Diabetic Breathalyzer

New and Noninvasive

Department of Electrical Engineering and Computer Science
University of Central Florida

Group 13

Jonathan Brown	Electrical Engineer
Christine Sleppy	Electrical Engineer
Noah Spenser	Electrical Engineer
Edert Geffrard	Electrical Engineer

Table of Contents

1.0 Executive Summary 1
2.0 Project Description 2
2.1 Project Motivation and Goals 2
2.2 Project Objectives 4
2.3 Requirements and Specifications 5
3.0 Research and Investigations 10
3.1 Research Related to Project Definition 10
3.1.1 Research in Breath Analysis 10
3.1.2 Advantages and Disadvantages of Breath Analysis 12
3.2 Project Based Research 13
3.2.1 University of Pittsburgh 13
3.2.2 Western New England University 14
3.2.3 Stony Brook University 15
3.3 Existing Similar Projects and Comparison 16
3.3.1 Breathometer 16
3.3.2 The Floome 17
3.3.3 BAC Breathalyzer 18
3.4 Parts and Components Research 20
3.4.1 Temperature and Humidity Sensor 20
3.4.2 High Concentration VOC Sensor 21
3.4.3 Low Concentration VOC Sensor 21
3.4.4 Wireless Communication 22
3.4.5 Microcontroller Unit 22
3.4.6 Power 23
3.4.7 PCB Vendor Info 24
4.0 Standards and Constraints 28
4.1 Project Related Standards 28
4.1.1 FDA standards 28
4.1.2 CE standard 29
4.1.3 FCC standard 31
4.1.4 IEEE standard 31
4.2 Realistic Design Constraints 34
4.2.1 Parts, Component and Testing Constraints 34
4.2.2 Economic and Time constraints 36
4.2.3 Environmental, Social, and Political Constraints 37
4.2.4 Ethical, Health, and Safety Constraints 38
4.2.5 Technical Constraints 39
4.2.6 Manufacturability and Sustainability constraints 41
5.0 Project Hardware and Software Design 42
5.1 Initial Hardware Design 43
5.1.1 Power 43
5.1.2 Control Unit 46
5.1.3 Communication 48

5.1.4 Sensors ... 49
5.1.5 Physical Design ... 52
5.2 Software Design ... 55
5.2.1 Device/Input Code ... 57
5.2.2 Phone Application Code ... 59
5.2.3 Device Communication ... 62
5.2.4 Sensor Calibration Coding ... 63
6.0 Prototype Construction ... 67
6.1 Master Parts List ... 67
6.2 BOM and parts acquisition ... 68
6.3 PCB Specs ... 81
6.3.1 PCB Design ... 85
6.3.2 PCB Vendor Info ... 86
6.4 Code Layout ... 87
6.4.1 Phone Application Framework ... 87
6.4.2 Conversion Code Layout ... 90
6.4.3 Device Code Layout ... 91
7.0 Project Prototype Testing ... 94
7.1 Hardware Prototype Environment ... 94
7.1.1 Breadboard Design ... 95
7.1.2 Testing Procedure ... 99
7.1.3 PCB Testing ... 101
7.2 Software Prototype Environment ... 102
7.2.1 IDE Used ... 102
7.2.2 ISP Programmer Setup ... 103
7.2.3 Testing Procedure ... 103
8.0 Administrative Content ... 106
8.1 Initial Budget and Finances Discussion ... 106
8.2 Initial Milestone Discussion ... 110
9.0 Project Summary and Conclusions ... 120
9.1 Project Feasibility ... 120
9.2 Project Operation ... 121
9.3 Project Conclusions ... 121
10.0 Experimental Results ... 123
10.1 Sensor Results ... 123
10.2 Hardware Results ... 124
10.3 Software Results ... 125
10.4 Human Results ... 126
10.5 Experimental Conclusions ... 128
Appendices ... 129
A: Software Use ... 129
B: References and Citations ... 129
C: Permissions ... 133

List of Tables and Figures:

Table 1: Relationship between project objectives, engineering requirements, and specifications 7
Table 2: The relationship between breath and disease has been is a well-known part of diagnosis 11
Table 3: Advantages versus disadvantages of breath analysis 13
Table 4: Comparison of the possible PCB vendors and their corresponding features 27
Table 5 : Specified power ratings from datasheets. (37)(38)(39)(40)(41)(42)(43) 44
Table 6: Comparison of researched MCUs. (41) 47
Table 7: Comparison of researched VOC sensors (37)(38)(44) 50
Table 8 : Master Parts List. 68
Table 9: Purchasing history for prototyping/final device components. 69
Table 10: Plot points for first equation used for a resistance ratio greater than or equal to 0.6 (37) 72
Table 11: Regression analyis results for the first equation of the TGS correlaiton. 73
Table 12: Plot points for the TGS for resistance values greater than 0.6 (37) 73
Table 13: Regression analysis results for the second equation for the TGS 74
Table 14: Plot points for the TGS dependency on temperature and humidity. (37) 74
Table 15: Regression analysis for temperature and humidity dependence for TGS 75
Table 16: Plot points for ratios above 0.45 for WSP (38) 75
Table 17: Regression analysis 76
Table 18: Plot points for ratios below 0.45 for WSP (38) 76
Table 19: Regression analysis for ratios below 0.45 76
Table 20: Plot points of ratio to humidity and temperature dependence for WSP (38) 77
Table 21: Regression analysis for temperature and humidity dependence for WSP 77
Table 22: Information regarding chosen PCB manufacturer Advanced Circuits. (15) 87
Table 23: Approximation of expenses and budget prediction 109
Table 24: Project milestone for summer semester 116
Table 25: Project Milestone for fall semester 119
Table 26 : Summary of results taken from healthy users with normal eating habits and after one week of following a ketogenic diet. 126
Table 27 : Summary of the results found from two diabetic users and their changes that created a rise in both ketone and acetone concentration. 128

Figure 1: The U.S. Post Office recognized the 1906 Act as a landmark of the 20th centry when it released this stamp. 29
Figure 2: Countires that require the CE marking 30
Figure 3: An example of a consequence of flicker 32
Figure 4: Relationship between brightness of LED and temperature 34
Figure 5: Block diagram of necessary hardware and responsible team member. 42
Figure 6 : Typical circuit configuration of LM2596 regulator. (42) 44
Figure 7: Original Multisim simulation showing overall current draw of device. 45
Figure 8: Drawing of MCU package we will use. (41) 48
Figure 9: Original drawing of circuitry representation of VOC sensors. 50
Figure 10: Eagle schematic configuration of transistor as a switch. 51
Figure 11: Original drawing showing mechanism of a ball spring check valve. 53
Figure 12: Side view, original SolidWorks drawing of outside of device. 54
Figure 13: Front and side view, original SolidWorks drawing. 54
Figure 14: Top and perspective view, original SolidWorks drawing. 55
Figure 15: Initial flowchart to show software processes. 55
Figure 16: Smartphone examples that will be used to connect to the device. 61
Figure 17: Plot of TGS822 resistance response to gas concentration (37) 71
Figure 18: Plot of TGS822 resistance response to humidity and temperature (37) 71
Figure 19: Plot of WSP2110 resistance response to gas concentration. (38) 71
Figure 20: Plot of WSP2110 resistance response to humidity and temperature. (38) 72
Figure 21: Original picture of purchased HC-05 wireless module. 78
Figure 22: Original picture of purchased sensor modules. 79
Figure 23: Original picture of programmer and chip used. 80
Figure 24: Entire original eagle schematic. 81
Figure 25: Original Eagle schematic of necessary connections for the operation of sensors. 82
Figure 26: Original Eagle schematic of management of the ATMega328p. (41) 83
Figure 27: Original Eagle shematic of power control and regulation setup. 84
Figure 28: Original Eagle schematic of standard 2x3 ISP configuration and the needed pins for the HC-05. (39) 84
Figure 29: Picure showing typical layering of a 2 layer PCB. (36) 85
Figure 30: Original PCB Eagle deisgn. 86
Figure 31: Initial Flowchart for Application Framework 89
Figure 32: Flowchart for Conversion Code 91
Figure 33: Flowchart for Device Code 93
Figure 34: Original PEBBLE breadboard design needed to program MCU. 95
Figure 35: Original drawing of dedicated programmer board. 96
Figure 36: Original PEBBLE breadboard design to run and test all three sensors. 97
Figure 37: Original breadboard design to run HC-05 wireless communication. 98

Figure 38: Original picture of our prototyping setup including a variable DC power supply. 99
Figure 39 : Prototype testing set up with real time response displaying resistance results (blue and orange) and acetone breath concentration values in parts per million (red and green)...124
Figure 40 : Finalized device set up and application output used for continued testing and to determine results. 125

1.0 Executive Summary

Research has shown that there is a firm relationship between the acetone concentration in human breath and ketone levels found in the human body. The Diabetic Breathalyzer was designed to test this research with its noninvasive method for monitoring blood glucose level. This project set out to confirm the relationship between acetone and ketones and the relationship between ketones and high blood glucose level. Our team obtained a working prototype that takes in breath and produce a diabetic friendly output allowing an alternate solution for monitoring blood glucose levels in diabetics throughout the day.

The Diabetic Breathalyzer has many features that are attractive to those who suffer from diabetes. The final device is small and compact for easy portability with room for further shrinking, and allows for a monitoring device to be on their person at all times. The low cost, low power features of the final product allow for reasonable purchasing and long lasting battery life. It sends data directly to the user's smartphone via wireless communication. This provides the diabetic with an easy way to see trends in their data and helps promote them to act accordingly and live more healthily. The device is noninvasive, which is a much better option compared to the current glucose monitoring system forcing them to prick their finger for blood every time they need to check their numbers. The constant finger pricking causes large amounts of discomfort and pain for the user, which results in the lack of proper management. Our non-invasive feature helps encourage the user to be aware of their blood glucose levels more often.

This is a project of electronics including hardware, software, and the communication between them. The device is powered by a rechargeable battery that lasts a good amount of time currently and with some power usage optimization should last a full day before needing to be charged. We built a small, hand held device that the user can blow into. Inside there is an air chamber where sensors detect the amount of acetone in the breath. The sensors' outputs are sent to the code where computations and conversions are performed. The code also sends the data to the user's smartphone application through wireless communication. The user has effortless access to constant everyday tracking and logging of their acetone levels and could easily track their blood glucose levels in the future using a later iteration of the device.

The testing procedure for accuracy and sensor calibration was carried out using several methods. We had different types of humans (healthy human, diabetic human, healthy human on ketogenic diet, fasting healthy human) blow into our device for comparison and calibration. With access to the known traditional urine strips that measure ketones and a blood glucose meter that measures blood glucose levels, we can made the appropriate conclusions that led us to the current functionality and future planning for the device.

2.0 Project Description

In this section, we briefly go over our idea for the project and describe the main goal we set out to achieve. This involves determination of our project objectives and the requirements and specifications to carry them out.

2.1 Project Motivation and Goals

Diabetes is a disease that causes elevated glucose levels due to the inability of the body to produce a hormone called insulin. This disease affects millions of people every day who have to constantly monitor themselves and their food intake. Insulin regulates the amount of glucose in the blood. When diagnosed with Type 1 Diabetes, this process is not so simple. Without the pancreas automatically working to supply insulin throughout the body, diabetics have to take care of this burdensome process themselves. The first step in properly carrying out this task is for patients to know their blood glucose level because there is a very strategic method to knowing how much insulin one needs to provide for oneself. Everyone has their own healthcare plan that works for their body, but the most important part of that plan is proper management.

Traditionally, monitoring and measuring ones blood glucose level requires a prick of the finger. In fact, this is currently the only way to accurately determine the value. Pricking the finger draws a drop of blood that is placed on a test strip and read by a meter. This process must be done before every meal or snack as well as any other time when the diabetic is feeling like they're experiencing symptoms of high or low blood sugar levels, both of which can be very dangerous to the body. In addition, keeping a log of results is vital to be sure that the body is agreeing with its current healthcare plan. This is very demanding, invasive, and painful resulting in the lack of proper monitoring by the diabetic. It is unhealthy for them to be unaware of their blood glucose levels throughout the day. Our project proposes a possible solution to these daily difficulties. Our Diabetic Breathalyzer project set out to create a device that can measure one's blood glucose level by breath. This method would be noninvasive and less painful to do multiple times throughout the day.

The idea can be implemented by the relationship between acetone levels in the breath and ketone levels in the body. Ketones are a class of chemical produced in the body when there is a shortage of insulin causing the body to use fat for energy instead of glucose. The acetone levels in one's breath can be used as an indicator for presence of these ketones. Diabetics with very high levels of glucose see these ketones in their urine. However, they will usually check their blood glucose number by pricking their finger and then doing a urine test to check for ketones. Our breathalyzer would skip this extra step and create a direct relationship between them. This idea of a diabetic breathalyzer would be able to determine the levels of acetone in the breath and directly connect it with the presence of ketones in the body, which leads to the indication of high blood glucose level without a single prick of the finger.

There has been research conducted in recent years to try and find a proper correlation between levels of acetone in the breath and the level of ketones in one's system. A paper published by the American Journal of Clinical Nutrition states that, "Breath acetone is as good a predictor of ketosis as is urinary acetoacetate."(22) Ketosis is the condition in which ketones are being created in the body as a result of lack of insulin. By showing that there is a clear correlation between measured acetone levels in breath and measured acetoacetate in urine, we can conclude that a device in which spikes in breath acetone are found, will be able to accurately show dangerous spikes in ketones. The connection between our design and methods already confirmed to be reliable allowed us to confidently create a necessary correlation.

Ultimately, diabetics do not want signs of ketones in their body. They typically only check their level of ketones when their blood glucose number rises above a 300. The breathalyzer can pick up these kinds of spikes in the glucose level and provide a time period of when it tends to happen. This is valuable to the user so they can easily see trends in their data.

This idea of breath analysis for clinical diagnosis has been under extensive research for the past decade. It is still under being researched and is not yet readily available for purchase on the market. This is due to the idea of creating a replacement device for glucose meters and the finger prick method. The breath analysis techniques that have been discovered so far are still not accurate enough to act as a replacement device. The main topic of research in this area of study is designing a sensor that can measure acetone concentration in the breath to the accuracy found in the blood check method.

Our device cannot serve as a replacement to blood glucose meters. Our design uses VOC sensors that are currently on the market for purchase. These available sensors are not accurate enough to replace finger pricks with breath analysis. Instead, our design is an alternative to constant finger pricking throughout the day to encourage proper diabetic monitoring. The user can blow into our breathalyzer to get an idea of a range of blood glucose concentration they fall into. This should correspond with the range of ketones found in the urine checks and be confirmed by the blood check. This will provide the user access to general spikes and drops during their day, which are crucial to proper monitoring.

The goal of this project was to provide an alternative to pricking the finger so many times per day. Our device allows the user to have a ballpark range of where their blood glucose level sits giving them enough information to decide if they need to proceed with checking more accurately with the blood glucose meter. This takes the pressure off repeatedly sticking themselves with a needle and gives them an easier, more relaxed way to monitor their disease.

2.2 Project Objectives

One objective for this project was to make the design small enough to easily fit inside a pocket or a purse. This characteristic is attractive to diabetics who are used to carrying around larger packs that contain their meter as well as test strips. The smaller size helps the user to be more inclined to bring the breathalyzer around with them, as they should.

Another objective was to feature a sensor that will have to be sensitive enough to give out accurate readings of the acetone level found. By using the previously stated research as a reference and also with the help of persons already suffering from diabetes, tests were performed to ensure that the data we receive is reliable and accurate. In order to be sure that the information we got is truly indicative of high ketones we compared those values with other proven methods of analysis (ie urinalysis ect…). This provided confirmation that our sensor is sensitive enough to make reasonably accurate readings. We needed to compute the corresponding ketone level from the breath, and convert the final reading to a number diabetics are used to seeing. This is normally a range between 70 and 200.

We also needed a system to make sure that the threshold for breath has been reached for accurate readings. The next objective was to implement a blinking LED that easily informs the user when they can stop blowing. This is done with a simple code that blinks one color to indicate that the user needs to keep blowing and blinks another color to indicate that the sensor has gathered enough information for an accurate reading and the user can stop blowing. In order to keep this from being a safety hazard, we also plan to implement a counter and breath power meter in future iterations so that during this process we make sure that the user is not blowing for an unhealthy amount of time.

When using a glucose meter, there is a long process of downloading all the information from days, months, or years onto a computer. Another objective of this project was to provide a user-friendly way to track the glucose levels for the day. With each reading, our device uses wireless communication to send the data directly to the smartphone for immediate and easy-to-use tracking and logging. This can display a trend in high ketone levels and provide a visual aid for personal spikes.

The final objective was to have a power supply that is lightweight, portable, and rechargeable. We chose a lithium ion battery to achieve the desirable weight.

The battery is attached to the device for portability and allowing for easy on-the-go use. The device is also wearable, meaning that it will have the ability to be carried on the person at all times, whether that might be in a purse or a pocket. The device can be able to be powered for a full days use, approximately 12 to 14 hours, before having to be recharged, though this requires some power optimization to fully achieve. It requires at least 300 mA of current at 5 Volts. Low power consumption allows the user to be worry free of their breathalyzer dying in the middle of the day. It will be charged and ready for use at breakfast, lunch, and dinner times throughout the day.

The overall goal of this project was to make a low cost, low power, user friendly breathalyzer that is sensitive enough to give out accurate readings at any time and allow diabetics another resource for health management.

Overall Objectives:

1. Hand-held design that can fit into a pocket or small purse allowing for easy, daily transportation.
2. Sensor that can make accurate readings of acetone level in the breath and converting these readings into familiar diabetic numbers.
3. Implement blinking LED to provide an easy way for the user to know how long to blow in order to receive proper readings.
4. Use wireless communication to send real time data to the users smartphone for easy tracking and logging allowing a visual aid for personal spikes.
5. Have the device be powered by a lightweight battery that will hold a charge for a day of use.

2.3 Requirements and Specifications

When completing any project, there is an arrangement of prerequisites and determinant factors that should be followed in order to achieve the end goal. The requirements and specifications section of this paper allows us a space to discuss the fundamental arrangement of necessities and details needed to succeed at making our Blood Glucose Breathalyzer. The relationship between the overall objectives and requirements and specifications for a project provide guidance on how to tackle such a big task. The objective gives us an end goal, the engineering requirement tells us how we can design something to fit that objective, and the specifications tells us details about what we need to do to make it all happen. This relationship is shown below in Table 1. The specifications listed below directly correspond with our overall project objectives. In order to achieve our objectives for this project, we must meet the following requirements specifications:

- Unit Size
- Sensors
- Connectivity
- Power

The unit size for the final product is important because the first overall objective is that the device will be hand held, meaning that it will have dimensions small enough to fit in someone hand or pocket for easy portability. It should also be made of a very lightweight material so that it ends up being no heavier than a typical cell phone. These days, cell phones are getting bigger and bigger and no one seems to have any trouble carrying those around so our plan for the device to be wearable or portable should be easy to realize.

The next objective that we need requirements and specifications for is obtaining a sensor that can make accurate readings that we can convert into diabetic friendly outputs. We will use a volatile organic compound (VOC) sensor to measure the acetone in the user's breath. In order to be sure that the reading is accurate, we will calibrate the sensor to our liking. This will be carried out through a series of tests and by comparison to other verified procedures of measuring blood glucose levels, such as by blood and urine. The requirement will double with two of our overall objectives for the project because the blinking LED is an objective that allows us to know how long to continue to blow into our device before the sensor has collected enough data to make an accurate reading. We will incorporate three sensors in total- one for high concentrations found in the breath, one for low concentrations found in the breath, and one temperature and humidity sensor to adjust the output as needed.

Connectivity is a requirement and specification that goes with the next of our overall objectives. In order to achieve our objective of allowing the user to receive data directly to their smart phone in real time, we will need wireless connectivity. This way, our device can wirelessly hook up to the user's smart phone to send their data to right away and easily conclude trends and patterns. We will implement this using an Android application to interface with our device. Since we are designing our device to be portable and on the person at all times, we do not need a large distance to cover for our connectivity. We are approximating that the connection between the two devices only needs to be a maximum of 20 feet away.

And the last objective we need requirements and specifications for is that of long lasting power. We will use a rechargeable battery that can last a whole day. If the user is out all day, we still want them to have access to our device even if it is late at night. It will need to be recharged every night, just as one would do with their cell phone. This rechargeable battery should be lightweight to continue to achieve other objectives like being hand held and lightweight.

Objective	Requirement	Specifications
Hand Held Design	Unit Size	• Will fit in a single hand • No heavier than a typical cell phone • Made of lightweight material
Accurate Sensor	Sensors	• Use a volatile organic compound sensor • Temperature and humidity sensors to properly adjust output • Blinking LED to tell when the sensor has received the information needed
Blinking LED		
Wireless Connectivity	Connectivity	• Android application interface • Connection up to 25 feet away
Rechargeable Battery	Power	• Should last entire day

Table 1: Relationship between project objectives, engineering requirements, and specifications

The entire unit will be able to fit easily in the hand. A box will contain two chambers; one for the microcontroller PCB control unit and another to house the air chamber. This air chamber will hold the breath, utilizing a one-way vent, and containing all the necessary sensors. The air chamber will comprise the majority of the size of the design. Connected to the box will be a mouthpiece for the user to breathe into.

The bottom of the unit will have a PCB mounted to it that contains the MCU and all required circuitry. There will be connector headers to connect the necessary sensors. This allows for some freedom of the sensor configuration. From here, there will be a micro USB port that will be able to charge the lithium-ion battery. The battery can easily be mounted on the side. The bottom part, or Control Unit, will be only as large as necessary to house the PCB.

There will be overall power control which, when turned on, will supply the heaters with power and have the MCU on standby. This should be left on since the heaters must be very hot to stabilize the resistance. This stability can then be conveyed to the user with an LED indicator. Once this occurs, the user would press the read button and after holding for 5-10 seconds, would breathe into the air chamber via the mouthpiece. Once the sensor stabilizes long enough with the new resistance level, the data will be outputted. Software flow will be important to assure the data received is accurate.

The expectations of this device are simple. The breathalyzer should be able to give the user a general estimate of large blood sugar spikes through ketone levels. There should be a clear range or level of acetone that shows a potential health concern. A single breath and minutes of analysis should be all that is needed to get usable data on the user's mobile device.

The largest component of this project is the Aluminum Enclosure -- Inches (4.39" x 2.34" x 1.06") mm (112mm x 60mm x 27mm) roughly. Concerning the enclosure, the goal is to utilize a design that gives insurance against access of dust and sprinkling water, and likewise accommodates enhanced EMI/RFI protecting. We are also pushing for a smooth surface with no sharp edges. The last thing a manufacturer wants to accomplish is designing a device that brings harm to consumers and is very threatening to the safety of children. Therefore, smooth edges on the enclosure are a requirement that can't be stressed enough.

To accomplish the power consumption desired for this project, a Lithium Polymer battery (henceforth referred to as "LiPo" batteries), is required. LiPo batteries are a newer type of battery now used in many consumer electronics devices. They have been gaining in popularity in the radio control industry over the last few years, and are now the most popular choice for anyone looking for long run times and high power. There are other top rated batteries in the market such as Nickel-Metal Hydride (NiMH) or Nickel Cadmium (NiCd) we could use to meet the specification requirements of this project but the LiPo provides the specifications needed to accomplish those goals. LiPo batteries offer three main advantages over the common Nickel-Metal Hydride (NiMH) or Nickel Cadmium (NiCd) batteries:

- LiPo batteries are much lighter weight, and can be made in almost any size or shape.

- LiPo batteries offer much higher capacities, allowing them to hold much more power.

- LiPo batteries offer much higher discharge rates, meaning they pack more punch

When it comes to transferring data or readings from one device to another, a lot of risks are taken but those risks can easily be eliminated with our design and I'm sure the consumer of this breathalyzer will be satisfied because it is designed so that the Bluetooth® chip isn't constantly on and searching for the phone continuously. The chip is to be programed to only be on if the device has a reading to send to the phone; otherwise, it's off, which contributes to the conservation of power. It should also be programmed to have the Bluetooth chip connect to an already known device spontaneously when the read button function is activated. On the other hand, it should be off when the read button isn't pressed to preserve power.

This device requires the ability to store information and data. The storage feature will also allow the device to save power. If there's no storage, the phone will be required to be on constantly, which is a deficit and makes the device harder to use.

Concerning the other requirements and specifications, the list below illustrates them all.

- Easy-to-use controls: The Breathalyzer ought to have controls or UI that are straightforward, so usability is an unquestionable requirement.
- Readable results: A breathalyzer ought to make perusing and translating the results of a Blood Glucose test very simple. Huge, splendid, or illuminated numbers are often highly recommended
- Low power requirements: The breathalyzer ought to be prepared for more than three hundred uses before the batteries need recharging.
- Quick execution: The more perplexing breathalyzer models in the business sector require 30 to 40 seconds of warm-up time. Our pick is prepared to go in around 20 seconds—sufficiently quick that a fretful individual won't leave before being tested
- An approach to share: Replaceable mouthpieces are a remarkable approach to expel a companion's (substantial) worries about germs, and utilizing them likewise makes the gadget's readings more exact.

3.0 Research and Investigations

3.1 Research Related to Project Definition

Our project depends on the understanding of breath analysis. Since it is still being heavily researched at multiple universities and institutes, we need to carry out extensive research of our own in order to wrap our heads around what needs to be done to make our project operational.

3.1.1 Research in Breath Analysis

Breath analysis is the one of the least invasive and increasingly popular methods for obtaining and monitoring common health concerns. There are several applications for breath tests, but for the sake of this project, we will focus only on breath analysis for clinical diagnosis. In the past, the only technology for diagnostic methods has been through urine and blood tests. In recent years, we have introduced breath analysis.

Breath analysis is the best alternative to the current methods for discovering and monitoring disease. It does not involve any discomfort or pain that usually comes with the urine and blood tests. It also provides a method through which patients are more inclined to properly take care of themselves and their health concerns. And lastly, breath analysis has better hygiene quality than other methods such as urine and blood. For instance, a diabetic who has to check their blood glucose level many times a day gets tired of constantly cleaning their fingertip and changing their needles for pricking. This causes the patient to get lazy and neglect these steps for accurate readings as well as hygiene. Dirty fingertips can cause the glucose meter to output inaccurate readings and reusing dirty needles is very unhygienic. A breathalyzer allows for the same accuracy in readings but without any of these annoyances on the patient. It is said that clinical breath analysis tests as well as personalized breath monitor will become a reality within the next decade.

The majority of breath is made up of nitrogen, oxygen, carbon dioxide, water, and inert gases. The rest of the content found in one's breath is a small fraction consisting of thousands of volatile organic compounds (VOC) with concentrations in unit of ppm (parts per million). There are about 3500 VOC's in the breath that exist at extremely low concentrations. Common VOC's found in human breath- including isoprene, acetone, ethane, and methanol- provide the link between breath analysis and clinical diagnosis.

The biggest difficulty in breath analysis is the necessary action needed to separate breath sample collection by means of preconcentration. Since VOC's make up tiny concentration areas of the breath, it is easy to overlook them before preconcentration because most conventional instruments wouldn’t pick them up. This process allows for the separation of large common breath components so that the VOC of interest that is only a few parts per million will be able to be extracted.

History in breath analysis goes back to Ancient Greek physicians who discovered the connection between bad breath and disease. They figured out that the aroma given out by human exhales breath could provide clues to diagnosis, as shown below in Table 2. The beginning of breath analysis began long ago with the following well-known odors in bad breath that have been linked to disease over time:

Aroma of Exhaled Breath	**Disease Connection**
The musty, fishy reek smell	Advanced Liver Disease
The putrid, sewer smell	Lung Abscess
The urine-like smell	Failing Kidneys
The sweet, fruity, “rotten apple” smell	Uncontrolled Diabetes

Table 2: The relationship between breath and disease has been is a well-known part of diagnosis

Ketones are produced by the liver and are used as a substitute form of energy when the body cannot use glucose as the energy source. Ketones are made up of acetoacetate, 3--hydroxybutyrate, and acetone. The concentration of ketones found in the body fluctuates due to diet and exercise. “The gas-phase acetone in the blood equilibrates with alveolar air (end-expired air) through the alveoli. Therefore, the concentration of acetone in breath can reflect metabolic products of diabetes.”(23)

Everyone has ketones, even healthy humans without diabetes. The difference is that if the diabetes is improperly managed, patients can enter what is called a diabetic ketoacidosis (DKA), a serious condition that can lead from diabetic coma to death and is caused by very high concentrations of ketones. Diabetics have to be more aware of the ketone levels in their body. When large amounts of ketone bodies are released, the acetone concentration rises significantly. The increased concentration of acetone in the blood can be exchanged with alveolar air. When compared with a healthy human, those with diabetes will exhibit a broad range of acetone levels in their breath. For this reason, acetone levels can be found in the breath and used as a signal for high blood glucose levels.

This idea of a breathalyzer for clinical diagnosis is a hot topic of the modern research world. Since it is still under extensive research, one would think our project isn’t realizable. We believe that with the right goals, we can achieve what we need. Our team understands that breath tests are currently not accurate enough to replace blood glucose tests. After our own research, we have decided to make this project possible by designing our breathalyzer device to show spikes and general trends. It will not be accurate enough to replace a blood glucose meter for blood tests and exact numbers. However, it will be an alternative to invasive finger pricking throughout the day. For example, a diabetic could use our device to blow and check for a number range. Each diabetic has their own specific heath plan that they are following and have a range they should try to stay inside of. The exact number isn't as important. The difference between a blood glucose level of 120 and 130 might not matter to the diabetic, but the difference between 120 and 180 is a big jump and approaching the 200's. At this point, the diabetic would responsibly take out their meter and check for an exact number after our device informed them that their numbers are rising. This approach can still lower the amount of discomfort from finger pricking all day to finger pricking only when necessary.

3.1.2 Advantages and Disadvantages of Breath Analysis

The main advantage of using breath tests is its safe and non-invasive quality. With the technology and research in today's world, breath analysis can be used to provide health information and the possibility to predict future outcomes. Despite the appealing advantages, there are also limitations to breath analysis, some of which are displayed below in Table 3. The major drawback is the lack of accuracy and stability that is provided by the standard analytical methods like drawing blood or urine.

Advantages	**Disadvantages**
Non- invasive	Lack of standard analysis
Does not cause discomfort or embarrassment	Sample collection and preconcentration are necessary because substance concentration in breath are ppb-ppt
New strategy for diagnosis of disease	Standardization of breath sample collection and preconcentration is more difficult than blood/serum
Evaluation of several disorders	Need to gain accuracy

Assessment of VOC's	Wide variation of results obtained in different studies
Less complicated mixture than blood/urine	Bulky instrumentation, time consuming, needs skilled workers
Correlated to arterial concentration hard to obtain by blood/urine	High water content of breath may affect sampling
Provides direct, sequential, longitudinal information on sampling of lower respiratory tract which is not obtainable by other means	Instruments for breath analysis are more expensive than the simple chemical tests for blood/urine
Easily repeatable	Need more established links between breath substance and disease
No work-up of a breath sample is required	-
Real time monitoring system	-

Table 3: Advantages versus disadvantages of breath analysis

3.2 Project Based Research

In this section, we will review the research that is being carried out or have already been carried out at other universities. This information is pertinent to our understanding of how to make our device.

3.2.1 University of Pittsburgh

Since our project is still highly research based, the competition appears to be in the research of sensors and technology rather than the race to put products on the market. In 2013, colleagues at the University of Pittsburgh investigated a method they call the "sol-gel approach" and combine it with their sensor design to develop "alternative devices that are noninvasive, inexpensive, and provide easy-to-use breath analysis that could completely change the paradigm of self-monitoring diabetes," said Alexander Star, the principle investigator of this project. Their method uses nanoscale molecules to make solid material by combining titanium dioxide (TiO_2) with single walled carbon nanotubes (SWNT). They designed their sensor using these materials as an electrical semiconductor, measuring its resistance.

The relationship between titanium dioxide and single walled carbon nanotubes displays a unique behavior of electronic coupling, charge separations, and photocatalytic activities upon photoexcitation and chemical exposure. Their research reports that a SWNT−TiO_2 nanohybrid with core/shell structure for the room-temperature detection of acetone vapors at ppm (parts per million) concentrations offers promising clinical applications as concentration of breath acetone has long been correlated with ketoacidosis and the blood sugar level in case of diabetes. (24)

The Journal of the American Chemical Society describes the experimental process Star and his team carried out to make the conclusions stated above. They compared the electrical behavior, UV illumination, charge separation, and acetone sensitivity of eight different SWNT−TiO_2 nanohybrid systems. All of the systems showed greater acetone sensitivity upon increasing the amount of TiO_2. The overall highest acetone sensitivity was provided by the SWNT−TiO_2 core/shell hybrid system.

This is due to two main factors:

1. Covalent bonding between SWNTs and TiO_2 creates interactions between them and electronic coupling, leading to a more sensitive interfacial charge transfer
2. TiO_2 has a less organized structure, creating more opportunity for unsaturated surface Ti centers that were more reactive to acetone molecules

This research is far beyond, yet closely related to our own. We do not have the means to research or build our own sensor in the allotted time period we have to finish this project. However, research seems to be proving that a titanium dioxide sensor is the strongest, most accurate way to implement this project.

3.2.2 Western New England University

Ronny Priefer, Professor at Western New England University and CEO of New England Breath Technologies, created a breathalyzer device that tracks blood sugar levels. This device is different from other related devices because it only accounts for the acetone levels in the breath and doesn't react with any other components found in breath. This is essentially what we want our breathalyzer to do too. Priefer has the same motives we do for this project, "a painless alternative to continuous finger-prick blood monitoring by diabetics".

Priefer's device is handheld size, similar to our idea, and uses "multilayer nanotechnology to detect acetone that has been shown to correlate with blood-glucose levels in the breath of diabetics". New England Breath Technologies, LLC is creating this device and will mimic a personal alcohol breathalyzer. They have a working first generation model; however it still needs to be down sized as it is currently about the size of a notebook. The design includes a small tube that the user breathes into with a disposable, coated slide that is inserted at the top of the device. These slides are UV-transmitting and are coated at differing pH values before being exposed to acetone and water vapor in the breath. “By designing a film with the suitable polyelectrolytes it is possible to crosslink them with acetone resulting in an alteration in the physicochemical nature of these multilayered films. By quantifying these changes (e.g. spectrally) the amount of acetone that was originally introduced to the system can be determined.”(25) This approach is advantageous because in concentrations above 0.5% of acetone, there is no other breath volatile containing a reactive carbonyl that will be found. This allows for the breathalyzer to accurately measure glucose through its relationship with ketone concentration.

One complication found in the research involved in creating this device was the previous breathalyzers they tried to make couldn’t handle the humidity or temperature in a person’s breath. This is a complication we are aware of. The sensors components used to measure acetone in the breath seriously react with the humidity and cause false readings. After studying nanotechnology for a few years, Priefer figured out that polymers could use the humidity found in natural breath to help the films measure acetone.

