11-2013

Physiological sensor.

Alan M. Jimenez
University of Louisville

PHYSIOLOGICAL SENSOR

By

Alan M. Jimenez
B.S., University of Louisville, 2012

A Thesis
Submitted to the Faculty of the
University of Louisville
J. B. Speed School of Engineering
as Partial Fulfillment of the Requirements
for the Professional Degree

MASTER OF ENGINEERING

Department of Electrical and Computer Engineering

November 2013

PHYSIOLOGICAL SENSOR

Submitted by: _____

Alan Jimenez

A Thesis Approved on

_____

(Date)

by the Following Reading and Examination Committee:

_____

Dr. Karla Conn Welch, Thesis Director

_____

Dr. John F. Naber

_____

Dr. Moises A. Carreon

Acknowledgements

This thesis would not have been possible to complete without the advice and guidance of my advisor, Dr. Karla Conn Welch, who instructed me through the whole process and always found time to answer my emails. Dr. Welch was always accessible when I needed help and was constantly providing valuable feedback. She kept me on track every step of the way, and for this, I am grateful. I would like to thank my family for their support and encouragement. My parents are deserving of recognition for instilling a great deal of respect for education within me from a young age. They taught me how to overcome adversity and block the voices of those who told me that I wasn't capable of getting an engineering degree. My mother constantly reminded me of the importance of getting an education and she made sure that I was exerting myself in my educational endeavors. In short, they are the best parents that a student can ask for. Finally, I would like to thank my mentors and friends – Anand Kulkarni and Nasser Sadaatzi – for their motivation and help through the whole process – you have become very good friends during the year and I will surely miss the fun times in the research lab.

Abstract

For the purpose of medical diagnosis and research, various physiological signals, such as Photoplethysmograph (PPG), Galvanic Skin Response (GSR) and Skin Temperature (SKT), are measured by different types of medical sensor equipment. However, the sensors are intrusive and the patients must endure some discomfort when encountering these types of medical sensor equipment. These bio sensors also fail to be implemented outside of lab or clinical settings. Recently the development in sensor technology and wireless communication technology have significantly improved the integration of wearable systems, so that we could find new ways to minimize the wearable circuits module, design layers of fabric for wearable system. This thesis documents the successful development of a novel, unobtrusive, low-cost, wrist-worn integrated sensors (PPG, GSR, SKT) system using wireless wearable technology capable of measuring real-time data collection, and monitoring which is important when dealing with treatment and management of many chronic illnesses, neurological disorders, and mental health issues. Examples can include: epileptic seizures, autism spectrum disorder (ASD), depression, drug addiction, and anxiety disorders.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The collection of physiological signals collected can aid in the process of medical diagnosis. However, many times the sensors used to collect physiological data are expensive and immobile with patients enduring some discomfort because of the intrusiveness of the sensors. To address this problem this thesis documents the design, development, and testing of a suite of physiological sensors; namely, Photoplethysmograph (PPG), Galvanic Skin Response (GSR), and Skin Temperature (SKT), that are low cost, low power, and non-intrusive to be embedded on a wristband. These wearable sensors will be capable of long-term physiological monitoring, which is important when dealing with the treatment and management of many chronic illnesses, neurological disorders, and mental health issues. Examples include: epileptic seizures, autism spectrum disorders (ASD), depression, drug addiction and anxiety disorders.

A low cost, non-obtrusive, low energy alternative to the collection of three informative signals, including wireless communication and real-time monitoring is feasible. With the exploitation of cutting-edge technology this device can only get smaller (i.e. with advancements of integrated circuits and surface mount devices). The low cost, non-intrusive sensors could impact medical applications - adding mobility, affordability and convenience. Monitoring these signals in a non-obtrusive way over a long –term period can open the previously unavailable opportunity

believe that research cannot monitor these signals in a non-clinical setting (e.g., at home, in a classroom). This sensor was designed under the working objective of monitoring teenagers with an ASD during hour-long session in classroom settings.

Intelligent wearable sensor systems are being sought after by the industry and public health-care system as well as institutional affective computing research. This partly due to the rapid medical development of our modern health-care system, the system is undergoing a great change, particularly in the United States. Affective computing is one of the emerging fields in health care industry. Affective computing is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects; it is an interdisciplinary field spanning computer sciences, psychology, and cognitive science [1]. Today, affective computing has already transcended in a growing number of applications in the healthcare industry. This outpour of progressiveness has been motivated by over a decade of research from neuroscience, cognitive science, and the arts about how human health, decision making, and behavior is influenced by emotions. The latest scientific findings show that emotions play an important role in decision making, perception, and learning. This relationship means that emotion influence the mechanisms of rational thinking. There is still no definitive agreement between scientists on the true definition of what are emotions, but many scientists do agree that its main characteristics can be described as arousal (calm or excited) and valence (negative or positive) [2].

The physiological states of arousal is achieved by changes within the autonomic nervous system (ANS) functioning. ANS is part of the nervous system that controls your involuntary actions [3]. There are differences between the sympathetic and parasympathetic nervous system even though they are sometimes mentioned together and illustrated together on diagrams. Functionality speaking, they are quite different anatomically and physiologically. They sometimes behave independently and sometimes cooperative in their functions. A way to describe their functions is that the parasympathetic produces responses to promote restoration and conservation of energy while the sympathetic nervous system is concerned with making the body ready to deal with external threats or dangers. That being said, increased activity in the sympathetic subdivision elevates heart rate, blood pressure, and sweating, as well as sending blood toward skeletal muscles, lungs, heart, and brain to help your body's motor action [3]. The ability to measure change in arousal with sensors such as the (PPG), (GSR), and (SKT) are essential for continuous monitoring ANS responses and, as a result, have the ability to assist in early detection of physiological disorders. Increased understanding of these responses will revolutionize the field of affective computing by providing the tools needed in developing physiological and behavioral measures to classify symptoms of emotional states such as autism spectrum disorder (ASD), depression, anxiety disorders, and personality disorders [4].

Motivation

Affective computing has a growing number of applications in healthcare. This emerging field has truly changed the definition of health-care and the way it is being delivered. Affective computing has made researchers in high demand of ambulatory monitoring of physiological signals to enable medical interventions and studies outside of the clinical setting [5]. The spread of personal computers (PCs) and the establishment of the *World Wide Web* have led to a revolution of the way humanity is communicating and sharing information. Today people can communicate their thoughts and ideas through social networks such as *Facebook* and *Twitter.* As new technology emerge, it also influences the lives of everyday people, from ordering a pizza online to watching a movie through cloud service. Yet, in the last decade there is an even a bigger technological innovation that is happening at the personal level. Computer technology has opened the doors to advances of miniature sensors, as well as mobile and ubiquitous computing; as a result it has garnered the high interest of wearable technology. Today, mobile devices such as smartphones and tablets are replacing PCs. As a result, the industry and public healthcare system as well as institutional affective research are seeing a fast increase of intelligent wearable sensor systems [5]. The low-cost and highly deployable technologies today could have significant impact on making physiological signal monitoring more widely available.

The high interest of these wearable systems arises from the technology that makes this possible as well as the interest of research and medical monitoring for the need of monitoring patients over extensive periods of time. Many researchers as well as physicians have a need to monitor individuals through personalized therapies, where the treatment can adapt to a person's individual affective state. This observation would need to be done with sensors that have the ability to measure changes within the physiological states of arousal and at the same time be unobtrusive, low cost, and comfortable enough for continuous wear.

The combination of ambulatory monitoring with physiological signals can potentially be used for a number of research applications. An important new direction for research in monitoring affective (i.e., emotional) states is the identification and development of assistive technology that can help optimize user experience. Such systems can be advisory to a wide population; specifically for individuals with autism, who exhibit deficits in emotion comprehension of themselves and others, physiology-based affect-sensitive systems could be notably informative for interpreting emotional cues [6]. The first aim of this thesis is to develop and test wearable physiological sensors for use in affect-sensitive feedback systems. This aim includes building sensors to transmit data on galvanic skin response (GSR), photoplethysmogram (PPG), and skin temperature (SKT). This thesis will contribute the analysis of affective information gathered from physiological data during Human Robot Interaction (HRI) to monitor the emotional states of individuals (age 13-21) with and without autism and help them cope with

stressful situations. The use of wearable bio-signal sensors to measure anxiety for the communication and characterization of emotion in children with autism, is the compelling application motivating the work of this thesis. Children with a diagnosis of autism spectrum disorders (ASD); get stressed but often do not express these feelings outwardly (e.g., in words, through typical facial expressions, or typical body gestures). Instead, they are often described as having unexpected meltdowns, which can result in aggression towards others and sometimes self-injury [7]. As a result of the unpredictable flurry of emotions, teachers and caregivers have a difficult time preventing meltdowns. Sometimes children with ASD are overwhelmed by the interpersonal interactions and can become frustrated and stressed. This limited expression of emotion adds hurdles to communication and can make traditional learning challenging. It has been researched that children with ASD have a better naturally independent connection with sensory based learning [8].

The wearable ambulatory sensor network technology can be used in numerous research applications to detect and record physiological signals capable of detecting signs of stress and excitement by measuring electrical changes in the skin (GSR), measure the heart rate (i.e., pulse measured in beats per minute, PPG), and skin temperature (SKT). Researchers have agreed that these signals are among ones with greatest potential for minimally-invasive data collection, and although typically used on the fingertips these signals could be feasibly incorporated into a wrist band for collecting data during a range of activities. The

physiological sensor system discussed in this thesis is a research tool. While the bio-sensor system has the potential to assist in therapeutic research to children with ASD, its effectiveness has yet to be proven. This thesis seeks to answer the question whether the developed in-house physiological sensor system is a viable research tool to analyze affect-sensitive interactions between an intelligent system (e.g., a robot's on-board computer) and a person by using psychophysiological techniques.

Present Day Scenario

Traditional biosensors primarily originate from medical-originated sensors. Commercially-available products, such as the FlexComp biofeedback system by Thought Technology and the MP150 system from Biopac Systems, are available for clinical uses and research purposes. Those traditional devices for monitoring physiological signals biosensors are capable of measuring many physiological signals, but are limited to clinical settings.

Mobile monitoring devices are also available, some of them being the BodyMedia SenseWear (Figure 1.2) Armband and Affectiva Q Sensor (Figure 1.3, discontinued around summer 2013). Both devices monitor galvanic skin response (GSR), body temperature, and acceleration. These sensors are costly and they lack an important physiological measure for affective computing, PPG. Conclusions from data collected from the SenseWear Armband are focused on calorie

calculations following long periods of activity, not predictions of emotional states on a short temporal scale. Another example of wearable sensor is the traditional clipped sensor that can be used when working out. This sensor can measure your heart rate but again it can be costly limited to exercise summaries. Wireless transmission of data and analysis in real-time also remain as challenges to physiological monitoring systems.
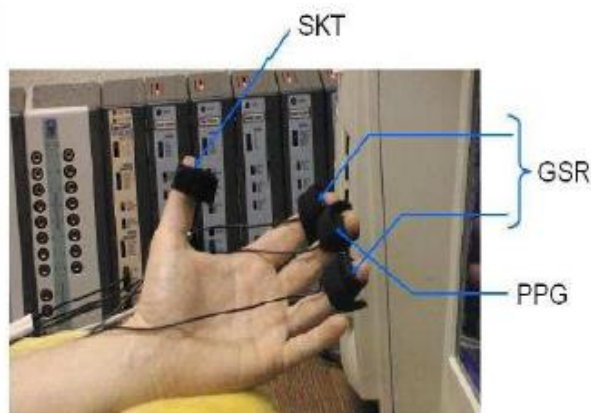
Figure 1.1: Biopac system (background) with PPG, GSR, and SKT sensors

Figure 1.2: BodyMedia SenseWear Armband

Figure 1.3: Affectiva Q Sensor

Objectives

The objectives of this thesis are:

- To develop a suite of a wearable physiological sensors for affective responses from physiological signals, namely PPG, GSR, and SKT.

- To design the sensors with ability to measure, store, and transmit physiological parameters using low-power wireless Bluetooth communication.

- To validate this in-house sensor with commercial PPG,GSR, and SKT sensors from Biopac [9] as per the comparison experiment in Poh et al.'s work [10].

The eventual application of this thesis is:

- To conduct testing of sensors involving children with ASD

Chapter 2

Literature Review

This chapter contains background information required to understand the motivation behind the research topic of this thesis. This chapter begins with a discussion about various sensor paradigms used to get the body signals from humans, which are called biosensors. Understanding of human cognitive and physical states is a basis for affective control of human-machine systems. The challenges involved with the development of sensors that are designed to be nonintrusive, long-term and measured in real-time are also discussed in this chapter. As previously mentioned, unobtrusive, low-cost ambulatory physiological sensors capable of monitoring affective responses has the potential to aid researchers to gain a better understanding of physiological responses in adolescents and young adults, both typically developing as well as those with pervasive development disorders.

Literature Review of Natural Contact Sensor Paradigms

The research on affect or emotion can be traced back to the 19th century [11]. Traditionally, affect was considered to be a term that meant "lifeless machines," and was normally studied by psychologists. This definition has changed since affective features can be captured and processed by a computer. Am affect-sensitive computer is based on the "affective model" which uses various sensor-captured information; and builds a personalized computing system with the capability of monitoring a human's feeling, interpretation of feelings, as well as giving us intelligent, sensitive, and friendly responses.

Affective computing is one of the most active research topics within human computer interaction, increasingly drawing interest as technology makes it possible to further advance. The strong interest is driven by a wide array of promising applications in many areas such as physiological monitoring, Human-Machine Interaction (HMI), etc. Technological advances, especially when it comes to cooperation between human and machines to complete a task is rapidly advancing. Affective computing is trying to establish that cooperation by giving computers the human-like capabilities of observation, interpretation, and generation of affect features. This relationship is important for harmonious human-computer interaction and requires not only humans to understand machines, but machines

to understand humans. This will certainly increase the quality of human-machine communication and improve the intelligence of machines.

The first step towards machines understanding humans is the development of sensors. Lin et al. [12] discusses various sensor paradigms used to gather body signals from people, which are also called biosensors. There are three existing paradigms for biosensors in HMI which include traditional sensors, wearable sensors, and non-contact sensors. Traditional biosensors originate from medical-oriented sensors. Commercial products such as the FlexComp biofeedback system by Thought Technology, and the Biopac system by Biopac Systems, are available for clinical and research studies. These systems are placed on the human subject and are used to monitor many physiological signals such as SKT, PPG, GSR, etc. [12]. These sensors make a wired contact to the console, where all the physiological signals are collected in real-time for the particular study being conducted. Traditional sensors give a very accurate real-time data but are intrusive sensors.

In order to avoid the clinical setting, wearable sensors emerged as a remedy to avoid clinic visits [12]. Wearable sensors can also measure human's physical movements along with health monitoring. Concerning with human machine interaction, wearable sensors provide the means to monitor the physiological signals of the operator. The physical appearance of wearable sensors can be finger, rings, ear rings, gloves, wrist watches, and wrist bands. If properly designed, wearable sensors can provide the lowest degree of intrusiveness,

furthermore, wearable sensors may still result in some degree of intrusiveness because of physical discomfort caused by them. This is due to issues with power supply and sensor size that may cause physical constraint causing them to be inconvenient to wear.

Noncontact sensors acquire signals without any contact to human body and are completely non-obtrusive. Examples of noncontact sensors are video capturing, thermal detection, and electromagnetic coupling. These sensors are capable of monitoring pupil size, gaze data, and eyelid closure patterns through infrared eye-gaze tracking systems [13]. However, these type of sensors are not currently able to monitor physiological signals [12].

In the study performed by Lin et al. [12] a new paradigm called natural contact sensor is introduced. These sensors have a major difference with other sensors, such that, these sensors are not worn by a human, instead they are embedded in a machine body. The study performed by Lin et al. [12] shows the concept of the natural contact sensors by showing an example of a "smart" steering wheel of a car. These sensors are embedded in the body of a steering wheel and can sense blood volume pulse, skin conductance, skin temperature, and respiration rate. The heart rate could also be monitored by counting the blood volume pulse per unit time. The smart wheel could be used as a close loop system to give researchers and operators feedback about driving events such as accidents or driving performance.

Based on the study conducted by Dr. Lin, the proposed smart wheel non-contact biosensor is compared with traditional and wearable sensors that are already available in the market, which helped him understand the usefulness and the impact of a non-intrusive sensor capable of human-machine interaction [12]. The literature helps us understand the problems and limitations that current paradigms face. The next section investigates the development of a wearable sensor that is unobtrusive for long term assessments.

Literature Review of Wearable Sensor for Unobtrusive Long-Term Assessment of Electro-dermal Activity

Galvanic Skin Response (GSR), sometimes called Electrodermal Activity (EDA) is directly correlated to the sympathetic nervous system activity and thus provides a powerful tool for monitoring arousal [14]. Poh et al.'s 2010 work mentions that due to the lack of sensors that are able to be worn comfortably during normal daily activity and over extensive periods of time, research in the area of GSR is limited to laboratory settings or clinical settings [10]. The research outlined in this literature review, offers a way to perform long-term, comfortable assessment of GSR. The methods used in this research opens up the opportunity for future research that was previously not possible and could further help the understanding of physiological or neurological conditions [10].

In the research conducted by Poh et al. [10] a system is proposed in the development of a novel, unobtrusive, nonstigmatizing, wrist-worn integrated sensor, capable of monitoring long-term, continuous assessment of GSR outside of a laboratory setting.

The human body produces several measurable signals that indicate activity in various systems. One such bio-signal is the galvanic skin response (GSR). GSR reflects activation of the autonomic nervous system, which is responsible for control of visceral functions such as heart rate, digestion, respiration, and perspiration. The majority of these functions operate on an unconscious level. More specifically, GSR is linked to activity in the sympathetic nervous system. As a part of the autonomic nervous system, the sympathetic nervous system stimulates metabolic output to deal with external challenges. When the body is under stress, the sympathetic nervous system responds by elevating heart rate, blood pressure, and sweat production, as well as redirecting blood from the digestive system to the skeletal muscles, lungs, and brain in preparation for motor activity. GSR is considered to be a function of sweat gland activity and pore size. As the body comes under stress, sweat production increases, and the sweat ducts fill. Sweat is a weak electrolyte and a good conductor, so GSR is measured by applying a low voltage electric current to the skin. Increased sweat production creates several low resistance pathways across the surface of the skin.

Despite improvements in measuring equipment since the discovery of electrodermal phenomena more than 100 years ago, much of the research in this

area is limited to observational measurements performed over short periods of time in laboratory settings or artificial clinical environments. For this reason, the experiment outlined in Poh et al.'s 2010 work present a system that consists of the use of the dorsal forearms as recording sites, a miniaturization of the GSR sensor module, as well as the design of a small wristband that allows for unobtrusive and nonstigmatizing continuous GSR measurements during everyday activities [10]. In addition, the study also investigates the choice of electrode material affects performance by comparing the use of conductive fabric electrodes to standards Ag/AgCl electrodes. In the study, they decided to use Ag/AgCl disc electrodes with contact areas of 1.0 cm^2 (Thought Technology, Ltd) as recommended in the literature [15]. These electrodes were chosen to be embedded along with the sensor in a wristband to provide a low degree of intrusiveness. In the study they decided to use the ventral side of the distal forearm as the recording site.

The proposed system was validated with a commercial system such as Flexcomp Infinity (Thought Technologies, Ltd). Overall the complete wearable system as designed by the authors is compact (70 mm x 70 mm x 20 mm), lightweight (40.3 g), and the components used can be purchased off the shelf for approximately $150. In contrast, a commercial system such as the Flexcomp Infinity (Thought Technologies, Ltd.) measures 130 mm x 37 mm, weighs 200 g, and costs $6000 for the data acquisition unit, and an additional $275 for an GSR sensor.

For experimentation setup in Poh et al's work, they had four distinctive setups. The first setup consisted of measurements of the right fingers with the Flexcomp system, left fingers with proposed sensor module, right distal forearm with proposed sensor module using Ag/AgCl electrodes and finally left distal forearm with the proposed sensor module using conductive fabric electrodes. Since comfort was a major consideration for the experiment since it was intended for long-term and continuous use, they integrated the module into a regular wristband made out of terrycloth, resulting in a comfortable, attractive, and lightweight wearable sensor. The data was collected from 26 participants ranging from ages 18-56 during three separate experiments (physical task, cognitive task, and emotional task) each consisting of a baseline, task, and recovery period.

In Poh et al's research findings, the proposed sensor wristband provides a practical and comfortable new solution for widespread GSR assessment [10]. They found out that using Ag/AgCl electrodes, there is a strong correlation between the proposed sensor wristband and an FDA approved GSR measurement system across all tests. They also presented successfully the first long term recordings of GSR during daily activity outside of a laboratory or clinical setting.

Paper written by Dr. Poh and his collaborators presented us with a detailed study indicating that the ventral side of the distal forearm is a viable alternative to the more popular palmar sites for GSR measurements across, physical, cognitive, and emotional stressors. The importance of this study is the unprecedented ability

to perform comfortable long-term and in situ assessment of GSR. This research

presents a significant advancement over existing systems in the GSR field.

Chapter 3

System Design

System Overview

As previously stated, the objective of this research is to develop and test wireless, wearable physiological sensors for affective responses to transmit data on galvanic skin response (GSR), photoplethysmogram(PPG), and skin temperature (SKT) for use in an affect-sensitive adaptive system. The proposed system will help make predictions about the user's affective states and adapt the interaction to alter and improve user experience. Settings to be examined include games of varying difficulty utilizing an embodied robot and activities focused on teenagers with and without autism during periods of waiting (a likely stress-inducing activity for which external interceding feedback may help autistic individuals cope). This research involves designing a system that will allow the interaction to make alterations based on physiological data and inferred affective states to improve human-robot interaction and address social stressors.

The low-cost and highly deployable technologies developed in this research could have significant impact on making physiological signal monitoring more widely available. The results of this research may create completely new intervention strategies for children with autism. In addition, the educational objectives will engage high school, undergraduate, and graduate students in the

design and research evaluation of advanced technologies for autism intervention, human-robot interaction, and affective computing. This research has potential for engaging a wide range of students from electrical engineering, computer engineering, and bioengineering as well as experimental psychology and psychiatry; and findings in this area will address unanswered questions about affective computing. The developed system is a research tool that can be advisory to a wide population; specifically for individuals with autism, who exhibit deficits in emotion comprehension of themselves and others, physiology-based affect-sensitive systems could be notably informative for interpreting emotional cues [6].

A block diagram of the complete system is shown in Figure 3-1. From the figure, the complete system is comprised of 5 components – three physiological sensors (PPG, GSR, and SKT), an Arduino micro-controller, and a Bluetooth Low Energy-based communication medium. The primary objective is to design a low-cost, comfortable, and robust sensor module system that provides the necessary set of measurements needed for affective sensing. In addition, the sensor hardware needs to be small and low power.

