Smartphone Application for m-health and environmental monitoring systems

by

Srisivapriya Ganesan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2012 by the
Graduate Supervisory Committee:

Nongjian Tao, Chair
Yanchao Zhang
Tsing Tsow

ARIZONA STATE UNIVERSITY

May 2012

ABSTRACT

Windows based mobile application for m-health and environmental monitoring sensor devices were developed and tested. With the number of smartphone users exponentially increasing, the applications developed for m-health and environmental monitoring devices are easy to reach the general public, if the applications are simple, user-friendly and personalized. The sensing device uses Bluetooth to communicate with the smartphone, providing mobility to the user. Since the device is small and hand-held, the user can put his smartphone in his pocket, connected to the device in his hand and can move anywhere with it. The data processing performed in the applications is verified against standard off the shelf software, the results of the tests are discussed in this document. The user-interface is very simple and doesn't require many inputs from the user other than during the initial setting when they have to enter their personal information for the records. The m-health application can be used by doctors as well as by patients. The response of the application is very quick and hence the patients need not wait for a long time to see the results. The environmental monitoring device has a real-time plot displayed on the screen of the smartphone showing concentrations of total volatile organic compounds and airborne particle count in the environment at the location of the device. The programming was done with Microsoft Visual Studio and was written on VB.NET platform. On the applications, the smartphone receives data as raw binary bytes from the device via Bluetooth and this data is processed to obtain the final result. The final result is the concentration of Nitric Oxide in ppb in the Asthma Analyzer device. In the

environmental monitoring device, the final result is the concentration of total Volatile Organic Compounds and the count of airborne Particles.

DEDICATION

To my parents, for their love, encouragement and endless support.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Nongjian Tao for his support and guidance. He has been a great source of inspiration to me and has helped me achieve great success as a graduate student. I take this opportunity to thank him personally and also appreciate the opportunities he provided me.

I would like to thank Prof. Francis Tsow, with his experience and knowledge in the field of Electrical engineering. He has always been there whenever I needed his help and guidance.

I would like to thank Prof. Yanchao Zhang, I am very grateful to have him as a member in my committee.

I would like to thank Dr. Xiaojun Xian, his knowledge in the field of Chemistry has helped me understand the projects better.

I would like to thank my peers who have incessantly given me support and worked hard with me. And I would like to thank all my friends and especially my boyfriend for giving me great support and moral strength.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

The evolution of wireless technology and capabilities of personal mobile devices are gradually changing the way healthcare is delivered to patients. The growth and acceptance of mobile information technology at the point of care creates opportunities for enhanced prevention, patient care and safety. It can provide ease of access to data of each patient and provide better understanding of the current health status of the patient under free living conditions.

1.2 Health Monitoring

The term 'm-Health' was coined by Professor Robert Istepanian as "use of emerging mobile communications and network technologies for healthcare [1]." m-Health and e-Health are very closely related fields. E-Health is characterized as technology that supports the functions and delivery of healthcare. On the other hand, m-health deals with providing tools for physicians, for researchers, which strictly speaking are not the same as the conventional understanding of providing healthcare access. In addition, cost savings, time and other resources saving are also important component of m-health. It has been projected that by the year 2014, public and private healthcare providers may save between $1.96 billion and $5.83 billion in healthcare costs by utilizing m-Health technologies for health monitoring worldwide [2]. M-Health, being a technology which relies on extensive usage of mobile technology, the reach of healthcare services is widespread among areas, people and healthcare practitioners who previously had very limited exposure to healthcare.

1.3 Environmental monitoring

Many environmental problems and habitat-monitoring tasks require near real-time field mapping and precise positional information. These tasks are time consuming and difficult to achieve using traditional field mapping and geographic information system techniques, which are often further limited to desktop computers and hard-wired network communications [3]. This is where hand-held environmental devices wireless connected to smartphones come to the rescue. Field workers can easily carry the hand-held devices to field locations for collecting the data. With wireless communication capability, users can perform real-time data transfers to a smartphone which performs the data processing as well as acts as the display unit to plot the real-time data.

1.4 Smartphone applications

"A smartphone is a mobile phone offering advanced capabilities, often with PC-like functionality [4]". A smartphone runs complete OS and even the processing power of the smartphone is as good as a personal computer some years back at 1GHz range. With some leading companies like Nvidia and Intel coming up with Quad core mobile processors [5, 6], the potentials of smartphones are sky-high. There is no question about the user-interface of the smartphones with displays 960-by-640-pixel resolution at 326 ppi and multi-touch screens. The acceptance of mobile technology by the healthcare practitioners and the capabilities of the smartphones pave way for the development of m-health and mobile environmental devices.

1.5 Summary of the following chapters

The next chapter gives an overview of a general system design and flow diagram of each system in detail. Chapter 3 is about Mobile application design and development. The various topics covered in chapter 3 are Mobile Application Structure, User-Interface, Communication and Data processing. Chapter 4 mainly discusses about testing the application. Various types of testing methods are discussed and the results obtained are shown. Chapter 5 gives a summary of the whole report and concludes the document. Chapter 6 is about possible future work that can be done to further enhance this project, cloud computing concepts are explained briefly.

Chapter 2

SYSTEM DESIGN

Fig.1. System Block Diagram

Our systems basically consist of sample collection, sample delivery and conditioning block, sensing element block, detection circuitry block, Bluetooth block and the smartphone. The smartphone is where the raw data is received, processed into meaningful data, and displayed to the user.

This report consists of the summary of smartphone applications for the support of 3 hand-held systems – pressure based asthma analyzer device, acoustic based asthma analyzer device and particle detector device.

2.1 Asthma Analyzer device

This is a hand-held exhaled Nitric Oxide (eNO) sensor for asthma diagnosis. Asthma is the most common chronic condition in children, which has an alarming death rate of up to 80% in people under age of 19, according to the study of Asthma and Allergy Foundation of America [7]. Exhaled nitric oxide (eNO) is a gaseous molecule produced by certain cell types in an inflammatory response [8]. The fraction of eNO is a promising biomarker for the diagnosis, follow-up, and also acts as a guide to therapy in children and adults. The level of eNO in the breath of a patient suffering from Asthma can be a portent of worsening of asthma.

The sensing element in this system is a chemical coated sensor chip, which changes color when exposed to Nitrogen Dioxide. The sensing area of the sensing element is exposed to the breath sample whereas the reference area of the sensing element is not exposed to the breath sample. Hence there is nitrogen dioxide related color change only in the sensing area and the reference area remains almost the same color. The color change is detected by two photodiodes with an LED source. The reference area photodiode voltage is almost flat since it is not exposed to the sample and hence little to no color change.

There is a feedback loop in this system, which classifies the system into two types of devices – pressure based and acoustic based. The feedback system in the device is used to regulate the flow rate at which the sample is collected by the system within a range of values. This is very important because, the flow rate at which the sample is collected by system affects the NO level in human breath [9]. It is found that the ideal flow rate is 3000 ml/min with a 10% error window, which leaves us with a value in the range of 2700 to 3300 ml/min [9].

Pressure-based system

The device, here, is equipped with a pressure sensor near the sample collection block. The pressure sensor readings are calibrated to the flow rate and hence when the value falls within our window, a feedback is provided indicating that the flow rate is good and the valve connected to the pump is switched to the sampling channel. When the value of flow rate is below the desired range, a feedback is provided indicating the flow rate has to be increased, and when the value of flow rate is above the desired range, a feedback is provided indicating the flow rate has to be decreased. In these two cases, the valve is not switched to the sampling channel. This feedback loop is performed autonomously by the analyzer and does not involve the use of the smartphone application.

Acoustic-based system

In this system, the pressure sensor is replaced with an acoustic sensor. The acoustic sensor generates an audio signal whose frequency is calibrated to the flow rate at which the sample is delivered. The application is required to listen to the acoustic tone generated by the sensor and send the feedback to the device based on where the value lies within, below or above the desired window. The feedback loop here involves the use of smartphone application as well. The smartphone used in this device is a HTC HD2 phone running on Windows 6.5.3 Operating System.

