EE202C

Networked Embedded Systems Design

Micropower Wireless Health Sensors:

A Micropower Wireless Oxygenation Monitor for Emergency Medicine

Design Document

Xiang Li

Teng Huang

Zi Huang

Yan Qu

Version 2

3/24/2013

Revision History

Version	Date	Author	Comments
Version No	Date Completed	Author(s)	Comments on Versions
1	2/8/2013	Zi Huang, Yan Qu, Xiang Li, Teng Huang	
2	3/24	Zi Huang, Yan Qu, Xiang Li, Teng Huang	

Table of Contents

Revision History 2

Table of Contents 3

1 Abstract 4

2 Team 4

3 Technical Approach 4

4 System Design 6

5 Implementation Schedule 9

6 Implementation Description 9

7 System Source Code 16

8 Test and Evaluation System 16

9 Test System Source Code 17

10 Test Results 17

11 Analysis 19

12 Shared Systems 20

13 References 20

1 Abstract

Mobile health devices to measure and monitor blood oxygenation and heart rate of patients are needed in case of emergencies to provide immediate and accurate data for clinicians. The development of novel micro power wireless health sensors, as well as low power platforms supporting Bluetooth communication and local sensor data processing will fulfill this need. Besides, the Android interfaces receiving data from Bluetooth module also enables us to monitor the real time measurement on smart phones. This project is aiming to develop a low-power, non-invasive, unobtrusive, and easy-to-use compact system that reports users' blood oxygenation level and heart rate. The design will make use of optical sensor to get analog signal, low power embedded system for local digital signal processing and oxygenation level calculation, a Bluetooth module to communicate with the Android application, and the Android interface to display the calculated results and its real time plot. The final design will be capable of providing reliable full optical blood oxygenation and heart rate measurement. The accuracy, robustness, and power consumptions in different situations of emergency are the several criterions to be considered during the process of design and testing.

2 Team

Xiang Li

Teng Huang

Zi Huang

Yan Qu

3 Technical Approach

3.1 General objectives

Primary objective:

Design and implement a compact micro power wireless health device for noninvasive, accurate, immediate blood oxygenation and heart rate measurement in emergency use.

Subdivided goals:

1) Acquire raw analog signal from optical sensor

The analog data is basically the absorption of different wavelength of light to indicate the absolute blood oxygenation level. It is transported through serial port from the optical sensor to the Mbed and an A/D converter converts the analog signal into a digital one for further signal processing.

2) Realize DSP algorithms and measurement data calculation on Mbed platform

The signal processing algorithm includes low pass filtering (FIR), peak detection, and oxygenation level calculation and calibration. The low pass filter filters out the high frequency noise and with peak detection to get the DC and AC component of the waveform for oxygenation calculation and heart rate. The calibration is needed to make the result more accurate.

3) Connection and data transfer between Bluetooth module and Android app

The Bluetooth module (RN-42) integrated on Mbed is to establish the connection with Android application and sent the calculated data to the smart phone. The process can be achieved using Android Bluetooth API.

4) Display the measurement on Android interface and generate the real time plot curve

A user-friendly interface is needed to display the real time heart rate and oxygenation level. A real time plot is of the waveform and bar graph is preferred for emergency uses.

3.2 Challenge and constraints

1) Motion noise: the movement of the subject or the sensor may cause abnormal waveform.

2) Power consumption: it will a challenge if expect to achieve high accuracy and data rate.

3) Sensor attachment lost: the attachment needs to be confirmed for accurate measurement.

4) Bluetooth connection lost: once the connection is lost, it should reconnect and report.

5) Subject to subject variance: the device may not adapt to some of the individuals.

3.3 Methods and approach

1) Reduce noise

With filtering algorithms on Mbed to filter out the high frequency noise. Besides, increase the sampling rate and do averaging will help reducing the noise.

2) Lower power consumption

The devices used such as Mbed and RN-42 are all low power devices. Optimizing the DSP algorithm will help to reduce power consumption. A lower-power mode can be introduced: the device self-start when attached to the subject (work mode) and switch to low power mode (sleep mode) after measurement. A battery state indicator reports low power status on Android application interface to remind user to charge.

3) Sensor attachment detection

An optical proximity sensor can detect the distance between the sensor and the subject. It should be able to report if the sensor is not placed properly.4) Bluetooth pairing and reconnection

The Mbed should only pair with the Android platform automatically with this specified application with code.

5) Calibration

Multiple sensors can be used and select the best signal to process in an array of sensors.

