand, MQ7 only determines the concentration of carbon monoxide in ppm from air. Both of these sensors use the I2C interface.

Figure 3: Pin diagram of MQ135

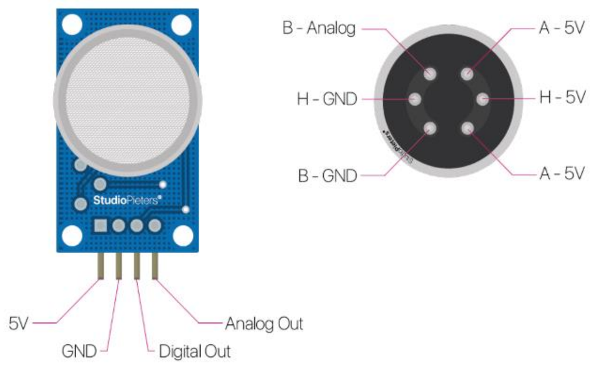
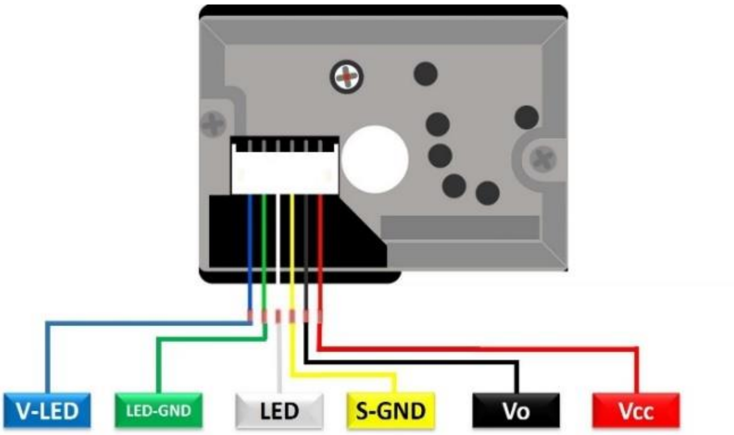


Figure 4: Pin diagram of MQ7

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

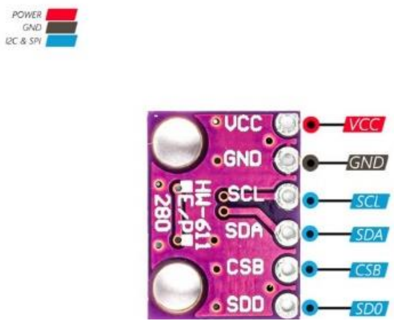
GP2Y1014AU0F dust sensor uses an optical sensing method to detect dust. A photosensor and an infrared light-emitting diode which is known as an IR LED are optically arranged in the dust sensor module. The photo-sensor detects the reflected IR LED rays which bounced off of the dust particles in the air.

Figure 5: Pin diagram of GP2Y1014AU0F dust sensor



BMP280 Barometric Pressure and Altitude sensor I2C/SPI module is a cheapest and tiny atmospheric sensor breakout to measure barometric pressure, altitude and temperature. Here in this system, we have used this sensor only to measure the pressure and altitude.

Figure 6: Pin diagram of BMP280



The DHT11 is a commonly used temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

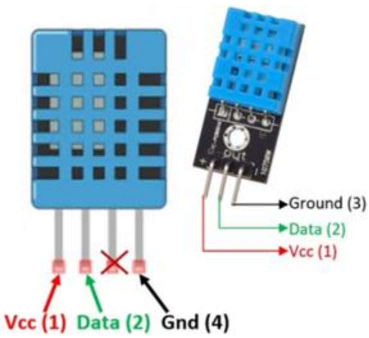


Figure 7: Pin diagram of DHT11

B. System Architecture and Design

The paper aims at designing an air quality monitoring system using Arduino which can be installed in a specific locality and to enhance the system from the previously developed systems beating earlier disadvantages by developing an android app for the public. The project uses Arduino integrated with individual gas sensors like carbon monoxide, carbon dioxide along with particulate matter, humidity, temperature and pressure which measures the concentration of substances separately from air. IoT mainly deals with connecting smart devices to internet by joining the advantage of OSI architecture. In the context of this work, we propose a cluster of Air Quality Monitoring system which uses the sensors: DHT11, MQ07, MQ135, BMP280 and GPY2Y1014AU0F Dust Sensor. These sensors are directly connected to Arduino uno and used ESP8266 chip as a Wi-Fi module to transfer the real time data to the server. The system architecture, design and flowchart are given in Fig. 8 and Fig. 9 and Fig. 10 respectively:

