International Journal of Science Engineering and Advance Technology

Design of an Optical Blood Pressure Sensor for Noninvasive monitoring of blood pressure

M.Raghunath[1],venneti.kiran[2],Srinubabu Aravapalli[3]
[1,2,3]Assistant Professor in Electronics and Communication Engineering
[1,2]Aditya College of Engg,Aditya Nagar,ADB Road,Surampalem
[3]Kakinada Institute of Technology and Science
[1]Raghu473@gmail.com,[2]venneti.kiran47@gmail.com,[3]srinubabu.ece@gmail.com

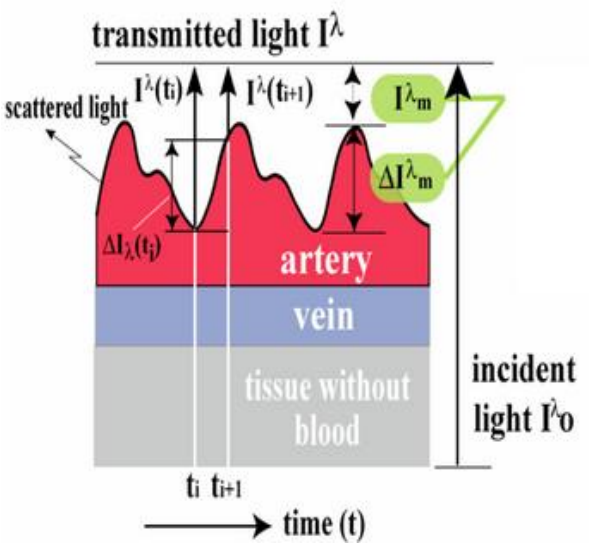
Abstract: **In modern society, patients often have access to a wealth of electronic data concerning their social networks and surroundings, but have little personalized insight into their own health. Our plan is to distribute a small handheld device with an array of biometric sensors to collect data about a user's body. With the patient's current and past data, we can give the patients a pre-diagnosis (outlining symptoms and possible sickness) using machine learning techniques to create a general summary of the patient's symptoms and potential treatment while collaborating with physicians to reach a final consensus of the patient's health status before the patient is aware of his/her illness. We want to empower users with basic knowledge of their vital signs and help them be more proactive in disease prevention while relying less on frequent primary care physician office visits.**

Keywords: Blood Pressure, AtMega 1284p, XBee, GUI, Pulse Time Transit.

I. INTRODUCTION:

BACKGROUND MATH

In order to measure blood pressure, most methods involve invasive intravenous (IV) force measurement or pressurized air cuffs. The IV methods are highly invasive and require skilled administration by a nurse, while pressurized air cuffs can be uncomfortable and inconvenient. We decided to pursue an optical pulsatile method of blood pressure measurement, using pulse transit time rather than direct force measurements. Similarly, to gather pulse data we utilized optical reflectivity. We can optically measure pulses because of the reflective characteristics of blood, as indicated in the figure below. Different volumes of blood have different reflective characteristics, so by measuring the changing in reflectivity we could measure the time between pulses in varying parts of the body. Our focus on optical measurements comes from our idea to reduce the level of invasiveness as far as possible. Compared to traditional methods, our optical measurement method is fast and simple enough to ideally be used by anyone with little training. In order to get pulsatile readings on two parts of the body simultaneously, we use a front and top facing sensor on the final device. The user positions his/her finger on the top sensor while pointing the front sensor to a different part of the body, thus measuring a difference in pulse time. As blood pumps through the body, the volume of blood in the points of measurement also changes. These volumetric changes modify the optical reflectivity of the measured area, since the composition of the surrounding tissue stays the same while the volume of blood changes. We can measure these volumetric changes and detect peaks in the signals to detect heart beats, and use the delay between peaks at different parts of the body to detect differences in pulse transit time.



The reason we chose to measure blood pressure and pulse were because of some preliminary machine learning statistics we gathered about heart disease prediction. Using the UCI dataset, we trained an SVM classifier to predict whether or not a patient had heart disease using only non-invasive data. We utilized a linear kernel, and our only features were

ISSN 2321-6905

age, sex, chest pain types, resting blood pressure, and maximum heart rate. Just utilizing these basic features, we were able to achieve a relatively high rate of accuracy in predicting heart disease, and so we wanted to build a prototype that could help us acquire the data necessary to begin data mining for the eventual purpose of diagnosing patients. Below we present the statistics gathered, where we show the obtained accuracies given number of training documents. By using just a few health metrics, we could obtain accuracies of >76% using purely these metrics alone.

