POLITECNICO DI MILANO

Ingegneria Industriale e dell'Informazione

Degree in Telecommunication Engineering

Master of Science

Implementation and Evaluation of a Portable Air Quality Monitoring System

Supervisor: ALESSANDRO ENRICO CESARE REDONDI

Graduation Thesis of:
Fan Wang
Identification Number: 833450
Personal Code: 10479655

Catalog

Abstract

Air quality is very important for the health of human being, especially the particulate matter. Particulate matter can be produced in many ways such as automobile exhaust and industrial waste gas. Hence some particulate matter may contain toxic substance. It can damage respiratory system of human being. So monitoring of particulate matter of environment is a great importance. Hence development of particulate matter monitoring system becomes more and more popular. The aim of this project is to design a small size, portable, low energy consumption particulate matter monitoring prototype system which users can put it in indoor or outdoor environment to monitor PM 1.0, PM 2.5, PM 10.

Chapter 1 - Related Works

There are a lot of air quality monitoring systems now. Most of them are used in indoor environment. For example, there are monitoring systems in indoor environment to monitor factor such as gases (VOC, carbon dioxide, and carbon monoxide), aerosols (PM2.5, PM10) and maybe also include some basic factors such as temperature and humidity. They use Zigbee as transmitting protocol, and in most cases, they have many sensors to form a sensor network. And they have a base station or database to collect data [1] - [7]. In some system, those data can be accessed by internet [1][4][5].

And there are some outdoor air quality monitoring systems too. Some of outdoor air quality monitoring systems put their sensors in a fixed position and the size of sensors is big. Some outdoor environment sensor systems also contain other functions like event detection (Street lamp, landslides, debris flows) [8]. Also there are also some portable outdoor environment monitoring systems like capturing real time air quality data and display it on LCD screen [9].

However, most air quality monitoring systems lack enough portable ability. And they do not associate air quality with monitoring places' precise position, which is longitude and latitude data. Besides WiFi communication is not used frequently. Even though it is used, it is just used as a way to connect a server or internet [10]. It never be used as transmitting method of sensors' data.

Chapter 2 - System Introduction

This chapter is to introduce the whole system. It includes basic concept of the system, what kind of devices the system uses, the working principle of devices, how these devices cooperate with each other, and the software designed for each device.

The basic concept for the whole system is small size, portable, low power consumption, with locating ability particulate matter monitoring system prototype. With these useful features, developers can develop other more complicate systems, such as a particulate matter monitoring device assembled on a bike or on a car.

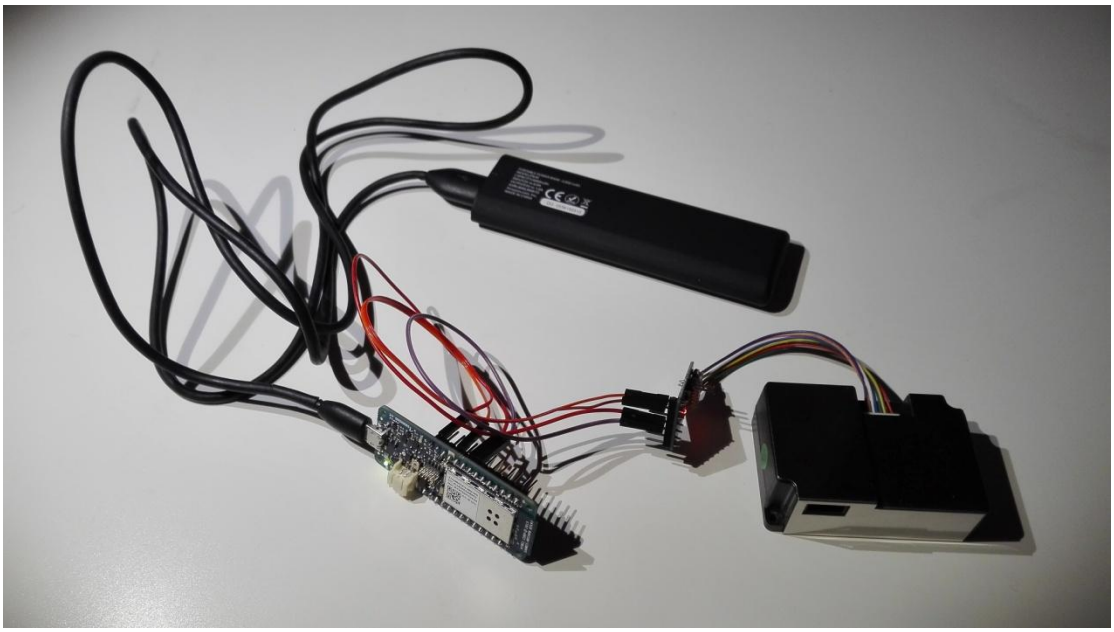
The choices of devices are based on the basic concept. First of all is the choice of sensors. There are many kinds of sensor in market. The system needs a small size, low power consumption sensor. In the end, PM2.5 laser dust sensor SKU sen0177 produced by DFROBOT is chosen. Next is choice of microcontroller. Arduino MKR1000 is chosen because it is assembled with WiFi module which can be used to transmit data and it is low energy cost. The last choice of devices is the device of receiving data. Android smart phone are good choice because smart phone are very popular and easy to use. And all smart phone have WiFi function which can connect with Arduino MKR1000 to receive data. About power supply, system is able to use batteries or direct power supply from socket.

Considering sometimes the system is deployed in a place for a long time, such as a few days or even several months. Using smart phone to receive data is not practical. So the system has another mode to work. In this mode, the system does not send data to Android smart phone, it sends data to an internet platform through WiFi connection. And it uses direct power supply from socket, not batteries. In the end, data is saved in a platform called "ThingSpeak", it can receive and save data that the system transmits to it.

In this article, working with Android smart phone is called mode 1, working with ThingSpeak is called mode 2. The article introduces each working mode separately. For each working mode, the article introduces both hardware and software.

● Mode 1: Working with Android Smart Phone

Smart phone are very popular now. Most of them have WiFi which is a simple and reliable communication protocol. In this project, WiFi is chosen as the communication method. Also smart phone are portable and programmable. Developers are able to design program to process and save data on smart phone. In this project, Android system smart phone are used.



Picture 2-1: Portable Particulate Matter Monitoring System for Working with Android Smart Phone

Besides Android system smart phone, system contains one PM2.5 laser dust sensor, one Arduino MKR1000 board, and one portable power supplying battery. Picture 2-1 shows those 3 devices and the way they connect with each other. These devices compose a portable particulate matter monitoring system. Users are able to take it to any places to monitor air quality there. But using portable power supplying cannot support very long time monitoring.

In this mode, the system needs one Android smart phone. And an Android application whose name is "PM Monitor" has to be installed on that smart phone. This Android application is developed to locate users based on Google Map API, to receive and save data transmitted from Arduino MKR1000. It can also display real time data on smart phone's screen. In this way, users can check PM1.0, PM2.5, PM10 and longitude, latitude data in real time.

Working flow of working mode 1 is showed in below figure 2-2:

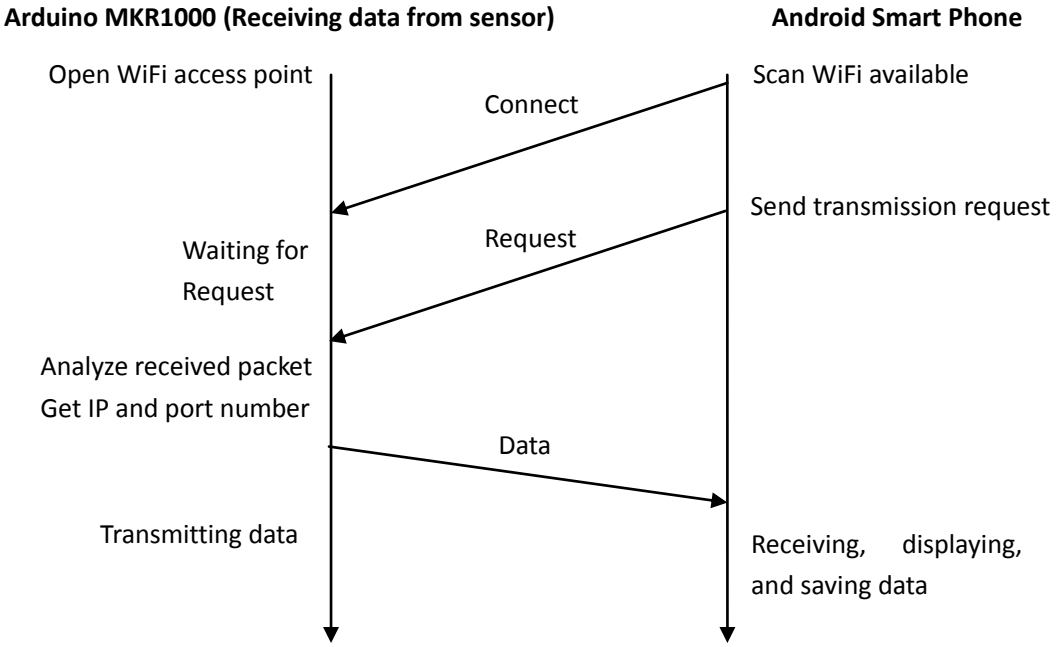


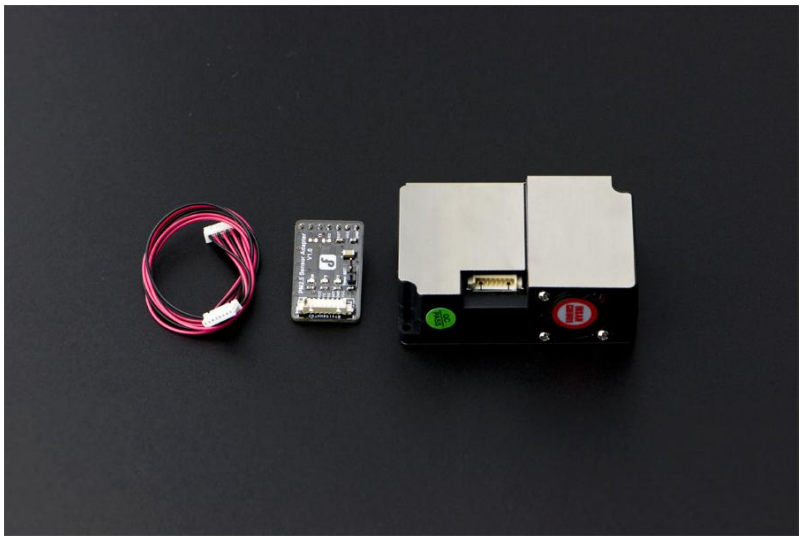
Figure 2-2: Working Flow of Mode 1

Firstly Arduino MKR1000 open WiFi as access point, in this way Android smart phone is able to find the WiFi. Secondly smart phone connects with Arduino MKR1000 using WiFi. When connection is done, users can use the application "PM Monitor" to send transmission request to Arduino MKR1000. When Arduino MKR1000 receives transmission request packet, it analyze the packet to get IP address and port number of the Android smart phone. Based on the IP address and port number, Arduino MKR1000 begin to transmit data to Android smart phone.

Hardware Introduction

This part introduces the hardware detail of mode 1, including parameters of devices, the way of devices connecting and cooperating

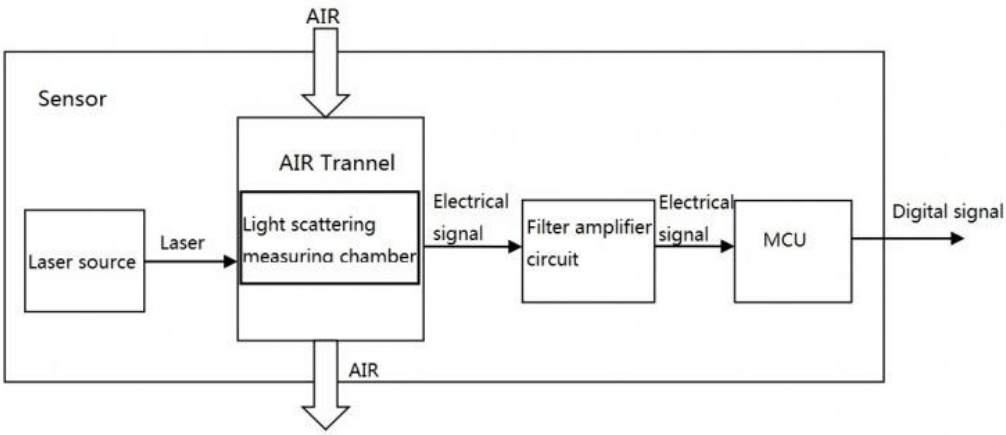
DFROBOT PM 2.5 Laser Dust Sensor SKU: SEN0177



Picture 2-3: The PM2.5 Laser Dust Sensor SEN0177 Produced by DFROBOT

DFROBOT PM 2.5 laser dust sensor comprises of one sensor, one adapter board and connection cable showed in picture 2-3. It can detect particulate matter concentration of air. It can be used to obtain the number of suspended particulate matter in a unit volume of air within 0.3 to 10 microns.

The working principle of this sensor is laser scattering theory. This sensor can collect the scattered light at a specific angle, and obtain the scattering intensity versus with time curve. Using Fourier transform, sensor gets the relationship between the time domain and frequency domain after the microprocessor data collection. And then use a series of complex algorithms to obtain the number of particles in the equivalent particle size and volume units of different size. The working flow is showed in picture 2-4 [11]



Picture 2-4: Working Flow of DFROBOT PM2.5 Laser Dust Sensor

Some important items of PM 2.5 laser dust sensor SKU: SEN0177

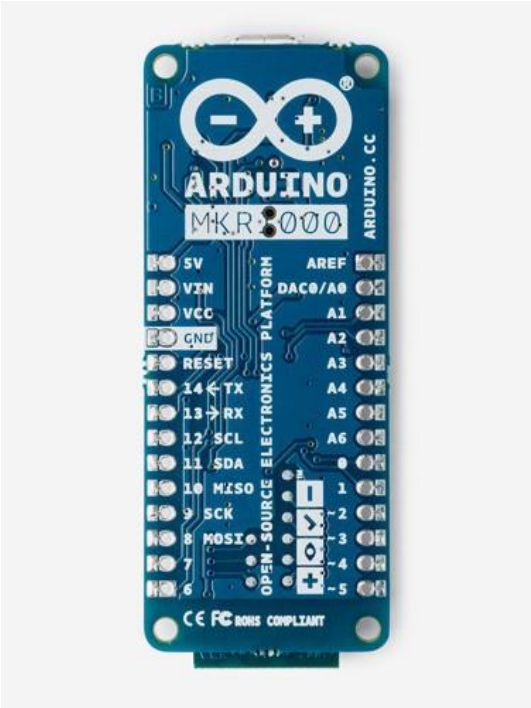
- Operating voltage :4.95 ~ 5.05V
- Maximum electric current: 120mA
- Measuring pm diameter: 0.3~1.0、1.0~2.5、2.5~10（um）
- Measuring pm range：0~500 ug/m3
- Standby current: ≤200 uA
- Response time: ≤10 s
- Operating temperature range:: -20 ~ 50C
- Operating humidity range: 0 ~ 99% RH
- Maximum size: 65 × 42 × 23 (mm)

Pins Of PM 2.5 Laser Dust Sensor SKU: SEN0177

Sensor Pin	Arduino Pin	Function Description
Pin 1	VCC	Positive Power
Pin 2	GND	Negative Power
Pin 3	SET	Mode setting (More hereof later)
Pin 4	RXD	receive serial port pin (3.3V level)
Pin 5	TXD	Transferring serial port pin (3.3V level)
Pin 6	RESET	Reset
Pin 7/ 8	NC	NUll

In this project, pin 1, pin 2, pin 4 and pin 5 are used. Pin 1 and pin 2 are used as power supply pins. Pin 4 and pin 5 are used as serial communication pins. Pin 4 RXD is connected with transmitting port of Arduino MKR1000, pin 5 TXD is connected with receiving port of Arduino MKR1000. In this way, Serial communication can be achieved between PM2.5 laser dust sensor and Arduino MKR1000.