Complete System
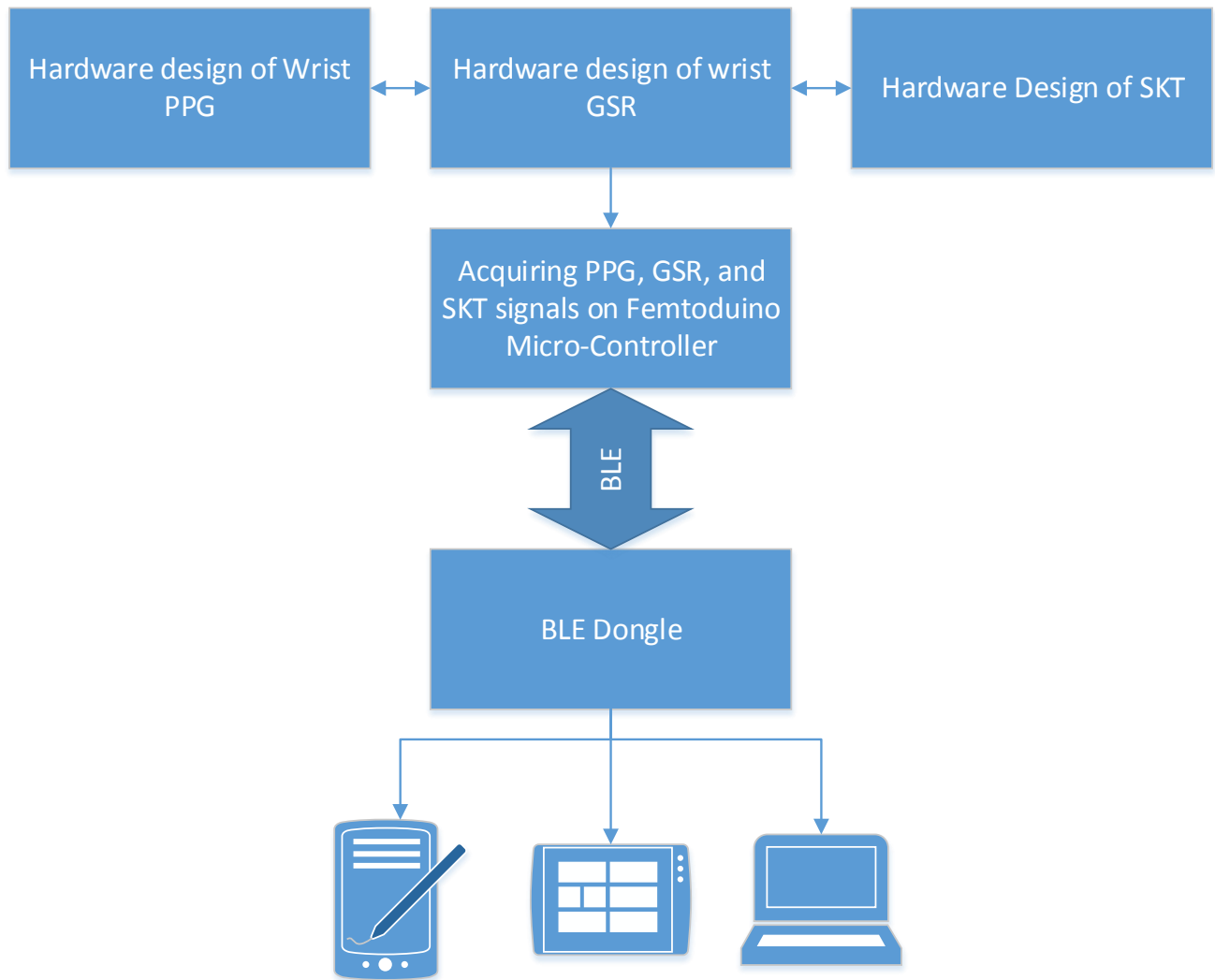
Figure 3.1: Block diagram of physiological monitor system

PPG Model

Photoplethysmography

Photoplethysmography (PPG) is an optical measurement technique that can be used to detect blood volume changes in the microvascular bed of tissue. This technique is equivalent as measuring the heart rate (i.e., pulse measured in beats per minute). It has widespread clinical application, with the technology utilized in commercial available medical devices, such as oximeters, vascular diagnostics and digital beat-to-beat blood pressure measurement systems. Researchers have concluded that these signals are among the ones with the greatest potential for minimally-invasive data collection, and although typically used on fingertips these signals could be feasibly incorporated into an arm band/wrist band for collecting data during a range of activities.

The tissue in the human body are generally considered to be nontransparent, while transmission of light is seen at the skin or soft tissue levels. By means of this property of tissue, PPG operates by observing the light absorption and scattering properties of capillaries, affected by composition of blood and engorgement during systole phase of the cardiac cycle.

The below figure (Fig: 3.2) shows a symbolic detected signal. The signal is primarily categorized into two components, a pulsatile ('AC') waveform attributed to cardiac synchronous changes in the blood volume with each heartbeat. The second category contains a slowly varying ('DC') baseline with various lower

frequency components attributed to respiration, sympathetic nervous system activity, and thermoregulation. Figure 3.3 is the detailed view of top portion of Figure 3.2 which is the pulsatile component of PPG signal.
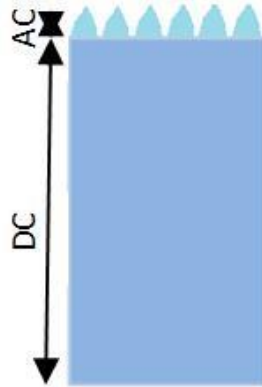
Figure 3.2: Reflected light intensity at PPG site (Image adapted from M. Kalsi(6))
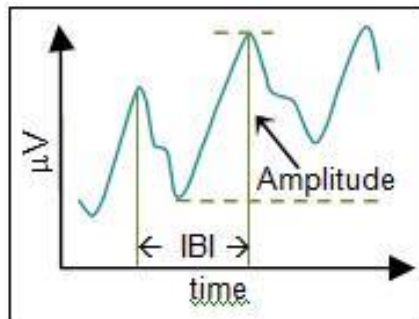
Figure 3.3: Typical photoplethysmogram (PPG) graph

The basic form of photoplethysmography technology requires only a few opto-electronic components: a light source to illuminate the tissue, and a photodetector to measure the small variation in light intensity associated with changes in blood volume. A light source is typically an LED (light emitting diode),

which converts electrical energy into light energy and has a very narrow bandwidth as compared to other sources of the light. LEDs are also preferred because they are very compact, have a very long operating life, operate over a wide range with small shift in the peak-emitted wavelength, and are mechanically robust and reliable. Photodiodes are a popular photodetector as they are compact, low-cost, sensitive to light wavelength they are designed for, and have a fast response time. A photodetector converts light energy into electrical energy. Therefore, a PPG sensor can use an LED, to light up a location of interest with capillaries, in conjunction with a photodiode, to convert the reflected light energy into electrical energy.
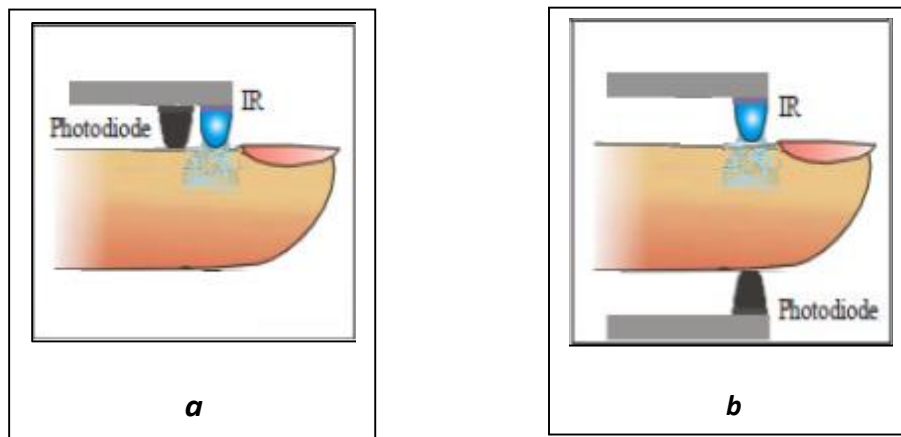
Figure 3.4: Operational configurations of PPG operation (Image adapted from D. Parekh [7])

PPG operates in two operational configurations - reflective or transmissive. Figure 3.4a shows a reflective PPG, where the photodiode and light source are placed next to each other on the same side of the body. The emitted light gets reflected by blood tissues, and the photodiode detects the reflected light. Figure 3.4b shows a transmissive PPG, where the light source and a photodiode are placed on the opposite side of the body part but perform the same task as the reflective mode. Reflective mode PPG is more commonly used because transmissive mode PPG imposes more restrictions on the body locations available for the study

Hardware Design

The figure below demonstrates the block diagram of the system used to acquire the PPG data though electrical signals from the PPG sensor. The analog sensor receives input from the sensor placed on the skin in the form of current. In the first stage current is converted to voltage; this is because the output from the optical sensor is current which has to be converted to voltage for further processing. In the second stage, the signal is fed to an amplifier with a gain of 4. In the later stages the signal is filtered using a passive RC filter with a HIGHPASS cutoff frequency of 0.159Hz and a LOWPASS cut off frequency of 5.307Hz.
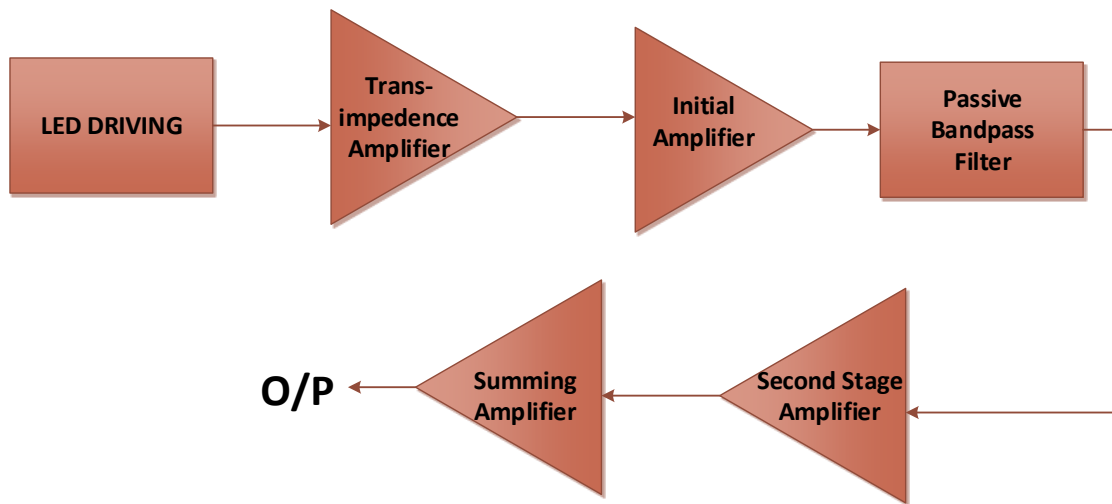
Figure 3.5: Block Diagram for PPG detector circuit

Figure 3.6 shows the schematic of the PPG sensor. The electronic circuit has six basic stages- LED driving circuit, transimpedence amplifier, initial amplification, passive bandpass filter, second stage amplifier, and summing amplifer
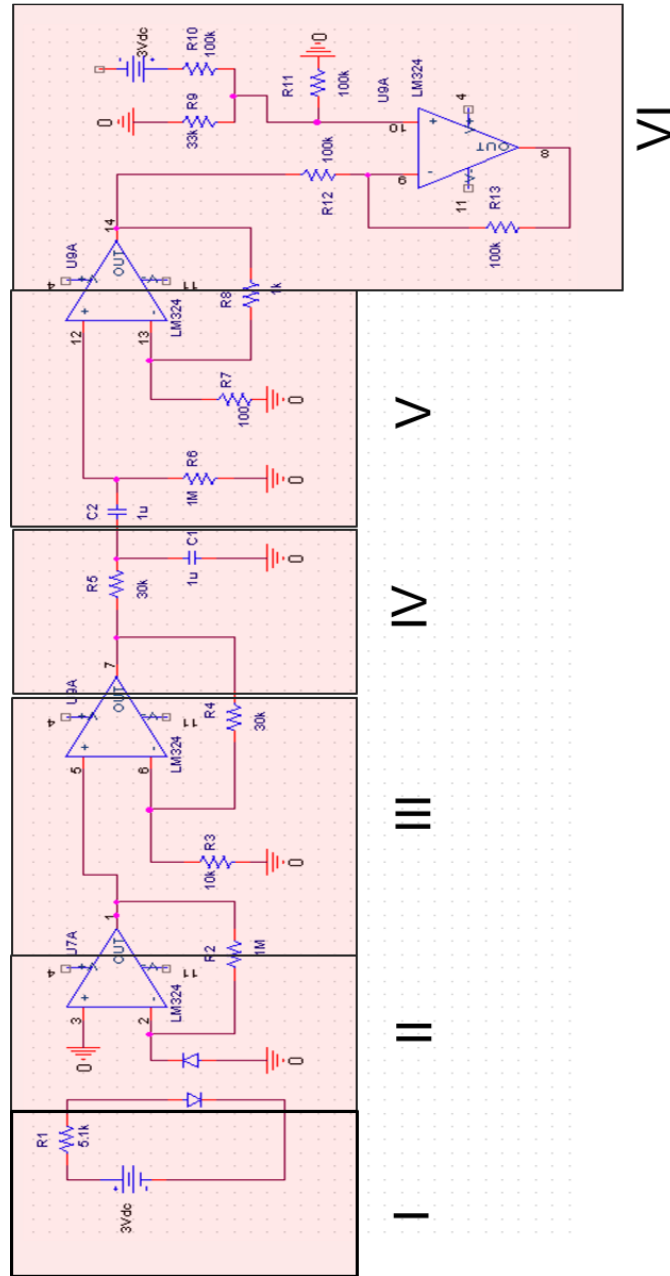
Figure 3.6: Schematic of PPG sensor

LED Driving Circuit

Figure 3.7 illustrates the circuit used to generate the light from an infrared LED. The forward bias voltage for the LED is 1.5 V and a 5.1kΩ resistor was used for current limiting. IR light cannot be seen with the normal human eye; therefore, a camera was used to confirm the LED was turning on properly.
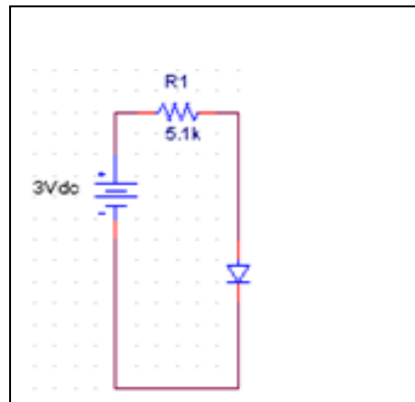
Figure 3.7: LED driving circuit (Block 1)

Photodiode amplifier

Figure 3.8 show the circuit used for converting photodiode current to voltage. IR LED generates the light of wavelength 850 nm and sends it in the body. Tissues absorb part of this light and remaining part gets reflected. Photodiode converts the reflected light into an electrical current. After the optical signal is converted to current, a resistor is used to convert the current to electrical voltage. We used the circuit as shown below from application notes of LM 324 to convert the current to voltage
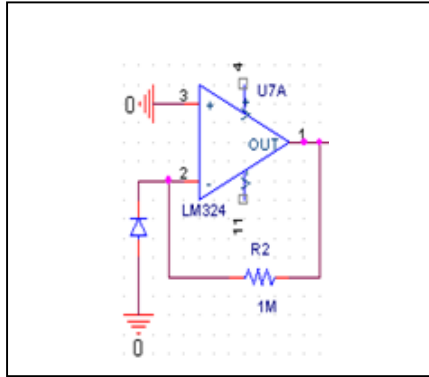
Figure 3.8: Photodiode amplifier (Block 2)

Initial Amplification

The voltage we get at the output of photodiode amplifier is in millivolts. We used LM324 as a non-inverting amplifier (see Fig. 3.9) to raise the voltage for further treatment. Input signal is given to the positive pin of an amplifier and a 10kΩ resistor is connected between negative pin of amplifier and the ground. We used a feedback resistor of 10kΩ. This gave us the gain of 4.
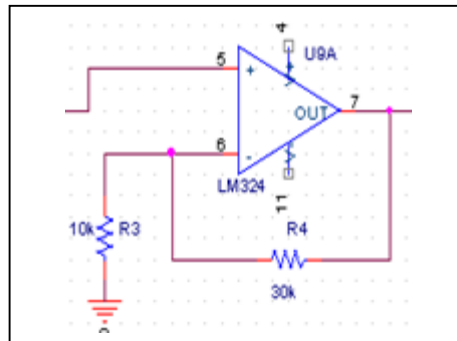
Figure 3.9: Initial amplifier (Block 3)

$$Gain = 1 + \frac{R_1}{R_2} \qquad\qquad Gain = 1 + \frac{30k}{10k} = 4$$

32

Figure 3.10 shows the gain characteristics of LM324. Initially we used a gain of 10, but that gain level did not result in a linear relationship with the input. Hence the gain was adjusted so that the output fits within the linear region of LM324.
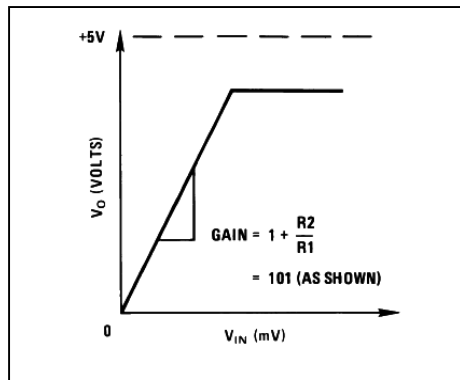
Figure 3.10: Gain Characteristics of LM324

Passive bandpass filter

Signal we obtain after the amplification consist of DC component due to absorption of light by blood tissue. It also contains high frequency signals generated due to ambient noise and motion artifacts. We used passive RC filter (see Fig. 3.11) to remove the DC component as well as high frequency components.
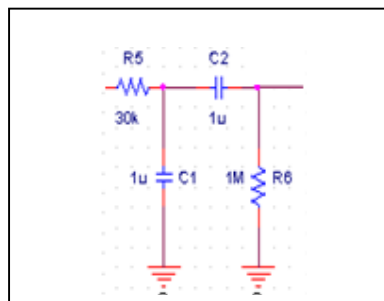
Figure 3.11: RC bandpass filter (Block 4)

$$Cut\ off\ freq.\ for\ LPF = \frac{1}{2\pi RC} = \frac{1}{2\pi(30k)(1\mu)} = 5.307\ Hz$$

$$Cut\ off\ freq.\ for\ HPF = \frac{1}{2\pi RC} = \frac{1}{2\pi(1M)(1\mu)} = 0.159\ Hz$$

Second stage amplifier

The figure 3.12 illustrates the LM324 non-inverting amplifier. The gain of the following circuit is 10.
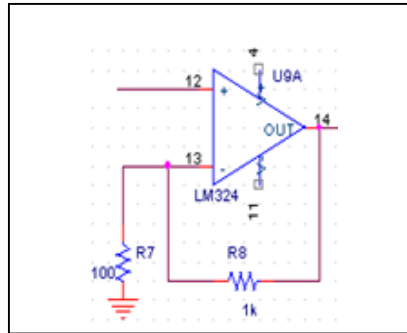
Figure 3.12: Second stage amplifier (Block 5)

Summing amplifier

The output of the second stage amplifier ranges from a few negative millivolts to a few volts. The Microcontroller does not process the negative voltage. To solve this issue we used a summing amplifier as shown in figure 3.13. A positive 1.5 volts was added to the output of the second stage amplifier to make sure that the output of the summing amplifier is always in the positive range.
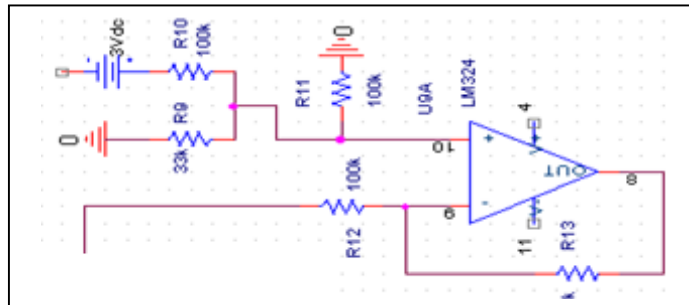
Figure 3.13: Summing amplifier (Block 6)

Selection of Components

LEDs and photodiode

Blood pulsations are difficult to detect at some wavelengths depending on measurement site and the subject hence the selection of appropriate LED and a photodiode are essential to obtain a good signal. Researchers have concludes that light with longer wavelength penetrates deeper, and the penetration can be more than 1 or 2 mm, which is enough to reach to the distributed capillary bed and reach the pulsatile blood vessel [17].
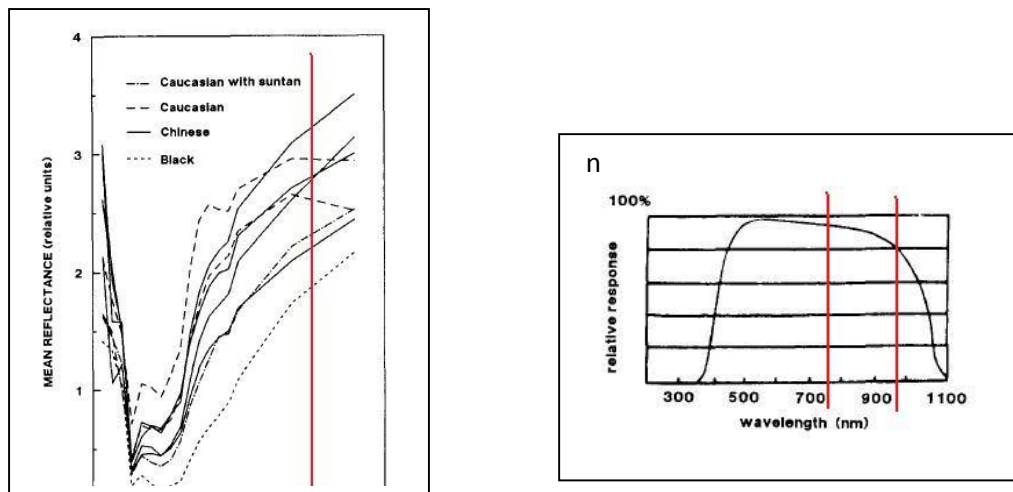
Figure 3.14: Measurement of reflectance vs. wavelength and typical response of photodetector (Images adapted from Cui W. et. Al.(8))

For this project we used 10mm infrared LED which has a wavelength of 850 nm shown by red line in figure 3.14. Photo diode selected has a spectral bandwidth of 750-980 nm as shown in figure 3 and it has a peak sensitivity of 900 nm. This selection of LED and photodiode gave us good reflectance and detection of light.

Selection op-amp

Initially the PPG circuit was constructed with the LM741 op-amp as shown in figure 3.15. This chip requires a dual power supply of ±16 volts. This substantially increased the wiring in the circuit. The figure below shows the construction of PPG sensor with LM741.
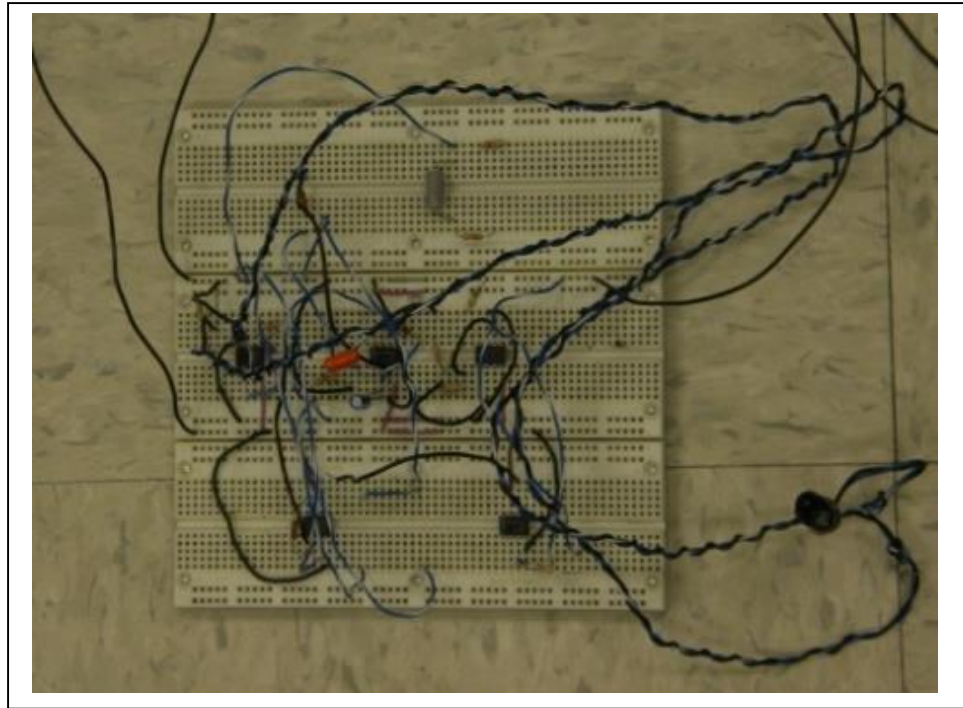
Figure 3.15: PPG sensor with LM741 (photo taken by author)

Later on we decided to use LM324, which is a low power quad operational amplifier. This IC consist of four independent, high gain, internally frequency compensated operational amplifiers which are designed specifically to operate from a single power supply over a wide range of voltages. Using LM324 eliminates the need for dual supplies.
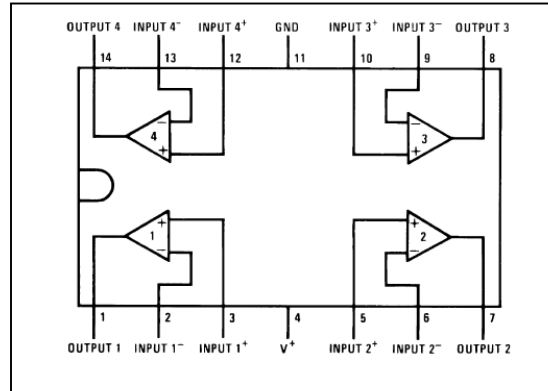
Figure 3.16: Connection diagram for LM324 (image
adapted from datasheet of LM324)

Figure 3.17 shows the construction of PPG sensor with LM324. It can be

seen that size and wiring has reduced drastically.
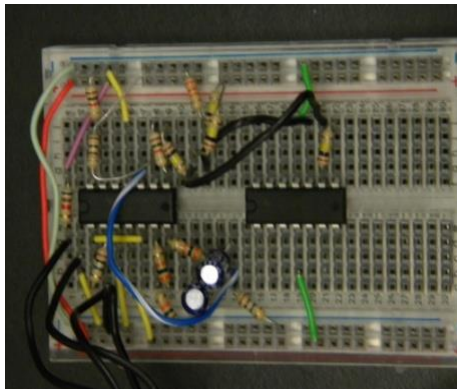
Figure 3.17: PPG sensor with LM324 (photo taken by author)

