

Untethered Bioinstrumentation Card



A Major Qualifying Project
Submitted to the Faculty of
Worcester Polytechnic Institute

Submitted By:

Samantha Fontaine:

Renzhi Jiang:

Submitted to:

Project Advisor:

John McNeill:

April 26, 2012

Table of Contents

Table of Figures.....	3
Previous Work.....	5
ECG circuit	5
PPG circuit	6
Research.....	7
Survey.....	7
Analog Device’s ICs	10
ECG Chip – ADAS 100	10
Analog Microcontroller – AduC70xx	11
Instrumentation Amplifier – AD8235.....	12
Biometric Signals.....	13
Heart rate and pulse	13
Skin temperature	18
Blood Pressure	19
Blood Alcohol Content.....	21
Hand Strength.....	21
Power	22
Rechargeable Batteries.....	22
Voltage regulators	24
Microcontrollers	27
USB Interface	30
User Interface	37
LCD Display.....	37
LED Warning System.....	40
Final Project Decision	40
Power	40
Size	41
Price.....	41

Biometric Signals.....	41
Communication to User.....	42
Block Diagram	42
Power	43
Digital	43
Analog/Signal Processing	44
User Interface	44
Detailed Design	44
Hardware.....	44
Circuits Design.....	45
Breadboard Design	52
PCB Design	53
Software.....	55
Microcontroller.....	55
LCD.....	63
Noise Analysis	64
Hardware.....	65
Software.....	66
.....	66
Conclusion.....	67
Bibliographies	68
Appendix A – Simulation results	69
Appendix B - Code.....	71

Table of Figures

Figure 1 Last Year's MQP ECG Model [1]	5
Figure 2 Last Year's MQP Full ECG Schematic [1]	6
Figure 3 Last Year's MQP Top Level PPG Block Diagram [1].....	6
Figure 4 Last Year's MQP LED Driving Circuit [1]	7
Figure 5 Survey Results 1.....	8
Figure 6 Survey Results 2.....	8
Figure 7 Survey Results 3.....	9
Figure 8 Survey Results 4.....	9
Figure 9 Survey Results 5.....	9
Figure 10 Typical ECG Signal [3]	14
Figure 11 Wavelength vs. Absorption for Hb and HbO2	16
Figure 12 Different formula applied for relationship between R and SaO2.....	17
Figure 13 Different portions of the origin signal	17
Figure 14 Different signals monitored to see PVC.....	18
Figure 15 Classification of BP.....	20
Figure 16 Graph of mass and volume energy density of several batteries	24
Figure 17 UART Communication between PC and Microcontroller.....	31
Figure 18 Data bit transition in UART	32
Figure 19 Pin configuration	34
Figure 20 ADI USB isolator configuration.....	35
Figure 21 Example for the ADI USB isolator.....	36
Figure 22 USB interface FT232R example	36
Figure 23 LCD with CCFL backlight Figure 24 LCD with LED matrix backlight	39
Figure 25 Block diagram	43
Figure 26 ADP2291.....	51
Figure 27 ADP1613.....	51
Figure 28 ADP123.....	52
Figure 29 First version of PCB	55
Figure 30 Microcontroller schematics.....	59
Figure 31 User's heart rate from the output of the PPG PCB.....	65
Figure 32 Another screenshot of the user's heart rate	66
Figure 33 DAC output which is used to control LED intensities	66
Figure 34 ECG Signal of each pulse of the PQRST Complex.....	70
Figure 35 ECG Signal SPICE file	71

Introduction

The purpose of this project is to demonstrate to prospective students what the WPI Electrical and Computer Engineering Department offers, which is education and technical training that would enable students to successfully complete design work in the Electrical Engineering field. Furthermore, our advisor proposed an interactive handheld, low-cost device that measures bioinstrumentation signals. The device would have multiple purposes: project demonstration to prospective students, body signal monitor for athletes and people who exercise and a health monitor when people are on the go.

The goal for this design was to measure a biometric signal and communicate the results to the user. This project goal was inspired on the previous two MQPs, which developed a design to demonstrate the capabilities of an integrated chip designed and manufactured by Analog Device Inc. The previous two MQPs motivated us to design a device that would encompass both aspects of their projects: ECG and PPG. The previous MQP device was solar-powered and utilized ECG to determine the user's heart rate while the year before that was battery powered and used PPG.

Our project does not have the same power constraints as last year's project. While the previous design was solar powered our project will be powered by a rechargeable battery. Although the power constraint isn't as fine as the previous years, we still need to take in power consideration since we are utilizing two biometric signal circuits. Our design may occupy the size of two standard business cards together (3.5 by 4 inches) up to smartphone size (4.5 by 3.5 inches). A cost goal of \$30 per board was set, which allows the

Electrical and Engineering Department to have a few boards easily accessible in most laboratories.

Background

Previous Work

ECG circuit

Last year's MQP expanded on a previous MQP with the objective of developing a low-cost, handheld bioinstrumentation demonstration board while meeting significantly stricter power, size and cost constraints than the previous project. Specifically, the device must operate using only power provided by a solar panel in indoor lighting conditions and be contained on a board the size of a standard business card. Their proposed solution measured electrocardiogram (ECG) signals taken from the fingertips and displayed the user's heart rate on a liquid crystal display panel. They designed and built a functional prototype using their method, meeting the power and size requirements.

They modeled the expected ECG signal, including their predicted noises, which can be seen below along with their complete ECG schematic. During our design process we tried to keep their ideas in mind.

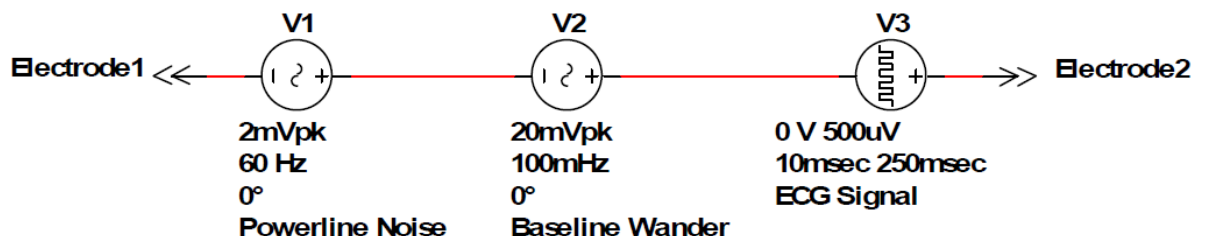


Figure 1 Last Year's MQP ECG Model [1]

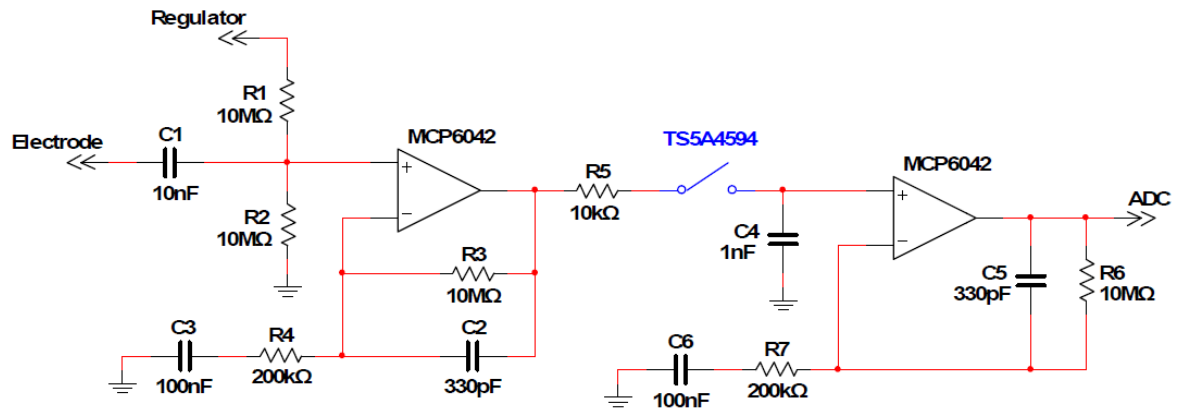


Figure 2 Last Year's MQP Full ECG Schematic [1]

PPG circuit

Last year's MQP also tried to incorporate PPG signal analysis. They were not successful in implementing both designs since they ran out of time and power. The two figures below shows their top level block diagram and their LED driving circuit. When designing our PPG circuit we kept in mind their ideas.

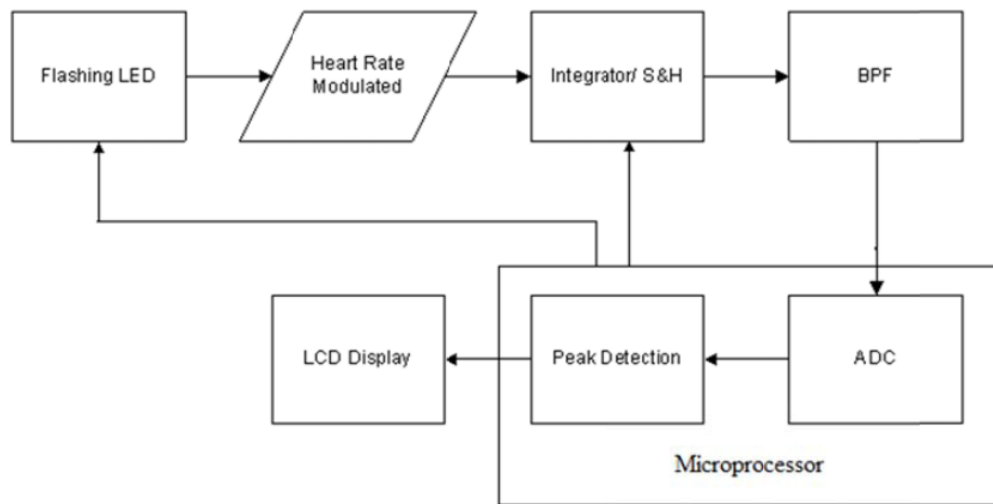


Figure 3 Last Year's MQP Top Level PPG Block Diagram [1]

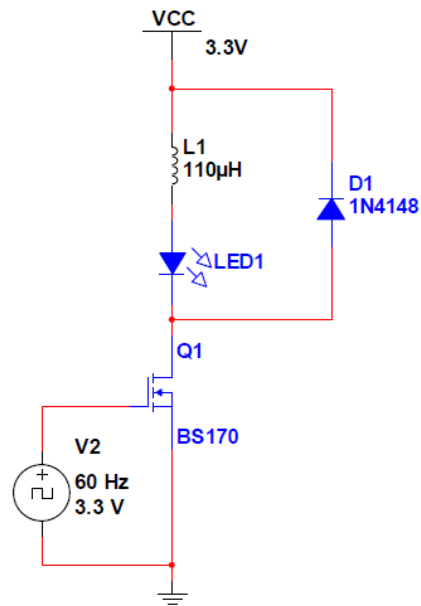


Figure 4 Last Year's MQP LED Driving Circuit [1]

Research

Survey

We performed an online survey and sent it out to all of the undergraduates at WPI. The survey was created to see what biometric signals potential users would want to measure. The survey also requested how much money they were willing to pay and at what size with what kind of features. A little over 100 undergraduates participated in the survey. The survey results of 100 participants are shown below.

1. Please indicate importance / interest to you for the following body signals: Create Chart Download					
	Not important	Somewhat important	Most important	Rating Average	Response Count
Pulse	6.0% (6)	46.0% (46)	48.0% (48)	2.42	100
Heart Rate	4.0% (4)	32.0% (32)	64.0% (64)	2.60	100
Body Temperature	16.0% (16)	41.0% (41)	43.0% (43)	2.27	100
Oxygen in your Blood	13.0% (13)	53.0% (53)	34.0% (34)	2.21	100
Blood Pressure	18.0% (18)	45.0% (45)	37.0% (37)	2.19	100
Blood Alcohol Content	27.0% (27)	38.0% (38)	35.0% (35)	2.08	100
Hand Strength	53.0% (53)	36.0% (36)	11.0% (11)	1.58	100
			Other (please specify) Show Responses		12
answered question					100

Figure 5 Survey Results 1






3. Do you have a preference on how it would be powered? Create Chart Download			
		Response Percent	Response Count
Solar		43.0%	43
Battery		66.0%	66
USB		32.0%	32
Self-powered (spinning a knob)		26.0%	26
Other (please specify) Show Responses		17.0%	17
answered question			100

Figure 6 Survey Results 2

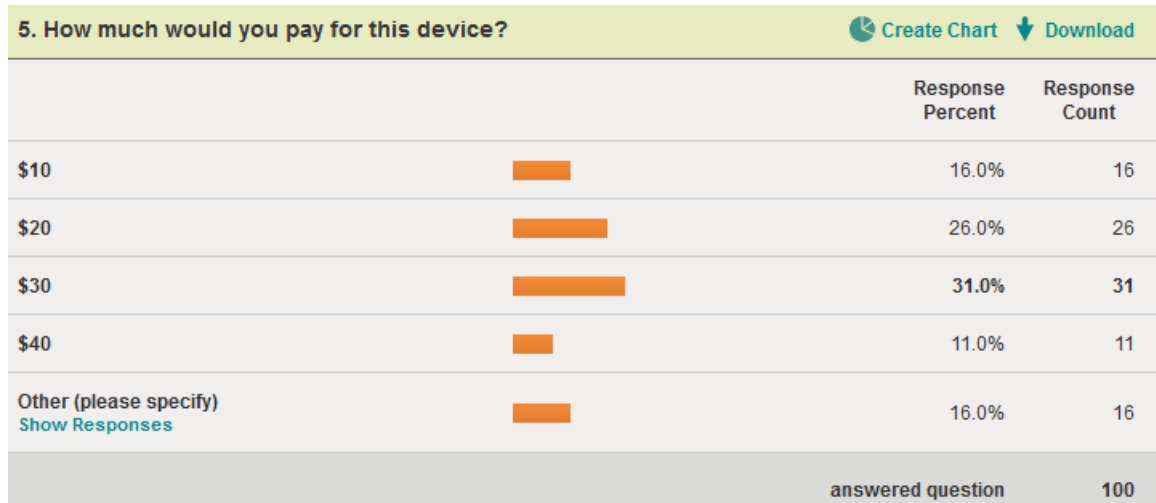


Figure 7 Survey Results 3

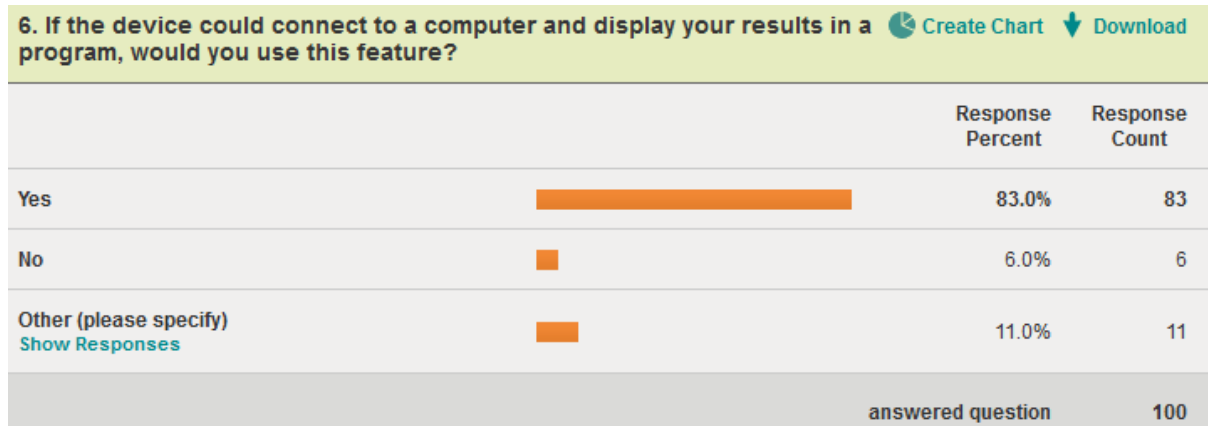


Figure 8 Survey Results 4

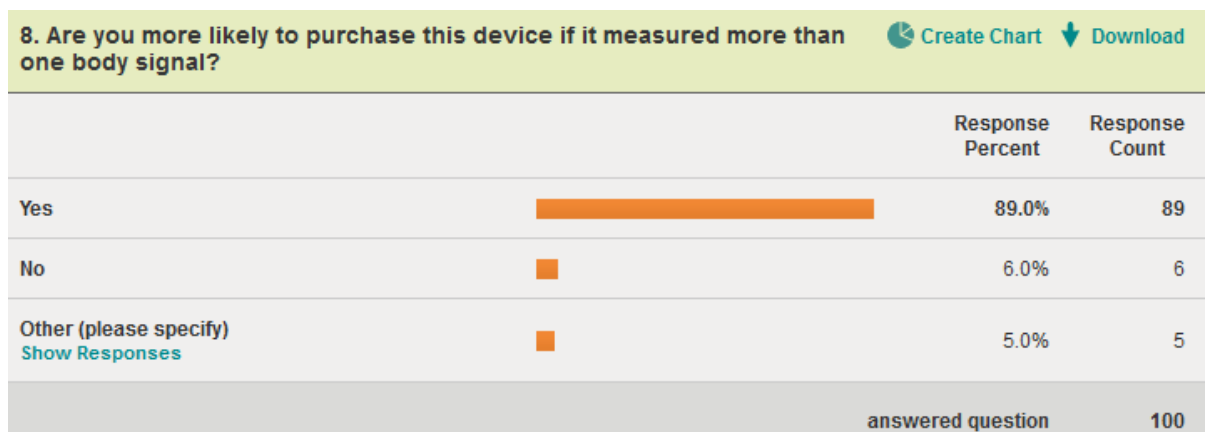


Figure 9 Survey Results 5

Question 2, Question 4, and Question 7 are not displayed because they were written responses. Question 2 asked participants to describe what kind of features they would want a handheld bioinstrumentation device to encompass. The top answers were that the device must be easy to use, contain some type of graphic display, be small but not too small, be able to connect to computer or cellphone, contain a warning system, and be accurate while still giving fast results. Question 4 asked participants what size they would want the device to be. The top answers were smartphone and mp3 player size, watch size, and portable videogame console size. Question 7 asked participants where they would most likely find themselves using this device. The top answers were at their home, working out (gym), and everywhere. We used the survey results to help shape the goals of our project and will try to meet some of the participant's recommendations.