Arduino MKR1000



Picture 2-5: Arduino MKR1000 Front Side

Picture 2-6: Arduino MKR1000 Back Side

Arduino MKR1000 has been designed to offer a practical and cost effective solution for makers seeking to add Wi-Fi connectivity to their Internet of Thing projects. Its

most important feature is it is assembled a WiFi shield which make it more convenient to use WiFi connection. It has one microcontroller Atmel ATSAMW25, and it is composed of 3 blocks [12]:

- SAMD21 Cortex-M0+ 32bit low power ARM MCU
- WINC1500 low power 2.4GHz IEEE® 802.11 b/g/n Wi-Fi
- ECC508 CryptoAuthentication

Arduino MKR1000 can be charged by using battery or USB charger with 5 volts. Program can be uploaded to microcontroller by connecting the board with computer using USB cable and Arduino IDE. This board can also supply 5 volts or 3.3 volts source power to other devices.

Arduino MKR1000 has 28 pins. In this project, the system only uses 4 pins, which are: pin 5V, pin GND, pin 14 TX, and pin 13 RX. Pin 5V and pin GND are used to supply power for PM 2.5 laser dust sensor. Pin 14 and pin 13 are Serial Communication port of this board. Pin 14 is connected with pin 4 of PM2.5 laser dust sensor, and pin 13 is connected with pin 5 of the sensor. They are used to receive data from PM 2.5 laser dust sensor.

Software Introduction

Program of Capturing Data from PM2.5 Laser Dust Sensor in Arduino MKR1000

PM2.5 laser dust sensor has one microprocessor inside which is in charge of processing data and calculating PM1.0, PM2.5 and PM10 value. In this way, data which is captured by Arduino board from the sensor are final result (PM1.0, PM2.5, PM10 value). Arduino MKR1000 only need to read bytes at specific position in the buffer of serial communication port.

Documentation of DFROBOT PM2.5 laser dust sensor introduces specific method of how to capture data from the sensor using Arduino board. It is a section of program

The program in charge of capturing data from sensor in Arduino MKR1000 [11]

```
#define LENG 31   //0x42 + 31 bytes equal to 32 bytes

unsigned char buf[LENG];

int PM01Value=0;         //define PM1.0 value of the air detector module

int PM2_5Value=0;       //define PM2.5 value of the air detector module
```

```
int PM10Value=0;       //define PM10 value of the air detector module


void setup()
{
  Serial1.begin(9600);   //this is the serial communicate with sensor
  Serial.setTimeout(1500);    //set the Timeout to 1500ms, longer than the data transmission periodic time of the sensor
}


void loop()
{
  if(Serial.find(0x42)){    //start to read when detect 0x42
    Serial.readBytes(buf,LENG);


    if(buf[0] == 0x4d){
      if(checkValue(buf,LENG)){
        PM01Value=transmitPM01(buf); //get PM1.0 value
        PM2_5Value=transmitPM2_5(buf);//get PM2.5 value
        PM10Value=transmitPM10(buf); //get PM10 value
      }
    }
  }
```

Function of capturing and calculating data from buffer (Take PM1.0 as example)

```
int transmitPM01(unsigned char *thebuf)
{
  int PM01Val;
  PM01Val=((thebuf[3]<<8) + thebuf[4]); //count PM1.0 value of the air detector module
  return PM01Val;
```

```
}
```

The above program provides the function of reading data from PM2.5 laser dust sensor in Arduino MKR1000. Particulate matter values are saved into those 3 integers: PM01Val, PM25Val , PM10Val. This part is the same both in working mode 1 and working mode 2.

Program Working Flow Used in Arduino MKR1000

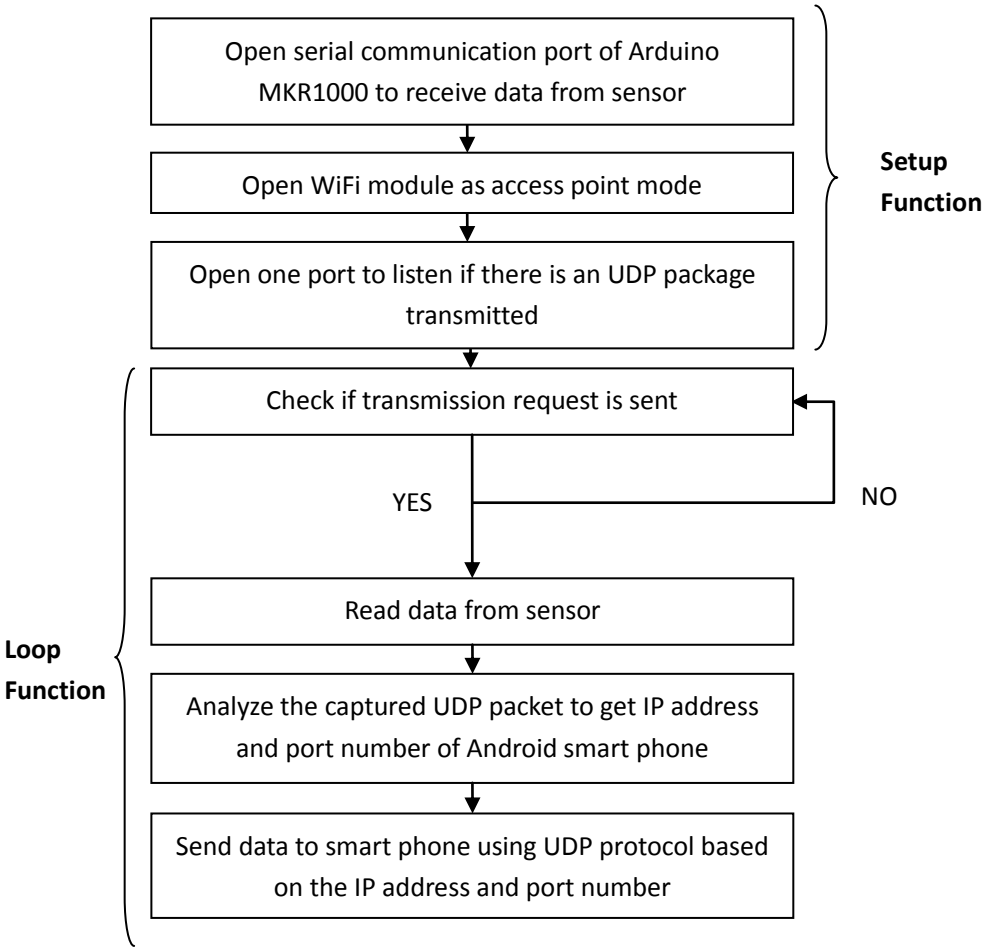


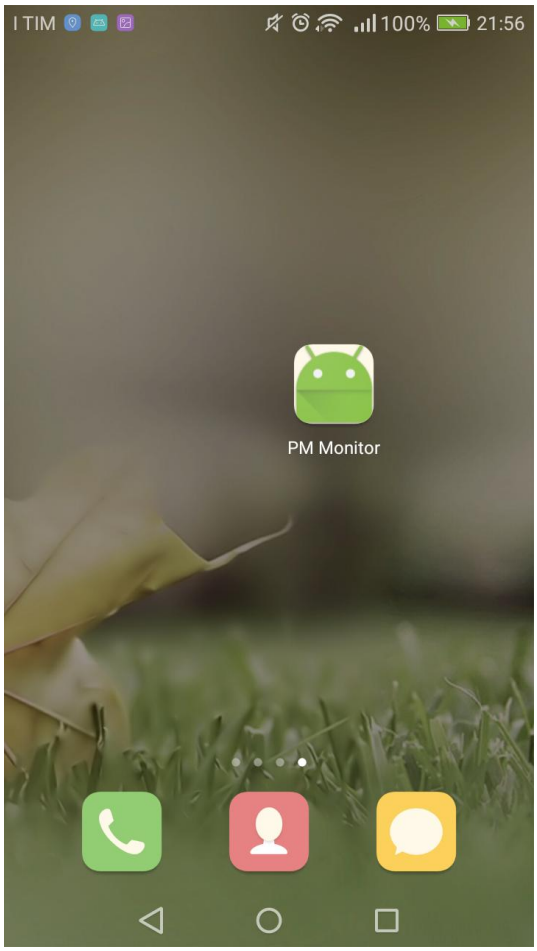
Figure 2-7: Working Flow of Arduino MKR1000 in Mode 1

The working flow is showed in figure 2-7. In setup function, system activates serial communication port (Pin 13 RX and Pin 14 TX) of Arduino MKR1000, which can receive data from PM2.5 laser dust sensor in real time. System opens WiFi module as access point, users can set SSID and password here. Next system opens an UDP port to listen transmission request UDP packet which is transmitted by Android smart phone.

In loop function, system keeps listening at the beginning. If the Android smart phone transmits a transmission request UDP packet to Arduino board, system will analyze the UDP packet to get IP address and port number of the Android smart phone. Based on the IP address and port number, Arduino MKR1000 transmits the data from PM2.5 laser dust sensor to smart phone in real time, the period is about 2 seconds

Android Application

An android application name "PM Monitor" is developed for this project (Showed in picture 2-8). This application is in charge of transmitting transmission request to Arduino MKR1000, receiving data transmitted from Arduino MKR1000, also displaying and saving these data. After Arduino MKR1000 receiving that transmission request, it transmits particulate matter data to Android smart phone in real time.



Picture 2- 8: The Android Application

In this mode, Arduino board works as WiFi access point, users can connect this access point using WiFi function of the smart phone. Data transmission between Arduino board and smart phone is based on UDP protocol.

This application is composed of one activity, one thread, and two buttons. The

activity is the UI activity and the only activity. In this activity, application creates a thread which is in charge of transmitting, receiving and saving data. Those two buttons are responsible for controlling transmitting and receiving data.

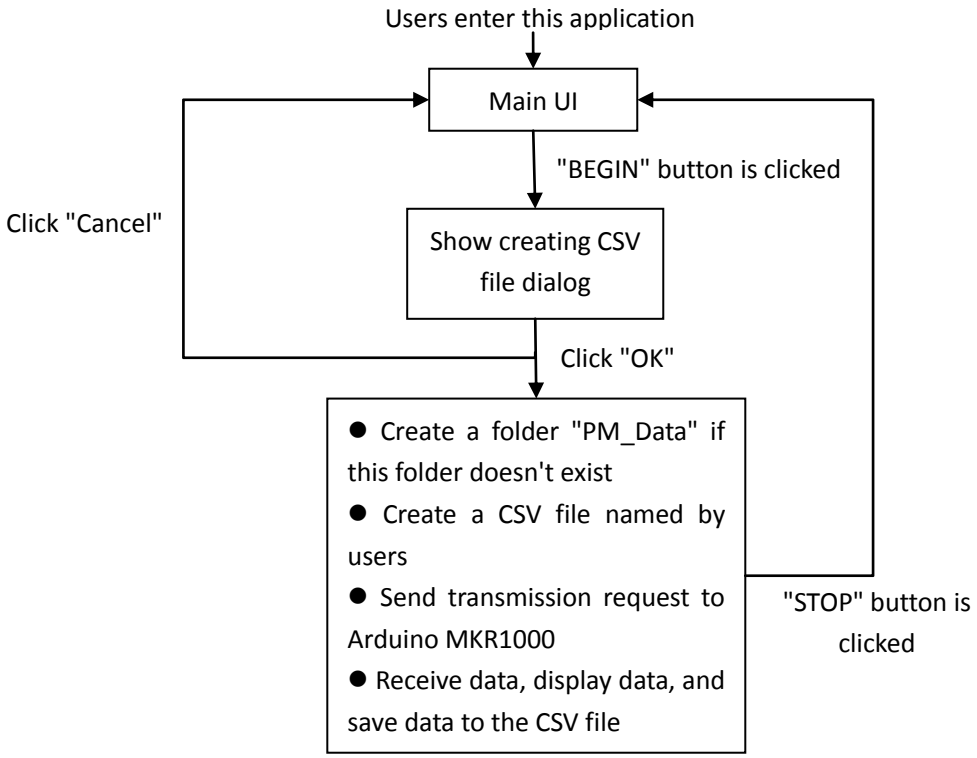


Figure 2-9: Working Flow of Android Application "PM Monitor"

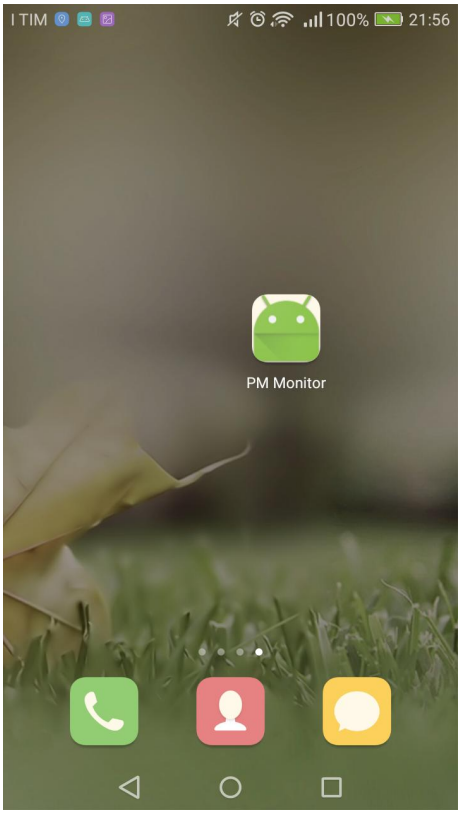
Figure 2-9 shows simple working flow of this Android application. The following content introduces the detail of how to use this application.

If users click this application, the main user interface showed in picture 2-11 will be seen. In that case, data has not been transmitted yet. The text view on top of screen shows nothing.

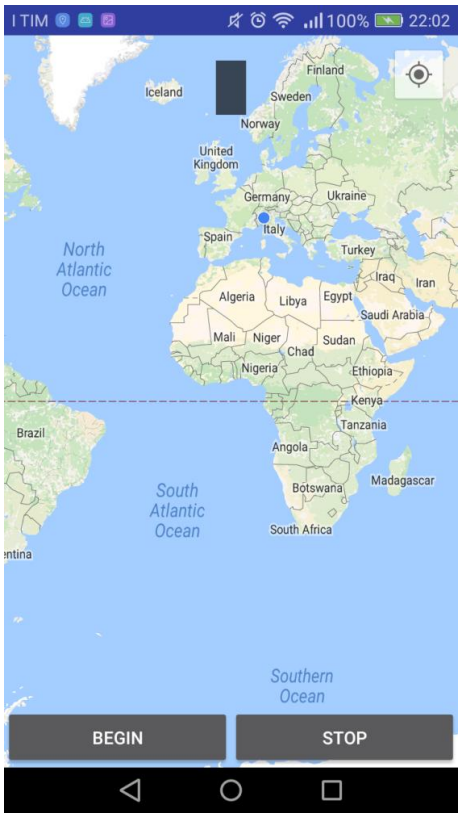
If users click button on top right, the application will locate users automatically, like picture 2-12 shows. At the bottom, there are 2 buttons, one is called "BEGIN", and the other is called "STOP".