In other words, the polymers and acetone join together to change the physical chemistry of the film. Then, the film will be able to read the amount of acetone found, which can be directly related to the blood glucose levels.

In 2013, the university started conducting clinical tests in which patients will use both the breathalyzer and the traditional finger prick method to compare results and check for accuracy. The patients will also keep track of what their diet looked like for each day as part of the study. This will show any relationship with other foods or surrounding factors that may affect the acetone readings, such as nail polish, cigarettes, or acidic fruits. This is also essentially our plan for testing our device. We will also incorporate urine test strips for an additional source of accuracy in measuring ketones/blood glucose level.

3.2.3 Stony Brook University

With support from the National Science Foundation (NSF), Professor Perena Gouma and her team at Stony Brook University in New York developed a sensor using nanotechnology. The sensor was coated with tiny nanowires that are specific to detecting different chemical compounds found in the breath. In the case of diabetes, the nanowires would be able to capture only the acetone molecules. This makes detection less complicated because it doesn’t pick up on any unwanted compounds that could alter the readings. The Single Breath Disease Diagnostics Breathalyzer works in a more general way than others do.

There is a small valve to blow into and wait for a light on the top of the structure. If it is green, the user essentially “passes” and has no sign of biomarkers for disease or health concern. If the light turns red, the user “fails” and should consider seeing a doctor to make sure there are no serious health conditions that they should be worried about.

The process Gouma used to create the crystal nanowires in the sensor is called electrospinning. A liquid compound is shot into an electric field from a syringe causing the liquid to crystalize into the nanowires on an aluminum backing. These nanowires allow for the sensor to be able to detect specific molecules in the breath due to their varying arrangement of metal and oxygen atoms that can capture a particular compound.

The research carried out at Stony Brook University with Gouma dates back to 2010. Technology has since advanced; however, her research was very much on track with what the future held in this area of study. She was able to create 3 applications for her breathalyzer design that we see as hot topics in today’s research community. These 3 applications are a metabolic rate monitor, diabetes monitor, and hemodialysis monitor. I will elaborate only on the acetone detecting applications due to it being our specific subject of interest.

The first application she used her breathalyzer for was a metabolic rate monitor. This application has the same biomarker we are focused on in this paper, acetone. Gouma found that acetone acts as a lipid degradation product and acetone levels of 500nmol/L in the breath indicates a weight reduction of about a half of a pound per week. (34) The next application Gouma looked into is the one of interest, diabetes monitoring. The biomarker is the acetone in the breath, which is very different between diabetics and otherwise healthy humans. Acetone concentration in a healthy human is approximately 0.8ppm or less and acetone concentration in a diabetic is often greater than 1.8ppm. The acetone concentration can be higher than 500ppm when a diabetic is in ketoacidosis.

3.3 Existing Similar Projects and Comparison

There have been some attempts at creating a device such that can measure health concerns by breath. Most of these devices start out as a blood alcohol concentration breathalyzer and the creators are still researching a way to accurately measure blood glucose concentration with the same method. This idea stems from the BAC breathalyzer, so we use this section to expand our understanding of how the original BAC breathalyzer works.

3.3.1 Breathometer

The theory behind our project is still being heavily researched and undergoing improvements for the best technology. There are not any products currently for sale that can do the functions that our project will be able to do. However, there are companies that plan to make these devices in the near future. Charles Michael Yim, founder and CEO of Breathometer, stated that they will no longer be making alcohol specific devices. Breathometer is the leader in breath analysis technology. In 2014, the company decided to expand their technology to be able to detect levels in the breath anywhere from bad breath to diabetes.

The Breeze is the second generation Breathometer product that sends data via Bluetooth communication to your smartphone. This device measures the ethanol in breath to determine the blood alcohol concentration to help users make responsible decisions when out drinking. In 2015, Breathometer created a partnership with Uber to promote the importance of making safe choices regarding drinking and driving. Uber offers discounts and free rides to Breathometer users with hopes that they will have a positive impact on the drinking and driving phenomena the world has today.

The Breeze was the last model that Breathometer made for determining alcohol levels by breath. Their future expands the technology to measure bad breath, weight loss, detect asthma, and detect diabetes. This can be done by changing what the Breeze measures by changing the chemicals within the sensor used. Mint, Breathometer's newest product, uses the same technology to measure stinky sulfide levels in the breath to alert you if you have bad breath. Since bad breath is a known symptom of dehydration, Mint also detects and alerts you if you're dehydrated.

The next model Breathometer plans for will measure acetone level to track dietary weight loss. Standing on a scale will show you a number of pounds, however it does not tell you what is actually going on inside your body. The idea of Breathometer's next model, Levl, will be to measure weight loss by breath because acetone is the byproduct of fat burning. Breathometer's ultimate goal is to combine all of these technologies into one big product using new sensor technologies that allow for more than one measurement per chip.

Although this does not directly reflect our projects goals, Breathometer is measuring acetone by breath and connecting it to a heath concern, which is essentially the big picture. It also unveiled the idea of a portable and wearable device that will track a health concern. The original Breathometer only detected ones blood alcohol concentration and did not have wireless communication. Breeze introduced the wireless features but still only measured ones blood alcohol concentration. It wasn't until Levl for the company to implement measuring acetone in the breath, connecting it to a health concern, and using wireless Bluetooth communication. Levl is Breathometer's product closest to our project, though it is still not exactly the same.

3.3.2 The Floome

The Floome is a portable smartphone breathalyzer that measures blood alcohol concentration (BAC). This device on the market has many of the same features as we have in our project. The only difference between BAC breathalyzers and our diabetic breathalyzer is the compound in the breath measured by the sensor. The user can plug their Floome into their smartphone via the audio jack and keep track of personal drinking habits. The Floome is also a learnable application that can calculate how long your body needs to lower your BAC into a safe range by entering personal information such as gender, weight, and height. It will be able to provide personal results after it is given the opportunity to learn your metabolic rate. Another feature is screen color after a blow indicating whether the user is above the legal limit or not. Red meaning the user is above the legal limit and green meaning the user is within a safe BAC range. This device is also electronically designed for ultra-low power consumption with no external batteries needed. The sensor featured in the Floome is the same sensor that law enforcement uses for BAC breathalyzers, giving it extreme accuracy. Additionally, the Floome includes a replaceable mouthpiece for hygiene and sanitation. Although the breathalyzer outputs a blood alcohol concentration instead of a blood glucose concentration, the overall concept of the product is very similar.

3.3.3 BAC Breathalyzer

The original idea of our project stems from the existing Blood Alcohol Concentration (BAC) Breathalyzer used by law enforcement to determine whether someone is driving while impaired or driving under the influence. If we have been using breath alcohol testing devices since the 1940's, should be we able to design other breath tests by now? Our team developed this project idea from one simple question, "If we can measure blood *alcohol* concentration from breath, why can't we measure blood *glucose* concentration from breath?" Apparently, this idea is easier said than done.

The traditional and more accurate tests for alcohol concentration are through blood tests and urine tests, just as they are for glucose. These kinds of tests are impractical for testing for the legal driving limit. Police needed a way to measure this offense without invading the suspect's body. Similarly, we want a non-invasive way to measure glucose levels in diabetics.

Alcohol is absorbed in the bloodstream and when the blood goes through the lungs, alcohol moves across the alveoli into the air. This is the same process that happens with acetone when there is presence of ketones in the blood. There is a direct relationship between the concentration of alcohol found in alveolar air and the concentration of alcohol found in the blood. This relationship is a ratio of 2100:1 meaning that it takes 2100 ml of alveolar air to match the equivalent concentration found in 1 ml of blood. The studies previously mentioned have proved that there is also a direct relationship between the concentration of acetone found in alveolar air and in the blood. This is due to alcohol and acetone being volatile. The two technologies are parallel.

There are three different methods that have evolved for determining a person's BAC. Every method described here involves a mouthpiece in which the person blows air into and a chamber that holds the air from the breath. The remaining design of the system varies depending on the method chose. These three methods are:

- Breathalyzer- chemical reaction from change in color
- Intoxilyzer- detection by infrared spectroscopy
- Alcosensor III or IV- chemical reaction in a fuel cell

The Breathalyzer device is based on a chemical reaction that results in a change of color when it reacts with alcohol. The person will blow into the mouthpiece and provide their breath sample of sulfuric acid, potassium dichromate, silver nitrate and water. Each of these substances is pertinent to creating the chemical reaction needed to detect BAC. During this reaction, the dichromate changes color. The amount of color change is related to the amount of alcohol in the breath. For instance, when a person blows into the breathalyzer device and the color changes tremendously, from red to dark green, it indicates a high level of blood alcohol concentration. However, if it barely turns color, the indication is that the person is probably within the legal limit. The reacted mixture from the breath is compared to an unreacted mixture in the photocell system, producing an electric current. The photocells are connected to a meter and the electric current causes the needle in the meter to move from its resting place. The operator will bring the needle back to its zero by turning a knob, which reads the level of alcohol. The farther the needle moves, the more turns by the operator, the higher the alcohol concentration.

The Intoxilyzer device is based on the way molecules absorb infrared light using infrared spectroscopy to detect alcohol concentration. Identifying ethanol in the breath can be determined by the wavelengths of ethanol's bonds and measuring the absorption of infrared light indicates how much ethanol is there. An infrared beam of light is generated by a lamp and focused by a lens onto a spinning filter wheel. The light passing through each filter is converted to an electrical pulse by detection from the photocell. The electrical pulse is sent to a microprocessor, which calculates the BAC.

The Alcosensor device is based on modern fuel-cell technology. The fuel cell is made up of two platinum electrodes with an acid electrolyte material in between them. The platinum oxidizes the alcohol found in the breath when the person blows air across one side of the fuel cell. This process produces protons and electrons that create an electrical current. The more the electrical current indicates higher alcohol content. A microprocessor is attached to measure the current and calculate the BAC. (33)

3.4 Parts and Components Research

Aside from the research that was performed for the original project idea and needed to learn about the topic, another important form of research was done to determine the correct parts to buy. More about these parts can be found in Section 5 and 6. The respective data sheets can be found in the appendix. In this section, we will analyze the options and reasons why we chose the parts for our project.

3.4.1 Temperature and Humidity Sensor

The temperature and humidity sensor is needed for our project because the concentration of volatile substances in human breath is typically in a unit of parts per million or parts per billion. Therefore, it is extremely hard to measure such a small concentration with the presence of large substances that take up most of the concentration of breath. There is a certain temperature/humidity dependency that our VOC sensors will require. This sensor will be used to adjust the output so that we have an accurate measurement of the acetone without interference from the other components in the breath. There are two main sensors of interest- the DHT11 and the DHT22. The DHT22 goes by multiple different names such as the RHT03 and the AM2302 but they are essentially all the same thing. The both of these sensors of interest are similar and very low cost.

The DHT22 provides more accurate readings for both humidity and temperature when compared to the DHT11. It allows for 0 to 100% humidity readings with 2% to 5% accuracy as opposed to 20% to 80% humidity readings with 5% accuracy. And it allows for -40 to 80 degrees Celsius temperature readings with 0.5 degree accuracy as opposed to 0 to 50 degrees Celsius temperature readings with 2 degree accuracy. The body size is a bit smaller for the DHT11, but the difference in the accuracy is worth getting the DHT22 even though it's a little bit bigger. We have plenty of room to implement the design and keep it hand-held size. The power and current are the same for both the DHT11 and the DHT22. Both also have four pins in the module even though we will only be using 3. The DHT11 has a faster sampling rate of once every second and the DHT22 has a sampling rate of once every two seconds.

The DHT22 will require some outside components as well in order for it to work as desired for the use in this project. It will involve a 3.3V linear voltage regulator and a pull up resistor. The regulator will be connected to the 5V source to achieve the 3.3V of DC power. The pull up resistor is to be connected from the DC power to the pin on the DHT22. The sensor will be connected to our MCU to transmit the data received.

3.4.2 High Concentration VOC Sensor

This sensor is required for our project to measure the high levels of concentration of the VOC's found in the breath. This measurement will be able to tell the user if they are experiencing a spike of their blood sugar levels due to high concentration of acetone in the breath. We intend on using the TGS822 for this part of our project. Acetone is directly referenced in the data sheet for the TGS822, so we know that it will measure our VOC of interest. This sensor features high sensitivity to organic solvent vapors, which in our case will be acetone. The concentration range in parts per million are between 50-1000 ppm. It also has long life expectancy, and low cost.

The TGS822 uses a tin dioxide semiconductor as its sensing element, which has a low conductivity in clean air. The conductivity of the sensor will go up when it detects a gas and increase depending on the gas concentration in the air. Then, we will use an electrical circuit to change the conductivity relationship to an output signal corresponding to the gas concentration. This sensor implements the combination of two important relationships. The relationship between the resistance of the TGS and the pressure of oxygen is important because reduced oxygen pressure increases the sensors conductivity. This is due to the transfer of electrons from differing energy levels of the gas molecules and the semiconductor surface. The relationship between resistance of the TGS and the concentration of gases is important because decreased resistance implies increased conductivity resulting in the detection of concentration of gases in ppm.

The main competition for this sensor is the MQ-3 because of similar features between the two. Both the MQ-3 and the TGS822 use tin dioxide as their sensing material resulting in very similar detection capabilities. The range of detection is 10-1000ppm for concentration, which is a larger range than the TGS822, but since we will be implementing a low concentration sensor as well, this does not apply to our decision making. A huge disadvantage for the MQ-3 with regards to our project is that the data sheet does not directly mention acetone as a detectable gas. We do not want to risk the sensor not being able to detect our target gas. Also, the heater power consumption is the highest for the MQ-3 with 900mW where the TGS822 only consumes 660mW of power. The MQ-3 can be compared to the MQ303A, which works the same way as the other sensors except works on a lower voltage. The concentration for the MQ303A is from 20-1000ppm. After weighing options and extensive decision making, the TGS22 is the best fit for the part we need in our project to measure high concentration of VOC's in the breath.

3.4.3 Low Concentration VOC Sensor

This sensor is important to our project in the same way that the high concentration sensor is. Low blood glucose level is just as dangerous as high blood glucose levels, so we need to be able to measure spikes as well as drops. The high concentration sensor will provide the spikes and we need a low concentration sensor so provide the user a way to know when they are experiencing low blood sugar levels. We intend on using the WSP2110 for this part of our project. Acetone is directly referenced in the data sheet for the WSP2110, so we know that it will measure our VOC of interest. The concentrations in parts per million are between 1-50 ppm. This sensor features high sensitivity to organic gases, such as acetone, low power consumption, and a long life expectancy.

The WSP2110 uses an aluminum oxide material as its sensing element. Conductivity of the sensor is increases as the concentration of the target gas increases. Then, we will use an electrical circuit to change the conductivity relationship to an output signal corresponding to a gas concentration. This sensor is more sensitive, but only detects a small range of concentration and it is more expensive. The WSP2110 is compared to the same sensors that the TGS822 was compared to so, with the same decision making process, we bought this one for the same reasons as stated above. For the sake of our project, we need the combination of the WSP2110 and the TGS822 to provide a more expansive scale for lower and higher concentration levels.

3.4.4 Wireless Communication

The main research involved in deciding how we can implement wireless communication on our device is deciding between a Bluetooth module and a WiFi chip. Bluetooth communication seemed to be the obvious answer for our project for a number of reasons. The biggest reason we chose Bluetooth communication is because of the fact that the user of our device would never be more than arm's length away from it. Our device is going to communicate with the users cell phone, which is most likely going to be in their hand or in their pocket. Bluetooth communication is useful for transferring information between two nearby devices whereas WiFi is better for communication of networks over a larger range of distance. The bandwidth is lower for Bluetooth communication than WiFi, but we don't need a high bandwidth for our use of this technology. And lastly, the cost and power for Bluetooth devices is lower than for WiFi.

The Bluetooth module that we decided on is the RS232 TTL HC-05. Although one option is to buy an MCU that has the Bluetooth module already built in, we opted to get a standalone module to provide us with more flexibility with exactly how we implement the module into our PCB. The HC-05 option features low power and low cost and can cover a range of about 10 meters. These features provide us with more than we need at a very low cost.

3.4.5 Microcontroller Unit

There are many ways we could go about picking an MCU, so we decided to start by agreeing to use Arduino to program our MCU. From here, we could filter our choices to Atmel chips since they are regularly used by Arduino devices. The three we focused on were the ATtiny85, ATMega328p, and the ATSAMB11.

The ATtiny85 is similar in structure to the ATMega328p. It features a very small size, low cost and uses less power than the other two options. However, there are limited GPIO and the flash storage size could be extremely limiting. This could be problematic for a group of four electrical engineers and no computer engineer to help us fit our code in the allotted space. The ATMega328p is the familiar choice and used in many embedded projects and only costs one more dollar than the ATtiny85. The flash memory size is much larger which will give us more room for our code. It also features 23 I/O pins.

The ATSAMB11 is much more expensive than the two options previously discussed. It features extra flash memory and available pins but we don't really need them. The ATMegal328p will give us enough flash and available pins for what we need to implement to get our device working. The main positive attribute for this chip is the built in Bluetooth module that we could use for communication. There are clear benefits with this option but, the combined price of ordering a standalone Bluetooth module and the ATMega328p is still less than ordering the ATSAMB11. After weighing all of these options, we will use the ATMega328p for our MCU in this project.

3.4.6 Power

A lithium-ion battery is a popular choice for today's devices despite the small chance that it could burst into flames. There are several advantages for lithium-ion batteries including that they are much lighter than most other battery types. The lightweight feature goes along with one of our overall project objectives describing that the final product of our device be small, hand held, portable, and lightweight. Lithium is a very lightweight and highly reactive element, which provides a higher energy density. The lithium ion batteries hold their charge longer and last longer than most other comparable batteries.

Another option of battery type that we could use is a nickel-based battery, such as nick-metal hydride (NiMH). However, this type will not last as long as a lithium ion battery would. Since our project is supposed to aim at making life easier for diabetics, we don't want them to have to worry about their battery dying all the time or having to replace the battery often. The rated voltage for NiMH is 1.25V versus the 1.5V from the lithium ion, which isn't a huge difference. The recharge capability for NiMH is several hundred of cycles, which is similar to the lithium ion.

The lead-acid battery is the last battery type that we researched and considered for our project. These are large, heavy, and would not be ideal to use for our small lightweight device. It lasts hundreds of recharge cycles and can store only 25 watt-hours per kilogram. A typical lithium ion battery can store 150 watt-hours. It is obvious that we will choose the lithium ion battery for our project.

A lithium ion battery is not just a standard battery. We still have to decide about shape, size, and how many amp hours will suffice. The cells can come in a cylindrical round shape or a flat rectangular shape. We will choose the flat rectangular shape for the battery because there will be more places that we can fit it into our design. We can place it along the edge or up on the top of our device. Lithium ion batteries come with a temperature sensor to keep battery from overheating and being ruined forever. A voltage converter and regulator circuit is a safety feature for the battery to be sure that it stays within the allotted levels of voltage and current that it can handle.

3.4.7 PCB Vendor Info

There are several vendors that can supply us our PCB. Since we have never ordered a PCB from our own design before, this took a bit of practice, research, and time. We will be using Eagle PCB design software to make our design and to send the file to the vendor we chose. The file includes the breadboard, schematic, and PCB design. The price of the PCB will depend on the complexity and size of our design. We believe that the final design that we come up with will be generally simple due to the fact that our project does not require many external components. Also, our device is being designed to be hand held and low cost. Our PCB order is going to be the most expensive part of this project, but with the simple design and small size, we should be able to keep the price down. Some options for PCB manufacturers are listed below:

- Advanced Circuits
- Barebones PCB
- 33Each
- FreeDFM
- Alberta Printed Circuits
- E-TekNet
- Gold Phoenix
- Olimex
- OurPCB
- PCBCART
- PCB Express
- Sierra Proto
- BatchPCB
- HeadStudio PCB Proto

- SeedStudio Fusion

Advanced Circuits is a well-known company that works with Aerospace/Defense and commercial markets and leads the industry in PCB manufacturing. BareBones PCB, 33Each, and FreeDFM are all part of Advanced Circuits providing different options for cheap PCB boards. They can be shipped as quickly as in one day, however it will cost more for a one-day turn. Advanced Circuits has been in the PCB manufacturing industry for over 25 years so they are a dependable option to go with for good quality in a timely manner. They offer 3 low cost options for PCB printing and also have a student program for discounts and sponsorships. They can provide 0-10 layers within 5 days. We are aiming to make our PCB only two layers, which can be done easily through Advanced Circuits. If we have 5 days to spare, which we probably will, it will only cost $33 for 2 layers and $66 for 4 layers. BareBones will do the 2 layers with more restrictions but will have it done in 1 day. FreeDFM is just a file check service that Advanced Circuits provides to make sure the file and data we send is compatible for proper manufacturability. This is nice because it will give us report stating any possible issues that our design could run into while being printed. Since we are on a tight schedule, this could be helpful because we don't have time to waste making an incorrect PCB or waiting for a new one. Advanced Circuits offers several features that are attractive for more advanced circuit designs, but we won't need most of these features for our simple design.

Alberta Printed Circuits (AP Circuits) is another well-known PCB fabrication company that has been around over 30 years, so they probably offer a dependable service just as Advanced Circuits did. They provide "Day Rush" service as well as "Next Day Shipping" with no additional fees. There are plenty of fabrication services that they offer such as multi-layers, laminate options, drill settings and sizes, and more. AP Circuits has a very helpful ordering system that shows all the possible options and what files need to be included. It does not give an estimated price without seeing the files, which makes sense so they can see how much work needs to be completed and base the price on that. They do have a job estimator section where we could fill in the details about our project and see estimation on how much the PCB will cost. There is only an even quantity option, so our job would cost approximately $50, but that would be for two PCB's. This is also for two layer boards only. To get an actual quote, we will need to gather official design files and prepare an order with those submitted before we have an idea of how much it will cost. AP Circuits is still a good option for our PCB vendor due to their reliable reputation, but the total cost is still up in the air.

E-TekNet is another PCB manufacturer that we looked into because they have low prices and on-time shipping. However, we learned that there is a minimum of ordering 4 pieces to get them at that low price of about $20 each for two layer boards and $47 each for four layer boards. They are known for mass production and would be good to go for if we needed a large amount of PCB's, but since we only really need one, this option isn't ideal for our project. E-TekNet is a "one stop shop" that will provide the bare board and PCB assembly, which makes it convenient for ordering. They offer prototype production as well as mass production. They also have bare board service completed in as quickly as 1 day and PCB assembly within 5 days. The reason they can produce this much in such a small amount of time is due to the fact they prepare everything in the same place. Their products and services include Bare Boards Manufacturing, PCB Assembly, PCB Solder Stencil, PSB Design, PCB Reverse Engineering, and PCB Assembly Turkey Service.

Gold Phoenix PCB Company seems to be another dependable option to order our PCB from. They have extended capabilities such as rigid PCB capacity, flex PCB capacity, metal core PCB capacity, PCB assembly capacity, PCB equipment, and PCB assembly equipment. They claim to have the "Best PCB Deal on the Internet", which caught our eye, but for the smallest size with two layers and no features, the price starts at $100 with a shipment of 5 business days. Since the previous options seem to have cheaper prices, we will go with a cheaper company. The reason this Gold Phoenix Company is said to have the best deal is because of the features it offers and it also offers PCB pools and a components mart.

	Low Cost	2-layer or 4-layer	File Check Service	Student Discount	Requires more than one quantity	Next Day Shipping
Advanced Circuits	✓	✓	✓	✓		
AP Circuits	✓	✓			✓	✓
E-TekNet	✓	✓			✓	
Gold Phoenix		✓			✓	
Olimex	X	X	X	X	X	X
OurPCB		✓			✓	

Olimex has a PCB prototype service that is temporarily suspended due to the lack of free capacity. That makes this option and easy one to cross out of the list.

OurPCB is the last company that we will be researching for our PCB vendor. OurPCB is a younger company compared to the other ones we looked into as it was only founded in 2005. This could be a good thing or a bad. It could mean that they have newer, more advanced, cutting edge technology. It could also mean that they don't have as much experience in good quality and reliability of their product. Their minimum time for bare PCB is 2 days, which is comparable to the other options seen above as well. Their features are also very similar to the other companies we checked out and they do not have a general cost set up for their PCB services, which makes it hard to estimate how much we would be spending.

At this point, we've done enough research to see the main features that we are going to need for our PCB. All of the companies described above are going to offer us essentially the same thing and the price difference is due to the amount of features they offer because it makes the order more complex. Our design won't need most of these extra features so it makes the most sense to go with the cheapest option with the least amount of extra features. A visual representation of this decision is shown above in Table 4. We want our final project to be low cost so we will have to make sure that we chose our PCB vendor wisely. We predict that we will be rather prepared for this order during Senior Design II, so we don't think a rushed shipment plan is an important part of our decision. It is obvious to go with our first option, Advanced Circuits, because they offer the lowest rate as well as a student discount. The student discount is important because it allows us to have no minimum quantity order. Most of the options have a minimum of 2 or 4 quantity in order to get the cheap rates, but we don't need that many. We only need one. An issue that might arise from using the student discount is that the product has to be shipped to a university, not to our homes. We will work out the details on shipment during Senior Design II.

Table 4: Comparison of the possible PCB vendors and their corresponding features

4.0 Standards and Constraints

There are standards and constraints for all new ideas and projects. In this section, we will explore the standards and constraints that we need to be aware of throughout the making of this device.

4.1 Project Related Standards

One can only envision the agony and frustration if they had to purchase light bulbs that didn't fit into lamps, or if there were no conjoint sizes for clothing, or common-sized spark plugs for automobiles, or common-sized wall outlets at different homes, common-sized gas pump handle at different gas station or if trains couldn't interchange from one state to another for the reason that the tracks were a different gauge. How horrible and agonized would everyday life be without having some sort of standard to regular conveniences?

Standards are something that the greater part of us acknowledges as a component of our regular life. Yet, they have turned out to be such a necessary piece of our presence that the normal individual gives next to zero thought to ordinary items and administrations and how they function. They make current accommodations conceivable: lights fit into lights, electronic records are exchanged over the Internet, trains move between states on the grounds that the tracks are the same gauge, and the list continues on. Standards are a vital part of engineering design. With the utilization of standards, organizations have the likelihood to disentangle item improvement and furthermore upgrade the adequacy of items. Norms are records that offer particular points of interest to be met by the items and produces. The benchmarks that will be identified with our project design are made by the FDA, CE, FCC and IEEE.

4.1.1 FDA standards

The Food and Drug Administration (FDA) is the most established and far-reaching shopper security office in the U. S. central government. Its inception can be traced back to the arrangement of Lewis Caleb Beck in the Patent Office around 1848 to complete compound examinations of rural items, a function that the then recently formed Department of Agriculture acquired in 1862. Despite the fact that it was not known by its present name until 1930, the FDA's current administrative capacities started with the section of the 1906 Pure Food and Drugs Act, a law a quarter-century really taking shape that restricted interstate business in corrupted and misbranded nourishment and medications. Figure 1 depicts the first stamp released by the US Post Office that recognizes this Act of 1906. Harvey Washington Wiley, Chief Chemist of the Bureau of Chemistry in the Department of Agriculture, had been the main drive behind this law and headed its requirement in the early years, giving essential components of assurance that shoppers had never realized that time.

Figure 1: The U.S. Post Office recognized the 1906 Act as a landmark of the 20th centry when it released this stamp.

The FDA and its obligations have experienced a transformation since 1906. Additionally, the commercial center itself, the sciences undergirding the items the organization controls, and the social, social, political, and monetary changes that have shaped the connection for these improvements, all have seen changes over the previous century. However the center general wellbeing mission of the organization remains now as it did then.

The FDA benchmarks are responsible for protecting the public health by assuring the safety and security of human and veterinary drugs, biological products, medical devices, our nation’s food supply, cosmetics, and other products. Personal breathalyzers fall under the "medical device" category and are required by US law to be FDA cleared for consumer sale and use. An important component of the FDA in protecting public safety is to oversee the manufacturing, performance, and safety of these devices. The FDA has issued recalls for breathalyzers that have not met FDA standards. This is a serious offense. U.S. Consumers should insist that a breathalyzer they purchase has been FDA cleared while the IEEE standards apply to the greater part of the electrical segments. All breath chemical analyzers that are permitted to be dispersed and sold in the United States for individual use must have FDA freedom. Numerous ones don't. Continuously enquire with regards to the FDA status of the breathalyzer you wish to buy.

4.1.2 CE standard

The CE checking is a key marker (yet not a verification) of an item's consistence with European Union (EU) wellbeing, security and ecological assurance orders and controls. On the off chance that your item goes under the extent of a mandate requiring CE stamping you should guarantee the item consents to the pertinent prerequisites and append the CE check before putting it in the business sector in the European Economic Area (EEA). This is similarly legitimate if your item is produced outside the EEA. Second hand items from outside the EEA that are put in the EEA market interestingly additionally require CE checking. There are many countries that require CE marking before this type of product is distributed to the public. Although the United States isn't one of them, our project will seek the CE approval because such a product will be beneficial worldwide.

The CE marking (also known as the CE mark) is mandatory on many products placed for sales on the market in Europe. The CE marking certifies that a product has met European Union (EU) consumer safety, health or environmental requirements. There are numerous "Agreements on Mutual Recognition of Conformity Assessment" between the European Union and other countries such as the USA, Canada, Japan, Australia, New Zealand and Israel. Consequently the CE mark is now found on many products from these countries. The CE marking is currently required in these 31 countries as shown below in Figure 2:

1. Austria (since 1995)
2. Belgium
3. Bulgaria (since 2007)
4. Czech Republic (since 2004)
5. Cyprus (since 2004)
6. Denmark
7. Estonia (since 2004)
8. Finland (since 1995)
9. France
10. Germany
11. Greece
12. Hungary (since 2004)
13. Iceland
14. Ireland
15. Italy
16. Latvia (since 2004)
17. Lithuania (since 2004)
18. Liechtenstein
19. Luxembourg
20. Malta (since 2004)
21. Norway
22. The Netherlands
23. Poland (since 2004)
24. Portugal
25. Romania (since 2007)
26. Slovakia (since 2004)
27. Slovenia (since 2004)
28. Spain
29. Sweden (since 1995)
30. United Kingdom (Great Britain)
31. Croatia (since July 1, 2013)

Figure 2: Countires that require the CE marking

Regularly, the CE checking is portrayed as an exchange international ID since it empowers the free development of items inside the European business sector. As per EU enactment, the EU Member States are not permitted to confine the setting available of CE checked items, unless such measures can be legitimized on the premise of proof of the resistance of the item. CE checking does not demonstrate that an item was made in the European Union. The CE check likewise is not a quality imprint.

By joining the CE checking to an item, the maker announces on his/her sole obligation that the item is in similarity with the key necessities of the appropriate Union coordination enactment accommodating its appending and that the significant congruity evaluation methods have been satisfied. Items bearing the CE stamping are designed to be in consistence with the pertinent Union coordination enactment and henceforth profit by free distribution in the European Market.

4.1.3 FCC standard

The Federal Communications Commission manages interstate and global correspondences by radio, TV, wire, satellite, and link in every one of the 50 states, the District of Columbia and U.S. domains. A free U.S. government organization supervised by Congress, the commission is the United States' essential power for correspondences laws, direction and mechanical advancement. In its work facing economic opportunities and challenges associated with rapidly evolving advances in global communications, the agency capitalizes on its competencies in:

- Promoting competition, innovation and investment in broadband services and facilities
- Supporting the nation's economy by ensuring an appropriate competitive framework for the unfolding of the communications revolution
- Encouraging the highest and best use of spectrum domestically and internationally
- Revising media regulations so that new technologies flourish alongside diversity and localism
- Providing leadership in strengthening the defense of the nation's communications infrastructure

The organization is coordinated by five chiefs who are named by the President of the United States and affirmed by the U.S. Senate. The president additionally chooses one of the magistrates to serve as administrator. Just three magistrates can be of the same political gathering at any given time and none can have a budgetary enthusiasm for any commission-related business. All magistrates, including the executive, have five-year terms, aside from when filling an unexpired term.

Most FCC guidelines are received by a procedure known as "notification and remark" rulemaking. Under that procedure, the FCC lets the general population see that it is thinking about revising or altering rules on a specific subject and looks for the general population's remark. The Commission considers the remarks received in creating last standards. The FCC issues an administrative principle under power given to it by Congress in statutes. The statutory appointment of power can extend from expansive optional power to a particular order. For instance, Congress comprehensively requires the FCC to concede show licenses in general society interest. Conversely, Congress particularly required that the FCC finish the change from analog to advanced or digital TV broadcasting by a specific date.

4.1.4 IEEE standard

The IEEE Standards Association (IEEE-SA) is a main accord building association that supports, creates, and progresses worldwide innovations, through IEEE outer connection. IEEEs unite a wide scope of people and associations from an extensive variety of specialized and geographic purposes of birthplace to encourage principles of improvement and gauges related joint effort. With community thought pioneers in more than 160 nations, they advance development, empower the creation and extension of universal markets and ensure wellbeing and open security. All things considered, their work drives the usefulness, capacities and interoperability of an extensive variety of items and administrations that change the way individuals live, work, and convey.

The Institute of Electrical and Electronic Engineers has distributed a prescribed practice entitled "Recommended Practice for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers" that characterizes worthy breaking points to flash in LED-based lighting frameworks, shown in Figure 3 below. The record, IEEE Std 1789-2015, characterizes key measurements and offers strong state lighting (SSL) item engineers direction on the best way to guarantee that LED-based items display no threat to people.

The stroboscopic effect is just one of many potential consequences of flicker. The lamp used for the image on the left does not flicker and thus the moving object is a smooth blur. Because it does flicker, the lamp used for the image on the right appears to create multiple instances of a moving object.

Figure 3: An example of a consequence of flicker

While conveying research on LEDs, it was discovered that IEEE doesn’t really have specific standards on them other than suggestions that may be beneficial to the consumer health. According to a very credible source: *“Solid State Lighting Technology Fact Sheet: Flicker,” US Department of Energy,* Lighting flash has for quite some time been known to cause diseases in people that extend from inconvenience and migraines to seizures. As Jim Brodrick of the Department of Energy SSL program noted in one of his Postings messages, the coming of high-recurrence electronic stabilizer in fluorescent lighting disposed of flash as a worry, however that LEDs with no constancy in delivering light implies that SSL reintroduced the glimmer issue.

Unmistakably, LED-based items with very much outlined drivers can convey light with no perceivable glint. In any case, dispensing with glint can expand the expense and/or size of electronic driver circuits. In fact, Brodrick noticed that miniature items, for example, MR16 substitution lights are more helpless to glint because of size confinements on the driver. Also, LED-based lights or luminaires intended to work with stage cut dimmers, for example, triacs, can display gleam at diminished settings notwithstanding while showing no risky flash at full yield.

The simply distributed IEEE standard uses the metric percent gleam, recurrence of a light source, and glint record. The creators made a diagram taking into account numerous investigations of gleam that outlines percent glint (likewise called adjustment) with respect to the working recurrence of a light source exchanging on and off. The diagram recognizes a sheltered area furthermore an okay locale for glimmer.

