Design of an integrated mobile system to measure blood pressure

Conference Paper · November 2011

DOI: 10.1109/SCVT.2011.6101306

Ashraf Tahat
Princess Sumaya University for Technology

Design of an Integrated Mobile System to Measure Blood Pressure

Ashraf Tahat, Anwar Sacca, Yazeed Kheetan

Communications Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
tahat@psut.edu.jo

***Abstract*— High blood pressure (BP) (hypertension) is a leading chronic condition in the globe and a major risk factor for severe diseases. However, the measurement and management platform can still be improved. In this paper, we describe a complete low-cost prototype system that we have developed for this purpose. Our BP telemedicine device is based on the oscillometric method for measuring BP. A microcontroller oversees measurement operations, to process acquired readings, and to calculate the heart-rate. A smart mobile phone commands the operation of our developed system via Bluetooth. A custom mobile application software was designed and developed for this purpose and to receive these vital measurements in a convenient manner. Collected measurements can be saved (accumulated) on the smart phone or transmitted through a cellular network.**

***Keywords*- personal area networks; pervasive computing; blood pressure; telemedicine; bluetooth**

I. Introduction

According to a World Health Organization (WHO) estimate, raised blood pressure (BP) is estimated to cause 7.1 million deaths, about 13% of the global total [1]. High BP (Hypertension) is the most common chronic medical problem prompting visits to primary health care providers. Almost 1 billion people have BP worldwide, and more than a half a billion more will acquire this silent killer by 2025 [2]. Uncontrolled hypertension is an ongoing health care challenge worldwide. To effectively diagnose and manage hypertension, proper measurement, management, and interpretation of BP is essential.

Telemedicine is widely considered to be part of the inevitable future of the modern practice of medicine. It is gaining more and more momentum as a new approach for patients' surveillance outside of hospitals (at home) to encourage public safety, facilitate early diagnosis, treatment, and for increased convenience. Telemedicine is currently being used by doctors, hospitals, and other health care providers around the world [3]. The mobile phone has been recognized as a possible tool for telemedicine since it became commercially available [4]. Currently, there are 5.3 billion mobile phone users in the world, and it is estimated that 72% of them are in developing countries [5]. Moreover, newer cellular access technologies, such as GPRS, EDGE, 3G, WiMAX, LTE, provide for much higher data transmission speeds (rates) than the basic 2G cellular systems, which offers future telemedicine solutions endless choices for high-end designs.

The proposed mobile BP (monitoring) system is shown in Fig. 1. The patient (client) and the health care professional (server) can be located anywhere in the globe where there is wireless (cellular) network coverage. The patient's BP and heart-rate can be acquired by the patient himself, a family member or a health care professional can assist and participate in this process in more serious cases.

The signal acquisition process is performed by wrapping the pneumatic system (cuff) around the patient's arm at a designated place as is normally done in a typical similar set-up. The client unit communicates with the mobile phone through a Bluetooth transceiver. The mobile phone application software can initiate the measurement process when instructed by the user via software interface menu. When BP and heart-rate measurements are ready, they are displayed on the smartphone's screen and in- turn stored or transmitted to a desired remote destination.

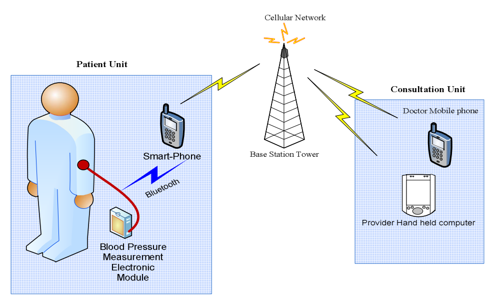


Figure 1. Mobile telemedicine measurement and monitoring system

II. The Blood Pressure Measurement System

Our device deploys the oscillometric BP measurement method. This method of measuring BP is very often used for the measurement of the BP because of its excellent reliability. The oscillometric method is based on observing the magnitude of oscillations caused by the blood as it begins to flow into the arm, again, after it had been occluded during the maximum inflation of the cuff**.** When the pressure decreases in the cuff, pulsations will begin to be emitted by the artery; the pressure then reported on the device defines the maximal BP or *Systolic blood pressure* (SBP). During the pressure decrease in the cuff, the oscillations will become increasingly significant, until maximum amplitude of these oscillations defines the average BP or mean arterial pressure (MAP). Then, the oscillations can still be observed while the pressure

978-1-4577-1289-0/11/$26.00 ©2011 IEEE

is decreasing in the cuff, until they disappear; the pressure then read on the device defines the minimal BP or *Diastolic blood pressure* (DBP).