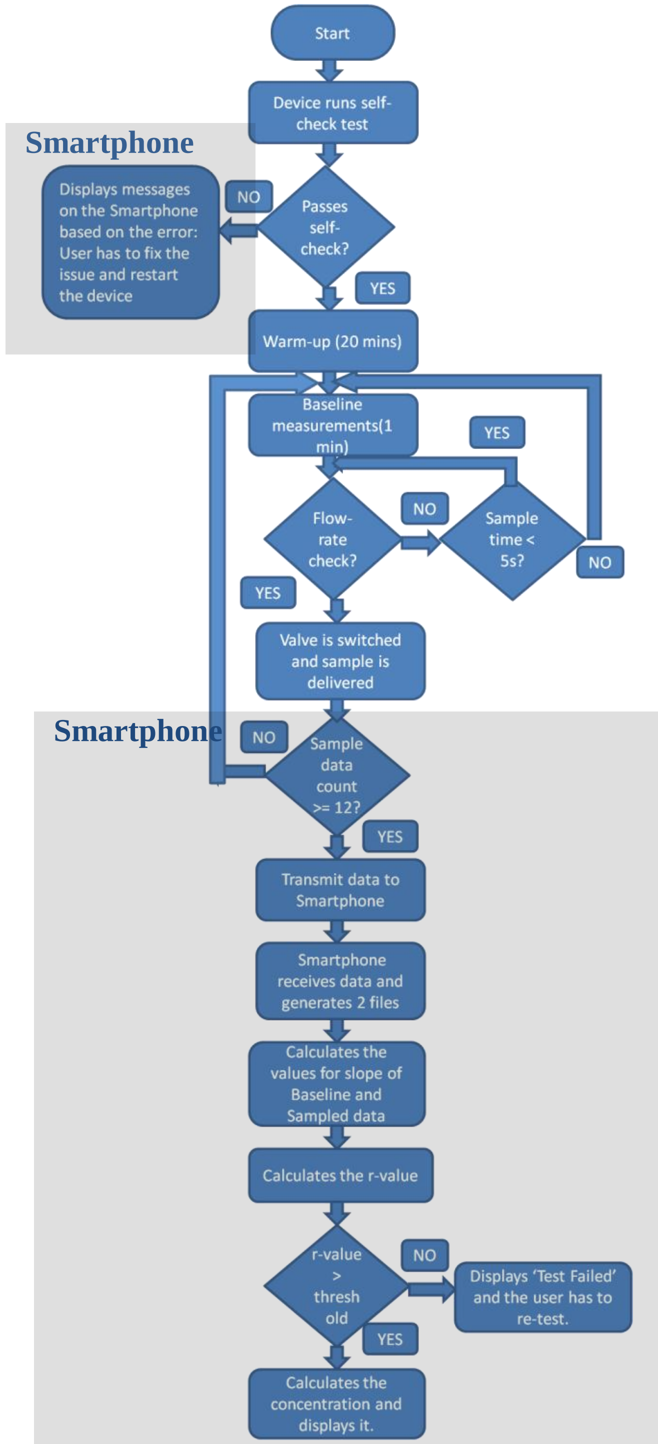


Fig.2. Flow diagram for Pressure-based system

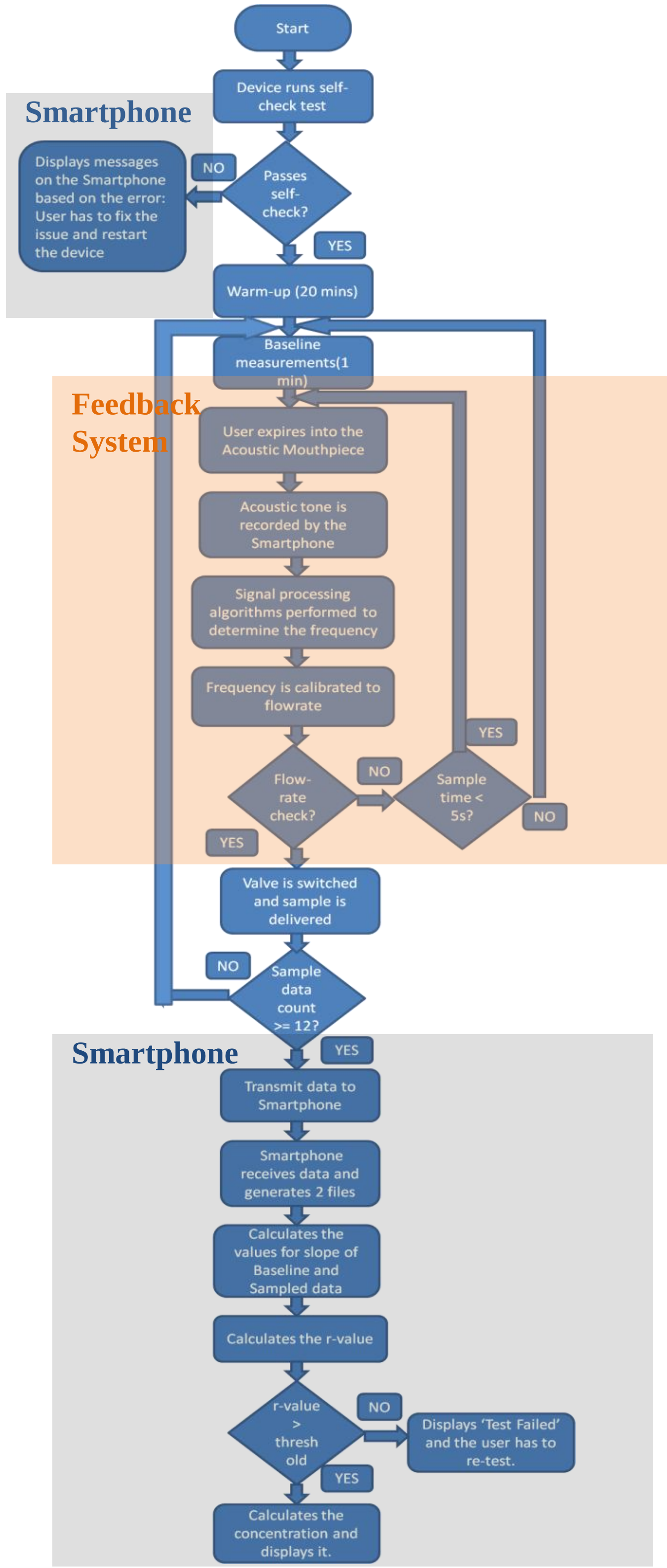


Fig.3. Flow diagram for Acoustic-based system

2.2 Environmental monitoring device

The Environmental monitoring device gives the concentration of Volatile Organic Compounds (VOC) and the count of airborne particles of diameter 2.5µm to 500nm in the environment. The VOC device, which was developed by Cheng Chen et al, [10] was modified with the addition of a particle detector module. The block diagram drawn below briefly illustrates the principle of operation of the device. The role of smartphone is very crucial here, since this device is a real-time device. Data is transferred to the device every second, and the concentration is delivered to the user immediately with the help of a plot as well as display of the value itself.

