Contents

List of Figures

List of Tables

1.0 Executive Summary

Currently, the optimal procedure to measure blood glucose concentration is through invasive blood sampling. Diabetics must test their blood multiple times daily by pricking their fingers, placing a drop on a test strip, and inserting the sample into a glucose meter. From a chemical reaction that releases a small electric current, the glucose meter estimates the concentration. Although this method is widespread and efficient, modern advances in technology should allow us to overcome the standard measurement method's four major drawbacks: invasiveness, cost, hygiene, and pain. With these problems in mind, in this study we develop a noninvasive, blood glucose measurement system and study its preliminary results and feasibility for future work. We plan to implement near-infrared LED absorbance spectroscopy and measure body parameters such as tissue thickness and blood oxygen saturation to aid in our glucose measurement model. The aim of this project is to provide a proof-of-concept that glucose levels can be measured in a non-invasive way using near infrared light. It is hypothesized that by adding physiological parameters, such as pulse oximetry and tissue thickness, a more accurate glucose measurement can be made.

In this project, near-infrared transmittance spectroscopy is used across the ear lobe to measure glucose. In this setup, a light source and light detector are positioned on either side of the ear lobe, and the amount of light passing through depends on the amount of glucose in that region. The ear lobe was chosen as the testing location because of its relatively small thickness, absence of muscle and bone, and absence of many other interfering biomaterials. While the NIR light is applied onto one side of the earlobe, the receiver on the opposite side receives attenuated light. This signal is then sampled and processed. An LED emitting light at 1550nm will be used as the light source, and an Indium Gallium Arsenide photodiode with a high response in the 1550nm region will be used for detection. In addition to realizing the NIR transmittance pertaining to glucose, it is also beneficial to know the amount of blood in the path of the light. In other words, for the same glucose concentration, a larger amount of blood will result in a lower NIR light transmittance, and a smaller amount of blood will give a higher NIR light transmittance. To control this factor, the amount of blood can be estimated by measuring blood oxygen levels via pulse oximetry. In this project, the AFE-4404 chip provided by Texas Instruments is used to transmit red and infrared light to distinguish between hemoglobin and oxyhemoglobin in the blood, on which further processing is applied to calculate the oxygen saturation, which is strongly correlated to the total amount of blood present.

Moreover, the transmittance of the NIR light also depends on the thickness of the ear lobe. To make the same device adaptable to multiple people whose ear lobes may vary in thickness, we propose to determine the path length of the NIR light by using a green light. Green light has high skin attenuation and will provide information regarding the ear lobe thickness. In developing the prototype, the aim is to provide the proof-of-concept to work accurately on test tubes. Experiments on human ear lobes will be conducted to the best possible; however, highly accurate results on human subjects is out of the scope of this work and will be explored in future work.

With the advancement of technologies such as analog to digital converters, microprocessors, and wireless technologies it is feasible, and in many testing environments shown to work, to build a noninvasive glucometer. Our method of choice is to utilize a technique that emulates near infrared spectroscopy using a broad LED light source at a frequency that is shown to be absorbed by the glucose molecule. Utilizing highly sensitive ADC electronics to correlate voltages given from a photodiode sensitive to the desired frequency band we will emulate a spectroscopy by detecting the absorption emissions. This technique is not satisfying by itself therefore the method requires other data inputs to increase the reliability of the measurement process. The technique will measure three other parameters before doing the spectroscopy, which are skin thickness, blood oxygen concentration through a pulse oximeter, and heart rate monitoring. Texas Instruments has sponsored our group and provided the AFE4044 analog front-end chip for testing pulse oximetry and optical heart rate monitoring. The reasoning for doing these tests is to improve reliability of the spectroscopy. The measurement of skin thickness will be used to determine the overall light attenuation through biological tissue. Next using a pulse oximeter our setup will be able to determine how much blood actually exists at the testing site and this is integral for determining when to test for glucose, and the optical heart rate monitor is important for detecting any unwanted deviations or noise. Finally after these first three parameters are ideal the device will utilize its spectroscopy method for the detection of Glucose.

2.0 Project Description

This section serves to provide the motivation for developing a glucose detector as well as to portray the goals and objectives of this project. The idea of detecting glucose with LEDs will be discussed to help the reader understand the overall concept of detecting glucose with LEDs. Finally, the projects requirement specifications will be presented to meet the goals and objectives.

2.1 Motivation

In the American population with a sample size of 29.1 million people, 9.3% of those are diagnosed with diabetes. [1] What is even more alarming is that in the group of people aged over 65 25.9% of those people are also diagnosed with diabetes. Not to mention that this rate is also increasing with 1.4 million Americans diagnosed ever year [1]. It is not hard to find a family or friend who is afflicted by this disease, and the current method to regulate and measure it requires a form of electrochemical or amperometry. These methods are efficient and have been used for many years but require a form of blood sampling that is invasive. Blood glucose monitors are used to measure the amount of glucose in blood. By measuring, the glucose levels in the blood diabetes patients are then able to determine when and how much insulin should be taken to reduce high levels of glucose. Each test also requires a new test-strip, contributing to the recurring cost of the blood monitors. The blood sampling is used with a needle which can leave you prone to infection and is not the most convenient thing to do. Constantly having to pick at the skin to test for blood also causes diabetes patients to feel uncomfortable or insecure when having to do it in public. Many different ideas have been tried to make checking for glucose levels much more comfortable. Although ideas have been tried, there is not a product in the market today that can be purchased and be used to detect glucose levels.

To better understand the treatment of the disease we must look into the history of diabetes and how in cultures past could identify those who had the disease and how they treated the disease. It is believed that the first documentation of diabetes comes from ancient Egypt where a physician described the disease as, “urination as a symptom of a mysterious disease that also caused emaciation,” and near the same time period in the far east it was discovered that those who had the disease had urine that attracted ants. [2] Then in later centuries came “water tasters” who would taste people’s urine and if the urine was sweet the patient would be diagnosed with diabetes. These diagnostic methods lead to the creation of the term Diabetes Mellitus. ‘Mellitus’ which means honey and ‘diabetes’ which means to siphon. It was not until circa 1700s and 1800s that physicians Figured out that the disease could be managed through diet. Then in the 1800s, chemical tests were developed to detect the presence of the disease. [2] The techniques that were developed throughout the 1800s and early 1900s that advised for certain diets and exercise are still being used today. In 1889, two French researchers discovered that when they removed the pancreas from a dog that they could induce diabetes and in the early 1900s a German researcher discovered that by injecting the extract created in the pancreas one could control diabetes. [2]

As history has told Diabetes Mellitus is a disease that has affected populations present and past. Although nowadays sufferers are more equipped to effectively manage the disease using a variety of techniques such as exercise, proper diet, and medication, the disease is still lethal and requires constant monitoring. With patients constantly testing their blood glucose concentration using glucometers they become more informed on what and when they should consume food and medication. Doctors are constantly trying to inform their patients on how to effectively manage their health. Using technologies such as the internet and smartphones that are capable of communicating quickly it is possible for doctors to be more informed on how well their patients are performing. Instead of glucometers just being able to tell the patient concentrations they are now able to be used in conjunction with software to log data, remind patients of when to test or take medicine, give health tips, send the logged data to their health care providers, and more. The integration of these technologies with the past technology of the glucometer has been proven to help patients and doctors be more informed, and the dispersion of information leads to better decision making when it comes to managing health. Despite the advancements in technology it is still difficult to get patients to comply with the regiment of testing, often diabetics will not test themselves because they loathe the lancing and anticipation of having to draw blood. Although this is not the only motivation for noninvasive testing, it is certainly a desired testing method and would help patients feel more comfortable with the process

Our goal for this project will be to provide a proof of concept that glucose levels can be measured in an inexpensive, and novel way. By showing a different method to measuring glucose then more attention can be geared towards developing an FDA approved method of glucose detection. This device can immensely impact a patient’s life forever.

2.2 Goals/Objectives

In designing the project, our main goal and objective is to create a portable glucose detector. The name of our device will be referred to GOLD (Glucose Optical LED Detector) The GOLD device shall implement at least one method of checking glucose levels such as a method described by low infrared spectroscopy to determine blood glucose levels based on transmittance spectroscopy on the ear lobe. We will originally aim to detect glucose in a water solution to show proof of concept. If this is a method we will implement, then we will shoot a near infrared led through the skin. The glucose molecules within the blood will absorb the correct wavelength of light. The light not absorbed will then be reflected back and measured by a receiver. The difference in light emitted and reflected will constitute as the parameter that will predict the amount of glucose present within the blood or water based solution. This information will then be sent to an android phone via Bluetooth. The phone will have an application that will display the glucose level information.

Another goal we have is to try to obtain the most accurate result possible. We initially hope to obtain at least a spearman's correlation of 0.70 the blood glucose levels and greater accuracy for the water-based solution. The purpose of our accuracy standards is to try to improve other current ideas that have been developed. Discussions on other noninvasive methods will be discussed later on, including the importance of accuracy.

In order for our device to compete with current technology it will need to have a user-friendly application. The application will be have software that can be show glucose levels overtime. By saving the information of the glucose levels, then it will make it easier for the patient to keep track. This application will also be secured with a password to keep patient information confidential. Since our project requires us to have connectivity with a smartphone, the algorithm for predicting the amount of glucose present in the blood will be processed on the smartphone, and the results of this calculation will be relayed to the user there as well. The means of communication between the smartphone and the sensor will be carried out using wireless communication.

The LED will operate in the infrared light spectrum between 1550nm to 1600nm. The goal is to be able to overcome the challenge, in this wavelength, of detecting specific features for each chemical compound. Our objective will be to create an algorithm that will provide us with the best estimation equation. This equation will be used both for the skin and water based solution.

We will also use the AFE4404 chip to communicate with light emitting diodes (LED) and a photodiode. This chip will measure oximetry and heart pulse rate. The oximetry measurements will be used to measure the amount of blood that is present. The LEDs and the photodiode connect to their respective nodes on the analog front-end (AFE) and they simultaneously operate to send and receive infrared wavelengths, ultimately obtaining an analog value that will be manipulated to display blood glucose levels. In order to achieve maximum performance from the analog front-end, radiant LEDs and highly sensitive photodiodes must be used but also ambient lighting can affect results so this must be reduced to increase the success of the photodiode.

Another goal set is to try to keep the device as small and compact as possible. The current invasive glucose monitors are small and are able to be carried in an average size pocket or purse. To keep this goal we must use small batteries but the batteries must be able to power our LEDs and photodiodes. This will be a key feature of our product. Our device will also include a power switch in order to save battery life. It is important for battery life to last as long as possible. Low powered devices will be used to achieve this goal.

Our goal is to make the device as simple as possible for patients to use, but allows users access to data through their computing devices. The interface will be a few pushbuttons on the actual device with LEDs to indicate which mode it is operating in, such as an indicator for the device being ready to sense, when the batteries are low, and when the device is sending data and when data transfer is complete. The main user interface will be through the application, which will display results and do other added functions.

2.3 Glucose Detection

Understanding how an LED can measure glucose is important. Apart from the level of glucose in blood, the transmittance of near infrared light also depends on the amount of blood in the path of the light. A large amount of blood will result in lower transmittance, whereas less blood will result in a larger transmittance. The glucose value needs to be scaled according to the amount of blood residing inside the ear lobe at a time of measurement. The amount of blood can be estimated by measuring the blood oxygen levels. Pulse Oximetry was used to measure blood oxygen. Pulse Oximetry uses Red and Infrared (IR) light to distinguish between Hemoglobin and Oxy-Hemoglobin in the blood, on which further processing is applied to get the oxygen saturation. Another physical parameter that affects the glucose measurement is the ear lobe tissue thickness. This is an issue when the same device is used by more than one person, in which case the ear lobe thickness could be different for each. Tissue thickness determines the ‘path length’ of NIR, so a greater path length would result in lower transmittance. Tissue thickness was measured using green light, which has high skin based attenuation.

Also understanding how an invasive glucose meter works. The skin is first pricked and the blood is then set on the test strip. This test strip goes inside the device, which holds optical, and electrochemical sensors. The signal of these sensors are then amplified and sent through a mux with logic gates. Finally this signal is sent to analog digital converters and to digital analog converters. Once the signal is processed it is sent to the microprocessing unit for further processing to be displayed on the LCD.

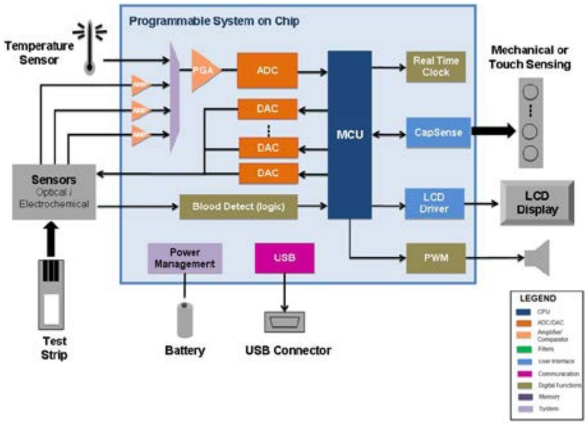


Figure 1: Invasive Glucose Detection (Courtesy of Texas Instruments)

Our device will mimic the current invasive methods but will have a different input. The input of our device would simply be what the LED detected from the skin or water based solution. As seen in Figure 2 the ear lobe will be used for the sensors to get information from. This signal will then go through an RC filter to remove any noise then amplified by an analog to digital converter. Once the signal is converted to digital then it is processed by the 32 bit arm and sent to the display unit.

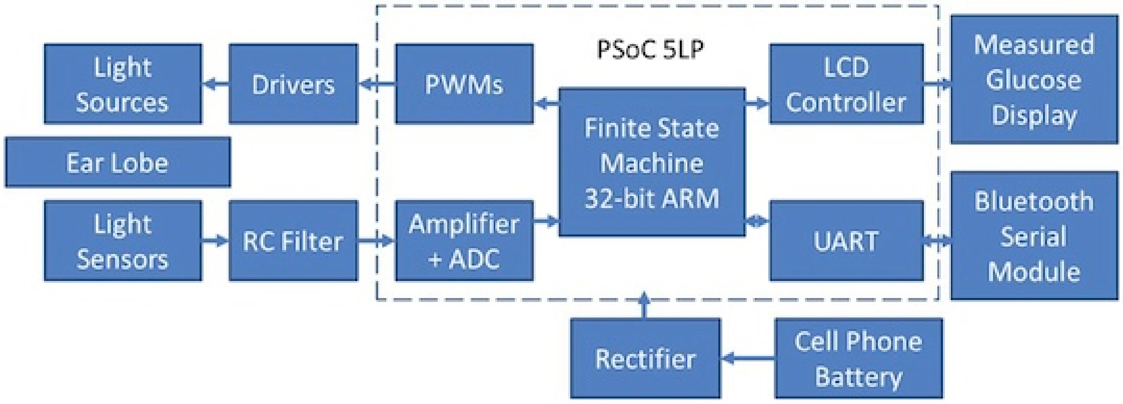


Figure 2: Noninvasive Glucose Detection (Courtesy of EDN Article)

The process of detecting glucose noninvasively by an LED brings many factors that can affect the glucose level results. By using different sensors such as, oximetry, thickness, and glucose the overall equation can be modified to be more accurate.

2.4 Requirements Specifications

The Glucose optical LED detector will have multiple requirements to be met to ensure accuracy, feasibility, and other core traits about its functionality. To increase user friendliness the design must be able to achieve a number of tasks within a certain time limit, utilizing communication methods, be able to last for extended periods of time, and be ergonomic and easy to use, to list just a few. As far as the manufacturing of the product it must be able to meet certain design needs while not being to expensive to produce the product therefore there are some tradeoffs that must be considered. These factors will determine whether or not the product can be successful in the marketplace and compete with products that have existed and dominated the market share. Our requirement specifications can be broken into multiple categories as such, functional, environmental, health and safety, physical, user interface, economic, and power consumption.

Our functional requirements include, being able to detect glucose concentration in real time, and in our case real time would be considered to be three minutes or less. Our system should be able to determine concentration to about 75% when tested properly. The start up time for the system to boot should be two minutes or less, and the system should include an interface of LED indicators. Integration to a mobile application is also required for the system requirements and our system will use less than 16MB of ram and less than 500MB of flash memory, which is important for reducing costs of components. These requirements are chosen to allow the user to be able use the device easily, but also so the user does not use too much time waiting for results, which will be queued in the mobile application.

The Environmental, Health and Safety requirements include that the system uses a low powered lasers limited to classes 1 and 1M which are considered safe with biological tissues.

Another requirement in this category is that the system does not overheat and cause to much discomfort to the patient, or maintaining temperatures under 90 degrees Fahrenheit in the system's most extreme operating modes. The last requirement is that the products used to create the enclosure, clamps, or any other casing that comes into contact with the user's skin is considered biocompatible or non-hazardous, to ensure safety of the user during prolonged use with the product. The physical requirements that will be implemented are to ensure portability. The volume of the device should be able to be stored in either the user's pocket, or carrying case, and the weigh should be small so that the device is not difficult to use. Our goals for the physical requirements is that the device shall not weigh more than 1.5 kilograms, and that the xyz dimensions is limited to 6 inches for each side respectively. The earpiece or finger clamp will be small enough to put in one's hand and will weigh less than 0.5 kilograms. The wire that will connect the external module and main modules will extend to be less than 3 feet, but that is up to the user's discretion. Requirements for the user interface will be limited on the main module and mainly done using the graphical user interface on the mobile application. The user interface for the hardware unit will be limited to a couple of buttons, one for power on/off, and another to begin the testing. The device will utilize an array of LED to indicate a number of different instances such as, when testing is occurring, when the device is ready to begin testing, once the device finish the test, low battery warning, and when the device is powered on. The main functionality for the interface is to allow the user a simplistic way to use the device and all of the user customization or data representation will be done on the smart phone.

Our goal for the economic design of our system is to limit the cost of production to $300 which will put our device into a reasonable price range to compete with similar devices on the market. While the device should not cost over $300 we will adhere to limiting our budget to $1000 and be as cost effective as possible while building the device. Our requirements for our power systems while not including power consumption due to data processing on the mobile phone will be limited to the hardware module produced by the Computer and Electrical engineers. Our device should not function on voltages greater than 5 volts, and should be able to run for nearly an hour when in testing mode. The device should also come complete with a low power mode that can be used for modes that might need idling for systematic alarms. The low power mode should be able to stay on standby for close to 5 hours, and should consume significantly less amounts of power than when in test mode. The design for the battery system can use different types of batteries, but ideally an easily accessible battery type such as AA or AAA, and should be limited to less than four batteries.

3.0 Research

In this section we will discuss research regarding noninvasive glucose sensing. The research that will be discussed below will range in topics from mobile applications, techniques for sensing, existing techniques, upcoming techniques that are being proposed, and etc. The subject of Noninvasive Glucose has not been perfected yet and therefore is being heavily researched. Our sources for research will be from many different higher education institutions, research articles, and any other supporting materials. By having many references on techniques that exist and the technique that will be implemented it will be easier to design our product according to medical standards. This will section serve to be an introduction to each area of design that must be taken to account and will be used to help create a new improved design.

3.1 Existing Products

The market for glucose meters is unfortunately a very reliable market, so it is not surprising that there are a few glucose sensors currently on the market. A few different glucose meters will be analyzed both invasive and noninvasive. In the present time there have been progressions in the market place including “alternate site testing” which uses blood from somewhere besides the fingertip to aid the pain. Additionally, systems that contain multiple test strips or systems that have application interaction are now becoming significantly more common than before. Hypoglycemia meaning glucose level is low can kill someone in less than an hour so some glucose monitors are continuous (CGM). This requires a sensor that is implanted in your skin, which has a lot of occlusion, and we can hopefully solve this problem. Insulin pumps are connected in the body, which leaves the user inhibited in a relative sense. Bio-implants leave the user susceptible to diseases but they are very common in the marketplace. The three glucose meter brands we will look at is Medtronic Diabetes, Amazon’s bestseller in Blood Lancets Active1st Bayer Contour 2, and Diabetes Sentry Hypoglycemic-Symptom Alarm.

3.1.1 Medtronic Diabetes

The way the CGM monitor works is that insulin is attached to your belt or placed in your pocket that connects to a CGM transmitter. This transmitter is the processor of the board that communicates with the pump that checks yours tissue fluid. The sensor has a sticky patch to hold it in place and is replaced often [3]. A needle is injected into with the sticky patch and worn commonly on the abdomen. It is notable that you still have to prick your finger once every 12 hours to calibrate the sensor but it reduces the measurements [4] and the injections of the insulin pump. Figure 3 above shows a minimized system. These systems are very expensive; in fact, the prices are not easy to find but they can range for over ten thousand dollars.

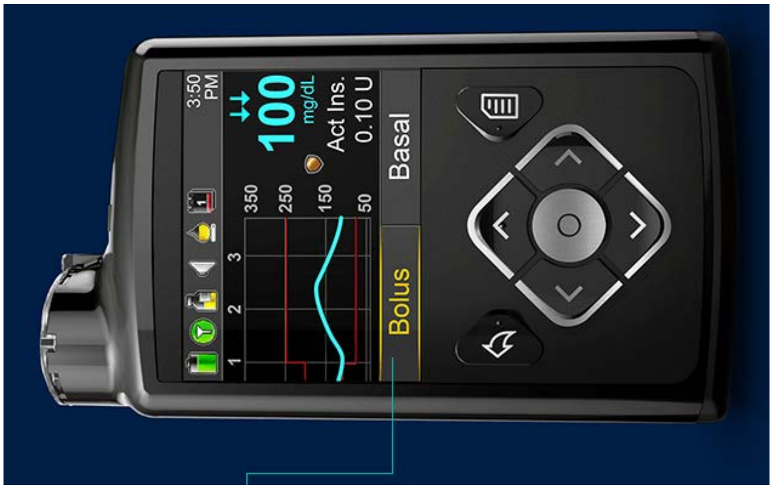


Figure 3: CGM monitor (Courtesy of Medtronic)

Medtronic diabetes is a line of continuous glucose monitors that is according to its own website the #1 prescribed pump brand. This device not only measures glucose levels but it is also a pump. One of the main selling points on its website is its SMARTGUARD™ TECHNOLOGY that “takes action for you if your sensor glucose values go below a preset level”. Essentially this device is an insulin pump and meter that has a wireless interface that shows visual feedback of your blood levels [3]. Although it is invasive so the user guide does warn of bleeding, discomfort or pain as well as localized infection and many more. An important feature that Medtronic implements with its enlite sensor is that it will alert the user progressively louder if the glucose levels are falling below normal. This feature is very useful and can potentially save lives. The alerts on this device get progressively louder so if you are sleeping it will wake you up. A noninvasive measurement cannot do this useful function as easily. You can see in the above Figure 3 that this system manages to make a glucose meter look like a cool and glamorous piece of technology almost representing an mp3 player. This Figure also shows a portion of the user interface that looks accessible and extremely user friendly, which is important due to alarming number of elders with diabetes that are afraid of technology. The pump shown below is a relatively small mechanical system but relatively large compared to the amount of occlusion it encompasses.

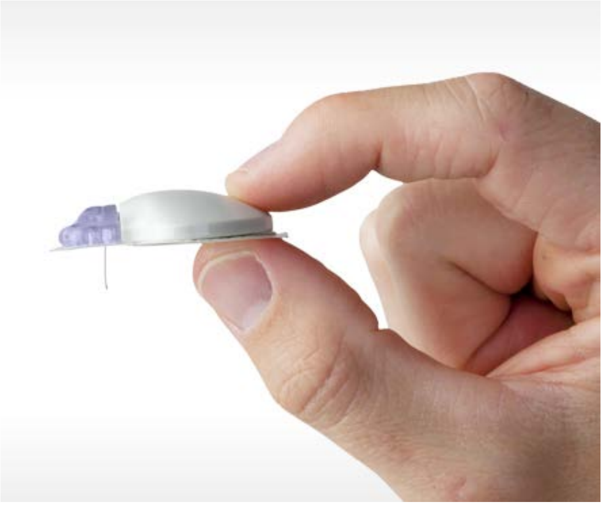


Figure 4: Insulin Pump courtesy of Medtronic

3.1.2 Active1st Bayer Contour Next complete Diabetes Testing Kit

This testing kit goes for $39.95 on amazon prime with 100 test strips. This is significantly cheaper than the Medtronic but it is also not a constant glucose meter. Also, this device does not have an interactive UI but it does display your glucose level for a very in expensive price. This kit is a straightforward example of an invasive method that is easy to come by and is complete with all things in order to run out of the gate with it. As one can see, the market for this is very well defined.

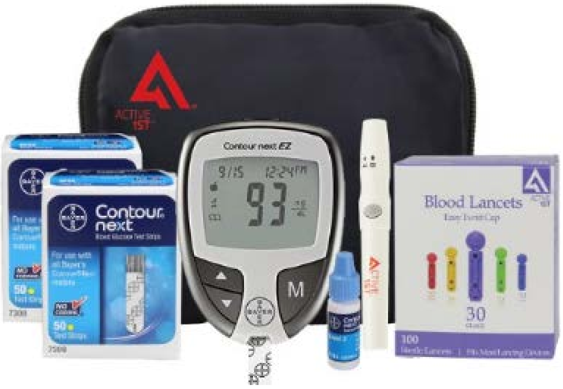


Figure 5: Bayer Amazon Diabetes Kit courtesy of Medtronic

3.1.3 Sentry Hypoglycemic-symptom Alarm

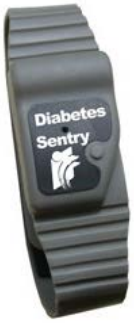
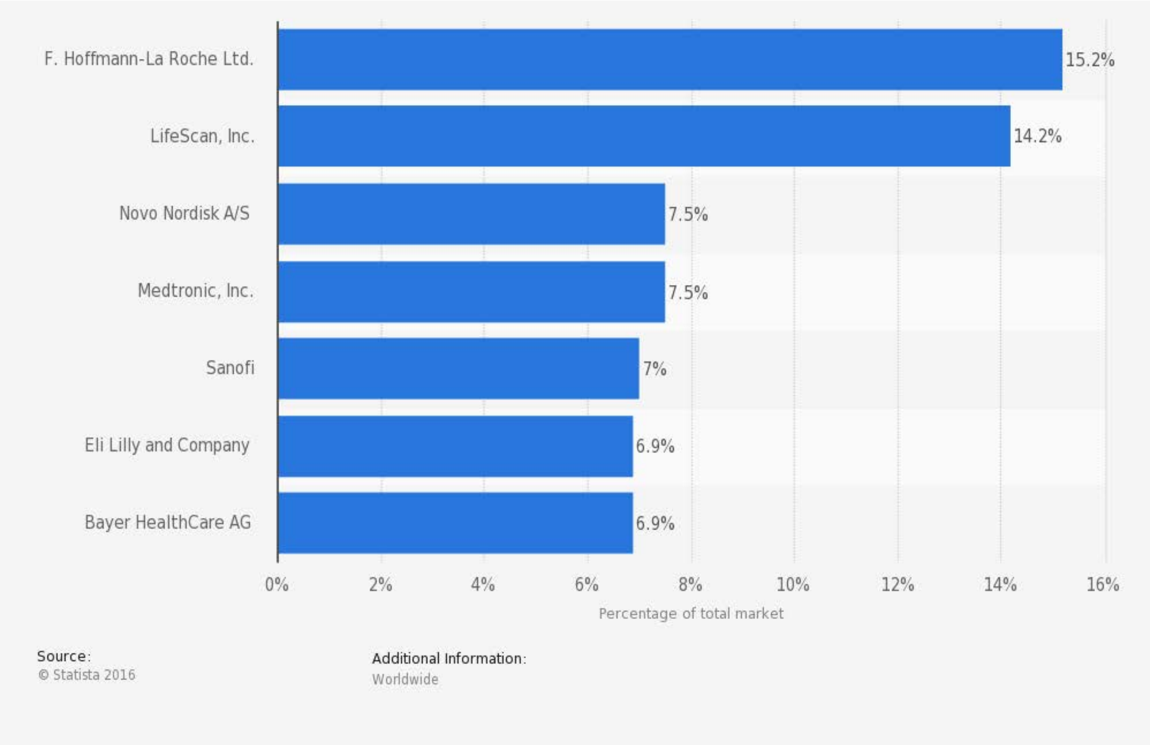


Figure 6: Sentry Hypoglycemic-symptom Alarm (courtesy of Diabetes Sentry)

This product advertises itself as an alarm that reduces the amount of emergency level hypoglycemic attacks that occur at night which make up 80% of these diabetic attacks. This brand poses as a simple and cost efficient solution [5]. It is worn on the wristwatch and checks for two symptoms skin perspiration and a drop in skin temperature. It is surprising to think that people can get away with charging five hundred dollars for a sensor that checks for temperature and skin perspiration but not when you compare CGM that cost upwards to ten thousand dollars. At that rate, this product almost seems like a good idea. Not only that, but also this is the only FDA approved noninvasive glucose meter and it doesn't even have an oximeter. The website claims that 90% of people with diabetes at least show one of the two symptoms. The device has a cell battery that last six months and only has one on and off button and supports 4 audible beeps. This is the only FDA approved noninvasive product on the market and it doesn't even do anything significantly hard. We propose a few more systems then this at almost a fifth of the price with app integration and this is our only competing product.

3.2 Market Analysis

Today those who struggle with diabetes have to monitor themselves daily to maintain their sugar levels. The typical method of detection that dominates the market are glucometers and they are key to at-home monitoring. Glucometers are particularly useful in helping keep patients informed of their sugar levels by allowing them to test themselves throughout the day and this can lead to the prevention serious health implications caused by both hypoglycemia and hyperglycemia. The principle of operation is a, "biosensor based on a thin layer of glucose oxidase (GOx) on an oxygen electrode, and the readout was the amount of oxygen consumed by GOx during the enzymatic reaction with the substrate glucose." [4] This technology was first presented by Leland C. Clark Jr. in 1956 and in 1962 Clark and Ann Lyon created the first glucose enzyme electrode. [6] During the early 1980s the first commercial at home glucometers became available and the market was dominated by Bayer and AccuCheck. [7] Figure 7 describes the overall market share by each company in the Glucose monitoring sector.



Source:
© Statista 2016

Additional Information:
Worldwide

Figure 7: Market Share of the Global top Diabetes care devices manufacturers in 2011 [8]

Glucometers come either utilizing a color test strip in which the patient can use a color chart to match the chemical color change, a non-meter test strip, or a meter that uses a display to show the concentration of glucose in the blood. As with improvement in the meter accuracy and insurance coverage improved came the phasing out of the non meter test strips. The qualities of today’s glucometers are they can fit within the palm of the hand, are battery powered, utilize a digital display to show the concentration in either mg/dl or mmol/l, use between 0.3 to 1 μl of blood for testing, test results occur between 3 to 60 seconds, use test strips impregnated with the chemical enzyme such as glucose oxidase, come with an adjustable lancing mechanism, allow for alternate testing sites than the fingertips, have an internal clocking mechanism set by the user for setting dates/times, internal memory for storage of previous test results, and the average cost of the meter is between $20-80 depending on added functionality. [9] Recently the integration of mobile applications has lead to glucometers with the capability of data transfer for sending results to the patients’ smartphone or computer. The cost of a glucometer is not what is expensive, it is actually the recurring expense of the testing strips and the added cost of lancing blades. “Test strips can run from 40 cents to well over $1 apiece, and people often question why they cost so much.” [10] Often diabetics are recommended to self test multiple times a day, typically between three to ten times, and have to use a new test strip each time therefore costs can add up quickly. Some models of glucometers require a process called ‘coding’ in which requires patients to manually enter in a code found on the vial of test strips or on a chip that comes with the test strip. This process will calibrate the glucometer to the specific test strip and done incorrectly can lead to errors in tests with measurement errors of substantial amounts. These errors are critical for patients that need great accuracy in measurement, and this inconvenience has lead companies such as OneTouch to create test strips that have a standardized code, thus removing the ‘coding’ process. Another differentiating feature is whether the glucometer tests whole blood or blood plasma, which glucose levels in blood plasma are higher than in whole blood. Some glucometers utilize equations built within the system to measure blood plasma glucose levels while testing whole blood. The glucometer is a great option for those coping with the disease from home however doctors report patient noncompliance.

Noncompliance is defined as patients not taking the right actions to properly controlling the disease. Often patients, young and old, state that they do not want to monitor their levels because of either anxiety, fear, or the inconveniences of lancing and using the glucometers, despite obvious health implications. As a research shows, “Medication noncompliance and clinic nonattendance, assessed during routine care by primary care physicians or their staff, were independently associated with increased all-cause mortality in patients with type 2 diabetes receiving insulin.” [11] A solution for monitoring levels is to use a continuous monitoring system. The continuous monitoring systems often operate by inserting a disposable sensor under the skin, a transmitter connected to the sensor and a reader that receives and displays the measurements. Typically these sensors need to be replaced after several days. “The devices provide real-time measurements, and reduce the need for fingerpick testing of glucose levels. A drawback is that the meters are not as accurate because the read the glucose levels in the interstitial fluid which lags behind the levels in the blood.” [12] [13]These types of monitors “themselves cost from about $1000 to $1400” and “depending on the model, these may need to be changed once a year or so and can cost around $500.” [14] Also the sensors are not cheap costing around $35-100 apiece and have to be changed every 3 to 7 days. [14] Despite all this, continuous monitors also require you carry a pager-sized box around with you at all times. The advantages and disadvantages are quite clear however, it all varies on patient preferences and need therefore which is better is up to debate. This has led researchers and engineers towards noninvasive solutions to monitor blood levels.

Noninvasive glucose monitors have been sought after since the mid-1970s however to date only few products exist on the market. “As of 1999, only one such product had ever been approved for sale by the FDA, and it was withdrawn after a short time owing to poor performance and occasional damage to the skin of users.” [15] A solution that integrates both the smartphone and a blood glucose meter is currently under heavy research since the biosensors create added capabilities to that of the phone. The U.S. FDA has approved one noninvasive glucose meter: The GlucoWatch G2 Biographer made by Cygnus Inc. Despite getting approval, the main reasons for its market failure were due to its difficulties in creating reproducible results and its inability to compare to conventional glucometers.

Reasons for failure were that Cygnus lacked an effective care management group, which made it difficult for healthcare organizations to provide the device to beneficiary's sustainable costs. Cygnus, also unable to demonstrate cost benefits while not having an effective marketing strategy, which is important for educating health care providers. With failing to have profits in its 17 years of existence, the company has struggled to maintain strong market share. With products was unable to deal with issues, such as perspiration at the testing site and weak backing from health insurance Cygnus dissolved.

A new product is in development by a company called Glucosense which its, "non-invasive technology will be configured as a small, portable device using a low-powered laser sensor to measure blood glucose levels within 30 seconds." This product is being modified so that it can provide continuous measurement as well bridging the gap while being able to send data wirelessly to the phone. The many issue with these noninvasive sensors is that they must meet an accuracy made by the International Organization of Standards, "ISO 15197 stating that Blood glucose meters must provide results that are within 20% of a laboratory standard 95% of the time (for concentrations about 75 mg/dL.)" There are other products that exist on the market such as The Diabetes Sentry which is priced at $450 that is not a glucose meter but an alarm that notifies the patient when symptoms of hypo or hyperglycemia occur. Another company called GlucoTrack is currently in clinical trials potential moving towards regulatory approval in the U.S. [16]

3.3 Relevant Technologies

Although there are not any products in the market at the moment that detect glucose the same way we are proposing we can still discuss aspects that can apply to all methods. This section will cover the research done in each particular method that we will be using in for our device. The different technology that will be used in the project is power, microcontroller, mobile application and sensors. It is important to discuss each one of these sections since it allows for a better of understanding of what is in the market. The best options from the market have to be chosen in order to make this project successful. Examining the different current technologies will allow GOLD to have the fastest and latest technology for best performance.

3.3.1 Power Sub-system

This section will go into detail to cover all the different methods of technologies that relate to certain power aspects. The first one discussed will be the different type of DC-DC converters and the second section will discuss the different type of batteries researched.

3.3.1.1 DC-DC Conversion

This section will discuss the DC-to-DC method and how it may be useful. DC- to-DC conversion converts one voltage level to another. The DC-to-DC conversion will happen between the battery and the load to allow the battery to provide the required power to the load. The three different forms of DC-to-DC converter will be linear regulator, switching regulator, and switching control. The three different methods are shown in Table 1. Once these methods are compared then an analysis on the switching modes will be discussed and compared.

Most DC-to-DC converters are designed to move power in only one direction, from dedicated input to output. DC-to-DC converters are available as integrated circuits requiring few additional components to connect them with. Converters are also available as complete hybrid circuit modules, ready for use within an electronic assembly. For our project, we believe an electronic assembly will be the best option

Switched-mode DC-to-DC converters convert one DC voltage level to another, which may be higher or lower, by storing the input energy temporarily and then releasing that energy to the output at a different voltage. This method can be classified as an electric power converter. Practical electronic converters use switching techniques. The storage may be in either magnetic field storage components (inductors, transformers) or electric field storage components (capacitors). This conversion method can increase or decrease voltage. A switching regulator uses a power switch, an inductor, and a diode to transfer energy from input to output. These different components can be arranged to form a step-down converter (buck), a step-up converter (boost), or an inverter (fly back). The buck–boost converter is a type of DC-to-DC converter that has an output voltage magnitude that is either greater than or less than the input voltage magnitude. It is equivalent to a fly back converter using a single inductor instead of a transformer. Two different topologies are called buck–boost converter. Both of them can produce a range of output voltages, ranging from much larger (in absolute magnitude) than the input voltage, down to almost zero.

Switching regulators provide efficient switching time and less energy loss. Most importantly, a switching regulator can store energy in the inductor, which can then be used for a greater input voltage value or lower. Switching controllers complete the switching function externally of the chip. This allows for much higher currents than switching regulators, since the controllers themselves do not have to handle the current. The circuit for a switching controller is more complex than that of a switching regulator.

Switched capacitor converters rely on alternately connecting capacitors to the input and output in differing topologies. For example, a switched-capacitor reducing converter might charge two capacitors in series and then discharge them in parallel. This would produce the same output power (less that lost to efficiency of under 100%) at, ideally, half the input voltage and twice the current. Because they operate on discrete quantities of charge, these are also sometimes referred to as charge pump converters. [17] They are typically used in applications requiring relatively small currents, as at higher currents the increased efficiency and smaller size of switch-mode converters makes them a better choice. They are also used at extremely high voltages, as magnetics would break down at such voltages.

A linear regulator is used to maintain a steady voltage in a circuit. Regulator resistance varies the voltage. Linear voltage regulators have been used in the industry for many years. It was the main method used for power supplies until the switching methods were invented. Linear regulators are still used today regardless. For our project, it was crucial to research this method as well to make an educated decision of what converter we should use. A drawback that comes with linear regulators is the amount of power that is consumed. Switching conversion is more power efficient (often 75% to 98%) than linear voltage regulation, which dissipates unwanted power as heat.

In Table 1, the different methods of DC/DC converter are compared. This comparison is useful for when deciding what kind of DC/DC converter should be used for our device.

Table 1: DC/DC Comparison Table

	Linear Regulator	Switching Regulator	Switching Controller
High Efficiency Region	Narrow	Wide	Wide
Component Count	Low	Moderate	High
Output Power	Low	Low to Mid	High
Design Effort	Low	Moderate	High
PCB Layout Effort	Low	Moderate	High

After completing the research on DC-to-DC converter, the conclusion came that a switching controller would provide the best use for our project. The different switching controlling methods will now be discussed.

A buck converter has a high efficiency and minimal loss of heat. A drawback from a buck converter is that it always needs greater input voltage than output. A bust-buck converter is capable of reduction and increase in voltage output. With these capabilities comes more components there taking away from the board.

3.3.1.2 Battery

This section will discuss the different types of batteries including the chemical material that makes them up. Coin lithium, silver oxide, mini alkaline button cell batteries will be discussed and compared. The button cell battery and AA battery will be discussed for size purposes.

A button cell is a small single cell battery shaped as a squat cylinder typically 5 to 25 mm in diameter and 1 to 6 mm high. Button cells are single cells, usually disposable primary cells. Common anode materials are zinc or lithium. Common cathode materials are manganese dioxide, silver oxide, carbon monofluoride, cupric oxide or oxygen from the air. Mercuric oxide button cells were formerly common, but are no longer available due to the toxicity and environmental effects of mercury. Cells of different chemical composition made in the same size are mechanically interchangeable. However, the composition can affect service life and voltage stability. Using the wrong cell may lead to short life or improper operation. Sometimes different cells of the same type and size and specified capacity in milliamps per hour are optimized for different loads by using different electrolytes, so that one may have longer service life, than the other if supplying a relatively high current.

Lithium cells are often called coin cells due to their shape. [18] The lithium cells rated at 3.0 volts. The other type of lithium cells commonly seen is Lithium-Thionyl Chloride, which are rated at 3.6 or 3.7 volts. An advantage of this battery is that the voltage is fairly constant over most of its useful life. It contains no chemicals harmful to the environment and has a long shell life of up to 10 years. A disadvantage is that it is suitable for low drain or only intermittent high drain applications.

Alkaline batteries are mainly used when cost is a factor. The voltage is rated at 1.5 volts. This voltage rating is already a downside for our project use since we will need 3 volts at least. This battery will not be discussed further for this reason.

Silver oxide cell battery is a primary cell with a high energy to weight ratio. Silver oxide battery has a higher open circuit potential and a flatter discharge curve than a standard alkaline battery. Silver oxide batteries become hazardous on the onset of leakage. This onset generally takes five years from the time they are put into use, which is their life shelf time.

Two AA lithium batteries to power GOLD were also researched. A good reason for using these batteries is so the user can easily replace the batteries when needed. Lithium batteries provide the most lasting power and can purchased as disposable or rechargeable. For our project, it will also be more cost efficient to use rechargeable batteries. The power for the device will be delivered from two 1.5-volt batteries. A better solution to power our device would be to use button cell batteries because they are much smaller and compact. The issue with that would be implementing the chassis for it. It could be a little more difficult when working with such a small component.

3.3.2 Microcontrollers

Most below statements in this section come from Texas Instruments documentations, Wikipedia pages on MSP430 devices, and are not my work, besides basic conclusions on applications suitable for our glucose detection system design. Microcontrollers are at the center of portable applications and provide a computing solution to tasks that want to be automated. These SOCs (system on a chip) include a processor core, memory, and programmable input/output peripherals all on a single integrated chip. By reducing the size and cost, microcontrollers make it economical to digitally control even more devices and processes. Therefore, the use of microcontrollers is very in volume with Figures of, “about 55% of all CPUs sold in the world were 8-bit microcontrollers and microprocessors, in 2002.” [19] These Figures are growing and the capabilities of some microcontrollers are comparable to the computers of yesterday with these processing units being capable of 32 and 64 bit processing. Thus, the use of microcontrollers allows many companies to achieve quick, effective, and cheap solutions for many applications ranging from toys, to medical devices.

3.3.2.1 Features

In Texas Instruments document on health applications are a list of suitable microcontrollers for applications including Blood Glucose detection. The microcontroller includes the MSP430 family as suitable options. The MSP430 family of microcontrollers includes features such as flash memory, RAM, LCD capabilities, and SPI, I2C, and UART communication protocols. The MSP430 family is capable of utilizing low power applications with idle modes having a current draw of less than 1uA. The top CPU in the family is 25MHz and it includes mode that can slow that clocking speed for lower power consumption. The device family includes six different low power modes that will turn off unneeded clocks and reduce CPU for even lower power consumption. On top of these features these Microcontrollers have a wake-up time of less than 1ms and this allows the Microcontroller to stay in sleep mode longer. There is a variety of configurations utilizing the different peripherals, which include the internal oscillator, the timer including PWM, watchdog, USART, SPI, I^{2}C, 10/12/14/16/24-bit ADCs, and brownout reset circuitry. Other peripherals include comparators, integrated op amps, 12-bit DACs, LCD drivers, hardware multipliers, USB, and DMA for ADC results.

These devices are programmable via JTAG capabilities or a built in bootstrap loader (BSL) using UART such as RS232, or USB on devices with USB support. The limitations of the MSP430 family is that they do not include an external memory bus and is limited to on chip memory which goes up to 512KB memory and 66KB RAM. This limits the MSP430 in terms of computing capabilities. Considering only the CPU draw a lithium coin cell battery like CR232 could provide a 0.7uA current draw for 32 years disregarding self-discharge. In real time clock mode the CPU can go to sleep with a clock running which will wake it up at a specific future time. In RAM retention mode, some external signal is required to wake it, e.g. I/O pin signal or SPI slave receive interrupt. The MSP430 family utilizes a 16 bit RISC processor, 16 bit registers, and constant generators which help with code efficiency.

There is also a digitally controlled oscillator that allows the microcontroller to wake up from low power mode in times that are less than 6us. These microcontrollers typically are configured with two 16-bit timers, high performance ADCs, multiple operational amplifiers, one universal serial communication interface (USCI), one universal synchronous/asynchronous communication interface (USART), DMA, 80 I/O pins, and a segment liquid crystal display (LCD) driver with regulated charge pump. Along with these features is the ability for these microcontrollers to be able use wireless communication modules such as RF modules or Bluetooth/ZigBee modules for communication protocols for further device integration. Typically, MSP430 microcontroller standalone ICs will cost less than $10 and can be purchased so they come with the features that are needed such as temperature sensors, extra operational amplifiers for signal conditioning, additional ADC with greater sensitivity, and other peripheral devices.

“The MSP430 CPU uses a von Neumann architecture, with a single address space for instructions and data. Memory is byte-addressed, and pairs of bytes are combined little-endian to make 16-bit words. The processor has 16 16-bit registers with 4 registers dedicated to general purpose, R0 is the program counter, R1 is the stack pointer, R2 is the status register, and R3 is a special register called the constant generator, providing access to 6 commonly used constant values without requiring an additional operand.” [20] The instruction set includes 27 different instructions with the instructions each being 16 bits followed by up to two 16-bit extension words and “addressing modes are specified by the 2-bit as field and the 1-bit Ad field.” As far as processing speeds in reference to cycles, “instructions generally take 1 cycle per word fetched or stored, so instruction times range from 1 cycle for a simple register-register instruction to 6 cycles for an instruction with both source and destination indexed.” The MSP430x extensions that include 20-bit registers include added instructions and these can take up to 10 cycles to execute. The MSP430 instruction set also includes a number of pseudo instructions and implements different forms of the regular instructions.

The MSP430FG2259 is a microcontroller unit that is produced by Texas Instruments that incorporates the MSP430’s 16-bit RISC CPU, peripherals, and a clock system that interconnect using a von Neumann architecture using a memory address bus and a memory data bus. The clocking system uses a low-frequency auxiliary clock, which is driven directly from a common 32-kHz watch crystal, and an integrated high-speed digitally controlled oscillator which can source the master clock that is used by the CPU and high-speed peripherals. The MSP430 families contain embedded emulation logic that is accessed via JTAG. The code access is always performed on even addresses, and data can be accessed as bytes or words. The low byte of a word is always an even address and the high byte is at the next odd address. The system reset circuitry sources both a power-on reset (POR) and a power-up clear signal (PUC). The power on reset is generated by any one of the following three conditions, powering up the device, a low signal on the RST/NMI pin when configured in the reset mode, or an SVS low condition. The power-up clear signal occurs from the following event triggers, a power on reset signal, watchdog timer expiration in watchdog mode only, watchdog timer security key violation, or a flash memory security key violation.

All devices in the MSP430 family have a brown out reset, and the brownout reset circuit detects low supply voltages such as when a supply voltage is applied to or removed from the VCC terminal. The brownout reset circuit resets the device by triggering a POR signal when power is applied or removed. [20] There are three types of interrupts, which are system reset, (Non)-maskable NMI, and maskable interrupt. Interrupt priorities determine what interrupt is taken when more than one interrupt is pending simultaneously. Non-maskable interrupts(NMI) are not masked by the general interrupt enable bit, but are enabled by individual interrupt enable bits. Maskable interrupts are caused by peripherals with interrupt capability including the watchdog timer overflow in interval-timer mode. This microcontroller CPU includes the following features: a RISC architecture with 27 instructions and 7 addressing mode, orthogonal architecture with every instruction usable with every addressing mode, full register access including program counter, status registers, and stack pointer, Single-cycle register operations, Large 16-bit register file reduces fetches to memory, 16-bit address bus allows direct access and branching throughout entire memory range, 16-bit data bus allows direct manipulation of word-wide arguments, constant generators provides six most used immediate values, direct memory-to-memory transfers without intermediate register holding, and word and byte addressing and instruction formats. The MSP430FG4618 includes two flash memory arrays, and all flash memory is segmented into segments where a single bit, bytes, or words can be written to flash memory. The flash memory is partitioned into main and information memory segment and there is no operational difference between the two sections.

The default mode of flash memory is read mode and in read mode the flash memory is not being erased or written, the flash timing generator and voltage generator are off, and the memory operates identically to ROM. MSP430 flash memory is in-system programmable (ISP) without the need for additional external voltage. The CPU can program its own flash memory. Reading or writing to flash memory while it is being programmed or erased is prohibited. If CPU execution is required during the write or erase, the code to be executed must be in RAM. The supply voltage supervisor (SVS) is used to monitor the AVCC supply voltage or an external voltage. The SVS can be configured to set a flag or generate a POR reset when the supply voltage or external voltage drops below a user-selected threshold.

The SVS detects if the AVCC voltage drops below a selectable level. The direct memory access (DMA) controller transfers data from one address to another, without CPU intervention, across the entire address range. For example, the DMA controller can move data from the ADC12 conversion memory to RAM. Using the DMA controller can increase the throughput of peripheral modules. It can also reduce system power consumption by allowing the CPU to remain in a low-power mode without having to awaken to move data to or from a peripheral. MSP430F5529 uses a Unified Clock System that supports ultra-low power consumption. The UCS module includes up to five clock sources, with some being low frequency and others up to 32MHz. Thus enabling us to use low power mode when the sensor is not in use.

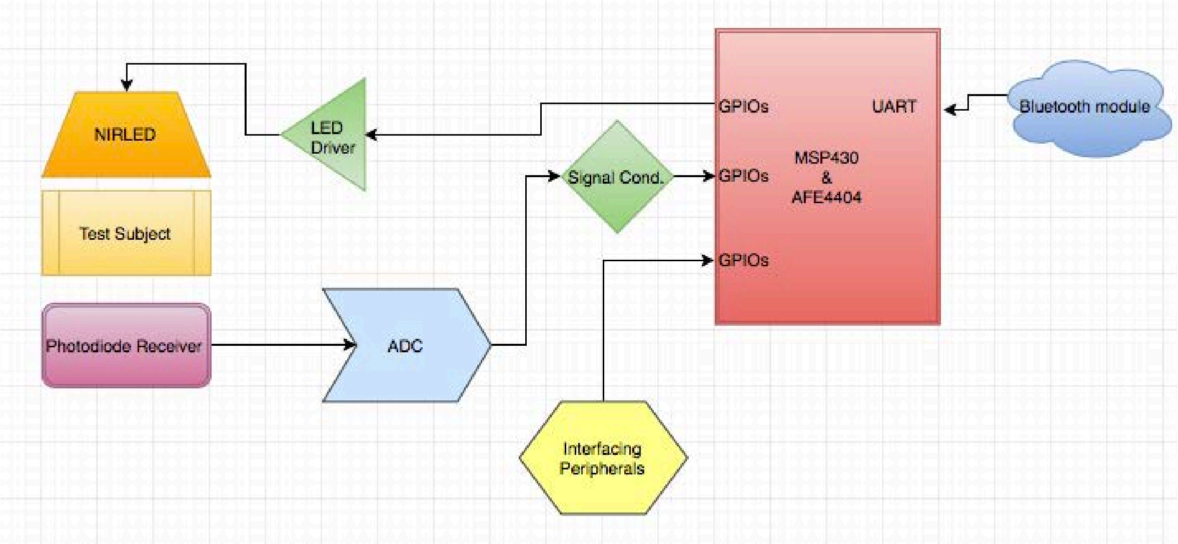


Figure 8: Functional Block Diagram of Project

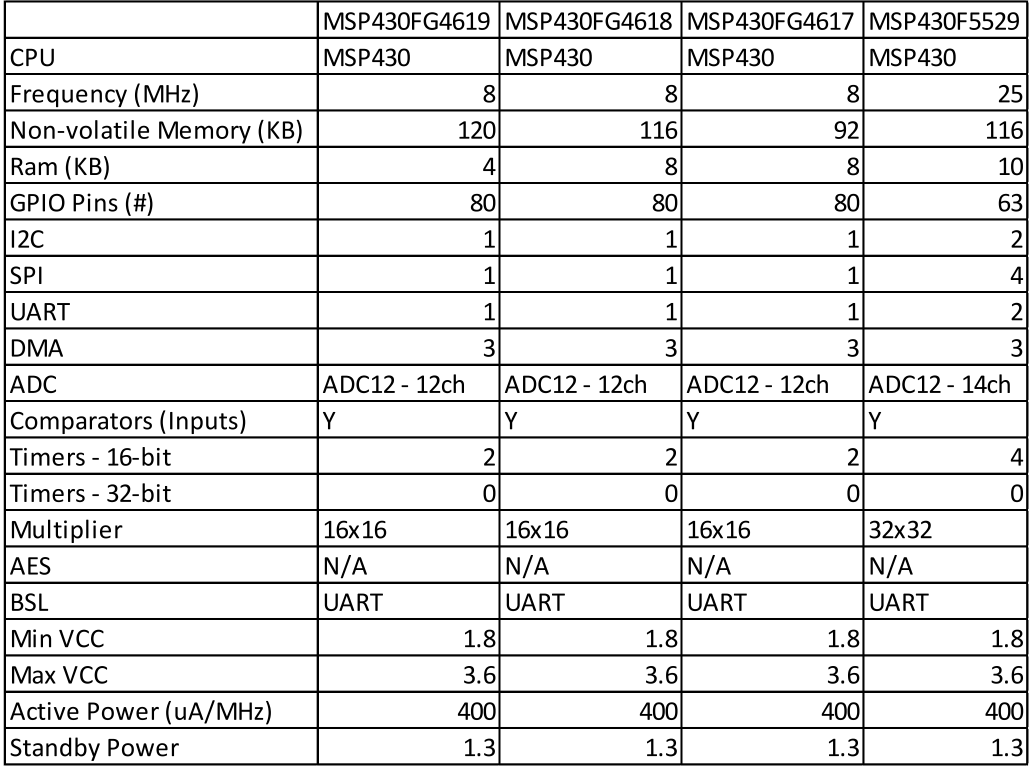
3.3.2.2 MCU Comparisons

The MSP430 family provides suitable microcontrollers for many applications, to decide which subset of MCU that fits the bill we must compare different device models. We will consider the MSP430FG461x, MSP430FG47x, and MSP430F552x families. Ideally, in our application it is important to have enough memory for the Microcontroller to store samples before they are sent wirelessly through Bluetooth therefore having a large amount of large volatile memory is needed. It should be stated that all these microcontrollers could be configured to support Bluetooth. I2C communications is needed for proper communication more than one port might be handy. Ideally, it would be convenient if the device came with a 16 bit or more ADC however our design will call for an ADC and DAC that will function outside of the microcontroller, which will create an 18 bit or greater resolution with many channels.