4 System Design

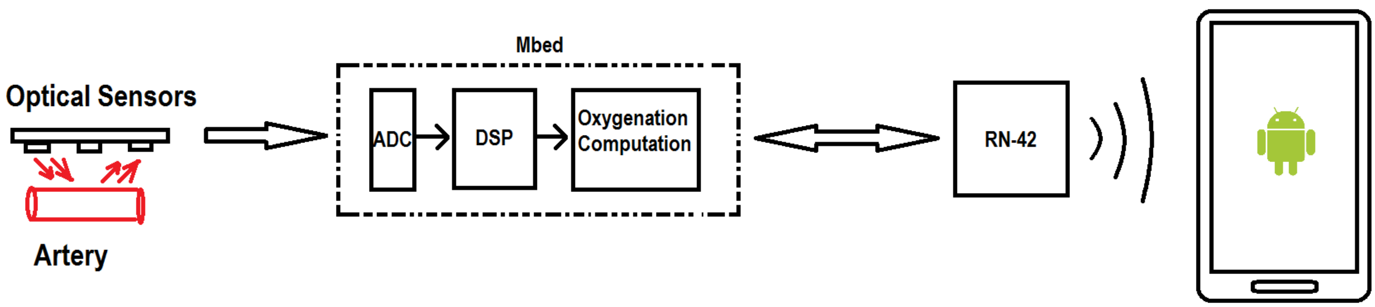


Figure 1

The proposed system consists of four major parts:

- Optical sensor system- Infrared and red sensor array
- mbed microcontroller- mbed NXP LPC11U24
- Bluetooth module- RN-42
- Custom designed Android Application

1) Optical sensor

- The sensor array can consist of 8/16/32 sets of optical sensors. The circuit is based on Kapton flexible electronic printed circuit technology which makes it possible for the sensors to be closely attached to human body. The sensors measure the optical absorption of blood and convert it into electrical signal.

2) mbed microcontroller

The mbed microcontrollers are a series of microcontrollers development boards designed for fast, flexible and low-risk and professional rapid prototyping [1]. In the oxygenation monitor system, it is responsible for taking in the analog signal from the sensor array, and converting it into digital data ready for blood oxygenation computation. Furthermore the mbed will complete the oxygenation computation on chip and transmit the result to the Andriod application via RN-42 the Bluetooth module.

3) Bluetooth module

RN-42 will be transmitting the blood oxygenation measurement from mbed to an Android device, and will also be responsible for receiving any configuration instructions from the Android application.

4) Android Application

The custom designed Android application will receive the blood oxygenation measurement and will make a clear, real-time display of the oxygenation level on screen. In addition, the application will also have the ability to communicate back to mbed via Bluetooth about basic measurement configurations.

Interfacing

- **Sensor-mbed**

The sensor array output signal swing depends on its Vdd. Mbed's 3.3V regulated out should be used to provide Vdd for the sensors. The sensor output will be feed into mbed's analog input to be digitalized. On the other hand, the mbed would use its digital output pins to drive the sensors. Specifically, the sensors—red LEDs and infrared LEDs-- should be driven by a square wave timed in such a way that the red LEDs and infrared LEDs toggle on and off alternatively.

- **mbed-RN42**

RN42 module will communicate with mbed via its UART transmit and receive pins, which will be connected to the corresponding UART pins on mbed. RN-42 will also use mbed's 3.3V regulated Vout as its power supply.

- **RN42-Android**

Upon running, Android application should automatically look for and setup the connection between RN-42and the Android device.

Algorithm implementation

- **DSP Algorithm**

The DSP algorithm implemented on mbed should clear out all the high frequency noise and movement noise, leaving the blood pulse characterized waveform. However, the DC offset of the sensor signal should be kept, since it will be used to calculate blood oxygenation and improve movement noise immunity and determine subject-to-subject variance.

- **Oxygenation Computation**

The blood oxygenation can be calculated by:

$$R = \frac{AC_{RED}/DC_{RED}}{AC_{IR}/DC_{IR}}$$

Where AC_{RED} and AC_{IR} are to the peak values of the waveform for red LED signal and infrared LEDs signal, and DC_{RED} and DC_{IR} are DC offsets for corresponding DC offset. [2]

The data will be processed in real time and transmitted to an Android device via RN-42.

Challenges:

1) Interface

The interfacing between the sensors and mbed may require more testing and careful design. The mbed only has 6 analog input pins, which means at most 6 sensors can take measurements at the same time.

On the other hand, the DC offset may take up most of the input swing of the ADC while the actual waveform has relatively small amplitude. The sampling rate of ADC and the frequency of the square wave that drives the sensors should be carefully chosen so that no additional noise is introduced.

2) Power requirement

The monitor should be able to take measurement for a relatively long period of time, which requires either a bigger battery or lower power consumption. Since the design is meant to be a low-cost product, having low-power consumption will be the reasonable solution.