Analog Device's ICs

In this part of the report, we discuss two available chips we considered to use in our project and show their specifications as to see whether it could meet our demand of power, budget and project requirement based on our later value analysis.

ECG Chip - ADAS 100

This is a low power, 5-Electrode Electrocardiogram (ECG) Analog Front End (AFE). It can measure ECG signals, thoracic impedance, and pacing artifacts, lead on/off status and output these information in the form of a data frame supplying either Lead or Electrode data at programmable data rates. Post-processing can be realized externally on a DSP or microprocessor. The ADAS 1000 has a flexible power/noise scaling architecture where noise

can be reduced at the expense of increasing power consumption while signal acquisition channels may be shutdown to save power. Data rates can also be reduced to save power. If we use this chip to measure and process ECG signals, we will choose our own microprocessor.

Features and Specifications:

Bi-potential signals in	Digitized signals out
5 ECG channels	11mW (1 lead), 14mW (3 leads), 19mW (all)
Size	56 lead 8mm x 8mm or 56 lead 10mm x 10mm
Power	3.3 – 5.5V supply rail
Price	\$28 for 64 lead

Analog Microcontroller – AduC70xx

The ADuc70xx are fully integrated, 1 MSPS, 12-bit data acquisition systems with multichannel ADCs, 16-bit/32-bit MCUs, and Flash memory on a single chip. If we want to implement both ECG and PPG on a chip, we will probably choose the Analog Device Microcontroller. Then we need to design our own ECG and PPG model.

Features and specifications:

Analog I/O	Multichannel, 12-bit 16 channel ADC, 1MSPS
------------	--

Digital I/O	Multichannel, 12-bit DAC outputs
Power	3V operation: 11mA @5MHz, 40mA @41.78MHz
Memory	SRAM 8192Bytes, No Flash Memory
Size	40-lead 6mm x 6mm LFCSP
Price	\$4.03 - \$8.91
Other features	On-chip voltage reference and temperature sensor

Instrumentation Amplifier - AD8235

Another ADI chip we may consider to use is the AD8235. It is the lowest power instrumentation amplifier from ADI which draws a maximum quiescent current of 40uA and a maximum 500nA of current during shutdown mode. Therefore, it is a good choice for monitoring and measuring biometric signals from our body. It can reject common-mode signals and serves as the primary gain stage set at G=5-200.

Features and Specifications:

Power	40uA max supply , 1.8V min supply
High Common Mode Rejection Ratio	100dB CMRR
Size	1.5mm x 2.2mm
Price	\$1.59 (1000pcs)

Biometric Signals

In this part of the report, we will discuss a few biometric signal interested and explore the method of measurement.

Heart rate and pulse

Heart is the number of heartbeats per unit of time, and typically expressed as beats per minute (bpm). Heart rate is measured by finding the pulse of the body. People can simply measure their pulse rate by pressing any point on the body where the artery's pulsation is transmitted to the surface with the index and middle fingers. A more precise method of determining pulse involves the use of an ECG or PPG.

Various formulas are used to estimate individual maximum heart rates based on age. The following formula shows a reference maximum heart rate and the table describes normal heart rates based on people from different age. We will probably use this information to design our warning communication system.

$$\text{Max safe HR} = 205.8 - (0.685 * \text{age}) \pm 6.4\text{bpm} [2]$$

Table 1 Heart rate vs. Age

New born	1-12months	1-2years	2-6years	7-12years	13-adults	Adult athletes
120-160	80-140	80-130	75-120	75-110	60-100	40-60

ECG Signal

An electrocardiogram (ECG) is a recording of the electrical activity on the body surface generated by the heart. ECG measurement is achieved by collecting signals from skin electrodes placed on the designed locations on the body. The ECG signal is characterized by six peaks and valleys.

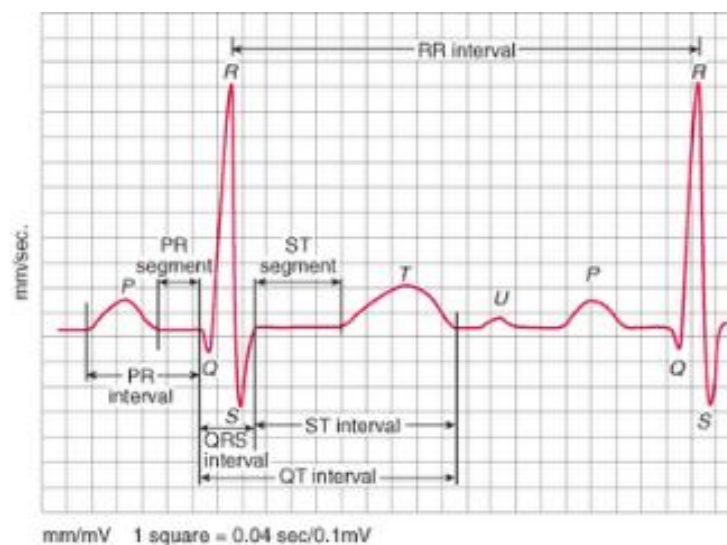


Figure 10 Typical ECG Signal [3]

Each time the heart beats, an ECG signal appears as above. By measuring the R-R intervals, we could determine the subject's heart rate. Therefore, what we concern most is the QRS wave. Its duration ranges from 0.06 – 0.1 sec while largest amplitude should not exceed 4.5mV (Increased amplitude indicated cardiac hypertrophy) with a dc component of 300mV caused by electrode-skin contact and a common mode component of up to 1.5V

resulting from potential between the electrodes and ground. The useful bandwidth can range from 0.5Hz to 50Hz.

ECG signals may have the following main kinds of noise:

1. Power-line interference: 50-60Hz from power
2. Electrode contact noise: baseline drift
3. Shifts in baseline caused by changes in the electrode-skin impedance
4. EMG signals mixed with ECG signals
5. Respiration causing drift in the baseline

PPG Signal

Photoplethysmogram (PPG) is a non-invasive method to detect cardio-vascular pulse wave that propagates through the body using a light source and a detector.

In a pulse-oximeter, the calculation of the level of oxygenation of blood (SaO_2) is based on measuring the intensity of light that has been attenuated by body tissue. The two common forms of the molecule, oxidized Hemoglobin and reduced Hemoglobin, have significantly different optical spectra in the wavelength range from 500nm to 1000nm, as shown in the figure below.

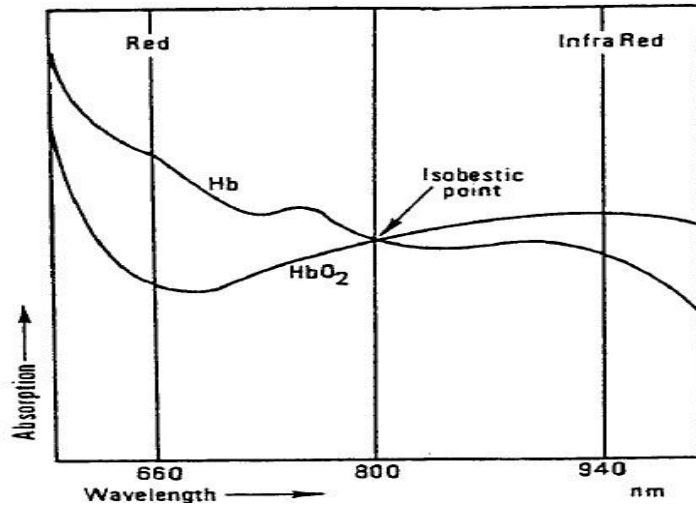


Figure 11 Wavelength vs. Absorption for Hb and HbO₂

SaO₂ is defined as the ratio of the level oxidized Hemoglobin over the total Hemoglobin level:

$$SaO_2 = \frac{HbO_2}{HbO_2 + Hb}$$

Two different wavelengths of lights are used; each is turned on and measured alternately. By using two different wavelengths, the final formula to calculate the level of oxygenation of blood is reduced to:

$$R = \frac{\log(Iac) \lambda_1}{\log(Iac) \lambda_2}$$

Where λ_1 and λ_2 represents the two different wavelengths of light used.

The graph below shows two relationships between the ratio R and the Oxygen Saturation.

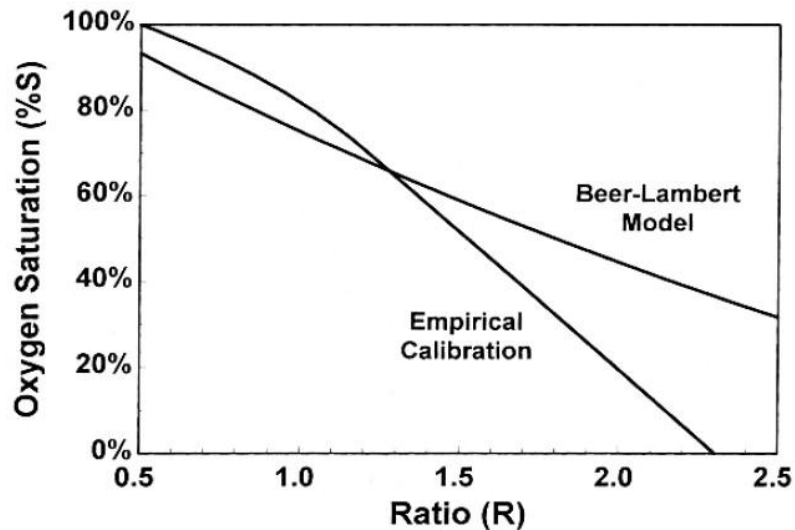


Figure 12 Different formula applied for relationship between R and SaO2

The PPG signal shows each cardiac cycle when the heart pumps blood to the periphery. The DC component is attributable to the absorption of the skin tissue, venous blood and non-pulsatile arterial blood while the AC component is directly attributable to variation in blood volume in the skin caused by the pressure pulse of the cardiac cycle. The amplitude of this cardiac-synchronous pulsatile signal is approximately 1% of the DC level.

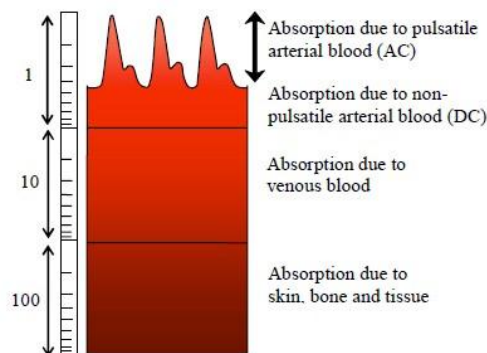


Figure 13 Different portions of the origin signal

When the DC value is removed from the signal, the AC part of the signal left reflects the arterial oxygenation level, which has a similar pattern as Blood Pressure and EKG signals.

(PVC is known as premature ventricular contraction which can be perceived as palpation in the chest.)

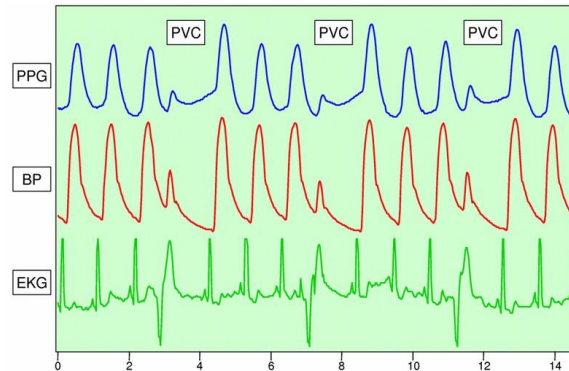


Figure 14 Different signals monitored to see PVC

Skin temperature

Normal body temperature is a concept that depends on the different situations where people are measured. The place in the body, the time of day and level of activity all could give different measuring results. The commonly accepted average core body temperature is 98.6 °F while the average skin temperature is around 91°F. Body temperature is widely used to diagnose irregularities for one's overall health. When one's body temperature is over 98.6°F, he or she may be considered as being ill. Taking a patient's temperature is usually an initial part of a full clinical examination.

To measure one's body temperature, a medical thermometer is used in the mouth or under the arm most likely, which we cannot achieve for the project. However, we can use a thermistor to measure one's skin temperature. A thermistor is a variable resistor whose

resistance increases or decreases nonlinearly with increasing temperature. It commonly has a higher precision within a limited temperature range [-90°C to 130°C]. There are two types of thermistor, PTC and NTC depending on its positive and negative temperature coefficient. The Steinhart- Hart equation is a widely used third-order approximation of the resistance of a thermistor at different temperature. The equation is shown as follows.

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3$$

A, B and C are the coefficient which vary depending on the type and model of thermistor and the temperature range of interest.

Blood Pressure

BP is the vital signs and is the pressure exerted by circulating blood upon the walls of blood vessels. During each heartbeat, BP varies between a maximum (systolic) and a minimum (diastolic) pressure. Classification of BP for adult is shown in the table below.

Table 1. Classification of Blood Pressure (BP) for Adults Age 18 Years and Older

Category	Systolic BP (mm Hg)	Diastolic BP (mm Hg)	Recommendations for Follow-up Dental Patients based on Initial BP measurements
Normal	<130	<85	Recheck at recall (within 2 years)
High Normal	130-139	85-89	Recheck at recall (within 1 years)
Stage 1 Hypertension	140-159	90-99	Recheck within 1 month; if still elevated have patient evaluated by physician within 2 weeks
Stage 2 Hypertension	160-179	100-109	Recheck within 2 weeks; if still elevated have patient evaluated by physician within 2 weeks
Stage 3 Hypertension	180-209	110-119	Have patient evaluated by physician within 1 week
Stage 4 Hypertension	=>210	=>120	Have patient evaluated by physician immediately

* Not taking antihypertensive drugs and not acutely ill. When systolic and diastolic pressures fall into different categories, the higher category should be selected to classify the individual's blood pressure. Isolated systolic hypertension is defined as a systolic blood pressure of 140mm Hg or more and a diastolic blood pressure of less than 90 Hg.
 * Based on the average of two or more readings taken at each of two or more visits following an initial screening.
 Adapted from *The Fifth Report of the Joint National Committee on Detection, Education, and Treatment of High Blood Pressure. 1993.*

Figure 15 Classification of BP

(The American dental hygienists' association)

The measurement of blood pressure refers to the pressure measured at a person's upper arm and is measured on the inside of an elbow at the brachial artery, which is the upper arm's major blood vessel that carries blood away from the heart. (Non-invasively) – Sphygmomanometer, oscillometric, white-coat hypertension

BP is more accurately measured invasively through an arterial line. This method refers to penetrating the arterial wall to take the measurement is much less common and usually restricted to a hospital setting.

ADI MEMS and sensor interface technologies enable highly accurate and cost-sensitive blood pressure monitoring designs. However, we still need to implement the pump and air valve to achieve the design, which are not directly related to our MQP.

Blood Alcohol Content

BAC is a percentage of alcohol in the blood. A BAC of 0.1 means that 0.1% of a person's blood is alcohol. Average individual appears normal with a BAC of 0.01-0.029. When the BAC appears to be over 0.1, individual is likely to show over-expression, emotional swings; when over 0.2, individual will probably experiencing memory blackout. And death is possible when BAC is over 0.4.

Most countries disallow operation of motor vehicles and heavy machinery above prescribed levels of blood alcohol content. In the United States, the legal limit can vary by states but for all states as of 2011 is 0.08 blood alcohol content as by a breath device.

There are three devices to measure BAC, Breathalyzer measures BAC by using a chemical reaction that produce a change in color. The alcosensor uses a fuel cell which detects a chemical reaction. And the Intoxilyzer uses infrared spectroscopy to detect alcohol.

Breathalyzer and Intoxilyzer both involves implement of chemical reaction while alcosensor use a fuel cell where alcohol becomes oxidized and thus produce a greater electrical current , which all very difficult to implement and not directly related to our project.

Hand Strength

Hand strength measurements are of interest to study pathology of the hand that involves loss of muscle strength. Examples of these pathologies are carpal tunnel syndrome, nerve injury, tendon injuries of the hand, and neuromuscular disorders. Measurement of hand

strength involves using grip and pinch dynamometers which is not directly related to our project.

Power

Rechargeable Batteries

According to the results of the survey, people expect our device to be portable so that they could use it during exercise time. Meanwhile, the essential to use the chips requires the power supply to output enough power in order to function all the blocks. Therefore, we decide to use a rechargeable battery as our power and apply USB to charge the battery as people wish.

Rechargeable batteries have lower total cost of use and more eco-friendly than disposable batteries. Although they have higher initial cost, but can be recharged very cheaply and used many times. Several different combinations of chemicals are commonly used including: lead-acid, nickel cadmium (NiCd), nickel metal hydride(NiMH), lithium ion(Li-ion), and lithium ion polymer(Li-ion polymer).

Nickel-cadmium battery (NiCd)

-It uses nickel oxide hydroxide and metallic cadmium as electrodes; however was banned for most uses by the EU in 2004 because of the toxic Cadmium element.

Nickel-metal hydride battery (NiMH)

-It supersedes the NiCd batteries nowadays and has a common consumer and industrial type.

Lithium-ion battery

-It has one of the best energy-to-mass ratios and a very low loss of charge when not in use.

Lithium-ion polymer battery

-It is light in weight and can be made in any shape desired.