Figure 8: Pin diagram of System Architecture

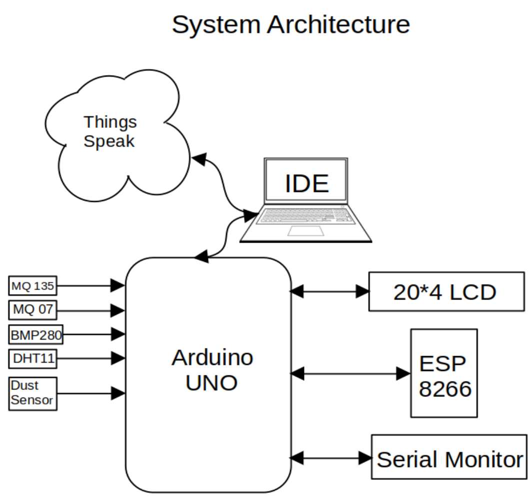


Figure 9: Flowchart for the system

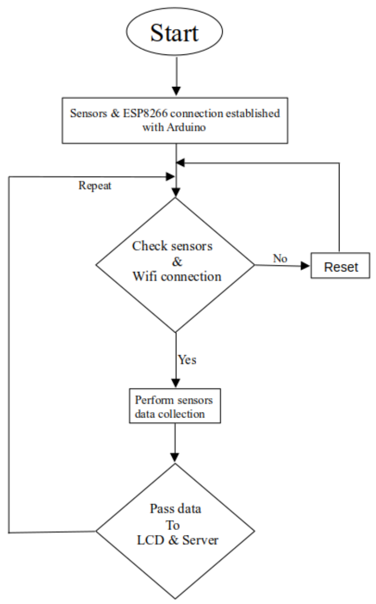
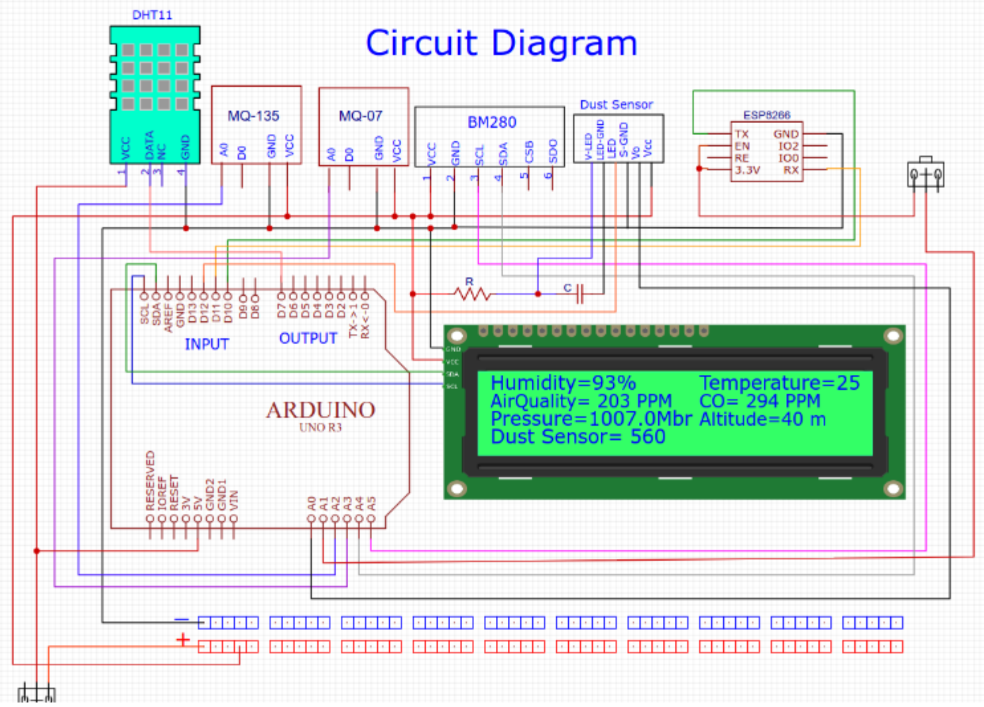
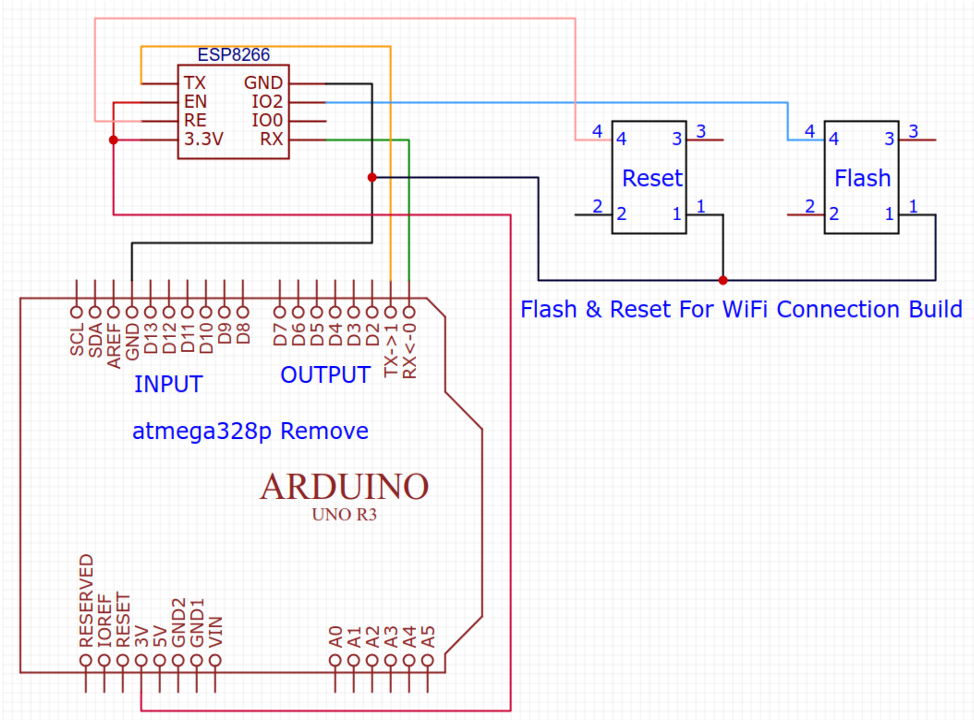


