

**Physiological Sensor System for a  
Wireless Tactile Display**

by

Nikhila Deo

Submitted to the Department of Electrical Engineering and Computer Science

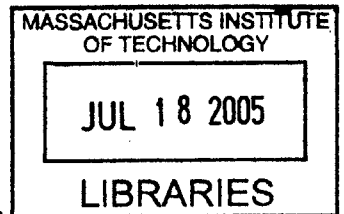
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 19, 2005 [June 2005]

© 2005 Massachusetts Institute of Technology. All rights reserved.



Author \_\_\_\_\_  
Department of Electrical Engineering and Computer Science  
May 19, 2005

*11*

Certified by \_\_\_\_\_  
Lynette A. Jones  
Thesis Supervisor

Accepted by \_\_\_\_\_  
Arthur C. Smith  
Chairman, Department Committee on Graduate Theses

**BARKER**

Physiological Sensor System for a Wireless Tactile Display  
by  
Nikhila Deo

Submitted to the  
Department of Electrical Engineering and Computer Science

May 19, 2005

In Partial Fulfillment of the Requirements for the Degree of Master of Engineering in  
Electrical Engineering and Computer Science

### **ABSTRACT**

This research focuses on the development of a wearable Physiological Sensor System (PSS) that can be used with a wireless tactile control unit to monitor the physiological status of a mobile user. Two physiological variables, heart rate and respiration rate, were selected for initial study, and technologies that could be used to measure these variables were evaluated. The PSS was required to be non-invasive, lightweight, and low-power. Initial studies indicated that heart rate could be most effectively measured using infrared light technology and so this was implemented in a wrist strap. It was determined that respiration rate could be recorded using a novel implementation of conducting polymer strain gages. For this purpose, polypyrrole-coated Lycra was fabricated and a chest strap was made that incorporated this sensor for recording respiration rate. Both sensors were connected to the PSS unit that was designed to provide the necessary signal processing, such as filtering and amplification, as well as produce the digital output signals that could be analyzed by the microcontroller in the tactile control unit. The complete circuit design process is described, from initial prototype development and testing to final layout and printed circuit board fabrication.

Thesis Supervisor: Lynette A. Jones

Title: Principal Research Scientist

# Acknowledgements

This research was supported by a grant from the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

I would like to thank all of the students, researchers, and professors in the MIT BioInstrumentation Laboratory for their never-ending willingness to help. The option for communication among the many specialty groups in the lab made it possible to explore a wide range of technologies for this research project.

I owe a tremendous thank you to Dr. Lynette Jones for giving me the opportunity to pursue this project and gain experience in tying together the electrical engineering and biomedical engineering fields. I sincerely appreciate all her guidance and support throughout the entire process.

I also owe an incredible thank you to Brett Lockyer for his expertise and patience. Without his help and friendship, this project would not have been as successful nor nearly as enjoyable.

Finally, I owe a thank you to two very special people in my life. Samir, thank you for understanding the intensity of this thesis project, helping out whenever you could, bringing me snacks, and keeping me balanced outside of lab. On tough days at work, getting to spend time with you and the end was all the motivation I needed to get through it.

And Mother, thank you so very much for your continual support in all of my educational endeavors, for knowing when to push me, and for knowing when to lend me a shoulder to lean on. Your work ethic will forever be an inspiration to me, and I appreciate all the opportunities you have given me more than words can say. Thank you, Mom.

# Table of Contents

ABSTRACT.....	2
LIST OF FIGURES .....	5
LIST OF TABLES.....	5
INTRODUCTION .....	6
Wireless Bio-Monitoring .....	6
Heart-Rate Monitoring .....	7
Respiration Rate Monitoring.....	9
High-End Vital Sign Monitoring Systems .....	10
Physiological Sensor System.....	12
SYSTEM DESIGN .....	13
Overview .....	13
Explored technologies.....	13
Wireless Tactile Control Unit .....	16
Microcontroller .....	17
Bluetooth .....	18
PROTOTYPE DESIGN AND CONSTRUCTION .....	20
Heart Rate Sensor .....	21
First Prototype .....	21
Second Prototype .....	27
System Results.....	34
Respiration Rate Sensor .....	35
Creating Polypyrrole Lycra Strain Gages .....	35
Properties of Polypyrrole Lycra Strain Gages .....	37
Primary Prototype .....	40
Software.....	44
LAYOUT AND FABRICATION.....	48
Compatible WTCU Layout .....	48
Printed Circuit Board Components .....	49
PSS Schematic.....	52
Printed Circuit Board Layout .....	55
FINAL PSS PRODUCT DESIGN.....	59
PCB Population and Testing .....	59
Wearable Sensors .....	61
FURTHER RESEARCH .....	65
REFERENCES .....	68
APPENDIX A.....	72
APPENDIX B .....	73
APPENDIX C .....	74



## List of Figures

Figure 1: Breadboard used for prototype design.....	20
Figure 2: Heartbeat transducer circuit.....	22
Figure 3: Unprocessed IR sensor output (finger).....	22
Figure 4: Unprocessed IR sensor output (wrist) .....	24
Figure 5: Unprocessed IR sensor output (collarbone) .....	24
Figure 6: Output of Butterworth low-pass filter (frequency sweep).....	26
Figure 7: Noise-free output of Butterworth low-pass filter (finger).....	27
Figure 8: Baseline drift in output of Butterworth low-pass filter (wrist).....	30
Figure 9: Reduced baseline drift at output of band-pass filter and diode (wrist) .....	32
Figure 10: Comparison of diode output and comparator output (wrist) .....	33
Figure 11: Block diagram of PSS heart rate sensor process .....	34
Figure 12: Electrochemical deposition setup.....	37
Figure 13: PPy Lycra strain gage (Batch Three) resistance verses stretch.....	38
Figure 14: PPy Lycra strain gage (Batch Two) strip comparison.....	39
Figure 15: Quarter Wheatstone bridge circuit .....	41
Figure 16: Amplified respiration voltage pattern.....	43
Figure 17: Block diagram of PSS respiration rate sensor process.....	44
Figure 18: GUI showing heart rate query and results.....	46
Figure 19: Overall schematic of the PSS .....	54
Figure 20: PSS footprint layout .....	55
Figure 21: Final PCB Layout, all layers of PSS board .....	58
Figure 22: Top layer PCB layout .....	58
Figure 23: Bottom layer PCB layout .....	58
Figure 24: Top layer of unpopulated PCB (actual size) .....	59
Figure 25: Bottom layer of unpopulated PCB (actual size).....	59
Figure 26: Final PSS board and physiological sensors circuit .....	60
Figure 27: PSS wrist strap design .....	62
Figure 28: PSS chest strap design.....	63
Figure A-1: Electrochemical deposition properties (Batch Three).....	72

## List of Tables

Table 1: Comparison of PSS heart rate sensor and Polar M32.....	35
Table 2: Gage Factors of PPy Lycra strain gages .....	40
Table 3: Supply placement options.....	48
Table 4: Total current and power used by PSS.....	61
Table B-1: List of PCB parts .....	73

# Introduction

## Wireless Bio-Monitoring

The first types of wireless physiological monitors were developed as early as the 1950s, when wireless communications technologies were used to measure the temperature and pH of internal cavities [1]. Two decades later, measurements taken from human subjects showed the feasibility of fetal heart rate monitoring in utero [2, 3, 4]. This wireless bio-monitoring technology developed for monitoring the fetal heart rate over 30 years ago is now becoming a remote sensing technology used for many vital sign measurements.

Bio-monitoring involves using remote sensing to measure one or many of the following vital signs: heart rate, respiration rate, activity (acceleration), temperature (skin or ambient), pulse oximetry, and diastolic and systolic blood pressure. Only recently have portable, affordable bio-monitoring devices become generally available for use by patients and caregivers [5]; however, the requirement of a very low power source continues to be an issue in the design of these devices. Even more recently, wireless technological developments have enabled short-range wireless monitoring, which was previously restricted by radio-frequency interference issues as well as health concerns over proximity to the body. New methods for wireless communication, such as ultra-wideband, are not affected by radio spectrum restrictions, and therefore have proven useful for monitoring vital signs in a variety of settings, such as in ambulances, hospitals, or the home [5].

## Heart Rate Monitoring

Heart rate monitoring can be performed using a number of electrode and non-electrode based methods. Studies in the 1970s involving wireless heart rate monitoring used surface electrodes on the skin to detect changes in the electrical potentials generated by the heart [6]. More recently, heart rate monitoring methods that have been used in conjunction with wireless communications include wireless heart rate chest straps containing electrodes to measure the electrocardiogram (ECG) signal, pulse oximeters, ultrasound Doppler, pulse pressure detection by strain gage impedance, and electromagnetic flow [5].

ECG monitoring is the most widely used method for studying the activity of the heart. ECG electrodes are attached to a person's body while he or she lies still. The electrodes are attached to between two and twelve pre-defined positions on the front of the chest and the extremities [7]. Gel is applied to the skin where each electrode will be placed to facilitate the transmission of electrical impulses from the heart to the ECG leads. Information that can be obtained from the ECG waveform includes the underlying rate and rhythmic mechanism of the heart, the orientation of the heart in the chest cavity, abnormal cardiac rhythm disturbances, muscle damage, impaired blood flow, and hypertrophy [7]. The main problems of real-time ECG monitoring include baseline drift, physiological variability of the main pulse peaks (or QRS complexes), power line interference, muscle noise, and artifacts due to electrode motion [8].

Wireless chest strap heart rate monitoring incorporates the ECG method to monitor heart rate and is frequently used by athletes and those in exercise programs. The electrodes in the chest strap collect the potential difference across the chest created by the

heart beating, and transmit the signal to a watch or other portable monitoring device. Electrode movement can cause a significant amount high-frequency noise and a number of motion artifacts to appear in the ECG signal, which is why accurate ECG monitoring is ideally performed on a stationary person. Commercial real-time heart rate monitors that can be used during exercise use low-pass filtering methods as well as software algorithms to try to provide an estimate of the heart rate in spite of these motion artifacts. The average cost for one of these individual heart-rate monitoring systems varies between \$100 and \$600 (Polar USA, Lake Success, NY). Polar USA is a major manufacturer and supplier of these devices.

Another method used to monitor the heart rate is based on infrared (IR) light that measures the changes in arterial blood volume associated with each heart beat. IR light is shined on a localized area of skin, usually on the finger. An IR light detector, placed either directly on the other side of the finger or a small distance down the finger in the direction of blood-flow, measures the amount of IR light that is absorbed. This amount varies due to the blood volume, which rises and falls with the pumping of the heart. Alternatively, if the finger can be isolated in a completely dark environment, then the IR light reflected by the skin can be measured, for the skin's reflectivity also depends on the changes in blood density due to the pumping of the heart. The magnitude of the signal depends on the amount of blood pulsing into the vascular bed, the optical absorption of the blood, skin and tissue, and the wavelength that is used to illuminate the blood [5]. The pulsating (AC) signal results only from the arterial blood component, whereas the non-pulsating (DC) signal is related to absorption by the bloodless tissue, venous blood and residual arterial blood during diastole, as well as skin pigmentation [5]. Information from

the AC signal component can be used to measure the heart rate, and information from both AC and DC components can potentially be used to determine the blood oxygen saturation level [5].

To measure the pulse oximetry, the AC signal at each wavelength is normalized by dividing it by its corresponding DC component. This process yields a normalized red/infrared ratio that is only dependent on the color of the arterial blood; therefore it does not need to be recalibrated for use on different people. However, to measure the heart rate, the peaks in the AC signal need to be counted in relation to time, and the DC signal can be ignored [5]. The smallest portable commercial system for measuring pulse oximetry is the wrist-worn device engineered by Minolta. It uses a fingertip sensor, has an accuracy of approximately 2%, requires two AAA batteries with a battery life of 48 hours, and has a mass of 42 grams. The cost of such a device is about \$500 (Minolta, Columbia, SC). Many other pulse oximetry or IR light heart rate monitoring devices exist, but most are costly, bulky, and only take measurements from the fingertip or earlobe.

## **Respiration Rate Monitoring**

Respiration monitoring methods fall into two categories: direct and indirect. Spirometers and nasal thermocouples are devices that measure air flow into and out of the lungs directly, whereas pneumatic respiration transducers, gas pressure belt sensors, transthoracic inductance and impedance plethysmographs, and strain gage measurement of thoracic circumference are indirect ways to measure the respiration rate [9]. Direct measurements are typically the most accurate, but they are intrusive and interfere with

normal respiration. A whole-body plethysmograph can give a highly accurate measurement and does not interfere with respiration, but it requires immobilizing the person [9]. The gas-pressure respiration belt monitors the changes in pressure of the belt bladder as a person breathes. This system is bulky and requires additional equipment involving a gas pump and a computer program for analysis [10]. This technology is also relatively expensive. A commercial transducer such as the TDR-50C costs \$250.00 and its signal-processing unit CANL-405 costs \$615.00 (Med Associates, St. Albans, VT).

The only method suitable for respiration rate monitoring that does not burden the individual is a transducer belt that is based on strain gages. These sensors respond to changes in length, due to an applied force, with changes in resistance [11]. The bridge circuit and applied voltage used in conjunction with strain gages enables the measurement of small changes in resistance by measuring voltage changes. As the resistance value pulses, the voltage will also pulse, thereby allowing the respiration rate to be measured.

## **High-End Vital Sign Monitoring Systems**

Two state-of-the-art wearable vital sign monitoring systems demonstrate the capabilities and purpose of developing these systems: LifeGuard and Vital Dust. LifeGuard has been developed by the NASA Ames Astrobionics team to log physiological data from astronauts and transmit it to a portable base station PC for display and further processing [12]. Although it was developed initially for astronauts, it has many other potential uses in clinical, home-health, and military applications. The main component of the small, lightweight, wearable system called the CPOD is worn on

the body and attached to several sensors [12]. The CPOD device contains a low-power microcontroller, an internal A/D converter, flash memory, and uses 916 MHz or Bluetooth technology [12]. ECG, respiration rate, three-axis activity acceleration, ambient temperature, heart rate, pulse oximetry, and blood pressure are all acquired and stored in the device for up to eight hours and can be downloaded or streamed to the base computer on demand. The sensors are mostly external, and the ECG and respiration signals are acquired with standard snap-on button electrodes [12].

Vital Dust is composed of a small, low-power computer connected to a fingertip sensor [13]. It runs on two AA batteries, includes memory, an embedded microprocessor, and a wireless communication interface. Vital Dust uses a pulse oximeter and 2-lead ECG to collect data, including heart rate and oxygen saturation, that are then sent over a short-range wireless network to a computer or PDA. A unique feature of Vital Dust is that it can store the pre-hospital electronic medical record as well as record a patient's vital signs [13].

There are three basic scenarios in which LifeGuard, Vital Dust, or any other wireless vital sign monitoring system can be of great value: monitoring the vital signs of an injured person during transport to a hospital, monitoring the health of a person at home or in the hospital, and monitoring the health of someone who is performing hazardous work or in combat. In the first scenario, an injured person's vital signs could be transmitted to a hospital while the ambulance is still on its way so that upon arrival, the doctor already has knowledge of the patient's health status [12]. In the second scenario, a patient's time in the hospital could be minimized if home health monitoring is available. In the third scenario, for a soldier in combat or a person in another occupation where

great physical strain is involved, wearing such a system would provide a mechanism for measuring the response of the individual to the environment in which they are working.

## **Physiological Sensor System**

This thesis details the design of a low-power, non-invasive, light-weight, wireless Physiological Sensor System (PSS) to measure heart rate and respiration rate. The main factors to be considered in the design of this system are power and energy efficiency, non-invasive sensing methods, and wearable sensor harnesses that are comfortable, unobtrusive, and well-positioned so as to not hinder any required body functionality. The PSS will be used in conjunction with a Wireless Tactile Control Unit (WTCU) [14], also developed in the MIT BioInstrumentation laboratory, which gives navigation commands to ambulatory human operators using vibrating motors [15].

The sensors worn by the human operator will record the heart pulse and respiration pattern. Processing of these signals within the PSS unit will turn the AC components of these signals into digital signals that will be sent to the microcontroller on the WTCU. An algorithm will determine the heart rate or respiration rate from the received digital signal, and the result will be transmitted wirelessly to a supervisory agent, who can monitor the status of the individual. The placement of the sensors and their construction will be designed for use in a military environment.



# System Design

## Overview

To meet the requirements of monitoring physiological signals in healthy individuals and interfacing this system with the WTCU, the wearable sensors and signal conditioning units were designed and built in-house. By designing the entire system in-house, it was possible to meet power guidelines, fit the physical requirements of the system, use the open ports on the WTCU's AT90LS8535 microcontroller, and remain relatively low-cost when compared to commercially available bio-monitoring systems. The technology pursued for measuring the heart rate was IR Light Monitoring, and a novel method was explored for measuring the respiration rate that was based on a polypyrrole Lycra strain gage. The heart rate sensor was implemented in a wrist strap so as to not interfere with the functionality of the hands. The respiration rate sensor was implemented in a chest strap.

## Explored technologies

ECG monitoring was the first method pursued for measuring the heart rate since it is the standard method used in commercial systems. The 9050 HeartTrace SolidGel electrodes were experimented with due to their use of solid gels instead of messy liquid gels (Cardiology Shop, Berlin, MA). The set-up time for the ECG procedure with one subject was approximately five minutes. An alternative to standard electrodes is the fabric wire mesh electrode, such as those employed in the Polar M32 (Polar USA, Lake Success, NY). The wire mesh is comfortable and only requires wetting with water rather

than a gel. However, this wire mesh technology is far less accessible and more expensive than any of the standard electrodes.

Additional problems regarding real-time ECG monitoring include dealing with aberrations in the ECG waveform. Power line noise, abnormal QRS shapes, high frequency artifacts due to electrode motion that may look like a QRS peak, and low frequency baseline drift due to respiration patterns are only some of the factors that can affect the ECG signal [8]. Since power line noise is approximately 60 Hz, this noise and other high frequency noise can be filtered out by a low-pass filter. Low frequency drift due to respiration can be reduced by applying a high-pass filter. However, abnormal QRS shapes and artifacts due to electrode motion require the application of more sophisticated methods and algorithms to reduce their effect on the heart rate reading. Therefore, while implementing a wearable ECG monitoring unit (in the same manner as Polar USA has already done) may be a suitable technological option for the PSS heart rate sensor, the ECG monitoring method as a whole does not appear to offer any new or significant advantages over other technologies for heart rate monitoring in the PSS.

The characteristics of the IR light monitoring method made IR monitoring an interesting and appealing choice to explore for use in the PSS. The use of IR technology eliminates the need for electrodes and conductive gels that are required for ECG monitoring, as well as reducing set-up time. Since the IR monitoring technology is based on detecting the levels of IR light reflected by the skin over time, the system is completely non-invasive, only requires that an IR emitter and detector (a phototransistor) be placed near the skin, and is consequently low-power. The IR emitter and detector are inexpensive, which keeps the total cost of the sensor itself very low. Although current

commercial systems using IR monitoring technology apply the sensors to the fingertip or earlobe for measuring heart rate, for the PSS design the assumption was made that neither of these places were available for sensor positioning. Therefore, applying IR technology to other areas of the body, such as the wrist or collarbone, probably requires more extensive or innovative signal processing. The end product will ensure that the PSS could be used in both military and non-military applications.

Strain gage technology was the main method pursued for measuring the respiration rate. HPI polyimide carrier HBP series strain gages were used initially due to their small size (25.4 mm x 12.7 mm x 0.127 mm) and possibility for use with many materials (Hitec Products Inc., Ayer, MA). Although the HBP series strain gages performed according to their specifications, with a small change in resistance as a force was applied on the ends, the rigid nature of the strain gage was its major weakness with regard to measuring respiration. During respiration, the thoracic cavity expands and contracts. A strap containing a rigid strain gage would not allow for this natural expansion of the chest during respiration, thereby making it necessary for part of the strap to have elastic properties. However, by allowing the elastic part of the strap to absorb the bulk of the chest expansion, the force on the strain gage would drastically decrease, effectively rendering the strain gage useless. Additionally, the largest change in resistance was measured when the force involved was bending the gage, not just stretching it. Since the force of the chest expanding is relatively evenly distributed, there is little chance that the strain gage would be bent as a result of this force.

Elastic fabrics, such as Lycra, coated with a layer of conducting polymer can also act as flexible strain gages [16, 17]. Polypyrrole (PPy) is an electron conjugated

conducting polymer that is environmentally stable as well as capable of mechanical and thermal transduction [18]. By chemically and then electrochemically depositing pyrrole on a strip of stretchable Lycra, a conducting PPy film can be created that allows the Lycra to act as a strain gage [16, 19, 20]. Although conductive polymer strain gages are still a relatively immature technology, a novel application of PPy Lycra strain gages explored in this research project was their use for respiration rate monitoring. The concept involves attaching a PPy Lycra strain gage to a non-elastic strap that allows the displacement of the chest as it expands and relaxes to stretch and contract the conductive PPy Lycra strain gage material. Since the strain gage would absorb most of the force of expansion during respiration, the change in resistance is also likely to be larger than that measured with the rigid HBS series strain gage. These characteristics of the PPy Lycra strain gage made it the technology of choice in the PSS.