We used the Moens-Korteweg model of blood pressure to justify the validity of our pulse transit time measurements to estimate blood pressure. The Moens-Korteweg model describes the relationship of PTT with blood pressure. Essentially, the PTT is directly proportional to the pulse wave velocity with linear offset based on the path length.

PWV = distance/PTT = sqrt (Eh/p2r)

E = elastic modulus of the blood vessel
h=vessel thickness
p=blood density
r = vessel radius

The elasticity modulus is further dependent upon the blood vessel wall pressure, related exponentially.

$$E=E0e^{aP}$$

In this equation a is a constant 0.017mmHg, while P is the mean arterial pressure. The elasticity modulus thus changes exponentially with changes in pressure. From this relationship, we can fit known PTT measurements and ground truth blood pressure measurements to a logarithmic model.

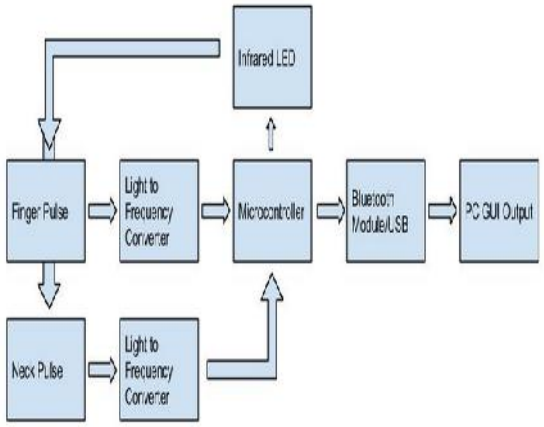
$$P = A\ln(PTT) + B$$

This can be done with standard regression techniques to find calibrated coefficients A and B. The model assumes the pulse transit time measures the time from the aortic valve of the heart opening to the pulse propagation to the measured area. However, for our model, we measured the difference between the propagation times to the neck and the finger. This approach has the advantage of using only pulse propagation methods and does not require additional instrumentation such as an electrocardiogram (ECG)

According to the Buenos Aires group, the PTT still follows the calibratable logarithmic model if certain features are fixed. The mean pulse wave velocity must be assumed to be similar for both pathways regardless of the different blood vessel thicknesses. As a result we have used this regression-based model to gauge the accuracy of the PTT measurements for this project.

II. Logical Structure

This diagram shows the overall structure of our system. At the heart of our device is the ATMEGA 1284p, this device is controlling the infrared LEDs that will transmit light which is reflected back to the light to frequency sensors. The microcontroller takes in the signals that reflect pulsatile measurements, and will digitally process these signals into pulse measurements, and send the data through either USB or Bluetooth to a PC. The PC will have a GUI that graphically shows the two pulses and the blood pressure and pulse.



There are two main components of our design: the base station unit and the mouse unit. The mouse unit contains the mouse controller, a mega1284P microcontroller, an ultrasonic sound transmitter, three ultrasonic sound receivers and an XBEE wireless transceiver. The base station contains a mega32 microcontroller and an XBEE wireless transceiver. It is connected to PC using a USB cable. Users input their commands using the mouse controller; the raw distance data will be generated in the mouse unit within the mega1284P microcontroller for it to output the x and y coordinate data. The microcontroller also handles the ISR generated by button click actions. The mouse unit finally encodes the data into packets and the data is then transmitted to the base station unit using XBEE radio transceiver. When the base station receives the data, it encodes and transmits the data to PC through serial connection. The serial data is read and decoded by java application and utilized for the java HID class. A diagram demonstrating the logical structure is shown as above.