Fig.4. Block Diagram for Particle counter device

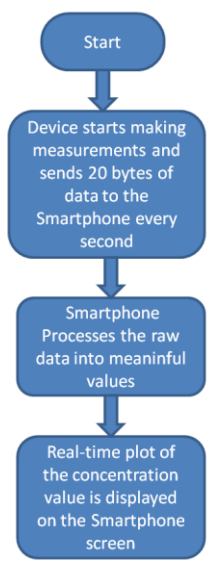


Fig.5. Flow diagram for Particle counter device

Chapter 3

MOBILE APPLICATION DESIGN

3.1 Mobile Application Structure

A mobile application can consist of a number of basic layers. The more common three-layer design shown in Figure 3.1 consists of the following layers [11]:

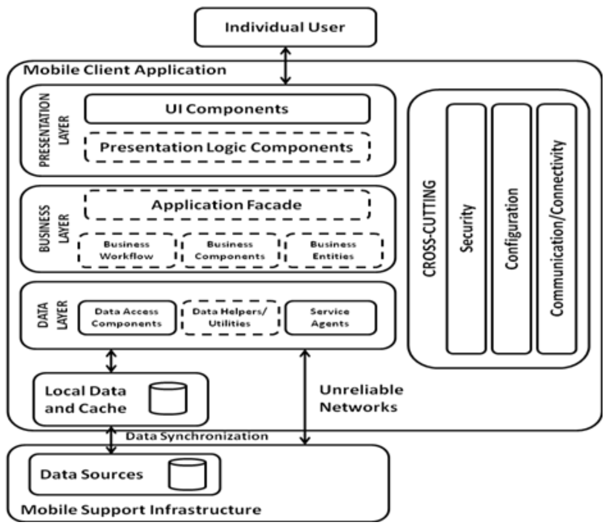


Fig.6. The typical structure of a mobile application [11]

- **Presentation layer:** This layer is very important as it represents the interface between the user and the rest of the application. It is not just about presenting information to the user, and also about providing the user with interactive environment in the application.

- **Business layer:** This layer implements the core functionality of the system and encapsulates the relevant business logic. Business logic is defined as "any application logic that is concerned with the retrieval, processing, and management of application data [12]".

- **Data layer:** This layer is used to access the data within the system and also the data transmitted by other systems in the network. The components in this layer make the application easier to configure and maintain.

3.2 User-Interface Design

The mobile environment is heterogeneous, and can change constantly. In general, mobile environment can be categorized as [13]:

- Dynamic, where the interaction between the system and user is highly fluid.
- Contextual, where applications are highly related to the context of where they are used.
- Limited attention, where users can only pay limited attention to applications.

Regardless of the type of mobile environment, there are certain design principles which are necessarily followed:

1) An interface should be easy to use from the first time user interacts with it. And the amount of functionality presented to the user should be limited to exactly what the user requires to achieve their goal.
2) Important information should be more visible and less important information should be less visible.
3) Usual tasks should be easy and less common tasks should be possible.
4) Number of steps it takes for a user to complete a task should be a bare minimum.

3.2.1 Asthma Analyzer

The Asthma Analyzer device is a personal sensor device and hence the user-interface is such that the user has to create a user id and every time the user enters the application, the user id and device id have to be entered to access the user's information or to run more tests. This way, the history of the user can be recorded and also, the user's privacy is protected.

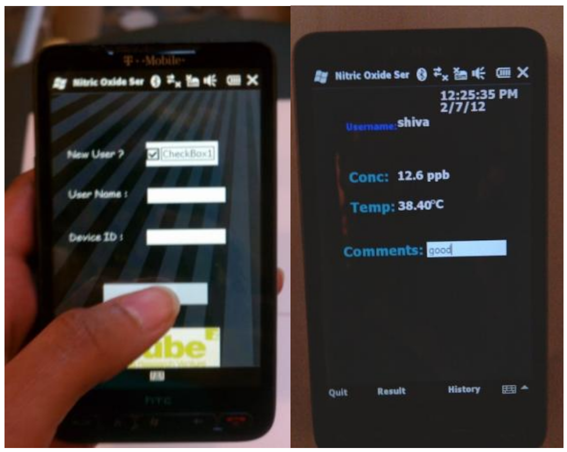


Fig.7. User Interface of Asthma Analyzer device

3.2.1 Environmental Monitor

The environmental monitoring device displays a real-time plot of the concentration of various components like Hydrocarbon A, Hydrocarbon B, Particle count, etc. The user can choose what he wants to see from the drop down menu on the smartphone.

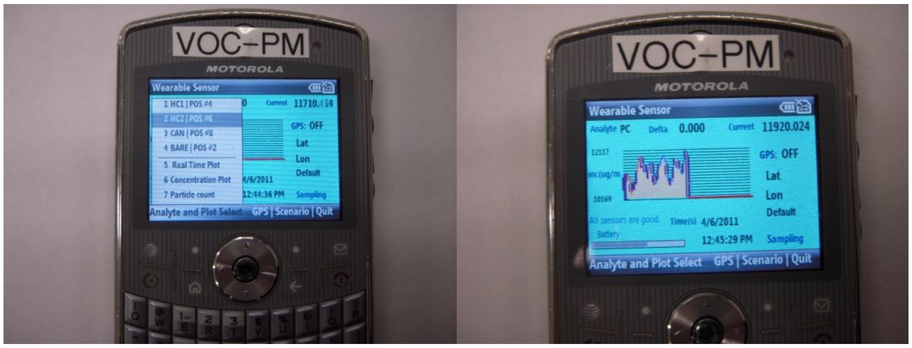


Fig.8. User-interface for Environmental monitoring device

3.3 Communication

Several wireless technologies such as GSM/GPRS, Bluetooth, ZigBee, WLAN IEEE 802.11, etc. can be used to transmit signals. The two main open standards currently used for medium data rate transfer are Bluetooth and Zigbee, both of which have the advantage of good connectivity between equipment from different manufacturers. In our system, Bluetooth is used to transmit data wirelessly, as Bluetooth has become a standard feature in the configuration of a smartphone. The RN-41 module designed by Roving Networks is used to provide Bluetooth connectivity. The Bluetooth connection is setup before data transmission. The baud rate of the serial-port on the smartphone is set to 19200 to synchronize with the baud rate set in the Bluetooth module and also in the device.

3.3.1 Asthma Analyzer

After one good measurement is obtained, the asynchronous data from the MSP430F5438 microcontroller is delivered to the RN-41 Bluetooth module on the serial port. In the whole system, the Bluetooth module is configured as a slave and the smartphone functions as the master. The data is sent as raw binary bytes.

Acoustic based device – Feedback system

Feedback is a mechanism, process or signal that is looped back to control a system within itself. Such a loop is called a feedback loop. In systems containing an input and output, feeding back part of the output so as to increase the input is called positive feedback, and feeding back part of the output in such a way as to partially oppose the input is called negative feedback [14].

Biofeedback is a non-medical process that involves measurement of a subject's specific and quantifiable bodily functions such as the activity of brain waves, blood pressure, heart rate, skin temperature, sweat gland activity and muscle tension, conveying the information to the patient in real-time [15]. This raises the user's awareness, and therefore the possibility of conscious control of those functions. By providing the user access to physiological information about which he may be unaware of, biofeedback may allow users to gain control of physical processes previously considered an automatic response of the autonomous nervous system.

In our system, the user has to expire into the mouthpiece for 6 seconds continuously at a flow rate in the range of 2700 to 3300 ml/min. When the user expires into the mouthpiece, based on the flow rate at which he expires, a single-tone acoustic signal is generated by the acoustic sensor. This signal is recorded by the microphone in the smartphone and the frequency of the signal is calculated as shown in section 3.4.3. The frequency is converted into flow rate in ml/min, based on the calibration that we have obtained. If the flow rate lies within the desired range of 2700 to 3300 ml/min, a feedback signal is sent to the device by the smartphone, that the frequency is in the desired range, and it is also displayed on the smartphone screen as well as on the device itself, through a Green LED that turns ON. When the flow rate is below the desired range, a feedback signal is sent to the device by the smartphone that the flow rate is lower than the range, and a Yellow LED turns ON in the device. If the flow rate is higher than this range, the smartphone sends a feedback that the flow rate is higher than the desired range and a Red LED turns ON in the device.

Fig.9. Visual Display providing feedback to the user

The user is given guidance to maintain the desired flow rate through the LED indicators. When the Green LED turns ON, the user has to maintain his expiration flow rate, when the Yellow LED turns ON the user has to expire harder and when the Red LED turns ON the user has to expire softer. This way, the user can streamline and maintain the flow rate in the desired range for 6 seconds continuously.

3.3.2 Particle Detector

As soon as the application is started, the smartphone connects to the Bluetooth module of the hand-held particle detector device. The data transfer here, is synchronous and real-time, taking place every second. Similar to the asthma analyzer device, the data is transferred as raw binary bytes. Every second, 20 bytes are transferred to the smartphone.