Testing of initial design

PPG waveform on the oscilloscope

Figure 3.18 illustrates the waveform seen on the oscilloscope. The sensor produces an accurate PPG waveform, including the diacritic notch as seen in the figure below.
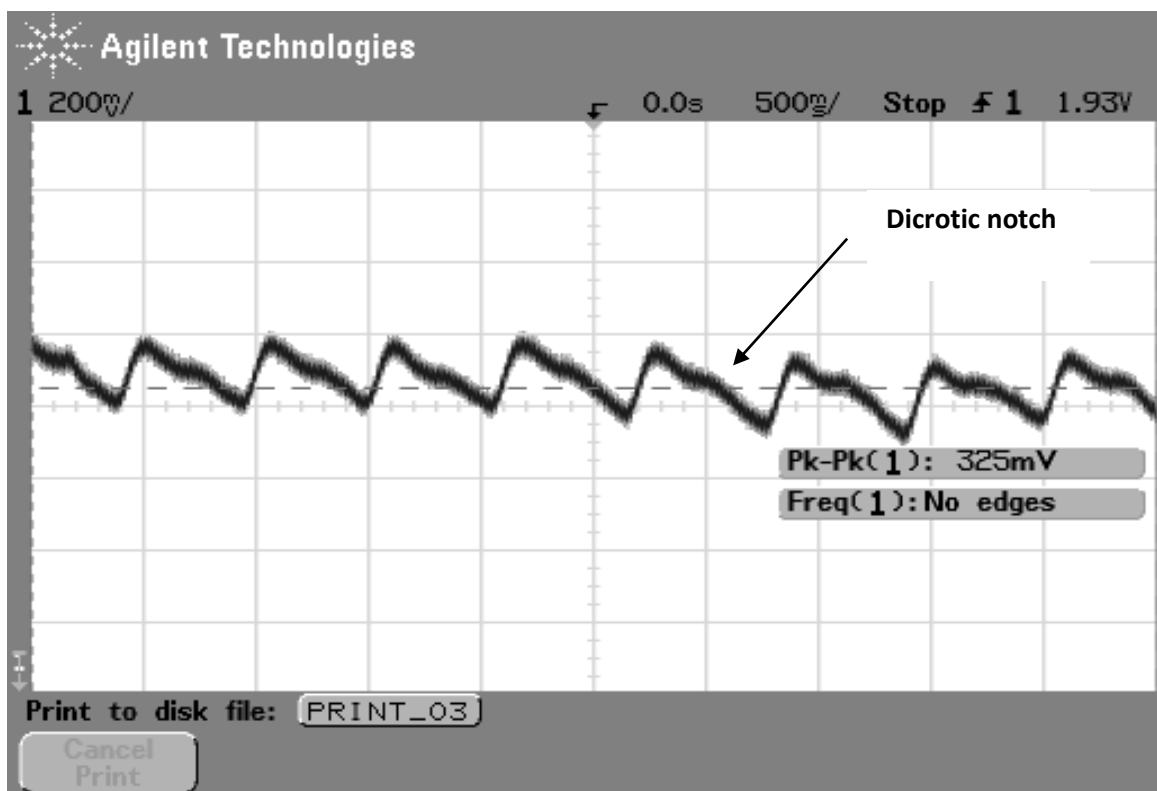
Figure 3.18: PPG waveform seen on oscilloscope

PPG sensor size reduction

As previously mentioned, the PPG circuit was constructed with an LM741 op-amp as seen in Figure 3.15, this requires a dual power supply of +/- 16 volts, but was later replaced with an LM324 (i.e., a low power quad op-amp), which reduced the number of required leads and the overall size of the circuit (Figure 3.17). The size of the current in-house sensors needs to be small enough to be embedded into a wrist band or glove. Therefore, the design was reduced even further by employing surface mount devices which minimizes the footprint to 25mm x 25mm (roughly the size of a US quarter) as seen in Figure 3.19. In order to accomplish the reduction of the circuit board, EAGLE PCB Software was used, in particular CadSoft EagleCAD PCB design software, which is available for free for the design of printed circuit boards.
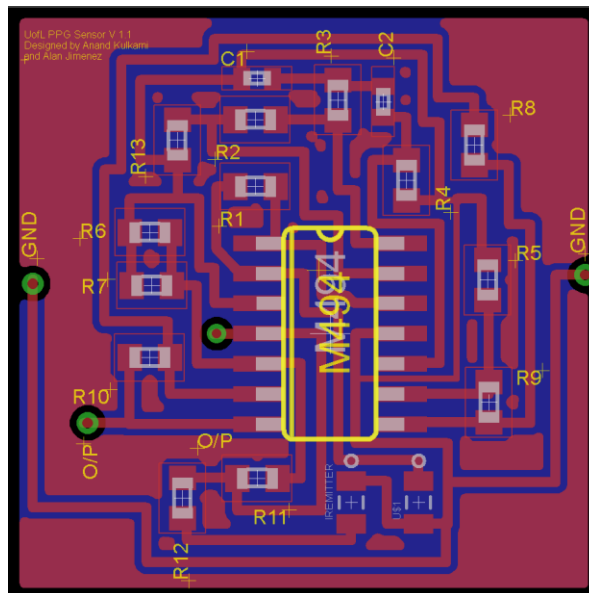
Figure 3.19: PPG manufactured to the PCB (printed circuit board) stage by using CadSoft EAGLE - solder mask and silk screen top side layout

Soldering PCB components

The printed circuit board was received from the manufacturer and it was time to solder the components to the board. The IC circuit was one of the hardest components to solder. The capacitor, being the smallest package (0402 1.02L x 0.51W) component on the parts list, was very difficult to solder. The resistors were a little easier. Their package is 0603 at 1.60L x 0.80W. With enough practice and patience we were able to solder all the components on the printed circuit board.

Soldering PCB components by stage

After our first failed attempt to see any generated signal, we decided to solder the components by different stages in the circuit in order for more efficient testing. If we break the soldering by stages, this will permit us to see if our soldering caused any short and to see that every stage is working as it should.

Stage 1- In this stage we moved forward by soldering the IR emitter to generate light from the infrared and a 5.1KΩ resistor was used for current limiting. We also solder 2 wires, one for ground and the other for applied voltage.

Testing- After the soldering was finished we proceeded to test the IR emitter by using our digital cameras and see if we could see a small white light. The human eye cannot detect the infrared white flash directly, that is why we needed to look though the camera's LCD screen. We were able to see the white light and this stage was completed!

Stage 2- In this stage we soldered the photodiode. IR LED generates the light of wavelength 850 nm and sends it to the body. Tissue absorbs part of this light and remaining part gets reflected. The photodiode converts the reflected light into electrical current. A resistor was used to convert the current into electrical voltage. Than the circuit 494-CSD was used to convert the current to voltage.

Testing- Initial testing was done by applying voltage to the circuit and using an oscilloscope to see the output of pin 1 of the circuit. In doing this we should see the generated signal change its amplitude by just putting our hand over the photodiode. We did not see any changes in the signal. This meant that something was wrong with the photodiode. After looking over our notes we found that the diode was positioned wrong! The cathode and the anode was to be reversed. After de-soldering the component and flipping the anode and the cathode we soldered the component again and we managed to see the change of amplitude in our signal when we applied our hands over the LED photodiode. This meant that the photodiode was now absorbing light. The amplitude was around 195 mV. This stage was completed!

Stage 3- In this stage we will be able to perform initial amplification. The voltage we get at the output of photodiode amplifier is at 150 mV. This voltage should get amplified using Maxims 494-CSD -as a non-inverting amplifier (see Fig. 3) to raise the voltage for further treatment. There is an input signal given to the positive pin the amplifier and the ground applied. A 10kΩ resistor is connected between negative pin and ground. For feedback we used a 10kΩ resistor. This gave us a gain of 4.

Testing – In this stage, after the proper soldering was completed we moved forward with our testing. We used the oscilloscope again to see if the voltage emitted by the photodiode was amplified. This was confirmed after we measured the amplitude of the signal to be around 780mV. This stage was completed!

Stage 4 – In this stage we wanted to apply a passive band pass filter. The signal that we obtained after amplification consists of DC component due to absorption of light blood tissue. It also contains high frequency signals generated due to ambient noise and motion artifacts.

Testing – In this stage we used the oscilloscope again to see the signal. We were able to see a more noise free signal. This stage was completed!

Stage 5- In this stage we repeated stage 3 and did a second amplification of the circuit. The gain of the following circuit is 110

Testing- Again we used the oscilloscope to check for the generated signal and we were able to confirm that the signal was amplified.

Stage 6 –This stage involved summing the amplifier. The output of second stage amplifier ranges from a few negative millivolts to a few volts. The Microcontroller does not process the negative voltage. To solve this issue we

used a summing amplifier as shown in figure 6. After soldering all the components on the PCB board we soldered a wire in pin 7 of the 494-CSD circuit for the output of the sensor.

Testing – For this stage we applied the oscilloscope to the output of the sensor to see if we could generate a signal confirming that the sensor works. After tweaking the oscilloscope and playing around with the signal, we were able to generate a wave form after we applied our index finger on the IR emitter and photodiode as seen in Figure 3.20. This waveform was in conjunction with the heart rate, demonstrating that we are in the right track and that the sensor works!

Figure 3.20: PPG waveform seen on oscilloscope
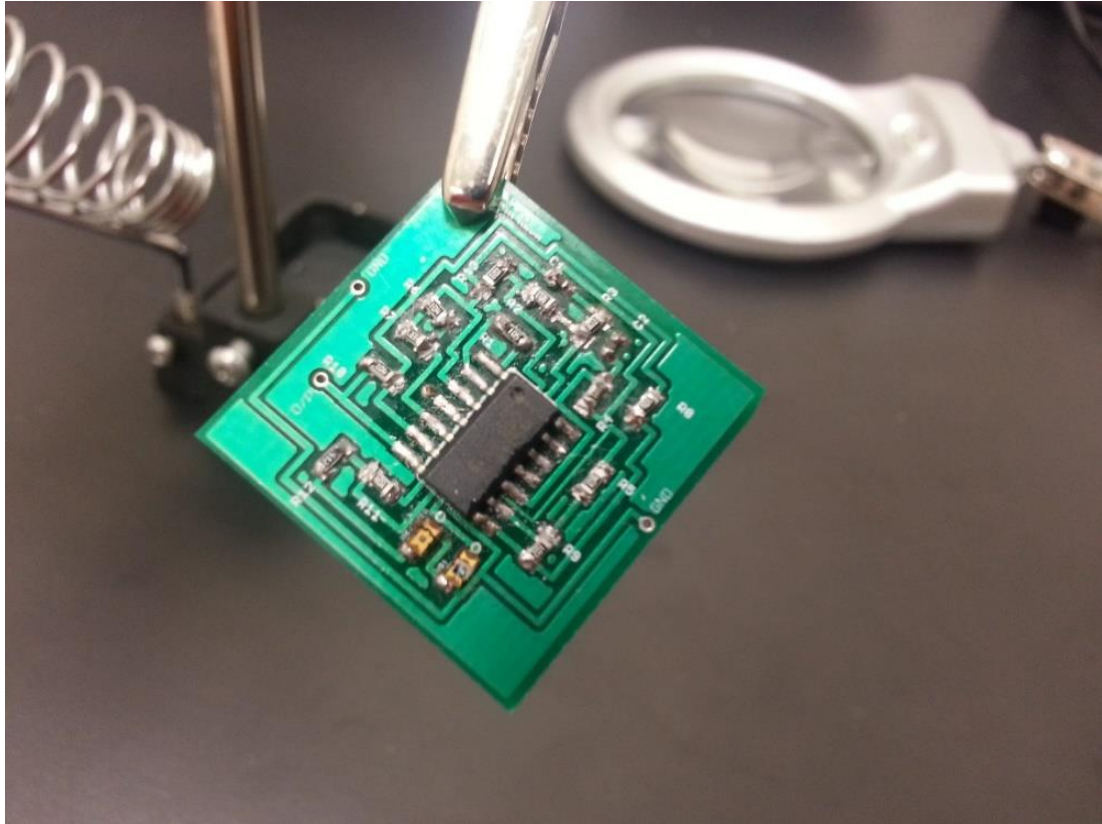
Figure 3.21: PCB design phase minimizes the footprint to 25mm x 25mm
(roughly the size of a US quarter)

GSR Model

Galvanic Skin Response

When a person becomes emotionally stressed or physically aroused, a response is triggered from that person's skin. This phenomenon is known as galvanic skin response (GSR), and this response can be an indicator of one's level of excitement. Galvanic skin response directly correlates to the sympathetic nervous system activity and provides a powerful tool for monitoring arousal and certain aspects of the autonomic regulation [18]. During arousal or stress, in accordance with the sympathetic response, sweat production increases, and the sweat ducts fill. Sweat is a weak electrolyte and a good conductor, so GSR is measured by applying a low voltage electric current to the skin.

The reversed process, known as parasympathetic response, is started when the body is relaxed. During relaxation the conductivity in the skin is diminished. The sympathetic and parasympathetic responses are two different phases when it comes to the change in skin conductivity.

When applying a conventional 0.5 Volts across the skin and measuring changes in the corresponding conductance, the emotional state of the person can be studied. This is because medical experiments have shown that the magnitude of the electrical conductance in a person's skin is directly correlated to their emotional state. The short term changes in electrical conductance are also correlated with a person's mental response. It is important to note that changes

in skin conductivity can be the result of many types of arousal. By observing these changes, it is impossible to conclude without prior knowledge whether the person has become angry, happy, surprised, physically stressed, etc.

Galvanic Skin Response (GSR) readings are simply measurements of electrical resistance through the body. The preferred way of measuring GSR data is by measuring the conductance rather than the resistance of the body. This is because of the true nature of the skin (it is not a single resistance, but instead a series of parallel resistors). The most common unit of measurement in GSR is the microSiemen (µS).

When setting up GSR measurements in a subject, it is usual to place electrodes (generally Ag/AgCl type), on palmar sites [10]. GSR measurements are conducted by attaching two leads to the palmar site of the skin of the hand and acquiring a base measure. Then, as the activity being studied is performed, recordings are made from the leads. One electrode emits an electrical signal while the other primarily acts as a receiver, this when the circuit is established and passes across the skin. The skin acts as a series of resistors.

Hardware Design

The figure below demonstrates the block diagram of the system used to acquire the GSR data though electrical signals from the GSR sensor. There are many circuit schematics for GSR sensors. Some GSR schematics are simple and

some are very complex but they provide better accuracy in the end results. After researching different circuit schematic designs, we decided to implement a working design from a group of researchers from the University of Cornell [19]. This design was modified and improved from the original adapted version. The design was chosen because of the detailed documentation and the low cost of materials used for the circuit.
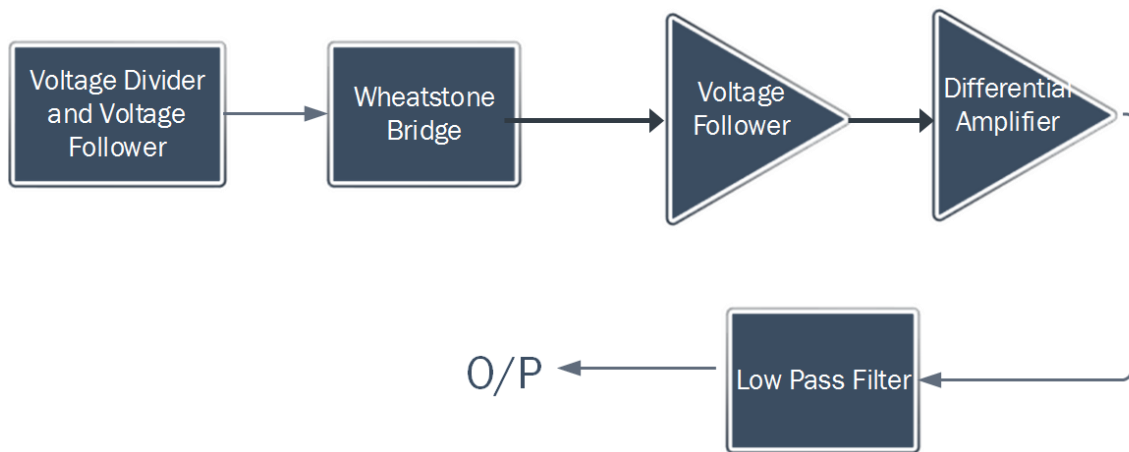
Figure 3.22: Block diagram for GSR circuit

Figure 3.23 shows the schematic of the GSR sensor. The electronic circuit has five different stages.
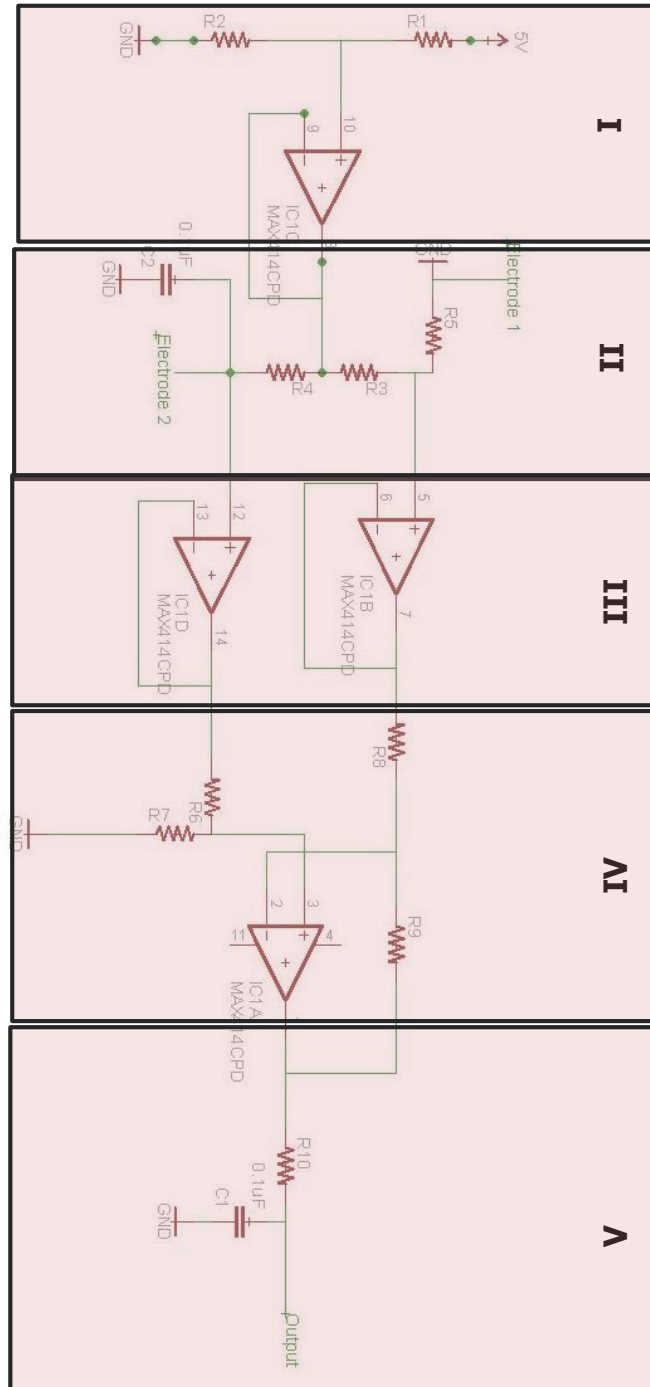
Figure 3.23: Schematic of GSR sensor

Voltage divider and voltage follower

Figure 3.24 illustrates the circuit used to generate 0.5V. This is because human skin conductance is only linear when voltages applied are below 0.5V. In order to accomplish this, we used a voltage divider to bring the bridge V_{dd} voltage to 0.5V. 5 Volts DC is used to power the circuit and the power is dropped across the 1K and 9K resistors for a voltage of 0.5V on the top of the wheatstone bridge. The output V_{dd} voltage of 0.5V is than fed into a voltage follower. The voltage follower is able to reproduce the signal from the source to its output without the problem of changing the original signal by drawing current from it.
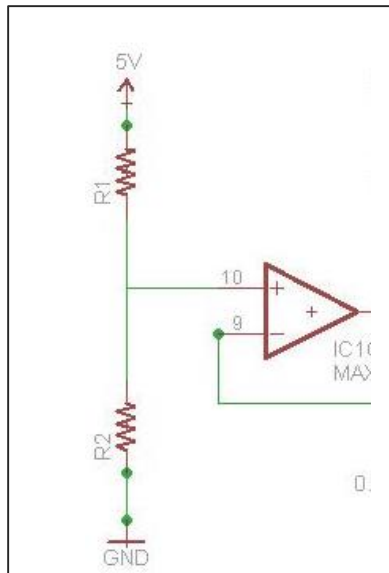
Figure 3.24: Voltage divider with voltage
follower (Block 1)

Wheatstone bridge

Figure 3.25 shows the circuit making use of a wheatstone bridge to measure the change in conductance of the skin. Two electrodes are place on the same palmar surface of the hand. One of these electrodes is connected to the ground (can be considered the driven electrode). The other electrode can be attached to the other side of the palmar surface. Each electrode starts with a 2 Kohm resistor. One electrode is non-grounded and the electrode is added. The driven electrode has a 40 Kohm resistor going to ground, where is attached to the other side of the skin. As the conductance of the skin between the two electrodes changes, the current through the leg of the bridge will have to increase to get a balanced bridge. This means that the voltages above the 40 K resistor and the top electrode vary with time, based on the conductance of the human skin.
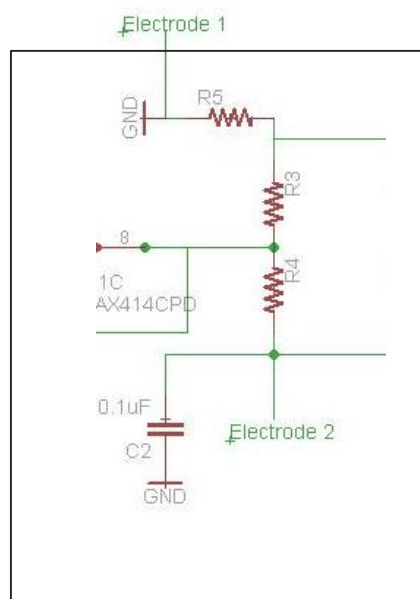
Figure 3.25: Wheatstone bridge (Block 2)

Voltage Followers

The voltage across the wheatstone bridge is used to measure the skin conductance across the two electrodes. Figure 3.26 shows the voltage at both ends of the bridge being fed into a voltage follower. The voltage follower turns out to be very useful, because the input impedance of the op amp is very high, giving effective isolation of the output from the signal source. This means that you can draw very little power from the signal source, avoiding "loading" effects.
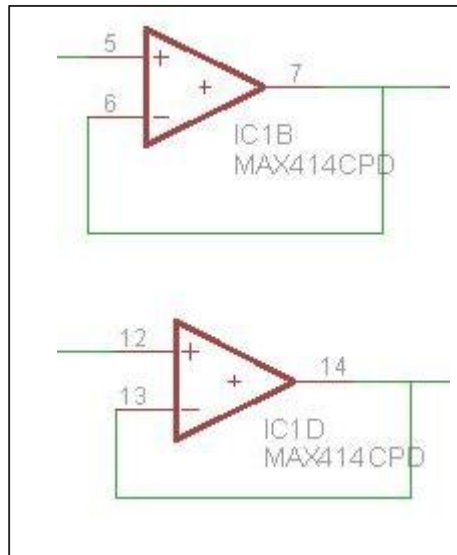
Figure 3.26: Voltage followers (Block 3)

Differential Amplifier

After the voltage followers the two inputs run into a differential amplifier, which amplify their difference by a factor of 24. The purpose of the differential amplifier is to amplify the difference between two input terminals. The differential amplifier rejects the noise common to both signals and in effect helps the signal difference between the input terminals.
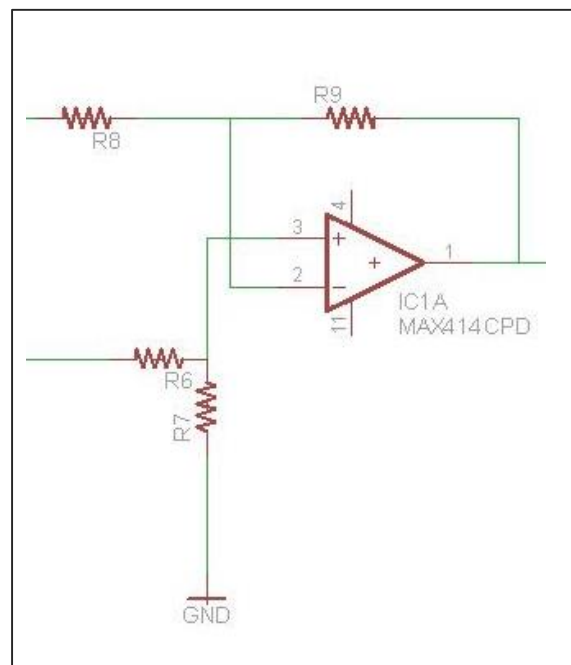
Figure 3.27: Differential Amplifier (Block 4)

Differential Amplifier Math

This section is used to explain the math behind the differential amplifier circuit found in the GSR system. The first step is to write the output due to each input. By driving the positive input $V_{in}+$ and setting the negative input $V_{in}- = 0$ Volts, you can get $V_0 1$

$$V01 = V_{in+} \times \frac{R7}{R6 + R7} \times \frac{R9 + R8}{R8}$$

Driving V_{in-} and setting $V_{in}+ = 0$ Volts, you get $Vo2$

$$Vo2 = V_{in-} \times \frac{R9}{R8}$$

Now we put it all together and add the two outputs to get $Vo = Vo1 + Vo2$

$$Vo = V_{in+} \times \frac{R7}{R6 + R7} \times \frac{R9 + R8}{R8} - V_{in-} \times \frac{R9}{R8}$$

Choosing ratios R9/R8 = R7/R6 will make the following equality: R7/ (R6+R7) = R9/ (R8 +R9). Substituting and simply we get

$$Vo = (V_{in+} - V_{in+}) \times \frac{R9}{R8}$$

Only the difference will get amplified.