Figure 10: (a) Circuit for the whole system



We have also developed a circuit diagram for enabling us to transfer the real-time data to the server (ThingSpeak). The circuit diagram can be seen in Fig. 10. (b).

Figure 10: (b) Circuit for wi-fi connection



C. Hardware and software configuration

All the sensors are connected with Arduino Uno and the Arduino Uno is connected to ESP8266 to transfer the data to server (ThingSpeak). The picture given in Fig. 10 is the final prototype:

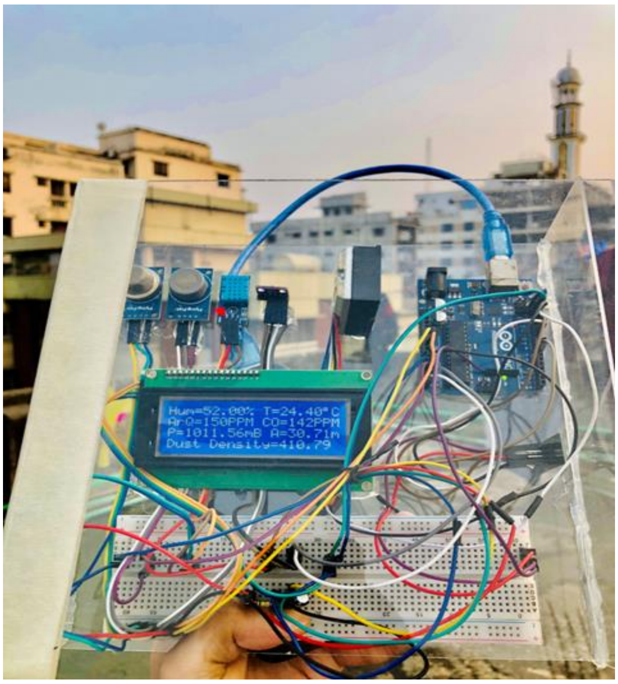


Figure 11: Final Prototype for the proposed system

The following figs contain codes for the system where the code in Fig. 12 is used to show the real-time data on display and the code in Fig. 13. (a) is used to enable wi-fi connection and the code in Fig. 13. (b) is used to send the real-time data of DHT11 to the server (ThingSpeak).

```
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SPI.h>
#define DHTPIN 7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#include <Adafruit_BMP280.h>
#define BMP_SCK (13)
#define BMP_MISO (12)
#define BMP_MOSI (11)
#define BMP_CS (10)
Adafruit_BMP280 bmp;
LiquidCrystal_I2C lcd(0x27, 20, 4);
int measurePin = A0;
int ledPower = 12;
int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
void setup() {
lcd.init();
lcd.backlight();
lcd.begin(20, 4);
dht.begin();

lcd.print(F("BMP280 test"));
unsigned status;
status = bmp.begin(0x76);
bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
Adafruit_BMP280::SAMPLING_X2,
Adafruit_BMP280::SAMPLING_X16,
Adafruit_BMP280::FILTER_X16,
Adafruit_BMP280::STANDBY_MS_500);
pinMode(ledPower, OUTPUT);
}
void loop() {
lcd.clear();
lcd.backlight();

float h = dht.readHumidity();
float t = dht.readTemperature();
```

```
lcd.setCursor(0, 0);
lcd.print("Hum=");
lcd.print(h);
lcd.print("% ");
lcd.setCursor(11, 0);
lcd.print("T=");
lcd.print(t);
lcd.print((char)223);
lcd.print(F("C"));
delay(800);
//Air Quality
int sensorValue = analogRead(A2);
lcd.setCursor(0, 1);
lcd.print("ArQ=");
lcd.print(sensorValue, DEC);
lcd.print("PPM");
delay(800);
// MQ-07
sensorValue = analogRead(A3);
lcd.setCursor(11, 1);
lcd.print("CO=");
lcd.print(sensorValue);
lcd.print("PPM");
delay(800);
// BMP180
lcd.setCursor(0, 2);
lcd.print(F("P="));
lcd.print(bmp.readPressure() / 100);
lcd.print("mB");
lcd.setCursor(12, 2);
lcd.print(F("A="));
lcd.print(bmp.readAltitude(1015.25));
lcd.print("m");
digitalWrite(ledPower, LOW);
delayMicroseconds(samplingTime);
voMeasured = analogRead(measurePin);
delayMicroseconds(deltaTime);
digitalWrite(ledPower, HIGH);
delayMicroseconds(sleepTime);
calcVoltage = voMeasured * (5.0 / 1024);
dustDensity = 170 * calcVoltage - 0.1;
lcd.setCursor(0, 3);
lcd.print("Dust Density=");
lcd.print(dustDensity);
delay(5000);
}
```

Figure 12: Code for the whole system

Figure 13: (a) Code for the wi-fi connection

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
//------- WI-FI details ----------//
char ssid[] = "xxxxxxxxx"; //SSID here
char pass[] = "yyyyyyyyy"; // Passowrd here
//----------- Channel details ----------------//
unsigned long Channel_ID = 12345; // Your Channel ID
const char * myWriteAPIKey = "ABCDEF12345"; //Your write API ke
const int Field_Number_1 = 1;
const int Field_Number_2 = 2;
String value = "";
int value_1 = 0, value_2 = 0;
int x, y;
WiFiClient client;
void setup()
{
Serial.begin(115200);
WiFi.mode(WIFI_STA);
ThingSpeak.begin(client);
internet();
}
void loop()
{
internet();
if (Serial.available() > 0)
    {
      delay(100);
      while (Serial.available() > 0)
     {
      value = Serial.readString();
      if (value[0] == '*')
       {
        if (value[5] == '#')
        {
         value_1 = ((value[1] - 0x30) * 10 + (value[2] - 0x30));
         value_2 = ((value[3] - 0x30) * 10 + (value[4] - 0x30));
        }
      }
    }
   }
   upload();
```