BEGIN button controls creating a folder (If there is no folder called "PM_Data" in internal storage of Android smart phone), creating CSV file for storing data, and sending transmission request to Arduino MKR1000 to begin transmission. If users click this button, application will go into picture 2-13 where users can type the name of CSV file they want to store captured data. If users continue to choose "OK", application will create the CSV file, and begin transmitting like showed in picture 2-14. "STOP" button is used in this case to stop transmitting data. After this, users can go

into internal storage of Android smart phone, open a folder called "PM_Data", the CSV file they created can be found in this folder (Picture 2-15).

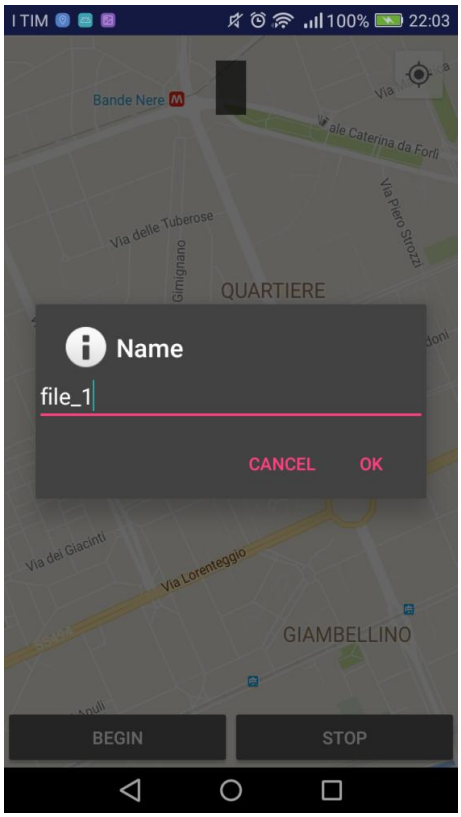


Picture 2-10: Application

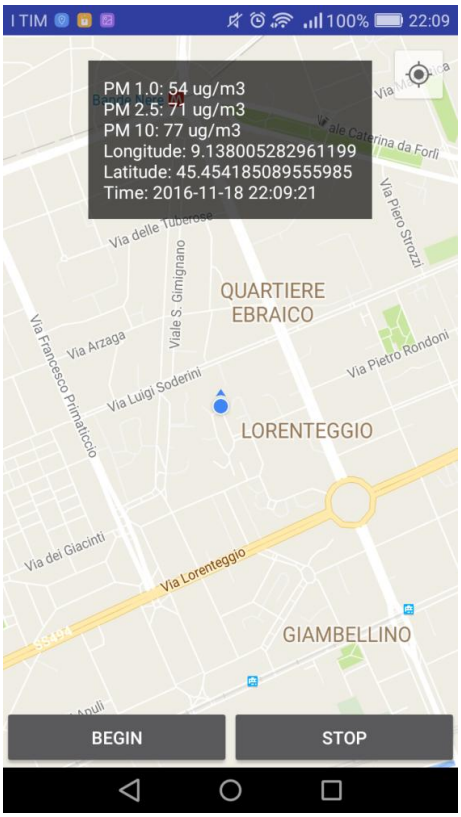


Picture 2-11: Main UI

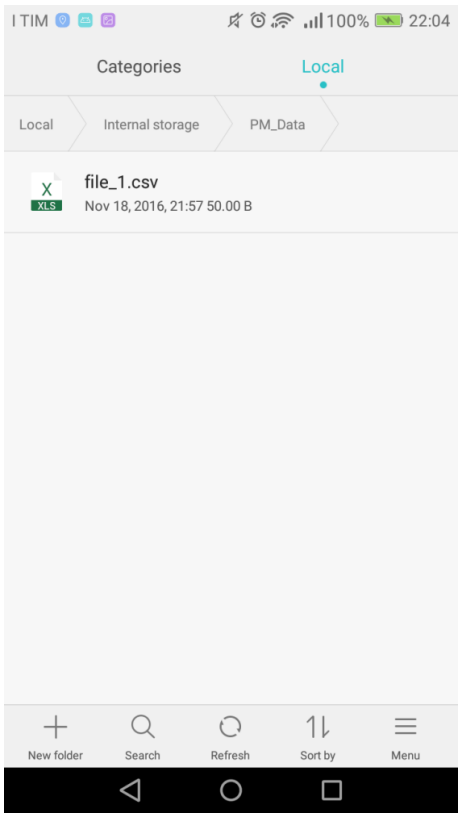
Picture 2-12: Click "My Location" Button



Picture 2-13: Click "BEGIN" Button



Picture 2-14: Click "OK" and Start to work



Picture 2-15: CSV file in "PM_Data" folder

This application uses Google Map API. Implementing the programming interface,

application can locate users automatically. To do this, application need request GPS and network location permission. With this permission, application can get longitude and latitude data in real time. Application stores these data in one CSV file that users create.

The most important part of this application is transmitting, receiving and saving data. This work is done in one thread mostly. As the figures show above, when users create the CSV file, the thread starts to work.

Working Flow of Thread

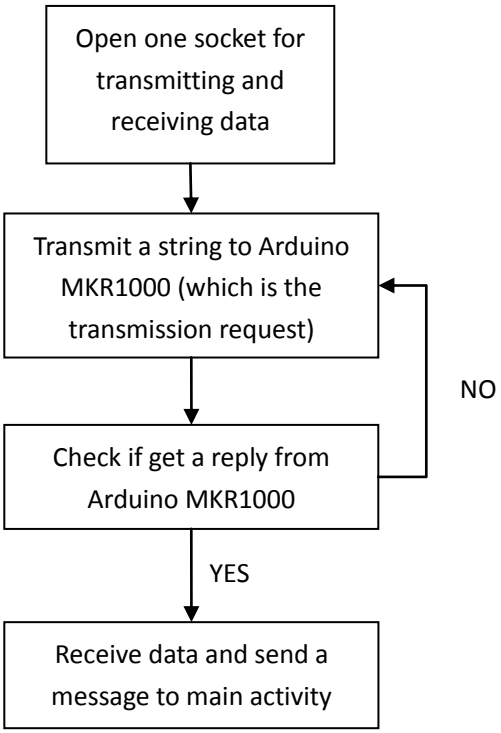


Figure 2-16: Working Flow of Thread

When the thread starts to run, it opens a socket for transmitting and receiving data. And then it transmits a string which is also the transmission request to Arduino MKR1000 and waits for replying. The reply is data comprise of PM values. When the thread receives one data, it sends a message to main activity (Showed in figure 2-16).

Hence, displaying and saving data are not done in this thread. It is done in main activity. Every time the thread receives a data from Arduino MKR1000, it sends a message to main activity. When main activity receives the message, it triggers a handle function which is in charge of displaying received data on screen of smart phone, and saving the data into CSV file that users create.

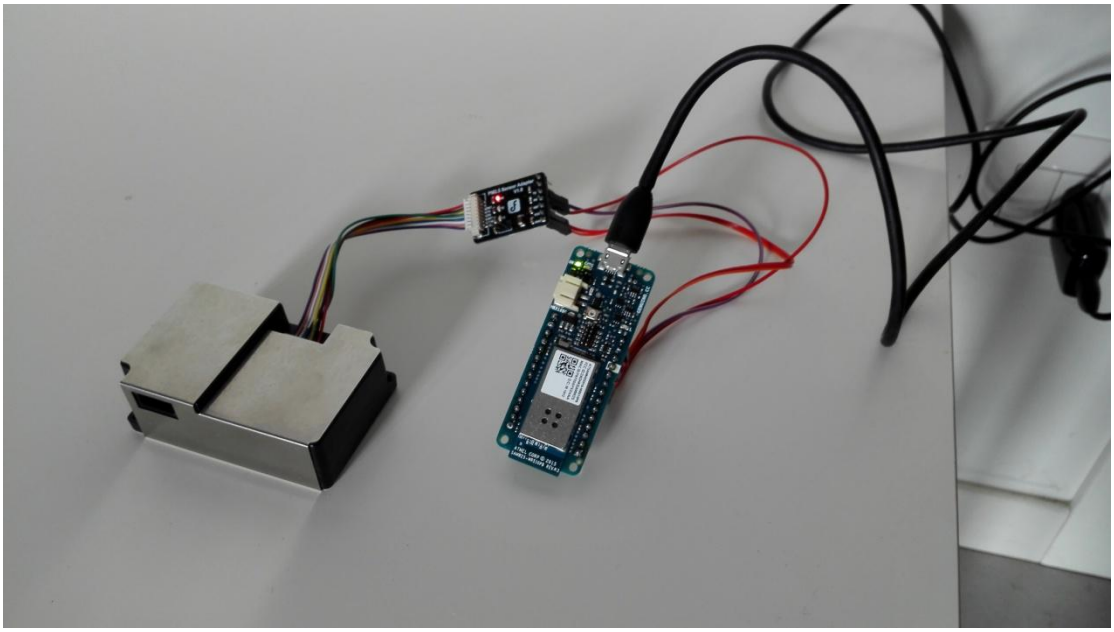
The data that to be saved into CSV file are: PM1.0, PM2.5, PM10 value, time point of

those PM value, Longitude and latitude of the location in which users are at that time. All CSV files are saved in a folder named "PM_Data". The folder can be found in internal storage of smart phone. If there is no such a folder, application will create one when users use this application at the first time

● Mode 2: Working With ThingSpeak

Hardware Introduction

This mode is especially developed for long time air quality monitoring, since using Android smart in long time monitoring is not practical. Arduino MKR1000 has WiFi module, it has ability to access internet through WiFi access point. So instead of using smart phone, this mode chooses to use an internet platform called "ThingSpeak" which can also receive and store data likes smart phone. Also the power supply method is using direct power supply from socket instead of portable batteries.



Picture 2-17: Portable Particulate Matter Monitoring System for Working with ThingSpeak

Picture 2-17 shows the hardware of working with ThingSpeak mode. It is powered by a USB smart phone charger whose output voltage is 5 V.

In this working mode, the function of capturing PM data from PM2.5 laser dust sensor is the same with Mode 1. The difference is how to process the captured data.

In this system, the connection between Arduino MKR1000 and PM2.5 laser dust sensor is the same with mode 1. Pin 1 is connected with pin 5 V of Arduino MKR1000

and pin 2 is connected with pin GND of Arduino board. These 2 pins supply power for this sensor. Pin 4 is connected with pin 14 TX of Arduino board. And pin 5 is connected with pin 13 RX of Arduino board. These 2 pins provide data transmitting channel using serial communication.

Charger

Only the Arduino MKR1000 needs power supply, the PM2.5 laser dust sensor can be powered by the board. According to charger for the Arduino board, system can accept any USB port charger whose voltage is 5 V. In this mode, direct power supply from socket is the best choice.

WiFi Access Point

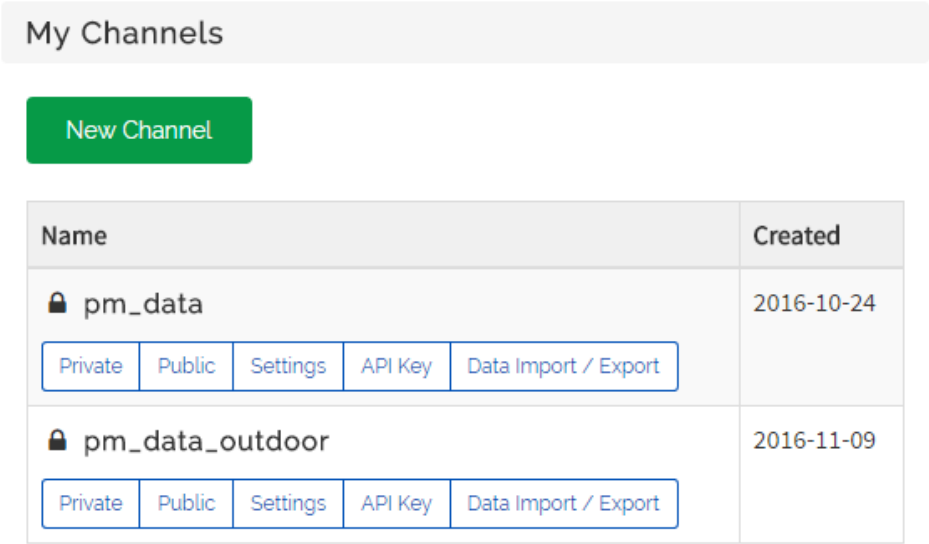
Working in mode 2, system needs one WiFi access point which is connected with internet. Also system requires SSID and password of the WiFi access point to connect.

Software Introduction

ThingSpeak Platform Introduction

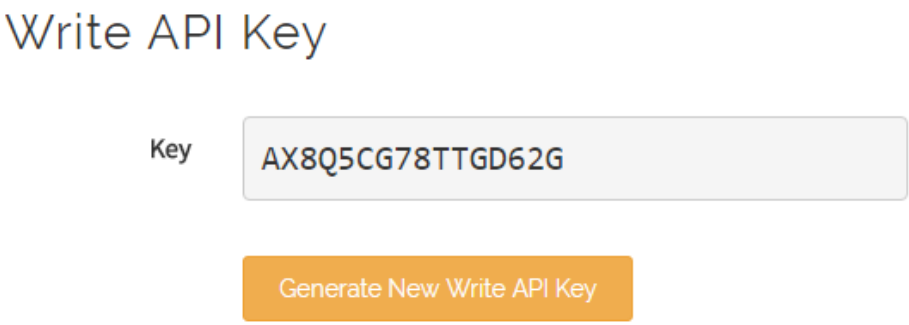
ThingSpeak is an open source Internet of Things platform. It has API which can help users to upload and store data using HTTP protocol over internet. ThingSpeak provides sensor logging application, location track application and so on [13].

Users can create channels which can receive data from internet using ThingSpeak (The channels examples are showed in picture 2-18). This system created 2 channels to receive data. One channel is for indoor environment monitoring, and the other channel is for outdoor environment monitoring. Both monitoring last 48 hours.



Picture 2-18: ThingSpeak Channels

After users creating one channel, ThingSpeak provides users an API key (Showed in picture 2-19). If users need to write data to the channel, they need the specific API key for that channel.



Picture 2-19: The API Key for One Channel

Program of Sending HTTP Request

The system uses HTTP request function to achieve uploading data to ThingSpeak. The HTTP request function is showed in following:

```
void updateThingSpeak(String tsData) {
  if (client.connect(thingSpeakAddress, 80)) {
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + APIKey + "\n");
```

```
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(tsData.length());
    client.print("\n\n");
    client.print(tsData);
    lastConnectionTime = millis();
  }
```

In above program, the connection type is "close". So the system disconnects with ThingSpeak server every time it uploads one data to ThingSpeak.

One important issue is ThingSpeak has limit on period of uploading data. If the period is too short, ThingSpeak server will lose data. To solve this, the system sets the period as 20 seconds which is safe enough. After transmitting one data, system waits for 20 seconds to transmit the next.

Another important issue is WiFi connection. WiFi connection between Arduino MKR1000 and WiFi access point may disconnect during monitoring process. Hence in the loop function of Arduino MKR1000, there is a function which is responsible for detecting WiFi status. If the status is not connected, the system reconnects with WiFi access point.