III. Hardware/Software Tradeoffs

To begin our project, we had to make a few decisions between hardware and software. Because this sub-project was part of a larger theme, we wanted our project to be very modular such that we could add sub systems for new sensors as we

developed more ideas. In order to ensure a more modular product, we decided that most we wanted to limit the hardware as much as possible. Adding more and more hardware to the medical scanning device would only make our product more expensive, so we decided that having a microcontroller that could digitally process and receive signals would allow for more future modularity, as we could add smaller pieces of cheaper hardware and filter all signals through the microcontroller. The tradeoff is that the software becomes increasingly complex, and at some points we even found that the ATMEGA1284p did not have enough RAM to handle some floating point operations, so an even more expensive microcontroller would have to be obtained.

IV. Hardware Selection

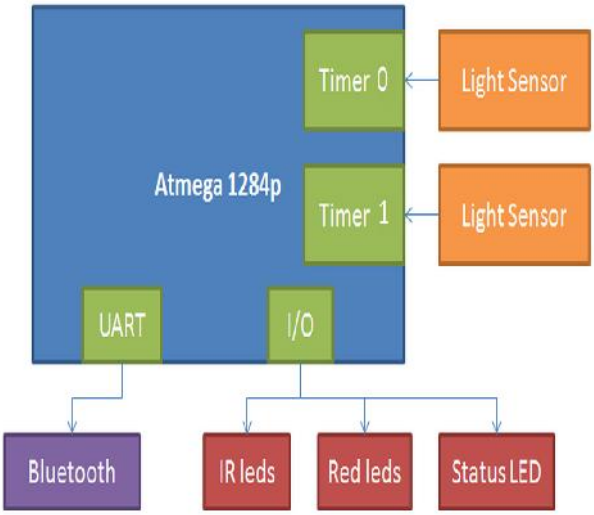
The TSL2561 digital light sensor communicated via I2C (or two wire interface TWI) and essentially combined a high resolution ADC with a high sensitivity analog photodiode. However, to achieve accurate measurements on the TSL2561, the sensor required long integration times to achieve a high enough signal to noise ratio. With our experimentation, we found this rate to be 50mS. Since the length of the pulse transit time averages around 200mS, this sampling rate would be too low temporal resolution to be practical.

We eventually chose the TSL235R, a light to frequency converter, because it allowed completely digital measurement of light. The typical square wave output from the sensor was several kilohertz, which was reasonably accurately detectable by our microcontroller. The advantage of the light-to-frequency sensor was its variable dynamic range. Since we ultimately used counters to measure the frequency, we could measure over a wide range of frequencies and light intensities by measuring timer overflows.

We also experimented with multiple microcontroller chips. In the interest of price, we initially attempted to use an Atmega328 chip. However, the Atmega328 only had three hardware timers, but our initial timer implementation of the frequency sensing required four timers. As a result, we experimented with polling methods of counting, but found that the approach resulted in high levels of jitter. Because we found that the polling method was unstable, we used the Atmega1284p in order to use the greater hardware capabilities and additional timers to build this project.

V. System Design

This is an overall view of our design project. Two light sensors connect to the ATMEGA1284p through pin B0 and B1, externally triggering timer 0 and timer 1. Our UART is connected through D0 and D1 to the Bluetooth module, although the project can also be configured to use USB to connect to a computer. Port A controls the various LEDs.



Optical Sensors

The TSL235R light to frequency converters used in this project were arranged on the front and top of the device, to allow for neck and finger pulse sensing. The optical sensors are placed in between two LEDs, one red 660nm and one 980nm IR led. Also included is a toggle switch for power and a momentary push button switch to signal data acquisition.

Bluetooth Module

The HC-06 Bluetooth module is mounted underneath the main board and is connected to UART0 on the microcontroller. This module is paired to a PC that has Bluetooth connectivity via a USB dongle.

Atmega 1284p

The Atmega microcontroller is the brain of the device. All of the sensors and LEDs are broken out and connected to the microcontroller, which is connected to a white board. The unit is still on battery power, and thus the entire device is still wireless. We used the board provided to us for this class for this project.

Physical Housing

The physical housing was designed in Solidworks and converted to an .STL file to be printed on a 3D printer. The housing includes ports for the light sensors and associated leds, as well as ports for a on/off toggle power switch and a momentary push button for user input. The ports are

designed so a plastic window is mounted on top, so the user has a place to rest his/her finger on top to reduce movement during measurement. Later revisions of the housing will more properly shroud the sensors from ambient light by better enclosing the sensing areas. There was also an issue with cross-talk between sensors within the housing, so thicker isolation between light sources and light sensors will increase information gain.