## **Wireless Tactile Control Unit**

The WTCU was developed to control a tactile display consisting of a 3-by-3 array of vibrating motors. This array of motors is worn on the lower back so that the user can feel the different vibration patterns. Each pattern is associated with a navigational or informational command, such as move to the left or stop, given by an operator who uses a notebook computer to transmit the commands using Bluetooth technology [14]. The WTCU is worn by the user, and the Bluetooth module in the WTCU receives the command and passes the information along to the Atmel AT90LS8535 microcontroller [21]. The microcontroller translates the command into a pre-programmed vibration

pattern and vibrates the motor array in the required pattern. No feedback from the user is implemented in the WTCU system.

In order to keep the whole system low power, the PSS design must interface with the WTCU. The WTCU uses a +7 V to +9 V power source in the form of a battery, and ideally the PSS will use the same power source. Additionally, the PSS can make use of the same microcontroller technology as the WTCU, adding new commands to those recognized by the microcontroller in order to record and analyze the heart pulse and respiration patterns of the user. The wireless component of the PSS can also be connected to the WTCU by using the same Bluetooth module and expanding the Bluetooth functionality to include the necessary feedback transmissions to provide the operator with the ability to monitor the vital signals of the user.

## **Microcontroller**

The PSS will make use of the open pins on the Atmel AT90LS8535, the microcontroller implemented in the WTCU, for analyzing the physiological signals processed by the PSS. The AT90LS8535 has I/O pins for sensor interfacing, a UART interface, an analog-to-digital converter (A/D), and interrupt pins that run a programmed sequence when an interrupt is triggered [21]. This microcontroller also features a fast clock speed, 16-bit timers for long delays and timing events, and low-power sleep modes [21]. It is readily available and is manufactured in 44-lead TQFP and PLCC surface mount packages [21]. It can be programmed in assembly language or C (using a compiler).

The PSS will use only the two interrupt pins (INT0 and INT1) on the AT90LS8535 ,since the sensor interfacing will be done on the PSS board. One interrupt pin will be used in conjunction with the heart rate sensor, and the other will be used similarly with the respiration rate sensor. Since the majority of the signal processing will be done in analog on the PSS board, the interrupt pins will be used to perform the pulse counting necessary to determine the heart and respiration rates. The AT90LS8535 code for the WTCU motor array has been programmed in the chip's assembly language, and so the code for the PSS bio-monitoring will also be written in assembly language. Further additions to the code will be necessary to implement Bluetooth wireless communication.

## **Bluetooth**

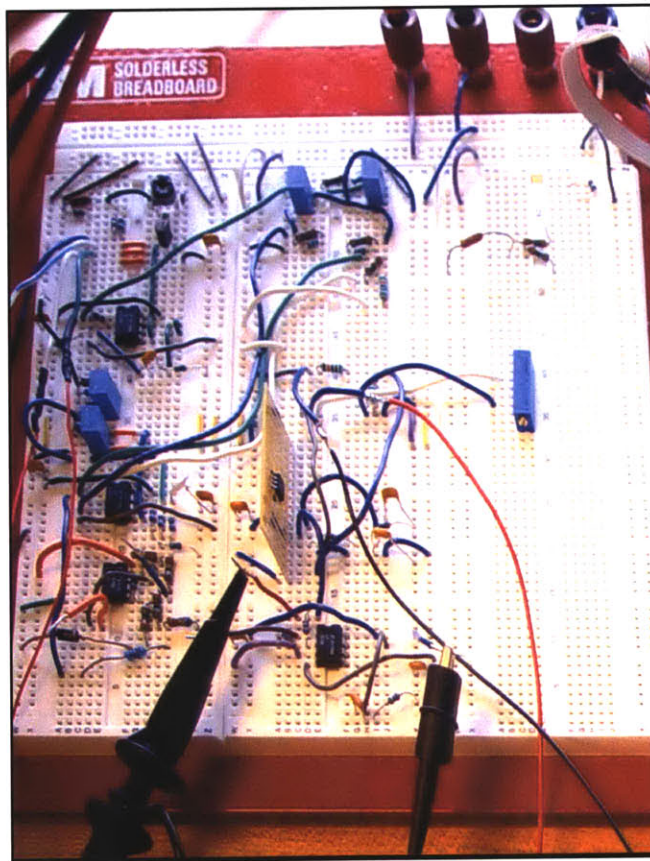
Bluetooth is a wireless technology that uses a spread spectrum, frequency hopping, full duplex signal at up to 1600 hops/sec to make connections between sensors and portable devices [5]. The Bluetooth spread-spectrum 2.4 GHz technology modules (BlueRadios, Castle Rock, CO) should be capable of high-speed communication of up to 721 kbps over air with transmission security via 128-bit encryption and a 10-digit PIN [22]. These modules are reported to have a 100-meter line-of-sight outdoors [22]. They can only support seven or fewer users, which may eventually be an issue for large-scale applications.

The AT90L8535 code will need to be augmented to receive ASCII characters from the Bluetooth wireless module that represent heart rate and respiration rate queries, as well as to send the resulting heart or respiration rate information back to the Bluetooth module for wireless transmission to a notebook computer. Bluetooth can be used with

Microsoft Windows if the operator on the computer has the correct software interface that provides the list of necessary commands. The software will need to interface with the computer's COM port, communicating via the RS-232 protocol, and manage the transmission and reception of characters and data [22]. The Bluetooth module plugs into the computer's USB port and creates a virtual COM port in software that allows the appropriate communication. The graphical use interface (GUI) and some of the Bluetooth commands have already been programmed (by previous UROP and M.Eng. students), so modifications and additions will be made mainly for the bio-monitoring sensing system.

# Prototype Design and Construction

The Physiological Sensor System prototype was initially constructed on a breadboard in order to facilitate modification of the design after preliminary testing. The temporary version of the circuit created on the breadboard was later transformed into the printed circuit board version. Larger dual in-line package (DIP) parts were required for use in the breadboard, whereas compact surface-mount parts (SMD) were necessary for the printed circuit board. The DIP packages are designed to fit into the breadboard and have identical pin layouts to their corresponding surface-mount packages, allowing for complete circuit construction on the breadboard. *Figure 1* shows the breadboard prototype during the design phase.



**Figure 1: Breadboard used for prototype design.**



# Heart Rate Sensor

## First Prototype

The goal of the first prototype was to establish a fundamental signal processing unit that could isolate the heart-beat signal. The design included a Radio Shack 276-142 matched IR emitter and detector, a 271-282 10 K $\Omega$  potentiometer, and a 276-2009 MPS2222A NPN transistor. The 276-142 IR emitters and IR detectors are 5 mm in diameter and 8.7 mm high; therefore, it was clear that they could not be used in the final design due to their size and shape. However, they were easily accessible and provided a starting point to test the IR technology.

A basic circuit for a simple heartbeat transducer involves an IR emitter and detector, transistor, potentiometer, two resistors, and a +5 V power source [23]. The circuit shown in *Figure 2* was constructed using the specific parts listed above and a power generator for the +5 V power supply. In order to focus the IR light projection and detection, simple opaque tubes cut from heat-shrink material were taped around the emitter and detector. The resulting signal,  $V_I$ , was attained initially by placing the IR emitter on the nail-side and IR detector on the pad-side of the index finger, modeled after commercial systems. Similar results were captured on an oscilloscope when the emitter and detector were placed approximately 15 mm apart lengthwise on the pad of the index finger. The output signal,  $V_I$ , of this simple heartbeat transducer used on the pad of the index finger is shown in *Figure 3*.

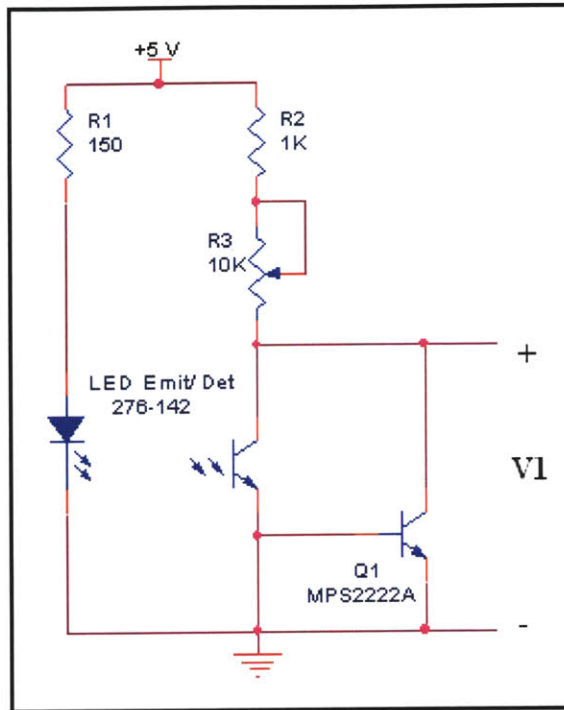


Figure 2: Heartbeat transducer circuit with output V1.

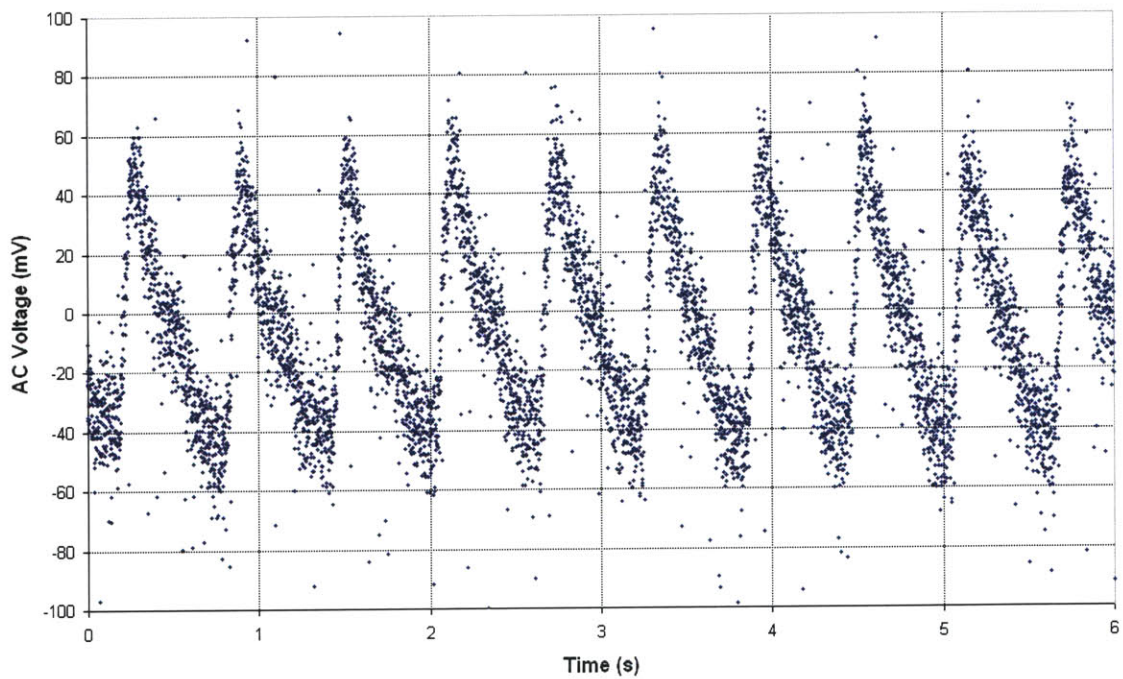
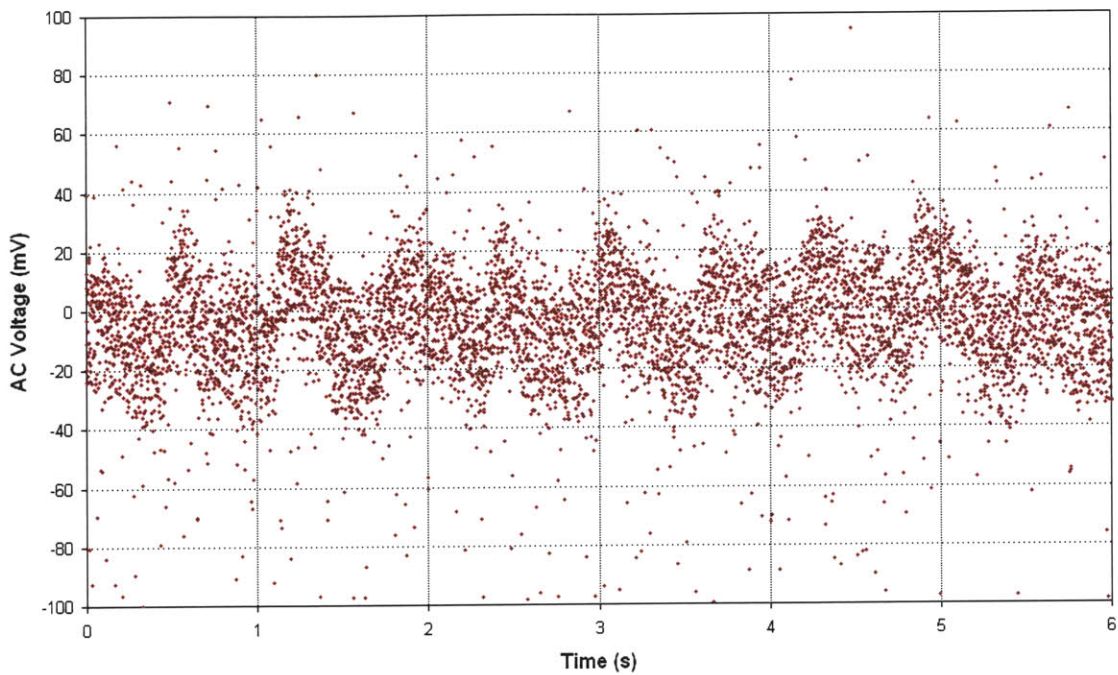
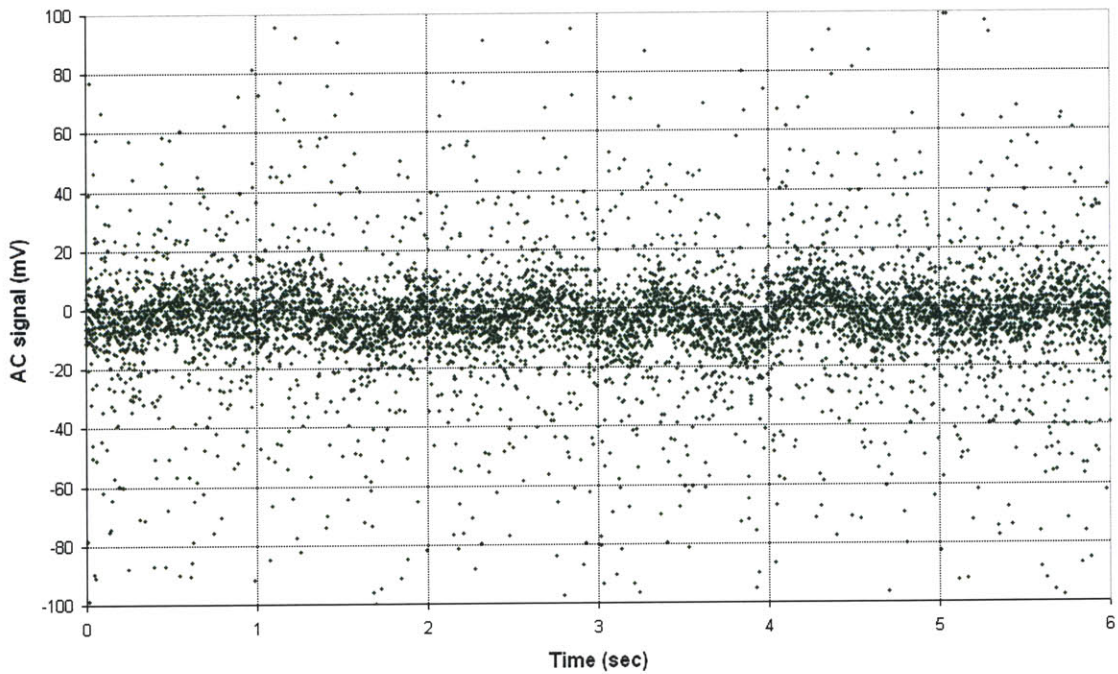


Figure 3: Unprocessed output (V1) of the heartbeat transducer circuit with the IR sensor applied to the index finger-pad.

Each of the five fingers was tested in order to determine the variability of this technology. The resulting signals showed that the index and middle fingers provided stronger signals than the others. The thumb provided the worst signal, probably due to added noise from the tissue on the thumb pad. Although some of the fingers provided adequate signals, it was necessary to look at other sites on the body, as the hands and fingers are required to perform many activities. Sites on the body tested with the circuit in *Figure 2* included the wrist, arm, leg, toe, ankle, knee, shin, lower back, neck, ear, upper chest, and collarbone. No detectable pulses were found in any of these locations when viewed with an oscilloscope on its highest (20 mV) gain because the high frequency noise dominated the signal. Though not visible on the oscilloscope, data captured from the wrist and collarbone did show some signs of the pulse. *Figure 4* and *Figure 5* show the output signal *VI* of the heartbeat transducer used on the wrist and collarbone, respectively. If the high frequency noise could be eliminated successfully, then one of these two sites could be used as the location for measuring the heart rate. Therefore, it was necessary to do further signal conditioning to improve the signal before determining the best site for the heart rate sensor placement.



**Figure 4: Unprocessed output (V1) of the heartbeat transducer circuit with the IR sensor applied to the left wrist.**

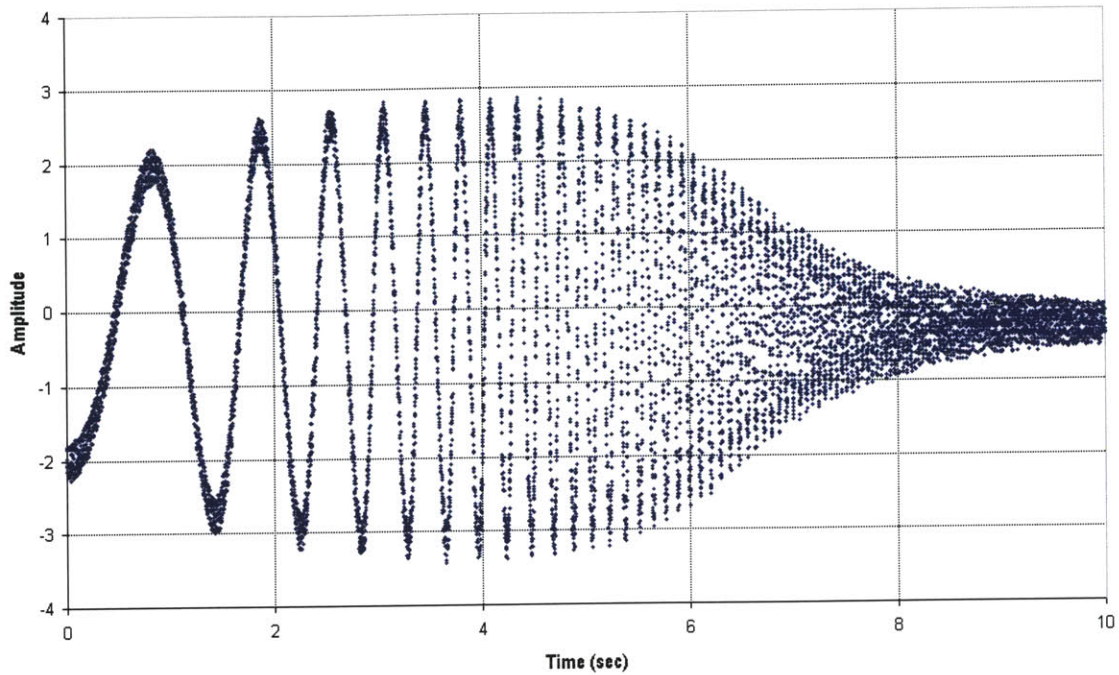


**Figure 5: Unprocessed output (V1) of the heartbeat transducer circuit with the IR sensor applied to the collarbone.**

A Butterworth second-order low-pass filter was designed to reduce the high frequency noise. Before the low-pass filter could be applied, however, the transducer output signal needed to be buffered, the DC bias eliminated, and the AC component amplified. Buffering would prevent other aspects of the circuit, especially feedback loops, from having undesirable effects on the transducer circuit that would change its output. After the buffer circuit, a simple RC high-pass circuit was implemented to remove the approximately 2 V DC bias from the signal. This was a necessary step before the amplification circuit in order to avoid damaging the rest of the system by sending forward an extremely large DC voltage. The RC circuit consisted of a 0.1  $\mu\text{F}$  ceramic capacitor in series between the buffer output and the amplifier input and a 1  $\text{M}\Omega$  resistor from the end of the capacitor to ground. Once the DC bias was eliminated, the AC component would then become the DC component, and would be amplified by the voltage-controlled voltage source (VCVS) amplifier. The low-pass filter also had a gain value, so the final amplification amount was the VCVS gain multiplied by the filter gain.