Program Working flow Used in Arduino MKR1000

The working flow is showed in figure2-20

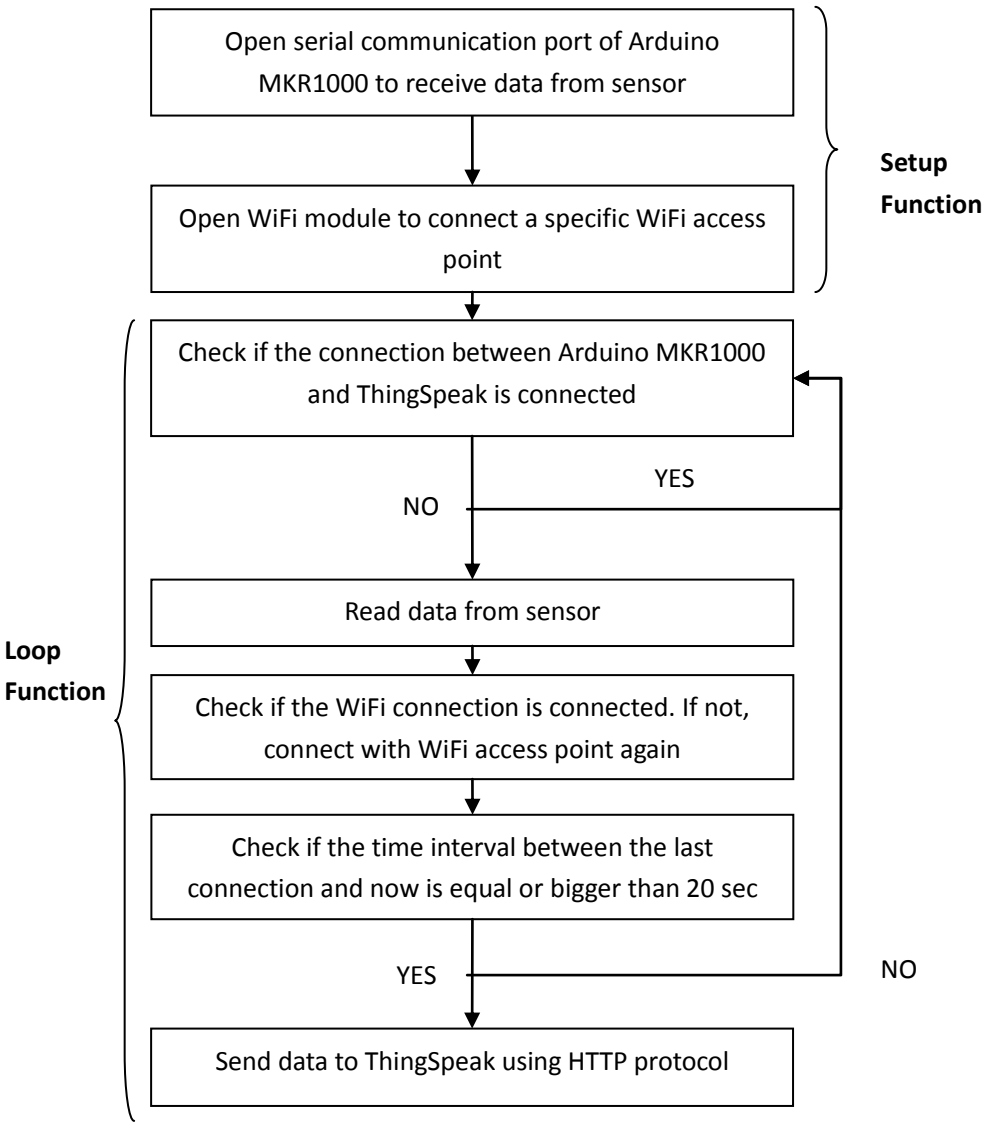


Figure 2-20: Working Flow of Arduino MKR1000 in mode 2

In setup function part, Arduino MKR1000 opens serial communication port to receive data from PM2.5 laser dust sensor firstly. And then it connects with the specific WiFi access point. The access point name and password is set like in picture 2-21

```
char ssid[] = "Vodafone-33673565"; // wifi access point name
char pass[] = "realegest";     // your network password
```

Picture 2-21: The Way to set WiFi access Point Name and Password

Chapter 3 - Evaluation of System

This section introduces practical experiments using this system to evaluate the system performance in both working modes. 4 places are chosen as experiment places in Milan city, 3 places for working with smart phone and 1 place for working with ThingSpeak. For working with smart phone, those 3 places are monitored for 20 minutes each, and sampling period is about 2 seconds. For working with ThingSpeak, the experiment place is monitored both indoor and outdoor. And both last for 48 hours.

Experiments' output is PM1.0, PM2.5, PM10 value, time points when these data are captured, and location data. Matlab is able to analyze these data and get several figures to check status of air quality in different situations. This chapter shows all output figures and analyzes the changing of air quality in different places.

In order to get precise result, the status of the experiment sensor is checked. The way is: choose another new sensor, and let both sensors work for same period of time in same place. Then check if the result is same. If those 2 sensor output same result, that means the sensor used in experiment work well. Otherwise the sensor cannot be used. Before every monitoring experiment, make sure the sensor warm up for 3 minutes and then begin monitoring.

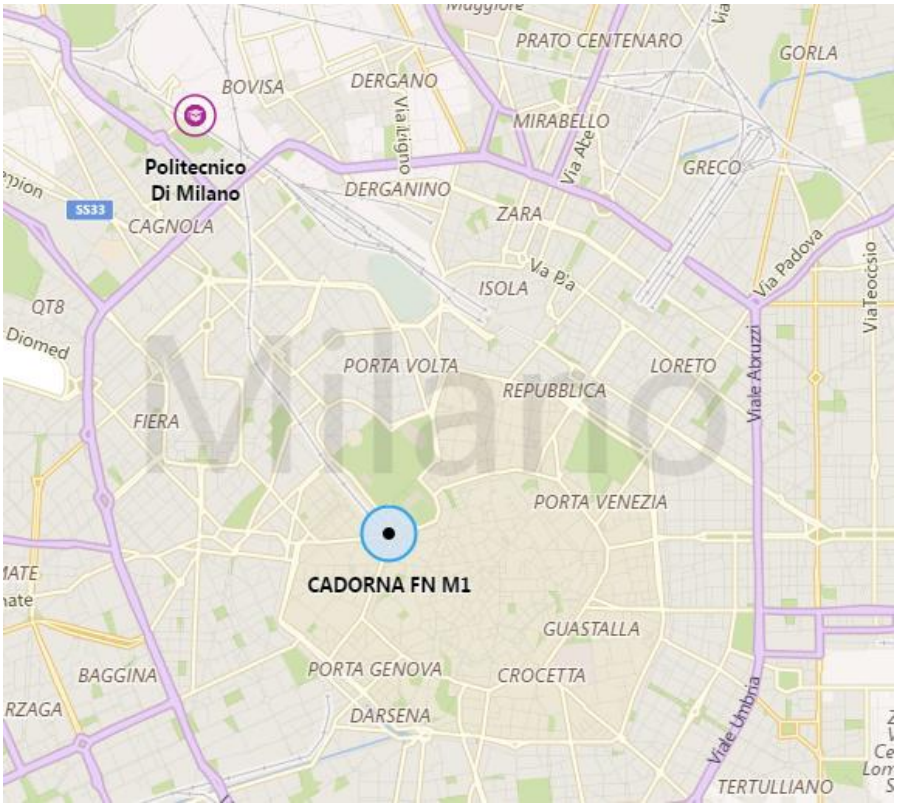
This chapter is divided into two parts. The first part is evaluation of working with Android smart phone, the second part is working with ThingSpeak.

Also in this chapter, the unit for all PM1.0, PM2.5 and PM10 data is $\mu g/m^3$

● Working With Android Smart Phone

Working with Android smart phone is portable. Users can take it to any place to capture particulate matter.

Users need to think about what kind of places are valuable to monitor. In this experiment, 3 places are chosen: a road aside Cadorna subway station in city center of Milan, inside of Cadorna subway station, and Politecnico di Milano Bovisa Campus in suburban of Milan. Choosing these places is intended to research and compare air quality in those different places of Milan.



Picture 3-1: Position of Places

The positions are showed in picture 3-1. Those two positions are about 4 kilometers away from each other.

The exact methods of detecting air quality are: For each place, detect particulate matter for 20 minutes (Warming up time of sensor does count) and calculate average PM1.0, PM2.5, PM10 value of that place. Monitoring of 3 places had better be done within short period of time like 2 or 3 hours in one day. Because in this way, environment factors such as temperature, humidity and wind can be considered as constant. It helps reduce error of this experiment.

This experiment was done on Saturday 12/11/2016. The first place is the inside of Cadorna Subway Station of Milan, this is a subway station locates in the center of Milan. The second place is a road with huge traffic aside Cadorna Subway Station. The third place is Politecnico di Milano Bovisa Campus in suburban of Milan. Since that day was Saturday, there were only a few people and there is no car in that campus. Those 3 places were detected within 3 hours. Within 20 minutes, each place got about 800 samples. The first 750 samples of each place are taken into consideration.

Monitoring In the Inside of Cadorna Subway Station

Picture 3-2: Screenshot of Monitoring inside of Cadorna Subway Station

From picture 3-2, we can see the exact beginning time and location data.
Result of monitoring in the inside Cadorna Subway Station is showed in figure 3-3:

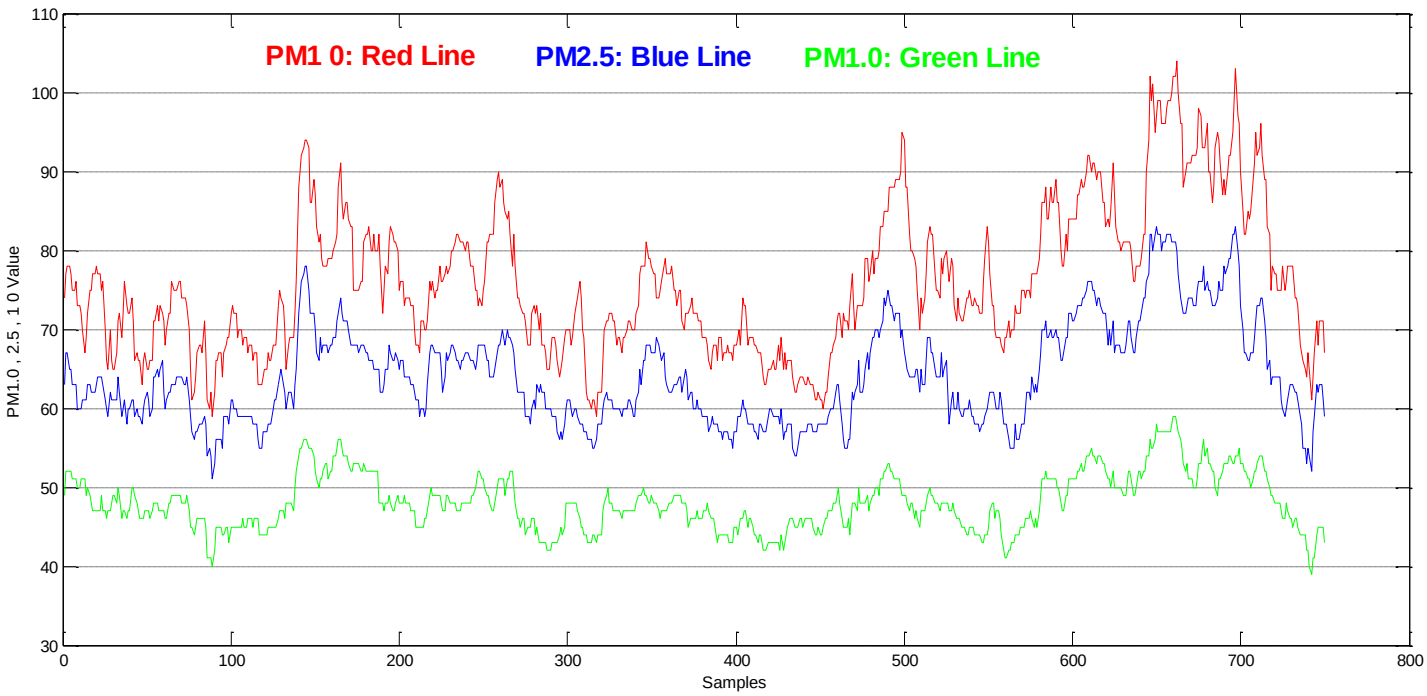


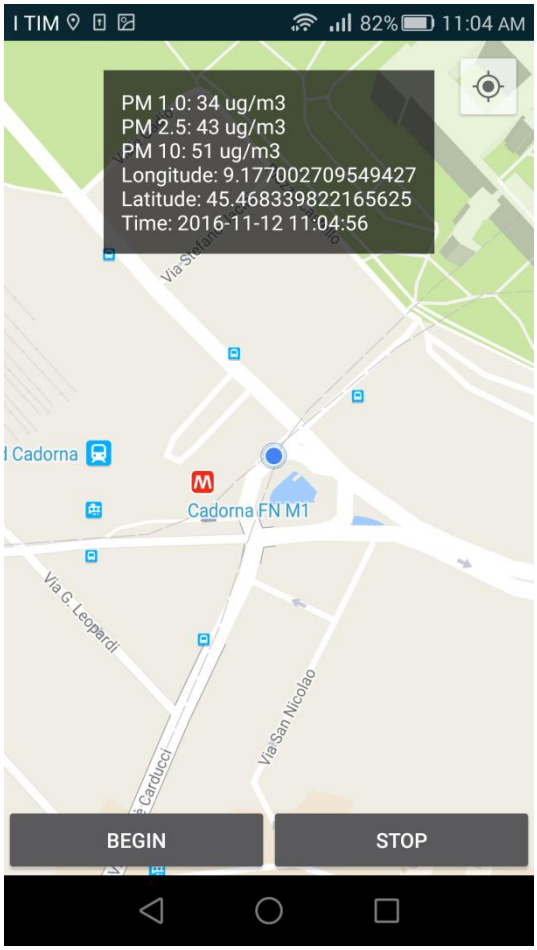
Figure 3-3: Result of Monitoring inside of Cadorna Subway Station

Mean value of PM1.0 in the inside Cadorna Subway Station: 48.10. The maximum value is 59. The minimum value is 39.

Mean value of PM2.5 in the inside Cadorna Subway Station: 64.14. The maximum value is 83. The minimum value is 51.

Mean value of PM10 in the inside Cadorna Subway Station: 75.96. The maximum value is 104. The minimum value is 59.