In our design, the BP monitoring device measures the blood pressure using a microcontroller (PIC16F877A), a pressure sensor MPX5050GP, and a Bluetooth module (LinkMatik 2.0). The microcontroller controls a valve and an air pump. These were integrated on custom-made printed circuit board (PCB) as shown in Fig. 2. Also, Fig. 3 is a block diagram that depicts interconnection of the device's components. When a measure command is received via Bluetooth from the mobile phone interface, the microcontroller will close the valve and trigger the air pump to inflate the cuff to a certain predefined high pressure value (180 mmHg). Then the cuff will start to deflate slowly through the bleeding valve. The microcontroller will also acquire the sensor's measurement through its analog-to-digital converter (A/D) module. It will also calculate the BP and heart-rate based on these measurements. When readings are ready, the microcontroller will open the valve, send the acquired readings through its UART (universal asynchronous receiver transmitter) to the Bluetooth module, and in-turn to the controlling mobile phone.

A. Air Pump and Valve Control

An output signal from the microcontroller should close the valve, and another signal should be able to turn on the air pump. The solenoid valve is a normally opened. In order to close the valve and trap the air inside the air chamber (cuff), the microcontroller should send a high signal to the valve. This can be done directly because the microcontroller's output voltage is sufficient to close the valve.

The process of interfacing the air pump to the microcontroller is a little more complicated than that of the valve. That is because the air pump is an inductive load (has a motor in it), thus it draws large amount of current; this implies that the microcontroller's output voltage/current will not be sufficient enough to turn on the motor. A relay was used to turn on the motor.

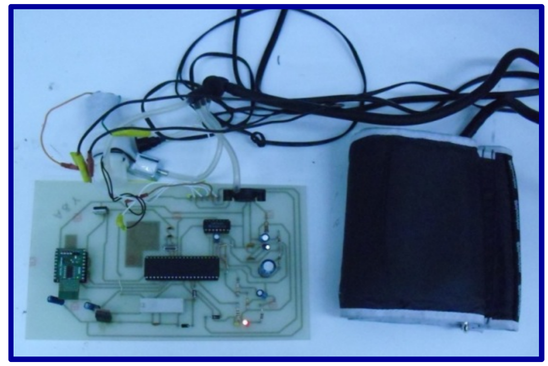


Figure 2. PCB of our blood pressure monitor device

B. Analog Pressure Sensor Output Filtering

The pressure sensor MPX5050GP [6], which is an integrated pressure sensor, has trimmed outputs, built-in temperature compensation and an amplified single-ended output which make it compatible with ADC of our micro-controller. This sensor has a built-in on chip differential amplifier that level shifts and amplifies the very small differential voltage produced by the transducer bridge, translating this voltage to a single-ended output voltage ranging from 0.2 to 4.7 volts.

The two main types of noise in a piezo-resistive pressure sensor are white and $1/f$ noise, which are shot noise and flicker noise, respectively. Noise can also come from external circuits. Hence, appropriate grounding of power supply and PCB layout is essential and needs special consideration. Although the transducer has a response of about 500 Hz, its noise output extends from 500 Hz to 1 MHz. To overcome this problem we used decoupling capacitor for the sensor's power supply input and hardware lowpass filter (LPF). This LPF has a cut-off frequency of approximately 650 Hz, thus omitting the white noise occurring at higher frequencies.