A microcontroller that uses the least amount of power is important if the device is to be used in continuous monitoring applications. Taking into consideration that has a low standby power is important however not as imperative as active power draw, and therefore does not take precedence. A large amount of GPIO pins is a luxury for integrating more sensors. Our application calls for 20-bit ADC therefore our data will take at least two registers to store information before being sent wirelessly therefore the limit of data capture is on how fast we can send that information through Bluetooth. Using multiple sensors can therefore be a bottleneck to the performance of the data transfer. Since the Microcontroller's main function is to store and send sensor data a large RAM for multiple program tasks is not very important, and 8KB will be plenty of memory to run our applications. It should be stated that these microcontrollers could be integrated with Bluetooth simply by communicating the Bluetooth module through UART.

For breadboard design it would be useful to have a device that utilizes a PDIP package type and the MSP430FG4618 has in its starter pack. Although the F4618 chip does not have a DAC component that is okay since the AFE4404 does most of the analog processing with a ADC and DAC that is needed for the applications such as transimpedance analog to digital and offset cancellation through the DAC. The MSP430FG4618 is a suitable and breadboard friendly solution that provides full functionality but easy implementation. This microchip provides enough memory space, has all the needed communication protocols, meets power requirements, and is relatively cheap while having additional hardware that might be useful such as LCD capabilities, a temperature sensor, 16x16 multiplier, 80 GPIO pins, an Operational Amplifier, and a 12-bit ADC that has 12 channels, all of which can add extra functionality to the desired application.

Table 2: Microcontroller Device Comparison



	MSP430FG4619	MSP430FG4618	MSP430FG4617	MSP430F5529
CPU	MSP430	MSP430	MSP430	MSP430
Frequency (MHz)	8	8	8	25
Non-volatile Memory (KB)	120	116	92	116
Ram (KB)	4	8	8	10
GPIO Pins (#)	80	80	80	63
I2C	1	1	1	2
SPI	1	1	1	4
UART	1	1	1	2
DMA	3	3	3	3
ADC	ADC12 - 12ch	ADC12 - 12ch	ADC12 - 12ch	ADC12 - 14ch
Comparators (Inputs)	Y	Y	Y	Y
Timers - 16-bit	2	2	2	4
Timers - 32-bit	0	0	0	0
Multiplier	16x16	16x16	16x16	32x32
AES	N/A	N/A	N/A	N/A
BSL	UART	UART	UART	UART
Min VCC	1.8	1.8	1.8	1.8
Max VCC	3.6	3.6	3.6	3.6
Active Power (uA/MHz)	400	400	400	400
Standby Power	1.3	1.3	1.3	1.3

3.3.4 Mobile Application Development

Following a glucose diagnosis, a user must check on their blood glucose levels frequently. This causes users to prick their skin to have the glucometer check their glucose level, then the need to log this value somewhere. This is where the current available mobile applications on the market step in.

3.3.4.1 Android

Within the Android application ecosystem, there are multitudes of applications available to help users keep track of their glucose levels. From a glance, there are mainly two types of glucose applications available on the Google Play store:

glucose applications that the user has to input the glucose measurement manually, and glucose applications that communicate with an external glucometer through wireless technology and get the glucose measurement through this method.

An example that uses the first method is the Blood Glucose Tracker application on the Google Play store. This application allows the user to keep all of their glucose measurements in one place, and back up the information to other services such as Google Drive, or Dropbox. When the user uses this app, they can manually enter their glucose measurements during the day for certain events such as before breakfast, before dinner and before they go to sleep. The user is not limited to these events alone, and can add more measurements if they wish to do so. Since the application stores the glucose measurements that the user has made, it can also show the user the change in their glucose levels over time through a line graph. It also reports their average glucose levels for the week, month or all time.

Another application that uses the first method is the Diabetes:M application on the Google Play store. This application is similar to the previously discussed application; however, it goes much further in informing the user of their condition. This application keeps track of more factors such as weight, caloric intake, cholesterol, blood pressure, insulin levels and more. It actively lets the user know if their glucose levels are at a safe level, and their previous glucose measurement. Unlike the previous application, this application can read glucose measurements from various glucometers through the file output of the measurement taken.

With the ability to read information directly from various glucometers, and the extra information that the app asks for when a user makes a log entry, it can perform tasks that are more sophisticated. It can calculate the active insulin in the bloodstream, and show the glucose state distribution over a week, two weeks, a month and ninety days. There are various graphs depicting average glucose per meal, insulin daily totals, glucose overlay, and more. This application also allows the user to select glucose injection sites for their torso and fingers. If the user is curious on how the application derived some values, the calculation details are derived with the values shown as well. Android Wear also has integration with this application, showing this information more readily to the user. Both applications that use the first method can send the information stored to a doctor through email as well.

While there are applications that allow manual input, there are applications available that get the readings directly from an external glucometer and make the measurement and logging seamless. An example of this is the OneTouch Reveal application on the Google Play Store. This application connects to the glucometer through Bluetooth and receives the glucose measurement. The application keeps track of each glucose measurement and it can show two weeks, a month and ninety-day glucose measurement patterns. This application lets the user know if their reading is indicating that they are experiencing hypoglycemia, hyperglycemia or they have normal glucose levels by indicating the corresponding condition with blue, red or green colors. The user also can send their glucose measurements to the doctor or export the data to an Excel csv file. In addition to keeping records of the glucose measurements, the application can track carbohydrates, exercise and insulin levels in its twelve-month logbook.

Another application that connects directly to a glucometer is True Manager Air. This application communicates with a glucometer through Bluetooth as well. Like the aforementioned application, when a user takes their glucose measurement, the results are color-coded based if the glucose measurement means the patient is hypoglycemic, normal or above normal. However, this application allows the user to set their own target ranges, since this may differ from person to person based on what the doctor says. Like the applications that use the first method, this application uses graphs such as pie charts, bar graphs and line graphs with trend lines to convey glucose information to the user. This application can also send the recorded glucose measurements to the doctor as well through email.

3.3.4.2 iOS

Like the Android application ecosystem, the iOS application ecosystem has many applications to help users with their day-to-day lives. Like Android, the iOS App Store has the same types of glucose applications listed on it: the glucose applications that require the user to enter glucose measurement and other information manually, and more specialized glucose measurement applications that pair with a glucometer and automatically read the information from the external device.

One of the examples on the iOS App Store that uses the first method is the Diabetes Tracker application. This application allows users to set target ranges for before eating and after eating, and highlights glucose measurements that exceed this set range. It also has a database back-end where they store food nutritional information so users can easily search for the food that they have eaten and it makes it painless to keep track of their caloric, carbohydrate and sugar intake. Since this application has the database back-end, it can use this information to relay healthier alternatives to the food that the user wants to eat by assigning a food grade to food products. It has reminders to make sure the user takes their measurements so they do not forget. The application also tracks the insulin used, what other medications they are using, A1C, LDL, HDL and BP levels, and if the user wishes to create a custom tracker, they can do so if they wish. The application also allows users to keep track of their weight, and exercise so they can hit weight targets.

Another application that is available on the iOS App Store that uses the second method is Glooko. This application uses the second method, but somewhat differently from the way this method was described before. In this application, the user is not limited to only one glucometer. The application can support over fifty glucometers, insulin pumps, and CGMs. Like the aforementioned application, this application also has a back-end database where food information is stored, and this is used to help the user log their carbohydrate intake. The user is also able to make notes manually or automatically about how carbohydrates, insulin and exercise affect their glucose measurements.

The specialty of this application is the ability to communicate with many different devices and take the information that each of these devices provide and give the user a more detailed overview of their condition. The ability to connect to fitness trackers such as Fitbit, Jawbone UP and more help create this logbook. With all of the data that the application collects, it can show the user daily, or weekly patterns in their glucose and insulin measurements, while if they want to make it more narrow, they can look at a 24-hour log of their glucose, insulin and carbohydrate measurements. The application also reminds the user to measure their glucose levels, take any other medicine, or about any other event that they wish to be reminded about. The data that the application logs can be shared with their doctor, so changes can be made if need be. This data is stored securely on remote servers, and if the user wants the healthcare professional to access their data, they must supply a code to give them access.

3.3.4.3 Platform Differences

When a developer wishes to develop for Android or iOS, there are some things that they must take into consideration with their choice. One of the major things to think about when a developer is considering a platform is the smartphone OS global market share. This lets the developer know the potential amount of users that their application can reach. For instance, the Android OS makes up 86% of the global market, and iOS take up about 13% of the global market [21]. Figure 9 shows us the global market share for the major smartphone OSes:

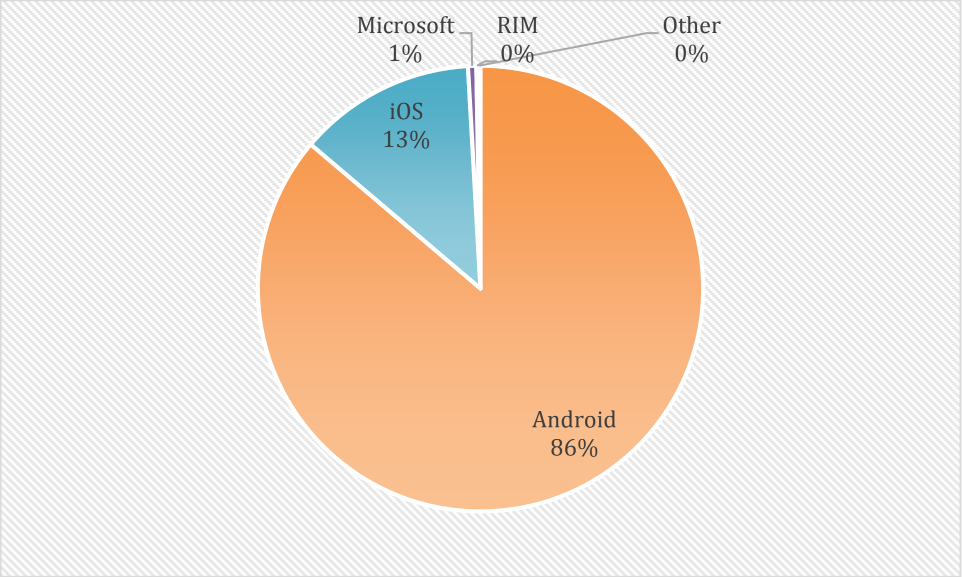


Figure 9: Global Market Share held by Major Smartphone OSes

Even though Android is the leader in the global market, it has problems that developers must take into consideration. One of those problems is Android fragmentation. Android fragmentation occurs because since Android was created with the intention of being open, which allows developers and manufacturers to customize the base Android experience for their devices.

The problem with this is when Google decides to push out a new version of the Android OS, it is left to the manufacturers to take that operating system and modify it so it can be pushed out to their devices on the market. Since this is left to the manufacturers to keep their devices up to date, older devices are pushed towards obsolescence since there is no incentive to use resources to bring the latest version to it. This causes users to be left with versions of the Android OS that lack the newer features and forces developers to consider how far back they wish to go in support of aging versions of the operating system. Even though the cellular market has been moving towards a bi-annual upgrade cycle, not everyone h According to Google, Figure 10 was the distribution of the Android OS versions during a collection of data in September 2016 [22]:

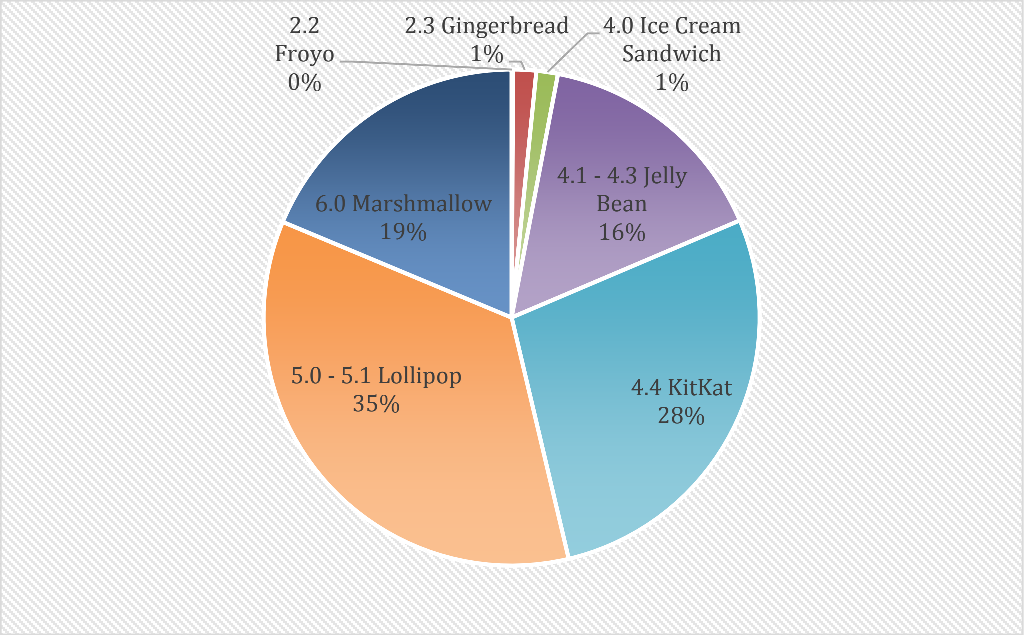


Figure 10: Android OS Version Distribution

For iOS, things are different in respect to fragmentation. Since iOS runs only on the devices that Apple creates, there is not a problem for having manufacturers come in to modify the base operating system to their liking. This closed approach to the OS distribution definitely helps Apple here in terms of fragmentation, but it can also cause dismay to some power users that wish to get more out of their iPhone.

Each year, Apple comes out with a new iPhone, and this new iPhone usually comes with a fresh version of iOS with it. Since Apple retains control over the operating system that their phones use, they can easily customize it for the various iPhones still out on the market, and push the latest OS to older devices. This is helpful for developers since the fragmentation of the iOS user base is much less significant, they do not have to worry about supporting very old versions of the operating system. Another aspect that helps Apple's overall iOS distribution chart is the brand loyalty of Apple consumers. Apple's overall brand strength and marketing help generate interest in getting their new iPhone. They tend to get the new yearly upgrade to the latest and greatest iPhone device and this overall mindset of yearly upgrading helps close holes of potential fragmentation as well. According to Apple, Figure 11 was the distribution of iOS during a collection of data in October 2016 [23]:

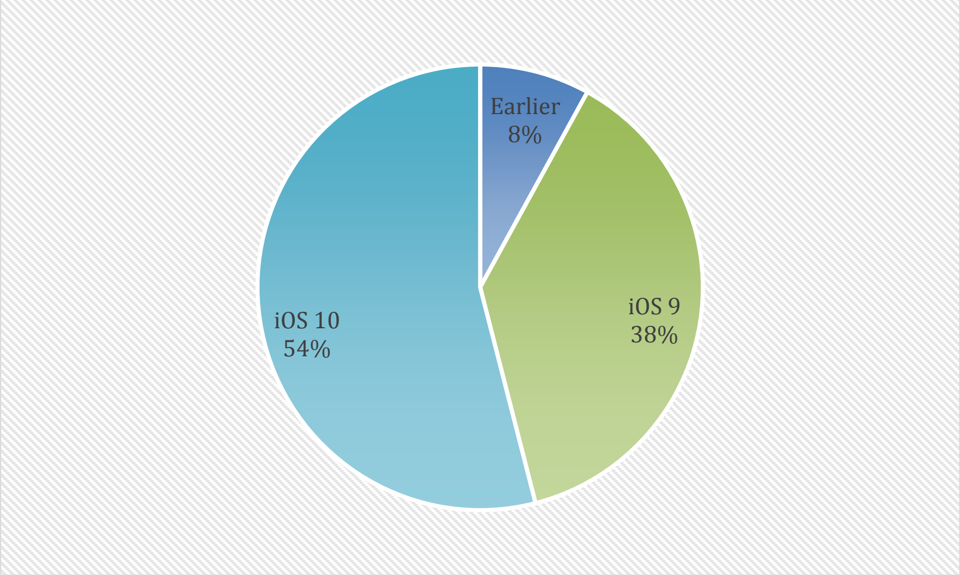


Figure 11: iOS Version Distribution

3.3.4.4 Development Tools

Since Android and iOS are two different platforms with different approaches on how they develop and distribute their operating system, the tools to develop for each operating system will differ as well.

For Android, developers use the Android Studio IDE for their application development. The main language used for application development is Java. Android Studio also includes various levels of the Android SDK, which includes the application programming interface to help developers perform tasks within the application. This is where the developers can choose what version of the OS to support in their application. Android Studio also includes an Android OS emulator, so if the developer does not have an Android device, they can still develop for the platform, but they would not be able to test features that require interaction with the physical hardware itself, such as Bluetooth communications, NFC and more.

In iOS’s case, developers use the Xcode IDE for their application development. The languages supported in Xcode are Swift, and Objective-C [24]. Apple supplies Xcode with the latest iOS SDK when it is downloaded. However, Apple pushes their developers to support the latest iOS version, and they enforce this by refusing to allow new applications to use obsolete OS versions. Like Android Studio, there is an iOS emulator, so developers are not required to own an iPhone or iPad to develop applications for these devices. However, the same problem occurs here where they cannot access actual hardware devices such as Bluetooth on it.

In the GOLD project, the Android device is not the only part of the project that requires programming. The MSP 430 microcontroller also requires programming, and this comes with its share of choices for development. The two options that the team had for development were the Texas Instruments Code Composer Studio and Energia.

The Texas Instruments Code Composer Studio is an integrated development environment that includes a C/C++ compiler, assembly compilation support, a project environment, a code debugger and more. Code Composer Studio is robust in what it offers us for development, and with all of its features comes a negative. In the case of wanting to do fast prototyping and lots of testing, using Code Composer Studio can be quite overwhelming. Working with this integrated development environment in C gives us the ability to manipulate the watchdog, various bits that trigger interrupts for low power mode and more, but with this, it makes the code more complex. Using the UART interface is also quite complex, requiring lots of bit manipulation, flag triggering and more. For our purposes, this would be a bit too low level for speedy prototyping and testing, especially since the team wanted to make sure that the Bluetooth module was functional and that the Android device was able to make a connection via pairing and establish a connection to transfer data.

The other option for an integrated development environment available to us was Energia. Similar to the Texas Instruments Code Composer Studio, this integrated development environment’s sketches are C/C++ based and these sketches are compiled with the open source MSPGCC compiler [25]. The difference between the two is with the approach of how things are coded. In Energia, the developer does not have to be concerned with the low end bit manipulation and setting of flags, nor have to worry about disabling the watchdog if there is a need for infinite loops in code. All of that is obfuscated from the developer and is taken care of. The approach to coding in the Energia integrated development environment is straightforward; there are two main method blocks that will be run within the code. There is the setup() method block, where the I/O ports on the MSP 430 microcontroller are configured for input or output and this is where the serial port is opened for communication. The other block is the loop() block, and this is where the code that needs to be looped indefinitely will run. This is the block where the developer would place checks for incoming signals, and then have the MSP 430 microcontroller begin methods to handle these incoming signals. With this style of coding, the difficulty of coding up a working prototype has gone down significantly, because of the aforementioned obfuscation, but as well as the simplicity of the C-based script language. With this straightforward way to program, based off the Wiring framework, which the Arduino boards use, makes prototyping and testing more simple and efficient.

However, this comes with some drawbacks. In the Texas Instruments Code Composer Studio integrated development environment, there is the ability to code in assembly, which is not present here. With the abstraction of low-level things such as registers, the ability to work on the register level is lost. Another big loss is the ability to use a debugger to find out problems in the code. The Texas Instruments Code Composer Studio integrated development environment has a debugger where you can see what values each variable is holding at a certain point in runtime if breakpoints are used. With this, you can step through the code to see what changes are occurring, not only at the high-level variables, but at the low-level registers as well. Not only that, but also the ability to check what changes are occurring in memory is a big plus as well. Without the debugger, it makes taking care of or finding bugs harder than usual. The loss of a project-based environment can be a negative if the sketch grows past a certain size in code length, but for quick testing this can be left as a neutral point.

3.3.4.4.1 Languages

With the range of development tools that the developer can use, there are quite a bit of different languages that can be used to program mobile applications. In regards to the languages between the platforms, Java, Swift and Objective-C are all object-oriented programming languages so the concepts behind the languages are similar, with only syntax differences. Java is maintained by Oracle and is developed with a platform agnostic approach, Java code can be run on mobile devices, embedded systems and computers in a similar fashion. Java classes can be defined and implemented in one file because the language supports importing other classes into the current working class for implementation.

Objective-C is an extension of the C programming language that adds object-oriented functionality. Unlike Java’s automatic garbage collector, and memory allocation handling, with Objective-C, the programmer must handle allocating and deallocating memory on their own, similar to how it would be done in C or C++. In Objective-C, class implementation is also different from Java. Classes are implemented with two files in Objective-C: header files (.h) and source files (.m). The developer can also use C++ code in their Objective-C program by including it in .mm source files. Objective-C also does not work with methods the same way that Java does. In Objective-C, methods are called with messages. Messages are method signatures with set parameters, and they are called programmatically. Even though the methods in which these methods are invoked are different from each other, the accessory methods for fields within the class can be accessed in a similar way [26]. However, Apple decided to replace this language with Swift.

Swift is based off C and Objective-C with safe programming patterns, and has an easier learning curve compared to Objective-C that was used in the past [27].

With the easier learning curve in mind, Swift takes on a more script-like language to programming, which is very different compared to the more structured and explicit programming done with Java and Objective-C. Swift does not require a main function to run code, nor does it require semicolons at the end of a statement either. Swift also does not require the developer to explicitly state the type of the variable that it will be using, similar to PHP. It will determine the type of variable through type inference. A major difference between Java, Objective-C and Swift is that Swift supports the returning of multiple values from a method. While Swift is more of a scripting language, it did borrow concepts from object oriented programming as well. There is the ability to create classes like how they would be created in Java or Objective-C. Like Java, Swift uses reference types when working with classes, since the references for a certain class will all point to the same point in memory. A difference from Java is that Swift does support structs, which is a concept used in C, C++ and Objective-C. While Swift does support structs, structs are value types in usage. More things that Swift can use natively are switches with tuples, and optional values. In Java and Objective-C, you can only evaluate one condition at a time in a switch case statement, while in Swift, you can evaluate more than one at a time. For optional statements, these are used when the program is not sure whether the value that it is working with is correct. Swift has extensive support for this, while optional statements has very limited support in Java, and no support at all in Objective-C [28].

3.3.4.4.2 Methods of Wireless Communication

Many applications out in the application ecosystem take different approaches to wireless communication. Some examples of wireless communication include communication between phones, communication between the phone and a remote server, communication between the phone and an application on the computer through Wi-Fi, and communication between the phone and an embedded system using Bluetooth.

One of the most common forms of wireless communication between devices on the internet and devices in a local area is through Wi-Fi. Wi-Fi is the implementation of the IEEE 802.11 standard for wireless local area network communication. The device that wishes to communicate can connect to a Wi-Fi network that it is in range of, and transmit information through this network to other devices connected in the same network. The devices may not be in the same room, but since Wi-Fi uses radio frequency waves, these waves can go through walls, transmitting the information over a larger range. Since Wi-Fi can be used to access the internet, provided the access point is connected to the internet, the devices connected to this Wi-Fi access point can use it to connect to remote servers to obtain and send information to and from the end user. Wi-Fi signals can operate at two different frequencies for transferring data: 2.4GHz or 5GHz. Within these two operating frequencies, there are different standards of operation such as

802.11a, 802.11b, 802.11g and 802.11n. These standards of operation dictate the frequency that is used, the coding of signals if coding is used, and the maximum transfer rate [29].

Another means of wireless communication is Bluetooth. Bluetooth differs from Wi-Fi because it adopts a master and slave system. When a piconet is established, a master device is designated, and other devices are designated as slaves. As of now, the range of Bluetooth is limited, and the data throughput is limited in the base specification. It is good for transferring small files between devices, or communication between a phone and an embedded system, however, Wi-Fi would be best for transferring bigger files. Like Wi-Fi, Bluetooth has different versions of the Bluetooth specification. There is Bluetooth basic rate/enhanced data rate which is used for playing music through headphones or speakers, Bluetooth with low energy functionality, which allows the usage of Bluetooth in embedded systems that are trying to be low in energy consumption, and Bluetooth High Speed [30]. Bluetooth High Speed has a higher data transfer rate than the Wi-Fi 802.11b specification, but it is still much slower than the other Wi-Fi specifications. Bluetooth can shine when end users wish to connect two devices without needing to have an internet connection, such as when someone wants to use an external device to take a measurement and have the smartphone process the data that it receives. Bluetooth also plays nice with UART interfaces, since Bluetooth can support many different baud rates, especially 9600, which is the baud rate used for communication using the UART interface.

Another means of communication between devices is Near Field Communication (NFC). NFC is quite different from Bluetooth or Wi-Fi in the fact that it does not use radio waves for communication. Devices that use NFC emit a small electric current that creates an electromagnetic field. This electromagnetic field can induce a current in the reader of another device, and information can be transferred through this induction. The limitation is that NFC can only communicate over very small distances, which tend to be less than 20cm [31].

3.3.5 Sensors

In non-invasive glucose measurements, multiple techniques are being researched and implemented in modern times. Google has created a patent that a user can wear an eye-mountable device and glucose can be sensed with something as thing as a contact lenses. In terms of the current market, it seems as if that type of product development is not to close by. Another approach that has been used to sense glucose levels non-invasively is with a breathalyzer. In a similar fashion that ethanol is detected through the breath this technique using acetone is being implemented to check glucose levels. Breathalyzers in the market can run up a cost of over fifty dollars per unit. In order to keep cost down and measurements reliable we propose using a photodiode to sense light absorption through transmission using either a laser or a LED. The choice between these devices is referenced in the sections that follow.

A Photodiode measures in a certain wavelength range that works well with spectroscopy. This device can be used to measure the photocurrent through a component using voltage divider shown in Figure 12 below. This is a simplified photodiode circuit but other methods will be looked at also.

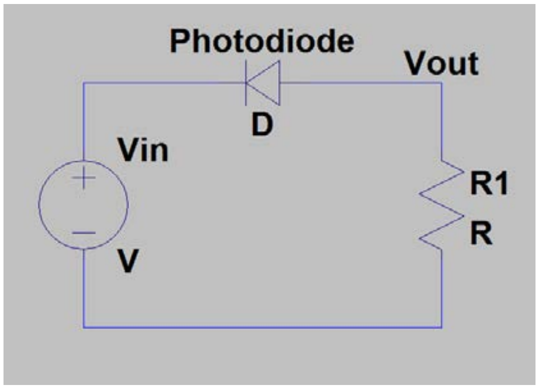


Figure 12: Basic Photodiode Circuit

The relation between the current of the diode is $I_{ph} = \eta eA\Phi$. Ƞ is the quantum efficiency, e is the electronic charge and Φ is the photon flux density while A is the junction area, assuming the reverse bias voltage across the diode is constant. The sensitivity of output depends on the value chosen for R1. If there is no light upon the photodiode then the output voltage will fall to zero. The diode having reverse polarity with the voltage source also the current to be rectified in the voltage output. When a light source is picked up then the resistor will have a small amount of current passing through it which will give a reading. The range of the resistor determines the amount and the intensity of light. This can be moved to sense a wider change or a narrower band. It is also possible to implement a filter stage to get rid of noise shown below.

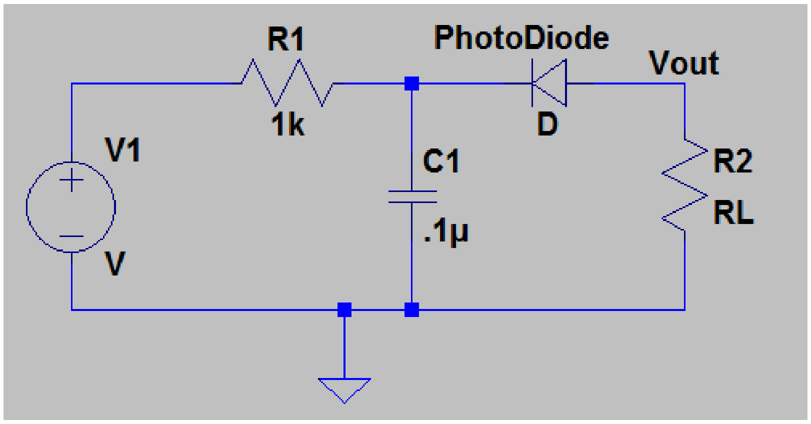


Figure 13: Photodiode Circuit using Low Pass

Introducing the capacitor c1 in the Figure above acts as a low pass filter this could become a problem with the loading current being drawn so a solution to this could be using an active filter difference amplifier.

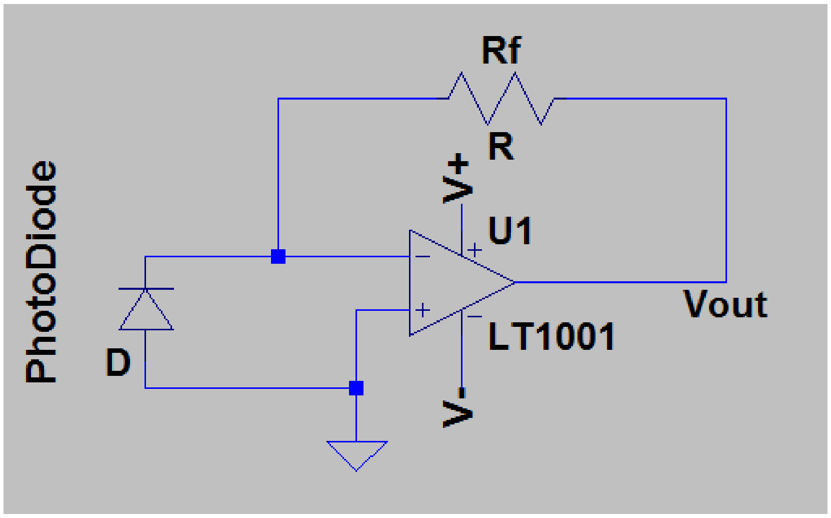


Figure 14: Photodetector Circuit with a loading Op Amp

Applying an active filter isn't necessarily better than Figure 13 because the power supplies of this amp and the schematic becomes more costly relative to power. Also, the slew rate and precision of this amplifier can affect the sample rate of the data if it is slower than the ADC.

The specific photodiode load resistance depends on the wavelength of the incident light the reverse bias and the temperature conditions. The photocurrent divided by the incident light power gives the responsivity. This will be talked about more in depth in the design section. The photodiodes that we will be analyzing will be the High Speed InGaAs PIN ($31.43) and the FGA01 - INGaAs photodiode, 300 ps Rise time ($56.70). The High Speed photodiode has a bandwidth of 800 nm to 1750 nm and a rise time of .3 ns [32]. The FGA01 has almost the same exact specifications except the wavelength goes to 1700 nm instead of 1750, which is not really significant. Although the High Speed Pin is relatively cheaper so it is an obvious choice.

3.3.5.1 Measuring Technology

In the following section, methods of spectroscopy will be analyzed and different light detection methods will be looked at and discussed, as well as parameters of non-invasive testing. Such as transmission through the ear lobes and keeping the measurement integrity consistent. Issues like having thicker earlobes will result in more blood in the tissue so this will affect the absorption. Ways to compensate the differences between people's ear will be looked as well as other ways to normalize measurements with temperature or heart rate.

3.3.5.1.1 LEDs

Light-emitting diode (LED) is a p-n junction that uses electroluminescence with the energy of the photon to display a certain wavelength. The band gap or the space in between the p and n junction is used to manipulate the corresponding energy to match at a certain wavelength. The way a diode works is by dope a semiconductor material to have an excess of holes and a separate one having excess of electrons and putting them together to make a p-n junction. With this diode a voltage source can be placed on either the forward bias or the reverse bias mode. These modes correspond with the flow of the metallurgical junction or space charge region(SCR) of the p and n. When the current is flowing towards the p then the electrons are moving away from p which leads for the holes to combine and push the SCR closer and allow for current flow causing it to be forward bias. On the other hand, if the current is flowing away from the p side then the SCR is expanded because the holes are not being pushed as closely to the SCR and the current cannot pass. There is a small amount current called reverse saturation current, but often times this is negligible. With the diode being able to limit current this corresponds nicely with a switch leaving the diode to emit a certain wavelength to shine a specific color.

One of the first common applications of LEDs was passing infrared light on remote control circuits. LEDs now are commonly used as house lights because they are superior in lifespan and low power comparatively to halogen light bulbs [33].

The application of LEDs for our purpose is to use them for spectroscopic readings. Spectroscopy is explained in a later section. Spectroscopy can use tunable lasers which can come across to being very expensive, but using an LED that naturally has a broadband of signals keeps cost down. We are planning on implementing Near infrared Spectroscopy with LEDs. These techniques are used currently and well reported to test the contents and maturity fruits. In order to use LEDs as a spectrometer you need to coordinate the LEDs in the specific absorption bands. In example, the fruits that were mentioned use wavelengths at 630 nm and 690 nm which is close to chlorophyll absorption peak [32]. Some experiments require optical filters or different light gratings to span the spectral bandwidth narrowly. This is emulating laser spectroscopy which by trait has a narrow wavelength so the gratings do this artificially with LEDs that are broadband.

Dye Laser are sometimes used for spectroscopy. Dye lasers have organic dye that is mixed into a dye cell. The Dye laser has the tradeoff of a much higher resolution because it is tunable and has a very minimally sized bandwidth but the expenses outweigh the benefits so the LEDs seem to be the best fit.

The specific LED that we are considering is the High-Power IR LED with Ball Lens, 1550 nm. These LED is described as having a high power application. The max reverse voltage is 3 volts which works out well according to our power supply. The typical operating current is 50mA and the forward voltage at that current is 1.1 V, and the output power at that voltage is 4mW. As the Wavelength of diodes get bigger than the optical power increases. This also makes sense because it is more capable of penetration at a big frequency. The bandwidth for this specific LED is 120 nm which is useful to perform the spectroscopy at different ranges. The rise time and fall time of this LED is 10ns which is fast enough for the sample rate of the ADC. More importantly this specific LED is less than twenty dollars which is important to keep the total project cost down.

An additional diode that we will compare is the ML925B45F - 1550 nm, 5mW, (0)5.6mm, D pin code, Laser Diode [32]. This is a different measuring technique all together because a diode is in a much smaller bandwidth. This will be discussed more thoroughly in the following section his specific bandwidth is half the size of the LED at 60 nm. The operational current is also much smaller at 10mA and the operating voltage is the same. The power at the operation current is 5mW which is only marginally bigger than the subsequent diode. The biggest tradeoff for lasers is the bandwidth and the price. This laser is relatively inexpensive but the bandwidth is big for a laser. On the other hand the LED in the above paragraph is measured with spectroscopy instead of just absorption so the bandwidth is expected. The takeaway of these comparisons without going into the different methodologies that will be covered shortly is that the LED is cheaper and the diode has a smaller range. The power is almost the same so the real decision factor is what methodology produces a more efficient and accurate outcome.

3.3.5.1.2 Temperature

Temperature can provide an error in measuring glucose even if it is done non-invasively. Just as there is a consideration for thickness, temperature also plays a vital role. Oxygen concentrations, altitude and temperature can alter glucose measurements significantly. For example, one mountain climber that measured 8° C showed a 7% error. These mountain climbers also found a 6-15% error when they were at 3KM above sea level. Noninvasive sensor provide an average error of less than 5% and for our purpose a 20% error is still acceptable and even strived for [34].

Naturally when looking into electronics temperature is a factor. The diode formula in fact changes exponentially when temperature is different due to Thermal voltage. When trying to standardized testing for glucose the temperature of a room should be determined in order to factor in the change. In colder rooms the photocurrent of a diode will be lowered so this change should be modeled.

3.3.5.1.3 Thickness

When transmitting light through tissue it is important to keep in mind that thickness varies between people and this causes a difference in measurements. When absorbing light is measured or calculated then the thickness is not as important because the return is what is important and not the passage. Although, when dealing with transmission of light through a tissue the thickness is crucial because if the tissue is too big then it may never penetrate. If the light never penetrates then it will be impossible to get a reading from the photodiode. Additionally, when measuring glucose non-invasively if the tissue is big then there will be more blood in that corresponding area. This will lead to a higher glucose level reading because of the extra blood. This discrepancy needs to be accounted for so the blood and thickness can be correlated properly to the appropriate amount of glucose as a ratio that way the device can be used properly between people. The way this measurement is accounted for is by measuring the attenuation on a green led and modeling the thickness appropriately.

3.3.5.1.4 AFE 4404

Texas Instruments have recommended using and even supplied us with the AFE4404 Ultra-Small, integrated AFE for Wearable, Optical, Heart-Rate Monitoring and Bio-Sensing. AFE is an acronym for analog front-end that supports three LEDs and one photodiode. The AFE takes the current from the photodiode readings and converts it into a voltage with a transimpedance amplifier and sends that information to an analog-to-digital converter (ADC_. this ADC is powered with a I^2C interface. The block diagram is shown below.

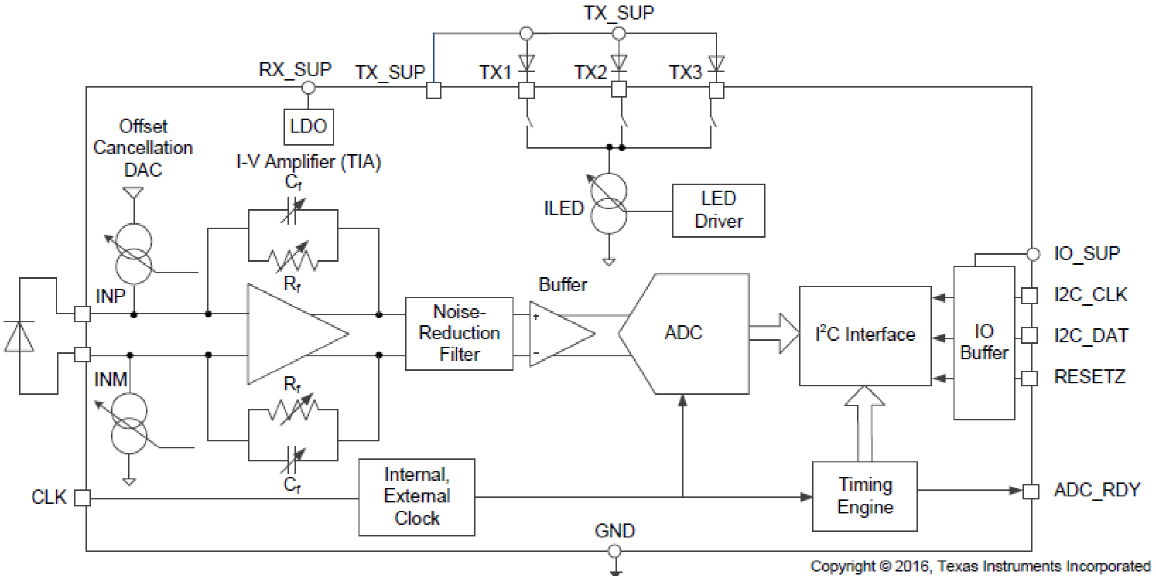


Figure 15: AFE4404 Block Diagram (courtesy of TI)

The range of the LED currents is 0 to 50mA. Due to the 6 bit current control that is 2^6 current controls or 64 steps from 0 to 50mA, or approximately 780uA. The AFE also has a 22-bit ADC from -1.2 Volts to +1.2 Volts, that is 2^22 steps in between approximately 57 microvolts. This is useful when measuring the small range of light absorption and change in the spectral overtones which can be very minimal.

The AFE current application are portable ECG or Electrocardiogram and EEG or Electroencephalogram. In other words measures a person’s heartbeat. The AFE is also used to implement a pulse oximeter. This application takes advantage of the high resolution ADC and the LED to measure the oxygen content non-invasively. The Pulse oximeter shows how much oxygen is in blood and the percentage of hemoglobin which closely related to glucose and provides valuable information when trying to measure glucose. The way the pulse oximeter works is by having a red LED at 660 nm and an NIR LED at 940 nm. Oxygenated hemoglobin absorbs infrared light more and transmits red light better. The duty cycle of these LEDs alternate at about 20 or more samples a second, while occasional having both on at the same time to compensate for ambient light baseline. The minimal amount of light is subtracted and after the interference for the other tissue is accommodated for and the ratio between hemoglobin with and without oxygen is used with a lookup Table on Beer-Lambert's Law. This ration is shown below in Figure 16.

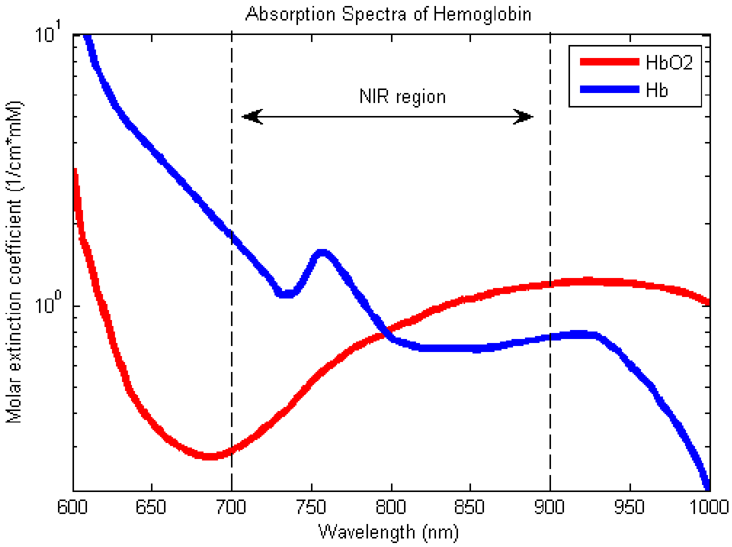


Figure 16: Absorption Model of Glucose (courtesy of Wikimedia)

Accompanied with the AFE, TI also has given us the evaluation module (EVM). This device comes with a graphic user interface as well as ten pin cables to extend the range of the AFE. This graphical user interface (GUI) can sample a 1000 times per second as well as be programmed and power via USB. Using the GUI the 6-bit LED current setting can be altered as well as having separate gain and being able to change feedback resistance and capacitance digitally. The timing of the LEDs can also be manipulated and configuration files can be saved and recreated very efficiently. The AFE4404 EVM automatically regulates the RX and TX as well as the IO digital supply. Additionally, the clock speed of this module is 4-Mhz.

3.3.6 Methodology

It was mentioned in earlier sections that there are different methods analyzed in how to measure the glucose levels in the blood. The first questions is should absorption be used at a single wavelength or should it be a range through spectroscopy. Additionally, should the spectroscopy be in the near or mid infrared range. Many variables come into play when deciding this such as cost, convenience, accuracy and reproducibility.

3.3.6.1 Near Infrared and Ramen Spectroscopy

When building an optical sensor picking the most appropriate wavelength is challenging. Two spectroscopic techniques will be looked at, analyzed and compared in the following text are Raman Spectroscopy and Infrared

Spectroscopy. Raman Spectroscopy deals with scattering of light, while Infrared Spectroscopy (IRS) or more specifically, mid-infrared Spectroscopy (MIRS), and near-infrared Spectroscopy (NIRS) is measured through absorption and transmission. The purpose of spectroscopy is to identify chemicals with how matter interacts with electromagnetic radiation. The Infrared range is from 700 nm to 1 mm, and the Near-Infrared range goes from 700 nm to 2500 nm. Near infrared is useful for oximetry and other biomedical uses because at this wavelength water is almost transparent. In addition, NIR light can penetrate deeper compared to MIR light which is useful for biomedical applications. The downside to NIRS is that the overtones of the molecule have to be analyzed because the fundamental frequencies occur in the mid infrared range [35]. An overtone band is when a spectral band transitions from the ground state to the second excited state.

NIR waves are often transmitted through light bulbs or LEDs, this is good for keeping power limited in contrast to Raman spectroscopy which requires a high-power laser. The way NIRS works is by radiating a light at multiple wavelengths. Spectroscopy is: over a broadband of frequencies. After the light interacts certain wavelengths are absorbed, the location and magnitude are plotted with absorption coefficient. This leaves a fingerprint of NIRS.

On the other hand, Ramen Spectroscopy measures the vibrational information of chemical bonds providing what that fingerprint. The way Ramen Spectroscopy works is a light is emitted onto a molecule and the small amount of light that is shifted is called Ramen Scatter, which is what is observed. The fingerprint is the ramen scatter on different wavelengths. You can compare fingerprints of unknown molecules with known molecules and match them with their respective fingerprints. The fingerprint of glucose in water is shown below in Figure 17.

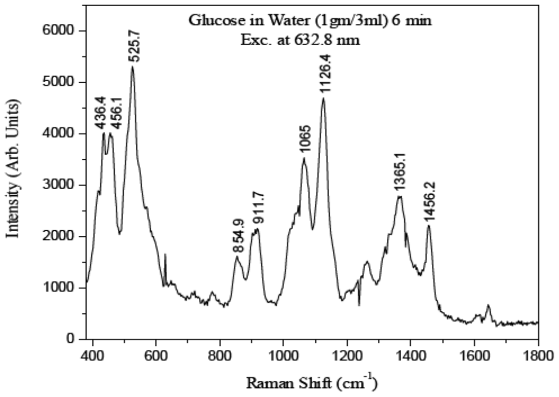


Figure 17: Raman Shift vs. Intensity

Glucose fundamental vibration however occur in the MIR range at 3000nm. In the NIR range the absorption overtones must be analyzed as opposed to the fundamental frequencies and these correspond to the ascending overtones. MIR absorbs water a lot and it does not penetrate very deep so NIR is a more practical wavelength spectrum to do this measurement.

Using Beer Lambert's law defined below as $\alpha(\lambda) = -\log_{10} (T_g(\lambda)/_{Tref}) = \alpha(\lambda)cl$. Where $\alpha(\lambda)$ is the absorption magnitude and T_g is the transmission spectrum of glucose while T_{ref} is a reference sample. C is the concentration and I is the path-length of the radiation through the sample. The wavelength selection is tricky because water is the main component of blood and tissue. If water is absorbs too much than the detected light would be week having the signal -to-noise ration be too low. That is why NIRS should be used because the fundamental frequencies of glucose is not very reliable with tissue and blood as a medium. Water has a low absorption at 1300 nm so the higher-order overtone for glucose is novel in this range if the absorption was strong, but it is not. In this case the first overtone is at 1600 nm to 1850 nm and this is a low water absorption range. That is why 1550 nm is chosen [35].

3.3.6.2 Laser Diode

A tunable laser diode can be used in order to range the spectrum similarly to how the LED does spectroscopy. The difference is that the tunable laser is in a sense more programmable because you are able to tune to a certain spectrum and absorption band and know exactly what absorption you are analyzing. This is not the case for an LED which sends a broadband light and the change is measured but the certain absorption bands are not as precise. Tunable laser diodes are very expensive so it is not a novel way to solve the problem compared to certain LEDs that can range between five dollars to fifty being expensive. Laser diodes can be used that are not tunable. The method is on a single wavelength as one can expect. The way the absorption works is that a laser is shone onto a surface and the photodiode measures the change. In theory if you are measuring glucose in tissue the light that reflects back to you should tell you how much light was absorbed and how much is retained. Lasers have a drawback compared to LEDs because if there is noise over the reading it is harder to normalize a single value compared to doing it through a spectrum and doing a linear filter on the readings.

4.0 Related Standards and Design Constraints

Standards are documents that explain procedures and specifications used to design different materials and products that people and businesses use every day. These standards provide guidelines for manufacturers and developers to utilize when creating and implementing a product. Numerous protocols must be followed in product design and application to avoid any scrutiny or detrimental liabilities that could arise if standards are not followed. For the glucose sensor application, several standards can be referenced to guide our decisions including the invasive glucose detection along with other documents. For some of the standards related to our device, IEEE and other organizations request money in exchange for the standards document; however, the related standards detailed in this section may either be free or be low cost.

4.1 Related Standards

Professionals and students can join a technical engineering organization called the Institute of Electrical and Electronics Engineers, which provides information on new technologies and standards for applications dealing with power, information technology, electronics and all other industries related to electricity. With over 1,000 published standards, the highly respected Institute of Electrical and Electronics Engineers acts as a leading force in the world to produce international standards.

For this design project, a plethora of related standards from IEEE will be acknowledged to ensure that hazards are prevented and that regulations are enforced. Since this device mainly deals with electronics, the protocols must be kept in mind for the electronic components. In addition to being an electronic device, the glucose sensor is also considered a medical device as the device could measure the glucose levels in blood. As the glucose device is understood as an electronics device and a medical device, a few IEEE standards will be discussed below.

4.1.1 ISO/IEEE 11073

Health informatics, the title of the standard ISO/IEEE 11073, details the communications standards between medical and health care devices and computers. In collaboration with the International Standards Organization, IEEE produced this standard that can be applied to devices used for diagnostic purposes by health care systems and organizations as well as personal medical products such as blood pressure monitors, blood glucose monitors, and pulse oximeters. In the stated purpose of ISO/IEEE 11073, the standard explains that interoperability is needed to allow people to “manage their own health independently”; therefore, the glucose detection device falls into the category of medical devices requiring interoperability. As a main concern in this standard, the personal health device such as our glucose sensor will communicate and interface with a phone application by transmitting data through Bluetooth for easy access to this information. However, some regulations and concerns are explained throughout this informative document.

According to this standards document, the glucose device falls under the disease management category where exchanging data information may be a frequent process depending on what the measurements are taken for. The method of exchanging data between the glucose device and a computer system is through Bluetooth, which is considered a wireless personal area network that communicates data over short lengths through radio frequency transmission. Through this Bluetooth method, some type of authentication and encryption is recommended for security purposes and due to the sensitive information that is sent to the mobile application, the glucose levels need to be protected when the transmission of data occurs.

The last few sections of the Health Informatics standard focus on the development of the medical device. The main factors to consider in the development process are listed below in sequential order:

1. Customer needs analysis
    - In addition to satisfying the customer, our group will keep in mind the user who may or may not be the customer. Both the user and the customer must be considered to design and build a product that accommodates to them. The glucose sensor device, considered as a personal health device (PHD), could satisfy a customer's needs by replacing current blood glucose monitors that require the user to prick his or her finger with a needle and test the blood for glucose measurements. For the development process, actors can be utilized for testing and can also contribute to the analysis of the user's needs.

2. Risk management
    - This process consists of risk analysis, evaluation, and risk control. Identifying the risks and hazards of this design's software plays an important in this second step of the development process. After identification, the risks are then assessed to determine if risk reduction is required and the following steps include:
        a) Risk control option analysis
        b) Risk control implantation
        c) Residual risk evaluation
        d) Risk/benefit analysis
        e) Risk arising from risk-control measures
        f) Completeness of risk control

      These steps assisted the group in discovering a few risks such as the miscommunication of data from the glucose sensor to the phone software application and the misuse of the LED and photodiode sensor mechanism to properly calculate the blood glucose concentration.

3. Security
    - Security plays one of the most important roles in the design process if the glucose measurements obtained from the device must be kept hidden from unauthorized entities. The three main fundamentals of security, better understood as information security, are confidentiality, integrity, and availability; in fact, the nickname “CIA triad” represents the three principles of security and provides the manufacturer a platform to ensure information security for their devices. Although this section talks about the CIA triad acting as a framework for developers, the standard does not specify a direct set of rules to abide by; instead, the main advice from the security section encourages the manufacturer to ensure a secure channel for data transmission. With the help of the IEEE 802 standard, the Bluetooth functionality shall be implemented to allow for communication between the glucose transducer and the phone’s software application.

4. Quality of service (QoS)
    - Attributes pertaining to the quality of service must be decided and confirmed during the development of a product for a specific customer demand. Many of those attributes are as follows:

      a) Reliability
      b) Priority
      c) Latency
      d) Bandwidth
      e) Forward and reverse channel setup
      f) Monetary cost
      g) Energy cost

5. Regulatory Issues
    - Lastly, the medical devices must be regulated if the device qualifies for a certain standard or set of standards. For the most part, medical devices such as the glucose monitor must comply with specific standards because according to this IEEE standard, medical professionals would only use a regulated medical device. This glucose project would not be required to comply with regulations set by the Food and Drug Administration (FDA) in the United States but to increase the product credibility, obtaining approval from the FDA greatly increases the success of product marketability.

4.1.2 IEEE 830

Known as the Recommended Practice for Software Requirements Specifications (SRS), the standard IEEE 830 lays out a guide for businesses and developers to set realistic goals and outlines when software needs to be developed. The SRS is a report that details the requirements and specifications of a software product with particular functions. The approaches detailed in the standard assist in the brainstorming process as well as give you the tools to select different software products for your intended purpose. In our case, our team developed a phone software application that displayed the glucose concentration measurement communicated through Bluetooth and to develop this software, several characteristics of a good SRS described were considered in the IEEE 830 standard.

1. Nature
    - The software designed for the glucose device is supposed to display accurate glucose concentration readings and measurements from the Bluetooth module connected to PCB. With the chosen electrical components, the speed of the calculation should be relatively quick and the software shall receive this information quickly to display the data on the phone application. The data delivered from the glucose device shall be encrypted and secure from the public.
2. Environment
    - The software for the application shall be used as an instrumental and final tool that will show the output of the measurement for the user to visualize. At the end of the project, this software shall work as intentioned.
3. Characteristics
    - The software requirements that are created shall be correct, unambiguous, concise, and complete. In addition, the requirements shall be modifiable and verifiable for the designed to produce a functional software as stated in the design constraints.
4. Joint preparation
    - This portion states that not only does the supplier develop the software design process but the customer also must agree upon the terms of the software requested. In our case, there is no particular customer that the project is specifically geared towards but the software requirement specifications must conform to a typical user’s needs.
5. SRS evolution
    - As the software is developed and designed, the software may usually need to evolve due to unforeseen obstacles or circumstances. For that reason, our requirement specifications will be broad and modifiable so that the changing software aspects can be anticipated. Also, documenting a trace of the different steps taken to change the software will help us understand the final software product by looking at the previous steps taken.
6. Prototyping
    - Once the software reaches a satisfactory operation, prototyping will be the first step in determining the software’s progress as well as its discrepancies and issues. While prototyping the software, the system is observed as a whole and is determined if the software fulfills the requirements of an end user and if there are any

unanticipated outcomes from the behavior of the system. Noting these differences can guide us to go back and provide a solution for the encountered problems.