The standard report eventually gives a condition through which most extreme percent gleams can be resolved. You increase the recurrence of a light source by 0.08 and round up to compute the greatest rate. At 120 Hz, 10% most extreme gleam is admissible. There are extra conditions that can be utilized to decide most extreme glimmer at low working frequencies.

After stumbling upon a paper, “Implementation of a Colorful RGB-LED Light Source With an 8-bit Microcontroller” by Yueh-Ru Yang at the Graduate Institute of Electro-Mechanical Engineering, Ming Chi University of Technology, Taipei, Taiwan, describes the implementation of a colorful RGB LED light with an 8-bit microcontroller. It makes the decision on which color LED to use become less complicated. Yang starts by clarifying that shading is comprised of Hue, Saturation, and Brightness. He goes ahead to clarify that the three hues, Red, Green and Blue, are viewed as opposite three unit vectors and structure a shading space. Since the human eye has diverse sensitivities for these hues, the unit estimations of these vectors are, along these lines, not equivalent to each other, which permitted the creator to clarify how the distinctive blends of the hues could set up the standard mix for white, and different hues could be delivered from the mix of this white with the fundamental hues

While the LED is lit up, the temperature in the gadgets ascends with time until parity is reached. Yang clarifies how typically the shine of the LED diminishes with the ascent of its temperature, yet this time is not equivalent for the three hues (Red, Blue and Green). So as to remunerate the shading deviation that outcomes from the adjustment in temperature, the LED lights will require temperature pay, and the separate amounts were controlled by experimentation, Yang's taking after diagram demonstrates the test outcomes.

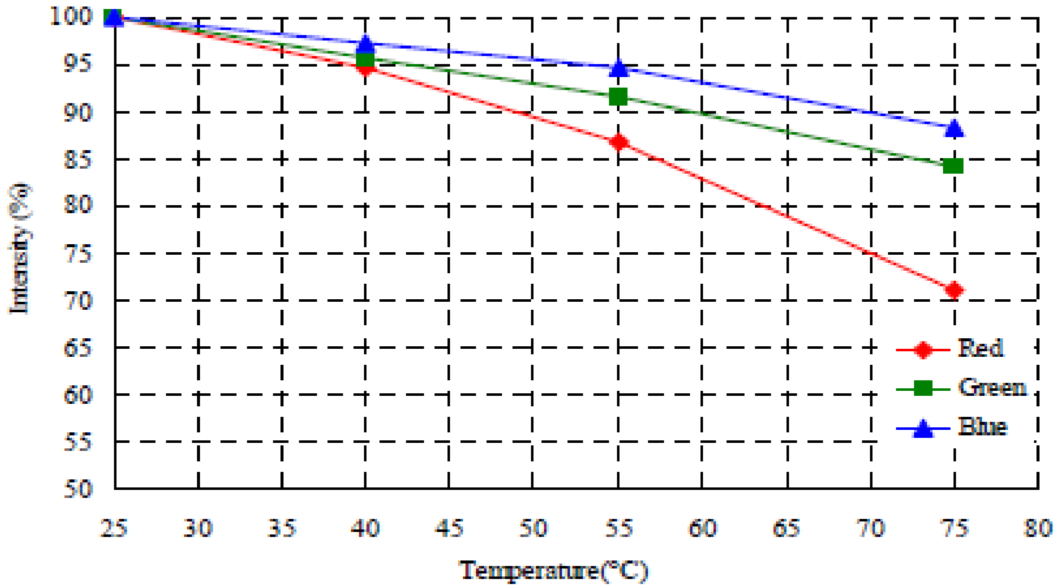


Figure 4: Relationship between brightness of LED and temperature

In conclusion, the test demonstrated that the red light diminished in power more unexpectedly than green and blue as the temperature rose. In this paper, Yang controlled his distinctive hues freely by isolating the three into strings and utilizing separate PWM streams and performed three separate tests to survey the viability of his configuration, temperature control on the colors (by changing the PWM estimations of Red, Green and Blue), shading test, and temperature encourage forward remuneration for the three shading yields.

4.2 Realistic Design Constraints

In this section, we explore the possible constraints that we need to be aware of while designing this device. This takes a great deal of research due to the fact that we will need to make any adjustments for our original design according to the possible constraints that follow.

4.2.1 Parts, Component and Testing Constraints

There are numerous constraints for the diabetic breathalyzer as the device is going to be tested on humans. As its clearly known to everyone who are enrolled at the University of Central Florida whether it's by employment or degree seeking purposes, it is prohibited that any lab project or course assignments to be tested on human beings nor animals on campus grounds. That in itself is already a disadvantage working against the display or presentation part of this blood glucose breathalyzer device design. Also, being that there's close to nothing similar to this project, it will be quite difficult to come up with a standard as to how accurate the design should be. Therefore, it is not expected to be as sensitive as some of the top pricking devices that already are on the market but will be easier to utilize. The system is designed with cost, well-being, and effectiveness in mind. We are creating a device to that has to be very user friendly and easy to carry around without putting the user in harm's way. It also had to be designed give fast readings at a fast rate.

In designing the Blood Glucose Breathalyzer device, the flowing parts are recommended: 2 Volatile Organic Compound (VOC2) Sensors, Temperature and Humidity Senor, Microcontroller, Bluetooth Module, PCB, and Android application. Explanation and description of each part is listed below:

- **VOC2**: The sensing element is comprised of a metal oxide semiconductor layer formed on an alumina substrate of a sensing chip together with an integrated heater. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal, which corresponds to the gas concentration.

- **Temperature and Humidity Senor:** We have options to choose between the DHT11 and the DHT22. The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

- **Microcontroller:** The ATMega328p is a standard ARM microcontroller from Atmel that is used in many embedded systems especially at the hobby level. It supplies ample general input output pins while also having analog pins which are a necessity to run the VOC sensors. It functions at a standard 5v and is low power especially during sleep mode, which will be used during standard operation. The ATMega328p also has serial communication capabilities which will make wireless communication very simple.

- **Bluetooth Module:** The Bluegiga WT11 module on the Arduino BT provides Bluetooth communication with computers, phones, and other Bluetooth devices. The WT11 communicates with the ATmega328 via serial (shared with the RX and TX pins on the board). It comes configured for 115200 baud communication. The module should be configurable and detectable by your operating system's Bluetooth drivers, which should then provide a virtual communication port for use by other applications. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board over this Bluetooth connection. The board can also be reprogrammed using this same wireless connection.

- **Android application:** This is an application that's going to provide user interface.

- **PCB:** A printed circuit board (PCB) is the board base for physically supporting and wiring the surface-mounted and socketed components in most electronics.

  In applications where fine conductive traces are needed, such as computers, PCBs are made by a photolithographic process, in a larger scale version of the way conductive paths in processors are made. Electronic components are typically placed by machine onto a finished PCB that has solder dabs in place. The PCB bakes in an industrial oven to melt the solder, which joins the connections. Most PCBs are made from fiberglass or glass-reinforced plastics with copper traces. PCBs can be single-layer for simple electronic devices. Printed circuit boards for complex hardware, such as computer graphics cards and motherboards, may have up to twelve layers. PCBs are most often green but they can come in any color. Other methods of PCB manufacturing include silk-screening and CNC-milling. This part is the heart of the device. It's the biggest circuit in our design and numerous companies can fabricate this part for the device.

- **Aluminum Casing:** This is the enclosure of the entire system to keep it intact and protect it.

4.2.2 Economic and Time constraints

The budget of the breathalyzer device will be split between team members, an amount that was approximately $350.00 in which it is more likely to be less than $300.00, which is very achievable, is the predicted cost. The most expensive part of the design is the PCB fabrication, which is roughly $200.00. There are many competitive companies who fabricate PCBs in the near-by radius, which would definitely help with the time despair that's playing a huge factor throughout the process of this project. Some of the notable manufactures are; Infinity PCB, Inc., TeligentEMS, Elreha Printed Circuits Corp, Logic Systems Corp, Intersil Corp, Ultra-Tech Enterprises Inc, Act Usa Intl Corp, Adic Inc, Advanced Design Svc Inc, and many more. But the company that will be used In the need of manufacturing the PCB is Eagle Electronics. It is highly recommended by other individuals who've been through the same process. Based on the intakes from others who have the history with Eagle Electronics, they provide the very best quality, delivery and budget combination.

Due to the fact that this is a Senior Design Project, time is very limited and it is well understood that the workforce is less patient and the workload will be quite similar to this one but with less time. A maximum of 2 months is given to complete the design of the Blood Glucose diabetic breathalyzer device. After the designs and parts are carefully chosen, then a very short span of 4 months will be given to buy the parts, accumulate them collectively to make the device and, test it. If any trials fail then we will have to apply any modifications so that the scheme is up-and-coming. The objective of the design is to also be time efficient. To cope with that, it is essential we stay ahead of the time given to complete this project 2 weeks before presentation. The ways to achieve that is to be self-starters and over achievers. We will be ordering parts of the break from summer to fall to implement them during the fall semester. It is also important that the coding aspect of the project is completed during the break because it's probably the most important part of the project. We can get to the assembly and testing junctures sooner so that the project can be properly sense the readings and we can determine the failures and if any of the parts need to be replaced for a better performing part or for a small alteration in design as well so a two week period has been set aside after testing to make sure that the breathalyzer will be ready to be utilized.

4.2.3 Environmental, Social, and Political Constraints

At some point in all of our lives, we all have seen litter. Furthermore, I can guarantee it that you witness or see litter every single day if you step foot outside. According to the National Geographic Society (NGS), because of this litter, the litter that is just a part of everyday life, there is an island of trash two times the size of Texas floating in the Pacific Ocean. Just think about that. You know the size of Texas and can you picture the 2 Texas states in the middle of the Pacific Ocean. One can only imagine the harm this filthy side is doing to the life in the middle of that Ocean and eventually the effect it will have on the lives on land. Another more obvious reason that littering is immoral is the eyesore, unsanitary looks of it. The trash can contain or carry diseases as well, that can be passed between animals that eat it. If trash is sitting in water, the water becomes contaminated, and when the water evaporates whatever was in the trash in now in the air.

According to Keep America Beautiful (KAB), the United States of America devotes 11.5 billion dollars on litter clean up every year. The reason for this littering background knowledge is because people don't realize how old-fashioned pricking of the finger to measure blood glucose level can actually add to the littering problem that the United States of America is facing today. From experience of living with individuals with finger pricking devices to monitor their blood glucose level, and also visiting friends with the same situation, it is almost inevitable to not find alcohol wipes with blood stains around the house and if they're laying around the house, imagine the people who operate in their car. Imaging how they simply tossed it out of their window with the blood residue on it flying around and landing wherever the wind takes it. Sound dangerous enough? Well, the Blood Glucose Breathalyzer device would definitely help start to decrease that. It will have a sensor that will be reusable for the User to blow into. All you have to do is wipe it after it is utilized. The best part about it is that now you don't have to expose others to your blood although it's very minimal but any amount of blood coming out the body is enough to contaminate the entire body and others around it.

4.2.4 Ethical, Health, and Safety Constraints

Here are the basic constraints for basic electronics:

- Safety of workers and consumers
- Safety of the public
- Noise causes hearing loss
- Hazardous materials and environment for workers
- Products require the use of radioactive materials
- Products use materials that have better appearance but are toxic
- Products for infants/children require special safety considerations
- Design of a control system with acceptable stability margins for machinery where safety is of concern.

It is very safe to say that anytime someone has blood come out their body is very unsafe and risky. It does not take much for someone to be infected because we are all exposed to contaminants anywhere we are on this planet called earth. It is also very likely that pricking fingers is a procedure that can be done incorrectly.

An underappreciated risk of blood glucose testing is the opening for exposure to blood borne viruses (HBV, hepatitis C virus, and HIV) over filthy gear and materials if devices used for testing and/or insulin administration (e.g., blood glucose meters, finger stick devices, insulin pens) are shared.

Outbreaks of hepatitis B virus (HBV) infection associated with blood glucose monitoring have been recognized with increasing uniformity, predominantly in long-term care settings, such as nursing homes and assisted living conveniences, where residents often necessitate support with observing of blood glucose levels and/or insulin administration. In the past decade alone, there have been at least 15 outbreaks of HBV infection linked with providers failing to follow basic principles of infection control when supporting with blood glucose monitoring. Due to under-reporting and under acknowledgment of critical infection, the number of outbreaks due to unsafe diabetes care practices identified to date is likely an underestimate.

It is specifically stated by the Center for Disease Control and Prevention (CDC) that “Although the majority of these outbreaks have been reported in long-term care settings, the risk of infection is present in any setting where blood glucose monitoring equipment is shared or those assisting with blood glucose monitoring and/or insulin administration fail to follow basic principles of infection control. For example, at a health fair in New Mexico in 2010, dozens of attendees were potentially exposed to blood borne viruses when finger stick devices were inappropriately reused for multiple persons to conduct diabetes screening. Additionally, at a hospital in Texas in 2009, more than 2,000 persons were notified and recommended to undergo testing for blood borne viruses after individual insulin pens were used for multiple persons” Unsafe practices during assisted monitoring of blood glucose and insulin administration that have contributed to transmission of HBV or have put persons at risk for infection include: using finger stick devices for more than one person, using a blood glucose meter for more than one person without cleaning and disinfecting it in between uses, using insulin pens for more than one person, and failing to change gloves and practice good hand hygiene between finger stick procedures.

4.2.5 Technical Constraints

Current VOC sensor technology takes advantage of certain properties of metal-oxide when exposed to particular gasses. Currently, commercial sensors available rely on variations of aluminum oxide and tin oxide. Both of these metals have different sensitivities and function over different ranges of gas concentration. The current technology, however, is not intended for the uses we hope to exploit. The sensors available for purchase today are meant to monitor dangerous gas level predominately in an industrial setting. Typically they will be hardwired into the building and run continuously, sounding an alarm when levels of dangerous gases get too high. Because of this, the sensitivity and exactness of the measured concentration was not very important. They aren’t meant for reading exact values, but instead to sound a warning when an arbitrary level is met, in the sensor’s case, an arbitrary resistance that indicates a dangerous level. Because of this, it will be tricky to get accurate readings in exact parts per million from our sensor.

Current research is attempting to further the use of metal-oxide sensing technology by looking for different metals to use. One metal-oxide that is showing promise is titanium oxide, which has a much greater sensitivity than other metals. With this in mind, current research is going into how to utilize titanium-oxide for bio-analysis, specifically breath analysis. If this technology was available, it would be the preferred sensor to use for our project. But it is currently still being researched and therefore is unavailable for commercial use. If the technology is researched further, some of the constraints seen with the current non-breath analysis sensors could be alleviated. There could be better designs that use less power and have proper sensitivity since they will be designed specifically for analysis use and not general dangerous gas detection.

Because of the fact that these sensors are intended to be hardwired to power and run continuously, they are not very energy efficient. The biggest power drain being the heaters used to get the metal-oxide in the sensors to a high enough temperature that the resistance is affected by gases in the air. This wouldn't be an issue if the heaters could only run when the device is in use, which would be periodically. If we could get reliable readings with no warm up time our energy needs would be far lower. However, this is not the case and these sensors do require many hours of warm up to full stabilize. It is important that the readings do stabilize since when we take the readings of the analog pins, we will be looking for discernable change. If we cannot differentiate the change due to VOC and the changes due to general instability, our data will be useless. Due to this fact it will be necessary that the sensors be given power continuously.

With continuous power comes the restraint of having a portable power supply keep enough power to run the sensors continuously while not dying. The sensors themselves will draw about 180 mA of current at 5V. The problem here is how to balance power capacity while also trying to minimize cost and weight. Typically, as batteries gain capacity they also become more expensive and bulky. Since our device is handheld, the portability is very important. While it would be great to have a battery that could run the sensors for days at a time would be a great solution, the feasibility just is not there. Because of this, we have decided to try to get a battery that will run the device for an entire day of use, or about 14 hours. This time was chosen because it would be ideal to be able to start using the device when you wake up and then plug it in to charge at night, similar to how cell phones are currently treated. In order to have this goal be met we must have a battery with a capacity of at least over 2500 mAh since that would run the sensors, at 180 mA, for just about 14 hours. This of course is assuming ideal consumption, which never happens practically, so our needs would be above the 3000 mAh range. This is not including the extra power needed to read and end data to the application, but those power needs are much smaller and for less time than the consumption of the sensor heaters, and can therefore be neglected when considering daily power use.

Another issue that comes about that was touched on briefly before is the simple fact that by creating our own equations based on the plots provided form the datasheets could cause discrepancies in what our data is interpreted to be. Since all the equations used were gathered by simply looking at the graph and getting a set of apparent data points, and then putting those into a spreadsheet and getting a line equation, there will be issues in accuracy. Hopefully this issue is mitigated by the fact that every data reading will use the same few equations, which would provide at least consistency to our results. Even if those results aren't exactly accurate to the precise concentration value, by having consistency we can still understand when levels are noticeably high and low. While we hope to get exact levels, and therefore exact levels of the ketones in question, our device would still function as intended even if we can only detect larger spikes. These spikes could still then be attributed to dangerous levels and therefore functionality is maintained.

One final issue could be verifying the assumption we are making, that measureable ketone levels in the breath are directly related to a person's blood sugar level. There is research being done to verify this claim, but it can be seen as preliminary. Our assumptions are based in biology and therefore should hold credibility, whether there is exact research to support it. We know that the body creates ketones in times of need and those ketones can break down into a measurable type of acetone. The problem arises when we try and further our assumptions. To say those levels of acetone that may be present in the breath can be directly related to blood sugar levels is one that will require great testing on our part. There is some research available to give us an idea of where to begin, but it will require us to have numerous trials comparing the breathalyzer results to other forms of ketone testing whether it is through blood analysis and urinalysis. Doing this will definitively show the correlation we hope to find. If for some reason that correlation is not found, our device would still function properly at detecting acetone and therefore would function. However, this would remove a lot of the desirability in having such a device. With this in mind, it is very important that we create a functional and consistent correlation between either blood sugar or ketone levels in the body and the amount of acetone our device can detect.

4.2.6 Manufacturability and Sustainability constraints

Our design fits under the following categories:

- Designs with an impossibly small manufacturing tolerance
- Designs with a required highly accurate first natural frequency
- Designs with an impossibly high stiffness
- Designs with a zero-friction contacting surface
- Designs with a no- mass part
- Perpetual machine
- Machines without vibration
- Can the proposed material be welded if welding is the proposed assembly method?
- Is the product's surface paintable if it is designed to have an artificial color?
- No gravity for manufacturing process in space

- Availability of chosen material
- Titanium alloy and ceramics require special cutting tools
- Design of a control system which is physically realizable with manufacturing constraints such as amplifier saturation and bandwidth

5.0 Project Hardware and Software Design

Overall Block Diagram

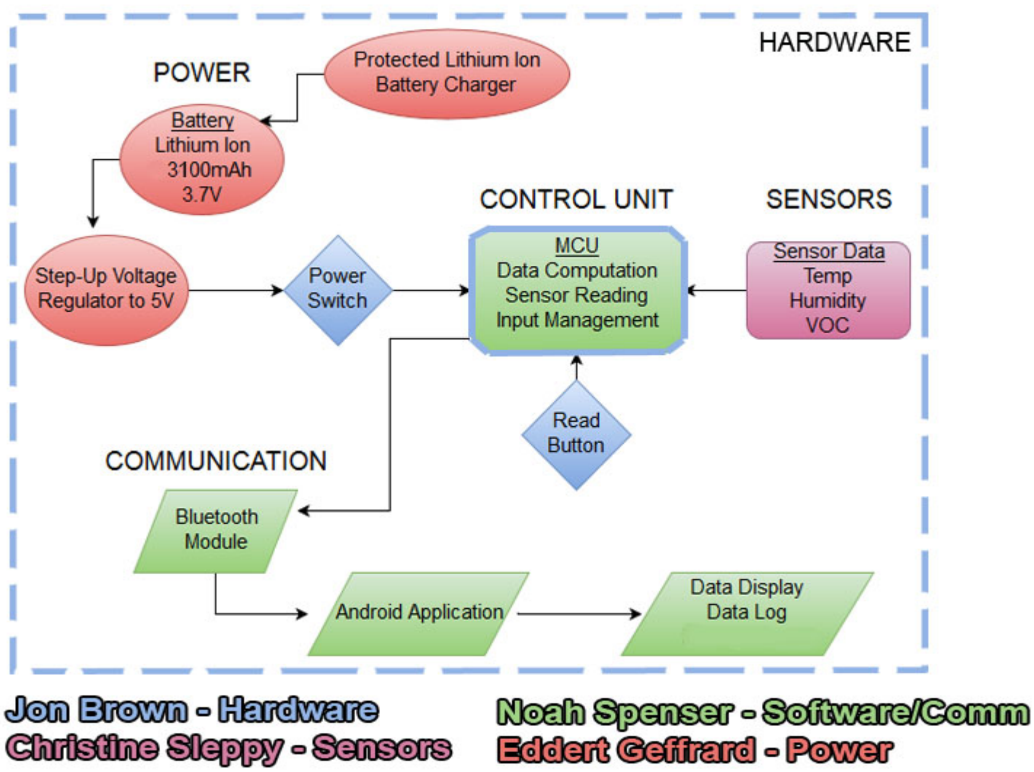


Figure 5: Block diagram of necessary hardware and responsible team member.

5.1 Initial Hardware Design

Hardware design holds the majority of this project design. We discuss our ideas for our original hardware design in this section including power, control unit, communication, and sensors.

5.1.1 Power

Continuous power will need to be supplied to the unit in order keep the sensor heaters running. This is to avoid long warm up times in order to get the sensor to stabilize. Through testing we may find that power can be supplied in increments rather than continuously. This would decrease the power used and rely on the fact that the heaters themselves cannot cool quickly. The goal will be, through testing, to find the lowest amount of power needed to maintain sensor stability. Until this is understood, the battery used will need to be sufficient enough to run the heaters for a minimum of 12-14 hours in order to provide functionality throughout the day. Frequent charging would also be necessary.

The type of battery for this unit will need to be lightweight, long lasting, and rechargeable. These specifications create a very specific need. In order to achieve the needed specifications, mainly the rechargeable nature of the battery, a choice must be made between NiMH and Li-Ion type batteries. Due to our continuous energy needs, the more powerful Li-ion battery will be one we choose. There is also the choice of lithium ion polymer battery, which provides flexibility and a further lightweight design. These, however, can be much more expensive, and would not add much functionality to the unit. Because of this, a lithium ion battery will be the choice for this unit.

The supply voltages will vary between 3.3 and 5 volts. Most of the operations on the PCB will be able to run at the 3.3v level. This includes the data portion of the sensors and the wireless module. The 5v will be needed to power the sensor heaters. These voltage levels will be achieved using a battery source of around 7.4 volts, provided from a lithium ion battery. Due to the continuous nature of the power, the voltage regulator used to deliver the 5 and 3.3 volts may need to be a higher efficiency buck regulator. Continuously drawing a few hundred milliamps while having a voltage difference of 2.4 volts in the regulator will cause large power loss. A highly efficient step down regulator would help this issue.

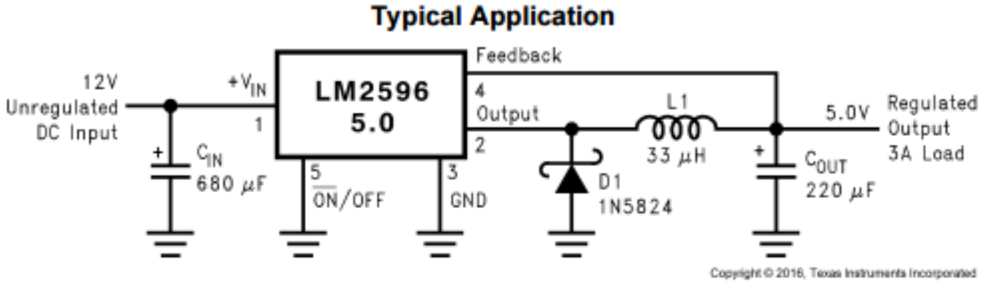


Figure 6 : Typical circuit configuration of LM2596 regulator. (42)

As seen in Figure 6 uses a LM2596 step down converter from TI to create an output of 5V. Since the majority of power drawn from the device will be at the 5V level to the sensor heaters, we only need a high efficiency regulator for the 5V. The 3.3V supply will be used sparingly in comparison, and therefore can be easily regulated using a linear regulator since the current draw from the microcontroller is too small to warrant the use of a high efficiency one.

Another option would be using a standalone regulator module. An example of this would be the RECOM R-78W5.0-0.5 Step-Down Regulator. Using this would cost more money, but would allow for a more simple design since it is all incorporated within a single module. By using a premade part, we can be sure to have the optimal efficiency in powering the unit.

Device	Power Rating (Peak)
WSP Heater/Circuit	< 660 mW
TGS Heater/Circuit	<330 mW
DHT Temp/Hum.	<1 mW
HC-05	<82 mW
ATMega328p	<10 mW
RBG LED	<10 mW

Table 5 : Specified power ratings from datasheets. (37)(38)(39)(40)(41)(42)(43)

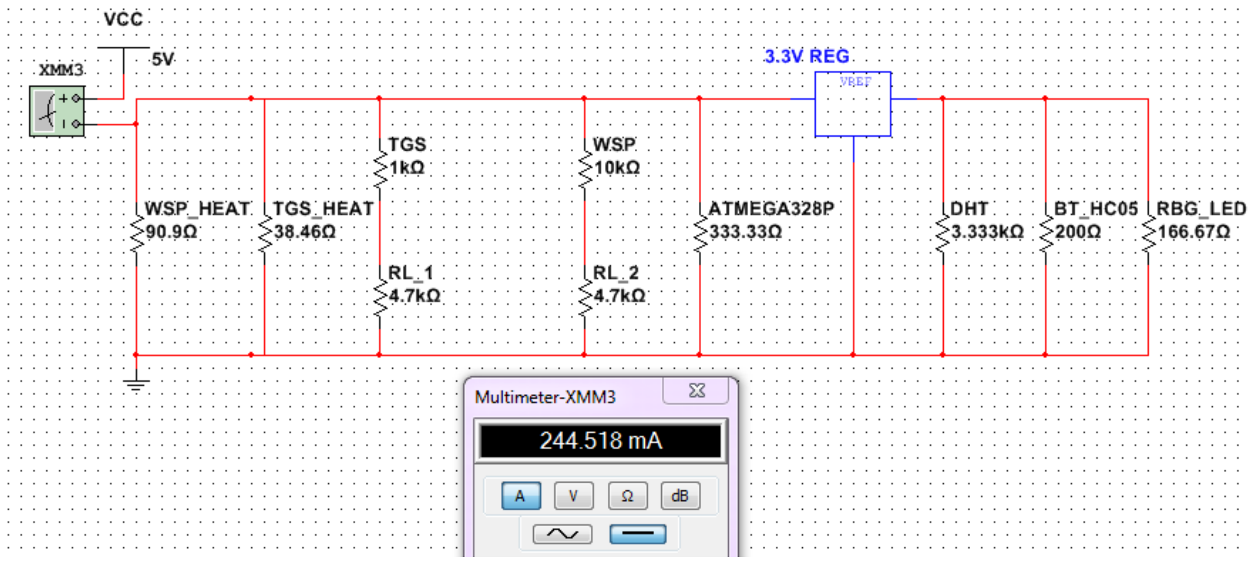


Figure 7: Original Multisim simulation showing overall current draw of device.

Figure 7 shows a Multisim diagram of the current draw the device will experience. Using the power ratings found in the referenced datasheets for each individual component at max conditions, a value was found for max current draw during peak usage. This value is just under 250 mA. During down time when the MCU is on standby and only the heaters of the sensors are running, the current draw will be just above 180 mA.

In order to determine what capacity our battery will require, we must consider the current draw of the device. Since the majority of the time the device will be in sleep mode and only powering the sensor heaters, we can look at just the current draw of those. All other components will be used sparingly and have a much smaller impact on power consumption than the sensor heaters. Both sensors will combine to draw about 180 mA of current during down time. If we are to maintain our initial requirements of a device that will hold a charge for 12-14 hours, ideally our battery would have a capacity of (180mA)*(12-14 hours) = 2160-2520 mAh. This would be the required capacity needed if our conditions were ideal. Unfortunately, many other factors can play a role in the lifetime of a battery including but not limited to temperature, current draw rate, and charging tendencies. Because of this it is recommended to apply an allowance for these unknown factors. We will use the assumption that 75% of the capacity will be used for our purposes as a way to assure the necessary leeway to achieve our requirements. This factor is chosen simply as precaution for this specific project and does not hold a particular value. Our value of 75% is to assure that our needs are more than met. The equation for this will look as follows:

$$Battery\ Capacity = Consumption\ (mA) \times Duration\ (h)] \div Alowance\ Factor$$

With the consumption figured to be about 180 mA and the desired duration of 12-14 hours along with an allowance factor of 80%, our new desired range of battery capacity is 2880 – 3360 mAh. So our needs require a power source with at least a capacity inside this range which is why we selected a battery with 3100mAh capacity.

5.1.2 Control Unit

In order for functionality to be controlled in the unit, a microcontroller is necessary. A field programmable gate array (FPGA) could be used to run many simple processes. However, with the current design, a more easily configurable microcontroller is the obvious answer. By being able to stay in an Arduino C-based software environment, we can have an easier time getting the exact functionality our objectives require. Our design requires that there be enough general-purpose input output (GPIO) pins in order to maintain control over the various sensors and the communication module. The sensors also require a dedicated analog pin in order to read the resistance of the sensors, which is the pertinent data needed from them. A low power MCU is also needed since most power will be dedicated to the sensor heaters. However, the current market of MCU's that would cover all requirements all tend to function at a low enough power consumption that relative to the sensors, it will not be a large concern.

Due to familiarity and a hope to maintain consistency, the Arduino IDE will be used to program our microcontroller. This gives us enough reason to limit our search for a MCU to the Atmel chips regularly used by Arduino devices. There are a few options available when considering a microcontroller even within this specific family. The most commonly used chip when it comes to Arduino use is the ATMega328 chip that can be found in many Arduino boards.

Our search can be narrowed down to three particular chips that all provide benefits while having clear disadvantages. The breakdown of the three chips can be found in the table below.

Chip	**ATtiny85**	**ATMega328p**	**ATSAMB11**
Cost ($)	0.77	1.80	9.81
Flash (kB)	8	32	256
Pin Count	8	32	48
Max I/O Pins	6	23	30

Built in Bluetooth support	No	No	Yes

Table 6: Comparison of researched MCUs. (41)

The first chip looked at is the ATtiny85. This chip is similar in structure to the ATMega328 but it is much smaller and uses less power. This smaller size and therefore lower cost comes at a price however, since there are limited GPIO and the flash storage size could be limiting at only 8kB. This would force us to create an extremely streamlined code, which may be problematic.

For an extra dollar in cost the next chip looked at becomes available. The ATMega328p is the familiar choice and used in many embedded projects. The flash memory size is much more freeing at 32 kB which should be plenty of room for the code we will need. With 23 I/O pins there will also be no shortage in availability.

The final chip is the most robust and therefore the most expensive. At an increase in price of almost 9-fold, the ATSAMB11 is rather expensive. The extra flash memory and available pins would almost certainly be unnecessary with our current project. The main positive attribute for this chip is the built in Bluetooth module that we could use for communication. This would provide a convenient single package for control and communication. The benefits would be nice, but the requirements for our wireless communication are small since the distances needed to be covered are limited. Because of this, a cheaper standalone Bluetooth module could be acquired for a cheaper total price than purchasing the ATSAMB11.

After considering all options, the choice for microcontroller will be the ATMega328p. Familiarity, low cost, and pin versatility are the reasons for choosing this chip. The Arduino bootloader can easily be installed on the chip with still plenty of room left over for our code. This will allow for easy use of the Arduino environment and libraries. Due to the all electrical engineer nature of our group, this simplicity in code is required.

The package we will use will require very little space while still providing all 32 pins to be used. The overall size of the chip will be about 5x5 mm. This will easily fit on the eventual PCB and therefore spacing should not be an issue.

Much of the functionality of the chip will be unused since our needs are specific, but with the cost so low it isn't worth it to go with a smaller unit. The AtMega family of chips run at a low enough power and come at a low enough cost that robustness does not need to be sacrificed. (41)

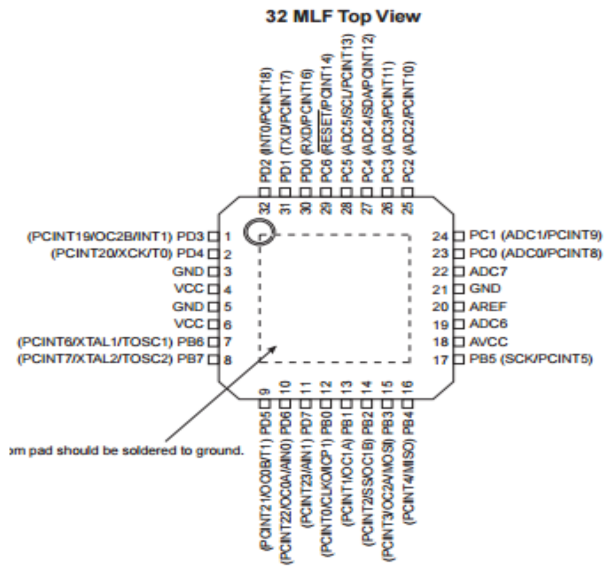


Figure 8: Drawing of MCU package we will use. (41)

Picture from ATMega328p Atmel Datasheet.

5.1.3 Communication

The communication requirements for this project are limited. Since the distance between unit and the mobile phone used as the output display will typically be small, the coverage of the wireless signal from the unit does not need to be large. There are currently low power 2.4 GHz transceiver options on the market, the most well known being the ZigBee wireless modules. These are great for power consumption since they cover only a small area. This type of setup would be ideal, however interface with a mobile app must be considered. There is no established way for a phone to receive this signal. Currently, most phones can receive a Wifi signal, Bluetooth signal, and Near-Field Communication (NFC) signal. Our search for a proper component will be limited to these three.

NFC is a decent choice since distance is not a huge factor in our design. However NFC requires the receiver to be almost touching the transmitter. While this could be achieved in most instances, the limitations it would create would cause usability of the unity to go down. The next option is Wi-Fi, which is a well-known and secure form of communication. Coverage distances of Wi-Fi are much larger than Bluetooth with a typical Wi-Fi module being able to cover around over 40 meters indoors. This compares to Bluetooth communication, which can cover a range of 5-30 meters. The extra robustness found in Wi-Fi typically comes at a cost of using more power. This fact alone makes Bluetooth a more promising option since the extra range is not needed. Because of this, Bluetooth communication will be the type of communication used for our unit.

There are many options for Bluetooth communication. They all vary in price and some options, as previously seen, are built into the MCU. Since our requirements call for lower power and a smaller range, a low cost module can be used. The module we require only needs to take in basic serial information and broadcast that information a distance of as little as a few feet. By choosing a standalone module we can have more flexibility with placement and PCB organization.