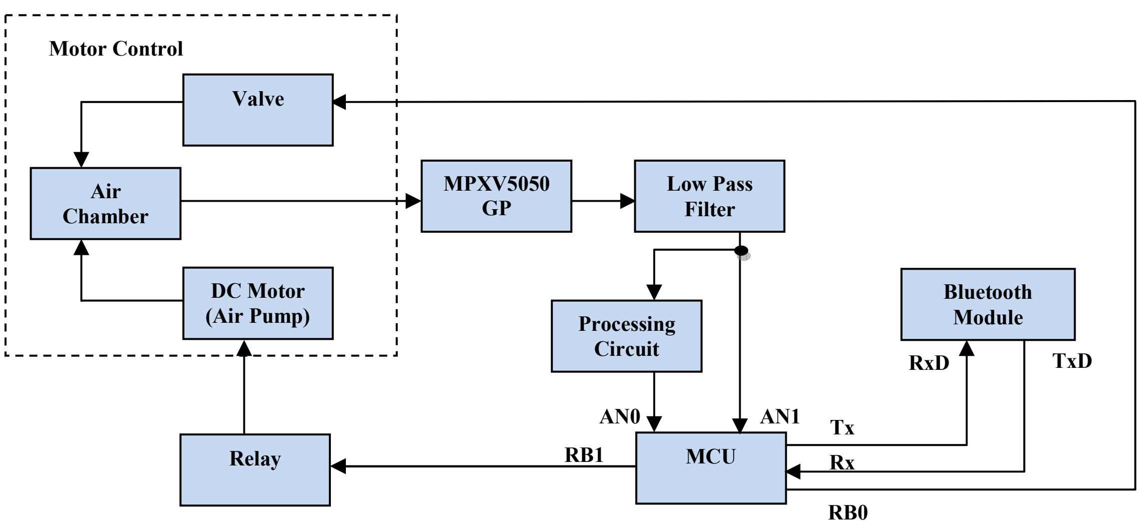


Figure 3. A block diagram of our blood pressure monitor device

C. *Processing Circuit and Interfacing to Microcontroller*

After removing the white noise riding on the sensors output, the LPF output, which we will refer to as the *sensors output* for simplicity, was split into two. That was done for two different purposes: The first, which we will refer to as the raw output, is used as the cuff pressure. The second, which we will refer to as *processed output*, is further processed by a circuit that will be explained below.

The cuff-pressure can be directly interfaced with an A/D converter for digitization since MPXV5050GP is signal-conditioned. So the raw output which will be used to measure the cuff-pressure is connected to the PIC16F877A microcontroller analog input channel-0 (AN0). The raw output provides a voltage ranging from 0.2 to 4.7 volts, which is internally converted by the microcontroller to an 8-bit unsigned integer.

The other path will filter and amplify the raw cuff pressure (CP) signal to extract an amplified version of the CP oscillations (referred to as the processed output). Those oscillations are produced by the expansion of the subject's arm, during cardiac systole, each time pressure in the arm increases. The output of the sensor consists of two signal components. These are: The oscillation signal component (≈ 1 Hz) riding on the CP signal component (≤ 0.04 Hz). Therefore, a high pass filter is utilized to block the CP signal before the amplification of the oscillation signal. The CP signal has to be appropriately attenuated so that the baseline of the oscillation will be constant, and the amplitude of each oscillation will have the same reference for comparison.

The oscillation signal varies from one person to another, where it varies from less than 1 mmHg to 3 mmHg in general. From the transfer function of MPXV5050GP, this will translate to a voltage output signal of 12 mV to 36 mV. Then the oscillation signal becomes 3.8 mV to 11.4 mV, respectively, because of the attenuation introduced by the filter which is 10 dB to the 1 Hz signal. As was indicated by the manufacturer application note [7], the amplification factor of the amplifier is chosen to be 150 for best results, so that the amplified oscillation signal is within the output limit of the amplifier (5.0 mV to 3.5 V) as shown in Fig. 4.