For our design, the battery needs to meet the following requirements. It needs to be rechargeable for the convenience of use. Meanwhile, it also need be relatively small and light so it could be implemented on the PCB board of a smart phone or a business card. It has to provide enough voltage in order to function all the blocks of design such as microcontroller, LCD displays and LEDs. In addition, it ought to have enough energy so that the whole design can work reasonable time. Based on the graph above, we have considered the following options:

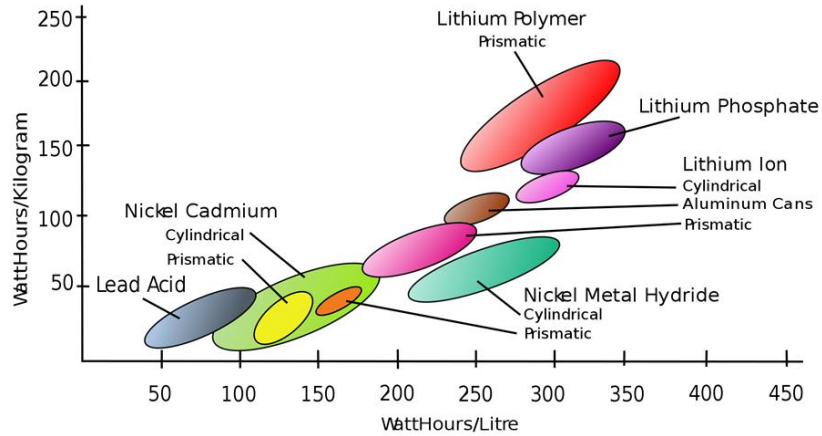


Figure 16 Graph of mass and volume energy density of several batteries

(http://en.wikipedia.org/wiki/File:Secondary_cell_energy_density.svg)

Table 2 several batteries features

Model	Nominal Voltage(V)	Nominal Capacity (mAh)	Dimension(mm x mm) (d x h)	Prize(\$) (101-500)
Li-ion 2032	3.6	40	20 x 3.2	3.23
Li-ion 18650	3.7	2150	18.3 x 64.9	5.39
NiMH Re	1.2	200	25.4 x 8.4	1.76
Li-ion 2450	3.6	120	24.5 x 5.0	3.56
NiCd Re	1.2	700	14.2 x 50.0	0.68

Voltage regulators

A voltage regulator is an electrical device which will automatically maintain a constant voltage level. It usually employs at least one active component such as a transistor or

operational amplifier. Depending on the region in which devices operate, the regulators are often classified into linear regulators and switching regulators. These two types of regulators have their different advantages and are shown in the table below.

Table 3 Comparison between linear and switching regulators

Linear Regulators	Switching Regulators
Low output noise	Better power efficiency
Faster response	A higher output voltage required
Cheaper and occupy less PCB space at low levels of power	Cheaper at high levels of power

In our project, the voltage regulators will convert a DC voltage from our Li-ion battery to several constant voltage levels in order to supply all the major components in the design such as the microcontroller, LCD display, and Analog Device ICs. These voltages range from 1.8V to 5V. Meanwhile our Li-ion battery can provide 3.6V in voltage level. Therefore, we will probably use a boost switching regulator to get higher voltage in order to power the LCD display. We will try to figure out which way is better for our design: implement both linear and switching regulators; only linear regulators with more expensive LCD displays. We have considered several voltage regulators listed below.

1. Linear voltage regulators

Model	Input (V)	Output (V)	Output Current max	Price(\$ in 1000 pcs)	Company
ADP121	2.5-5.5	1.2-3.3(9)/preset	150mA	0.27	ADI
ADP122	2.3-5.5	2.5-3.3(7)/preset	300mA	0.34	ADI
ADP123	2.3-5.5	0.8-5.0/Adjustable $V_{out} = 0.5V (1+R1/R2) + ADJ (R1)$ ADJ = 15nA	300mA	0.34	ADI
LP2951	1.6-30	5,3.3,3V/Adjustable $V = V_{ref} \times (1+R1/R2) - I_{fb} \times R1$ $V_{ref} = 1.235V$ $I_{fb} = 20nA$	100mA	0.27	TI
TL5209	2.5-16	$V_{out} = 1.242 \times (1+R1/R2)$	500mA	0.39	TI
TPS73001	2.7-5.5	$V_{out} = V_{ref} \times (1+R1/R2)$ $V_{ref} = 1.225V$	200mA	0.23	TI

Table 4 Linear VR considered

2. Switching voltage regulators

Table 5 Boost VR considered

Model	Input voltage	Output range	Output current	Price	Company
-------	---------------	--------------	----------------	-------	---------

ADP1612	1.8-5.5V	Vin-20V	-	0.78	ADI
ADP1613	2.5-5.5V	Vin-20V	-	0.70	ADI
MC33063A	3-40V	1.25-40V	750mA	0.27	TI
MC34063A	3-40V	1.55-40V	750mA	0.25	TI

Microcontrollers

A microcontroller is like a small computer on a single IC that contains a processor core, memory, and programmable input and output peripherals. The microcontroller is necessary for taking in data from the signal processing circuit and displaying results on an LCD display. The microcontroller will also control the LED warning system. It will contain a function that will pulse the red LED when the heart rate is too high or too low as well as lighting up the green LED when the measured heart rate is “normal” or average. Lastly the microcontroller will send a warning sound or voice through a speaker if the user needs to go to the hospital due to a high or low heart rate. There are LCD display modules that already contain a controller inside the IC. Although this would be beneficial it is also more expensive.

Some microcontrollers operate at low clock rates and only use 4 bit words for low power consumption. These microcontrollers have the ability to retain functionality while waiting for an interrupt. During this time the power consumption is negligible and is best used for applications that need long lasting batteries. This is important to note, especially if we decide to make our device untethered.

There are several microcontroller architectures. Some of our options include MSP430, Arduino, last year's microcontroller and Analog Device's Microcontrollers. Why are these options? The MSP430 is a microcontroller that my partner and I are familiar with from an embedded systems course. Arduino microcontroller was used in my 2799 project and I have heard from fellow students in the ECE and RBE program that it is a great microcontroller. Last year's microcontroller worked for their project and it could still be a viable option for this years. Lastly using an Analog Device's microcontroller would fulfill one of our goals of implementing one of Analog's ICs into our design.

To settle down on a microcontroller we must decide what we are controlling first. When we pick an electronic display we will know the number of ADC channels we will need. If we choose to incorporate the audio warning system we need to have at least 1 DAC output.

There are several microcontrollers available. The microcontrollers that we choose to research are shown below in Table along with their advantages and disadvantages. All four of these microcontrollers would work for our project, but we need the best fit microcontroller. After researching the microcontrollers it was understood that the MSP430 surpassed the others. The MSP430 not only is beneficial because of previous experience but the fact that it is designed for battery powered applications and has documentation on portable medical applications such as a heart rate monitor. As many would say we don't need to reinvent the wheel so to have sample code of a pre-existing application will greatly benefit our project. There are numerous features of the MSP430. One of the most important

features of the MSP430 is the internal digitally controlled oscillator which will allow precision timing.

Table 6 Microcontroller Options

Microcontroller	Advantages	Disadvantages
TI MSP430	<ul style="list-style-type: none"> • Medical applications guide • Previous experience • Ease of programming • Price • LCD driver • Launchpad/Evaluation board • Lowest power consumption • Designed for battery powered applications 	<ul style="list-style-type: none"> • Launchpad uses JTAG connector instead of USB • Launchpad size
Arduino Pro	<ul style="list-style-type: none"> • Evaluation board • USB connector • Ease of programming • LCD driver 	<ul style="list-style-type: none"> • Price • Evaluation board size
ADI	<ul style="list-style-type: none"> • Goal of using an ADI chip • Evaluation board 	<ul style="list-style-type: none"> • No LCD driver • Price • Difficult to program • Evaluation board uses JTAG
Microchip PIC	<ul style="list-style-type: none"> • Price • PICKit evaluation board • USB connector 	<ul style="list-style-type: none"> • Difficult to program

USB Interface

USB, which has come to replace RS-232 as a standard port in personal computers and their peripheral, has features that are far superior to the older serial port. Therefore, we decide to use USB as a connection between our instrumentation device and PC. However, it has been difficult to provide the necessary isolation for medical and industrial application.

UART stands for Universal Asynchronous Receiver/Transmitter. The UART is a piece of hardware that translates data between parallel and serial ports. The UART is a feature inside of the MSP430 and will be used for communicating serial data (text, numbers, etc.) to a PC. The UART changes incoming parallel information to serial data which can be sent on a communication line. UARTs are also used to send data from the microcontroller to an LCD.

There are different standards/protocols used from transmitting data such as RS232, TTL, serial, and USB. They are incompatible with each other, but if they can be easily converted. RS232 is the old standard and is starting to become obsolete. Few if any laptops even have the RS232 ports today. USB has become the new universal standard for attaching hardware.

The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. TTL is the signal that is transmitted and received by the MSP430 UART. TTL signals are different from what a PC/USB port understands so it needs to be converted using a USB interface. The FT232RL and CP2102 are two USB interface ICs that convert TTL signal from the MSP430 to the USB/PC.

UART Communication

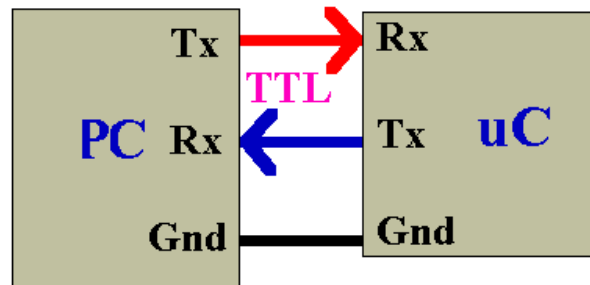


Figure 17 UART Communication between PC and Microcontroller

Tx represents transmit and Rx represents receive. Tx always connects to Rx and vice versa. If Tx is connected to Tx then there will be a lot of current and something will fry.

Baud Rate is the measurement of transmission speed in asynchronous communication. The UART of the microcontroller and the UART going to the PC must agree on a single speed of information (bits per sec). Both devices need to be configured for the same baud rate.

Asynchronous serial transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word which are used to synchronize the sending and receiving data.

When a word is given to the UART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. These two clocks must be

accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word.

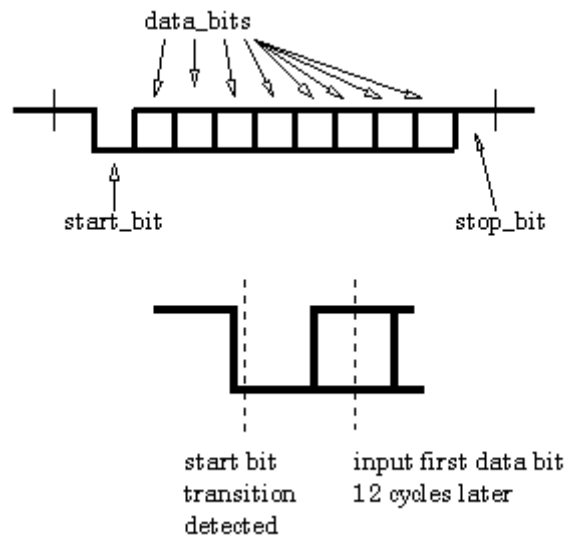


Figure 18 Data bit transition in UART

When data is being transmitted, the sender does not know when the receiver has 'looked' at the value of the bit - the sender only knows when the clock says to begin transmitting the next bit of the word. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter.

When the receiver has received all of the bits in the data word, it may check for the Parity Bits (both sender and receiver must agree on whether a Parity Bit is to be used), and then the receiver looks for a Stop Bit. If the Stop Bit does not appear when it is supposed to, the UART considers the entire word to be garbled and will report a Framing Error to the host

processor when the data word is read. The usual cause of a Framing Error is that the sender and receiver clocks were not running at the same speed, or that the signal was interrupted.

Regardless of whether the data was received correctly or not, the UART automatically discards the Start, Parity and Stop bits. If the sender and receiver are configured identically, these bits are not passed to the host. If another word is ready for transmission, the Start Bit for the new word can be sent as soon as the Stop Bit for the previous word has been sent. In short, asynchronous data is 'self-synchronizing'.

UARTs are defined as Full or Half duplex. If the UART is considered fully duplex then the UART has the ability to send and receive data simultaneously; where the half duplex cannot.

In our project, we will use USB to charge the battery as well as transfer digitalized signal to PC. The USB 1.X and 2.0 specification provide a 5V supply on a single wire from which connected USB device may draw power. A unit load is defined as 100mA in USB2.0 and 150mA in USB3.0. A device may draw a maximum of 5 unit loads (500mA) from a port in USB2.0 and 6(900mA) in USB3.0.

There are many types of USB connectors while the original specification details standard-A and standard-B plugs. For pin configuration of the USB connectors, D+ stands for data+; D- stands for data-. VBUS and GND are presented as "+" and "-".

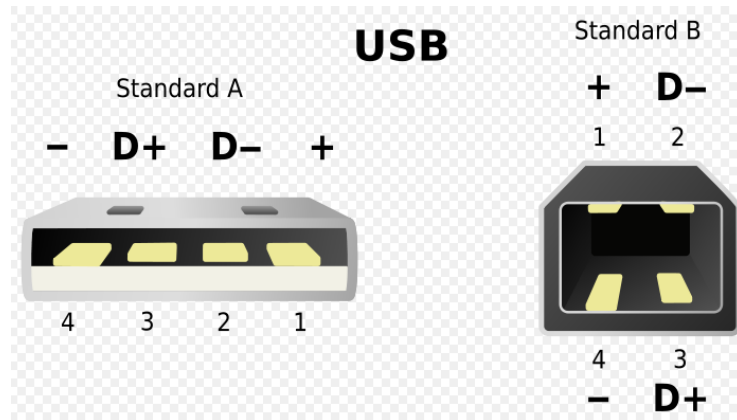


Figure 19 Pin configuration

(<http://en.wikipedia.org/wiki/File:USB.svg>)

Table 7 Pin configuration

Pin	Name	Cable color	Description
1	VBUS	Red	+5V
2	D-	White	Data-
3	D+	Green	Data+
4	GND	Black	Ground

First we considered using an RS-232-to-USB interface, but this approach does not capitalize on the advantage of USB which means the speed of the interface would be limited

to that of standard RS-232. Then we have considered two options, one is from Analog Device called **ADuM4160** USB isolator and another is from FTDI chip called **FT232R** which is a USB to serial UART interface.

1. ADuM4160

It is USB2.0 compatible digital isolator which has a low and full speed data rate at 1.5Mbps and 12Mbps. However, it has a price of **\$4.89** for 1000 pieces. Here shows a typical application for ADuM4160.

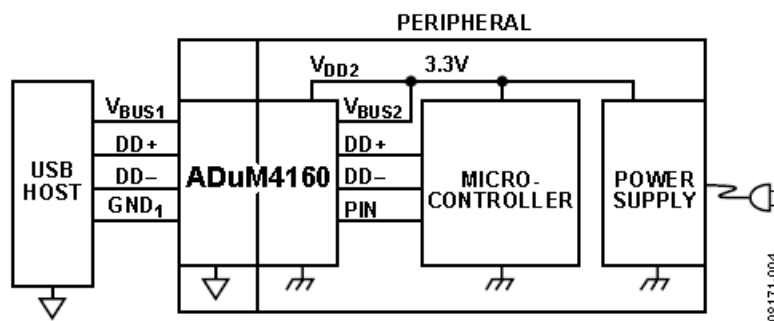


Figure 20 ADI USB isolator configuration

This application involves the peripheral which has its own power source, where no power is required from the USB cable. For low-power peripherals that do not have their own supply, an isolated dc-to-dc converter (such as **ADuM5000**), can be used to supply the peripheral and the ADuM4160, drawing power from the USB cable.

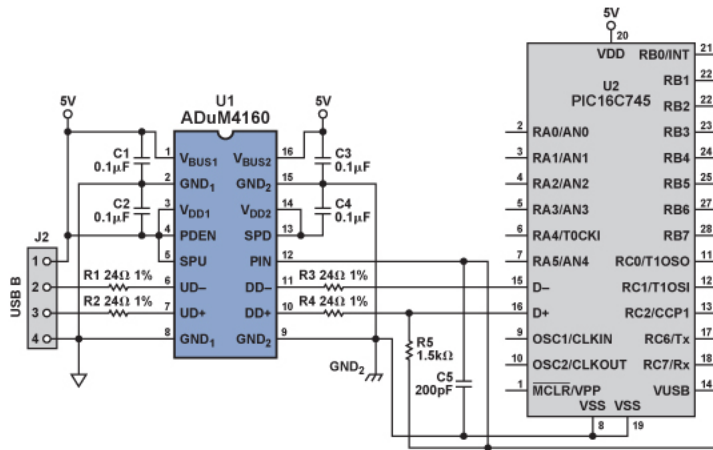


Figure 21 Example for the ADI USB isolator

2. FT232R

The FT232R is a USB to serial UART interface with USB 2.0 full speed compatible. The price would be \$3.14 for over 500 pieces of order. An example of using the FT232R as a USB to MCU UART interface is shown in the figure below.

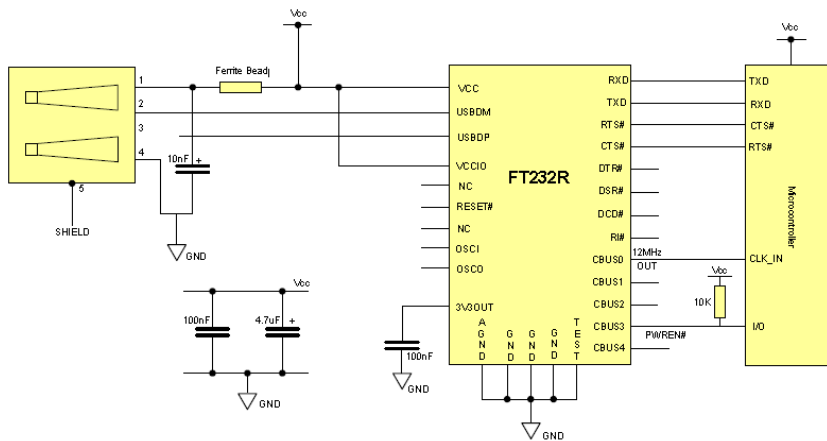


Figure 22 USB interface FT232R example

In this example, the FT232R uses TXD and RXD for transmission and reception of data, and RTS#/CTS# signals for hardware handshaking. (The ferrite bead here is a passive electric component used to suppress high frequency noise.)