There are various options that could handle what our unit will require, but again familiarity will be a helpful deciding factor for this decision. The HC-0x series of Bluetooth modules provide us with low enough power, (8-25mA) and are also low in cost, (2-3$). They can cover a range of about 10 meters like most modules, which is more than enough to handle our needs. The HC-05 module is going to be the component of choice here due to its low cost and availability.

The connection of the HC-05 module is very straightforward. Serial communication connects to the RX and TX pins on the MCU. This along with a 3.3V source and ground are all that is needed for communication. By having the module be separate from the PCB, space allocation will be easier since we can place it anywhere inside the unit. This also allows for easier testing since the entire module can be removed for troubleshooting. (39)

5.1.4 Sensors

The most integral part of the Breathalyzer unit is the Volatile Organic Compound sensor. These devices incorporate the use of a heated metal-oxide element that has a variable resistance. The resistance is dependent on the concentration of certain carbon-based chemicals called Volatile Organic Compounds. The most notable of these are methane, ethanol, carbon monoxide, and the compound this project is concerned with, acetone. The varying resistance caused by the presence of these compounds is directly correlated to its concentration. It is from the acetone concentration data that the glucose estimation will come from.

Sensor	TGS822	WSP2110	MQ-3
Concentration Range (ppm)	50-1000	1-50	10-200
Acetone directly referenced in datasheet	Yes	Yes	No
Typ. Heater Power Consumption	660	300	900

(mW)			
Sensing Material	Tin-oxide	Aluminum-oxide	Tin-oxide

Table 7: Comparison of researched VOC sensors (37)(38)(44)

Due to the relatively limited data available about this idea, there will need to be experimental testing in order to find true correlation between acetone and glucose. This will require us to initially cover a large range of concentration. Because of this, the use of two sensors will probably be necessary. Three available sensors were found and researched to see what would best fit our design.

Due to its larger power consumption, smaller range, and insufficient documentation, the MQ-3 would not be a suitable sensor for this project. (44) A combination between the two remaining sensors would cover a very large range allowing us to get as exact levels when detecting acetone. There is a possibility that, through testing, we find that acetone levels typically fall in a range that is covered by a single sensor. If this were the case, the other could be scrapped in order to free up energy. The current research is inconsistent in saying what a typical acetone concentration could be since it can vary from person to person. An expected range would be from 1ppm to over 1000ppm. This would require the use of both sensors as the WSP2110 would be used to detect levels in perfectly healthy individuals, and the TGS822 would be used to detect levels in individuals with large spikes. This is going to be the configuration going forward, using both sensors, and will be tested greatly during prototyping. (37)(38)

Each sensor works by utilizing the analog inputs of the ATMega328p chip. We are able to measure the voltage level at a certain pin by way of internal voltage divider inside the ATMega328p. The sensor itself makes up one of two resistors in a voltage divider circuit outside of the MCU as seen below:

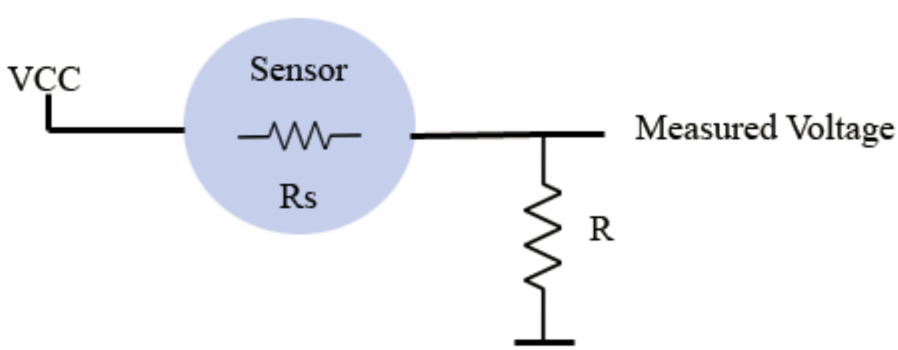


Figure 9: Original drawing of circuitry representation of VOC sensors.

We can find the value of Rs and therefore the acetone concentration by using the following equation:

$$Rs = R(\frac{VCC}{MV} - 1)$$

Where Rs is the sensor resistance, R is implemented resistor (typ. 10KOhms), VCC is the supply voltage, which should be 5V, and MV is the measured voltage that the analog pin of the microcontroller takes in and converts to Rs value.

One issue with the hardware implementation of the sensors is the current draw of the heaters. The metal-oxide elements of the sensors require that they operate at a very high temperature (about 300C), and also have long warm-up times. This will constitute the majority of the power consumed. In order to use the device multiple times throughout the day, a near constant supply of power would be needed, in order to avoid having to wait long periods of time after start-up. The data sheets to these sensors are unclear on actual warm-up time in order to get a stable enough resistor value to use. Each state, however, that multiple days were used to initially test the sensors. This initial 'seasoning' may be necessary, but warm up times can be brought down greatly after testing and calibration.

A manual switch could be used to operate the heaters, which would supply a regulated 5V directly from the battery. The limitations of this would be that there could be no control of that heating element from the phone app, which could be beneficial. The microcontroller itself cannot directly power the sensors since there is a limit of 40mA of current for each pin, and these sensors would require upwards of 250-300 mA. A solution to this would be to use a NPN transistor as a switch. The Arduino pin would 'turn on' the switch and allow for the heaters to get power from the regulated 5V at a higher current level. The configuration would look as follows:

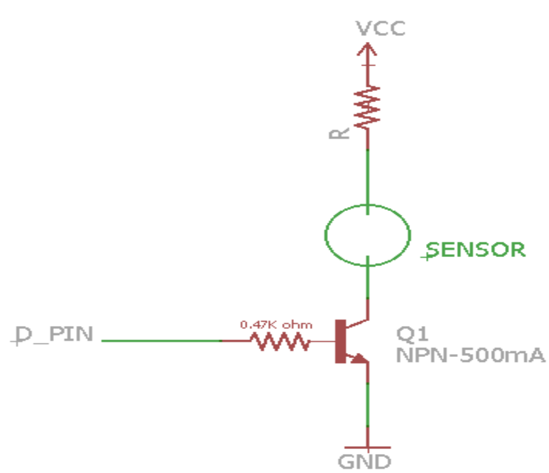


Figure 10: Eagle schematic configuration of transistor as a switch.

This setup would allow the MCU to turn on and off the heater power supply which could allow for better power management and would allow for wireless control over the heaters. This would require the MCU to maintain a low power standby mode between uses, but would be easily implemented through the software.

With simplicity being key with finishing our project, the decision to simply use a manual power switch located on the outside of the device was made. Further versions of the breathalyzer could allow for switches to be implemented directly on the PCB and could then be controlled by the phone application.

5.1.5 Physical Design

The important factor with this project when considering physical design is portability, since the device will be used repeatedly throughout the day, and should be able to be carried with little inconvenience. One factor in portability is the weight of the unit. To achieve this, the casing will be made of a lightweight metal to ensure durability while still conforming to the lightweight requirement.

The mouthpiece in which the user breathes into will be durable plastic tubing that will be detachable. By allowing the user to remove the mouthpiece, the device will be easier to transport. Due to the fact that germs from the users mouth could build up on the mouthpiece with repeated use, implementing the use of antimicrobial plastic for the mouthpiece could be necessary. Being able to remove it from the unit will allow the user to clean it as well.

Since the device relies on the analysis of exhaled breath, it is important that once the inner chamber fills, it remains as stable as possible for as long as possible. This will assure that the data received from the VOC sensors is reliable. Air must be able to enter the chamber from a specific point, but be prevented to exit once the exhaled breath is complete. The first design of the outside included simply a ventilation hole that is controlled directly by the user’s finger. This is a simple and easily accomplished goal, however the simplicity is at the cost of efficiency. Another option would be to have a vent that is controlled directly through the read button. This can be achieved by having a mechanical push button that opens a vent upon being pressed and closes when depressed. The mechanical button will also have to have a way of incorporating electrical leads that will notify the MCU of a read being attempted. One final approach to assure the gases being analyzed are contained completely would be to implement a check valve on the side of the unit. A small, two pronged ball spring check valve could be connected to the proposed air vent. This would allow the user to breath into the device, while not letting any air out. By sealing all other parts of the device that could let air out, we can assure the data gained form the VOC sensors is as accurate as possible. That possible configuration could look like the Figure 11.

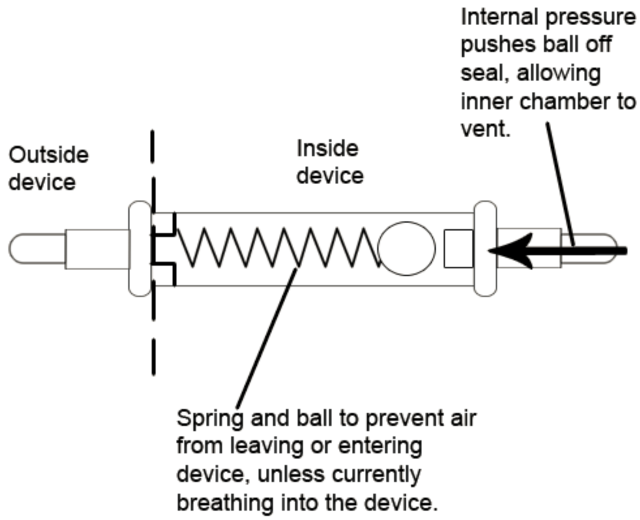


Figure 11: Original drawing showing mechanism of a ball spring check valve.

The main unit will have a rectangular prism shape with the dimensions of 120x65x40mm.This will give enough room to house the VOC/temperature sensors and the control PCB. Enough room will be needed to give space between the VOC sensors and the temperature and humidity sensor so that there is not any interference from the heat given off by the VOC sensors. A pushbutton will be placed on the side to be pressed by the user's finger while holding the device in one hand. This button will be the read button which when pressed, will wake up the MCU and begin transmitting data to the Android application, once the sensors have stabilized.

A RGB color LED will be placed near the mouthpiece in order to inform the user when the reading is stable and potentially when the battery is low. The LED will be able to light up to three different indication colors. If the LED is blue, that means the read button is being pressed, but the sensors are still stabilizing. Once the LED is showing green it means the device is stabilized and data can be transmitted to the Bluetooth module. During this time the user should fully exhale into the devic. If the read button is not being pressed the LED will remain off unless there is indication that the battery has fallen below a desired voltage level. This would indicate that the battery is low and the device needs to be recharged. If this is the case the LED will shine red. The overall design can be seen in Figure 12, while the dimensions are shown in figures 13 and 14.

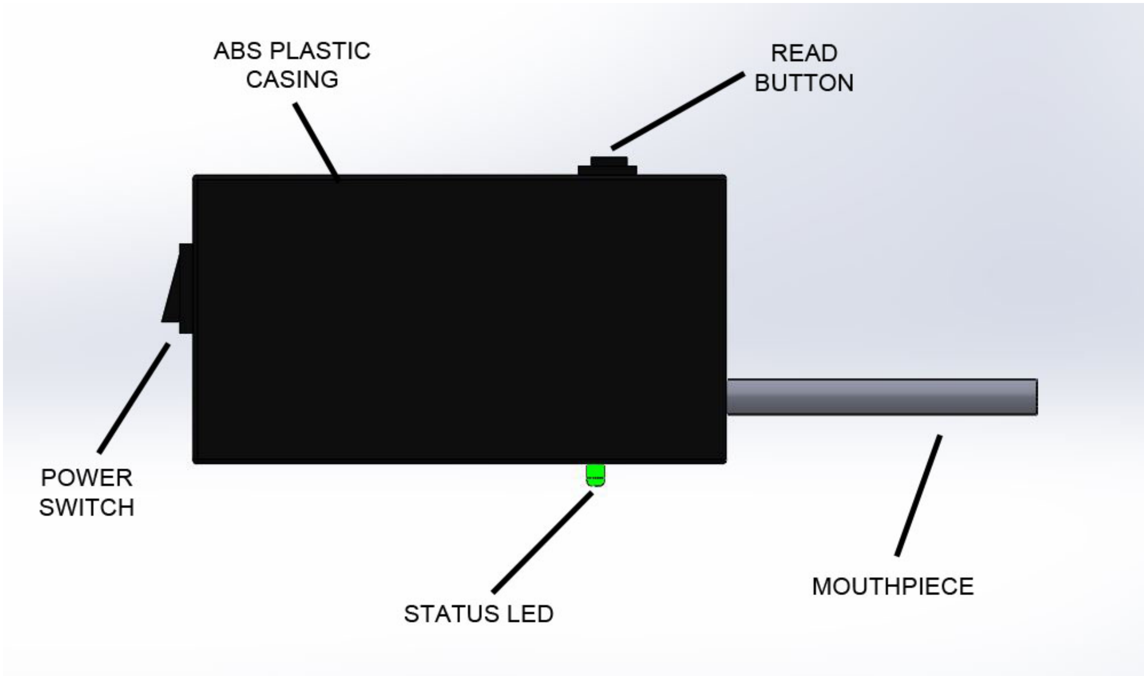


Figure 12: Side view, original SolidWorks drawing of outside of device.

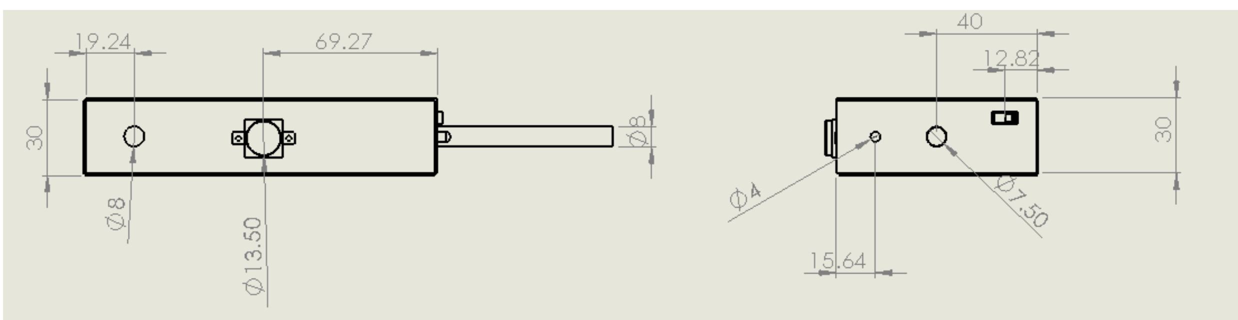


Figure 13: Front and side view, original SolidWorks drawing.

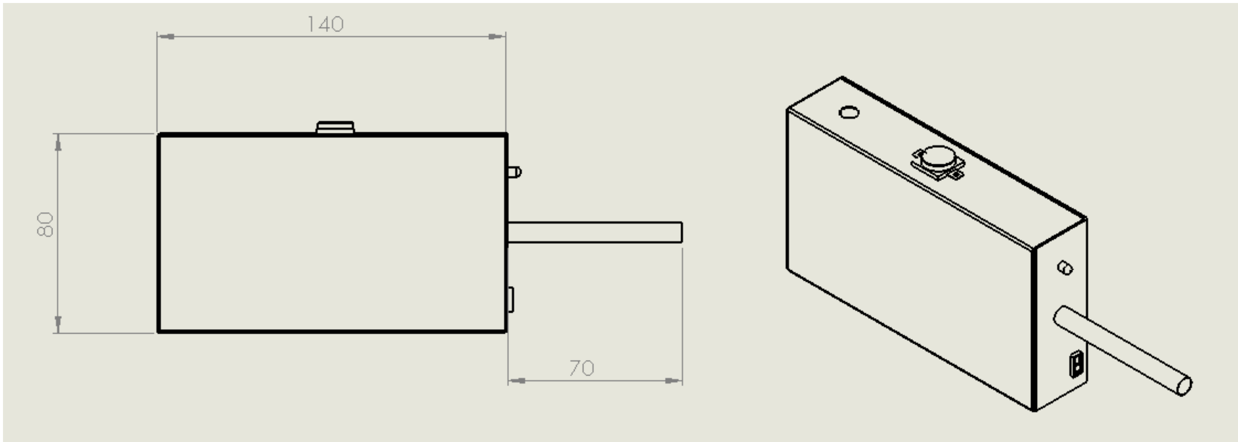


Figure 14: Top and perspective view, original SolidWorks drawing.

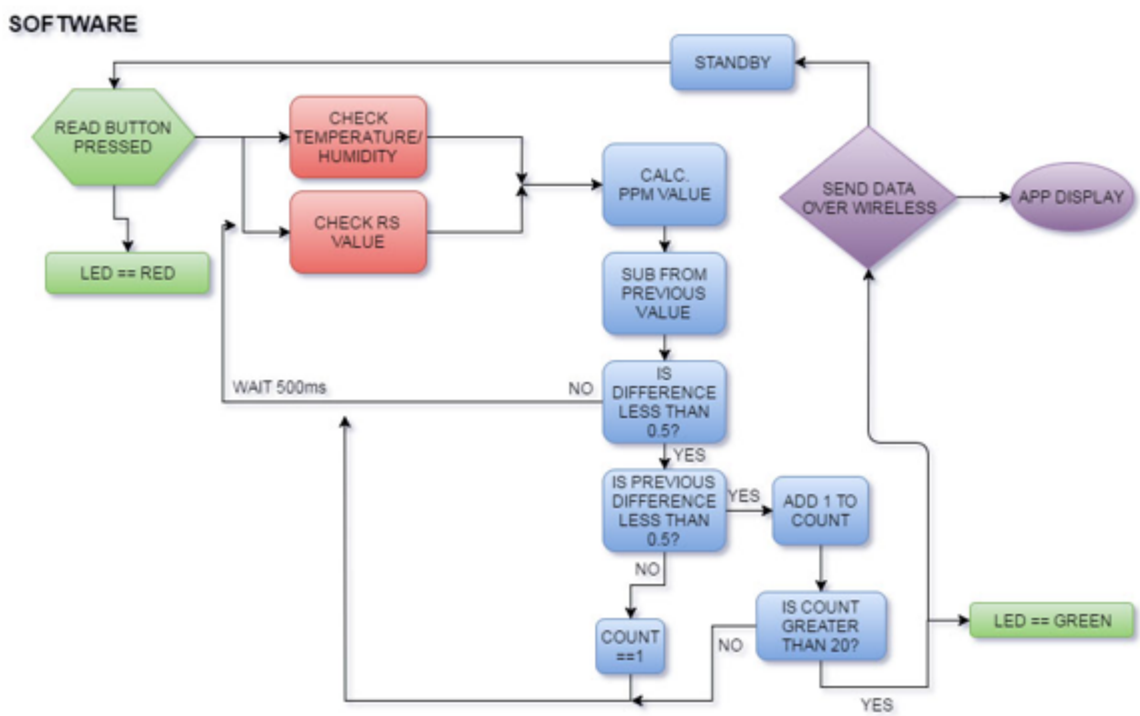


Figure 15: Initial flowchart to show software processes.

5.2 Software Design

There were several plausible ways to code this system. The required elements of the functional software are an input system that receives data from the sensors and stores or handles it in a meaningful way, a communication system that displays meaningful data to the user, and a storage system that maintains data over a long period of time in a meaningful and helpful way. It is very possible to contain the entire code in a single device, and this was the original plan. However, it is much more cost-efficient and space-efficient to use an external device to display and interpret data, and to manage the device over time. In addition, power management is extremely important to this project, as the device must be able to maintain a charge and use energy consistently throughout each day. For these reasons, the most practical design setup for the parameters and concerns of the project was to use a smartphone as the main component and the device to function almost as a peripheral.

The system must then be able to communicate the data it receives/creates to the phone and it needs to either be paired when the reading takes place or store readings and transmit them as a separate function. Since the device already requires at least one hand to hold and operate, forcing the user to use both hands in a coordinated fashion to take every reading was going to be a huge drawback to the effective usage of the device. There was initial discussion as to whether or not this was an effective way to use the device, as it would significantly simplify the code, reduce battery consumption, and force the user to be uploading each reading to the app as it is taken. This method would significantly improve the device's ability to act as a warning system and prevent people from ignoring the results of each reading, as well as simplify the code and help with power consumption, as the Bluetooth device only needs to be on for a very short amount of time. However, the requirement that the user use both the device and the phone in a concerted fashion as well as needing the phone on and active and at hand throughout the reading was simply too much to make the design work.

The obvious alternative is to store data and upload it to the Android device. As a result, the device stores readings and saves them for transmission at a later time (but preferably relatively soon after the reading is taken as the functionality of the device often requires rapid information gathering). The readings will also be processed and interpreted by the Android application as this saves the device from wasting battery life on interpreting data, as the Android application will still be storing them for the long term. This does mean, however, that each reading needed to be accompanied by its date and time. Instead of that, however, the Android application in its current state populates the reading with the date and time at the moment the value is saved. The user is very strongly encouraged to upload the data as soon as possible to minimize the amount of time that the Bluetooth chip needs to be on and active and that data needs to be stored and to maximize the ability of the device to warn the user of danger to himself. It is essentially a pointless reading if the user isn't given the information rapidly. In addition, this is even more important when the user is taking readings with a separate monitor if the device needs to be calibrated before effective usage because the readings will need to be as close in time as possible.

The Android application is, as a result, responsible for receiving any and all readings present in the active storage of the device, interpreting them and organizing them based on their date and time, storing them for long term pattern tracking and also interpretation in the future, and then reacting appropriately to any abnormal or dangerous readings. The high demand being placed on the Android phone allows the device to maximize its ability to save power and therefore maintain the usability of the sensors over the long term.

The system had to be coded such that the device can easily and efficiently transmit the relevant data to the Android application. This involved proper storage, transmission, and handling of values by the device, as well as effective control and operation when reading and idle. The Android application must then interpret the data as it is received and convert it into meaningful values as well as keep track of the user's personal data in a meaningful way.

5.2.1 Device/Input Code

The main function of the code in the device is to take and store breath acetone levels on demand and store them for offloading into the Android application. Additionally, it should have to keep track of and manage the power levels of the sensors and its own battery levels. Since the sensors need to be heated consistently in order for the device to be used on demand, there needs to be persistent power output and usage in order for proper functionality. As a result, there needs to be effective temperature management and checking within the sensors, as well as effective power management and an ability to alert the user if the charge is nearing critical levels as a full discharge will result in the device being out of commission for a decent period of time. Managing this extreme fault in the ability for the device to function in a mobile, daily setting made power saving and management one of the most important design considerations for consumer use in the long term.

The device must effectively receive inputs when the read button is pressed and store them, along with the date and time and maybe other potentially relevant data like temperature or the readings from other sensors that become necessary. Each entry should be stored in a single closed entity that can be transmitted as a single unit or along with others without being confused. These entries should be stored with minimal power consumption and deleted upon successful transmission to the phone (this requires acknowledgement from the phone that the data has been received successfully upon transmission). In addition, the device must constantly or regularly measure and manage the temperature of the sensors, and ensure proper functionality and on demand effectiveness of the device with minimal changes due to external concerns like ambient temperature and where the device is being stored. This needs to be conducted in a way that makes the device function optimally without constantly drawing so much power that the device can not survive with a meaningful charge through out regular usage over the course of a full day before being plugged in again. Ideally, it should be able to last through at least two full days in order to correct for user mistakes in usage, but the full form of the code that allows for this has not been fully implemented as it is not completely possible with the current wiring and device setup in the version of the device we have produced.

The device itself must perform the entire process of taking a reading, from initially pressing the button to storing the data. In addition, it detects if a reading can even be realistically taken at any given time based on the parameters of the other sensors. It should also be able to handle false presses and readings that don't count or need to be ignored, but this could be improved. This is important as extra readings, especially if they are taken when the device is not intentionally being used, could theoretically skew data significantly and also waste a large amount of power. A hold mechanic is one theoretical future method of preventing false readings; another would be a detecting a lack of actual usage or blowing into the mouthpiece.

An important function that was considered seriously for the device code was the event of power being turned off either manually or due to a full battery discharge. In the event that the device turns off, the sensors will be unusable upon power-up again for a period of time until they are readied once more. This is something that should be avoided as much as possible, but it will invariably happen. In addition, the initial power on of a new device will be almost exactly the same situation, especially considering that even a whole new device could potentially be used with the same saved application data on the phone and it would be seen as no different to the phone application. Therefore, all powering up sequences are essentially the same, and therefore there is a required time amount and measurable base resistance value after which the device is accurate and usable. The simplest method for handling this situation is to ensure the user is aware of the meanings of different statuses of the LED indicator and to indicate to them that the device is not yet ready for use while it is still preparing. The device ignores any readings or inputs that the user attempts to take while it is still preparing.

The final important consideration for device usage was that given that the Android application will be used to manage, control, and set up the device in almost every way besides the actual taking of the readings, it is important that the device be able to be connected to by the app even when it is not waiting to upload a reading. In order for this to be possible, a method of turning on the Bluetooth chip and preparing for a connection without necessarily having a ready reading is important. One valid method would simply be to allow the Android application to simply throw away a reading that is taken simply to allow for a connection, but being able to ignore readings so easily could definitely be a downside. Unless there is no reliable way to do this without allowing users to invalidate dangerous readings in order to avoid feeling responsible for their own health, a somewhat complex method of allowing this through the Android application will probably be the most efficient method.

In its current iteration, the device code does store and send values effectively, and smooths over readings by taking a weighted average of multiple readings and sending that as a final value. In addition, it decides which between the two sensors has the correct reading to send. The main limitation, however, is that there is no sleep mode for the device. It is simply running through a loop at all times, and, in addition, the Bluetooth chip is on at all times. In order to solve these problems in the future, plans have been made to refocus the code for a sleep mode as well as to add a power gate (probably using a small transistor) to the Bluetooth power in order to give the processing unit control over the power state of the Bluetooth transmitter. Both of these changes should vastly improve battery life as it currently sits at around 6 hours, if readings are consistently taken throughout that time. Additionally, the device can only hold one reading at a time to send through the Bluetooth transmitter, and this is something that could be examined in the future although it seems to work reasonably well.

5.2.2 Phone Application Code

The primary functional part of the Android application is to control the device and display the output. Therefore, the communication between the device and the phone is of utmost importance. The first thing the Android application had to be able to do was send signals to the device that can function to power it on and off, ask for information about the device's operations, ping the connection, and ask for data from the cache. As the device was originally going to be able to store multiple readouts before sending them to the Android application, it had to therefore be able to take these inputs and organize them and interpret the different data being sent in each file into meaningful information to place into the storage on the phone, as well as react to and display the data in different ways depending on the values present. Because the device only stores and sends single values in the current iteration, however, the phone application simply needs to open a connection channel and receive the reading from the transmitter on the device.

Another important function of the Android application is long-term storage and tracking of the information gathered by the device. This allows it to keep watch over longer-term trends in the acetone levels and therefore potentially the blood sugar of the user. Additionally, it was important for the application to allow the user to add notes and information about the time leading up to each reading, including but not limited to what kind of food was eaten, activities being performed, location, and how the user was feeling. Also important is the ability to add extra data points when a different, more accurate device is used to measure blood sugar. This long-term storage could also possibly be able to act as a basis for finer tuning/calibration of the device. The long-term storage does exist in the device and the user is able to go through and view all previous readings and their date and time data, however, more complex data editing has not been done, although the space and potential for expansion within the code definitely makes this a very reasonable extension of the current design.

The Android application also needed to be easily usable and readable. The user interface must be intuitive and make sense, even given the somewhat complex calculations that may going on in the background. Additionally, it had to ensure that any changing of the settings of the device that could seriously affect operation is difficult or accompanied by proper warnings; this especially includes powering off the device as that will leave it unusable for an extended period of time. That being said, there currently is only one operation mode and so no real settings for the phone application to even change, as well as the existence of a power switch on the device that means that the user simply need use that instead of a connection-based setting change. Other potential device settings that were considered in the Android application are choosing between different timings or modes for taking inputs, choosing between different alert styles with the LED, and managing how the Bluetooth communication interacts with the Android application and phone. A settings window and expansion to include that is very plausible given a revamping of the device code to allow for meaningful settings usage.

The application also had to be able to function and provide meaningful information to the user in a meaningful and important way. This involved having several potential methods of presenting the history of readings and showing trends over time, as well as being able to organize data and readings based on user inputs and tags for each one. The effectiveness of the organization of the data and the effectiveness of the presentation of data to the user highly determine the advantages of the application, and considering that it has to gather and hold all of this data for its own usage anyways, it is not unreasonable to expect its efficient and effective usage for information storage and tracking for the user. In fact, it can be considered that one of the most major user benefits of this application could easily become its data handling and tracking capabilities, with the running of the device only a minor part of the overall functionality. Of course, the main focus of the application during this project is to ensure effective and efficient information transfer and handling of the data taken by the device, as well as effective and powerful handling of the sensors and calibration of their input values. Still, the potential for a very powerful standalone data tracking tool is very close to the intended minimum end goals for the functions of the application, and is something to keep in mind. In the end, the data handling, display, and storage, are all functional, and the information is stored in an efficient and transferrable way which definitely allows for significant data manipulation as well as things like online storage and syncing with physicians in the future, although these concepts have not been included in the current iteration.

In terms of top-level design for the code, it is important to recognize that, especially when working with Android devices, ensuring that the majority of devices can easily handle the application, and that it will run well and perform its functions effectively on the majority of devices was extremely important. Since Android devices very commonly are quite variable in hardware, power, usage style, and even small details of organization, great care must be taken to not simply optimize the application for whatever device is used as a test device, and instead consider the Android community as a whole. Of course, this detailed level of optimization and higher-end tuning of the application is only a goal if feasible and only if it does not interfere with the rest of the project as a whole. Currently, the code is simple and efficient enough to be functional on a very respectable number of Android devices. In order to help achieve this goal, there were two devices on which the test cases of the Android applications will be performed, in addition to the simulated devices inside the IDE. These devices are pictured below in Figure 16.

Figure 16: Smartphone examples that will be used to connect to the device.

The essence of the Android application could be seen as its ability to effectively connect through Bluetooth communication to the device and then gather the data in a reliable and effective fashion before telling the device that the information is present and correct, at which point the device could prepare to disconnect and save power if it were able to turn off the parts. Then, it needed to be able to interpret that raw acetone data from the device as a meaningful value based on calibration and react to these values and present them to the user in an effective and meaningful way. In addition, the application will store these values with their date and time attached to the phones memory. Since the application can successfully perform these tasks, the more complex and higher-end goals and intentions with the Android application code can be worked towards, including but definitely not limited to more complex blood glucose interpretation.

5.2.3 Device Communication

All of the interaction between the device and the Android application takes place over a Bluetooth link. This link needs to be able to effectively and meaningfully transmit the data about each reading stored in the cache of the device. This should all be done while preserving the personal and easy-to-use functions of the device.

Although the Bluetooth chip doesn't consume a huge amount of power, it was still important to restrict its active time as much as possible in order to preserve battery life. The first clear method of solving this issue was to simply keep the Bluetooth chip off unless there is a reading waiting to be transferred to the phone. While this is probably going to be implemented in some way in the future, and is reasonably effective, it is not without its own problems. Chief among those issues is that it becomes much more difficult to simply connect in order to change settings or mess with the device in the event that settings and device management would theoretically be taking place on the application. In addition, there arises an issue with how the chip decides when to disconnect. More specifically, after a reading is transferred, the chip might be simply set to shut off, but this would make it virtually impossible to change settings as the only time the Bluetooth chip is active and connected to the phone is during the specific window of time during which it is transferring the stored data. If the Android application is intended to control settings in the device, there must be an easier and more reliable method for connecting for that purpose alone. However, the first important concern for a future version of the device is definitely to allow the phone to disable the Bluetooth chip to conserve power.

The Bluetooth chip should still be turned on and wait for the Android phone to attempt a connection in order for it to transmit data, but the application would have to be the one that triggers the low-power mode that turns off the Bluetooth chip, as otherwise it would be difficult to set it up in a way that it can smartly turn off and still maintain its function as a path through which settings can be changed. In addition, a method for simply connecting to change settings would need to be implemented in order to allow for settings changing in the first place. There are two relatively simple ways to do this, either by using a different input pattern or extra button on the device to set the Bluetooth chip to on (especially if there's a method of simply activating it for a short time so that if a connection is not made it turns back off to avoid extra button presses) or by making the user press the input button as normal but allowing them to throw out an input by declaring it simply a settings test input. In addition, the Bluetooth link should update the settings on the device whenever it connects, so that changed settings in the application while not connected can still be uploaded to the device and set upon that connection. In fact, if that specific functionality is powerful enough, it could remove a lot of the importance of this requirement for alternative settings connections.

It was very important to ensure the reliability of the Bluetooth link and of the data that is sent across it. Incorrect readings and date/time values could cause a lot of problems, and the data from any and all readings should be transmitted in an effective and organized manner. Care had to be taken to detect and handle any errors that could occur in transmission. In addition, it was important to ensure that the Android device has a relatively easy time connecting and maintaining the connection for the duration that it is important. Managing the connection, its reliability, and ensuring proper communication that allows for all of the functions that the connection will fill were tasks that were highly important and that required balancing and attention from the coding on both the device and the Android application. The end result is a style of programming that causes the device to push the reading values, after several are taken and a weighted average is calculated by the device, to the chip in a string format, which results in the Bluetooth device waiting to transmit until the Android application is set to connect and receive, at which point the device sends the value and the application receives and interprets it into a full reading, which includes saving the date and time information. Then, the user returns to the main menu and saves the new reading to the phone's memory permanently.

The Android application also had to include the ability to manage the connection and also identify the device as the specific device used by the same user, in order to ensure reliability and continuity in data. Assuming a situation where the device could possibly have to interact with multiple devices in proximity, care should be taken to ensure the connection is made to the correct device and that the inputs are saved properly. The device code will have to be able to properly detect the correct Android device to connect to and include a way to display its own unique ID when using Bluetooth to make connections. Currently, the device connects through the unique MAC address of the device, although hard coding each application for the MAC address is slightly unreasonable, so a different method of ensuring security will be examined into in future iterations.

5.2.4 Sensor Calibration Coding

A very important part of Android application's job (and almost a completely separate coding entity from the rest of the application) was to interpret and convert the sensor's output into a meaningful value based on the calibration and information stored in the app. It may also be wise in the future to offload a copy of this data to the device to store as the user's personal information so if the phone were to break or the app to be reinstalled, the data wouldn't have to be collected from the beginning over again.

Because very little headway was made in terms of finalizing a conversion between the acetone values that were read and the levels of blood glucose in the user, the calibration of conversion remains a very theoretical part of the device. Many plans were made in order to simplify the process of developing the proper algorithm, but there are still currently fundamental issues with the entire process of coding for a conversion in the first place. The main issues with the current iteration of the device are that it is not reliable and consistent enough with readings due to factors like inconsistent breath by the user not being corrected for as well as a lack of ensuring no air is bled out of the device or even a lack of finding the optimal way to take the reading while or after the user is taking the reading. The coding for the conversion is still being converted, and the results from testing do suggest a correlation but there is still much work to be done before this section can in any meaningful way be implemented.