Low pass filter

The output of the differential amplifier is than fed into a low pass filter with cutoff frequency of 0.5Hz. The low pass filter enable us to filter out unwanted signals and separate and pass sinusoidal input signals based upon their frequency. The low pass filter will essentially allow low frequency signals from 0Hz to the cut-off frequency of 5Hz, while blocking any higher frequencies. After the low pass filter the output is fed to Port A0 of an Arduino Uno micro controller. [20]
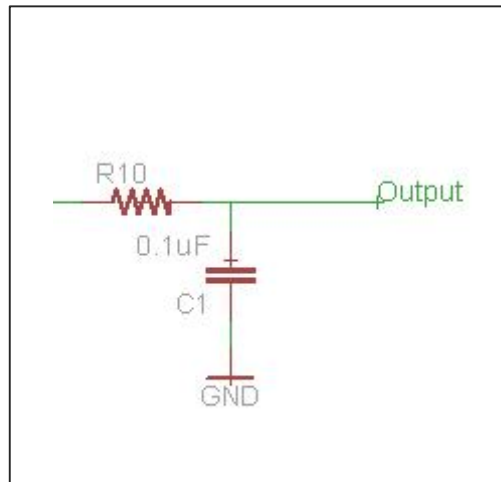
Figure 3.28: Low pass filter (Block 5)

Initial breadboard prototype

The initial hardware design of the GSR system was done on a breadboard. Initially the GSR circuit was constructed with a LM324 op-amp as shown in Figure 3.29. The LM324 is the same low power quad operational amplifier that was used for the PPG breadboard system.
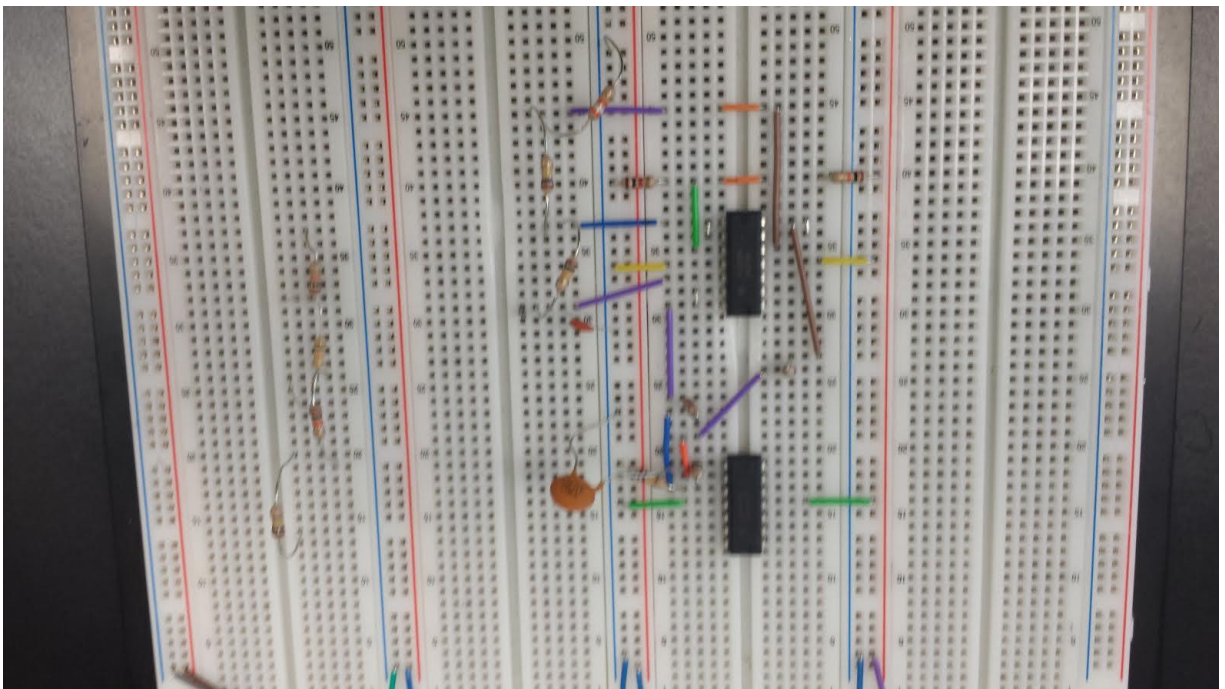
Figure 3.29: GSR sensor with LM324 (photo taken by author)

Selecting the electrodes

Choosing the right electrodes for the leads can be important because they are essential in increasing the resolution of your data sampling. There are different kinds of electrodes that you can use such as O-rings, pennies, washers, skin conductance fabric and professional manufactured Ag/AgCl electrodes. As mentioned earlier, Poh et al.'s 2010 work recommends Ag/AgCl electrodes as the recommended standard for more sensitive analysis of GSR measurements. We chose to follow the recommendation and decided to use Ag/AgCl electrodes.
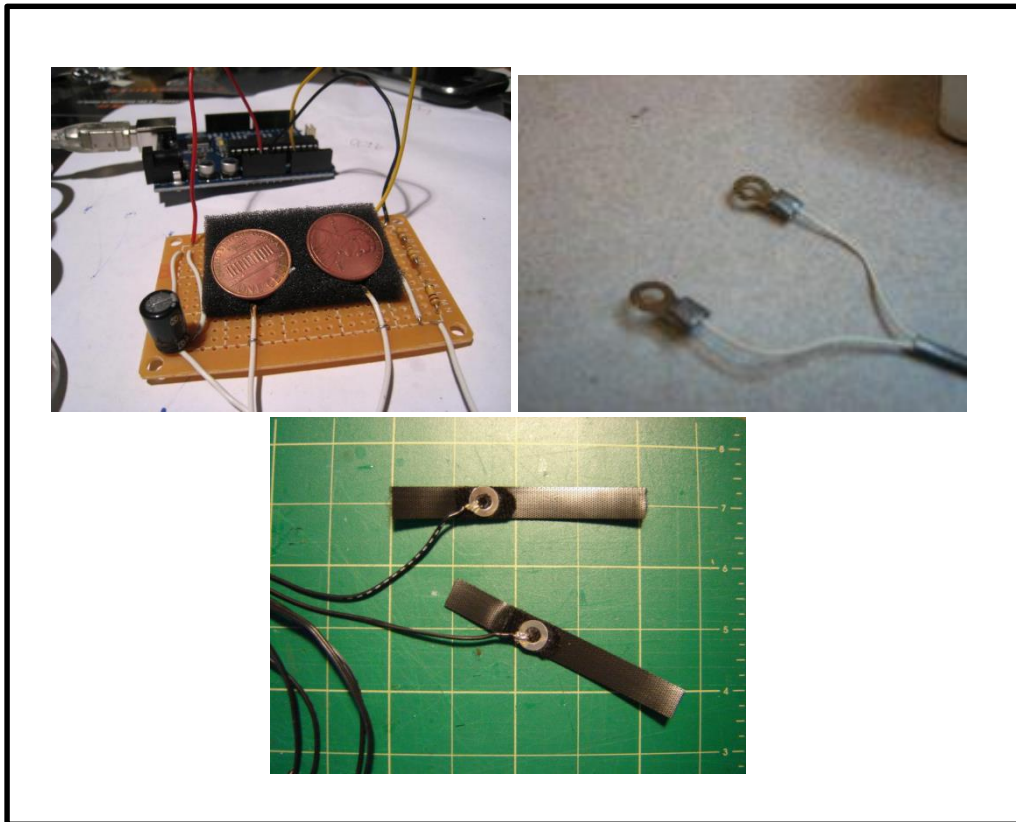
Figure 3.30: Selection of different electrodes

Testing of initial design

Validating and calibrating the newly designed GSR sensor can be complicated given the many parameters contributing to the raw GSR signal. One of these parameters can be circuit design, which includes precision, dynamic range, and noise level. Another parameter that can affect the results can be the electrode design, which includes contact area, material selection, placement, and contact pressure. Another important contributing factor is the concentration of sweat glands at the specific measurement area.

All of these parameters are important to consider and the results can vary depending on these factors. In order to address this factors we carried two types of tests. The first carried test was to test the GSR system independently of human configuration. The GSR sensor was isolated when testing was conducted. To do this we used a series of 5 % fixed resistors ranging from 40kΩ to 3MΩ in place of the electrodes and human skin. This ranges are in accordance with the normal ranges of human skin resistance, allowing an appraisal of the performance over the normal operating range of the sensor. The output of the sensor produced a voltage output. This measured resistances were compared with the laboratory multimeter, with an accuracy of 1%. Figure 3.31 shows the graph that was produces based on the range of fixed resistances and the voltage output.
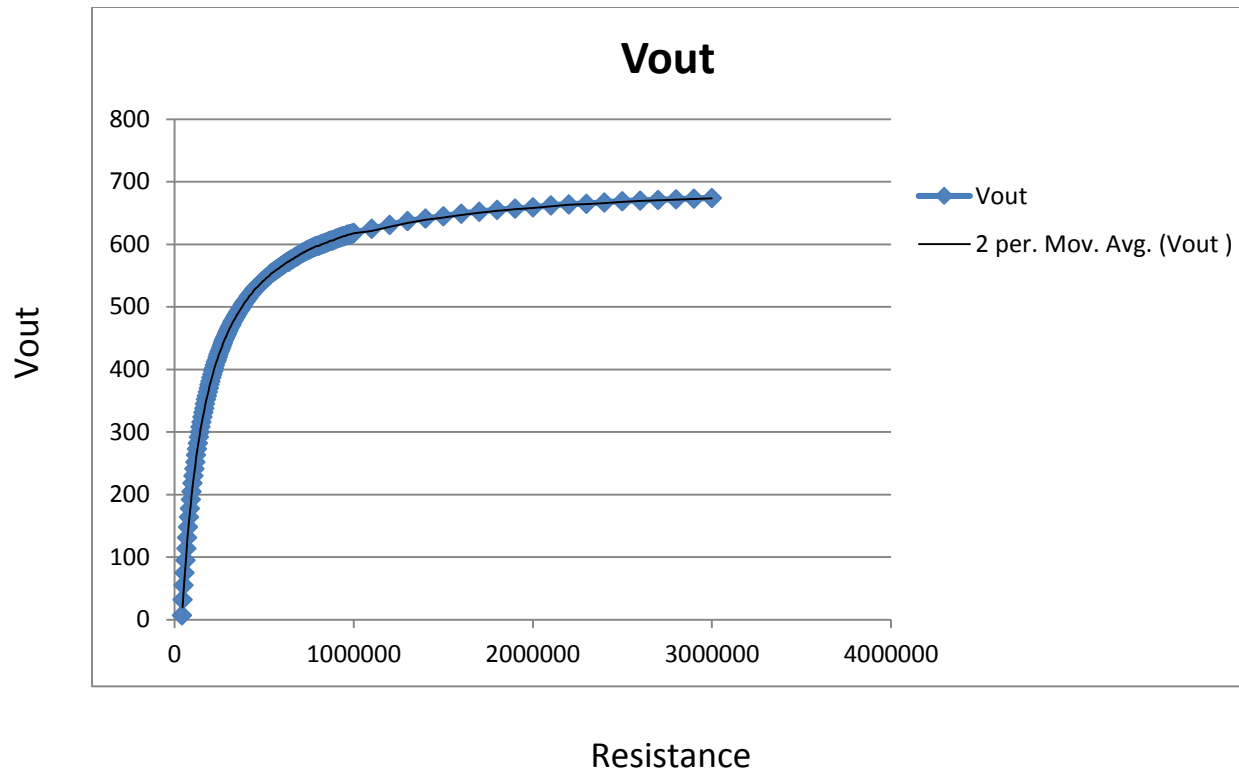
Figure 3.31: Voltage vs. Resistance graph

After observing the voltage output of the in-house GSR sensor on the multimeter and confirmed it was right, the next step was to read the data and store the data for further analysis. We used Arduino Uno (see Fig. 3.32) microcontroller board to record the data. The output of the GSR sensor was sent to the microcontroller through Port A0, the input for the ADC (Analog to Digital Converter). The unit for skin conductance is microSiemens. In order to make the conversion from voltage to microSiemens we inverted our previous fixed resistance data to get the resistance output. The ultimate goal is to get a linear resistance equation so that we can use it to convert to microSiemens. By Ohm's law,

59

resistance equals voltage divided by current, therefore conductance, the reciprocal of resistance, equals current divided by voltage. The unit of resistance is the Ohm (Ω), and the preferred unit of conductance is microSiemens (μS). It is the reciprocal of MegaOhm. Zero resistance (a short circuit) is infinite conductance; one MegaOhm is one microSiemens; 2 MegaOhms is 0.5 microSiemens; 100kOhms is 10 microSiemens, and so on.
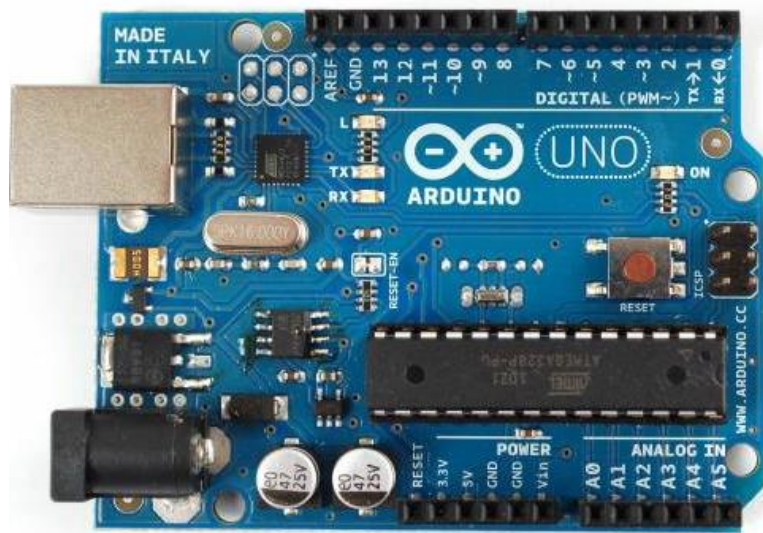
Figure 3.32: Arduino Uno Microcontroller

We followed by using MATLAB in order get the linear regression equation of the resistance curved. MATLAB is a high level language and interactive environment that enables you to perform computational intensive tasks faster than traditional languages. Matching the curved resistance plots proved to be challenging because the equation is a high order degree polynomial. To address this challenge we used multiple tangent line linear equations and used -"if"-

statements to reflect each linear equation when programing the output of the resistance. Finally, we used the following mathematical relationships and programmed these in the Arduino open source environment to convert to microSiemens. The development environment is written in Java. The code for the Arduino output is included in an appendix.

$$Resistance = R = \frac{V}{I} \ (MegaOhms)$$

$$Conductance = C = \left(\frac{1}{R}\right) \ \times 1000000 \ (microSiemens)$$
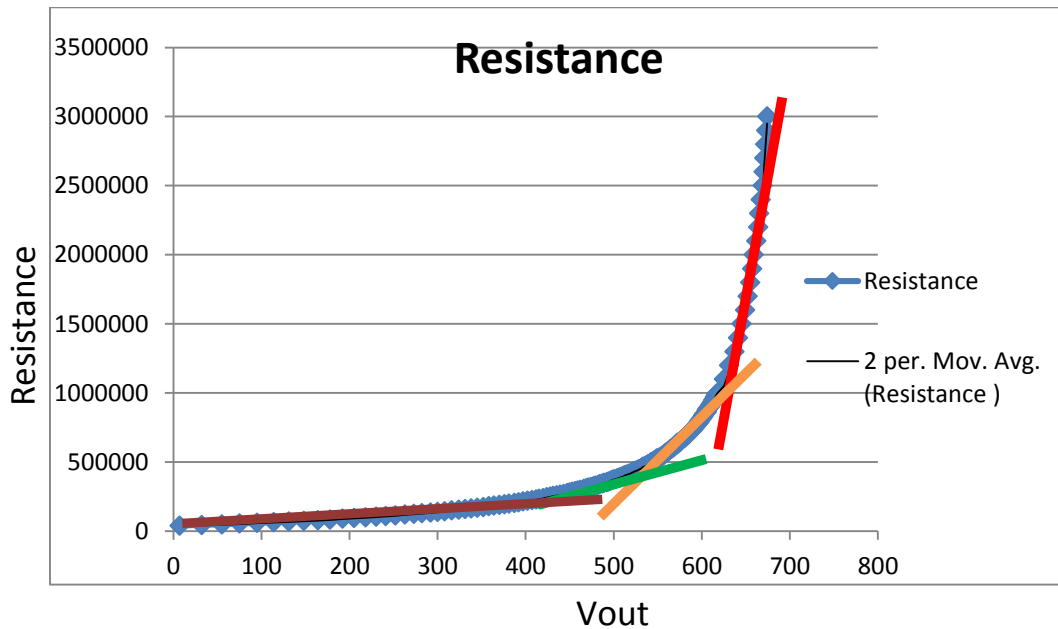
Figure 3.33: Resistance vs. Voltage graph (Tangent lines overlaid)

61

Now that we were able to convert our readings from voltage to microSiemens, the next step was to read the data and store for further analysis. We used an Arduino Uno microcontroller board to record the data. The data was internally stored on a PC with frequency 40 samples/sec. Figure 3.34 shows a few entries of the data.

Date	time stamp	µS
6/25/2013	6:24:58 PM	1.14
6/25/2013	6:24:58 PM	1.05
6/25/2013	6:24:58 PM	1.15
6/25/2013	6:24:58 PM	1.05
6/25/2013	6:24:58 PM	1.15
6/25/2013	6:24:58 PM	1.05
6/25/2013	6:24:58 PM	1.15

Figure 3.34: Samples of data entries

GSR sensor size reduction

The GSR was originally constructed with an LM 324 op-amp as seen in Figure 3.17. This was the original prototype of the sensor and after validating that the sensor works we decided to drastically reduce the size of the senor so that it can be small enough to be embedded into a wrist band. Therefore, the design was reduced by employing surface mount devices which minimizes the footprint to 17mm x 17mm as seen in Figure 4.35. We employed the same PCB software as the PPG sensor and used CadSoft EagleCAD PCB design software.

Figure 3.35: GSR manufactured to the PCB (printed circuit board) stage using CadSoft EAGLE - solder mask and silk screen top side layout

SKT Model

Skin Temperature Sensor

Environmental temperature can influence a person's cardiac and sympathetic response [8]. For this reason we decided to add a Skin Temperature sensor (SKT) in our suite of sensors. For temperature sensor, we used Texas Instruments LM35 precision centigrade temperature sensor. This sensor is ideal because it has an advantage over linear temperature sensors calibrated in Kelvin, since the user is not required to subtract a large constant voltage from its output to obtain convenient centigrade scaling.

This analog temperature sensor is essentially a chip that tells you what the ambient temperature is. It has a 0.1 degree precision and the sensor may be used in temperatures ranging from +32 to +212 °F which is more than adequate for skin temperature. The sensor uses a solid-state technique to determine the temperature. As the temperature increases, the voltage across the diode increases at a known rate. Technically speaking, this is actually a voltage drop between the base and the emitter, the V_{be} of the transistor. By precise amplification of the voltage change, you can easily generate an analog signal that is proportional to the temperature. The temperature sensor has no moving parts and it is for this reason that the sensor is precise, never wears out, does not need calibration, works under many environmental conditions, and it is consistent within its readings. It is also very inexpensive and easy to use.

Hardware Design

The initial design of the sensor was done in a breadboard. The sensor comes in a "TO-92" package which means the chip is housed in a plastic hemi-cylinder with three legs. As seen in Figure 3.36. In order to freely use the sensor, we solder the pins to connect to long wires.

The sensor uses anywhere between 2.7V and 5.5V as the power supply. We decided to use 5V as our power supply. An Arduino Uno circuit board was used to connect to the LM35 Celsius Temperature sensor. We simply connect the LM35 to +5V and Gnd and the output to Arduino Analog Input. As shown in Figure 3.37 and Figure 3.38.

Typical Applications:
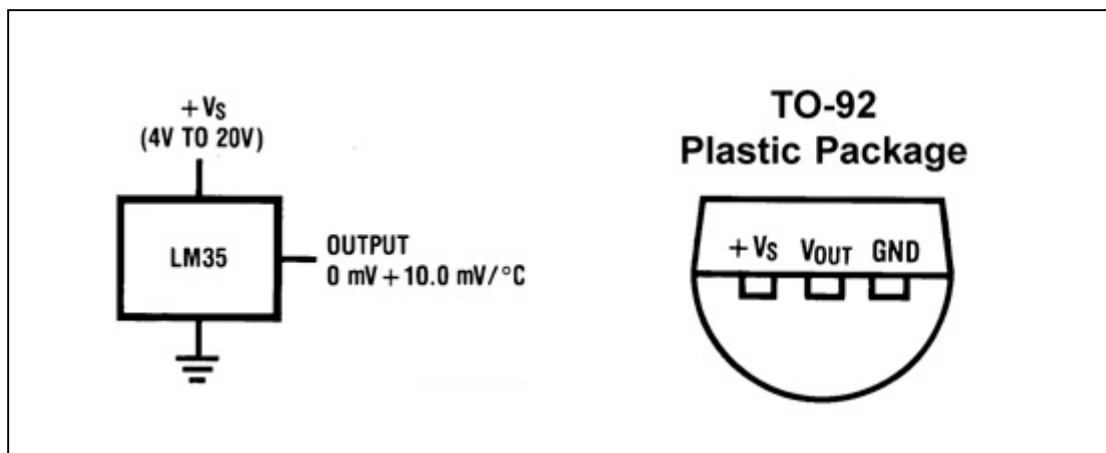
Figure 3.36: Basic Centigrade Temperature Sensor

2.7-5.5V in

Ground

Analog voltage out
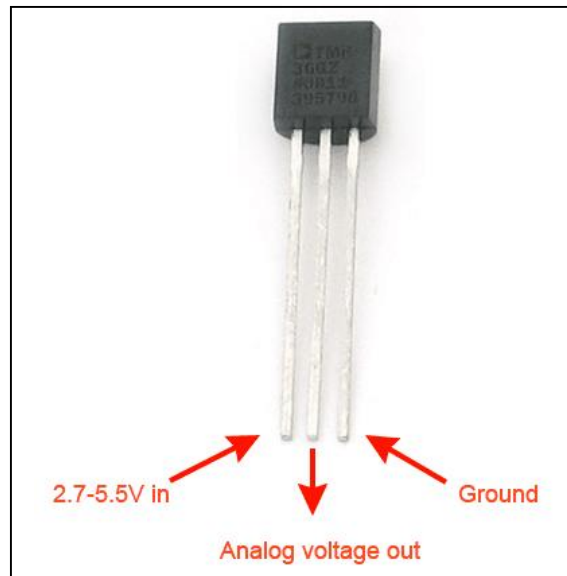
Figure 3.37: LM35 connections
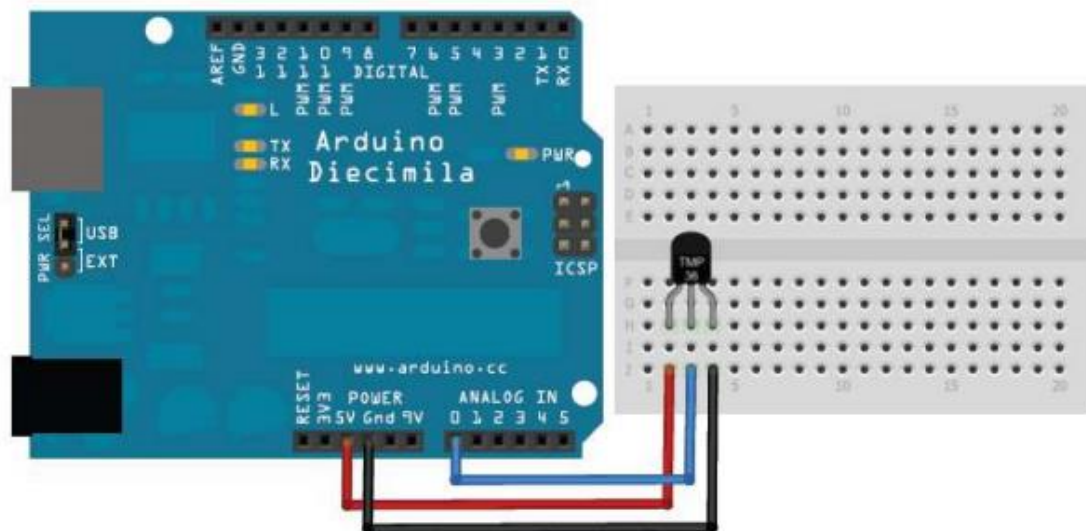
Figure 3.38: Illustration for Arduino Uno connected to LM35

SKT with Arduino program

Arduino Code

Arduino Uno open source environment was used to write the code and uploaded into the Arduino memory board. The development of this code is written in java and based on Processing, and other open source software.