7. Embedding design
   - At this stage, the incorporation of the design features is initiated and the interface displayed on the phone can be focused on. Later on in the Design section, the software design layout is explained and the first step of designing the phone application was through computer drawings. This may not be considered a requirement but this portion is crucial for the user's ease and satisfaction.
8. Embedding project requirements
   - Although our primary focus on the requirement specifications and the design constraints involve the hardware portion of the project, the software requirements can be recognized also to implement a successful application that falls under each implicit and explicit requirement specification. Typically, some of the requirements considered are as follows:
     a) Cost
     b) Delivery Schedule
     c) Quality Assurance
     d) Validation and verification

Understanding these concepts of the software requirement specifications can guide us to produce a software interface that can display the information obtained from the glucose measurement device.

4.2 Design Constraints

The following pages describe the multiple categories of realistic constraints that guided the design of this project. These constraint parameters are formed in conjunction with the identification of the project's requirement specifications and are also decided based on the end user's needs, the manufacturer's resources, and a plethora of other outside factors. With the use of realistic constraints, the project becomes more challenging and this information is considered to make good decisions through gain of knowledge. These assessed constraints usually relate to the improvement of society and quality of life by observing the realistic factors such as:

- Economic
- Manufacturability
- Sustainability
- Environmental
- Health
- Safety
- Ethical
- Social
- Political

4.2.1 Economic and Time Constraints

With a financial sponsorship and project budget of $1,000, the goal is to aim for the cost of a singular glucose biotransducer prototype to amount to no more than half our budget ($500), which reduces the quality of parts such as the LEDs, the photodiode, the operational amplifiers, and several other components. Although the parts of the glucose sensor are compromised due to costs and resource availability, the economic implications could still be in favor of the glucose sensor. The United States market has blood glucose monitors that are only intended for blood glucose level measurement but our design measures glucose concentration in solutions through a different means. The cost to manufacture and develop this product would greatly decrease for a glucose monitoring system due to ease of resource availability and quicker assembly time. The goal was to use widely accessible products currently on the market and those items are easy to purchase and receive through shipping.

In addition to detecting glucose levels through a different method, our glucose sensor design could potentially decrease an end user's expenses in replacement of the glucose meters that require test strips every time the meter is used. The cost of each test strip for a glucose measurement meter costs anywhere from 40 cents to over $1 and by eliminating this singular cost for users, this money could be saved. Marketing this device as a tool that does not require reoccurring costs to customers who are either trying to save money or are financially unable to purchase test strips every so often.

The time schedule for the project begins in the Senior Design 1 course and ends near the last quarter of the Senior Design 2 course; in total, our project must be completed in April 2017, which is about 8 months from the end of August 2016. During this time, the design, prototyping, and implementation of this project must be completed and the product must be fully functional as to fulfill the engineering requirements. The glucose measurement device must be able to accurately determine if the user's glucose measurement is too high, too low, or satisfactory. To ensure that these goals are met, the timeline of the project described in the Milestones section and that Table explicitly lists each task and what time period is stated to complete the task in chronological order.

4.2.2 Manufacturability and Sustainability Constraints

Manufacturability is considered to be the art of designing a device or product that can be easily constructed or mass-produced. The manufacturing functions that makers should keep in mind consist of fabrication, assembly, testing, acquisition, shipping, and repair. Our goal with this project is to design a product that does not require too much in-house component manufacturing and that remains a simply made product that could take at most an hour to wire and piece together with components purchased from third-party manufacturers.

In the Mechanical section of this documentation, the earpiece design for the LED and photodiode installation will detail the specifics of how the earpiece can easily be modeled and possibly built through 3D printing. For the electrical circuitry and components, these parts will be bought through a few companies as these companies have already undergone the process of designing products with manufacturability as a key point. By purchasing these products from these companies, several benefits arise such as reduced costs, decreased lead times, higher quality, and increased reliability. The only two downsides to purchasing these parts is the time that it takes for the parts to be shipped and the cost of shipping.

A major concern of the noninvasive glucose measurement device relates to the product's reliability of its intended function. The sensor's sustainability plays a major role in whether the product will remain on the market or even if the product will survive. Many outside factors must be considered to prevent the downfall of the product's lifespan or else the glucose device will fail. Understanding the sustainability aspect of the project requires that all other design constraints are considered to clearly state the issues that arise. For example, the cost of the product, which is an economic consideration, could lead to the demise of its sustainability in the market.

When considering sustainability for this project, the parts used must be built from parts that are common and that can be easily ordered. Also, to minimize damage to the electrical boards, an electrical enclosure shall be designed and shall house the circuits responsible for the device's functions. Other than the constraints specified in the sustainability section, there are no other factors such as environmental elements that have a large effect on the glucose sensor.

4.2.3 Environmental, Health and Safety

It is extremely important for our device to meet certain Environmental, Health and Safety standards. Since our device would essentially be used for diabetes patients it would have to meet many requirements. The major topics that will be discussed is RoHS compliant parts, heat dissipation for health purposes and how LED effects your health.

RoHS is the acronym for Restriction of Hazardous Substances. RoHS originated in the European Union and restricts the use of specific hazardous materials found in electrical and electronic products. The substances banned under RoHS are lead (Pb), mercury (Hg), cadmium (Cd), hexavalent chromium (CrVI), polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE). The restricted materials are hazardous to the environment and pollute landfills, and are dangerous in terms of occupational exposure during manufacturing and disposal. When selecting electrical components it was important for us to follow this standard.

Another important aspect to look into for us was how LED affects the human body. After extensive research it was discovered that exposing the skin to near infrared red LED is not a large concern. Many of the research found mentioned that LED street lights could cause some health issues in the long term. A senior thesis by Stony Brook University Chemical Engineering undergraduates revealed facts about LED and skin contact. “The team exposed fibroblast cultures to four LED wavelengths—red, green, blue, and white—for intensity-normalized time periods, so all samples were being exposed to the same power within one test,” Rafailovich says. “The cells were counted using a hemocytometer; the mitochondrial activity was determined through MTS Assay analysis; and cell morphology was analyzed through confocal microscopy images. The cultures were exposed to two different conditions: longer exposure trials with Dulbecco’s Modified Eagle Medium [DMEM], with no phenol red on the samples, and trials without any media, a maximum exposure of 3 minutes.” It was concluded that the consistent trend of LED exposure is not harmful to human dermal fibroblasts. This was a great sign for our project design since it eliminated the stress factor.

Safety is and always will provide key constraints that cannot be ignored. The final prototype needs to be fully capable of housing the key functional components safely and prevent a user from being exposed to any risk. Our next concern to be aware of is how much heat dissipation our battery would produce. Since our device would essentially be small enough to fit in your pocket we would need to be aware of what temperature our device should reach. It is important that the device stays at a cool temperature such as the same a phone would have. This is also part of the safety issue that we would like to mitigate. A good PCB design will also help with this issue. Making sure that all the connection wires are separately correctly is key.

The Table below includes all the requirements and standards that must be set for our design.

Table 3: Power Requirements

#	Requirement	Reason
1	Low powered LED and Photodiode	To prevent any injuries when in contact with body. We will use 1550nm wavelength.
2	Control heat discharge	Device must not overheat since patient will be carrying it. Keep temperature under 90 degrees Fahrenheit.

4.2.4 Ethical, Social, Political

Our device must consider ethical, social, and political issues that could come into play from designing a noninvasive glucose meter. A main concern that will be discussed is how our device would allow easier and faster access for patients to check their blood at any time. Political issues will also be discussed such as FDA approval through clinical trials and health care companies providing coverage for a noninvasive device.

The biggest potential driving problems for people with diabetes, in the short-term, are low or high blood sugar episodes. An issue that comes with patients being able to check their glucose levels so easily with minimal effort is that it can be done in scenarios like driving. Our method of glucose monitoring could bring social issues in that way. For legal purposes our design would include a warning notice to users. This notice would mention how important it is to not check glucose levels while driving. It's important to check your blood glucose level before getting in the car, especially if you don't always feel the symptoms of hypoglycemia. On the other hand, blood sugar that is too high also can cause problems for drivers, including blurred vision. So it’s vital to keep your blood sugar in an optimal range while you’re driving.

A benefit that our device would have is that it would be eco-friendly. Normal glucose meters contain test trips that our thrown out after use. This eventually becomes very wasteful and potentially acts as a way of littering when patients cannot find a nearby trash bin. Our device would bring positive feedback, socially. GOLD would also eliminate the risk of having blood or open cuts affect other people.

Since our device would be an innovative solution to what is currently in the market many political concerns would arise. The main issue that would come up is making the device FDA approved. For the product to sell in the market it would have to be approved through many clinical trials. For this reason, we are not trying to put our product in the market at this time. Instead, we are looking to show that it is possible to measure glucose levels in the blood through a noninvasive method. Another issue that arises is that current glucose meters are so accurate. Bringing in a different method that is not as accurate limits the market potential. For users to feel like they may rely on this device it must be as accurate as the current invasive technology. Each country would have to make regulations as to what must be met to consider a glucose meter as accurate enough. This standard would have to be followed and met by our noninvasive method in order to properly and effectively join the market.

Another factor to consider politically is how health care companies will cover these devices. With the currently monopoly that exist for invasive glucose meters it could be hard to bring in our product.

5.0 System Design Details

This section will discuss the design details of our GOLD device. The sections that will be discussed are the power system, microcontroller, user interface, software design, mechanical design, AFE4404 evaluation module, and finally aesthetics.

An important part of our design is how the power of the system will be handled. Through much research, different ways were discussed that would benefit our system the most. The main components that we are looking to provide power for are the led, photodiode, MCU, ADC, and Bluetooth module. The AFE4404 has its own evaluation module that will work separately with power from a USB connection. The housing of the battery will also be discussed to determine how it will be implemented and designed on the PCB.

The microcontroller section will discuss how our PCB design will be integrated with a microcontroller to function with the sensors we are adding. Essentially the goal is to make a separate PCB that can be easily hooked on to the evaluation module so that we can utilize Texas Instruments AFE Chip. The user interface will consist of an android application that can view statistics for the glucose results. The software design and algorithms used to make this will be discussed. Full code and algorithm will not be provided for proprietary reasons. The mechanical design will discuss making a wearable earpiece. This would be beneficial for testing and allow stability. Ideally the mechanical design should also include a sensor patch version so that it could be easily removed and attached to the earpiece.

The AFE4404 evaluation module will also be discussed in this section. This has been the core of the GOLD device. Incorporating Texas Instrument's evaluation module to provide glucose-measuring results would be the ideal scenario. This section will discuss how that integration can be done. Finally, the aesthetics will be another important aspect of this design since the final target would be to a customer then it needs look aesthetically pleasing.

5.1 Power System

Our device will contain a mobile power system. This means that our battery will be integrated into our overall PCB design. It will be important to do this sort of integration because it allows the device to be more compact and useful. Usually power systems are described, as Electric power systems are comprised of components that produce electrical energy and transmit this energy to consumers. For our device the power system title will be used to describe the battery supply for the entire PCB design. The DC-DC design will be set up with two supply sources to step up or down the voltage that is needed for each load. The results that the program WEBENCH used will be reviewed to assure that they are accurate recommendations. This will be discussed further in the DC-DC section. The different values mentioned in the datasheet will be compared to the current values that are needed and also check if they match the findings in the WEBENCH results. Alternate software may be considered in the future to also understand what the most efficient way is to design a DC-DC circuit. Integrating the AFE4404 evaluation module with our power supply will not be necessary since that module already contains its own power management source. Expanding on what is already created would be the most beneficial and efficient route to go down through.

5.1.1 DC-DC Conversion

Texas Instruments Webench tool was used to prototype a design with the proper DC-to-DC part numbers. The way this tool was used was by inserting the voltage and current parameters for each load component. In this design a single cell battery of 3 volts was selected. Although not shown in design, another batter would be added in parallel to increase the voltage and run time for the circuit. Many factors were looked into for this circuit one of them being the efficiency factor. If the design had less than 100% efficiency then the supply would need to provide more storage. Picking loads that have similar voltages helped used fewer amounts of DC-to-DC converters. Table 4 relays the information from datasheets into a Table for a better understanding of why certain values were chosen. Table 4 below will show all the loads that were added to the Webench tool as part of final calculations. The information was taken directly from datasheets.

Table 4: Powered Devices in GOLD prototype

Qty.	Part	Nominal Voltage	I Max (Current, Amps)	Power (watts)
1	Bluetooth	3.3	0.04	0.132
1	MCU	3.6	0.006	0.0216
1	ADC	3.3	0.005	0.0165
1	LED	5	0.1	0.5
1	Photodiode	5	0.000002	0.00001
1	Op amp	12	0.0002	0.0024

As mentioned before Table 4 mentions the loads that are used for GOLD. The important part of these parameters was making sure to get a good range for voltage and current usage. By using the same voltage for devices then the amount of DC-to-DC converters were minimized. Minimizing this in our circuit design provides more cost efficiency since we no longer need to purchase more components to make the circuit work. Another important factor we must consider as well is to check the spec sheets of the recommended DC-to-DC parts of the circuit. Checking the specifications will help us verify that the parts chosen are the best option.

Table 5: Parameters Entered into Webench

Load	Load Name	Voltage (V)	Current (A)	Vout ripple
1	Bluetooth	3.3	0.04	5%
2	MCU	3.6	0.006	5%
3	ADC	3.3	0.005	5%
4	LED	5	0.1	5%
5	Photodiode	5	0.000002	5%
6	Op amp	12	0.0002	5%

With the information from Tables 4 and 5 loaded into the WEBENCH® program, the software then generates the overall schematics of the DC-DC conversion. This prototype is comprised up of two different voltage conversion stages to provide the proper voltage and current for various loads. Originally the efficiency was around 72% we decided to increase this to the highest possible efficiency in order to provide the most “working time” possible. As a result the efficiency increased to 96.11 percent. The increase in efficiency did not affect how many voltage converters were used but it did increase the price. For example, the price went from $1.70 to $3.99 dollars. This price is made up of nineteen components. Now instead of having a boost and buck we have a design with two boost converters. They would directly connect to the battery source and divide the power to at least two loads. Table 5 provides information for the power supply that was used. The power supply can be easily changed till final design is reached. The main concern of the project is to make sure we have all the main components connected and then the power can be decided. For this reason we will use a power supply bench to supply current power to our circuit. The power supply will have a current limit so the circuit doesn’t receive too much current. If too much current is applied to the circuit then it can burn components.

Table 6: Battery input from Webench

Voltage (V)	Calculated Current Max (A)
Minimum: 2.4 Maximum: 3.0	10

The Webench program by Texas Instruments provided a complete Table that summarizes the specific DC-DC conversion components used and their respective performance details. This info is shown in Table 6. A high level display of all seven DC-DC stages and schematics are shown in Figures 8-10. Table 5 will include the part number of each symbol used for in the schematics. Table 6 will describe the operating values of the DC-DC stages. It is important to consider each operating value because it determines how the circuit will operate.

Table 7: DC-to-DC Converter Characteristics

#	Name	Part Number	Description	Vout	Iout	Efficiency	Footprint	Cost
1	SUPPLY_1	TPS61236P	Switcher: TPS6123x 7.5-A Valley Current Synchronous Boost Converter w/ Constant Current Output Feature	5 V	0.1 A	96.30%	152	$2.50
2	SUPPLY_2	TPS61021A	Switcher: High Efficiency Boost Converter	3.3 V	0.051 A	95.50%	126	$1.49

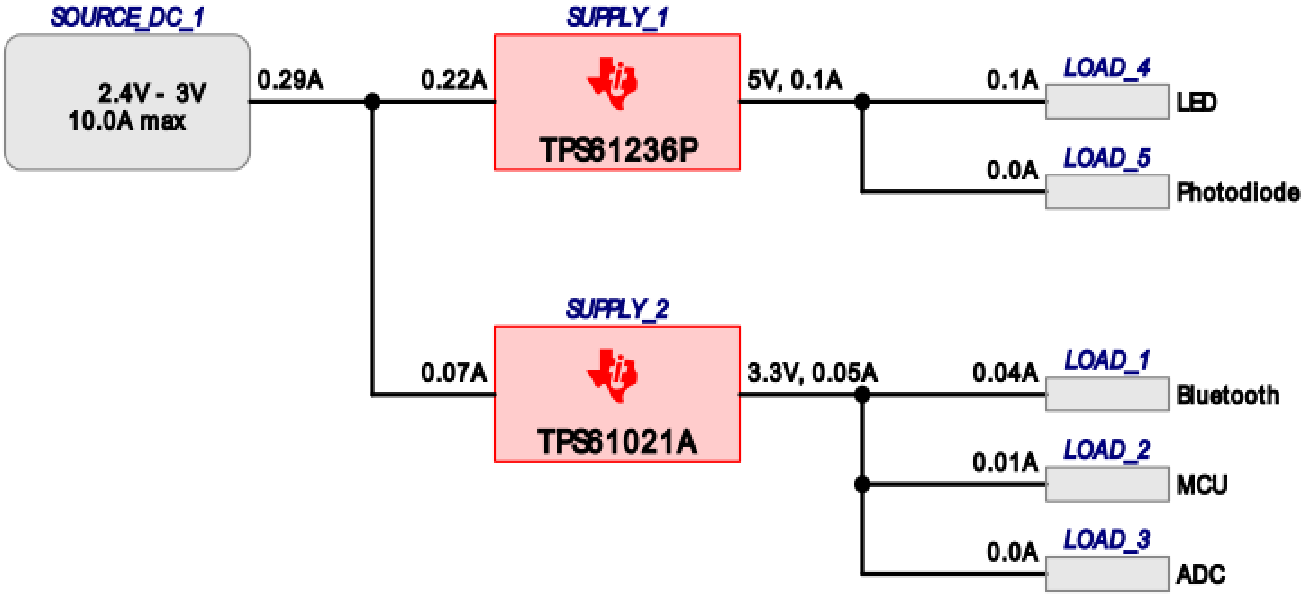


Figure 18: DC-to-DC Conversion Layout

As mentioned before, Figure 18 shows the overall view of the DC-to-DC block diagram. The source starts by coming in with 2.4 to 3 volts. The current that flows out of the source is at 0.29 Amps. This source is then automatically split to the two power supplies that will step either up or down the connected loads. It is important to note that the efficiency of the circuit is so high because supply 2 connects directly to the battery source. Modifying the different characteristics on WEBENCH will provide a different block diagram. The first power supply draws in 0.22 amps this is because it must step up the voltage to 5 volts from the original 3 volts. It is obvious to see that supply 1 is a step up converter. The load that is consuming the most current in that circuit is the LED. Since the LED has such a high wavelength then it requires a lot of current. Notice that the photodiode does not need any current since it is acting as receiver. The next component is supply 2. As seen in the diagram it is taking in 0.07 amps to step up the voltage to 3.3 volts.

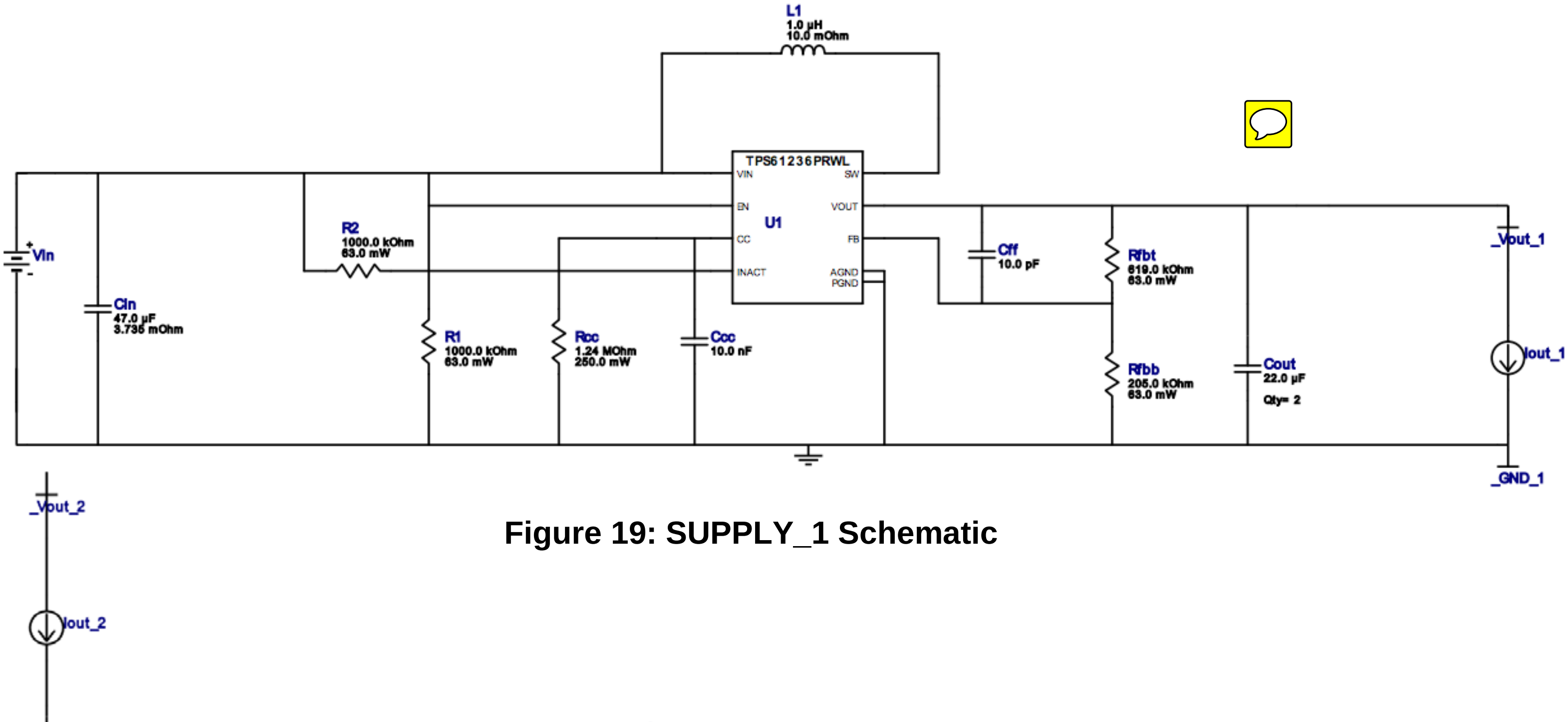


Figure 19: SUPPLY_1 Schematic

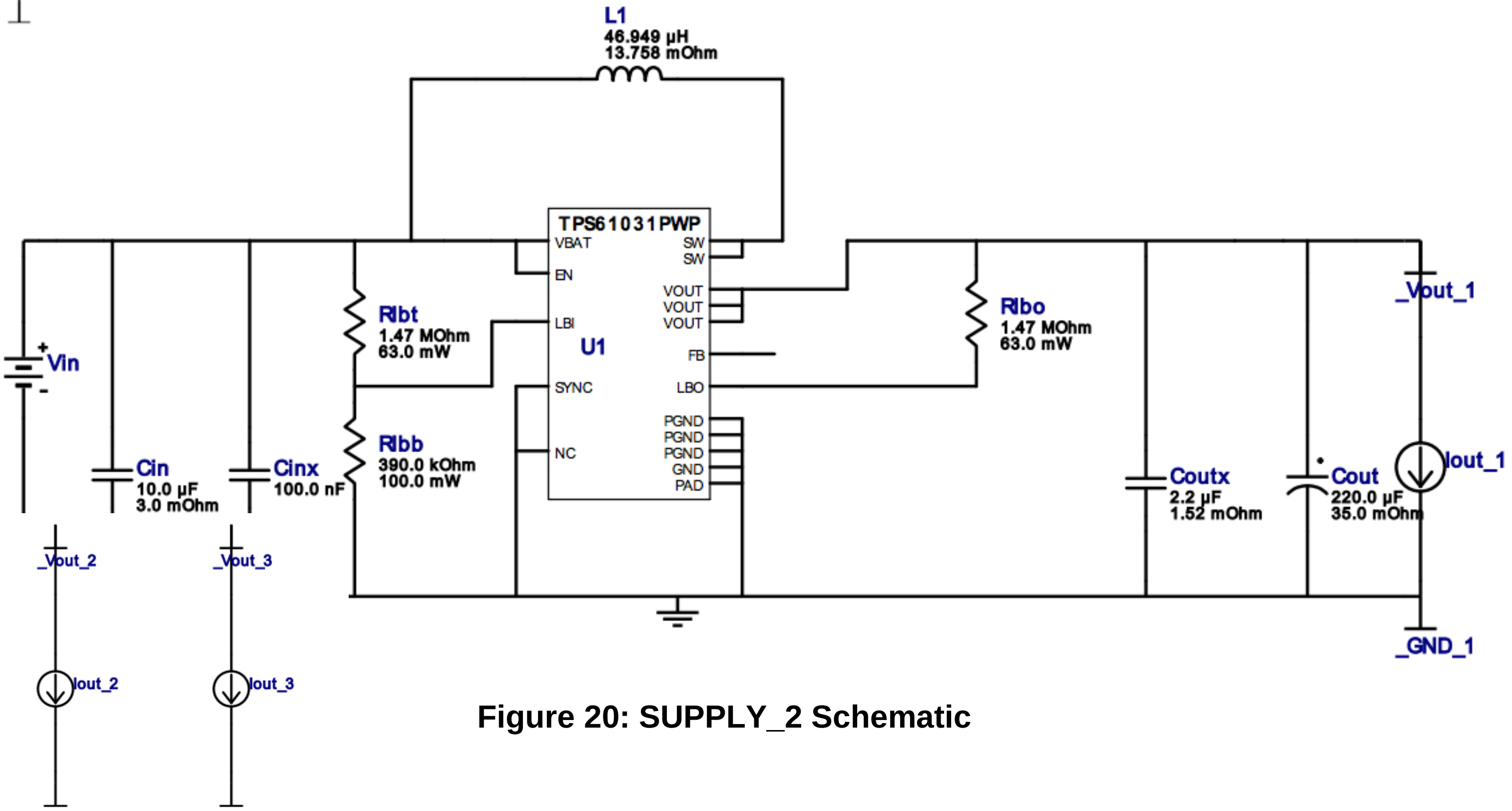


Figure 20: SUPPLY_2 Schematic

Table 8: Operating Values for Supply 1

Name	Value	Category	Description
Cin IRMS	0.360266568	Current	Input capacitor RMS ripple current
Cout IRMS	0.517699639	Current	Output capacitor RMS ripple current
IC Ipk	0.832745241	Current	Peak switch current in IC
Iin Avg	0.21631	Current	Average input current
L Ipp	1.248	Current	Peak-to-peak inductor ripple current
BOM Count	12	General	Total Design BOM count
FootPrint	152	General	Total Foot Print Area of BOM components
Frequency	343114.3806	General	Switching frequency
Mode	BOOST PFM	General	PWM/PFM Mode
Pout	0.5	General	Total output power
Total BOM	2.5	General	Total BOM Cost
Vout Actual	5.000273171	General	Vout Actual calculated based on selected voltage divider resistors
Vout Tolerance	3.1497	General	Vout Tolerance based on IC Tolerance (no load) and voltage divider resistors if applicable
Duty Cycle	52.0990082	Op_Point	Duty cycle
Efficiency	96.312	Op_Point	Steady state efficiency
IC Tj	30.27319418	Op_Point	IC junction temperature
ICThetaJA	28.7	Op_Point	IC junction-to-ambient thermal resistance
IOUT_OP	0.1	Op_Point	Iout operating point
VIN_OP	2.4	Op_Point	Vin operating point
Vout p-p	0.002715152	Op_Point	Peak-to-peak output ripple voltage
Cin Pd	4.85E-04	Power	Input capacitor power dissipation
Cout Pd	0	Power	Output capacitor power dissipation
IC Pd	0.009518961	Power	IC power dissipation
L Pd	0.009111398	Power	Inductor power dissipation
Total Pd	0.019146031	Power	Total Power Dissipation
Rload_crit	50	Unknown	Minimum Rload required during Start up

The Table above describes the different operating values for the DC-DC circuit. These values are important because they will determine how well the circuit can function. Calculating these values by hand will not be necessary since the WEBENCH software already does that for you. It will be important to compare these values with the operation requirements that each load needs to work properly.

Table 9: BOM List for DC-DC

Part	Manufacturer	Part Number	Quantity
Ccc	MuRata	GRM033R61A103KA01D	1
Cff	Samsung Electro-Mechanics	CL02C100JO2ANNC	1
Cin	MuRata	GRM31CR60J476ME19L	1
Cout	Taiyo Yuden	LMK212BJ226MG-T	2
L1	Bourns	SRP7030-1R0M	1
R1	Vishay-Dale	CRCW04021M00FKED	1
R2	Vishay-Dale	CRCW04021M00FKED	1
Rcc	Vishay-Dale	CRCW12061M24FKEA	1
Rfbb	Vishay-Dale	CRCW0402205KFKED	1
Rfbt	Vishay-Dale	CRCW0402619KFKED	1
U1	Texas Instruments	TPS61236PRWLR	1

Table 9 list specifications for the DC-DC circuits. BOM is the bill of materials that make up the circuit, which are twelve. The duty cycle value of 52 for the op amp is important because it means it is half off and half on. The operating values of supply 1 are important because it determines how well the circuit will function. Comparing the resulting to values to what the loads need will be key to building a good circuit. The efficiency of the circuit is very high, exactly 97%. Depending on the efficiency and level of voltage change, the number of components required will vary. When designing a printed circuit board (PCB), board space is an important consideration as well as overall number of components. Throughout the DC/DC conversion section, the difference between a 5%-10% in efficiency may mean a difference of 200 components and a change in cost. For the design, it was important to consider the efficiency and footprint as well. The footprint will determine how many components will be in the circuit. To keep the board compact it is important to have few components. The op amp acts a way of amplifying the power in the circuit.

Table 10 will have the operating values for Supply 2. The importance of some of these values will be discussed. For example, the Cin IRMS is the input capacitor RMS ripple current. It is important that input capacitors have low ripple voltage amplitude seen at the input of the module. This reduces the RMS ripple current to a level, which can be handled by bulk capacitors. Ceramic capacitors placed right at the input of the regulator reduce ripple voltage amplitude. Only ceramics have the extremely low ESR that is needed to reduce the ripple voltage amplitude. These capacitors must be placed close to the regulator input pins to be effective.

Even a few nanohenries of lost inductance in the capacitor current path raises the impedance at the switching frequency to levels that negate their effectiveness. Large bulk capacitors do not reduce ripple voltage. Large input ripple voltage can cause large amounts of ripple current to flow in the bulk capacitors, causing excessive power dissipation in the ESR parasitic. To reduce the RMS current in the bulk capacitors the ripple voltage amplitude must be reduced using ceramic capacitors.

Table 10: Operating Values for Supply 2

Name	Value	Category	Description
Cin IRMS	0.006794581	Current	Input capacitor RMS ripple current
Cout IRMS	0.028824804	Current	Output capacitor RMS ripple current
Coutx IRMS	0.007140028	Current	Output capacitor_x RMS ripple current
IC Ipk	0.082268843	Current	Peak switch current in IC
Iin Avg	0.071777	Current	Average input current
L Ipp	0.023537	Current	Peak-to-peak inductor ripple current
BOM Count	9	General	Total Design BOM count
FootPrint	186	General	Total Foot Print Area of BOM components
Frequency	600000	General	Switching frequency
Mode	PWM CCM	General	PWM/PFM Mode
Pout	0.1683	General	Total output power
Total BOM	0	General	Total BOM Cost
Vout Tolerance	0.60606	General	Vout Tolerance based on IC Tolerance (no load) and voltage divider resistors if applicable
Duty Cycle	27.6598652	Op_Point	Duty cycle
Efficiency	97.698	Op_Point	Steady state efficiency
IC Tj	30.18091441	Op_Point	IC junction temperature
ICThetaJA	46.9	Op_Point	IC junction-to-ambient thermal resistance
IOUT_OP	0.051	Op_Point	Iout operating point
VIN_OP	2.4	Op_Point	Vin operating point
Vout p-p	8.24E-04	Op_Point	Peak-to-peak output ripple voltage
Cin Pd	1.38E-07	Power	Input capacitor power dissipation
Cout Pd	2.91E-05	Power	Output capacitor power dissipation
Coutx Pd	7.75E-08	Power	Output capacitor_x power loss
IC Pd	0.00385745	Power	IC power dissipation
L Pd	6.90E-05	Power	Inductor power dissipation
Total Pd	0.003965536	Power	Total Power Dissipation

5.1.2 Battery Housing

The battery will be placed on the PCB since it is a coin cell battery therefore small enough to fit on the PCB. The surface mount coin cell battery holder will be bought from Keystone it is part number 1061TR. The coin cell battery holder holds two batteries. Currently the design is set to work with one battery but in the future it may be changed to work with two depending on how much current is drawn per load. Considering what kind of mount the battery holder should have depends on how the soldering will be done. If the PCB board can be taken to a professional for soldering purposes then it would not matter how the pins looked like. Most professional PCB making stores will have machines that are very precise and provide a clean finish of the final product. Determine what pins will be used to connect the battery holder will be determined when discussing PCB and MCU design specifications. Essentially the battery will provide all the power to the MCU and the rest of the components on the board so it will be necessary to connect it properly so that the PCB can function at the highest level of performance. The battery cell holder size is 20 mm. The battery that we will use also has a size of 20 mm. It is a Panasonic CR2032 non-rechargeable lithium battery with 225 mAh.

5.1.3 Battery Design Details

Our current design is composed of two cell batteries with each at 3 volts. It was important for our design to stay as compact as possible which is why we are using coin cell batteries. This battery is key to the entire design since it provides power for the entire system. The equation 1 below was used to calculate the amp rating of the battery. This equation can also be manipulated to give how much time the system would be able to run for during full operating use. For the purpose of prototyping, this device we are looking to have a minimum of 2 hours operating. Currently in our design one battery will use a total of 289 mA. The specifications for a CR2032 battery mention it will operate with 224 mAh. This means that if we divide 224 by 289 and then multiply it by 100 then the result will be 77 minutes of full operation. To add more operating time a second battery in parallel will be added. For demo purposes 77 minutes is plenty.

$$[\mathit{Amps\ used\ in\ Load}]\ x[\mathit{\#of\ Hours\ Needed\ to\ Run\ for}] = [\mathit{Amp\ Hour\ Rating\ on\ Battery}]$$

Equation 1: Amp Rating

While more design and integration of the system occurs then power supply will be adjusted. A big factor that can make an impact on our design is using the AFE4404 evaluation module. Since the evaluation module already contains a power management circuit then it could be useful to just rely on that same integration for the entire power. Figure 19 describes the setup of the power distribution. The coin cell will most likely change throughout more product development as we find better solutions to power the system. Firstly, it is important to have a schematic of the entire system so that the battery determination can be considered better. When the PCB design section comes along it will be important to discuss how heat transfer can immensely affect the entire circuit. The correct current path widths will have to be determined to prevent short-circuiting any components.

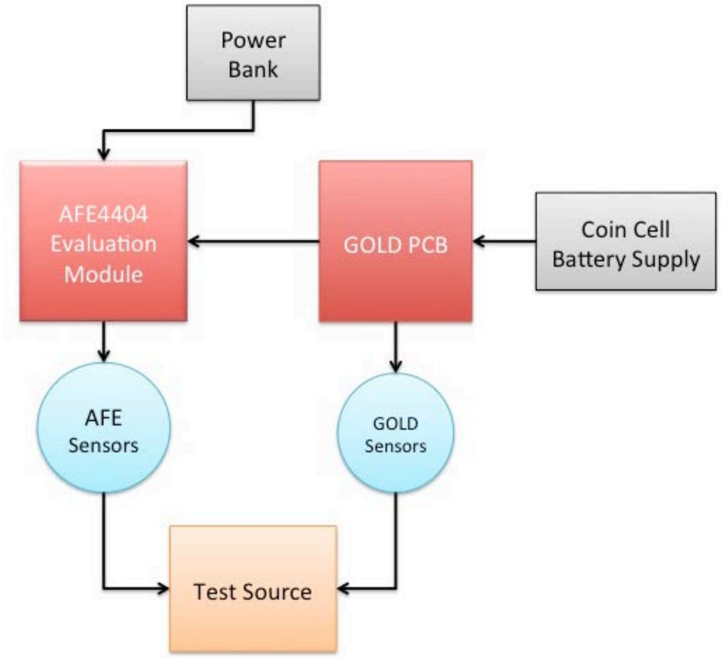


Figure 21: Power Distribution Block Diagram

5.1.4 AFE4404 Power Supply

To make the project wireless then a battery component would need to be added to the evaluation module. The best way to do this would be by making a portable USB port battery. A battery that provides at least 3.3 volts would work as a power mechanism. Perhaps going the route of purchasing a power bank would be the most beneficial and fastest. Portable Power Banks are comprised of a special battery in a special case with a special circuit to control power flow. They allow you to store electrical energy (deposit it in the bank) and then later use it to charge up a device (withdraw it from the bank). Using a power bank would also help reduce the noise. The power source that enters the evaluation module cannot be AC because this would then introduce noise to the circuit. AC power would mean plugging in a power source into an outlet. The evaluation module includes a battery fuel gauge, which would protect such things as over-temperature, under-temperature, and additional levels of over/under-voltage, discharge short circuit, discharge over-current, and charge over-current. This is a key component to making sure the battery doesn’t over charge.

The power supply for the evaluation module has to be considered since it is currently set to get power from a USB cable connected to a laptop or computer. The schematic for the AFE4404 describes the power as the following: AFE4404 can operate from 2.0-V to 3.6-V Rx analog supply (RX_SUP), 1.8-V to 3.6-V IO digital supply (IO_SUP), and 3.0-V to 5.25-V Tx Control supply (TX_SUP). The power for the board is derived from the USB input (J1) through a forward-biased diode (D1) to avoid reverse current flow. The USB data bus is ESD protected using TI's ESD protection diode array TPD4E004DRYR (U5). The USB VBUS is fed to the integrated Li-Ion linear charger and system power path management module, bq24032ARHLR (U3), which generates greater than 4.2-V output (VCC_BAT). This output is fed to TI's low-input boost converter with integrated power diode and input/output isolation, TPS61093 (U7), for generating a boosted voltage of 8.97 V. This output is fed to low-noise voltage regulator LP3878-ADJ (U9) for generating 5 V for TX_SUP. The VCC_BAT is also fed to the ultra low noise linear voltage regulator TPS7A4901DGN (U6) for generating 3V for the MSP_DVCC, MSP_AVCC, RX_SUP and IO_SUP. Series jumper resistors are provided to make sure the power supplies to the board are correct.

Another method that can be implemented for the AFE4404 evaluation module is leaving it connected via USB to a laptop. The information could be gathered and then transferred over to the mobile application. This data could be used like a look up Table for predefined test sources. Once these test sources have been determined then changing the results would already be incorporated in the mobile application. The problem with taking this approach is that the product would no longer be wireless from the power source. This would take away from the initial design requirements. Overall, this project is being developed to show a proof of concept that near infrared LED can be used to determine glucose levels in the body. At the moment, any method will be considered for full implementation depending on what gives better results. Connecting the evaluation module to the PCB could introduce unnecessary noise since the pins would be so close together. If this is the case then the implementation of making an entire new PCB design or perhaps using the sensors separately will need to be looked at and reviewed.

5.2 Microcontroller

The MSP430F5529 serves to be one half of the nervous system of this design, functioning as the spinal cord and the eyes, ears, and extremities. The purpose of using this controller is to act as the bridging gap between the data acquisition and the communication between the brain of the device which is the mobile application. The intensive process of the signal correlation, and mathematical transformations will be done on a server-based application. The MSP430 will do the signal sensing and signal smoothing for that data to be properly interpreted. We can think of the spinal cord as being the channels that will send the data. The eyes and ears will be the NIR LED control with the PWM and the photodiode sensing. The extremities are the external peripherals that the user will interact with the control the device. These 'organs' will be used together to create an immersive product that will be able to do testing and sensing, communication, and interaction thus creating a controlled feedback system.

5.2.1 Peripherals

This section will discuss all the peripherals that will be include in the system design and how they will be configured to the microcontroller, how they will be programmed, and how they will communicate back with the microcontroller. The included sections will discuss the three different classes of peripherals that will be included in the system design, control peripherals, communication peripherals, and finally the external IC peripherals. Control peripherals will discuss the hardware that will be used to interface the device with the user. The communication peripherals will discuss the Bluetooth module and how it will interface using wireless communication. The external peripherals will go over all external IC devices that will be needed to be utilized in the system.

5.2.1.1 Control Peripherals

The commands that are sent to the microcontroller will dictate how the device will be used. In the case for using an MSP430 it is easy to set up LEDs, speakers, Switches, Pushbuttons, temperature sensors, light sensors, and motors. As these peripherals become more complicated the devices become more complex in how they communicate with the microcontroller. Peripherals such as Analog to digital converters, LED drivers, digital to analog converters, cameras, wireless modules, servos, digital potentiometers, and etc. will require both more complicated software and more complex wiring schemes. The complexed peripherals might need specialized signals such as PWM signals, synchronized clocking methods, and drivers to properly connect the external devices. Below I will include the basic code for setting up pin functionality.

Table 11: Basic configuration commands for MSP430 [36]

Function	Description
PxSEL	This register selects with in GPIO mode or internal peripheral mode
PxDIR	If set for GPIO mode this pin will set either Input or Output
PxOUT	If set in GPIO mode while set as an Output this will set the pin as HIGH or LOW logic.
PxIN	If set in GPIO mode while set as an Input this will set the pin as HIGH or LOW logic.
PxIFG	Interrupt flag for corresponding I/O and is set when the appropriate signal condition is made
PxIES	Interrupt transition selection register.
PxIE	Interrupt enable register
PxRES	Internal pull up resistor is enable when bit is set to 1, and internal pull down is enabled when bit is set to 0.

Interfacing to the microcontroller is one of the most important concepts in utilizing the microcontroller. By interfacing the microcontroller with the user in a simple manner we can effectively reduce set up time, and make it possible for children to use the product. It will important to keep the operation very simple, and easy to understand. Therefore simply by having three pushbutton switches the user can interact with the device to turn ON/OFF the device, set the device into standby mode, and lastly set the hardware into testing mode. Any external features such as calendar and medical feedback will be standalone on the mobile application. The speaker can indicate when the testing is done or notify the user that they need to test again. The LED indicators can notify the user when the device is on, in Standby mode, Testing mode, and when the testing is complete. For instance, the Green light means the device is on, the yellow light means the device is in idle or standby mode, the Red light means the device is testing and the blue means that the test is complete.

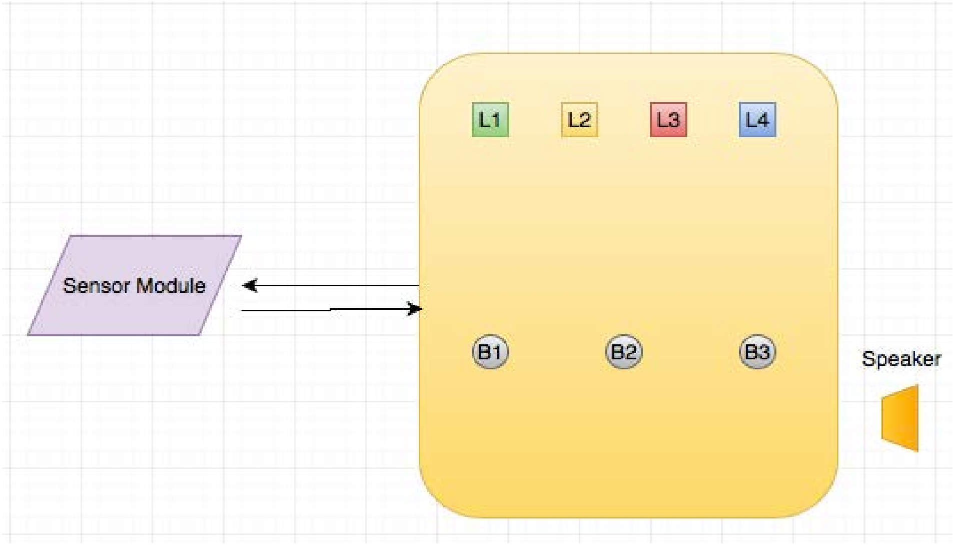
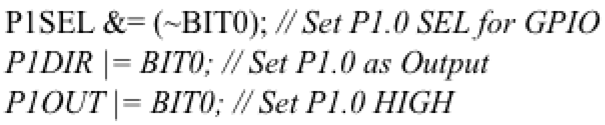


Figure 22: Block Diagram for User Interface

This interface will be simple enough for young users to control and sufficient so the device can capability function The MSP430 design uses general-purpose input/output pins (GPIO) to link external devices as either inputs or outputs. The devices listed in the first Figure in this section will set LED one through four and the speaker as output devices, and set the push buttons one through three as input devices. To configure these devices properly you must connect the device to a specific pin and then by setting up the pins to communicate either as input or output the device will send commands or receive data from each peripheral. An example of setting pin P1.0 as Output is listed below.



```
P1SEL &= (~BIT0); // Set P1.0 SEL for GPIO
P1DIR |= BIT0; // Set P1.0 as Output
P1OUT |= BIT0; // Set P1.0 HIGH
```

Figure 23: Pin Configuration for LED

By setting a GPIO pin as an output it is possible to control a device such as an LED or switch. Our code will send a HIGH value to any LED to turn the device on using a series resistor to protect the diode from overcurrent. Then we can turn the device off if a LOW signal is sent to the device. Device can use interrupts to help determine what actions to take while receiving commands from inputs. For instance, button 1 will turn on the device by creating a connection between the power supply and the main power line to the microcontroller. Button 2 will tell the device to generate an interrupt to configure the device into a low power mode with standby. Button 3 will generate an interrupt to tell the device to either wake up from standby followed by testing or just to begin testing. Each interrupt will generate the corresponding signal for the LED indicators. The indicators' resistance will be dependent on the voltage given to the LEDs and if needed, a LED driver can be used to dim the LEDs or light them brighter. The interfacing between the LEDs is a simple process that will only consist of Boolean logic code that will indicate whether the LED should be ON or OFF.

Configuring our pushbutton properly will prevent any jitters from the button and prevent false signals from arising. The typical way to trigger a pushbutton is using a falling edge, in which the pushbutton is configured between the pin it is connected to and ground. By pressing the button, the signal will thus fall. The issue with using momentary pushbuttons is that the signal is sometimes faulty and when pressed will generate multiple signal falls, and that the settling time is not instant. To do this the button needs to be biased to Vcc, which requires a pull-up resistor typically about 100k. The pull-up resistor is typically a large value to reduce the wasted current. To debounce the signal we will use a capacitor placed in parallel with the switch. In this configuration when the pin is open, the capacitor will charge to Vcc and then when the pin is closed the capacitor will discharge to ground but with a transient signal. That transient signal will be sufficient to smooth out the bouncing of the mechanical switch, since the voltage will drop off quickly only one transient will be seen by the Input. The typical response of the RC circuit are dependent on both the value of the resistor and the capacitor but a value of 1uf to 10uf should be sufficient to give a long enough response to give a single transient. [37]

The MSP430's clocks will be the heartbeat that will control the communication between the peripherals and the MSP430. "CPUs require clocks to run the CPU since asynchronous operation is not possible in computer processing."[26] Therefore MCUs are often flexible in the clock methods and include many clock sources. In the MSP430, there are three clock sources that are available for the CPU and its modules, the MCLK, SMCLK, and the ACLK. The ACLK, MCLK, and SMCLK can be selected for the peripherals,but the MCLK is reserved for the CPU and system. The MSP430 features a system called the Universal Asynchronous Receiver/Transmitter for to take bytes of data to send them serially and the receiver of the data will reconstruct the bits to represent the original data. The each UART includes a shift register to go from parallel to serial data forms. UART is implemented using the TX and RX lines for communication. "MSP430 families contain different peripherals capable of UART communications. Among these are USCI (Universal Serial Communication Interface), USART (Universal Synchronous/Asynchronous Receiver/Transmitter) and eUSCI (Extended Universal Serial Communication Interface) modules." [38] The general method that the UART odules work is that, "two main interrupt sources: RX and TX. The

RX interrupt fires when a character is received and has been placed into the buffer, whereas the TX interrupt is set when the TX buffer is available to be filled with a data.” [38]

5.2.1.2 Communication Peripherals

Our means of wireless communication will be based on an external Bluetooth module that will be interfaced to the microcontroller. Bluetooth is a wireless communication standard that uses radio frequencies ranging from 2.4GHz to 2.48GHz to transmit data through a personal area network securely. “Selecting the right Bluetooth module for one´s application depends on the data rate, range and maximum power consumption of the module the developer expects.” [38] The procedure of properly connecting the device can be done in three phases. First is the hardware setup, second is software programming, and then third is testing. This section will cover both hardware setup and software programming.

Bluetooth is the peripheral means of communication between the MCU and the mobile application and server. To connect the Bluetooth module with UART, wire the RX and TX pins on the Bluetooth module to the Rx and TX pins on the MSP430, which will serve as the communication bus, and connect Vcc and GND on the Bluetooth to the proper 5V, or 3.3V and GND of the MSP430. The Bluetooth module needs an antennae included to communicate and transmit the communication signal.

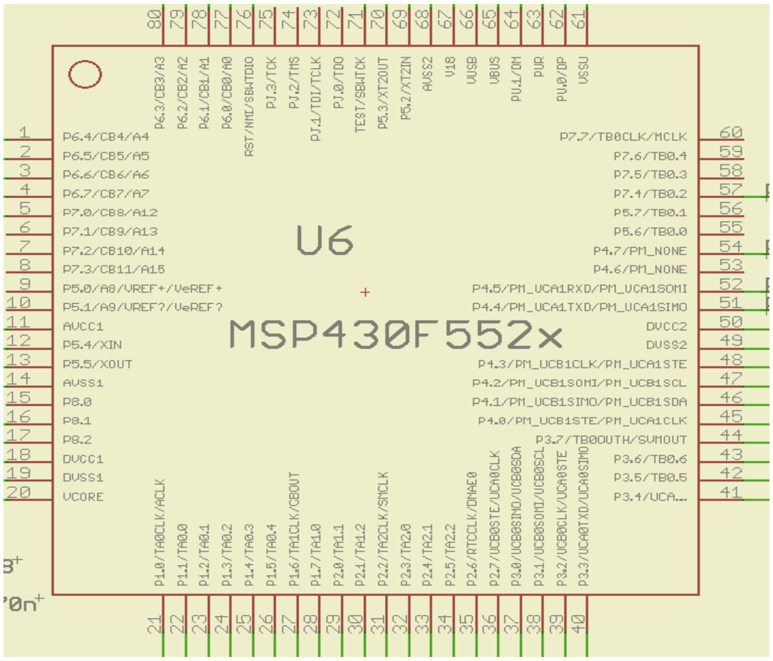


Figure 24: Pin Layout of MSP430F5529

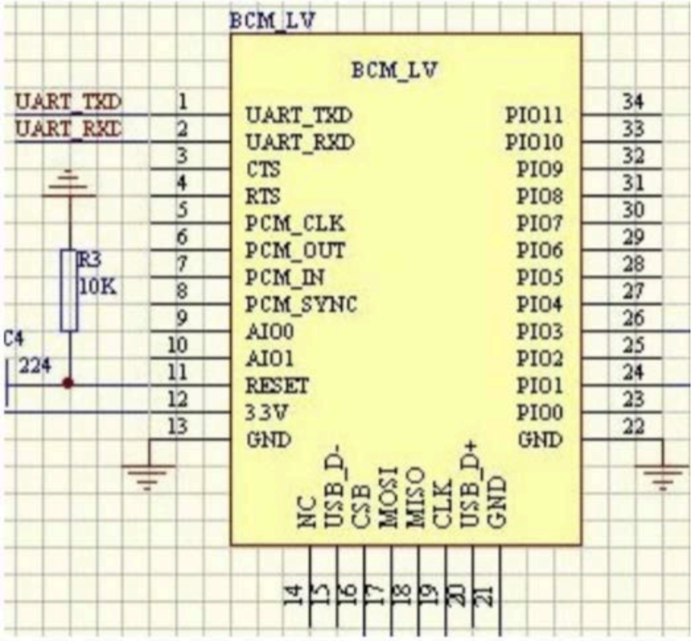


Figure 25: HC-06 Bluetooth pin layout courtesy of Olimex

The general pin layout will be configured that the UART TX & RX pins will be wired to the UCATX and UCARX pins on the MSP430, and finally the CTS will wire to TA1.0 as a UART clear to send with an active low configuration, and RTS will wire to TA1CLK as a UART request to send with an active low configuration. In using UART these are the most important pins to interface aside from the GND and VCC between the module and the MCU. The other pins are used for SPI communication and are not needed. The CTS and RTS pins control when the module is enabled to send and receive data. PIO or AIO pin are programmable lines for the Bluetooth module for testing, and will be used to verify that data is being sent when in testing. P11 or RESET will be used for resetting the Bluetooth module and will be configured as an active low set up to reset the module when needed. For stability of the device, it is important to use decoupling capacitors between the VCC of the Bluetooth module and the ground, 1uF and 10uF in parallel are recommended. “This improves the setup (circuit) in such a way that noise coming from the voltage supply and due to the connection with MSP430 was shunted through the capacitor. In addition, it supports the huge current demand (up to 100mA) during transmission.” [38]

5.2.1.3 External Peripherals

The external peripherals will be more complexed to setup than the control peripherals since they will require multiple connector pins. The external peripherals that are planned to be used are a high resolution ADC connected to a photodiode(s), and an LED driver for the NIR infrared LED. Another external peripheral that will be used is the AFE4404 microchip if it is not integrated into our PCB design, however this interfacing will be discussed in the section about the

AFE to MSP430 communication. This section will discuss how each external peripheral will be implemented in hardware and software.