VI. Software

Overview

The main tasks of our program included acquiring data from the light sensors, controlling the output of the LEDs, processing the acquired data and extracting measurements in real time, and transmitting the measurements to a PC for visualization with a GUI.

Digital Optical Data Acquisition

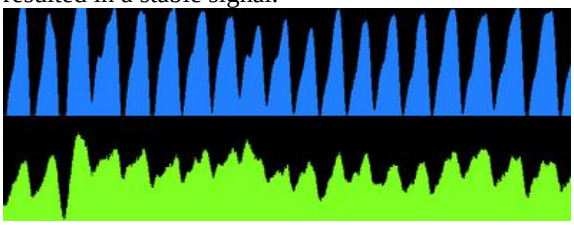
As described in the hardware selection section, we experimented with several light sensors for this project. The analog based I2C sensor required integration to stabilize the light reading to have a high enough signal to noise ratio to be useful. However, the minimum integration time for this was 50mS, which was too long since we required a high sampling rate for temporal accuracy. With the light to frequency converter, optical data acquisition methods always had a tradeoff between hardware requirements and software time constraints. In order to have a fast sampling rate with low signal jitter, we needed more hardware timers and a fast clock speed in order to process data accordingly. However, since we were constrained by the hardware on the At mega 1284p, we had to make tradeoffs to using software solutions.

Our final optical data acquisition method used a mixture of externally triggered counters and internal timer interrupts in order to sample the number of pulses from the light sensors in a given amount of time. We used a total of four timers. Timer 2 and timer 3 were used for timekeeping. Timer 2 was set with a clock divider of 64 with an output compare register with value 249, in order to trigger an interrupt every 16000 cycles, or 1mS. This became the time base for the system. Timer 3 was set as a precise 10mS timer, and since it is a 16-bit counter it can count to a higher value with minimal jitter. Timer 0 and 1 were both externally triggered by the light to frequency sensors. However, timer 0 is only 8-bit so the anticipated counts of approximately 300 per sample exceed the limit without overflow. As a result we added an ISR for the overflow of timer 0 to increment a variable which is later used to determine the final count of timer 0 after the sampling period.

Digital Signal Processing

Though our data acquisition methods were designed to reduce jitter and noise, the signals were still unstable. Since the data acquired was proportional to the total measured light, we needed to add a degree of high pass filtering in order to extract the AC values. In order to do this we created a moving average filter of 20 samples. To extract the AC components we then subtracted the original signal from the mean of the 20 previous samples in the moving average filter.

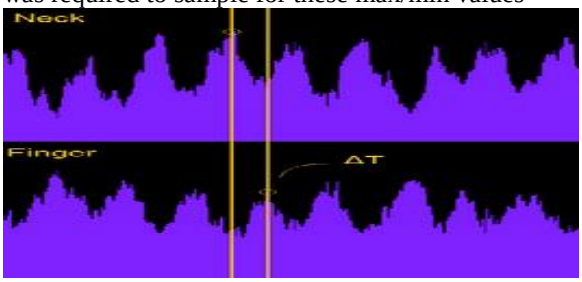
In addition, we needed to add a low pass element to the filtered data in order to smooth some of the high frequency jitter. To do this we additionally added a small 5 element low pass filter and extracted the mean of the samples. This effectively reduced the variability due to jitter and resulted in a stable signal.



Filtered signal Acquisition.

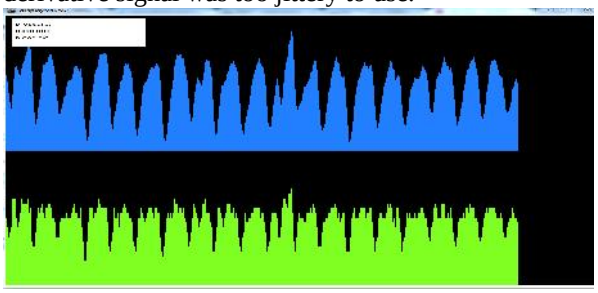
Peak Detection

In order to implement a reliable method of peak detection for this project, we went through several strategies of peak detection. Initially, we aimed to use thresholding to determine peaks over a certain size. We logged the max and minimum values sampled within a time period, then counted the entire pulse signal as a peak until the signal dropped below the minimum threshold. This approach had several disadvantages. By thresholding, slight movements by the patient could easily result to false peak detection. The max and min values sampled were difficult to extract especially if the range was constantly changing. Another downside was that additional time was required to sample for these max/min values



