Final Report
Vital Signs Monitor
By
Maysarah Shahabuddin, David Knoff, and Jacob Adams
Team No. 5
Client: John Enderle, University of Connecticut, (860) 486-5521
# Table of Contents:

1 Abstract

2 Introduction
   a Background
   b Purpose
   c Previous Work
      i Products
      ii Patents
   d Map

3 Project Design
   a 3.1 Introduction
      i 3.1.1 Chest Strap
      ii 3.1.2 Control box
      iii 3.1.3 Body Temperature Biosensor
      iv 3.1.4 Blood Oxygen Biosensor
      v 3.1.5 Fall Detection Biosensor
      vi 3.1.6 Stethoscope Biosensor
      vii 3.1.7 ECG sensor
      viii 3.1.8 Heart/Respiration Rate
      ix 3.1.9 Weight Biosensor
      x 3.1.10 Blood Pressure Biosensor
      xi 3.1.11 LabVIEW Program
   b 3.2 Optimal Design
      i 3.2.1 Objective
      ii 3.2.2 Sub-units
         1 3.2.2.1 Chest Strap
         2 3.2.2.2 Master Controller
         3 3.2.2.3 Body Temperature Biosensor
         4 3.2.2.4 Blood Oxygen Biosensor
         5 3.2.2.5 Fall Detection Biosensor
         6 3.2.2.6 Stethoscope Biosensor
         7 3.2.2.7 ECG sensor
         8 3.2.2.8 Heart/Respiration Rate
         9 3.2.2.9 Weight Biosensor
        10 3.2.2.10 Blood Pressure Biosensor
        11 3.2.2.11 Blood Glucose
        12 3.2.2.12 LabVIEW Program

4 Realistic Constraints

5 Safety

6 Impact of Engineering Solutions

7 Lifelong learning

8 Budget and timeline
   a Budget
   b Timeline

9 Team Member Contribution

10 Conclusion
Abstract:
The Vital Signs Monitor will be an effective way of monitoring multiple signals from the patient, at home or in the clinic/hospital, through a single, simple device apparatus that wirelessly transmits data to a mobile tablet for processing and ease of use. Currently, there is no such singularly wireless device that acquires this multitude of signals including blood glucose level, blood oxygen level, blood pressure, core body temperature, ECG, EMG, motion detector (for falls), respiration/heart rate, and patient’s weight.

For this project, we will combine as many of these data collection devices as we can for ease of access, safety, and portability to provide for patients with little to no clinical expertise as well as providing clinicians benefits from this system by monitoring their patients in an effective manner. Our end goal is for this Vital Signs Monitor to be an inexpensive yet powerful solution for our potential customer’s uses.
2. Introduction:

2.1 Background:
Professor Enderle has been a member of the UConn Electrical Engineering department. Among his many research projects, he currently focuses on the innervation of eye and ear muscle systems. Through his research he aims to define an accurate and comprehensive model for simulating eye movements. A well respected professor, and researcher in the biomedical engineering field, his work continues to contribute to the rising interest and success of Biomedical engineering.

Alongside his numerous research projects, Professor Enderle has been working on this Vital Signs monitor for several years. His aims are to provide nonmedical personnel with the ability to obtain a great array of commonly needed signals that the body provides in the form of a chest strap worn around the patient’s body. This comprehensive point of view can then be sent to a doctor and used to keep track of a patient’s health in a manner that does not require constant supervision. Another benefit to this method is that the patient can also continue on their day, eliminating the need for an expensive visit to the doctors, as well as capturing a more accurate glimpse into the body during the daily life of a patient. Finally the presence of automated sensors reduces the need for in-home nurses, and hopefully will lower the costs of obtaining all the important measurements for anyone who needs them.

2.2 Purpose:
Many patients cannot spend the money or the time needed to obtain all of the data for such an extensive set of observations. Current monitors are bulky and contain many wires, which both cause hazard to the patient and people using them. Since the design aims to be mobile, certain considerations must be met to keep the weight and size of the device low. These limits to the patient’s accessibility to such care, and in many cases, the availability of up to date and accurate information on demand on the part of the doctor can result in many mistakes and faulty diagnoses of a patient.

On the other side of the health problem, during rehabilitation and recuperation, if the proper care is not observed, there may be issues that can return the patient to a state of illness. This device could introduce the concept of minimal upkeep and observation on the doctor’s part to ensure the proper actions are taking place. The Vital Signs Monitor will provide consolidated, continuous, and archived reports of the patient’s health status for the use of the patient, nurses, physicians, and any future caretaker. This allows the patient’s history of vital signs to be analyzed as the patient progresses through certain health conditions, rehabilitation or just normal life. Each vital sign is important in monitoring the patient’s health as well as the ability to call for aid if needed.

2.3 Previous Work:
In this project we will be picking up on a decade of work left by various professors and students. Most recently, the undergraduate Tom Capuano has revamped the entire project by reprogramming the circuit board, and finalizing the ECG function. The existing device is broken into three parts; the controller, the devices, and the Bluetooth program. The circuit board controller is programmed in C, and takes all of the inputs and selects the channels to send via Bluetooth to the computer. Each of the individual devices has their own output signal, and upon measuring, sends the data to the controller. The computer, upon receiving the Bluetooth signal takes the data and uses Bluetooth to analyze the signal.
However the existing work does not meet the specifications for accuracy that Professor Enderle requires, so each of the existing sensors needed to be rebuilt to meet our clients specifications.

2.3.1 Products:
Currently, there are other products existing in the market under the multiple sensor, vital sign monitor device. None of these current devices contain as many sensors as we are implementing or the usage of how this device is going to work. Devices like the Creative Medical® PC-900 monitoring system only records non-invasive blood pressure, pulse oxygen saturation, pulse rate, ECG, and temperature with the display on a standalone device while other vital sign monitors can only record one parameter or set of data. Other devices such as the Multi-Parameter Vital Signs Monitor (CareTaker™) by David W. Gerdt require disposable apparatuses but can transmit Bluetooth signals to other devices.

We also integrated already existing standalone products into the design of our device as incorporating some Vital Signs onto the main chest strap mentioned in section 2.1 was unfeasible such as patient’s weight, blood pressure, ECG, and blood glucose. The existing products being used in our device are the Withings WS-30 weight scale and Bluetooth Blood Pressure cuff, iHealth Bluetooth Blood Pressure Cuff, Texas Instruments ADS ECG chip, and the Fora G31B Bluetooth Blood Glucose monitor. Using the wireless nature of these pre-existing products fits into our design of acquiring signals onto one device to display to the user. However useful these devices are, not one of these single devices can incorporate more than 2 Vital Sign signals.

What sets our product apart from these other devices is that our Vital Sign monitor includes many more vital sign parameters, that other devices don’t have, including: ECG, temperature, heart rate, respiration rate, heart and lung sounds, blood oxygen level, blood glucose level, fall detection and patient weight. With all of these parameters being converted into electrical signals, the data can be transmitted wirelessly to a tablet such as the Microsoft Surface for filtering, analysis, and display for the patient, which many of these preexisting products are incapable of. Lastly, other multi-parameter vital sign monitor devices can be expensive, into the thousands of dollars range, which we will try to minimize as outlined in further sections.

2.3.2 Patent Search Results:
There was only one patent (US20120022348) that fit the description of a multi-parameter vital sign monitor device. Other patent searches included only standalone devices that could display up to 2 signals, such as blood pressure cuffs able to display heart rate and blood pressure or ECG devices able to measure ECG waveform and heart rate. Another device that is able to measure a few signals is the Bodymedia Fit series which have the ability to measure body temperature, skin temperature, heart rate, caloric intake and number of steps taken. Lastly, a device such as this that we have designed is new and highly desired in the field as Qualcomm is offering a $10 million prize awarded to such a wireless, comprehensive, and modular device.

2.4 Report Map
In subsequent sections, we will show how this device will function starting with the design of the main device strap around the patient’s torso. From there, we will show how each sensor will collect the different types of data necessary for our goals and how the device will sort
the data inputs. Lastly, we will provide insight to further enhancement of the device and other implementation of how the Vital Sign monitor will better the lives of patients and doctors.

3 Project Design

3.1 Introduction

3.1.1 Chest Strap

The chest strap is a comfortable, adjustable strap with 4 open leads. The sensors use pins to align and connect to the 4 leads in order to sit in position and collect data. The 4 leads from top to bottom correspond to ground, a clock and data line, both needed for I²C communication, and voltage source for the microcontrollers. Each of the sensors draw a small amount of power for their continuous operation while waiting for a data request, and then only turn on the bulk of their collection instruments when the data is requested, and then turn off after the data has been sent. The strap will consist of a stretchable conductive cloth sewed onto T-shirt cloth to eliminate the need for wires.