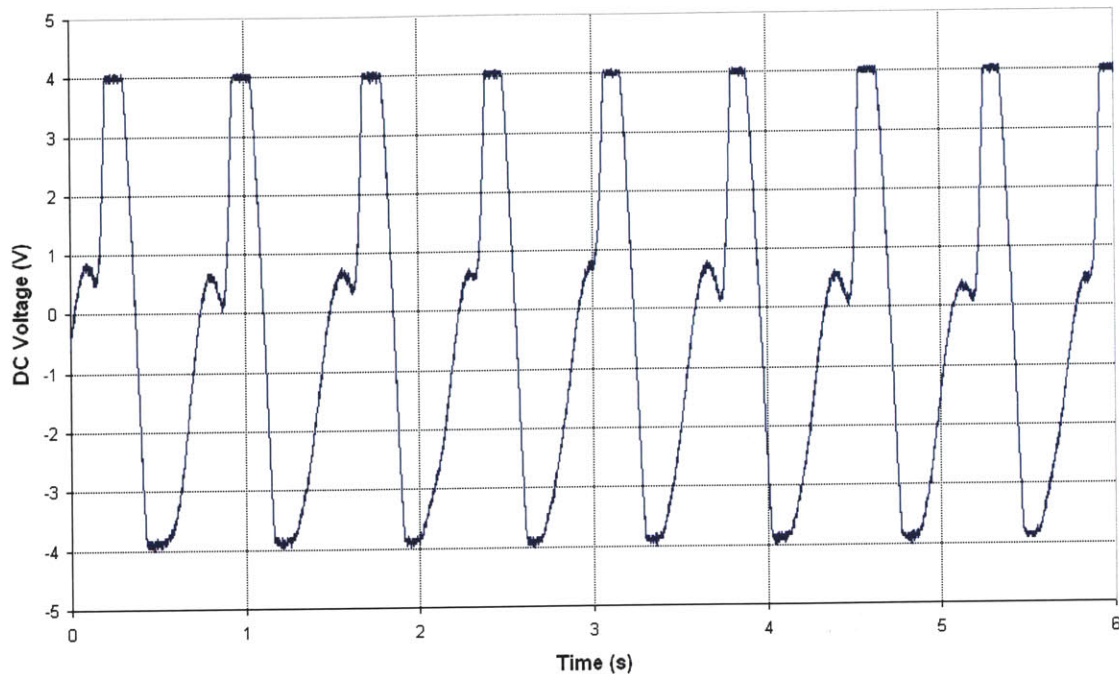
Since the normal heart beat range is 40-120 beats per minute (bpm) and the exercising heart rate does not tend to go much above 200 bpm, the maximum heartbeat rate would be around 3.33 Hz [1]. Therefore, 10 Hz appeared to be a reasonable cutoff frequency to start with in the low-pass filter to attenuate high frequency noise while avoiding the reduction of higher frequency components that may contribute to the heartbeat signal peaks. The properties of the low pass filter constructed were tested by sending a frequency sweeping waveform through the filter. The resulting attenuation pattern is shown in *Figure 6*. Attenuation of the signal begins near 5 Hz and becomes severe above 10 Hz.





**Figure 6: Output of Butterworth second order low-pass filter with frequency sweeping input. The cutoff frequency of the filter is designed to be 10 Hz; however, the actual attenuation begins near 5 Hz since the filter is second order.**

DIP versions of the high-speed LF353 series op-amps were implemented for the buffering and VCVS amplification circuits, due to their low supply current requirements (3.6 mA) and high slew rates (13 V/us) [24]. An LF353 series op-amp was also used to create the Butterworth low-pass filter because of its additional low noise properties. To further reduce interference by noise, ceramic 0.1 uF de-coupling capacitors were positioned between power and ground at each place in the circuit that an op-amp was connected to a power source. The low-pass filter was designed to have a gain of 6, and the VCVS was designed to have a gain of 40. *Figure 7* shows the resulting signal at the output of the low-pass filter. The signal is free of noise, and the individual beats of the heart are very clear.



**Figure 7: Noise-free output of Butterworth second order low-pass filter with the IR sensor applied to the index finger-pad.**

## Second Prototype

The success of the first prototype provided a promising basis for the next prototype. The two main signal processing goals of the second prototype were to find an area on the body other than the finger that could be used as a site to measure the heart rate signal using the IR light method, and to create a digital signal as the final output that could be transmitted to the microcontroller. A further goal was to create a wearable device by finding and implementing flat IR emitters and detectors that could make full contact with the skin, instead of the round ones used in the first prototype.

The first set of flat IR emitters and detectors that looked promising were the Fairchild Semiconductor QEB421 IR emitter and QSB320FTR IR phototransistor. Emitters such as the QEB363GR and QEB441TR, and detectors such as the QS320,

QSB34GR, and QS363GR, have similar characteristics to the QEB421 and QSB320FTR, respectively; however, the QEB421 emitter and QSB320FTR detector are matched with peak wavelengths of 880 nm [25, 26], and the QSB320FTR detector has a narrower detection range (700-1000 nm) than the QSB320 (400–1000 nm) [26, 27], making the QEB421 and QSB320FTR the best pair. Both the QEB421 and QSB320FTR are packaged in an opaque casing. The emission and detection angles of both are wide at 120 degrees, and their dimensions are the same, approximately 3.2 x 2.8 x 2.0 mm.

An initial observation from the use of the flat IR emitter and detector was that in order to record a signal, the IR emitter and detector needed to be placed about 15-25 mm apart. If they were placed closer together or farther apart, the probability of capturing the heart pulse signal was low. However, once the signal was captured, the small, flat IR emitter and detector performed just as well as the Radio Shack 276-142 pair, except that the amplitude of the signal was about half of the 276-142 pair. This was easily resolved by doubling the VCVS amplifier to 80.

The next flat IR emitter and detector pair considered were the Digikey 365-1043 IR emitter and the 365-1070 IR detector. Another emitter 365-1045, and detector 365-1073 have similar properties to the 1043/1070 pair; however, the 1043 emitter is matched to the 1070 and 1073 detectors, and the 1070 detector contains only one phototransistor whereas the 1073 contains two [28, 29]. Therefore, since the 1070 would be able to replace directly the other phototransistor in the heartbeat transducer circuit already developed, the 1043/1070 pairing was chosen.

Although very similar to the QEB421/QSB320FTR pair, the benefits of the 1043/1070 pair include lower maximum forward current, less maximum power

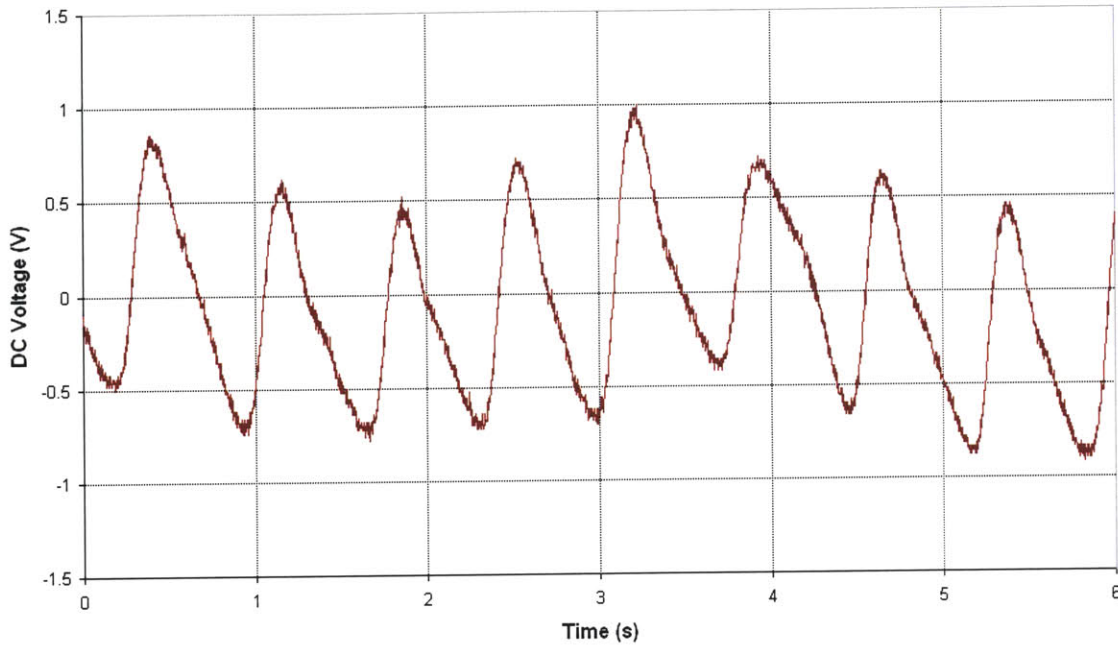


dissipation, reduced thickness, and larger surface area. The 1043 maximum continuous forward current is 50 mA [30] as compared to 100 mA for the QEB421 emitter [25]. Both the 1043 and 1070 have 100 mW maximum power dissipation [28, 30], whereas the QEB421 has 180 mW and the QSB320FTR has 165 mW [25, 26]. Additionally, the 1043/1070 dimensions are 5.75 x 4.45 x 1.6 mm, providing a larger and thinner flat surface [28, 30]. The only disadvantage of the 1043/1070 pair is that they are both packaged in a clear case, making it more difficult to keep the IR light that is being emitted away from the skin from also being detected by the phototransistor, thereby corrupting the heart pulse signal.

Although both the QEB421/QSB320FTR and 1043/1070 pairs were suitable for use with the PSS, the 1043/1070 was pursued further due to its size and electrical properties. The 1043/1070 pair must also be placed 15-25 mm apart to receive the strongest signal. When using the 1043/1070 pair, the amplitude of the heart pulse signal was double that of the QEB421/QSB320FTR pair, which means that the 1043/1070 signal amplitude is comparable to that of the original signals captured by the large Radio Shack 276-142 pair.

When the 1043/1070 emitter-detector pair was used in conjunction with the first prototype circuitry, it proved successful in capturing the heart pulse signal from the finger, provided that the emitter and detector were isolated in a low-light area. Additionally, repeated testing showed that the amplification and filtering of the first prototype circuitry was sufficient to capture a strong heart pulse signal from the upper wrist, as shown in *Figure 8*. The amplitude of the wrist heart pulse signal is lower than that of the finger, as expected. However, the pulsating nature of the signal is still very

clear. It was also possible to measure a pulse from the collarbone and forehead; however, these areas did not produce as consistent data as the finger and wrist.



**Figure 8: Output of Butterworth second order low-pass filter with the IR sensor applied to the upper wrist. Note the low-frequency baseline drift causing the peaks to range from ~0.4 V to 1.0 V.**

The initial wrist pulse signal was amplified a total of 240 times by the output of the low-pass filter; therefore, minor fluctuations in the initial pulse signal amplitude due to a low-frequency baseline drift were exaggerated at the output of the low-pass filter. The finger pulse had been so much larger in amplitude that the amplified signal was saturated, which obscured the effect of the baseline drift. A simple solution was to replace the low-pass filter by a band-pass filter to help attenuate the baseline drift. Additionally, the high-side cutoff frequency of the band-pass filter was lowered from the 10 Hz used in the low-pass filter to help filter out as many movement artifacts as

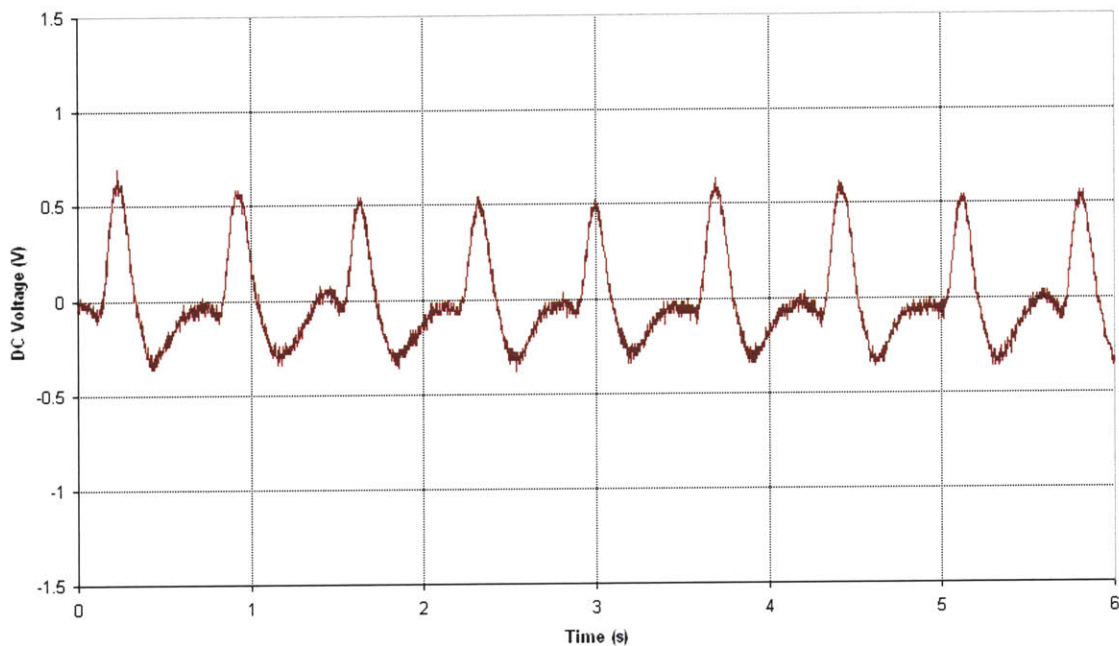
possible, since the movement artifacts had sharper peaks (higher frequency) than the heart pulse.

Since the heart rate is most commonly in the 1.0-3.3 Hz (60-200 bpm) range, the band-pass filter was designed to pass the 0.75 – 3.75 Hz range, with a center frequency of 2.25 Hz and a bandwidth of 3 Hz. The quality factor, determined by dividing the center frequency by the bandwidth, is low (0.75) because the pass-band is relatively wide. According to the equations governing the second-order band-pass filter, the  $\mu$  factor must be between 0 and 4 [31]. By choosing  $\mu$  to be 3, a gain of 3 results, since  $\mu$  is related to the gain of the filter by  $\mu/(4-\mu)$ . In order to keep the total amplification the same as when the low-pass filter of gain 6 was implemented, the VCVS amplification was increased to 80 to bring the total amplification of the VCVS and band-pass filter back to 240.

The final step in processing the heart rate signals was to turn the signal into a digital output that could be sent to the microcontroller. The desired digital signal would alternate between 0 V and +5 V in a pattern indicating the heart pulses. A comparator circuit was chosen for this so that the output would flip from one saturation limit to the other based on the results of comparing the input signal to a reference voltage. The DIP version of the LT1498 rail-to-rail dual precision amplifier was implemented for the comparator circuitry, with 0 V and +5 V rails [32].

The input signal needed to be a positive signal, unlike the output signal of the band-pass filter which could span the entire -5 V to +5 V range. A diode was inserted between ground and the band-pass circuit output (also considered the comparator circuit input) in order to eliminate the negative parts of the band-pass output signal. Since the

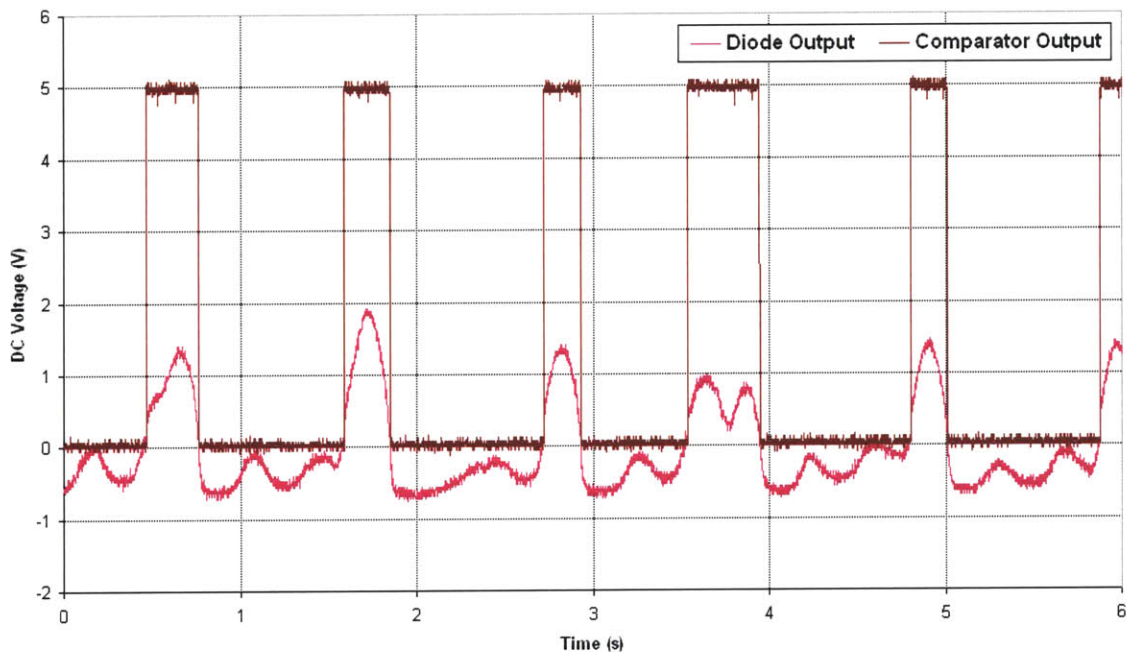
anode was connected to ground and the cathode to the band-pass output, the diode would be on whenever the band-pass output signal was negative. When the diode was on, there was a forward voltage drop across the diode of approximately 0.5 V. The cathode, connected to the input of the comparator circuitry, would therefore remain near -0.5 V during the portions of the band-pass output signal that went negative, thereby reducing any large negative values to a value close to 0 V. *Figure 9* shows the upper wrist pulse after the band-pass filter and diode. The baseline drift has been reduced by the band-pass filter, and the signal no longer goes below approximately -0.5 V



**Figure 9:** Output after the band-pass filter and diode with the IR sensor applied to the upper wrist. The band-pass filter greatly reduces the baseline drift (seen in Figure 8) for a more stable pulse amplitude, and the diode eliminates the negative signal below approximately -0.5 V.

The comparator circuitry was designed with a relatively low voltage reference in order to measure the occasionally weaker pulses. The voltage reference was set to 0.8 V, which meant that whenever the comparator's input signal was above 0.8 V, the

comparator would produce a +5 V output, and whenever the signal was below 0.8 V, the comparator would produce a 0 V output. To avoid erroneous triggering due to the op-amp's input offset voltage, and to reduce the possibility that anomalies in the signal could cause erroneous triggering, hysteresis was added to the comparator circuit. The hysteresis value was set to 0.7 V so that the comparator output would go high when the input became larger than 0.8 V, but would not go low until the input went below 0.1 V (the voltage reference minus the hysteresis value). *Figure 10* demonstrates how the comparator interprets the heart pulse input signal. A block diagram of the PSS heart-rate sensing process is shown in *Figure 11*.



**Figure 10: Lower wrist heart pulse comparison at the diode output and comparator output. Due to the reference voltage and hysteresis levels chosen, aberrations such as the one seen between time 3 s and 4 s do not cause an extra beat to be registered by the comparator.**

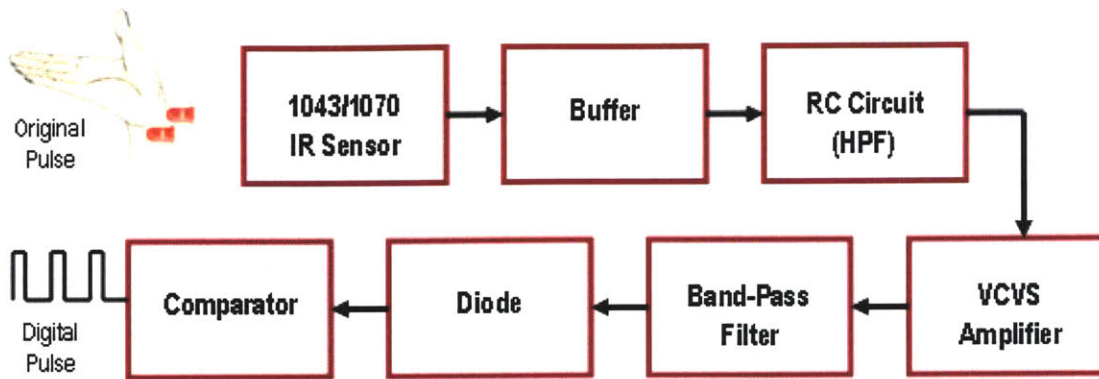


Figure 11: Block diagram of PSS heart rate sensor process to produce a digital pulse signal.

## System Results

The PSS heart rate sensor was compared to the Polar M32 heart rate monitor to evaluate its validity. To calculate the PSS heart rate, the times between the rising edges of the five most recent digital pulses was averaged, and the heart rate (in bpm) was calculated by taking the inverse of average time (in seconds) per pulse and multiplying by 60. The PSS heart rate was then rounded to the nearest integer to be comparable to the Polar M32 readings. Heart rate readings were taken while stationary and after a range of physical activities. *Table 1* shows a summary of the results.

The PSS heart rate sensor using IR technology produces the same result as the Polar M32, with an average difference of 1.1%. This percentage difference is extremely small, indicating that the PSS heart rate sensor is as successful at determining the heart rate as commercial systems.