User Interface

LCD Display

There are several different types of electronic displays. Our options are CRTs (cathode ray tube), LCDs (liquid crystal display), LEDs, and OLEDs (organic LED) displays.

CRTs are ancient and almost no one uses them anymore. The advantages of CRTs are the resolution and aspect ratio (highest resolutions), black-level and contrast, color and gray-scale accuracy, motion artifacts, and cost. The disadvantages are its sharpness (soft edges), interference (Moire patterns), geometric distortion, brightness, screen shape, emissions (electric, magnetic, electromagnetic fields), and its physical appearance. The disadvantages most definitely outweigh the advantages. CRTs would not be a viable option for our MQP.

LCDs are one of the most popular electronics displays. There are three main types of LCD displays: Segment LCD, Dot Matrix LCD, and Graphic LCD. The segment LCD is also referred to as an alphanumeric LCD. It can display Arabic numbers represented by 7 segments or Roman letters represented by 14 segments; were each signal is treated as one segment. This type of LCD is widely used as the display for scientific instruments. It is easy to control and most cost-effective to develop. Although there are these advantages, it is only

limited to displaying numbers, Roman letters, and fixed symbols. If anything else needs to be displayed a dot matrix or graphic display needs to be used.

Dot Matrix LCD is also known as a Character LCD. It is used to display number of lines of characters. The most commonly used dot matrix LCD displays 1-4 lines of 16-40 characters. Each character is represented by 5x7 dots and its cursor which is typically referred to as 5x8 dots (already includes the cursor). Every character block is addressed separately and can form numbers, Roman letters, characters in other languages and a limited number of symbols. This type of display is used on average for displaying more characters than those in the English alphabet. This display is relatively simple to control and is inexpensive compared to graphic LCDs.

Graphic LCD provides users with a greater degree of flexibility. They are composed of pixels arranged in rows and columns. Each pixel can be addressed individually for text, graphics, or both. This type of display is only used in applications where the user needs to have total control of the whole viewing area. With this flexibility it is very difficult in control.

LED displays are very similar to LCD displays.. A key note about LED displays is that they use LCD screens. The difference is the backlighting of the displays.. LCD displays use CCFL (cold-cathode fluorescent lamps) and incandescent light bulbs. LED displays use LEDs.

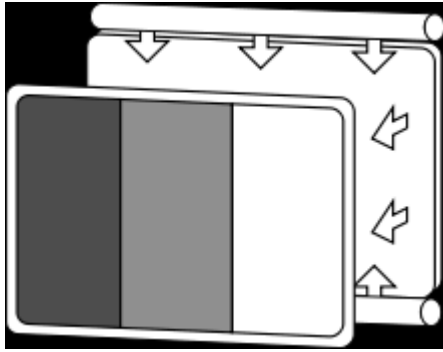


Figure 23 LCD with CCFL backlight

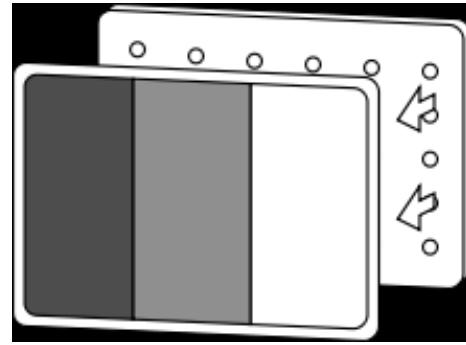


Figure 24 LCD with LED matrix backlight

LED backlighting is most commonly used in small, inexpensive LCD panels whereas CCFL backlights are used on larger displays like computer monitors. Incandescent light bulbs can also be used for backlighting but it has limited lifetime compared to the LED backlight. The top pictures shows how the CCFL backlight lights up the LCD screen in rows and columns whereas the LED backlight uses a matrix.

Since our project isn't constrained by power we don't necessarily need to use a reflective display. Although saving power and money would be nice. But for our project the transmissive display will be the best option. This will allow the user to use the display at any time of day and not need to worry about ambient lighting.

We researched 16x2, 20x2, and 20x4 LCD displays. The first number is the number of characters per line and the second is the number of lines. The 20x4 might be a little too much for our project. These LCD displays can operate at 3.3V or 5V, but to have a lower operating voltage of about half it is roughly double the price. Therefore the search narrowed down to 16x2 displays with an operating voltage of 5V.

LCM type: Characters

Can display 2-lines X 16-characters

Operate with 5V DC

Module dimension: 80mm x 36mm x 12mm

Viewing area size: 64.5mm x 16mm



LED Warning System

The LED warning system as mentioned earlier will consist of 2 LEDs: red and green. If the user's heart rate is not normal the red LED will pulsate and if it is normal the green LED will light up solid. This warning system is important because not everyone knows how to read their heart rate. Fortunately people have learned the symbolism of colors their whole lives. This coming from a simple game as red light – green light or power on (green) or power off (red) and especially good (green) and bad (red). The goal of the LED warning system is to warn the user of their heart rate whether it is good or bad. The user can also make a comparison of the flashing red LED to that of the flashing light on top of an ambulance.

Final Project Decision

Power

The final power decision was that the biocard be both untethered and tethered. This meaning that the biocard would be powered by rechargeable batteries, thus being called

untethered as well as being able to be powered via USB, which is how it would be considered tethered. The USB will also be used to recharge the rechargeable batteries.

Size

The final decision on size was determined to be comparable to that of an index card which is roughly about two business cards side by side. Our decision for this size was based on last year's MQP who had only implemented the ECG and was able to fit it on a business sized PCB. Since we will be implementing both ECG and PPG it only seems reasonable that double the space is necessary.

Price

The final decision as to what price our biocard will be is determined from the survey results as well as last year's MQP and of course the price of the parts we use to build the biocard. We are hoping that the end result of the biocard will be cheap and at the lower end of the price spectrum of our competitors.

We also wanted to keep in mind that this biocard will have multiple purposes included that of an example of what a typical ECE MQP looks like to future students exploring the labs. We would like the price of the biocard to be close to a free "giveaway" to perspective students.

Biometric Signals

Our final decision on to which biometric signals will be measured was determined by the survey results as well as last year's MQP. We wanted to be able to improve upon last

year's MQP as well as meeting some of the requests of the surveyors. The two biometric signals that we decided to measure are ECG and PPG.

Communication to User

Lastly, we wanted to be able to communicate to the user the results of measuring their ECG and PPG signals. To do this we determined an LCD would be the most direct communication which would show the derived numerical results. Along with the LCD, we would like to incorporate a GUI through a PC that would allow users to track their heart rate and oxygen saturation over time. The GUI would also show the user's heart rate on a graph as well just as if they were in the hospital as well as their numerical results of blood oxygen percentage and heart rate.

Block Diagram

This is the block diagram of our project.

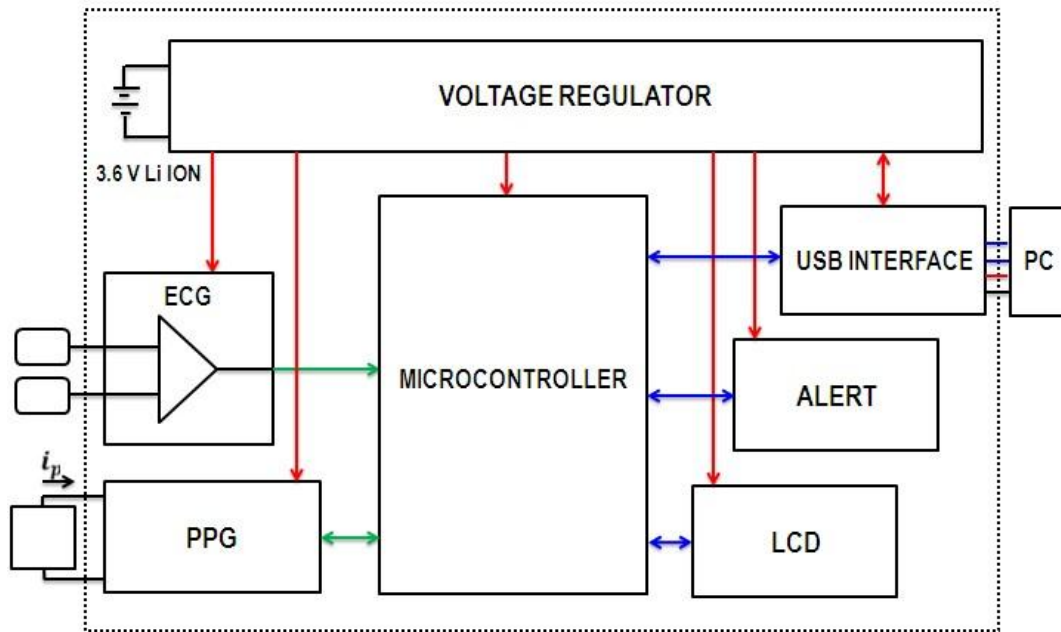


Figure 25 Block diagram

Power

The power system includes a rechargeable battery which provides energy through a voltage regulator to almost all other major subsystem of our project such as the microcontroller, analog signal processing system and the LCD display. The rechargeable battery can be charged when the product is connected to a PC through USB. A charging control need to be implemented between the USB port and the rechargeable battery so that it would not be overcharged to cause possible danger of use.

Digital

The microcontroller will take the sampled biometric signals from on-chip analog-to-digital converter and further filter out the line-frequency noise and display the

signals on a PC screen through USB. Meanwhile, the microcontroller will analyze the QRS complex and calculate the heart rate, display the result on the LCD screen and control the warning system at the same time.

Analog/Signal Processing

Analog Device chips will be used in this part of design to eliminate the dominate 50Hz line-frequency noise and meanwhile amplifies the small biometric signals to take advantage of ADC and be usable for heart-rate detection. Because of the large amplification factor, the output will be sensitive to the variations in electrode to skin contact resistance, which is called baseline wandering. This issue is also managed in this subsystem to allow a constant DC level at the output regardless of the change in skin contact resistance.

User Interface

Our user interface incorporates communication with both customers and computers. It consists of a keypad or buttons to input their ages, a LCD display to show their heart rate, a LED warning system with green and red lights , and PC application through USB.

Detailed Design

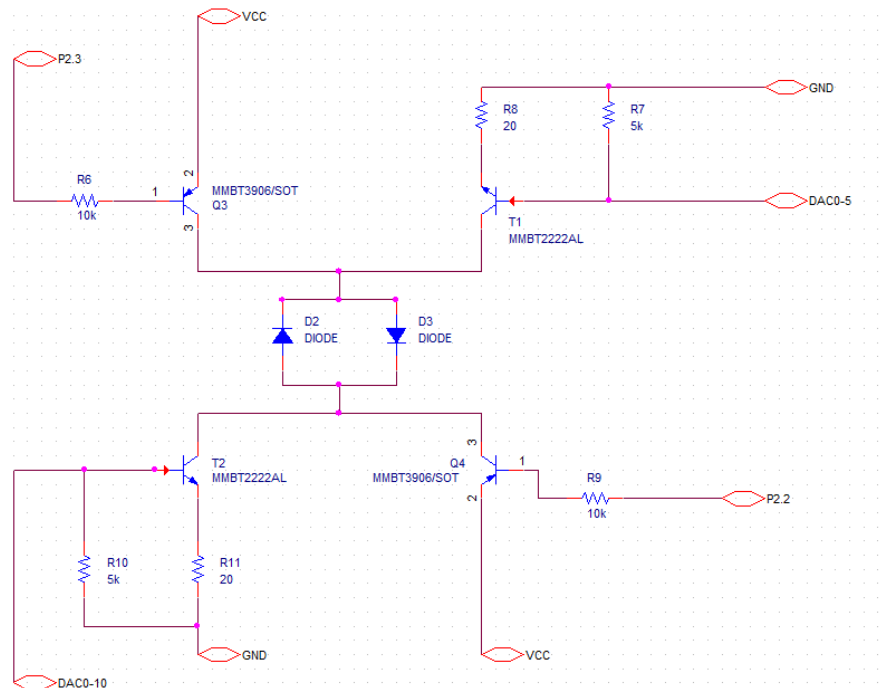
Hardware

This section of the report will concentrate on explaining how each sub block of the hardware are designed and how they are connected and function as a whole. It also details into the choice of values of every components in the circuits. In the second part, it will

introduced the software used for designing the printed circuit board and go through crucial procedures of design.

Circuits Design

LED Driving Circuit



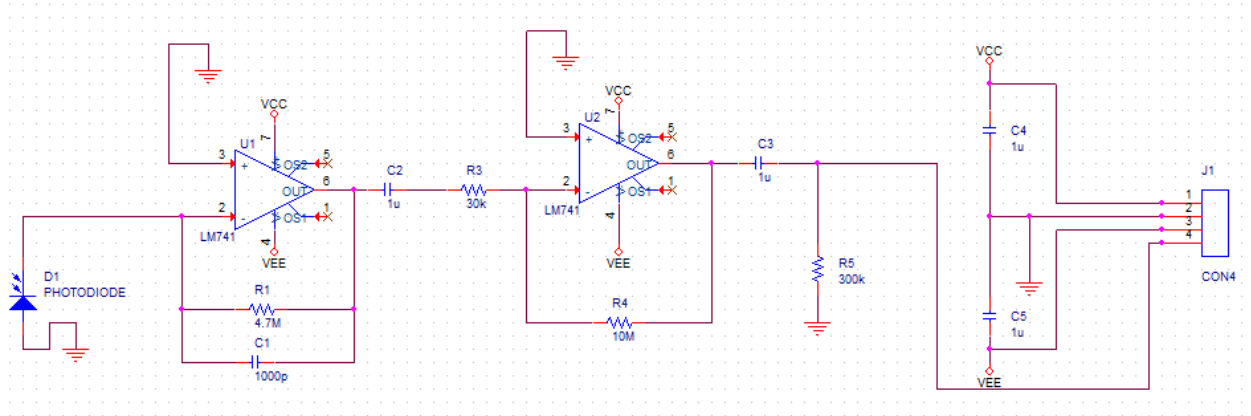
The LED Driving Circuit in the graph above is used to generate the LED Pulses through a H-bridge arrangement. Two LEDs are powered: one for the visible red wavelength at 680nm and another for the infrared wavelength at 940nm. These two LEDs are connected back to back and will be turned on alternatively by two PNP BJTs directly controlled by port2.2 and port2.3 on the MSP430FG439 microcontroller. Two internal 12-bit DACs are connected to NPN transistor in order to control the current through the LEDs and therefore its light output level. The base of each transistor has a pull-down resistor to make sure the transistor is turned off when it is not selected.

The table below gives an approximate estimate of the total current consumed by the H-Bridge. The data are collected using the first printed circuit board designed. VCC is selected to be 3V. Port2.2 and Port2.3 are set to 0 for each testing while the DAC is chosen between values of 0.7V and 1.5V. Each node of the circuit is measured. Case 1 and Case 2 are two extreme consumption cases for the H-bridge arrangement, therefore the consumption of the driving circuit will always be between the current values shown in the table below.

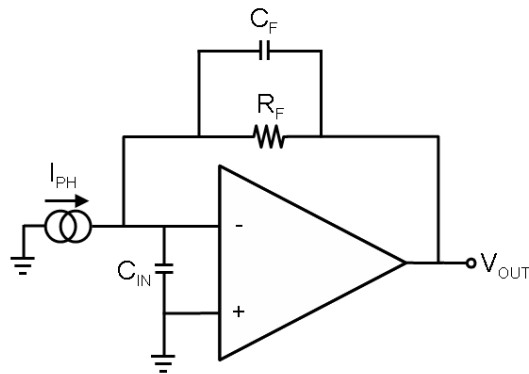
	Case 1	Case 2
Current Path	Current Consumed	Current Consumed
Q3-D2-T2	$0.06V/20\Omega = 3mA$	$0.72V/20\Omega = 36mA$
Q3-R6-Port2.3	$2.3V/10k\Omega = 0.23mA$	$2.3V / 10k\Omega = 0.23mA$
T2-R20-GND	$0.7V/5k\Omega = 0.14mA$	$1.5V / 5k\Omega = 0.3mA$

	Case 1	Case 2
Current Path	Current Consumed	Current Consumed
Q4-D3-T1	$0.06V/20\Omega = 3mA$	$0.06V/20\Omega = 3mA$
Q4-R9-Port2.2	$2.3V/10k\Omega = 0.23mA$	$2.3V/10k\Omega = 0.23mA$
T2-R7-GND	$0.7V/5k\Omega = 0.14mA$	$1.34V/5k\Omega = 0.27mA$

PPG Circuit



The circuit shown above is used to monitor and process the PPG signal. A phototransistor will be placed between two LEDs to catch the light and transfer to the current, and then the current signal will be converted to the voltage signal through a transimpedance amplifier design.



$$V_{out} = -I_{ph} * (Z_c // Z_r)$$

$$V_{out} = -I_{ph} * (1/j\omega C_f // R)$$

$$V_{out} = -I_{ph} * \frac{\frac{R}{j\omega C f}}{\frac{1}{j\omega C f} + R}$$

$$V_{out} = -I_{ph} * \frac{R f}{1 + j\omega C f R f}$$

$$Gain = -\frac{R f}{1 + j\omega C f R f}$$

From the circuit calculated above, it is obvious that the output of the transimpedance amplifier circuit gain is only dependent on resistor between inverting and output node and the current signal has been converted to voltage signal.