3.1.2 Control box

The master controller is comprised of three separate sections behind the theory of operation. The first and most prominent division will be the master controller for the I²C communication protocol. This will be responsible for choreographing all of the communication between slave controllers and the Bluetooth antenna. The second division will be the fall detection code. This will monitor the acceleration of the chest area of the wearer. This is intended to detect a fall by alertsing the base computer of rapid acceleration changes of the control box. The third section of the master control unit is the Bluetooth communication antenna. Since this is where all of the data from sensors will be kept, this is also the location responsible for sending the data to a computer running LabVIEW. The battery and power source for all sensors will also originate from here to minimize the overall weight and maintenance of the device.

3.1.3 Body Temperature Biosensor

The body temperature biosensor will require the least amount of power of the biosensors and will take a reading of the patient’s body temperature from inside their armpit upon request from the master controller. Specifications for our design include a fast response time and a high surface area to volume ratio. If a patient or caretaker commands the LabVIEW program to measure the patient’s body temperature, current will run through the Negative Temperature Coefficient (NTC) thermistor. The LabVIEW program will convert the voltage drop across the NTC thermistor to resistance using a voltage divider equation and convert resistance to temperature based on the NTC thermistor’s material constant. The patient’s body temperature will then be outputted to the front panel in degrees Fahrenheit. The circuit construction of the body temperature biosensor ensures the least amount of error by eliminating lead wire noise using a 4-wire Wheatstone bridge.

3.1.4 Blood Oxygen Biosensor

The pulse oximeter will clip directly to the chest strap, but will have a smaller clip containing a light to frequency converter coupled with two LED’s to clip onto the ear. The ear clip will extend via wires of length of around 10-15cm. This will not hamper the subject’s range
of motion, while still maintaining a firm connection to the microcontroller. The master controller will request data from the sensor which will then briefly turn on the lights to get an intensity reading from each of the red and infrared frequencies. The absorbances are sent to the LabVIEW program and the ratio of oxygenated to de-oxygenated blood will be calculated to obtain the blood oxygen content using the following equation:

3.1.5 Fall Detection Biosensor

The fall detection sensor is an ADXL345 triple axis accelerometer to detect static and dynamic acceleration in all three axes. The ADXL345 has a high sensitivity of 4 mg/LSB which permits accurate acceleration readings along with inclination measurements of 1 degree. The accelerometer has the capability to detect when a patient is actively falling, fallen, and unresponsive based on programming interrupts. Based on this data, an alarm function will provide the ability to alarm LabVIEW in the event that the patient’s acceleration reaches or exceeds the programmed threshold acceleration. The fall detection biosensor has the most ability to be improved regarding its alarm functionality because of its ability to sense a fall before the patient has hit the ground. Possible future improvements could include a preemptive action being triggered based on a fall being detected which somehow increases the safety of the patient before they hit the ground. At the very least, the fall detection biosensor allows the quickest possible assessment of the patient’s status and the ability to use Bluetooth to contact emergency services.

3.1.6 Stethoscope Biosensor

The stethoscope we have chosen to use is the electronic microphone patch developed by Chad Lyons. His design creates a pair of electrical signals through a piezoelectric sensor which is sent to an instrumentation amplifier, to subtract the two signals from each other and amplified. Since the Arduino can only read analog signals above 0 volts, a DC offset needs to be introduced. Further, since the signal is outputted with a peak of 2.3 mV, a 741 amplifier will be used with a 3 volt source to amplify the signal 200 times to an amplitude of .5 volts. In addition to counting the respiration rate in LabVIEW, we include the ability to play the signal back in LabVIEW so that the patient or physician can listen to the signal. This minimizes the need to remove the stethoscope repeatedly if needed to visit the doctor, increasing the longevity of a single sensor. Due to issues raised by Dr. Lyons however, the use of a single stethoscope will be limited by the adhesion properties of the patch, and thus the sensor clip will come built in with the ability to quickly swap out the patches as needed.

3.1.7 ECG Sensor

The ECG biosensor will use a standalone Texas Instrument ECG Front End Performance kit to accurately measure a patient’s ECG signal, heart rate, and respiration. The 12 lead configuration will plug into the analog front end of the Texas Instruments ECG Front End Performance kit to increase signal clarity by providing sufficient references which detect electric potential across the chest. Active filters remove noise from respiration, in addition to pink and white noise. This signal will then be sent to a LabVIEW based Texas Instruments program on the Microsoft Surface for the patient and caretakers to view and analyze. This analysis tool allows researchers and caretakers to use signal integration to compare vital signs, for example using the stethoscope and ECG signal to make correlations regarding diseases or abnormalities.
3.1.8 Heart/Respiration Rate

Heart rate is an important vital sign of all patients, if the rate is at an unusual pace the patient is under cardiac arrest and their heart could stop at any moment. This vital sign can be calculated using multiple sensors including the pulse oximeter, ECG sensor, Blood Pressure cuffs, or stethoscope. In our optimal design, we are using the Blood Pressure cuffs to send heart rate data in conjunction with the ECG chip to do the same measurement.

Respiration rate is a little more difficult to compute as we don’t have a pressure transducer measuring chest movement or microphone in front of the nose/mouth to acquire these signals. A solution to this issue is to take the stethoscope data and use LabVIEW to count the peaks over specified time to calculate the respiration rate. We suggest placing the stethoscope on the patient’s back of the strap to avoid listening to heart sounds which would create artefacts in the signal. In addition, the TI ECG Front End Performance kit has the ability to measure respiration rate based on respiration pneumography. This process sends a square wave signal across the chest through reference leads and measures the resistance that occurs during each breath.

3.1.9 Weight Biosensor

Patient’s weight is an important aspect to their overall health, as obesity can cause serious health concerns. Their weight will be monitored through a separate apparatus from the overall device but will still communicate via Bluetooth to a separate device which will store the data on an online server as a Microsoft Excel file. From this Excel file, LabVIEW will take this data and graph it with the corresponding date/time of measurements taken. The standalone device we chose was the Withings WS-30 as it is Bluetooth enabled and has a good interface in itself.

3.1.10 Blood Pressure Biosensor

Blood pressure is another majorly important aspect to the vitals of any patient as any sudden drop in blood pressure can mean heart failure and cause death. In order to monitor this signal, a blood pressure cuff will be worn on the arm which will periodically take data and transmit it via Bluetooth to a separate device just like the weight sensor. We chose to use two devices including the Withings Blood Pressure cuff and the iHealth Blood Pressure cuff as both devices have certain advantages.

3.1.11 LabVIEW Progamming

National Instruments: LabVIEW programming software is an important aspect to this device. Without this software, data analysis of these signals, transmitted via Bluetooth, would be near impossible to program/code and display to the user would be difficult if coded in other means. LabVIEW has fantastic, easy user display, as well as superior data analysis collection and processing. Bluetooth sub-programs in LabVIEW aid in this process of data collection which is a huge advantage to the device in the sense that the patient can be mobile and not have to worry about carrying their base station on person and the main program will handle the data being transmitted. Also, LabVIEW has multiple functions in analyzing many types of data from waveforms to arrays and clusters which can be manipulated to display to the user of what their Vital Signs are. The main LabVIEW program will also need some direct user input to connect to which Bluetooth device is being used. Lastly, the viewer will see Fig 1 as the front panel when
the Vital Sign monitor is collecting data.

3.2 Optimal Design

3.2.1 Objective

Our group’s interpretation of how best to organize this project suggests that the simplest design is the best. As any engineer should know, the less complicated something is, the less things that can break. We have decided that the best way to organize the vital signs monitor is to minimize the number of components to the bare minimum. This means that each sensor will be a part of a whole as opposed to being individual devices sharing only proximity. For a device that intends to provide a comprehensive look into the human body, we can do an optimal job of this by minimizing the sensors and stripping them of their luxuries. This way we can get rid of one Bluetooth module for each sensor, as well as remove the need to replace/charge several batteries. By separating everything, we can work backwards from the “the whole is greater than the sum of its parts” philosophy, and make each part an individually smaller portion that sums to the original whole.

The chest strap will be sweat resistant, elastic, soft, breathable material on the side closest to the body. On the other side, facing away from the patient however, there will be a few “rails” of open electrically conductive strips running parallel to each other throughout the length of the strap. The purpose of these striped wires will be to provide a ground, a power source, and a data channel. Connecting to each of these wires will be a box to which the chest strap originates. This box will be worn such that it will lie between the scapulae, and not impede any daily function of the user. This box will provide power and ground along two of the three parallel stripes. The third will be a data wire used to communicate between sensors and the main strap computer. This box will then take all of the data (communicated via a wired connection), consolidate it into a signal, and communicate via a Bluetooth chip to whatever base station or enabled device of one's choosing. The rails will provide power, and a means of communicating to the chest strap box, which will carry the brunt of the data processing load.