3.4 Data Processing

The raw binary data received from the device through Bluetooth is processed in the smartphone to obtain meaningful data, which can be used for calculation of output in a format deliverable to the user. When the raw data has been converted to meaningful data, further data processing is performed in each system in accordance with its own requirements.

3.4.1 Asthma analyzer

In the asthma analyzer device, the data transferred from the device to the smartphone are the resolution, photodiode voltages and the thermistor voltage. The device makes 2 measurements every second, which results in 2 data points per second. The baseline measurement is made for 60 seconds and the sampling measurement is done for 6 seconds, hence, in total there will be 60*2+6*2 = 132 data points generated. The data is sent in the following order: 2 bytes of resolution data, 4 bytes of photodiode1 data, 4 bytes of photodiode2 data and 4 bytes of thermistor voltage data. After combining the bytes together, the photodiode1, photodiode2 and thermistor voltage data are divided by the resolution to obtain the digital voltage. The value of the digital voltage is then converted to analog voltage by multiplying it by 3.3 and dividing the result by 4096. 3.3V is the reference voltage in the Analog-Digital Converter (ADC) in the microcontroller of the device and since the ADC has 12 bit resolution, there are 4096 possible digital representations of analog signal spread over the 3.3V.

The resolution represents the total number of measurements made every time period, which is 250 milliseconds. The photodiode voltage is the measure of voltage from the photodiode amplified by an operational transconductance amplifier. Photodiode1 is the photodiode for sensing channel and photodiode 2 is the photodiode for the reference channel. The thermistor voltage value is converted into temperature in degrees Centigrade using the conversion equation provided by the thermistor manufacturer [16], which is given by:

$$Temp = \frac{1}{\frac{1}{298} + \frac{1}{3518} * \ln\left(\frac{v_t}{0.0002} * \frac{1}{10000}\right)} - 273$$

where V_t is the voltage measured across the thermistor.

The negative logarithmic value of the ratio of the sensing channel voltage to the reference channel voltage is calculated to obtain the absorbance value.

$$\text{Absorbance} = -\log(\text{Photodiode}_{sensing}/\text{Photodiode}_{reference})$$

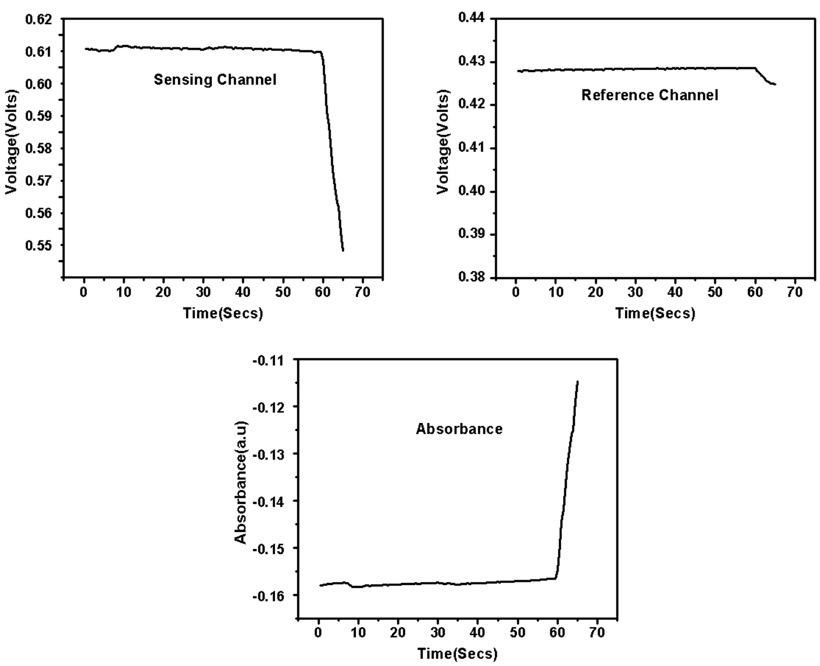


Fig.10. Voltage of Photodiode1, Photodiode 2 and Absorbance data

Calculation of Slope

The concentration of nitric oxide in the sample is proportional to the slope of the sampling absorbance value subtracted by slope of the baseline absorbance value. In order to obtain the slope values, we have to curve-fit the data. Using the concepts of Linear algebra [17], an equation for slope is derived, which is shown below.

'n' data points obtained need to be fitted to the straight line, the general form of which is,

$$y = mx + c$$

Each data point has to satisfy the above equation, so we have

$$y_1 = mx_1 + c; y_2 = mx_2 + c\ ; y_3 = mx_3 + c; \ldots\ldots\ldots; \ y_n = mx_n + c.$$

We need to solve for the values of slope 'm' and intercept 'c' from the above set of equations, which can be represented in matrix form as

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

This is an over-determined system with 'n' equations and only 2 unknowns. The best solution for 'm' and 'c' is that which minimizes the squared error [17]. For a system of linear equations with more number of equations than the number of unknowns represented by **Ax = b,** the least square solution or in other words, the best fit solution, $\mathbf{x}'$ is obtained by solving the set of equations,

$$(\mathbf{A}^T\mathbf{A})\,\mathbf{x}' = \mathbf{A}^T\mathbf{b}$$

Hence we can find the values of 'm' and 'c' as follows:

$$\begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

$$\begin{pmatrix} \Sigma x^2 & \Sigma \mathrm{x} \\ \Sigma \mathrm{x} & n \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \Sigma \mathrm{xy} \\ \Sigma \mathrm{y} \end{pmatrix}$$

Solving the two equations above for m and c,

$$m = \frac{\mathrm{n}\Sigma\mathrm{xy} - \Sigma\mathrm{x}\Sigma\mathrm{y}}{\mathrm{n}\Sigma x^2 - (\Sigma\mathrm{x})^2}$$

$$c = \frac{\Sigma\mathrm{y}\Sigma x^2 - \Sigma\mathrm{xy}\Sigma\mathrm{x}}{\Sigma x^2 - (\Sigma\mathrm{x})^2}$$

Using this equation of 'm' in the algorithm, we obtain the slope for baseline m_b and slope for sampling m_s. The concentration is proportional to the difference between these slope values (m_s-m_b).

Calculation of r-value

The overall quality of the fit is parameterized in terms of a quantity known as the Correlation Coefficient (r), which is defined as [17]

$$r^2 = m * m'$$

where m' is the coefficient in

$$x = m'y + c'$$

The coefficient m' is determined in the same fashion, so as to minimize the square of the residual on the basis of the data obtained.

$$m' = \frac{\mathrm{n}\Sigma\mathrm{xy} - \Sigma\mathrm{x}\Sigma\mathrm{y}}{\mathrm{n}\Sigma y^2 - (\Sigma\mathrm{y})^2}$$

$$r^2 = \frac{(\mathrm{n}\Sigma\mathrm{xy} - \Sigma\mathrm{x}\Sigma\mathrm{y})}{(\mathrm{n}\Sigma x^2 - (\Sigma\mathrm{x})^2)} \frac{(\mathrm{n}\Sigma\mathrm{xy} - \Sigma\mathrm{x}\Sigma\mathrm{y})}{(\mathrm{n}\Sigma y^2 - (\Sigma\mathrm{y})^2)}$$

$$r = \frac{(n\Sigma xy - \Sigma x \Sigma y)}{\sqrt{(n\Sigma x^2 - (\Sigma x)^2)(n\Sigma y^2 - (\Sigma y)^2)}}$$

If the value of $r \geq 0.90$, the fit is considered good, if it is $r < 0.90$, then the smartphone displays 'Test Failed' and the user is instructed to re-test.