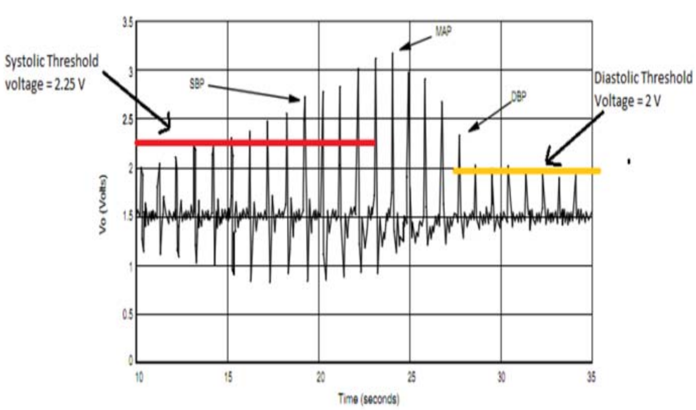


Figure 4. Illustration of threshold voltages on an extracted oscillation signal

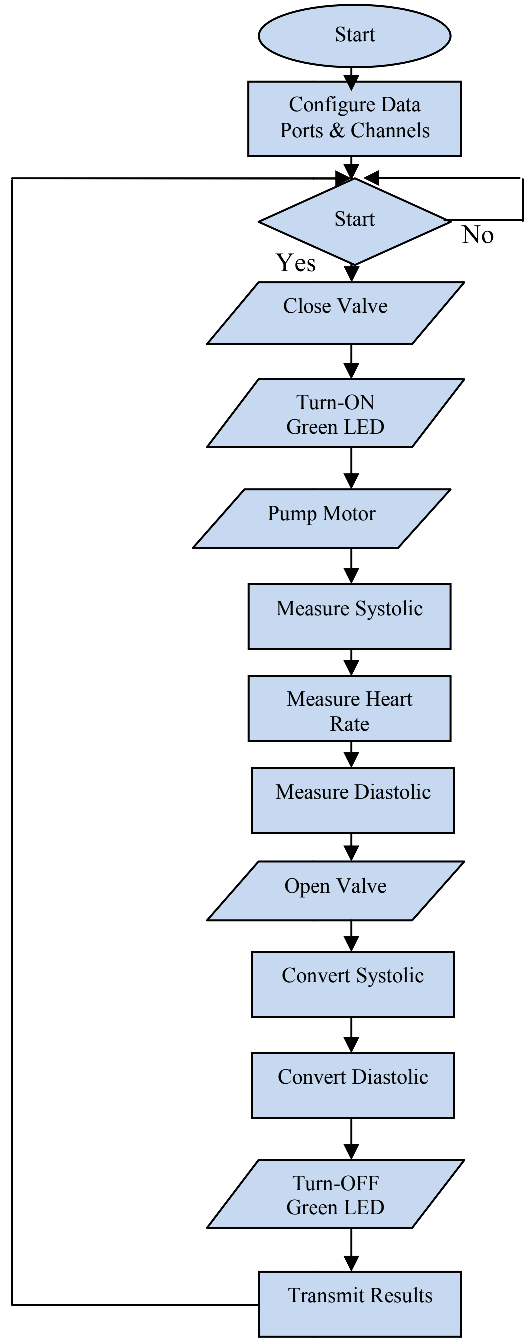


Figure 5. Microcontroller program flowchart

In order for the PIC16F977A to observe and make-use of the oscillation signal in the measurement process, the output of this circuit (processed output) was connected to its A/D channel-1 (AN1). The DC value recorded then will be the value that represents the diastolic pressure.

D. *Heart-Rate Measurement Method*

In this mode, the device will measure the heart-rate of the subject. The device is designed to enter this mode right after the program finishes calculating the SBP and before calculating the DBP. Fig. 5 depicts a flow chart of the microcontroller program. We choose to determine the pulse rate right after determining the SBP because at this point the oscillation of the waveform is strongest. The main idea here is that we want to measure the period of the oscillation (1 pulse). To accomplish this; the program samples the AC waveform every 40 millisecond. It then records the time interval between the times at which the values of the AC waveform cross the voltage value of 1.75 volts. The threshold level of valid pulse is set to be 1.75 volts to eliminate noise or spike, and to take the oscillation at its beginning. Immediately after identification of the amplitude of the pulse, the microcontroller neglects the signal for 450 ms to circumvent any false identification due to possible existence of premature pulse "overshoot". Therefore, we are able to detect a pulse rate which is less than 133 beats per minute (60/0.45 = 133) only with this algorithm. The time delay is chosen to be 450 ms. This is equal to half the average period of a heart pulse for a normal healthy person, which is typically 0.9 seconds. To increase the accuracy of the heart-rate reading, the program takes the average of five time intervals.

E. Bluetooth Tranceiver

The microcontroller is programmed to send the measurements on its UART to the Bluetooth module. Bluetooth provides a means to connect devices such as mobile phones over a secure, globally unlicensed short-range radio frequency (2.45 GHz) and to enable the exchange of information between them. We used the LinkMatik 2.0 Bluetooth transceiver module, which is a class 1 (20 dBm) model that has an approximate range of 100 meters. The asynchronous data from/to the PIC16F877 microcontroller is delivered to/from the LinkMatik Bluetooth module on the serial port at a speed of 9600 bps. The Bluetooth module is configured as a *Slave* and the mobile phone is considered to be functioning as a *Master*. The microcontroller sends/receives data to/from the Bluetooth module, which transmits/receives data continuously as raw binary bytes.