The sensors then will be used by sliding the clips into the strap such that the conducting pins will connect and provide power to each individual sensor (see Figure 2). A third and fourth pin will then be used to communicate the single channel of data to the box. Communicating multiple sources of data on a single wire can be accomplished using the I2C communication bus developed by Phillips. I2C is built into the existing Arduino library making it easy to use. Each sensor will be programmed to handle its own input, translate it into a language communicable via this single wire, and then communicate to the main box on the chest strap back. This reliable method of data transfer will offer built in support for a variable amount of sensors (if two temperature sensors are needed for example) because each sensor will have its own unique serial number so the same type of data won’t confuse the microcontroller. The main box will then take each of the sensors data, and prepare it for interpretation by the existing circuitry. Ultimately the presence of three wires in the strap may or may not be necessary, dependent on whether we use this communication method or not. Since it is low power and low speed, the novelty of minimizing the expose wires will need to be addressed.

This system is efficient for several reasons. It will be cheaper to produce, lighter, user friendly, and customizable. It will be cheaper to produce because each sensor will be small, and lack complicated communication relay, or batteries. Due to the lack of batteries in each device, the total weight of all sensors will be minimized, adding to the comfort of the user. The modular
nature of each sensor means that replacement of faulty parts is simple. Either replaces the main battery, or a cheap sensor module, not both. Diagnosing such problems will be incredibly easy since either all parts are powered, or just one is not relaying data, or no data is being interpreted, but power is being sent to them. Troubleshooting will essentially be reduced to ensuring the proper contacts are maintained, and placing the sensors (which by attaching via clips, are variable in their placement) in the proper locations. From there, it would be easy to isolate which, if any, parts are not working. Since the wires run across the entirety of the strap, sensor position can be adjusted for any patient by clipping each of them wherever they may be optimally positioned on their body. Finally it is customizable by allowing doctors to purchase only specific sensors. Each one will be self-contained, and will be able to run independently of other sensors.

3.2.2 Subunits
3.2.2.1 Chest strap

We chose to use a strap made of T-shirt material primarily for its similarity to intended use, as well as its availability, the strap will be embedded with 4 bare strips of medtex-130 conductive cloth. We chose the conductive cloth for its wire like properties as well as its marriage with t-shirt like properties. With both of them sewn into to same backing, there is minimal difference in the stretchiness of regular t-shirt cloth and this strap. There need to be 4 pathways because there are two needed for power and ground respectively, while the two middle pathways will be for \( \text{I}^2\text{C} \) Clock signal, and \( \text{I}^2\text{C} \) Data Signal. \( \text{I}^2\text{C} \) works by using the clock wire to choreograph each bit sent from each sensor connected to the data wires. The clock channel will also send an activation signal to the listening sensors. Since each sensor will not send data unless it is its “turn” we can use the activation as a trigger for a power save mode. Once the master requests that the sensor start sending data, the sensor can take a moment to begin drawing power, take measurements, send data and then stop and enter a low power state once the stop bits have been received. A limitation to \( \text{I}^2\text{C} \) is that the data cannot be transmitted simultaneously from multiple sensors, however with a data transmission rate of 2-5 times per second, and a relatively low volume of data, this is unlikely to be a problem. The strap will be connected at one end inside the master controller box, and the other end will loop through a hook on the other end of the box to be cinched, where a simple Velcro strap may be used to fasten it around the user at their comfort.

Figure 1: Front section of chest strap
3.2.2.2 Master controller

The main controller box will use a more powerful microcontroller than any inside the sensors and will coordinate the communication from the sensors to the base station. This will involve programming revolving around two modes of communication. The main box will first take the I\textsuperscript{2}C data received from all the sensors. This microcontroller will be the master responsible for taking and requesting data from the sensors and then send it to Bluetooth. It is important to keep track of the device identifiers while adding sensors to the strap because each one needs to have a unique identifier associated with it during this section of the code. This is implemented in LabVIEW where it will request a specific device identifiers data. The VI will then read the data streamed, and as needed process and analyze the raw data to display into the desired format. The VI is explained in the relevant section below.

The code included below for the master controller lays out the basic information collected by the I\textsuperscript{2}C interface and describes the loop by which it will wait to request data from each sensor and pass it via Bluetooth to the computer.

```c
// Echo Program
// 0: Low Power mode
// 1: Temperature
// 2: SpO2
```


// A: Start Stethoscope read
// B: Stop Stethoscope read
#include <Wire.h>
int incomingByte;
int response = 0;
//int numDevices = 1;
int temp = 0;
int temporary = 0;
int DeviceID;
int temperatureData = 1;
//int spo2Data = 0;

void setup()
{
    Serial.begin(115200);
    Wire.begin();
}

void loop()
{
    //Serial.println("TempData: ");
    if (Serial.available()>0)
    { // read the oldest byte in the serial buffer:
        incomingByte = Serial.read();
        // Serial.print("incomingByte= ");
        // Serial.println(incomingByte);
        // Request/reply
        // while (incomingByte != 0)
        // {
        //   int temperatureData = collectData(1);
        //   //int spo2Data = collectData(2);
        // }
        if (incomingByte == '1')
        {
            //response = 1;
            DeviceID = 1;
        }
        if (incomingByte == '2')
        {
            //response = 2;
            DeviceID = 2;
        }
        if (incomingByte == '0')
        {
            //response = 0;
            DeviceID = 0;
        }
        if (incomingByte == 'S')
        {
            response = 'S';
            //DeviceID = 3;
        }
        if (incomingByte == 'T')
        {
            response = 'T';
            //DeviceID = 3;
        }
        //Serial.println("Response: ");
        temporary = collectData(DeviceID);
        //Serial.println("Data collected");
        Serial.println(temporary);
        Serial.println(response);
        delay(1000);
    }
}

int collectData(int deviceID)
{
    //ask for data from the device with the current identifier
    Wire.requestFrom(deviceID, 8, true); // Takes data from deviceID, an amount of "8" bytes, and (true) releases the slave when done
    while(Wire.available()) // The slave may send less than requested
        { // Read the oldest byte in the serial buffer:
            incomingByte = Wire.read();
            // Serial.print("incomingByte= ");
            // Serial.println(incomingByte);
            // Request/reply
            // while (incomingByte != 0)
            // {
                // Collect data from the device:
                temperatureData = collectData(1);
            // }
            if (incomingByte == '1')
            {
                //response = 1;
                DeviceID = 1;
            }
            if (incomingByte == '2')
            {
                //response = 2;
                DeviceID = 2;
            }
            if (incomingByte == '0')
            {
                //response = 0;
                DeviceID = 0;
            }
            if (incomingByte == 'S')
            {
                response = 'S';
                //DeviceID = 3;
            }
            if (incomingByte == 'T')
            {
                response = 'T';
                //DeviceID = 3;
            }
            //Serial.println("Response: ");
            temporary = collectData(DeviceID);
            //Serial.println("Data collected");
            Serial.println(temporary);
            Serial.println(response);
            delay(1000);
        }
}
{ temp = Wire.read(); // receive a byte as character
  // Format the data and then return it
} return temp;

Figure 1: Master Controller Program

Figure 4: Master controller communicating with slave controller
Figure 5: Sensor clip

3.2.2.3 Temperature sensor
This small device will clip on the chest strap behind the armpit and a Negative Temperature Coefficient (NTC) thermistor probe will protrude out of the housing and be placed in the armpit to get an accurate surface body temperature from the patient. This is the closest location for core body temperature that will be accessible to the chest strap. The inside of the housing will contain a PCB circuit which has four contacts for the power, ground, clock and data transmission. The resistance of the NTC thermistor will change as the patient’s surface body temperature changes. This change in resistance can be sensed by the microcontroller based on the voltage drop across the NTC thermistor. The LabVIEW program will convert the voltage drop across the NTC thermistor to resistance using a voltage divider equation and convert resistance to temperature based on the NTC thermistor’s material constant. The LabVIEW calculations can be viewed below in Figure 3.

The body temperature biosensor circuitry will consist of a 4-wire bridge to eliminate lead impedance which can create error in the temperature measurement. Figure 1 shows the NTC thermistor circuit.

Figure 6: NTC Thermistor Sensor Multisim Diagram
Where R1, R2, and R3 are 2000 ohms. The resistance of the NTC thermistor will be determined by reading the voltage at node B with respect to the voltage at node D. Using this value and Equation 3, shown below, we can find the resistance $R_T$ which will be plugged into Equation 2.

$$V_G = \left( \frac{R_x}{R_3 + R_x} - \frac{R_2}{R_1 + R_2} \right) V_s$$

Solving this using the known values will output the temperature in Kelvin which will be converted to Celsius and Fahrenheit. I am writing a subVI which will perform these calculations and output temperature.