Monitoring on A Road with Huge Traffic Aside Cadorna Subway Station



Picture 3-4: Screenshot of Monitoring aside a road of Cadorna Subway Station

From picture 3-4, we can see the exact beginning time and location data.
Result of monitoring on a road with huge aside Cadorna Subway Station is showed in figure 3-5:

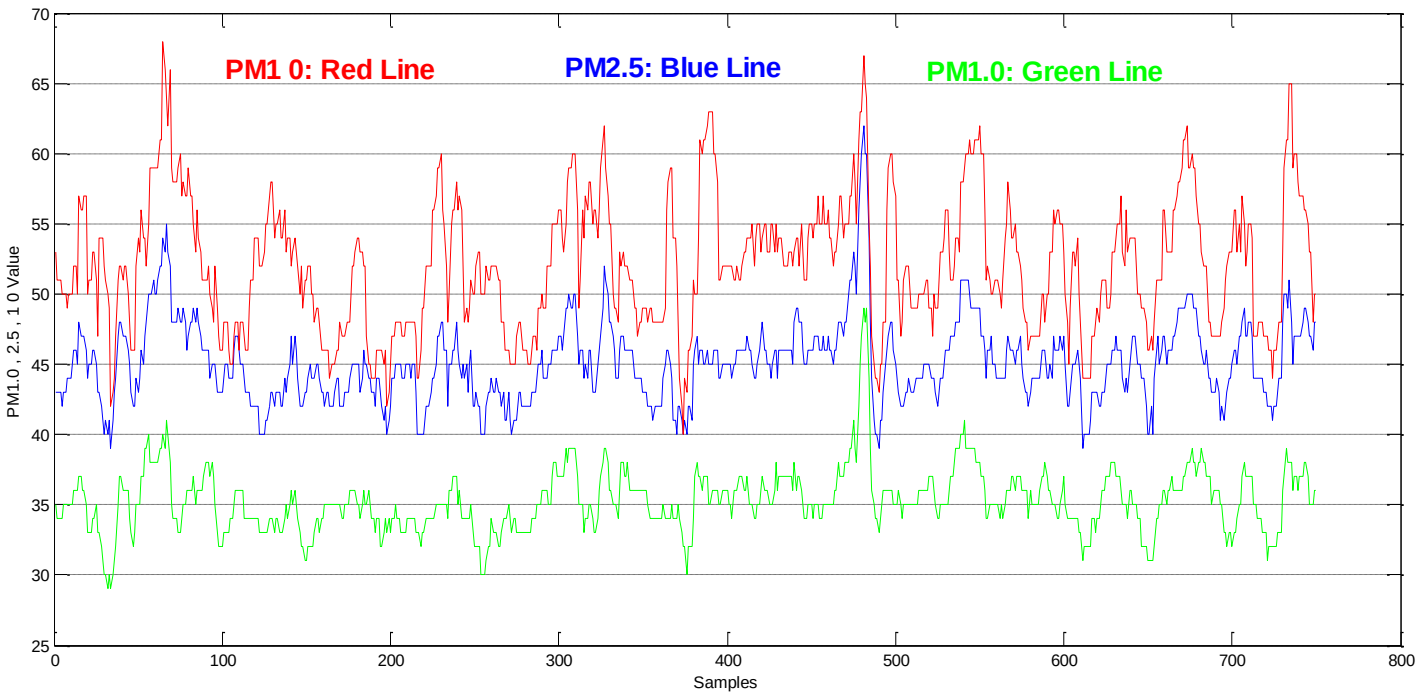


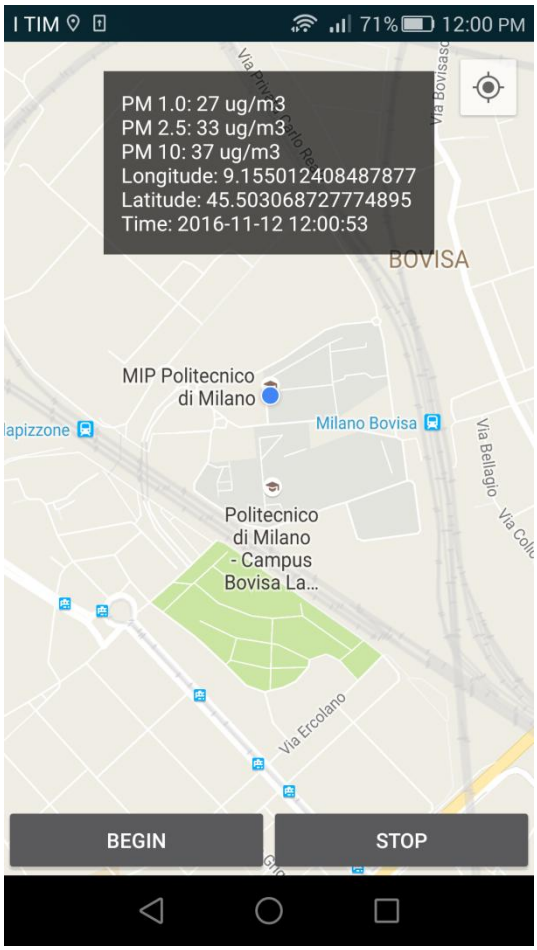
Figure 3-5: Result of monitoring on a road with huge traffic aside Cadorna Subway Station

Mean value of PM1.0 on a road with huge traffic aside Cadorna Subway Station: 35.24. The maximum value is 49. The minimum value is 29.

Mean value of PM2.5 on a road with huge traffic aside Cadorna Subway Station: 45.08. The maximum value is 62. The minimum value is 39.

Mean value of PM10 on a road with huge traffic aside Cadorna Subway Station: 52.09. The maximum value is 68. The minimum value is 40.

Monitoring in Politecnico Di Milano Bovisa Campus in Suburban of Milan



Picture 3-6: Screenshot of Monitoring in Politecnico di Milano Bovisa Campus

From picture 3-6, the exact beginning time and location data can be seen

Result of monitoring Politecnico di Milano Bovisa Campus in suburban of Milan is showed in figure 3-7

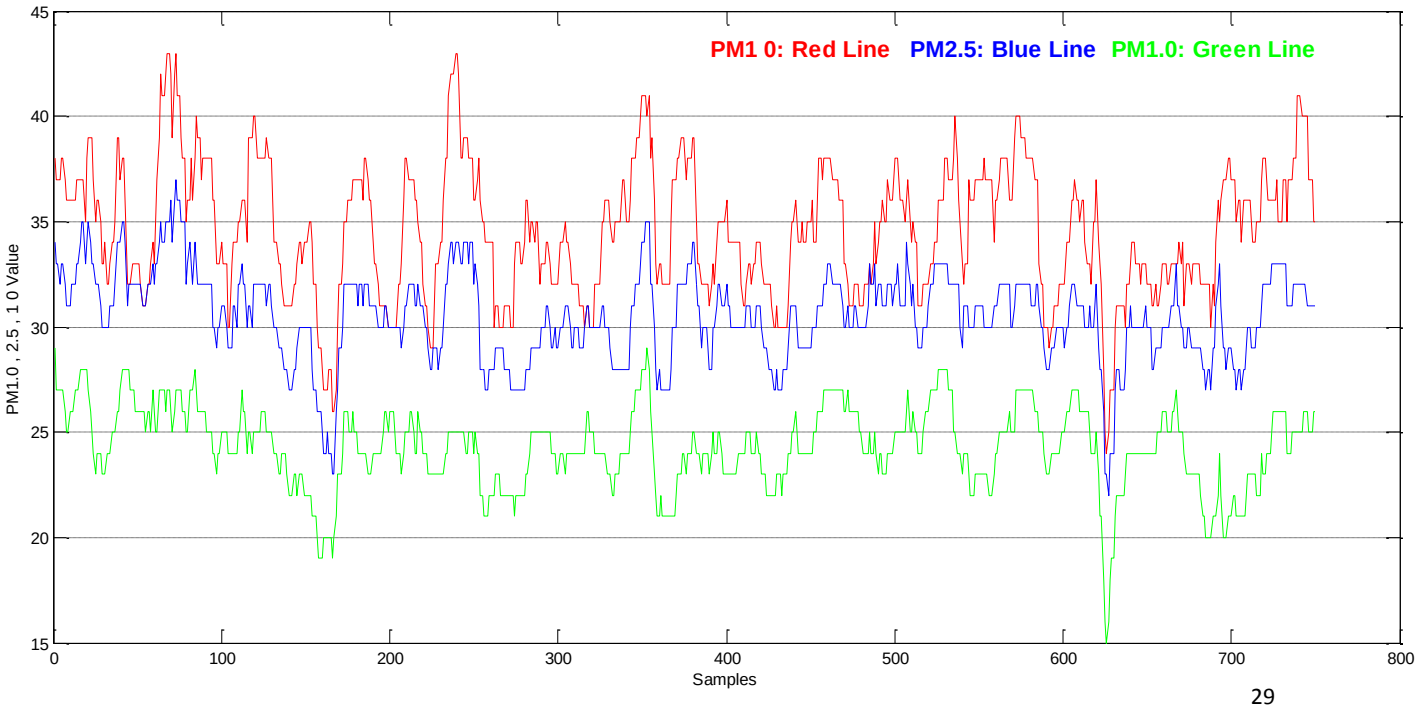


Figure 3-7: Result of monitoring in Politecnico di Milano Bovisa Campus in suburban of Milan

Mean value of PM1.0 in suburban of Milan: 24.26. The maximum value is 29. The minimum value is 15.

Mean value of PM2.5 in suburban of Milan: 30.48. The maximum value is 37. The minimum value is 22.

Mean value of PM10 in suburban of Milan: 34.60. The maximum value is 43. The minimum value is 24.

Compare Data in Different Places

Compare PM1.0 in 3 different places (figure 3-8):

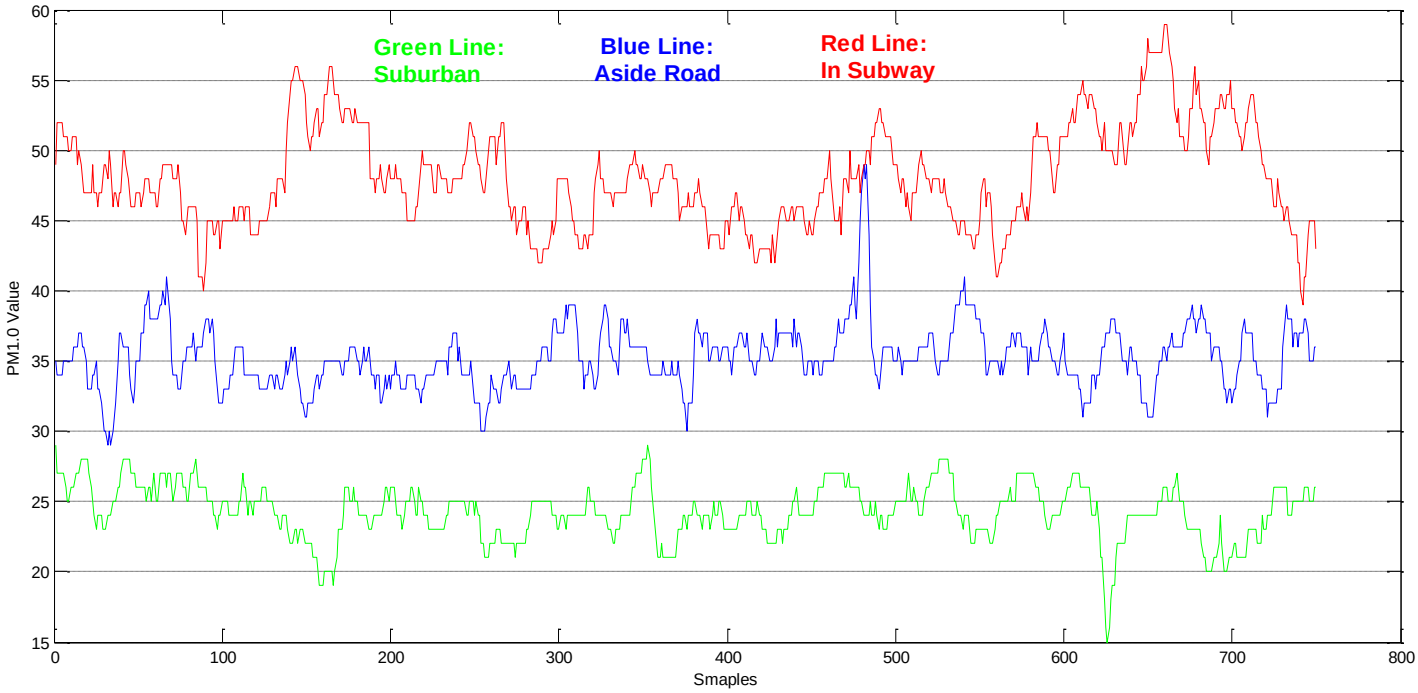


Figure 3-8: Comparison of PM1.0 in 3 Places

Result is : In Subway ⩾ Aside Road in Milan City Center ⩾ Suburban of Milan for PM1.0

PM1.0 Value Comparison

	Average	Maximum	Minimum
Inside Subway	48.10	59	39
Aside Road	35.24	49	29
Suburban	24.26	29	15

Compare PM2.5 in 3 different places (3-9):

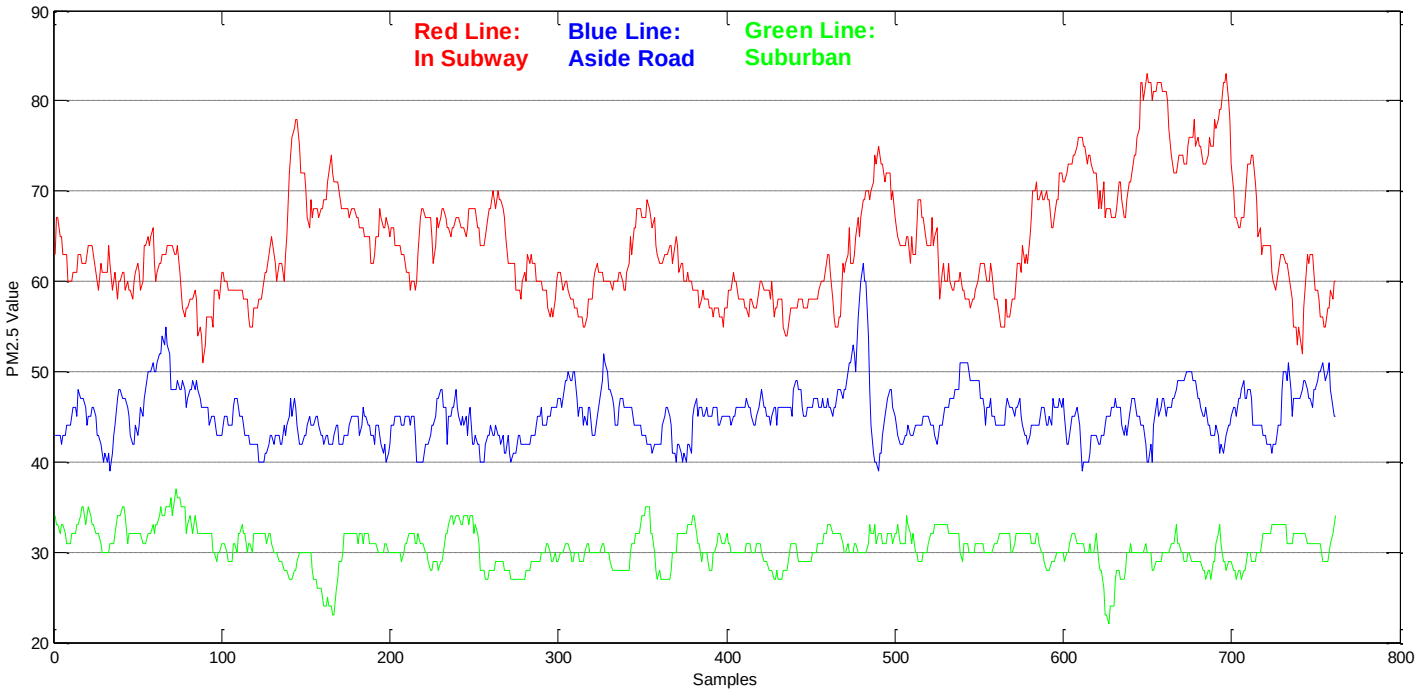


Figure 3-9: Comparison of PM2.5 in 3 Places

Result is : In Subway ⩾ Aside Road in Milan City Center ⩾ Suburban of Milan for PM2.5