3.4.2 Particle Detector Sensor

In the particle detector the data transferred from the device to the smartphone are tuning fork measurements, particle count data and valve status data. The data is sent in order as, 8 bytes of tuning fork data, 2 bytes of unused data, 2 bytes of particle count measurement1, 2 bytes of particle count measurement2, 2 bytes of particle count measurement3 and 4 bytes of valve status data, constituting a total of 20 bytes of data. The digital voltage measurement is obtained by combining the bytes together and will be converted to their analog voltage values

The particle detector device measures the concentration of total VOC with the help of miniaturized tuning forks. The tuning forks are chemically coated and their resonant frequency changes when hydrocarbon attaches to the tuning fork surface. The change in the resonating frequency is used in detemining the concentration of the hydrocarbon. There is a 2 mins purging period and a 1 min sampling period. The valve is used to switch between purging and sampling periods. The valve status is the variable which is used to determine whether the device is in purging or sampling state.

The particle count voltage is measured thrice and sent from the device, which is done in order to reduce noise. The average of only the good measurements should be used. In order to implement this, difference between the three voltage values are obtained, giving 3 new data.

diff1 = |PC1 – PC2|; diff2 = |PC2 - PC3|; diff3 = |PC3 - PC1|

where PC1, PC2 and PC3 are the analog particle count voltage measurements.

Now each of this difference data is checked to see if it is below a certain threshold value. Based on some experimentation done, the threshold value is set to 0.01V. If any two of the difference values are below 0.01V, then the average of all 3 data points is obtained. If only one of the difference values is below the threshold value, the average of those voltage values which were used to obtain the difference is considered.

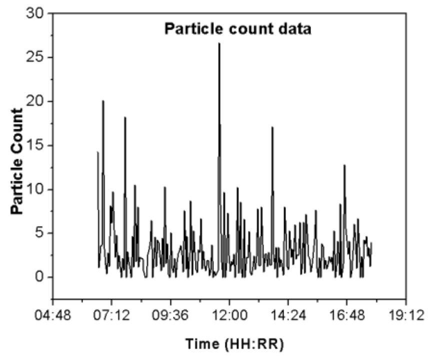


Fig.11. Real-time plot of averaged particle count data.

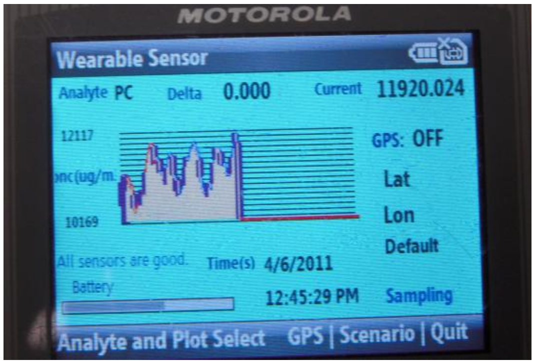


Fig.12. Real-time plot of averaged particle count data on smartphone.

3.4.3 Audio signal processing in the acoustic-based device

The output frequency of the acoustic mouthpiece is calibrated with the expiration flow rate. Hence, the application records the audio signal, processes it to get the frequency and sends a signal to the device if the flow rate is within range or otherwise. All these steps are performed every 250 milliseconds during the whole sampling period of 6 seconds. The frequency of the acoustic signal is obtained through a FFT.

The Cooley–Tukey algorithm is the most common Fast Fourier Transform (FFT) algorithm [18]. It re-expresses the Discrete Fourier Transform (DFT) of an arbitrary composite size, $N = N_1N_2$ in terms of smaller DFTs of sizes N_1 and N_2 recursively, in order to reduce the computation time to O ($N \log N$) for highly-composite N (smooth numbers).

The discrete Fourier transform is defined by the formula [18],

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}$$

,where k is an integer ranging from 0 to (N-1).

A radix-2 decimation-in-time (DIT) FFT is the simplest and most common form of the Cooley–Tukey algorithm, although highly optimized Cooley–Tukey implementations typically use other forms of the algorithm as described below. Radix-2 DIT divides a DFT of size N into two interleaved DFTs of size $N/2$ with each recursive stage, which give it the name "radix-2".

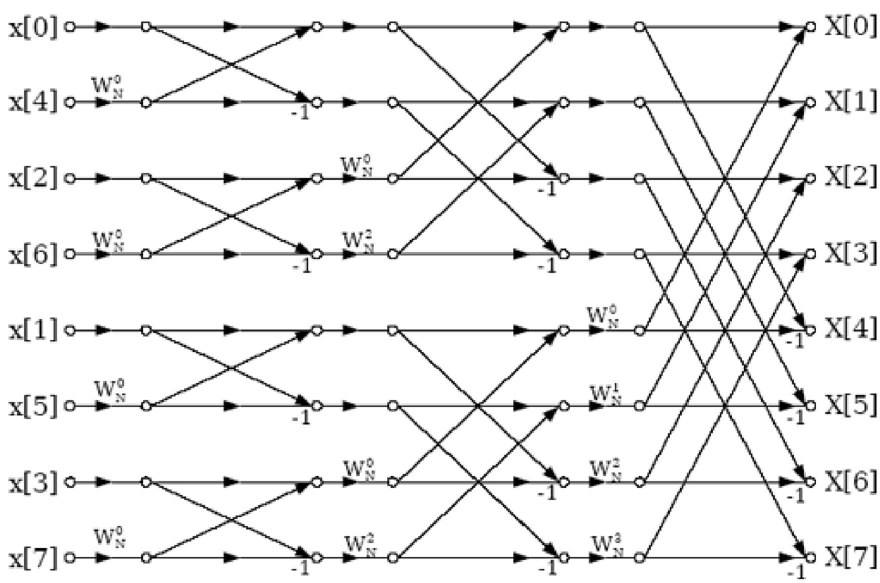


Fig.13. Radix-2 butterfly diagram [19]

Radix-2 DIT first computes the DFTs of the even-indexed inputs x_{2m}($x_0, x_2, \ldots, x_{N-2}$) and the odd-indexed inputs x_{2m+1} ($x_1, x_3, \ldots, x_{N-1}$), and then combines those two results to produce the DFT of the whole sequence. This idea can then be performed recursively to reduce the overall runtime to O (*N* log *N*).

Also seen in figure 13, the order of the input data is not x[0], x[1], x[2], x[3].., etc, it is instead x[0], x[2], x[4], x[6], which is called Bit Reversal. The important idea is that the binary numbers are the reversals of each other. The FFT time domain decomposition is usually carried out by a bit reversal sorting algorithm. This involves rearranging the order of the *N* time domain samples by counting in binary with the bits flipped left-for-right.

Sample numbers in normal order	
Decimal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Sample numbers after bit reversal	
Decimal	Binary
000	0
001	4
010	2
011	6
100	1
101	5
110	3
111	7

Table.1. Bit reversal

The Radix-2 DIT algorithm rearranges the DFT of the function x_n into two parts: a sum over the even-numbered indices $n = 2m$ and a sum over the odd-numbered indices $n = 2m + 1$.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}.$$

The DFT of the Even-indexed inputs x_{2m} are denoted by E_k and the DFT of the Odd-indexed inputs x_{2m+1} by O_k and we obtain:

$$X_k = E_k + e^{-\frac{2\pi ik}{N}} O_k$$

However, these smaller DFTs have a length of *N*/2, so we need compute only *N*/2 outputs, thanks to the periodicity property of the DFT, the outputs for $\frac{N}{2} \leq k < N$ from a DFT of length *N*/2 are identical to the outputs for $0 \leq k < \frac{N}{2}$. That is, $E_{k+\frac{N}{2}} = E_k$ and $O_{k+\frac{N}{2}} = O_k$. The phase factor $e^{-2\pi ik/N}$ (called a twiddle factor) obeys the relation, $e^{-2\pi i(k+\frac{N}{2})/N} = e^{-\pi i}e^{-2\pi ik/N}$ = -$e^{-2\pi ik/N}$, flipping the sign of the $O_{k+\frac{N}{2}}$ terms. Thus, the whole DFT can be calculated as follows:

$$X_k = \begin{cases} E_k + e^{-\frac{2\pi i}{N}k} O_k & \text{if } k < N/2 \\ E_{k-N/2} - e^{-\frac{2\pi i}{N}(k-N/2)} O_{k-N/2} & \text{if } k \geq N/2. \end{cases}$$