```
}
void internet()
{
 if (WiFi.status() != WL_CONNECTED)
 {
  while (WiFi.status() != WL_CONNECTED)
  {
   WiFi.begin(ssid, pass);
   delay(5000);
  }
 }
}
void upload()
{
 ThingSpeak.writeField(Channel_ID, Field_Number_1, value_1, myWriteAPIKey);
 delay(15000);
 ThingSpeak.writeField(Channel_ID, Field_Number_2, value_2, myWriteAPIKey);
 delay(15000);
 value = "";
}
```

Figure 12: Code for Thingspeak

```
#include <dht.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11);
dht DHT;
#define DHTxxPIN 2
int ack;
void setup()
{
Serial.begin(9600);
mySerial.begin(115200);
}
void loop()
{
 ack = 0;
 int chk = DHT.read11(DHTxxPIN);
 switch (chk)
 {
  case DHTLIB_ERROR_CONNECT:
  ack = 1;
  break;
 }
 if (ack == 0)
 {
  Serial.print("Temperature(*C) = ");
  Serial.println(DHT.temperature, 0);
  Serial.print("Humidity(%) = ");
  Serial.println(DHT.humidity, 0);
  Serial.println("-------------------------");
  //------Sending Data to ESP8266--------//
  mySerial.print('*'); // Starting char
  mySerial.print(DHT.temperature, 0); //2 digi
  mySerial.print(DHT.humidity, 0); //2 digit d
  mySerial.println('#'); // Ending char
  //------------------------------------//
  delay(2000);
 }
 if (ack == 1)
 {
  Serial.print("NO DATA");
  Serial.print("\n\n");
  delay(2000);
 }
}
```

Results & Discussions

The Final prototype that has been presented in Fig. 10 can show real-time data on LCD and pass data to the server. We gathered 7 days of data from South and North City of Dhaka using our prototype and imported it into Excel to plot and compare it between them.

Figure 14: (a) Outdoor Air Quality of South vs North city Dhaka

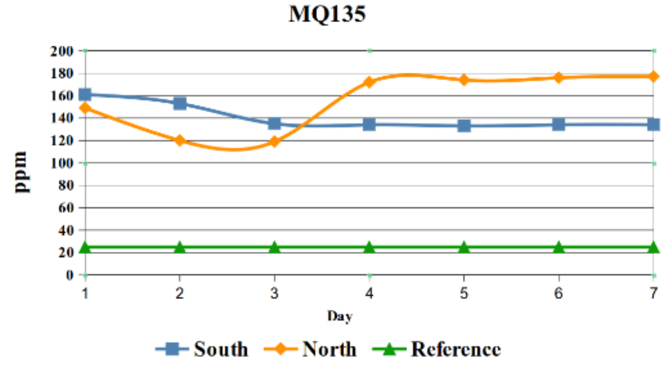


Figure 14: (b) Outdoor Carbon Monoxide of South vs North city Dhaka

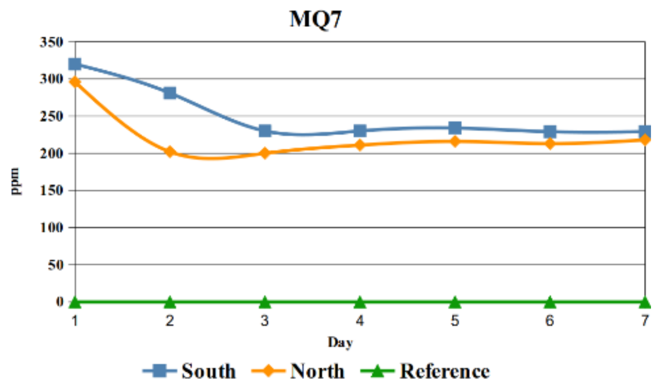