The temperature sensor will be powered by an ATtiny4313 microprocessor, which has 4 kilobytes of on board memory, and support for the Arduino bootloader. This small chip can use one of its serial ports as to detect voltage. This voltage reading is then sent via I2C to the master controller, where upon connecting to the base station will convert it to a resistance and subsequently, a temperature reading. The will be used to calculate the temperature from voltage due to memory considerations because an early draft program that attempted to calculate the temperature on the sensor itself ended up being around 8 kilobytes. Outsourcing the math to the LabVIEW program proved to be a much better decision because the computer will typically have less size and power restrictions than the sensors themselves. In order to minimize the power consumption, the voltage divider will be connected to power via a digital switch. This will remain open until the data is requested from the master controller. At this point, the switch will momentarily close, the serial port will read the voltage out, and then the switch will open again. This ensures that there is no power wasted on continuously running a voltage divider. The code so far for the temperature sensor is included below. This is a combination of serial read statements, and the Wire library slave write functions:

```c
#include <Wire.h>
int analogPin = 0; // potentiometer wiper (middle terminal) connected to analog pin 3
                     // outside leads to ground and +5V
float Vout;        // variable to store the output voltage
int ID = 2;        // I2C address

void setup()
{
  Wire.begin(ID);  // turn on power to the thermistor using a switch
  float Vout = takeVoltage();
  Wire.onRequest(requestEvent);
  Serial.begin(9600);   // Setup serial
  digitalWrite(13, HIGH);  // Indicates that the program has initialized
}

void loop()
{
  //testbed
}

void requestEvent()
{
  char voltage = char(Vout);
  Wire.write(voltage); // Write the voltage value to the master
}

float takeVoltage()
{
  Vout = analogRead(analogPin);  // Reads the Input PIN
  return Vout;
}
```
Figure 7: Body Surface Temperature Arduino Program

The voltmeter represents the output data of the temperature biosensor which is then processed by LabVIEW. Using the output voltage, the program will process the data to find the final resistance of the RTD and evaluate the patient’s body temperature by comparing the final resistance to the initial resistance as seen in Figure 3.

The product is model 427 reusable skin/surface probe made by Measurement Specialties. It operates under the 400 series nominal resistance response curve. The nominal resistance is 2252.4 ohms at 25 degrees Celsius including the lead wire resistance. We cut the wire which will decrease the lead wire resistance, therefore the resistance diminished to 1650 ohms at the temperature in the lab. I have selected three 2000 ohm resistors to complete the wheatstone bridge circuit. Testing the probe in my armpit gave a reading of 1100 ohms using a multimeter. The equation for the NTC thermistor response is seen in Equation 1 below.

\[
R_T = \frac{R_{25}}{\exp \left( \frac{B_{25/85} \cdot \left( \frac{1}{T_{25}} - \frac{1}{T} \right)}{R_{25}} \right)}
\]  

\( R_T \) is the thermistor resistance, \( T \) is its temperature in Kelvin, and \( T_{25} \) is the Kelvin temperature at 25 degrees Celsius (298.15 K). Solving Equation 1 for temperature (in Kelvin) we get Equation 2, shown below.

\[
T = \frac{B_{25/85} \cdot T_{25}}{B_{25/85} - T_{25} \cdot \ln \left( \frac{R_{25}}{R_T} \right)}
\]

\( B_{25/85} \) is the material constant, 3976 K for our NTC thermistor surface probe. \( T_{25} \) is 298.1 K, \( R_{25} \) is roughly 2000 ohms but will be confirmed before we enter it into code, and \( R_T \) is the resistance of the NTC thermistor which will be exported as data from the Arduino microcontroller to the LabVIEW program via Bluetooth. Figure 2 shows the Wheatstone bridge circuit with the NTC thermistor element.
Figure 9: LabVIEW Surface Body Temperature SubVI Program

The simulated signal represents the signal sent from the body temperature biosensor. Once processed, the final output displays the user’s body temperature to the user on their Microsoft Surface tablet.

Figure 10: Front panel of Temperature sensor  Figure 11: Circuit board for Temperature
3.2.2.4 Blood Oxygen monitor

Figure 12: Absorbances of hemoglobin and saturated hemoglobin

Pulse oximetry is the measure of blood oxygen content using the optical properties of hemoglobin. Oxygenated hemoglobin absorbs infrared light and lets red light pass through, while deoxygenated hemoglobin displays the opposite properties. Pulse oximetry is thus accomplished by shining a red and an infrared light through a translucent part of the body (useful contact points for this are the finger, toes, the earlobe, and the tip of the ear). Opposite to this point is a light detector that detects the light that has passed through the thin section of the body. After the sensor detects this, a computer calculates the ratio of absorbed red to absorbed infrared light ratio.

Upon obtaining this ratio, the software can then consult a table of values, usually from reference, or calibration data, to determine the blood oxygen content ratio. There are two placements of sensors for detection, transmittance and reflectance. In transmittance, the sensor sits opposite to the light sources, and the absorbance is calculated dependent on the light that passed through the skin.

Figure 13: Pulse oximetry on a finger, with two LED’s

In reflectance, the sensor is placed on the same side as the light sources, and the light that is returned is used to obtain measurements. Since the two methods require two different theories behind operation, and transmittance is more accurate, we have chosen to use transmittance as our method.

A blood oxygen monitor needs two different LEDs to shine through a thin portion of the patient’s skin, in this case we chose the earlobe. This was for its out of the way location coupled
with the thin skin in the area. A detector then picks up the light that has shined through the patient and a ratio comparing the two absorbance’s must either be calculated on the sensor itself, or be sent to the controller, dependent on the limitations provided by the microchip on board, and the requirements of the I²C communication protocol. This is then plugged into a function, or compared to an absorbance curve based on standard data to determine the blood oxygen content in the blood. In this design, it would be sensible to place a sensor box on the patients back, and have a wire that contains the clip with two LED’s and the one photosensor snaking up under the shirt to the patients earlobe. This thin area of the body also serves as a good location for pulse oximetry compared to the finger, and has the added benefit of being very near the chest strap. A small cable no longer than 10 inches will be all that is needed to ensure that the signal can be obtained by the sensor wherever it is placed. The earlobe clip will need to have power going to both sides, with just a power and a ground wire for the lights, and a third wire for the photosensor. We will be using a light to frequency converter to simplify the intensity acquisition. This will then perform the following calculation, which returns the ratio of oxyhemoglobin to hemoglobin:

\[
SpO2 = \frac{\alpha_{r2} - \alpha_{r1}}{(\alpha_{r2} - \alpha_{o2})R - (\alpha_{r1} - \alpha_{o1})}
\]

Where

- \(\alpha_{r2}\) = absorption coefficient of hemoglobin at 950 nm
- \(\alpha_{r1}\) = absorption coefficient of hemoglobin at 620 nm
- \(\alpha_{o2}\) = absorption coefficient of hemoglobin at 950 nm
- \(\alpha_{o2}\) = absorption coefficient of hemoglobin at 620 nm
- \(R\) = Ratio of absorbances

Once that data is sent directly to the chip, the ratio and then blood oxygen content can be transmitted to the controller box.

```cpp
//PulseOx
#include <Wire.h>
int redPin = 2;//Pin connected to the Red LED (Digital output)
int infPin = 3 ;// Pin connected to the Infrared LED (Digital output)
int converterPower = 5;// Pin connected to the Frequency converter power (Digital output)
int converterOut = 4 ;// Pin that l2f converter sends square wave to (Digital input)
double redFreq;
double infraredFreq;
float ratio;
int deviceID = 2;//I2Caddress
void setup()
{
    pinMode(redPin,OUTPUT);
    pinMode(infPin,OUTPUT);
    pinMode(converterPower, OUTPUT);
    pinMode(converterOut, INPUT);
    Wire.begin(deviceID);
    Wire.onRequest(requestEvent);
    Serial.begin(9600);             // Setup serial
}
void loop()
{
    // takeAbsorbance();
    // Serial.print("redFreq: ");   //
    // Serial.println(redFreq);     // Outputs the information
    // Serial.print("infraredFreq:"); //
    // Serial.println(infraredFreq);
    // Serial.print("Ratio: ");
    // Serial.println(ratio);
```
// double pulseox = calcAbsorbance(redFreq, infraredFreq);
// Serial.print("SpO2: ");
// Serial.println(pulseox);
// delay (200);
}
void requestEvent()
{
    takeAbsorbance();
    int spO2 = calcAbsorbance(redFreq, infraredFreq);
    Wire.write(spO2);
}
void takeAbsorbance()
{
    redFreq = blinkFreq(redPin, 100, converterPower, converterOut);
    // turn on red light, read frequency
    infraredFreq = blinkFreq(infPin, 100, converterPower, converterOut);
}
double blinkFreq (int ledPIN, int time, int powerPin, int readPin)
{
    digitalWrite(ledPIN, HIGH);  // turn the LED on (HIGH is 5 volts)
    double frequency = getFrequency(powerPin, readPin);
    delay(time);               // wait for a millisecond
    digitalWrite(ledPIN, LOW);  // turn the LED off by making the voltage LOW
    return frequency;
}
long getFrequency(int powerPin, int pin)
{
    digitalWrite(powerPin, HIGH);
    int samples = 4096;
    long freq = 0;
    for(unsigned int j=0; j<samples; j++)
    {
        freq+= 500000/pulseIn(pin, HIGH, 250000);
    }
    return freq / samples;
    digitalWrite(powerPin, LOW);
}
int calcAbsorbance(double red, double infrared)
{
    float redIntensity = (red - 0.4)/(581.4);
    float infraredIntensity = (infrared - 0.4)/(581.4);
    ratio = log10(redIntensity)/log10(infraredIntensity);
    double realspo2 = (602.24*(ratio)-6509.6)/((602.24-1024)*(ratio)-(5567.6));
    double spo2 = (602.24*(ratio)-6509.6)/((602.24-1024)*(ratio)-(6509.6-942));
    int spo2 = realspo2*1000;
    return spo2;
}