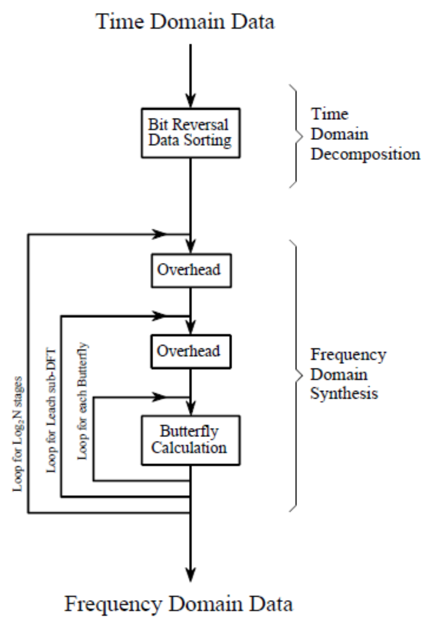


Fig.14. Flow diagram of FFT [18]

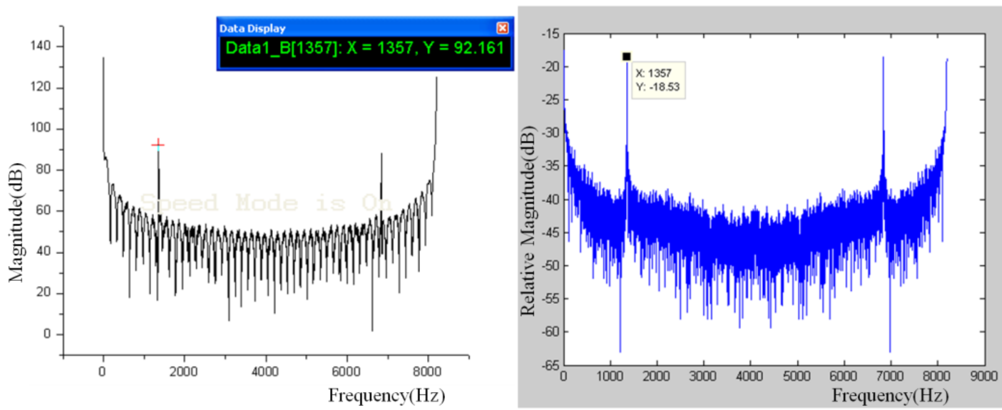


Fig.15. Result from the smartphone compared with the result obtained from MATLAB program

The signal is recorded by the microphone with a sampling frequency of 22050 samples per second. The signal is recorded for 180 milliseconds, which gives $180*22050*10^{-3} = 3969$ samples per frame. The Radix-2 algorithm works only for a sample size in the powers of 2, therefore the samples are appended with zero to the next power of 2. By adding 127 zeroes to 3969 samples we obtain 4096 samples, which is 2 to the power 12, hence the value of N = 4096. This corresponds to 1 frame of audio data, and the same is repeated for 6 whole seconds which implies $6*10^{3}/250 = 24$ times.

After obtaining the FFT spectrum, the maximum value and the index of the maximum value are found. Using this information in the formula below, the value of frequency is obtained.

$$f = index_{max} * f_s * N/2 \quad \text{, where } f_s = 22050 \text{ and } N=4096$$

Algorithm for Noise Reduction:

In order to reduce the interference caused by ambient/background noise present in the room, a very simple algorithm is developed. Before recording the signal generated by the acoustic mouthpiece, a baseline audio signal of the environment is recorded as soon as the application begins, and waits for the ready signal from the device to start recording the audio signal from the mouthpiece. This baseline signal is then subtracted from each frame of audio data before doing the FFT calculation. This algorithm is found to improve the performance of the system significantly.

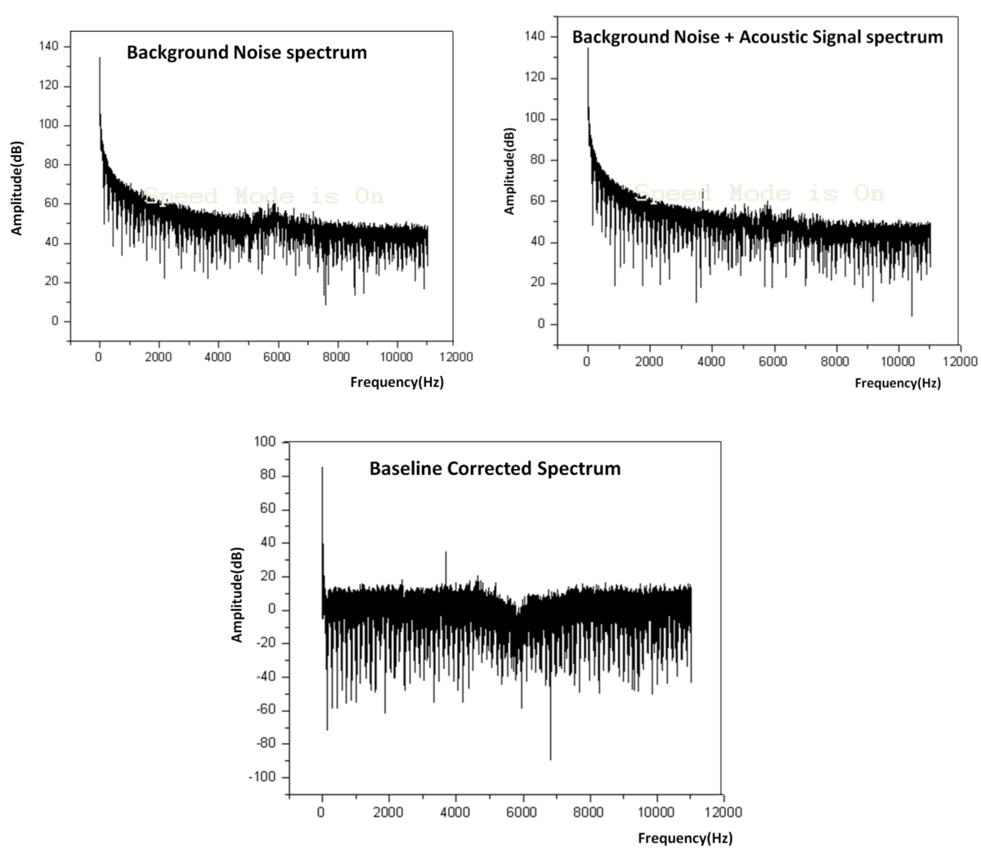
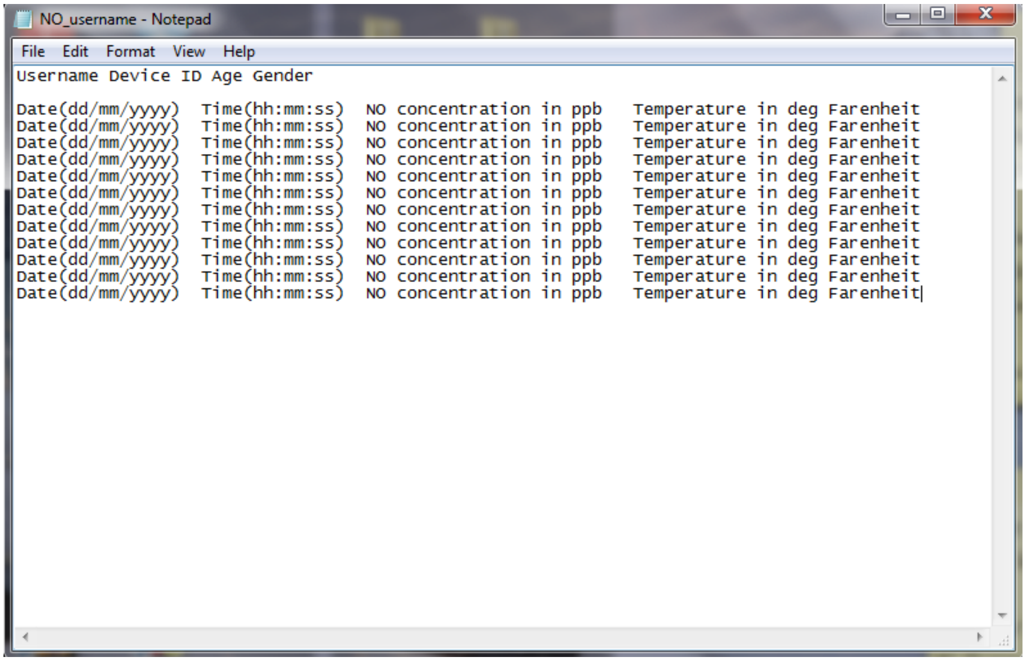


Fig.16. Result showing reduction of background noise.