Figure 14: (c) Outdoor PM 2.5 of South vs North city Dhaka

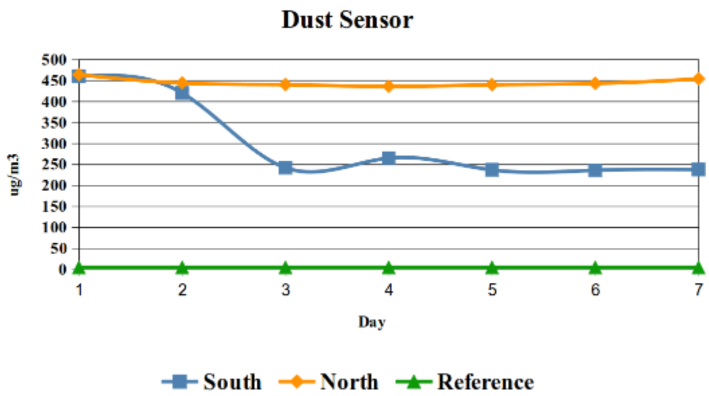


Figure 14: (d) Outdoor Pressure level of South vs North city Dhaka

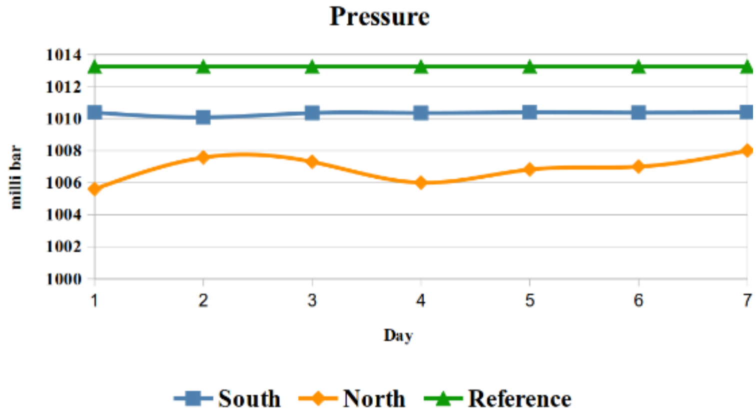


Figure 14: (e) Outdoor Humidity level of South vs North city Dhaka

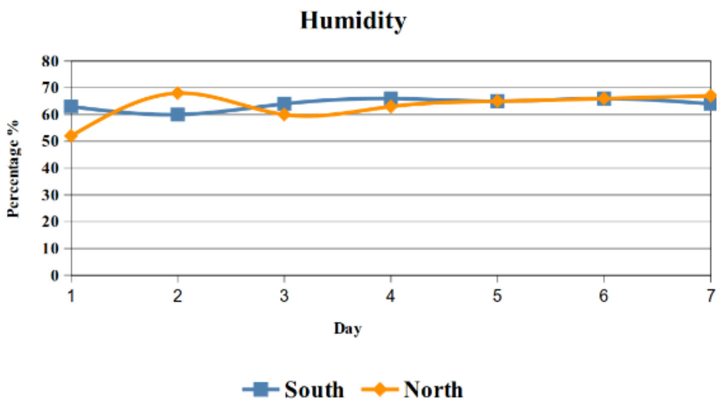


Figure 14: (f) Outdoor Temperature level of South vs North city Dhaka

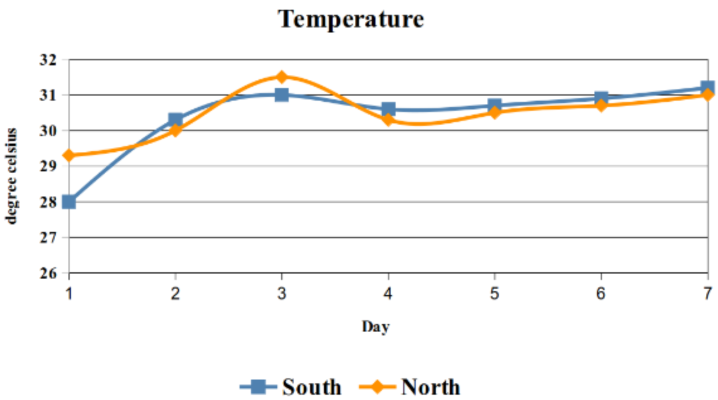
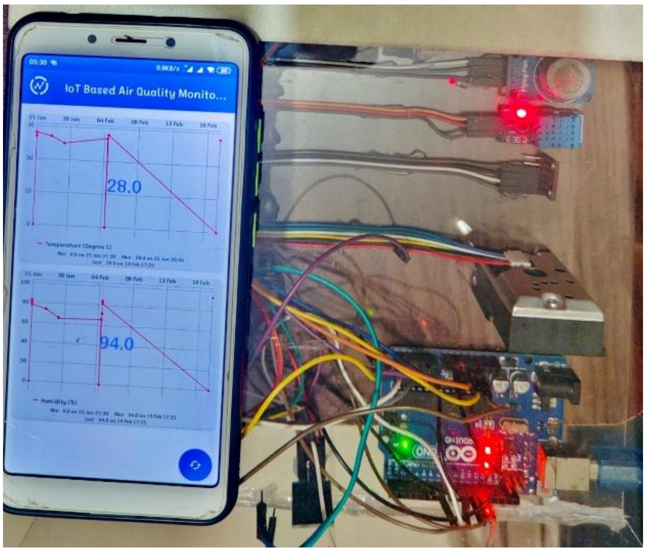


Figure 14: (f) Monitoring real-time Temperature and Humidity of South vs North city Dhaka



Conclusion

The main purpose of this paper was to present a system where it will not only help the environment but also will improve the healthy life cycle of our planets ecosystem by ensuring a good air quality. Air Quality Monitoring allow the measurements, operation and predictive analysis of the evolution of air pollution in different areas. In the near future, this system can be turned into a compact and advanced device that can monitor the data in any type of environment. Further implementation or modifications can help this system to improve better analyze our air quality levels.

Acknowledgement

This work would have not been possible without the support of many people over the last trimester. We would like to express our gratitude to everyone who contributed to it. We are highly indebted to the faculty member of Stamford University Bangladesh for their guidance and constant supervision by Assistant Professor Shazia Afroze as well as for providing necessary information regarding the project & their support in completing it. Last but not least, we owe a debt of gratitude to our loved ones, especially our parents, for their unwavering support and unending love.

References