Depending on the results of testing the app across multiple people, it could be that each individual's levels of acetone correlate differently to their blood glucose levels. In this case, it would require that each user go through a setup process that involves taking samples with the device and, at the same time, using a conventional blood glucose meter to test for a value, and then inputting that value into the application to allow for effective conversion. Alternatively, if there does happen to be a consistent and common correlation between acetone and blood glucose, then this coding and optimizing will simply have to take place once to as precise a level as possible while building the device.

The planning for the setup and coding of the sensor calibration will assume the most complex situation possible, and then if the situation becomes one of those that is a simpler or more easily solvable one, then the work and setup will be scaled down to fit that situation. The most complex situation predictable for the way this correlation and interaction of blood glucose and breath acetone happens (assuming it is measurable and functional in the end anyways) is that the acetone to blood glucose correlation is variable from person to person, and is variably nonlinear and unique to each user. In fact, it is also possible that the correlation changes over time based on things like diet and lifestyle in each user. This situation would not make the project impossible but it might make consistent long-term usage of the device difficult to manage or make highly user-friendly.

In this most complex situation, the initial startup sequence of every iteration of the device will have to run through a calibration setup where the user sends readings and then inputs their accurate blood sugar using a more accurate and reliable meter. After this is done several times at different blood sugar levels, a best fit line or curve will be modeled in the Android application. This will serve as the basis from which to measure and describe all future readings. That being said, it needs to still be able to update the curve and keep the calibration aligned with time. As a result, the user will need to periodically take readings that are validated with a more reliable device and then those data points will be added to the calibration curve and weighted in such a way to make the readings more accurate.

The device will ideally have to maintain a copy of the best fit line or curve (or at least of the degree and coefficient values so as to easily recreate it in future iterations) in its memory in case the Android application or device or memory fails in order to avoid forcing the user to recalibrate entirely from scratch, but this won't be implemented unless other higher-end and less relevant goals are achieved after successful creation and programming of the entire system. The formula will probably also be saved as the coefficient values and the degree of the polynomial and then inputted values will be multiplied through to get the projected value of the blood sugar according to the graph. The precision will be thoroughly explored and tested for and discussed after creation and coding, as well. In addition, displayed values and precision levels will hopefully coincide with this examination. The accuracy of the values are still the most important consideration and using the formula values and polynomial degree that most effectively models the true correlation will be the ultimate goal for the calibration. Of course, the calculation and management of these values needs to be well within the processing power of even more basic Android devices, and so this will definitely be taken into account while coding and testing this part.

The sensor calibration and conversion section of the code are written as a somewhat separate code entity within the Android application. This is done so that the Android application can serve as a scaffolding for reception of data and so that it can still display values and help with hardware testing, but when it is sufficiently complete and satisfactory the separate sensor calibration code can still be regularly changed and edited and error-corrected without affecting the rest of the Android application. This is intended to make the eventual testing, writing, and essentially most of the work done while creating the device much simpler. In addition, diagnosing errors within the code will be much easier if the code is separated into distinct complex sections. A small error in the coding of the user interface should not affect the outcome and interpretation of a value from the device. While it will be split up into more entities and units in order to further achieve these goals, this sensor calibration code will be almost a completely separate entity for very intentional reasons.

In addition, since there will, at least initially, be two sensors that are detecting the levels of breath acetone, the calibration will have to take into account both readings and be able to detect which one to prioritize based on their recommended ranges of detection. In addition, for values close to the borderline or where both can be accurate, a weighted average may be used to ensure higher accuracy. In addition, other sensors may be needed for comparison or management of the requirements of the main two sensors and to allow for insurance that the values measured are accurate in different environmental settings. In fact, it is also possible that the values may vary based on some other measurable value, in which case it will be necessary to also measure that and apply it to the formula for the calibration and to be saved as part of the long term storage and tracking in the device.

There are many things to take into account when managing sensors, especially ones that can be as fickle as those that are being used in the diabetic breathalyzer. However, there are certainly ways to correct for and work around most of the issues or nonlinearities that arise in this setting. Ensuring the most accurate and precise values for blood sugar is highly important and necessary when considering how dangerous bad readings could be to those suffering from diabetes.

6.0 Prototype Construction

6.1 Master Parts List

Table 8, shown below, displays the Master parts list for our final product. After comparison and decision making about which parts to use in the previous section, these are the parts that we decided to use for our design.

	Part	Part #
Control Unit	Microcontroller	ATMega328p
	PCB + misc. resistors and capacitors	-
Sensors	Low Concentration VOC	WSP2110
	High Concentration VOC	TGS822
	Temperature and Humidity	DHT22
Communication	Bluetooth module	HC-05
Power	7.4V 2300mAh lithium ion battery	-
	Lithium ion battery charger	-
	Step down regulator	R-78W5.0-0.5
External Components	Aluminum casing	-
	Pushbutton	-
	Slide switch	-
	Antimicrobial tubing	-

	RBG LED	WP154A4SUREQBFZGW

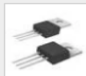
Table 8 : Master Parts List.

6.2 BOM and parts acquisition

The parts aquisition list for our device can be found in table 8. Most components that are not the MCU and sensors will be used for the prototyping phase on the breadboard. All of those components will be ordered with the PCB when final testing begins.

From Digikey:

Quantity	Image	Part Number	Description	Customer Reference	Backorder	Unit Price	Extended Price
4		ATMEGA328P-PU-ND ATMEGA328P-PU	IC MCU 8BIT 32KB FLASH 28PDIP		0	3.70000	$14.80
50		CF14JT1K00CT-ND CF14JT1K00	RES 1K OHM 1/4W 5% CARBON FILM		0	0.02880	$1.44
50		CF14JT4K70CT-ND CF14JT4K70	RES 4.7K OHM 1/4W 5% CARBON FILM		0	0.02880	$1.44
10		P13478-ND EEU-EB1H100S	CAP ALUM 10UF 20% 50V RADIAL		0	0.30000	$3.00
10		P5165-ND ECA-1VM101	CAP ALUM 100UF 20% 35V RADIAL		0	0.17200	$1.72
5		C503B-BCN-CV0Z0461-ND C503B-BCN-CV0Z0461	LED BLUE CLEAR 5MM ROUND T/H		0	0.21000	$1.05
4		754-1492-ND WP154A4SUREQBFZGW	LED RGB DIFF 5MM ROUND T/H		0	1.78000	$7.12

5		490-5999-1-ND CSTLS16M0X53-A0	CER RES 16.0000MHZ 15PF T/H	0	0.42000	$2.10
4		MCP1700-3302E/TO-ND MCP1700-3302E/TO	IC REG LDO 3.3V 0.25A TO92-3	0	0.37000	$1.48
2		IFX25001TS V50-ND IFX25001TS V50	IC REG LDO 5V 0.4A TO220-3	0	1.31000	$2.62
4		ED3050-5-ND ED281DT	CONN IC DIP SOCKET 28POS TIN	0	0.33000	$1.32

Qty.	Picture	Name	Part Type	Source	Unit Price	Total Price
2		DHT22/AM2302	Temp/Humidity Sensor	Aliexpress	7.84	7.84
2		TGS822	VOC Sensor High Concentration	Aliexpress	5.94	11.88
1		SeedStudio WSP2110	VOC Sensor Low Concentration	Amazon	16.80	16.80
2		Sunkee RS232 TTL HC-05	30 ft Bluetooth Module	Amazon	8.80	17.60

Table 9: Purchasing history for prototyping/final device components.

Sensor Calibration

In order to properly calibrate the VOC sensors, linear line regression is necessary to turn the provided plots from the datasheets into workable equations that our code can then handle. These equations turn our measured analog voltage from the voltage divider configuration into a perceived sensor resistance. This resistance is then plugged into the gathered equation to get a usable value in parts per million of the particular gas we are concerned with, acetone.

Below are the plots attained from the datasheets of the two VOC sensors, the TGS822 and the WSP2110. Each sensor has two separate plots. The first plot relates the concentration of various gasses to the ratio of the actual sensor resistance and a predefined resistance. For the TGS, the ratio is of the sensor resistance to the resistance of the sensor at 300 ppm ethanol. Since it would be very difficult to get a setting with exactly 300 ppm concentration of ethanol to get the numerical value for that resistance, we can use the levels the chart give for how the sensor reacts to open air. For the TGS, the ratio of the sensor resistance in air to the sensor resistance in 300 ppm ethanol is about 18. Knowing this, we will allow the sensor to be on for a few days to assure it is properly heated, and then take that measure resistance and divide it by 18 to get the needed baseline resistance. For the WSP sensor we will use a similar strategy, however since the ratio is equal to 1 in open air, we can just use that measured value as our baseline during the calculations.

The other plots provided are the correlation that temperature and humidity have on the ratio to the baseline resistance. With these relations we are able to compute a scaling factor, which we will divide our computed ratio by. By doing this we hope to stabilize the readings even when a sudden change in temperature and humidity is seen. This of course will happen each time the device is used since the temperature and humidity of breath is higher than open air. (37)(38)

In order to get an equation for our program we have to use the given plots and line regression. To achieve this easily we took all the data and plugged it into Microsoft Excel. Using the Data Analysis Tools, we gathered a best-fit line for all sets of data. The results gave us an R-squared value, the line of intercept, and the needed coefficients for the equation. Using this, an equation was easily obtained. The R-squared value attained assured that the equation we were using had a strong enough correlation to be able to use. It was seen that with the TGS sensor specifically there was a need to break up the plot into two different lines since the correlation between resistance and concentration is stronger as the resistance drops further. When we tried to have a single line for the entire plot, we found the R-squared was too low. By breaking it into two separate equations, one that handles cases for a resistance ratio of more than 0.6 and one for a ratio less than 0.6, we were able to get a high R-squared value and therefore the necessary correlation.

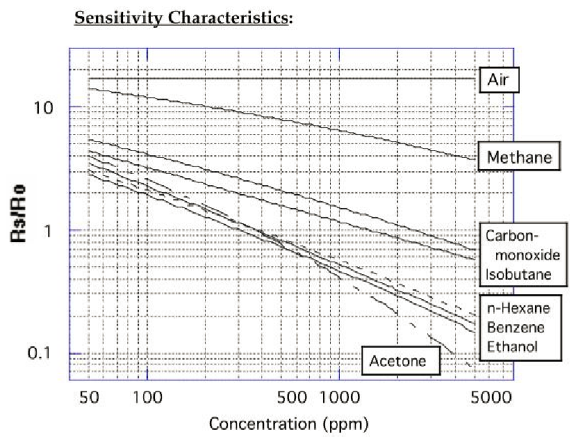


Figure 17: Plot of TGS822 resistance response to gas concentration (37)

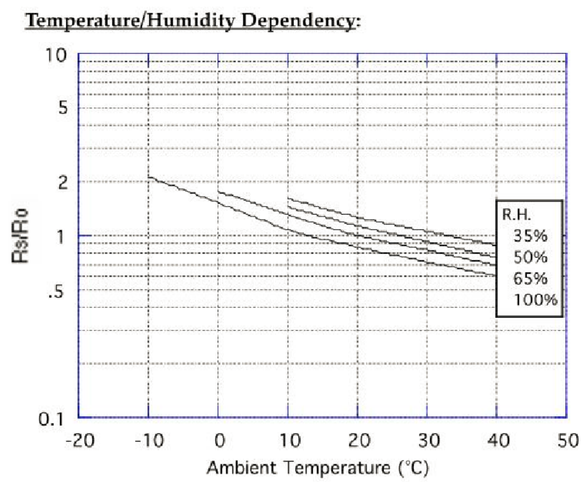


Figure 18: Plot of TGS822 resistance response to humidity and temperature (37)

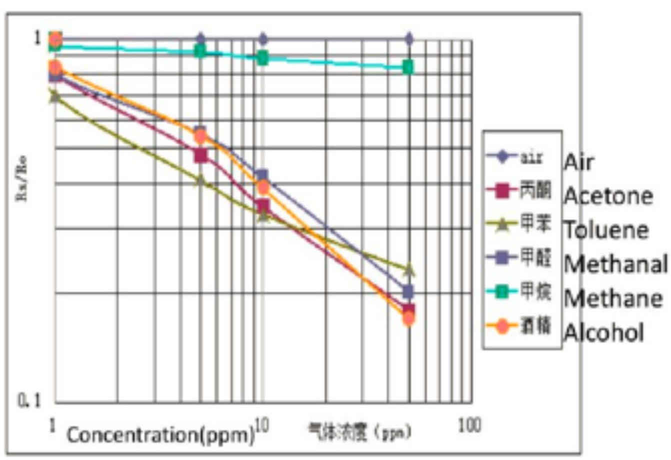


Figure 19: Plot of WSP2110 resistance response to gas concentration. (38)

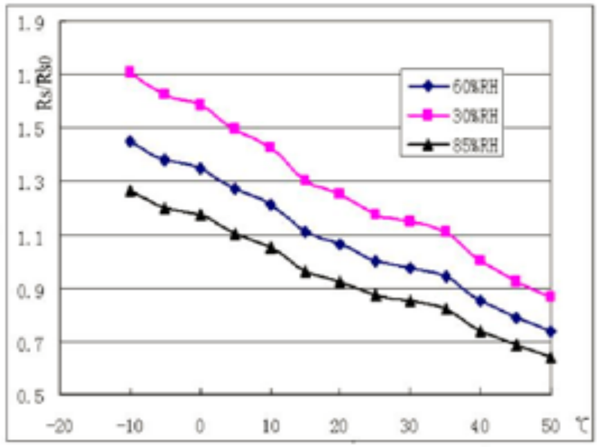


Figure 20: Plot of WSP2110 resistance response to humidity and temperature. (38)

As we can see from figures 17 and 19, there is a correlation between lower measured resistance and higher concentration of gasses. As shown in figures 18 and 20, there is a similar correlation with temperature and humidity. The sensors are susceptible to various gasses but we make the assumption for this project that they will not be present in high enough numbers in the breath to have an impact. The only substance that could show up in large enough quantities would be ethanol a therefore in order to get an accurate reading, the user cannot drink any alcohol sine the sensor would not be able to differentiate it from acetone.

TGS: Resistance ratio to concentration correlation:

$$\frac{R_S}{R_O} \geq 0.6$$

Y	log(Y)	X	log(X)
4	0.6021	50	1.699
3.6	0.5563	60	1.7782
3.4	0.5315	70	1.8451
3.05	0.4843	80	1.9031
2.85	0.4548	90	1.9542
2.55	0.4065	100	2
2	0.301	150	2.1761
1.6	0.2041	200	2.301
1.45	0.1614	250	2.3979
1.25	0.097	300	2.4771
1.125	0.0512	350	2.5441
0.95	-0.0223	400	2.6021
0.875	-0.058	450	2.6532
0.75	-0.1249	500	2.699
0.69	-0.1612	550	2.7404
0.615	-0.2111	600	2.7782
0.6	-0.2218	650	2.8129

Table 10: Plot points for first equation used for a resistance ratio greater than or equal to 0.6 (37)

Table 9 shows the interpreted data from the plot provided form the TGS822 data sheet. This portion only includes data when the resistance ratio is above or equal to 0.6. The Y and X columns are the real values shown on the plot, while the log columns provide the logarithmic values of the data, which is what the lines in the plot are based on. We must use these values when forming an equation to use. Table 10 has the results of the regression analysis. R square shows the correlation level, while the coefficients are the values used in the actual equation. These same statistics are found in Tables 11 and 12 for ratios above 0.6.

Regression Statistics			*Coefficients*	*Standard Error*
Multiple R	0.995498677	Intercept	1.897972555	0.042791703
R Square	0.991017616	X Variable 1	-0.742211532	0.018244719

Table 11: Regression analyis results for the first equation of the TGS correlaiton.

$$\log\left(\frac{R_S}{R_O}\right) = 1.897973 - 0.742212*\log(PPM) \quad ⑦$$

$$\frac{R_S}{R_O} < 0.6$$

Y	log(Y)	X	log(X)
0.57	-0.2441	700	2.8451
0.545	-0.2636	750	2.8751
0.5	-0.301	800	2.9031
0.485	-0.3143	850	2.9294
0.455	-0.342	900	2.9542
0.43	-0.3666	950	2.9778
0.4	-0.3979	1000	3
0.29	-0.5376	1500	3.1761
0.2	-0.699	2000	3.301
0.18	-0.7447	2500	3.3979
0.14	-0.8539	3000	3.4771
0.12	-0.9208	3500	3.5441
0.095	-1.0223	4000	3.6021
0.078	-1.1079	4500	3.6532
0.07	-1.1549	5000	3.699

Table 12: Plot points for the TGS for resistance values greater than 0.6 (37)

Regression Statistics	
Multiple R	0.99659291
R Square	0.993197428

	Coefficients	*Standard Error*
Intercept	2.7259232	0.077098644
X Variable 1	-1.037741604	0.023819703

Table 13: Regression analysis results for the second equation for the TGS

$$\log\left(\frac{R_S}{R_O}\right) = 2.725923 - 1.037742*\log(PPM)$$ ②

Scaling Factor:

In order to calculate the scaling factor, analysis must be done using two independent variables. While the math to do this is tricky, again Excel can handle it with the same analysis tools as before. One difference between this data and the previous is that only the Y values are in logarithmic form. The Two independent variables, temperature and humidity, use a typical scale and therefor don't need any special treatment. This analysis can be found in tables 13 and 14.

Y	log(Y)	X1(T)	X2(H)
2.35	0.3712	0	35
2.05	0.3118	0	50
1.78	0.2504	0	65
1.6	0.2041	0	100
1.65	0.217	10	35
1.5	0.176	10	50
1.4	0.146	10	65
1.1	0.041	10	100
1.45	0.161	20	35
1.2	0.079	20	50
1	0	20	65
0.75	-0.125	20	100
1.05	0.021	30	35
0.9	-0.0458	30	50
0.8	-0.086	30	65
0.7	-0.155	30	100
0.9	-0.046	40	35
0.78	-0.108	40	50
0.7	-0.155	40	65
0.6	-0.222	40	100

Table 14: Plot points for the TGS dependency on temperature and humidity. (37)

Regression Statistics	
Multiple R	0.984680548
R Square	0.969595783
Adjusted R Squa	0.966018816

	Coefficients	Standard Error
Intercept	0.445458602	0.02197672
X Variable 1	-0.010457	0.000498431
X Variable 2	-0.002952538	0.000292374

Table 15: Regression analysis for temperature and humidity dependence for TGS

$$\log(SF) = 0.44546 - 0.010457*T - 0.002953*H \quad ③$$

(X1 = Temperature|X2 = Humidity)

Above are the analysis reports generated by Microsoft Excel. Since the information used to create these reports was gathered by looking at a plot that does not have exact information, the equations gathered will not be perfect. Without lengthy testing and calibration with proper equipment, there is no way to get an exact model for data acquisition. The best we can do is get an equation that has an R-squared value of over 95%, which assures that the equations gathered fit our needs. Equations ① and ② show the relationship between the resistance ratio and gas concentration for the TGS. The first equation covers lower concentrations with a ratio value of greater than 0.6. Once that ratio dips below 0.6, it is necessary to use the second equation to get a proper reading. Before either of these equations can be used, it is necessary to account for temperature and humidity which directly affect the read resistance values. The Scaling Factor, ③, takes care of this and once it is found, we can divide the perceived ratio by it to get the adjusted proper ratio.

WSP resistance ratio to concentration correlation:

For the WSP2110 again there is a rather significant change in the linearity but for this sensor that occurs when the resistance ratio is greater or equal to 0.45 and when it is less than 0.45.

$$\frac{R_S}{R_O} \geq 0.45$$

Y	log(Y)	X	log(X)
0.8	-0.0969	1	0
0.64	-0.1938	2	0.301
0.56	-0.2518	3	0.4771
0.51	-0.2924	4	0.6021
0.48	-0.3188	5	0.699
0.45	-0.3468	6	0.7782

Table 16: Plot points for ratios above 0.45 for WSP (38)

Regression Statistics	
Multiple R	0.999841592
R Square	0.999683208

	Coefficients	Standard Error
Intercept	-0.097545799	0.001550599
X Variable 1	-0.320299994	0.002850904

Table 17: Regression analysis

$$\log(\frac{R_S}{R_O}) = -0.097546 - 0.3203*\log(PPM) \quad (4)$$

$$\frac{R_S}{R_O} < 0.45$$

Y	log(Y)	X	log(X)
0.415	-0.382	7	0.8451
0.375	-0.426	8	0.9031
0.35	-0.4559	9	0.9542
0.33	-0.4815	10	1
0.3	-0.5229	15	1.1761
0.265	-0.5768	20	1.301
0.245	-0.6108	25	1.3979
0.23	-0.6383	30	1.4771
0.22	-0.6576	35	1.5441
0.2	-0.699	40	1.6021
0.195	-0.7099	45	1.6532
0.188	-0.7258	50	1.699

Table 18: Plot points for ratios below 0.45 for WSP (38)

Regression Statistics	
Multiple R	0.996052817
R Square	0.992121215

	Coefficients	Standard Error
Intercept	-0.083748479	0.014172121
X Variable 1	-0.378162159	0.010656764

Table 19: Regression analysis for ratios below 0.45

$$\log(\frac{R_S}{R_O}) = -0.083748 - 0.378162*\log(PPM) \quad (5)$$

Scaling Factor:

Y	log(Y)	X1(T)	X2(H)
1.6	0.2041	0	30
1.35	0.1303	0	60
1.18	0.0719	0	85
1.42	0.1523	10	30
1.27	0.1038	10	60
1.05	0.02119	10	85
1.28	0.1072	20	30
1.08	0.0334	20	60
0.92	-0.0362	20	85
1.14	0.0569	30	30
0.99	-0.00436	30	60
0.86	-0.0655	30	85
1.05	0.0212	40	30
0.85	-0.0705	40	60
0.73	-0.1367	40	85

Table 20: Plot points of ratio to humidity and temperature dependence for WSP (38)

Regression Statistics	
Multiple R	0.995667509
R Square	0.991353788
Adjusted R Squ	0.989912753

	Coefficients	*Standard Error*
Intercept	0.282992256	0.00750999
X Variable 1	-0.004916167	0.000170236
X Variable 2	-0.002492576	0.000107073

Table 21: Regression analysis for temperature and humidity dependence for WSP

log(SF) = 0.282992 – 0.004916*T – 0.002493*H ⑥
(X1 = TemperatureIX2 = Humidity)

In order to have these equations work for us in the code computation they will need to be changed around some. This is because, the resistance ratio was the y-axis variable in the charts and therefore is the y variable in the analysis. This is fine, but we will have the ratio once the analog read is performed on the MCU. The unknown variable is the PPM or concentration in parts per million. In order to achieve this, the first two equations for the TGS and the first two equations for the WSP will have to be put in this form:

① $PPM = 10^{\left[2.55718-\left(\log\left(\frac{R_S}{R_O}\right)\times 1.347324\right)\right]}$ ② $PPM = 10^{\left[2.626783-\left(\log\left(\frac{R_S}{R_O}\right)\times 0.963631\right)\right]}$

④ $PPM = 10^{\left[-0.30455-\left(\log\left(\frac{R_S}{R_O}\right)\times 3.12207\right)\right]}$ ⑤ $PPM = 10^{\left[-0.221461-\left(\log\left(\frac{R_S}{R_O}\right)\times 2.64437\right)\right]}$

The ratio used in these equation will have to be adjusted by first dividing it by the scaling factor. A similar strategy must be used to calculate the two separate scaling factors with those equations looking like:

③$SF = 10^{[0.44546 - 0.010457 \times T - 0.002953 \times H]}$ ⑥ $SF = 10^{[0.282992 - 0.004916 \times T - 0.002493 \times H]}$

Hardware Pictures

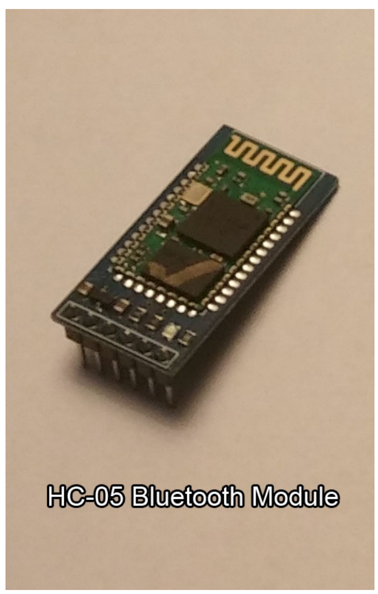


Figure 21: Original picture of purchased HC-05 wireless module.

Figure 21 shows the HC-05 Bluetooth module that we will be using as our means of wireless communication between the device and Android application.

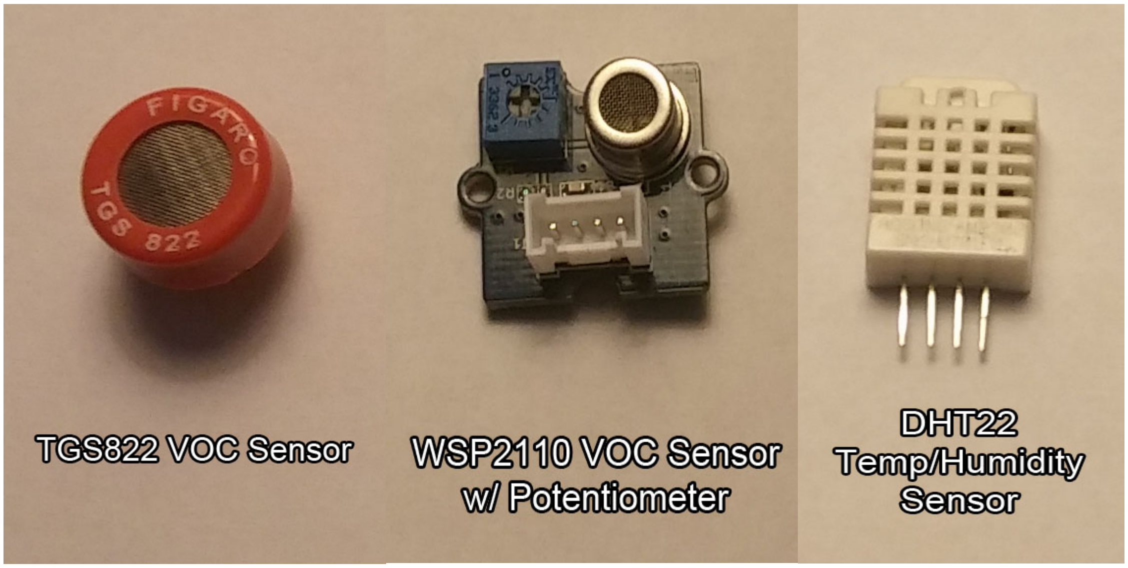


Figure 22: Original picture of purchased sensor modules.

Figure 22 shows all three sensors that will be used to compute the proper data for the device. The two VOC sensors are pictured on the left with the TGS being involved in higher concentrations and the WSP used for lower concentrations of acetone. The sensor on the right is the DHT22 and will measure the temperature and humidity since the VOC sensors have dependencies on those measurements.

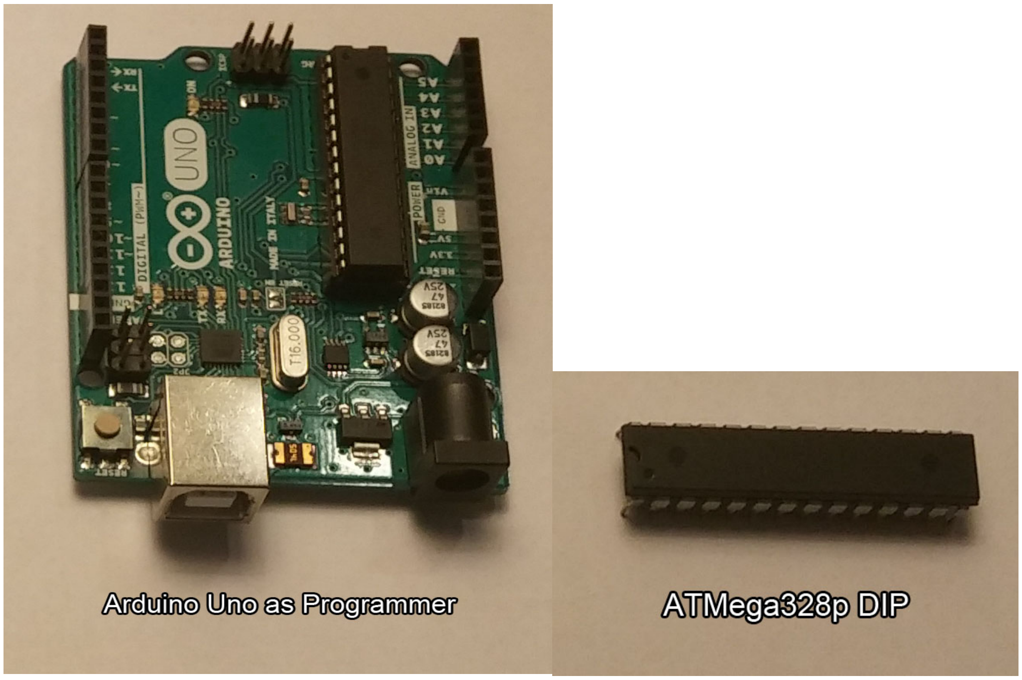


Figure 23: Original picture of programmer and chip used.

Figure 23 shows the Arduino Uno and the standalone chip, the ATMEGA328p that we will use during prototyping. The actual chip we will use on the PCB will be a smaller surface mount version of the chip that was described earlier in the paper. The plan is to use the standalone chip pictured to create the appropriate code. This will be accomplished using the pictured Arduino Uno as a programming interface with a computer and subsequently the Arduino IDE. Once the PCB arrives we should only have to program it one time, with the functioning code.

The sensors and the Bluetooth module will be directly used in the device. Except for those components, all other parts will be incorporated directly on the PCB. This will include an assortment of resistors and capacitors along with a 16 MHz resonator for the ATMega328p to function properly. Not included in these hardware pictures are the yet to be ordered battery and charging system. As the design process progresses they will be purchased according to the details outlined in power section of 5.1.1.

6.3 PCB Specs

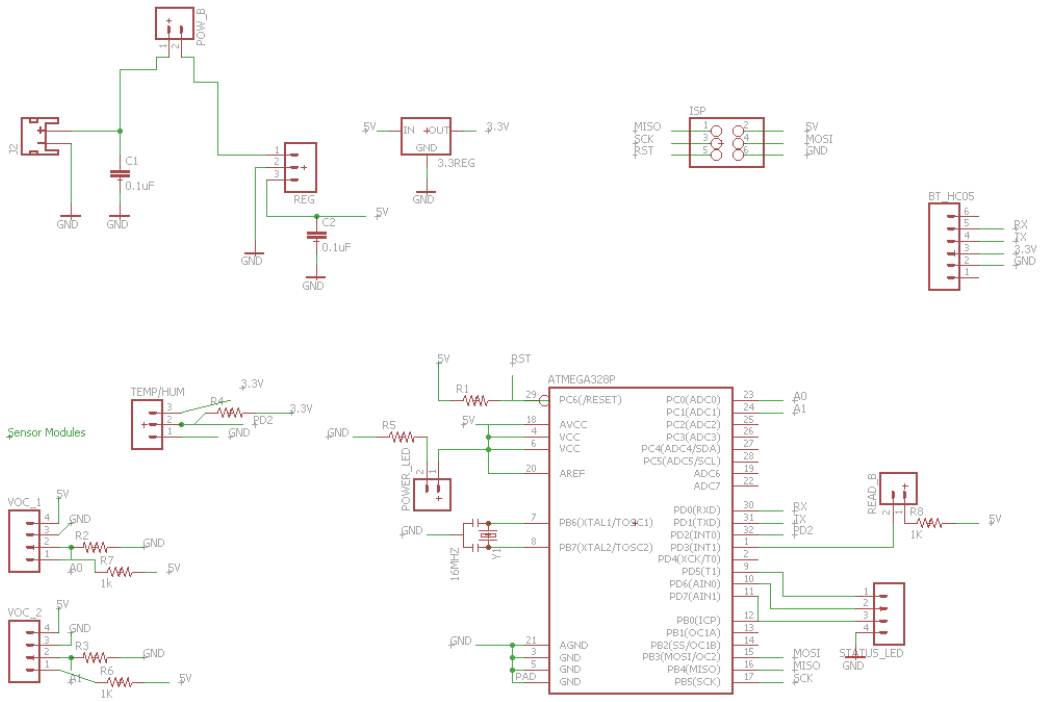


Figure 24: Entire original eagle schematic.

Figure 24 shows the Eagle schematic for the entire device. Each section is further explained with that part of the schematic zoomed in upon in order to see detail like which nets each pin gets connected to. As we can see from the entirety of the picture above, the majority of our PCB will be empty through holes in which we will manually solder wires to that connect all of our external devices. Though this may be manually intensive it does allow for a simple PCB design, which equates to an even cheaper PCB.

Sensor Schematic:

Figure 25 shows the Eagle Schematic design of how the two VOC sensors and one Temperature and Humidity sensor will be connected in the device. In order to connect the sensor modules to the PCB we have fabricated, will have to solder the wires directly to the through holes on the PCB at the designated positions as specified in the PCB design. The PCB itself will have the current limiting resistors and the voltage divider resistors built onto it. One issue here is that the WSP2110 module came in a pre-built package that has a potentiometer built in to create the voltage divider resistor. If this module is used, there will be no need to implement a voltage divider resistor (RL) for that particular module, VOC2. The TGS822 however is only a standalone sensor and will therefore require the RL to be on the actual PCB. (37)(38)(40)

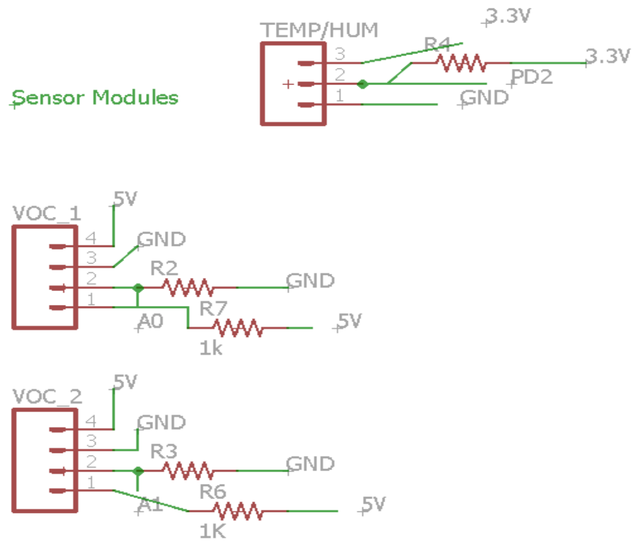


Figure 25: Original Eagle schematic of necessary connections for the operation of sensors.

Control Unit Schematic:

The basic setup of the standalone ATMega328p is shown in figure 26. Located on the PCB will be a 16MHz ceramic resonator, a power indication Led with a current limiting resistor, a pull up resistor on the reset pin, and position for a read button to be connected. When the digital pin connected to the read button goes high, the device will know to poll the two analog pins to gather the sensor information. All other connections to the ATMega328p involve the sensor and communication modules. The 5V to run the microcontroller will come from the regulated 5V that will be produced by the step down converter.