PM2.5 Value Comparison

	Average	Maximum	Minimum
Inside Subway	64.14	83	51
Aside Road	45.08	62	39
Suburban	30.48	37	22

Compare PM10 in 3 different places (3-10)

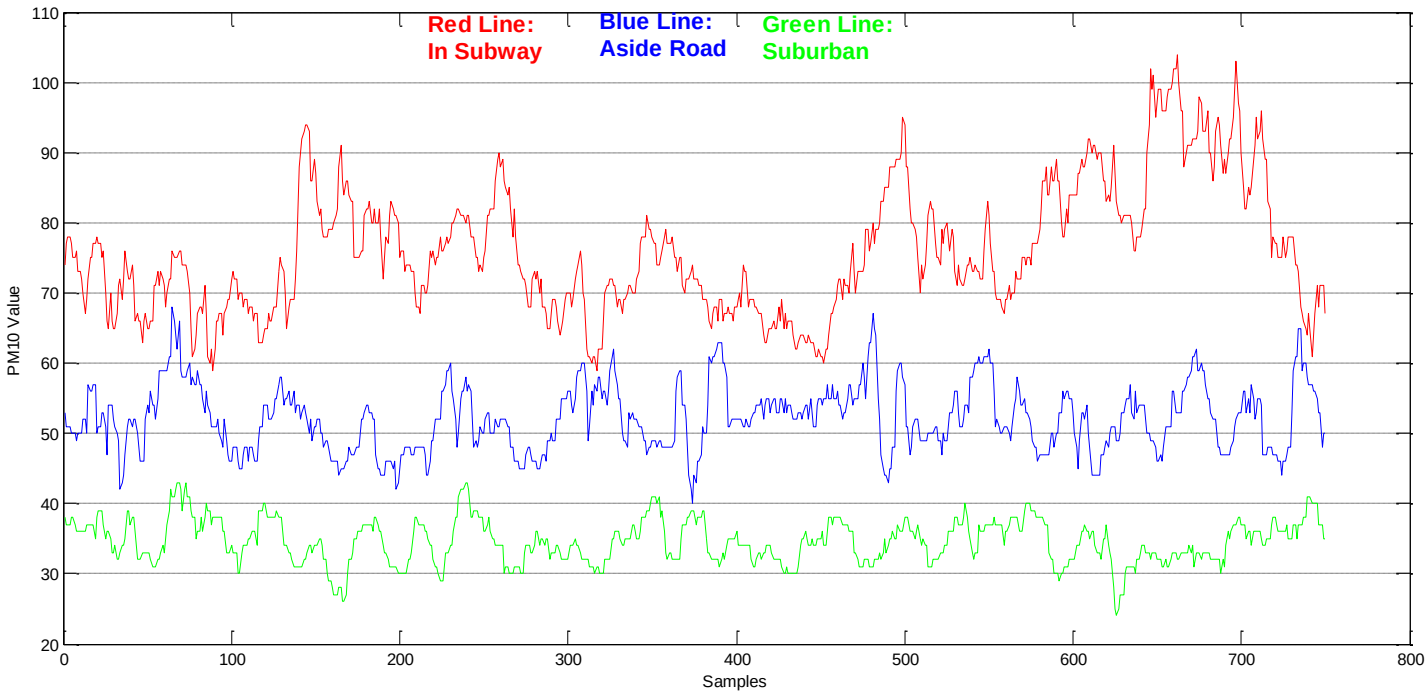


Figure 3-10: Comparison of PM10 in 3 Places

PM10 Value Comparison

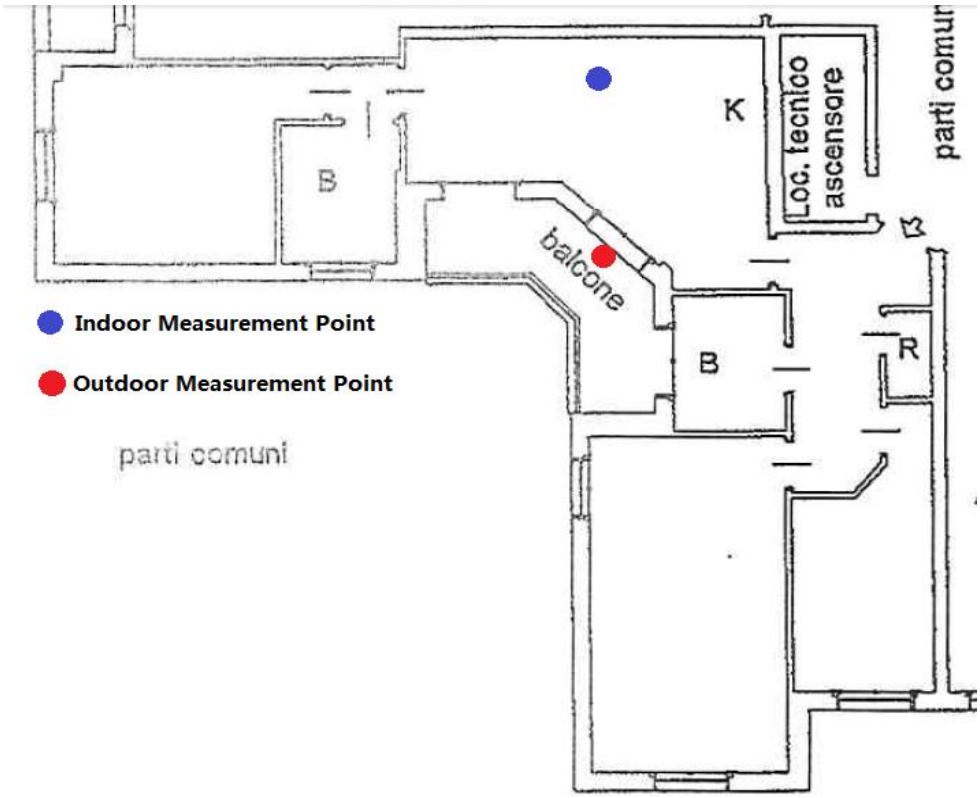
	Average	Maximum	Minimum
Inside Subway	75.96	104	59
Aside Road	52.09	68	40
Suburban	34.60	43	24

Result is : In Subway ⩾ Aside Road in Milan City Center ⩾ Suburban of Milan for PM10

Based on above comparisons, the following result can be seen: in those 3 places, air quality in suburban of Milan is the best, a road with huge traffic in city center is middle level, inside of Cadorna subway station is the worst according to PM1.0, PM2.5, PM10 data in those 3 places.

● Working with ThingSpeak

Monitoring was done in an apartment in Milan City. The apartment is in ninth floor. It has a balcony and 3 bedrooms.



Picture 3-11: Monitoring Positions

Like it shows in picture 3-11, blue point is the monitoring point in indoor environment. It was done in the kitchen. And red point is the monitoring point of outdoor environment. It was done on the balcony. Each measurement lasts 48 hours.

There is a WiFi access point in the kitchen. Arduino MKR1000 connects with this access point and uploads data to ThingSpeak both in indoor and outdoor environment monitoring.

Result of Indoor Monitoring

This monitoring started around 10:45:41 AM on 07/11/2016, and finished at 12:00:10 AM on 09/11/2016. There was no stop during this period. System sent data to ThingSpeak every 20 seconds. In the end, ThingSpeak received 8517 samples (One sample contains one PM1.0 value, one PM2.5 value, one PM10 value)
Put all samples into a figure based on time dimension:

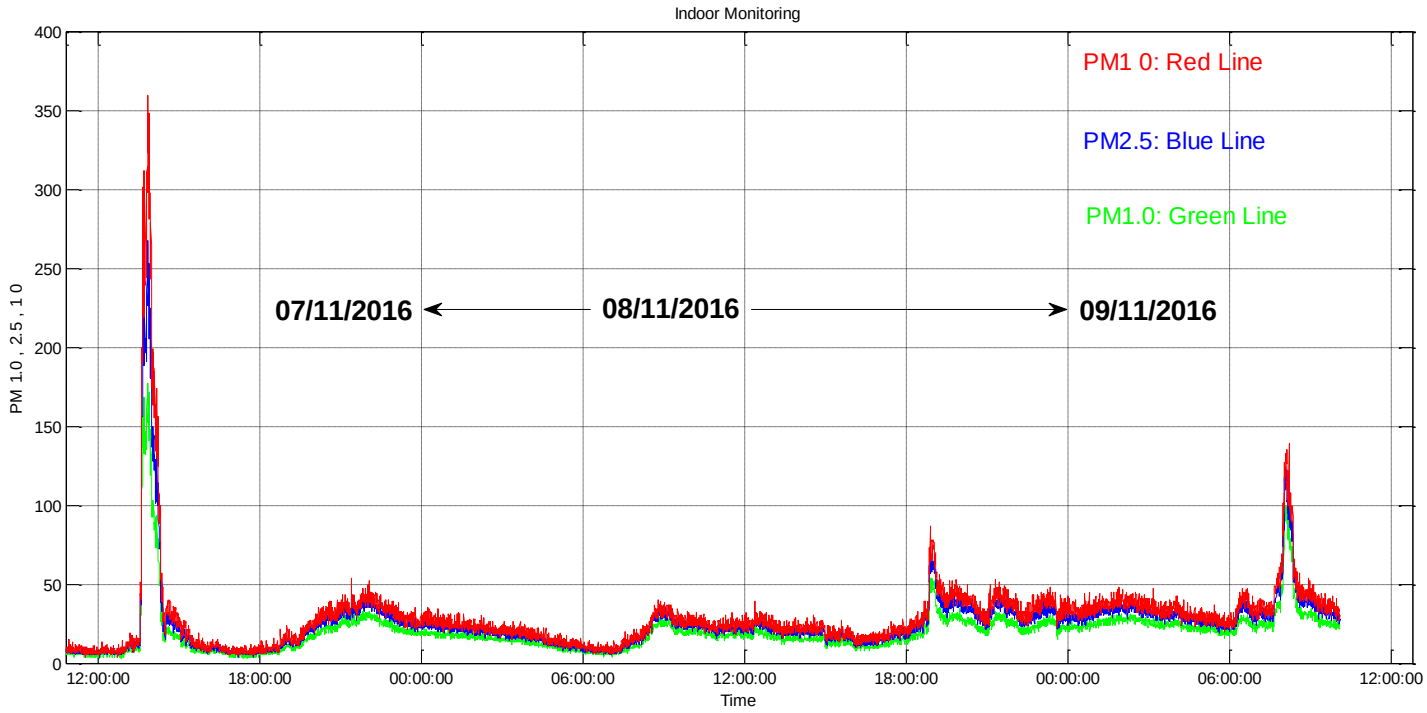


Figure 3-12: Indoor Monitoring Result of PM1.0, PM2.5, PM10

Figure 3-12 contains all types of data: PM1.0, PM2.5, PM10. And they have this relationship: PM1.0 $\leqslant$ PM2.5 $\leqslant$ PM10.

It is obvious that PM1.0, PM2.5, PM10 have same changing rule. They increase or decrease together. Take PM1.0 data out and show it in figure 3-13:

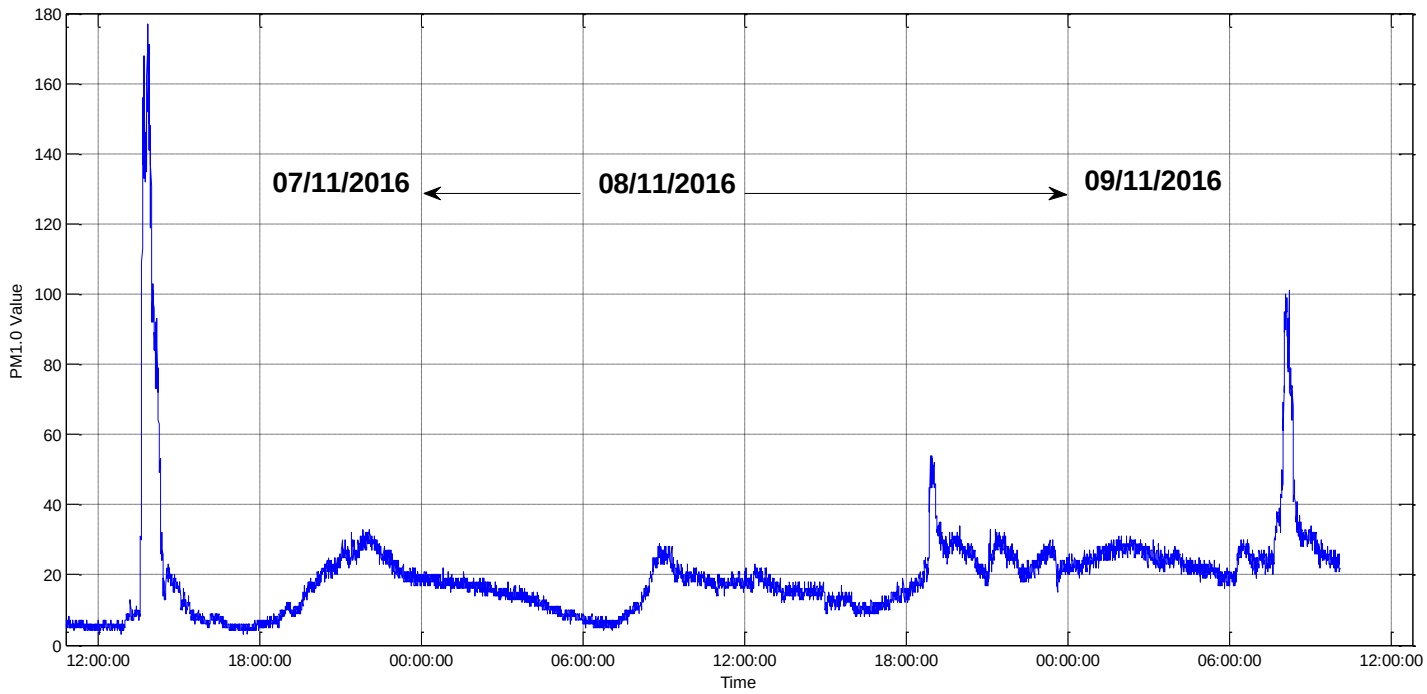


Figure 3-13 Indoor Result of PM 1.0

Mean value of PM1.0 is: 19.88, Maximum is: 177, Minimum is: 3.