Figure 14: SpO2 code
3.2.2.5 Fall Detector

In interest of keeping as many sensors as close to the chest strap as possible (to remove the hassle of peripheral devices), the fall detection biosensor will use a triple axis accelerometer attached to the chest strap to detect abnormal 3D acceleration. As opposed to extremities such as the arms or legs, the core of the patient’s body will only have a drastic change in motion and positioning during a fall. Extremities will have constant change in motion and positioning during daily activities. The triple axis accelerometer being placed on the chest strap allows the fall detector biosensor to evaluate the patient’s stability closer to their center of gravity which should allow for more accurately identifying when the patient falls. The data sent by the fall detection biosensor will be calibrated to prevent a false identification of a fall and allow proper evaluation of abnormal movements in the patient’s center of gravity. The triple axis accelerometer detects proper acceleration, or acceleration relative to free fall, in all three axes of movement. After processing the data using a microchip to determine whether abnormal motion occurred, the output will be sent to the LabVIEW program which will indicate the patient’s stability as well as whether they have fallen. This biosensor will send an alarm signal to LabVIEW in order to indicate that an emergency situation such as a fall or seizure has occurred based on the patient’s acceleration reaching or exceeding a threshold acceleration.

In order to continuously monitor the patient to detect a fall at any point, the fall detection biosensor will need to draw a minimum amount of power continuously to power the accelerometer and microchip. Minimizing the energy consumed by the fall detection biosensor
will ensure our goal of creating an efficient and environmentally responsible vital signs monitor. In order to meet these design requirements, the fall detection biosensor will use the ADXL345 triple axis accelerometer made by Sparkfun Electronics (Boulder, CO) seen in Figure 6.

![Sparkfun Electronics ADXL345 Triple-Axis Accelerometer](image)

**Figure 16:** Sparkfun Electronics ADXL345 Triple-Axis Accelerometer

This product is more attractive than our previous selection, the ADXL335, because it has a higher full sensing range, it is accessible through SPI digital interface, and it has a higher resolution. Having a resolution of 4mg/LSB allows the ADXL345 to measure inclination changes of less than 1.0 degrees. This allows the fall detection biosensor to sense when the patient has tilted past the point in which it is capable to maintain stability, indicating a fall. This knowledge could also lead to future developments in which some sort of precaution is activated after the tilt has reached the threshold but before the patient has completed their fall. Currently, it will alert caretakers as quick as possible by reporting an emergency as the fall is occurring. The ADXL345 measures up to 16g and outputs the data digitally formatted as a 16-bit two's complement. This generation includes the ability to measure tilt-sensing applications, dynamic acceleration resulting from motion or shock, and inclination changes. This information will upload into software on the Microsoft Surface which will determine whether the patient is upright, falling, or has fallen and is not getting back up. Having this ability allows us to possibly alert the patient with an auditory message or command after experiencing a fall and refraining to send a false alarm emergency contact until a determined time has passed without a response by the patient manually or by the accelerometer’s detection of a lack of movement. Additional capabilities include activity and inactivity sensing to detect presence or lack of motion, the ability to detect if the acceleration on any axis exceeds a user-set level, free-fall sensing, tap sensing, a 32-level FIFO buffer which can be used to store data in order to minimize host processor intervention, and low power modes which enable intelligent motion-based power management with threshold sensing and active acceleration measurements at extremely low power dissipation. The tap sensing ability could be incorporated as the manual response from the patient after a fall. For example, the auditory message would command the patient to tap
once to contact emergency caretakers or tap twice to indicate they have not suffered life threatening injuries.

One limitation includes the need for a lower voltage power supply to the accelerometer. If we assume the power line of the chest strap has a voltage of 5 V, there will need to be a voltage drop in the circuit to accommodate the accelerometer’s desired voltage range of 1.8-3.6VDC for optimal performance. Also, based on the previous design of the vital signs monitor, only one output of data from a biosensor can be received at a time. In order to have the fall detection biosensor constantly monitoring without using the data transmission line, the output of the biosensor must only transmit data in the case of a fall, rather than continuously occupying the line.

In order to calibrate the ADXL345 accelerometer, it is necessary to remove any biases from the readings. The following code will ensure the base values for the x and y axes are approximately zero and the z axis reads approximately 1g:

```c
void calibrateAccelerometer(){
    //Take a number of readings and average them
    //to calculate any bias the accelerometer may have.
    for (int i = 0; i < 32; i++) {
        accelerometer.getOutput(readings);
        a_x += (int16_t) readings[0];
        a_y += (int16_t) readings[1];
        a_z += (int16_t) readings[2];
    }
    //50Hz data rate.
    delay(20); // Arduino does things in milliseconds

    a_x /= 128;
    a_y /= 128;
    a_z /= 128;

    //At 4mg/LSB, 250 LSBs is 1g.
    a_xBias = a_x;
    a_yBias = a_y;
    a_zBias = (a_z - 250);

    a_x = 0;
    a_y = 0;
    a_z = 0;
}
```

**Figure 17: ADXL345 Calibration Program**

During the calibration, the accelerometer must be on a stable surface and oriented in the resting position of how it will be placed on the chest strap. One discrepancy we have found for this product is the datasheet says the sensitivity is 4 mg/LSB, however the specifications says it is 3.9mg/LSB. If we are doing a proper calibration that produces significant error due to the 0.1 mg/LSB discrepancy then we would have to change the z axis bias. The z-axis bias would become 256 as opposed to 250.

The fall detection biosensor does not necessarily need a PCB board to function. It is on a breakout board which allows us to connect the ADXL345 accelerometer to the Arduino master controller using a protoboard. All we need to begin testing the fall detection biosensor are the pins which will secure the ADXL345 breakout board into the Arduino protoboard.
The ADXL345 has 8 pins. Ground is wired to the ground port and Vcc is wired to the voltage supply. The chip select (CS), serial data input (SDI), serial data output (SDO), serial port clock (SCLK) are wired dependent on the serial communication used. The serial communication options are I2C and SPI (3- and 4- wire configurations). INT 1 and INT2 are wired to the analog input and are expressed as high when the specific function programmed to each port occurs.

This biosensor is connected directly to the Arduino master controller therefore it is not required to communicate data through I2C because of its direct connection. There are no other sensors that share the line of communication, therefore SPI may be the better option for serial communication. The major benefits of SPI are its speed in transferring data and its full duplex operation when wired as a 4-wire configuration. In this application, most of the disadvantages of using SPI as a serial communication do not apply therefore I believe this is the best method.

Figure 19 shows the connection diagram needed in order to wire the ADXL345 to the Arduino master controller to communicate through SPI.

Figure 3 shows the current code, excluding programmed interrupts which will provide alarm functionality.

//Add the SPI library so we can communicate with the ADXL345 sensor
#include <SPI.h>

//Assign the Chip Select signal to pin 10.
int CS=10;

//This is a list of some of the registers available on the ADXL345.
//To learn more about these and the rest of the registers on the ADXL345, read the datasheet!
char POWER_CTL = 0x2D; //Power Control Register
char DATA_FORMAT = 0x31;
char DATAX0 = 0x32;  //X-Axis Data 0
char DATAX1 = 0x33;  //X-Axis Data 1
char DATAY0 = 0x34;  //Y-Axis Data 0
char DATAY1 = 0x35;  //Y-Axis Data 1
char DATAZ0 = 0x36;  //Z-Axis Data 0
char DATAZ1 = 0x37;  //Z-Axis Data 1

//This buffer will hold values read from the ADXL345 registers.
char values[10];
//These variables will be used to hold the x,y and z axis accelerometer values.
int16_t x,y,z;

void setup(){
  //Initiate an SPI communication instance.
  SPI.begin();
  //Configure the SPI connection for the ADXL345.
  SPI.setDataMode(SPI_MODE3);
  //Create a serial connection to display the data on the terminal.
  Serial.begin(115200);

  //Set up the Chip Select pin to be an output from the Arduino.
  pinMode(CS, OUTPUT);
  //Before communication starts, the Chip Select pin needs to be set high.
  digitalWrite(CS, HIGH);