On the other hand, when the sensors are driven by a lower voltage, the signal gets weaker thus resulting in a larger SNR, which will be demanding for the DSP algorithm. Having a lower-power mode should be considered a possible solution. However the configurable options will require more complicated implementation, thus increase the design complexity.

3) Placement and calibration

The final design should be able to auto-calibrate to different location of a human body and subject subject-to-subject variance. Having an array of sensors helps because the mbed can be programmed to recognize the situation based on the difference of each sensor, and can also pick the best signal to process. However a 6-sensor array may not be sufficient under all circumstances and may require more effort on the DSP part.

5 Implementation Schedule

This is the general format. For our first assignment, an abbreviated format is quite adequate. For this section please include dates for:

1. Design completion date

   2/19/2013
2. Test plan completion date

   2/8/2013
3. Design review date (this would be during one of the Thursday or Friday laboratory hours). This design review may also include a comments back from your instructor.

   2/21/2013
4. Implementation completion date (code freeze)

   2/28/2013
5. Testing phase complete

   3/8/2012

6 Implementation Description

1. Android Application

   8 Java classes: BluetoothChat; BluetoothChatService; DeviceListActivity; BarActivity & BarGraph; LineActivity & LineGraph; MockData & Point.

   BluetoothChat class is the holding the main user interface. It sets up everything for Bluetooth connection and real-time plot data. It is also responsible for reading message from remote Bluetooth device.

   BluetoothChatService and DeviceListActivity classes are in charge of Bluetooth connection management. DeviceListActivity is for showing the Bluetooth devices detected and respond when user clicks on one item. BluetoothChatService class manages 3 threads for Bluetooth connection: AcceptThread, ConnectThread and ConnectedThread. The ConnectedThread is the thread we use to communicate with Bluetooth device. We use socket and stream to read/write message from/to Bluetooth device, then send the message to main activity(BluetoothChat) for display.

BarActivity and BarGraph are two classes which manage the bar graph plot. They import the AchartEngine library we imported in libs folder(built in path). In BarActivity class, we create a new thread to control the plot to update every one second. And because the BarGraph is built as a view, we create a method getView () in BarGraph for BarActivity to get the updated view and plot the new data. The main part of BarGraph class is just some AChartEngine methods in BarGraph. In addition, we create two addNewPoints method in Bargraph for adding new data to graph objects.

LineActivity and LineGraph classes are just the same as BarActivity and Bargraph except that they are for line graph.

MockData and Point classes are used for getting data from Bluetooth device. In BluetoothChat class, when it detects the MESSAGE READ message, it will read the data and use the setSignal method in MockData class to send the data to variables in MockData class. The MockData class would use getDataFromReceiver method to create two Point objects. In Point class's constructor, it has two parameters: X and Y. We can call getX() and getY() methods to get the first and second data.

2. Mbed Program

The mbed program can be further divided into three parts: sensor data sampling and processing, result calculating, and RN-42 transmitting.

a) Sensor data sampling and processing

The analog data from the sensors are sampled using mbed's analogIn pins. Sampling rate is 200 Hz. Each data point is stored as a 16 bit floating point number, ranging from 0 to 3.3.

The digital filter is designed to be an FIR low pass filter, with cutoff frequency of 5Hz. The filter coefficients are generated by Matlab with its filter design GUI '*fdatool*'.