Take PM2.5 data out and show it in figure 3-14:

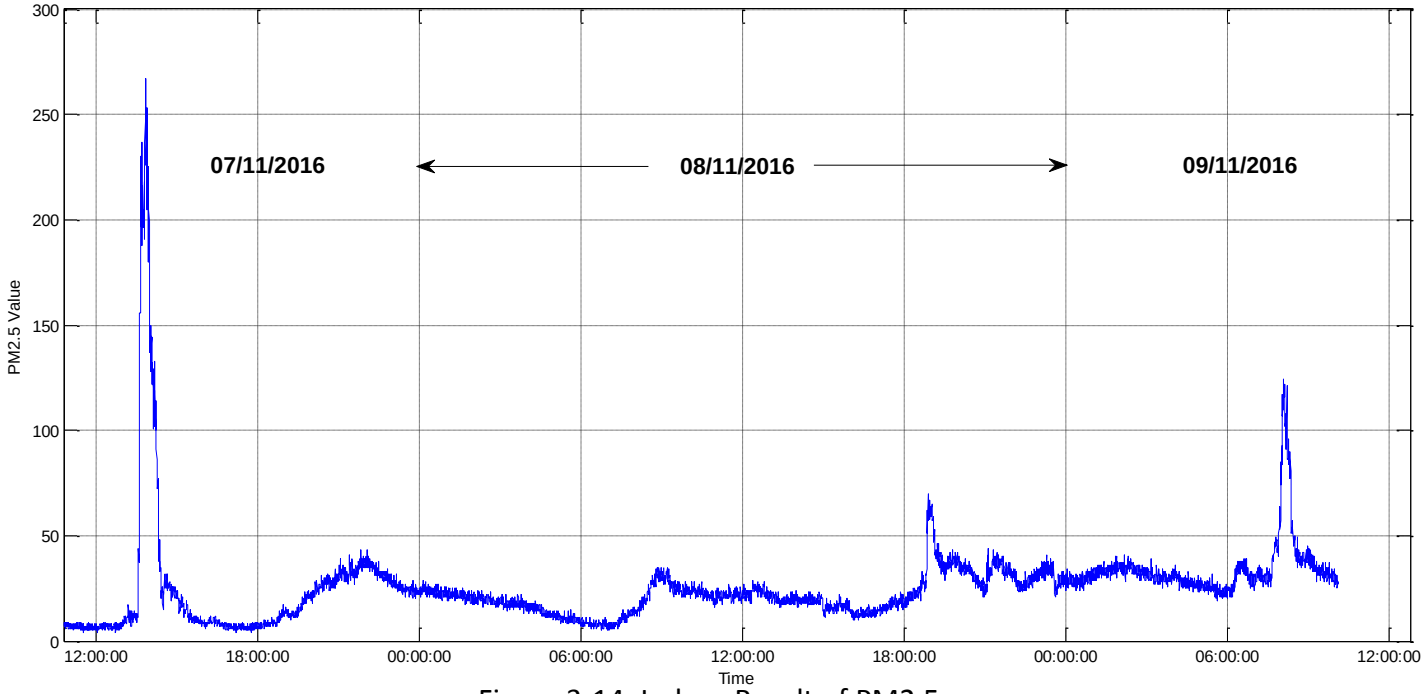


Figure 3-14: Indoor Result of PM2.5

Mean value of PM2.5 is: 25.28, Maximum is: 267, Minimum is: 4.

Take PM10 data out and show it in figure 3-15:

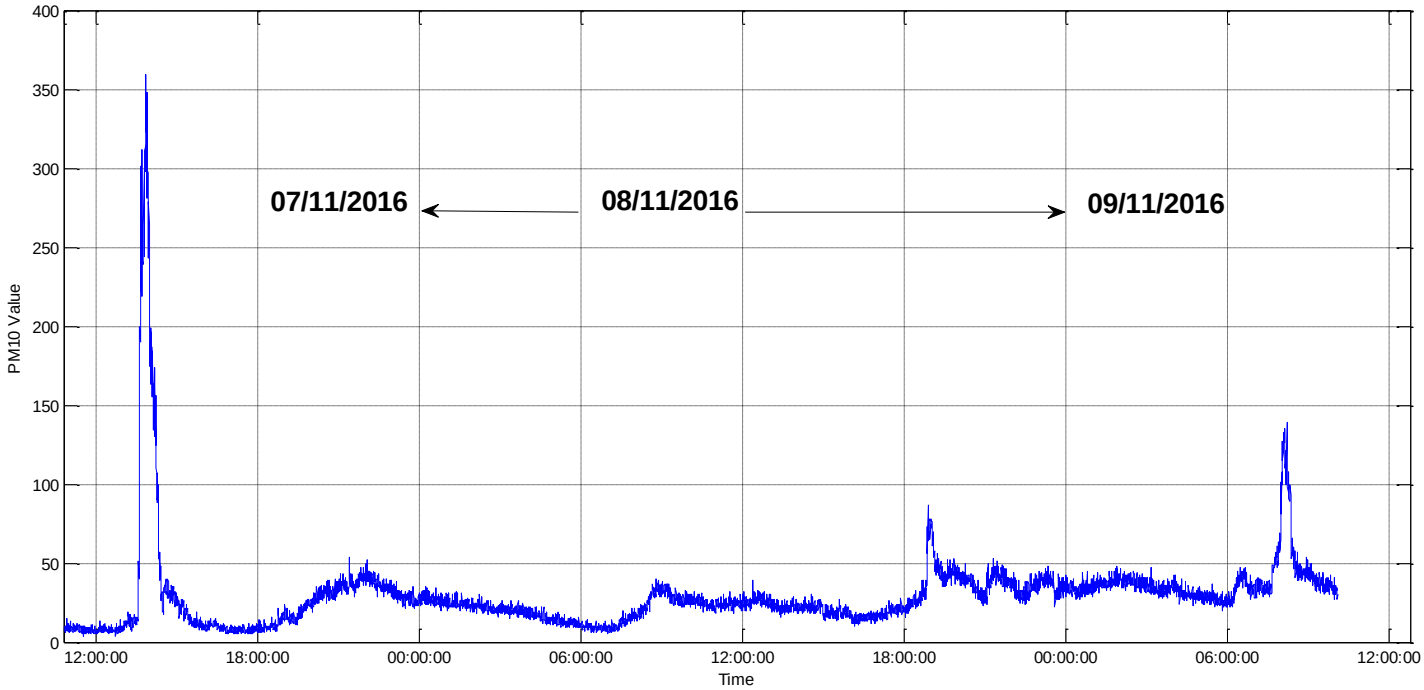


Figure 3-15: Indoor Result of PM10

Mean value of PM10 is: 29.13, Maximum is: 359, Minimum is: 4.

There are several peaks with high value in those figures: one is around 14:00 on 07/11/2016, one is around 19:00 on 08/11/2016, and one is around 09:00 on 09/11/2016. There are also 2 peaks with lower value around 21:00 on 07/11/2016 and 09:00 on 08/11/2016. It is obvious that these peaks all locate in time when we cook in the kitchen. Frying and baking may produce many particulates into air. It causes concentration of particulate of air increasing quickly. And also we can see that high particulate matter status stays about one hour based on this figure

Another observation is from 00:00 to 06:00, the air quality is stable. There is no quick increase or decrease of particulate matter. One possible reason is there is no activity of people in this period of time.

Result of Outdoor Monitoring

This monitoring started at 12:59:34 on 09/11/2016, and finished at 13:36:26 on 11/11/2016. There was no stop during this period. System sent data to ThingSpeak every 20 seconds. In the end, ThingSpeak received 8407 samples (One sample contains one PM1.0 value, one PM2.5 value, one PM10 value)

Put all samples into figure 3-16 based on time dimension:

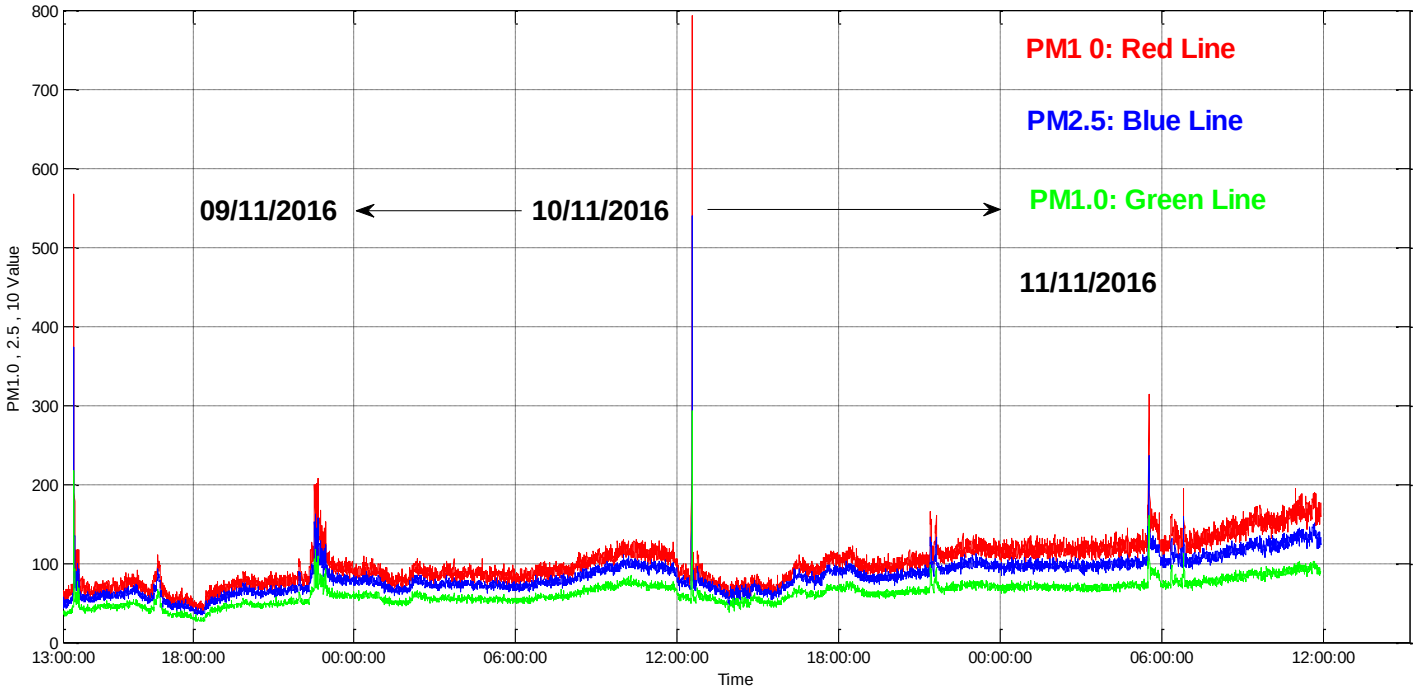


Figure 3-16: Outdoor result of PM1.0, PM2.5, PM10

Figure 3-16 contains all types of data: PM1.0, PM2.5, PM10. And they have this relationship: PM1.0 ≤ PM2.5 ≤ PM10.

It is obvious that PM1.0, PM2.5, PM10 have same changing rule, too. They increase or decrease together and have the relationship: PM1.0 ≤ PM2.5 ≤ PM10

There are a few very high value peaks within very short time, such as peak around 13:00 on 09/11/2016 and peak around 12:00 on 10/11/2016. One possible reason is system error, since particulate concentration cannot change extremely quick. So those data can be ignored in the following calculation of mean value, maximum and minimum value.

Take PM1.0 data out and show it in figure 3-17:

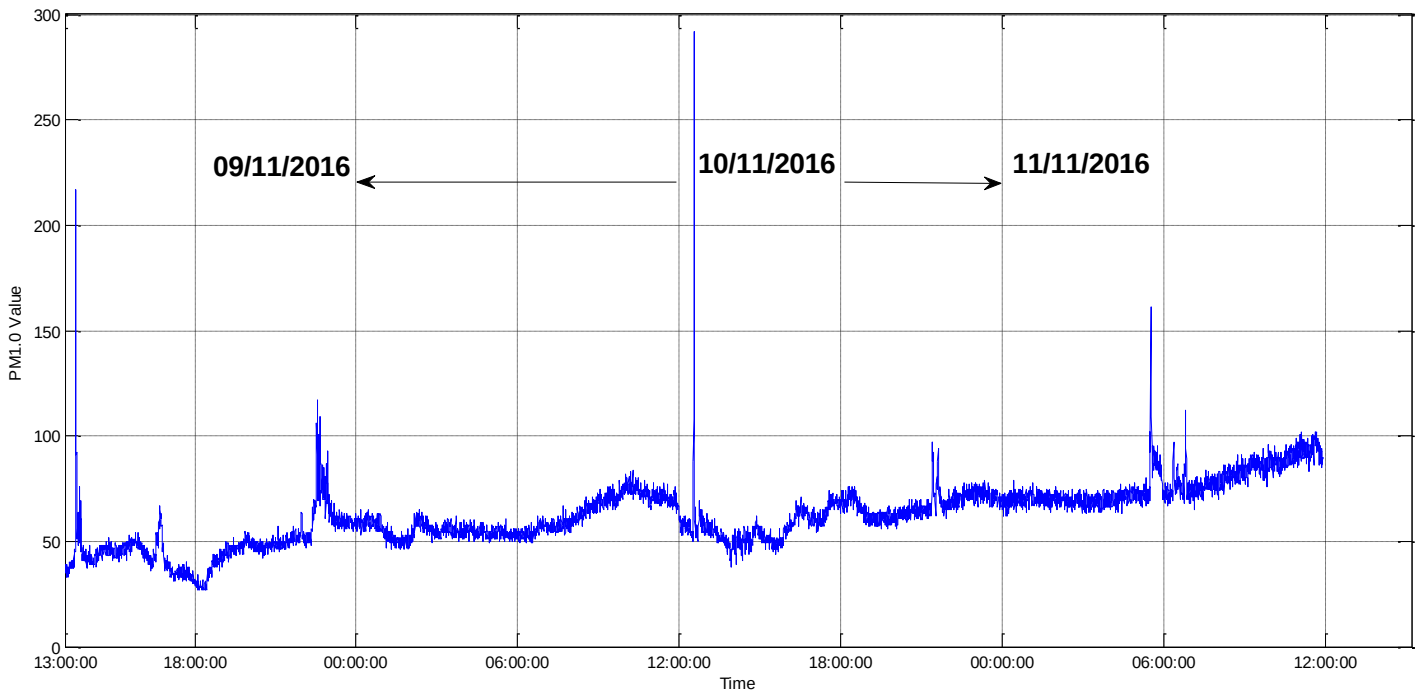


Figure 3-17: Outdoor Result of PM1.0

Mean value of PM1.0 is: 61.93, Maximum is: 117, Minimum is: 27.