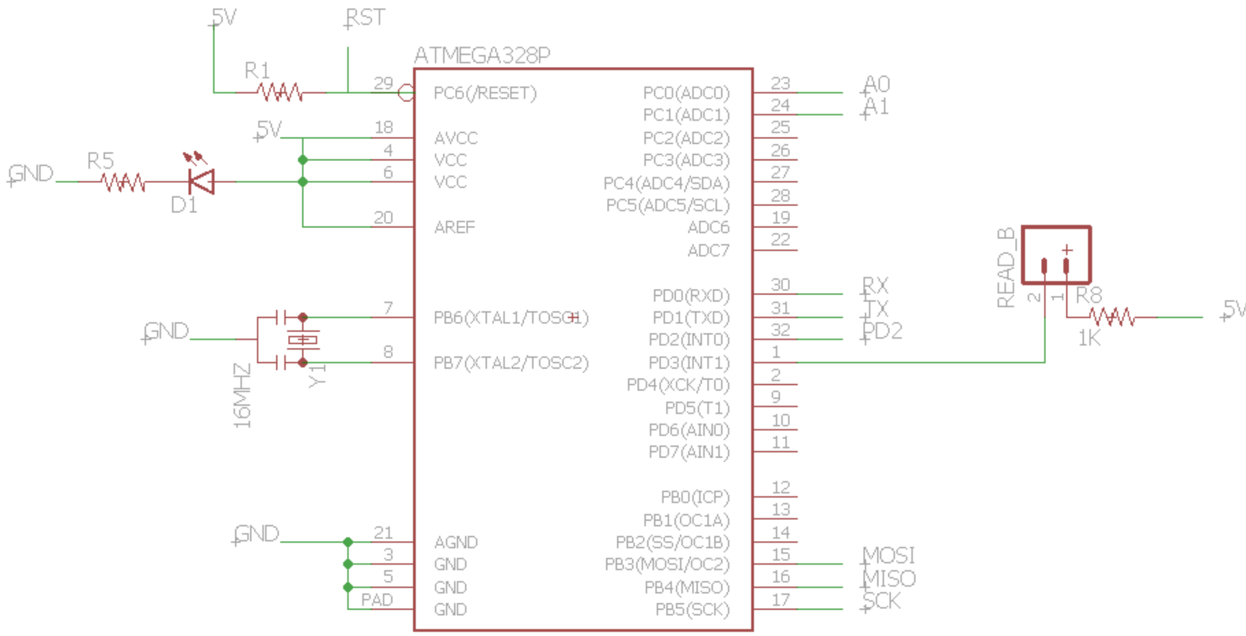


Figure 26: Original Eagle schematic of management of the ATMega328p. (41)

Power Schematic:

Figure 27 shows the necessary schematic for the power that will be running the device. From a lithium ion battery connector, which will be mounted on the PCB, 7.4V will be applied to the high efficiency step down regulator to achieve a constant Dc supply of 5V with little power loss. The more power we can conserve, the longer the battery will be able to last. The connection between the 7.4V source and the step down regulator will be controlled by a simple slide switch. This will be accessed on the outside of the unit like the read button, but will be placed near the mouthpiece. This switch will be left on during times where continuous reading is needed. Filter caps are used on incoming voltage sources to make sure fluctuations do not happen. Since the device is heavily dependent on exact readings, it is important to have a constant, unwavering supply. The final piece of the power system is the 3.3V linear regulator, which will provide a voltage source to the DHT22 sensor and the HC-05 BT module.

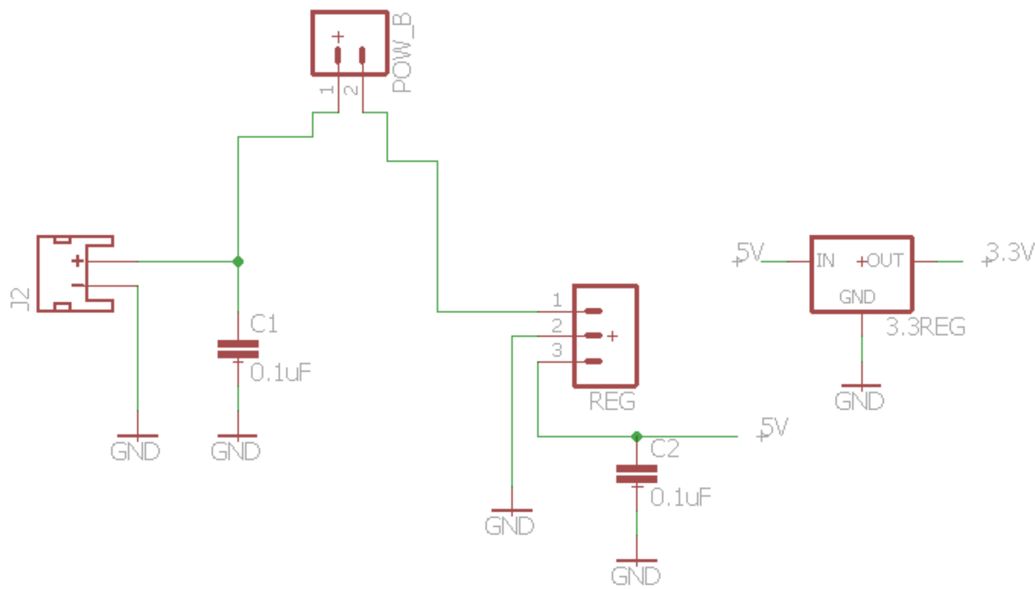


Figure 27: Original Eagle shematic of power control and regulation setup.

Figure 28 shows the schematics for the In System Programmer and the Bluetooth module. The ISP will be located in the center of the PCB and will allow us to rewrite the code on the ATMega328p whenever we like. All programming changes will likely happen while in prototyping, and therefore these connections will not be accessed regularly once the PCB is manufactured. But we need the connection at least one time to initially program the surface mounted chip on the PCB. The other schematic shows how the HC-05 Bluetooth module will be connected. It only requires the 3.3V regulated DC source along with a ground connection. The two other pins used will connect directly to the serial connections of the ATMega328p, which are the RX and TX pins.

In System Programmer Schematic: Module Schematic:

Figure 28: Original Eagle schematic of standard 2x3 ISP configuration and the needed pins for the HC-05. (39)

6.3.1 PCB Design

Due to the small nature of our circuitry requirements it will be necessary to design and purchase a printed circuit board or PCB. Since the equipment needed to manufacture PCB's personally is out of our reach, it will be necessary to send out designs to an online company that can then manufacture what we need. The cost of this process is one that must be considered with caution since much of the price of our unit could come from the PCB purchase depending on where it comes from. Ways to diminish costs would be to minimize the size and complexity of our circuit.

The PCB we will purchase will be one that utilizes a standard two-layer setup. The way this is configured is in the middle of our board will be an insulator substrate, typically a fiberglass material, that will spate our two layers of copper trace. This copper will be what connects each pin to its necessary network. On top of the copper will be some form of mask that protects the bare copper from having interference. Our boards will have a composition similar to the picture in figure 29:

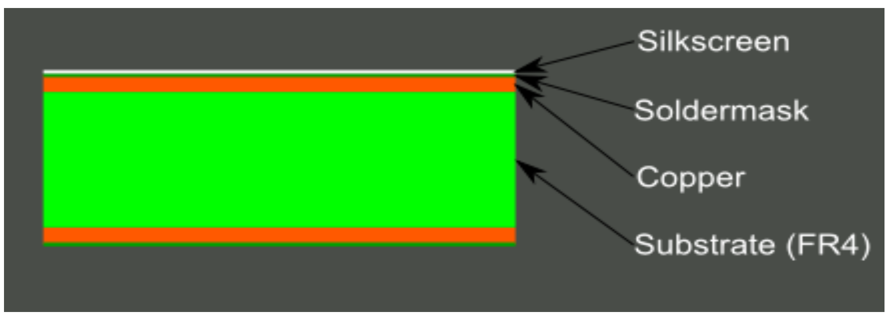


Figure 29: Picure showing typical layering of a 2 layer PCB. (36)

Another important factor to consider when designing a PCB is the number of through hole vias needed. These are instances when a bridge is needed to connect one layer to the other. This is typically utilized when you need a trace to cross over another. By bridging to the other side and then back we can jump over that trace that was in the way. This is needed a lot in devices that are small like ours due to space constraints. (36)

In our Eagle design, each connection is shown with a yellow line. This is how it looks before the routing process is started. Routing the board is when we tell the program exactly where our traces are going to go. This process can be tricky and will require great care. Any design issues that are not caught by the error program of Eagle will result in a faulty PCB. To help us in this process Eagle has a few tools that can tell us when there are problems. This is called the design rule check or DRC. It will make sure there is ample spacing given between traces and vias. This assures that no lines will be shorted or interfered with.

Figure 30 shows the initial design of the PCB. While the position of the components will surely change over the course of the design process, all components are shown. Once we finalize all desired parts through prototyping, our design will be complete and will be routed in Eagle.

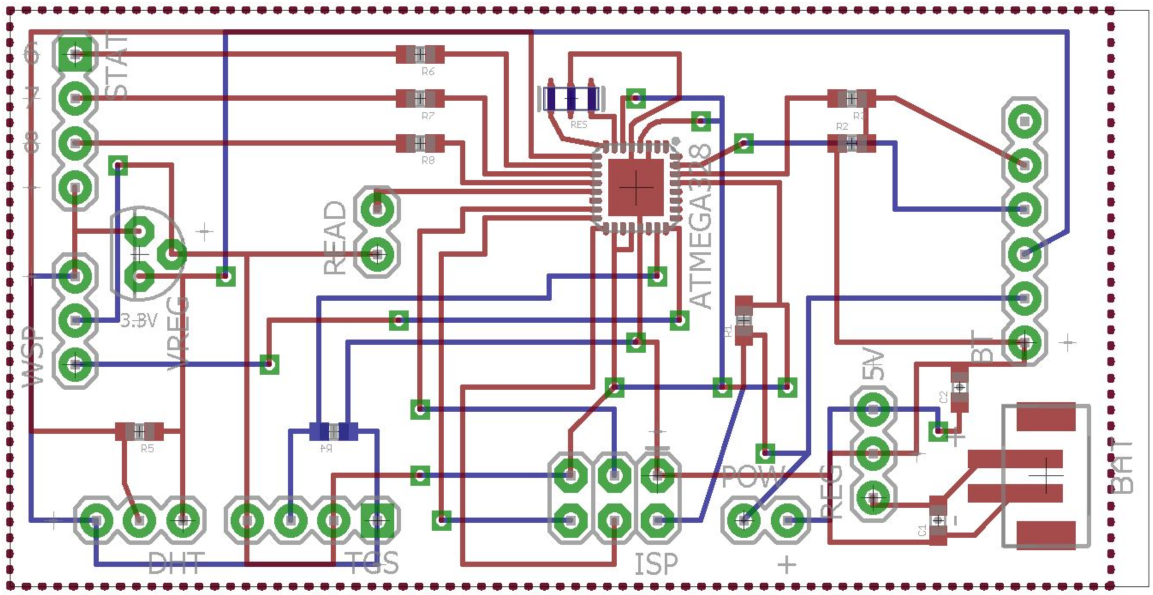


Figure 30: Original PCB Eagle deisgn.

6.3.2 PCB Vendor Info

The vendor we will be using is an online company called Advanced Circuits. They have relatively low prices for small, single order boards. There are also student discounts available to us through the company. Our board will be small and thus easily manufactured. Advanced Circuits has many low cost options for simple circuits, which is exactly what we will need. Their student options will allow us to purchase a single two-layer board instead of having to reach a certain requirement. They have bare bones boards, which have no mask and are not assembled. Due to the extra labor this would require of us, it is not a viable option. Their current student options are shown in table 21:

PCB 2 Layer Spec	**$33**	**Up to 60 sq inches**	**No minimum order**
PCB 4 Layer	**$66**	**Up to 30 sq**	**No minimum**

Spec	inches	order

Table 22: Information regarding chosen PCB manufacturer Advanced Circuits. (15)

There are other options available from the site, however they all have features that will not be necessary for our project. Because of this we can narrow down our search to the two options above. And considering that a 4 layer board will not be needed to achieve or desired design, the 2 layer 33 dollar option will be the one we choose.

Advanced Circuits is a good site because of its support of student engineers. Through their student-engineering program, if we get our board delivered directly to the school, we will not have to meet the usual minimum of 4 boards that the 33-dollar option typically comes with. This is great since we only need one board to do our testing on and eventually to run our device. Many companies require a minimum order since mass production of PCBs is far cheaper, however for our needs, a single cheap board is a much better option. (15)

6.4 Code Layout

As stated in the design section, the code is organized into 3 main units. Clearly, the Android application and the device software are very clear separate units, only connected on occasion by the Bluetooth link; but, the Android application is split into the overarching framework, which contains the user interface and storage components, and the conversion code, which will handle the conversion of inputs into blood glucose values and the creation and updating of the best fit line or curve from which to calculate those values. A text description of the layout that will be attempted first for each unit follows, which will be followed by a flowchart for each of the units that will be followed as closely as possible for the creation of the initial code.

6.4.1 Phone Application Framework

The main Android application framework (along with all other parts of the Android application code) was created and updated using the Android IDE and the Android Studio program and environment. The code was all be written in the Java language given that that is the native language for Android applications and it proved more than sufficient for all of the requirements for the application. A code framework and basic user interface was built using tools from the development kit in order to have a place from which to write the rest of the functions.

The first major action taken by the application upon launch, as well as the first major section of code to be written and tested for, is to populate the active memory of the device with all of the previous readings that are stored in the long-term memory of the phone. Upon pressing the button to take a reading, it will attempt to locate and connect, using the Bluetooth link, to the breathalyzer device. If this connection is not detected or made within a reasonable (specific length to be determined based on testing) amount of time, then the application will return to the menu screen. This is a standard screen displaying options, including the ability to view previous readings as well as to try again to receive a new reading.

If the connection to the device is made, it will wait to receive a message that either includes a reading or stating that no readings exist in the device memory. If a reading is sent, the application will send them to the conversion code and save the converted value, original value, and date and time information to the memory. It will then display the results of the most recent reading received to the user. This display currently only includes the received value, but will include the converted value, as well as information regarding the health implications of the converted value upon further work and especially development of the actual conversion code.

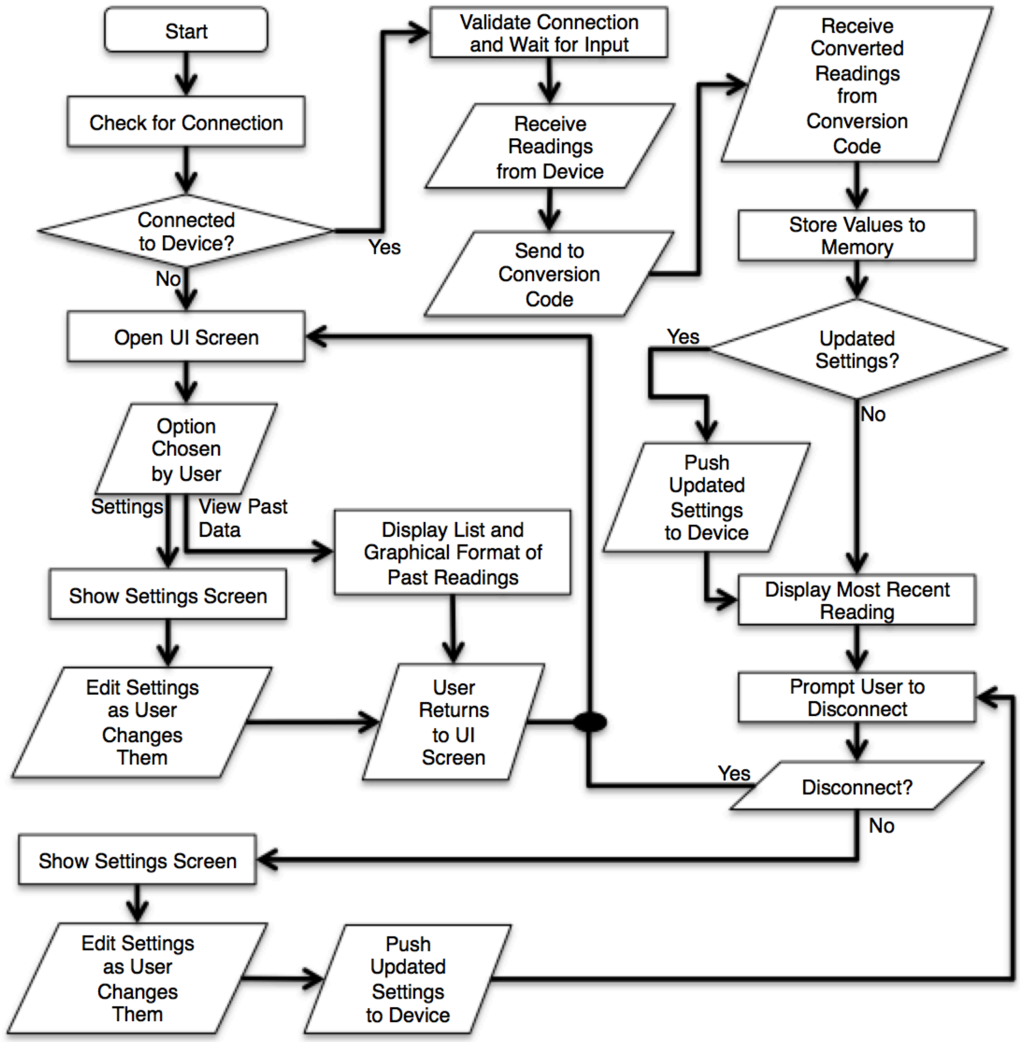


Figure 31: Initial Flowchart for Application Framework

Depending on the value post-conversion and how the program reacts to it, the program will either recommend immediate action taken by the user or prompt the user to either allow the device to disconnect from the phone or to maintain connection for a short time in order to allow for some setting changes. If the device is disconnected then the application will go to the standard screen with selectable options in order to function without a connection. A full flowchart of the initial design concept that the code was based on, and reiterating some of the preceding information and also providing a more effective visual layout of the code as well as a baseline for coding while the project was actually being built is included above. This flowchart shows the main organization of the code as well as the loops that the code falls into based on user and connectivity inputs from when the code was initially planned. This is not exactly accurate to the current state of its functionality, though a lot of what is there that does not exist is planned to be put in place with more future work on the device.

6.4.2 Conversion Code Layout

Currently, there is nothing but a placeholder where the conversion code is to go as the device is not sufficient to reliably and accurately predict blood glucose values. However, the potential in future iterations does exist and so the plan for the layout of that code is as follows. The conversion code will exist to handle converting values from the raw breath acetone values calculated within the device to the potentially more important and relevant blood glucose values calculated for each individual. This code will also be written in the Java language and created using the Android IDE and likely the Android Studio program and environment. The code will be a somewhat separate entity from the Android application framework but will still exist within the application and will therefore be just another major part of that overall code.

The first major action involving the conversion code will occur upon being called by the Android application framework and it will be sent the full data set for the reading being handled. The conversion code will then store a copy of the value in extra variables to be returned at the end of the code. The code will then take the received breath acetone value and prepare to convert it to a blood glucose value.

The value will be multiplied into the equation for the line or curve of best fit calculated in advance and then upon completion of the mathematical operation, stored to output. The application will then prompt the user asking if this value is being corroborated by an external and more accurate meter or not. If the user answers no, the value will be sent back along with the rest of the data to the application framework code where it will be reacted to and stored.

If the user instead answers yes, then the application will prompt the user for his tested value and check to see how accurate the predicted value was. If the predicted value was very close to or exactly what the tested value is, then the program will simply add the new value and reading to which it corresponds to the set of points (likely saved as an array) and leave them without updating the coefficients. The projected accuracy requirement to meet this condition will be solved for during testing.

If instead the value is outside of a reasonable range, it will be added to the array and the program will re-solve for the line or curve of best fit, with the new point weighed slightly more strongly than the older points. The new coefficients will then be saved to the coefficient variables in memory and held to use for the next input. The results will still be sent out to the rest of the code to be saved and reacted to. A full flowchart reiterating the preceding information and also providing a more effective visual layout of the code as well as a baseline for coding while the project is actually being built follows.

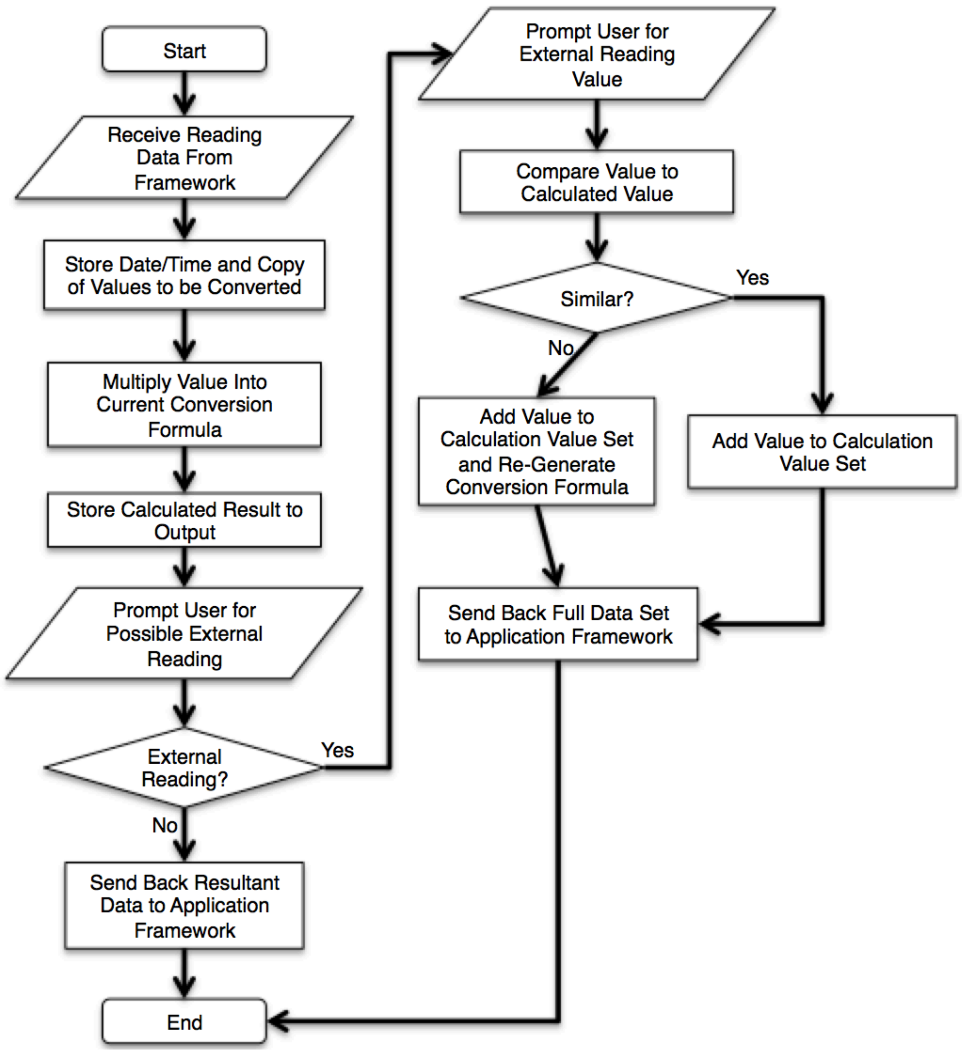


Figure 32: Flowchart for Conversion Code

This flowchart shows the main organization of the code without going into entirely too much detail about the specifics of the conversion formula, as there is not enough information currently to even create a reasonable framework for that.

6.4.3 Device Code Layout

The device code was written in the C code language that is standard for Arduino devices. C++ is also viable and standard for this application, but since C is the more familiar coding language it was the one used. It was written using the Arduino Software IDE. The organization scheme is relatively simple as the tasks required of the device code are also relatively simple. The device has a startup set of code as well as its idling loop, button press loop and reading storage, handling, and sending functions.

The power-on sequence initiates as soon as the device has the power to turn on, which includes a power switch being turned on to connect the device to power. The first major responsibility of the device is to ensure that the sensors are being readied to be used, as, if the device has been off, especially for an extended period of time, the sensors are definitely not in a position to be used. The device begins heating them and remains in this powering-on state until it detects from baseline readings that they are ready to ensure their accuracy in sensing. The device displays a solid one color to the LED during the beginning of this power-on sequence, and there are small extra charging and battery power LED's that the user can also see. The LED color will turn off and not react unless the user presses down on the button after a small initial startup period.

In its idle loop, the device waits for either button press detection and then it sends its code to handle the button. The main purpose of the idle state is to maintain the functionality and heat of the sensors so that they can be used as needed but to also save as much battery power as possible. Upon button press, the device checks with the sensors and prepares them for taking a reading rapidly. During this time the user is exhaling through the device through a full breath. When the sensors are ready to read, they wait for the user to release the button and then the sensors will send their values to the main code of the device. When these values are received, they are converted according to the data and conversion graphs in the data sheet by using the temperature and humidity sensor readings to their actual breath acetone values, and then a weighted average of several readings over time is used to ensure the accuracy of the reading.

When the Bluetooth chip detects and makes a connection with the correct Android application, it will implement a reliable communication protocol in order to send the data error free to the application. The device then sends the reading in a specified manner in order to allow for the device to read all readings the same way and handle them properly. Then, it transmits the value and then returns to the normal waiting loop.

A full flowchart reiterating some of the preceding information and also providing a more effective visual layout of the initial code plan as well as what was a baseline for coding while the project is actually being built follows.

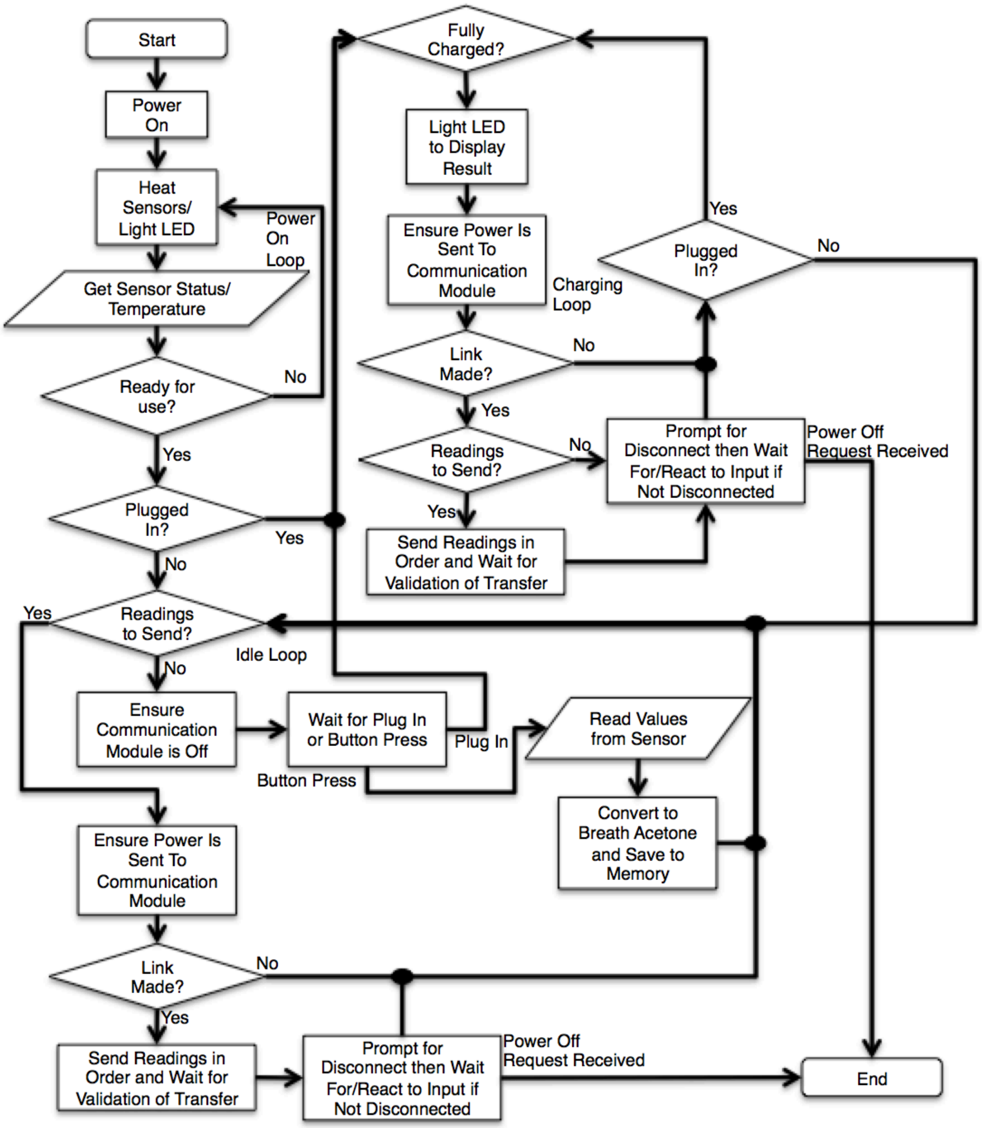


Figure 33: Flowchart for Device Code

The above flowchart shows the four major sections of the code that were used in the device and effectively visually represents the organization of the coding section of the device as a whole. This does not very well reflect the current version of the device code, though it does include some planned future functionality that didn't work very well with the current version of what the device could do. This includes chiefly the ability for the device to manage the power state of the Bluetooth chip.

7.0 Project Prototype Testing

Building our prototype and determining a proper testing procedure for it is an important part of overall device operation. In this section, we will discuss how we plan to carry out this out.

7.1 Hardware Prototype Environment

Now that all parts and designs have been laid out, it is important that the environment in which testing takes place has all the necessary requirements met. In order to assure that the PCB we design and order will work as expected, the design must be proven at the breadboard level. Breadboard prototyping is a very important step in device creation. It allows the most freedom to decide whether or not certain designs will succeed and if not, how to properly redesign them in order to function. Once this breadboard configuration is built and tested, we will be able to move on to order and perform similar tests with the PCB.

Our testing procedure, on breadboard or PCB, will be composed of a few important steps.

1. Creating functional communication system with serial connections using the Bluetooth module. Without this there is no way to read the data while we test and calibrate.
2. Assure VOC sensors are stabilized at the baseline value found during calibration. It is very important that the analog reads we take in are stable before any further testing can be done.
3. Check values that the DHT22 temperature and humidity sensor are producing are accurate. If the DHT22 malfunctions in any way, the readings form the VOC sensors will be incorrect.
4. Confirm that the computed resistance ratio is being processed properly through the use of our specified equations. There will be multiple equations that are used that are specific to VOC sensor type and to the read ratio, so it is important that these are used properly.
5. Once the data is confirmed to be accurate, the communications system must be completed with crating a function Android application to compute, store, and manage the data received.

Once these steps are accomplished with the breadboard prototype, the process will move on to PCB design and testing. The end goal is to be able to duplicate the proper results found during prototyping for our fully integrated system on the PCB.

Using an ISP configuration we will be able to program a stand-alone ATMega328p chip on the breadboard using an Arduino Uno as the programmer.

Doing this allows us to use the Arduino IDE to program our device. A regulated 5V power source will be needed to run the chip along with a 16MHz crystal to handle the clock. Below are designs created using the open source breadboard planning software PEBBLE. All designs will be implemented on a single breadboard for prototyping needs, but are shown separate here in order to be able to clearly see all needed specifications.

7.1.1 Breadboard Design

Programmer

The circuitry required to run the ATMega328p requires only 5V of DC power, a pull up resistor on the reset pin, and a 16MHz crystal connected to pin 9 (XTAL1) and pin 10 (XTAL2). Once this is completed, the programming interface can be created. We will use the standard 3x2 ISP header configuration to program the chip. The ATMega328p has 3 pins that are utilized during the programming process along with a connection to the reset pin. The ISP connects to pins 1(RESET), 17(MISO), 18(MOSI) and 19(SCK), along with connections to 5V power and ground. The Master Input Slave Output (MISO) and Master Output Slave Input (MOSI) pins are used to communicate data between the chip and programmer. The Serial Clock (SCK) pin is used to properly synchronize data communications. This setup can be seen below in the PEBBLE drawing in figure 34.

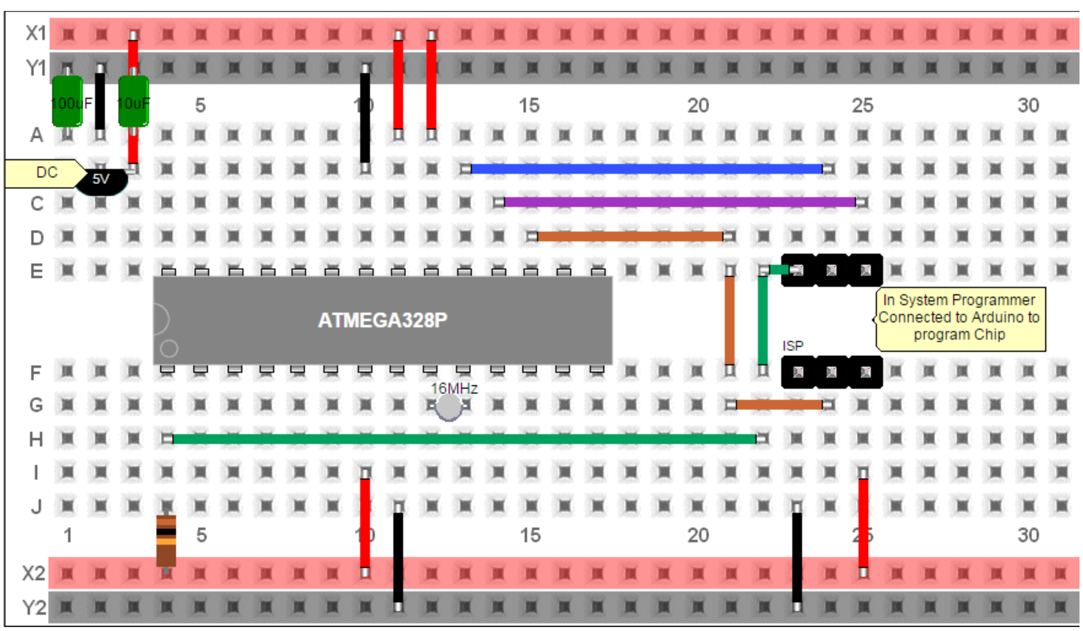


Figure 34: Original PEBBLE breadboard design needed to program MCU.

When it comes to actually testing and programming our device during the prototyping phase, it may be beneficial to have a separate programmer that our chip can be plugged into and programmed quickly. The reason for this is that certain pins that are used to work the device may also be used to connect the chip through the ISP configuration. This could cause undesirable actions to be taken or for the programmer to no properly recognize the chip. To combat this issue, a separate standalone device can easily be put together that would give the basic infrastructure to run the ATMega328 and make the ISP connection readily available. This device could use a ZIF header, which would allow for easy removal and implementation.