5.2.1.3.1 LED Driver Controls

The importance of the LED driver is to drive the NIR LED so that we can apply different amounts of power to the LED in attempt to shift the emitted frequency. The driver will serve to be a constant current generator and will be controlled via a PWM signal to the LED driver control pin. The device that will be chosen is dependent on the amount of power that is delivered and should match the ratings of the LED. By sending a small duty signal and sweeping it to a larger duty cycle will enable us to sweep the frequency emitted through the NIR LED. The connection to the LED drive generally goes as such, the Vin will be powered by the 5V supply of the MSP, the DIM/EN pin will be fed the PWM signal from the MSP430 pin that and this will control the constant current, the power to the led will come from the LED driver and will use Zener and regular diodes for the protection of the circuit. The LED will be connected in parallel with a bypass capacitor to prevent and sudden voltage changes in the LED, and then in series with a resistor to ground to dissipate and additional power. It is very important to take these overcurrent measures to protect the LED from overdrawing and potentially damaging the part. Using the LED drive will only a few number of GPIO pins to power multiple LEDs simultaneously and thus will allow our group the ability to expand on the LED layout. The software for the LED driver will have functions for different duty cycles to call. This is an important part of implementing creating a fully controllable system. As the PWM duty cycle will determine the frequency shift of the LED’s emitting light it will be important for generating more conclusive datasets. There reasoning is that once data acquisition is initiated by using different functions for different duty cycles the data sets that are returned to the server through Bluetooth can be matched to the frequency setting that is implemented. That will allow correlations to be matched better since that absorption model of glucose varies with frequency. Thus by matching each data set to the emitted frequency the correlation estimations can be more precise and will aid greatly in narrowing down false positives and true negative correlations. This being said creating the software for implementing the LED driver will be based on different PWM duty cycles and different clock settings for the PWM signal. These settings can be pulled from the driver that will be linked as a header file to for MSP430 operating modes. The configuration will be sent to the server first and then the server will start receiving data until sufficient data samples are collected. Once that process is complete then the next PWM setting will be implemented and then data acquisition and transmission will occur again. This will be repeated until a full sweep of different duty cycle stimulations occurs and all data sets are transmitted.

5.2.1.3.2 ADC Module

The ADC module is important for converting the signal received from the photodiode into a high resolution signal that can detect a minute change in voltage.

That is why our group has chosen to implement an ADC module that can implement up to 24 bits resolution. Below is the basic calculation of the analog step value based on n-bit analog to digital conversion using Vref as listed in the equation below.

$$Vstep = Vref/2^n$$

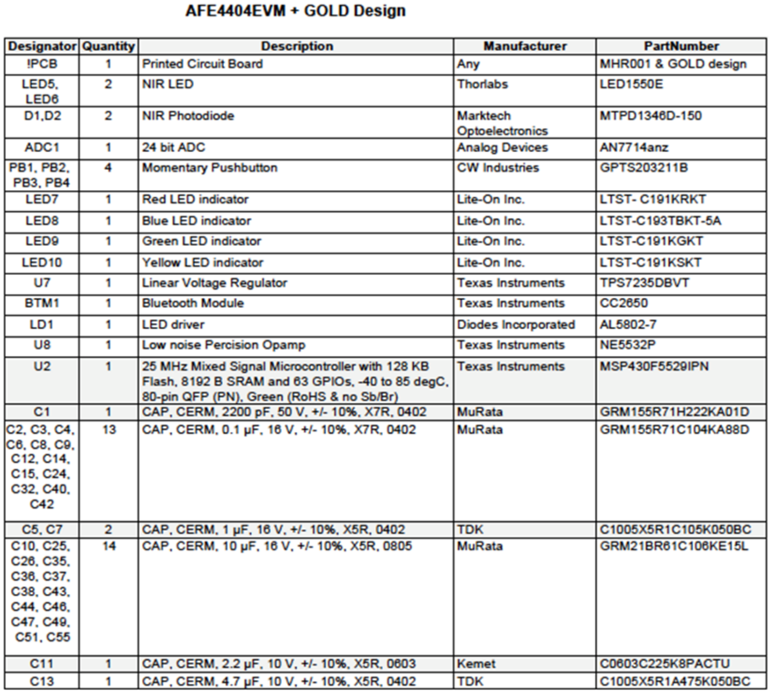
Equation ###: Calculation for Analog Step Value for ADC Module

Creating the proper pin connection will be important for properly using the ADC module, and typically the ADC integrated circuit will implement many features such as an offset shift correction, multiple types of filters for low, high, and band pass filtering and notch filtering, and internal programmable gains. To use all these features the device will have to be configured through the MSP430 by creating commands. Another feature that should be discussed is that often this ADC include internal memory and this will require that the communication between the ADC and MSP430 should be clocked properly. The basic pin lay out for the ADC will require that the power be supplied by the MSP430 and the GND will be connected together. Communication between the MSP430 and the AD7714 will be using SPI communication protocol. The ADC module will source from a clock on the MSP430 and will use the SDA and SCL to communicate with the microcontroller, sending commands and clock signals so that the device is synchronized during data transfer. The ADC module will use another clock for its internal analog to digital conversion. The device will also need to be powered by the MSP430; therefore, it will source the 5V or 3.3V source and ground from the MSP430. There is also pins for device reset, device standby, and device synchronization that can be controlled through using GPIO pins. The external features will be wired also to GPIO pins to configure them as applications call. By connecting all our analog signals to analog input pins on the ADC module, we will set up the device to read incoming analog signals. Indicator pins that can trigger a HIGH or LOW logic after either a byte or word is sent is important for our applications as well therefore the ADC module will notify the MSP430 as to when data is completely sent. This is very important since the ADC will return values of 24 bits and the MSP430 uses 16-bit registers therefore properly sending data is very important. Using on board features for filtering and offset is important for utilizing the ADC, therefore creating functions for setting up the ADC in many device configurations is important.

5.2.1.4 Software Link with Peripherals

The first step in creating the software setup for the Bluetooth module is to create the API hardware interface for the MSP530F5529. The API will configure the pins, timers, oscillators, and system clock that are needed. The actual register words that needed to be sent for each setup will be covered in the main software setup section. In the API we will set up different baud rates, and clock dividers. The API will act as a header file for when we want to utilize the different modes of operation. Once the API is complete with all the proper commands to enable different modes, a main file can be used in order to implement the settings by calling each specific function. For instance when a Button 1 is pressed it is desired so the Bluetooth module is unused therefore the called function to turn off power to the Bluetooth device will be used and for instance Timer_A can be turned off. If button two is pressed then the device will enter Low power Bluetooth mode and use a low frequency clock or a digital clock such as the DCO or utilize the FLL. If button three is press and then the device will enter UART communication mode and then supply the power to the module and begin sending data as needed and use the appropriate transmission rate, which will be the baud rate of 9600. Assigning the proper communication baud rate is important for the main receiver to interpret the data, and therefore we assign the baud rate according to the mode the Bluetooth should be used, where in client or host mode. Interrupts will be a major portion of utilizing the proper settings and it is important that when an instance is set that the interrupts that need to be used are enabled and that the ones that do not want to be used are disable. This will ensure that the module operates as intended and does not consume too much power. On the computer end shall match the baud rate in with the baud rate for the serial COM ports to ensure proper rates do not alias the data received/transmitted. Then using the terminal the computer can be setup to receive data through the Bluetooth module to these ports by accessing the respective com port. Setting up our software so our Bluetooth module does not consume too much power is very important in the design for our system therefore measures need to be taken to turn off power to the Bluetooth module when not in use therefore it is important to use the DCO or a Low power crystal clock when it is suitable and only use the high frequency crystals when absolutely necessary.

5.2.2 Schematic & BOM



AFE4404EVM + GOLD Design

Designator	Quantity	Description	Manufacturer	PartNumber
!PCB	1	Printed Circuit Board	Any	MHR001 & GOLD design
LED5, LED6	2	NIR LED	Thorlabs	LED1550E
D1,D2	2	NIR Photodiode	Marktech Optoelectronics	MTPD1346D-150
ADC1	1	24 bit ADC	Analog Devices	AN7714anz
PB1, PB2, PB3, PB4	4	Momentary Pushbutton	CW Industries	GPTS203211B
LED7	1	Red LED indicator	Lite-On Inc.	LTST- C191KRKT
LED8	1	Blue LED indicator	Lite-On Inc.	LTST-C193TBKT-5A
LED9	1	Green LED indicator	Lite-On Inc.	LTST-C191KGKT
LED10	1	Yellow LED indicator	Lite-On Inc.	LTST-C191KSKT
U7	1	Linear Voltage Regulator	Texas Instruments	TPS7235DBVT
BTM1	1	Bluetooth Module	Texas Instruments	CC2650
LD1	1	LED driver	Diodes Incorporated	AL5802-7
U8	1	Low noise Percision Opamp	Texas Instruments	NE5532P
U2	1	25 MHz Mixed Signal Microcontroller with 128 KB Flash, 8192 B SRAM and 63 GPIOs, -40 to 85 degC, 80-pin QFP (PN), Green (RoHS & no Sb/Br)	Texas Instruments	MSP430F5529IPN
C1	1	CAP, CERM, 2200 pF, 50 V, +/- 10%, X7R, 0402	MuRata	GRM155R71H222KA01D
C2, C3, C4, C6, C8, C9, C12, C14, C15, C24, C32, C40, C42	13	CAP, CERM, 0.1 µF, 16 V, +/- 10%, X7R, 0402	MuRata	GRM155R71C104KA88D
C5, C7	2	CAP, CERM, 1 µF, 16 V, +/- 10%, X5R, 0402	TDK	C1005X5R1C105K050BC
C10, C25, C26, C35, C36, C37, C38, C43, C44, C46, C47, C49, C51, C55	14	CAP, CERM, 10 µF, 16 V, +/- 10%, X5R, 0805	MuRata	GRM21BR61C106KE15L
C11	1	CAP, CERM, 2.2 µF, 10 V, +/- 10%, X5R, 0603	Kemet	C0603C225K8PACTU
C13	1	CAP, CERM, 4.7 µF, 10 V, +/- 10%, X5R, 0402	TDK	C1005X5R1A475K050BC

Figure 26: AFE/GOLD BOM

Our schematics will be simulated in OrCad’s capture using PSpice to model the components. In designing our schematic, there will be libraries for the parts that will be simulated and placed into the schematic. Not all of our parts will be in included in the Capture program. Typically, if a part is need then it will be necessary to create the PSpice model, footprint, and symbol for the circuit design. To design the part we must know the dimensions of the component, footprint, electrical and thermal characteristics, and the pin layout. This information will be found in the data sheet. If we can not find extensive information on the part, or it is simply not needed then we do not have to create a PSpice model for simulation, however creating the footprint and corresponding pins is important for the final PCB design. When design the part we will first have to consider whether or not we want our part to be homogenous or heterogeneous, where homogenous parts to do not have hierarchies in the design symbol. Then next we create the pins, there are different types of pins, and each pin has a number, name and an order number. The different type of pins are power/GND, three state, Bidirectional, open collector, open emitter, input, output, and passive, however regarding the PCB layout there is only power and non-power pins.[28] Once we decide whether our part will be homogenous or heterogeneous we will begin designing the symbol, then next placing the pins. While creating the pins specify the pin type and pin length. A more expedient way to design is to use a template for the part and simulation then modify the existing design to match the component you are looking to create. It is possible to use the PSpice Model Editor program to, “create Capture parts that can be used in schematic designs, circuit simulations, and PCB layouts.” In this method you can either start from scratch or design with preexisting models. The easiest way to get new components is to download the library from the manufacture’s website when it is available. After creating the PSpice model you will create the footprint which is a means for physically attaching components to the PCB. [39] “A footprint symbol is made up of pins(padstacks), graphics, and text. Padstacks define every aspect of how a component’s pins will be fastened to the PCB and how the traces will be connected to them.” [39] Padstacks will define areas for copper pads on inner and outer routing layers, thermal reliefs, openings in soldermasks, and solder paste. [39] The minimum for a footprint requires are the pins, a reference designator, a component outline, and a placement boundary. [39] Once the footprint is made and the symbol is made, the schematic will be created in the Capture program which then will be passed to the PCB editor to create the PCB.

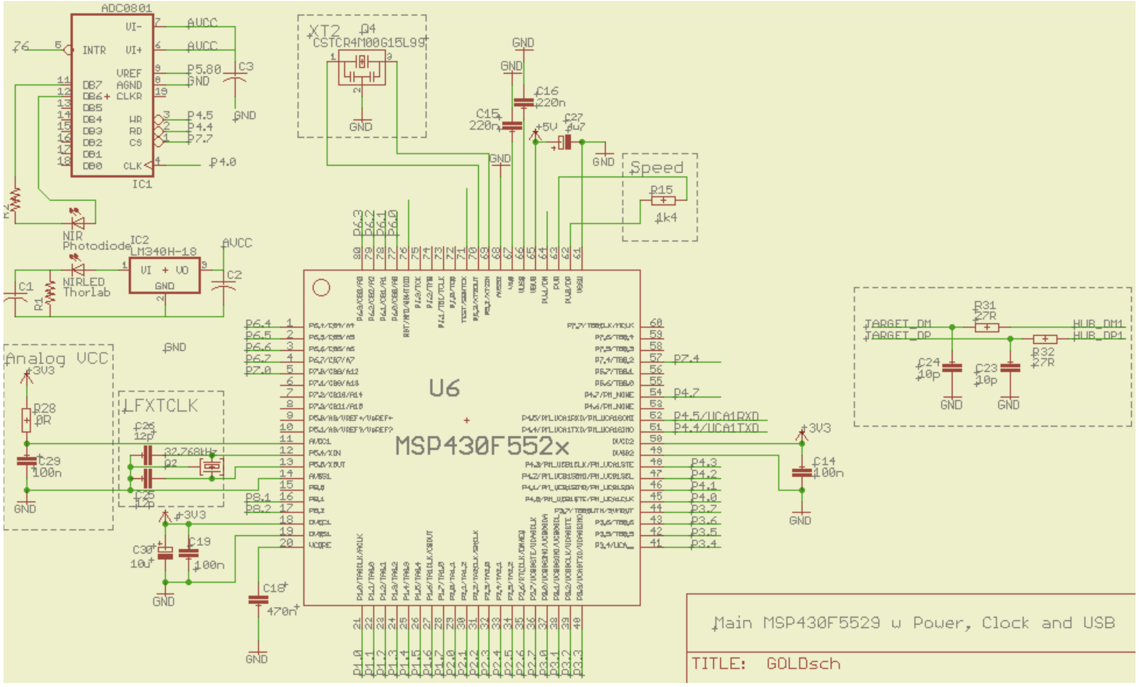


Figure 27: GOLD Project Device Schematic

5.2.3 Performance

The performance of the device can be measured using different metrics. For instance, our performance standard maybe be based on CPU efficiency where the efficiency is based on processing time, amount of modules and registers used, code efficiency, and power consumption. The goal of this project is to be able to accurately measure glucose levels within biological tissues therefore as long as the project can complete this task. The main measure that we are aiming to achieve is that the photodiode's values are of very high resolution and has very low noise, the microcontroller's task is to be able to take those measurements and then pass them back to the server. As long as the MCU does this task efficiently then it will succeed with its performance. Although that task is the goal it will be important to reduce the code design to using only components that it needs, and if it uses multiple registers it does so in an organized and thorough manner. For instance, if it is possible to limit the number of memory map registers then we must do so, and that can make our device be more capable to do more tasks. If the device needs to use more than one I2C communication or UART then the device must switch between the communications of the different tasks in an orderly and complete manner.

The timing characteristics will influence the performance of the system since the tasks will need to be done in order therefore it must be able to do each objective in a desired computing time. IF the device takes too long to process the information then that is an unideal situation and the code must be optimized to reduce implementation time. It is also important for the device to use the least amount of power. It will ideal to switch between low power modes and high consumption modes when the situation does not need to consume power. If the device loses battery too quickly then the device has not performed up to par and must be optimized to reduce power consumption, which can be done by using lower frequency clock sources. Optimizing the capabilities of the external peripherals is also a major talking point since the performance of those devices maybe dependent on how they are configured. For instance, when considering the ADC or Bluetooth module there are many different modes they can be set into. Each respective mode will influence its operating principle. The performance of the System design will be based on how well the device can represent the original signal given from the photodiode back to the server therefore the ADC is very important in reducing noise, filtering unwanted high frequency signals, and removing hum from background noise, and the Bluetooth module is the link between the MCU and the computer.

The Bluetooth module can be optimized to function in different modes and at different baud rates, which can influence to project’s implementation time and/or its ability to reconfigure the system on the fly. It maybe important to return device instructions to the MSP430 since it may improve the outcome. The ADC can be optimized by utilizing the many different features that are include such as the reference signal or the multiple channels include. These different optimizations will be done through coding the microcontroller. The performance of the microcontroller will depend on how we configure the device. Whether our project will call to use the high frequency clock crystal, or a digitally controlled oscillator, using the calendar timer, or using a watchdog timer. The program flow will be very important while utilizing different interrupts for different tasks such as responding to an input peripheral, interrupting the device operation to send data when data is ready, interruptions due to communication flags, and etc. It will be important to try to optimize the device therefore the program flow is continuous and will stay in sync with the tasks that it is performing. Using a JTAG debugger can help greatly in understanding the PC and SR and will coming in handy when trying to get the best performance from our device.

5.3 Schematic

The following Figure below is a spice model of the receiver and transmitter schematic. In short the receiver is a three stage op amp circuit that introduces a photodiode into a low pass filter and finally a gain stage. The transmitter is a one stage circuit that induces a ten milliamp output into the LED.

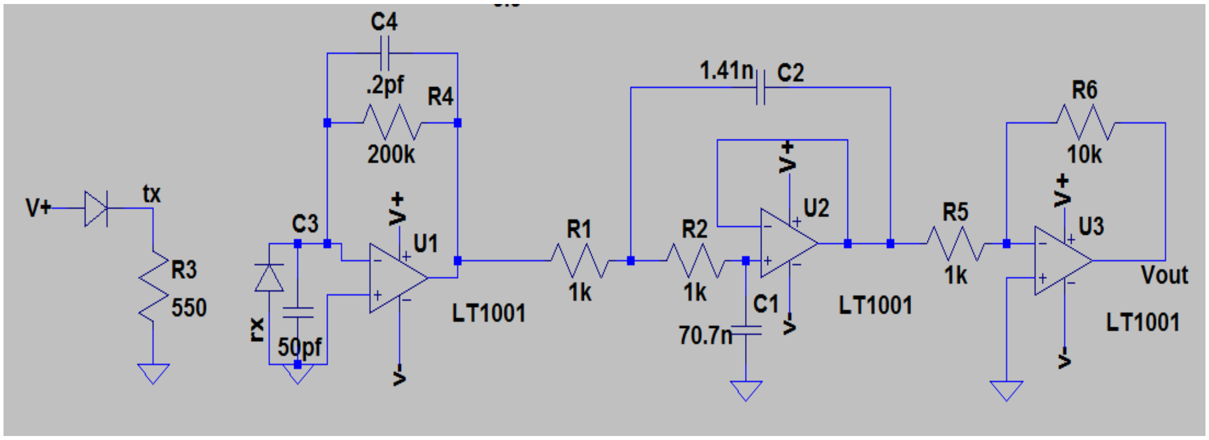


Figure 28: RX TX Transmission

The first stage of the receiver circuit, labeled as rx in this circuit show a photodiode in reverse bias into an inverting terminal of an op amp. The op amp in this specific schematic is the LT1001 which is tentative to change and will be discussed more in depth in a subsequent section. For the sake of the rough draft of the schematic, assuming the input bias current is low enough to not create noise there is a bypass capacitor shunted to ground and a feedback capacitor upon the feedback resistor. The feedback resistor adds more gain the higher the value it is but this draws a problem when the PCB is fabricated and manufactured because this could potentially be a big value and take up space on a PCB board but this problem and solution will be explored later. The upside to a large feedback resistor is that it will reduce the Johnson noise. The Johnson noise is noise generated by thermal agitiaiton of charge carriers. Once the reverse bias photodiode is in place there is a certain amount of noise subject to the op amp. This relates specifically to dark current which is the reverse saturation current of a diode. A proper circuit for a photodiode is a transimpedance amplifier which is another way of saying current to voltage. The biggest calculation for a transimpedance amplifier is NEP rating which stands for noise equivalent power. This rating is the amount of light needed to obtain a SNR ratio of 1. This is a problem when there is no light upon the circuit and you receive a very noisy signal. When the light level is higher than NEP is stabilized. The time constant of the first stage is shown below:

$$T_{max} = R_f \text{ x } C_f$$

Equation ###: Calculation for the Time Constant

For this circuit the response time constant would be .2p x 200k = 40n. These values are going to be manipulated through trial and error and will be polished as the op amps and photodiodes are tweaked. The rise times are how long it takes for a circuit to obtain the specific gain at 40 nanoseconds this is not a relatively high number in terms of analog devices. This exceeds the clock speed so it should not be a concern when our sample speed is slower.

The stage following the transimpedance amplifier is a first order low pass Butterworth filter. This certain low pass filter was chosen because of the maximally flat filter with a minimum rate of change. The bode plot for the specific low pass filter implemented is shown in the Figure below:

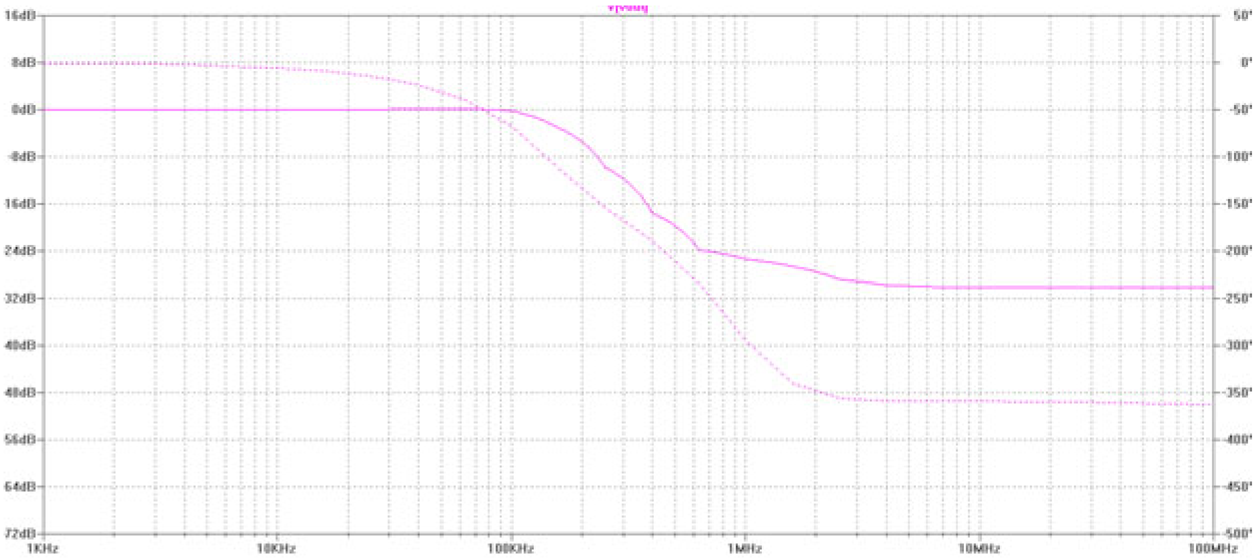


Figure 29: Bode plot of 1st Order Low Pass Butterworth Filter

You can see how the attenuation factor of the signal quickly drops and soon it is at -32dB down which is approximately $10^{-3.2/2}$. The biggest concern of picking the cut off frequency if it is the right frequency to eliminate noise.

The following and final stage is a simple inverting op amp circuit, which turns the negative voltage from the reverse bias first stage circuit into a positive voltage. This gain will be modified from its -10Av accordingly but now it is at a standard value of ten, this of course was picked arbitrarily but will soon be tested and switched if needed. In summation, the receiver stage receives the light that is propagated from the near infrared LED and is passed through a photodetector circuit to minimize the sinusoidal noise. Afterwards, the signal is driven through a low pass filter to again minimize noise and reduce any unneeded signal leading into the gain stage which will boost the signal by ten as of now but that is subject to change. It is also noteworthy that the op amps are going to be powered by the logic level of v+ and v- in the diagram but in real world terms, these devices are actually from the battery of choice that are boosted or inverted into the proper logic levels. For simulations purposes the logic level of 3.3 was chosen but the proper power rails of this device are to be determined in the power section of this paper that will be explored in the appropriate place.

Lastly, the LED circuit or the transmitter circuit seen on the far left of the Figure above. This step is a lot less complex than the receiving stage because of the nature of sending a signal and processing a signal is much more intuitive. In order for the LED to be utilized, the logic level has to be above the turn on voltage.

Afterwards the current will have to be at a specific point to optimize the wavelength transmitted. This certain wavelength relates to the power the LED has. In other words the wavelength may when it is dissipating lower power. This will be manipulated in order to go through a spectrum of the LEDs broadband to create the spectroscopy and relating it to the glucose absorption.

5.3.1 LED Driver

LED drivers are mainly used in order to buck higher voltages from outlets and other things to drive LEDs at lower voltages. Although, this is not an issue for our scope, LED drivers are also used to protect LEDs from changes of voltage and current. At different power levels the light has the proportional power emission which relates directly to the wavelength it propagates. This is a non-issue in the sense that we need to manipulate our wavelengths, but it is an issue when it comes to voltages dropping when it is not intentional. Another advantage to LED drivers is that this rectifies the voltage in order to prolong the lifespan of the device. If an LED is operating at a power that is too high it can damage the duration that that LED will last. This is good when thinking of a long-term build, it is not an immediate solution in terms of a semester project but for an efficient design it is something to keep in mind. The design for an LED driver also has to be able to regulate current and voltage but not interfere with the PWM of the LED shifting through the wavelengths. That is why an LED driver needs to be picked carefully.

5.3.2 Operational Amplifier

In order for the reverse bias photodiode transimpedance amplifier to work effectively, the op amp must have a low input bias current so that the bias current won't be amplified. In other words, ideal op amps have certain characteristics that the offset voltage is zero and so is the current in the inverting and non-inverting terminals but this of course is hard to recreate in the real world. Although, there are certain amplifiers made specifically for these purposes it is impossible as of now to have a completely real world ideal op amp. Certain op amps only have around 60 femtoamps but some device have three magnitude greater than that such as the op amps used in the electronics lab (TL084) or even the LM741 has up to a microvolt. At 1MΩ a 1.5μA input bias current can cause a significant voltage change and will not be able to provide the small resolution difference that is needed. The application of our suggested project requires a very minimal change to be recognized because the absorption of the glucose molecule and water molecule at 1550nm is not an easy task to replicate and any amount of error will be subject to losing valuable data and being able to effectively predict a proper outcome. At this precision the power rails are even up for discussion.

A chart below will be used to demonstrate how the component chosen is different from the lab component used in Electronics 2 lab in the UCF curriculum and how they differ from the LM741 which is one of the pioneer operational amplifiers created decades ago.
These ranges are at supply voltage of ±15V.

Table 12: Op amp Comparison for Design

	LTC1150	TL084	LM741	Units
Input Offset Voltage	± 0.05	3,000	1,000	µV
Input Offset Current	± 10	5	20,000	pA
Input Bias Current	± 5	30	80,000	pA
Slew Rate	3*	13**	.5	V/µs

* at V_i = 15V, R_L = 10kΩ, C_L = 50pF

** at V_i = 10V, R_L = 2kΩ , C_L = 100pF

It is relatively apparent why the LTC 1150 is chosen, but this graph doesn't establish the low noise current that the part has to offer which ultimately is better than just the input bias current being small.

The op amp that was chosen was the LTC1150 Zero-drift operational amplifier with internal capacitors. This device is appropriate not only because it does not require additional components but also because the input bias current for this specific op amp is in the femto region, which is almost a million times smaller than the LM741. The performance characteristics can be shown in the Figure below. As you can see the division is very small and the noise is minimized very efficiently.

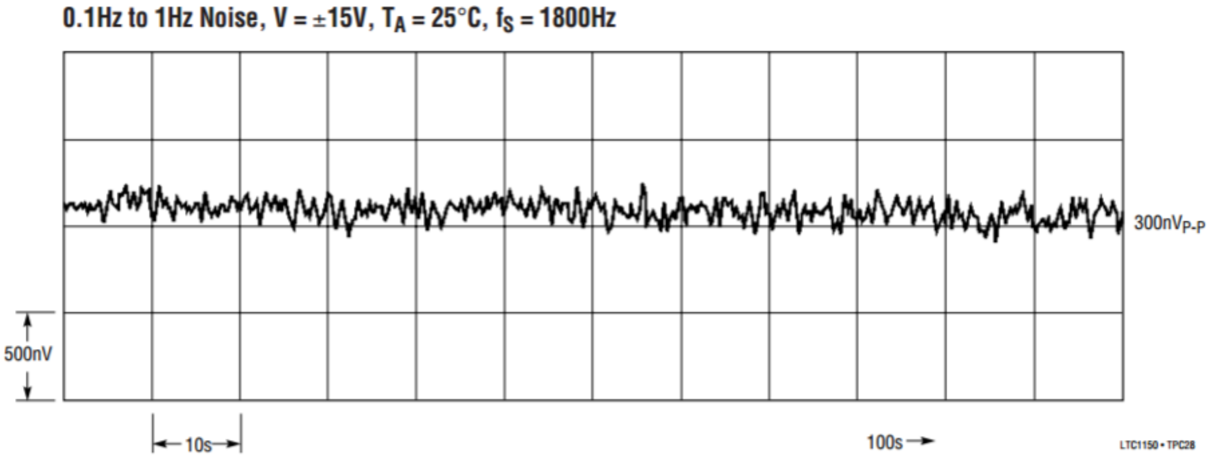


Figure 30: Low Noise LTC1150 Circuit (courtesy of LT)

According to the data sheet of the LTC1150 the power rails increasing have the ability to change the bias current shown in the Figure below:

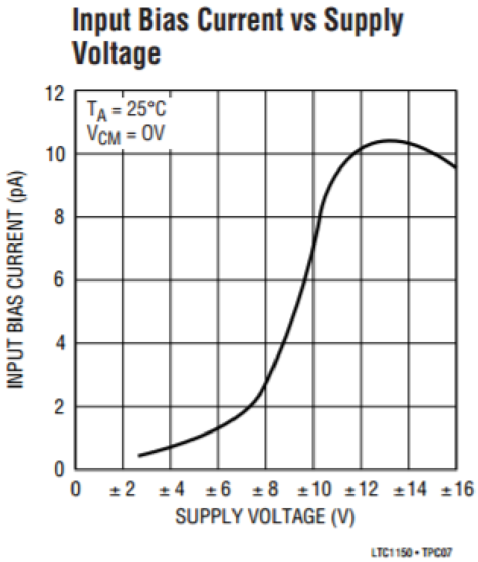


Figure 31: Input Bias current Vs Supply Voltage (courtesy of LT)

As you can see above the power rails or supply voltage can add almost double the input bias current so there are many things to keep in mind. According to the datasheet there are multiple precautions to go through to actually achieving the ideal amperage. For example leakage current can degrade the performance so high quality insulation should be used, they suggest Teflon or Kel-f. Also, surface coating may be necessary for a moisture barrier in high humidity environments so that is something we will absolutely look into. The datasheet also recommends to keep in mind when using wires and picking components that these connections introduce EMF generation. In order to minimize this we must keep in mind when selecting components to consider temperature gradient across the resistors. Also high resistor values have high EMF numbers.

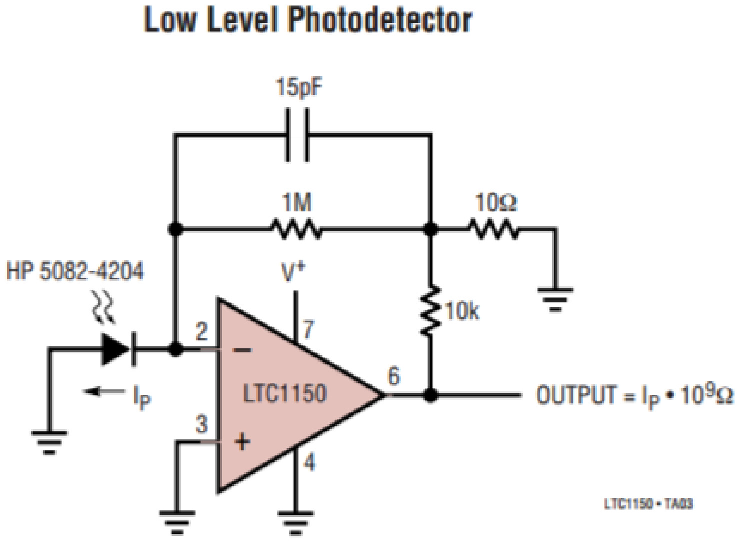


Figure 32: Photodetector Circuit (Courtesy of LT)

You can see in the Figure above that a suggested application for this product is a low-level photodetector circuit. This circuit varies from the one we decided to implement at the 10Ω and 10KΩ resistors at the feedback junction. Although, this circuit shows that this specific op amp was chosen and even designed to use photodiodes as an application.

After the first stage of the receiver side of the schematic there are two additional operational amplifiers being used. This raises a certain question. Does the input bias current influence the other stages as severely as the first stage? The short answer to that question is that no, the input bias current does not matter as much as soon as you pass the first stage. This is because the first stage of the receiver there is a 200MΩ resistor put in feedback, which would amplify a small amount of input bias current a significant amount, but after this stage is done the following op amps do not have to do this. The reason the gain is so high is because the difference in voltage for the photodiode is very small so once it passes through the first stage and the input bias current is minimized that is no longer a concern. Also, the low pass filter and the non-inverting op amp do not amplify the signal to that large of a magnitude. This raises a further question of which operational amplifier should follow the LTC1150. It would be possible to use all three op amps with the LTC1150 but it would be an overkill in terms of price and other things. Designing this project in the scope of a consumer product every penny counts. The LTC1150 goes for $4.55 while a regular LT1001 can go for as little as $1.90 (both of these prices are assuming the quantity purchased is 100+). The LT1008 even has a picoamp input current so it is not as low as a femtoamp it is actually bigger by a magnitude of 1000 it does enough to subdue it. The offset voltage of this amplifier is in the microamps which helps the signal as well.

5.4 User Interface

One of the most important things that comes into play in a mobile application is the design. A typical mobile phone user expects the application designed with the goal of allowing the user to reach, configure and manipulate data in an efficient manner. If the design is unintuitive, or cumbersome to get around the application with, users will quickly lose interest in using the application, and then disappointment follows suit.

Users expect certain things to be there, such as using the back button to return to the last activity, or use the back button enough times to back out of the application itself as a whole. Another thing users expect is a simple menu for easy access to all of the features that the application has to offer. Having users sift through many menus or activities to get to a certain activity or place in the application would be poor design and will cause frustration. In the end, poor application design results in users dropping use of the application, whether the application itself is beneficial to them or not.

5.4.1 Main Application Design

When the user starts the Android application, they will be shown a login screen. The user then can create a new user, or login to an existing one to enter the main screen. The main screen will show them basic info about themselves, and their most recent glucose measurement. When they want to either go to take a new glucose measurement or check on previous logged glucose measurements, they can pull out the navigation drawer by swiping to the left or tapping the menu button at the top left. This navigation bar will also act as the means of traversal throughout the application. This navigation drawer allows the application to avoid the problem of packing too many elements on the screen at once, which can be confusing and overwhelming to the user. Another feature is that when the user is logged in, their name will be displayed in the top of the navigation bar as well. This is done by pulling the username from the backend database. Depicted below in Figure 32 and Figure 33 is the Android application with the navigation drawer pulled out

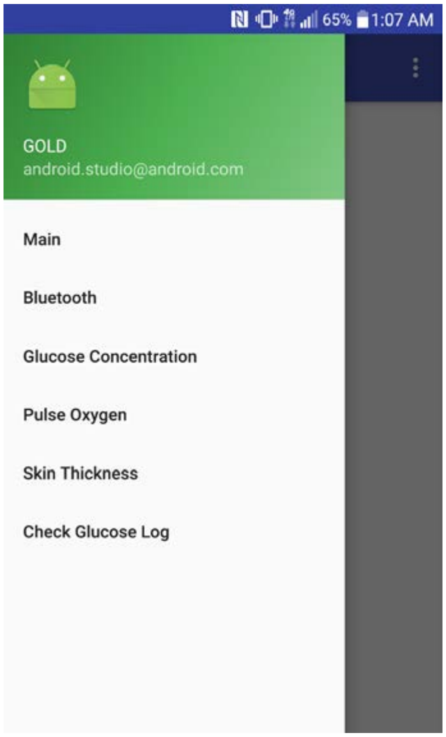
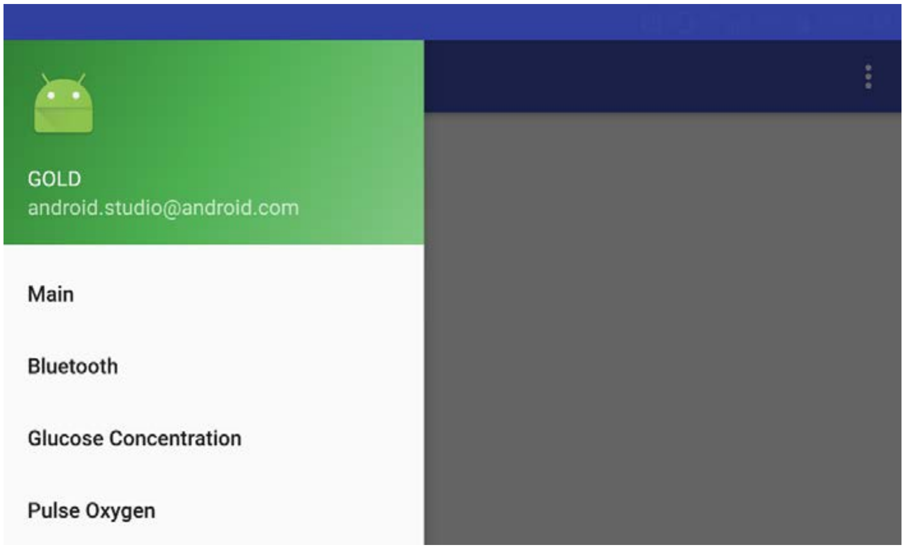


Figure 33: Android Application Interface in Portrait Mode



GOLD

android.studio@android.com

Main

Bluetooth

Glucose Concentration

Pulse Oxygen

Figure 34: Android Application Interface in Landscape Mode

After the navigation drawer is pulled out, the user can click on “Measure Glucose” to be taken to the Bluetooth fragment. In this fragment, there will be a button the user can press which will then activate the Bluetooth adapter, then set it in pairing mode. Then, the user will be able to pair their device with the embedded MSP 430 microcontroller and then have their glucose measurement taken. The user also can click on the “Main” tab to be directed to the starting page that they encountered when they first logged into the GOLD Android application.

In the “Glucose Concentration” tab in the navigation drawer, the user can click on this to be taken to a fragment that will show the level of glucose concentrated in their blood from their previous measurements. In the “Pulse Oxygen” tab in the navigation drawer, the user would be able to see how much oxygen the measurements their blood had when it was used for calculating the amount of glucose they had in their blood. In the “Skin Thickness” tab in the navigation drawer, this will direct the user to a fragment that will show how thick it estimated their skin to be when it was used in calculating the amount of glucose in their blood.

5.4.1.1 Glucose Log

In the “Check Glucose Log” tab in the navigation drawer, the user will be taken to a fragment where their previous recorded glucose measurements are presented to them. This fragment is powered mainly with methods from the DatabaseHelper class. Within the DatabaseHelper class, there is a class to help with the search() method, which is used to populate the table shown in this fragment. This class is called QuerySet, and it has two fields: glucose reading and timestamp. This class is used to easily hold the results returned from the SQL query result set without having to worry about how many rows there are within the result set. This is important because the table that will be shown is created programmatically. This table will have the glucose measurement followed by the date and then the time of the reading. The time of the reading will be displayed in 24-hour format.

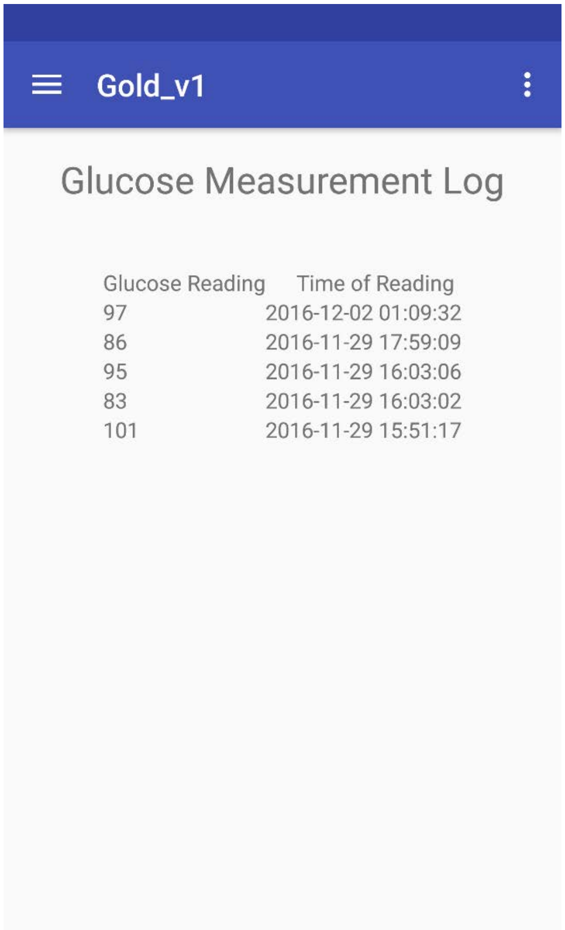
In the GOLD Android application, when the user logs in, the application will take their username and password and use them to query the user's table. It uses the username and password to retrieve the userID associated with that user, and then it uses that information to send a query to the userinfo Table. These values are passed as parameters to the search() method.

In the search() method, the username and userID are used in a SQL query to get information from the glucose_reading and the time_of_reading columns located within the userinfo table. The username and userID columns are used to filter the result set of the query to where only the records that pertain to the user that is currently logged in are returned. This ties back into the security, where the team does not want users to see the biometric information of other individuals without their permission. Another feature that was used for security is the way that the SQL queries were processed. All of the SQL queries are handled with the rawQuery() method. The rawQuery() method allows the GOLD Android program to parse through a SQL query and substitute question marks with the fields the application wishes to use. This helps with the prevention of SQL injections because it strips the incoming information of escaping characters that can be used for malicious purposes.

When the query is complete, the result set is passed into a cursor that is used to parse through the result set. Within this method, a QuerySet ArrayList is created, and through a do-while method, the cursor is used to iterate through the result set, assigning the value from the glucose_reading column into the QuerySet object's glucosereading field, and pass the value from the time_of_reading column into the QuerySet object's timestamp field. Once this is complete, the QuerySet object is then added to the ArrayList, and then this ArrayList is returned to the GlucoseLogFragment fragment.

Within the GlucoseLogFragment fragment, the ArrayList that was returned from the search() method is then checked to make sure that it has existing QuerySet objects within it. If the ArrayList is empty, then the table generation is skipped, and the user is notified that there aren't any logs available to display through a Toast. If the ArrayList is not empty, then it iterates through each QuerySet object until it is complete. It generates the rows dynamically by creating a reference to the existing TableLayout, then referencing a table row layout, generating the TextView for both the glucose reading and the timestamp, then adding the new table row to the existing TableLayout.

This fragment will be very helpful to the user that is trying to get information on how their glucose levels were doing throughout the day. The user may want to use their logged glucose measurement information to give report to their healthcare professional in the instance they have signs of a dangerous total increase or decrease in their glucose levels. The table also will show the user their glucose information for a long time, since it won't only be limited to recent glucose measurement entries. They will be able to scroll up or down to see multiple records if the table gets enough records to fill up the device's screen. While the table may be generated programmatically, the results from the SQL query will be returned with the most recent glucose reading placed at the top, and have the rest placed in order by how recent the measurement was taken. An example of the information displayed in this activity is as shown in Figure 35:



Glucose Measurement Log

Glucose Reading	Time of Reading
97	2016-12-02 01:09:32
86	2016-11-29 17:59:09
95	2016-11-29 16:03:06
83	2016-11-29 16:03:02
101	2016-11-29 15:51:17

Figure 35: Android Application Glucose Measurement Log with Test Values

5.4.2 Mobile Application

The mobile application will be the interface between the microcontroller's raw data and the glucose reading that the user gets. The mobile application will use Bluetooth to communicate with the embedded MSP430 microcontroller, and the data read from the sensors will be processed in the mobile application itself. The user will be able to use the mobile application to measure their glucose, and have those glucose measurements kept for logging purposes. The mobile application will support having multiple users log in, and will keep the information between the two users separate from each other.

5.5.2.1 Android Choice

When the design of the GOLD project was underway, the team had to decide which platform the team was going to use in developing the application. The team had to take our experience with the two platforms, cost of the distribution of the application, cost of the resources that the team would need for development, and end cost to the user when the team made our decision. With the topics discussed earlier in 3.3.4.3 taken into account, the team decided to develop the application using Android as our main platform.

Within our group, all three of the developers had some experience in working with Android before, so it wouldn't take as much effort to begin the project if the team went with Android in the first place instead of using iOS. Another reason tying into our experience with the platforms is the fact that Android applications are created using Java. In the past, iOS applications were programmed using Objective-C as the main language, but now, the main language of choice is Swift, which was released by Apple about two years ago. While Swift is touted as being easier for developers to learn, the team believed that our experience with Java would be more helpful in approaching this task rather than trying to learn a new language from scratch.

The next reason the team decided to go with developing using the Android platform is the cost for distributing the application in their respective app stores. To publish any Android application on the Google Play Store, the developer must pay a one-time fee of $25, and then they can publish as many Android applications as they want without any recurring fees. However, if a developer wants to publish an iOS application on the Apple App Store, the developer must pay a recurring annual fee of $99, and there is more difficulty getting the application up on the store, which can hurt the developer in terms of time and revenue if the application is rejected. With the maintenance of the application in mind, the recurring fee of the Apple App Store was another reason why the team decided to stop considering iOS and moved to develop using the Android platform.

Another reason the team decided to develop the application for the Android platform is that the cost of the resources the team would need for development. Since this is a team effort, all of the developers need to cooperate in contributing to the project. iOS requires the developer to have a computer running macOS, and since the team was using the physical Bluetooth adapter on the device itself, the team must have a phone to deploy and test the program, since Bluetooth communications cannot be emulated using the computer. Unfortunately, not every developer on this team owns a Mac, or has an iPhone available to develop on. With the typical cost of an iPhone exceeding $600 and the typical cost of a Mac exceeding $1000 in mind, procuring the required development tools would be too expensive, and would end up violating our $2000 project budget limit. However, programming on Android does not require the developer to have a certain operating system, since Android Studio can be run on Mac, Windows and Linux. The problem with the need to have physical Android devices available for testing and developing still persists, since the Bluetooth adapter cannot be emulated through a computer. Like stated earlier, Bluetooth communications cannot be emulated using an Android image running on the computer, and a physical Bluetooth adapter is required. However, since most of the developers on this team own an Android device already, this problem is negligible.

Lastly, the reason the team decided to develop the application for the Android platform instead of iOS was because of the cost of Android devices themselves. When developing for iOS, Apple pushes developers to develop for the latest version of iOS so new applications would typically be running on the latest iPhone device, and then older iOS devices that may be running the latest version of the operating system. While Apple does support older devices with updates, eventually, they will be pushed into obsolescence. However, Google does not force you to develop for the latest version of Android, so while it is not recommended, it is possible to distribute to older phones that may have been declared obsolete. In the user's case, when they wish to use this program after purchasing the glucose reader, they could go and find many different Android phones under $200. As mentioned before, the price of a new iPhone will typically exceed $600, and used iOS devices can still hover around $150 with respect to the oldest model supported by the latest version of iOS. While the upfront cost may be a bit high with respect to buying an Android device if the user doesn't own one already, with the need to purchase glucose strips being eliminated, the user ends up saving money in the long run.

5.5.2.2 Bluetooth

The GOLD project's backbone is the communication between the embedded MSP430 microcontroller and the Android device running the application using Bluetooth. One of the major aspects that drove us to choose using Bluetooth over Wi-Fi was that Wi-Fi is not readily available anywhere a person may travel to. Since it is dangerous for a diabetic patient to neglect checking their glucose levels at certain intervals throughout the day, not having an internet connection can be a big problem if this application was used to replace the traditional combination of the glucometer and test strips used in glucose measurement. In the case of using the glucometer and test strips, the only thing limiting the user would be the battery level of the glucometer and electricity, in case it needed to be recharged. Since electricity can be more available than Wi-Fi in certain areas, the user may be inclined to stay with their current solution instead.

When the team takes Bluetooth into consideration, Bluetooth establishes a connection directly with the device it is trying to communicate with, thus eliminating the need to rely on the maintenance of Wi-Fi signals outside of our control. This one-to-one connection between the two devices eliminates the need for a backend, since the team would not have to create, nor maintain servers. Bluetooth also brings the potential of danger of missing glucose measurements down compared to using a Wi-Fi implementation, since the only factor that would prevent the user from measuring their blood sugar would be the level of battery left on the two devices and electricity available to them in their current area.

In the GOLD system project, the communications handled through Bluetooth is done through the UART interface in the embedded MSP430 processor, and on the Android side, there are methods to handle the communications such as socket.getOutputStream(), and socket.getInputStream(). The communications are handled through a bytestream on the Android application side, and then the data must be casted to the respective primitive or object desired. The overview of the Bluetooth implementation is as follows: in the GOLD system project, the embedded MSP 430 microcontroller will receive a value from the Android application, and then it will activate the lasers and photodiodes for measuring the glucose raw information. After a certain amount of time, the microcontroller will then send the raw information back to the Android application through Bluetooth using the UART interface, and then the raw information received will be sent to the respective functions for processing. An overview of the connections and communications between the embedded MSP 430 microcontroller and the Android application are as shown in Figure 35:

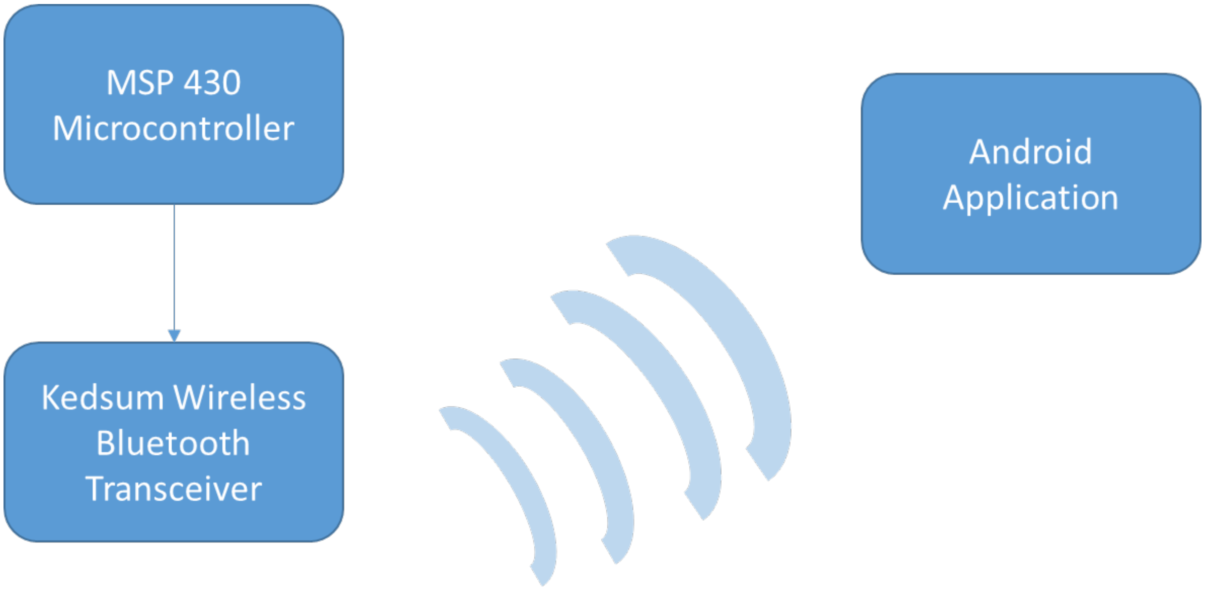


Figure 36: Connectivity between MCU and Android Application

In the Android application, all of the Bluetooth readings are implemented in the Bluetooth fragment. In Android development, a fragment is a portion of a user interface in an Activity. Multiple fragments can be combined to create a multi-pane UI, so not everything has to be placed on one screen [40]. In the Bluetooth fragment, the user can press a button to connect their device with the MSP 430 microcontroller. When the user presses the connect button, there is a check to make sure the device that is running the application has a working Bluetooth adapter available, and if the device does not have one, then the user is made aware of this through a Toast. A toast is a popup that gives feedback on what is going on within an application. If the device does have a working Bluetooth adapter, then the application queries the device for the current paired devices. When the querying is complete, the application gets the address of the paired device that it is looking for then it passes this address information to A BluetoothDevice. The BluetoothDevice is an object that represents the remote Bluetooth device the application is attempting to connect with. After this, the application passes this BluetoothDevice reference into the connectThread object.

The connectThread object will be started as a thread in the application that handles all of the connection between the Android device and the remote MSP 430 microcontroller. When this thread is started, it will use the Android application's universally unique identifier to create an encrypted and secure RFCOMM Bluetooth socket to listen on. Then, if the Bluetooth device was already in discovery mode, there is a method that checks whether the Bluetooth adapter is in discovery mode, then disables it before proceeding with the program. This is done because the process of device discovery is heavy on the Bluetooth adapter, and ends up consuming many resources. This heavy use of resources can significantly reduce the bandwidth available for currently paired devices as well [41]. Now, since the application is sure that the Bluetooth adapter is not in discovery mode, it now attempts to connect with the remote MSP 430 microcontroller through the socket using the connect() method. In this method call, the application will wait for a connection to occur. Usually in Android devices, they are set to be masters in a Bluetooth connection, so this will make the embedded MSP 430 microcontroller have to use a Bluetooth adapter that is a slave type to avoid problems with connectivity. Since this method call blocks, this is the reason that the connectThread object must run as a separate thread in the application, otherwise the main thread that maintains the UI would be blocked, causing the UI to be unresponsive and freeze up. As a result, the user would not be able to do anything else until the connection succeeded or failed, and this would look bad on the application. When this connection succeeds, the socket the application is working with is now passed onto the next object: the connectedThread object.