The Arduino code is scanned every second to read the output from the LM35, than it converts the analog output into Celsius and sends the data to the computer via USB serial communication connection for further analysis. The code used to run the Arduino board is included in the Appendix.

Processing Code

In order to properly visualize the code, the Arduino code portion of this project is sent to Processing. Processing is a language and open source development environment similar to Arduino and it is intended for creating visual effects of the program. Processing was used to read the serial data from the Arduino board and display the temperature in a small graphical interface in both Celsius and Fahrenheit. A small rolling 100 data point graph was also integrated to better visualize the temperature. As shown in Figure 3.39.
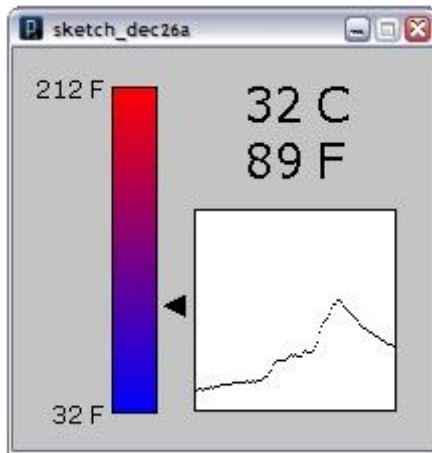
Figure 3.39: Temperature data visualized with processing

Testing of initial design

In order to test that our LM35 temperature sensor was working we decided to read the output at different temperature environments. When the system was powered we observed the output of our small graphical interface and we could see the output of about 25.5 °C or 77.9 °F. This is equivalent to room temperature. The temperature was increased when we applied our fingers to the sensor, heating up a little, the temperature reading increased to 28.8° C or about 83.8 °F. Our next test was to touch the sensor with an ice-cube. In order to avoid getting the sensor wet, we used a plastic bag wrapped around the ice-cube. When pressing the ice-cube against the sensor, the temperature decreased down to about 17.8° C or 64.04°F. This series of tests confirms that our sensor is fully operational. In the later chapter we will validate our sensor with a professional BIOPAC system [9].

SKT Size Reduction

The SKT was originally constructed with an LM35 Precision Centigrade Temperature Sensor as seen in Figure 4.8. The original sensor was packaged as a "TO-92" which means the chip is housed in a plastic hemi-cylinder with three legs and intended for a breadboard. We drastically reduced the size of the sensor by using the same chip packages in an "8-SOIC" (0.154" 3.90mm width) chip. Therefore, the design was reduced by employing surface mount chip which minimizes the footprint to 14mm x 10mm as seen in Figure 3.40. We employed the same PCB software as the PPG and GSR sensors and used CadSoft EagleCAD PCB design software.
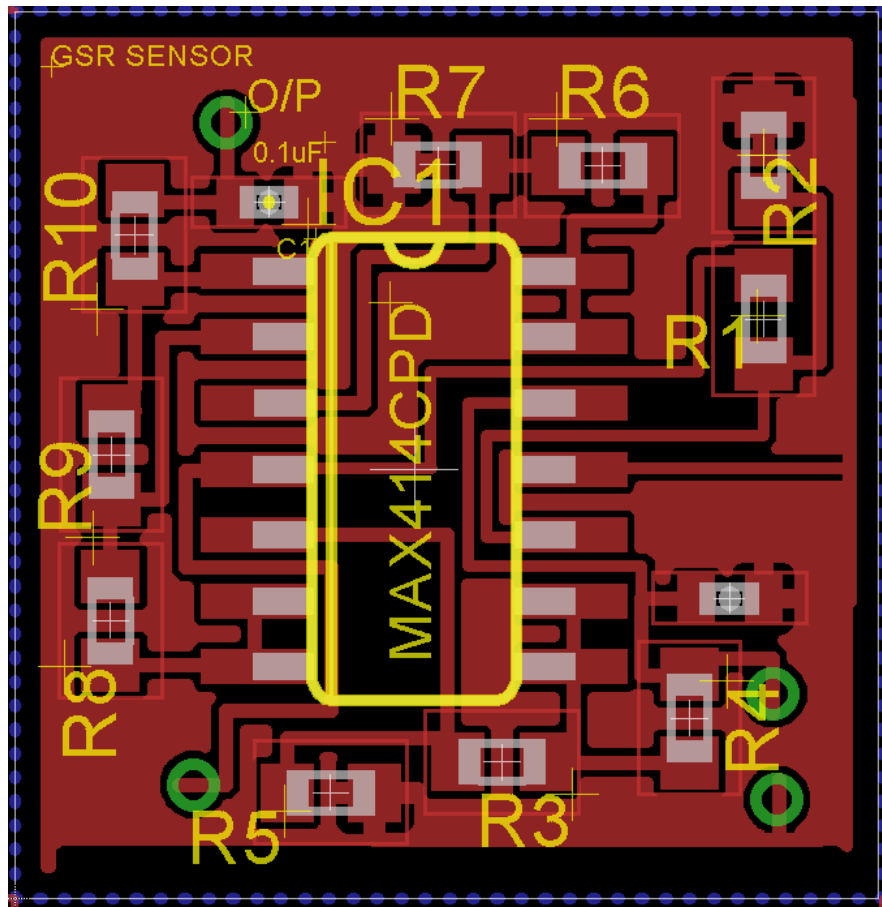

Figure 3.40: SKT manufactured to the PCB (printed circuit board) stage using CadSoft EAGLE - solder mask and silk screen top side layout

Bluetooth Serial Port

Bluetooth is a universal radio interference that enables various devices, including mobile phones, alarm systems, appliances, sensors… etc., to communicate wirelessly through a short range radio connection. The standard for Bluetooth is described in detail in the Bluetooth Specification and maintained[1] by the *Bluetooth Special Interest Group (SIG).* The development of this technology was instrumental in eliminating the requirement for wired connections, and eased the communication between devices, as well as enabling the formation of personal networks. Bluetooth has allowed the development of many applications because of its unique ubiquitous connectivity. It is for this reason that the recent advances in miniature devices, as well as Bluetooth ubiquitous computing, have fostered a dramatic growth of interest for non-intrusive wearable technology.

A Bluetooth device uses a license-free frequency band at 2.45 GHz. This band is known as the Industrial-Scientific-Medical (ISM) band and its range is from 2.4 GHz to 2.4835 GHz. This band is free and is often used by cordless phones, Bluetooth avoids interference by using a frequency-hopping spread-spectrum. The communication range is what differentiates the Bluetooth radios. The communication range varies from 1 meter for class 3 to 100 meter for class 1. Class 2 is the most common class and its range is at 10 meters. Data rates is also

---

[1] available at https://www.bluetooth.org/Technical/Speci_cations/adopted.htm

varied depending on the class and it ranges from 1Mbps to 24Mps. The network

of Bluetooth radios consists of two different types of devices: a slave and a master.

Bluetooth is a packet-based protocol with a master-slave structure. Each Bluetooth

device has the ability to be either a slave or a master or both at the same time. A

Bluetooth network consists of small subnets or piconets. The term "piconet" is

used to refer to the network formed by one device and all devices found within its

range. Up to 10 piconets can coexist within a single coverage area. In each piconet,

there is only one master and up to 7 slaves. The master is the "brain" of the

communication and all the communication through the slaves goes through the

master. A scatternet is formed when two or more piconets are connected and the

connection is formed when piconets have a device in common. The protocol stack

of Bluetooth is shown in Figure 3.41 and the details are given in (Labiod et al.,

2007).
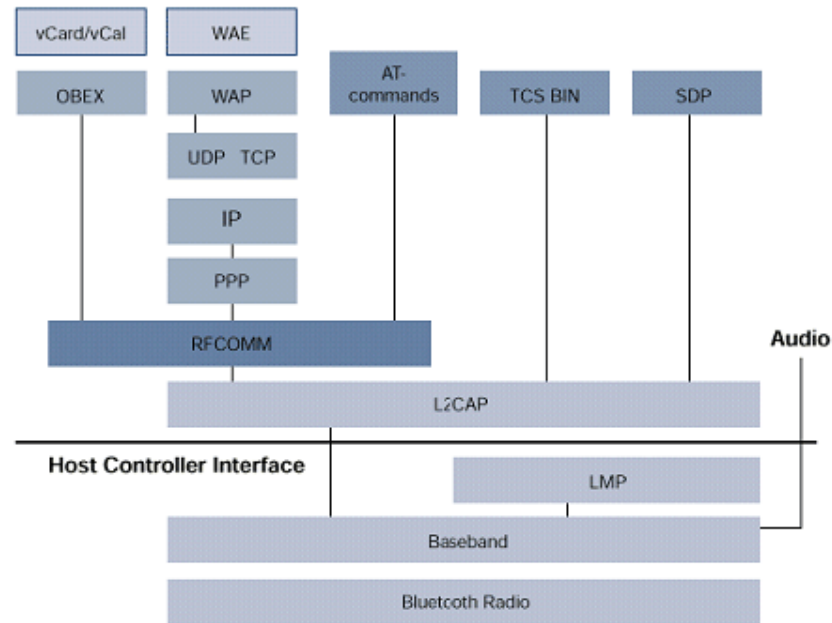
Figure 3.41: General Bluetooth protocol stack

For the purpose of our developed suite of sensors we decided to transfer our data using Bluetooth® communication protocol. From many Bluetooth devices we used a Bluetooth Mate Gold RN-41 module due to its performance ratio. This module is integrated with an antenna and a processor with 512 KB of flash memory. We chose this Bluetooth module because it is designed specifically to work with Arduino Uno's. These modems work as a serial (RX/TX) pipe, and are a great wireless replacement for serial cables. Any serial stream from 2400 to 115200bps can be passed seamlessly from the PC to our sensors. During normal operation, the module streams PPG, GSR, and SKT outputs received from the Arduino Uno microcontroller over the wireless link to a nearby Bluetooth-equipped computer. Being a class 1 module, the RN-41 module is specified to maintain

connections up to 100 meters. This Bluetooth module is also bi-directional meaning it can also send information received over the wireless link to the micro-controller.
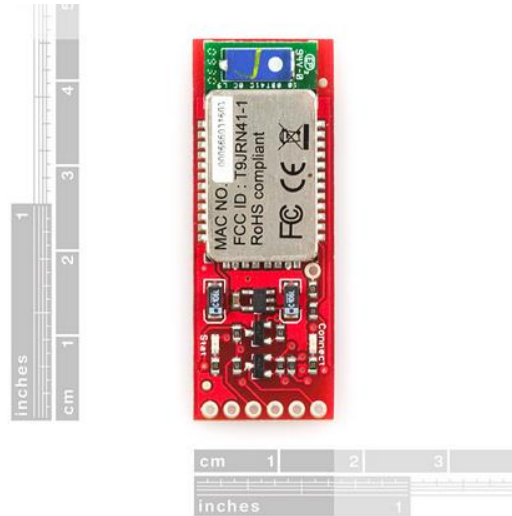
Figure 3.42: Bluetooth Mate Gold RN-41 Module

The RN-41 module has the following features:

- Bluetooth standard Ver. 2.0 + EDR support

- Low power (30mA connected, , < 10mA sniff mode)

- Class 1 high power amplifier with on board ceramic RF chip antenna.

- Operating Voltage: 3.3V – 6V

- Support for up to seven slaves

- Interface: USB, UART, PCM

- SPP firmware with AT command sets

The RN-41 module can be programmed by using simple ASCI command language which is similar to the industry standard Hayes AT protocol.

Function of Bluetooth Mate Gold RN-41

After all of the outputs from the suite of sensors are converted to ADC (Analog to Digital), the Arduino Uno micro-controller will send data to the RX pin of the module and the data from all of the sensors will be automatically sent to the PC by this module.

Summary of Design Consideration

After the successful design and tests of all the physiological sensors, it was time to combine all the sensors into one system. At this stage all of the sensors had been manufactured to the PCB (printed circuit board) stage though CadSoft EAGLE as shown in Figure 3.43. It was time to once again use the free EAGLE software to meticulously put together all the sensors into one system.
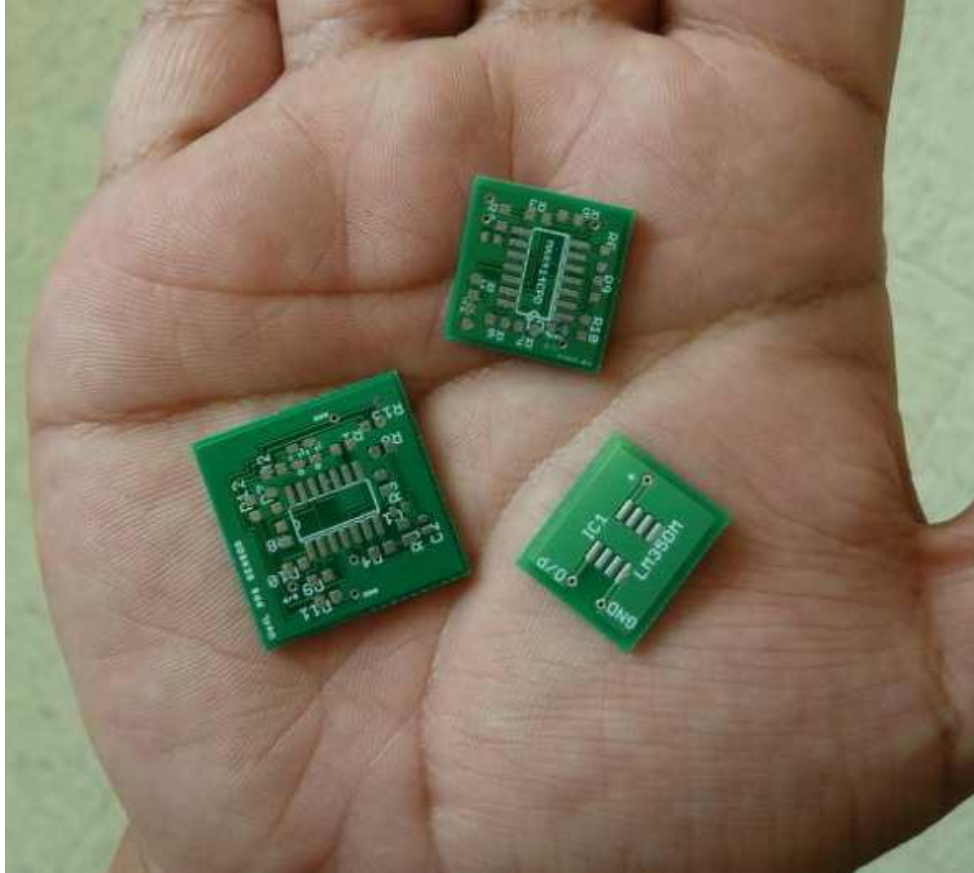
Figure 3.43: Sensor suite of the three independent physiological sensors (Picture taken by author)

In the final design of the physiological sensor system, many design considerations and techniques were combined in order to reduce the size of the system to its maximum capacity. The re-arrangement of different components and the geometry re-design of the circuit board, significantly reduced the size of the device. These design factors helped make the physiological suite of sensors smaller and thereby less obtrusive to the wearer. The final design with all three sensors was reduced to an impressive 25mm x 31mm. This time the sensors were designed to have two solder mask and silk screen layers. The top layer is reserved for the

GSR and the PPG sensor. The bottom layer is reserved for the SKT sensor. The reason why the SKT is at the bottom is because it will be touching the skin to sense the skin temperature. Figure 3.44 and Figure 3.45 show the top and bottom solder mask and silk screen layout.
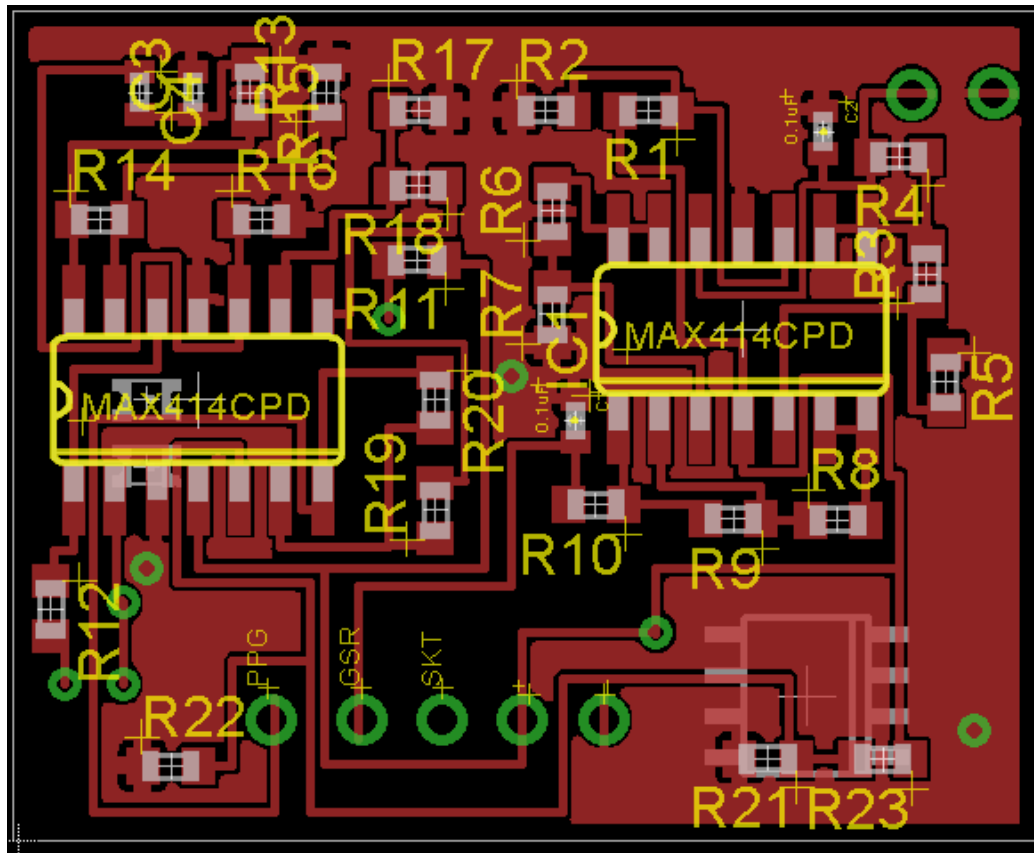
Figure 3.44: PPG, GSR, and SKT manufactured to the PCB (printed circuit board) stage using CadSoft EAGLE - solder mask and silk screen top side layout

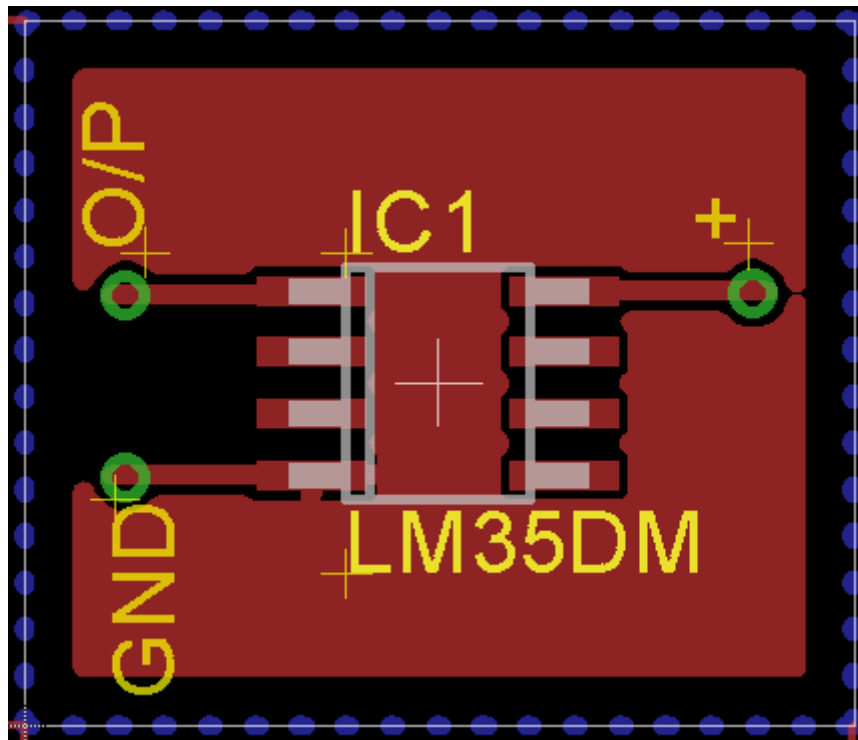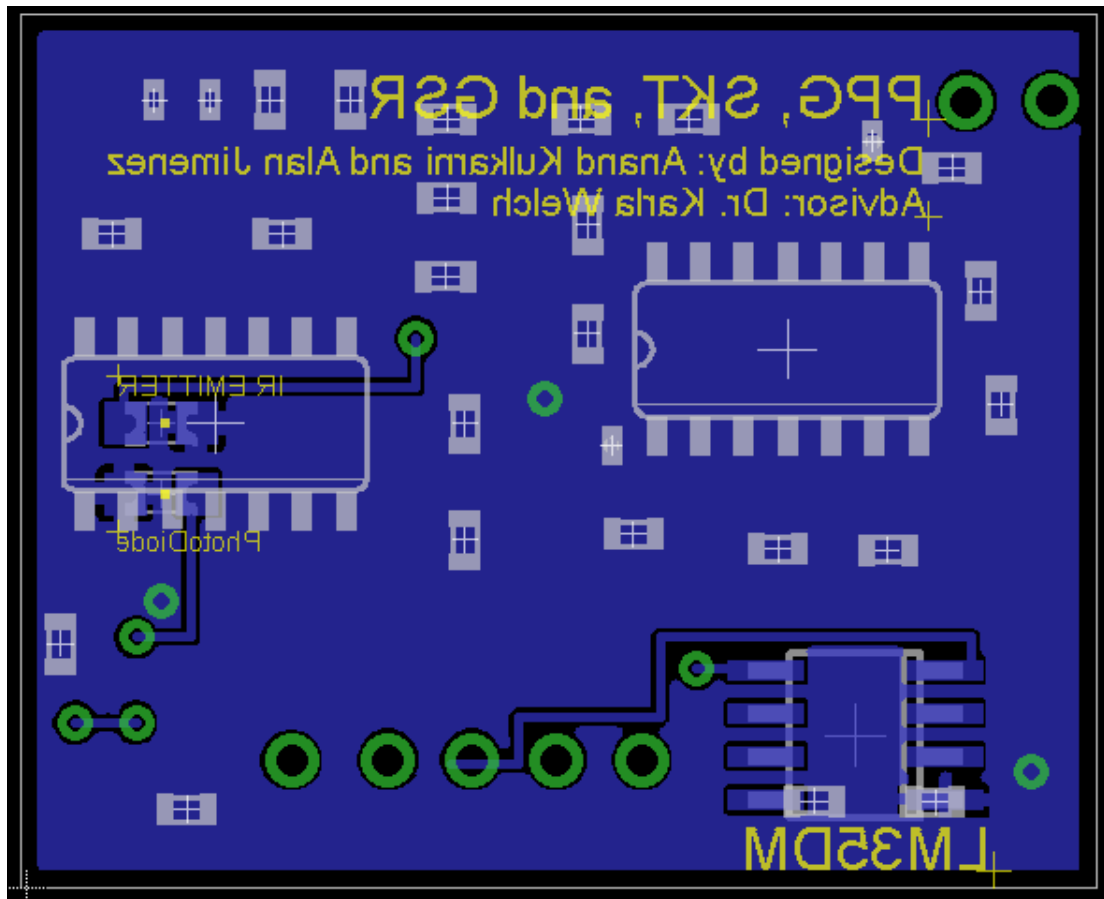Figure 3.45: PPG, GSR, and SKT manufactured to the PCB (printed circuit board) stage using CadSoft EAGLE - solder mask and silk screen bottom side layout
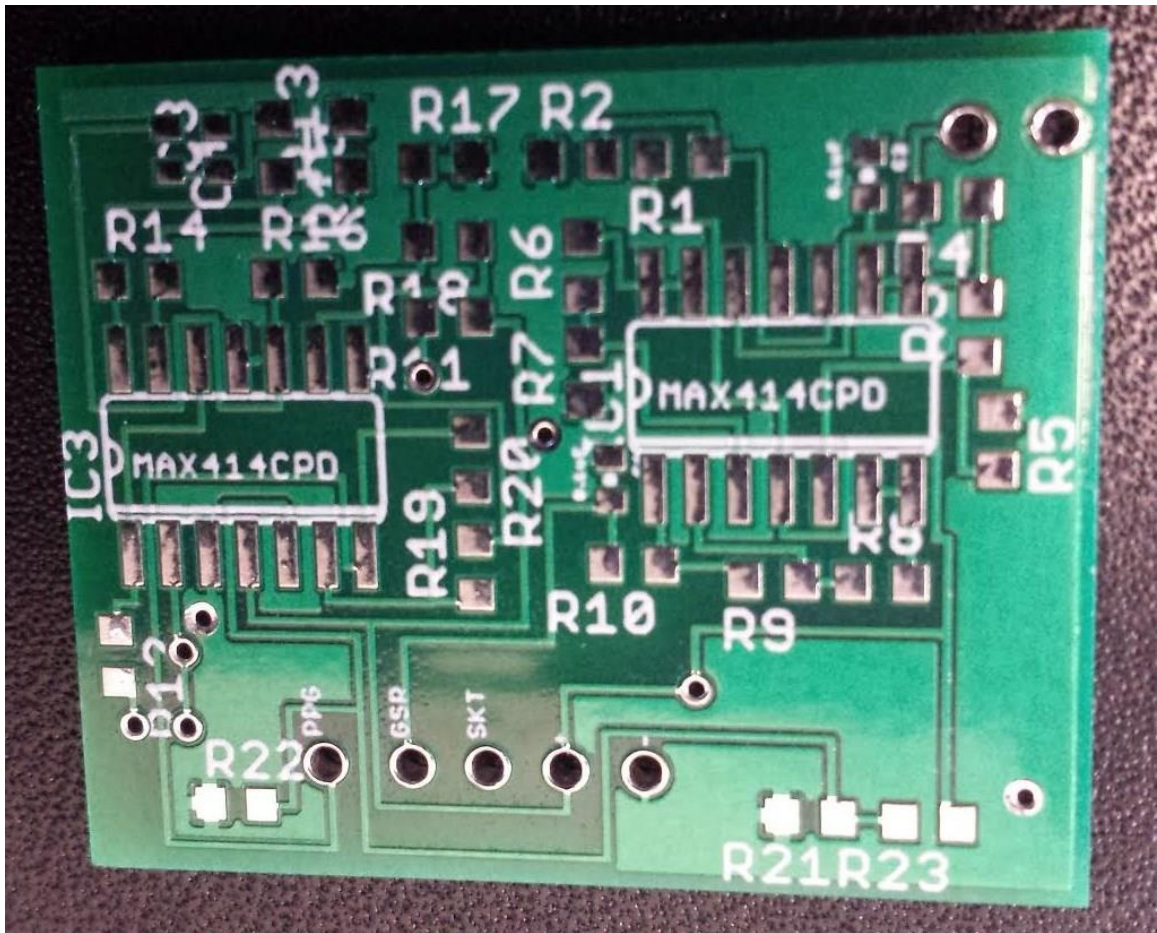
Figure 3.46: PPG, GSR, and SKT suite of sensors combined to one PCB (photo taken by author)