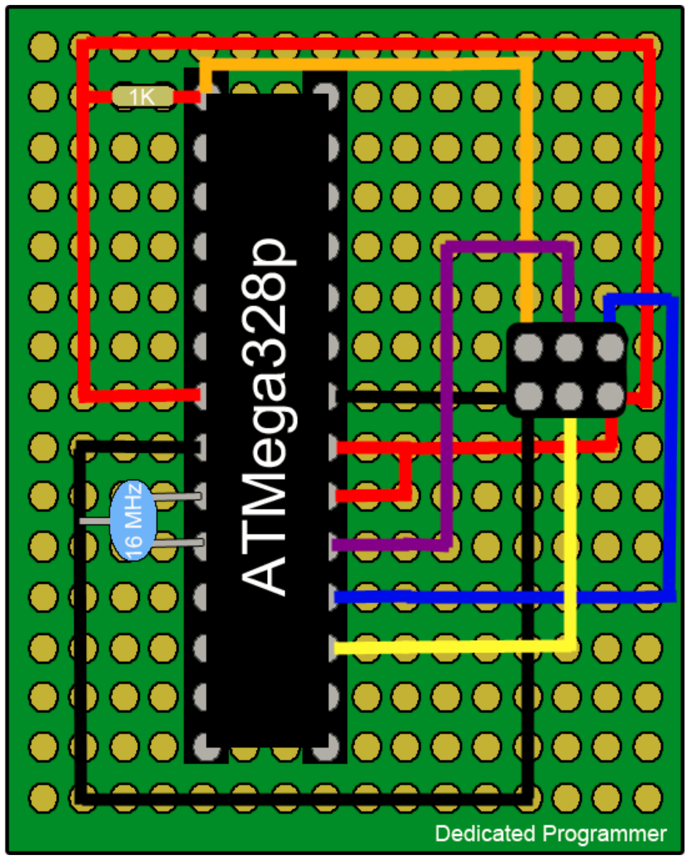


Figure 35: Original drawing of dedicated programmer board.

Figure 35 shows the proposed separate dedicated programmer. As stated before, there will be headers in place to make the removal and placement of the ATMega328p chip easy, since there could be a lot of movement back and forth during testing. The circuit uses all the same components as the breadboard design, but will be put together on a separate perfboard. All connection will be made through wire wrapping and soldering those wraps on the necessary pins. An additional feature that could be implemented would be a simple red power LED that would assure that the device is getting power. However this may be unnecessary considering it will be attached to the Arduino or some computer interface in which the power can easily be seen as on or off.

Sensors

The main part of the unit will be the sensor modules. The two VOC sensors will require 5V DC at a current under 300mA total. The ATMega328p connects to one of the sensor pins through its analog pins which is the output voltage of a voltage divider circuit. That output is measured through the analog pin and using a specified resistor value (4.7KΩ) the resistance of the sensor is found mathematically. It is this resistance that is the indication of concentration of volatile gasses and will be polled continuously when the read button is pressed. The actual pins used on the Arduino are pins 23 (A0) and 24(A1).

The other sensor is the temperature and humidity sensor (DHT22). This is a four pin module but only 3 of the pins are used. The DHT22 will require 3.3V of DC power which can be achieved through the use of a 3.3V linear voltage regulator connected to the 5V source. The DHT22 can be connected to any digital pin on the ATMega328p to transmit data, so we choose the closest, pin 14(PB0). The last thing this sensor requires is a pull up resistor connected from the 3.3V DC power to the data pin on the DHT22.(40)

It will be important to be careful while prototyping because of the differences in currents among modules being used. The ATMega32p has a max current rating on each pin of 40mA. This is fine for all other components except the VOC sensor heaters. They will draw up to 300mA of current, so consideration needs to be had while working with the higher power VOC sensors to avoid damaging the chip. The desired setup can be seen in figure 36.

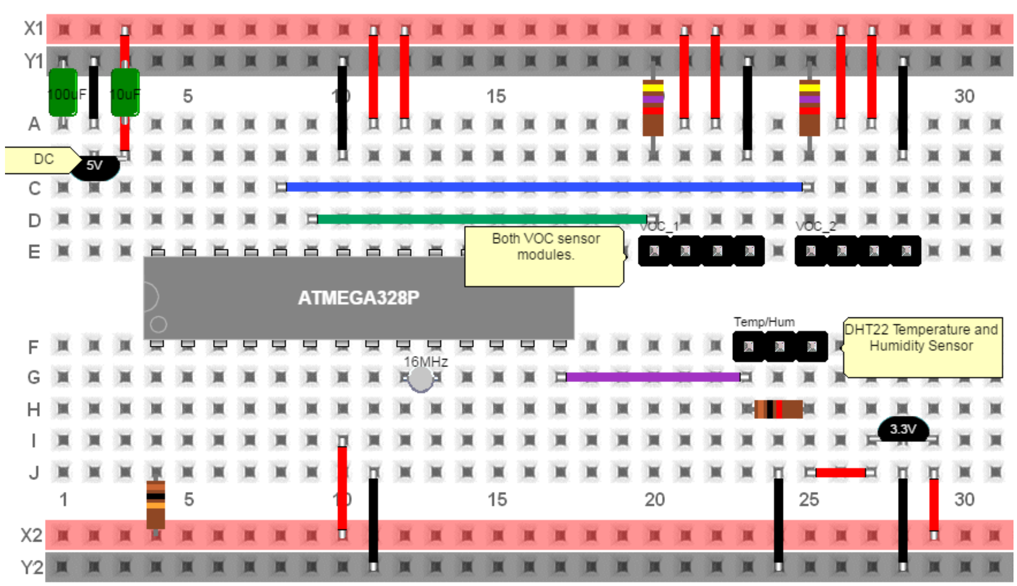


Figure 36: Original PEBBLE breadboard design to run and test all three sensors.

Bluetooth Communication Setup

In order to test the Bluetooth module we will be making use of the serial pins on the ATMega328p. The HC 05 module has 6 pins, but only 4 are actively used. The module runs on 3.3V of DC power so one pin will be connected to the regulated 3.3V and the ground. The other two pins are connected to the RX (pin2) and TX (pin3) pins of the ATMega328p, which are the serial communication pins. Using a terminal emulator on a Bluetooth communication enabled computer we will be able to receive the data form the HC-05 being read from the ATMega328p. A likely software to help us utilize this is the open source Tera Term, which will set up the terminal emulator. The ATMega328p circuitry will remain the same, as seen in figure 34. The Bluetooth setup should come near the end of testing since we need to be able to transmit the proper data form device to computer. Until we can procure that data, the Bluetooth setup is unnecessary. (39)

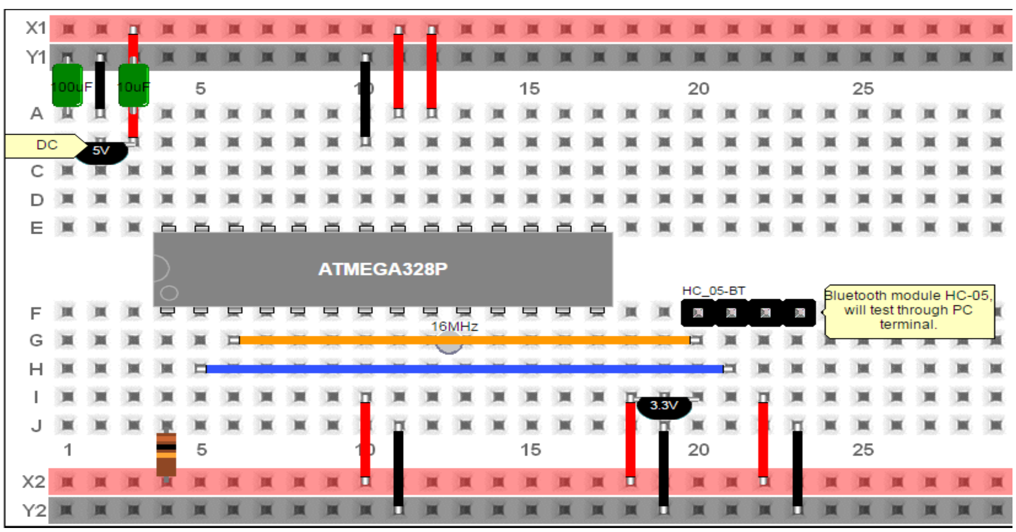


Figure 37: Original breadboard design to run HC-05 wireless communication.

Entire Setup:

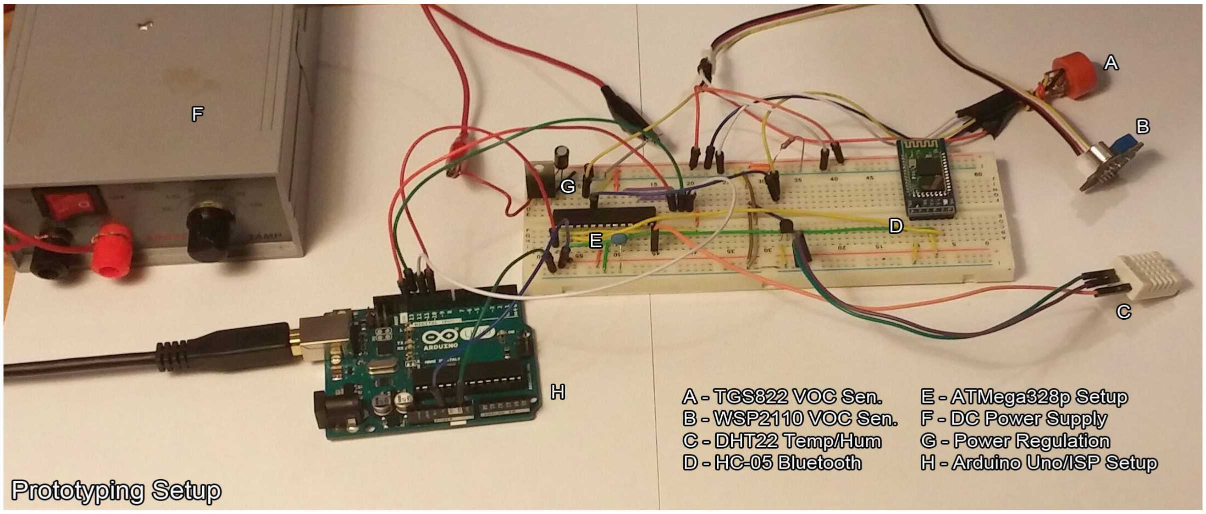


Figure 38: Original picture of our prototyping setup including a variable DC power supply.

Figure 35 shows a current picture of the prototyping setup for the device. All of the important components are labeled. The current setup is in the early phases of testing and does not exactly match the designs created above using PEBBLE. However, everything is connected in the same way and the references of the PEBBLE drawings will be very useful throughout the testing process. The setup above has the ISP connections hooked up to the ATMega328p along with everything else. However, since the Bluetooth module uses some of the same data paths, they can interfere if the Bluetooth module is connected while trying to program. This can easily be solved by creating the standalone programmer that will connect directly to the Arduino. Using this, all we must do is move the standalone chip itself around instead of having to rewire the device each time our code changes, which will be a lot during the initial testing procedures.

7.1.2 Testing Procedure

The testing procedure is going to be a long and drawn out series of tests to collect as much data as we can and compare with a control to determine if there is device functionality.

7.1.2.1 Sensor Testing

There will be extensive testing carried out for the sensors in a number of different ways in order to make sure the design is working as it should be. We can assure the accuracy of our tests by using the urine test strips that give ketone readings and compare that with our sensor readings. Calibration is going to be required and we will handle that by using the following methods to compare results so that accurate conclusions can be drawn:

- Ketogenic Diet
- Fasting
- Healthy Human
- Diabetic Human

A ketogenic diet can be obtained by following guidelines to help create a metabolic change in the system. It is mainly a low carb diet. This works because the body typically runs on carbohydrate or glucose based fuels, but with the elimination of such foods, the body is forced to rely on fat metabolism products. The diet forces the body to burn fats instead of carbs. Normally carbs are converted by the body and turned into glucose for energy and brain functionality. Without intake of carbs, the body turns fats into fatty acids, which produce the ketone bodies, replacing the glucoses job of running the body. This method is going to be hard to implement with our team because one of us or all of us will have to start this diet with enough time to make sure that our body starts burning fats and producing ketones to measure with the sensors. Fasting will also produce the presence of ketone bodies in the system for the same reasons as a low carb diet. People fast for different religious beliefs and other cleansing reasons and this causes the body to be without any food intake for a long period of time. When this happens, there is no more carbs or glucose in the body for it to run on so the body must resort to burning fat in order to stay alive. This, once again, will provide the body with ketones and allow us another method for measuring ketones with our sensors. As stated above, both methods will be checked and assured with the use of an existing accurate urine test to have side-by-side results between our measurements from the sensors and the measurements from the test strips.

The other two methods we will use to assure accurate sensor calibration is to compare the results from a healthy human and a diabetic human. The heathy human will show a small range of ketones that will probably be measured by our sensor that picks up lower concentrations and the diabetic human will show a range of ketones that are generally higher than the healthy human which will be measured by our sensor that picks up higher concentrations. The range for a healthy human is typically less than 10ppm whereas the range for a diabetic is much greater than that. When the diabetic is under extreme conditions, there is a very high level of concentration of ketones that will be found. However, this will also be hard to implement because we do not want anyone to risk unhealthy conditions for the sake of this project. So, we will compare our readings between the healthy human and the typical diabetic which will always show a relative enough of a difference between them. Also, as stated above, each of these methods will be assured by the accuracy of a urine test to be sure that our sensor is correctly calibrated.

By the end of this chain of tests, we will have produced enough data to compare our measured values from the sensors with the existing accurate urine test strips. This allows for us to create conclusions and add any other computations or ratios needed to draw connections between our sensors readings and actual ketone levels found in the urine. We will also be sure to do a side by side blood test with these urine tests and sensor tests to be sure that all the readings match up to the same conclusion. We will present this data with a graph or chart that has multiple series of measurements taken and the readings they provide for a visual on our testing methodology.

7.1.2.2 Communication Radio Testing

The first step in testing the Bluetooth radio will be to assemble the previously mentioned breadboard layout. Once this is functional, a link will need to be set up between our standalone BT radio and a viewable display. Eventually the destination for our data will be to show up on an application on mobile phone. Until that application is built and usable, we will use a Bluetooth communication enabled computer to pair with the HC-05 module. On the computer, we will make use of an emulated terminal. The easiest software to use for this appears to be Terra Term, an open source software. Terra Term will allow us to see the data received from the ATMega328p over the serial communications, on the computer screen. By making sure that data received is expected and consistent, we can assure later adoption to other communication recipients will be seamless.

7.1.3 PCB Testing

Once our PCB is designed and manufactured, the next step will be to fasten all necessary headers and connectors with the use of a soldering iron. Since many of our components are external to the PCB, there will be a need for either soldering headers to the PCB or soldering the wires of the sensors directly to the board. Depending on difficulty, one way may be chosen over the other.

Once all connectors are properly fastened, the battery will need to be connected, and voltages checked. It is very important for our VCC voltage to be stable, as our math to determine sensor resistance has little room for error. Along with the 5V VCC, we will need to assure that the 3.3V regulator has the proper output. During this time, with the use of a multimeter, we should check that pins that need to be high are high when the power is turned on, and pins that should be low are low.

Once we make sure all the connections are correct, we will attempt to make a connection with either our dedicated Android application or through an emulated terminal on the computer. Once this connection is made, we can monitor the data coming in. Using this observation we can get a better understanding of warmup times and expected values. From here we can assure that the success found in the prototyping is duplicated with the PCB.

7.2 Software Prototype Environment

In order to prototype the software, initial coding and testing took place in the simulated IDE environments for both the device Arduino software and the Android application software. Both are capable of simulating the running environments of the devices they are designed for, so both of those simulations were used to get initial prototyping of the code while the actual board is being built. When a sufficient level of effectiveness has been reached with the setup and function of both code sets, the codes will then be actually sent to the devices for further testing, especially of the Bluetooth environment, and more iterative testing and problem solving will be performed in that situation.

7.2.1 IDE Used

There were two IDE's used for this project, one for the development of the Arduino code and one for the development of the Android application. The Arduino code was written using the Arduino Software IDE for ease of use and compatibility. The Android application was written using the Android Studio program and IDE to provide for the most manageable and easy development environment for that as well. Both of these listed programs contain the ability to effectively simulate the device environments as well as provide significant support and ease to the coding procedure of each section of the code.

The Arduino Software IDE is a free program designed for use with the Arduino boards that can be downloaded from the official Arduino website. It contains a lot of information and an easy native environment that uses C or C++ to code the board. In addition, the website and program provide support for using the board as an AVR ISP that allows the board to run with a programmer. This allows for easier prototyping and coding of the board as well as more direct interfacing with the hardware for the software code.

The Android Studio program is a Java IDE that is provided by the Android developer team and contains a lot of prepackaged code and layouts to vastly help with the creation of user interface screens and interactions as well as code snippets and information about proper usage and functionality as a whole. It provides an environment designed for use with the creation of Android applications as well as a potent simulator to help with the iterative design process over the course of the time spent building the application.

7.2.2 ISP Programmer Setup

As long as an external programmer chip is made available, Arduino boards perform extremely well in the context of using an ISP Programmer. In fact, as they run on AVR boards already, conversion into an AVR ISP is relatively simple and easy to do. It is even a native function on the Arduino Software IDE and can be performed very easily using an external programmer or even a second Arduino board as the programmer.

The advantages to using the board as an AVR ISP and having a programmer setup are clear in that the design process, especially for the code, required significant iterations in order to perfect sensor calibration both in terms of calculating even just the amount of breath acetone present in each reading (using provided data to calculate this against sensor voltage and resistance as well as readings from other sensors) as well as the more difficult task of calculating blood glucose from this. In order to rapidly change the code scheme or even the setup of the Bluetooth module, as well as the specifics of the conversion and interpretation of each reading, a setup for the device involving a programmer clearly had its advantages in the context of this building process.

7.2.3 Testing Procedure

Clearly, the program required a significant amount of testing and iterative design as well as significant scrutiny (especially with respect to calibrating the readings of the sensors and the Bluetooth communication) in terms of finding a design that meets all requirements and allows for improvement and achievement of some of the more higher-end goals especially within the realm of programming. As a result of this, the testing procedure is probably by far the most important part of the entire programming process, as no real progress can be made without first gathering some information from the sensors and then moving into a situation where it can be processed and handled at a much more detailed level.

However, the sections of code that allow for proper functioning over a longer period of time of the device, chief among them the idle state and maintenance of the functionality of the sensors, are of similar functional importance and will be tested for and proved functional before any further more complex work can be done. In addition, the most relevant follow-up step is to create meaningful Bluetooth communication between the device and the Android application, so that real-world tests from start to finish of the entire code can be performed as soon as possible.

7.2.3.1 Sensor Code

The calibration of the sensor code can actually be split up into two major sections. The breath sensors need to be coded to convert the electrical parameters of the sensors, which are the raw data that is returned to the device, into meaningful relevant values that can be handled from a biological perspective. This code needed to be as accurate and precise as possible in order to eliminate exponential error from arising due to further conversion of those values, and so definitely required testing and fine tuning as well as detailed attention to the returned values of the other supporting sensors and how they affect the readings. In addition, the returned values, after being sent to the Android device, needed to be run through code that converts them based on an entirely original algorithm and handling structure into meaningful blood glucose values, although this algorithm was never fully developed or achieved during the scope of this project. This would have require a significant amount of testing and was easily the most intensive software design goal for the project, and additionally would have required several further iterations of hardware design and setup to make plausible.

As a large portion of the testing process was to be spent perfecting this conversion code and putting as much work into ensuring the device can actually function for its intended purpose, the testing procedure was critical in many respects. This procedure entailed establishment of a baseline measure and return value and ensuring that it always read functionally zero for a blank reading of standard air. The members of the group underwent diet changes in order to put their bodies in states that mirrored different states that the sensor and coding needed to handle, as well as ensuring that it could actually function for a diabetic person, as exclusively functioning for those who have effective insulin control in their bodies would be entirely useless. Many readings and stored values for many people in many states were used to test the reliability and accuracy of the device, and ideally it will be discovered that the device can in fact be used in its intended capacity, even if the required code and conversion algorithms got highly complex.

7.2.3.2 Communication Radio Code

A highly important piece of the project and something that needed to be tested and perfected very early in the design is the functionality of the Bluetooth communication used in the device's use. Since the device could function effectively without the ability to connect to and interact with the application on a regular basis, it was imperative that this connection be made reliable and effective as soon as possible to allow for the rest of the device's testing and perfection to be realized. That being said, it was important to not rush this process and ensure the communication is working well. There was already a framework laid out and effective communication made between an early test of the device and chip that seemed to properly be able to communicate with a computer that had compatible receiving hardware. Since this was already being done effectively, it stood to reason that setting the phone up to connect to the device in a similar fashion wouldn't be altogether too difficult, and allowing permissions to the application as well as ensuring proper reception, parsing, and handling by the code of the received values was simply be a function of the coding of the application itself and not of the communication scheme. However, managing the communication protocol and ensuring effective communication as well as detection of communication by the device were very important things to keep in mind, as the device has to perform several functions based on detection of communication. In the end, the actual duration of the radio testing was very short and the coding only had to take the device into account in order to be reasonably functional over communication.

8.0 Administrative Content

Designing this project takes numerous amounts of administrative content to carry out a procedure a great as this one. Coordinating between four group members can be difficult, so one person stepped up to take on the leadership role, assigned tasks and kept track of time. Within our allotted time period, we were able to become quite the team and made sure everything that needed to be done, was done correctly and efficiently.

8.1 Initial Budget and Finances Discussion

Our project is based upon the idea that we can design and implement a breathalyzer with low cost and low power. These features are hard on budget because, ideally, the lowest power consuming parts and components will cost more due to the fact that it's low power. We have to keep the cost down due to the specs of this project as well as our own student incomes. The following section discusses how we managed to meet our own requirements and still keep the price low enough for our own incomes and still implement a device that can be considered low cost.

Funds are an essential part when managing a lot of materials that should be acquired between four gathering individuals. Every gathering part needs to concur with the cost of every segment. They additionally need to concur on what the general spending plan ought to be, especially with this being a huge undertaking with various separate parts that need to work appropriately prior to the end of Senior Design II. This makes the significance of every segment even that much more prominent. Additionally, all of the four gathering individuals are understudies that have low wages. During the school year, a large portion of us doesn't work in light of the additional work burden it would give us on top of our schoolwork. Therefore, the majority of us don't have much cash to spend, particularly on additional things, for example, a school project. A large portion of an undergrad's wage goes to lodging, educational cost and nourishment. Therefore, the spending should be something that individuals can accomplish without taking on a monetary weight that could place them in a risky position.

Towards the start of our task proposition, we as a group talked about the way that every individual will have the capacity to, at most, spend roughly $75 on the project. Having every party in the group consent to this gave us a most extreme spending plan of $300. This sum may appear to be plenty for a project yet for our situation we have to spend it wisely. A portion of the parts for this anticipates alone could cost close $100. A case would be buying a breathalyzer. Nobody in the gathering as of now possesses one or knows somebody who might give us a chance to obtain theirs. Therefore, the purchase of one could be important for our project. If we somehow happened to spend close 10% of our financial plan on a solitary item it would eliminate limitations on the greater part of this project.

Alternate parts may likewise not end up being cheap either. Consider the possibility that a vital part of our own design, the PCB, can't be fabricated for under $150 dollars; our financial plan would just have $150 left in it. Furthermore, imagine a scenario in which a section glitches, we would need to buy another. Therefore we thought that it was important to precisely go through everything required and put a perfect most extreme value we would spend on it. Obviously as we go and search for every component we will attempt to locate the least expensive choice with expectations of sparing cash on the grounds that as a general rule it would be decent if every party would not have to burn through $75 of their own money. Despite the fact that we might want to locate the least expensive models of every segment, that may not be conceivable if certain parts have certain configuration requirements that must be managed.

One part of financing that could end up being very useful is having a supporter or sponsor. Sponsors can offer an extraordinary measure of budgetary backing if necessary. Contingent upon the kind of supporter our task gets will rely on upon how that sponsor will need to help us in our task. The sponsor for our project could either give us cash that would go towards our financial plan, diminishing the sum each of us have to spend for the task, or they could supply us with a breathalyzer that we can analyze. Preferably in the event that we were hoping to get a Breathalyzer from our supporter we would contact a doctor’s office in Orlando. In actuality any sort of sponsorship that our team would have would be beneficial to our budget, whether they need to help our project monetarily or help by supplying parts.

Another part of funding is the manner by which our group will split the costs equitably. Numerous ways have been talked about so as to locate the best approach to ensure that the costs have been conveyed equally amongst each other. One approach to do this is while everybody is making purchases of parts there will be a document that contains what was purchased by who and the amount it cost. Toward the end of the task the totals would be summed up and the costs would be split uniformly amongst every party. Another alternative would be that once one person achieves their most extreme of $75, that individual would not buy any more segments for the project. Given the chance that toward the end of the project we find that the costs between gathering individuals are not equivalent, we would then continue to split the costs uniformly. A third choice would be that at every purchase made the cost of that part would be split by four and every member would add to that individual purchase. This would keep the need to ascertain the aggregate costs and split the expense toward the end of the project.

Presently in spite of the fact that the third choice is by all accounts the least difficult it might be hard to maintain. On the off chance that a person at the time of purchase does not have the required amount of money the concept won't work. Therefore, as a team, we have chosen to not utilize this choice. Instead, the second choice was settled upon. This alternative was consistently voted in favor of for the most part on the grounds that an individual could never need to surpass $75. Additionally, on the off chance that one individual goes through $75 and every single other individual spent not as much as that individual would get some cash back so as to make a harmony between everybody.

In the event that the project winds up getting a supporter, we will adjust the collective choices on the best way to divide the costs uniformly. In the event that a sponsor is gained, the choice that would be most reasonable would be the first alternative. We would start by utilizing whatever the supporter will give and after that if there still need to purchase parts it will originate from every individuals own pocket. Toward the end of the project the aggregate costs would be summed up and afterward split equally between every person. This choice is by all accounts the best because of the way that with a sponsor it would be impossible that somebody will burn through $75.

The original budget is shown in Table 23 below:

Project Budget

Part	Quantity	Prices
High Concentration Volatile Organic Compound (VOC) Sensor	1	<$10
Low Concentration Volatile Organic Compound (VOC) Sensors	1	<$10
Temperature and Humidity Sensor (DHT22)	1	<$10
Microcontroller (ATMEGA32)	1	<10
Power Supply (Lithium Ion battery and Charger)	1	<$60

Bluetooth Module (HC06)	1	<$10
Aluminum Cashing	1	<$30
PCB Fabrication	1	<$200
Android Application	1	<$30

Table 23: Approximation of expenses and budget prediction

Total: <$370.00

Note: The parts tabled above may be subject to change upon completing this project. Some parts might not be suitable to make this project work to the projected requirements.

We picked this financial plan after a lot of exploration of every individual. While inquiring about the breathalyzer we needed to utilize, costs went from $75 to an unfavorable amount of cash. As a group we didn't see the need in purchasing a costly breathalyzer because of the way that we are just utilizing it for this project. Likewise, the breathalyzer is not the primary center of the task so we would prefer not to spend the whole project budget on it. Consequently, we have to put aside approximately $90 for the cost of the parts needed to build our breathalyzer.

DHT22 Sensor

We wanted the temperature and humidity sensor to have characterized the highest standard and the DHT22 provided it. This part of our project will cost $8 roughly including shipping.

ATMega 328p Microcontroller

Another aspect of this project is that we need a microcontroller that's commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno and Arduino Nano models. With that in mind, we decided to use the ATMega328p controller. The Atmel 8-bit AVR RISC-based microcontroller combines 32 kB ISP flash memory with read-while-write capabilities, 1 kB

EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8 and 5.5 volts. The device achieves throughput approaching 1 MIPS per MHz. The microcontroller will cost roughly $4.

TGS822 Sensor / WSP2110 Sensor

The TGS 822 sensor is used for the higher concentration levels found which is pertinent to recording spikes. The WSP 2110 sensor is used for the lower concentration levels found which is pertinent to recording major drops. The specs for these sensors can be found in their respective data sheets, which are provided in the appendix. Both of these sensors have been chosen because the pricing and power consumption is low enough for our low cost low power design. The TGS822 was priced at about $6 not including shipping, which is very affordable. We went ahead and bought two of these for any anticipation of needed another during the testing period. The WSP2110 costs about $17 for one, so it was a bit pricier but provides more in one purchase. The WSP2110 consumes less power than the TGS822, but both sensors are very low power.

8.2 Initial Milestone Discussion

Team Member 1:

Senior Design I during the summer was a brave move because we had to write the project paper of the same amount of 120 pages in less time. Summer semesters consist of only 12 weeks instead of 16 so we had to write our final paper rather quickly. It took a little bit of time to get to know each of the group members to feel comfortable around them and for assigning tasks. In the beginning of the semester, it took quite a long time to get going. The pace of the class was too slow. Our initial 10-page document took way longer than I imagined it would. This was due to the class schedule because it wasn’t due until a month into the semester. By that time, we should have had close to 30 pages completed, but we only had 10. This was a huge set back, but we cranked out some pages fast after that to try to make up for it. Weeks 4 to 8 went generally smoothly as everyone was trying to get their pages written. When we had our draft document discussion with the professor, he said we were on track. So that was good. But by week 9, I got a little worried about not having our total pages written on time, even though I already had most of my 30 written. As crunch time got even closer, everyone came through and our total pages got written on time. As a whole group, we stayed pretty much on top of things and were able to work together nicely when it really counted.

Team Member 2:

One of the hardest parts of this project to date has been proper time management. Due to the shortened schedule for completion, our success hinged on being able to stay on top of all tasks, at all times. We got off to a slow start due to the fact that a single idea had not officially been decided, and also because once that decision was made, there was no real urgency to get many pages written. This was partly due to the fact that there were no page amounts due to begin the class. But as deadlines of pages written approached we all began to get on track. By staying in touch over group chat and occasionally meeting up on campus to discuss who still needs to do what, we were able to reach our end goal. From the very beginning we were able to divvy up the required sections. It was important for us to be able to do this early, since we had plenty of time to tackle small parts instead of everyone going off and doing the wrong things. Once the professor confirmed that we were on the right track during our second meeting, the rest was just finishing what we all had already started, making the completion that much easier.

Team Member 3:

Considering the fact that Senior Design I is designed to be taken over a 16-week period instead of a 12-week period, time constraints can be a huge issue. The initial couple weeks in the class moved very quickly, requiring rapid group formation and getting deeper into deciding on a project and being prepared to get up and go at what seemed like a completely unreasonable speed. However, this time-efficient process proved to be absolutely necessary and almost not enough. Getting together the pages for each step in the process was difficult and required a lot of intensive work from everyone involved. We all had our work cut out for us and close and difficult timelines to meet to keep up with all of the deadlines. Keeping to the milestone and timelines decided on at the beginning of the semester was difficult and imperfect, but as a whole our group has managed to come together, iron out what needs to be done, and really put together a solid paper that not only meets the requirements but avoids many of the common mistakes described to us throughout the course. Instead, this document contains almost exclusively group-authored documents and pictures and diagrams and a lot of effective description and text that goes into solid detail about what is being done. In addition, the group came into this project quickly with a ton of parts and designs and ideas prepared and ready to go as soon as this specific task was decided on, and this allowed our writing process to move much more smoothly and effectively throughout the semester.

Team Member 4:

Project Milestones offers reference to the time when real occasions in the project, is utilized to screen the general advancement of said task. The points of reference for a task ought to exhibit a reasonable grouping of events that will incrementally develop to the result of the validated project. Project milestones not only help your group remain focused, they are likewise valuable to the project manger to all the more precisely figure out if or not your task is on timetable. There are different techniques in which a group can utilize to monitor project turning points. The techniques that we used to monitor our task development were week-by-week meeting, a Gantt outline, and chapter-by-chapter guide. We additionally utilized online resources to habitually upload information on each of the finalized sections of the project.

As week after week meetings and gatherings and table of contents are straightforward, we will investigate what a Gantt outline is. In project administration, Gantt graphs are utilized to demonstrate the exercises, for example, the assignments and occasions plotted against time. The main Gantt diagram was formulated in the mid-1890s by Karol Adamiecki, a Polish specialist who ran a steelworks in southern Poland and was inspired by administration concepts and strategies. Around 15 years after Adamiecki, Henry Gantt, an American designer and administration specialist, came up with his own particular form of the graph and it was this that turned out to be generally well known in western nations. Thusly it was Henry Gantt whose name was to wind up connected with diagrams of this style. In its initial origin Gantt graphs were to some degree impeded in their convenience as these outlines were made by hand and would have to be redrawn every time the task change or made overhauls. This issue would change with appearance of PCs and task administration programming, because of the way that, as overhaul is required, the PC can reproduced the diagrams with the new upgrades in a way or minutes.

To make our Gantt graph we utilized Microsoft Excel because it already had a nonexclusive format design of this outline. Be that as it may, keeping in mind the end goal to make a Gantt diagram without any preparation in exceeds expectations one must do the following:

Step 1 is to "Develop a Work Breakdown Structure". This is a process by which a project is planned by breaking it into easily definable and understandable goals, milestones and tasks. Listing the major components first is the first step in developing a WBS. This can be done by in a word processor, spreadsheet, or using a Gantt chart program.

A key component in the WBS is to plan for proposed results, instead of arranging activities. That is, comprehend what the objectives of the project are, characterize key breakthroughs, and after that begin the way toward separating those pieces into tasks. In the event that there are altered dates that should be met, ensure those are appeared in the Gantt graph. Along these lines, as the points are broken into undertakings, it will turn out to be clear forthright whether more assets should be added to meet these due dates

The second step to “Assign Tasks”. One of the most critical pieces in how to build a Gantt chart is the distribution of work. There are several things to consider.

- Who is most qualified to complete this task?
- What is their availability as compared to their currently scheduled workload?
- What is a reasonable expectation of their time needed to complete the task(s)?
- Will additional people or resources be necessary to get these tasks completed on time?

For our case, utilizing SmartDraw to make a Gantt graph, we have two choices for doling out tasks. We can either include the name of the individual doled out every task specifically into the Gantt diagram, or we can tap on the Assignment View catch.

Step 3 is to “Build Timelines”. After the work breakdown is finished and tasks are relegated, the way toward building timetables happens. This should be possible in various ways. When utilizing a spreadsheet program, one can make segments with dates every assignment starts and ends. If this is done, you'll need to ensure you have a timetable open so you don't coincidentally plan assignments to be done amid weekends and occasions. When you make a Gantt outline with SmartDraw, you can manufacture your course of events in various ways. You can sort dates or time length specifically into the sections, or drag the bars to extend or decrease the course of events for every assignment. SmartDraw additionally demonstrates weekends and major occasions, to help you abstain from booking undertakings or meetings on nonworking days.