The connectedThread object will become another thread in the application that will manage all of the communication going on between the MSP 430 microcontroller and the Android devices. When this thread is created, it gets the input and output streams from the socket, and initializes them. The input and output streams are what is used to communicate between the MSP 430 microcontroller and the Android devices. For the input stream, this thread will always listen for new incoming data. If it receives more than zero bytes from the input stream, then it allocates these bytes to the buffer to give the buffer the size of the data in the stream and then these bytes are placed in an object that is sent to a handler. The handler is used to pass the data received from the connectedThread running in the background and update the UI that the user would see. This takes place within the run() method in the connectedThread object, and to manage the data that could be coming in from the MSP 430 microcontroller, there are two ways that have been implemented. The first way is to declare the run() method as a synchronized method. What this does is this makes the multiple thread synchronized. When a thread has a connection with the MSP 430 microcontroller and reads the float byte array through the input stream, no other thread can modify the buffer that is holding the read data until the thread is done working with this information. This is important because if not for this, threads can come in and overwrite the data within the buffer that you are working with, and this can cause incorrect data to be processed and put out to the user. The second step to make sure that the data that the application is working with is correct is to impose a sleep on the thread when there aren't any available bytes to work with. This helps keep multiple threads from jumping at one resource at the same time and effectively stops data corruption as well. In the program, the sleep timer has been set to make the thread sleep for a thousand milliseconds. From testing, this seems like the optimal time between responsiveness and the prevention of data corruption.

While the integrity of the incoming byte array has been verified, there still can be a problem. The MSP 430 microcontroller is a processor that uses little endian by default. The concept of endianness is how bytes are stored in memory. There are two variants of endianness, big endian and little endian. Big endian stores the most significant byte in the smallest address, and little endian stores the least significant byte in the smallest address. The significance of this is that when bytes are transmitted between devices, the respective devices can be using different variants of endian, which can lead to misinterpretation of the information sent and or received. This happened to be the case here. In the Android application, the ByteBuffer's wrap method uses big endian by default because this is what Java uses as default. The difference in endian caused the transmitted value from the MSP 430 microcontroller to come in completely incorrect. After some researching, there was a method called order, which can be used to explicitly set what endian the bytes should be set in. The application uses the ByteOrder.LITTLE_ENDIAN constant provided by that class, and this fixed the incorrect values from displaying when they came in from the MSP 430 microcontroller. For the output stream, there is a write method, where the application passes a byte array of the information into the output stream using the write method. Then this byte array is sent over to the MSP 430 microcontroller for processing. In our project, the GOLD Android application will be sending over a simple character and have the MSP 430 microcontroller check for this character over in its loop method. When it is found, begin the corresponding function.

When the phone is paired with the embedded MSP 430 microcontroller, the Android application will then send a value to the embedded MSP 430 microcontroller, which will then prompt it to activate its sensors and LEDs so glucose values can be measured from the user. When it is done taking a measurement of the data, then it will send this data to the Android application via Bluetooth through UART. When the Android device receives the information, it then takes this information, puts it in a bundle then passes it to the fragment that will process the information.

5.5.2.3 Application Controls

In a mobile application, the controls to use it are important. There must be an emphasis on simplicity, and intuitiveness within the control design. With the GOLD application, the controls are straightforward. Upon opening the application, the user will be shown the login page. They can click on either of the EditText boxes to enter their credentials and press the login button to login, or the user can click on the create user button to be directed to the create user activity to create a new user. On this page, the same follows, where the user can fill out their information to register a new user in the database. When they are done, the GOLD Android application will automatically take the user back to the login page, and they can use that to login.

When the user is within the application, they can navigate through the application by clicking the menu button on the top left. They will be shown the navigation menu, and then they can click on the respective menu tab to be taken to that fragment. Within the “Measure Glucose” tab, they will be taken to the BluetoothFragment, and this is where the user can connect to the MSP 430 microprocessor for glucose level measurements. They will press a button to connect their Android device to the MSP 430 microcontroller, and then they can press the measure button to have the MSP 430 microcontroller perform its measurements and report the results back to the phone. When the user is finished using the application, they can exit the application by pressing back.

5.5 Software Design

In this section and the following subsections, the software used in the GOLD project is discussed. The algorithms used within the MSP430 and the Android application are explained, as well as the intricacies with the implementation of ideas within the GOLD project. The algorithms used in deriving the glucose value will be explained here, as well as the login process, the structure of the backend local database, the encryption on the database, and other features implemented in software.

5.5.1 Microcontroller

The software for the system can be programed in Code Composer, the Integrated Development Environment for MSP chips or in Energia a community made IDE. By configuring the microcontroller to properly utilize many features that are included in the microprocessor the device can handle the data. The software that will be included on the microcontroller will have to be uploaded to the microcontroller before testing occurs. Our design will call for using a USB 2.0 port therefore firmware will be easily accessed and uploaded. The software will include a codes to link different operational modes to LED indicators, a method to sparse a 24-bit ADC down to multiple 16 bit registers, and a way to transfer multiple data sets from the multiple photodiode sensors to a Bluetooth module to send data serially. In order to obtain such a high resolution an external ADC module will be needed. The AD7714 is a nice device that combines signal conditioning capabilities with its high ADC conversion. Using a three wire SPI digital interface the ADC module will need to receive incoming signals from the microprocessor to be configured properly. Other features in our software will include our human interface such as LED indicators, and buttons. The last device feature will be protocol for wireless data through the Bluetooth module. The core of the communication between hardware will be configuring the microcontroller’s GPIO pins. Here is a pseudo code given as an example in the Integrated Development Environment.

```
GPIO_setAsOutputPin(GPIO_PORT_P1, GPIO_PIN0);
GPIO_setOutputLowOnPin(GPIO_PORT_P1, GPIO_PIN0);
```

By properly routing our pins to the hardware, LED drivers, ADC, photodiodes, and other circuitry we can effectively control and test parameters. The LED drivers will need a PWM signal which can be clocked of a number of timers, and the MSP430F5529 uses a Unified Clock System that supports ultra-low power consumption. The UCS module includes up to five clock sources, with some being low frequency and others up to 32MHz. Thus enabling us to use low power mode when the sensor is not in use. There a number of different ways we can implement a PWM driving signal to our LED drivers.

In order to process the data locally a microcontroller is used. This data is eventually sent into an app to do heavy processing but the microcontroller does have a significant amount of information that it handles. The microcontroller or MCU drivers the sensor transmitter LEDs with a pulse width modulated signal in order to do a frequency sweep through the near infrared spectroscopy. Also the MCU has to process all the data output that it receives from the photodiode and through the ADC and then it has to store that data appropriately in order to be sent through Bluetooth into a phone.

The first software aspect that will be looked at is driving the LED into the right wavelength. This is done by changing the power that the signal has. A good way to do that digitally is to a have a pulse width modulator shown below:

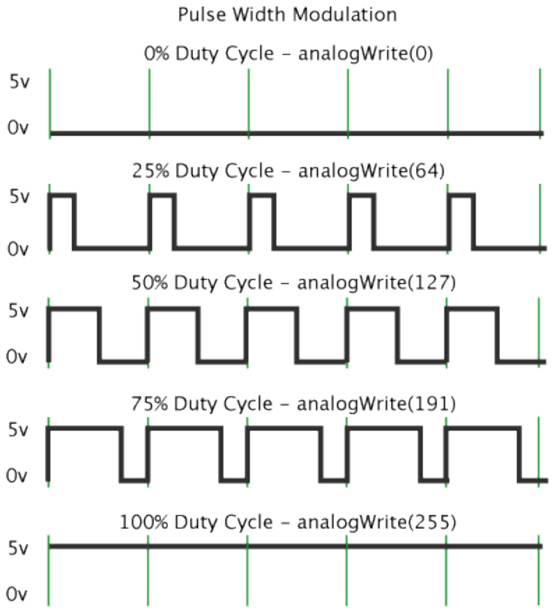


Figure 37: PWM Chart (Courtesy of Ardunio)

You can see above that the pulse changes the logic levels of the signal at a certain amount. A fifty percent duty cycle is a direct square wave and this will have an average of 2.5V if 5V is the logic level high. This could be useful if the LED power is directly related to the wavelength. The right amount of duty cycle varies but it will not be going much lower than sixty percent. Also the spectroscopy is a ramp up or down through wavelengths so the duty cycle will probably increase a tenth of a percent or less every instruction in order to ramp appropriately and then repeat. These transitions will happen more than once in order to stabilize and smooth the input. The photodiode will be sampled at its full bandwidth at least three times and the results will then be saved in a 3x20 matrix, depending on how many wavelengths are needed to be sampled. After this is done the information will then be processed in a phone application to determine what these voltage levels mean. The data on the microcontroller and software is for sampling more so then processing.

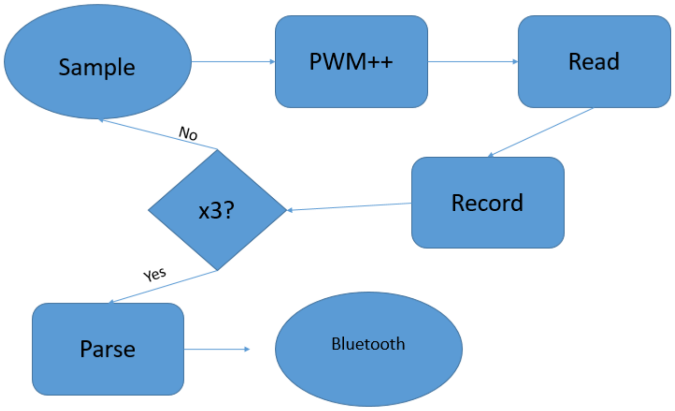


Figure 38: Microcontroller Logic path

This Figure illustrates how the MCU samples and drives the PWM. The frequency is changing with the pulse width modulation in a matrix of certain sample periods will be read and recorded in a data structure algorithm. In order for the data structure to process this information correctly a linked list will be used in order to save memory heaps from getting to big. The benefit of using a linked list is that it points to a different node but unlike an array that mallocs memory continuously which can eat up the small ram that our microprocessor has. This is why the linked list is ideal because the way the memory is used in a link list is separate but the individual nodes have a reference to the following memory block. This is extremely convenient because our MCU does not have a big amount of space so it is vital for the system to have many little blocks of memory as opposed to one big one. This will fix the system from overwriting and corrupting the information. After the software algorithm is recorded there will be a for loop doing this three times. On the third time the condition of the four loop will break and the information will be parsed. This is an issue because the way the Bluetooth was designed is that only one word can be transferred at a time. This is not a big issue because once the information is recorded we can manipulate it in order to send it in the appropriate amount of transferable space. In other words, if you have a 3x30 matrix and each value of the matrix is an integer, you are able to parse each integer digit as its own string and send it digit by digit if needed. This is of course an extreme case of parsing because a single string is less than 8 bytes and that will never be the maximum information you are able to send.

In summation the microcontroller has a simple logic progression of increases the duty cycle in a PWM to drive the LEDs and eventually there will be a certain amount of samples at each respective wavelength. This information will be parsed and afterwards used in an appropriate manner in order to parse the recorded data and send it through a Bluetooth communication protocol.

5.5.1.1 Microcontroller Algorithms

This is where the data that is read from the AFE chip is lightly processed. When the Android device sends a signal to the MSP 430 microprocessor through Bluetooth, there will be a function sendData() that will read information from the port that the AFE chip is connected to, then place that information into a union block. A union is similar to a struct, but there are some differences. Namely, there is more memory allocated to a struct because its memory size is the sum of the total amount of members within it, but a union only has the size of the largest element within it. This is because in a union, all of the members share the same memory space, and because of this, when one member of a union is in use, the rest of the members will have garbage values [42]. This is important because this is crucial to the transfer of the value the application needs.

For the GOLD Android application, it is expecting to see a byte array when it is transferred, however when the program gets our reading from the AFE chip, the application will be given float values. This is where problems occurred in the beginning. In Energia, there are two functions for communicating over the Serial interface: Serial.print() and Serial.write(). There is a difference between the two, but initially, the team decided to go with using Serial.print(). This was done because it seemed that it would be simple to convert the float the application read from the AFE chip into a String, which this function does implicitly because it always sends whatever value it gets, whether it is a character, a string, an int or a float over Serial as a String. The problem occurred when the Android device tried to process the information. On that side, there was a function to convert the incoming bytes into a string, and while it worked, after the first digit of precision, it truncated the rest, or in some cases, it ended up giving the value leading zeroes and values after the actual float itself. This caused us to change how the information was sent over. The next step was to change the function to use the Serial.write() one instead. However, this brought over different problems. While this function was able to send float value information from the MSP 430 microcontroller to the Android device, this caused the Android application to crash with an underflow exception error. After testing was done on the Android application, it was determined that the Android application was not the faulty one, but the code on the MSP 430 microcontroller was. When some more research was done, it was determined that unions were the struct-like type to use to fix this problem.

In Arduino, and essentially Energia, floats are stored as four bytes. This was a problem because when the float value was sent directly, it did not transmit over as a four byte array, which led to the underflow error that occurred when the byte array to float conversion occurred on that side. This is where the union comes in. In the MSP 430 microcontroller program, the union that was used had two members. There was the float member, and then there was an explicit byte array with a byte size of four. Since unions share memory between all of their members, this allowed us to set the float member to a specific float value, and then use the union to read the float data as a byte array by accessing the byte array member after the float member was set. So this was passed into the Serial.write() function with another parameter that explicitly set the length of the buffer to four bytes, and this allowed the MSP 430 microcontroller to send the float value over Bluetooth without having the Android application crash because of an underflow exception error.

5.5.1.2 Android Algorithms

When the MSP 430 microcontroller sends over the floating point value, this floating point value will not be the exact measurement of the current glucose levels of the user. The values sent will be in the form of voltage values produced from the photodiodes. When these values are sent over to the GOLD Android application, they will be stored in a 2x2 multidimensional array. This array will then be processed further to see its correlation with glucose concentrations, and for this to be done, there are a couple of things that need to be done first. The preliminary step is to import the Efficient Java Matrix Library. The Efficient Java Matrix Library is a linear algebra library used in processing multidimensional arrays, which will act as matrices in calculations. This library was used because of its computational and memory efficiency in working on processing small and large multidimensional arrays. After the groundwork has been set, there are a number of various statistical methods that are used in helping determine the glucose levels that the user has based on what the AFE4400 has calculated and sent over. The algorithms used in determining the user's glucose level are: polynomial fit, Beer-Lambert, and principal component analysis.

One of the algorithms in determining the user's glucose measurement is polynomial curve fitting. This is the process of creating a mathematical function to have the best fit to a set of data points. This process may involve smoothing, where a smoothing function is created that fits the data approximately. This is used to reduce the amount of noise that may occur during the generation of the mathematical function. Another process that is used in helping generate the best fit line is the use of fitted curves. These may be used to aid in visualizing the overall data and this also can be used for data prediction, where this can be used to infer values that may come up in the future when there is no more data available to use. This also can be used to help encapsulate the relationships between the data of two or more variables in use.

5.5.1.2.1 Polynomial Curve Fitting

In the GOLD Android application, the polynomial curve fitting algorithm is used with three constraints. These constraints are required because they are needed to help with the prediction of glucose. The constraints are: pulse oximetry, skin thickness, and the heart rate. As a result of these constraints, the polynomial that would be required is a second degree polynomial with the form y = ax2 + bx + c will be used to approximate and then fit these three data points to help with the overall glucose prediction algorithm.

Within the GOLD Android application, there is a class called PolynomialFit where the polynomial curve fitting algorithm will be implemented. The implementation is done by using a least squares solver where the parameters used within the least squares solver can be passed in, making it adjustable. Giving the least squares solver the ability to be dynamically run allows the coefficients within the polynomial to be recomputed on the fly based on the data given. This data will be passed in and performing the least squares solving this way is much less expensive than doing it with a non-adjustable least squares solver that has to resolve the whole system from scratch.

Going into the PolynomialFit class, the Efficient Java Matrix Library's DenseMatrix64F is called to create a Vandermonde matrix, a matrix to hold all of the computed polynomial coefficients that best fit the data given, and an observation matrix. The Vandermonde matrix is used to help with the evaluation of the polynomial given a set of points. The observation matrix is used to hold the initial points that will be used in the polynomial fitting later on. In the class's constructor, the PolynomialFit constructor takes an integer that will be used for the degree. This degree will be used to determine how much the polynomial will be fitted to the given observations available. This degree is then used to create new DenseMatrix64F objects that will use this degree. The coefficient matrix and the Vandermode matrix will use the degree + 1 in their calculations, while the observation matrix will just use the degree given. Next, the solver will be set to use an adjustable variant of the LinearSolverFactory from the Efficient Java Matrix Library, and this adjustable setting allows the solver to easily add elements and remove elements efficiently.

The PolynomialFit class also has other methods that are needed in the main glucose measurement class. These other methods are the getCoef() method, which returns the polynomial coefficients that best fit the data that was given, the fit() method, the removeWorstFit() method, and the removeObservation() method. In the fit() method, this is where the polynomial coefficients are computed using the provided observations to generate the best fit set. This method takes a double array of sample points, and a double array of observations when it is called. It then creates a copy of the observations to place in the observation matrix, then the Vandermode matrix is reshaped using the number of rows from the observation matrix and the number of rows from the coefficient matrix. When this occurs, the initial values in the Vandermode matrix are cleared out and this makes it ready to accept new values for the polynomial fit. The reshaping of the matrix allows the method to be resourceful with the memory it has. After the Vandermode matrix is ready to accept new values, it is filled with values from the sample points parameter that has been multiplied by a factor based on how many rows exist in the Vandermode matrix. When the setup of ready values is complete, then the Vandermode matrix is passed to the LinearSolverFactory for processing. If the processing succeeds, then the solved coefficients are stored in the coefficient matrix.

The next method that used is the removeWorstFit() method. This method is used to remove the observation that fits the model the worst and then reprocesses the coefficients. The adjustable LinearSolverFactory is used her to make the process more efficient. This is done because within a set of given data, there can be a chance for outliers to occur, and since precision and accuracy is a goal with the GOLD project, this helps eliminate these outliers and allows the polynomial curve fitting to be more accurate. This method works by starting off with a worstIndex integer and worstError double used for locating the error. The method uses a for loop to iterate through the observation matrix’s rows then have a double to keep track of predicted observations. Within this for loop, there is another for loop to iterate through the coefficient matrix’s rows, and then the predicted observation is set to the product of the value held at the Vandermode matrix and the coefficient matrix. This predicted observation is used to generate an error double when it is subtracted from what the current observation matrix and then the absolute value of this result is taken. Following this, the computed error is compared to the current worstError. If it is greater, then the worstError is set to the value of the error, and the worstIndex is set to the index of where this error occurred within the observation matrix. After the for loop finishes, it checks to see if the worstIndex has been changed from its initial value. If it has not, then the method will return and no observations will be removed. However, if it does not equal the initial value, then the observation at the worstIndex is removed using the removeObservation() method. After this, the Vandermode matrix is updated by removing that index and then the coefficients are processed again. In the removeObservation() method, it takes the index of the element that is to be removed from the observation matrix as a parameter. When the method is started, it creates an integer that holds one less that the current total amount of rows in the observation matrix and then the data held within the observation matrix is copied over to an array for processing. The method to remove the observation is to use a for loop to shift all of the values up one from the index that was passed in. When the for loop is completed, the total amount of rows in the observation matrix is lowered by one.

5.5.1.2.2 Beer-Lambert

The next major algorithm used in the GOLD Android application is the Beer-Lambert law. The Beer-Lambert law is used to relate the attenuation of light to the properties of the substance that the light is passing through. How the Beer-Lambert law works is when a light passes through a spectrometer, the intensity of this light is measured. The wavelength of the light of this light is also measured and compared to the light before it passed through the material that you are measuring it through. If the intensity of the light that passed through the substance is lower than the intensity of the light originally, then the substance has absorbed some of the light. This difference is used in determining the absorbance of the substance and this is used in further calculations within this law. This absorbance relies on two assumptions: That this absorbance is directly proportional to how concentrated the substance that was measured earlier is, and that this absorbance is also directly proportional to the length of the light path [43].

With the basics of the Beer-Lambert law explained, this brings us to the reason why this law is implemented in the GOLD Android application as an algorithm. The Beer-Lambert law algorithm requires information on how concentrated the glucose level is in the blood where the glucose measurement took place, and this is required because it lets the algorithm know how concentrated the glucose level is and with this, it can get the absorbance. This is also important because the light needs to be emitted at a certain wavelength for the glucose molecules to absorb it. The next thing that the Beer-Lambert algorithm needs to know is the skin thickness of where the measurement took place. As mentioned earlier, the absorbance is directly proportional to the length of the light path, and with the photodiodes used to measure the amount of light that passes through the skin, then blood, the skin thickness acts as the light path that will be used in the algorithm.

Going into the Beer-Lambert class, when the class is instantiated, the class asks for a wavelength integer, a concentration double, and a path length double. After the class has been instantiated, there are a number of methods available to help calculate the concentration, absorption, coefficient, and then the compute() method, which is used to apply the formula of the law to the multidimensional matrix that is passed in. To start off with, the getC() method is used to calculate the concentration, and this works by having the wavelength passed in, then this calculation is applied:

$$-1 * \log_{10} \frac{\frac{Intensity\ of\ light\ after\ passthrough}{Original\ intensity\ of\ light}}{\frac{wavelength\ of\ light}{coefficient}}.$$

Equation ###: Calculation for the Glucose Concentration using NIR LED Method

This calculates the concentration of the substance after the light has been passed through. The next method is the getAbsorption() method, and this method takes an input of a two dimensional array. This function then goes iterates through the matrix using a for loop, then for every element, it calculates the absorption using this formula:

$$-1 * \log_{10} \frac{\frac{Intensity\ of\ light\ after\ passthrough}{Original\ intensity\ of\ light}}{\frac{light\ path\ length}{concentration\ of\ substance}}.$$

Equation ###: Calculation for Absorption of Light using NIR LED Method

After the calculation is done, it then stores the data, and returns it. The getCoeff() method is used to calculate the coefficients that will be used in the Beer-Lambert algorithm, and it calls the getC() method, then divides it by the data. Lastly, there is the compute() method, which applies all of the calculations done up to now, and applies it to the input two dimensional array that is passed in.

5.5.1.2.3 Principal Component Analysis

Within the GOLD Android application, the next algorithm implemented is the principal component analysis algorithm. Principal component analysis is a statistical procedure, which uses orthogonal transformations to convert a set of observations of potentially correlated variables into a new set of linearly uncorrelated variables which are called potentially components. The amount of generated principal components is less than or equivalent to the amount of variables that were used in the analysis. When the variables are generated, it is done so the first principal component has the largest variance. This means that this principal component accounts for as much variability in the given data set as possible. The following principal components also account for as much variability in the given data set, but it must not account for the variability that was accounted for in the previous principal components. The result of this is that the final principal component set are uncorrelated, and now is a statistically independent basis set. With the principal component analysis complete, this can be used in making predictive models.

Within the GOLD Android application, this class also uses the Efficient Java Matrix Library. The variables that this class creates are a DenseMatrix64F principal component subspace, an integer to show how many principal components are used, another DenseMatrix64F to hold the data that is computed, an integer to keep track of all of the principal components used, and a double array to hold the mean values of each element across all of the samples. In this class, there are methods to setup the data structure, add a new sample to the existing data set, compute a new principal component get a new vector from an existing principal set, and determine whether a sample is a member of a set, given that it's error threshold is not surpassed. Starting off with the setup() method, it requires an integer representing the number of samples, and an integer representing the number of elements within each sample. The sample size is used for the size of the mean double array, and then the Vandermode matrix is reshaped with the given number of elements per sample, and the sample size. After this, the sampleIndex is set to zero, and the number of components is set to negative one. In the addSample() method, this takes a double array representing the sample data, and with this, it sets data within the Vandermode matrix. The next method, computeBasis() is used to compute the principal components from the most dominant eigenvectors.

5.5.2 User Control and Security

In this application, it will be potentially handling the biometric information of multiple users, and because of this, this biometric information needs to be kept securely. With this in mind, the developers went to create a user login system, and have a local database encrypted with the user information and glucose biometric information kept secure.

5.5.2.1 Login Process

When the user first opens the application, they will be shown a login activity. They will be able to enter their name and a password to login, or create a local account on the device. When the user clicks on the log in button, the on click listener fires, pulling the information from the two edit text fields, and putting them into strings. The application then uses the name and the password given and then calls the DatabaseHelper object, passing the user as a string into the searchPass method. In this method, the application initializes the local database in a read-only mode, and the application runs a SQL query against the users table asking for the users and their passwords. This query is passed into a cursor that is used to parse through the result set returned from the SQL query. The application then sets the cursor to begin at the start of the result set and then there is a do-while loop where the application grabs the username and checks if it is equal to the username that was passed in. If it is not, then it moves to get the next username. In the case that the username is equivalent, then the application moves the cursor to get the password, and then returns the password that was retrieved. In the case that the user asked for did not exist in the database, then a random password is returned. When this is done, the database connection is closed.

Back in the main activity, this returned password is then compared with the password the user has just given. If the passwords match, then the intent to move to a different activity is created, and then the username and ID are passed along with it. If the user tries to login and they have not created an account or put in the incorrect password, they will be notified through a toast that tells them that their credentials are incorrect. To facilitate a session based login, this works by making the application keep track of the credentials used to login as long as the user does not back out of the application. When the user backs out of the GOLD Android application, the current session will be destroyed, and they will have to log back in to the GOLD Android application once again. The design of the log in activity is shown in Figure 39:

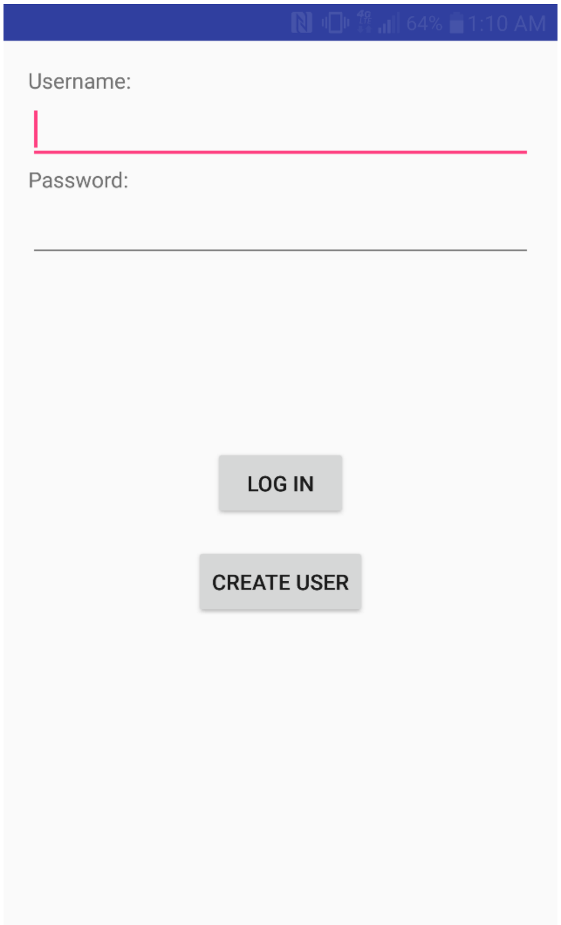


Figure 39: Login Activity

If the user wishes to create a new user account, they can click the create user button, and this will take them into the sign up activity. When the user does this, this will clear the login page of any information so when they come back to it, it looks like how it was when they opened the application in the first place. In the sign up activity, the user can enter a name, and a password. The user also has to confirm the password to make sure they are entering the correct password. When the user clicks create, this fires off the on click listener placed on the button. This then checks if the passwords are different, and if so, it lets the user know that the passwords do not match. If they do match, then the user object is created, and then the name and the password are passed into this object. After this, the user object is passed into the DatabaseHelper class's insert user method. In this method, the application initializes the local database in a write mode.

Next, the application creates a new content values object, where the ID of the user, the name of the user and the password of the user will be placed. To generate the ID of the user, a select * statement is run against the users table, and the count of the amount of rows is used to generate the ID for the user. For the name and the password of the user, these are pulled from the passed in user object mentioned earlier. After these are placed in the content values object, it is then inserted into the users table, and after this, the database connection is closed. When this is complete, the user will be notified, and then they will be taken back to the main activity. Now, they can make an attempt at logging into the application. The design of the create user activity is shown in Figure 40:

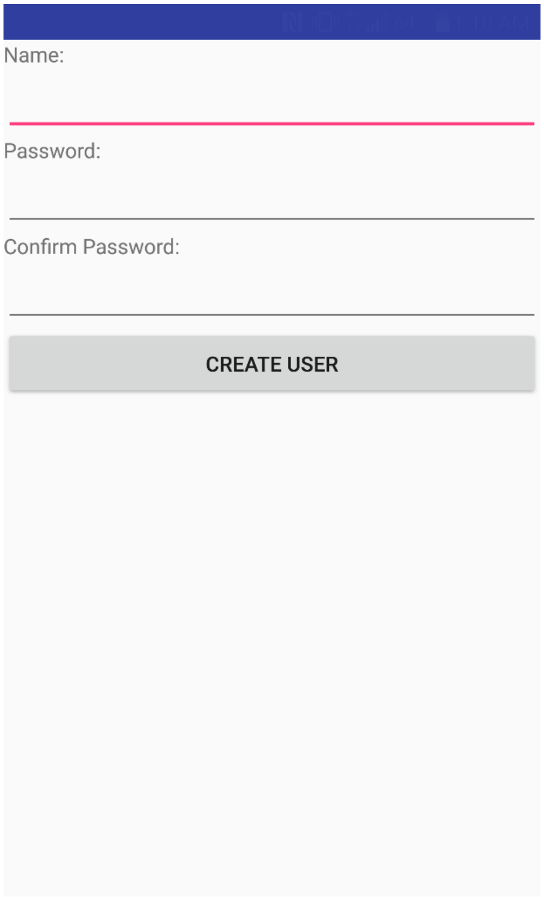


Figure 40: Create User Activity

5.5.2.2 Security and Database Design

Since the GOLD Android application will be handling and storing a user's biometric information such as their glucose measurements. This data is stored within a local

SQLite database that resides on the phone itself. To make the data more secure, there are two levels of security that this app supports natively, and then there are external factors of security that were taken into consideration when approaching the security problem.

The first level of security is the login feature. This will allow the user, and potentially other users to use the same application without being able to see the other user's information. The user can login, check their glucose measurement logs, take new measurements, then when they are finished, they can back out of the application, and when they do, the application will prompt for a login information before a user's information is accessed. The login security is fundamental to the application, because not only does it make sure that the user that is looking at the sensitive biometric information is authorized to do so, it allows the backend database to know what user is using the device at the time so it can return the correct information for the current user.

The second level of security is the local backend database and the encryption feature that it has applied to it. Before the encryption is explained, the backend database has two tables that work together. These tables are the users and userinfo tables. In the users table, the information that the user provides during user creation is stored here. The ID is incremented based on previous records that exist in the table. The ID is used to make sure that each record is distinct, and in the case that two users have the same name, they can be differentiated based on this column. Since the ID is used to make the records distinct from each other, this value will act as the primary key for this table. The user and password columns are populated from the information that was provided during the user creation process. There are no foreign keys in this table. The layout of this table with example data is shown in Table 13:

Table 13: Users Table

Users Table		
ID	User	Password
0	Sam	Test
1	Arjun	Glucose
2	Justyn	CompSci
3	Lambert	Algorithm

The other table that is used in the database is the userinfo table. This table is where the information for the glucose readings are stored. In this table, the infoID is used to make each glucose measurement log record distinct. Since the infoID is used to make the records distinct from each other, this value will act as the primary key for this table. The user comes from the users table, and the userID comes from the users table as well. The user and userID are foreign keys from the users table and this is how the two tables are related. The glucose reading is the value that will be returned after processing done by the other algorithms. The time of reading will act as a timestamp for when the measurement was taken, and this will be used for sorting the glucose measurements within other activities. This time of reading column will be populated with the getDateTime() method, where it uses the standard Java functions to get the current date and time according to the current default locale of the Android device. Then it formats this string into the standard datetime format for SQLite, then it returns it to be set. This is used with the glucose measurement log to accurately track when glucose measurements have been taken. The layout of this table with example data is shown in Table 14:

Table 14: User info Table

User Info Table				
infoID	User	userID	glucose_reading	time_of_reading
0	Sam	0	105	2016-10-31 11:59:59
1	Sam	0	101	2016-11-12 12:06:12
2	Arjun	1	102	2016-11-15 10:04:56
3	Arjun	1	104	2016-11-20 10:04:56

While the database is better to have than a flat file to store potentially large amounts of data, and with the need to access specific parts of data easily, it is perfect for our needs. However, by itself, the database is insecure. All of the data that would be placed within its tables would be stored as plaintext. This is bad because if someone other than the user has the ability to access the phone’s internal storage, and locate where the database is stored, they would be able to easily read the data that has been stored within the database with ease. With encryption placed on the database and the tables that reside within it, the information becomes obfuscated, and instead of plaintext, the information comes out as jumbled up text, and is not as easily compromised.

The security extension to the standard SQLite database that gives this local database encryption is SQLCipher. This extension uses the standard SQLite API to insert a callback into the pager system to operate on the database pages before they are modified or read from storage [44]. More features that the SQLCipher extension uses are that it uses a 256-nit AES in CBC mode for encryption, so the level of encryption is pretty strong. Each database page is encrypted and decrypted individually, so the encryption is very thorough and deep-rooted, instead of encrypting at the table level or even the database level. Each database page in SQLCipher has its own random initialization vector. This initialization vector is generated by a cryptographically secure number generator and is stored at the end of the database page. When an initialization vector is generated, and written, they are regenerated on write so that the initialization vector is not reused on subsequent writes on the same database page data [44]. An initialization vector is an arbitrary number that can be used along with a secret key for encryption. These are usually only employed one time in any session. Since each initialization vector is only employed once in a session, this helps with preventing repetition in data encryption, which will make it more secure against brute force attacks such as dictionary attacks [45].

In addition to these features, when SQLCipher writes a page to the database, each write includes a Message Authentication Code of the ciphertext and the initialization vector at the end of the page. The Message Authentication Code is checked when this database page is accessed to be read from the disk. If the Message Authentication Code or the ciphertext are invalid, then SQLCipher would report that a problem has occurred within the database. When SQLCipher is initialized with a passphrase, it uses Password-Based Key Derivation Function 2 from RSA Laboratories' Public-Key Cryptography Standards to derive the key data. Each database is initialized with a unique random salt in the first 16 bytes of the file. This salt is used in deriving the key used to gain access to the database, and because of this, if two databases are created using the same password, their encryption keys will be different. In SQLCipher's initial state, 64,000 iterations are used to derive the keys for database access [44].

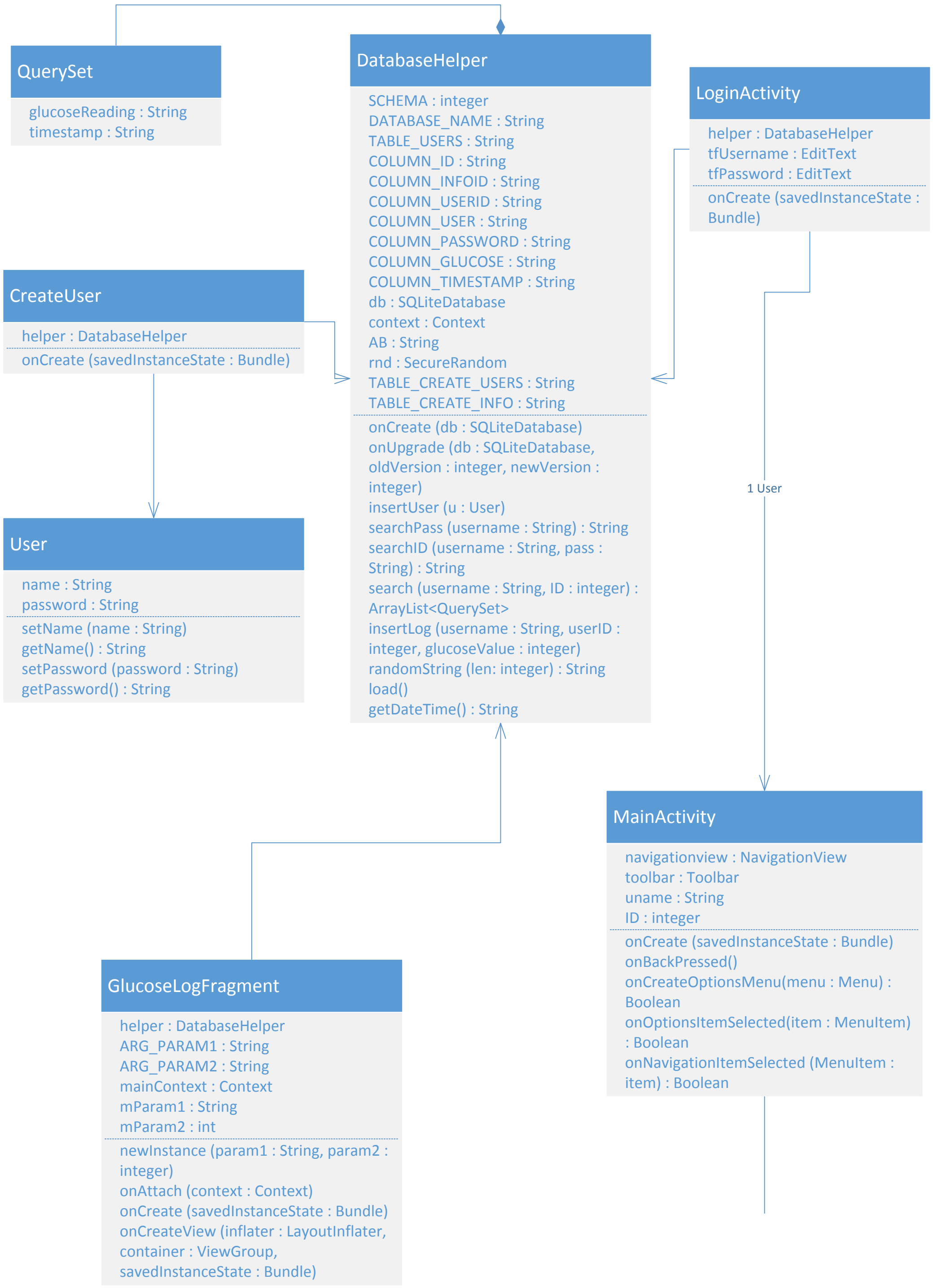
In the GOLD Android application, the SQLCipher database is maintained in the DatabaseHelper class. Since the developers want to use SQLCipher for database encryption, the first thing that has to be done is use the loadLibs() function. This loads native libraries that are required for SQLCipher to run correctly, otherwise it will generate runtime errors such as the java.lang.UnsatisfiedLinkError: No implementation found for void net.sqlcipher.database.SQLiteDatabase.dbopen(java.lang.String,int). This basically means that the database would not be able to be opened since it does not have the required library to do so. Next, depending on whether the application wishes to insert a user into the database, insert a new glucose measurement into the database, or read information from the database, the application has to provide a passphrase to allow the application to either write information to the database, or read information from the database.

Other facets of security and shortcomings are that with the design of this application in mind, the developers designed the security of the database with the assumption that the device running the application could be either used day to day with a passcode or pattern lock, and only while it is unlocked. With that in consideration, SQLCipher helps with obfuscating the data if the information was attempted to be seen by someone other than the user. However, there are some shortcomings in the security of the application. One of them is that since the database is local and on the phone itself, there is a problem on where to store the passphrase for accessing the database. At the moment, it is a static hardcoded phrase within the application, which lowers the security level of the application, but with that in mind, this problem can be subverted if the phone has passcodes or pattern locks enabled, which tend to block access to the file system if the device is not unlocked. The next thing that makes this work is that even though the passphrase is hardcoded, the only way to get to is to reverse engineer the Android apk file and look through the source code to find out what the passphrase is. This would require the phone to be stolen, then the thief would have to know what the passphrase or pattern lock would be to get to the file system then use the necessary tools to reverse engineer this application.

The two alternatives to this were that since the GOLD Android application supports multiple users, have the main user’s password work as the passphrase to unlock the SQLite database, which would work if there were only one user using the GOLD application. Since there can be more than one user, this type of implementation would require that the main user’s password would be stored unencrypted somewhere within the application so secondary users would be able to open the encrypted SQLCipher database for regular reading and writing operations. This would make the application even more insecure and could potentially allow for easier access to the sensitive biometric information that would be stored within the database. The current implementation keeps all of the users’ passwords encrypted, while this would have to expose the main user’s password in plaintext. The other alternative would be to have the database run remotely, and have the phone access and pull the information over the internet, but since the aim of this application is to avoid requiring the internet connection this alternative is not viable.

5.5.3 Class Diagram

The class diagram of the GOLD Android application is shown in Figure 40 below. This contains the complete and working classes in the app as of the writing of this document.


QuerySet
glucoseReading : String
timestamp : String
DatabaseHelper
SCHEMA : integer
DATABASE_NAME : String
TABLE_USERS : String
COLUMN_ID : String
COLUMN_INFOID : String
COLUMN_USERID : String
COLUMN_USER : String
COLUMN_PASSWORD : String
COLUMN_GLUCOSE : String
COLUMN_TIMESTAMP : String
db : SQLiteDatabase
context : Context
AB : String
rnd : SecureRandom
TABLE_CREATE_USERS : String
TABLE_CREATE_INFO : String
onCreate (db : SQLiteDatabase)
onUpgrade (db : SQLiteDatabase, oldVersion : integer, newVersion : integer)
insertUser (u : User)
searchPass (username : String) : String
searchID (username : String, pass : String) : String
search (username : String, ID : integer) : ArrayList<QuerySet>
insertLog (username : String, userID : integer, glucoseValue : integer)
randomString (len: integer) : String
load()
getDateTime() : String
LoginActivity
helper : DatabaseHelper
tfUsername : EditText
tfPassword : EditText
onCreate (savedInstanceState : Bundle)
CreateUser
helper : DatabaseHelper
onCreate (savedInstanceState : Bundle)
User
name : String
password : String
setName (name : String)
getName() : String
setPassword (password : String)
getPassword() : String
1 User
MainActivity
navigationview : NavigationView
toolbar : Toolbar
uname : String
ID : integer
onCreate (savedInstanceState : Bundle)
onBackPressed()
onCreateOptionsMenu(menu : Menu) : Boolean
onOptionsItemSelected(item : MenuItem) : Boolean
onNavigationItemSelected (MenuItem : item) : Boolean
GlucoseLogFragment
helper : DatabaseHelper
ARG_PARAM1 : String
ARG_PARAM2 : String
mainContext : Context
mParam1 : String
mParam2 : int
newInstance (param1 : String, param2 : integer)
onAttach (context : Context)
onCreate (savedInstanceState : Bundle)
onCreateView (inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle)

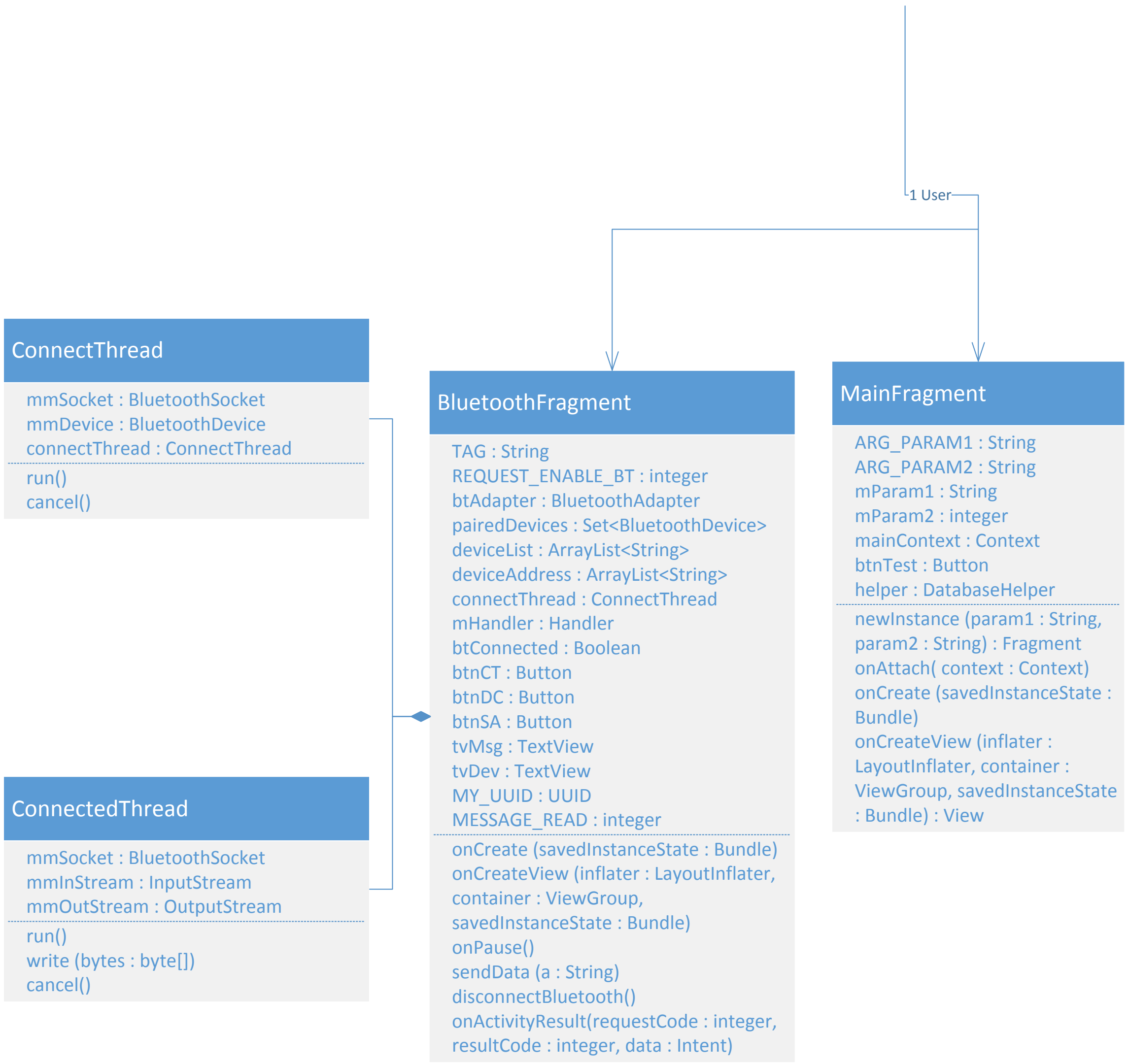

1 User
ConnectThread
mmSocket : BluetoothSocket
mmDevice : BluetoothDevice
connectThread : ConnectThread
run()
cancel()
BluetoothFragment
TAG : String
REQUEST_ENABLE_BT : integer
btAdapter : BluetoothAdapter
pairedDevices : Set<BluetoothDevice>
deviceList : ArrayList<String>
deviceAddress : ArrayList<String>
connectThread : ConnectThread
mHandler : Handler
btConnected : Boolean
btnCT : Button
btnDC : Button
btnSA : Button
tvMsg : TextView
tvDev : TextView
MY_UUID : UUID
MESSAGE_READ : integer
onCreate (savedInstanceState : Bundle)
onCreateView (inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle)
onPause()
sendData (a : String)
disconnectBluetooth()
onActivityResult(requestCode : integer, resultCode : integer, data : Intent)
MainFragment
ARG_PARAM1 : String
ARG_PARAM2 : String
mParam1 : String
mParam2 : integer
mainContext : Context
btnTest : Button
helper : DatabaseHelper
newInstance (param1 : String, param2 : String) : Fragment
onAttach(context : Context)
onCreate (savedInstanceState : Bundle)
onCreateView (inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
ConnectedThread
mmSocket : BluetoothSocket
mmInStream : InputStream
mmOutStream : OutputStream
run()
write (bytes : byte[])
cancel()

Figure 41: GOLD Class Diagram

Starting from the top, the first class block is the LoginActivity. The LoginActivity is the entry point to the program, and with this, it requires an instance of the DatabaseHelper class to perform its login functions. The create user class also requires an instance of DatabaseHelper to work, so the relation between these classes are shown with an association between them. The next part of this diagram is the MainActivity. This activity powers the navigation menu and effectively acts as the central hub of the GOLD application. This is shown through the multiple one-way associations that branch off from the MainActivity. While the user can jump between different fragments using the navigation menu, the individual fragments are not aware of each other. With this, there is not a direct connection between the fragments. Moving over to the BluetoothFragment, this class has two private inner classes, the ConnectThread class and the ConnectedThread class. These two classes are crucial to the BluetoothFragment class's functionality, and because of this, the relationship between these classes and the BluetoothFragment is that of a composition aggregation.

5.6 Mechanical Design

Although this project mainly focuses on the electrical hardware and software design, this equipment needed to be housed in a device for easily detecting the glucose levels in the blood noninvasively. The following section details the mechanical design that holds the near infrared LED and the photodiode and how it is designed.

5.6.1 Earpiece

The earpiece housing was designed to hold the LED and the photodiode on opposite sides so that the photodiode can absorb the remaining light from the LED shining through the ear. Since the best way to obtain blood glucose measurements is through the flesh of the earlobe, this earpiece design was chosen.

In the CAD software program SolidWorks, the earpiece shown in the drawing in the drawing is modeled after a typical Bluetooth headset with the LED and the photodiode placed on the earlobe. Two hole mounts are placed on the earpiece for easy attachment and removal. The earpiece protects and positions the LED and the photodiode to perform its function safely. Without the earpiece, our design and testing would be flawed due to the calibration already implemented. By installing the two electrical components on the earpiece, the LED and the photodiode cannot be moved or misaligned and the performance of the glucose measurement can be maximized by testing the function repeatedly with the same exact parameters such as placement and distance between the two components.

To build the earpiece model for use with our project, we will use the ear shape of a group member and the design will be printed using a 3D printer with a plastic filament. Although the components are held firmly onto the earpiece, they shall be removable from the earpiece so that the LED and the photodiode can be used in different applications such as testing the forearm or testing the glucose in a solution. The earpiece shall be considered an attachment for the glucose measurement device.

In order to hold the LED and the photodiode on the earpiece and have the ability to remove the two components, the earpiece shall allow the two components to be detachable and twist the LED and the photodiode off the earpiece. Similar to the capability of a patch, the two electrical components that help measure the glucose concentration can be removed from the earpiece. Due of this functionality, the earpiece acts as an accessory for the glucose device and the components can be used for a plethora of distinctive applications.

5.6.2 Cable Housing

When the systems electrical components are finalized the design of the cable housing is taken into consideration. This system cable housing is straightforward in the sense that there are not many wires to keep track of and cover. Also, a spare wire every now and then would not be detrimental in regards to functionality. Although, for professionalism it is always important to keep your project presentable. Many designs for cable housing will be compared to determine the most efficient and provocative way to display our models.

First and foremost, there is a traditional and overly simplified design of having a PVC tubing encompass any loose wires. This approach is extremely feasible when it comes to design and planning. At most a slit will have to be cut in one PVC to slide the wires in but that is not a hard task to overcome. This however is an easy solution, but at the same time it looks like an easy solution. In terms of cable housing it is always important to keep a relation between how hard a design is and how hard it looks to make. For example, a computer casing with see through components and tubing that wraps and warps creates a very intricate appearance. The difficult appearance often rewards the end user for having fancy or up to date technology. However, the PVC pipe design does not lend its self to complexity. No one will ever see a PVC tube with a few wires in it and be impressed so this approach will not be used. An alternative to having a simple looking cable system will be to have a metallic rig. In fact, the shinier something is the better result the end user will expect to see. This phenomenon is often explored when it comes to building mobile phones. A lot phones have metallic matting and glossy finishes in order to impress the buyer. This is the exact ideology that is needed when it comes to developing a cable housing. A lot of mobile phone reviewers criticized the new google phone the google pixel because it wasn’t shinny enough even though it sounds ridiculous. Phones such as the new iPhone 7 are actually praised for the matte black casing because of the amount of reflection. This is crucial when deciding how to develop a product for consumers because a lot of buyers are finicky.

When it comes to a product working or not it is not important how it looks but what it does but when selling a product is an end goal then this may be the most important part of the project. The downside to customize a reflective housing is that it is a waste of time printing this design and fabricating it and should never be acknowledged. A third choice for cable housing is an old ephemeral wood housing. A nice thing about wood is that is has an old time, grandfather vibe to it that resonates with different generations. This is a huge bonus in using material that have a rewarding recollection of senses. Wood adds the option of being able to sand down the finish or paint it. It adds a lot of customization in terms of different paints colors and laser printed designs on the side. This customization works when printing a logo or doing something of this sort in terms of designing, but having wood as a finished product is not very economical or light weight. In fact, a plastic casing would even surpass this although that is not the best solution. When analyzing stress and strain for a material the young’s modulus table for stress over strain comes into play and follows the graph until a fracture is formed. However, this chart can easily be found online it is almost excessive when it comes to designing housing for a table. The problem is not a problem on longevity of material or support for weights its more of an aesthetic look.

Before a material is chosen the scope of the design will have to be analyzed. Where will the cables be routing from is the major question at hand. If the sensor has a casing on the year and the Bluetooth module is in the ear casing then it can be wireless and the only wire will be contained in the main PCB hub. This is probably not the case because the Bluetooth module being in the ear cover will be too cumbersome to clamp on someone's ear. The way to have the ear clamp as light weight as possible then leads to a design that contains a minimal amount of electronics housed in the ear clamp. This is a problem when it comes to noise but this will be explored in different sections. For now the cable housing will be subject to discussion. If the ear piece is simply a photodiode to receiving and the information is processed somewhere this will solve the problem of keeping the ear piece light weight and this is where the cable housing problem gets interesting. How will the connection be fed to the main processing hub? The most natural solution to this problem is to have a normal awg cable with a ferrite bead attached at the end. The ferrite beads are at the end of wires to have anti electromagnetic interference and radiofrequency interference. When cables are longer they can receive signals that are floating around so a ferrite bead or a cable choke will be needed to be used. The inside of the cables prevents from interference as well and they are shown in the Figure below.