III. MOBILE APPLICATION

A. Application Software Development

Our main objective here was to develop a mobile application that is capable of communicating with our BP monitoring device remotely over Bluetooth technology. We wanted to achieve two main ideas in our application: the first was to build a mobile application that has the ability to remotely start the BP measurement operation on the device. While the second was to be able to received data from the BP monitor device, process it and display it in a convenient manner on the screen.

To accomplish both tasks, we decided to build a Windows Mobile application that can be run on any Microsoft Windows CE device. The basic requirement for our application is simply a mobile device or pocket PC that runs Windows Mobile 6 or later, as its operating system. Our development tool was Microsoft Visual Studio 2005 with service pack-1. In addition, to develop windows mobile application in Microsoft Visual Studio, the Windows Mobile 6 SDK (software development kit) had to be installed. In building our application, we also used the .Net compact framework version 3.5 accompanied by the Windows Embedded Source Tools for Bluetooth Technology, from MSDN shared resources to control the Bluetooth radio in the mobile.

We decided to adopt C# as our programming language for developing our mobile application,. It is worth mentioning that our software application can still run on earlier versions of Windows CE such as Windows Mobile 5.0 under one condition which is installing the required service pack and .Net compact framework version 3.5 or later.

In our mobile application, we chose communication using Bluetooth via the serial port profile for various reasons. To start with, the microcontroller in our device communicates with the Bluetooth module serially using UART serial communication port. This implies that our Bluetooth module supports the serial profile. In addition, Windows Mobile has a nice feature which is COM Port emulation; this enables virtual COM ports to be created over radio frequency channels. In other words, once Bluetooth is pair to another Bluetooth device, you can map it to a virtual COM port and communicate with it using the serial port profile.

B. Application Software Operation

When the application software is first run, the mobile device will not be connected to the BP monitoring device, thus the connection status will be set to "disconnect". Also, it will check the status of the Bluetooth radio to see whether it is *turned-on* or *turned-off* and display the correct status. The user has to select a proper COM port to establish the connection as shown in Fig. 6. The program will then check the status of the Bluetooth radio. If the Bluetooth radio is *turned-off*, it will be *turned-on* automatically in order to establish the connection. Moreover, there will be a separate thread from the main program thread that will be running in the background, which will continuously monitor and check for changes in the status of the Bluetooth radio and connection status. This thread will be able to detect changes made by the program or by the user external to the application software, from the mobile device operating system. If any change has been detected, this thread will invoke another function on the main program thread to make the corresponding updates on the user interface elements. For instance, if Bluetooth was *turned-on* from the mobile itself, the Bluetooth status will change from *OFF* to *ON*, and the color of this label will change from red to green, and a notification message will appear on the screen.

Now once the program is running and a proper connection has been established, the user will be capable of measuring the BP and heart-rate using the BP monitor device. This would be accomplished by selecting the "*Measure Blood Pressure*" button under *Action* menu. A progress bar will be shown on the screen indicating operation progress as shown in Fig. 7. The user will not be able to use any function on the main application unless the measurement process has ended. This is a form that shows that the BP measuring process is in progress. It will show up once the measurement process is started. It contains a picture box that contains an image of a dummy ECG signal, and the picture box is animated to move from left to right, from one edge of the form to the other.