<b>PSS Heart Rate Sensor</b>	<b>Polar M32</b>	<b>Difference</b>	<b>Percentage Difference</b>
87	88	-1	1.1%
90	88	+2	2.3%
95	94	+1	1.1%
97	97	0	-
103	103	0	-
113	114	-1	0.9%
122	121	+1	0.8%
136	139	-3	2.1%
141	140	+1	0.7%
156	153	+3	2.0%
<b>Average Percentage Difference</b>			<b>1.1%</b>

**Table 1: Comparison of the PSS heart rate sensor and Polar M32 in determining the heart rate. The average percentage difference between the rounded PSS and Polar M32 readings is 1.1%.**

## **Respiration Rate Sensor**

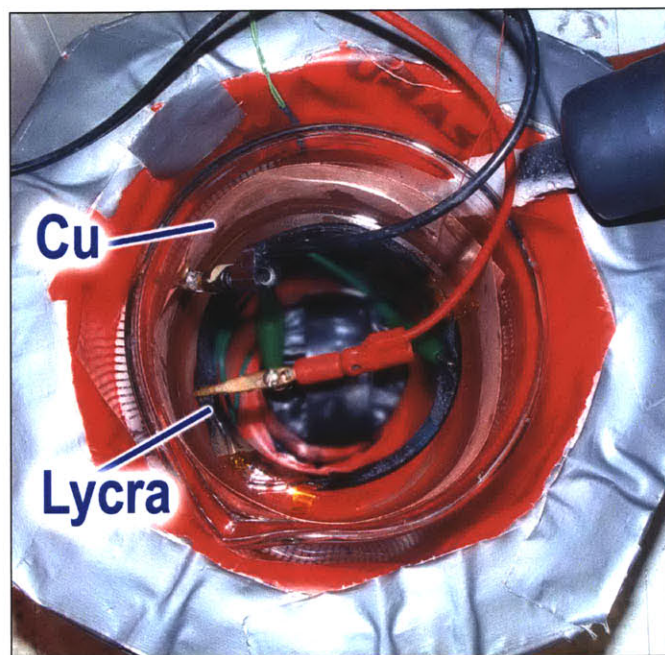
### **Creating Polypyrrole Lycra Strain Gages**

The chemical formula of the pyrrole monomer is  $C_4NH_5$  [19]. The polymer, PPy, is a conducting polymer that can potentially be used as a muscle-like polymer actuator, strain gage, or organic transistor [17]. Its limitations depend on its material properties. Although pure conducting polymers can be used as strain gages, better strain gages have been created by coating a flexible elastic fabric with the polymer [20, 33, 34]. The conformability, high possible strain, and low stiffness make this material an excellent candidate for use as a flexible strain gage, and it has been utilized in research devices such as position sensing gloves [35]. These PPy Lycra strain gages are created by performing a chemical deposition, followed by an electrochemical deposition that improves the conductivity of the polymer coating.

To create the strain gage for the PSS respiration rate sensor prototype, conducting polymer PPy was deposited onto Lycra fabric [16, 20]. The chemical deposition step coats the Lycra with chemically oxidized PPy. Two aqueous solutions of equal volume were prepared for use in the chemical deposition. The first was a 0.046 M ferric (III) chloride ( $\text{FeCl}_3$ ) solution [19]. The second was a 0.02 M pyrrole and 0.006 M 1,5 Naphthalenedisulfonic acid tetrahydrate solution [19]. After the two solutions were mixed, the Lycra was immersed in this mixed solution, and the container with the solution and Lycra was stored in a 4° C refrigerator. The solution was stirred once during the first 2 hours and was then allowed to sit overnight.

After the chemical deposition, the coated Lycra strips were rinsed with distilled water to remove any loosely deposited chemical. For the electrochemical deposition, the solution was formed by combining an appropriate volume of propylene carbonate (and 1% of the propylene carbonate volume of water), 0.05 M pyrrole, and 0.05 M tetraethylammonium hexafluorophosphate (TEAP) [19]. The coated Lycra strip was taped to a small beaker and immersed in the solution. A copper sheet, serving as the counter electrode, was placed in the solution completely surrounding the small beaker with the coated Lycra. The electrochemical deposition, setup shown in *Figure 12*, was carried out at approximately -40° C for 14-18 hours and was voltage-controlled at 3 V. The properties of the deposition over time can be seen in *Appendix A, Figure A-1*.





**Figure 12: Electrochemical deposition setup.**

### **Properties of Polypyrrole Lycra Strain Gages**

Although the electrochemically grown PPy improved the conductivity and mechanical properties of the chemically coated fabric strip, there was considerable variability between PPy Lycra strain gages made in different batches. The average unstrained resistance (measured from end to end) of the four strips produced in Batch Three was almost 100 times that of Batch Two. Within the same batch, there was significant variation between different strips as well as between the results of the same strip as it was repeatedly stretched. Two strips from Batch Three were stretched at a constant rate of 100 mm per minute, and *Figure 13* shows their changes in resistance during this process. The probes were placed 15 mm apart lengthwise on the strip at the beginning of each trial, and the initial resistance was recorded for some time before the stretching device was activated (at the 0.0 mm mark in *Figure 13*). Since the stretching

rate was held constant and the time was recorded, the stretched distance could be determined.

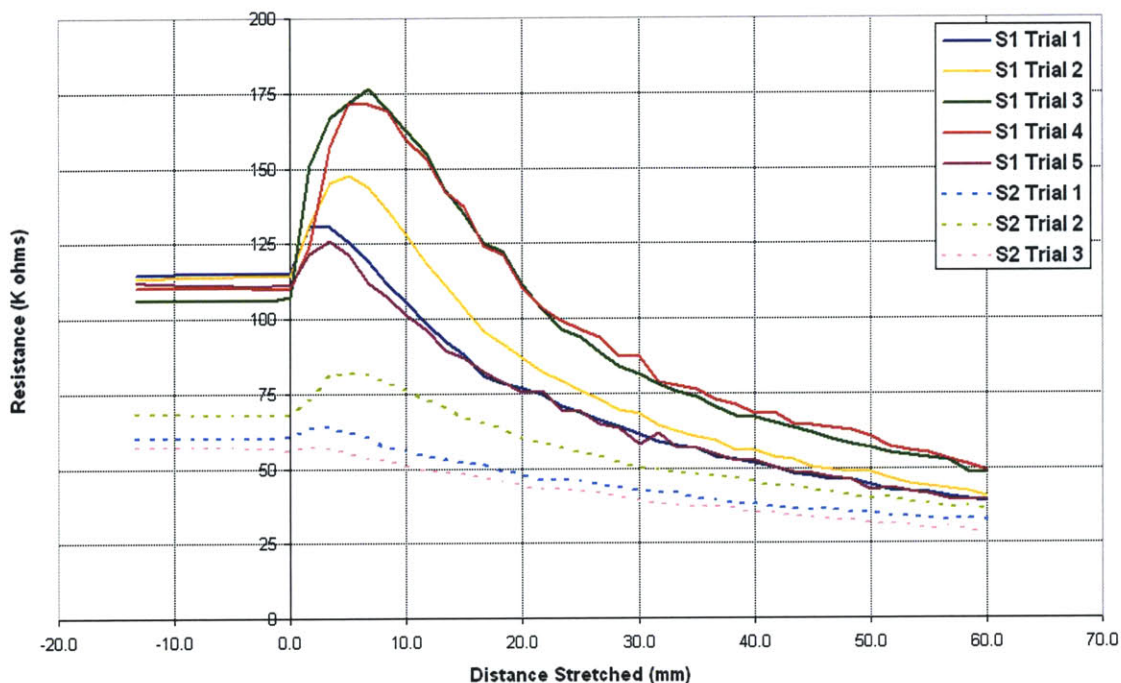
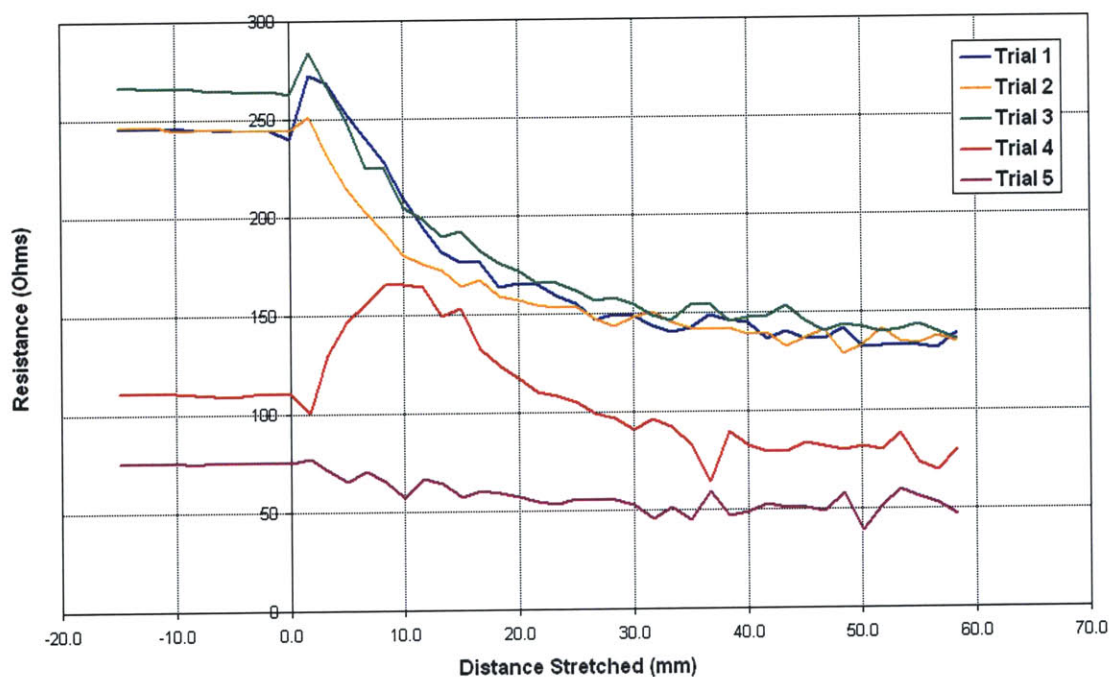


Figure 13: PPy Lycra Strain Gage (Batch Three, Strip 1 and Strip 2) changes in resistance as a function of the distance stretched. Both the strips are from Batch Three. Strip 1 (S1) trials are shown in solid lines and Strip 2 (S2) trials are shown in dashed lines.

Although there was significant variation, the initial resistances for each strip remained in the same region for each trial, and all of the curves were similar in shape. For each strip and each trial, there was an initial spike in the resistance immediately after the stretching begins, followed by decreasing resistance values as the strip continues to be stretched. The values appear to follow the formula  $R = \rho \times L/A$  (resistance R, resistivity  $\rho$ , length L, area A). As the area of the strip is increased during the stretching, the resistance decreases. However, it appeared that although the initial resistances between strips (measured with approximately the same probe separation) were different, the

resistances for the completely stretched strips were in the same vicinity. The polymer-stretching machine, built in the MIT Bioinstrumentation Lab, was used to stretch the PPy Lycra strips. Due to the size of the machine, it could stretch the strips up to 60 mm.

While testing two strips in Batch Two, it was noticed that if the probes were placed so close together that the initial resistance was less than the stretched resistance of the majority of strips, then the variability in the properties of the same strip dramatically increased. *Figure 14* shows 5 trials of one strip in Batch Two. The first three trials followed the same procedure as outlined previously, and the results of the trials are similar. Trials 4 and 5 followed the same procedure except that the probes were placed much closer together to reduce the initial resistance to be below the average stretched resistance of the first three trials.



**Figure 14: PPy Lycra Strain Gage (Batch Two, Strip 1) changes in resistance verses distance stretched. Trials 4 and 5 have initial resistances below the strained resistance limit of the first three trials.**

The gage factor is a value that relates the strain gage's resistance value to its strain. It is determined by the relationship  $(\Delta R / R) / (\Delta L / L)$  [19]. For the PPy Lycra strain gages created, the gage factor was affected by the location of the probes measuring the resistance on the strip, as well as the stretch history, size, and tension of the strip. However, even though the unstrained resistances for Batch Two and Batch Three vary by a factor of 100, their gage factors (with probes attached 15 mm apart and for a 10.0-20.0 mm stretch) are all within the same general range, as shown in *Table 2*. Therefore, despite the considerable variation in these conducting polymer strain gages, it is possible to use the strain gages to provide some measure of strain, especially when the measurements do not have to be exact.

	<b>Batch Two</b>	<b>Batch Three</b>
<b>Strip 1, Trial 1</b>	1.4509	2.0051
<b>Strip 1, Trial 2</b>	0.7627	2.8883
<b>Strip 1, Trial 3</b>	0.9750	3.8710
<b>Strip 1, Trial 4</b>	<i>3.4698</i>	3.6012
<b>Strip 1, Trial 5</b>	<i>0.0134</i>	1.8341
<b>Strip 2, Trial 1</b>		1.0485
<b>Strip 2, Trial 2</b>		1.8963
<b>Strip 3, Trial 3</b>		0.9711

**Table 2: Gage Factors of the PPy Lycra strain gages shown in Figures 13 and 14. Strip 1, Trials 4 and 5 give noticeably different results due to their inappropriate probe placement discussed earlier.**

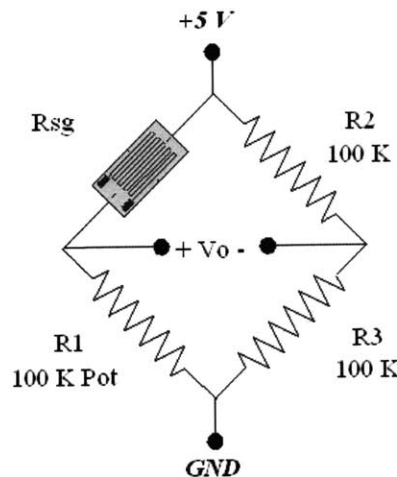
## Primary Prototype

To measure respiration rate, it is not necessary to know exactly how much the strain gage is being stretched. All that is required is that the changes in resistance trigger the system appropriately to record the respiration rate. The circuitry designed for the PSS respiration rate sensor takes into account the large variability between PPy Lycra strain gages and focuses on simply creating a pulsating signal that corresponds to the breathing



pattern. The first step was to create a bridge circuit that would capture the changes in resistance as changes in voltage. The changes in voltage were then amplified and a comparator was again used to create a digital output signal representative of the respiration pattern.

A Quarter Wheatstone bridge was implemented to capture the resistance changes in the strain gage. As shown in *Figure 15*, a +5 V source is applied to one end of the bridge while the other is connected to ground. The output voltage is measured across the middle of the bridge, and the bridge is considered balanced when  $R_{sg}/R1 = R2/R3$ .

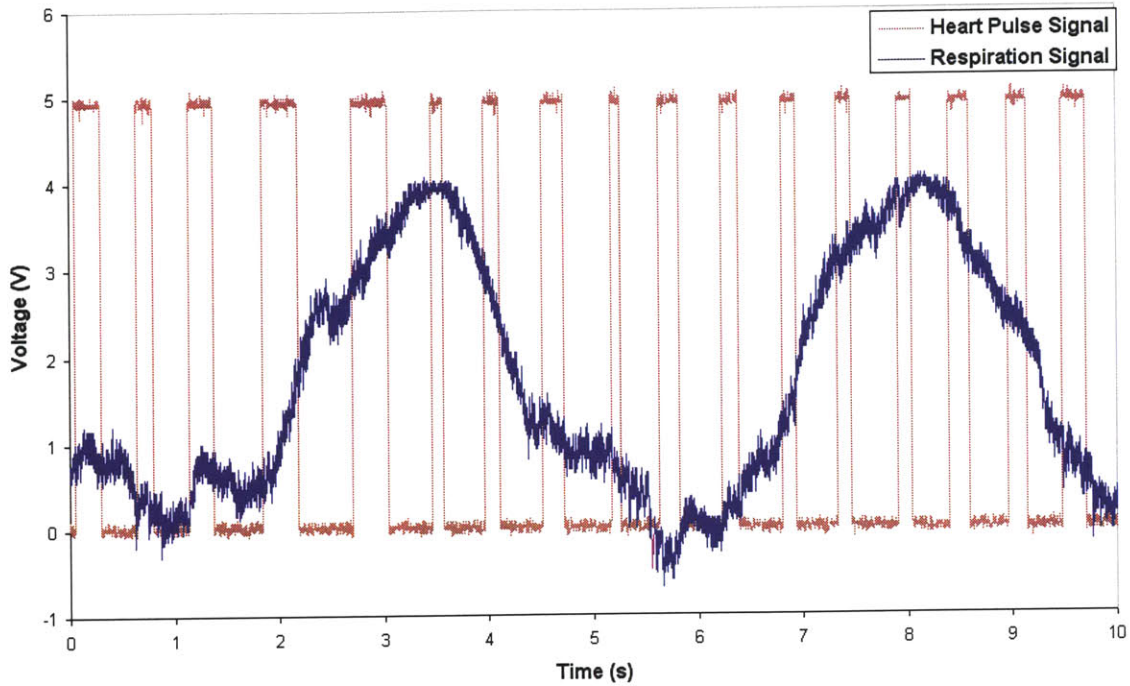


**Figure 15: Quarter Wheatstone bridge circuit used in the PSS Respiration Rate Sensor circuit.**

Due to the large variation in resistances possible for the strain gage, a 100 K $\Omega$  potentiometer was used as the bottom resistor in that branch. The other branch was made up of two 100 K $\Omega$  resistors. Therefore, for any initial strain gage resistance value between 0-100 K $\Omega$ , the bridge could be balanced by matching the potentiometer to the strain gage resistance. Additionally, since the resistance on the strain gage varies based

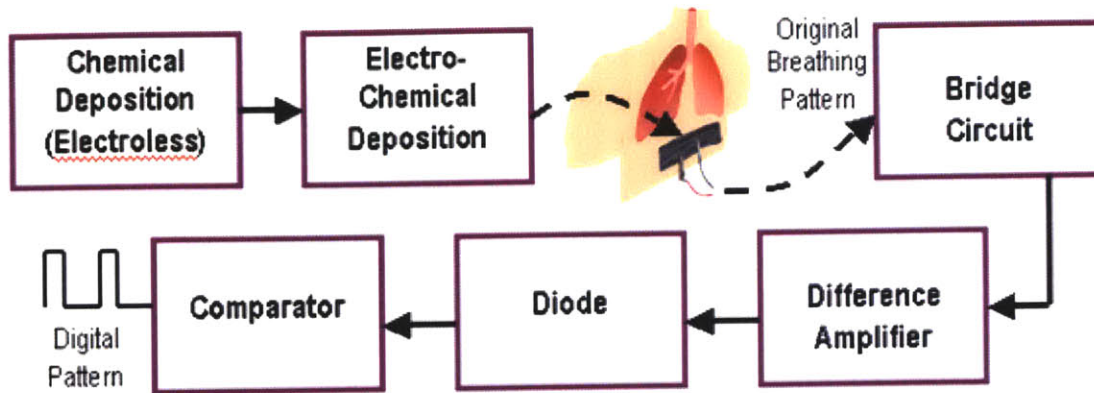
on the probe position, it is likely that there will be some probe placement that will yield an initial resistance somewhere between 0-100 K $\Omega$ .

As the strain gage stretches, the resistance generally decreases, thereby causing the voltage drop across the strain gage to decrease (assuming the current is relatively constant). This voltage drop across the strain gage then causes the output voltage of the bridge ( $V_o$  in *Figure 15*) to increase. Because the interest was in capturing only the voltage  $V_o$ , the INA145UA precision, unity-gain difference amplifier was chosen. The INA145UA difference amplifier has a high common-mode rejection ratio of 86 dB [36], thereby amplifying only the difference between the middle terminals of the bridge circuit ( $V_o$ ) as desired, instead of amplifying the full voltage at each of the middle terminals. The gain of the INA145UA is easily set by choosing the ratio between two resistors [36], and the final gain values were set to 20. Therefore, the output of the difference amplifier is an amplified version of  $V_o$ . *Figure 16* shows the amplified  $V_o$  displayed against the digital heart pulse signal that was recorded at the same time.



**Figure 16: Amplified respiration voltage pattern  $V_o$  displayed against the concurrent digital heart pulse signal.**

Because the signals received from the PPy Lycra strain gage were relatively noise-free, it was not necessary to implement any type of low-pass or band-pass filter. Instead, the output of the difference amplifier could go straight into comparator circuitry similar to that used in the PSS heart rate sensor. A diode was again used to eliminate the negative part of the signal, and a comparator with a reference voltage 1.75 V and hysteresis value 0.75 V was constructed. The resulting output signal was again a digital signal, this time indicating the respiration rate. A block diagram of the respiration rate sensing process is shown in *Figure 17*.



**Figure 17: Block diagram of respiration rate sensor process using PPy Lycra strain gages to produce a digital output signal representing the respiration pattern.**

## Software

The software required for the PSS is written in assembly language for the AT90LS8535 microcontroller and in Microsoft Visual Basic .NET for the GUI run on the notebook computer. The assembly program is written using Atmel's AVR Studio 4.0 and is stored in the microcontroller's flash memory. The program can be run by the notebook computer operator through the GUI. The software for the GUI is stored on the notebook computer's hard drive and can be accessed by the operator. Both programs can be modified easily to accommodate changes in program features such as new commands. Since the GUI designed for the WTCU was already set up to run the assembly program on a command basis, the assembly program was expanded to include commands to query the PSS sensors. To minimize the average power consumption, the microcontroller was instructed to enter idle mode whenever possible.