Battery

All of the individual components of the physiological sensors require a 3.3 V supply. In order to provide power to the system, a 3.3 V, 285 mAh CR2430 coin cell lithium battery was used. The coin cell battery was chosen due to its small slim design, low profile, and its high capacity.

Sensor Housing

In order to accommodate all of the hardware of the previously mentioned sensors, we used a 3-D printer to print our physiological sensor housing. A 3D printer is the process of making three-dimensional solid objects from digital designs. It is now more affordable to the individual customer. Recently it has acquired a lot of interest in the technology world. Chuck Hull invented and patented stereolithography (also known as solid imaging) in the mid-1980s, when he founded 3D Systems, Inc. Since Chuck's discovery, advances in the technology have been extraordinary, including the size of printers and the material they can use [23]. 3D printer owners can choose between two types of plastic: acrylonitrile butadiene styrene (ABS) or polylactic acid (PLA). For our housing, we decided to use ABS, which is used to make Legos®. ABS is a strong, durable production-grade thermoplastic and it is an ideal material for conceptual prototyping. ABS is considered superior than PLA for its hardness, gloss, toughness, and electrical insulation properties. Furthermore, ABS is quite durable, which provides a strong,

protective housing. The first step was to design the housing using 3D CAD design software. We decided to use SolidWorks. SolidWorks is a 3D mechanical CAD program free of charge by our educational institution.

The housing was designed to be ergonomically as possible. The housing was designed as a small thin square box. The housing itself has one cavity for all of the components including, PPG, SGR, SKT, Femtoduino, and 3.3V lithium ion coin cell battery. There is also a single top port with all the sensor outputs, power and ground. This port also includes the inputs for connecting the electrodes to the sensor board. There is a housing lid that attaches to the side for protection and easy access of the sensors. At the bottom of the housing there is two small openings. One of the openings is for the SKT sensor that needs to be exposed to the skin to read your skin temperature. The other opening is for the PPG sensor that needs to be exposed to the lower palmar surface of your hand to detect blood volume changes in the microvascular bed of tissue. The initial prototype housing is (34) x H (28) L x (6) W. Figure 3.47 shows the 3D rending of our home-made housing while Figure 3.48 shows the 3D rending of our housing with the lid on and off.
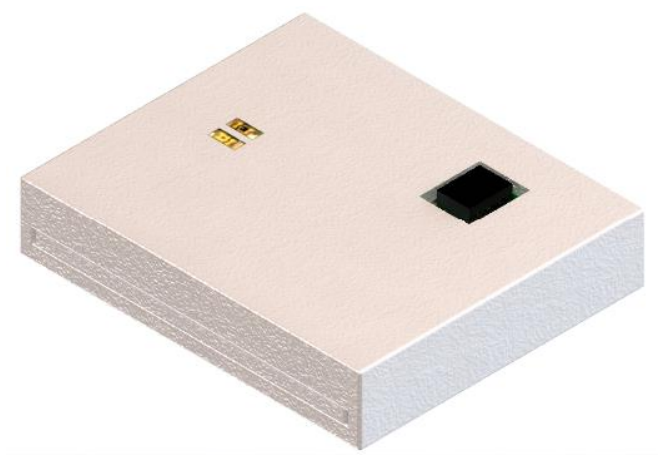
Figure 3.47: Top and bottom 3D rendering of housing
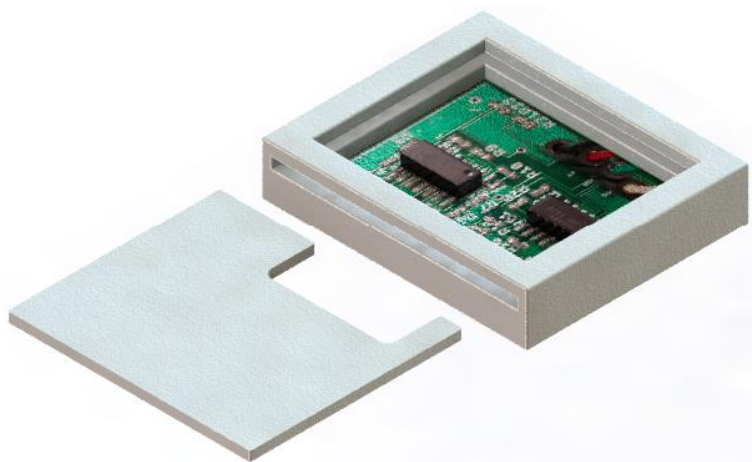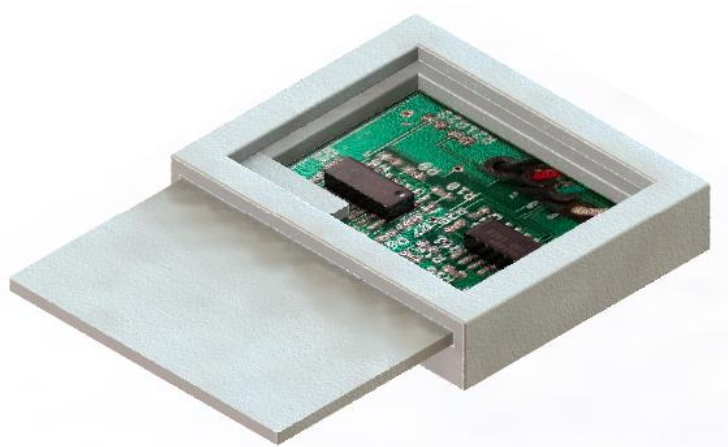
Figure 3.48: 3D rendering of sensor with lid on and off

Chapter 4

Performance Evaluation

This research has been improved through various test procedures and techniques. Initially, basic features of each sensor were tested to ensure that each component worked and then as testing progressed, modifications or adjustments were made to the circuits so they functioned well practically. The PPG, GSR, and SKT sensors used in our physiological suite of sensors was validated and tested using the commercial off the shelf Biopac MP150 System by the Biopac Systems [9].

In order to validate our physiological sensors we will be using discriminant properties called features that are derived from the signals using various signal-processing techniques, such as moving average windows, thresholding, and peak detection. Data vectors are made of twelve normalized physiological features, as listed in Table 1, extracted from the three signals. For example, PPG_PeakMax is the maximum amplitude of detected PPG peaks, measured in μV, which showed significant affective responses in the PI's previous work [24]. The PPG signal measures changes in the volume of blood associated with the blood volume pulse cycle and provides an index of the relative constriction versus dilation of blood vessels. Also derived from the PPG signal is the mean amplitude of peaks, and the mean and standard deviation (SD) of the time interval between peaks, or inter beat interval (IBI).
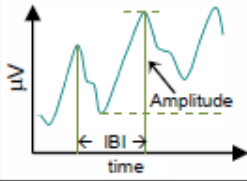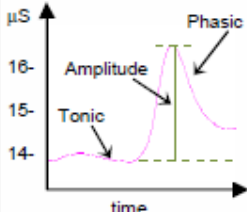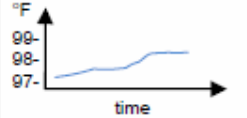
Physiological Signal	Features Derived	Typical Graph	Label	Units
Photoplethysmogram (PPG)	mean amplitude of the peak values of PPG		PPG_PeakMean	microvolt (µV)
	maximum amplitude of the peak values of PPG		PPG_PeakMax	µV
	mean of IBI of PPG		IBI_PPG_Mean	ms
	SD of IBI of PPG		IBI_PPG_SD	ms
Galvanic Skin Response (GSR)	mean tonic activity level		TonicMean	microSiemens (µS)
	slope of tonic activity		TonicSlope	µS/s
	mean amplitude of GSR (phasic activity)		PhasicMean	µS
	maximum amplitude of GSR (phasic activity)		PhasicMax	µS
	rate of phasic activity		PhasicRate	peaks/min
Skin Temperature (SKT)	mean temperature		TempMean	degree Fahrenheit (°F)
	slope of temperature		TempSlope	°F/s
	SD of temperature		TempSD	°F

Table 4.1: Physiological Features

Derived from the PPG signal is the mean amplitude of peaks, and the mean and standard deviation (SD) of the time interval between peaks, or inter beat interval (IBI). GSR phasic activity represents an event-related rapid increase in skin conductance similar to a peak, whereas GSR tonic activity refers to the ongoing level of skin conductance in the absence of any particular discrete environmental events. The GSR phasic peak detection algorithm uses the following heuristics: (1) amplitude greater than 0.05mS, (2) slope of the rise to the peak greater than 0.05mS/min, and (3) rise time greater than 0.25s. Once the phasic responses are identified, the rate of the responses (PhasicRate) and the mean and maximum phasic amplitude are determined. All the signal points that were not included in the response constitute the tonic part of the GSR signal. Features derived from the GSR signal are the mean and SD of tonic activity, the mean and

maximum amplitude of phasic activity, and the phasic rate in peaks per minute. The SKT signal is filtered to remove high-frequency noise and then the mean, SD, and slope are calculated as features. A more-extensive description of the signal processing, feature extraction routines, and significance of these physiological features for indicating an effective response can be found in [25] and [26]. This chapter describes the tests used to evaluate and validate each individual systems. After this, results for each of the tests are presented.

PPG

After building the PPG circuit it was initially tested by using an oscilloscope and capturing the output of the sensor. The sensor produces an accurate PPG waveform, including the diacritic notch as seen in Figure 3.18. This time we tested the circuit again with the implementation of all the sensors in an updated printed circuit board and a complete new system that includes a micro-controller and Bluetooth radio. The sensors were embedded in a wristband for minimal intrusiveness for the user. The voltage input into the circuit was 3.3 V for the positive terminal and the negative terminal was grounded. The implementation of the new housing was instrumental because it has a small opening at the bottom. On one side, an LED (light emitting diode) is used to convert electrical energy into light energy and on its opposite side, the light-to-voltage sensor is placed to measure the amount of light transmitted through. When the palm surface of your hand is exposed to the small opening, the LED shines on one side and the

transmitted light is captured on the other side of the finger. The housing is essential because it reduces interference from other light source.

Then, PPG circuit is tested on a subject. The next step was to use Femtoduino microcontroller board to record the data from the in-house PPG sensor. The data was read using an analog input of the microcontroller which mapped the sensor output from 0 to 1024. This reading was later converted back to the voltage by multiplying the reading by the reference voltage and dividing it by 1024. The data was wirelessly sent to a PC file directory with frequency 144 samples/sec. Figure 4.1 shows a few entries of the data.

millis	stamp	datetime	reading	voltage
71	1322612932	"2011/11/30 0:28:52"	65	0.19
78	1322612932	"2011/11/30 0:28:52"	62	0.18
84	1322612932	"2011/11/30 0:28:52"	64	0.19
91	1322612932	"2011/11/30 0:28:52"	55	0.16
97	1322612932	"2011/11/30 0:28:52"	61	0.18
103	1322612932	"2011/11/30 0:28:52"	67	0.2
110	1322612932	"2011/11/30 0:28:52"	49	0.14
116	1322612932	"2011/11/30 0:28:52"	57	0.17
122	1322612932	"2011/11/30 0:28:52"	52	0.15
129	1322612932	"2011/11/30 0:28:52"	52	0.15
142	1322612932	"2011/11/30 0:28:52"	47	0.14

Figure 4.1: Sample data entries

The next step was to use MATLAB to plot the PPG data that was stored on the PC. Figure 4.2 shows the plot of the PPG data. Most of the high frequency noise was already removed by hardware RC filter but some noise is still seen with the data.
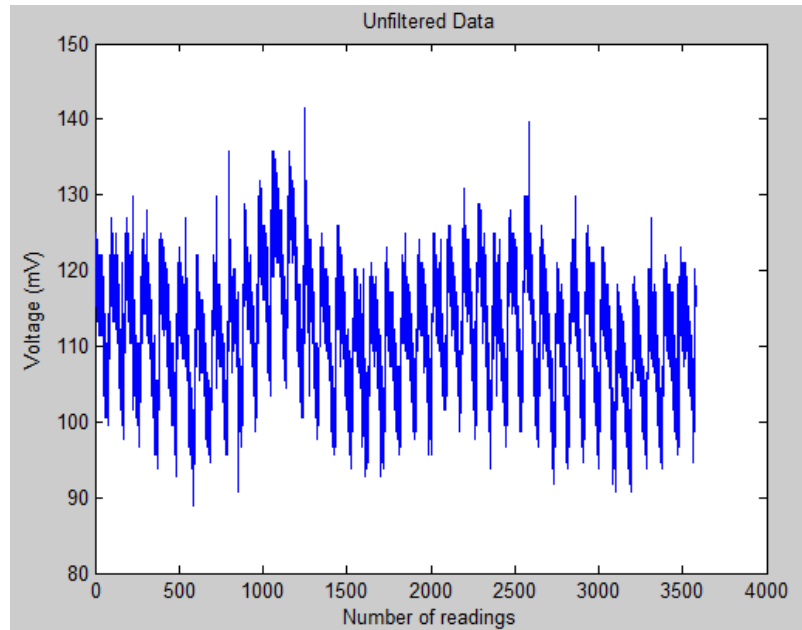
Figure 4.2: PPG data from PC file directory

Figure 4.3 shows the filtered data, we used 10 point moving average filter to remove the noise present in the data.
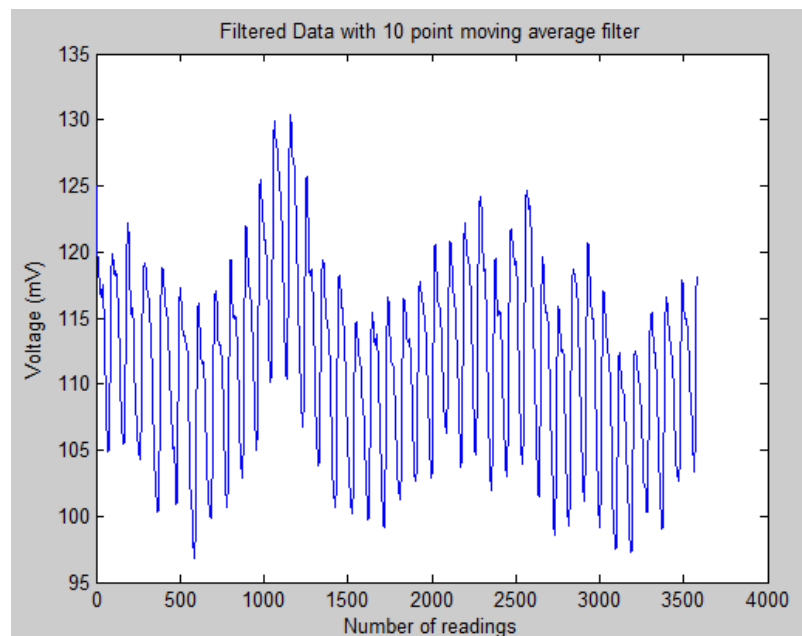
Figure 4.3: Filtered PPG data

Once the data was filtered we used peak detection algorithm to find number of peaks, which corresponds to heart beats present in the data. Following data shows inter-bit interval, peak time, PPG mean and PPG max obtained from the MATLAB code. The next step is to compare this data with a commercially available FDA approved BIOPAC system. The features extracted from the signals, such as mean interbeat interval (IBI) from PPG, match between BIOPAC and the in-house designed sensor. The validation of the in-house PPG sensor has been successful.

MATLAB Output

[ibi,peak_time,ppg_mean,ppg_max] = peak_detection(data)

ibi = Columns 1 through 14

  107 171 171 122 121 43 42 131 131 62 61 72 72 40

 Columns 15 through 27

  39 129 128 134 134 138 138 151 151 124 123 142 141

peak_time = Columns 1 through 7

    287   394   565   736   858   979   1022

 Columns 8 through 14

| 1064 | 1195 | 1326 | 1388 | 1449 | 1521 | 1593 |

Columns 15 through 21

| 1633 | 1672 | 1801 | 1929 | 2063 | 2197 | 2335 |

Columns 22 through 28

| 2473 | 2624 | 2775 | 2899 | 3022 | 3164 | 3305 |

Columns 22 through 28

| 2132 | 2211 | 2289 | 2345 | 2401 | 2437 | 2473 |

ppg_mean =   0.0148

ppg_max =   0.0286

GSR

To evaluate and validate the discriminant features of our in-house GSR sensor module, we correlated the measured GSR signals by the proposed device with the Biopac system on the dominant finger as shown in figure 1.1. Our in-house GSR sensor module enclosed in its new housing is embedded in a wrist band with two Ag/AgCl electrodes touching the palmar surface of the dominant hand. There was no pretreatment of recording sites and there was conductive gel applied to the electrodes.

A single subject was recorded while wearing the in-house GSR designed sensors and Biopac GSR sensor during 1 minute of resting followed by 3 minutes playing Pong on a computer and finishing with another 1 minute of resting. During the baseline, task, and recovery conditions the sensors have proven to be highly correlated with a Pearson correlation coefficient of 0.9752. The clocks for the proposed sensor and the Biopac system were synchronized with a time stamp prior to the start of each experiment, and the time was recorded at the beginning and end of each condition. The sampling rate for both signals was fixed at 40Hz. Figure 4.4 shows the plot (conductivity vs. time) of the Biopac GSR sensor. Figure 4.5 shows the same results for the in-house GSR sensor. We can see many similar features such as the mean tonic activity level, slope of tonic activity, phasic activity and rate of phasic activity. The unit of measure for conductivity is microSiemens (µS). The MATLAB code for the validation of the in-house GSR sensor can be found in the appendix.
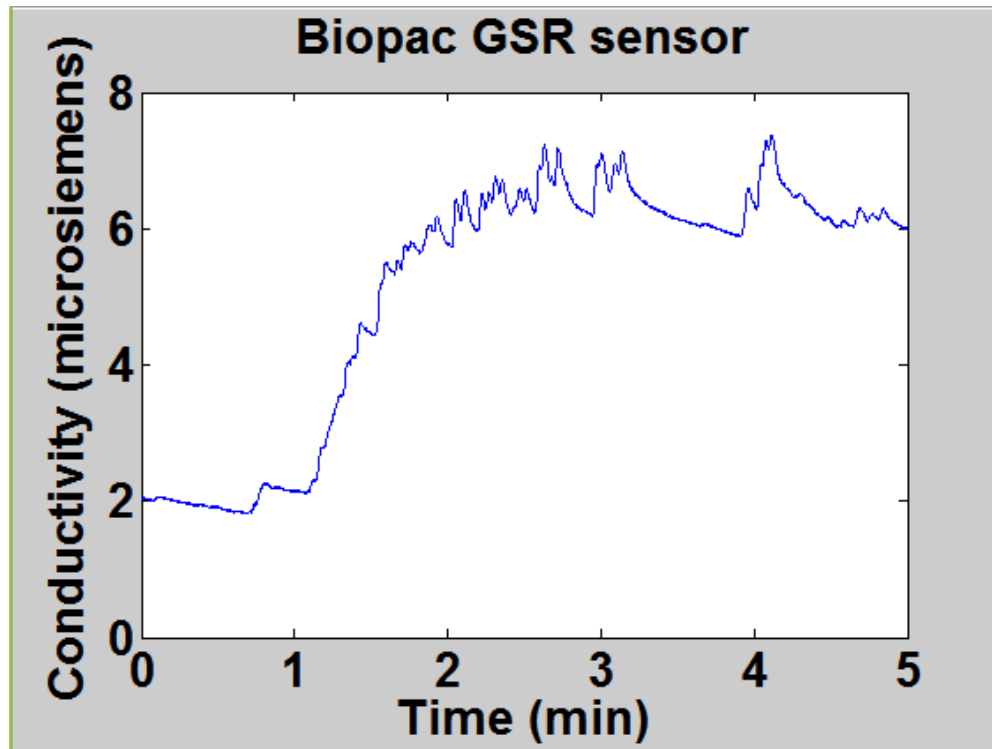
Figure 4.4: Biopac GSR sensor recorded at 40Hz with 0.975 Pearson correlation
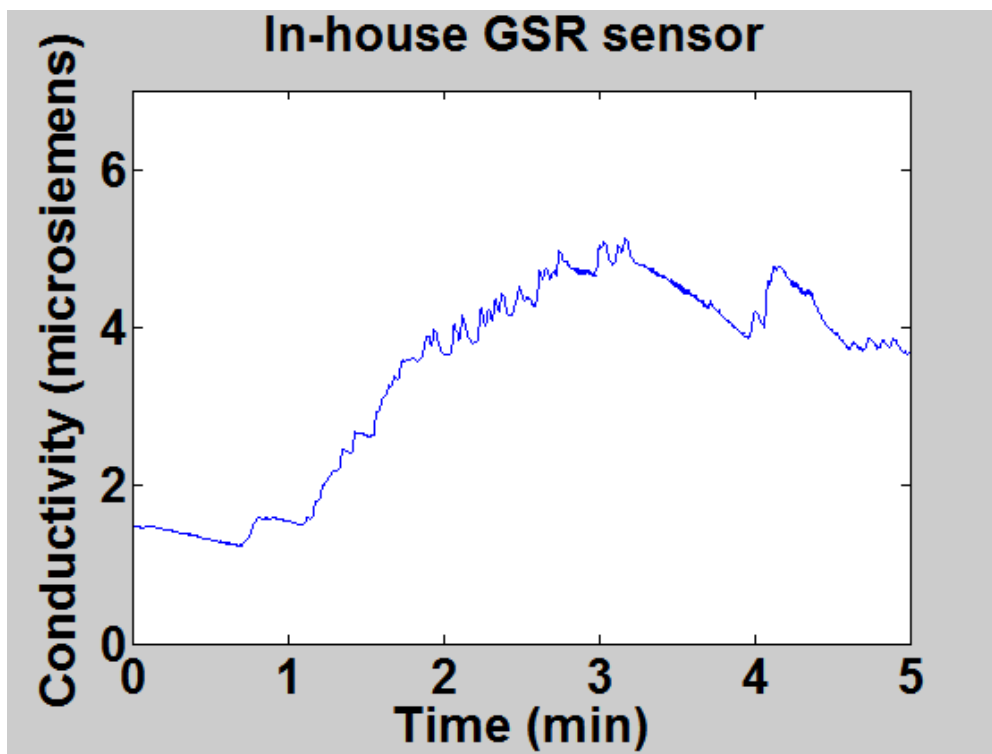
Figure 4.5: In-House GSR sensor recorded at 40Hz with 0.975 Pearson correlation

For our validation purposes we were able to collect data at 40Hz. Previous lab experiments using the Biopac system used a collection rate of 1000Hz. This rate would severely limit the battery life of our wireless sensor transmitting data as part of a feedback system. Furthermore, this rate is not required, as Zhai and Barreto (2006) were able to collect reliable PPG, GSR, and SKT readings at 360Hz from a commercial tethered system but did not have the battery constraints of a wireless system. Since our system has battery constraints of 3.3V, we were happy with the 40Hz results because it falls within the frequency spectrum of the sensor. Redesigns and future improvements will look for a microcontroller most appropriate for handling all three signals, and will work up sending data at 300Hz for over 6h on a 6V battery to a Bluetooth receiver (i.e., on-board robot computer or smartphone) for processing. Figure 4.6 shows a simple flow chart of the process that was stepped through when validating the GSR sensor.
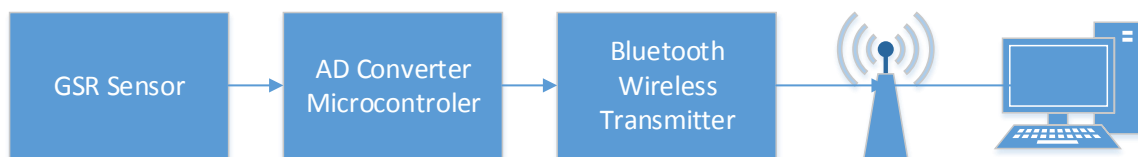
Figure 4.6: Steps to validate the GSR signal

SKT

To evaluate and validate the SKT sensor we used the FDA approved Biopac MP150 system as the standard. The SKT sensor from Biopac was placed on the dominant thumb finger of the right hand while the in-house wearable SKT sensor (wristband) was placed on the distal forearm with the sensor module exposed to the palmar surface of the hand. There was no pretreatment of the recording sites and no conductive gel was applied to the contact area. As explained earlier in chapter 4 under "Skin Temperature Sensor", we used Texas Instrument's LM35 precision centigrade temperature sensor with tolerance of +-0.1 degree precision. It can measure temperatures ranging from +32 to +212 °F and has a fast response time and low power dissipation, which makes it ideal for skin temperature measurement.