1. Md. Mohiuddin Ahmed, Suraiya Banu, Bijan Paul, "Real-time Air Quality Monitoring System for Bangladesh's perspective based on Internet of Things", International Conference on Electrical Information and Communication Technology (EICT), 9th December 2017;
2. Pallavi Asthana, Sumita Mishra, "IoT Enabled Real Time Bolt based Indoor Air Quality Monitoring System", International Conference on Computational Techniques and Characterization Techniques in Engineering & Sciences (CCTES), 15th December 2018;
3. Mursil Mahmud, "IoT Based Air Pollution Detection Monitoring System with Arduino", December 2019;

4. J. S. Kiurski, I. B. Oros, V. S. Kecic, I. M. Kovacevic and S. M. Aksentijevic, "The temporal variation of indoor pollutants in photocopying shop", Stochastic Environmental Research and Risk Assessment, vol. 30, no. 4, pp. 1289-1300, 2016;
5. The Business Standard, "Dhaka had just 38 days of fresh air in six years: Study", 27th January 2022;
6. Holli Riebeek, Robert Simmon, "Why Monitor Air Quality", TERRA (The EOS Flagship), NASA, 19th May, 2009;
7. https://www.altexsoft.com/;
8. D.Arunkumar, K.Ajaykanth, M.Ajithkannan, M.Sivasubramanian, "SMART AIR POLLUTION DETECTION AND MONITORING USING IOT", International Journal of Pune and Applied Mathematics, Volume 119, No. 15, 935-941, 2018;
9. Challenges in Internet of Things (IoT), https://www.geeksforgeeks.org/challenges-in internet-of-things-iot/;
10. https://www.engineeringtoolbox.com/ammonia-health-symptoms-d_901.html;
11. https://www.energy.wsu.edu/documents/co2inbuildings.pdf;
12. https://www.co2meter.com/blogs/news/carbon-monoxide-levels-chart;
13. https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in rooms;
14. https://www.epa.vic.gov.au/for-community/environmental-information/air-quality/pm25-particles-in-the-air;