The goal of the assembly program for the PSS sensors is to find and return the average time between the heart or respiration pulses. The program has been set up to calculate the average time between heart pulses (or respiration pulses) using the method

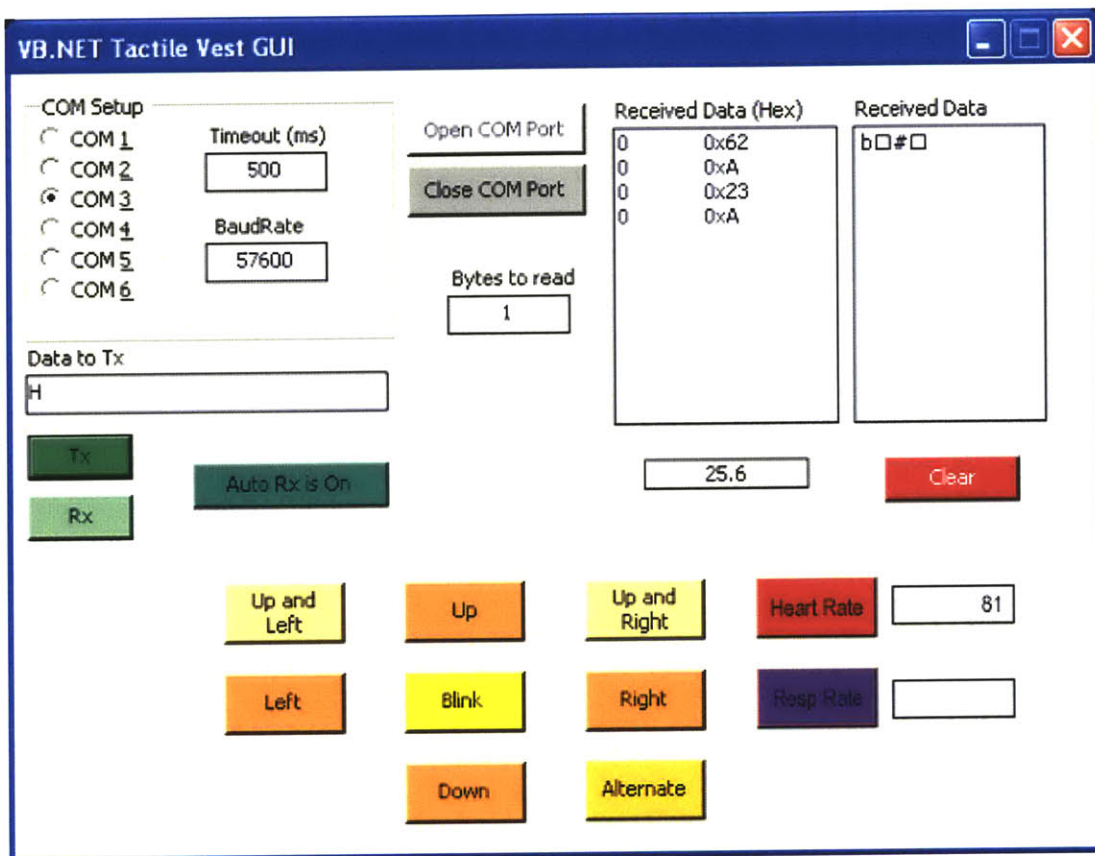


described in the *Heart Rate Sensor – System Results* section. The commands for querying the heart rate and respiration rate sensors have been assigned “H” and “S” respectively. When the notebook computer operator types “H” in the GUI prompt, the Bluetooth module receives this command and wakes up the microcontroller. All registers used in the heart rate sensor routine are cleared, as well as timer 1. The microcontroller receives the digital heart pulse signal at its INT0 pin, an interrupt pin, and is set to trigger on the rising edge of the 0 V to +5 V pulse. While it waits for the first trigger, it goes into sleep (idle) mode.

At the first rising edge of the digital heart pulse, interrupt INT0 is triggered, and the microcontroller performs the interrupt routine. For this first trigger, all the interrupt routine instructs the microcontroller to do is to turn on timer 1 and increment the counter that keeps track of the number of pulses counted. The microcontroller returns to the main heart rate sensor routine and goes back to sleep. At the next rising edge, the interrupt routine is called again. For this pulse and three more pulses, the interrupt routine is responsible for summing the number of timer periods between the rising edges of consecutive pulses. To avoid counting any false pulses produced by aberrations in the signal less than a quarter of a second after a previous pulse, a delay is used to disable triggering.

Each time (after the first) that the interrupt routine is called, it adds the current value of timer 1 to its value at the previous interrupt, clears the timer, waits approximately 0.21 seconds, and increments a pulse counter. When the interrupt routine is complete, the program returns to the main heart rate sensor routine to determine if it has finished querying the heart rate sensor by checking the value of the pulse counter. If

the counter value is less than 5, the program returns to sleep and waits for the next interrupt trigger. When the counter equals 5, the program divides the total number of timer periods by 4 (the number of pulses used), and sends the resulting average number of timer periods per pulse to the Bluetooth module. This average is then sent to the GUI in hex form and can be seen by the operator in the Received Data box. A version of the GUI is shown in *Figure 18*.



**Figure 18:** GUI showing an example of how to query the PSS to measure the user’s heart rate and showing the results of the query. The “H” in the *Data to Tx* box sends the query, and the first two hex numbers (low byte followed by high byte) in the *Received Data* box are the result of the digital heart pulse signal analysis.

The first number in the Received Data box is the low byte of the average, and the second number is the high byte. Therefore, in *Figure 18*, the average number of timer

periods per pulse is 0A62 hex. This corresponds to 2658 in decimal. To compute the number of seconds per heartbeat, this value is multiplied by the number of timer cycles per period (1024 for timer 1) and divided by the clock rate ( $3.6864 \times 10^6$  cycles per second). The heart rate (in bpm) can be found by inverting the resulting value and multiplying by 60. Therefore, the query resulting in the received data shown in *Figure 18* found the PSS user's heart rate to be 81 bpm. The respiration rate sensor assembly program works in exactly the same manner, except that the operator sends the letter "S" as the query. The delay in the interrupt program is increased to 0.39 seconds, since respiration rates generally do not exceed 2.5 Hz.

# Layout and Fabrication

## Compatible WTCU Layout

In order to establish compatibility with the WTCU, the PSS was designed to use the open ports of the Atmel AT90LS8535 Microcontroller on the WTCU. The appropriate INT0 and INT1 interrupt pins on the microcontroller were linked to a connector that would allow for information to be sent to and received from the PSS. Additionally, the number and placement of the +9 V, +5 V, and -5 V power supplies had to be decided upon so that if any power was to be supplied by the WTCU to the PSS, new traces could be added on the WTCU to connect these power supplies to the PSS connector.

Option	PSS Board	WTCU Board	Advantages	Disadvantages
A	+5 V, -5 V	+9 V	Sources close to op amps on PSS, same +9V source as on WTCU	Extra cost of the +5 V supply
B	-5 V	+9 V, +5 V	Only new cost is -5V supply	+5 V source will be used for both WTCU motor controllers and PSS circuitry, and will be far away from PSS
C	+9 V, +5 V, -5 V		Completely self-powered board	Extra cost of +9 V and +5 V sources, and extra weight and bulk of +9 V battery

**Table 3: +9 V, +5 V, and -5 V supply placement options and their advantages and disadvantages.**

The three main options considered for supply placement are shown in *Table 3*. Option C was ruled out due to the unnecessary additional +9 V battery which would add weight and bulk to the final unit. Option B was initially the option of choice due to cost and function. However, it was then ruled out because of the issue of having so many parts in both the WTCU and PSS drawing power from the same source and possibly requiring

a significant amount of current. More importantly, it was thought that transporting the +5 V power supply from the WTCU to PSS via a connection wire would introduce variation and noise into the PSS supply. Since the op-amps and sensors directly use the +5 V supply, any noise in the supply would affect the functioning and output signals of these parts, which would be undesirable. In contrast, transporting the +9 V power supply from the WTCU to PSS would be tolerable because the +9 V supply would only be a source to the voltage regulators that are capable of working for a range of input voltages. Therefore, Option A was chosen to be implemented in the new WTCU and final PSS design because the tradeoff between the additional +5 V voltage regulator cost and having a clean +5 V power supply was acceptable.

## Printed Circuit Board Components

After the PSS circuit design was completed and tested on the breadboard, the next step toward creating a functional printed circuit board was entering the circuit design into OrCAD's Schematic Capture CIS program. The schematic closely followed the breadboard design; however, the addition and modification of some parts was necessary in order to fabricate the printed circuit board. Basic changes included adding connectors *J1* and *J2* for the sensor attachments and connector *J3* for communication with the WTCU and the +9 V power supply. Power supplies *U5* (-5 V) and *U6* (+5 V), were added to replace the power generator used during the breadboard design process. SMD parts corresponding to each DIP part that had been used on the breadboard were also determined. For each of the op-amps used in the breadboard design, SMD versions were available.

Two models that were considered for the +5 V and -5 V voltage regulators that would replace the power generator were the PT78NR105V and LM2595S. The LM2595S is used on the WTCU to supply +5 V to the motor controllers; however, it requires a number of large external parts such as inductors, capacitors, and diodes. Although the PT78NR105V comes in both +5 V and -5 V units, only requires a few external parts, and is slightly smaller than the LM2595S with its external parts, the deciding factor in selecting the regulators was the minimum output current. The LM2595S does not have a minimum output current, whereas the PT78NR105V minimum output current is 100 mA for the +5 V source and 50 mA for the -5 V source [37]. Since the final PSS design only draws approximately 45 mA from the +5 V source and 13 mA from the -5 V source, the PT78NR105V would be overloading the PSS. Therefore, it was eliminated as an option. The LM2595S +5 V regulator could be turned into a -5 V regulator with a few more external components, and so it was chosen for both the +5 V and -5 V power supplies on the PSS board [38].

Size is an important feature of the PSS board, and the smallest components available, with appropriate power and current ratings and at a reasonable price, were chosen when possible. Due to the circuit being essentially low-current, resistors with power ratings of 1/20 W were acceptable in most cases; therefore, size 0201 resistors (length 0.6 mm) [39], the smallest available, were used in all situations except for  $R_I$ . The current drawn by the IR emitter LED can be up to 25 mA, which combined with the low resistance of  $R_I$ , causes the power rating necessary for  $R_I$  to be larger than 1/10 W. Due to this power restriction,  $R_I$  was chosen to be a size 0805 resistor (length 2.0 mm) capable of handling up to 1/8 W [40].

Ceramic capacitors were preferred over tantalum capacitors due to their superior impedance and low equivalent series resistance (ESR) [41]. Additionally, since the initial voltages of the signals from both the heart rate and respiration rate sensors fluctuate between positive and negative, the non-polar nature of ceramic capacitors was crucial in dealing directly with the raw physiological signals. The only tantalum capacitors used in the circuit were for the LM2595S regulators, for which the values of tantalum capacitors had been specified [38].

Diodes were chosen to have a low forward voltage drop, appropriate voltage and current rating, and to be as small as possible. The three sizes, SMA, SMB, and SMC, have areas of 13.4 mm, 18.0 mm, and 44.2 mm respectively. In the WTCU, the diode used in the +5 V regulator circuit was the SMC size MBRS340T3OSCT Schottky diode, a replacement for the suggested IN5822. Minimizing the size of the PSS board was important; therefore, for the PSS, the SMA size B340LADICT Schottky diode was used as it has the same 40 V voltage rating and 3 A current rating [42]. Similarly, instead of using the SMC size MBRS320T3OSCT to replace the IN5820, the B340LADICT was used again since there was no 20 V equivalent. The IN5400, suggested in the LM2595S datasheet, was replaced by an equivalently-rated 50 V and 3 A SMC size S3ADICT diode, its large size due to its high voltage rating [43]. The diodes used before the comparators in both the sensor circuits did not need to have as high a current rating as those in the regulator circuits since the total current being drawn by the sensor circuits is ~60 mA. Therefore, the 30 V and 1 A rated SMA size B130DICT was selected [44].

The issue of voltage regulator precision was also examined to see if it would be necessary or even possible to use a +5 V voltage reference as the supply to the heart rate

and respiration rate sensors. A standard high-quality voltage reference is the LT1019ACS8-5, a +5 V voltage reference with a 0.02% error tolerance (equivalent to an error tolerance of 1 mV) [45]. Although the LT1019ACS8-5 would provide an extremely stable +5 V supply to the heart-rate sensor and respiration-rate sensor, the major problem with this and many other voltage references is their output current. The maximum output current for the LT1019ACS8-5 is 10 mA [45]; however, the IR emitter LED alone requires approximately 25 mA [30]. The LM2595S, though not as stable a +5 V source as the voltage reference, can provide the necessary current and is still sufficiently stable to be a direct voltage source to the sensors. Temperature changes can cause an error of 0.15% in the voltage regulator [38], which translates into a maximum current error of 0.2% or 0.05 mA. Considering that the variation in the light-reflecting properties of the skin within a single person's wrist area can produce considerable variability in the heart rate signals, this voltage regulator error should not have a negative effect on the functionality of the circuit.

A complete list of all the parts used for the PSS printed circuit board can be found in *Appendix B, Table B-1*.

## **PSS Schematic**

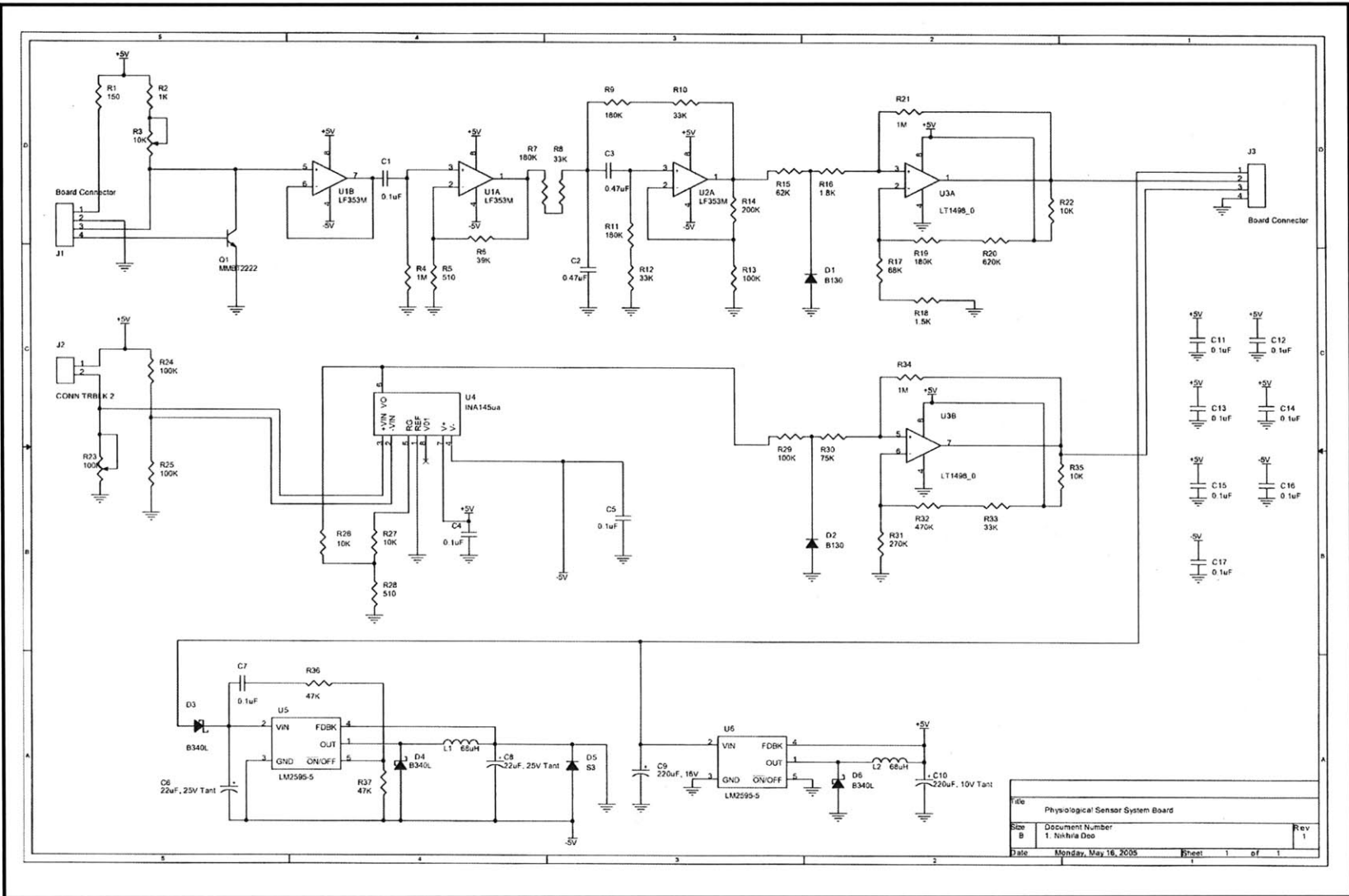
The development of the PSS schematic involved representing each part of the PSS circuit by a symbol containing the correct number of pins and a reference name in OrCAD's Schematic Capture CIS program. Standard symbols were used for resistors, capacitors, diodes, and transistors, and op-amps, while custom parts such as the connectors were generally represented by rectangles with the appropriately numbered and



defined pins. Once the pins on each part were connected as dictated by the PSS circuit, each part was given a reference name for identification, such as *R1* or *U1*.

To link the schematic to the printed circuit board layout, each part in the schematic had to be assigned a specific footprint that had been created in OrCAD's Layout Plus program. The footprint file came from a custom or standard library and conveyed the size and shape of the actual part as well as which areas would be etched for soldering. In order to create a custom footprint for a part, the dimensions of the part's case, pin-size, and pin layout were required and could be determined from the part's datasheet. To improve the chances that the part would fit well in the allotted space and that the pins would make a solid connection with the board, it was necessary to enlarge the solder pads in the footprint to allow for some extra solder area beyond the largest pin length.

After each part was chosen and linked to both a symbol and a footprint, the symbols were arranged in the Schematic Capture program such that connections between appropriate symbols' pins could be made by adding wire symbols to the diagram. A wire symbol drawn from one pin to another would indicate to the program that these pins are to be connected in the circuit. *Figure 19* shows the complete PSS schematic created in the OrCAD Schematic Capture CIS program. After the schematic was completed, a netlist file was created to transfer all of the part, footprint, and connectivity information to the OrCAD Layout Plus program.

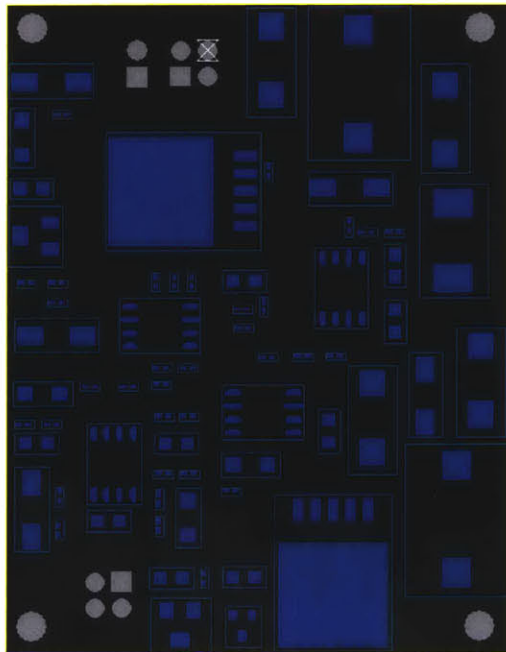


File	Physiological Sensor System Board	
Size	Document Number	Rev
B	1, Vishva Doo	1
Date	Monday, May 16, 2005	Sheet 1 of 1

Figure 19: Overall schematic of the PSS.

## Printed Circuit Board Layout

The entire layout of the PSS printed circuit board (PCB) was done in OrCAD's Layout Plus program. The process began by importing the netlist file created in OrCAD's Schematic Capture CIS program. The Layout program then generated a "ratsnest" or large disorderly cluster containing the footprints of all of the parts in the schematic and connecting the appropriate pins together with yellow lines. Once the PCB dimensions were estimated and a tentative board outline was established, the manipulation of each part in the "ratsnest" could begin. The objective was to fit all the parts on the board into the smallest configuration possible, while still allowing room for traces to be added. The final board size was 48.5 mm x 62 mm. *Figure 20* shows the untangled and final layout of the footprints. This organization of the "ratsnest" and the subsequent trace routing comprised the bulk of the PCB development process.



**Figure 20: PSS footprint layout. (The yellow lines connecting each part directly in an organized ratsnest have been hidden for clarity of the layout.)**

After all of the parts' footprints were untangled and placed appropriately within the board outline, the yellow lines representing the connectivity information were converted into copper traces that serve as the actual paths on which electrical information can travel from part to part. Trace routing can be done manually or with the automatic router in the software. Due to the desire to minimize the size of the PSS, the parts were placed together in a logical but space-limited configuration. Although these space restrictions could cause the auto-router to fail to find a routing solution, auto-routing the entire PSS board was actually possible. However, the result contained unnecessarily long traces as well as less than ideal connection patterns and usage of space. Since the auto-router did not maximize efficiency and noise control, it was necessary to use the manual routing option.