3.5 Database

Currently, the applications store all the raw data as well as processed data in the smartphone memory. Data stored in the memory takes up to 21 kilobytes of memory for every test. In the future, the data will be stored in a network by means of Cloud Computing technology.


NO_username - Notepad

File Edit Format View Help

```
Username Device ID Age Gender

Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
Date(dd/mm/yyyy)  Time(hh:mm:ss)  NO concentration in ppb   Temperature in deg Farenheit
```

Fig.17. Typical data file of a user

Chapter 4

TESTING

4.1 Need for Testing

A famous quote by Bruce Sterling goes - "*Some software is bad and buggy. Some is 'robust,' even 'bulletproof.' The best software is that which has been tested by thousands of users under thousands of different conditions, over years. It is then known as 'stable.' This does NOT mean that the software is now flawless, free of bugs. It generally means that there are plenty of bugs in it, but the bugs are well-identified and fairly well understood [21]*".

Software development involves human errors which may be observational errors, conceptual errors, discoursive errors etc. Every change made to a piece of software may introduce some errors to it, and the software may not function the way it is expected to do. There may be errors as a result of defects in the design or modeling of the software itself. Testing exposes all the errors, paves a pathway to solve the errors and reduces the probability of the software failing at critical conditions.

4.2 Black-box vs. White-box Testing

The black-box approach, as the name suggests, is a testing method in which test data are fed to the system and the output data is obtained without any consideration about the structure or flow of the program. It is also termed data-driven, input/output driven, or requirements-based testing. Functional testing falls under the category of Black-box testing for the same reason that only the functional requirements of the program needs to be satisfied for this kind of testing, which means for the given set of inputs, the output should correlate with the expected output of the program. In testing, various inputs are exercised and the outputs are compared against standard values to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered. User Acceptance Testing (UAT) and Systems Testing are classic examples of black-box testing.

As opposed to black-box testing, the structure and flow of the software under test are given great importance in the white-box testing. Testing plans are modeled based on the implementation of the software, such as programming language, the logic flow and even the memory access. Test cases are derived from the program structure. White-box testing is also called glass-box testing, logic-driven testing or design-based testing. Although white-box testing techniques can be used at any stage in a software product's life cycle, they tend to be found in unit testing activities [21].

4.3 Unit Testing

The first type of testing that can be conducted in any development phase is unit testing. The primary goal of unit testing is to take the smallest block of functional software in the application, isolate it from the rest of the code, and determine whether it behaves exactly as expected. The whole software is split in different units and each unit is tested separately before integrating them into complete software application. Unit testing has been proven to be very effective in determining the defects in the program as the code is smaller and easier to track bugs and errors down at this stage. Unit testing is a white-box testing technique.

Calculation of Slope

A set of test data points are generated and delivered to the program unit which calculates the slope of the data points. The same set of data points are used in ORIGIN (off-the-shelf software) and the slope is calculated by using the linear fit method. Also, the data points are used to manually calculate the slope value using MS-EXCEL by means of the equation which was derived in section 3.4.1. The results are tabulated below:

Smartphone App	Origin	Excel
-0.1257	-0.1296	-0.12765
-0.16529	-0.16529	-0.16529
-0.07929	-0.07929	-0.07929
-0.0881	-0.08944	-0.08877
-0.10759	-0.10759	-0.10759
-0.06376	-0.06376	-0.06376
-0.06425	-0.06425	-0.06425
-0.05967	-0.05967	-0.05967
-0.30096	-0.30096	-0.30096
-0.49802	-0.49802	-0.49802
0.4396	0.4396	0.4396
-0.243	-0.24276	-0.24288
1.4595	1.45909	1.459295
0.2765	0.27622	0.27636

Table.2. Comparison of slope values

This proves that the part of the program which calculates the slope does its function effectively and has passed the Unit Test.

Testing Bluetooth data reception

Bluetooth reception by the serial-port is checked by connecting the microcontroller to the programming interface by cables and the same set of data which is wirelessly transmitted, is verified at the computer through the programming interface. The data is listed below in table.3, which shows the accuracy of the communication between the Bluetooth module and the Serial-port of the smartphone.

Received by Smartphone	Reading from Memory
10752944	10752944
16178685	16178685
16181515	16181515
16180866	16180866
16176334	16176334
16188204	16188204
16176402	16176402
16176906	16176906
16177269	16177269
16176284	16176284
16175665	16175665

Table.3. Bluetooth communication verification

Calculation of FFT

The accuracy of the FFT calculation is verified by comparing the results produced by the smartphone application and a MATLAB code for calculation of FFT using the standard fft MATLAB function.

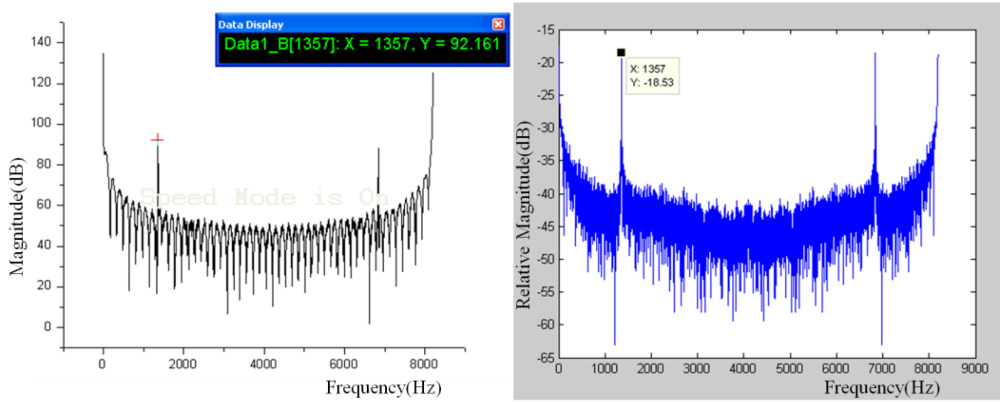


Fig.18. Comparing the output

Frequency in Hz	
Smartphone App.	MATLAB code
3685	3685
4342	4342
3534	3534
2875	2875
3318	3318
4129	4129

Table.4. FFT algorithm verification

The same source of acoustic signal is recorded by the microphone of the smartphone and the microphone of a personal computer (Laptop) simultaneously. FFT calculations are made on the recorded acoustic wave signal in the personal computer by a standard MATLAB FFT code and these values are compared with the values produced by the smartphone application. The frequency values are always found to be the same. This test not only validates the FFT algorithm, but also confirms the performance of the smartphone microphone as compared to the microphone of a personal computer.

Calculation of R-value

Coefficient of correlation is calculated to check the linearity of the data point. Based on the linearity value, it is determined if a test is a good or bad test. Hence making sure that this part of the software works to requirement is very essential as it can help reduce the risk of making wrong decisions about a test result being good or bad. The r-value of some data points are calculated using Origin and the program, and the values are compared. Results are as shown below in the table.5, which shows that there is a good correlation between the values calculated by the program and Origin.

Smartphone App.	Origin
-0.81417	-0.8091
-0.21209	-0.2040
0.7214	0.7441
0.9812	0.9830
0.8854	0.8774
0.5140	0.5562
0.7896	0.7925
0.9213	0.9168
0.8762	0.8814

Table.5. Comparison of r-value generated by the smartphone app and Origin.