Take PM2.5 data out and show it in figure 3-18

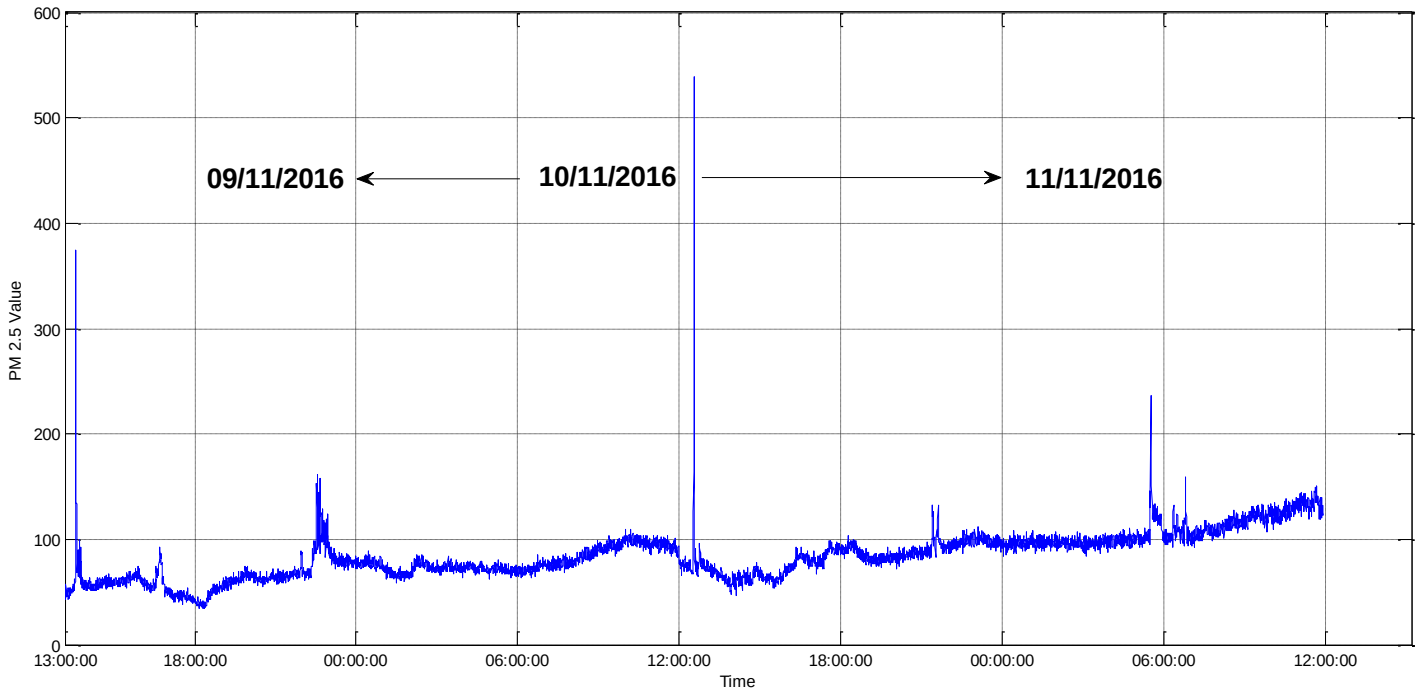


Figure 3-18: Outdoor Result of PM2.5

Mean value of PM2.5 is: 83.58, Maximum is: 162, Minimum is: 34

Take PM10 data out and show it in figure 3-19

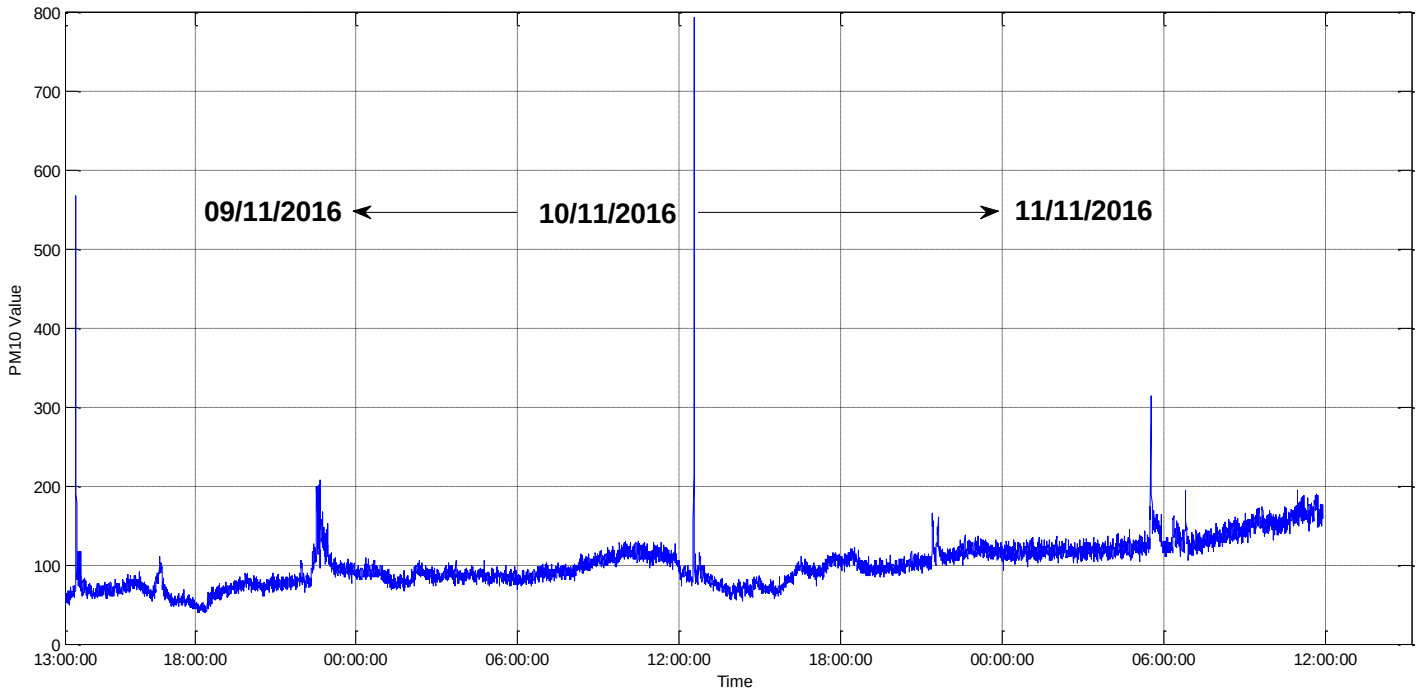


Figure 3-19: Outdoor Result of PM10

Mean value of PM10 is: 99.64, Maximum is: 208, Minimum is: 39.

Obviously particulate matter concentration increased from 09/11/2016 to 11/11/2016. One possible reason is Milan City began to warm heater in November. Because heating need burn coal and petroleum, it may increase particulate matter of air.

Compared with changing status of indoor air condition, changing of outdoor air condition is more stable. Because outdoor environment has much bigger space than indoor environment, hence air quality changing is slow. In indoor environment, activities of people have an obvious effect on air quality, because space of indoor environment is much smaller.

There is an important issue that needs to be mentioned. After all experiments, the data is compared with data from a website:
http://www2.arpalombardia.it/sites/qaria/_layouts/15/qaria/idati.aspx
This is a website where users can check PM2.5 and PM10 data of Milan. Their data is captured by devices deployed in some places of Milan. For data of long time ago, users can only check PM10, so PM10 is chosen as index.

First of all, the average value of PM10 on 10/11/2016 using the data from working mode 2 is (Delete error data of that data already): 94.31. And then compare those results with data from that website. The data from the website is 56. And the monitoring positions are showed in picture 3-20. Which one is more precise need to be considered and verified in future work.

Picture 3-20: Positions of Comparison Data

Chapter 4 - Conclusion and Future Application of This Prototype

● Conclusion

In this article, a system with 2 modes is developed to monitor outdoor and indoor environment air quality. Mode 1 is working with Android smart phone, and mode 2 is working with ThingSpeak platform. Both systems are tested and evaluated in real environment. Mode 1 is tested and evaluated in 3 places for short time. Mode 2 is tested and evaluated in 1 place for both indoor and outdoor environment and long time. From the testing result, difference of particulate matter in different places and particulate matter changing rules can be analyzed. Hence this prototype has availability and potential to develop more powerful air quality monitoring systems

This portable and low power consumption particulate monitoring system prototype can be used to develop into many more complicated systems. For working with Android smart phone mode, one considerable application is a monitoring system assembled on a bike; for working with ThingSpeak mode, one considerable application is add more sensors such as temperature sensor or humidity sensor to capture more kinds of data, and analyze the relation between them.

● Future Application for working with Android smart phone

Consider a monitoring system assembled on a bike. When users ride those bikes, monitoring system is able to capture real time particulate matter data, real time position data (longitude and latitude), and transmit these to users' smart phone. With enough data, a particulate matter map can be drawn. Analysis can be done based on time dimension or location dimension.

In order to capture enough data, this system can be implemented on public open bikes system which is open for citizens like BikeMi. This kind of public open bike system covers big enough area. Also it has enough bikes and users which are important to collect data.

To design and develop a system used on a bike using this prototype, there are several problems need to be considered. First is the sensor problem. Bikes in public open bikes system expose in outdoor environment for very long time. Sensors to be used

in this kind of system need to be tolerant of outdoor environment. In outdoor environment, there are several factors need to be considered: sun exposure, rain, dust, high humidity and very high or very low temperature. Sensors used on a bike must be able to tolerate all these problems. If sensors break down frequently, it may increase the error possibility of data, also it may increase the total cost for the bike system.

The second problem is the error of data. Suppose the sensor used can tolerate all outdoor environment problems. There is still one problem that has effect on the error possibility of data. That is bikes are moving object when users use them and all sensors have response time (The concept of response time is basically the delay between time of environment change and time of output data change) when particulate matter changes. In this way, there will be error in final data table. The error is one particulate matter data corresponding to one location (longitude and latitude data) may be not the exact particulate matter for that position, it is the value for another position away from that position. The distance between them is the error. In this prototype, there is a method to estimate this error.

The DFROBOT PM2.5 laser dust sensor has a response time which is less than or equal to 10 seconds. This means that if particulate matter environment changes, the sensor need maximum 10 seconds to capture the data in new environment. And during this period of time, the bike is moving for a distance where the data captured is not correct. It can be calculated in this way:

$$D = \mathrm{V} \times \mathrm{T}$$

Where D is maximum distance of error, V is average speed, and T is maximum response time.

For example, if users want the error less than 20 meters, which means

$$\mathrm{V} \times \mathrm{T} \leq 20$$

Where T = 10 seconds, it must be

$$\mathrm{V} \leq 2 \text{ meters per second}$$

Which means speed of the bike must be less than or equal to 2 meters per second

But the speed is not easy to control when users are riding a bike, there will be some time when users speed up over 2 meters per second (or other speed limit). So a good solution is to drop the data when users' speed is over limit.

Let T' is the sampling period, and Vm is the maximum speed limit. If users do not exceed the speed limit, it has:

$$Dm = \mathrm{T}' \times \mathrm{Vm}$$

Where Dm is the maximum distance between two adjacent samplings conditioned to the speed limit. Every time the system takes a sample, it calculates the distance between this sample and the previous one. If the distance between them is longer than Dm, that means users' speed is over the limit, and this sample is discarded. In this way, the error of data can be controlled within the limit that users set.

● Future Application for working with ThingSpeak

The prototype of working with ThingSpeak has a great potential to develop a more powerful system, especially for indoor air quality monitoring system. Because people spend 80% their time in indoor environment [14] and the system is able to work continually for very long time such as several months. And Arduino MKR1000 has many analog and digital input/output pins which can be used by other devices. For example, add one LM35 Precision Centigrade Temperature Sensor to Arduino MKR1000 based on this prototype. Deploy this new system in a room with WiFi network, and let it for 3 months.

Picture 4-1: Temperature Sensor LM35

LM35 Precision Centigrade Temperature Sensor has those important features [15]:
• Calibrated Directly in Celsius (Centigrade)
• Linear + 10-mV/°C Scale Factor
• 0.5°C Ensured Accuracy (at 25°C)
• Rated for Full −55°C to 150°C Range

This sensor outputs voltage to represent temperature in Centigrade, they have simple linear transfer relationship. And the transfer function is:

$$\mathrm{V_{out}} = 10\mathrm{mV/°C} \times \mathrm{T}$$

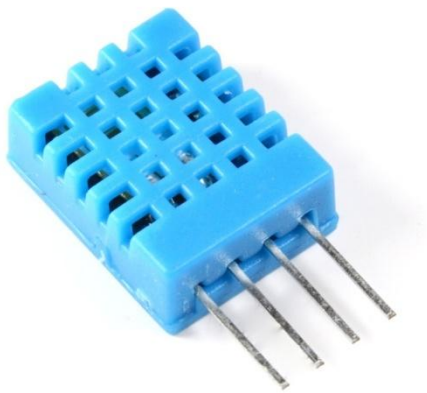
Where:
$\mathrm{V_{out}}$ is output voltage
T is the temperature in °C

This sensor has three pins, the left pin is VCC pin, and the right pin is GND pin. These two pins are responsible for power supply of this temperature sensor. Pin in the

middle is output pin. This pin can be connected with any analog input pins of Arduino MKR1000 to let Arduino read the output voltage. Using the relation equation between the output voltage and the temperature above, Arduino MKR1000 is able to calculate the temperature.

Let the period of sampling temperature be the same with period of reading particulate matter data from PM2.5 laser dust sensor. When system reads a particulate matter data, it also reads a temperature data. In the end users can get same amount temperature data. Users can analyze particulate matter data on both time dimension and temperature dimension. Or users can analyze the relationship between particulate matter and temperature.

If users want more data, humidity factor of air can be considered. Users can add a humidity sensor based on the system above or users can choose a better sensor like DHT11 instead of LM35. DHT11 is able to detect both temperature and humidity [16].



Picture 4-2: Humidity and Temperature Sensor DHT11

In this way, the system detects both temperature and humidity when it monitors particulate matter. In the end, users can get particulate data, temperature data, and humidity data. Users can analyze the relationship among them.

Reference

[1] C. Peng, K. Qian, C. Wang, "Design and Application of a VOC-Monitoring System Based on a ZigBee Wireless Sensor Network", IEEE Sensors Journal , Vol. 15, no. 4, April 2015

[2] P. Spachos; D. Hatzinakos , "Real-Time Indoor Carbon Dioxide Monitoring Through Cognitive Wireless Sensor Networks", IEEE Sensors Journal, pp. 506-514, vol. 16, no. 2, Jan.15, 2016

[3] D. Brunelli, I. Minakov, R. Passerone, M. Rossi, "POVOMON: an Ad-hoc Wireless Sensor Network for Indoor Environmental Monitoring", 2014 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems Proceedings, pp. 1-6, Sept. 2014

[4] S. M. Saad, A. R. M. Saad, A. M. Y. Kamarudin, A. Zakaria, A. Y. M. Shakaff, " Indoor Air Quality Monitoring System Using Wireless Sensor Network (WSN) with Web Interface ", Electrical, Electronics and System Engineering (ICEESE), 2013 International Conference on, pp. 60-64, Dec. 2013

[5] I. Peng, Y. Chu, C. Kong, Y. Su, "Implementation of Indoor VOC Air Pollution Monitoring System with Sensor Network", 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 639-643, July 2013

[6] D. M. G. Preethichandra, "Design of a Smart Indoor Air Quality Monitoring Wireless Sensor Network for Assisted Living", 2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1306-1310, May 2013

[7] S. Bhattacharya, S. Sridevi; R. Pitchiah, "Indoor Air Quality Monitoring Using Wireless Sensor Network", 2012 Sixth International Conference on Sensing Technology (ICST), pp. 422-427, Dec. 2012

[8] H. C. Lee , "Towards a General Wireless Sensor Network Platform for Outdoor Environment Monitoring", SENSORS, 2012 IEEE, pp. 1-5, Oct. 2012

[9] A. Priyadarshini, N. Dehury, A. K. Samantaray, "A real time portable embedded system design for particulate matter monitoring ", 2015 IEEE Bombay Section Symposium (IBSS), pp. 1-5, Sept. 2015

[10] Q. Yang, G. Zhou, W. Qin, B. Zhang, P. Chiang, "Air-Kare: A Wi-Fi Based, Multi-Sensor, Real-Time Indoor Air Quality Monitor", Wireless Symposium (IWS), 2015 IEEE International, March 30 2015-April 1 2015, On pp. 1 - 4,

[11]https://www.dfrobot.com/wiki/index.php/PM2.5_laser_dust_sensor_SKU:SEN0177

[12] https://www.arduino.cc/en/Main/ArduinoMKR1000

[13] https://thingspeak.com/

[14] S. N. Kamaruzzaman, et al., “The Effect of Indoor Air Quality (IAQ) Towards Occupants’ Psychological Performance in Office Buildings,” Journal of Design and the Built Environment, vol. 4, no. 2001, pp. 49–61, 2011.

[15] http://www.ti.com/lit/ds/symlink/lm35.pdf

[16] http://www.micropik.com/PDF/dht11.pdf