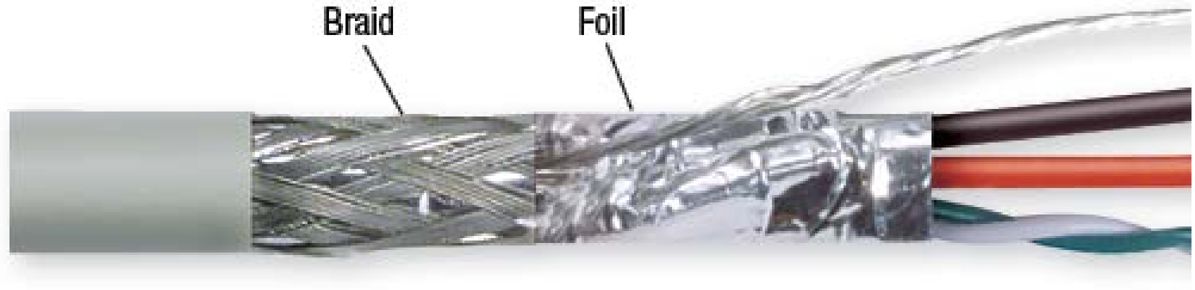


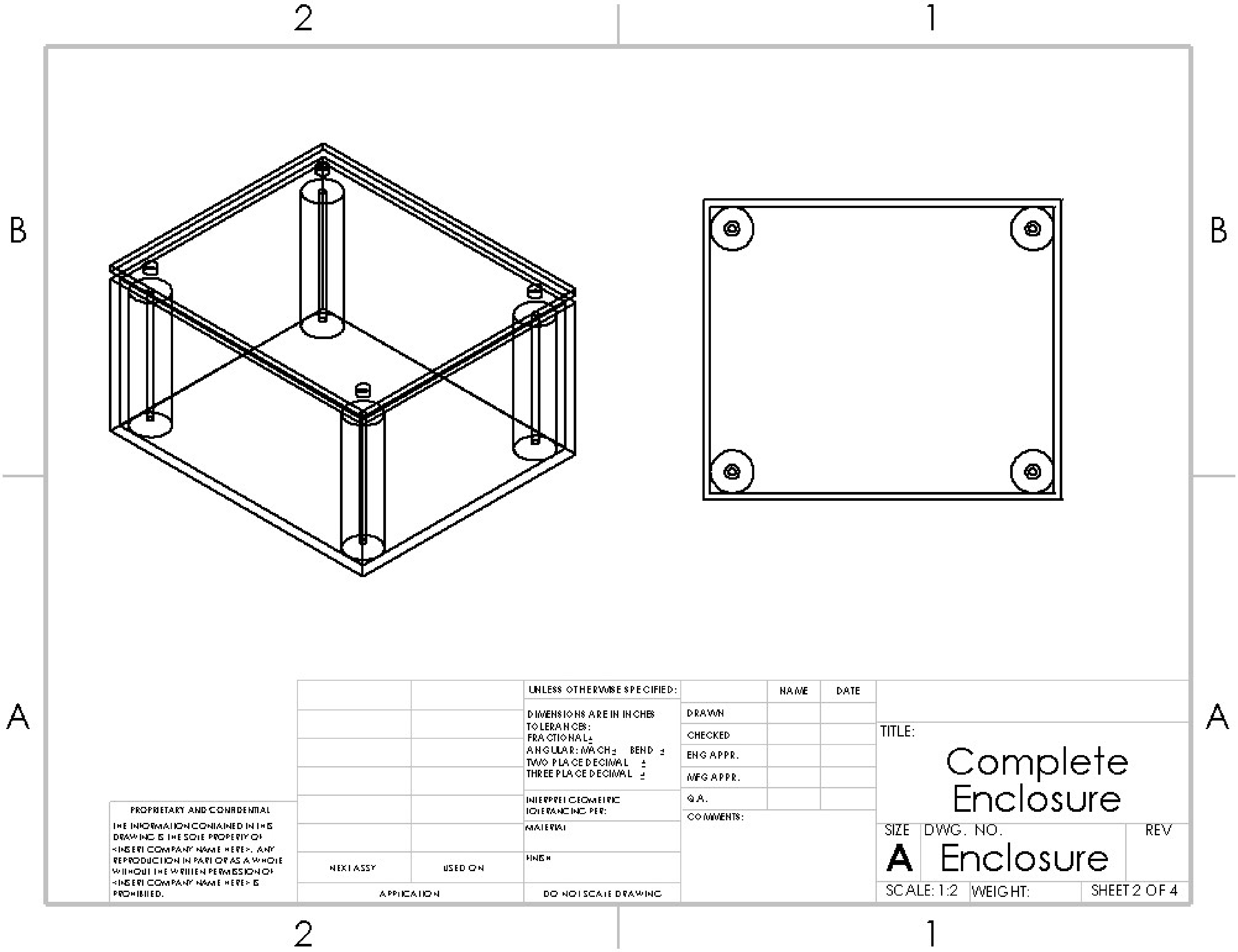
Figure 42: Cable Jacket and Insulation Diagram

This Figure illustrates the material wrapping around a cable a standard USB 2.0 to better show a diagram of how cables are self-maintained. This is an alternative of just have the regular rainbow cables being used freely. This is a design that would work better in a consumer product but for the scope of this design, proof of concept, a simple rainbow cable will suffice. If things were to be designed a little differently than normal, a plastic 3D printed module would be too rigid.

5.6.3 Electrical Housing

The electrical enclosure designed in the drawing on the next page will act as a housing for the electrical boards such as the PCB, the AFE4404 used for pulse oximetry, and the microcontroller among possibly a few more components. This encasing shall protect the electrical equipment from accidental damage or the slight ingress of any liquids or substances that could potentially damage the electrical parts. For ease, the encasing will be made from plastic materials and may either be bought offline for simplicity purposes or built through a 3D printer.

The electrical housing shown in the drawing below is a model of what the entire system will resemble when all the parts are correctly placed in the enclosure. A slot for the on-off position will be located on the enclosure and this slot must be indented and flush with the entire enclosure design to prevent an outside object from turning the system on or off accidentally. Also, another button may be left open for a test button to initiate the glucose measurement process.



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

				NAME	DATE
		UNLESS OTHERWISE SPECIFIED:			
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL± ANGULAR: MACH± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN		
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL	COMMENTS:		
NEXT ASSY	USED ON	FINISH			
APPLICATION		DO NOT SCALE DRAWING			

TITLE:

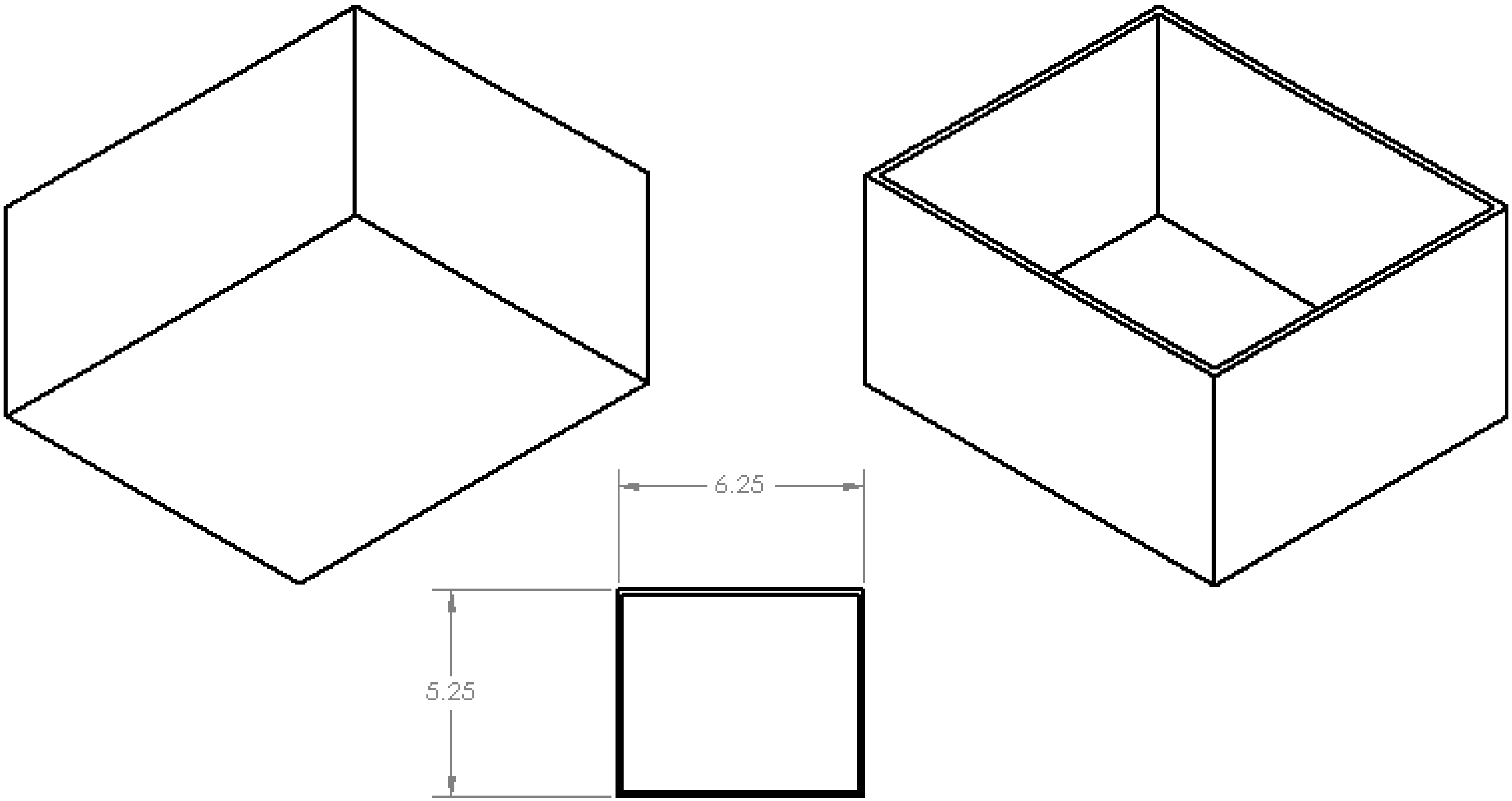
Complete Enclosure

SIZE A

DWG. NO. Enclosure

REV

SCALE: 1:2 WEIGHT: SHEET 2 OF 4



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC
TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

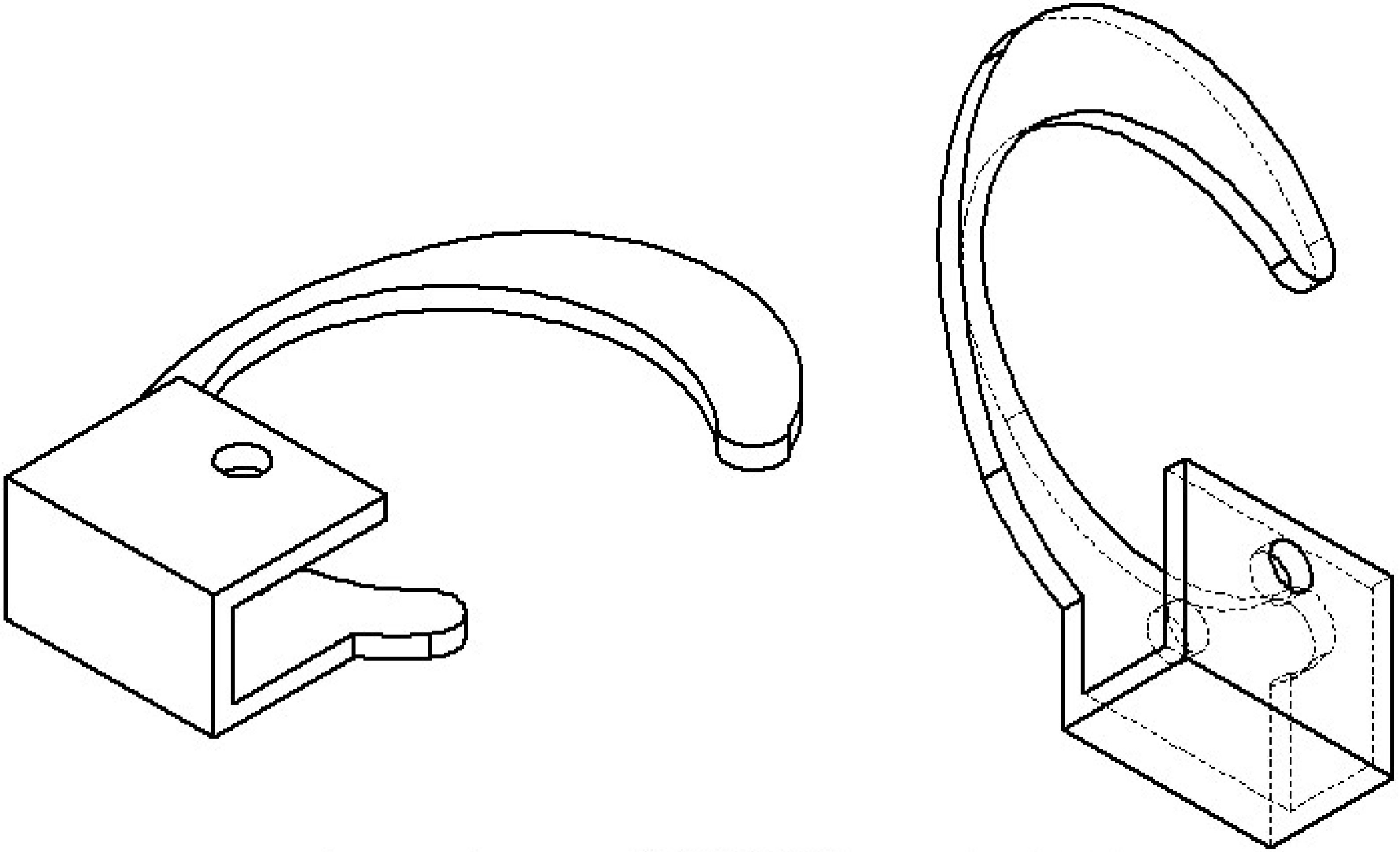
Bottom of Electrical Enclosure

SIZE A

DWG. NO. Enclosure

REV 1

SCALE: 1:2 WEIGHT: SHEET 1 OF 4



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF INSERT COMPANY NAME HERE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF INSERT COMPANY NAME HERE IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC
TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

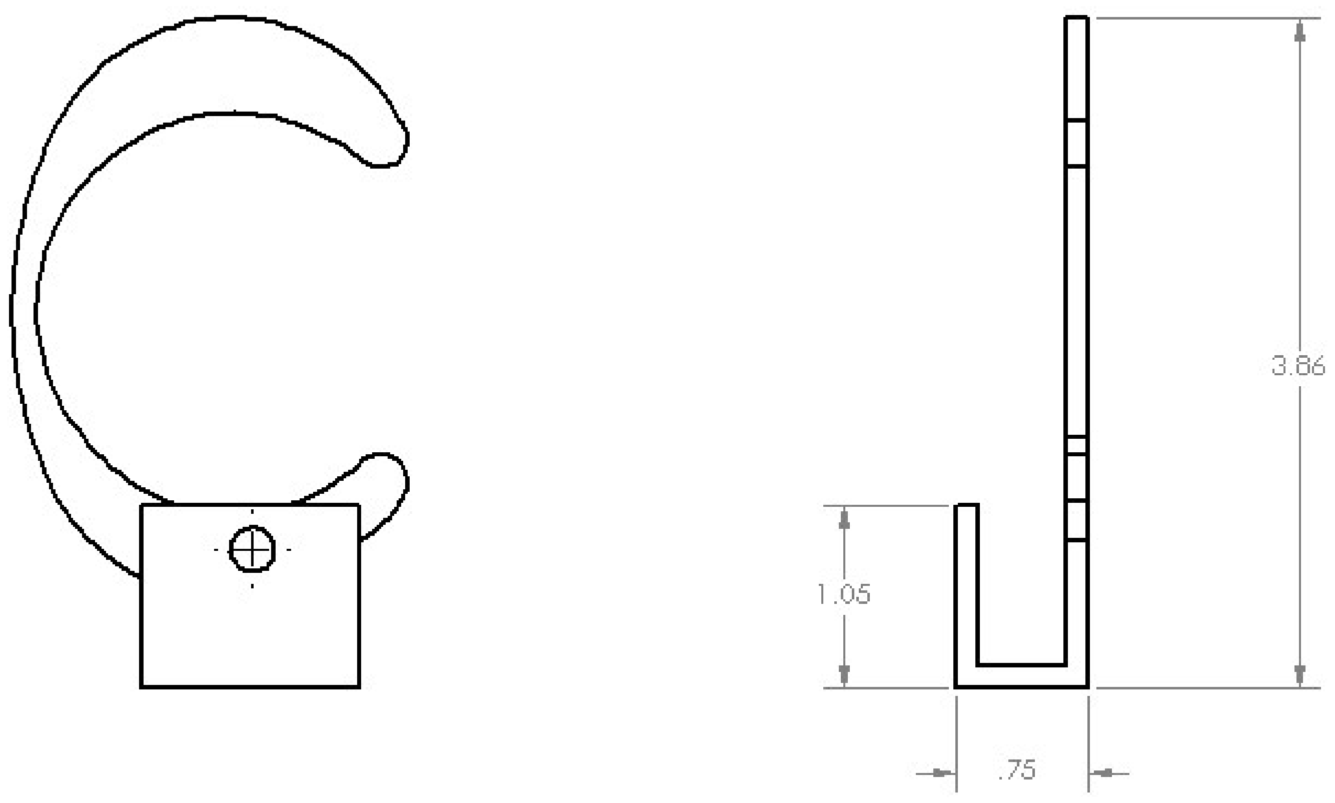
Earpiece Accessory
(cont'd)

SIZE
A

DWG. NO.
Enclosure

REV

SCALE: 1:2 WEIGHT: SHEET 4 OF 4



UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN		
	CHECKED		
	ENG APPR.		
	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL	COMMENTS:		
FINISH			
DO NOT SCALE DRAWING			

NEXT ASSY	USED ON
APPLICATION	

TITLE:

Earpiece Accessory

SIZE	DWG. NO.	REV
A	Earpiece	
SCALE: 1:2	WEIGHT:	SHEET 3 OF 4

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

5.7 AFE4404 Evaluation Module

This section will discuss how the evaluation module that Texas Instruments provided us with will be integrated with our PCB design. Many ideas were sought out for this integration and they will also be discussed. The reasons as to why idea was chosen over the other will also be discussed. The final integration will use the jumper holes in the AFE evaluation module to connect to the MCU pins.

5.7.1 AFE4404 Integration

The AFE4404 is integrated in an evaluation module that Texas Instruments donated to us. The schematic is composed of six pages, which shows how the evaluation module is constructed. The first includes a diagram of the AFE4404 chip connections. The second sheet includes a diagram with the pin layout for the microcontroller used in the evaluation module. This is the diagram that is most important to us when figuring out how the evaluation module will be integrated. The evaluation module contains six open connectors that are connected to the microcontroller. These six connectors will be used to connect our PCB to the MCU. Figure 42 shows how the connectors look on the PCB and the name of their pin connection.

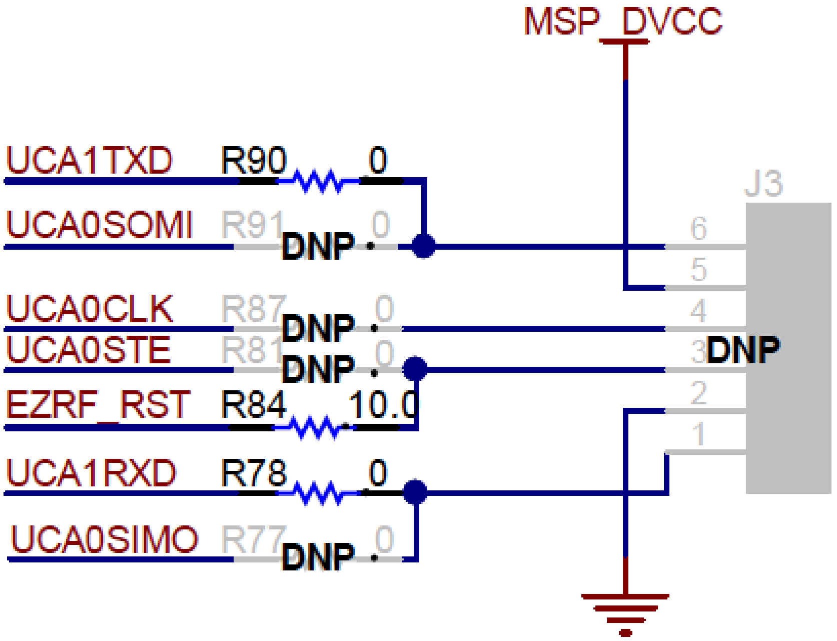


Figure 43: Jumper Connectors to Evaluation Module

The next important factor is to check where the pins connect to on the MCU. The idea is to have our PCB with one transmitter and receiver input/output. The MCU diagram is shown in Figure 43. As seen in the schematic many of the pins are not being used and can be helpful to use for our PCB integration. The problem with the unused pins is how difficult it becomes to successfully connect to them. That is why the jumper (J3) as shown in Figure 42 is so beneficial. Since J3 are just open connector holes in the evaluation module then it allows for easier connection to the MCU. A different way to connect to the MCU would be by using jumper cables and soldering them on to the MCU pins that are not in use. Table 15 will describe the significance of each pin on J3 and what their function is in relation to the MCU.

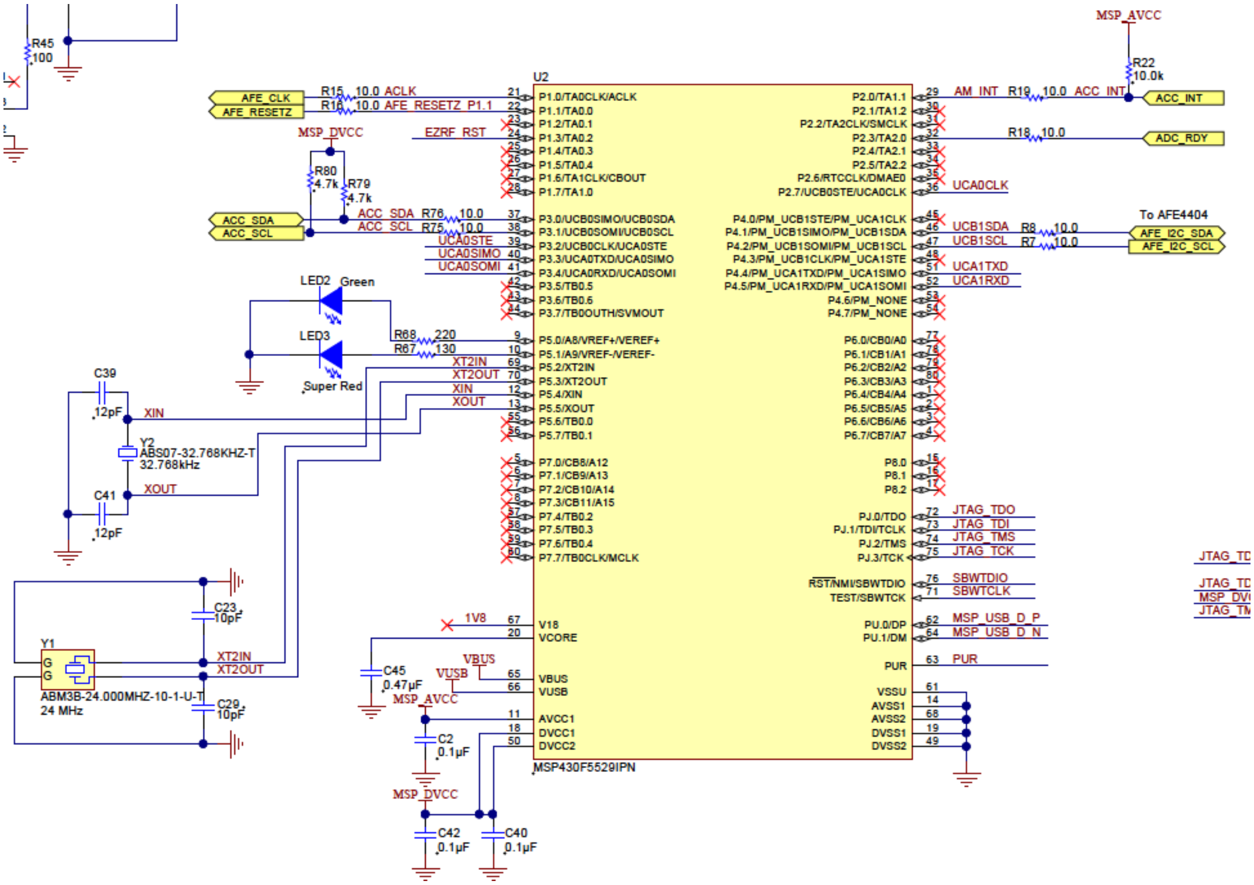


Figure 44: MCU for Evaluation Module

Table 15: Jumper Ports

Name	Pin No. on MCU	Pin No. on J3	Description
UCA1TXD	51	6	General-purpose digital I/O with reconfigurable port mapping secondary function Default mapping: Transmit data – USCI_A1 UART mode Default mapping: Slave in, master out – USCI_A1 SPI mode
UCA0SOMI	41	6	General-purpose digital I/O Receive data – USCI_A0 UART mode Slave out, master in – USCI_A0 SPI mode
UCA0CLK	36	4	General-purpose digital I/O with port interrupt Slave transmit enable – USCI_B0 SPI mode Clock signal input – USCI_A0 SPI slave mode Clock signal output – USCI_A0 SPI master mode
UCA0STE	39	3	General-purpose digital I/O Clock signal input – USCI_B0 SPI slave mode Clock signal output – USCI_B0 SPI master mode Slave transmit enable – USCI_A0 SPI mode
EZRF_RST	24	3	Reset
UCA1RXD	52	1	General-purpose digital I/O with reconfigurable port mapping secondary function Default mapping: Receive data – USCI_A1 UART mode Default mapping: Slave out, master in – USCI_A1 SPI mode
UCA0SIMO	40	1	General-purpose digital I/O Transmit data – USCI_A0 UART mode Slave in, master out – USCI_A0 SPI mode
MSP_DVCC	18	5	Digital power supply
GND	N/A	2	Ground

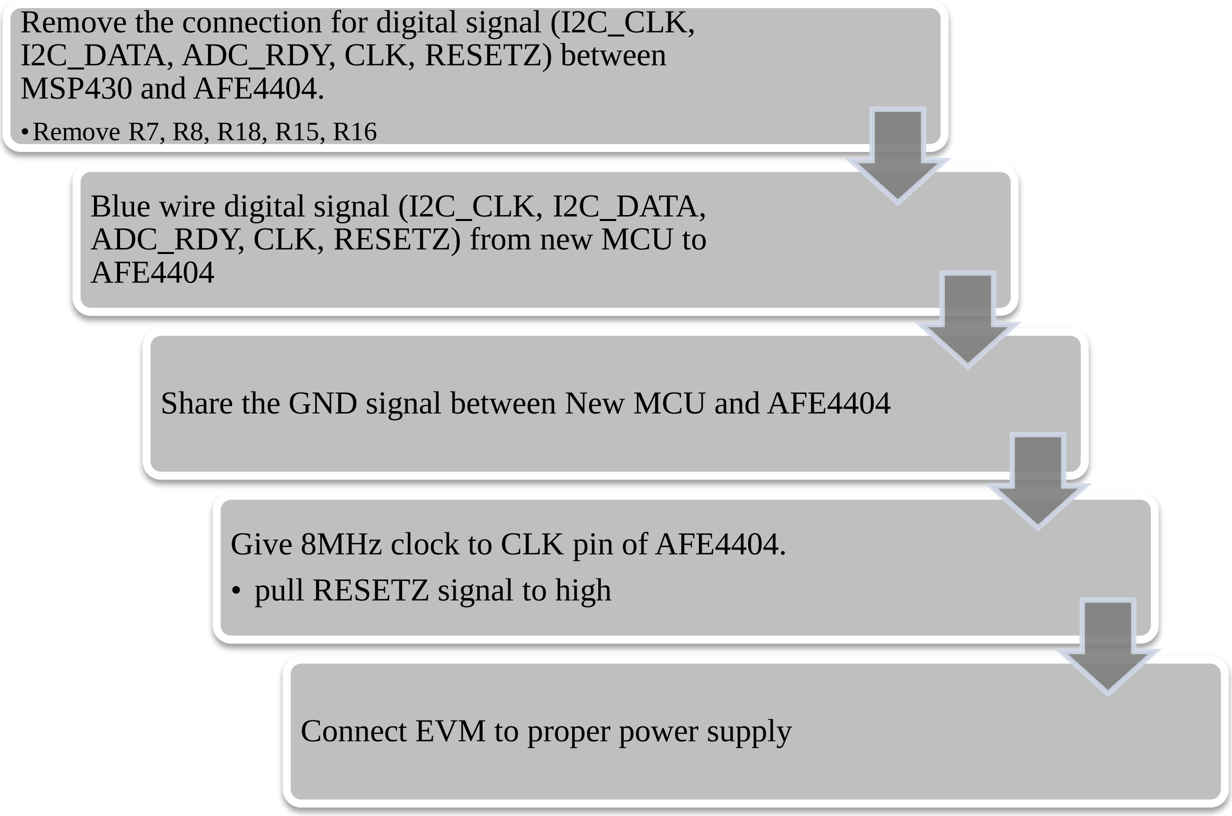
Table 15 is introduced to show the different ports that can be used for the connectors. The idea is to connect our PCB design to J3 via all the different ports. The complicated task that would follow this implementation is reprogramming the microcontroller of the evaluation module. Texas Instruments is unable to provide us with the software they used to program the microcontroller so we would basically have to start from scratch to reprogram it. This is something that can be done but would require a lot of time.

Another method that could be implemented is simply making a brand new oximetry and heart rate monitoring design. This would require us to use the AFE4404 chip and implement it with our own component. The issue that comes with this is that making a new oximetry design would be time consuming and may also not be as good as the current one. Alternatively, a method that could work is having two PCBs. The first PCB would be the AFE evaluation module and the second one would be our board. The two boards could work separately and send information via Bluetooth to the android phone. The information set via Bluetooth would then be processed on the android application. This would only be possible by making one microcontroller the slave and making the other one a master.

5.7.1.1 AFE4404 Software

The evaluation module that was donated to us by Texas Instruments includes a GUI (graphical user interface) that TI developed for the use of the evaluation module. Since the software developed is propriety we were not allowed to have access to the code. A solution to this is directly modifying the code by hacking into the microcontroller used in the evaluation module. The lay out of the module provides pins that allow access to reconfigure the entire module. In order to access the digital interface of the AFE4404 evaluation module using an external micro controller, the resistors mentioned in the evaluation datasheet will have to be removed and then wires will need to be soldered from the resistor pads closer to the AFE to the external MCU GPIO pins. It would be necessary to access the microcontroller so that new software can be implemented. In order to reprogram the AFE chip it would be necessary to install a development environment such as code composer or IDE (integrated development environment). The digital interface of the AFE is I2C. I2C is a bus used for communication between a master and a single or multiple slave devices. An issue that arises in doing this is that the pads of the resistors are extremely small and it is practically not possible to connect any sort of cables or pins so as to interface another MCU. A more secure route to take would be to make the AFE board separately with proper connectors so that it is possible to reliably connect the MCU. Table 13 explains the process needed to connect and external MCU.

Table 16: Steps to connecting external MCU to Evaluation



Remove the connection for digital signal (I2C_CLK, I2C_DATA, ADC_RDY, CLK, RESETZ) between MSP430 and AFE4404.

- Remove R7, R8, R18, R15, R16

Blue wire digital signal (I2C_CLK, I2C_DATA, ADC_RDY, CLK, RESETZ) from new MCU to AFE4404

Share the GND signal between New MCU and AFE4404

Give 8MHz clock to CLK pin of AFE4404.

- pull RESETZ signal to high

Connect EVM to proper power supply

As mentioned above, Table 16 describes the steps needed to connect an external microcontroller to the evaluation module. The external microcontroller would be already on the GOLD PCB. The first step would be removing the resistors Remove R7, R8, R18, R15, R16 in order to disconnect some signals. The signals that would be disconnected are I2C_CLK, I2C_DATA, ADC_RDY, CLK, RESETZ. The first signal I2C_CLK is a bus port that is utilized to control the clock of digital signal. Alongside the CLK is the data bus path. This path allows for information to flow to the microcontroller. The RESETZ signal is used to reset the clock cycle and start a new one. This is a key component in digital signal processing because resetting a clock cycle can introduce new commands or features. The resistors that are connected to these signals paths are in place to limit the amount of current that may flow to each signal. Limiting the current allows for power control and proper clock cycles. For example, if a strong input current is given to the circuit then there would be a spark in the clock cycles compared to if a small input current was provided. Resistors are a great way to limit current and provide control flexibility for a circuit. The next step in integration process is creating a blue wire digital signal from the new microcontroller to the AFE4404 evaluation module. The signals that would be connected with the blue wire are those that had resistors removed from step one.

The next important step would be creating a common ground between all the components. A common ground must be shared when creating circuits so that short circuits don’t occur. This ensures that there is a clear path for current to travel. Also it can reduce losses since distance between the two points will be minimized. Bringing the two modules together would allow for better current flow and therefore circuit performance. The next step in the process is to provide the CLK pin in the AFE4404 with 8MHz clock speed. The reason why you would need to provide 8MHz is because that is the operating point for the AFE4404. Providing the current clock speed would allow the circuit to properly function and complete all the desired functions. Finally the evaluation module would need to be connected to a power supply as mentioned in prior sections. The power supply that would be used for the evaluation module is a power bank.

The graphical user interface (GUI) provided with the AFE4404 evaluation module that Texas Instrument created includes different results. One of the results that it includes is showing how voltages change over a period of time when the sensors are pointing at the test source.

5.8 Aesthetics

The following section describes the artistic design involved in this project, particularly focusing on the mechanical design of the earpiece accessory and the electrical enclosure. Both parts have been modeled in the SolidWorks CAD software and they have been created to maintain a functional yet interesting structure and appearance.

5.8.1 Overall Design Aesthetics

For the earpiece design, the shape of the ear resembles a Bluetooth earpiece or audio earphones that hook onto the ear. With this design, the earpiece was modeled after an earlike shape and was intended to fit onto the ear of an average adult. The earpiece has a simple, clean appearance that has slots for the LED and the photodiode to slide in and out of when the two components need to be used or removed.

For the electrical enclosure, the appearance is very basic and it was designed as a rectangular encasing with a few slots for buttons and switches. Although simple, the electrical enclosure hides the wiring and the electric circuitry that may seem messy and unorganized if exposed. Hiding the breadboard and the other electrical components give the glucose measurement device a sleek look.

6.0 Integration & Testing

In regards to testing, it is imperative in a system design to be able to validate that all the separate sections and subsections are working to their expected standard. If a single section is not working, it is also vital that someone can identify the problem. In order to have a cohesive system the integration and testing must be established clearly and be administered thoroughly. Every single component must be tested in order to determine if each part is working properly. That includes every resistor and capacitor up to the microprocessors. After each component is tested individually then the completed subsystem must be tested in order to establish the design was implemented and created properly. Finally, when every subsystem is taken into account then the system as a whole will be tested and that ultimately leads to the end product. The actual integration of system is more in the scope of designing rather than quality checking. The challenge of integration is the last step in obtaining a finalized product. Connecting each subsystem brings the questions of which communication protocol to use as well as how power will be distributed. These problems will be looked into more in the following sections.

6.1 Integration

Integration in short is how to connect datelines and powerlines across systems when it comes to electrical components. In the software sense, the program integration is more of a transfer of information. Without reiterating too much on the power distribution and the communication protocol of this designs integration will be looked upon on a more high level view.

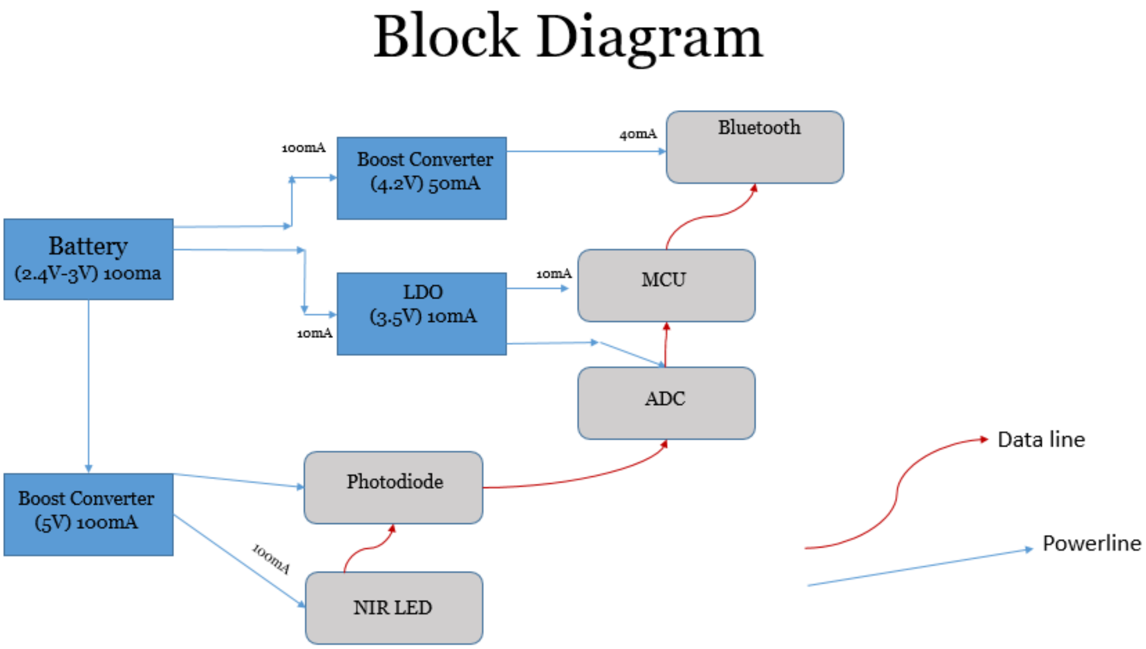


Figure 45: Block Diagram of GOLD

In Figure 42 you can see how the data path ultimately ends in a Bluetooth module. If the data path was to continue it would be leading into a phone app. Because this was an early block diagram it fails to show how the AFE will be integrated. The integration can be broken down simply as having two analog front ends that communicate with each other and then are transmitted to a phone application. Each specific system will be analyzed in the section below labeled system integration but in the biggest scope there are two microcontrollers. This is because integrating the AFE can be done by manipulating and changing that devices parameters to fit our need but this will be dealing with a system fabrication and the most time efficient way to go about this problem is to have a microcontroller linked with a data path to the AFE. This solution can be clear when it comes to Bluetooth. For example, Bluetooth on an MSP430 has been done millions of times but Bluetooth on the AFE4404 has substantially less resources and libraries to obtain the outcome that can be done on the MSP430. This is not the only problem using one microcontroller faces. When it comes to making a printed circuit board it is obvious that using an already made designed evaluation module and then creating another to suite us is easier than changing something that is already made. This might not sound easier because sometimes modifying something is easier than starting over, but this is not the case in our situation. This is because using the AFE in a new PCB will require us to reprint it with a new LED and a new photodiode which then the data transfer will be completely different as will the power so it would be easier to use a system that is already made. This problem has been debated on both sides multiple times, ultimately when modifying a PCB there is a lot more room for error than when it comes to jumping it.

6.1.1 System Integration

Various systems are being used in this project design and the integration of each specific device cannot be copied exactly because multiple parameters change for each system. In example there are a few sensors such as an oximeter and a near-infrared sensor and the op amps are not powered the same because the input bias current changes with higher power supplies in the power rails and this is only one problem with just linking all the power supplies to the same to the same source. In regards to software there are multiple programs running to parse and manipulate the code in order to do correct signal processing. This software integration must be put into a user interface in order to display the output. The multiple systems that will be looked at are the following: The oximeter, the ear green light attenuation, the near-infrared sensor, battery source, the microprocessor, the AFE4404, the Bluetooth module, the ADC, the software signal processing, and the software GUI. Some of these components only integrate with another while some of these components connect with more than five other systems which is where the challenges comes into play.

First and foremost, AFE integration closely relates to the oximeter because those devices are contained together in the AFE4404 evaluation module. In this list above an ADC is mention as a system of integration, but the EVM of the AFE already has its own ADC so it is not mentioned. The EVM comes with a lot of processes already finished because it is created to have direct applications built on it so it is designed for feasibility of integration. That is why the power system is self-contained as well with its own DC-DC converters. The AFE connecting to another microprocessor is where the challenge is. It can be passed through UART or through SPI and then the MCU has to manipulate the following information. The microprocessor is the tertiary step to the final product. All of the information sampled comes to MCU before it is transmitted wirelessly. All the sensor modalities pass through their respective ADCs and then are pushed through Bluetooth communication in a phone application. The problem with having multiple sensor information is that it becomes a cumbersome amount of information to transmit. This is the main problem when it comes to the communication integration. How fast and how much data is the two main questions and tradeoffs when it comes to Bluetooth.

Data structures are implemented in order to solve this integration problem and are spoken more in-depth in the respective sections, but in short. Information will be parsed to the biggest possible word length and then delivered one word at a time until the information is received. This will delay the data processing a certain amount but is negligible if done properly. The data will never be immediate due to complications of passing through Bluetooth and even in the analog section there slew rates to keep in mind, but for the scope of this design these are also negligible. In order for the analog portion to come to the MCU it must pass through the ADC which converts analog values into bits so it can be interpreted by the microprocessor and this is discussed thoroughly in the ADC design portion of this paper. Finally, once all the analog sensors are sampled and converted to digital readings and the MCUs transfer this data via Bluetooth, then the phone application does the heavy lifting in terms of digital signal processing and integrating the readings with a regression model and a statistical database. Once the computation is done then the phone application will integrate with a user interface to be used as an end product that will be able to tell you your glucose level.

6.1.2 PCB Fabrication

Printed Circuit Board fabrication is a process that has evolved over the years and now has a many supporting programs to expedite the process. Typically, in the past PCBs could be processed in house by using a photoetching chemical process that would use chemicals to eat away the conductive layers. With new technologies emerging PCB designs can now even be printed by printers, or machined by drilling machines in as short as a few hours. Despite the hurdles made in the recent decades a full scale design must still be made by a manufacturer since the previously mentioned methods are not as sturdy and do not last as long. Many manufactures will take board designs and print them such as Advanced Circuits, APCircuit, E-TekNet, Olimex, or PCBart. These companies will take your Gerber files which are the design of the PCB and then subsequently print the circuit board for a fee. The fee depends on the complexity and size of the design and can range anywhere from $20 to +$200, and these increasing costs have lead to the development of batching manufactures which take large orders then create very large tiles of PCB thus to reduce the cost of manufacturing. Companies like BatchPCB, Oshpark, and SeeedFusion Studio use this technique and thus can produce your circuit board a lowered cost, but often take longer to manufacture.

PCB design is a task that has many different software environments to use. There is KiCAD, Altium, OrCAD, EagleCAD, and many more. EagleCAD is a free software that anyone can have access to, whereas Altium and Orcad are not free and require a license. Luckily, I have access to the OrCAD software and will most likely do all the design in that environment since it is much more capable free softwares. With these computer aided engineering tools users can go from drawings, schematics, to analysis and manufacturing. In the OrCAD suite Capture is the centerpiece, and it contains extensive part libraries that can be used to generate schematics either with simulation by PSpice, PCB editor, or both. When one is designing in Capture, there is three layers of design that make up the schematic design. The three layers are the model, Symbol, and the Footprint.

The model is the analysis where the component is simulated by PSpice and can be simulated in context, with AC simulations, DC sweep, Transient Response, and more types of different circuit simulations. The symbols will represent the schematic however it represents the pins of each part. The footprint will represent the size and layout of each component on the schematic, for instance a footprint can represent a soic, smd, dip, or any other type of component footprint. Once these parts are designed the netlist will link all the components on the circuit board and transfer the file to the PCB editor. A PCB is consistent of two basic parts, the substrate and the copper traces. The substrate acts as a structure to hold all the parts together and as an insulator of electricity.

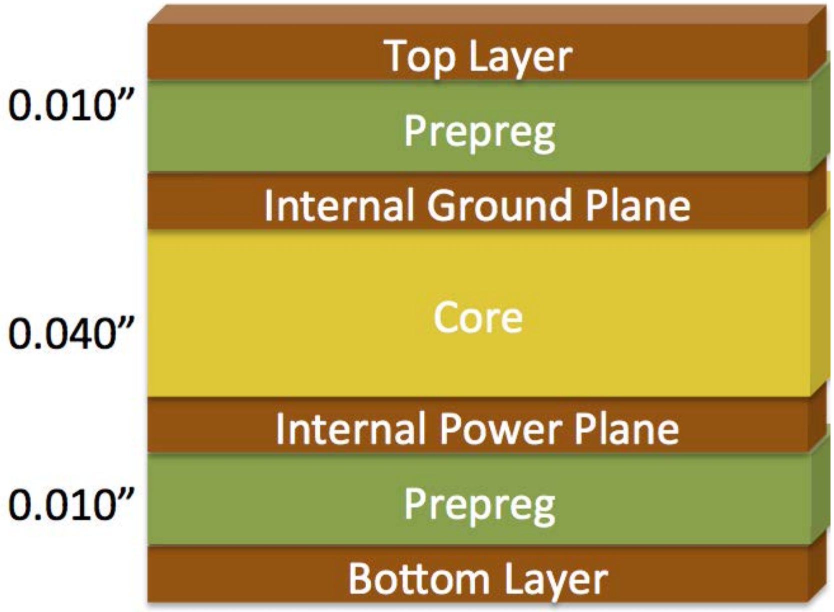


Figure 46: PCB Core Stack [46]

A circuit board consists of PCB Cores, which are copper clad substrates. [39] These layers can be stack upon one another and create multilayer boards. A PCB stack up can have vias that are plated, unplated, buried, and plated through holes that are back drilled therefore the design can be very complexed, or simply just one core that is double sided, and below is a picture of a 6 layer core stack.

Once the inner layers have been patterned, the cores are aligned and glued together. Once everything is properly aligned the pcb can be drilled for its through holes. Not all layers will have traces and some layers are planes which is good to provide low impedance connections to power and ground. [39] Ground and power planes are often inner layers and signal layers will be above and below the planes. A clearance can be used to provide isolation between plated hole and a plane. [39] After the through holes are plated the top and bottom layers are patterned using a photolithography process. After this a thin polymer layer is applied to the top and bottom of the board called the soldermask and then using photolithography to reveal the pads and holes. The soldermask protects the top and bottom layers from oxidation and prevents solder bridges between closely spaced pads. [39] PCB editor is used to design the PCB by generating files that will describe the board layers for the photoplotter and CNC machines. “There are separate layers are use for routing copper traces on the top, bottom and all inner layers; drill hole sizes and locations; soldermasks; silkscreens; solder paste; part placement; and board dimensions.” [39] Not all of these different layers are displayed equally. Therefore while designing the PCB not all of these layers will be apparent. Often the artwork (Gerber) files will generate more than 25 different layers. Here is a list of these layers, their extensions, and functions.

Table 17: Gerber File Extensions

Extension	**Function**
Example.brd	Main board project file
Assembly_TOP.art	Top Side assembly file
Solderpaste_TOP.art	Solder paste file top side
Silkscreen_TOP.art	Silkscreen top side
PWR.art	Power layer
...	Other layer and misc files
Example.rou	Board outline cutting path
Example.drl	Drill hole data

Now that the end product is familiar next is the process of creating a PCB starts with creating the schematic in OrCAD Capture. In this step you will add all the components that is included in your design. If a component does not exist in your library, you must find it in an external library or create the part. The next step is you will go through three different steps, first which is not necessary but is to create the PSpice model. This can be done so using PSpice and the datasheet to create an analytical model the part to be simulated. Once you create the model you will create the symbol of the device so that the pins number will be created. After this step you create the footprint by using PCBeditor. This will be covered later. Let’s all of our components are built and have corresponding footprints. Once the footprints are linked to each component and the schematic is complete and then you create the Bill of Materials (BOM). The next step is to create the netlist for the PCB editor. This led you to the PCB Editor program of the OrCAD suite. In this environment you will create the gerber artwork. First you begin by creating the board outline.

It is important to have design rule check while you create a PCB design. These DRCs will notify you once a design parameter is violated for instance having parts to close, or having traces be too long.

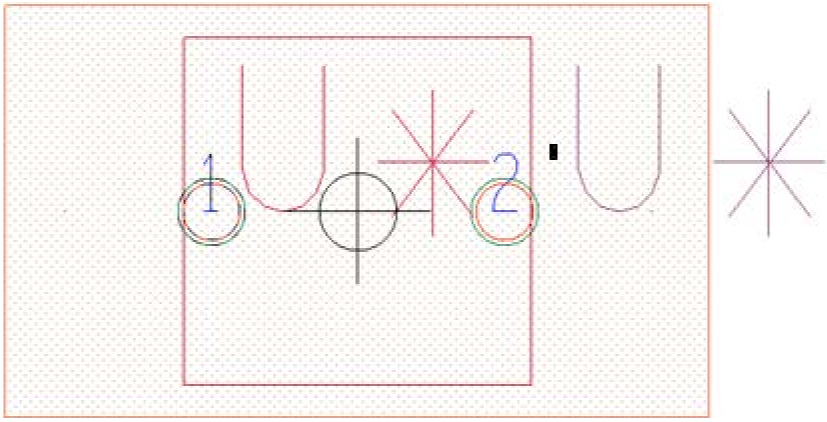


Figure 47: Thor LED1550e Footprint made in OrCAD

The board outline will have a board edge clearance so that parts do not get too close to the PCB's edges. Once the board layout is created the next step is to place the parts. Place the parts onto the board layout and each part will have multiple layer representations linked with it, such as the assembly top, silkscreen top, and so on. Once the parts have been then next step is to route the board, which can be done with a autorouter, but more often it is better to manual route the design. Then once the routing is done you can add the vias. The final step to finishing your design is to check your design against the DRC and see if there are any design rule violations. Once there are no errors in your design you are ready to create the artwork for manufacturing, and viola you've created a PCB.This is the general flow of creating a PCB design in OrCAD.
PCB routing can directly impact the performance of a PCB therefore the routing must be done carefully. The goal is to design the traces to reduce the amount of chatter or cross talk noise. "There are three goals in designing PCBs for electrical performance and signal integrity: (1) The PCB should be immune from interference from other systems, (2) it should not produce emissions that cause problems for other systems, and (3)it should demonstrate the desired signal quality." [39] Therefore the goal is to reduce the interference produced by EMI and the device should be designed to have good EMC. To design for low EMI and high EMC we must first consider magnetic fields and inductive coupling. Maxwell found that changing electric fields produce magnetic fields. "Emf induced into the conductor produces a current in the conductor that creates a magnetic flux that will oppose

the changing magnetic flux." and this known as self-induced electromotive force. The self inductance limits how fast the current can change in a conductor, which is why inductors oppose AC currents." [39] Therefore a primary conductor is near a secondary conductor it may induce emi in the secondary conductor. Therefore we must be conscience of the right hand rule for electrical engineers since the flux fields created might interfere with one another. It is good to have your secondary conductor be the return plane and that will reduce loop inductance. The goal is to design the circuits so that the circuits' traces create magnetic fields that flow in the same direction. "To maintain a small low loop inductance, we need to have the return path as wide as possible (low self inductance) and as close as possible to the signal path wherever possible (maximum coupling and small cross-sectional areas). The easiest way to do this is by using a plane layer as the return path." [39] This brings us to the Electric Field and Capacitive Coupling which is caused by parallel plates in the Figure below is an image comparing good EMC design to bad so that the design will reduce loop inductance.

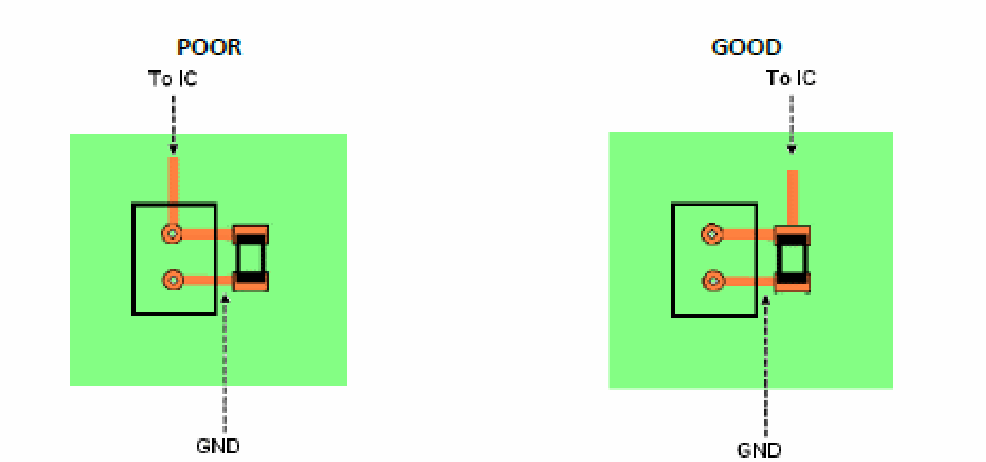


Figure 48: Good versus Bad EMC Design [47]

After your circuit board design is completed and your Gerber files have been processed, the next step is to send your design to a manufacturer. In our case we will likely choose the manufacturer that has low costs and a quick manufacturing time since we might be pressed for a deadline. Once you receive your PCB you must populate the board with the components. It is possible to pay a manufacturer to place each part which costs typically around $0.10 to $5 per part depending on the size and difficulty of placing the part. If the design has over a hundred parts then it might become too expensive to have the manufacture place the parts. Placing components can be a tricky process that can require certain soldering techniques to ensure that the device is properly connected to the board. The soldering technique that should be chosen depends a variety of parameters and often requires specialized tools. Soldering is the process of combining two pieces of metal using a metal filler that has a low melting point. For our fine parts using a smaller tip on the soldering iron, or if these do not adequately give a good joint then using thinner solder will do the trick.

For the SMD package type it is important to be delicate so that the circuit’s leads aren’t overlapped but still have a good connection. These package types, small pins, or hard to reach pins might need soldering paste and a reflow technique to properly handle the soldering. While soldering our joints we need to make sure that the tip is properly cleaned and is cleaned after soldering so that we prevent any weak joints or dry joints. The MCU that is to be used in our design will be a surface mounted package type and these types of components have specific methods to solder properly. The first step is to apply flux to the pad that will be soldered. To do so it is recommended to apply the solder to the soldering tip and then just touch the pad. Once that is completed just take your component and place it over the pad and then heat it with the soldering iron until the solder is wetted. After that take put more solder on your soldering iron and then touch that to the other pad. Often while using very small components it is necessary to use a solderpaste file that will plot the PCB with solder paste and then the components can be placed onto the paste then baked to create the solder joint. A picture of the pads of the PCB with solder paste on each component is shown below.