A single subject was recorded while wearing the in-house SKT sensor and the Biopac sensor during 1 minute. The clocks for the SKT sensor and the Biopac SKT sensor were synchronized prior to the start of the experiment. The time was recorded at the beginning and end of each experiment. The sampling frequency for both sensors was fixed at 32 Hz.
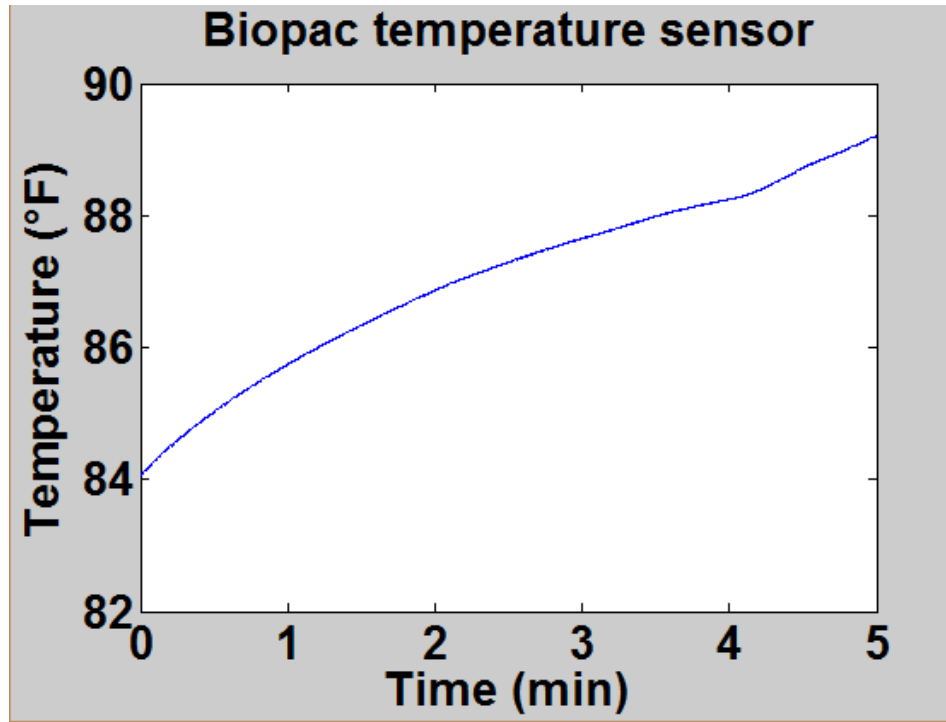
Figure 4.7: Temperature vs. Time of Biopac System temperature sensor
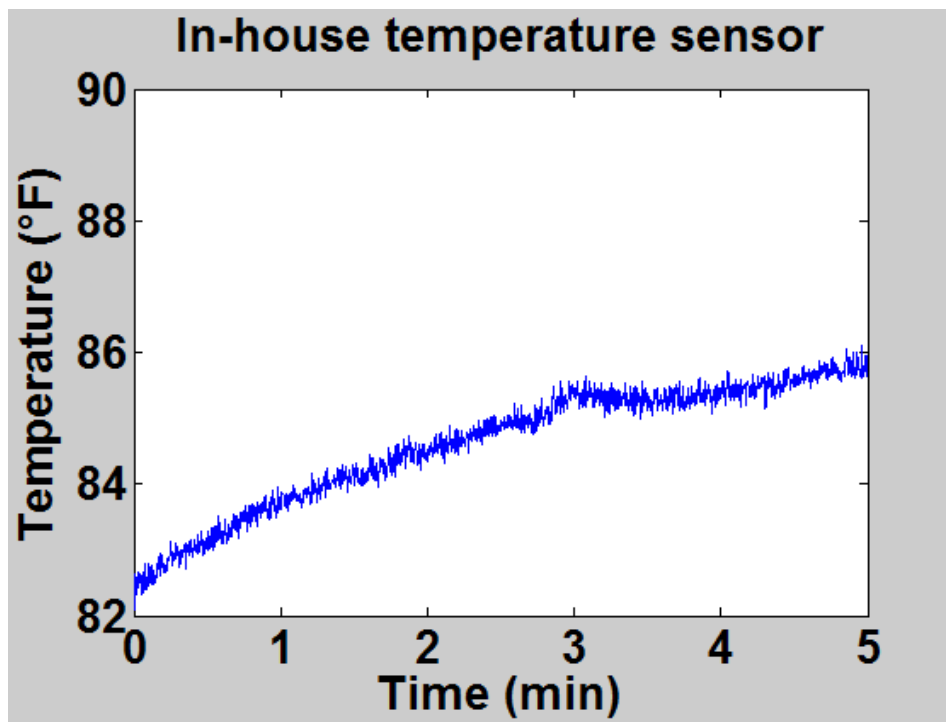
Figure 4.8: Temperature vs. Time of in-house temperature sensor

The above graphs show the result of the Biopac skin temperature sensor and the in-house SKT proposed sensor. To evaluate and validate the performance characteristics of our sensors, we used Pearson's coefficient correlation. Overall, recordings from the proposed SKT sensor and the Biopac SKT sensor were strongly correlated at 0.989. Using Table 4.1, we can use the discriminant features of mean temperature and slope of the temperature and we can clearly see that they are correlated. The MATLAB code for the plot comparisons can be found in the Appendix.

Cost Analysis of 3-in-1 (PPG, SKT, and GSR)

A cost analysis was done for the final suite of sensors. The total cost of our in house sensor came to be $108.137 as shown in Figure 4.9. The cost can certainly be reduced if you order more components by receiving a discount from the manufacturers.

No	Component description	Quantity	Price of individual item	Total price
1	Resistor	23	0.02	0.46
2	Capacitor	4	0.075	0.30
3	Max494 (Op-amp)	2	10.0	20.0
4	IR Emitter	1	0.969	0.969
5	Photodiode	1	0.836	0.836
6	GSR electrodes	1 pair	15.0	15.0
7	PCB printing	1	8.572	8.572
8	Bluetooth Module(RN-41)	1	35.0	35.0
9	Femtoduino	1	17.0	17.0
10	Miscellaneous $	-	-	10.0
			Total	$ 108.137

Table 4.2: Cost of individual In-House sensors

Cost Analysis of Different Systems
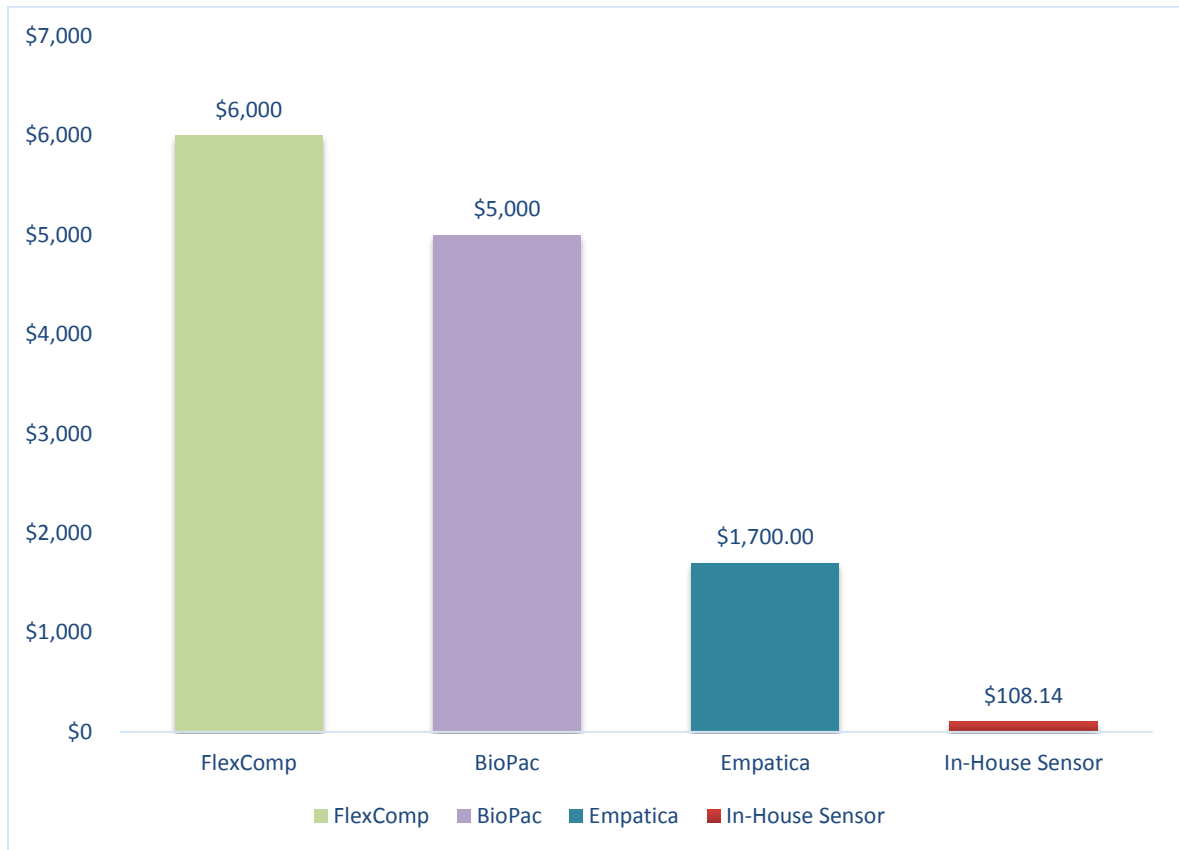
Figure 4.9: Graph comparing the price of different physiological systems to our in-house sensor

The In-House sensor is drastically cheaper when compared with commercially off the shelf systems. Empatica is the most recent company that has a physiological signal monitoring device on the market. The name of the sensor is E3 Wristband and it is developed by a diversified group of individuals focused on affective computing and data analytics. The bracelet worn device monitors physiological signals in real-time. This signals include PPG, GSR, and SKT. This sensor is very expensive with current price at $1700.

Chapter 5

Conclusion

In this thesis, the development of wearable physiological sensors and affect-sensitive feedback system has been described from concept to realization and testing. The objective of this thesis was to build a low power, low cost, reliable, non-intrusive, and non-invasive monitoring system that would accurately measure physiological signals. This aim includes building sensors to wireless transmit data on galvanic skin response, Photoplethysmogram, and skin temperature and conducting usability tests to determine collection sites that users find most comfortable. A reliable and continuous physiological signal monitoring system targeted towards children with ASD (Autism Spectrum Disorder) has been successfully built. The resulting system was also low in power and cost, non-invasive, and provided real time monitoring. The system developed is easy to use and provides accurate measurements when validated with an FDA approved Biopac system.

Perhaps the biggest achievement of this thesis is that the proposed developed sensors will aid researchers with current ongoing research to study specific sections of the population for which emotion mapping has not been fully explored, namely typically-developing children and children with autism. Current research is seeking to build an affect-sensitive architecture for interaction between a child and an intelligent system. The wireless capabilities of the proposed sensors serve as a valuable tool towards wireless human-interaction (HRI). With the use of the

proposed physiological sensors, researchers will be able to bridge that interaction by utilizing the monitoring system to analyze affective information gathered from the physiological data. The proposed thesis is important for the development of affect-sensitive HRI. Affective computing allows a computer to emulate the human ability to detect and respond to emotions and other affective phenomena for improved human-machine interaction. Most importantly, children with ASD will not have to use tethered sensors and as a result they will not be limited in the range of their activities when testing.

Summary

Given the scope of this thesis, the proposed sensors were successfully designed based on the objectives that were presented. The sensors first started at the prototype stage, each circuit block was tested to ensure functionality. Through the various test procedures and techniques, many parts of this sensors were modified and improved. As soon as the prototype considered working properly through lab testing, the sensors were drastically minimized in size by using surface mount devices on a printed circuit board. Using PCB software called Eagle CAD, many revisions were implemented, each revision implored modifications or adjustments that were strategically made to make the sensor smaller in dimensions. This was crucial in order to make the final printed circuit board with all three sensors small enough for an in-house designed housing to be embedded into a wrist band or glove.

The proposed physiological sensors were validated with a commercial off the shelf product called Biopac system by Biopac Systems. The MP150 Biopac system is capable of precise measurements of many physiological signals, but its big drawback is that the user is tethered to the system. One important feature of the proposed sensors is that it uses Bluetooth technology for data transferring providing an additional degree of flexibility when compared to the tethered Biopac system and other physiological systems. The use of Bluetooth technology and a small lithium ion coin cell battery size will allow portability and quick system integration.

Future Research

The proposed physiological sensors can be further improved. The size of the current printed circuit board can be further reduced using small surface mount devices. There is still room of improvement and using new design techniques and modifications will improve the size dimensions.

Another improvement can be made by using *Bluetooth Low Energy (BLE)*, which is an extension to the Bluetooth 4.0 standard. This new low energy Bluetooth radio got introduced by the *Bluetooth Special Interest Group (SIG)* in late 2009 [27] and is optimized specifically for small battery operated devices that require almost no power [28]. This Bluetooth module would be a perfect improvement for future research. The big advantage over our current classic Bluetooth is the much lesser power consumption of BLE devices, which is made

possible by the much more simplified device discovery and connection establishment process which it usually only sends small data packets every several seconds and going into sleep mode in between the sending and receiving.

An improvement of the current system housing can also be made. One of the improvements that can be made is the reduction of size. The current housing is small, but with the future size reduction of the current physiological sensor system the housing can be designed even smaller. The housing can also ergonomically improve to maximize comfort and minimize intrusiveness to the patient.

Finally, the proposed physiological sensor system needs a mobile application that can be used to wirelessly transmit the bio signals from the proposed sensors to a mobile phone so that the user can analyze the signals in real time. This application would have to be user friendly and the signals would have to be displayed in their proper graph.

Vita

Alan Maciel Jimenez was born in Mexico City, Mexico, the son of Rita Morales and Antonio Estrada. After completing his school work at Jefferson County High School, he went to Jefferson Community and Technical College at Louisville, KY where he studied mathematics. In the summer of 2008, he enrolled at University of Louisville J.B. Speed School of Engineering where he studied electrical engineering and received his Bachelor of Science in Electrical Engineering in May 2012. Immediately after completing his Bachelor's degree he entered Graduate School at J.B. Speed School of Engineering. After his first year in graduate school he was employed as a full-time electrical engineer at General Electric Lighting at Bucyrus, Ohio.

References

[1] T. Jianhua and T. Tan, "Affective Computing: A Review," *Affective Computing and Intelligent Interaction*. LNCS 3784. Springer. pp. 981–995.

[2] H. Schlosberg, "Three dimensions of emotion," *Psychol. Rev.*, vol. 61, pp. 81–88, March. 1954.

[3] A. O'Donnell, "The autonomic nervous system," *The New Zealand Medical Student Journal,* vol. 26, pp. 11-13, Jul. 2011.

[4] A. P. Pentland, "Healthwear: Medical technology becomes wearable," *Studies in Health Technologies and Informatics*, vol. 118, pp. 55–65, May 2005.

[5] P. Bonato, "Wearable sensors/systems and their impact on biomedical engineering," *IEEE Engineering In Medicine and Biological Magazine,* vol. 22, pp. 18-20, May-Jun 2003.

[6] K. C. Welch, "Physiological Signals of Autistic Children Can be Useful," *IEEE Instrumentation and Measurement Magazine*, vol. 15, pp. 28-32, 2012.

[7] F. J. Barrera, R. A. Violo, and E. E. Graver, "On the form and function of severe self-injurious behavior," *Behavioral Interventions*, vol. 22, pp. 5–33, Feb. 2007.

[8] R. W. Picard," Emotion Research by the People, for the People," *Emotion Review*, vol. 2, pp. 250-254, Jul. 2010.

[9] Biopac Systems, Inc., Biopac, [online] 2013, http://www.biopac.com/ (Accessed: 10 February 2013).

[10] M. Z. Poh, N. C. Swenson, and R. W. Picard, "A Wearable Sensor for Unobtrusive, Long-Term Assessment of Electrodermal Activity," *IEEE Transactions on Biomedical Engineering*, vol. 57, 2010.

[11] W. James, "What is emotion?," *Mind,* vol. 9, pp. 188-205, 1884.

[12] Y. Lin, "A Natural Contact Sensor Paradigm for Nonintrusive and Real-Time Sensing of Biosignals in Human-Machine Interactions," *IEEE Sensors Journal*, vol. 11, 2011.

[13] A. Barreto, J. Zhai, N. Rishe, and Y. Gao, "Measurement of pupil diameter variations as a physiological indicator of the affective state in a computer user," *International ISA Biomedical Science Instrumentation Symposium,* vol. 43, pp. 146-151, 2007.

[14] W. Boucsein, *Electrodermal activity*, 2d. ed. New York: Springer, 2012.

[15] D. C. Fowles, M. J. Christie, R. Edelberg, W. W. Grings, D. T. Lykken, and P. H. Venables, "Committee report. Publication recommendations for electrodermal measurements," *Psychophysiology*, vol. 18, pp. 232–239, May 1981.

[16] M. S. Kalsi, "Design of Arterial Blood Pressure, Heart Rate Variability, and Breathing Rate Monitoring Device," Capstone, McMaster University, Hamilton, ON, Canada, 2009.

[17] W. J. Cui, L. E. Ostrander, and B. Y. Lee, "In vivo reflectance of blood and tissue as a function of light wavelength," *IEEE transactions on bio-medical engineering*, vol. 37, pp. 632-639, Jun. 1990.

[18] J. Malmivuo and R. Plonsey, "Bioelectromagnetism," New York: Oxford University Press, 1995.

[19] EDA Sensor., Cornell Edu, [online] 2013, https://courses.cit.cornell.edu/courses/ee476/FinalProjects/s2006/hmm32_pjw32/index.html (Accessed: 10 January 2013).

[20] Arduino Uno., Arduino, [online] 2013, http://www.arduino.cc/ (Accessed: 12 June 2013).

[21] T. J. Thompson, C. B. Kumar, and P. J. Kline, *Bluetooth Application Programming with the Java APIs*, London: Morgan Kaufmann, 2008.

[22] H. Labiod, H. Afifi, and C. De Santis, *WI-Fi Bluetooth Zigbee and Wimax*, New York: Springer, 2007.

[23] B. Kaelin, "3D Systems Founder Chuck Hull Honored with Award," (3D Printer World), [online] 2013, http://www.3dprinterworld.com/article/3d-systems-founder-chuck-hull-honored-with-award (Accessed: 7 July 2013).

[24] K.C. Welch, U. Lahiri, Z. Warren, and N. Sarkar, "An Approach to the Design of Socially Acceptable Robots for Children with Autism Spectrum Disorders," *International Journal of Social Robotics*, vol. 2 pp. 391-403, 2010.

[25] C. Liu, K. Conn, N. Sarkar, and W. Stone, "Online Affect Detection and Robot Behavior Adaptation for Intervention of Children with Autism," *IEEE Transactions on Robotics*, vol. 24 pp. 883-896, 2008.

[26] K.C. Welch, U. Lahiri, N. Sarkar, Z. Warren, W. Stone, and C. Liu, "Affect-sensitive computing and autism," *Affective Computing and Interaction: Psychological, Cognitive and Neuroscientific Perspectives*. IGI Global, Hershey, Pennsylvania, 2011.

[27] Sig Introduces Bluetooth Low Energy Wireless Technology, the Next Generation of Bluetooth Wireless Technology., Bluetooth Special Interest Group, [online] 2009, http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=4 (Accessed: 15 October 2013).

[28] N. Lee, "*Bluetooth 4.0: What is it, and does it matter?,*" (CNET), [online] 2011, http://reviews.cnet.com/8301-19512_7-20116316-233/bluetooth-4.0-what-is-it-and-does-it-matter/ (Accessed: 15 October 2013).

Appendices

The appendices section contains the code configurations software for the physiological sensor system and any custom code developed and used during the research. Any code not included was not used to collect data, not original to this thesis, or a derivation of included code too similar to require separate attachment.