  //Put the ADXL345 into +/- 4G range by writing the value 0x01 to the DATA_FORMAT register.
  writeRegister(DATA_FORMAT, 0x01);
  //Put the ADXL345 into Measurement Mode by writing 0x08 to the POWER_CTL register.
  writeRegister(POWER_CTL, 0x08);  //Measurement mode
}

void loop(){
  //Reading 6 bytes of data starting at register DATAX0 will retrieve the x,y and z acceleration values from the
  //The results of the read operation will get stored to the values[] buffer.
  readRegister(DATAX0, 6, values);

  //The ADXL345 gives 10-bit acceleration values, but they are stored as bytes (8-bits). To get the full value, two
  //The X value is stored in values[0] and values[1].
  x = ((int)values[1]<<8)|(int)values[0];
  //The Y value is stored in values[2] and values[3].
  y = ((int)values[3]<<8)|(int)values[2];
  //The Z value is stored in values[4] and values[5].
  z = ((int)values[5]<<8)|(int)values[4];

  //Print the results to the terminal.
  Serial.print(x, DEC);
  Serial.print(',');
  Serial.print(y, DEC);
  Serial.print(',');
  Serial.print(z, DEC);
  Serial.println();
}
void writeRegister(char registerAddress, char value){
    //Set Chip Select pin low to signal the beginning of an SPI packet.
    digitalWrite(CS, LOW);
    //Transfer the register address over SPI.
    SPI.transfer(registerAddress);
    //Transfer the desired register value over SPI.
    SPI.transfer(value);
    //Set the Chip Select pin high to signal the end of an SPI packet.
    digitalWrite(CS, HIGH);
}

void readRegister(char registerAddress, int numBytes, char * values){
    //Since we're performing a read operation, the most significant bit of
    //the register address should be set.
    char address = 0x80 | registerAddress;
    //If we're doing a multi-byte read, bit 6 needs to be set as well.
    if(numBytes > 1)address = address | 0x40;
    //Set the Chip select pin low to start an SPI packet.
    digitalWrite(CS, LOW);
    //Transfer the starting register address that needs to be read.
    SPI.transfer(address);
    //Continue to read registers until we've read the number specified, storing the results to the input buffer.
    for(int i=0; i<numBytes; i++){
        values[i] = SPI.transfer(0x00);
    }
    //Set the Chips Select pin high to end the SPI packet.
    digitalWrite(CS, HIGH);
}

Figure 20: Arduino Due Program for ADXL345 Breakout Board

\[
\text{Scale} = \frac{8}{2^{10}} = \frac{8}{1024} = 0.0078
\]

The above equation shows the scale needed to convert the ADXL345 output acceleration values to export data in g’s. This scale is for the +/- 4g mode on the ADXL345. I chose this mode because for our application +/- 4gs is more than enough acceleration detection sensitivity than we need.
3.2.2.6 Stethoscope Biosensor

The stethoscope biosensor will use the electronic sensor developed by Chad Lyons and have a headphone jack for local output. There will be no digitizing signal, it will just calculate the respiration rate on the sensor itself, and then send that information when requested. The stethoscope can be used on both sides of the body to listen to different areas, as physicians are able to detect different sounds from the front of the patient and their back. There are some drawbacks using a stethoscope including frequencies too low for humans to hear, artifacts being introduced through movement of the sensor, and in Chad’s design the adhesive dries out so it will need to be reapplied every few hours. Once the data is transmitted to LabVIEW, the user has the option to play back the sounds transmitted from the device and in the programming this data will be used to calculate respiration rate as mentioned in section 3.2.2.8.

```c
#include <Wire.h>
int analogPin1 = 0;
int incomingAudio = 0;
int deviceID = 3; // I2C address

void setup()
{
  Wire.begin(deviceID);
  // turn on power to the thermistor using a switch
  Wire.onRequest(requestEvent);
  Serial.begin(115200);             // Setup serial
  //digitalWrite(13, HIGH);         // Indicates that the program has intialized by turning on the LED
}

void loop()
{
  //incomingAudio = analogRead(A0);//read input from A0
  //do stuff with the variable "incomingAudio"
}

void requestEvent()
{
  int Audio = takeAudio();
  Wire.write(incomingAudio);// Write the voltage value to the master
}
int takeAudio()
{
  int incomingAudio = analogRead(analogPin1);//read input from A0
  return incomingAudio;
}
```
Figure 24: Front panel of Stethoscope Sensor

Figure 25: Block diagram of Stethoscope sensor requesting the user to input how long of a sample
3.2.2.7 ECG sensor

The ECG sensor will utilize either 12 lead configuration to increase accuracy by providing sufficient references to detect the electric potential of the heart across the chest. The leads will plug into an integrated analog front end of the Texas Instruments ADS1298R ECG Front End Performance kit using a DB15 connector. The ADS1298R chip packs 44 discrete which reduces the power consumption by 95% compared to a fully discrete design and the component count. The chip is available in eight-, six-, and four- channel versions which all provide 20 milliohm of resolution for respiration impedance. The chip consists of 8 analog-to-digital converters, 8 programmable gain amplifiers, and 8 active filters. This front end performance kit is beneficial to our design based on the low power consumption and its capabilities which include a reliable ECG signal, integrated respiration impedance meter, and an analog front end. The following figure on the next page shows the block diagram overview of the ADS1298R chip and its hardware capabilities. Once transmitted to the Texas Instruments LabVIEW based program, the ECG will be displayed in real-time on a graph on the front panel as shown in Figure 1, and be used to calculate the heart rate and respiration rate of the patient.
The TI ECG Front End Performance kit has the ability to measure the patient’s respiration rate based on the principle of impedance pneumography. This is an internal function of the ADS1298R chip which uses a square wave for modulation. The modulation frequency used is 32 kHz and the on-chip reference is used as the modulating signal. Demodulation is done with a blocking scheme. The following figure shows the respiration portion of the circuit:
To obtain a baseline impedance ($R_B$) and the varying component ($\Delta R$), a patient simulator can be connected between leads ELEC_RA and ELEC_LA. Resistors R96 and R97 limit the amount of ac current that flows into the body. Capacitors C108 and C109 block any dc current from flowing into the body from the transmission side. Capacitors C99 and C100 serve the same purpose on the receiver side. Capacitors C113 and C114 serve as a secondary means to prevent a single fault from causing excessive dc currents through the patient. The respiration signals are routed to Channel 1 which has respiration capability. The expected dc output can be calculated using the equation:

$$DC\_V = \frac{R_B}{R_B + R96 + R97} \cdot (VREFP - VREFM)$$

$$= \frac{0.5 \text{ k}}{0.5 \text{ k} + 40 \text{ k} + 40 \text{ k}} \cdot 2.4 = 14.9 \text{ mV}$$

(Eq. 1)
The current flowing through the body is calculated using the equation:

\[
I_B = \frac{V_{REFP} - V_{REFM}}{R96 + R97 + R_B} = \frac{2.4}{80.5 \text{ k}} = 29.81 \mu A
\]  
(Eq. 2)

The peak-to-peak output can be calculated using the equation:

\[
\Delta V = \Delta R \cdot I_B = 1 \cdot 29.15 = 29.1 \mu V \quad (\Delta R = 1 \Omega)
\]

\[
\Delta V = \Delta R \cdot I_B = 0.1 \cdot 29.15 = 2.91 \mu V \quad (\Delta R = 0.1 \Omega)
\]  
(Eq. 3)

The following figure shows two examples of output graphs. These were done by Texas Instruments using a Fluke MedSim 300B simulator. Our models will look similar, however the programming and output graphs will be done using LabVIEW.

![Figure 23: Respiration Impedance Pneumography Output Models](image)

3.2.2.9 Weight sensor

As mentioned in the introduction, patient’s weight will be measured through an external Bluetooth enabled scale that is separate from the main Vital Sign Monitor device. The scale will still be in contact with the base station and have the data transmitted to the Microsoft Surface for analysis and display through LabVIEW. Each set of data taken from the scale will be paired with the date and sent to a separate file so the patient can monitor their weight over time.
3.2.2.10 Blood pressure cuff

Blood pressure contains 2 sets of data, diastolic and systolic (or minimum and maximum). This vital sign must be collected by constricting vessels in order to determine these minimum and maximum points, therefore a blood pressure cuff around the arm will be necessary to monitor this data. Since this cuff is around the arm, and away from the main device, it will need its own Bluetooth transmitter which Bluetooth enabled cuffs already exist in the market. The data sent from this cuff will be broken into the systolic and diastolic measurements needed for display to the patient.

3.2.2.11 Blood Glucose
3.2.2.12 LabVIEW Programming

As mentioned above in the introduction, LabVIEW is an important aspect to this project. The block diagram, as shown in Fig 8, shows a majority of the programming behind the front panel as shown in Figure 1.
3. Realistic Constraints and Standards

Engineering standards enable devices for universal use by any individual and enable different manufacturers to create products that work together. Standards for the vital sign monitor include: power supply in the form batteries; biosensor devices adhering to existing standards in the medical field and IEEE standards; thin film temperature sensor adhering to US, German, and British standards which displays Celsius and Fahrenheit; Withings scale displaying weight in kilograms, pounds, and stone-pounds; blood-pressure cuff displaying diastolic and systolic measurements; pulse rate in beats per minute. These standardized measurement systems and components will be implemented in the design of this vital sign monitor.