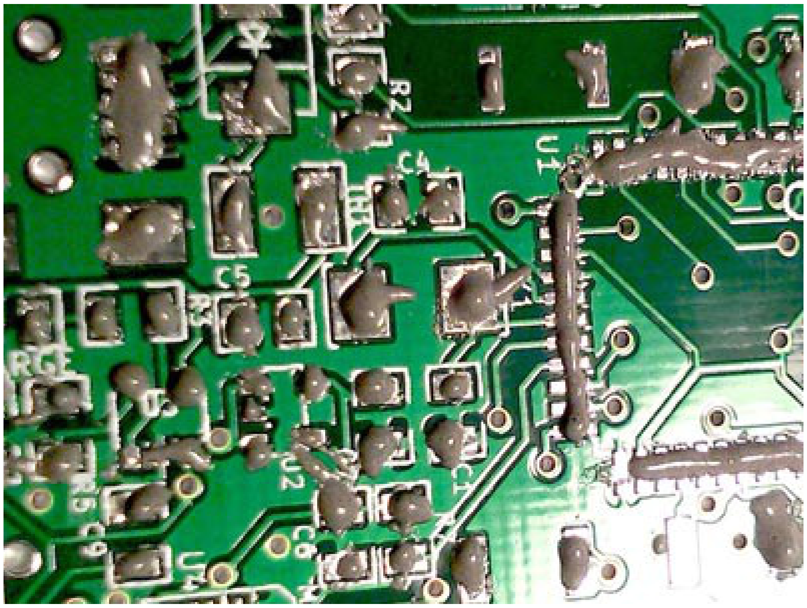


Figure 49: PCB pads populated with solder paste [48]

When soldering a larger IC do as you did before but start from the corner pin and then once that is placed properly you proceed to apply solder to all other pins. If the parts are incredibly small and difficult to hold in place steadily then there are alternative methods. While using OrCAD there is an option to create a Gerber file that will tell machinery to place the solder paste on the PCB. If you will place the solder paste by hand then the process will go as such, first you add flux to the pad, then you add the solder paste, and then finally you place your part then lightly touch the soldering iron to create a cohesive connection. Another method would be to flood with solder, in which you do the same technique as the first two methods, but the difference is that you would cover multiple pins with solder. Once the pins are covered with solder, the next step is to heat the solder and then remove the excess solder with a soldering sucker. This method is repeated until all solder bridges are removed and the connections are sufficiently soldered. The most common method of soldering in commercial manufacturing is doing reflow with a soldering oven. In this technique you apply the flux and soldering paste to each pad, and then place all the components on the soldering paste. Once everything is placed properly, simply put the circuit board in the soldering oven and this will heat the solder and thus create a solid connection between all the pins and the pads. It is important while soldering components and joints to be aware of dry solder joints list show in the Figure below. The component should have a complete wetting and should be connected securely

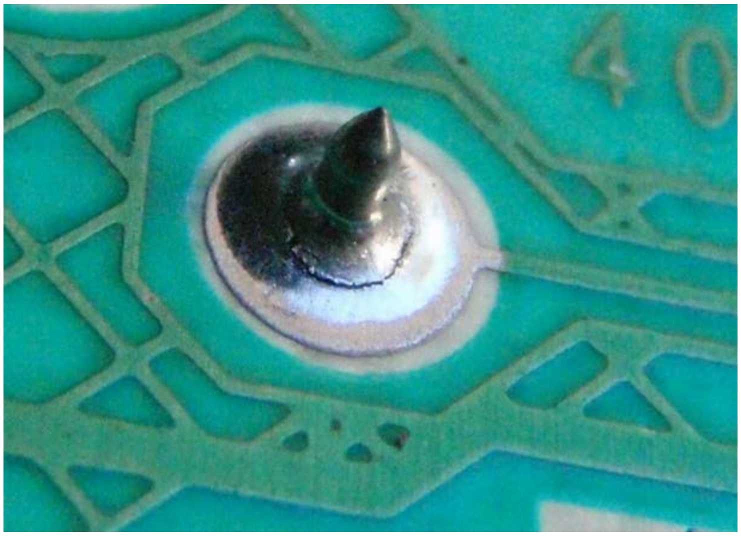


Figure 50: Dry Solder Joint

6.1.3 Parts Acquisition and Bill of Materials (BOM)

The following section provides the list of parts used for this glucose project and describes the purpose for choosing each of these parts. Most of the parts will have to be ordered from the two online suppliers Mouser Electronics and Digi-Key Electronics. The remaining parts, the 1550 nm LED and the Bluetooth Arduino module, have been purchased from Thorlabs and Amazon respectively. Texas Instruments have donated the analog front-end AFE4404 evaluation board, as they prefer that this module is used for pulse oximetry. With the funds provided by Texas Instruments and the team leader, these necessary items for glucose measurement can be purchased from the manufacturers or from the vendor Mouser Electronics.

6.1.3.1 Acquisition of the LED and Photodiode

These two items, the 1550 nm LED and the photodiode, must be purchased from the Thorlabs manufacturer and the Mouser Electronics vendor respectively due to the high quality and precision delivered from this company. In addition to its quality, Thorlabs seems to be one of the few manufacturers that carry these two specific items.

Figure 51: 1550 nm LED from (courtesy of Thorlabs)

The near infrared LED purchased from Thorlabs emits a wavelength centered at 1550 nm with a bandwidth of 1500 nm and 1600nm. The reason for this bandwidth is to account for any other light absorption by the glucose molecules in the wavelength bandwidth.

Figure 52: InGaAs Photodiode

The photodiode from Marktech Optoelectronics also is very unique as its sensitivity wavelength range is between 800 nm and 1750 nm. Although its bandwidth seems large, the photodiode successfully absorbs the desired 1550 nm wavelength with very little interference with ambient light sources. The photodiode requires the direction of the light from the LED point straight towards the material inside for the best result. In addition to its sensitivity of the desired wavelength, the photodiode material is made from an alloy compound called Indium Gallium Arsenide (InGaAs), which are used to achieve high speed in detecting the absorbed light intensity. With this high speed, the calculations performed for glucose measurement can be expedited with the measurement obtained from the InGaAs photodiode.

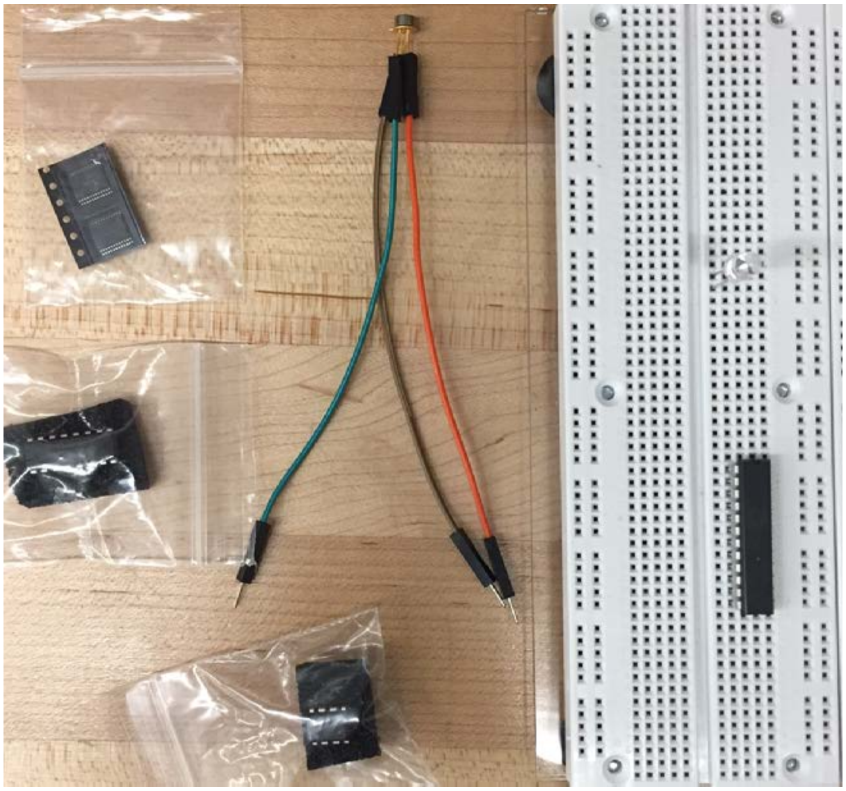
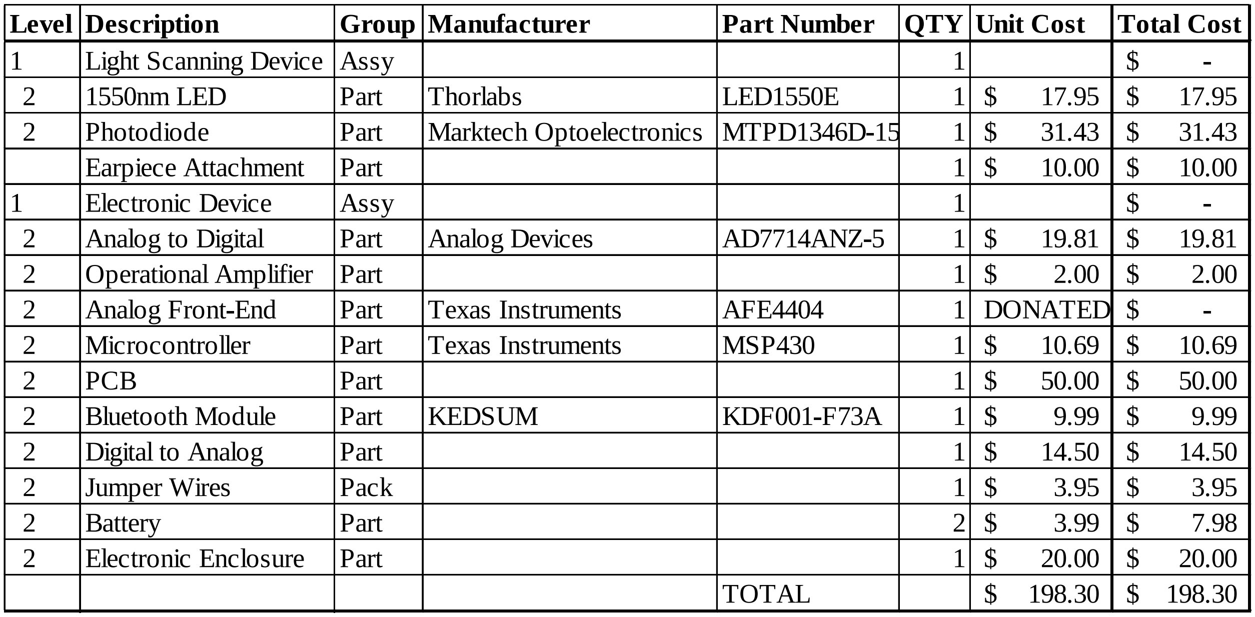


Table 18: Bill of Materials for GOLD Project



Level	Description	Group	Manufacturer	Part Number	QTY	Unit Cost	Total Cost
1	Light Scanning Device	Assy			1		$ -
2	1550nm LED	Part	Thorlabs	LED1550E	1	$ 17.95	$ 17.95
2	Photodiode	Part	Marktech Optoelectronics	MTPD1346D-15	1	$ 31.43	$ 31.43
	Earpiece Attachment	Part			1	$ 10.00	$ 10.00
1	Electronic Device	Assy			1		$ -
2	Analog to Digital	Part	Analog Devices	AD7714ANZ-5	1	$ 19.81	$ 19.81
2	Operational Amplifier	Part			1	$ 2.00	$ 2.00
2	Analog Front-End	Part	Texas Instruments	AFE4404	1	DONATED	$ -
2	Microcontroller	Part	Texas Instruments	MSP430	1	$ 10.69	$ 10.69
2	PCB	Part			1	$ 50.00	$ 50.00
2	Bluetooth Module	Part	KEDSUM	KDF001-F73A	1	$ 9.99	$ 9.99
2	Digital to Analog	Part			1	$ 14.50	$ 14.50
2	Jumper Wires	Pack			1	$ 3.95	$ 3.95
2	Battery	Part			2	$ 3.99	$ 7.98
2	Electronic Enclosure	Part			1	$ 20.00	$ 20.00
				TOTAL		$ 198.30	$ 198.30

Figure 53: Image of the Parts Currently Purchased

6.2 Testing Procedures

This section will go into great detail on how each part will be tested to ensure the device is working to specifications. It is important to make sure that the product works to specs so that the results on glucose measurements will be accurate. Testing each part individually will make integrating the entire system a smoother transition. A few of the components that need to be test is the DC-DC circuits, AFE4404 power supply, and the battery on the PCB design.

Testing the DC-DC circuits will be furthered discuss in this section. The main part of doing this is reviewing the datasheets for each of the recommended components from Texas Instruments WEBENCH software. The next part that will require testing is the AFE4404 power bank that will be added to the evaluation module. The purpose of this power bank is to eliminate having to use a laptop or computer as the power source. Finally the cell coin battery will be tested on the PCB design to determine if it was mounted properly and routed to the correct pins.

6.2.1 Power Testing

When testing the DC-DC conversion, the key values to measure are voltage and current. For the all the DC-DC conversions, the voltages should be within five percent of the expected output since the Webench design files had an output specification of maintaining within five percent of the desired output voltage ripple. It is important to make sure device is fully functional when all components are turned on. This function will also be tested to make sure the device can run for a minimum of thirty minutes.

In Table 19 the voltage and current tests are reviewed that will be performed for the two stages of DC-DC conversion. The time for which the device should be operating for will also be reviewed. A list of steps will be taken when troubleshooting the circuit on the breadboard and PCB as well. First will be checking the circuit to ensure that it has power being supplied to it. This is easily done using a multimeter set to a voltage range. Measuring the voltage using a multimter at the points where the supply enters the circuit board will provide a result. If the multimeter indicates that there is no supply voltage then there can be a number of possibilities to investigate. Common errors are that the battery could be flat if the equipment is battery powered. On-off switch faulty is another error, which can be checked by disconnecting any power source and checking for continuity across the switch.

The PCB design will require having test points to determine if each mounted piece is receiving the correct voltage. Designing the PCB test points will require placing an open connector hole with a positive and negative terminal. This can be designed in the PCB design software. The test points in the PCB would need to be at least 3.2 mm apart from board edges. This is done to prevent shorting any components in the PCB. Usually inserting the test points on the bottom of the board is an easier design method. Test point sites can be through-hole leads, dedicated pads or small diameter vias. Test points should be evenly distributed over the surface of the board. High stress in congested areas can cause board to warp. Further explanation on how the PCB should be designed for testing will be discussed in the next sections.

Table 19: DC-DC Testing Plan

#	Description	Conditions	Expected Results
1	Ensuring all DC-DC components properly convert input voltage to desired voltage.	The input of each supply will be connected to a DC power supply set to the correct voltage level and the output voltage will be measured using a digital multimeter (DMM).	Each supply circuit should have and output voltage that is 5% of the voltage level desired from section 6.
2	Ensuring all DC-DC components properly convert input voltage to desired current.	The loads will be tested on a breadboard to simulate what the final project would be like. Voltage would be applied and the current will be measured at each load.	Each supply should have a current that is within 5% of the desired value mentioned from section 6.
3	Ensure that all electronics are functioning with DC-DC conversions	Device will be turned on and have all components functioning at typical values.	Device should be able to fully function for at least 30 minutes.

Given the utilization of multiple test points on the printed circuit board, the ability to reduce test time is achieved. Additionally, in the event that specific components fail, it will take less time to identify the integrated circuit or other component that has failed. Given the sharpness of probe tips in the testing locations on the university campus, test points provide the ability to easily connect to locations where various tests need to occur. Solder locations are not ideal due to their variable nature of shape and contact surface and therefore a test point would be ideal.

To add additional ease of testing, breakout pins can be utilized to apply connectors or clamps to ensure that the proper waveform or frequency is being produced. Given the probe style of oscilloscopes, it would be ideal if there were a breakout pin to connect to. With these additional testing locations with breakout pins, testing time is reduced and therefore enables the overall testing sequence to be more thorough in less time and successful integration can be achieved. With a variety of test point locations, testing of resistors, capacitors, inductors, diodes, integrated circuits, light emitting diodes, and microcontrollers is possible. With the use of testing the multiple input and output pins of the microcontroller, robust quality can be achieved by ensuring that there is a way to validate the software functions.

6.2.2 Hardware Testing

The following section lays out the testing procedures for the microcontroller, the sensors, and the analog front-end AFE4404 evaluation module. These three components are the most important for the glucose measurement function and if these are not calibrated or installed correctly, measurements will be skewed.

6.2.2.1 Microcontroller

For our design we will use the Brown Out reset to protect the system architecture from damage from any surges or incorrect power levels and pull up/down resistors and diodes for protection. Our main mode of system reset will be utilized through the Power up clear or the Power on Reset. These resets will be useful if the system is not functioning properly or there are security issues, or detection of a violated passcode, and will reset the Stack pointer and Program counter. These resets can be triggered through a number of different methods however the hierarchy will function as such, BOR will be triggered by a system reset interrupt when either a device failure occurs and it would be ideal to return to initialized settings. The PUC and POR will be triggered by a non-maskable interrupt when the device is either in finished receiving and sending data, there is a violation of the watchdog timer, or other types of scenarios. Interrupts will be very useful in our system design since it will allow a point for our CPU to jump to a specific task that will implement what is needed. For instance, when our device is in its testing mode where it is receiving data, then sending it through the Bluetooth module, we can use an interrupt when the data in the registers are ready to be sent. Therefore a maskable interrupt can be used to trigger an event that will send data to the Bluetooth module and then using the RETI function to return to the task of data acquisition.

Our system architecture contains addressing for 64 different interrupts therefore this function will provide needed flexibility. The MSP430 chip is designed for low power operations and “the low-power modes LPM0 through LPM4 are configured with the CPUOFF, OSCOFF, SCG0, and SCG1 bits in the SR.” [20] This is useful because the device will save the SR and once taken out of low power mode can resume with previous tasks, and “program flow can be returned to a different operating mode by manipulating the saved SR value on the stack inside of the interrupt service routine.” [20] The device will operate mainly in the active mode and one of the four LPM modes just dependent on which mode can retain information and uses the least amount of power. To upload our program onto the MSP430 we will use the JTAG interface in 4 wire mode, or we can also utilize the Bootstrap Loader however this is not the preferred method. In the event we use the JTAG interface we can setup our device to be protected so that, “the device can be protected from unauthorized access by disabling the JTAG and SBW interface.

Table 20: JTAG Pin Requirement and Functions

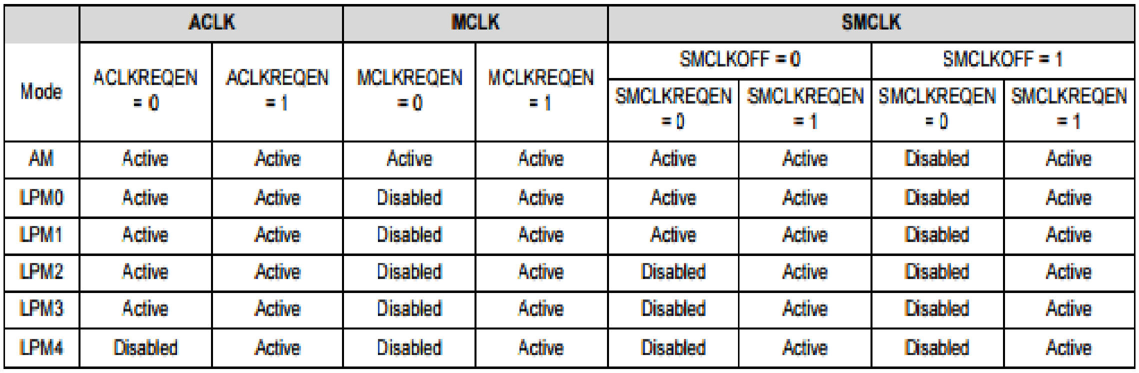
DEVICE SIGNAL	DIRECTION	FUNCTION
PJ.3/TCK	IN	JTAG clock input
PJ.2/TMS	IN	JTAG state control
PJ.1/TDI/TCLK	IN	JTAG data input, TCLK input
PJ.0/TDO	OUT	JTAG data output
TEST/SBWTCK	IN	Enable JTAG pins
RST/NMI/SBWTDIO	IN	External reset
VCC		Power supply
VSS		Ground supply

This is achieved by programming the electronic fuse. Programming the electronic fuse, completely disables the debug and access capabilities associated with the JTAG and Spy-Bi-Wire interface.” [20] To properly configure the timers and the peripherals these components need to be properly addressed, which can be done using the device’s peripheral IDs, which lists all the corresponding address spaces. For instance our 16kB RAM is addressed starting at 01C00h (0Eh × 0200h). The MSP430 includes a TLV structure which contains calibration values that can account for offset and gain, and can make the on board ADC modules more accurate. Using the Power management module the device can be configured to operate with different voltage values, which can be very useful when powering the LED driver. This on board power module can supply a wide range of voltages, 1.8v to 3.6v, “generate voltage for the device core (VCORE) with up to four programmable levels, supply voltage supervisor (SVS) for DVCC and VCORE with programmable threshold levels, supply voltage monitor (SVM) for DVCC and VCORE with programmable threshold levels, brownout reset (BOR), Software accessible power-fail indicators, I/O protection during power-fail condition, and has software selectable supervisor or monitor state output.” [20] The PMM provides a method to detect and monitor power supplies and then can function accordingly.

Our design will need to utilize the programmable threshold levels and brownout levels to prevent damage and keep consumption low. The PMM can also enable interrupts and these interrupts are routed to the Nonmaskable interrupt vector. Another function to use the PMM for is ensure that the I/O pins do not behave erratically in event of under voltage. In the event that our device uses an external supply voltage the AUX module can help control device operations. “The AUX module allows the device to switch between the primary supply (DVCC and AVCC) and up to two auxiliary supplies (AUXVCC1 and AUXVCC2) while maintaining full device functionality. When using an auxiliary supply, both AVCC and DVCC are switched to the same auxiliary supply.”[22] Using the Auxiliary supply monitor the program can effectively reduce consumption because, “supplies that are not currently in use and that are not software controlled (AUXxMD = 0) are monitored using a low-power comparator." By using a number of different clock sources for different operational modes device consumption will be reduced.

The Unified Clock System (UCS) provides a multitude of clock sources that each have advantages for different tasks, "the UCS module supports low system cost and ultra-low power consumption." [20] Of the five clocks contained in the UCS, XT1CLK, VLOCLK, or REFOCLK provides a low frequency and low power operation that would be useful when the device is in the standby mode. Another useful feature of the UCS module that some clocks can be used to reference against the FLL, and this can be used to reduce any jitter errors or mistiming. There exists three clock signals from the UCS module, MCLK, ACLK, and SMCLK, which are all divisible by 1,2,4,8, 16, or 32. The UCS module provides flexible and robust clocking system that allows the user to clock in a number of different ways for low power consumption, for instance, "low clock frequency for energy conservation and time keeping, high clock frequency for fast response times and fast burst processing capabilities, clock stability over operating temperature and supply voltage, and low-cost applications with less-constrained clock accuracy requirements." [20] As in Table 21 is a list of the different settings that are capable through the UCS module. These great capabilities are achieved by the fact that the UCS module allows, "all three available clock signals can be sourced from any of the available clock sources."

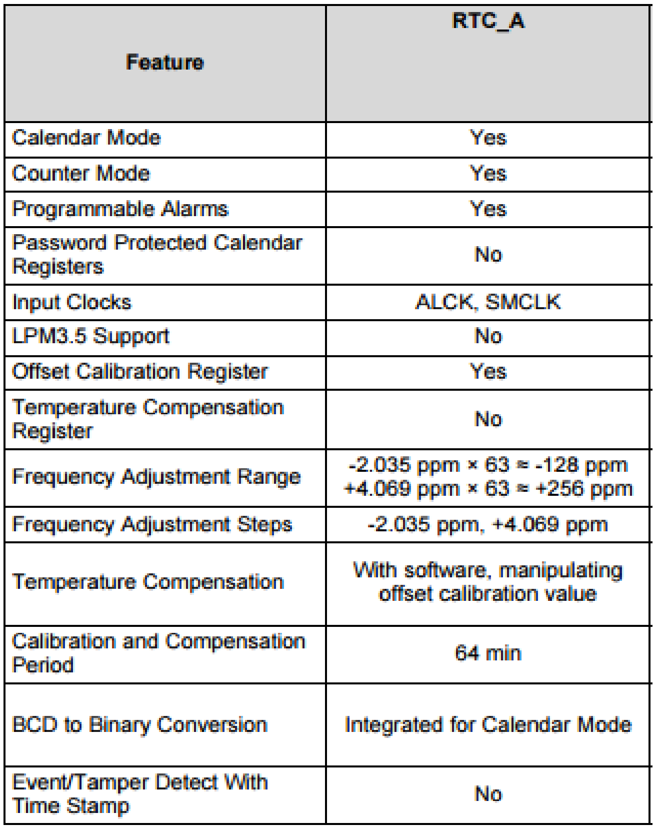
Table 21: UCS module Clock Request System and Power Modes



	ACLK		MCLK		SMCLK			
Mode	ACLKREQEN = 0	ACLKREQEN = 1	MCLKREQEN = 0	MCLKREQEN = 1	SMCLKOFF = 0		SMCLKOFF = 1	
					SMCLKREQEN = 0	SMCLKREQEN = 1	SMCLKREQEN = 0	SMCLKREQEN = 1
AM	Active	Active	Active	Active	Active	Active	Disabled	Active
LPM0	Active	Active	Disabled	Active	Active	Active	Disabled	Active
LPM1	Active	Active	Disabled	Active	Active	Active	Disabled	Active
LPM2	Active	Active	Disabled	Active	Disabled	Active	Disabled	Active
LPM3	Active	Active	Disabled	Active	Disabled	Active	Disabled	Active
LPM4	Disabled	Active	Disabled	Active	Disabled	Active	Disabled	Active

In our application the VLO(very low frequency oscillator) which operates at 10kHz will be used for standby mode in which the device is waiting to receive instructions to start testing. It might be best to source the VLO to the ACLK since the MCLK can be used for internal time keeping, and since the device will not need a fast clock a 10kHz clock will reduce power consumption greatly. Using a high frequency clock will be integral when communicating with the ADC module, therefore the XT2CLK will provide a high enough frequency clock to prevent and loss in data transmission. In applications for reminders the MSP430 utilizes a module called the RTC_A and RTC_B modules which can provide a real time clock and calendar operations, and can be configured as a general purpose counter. This can be used for programmable alarms and it utilized using a 32 bit general purpose counter.

With the ability to be clocked by the ACLK, and SMCLK these modules provide easy access and great capabilities, as listed below in Table 22.



Feature	RTC_A
Calendar Mode	Yes
Counter Mode	Yes
Programmable Alarms	Yes
Password Protected Calendar Registers	No
Input Clocks	ALCK, SMCLK
LPM3.5 Support	No
Offset Calibration Register	Yes
Temperature Compensation Register	No
Frequency Adjustment Range	-2.035 ppm × 63 ≈ -128 ppm +4.069 ppm × 63 ≈ +256 ppm
Frequency Adjustment Steps	-2.035 ppm, +4.069 ppm
Temperature Compensation	With software, manipulating offset calibration value
Calibration and Compensation Period	64 min
BCD to Binary Conversion	Integrated for Calendar Mode
Event/Tamper Detect With Time Stamp	No

Figure 54: RTC_A Module Overview

The Watchdog timer A (WDT_A), is a timer whose primary function is to perform system restarts after software issues occur. “If the selected time interval expires, a system reset is initiated.”[22] The watchdog Timer_A can also be used as an interval timer to generate interrupts. The WDT_A module can be configured as a watchdog timer or as an interval timer in the WDTCTL register, in our applications once our software is fully debugged the WDT_A timer will be used for interval timing. It is also possible to turn off the WDT_A module in the WDTCTL register by writing 1 to the hold bit as seen in Table 22. Timer_A module is a 16 bit timer that can capture/compare up to seven different registers. “Timer_A can support multiple capture/compares, PWM outputs, and interval timing and Timer_A also has extensive interrupt capabilities.”[22] Timer A module can be configured to four different modes of operation, up, down, continuous, and Up/Down and triggered on rising or falling edge. In our application we will use the Up mode of operation that is triggered with a rising edge and will utilize set/reset output mode. The capabilities of capture and compare will be useful for syncing multiple clocks, and storing different intervals that could be utilized. Timer_B is identical to Timer_A in almost every way except a few and will be utilized as a PWM source. Timer_D is also identical to timer A and B however can support high resolution modes to divide the clock source with a higher bit count. Other features include modules that can perform 32-bit MUX, a functional operational amplifier, 12 bit ADC, and those are features that seem unnecessary at the moment however since this is such an open ended problem these modules might be useful for signal smoothing and handling. Communication with the AFE4404 microchip will be important for data combination therefore it is necessary to understand the I2C communication that is used for the protocol to link with the AFE chip.

Table 22: WDTCTL Register Description

Bit	Field	Type	Reset	Description
15-8	WDTPW	RW	69h	Watchdog timer password. Always read as 069h. Must be written as 5Ah; if any other value is written, a PUC is generated.
7	WDTHOLD	RW	0h	Watchdog timer hold. This bit stops the watchdog timer. Setting WDTHOLD = 1 when the WDT is not in use conserves power. 0b = Watchdog timer is not stopped. 1b = Watchdog timer is stopped.
6-5	WDTSSEL	RW	0h	Watchdog timer clock source select 00b = SMCLK 01b = ACLK 10b = VLOCLK 11b = X_CLK; VLOCLK in devices that do not support X_CLK
4	WDTTMSEL	RW	0h	Watchdog timer mode select 0b = Watchdog mode 1b = Interval timer mode
3	WDTCNTL	RW	0h	Watchdog timer counter clear. Setting WDTCNTCL = 1 clears the count value to 0000h. WDTCNTCL is automatically reset. 0b = No action 1b = WDTCNT = 0000h
2-0	WDTIS	RW	4h	Watchdog timer interval select. These bits select the watchdog timer interval to set the WDTIFG flag and/or generate a PUC. 000b = Watchdog clock source /(231) (18h:12m:16s at 32.768 kHz) 001b = Watchdog clock source /(227) (01h:08m:16s at 32.768 kHz) 010b = Watchdog clock source /(223) (00h:04m:16s at 32.768 kHz) 011b = Watchdog clock source /(219) (00h:00m:16s at 32.768 kHz) 100b = Watchdog clock source /(215) (1 s at 32.768 kHz) 101b = Watchdog clock source /(213) (250 ms at 32.768 kHz) 110b = Watchdog clock source /(29) (15.625 ms at 32.768 kHz) 111b = Watchdog clock source /(26) (1.95 ms at 32.768 kHz)

In the MSP430F5529 chip the Universal Serial Communication Interface (USCI) module can utilize the I2C protocol. The USCI module supports multiple serial communication modes, which will be used in our case for Bluetooth, interfacing with the AFE chip, and sending commands to the ADC module. Typically the microcontroller comes with multiple USCI modules and these modules support UART, Pulse shaping for IrDA communications, automatic baud-rate detection for LIN communications, I2C, and SPI. The I2C mode features the compliance standards set by Phillips, 7-bit and 10-bit addressing modes, General call, START/RESTART/STOP, Multi-master transmitter/receiver mode, slave receiver/transmitter mode, standard mode up to 100 kbps and fast mode up to 400 kbps support, programmable UCxCLK frequency in master mode, a slave receiver START detection for auto wake up from LPMx modes, and slave operation in LPM4. The I2C supports any slave or master device that has I2C and any I2C can be used as a transmitter or receiver.

This device will use the 2 wire interface where SDA pin for data transfer and the SCL pin for clocking. Both of these pins need to be connected to a positive power supply with pull-up resistors. Since the USCI module is reset by a PUC it is recommended to configure the module only UCSWRST bit is set and this will prevent any unpredictable behavior. In the I2C communication a clock pulse is generated for each bit transferred and I2C operates with byte data in which data is transferred MSB. “The first byte after a START condition consists of a 7-bit slave address and the R/W bit. When R/W = 0, the master transmits data to a slave. When R/W = 1, the master receives data from a slave. The ACK bit is sent from the receiver after each byte on the ninth SCL clock.” [22] The start and stop conditions are generated by the Master and the START is initiated by a high to low transition by the SDA line while SDL is high, and STOP is initiated by a low to high transition by the SDA line while the SDL is high.

While in I2C mode the module can operate in four different modes, master transmitter, master receiver, slave transmitter, or slave receiver. In slave transmitter mode the slave device will send data serially through the SDA using the master’s clock pulses once the address that the master device sends matches the address of the slave’s. In slave receiver mode the slave device will push data into the address received by the SDA line only once the addresses match. Master transmitter mode is initiated by writing the desired slave address to the UCBxI2CA register and then generating the START condition by setting UCTxSTT. Once the bus is available the master device will transmit the data stored in the address specified through the SDA line using its own clock pulses on the SCL line. Master receiver mode is initiated by writing the desired slave address to the UCBxI2CSA register, and once UCTxSTT is set this will generate the START condition. Afterwards the USCI module checks if the bus is available and once it is available it will transmit the slave address. After the slave acknowledges the address it will clear the UCTxSTT bit and then sends the first byte of data and will set the

UCRXIFG flag, and Data is received from the slave, as long as UCTXSTP or UCTXSTT is not set. Our application will call for both using master receiver for communication between microcontrollers. “When the USCI is in master mode, BITCLK is provided by the USCI bit clock generator and the clock source is selected with the UCSSELx bits. In slave mode, the bit clock generator is not used.”[22] The USCI module is an incredibly flexible and robust communication module that allows for multiple clocking frequencies, a multi-master mode, and applications for use in Low powered modes. The configuration of our MSP430 device will be as a master receiver and thus will enable communications with the AFE4404’s MCU, which will be configured as a slave transmitter as pictured in Figure 55.

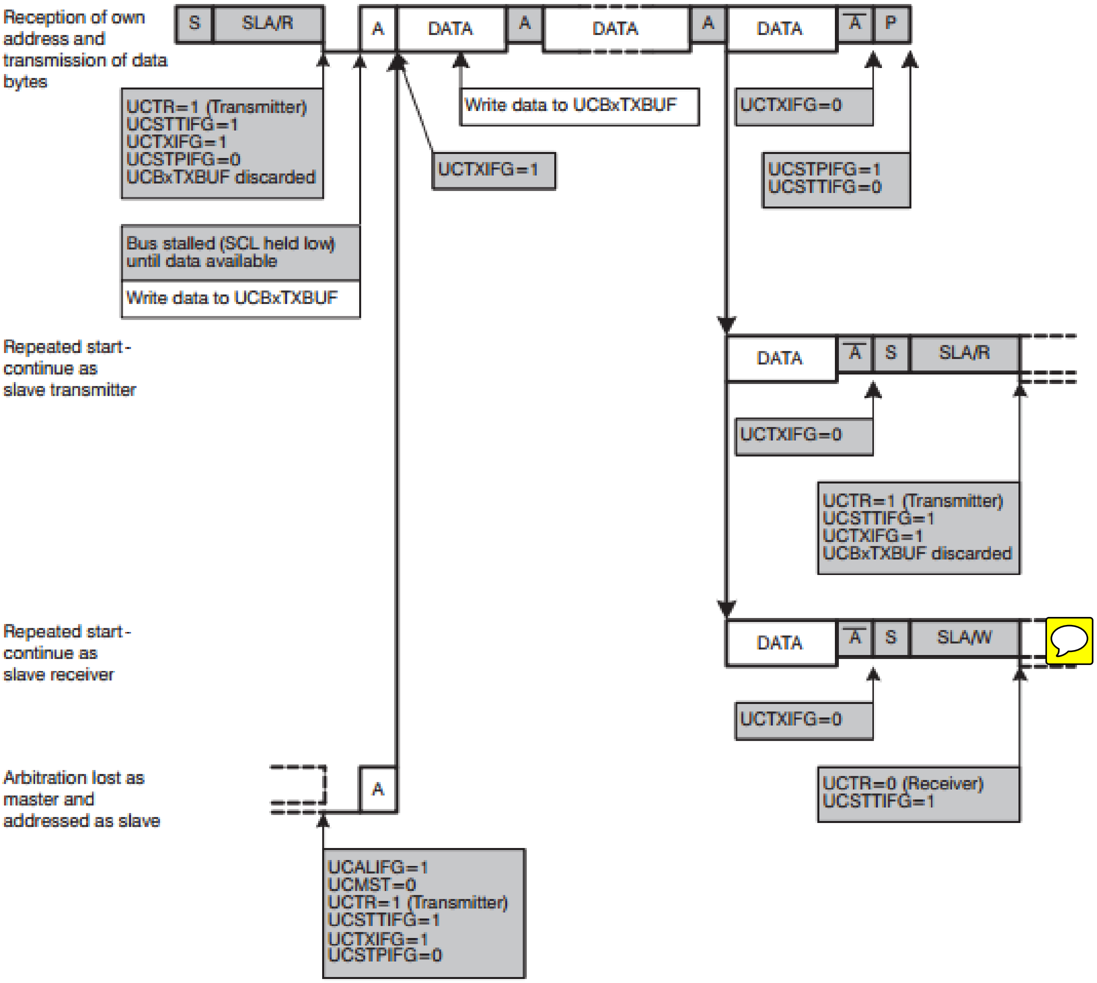


Figure 55: I2C Slave Transmitter Mode courtery of Texas Instruments

In Figure 54 [49] the device when in slave transmitter mode that when data is transferred there are specific flags UCTxIFG is set high then the device will retrieve the data from the specified address given by the master, or in the register called UCBxTXBUF while the SCL line is held low. Once the data is filled and ready to be sent the UCTxIFG flag is set once again which will tell the master that the data is ready to be sent.

6.2.2.2 Sensors

When testing the sensors system it is a good idea to test if individual components are working properly. This means that each resistor should be checked and each capacitor on a multi-meter. After each farad and ohm are confirmed to their expected value then one can begin testing each respective operational amplifier. The first test circuits to do on an operational amplifier is to test if they are working properly in terms of a simple inverting circuit and a simple buffer.

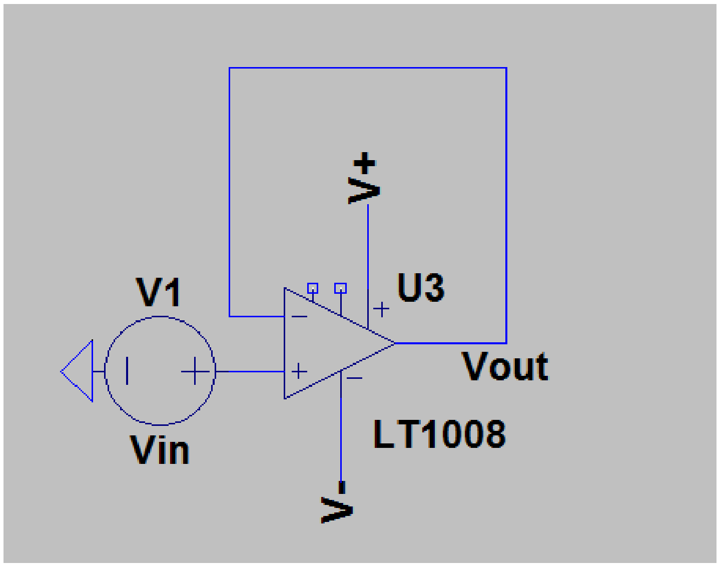


Figure 56: Schematic of a Unity Gain Buffer

This unity gain buffer will determine if the input and the output are the same and it more or less is indicative of if the operational amplifier is working in its most primitive form. In the schematic the LT1008 is being shown but this can and should work on all the op amps used and must be tested on each one. After this test is administrated then a simple non inverting amp should be used in order to test if a gain will work. If the output is -2 * V_{in} then you can be confident that the operational amplifier works with a unity gain and a simple gain stage so it is more than likely not damaged or shorted on the inside.

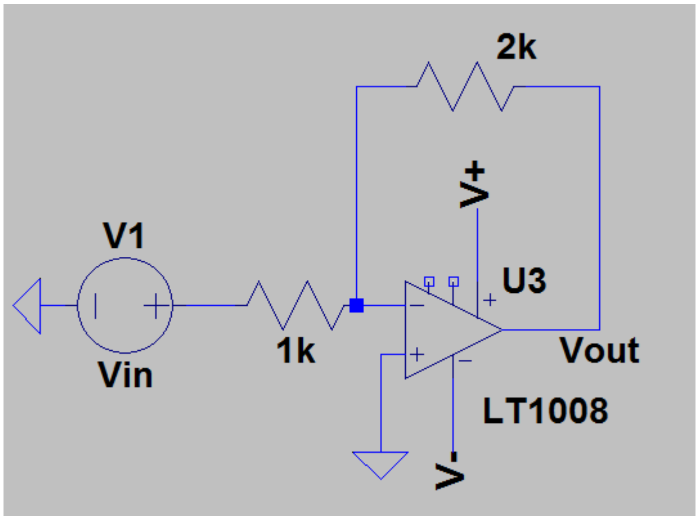


Figure 57: Schematic of an Inverting Op amp with Gain of 2

After you test the op amps for their ideal behavior than each individual operational amplifier has specific test that can be administered to check their ideal values. Such as the dc offset or the slew rate or the frequency conversion. Below in Figure 57 is a .1Hz to 10Hz noise test circuit. It is note worth that the second op amp in this Figure must already be tested because this test is only for the first stage. Because the first stage doesn't have an input the output should have a relatively small voltage. Something lower than an microvolt if you are fortunate. If this circuit does not output a low output than you have a noisy amplifier and should not use it for precise applications such as a glucose meter, so get a new one.

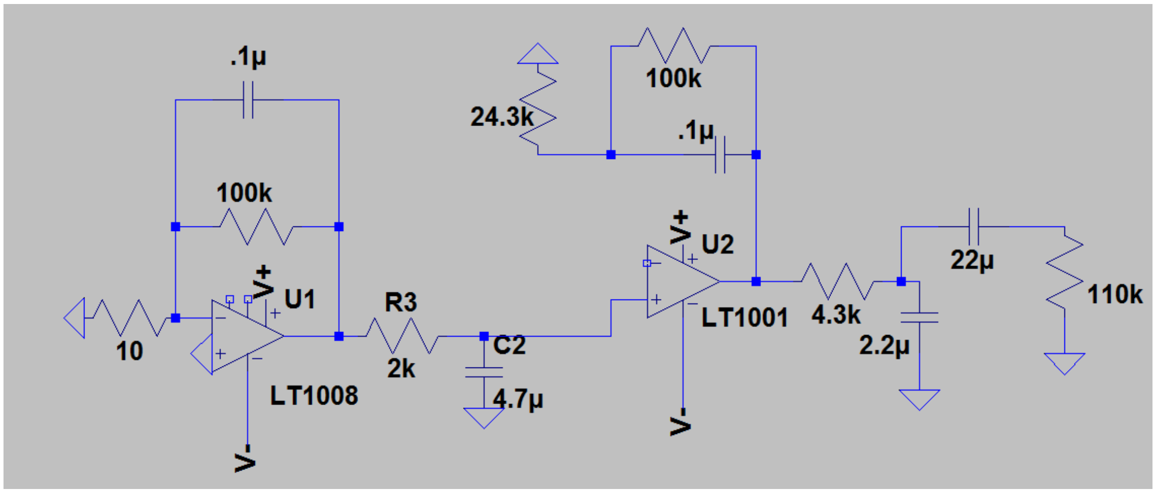


Figure 58: LTC1150 Low Input Bias test circuit

After the LT1008 is tested than the LTC1150 will be analyzed. This op amp has a way higher priority because if this circuit has a big noise or a big input bias current than the whole project will not be able to create accurate results. This is a very serious problem in the application of a glucose sensor because people can die if their glucose is too low and a reading the offset voltage tells them that it is sufficiently high enough. The first Figure test if offset voltage is small enough for this device to work properly.

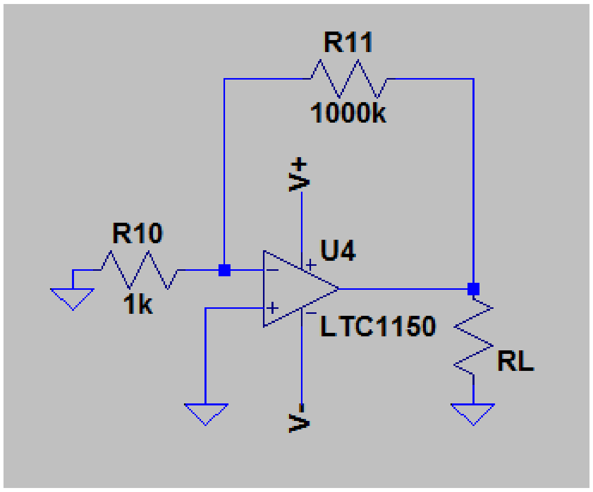


Figure 59: LTC1150 DC Offset test

This is essentially the inverting op amp without a source so the noise would be amplified by a magnitude of a thousand and then voltage divided. The offset voltage is the pertaining to the differential amplifier in the circuit and it will be clear if it is too high and that this part needs to be replaced or not.

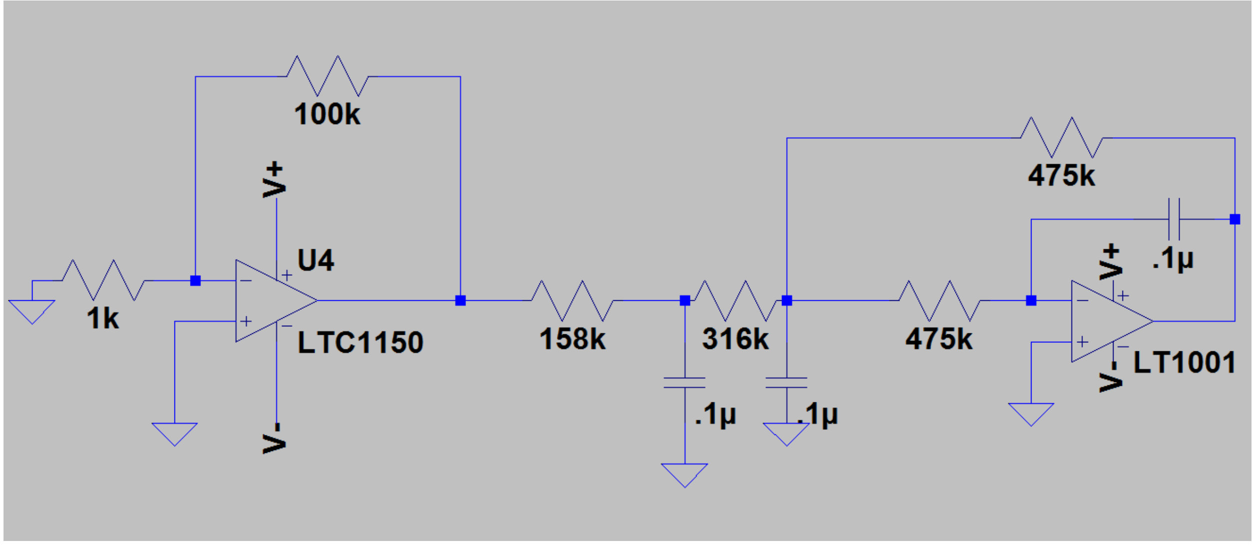


Figure 60: DC-10Hz Noise Test Circuit

In the Figure above the LTC1150 will be tested in the DC-10Hz Noise Test Circuit from the datasheet. For 1Hz noise BW, increase all the capacitors by a factor of 10. This circuit makes a gain of one thousand in the first stage and then decouples and dissipates the voltage followed by a filter to check if the input bias is marginally small. Although it has been said numerous times this project won't work if the input bias voltage is too high. The photodiode has such refined resolution that an infinitesimal spike in noise will make the readings useless.

Following testing the operational amplifiers the ADC and the microcontroller need to be tested as well. The ADC can be tested connecting the unity gain schematic above in Figure 55 and the inverting op amp directly into the ADC instead of an oscilloscope and compare it respectively. The most common way to test a microcontroller is using the preset code for LED flashes and having the serial pins testing again with inverting op amp schematic. Once the microcontroller and the ADC are confirmed to be working the LED and photodiode must be tested. The easiest way to do this is to have three NIR LEDs and three photodiodes and to three test each and get verify if they are precise. If they are precise than you can assume they are accurate.

Another form of testing that is worth mentioning is not to validate the components and verify them but to conduct the appropriate regression models of glucose levels and test if the systems are designed well. In other words if the system components are working properly but the design does not work than this should be tested in the following method. In order to check if the glucose level reflects the reading of the user than you must check with an invasive method. For this method we will be pricking ourselves while running our design to see if our final outcome is right but we obviously will test each system on the way so we don’t print a whole circuit board and just hope we are correct. The way to test every step is to start with detecting glucose in a water beaker. With this experiment you can control the glucose level with the amount of solution you put in the beaker water. Once you know there is for example fifty percent glucose to fifty percent water you can receive a voltage reading and compare that to 0%, 25%, 75% and 100% glucose levels and the voltage readings in the photodiode on the sensor should increase linearly by the respective percentage. Also this method could work without the sensor and just the photodiode plugged into an oscilloscope. After the photodiode is verified with the oscilloscope than the next logical progression is to put it into the sensor schematic and test if the noise is rectified and minimized. Afterwards the ADC can be integrated in the testing to see if the resolution is accurate.

In conclusion, when it comes to testing integration it is vital to verify every single component before moving into the whole system testing. Afterwards it is crucial to test the design of the system to verify that it is capable of obtaining the desired results. When the design and implemented the end user essentially is testing the design every time they try to use it. This is why it is beneficial to think of the design and testing and integration as the same thing that way it is easier to keep track and develop appropriate techniques. When it comes to testing the PCB then every electrical connection has to be tested in order to verify the design is coherent. There are often overlooked issues in PCB design and the first iteration of a printed circuit board can almost always be assumed to have errors. The best thing you can do when receiving a circuit is to identify all the mistakes the first time so you don’t solve one and assume that was the only one. Current transfer through each wire is an issue to keep in mind because the material has to thick enough to be able to handle all the wire. When it comes to testing the MCU every pin that is being used should be verified with another pin of the same sort. For example analog pin 1 should equal analog pin 0. Also it is worth mentioning testing oscilloscopes and wave generators when running samples in the lab should also be verified and not taken for granted as assuming to work because this could lead to inaccurate results.

6.2.2.3 AFE4404 Testing

The AFE4404 evaluation module is used for pulse oximetry, which detects the oxygen levels in blood, and is essential for blood glucose measurement. Since Texas Instruments provided the AFE module, the analog front-end shall work as intended but to incorporate the module with our design, the steps for testing the AFE4404 are listed below.

Table 23: Testing Protocol for the AFE4404

Testing Process	Procedure
Step 1	Simulate the pulse oximetry process without the LED and the photodiode installed.
Step 2	Test the pulse oximetry process on multiple volunteers to compare values obtained from the pulse oximetry measurement.
Step 3	Calibrate the AFE4404 module to maximize efficiency and produce accurate results (if needed).
Step 4	Incorporate the testing of the mechanical earpiece and the electrical enclosure with the testing of AFE4404 evaluation module.
Step 5	Test the glucose measurement process with the pulse oximetry procedure.
Step 6	Connect the PCB to complete testing the entire process.

The first step of the testing process is to ensure the AFE4404 works as intended without attached and wiring any new electrical components to the board. Since the entire schematic and circuit board is already built, the AFE4404 can be turned on and instantly begin testing for correctness. The second step of the testing process requires that we test the pulse oximetry process on multiple volunteers to ensure that the AFE4404 is working correctly and that the module can work on multiple people instead of one test subject. If the AFE4404 does not work as intended or the results are completely inaccurate, we shall calibrate the AFE4404 module according to any instructions provided by Texas Instruments.

Once the AFE4404 is calibrated and ready to go, we can place the module inside the plastic electrical enclosure and install the LED and the photodiode to the earpiece. At this stage, testing will now involve the glucose detection measurement to validate that the AFE4404 is producing accurate results. If both the AFE4404 module and the glucose measurement device cooperate correctly, then the circuit can all be connected, including the PCB if available.

6.2.2.4 ADC

The ADC module will serve as a very high resolution module that will return data from the photodiode. The component that was chosen to do this implementation is the AD7714anz ADC integrated circuit. The features listed in the device's datasheet are, "Charge Balancing ADC,24 Bits No Missing Codes with 0.0015% Nonlinearity and a Five-Channel Programmable Gain Front End that have Gains from 1 to 128, the device can Be Configured as Three Fully Differential Inputs or Five Pseudo-Differential Inputs, uses the Three-Wire Serial Interface, SPITM, QSPITM, MICROWIRETM and DSP Compatible communication protocols, can use either 3 V (AD7714-3) or 5 V (AD7714-5) Operation, has Low Noise (<150 nV RMS) and Low Current (350 A typ) with Power-Down (5 A typ) with grades of +2.7 V to 3.3 V or +4.75 V to +5.25 V Operation, a 0.0010% Linearity Error, a –40 C to +105 C Temperature Range for operation, uses a Schmitt Trigger on SCLK and DIP, has Low Current (226 A typ) with Power-Down (4 A typ), and comes with Low-Pass Filter with Programmable Filter Cutoffs Ability to Read/Write Calibration Coefficients." [49] Therefore this device can operate as an analog front for low frequency measurement applications. The device accepts low lever signals from a transducer and then outputs a serial data word. By implementing a sigma to delta conversion the device can realize up to 24 bit ADC with no missing code. [49] The input signal is fed to the programmable gain amplifier and then fed to the digital filters which can be configured through an on-chip control register. This device features three differential inputs and a reference input. The AD7714 module can perform up all signal processing for up to five channels. The serial interface is a three wire setup and gain settings, signal polarity, and channel select can be configured in software by means of the serial port. This chip also features self calibration, system calibration, and background calibration for any noise offset and drift values and allows users to read/write the calibration settings. [49] The AD7714 is a low power device that is ideal use in power loop systems, which only consumes less than 500uA to 1mA depending on the clock frequency. The programmable gain will be useful for any signal condition that might need to be used for the photodiode signal. "This part contains on-chip registers that allow control over filter cutoff, input gain, channel selection, signal polarity and calibration modes." [49]

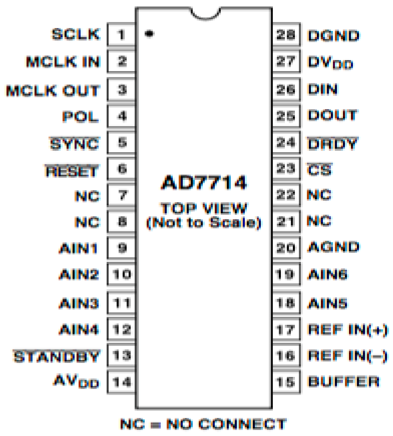


Figure 61: AD7714 pin layout [49]

This device can be communicated with using a three wire interface and will require a crystal resonator between 400kHz and 2.5MHz placed between MCLK IN and OUT. For lower power consumption we will use a lower frequency clock. Other components that will be need are bypass capacitors between AVdd and DVdd and ground, and a voltage regulator for the REF IN(+) pin supplying a constant 2.5V reference. [49] The photodiode will route between AIN1 and AIN2, and any extra photodiodes will do the same for AIN 3 & 4, and AIN 5& 6, which is the analog input signal. A 5V signal can be sent to the RESET, SYNC, and STANDBY pin for a basic configuration. DOUT will send data to any MISO, DIN will receive data from any SIMO pin for serial data in, and SCLK will be the serial clock that will be connected to a ACLK pin, or any pin capable of generating a clocking signal. The DRDY pin will notify the MCU when the data is ready and can be sent to any GPIO pin.