Step 4 is to “Evaluate Task Dependencies”. One of the advantages of making a Gantt outline for project planning is that it makes it simpler to see conditions. This is a circumstance where one task is dependent upon the finishing or result of another. For instance, a website admin can't assemble a page unless the marketing specialist and artist have completed their task. Distinguishing these dependent connections is basic, as postponements in the essential strides will in all likelihood swell through the whole project. Automatic programming will permit you to include conditions as you make your Gantt diagram. In case you're doing this by hand or utilizing a less complex project, you'll have to recall to do this critical stride physically.

The Final step is to “Share and Evaluate the Plan with Your Team”. At the point when the Gantt diagram is finished, show it to colleagues for audit and input. It's essential that every individual from the project signs off on the arrangement quickly. This guarantees the arrangement is precise and sensible. It's much less demanding to take into account possibilities, arrange extra assets, or even propose a reconsidered plan at this stage, as opposed to at a basic crossroads later.

Project Milestone:

Summer Semester Timeline	
Week 1 (5/16-5/20)	• Group selection • Brainstorm and examine project ideas • Begin to think about finalizing the project choice • Learn more about the specifics and the goals of each project
Week 2 (5/23-5/27)	• Finalize project choice • Begin work on Divide + Conquer Initial Document Assignment • Discuss roles and each person's focus • More specifics about what is needed for the final project • Begin more intensive research about the details of the project
Week 3 (5/30-6/3)	• Finish Divide + Conquer assignment • Discuss the project more in depth • Assign roles and focuses for each member • Have a good set of research and requirements for what will make the project work
Week 4 (6/6-6/10)	• 30 minute meeting regarding Divide + Conquer Initial Document assignment

	• Update submission with regard to what had been discussed in the meeting • Begin a more detailed research for preparation into specifics of the final design
Week 5 (6/13-6/17)	• Work on more detailed constraints and standards from lecture • Begin design with more in depth exploration of the specifics of the project • Expand the document with findings
Week 6 (6/20-6/24)	• Continue work on the standards assignment • Work towards a more specific and more finalized designs and part selection • Work towards the specifics of the pieces of the final document • Start to break up the final document to each group member
Week 7 (6/27-7/1)	• Work on Table of Contents submission • Have a specific plan for how the full document will be completed • Have at least 60 pages completed
Week 8 (7/4-7/8)	• Put together a reasonable draft of the final document • Continue editing with table of contents • Finalize parts research/selection • Work on eagle design • Create more detailed plans for each group member
Week 9 (7/11-7/15)	• Complete the eagle design for the project

	• Continue to work on the final document
Week 10 (7/18-7/22)	• Compile all updates and changes into final form of the Final Documentation • Check for reliability, feasibility, and any mistakes/oversights.
Weeks 11&12 (7/25-8/2)	• Last minute checks/updates • Be sure that the format is correct and page count is exact • Final Document is due

Table 24: Project milestone for summer semester

Fall Semester Timeline	
Week 1 (8/22-8/26)	• Prepare for semester • Begin gathering parts • Make sure everything is in order to begin working
Week 2 (8/29-9/2)	• Gather initial and baseline parts and materials • Go over designs again with group and ensure their quality and feasibility
Week 3 (9/5-9/9)	• Ensure that we have all of the necessary parts and materials needed to make our device • Work on any necessary conversions needed for acetone to ketone analysis • Work on circuitry and putting together

	functional breadboard
Week 4 (9/12-9/16)	• Work on code for phone application • Work on communication between the device and the phone • Write any other needed code for computations and conversions from the sensor output
Week 5 (9/19-9/23)	• Continue to work on circuitry and breadboard • Work on code for phone application and its wireless connectivity • Be sure that sensors are working and output is accurate • Begin troubleshooting on completed parts
Week 6 (9/26-9/30)	• Have meaningful progress and the beginning of a functional prototype for both code and circuitry for whole device • Achieve effective demonstrable use of sensors and parts
Week 7 (10/3-10/7)	• Begin working towards completed printed circuit board design(s) • Continue troubleshooting on circuits and code • Focus on power consumption and make adjustments accordingly
Week 8 (10/10-10/14)	• Put in the order our final printed circuit board design

	• Put together initial iterations of full device with reasonably completed code
Week 9 (10/17-10/21)	• Give the printed circuit board a week to come in • Test and troubleshoot printed circuit board(s) with the rest of the device (including size, sensors, code, power, wireless connectivity)
Week 10 (10/24-10/28)	• Continue testing and make necessary changes needed to make the device work • Check for any power consumption issues
Week 11 (10/31-11/4)	• Major data collection from testing and optimizing device for effective detection and algorithms for the code
Week 12 (11/7-11/11)	• Continue data collection and algorithm generation • Major testing/troubleshooting/error correction of the device
Week 13 (11/14-11/18)	• Finish final work on project • Close all tasks that are still on-going • Make sure that all group members are done with their tasks and help each other if needed • Begin putting together final presentation
Week 14 (11/21-	• Continue to work on putting together the

11/25)	final presentation • Have a plan to show that our device works • Prepare demonstration of working device
Week 15 (11/28-12/2)	• Final preparation for final presentation and demonstration • Be sure that each member has completed all their assigned tasks and are ready to present on them
Week 16 (12/5-12/12)	• Final presentations/demonstrations

Table 25: Project Milestone for fall semester

9.0 Project Summary and Conclusions

In conclusion, with a few minor adjustments, creation of code, and testing procedures, we will have a working Diabetic Breathalyzer.

9.1 Project Feasibility

Our project will be feasible contingent on two main conditions: that the hardware and software communicate correctly and that the sensors are calibrated to output a value that is accurately relatable to blood glucose level.

Since we are a group of four electrical engineers, the software will be the most difficult part for us to implement. We believe that with the previous software experience from one of our group members, we will be able to make sure the communication features of this project work correctly. We spent the length of Senior Design I trying to plan out the code and how the software will work, such as communication methods, memory banks, etc. We have multiple flow charts to follow showing how we hope to implement the code during Senior Design II. Also, by using the Arduino IDE and its libraries, programming the communication portion will be much easier. There are many available libraries to run the temperature and humidity sensor as well, which will also make things easier. Our project feasibility greatly depends on this communication between devices. Fortunately, we have been working with the hardware throughout Senior Design I because it was needed for this paper. We already have all the parts and components ordered and have mapped them to the breadboard, so the main part of the hardware that we need to carry out for feasibility is ordering the PCB and testing that to make sure it is up to spec as well.

Our VOC sensors measure acetone concentration in the breath; however, there will still need to be a number of repetitive tests carried out to gather data on how the output from the VOC sensor is related to blood glucose level. We will use traditional methods of blood glucose measuring, such as finger pricks and urine strips, to determine this relationship. We will also compare between healthy humans and diabetic humans to be sure that our device is working correctly. Once we have a large amount of data to analyze, we will determine the relationship between the value of acetone concentration in breath and the actual blood glucose level that it corresponds to. Without this relationship, our device will not fulfill its main task of serving as an option for diabetic monitoring.

We do not aim to be as accurate as a certified method like blood tests and we do not advertise as a replacement but more as an additional option for daily diabetic monitoring. We hope to provide a range of blood glucose levels that the user's breath will relate to. This will let the user know whether they are healthy and where they should be versus experiencing a spike in the glucose level. This is the most important part of daily monitoring and our project will provide a noninvasive method to execute it.

9.2 Project Operation

The operation of our device is described throughout the contents of this paper. It will include a small total unit size that can be easily transported for easy daily use. When the user would like to check their blood glucose level, they will blow into the mouthpiece. While blowing, the device will show an LED that will let the user know how long they need to keep blowing for. This is so that the sensors can collect enough breath to measure acetone from. The value from the sensor will be processed and computed through layers of hardware and software to produce a number that is recognizable to diabetics. Using a Bluetooth module, we will communicate the devices data with a smart phone for easy access to the user's data from the day. Our device operation will be low cost and user friendly.

In order to be sure that the project is operational, it will return a value that is parallel to a number that measures blood glucose levels for diabetics. This will endure a lot of testing and sensor calibration and will be the most meticulous part of the project to carry out. By cooperating with our group members and with the help and resources from the University of Central Florida, we will achieve our end goal. After building the hardware, programming the software, and analyzer sensor data, we will be able to build ourselves an operating Diabetic Breathalyzer.

Even though each group member has specific assigned tasks to focus on throughout the duration of Senior Design II, we will be working together as a team and helping everyone troubleshoot and debug. Our project operation depends on the cooperation between the team and the ability to work together to find issues and fix them. We shall become the expert at our specific task along with learning how to adjust it accordingly in order make the project operational.

9.3 Project Conclusions

Throughout the course of designing this blood glucose breathalyzer device, immeasurable knowledge was gained through research and perseverance. This is a lengthy project for our team to do over the course of about 2 full semesters, keeping in mind that we have unreliable schedules of on and off work while in school. Since our project was not a typical one, we had to do much research and investigation on breathalyzers, diabetes, ketones, VOC sensors, and the relationships between them. One of our team members has relationships with individuals with diabetes who felt outdated by the process of pricking fingers to monitor blood glucose levels. This can be considered a gift to them and others who have to deal with the traditional method of the finger pricking process. Our team's high-level knowledge of diabetes was not enough to sustain the amount of knowledge needed for this project. We learned a great deal through the research carried out and the collaboration between members of the group.

We designed the schematic in a timely matter and were able to incorporate the several features we had in mind. We wanted it to be relatively small and hand-held so consumers could carry it in their pockets or purses and for it to be self-contained. Because we wanted it to be as small as possible, we knew the enclosure was going to be crammed with parts. It was determined to be unnecessary to make it in such way that the consumer would have to take things apart and replace batteries, so it was decided that it needed a rechargeable battery. Having too many AC adapters to charge various devices can be very irritating, so we wanted to make the device charge over USB. We wanted the interface to be as simple and user-friendly as possible and came up with a way to do it with only one button.

Although this project was very time consuming, a great degree of teamwork ability was gained. Attempting to schedule meetings to satisfy the calendar of every team member didn't seem too difficult but there we some bumps along that road. The main factor that allowed us to complete the research part of this project was all of the team members setting their egos aside for the greater goal of completing this project. We knew what lied ahead of us and the sacrifices it would take to accomplish our goal. One aspect that really satisfied our expectations was the budget. It's very unlikely to come up with a senior design project that has the right amount of computer engineering and the electrical engineering tasks within it to satisfy the faculty standards, and we were able to accomplish that. All four members of our group are electrical engineers, so the software tasks will be more difficult for us to implement than the hardware. We are fortunate that one of our group members has experience in the software we wish to use for our project, so that will help the next semester move smoothly along on the computer engineering side of things.

A handful of knowledge on the designing a PCB was gained, we learned that printed circuit boards were invented in the 1930's and are utilized as a part of an assortment of electronic circuits. The utilizations range from straightforward one-transistor enhancers to super PCs. Printed circuit boards are a huge piece of innovation in the cutting edge world. They can be found in many items, for example, autos, toys, TV's, PCs, phones, stoves and so on. The spacing on the boards is more important than most people think. PCB's are priced based on the complexity of the design. No one in our group has had previous experience in this area of designing and ordering a PCB so we had to start from nothing and work our way up to learning our how to use Eagle to display our design on the PCB and how to order it. This is a huge part of today's electronics industry and our new experience in this area will be greatly valued.

One of the biggest roles in this project is the VOC sensors. This is another area that no one in our group had much experience with and we got to learn so much about them. We aim for our breathalyzer to have the ability to measure acetone in the breath and believe that we have made the correct computations and assumptions so that it will be able to. There will be a series of complicated testing that will be carried out throughout the next semester when the entire device has been built. Upon testing, we will be able to calibrate the sensors how we want them and learn so much more about the process needed for building electronics with sensors.

With the combination of our ideas, research, design, and collaboration our Diabetic Breathalyzer is set up to succeed.

10.0 Experimental Results

After carrying out extensive testing for our experiment, we found both expected and non-expected results. Because of these large spread results, we spent ample time determining how to create a valid conclusion.

10.1 Sensor Results

All three of our sensors did their respective job. The temperature and humidity (DHT22) sensor was able to keep track of the temperature and humidity of the breath that was entered into the chamber. However, the humidity shot up to 100% every time it read the breath, which makes sense since the breath is extremely humid. We were unsure if this was going to create problems down the road from maximization of the sensor every time it was in use. But even with the repeated use, it continued working as planned. The high concentration sensor (TGS822) also did its respective job in measuring the high concentration values when it was needed. Since we did not often get high concentration readings, this sensor was not highly in use. The low concentration sensor, on the other hand, was used much more often. Since our team was the primary source for testing and we are all healthy, non-diabetics, the majority of our readings came through this low concentration sensor (WSP2110).

Our biggest obstacle regarding the sensors was determining valid and stable baseline resistance values. This was difficult because open air is variable. So, depending on the location of testing, whether it is a place of residence, the senior design lab, or the presentation conference room, the sensor readings for open air can fluctuate. These baseline values are important for several reasons. Without a baseline resistance, the ratio and variance of the resistances caused by the breath would mean nothing because they wouldn't have a reference point. This would make our results non-relatable and incomparable. The baseline basically provides a reference point in which the sensors can react upon and determine a value against. In order to fix the issue of irregular baseline values, we integrated the sensor stability with our LED found on the outside of the breathalyzer container. Before blowing into the device, hold down the button until the LED turns from blue to green. When the LED is showing green, this means that the sensors have reached a stable baseline value and the device is ready for use.

10.2 Hardware Results

Upon original prototype testing, the hardware setup communicated nicely with the Bluetooth module. Our set up allowed for a real time response from breathing into the device to displaying a continuous graphical representation of the sensor values. This was very helpful throughout troubleshooting and determining valid sensor readings. Our prototype testing set up is shown below in Figure 39. Since our project is highly dependent on values received from testing, we had to be able to begin our testing procedure before the time costly shipment of the PCB arrived.

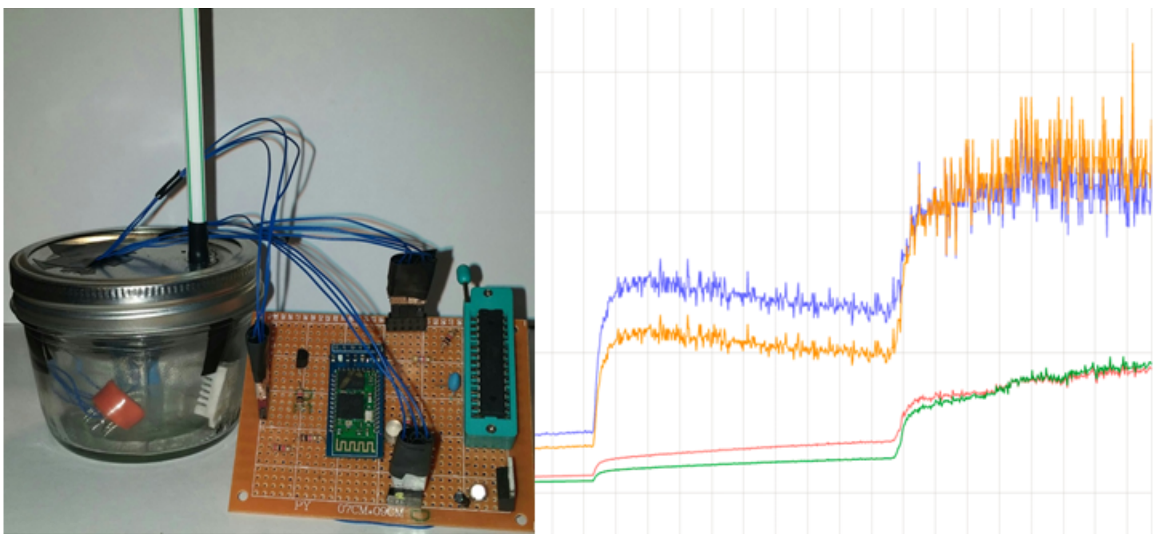


Figure 39 : Prototype testing set up with real time response displaying resistance results (blue and orange) and acetone breath concentration values in parts per million (red and green).

Once the PCB arrived, we were able to create a final product with a more accurate test set up. We worked very hard to make the PCB as small as possible so that it would be able to fit on the side of our final container. When the PCB arrived, it was slightly bigger than expected and did not fit perfectly in the box. But, we were able to arrange it differently to fit inside the box without disturbing any of the other components. With the finalization of organizing the inside of the device container along with the device code and phone application code, we were able to create a finalized testing set up, shown below in Figure 40.

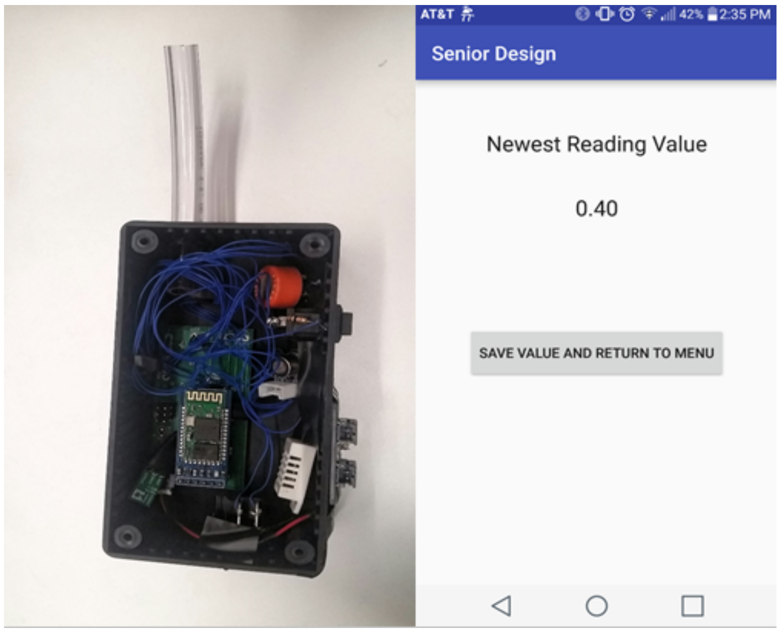


Figure 40 : Finalized device set up and application output used for continued testing and to determine results.

10.3 Software Results

The software was easily able to show a meaningful acetone level from the sensors and transmit it to the phone application in a reasonable and fast way. The levels found on the phone application were stored effectively and through only a small amount of testing and reorganization of some of the reading functions and some minor errors initially when testing the phone application Bluetooth connection, the code was made effective, quick, and usable. There are definitely upgrades to be made, both in terms of actually developing a conversion algorithm for the phone application and allowing the device board more fine control over its different parts. That being said, a few of the requirements for the device were unable to be met not because the code was unreasonable or impossible but because the hardware design was not refined enough for these goals to be met, and, as there was only time enough for one iteration, it was effectively impossible to meet these goals within the scope of the project.

However, they were shown to be very reasonable and achievable with a small amount of effort and time in the future.

10.4 Human Results

In order to draw real conclusions about the functionality of our device, we needed to be able to test it on humans of all sorts of health. We began testing our device on ourselves, who are all healthy and not a diabetic. Sometimes, even with no change in the breath, the sensors would get jumpy. This is where we learned that we needed a valid and stable baseline resistance value in order to have consistent results. When the sensors were stable, the typical value that our breathalyzer would output for a normal healthy human would be around 1 to 2 ppm. That is such a small amount of acetone that it is basically negligible, which is exactly what we expected for a person without diabetes. We also took note of these healthy persons ketone levels during this test and all suspects had trace amounts of ketones found in the urine. This corresponds with the trace amounts of acetone found in the breath. In order to see if we could increase our ketones and look for a corresponding increase of acetone, we put ourselves on a ketogenic diet for one week. At the end of that week, we found a slight increase in ketones as well as a slight increase in acetone concentration. This was the result we expected and showed consistency between the relationship of the ketones and acetone, as shown below in Table 26.

	Urine Ketone Level	Breathalyzer acetone PPM	Notes
User 1	5-Trace	1.75	No diet changes
User 2	5-Trace	1	
User 1	15-Small	8.7	After a week of low carb dieting
User 2	10-Trace	6.2	

Table 26 : Summary of results taken from healthy users with normal eating habits and after one week of following a ketogenic diet.

Although this did support our hypothesis that when ketone levels increase, so will the acetone concentration found in the breath from our device. We wanted to be able to support this idea with a more realistic testing set up. We brought in two diabetic users who agreed to using our breathalyzer so we could gather more data. When they used our device with no diet change, the results were generally the same as if a non-diabetic user blew. This is due to having their disease under control. If diabetics take care of themselves, take insulin when they need to, and regulate their food intake according to their personal healthcare plan, they will not see any signs of ketones. Without signs of ketones, the acetone won't show up in their breath either. Scientifically, this is correct. However, we were hoping for a more drastic difference between the results from the diabetic users and the non-diabetic users. Our first diabetic user is a young 20 year old man who keeps his disease under much control. We asked this diabetic user to drink a soda to see if we could find any immediate changes. The values were raised slightly for both the ketone and acetone concentration, but the values were not much different than when the healthy users did their ketogenic diet. The second diabetic user is an older lady who does not have her disease in as much control as user 1. We noticed with her that she needed to take a larger breath when using the breathalyzer to get an accurate reading. The results of both diabetic users are shown below in Table 27. The results provided us with important conclusions about using the device. The most important realization is that there is a time variation between ketones and blood glucose concentration. When user 1 drank his soda, it is not an immediate ketogenic effect. The problem this causes is that the when the blood glucose concentration rises, the ketones eventually rise. And when the blood glucose concentration lowers, there may still be presence of ketones in the body. This creates a time variation in the body that we can't account for. Another important conclusion that was found through testing was that the amount of breath given into the chamber can change the acetone concentration found. With a short breath, there is not enough breath to clear out the original air and give the sensors a new and accurate reading. A longer larger breath will be able to give the sensors more breath to read. Short breath readings are much lower than long breath readings. The breathalyzer readings still correctly correspond with the ketone/acetone relationship even though the results were not as drastic as expected.

	Urine Ketone Level	Breathalyzer Acetone PPM	Notes
User 1	5-Trace	1.2	No diet change
User 2	5-Trace	0.8	Small breath
User 1	15-Small	5	After drinking a soda
User 2	15-Small	8.83	Larger breath

Table 27 : Summary of the results found from two diabetic users and their changes that created a rise in both ketone and acetone concentration.

10.5 Experimental Conclusions

After carrying out extensive testing on ourselves and others, we were able to conclude that our device is able to provide the user with a healthy versus an unhealthy range by blowing into our breathalyzer. Due to the technology available at the time, we were unable to create any direct correlations between acetone concentration in the breath and the blood glucose concentration. One can infer, however, that if they are showing high acetone values from the breathalyzer, that they have a larger amount of ketones present which can lead to the inference that one's blood sugar numbers are high and need to be checked with a glucose meter. This is still helpful in the state that our breathalyzer provides a noninvasive way to tell the user whether they need to prick their finger or not, which can lower the total amount of times that the diabetic would need to use the invasive method. It is also easier than checking for ketones by method of urine analysis. Our project objectives have all been met and completed. Our final device can take in breath, read the levels of acetone, send the final averaged value to our phone application where is displays the value as well as a running log with date and time for diabetic tracking and logging. Our device can be used to help diabetics have another option for daily monitoring that should encourage them to continue proper management of their health.

Appendices

A: Software Use

- SolidWorks 2016, Dassault Systèmes, 2016
- Multisim 12, National Instruments, 2013
- EAGLE 7.5.0, CadSoft Computer, 2015
- Photoshop CS5, Adobe, 2011
- Microsoft Excel, Microsoft, 2013
- PEBBLE, PICAXE, 2016

B: References and Citations

1. Standard.org, ‘ Why Standards Matter’ 2015, [online]
   Available: http://www.standardslearn.org/lessons.aspx?key=1
   [Accessed: 20- June- 2016].
2. FDA.gov,’History’, 20115 [online]
   Available: http://www.fda.gov/AboutFDA/WhatWeDo/History/
   [Accessed: 17- June- 2016].
3. Rick, Booth. ‘Webapp for CE marking: *CE Marking Quick Guide’* 2016 [online]
   Available: https://cemarking.net/
4. FCC.gov,’ What we do,’ 2016[online]
   Available: https://www.fcc.gov/about-fcc/what-we-do
5. Y. Yang, 'Implementation of a colorful RGB-LED light source with an 8-bit microcontroller', *2010 5th IEEE Conference on Industrial Electronics and Applications*, 2010
6. playground, ‘Arduino playground – MQGasSensors’, 2016[online]
   Available: http://playground.arduino.cc/Main/MQGasSensors
   [Accessed: 29- June- 2016].
7. Article Title: This Breathalyzer Will Track Much More Than Your Drunkeness
   Website Title: Co.Design
   URL: http://www.fastcodesign.com/3037036/this-breathalyzer-will-track-much-more-than-your-drunkeness
8. Article Title: 'Smell tech' firms sniff out bad breath, alcohol, air pollution
   Website Title: Investors Business Daily Smell tech firms sniff out bad breath alcohol air pollution Comments
   URL: http://www.investors.com/smell-tech-firms-sniff-out-bad-breath-alcohol-air-pollution/
9. Article Title: This Is Definitely The Coolest Way To Tell If Your Breath Is Smelly
   Website Title: Refinery29
   URL: http://www.refinery29.com/breathometer-mint-health

10. Article Title: The Diabetes 'Breathalyzer'
Website Title: University of Pittsburgh News
URL: http://www.news.pitt.edu/news/diabetes-breathalyzer
11. Henderson MJ, Karger BA, Wrenshall GA. Acetone in the breath: a study of acetone exhalation in diabetic and nondiabetic human subjects. Diabetes 1952;1:188 –93.
12. Sulway MJ, Malins JM. Acetone in diabetic ketoacidosis. Lancet 1970;1:736 –40.
13. Crofford OB, Mallard RE, Winton RE, Rogers NL, Jackson JC, Keller U. Acetone in breath and blood. Trans Am Clin Climatol Assoc 1977;88:128 –39.
14. Article Title: PCB Manufacturers Reviews and Notes
Website Title: ladyadanet blog RSS
URL: http://www.ladyada.net/library/pcb/manufacturers.html
15. Article Title: Advanced Circuits
Website Title: Printed Circuit Board Manufacturer & PCB Assembly
URL: http://www.4pcb.com/
16. Article Title: Custom Printed Circuit Board Fabrication
Website Title: Printed Circuit Board Prototypes PCB Express Fabrication Service
URL: http://www.apcircuits.com/
17. Article Title: E-TekNet
Website Title: E-TekNet
URL: http://e-teknet.com
18. Article Title: PCB board manufacturing, PCB prototyping service & PCB assembling services - Gold Phoenix
Website Title: PCB board manufacturing, PCB prototyping service & PCB assembling services - Gold Phoenix
URL: http://www.goldphoenixpcb.com/
19. Article Title: New Products
Website Title: Olimex
URL: https://www.olimex.com/
20. Article Title: PCB Assembly,PCB Manufacturing,PCB design - OURPCB
Website Title: PCB AssemblyPCB ManufacturingPCB design OURPCB
URL: http://www.ourpcb.com/
21. Article Title: A review of breath analysis for diagnosis of human health
Website Title: Academia.edu
URL:
http://www.academia.edu/8424495/A_review_of_breath_analysis_for_diagnosis_of_human_health

22. Kathy Musa-Veloso, Sergei S Likhodii, and Stephen C Cunnane. Breath acetone is a reliable indicator of ketosis in adults consuming ketogenic meals. U.S. 2002

23. Cao, Wenqing, and Yixiang Duan. "Breath Analysis: Potential For Clinical Diagnosis And Exposure Assessment." *Clinical Chemistry* 52.5 (2006): 800-811. *MEDLINE*. Web. 23 June 2016.
24. Photoinduced Charge Transfer and Acetone Sensitivity of Single-Walled Carbon Nanotube–Titanium Dioxide Hybrids. Mengning Ding, Dan C. Sorescu, and Alexander Star. Journal of the American Chemical Society 2013 135 (24), 9015-9022. DOI: 10.1021/ja402887v.
25. Article Title: OUR DEVICE
Website Title: New England Breath Technologies
URL: http://www.newenglandbreath.com/device/
26. Musa-Veloso, K, SS Likhodii, and SC Cunnane. "Breath Acetone Is A Reliable Indicator Of Ketosis In Adults Consuming Ketogenic Meals." American Journal Of Clinical Nutrition 76.1 (n.d.): 65-70. Science Citation Index. Web. 26 July 2016.
27. Cao, Wenqing, and Yixiang Duan. "Breath Analysis: Potential For Clinical Diagnosis And Exposure Assessment." Clinical Chemistry 52.5 (2006): 800-811. MEDLINE. Web. 26 July 2016.
28. Itoh, T, et al. "Gas-Sensing Properties Of Tin Oxide-Based Volatile Organic Compound Sensors For Total Volatile Organic Compound Gases." Sensors And Materials 21.5 (2009): 251-258. Science Citation Index. Web. 26 July 2016.
29. Lourenço, Célia, and Claire Turner. "Breath Analysis In Disease Diagnosis: Methodological Considerations And Applications." Metabolites 4.2 (2014): 465-498. MEDLINE. Web. 26 July 2016.
30. Fine, GF, et al. "Metal Oxide Semi-Conductor Gas Sensors In Environmental Monitoring." Sensors 10.6 (n.d.): 5469-5502. Science Citation Index. Web. 26 July 2016.
31. Priefer, R, and MJ Rust. "Point-Of-Care Diabetes Monitoring Via Breath Acetone Detection." Abstracts Of Papers Of The American Chemical Society 248.(n.d.): Science Citation Index. Web. 26 July 2016.
32. Chu, BH, et al. "Wireless Detection System For Glucose And Ph Sensing In Exhaled Breath Condensate Using Algan/Gan High Electron Mobility Transistors." Ieee Sensors Journal10.1 (n.d.): 64-70. Science Citation Index. Web. 26 July 2016.
33. Craig Freudenrich, Ph.D. "How Breathalyzers Work" 20 October 2000. HowStuffWorks.com. <http://electronics.howstuffworks.com/gadgets/automotive/breathalyzer.htm> 7 July 2016
34. Antony Monalis, “The Diagnostic Potential of Breath Analysis” 1983 http://www.clinchem.org/content/29/1/5.full.pdf
35. Charles Yim, “Crunch Base https://www.crunchbase.com/person/charles-michael-yim#/entity
36. Article Title: PCB Basics
Website Title: Learn at SparkFun Electronics

URL: https://learn.sparkfun.com/tutorials/pcb-basics

Datasheets:
37. Figaro, “TGS 822-for the detection of Organic Solvent Vapors,” TGS822 datasheet, Revised Feb. 2009. www.figarosensor.com/products/822pdf.pdf
38. Winsen, “VOC Gas Sensor Manual,” WSP2110 datasheet, May, 2013. http://www.winsen-sensor.com/d/files/flat-surfaced/wsp2110.pdf
39. Guangzhou HC Information Technology, “HC Serial Bluetooth Products User Instructional Manual,” HC05 datasheet, https://www.rcscomponents.kiev.ua/datasheets/hc_hc-05-user-instructions-bluetooth.pdf
40. Aosong Electronics Co, “Digital-output relative humidity & temperature sensor/module,” DHT22/AM2302 datasheet, https://www.sparkfun.com/datasheets/Sensors/.../DHT22.pdf
41. Atmel, “8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash,” ATMega328p datasheet, http://www.atmel.com/devices/atmega328.aspx
42. Texas Instruments, “LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator,” LM2596 datasheet, www.ti.com/lit/ds/symlink/lm2596.pdf
43. Kingbright, “Full color led lamp,” WP154A4SUREQBFZGW datasheet, https://www.kingbrightusa.com/images/.../WP154A4SUREQBFZGW.pdf
44. Pololu, “ MQ-3 Semiconductor Sensor for Alcohol,” MQ-3 datasheet, https://www.pololu.com/file/0J310/MQ3.pdf

C: Permissions

Shown below are the permissions pending for the figures and trademarks used throughout the paper:

1: The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by this group is with permission. Other trademarks and trade names are those of their respective owners.

Submit a Support Request Anonymously

This request is being submitted anonymously. If you continue, you will not be able to check the status of, manage, and/or comment on this request. If you want the ability to manage this request after it has been submitted, log in before submitting your request.

Name *

Noah Spenser

Company

Email *

noah.spenser@knights.ucf.edu

Category *

Brand

Subcategory *

Using the Brand

Urgency *

More information

2 - Medium

Visibility

More information

Private

Request Summary *

Brand Name Usage in Documentation by a Non-Company Senior Design Group

Request Description *

I am part of a four-person group currently enrolled in senior design as electrical engineering majors at the University of Central Florida. Our group is looking to incorporate your wireless standard (through the usage of a unmodified pre-built chip attached to the main board in our device) to allow us to have the device communicate with a mobile phone. This project requires a large amount of description and specificity in the design documentation we intend to submit in the near future. In order to make the design and planning descriptions as clear and accurate as possible, we intend to incorporate your word mark as a descriptor of the wireless standard we are using. There is no way for us to register as a company or corporation with the SIG and as a result can't go through most or any of the steps required to use the brand name in our documentation. We are requesting the rights to use the brand name including footnote and ® descriptors for all usage for the purposes of description.

Submit

Thank You For Your Support Request CASE0008268.

Bluetooth Support <support@bluetooth.com>
Today 2:07 PM
Noah Spenser

Reply all

Thank you for contacting the Bluetooth SIG. We received your request CASE0008268 .

Your satisfaction is very important to us and we will get back with you within 2 business day(s).

If you are a Bluetooth member or guest user, you can check the status of your request anytime by clicking your case number and logging in.
If you submitted your request anonymously without logging in, you will not be able to access your request online.

Ref:MSG0057577

Brand Usage

Noah Spenser
Today 2:26 PM
trademark@arduino.cc

Reply all

My name is Noah Spenser and I am a member of a four-person senior design team of electrical engineering majors at the University of Central Florida. We are designing a device that is based on a purchased Arduino board. We are required to submit documentation detailing our design. In order to do so in the most effective and efficient way we would like to incorporate your brand name in order to describe the board being used (along with any usage stipulations or footnote attribution required). Thank you for any help you can provide.
-Noah Spenser

Edert Geffrard
to fdaoma@fda.hhs.gov

Thu, Jul 21 11:24 AM

permission

1 file attached

Hello,

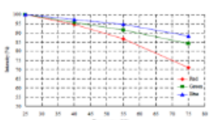
My name is Edert Geffrard and I am conducting a research at the University of Central Florida to make a Breathalyzer as my senior project. It's is with great admiration I you to allow me and my team to use the picture attached to this message to complete our research. Thnk you!

Edert Geffrard
to yryang@mail.mcut.edu.tw

Thu, Jul 21 11:47 AM

permission

1 file attached



Hello,

My name is Edert Geffrard and I am conducting a research at the University of Central Florida to make a Breathalyzer as my senior project. It's is with great admiration I you to allow me and my team to use the picture attached to this message to complete our research. Thnk you!

Edert Geffrard
to The.Secretary@hq.doe.gov

Thu, Jul 21 11:44 AM

permission

1 file attached

Hello,

My name is Edert Geffrard and I am conducting a research at the University of Central Florida to make a Breathalyzer as my senior project. It's is with great admiration I you to allow me and my team to use the picture attached to this message to complete our research. Thnk you!

Sent from Windows Mail