One of the most important constraints for this project is power consumption. Most of the sensors are relatively low in their power consumption, and the I²C protocol has some built in ways to trigger a “power save” mode for sensors when they are not collecting data. However, for the temperature sensor, our current design accidentally has a current bridge between the power source and the ground. This will lead to a constant leakage of battery power. We will include the correct Ultiboard layouts when we turn the project in. During the current time however, it would be prudent to remove the temperature sensor whenever it is not in use.

In order to continuously monitor the patient to detect a fall at any point, the fall detection biosensor will need to draw a minimum amount of power continuously in order to power the accelerometer and microchip. Minimizing the energy consumed by the fall detection biosensor will ensure our goal of creating an efficient and environmentally responsible vital signs monitor. Based on the previous design of the vital signs monitor, only one output of data from a biosensor can be received at a time. In order to have the fall detection biosensor constantly monitoring without using the data transmission line, there will be an additional wire so that the biosensor is directly connected so as to not continuously utilize the data transmission line. The Sparkfun...
Electronic’s ADXL345 Triple Axis Accelerometer needs the power supply to exist in the range of 1.8-3.6 VDC which creates the need to make a voltage drop from the chest strap Vcc power line if it is above the given voltage range. The ADXL345 accelerometer has a power management system which will only consume power when a user-set acceleration or degree of inclination occurs. This will ensure the least amount of power consumption as possible.

The temperature biosensor includes a Negative Temperature Coefficient (NTC) thermistor which requires several constraints in order to perform accurately and efficiently. Voltage above 100Vdc will destroy the NTC thermistor. The NTC thermistor element must be separated from the other circuit elements which may increase in temperature to avoid self-heating error and noise. The minimum and maximum temperature the NTC thermistor can measure are 0°C (32°F) and 70°C (158°F) respectively. The NTC thermistor must also avoid internal heating due to excess power from a high lead current. If extra heat cannot be dissipated, heating caused by excitation current can raise the temperature of the sensing element above the ambient temperature causing error due to a change in resistance of the NTC thermistor.

The ECG biosensor will consist of 12 chest leads which plug into the analog input for the Texas Instruments ADS1298R ECG Front End Performance kit. The chest leads must be made out of non-allergenic materials and may require electrode gel to ensure proper conduction. The ADS1298R ECG chip has several realistic constraints. The operating temperature range is -40 degrees Celsius to 85 degrees Celsius. The momentary input current is 100 mA and the continuous input current is 10 mA. The digital input and output voltages range from -0.3 V to 0.3 V. The ECG chip resolution is 17 bits with the maximum 32kSPS data rate, 19 bits with a 16kSPS data rate, and 24 bits with data rates up to 8kSPS. A USB to micro-USB cable is required to connect the TI ADS1298R ECG Front End Performance kit to the Microsoft Surface tablet.

A severe limitation on stethoscopes is that they yield poor analysis even by a trained professional, nevertheless they can be used to calculate respiration rate as desired in the vital sign monitor. Stethoscopes come in a variety of different products, from classical stethoscopes used by a physician to electronic stethoscopes converting physical audio signals to electronic signals for amplification and processing. Classical stethoscopes can’t be used due to an electrical signal and can’t detect subtle heart/lung irregularities while electronic stethoscopes generate electrical signals and detect many more defects but can be quite costly as the technology put into them are more advanced than other devices and are still bulky. Prices range from $125-600 for electronic stethoscopes, possibly lower. To get around these constraints, we found a stethoscope device that generates an electrical signal from sound, is small enough for patient use, excellent irregularity detection, and cost effective. This Stethoscope Receptacle, patented by Chad Lyons, will be used in this device with limitations on having to replace the device after use, and hydrogel adhesive being used which has to be applied for the device to attach to the patient’s skin.

An often seen constraint in patient weight is what type of measurement system they are using, US or SI. The scale that we implemented in this design provides display in both formats. Further constraints to scales, are weight limits as the Withings scale can load up to 300 pounds, and type of surface it can rest on, which the device comes with carpet feet attachments.
4. Safety Issues

Safety is always a primary concern when dealing with patients as they are already in a less than normal state and with our monitor, it will hopefully be used by a numerous amount of patients with varying states of health and/or disease ranging from cancer, heart disease, infection, trauma, and other medical uses. No device is perfectly safe as people manage to harm themselves accidentally in strange ways and we must account for these odd variables. We will try to minimize hazardous design as much as possible.

Devices that contain electrical elements always will pose a safety issue. However a low current of around 40 mA should prevent any damage to wearers. Each sensor must have an adequate power source (through the single wire) and contact with skin, so one safety issue is the contact with skin posing an electrocution problem and can be minimized by insulating material surrounding contacts (besides the skin) and the voltage leading to the sensors. Infrared radiation from the pulse oximeter is also a possibility which may have a harmful effect and will be noted to the patient. Coinciding with electricity and radiation, these elements can heat up therefore material that dissipates heat is desired to not burn patient’s skin.

Since each sensor will be removable via clip, there can be potentially small parts that can be hazardous to infants. There will be a warning label and possibly a case that can house each sensor is possible. Another mechanical issue is the clips in itself they can pinch and perhaps pierce patient’s skin.

Materials are important in design as they are in direct contact with skin and clothing. Safety hazards with the materials are scratching, skin irritation, allergic reaction, and immune response among others. Our chest strap is made out of a t-shirt to avoid all of these potential hazards as well as soft conductive fabric to propagate the electrical signal. The housings were designed with slightly rounded corners to prevent poking and potential for injury.

5. Impact of Engineering Solutions

This vital signs monitor will have the opportunity to become a solution to many global, economic, environmental, and societal problems which are prevalent today by providing an inexpensive way to monitor the heart using an ECG, respiration, blood pressure, body temperature, blood oxygen level, weight, and detect falls in the case of an emergency. The goal of this project is to make an efficient vital signs monitor that emphasizes on accuracy and economic accessibility. As health care remains a significant issue here in the United States and the rest of the world, an inexpensive vital signs monitor will enhance each individual’s ability to understand their health, improve their well-being, and collaborate with their caretakers through the quick and easy transmission of vital data in real time. Patients suffering from numerous different conditions will attain the ability to properly monitor themselves outside of a hospital or doctor’s office while simultaneously providing their caretakers with data to progress their health over time and in several different circumstances of daily life.

Providing an inexpensive vital signs monitor allows many members of society which struggle economically to assess their health at home and eliminates many unnecessary doctor or emergency room visits which can become extremely costly. This product improves the process of diagnosing conditions, doctor evaluations, self-evaluations, and examining health progress over time. With the ability to transfer information wirelessly from the biosensors on the patient’s
body to LabVIEW on the Microsoft Surface for processing to any internet platform for the viewing of the patient, their family, their doctors, and other caretakers the time it takes to diagnose and assess a problem will be less than the patient’s time spent in the waiting room.

This project is capable of having a global effect due to its compatibility and portability. The Microsoft Surface will be available worldwide following its release date allowing our vital signs monitor to be relevant all over the globe. With the global economy suffering, revolutionizing basic vital signs monitoring with our relatively inexpensive device can result in many people being provided with proper health care whom normally could not receive the proper attention.

Our efficient vital signs monitor design stresses environmental responsibility through the limited use of materials, elimination of waste by making each biosensor capable of clipping on and off the base chest strap, and limited power usage due to our effective circuit designs in order to use the absolute minimum amount of energy possible. Making each biosensor individual with the clip-on system is an advantage for consumers and environmentalists as it reduces the waste in the event of a sensor error or malfunction. If one biosensor malfunctions, instead of needing to replace the entire chest strap, the patient need only replace the specific biosensor which no longer works accurately.

6. Life-long Learning

This device will take advantage of a simple and modular design that will allow future development to continue to add to the existing project with a simple hardware interface. This will provide an easy template and route of thinking that makes it easy for future engineers to design around our work without it presenting a major obstacle in functionality of the device. This is an important skill to learn because we feel that starting to work on a project with existing progress is frustrating because it requires new engineers with different ideas and experiences to conform to existing limitations and blind spots. While we don't presume that our design doesn't have such problems, we feel that the open ended format we have presented allows for the greatest amount of expansion, considering that beyond offering a platform that provides power, and a means of transmitting data, there are no restrictions or adjustments that need to be made to existing sensors to make new sensors work alongside future developments, indeed the way we plan to implement our one wire communication array is built on the assumption that there may need to be a variable amount of sensors and future devices attached to the same product.