Although manually routing the board takes a significantly longer time than auto-routing, the result is a space-efficient routing configuration that takes into account the designer's circuit specifications and any individual part requirements, thus requiring less debugging in the end. Overall noise control, space usage, and current flow were the three main factors considered when routing the traces manually. Great effort was taken to keep traces as short as possible in order to minimize the circuit's exposure to external noise. A ground plane was also added to the PSS board to reduce noise across the entire board and to allow for shorter, more space-efficient routing of the many parts that needed to be connected to ground. Additionally, the ground plane would improve the performance of the LM2595S voltage regulators since they provide a more stable voltage source when placed over or near a ground plane [38]. Efficient current-flow was handled by varying the trace widths so that wider traces were used for the higher current levels associated

with the power sources. For most of the connection lines on the PSS board, 0.254 mm (10 mil, or 0.010 inch) traces were sufficient. The +5 V and -5 V connection lines were routed with 0.508 mm (20 mil) traces, and the +9 V line was routed with a 1.016 mm (40 mil) trace.

The PSS was designed to use only the top and bottom layers in order to simplify the board construction and reduce costs; however, it is possible to route traces on several different intermediate layers of the board. All of the PSS components are surface-mount (except for the connectors which are through-hole), so the components were all mounted on the top layer. The ground plane covers most of the bottom layer. Traces were routed mostly on the top layer, switching to the bottom layer by *vias* when necessary. Vias are metal-plated holes drilled through the board that electrically connect a trace on one layer of the board to a trace on another. It is often necessary to use a via when two traces on the same layer must cross paths in order to avoid overlapping traces that should not be electrically connected. The trace on the top layer terminates in a via, and the trace on the bottom layer begins at the same via and continues until it returns to the top layer if necessary with another via.

The final OrCAD layout design of the PSS is displayed in *Figure 21*. The trace patterns for the top and bottom layers are shown in *Figures 22* and *23*, respectively.

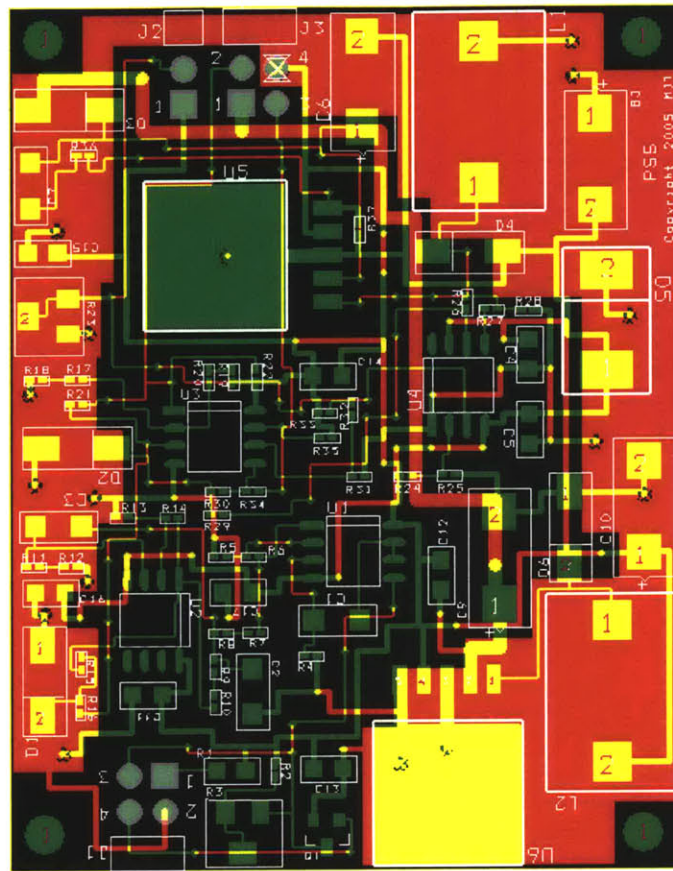


Figure 21: Final PCB Layout (from OrCAD Layout Plus) showing all layers of the PSS board. (Top is green, bottom is red, overlapping top and bottom areas are yellow, and silk screen is white.)

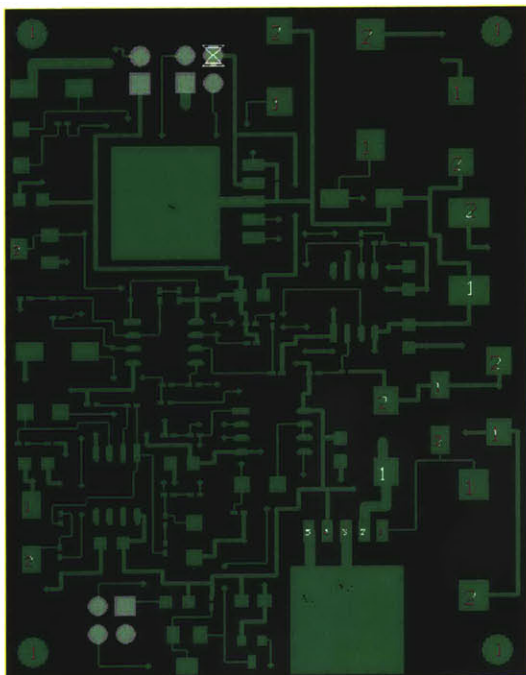


Figure 22: Top layer PCB layout.

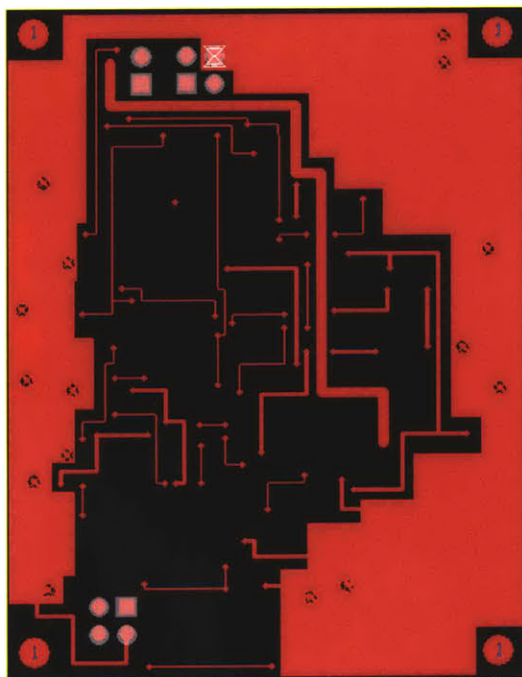
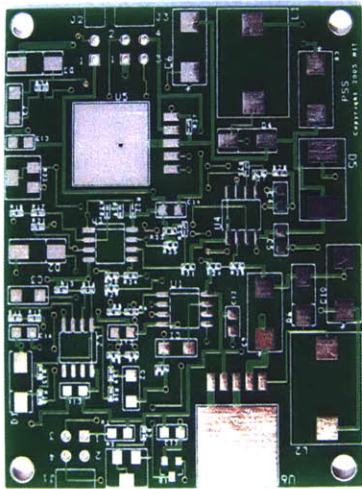


Figure 23: Bottom layer PCB layout.

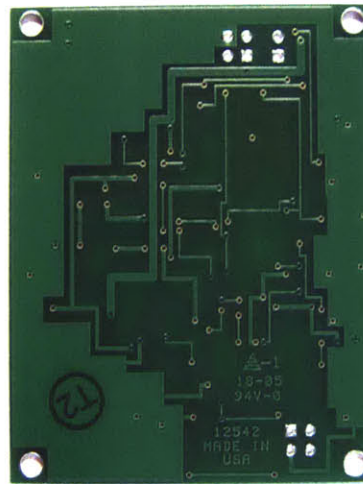
# Final PSS Product Design

## PCB Population and Testing

The final layout design of the PSS board was sent to Power Design Services (San Jose, CA) for fabrication and population. They accepted the files generated by OrCAD Layout Plus and then etched the appropriate patterns into a Standard FR-4 copper board. The board was subsequently coated with green insulation everywhere except those areas where parts would be soldered (see *Figure 20*). The text was printed using a silkscreen process. *Figure 24* and *Figure 25* show the top and bottom of the unpopulated PCB, respectively. *Figure 26* shows the top view of the final populated PSS board.

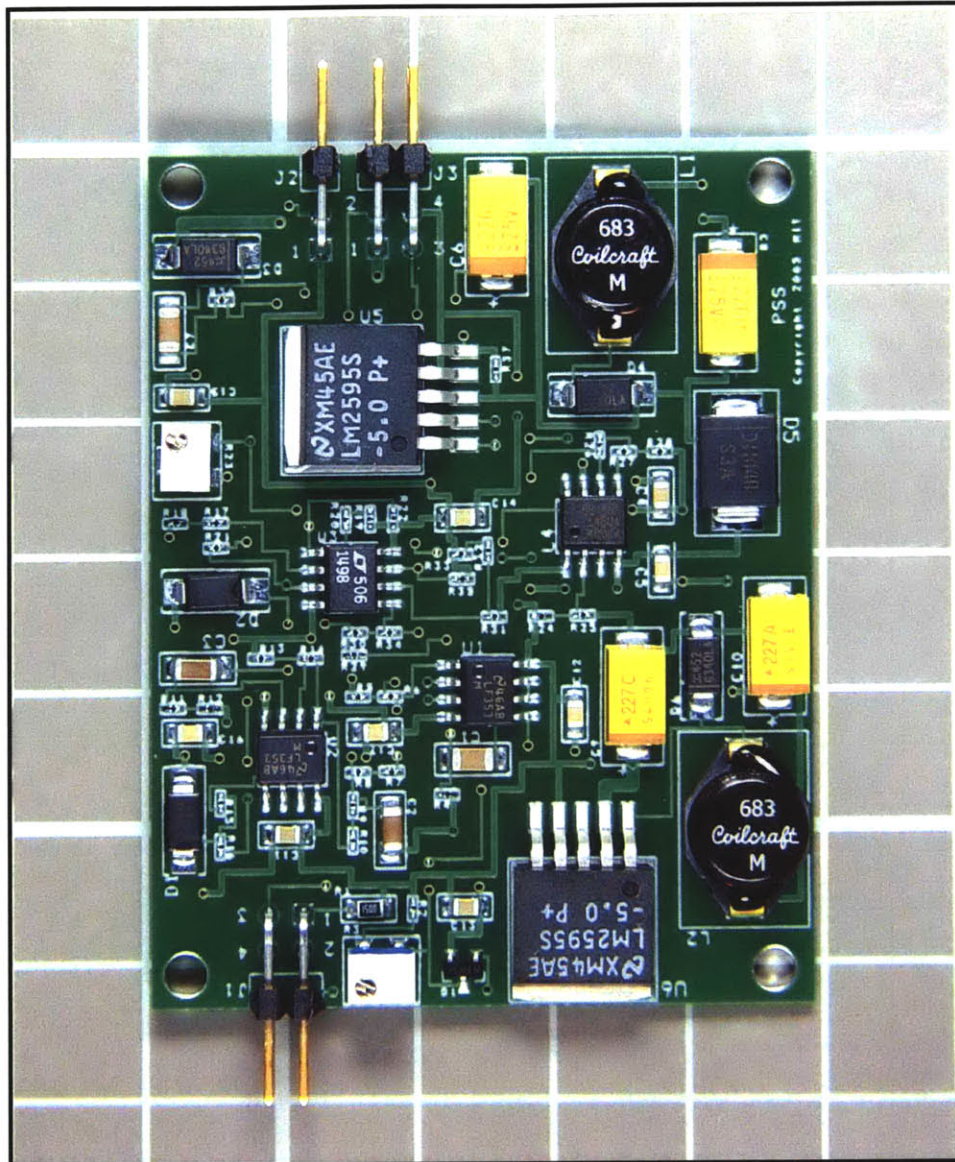


**Figure 24: Top layer of unpopulated PCB.**  
(Image approximates actual board size of 48.5 mm x 62.0 mm)



**Figure 25: Bottom layer unpopulated PCB.**  
(Image approximates actual board size of 48.5 mm x 62.0 mm)





**Figure 26: The final PSS board and physiological sensors circuit. One square is ~ 10 x 10 mm.**

The WTCU is set up to send power to the PSS board only when a heart rate or respiration rate query has been made. The power source currently used by the WTCU is a +9 V battery; however, other lower power sources have been tested and may be used. *Table 4* shows the changes in current and power for +7 V to +9 V power sources.



<b>WTCU Source</b>	<b>Total Current</b>	<b>Total Power</b>
7 V	77 mA	0.593 W
8 V	65 mA	0.520 W
9 V	58 mA	0.522 W

**Table 4: Total current and power used by the PSS in relation to the WTCU source.**

## **Wearable Sensors**

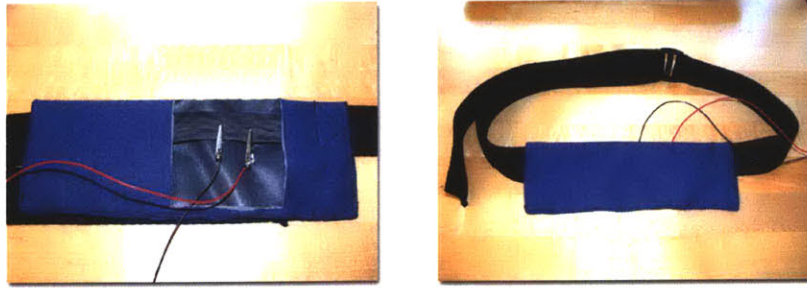
The sensors were encased to that they could be worn. Since the heart rate is measured on the left wrist, a wrist strap was the most logical heart rate sensor device. The IR sensor unit needs to be held firmly against the left wrist, and the 1043/1070 IR emitter and phototransistor must be isolated from each other so that the only IR light detected by the phototransistor is that reflected by the skin. Therefore, the material chosen for the wrist strap is terry-backed 1/16 inch Neoprene (Rubatex) because it can be stretched, provides some protection for the sensors, keeps the sensors from pressing directly onto the wrist, and occludes light.

The IR emitter and detector were glued between two layers of the neoprene and sewn into place to further isolate one from the other. Wires attached to the IR sensor transmit the wrist pulse signal to the PSS board via connector *J1*. The strap is held tightly to the wrist by Velcro. To get the most reliable signal, the IR sensor should be placed securely on the lower inside of the left wrist, with the detector positioned near a visible vein. *Figure 27* shows the wrist strap design from different angles.



**Figure 27: PSS wrist strap design. (Left) Sensor encasement. (Middle) Sensor-side of strap positioned on lower inside wrist over veins. (Right) Velcro and buckle side of wrist strap.**

The respiration rate is measured as the chest expands and contracts; therefore, the ideal respiration rate sensor device was a chest strap. A 20 mm x 100 mm PPy Lycra strain gage was sewn to a soft but non-flexible strap material. Due to the chemical nature of the PPy Lycra strain gage, it needs to be isolated from repeated contact with the skin. Additionally, the properties of the PPy might change over time due to solvent loss or oxidation by the environment [19]; therefore, isolation of the PPy Lycra strain gage from the environment may help extend the life of the strain gage. A grey XALT 3-layer waterproof breathable material (Burlington Fabrics) was used to surround the PPy Lycra strain gage. To further isolate the body from the PPy Lycra strain gage, and to help pull away moisture from the body and the waterproof material, a blue wick-away 100% polyester knit material (Milliken 694346) was sewn over the waterproof material to provide a comfortable interface between the waterproof strain gage covering and the skin. *Figure 28* shows the chest strap design. Velcro was again used to fasten the chest strap.



**Figure 28: PSS chest strap design. (Left) Wick-away and waterproof pouch is cut open to reveal the PPy Lycra strain gage. (Right) Overall view of the chest strap.**

4.8 mm steel alligator clips (Digikey 314-1010) were used to clip onto the PPy Lycra strain gage. (If necessary, any non-conductive flexible smooth material can be inserted between the alligator clips and the waterproof material to reduce the friction of the clips against the waterproof material when the strain gage is being stretched and released.) Wires attached to the clips connect to the PSS at connected  $J2$ . To calibrate the respiration rate sensor, the chest strap is tightened around the chest so that the strain gage is partially stretched when the chest is not expanded. The distance between the alligator clips should be adjusted until the resistance reading is less than  $100\text{ K}\Omega$ . The closer the resistance is to  $100\text{ K}\Omega$  while staying below it, the better chance that the probes are far enough apart to not touch, the resistance reading is above the strain gage's fully-stretched resistance, and the resistance has a large enough range to decrease as it is stretched. The potentiometer  $R23$  on the PSS board should then be adjusted to match the strain gage resistance by turning the dial on the top of the potentiometer. After the calibration is complete, the bridge circuit is balanced when the chest is not expanded, and it should respond appropriately when the chest expands and stretches the PPy Lycra strain gage.

The PSS board is secured to the upper left arm or in a pocket on the back of the left shoulder to provide ideal placement of the connectors to the wrist strap, chest strap, and WTCU. A small box to encase the PSS will be necessary to protect the board from the environment.

## Further Research

There are many possibilities for further development of the PSS, including modifications of the hardware, software, wearable sensors, or applications. The largest heart rate sensor issue that could not be fully explored during this research project was how to provide accurate physiological measurements while the person using the PSS is not stationary. Movement of the IR sensor creates erroneous high frequency peaks in the heart pulse signal, similar to the effects of electrode movement in ECG monitoring [46]. Although band-pass filtering has been implemented to eliminate high frequency noise and help reduce the effects of minor movements, it is not sufficient to compensate for significant movement [8].

Digital signal processing of the heart rate signal may provide some avenues for improvement in this area [8]. Algorithms that take into account how quickly a person's heart rate can increase or decrease by a certain amount may also be useful. Since the peaks due to movement are very large and very narrow, whereas the peaks due to the heart rate are not as large and are wider, there are two more specific methods of digital signal processing that might help avoid counting these peaks as heart beats. One method would be to set a high threshold voltage at some point in the circuit before the comparator, and any peaks that go above that voltage would not be counted as heart beats. Another method would be to evaluate the digital signal at the output of the comparator and measure the amount of time each peak stays high (+5 V). Since the movement-related peaks remain high for a much shorter time than the heart pulse peaks, the interrupt could be set not to trigger unless the voltage has remained high for a sufficient amount of time.

IR technology has uses beyond measuring the heart rate. Further research could be done into expanding the use of the IR sensor to measure not only heart rate but also pulse oximetry [5]. Additionally, if wire mesh electrodes can be easily obtained or created, it might be interesting to explore using both IR technology and ECG signals to obtain more detailed information about the PSS user's cardiac activity. The analog signal processing for both technologies will be similar since the problems faced by each technology are similar. The addition of digital signal processing would allow for detection of the heart rate, pulse oximetry, cardiac arrhythmias, and muscle damage. Because a flexible chest-strap has already been designed for the respiration rate sensor, fabric electrodes could be incorporated into the stationary part of the chest strap to combine the ECG and respiration sensor into one device.

The PPy Lycra strain gage technology also provides many opportunities for further improvement. The properties of the PPy Lycra material are not yet fully understood in relation to bio-monitoring (see Skotheim et al. [47] and Osada et al. [48] for overviews of the properties). Lifetime and consistency of these strain gages were two limitations of this technology noticed while testing the respiration rate sensor. Development of a coating to reduce solvent loss and oxidation as well as more precise manufacturing of the PPy Lycra strain gages could solve these problems [16, 19]. Thicker electrochemical PPy coatings of higher quality may also improve the functionality of these strain gages [19].

With or without more information on the PPy Lycra strain gage properties, digital calibration of the respiration rate sensor would be useful in reducing the set-up time. By developing a unit to analyze the specific PPy Lycra strain gage being used for a particular

sensor, the bridge could be balanced by relaying the resistance information to a digital potentiometer. Also, making use of the microcontroller pins that can function as a digital comparator might allow for adjustable reference voltage and hysteresis levels.

The PSS and WTCU together form a foundation for a wireless system that can be used to send tactile commands to person as well as to monitor their physiological status. Although the system is being created for use in the military, it can easily be adapted for use in hospitals or home-monitoring systems as well. The low power, low cost, and light-weight nature of this system provide the groundwork necessary for expanding the physiological monitoring system. The open A/D ports on the AT90LS8535 microcontroller provide room for additional physiological sensors to be added to this system even without the addition of other micro-processing units. The easily changeable software provides the tools necessary to explore further processing of the physiological signals.

The most logical next step in the development of the PSS would be to explore the benefits of adding a digital signal-processing unit. This unit could greatly increase the current functionality of the heart rate sensor and would be valuable for use with any other PSS sensors as well. Improvements to the respiration rate sensor could also be made by digital analysis and calibration of the PPy Lycra strain gage properties. Following the development of this unit, other physiological sensors such as temperature, blood pressure, or activity monitors could be added to the PSS to create a more complete physiological monitoring system.