In the final version of printed circuit board, a built-in operational amplifier is used to amplify this signal. The signal after this stage consists of a large DC component (around 1V) and a small AC component (around 10mV). The large DC component is caused by the lesser oxygen parts of the body tissue and venous blood which do not contribute to the variation of the AC component. The small AC component is made up of the light modulation by artery blood plus ambient light at 50/60Hz. This ambient noise will be filtered out using digital signal in the microprocessor after the analog signal is sampled by ADC.

Then a large capacitor is used to extract out the DC part of the signal and leave the small AC signal to the gain stage. Then an inverting amplifier is designed to amplify the signal to about 1 volt peak to peak. However, the offset of the second op-amp is also amplifier and added to the output signal. Therefore, a second high pass filter is used to filter out this part in order to take advantage of the full scale of ADC.

The table below shows the measurement results of current signal from the phototransistor with different DAC control voltages.

PCB	IR LED		VIS LED	
V_{DAC} (V)	V_{OUT} (mV)	I_{sig} (nA) = V_{OUT} / R (4.7M Ω)	V_{OUT} (mV)	I_{sig} (nA) = V_{OUT} / R (4.7M Ω)
0.6	0.787	0.167	4.16	0.885
0.7	5.84	1.24	4.17	0.887
0.8	31.8	6.76	4.27	0.909
0.9	70.6	15.0	4.38	0.955
1.0	116.4	24.8	4.49	0.983
1.1	161	34.3	4.62	1.00
1.2	209	44.5	4.71	1.02
1.3	262	55.7	4.78	1.03
1.4	307	65.3	4.83	1.04
1.5	347	73.8	4.86	1.04

Breadboard	IR LED		VIS LED	
V_{DAC} (V)	V_{OUT} (mV)	I_{sig} (nA) = V_{OUT} / R (1M Ω)	V_{OUT} (mV)	I_{sig} (nA) = V_{OUT} / R (1M Ω)
0.6	0.00	0.00	1.05	1.05
0.7	2.12	2.12	1.25	1.25
0.8	18.4	18.4	1.61	1.61
0.9	42.6	42.6	2.32	2.32
1.0	70.1	70.1	3.69	3.69

1.1	100	100	4.92	4.92
1.2	131	131	5.67	5.67
1.3	163	163	6.27	6.27
1.4	195	195	7.26	7.26
1.5	208	208	8.19	8.19

Power Circuits

A 3V li-ion battery is used as the power supply for the project. However, the LCD display need a 5V power supply, therefore a boost voltage regulator is implemented in the power sub block. In addition, a battery charger is used to charge the li-ion battery. All these chips are from Analog Devices and suggested designs from their data sheets are used here. The power block consists of a rechargeable li-ion battery, voltage regulators and a battery charger. (ADP2291, ADP1613, ADP123)

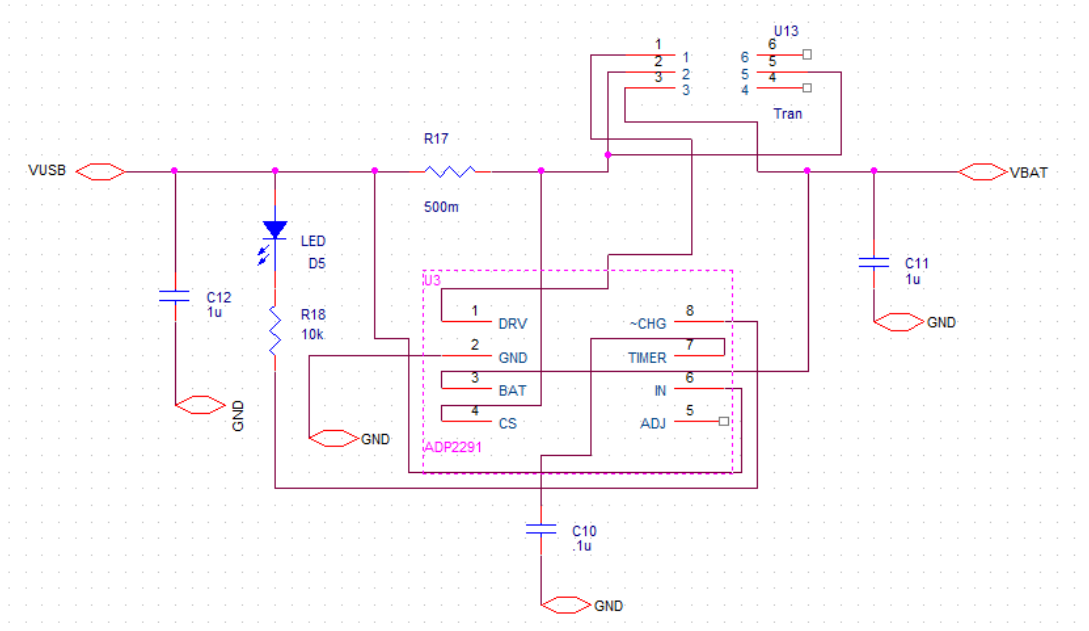


Figure 26 ADP2291

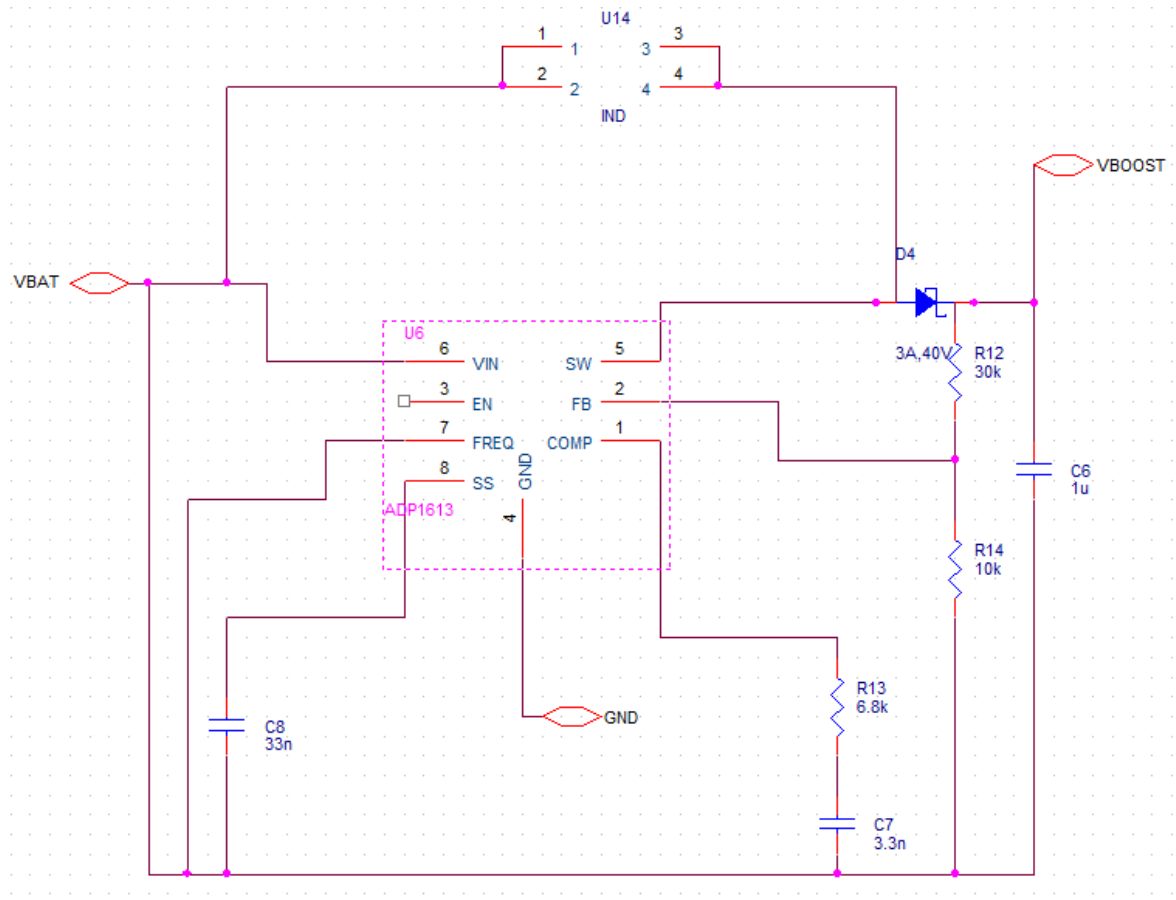


Figure 27 ADP1613

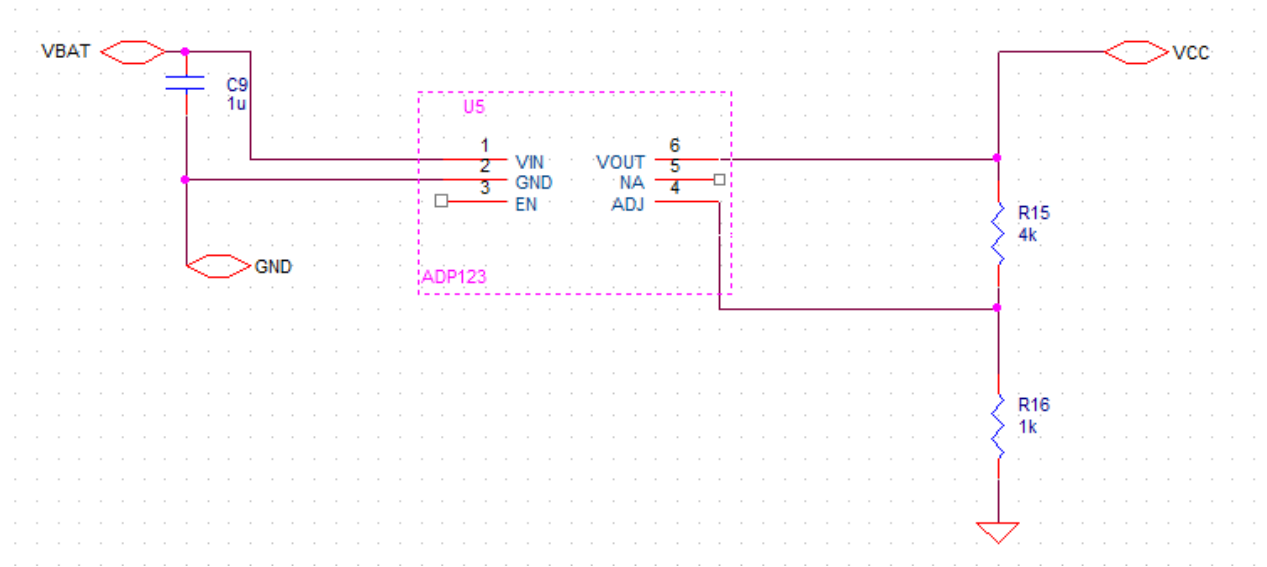
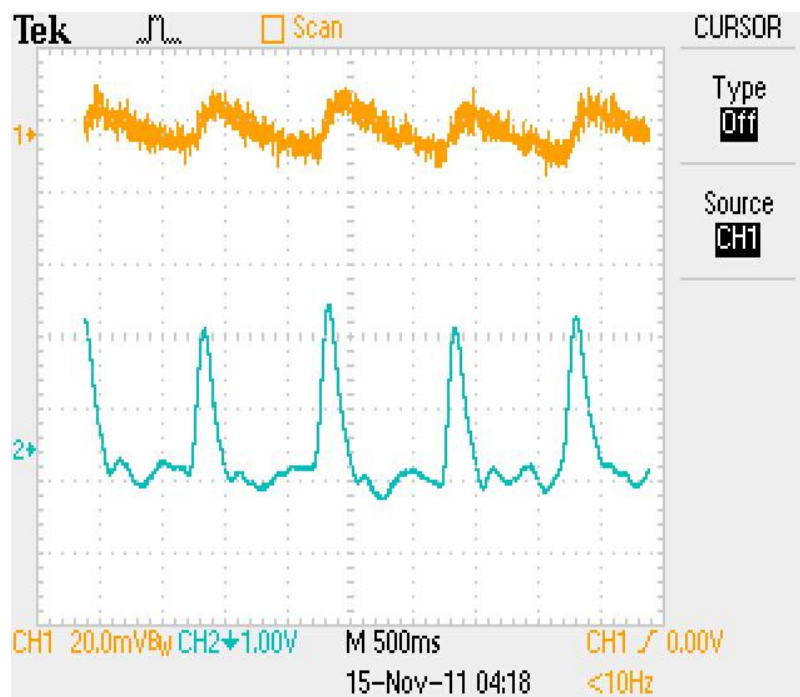


Figure 28 ADP123

Breadboard Design

Before the PCB design, a breadboard prototype is made to test the function of the LED driving circuit and PPG circuit. The blue signal in the graph below is the output of the whole analog front end and is ready to be sampled by ADC.



PCB Design

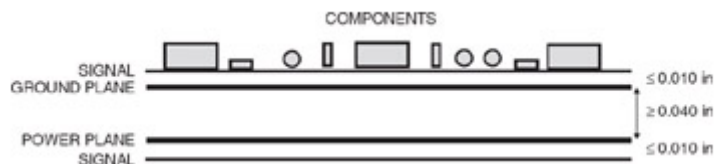
Software and approach

The Cadence Design Systems is used to draw the schematics, run simulations, and design the printed circuit board. Some basic PCB layout considerations includes: components should be grouped into logical functional blocks while the high-speed logic and memory should not be located near the I/O area; keep the oscillators, crystals, and other high-frequency circuitry away from the I/O area. (0.5 inch / 13mm from the I/O area); define a keep out zone around the periphery of the board.

The printed board is designed to have four layers including two signal layers, one ground and one power layer. More considerations are made:

1. A signal layer should always be adjacent to a plane.
2. Signal layers should be tightly coupled (close) to their adjacent planes.
3. Power and ground planes should be closely coupled together.
4. High-speed signals should be routed on buried layers located between planes. The planes can then act as shields and contain the radiation from the high-speed traces.
5. Multiple-ground planes are very advantageous, because they will lower the ground (reference plane) impedance of the board and reduce the common-mode radiation.
6. When critical signals are routed on more than one layer, they should be confined to two layers adjacent to the same plane. As discussed, this objective has usually been ignored.

7. It is usually more important to have the return current flow on a single plane than to bury the signal layers between planes.
8. Space the signal layers as close to the planes as possible (<0.010 in) and use a large core between the power and ground planes.



Procedures

There are two versions of PCB to be made. The first one includes most parts of the hardware design including LED driving circuit and the PPG circuit. Doing the first PCB helps learning the procedures of using Cadence software package as well as testing the hardware part of the design.

In the first version, all components are through-hole except for the LEDs and photodiode because it is very important to make sure the layout of these three components meets the requirements of signal collecting and user experience. Professor McNeill suggested that some part of the components could be put on the back side of the board in order to achieve large area for diodes and better touch between the components and users' finger. The phototransistor is placed between two LEDs so that when users put their fingers press that area, the phototransistor will only receive lights emitted from those two LEDs.

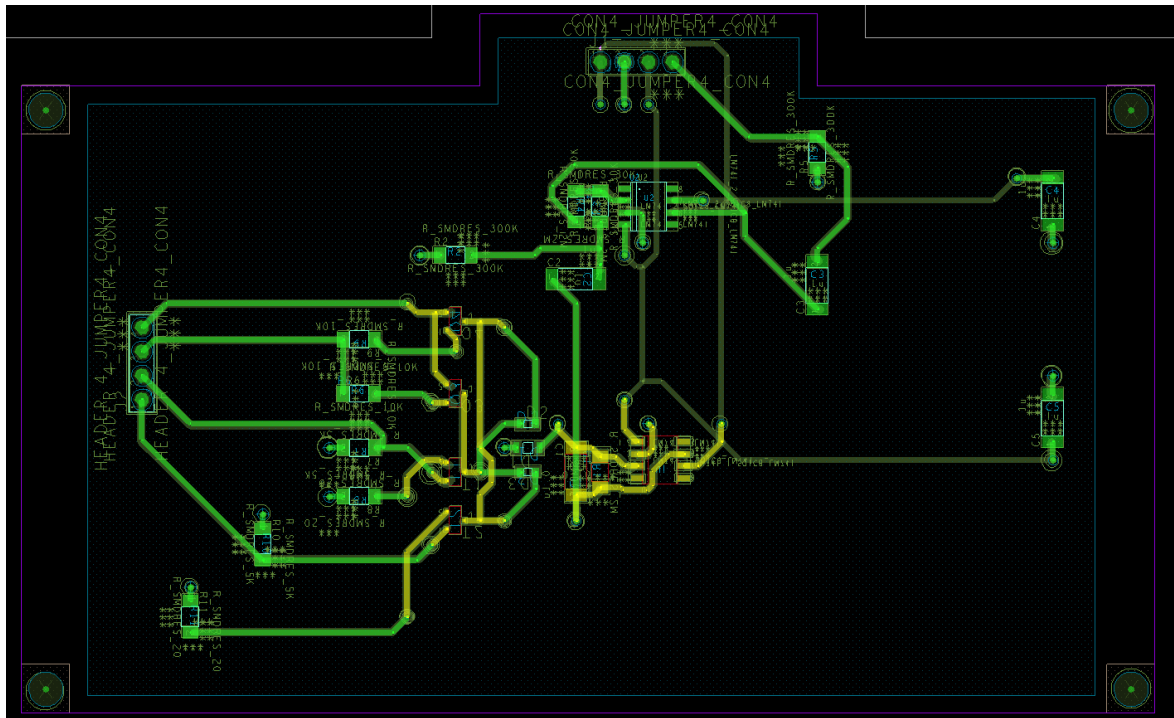


Figure 29 First version of PCB

Software

Microcontroller

Including a microcontroller in our design adds digital signal processing capabilities as well as a simple way to drive the LCD. The microcontroller that was chosen was the MSP430FG439, which was discussed in the section earlier on decisions on components.