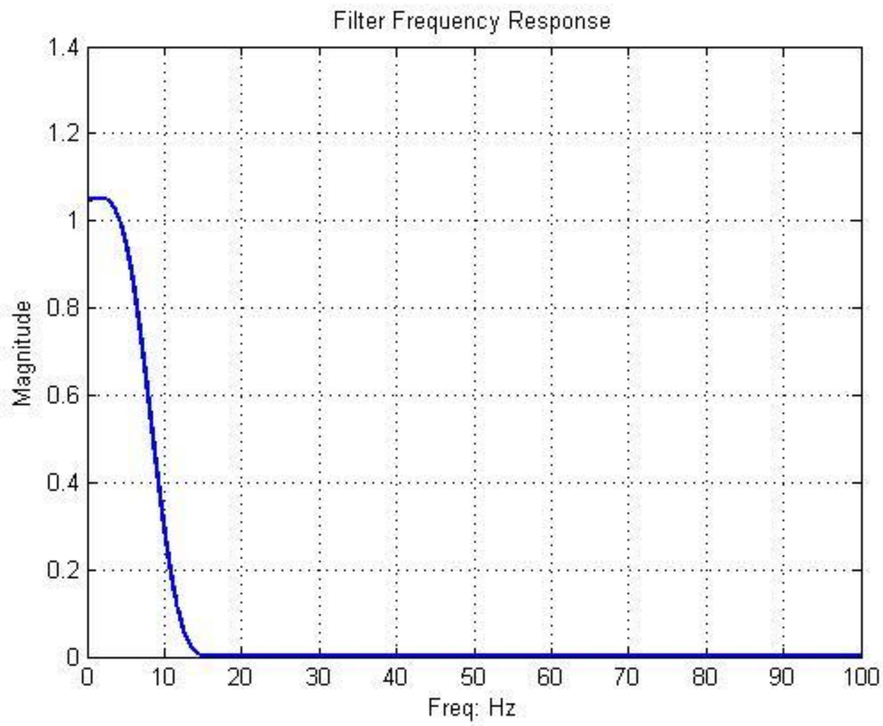


Figure 2. Filter Frequency response

The filtering code utilizes the DSP library provided by mbed. For every 32 data points, the *fir.process* function filters with given coefficients set by *FIR_f32<> fir()* function. The filtered data is then stored in a 64-byte buffer. In addition, a four-point based averaging of the raw data is applied to pre-process the data before the FIR filter. This method is put here to minimize any higher frequency noise and additional random fluctuations.

The result calculation requires two parameters to be extracted from the data stream: peak values and the DC values. For DC values, a windowed averaging is applied. The average of the data stream is computed every 192 data points (~0.96s).

For peak values, a peak detection algorithm is implemented. The algorithm defines an S function to be calculated for each data points, and then determine whether to keep the data points as a peak candidate or not based on its S value, the mean and standard deviation of the data stream. The S function is calculated as below: [3]

$$S_2(k,i,x_i,T) = \frac{\frac{(x_i - x_{i-1} + x_i - x_{i-2} + \ldots + x_i - x_{i-k})}{k} + \frac{(x_i - x_{i+1} + x_i - x_{i+2} + \ldots + x_i - x_{i+k})}{k}}{2}$$

An algorithm flow chart is below:

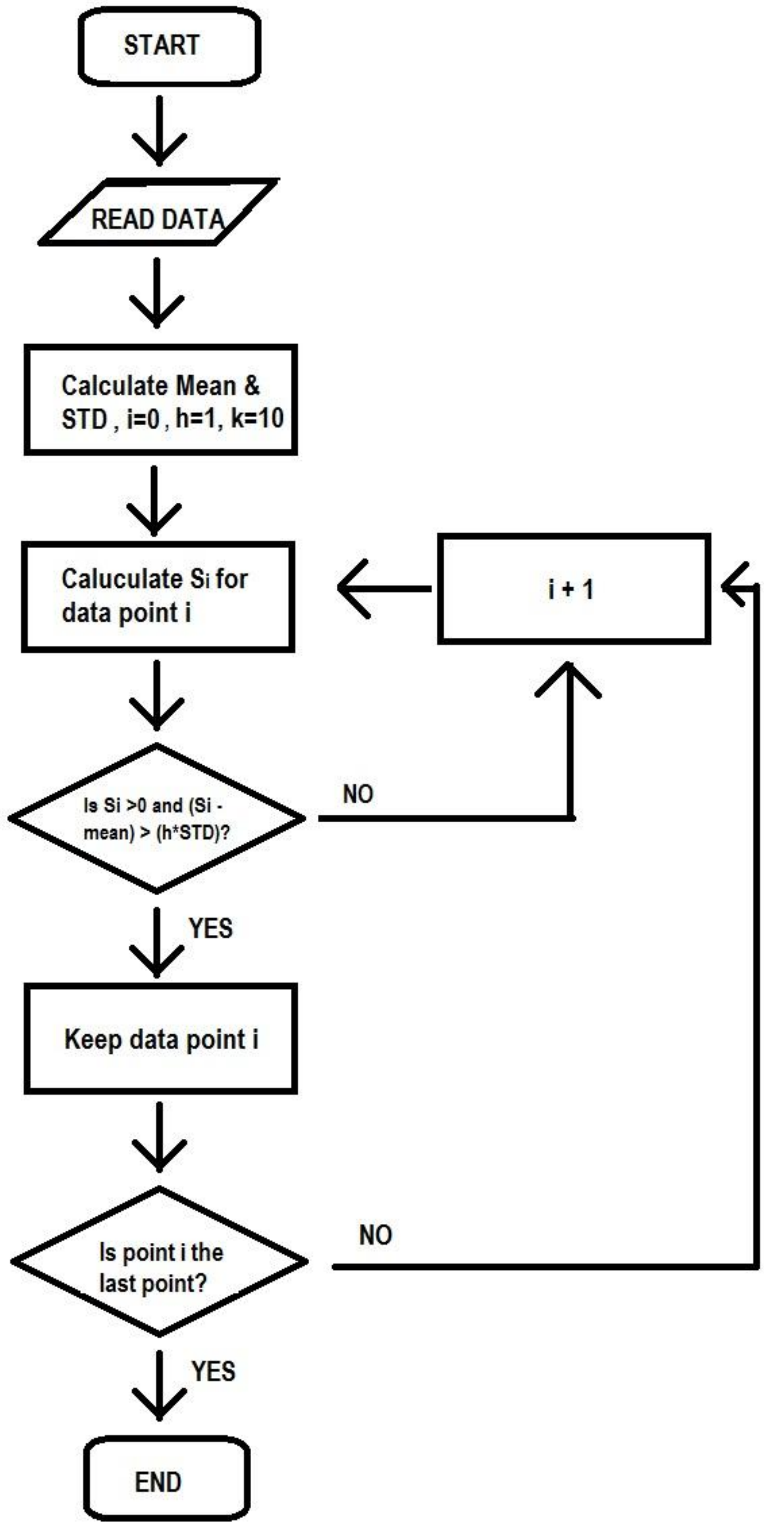


Figure 3: Peak detection algorithm

For most of time, for every local peak, there will be several points near the real peaks are kept. An additional step of taking the maximum of every k data points window will leave us with the real peak points.

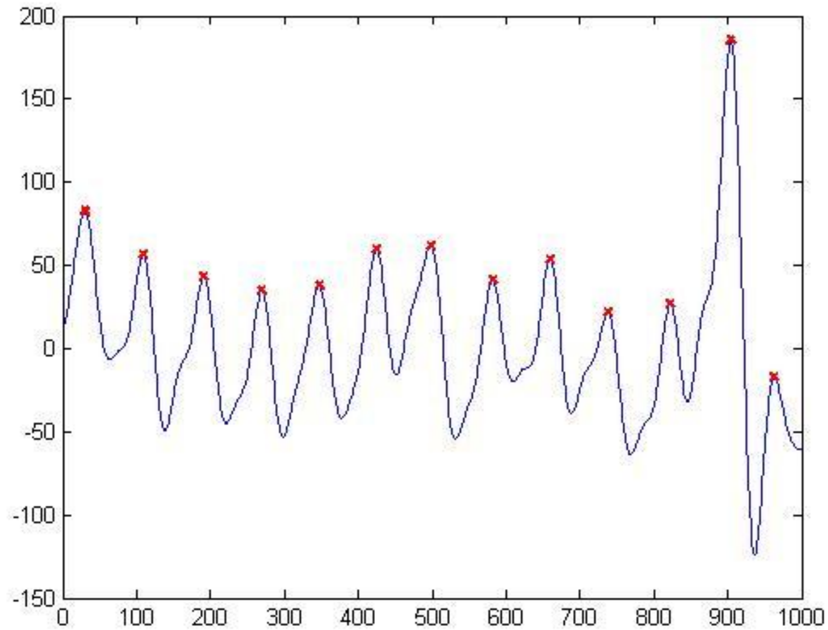


Figure 4: Peak Detection Result Sample

b) Result calculating

The Time interval for calculating is the same as the DC average window. For every 192 filtered data, the number of peaks is divided by the time interval to get the heart rate. The oxygenation value is calculated by the level of peaks and DC values.

Due the different optical responses of photodiodes towards red and infrared light, a calibration factor must be introduced to compensate this effect. This factor can be determined using the curve below [4]:

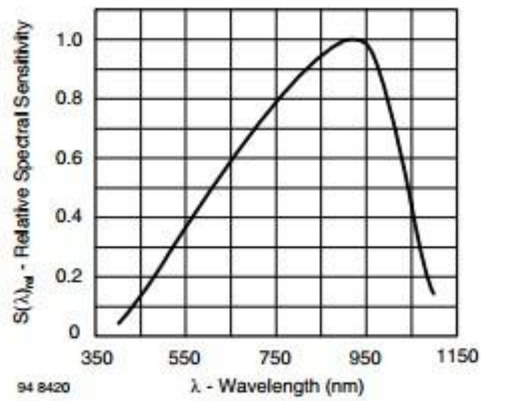


Fig. 6 - Relative Spectral Sensitivity vs. Wavelength

c) Bluetooth transmitting

The mbed provides a convenient way to interface with the Bluetooth module RN-42. Use *Serial()* to define RN-42 and its baud rate. The data can be sent via the tx pin using *printf().*

Architecture:

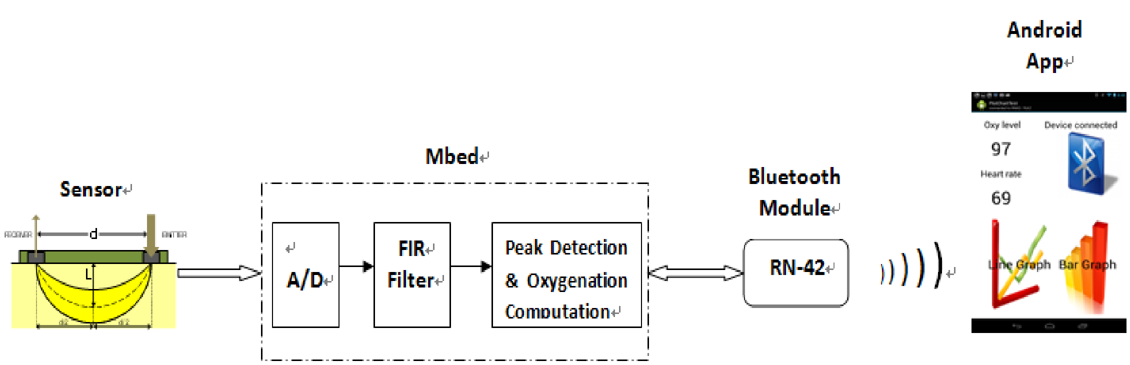


Figure 5: The block diagram of the system

Hardware description:

1. Optical sensor:

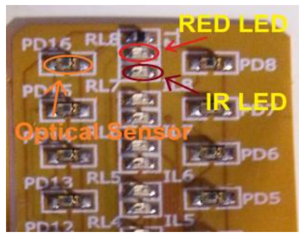


Figure 6: The Optical Sensors

2. Mbed:

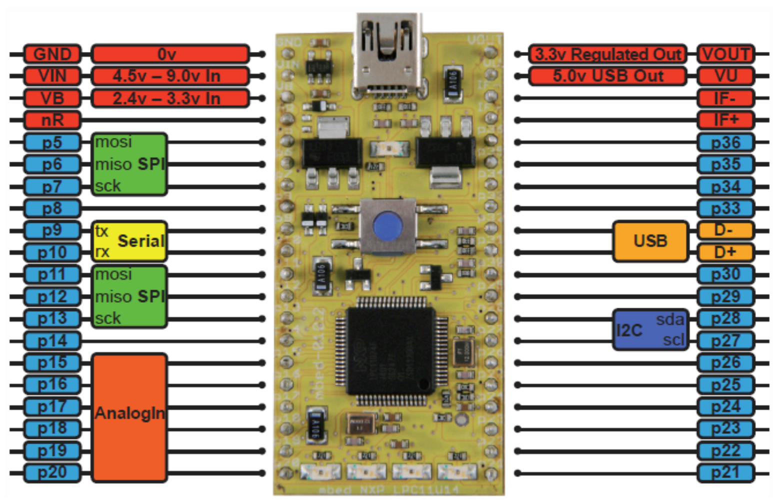


Figure 7: Pinout of Mbed

3. RN-42:

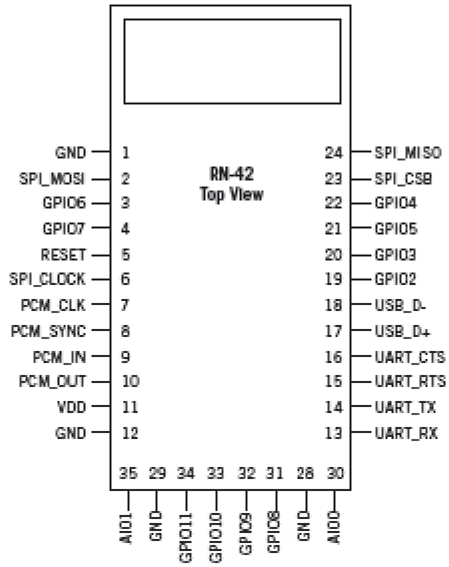


Figure 8: Pinout of RN-42 Bluetooth module

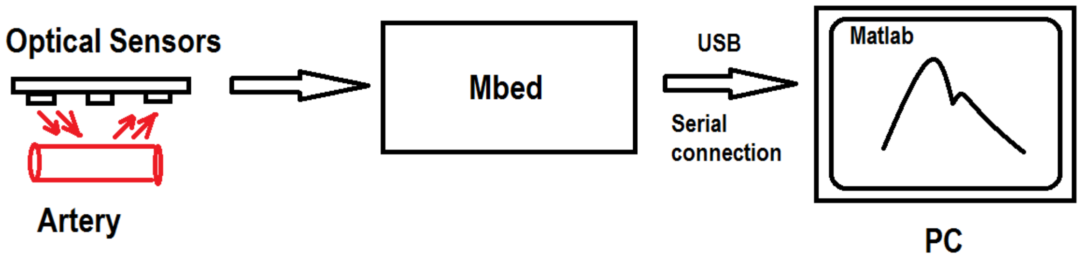
7 System Source Code

Android APP: ~**/Android/OxygenationMonitor/**

Mbed program: **~/mbed/mbed_DSP/**

8 Test and Evaluation System

A. The following setup is used to test the sensor-mbed interface:



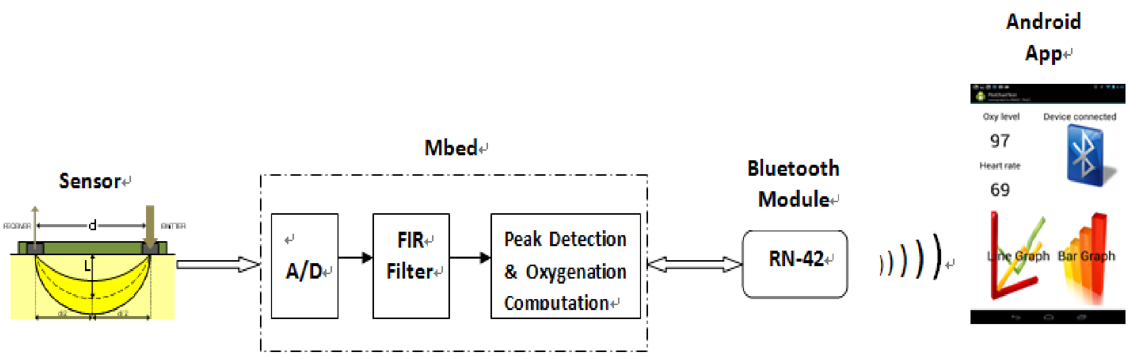
- Mbed sampling program: Drives sensor with digital output, samples sensor data at 200Hz, outputs data as 16bit unsigned numbers to PC via serial port.

- Basic two stage filter implemented in Matlab: 1st stage: Butterworth low pass filter, 2nd stage: Chebychev band pass filter. Pass band: [0.5Hz-10Hz]

- Data logging and plotting: Matlab real-time plotting script, written by Zi Huang and Yan Qu.

- Sampling subjects and location: Zi Huang, Yan Qu. Index fingers, thumbs, wrists.

B. The system below is for testing final functionality:



- Mbed program: Final designed mbed program with data processing and result output.

- Sampling subjects and location: Zi Huang, Yan Qu, Xiang Li, Index fingers, thumbs.

9 Test System Source Code

Android APP: ~**/Android/OxygenationMonitor/**

Mbed program: **~/mbed/mbed_DSP/**

Matlab scripts: **~/matlab_DSP/**

10 Test Results

Final functionality test:

The test results below are of a time period of 2 min, with subjects' fingers hold still on the sensor array. The sensors used were PD8 and RL/IL 5-8.

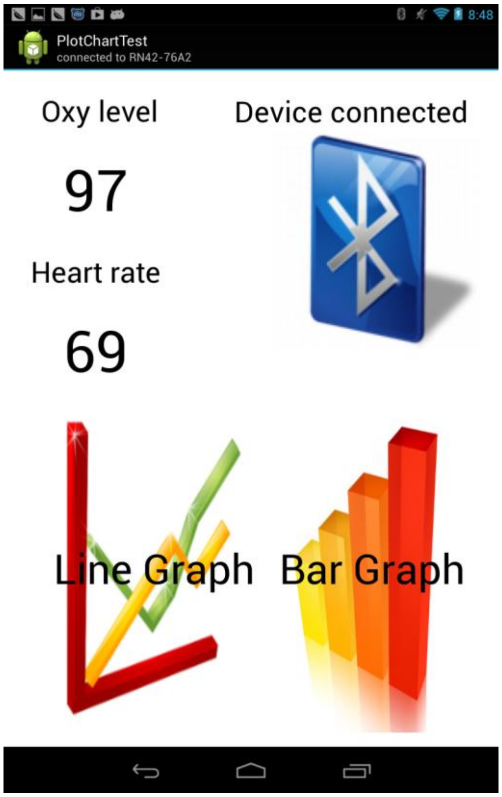
Subject	Location	Oxygenation	Heart rate
Zi Huang	Index finger	96~97	70~78
Yan Qu	Thumb	98~99	72~79
Xiang Li	Thumb	96~98	66~68
Teng Huang	Index finger	96~98	65~70