## References

- [1] Noller, HG. In *Medical Electronics*, ed. CN Smyth, pp. 296. London: Iliffe, 1960.
- [2] Neuman MR, Picconnatto J, Roux JF. "A wireless radiotelemetry system for monitoring fetal heart rate and intrauterine pressure during labor and delivery." *Gynecol. Invest.* 1:92-104, 1970.
- [3] Neuman MR, O'Connor E. "A two-channel radiotelemetry system for clinical fetal monitoring." *Biotelemed. Patient Monit.* 7:104-21, 1980.
- [4] Hess OW, Litvenko W. "Instrumentation of a telemetry system for monitoring fetal and neonatal cardiac activity." *Conn. Med.* 43:14-16, 1979.
- [5] Budinger, TF. "Biomonitoring with Wireless Communications." *Annual Review of Biomedical Engineering* Vol. 5: 383-412, August 2003.
- [6] Ko WH, Yon ET, Mabrouk S, Hynceck J. "Taped on heart rate and electrocardiogram telemetry transmitters." *J. Assoc. Adv. Med. Instrum.* 5:268-72, 1971.
- [7] Kulick, DL, M.D. *Electrocardiogram*. Last modified April 9, 2002. Viewed August 29, 2004. Online:  
[http://www.medicinenet.com/Electrocardiogram\\_ECG\\_or\\_EKG/article.htm](http://www.medicinenet.com/Electrocardiogram_ECG_or_EKG/article.htm)
- [8] T. Martin, E. Jovanov, and D Raskovic. "Issues in Wearable Computing for Medical Monitoring Applications: A Case Study of a Wearable ECG Monitoring Device." *IEEE Transactions on Biomedical Engineering*, June 2000, pp. 43-48.
- [9] Moody G.B., Mark R.G., Zoccola A., Mantero S. *Derivation of Respiratory Signals from Multi-Lead ECGs*. Computers in Cardiology 1985, vol. 12, pp. 113-116 (Washington, DC: IEEE Computer Society Press).
- [10] Vernier Software and Technology. Revised May 26, 2000. Viewed August 28, 2004. Online: <http://www.ti-edu.com/pdf/rmb.pdf>
- [11] National Instruments. *Measuring Strain with Strain Gages*. Viewed on August 24, 2004. Online:  
<http://zone.ni.com/devzone/conceptd.nsf/webmain/C83E9B93DE714DB08625686600704DB1?OpenDocument>
- [12] NASA Ames Astrobionics. *LifeGuard – A Wearable Vital Signs Monitoring System*. Viewed August 28, 2004. Online:  
[http://lifeguard.stanford.edu/lifeguard\\_writeup\\_medium.pdf](http://lifeguard.stanford.edu/lifeguard_writeup_medium.pdf)



- [13] Cardiology Online, International Academy of Cardiology. *Vital Sign Device Improves Resuscitation Monitoring*. Viewed August 28, 2004. Online: [http://www.cardiologyonline.com/journal\\_articles/Vital\\_sign.htm](http://www.cardiologyonline.com/journal_articles/Vital_sign.htm)
- [14] Lockyer, B. *A Wireless Communication System for a Tactile Vest*. Masters of Engineering Thesis. Cambridge, MA: Massachusetts Institute of Technology; 2004.
- [15] Jones, L.A. Lockyer, B. and Piatetski, E. (2005) "Tactile Display and Vibrotactile Pattern Recognition on the Torso." Submitted to Advanced Robotics.
- [16] Kuhn, H.H. and Kimbrell, Jr. W.C., inventors. *Electrically conductive textile materials and method for making same*. Milliken, assignee. USA 4803096. 1989 Feb 7.
- [17] Madden, J.D. *Conducting Polymer Actuators*. Ph.D. Thesis. Cambridge, MA: Massachusetts Institute of Technology; 2000.
- [18] Lorussi F., Rocchia W., Scilingo E.P., Tognetti A., De Rossi, D. "Wearable, Redundant Fabric-Based Sensor Arrays for Reconstruction of Body Segment Posture." *IEEE Sensors Journal*, Vol. 4, No. 6, December 2004, pp. 807-818.
- [19] Madden, P.G.A. *Development and Modeling of Conducting Polymer Actuators and the Fabrication of a Conducting Polymer Based Feedback Loop*. Ph.D. Thesis. Cambridge, MA: Massachusetts Institute of Technology; 2003.
- [20] De Rossi, D., Della Santa, A. and Mazzoldi, A. "Dressware: Wearable hardware." *Materials Science and Engineering C*. 1999; 7(1):31-35.
- [21] Atmel, "8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash: AT90S8535, AT90LS8535," Data Sheet Rev. 1041H, November 2001.
- [22] BlueRadios, Inc., "Bluetooth Module BR-C11 Class 1," Data Sheet.
- [23] *Heartbeat Transducer circuit*. Viewed July 29, 2004. Online: <http://circuitos.tripod.cl/schem/r100.gif>.
- [24] National Semiconductor, "LF353 Wide Bandwidth Dual JFET Input Operational Amplifier", Data Sheet No. DS005649, December 2003
- [25] Fairchild Semiconductor, "QEB421 Surface Mount Infrared Light Emitting Diode", Data Sheet No. DS300385, February 2001
- [26] Fairchild Semiconductor, "QSB320F Surface Mount Silicon Infrared Phototransistor", Data Sheet No. DS300387, February 2001

- [27] Fairchild Semiconductor, "QSB320 Surface Mount Silicon Infrared Phototransistor", Data Sheet No. DS300386, February 2001
- [28] Optek Technology, "NPN Silicon Phototransistors Types OP550A, OP550B, OP550C, OP550D", Data Sheet No. OP550A, June 1996
- [29] Optek Technology, "NPN Silicon Photodarlington Types OP560A, OP560B, OP560C, OP560D", Data Sheet No. OP560A, June 1996
- [30] Optek Technology, "GaAs Plastic Infrared Emitting Diodes Types OP140A, OP140B, OP140C, OP140D", Data Sheet No. OPI40A, May 1996
- [31] Hilburn JL, Johnson DE. *Manual of Active Filter Design* Second Edition. USA: McGraw-Hill, 1983.
- [32] Linear Technology, "LT1498/LT1499 10MHz, 6V/us, Dual/Quad Rail-to-Rail Input and Output Precision C-Load Op Amps", Data Sheet No. 14989fa, 1996.
- [33] Spinks, G.M., G. G. Wallace and et al. "Conducting polymers and carbon nanotubes as electromechanical actuators and strain sensors." *Materials Research Society Symposium Proceedings 698 (Electroactive Polymers and Rapid Prototyping)*. 2002; 5-16.
- [34] Spinks, G.M., Wallace, G.G., Liu, L. and Zhou, D. "Conducting Polymers Electromechanical Actuators and Strain Sensors." *Macromolecular Symposia*. 2003 Mar; 192: 161-169.
- [35] De Rossi, D., Carpi, F., Lorussi, F., Mazzoldi, A., Scilingo, E.P., and Tognetti, A. "Electroactive Fabrics for Distributed, Conformable and Interactive Systems." *IEEE Sensor 2002 Conference* (Hyatt Orlando, Kissimmee, Florida. 11-14 June 2002). 5.1-6.
- [36] Burr-Brown, "Programmable Gain Difference Amplifier", Data Sheet No. INA145, April 2005.
- [37] Texas Instruments, "PT78NR100 Series: 1 Amp Plus to Minus Voltage Integrated Switching Regulator", Data Sheet No. SLTS058B, August 2000.
- [38] National Semiconductor, "LM2595 SIMPLE SWITCHER Power Converter 150 kHz 1A Step-Down Voltage Regulator", Data Sheet No. DS012565, May 1999
- [39] Panasonic, "Precision Thick Film Chip Resistors (ERG1J)", Data Sheet No. ER3.
- [40] Phycomp (Yaego), "1% Precision Chip Resistors (0805)", Datasheet. April 9, 2002. Rev. 9.

- [41] Harayda, G.M., Omi, A., Yamamoto, A. "Improve Your Designs with Large Capacitance Value Multi-Layer Ceramic Chip (MLCC) Capacitors". *Panasonic Electronic Components*. Publication # PG33. August 10, 2004.
- [42] Diodes Incorporated, "B340LA/B 3.0A Low VF Schottky Barrier Rectifier", Data Sheet No. DS30240 Rev. B-2.
- [43] Diodes Incorporated, "S3A/B – S3M/B 3.0A Surface Mount Glass Passivated Rectifier", Data Sheet No. DS16005 Rev. 10-2.
- [44] Diodes Incorporated, "B120/B – B160/B 1.0A Surface Mount Schottky Barrier Rectifier", Data Sheet No. DS13002 Rev. 11-2.
- [45] Linear Technology, "LT1019 Precision Reference", Data Sheet No. LT/TP 1229, 1993. Rev. C.
- [46] Dashi, P.K. "Electrocardiogram Monitoring." *Indian Journal of Anaesthesiology*, Vol. 64, No. 6, 2002, pp. 251-260.
- [47] Skotheim, T.A., Elsenbamer, R.L., and Reynolads, J.R. "Handbook of Conducting Polymers." *Editors ed.* New York: Marcel Dekker; 1998.
- [48] Osada, Y., and De Rossi, D.E. "Polymer Sensors and Actuators." Berlin: Springer; 2000.

# Appendix A

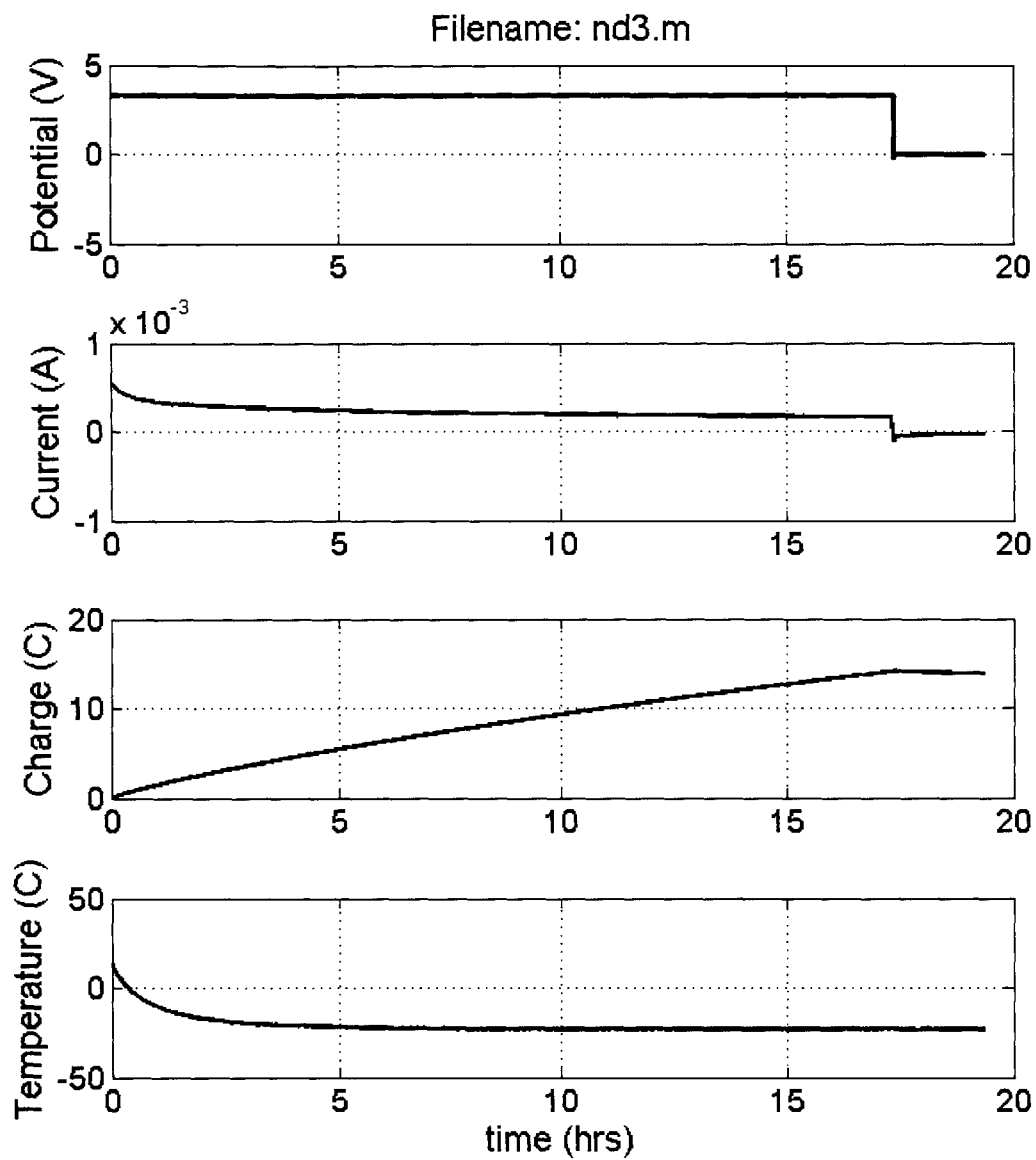


Figure A-1: Electrochemical deposition properties over the course of reaction for Batch Three.

## Appendix B

Description	Part Reference	Part Number
Diode, 30V, 1A, SMA	D1, D2	B130DICT-ND
Diode, Schottky 40V, 3A, SMC	D3, D4, D6	B340LADICT-ND
Diode, 50 V, 3A, SMC	D5	S3ADICT-ND
BJT, NPN, 30V, 600mA, smd	Q1	MMBT2222LT1OSCT-ND
LF353M 8-SOIC Dual JFET	U1, U2	LF353M-ND
LT1498 8-SOIC Dual Precision Amp	U3	LT1498CS8-ND
INA145UA 8-SOIC Diff Amp	U4	INA145UA-ND
pos 5V Regulator	U5, U6	LM2595S-5.0-ND
0.1uF, 50V, 10%, 1206, smd CER (non-pol)	C1, C7	95B9344 (Newark)
0.47uF, 50V, 10%, 1206, smd CER (non-pol)	C2, C3	86H5315 (Newark)
0.1uF, 16V, 10%, 0805, smd CER (non-pol)	C4, C5, C11-C17	311-1142-1-ND
CAP TANT 22UF 25V 10% LOESR SMD	C6, C8	495-1567-1-ND
220uF, 16V, 10%, Tant Smd (pol)	C9	478-1800-1-ND
220uF, 10V, 10%, Tant Smd (pol)	C10	478-1784-1-ND
TRIMPOT 100K OHM 4MM TOP ADJ SMD	R23	3214W-104ECT-ND
TRIMPOT 10K OHM 4MM TOP ADJ SMD	R3	3214W-103ECT-ND
RES 150 OHM 1/20W 1% 0805 SMD	R1	311-150CCT-ND
RES 1K OHM 1/20W 1% 0201 SMD	R2	P1.0KABCT-ND
RES 1.0M OHM 1/20W 5% 0201 SMD	R4, R21, R34	P1.0MAGCT-ND
RES 510 OHM 1/20W 1% 0201 SMD	R5, R28	P510ABCT-ND
RES 39K OHM 1/20W 1% 0201 SMD	R6	P39KABCT-ND
RES 180K OHM 1/20W 1% 0201 SMD	R7, R9, R11, R19	P180KABCT-ND
RES 33K OHM 1/20W 1% 0201 SMD	R8, R10, R12, R33	P33KABCT-ND
RES 100K OHM 1/20W 1% 0201 SMD	R13, R24, R25, R29	P100KABCT-ND
RES 200K OHM 1/20W 1% 0201 SMD	R14	P200KABCT-ND
RES 62K OHM 1/20W 1% 0201 SMD	R15	P62KABCT-ND
RES 1.8K OHM 1/20W 1% 0201 SMD	R16	P1.8KABCT-ND
RES 68K OHM 1/20W 1% 0201 SMD	R17	P68KABCT-ND
RES 1.5K OHM 1/20W 1% 0201 SMD	R18	P1.5KABCT-ND
RES 620K OHM 1/20W 1% 0201 SMD	R20	P620KABCT-ND
RES 10K OHM 1/20W 1% 0201 SMD	R22, R26, R27, R35	P10KABCT-ND
RES 75K OHM 1/20W 1% 0201 SMD	R30	P75KABCT-ND
RES 270K OHM 1/20W 1% 0201 SMD	R31	P270KABCT-ND
RES 470K OHM 1/20W 1% 0201 SMD	R32	P470KABCT-ND
RES 47K OHM 1/20W 1% 0201 SMD	R36, R37	P47KABCT-ND
4 pin Header	J1, J3	WM7080-ND
2 pin Header	J2	WM7080-ND
68uH Inductor 3316	L1, L2	DO3316P-683 (Coilcraft)

**Table B-1: List of PSS printed circuit board parts. All parts are Digikey unless otherwise noted.**

# Appendix C

```
;PSS and WTCU
;AT90LS8535 Microcontroller code
;Version 6 (Updated Bluetooth 1.1.15)
;5/19/05

;3x3 code modified for sensor board and rev. 1 WTCU

; ** indicates code added for use of PSS

.include "8535def.inc"

.org 0x000
start:    rjmp setup

.org 0x001                ; **
                rjmp exzero                ; **

.org 0x002                ; **
                rjmp exone                ; **

.org 0x008
                rjmp tisir                ;timer 1 has overflowed

.org 0x00B
                rjmp rxisr                ;character received

.org 0x100
setup:    ldi r16, 0x9F
                out SPL, r16                ;initialize stack pointer to 0x9F
                clr r16
                out SPH, r16
                out DDRA, r16                ;configure port A for inputs
                out PORTA, r16                ;clear port A
                out ADMUX, r16                ;ADC0 is input (pin A0)
                ser r16
                out DDRB, r16                ;configure Port B as outputs only
                out DDRC, r16                ;configure Port C as outputs only
                out DDRD, r16                ;configure Port D as outputs only
                ldi r16, 0x0F
                out PORTB, r16                ;port B = 0x0F
                clr r16
                out PORTC, r16                ;port C = 0x00
                ldi r16, 0x60
                out PORTD, r16                ;port D = 0b01100000
                ldi r16, 0x03
                out UBRR, r16                ;set UBRR to 57600 baud, 3.6864 MHz
                ; crystal
                ldi r16, 0x98
                out UCR, r16                ;set UCR to enable interrupt on RX
```

```

; complete
ldi r16, 0x85
out ADCSR, r16 ;enable ADC, no interrupts, CLK/32
; freq, single conv mode

ldi r16, 0x04
out TIMSK, r16 ;enable interrupt on timer 1
; overflow

ldi r16, 0x43
out MCUCR, r16 ; ** enable idle sleep mode and
; INT0 on rising edge

ldi r31, 0xFE
ldi r30, 0x97
rcall delay ;delay 100ms (for Bluetooth to
; reset)

ldi r16, 0x40
out PORTD, r16 ;remove Bluetooth reset condition
ldi r31, 0xEA
ldi r30, 0xE7
rcall delay ;delay 1.5 seconds for module to
; start up

rcall btcmd
ldi r31, 0xFE
ldi r30, 0x97
rcall delay ;delay 100ms
rcall btconn ;make connection to USB Bluetooth
; module

rjmp rxchar

```

```

;*****
;BLUETOOTH CONFIGURATION AND CONNECTION ROUTINES

```

```

btconn:  ldi r18, 0x41
rcall txchar ;send A
ldi r18, 0x54
rcall txchar ;send T
ldi r18, 0x44
rcall txchar ;send D
ldi r18, 0x53
rcall txchar ;send S
ldi r18, 0x0D
rcall txchar ;send cr
ldi r18, 0x0A
rcall txchar ;send lf
ldi r16, 0x09
mov r7, r16
rcall rxrespl ;looking for 00 (OK) which is
; 9 chars (18-9) (i.e. 22,00)

rcall respok
ldi r16, 0x80
out SREG, r16 ;enable interrupts (global)
sleep ;waiting for receive char
; (connect= 1)

nop
nop
nop
clr r16
out SREG, r16 ;disable interrupts (global)

```

```

clr r7                ;looking for 19-1=18 chars now
                    ; since 1st was rcvd during sleep

rcall rxrespl
rcall respconnl      ;looks for 8 at end of master's
                    ; 12-digit address, assuming a
                    ; connection was made

ldi r16, 0x80
out SREG, r16        ;enable interrupts (global)
ldi r16, 0x02
out PORTB, r16      ;set port pin PB1 high if
                    ; successful connection made
ret                 ;the module should be connected by
                    ; the time this is executed

btcmd:
ldi r18, 0x41
rcall txchar         ;send A
ldi r18, 0x54
rcall txchar         ;send T
ldi r18, 0x4D
rcall txchar         ;send M
ldi r18, 0x43
rcall txchar         ;send C
ldi r18, 0x0D
rcall txchar         ;send cr
ldi r18, 0x0A
rcall txchar         ;send lf
ldi r16, 0x09
mov r7, r16
rcall rxrespl        ;look for 9 chars (18-9)
                    ; (i.e. 32,00)
rcall respok         ;is response 00=OK ?
ldi r16, 0x01
out PORTB, r16      ;set port pin PB0 high if module
                    ; communication successful
ret

;*****
;BLUETOOTH MODULE RESPONSE RECEPTION ROUTINES

rxrespl:  in r19, USR                ;MUST BE CALLED WITH THE PROPER
                    ; VALUE LOADED INTO R7!

sbrs r19, 7
rjmp rxrespl        ;loop back if char not received yet
rcall rxchekl       ;read char to r17 after reception
                    ; (if no data errors)
inc r7              ;r7 can be used as an
                    ; indicator of # chars it's looking
                    ; for (call w/ r7=x, 18-x=#chars)

mov r16, r7
cpi r16, 0x12       ;looking for 18 chars
breq rxendl
cpi r17, 0x0D       ;is char cr?
breq rxrespl
cpi r17, 0x0A       ;is char lf?
breq rxrespl
mov r0, r17         ;***store response in r0 (will not
                    ; store CR or LF chars)***

```



```

rxendl:      rjmp rxrespl
            clr r7
            ret                                ;r17 contains 0x0A here (usually,
                                            ; if not always)

rxchekl:in  r20, USR
            sbrs r20, 4                        ;check USR's FE bit for framing
                                            ; error

            rjmp chekorl
            rjmp ferrl
chekorl:    sbrs r20, 3                        ;check USR's OR bit for overrun
                                            ; error

            rjmp rxokl
            rjmp orerrl
ferrl:     rjmp end
orerrl:    rjmp end
rxokl:     in r17, UDR                        ;received char --> r17
            ret

```

```

;*****
;BLUETOOTH MODULE RESPONSE HANDLERS
;The response is expected in r0 for all routines.