Currently, we are all learning about the internal design of medical devices, as we continue to design these components, we will move from modeling to making the actual device, when we will run into a better idea of the software to hardware constraints. Since we intend to make the sensors out of the smallest microchips we can program to keep them light, we might run into memory issues that are currently unforeseen.
7 Budget and Timeline

7.1 Budget

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Withings WS-30 Bluetooth Scale</td>
<td>$121.99</td>
</tr>
<tr>
<td>FORA G31b Bluetooth Blood Glucose Meter</td>
<td>$125.00</td>
</tr>
<tr>
<td>iHealth Bluetooth Blood Pressure Cuff</td>
<td>$83.00</td>
</tr>
<tr>
<td>Withings Bluetooth Blood Pressure Cuff</td>
<td>$129.00</td>
</tr>
<tr>
<td>Temperature Sensor</td>
<td>$100.00</td>
</tr>
<tr>
<td>Blood Oxygen Sensor</td>
<td>$38.00</td>
</tr>
<tr>
<td>Stethoscope Sensor</td>
<td>$45.00</td>
</tr>
<tr>
<td>Fall Detection Sensor</td>
<td>$27.95</td>
</tr>
<tr>
<td>TI ECG Chip</td>
<td>$339.00</td>
</tr>
<tr>
<td>Mastercontroller</td>
<td>$56.44</td>
</tr>
<tr>
<td>Microsoft Surface</td>
<td>$1,270.59</td>
</tr>
</tbody>
</table>

Total Expenses: $2,335.97
Total Expenses without Surface: $1,065.38
Total Expenses without standalone: $1,876.98
Total Expenses with neither: $606.39

7.2 Timeline

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>51 days</td>
<td>Mon 1/21/13</td>
<td>Mon 4/1/13</td>
</tr>
<tr>
<td>Parts Ordering</td>
<td>39 days</td>
<td>Wed 11/28/12</td>
<td>Mon 1/21/13</td>
</tr>
<tr>
<td>LabVIEW Programming</td>
<td>54 days</td>
<td>Wed 11/28/12</td>
<td>Mon 2/11/13</td>
</tr>
<tr>
<td>Arduino Master Controller</td>
<td>54 days</td>
<td>Wed 11/28/12</td>
<td>Mon 2/11/13</td>
</tr>
<tr>
<td>Arduino Bluetooth Programming</td>
<td>6 days</td>
<td>Mon 2/11/13</td>
<td>Mon 2/18/13</td>
</tr>
</tbody>
</table>
8 Team Members Contributions to the Project

8.1 Team Members

8.1.1 David Knoff

David Knoff’s contribution to the vital signs monitor focused on the body temperature biosensor, fall detection biosensor, and ECG biosensor. He designed the body temperature biosensor 4-wire NTC thermistor circuit and wrote the subvi LabVIEW program to process the body temperature in Fahrenheit from the voltage drop across the NTC thermistor surface probe. He selected the Sparkfun Electronics ADXL345 triple-axis accelerometer for the fall detection...
biosensor and wrote both the program for SPI communication with the master controller and the calibration program. He also designed the alarm feedback flow chart to detect a lack of balance and fall. For the ECG biosensor he researched lead and electrode designs to fit the group’s design specifications while also providing the most accurate ECG waveform. He chose a 12 lead configuration to ensure an the most accurate ECG waveform due to sufficient references to detect electric potential across the chest. He researched the Texas Instruments ADS1298R ECG Front End Performance kit which will make the biosensor design more efficient by reducing power consumption and board size by 95% each. Upon arrival, David installed the Texas Instruments LabVIEW-based software including the drivers for the connection between the motherboard and the software program. David was also responsible for selecting and ordering the 12 lead configuration ECG cables, electrode pads, and snap clip-banana clip adapters. David programmed the housings in Solidworks and collaborated with Thomas Mealy and Solomon for 3D printing. The overall concept and design of the vital signs monitor including the chest strap, each biosensor, and programming was a result of constant collaboration between each member of the group to ensure compatibility, efficiency, and quality.

8.1.2 Maysarah Shahabuddin

Maysarah Shahabuddin’s contributions to this project have revolved around the design of the chest strap, and subsequently the sensor programming. His job began with the design of the current system, from the use of the I2C interface to the chest strap to the decision to make a modular device that “cooperates” to all use one battery/Bluetooth antenna. From there he directed his teammates and divided each of the sensors between his group members. Once it was decided that the scale, blood glucose, ECG, and the blood pressure sensor would be adopted from existing Bluetooth enabled sensors, the rest of the vital signs were going to be mounted on the chest strap. His idea to connect all of the sensors via the I2C bus seemed to be the best compromise of economy and function. Since I2C was by and large Maysarah’s idea, creating the sensors on the strap became his priority. From there, the programming of any sensor that would use I2C via the wires on the strap fell under his jurisdiction. Then, in collaboration with David, who did the research on the hardware for the temperature sensor, they designed the temperature sensor from the bottom up, and Maysarah made an I2C compatible program that would output the temperature. He then designed and made the pulse oximeter himself. This pulse oximeter when done could be able to work standalone as a blood oxygen content monitor if it were provided with the power and Bluetooth antenna. After meeting with Chad Lyons, and acquiring the stethoscope, he also wrote the code to collect sound from the device. In collaboration with Jake, he worked on Bluetooth communication between the computer and the strap. With David, he drew mockups and designs for the cases, which David made and ordered using Solidworks. Other minor work contributed includes sewing the chest strap together, and soldering the PCBs, including surface mount components.

8.1.3 Jacob Adams

Jacob Adams’ contributions to this project have focused primarily on the LabVIEW programming interface for the physician or at home user. His primary objective was to design a way for LabVIEW to acquire the information, extract the different sets of data for each sensor, and distribute each signal to be processed and displayed in an effective manner. Along with the LabVIEW coding, Jacob was in primary charge of the scale, stethoscope, Microsoft Surface, blood pressure cuff, and website while he contributed to the aid of Maysarah with the coding of
the microcontrollers in C++ programming language as well as the pulse oximeter. Up until now, Jacob has completed the extraction and distribution steps of the LabVIEW but still has to display the information in a concise manner that can be easily read by a patient with no medical knowledge whatsoever. Further work will be in testing how each of his main devices communicates with the tablet. Further, Jake has been the main liaison between the budget office and has been the one on the team to fill out order forms. Throughout the semester, he has maintained an itemized budget, as well as kept the website up to date.

9 Conclusion

This vital signs monitor will provide an inexpensive way to monitor blood pressure, blood glucose level, heart rate, respiration rate, body temperature, blood oxygen level, heart and lung sounds through a stethoscope, weight and detect falls in the case of an emergency. With the majority of our biosensors efficiently designed to fit in a strap across the patient’s chest, this device is portable and has the ability to monitor vital signs comfortably in a variety of settings including during exercise, daily tasks, sleep, and many other activities. In the design of our vital signs monitor, we stress simplicity to accommodate the average consumer with little or no clinical experience and possible disabilities. The patient’s vital signs will be wirelessly transmitted via Bluetooth to a tablet, such as the Microsoft Surface, through a remote receiving and storing device. From the tablet, the patient’s vital signs can be transferred over the internet in order to be accessed by themselves and their doctors on several different platforms. Our vital signs monitor provides a unique, inexpensive and effective way for consumers to monitor themselves during daily activities and efficiently relay relevant medical information to their doctors.

Our design platform provides the opportunity for future developments through the addition of biosensors with relative ease. New, state of the art biosensors can be implemented into our vital signs monitor through simple design and programming at any point in the product’s lifetime.
10 References


Sparkfun Parts used for images
https://www.sparkfun.com/products/9768
https://www.sparkfun.com/products/9269

Microsoft Surface
www.microsoftsurface.com/surface

Withings Bluetooth scale
11 Acknowledgements

Tom Capuano
Tom Capuano has done work in professor Enderle’s lab making a Bluetooth EKG and temperature sensor that works with LabVIEW. Due to his prior success, we have used his knowledge, and with his permission, his VI for much of the inspiration behind our LabVIEW code. His suggestions for various hardware upgrades, and existing issues in the project has also shaped the direction we have gone in regards to many design decisions.

Chad Lyons
Chad Lyons has worked with Professor John Enderle in the past to develop a small, stethoscope patch. This patch has piqued our interest as we would like to use it as our stethoscope sensor.

Dr. Faqir Jain
Dr. Jain, of the electrical engineering faculty in UConn has developed a blood glucose sensor that can be implanted under the skin. His insight and experience on blood glucose detection proved useful when attempting to include this sensor into our project. Since no current sensors exist to fit our preferences, we had to abandon this venue. However, his advice and resources will not go unacknowledged.

Dave Kaputa
His work and aid with the Vital Signs Monitor, past and present has been incredible, for we’ve shared many hours in the lab while he watches our frustration, only to have his quiet suggestions provide the direction we’ve needed.

Thomas Mealy for his help with 3-D printing of the sensor housings

Sarah Brittain
Our TA for this class has been a guardian angel of sorts, offering advice, suggestions, caveats, and most importantly a buffer for tense emotions throughout the semester. Thank you for looking out for us.

Dr. John Enderle
Dr. Gielo-Perczak