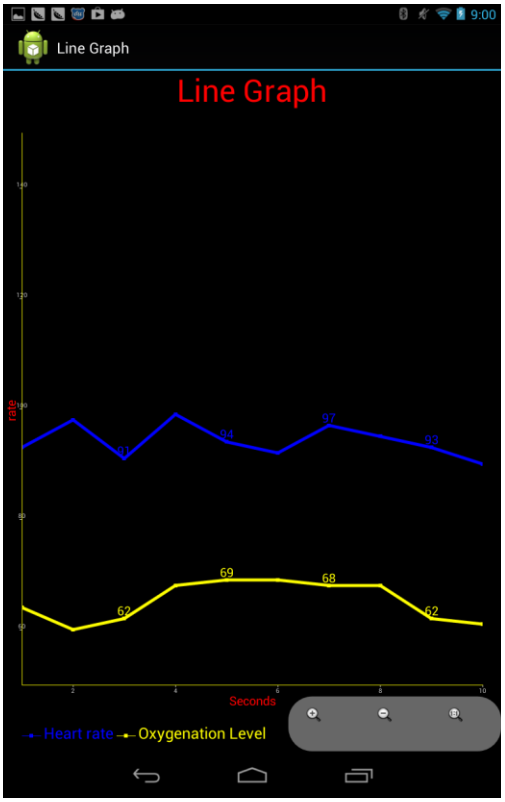
The test results demonstrated the satisfactory functionality of our system of data sampling, processing and transmitting. The real-time plotting feature of our Android application provides graphical presentation of oxygenation level and heart rate, and the mbed-Bluetooth interface prove to have sufficient speed to perform the task real-time.

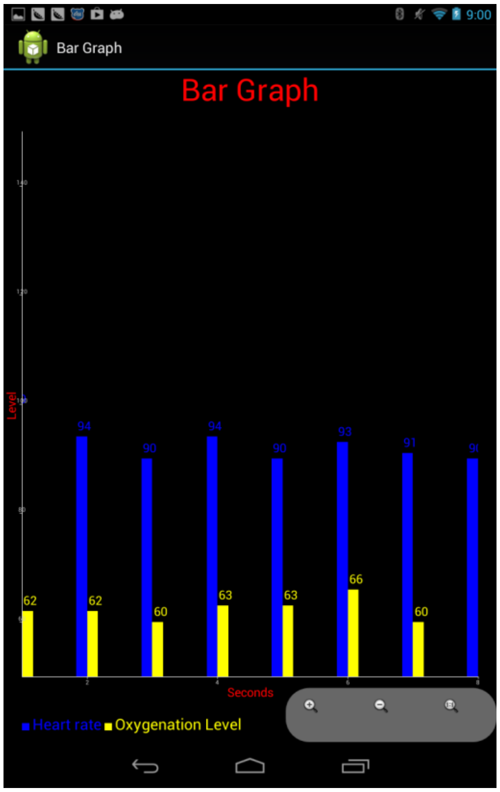
The results from the test fall into the expected range of normal health people. However, to further verify our system's accuracy, we require professional medical devices which will need extracted blood to measure oxygenation level.

Another aspect that our test didn't cover is other various locations throughout the human body other than finger tips. Other locations will require different calibration factor that will also require an accurate reading of the subject's oxygenation.

Below are a few screen capture of the Android application displaying the output:







11 Analysis

Our system has the ability to perform our main task: blood oxygenation and heart rate measurement. The efficiency is satisfactory. However, the verification of our measurements is somehow questionable. This is due to the fact that we lack of professional medical supply that will provide us accurate results for us to compare with our system's output. This issue must be addressed in the future.

On the other hand, the calibration process of our system is still undetermined. Other than accounting for photodiode's optical responses, the calibration also needs address subject-to-subject variance, temperature variance, and variance due to different body locations. To achieve this, again we need an accurate comparison.

Therefore, even though we accomplished our goal to realize satisfying functionality, the accuracy of our measurements is still on some level in doubt, which will definitely requires more future work.

Other Future improvements:

1) Robustness: more tests are needed to make the device adapt to different environment.

2) Power analysis: the aspect of power has not been considered in the prototype, the way to measure power needs to be established to achieve the low power goal.

3) Algorithm optimization: the further optimization of the algorithm could increase the data rate and lower the power consumption.

4) Multiple channels: multiple sensors can be applied to measure data at different locations and send multiple channel measurement results to the app. It also helps to calibrate the data to make it more accurate.

5) Verification: the results need to be compared with that of the mature product in the market to further modify the system.

12 Shared Systems

1. Mbed sampling/DSP program
2. Matlab test script/generated filters
3. Android application
4. All test data and related documents

13 References

[1] "Mbed Microcontrollers." - Handbook. N.p., n.d. Web. 08 Feb. 2013.

[2] Mapar, Bijan D. *Wearable Sensor for Continuously Vigilant Blood Perfusion and Oxygenation Monitoring*. Thesis. UNIVERSITY OF CALIFORNIA Los Angeles, 2012.

[3] Simple Algorithms for Peak Detection in Time-Series In Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence (2009) by Girish K. Palshikar

[4] "TEMD7000X01." Silicon PIN Photodiode. Vishay Semiconductors, n.d. Web. <http://www.vishay.com/docs/81951/temd7000.pdf>.