Digital I/O Ports

The MSP430FG439 has 6 ports, where each port has 8 I/O pins totally in 48 I/O pins. Port 1 and Port 2 of the MSP430FG439 have interrupt capabilities. We designated 20 I/O ports (P5.1, P5.0, and P4.7-P4.0) for the dot matrix LCD, 2 I/O ports (P2.2 and P2.3) for

driving the infrared and visible LEDs, and 2 I/O (P1.0 and P1.1) for driving the warning LEDs.

We used half of the I/O ports that the MSP430 provides.

The visible and infrared LEDs are time multiplexed at 500 samples per second (500 Hz) each totaling in 1000 samples per second (1 kHz). The visible LED is first sampled at 1ms with the LED on for 400 μ s and off for another 600 μ s. The same goes for the infrared LED, sampled at 1ms with the LED on for 400 μ s and off for another 600 μ s.

Timer

Timer A is used to control the multiplex sequence and automatically start the ADC conversion. Timer A has been setup that it is operating in UP Mode with a period of 1ms or 1 kHz frequency. Timer A duty cycle is set to 69% with TACCRO = 31 and TACCR1 = 10. At the start of Timer A the first interrupt CCR1 is enabled which designates the first LED sequence until CRR0 which is the second interrupt. Once the second interrupt is triggered a new LED sequence initiates.

This can be seen that at 0ms or 0 Hz Timer A is first interrupted (watchdog timer is turned off) the first LED sequence starts until 1ms and then the second interrupt is enabled starting the second LED sequence.

ADC

The ADC is used to convert the analog signals that are multiplexed together into digital signals. The ADC needs to take at least 2 samples to calculate the heart rate and oxygen saturation level. The first is at the transimpedance op-amp (OAO) output (for dc

tracking) and the second is from the second op-amp (OA1) output. These 2 samples are taken back to back using the internal sample timer of the ADC. The internal sample timer is set configured by setting the MSC bit in the ADC control register.

The ADC clock is set to the internal ADC oscillator which is 5MHz. The sampling time is then 64 ADC clock cycles which is $64 * 200\text{ns} = 1.28\mu\text{s}$. The time to convert 1 sample is $(64 + 2^{n+1}) * \text{ADC clock} = (64 + 13)(1.28\mu\text{s}) = 15.4\mu\text{s}$. Since at least samples are necessary to calculate the heart rate and oxygen saturation then the time it will take to sample and convert the signals will be $2 * 15.4\mu\text{s} + 2 * 1.28\mu\text{s} = 33.36\mu\text{s}$.

DAC

There are two DACs in the MSP430FG439. The first DAC, DAC0, is used to control the light intensity of the visible and infrared LEDs and the second DAC, DAC1, is used as an offset input of the second op-amp (OA1).

The first DAC, DAC0, must output a voltage between .7V and 2.5V for operation of the LEDs. The current consumed is determined by the output of the DAC0. We set the DAC0 in a target range so the current consumption of the board would not vary so extremely. To get the best results to calculate the heart rate and oxygen saturation a large signal is needed. To get this large signal a large amount of light needs to be emitted from the LEDs. Knowing this, the target range was set to the higher end of the DAC0s operation voltages.

The target range has high and low thresholds similar to a Schmitt trigger. In the program that light intensity is controlled by comparing the LED light intensities previously

sampled with the target high and low thresholds. If the LED intensity is out of the range then small adjustments are made to bring the LED back into range. Once the LED intensity is back in range the data is then able to be sampled and converted by the ADC.

The second DAC, DAC1, is the DC component that has been extracted from the DC tracking filter which is around 1V. The extracted voltage is used to offset the non-inverting input of the second op-amp (OA1).

Operational Amplifier

The MSP430FG439 has 3 internal op-amps, which we utilized 2 of them. The first op-amp (OA0) is used as a transimpedance amplifier. The non-inverting input is grounded and the inverting input is connected to the phototransistor. OA0 also has feedback from the inverting input to its output. The feedback consists of a resistor and capacitor. The output of the op-amp which consists of both ac and dc components is fed into the inverting input of the second op-amp OA1. The output of OA0 also is fed into the ADC and into a DC tracking filter which removes the ac component holding onto the DC component. The DC component held in DAC1 is fed into the non-inverting input of the second op-amp. OA1 now has the DC component on its non-inverting input and DC and ac on its inverting input. OA1 compares the two inputs and cancels out the DC component thus only amplifying the ac component. The ac component is roughly around 1mVpp and the DC at 1 V.

Schematics and pin configuration

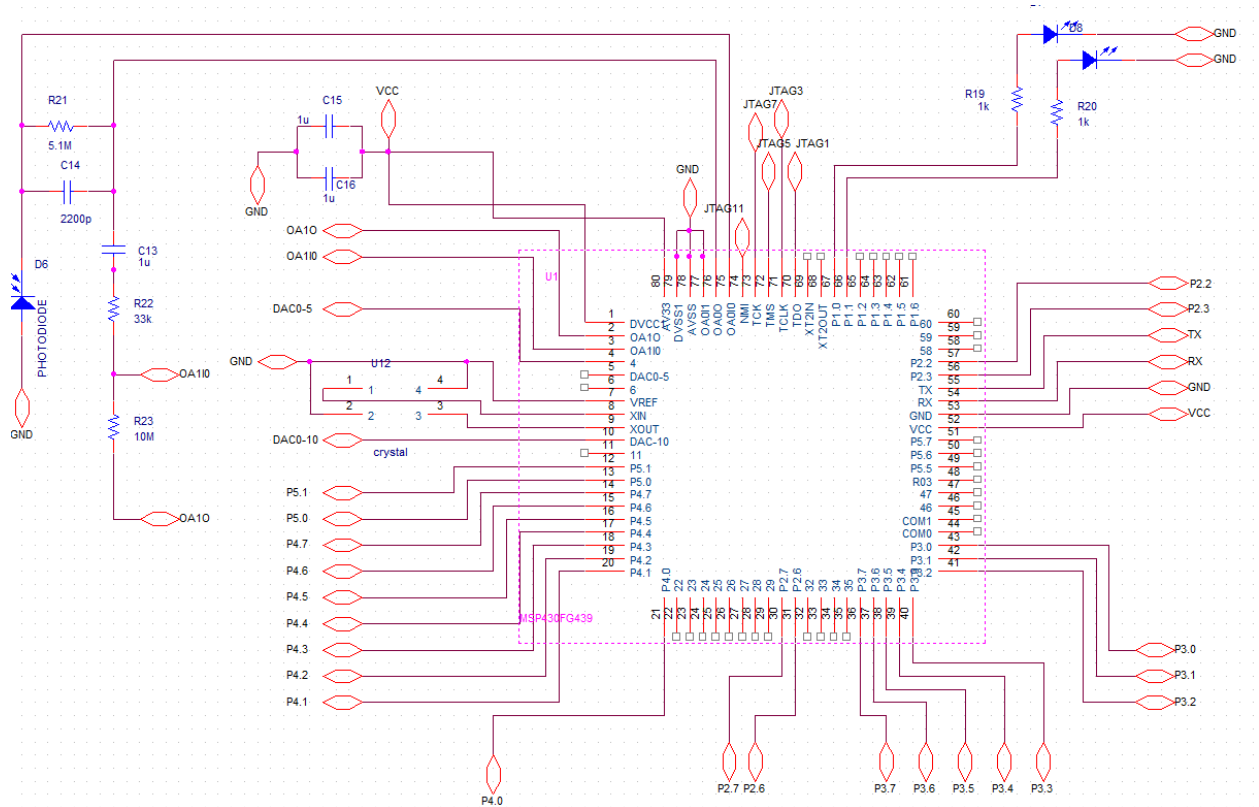


Figure 30 Microcontroller schematics

PIN	NAME	DESCRIPTION
1	DVCC1	Digital voltage supply (positive terminal)
2	OA10	Output of op-amp 1
3	OA10	Inverting input of op-amp 1
4	N/A	Not Used
5	DAC0	Output of DAC0
6	N/A	Not Used
7	VREF	Positive output terminal of the reference voltage in the ADC

8	XIN	Input terminal of the crystal oscillator
9	XOUT	Output terminal of the crystal oscillator
10	DAC0	Output of DAC0
11	N/A	Not Used
12	P5.1	General digital I/O used for the dot matrix LCD
13	P5.0	General digital I/O used for the dot matrix LCD
14	P4.7	General digital I/O used for the dot matrix LCD
15	P4.6	General digital I/O used for the dot matrix LCD
16	P4.5	General digital I/O used for the dot matrix LCD
17	P4.4	General digital I/O used for the dot matrix LCD
18	P4.3	General digital I/O used for the dot matrix LCD
19	P4.2	General digital I/O used for the dot matrix LCD
20	P4.1	General digital I/O used for the dot matrix LCD
21	P4.0	General digital I/O used for the dot matrix LCD
22	N/A	Not Used
23	N/A	Not Used
24	N/A	Not Used
25	N/A	Not Used
26	N/A	Not Used
27	N/A	Not Used
28	N/A	Not Used

29	N/A	Not Used
30	P2.7	General digital I/O used for the dot matrix LCD
31	P2.6	General digital I/O used for the dot matrix LCD
32	N/A	Not Used
33	N/A	Not Used
34	N/A	Not Used
35	N/A	Not Used
36	P3.7	General digital I/O used for the dot matrix LCD
37	P3.6	General digital I/O used for the dot matrix LCD
38	P3.5	General digital I/O used for the dot matrix LCD
39	P3.4	General digital I/O used for the dot matrix LCD
40	P3.3	General digital I/O used for the dot matrix LCD
41	P3.2	General digital I/O used for the dot matrix LCD
42	P3.1	General digital I/O used for the dot matrix LCD
43	P3.0	General digital I/O used for the dot matrix LCD
44	N/A	Not Used
45	N/A	Not Used
46	N/A	Not Used
47	N/A	Not Used
48	N/A	Not Used
49	N/A	Not Used

50	N/A	Not Used
51	N/A	Not Used
52	DVCC2	Digital voltage supply (positive terminal)
53	DVSS2	Digital voltage supply (negative terminal)
54	RX	Receive data in (UART mode)
55	TX	Transmit data out (UART mode)
56	P2.3	General digital I/O used to drive the Visible RED LED
57	P2.2	General digital I/O used to drive the Infrared LED
58	N/A	Not Used
59	N/A	Not Used
60	N/A	Not Used
61	N/A	Not Used
62	N/A	Not Used
63	N/A	Not Used
64	N/A	Not Used
65	N/A	Not Used
66	P1.1	General digital I/O used to drive the warning Visible RED LED
67	P1.0	General digital I/O used to drive the warning Visible GREEN LED
68	N/A	Not Used
69	N/A	Not Used
70	TDO	Programming data input port used for the JTAG

71	TCLK	Test data input used for the JTAG
72	TMS	Test mode select used for the JTAG for input programming and testing
73	TCK	Test clock used for the JTAG for input programming and testing
74	NMI	Non-maskable interrupt input used for the JTAG also allows reset
75	OA0I0	Inverting input terminal of op-amp 0
76	OA0O	Output of op-amp 0
77	OA0I1	Non-inverting input terminal of op-amp 0
78	AVSS	Analog supply voltage (negative terminal)
79	DVSS1	Digital supply voltage (negative terminal)
80	AVCC	Analog supply voltage (positive terminal)

LCD

The Softbaugh SBLCDA4, which is located on the TI Experimenter Board, was used to communicate the results of the user's hear rate and percent of oxygen saturation. The final design used a dot matrix LCD not the segment LCD. The SoftBaugh SBLCDA4 supports 4MUX operation and interfaces to the driver peripheral of the MSP40FG4618. On the MSP430FG4618 S0-S22 pins were used to drive the LCD segments.

Noise Analysis

The quantization error models of an ADC are used to determine the rounding and truncating inaccuracies caused by quantizing an analog signal. The original input signal must be greater than or equal to one least significant bit (LSB) in uniform distribution. The quantization error for the case of rounding has a mean (μ) of 0 and a RMS value or standard deviation (σ) of $1/\sqrt{12}$ LSB. For truncating the $\mu = \frac{1}{2}$ LSB and $\sigma = 1/\sqrt{3}$ LSB.

At lower amplitudes the quantization error becomes dependent on the input signal, which causes distortion. To prevent this distortion the error must be made independent of the input signal noise with an amplitude of 2 LSB.

The quantization noise model is non-linear and signal dependent. For an ideal ADC the quantization error uniformly distributed between $\pm\frac{1}{2}$ LSB. The signal to quantization noise ratio can be written as $SQNR = 20 \log (2^Q) \approx 6.02Q$. Where Q is the number of quantization bits. For testing the signal needs to be full amplitude triangle wave.

If the signal is not a triangle wave and instead is a sine wave then the equation changes to the following $SQNR = 1.761\text{dB} + 6.02Q$. Also to determine the quantization noise power the following equation is used $W = (\delta v^2)/12$, where δv is the voltage of level. Again, if the input signal is too small then there will be quantization distortion which will be large.

The resolution of an ADC is its full scale range of the $2^n - 1$. For a 12 bit ADC with a full scale range of 2.5 V, the result is a LSB = 610 μ V. The SQNR of the 12 bit ADC is 74 dB, usually the ideal model.

Figure 1 shows the schematic of the MSP430FG439 configured for our MQP. Table 1 shows the pin designation and its functions for the MSP430FG439.

Results

Hardware

This is a screenshot showing the output signal of the PPG PCB. The output contains a lot of noise interference and can be seen through the oscillations that the heart rate is riding on. These interferences are from the user's fingers and the lights. To get a better result we would need to use a filter to remove the 50/60Hz as well as the 100/120Hz.

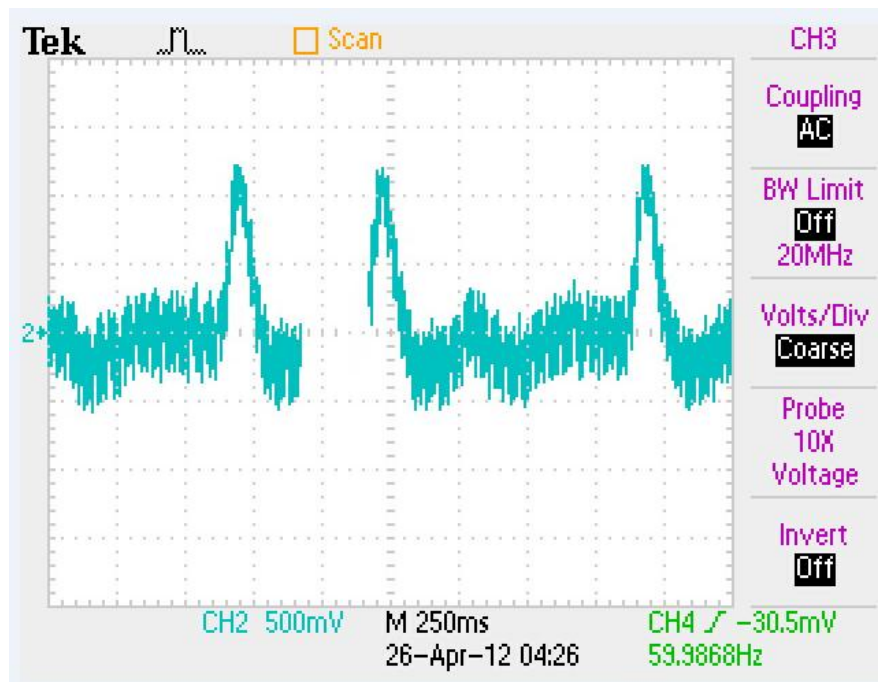


Figure 31 User's heart rate from the output of the PPG PCB

The heart rate was amplified and filters before reaching this point. The original input was around 1mV riding on a DC offset from the user's muscles and now shown is an amplified small signal heart rate around 1.5V.

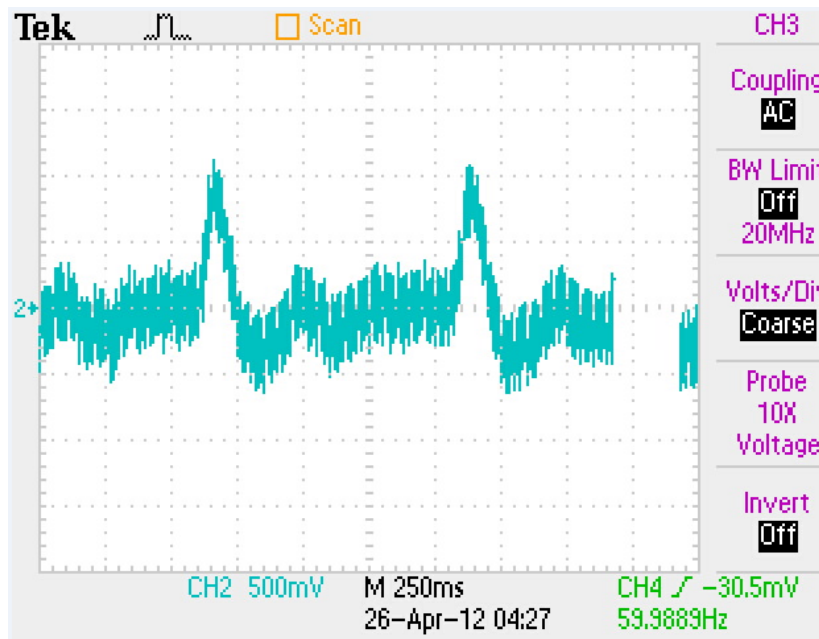


Figure 32 Another screenshot of the user's heart rate

Software

Figure 33 shows the DAC output that controls the light intensity of the LEDs.