Code Configurations

Arduino Code

```
#include <SD.h>        // Define SD card library
#include <Wire.h>
#include "RTClib.h"    // Define ral time clock library

#define LOG_INTERVAL  1000 // (1000 mills = 1 second)

#define SYNC_INTERVAL 1000 // mills between calls to flush() - to write data to
the card
uint32_t syncTime = 0; // time of last sync()

#define ECHO_TO_SERIAL   0 // echo data to serial port
#define WAIT_TO_START    0 // Wait for serial input in setup()

// The digital pins that connect to the LEDs
#define redLEDpin 2
#define greenLEDpin 3

//..................The analog pins that connect to the sensors........................

#define sensorPin 0        // analog 0

//#define tempPin 1             // analog 1
```

```
//#define BANDGAPREF 14          // special indicator that we want to measure
the bandgap

#define aref_voltage 3        // we tie 3.3V to ARef and measure it with a
multimeter!
#define bandgap_voltage 1.1     // this is not super guaranteed but its not -too-
off

//................ define the Real Time Clock object...................................

RTC_DS1307 RTC;

//............ for the data logging shield, we use digital pin 10 for the SD cs
line..........

const int chipSelect = 10; // This number is different for all data loggers, its 10
for adafruit


//........................... The log file name can be changed here............................
File logfile;

//....................This program sets flag if there is an error ............................

void error(char *str)
{
  Serial.print("error: ");
  Serial.println(str);

  // red LED indicates error
  digitalWrite(redLEDpin, HIGH);

  while(1);
}
//...............Serial port initialization ...............................................

void setup(void)
{
 Serial.begin(230400);
```

```
  Serial.println();

  // use debugging LEDs
  pinMode(redLEDpin, OUTPUT);
  pinMode(greenLEDpin, OUTPUT);


#if WAIT_TO_START
  Serial.println("Type any character to start");
  while (!Serial.available());
#endif //WAIT_TO_START

  // initialize the SD card
  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);


  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    error("Card failed, or not present");
  }
  Serial.println("card initialized.");

  // ........................This program makes logger file and increments it by 1
everytime.....................
  char filename[] = "LOGGER00.CSV";
  for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
      // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop!
    }
  }

  if (! logfile) {
```

```arduino
    error("couldnt create file");
  }

 Serial.print("Logging to: ");
 Serial.println(filename);

  // connect to RTC
  Wire.begin();
  if (!RTC.begin()) {
    logfile.println("RTC failed");

#if ECHO_TO_SERIAL
    Serial.println("RTC failed");
#endif  //ECHO_TO_SERIAL
  }

  logfile.println("millis,stamp,datetime,reading,voltage");
#if ECHO_TO_SERIAL
  Serial.println("millis,stamp,datetime,reading,voltage");
#endif //ECHO_TO_SERIAL

  // If you want to set the aref to something other than 5v
  analogReference(EXTERNAL);
}

void loop(void)
{
  DateTime now;
//------------------------------------------------------------------------------------------------
------
  // delay for the amount of time we want between readings
  //delay((LOG_INTERVAL -1) - (millis() % LOG_INTERVAL));
  delay(0.05);

 //----------------------------------------------------------------------------------------------
----
  digitalWrite(greenLEDpin, HIGH);

  // log milliseconds since starting
```

```cpp
  uint32_t m = millis();
  logfile.print(m);              // milliseconds since start
  logfile.print(", ");
#if ECHO_TO_SERIAL
  Serial.print(m);          // milliseconds since start
  Serial.print(", ");
#endif

  // fetch the time
  now = RTC.now();
  // log time
  logfile.print(now.unixtime()); // seconds since 1/1/1970
  logfile.print(", ");
  logfile.print('"');
  logfile.print(now.year(), DEC);
  logfile.print("/");
  logfile.print(now.month(), DEC);
  logfile.print("/");
  logfile.print(now.day(), DEC);
  logfile.print(" ");
  logfile.print(now.hour(), DEC);
  logfile.print(":");
  logfile.print(now.minute(), DEC);
  logfile.print(":");
  logfile.print(now.second(), DEC);
  logfile.print('"');
#if ECHO_TO_SERIAL
  Serial.print(now.unixtime()); // seconds since 1/1/1970
  Serial.print(", ");
  Serial.print('"');
  Serial.print(now.year(), DEC);
  Serial.print("/");
  Serial.print(now.month(), DEC);
  Serial.print("/");
  Serial.print(now.day(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(":");
  Serial.print(now.minute(), DEC);
```

```
  Serial.print(":");
  Serial.print(now.second(), DEC);
  Serial.print("");
#endif //ECHO_TO_SERIAL

  analogRead(sensorPin);
  delay(0.05);
  //int reading = analogRead(sensorPin);
  float reading = analogRead(sensorPin);

 float voltage = reading * 3;
 voltage /=1024.0;

  //..........................................................................

  logfile.print(", ");
  logfile.print(reading);
  logfile.print(", ");
  logfile.print(voltage);
#if ECHO_TO_SERIAL
  Serial.print(", ");
  Serial.print(reading);
  Serial.print(", ");
  Serial.print(voltage);

#endif //ECHO_TO_SERIAL

  logfile.println();
#if ECHO_TO_SERIAL
  Serial.println();
#endif // ECHO_TO_SERIAL

  digitalWrite(greenLEDpin, LOW);

  // Now we write data to disk! Don't sync too often - requires 2048 bytes of I/O
to SD card
  // which uses a bunch of power and takes time
  if ((millis() - syncTime) < SYNC_INTERVAL) return;
  syncTime = millis();
```

```
  // blink LED to show we are syncing data to the card & updating FAT!
  digitalWrite(redLEDpin, HIGH);
  logfile.flush();
  digitalWrite(redLEDpin, LOW);}
```

```
//---- The code below is used to extract GSR data and convert resistance to microSiemens------
```

```
int sensorPin=0;

float ADC_out;

float resi;

void setup()

{

  Serial.begin(9600);

}

void loop()

{

 ADC_out=analogRead(sensorPin);

//----------------------0------------------------------------

  if (ADC_out < 32)

{

  Serial.println("That's too much prespiration !");

  resi = 0;

}

//------------------------1------------------------------------

  else if (ADC_out >=32 && ADC_out < 300)
```

```
{

 resi = ((355.29*ADC_out)+26060);

}

//------------------------2------------------------------------

else if (ADC_out >=300 && ADC_out < 405)

{

  resi = ((799.48*ADC_out)-104360);

}

//------------------------3------------------------------------

else if (ADC_out >= 405 && ADC_out < 501)

{

  resi = ((1605*ADC_out)-437220);

}

//------------------------4------------------------------------

else if (ADC_out >=501 && ADC_out < 541)

{

  resi = (2869.8*ADC_out)-1066700;

}

//------------------------5------------------------------------

else if (ADC_out >=541 && ADC_out < 590)

{

   resi = (5047.1*ADC_out)-2251400;

}

//------------------------6------------------------------------

else if (ADC_out >=590 && ADC_out < 621)

{
```

```
  resi = (9466.9*ADC_out)-4858400;

}


//------------------------7------------------------------------
else if (ADC_out >=621 && ADC_out < 643)

{

  resi = (18367*ADC_out)-10386000;

}
//------------------------8------------------------------------
else if (ADC_out >=643 && ADC_out < 660)

{

  resi = (35445*ADC_out)-21389000;

}
//------------------------9------------------------------------
else if (ADC_out >=660 && ADC_out < 664)

{

  resi = (50000*ADC_out)-74506000;

}
//------------------------10------------------------------------
else if (ADC_out >=664 && ADC_out < 1023)

{

  resi = (79383*ADC_out)-50547000;

}
//--------------------------------------------------------------
else {Serial.println("Incorrect ADC output");}
```

```
//Serial.print(ADC_out);
//Serial.print("            ");
//Serial.println(resi);
//delay(1000);


float cond =((1/resi)*1000000);
Serial.print(resi);


Serial.print("           ");
Serial.print(cond);
Serial.print(" ");
Serial.println("uS");
delay(1000);


}
```

Processing Code

```
//import Serial communication library
import processing.serial.*;


//init variables
Serial commPort;
float tempC;
float tempF;
int yDist;
PFont font12;
PFont font24;
float[] tempHistory = new float[100];


void setup()
{
  //setup fonts for use throughout the application
  font12 = loadFont("Verdana-12.vlw");
  font24 = loadFont("Verdana-24.vlw");


  //set the size of the window
  size(210, 200);


  //init serial communication port
  commPort = new Serial(this, "COM5", 9600);
```

```
  //fill tempHistory with default temps
  for(int index = 0; index<100; index++)
    tempHistory[index] = 0;
}


void draw()
{
  //get the temp from the serial port
  while (commPort.available() > 0)
  {
    tempC = commPort.read();


    //refresh the background to clear old data
    background(123);


    //draw the temp rectangle
    colorMode(RGB, 160);  //use color mode sized for fading
    stroke (0);
    rect (49,19,22,162);
    //fade red and blue within the rectangle
    for (int colorIndex = 0; colorIndex <= 160; colorIndex++)
    {
      stroke(160 - colorIndex, 0, colorIndex);
      line(50, colorIndex + 20, 70, colorIndex + 20);
    }
```

```
//draw graph
stroke(0);
fill(255,255,255);
rect(90,80,100,100);
for (int index = 0; index<100; index++)
{
  if(index == 99)
    tempHistory[index] = tempC;
  else
    tempHistory[index] = tempHistory[index + 1];
  point(90 + index, 180 - tempHistory[index]);
}


//write reference values
fill(0,0,0);
textFont(font12);
textAlign(RIGHT);
text("212 F", 45, 25);
text("32 F", 45, 187);


//draw triangle pointer
yDist = int(160 - (160 * (tempC * 0.01)));
stroke(0);
triangle(75, yDist + 20, 85, yDist + 15, 85, yDist + 25);


//write the temp in C and F
```

```
    fill(0,0,0);

    textFont(font24);

    textAlign(LEFT);

    tempC = 5*tempC*100/1024;

    text(str(int(tempC)) + " C", 115, 37);

    text(str(int(tempC*1.8+32)) + " F", 115, 65);
  }
}
```

MATLAB Code

```matlab
function [ibi,peak_time,ppg_mean,ppg_max] =  peak_detection(ppg)

% The code below filters the incoming ppg signal using a 10 point moving
% average
%-----------------------------------------------------------------------
a=1;     % Initialized the variable
ppg=ppg*1000;         % multiplied the each data point by 1000
b=0.1*ones(1,10);     %[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ]
sp=0;
peak_found=0;
ppg_filt=filtfilt(b,a,ppg);
mean_level=mean(ppg_filt);

if mean_level< 0
   ppg_filt = ppg_filt-mean_level;
end
%-----------------------------------------------------------------------

peak_pos=[];
len=length(ppg_filt);
k=1;
count=1;
prev=ppg_filt(k);
k=k+1;
missed=0;
flag_ppg=0;
peak_amp=[];
push_back=0;

%-----------------------------------------------------------------------
%The code below checks ff there is a flat area in the close to peak. If the
%flat area is present then it will be tested as well
while k < len-1

  curr=ppg_filt(k);
  next=ppg_filt(k+1);
if (curr == prev) | (curr == next)
   if k <= 10
      m = k-1;

   else m = 10;
```

```matlab
        end
        if (k+10)>len
            n = len - k;
          else
            n = 10;
        end

    if (curr > ppg_filt(k-m))& (curr > ppg_filt(k+n))
       if (sp == 1)|(peak_found == 1)
          sp = 0;
          peak_found = 0;
       else
          sp = 1;
       end
    end
    end
%------------------------------------------------------------------------
%The code below checks two adjacent points to see if there is a peak

    if ((curr > prev) & (curr > next)) | (sp == 1)
        if k <= 250
            m = k - 1;
        else

            m = 250;
        end
        if (k + 200) > len
            n = len - k;
        else
            n = 200;
        end
% The code below checks if the amplitide difference between the peak is large
enough
% This step takes care of small peaks present in the signal.
    if ((ppg_filt(k) - ppg_filt(k-m)) > 0.5) & ((ppg_filt(k) - ppg_filt(k+n)) > 0.5)

       if isempty(peak_pos)
          peak=curr;
          peak_pos = k ;
          first_peak =peak_pos;
          peak_time(1) = peak_pos;
          peak_found = 1;

       else
```

119

```matlab
    if (k - peak_pos) > 50    % Here we look for the distance between peaks
             peak = curr;  % if it is large enough we treat it as a valid peak
             new_peak_pos = k;
             flag_ppg = 1;
             k = k +25;

  % If two peaks are close to each other and the current one is larger, the program push_back
  % and replace the previous one with the current larger peak.

    elseif  (abs(ppg_filt(k)-ppg_filt(peak_pos))  >  .25)  &  (abs(ppg_filt(k)-ppg_filt(peak_pos)) < 20)
        if((k - peak_pos)< 70)& (curr> peak)
           if count > 1

               peak_pos = peak_time(count-1);
               flag_ppg = 1;
               new_peak_pos = k;
               peak = curr;
               push_back = 1;

           else
               peak = curr;
               peak_pos = k;
               first_peak =peak_pos;
               peak_time(1) = peak_pos;
               peak_found = 1;
           end
        end
   end
%--------------------------------------------------------------------------
% Once the peak is found, the program testify if there are any missing peaks between
% previous and current peak. If the distance between the peaks is too large
% then it indicates that we have missed one or multiple peaks in between.

if flag_ppg == 1

   if ((new_peak_pos - peak_pos) > 70) & (count ~= 1)
               missed_peak_pos = round(mean(peak_pos:new_peak_pos));
               count = count + 1;
               peak_time(count) = missed_peak_pos;
               ibi(count-1) = missed_peak_pos - peak_pos;
               if ~isempty(peak_amp)
```

```matlab
                    peak_amp(count-1) = peak_amp(count-2);
                end
                peak = ppg_filt(missed_peak_pos);
                peak_pos = missed_peak_pos;
            end
            if push_back == 0
                count = count + 1;
            elseif push_back == 1
                push_back = 0;
            end
% Here the peak time and ibi is recorded and peak_found flag is set
            peak_time(count) = new_peak_pos;
            ibi(count-1) = new_peak_pos - peak_pos;
            peak_found = 1;
            k = k  + 85;
            if (new_peak_pos - peak_pos) < 1000
                infl_val = min(ppg_filt(peak_pos:new_peak_pos));
                llen = find((ppg_filt(peak_pos:new_peak_pos)) == infl_val);
                if length(llen) > 1
                    if mod(length(llen),2) == 0
                        llen = llen(length(llen)/2+1);
                    else
                        llen = llen((length(llen)+1)/2);
                    end
                end
                infl = peak_pos + llen;
            else
                infl_val = min(ppg_filt((peak_pos):new_peak_pos));
                llen = find((ppg_filt((peak_pos):new_peak_pos)) == infl_val);
                if length(llen) > 1
                    if mod(length(llen),2) == 0
                        llen = llen(length(llen)/2+1);
                    else
                        llen = llen((length(llen)+1)/2);
                    end
                end
                infl = (new_peak_pos-900) + llen;
            end
            peak_amp(count-1) = ppg_filt(new_peak_pos) - ppg_filt(infl);
            peak_pos = new_peak_pos;
            flag_ppg = 0;
        end
    end
elseif ((ppg_filt(k) - ppg_filt(k-m)) > 0.5) & ((ppg_filt(k) - ppg_filt(k+n)) < 0.5)
```

```
            if (k + n + 200) > len
                j = len - k - n;
            else
                j = n + 200;
            end
            if ((ppg_filt(k) - ppg_filt(k+j)) > j*8/400) | (k - peak_pos > 600)
                if isempty(peak_pos)
                    peak = curr;
                    peak_pos = k;
                    first_peak = peak_pos;
                    peak_time(1) = peak_pos;
                    peak_found = 1;
                else
                    if (k - peak_pos)  > 70
                        peak = curr;
                        new_peak_pos = k;
                        flag_ppg = 1;
                    elseif (abs(ppg_filt(k)-ppg_filt(peak_pos)) > .5) & (abs(ppg_filt(k)-
ppg_filt(peak_pos)) < 20)
                        if ((k - peak_pos) < 50) & (curr > peak )
                            if count > 1
                                peak_pos = peak_time(count-1);
                                flag_ppg = 1;
                                new_peak_pos = k;
                                peak = curr;
                                push_back =1;
                            else
                                peak = curr;
                                peak_pos = k;
                                first_peak = peak_pos;
                                peak_time(1) = peak_pos;
                                peak_found = 1;
                            end
                        end
                    end
                    if flag_ppg == 1
                        if (new_peak_pos - peak_pos) > 80
                            missed_peak_pos = round(mean(peak_pos:new_peak_pos));
                            count = count + 1;
                            peak_time(count) = missed_peak_pos;
                            ibi(count-1) = missed_peak_pos - peak_pos;
                            if ~isempty(peak_amp)
                                peak_amp(count-1) = peak_amp(count-2);
                            end
```

```matlab
                peak = ppg_filt(missed_peak_pos);
                peak_pos = missed_peak_pos;
            end
            if push_back == 0
                count = count + 1;
            elseif push_back == 1
                push_back = 0;
            end
            peak_time(count) = new_peak_pos;
            k = k + 70;
            ibi(count-1) = new_peak_pos - peak_pos;
            peak_found = 1;
            if (new_peak_pos - peak_pos) < 100
                infl_val = min(ppg_filt(peak_pos:new_peak_pos));
                llen = find((ppg_filt(peak_pos:new_peak_pos)) == infl_val);
                if length(llen) > 1
                    if mod(length(llen),2) == 0
                        llen = llen(length(llen)/2+1);
                    else
                        llen = llen((length(llen)+1)/2);
                    end
                end
                infl = peak_pos + llen;
            else
                infl_val             =             min(ppg_filt((new_peak_pos-
900):new_peak_pos));
                llen   =   find((ppg_filt((new_peak_pos-900):new_peak_pos))
== infl_val);
                if length(llen) > 1
                    if mod(length(llen),2) == 0
                        llen = llen(length(llen)/2+1);
                    else
                        llen = llen((length(llen)+1)/2);
                    end
                end
                infl = (new_peak_pos-900) + llen;
            end
            peak_amp(count-1) = ppg_filt(new_peak_pos) - ppg_filt(infl);
            peak_pos = new_peak_pos;
            flag_ppg = 0;
        end
    end
end
```

```
elseif ((ppg_filt(k) - ppg_filt(k-m)) < 1) & ((ppg_filt(k) - ppg_filt(k+n)) > 1)
    if k < m + 25
        j = k-1;
    else
        j = m + 25;
    end
    if ((ppg_filt(k) - ppg_filt(k-j)) > (j*8/450)) | (k - peak_pos > 80)
        if isempty(peak_pos)
            peak = curr;
            peak_pos = k;
            first_peak = peak_pos;
            peak_time(1) = peak_pos;
            peak_found = 1;
        else
            if (k - peak_pos)  > 50
                peak = curr;
                new_peak_pos = k;
                flag_ppg = 1;
            elseif (abs(ppg_filt(k)-ppg_filt(peak_pos)) > .5) & (abs(ppg_filt(k)-
ppg_filt(peak_pos)) < 20)
                if ((k - peak_pos) < 70) & (curr > peak )
                    if count > 1
                        peak_pos = peak_time(count-1);
                        flag_ppg = 1;
                        new_peak_pos = k;
                        peak = curr;
                        push_back =1;
                    else
                        peak = curr;
                        peak_pos = k;
                        first_peak = peak_pos;
                        peak_time(1) = peak_pos;
                        peak_found = 1;
                    end
                end
            end
            if flag_ppg == 1
                if (new_peak_pos - peak_pos) > 70
                    missed_peak_pos = round(mean(peak_pos:new_peak_pos));
                    count = count + 1;
                    peak_time(count) = missed_peak_pos;
                    ibi(count-1) = missed_peak_pos - peak_pos;
                    if ~isempty(peak_amp)
                        peak_amp(count-1) = peak_amp(count-2);
```

124

```
                    end
                    peak = ppg_filt(missed_peak_pos);
                    peak_pos = missed_peak_pos;
                end
                if push_back == 0
                    count = count + 1;
                elseif push_back == 1
                    push_back = 0;
                end
                peak_time(count) = new_peak_pos;
                k = k+ 25;
                ibi(count-1) = new_peak_pos - peak_pos;
                peak_found = 1;
                if (new_peak_pos - peak_pos) < 70
                    infl_val = min(ppg_filt(peak_pos:new_peak_pos));
                    llen = find((ppg_filt(peak_pos:new_peak_pos)) == infl_val);
                    if length(llen) > 1
                        if mod(length(llen),2) == 0
                            llen = llen(length(llen)/2+1);
                        else
                            llen = llen((length(llen)+1)/2);
                        end
                    end
                    infl = peak_pos + llen;
                else
                    infl_val               =               min(ppg_filt((new_peak_pos-
600):new_peak_pos));
                        llen   =   find((ppg_filt((new_peak_pos-600):new_peak_pos))
== infl_val);
                    if length(llen) > 1
                        if mod(length(llen),2) == 0
                            llen = llen(length(llen)/2+1);
                        else
                            llen = llen((length(llen)+1)/2);
                        end
                    end
                    infl = (new_peak_pos-900) + llen;
                end
                peak_amp(count-1) = ppg_filt(new_peak_pos) - ppg_filt(infl);
                peak_pos = new_peak_pos;
                flag_ppg = 0;
            end
        end
    end
```

```matlab
        end
    end
    if sp == 1
        sp = 0;
    end

    prev = curr;
    k = k+1;
end
peak_amp(count) = peak-ppg_filt(infl);
ppg_mean = mean(peak_amp)/1000;
ppg_max = max(peak_amp)/1000;
```

```matlab
% the code below is used for SKT comparisons to the Biopac system


data=xlsread('Anand_temp_1.xlsx');

biopac_data=data(:,1);
inHouse_data=data(:,2);

 figure

 plot(biopac_data)
  set(gca,'XLim',[0 12000])
   set(gca,'YLim',[82 90])
   set(gca,'XTick',[0:2400:12000])
   set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
   set(gca,'FontSize',20,'fontWeight','bold')
   title('Biopac temperature sensor')
   xlabel('Time (min)')
   s=sprintf('Temperature (%cF)', char(176));
   ylabel(s)
%  figure
% %
%  plot(inHouse_data)
```

```matlab
    a=1;
% % %lm35_data=lm35_data*1000;
    b=0.1*ones(1,10);
%  %
     temp_filt=filtfilt(b,a,inHouse_data);
    figure
    plot(temp_filt)
 %--------------------------------------------------------%
    set(gca,'XLim',[0 12000])
    set(gca,'YLim',[82 90])
    set(gca,'XTick',[0:2400:12000])
    set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
    set(gca,'FontSize',20,'fontWeight','bold')
    title('In-house temperature sensor')
    xlabel('Time (min)')
    s=sprintf('Temperature (%cF)', char(176));
    ylabel(s)
 %--------------------------------------------------------%
  R1=corrcoef(biopac_data,temp_filt);


% %----------------------------------------------------------------
%
%   y1=medfilt1(inHouse_data,2);
% % %y2=medfilt1(temp_filt,30);
%  %
%    figure
% % %
%   plot(y1)
% %
%


% output=tsmovavg(inHouse_data','s',100);
%
% figure
%
% plot(output)
%
% R2=corrcoef(biopac_data,output);
```

```matlab
% The code below is used for the GSR comparisons to the Biopac system

data=xlsread('Alan_GSR_2.xlsx');

biopac_data=data(:,1);
inHouse_data=data(:,2);

  figure

  plot(biopac_data)

    set(gca,'XLim',[0 12000])
    set(gca,'YLim',[0 8])
  set(gca,'XTick',[0:2400:12000])
  set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
  set(gca,'FontSize',20,'fontWeight','bold')
  title('Biopac GSR sensor')
  xlabel('Time (min)')
  ylabel('Conductivity (microsiemens)')

%
%  figure
% %
%  plot(inHouse_data)


%
   a=1;
% % % %lm35_data=lm35_data*1000;
   b=0.1*ones(1,10);
% %  %
   gsr_filt=filtfilt(b,a,inHouse_data);
   figure
   plot(gsr_filt)
   set(gca,'XLim',[0 12000])
   set(gca,'YLim',[0 7])
```

```matlab
set(gca,'YTick',[0:2:6])
set(gca,'XTick',[0:2400:12000])
set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
set(gca,'FontSize',20,'fontWeight','bold')
 title('In-house GSR sensor ')
xlabel('Time (min)')
ylabel('Conductivity (microsiemens)')




R=corr(biopac_data,gsr_filt,'type', 'pearson');

get(gca)
% %-------------------------------------------------------------
%
%   y1=medfilt1(inHouse_data,10);
% % % % %y2=medfilt1(temp_filt,30);
% % % %
%   figure
% % %
%   plot(y1)
% %
%


data=xlsread('Anand_temp_1.xlsx');

biopac_data=data(:,1);
inHouse_data=data(:,2);

 figure

 plot(biopac_data)
  set(gca,'XLim',[0 12000])
   set(gca,'YLim',[82 90])
   set(gca,'XTick',[0:2400:12000])
   set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
   set(gca,'FontSize',20,'fontWeight','bold')
   title('Biopac temperature sensor')
```

```matlab
    xlabel('Time (min)')
    s=sprintf('Temperature (%cF)', char(176));
    ylabel(s)
% figure
% %
% plot(inHouse_data)




    a=1;
% % %lm35_data=lm35_data*1000;
    b=0.1*ones(1,10);
% %
     temp_filt=filtfilt(b,a,inHouse_data);
    figure
    plot(temp_filt)
 %---------------------------------------------------------%
    set(gca,'XLim',[0 12000])
    set(gca,'YLim',[82 90])
    set(gca,'XTick',[0:2400:12000])
    set(gca,'XTickLabel',['0';'1';'2';'3';'4';'5'])
    set(gca,'FontSize',20,'fontWeight','bold')
    title('In-house temperature sensor')
    xlabel('Time (min)')
    s=sprintf('Temperature (%cF)', char(176));
    ylabel(s)
 %---------------------------------------------------------%
  R1=corrcoef(biopac_data,temp_filt);

% %-----------------------------------------------------------------
%
%   y1=medfilt1(inHouse_data,2);
% % %y2=medfilt1(temp_filt,30);
% %
%    figure
% % %
%   plot(y1)
% %
%

% output=tsmovavg(inHouse_data','s',100);
%
% figure
```

```
%
% plot(output)
%
% R2=corrcoef(biopac_data,output);
```