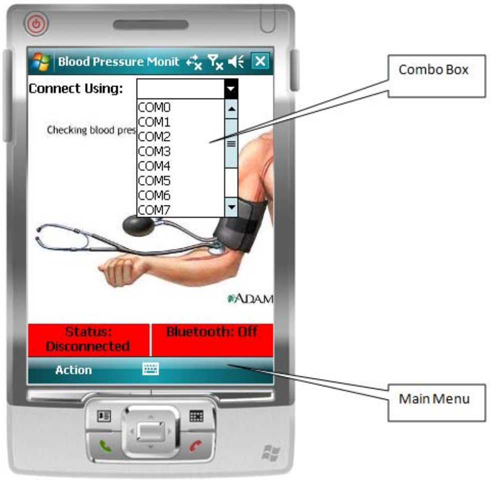


Figure 6. Selecting COM port in mobile application

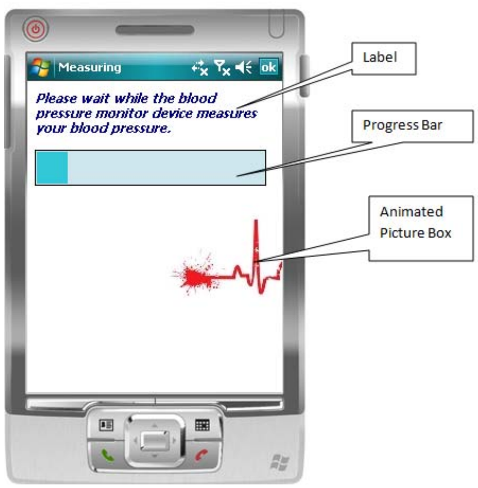


Figure 7. Measuring form

It will automatically disappear once the measurement process has ended and data is received from the BP device by the mobile. A measurement panel is used to display the results (readings) received from the BP monitor device. This panel will show up in the application immediately displaying the results once data is received as shown in Fig. 8. The panel consists of several items as follows: a) the SBP and DBP readings in mmHg. b) the heart-rate reading in pulses/min. c) picture box that shows a dummy heart image. d) a *close-form* button that will allow the user to close the panel and return back to main form.

IV. Analysis of Measurements and Results

In order to verify the reliability and accuracy of the readings obtained using our BP monitoring device, we compared them to other readings obtained by a commercially available device that has already been calibrated and tested for reliability. We chose the Visomat handy BP monitor found in the market as our reference. We compared results obtained by our device to results using the Visomat, assuming that the Visomat readings are nominal and error free.

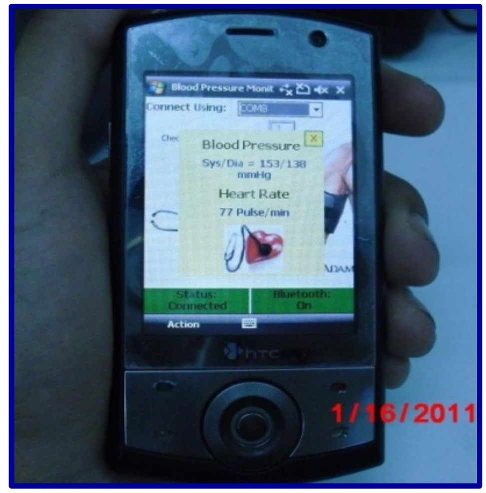
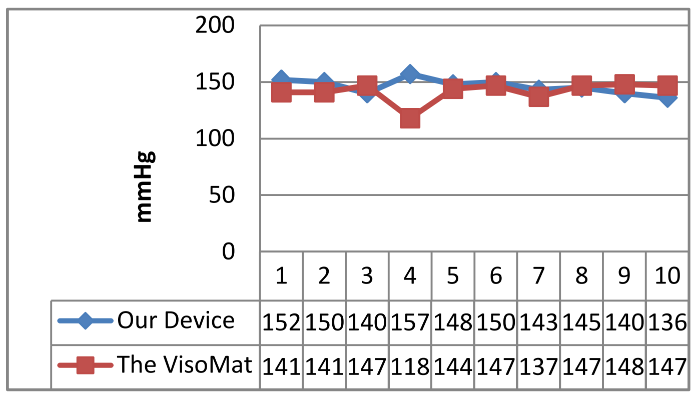


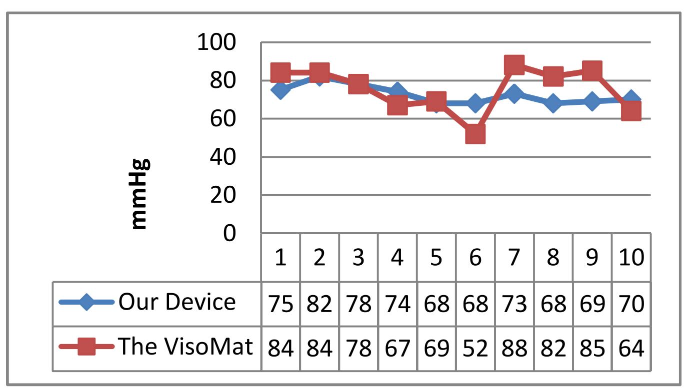
Figure 8. Mobile phone displaying blood pressure and heart-rate measurements.