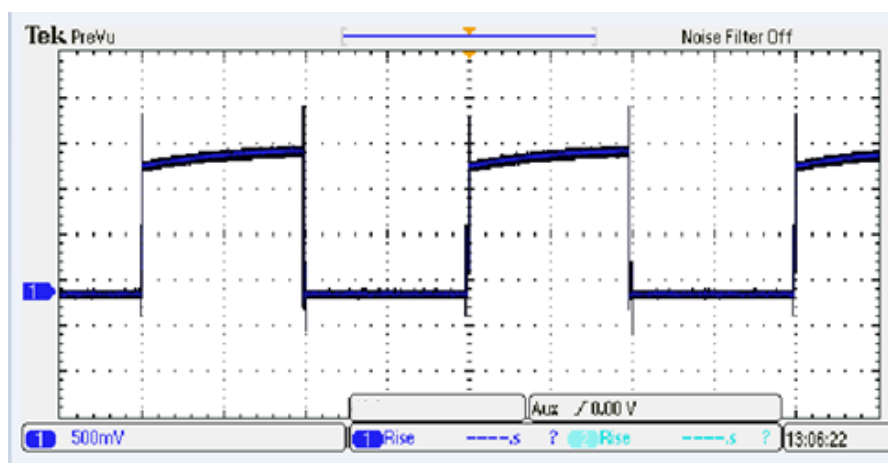


Figure 33 DAC output which is used to control LED intensities

Conclusion

We were able to design both ECG and PPG circuits. We were able to test and verify functionality of the PPG circuit through a breadboard layout. From there we were able to design and fabricate a PCB that contained only the PPG circuit. The fabricated PPG PCB was tested and verified for functionality. The PPG PCB was able to measure the user's heart rate. Unfortunately the PCB design of the PPG circuit was sensitive to light as well as other noise contributing factors such as the 60Hz from the human body.

Regrettably we did not have enough time to finish a complete a final PCB which would incorporate both ECG and PPG circuitry as well the power circuits and the microcontroller. Although we did not get as far as we planned we were able to learn a great deal from our MQP.

Before the MQP started we had never had experience with PCB design and very little understanding of microcontrollers. Now, we can say that we have PCB design experience and a much larger understanding of microcontrollers. Our understanding and skill of PCB design and programming microcontrollers is not at its highest but it is by far greater than it once was and will only be able to grow from here on out.

Our MQP offers promising future work, with the final PCB design fabrication, test, and verification of the ECG and PPG circuit along with the power circuitry and microcontroller. Along with the final PCB we hope that future work will be able to determine the appropriate amount of interrupt time for the ADC to be able to sample and convert the signals it receives.

Bibliographies

[1]

http://www.wpi.edu/Pubs/E-project/Available/E-project-031411-141305/unrestricted/Final_Report.pdf

[2] Robergs R and Landwehr R (2002). "The Surprising History of the $H_{rmax} = 220 - \text{age}$ Equation" , *Journal of Exercise Physiology* 5(2): 1-10

[3] <http://www.analog.com/library/analogDialogue/archives/37-11/ecg.html>

1. "AD8236 | 40 μ A Micro power Instrumentation Amplifier in MSOP | Power Monitors | Power Management | Analog Devices." *Analog Devices | Semiconductors and Signal Processing ICs*. Web. 15 Oct. 2011.
<<http://www.analog.com/en/power-management/power-monitors/ad8236/products/product.html>>.
2. "Analog Devices : Technical Support : Design Tools : SPICE License." *Analog Devices | Semiconductors and Signal Processing ICs*. Web. 15 Oct. 2011.
<http://www.analog.com/Analog_Root/static/techSupport/designTools/spiceModels/license/spice_general.html?cir=ad8236.cir>.
3. Webster, J. G. (2010). *Medical Instrumentation Application and Design*. Hoboken: John Wiley& Sons
4. MSP430™16-bit Ultra-Low Power MCUs - 4xx 8/16MHz LCD Series - MSP430FG439 - TI.com." *Analog, Embedded Processing, Semiconductor Company, Texas Instruments*. Web. 15 Oct. 2011. <<http://www.ti.com/product/msp430fg439>>.
5. "Reflective vs. Transflective Displays." *Laptop LCD Screens from ScreenTek*. Web. 15 Oct. 2011.
<<http://www.screentekinc.com/resource-center/reflective-vs-transflective.shtml>>.
6. "Pulse Oximetry | Patient Monitoring." *Healthcare | Analog Devices*. Web. 15 Oct. 2011.
<<http://healthcare.analog.com/en/patient-monitoring/pulse-oximetry/segment/health.html>>.
7. "ECG-Diagnostic Battery Powered | Patient Monitoring." *Healthcare | Analog Devices*. Web. 15 Oct. 2011.

- <<http://healthcare.analog.com/en/patient-monitoring/ecg-diagnostic-battery-powered/segment/health.html>>.
8. "Analog Devices : Analog Dialogue : Pulse Oximeter." *Analog Devices / Semiconductors and Signal Processing ICs*. Web. 15 Oct. 2011.
<http://www.analog.com/library/analogDialogue/archives/41-01/pulse_oximeter.html>
 9. Cantrell, Mark. "USB Isolator: Analog Dialogue: Analog Devices." *Analog Devices / Semiconductors and Signal Processing ICs*. Web. 15 Oct. 2011.
<http://www.analog.com/library/analogdialogue/archives/43-06/USB_isolator.html>.

Appendix A – Simulation results

Our project we need to simulate the ECG and PPG signals our project design and then simulate the ECG and PPG circuits. This step in our project is extremely important because it allows us to test and analyze our circuit design before spending time and money on a physical design that might not be adequate and function the way we hope. The greatest advantage of simulating these circuits is that it permits us to identify any problem areas before implementing the design and working out those kinks.

Currently we are in the process of simulating the ECG circuit using Cadence PSpice. We attained the SPICE file for the AD8236 Instrumentation Amplifier from Analog Device's technical documents for the AD8236, which was found on the Analog Device's website. Only the sub circuit of AD8236 is shown in APPENDIX A. Also included in the simulation .cir file is another SPICE file that is used to simulate ECG signals. This SPICE file, which was given to us by Professor McNeill, consists of the sum of five pulse voltage sources for each pulse of the

PQRST complex. Figure shows the output of the ECG signal which replicates each pulse of the PQRST complex. This SPICE file is shown in APPENDIX A.

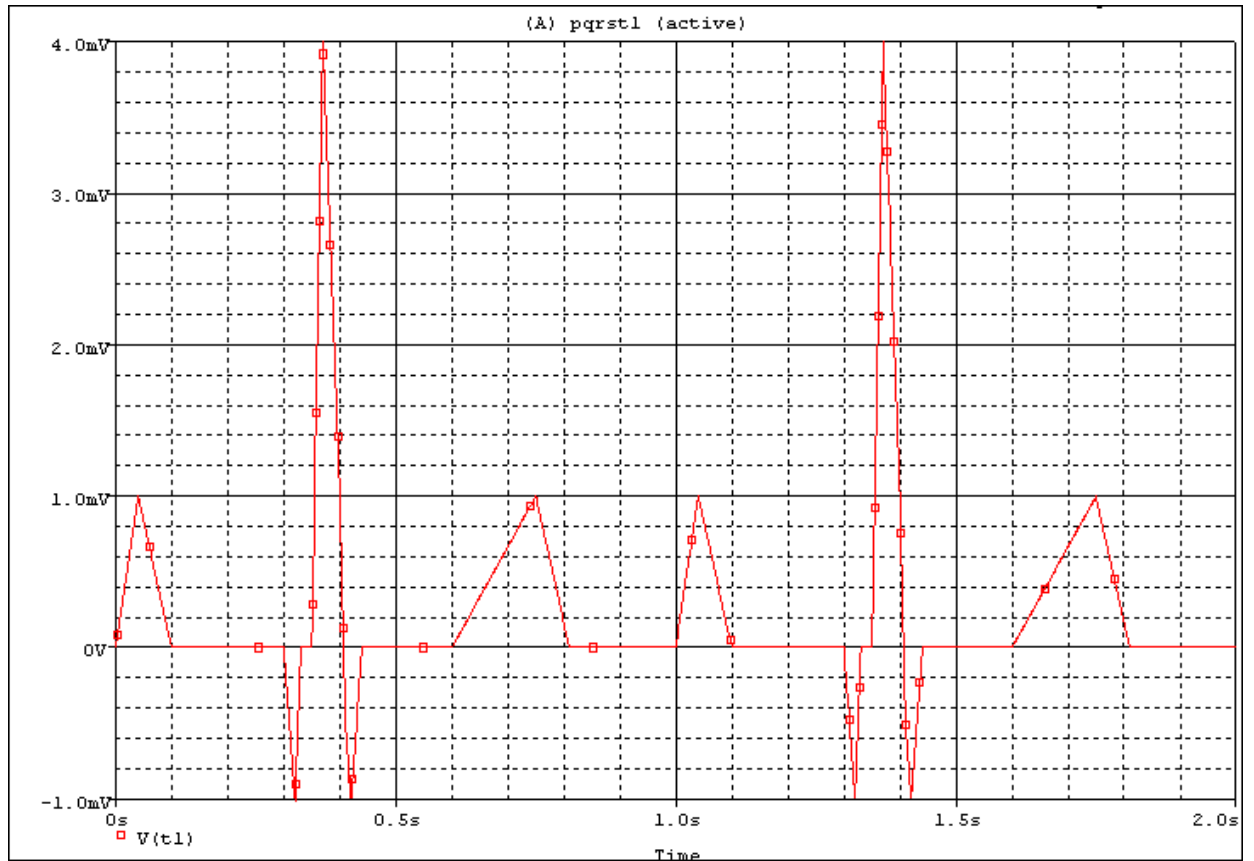


Figure 34 ECG Signal of each pulse of the PQRST Complex

```

00 * Simulation of ECG waveform
01
02 * Sum voltage sources for each pulse of PQRST complex
03
04 Vp p1 0 pulse (0 {vph} {tpd} {tpr} {tpf} {tpw} {tper})
05 Vq q1 p1 pulse (0 {vqh} {tqd} {tqr} {tqf} {tqw} {tper})
06 Vr r1 q1 pulse (0 {vrh} {trd} {trr} {trf} {trw} {tper})
07 Vs s1 r1 pulse (0 {vsh} {tsd} {tsr} {tsf} {tsw} {tper})
08 Vt t1 s1 pulse (0 {vth} {ttd} {ttr} {ttf} {ttw} {tper})
09
10 * Period (same for all sources)
11 .param tper=1
12
13 * Pulse amplitude parameters chosen to look like PQRST complex
14 .param vph=1m tpd=0 tpr=.04 tpf=.06 tpw=1u
15 .param vqh=-1m tqd=0.3 tqr=.02 tqf=.01 tqw=1u
16 .param vrh=4m trd=0.35 trr=.02 trf=.04 trw=1u
17 .param vsh=-1m tsd=0.4 tsr=.02 tsf=.02 tsw=1u
18 .param vth=1m ttd=0.6 ttr=.15 ttf=.06 ttw=1u
19
20 .tran 1m 2
21
22 .probe
--

```

Figure 35 ECG Signal SPICE file

Appendix B - Code

```

/* Samantha Fontaine
MQP 2011-2012
PPG Program
*/

#include "msp430xG46x.h"
#include "stdint.h"
#include "intrinsic.h"
#include "math.h"
#include "LCD.h"

/***** DEFINITIONS *****/
/* LCD Segment Configuration */
#define seg_a 0x01
#define seg_b 0x02
#define seg_c 0x04
#define seg_d 0x08
#define seg_e 0x40
#define seg_f 0x10
#define seg_g 0x20
#define seg_h 0x80

```



```

#define NUM_0    (seg_a | seg_b | seg_c | seg_d | seg_e | seg_f)
#define NUM_1    (seg_b | seg_c)
#define NUM_2    (seg_a | seg_b | seg_d | seg_e | seg_g)
#define NUM_3    (seg_a | seg_b | seg_c | seg_d | seg_g)
#define NUM_4    (seg_b | seg_c | seg_f | seg_g)
#define NUM_5    (seg_a | seg_c | seg_d | seg_f | seg_g)
#define NUM_6    (seg_a | seg_c | seg_d | seg_e | seg_f | seg_g)
#define NUM_7    (seg_a | seg_b | seg_c)
#define NUM_8    (seg_a | seg_b | seg_c | seg_d | seg_e | seg_f | seg_g)
#define NUM_9    (seg_a | seg_b | seg_c | seg_d | seg_f | seg_g)
#define NUM_A    (seg_a | seg_b | seg_c | seg_e | seg_f | seg_g)
#define NUM_B    (seg_c | seg_d | seg_e | seg_f | seg_g)
#define NUM_C    (seg_a | seg_d | seg_e | seg_f)
#define NUM_D    (seg_b | seg_c | seg_d | seg_e | seg_g)
#define NUM_E    (seg_a | seg_d | seg_e | seg_f | seg_g)
#define NUM_F    (seg_a | seg_e | seg_f | seg_g)

const unsigned char hex_table[] =
{
    NUM_0, NUM_1, NUM_2, NUM_3, NUM_4, NUM_5, NUM_6, NUM_7,
    NUM_8, NUM_9, NUM_A, NUM_B, NUM_C, NUM_D, NUM_E, NUM_F
};

int32_t mul16(register int16_t x, register int16_t y);

/* FIR filter coefficient for removing 50/60Hz and 100/120Hz from the signals
*/
#if 0
static const int16_t coeffs[9] =
{
    5225,
    5175,
    7255,
    9453,
    11595,
    13507,
    15016,
    15983,
    16315
};
#else
static const int16_t coeffs[12] =

```

```

{
    688,
    1283,
    2316,
    3709,
    5439,
    7431,
    9561,
    11666,
    13563,
    15074,
    16047,
    16384
};
#endif

/* SaO2 Look-up Table */
const unsigned int Lookup [43] =
{100,100,100,100,99,99,99,99,99,99,98,98,98,98,

98,97,97,97,97,97,97,96,96,96,96,96,96,95,95,

95,95,95,95,94,94,94,94,94,93,93,93,93,93};

/* LED Target Range */
#define FIRST_STAGE_TARGET_HIGH      3500    // 1.28V
#define FIRST_STAGE_TARGET_LOW       3000    // 1.10V
#define FIRST_STAGE_TARGET_HIGH_FINE 4096    // 1.50V
#define FIRST_STAGE_TARGET_LOW_FINE  2700    // 1.0V
#define FIRST_STAGE_STEP              5     // 1.83mV
#define FIRST_STAGE_FINE_STEP         1     // 366uV

/*****      DECLARATIONS      *****/
int ir_dc_offset = 2000;           //IR LED DC offset
int vs_dc_offset = 2000;           //VS LED DC offset
int ir_LED_level;                  //IR LED level
int vs_LED_level;                  //VS LED level
int ir_sample;                     //IR sample
int vs_sample;                     //VS sample
char is_IR;                         //IR LED switch
int ir_heart_signal;               //IR LED heart signal
int vs_heart_signal;               //VS LED heart signal

```

```

int ir_heart_ac_signal; //IR LED ac heart signal
int vs_heart_ac_signal; //VS LED ac heart signal
unsigned int rms_ir_heart_ac_signal; //IR LED ac RMS heart signal
unsigned int rms_vs_heart_ac_signal; //VS LED ac RMS heart signal
int32_t ir_2nd_dc_register = 0; //IR LED dc estimator reg
int32_t vs_2nd_dc_register = 0; //VS LED dc estimator reg
unsigned long log_sq_ir_heart_ac_signal; //log of IR LED signal for
calculations
unsigned long log_sq_vs_heart_ac_signal; //log of VS LED signal for
calculations
unsigned long sq_ir_heart_ac_signal; //square of IR LED signal
for calculations
unsigned long sq_vs_heart_ac_signal; //square of VS LED signal
for calculations
unsigned int pos_edge = 0; //positive edge of tracking
signal before first opamp
unsigned int edge_debounce; //edge of signal that is
reflected
unsigned int heart_beat_counter; //variable that will keep
count of heart signal
unsigned int log_heart_signal_sample_counter; //log of counter signal for
calculations
unsigned int heart_signal_sample_counter; //variable that will keep
count of sample heart signal

volatile unsigned int j;

/* The results */
unsigned int heart_rate;
unsigned int heart_rate_LSB = 0;
unsigned int SaO2, Ratio;
unsigned int SaO2_LSB = 0;

/***** Function prototypes *****/
int16_t dc_estimator(register int32_t *p, register int16_t x);
int16_t ir_filter(int16_t sample);
int16_t vs_filter(int16_t sample);
void set_LCD(void);
void display_number(int value, int start, int width);
void display_pulse(int on);
void display_correcting(int x, int on);
void setup_GPIO(void);
void setup_ADC12(void);

```

```

void setup_Timer(void);
void setup_OPAMP1(void);
void setup_OPAMP2(void);
void setup_DAC1(void);
void setup_DAC0(void);
void setup_UART(void);
void VIS_on_IR_off(void);
void IR_on_VIS_off(void);
void heart_beat_tracker(void);
void loop_pos_edge(void);
void IR_heart_ac_sig_200(void);
void more_than_3_beats(void);
void VIS_sig_into_Range(void);
void IR_sig_into_Range(void);

/***** Start of MAIN function *****/

void main(void)
{
    long f1;
    int32_t x;
    int32_t y;

    WDTCTL = WDTPW | WDTHOLD;           // Turn off watchdog
    SCFIO |= FN_4;                       // x2 DCO frequency, 8MHz nominal
    DCO
    SCFQCTL = 91;                         // 32768 x 2 x (91 + 1) = 6.03 MHz
    (N=91)
    FLL_CTL0 = DCOPLUS + XCAP14PF;       // DCO+ set so freq = xtal x D x (N
    + 1)

    // Loop until 32kHz crystal stabilizes
    do
    {
        IFG1 &= ~OFIFG;                   // Clear oscillator fault flag
        //for (j = 50000; j; j--);         // Delay
    }
    while (IFG1 & OFIFG);                 // Test osc fault flag

    // Setup GPIO
    setup_GPIO();

    // Setup LCD