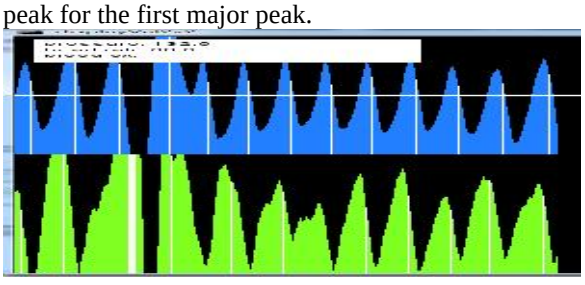
Jitter from polling method

We also experimented with taking the derivative of the signal to extract peak information with zero crossings. The derivative was taken by subtracting the signal with its previous sampled value. Ideally, when the derivative would be near zero, it would signify either a maxima or minima. We could then check the signage of the signal to specify a maxima or minima. However we did not use this method because the difference between samples was too great as our sampling rate was not high enough. The result of our 10mS sampling period was that the derivative signal was too jittery to use.



Derivative method of peak detection.

Ultimately, we were able to reliably detect peaks in a signal by finding the first local maxima in a positive pulse. The sequence waits in the first state if the signal is negative or zero, and changes state when the signal becomes positive. In this state, we check if the signal is increasing or decreasing. When the signal begins decreasing, we have detected the first local maxima, and a flag is set. In order to make sure the signal was not due to jitter, we increment a counter. When the counter is incremented a threshold amount of times and the values continue to decrease, the state changes to set another flag that a valid peak has been detected. This flag is reset when the signal returns to the original state when the signal drops below zero. We have found that this method works reliably with the optical signals, as it only counts a peak for the first major peak.



Final Data capture

Bluetooth

For this project we used Bluetooth to communicate between our microcontroller and a PC application we created to display the collected data. We used the HC-06 Bluetooth to UART converter to transmit and receive from the microcontroller. A standard USB Bluetooth dongle was used to provide connectivity on the PC side.

Graphical User Interface

To visualize our collected data and aid with debugging, we created a simple GUI to use with the unit. The GUI was helpful to visually represent our peak detection and filtering methods, and also provides a convenient method to validate if the data collection is working. To implement the GUI, we used a visual Java IDE called Processing to quickly build the real time display. The program simply updates on each valid reception from the Bluetooth connection, delimited by a newline character. We transmit multiple variables indicating sensor readings and detected peaks for pulse transit time. We made a predefined packet structure and delimited between different transmitted variables with a space character. The transmitted variables are then displayed on the GUI, including the measured pulse transit time and measured heart rate.

VII. Results

Overview

With our maximum sampling rate we obtained 5% accuracy with our pulse transit time. Due to the limitations of the microcontroller, we utilized a scheme through which we capture frequencies from the light to frequency converter that samples every 10ms, while pulse transit times are on the order of 200ms, thus our accuracy for pulse transit times and pulses are on the order of 5%. The filtering of the signal is very good, as we have a very clean signal, but we could obtain a higher accuracy if we obtained frequencies more accurately. With a higher clock speed we could create a control scheme that captures the frequencies at a faster sampling rate. Since we continually capture the data and process it, there is no problem with execution speeds and delays.

The pulse transit time data is also highly personalized, which is useful for our final machine learning goal. Variability amongst different people result in different associated blood pressures for each pulse transit time. As a result, a model must be made based on different body types and demographics in order to approximate true blood pressure based on pulse transit time. Since we have not yet built a large enough dataset to achieve this, we were unable to analyze the PTT data to fully predict blood pressure.