```

```

respok:     ldi r16, 0x30
            cpse r16, r0
            rjmp end
            ret                                ;returns if response = OK
                                            ; = 0x30 = 0 ascii

respokl:    ldi r16, 0x4B
            cpse r16, r0                        ;does r0 contain K=4B? (from
                                            ; response cr,lf,O,K,cr,lf
                                            ; = 6 chars)
            rjmp end                            ;skip this if response = OK (long)
            ret

respconn:   ldi r16, 0x31
            cpse r16, r0
            rjmp end
            ret                                ;returns if response = CONNECT
                                            ; = 0x31 = 1 ascii

respconnl:  ldi r16, 0x38
            cpse r16, r0                        ;looking for last digit (8) in
                                            ; 12-digit master's address
                                            ; (**NOTE: Will only recognize
                                            ; certain masters)
            rjmp end
            ret

```

```

;*****
;CHARACTER TRANSMISSION ROUTINES

```

```

txchar:     rcall txloop
            out UDR, r18                        ;***transmits char in r18***
            rcall txloop
            ret

```

```

txloop:    in r20, USR
           sbrs r20, 5                ;check UDRE to see if ready for
                                           ; new char
           rjmp txloop
           ret                        ;return when ready to transmit

;*****
;COMMAND RECEPTION/READY STATUS ROUTINES

rxchar:    sei
           rcall ready                ;transmit ready (# LF)
           sbi UCR, 7                 ; ** set bit 7 high to enable RX
                                           ; ** complete interrupt

           sleep
           nop                        ;nops in case more instructions
                                           ; are executed before processing
                                           ; interrupt

           nop
           nop
           cli
           cbi UCR, 7                 ; ** clear bit 7 in UCR to disable
                                           ; ** RX complete interrupt

           rcall motor                ;received char in r17
           rjmp rxchar

ready:     ldi r18, 0x23               ;send #
           rcall txchar
           ldi r18, 0x0A               ;send LF (new line)
           rcall txchar
           ret

;*****
;MOTOR CONTROL ROUTINES
;Want to sleep during timing and wake up on timer overflow if possible
;(idle mode)
;Each increment of the 16-bit timer register equals 256 microseconds at
;CK/1024 (i.e. 2 seconds should be about 0x1E84 increments,
;so load into timer register 0xE17B because it counts UP until
;overflow) **0xF85E for 0.5 second**
;EN1 and EN2 grounded; DATA1 on PC3, DATA2 on PC7; CLR1 and CLR2 on
;PD7. Control lines for 1 on PC0,1,2 and for 2 on PC4,5,6.
;*MOTOR CONTROLLER ADDRESS LINES MUST NOT CHANGE AT THE SAME TIME AS
; DATA LINES*

motor:     cpi r17, 0x55               ;must translate command char
                                           ; (in r17, U,D,L,R,F,B,W) to
                                           ; actuations

           breq brup                  ;char=U
           cpi r17, 0x44
           breq brdown                ;char=D
           cpi r17, 0x4C
           breq brleft                ;char=L
           cpi r17, 0x52
           breq brright               ;char=R
           cpi r17, 0x46
           breq brforwe               ;char=F
           cpi r17, 0x42

```

```

    breq brback          ;char=B
    cpi r17, 0x57
    breq brwarn         ;char=W
    cpi r17, 0x5A
    breq brana          ;char=Z
    cpi r17, 0x48       ; ** char=H
    breq brheart        ; ** char=H
    cpi r17, 0x53       ; **
    breq brrespir       ; ** char=S
    ldi r18, 0x3F
    rcall txchar        ;send ? for unrecognized char
    ret

brup:    rcall up
         ret
brdown:  rcall down
         ret
brleft:  rcall left
         ret
brright: rcall right
         ret
brforwe: rcall alternvert
         ret
brback:  rcall dotblink
         ret
brwarn:  rcall blink
         ret
brana:   rcall alternhoriz
         ret
brheart: rcall heart          ; **
         ret                  ; **
brrespir: rcall respir        ; **
         ret                  ; **

up:      ldi r16, 0x01
         out PORTB, r16
         sbi PORTD, 7          ;set CLR high
         clr r16
         out PORTC, r16
         sbi PORTC, 3
         ldi r16, 0x09
         out PORTC, r16
         ldi r16, 0x0A
         out PORTC, r16
         ldi r31, 0xF8        ;high byte of timer 1 count
         ldi r30, 0x5E        ;low byte of timer 1 count
         rcall delay
         cbi PORTC, 3
         ldi r16, 0x01
         out PORTC, r16
         clr r16
         out PORTC, r16
         ldi r16, 0x03
         out PORTC, r16
         sbi PORTC, 3
         ldi r16, 0x0C

```

```

out PORTC, r16
ldi r16, 0x0D
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
cbi PORTC, 3
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x03
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
sbi PORTC, 3
ldi r16, 0x0F
out PORTC, r16
ldi r16, 0x8F
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
cbi PORTC, 7
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

down:
ldi r16, 0x02
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x8F
out PORTC, r16
ldi r16, 0x8E
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x05
out PORTC, r16
ldi r16, 0x0D
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r16, 0x0B
out PORTC, r16

```

```

ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x03
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x05
out PORTC, r16
clr r16
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r16, 0x09
out PORTC, r16
ldi r16, 0x0A
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x01
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

left:
ldi r16, 0x04
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r16, 0x85
out PORTC, r16
ldi r16, 0x8D
out PORTC, r16
ldi r16, 0x8A
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x05
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x0F
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r16, 0x09
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count

```

```

rcall delay
ldi r16, 0x01
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x0B
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E          ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x03
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

```

```

right:
ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x0B
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E          ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x03
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x0F
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16

```

```

ldi r16, 0x09
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E          ;low byte of timer 1 count
rcall delay
ldi r16, 0x01
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x85
out PORTC, r16
ldi r16, 0x8D
out PORTC, r16
ldi r16, 0x8A
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E          ;low byte of timer 1 count
rcall delay
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x05
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

alternvert: ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
sbi PORTC, 3
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x0F
out PORTC, r16
ldi r16, 0x8F
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E          ;low byte of timer 1 count
rcall delay
cbi PORTC, 3
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
clr r16
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
sbi PORTC, 3
ldi r16, 0x09

```

```

out PORTC, r16
ldi r16, 0x0A
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x01
out PORTC, r16
clr r16
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x0F
out PORTC, r16
ldi r16, 0x8F
out PORTC, r16
ldi r31, 0xF8           ;high byte of timer 1 count
ldi r30, 0x5E           ;low byte of timer 1 count
rcall delay
ldi r16, 0x07
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

dotblink: ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r31, 0xF0           ;high byte of timer 1 count
ldi r30, 0xBD           ;low byte of timer 1 count
rcall delay
ldi r16, 0x04
out PORTC, r16
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r31, 0xF0           ;high byte of timer 1 count
ldi r30, 0xBD           ;low byte of timer 1 count
rcall delay

```



```

clr r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x0C
ldi r31, 0xF0           ;high byte of timer 1 count
ldi r30, 0xBD           ;low byte of timer 1 count
rcall delay
cbi PORTD, 7           ;set CLR low
ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
ldi r31, 0xF0           ;high byte of timer 1 count
ldi r30, 0xBD           ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x0C
ldi r31, 0xF0           ;high byte of timer 1 count
ldi r30, 0xBD           ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
cbi PORTD, 7           ;set CLR low
ret

blink:
ldi r16, 0x08
out PORTB, r16
sbi PORTD, 7           ;set CLR high
clr r16
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r16, 0x0A
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x8E
out PORTC, r16
ldi r31, 0xF9           ;high byte of timer 1 count
ldi r30, 0x22           ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x00
out PORTC, r16
ldi r31, 0xFE           ;high byte of timer 1 count

```

```

ldi r30, 0x79          ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r16, 0x0A
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x8E
out PORTC, r16
ldi r31, 0xF9         ;high byte of timer 1 count
ldi r30, 0x22         ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x00
out PORTC, r16
ldi r31, 0xFE         ;high byte of timer 1 count
ldi r30, 0x79         ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x08
out PORTC, r16
ldi r16, 0x0A
out PORTC, r16
ldi r16, 0x0C
out PORTC, r16
ldi r16, 0x0E
out PORTC, r16
ldi r16, 0x8E
out PORTC, r16
ldi r31, 0xF9         ;high byte of timer 1 count
ldi r30, 0x22         ;low byte of timer 1 count
rcall delay
clr r16
out PORTC, r16
ldi r16, 0x06
out PORTC, r16
ldi r16, 0x02
out PORTC, r16
ldi r16, 0x04
out PORTC, r16
ldi r16, 0x00
out PORTC, r16
cbi PORTD, 7          ;set CLR low
ret

```

```

alternhoriz: ldi r16, 0x08
              out PORTB, r16
              sbi PORTD, 7           ;set CLR high
              clr r16
              out PORTC, r16
              ldi r16, 0x06
              out PORTC, r16
              ldi r16, 0x0E
              out PORTC, r16
              ldi r16, 0x0B
              out PORTC, r16
              ldi r16, 0x08
              out PORTC, r16
              ldi r31, 0xF8         ;high byte of timer 1 count
              ldi r30, 0x5E         ;low byte of timer 1 count
              rcall delay
              clr r16
              out PORTC, r16
              ldi r16, 0x03
              out PORTC, r16
              ldi r16, 0x06
              out PORTC, r16
              ldi r16, 0x05
              out PORTC, r16
              ldi r16, 0x85
              out PORTC, r16
              ldi r16, 0x8D
              out PORTC, r16
              ldi r16, 0x8A
              out PORTC, r16
              ldi r31, 0xF8         ;high byte of timer 1 count
              ldi r30, 0x5E         ;low byte of timer 1 count
              rcall delay
              ldi r16, 0x02
              out PORTC, r16
              ldi r16, 0x05
              out PORTC, r16
              ldi r16, 0x06
              out PORTC, r16
              ldi r16, 0x0E
              out PORTC, r16
              ldi r16, 0x0B
              out PORTC, r16
              ldi r16, 0x08
              out PORTC, r16
              ldi r31, 0xF8         ;high byte of timer 1 count
              ldi r30, 0x5E         ;low byte of timer 1 count
              rcall delay
              ldi r16, 0x00
              out PORTC, r16
              ldi r16, 0x03
              out PORTC, r16
              ldi r16, 0x06
              out PORTC, r16
              clr r16
              out PORTC, r16

```

```

        cbi PORTD, 7          ;set CLR low
        ret

heart:   clr r3              ; ** clear low sum reg
        clr r4              ; ** clear high sum reg
        clr r5              ; ** clear number of
                                ; ** heartbeats register
        clr r1              ; ** clear r1
        clr r2              ; **
        out TCNT1H, r2      ; ** loads zero into timer high
        out TCNT1L, r1      ; **
hearttwo: ldi r16, 0x40      ; **
        out GIMSK, r16      ; ** enable external interrupt
                                ; ** INT0
        sleep              ; **
        nop                ; **
        nop                ; **
        nop                ; **
        clr r16             ; **
        out GIMSK, r16      ; ** disable int0 interrupt
        ldi r16, 0x05       ; ** loads the value 5 into r16
        cpse r16, r5        ; ** compare r16 (5) and r5 (number
                                ; ** heart beats + 1 reg), skip
                                ; ** next instr. if equal
        rjmp hearttwo       ; ** loop
        lsr r4              ; **
        ror r3              ; ** divides sum registers by 2
        lsr r4              ; **
        ror r3              ; ** divides again by 2, therefore
                                ; ** total of 4 (creates ave
                                ; ** number of cycles between
                                ; ** heart beats)
        mov r18, r3         ; ** makes a copy of r3 (low sum
                                ; ** reg byte) into r18
        rcall txchar        ; ** transmits
        mov r18, r4         ; ** makes a copy of r4 (high sum
                                ; ** reg byte) into r18
        rcall txchar        ; ** transmits
        clr r16             ; **
        out TCCR1B, r16     ; ** turn timer 1 off
        ret                ; **

respir:  clr r3              ; ** clear low sum reg
        clr r4              ; ** clear high sum reg
        clr r5              ; ** clear number of resp pulses
                                ; ** register
        clr r1              ; ** clear r1
        clr r2              ; **
        out TCNT1H, r2      ; ** loads zero into timer high
        out TCNT1L, r1      ; **
respirwo: ldi r16, 0x80     ; **
        out GIMSK, r16      ; ** enable external interrupt
                                ; ** INT1 (disable INT0)
        sleep              ; **
        nop                ; **

```

```

nop                ; **
nop                ; **
clr r16            ; **
out GIMSK, r16     ; ** disable INT1 (and INT0)
ldi r16, 0x05      ; ** loads the value 5 into r16
cpse r16, r5       ; ** compare r16 (5) and r5 (number
                   ; ** resp pulses + 1 reg), skip
                   ; ** next instr. if equal
rjmp respirwo     ; ** loop
lsr r4             ; **
ror r3             ; ** divides sum registers by 2
lsr r4             ; **
ror r3             ; ** divides again by 2, therefore
                   ; ** total of 4 (creates ave
                   ; ** number of cycles between
                   ; ** resp pulses)
mov r18, r3        ; ** makes a copy of r3 (low sum
                   ; ** reg byte) into r18
rcall txchar       ; ** transmits
mov r18, r4        ; ** makes a copy of r4 (high sum
                   ; ** reg byte) into r18
rcall txchar       ; ** transmits
clr r16            ; **
out TCCR1B, r16    ; ** turn timer 1 off
ret                ; **

;analog:          ldi r16, 0x80
;                 out PORTB, r16
;                 ldi r16, 0xD5           ;11010101 = D5 to clear ADIF first
;                 out ADCSR, r16
;                 clt                     ;clear T flag
;aloop:           in r16, ADCSR
;                 bst r16, 4              ;store bit 4 of ADCSR (ADIF) to
;                                         ; T flag
;                 brts axmit             ;branch to axmit if T flag is set
;                                         ; (conversion done, regs updated)
;                 rjmp aloop
;axmit:           ldi r16, 0x95
;                 out ADCSR, r16         ;95=10010101 to clear ADIF flag
;                 clt                     ;clear T flag
;                 in r18, ADCL           ;conversion done, read low byte and
transmit
;                 rcall txchar
;                 in r18, ADCH           ;read high byte and transmit
;                 rcall txchar
;                 ret

;*****
;DELAY ROUTINES

delay:            out TCNT1H, r31        ;always write HIGH byte first
;                 ; (assumes this is in R31 when
;                 ; called)
;                 out TCNT1L, r30      ;writing low byte (assumes this is
;                 ; in R30 when called)

```

```

        ldi r16, 0x05
        out TCCR1B, r16          ;turn on timer 1 with CK/1024
                                   ; prescale value

        sei
        sleep
        nop
        nop
        nop
        cli
        ret

wait:    ser r27                ;this subroutine just delays
waitlp:  dec r27                ;1 cycle per dec, 1 cycle per false
                                   ; breq, 2 cycles per rjmp =
                                   ; 1*255+1*254+2+2*254 = 1019
        breq goback            ;plus 1 for ser, 4 for ret = 5
        rjmp waitlp           ;total clock cycles = 1024, so at
                                   ; 4MHz, that's a delay of 256
                                   ; microseconds

goback:  ret

waitmore: ser r28              ;this should delay for 256*256us =
                                   ; 65.536ms

waitmlp: rcall wait
        dec r28
        breq dun
        rjmp waitmlp

dun:    ret

waitlong: ldi r29, 0x19        ;this should delay for 25*65.536ms
                                   ; = 1.6384s

waitllp: rcall waitmore
        dec r29
        breq dunl
        rjmp waitllp

dunl:   ret

;*****
;BLUETOOTH MODULE SOFTWARE RESET ROUTINE

btreset:rcall btcmd            ;issues reset command to Bluetooth
                                   ; module from software (in command
                                   ; mode)

        ldi r18, 0x41
        rcall txchar           ;send A
        ldi r18, 0x54
        rcall txchar           ;send T
        ldi r18, 0x55
        rcall txchar           ;send U
        ldi r18, 0x52
        rcall txchar           ;send R
        ldi r18, 0x53
        rcall txchar           ;send S
        ldi r18, 0x54

```

```

        rcall txchar          ;send T
        ldi r18, 0x0D
        rcall txchar          ;send cr
        ldi r18, 0x0A
        rcall txchar          ;send lf
        ret

;*****
;ERROR HANDLING ROUTINE

end:      clr r16
          out SREG, r16      ;disable interrupts
          ldi r16, 0x0AA
          out PORTB, r16     ;Port B set high/low alternating
endlp:    rjmp endlp

;*****
;;INTERRUPT SERVICE ROUTINES

rxisr:    in r20, USR
          sbrs r20, 4        ;check USR's FE bit for framing
                               ; error
          rjmp checkor
          rjmp end
checkor:  sbrs r20, 3        ;check USR's OR bit for overrun
                               ; error
          rjmp rxokay
          rjmp end
rxokay:   in r17, UDR        ;received char --> r17
          reti

tISR:     clr r16
          out TCCR1B, r16    ;turn off timer 1
          reti

exzero:   clr r16           ; **
          cp r5, r16         ; ** compare
          breq exztwo       ; **
          in r1, TCNT1L      ; ** loads timer low into r1
          in r2, TCNT1H      ; ** loads timer high into r2
          add r3, r1         ; ** adds timer low byte r1 to low
                               ; ** sum reg r3
          adc r4, r2         ; ** adds timer high byte r2 to
                               ; ** high sum reg r4 with carry
          inc r5             ; ** increments number of
                               ; ** heartbeats reg r5
          clr r1             ; ** clear r1
          clr r2             ; **
          out TCNT1H, r2     ; ** loads zero into timer high
          out TCNT1L, r1     ; **
          rcall waitmore     ; **
          rcall waitmore     ; **
          rcall waitmore     ; ** delay of about 0.21s
          ldi r16, 0x40      ; ** sets bit 6 high
          out GIFR, r16     ; ** clear int0 flag to erase any

```

```

; ** previous interrupts
; **
exztwo:  rjmp exzend          ; **
        ldi r16, 0x05      ; **
        out TCCR1B, r16   ; ** turn on timer 1 with CK/1024
        inc r5             ; ** prescale value
        rcall waitmore    ; ** increments number of
        rcall waitmore    ; ** heartbeats reg r5
        rcall waitmore    ; **
        rcall waitmore    ; ** delay of about 0.21s
        ldi r16, 0x40     ; ** sets bit 6 high
        out GIFR, r16     ; ** clear int0 flag to erase any
; ** previous interrupts
; **
exzend:  reti

exone:   clr r16           ; **
        cp r5, r16        ; ** compare
        breq exotwo      ; **
        in r1, TCNT1L     ; ** loads timer low into r1
        in r2, TCNT1H     ; ** loads timer high into r2
        add r3, r1        ; ** adds timer low byte r1 to low
        adc r4, r2        ; ** sum reg r3
        inc r5            ; ** adds timer high byte r2 to
        clr r1            ; ** high sum reg r4 with carry
        clr r2            ; ** increments number of resp
        out TCNT1H, r2    ; ** pulses reg r5
        out TCNT1L, r1    ; ** clear r1
        rcall waitmore    ; ** loads zero into timer high
        rcall waitmore    ; **
        rcall waitmore    ; **
        rcall waitmore    ; ** delay of about 0.39s
        rcall waitmore    ; **
        rcall waitmore    ; **
        rcall waitmore    ; **
        ldi r16, 0x80     ; ** sets bit 7 high
        out GIFR, r16     ; ** clear INT1 flag to erase any
; ** previous interrupts
; **
exotwo:  rjmp exoend      ; **
        ldi r16, 0x05      ; **
        out TCCR1B, r16   ; ** turn on timer 1 with CK/1024
        inc r5            ; ** prescale value
        rcall waitmore    ; ** increments number of
        rcall waitmore    ; ** resp pulses reg r5
        rcall waitmore    ; **
        rcall waitmore    ; **
        rcall waitmore    ; ** delay of about 0.39s
        rcall waitmore    ; **
        rcall waitmore    ; **
        rcall waitmore    ; **
        ldi r16, 0x80     ; ** sets bit 7 high
        out GIFR, r16     ; ** clear INT1 flag to erase any
; ** previous interrupts
; **
exoend:  reti

```