```

```

initLCD_A();

// First amplifier stage - transimpedance configuration
setup_OPAMP1();

// Second amplifier stage
setup_OPAMP2();

// Configure DAC 1 to provide bias for the amplifier
setup_DAC1();

// Configure DAC 0 to provide variable drive to the LEDs
setup_DAC0();

// Set initial values for the LED levels
ir_LED_level = 1300;
vs_LED_level = 1450;

// Configure ADC12
setup_ADC12();

// Configure Timer
setup_Timer();

// Configure USART, so we can report readings to a PC
setup_UART();

while(1)
{
    __bis_SR_register(LPM0_bits + GIE);

    /* Heart Rate Computation */
    f1 = (60*512*3)/log_heart_signal_sample_counter;
    heart_rate = f1;
    // dispChar(0, 1);
    display_number(heart_rate, 3, 3);

    heart_rate_LSB = heart_rate & 0x00FF;

    /* SaO2 Computation */
    x = log_sq_ir_heart_ac_signal/log_heart_signal_sample_counter;
    y = log_sq_vs_heart_ac_signal/log_heart_signal_sample_counter;

```

```

Ratio = (unsigned int) (100.0*log(y)/log(x));
if (Ratio > 66)
    SaO2 = Lookup[Ratio - 66];          // Ratio - 50 (Look-up Table Offset)
- 16 (Ratio offset)
else if (Ratio > 50)
    SaO2 = Lookup[Ratio - 50];        // Ratio - 50 (Look-up Table Offset)
else
    SaO2 = 100;
display_number(SaO2, 7, 3);
SaO2_LSB = SaO2 & 0x00FF;
}
}

/***** End of MAIN function *****/

/***** Timer A0 interrupt service routine *****/
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A0(void)
{
    int i;
    if ((DAC12_OCTL & DAC12OPS))      // Which DAC port is on??
    {
        VIS_on_IR_off();             // turn VIS on and IR off

        ir_sample = ADC12MEM0;        // Read the IR LED results
        i = ADC12MEM1;

        ADC12CTL0 &= ~ENC;           // Enable the next conversion
sequence.
        ADC12CTL0 |= ENC;

        ir_heart_signal = ir_filter(i); // Filter away 50/60Hz and 100/120Hz

                                           // Filter away the large DC
component from the sensor
        ir_heart_ac_signal = ir_heart_signal -
dc_estimator(&ir_2nd_dc_register, ir_heart_signal);

        IR_sig_into_Range();         // bring IR signal into range

        heart_signal_sample_counter++;
        if (pos_edge)
        {

```

```

        loop_pos_edge();
    }
    else
    {
        if (edge_debounce < 120)
        {
            edge_debounce++;
        }
        else
        {
            if (ir_heart_ac_signal > 200)
            {
                IR_heart_ac_sig_200();
                if (++heart_beat_counter >= 3)
                {
                    more_than_3_beats();
                    _BIC_SR_IRQ(LPM0_bits);
                }
            }
        }
    }
}
else
{
    IR_on_VIS_off(); // turn on IR and VIS off

    vs_sample = ADC12MEM0; //Read the visible LED results
    i = ADC12MEM1;

    ADC12CTL0 &= ~ENC; //Enable the next conversion
sequence.
    ADC12CTL0 |= ENC;

    vs_heart_signal = vs_filter(i); // Filter away 50/60Hz and 100/120Hz

// Filter away the large DC
component from the sensor
    vs_heart_ac_signal = vs_heart_signal -
dc_estimator(&vs_2nd_dc_register, vs_heart_signal);

    VIS_sig_into_Range(); // bring VIS signal into range
}

```

```

}
/***** End of TimerA0 Interrupt Service Routine *****/

/***** Start of ADC12 Interrupt Service Routine *****/
#pragma vector=ADC12_VECTOR
__interrupt void ADC12ISR(void)
{
    ADC12IFG &= ~BIT1;           // Clear the ADC12 interrupt flag
    DAC12_0DAT = 0;              // Turn OFF the LED
    DAC12_1DAT = 0;              // Turn OFF the H-Bridge completely
    if(is_IR)                    // If IR LED was ON in TAO ISR
        P2OUT |= BIT2;           // P2.2 = 1
    else                          // Else if VS LED ON in TAO ISR
        P2OUT |= BIT3;           // P2.3 = 1
}

/***** End of ADC12 Interrupt Service Routine *****/

/***** Digital Filter Functions from TI *****/
int16_t ir_filter(int16_t sample)
{
    static int16_t buf[32];
    static int offset = 0;
    int32_t z;
    int i;

    //Filter hard above a few Hertz,
    //using a symmetric FIR.
    //This has benign phase
    //characteristics */

    buf[offset] = sample;
    z = mul16(coeffs[11], buf[(offset - 11) & 0x1F]);
    for (i = 0; i < 11; i++)
        z += mul16(coeffs[i], buf[(offset - i) & 0x1F] + buf[(offset - 22 + i)
& 0x1F]);
    offset = (offset + 1) & 0x1F;
    return z >> 15;
}

int16_t vs_filter(int16_t sample)
{
    static int16_t buf[32];
    static int offset = 0;

```



```

int32_t z;
int i;

//Filter hard above a few Hertz,
//using a symmetric FIR.
//This has benign phase
//characteristics */

buf[offset] = sample;
z = mul16(coeffs[11], buf[(offset - 11) & 0x1F]);
for (i = 0; i < 11; i++)
    z += mul16(coeffs[i], buf[(offset - i) & 0x1F] + buf[(offset - 22 + i)
& 0x1F]);
offset = (offset + 1) & 0x1F;
return z >> 15;
}
/***** End of TI Digital Filters *****/

/***** DC estimator from TI *****/
int16_t dc_estimator(register int32_t *p, register int16_t x)
{
    *p += (((int32_t) x << 16) - *p) >> 9);
    return (*p >> 16);
}
/***** Enf of DC estimator *****/

/***** Setup GPIO *****/
void setup_GPIO(void)
{
    P1DIR = 0xFF;
    P1OUT = 0;
    P2DIR = 0xFF;
    P2DIR |= BIT2 + BIT3; // P2.2 and P2.3 o/p direction -
                          // drives PNP transistors in
H-Bridge
    P2OUT = 0;
    P3DIR = 0xFF;
    P3OUT = 0;
    P4DIR = 0xFF;
    P4OUT = 0;
    P5DIR = 0xFF;
    P5OUT = 0;
    P6OUT = 0;

```

```

}

/***** Setup/Configure ADC12 *****/
void setup_ADC12(void)
{
    ADC12CTL0 &= ~ENC;           // Enable conversions
                                // Turn on the ADC12, and
                                // set the sampling time
    ADC12CTL0 = ADC12ON + MSC + SHT0_4 + REFON + REF2_5V;
    ADC12CTL1 = SHP + SHS_1 + CONSEQ_1; // Use sampling timer, single
sequence,
                                // TA1 trigger(SHS_1), start with
ADC12MEM0
    ADC12MCTL0 = INCH_1 + SREF_1; // ref+=Vref, channel = A1 = OA0
    ADC12MCTL1 = INCH_3 + SREF_1 + EOS; // ref+=Vref, channel = A3 = OA1
    ADC12IE = BIT1;           // ADC12MEM1 interrupt enable
    ADC12CTL0 |= ENC;         // Enable the ADC
    ADC12CTL0 |= ADC12SC;     // Start conversion
}

/***** Setup/Configure Timer *****/
void setup_Timer(void)
{
    TACTL = TASSEL0 + TACLK; // ACLK, clear TAR,
    TACCTL1 = OUTMOD_2;
    TACCTL0 = CCIE;
                                // This gives a sampling rate of
                                // 512sps
    TACCR0 = 9999;           // Do two channels, at
                                // 512sps each.
    TACCR1 = 9979;           // Allow plenty of time for the
                                // signal to become stable before
                                // sampling
    TACTL |= MC_1;           // Timer A on, up mode
}

/***** Setup/Configure OPAMP1 *****/
void setup_OPAMP1(void)
{
    /* First amplifier stage - transimpedance configuration */
    P6SEL |= (BIT0 | BIT1 | BIT2); // Select OA00, -ve=OA0I0, +ve=OA0I1
    OA0CTL0 = OAN_0 | OAP_1 | OAPM_3 | OAADC1;
    OA0CTL1 = 0;
}

```

```

}

/***** Setup/Configure OPAMP2 *****/
void setup_OPAMP2(void)
{
    /* Second amplifier stage */
    P6SEL |= (BIT3 | BIT4);           // Select OA10 OA11, -ve=OA1I0,
+ve=DAC1
                                     // -ve=OA1I0, +ve=DAC1

    OA1CTL0 = OAN_2 + OAP_3 + OAPM_3 + OAADC1;
    OA1CTL1 = OAFBR_7 + OAF_6;       // OA as inv feedback amp, internal
                                     // gain = 15;
}

/***** Setup/Configure DAC1 *****/
void setup_DAC1(void)
{
    /* Configure DAC 1 to provide bias for the amplifier */
    P6SEL |= BIT7;
    DAC12_1CTL = DAC12CALON | DAC12IR | DAC12AMP_7 | DAC12ENC;
    DAC12_1DAT = 0;
}

/***** Setup/Configure DAC0 *****/
void setup_DAC0(void)
{
    /* Configure DAC 0 to provide variable drive to the LEDs */
    DAC12_0CTL = DAC12CALON | DAC12IR | DAC12AMP_7 | DAC12ENC; // VRef+, high
speed/current,
                                     // DAC12OPS=0 => DAC12_0 output on
P6.6 (pin 5)
                                     // DAC12_1 output on P6.7 (pin 10)
                                     // Configure P2.2 and P2.3 to
                                     // provide variable drive to LEDs

    P2OUT |= BIT2;                   // turn off source for tran
    P2OUT &= ~BIT3;                  // turn on source for tran
    DAC12_0DAT = 3340;
}

```

```

/***** Turn on VIS and IR
off*****/
void VIS_on_IR_off(void)
{
    // Immediately enable the visible
    // LED to allow time for the
    // transimpedance amp to settle
    DAC12_OCTL &= ~DAC12ENC;
    P2OUT &= ~BIT3; // turn on source for tran
    DAC12_OCTL &= ~DAC12OPS; // Disable IR LED, enable visible
    LED
    DAC12_OCTL |= DAC12ENC;
    DAC12_0DAT = vs_LED_level;
    DAC12_1DAT = vs_dc_offset; // Load op-amp offset value
    P2OUT |= BIT2; // for visible turn off source for
    tran
    is_IR = 0; // IR LED OFF
}

/***** Turn on IR and VIS off
*****/
void IR_on_VIS_off(void)
{
    //Immediately enable the IR LED,
    //to allow time for the
    //transimpedance amp to settle */
    DAC12_OCTL &= ~DAC12ENC;
    P2OUT &= ~BIT2; //turn on source for tran
    DAC12_OCTL |= DAC12OPS; // Disable visible LED, enable IR
    LED
    DAC12_OCTL |= DAC12ENC;
    DAC12_0DAT = ir_LED_level;
    DAC12_1DAT = ir_dc_offset; // Load op-amp offset value for IR
    P2OUT |= BIT3; //turn off source for tran
    is_IR = 1; // IR LED ON
}

/***** Heart Beat Tracker - if positive edge
*****/
void loop_pos_edge(void)

```

```

{
    if (edge_debounce < 120)
    {
        edge_debounce++;
    }
    else
    {
        if (ir_heart_ac_signal < -200)
        {
            edge_debounce = 0;
            pos_edge = 0;
            display_pulse(0);
        }
    }
}

/***** Heart Beat Tracker - if IR ac sig > 200
*****/
void IR_heart_ac_sig_200(void)
{
    edge_debounce = 0;
    pos_edge = 1;
    display_pulse(1);
    display_correcting(1, 0);
}

/***** Heart Beat Tracker - Heart Beat more than 3 counts
*****/
void more_than_3_beats(void)
{
    log_heart_signal_sample_counter =
heart_signal_sample_counter;
    log_sq_ir_heart_ac_signal = sq_ir_heart_ac_signal;
    log_sq_vs_heart_ac_signal = sq_vs_heart_ac_signal;
    heart_signal_sample_counter = 0;
    sq_ir_heart_ac_signal = 0;
    sq_vs_heart_ac_signal = 0;
    heart_beat_counter = 0;
}

/***** Bring VIS signal into range
*****/
void VIS_sig_into_Range(void)
{

```

```

int i;
i = ADC12MEM1;

if (i >= 4095)
{
    if (vs_dc_offset > 100)
        vs_dc_offset--;
}
else if (i < 100)
{
    if (vs_dc_offset < 4095)
        vs_dc_offset++;
}

sq_vs_heart_ac_signal += (mul16(vs_heart_ac_signal, vs_heart_ac_signal) >>
10);

if (vs_sample > FIRST_STAGE_TARGET_HIGH
    ||
    vs_sample < FIRST_STAGE_TARGET_LOW)
{
    /* We are out of the target range */
    display_correcting(1, 1);
    if (vs_sample > FIRST_STAGE_TARGET_HIGH)
    {
        if (vs_sample >= FIRST_STAGE_TARGET_HIGH_FINE)
            vs_LED_level -= FIRST_STAGE_STEP;
        else
            vs_LED_level -= FIRST_STAGE_FINE_STEP;
        if (vs_LED_level < 0)
            vs_LED_level = 0;
    }
    else
    {
        if (vs_sample < FIRST_STAGE_TARGET_LOW_FINE)
            vs_LED_level += FIRST_STAGE_STEP;
        else
            vs_LED_level += FIRST_STAGE_FINE_STEP;
        if (vs_LED_level > 4095)
            vs_LED_level = 4095;
    }
}
}

```

```

/***** Bring IR signal into range
*****/
void IR_sig_into_Range(void)
{
    int i;
    i = ADC12MEM1;

    if (i >= 4095)
    {
        if (ir_dc_offset > 100)
            ir_dc_offset--;
    }
    else if (i < 100)
    {
        if (ir_dc_offset < 4095)
            ir_dc_offset++;
    }

    sq_ir_heart_ac_signal += (mul16(ir_heart_ac_signal, ir_heart_ac_signal) >>
10);

//Tune the LED intensity to keep
//the signal produced by the first
//stage within our target range.
//We don't really care what the
//exact values from the first
//stage are. They need to be
//quite high, because a weak
//signal will give poor results
//in later stages. However, the
//exact value only has to be
//within the range that can be
//handled properly by the next
//stage. */

    if (ir_sample > FIRST_STAGE_TARGET_HIGH
        ||
        ir_sample < FIRST_STAGE_TARGET_LOW)
    {
        //We are out of the target range
        //Starting moving the LED
        //intensity in the right

```

```

//direction to bring us back
//into range. We use fine steps
//when we are close to the target
//range, and coarser steps when
//we are far away.

if (ir_sample > FIRST_STAGE_TARGET_HIGH)
{
    if (ir_sample >= FIRST_STAGE_TARGET_HIGH_FINE)
        ir_LED_level -= FIRST_STAGE_STEP;
    else
        ir_LED_level -= FIRST_STAGE_FINE_STEP;
        // Clamp to the range of the DAC

    if (ir_LED_level < 0)
        ir_LED_level = 0;
}
else
{
    if (ir_sample < FIRST_STAGE_TARGET_LOW_FINE)
        ir_LED_level += FIRST_STAGE_STEP;
    else
        ir_LED_level += FIRST_STAGE_FINE_STEP;
        // Clamp to the range of the DAC

    if (ir_LED_level > 4095)
        ir_LED_level = 4095;
}
}

/***** LCD number Display Function *****/
void display_number(int value, int start, int width)
{
    unsigned int i;
    unsigned int Output;
    char *pLCD = (char *)&LCDMEM[7-start];

    for (i = 16, Output = 0; i; i--) // BCD Conversion, 16-Bit
    {
        Output = __bcd_add_short(Output, Output);
        if (value & 0x8000)
            Output = __bcd_add_short(Output, 1);
        value <<= 1;
    }
}

```



```

    for (i = 0; i < width; i++)                // Process 4 digits
    {
        *pLCD++ = hex_table[Output & 0x0f];    // Segments to LCD
        Output >>= 4;                          // Process next digit
    }
}

/***** LCD Pulse Display Function *****/
void display_pulse(int on)
{
    if (on)
        LCDMEM[7] |= 0x4;
    else
        LCDMEM[7] &= ~0x4;
}

/***** LCD Correcting info Display Function *****/
void display_correcting(int x, int on)
{
    if (on)
        LCDMEM[3] |= ((x) ? seg_a : seg_d);
    else
        LCDMEM[3] &= ~((x) ? seg_a : seg_d);
}

/***** Configure LCD Function *****/
void set_LCD(void)
{
    /* LCD_A Controller with LCD */
    P5DIR |= 0x1E;    //Ports P5.2, P5.3, and P5.4 as outputs
    P5SEL |= 0x1E;    //Ports P5.2, P5.3, and P5.4 as special function (COM0-COM3)

    LCDAPCTL0 = LCDS24 | LCDS20 | LCDS16 | LCDS12 | LCDS8 | LCDS4; //Enable S4-S25

    /* LCD Operating Frequency */
    LCDACTL = LCDFREQ_64 | LCD4MUX;    //(ACLK =32768)/64 =512 Hz and 4-MUX LCD

    /* LCD_A Configuration */
    LCDACTL |= LCDSON | LCDON;        // LCD_A on and segments are on
}

```