The device is fairly simple to use, as you only need to cover one part with your finger and hold the device to your neck to gather all readings. There was an issue with interference with Bluetooth requiring too much current from the battery and

ISSN 2321-6905

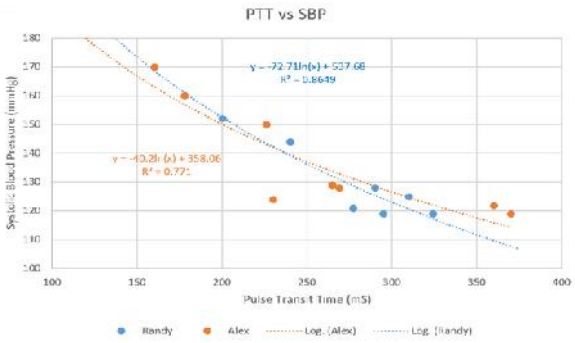
reducing power to the LEDs. However, that issue was fixed with stabilizing capacitors and fresh batteries. The range of the Bluetooth was acceptable for our given operating conditions. We tested the range of the Bluetooth to around 20ft, and the signals were still valid. However, the intended usage of this device should always stay within several feet of the Bluetooth receiver, so further testing was unnecessary.

VIII. Accuracy of Model

To extract useful blood pressure information from the PTT, we created a regression model to predict blood pressure from pulse transit time. We used a standard cuff sphygmomanometer to measure systolic blood pressure (SBP). Since we did not have a stethoscope, we logged the systolic blood pressure as the pressure where the first peak from blood flow was visible. Diastolic blood pressure was too difficult to measure accurately without a stethoscope, and will be a topic of further study not included in the scope of this project. In order to remove outlier measurements due to movement, we took several blood pressure measurements at each reading and took the median value of the measurements as the valid reading. The same method was done for the pulse transit time, where measurements were taken until a series of precise measurements were taken. We then took the median of those measurements as the valid data point. This was necessary because of occasional invalid readings due to movement artifacts. Since our design requires the user to hold the device in one hand while positioning the two sensor heads, movement artifacts would be unavoidable.

The data was collected from a variety of conditions. The conditions include sitting, lying down, and sitting/lying down after various degrees of physical activity. As a result, we were able to collect data for a wider range of blood pressures with two test subjects. We used ourselves as the main test subjects for data collection, and created two models relating PTT and SBP. The extracted models are shown below:

The reason why this mode of blood pressure measurement requires some baseline measurement is that there are inherent differences in body types and measurement methods between people. For example, differences in arm lengths and body proportions can increase the path length and thus rescale pulse transit time accordingly. Factors such as height, weight, and BMI can influence the final measurements. Though we plan to use these factors in a later revision of the device to select a model,



for this project we used the ground truth blood pressure with the sphygmomanometer to test the accuracy of the PTT measurements. Nevertheless, the different models correspond to each subject's body type. In our collected data, we noticed that the models were not identical, though they were very similar. This is because both test subjects have very similar body types, heights, and lifestyles. These results suggest that the model could be used to extract blood pressure from PTT for other subjects with similar body types.

With these models, we can gauge the accuracy of our system. Since we run with 10mS sampling increments, we can use the model to find that the resulting accuracy of the systolic blood pressure is around 1.2mmHg/10mS sampling. This is fairly acceptable for gauging blood pressure, though it may not be as exact as different methods. This uncertainty error could always be decreased by increasing the sampling rate.

IX. References:

1. Darovic, Gloria Oblouk. Hemodynamic Monitoring: Invasive and Non-Invasive Clinical Application, 2nd edition. Philadelphia: W.B. Saunders Company, 1995.

2. Dean L. Franklin, Techniques for measurement of blood flow through intact vessels, Med. Electron. Biol. Engg. 3, 27-37 (1965).

3. P.R. Hoskins, A review of the measurement of blood flow velocity and related quantities using Doppler ultrasound, Journal of Engineering in Medicine, 213, 391-400 (1999).

4. Peter Vennemann, Ralph Lindken, Jerry Westerweel, In vivo whole field blood velocity measurement techniques, Exp. Fluids, 42, 495-511 (2007).

5. M.D.Stern, "In Vivo evaluation of microcirculation by coherent light scattering", Nature 254, pp. 56-58, 1975.

6. H.Hong and M.Fox, "Noninvasive detection of cardiovascular pulsations by optical Doppler techniques", Journal of Biomedical Optics, No 4, pp.382-390, 1997.