We fixed all variables in our testing environment so that the only changing variable is the measuring device.
Our statistical analysis was done by taking several readings of BP for a human subject using our device and the Visomat device.



	1	2	3	4	5	6	7	8	9	10
Our Device	152	150	140	157	148	150	143	145	140	136
The VisoMat	141	141	147	118	144	147	137	147	148	147

Figure 9. SBP comparison chart



	1	2	3	4	5	6	7	8	9	10
Our Device	75	82	78	74	68	68	73	68	69	70
The VisoMat	84	84	78	67	69	52	88	82	85	64

Figure 10. DBP comparison chart

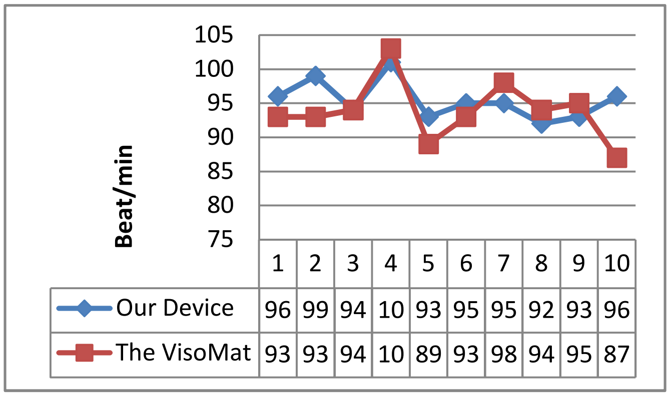


Figure 11. Heart-rate comparison chart

The measurement values taken are listed below the corresponding graph of measurements from both devices for DBP, SBP, and Heart-rate in Fig. 9, Fig. 10, and Fig. 11, respectively.

To further analyze the accuracy of our device, we calculated the mean value of each reading obtained by the Visomat device and our device, and then compare the two values to calculate the error. The mean was calculated by the weighted average method. Table I illustrates the results of analysis, which proves the effectiveness of our versatile BP measurement system.

TABLE I. MEAN VALUES AND ERRORS

	Our Device (mean)	**Visomat (Mean)**	**Absolute Error**	**Percentage Error (%)**
Systolic	146.1	141.7	4.4	3.01
Diastolic	72.5	75.3	2.8	3.86
Heart Rate	95.4	93.9	1.5	1.6

V. CONCLUSION

We were able to implement a reliable low-budget telemedicine prototype of a wireless portable monitoring device that can measure blood pressure (BP) and heart-rate accurately using commercially available components.

The main aspect of our system design is the utilization of a smart- phone and its built-in components to control our device and to display measurement results. That has reduced overall hardware cost. A custom made mobile application software with practical interface was also developed to simplify user operation. This proof of concept has demonstrated its applicability to telemedicine applications. This monitoring system can be deployed in many settings such as hospitals, nursing homes, health centers, and homes for its versatility and convenience.

REFERENCES

[1] P. Puska, S. Mendis, D. Porter, "Chronic diseas – key risk factors," World Health Organization (WHO), 2003.

[2] P. Kanavos, Jan Östergren, High blood pressure and health policy: where we are and where we need to go next, Report*, 2007*

[3] Y. Choi, J. Krause, H. Seo, K. Kaptain, K. Chung, Telemedicine in the USA: standardization through information management and technical application, IEEE Communications. Magazine, vol. 44, pp. 41–48, 2006.

[4] A. Tahat, "Mobile messaging services-based personal electrocardiogram monitoring system," International Journal of Telemedicine and Applications, vol. 2009, pp. 1-7, June 2009.

[5] International Telecommunications Union (ITU), The World in 2010, http://www.itu.int/ITU-D/ict/statistics/ , February 2011.

[6] Freescale Semiconductor, Integrated Silicon pressure sensor (MPX5050) datasheet, Chandler, Arizona, USA, Mach 2010.

[7] Freescale Semiconductor, Digital Blood Pressure Meter (AN1571) application note, Chandler, Arizona, USA, May 2005.