Chapter 5

CONCLUSION

This work demonstrates designing, developing and testing of reliable smartphone applications for supporting m-health and environmental monitoring devices. The layered application design, as guided by Microsoft's msdn [11] guidelines is followed and implemented. The user-interface is fairly simple and the feedback system improves the performance of the hand-held device. The algorithms are fast, robust and reliable and tested against standard methods.

There are two systems which are discussed mainly in the report; one is a m-health system and the other an environmental monitoring system. Both have suitable user-interface and respective data processing performed. The Bluetooth communication between the device and the smartphone is the very important and common feature in both systems. Nevertheless, the data transfer are different in the two systems: the m-health system has an asynchronous data transfer while the environmental system has synchronous real-time data transfer. The conclusion from the report is that smartphone application is an appropriate solution for mobility in real time data analysis in environmental monitoring system as well as in healthcare systems.

Chapter 6

FUTURE WORK

"Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility over a network, typically the Internet." – as defined by Wikipedia [22].

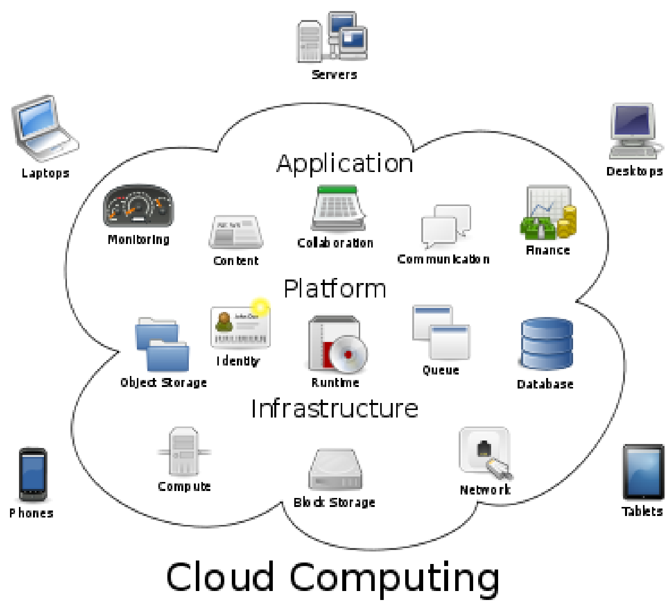


Fig.19. Cloud computing concept [22]

In a cloud computing system, there is a significant workload shift. The devices which are connected to the cloud need to be able to run the cloud computing software which could be a web browser or a simple mobile application and the cloud computing application servers provide the same service and performance as if the software program were installed on the device itself. Hence the specifications on personal devices get simplified and the devices will get lighter and smaller eventually.

A cloud computing system can be divided into two sections: the **front end** and the **back end**. They are connected to each other through the Internet. The front end is the side the computer user, or client, sees. The back end is the "cloud" section of the system which includes various computers, servers and data storage systems. A central server administrates the system, monitors traffic and client demands to ensure everything runs in a smooth manner [23].

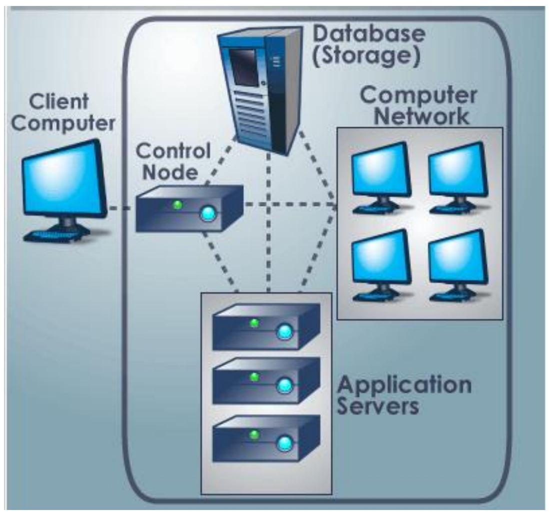


Fig.20. Architecture of Cloud computing [23]

Cloud computing servers offer three kinds of services based on their fundamental model.

1) Infrastructure as a Service: This is the most basic model of cloud service, cloud computing servers provide virtual machines to the users i.e. they are provided with resources like memory.
2) Platform as a Service: As the name suggests, this service provides a platform for running or even developing software. This may include operating system, compiler/interpreter for running the program etc.

3) Software as a Service: This is the lightest service for the cloud computing servers to offer, providing application software. Examples include Gmail, Yahoo, Facebook, some online games like World of Warcraft.

Cloud computing can be implemented in our systems as well as a Software service or as a Infrastructure or as both, where the users can run the application on the cloud as well as save the data on the cloud. If this is done, then:

- Users would be able to access their applications and data from anywhere and at any time. Data wouldn't be confined to the memory of a user's smartphone.

- Users wouldn't need to allocate significant amount of memory for running our applications, because they would be able to store all their information on a remote computer.

But, there are some issues which are to be addressed as well, for example, the user may not be able to run the application when he is in an area where the reception is poor and hence he is not able to access the internet. One solution to this problem is to have a backup application stored in the smartphone itself, which can be run without accessing the cloud and when he gets reception back, he could upload the result of that particular test to the cloud.

REFERENCES

[1] (16 March 2012). from http://en.wikipedia.org/wiki/MHealth#Definitions

[2] “Mobile Healthcare Opportunities, Monitoring Applications & m-Health Strategies” 2010-2014. Juniper Research. April 13, 2010.

[3] Tsou, M.-H. "Integrated Mobile GIS and Wireless Internet Map Servers for Environmental Monitoring and Management."Cartography and Geographic Information Science, Vol. 31, No. 3, 2004, pp. 153-165

[4] http://www.smartphoneappspedia.com/

[5]http://pressroom.nvidia.com/easyir/customrel.do?easyirid=A0D622CE9F579F09&version=live&prid=819304&releasejsp=release_157

[6] http://www.intel.com/products/processor/core2xe/index.htm

[7] “New Asthma Estimates: Tracking Prevalence, Health Care and Mortality,” NCHS, CDC, 2001

[8] Yates, D. H. (2001). "Role of exhaled nitric oxide in asthma." Immunology and cell biology 79(2): 178-190.

[9]Hogman,M. Stromberg,S(1997) “Nitric oxide from human respiratory tract efficiently quantified by standardized single breath measurements” , Scandanavian Physiological Society.

[10] Chen, C (2011). “A Wireless Hybrid Chemical Sensor for Detection of Environmental Volatile Organic Compounds”

[11] Microsoft Corporation. (2009). "Microsoft Application Architecture Guide." 2nd. from http://msdn.microsoft.com/en-us/library/ff650706.aspx.

[12] http://msdn.microsoft.com/en-us/library/ee658103.aspx

[13] Moe, K. H., B. Dwolatzky, et al. (2004). Designing a usable mobile application for field data collection, IEEE.

[14] (16 March 2012).from http://en.wikipedia.org/wiki/Feedback

[15] (24 February 2012).from http://en.wikipedia.org/wiki/Biofeedback

[16] http://sensing.honeywell.com/product%20page?pr_id=53767

[17] Strang, G. (2006). Linear Algebra and its Applications 4th Edition , Brooks/Cole Pub Co.
[18] Smith, S. (1999). The Scientist and Engineer's Guide to Digital Signal Processing.

[19] http://cnx.org/content/m12016/latest/.

[20] Jenkins, N. (2008).”A Software Testing Primer”, Creative Commons, SanFrancisco, California, USA.

[21] Pan, J. (1999). "Software Testing." From http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/

[22] (15 March 2012). from http://en.wikipedia.org/wiki/Cloud_computing.
[23] Strickland, J. "How Cloud Computing Works." from http://computer.howstuffworks.com/cloud-computing/cloud-computing.htm
[24] Holtzblatt, K. (2005). "Customer-centered design for mobile applications." Personal and Ubiquitous Computing **9**(4): 227-237.
[25] Satyanarayanan, M., P. Bahl, et al. (2009). "The case for vm-based cloudlets in mobile computing." Pervasive Computing, IEEE **8**(4): 14-23.