6.2.3.2 Microcontroller

Testing the microcontroller is a task that will include both physically probing our designed circuit board with a logic analyzer, and creating software for each module to test if each module is working properly. Testing the microcontroller will involve each specific module that we plan on using on the microchip. For instance the plan is to utilize the USCI module for I2C communication between microcontrollers, and with Bluetooth. Our test might function as a software in which we will check display the data of a register in the USCI module and compare the results of the data to the expected data in the datasheet. This can be done with both software, and with hardware, since the USCI's module will route its information to an external module. Logic analyzers can range anywhere from $100 to over a thousand dollars therefore using an oscilloscope might be the best option for hardware. Luckily there are software realizations of logic analyzers, in Figure 61 shows the logic of the SPI protocol, output and Code composer studio can support plugins for this purpose for some types of MCUs,“ if your device supports CPU trace and you had an emulator that can support CPU trace, you can monitor the program bus to get a record of the PC and graph that data via the trace analyzer.” [50]

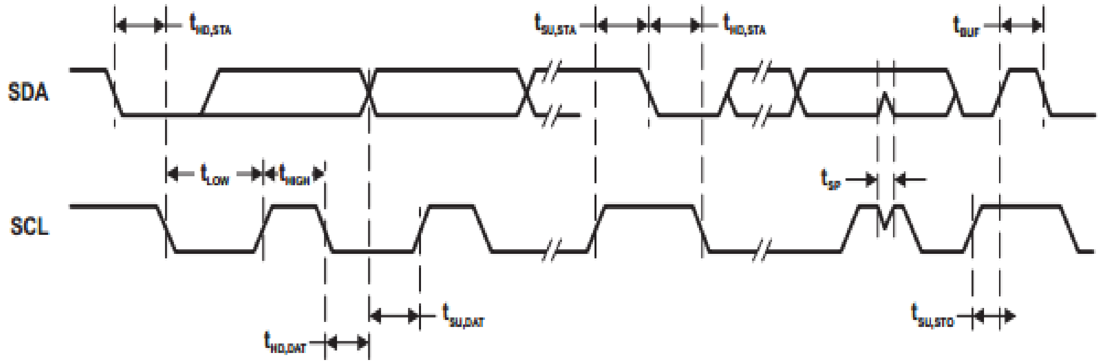


Figure 62: SPI Logic Analyzer courtesy of Texas Instruments

According to MSP430’s Wikipedia page the MSP430F5529 does include watchpoints and which “can monitor the program and data bus for various activity. [50] Therefore we can setup various watchpoints to monitor for erroneous activity. Although it does support watchpoints, unfortunately it seems as though it does not support CPU trace. There are other ways to test the Microcontroller’s internal logic, for instance Code Composer Studio does have a register view which will be sufficient for us to view the data. Therefore our test might be structured as the following. Upload a program in which we will test a specific module, for instance the ADC module. The program will send an instruction to the module and then the module will send the data back to the microcontroller. The data that will be produced will be a controlled signal in which we will know what the outcome will be, for instance the voltage amplitude will be constant and occur at a specific timing interval. Once these steps are complete we then can probe the register value and verify that the device is properly working. Using the oscilloscope it will be possible to analyze the traces, however the timing of the signal might be more difficult to ascertain, and it would be very useful to have more than four input signals. Our software tests will include creating a specific code that will configure the MCU to operate with a desired mode. Once this is uploaded testing signals sent to peripherals, and values stored at addresses to verify the operation of the MCU.

“The testing architecture will greatly depend on how your MSP430 MCU code has been structured. For native C applications, a series of function calls may be required to determine an ultimate result, and in a more object-oriented model, a simple constructor initialization and call to one public function may yield the results desired.” [50] The key to testing the unit properly is to try to get as much code coverage as possible and to vary the inputs in many different ways. “The tests are designed to instantiate objects at any point within the test execution and make appropriate calls to the public member functions of each object.” [50] Since the microcontroller is main controlled by the registers it is very important for the program to be able to execute read and write from the registers. “The key for register definition in the testing program is to redefine the macros that initialize each register.” [50] Therefore using this function to call different registers will not run into errors due to different register sizes, bit overflow, and to be able to do bitwise operations.

Finally, to complete a program that would use a function to define each register we must create a program to sparse each definition in the header file and create a program file that will properly call each function. Testing the flash memory is task that will require read and write capabilities to be able to detect what is actually stored in each flash memory bank and whether or not the flash memory is properly storing the information. “The MSP430 program should invoke a method to set variable addresses at runtime, which are then referenced for all reads and writes. The default implementation would set addresses to the BANK definitions, and the test program would then be able to make a subsequent call to set addresses from a defined memory space that was setup on the desktop environment.” [50] The trick to testing the flash is that the program must be able to account for the Wait bit so that the program is not left hanging, therefore by detecting when the bit is set will tell the program to read the bank. It is considered more difficult to test the communication buses since these channels require an outside source to generate interrupts. Therefore a method to do so would be to write a program that can generate the interrupt instances. By doing this we can bypass the outside source that would call the interrupt and do so within the testing environment. Similar to communication buses it is also difficult to test timers since their reliance on interrupts.

Simulating timers is possible through software and can be used for testing purposes on the device. An important concept to implement while doing testing is Thread Synchronization. This ensures that the test code does not get stuck at a certain instruction and that the register values get set at the proper timing intervals. An example of this is while simulating a communication bus that relies on an interrupt that once the interrupt is called the main worker thread does not become out of sync and overtake the main thread and therefore the proper wait conditions are required. The goal of tests is to, “design to instantiate objects at any point within the test execution and make appropriate calls to the public member functions of each object.”

6.2.3 Software

Each component of our Android application should be tested to ensure it is carrying out the functions and activities we aim for. In the general order of their usage and implementation, we describe our testing procedure for a physical Android mobile phone.

6.2.3.1 Create User

A user must create an account to log in to the application. Successful creation of an account will result in secured access into the application and the loading of user-specific data.

1. At the login page, select *Create User.*

2. Enter name, password, and confirm password.
3. Select *Create User.* Successful implementation can be demonstrated through a toast that displays “Account created”, and the activity should return automatically.
4. Check if any account information was stored in the accounts database.

6.2.3.2 Login

In order to utilize the features of the application, the user must login. Successful logging in will result in access to the glucose measurement interface.

1. At the login page, enter correct username and password. Select *Login*. The activity should be automatically switched to the glucose measurement interface.
2. At the login page, enter correct username and password. Select *Login*. The activity should be remain at the login page and a toast should pop up notifying the user "Incorrect username/password".
3. Successful implementation of login will be demonstrated with the proper credentials allowing access to the glucose measurement interface.

6.2.3.3 Bluetooth Connection to Hardware

The glucose measurement system's hardware must connect to the Android phone wirelessly via Bluetooth. Successful Bluetooth connection will result in the phone notifying the user with a solid-color Bluetooth sign in the phone's status bar, as well as a toast from the application.

1. Enable Bluetooth connection from the phone settings.
2. From the main activity page, click the shelf icon in the top left to open the activity options.
3. Select *Bluetooth*. Then, select *Connect*.
4. Successful implementation should result in a toast pop up saying "Connected to GOLD Meter".

6.2.3.4 Spectral Measurements

The glucose measurement, in addition to identifying skin thickness and pulse oximetry, is the core of this project. While we aim to result in the most accurate concentration measurement as possible, it is understood that this is a research-based project. Thus, concentration values may be off significantly. In this case, improvements may be made through better hardware implementation and/or statistical analysis and innovation. In order to make sure our code works properly for the data filtering, processing, and modeling, we will provide simulated data to produce expected results. Our data will be in the form of simple integer 2-D arrays. We will test each algorithm we attempt and finally decide to implement in the final product. After confirming the simulated results' accuracy are up to par, we will understand if our analytical methods are working correctly for the hardware implementation. Our goal is to result in a Spearman's correlation coefficient of at least approximately 0.70 for glucose measurements in water in order to provide a proof-of-concept that our method is viable. If the statistical methods are not resulting in accurate measurements, this means our hardware implementation must improve its detection methods. The following section describes our glucose measurement testing.

6.2.4 Mechanical

The following section describes the physical testing of the earpiece and the electrical enclosure to ensure that they both can withstand falling and banging against an object. In addition, these two mechanical structures must protect the electrical components from damage to ensure glucose detection.

6.2.4.1 Earpiece Testing

The testing procedure for the 3D printed earpiece is as follows:

Table 24: Earpiece Testing Procedures

Testing Process	**Procedure**
Step 1	Ensure the LED and the photodiode easily fit into the slots provided for them.
Step 2	Place the earpiece on the ears of some volunteers to check for size.
Step 3	Test the glucose measurement process while the LED and the photodiode are installed on the ear.
Step 4	Redesign and reprint earpiece if any of the above steps fail.

6.2.4.2 Enclosure Testing

The electrical housing for the electrical components must undergo a basic set of testing to ensure its security. The following is the testing protocol:

Table 25: Enclosure Testing Procedures

Testing Process	**Procedure**
Step 1	Ensure that the electronic components (AFE4404, breadboard, PCB, etc.) fit comfortably inside the enclosure.
Step 2	Shake the enclosure to verify that parts inside the enclosure do not move out of place to prevent damaging and incorrect results.
Step 3	Drop the enclosure on the ground to test its sturdiness.
Step 4	Test the glucose measurement process while the electronic components are placed inside the enclosure and verify accuracy of the results.

6.2.5 Glucose Testing

This section will discuss the different methods of glucose testing that will be used in order to determine reliability of product and check performance. Our original goal is to detect glucose in a water sample. Once we are able to obtain this goal then testing on a human would be the next step.

The test plan has two phases. Phase one is the current test phase, phase one involves getting the sample data that is needed to seed the database with initial concentration values and transmission spectra associated with their absorption spectra. In this phase, the other needed hardware components of the sensor unit are also being developed. This phase allows for insights in the data seeding process and the user-sensor interface to be assessed. Phase two involves the live, real data test of using the matured sensor unit to test glucose levels unlike in phase one which is being used to seed the database. If everything goes according to plan, the phase two test will start by the user placing the noninvasive glucose sensor unit on their body, potentially their earlobe or other thin and easily accessible body part. The user will then begin the testing process by pressing the app buttons to collect the various sample data needed for a glucose reading. After all data has been collected, the app should analyze the data and provide a reading.

6.2.5.1 In-vitro

Taking place in a test tube, culture dish, or elsewhere outside a living organism performs in-vitro testing. The first step will be to test the glucose solution in water. The in-vitro prototype is designed using LED at 1550nm and a photo diode. The photodiode is placed in opposite direction to the LED to get transmitted signal. The problem of stabilizing the photodiode is fixed by making sure it is in a box with no other lighting. Making the LED supply current constant also minimizes variation in the output voltage. Samples are prepared by dilution of 40mg/dl glucose in each step to make variation of glucose form 40mg/dl to 400mg/dl. For in-vitro measurement total ten readings are taken to investigate the affect of variation in glucose concentration on near infrared radiation. As the glucose concentration increases the output of photodiode decreases. Overall the transmitted signal should decrease with increase in glucose concentration due to the absorption of radiation by glucose. Figure 63 shows an illustration of the test set up.

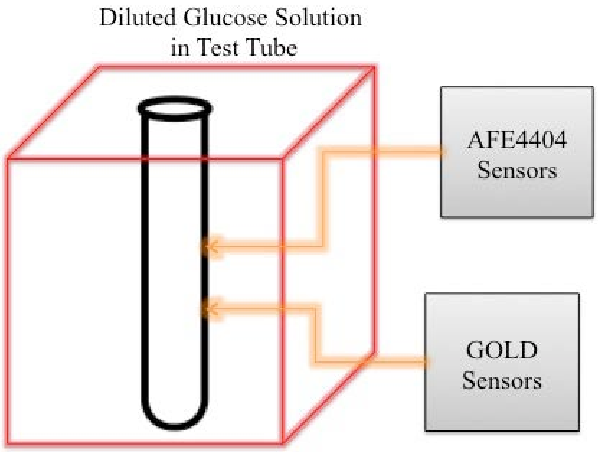


Figure 63: In-vitro test setup

6.2.5.2 In-vivo

Outer most layer of skin is the epidermis and middle layer is the dermis. Innermost layers are the subcutaneous layer that contains fat. The Dermis layer contains most of blood vessels. In order to properly measure the glucose levels then radiation must penetrate the dermis layer of the skin. The diffused photons illuminate the penetrated tissue to a maximum depth d/2 from the surface of the skin where d is the distance between source and detector. The average optimal optical path length for glucose measurement through forearm should be more than 1.3mm. A patch with the sensors will be placed on the skin. The sensor patch is designed by placing the 1550nm led with a photodiode at 4mm distance from each other. The experiment will be tested on all team members of GOLD. It is important for the person to not move and stay still during the data collection. The patch is placed on the ear lobe and forearm. This information is gathered in an excel sheet to create a correlation graph between the methods. Determining how much light is reflected back to the photodiode collects the data. Figure 63 shows an example of the skin penetration vs light.

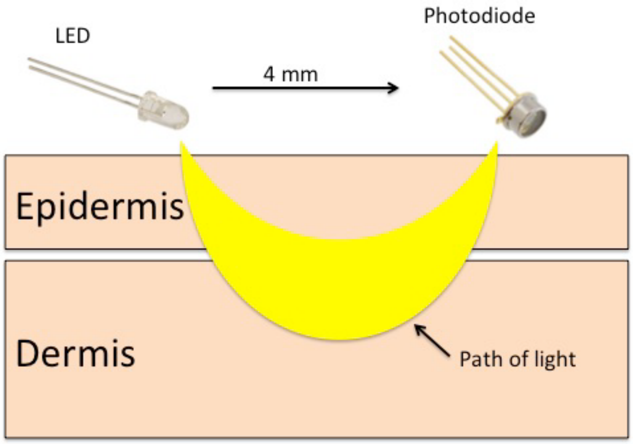


Figure 64: Skin Layer vs. Light Path

To examine the sensitivity of sensors, seven non-diabetic subjects (6 male and 1 female) between 21 and 23 years of age were considered. The test is done before eating a meal and then again a few hours later. Table 26 shows the expected values that we will obtain before meal testing. Table 27 shows the expected values for glucose concentration after a meal. The results are compared against a commercial glucose meter. Essentially, the results will show a decrease in the signals voltage amplitude observed after the meal was consumed as compared with before meal.

Table 26: Glucose Concentration before Meal

Subject	Average Voltage (V)	Glucose mg/dl
A	1.7	80
B	2.5	80
C	2.5	80
D	2.5	80
E	2.5	90
F	2.8	90
G	2.8	120

Table 27: Glucose Concentration after Meal

Subject	Average Voltage (V)	Glucose mg/dl
A	1.55	120
B	2.2	120
C	2.2	140
D	2.2	140
E	2.2	140
F	2.2	140
G	2.2	140

7.0 Project Operation

The glucose detection and measurement device can measure glucose in either a sample of a solution or can possibly determine the blood glucose measurement in a human. Although considered a medical device, the device is by no means a reliable substitute for professional medical glucose measurement devices; instead, this device should be used as a household tool that provides diagnosis at-home similar to the blood glucose monitors purchased in stores. If possible, call your local physician to setup an appointment for blood glucose measurements, as we are not liable for any injuries or mishaps.

This user manual is designed to help the user successfully measure glucose levels through a medium and to accompany the user through installation directions, troubleshooting tips and safety precautions. Below you will find all the necessary information previewed in the above statements.

7.1 Safety Precautions

Overall, the glucose measurement device is a safe device to use at home with certain applications but to reduce foreseeable safety risks, read the following safety precautions below before turning on the device or applying power to the electronic device:

1. Keep the device away from water or any other liquid to prevent the damage of the equipment.
2. Make sure that no wires are exposed to the elements and that all wires are either covered by a jacket or insulation due to electrical shock.
3. Check for the electrical connections and do not turn on the device if there are any loose or torn cables.
4. If unsure whether the device has electricity stored in the circuit, always check for power with an instrument such as a multimeter.
5. If the device must be opened, ensure the power is turned off and wait a few minutes for the circuits to discharge electricity.
6. Wear insulator gloves or remove any metallic jewelry to prevent electrical shock when opening the electrical enclosure.

7.2 General Information

The glucose measurement device, in conjunction with the pulse oximeter, proves to be a beneficial tool that provides many capabilities such as:

1. Measurement of heart rate.
2. Measurement of blood glucose levels.
3. Measurement of glucose concentration in a solution.
4. Transfer of data from glucose device to mobile phone through Bluetooth connection.
5. Display of glucose measurement according to application.

7.3 Using the Glucose Device

The device follows a routine set of instructions to reliably measure, calculate, and process the information in a timely manner. For accurate results, please follow the instructions as these instructions have been tested for maximum performance:

1. Before using the device, do not consume any food, drink, or other substance that could alter and skew the blood glucose measurements. Also, do not exercise before performing the blood glucose measurement. Please wait at least 30 minutes for better results as the pulse oximetry calculation is vital in the entire process.
2. Read the safety precautions above to minimize any risk or safety issues associated with use of this product.
3. To turn on the device, flip the switch to the ON position located on the electrical enclosure for the electronics.
4. If not turned on, turn on the phone that will be connected to the glucose measurement device.
5. Go to the Settings application on your phone and turn on your Bluetooth connectivity. Wait a few moments to enable the discoverable option.
6. Once Bluetooth is on, connect your phone to the appropriate Bluetooth device and ensure that the connection has been established.
7. If measuring the blood glucose concentration through skin, apply the earpiece attachment and wear it with the LED and photodiode installed for best results.
8. If measuring glucose concentration in another solution, remove the earpiece and simply point the LED and photodiode at the solution. For better results, place the solution in an opaque test tube or beaker.
9. Press the Test button on the glucose measurement device to commence.
10. Place the AFE4404 on the skin for blood glucose measurement. The AFE4404 pulse oximeter will measure heart rate and blood oxygen levels primarily.
11. If wearing the earpiece, ensure the earpiece is worn snug and patiently wait for the device to measure glucose concentration in the blood through the ear. Make sure no ambient light can enter the photodiode to prevent skewed results.
12. After notification that the measurement process is complete, view the results on your phone.
13. To repeat the measurement process, jump back up to step 7 for instructions for the two different applications.
14. To turn the device OFF, disconnect your phone from the glucose device and flip the switch on the electrical enclosure to OFF.
15. Place device in a dry area at room temperature.

7.4 Troubleshooting Tips

If you require assistance in using the glucose device, the following tips can be utilized in diagnosing the issue and solving the problem:

Problem	**Solution**
1. The device will not turn on.	• Change the batteries. • Check for any loose wires and connections. • Measure the voltage across the battery and ensure that the battery is running at the rated voltage.
2. The device is not giving a reasonable output.	• Make sure the LED and the photodiode are properly installed on the earpiece. • Make sure the LED and the photodiode are properly aligned. They must either face each other or sit next to each when measuring in a test tube. • Measure voltage across the photodiode; if any higher than the rated voltage, turn power off for 2 minutes and turn on the device.
3. The device turns off during glucose measurement process.	• Check for any loose wires and connections. • Measure the voltage across the battery and ensure that the battery is running at the rated voltage.
4. The phone will not connect to the device through Bluetooth.	• Turn the device off, wait 15 seconds, and turn the device back on. • Turn off the Bluetooth feature on your phone and turn it back on. • Reattempt connectivity; if it fails, contact a technician.
5. The phone application does not display the measurements.	• Disconnect and turn off Bluetooth on the phone. • Turn the device on and off for reset. • Reconnect the Bluetooth connection between the phone and the device.

8.0 Administrative Content

This section will discuss two main aspects that helped drive the project immensely. Dividing the project into two main sections set up the milestones. The first section was research and prototyping. In this area we concentrated on research what parts work best for our project and determining what would give us the best results. Once we researched what would work best then we purchased parts and started testing them to see if it gave us the results we wanted. The next section was for semester two. This section consisted of acquiring the final parts we would use and integrating them into a fully functional system. This also required us to do a lot of testing to ensure our device was optimized.

8.1 Milestones

As previously mentioned, the milestones were broken into two major sections. The first section was senior design 1 where research and testing was done. The second section is senior design 2 where most of the integrating was done. Most of the dates were estimated and used as a target. It was important for us to keep a rough idea of when things would get done so we could plan ahead. Table 20 describes each step taken.

8.2 Budget Analysis

Since our project was provided sponsorship from Texas Instruments we were able to obtain $1,000. Although we received spending money from Texas Instruments it was still important for us to keep a good budget so that this project could be easily marketable to end users. By Texas Instruments providing us with the AFE4404 evaluation module we were able to save the cost of $200. In an ideal situation we would recreate this evaluation module and lower the cost of our project. For prototyping purposes it was not necessary for us to reproduce this evaluation module. Table 29 shows a list with purchased parts that were used for project.

Table 29: Cost of Project

Item	Qty.	Price with Qty.
ADC Converter	1	$39.15
Photodiode	1	$31.43
DAC Converter	1	$14.50
MCU	1	$10.69
Amplifier	1	$2.00
Battery	2	$7.98
Misc.	1	$100.00
Packaging	1	$30.00
Bluetooth Transceiver	1	$9.99
PCB	1	$50.00
LED 1550nm	1	$27.58
AFE4404 Eval Module	1	donated from TI
	Total	$323.32
	TI Sponsorship	$1,000.00
	Net total	**-$676.68**

9.0 Project Summary and Conclusion

As we designed and developed the noninvasive glucose detection device, many obstacles crossed our paths but as a group of seven members, we have been able to overcome most of them. This noninvasive glucose detection project has been worked on for many decades and after researching for previous experiments related to our project, there’s not too much information on this challenging discovery.

With the information we did obtain, we created a design with the idea of using a near infrared LED and a photodiode to measure the amount of the light that had not been absorbed the glucose molecules, which had been previously discovered before. The guidance provided by Texas Instruments pushed us to continue working on this project as many people in the world would prefer to possess a noninvasive glucose measurement device instead of a glucose monitor that requires them to prick a finger and purchase test strips for each measurement. This device could assist diabetics who require constant monitoring daily; in addition, they can save money from the test strips they no longer need to purchase. Lastly, diabetics would no longer need to worry about infecting the prick on the finger for testing purposes.

The reason we chose to do this project is not only to challenge ourselves to find a better solution for glucose measurement but to also prove that glucose measurement can be attained through near infrared spectroscopy. Our objective was to incorporate the glucose measurement into a portable device that can be easily carried around. The different components required for this project were strategically chosen for cost savings and efficiency purposes.

The AFE4404 module can be manipulated to act as a pulse oximeter, which helps us figure out how much oxygen content is flowing through the blood within the area that is measured. In addition to determining the amount of oxygen in the blood, the heart rate will also be determined to correlate the amount of glucose in the blood with the frequency of the heart pumping blood.

All in all, the entire project has the potential to be a desired product by millions of people due to the fact that many suffer from this common disease. With our finished product, we hope to achieve the goals that we have set out and we are confident that this product will produce meaningful results.

10.0 Appendices

This section is used to show the bibliography containing all the material that was referenced throughout this paper. Also a section for permissions follows to show permission for each outsource image used.

10.1 Bibliography

[1] National Center for Chronic Disease Prevention, "National Diabetes Statistics Report," National Center for Chronic Disease Prevention, Atlanta, 2014.

[2] M. M. R. b. P. F. B. I. M. M. By Krisha McCoy, " "The History of Diabetes."," *EverydayHealth.com.*

[3] Medtronic, "Medtronic," 01 November 2008. [Online]. Available: www.medtronic,com/us-en/index.html. [Accessed 12 November 2016].

[4] J. K. D. M. &. A. R. C. (. Lipkowski, "Bioelectrochemistry : Fundamentals, Applications and Recent Developments.," *Weinheim: Wiley-VCH..*

[5] Diabetes Sentry, "Diabetes Sentry," [Online]. Available: diabetessentry.com. [Accessed 1 November 2016].

[6] John Wiley & Sons, 2013., "[5] Advances in Electrochemical Sciences and Engineering : Bioelectrochemistry : Fundamentals, Applications and Recent Developments.," *John Wiley & Sons, 2013.,* 2013.

[7] Pittsburgh Press, November 5, 1981, p. A-6, ""Portable Meter To Aid Diabetics"," *Pittsburgh Press,* p. p A.6, November 5, 1981.

[8] Statista, "Market share of the global diabetes care devices manufacturers 2011 | statistic," *Statista,* 2016.

[9] CostHelper, ""How Much Does a Glucose Meter Cost? "," CostHelper, 13 Oct. 2016.. [Online].

[10] @Diabetes4cast., " Diabetes Forecast. N.p., n.d. Web.," @Diabetes4cast, 13 Oct. 2016.. [Online]. [Accessed 2016].

[11] n. W. 1. O. 2. Diabetes Care. N.p., ""The Impact of Treatment Noncompliance on Mortality in People With Type 2 Diabetes | Diabetes Care." The Impact of Treatment Noncompliance on Mortality in People With Type 2 Diabetes," Diabetes Care. [Online].

[12] C. G. M. T. N. I. o. D. a. D. a. K. Diseases.., The National Institute of Diabetes and Digestive and Kidney Diseases., December 2008. [Online]. [Accessed Retrieved 21 February 2016.].

[13] ". L. A. L. R. 2. F. 2016., ""FreeStyle Libre"," Abbott Laboratories. [Online]. [Accessed Retrieved 21 February 2016].

[14] D. S. BSN, ""Is Continuous Glucose Monitoring Worth It? - Diabetes Self-Management."," Diabetes Self-Management." Diabetes SelfManagement. [Online]. [Accessed 13 Oct. 2016.].

[15] G. S. J. L. P. K. F. S. P. R. Tamada JA, ""Noninvasive glucose monitoring: comprehensive clinical results. Cygnus Research Team"," JAMA. 282, (November 1999). [Online]. Available: JAMA. 282 (19): 1839–44. doi:10.1001/jama.282.19.1839. PMID 10573275.

[16] M. N.p., ""GlucoTrack Non-Invasive Ear Lobe Glucometer to Go to Clinical Trial in U.S. |."," 17 Aug 2015. [Online]. [Accessed Web. 13 Oct. 2016.].

[17] M. S. M. Wens, Design and Implementation of Fully-Integrated Inductive, Springer Science+Business Media.

[18] "Battery University," Cadex Electronics Inc, [Online]. Available: http://batteryuniversity.com/learn/article/lithium_based_batteries. [Accessed 16 October 2016].

[19] J. Turley, ""The Two Percent Solution"," 2002. [Online].

[20] T. Instruments, "MSP430Fxxx Datasheet," Texas Instruments, [Online]. Available: http://www.ti.com/lit/ug/slau208p/slau208p.pdf. [Accessed Oct-Nov 2016].

[21] BForsite, "Android OS reaches 86% market share world wide," 22 8 2016. [Online]. Available: http://www.bforsite.com/android-os-reaches-86-market-share-world-wide/.

[22] Google, "Dashboards | Android Developers," Google, 5 September 2016. [Online]. Available: https://developer.android.com/about/dashboards/index.html. [Accessed 14 October 2016].

[23] Apple, "App Store - Support - Apple Developer," Apple, 7 October 2016. [Online]. Available: https://developer.apple.com/support/app-store/. [Accessed 14 October 2016].

[24] A. Sinicki, "Developing for Android vs developing for iOS – in 5 rounds | Android Authority," Android Authority, 9 June 2016. [Online]. Available: http://www.androidauthority.com/developing-for-android-vs-ios-697304/. [Accessed 14 October 2016].

[25] Energia, "Energia," [Online]. Available: http://energia.nu/. [Accessed 26 November 2016].

[26] S. Vender, "Short Introduction to iOS for Java Developers: Objective-C - codecentric AG Blog : codecentric AG Blog," 19 April 2011. [Online]. Available: https://blog.codecentric.de/en/2011/04/short-introduction-to-ios-for-java-developers-objective-c/. [Accessed 30 November 2016].

[27] Apple, "The Swift Programming Language (Swift 3): About Swift," Apple, 13 September 2016. [Online]. Available: https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/index.html. [Accessed 14 October 2016].

[28] T. Gühring, "Swift 2 - Introduction and Comparison - NovaTec Blog," 6 October 2015. [Online]. Available: http://blog.novatec-gmbh.de/swift-2-introduction-comparison/. [Accessed 30 November 2016].

[29] CCM, "What is WiFi and How Does it Work?," CCM, October 2016. [Online]. Available: http://ccm.net/faq/298-what-is-wifi-and-how-does-it-work. [Accessed 14 October 2016].

[30] Bluetooth, "Top Things to Know About Bluetooth Technology | Bluetooth Technology Website," Bluetooth, 2016. [Online]. Available: https://www.Bluetooth.com/what-is-Bluetooth-technology/Bluetooth-technology-basics. [Accessed 14 October 2016].

[31] APC, "Inside NFC: how near field communication works - APC," APC, 17 August 2011. [Online]. Available: http://apcmag.com/inside-nfc-how-near-field-communication-works.htm/. [Accessed 14 October 2016].

[32] ThorLabs, "ThorLabs," [Online]. Available: Thorlabs.com.

[33] J. R. Biard, "The life and times of the LED - a 100 year history," *Optoelectronics Research Centre, University of Southhampton,* 1966.

[34] B. H. Hinsberg, "Journal of Diabetes Science and Technology," [Online]. Available: NCBI.nlm.nih.gov.

[35] V. A. Saptari, "A spectroscopic System for Near Infrared Glucose Measurment," Massachusetts Institute of Technology, Massachusetts, 2004.

[36] "MSP430 general purpose input output (GPIO)," 6th July 2014. [Online]. Available: http://www.argenox.com/library/MSP430/general-purpose-input-output-gpio-chapter-5/. [Accessed 2 December 2016].

[37] K. D. –. B.Tech, "Unit Testing the MSP430 within Desktop Environment," Clarius Mobile Health. [Online].

[38] A. Biruk, "Bluetooth Data Transmission from MSP430 to Smartphone," [Online]. [Accessed 1 December 2016].

[39] K. Mitzner, "Complete PCB design using ORCAD capture and PCB editor," in *Complete PCB design using ORCAD capture and PCB editor* .

[40] Google, "Fragments | Android Developers," Google, 2016. [Online]. Available: https://developer.android.com/guide/components/fragments.html. [Accessed 23 October 2016].

[41] Google, "Bluetooth | Android Developers," Google, 2016. [Online]. Available: https://developer.android.com/guide/topics/connectivity/Bluetooth.html. [Accessed 23 October 2016].

[42] Programiz, "C Programming Unions (With Examples)," [Online]. Available: https://www.programiz.com/c-programming/c-unions. [Accessed 26 November 2016].

[43] G. G. Jim Clark, "The Beer-Lambert Law - Chemistry LibreTexts," 17 September 2016. [Online]. Available: http://chem.libretexts.org/Core/Physical_and_Theoretical_Chemistry/Spectroscopy/Electronic_Spectroscopy/Electronic_Spectroscopy_Basics/The_Beer-Lambert_Law. [Accessed 1 December 2016].

[44] Zetetic, "SQLCipher - Zetetic," 2016. [Online]. Available: https://www.zetetic.net/sqlcipher/design/. [Accessed 26 November 2016].

[45] M. Rouse, "What is initialization vector (IV)? - Definition from WhatIs.com," March 2011. [Online]. Available: http://whatis.techtarget.com/definition/initialization-vector-IV. [Accessed 26 November 2016].

[46] N. T, "Standard 4- and 6-Layer PCB Stackups," [Online]. Available: http://www.bitweenie.com/listings/standard-4-and-6-layer-pcb-stackups.

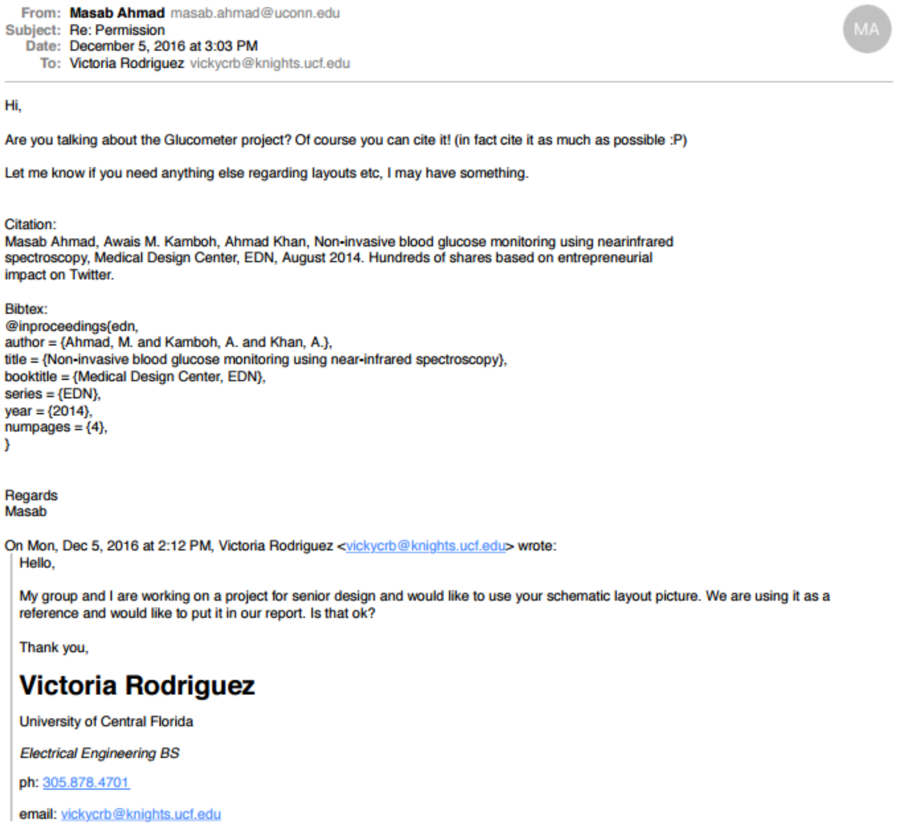
[47] "Essential EMC design rules to avoid the 7 most common EMC failures," EMC Design Tips, [Online]. Available: http://www.emcfastpass.com/avoid-the-most-common-failures-with-these-7-essential-emc-design-rules/.

[48] XueMing, "Reflow oven soldering without stencils," [Online]. Available: https://zavax.wordpress.com/2013/06/30/reflow-oven-soldering-without-stencils/.

[49] Analog Devices, "ad7714 datasheet," [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD7714.pdf.

[50] Adil, "Intellectual," [Online]. Available: https://e2e.ti.com/support/development_tools/code_composer_studio/f/81/p/232571/816151.

10.2 Permissions



From: **Masab Ahmad** masab.ahmad@uconn.edu
Subject: Re: Permission
Date: December 5, 2016 at 3:03 PM
To: Victoria Rodriguez vickycrb@knights.ucf.edu

MA

Hi,

Are you talking about the Glucometer project? Of course you can cite it! (in fact cite it as much as possible :P)

Let me know if you need anything else regarding layouts etc, I may have something.

Citation:
Masab Ahmad, Awais M. Kamboh, Ahmad Khan, Non-invasive blood glucose monitoring using nearinfrared spectroscopy, Medical Design Center, EDN, August 2014. Hundreds of shares based on entrepreneurial impact on Twitter.

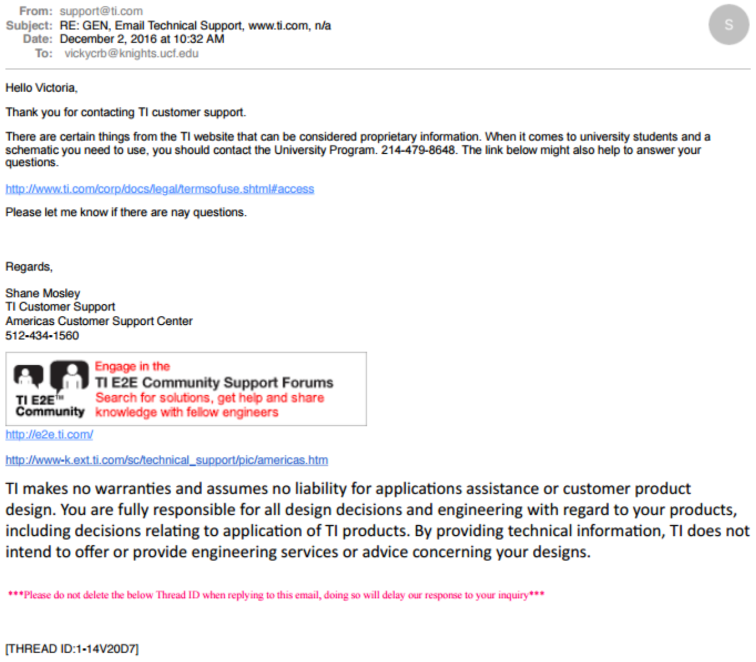
Bibtex:

```
@inproceedings{edn,
author = {Ahmad, M. and Kamboh, A. and Khan, A.},
title = {Non-invasive blood glucose monitoring using near-infrared spectroscopy},
booktitle = {Medical Design Center, EDN},
series = {EDN},
year = {2014},
numpages = {4},
}
```

Regards
Masab

On Mon, Dec 5, 2016 at 2:12 PM, Victoria Rodriguez <vickycrb@knights.ucf.edu> wrote:

> Hello,
>
> My group and I are working on a project for senior design and would like to use your schematic layout picture. We are using it as a reference and would like to put it in our report. Is that ok?
>
> Thank you,
>
> **Victoria Rodriguez**
>
> University of Central Florida
>
> *Electrical Engineering BS*
>
> ph: 305.878.4701
>
> email: vickycrb@knights.ucf.edu



From: support@ti.com
Subject: RE: GEN, Email Technical Support, www.ti.com, n/a
Date: December 2, 2016 at 10:32 AM
To: vickycrb@knights.ucf.edu

S

Hello Victoria,

Thank you for contacting TI customer support.

There are certain things from the TI website that can be considered proprietary information. When it comes to university students and a schematic you need to use, you should contact the University Program. 214-479-8648. The link below might also help to answer your questions.

http://www.ti.com/corp/docs/legal/termsofuse.shtml#access

Please let me know if there are nay questions.

Regards,

Shane Mosley
TI Customer Support
Americas Customer Support Center
512-434-1560

TI E2E™ Community
Engage in the
TI E2E Community Support Forums
Search for solutions, get help and share knowledge with fellow engineers

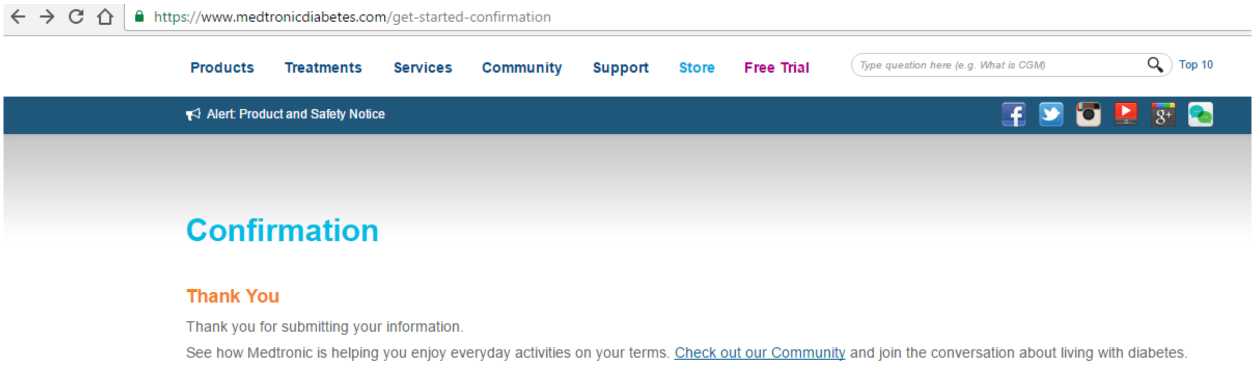
http://e2e.ti.com/

http://www-k.ext.ti.com/sc/technical_support/pic/americas.htm

TI makes no warranties and assumes no liability for applications assistance or customer product design. You are fully responsible for all design decisions and engineering with regard to your products, including decisions relating to application of TI products. By providing technical information, TI does not intend to offer or provide engineering services or advice concerning your designs.

Please do not delete the below Thread ID when replying to this email, doing so will delay our response to your inquiry

[THREAD ID:1-14V20D7]

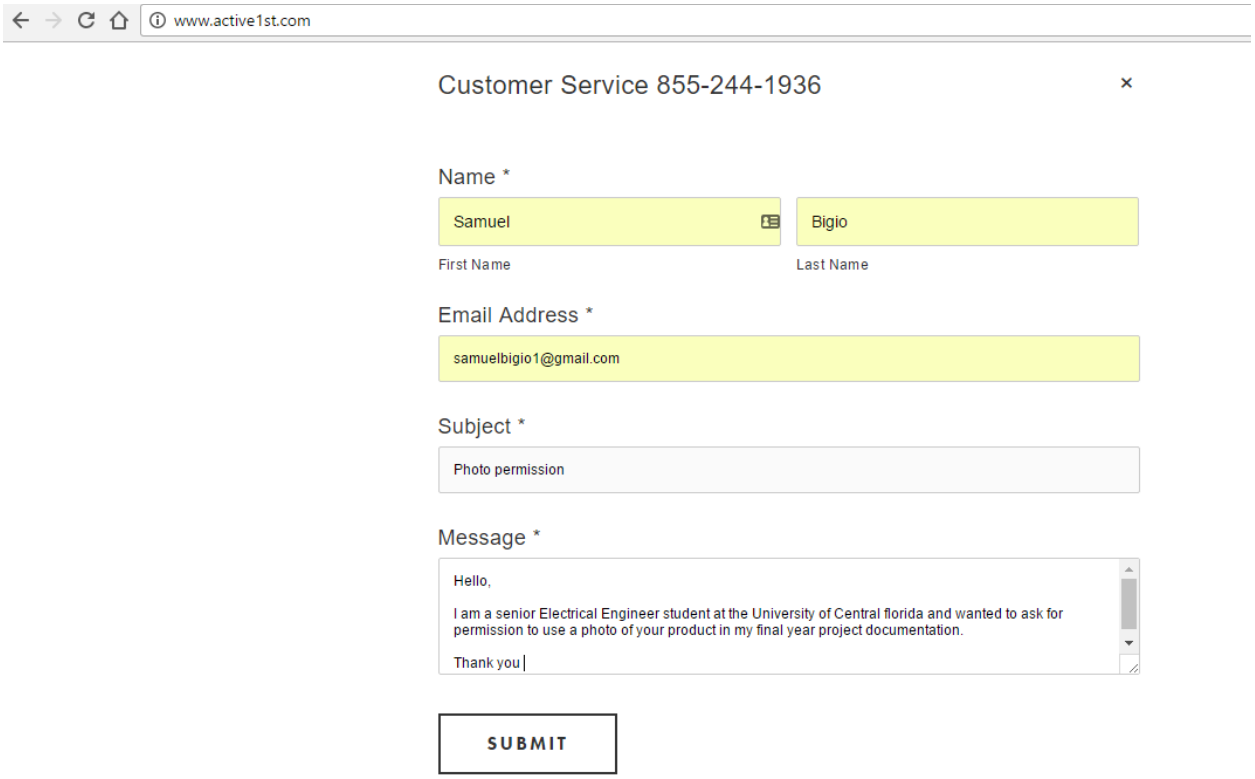

https://www.medtronicdiabetes.com/get-started-confirmation
Products
Treatments
Services
Community
Support
Store
Free Trial
Type question here (e.g. What is CGM)
Top 10
Alert: Product and Safety Notice

Confirmation

Thank You

Thank you for submitting your information.

See how Medtronic is helping you enjoy everyday activities on your terms. Check out our Community and join the conversation about living with diabetes.



Customer Service 855-244-1936

Name *

Samuel | Bigio

First Name | Last Name

Email Address *

samuelbigio1@gmail.com

Subject *

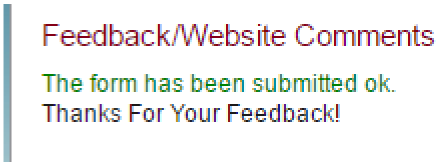
Photo permission

Message *

Hello,

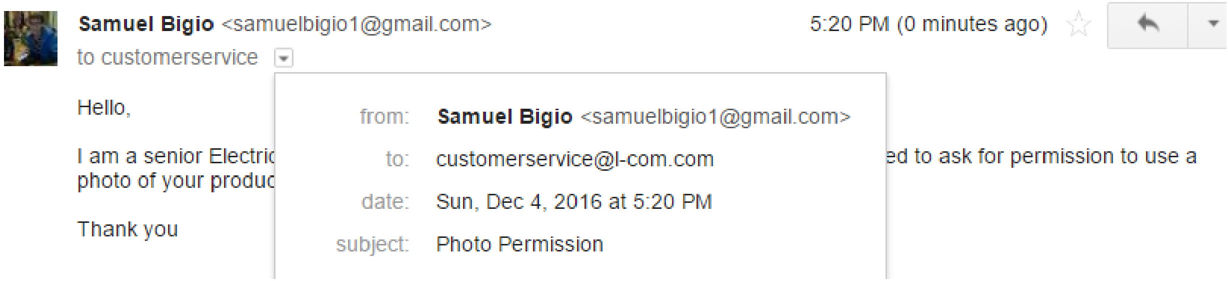
I am a senior Electrical Engineer student at the University of Central florida and wanted to ask for permission to use a photo of your product in my final year project documentation.

Thank you



Feedback/Website Comments

The form has been submitted ok.
Thanks For Your Feedback!



Samuel Bigio <samuelbigio1@gmail.com> 5:20 PM (0 minutes ago)

to customerservice

from: **Samuel Bigio** <samuelbigio1@gmail.com>

to: customerservice@l-com.com

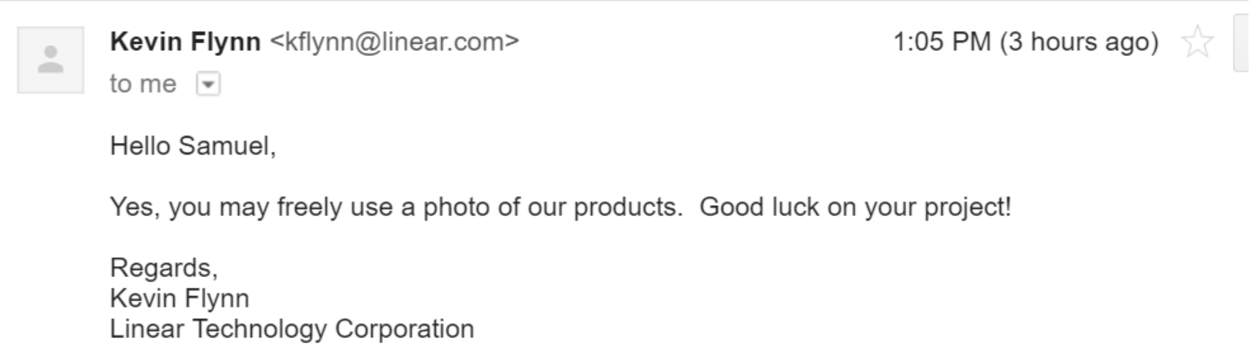
date: Sun, Dec 4, 2016 at 5:20 PM

subject: Photo Permission

Hello,

I am a senior Electric ed to ask for permission to use a
photo of your produc

Thank you



Kevin Flynn <kflynn@linear.com>

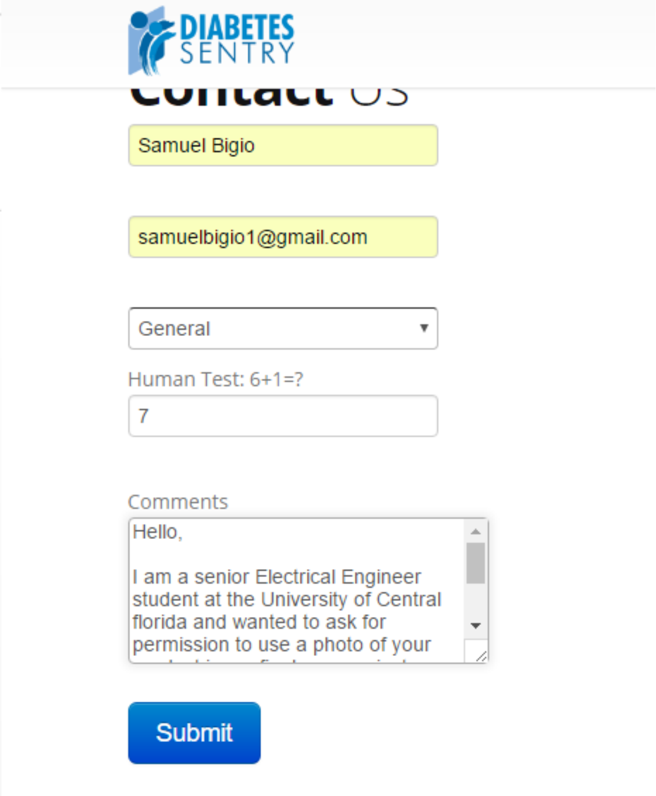
1:05 PM (3 hours ago)

to me

Hello Samuel,

Yes, you may freely use a photo of our products. Good luck on your project!

Regards,
Kevin Flynn
Linear Technology Corporation



Contact Us

Samuel Bigio

samuelbigio1@gmail.com

General ▾

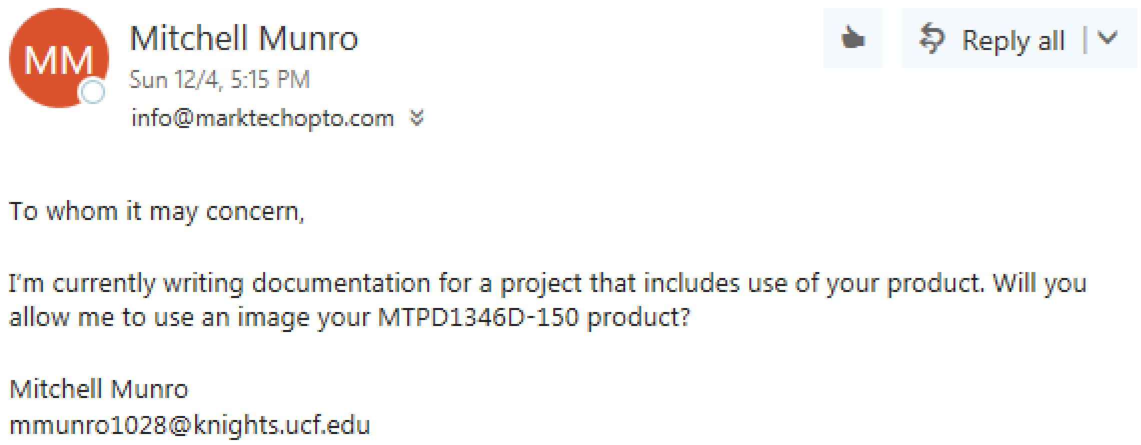
Human Test: 6+1=?

7

Comments

Hello,

I am a senior Electrical Engineer student at the University of Central florida and wanted to ask for permission to use a photo of your

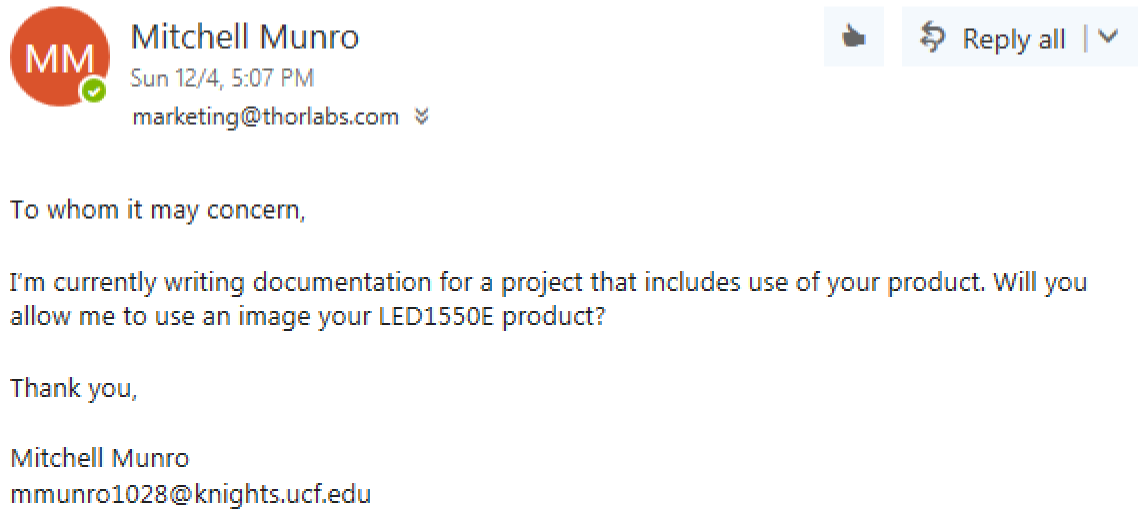


Mitchell Munro
Sun 12/4, 5:15 PM
info@marktechopto.com

To whom it may concern,

I'm currently writing documentation for a project that includes use of your product. Will you allow me to use an image your MTPD1346D-150 product?

Mitchell Munro
mmunro1028@knights.ucf.edu



Mitchell Munro
Sun 12/4, 5:07 PM
marketing@thorlabs.com

To whom it may concern,

I'm currently writing documentation for a project that includes use of your product. Will you allow me to use an image your LED1550E product?

Thank you,

Mitchell Munro
mmunro1028@knights.ucf.edu